

Metoda najdubljega spusta za brzo numeričko rješavanje polinomnih sustava

Beroš, Ivo

Doctoral thesis / Disertacija

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:683771>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-27**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)





Sveučilište u Zagrebu
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ivo Beroš

**METODA NAJDUBLJEGA ŠPUSTA ZA
BRZO NUMERIČKO RJEŠAVANJE
POLINOMNIH SUSTAVA**

DOKTORSKI RAD

Zagreb, 2020.



Sveučilište u Zagrebu
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ivo Beroš

**METODA NAJDUBLJEGA ŠPUSTA ZA
BRZO NUMERIČKO RJEŠAVANJE
POLINOMNIH SUSTAVA**

DOKTORSKI RAD

Mentor: Izv. prof. dr. sc. Nikica Hlupić

Zagreb, 2020.



University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ivo Beroš

**DEEPEST DESCENT METHOD FOR
EFFICIENT NUMERICAL SOLVING OF
POLYNOMIAL SYSTEMS**

DOCTORAL THESIS

Supervisor: Associate professor Nikica Hlupić, PhD

Zagreb, 2020

Doktorski rad izrađen je na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva,
na Zavodu za primijenjeno računarstvo

Mentor: izv. prof. dr. sc. Nikica Hlupić

Doktorski rad ima: 175 stranica

Doktorski rad br.: _____

O mentoru

Nikica Hlupić rođen je 1969. god. u Zagrebu gdje je stekao i svo obrazovanje. Fakultet elektrotehnike i računarstva (FER) završio je obranom diplomskog rada u prosincu 1993. god. i nedugo nakon studija zaposlio se u razvojnom odjelu ATM-a (danas SIEMENS) gdje je radio na konstrukciji automatiziranih mjerno-regulacijskih sustava za industrijsku primjenu. Ubrzo upisuje poslijediplomski magistarski studij iz područja elektroničkih mjerenja i tijekom tog studija zapošljava se na FER-u. Poslijediplomski studij završava u travnju 1997. god. obranom magistarskog rada. Nakon toga dobiva mjesto asistenta i aktivno se uključuje u nastavu iz grupe predmeta *Mjerenja u elektrotehnici*, kao i u razvoj *Primarnog elektromagnetskog laboratorija* na FER-u. Usporedno s nastavnim radom, nastavlja sa znanstvenim istraživanjima koja su se odnosila na primjenu teorije procjene, optimizacije i statistike te uzorkovanja i računalne obrade signala i podataka u analizama u kojima se zahtijevala vrhunska točnost. Iz toga je proizašla i disertacija pod naslovom *Precizno mjerenje snage metodom uzorkovanja* obranjena 24. svibnja 2001. godine. Po obrani disertacije sudjeluje u nastavi iz predmeta *Procesna mjerenja, Programiranje i programsko inženjerstvo, Algoritmi i strukture podataka* te *Napredni algoritmi i strukture podataka*, a vodi jedan tehnologijski i sudjeluje u nekoliko znanstvenih projekata. U ožujku 2008. godine postaje docent i u travnju 2014. izvanredni profesor u Zavodu za primijenjeno računarstvo. U poslijediplomski doktorski studij FER-a uvodi predmet *Statističko modeliranje i identifikacija*, u studijski program FER3 izborni predmet *Odabrane optimizacijske metode i algoritmi*, a na njegov prijedlog FER uspostavlja *Laboratorij za statističko modeliranje i optimizaciju*.

About the Supervisor

Nkica Hlupić was born in 1969 in Zagreb where he acquired all his qualifications. He graduated from the Faculty of Electrical Engineering and Computing in December 1993. He obtained employment at the ATM (now SIEMENS) where he worked on constricting measurement and regulation systems for industrial purposes. He soon enrolled a postgraduate master's degree in the field of electronic measurements and during the course of the study he obtained employment at the same Faculty. He finished his postgraduate study in April 1997 and became a teaching assistant for subjects related to *Electrical Measuring Techniques* and a participant in the development of the *Primary electromagnetic laboratory* at the Faculty of Electrical Engineering and Computing. Along with teaching he continues his scientific research on the application of the theoretical estimation, optimization and statistics as well as the sampling and computer processing of signals and data in analyses which require high precision. The research resulted in a PhD thesis titled *Precise measurement of power applying a sampling method* which

was defended on 24th May 2001. After defending his doctoral dissertation, he started teaching the subjects *Process Measurement, Programming and Software Engineering, Algorithms and Data Structures* as well as *Advanced Algorithms and Data Structures*. He also led one and participated in several scientific projects. In March 2008 he became an assistant professor and in April 2014 an associate professor at the Department of Applied Computing. He introduced two courses into existing programs. the course *Statistical Modeling and Identification* into the postgraduate doctoral program and an elective course *Selected Optimization Methods and Algorithms* into the FER3 program. *Laboratory of statistical modeling and optimization* was grounded at his suggestion.

Zahvale

Zahvaljujem se svoj rodbini, prijateljima i kolegama s posla koji su me podržavali čak i ako nisu znali točno čime se bavim.

Nadalje, zahvaljujem se članovima povjerenstva, prof. dr. sc. Damiru Kalpiću, prof. dr. sc. Sanji Singer i prof. dr. sc. Danku Baschu, koji su svojim prijedlozima i savjetima učinili ovaj rad boljim.

Posebno se zahvaljujem svom mentoru, izv. prof. dr. sc. Nikici Hlupiću, na sugestiji pri izboru teme i nesebičnoj podršci tijekom istraživanja i oblikovanja rada.

Konačno, zahvaljujem se svojoj supruzi Ani i kćeri Juliji na beskonačnoj ljubavi, strpljenju i razumijevanju koje su mi pružile tijekom pisanja ovog rada.

Sažetak

Tema ovog rada je primjena metoda najdublje spusta na rješavanje polinomnih sustava. Metode najdublje spusta definirane su kao iterativne metode za rješavanje sustava jednačbi u kojima se sljedeća točka iterativnog postupka određuje kao točka na pravcu određenom trenutnom točkom i smjerom spusta, u kojem pridružena funkcija postiže najmanju vrijednost. Zapravo pod pojmom metoda najdublje spusta podrazumijeva se niz metoda koje se razlikuju prema načinu odabira smjera spusta i određivanja duljine koraka spusta, ali im je zajedničko da se uvijek traži egzaktna točka na pravcu spusta u kojem pridružena funkcija postiže najmanju vrijednost. Kao pridružena funkcija koristi se najčešće kvadrat euklidske norme reziduala, ali može se koristiti i neka druga norma.

Nakon opisa postojećih teorijskih rezultata i kratkog pregleda metoda koje se koriste za rješavanje polinomnih sustava, prikazan je razvoj metode najdublje spusta, počevši s kvazilinearnim sustavima i odabirom smjerova koordinatnih osi za smjerove spusta, a osnovna ideja je potom razrađena za opći polinomni sustav i smjer spusta. Kako se za određivanje duljine spusta, osim kvadrata euklidske norme reziduala, može koristiti i max–norma reziduala, detaljno je opisan postupak kojim se izračunava točka u kojoj max–norma ima najmanju vrijednost. Kako su linearni sustavi ujedno i polinomni, njih se može rješavati metodom najdublje spusta te je za jednu inačicu metode dokazano da postupak konvergira rješenju ako rješenje postoji.

Opisane su razne varijante metode najdublje spusta obzirom na način odabira smjera i određivanje duljine spusta (odabir pridružene funkcije). Osim standardnih smjerova spusta poput Newtonova ili smjera najstrmijeg spusta, koriste se i kombinacija ta dva smjera, odabir smjera spusta kao smjera koordinatnih osi, Gauss–Seidelov način odabira smjera te odabir smjera u smislu minimizacije max–norme reziduala. Za skoro sve smjerove napravljene su varijante metode u kojima se kao pridružene funkcije koriste kvadrat euklidske norme reziduala i max–norma reziduala.

Metode su uspoređene s dva referentna algoritma za rješavanje nelinearnih sustava, Min–Pack i NLEQ. Usporedba je rađena na više od stotinu polinomnih sustava koji su se pojavljivali kao test primjeri za evaluaciju drugih metoda. Svaki sustav riješen je mnogo puta s različitim, unaprijed zadanim početnim točkama, a na kraju iterativnog postupka zabilježeno je u kojoj točki je postupak završio i je li to rješenje sustava. Glavni kriterij usporedbe metoda bila je uspješnost rješavanja na nekom problemu, definirana kao odnos između broja iterativnih postupaka koji su doveli do rješenja polinomnog sustava i ukupnog broja iterativnih postupaka. Dobiveni rezultati su pokazali da je većina inačica metode najdublje spusta postiglo bolju uspješnost od referentnih metoda.

Ključne riječi: polinomni sustav, metoda najdublje spusta, numeričke metode.

Deepest Descent Method for Efficient Numerical Solving of Polynomial Systems

This thesis deals with the methods of deepest descent applied on solving polynomial systems of equations.

Motivation

Polynomial systems are a special class of nonlinear systems of equations where all equations are multivariate polynomials. These systems are very important and frequently appear in many application areas: computer-aided design, mechanical design, chemical kinetics modeling, nonlinear circuit analysis, robotics, neurophysiology, computer vision, and numerous other areas. When the polynomial systems are inconsistent there is no solution. A polynomial system is zero-dimensional if it has finite number of isolated solutions. A system with infinite number of solutions is said to be positive-dimensional. Positive-dimensional systems can have finite number of isolated solutions. The upper bound on the number of isolated solutions of polynomial systems is given by Bezout's Theorem and Bernstein Theorem.

The problem of computing the isolated solutions of the polynomial equations system can be approached in various ways such as a homotopy continuation, symbolic methods, and local iterative methods.

The most popular approach is the homotopy continuation. The homotopy continuation is an approach to solving a system of equations by tracking the solutions of the "nearby" systems of equations which are easy to solve. The advantages of the homotopy continuation are that it finds all the solutions of the system and it does not need the initial solution unlike some other methods. The main disadvantage of the homotopy continuation is that it is relatively slow because it finds all the solutions including those which do not have physical meaning. Moreover, all the computations are done in complex arithmetic.

Other popular techniques for solving polynomial systems are symbolic methods which have roots in algebraic geometry. These methods, same as the methods based on the resultants and Gröbner based algorithm, can be used for eliminating variables and thereby reduce the problem of finding the roots of the univariate polynomials. The major problem in the implementation of the symbolic methods arises from the fact that computing the roots of a univariate polynomial can be ill-conditioned for high-degree polynomials. Also, symbolic algorithms are better suited for the problem where coefficients of polynomial systems are rational numbers. However, by the use of symbolic methods on low-degree polynomial systems with a few equations (the most often two or three) it is possible to obtain solutions in the closed form. Moreover, special class of these methods can find the solution of positive-dimensional polynomial systems.

The local iterative methods for solving nonlinear systems of equations, such as the Newton method or the conjugate gradient method need initial solution estimates. If the estimate is good, the methods are reliable and therefore they find one of the solutions in few iterations. On the other hand, if the initial estimate is not good, the methods converge poorly or do not converge at all. These methods are suitable to find one solution of the system. In order to find all the solutions of the system, it is possible to start with a set of different initial solutions and gather all the different solutions, but generally there is no guarantee that all the solutions will be found.

Computing the step length in an arbitrary direction is a compromise. The step length has to provide a substantial reduction of a certain goal function along an arbitrary direction, but at the same time that choice has to be done with a minimal computational effort. Finding a local minimizer of the goal function along the arbitrary direction requires too many evaluations so the practical line-search algorithms perform an inexact line search to achieve an adequate decreasing of the goal function at a minimal cost. But, for the polynomial systems, the goal function along an arbitrary direction is an univariate polynomial so it is possible to compute the global minimizer of a goal function effortlessly.

This thesis studies the local iterative methods for solving polynomial systems which employ a strategy as follows: determining suitable directions, finding the global minimizer of chosen goal function along direction (“deepest” point) and moving to it. The natural title for this strategy are the deepest descent methods. We examined several suitable directions and two goal functions as well as compared the results with those from the two well-known systems of nonlinear equations solvers: Minpack and NLEQ. The evaluation is based on the collection of more than 100 polynomial systems collected from relevant sources.

Preliminaries

The first chapter of the thesis consists mainly of the definitions and the overviews of the main results. The chapter starts with the definition of multivariate polynomial and polynomial systems. A polynomial system is a system of nonlinear equations where each equation is a multivariate polynomial. Formally, let $f_1(\mathbf{x}), \dots, f_n(\mathbf{x}) : \mathbb{R}^m \rightarrow \mathbb{R}$ are multivariate polynomials. The polynomial system is the equation $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, where $\mathbf{F} : \mathbb{R}^m \rightarrow \mathbb{R}^n$, $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$. The thesis includes only the case when the number of equations is equal to the number of indeterminates ($m = n$).

The main results about the nature and the number of solutions are presented, along with the general ideas of the most frequently used methods for solving polynomial systems. Since the deepest descent methods are a subset of the local iterative methods for solving nonlinear systems of equations, the local iterative methods are outlined in the second part of the chapter. According to the deepest descent methods, computation of zeroes of a univariate polynomial is very important so the chapter finishes with the overview of the methods for root finding of the

univariate polynomials.

Deepest descent method

The second chapter presents the fundamentals of the deepest descent method applied to the polynomial system solving.

The deepest descent methods are a special class of local iterative methods. The framework of the deepest descent method can be described as:

1. Choose initial iteration point \mathbf{x}_0 , set $k = 0$.
2. Determine search direction \mathbf{s}_k .
3. For given \mathbf{x}_k , \mathbf{s}_k construct goal function $g_{\mathbf{x}_k, \mathbf{s}_k}$. Function $g_{\mathbf{x}_k, \mathbf{s}_k} : \mathbb{R} \rightarrow \mathbb{R}$, defined by $g_{\mathbf{x}_k, \mathbf{s}_k}(t) = g(\mathbf{x}_k + t \cdot \mathbf{s}_k)$, is a restriction of goal function g on the line determined by point \mathbf{x}_k and direction \mathbf{s}_k .
4. Compute step length t_k as a global minimizer of $g_{\mathbf{x}_k, \mathbf{s}_k}$. The step length t^* formally is written as

$$t^* = \arg \min_{t \in \mathbb{R}} g_{\mathbf{x}_k, \mathbf{s}_k}(t) = \arg \min_{t \in \mathbb{R}} g(\mathbf{x}_k + t \cdot \mathbf{s}_k).$$

5. Compute the new iteration point $\mathbf{x}_{k+1} = \mathbf{x}_k + t^* \cdot \mathbf{s}_k$.
6. If the stopping criteria are not satisfied, increment k and go to the step 2.

As it can be seen from the framework, the most distinguished property of the deepest descent method is the use of global minimizer of the goal function along an arbitrary direction. If the best point on the line of search direction is “behind” some obstacle, this property allows the deepest descent methods, to see it anyway and move directly to the minimizer, regardless of how distant it is from the current position.

There are many deepest descent methods. They differ with respect to the choice of the search direction and goal function g . The most common choice of g is the square of the Euclidean norm, i.e. in arbitrary point \mathbf{x} , $g(\mathbf{x})$ is a square of Euclidean norm of $\mathbf{F}(\mathbf{x})$. This choice is referred to as *RSS* in the rest of the thesis. The max-norm and 1-norm of $\mathbf{F}(\mathbf{x})$ are also used. The choice of max-norm as a goal function is referred to as *MRN*, while the choice of 1-norm is not studied in the thesis. For the general nonlinear system of equation, and the given point \mathbf{x} and direction \mathbf{s} , it is difficult to find the global minimizer of function $g_{\mathbf{x}, \mathbf{s}}$. Fortunately, in the case of polynomial systems, function $g_{\mathbf{x}, \mathbf{s}}$ is a univariate polynomial and therefore the calculation of the global minimizer is straightforward.

After the presentation of the general framework for the deepest descent method, the second chapter is divided into five parts: solving the quasilinear polynomial systems, construction of *RSS* function in dependence of \mathbf{x} and \mathbf{s} , construction of *MRN* function in dependence of \mathbf{x} and \mathbf{s} , brief discussion about stopping criteria and brief review of the application of the deepest decent methods to the solving of linear systems.

In the first part of the second chapter the deepest descent method applied to the solving of the quasilinear polynomial systems is analysed. The method uses the coordinate axes as a search direction and the RSS as a goal function. It is shown that $RSS_{\mathbf{x},\mathbf{s}}$ is a quadratic function for an arbitrary \mathbf{x} and \mathbf{s} and the closed form for step length calculation is given. In the next part of the chapter, the application of the deepest descent methods to the general polynomial systems is studied. The methods that use the RSS as the goal function and the arbitrary direction are observed. The main emphasis is placed on the problem of the construction of polynomial $RSS_{\mathbf{x},\mathbf{s}}$ for the given \mathbf{x} and \mathbf{s} . The key results are the conditions which \mathbf{F} , \mathbf{x} , and \mathbf{s} must satisfy to guarantee that $RSS_{\mathbf{x},\mathbf{s}}$ is of maximal degree. That result is important because, due the numerical reason, it is preferable to know in advance what the degree of the polynomial is. The other important result is that global minimizer of the $RSS_{\mathbf{x},\mathbf{s}}$ exists and it is one of the local minimizers of $RSS_{\mathbf{x},\mathbf{s}}$, i.e., one of the real solutions of equation $RSS'_{\mathbf{x},\mathbf{s}}(t) = 0$. Furthermore, the two additional approaches to the construction of $RSS_{\mathbf{x},\mathbf{s}}$ are suggested:

- the numerical approach which employs the numerical interpolation to obtain a polynomial and the Taylor series expansion to convert the polynomial in the standard base,
- the symbolic calculation approach where a polynomial is constructed by performing mathematical operations on the symbolic level.

Even though slower, the symbolic approach is usually more precise. Therefore, the program PolySolver, written for the numerical experiment with the polynomial system solving, uses numerical approach to construct $RSS_{\mathbf{x},\mathbf{s}}$ when degree of $RSS_{\mathbf{x},\mathbf{s}}$ is known in advance. Otherwise, the symbolic approach is used.

The third part of the chapter deals with the use of the max-norm (function MRN) as a goal function in the deepest decent methods. For the given \mathbf{x} and \mathbf{s} , the construction of $MRN_{\mathbf{x},\mathbf{s}}$ as well as its use are more complex compared to function $RSS_{\mathbf{x},\mathbf{s}}$. The function $MRN_{\mathbf{x},\mathbf{s}}$ is piecewise polynomial, thus the major problem is to find the intervals I_s , $s = 0, \dots, S$, defined by $I_0 = \langle -\infty, t_1 \rangle$, $I_s = \langle t_i, t_{i+1} \rangle$, $1 = 1, \dots, S-1$ and $I_S = \langle t_n, \infty \rangle$, which form partition of \mathbb{R} and for each of the interval I_s there is $f_j \in \{f_1, \dots, f_n\}$ such that for all $t \in I_s$ is $|f_j(\mathbf{x} + t \cdot \mathbf{s})| \geq |f_k(\mathbf{x} + t \cdot \mathbf{s})|$, $k \in \{1, \dots, n\}$, $i \neq j$. The algorithm for the construction of $MRN_{\mathbf{x},\mathbf{s}}$, and evaluation of the global minimizer of $MRN_{\mathbf{x},\mathbf{s}}$ is presented. The main advantage of function $MRN_{\mathbf{x},\mathbf{s}}$ compared to $RSS_{\mathbf{x},\mathbf{s}}$ is that $MRN_{\mathbf{x},\mathbf{s}}$ deals with a polynomial of a smaller degree.

The stopping criteria are discussed in the fourth part of the chapter. The iteration process is terminated successfully if the last iteration point \mathbf{x}_N yields small max-norm of $\mathbf{F}(\mathbf{x}_N)$. On the other hand, the iteration process does not converge if it is not terminated successfully after arbitrary number of iterations. Also, the iteration process does not converge if relative change of two successive iteration points is smaller than the defined tolerance or if the large difference between successive iteration points yields a small decrease of the specified goal function.

The chapter finishes with the application of the deepest descent method to the linear systems

of equations. This is possible because the linear systems are a special subset of the polynomial system. The applied method uses coordinate axes as a search direction and RSS as a goal function. The coordinate axes are used circularly: the first search direction is \mathbf{e}_1 , the next one is \mathbf{e}_2 , etc. until the direction \mathbf{e}_n is used. Then the choice of search direction starts again with \mathbf{e}_1 . This variant of the deepest decent methods is named DD-LS. It is proven that for linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, DD-LS method and the well-known Gauss-Seidel method are connected. Namely, one step of the DD-LS method applied to the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ generates same new iteration point as the one step of Gauss-Seidel method applied to linear system $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$. This result shows that DD-LS method is convergent if matrix A is regular.

Different deepest descent methods

In the third chapter various deepest descending methods are introduced. They differ as regards the choice of search direction and goal function. The name of the single method is a combination of the abbreviation for the search direction and the suffix “-E” or “-M”, depending on the choice of a goal function. All methods that use square of the Euclidean norm (RSS function) as a goal function have suffix “-E”. Similarly, all methods that use max-norm (MRN function) as a goal function have suffix “-M”. The suffix “-*” denotes both methods for the given search direction.

The most popular local iterative methods are the Newton method and the steepest descent method, so it is natural to explore the deepest decent methods with the same search directions. According to the Newton method, for the given point \mathbf{x} the search direction \mathbf{s}_N (Newton direction) is determined as a solution of the equation $\nabla\mathbf{F}(\mathbf{x})\mathbf{s}_N = -\mathbf{F}(\mathbf{x})$. The deepest descent methods that employ the Newton direction are named NWT-E and NWT-M. The other popular local iterative method is the steepest descent method where, for given point \mathbf{x} , search direction \mathbf{s}_G is computed as a gradient of the square of the Euclidean norm of the $\mathbf{F}(\mathbf{x})$, $\mathbf{s}_G = -(\nabla\mathbf{F}(\mathbf{x}))^T\mathbf{F}(\mathbf{x})$. This choice of the search direction generates two methods: GN-E and GN-M. Because \mathbf{s}_G is better suited for the Euclidean norm additional checking is built in the GN-M method: if there is no change in the direction \mathbf{s}_G , replace \mathbf{s}_G with \mathbf{s}_N . The following methods, BGN-E and BGN-M, in one iteration step try both directions \mathbf{s}_N and \mathbf{s}_G and take the direction which gains a better improvement of the goal function. When the computational effort to evaluate one iteration step is considered, the GN-* methods are faster than NWT-* methods, but they converge slowly. BGN-* methods are slower than GN-* and NWT-* methods, but they are more efficient.

The methods GS-E and GS-M share some similarities with the Gauss-Seidel method for solving systems of linear equations. For the given \mathbf{x} the direction \mathbf{s}_{GS} is defined by $\mathbf{s}_{GS} = \mathbf{x}^{(n)} - \mathbf{x}^{(0)}$. The points $\mathbf{x}^{(0)} = \mathbf{x}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$ are obtained by the iteration process, $\mathbf{x}^{(j)} = \mathbf{x}^{(j-1)} + t_j^* \mathbf{e}_{p(j)}$, $j = 1, \dots, n$. In j -th step, the function f_M , with the maximal value $|f_i(\mathbf{x}^{(j-1)})|$ is found. Then the absolute values of the partial derivatives of $f_i(\mathbf{x}^{(j-1)})$ with respect to x_k , $k = 1, \dots, n$, are

evaluated, but only for those components which are not already used. The $p(j)$ is the index of the component where the greatest value of partial derivatives is achieved. The GS-* methods are slower compared to NWT-* or GN-* methods, but they are comparable to BGN-* methods.

The direction \mathbf{s}_{LP} is evaluated from the solution of arbitrary linear programming problem. Let Δ be the ball of radius δ around origin in max-norm. For the given \mathbf{x} , the direction \mathbf{s}_{LP} is determined as such $\mathbf{h} \in \Delta$ that minimize max-norm of $\mathbf{F}(\mathbf{x}) + \nabla\mathbf{F}(\mathbf{x})\mathbf{h}$. This problem of minimization can be regarded as a linear programming problem and solved by the use of appropriate techniques such as the simplex method. The method generated by \mathbf{s}_{LP} is named LP-M. Note that \mathbf{s}_{LP} is determined to reduce max-norm and is not appropriate for the use of *RSS* as a goal function. The LP-M method is slow because the evaluation of direction \mathbf{s}_{LP} is computationally demanding.

The least efficient methods are KO-E and KO-M, where search direction is one of the coordinate axes. These methods are fast because it is easy to determine the direction and the construct functions *RSS* and *MRN*, but iteration process usually does not converge.

The chapter ends with the application of the deepest descent methods to the polynomial systems with complex coefficients. This is still an open question, but the initial results are promising.

Numerical results

The fourth chapter presents the results of the numerical experiment. In the experiment, the deepest descent methods described in the third chapter are compared with two benchmark solvers: the `MinPack` and the `NLEQ`. The `MinPack` is a library of FORTRAN subroutines for solving of the nonlinear equations systems, or the least squares minimization of the residual of a set of linear or nonlinear equations. The `MinPack` is a free and highly portable, robust, and reliable software. The function `fsolve` from the program package `Octave` is based on the subroutines from `MinPack`. `NLEQ` is the solver for solving the systems of nonlinear equations using the exact global Newton method. It is a part of the `CodeLib`, a collection of algorithms for the applications in the area of Scientific Computing. The `NLEQ` is also written in FORTRAN.

The comparison between the deepest descent methods and benchmark solvers is made on the collection of more than a 100 polynomial systems of various dimensions which were collected from the relevant sources. For each method and problem, the success rate is calculated, whereby success rate is defined as a ratio between the number of the iterative processes which finished successfully, and the number of the iterative processes attempted. Iterative processes differ by the initial iterative points. A few thousands initial points for each dimension are generated randomly in advance. Since many of the observed systems have solutions around the origin, the iterative points are generated from three sets D_1 , D_2 , and D_3 , defined as $D_1 = K(2)$, $D_2 = K(5) \setminus K(2)$, and $D_3 = K(10) \setminus K(5)$, where $K(r)$ is the ball of the radius r around the origin in

max-norm. In other words, D_2 is a ring in the max-norm around the origin with the inner radius 2 and outer radius 5 while D_3 is a ring around the origin with the inner radius 5 and outer radius 10.

The methods are compared using three indices. Any index can be calculated on the whole set of problems or some specific subset of that set. The first index, the index of maximal successfulness I_M , for arbitrary method is defined as the number of problems for which the method was most successful. This index is easy to compute but it can reveal only most dominant methods. The second index, index of mutual successfulness I_B^E , for arbitrary method is defined as the number of methods which have a lower success rate than the observed method. Precisely, how many other methods have lower success rate than observed method is counted for each problem, and the numbers for all the problems are summarized. There are several ways to find which method has a lower success rate: direct comparison, comparison with threshold (method A is better than method B if the success rate of A is greater than success rate of B plus threshold) or use of statistical test to find whether the method A is statistically better than the method B (McNemar's test). In the thesis, all these index variations gave similar results. Finally, the third index used is the index of rank I_R . For each problem, the methods are sorted in the ascending order by the success ratio and according to the chosen criteria (direct comparison, comparison with threshold or statistical comparison). The method with the lowest success rate gets value 0, the next one gets 1, etc. If there are several methods with the same rank then each of them gets the average of numbers associated with their position (for example, if there are three methods with the lowest score then each of them gets $(0+1+2)/3 = 1$). The value of I_R for arbitrary method is the sum of the values a particular method obtained from each problem in the collection.

The results of the experiment show that BGN-E and GS-M are the methods with the highest score, followed by BGN-M and GS-E methods. On the other hand, the GN-E, KO-E and KO-M are methods with the lowest score. The results reveal that the choice of the search direction is a more important factor for the successfulness of the method than the choice of a goal function. The best score was achieved by BGN-* methods, followed by GS-* methods, NWT-* methods, GN-* methods and KO-* methods. Generally, it seems that the methods which use *RSS* as the goal function are more successful than those methods which use *MRN* function. The deepest descent methods are fairly good as compared to MinPack and NLEQ methods and only the GN-E and KO-* methods had lower score than benchmark solvers.

Conclusion

The deepest descent methods are a class of iterative methods for solving the systems of nonlinear equations. They differ from the local iterative methods in so far as they do not use a line search in the descent direction to determine a new iteration point. Instead, they seek an exact global minimizer along an arbitrary direction to determine a new iteration point. That

approach is impractical for general nonlinear systems, but for the polynomial systems and appropriate goal functions (square of Euclidean norm or max-norm), determination of the exact global minimizer comes down to finding roots of univariate polynomial.

In the thesis, the construction of the restriction of the goal function along arbitrary direction is described. In the case of the polynomial system, the restriction is a univariate polynomial. The conditions for evaluating the degree of that polynomial are imposed and it is proven that a polynomial always has a global minimizer.

The various deepest descent methods are studied. The search direction can be any standard direction such as Newton direction, the steepest descent direction or the coordinate axes, or a combination of standard search directions. Furthermore, two new search directions are analysed: The Gauss-Seidel direction, where the search direction is determined by a solving series of polynomial equations, and the LP direction, which is computed as the solution of linear programming problem. Additionally, the use of different goal functions is evaluated.

In order to compare different methods, the collection of more than a 100 benchmark polynomial systems from various sources is prepared. Every problem from the collection is solved by each method several thousand times with a different initial iteration point. The success rate for the pair (method, problem) is measured as a ratio between the number of the iteration processes where the method finds any solution of the system divided by the number of iteration processes attempted. The best results are accomplished with the BGN-* methods that combine two search directions, followed by GS-* methods. The deepest descent methods are compared with two benchmark solvers and results suggest that nearly all the deepest descent methods have better success rate than the benchmark methods. The exceptions are the methods that use coordinate axes and steepest descent direction as a search direction.

The deepest descent methods have few limitations. They are relatively slow, but that problem can be solved by parallelization. Furthermore, one iteration process of the deepest descent methods can find only one solution, but by the change of the initial iteration point, it is possible to find all other solutions of the polynomial system.

Compared to the local iterative methods, the deepest descent methods have a better success rate. Applied on the appropriate polynomial systems the methods will show excellent results.

Sadržaj

1. Uvod	1
1.1. Definicija sustava i teorijski rezultati	1
1.1.1. Polinomni sustavi	1
1.1.2. Teorijski rezultati	2
1.2. Algoritmi za rješavanje polinomnih sustava	4
1.2.1. Analitički algoritmi	5
1.2.2. Metode homotopnog kontinuiteta	6
1.2.3. Algoritmi zasnovani na podjeli	8
1.2.4. Algebarski i geometrijski algoritmi	9
1.3. Metode spusta	9
1.4. Metode za traženje nultočaka polinoma	11
2. Rješavanje polinomnih sustava metodom najdublje spusta	13
2.1. Rješavanje kvazilinearnih sustava	14
2.2. Rješavanje općih polinomnih sustava – slučaj kada je vektor spusta koordinatna os	16
2.3. Rješavanje općih polinomnih sustava – slučaj proizvoljnog vektora spusta	21
2.4. Određivanje koeficijenata polinoma $RSS_{x,s}(\lambda)$	27
2.5. Kriteriji zaustavljanja	31
2.6. Minimizacija u max–normi	34
2.6.1. Funkcije zadane po dijelovima	35
2.6.2. Max–funkcija za dva polinoma	37
2.6.3. Max–funkcija za više polinoma	42
2.6.4. Rješavanje polinomnih sustava primjenom funkcije MRN	45
2.7. Rješavanje linearnih sustava metodom najdublje spusta	50
3. Odabir smjera spusta	55
3.1. Standardni odabir smjera spusta	56
3.1.1. Newtonov smjer spusta	57

3.1.2.	Najstrmiji smjer spusta	59
3.1.3.	Usporedba NWT* i GN* metoda	60
3.2.	Kombiniranje standardnih smjerova spusta	63
3.3.	Gauss–Seidelov smjer	66
3.4.	Smjer spusta određen max–normom	72
3.5.	Smjerovi koordinatnih osi	78
3.6.	Metoda najdublje spusta u \mathbb{C}^n	79
4.	Rezultati	82
4.1.	Područja konvergencije različitih metoda	82
4.1.1.	Problem mickey	82
4.1.2.	Problem leary	87
4.1.3.	Problem himmelbaum	90
4.2.	Pregled rezultata eksperimenta	92
4.3.	Određivanje indeksa uspješnosti	107
4.3.1.	Indeks maksimalne uspješnosti	107
4.3.2.	Indeks međusobne uspješnosti	109
4.3.3.	Indeks poretka	112
5.	Zaključak	114
	Literatura	117
	Prilog A. Popis polinomnih sustava	121
	Prilog B. Opis programa PolySolver	167
	Životopis	173
	Biography	175

Poglavlje 1

Uvod

Polinomni sustavi se pojavljuju i igraju važnu ulogu u brojnim područjima znanosti. Potreba za rješavanjem polinomnih sustava može se pronaći u robotici (kinematika, detekcija sudara), kemiji [1], mehanici [2], [3], neurofiziologiji [4] i računalnom vidu. Sam problem rješavanja polinomnog sustava se smatra numerički teškim radi složenosti računanja, a dodatni problem je što unaprijed nije poznato koliko rješenja ima određeni polinomni sustav, a algoritam koji rješava problem trebao bi naći sva rješenja [5].

1.1 Definicija sustava i teorijski rezultati

1.1.1 Polinomni sustavi

Realni polinom stupnja d se definira kao funkcija $p : \mathbb{R} \rightarrow \mathbb{R}$, oblika

$$p(x) = a_0x^d + a_1x^{d-1} + \dots + a_{d-1}x + a_d, \quad (1.1)$$

pri čemu su $a_0, a_1, \dots, a_d \in \mathbb{R}$, $a_0 \neq 0$, a d je nenegativan cijeli broj. Brojevi a_0, a_1, \dots, a_d se nazivaju koeficijenti, dok se cjelobrojne potencije od x ($1, x, x^2, \dots, x^d$) nazivaju monomi.

Analogno će se definirati i polinomi više varijabli, samo je prvo potrebno definirati izraz analogan monomu.

Definicija 1.1 *Neka su $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ i $\mathbf{d} = (d_1, \dots, d_n) \in \mathbf{N}_0^n$ ($d_i \geq 0$, $i = 1, \dots, n$). Za \mathbf{x} i \mathbf{d} definira se generalizirana potencija $\mathbf{x}^{\mathbf{d}}$ s*

$$\mathbf{x}^{\mathbf{d}} := x_1^{d_1} x_2^{d_2} \dots x_n^{d_n}.$$

Izraz $\mathbf{x}^{\mathbf{d}} = x_1^{d_1} x_2^{d_2} \dots x_n^{d_n}$ naziva se član (engl. term). Stupanj člana jednak je zbroju svih eksponentata koji ga čine,

$$\deg \mathbf{x}^{\mathbf{d}} = d_1 + d_2 + \dots + d_n.$$

Sada se može definirati polinom u više varijabli.

Definicija 1.2 Funkcija $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ u n varijabli $\mathbf{x} = (x_1, \dots, x_n)$ naziva se polinom ako se može prikazati kao linearna kombinacija članova

$$\begin{aligned} f(\mathbf{x}) &= \sum_{j=1}^k a_j x_1^{d_{1j}} x_2^{d_{2j}} \dots x_n^{d_{nj}} \\ &= \sum_{j=1}^k a_j \mathbf{x}^{\mathbf{d}_j}, \end{aligned} \quad (1.2)$$

gdje su vektori $\mathbf{d}_j \in \mathbb{N}_n^0$ definirani s $\mathbf{d}_j = (d_{1j}, \dots, d_{nj})^T$. Stupanj polinoma f jednak je najvećem stupnju člana,

$$\deg f(\mathbf{x}) = \max_j \deg \mathbf{x}^{\mathbf{d}_j}.$$

U slučaju kad treba raditi s kompleksnim brojevima, jedina izmjena u odnosu na definiciju 1.2 je promjena domene i kodomene polinoma f , koji se tada definira s $f(\mathbf{x}) : \mathbb{C}^n \rightarrow \mathbb{C}$. U teorijskim razmatranjima se uglavnom promatraju polinomi čije su varijable kompleksni brojevi, a i neke metode funkcioniraju samo ako rade nad polinomima s kompleksnim varijablama.

Definicija 1.3 Polinomni sustav je jednadžba oblika $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, pri čemu je funkcija $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ oblika $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$, gdje su f_1, \dots, f_m polinomi više varijabli, $\mathbf{x} = (x_1, \dots, x_n)$, a $\mathbf{0}$ je nul-vektor. Potpuno analogno se definira polinomni sustav ukoliko se promatraju polinomi nad kompleksnim brojevima, samo se u definiciji zamijeni \mathbb{R} s \mathbb{C} .

Polinomni sustav može imati jednaki broj jednadžbi i nepoznanica ($m = n$) i takvi sustavi su glavna tema ovog rada. Polinomni sustav može imati i različit broj jednadžbi od broja nepoznanica. Ako sustav ima više jednadžbi od nepoznanica ($m > n$) govori se o predefiniranom (engl. *overdetermined*) sustavu. S druge strane, ako je broj jednadžbi manji od broja nepoznanica, onda se govori o poddefiniranom (engl. *underdetermined*) sustavu.

U ovom radu težište je na polinomnim sustavima koji imaju jednaki broj jednadžbi i nepoznanica. Neki od algoritama koji će biti predstavljeni u radu mogu raditi i sa predefiniranim sustavima, a poddefinirani sustavi izlaze izvan okvira ovog rada jer nemaju konačan broj rješenja.

1.1.2 Teorijski rezultati

Poznato je da sustavi linearnih jednadžbi mogu imati jedno, niti jedno ili beskonačno mnogo rješenja, no kod sustava nelinearnih jednadžbi situacija je mnogo teža [6]. Srećom, kod polinomnih sustava postoji teorija koja daje gornju ogradu za broj njihovih rješenja, no da bi se centralni rezultat mogao iskazati potrebno je uvesti nekoliko definicija [3]. Nadalje, u većini

polinomnih sustava koji nastaju u praksi, koeficijenti u jednadžbama su realni brojevi i rješenja koja se traže su realni brojevi, a sustav može imati i rješenja koja su kompleksni brojevi. Stoga je za teorijska razmatranja pogodnije promatrati polinome nad skupom kompleksnih brojeva, pa će se u ostatku poglavlja podrazumijevati da je polinomni sustav zadan kao $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, $\mathbf{F} : \mathbb{C}^n \rightarrow \mathbb{C}^n$.

Definicija 1.4 *Neka je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ zadani polinomni sustav s n jednadžbi i n nepoznanica, $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$, $\mathbf{x} = (x_1, \dots, x_n)$. Kaže se da \mathbf{F} ima u točki \mathbf{x}^* izolirano rješenje ako postoji $\varepsilon > 0$, takav da za svaki $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} \neq \mathbf{x}^*$ vrijedi*

$$\|\mathbf{x} - \mathbf{x}^*\|_2 < \varepsilon \Rightarrow \mathbf{F}(\mathbf{x}) \neq \mathbf{0}. \quad (1.3)$$

Izolirana rješenja su najčešće ona rješenja koja su u praksi zanimljiva. Osim izoliranih rješenja, polinomni sustavi mogu imati i rješenja u beskonačnosti.

Neka je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ zadani polinomni sustav, $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$. Neka je s d_i označen stupanj polinoma f_i , $i = 1, \dots, n$. Nadalje, za svaki polinom f_i definira se njegov homogeni dio f_i^0 , koji se dobije zanemarivanjem svih članova od f_i čiji je stupanj manji od d_i . Od tako dobivenih polinoma formira se funkcija $\mathbf{F}^0 = (f_1^0, \dots, f_n^0)$, koja se naziva homogeni dio polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

Definicija 1.5 *Rješenje u beskonačnosti polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ je svako rješenje \mathbf{x}^* sustava $\mathbf{F}^0(\mathbf{x}) = \mathbf{0}$, koje je različito od nule, a prvi član različit od nule je jednak 1, $\mathbf{x}^* = (0, \dots, 0, 1, x_{i+1}^*, \dots, x_n^*)$ [7].*

Kod traženja rješenja polinomnog sustava, numerički problem predstavljaju singularna rješenja:

Definicija 1.6 *Neka je \mathbf{x}^* rješenje sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. Rješenje \mathbf{x}^* je singularno rješenje sustava ako je $\det \mathbf{J}(\mathbf{x}^*) = 0$, pri čemu \mathbf{J} označava Jacobijevu matricu funkcije \mathbf{F} . Jacobijeva matrica funkcije \mathbf{F} je $n \times n$ matrica parcijalnih derivacija, čiji su elementi definirani s*

$$J_{r,s}(\mathbf{x}) = \frac{\partial f_r}{\partial x_s}(\mathbf{x}). \quad (1.4)$$

Sada je moguće iznijeti glavni teorijski rezultat koji daje gornju ogradu za broj rješenja polinomnog sustava [3].

Teorem 1.7 (Bezoutov teorem) *Neka je zadan polinomni sustav $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, gdje je $\mathbf{F} : \mathbb{C}^n \rightarrow \mathbb{C}^n$, $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$ i $f_i : \mathbb{C}^n \rightarrow \mathbb{C}$, $i = 1, \dots, n$, su polinomi. Neka d_i , $i = 1, \dots, n$, označava stupanj polinoma f_i . Definiramo ukupni stupanj od \mathbf{F} s $d = d_1 \cdots d_n$. Tada vrijedi:*

1. Ukupan broj izoliranih rješenja i rješenja u beskonačnosti polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ nije veći od d .
2. Ako sustav $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ nema niti beskonačan broj rješenja niti beskonačan broj rješenja u beskonačnosti, tada sustav ima točno d rješenja i rješenja u beskonačnosti, računajući njihovu višestrukost.

Dokaz Bezoutova teorema može se naći u [7].

Ako ne postoje singularna rješenja ili singularna rješenja u beskonačnosti, tada sva rješenja imaju višestrukost jedan, tako da je ukupan broj različitih rješenja jednak d . Bezoutov teorem ne govori ništa o broju realnih rješenja polinomnog sustava.

Precizniju granicu za broj rješenja polinomnog sustava daje Bernsteinov teorem, no da bi ga mogli iskazati potrebno je uvesti pojam mješovitog volumena (engl. *mixed-volume*).

1.2 Algoritmi za rješavanje polinomnih sustava

Obzirom da problem rješavanja polinomnih sustava vuče svoje korijene još od antičke Grčke i Kine [8], nije čudno da je stvoren veliki broj algoritama za njihovo rješavanje.

Elkadi i Mourrain [8] su napravili sljedeću podjelu:

- Analitički algoritmi: ovoj kategoriji pripadaju algoritmi koji koriste vrijednosti funkcije i njenih derivacija u cilju formiranja iterativnog postupka koji konvergira rješenju ili rješenjima polinomnog sustava. Tipični algoritmi bazirani su na Newtonovoj metodi ili na nekoj od optimizacijskih metoda.
- Algoritmi zasnovani na metodi homotopnog kontinuiteta (engl. *homotopy continuation*): u ovoj kategoriji se nalaze algoritmi u kojima se prvo pronalazi rješenje lako rješivog sustava $\mathbf{G}(\mathbf{x}) = 0$, "sličnog" polaznom sustavu $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. Nakon toga se rješavaju sustavi sve "sličniji" polaznom sve dok se ne dođe do rješenja originalnog problema.
- Algoritmi zasnovani na podjeli (engl. *subdivision solvers*): koriste razne kriterije kojima se dijelovi domene isključuju iz daljnjeg pretraživanja ako ne sadrže rješenje polinomnog sustava. Često se koriste za traženje izoliranih realnih rješenja. U ovim algoritmima se intenzivno koristi intervalna aritmetika, pa se stoga ponekad nazivaju i intervalni algoritmi [5], dok se u [9] nazivaju metode isključivanja (engl. *exclusion methods*).
- Algebarski algoritmi: ovi algoritmi koriste poznate relacije između nepoznanica čime polazni problem reduciraju na problem traženja nultočaka polinoma jedne varijable ili na problem traženja svojstvenih vrijednosti. Pri tome se koriste alati poznati u algebarskoj geometriji poput Gröbnerove baze i normalnog oblika polinoma.
- Geometrijski algoritmi: kod ovih algoritama se polazni problem projicira na manji potprostor, a potom se koriste geometrijska svojstva skupa rješenja. Kao i kod algebarskih

algoritama, polazni problem se reducira ili na problem traženja nultočka polinoma jedne varijable ili na problem traženja svojstvenih vrijednosti.

Postoje i druge podjele algoritama [5, 9, 10], ali navedena najbolje opisuje trenutno stanje.

U nastavku poglavlja detaljnije su opisane prve tri grupe algoritama. Geometrijski i algebarski algoritmi su više bazirani na metodama algebarske geometrije i simboličkog računa, što izlazi iz okvira ovog rada, stoga je za njih dan samo kratki opis.

1.2.1 Analitički algoritmi

Vjerojatno najpoznatiji algoritam koji pripada ovoj klasi je Newtonova metoda* [11]. Primjenjuje se za rješavanje općih nelinearnih sustava n jednažbi s n nepoznanica, oblika $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, gdje je \mathbf{F} neprekidno diferencijabilna funkcija $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ i $\mathbf{x} \in \mathbb{R}^n$. Ideja metode je sljedeća: u okolini točke \mathbf{x} , \mathbf{F} se može linearno aproksimirati razvojem u Taylorov red, $\mathbf{F}(\mathbf{x} + \mathbf{s}) \approx \mathbf{F}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\mathbf{s}$, pri čemu je $\mathbf{J}(\mathbf{x})$ Jacobijeva matrica od \mathbf{F} u \mathbf{x} , čiji su elementi definirani sa (1.4), a potom se traži nultočka te aproksimacije. Na taj način Newtonova metoda generira niz $\mathbf{x}^0, \mathbf{x}^1, \dots$, u kojem se svaka sljedeća točka \mathbf{x}^{k+1} dobije kao nultočka linearne aproksimacije od \mathbf{F} u okolini trenutne točke \mathbf{x}^k , $k = 0, 1, \dots$. Početna točka iteracija, \mathbf{x}^0 , trebala bi biti odabrana tako da se nalazi razumno blizu rješenja. Prethodno opisani algoritam može se zapisati kao

Algoritam 1.1 Newtonova metoda za rješavanje nelinearnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

- 1: Neka je zadana početna točka $\mathbf{x}^{(0)}$.
 - 2: $k=0$
 - 3: **while** nije zadovoljen kriterij zaustavljanja
 - 4: Izračunaj Jacobijevu matricu $\mathbf{J}(\mathbf{x}^{(k)})$
 - 5: riješi sustav $\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)})$,
 - 6: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$.
 - 7: $k = k + 1$
 - 8: **end while**
-

U općem slučaju, izračun matrice $\mathbf{J}(\mathbf{x}^k)$ može biti problematičan ukoliko nije moguće analitički izračunati parcijalne derivacije od \mathbf{F} , jer se tada parcijalne derivacije trebaju numerički izračunati, a taj postupak može usporiti algoritam. Zato su razvijene razne metode u kojima se $\mathbf{J}(\mathbf{x}^k)$ aproksimira (npr. Broydenova metoda [11]).

Pri rješavanju polinomnog sustava uvijek je moguće analitički izračunati parcijalne derivacije, no javljaju se drugi problemi [3]. Naime, zbog svog lokalnog karaktera, Newtonova metoda može konvergirati rješenju koje nema praktičnog smisla ili ne konvergirati uopće. Također se javljaju problemi s konvergencijom u slučaju da sustav ima i singularna rješenja (definicija

*Ovdje opisana metoda se u dijelu literature naziva i Newton-Raphsonova metoda [6].

1.6). Konačno, polinomni sustavi imaju često više rješenja, stoga je potrebno razraditi posebne strategije pomoću kojih se mogu naći sva rješenja koristeći Newtonovu metodu.

Ako se početna točka \mathbf{x}^0 Newtonove metode nalazi blizu rješenja sustava koje nije singularno, tada metoda konvergira kvadratično [11, 12]. U slučaju da se početna točka nalazi blizu singularnog rješenja sustava, konvergencija je sporija.

Druga vrsta metoda koja se nalaze u ovoj klasi, svoj izvor nalaze u optimizacijskim metodama. Neka je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ zadani polinomni sustav kao u definiciji 1.3, $\mathbf{F} = (f_1, \dots, f_n)$. Za taj sustav se definira funkcija cilja $G : \mathbb{R}^n \rightarrow \mathbb{R}$, s $G(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x})^2$. Problem rješavanja $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ je ekvivalentan problemu minimizacije funkcije G .

Funkcija G može se minimizirati koristeći neke od standardnih tehnika minimizacije, ali zbog prirode problema, koriste se i druge metode [13, 14].

Na primjer, algoritam predstavljen u [14] sastoji se od sljedećih koraka:

1. Definira se domena unutar koje se izvodi pretraživanje. Domena je oblika $[l_1, u_1] \times \dots \times [l_n, u_n]$, $l_i \leq u_i$, $i = 1, \dots, n$.
2. Definira se skup točaka S u kojima će se gledati vrijednost funkcije i to tako da se na slučajan način odrede njihove koordinate unutar domene. S se sastoji od unaprijed poznatog broja točaka.
3. Provodi se modificirano traženje po liniji i to tako da se za svaku točku \mathbf{x} iz S na slučajan način odredi smjer traženja \mathbf{p} , a duljina koraka se definira s $\alpha_k = 2 + 3/(2^{k^2+1})$, gdje je k broj iteracije. Na taj način se odredi nova točka $\bar{\mathbf{x}}$. Ukoliko je vrijednost $\mathbf{F}(\bar{\mathbf{x}}) < \mathbf{F}(\mathbf{x})$, stavi se $\mathbf{x} = \bar{\mathbf{x}}$. Postupak se ponavlja određeni broj puta.
4. Među svim točkama iz S se odredi ona točka $\mathbf{x}^* = (x_1^*, \dots, x_n^*)$ u kojoj je vrijednost funkcije cilja najmanja. Nakon toga, se na temelju vrijednosti parcijalnih derivacija od \mathbf{F} u \mathbf{x}^* promijeni domena pretraživanja. Konkretno: ako je $(\partial \mathbf{F} / \partial x_i)(\mathbf{x}^*) > 0$, tada je $u_i = x_i^*$, a ako je $(\partial \mathbf{F} / \partial x_i)(\mathbf{x}^*) < 0$, tada je $l_i = x_i^*$.
5. Koraci 2–4 se ponavljaju unaprijed određeni broj puta.

Ovakvi algoritmi su robusni, ali mogu biti jako spori.

1.2.2 Metode homotopnog kontinuiteta

Metode homotopnog kontinuiteta su metode u kojima se polazni sustav zamjenjuje sustavom "sličnim" polaznom, ali kojeg je lako riješiti, a onda se postupnim promjenama dolazi do rješavanja polaznog sustava, [2, 3, 4, 8, 9]. U literaturi se te metode znaju navoditi i kao homotopne metode [10], odnosno metode kontinuiteta [2, 3].

Metode homotopnog kontinuiteta rade s polinomima nad kompleksnim brojevima. Općenito sastoje od dvije faze [8, pog. 8]. U prvoj fazi se iz strukture polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ određuje broj rješenja sustava d i stvara početni sustav $\mathbf{G}(\mathbf{x}) = \mathbf{0}$ koji ima točno d regularnih

rješenja. Iz originalnog i početnog sustava se formira homotopija[†] $\mathbf{H} : \mathbb{C}^n \times [0, 1] \rightarrow \mathbb{C}^n$ s

$$\mathbf{h}(\mathbf{x}, t) = \gamma(1-t)\mathbf{G}(\mathbf{x}) + t\mathbf{F}(\mathbf{x}), \quad t \in [0, 1], \quad (1.5)$$

pri čemu je $\gamma \in \mathbb{C}$ slučajan broj.

U drugoj fazi, kako se parametar t mijenja od 0 do 1, metoda prati puteve koji vode od rješenja početnog sustava $\mathbf{G}(\mathbf{x}) = \mathbf{0}$ ($\mathbf{H}(\mathbf{x}, 0) = \mathbf{0}$) prema rješenjima sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ ($\mathbf{H}(\mathbf{x}, 1) = \mathbf{0}$). Neka su za neki $t \in [0, 1]$ rješenja od $\mathbf{H}(\mathbf{x}, t) = \mathbf{0}$ označena s $\mathbf{x}^{1,t}, \dots, \mathbf{x}^{M,t}$. Tada se može očekivati da se rješenja od $\mathbf{H}(\mathbf{x}, t + \Delta t) = \mathbf{0}$ nalaze "blizu" rješenja od $\mathbf{H}(\mathbf{x}, t) = \mathbf{0}$, pa se za njihovo pronalaženje može koristiti Newtonova metoda, uz $\mathbf{x}^{1,t}, \dots, \mathbf{x}^{M,t}$ kao početne točke. Bolja aproksimacija početnih točaka za Newtonovu metodu se postiže ako se koristi linearni prediktor, odnosno ne koriste se samo rješenja od $\mathbf{H}(\mathbf{x}, t) = \mathbf{0}$, nego se koriste i rješenja od $\mathbf{H}(\mathbf{x}, t - \Delta t) = \mathbf{0}$. "Razmak" između rješenja Δt , može biti fiksna, $\Delta t = 1/K$, gdje je K unaprijed određeni broj koraka, a postoje i tehnike kojima se Δt određuje u ovisnosti o problemu (adaptivni korak) [15].

Dobra svojstva koja se očekuju od homotopije su [16]:

1. (jednostavnost) Rješenja od $\mathbf{H}(\mathbf{x}, 0) = \mathbf{0}$ se mogu jednostavno naći.
2. (glatkoća) Ne postoje singulariteti duž puteva rješenja.
3. (dosezivost) Izolirano rješenje višestrukosti m se doseže s točno m puteva.

Metode homotopnog kontinuiteta rade s polinomima nad \mathbb{C}^n , stoga je potrebno na kraju postupka izdvojiti realna rješenja, no i kompleksna rješenja mogu dati korisne informacije. Na primjer, ako neko kompleksno rješenje ima mali imaginarni dio, onda perturbacija parametara problema može dovesti do pojave novog realnog rješenja [9]. Same metode su pogodne za paralelizaciju, jer je praćenje puta jednog rješenja neovisno o ostalim putovima. Metode se dobro ponašaju i ako sustav ima nultočke višestruke kratnosti, što ne mora biti slučaj kod svih metoda.

Slabosti metode homotopnog kontinuiteta su da se ne može ograničiti područje na kojem se traži rješenje i jedini način za pronalazak realnih rješenja je pronalaženje svih rješenja, što usporava algoritam. Na primjer, ukoliko se rješava polinomni sustav s 5 jednadžbi, od kojih je svaka polinom stupnja 3, tada je, po Bezoutovom teoremu, maksimalni broj rješenja sustava $3^5 = 243$, što znači da je potrebno pratiti 243 puteva, a za svaki put treba višekratno primijeniti Newtonovu metodu.

[†]Neka su X i Y topološki prostori. Homotopija iz X u Y je bilo koje neprekidno preslikavanje $F : X \times [0, 1] \rightarrow Y$. Neka su $f, g : X \rightarrow Y$ preslikavanja. Preslikavanja f i g su homotopna ukoliko postoji homotopija iz X u Y takva da je $f(x) = F(x, 0)$ i $g(x) = F(x, 1)$.

Prostori \mathbb{R}^n , odnosno \mathbb{C}^n , koji se promatraju u metodi homotopnog kontinuiteta su topološki.

1.2.3 Algoritmi zasnovani na podjeli

Algoritmi zasnovani na podjeli, nazivaju se još i metode isključivanja ili generalizirana metoda bisekcije [5, 17, 18]. Rade tako da podijele područje pretraživanja na manja područja, a onda isključuju područja u kojima ne može biti rješenja, čuvaju područja u kojima se sigurno nalazi jedno rješenje, a ostala područja dijele na manja područja i ponavljaju postupak.

Neka je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ polinomni sustav u smislu definicije 1.3. Da bi algoritam zasnovan na podjeli uspješno radio potrebno je imati [9]:

- Test funkciju T , koja za svako područje B vraća jednu od tri vrijednosti:
 - 1 – znači da postoji jedinstveni $\mathbf{x} \in B$, takav da je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, tj. da sustav ima jedinstveno rješenje u B ,
 - 1 – znači da ne postoji $\mathbf{x} \in B$, takav da je $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ i
 - 0 – znači da nije moguće utvrditi postoji li jedinstveno rješenje sustava u B ili u B uopće nema rješenja sustava.
- Realnu funkciju koja mjeri veličinu područja B , u oznaci $|B|$.
- Algoritam razdvajanja (engl. *splitting algorithm*) $S(B)$, koji za neki realni broj $0 < \rho < 1$, vraća područja $B_1, \dots, B_k \subset \mathbb{R}^n$ (ili \mathbb{C}^n), $k \geq 2$, takva da je $B \subset \cup_{i=1}^k B_k$ i za svako novo područje B_i vrijedi $|B_i| \leq \rho|B|$.

Algoritam će biti efikasan ako test funkcija T uspijeva za neko područje vratiti ± 1 dok je to područje relativno veliko, tj. isključiti područje iz daljnjeg dijeljenja. Naime, svakim dijeljenjem se povećava broj područja koje treba provjeriti, što usporava algoritam. Najpopularniji pristup je korištenje intervalne aritmetike, a posebno intervalnog Newtonovog testa. Osnovna je ideja izračunati intervalni Newtonov korak [19] i provjeriti preklapanje rezultirajućeg područja s područjem koje se testira.

Glavna prednost algoritama zasnovanih na podjeli je da mogu biti u cijelosti provedeni unutar skupa realnih brojeva, te da se mogu ograničiti samo na jedno područje unutar \mathbb{R}^n , tako da se ne moraju izračunavati rješenja koja nisu zanimljiva ili fizički moguća. Što je još važnije, ovi algoritmi daju matematički pouzdane granice područja u kojima se nalaze rješenja.

Slabosti ima nekoliko. Intervalni Newtonov test ne dovodi do zaključka u okolini singularnog rješenja i tada metoda konvergira sporo. Ukoliko postoje rješenja sustava koja nisu izolirana, tada metoda vraća područja koja pokrivaju cijeli skup takvih rješenja, što može potpuno narušiti postupak računanja. Konačno, intervalna aritmetika ne daje precizne rezultate i sa svakom aritmetičkom operacijom rezultati su neprecizniji, što pogotovo dolazi do izražaja kod funkcija s velikim brojem operacija. Generalno, za male sustave ovi algoritmi mogu biti efikasni, dok kod većih sustava slabosti više dolaze do izražaja [9].

1.2.4 Algebarski i geometrijski algoritmi

Algebarski i geometrijski algoritmi se razlikuju od prethodno spomenutih algoritama jer zahtijevaju više simboličkog računa, a manje numeričkih metoda, tako da ih neki autori i ne razlikuju. Manocha [10] u svojoj podjeli algoritama, obje vrste algoritama svrstava u kategoriju simbolički algoritmi. On ističe da se algoritmima zasnovanim na upotrebi Gröbnerove baze i rezultanti mogu eliminirati varijable i na taj način svesti problem rješavanja sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ na problem traženja nultočaka polinoma jedne varijable. Međutim, takvi algoritmi su efikasni samo na problemima s 3 do 4 jednadžbe, jer traženje nultočaka polinoma velikog stupnja (većeg od 15) može biti loše uvjetovano [20].

U [10] je predstavljena metoda kojom se pomoću rezultanti rješenje od $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ može tražiti rješavanjem problema svojstvenih vrijednosti matrice. Pristup kojim se rješavanje polinomnog sustava svodi na rješavanje problema svojstvenih vrijednosti može se naći i u [21], gdje su prikazane i druge primjene Gröbnerove baze.

Generalno, za ove metode se može reći da daju dobre rezultate na malim sustavima, a u slučajevima kada su sva rješenja nesingularna i izolirana, metode mogu biti vrlo brze. Međutim, za sustave većeg stupnja i s više varijabli ove metode mogu postati vrlo zahtjevne [9].

1.3 Metode spusta

U ovom radu se promatra novi pristup traženju rješenja polinomnog sustava zasnovan na metodi najdublje spusta. U osnovi, metoda najdublje spusta pripada metodama minimizacije realne funkcije više varijabli koje se nazivaju metode spusta.

Metode spusta su [22] iterativne metode koje, počevši od neke zadane točke $\mathbf{x}^{(0)} \in \mathbb{R}^n$, generiraju niz točaka $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots$, sve dok nije zadovoljen neki od uvjeta zaustavljanja. Preciznije, neka je zadana funkcija $f : \mathbb{R}^n \rightarrow \mathbb{R}$ za koju se traži minimum. U metodama spusta, za zadanu točku $\mathbf{x}^{(i)} \in \mathbb{R}^n$ te za zadani smjer $\mathbf{s}^{(i)} \in \mathbb{R}^n$, nova točka $\mathbf{x}^{(i+1)} \in \mathbb{R}^n$ određuje se kao točka na pravcu određenom točkom $\mathbf{x}^{(i)}$ i smjerom $\mathbf{s}^{(i)}$, tj. $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \lambda_i \mathbf{s}^{(i)}$, pri čemu je $\lambda_i \in \mathbb{R}$ duljina koraka, za koju vrijedi $f(\mathbf{x}^{(i+1)}) < f(\mathbf{x}^{(i)})^\ddagger$. Način na koji se za zadanu točku $\mathbf{x}^{(i)}$ određuju smjer spusta $\mathbf{s}^{(i)}$ i duljina koraka λ_i definira samu metodu.

Neka je $\nabla f(\mathbf{x}^{(i)})$ gradijent funkcije f u točki $\mathbf{x}^{(i)}$. U većini metoda spusta se zahtijeva da smjer spusta $\mathbf{s}^{(i)}$ zatvara kut veći od $\pi/2$ s $\nabla f(\mathbf{x}^{(i)})$, odnosno da vrijedi $(\mathbf{s}^{(i)})^T \nabla f(\mathbf{x}^{(i)}) < 0$. Ako smjer $\mathbf{s}^{(i)}$ zadovoljava spomenuti uvjet govori se da je $\mathbf{s}^{(i)}$ smjer silaska (eng. *descent direction*). Za duljinu koraka λ_i se pretpostavlja da vrijedi $\lambda_i > 0$. U idealnom slučaju λ_i bi se

[‡]Postoje metode spusta u kojima se ne zahtijeva da vrijedi $f(\mathbf{x}^{(i+1)}) < f(\mathbf{x}^{(i)})$, ali one zahtijevaju da vrijedi $f(\mathbf{x}^{(i+m)}) < f(\mathbf{x}^{(i)})$, za neki cijeli broj $m \geq 2$.

moglo odrediti tako da vrijedi

$$\lambda_i = \arg \min_{\lambda > 0} f(\mathbf{x}^{(i)} + \lambda \mathbf{s}^{(i)}),$$

ali taj uvjet je najčešće teško zadovoljiti pa se za određivanje duljine koraka koristi neka varijanta algoritma pretraživanja po pravcu (eng. *line search*).

Navedimo najpoznatije i najupotrebljavanije smjerove spusta prema [22].

- Smjer suprotan od smjera gradijenta funkcije, $\mathbf{s}^{(i)} = -\nabla f(\mathbf{x}^{(i)})$.
- Newtonov smjer, za čije određivanje je potrebna Hesseova matrica $\nabla^2 f(\mathbf{x}^{(i)})$, čiji su elementi određeni s $(\nabla^2 f(\mathbf{x}^{(i)}))_{j,k} = \frac{\partial^2 f}{\partial x_j \partial x_k}(\mathbf{x}^{(i)})$. Newtonov smjer $\mathbf{s}^{(i)}$ određuje se kao rješenje jednadžbe

$$\nabla^2 f(\mathbf{x}^{(i)}) \mathbf{s}^{(i)} = -\nabla f(\mathbf{x}^{(i)}). \quad (1.6)$$

- Smjer dobiven nekom od kvazi-Newtonovih metoda (npr. BFGS metoda). Za određivanje smjera $\mathbf{s}^{(i)}$ potrebno je riješiti jednadžbu $\mathbf{B}^{(i)} \mathbf{s}^{(i)} = -\nabla f(\mathbf{x}^{(i)})$, pri čemu matrica $\mathbf{B}^{(i)} \in \mathbf{M}_n$ predstavlja aproksimaciju Hesseove matrice.
- Smjer dobiven nekom od metoda konjugiranog gradijenta, $\mathbf{s}^{(i)} = -\nabla f(\mathbf{x}^{(i)}) + \beta_i \mathbf{s}^{(i-1)}$.

Prva spomenuta metoda u kojoj je $\mathbf{s}^{(i)}$ suprotan smjeru gradijenta se obično naziva metodom najstrmijeg spusta (eng. *steepest descent*). S jedne strane, ona je izrazito jednostavna jer za određivanje smjera je dovoljno odrediti gradijent funkcije, no s druge strane, pokazuje se da u određenim situacijama može biti jako spora.

Metoda u kojoj se za smjer spusta koristi Newtonov smjer je značajna jer u slučaju kada se $\mathbf{x}^{(i)}$ nalazi blizu rješenja, konvergencija je kvadratna. Glavni nedostatak je potreba za izračunom Hesseove matrice $\nabla^2 f(\mathbf{x}^{(i)})$ u svakom koraku iterativnog postupka. Eksplicitno izračunavanje u općem slučaju je složeno, a izračun derivacija numeričkim putem je računalno zahtjevan proces. Da bi se izbjegla potreba za računanjem Hesseove matrice, koriste se različite metode za njenu aproksimaciju koje se zbirno nazivaju kvazi-Newtonove metode. Ideja je da se u svakom koraku nova aproksimacija Hesseove matrice $\mathbf{B}^{(i)}$ dobije iz prethodne aproksimacije $\mathbf{B}^{(i-1)}$ i neke jednostavne korekcije. Kvazi-Newtonove metode su atraktivne jer su relativno jednostavne za izračun, a red konvergencije je superlinearan. Neke varijante kvazi-Newtonove metode su napravljene tako da se u svakom koraku ne obnavlja matrica $\mathbf{B}^{(i)}$ nego $(\mathbf{B}^{(i)})^{-1}$, tako da se i rješavanje sustava može izbjeći.

Konačno, metode konjugiranog gradijenta su obično efikasnije od metode najstrmijeg spusta, ali ne dosežu brzinu konvergencije kao metode koje koriste Newtonov ili kvazi-Newtonov smjer. Prednost metoda konjugiranog gradijenta je što njihova implementacija nema velike memorijske zahtjeve jer nije potrebno čuvati nikakvu matricu, već samo nekoliko vektora.

Metoda najdublje spusta također pripada metodama spusta, a glavna razlika je u načinu izbora duljine koraka silaska. Za zadanu točku $\mathbf{x} \in \mathbb{R}^n$ i smjer silaska $\mathbf{s} \in \mathbb{R}^n$, nova točka

iterativnog postupka, $\bar{\mathbf{x}} \in \mathbb{R}^n$, je oblika $\bar{\mathbf{x}} = \mathbf{x} + \lambda^* \mathbf{s}$, a duljina koraka $\lambda^* \in \mathbb{R}$ je određena kao točka u kojoj se postiže globalni minimum restrikcije funkcije f na točke pravca definiranog smjerom \mathbf{s} i točkom \mathbf{x} . Formalno, definira li se funkcija $f_{\mathbf{x},\mathbf{s}} : \mathbb{R} \rightarrow \mathbb{R}$ s $f_{\mathbf{x},\mathbf{s}}(\lambda) = f(\mathbf{x} + \lambda\mathbf{s})$, može se pisati da je

$$\lambda^* = \arg \min_{\lambda \in \mathbb{R}} f_{\mathbf{x},\mathbf{s}}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} f(\mathbf{x} + \lambda\mathbf{s}). \quad (1.7)$$

Premda je funkcija $f_{\mathbf{x},\mathbf{s}}$ realna funkcija realne varijable, problem pronalaženja njenog globalnog minimuma je općenito vrlo težak, bez obzira koriste li se metode za traženje minimuma funkcije jedne varijable ili se koriste metode za rješavanje jednadžbe

$$\frac{d}{d\lambda} f_{\mathbf{x},\mathbf{s}}(\lambda) = 0 \Leftrightarrow (\nabla f(\mathbf{x} + \lambda\mathbf{s}))^T \mathbf{s} = 0. \quad (1.8)$$

Naime, funkcija $f_{\mathbf{x},\mathbf{s}}$ je općenito nelinearna i za izračunati barem lokalni minimum potreban je veliki broj evaluacija funkcije f , odnosno njezina gradijenta [22], stoga je određivanje duljine koraka rješavanjem problema (1.7), odnosno (1.8) u općem slučaju nepraktično, ali kod polinomnih sustava se pokazuje da je funkcija $f_{\mathbf{x},\mathbf{s}}$ polinom određenog stupnja, a za polinome postoje algoritmi pomoću kojih je moguće riješiti probleme (1.7), odnosno (1.8), u razumnom vremenu.

Metoda najdublje spusta ima prednost da može pronaći područje u kojem je vrijednost funkcije manja čak i ako se nalazi u okolini lokalnog ekstrema, što standardne metode spusta ne mogu.

1.4 Metode za traženje nultočaka polinoma

Problem traženja nultočaka polinoma pripada među najstarije i najvažnije probleme u matematici. Postupak za rješavanje kvadratne jednadžbe je bio poznat još u starom Babilonu te arapskim srednjovjekovnim matematičarima. U šesnaestom stoljeću su talijanski matematičari Cardano i Ferrari dali postupak za rješavanje polinomne jednadžbe stupnja tri, odnosno četiri. No, sredinom devetnaestog stoljeća, norveški matematičar Nils Abel je dokazao da nije moguće naći rješenje opće polinomne jednadžbe stupnja većeg od četiri pomoću radikala [23], čime je dan veliki poticaj razvoju metoda za iterativno traženje nultočaka polinoma. Koncem devetnaestog stoljeća je dokazan i osnovni teorem algebre koji tvrdi da svaki polinom stupnja n s kompleksnim koeficijentima ima točno n kompleksnih nultočaka ako se broji njihova višestrukost.

Literatura o traženju nultočaka polinoma je vrlo opsežna. U monografiji *Numerical Methods for Roots of Polynomials* ([24],[25]), McNamee spominje da bibliografija o traženju nultočaka polinoma, koju je on sastavio, ima oko 8000 jedinica, od čega je u 2005. nastalo 50 jedinica.

Metoda za traženje nultočaka polinoma ima jako puno i o ovom poglavlju je nemoguće

nabrojati sve metode, a mogu se u grubo podijeliti u tri grupe.

Prvu grupu čine metode temeljene na postupku deflacije. Prvo se odredi jedna nultočka polinoma x^* , a onda provedu deflacija, odnosno originalni polinom se podijeli s $(x - x^*)$. Na taj način dobije se polinom nižeg stupnja, a postupak se ponavlja na njemu. Za traženje jedne nultočke polinoma može se koristiti Newtonova metoda, neka od metoda baziranih na interpolaciji poput metode sekante (linearna interpolacija) ili Mullerova metode (kvadratna interpolacija), Laguerrova metoda, Jenkins-Traubova metoda ([26]) itd.

U drugoj grupi nalaze se one metode koje pokušavaju istovremeno naći sve nultočke polinoma. Za zadani polinom p stupnja n , iterativnim postupkom, koji svoj izvor ima u Weierstrassovu dokazu osnovnog teorema algebre, žele se naći sve nultočke polinoma p i to tako da se započne s n međusobno relativno udaljenih točaka u kompleksnoj ravnini. Najpoznatije metode ovog tipa su Durant-Kernerova ([27]) i Aberhtova metoda, ali postoji još puno metoda istog tipa. Na primjer, jedan korak Durant-Kernerove metode je dan s

$$z_i^{(k+1)} = z_i^{(k)} - \frac{p(z_i^{(k)})}{\prod_{\substack{j=0 \\ j \neq i}}^n (z_i^{(k)} - z_j^{(k)})}, \quad i = 1, \dots, n, \quad k = 0, 1, \dots$$

Ove metode se izvode u kompleksnoj aritmetici, a rezultat su kompleksne nultočke. Eksperimentalno je utvrđeno da je postupak skoro uvijek konvergentan.

Posljednju grupu metoda koja će se ovdje spomenuti je grupa metoda u kojima se problem traženja nultočaka polinoma pretvara u problem traženja svojstvenih vrijednosti pripadne matrice ([28]). Konkretno, ako je $p(z) = \sum_{i=0}^n a_i z^i$, $a_i \in \mathbb{C}$, polinom stupnja n s kompleksnim koeficijentima čiji je vodeći koeficijent $a_n = 1$, tada se nultočke polinoma p poklapaju sa svojstvenim vrijednostima matrice

$$\begin{bmatrix} 0 & \dots & 0 & -a_0 \\ 1 & \ddots & \vdots & -a_1 \\ & \ddots & 0 & \vdots \\ 0 & \dots & 1 & -a_{n-1} \end{bmatrix}.$$

Ovakva vrsta algoritma se koristi pogotovo kod polinoma velikog stupnja. Unutar MATLAB-a, naredba `roots` je implementirana baš korištenjem ovakvih metoda.

Poglavlje 2

Rješavanje polinomnih sustava metodom najdublje spusta

Tema ovog rada je primjena metoda najdublje spusta na rješavanje polinomnih sustava.

Neka je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ polinomni sustav u smislu definicije 1.3, $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Da bi se metoda najdublje spusta mogla primijeniti na rješavanje polinomnog sustava, prvo je potrebno definirati funkciju $g : \mathbb{R}^n \rightarrow \mathbb{R}$ koja je povezana s polinomnim sustavom $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ u smislu da su izolirana rješenja polinomnog sustava ujedno i rješenja problema minimizacije funkcije g ,

$$\min_{\mathbf{x} \in \mathbb{R}^n} g(\mathbf{x}).$$

Nakon toga, problem minimizacije funkcije g se rješava primjenom metode najdublje spusta. Ako je funkcija g polinom, onda će i njena restrikcija na pravac određen zadanim smjerom i točkom kroz koju prolazi biti također polinom, čime se rješavanje problema (1.7), odnosno (1.8) svodi na rješavanje problema nalaženja ekstrema polinoma jedne varijable, odnosno traženje nultočaka polinoma jedne varijable. Najčešće se za prateću funkciju bira kvadrat euklidske norme reziduala (*RSS*), definiran s

$$RSS(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2 = \sum_{i=1}^m f_i^2(\mathbf{x}).$$

Problem traženje nultočaka polinoma jedne varijable jedan je od najstarijih i najistraživanijih problema u matematici [29], za koji je razvijen veliki broj metoda rješavanja [24, 25]. Metode variraju od primjene klasičnih metoda koje se mogu koristiti za rješavanje nelinearnih jednadžbi (Newtonova metoda, metoda sekante), do raznih metoda koje pronalaze istovremeno sve nultočke polinoma, matrice metode itd. (potpoglavlje 1.4).

Opća shema rješavanja polinomnog sustava metodom najdublje spusta prikazana je u Algoritmu 2.1.

Algoritam 2.1 Opća shema rješavanja polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ metodom najdublje spusta.

Pretpostavke: definirana funkcija $g : \mathbb{R}^n \rightarrow \mathbb{R}$ povezana sa sustavom $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

- 1: Neka je zadana početna točka $\mathbf{x}^{(0)}$.
 - 2: $k=0$
 - 3: **while** nije zadovoljen kriterij zaustavljanja
 - 4: Odredi smjer $\mathbf{s}^{(k)}$.
 - 5: Izračunaj duljinu koraka $\lambda^{(k)}$ kao $\lambda^{(k)} = \arg \min_{\lambda \in \mathbb{R}} g(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$.
 - 6: Nova točka je $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)} \mathbf{s}^{(k)}$.
 - 7: $k = k + 1$
 - 8: **end while**
-

U ostatku poglavlja prikazana je evolucija primjene metode najdublje spusta na problem rješavanja polinomnog sustava. Prvo je prikazana varijanta metode za rješavanje kvazilinearnih polinomnih sustava [30, 31] u kojoj se za smjer spusta koriste koordinatne osi. Nakon toga je prikazano proširenje te varijante na opće polinomne sustave te varijanta metode u kojoj se za smjer spusta koristi proizvoljni vektor. Na samom kraju poglavlja prikazana je primjena metode najdublje spusta na rješavanje linearnih sustava.

2.1 Rješavanje kvazilinearnih sustava

Kvazilinearni sustavi su polinomni sustavi kod kojih je svaka jednadžba kvazilinearni polinom u više varijabli. Polinom u više varijabli je kvazilinearan ako se može zapisati kao linearna kombinacija monoma oblika $x_1^{d_1} \dots x_n^{d_n}$ pri čemu eksponent d_i može biti samo 0 ili 1.

Formalno, neka je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ polinomni sustav, $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$. $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ je kvazilinearni polinomni sustav ako je svaka od funkcija $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, kvazilinearni polinom u više varijabli.

U [30, 31] je prikazano rješavanje kvazilinearnih sustava metodom najdublje spusta. Preciznije, u radu se traži rješenje problema $\|\mathbf{F}(\mathbf{x})\|_2 \rightarrow \min$, pri čemu je $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ i za $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ vrijedi da je svaka od funkcija f_1, \dots, f_m kvazilinearni polinom.

Metoda najdublje spusta promatrana u [31] se sastoji od toga da se za dani sustav formira funkcija RSS : $\mathbb{R}^n \rightarrow \mathbb{R}$ kao zbroj kvadrata reziduala,

$$\text{RSS}(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2 = \sum_{i=1}^m f_i^2(\mathbf{x}),$$

a onda se pomoću iterativnog postupka traži \mathbf{x}^* koji minimizira RSS(\mathbf{x}). Kao smjerovi spusta koriste se koordinatne osi.

Preciznije, neka su zadane točka $\mathbf{x} = (x_1, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_n)$ i koordinatna os $\mathbf{e}_j =$

$(0, \dots, 0, \underbrace{1}_j, 0, \dots, 0)^T$ (smjer spusta). Proizvoljna točka s pravca određenog s \mathbf{x} i \mathbf{e}_j opisana je s $\mathbf{x} + \lambda \mathbf{e}_j$, $\lambda \in \mathbb{R}$. Neka je s $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}$ označena restrikcija funkcije RSS na točke s pravca koji prolazi točkom \mathbf{x} i ima smjer \mathbf{e}_j . Vrijedi da je $\text{RSS}_{\mathbf{x}, \mathbf{e}_j} : \mathbb{R} \rightarrow \mathbb{R}$ i $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda) = \text{RSS}(\mathbf{x} + \lambda \mathbf{e}_j)$. Obzirom da je f_i kvazilinearni polinom, za svaki f_i vrijedi da je $f_i(\mathbf{x} + \lambda \mathbf{e}_j)$ oblika $a_i \lambda + b_i$, stoga je

$$\begin{aligned} \text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda) &= \sum_{i=1}^m f_i^2(\mathbf{x} + \lambda \mathbf{e}_j) = \sum_{i=1}^m (a_i \lambda + b_i)^2 \\ &= \left(\sum_{i=1}^m a_i^2 \right) \lambda^2 + 2 \left(\sum_{i=1}^m a_i b_i \right) \lambda + \sum_{i=1}^m b_i^2. \end{aligned}$$

Ako je $\sum_{i=1}^m a_i^2 = 0$, onda za sve a_i vrijedi $a_i = 0$, pa je $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda) = \sum_{i=1}^m b_i^2$, odnosno RSS je na točkama pravca određenog s \mathbf{x} i \mathbf{e}_j konstanta. U tom slučaju, odabere se druga koordinatna os i postupak se ponovi. S druge strane, ako je $\sum_{i=1}^m a_i^2 \neq 0$, onda je $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ kvadratna funkcija po varijabli λ koja postiže najmanju vrijednost u točki λ^* ,

$$\lambda^* = \frac{\sum_{i=1}^m a_i b_i}{\sum_{i=1}^m a_i^2}. \quad (2.1)$$

Sada se definira nova točka $\bar{\mathbf{x}} = (x_1, \dots, x_{j-1}, x_j + \lambda^*, x_{j+1}, \dots, x_n)$, za koju vrijedi $\text{RSS}(\bar{\mathbf{x}}) \leq \text{RSS}(\mathbf{x})$, a postupak se ponavlja dok se ne steknu uvjeti za zaustavljanje procesa.

Neka je \mathbf{x}^* točka u kojoj se postupak zaustavlja. Ako vrijedi da je $\text{RSS}(\mathbf{x}^*) = 0$, točka \mathbf{x}^* je rješenje polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

U [30, 31] korištene su dvije strategije odabira smjera spusta. U prvoj strategiji za smjer spusta se biraju koordinatne osi redom $(e_1, e_2, \dots, e_n, e_1, e_2, \dots)$, dok se u drugoj, “pohlepnoj” varijanti, za vektor spusta bira ona koordinatna os kod koje je uočeno najveće smanjenje funkcije RSS. U [31] je napravljena analiza ponašanja algoritma u terminu postotka uspješnosti, tj. u koliko slučajeva će algoritam uspjeti naći nultočku sustava. Na skupu od 500 slučajno generiranih kvazilinearnih sustava uočeno je da je uspješnost rješavanja na razini Levenberg–Marquardtove metode, te da uspješnost rješavanja pada povećavanjem broja jednadžbi i nepoznanica. Algoritam 2.2 je implementacija “pohlepne” strategije biranja smjera silaska.

Kako su linearni sustavi zapravo podskup kvazilinearnih sustava, metoda najdublje spusta se može direktno primijeniti na njih. Više o tome u potpoglavlju 2.7.

Algoritam 2.2 Rješavanje kvazilinearnog polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ metodom najdublje spusta po koordinatnim osima.

```

1: Neka je zadana početna točka  $\mathbf{x}^{(0)}$ .
2:  $k=0$ 
3: while nije zadovoljen kriterij zaustavljanja
4:    $RSS_{min} = RSS(\mathbf{x}^{(k)})$ .
5:   for  $j = 1$  to  $n$ 
6:     Odredi  $\lambda_j^* = \arg \min_{\lambda \in \mathbb{R}} RSS_{\mathbf{x}^{(k)}, \mathbf{e}_j}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} RSS(\mathbf{x}^{(k)} + \lambda \mathbf{e}_j)$  koristeći (2.1).
7:     if  $RSS(\mathbf{x}^{(k)} + \lambda_j^* \mathbf{e}_j) < RSS_{min}$  then
8:        $\mathbf{s}^{(k)} = \mathbf{e}_j$ ,  $\lambda^{(k)} = \lambda_j^*$ 
9:        $RSS_{min} = RSS(\mathbf{x}^{(k)} + \lambda_j^* \mathbf{e}_j)$ 
10:    end if
11:  end for
12:  Nova točka je  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)} \mathbf{s}^{(k)}$ .
13:   $k = k + 1$ 
14: end while

```

2.2 Rješavanje općih polinomnih sustava – slučaj kada je vektor spusta koordinatna os

Kako je svaki kvazilinearni sustav ujedno i polinomni sustav, razumno je pretpostaviti da bi se metoda opisana u poglavlju 2.1 mogla prilagoditi tako da radi i na općim polinomnim sustavima. U principu, sve ostaje isto jedino se mijenja postupak izračuna duljine koraka λ^* , jer u slučaju općeg polinomnog sustava izračun λ^* se svodi na rješavanje polinomne jednadžbe u jednoj varijabli.

Neka je zadan opći polinomni sustav

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n,$$

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})), \quad f_1, \dots, f_m : \mathbb{R}^n \rightarrow \mathbb{R}, \quad (2.2)$$

f_i realni polinomi u više varijabli.

Kao i kod kvazilinearnih sustava, definira se funkcija $RSS : \mathbb{R}^n \rightarrow \mathbb{R}$ s

$$RSS(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2 = \sum_{i=1}^m f_i^2(\mathbf{x}). \quad (2.3)$$

Pretpostavimo da je zadana točka $\mathbf{x} = (x_1, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_n)$ te neka je \mathbf{e}_j koordinatna os u smjeru koje se želi naći nova točka u iterativnom postupku, $\bar{\mathbf{x}}$. U cilju pronalaženja točke $\bar{\mathbf{x}}$ promatra se funkcija $RSS_{\mathbf{x}, \mathbf{e}_j}$, koja predstavlja restrikciju funkcije RSS na pravac koji prolazi

točkom \mathbf{x} i zadan je smjerom \mathbf{e}_j ,

$$\text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda) = \text{RSS}(\mathbf{x} + \lambda \mathbf{e}_j) = \text{RSS}(x_1, \dots, x_{j-1}, x_j + \lambda, x_{j+1}, \dots, x_n). \quad (2.4)$$

Duljina koraka spusta λ^* se određuje kao točka u kojoj se postiže minimum funkcije $\text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda)$,

$$\lambda^* = \arg \min_{\lambda \in \mathbb{R}} \text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} \text{RSS}(\mathbf{x} + \lambda \mathbf{e}_j) \quad (2.5)$$

pa je nova točka iterativnog postupka $\bar{\mathbf{x}} = \mathbf{x} + \lambda^* \mathbf{e}_j = (x_1, \dots, x_{j-1}, x_j + \lambda^*, x_{j+1}, \dots, x_n)$.

Preostaje još vidjeti kakvog je oblika funkcija $\text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda)$. Prema definiciji 1.2 vrijedi da se proizvoljni polinom u više varijabli $f: \mathbb{R}^n \rightarrow \mathbb{R}$ može zapisati kao linearna kombinacija članova $\mathbf{x}^{\mathbf{d}_r}$, $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $\mathbf{d}_r = (d_{1,r}, \dots, d_{n,r}) \in \mathbb{N}_0^n$,

$$f(\mathbf{x}) = \sum_{r=1}^k a_r \mathbf{x}^{\mathbf{d}_r}, \quad \mathbf{x}^{\mathbf{d}_r} = x_1^{d_{1,r}} \cdots x_n^{d_{n,r}}.$$

Sljedeće dvije propozicije daju bolji uvid u strukturu funkcije $f(\mathbf{x} + \lambda \mathbf{e}_j)$.

Propozicija 2.1 *Neka su zadani $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $\mathbf{d} = (d_1, \dots, d_n)$, $d_i \in \mathbb{N}_0$. Nadalje, neka su zadane koordinatna os $\mathbf{e}_j \in \mathbb{R}^n$ i $\lambda \in \mathbb{R}$.*

Tada je

$$(\mathbf{x} + \lambda \mathbf{e}_j)^{\mathbf{d}} = x_1^{d_1} \cdots x_{j-1}^{d_{j-1}} \cdot (x_j + \lambda)^{d_j} \cdot x_{j+1}^{d_{j+1}} \cdots x_n^{d_n}.$$

Dokaz: Direktnim uvrštavanjem. ■

Na temelju propozicije 2.1, lako se pokazuje i sljedeća propozicija kojom su opisana svojstva polinoma $f_i(\mathbf{x} + \lambda \mathbf{e}_j)$.

Propozicija 2.2 *Neka su zadani $\mathbf{x} \in \mathbb{R}^n$, koordinatna os \mathbf{e}_j i polinom*

$$f(\mathbf{x}) = \sum_{r=1}^k a_r \mathbf{x}^{\mathbf{d}_r}, \quad \mathbf{x}^{\mathbf{d}_r} = x_1^{d_{1,r}} \cdots x_{j-1}^{d_{j-1,r}} \cdot x_j^{d_{j,r}} \cdot x_{j+1}^{d_{j+1,r}} \cdots x_n^{d_{n,r}}, \quad \mathbf{d}_r \in \mathbb{N}_0^n.$$

Neka su članovi od $f(\mathbf{x})$ poredani silazno prema eksponentu od x_j te neka je $1 \leq k^M \leq k$ određen tako da vrijedi

$$d_j^M = d_{j,1} = \dots = d_{j,k^M} > d_{j,k^M+1} \geq \dots, \quad d_j^M = \max_r d_{j,r}. \quad (2.6)$$

Tada je $f_{\mathbf{x},\mathbf{e}_j}(\lambda) = f(\mathbf{x} + \lambda \mathbf{e}_j)$ polinom u varijabli λ čiji je stupanj manji ili jednak d_j^M i koji se može zapisati kao

$$f_{\mathbf{x},\mathbf{e}_j}(\lambda) = \sum_{s=0}^{d_j^M} \bar{b}_{s,j} (x_j + \lambda)^s.$$

Koeficijenti polinoma su

$$\bar{b}_{d_j^M, j} = \sum_{r=1}^{k^M} b_r, \quad \bar{b}_{s, j} = \sum_{\substack{r=k^M+1 \\ d_{j,r}=s}}^k b_{r, j}, \quad s = 0, \dots, d_j^M - 1,$$

gdje je

$$b_{r, j} = a_r \cdot x_1^{d_{1,r}} \cdots x_{j-1}^{d_{j-1,r}} \cdot x_{j+1}^{d_{j+1,r}} \cdots x_n^{d_{n,r}} = a_r \frac{\mathbf{x}^{\mathbf{d}_r}}{x_j^{d_{j,r}}}, \quad r = 1, \dots, k. \quad (2.7)$$

Također, $\deg f_{\mathbf{x}, \mathbf{e}_j} = d_j^M$ ako i samo ako je $\bar{b}_{d_j^M, j} = \sum_{r=1}^{k^M} b_{r, j} \neq 0$.

Dokaz: Pretpostavimo prvo da je $k^M < k$, tj. da postoje članovi polinoma koji nemaju jednaki eksponent uz x_j . Koristeći propoziciju 2.1 i uvažavajući (2.6) dobije se

$$\begin{aligned} f_{\mathbf{x}, \mathbf{e}_j}(\lambda) &= f(\mathbf{x} + \lambda \mathbf{e}_j) = \sum_{r=1}^k a_r (\mathbf{x} + \lambda \mathbf{e}_j)^{\mathbf{d}_r} \\ &= \sum_{r=1}^k a_r \cdot \underbrace{x_1^{d_{1,r}} \cdots x_{j-1}^{d_{j-1,r}} \cdot x_{j+1}^{d_{j+1,r}} \cdots x_n^{d_{n,r}}}_{b_{r, j}} \cdot (x_j + \lambda)^{d_{j,r}} \\ &= \sum_{r=1}^{k^M} b_{r, j} \cdot (x_j + \lambda)^{d_j^M} + \sum_{r=k^M+1}^k b_{r, j} \cdot (x_j + \lambda)^{d_{j,r}} \\ &= \left(\sum_{r=1}^{k^M} b_{r, j} \right) \cdot (x_j + \lambda)^{d_j^M} + \sum_{s=0}^{d_j^M-1} \bar{b}_{s, j} \cdot (x_j + \lambda)^s. \end{aligned} \quad (2.8)$$

Iz (2.8) slijedi tvrdnja propozicije.

Ako svi članovi polinoma imaju jednaki eksponent uz x_j (slučaj $k = k^M$) dokaz se provodi analogno, samo je nešto jednostavniji:

$$\begin{aligned} f_{\mathbf{x}, \mathbf{e}_j}(\lambda) &= \sum_{r=1}^{k^M} a_r (\mathbf{x} + \lambda \mathbf{e}_j)^{\mathbf{d}_r} = \sum_{r=1}^{k^M} a_r \cdot \underbrace{x_1^{d_{1,r}} \cdots x_{j-1}^{d_{j-1,r}} \cdot x_{j+1}^{d_{j+1,r}} \cdots x_n^{d_{n,r}}}_{b_{r, j}} \cdot (x_j + \lambda)^{d_{j,r}} \\ &= \left(\sum_{r=1}^{k^M} b_{r, j} \right) \cdot (x_j + \lambda)^{d_j^M}. \end{aligned}$$

■

Propozicija 2.2 je važna jer je u njoj dan praktičan postupak za izračun koeficijenata polinoma $f_{\mathbf{x}, \mathbf{e}_j}(\lambda) = \sum_{s=0}^{d_j^M} \bar{b}_{s, j} (x_j + \lambda)^s$. Koeficijent $\bar{b}_{s, j}$ se dobije tako da se zbroje vrijednosti $b_{r, j}$ za one članove kojima je eksponent uz x_j jednak s . Ako takvih $b_{r, j}$ nema, onda je $\bar{b}_{s, j} = 0$. U praksi će se $b_{r, j}$ računati primjenom formule (2.7), ovisno o načinu na koji se polinom i njegovi članovi spremaju u računalo.

Također, propozicija je bitna i zato što ukazuje na potencijalni izvor numeričkih problema prilikom računanja koeficijenta polinoma $f_{\mathbf{x},\mathbf{e}_j}(\lambda)$. Npr. koeficijent uz $\lambda^{d_j^M}$ u polinomu $f_{\mathbf{x},\mathbf{e}_j}(\lambda)$ je $\sum_{r=1}^{k^M} b_r$. Ako su svi b_r istog predznaka, problema neće biti. No, ako su b_r različitog predznaka, može doći do oduzimanja brojeva koji su sličnih apsolutnih vrijednosti što nije numerički stabilno. Krajnja posljedica je da koeficijent uz $\lambda^{d_j^M}$ bude 0, a trebala bi biti neka mala vrijednost različita od nule ili obrnuto. Da stvar nije beznačajna, pokazuje jedan jednostavni primjer: polinom $p(x) = x^3 + x$ ima jednu realnu nultočku, $x = 0$. S druge strane, za mali $\varepsilon > 0$, polinom $-\varepsilon x^5 + x^3 + x$ ima tri realne nultočke: $x_1 = 0, x_{2,3} = \pm \sqrt{\frac{\sqrt{4\varepsilon + 1} + 1}{2\varepsilon}}$.

Propozicija 2.2 ne daje odgovor na spomenuti problem već više ukazuje na potencijalni izvor problema, pogotovo ako se uzme u obzir da koeficijenti uz λ^k u polinomu $f_{\mathbf{x},\mathbf{e}_j}(\lambda)$ ne ovise samo o koordinatnoj osi e_j , već i o točki $\mathbf{x} = (x_1, \dots, x_n)$.

Kad se propozicija 2.2 primijeni na polinome f_i koji čine polinomni sustav, dobiju se polinomi $f_{i,\mathbf{x},\mathbf{e}_j}$ za koje vrijedi

$$f_{i,\mathbf{x},\mathbf{e}_j}(\lambda) = f_i(\mathbf{x} + \lambda \mathbf{e}_j) = \sum_{s=0}^{d_j^{M,i}} \bar{b}_{s,j}^i (x_j + \lambda)^s. \quad (2.9)$$

Koeficijenti $\bar{b}_{s,i}$ dobiveni su kao u propoziciji 2.2, a $d_j^{M,i}$ je najveći eksponent uz x_j u polinomu $f_i(\mathbf{x})$. Također vrijedi i da je $\deg f_{i,\mathbf{x},\mathbf{e}_j} \leq d_j^{M,i}$.

Koristeći (2.9), slijedi da se $\text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda)$ može zapisati kao polinom u varijabli $x_j + \lambda$. Naime, iz definicije od $\text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda)$ i (2.9) slijedi

$$\begin{aligned} \text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda) &= \sum_{i=1}^m f_{i,\mathbf{x},\mathbf{e}_j}^2(\lambda) = \sum_{i=1}^m \left(\sum_{s=0}^{d_j^{M,i}} \bar{b}_{s,j}^i (x_j + \lambda)^s \right)^2 \quad [\text{stavimo } t = x_j + \lambda] \\ &= \sum_{i=1}^m \left(\sum_{s=0}^{d_j^{M,i}} \left(\bar{b}_{s,j}^i \right)^2 \cdot t^{2s} + 2 \sum_{s=1}^{d_j^{M,i}} \sum_{r=0}^{s-1} \bar{b}_{s,j}^i \cdot \bar{b}_{r,j}^i \cdot t^{s+r} \right) \\ &= \sum_{s=0}^{\deg \text{RSS}_{\mathbf{x},\mathbf{e}_j}} \bar{a}_s t^s. \end{aligned} \quad (2.10)$$

Iz (2.10) se može zaključiti da je $\text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda)$ polinom parnog stupnja u varijabli λ , s time da za stupanj od $\text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda)$ vrijedi da je

$$\deg \text{RSS}_{\mathbf{x},\mathbf{e}_j} = 2 \max_i \deg f_{i,\mathbf{x},\mathbf{e}_j} \leq 2 \max_i d_j^{M,i}$$

i koeficijent uz potenciju najvišeg stupnja je pozitivan. Također, (2.10) povlači da je

$$\deg \text{RSS}_{\mathbf{x}, \mathbf{e}_j} = 0 \quad \text{ako i samo ako je} \quad \deg f_{i, \mathbf{x}, \mathbf{e}_j} = 0, \quad i = 1, \dots, m.$$

Ako je polinom $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ konstanta, onda treba promijeniti smjer spusta, jer se u smjeru osi \mathbf{e}_j ništa neće promijeniti.

Iduća lema govori da je problem određivanja duljine koraka kao točke u kojoj se postiže minimum funkcije $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ dobro definiran i da se minimum od $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ dostiže u nekoj od nultočaka derivacije.

Lema 2.3 *Neka je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ polinomni sustav zadan s (2.2), \mathbf{x} zadana točka, \mathbf{e}_j smjer spusta i $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ definirana s (2.4).*

Ako je $\deg \text{RSS}_{\mathbf{x}, \mathbf{e}_j} > 0$, tada postoji $\lambda^ = \arg \min_{\lambda \in \mathbb{R}} \text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$.*

Nadalje, neka je $\{\lambda_1, \dots, \lambda_k\}$, $1 \leq k \leq \deg \text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda) - 1$, skup svih realnih rješenja jednadžbe $(d\text{RSS}_{\mathbf{x}, \mathbf{e}_j}/d\lambda)(\lambda) = 0$. Tada je

$$\lambda^* = \arg \min_{\lambda \in \{\lambda_1, \dots, \lambda_k\}} \text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda).$$

Dokaz: Ako je $\deg \text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda) = 0$ (polinom $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ je konstanta) onda se nema što dokazivati. Neka je $\deg \text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda) > 0$. Zbog činjenica da je $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ polinom parnog stupnja, a koeficijent uz monom najvišeg stupnja pozitivan, vrijedi

$$\lim_{\lambda \rightarrow -\infty} \text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda) = \infty \quad \text{i} \quad \lim_{\lambda \rightarrow \infty} \text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda) = \infty,$$

iz čega slijedi da je neki od lokalnih minimuma funkcije $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ ujedno i njen globalni minimum. S druge strane, kako je $(d\text{RSS}_{\mathbf{x}, \mathbf{e}_j}/d\lambda)(\lambda)$ polinom neparnog stupnja, sigurno postoji barem jedno realno rješenje jednadžbe $(d\text{RSS}_{\mathbf{x}, \mathbf{e}_j}/d\lambda)(\lambda) = 0$. Stoga se $\lambda^* = \min_{\lambda \in \mathbb{R}} \text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ može izračunati tako da se među svim nultočkama polinoma $(d\text{RSS}_{\mathbf{x}, \mathbf{e}_j}/d\lambda)(\lambda)$ odabere ona za koju funkcija $\text{RSS}_{\mathbf{x}, \mathbf{e}_j}(\lambda)$ ima najmanju vrijednost. ■

Prikaz prethodno opisanog postupka u obliku algoritma dan je u algoritmu 2.3, koji slijedi opću shemu danu u algoritmu 2.1:

- funkcija povezana sa sustavom je RSS,
- smjer silaska je koordinatna os na kojoj je zabilježeno najveće smanjenje funkcije RSS,
- duljina koraka je dobivene kao globalni minimum od RSS na pravcu koji prolazi zadanom točkom i smjerom silaska.

Na žalost, uspješnost algoritma, čak i za sustave s malim brojem nepoznanica i jednadžbi, nije pretjerano visoka, stoga je potrebno proširiti pristup rješavanju tako da se za smjer silaska bira proizvoljan vektor.

Algoritam 2.3 Rješavanje općeg polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ metodom najdublje spusta po koordinatnim osima.

- 1: Neka je zadana početna točka $\mathbf{x}^{(0)}$.
 - 2: $k=0$
 - 3: **while** nije zadovoljen kriterij zaustavljanja
 - 4: $RSS_{min} = RSS(\mathbf{x}^{(k)})$.
 - 5: **for** $j = 1$ **to** n
 - 6: Korištenjem (2.7) nađi koeficijente $\bar{b}_{s,j}^i$ polinoma $f_{i,\mathbf{x},\mathbf{e}_j}(\lambda) = \sum_{s=0}^{d_j^{M,i}} \bar{b}_{s,j}^i (x_j + \lambda)^s$.
 - 7: Izračunaj koeficijente \bar{a}_s polinoma $RSS_{\mathbf{x}^{(k)},\mathbf{e}_j}(t)$ koristeći (2.10).
 - 8: Odredi $\lambda_j^* = \arg \min_{\lambda \in \mathbb{R}} RSS_{\mathbf{x}^{(k)},\mathbf{e}_j}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} RSS(\mathbf{x}^{(k)} + \lambda \mathbf{e}_j)$ prema lemi 2.3.
 - 9: **if** $RSS(\mathbf{x}^{(k)} + \lambda_j^* \mathbf{e}_j) < RSS_{min}$ **then**
 - 10: $\mathbf{s}^{(k)} = \mathbf{e}_j$, $\lambda^{(k)} = \lambda_j^*$
 - 11: $RSS_{min} = RSS(\mathbf{x}^{(k)} + \lambda_j^* \mathbf{e}_j)$
 - 12: **end if**
 - 13: **end for**
 - 14: Nova točka je $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)} \mathbf{s}^{(k)}$.
 - 15: $k = k + 1$
 - 16: **end while**
-

2.3 Rješavanje općih polinomnih sustava – slučaj proizvoljnog vektora spusta

U potpoglavlju 2.2 opisan je postupak za slučaj kada su vektori spusta koordinatne osi. U ovom potpoglavlju promatra se izvod metode za slučaj kada je vektor spusta proizvoljni vektor \mathbf{s} .

Neka je, kao u (2.2), zadan opći polinomni sustav $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ te neka je funkcija $RSS : \mathbb{R}^n \rightarrow \mathbb{R}$ definirana s (2.3).

Pretpostavimo da je zadana točka $\mathbf{x} = (x_1, \dots, x_n)$ i vektor $\mathbf{s} = (s_1, \dots, s_n)^T$ koji predstavlja smjer spusta, odnosno smjer u kojem se želi naći sljedeća točka iterativnog postupka, $\bar{\mathbf{x}} = \mathbf{x} + \lambda^* \mathbf{s}$. Duljina koraka spusta, λ^* , kao i u potpoglavlju 2.2, određuje se kao točka u kojoj se postiže globalni minimum funkcije $RSS_{\mathbf{x},\mathbf{s}}(\lambda)$, koja je restrikcija funkcije RSS na pravac određen točkom \mathbf{x} i vektorom smjera \mathbf{s} ,

$$RSS_{\mathbf{x},\mathbf{s}}(\lambda) = RSS(\mathbf{x} + \lambda \mathbf{s}),$$

$$\lambda^* = \arg \min_{\lambda \in \mathbb{R}} RSS_{\mathbf{x},\mathbf{s}}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} RSS(\mathbf{x} + \lambda \mathbf{s}).$$

Za izračunati λ^* potrebno je bolje upoznati oblik funkcije $RSS_{\mathbf{x},\mathbf{s}}$, a za to će od koristi biti sljedeća propozicija.

Propozicija 2.4 Neka su zadani $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $\mathbf{s} = (s_1, \dots, s_n) \in \mathbb{R}^n$ i $\mathbf{d} = (d_1, \dots, d_n)$,

$d_i \in \mathbb{N}_0$, $\tilde{d} = d_1 + \dots + d_n$. Neka je $\lambda \in \mathbb{R}$.

Tada je $(\mathbf{x} + \lambda \mathbf{s})^{\mathbf{d}}$ polinom u varijabli λ čiji je stupanj manji ili jednak \tilde{d} . Nadalje, ako je $s_1^{d_1} \dots s_n^{d_n} \neq 0$, onda je $\deg(\mathbf{x} + \lambda \mathbf{s})^{\mathbf{d}} = \tilde{d}$ i koeficijent uz $\lambda^{\tilde{d}}$ je $s^{\mathbf{d}}$.

Napomena: Umnožak $s_1^{d_1} \dots s_n^{d_n} = 0$ ako postoji barem jedan indeks i takav da je $s_i = 0$ i $d_i > 0$.

Dokaz: Neka je $\tilde{d} = d_1 + \dots + d_n$. Prema definiciji člana (Definicije 1.1), za $(\mathbf{x} + \lambda \mathbf{s})^{\mathbf{d}}$ vrijedi

$$\begin{aligned} (\mathbf{x} + \lambda \mathbf{s})^{\mathbf{d}} &= (x_1 + \lambda s_1)^{d_1} \dots (x_n + \lambda s_n)^{d_n} \\ &= (s_1^{d_1} \lambda^{d_1} + P_1^{(d_1-1)}(\lambda)) \cdot (s_2^{d_2} \lambda^{d_2} + P_2^{(d_2-1)}(\lambda)) \dots (s_n^{d_n} \lambda^{d_n} + P_n^{(d_n-1)}(\lambda)) \\ &= s_1^{d_1} \dots s_n^{d_n} \cdot \lambda^{d_1 + \dots + d_n} + P^{(d_1 + \dots + d_n - 1)}(\lambda) \\ &= s_1^{d_1} \dots s_n^{d_n} \cdot \lambda^{\tilde{d}} + P^{(\tilde{d}-1)}(\lambda), \end{aligned} \quad (2.11)$$

gdje je $s P_i^{(K)}(\lambda)$ označen realni polinom u varijabli λ čiji je stupanj manji ili jednak K . Iz (2.11) slijedi da ako je $s_1^{d_1} \dots s_n^{d_n} \neq 0$, onda je $(\mathbf{x} + \lambda \mathbf{s})^{\mathbf{d}}$ polinom stupnja jednakog \tilde{d} , a ako je $s_1^{d_1} \dots s_n^{d_n} = 0$, onda je stupanj polinoma $(\mathbf{x} + \lambda \mathbf{s})^{\mathbf{d}}$ manji od \tilde{d} . ■

Neka je $f(\mathbf{x}) = \sum_{r=1}^k a_r \mathbf{x}^{\mathbf{d}_r}$ polinom u više varijabli te neka su članovi polinoma poredani silazno prema svojim stupnjevima, $\deg \mathbf{x}^{\mathbf{d}_1} \geq \deg \mathbf{x}^{\mathbf{d}_2} \geq \dots$. Iz definicije 1.2 je poznato da je stupanj polinoma f jednak najvećem stupnju člana, no nas zanima kojeg je stupnja polinom $f_{\mathbf{x},s}(\lambda) = f(\mathbf{x} + \lambda \mathbf{s})$. Sljedeća propozicija daje gornju ogradu za stupanj tog polinoma.

Propozicija 2.5 Neka je $f(\mathbf{x}) = \sum_{r=1}^k a_r \mathbf{x}^{\mathbf{d}_r}$ polinom u više varijabli te neka je $\deg f(\mathbf{x}) = \tilde{d}$. Neka su članovi od $f(\mathbf{x})$ poredani silazno prema stupnju člana te neka je $1 \leq k^M \leq k$ određen tako da vrijedi

$$\tilde{d} = \deg \mathbf{x}^{\mathbf{d}_1} = \dots = \deg \mathbf{x}^{\mathbf{d}_{k^M}} > \deg \mathbf{x}^{\mathbf{d}_{k^M+1}} \geq \dots \quad (2.12)$$

Tada je

$$f_{\mathbf{x},s}(\lambda) = f(\mathbf{x} + \lambda \mathbf{s}) = \left(\sum_{r=1}^{k^M} a_r \mathbf{s}^{\mathbf{d}_r} \right) \lambda^{\tilde{d}} + P^{(\tilde{d}-1)}(\lambda), \quad (2.13)$$

gdje je $P^{(\tilde{d})}(\lambda)$ polinom stupnja najviše $\tilde{d} - 1$. Nadalje,

$$\deg f_{\mathbf{x},s} \leq \tilde{d} = \deg f(\mathbf{x}),$$

a jednakost vrijedi ako i samo ako je koeficijent $\sum_{r=1}^{k^M} a_r \mathbf{s}^{\mathbf{d}_r} \neq 0$.

Dokaz: Neka je $k^M < k$, odnosno postoje članovi polinoma $f(\mathbf{x})$ koji nisu istog stupnja. Iz

propozicije 2.4 i uvjeta (2.12) slijedi

$$\begin{aligned} f_{\mathbf{x},\mathbf{s}}(\lambda) &= f(\mathbf{x} + \lambda \mathbf{s})(\lambda) \\ &= \sum_{r=1}^{k^M} a_r \mathbf{x}^{\mathbf{d}_r} + \sum_{r=k^M+1}^k a_r \mathbf{x}^{\mathbf{d}_r} \end{aligned} \quad (2.14)$$

$$\begin{aligned} &= \sum_{r=1}^{k^M} a_r (\mathbf{s}^{\mathbf{d}_r} \lambda^{\tilde{d}} + P_r^{(\tilde{d}-1)}(\lambda)) + \sum_{r=k^M+1}^k a_r P_r^{(\tilde{d}-1)}(\lambda) \\ &= \left(\sum_{r=1}^{k^M} a_r \mathbf{s}^{\mathbf{d}_r} \right) \lambda^{\tilde{d}} + P^{(\tilde{d}-1)}(\lambda), \end{aligned} \quad (2.15)$$

gdje, kao i u dokazu propozicije 2.4, $P_i^{(K)}(\lambda)$ označava polinom čiji je stupanj manji ili jednak K . Iz 2.15 slijedi da je $\deg f_{\mathbf{x},\mathbf{s}} \leq \tilde{d}$, a jednakost će se postići samo ako je član uz vodeću potenciju različit od nule, odnosno ako vrijedi $\sum_{r=1}^{k^M} a_r \mathbf{s}^{\mathbf{d}_r} \neq 0$.

Ako su svi članovi polinoma $f(\mathbf{x})$ istog stupnja, onda je $k^M = k$, a dokaz se provodi na isti način:

$$f_{\mathbf{x},\mathbf{s}} = \sum_{r=1}^{k^M} a_r \mathbf{x}^{\mathbf{d}_r} = \sum_{r=1}^{k^M} a_r (\mathbf{s}^{\mathbf{d}_r} \lambda^{\tilde{d}} + P_r^{(\tilde{d}-1)}(\lambda)) = \left(\sum_{r=1}^{k^M} a_r \mathbf{s}^{\mathbf{d}_r} \right) \lambda^{\tilde{d}} + P^{(\tilde{d}-1)}(\lambda).$$

■

Na ovom mjestu je potrebno napomenuti da je izračun koeficijenta $\sum_{r=1}^{k^M} a_r \mathbf{s}^{\mathbf{d}_r}$ numerički osjetljiv jer tu može doći do zbrajanja bliskih brojeva i gubitka broja točno izračunatih znamenki. Naime, neka je s_{A_E} označena točna vrijednost koeficijenta $\sum_{r=1}^{k^M} a_r \mathbf{s}^{\mathbf{d}_r}$, a neka je s_{A_S} vrijednost koja se dobije računanjem vrijednosti $\sum_{r=1}^{k^M} a_r \mathbf{s}^{\mathbf{d}_r}$ u aritmetici pomičnog zareza. Sa stajališta određivanja stupnja polinoma $f_{\mathbf{x},\mathbf{s}}$, moguća su tri slučaja:

- ili je $A_E \neq 0$ i $A_S \neq 0$ ili su A_E i A_S jednaki 0 – ovo je najpovoljniji slučaj jer će stupanj polinoma $f_{\mathbf{x},\mathbf{s}}$ biti točno određen.
- $A_E \neq 0$ i $A_S = 0$ – ovaj slučaj je malo vjerojatan u smislu da će A_S biti točno 0, ali je opasan jer je izgubljen stupanj polinoma.
- $A_E = 0$, a $A_S \neq 0$ – ovaj slučaj može nastupiti češće od prethodnog i jako ga je teško prepoznati.

Srećom, u praksi se događa da najčešće nastupa slučaj (a), a slučaj (c) se rijetko pojavljuje.

Najvažnija posljedica propozicije 2.5 je ta što daje kriterija po kojem se može jednostavno odrediti vrijedi li $\deg f_{\mathbf{x},\mathbf{s}} = \deg f$. Ako to nije slučaj, onda određivanje $\deg f_{\mathbf{x},\mathbf{s}}$ nije jednostavno. Ta činjenica će se koristiti u poglavlju 2.4.

Kao i u poglavlju 2.2, propozicija 2.5 se koristi za konstrukciju polinoma $RSS_{\mathbf{x},\mathbf{s}}(\lambda)$, samo

što u slučaju općenitog smjera spusta ima više tehničkih detalja.

Prvo se rezultat propozicije primijeni na polinome f_i koji čine polinomni sustav. Dobiju se polinomi $f_{i,\mathbf{x},\mathbf{s}}(\lambda) = f_i(\mathbf{x} + \lambda\mathbf{s})$, za koje vrijedi

$$f_{i,\mathbf{x},\mathbf{s}}(\lambda) = f_i(\mathbf{x} + \lambda\mathbf{s}) = \left(\sum_{r=1}^{k_i^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}} \right) \lambda^{\tilde{d}_i} + P_i^{(\tilde{d}_i-1)}(\lambda).$$

Po definiciji je

$$\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda) = \sum_{i=1}^n f_{i,\mathbf{x},\mathbf{s}}^2(\lambda) = \sum_{i=1}^n \left(\left(\sum_{r=1}^{k_i^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}} \right) \lambda^{\tilde{d}_i} + P_i^{(\tilde{d}_i-1)}(\lambda) \right)^2. \quad (2.16)$$

Iz definicije je jasno da je polinom $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$ polinom parnog stupnja te da vrijedi

$$\deg \text{RSS}_{\mathbf{x},\mathbf{s}} = 2 \max_i \deg f_{i,\mathbf{x},\mathbf{s}}.$$

Kako stupanj polinoma $f_{i,\mathbf{x},\mathbf{s}}(\lambda)$ ovisi i o točki \mathbf{x} i smjeru \mathbf{s} , tako se i stupanj polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$ mijenja iz iteracije u iteraciju.

Prikaz polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$ u obliku (2.16) nije pogodan za određivanje duljine koraka spusta, već $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$ treba zapisati u obliku

$$\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda) = \sum_{s=0}^{\deg \text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)} b_s \lambda^s. \quad (2.17)$$

Način na koji se to može napraviti predstavlja poseban problem o kojem će više riječi biti u poglavlju 2.4.

Kao i u slučaju kada je smjer spusta koordinatna os, i za proizvoljni smjer spusta može se dati kriterij za određivanje stupnja polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$.

Neka je maksimalni stupanj sustava, u oznaci \tilde{d}^M , definiran s

$$\tilde{d}^M = \max_i \deg f_i(\mathbf{x}) = \max_i \tilde{d}_i. \quad (2.18)$$

Za stupanj polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$ vrijedi $\deg \text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda) \leq 2 \cdot \tilde{d}^M$. Sljedeća lema daje jasni kriterij kada je stupanj polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$ jednak maksimalnom stupnju sustava.

Lema 2.6 *Ako postoji barem jedan polinom $f_{i,\mathbf{x},\mathbf{s}}(\lambda)$ čiji je stupanj jednak maksimalnom stupnju sustava ($\deg f_{i,\mathbf{x},\mathbf{s}} = \tilde{d}^M$), onda je*

$$\deg \text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda) = 2 \cdot \tilde{d}^M.$$

Dokaz: Prema propoziciji 2.5, svaki polinom $f_{i,\mathbf{x},\mathbf{s}}(\lambda)$ se može zapisati kao

$$f_{i,\mathbf{x},\mathbf{s}}(\lambda) = f_i(\mathbf{x} + \lambda\mathbf{s}) = \left(\sum_{r=1}^{k^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}} \right) \lambda^{\tilde{d}_i} + P^{(\tilde{d}_i-1)}(\lambda).$$

Bez smanjenja općenitosti, može se pretpostaviti da prvih $1 \leq n^M \leq n$ polinoma $f_i(\mathbf{x})$ ima stupanj jednak maksimalnom stupnju sustava:

$$\tilde{d}^M = \deg f_1(\mathbf{x}) = \dots = \deg f_{n^M}(\mathbf{x}) > \deg f_{n^M+1}(\mathbf{x}) \geq \dots \geq \deg f_n(\mathbf{x}).$$

Tada je

$$\begin{aligned} \text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda) &= \sum_{i=1}^{n^M} f_{i,\mathbf{x},\mathbf{s}}^2(\lambda) + \sum_{i=n^M+1}^n f_{i,\mathbf{x},\mathbf{s}}^2(\lambda) \\ &= \sum_{i=1}^{n^M} \left(\left(\sum_{r=1}^{k^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}} \right)^2 \lambda^{2\tilde{d}^M} + P_i^{(2\tilde{d}^M-1)}(\lambda) \right) + \sum_{i=n^M+1}^n P_i^{(2\tilde{d}^M-2)}(\lambda) \\ &= \left(\sum_{i=1}^{n^M} \left(\sum_{r=1}^{k^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}} \right)^2 \right) \lambda^{2\tilde{d}^M} + P^{(2\tilde{d}^M-1)}(\lambda), \end{aligned}$$

pri čemu je $s P^{(K)}(\lambda)$ označen polinom čiji je stupanj manji ili jednak K . Član uz $\lambda^{2\tilde{d}^M}$ je različit od nule ako je barem jedan od koeficijenta $\sum_{r=1}^{k^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}}$, $i = 1, \dots, n^M$, različit od nule, što prema propoziciji 2.5 znači da barem jedan od prvih n^M polinoma $f_i(\mathbf{x} + \lambda\mathbf{s})$ ima stupanj jednak maksimalnom stupnju sustava. ■

Kada je polinom $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$ formiran, problem pronalaženja duljine koraka je dobro definirano, o čemu govori sljedeća lema.

Lema 2.7 *Neka je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ polinomni sustav zadan s (2.2), \mathbf{x} zadana točka, \mathbf{s} smjer spusta i $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$ definiran s $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda) = \text{RSS}(\mathbf{x} + \lambda\mathbf{s})$.*

Ako je $\deg \text{RSS}_{\mathbf{x},\mathbf{s}} > 0$, tada postoji $\lambda^ = \arg \min_{\lambda \in \mathbb{R}} \text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$.*

Nadalje, neka je $\{\lambda_1, \dots, \lambda_k\}$, $1 \leq k \leq \deg \text{RSS}_{\mathbf{x},\mathbf{e}_j}(\lambda) - 1$, skup svih realnih rješenja jednadžbe $(d\text{RSS}_{\mathbf{x},\mathbf{s}}/d\lambda)(\lambda) = 0$. Tada je

$$\lambda^* = \arg \min_{\lambda \in \{\lambda_1, \dots, \lambda_k\}} \text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda).$$

Dokaz: Dokaz se provodi na isti način kao i za lemu 2.3. ■

Slično kao i za slučaj kada je smjer spusta koordinatna os, postupak za traženje rješenja polinomnog sustava metodom najdublje spusta se može zapisati kao algoritam 2.4.

U algoritmu 2.4 postoje problemi koji nisu do kraja riješeni:

- Na koji način odabrati vektor spusta $\mathbf{s}^{(k)}$ u zadanoj točki $\mathbf{x}^{(k)}$?
- Koji su kriteriji zaustavljanja iterativnog postupka?
- Kako odrediti prikaz polinoma $\text{RSS}_{\mathbf{x}^{(k)}, \mathbf{s}^{(k)}}(t)$ u obliku (2.17)?
- Na koji način izračunati λ^* , odnosno kako riješiti jednadžbu oblika $p(x) = 0$ za p koji je realni polinom?

Svako od ovih pitanja je dovoljno složeno da zaslužuje poseban osvrt.

Odabir vektora spusta određuje metodu. U ovom radu promatraju se različiti načini odabira vektora spusta kao što su Newtonov smjer, smjer najstrmijeg spusta, smjer koordinatnih osi i drugi. Također, moguće je u jednom koraku isprobati više smjerova spusta pa odabrati onaj koji daje najmanju vrijednost funkcije $\text{RSS}_{\mathbf{x}, \mathbf{s}}$. Detaljni pregled načina na koji se može odabrati smjer spusta dan je u poglavlju 3.

Za potrebe isprobavanja metoda najdublje spusta napisan je program PolySolver, a u njemu se za traženje nultočaka polinoma koristi Jenkins–Traubov algoritam ([26]). Originalni algoritam za traženje nultočaka realnog polinoma koji je napisan u programskom jeziku FORTRAN (rpoly.f) je preveden u Visual Basic. Nikakvih značajnih promjena nije bilo osim kod nekoliko konstanti povezanih s prikazom broja u pokretnom zarezu. Jenkins–Traubov algoritam koristi i program Mathematica za traženje nultočaka polinoma, u slučaju kada korisnik primijeni naredbu NSolve, a argument je polinom.

Kriteriji zaustavljanja iterativnog postupka su detaljno objašnjeni u poglavlju 2.5, dok je način na koji se polinom $\text{RSS}_{\mathbf{x}^{(k)}, \mathbf{s}^{(k)}}(t)$ prikazuje u obliku (2.17) opisan u sljedećem potpoglavlju.

Algoritam 2.4 Rješavanje općeg polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ metodom najdublje spusta po proizvoljnom smjeru.

- 1: Neka je zadana početna točka $\mathbf{x}^{(0)}$.
- 2: $k=0$
- 3: **while** nije zadovoljen kriterij zaustavljanja
- 4: Neka je neki način određen smjer $\mathbf{s}^{(k)}$.
- 5: Odredi koeficijente b_s tako da je

$$\text{RSS}_{\mathbf{x}^{(k)}, \mathbf{s}^{(k)}}(t) = \sum_{s=0}^{\deg \text{RSS}_{\mathbf{x}^{(k)}, \mathbf{s}^{(k)}}} b_s t^s.$$

- 6: Odredi $\lambda^* = \arg \min_{\lambda \in \mathbb{R}} \text{RSS}_{\mathbf{x}^{(k)}, \mathbf{s}^{(k)}}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} \text{RSS}(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$ prema lemi 2.7.
 - 7: Nova točka je $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)} \mathbf{s}^{(k)}$.
 - 8: $k = k + 1$
 - 9: **end while**
-

2.4 Određivanje koeficijenata polinoma $RSS_{\mathbf{x},\mathbf{s}}(\lambda)$

Neka se iterativni postupak nalazi u točki \mathbf{x} te neka je zadan smjer spusta \mathbf{s} . U poglavlju 2.3 spomenuto je da za potrebe izračuna duljine koraka u zadanom smjeru, polinom $RSS_{\mathbf{x},\mathbf{s}}$ treba zapisati u standardnoj bazi prostora polinoma $\{1, x, x^2, \dots\}$, u obliku

$$RSS_{\mathbf{x},\mathbf{s}}(\lambda) = \sum_{s=0}^{\deg RSS_{\mathbf{x},\mathbf{s}}} b_s \lambda^s.$$

Da bi se taj problem uspješno riješio, potrebno je napraviti algoritam koji će efektivno i točno odrediti koeficijente $b_s, s = 0, \dots, \deg RSS_{\mathbf{x},\mathbf{s}}(\lambda)$.

Na raspolaganju ima više mogućnosti, a u ovom radu se koriste dva postupka:

- računanje koeficijenata korištenjem simboličkog računanja i
- računanje koeficijente korištenjem numeričke interpolacije.

Metoda računanja koeficijenata korištenjem simboličkog računa bazira se na činjenici da je dva polinoma koja su zapisana u standardnoj bazi prostora polinoma jednostavno zbrojiti i pomnožiti. Svaki od polinoma $f_{i,\mathbf{x},\mathbf{s}}$ se može zapisati kao linearna kombinacija svojih članova:

$$f_{i,\mathbf{x},\mathbf{s}}(\lambda) = f_i(\mathbf{x} + \lambda \mathbf{s}) = \sum_{r=1}^{k_i} a_{r,i}(\mathbf{x} + \lambda \mathbf{s})^{\mathbf{d}_{r,i}},$$

a za članove vrijedi

$$(\mathbf{x} + \lambda \mathbf{s})^{\mathbf{d}_{r,i}} = (x_1 + \lambda s_1)^{d_{r,i}^1} \cdots (x_n + \lambda s_n)^{d_{r,i}^n}, \quad \mathbf{d}_{r,i} = (d_{r,i}^1, \dots, d_{r,i}^n).$$

Dakle, svaki član je ili potencija binoma (izraz oblika $(x + s \cdot \lambda)^d, d \geq 1$) ili umnožak dvije ili više potencija binoma. Kako se u standardnoj bazi prostora polinoma potencija binoma prikazuje pomoću binomne formule

$$(x + s \cdot \lambda)^d = \sum_{i=0}^d \binom{d}{i} x^i \cdot s^{d-i} \cdot \lambda^{d-i},$$

slijedi da se prikaz člana u standardnoj bazi dobije kao umnožak prikaza potencije binoma u standardnoj bazi, dok se prikaz polinoma $f_{i,\mathbf{x},\mathbf{s}}$ u standardnoj bazi dobije kao linearna kombinacija prikaza svojih članova. Koeficijenti od $RSS_{\mathbf{x},\mathbf{s}}(\lambda)$ se dobiju zbrajanjem kvadrata polinoma $f_i(\mathbf{x} + \lambda \mathbf{s})$.

Glavni nedostatak ovog pristupa je brzina jer je potreban veliki broj računskih operacija da bi se izračunali koeficijenti b_s . Također, moguće je da i prilikom izračuna koeficijenata dođe do oduzimanja bliskih brojeva.

Drugi pristup je korištenje interpolacije polinomom, ali da bi se takav pristup mogao koristiti

potrebno je znati koliki je stupanj polinoma kojim se interpolira, pri čemu veliku važnost imaju kriteriji dani u propoziciji 2.5 i lemi 2.6.

Kako je već prije spomenuto, za stupanj polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}$ vrijedi

$$\deg \text{RSS}_{\mathbf{x},\mathbf{s}} = 2 \max_i \deg f_{i,\mathbf{x},\mathbf{s}} \leq 2 \max_i \deg f_i = 2\tilde{d}^M$$

Nadalje, iz dokaza leme 2.6 proizlazi da je stupanj polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}$ jednak $2 \cdot \tilde{d}^M$ ako i samo ako vrijedi

$$\sum_{\substack{i \\ \deg f_i(\mathbf{x}) = \tilde{d}^M}} \left(\sum_{r=1}^{k_i^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}} \right)^2 > 0. \quad (2.19)$$

Uvjet (2.19) je relativno jednostavno provjeriti, ali postoji problem potencijalne nestabilnosti koji se javlja kod izračuna veličina $\sum_{r=1}^{k_i^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}}$, gdje može doći do oduzimanja bliskih brojeva.

U slučaju da je $\deg \text{RSS}_{\mathbf{x},\mathbf{s}} < 2\tilde{d}^M$, nema dobrog kriterija za određivanje $\deg \text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$, ali može se koristiti sljedeća lema koja daje dovoljan uvjet kada je $\deg \text{RSS}_{\mathbf{x},\mathbf{s}} = 2\tilde{d}^M - 2$.

Lema 2.8 *Neka je $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x})) = \mathbf{0}$ polinomni sustav te neka je $\tilde{d}^M = \max_i \deg f_i$. Neka je zadana točka \mathbf{x} i vektor spusta \mathbf{s} te neka ne postoji niti jedan polinom $f_{i,\mathbf{x},\mathbf{s}}$ koji zadovoljava uvjete leme 2.6 ($\deg f_{i,\mathbf{x},\mathbf{s}} = \tilde{d}^M$).*

Ako postoji barem jedan polinom $f_{i,\mathbf{x},\mathbf{s}}$ za kojeg vrijedi $\deg f_{i,\mathbf{x},\mathbf{s}} = \deg f_i = \tilde{d}^M - 1$, tada je $\deg \text{RSS}_{\mathbf{x},\mathbf{s}} = 2\tilde{d}^M - 2$.

Dokaz: Ako postoji polinom $f_{i,\mathbf{x},\mathbf{s}}$ za kojeg vrijedi $\deg f_{i,\mathbf{x},\mathbf{s}} = \deg f_i = \tilde{d}^M - 1$, tada je prema propoziciji 2.5 koeficijent $\sum_{r=1}^{k_i^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}} \neq 0$.

Iz konstrukcije polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}$ i činjenice da je barem jedan od polinoma stupnja $\tilde{d}^M - 1$ slijedi da je $\deg \text{RSS}_{\mathbf{x},\mathbf{s}} \geq 2\tilde{d}^M - 2$, a kako je u uvjetima leme navedeno da ne postoji polinom $f_{i,\mathbf{x},\mathbf{s}}$ čiji je stupanj jednak \tilde{d}^M , tada je $\deg \text{RSS}_{\mathbf{x},\mathbf{s}} = 2\tilde{d}^M - 2$. ■

Obrat tvrdnje općenito ne vrijedi: stupanj polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}$ može biti $2\tilde{d}^M - 2$, a ne treba postojati niti jedan polinom $f_{i,\mathbf{x},\mathbf{s}}$ takav da je $\deg f_{i,\mathbf{x},\mathbf{s}} = \deg f_i = \tilde{d}^M - 1$. Naime, neka je $f_{i,\mathbf{x},\mathbf{s}}$ polinom za kojeg vrijedi $\deg f_i = \tilde{d}^M$. Prema propoziciji 2.5, vrijedi

$$f_{i,\mathbf{x},\mathbf{s}}(\lambda) = \left(\sum_{r=1}^{k_i^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}} \right) \lambda^{\tilde{d}^M} + P_i^{(\tilde{d}^M-1)}(\lambda),$$

gdje je $P_i^{(\tilde{d}^M-1)}$ polinom čiji je stupanj manji ili jednak $\tilde{d}^M - 1$. Po pretpostavci leme je $\sum_{r=1}^{k_i^M} a_{r,i} \mathbf{s}^{\mathbf{d}_{r,i}} = 0$ pa je $\deg f_{i,\mathbf{x},\mathbf{s}} = \deg P_i^{(\tilde{d}^M-1)}$. Ako vrijedi $\deg P_i^{(\tilde{d}^M-1)} = \tilde{d}^M - 1$, onda je si-

gurno $\deg \text{RSS}_{x,s} = 2\tilde{d}^M - 2$, bez obzira postoji li polinom $f_{j,x,s}$, takav da je $\deg f_{j,x,s} = \deg f_j = \tilde{d}^M - 1$.

Iz leme 2.6 slijedi nužan i dovoljan uvjet koji treba vrijediti da bi polinom $\text{RSS}_{x,s}$ bio maksimalnog stupnja (uvjet (2.19)), dok lema 2.8 daje dovoljan uvjet da bi stupanj od $\text{RSS}_{x,s}$ bio $2\tilde{d}^M - 2$. Ako niti jedan od tih uvjeta nije ispunjen, onda sam stupanj polinoma i nije bitan jer će se za određivanje koeficijenata polinoma $\text{RSS}_{x,s}$ koristiti metoda koja koristi simbolički račun.

Problemu određivanja koeficijenta polinoma $\text{RSS}_{x,s}$ putem interpolacije polinomom moguće je pristupiti na dva načina:

- pomoću interpolacije polinomom odrediti polinome $f_{i,x,s}(\lambda)$, a polinom $\text{RSS}_{x,s}(\lambda)$ odrediti kao zbroj kvadrata polinoma $f_{i,x,s}(\lambda)$, odnosno
- pomoću interpolacije polinomom odmah odrediti polinom $\text{RSS}_{x,s}(\lambda)$.

U oba slučaja postupak interpolacije je isti. Neka je \mathcal{P}_K skup svih realnih polinoma stupnja manje ili jednako K , te neka je $f \in \mathcal{P}_K$, takav da je $\deg f = K$. Za polinom f se želi naći interpolacijski polinom p , koji se s polinomom f poklapa u $K + 1$ točaka x_0, x_1, \dots, x_K . Interpolacijski polinom se može zapisati na više načina, a ovdje se koristi Newtonov interpolacijski polinom. Polinom p je definiran s [29]

$$p(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_0, x_1, \dots, x_K](x - x_0)(x - x_1) \dots (x - x_{K-1}),$$

gdje se veličine $f[x_i, \dots, x_{i+s}]$ nazivaju podijeljene razlike i definirane su rekurzivno s

$$\begin{aligned} f[x_i] &= f(x_i), \quad i = 0, \dots, K; \\ f[x_i, \dots, x_{i+s}] &= \frac{f[x_{i+1}, \dots, x_{i+s}] - f[x_i, \dots, x_{i+s-1}]}{x_{i+s} - x_i}, \quad s = 1, \dots, K, \quad i = 0, \dots, K - s. \end{aligned}$$

Polinom $p \in \mathcal{P}_K$, $\deg P = K$, a kako se s polinomom f poklapa u $K + 1$ točaka, vrijedi $p(x) = f(x)$, za svaki $x \in \mathbb{R}$. Budući da je p Newtonov interpolacijski polinom, može se reći da je p zapis polinoma f u bazi $B_N = \{1, x - x_0, (x - x_0)(x - x_1), \dots, (x - x_0)(x - x_1) \dots (x - x_{K-1})\}$ prostora \mathcal{P}_K . Za postupak deriviranja i traženja nultočaka, povoljnije bi bilo da je polinoma p zapisan u standardnoj bazi prostora \mathcal{P}_K , $B_S = \{1, x, \dots, x^K\}$.

Za prebacivanje u standardnu bazu koristi se postupak utemeljen na razvoju funkcije u Taylorov red oko $x = 0$ te Hornerovoj shemi adaptiranoj za Newtonov interpolacijski polinom.

Neka je polinom $p(x)$ stupnja K , zapisan u standardnoj bazi, $p(x) = a_0 + a_1x + \dots + a_Kx^K$. Kada se polinom $p(x)$ razvije u Taylorov red oko $x = 0$, dobije se

$$p(x) = p(0) + \frac{p'(0)}{1!}x + \frac{p''(0)}{2!}x^2 + \dots + \frac{p^{(K)}(0)}{K!}x^K,$$

iz čega slijedi da je $a_i = p^{(i)}(0)/i!$, $i = 0, \dots, K$. Ostaje još samo odrediti način na koji će se izračunati derivacije polinoma p .

Neka su b_k koeficijenti prikaza polinoma p u bazi B_N . Kako je riječ o bazi povezanoj s Newtonovim interpolacijskim polinomom, $b_0 = f[x_0]$ i $b_i = f[x_0, \dots, x_i]$, $i = 1, \dots, K$. Za računanje vrijednosti ovako prikazanog polinoma p u nekoj zadanoj točki x koristi se modificirana Hornerova shema:

$$\begin{aligned} s_K &= b_K \\ s_r &= s_{r+1}(x - x_r) + b_r, \quad r = K - 1, \dots, 0. \end{aligned} \quad (2.20)$$

No, ova shema se može iskoristiti i za računanje derivacija. Naime, svaka veličina s_R iz (2.20) se može zapisati i kao polinom

$$\begin{aligned} s_K(x) &= b_K \\ s_r(x) &= s_{r+1}(x - x_r) + b_r, \quad r = K - 1, \dots, 0, \end{aligned} \quad (2.21)$$

i vrijedi $p(x) = s_0(x)$. Ako se relacije iz (2.21) deriviraju k puta, primjenom Leibnitzova pravila za višestruku derivaciju umnoška dobiva se

$$\begin{aligned} s_r^{(k)}(x) &= 0, \quad r = K, \dots, K - k + 1 \\ s_r^{(k)}(x) &= s_{r+1}^{(k)}(x - x_r) + k s_{r+1}^{(k-1)}(x), \quad r = K - k, \dots, 0, \end{aligned} \quad (2.22)$$

i vrijedi $p^{(k)}(x) = s_0^{(k)}(x)$. Na temelju (2.22) napravljen je algoritam kojim se sukcesivno računaju vrijednost polinoma p i njegovih derivacija u zadanoj točki (algoritam 2.5), ako je p prikazan u bazi B_N . Ako se u algoritmu 2.5 korak 5 zamijeni s $b_{i+1} = s_{i+1}$, tada za elemente polja s vrijedi $s_k = g^{(k)}(x)/k!$, odnosno s_k su koeficijenti u prikazu polinoma $p(x)$ u standardnoj bazi.

Algoritam 2.5 Hornerov algoritam za računanje vrijednosti polinoma p i njegovih derivacija u zadanoj točki x ako je polinom p zadan s

$$p(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \dots + b_K(x - x_0) \cdots (x - x_{K-1}).$$

Pretpostavke: Zadani su x – točka u kojoj se traži vrijednost polinoma i njegovih derivacija.

(c_k) – pomoćno polje, čiji elementi su definirani s $c_k = b_k$, $k = 0, \dots, K$.

(s_k) – na izlazu polje čiji elementi su definirani s $s_k = p^{(k)}(x)$, $k = 0, \dots, K$.

- 1: **for** $k=0$ **to** K
 - 2: $s_K = c_K$
 - 3: **for** $i = K - 1$ **to** k **step** -1
 - 4: $s_i = s_{i+1}(x - x_{i-k}) + c_i$.
 - 5: $c_{i+1} = (k + 1) s_{i+1}$.
 - 6: **end for**
 - 7: **end for**
-

Sada kada je poznat način na koji se polinomi $f_{i,x,s}(\lambda)$ i $RSS_{x,s}(\lambda)$ mogu zapisati u standardnoj bazi, ostaje samo odabrati hoće li se $RSS_{x,s}(\lambda)$ prikazati u standardnoj bazi direktnom interpolacijom ili će se prvo naći zapisi polinoma $f_{i,x,s}(\lambda)$ u standardnoj bazi, a onda će se $RSS_{x,s}(\lambda)$ odrediti kao zbroj kvadrata polinoma $f_{i,x,s}(\lambda)$. Zbog numeričkih razloga (pojava grešaka zaokruživanja, nepreciznost zapisa broja u računalu), ako je $\deg RSS_{x,s}$ relativno mali (npr. $\deg RSS_{x,s} \leq 10$), onda se zbog brzine isplati koristiti postupak s direktnom interpolacijom polinoma $RSS_{x,s}(\lambda)$, ali ako je $\deg RSS_{x,s}$ velik, onda je bolje interpolacijom odrediti polinome $f_{i,x,s}(\lambda)$, a koeficijente polinoma $RSS_{x,s}$ odredimo kao zbroj kvadrata polinoma $f_{i,x,s}(\lambda)$ pomoću simboličkog računa.

U programu PolySolver se za određivanje prikaza polinoma $RSS_{x,s}$ u standardnoj bazi koristi postupak u kojem se polinomna interpolacija upotrebljava kad god je to moguće, a kad je nije moguće koristiti, onda se upotrebljava simbolički račun. Postupak je opisan u algoritmu 2.6.

Treba napomenuti da efikasnost algoritma 2.6 ovisi i o strukturi podataka koja se koristi za opis polinomnog sustava, odnosno samih polinoma u više varijabli. Ako se iz strukture podataka koja opisuje konkretan polinom može lako očitati koliki je stupanj polinoma \tilde{d}_i te koji monomi imaju stupanj jednak \tilde{d}_i , onda se računanje uvjeta iz propozicije 2.5, leme 2.6 i leme 2.8 može efikasno provesti.

2.5 Kriteriji zaustavljanja

Za svaki iterativni algoritam potrebno je definirati postupak kojim će se provjeriti jesu li ispunjeni uvjeti za zaustavljanje iterativnog procesa. Iterativni proces treba biti prekinut ako nastupi neka od sljedećih situacija:

- (a) pronađeno je rješenje,
- (b) proces ne ostvaruje napredak,
- (c) nastupila je divergencija procesa te
- (d) napravljen je prevelik broj koraka.

Kriterij kojim se određuje je li točka $\mathbf{x}^{(k)}$ rješenje problema je

$$\|\mathbf{F}(\mathbf{x}^{(k)})\|_\infty < \varepsilon_1 \quad \Leftrightarrow \quad |f_i(\mathbf{x}^{(k)})| < \varepsilon_{\text{sol}}, \quad i = 1, \dots, n. \quad (2.23)$$

Kriterij (2.23) nije idealan ako polinomi f_i koji čini sustav nemaju približno jednaki red veličine u nekoj okolini nultočke. Stoga Dennis i Schabel [11] sugeriraju da bi bilo bolje koristiti skaliranu verziju kriterija (2.23):

$$\|D_F \mathbf{F}(\mathbf{x}^{(k)})\|_\infty < \varepsilon_{\text{sol}},$$

gdje je D_F dijagonalna matrica reda n , čiji su dijagonalni elementi definirani s $(D_F)_{ii} = 1/\text{typ} f_i$,

Algoritam 2.6 Algoritam za određivanje prikaza polinoma $RSS_{\mathbf{x},s}$ u standardnoj bazi prostora polinoma.

Pretpostavke: Polinomni sustav $F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x})) = \mathbf{0}$.

Izračunati stupnjevi polinoma f_i , $\tilde{d}_i = \deg f_i$ i najveći među njima $\tilde{d}^M = \max_i \tilde{d}_i$.

Zadana trenutna točka \mathbf{x} i smjer spusta \mathbf{s} .

Najveći stupanj polinoma za koji se radi interpolacija $G = 10$.

- 1: **if** $\max_i \deg f_{i,\mathbf{x},s} = \tilde{d}^M$ (uvjet iz leme 2.6) **then**
- 2: status=0; $K=2\tilde{d}^M$.
- 3: **else if** $\max_i \deg f_{i,\mathbf{x},s} = \tilde{d}^M - 1$ (uvjet iz leme 2.8) **then**
- 4: status=0; $K=2\tilde{d}^M - 2$.
- 5: **else**
- 6: status=1.
- 7: **end if**
- 8: **if** status=0 **and** $K \leq G$ **then**
- 9: Nađi Newtonov interpolacijski polinom P za $RSS_{\mathbf{x},s}$ u $K + 1$ točaka.
- 10: Pomoću algoritma 2.5 odredi koeficijente od $RSS_{\mathbf{x},s}$ u standardnoj bazi.
- 11: Idi na **Kraj**.
- 12: **else if** status=0 **and** $K > G$ **then**
- 13: **for** $i=1$ **to** n
- 14: **if** $\deg f_{i,\mathbf{x},s} = \tilde{d}_i$ (uvjet iz propozicije 2.5) **then**
- 15: Nađi Newtonov interpolacijski polinom P za $f_{i,\mathbf{x},s}$ u $\tilde{d}_i + 1$ točaka.
- 16: Pomoću algoritma 2.5 odredi koeficijente od $f_{i,\mathbf{x},s}$ u standardnoj bazi.
- 17: **else**
- 18: Upotrebom simboličkog računa odredi koeficijente od $f_{i,\mathbf{x},s}$ u standardnoj bazi.
- 19: **end if**
- 20: **end for**
- 21: **else**
- 22: **for** $i=1$ **to** n
- 23: Upotrebom simboličkog računa odredi koeficijente od $f_{i,\mathbf{x},s}$ u standardnoj bazi.
- 24: **end for**
- 25: **end if**
- 26: Korištenjem izraza za kvadriranje i zbrajanje polinoma odredi koeficijente od $RSS_{\mathbf{x},s}$ u standardnoj bazi.
- 27: **Kraj**.

a tip f_i predstavlja red veličine funkcije f_i u nekoj okolini nultočke. No, određivanja vrijednosti tip f_i nije jednostavno. Stoga se u ovom radu razmatra samo kriterij (2.23), a ako korisnik zna da neki od polinoma koji čine sustav ima značajno veću magnitudu od drugih, uvijek može koeficijente tog polinoma pomnožiti s odgovarajućom vrijednosti.

Najbolji slučaj za iterativni proces je da dođe do rješenja, ali moguća je situacija da proces ne ostvaruje napredak, odnosno da konvergira prema točki koja nije rješenje sustava. Za detekciju tog slučaja koristi se kriterij iz ([11]) koji kaže da iterativni proces treba zaustaviti ako je relativna promjena dvije uzastopne iteracije $\mathbf{x}^{(k-1)}$ i $\mathbf{x}^{(k)}$ manja od zadane tolerancije $\varepsilon_{\text{step}}$. No,

standardni izraz za relativnu promjenu

$$\frac{|\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}|}{|\mathbf{x}_i^{(k)}|} < \delta_{\text{step}}, \quad i = 1, \dots, n,$$

nije pogodan u slučaju kada neka od komponenti konvergira prema 0, pa se zato kao kriterij koristi izraz

$$\frac{|\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}|}{\max\{|\mathbf{x}_i^{(k)}|, \delta_{\text{rel}}\}} < \delta_{\text{step}}, \quad i = 1, \dots, n. \quad (2.24)$$

Veličina δ_{rel} u $\max\{|\mathbf{x}_i^{(k)}|, \delta_{\text{rel}}\}$ je neki mali broj, npr. 10^{-2} ili 10^{-3} . Ovakav odabir δ_{rel} ima svojih prednosti i nedostataka. Tipično, ako se iterativni proces želi zaustaviti kada se sve komponente dvije uzastopne točke iterativnog procesa poklapaju u p decimala, treba staviti $\delta_{\text{step}} = 10^{-p}$. No ako neka od komponenti teži prema nuli, proces će se zaustaviti kada je apsolutna vrijednost razlike dvije uzastopne komponente reda veličine $\delta_{\text{rel}} \cdot 10^{-p}$ i u ovom slučaju bi konstanta δ_{rel} trebala biti što manja da bi se detektirala koja je točna značajna znamenka. S druge strane, situacija u kojoj iterativni proces ne ostvaruje napredak će se prije detektirati u slučaju kada je δ_{rel} veći.

U općem slučaju, ako se ne zna koliko velika može biti komponenta rješenja, može se dogoditi da se iterativni postupak zaustavi prerano. Takav slučaj nastupa kada je neka komponenta rješenja jako mala, ali nije nula. Stoga je potrebno poznavati problem i znati mogu li takva rješenja nastupiti te po potrebi napraviti skaliranje varijabli. Na primjer, neka je zadan sustav jednadžbi

$$x_1^3 + x_2^2 - 1,00000004 = 0$$

$$x_1 + x_2 - 1,0002 = 0$$

Jedno rješenje tog sustava je točka $x_1 = 1$, $x_2 = 0,0002$. Ako je poznato da je rješenje x_2 reda veličine 10^{-4} , može se napraviti skaliranje, odnosno zamjena $x_2 = 10^{-4} \cdot y_2$ čime početni problem prelazi u

$$x_1^3 + 10^{-8} \cdot y_2^2 - 1,00000004 = 0$$

$$x_1 + 0,0001y_2 - 1,0002 = 0$$

čije rješenje je $x_1 = 1$ i $y_2 = 2$.

Treći slučaj u kojem iterativni proces treba prekinuti je slučaj divergencije. Ovaj slučaj se može prepoznati po tome što se između dvije uzastopne iteracije vrijednost pripadajuće funkcije sustava (tipično $\|\mathbf{F}(\mathbf{x})\|_2$) promijenila jako malo, ali je udaljenost između iteracija relativno

velika. Stoga je kriterij za prepoznavanje ovog slučaja

$$1 - \frac{\|\mathbf{F}(\mathbf{x}^{(k)})\|_2}{\|\mathbf{F}(\mathbf{x}^{(k-1)})\|_2} < \varepsilon_{\text{div}} \quad \text{i} \quad \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 > \delta_{\text{tol}}. \quad (2.25)$$

U prvom dijelu uvjeta se ispituje je li došlo do dovoljno velike promjene u odgovarajućoj normi i to tako da se pogleda je li relativna promjena norme manja od ε_{div} , dok drugi dio uvjeta provjerava je li udaljenost dvije uzastopne iteracije dovoljno velika. Svi numerički eksperimenti u ovom radi su napravljeni uz $\varepsilon_{\text{div}} = 10^{-6}$ te $\delta_{\text{tol}} = 10^{-2}$.

Konačno, posljednji slučaj koji treba pogledati je da li je iterativni postupak napravio previše koraka, a nije donio odluku. Ovaj slučaj je jednostavan za provjeru, dovoljno je pogledati koliko je iteracija do sada napravljeno te je li taj broj veći od vrijednosti *maxiter*. U skladu s [11], za rješavanje polinomnog sustava reda n koristi se $\text{maxiter} = (n + 1) \cdot 100$.

Napomena 2.1 U programu *PolySolver*, korištenom za provođenje numeričkih eksperimenata vezanih uz metode najdublje spusta, koriste se sljedeće vrijednosti parametara ε_{sol} , δ_{step} , δ_{rel} , ε_{div} , δ_{tol} i *maxiter*:

$$\varepsilon_{\text{sol}} = 10^{-8},$$

$$\delta_{\text{step}} = 10^{-4},$$

$$\delta_{\text{rel}} = 10^{-3},$$

$$\varepsilon_{\text{div}} = 10^{-6},$$

$$\delta_{\text{tol}} = 10^{-2},$$

$$\text{maxiter} = (n + 1) \cdot 100.$$

2.6 Minimizacija u max–normi

U dosadašnjim poglavljima se kao prateća funkcija polinomnog sustava koristila funkcija

$$\text{RSS}(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2 = \sum_{i=1}^n f_i^2(\mathbf{x}),$$

odnosno njezina restrikcija na pravac određen točkom \mathbf{x} i smjerom \mathbf{s} , $\text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda) = \text{RSS}(\mathbf{x} + \lambda\mathbf{s})$.

Primjena funkcije RSS je prirodna: riječ je o diferencijabilnoj funkciji koju je relativno jednostavno izračunati. Jedini razlog zbog kojeg bi imalo smisla razmatrati neku drugu funkciju je činjenica da se prilikom određivanja duljine koraka u zadanom smjeru iz neke točke pojavljuje potreba za traženjem nultočaka polinoma koji mogu biti visokog stupnja (lema 2.6). Stoga ima smisla razmišljati o tome da se neka druga funkcija upotrijebi kao prateća funkcija polinomnog

sustava.

U ovom poglavlju razmatra se upotreba funkcije MRN^* kao prateće funkcije sustava. Funkcija MRN predstavlja max–normu reziduala i definirana je s

$$\text{MRN}(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_\infty = \max_{i \in \{1, \dots, n\}} |f_i(\mathbf{x})|.$$

Kao i kod funkcije RSS, najvažnija primjena funkcije MRN je da se pomoću nje odredi duljina koraka λ^* za koji će se iterativni postupak pomaknuti iz trenutne točke \mathbf{x} u zadanom smjeru \mathbf{s} . U tu svrhu se promatra restrikcija MRN na pravac koji prolazi kroz \mathbf{x} i ima vektor smjera \mathbf{s} , $\text{MRN}_{\mathbf{x}, \mathbf{s}}$, definirana s

$$\text{MRN}_{\mathbf{x}, \mathbf{s}}(\lambda) = \text{MRN}(\mathbf{x} + \lambda \mathbf{s}) = \max_{i \in \{1, \dots, n\}} |f_i(\mathbf{x} + \lambda \mathbf{s})|.$$

Duljina koraka spusta λ^* je točka u kojoj funkcija $\text{MRN}_{\mathbf{x}, \mathbf{s}}$ postiže najmanju vrijednost,

$$\lambda^* = \arg \min_{\lambda \in \mathbb{R}} \text{MRN}_{\mathbf{x}, \mathbf{s}}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} \max_{i \in \{1, \dots, n\}} |f_i(\mathbf{x} + \lambda \mathbf{s})|.$$

Funkcija $\text{MRN}_{\mathbf{x}, \mathbf{s}}$ nije ni diferencijabilna, niti ju je lako izračunati, ali ima prednost da se u njezinoj primjeni pojavljuju polinomi nižeg stupnja s kojima je lakše raditi.

No, da bi se uopće mogla koristiti funkcija $\text{MRN}_{\mathbf{x}, \mathbf{s}}$, potrebno je konstruirati algoritam koji će omogućiti njezino izračunavanje. Funkciju $\text{MRN}_{\mathbf{x}, \mathbf{s}}$ je lako izračunati u nekoj određenoj točki, ali ako se želi naći egzaktna točka u kojoj se postiže najmanja vrijednost, potrebno je naći formule za izračun vrijednosti funkcije $\text{MRN}_{\mathbf{x}, \mathbf{s}}$.

Radi jednostavnosti, uvodi se pojam max–funkcija. Max–funkcija je funkcija čija vrijednost u nekoj točki je definirana kao najveća od apsolutnih vrijednosti dvije ili više funkcija. Max–norma reziduala je primjer max–funkcije.

Ostatak potpoglavlja je podijeljen na četiri dijela. U prvom dijelu se promatraju funkcije definirane po dijelovima na primjeru apsolutne vrijednosti funkcije. Zatim se pokazuje kako konstruirati max–funkciju za dva, odnosno više polinoma, a na kraju se opisuje algoritam za rješavanje polinomnih sustava u kojima se koristi funkcija MRN .

2.6.1 Funkcije zadane po dijelovima

Neka je zadana funkcija $\mathbf{F} : \mathbb{R} \rightarrow \mathbb{R}^n$, $\mathbf{F}(t) = (f_1(t), \dots, f_n(t))$, $f_1, \dots, f_n : \mathbb{R} \rightarrow \mathbb{R}$ realni polinomi. Max–norma funkcije \mathbf{F} definirana je s

$$\mathbf{F}_M(t) = \|\mathbf{F}(t)\|_\infty = \max_{i \in \{1, \dots, n\}} |f_i(t)|.$$

*MRN - Maximum Residual Norm.

Iz definicije funkcije F_M može se naslutiti da ona neće biti definirana jednom formulom na cijelom \mathbb{R} , već će biti definirana po dijelovima.

Definicija 2.9 Neka je $-\infty < a_1 < \dots < a_{k-1} < \infty$ niz realnih brojeva, koji definiraju k intervala I_1, I_2, \dots, I_k , $I_1 = \langle -\infty, a_1 \rangle$, $I_i = \langle a_{i-1}, a_i \rangle$, $i = 2, \dots, k-1$, $I_k = \langle a_{k-1}, \infty \rangle$.

Za funkciju $g : \mathbb{R} \rightarrow \mathbb{R}$ kaže se da je definirana po dijelovima ako postoje funkcije g_1, \dots, g_k , $g_i : I_i \rightarrow \mathbb{R}$, $i = 1, \dots, k$, takve da vrijedi $g(t) = g_i(t)$ za $t \in I_i$.

Iz definicije 2.9 slijedi da je funkcija definirana po dijelovima definirana na cijelom skupu realnih brojeva. Također, u definiciji funkcije definirane po dijelovima nema nikakvih zahtjeva na neprekidnost funkcije ili na predznak funkcije.

Lema 2.10 Neka je p polinom stupnja n . Tada vrijedi:

- (a) Funkcija $|p|$ definirana je po dijelovima.
- (b) Za broj dijelova k vrijedi $1 \leq k \leq n+1$.
- (c) Rubovi intervala a_1, \dots, a_{k-1} su realne nultočke polinoma p .
- (d) Za $t \in I_i$ je $|p(t)| = p(t)$ ili $|p(t)| = -p(t)$.

Dokaz: Ako polinom p nema realnih nultočaka, onda je za $t \in \mathbb{R}$ ili $|p(t)| = p(t)$ ili $|p(t)| = -p(t)$. Funkcija $p(t)$ je definirana na jednom intervalu $I_1 = \langle -\infty, \infty \rangle = \mathbb{R}$ i time su dokazane sve tvrdnje leme.

Ako polinom p ima realne nultočke, ali su sve realne nultočke parnog stupnja, onda p ne mijenja predznak pa za funkciju $|p|$ vrijedi sve isto kao i u slučaju da p nema realnih nultočaka.

Konačno, neka polinom p ima realne nultočke neparnog stupnja, $a_1 < \dots < a_k$. Jer je riječ o nultočama neparnog stupnja, funkcija p mijenja predznak u okolini nultočke a_i , odnosno postoji $\delta > 0$ takav da vrijedi

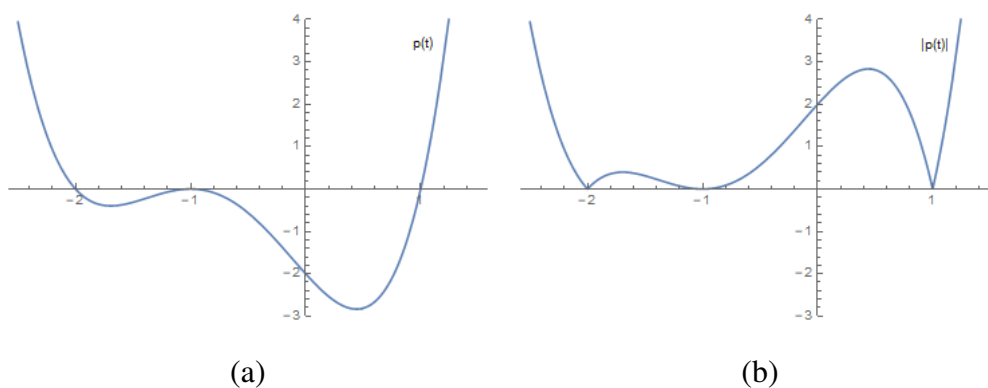
$$p(a_i - t) \cdot p(a_i + t) < 0 \quad \text{za } t \in \langle 0, \delta \rangle.$$

Na intervalu $I_1 = \langle -\infty, a_1 \rangle$ polinom p nema realnih nultočaka jer je a_1 najmanja realna nultočka, a kako su polinomi neprekidne funkcije slijedi da p na intervalu I_1 ne mijenja predznak. Stoga je restrikcija funkcije $|p|$ na intervalu I_1 jednaka ili p ili $-p$.

Iz istih razloga, na intervalu $I_k = \langle a_{k-1}, \infty \rangle$ polinom p ne mijenja predznak jer je a_k najveća realna nultočka. Stoga je restrikcija funkcije $|p|$ na intervalu I_k jednaka ili p ili $-p$.

Konačno, na intervalima I_i , $i = 2, \dots, k-1$, polinom p je stalnog predznaka zbog toga jer je neprekidna funkcija, a između a_{i-1} i a_i ne postoji druga realna nultočka. Zato je i na intervalima I_i , $i = 2, \dots, k-1$, restrikcija funkcije $|p|$ jednaka ili p ili $-p$. ■

Primjena leme 2.10 može se bolje razumjeti na sljedećem primjeru.



Slika 2.1: Grafovi funkcija p (slika (a)) i $|p|$ (slika (b)) iz primjera 2.11.

Primjer 2.11 Neka je zadan polinom $p(t) = t^4 + 3t^3 + t^2 - 3t - 2$. Želimo naći zapis funkcije $|p(t)|$.

- Nultočke polinoma p su redom -2 , -1 (dvostruka nultočka) i 1 .
- Za $t \in \langle -\infty, -2 \rangle$ je $p(t) > 0$ pa je $|p(t)| = p(t)$.
- Kako je nultočka -1 nultočka parnog stupnja, predznak funkcije p na intervalu $\langle -2, -1 \rangle$ je isti kao i na $\langle -1, 1 \rangle$, pa je $|p(t)| = -p(t)$ na $\langle -2, 1 \rangle$.
- Konačno za $t \in \langle 1, \infty \rangle$ je $p(t) > 0$ pa je $|p(t)| = p(t)$.
- U oznakama leme 2.10, funkcija $|p|$ je definirana na $k = 3$ intervala: $I_1 = \langle -\infty, -2 \rangle$, $I_2 = \langle -2, 1 \rangle$ i $I_3 = \langle 1, \infty \rangle$. Vrijedi

$$|p(t)| = \begin{cases} p(t), & t \in I_1; \\ -p(t), & t \in I_2; \\ p(t), & t \in I_3. \end{cases}$$

- Na slici 2.1 prikazani su grafovi funkcija $p(t)$ i $|p(t)|$.

2.6.2 Max-funkcija za dva polinoma

Nakon što je uveden pojam funkcije po dijelovima i pokazano kako se apsolutna vrijednost polinoma može prikazati kao funkcija po dijelovima, idući korak je odrediti postupak kojim će se dati egzaktan prikaz max-funkcije za dva polinoma. Preciznije, za zadane polinome p i r , max-funkcija g_M je definirana s $g_M(t) = \max\{|p(t)|, |r(t)|\}$. Kako su funkcije $|p|$ i $|r|$ zadane po dijelovima, za očekivati je da je i g_M također zadana po dijelovima, ali to treba dokazati.

Sljedeća lema ne vrijedi samo za polinome, a koristit će se za određivanje granica intervala koji određuju dijelove.

Lema 2.12 Neka su p i r polinomi. Točka t^* je rješenje jednadžbe $|p(t^*)| = |r(t^*)|$ ako i samo ako je t^* nultočka ili polinoma $p + r$ ili polinoma $p - r$.

Dokaz: Neka je $|p(t^*)| = |r(t^*)|$. Tada su moguća četiri slučaja:

1. $p(t^*) = r(t^*) \Leftrightarrow (p - r)(t^*) = 0$,
2. $p(t^*) = -r(t^*) \Leftrightarrow (p + r)(t^*) = 0$,
3. $-p(t^*) = r(t^*) \Leftrightarrow (-p - r)(t^*) = 0 \Leftrightarrow (p + r)(t^*) = 0$,
4. $-p(t^*) = -r(t^*) \Leftrightarrow (-p + r)(t^*) = 0 \Leftrightarrow (p - r)(t^*) = 0$,

čime je tvrdnja leme dokazana.

Sada smo u mogućnosti dokazati da je g_M po dijelovima zadana funkcija.

Lema 2.13 *Neka su zadani polinomi p i r .*

- (a) *Tada je $g_M(t) = \max\{|p(t)|, |r(t)|\}$ po dijelovima zadana funkcija na intervalima I_1, I_2, \dots, I_k ,
 $I_1 = \langle -\infty, a_1 \rangle$, $I_j = \langle a_{j-1}, a_j \rangle$, $j = 2, \dots, k-1$, $I_k = \langle a_{k-1}, \infty \rangle$.*
- (b) *Rubovi intervala I_j , a_1, \dots, a_{k-1} su nultočke ili polinoma $p + r$ ili polinoma $p - r$.*
- (c) *Na intervalu I_j funkcija g_M definirana je samo jednom formulom, tj. za sve $t \in I_j$ je ili
 $g_M(t) = |p(t)|$ ili $g_M(t) = |r(t)|$.*

Dokaz: Neka su $a_1 < a_2 < \dots < a_{k-1}$ sve nultočke polinoma $p + r$ i $p - r$ poredane po veličini. Prema lemi 2.12, točke a_1, \dots, a_{k-1} su i sva rješenja jednadžbe $|p(t)| = |r(t)|$, odnosno sve nultočke funkcije $g(t) = |p(t)| - |r(t)|$. Funkcija g je neprekidna funkcija.

Sve tvrdnje leme se mogu dokazati ako se pokaže da na intervalu I_j za sve $t \in \langle a_{j-1}, a_j \rangle$ vrijedi ili

$$\max\{|p(t)|, |r(t)|\} = |p(t)| \Leftrightarrow |p(t)| > |r(t)|,$$

ili

$$\max\{|p(t)|, |r(t)|\} = |r(t)| \Leftrightarrow |r(t)| > |p(t)|.$$

Pretpostavimo suprotno: neka su $t_1, t_2 \in I_j$ za koje vrijedi $|p(t_1)| > |r(t_1)|$ i $|p(t_2)| < |r(t_2)|$. Zbog neprekidnosti funkcije g , tada postoji $t^* \in \langle t_1, t_2 \rangle$ takav da je $g(t^*) = 0$, što je kontradikcija s pretpostavkom da su $a_1 < \dots < a_{k-1}$ sve nultočke od g .

Ovim je dokazano da je g_M po dijelovima zadana funkcija. ■

Da je g_M neprekidna funkcija, jasno je već iz leme 2.13, zato što je njena restrikcija na bilo koji interval neprekidna funkcija, a u rubovima intervala a_1, \dots, a_{k-1} vrijedi $|p(a_i)| = |r(a_i)|$, tako da je sigurno i u njima neprekidna. No, g_M je isto tako neprekidna funkcija jer je maksimum dvije i više neprekidnih funkcija isto tako neprekidna funkcija, što je dokazano u sljedećoj lemi.

Lema 2.14 *Neka su $f_1, \dots, f_n : \mathbb{R} \rightarrow \mathbb{R}$ neprekidne funkcije. Tada je i funkcija*

$$f_M = \max\{f_1, \dots, f_n\}$$

neprekidna funkcija.

Dokaz: Dokaz je indukcijom po funkcijama f_i . Prvo se pokaže da je funkcija $f_{M2} = \max\{f_1, f_2\}$ neprekidna funkcija. Pretpostavimo suprotno, neka postoji točka x_0 u kojoj je funkcija f_{M2} nije neprekidna.

- i. Ako je $f_1(x_0) > f_2(x_0)$, tada postoji $\delta > 0$ i interval $I_\delta = \langle x_0 - \delta, x_0 + \delta \rangle$ takav da za $x \in I_\delta$ vrijedi $f_1(x) > f_2(x)$, iz čega slijedi da je $f_{M2}(x) = f_1(x)$ za $x \in I_\delta$. Ali, jer je f_1 neprekidno u x_0 , onda je i f_{M2} neprekidno u x_0 .
- ii. Analogno, ako je $f_2(x_0) > f_1(x_0)$, onda postoji I_δ takav da je $f_{M2}(x) = f_2(x)$ za $x \in I_\delta$, pa neprekidnost od f_{M2} u x_0 slijedi iz činjenice da je f_2 neprekidna u x_0 .
- iii. Ako je $f_1(x_0) = f_2(x_0)$, tada mogu nastupiti tri slučaja:
 - iii.a Postoji I_δ takav da je $f_1(x) \geq f_2(x)$ za $x \in I_\delta$. Da je f_{M2} neprekidno dokazuje se na isti način kao u slučaju (i).
 - iii.b Postoji I_δ takav da je $f_2(x) \geq f_1(x)$ za $x \in I_\delta$. Da je f_{M2} neprekidno dokazuje se na isti način kao u slučaju (ii).
 - iii.c Postoje $I_\delta^- = \langle x_0 - \delta, x_0 \rangle$ i $I_\delta^+ = \langle x_0, x_0 + \delta \rangle$ takvi da je $f_1(x) \geq f_2(x)$ za $x \in I_\delta^-$ i $f_2(x) \geq f_1(x)$ za $x \in I_\delta^+$. Tada je $f_{M2}(x) = f_1(x)$ za $x \in I_\delta^-$ i $f_{M2}(x) = f_2(x)$ za $x \in I_\delta^+$. No zbog neprekidnosti od f_1 i f_2 vrijedi

$$\lim_{x \rightarrow x_0^-} f_{M2}(x) = \lim_{x \rightarrow x_0^-} f_1(x) = f_1(x_0) = f_2(x_0) = \lim_{x \rightarrow x_0^+} f_2(x) = \lim_{x \rightarrow x_0^+} f_{M2}(x),$$

iz čega slijedi da je f_{M2} neprekidno u x_0 .

- iii.d Slučaj kada je $f_2(x) \geq f_1(x)$ za $x \in I_\delta^-$ i $f_1(x) \geq f_2(x)$ za $x \in I_\delta^+$ se dokazuje isto kao slučaj iii.c.

Ovime je dokazano da je maksimum dvije neprekidne funkcija na \mathbb{R} neprekidna funkcija.

Pretpostavimo da tvrdnja vrijedi za $k = n - 1$ ($\max\{f_1, \dots, f_{n-1}\}$ je neprekidna funkcija) i dokažimo da vrijedi za $k = n$. Funkciju $f_M = \max\{f_1, \dots, f_n\}$ možemo zapisati kao

$$f_M = \max\{f_1, \dots, f_n\} = \max\{\underbrace{\max\{f_1, \dots, f_{n-1}\}}_{\text{funkcija } g}, f_n\}$$

Po pretpostavci indukcije, funkcija $g = \max\{f_1, \dots, f_{n-1}\}$ je neprekidna pa je $f_M = \max\{g, f_n\}$ maksimum dvije neprekidne funkcije. Kako je u prvom dijelu dokaza dokazano da je maksimum dvije neprekidne funkcije neprekidna funkcija, slijedi da je i f_M neprekidna funkcija, što je i trebalo dokazati. ■

Prema lemi 2.14, funkcija g_M je neprekidna jer su polinomi neprekidne funkcije, a maksimum dvije neprekidne funkcije je također neprekidna funkcija. No, funkcija g_M nije samo neprekidna, za nju vrijedi i da je neprekidno diferencijabilna na intervalima $\langle a_{j-1}, a_j \rangle$.

Propozicija 2.15 Funkcija g_M je diferencijabilna na intervalima $\langle a_{j-1}, a_j \rangle$, $j = 1, \dots, k$.

Štoviše, restrikcija funkcije g_M na interval $\langle a_{j-1}, a_j \rangle$ je točno jedna od četiri moguće funkcije: p , $-p$, r i $-r$.

Dokaz: Iz definicije funkcije g_M te iz leme 2.13 slijedi da u otvorenim intervalima $\langle a_{j-1}, a_j \rangle$ ne mogu postojati rješenja jednadžbe $|p(t)| - |r(t)| = 0$ jer bi to bila kontradikcija s činjenicom da su točke a_1, \dots, a_{k-1} sva rješenja te jednadžbe.

Bez smanjenja općenitosti može se pretpostaviti da na intervalu $\langle a_{j-1}, a_j \rangle$ vrijedi $g_M(t) = |p(t)|$. Treba pokazati da je za sve t iz tog intervala ili $g_M(t) = p(t)$ ili $g_M(t) = -p(t)$.

Pretpostavimo suprotno: neka su $t_1, t_2 \in \int I_j$ takvi da je $|p(t_1)| = p(t_1)$ ($p(t_1) > 0$) i $|p(t_2)| = -p(t_2)$ ($p(t_2) < 0$). Tada zbog neprekidnosti polinoma p , treba postojati točka $t^* \in \langle t_1, t_2 \rangle$, takva da je $p(t^*) = 0$. Ali, jer je $g_M(t^*) = |p(t^*)| = 0$, slijedi da je i $r(t^*) = 0$, odnosno da je t^* jedno od rješenja jednadžbe $|p(t)| - |r(t)| = 0$, što je kontradikcija s činjenicom da na intervalu $\langle a_{j-1}, a_j \rangle$ ne postoje rješenja jednadžbe $|p(t)| - |r(t)| = 0$.

Dakle, za sve $t \in \langle a_{j-1}, a_j \rangle$ vrijedi ili $g_M(t) = p(t)$ ili $g_M(t) = -p(t)$, a jer je p neprekidno diferencijabilna funkcija, onda je i g_M neprekidno diferencijabilna funkcija na intervalu $\langle a_{j-1}, a_j \rangle$. ■

Činjenica da je g_M neprekidna funkcija te da je po dijelovima diferencijabilna omogućuje jednostavno definiranje postupka za određivanje točke t_{min} u kojoj g_M postiže najmanju vrijednost,

$$t_{min} = \arg \min_{t \in \mathbb{R}} g_M(t) = \arg \min_{t \in \mathbb{R}} \max\{|p(t)|, |r(t)|\}.$$

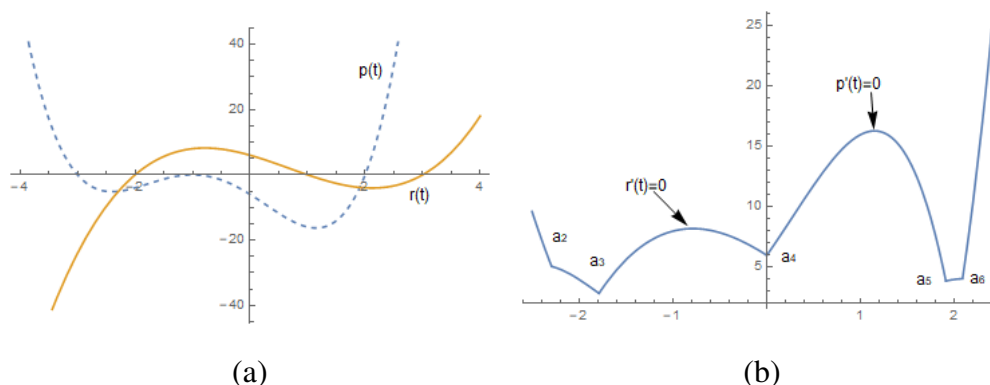
Ideja je sljedeća: obzirom da su p i r polinomi, vrijedi da je

$$\lim_{t \rightarrow \pm\infty} g_M(t) = \lim_{t \rightarrow \pm\infty} \max\{|p(t)|, |r(t)|\} = \infty.$$

Na intervalu $[a_{j-1}, a_j]$, $j = 2, \dots, k-1$, funkcija g_M zbog neprekidnosti dostiže minimum ili u rubovima intervala (točkama a_{j-1} i a_j) ili u nekom lokalnom ekstremu. Kako je prema propoziciji 2.15 restrikcija g_M na interval $\langle a_{j-1}, a_j \rangle$ neka od funkcija p , $-p$, r ili $-r$, onda je lokalni ekstrem od g_M lokalni ekstrem funkcije p ili r , ovisno o tome s kojom je funkcijom g_M definiran na intervalu $\langle a_{j-1}, a_j \rangle$. Isto je i na intervalima $I_1 = \langle -\infty, a_1 \rangle$ i $I_k = \langle a_{k-1}, \infty \rangle$, samo što se tu gleda vrijednost funkcije u rubnoj točki i eventualnom lokalnom ekstremu.

Propozicija 2.16 Neka je zadana funkcija $g_M = \max\{|p|, |r|\}$. Neka su a_1, \dots, a_{k-1} nultočke polinoma $p+r$ i $p-r$. Nadalje, neka su $s t_1^p, \dots, t_{n_p}^p$ označene sve stacionarne točke polinoma p (rješenja jednadžbe $p'(t) = 0$), a s $t_1^r, \dots, t_{n_r}^r$ sve stacionarne točke polinoma r (rješenja jednadžbe $r'(t) = 0$). Tada je

$$\min_{t \in \mathbb{R}} g_M(t) = \min\{g_M(a_1), g_M(a_{k-1}), g_M(t_1^p), \dots, g_M(t_{n_p}^p), g_M(t_1^r), \dots, g_M(t_{n_r}^r)\}. \quad (2.26)$$



Slika 2.2: Grafovi polinoma p i r (slika (a)) i funkcije $g_M = \max\{|p|, |r|\}$ (slika (b)) iz primjera 2.17.

Dokaz: Formalno,

$$\min_{t \in \mathbb{R}} g_M(t) = \min_{j \in \{1, \dots, k\}} \min_{t \in I_j} g_M(t). \quad (2.27)$$

Na unutarnjem intervalu I_j , $j = 2, \dots, k-1$ vrijedi da je

$$\min_{t \in I_j} g_M(t) = \min\{g_M(a_{j-1}), g_M(a_j), g_M(t_{j_1}), \dots, g_M(t_{j_s})\}, \quad (2.28)$$

gdje su točke t_{j_1}, \dots, t_{j_s} stacionarne točke polinoma p i r koje se nalaze u intervalu I_j . Na vanjskim intervalima I_1 i I_k stvar je ista, samo što se gledaju vrijednosti $g_M(a_1)$ i $g_M(a_{k-1})$ i one stacionarne točke polinoma p i r koje se nalaze u intervalima I_1 i I_k .

Nakon što se (2.28) uvrsti u (2.27) slijedi (2.26), što je i trebalo dokazati. ■

Propozicija 2.16 daje način na koji se može izračunati minimum funkcije g_M : potrebno je naći najmanju od vrijednosti koje funkcija g_M postiže u nultočkama polinoma $p+r$ i $p-r$ te u stacionarnim točkama polinoma p i r , te zapamtiti u kojoj se točki ta vrijednost postiže.

U sljedećem primjeru dana je ilustracija ovih rezultata.

Primjer 2.17 Neka su zadani polinomi $p(t) = t^4 + 3t^3 - 3t^2 - 11t - 6$ i $r(t) = t^3 - 2t^2 - 5t + 6$ (slika 2.2.a). Realne nultočke polinoma $p+r$ su $-4,297$, $-1,787$, 0 i $2,084$. Realne nultočke polinoma $p-r$ su $-2,291$ i $1,907$. Kada ih se poreda dobije se $a_1 = -4,297$, $a_2 = -2,291$, $a_3 = -1,787$, $a_4 = 0$, $a_5 = 1,907$ i $a_6 = 2,084$. Definicija funkcije g_M po intervalima je

Interval	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$g_M(t) =$	$p(t)$	$-r(t)$	$-p(t)$	$r(t)$	$-p(t)$	$-r(t)$	$p(t)$

Nadalje, stacionarne točke polinoma p su $-2,397 \in I_2$, $-1 \in I_4$ i $1,147 \in I_5$, a polinoma r su $-0,786 \in I_4$ i $2,119 \in I_7$.

Na slici 2.2.b prikazan je graf funkcije g_M . Naglašeni su rubovi intervala po kojima je g_M definirana. Bitno je primijetiti da g_M ima samo dva lokalna ekstrema dobivena deriviranjem, a ostali su rubovi intervala. Na intervalu I_4 je $g_M(t) = r(t)$, a $t = -0,786$ je lokalni ekstrem

polinoma r . Na intervalu I_5 je $g_M(t) = -p(t)$, a $t = 1,147$ je lokalni ekstrem polinoma p . Ostale stacionarne točke polinoma p i r nisu lokalni ekstremi g_M jer se nalaze u intervalima u kojima je g_M definiran drugom funkcijom ($-2,397$ je stacionarna točka od p , ali u intervalu I_2 je $g_M(t) = -r(t)$, analogno za -1 i za $2,119$).

Najmanja vrijednost funkcije g_M se postiže u točki $a_3 = -1,787$.

Do sada iznesen sadržaj se bazirao na činjenici da postoje realne nultočke polinoma $p + r$ i $p - r$ koji definiraju intervale na kojima je funkcija g_M zadana. Ako ne postoje nultočke tih polinoma onda je ili $|p(t)| > |r(t)|$ ili $|r(t)| > |p(t)|$ za sve $t \in \mathbb{R}$, iz čega slijedi $g_M = |p|$ ili $g_M = |r|$. Jer ne postoje nultočke polinoma $p + r$ i $p - r$, onda ni polinomi p i r nemaju realnih nultočaka pa je $g_M = \pm p$ ili $g_M = \pm r$, ovisno o tome koji od polinoma ima veću apsolutnu vrijednost u nekoj točki te kakav je taj polinom po predznaku. Uglavnom, tvrdnje leme 2.13 i propozicija 2.15 i 2.16 vrijede i za ovaj slučaj, samo što je g_M zadana na jednom intervalu (cijelom skupu \mathbb{R}).

2.6.3 Max-funkcija za više polinoma

U poglavlju 2.6.2 je objašnjen postupak za formiranje max-funkcije za dva polinoma. U ovom poglavlju se objašnjava na koji način definirati max-funkciju za slučaj tri i više polinoma. Postupak je induktivan, prvo se formira max-funkcija s dva polinoma, a onda se ostali polinomi dodaju jedan po jedan.

Neka su p_1, \dots, p_s polinomi te neka je za njih definirana max-funkcija $g_M = \max\{|p_1|, \dots, |p_s|\}$. Domena funkcija g_M je cijeli skup realnih brojeva, a zadana je po dijelovima na k intervala I_1, \dots, I_k . Restrikcija funkcije g_M na neki interval I_j je točno jedna od funkcija $|p_1|, \dots, |p_s|$.

Pretpostavimo da je zadan polinom r i da želimo odrediti max-funkciju $g_M^N = \max\{g_M, |r|\} = \max\{|p_1|, \dots, |p_s|, |r|\}$. Da bi to mogli, potrebno je odrediti intervale I_1^N, \dots, I_m^N na kojima je g_M^N definirana te koja funkcija reprezentira g_M^N na kojem intervalu.

Postupak kojim se g_M^N određuje iz g_M u grubim crtama je opisan sljedećim koracima.

K0 Neka su inicijalno intervale na kojima je definirana funkcija g_M^N jednaki intervalima na kojima je definirana funkcija g_M , $I_j^N = I_j$, $j = 1, \dots, k$. Također, inicijalno je $g_M^N(t) = g_M(t)$.

K1 Neka je $p \in \{p_1, \dots, p_s\}$.

K2 Treba odrediti nultočke polinoma $p + r$ i $p - r$. Neka su te nultočke označene s $b_1 < \dots < b_n$.

K3 Sada treba proći kroz sve intervale I_j^N , $j = 1, \dots, k$. Intervali na kojima je g_M zadan s nekim polinomom različitim od p se ignoriraju, a treba gledati samo one intervale ne kojima je g_M zadan polinomom p . Sumarno, razlikuju se tri slučaja:

(a) Ako na intervalu I_j^N funkcija g_M^N nije zadana s $|p|$, ne radi ništa.

- (b) Ako je na intervalu I_j^N funkcija g_M^N zadana s $|p|$, ali niti jedna od nultočaka b_1, \dots, b_n nije element intervala I_j^N , tada za sve $t \in I_j^N$ vrijedi ili $|p(t)| > |r(t)|$ ili $|r(t)| > |p(t)|$. Ako je $|p(t)| > |r(t)|$, ništa se ne mijenja, a u suprotnom proglasi da je na intervalu I_j^N funkcija g_M^N zadana s $|r(t)|$ jer je $|r|$ veće od do tada najveće funkcije $|p|$.
- (c) Neka je restrikcija od g_M na intervalu $I_j^n = \langle a_{j-1}, a_j \rangle$ jednaka $|p|$ i neka su $b_1^*, \dots, b_r^* \in I_j$ one nultočke polinoma $p + r$ i $p - r$ koje se nalaze u intervalu I_j^n . ($b_1^*, \dots, b_r^* \in \{b_1, \dots, b_n\}$). Interval I_j^N se sada podijeli na $r + 1$ podintervala $I_{j,1}^N = \langle b_0^*, b_1^* \rangle, \dots, I_{j,r+1}^N = \langle b_r^*, b_{r+1}^* \rangle$ (uz dodatne definicije $b_0^* = a_{j-1}$ i $b_{r+1}^* = a_j$), a na svakom podintervalu $I_{j,l}^N$ je funkcija $|p| - |r|$ stalnog predznaka. Ako za svaki $t \in I_{j,l}^N$ vrijedi $|p(t)| > |r(t)|$ onda je $g_M^N(t) = |p(t)|$, jer je polinom p na intervalu $I_{j,l}^N$ već bio veći od preostalih polinoma iz skupa $\{p_1, \dots, p_s\}$, a veći je i od r . S druge strane, ako za svaki $t \in I_{j,l}^N$ vrijedi $|r(t)| > |p(t)|$, onda je $g_M^N(t) = |r(t)|$, jer je r veći od p , a p je na intervalu $I_{j,l}^N$ bio najveći.

Na kraju ovog koraka jedine izmjene u odnosu na početno stanje su nastale u onim intervalima I_j^n u kojima je g_M^n bio zadan s p . Ostali intervali su ostali nepromijenjeni.

K4 Uzmi sljedeći p i ponovi postupak (idi na K2).

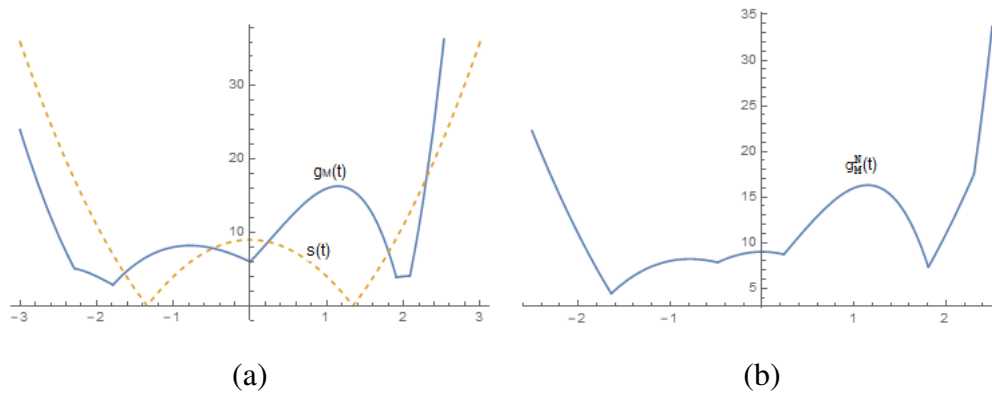
K5 Prođi kroz sve intervale na kojima je definirana funkcija g_M^N . Ako je na dva ili više uzastopnih intervala g_M^N definirana istom funkcijom, treba napraviti novi interval koji je unija takvih intervala. Na ovaj način su definirani intervali I_j^N .

Ono što je zgodno istaknuti je da koraci K2 te K3.b i K3.c nisu ništa drugo nego opis postupka za rješavanje jednadžbe $g_M(x) = |r(x)|$. Ako usporedimo postupak traženja max-funkcije za $n \geq 3$ polinoma s postupkom za traženje max-funkcije za dva polinoma, suštinski je isti, ali je tehnički složeniji, jer se spajaju polinom i max-funkcija za $n - 1$ polinom koju je složenije pamtit.

Također, bitno je istaknuti da ako postoji polinom $p^* \in \{p_1, \dots, p_s\}$ koji se ne pojavljuje u definiciji funkcije g_M (ne postoji interval I_j na kojem je $g_M = p^*$), tada nije potrebno tražiti nultočke polinoma $p^* + r$ i $p^* - r$, jer se p^* ne može pojaviti u definiciji od g_M^N .

Funkcija g_M^N je neprekidna jer je maksimum dvije neprekidne funkcije (lema 2.14), a također vrijedi i da je g_M^N po dijelovima (unutrašnjost intervala I_j^N) neprekidno diferencijabilna (dokaz bi išao slično kao dokaz propozicije 2.15). Za izračun minimuma funkcije g_M^N vrijedi analogon propozicije 2.16.

Propozicija 2.18 *Neka je max-funkcija $g_M = \max\{|p_1|, \dots, |p_s|\}$ zadana na intervalima I_j , $j = 1, \dots, k$, $I_1 = \langle -\infty, a_1 \rangle$, $I_j = \langle a_{j-1}, a_j \rangle$, $j = 2, \dots, k - 1$ i $I_k = \langle a_{k-1}, \infty \rangle$. Nadalje, neka su za $j = 1, \dots, s$, s $t_{j,1}, \dots, t_{j,n_j}$ označeni sve stacionarne točke polinoma p_j (rješenja jednadžbe*



Slika 2.3: Grafovi funkcija g_M (puna crta) i s (isprekidana crta) (slika (a)) i funkcije $g_M^N = \max\{g_M, |s|\}$ (slika (b)). iz primjera 2.19

$p'_j(t) = 0$). Tada je

$$\min_{t \in \mathbb{R}} g_M(t) = \min\{g_M(a_1), g_M(a_{k-1}), \min_{j=1, \dots, s} \{g_M(t_{j,1}), \dots, g_M(t_{j,n_j})\}\}. \quad (2.29)$$

Izračun u formuli (2.29) se može i ubrzati. Naime, ako postoji polinom $p^* \in \{p_1, \dots, p_s\}$ koji se ne pojavljuje u definiciji od g_M , nije potrebno tražiti stacionarne točke od p^* . Također, umjesto računanja vrijednosti $g_M(t_{j,l})$, gdje je $t_{j,l}$ stacionarna točka polinoma p_j , može se vidjeti je li na intervalu u kojem se nalazi $t_{j,l}$ funkcija g_M zadana polinomom p_j . Ako je, dovoljno je izračunati $|p_j(t_{j,l})|$, a ako nije, ne treba ništa računati.

Propozicija 2.16 je samo specijalni slučaj propozicije 2.18, gdje je korištena činjenica da se rubovi intervala na kojima je definirana max-funkcija mogu jednostavno odrediti, što nije slučaj kod max-funkcije za više polinoma.

Primjer 2.19 *Nastavno na primjer 2.17, neka je max-funkcija g_M definirana s*

$$g_M(t) = \max\left\{\underbrace{|t^4 + 3t^3 - 3t^2 - 11t - 6|}_{p(t)}, \underbrace{|t^3 - 2t^2 - 5t + 6|}_{r(t)}\right\}$$

te neka je zadan polinom $s(t) = 9 - 4t^2$ (slika 2.3.a). Funkcija g_M definirana je na intervalima $I_1 = \langle -\infty, a_1 \rangle$, $I_j = \langle a_{j-1}, a_j \rangle$, $j = 2, \dots, 6$, i $I_7 = \langle a_6, \infty \rangle$, pri čemu su točke a_i definirane s $a_1 = -4,297$, $a_2 = -2,291$, $a_3 = -1,787$, $a_4 = 0$, $a_5 = 1,907$ i $a_6 = 2,084$. Definicija funkcije g_M po intervalima je

Interval	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$g_M(t) =$	$p(t)$	$-r(t)$	$-p(t)$	$r(t)$	$-p(t)$	$-r(t)$	$p(t)$

Želimo formirati funkciju $g_M^N = \max\{g_M, |s|\}$. Slijedeći korake iz postupka, prvo izračunamo nultočke polinoma $p + s$ i $p - s$. Redom, to su točke $b_1 = -4,236$, $b_2 = -1,407$, $b_3 = -1,303$,

$b_4 = 0,236$, $b_5 = 1,807$ i $b_6 = 2,303$. Točke b_1, b_2, b_3 su elementi intervala na kojima je g_M definiran s $|r|$. Točke $b_4, b_5 \in I_5$, stoga one generiraju podintervale $I_{5,1}^N = \langle a_4, b_4 \rangle$, $I_{5,2}^N = \langle b_4, b_5 \rangle$ i $I_{5,3}^N = \langle b_5, a_5 \rangle$. Konačno, $b_6 \in I_7$, pa ona generira podjelu intervala I_7 na podintervale $I_{7,1}^N = \langle a_6, b_6 \rangle$ i $I_{7,2}^N = \langle b_6, \infty \rangle$.

Kada se izračunaju nultočke polinoma $r + s$ i $r - s$, dobiju se točke $c_1 = -4,051$, $c_2 = -1,638$, $c_3 = -0,483$, $c_4 = 1,237$, $c_5 = 1,534$, $c_6 = 7,402$. Točke c_4, c_5 i c_6 nalaze se u intervalima u kojima je g_M definirana s $|p|$ pa ih možemo zanemariti. Točka $c_1 \in I_2$ pa generira podjelu intervala I_2 na $I_{2,1}^N = \langle a_1, c_1 \rangle$ i $I_{2,2}^N = \langle c_1, a_2 \rangle$. Točke c_2, c_3 su iz intervala I_4 što rezultira njegovom podjelom na tri podintervala: $I_{4,1}^N = \langle a_3, c_2 \rangle$, $I_{4,2}^N = \langle c_2, c_3 \rangle$ i $I_{4,3}^N = \langle c_3, a_4 \rangle$.

Na intervalima I_1, I_3 i I_6 nije bilo podjela pa treba vidjeti postiže li na tom intervalu veću vrijednost funkcija g_M ili $|s|$. Za provjeru te činjenice dovoljno je izračunati g_M i s u jednoj točki intervala i na osnovu tog rezultata vidjeti kojom funkcijom će biti definirana g_M^N na tom intervalu.

Nakon svih ovih podjela funkcija g_M^N je definirana s

$$g_M^N = \frac{\text{Interval} \quad I_1^N \quad I_{2,1}^N \quad I_{2,2}^N \quad I_3^N \quad I_{4,1}^N \quad I_{4,2}^N \quad I_{4,3}^N \quad I_{5,1}^N \quad I_{5,2}^N \quad I_{5,3}^N \quad I_6^N \quad I_{7,1}^N \quad I_{7,2}^N}{p \quad -r \quad -s \quad -s \quad -s \quad r \quad s \quad s \quad -p \quad -s \quad -s \quad -s \quad -s \quad p}$$

Sada još preostaje posljednji korak (korak K5 iz postupka za određivanje g_M^N), objedinjavanje intervala. Iz prethodne tablice je vidljivo da je na uzastopnim intervalima $I_{2,2}^N, I_3^N, I_{4,1}^N, g_M^N$ zadana s $-s$, na intervalima $I_{4,3}^N$ i $I_{5,1}^N$ sa s , a na intervalima $I_{5,3}^N, I_6^N$ i $I_{7,1}^N$ opet s $-s$.

Nakon objedinjavanja i renumeracije intervala, funkcija g_M^N se može zapisati kao

$$g_M^N(t) = \frac{\text{Interval} \quad I_1^N \quad I_2^N \quad I_3^N \quad I_4^N \quad I_5^N \quad I_6^N \quad I_7^N \quad I_8^N}{p(t) \quad -r(t) \quad -s(t) \quad r(t) \quad s(t) \quad -p(t) \quad -s(t) \quad p(t)}$$

pri čemu su intervali I_j^N određeni točkama $a_1^N = -4,297$, $a_2^N = -4,051$, $a_3^N = -1,638$, $a_4^N = -0,483$, $a_5^N = 0,236$, $a_6^N = 1,807$ i $a_7^N = 2,303$.

Konačni graf funkcije g_M^N nalazi se na slici 2.3.b.

2.6.4 Rješavanje polinomnih sustava primjenom funkcije MRN

U potpoglavljima 2.6.1–2.6.3 je opisan postupak kojim se konstruira max-funkcija te pronalazi točka u kojoj se postiže njezin minimum. U ovom poglavlju se pokazuje kako se taj postupak primjenjuje na funkciju $MRN_{\mathbf{x},s}$. Nakon toga je dan algoritam za rješavanje općeg polinomnog sustava metodom najdublje spusta za slučaj kada se dubina spusta određuje pomoću $MRN_{\mathbf{x},s}$.

Po definiciji, funkcija $MRN_{\mathbf{x},s} : \mathbb{R} \rightarrow \mathbb{R}$ je restrikcija funkcije

$$MRN = \|\mathbf{F}(\mathbf{x})\|_\infty = \max_{i \in \{1, \dots, n\}} |f_i(\mathbf{x})|$$

na pravac koji prolazi kroz zadanu točku \mathbf{x} i ima vektor smjera \mathbf{s} ,

$$\text{MRN}_{\mathbf{x},\mathbf{s}}(\lambda) = \text{MRN}(\mathbf{x} + \lambda\mathbf{s}) = \max_{i \in \{1, \dots, n\}} |f_i(\mathbf{x} + \lambda\mathbf{s})| = \max_{i \in \{1, \dots, n\}} |f_{i,\mathbf{x},\mathbf{s}}(\lambda)|.$$

Da bi se uopće moglo raditi s funkcijom $\text{MRN}_{\mathbf{x},\mathbf{s}}$, potrebno je prvo polinome $f_{i,\mathbf{x},\mathbf{s}}$ zapisati u standardnoj bazi prostora polinoma. Pri tome se koriste postupci opisani u poglavlju 2.4, jedino što naglasak nije na prikazu jednog polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}$ u standardnoj bazi, već treba prikazati cijeli niz polinoma $f_{i,\mathbf{x},\mathbf{s}}$ u standardnoj bazi. Formalno, za svaki polinom $f_{i,\mathbf{x},\mathbf{s}}$ potrebno je odrediti koeficijente $b_{r,i}$ takve da vrijedi

$$f_{i,\mathbf{x},\mathbf{s}}(\lambda) = f_i(\mathbf{x} + \lambda\mathbf{s}) = \sum_{r=1}^{k_i} a_{r,i}(\mathbf{x} + \lambda\mathbf{s})^{\mathbf{d}_{r,i}} = \sum_{r=0}^{\deg f_i} b_{r,i}\lambda^r.$$

Za određivanja stupnja polinoma $f_{i,\mathbf{x},\mathbf{s}}$ važnu ulogu ima propozicija 2.5.

Kako je spomenuto u potpoglavlju 2.4, koeficijenti $b_{r,i}$ se mogu odrediti ili primjenom simboličkog računa ili korištenjem numeričke interpolacije polinomom. Ako je struktura polinoma $f_{i,\mathbf{x},\mathbf{s}}$ relativno jednostavna u smislu da ima mali broj članova u kojima se pojavljuje više varijabli[†], onda je bolje koristiti simbolički račun. S druge strane, ako je struktura složena u smislu da ima puno članova koji su umnožak više varijabli, bolje je koristiti postupak s polinomnom interpolacijom. U programu `PolySolver` za određivanje prikaza polinoma $f_{i,\mathbf{x},\mathbf{s}}$ u standardnoj bazi polinoma koristi se simbolički račun.

U algoritmu 2.7 je opisan postupak kojim se formira funkcija $\text{MRN}_{\mathbf{x},\mathbf{s}}$ i pronalazi točka u kojoj postiže minimum. Algoritam je utemeljen na lemi 2.13, postupku opisanom u potpoglavlju 2.6.3 i propoziciji 2.18. U algoritmu se pretpostavlja da postoji struktura za prikaz po dijelovima zadane funkcije (potpoglavlje 2.6.1), koja može reprezentirati intervale i funkcije kojima je definirana funkcija $\text{MRN}_{\mathbf{x},\mathbf{s}}$, ne ulazeći u detalje implementacije.

Nakon što je detaljno opisano kako formirati funkciju $\text{MRN}_{\mathbf{x},\mathbf{s}}$ i naći točku u kojoj postiže najmanju vrijednost, moguće je napraviti algoritam sličan algoritmu 2.4, s tom razlikom da se umjesto funkcije $\text{RSS}_{\mathbf{x},\mathbf{s}}$ za određivanje dubine spusta koristi $\text{MRN}_{\mathbf{x},\mathbf{s}}$.

Određivanje duljine spusta pomoću funkcije $\text{MRN}_{\mathbf{x},\mathbf{s}}$ je bitno složenije od određivanjem duljine spusta pomoću $\text{RSS}_{\mathbf{x},\mathbf{s}}$, što se lako vidi uvidom u algoritme 2.7 i 2.8 (za korištenje $\text{MRN}_{\mathbf{x},\mathbf{s}}$) te 2.4 i 2.6 (za $\text{RSS}_{\mathbf{x},\mathbf{s}}$).

U tablici 2.1 su popisani glavni zadatci koje treba obaviti pri određivanju duljine spusta korištenjem odgovarajuće funkcije. Kako se vidi iz tablice, funkciju $\text{MRN}_{\mathbf{x},\mathbf{s}}$ se u pravilu ne isplati koristiti sa stajališta brzine jer je računski zahtjevnija. Eventualno, njezino korištenje može biti isplativo kada je riječ o relativno malom problemu (mali broj jednadžbi n), pogotovo ako je

[†]Npr. u polinomu $2x^3 + xy - x^2y^3 + y^2 + 10$ članovi $2x^3$ i y^2 su jednostavni za simbolički izračun jer se u njima pojavljuje samo jedna varijabla, dok se u članovima xy i x^2y^3 pojavljuju dvije varijable, a član x^2y^3 je složeniji za izračun od člana xy .

Algoritam 2.7 Algoritam za određivanje $\lambda^* = \arg \min_{\lambda \in \mathbb{R}} \text{MRN}_{\mathbf{x},\mathbf{s}}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} \text{MRN}(\mathbf{x} + \lambda \mathbf{s})$

Pretpostavke: Polinomni sustav $F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x})) = \mathbf{0}$.

Zadana trenutna točka \mathbf{x} i smjer spusta \mathbf{s} .

I_j su intervali, a g_j je restrikcija $\text{MRN}_{\mathbf{x},\mathbf{s}}^s = \max\{|f_1|, \dots, |f_s|\}$ na I_j .

```

1: for i=1 to n
2:   Nađi  $f_i^*$ , prikaz polinoma  $f_{i,\mathbf{x},\mathbf{s}}$  u standardnoj bazi koristeći simbolički račun.
3: end for
4: Nađi sve realne nultočke polinoma  $f_1^* + f_2^*$  i  $f_1^* - f_2^*$  te ih poredaj po veličini:  $a_1 < \dots < a_k$ .
5: Definiraj  $a_0 = -\infty$  i  $a_{k+1} = \infty$  i iz njih generiraj intervale  $I_j = \langle a_{j-1}, a_j \rangle$ ,  $j = 1, \dots, k+1$ .
6: for i=1 to k+1
7:   Uzmi točku  $t \in I_i$ . ▷ npr. sredina intervala
8:   if  $|f_1^*(t)| > |f_2^*(t)|$  then  $g_i = f_1^*$  else  $g_i = f_2^*$ .
9: end for ▷ formirana funkcija  $\text{MRN}_{\mathbf{x},\mathbf{s}}^2$ 
10: for s=3 to n
11:   for r=1 to s-1
12:     Nađi točke  $b_1 < \dots < b_m$ , realne nultočke polinoma  $f_r^* + f_s^*$  i  $f_r^* - f_s^*$ .
13:     for j=1 to k+1
14:       if  $g_j = f_r^*$  and  $\{b_1, \dots, b_m\} \cap I_j = \emptyset$  then
15:         Uzmi točku  $t \in I_j$ 
16:         if  $|f_s^*(t)| > |f_r^*(t)|$  then  $g_j = f_s^*$ 
17:       else if  $g_j = f_r^*$  and  $\{b_1, \dots, b_m\} \cap I_j \neq \emptyset$  then
18:         Neka su  $b_1^j, \dots, b_{m_j}^j \in I_j = \langle a_{j-1}, a_j \rangle$ , nultočke  $f_r^* + f_s^*$  i  $f_r^* - f_s^*$  iz  $I_j$ 
19:         Definiraj  $b_0^j = a_{j-1}$ ,  $b_{m_j+1}^j = a_j$ 
20:         for i=1 to  $m_j + 1$ 
21:           Neka je interval  $I_{j,i} = \langle b_{i-1}^j, b_i^j \rangle$ ;  $t = (b_{i-1}^j + b_i^j)/2$ 
22:           if  $|f_r^*(t)| > |f_s^*(t)|$  then  $g_{j,i} = f_r^*$  else  $g_{j,i} = f_s^*$ .
23:         end for
24:       end if
25:     end for
26:   end for
27:   Objedini susjedne intervale koji imaju jednaku funkciju. ▷ formirana funkcija
28:   Renumeriraj intervale da ostane  $I_1, \dots, I_k$ ,  $I_j = \langle a_{j-1}, a_j \rangle$ . ▷  $\text{MRN}_{\mathbf{x},\mathbf{s}}^s$ 
29: end for
30: fmin=+INF ▷ neki jako veliki broj
31: for i=1 to k-1 ▷ Provjeri rubove intervala (propozicija 2.18)
32:   if  $|g_i(a_i)| < \text{fmin}$  then fmin= $|g_i(a_i)|$ :  $\lambda^* = a_i$ .
33: end for
34: for s=1 to n
35:   Nađi  $c_1, \dots, c_{m_s}$  nultočke polinoma  $(f_s^*)'(\lambda) = 0$ .
36:   for i=1 to  $m_s$  ▷ Provjeri stacionarne točke polinoma  $f_s^*$  (propozicija 2.18)
37:     Odredi  $j$  takav da je  $c_i \in I_j$ .
38:     if  $g_j = f_s^*$  then
39:       if  $|g_j(c_i)| < \text{fmin}$  then fmin= $|g_j(c_i)|$ :  $\lambda^* = c_i$ .
40:     end if
41:   end for
42: end for
43: vрати  $\lambda^*$ .

```

Algoritam 2.8 Rješavanje općeg polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ metodom najdublje spusta po proizvoljnom smjeru. Dubina spusta iz zadane točke \mathbf{x} u smjeru \mathbf{s} se određuje iz funkcije $\text{MRN}_{\mathbf{x},\mathbf{s}}$.

- 1: Neka je zadana početna točka $\mathbf{x}^{(0)}$.
- 2: $k=0$
- 3: **while** nije zadovoljen neki kriterij zaustavljanja iz poglavlja 2.5
- 4: Neka je neki način određen smjer $\mathbf{s}^{(k)}$.
- 5: Primjenom algoritma 2.7 odredi

$$\lambda^* = \arg \min_{\lambda \in \mathbb{R}} \text{MRN}_{\mathbf{x}^{(k)},\mathbf{s}^{(k)}}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} \text{MRN}(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}).$$
- 6: Nova točka je $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)} \mathbf{s}^{(k)}$.
- 7: $k = k + 1$
- 8: **end while**

stupanj polinoma $f_{i,\mathbf{x},\mathbf{s}}$ mali (do četiri) tako da se nultočke mogu naći korištenjem zatvorenih formula.

Tablica 2.1: Računski zahtjevne zadaće koje treba izvesti kod određivanja duljine spusta korištenjem $\text{RSS}_{\mathbf{x},\mathbf{s}}$, odnosno $\text{MRN}_{\mathbf{x},\mathbf{s}}$.

Zadatak	$\text{RSS}_{\mathbf{x},\mathbf{s}}$	$\text{MRN}_{\mathbf{x},\mathbf{s}}$
Prikazati polinome $f_{i,\mathbf{x},\mathbf{s}}$ u standardnoj bazi prostora polinoma	DA	DA
Izračunati koeficijente polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}$ u standardnoj bazi	DA	NE
Traženje nultočaka polinoma	DA ^a	DA ^b

^a - jedna jednadžba s polinomom stupnja $2d^M - 1$, $d^M = \max_i \deg f_{i,\mathbf{x},\mathbf{s}}$

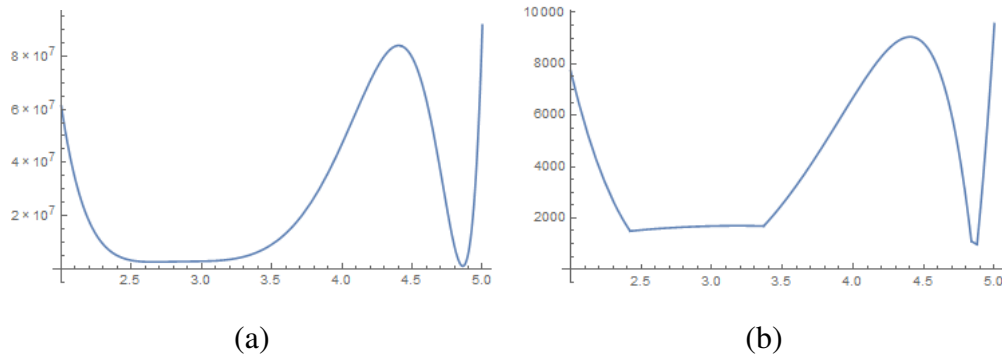
^b - $\binom{n}{2}$ jednadžbi s polinomom stupnja d^M i n jednadžbi s polinomom stupnja $d^M - 1$.

Ipak, zanimljivo je vidjeti kako primjena funkcije $\text{MRN}_{\mathbf{x},\mathbf{s}}$, odnosno $\text{RSS}_{\mathbf{x},\mathbf{s}}$ može voditi do različitih rješenja.

Test primjer koji se promatra je problem sendra, iz kolekcije PoSSo test primjera, originalno stvorenih za testiranje algoritama koji rješavaju probleme iz područja računalne algebre.

Problem sendra se sastoji od rješavanja polinomnog sustava $\mathbf{F}(x,y) = (f_1(x,y), f_2(x,y)) = \mathbf{0}$, gdje su funkcije f_1 i f_2 definirane s

$$\begin{aligned}
 f_1(x,y) &= -270x^4y^3 - 314xy^4 - 689xy^3 + 1428 \\
 f_2(x,y) &= 36x^7 + 417x^6y - 200x^6 - 422x^5y^2 - 174x^5y - 270x^4y^3 - 966x^4y^2 \\
 &+ 529x^4y + 36x^4 + 1428x^3y^4 + 529x^3y^3 - 1475x^2y^5 + 269x^2y^4 + 1303x^2y^3 \\
 &- 788x^2y^2 + 510xy^6 + 49xy^5 - 314xy^4 - 689xy^3 - 267y^6 + 262y^5 + 177y^4
 \end{aligned}$$



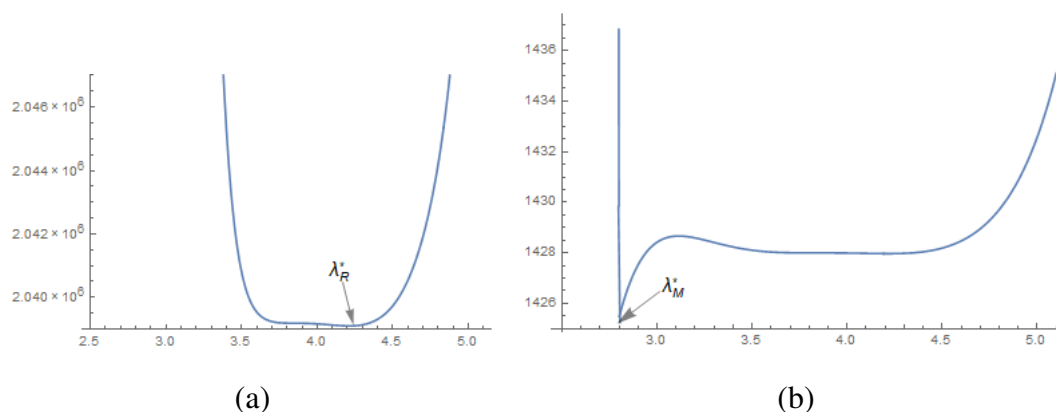
Slika 2.4: Grafovi funkcija $RSS_{\mathbf{x},s}(\lambda)$ (slika (a)) i $MRN_{\mathbf{x},s}(\lambda)$ (slika (b)) za $\lambda \in [2, 5]$.

Objektive jednadžbe sustava imaju monome stupnja 7, što znači da vrijedi $\deg f_{1,\mathbf{x},s}, \deg f_{2,\mathbf{x},s} \leq 7$. Također, parcijalne derivacije pojedinih jednadžbi imat će velike vrijednosti što znači da mala promjena argumenata uzrokuje veliku promjenu vrijednosti.

Ako se koristi $RSS_{\mathbf{x},s}$ za određivanje duljine spusta, polinom $RSS_{\mathbf{x},s}$ je stupnja 14, a potrebno je naći rješenje jedne jednadžbe reda 13. S druge strane, ako se za određivanje duljine koraka koristi $MRN_{\mathbf{x},s}$, onda je potrebno riješiti dvije polinomne jednadžbe reda 7 i još naći nultočke derivacija polinoma $f_{1,\mathbf{x},s}$ i $f_{2,\mathbf{x},s}$ (dvije jednadžbe stupnja 6).

Neka je $\mathbf{x} = (-2,3245, -2,9128)$ i smjer spusta \mathbf{s} za \mathbf{x} određen kao Newtonov smjer. Na slici 2.4 su prikazani grafovi funkcija $RSS_{\mathbf{x},s}(\lambda)$ i $MRN_{\mathbf{x},s}(\lambda)$ za $\lambda \in [2, 5]$. Funkcija $RSS_{\mathbf{x},s}(\lambda)$ poprima puno veće vrijednosti na promatranom intervalu, ali načelno, oblik $RSS_{\mathbf{x},s}(\lambda)$ se skoro poklapa s $MRN_{\mathbf{x},s}(\lambda)$. Iz slike se vide dva područja gdje bi mogla biti najmanja vrijednost odgovarajuće funkcije - jedno područje je oko $\lambda = 3$, a drugo oko $\lambda = 5$. Najmanju vrijednost $RSS_{\mathbf{x},s}(\lambda)$ postiže za $\lambda_R^* = 4,8604$ pa je nova aproksimacija $\mathbf{x}_R^{(1)} = (-4,3402, 0,7070)$. S druge strane, $MRN_{\mathbf{x},s}(\lambda)$ najmanju vrijednost postiže za $\lambda_M^* = 4,8772$ i nova aproksimacija je $\mathbf{x}_M^{(1)} = (-4,3351, 0,6980)$. Premda se razlika između $\mathbf{x}_R^{(1)}$ i $\mathbf{x}_M^{(1)}$ ne čini velikom, postupak iteracija koji započinje u točki $\mathbf{x}_R^{(1)}$, za vektor smjera spusta bira Newtonov smjer, a duljinu spusta određuje pomoću $RSS_{\mathbf{x},s}$, konvergira prema točki $(0,6372, 1,2459)$ koja nije rješenje problema, ali postupak iteracija koji započinje u istoj točki i na isti način određuje smjer spusta, samo što duljinu spusta određuje pomoću funkcije $MRN_{\mathbf{x},s}$, konvergira prema rješenju $\mathbf{x}_1^* = (0,28004, 1,6183)$. S druge strane, iterativni postupak koji započinje u $\mathbf{x}_M^{(1)}$ i za vektor spusta uzima Newtonov smjer, bez obzira koristi li se za određivanje duljine silaska $RSS_{\mathbf{x},s}$ ili $MRN_{\mathbf{x},s}$, konvergira prema rješenju $\mathbf{x}_2^* = (-2,568, 0,5309)$.

Na slici 2.5 prikazani su grafovi funkcija $RSS_{\mathbf{x},s}(\lambda)$ (slika (a)) i $MRN_{\mathbf{x},s}(\lambda)$ (slika (b)) za slučaj kada je $\mathbf{x} = \mathbf{x}_R^{(1)}$, a smjer spusta je Newtonov smjer. Na slici su naznačene točke u kojima obje funkcije postižu svoj minimum i vidljivo je da u ovom slučaju postoji velika razlika između λ_M^* i λ_R^* , točaka u kojima $MRN_{\mathbf{x},s}(\lambda)$ i $RSS_{\mathbf{x},s}(\lambda)$ postiže minimum, respektivno. Razlika između λ_M^* i λ_R^* uzrokuje razliku između točaka $\mathbf{x}_M^{(2)} = \mathbf{x}_R^{(1)} + \lambda_M^* \mathbf{s}$ i $\mathbf{x}_R^{(2)} = \mathbf{x}_R^{(1)} + \lambda_R^* \mathbf{s}$,



Slika 2.5: Grafovi funkcija $RSS_{\mathbf{x},s}(\lambda)$ (slika (a)) i $MRN_{\mathbf{x},s}(\lambda)$ (slika (b)) za $\mathbf{x} = \mathbf{x}_R^{(1)}$, s Newtonov smjer i $\lambda \in [2, 5, 5]$.

što kao rezultat daje da jedan proces konvergira, a drugi ne.

U ovom slučaju se korištenje max-norme za određivanje duljine spusta pokazalo boljim od korištenja euklidske norme, ali lako se može naći i obrnuti slučaj.

2.7 Rješavanje linearnih sustava metodom najdublje spusta

Kako su linearni sustavi isto tako polinomni sustavi, moguće je primijeniti metodu najdublje spusta i na rješavanje linearnih sustava.

Neka je zadan sustav linearnih jednačnji

$$\mathbf{Ax} = \mathbf{b}, \quad \mathbf{A} = (a_{ij}) \in \mathbf{M}_n, \quad \mathbf{x}, \mathbf{b} \in \mathbb{R}^n. \quad (2.30)$$

U skladu s notacijom polinomnih sustava, linearni sustav (2.30) se može zapisati kao $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, pri čemu su funkcije f_1, \dots, f_n definirane s

$$f_i(\mathbf{x}) = a_{i1}x_1 + \dots + a_{in}x_n - b_i.$$

Jedini smjer spusta koji ima smisla koristiti je smjer koordinatnih osi, jer kod ostalih načina je određivanje smjera spusta preskupo (poglavlje 3).

Analogno postupku opisanom u potpoglavljima 2.1 i 2.2, definira se funkcija RSS kao zbroj kvadrata reziduala,

$$RSS(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2 = \sum_{i=1}^n f_i^2(\mathbf{x}).$$

U slučaju kada je vektor spusta smjer koordinatne osi e_k , promatra se restrikcija funkcije RSS

na pravac koji prolazi kroz \mathbf{x} i ima vektor smjera \mathbf{e}_k ,

$$\begin{aligned} \text{RSS}_{\mathbf{x}, \mathbf{e}_k}(\lambda) &= \|\mathbf{F}(\mathbf{x} + \lambda \mathbf{e}_k)\|_2^2 = \|\mathbf{A}(\mathbf{x} + \lambda \mathbf{e}_k) - \mathbf{b}\|_2^2 \\ &= \|\mathbf{A}\mathbf{x} - \mathbf{b} + \lambda \mathbf{A}\mathbf{e}_k\|_2^2 \\ &= \|\mathbf{r} + \lambda \mathbf{a}^k\|_2^2 = (\mathbf{r} + \lambda \mathbf{a}^k | \mathbf{r} + \lambda \mathbf{a}^k) \\ &= (\mathbf{r} | \mathbf{r}) + 2\lambda (\mathbf{r} | \mathbf{a}^k) + \lambda^2 (\mathbf{a}^k | \mathbf{a}^k), \end{aligned}$$

pri čemu je $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$, a \mathbf{a}^k je k -ti stupac matrice \mathbf{A} , $\mathbf{A} = [\mathbf{a}^1, \dots, \mathbf{a}^n]$. Točka λ^* u kojoj se postiže minimum funkcije $\text{RSS}_{\mathbf{x}, \mathbf{e}_k}$ dobije se izjednačavanjem derivacije od $\text{RSS}_{\mathbf{x}, \mathbf{e}_k}$ s nulom,

$$\lambda^* = -\frac{(\mathbf{r} | \mathbf{a}^k)}{(\mathbf{a}^k | \mathbf{a}^k)} = -\frac{(\mathbf{r} | \mathbf{a}^k)}{\|\mathbf{a}^k\|_2^2}. \quad (2.31)$$

S (2.31) je definirana duljina spusta, a sljedeća točka u iterativnom postupku se dobije kao

$$\mathbf{x}^* = \mathbf{x} + \lambda^* \mathbf{e}_k.$$

Kako je već spomenuto, metoda je određena načinom odabira smjera spusta i određivanja duljine spusta. Varijanta metode najdublje spusta za rješavanje linearnih sustava u kojoj se za smjer spusta primjenjuje ciklički odabir koordinatnih osi (prvo \mathbf{e}_1 , pa $\mathbf{e}_2, \dots, \mathbf{e}_n$, pa onda opet \mathbf{e}_1 itd.), a duljina spusta je određena s (2.31) naziva se DD-LS[‡].

Postupak rješavanja linearnog sustava metodom DD-LS prikazan je u algoritmu 2.9. Potrebno je naglasiti da se pod pojmom “jedan korak” metode DD-LS podrazumijeva izmjena svih koordinatnih osi kao vektora spusta (koraci 5–8 u algoritmu 2.9), odnosno prelazak s $\mathbf{x}^{(j)}$ na $\mathbf{x}^{(j+1)}$. Pod pojmom “međukorak” podrazumijeva se korak samo za jednu koordinatnu os (koraci 6 i 7), $\mathbf{x}^{(j,k-1)}$ na $\mathbf{x}^{(j,k)}$.

Algoritam 2.9 Rješavanje linearnog sustava $\mathbf{A}\mathbf{x} = \mathbf{b}$ metodom najdublje spusta po koordinatnim osima.

- 1: Neka je zadana početna točka $\mathbf{x}^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})$.
 - 2: $j=0$
 - 3: **while** nije zadovoljen kriterij zaustavljanja
 - 4: $\mathbf{x}^{(j,0)} = \mathbf{x}^{(j)}$.
 - 5: **for** $k = 1$ **to** n
 - 6: Odredi duljinu spusta λ^* za točku $\mathbf{x}^{(j,k-1)}$ i vektor spusta \mathbf{e}_k koristeći (2.31).
 - 7: $\mathbf{x}^{(j,k)} = \mathbf{x}^{(j,k-1)} + (\lambda^* - x_k^{(j,k-1)})\mathbf{e}_k$ (promjena k -te koordinate točke $\mathbf{x}^{(j,k-1)}$).
 - 8: **end for**
 - 9: $\mathbf{x}^{(j+1)} = \mathbf{x}^{(j,n)}$.
 - 10: $j = j + 1$.
 - 11: **end while**
-

[‡]Skraćenica za *Deepest Descent Linear System*.

Rješavanja linearnog sustava metodom DD-LS povezano je s Gauss-Seidelovom iterativnom metodom za rješavanje linearnih sustava [32]. Gauss-Seidelova iterativna metoda za rješavanja linearnog sustava $\mathbf{Ax} = \mathbf{b}$ zasnovana je na transformaciji

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow (\mathbf{L} + \mathbf{D})\mathbf{x} + \mathbf{Rx} = \mathbf{b} \Leftrightarrow (\mathbf{L} + \mathbf{D})\mathbf{x} = \mathbf{b} - \mathbf{Rx}, \quad (2.32)$$

a definirana je s

$$(\mathbf{L} + \mathbf{D})\mathbf{x}^{(j+1)} = \mathbf{b} - \mathbf{Rx}^{(j)}. \quad (2.33)$$

U rastavu (2.32) su matrice \mathbf{L} , \mathbf{D} , \mathbf{R} redom donji trokut, dijagonala i gornji trokut matrice $\mathbf{A} = (a_{ij})$,

$$\mathbf{L} = (l_{ij}) = \begin{cases} a_{ij}, & i > j, \\ 0, & \text{inače;} \end{cases} \quad \mathbf{D} = (d_{ij}) = \begin{cases} a_{ii}, & i = j, \\ 0, & \text{inače;} \end{cases} \quad \mathbf{R} = (r_{ij}) = \begin{cases} a_{ij}, & i < j, \\ 0, & \text{inače.} \end{cases}$$

K -ta jednadžba sustava (2.33) se može zapisati kao

$$\begin{aligned} ((\mathbf{L} + \mathbf{D})\mathbf{x}^{(j+1)})_k &= (\mathbf{b} - \mathbf{Rx}^{(j)})_k \\ \sum_{i=1}^{k-1} l_{ki}x_i^{(j+1)} + d_{kk}x_k^{(j+1)} &= b_k - \sum_{i=k+1}^n r_{ki}x_i^{(j)} \\ \sum_{i=1}^{k-1} a_{ki}x_i^{(j+1)} + a_{kk}x_k^{(j+1)} &= b_k - \sum_{i=k+1}^n a_{ki}x_i^{(j)} \\ a_{kk}x_k^{(j+1)} &= b_k - \sum_{i=1}^{k-1} a_{ki}x_i^{(j+1)} - \sum_{i=k+1}^n a_{ki}x_i^{(j)}, \end{aligned}$$

iz čega slijedi

$$x_k^{(j+1)} = \frac{1}{a_{kk}} \left(b_k - \sum_{i=1}^{k-1} a_{ki}x_i^{(j+1)} - \sum_{i=k+1}^n a_{ki}x_i^{(j)} \right). \quad (2.34)$$

Relacija (2.34) pokazuje kako izračunati k -tu komponentu novog vektora iterativnog postupka $\mathbf{x}^{(j+1)}$, ako se znaju vrijednosti prethodnih $k - 1$ komponenti vektora $\mathbf{x}^{(j+1)}$ i vektor $\mathbf{x}^{(j)}$.

Sljedeća propozicija opisuje vezu između Gauss-Seidelove metode i DD-LS metode.

Propozicija 2.20 *Neka je $\mathbf{Ax} = \mathbf{b}$ zadan linearni sustav reda n . Tada je jedan korak metode DD-LS za rješavanje sustava $\mathbf{Ax} = \mathbf{b}$ jednak jednom koraku Gauss-Seidelove metode za rješavanje sustava $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$.*

Dokaz: Neka je $\mathbf{x}^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})$. Želimo pokazati da je vektor $\mathbf{x}^{(1)} = (x_1^{(1)}, \dots, x_n^{(1)})$ koji se dobije nakon jednog koraka metode DD-LS pri rješavanju sustava $\mathbf{Ax} = \mathbf{b}$ isti kao i vektor $\mathbf{x}_{GS}^{(1)}$ koji se dobije nakon jednog koraka Gauss-Seidelove metode na sustavu $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$, što

će biti dokazano ako se pokaže da su vektori koji se dobiju nakon k -tog međukoraka, $\mathbf{x}^{(0,k)}$ i $\mathbf{x}_{GS}^{(0,k)}$, jednaki. Iz načina na koji su međukoraci definirani, jasno je da vrijedi

$$\mathbf{x}^{(0,k)} = (x_1^{(1)}, \dots, x_k^{(1)}, x_{k+1}^{(0)}, \dots, x_n^{(0)}), \quad (2.35)$$

odnosno da je prvih k komponenti vektora $\mathbf{x}^{(0,k)}$ jednako komponentama vektora $\mathbf{x}^{(1)}$, a posljednjih $n - k$ komponenti vektora $\mathbf{x}^{(0,k)}$ je jednako komponentama od $\mathbf{x}^{(0)}$. Analogan rezultat vrijedi i za vektor $\mathbf{x}_{GS}^{(0,k)}$.

Neka je dan zapis matrice po stupcima, $\mathbf{A} = [\mathbf{a}^1, \dots, \mathbf{a}^n]$.

Za $k = 1$, metoda DD-LS daje vektor $\mathbf{x}^{(0,1)} = (x_1^{(1)}, x_2^{(0)}, \dots, x_n^{(0)})$, a za $x_1^{(1)}$ vrijedi

$$x_1^{(1)} = -\frac{(\mathbf{r}|\mathbf{a}^1)}{\|\mathbf{a}^1\|_2^2} - x_1^{(0)} = -\frac{(\mathbf{A}\mathbf{x}^{(0)} - \mathbf{b}|\mathbf{a}^1)}{\|\mathbf{a}^1\|_2^2} - x_1^{(0)} = \frac{(\mathbf{a}^1)^T \mathbf{b} - (\mathbf{a}^1)^T \mathbf{A}\mathbf{x}^{(0)}}{(\mathbf{a}^1)^T \mathbf{a}^1} - x_1^{(0)}. \quad (2.36)$$

Uvaži li se da je $(\mathbf{a}^1)^T \mathbf{A}$ prvi redak matrice $\mathbf{A}^T \mathbf{A}$, a $(\mathbf{a}^1)^T \mathbf{b}$ prva komponenta vektora $\mathbf{A}^T \mathbf{b}$, (2.36) se može zapisati

$$x_1^{(1)} = \frac{1}{(\mathbf{A}^T \mathbf{A})_{11}} \left(\sum_{i=1}^n (\mathbf{A}^T \mathbf{A})_{i1} x_i^{(0)} \right) - x_1^{(0)} = \frac{1}{(\mathbf{A}^T \mathbf{A})_{11}} \left(\sum_{i=2}^n (\mathbf{A}^T \mathbf{A})_{i1} x_i^{(0)} \right). \quad (2.37)$$

No, (2.37) odgovara izrazu za međukorak Gauss–Seidelove metode (2.34) za sustav $\mathbf{A}^T \mathbf{A}\mathbf{x} = \mathbf{A}^T \mathbf{b}$, uvrsti li se $k = 1$, stoga se može zaključiti da su prve komponente vektora $\mathbf{x}^{(1)}$ i $\mathbf{x}_{GS}^{(1)}$ jednake.

Pretpostavimo sada da je prvih $k - 1$ komponenti vektora $\mathbf{x}^{(1)}$ i $\mathbf{x}_{GS}^{(1)}$ jednako. Na isti način kao u (2.36) (slučaj $k = 1$) dobije se,

$$x_k^{(1)} = \frac{(\mathbf{a}^k)^T \mathbf{b} - (\mathbf{a}^k)^T \mathbf{A}\mathbf{x}^{(0,k-1)}}{(\mathbf{a}^k)^T \mathbf{a}^k} - x_k^{(0)}. \quad (2.38)$$

Opet, uvaži li se da je $(\mathbf{a}^k)^T \mathbf{A}$ k -ti redak matrice $\mathbf{A}^T \mathbf{A}$, a $(\mathbf{a}^k)^T \mathbf{b}$ k -ta komponenta vektora $\mathbf{A}^T \mathbf{b}$, (2.38) se može zapisati

$$\begin{aligned} x_k^{(1)} &= \frac{1}{(\mathbf{A}^T \mathbf{A})_{kk}} \left(\sum_{i=1}^n (\mathbf{A}^T \mathbf{A})_{ik} x_i^{(0,k-1)} \right) - x_k^{(0,k-1)} \\ &= \frac{1}{(\mathbf{A}^T \mathbf{A})_{kk}} \left(\sum_{i=1}^{k-1} (\mathbf{A}^T \mathbf{A})_{ik} x_i^{(1)} + (\mathbf{A}^T \mathbf{A})_{ik} x_k^{(0)} + \sum_{i=k+1}^n (\mathbf{A}^T \mathbf{A})_{ik} x_i^{(0)} \right) - x_k^{(0)} \\ &= \frac{1}{(\mathbf{A}^T \mathbf{A})_{kk}} \left(\sum_{i=1}^{k-1} (\mathbf{A}^T \mathbf{A})_{ik} x_i^{(1)} + \sum_{i=k+1}^n (\mathbf{A}^T \mathbf{A})_{ik} x_i^{(0)} \right), \end{aligned} \quad (2.39)$$

pri čemu je korišteno svojstvo (2.35). Kao i za slučaj $k = 1$, izraz (2.39) je jednak izrazu (2.34)

za Gauss–Seidelovu metodu i time je dokazano da svi međukoraci metode DD–LS primijenjene na sustav $\mathbf{Ax} = \mathbf{b}$ daju isti rezultat kao i međukoraci Gauss–Seidelove metode primijenjeni na sustav $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$. ■

Glavna posljedica propozicije 2.20 je dokaz da je metoda DD–LS konvergentna. Tvrdnja propozicije je da je ekvivalentno rješavati sustav $\mathbf{Ax} = \mathbf{b}$ DD–LS metodom i sustav $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$ Gauss–Seidelovom metodom. Kako za Gauss–Seidelovu metodu vrijedi da je konvergentna ako je matrica sustava simetrična i pozitivno definitna [32], Gauss–Seidelova metoda za rješavanje sustava $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$ konvergira ako je $\mathbf{A}^T \mathbf{A}$ pozitivno definitna, što je slučaj kada je \mathbf{A} regularna matrica. Stoga je metoda DD–LS za rješavanje sustava $\mathbf{Ax} = \mathbf{b}$ konvergentna ako je matrica \mathbf{A} regularna.

Glavno područje primjene metode DD–LS bi mogli biti rijetki sustavi, pogotovo ako je euklidska norma stupaca matrice unaprijed poznata ili izračunata točno jednom. Ako matrica \mathbf{A} u (2.30) nije kvadratna, ali je punog ranga, tada metoda konvergira rješenju problema $\mathbf{Ax} = \mathbf{b}$ u smislu najmanjih kvadrata i može se koristiti u problemima proučavanim u [33].

Poglavlje 3

Odabir smjera spusta

U poglavljima 2.3 i 2.6 predstavljeni su algoritmi za rješavanje polinomnog sustava metodom najdublje spusta u kojima se duljina spusta određivala minimizacijom kvadrata euklidske norme reziduala (algoritam 2.4), odnosno minimizacijom max–norme reziduala (algoritam 2.8). U oba algoritma se pretpostavlja da je smjer spusta na neki način određen, ali se do sada nije ulazilo u detalje kako to napraviti.

U ovom poglavlju se opisuju različiti načini na koje se određuje smjer spusta, kao i strategije odabira smjera spusta koje se koriste za popravljavanje vjerojatnost da iterativni proces iskonvergira prema rješenju zadanog polinomnog sustava.

Razmatra se primjena standardnih smjerova spusta (smjer gradijenta, Newtonov smjer) već spomenutih u poglavlju 1.3, njihova kombiniranja, ali i primjena drugih smjerova poput varijante Gauss–Seidelova postupka odnosno način odabira smjera u kojem se smjer određuje rješavanjem zadatke linearnog programiranja.

Obzirom da se metode razlikuju prema izboru smjera spusta i funkcije koja će poslužiti za određivanje duljine spusta, sve metode razvijene u okviru ovog rada bit će imenovane na isti način:

{oznaka načina odabira smjera}- {oznaka načina određivanja duljine spusta}

a ako se želi referirati na sve metode koje koriste isti način odabira smjera, koristit će se oznaka {oznaka načina odabira smjera}*.

Na primjer, metoda koja koristi Newtonov smjer kao smjer spusta i minimizaciju kvadrata euklidske norme reziduala za određivanje duljine spusta će imati naziv NWT–E, a metoda koja isto koristi Newton smjer za smjer spusta, ali koristi minimizaciju max–norme reziduala će imati naziv NWT–M. Ako se misli referirati na obje metode koje koriste Newtonov smjer kao smjer spusta, to će se postići s NWT*.

Prije nego se krene u diskusiju odabira pojedinog smjera, još jednom će se ponoviti što je problem. Neka je s $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ zadan nelinearni sustav koji se sastoji od n jednadžbi s n nepoznanica. Tada je $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ i $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$. Jacobijeva matrica funkcije \mathbf{F} u

točki $\mathbf{x}^{(k)}$, $\nabla \mathbf{F}(\mathbf{x}^{(k)})$, definirana je po elementima s

$$(\nabla \mathbf{F}(\mathbf{x}^{(k)}))_{i,j} = \frac{\partial f_i}{\partial x_j}(\mathbf{x}^{(k)}).$$

Ako su sve funkcije f_1, \dots, f_n polinomi u jednoj ili više varijabli, onda je sustav $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ polinomni sustav. U ostatku poglavlja se podrazumijeva da se radi s polinomnim sustavom, osim ako nije eksplicitno naglašeno da se radi o općem nelinearnom sustavu.

Kako se u svim metodama najdublje spusta koristiti jedan od dva načina za određivanje duljine spusta u zadanom smjeru, umjesto da se svaki put opisuje postupak, definirane su funkcije `duljinaSpustaRSS` i `duljinaSpustaMRN` koje vraćaju duljinu spusta u zadanom smjeru obzirom na normu pomoću koje se određuje duljina spusta (algoritam 3.1).

Algoritam 3.1 Funkcije koja vraćaju duljinu spusta iz zadane točke \mathbf{x} u smjeru \mathbf{s} . Kao kriterij određivanja duljine spusta koristi se:

- kvadrat euklidske norme reziduala (RSS) (funkcija `DULJINASPUSTARSS`), odnosno
 - max–norma reziduala (MRN) (funkcija `DULJINASPUSTAMRN`).
-

function `DULJINASPUSTARSS(x,s)`

ULAZ: \mathbf{x} zadana točka, \mathbf{s} zadan smjer spusta.

Odredi prikaz polinoma $\text{RSS}_{\mathbf{x},\mathbf{s}}$ u standardnoj bazi primjenom algoritma 2.6.

Riješi jednadžbu $\text{RSS}'_{\mathbf{x},\mathbf{s}} = 0$. Neka su $\lambda_1^*, \dots, \lambda_k^*$ sva realna rješenja.

$$\lambda^* = \arg \min_{i \in \{1, \dots, k\}} \text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda_i^*) = \arg \min_{i \in \{1, \dots, k\}} \text{RSS}(\mathbf{x} + \lambda_i^* \mathbf{s}).$$

IZLAZ: duljina spusta λ^* ($\lambda^* = \arg \min_{\lambda \in \mathbb{R}} \text{RSS}_{\mathbf{x},\mathbf{s}}(\lambda)$)

end function

function `DULJINASPUSTAMRN(x,s)`

ULAZ: \mathbf{x} zadana točka, \mathbf{s} zadan smjer spusta.

Primjenom algoritma 2.7 odredi $\lambda^* = \arg \min_{\lambda \in \mathbb{R}} \text{MRN}_{\mathbf{x},\mathbf{s}}(\lambda) = \arg \min_{\lambda \in \mathbb{R}} \text{MRN}(\mathbf{x} + \lambda \mathbf{s})$.

IZLAZ: duljina spusta λ^*

end function

3.1 Standardni odabir smjera spusta

Od standardnih načina odabira smjera u ovom poglavlju se obrađuju Newtonov smjer te smjer najstrmijeg spusta. Na temelju tih smjerova formiraju se metode najdublje spusta koje kombiniraju odgovarajući smjer spusta i način određivanja duljine spusta.

U poglavlju 3.1.1 je prikazan razvoj metoda najdublje spusta kod kojih je smjer spusta Newtonov smjer, dok je u poglavlju 3.1.2 prikazan razvoj metoda najdublje spusta u kojima je smjer spusta dan kao gradijent funkcije $g_{\mathbf{F}} = \frac{1}{2} \|\mathbf{F}\|_2^2$. Konačno, u poglavlju 3.1.3 napravljena

je usporedba tih metoda na odabranim polinomnim sustavima reda 2.

3.1.1 Newtonov smjer spusta

Pri rješavanju sustava nelinearnih jednadžbi, odnosno pri traženju minimuma funkcije više varijabli, Newtonova metoda je često prvi izbor pa se i u metodi najdublje spusta odabir Newtonova smjera za smjer spusta čini logičnim.

Sama Newtonova metoda proizlazi iz aproksimacije funkcije \mathbf{F} u okolini neke točke $\mathbf{x} \in \mathbb{R}^n$ linearnom funkcijom koja se dobije iz razvoja \mathbf{F} u Taylorov red oko točke \mathbf{x} . Tada vrijedi

$$\mathbf{F}(\mathbf{x} + \mathbf{s}) \approx \mathbf{F}(\mathbf{x}) + \nabla\mathbf{F}(\mathbf{x})\mathbf{s}. \quad (3.1)$$

Uz pretpostavku da je linearna aproksimacija od \mathbf{F} dovoljno dobra, može se staviti da je lijeva strana u (3.1) jednaka nuli i odrediti \mathbf{s} kao rješenje sustava linearnih jednadžbi

$$\mathbf{0} = \mathbf{F}(\mathbf{x}) + \nabla\mathbf{F}(\mathbf{x})\mathbf{s} \quad \Leftrightarrow \quad \nabla\mathbf{F}(\mathbf{x})\mathbf{s} = -\mathbf{F}(\mathbf{x}). \quad (3.2)$$

Imajući to u vidu, dobije se osnovna varijanta Newtonove metode, tj. iterativni proces kojim se počevši od neke zadane točke $\mathbf{x}^{(0)}$ formira niz točaka $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots$ pomoću

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}, \quad k = 0, 1, \dots, \quad (3.3)$$

pri čemu se vektor $\mathbf{s}^{(k)}$ određuje kao rješenje jednadžbe

$$\nabla\mathbf{F}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)}). \quad (3.4)$$

Vektor \mathbf{s} određen s (3.4) naziva se Newtonov smjer.

Osnovna varijanta Newtonove metode ima i razne varijante [12], koje su povezane s raznim problemima koji se javljaju kod osnovne varijante. Na primjer, izračunavanje Jacobijeve matrice u svakom koraku može biti računski jako zahtjevno, pogotovo ako Jacobijeva matrica za problem nije unaprijed zadana već je treba na neki način izračunati. U tom slučaju se može koristiti pojednostavljena Newtonova metoda u kojoj se samo za prvu točku izračuna Jacobijeva matrica čime je iterativni proces definiran s

$$\begin{aligned} \nabla\mathbf{F}(\mathbf{x}^{(0)})\bar{\mathbf{s}}^{(k)} &= -\mathbf{F}(\mathbf{x}^{(k)}), \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \bar{\mathbf{s}}^{(k)}, \quad k = 0, 1, \dots \end{aligned}$$

Prednost ove metode je što se izbjegava izračunavanje Jacobijeve matrice u svakom koraku procesa. S druge strane, ako se Jacobijeve matrice u točkama $\mathbf{x}^{(k)}$ dosta razlikuju od početne,

onda će iterativni proces konvergirati sporije ili uopće neće konvergirati. Da bi se takva situacija izbjegla, napravljene su i varijante pojednostavljene Newtonove metode u kojima se Jacobijeva matrica računa svakih M koraka.

U svojoj osnovnoj varijanti, Newtonova metoda je lokalno konvergentna u smislu da ako je početna točka dovoljno blizu rješenja onda će uz određene pretpostavke (teorem 5.2.1 u [11], najvažnija pretpostavka je da Jacobijeva matrica nije singularna u rješenju) iterativni postupak (3.3)–(3.4) konvergirati rješenju i konvergencija će biti kvadratična. Problem je što granice područja u kojem će metoda konvergirati u praksi nije moguće odrediti. Stoga su napravljene varijante Newtonove metode koje bi trebale konvergirati čak i u slučaju da je početna točka $\mathbf{x}^{(0)}$ relativno “loša”. U grubim crtama, osnovna razlika u odnosu na osnovni algoritam je što se ne izvodi cijeli korak u Newtonovom smjeru u 3.3, već se napravi smanjeni korak

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}, \quad 0 < \lambda \leq 1.$$

Glavni razlog za korištenje parametra smanjenja λ leži u činjenici da se Newtonova metoda izvodi iz linearne aproksimacije od \mathbf{F} oko zadane točke \mathbf{x} . No, ako \mathbf{x} nije blizu rješenja, onda ni $\mathbf{x} + \mathbf{s}$ nije blizu \mathbf{x} pa je upitno koliko je dobra pretpostavka o aproksimaciji (3.1). Stoga se umjesto punog koraka, izvodi smanjeni korak. Popularna strategija za odabir duljine koraka je Armijovo pravilo [12, 34].

No, metoda najdublje spusta ide s drugom pretpostavkom: pronađi točku na zadanom smjeru spusta bez obzira gdje se nalazi. Metode NWT-E i NWT-M su definirane tako da koriste Newtonov smjer kao smjer spusta. Od svih metoda koje su razvijene za potrebe ovog rada kod njih je napravljeno najmanje dodatnih radnji da bi ih se moglo koristiti. Jedan korak metode NWT-E je opisan u algoritmu 3.2. Algoritam 3.2, kao i svi budući algoritmi koji za određivanje duljine spusta koriste minimizaciju kvadrata euklidske norme reziduala, zamjenjuje korake 4–7 u općem algoritmu za rješavanje polinomnog sustava (algoritam 2.4).

Algoritam 3.2 Jedan korak metode najdublje spusta NWT-E za rješavanje polinomnog sustava.

Pretpostavke: Zadana početna točka $\mathbf{x}^{(k)}$.

- 1: Odredi \mathbf{s}^N kao Newtonov smjer, $\mathbf{s}^N = -(\nabla \mathbf{F}(\mathbf{x}^{(k)}))^{-1} \mathbf{F}(\mathbf{x}^{(k)})$.
 - 2: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSSTARSS}(\mathbf{x}^{(k)}, \mathbf{s}^N)$ (algoritam 3.1).
 - 3: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^N$.
-

Analogno, jedan korak metode NWT-M je opisan u algoritmu 3.3. Algoritam 3.3, kao i ostali algoritmi koji za određivanje duljine spusta koriste minimizaciju max-norme, zamjenjuje korake 4 i 5 u općem algoritmu za rješavanje polinomnog sustava (2.8).

Algoritam 3.3 Jedan korak metode najdublje spusta NWT-M za rješavanje polinomnog sustava.

Pretpostavke: Zadana početna točka $\mathbf{x}^{(k)}$.

- 1: Odredi \mathbf{s}^N kao Newtonov smjer, $\mathbf{s}^N = -(\nabla \mathbf{F}(\mathbf{x}^{(k)}))^{-1} \mathbf{F}(\mathbf{x}^{(k)})$.
- 2: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSTAMRN}(\mathbf{x}^{(k)}, \mathbf{s}^N)$ (algoritam 3.1).
- 3: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^N$.

3.1.2 Najstrmiji smjer spusta

Drugi standardan smjer spusta se dobije kada se problem traženja rješenja sustava jednadžbi s više varijabli zamijeni s problemom traženja najmanje vrijednosti realne funkcije više varijabli.

Opet, neka je $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ polinomni sustav. Definiramo nivo-funkciju $g_{\mathbf{F}} : \mathbb{R}^n \rightarrow \mathbb{R}$ s

$$g_{\mathbf{F}}(\mathbf{x}) = \frac{1}{2} \|\mathbf{F}(\mathbf{x})\|_2^2 = \frac{1}{2} \mathbf{F}(\mathbf{x})^T \mathbf{F}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n f_i^2(\mathbf{x}).$$

Očito vrijedi da je $g_{\mathbf{F}}(\mathbf{x}) \geq 0$ za svaki $\mathbf{x} \in \mathbb{R}^n$, a jednakost vrijedi samo za one \mathbf{x}^* koji su rješenja sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$:

$$g_{\mathbf{F}}(\mathbf{x}^*) = 0 \quad \Leftrightarrow \quad \mathbf{F}(\mathbf{x}^*) = \mathbf{0}.$$

Problem rješavanja nelinearnog sustava jednadžbi $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ ekvivalentan je problemu traženja minimuma funkcije $g_{\mathbf{F}}$.

Problem traženja minimuma funkcije više varijabli se može pristupiti na razne načine, a u ovom slučaju koristi se metoda najstrmijeg spusta, odnosno koristit će se iterativni postupak u kojem je smjer spusta definiran s

$$\mathbf{s}^{(k)} = -\nabla g_{\mathbf{F}}(\mathbf{x}^{(k)}).$$

Budući da je

$$\frac{\partial g_{\mathbf{F}}}{\partial x_i} = \frac{1}{2} \cdot \frac{\partial}{\partial x_i} \left(\sum_{j=1}^n f_j^2 \right) = \frac{1}{2} \cdot \sum_{j=1}^n \frac{\partial f_j^2}{\partial x_i} = \sum_{j=1}^n \frac{\partial f_j}{\partial x_i} f_j,$$

slijedi da se $\nabla g_{\mathbf{F}}(\mathbf{x})$ može zapisati kao

$$\begin{aligned} \nabla g_{\mathbf{F}}(\mathbf{x}) &= \begin{bmatrix} \frac{\partial g_{\mathbf{F}}}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial g_{\mathbf{F}}}{\partial x_n}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n \frac{\partial f_j}{\partial x_1}(\mathbf{x}) f_j(\mathbf{x}) \\ \vdots \\ \sum_{j=1}^n \frac{\partial f_j}{\partial x_n}(\mathbf{x}) f_j(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial f_n}{\partial x_1}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_n}(\mathbf{x}) & \dots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix} \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} \\ &= (\nabla \mathbf{F}(\mathbf{x}))^T \mathbf{F}(\mathbf{x}). \end{aligned}$$

Smjer spusta \mathbf{s} određen sa

$$\mathbf{s} = -\nabla g_{\mathbf{F}}(\mathbf{x}) = -(\nabla \mathbf{F}(\mathbf{x}))^T \mathbf{F}(\mathbf{x}) \quad (3.5)$$

naziva se smjer najstrmijeg spusta, a na njega će se referirati i kao smjer spusta određen gradijentom od $g_{\mathbf{F}}$.

Kako je vidljivo iz prethodnog računa, za određivanje smjera najstrmijeg spusta nije potrebno rješavati nikakav sustav, već se smjer dobije kao umnožak transponirane Jacobijeve matrice funkcije \mathbf{F} i vrijednosti funkcije \mathbf{F} u zadanoj točki \mathbf{F} .

Na žalost, metoda najstrmijeg spusta spada u metode koje jako sporo konvergiraju ([12, 22]). Numerički eksperimenti rađeni za metodu najdublje spusta u kojoj je smjer spusta \mathbf{s} definiran kao smjer najstrmijeg spusta, pokazali su da metoda vrlo sporo konvergira, pogotovo ako je \mathbf{x} točka u okolini nekog rješenja. Stoga je metoda prilagođena tako da u slučaju kada silazak u smjeru najstrmijeg spusta ne postigne željeno smanjenje, za vektor spusta se uzme Newtonov smjer.

Metode najdublje spusta koje koriste gradijent nivo-funkcije $g_{\mathbf{F}}$ (3.5) kao smjera spusta zovu se GN-E i GN-M, ovisno o tome koju funkciju koriste za određivanje duljine spusta.

Jedan korak metode GN-E prikazan je u algoritmu 3.4, a cijela metoda se dobije kada se algoritam 3.4 koristi umjesto koraka 4–7 u općem algoritmu za rješavanje polinomnih sustava (algoritam 2.4).

Parametar ε_D kontrolira kada treba odustati od smjera najstrmijeg spusta: ako je omjer između nove i stare vrijednosti euklidske norme reziduala veći od $1 - \varepsilon_D$, onda treba pokušati s Newtonovim smjerom. Parametar ε_{NWT} služi za izbjegavanje problema spore konvergencije kada su točke $\mathbf{x}^{(k)}$ blizu rješenja, pri čemu se koristi pretpostavka da ako je norma od $\mathbf{F}(\mathbf{x}^{(k)})$ mala da je onda $\mathbf{x}^{(k)}$ blizu rješenja. Kad nastupi slučaj $\|\mathbf{F}(\mathbf{x}^{(k)})\| < \varepsilon_{NWT}$, odmah se kao smjer spusta koristi Newtonov smjer. Vrijednosti parametara ε_D i ε_{NWT} u programu PolySolver su $\varepsilon_D = 10^{-8}$ i $\varepsilon_{NWT} = 10^{-3}$.

Jedan korak metode GN-M prikazan je u algoritmu 3.5, a cijela metoda se dobije kada se koristi algoritam 3.5 umjesto koraka 5–6 u općem algoritmu za rješavanje polinomnih sustava korištenjem max-norme reziduala (algoritam 2.8). Kako se može vidjeti, metode GN-E i GN-M su konceptijski iste, jedina razlika je upotreba različitih vektorskih normi u koracima 1 i 6, te različite funkcije za određivanje vektora spusta (koraci 4, 9 i 16).

3.1.3 Usporedba NWT* i GN* metoda

U tablici 3.1 prikazana je uspješnost metoda najdublje spusta pri rješavanju odabranih polinomnih sustava reda 2. Da bi se izračunala uspješnost određene metode na nekom problemu, iterativni postupak se ponavlja veliki broj puta s različitim početnim točkama, a uspješnost

Algoritam 3.4 Jedan korak metode najdublje spusta GN-E za rješavanje polinomnog sustava.

Pretpostavke: Zadana početna točka $\mathbf{x}^{(k)}$.

Parametar ε_D određuje je li postignut dovoljno veliki napredak. Postignut je dobar napredak ako je $1 - \|\mathbf{F}(\mathbf{x}^{(k+1)})\|_2 / \|\mathbf{F}(\mathbf{x}^{(k)})\|_2 \geq \varepsilon_D$.

Parametar ε_{NWT} služi za prepoznavanje blizine rješenja. Ako je vrijednost $\|\mathbf{F}(\mathbf{x}^{(k)})\|_2$ manja od ε_{NWT} , onda se koristi Newtonov smjer.

- 1: $R_{OLD} = \|\mathbf{F}(\mathbf{x}^{(k)})\|_2$.
 - 2: **if** $R_{OLD} > \varepsilon_{NWT}$ **then** ▷ Nismo blizu rješenja.
 - 3: Odredi $\mathbf{s}^G = -(\nabla \mathbf{F}(\mathbf{x}^{(k)}))^T \mathbf{F}(\mathbf{x}^{(k)})$, kao smjer najstrmijeg spusta.
 - 4: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSSTARSS}(\mathbf{x}^{(k)}, \mathbf{s}^G)$ (algoritam 3.1).
 - 5: $\mathbf{x}^G = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^G$.
 - 6: $R_{NEW}^G = \|\mathbf{F}(\mathbf{x}^G)\|_2$.
 - 7: **if** $R_{NEW}^G / R_{OLD} > 1 - \varepsilon_D$ **then** ▷ Nije postignut dobar napredak
 - 8: Odredi \mathbf{s}^N kao Newtonov smjer, $\mathbf{s}^N = -(\nabla \mathbf{F}(\mathbf{x}^{(k)}))^{-1} \mathbf{F}(\mathbf{x}^{(k)})$.
 - 9: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSSTARSS}(\mathbf{x}^{(k)}, \mathbf{s}^N)$.
 - 10: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^N$.
 - 11: **else** ▷ Postignut dobar napredak
 - 12: $\mathbf{x}^{(k+1)} = \mathbf{x}^G$.
 - 13: **end if**
 - 14: **else** ▷ Blizu je rješenje sustava. Koristi Newtonov smjer.
 - 15: Odredi \mathbf{s}^N kao Newtonov smjer, $\mathbf{s}^N = -(\nabla \mathbf{F}(\mathbf{x}^{(k)}))^{-1} \mathbf{F}(\mathbf{x}^{(k)})$.
 - 16: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSSTARSS}(\mathbf{x}^{(k)}, \mathbf{s}^N)$ (algoritam 3.1).
 - 17: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^N$.
 - 18: **end if**
-

se računa kao postotak iterativnih postupaka koji su uspješno završili (konvergirali su prema rješenju polinomnog sustava). Detalji numeričkog eksperimenta su opisani u poglavlju 4.2. U primjerima navedenim u tablici, za svaku kombinaciju (metoda, problem) je napravljeno 10.000 različitih pokusa.

Podatci iz tablice pokazuju neka zanimljiva svojstva. Usporedi li se uspješnost metoda NWT-E i GN-E, metoda NWT-E je znatno uspješnija od metode GN-E na pet primjera, a znatno neuspješnija na dva. Kod usporedbe metoda NWT-M i GN-M, NWT-M je znatno uspješniji na jednom primjeru, podjednaki su na tri primjera, a GN-M je uspješnija na tri primjera.

Relativan neuspjeh metoda NWT-E i NWT-M na primjeru freudenstein-roth je posljedica činjenice da je riječ o primjeru konstruiranom tako da Newtonova metoda ima poteškoće [35]. Na primjeru rosenbrock sve metode imaju slabu uspješnost, ali i ovdje je riječ o čuvenom test problemu iz područja numeričke optimizacije, napravljenom tako da bude težak za rješavanje.

Algoritam 3.5 Jedan korak GN-M metode najdubljega spusta za rješavanje polinomnog sustava.

Pretpostavke: Zadana početna točka $\mathbf{x}^{(k)}$.

Parametri ε_D i ε_{NWT} isti kao u algoritmu 3.4.

- 1: $R_{OLD} = \|\mathbf{F}(\mathbf{x}^{(k)})\|_\infty$.
 - 2: **if** $R_{OLD} > \varepsilon_{NWT}$ **then** ▷ Nismo blizu rješenja.
 - 3: Odredi $\mathbf{s}^G = -(\nabla\mathbf{F}(\mathbf{x}^{(k)}))^T \mathbf{F}(\mathbf{x}^{(k)})$, kao smjer najstrmijeg spusta.
 - 4: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSTAMRN}(\mathbf{x}^{(k)}, \mathbf{s}^G)$ (algoritam 3.1).
 - 5: $\mathbf{x}^G = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^G$.
 - 6: $R_{NEW}^G = \|\mathbf{F}(\mathbf{x}^G)\|_\infty$.
 - 7: **if** $R_{NEW}^G / R_{OLD} > 1 - \varepsilon_D$ **then** ▷ Nije postignut dobar napredak
 - 8: Odredi \mathbf{s}^N kao Newtonov smjer, $\mathbf{s}^N = -(\nabla\mathbf{F}(\mathbf{x}^{(k)}))^{-1} \mathbf{F}(\mathbf{x}^{(k)})$.
 - 9: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSTAMRN}(\mathbf{x}^{(k)}, \mathbf{s}^N)$.
 - 10: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^N$.
 - 11: **else** ▷ Postignut dobar napredak
 - 12: $\mathbf{x}^{(k+1)} = \mathbf{x}^G$.
 - 13: **end if**
 - 14: **else** ▷ Blizu je rješenje sustava. Koristi Newtonov smjer.
 - 15: Odredi \mathbf{s}^N kao Newtonov smjer, $\mathbf{s}^N = -(\nabla\mathbf{F}(\mathbf{x}^{(k)}))^{-1} \mathbf{F}(\mathbf{x}^{(k)})$.
 - 16: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSTAMRN}(\mathbf{x}^{(k)}, \mathbf{s}^N)$ (algoritam 3.1).
 - 17: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^N$.
 - 18: **end if**
-

Tablica 3.1: Uspješnost primjene metoda NWT-E, NWT-M, GN-E i GN-M na odabranim primjerima polinomnih sustava reda 2.

	NWT-E	NWT-M	GN-E	GN-M
freudenstein-roth	23,4	23,1	69,9	99,5
leary	64,9	40,1	56,5	68,6
mickey	63,2	63,2	96,1	100,0
morgan	61,0	61,6	22,5	62,2
reimer2	75,4	55,6	61,9	58,1
rosenbrock	13,1	13,6	1,5	16,0
sendra	77,2	72,9	45,5	62,7

Također ako se usporede metode s istim smjerom spusta, a različitim načinom određivanja duljine spusta, kod Newtonova smjera se boljom pokazuje metoda NWT-E koja je uspješnija od NWT-M u tri slučaja, a u ostalima su podjednaki. S druge strane kod metoda kojima spust ima smjer najstrmijeg spusta bitno boljom se pokazuje metoda GN-M. Metoda GN-M je uspješnija od GN-E u pet slučajeva, a u dva slučaja su uspješnosti podjednake.

Generalno upada u oči činjenica da metoda GN-E ima na tri primjera znatno lošiji rezultat od svih ostalih metoda. Na slici 3.1 je prikazan rezultat rada programa `Polysolver` za problem freudenstein-roth, početnu točku $\mathbf{x}^{(0)} = (-8, -1)$ i metode GN-E (slika (a)) i GN-M (slika (b)). Značenje stupaca* u izlazu:

- RB - redni broj iteracije.
- Način - način odabira smjera spusta i određivanja duljine spusta. Na slici 3.1 su prikazana tri načina, Grd2, GrdINF i NwtINF. Načini Grd* koriste smjer gradijenta na nivo funkciju g_F kao smjer spusta, dok načini Nwt* za smjer spusta koriste Newtonov smjer. Načini *2 koriste za određivanje duljine spusta minimizaciju euklidske norme reziduala, dok se kod načina *INF za određivanje duljine spusta koristi minimizacija max-norme reziduala.
- Tocka - točka iteracije, $\mathbf{x}^{(k)}$, gdje je $k = \text{RB}$.
- Max norm - max norma reziduala u $\mathbf{x}^{(k)}$, $\|\mathbf{F}(\mathbf{x}^{(k)})\|_\infty$.
- L2 norm - euklidska norma reziduala u $\mathbf{x}^{(k)}$, $\|\mathbf{F}(\mathbf{x}^{(k)})\|_2$.

Na slici 3.1.a prikazano je prvih pet iteracija metode GN-E. Kako se vidi sa slike, cijelo vrijeme se za spust koristi smjer gradijenta od g_F . Nakon prva tri koraka smanjenja euklidske norme su relativno mala (oko 5%), ali nedovoljno da se proba s Newtonovim smjerom. S druge strane, sa slike 3.1.b je vidljivo da kod metode GN-M, već kod druge iteracije spust u smjeru gradijenta od g_F nije postigao nikakav rezultat pa je trebalo koristiti Newtonov smjer.

Na prvi pogled, činjenica da već na drugoj iteraciji metode GN-M pomak u smjeru gradijenta od g_F (najstrmiji spust) ne daje nikakav rezultat može izgledati čudno, ali treba uzeti u obzir da GN-M metoda koristi max-normu, a smjer najstrmijeg spusta daje najveće smanjenje u euklidskoj normi. Zato se može zaključiti da je metoda GN-E koja je direktni ekvivalent metode najbržeg spusta, dok će kod GN-M puno češće dolaziti do situacije da nema napretka u smjeru najstrmijeg spusta, već treba koristiti Newtonov smjer. Upravo kombiniranje smjerova spusta povećava uspješnost metode GN-M.

3.2 Kombiniranje standardnih smjerova spusta

Usporedba rezultata rada metoda NWT-E i GN-E, te NWT-M i GN-M na odabranim polinomnim sustavima reda 2 (tablica 3.1) pokazuje da postoje problemi u kojima metode bazirane na Newtonovu smjeru imaju veću uspješnost od metoda zasnovanih na smjeru najstrmijeg spusta, ali i

*U samom programu se prikazuje još nekoliko stupaca za svaku iteraciju. Više o tome u prilogu 5.

RB	Način	Točka	Max norm	L2 norm
0		-8,00000; -1,00000	2,300E+01	2,642E+01
1	Grd2	-7,9433; -1,7777	1,451E+01	1,506E+01
2	Grd2	9,4684; -0,5088	1,228E+01	1,233E+01
3	Grd2	9,1452; 3,9257	4,850E+00	4,972E+00
4	Grd2	8,7466; 3,8967	4,706E+00	4,728E+00
5	Grd2	8,7439; 3,9332	4,381E+00	4,493E+00

(a)

RB	Način	Točka	Max norm	L2 norm
0		-8,00000; -1,00000	2,300E+01	2,642E+01
1	GrdINF	-7,9238; -2,0459	1,266E+01	1,790E+01
2	NwtINF	6,9657; -1,3115	7,445E+00	8,552E+00
3	GrdINF	6,7983; 4,0000	1,798E+00	2,543E+00
4	NwtINF	5,0000; 4,0000	0,000E+00	0,000E+00

(b)

Slika 3.1: Prvih nekoliko iteracija metoda GN-E (slika (a)) i GN-M (slika (b)) na problemu freudenstein-roth za početnu točku $\mathbf{x}^{(0)} = (-8, -1)$

da postoje problemi kod kojih su uspješnije metode zasnovane na smjeru najstrmijeg spusta. Stoga se logičnom čini ideja da se smjer spusta odabere između Newtonova smjera i smjera najstrmijeg silaska kao onaj smjer koji daje veće smanjenje reziduala.

U algoritmu 3.6 je prikazana realizacija metode najdublje spusta BGN-E[†], u kojoj se duljina spusta određuje minimizacijom kvadrata euklidske norme reziduala, a za smjer spusta se uzima ili smjer najstrmijeg spusta ili Newtonov smjer.

Algoritam 3.6 Jedan korak metode najdublje spusta BGN-E za rješavanje polinomnog sustava.

Pretpostavke: Zadana početna točka $\mathbf{x}^{(k)}$.

- 1: Odredi $\mathbf{s}^G = -(\nabla \mathbf{F}(\mathbf{x}^{(k)}))^T \mathbf{F}(\mathbf{x}^{(k)})$, kao smjer najstrmijeg spusta.
 - 2: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSTARSS}(\mathbf{x}^{(k)}, \mathbf{s}^G)$ (algoritam 3.1).
 - 3: $\mathbf{x}^G = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^G$.
 - 4: Odredi \mathbf{s}^N kao Newtonov smjer, $\mathbf{s}^N = -(\nabla \mathbf{F}(\mathbf{x}^{(k)}))^{-1} \mathbf{F}(\mathbf{x}^{(k)})$.
 - 5: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSTARSS}(\mathbf{x}^{(k)}, \mathbf{s}^N)$.
 - 6: $\mathbf{x}^N = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^N$.
 - 7: **if** $\|\mathbf{F}(\mathbf{x}^N)\|_2 < \|\mathbf{F}(\mathbf{x}^G)\|_2$ **then**
 - 8: $\mathbf{x}^{(k+1)} = \mathbf{x}^N$
 - 9: **else**
 - 10: $\mathbf{x}^{(k+1)} = \mathbf{x}^G$.
 - 11: **end if**
-

Realizacija metode najdublje spusta BGN-M u kojoj se duljina spusta određuje minimizacijom max-norme reziduala, a za smjer spusta se uzima ili smjer najstrmijeg spusta ili Newtonov smjer, prikazana je u algoritmu 3.7.

U tablici 3.2 rezultati iz tablice 3.1 prošireni su s rezultatima uspješnosti za metode BGN-E i BGN-M. Iz tablice se može očitati da je u većini slučajeva najveću uspješnost pokazala metoda

[†]Oznaka BGN je nastala kao akronim izraza Bolji, Gradijent, Newton, pri čemu se pojam “Gradijent” odnosi na odabir smjera kao gradijenta od \mathbf{g}_F , a “Newton” na Newtonov odabir smjera.

Algoritam 3.7 Jedan korak metode najdublje spusta BGN-M za rješavanje polinomnog sustava.

Pretpostavke: Zadana početna točka $\mathbf{x}^{(k)}$.

- 1: Odredi $\mathbf{s}^G = -(\nabla \mathbf{F}(\mathbf{x}^{(k)}))^T \mathbf{F}(\mathbf{x}^{(k)})$, kao smjer najstrmijeg spusta.
 - 2: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSTAMRN}(\mathbf{x}^{(k)}, \mathbf{s}^G)$ (algoritam 3.1).
 - 3: $\mathbf{x}^G = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^G$.
 - 4: Odredi \mathbf{s}^N kao Newtonov smjer, $\mathbf{s}^N = -(\nabla \mathbf{F}(\mathbf{x}^{(k)}))^{-1} \mathbf{F}(\mathbf{x}^{(k)})$.
 - 5: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSTAMRN}(\mathbf{x}^{(k)}, \mathbf{s}^N)$.
 - 6: $\mathbf{x}^N = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^N$.
 - 7: **if** $\|\mathbf{F}(\mathbf{x}^N)\|_\infty < \|\mathbf{F}(\mathbf{x}^G)\|_\infty$ **then**
 - 8: $\mathbf{x}^{(k+1)} = \mathbf{x}^N$
 - 9: **else**
 - 10: $\mathbf{x}^{(k+1)} = \mathbf{x}^G$.
 - 11: **end if**
-

Tablica 3.2: Uspješnost primjene metoda NWT-E, NWT-M, GN-E, GN-M, BGN-E i BGN-M na odabranim primjerima polinomnih sustava reda 2.

	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M
freudenstein-roth	23,4	23,1	69,9	99,5	87,7	83,5
leary	64,9	40,1	56,5	68,6	80,5	70,3
mickey	63,2	63,2	96,1	100,0	100,0	100,0
morgan	61,0	61,6	22,5	62,2	61,5	61,4
reimer2	75,4	55,6	61,9	58,1	95,1	67,2
rosenbrock	13,1	13,6	1,5	16,0	66,0	17,2
sendra	77,2	72,9	45,5	62,7	84,4	79,6

BGN-E, čime je opravdano uvođenje ove vrste metoda. Pogotovo je drastična razlika u uspješnosti metoda primjetna na problemima rosenbrock i reimer2. Zanimljivo je primijetiti da je kod metoda GN-M i BGN-M teško reći koja je bolja, jer na četiri primjera imaju podjednaku uspješnost, u dva slučaja je uspješnija metoda BGN-M, a u jednom GN-M. Razlog tome već je spomenut u poglavlju 3.1.3: metoda GN-M u dosta slučajeva koristi različite smjerove spusta čime popravljaju uspješnost.

Što se tiče računalne zahtjevnosti metoda BGN* u odnosu na metode NWT* i GN*, jasno je da je jedan korak BGN* metoda sporiji od jednog koraka NWT* jer ima jedno određivanje duljine spusta više, a samo dodatno određivanje smjera \mathbf{s}^G nije pretjerano skupo, jer je Jacobijeva matrica

već izračunata. Što se tiče odnosa metoda BGN* i GN*, u većini slučajeva će metode BGN* biti sporije zbog dodatnog izračuna duljine spusta te činjenice da za određivanje Newtonova smjera treba riješiti linearni sustav. No, razlika u brzini između metoda BGN-M i GN-M može biti vrlo mala jer kod metode GN-M dosta često može nastupiti slučaj u kojem je u jednom koraku iteracije potrebno izračunati spust i u smjeru najstrmijeg spusta i u Newtonovom smjeru, odnosno potrebno je računati dvije duljine spusta u određenim smjerovima.

Metode BGN-E i BGN-M bi se mogle znatno ubrzati paralelizacijom tako da se koraci 1–3, odnosno 4–6, iz algoritma 3.6 (3.7) provedu svaki u svojoj niti. Time bi se postiglo značajno ubrzanje jer preostali dijelovi algoritma nisu računski zahtjevni, a metode bi se mogle proširiti i s još nekim smjerom pretraživanja.

3.3 Gauss–Seidelov smjer

Grupa metoda u kojima se veći problem svodi na veći broj manjih problema je jedna od najčešće korištenih metoda u matematici. U ovom slučaju, cilj je da se, analogno Gauss–Seidelovoj metodi za iterativno rješavanje linearnih sustava, kod određivanja smjera spusta pokuša izbjeći rješavanje cijelog sustava linearnih jednadžbi na način da se riješi niz jednostavnih jednadžbi u jednoj varijabli.

U rješavanju sustava nelinearnih jednadžbi nelinearna Gauss–Seidelova metoda se spominje u [36], u smislu da se u $k + 1$ koraku iterativnog postupka i -ta komponenta točke $\mathbf{x}^{(k+1)}$ određuje rješavanjem problema

$$f_i(x_1^{(k+1)}, x_{i-1}^{(k+1)}, x, x_{i+1}^{(k)}, \dots, x_n^{(k)}) = 0, \quad (3.6)$$

tj. rješavanjem problema u samo jednoj dimenziji. Štoviše, prema [36], može se staviti da je

$$x_i^{(k+1)} = x_i^{(k)} + \omega_k(x_i^* - x_i^{(k)}), \quad (3.7)$$

gdje je $\omega \in \langle 0, 1 \rangle$, a x_i^* rješenje jednadžbe (3.6) (u [36] se metoda koja koristi (3.7) naziva nelinearna SOR metoda). Parametar ω se bira tako da se postigne smanjenje norme reziduala. Međutim, kako rješavanje problema (3.6) može biti općenito zahtjevno, umjesto x_i^* koristi se aproksimacija koja se dobije primjenom m koraka jednodimenzionalne Newtonove metode, a pri tome je posebno istaknut slučaj u kojem je $m = 1$.

Pristup koji se upotrebljava u ovom radu koristi činjenicu da se rješavaju polinomni sustavi. U tom slučaju je svaka od funkcija f_i polinom pa se rješavanje jednadžbe (3.6) svodi na traženje nultočka polinoma, a “prava” nultočka x_i^* se odabire tako da norma reziduala funkcije \mathbf{F} bude najmanja. Također, promijenjen je način na koji se odabire kako i kada će se odrediti i -ta komponenta nove točke $x^{(k+1)}$, pa se nova komponenta ne određuje kao kod [36], već je redosljed jednadžbi i varijabli postavljen tako da se napravi najveće smanjenje norme reziduala.

Radi jednostavnosti, potrebno je uvesti neke oznake koje će pojednostaviti daljnji postupak. Pretpostavimo da je zadana početna točka $\mathbf{x} = (x_1, \dots, x_n)$. U prvom koraku će se dobiti točka \mathbf{x}^1 koja se razlikuje od $\mathbf{x}^0 = \mathbf{x}$ samo na nekom mjestu j_1 ,

$$\mathbf{x}^1 - \mathbf{x}^0 = (0, \dots, 0, x_{j_1}^1 - x_{j_1}, 0, \dots, 0).$$

Dalje, u drugom koraku će se dobiti točka \mathbf{x}^2 , koja se od točke \mathbf{x}^1 razlikuje u samo jednoj komponenti, $j_2 \neq j_1$,

$$\mathbf{x}^2 - \mathbf{x}^1 = (0, \dots, 0, x_{j_2}^2 - x_{j_2}^1, 0, \dots, 0).$$

Analogno, u m -tom koraku, $m \leq n$, dobit će se točka \mathbf{x}^m koja se od \mathbf{x}^{m-1} razlikuje samo u komponenti j_m , a $j_m \notin J_{m-1} = \{j_1, \dots, j_{m-1}\}$. Uvjet $j_k \notin J_{m-1}$ implicira da se točke \mathbf{x}^m i \mathbf{x}^0 razlikuju u m komponenti i to su komponente j_1, \dots, j_m . Postupak se zaustavlja nakon n koraka, a točka $\mathbf{x}^* = \mathbf{x}^n$ se koristi za određivanje smjera spusta \mathbf{s}^{GS} , definiranog sa $\mathbf{s}^{GS} = \mathbf{x}^* - \mathbf{x}$.

Neka je $\mathbf{x}^0 = \mathbf{x}$ zadana početna točka. Pretpostavimo da treba odrediti točku \mathbf{x}^1 . Za određivanje točke \mathbf{x}^1 koristit će se funkcija f_b , pri čemu je b indeks najveće komponente vektora $|\mathbf{F}(\mathbf{x}^0)|$, definiran sa

$$b = \arg \max_{i \in \{1, \dots, n\}} |f_i(\mathbf{x}^0)|. \quad (3.8)$$

Nakon što je određena funkcija pomoću koje će se odrediti \mathbf{x}^1 , potrebno je naći i komponentu j_1 koja će se promijeniti. Komponenta j_1 određuje se kao ona komponenta gradijenta funkcije f_b koja ima najveću vrijednost u točki \mathbf{x} . Formalno,

$$j_1 = \arg \max_{j \in \{1, \dots, n\}} \left| \frac{\partial f_b}{\partial x_j}(\mathbf{x}^0) \right|. \quad (3.9)$$

Za točku \mathbf{x} i indeks j_1 se definira vektor $\mathbf{y}^{j_1} = \mathbf{x}^0 - x_{j_1} \mathbf{e}_{j_1}$ pomoću kojeg se definira realni polinom jedne varijable $f^*(t) = f_b(\mathbf{y}^{j_1} + t \mathbf{e}_{j_1})$. Koeficijenti polinoma f^* se dobiju na način opisan u propoziciji 2.2.

U kasnijim koracima, polinom f^* može biti konstanta, ali ako se to dogodi u prvom koraku, onda bi to značilo da za sve $i = 1, \dots, n$ vrijedi $\frac{\partial f_b}{\partial x_i}(\mathbf{x}^0) = 0$, odnosno da je $f_b(\mathbf{x}) = c \in \mathbb{R}$ konstanta, što je nemoguće.

Pretpostavimo da jednačina $f^*(t) = 0$ ima realna rješenja t_1, \dots, t_s . Definiramo t^* kao onu nultočku od f^* za koju je vrijednost $\|\mathbf{F}(\mathbf{y}^{j_1} + t \mathbf{e}_{j_1})\|_\infty$ najmanja,

$$t^* = \arg \min_{t \in \{t_1, \dots, t_s\}} \|\mathbf{F}(\mathbf{y}^{j_1} + t \mathbf{e}_{j_1})\|_\infty. \quad (3.10)$$

S druge strane, ako jednačina $f^*(t) = 0$ nema realnih rješenja, onda $\deg f^*$ treba biti paran broj veći od nule. No, tada je derivacija polinoma f^* polinom neparnog stupnja i sigurno ima realnu

nultočku. Zato u slučaju kada f^* nema realne nultočke, treba riješiti jednadžbu $(f^*)'(t) = 0$ i među njenim rješenjima t'_1, \dots, t'_r odabrati ono za koje će vrijednost $\|\mathbf{F}(\mathbf{y}^{j_1} + t \mathbf{e}_{j_1})\|_\infty$ biti najmanja (prema (3.10)). Nova točka \mathbf{x}^1 je definirana s $\mathbf{x}^1 = \mathbf{y}^{j_1} + t^* \mathbf{e}_{j_1}$.

Za određivanje sljedećih točaka $\mathbf{x}^2, \dots, \mathbf{x}^n$ postupak je u principu isti, jedino treba pripaziti kod odabira komponente po kojoj će se kasnije rješavati jednadžba. Naime, nova se komponenta bira samo među neiskorištenima.

Prvo treba odrediti funkciju f_b . Kao i u (3.8), b se odredi kao indeks one komponente vektora $|\mathbf{F}(\mathbf{x}^{m-1})|$ koja ima najveću vrijednost,

$$b = \arg \max_{i \in \{1, \dots, n\}} |f_i(\mathbf{x}^{m-1})|. \quad (3.11)$$

Nakon toga određuje se komponenta j_m . Kako je već spomenuto, komponenta se bira među komponentama koje nisu do sada korištene. Postupak određivanja j_k se može zapisati kao

$$j_k = \arg \max_{j \in \{1, \dots, n\} \setminus J_{m-1}} \left| \frac{\partial f_b}{\partial x_j}(\mathbf{x}^{k-1}) \right|, \quad (3.12)$$

pri čemu je $J_{m-1} = \{j_1, \dots, j_{m-1}\}$ skup komponenti koje su korištene u prethodnih $m - 1$ koraka.

Ako je $\frac{\partial f_b}{\partial x_j}(\mathbf{x}^0) = 0$, onda funkcija f_b nema pomaka u smjeru niti jedne neiskorištene komponente, odnosno vektor f^* će biti konstanta. U ovom slučaju proglašava se da je nova točka \mathbf{x}^* iz koje će se odrediti smjer spusta jednaka \mathbf{x}^{m-1} i prelazi se na određivanje duljine spusta. Druga moguća rješenja su potražiti neku drugu funkciju f_b pa ponoviti postupak odabira komponente ili tražiti istovremeno i b i j_m pomoću formule

$$(b, j_m) = \arg \max_{\substack{s \in \{1, \dots, n\} \\ j \in \{1, \dots, n\} \setminus J_{m-1}}} \left| \frac{\partial f_s}{\partial x_j}(\mathbf{x}^{m-1}) \right|.$$

Slično kao i kod slučaj< određivanja \mathbf{x}^1 , definiraju se $\mathbf{y}^{j_m} = \mathbf{x}^{m-1} - x_{j_m} \mathbf{e}_{j_m}$ i polinom $f^*(t) = f_b(\mathbf{y}^{j_m} + t \mathbf{e}_{j_m})$.

Ako f^* ima realne nultočke t_1, \dots, t_s , onda se skup T definira kao skup tih nultočaka, $T = \{t_1, \dots, t_s\}$. S druge strane, ako polinom f^* nema realne nultočke, onda ih sigurno ima njegova derivacija $(f^*)'$ pa se T sastoji od nultočaka t'_1, \dots, t'_r polinoma $(f^*)'$, $T = \{t'_1, \dots, t'_r\}$.

Nakon provođenja ovog koraka skup T je neprazan pa je dobro definirana vrijednost

$$t^* = \arg \min_{t \in T} \|\mathbf{F}(\mathbf{y}^{j_m} + t \mathbf{e}_{j_m})\|_\infty. \quad (3.13)$$

Ovako definiran t^* je u skladu s metodom u kojem se duljina spusta određuje minimizacijom max–norme reziduala. Kada se želi koristiti metoda koja duljinu spusta određuje minimizaci-

jom kvadrata euklidske norme reziduala, onda je potrebno u (3.13) promijeniti max–normu u euklidsku i računati t^* prema

$$t^* = \arg \min_{t \in T} \|\mathbf{F}(\mathbf{y}^{j_m} + t \mathbf{e}_{j_m})\|_2. \quad (3.14)$$

Nova točka \mathbf{x}^m definirana je s $\mathbf{x}^m = \mathbf{y}^{j_m} + t^* \mathbf{e}_{j_m}$, a postupak se ponavlja sve dok se ne izračuna \mathbf{x}^n . Onda je smjer spusta \mathbf{s}^{GS} određen sa $\mathbf{s}^{GS} = \mathbf{x}^n - \mathbf{x}^0$.

Idealno bi bilo kada bi vrijedilo da je $\|\mathbf{F}(\mathbf{x}^n)\|_\infty < \|\mathbf{F}(\mathbf{x}^0)\|_\infty$, jer to znači da sigurno dolazi do pomaka u smjeru spusta \mathbf{s}^{GS} , ali to ne mora uvijek vrijediti. Ako je $\|\mathbf{F}(\mathbf{x}^n)\|_\infty > \|\mathbf{F}(\mathbf{x}^0)\|_\infty$ onda u smjeru spusta \mathbf{s}^{GS} može i ne mora doći do pomaka.

Stoga, da bi se osiguralo da Gauss–Seidelov smjer u što više slučajeva daje napredak, uvedena je dodatna provjera. Prvo se pogleda da li je ostvaren napredak u smjeru \mathbf{s}^{GS} . Ako nije ostvaren napredak, onda se definira novi smjer spusta $\mathbf{s}_{n-1}^{GS} = \mathbf{x}^{n-1} - \mathbf{x}^0$ pa se pogleda je li tu ostvaren napredak. Ako je, onda dobro, a ako nije, onda se pokušava sa smjerom $\mathbf{s}_{n-2}^{GS} = \mathbf{x}^{n-2} - \mathbf{x}^0$ itd. Ako niti jedan od smjerova spusta \mathbf{s}_i^{GS} , $i = 1, \dots, n-1$, nije pokazao napredak onda ova metoda ne može napredovati i iterativni postupak treba prekinuti.

Kao i kod prethodnih metoda najdublje spusta, slučaj kada je vektor spusta dobiven kao Gauss–Seidelov smjer određuje dvije metode: GS–E i GS–M. Metoda GS–M je opisana u algoritmu 3.8. Algoritam za metodu GS–E je praktično isti, jedino treba u koracima 16 i 20 zamijeniti max–normu s euklidskom i duljinu spusta određivati pomoću funkcije DULJINASPUSTARSS.

Izračun jednog koraka metoda GS* je računski zahtjevniji od izračuna jednog koraka u metodama GN* i NWT*. Naime, za izračunavanje jednog koraka kod metoda GN* (smjera najstrmijeg spusta), uz pretpostavku da je korak dobro izveden, potrebno je

- izračunati vrijednost funkcije \mathbf{F} ,
- izračunati Jacobijevu matricu $\nabla \mathbf{F}$,
- napraviti umnožak matrice i vektora \mathbf{i}
- odrediti duljinu spusta.

Kod izračunavanja jednog koraka metodama NWT* (Newtonov smjer) treba

- izračunati vrijednost funkcije \mathbf{F} ,
- izračunati Jacobijevu matricu $\nabla \mathbf{F}$,
- riješiti linearni sustav \mathbf{i}
- odrediti duljinu spusta.

Kod određivanja jednog koraka metodama GS* (Gauss–Seidelov smjer) postoji n međukoraka, a u svakom od međukoraka treba izračunati

- vrijednost funkcije \mathbf{F} (ukupno n izračuna funkcije \mathbf{F}),
- vrijednost gradijenta neke funkcije f_b (ukupno n takvih izračuna što je jednako količini posla kod izračuna $\nabla \mathbf{F}$),

Algoritam 3.8 Jedan korak metode najdublje spusta GS-M za rješavanje polinomnog sustava.

Pretpostavke: Zadana početna točka $\mathbf{x}^0 = \mathbf{x}^{(k)}$.

- 1: $m = 0$. $J = \emptyset$.
 - 2: **repeat**
 - 3: $m = m + 1$.
 - 4: Odredi indeks b kao $b = \arg \max_{i \in \{1, \dots, n\}} |f_i(\mathbf{x}^{m-1})|$.
 - 5: Odredi indeks j kao $j = \arg \max_{i \in \{1, \dots, n\} \setminus J} \left| \frac{\partial f_b}{\partial x_i}(\mathbf{x}^{m-1}) \right|$.
 - 6: **if** $\frac{\partial f_b}{\partial x_j}(\mathbf{x}^{m-1}) = 0$ **then**
 - 7: $\mathbf{x}^n = \mathbf{x}^0$. Prekini izvođenje petlje.
 - 8: **end if**
 - 9: $\mathbf{y} = \mathbf{x}^{m-1} - x_j \mathbf{e}_j$.
 - 10: Definiraj polinom $f^*(t) = f_b(\mathbf{y} + t \mathbf{e}_j)$.
 - 11: **if** $f^*(t) = 0$ ima realna rješenja **then**
 - 12: $T = \{t_1, \dots, t_s\}$, t_i , $i = 1, \dots, s$, su realna rješenja od $f^*(t) = 0$.
 - 13: **else**
 - 14: $T = \{t'_1, \dots, t'_r\}$, t'_i , $i = 1, \dots, r$, su realna rješenja od $(f^*)'(t) = 0$.
 - 15: **end if**
 - 16: $t^* = \arg \min_{t \in T} \|\mathbf{F}(\mathbf{y} + t \mathbf{e}_j)\|_\infty$. $\mathbf{x}^m = \mathbf{y} + t^* \mathbf{e}_j$. $J = J \cup \{j\}$.
 - 17: **until** $m = n$
 - 18: Vektor spusta je $\mathbf{s}^{GS} = \mathbf{x}^n - \mathbf{x}^0$.
 - 19: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSAMRN}(\mathbf{x}^{(k)}, \mathbf{s}^{GS})$.
 - 20: **while** $\|\mathbf{F}(\mathbf{x}^{(k)} + \lambda^* \mathbf{s}^{GS})\|_\infty \geq \|\mathbf{F}(\mathbf{x}^{(k)})\|_\infty$
 - 21: $m = m - 1$.
 - 22: **if** $m = 0$ **then**
 - 23: Postavi $\lambda^* = 0$.
 - 24: Prekini izvođenje petlje.
 - 25: **end if**
 - 26: $\mathbf{s}^{GS} = \mathbf{x}^m - \mathbf{x}^0$.
 - 27: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSAMRN}(\mathbf{x}^{(k)}, \mathbf{s}^{GS})$.
 - 28: **end while**
 - 29: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^{GS}$.
-

- naći nultočke jednog ili dva polinoma (ukupno treba naći nultočke najviše $2n$ polinoma) i
- nakon svih međukoraka odrediti duljinu spusta.

Metode GS* u jednom koraku imaju sigurno $n - 1$ evaluacija funkcije \mathbf{F} više u odnosu na GN* i NWT* metode. Nadalje, u jednom koraku metoda GS* treba naći nultočke između n i $2n$ poli-

Tablica 3.3: Uspješnost primjene metoda NWT-E, NWT-M, GN-E, GN-M, BGN-E, BGN-M, GS-E i GS-M na odabranim primjerima polinomnih sustava reda 2.

	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M
freudenstein-roth	23,4	23,1	69,9	99,5	87,7	83,5	81,1	100,0
leary	64,9	40,1	56,5	68,6	80,5	70,3	45,5	90,3
mickey	63,2	63,2	96,1	100,0	100,0	100,0	99,2	100,0
morgan	61,0	61,6	22,5	62,2	61,5	61,4	37,4	38,4
reimer2	75,4	55,6	61,9	58,1	95,1	67,2	99,6	80,2
rosenbrock	13,1	13,6	1,5	16,0	66,0	17,2	41,0	19,3
sendra	77,2	72,9	45,5	62,7	84,4	79,6	90,7	98,1

noma, a vrijeme potrebno za pronalaženje nultočka ovisi o stupnju polinoma.

U usporedbi s metodama BGN*, metode GS* bi trebale biti brže, jer metode BGN* imaju dva izračuna duljine spusta, a izračun duljine spusta je zahtjevan zadatak, pogotovo u slučaju korištenja max-norme reziduala za određivanje duljine spusta.

U tablici 3.3 prikazani su rezultati usporedbe uspješnosti metoda GS* s ostalim metodama najdublje spusta. Kako se može uočiti, uspješnost metoda GS* je dobra u usporedbi s ostalim metodama. To se pogotovo može vidjeti kada se uspoređuju metode koje koriste jednaki način za određivanje duljine spusta. Među metodama koje koriste minimizaciju kvadrata euklidske norme reziduala (NWT-E, GN-E, BGN-E i GS-E), u većini primjera GS-E je prva ili druga po uspješnosti, s time da se naročito dobrom pokazala kod problema reimer2 i sendra. Također, i među metodama koje koriste minimizaciju max-norme reziduala (NWT-M, GN-M, BGN-M i GS-M) metoda GS-M je uglavnom prva ili druga po uspješnosti, s time da se naročito ističe uspješnost pri rješavanju problema leary, reimer2 i sendra.

U tablici 3.4 prikazan je prosječan broj iteracija potrebnih da bi se došlo do rješenja polinomnog sustava primjenom raznih metoda. Očekivano, najveći broj iteracija se očekuje kod metode GN-E jer ona najviše puta koristi smjer najstrmijeg spusta, koji rezultira sporom konvergencijom. Jako mali broj prosječan broj iteracija imaju metode BGN* dok metode koje koriste Gauss-Seidelov smjer spadaju u metode kojima treba više iteracija za dobivanje približnog rješenja.

Tablica 3.4: Prosječan broj iteracija koje su dovele do rješenja kod primjene metoda NWT-E, NWT-M, GN-E, GN-M, BGN-E, BGN-M, GS-E i GS-M na odabranim primjerima polinomnih sustava reda 2.

	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M
freudenstein-Roth	4,1	2,6	63,7	2,2	4,1	2,3	5,4	26,1
leary	10,1	7,5	76	10	5,6	6,5	6,8	18,4
mickey	4,8	4,8	24,3	8,2	5,5	5,4	8,5	14,7
morgan	6,1	5,8	127,3	11,3	5,9	5,8	6,6	20,2
reimer2	9,7	7,3	63,3	12,9	8,9	8,2	12,6	8
rosenbrock	102,1	104,8	174,2	163,9	91,1	152,2	107	152,1
sendra	43,2	6	81,2	50,3	39,9	5,7	28,9	34,5

3.4 Smjer spusta određen max-normom

U poglavlju 3.1.3 spomenuto je da se kod metode GN-M relativno često događa da spust u smjeru najstrmijeg spusta (gradijenta funkcije $g_{\mathbf{F}}$) ne daje dovoljan napredak pa je potrebno koristiti i Newtonov smjer. Razlog tome je što smjer najstrmijeg spusta garantira napredak ako se za određivanje duljine spusta koristi minimizacija kvadrata euklidske norme reziduala, ali ga ne garantira za slučaj kada se duljina spusta određuje minimizacijom max-norme reziduala.

Zato je zanimljivo pitanje može li se naći metoda koja bi generirala smjer spusta koji garantira da će se u tom smjeru postići smanjenje max-norme reziduala? Odnosno, može li se za zadanu točku \mathbf{x} odrediti vektor spusta \mathbf{s} koji garantira da postoji $\lambda \in \mathbb{R}$ takav da je

$$\|\mathbf{F}(\mathbf{x} + \lambda \mathbf{s})\|_{\infty} < \|\mathbf{F}(\mathbf{x})\|_{\infty}.$$

Za početak krenimo sa sljedećim problemom: za zadani \mathbf{x} treba odrediti \mathbf{s} takav da je

$$\mathbf{s} = \arg \min_{\mathbf{h} \in \mathbb{R}^n} \|\mathbf{F}(\mathbf{x} + \mathbf{h})\|_{\infty}. \quad (3.15)$$

Problem (3.15) je ekvivalentan problemu rješavanja polinomnog sustava $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, ako takav sustav ima rješenja. Neka je \mathbf{x}^* rješenje polinomnog sustava, tj. vrijedi $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$. Ako je $\mathbf{s} = \mathbf{x}^* - \mathbf{x}$, onda vrijedi $\|\mathbf{F}(\mathbf{x} + \mathbf{s})\|_{\infty} = 0$, što je rješenje od (3.15). S druge strane neka je \mathbf{s} rješenje (3.15). Tada je $\|\mathbf{F}(\mathbf{x} + \mathbf{s})\|_{\infty} = \min_{\mathbf{h} \in \mathbb{R}^n} \|\mathbf{F}(\mathbf{x} + \mathbf{h})\|_{\infty} = 0$, zbog pretpostavke da polinomni sustav ima rješenje i to rješenje je $\mathbf{x} + \mathbf{s}$.

Obzirom da se problem (3.15) ne može jednostavno riješiti, funkcija \mathbf{F} se zamijeni njezinom linearnom aproksimacijom

$$\mathbf{F}(\mathbf{x} + \mathbf{h}) \approx \mathbf{F}(\mathbf{x}) + \nabla \mathbf{F}(\mathbf{x}) \cdot \mathbf{h},$$

pa (3.15) prelazi u problem: odrediti \mathbf{s} tako da je

$$\mathbf{s} = \arg \min_{\mathbf{h} \in \mathbb{R}^n} \|\mathbf{F}(\mathbf{x}) + \nabla \mathbf{F}(\mathbf{x}) \cdot \mathbf{h}\|_{\infty}. \quad (3.16)$$

Rješenje problema (3.16) je naravno Newtonov smjer $\mathbf{s} = (\nabla \mathbf{F}(\mathbf{x}))^{-1} \mathbf{F}(\mathbf{x})$, ali je upitno koliko dobro taj smjer aproksimira polazni problem (3.15). Dva su moguća razloga zašto Newtonov smjer nije dobar: moguće je da Jacobijeva matrica $\nabla \mathbf{F}(\mathbf{x})$ nije regularna ili je loše uvjetovana, a isto tako je poznato da linearna aproksimacija \mathbf{F} oko točke \mathbf{x} dobro aproksimira \mathbf{F} samo u nekoj okolini od \mathbf{x} . Ako je \mathbf{s} dosta udaljen od \mathbf{x} , razlika između \mathbf{F} i njene linearne aproksimacije može biti dosta velika. Stoga se (3.15)–(3.16) zamjenjuju s problemom optimizacije na manjem području. Problem (3.15) se zamijeni s problemom: odrediti \mathbf{s} tako da je

$$\mathbf{s} = \arg \min_{\mathbf{h} \in \Delta} \|\mathbf{F}(\mathbf{x} + \mathbf{h})\|_{\infty}, \quad \Delta \subset \mathbb{R}^n, \quad (3.17)$$

a kako je taj problem teško riješiti, funkcija \mathbf{F} se zamijeni s njezinom linearnom aproksimacijom. Tako (3.17) prelazi u problem: odrediti \mathbf{s} takav da je

$$\mathbf{s} = \arg \min_{\mathbf{h} \in \Delta} \|\mathbf{F}(\mathbf{x}) + \nabla \mathbf{F}(\mathbf{x}) \cdot \mathbf{h}\|_{\infty}, \quad \Delta \subset \mathbb{R}^n, \quad (3.18)$$

s time da je područje Δ definirano s

$$\Delta = [-\delta_1, \delta_1] \times \cdots \times [-\delta_n, \delta_n], \quad \delta_i > 0, \quad i = 1, \dots, n. \quad (3.19)$$

Uz ovakvu definiciju područja Δ , problem (3.18) se može zapisati kao zadaća linearnog programiranja, ali prije toga je potrebno uvesti oznake koje će olakšati zapis problema (3.18) kao zadaće linearnog programiranja.

Neka su

$$\begin{aligned} \mathbf{A} &= \nabla \mathbf{F}(\mathbf{x}), \quad \mathbf{A} \in \mathbb{R}^{n \times n}, \quad \mathbf{f} = \mathbf{F}(\mathbf{x}), \\ \Delta &= [-\delta_1, \delta_1] \times \cdots \times [-\delta_n, \delta_n], \quad \Delta \subset \mathbb{R}^n, \\ \mathbf{d} &= (\delta_1, \dots, \delta_n)^T, \quad \mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^n. \end{aligned} \quad (3.20)$$

Da bi se mogla napraviti minimizacija u max–normi, potrebno je uvesti dodatnu varijablu μ :

$$\mu = \|\mathbf{f} + \mathbf{A}\mathbf{h}\|_{\infty} = \max_{j=1, \dots, n} |f_j + (\mathbf{A}\mathbf{h})_j|.$$

Uz ovako uvedene oznake, problem (3.18) može se zapisati kao

$$\begin{aligned} \mu &\longrightarrow \min, \\ -\mu \mathbf{1} &\leq \mathbf{f} + \mathbf{A}\mathbf{h} \leq \mu \mathbf{1}, \quad \mathbf{h} \in \Delta, \\ \mu &\leq M, \quad M = \max_{j=1, \dots, n} f_j. \end{aligned} \quad (3.21)$$

Problem (3.21) je zadaća linearnog programiranja, ali treba napraviti još nekoliko transformacija da bi ga se moglo zapisati u standardnom obliku.

$$-\mu \mathbf{1} \leq \mathbf{f} + \mathbf{A}\mathbf{h} \leq \mu \mathbf{1} \Leftrightarrow \begin{array}{l} -\mu \mathbf{1} \leq \mathbf{f} + \mathbf{A}\mathbf{h} \\ \mathbf{f} + \mathbf{A}\mathbf{h} \leq \mu \mathbf{1} \end{array} \Leftrightarrow \begin{array}{l} \mathbf{A}\mathbf{h} + \mu \mathbf{1} \geq -\mathbf{f} \\ -\mathbf{A}\mathbf{h} + \mu \mathbf{1} \geq \mathbf{f} \end{array} \quad (3.22)$$

$$\mathbf{h} \in \Delta \Leftrightarrow -\mathbf{d} \leq \mathbf{h} \leq \mathbf{d} \Leftrightarrow \begin{array}{l} \mathbf{h} \geq -\mathbf{d} \\ -\mathbf{h} \geq -\mathbf{d} \end{array}. \quad (3.23)$$

Nadalje, jer \mathbf{h} može imati i pozitivne i negativne vrijednosti, bolje je zapisati \mathbf{h} kao

$$\mathbf{h} = \mathbf{h}_1 - \mathbf{h}_2, \quad \mathbf{h}_1, \mathbf{h}_2 \geq \mathbf{0}, \quad (3.24)$$

pa (3.21) zajedno s (3.22)–(3.24) daje

$$\begin{aligned} \mu &\rightarrow \min, \\ \mathbf{A}\mathbf{h}_1 - \mathbf{A}\mathbf{h}_2 + \mu \mathbf{1} &\geq -\mathbf{f}, \\ -\mathbf{A}\mathbf{h}_1 + \mathbf{A}\mathbf{h}_2 + \mu \mathbf{1} &\geq \mathbf{f}, \\ \mathbf{h}_1 - \mathbf{h}_2 &\geq -\mathbf{d}, \\ -\mathbf{h}_1 + \mathbf{h}_2 &\geq -\mathbf{d}, \\ -\mu &\geq -M, \\ \mathbf{h}_1, \mathbf{h}_2 &\geq \mathbf{0}, \\ \mu &\geq 0. \end{aligned} \quad (3.25)$$

Uvede li se varijabla t i označi $\mu = M - t$, $t \geq 0$, problem traženje minimuma (3.25) postaje

problem traženja maksimuma,

$$\begin{aligned}
 t &\rightarrow \max, \\
 -\mathbf{A}\mathbf{h}_1 + \mathbf{A}\mathbf{h}_2 + t\mathbf{1} &\leq M\mathbf{1} + \mathbf{f}, \\
 \mathbf{A}\mathbf{h}_1 - \mathbf{A}\mathbf{h}_2 + t\mathbf{1} &\leq M\mathbf{1} - \mathbf{f}, \\
 -\mathbf{h}_1 + \mathbf{h}_2 &\leq \mathbf{d}, \\
 \mathbf{h}_1 - \mathbf{h}_2 &\leq \mathbf{d}, \\
 t &\leq M, \\
 \mathbf{h}_1, \mathbf{h}_2 &\geq \mathbf{0}, \quad t \geq 0.
 \end{aligned} \tag{3.26}$$

U matričnom obliku, uvjeti problema (3.26) mogu se zapisati kao

$$\begin{bmatrix} -\mathbf{A} & \mathbf{A} & \mathbf{1} \\ \mathbf{A} & -\mathbf{A} & \mathbf{1} \\ -\mathbf{I}_n & \mathbf{I}_n & \mathbf{0} \\ \mathbf{I}_n & -\mathbf{I}_n & \mathbf{0} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ t \end{bmatrix} \leq \begin{bmatrix} M\mathbf{1} + \mathbf{f} \\ M\mathbf{1} - \mathbf{f} \\ \mathbf{d} \\ \mathbf{d} \\ M \end{bmatrix}, \quad \mathbf{h}_1, \mathbf{h}_2 \geq \mathbf{0}, \quad t \geq 0. \tag{3.27}$$

Svi elementi vektora s desne strane znaka nejednakosti su nenegativni, što slijedi iz činjenice da je $M = \|\mathbf{f}\|_\infty$. Također, za $\mathbf{h}_1 = \mathbf{0}, \mathbf{h}_2 = \mathbf{0}$ i $t = 0$ uvjeti (3.27) su zadovoljeni.

Problem (3.26) je standardni problem linearnog programiranja s $2n + 1$ varijabli i $4n + 1$ uvjeta koji se može riješiti simpleks metodom.

Neka su $\mathbf{h}_1^*, \mathbf{h}_2^*, t$ rješenja problema (3.26). Tada je $\mathbf{s}^* = \mathbf{h}_1^* - \mathbf{h}_2^*$ rješenje problema (3.18). Postupak kojim se dolazi do \mathbf{s}^* za zadani \mathbf{x} i specijalni oblik područja Δ^\ddagger dan je u algoritmu 3.9.

Pitanje koje ostaje otvoreno je na koji način odrediti područje Δ . Opišimo strategija koja se koristi u ovom radu:

- Pažnja je koncentrirana samo na područja oblika $\Delta(\delta) = [-\delta, \delta] \times \cdots \times [-\delta, \delta] = [-\delta, \delta]^n$.
- Formiran je niz područja $\Delta^1 = \Delta(\delta^1), \dots, \Delta^M = \Delta(\delta^M)$, pri čemu za brojeve $\delta^1, \dots, \delta^M$ vrijedi $\delta^1 < \cdots < \delta^M$, što povlači $\Delta^1 \subset \Delta^2 \subset \cdots \subset \Delta^M$.
- Neka je zadana točka \mathbf{x} . Za zadani \mathbf{x} se problem (3.26) rješava na inicijalnom području Δ^r , na njegovom podskupu Δ^{r-1} i, po potrebi, na njegovom nadskupu Δ^{r+1} .
- Neka su sa $\mathbf{s}^{r-1}, \mathbf{s}^r, \mathbf{s}^{r+1}$ označena rješenja (3.26) na Δ^{r-1}, Δ^r i Δ^{r+1} , respektivno.
- Za smjer spusta \mathbf{s}^* se odabire smjer koji je najviše smanjio max-normu reziduala,

$$\mathbf{s}^* = \arg \min_{s \in \{\mathbf{s}^{r-1}, \mathbf{s}^r, \mathbf{s}^{r+1}\}} \|\mathbf{F}(\mathbf{x} + \mathbf{s})\|_\infty.$$

Za razliku od prethodnih metoda, koje su za određeni način odabira vektora spusta imale

[‡]Algoritam 3.9 se može s minornim modifikacijama koristiti i ako je područje Δ definirano kao u (3.20), ali je napisan na ovakav način jer se u algoritmu za određivanje smjera spusta koristi specijalni oblik od Δ .

Algoritam 3.9 Rješavanje problema

$$\mathbf{s} = \arg \min_{\mathbf{h} \in \Delta} \|\mathbf{F}(\mathbf{x}) + \nabla \mathbf{F}(\mathbf{x}) \cdot \mathbf{h}\|_{\infty}$$

na području $\Delta = [-\delta, \delta] \times \dots \times [-\delta, \delta] = [-\delta, \delta]^n$.

1: **function** SMJERLP(\mathbf{x}, δ)

ULAZ: Zadana točka \mathbf{x} . Zadan δ koji definira područje $\Delta = [-\delta, \delta]^n$.

IZLAZ: Smjer \mathbf{s}^* .

2: Neka je $\mathbf{A} = \nabla \mathbf{F}(\mathbf{x}) \in \mathbf{M}_n$ i $\mathbf{f} = \mathbf{F}(\mathbf{x}) \in \mathbb{R}^n$ i $M = \|\mathbf{F}(\mathbf{x})\|_{\infty}$.

3: Neka je $\mathbf{d} = (\delta, \dots, \delta) \in \mathbb{R}^n$ i $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^n$.

4: Formiraj matricu $\mathbf{C} \in \mathbf{M}_{4n+1, 2n+1}$ i vektor $\mathbf{b} \in \mathbb{R}^n$,

$$\mathbf{C} = \begin{bmatrix} -\mathbf{A} & \mathbf{A} & \mathbf{1} \\ \mathbf{A} & -\mathbf{A} & \mathbf{1} \\ -\mathbf{I}_n & \mathbf{I}_n & \mathbf{0} \\ \mathbf{I}_n & -\mathbf{I}_n & \mathbf{0} \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} M\mathbf{1} + \mathbf{f} \\ M\mathbf{1} - \mathbf{f} \\ \mathbf{d} \\ \mathbf{d} \\ M \end{bmatrix}.$$

5: Neka je $\mathbf{y}^* = (y_1^*, \dots, y_{2n}^*, y_{2n+1}^*) \in \mathbb{R}^{2n+1}$ rješenje zadane linearnog programiranja
 $y_{2n+1} \rightarrow \max$, uz uvjete $\mathbf{C}\mathbf{y} \leq \mathbf{b}$ i $\mathbf{y} \geq \mathbf{0}$.

6: $\mathbf{h}_1^* = (y_1^*, \dots, y_n^*)$, $\mathbf{h}_2^* = (y_{n+1}^*, \dots, y_{2n}^*)$.

7: $\mathbf{s}^* = \mathbf{h}_1^* - \mathbf{h}_2^*$

8: **end function**

dvije inačice, ovisno o načinu određivanja duljine spusta, za slučaj kada se u metodi najdublje spusta koristi smjer određen kao smjer pada u max-normi razvijena je samo jedna metoda, LP-M.

Postupak za provođenje jednog koraka metode LP-M prikazan je u algoritmu 3.10.

Brojevi δ^i se u konkretnoj realizaciji algoritma nalaze u intervalu od 0,0001 pa do 5. Strategija odabira smjera malo forsira smjer koji je dobiven na manjem području (korak 8 u algoritmu 3.10, smjer \mathbf{s}^M) u odnosu na smjer dobiven na inicijalnom području (smjer \mathbf{s}^0) ili većem (smjer \mathbf{s}^V), jer u slučaju u kojem je $f^M < f^0 < f^{OLD}$ odmah proglašava da je smjer spusta \mathbf{s}^M , dok se u svim ostalim slučajevima se provjerava da je jedan smjer daje veći pad od preostala dva. Metoda LP-M započinje iterativni proces s $\delta^r = 1$.

Izračun jednog koraka metode LP-M je dosta skup jer pravi smjer treba odabrati između tri ponuđena, a za svaki od njih treba riješiti odgovarajuću zadaću linearnog programiranja. Obzirom da su sve te zadaje vrlo slične (razlikuju se samo u veličini područja na kojem se traži rješenje problema), moguće je da bi se postupak mogao ubrzati istovremenim rješavanjem sve tri zadaje, što bi zahtijevalo razvoj specijalnog algoritma.

U tablici 3.5 prikazana je usporedba između uspješnosti metode LP-M i do sada spomenutih metoda (NWT-M, GN-M, BGN-M i GS-M) na odabranim polinomnim sustavima reda 2.

Na spomenutim primjerima metoda LP-M je uvjerljivo najlošija, ali kod problema višeg reda

Algoritam 3.10 Jedan korak metode najdublje spusta LP-M za rješavanje polinomnog sustava.

Pretpostavke: Zadana početna točka $\mathbf{x} = \mathbf{x}^{(k)}$. Zadan δ^r .

- 1: $f^{OLD} = \|\mathbf{F}(\mathbf{x})\|_{\infty}$. $\delta^{NEW} = \delta^r$.
 - 2: $\mathbf{s}^0 = \text{SMJERLP}(\mathbf{x}, \delta^r)$, rješenje problema (3.18) na području Δ^r . (algoritam 3.9).
 - 3: $\mathbf{s}^{LP} = \mathbf{s}^0$.
 - 4: $f^0 = \|\mathbf{F}(\mathbf{x} + \mathbf{s}^0)\|_{\infty}$.
 - 5: $\mathbf{s}^M = \text{SMJERLP}(\mathbf{x}, \delta^{r-1})$, rješenje problema (3.18) na području $\Delta^{r-1} \subset \Delta^r$.
 - 6: $f^M = \|\mathbf{F}(\mathbf{x} + \mathbf{s}^M)\|_{\infty}$.
 - 7: **if** $f^0 < f^{OLD}$ **then**
 - 8: **if** $f^M < f^0$ **then**
 - 9: $\mathbf{s}^{LP} = \mathbf{s}^M$. $\delta^{NEW} = \delta^{r-1}$.
 - 10: **else**
 - 11: $\mathbf{s}^V = \text{SMJERLP}(\mathbf{x}, \delta^{r+1})$, rješenje problema (3.18) na području $\Delta^{r+1} \supset \Delta^r$.
 - 12: $f^V = \|\mathbf{F}(\mathbf{x} + \mathbf{s}^V)\|_{\infty}$.
 - 13: **if** $f^V < f^0$ **then**
 - 14: $\mathbf{s}^{LP} = \mathbf{s}^V$. $\delta^{NEW} = \delta^{r+1}$.
 - 15: **end if**
 - 16: **end if**
 - 17: **else**
 - 18: $\mathbf{s}^V = \text{SMJERLP}(\mathbf{x}, \delta^{r+1})$, rješenje problema (3.18) na području Δ^{r+1} .
 - 19: $f^V = \|\mathbf{F}(\mathbf{x} + \mathbf{s}^V)\|_{\infty}$.
 - 20: **if** $f^V < f^0$ **and** $f^V < f^M$ **then**
 - 21: $\mathbf{s}^{LP} = \mathbf{s}^V$. $\delta^{NEW} = \delta^{r+1}$.
 - 22: **else if** $f^M < f^0$ **and** $f^M < f^V$ **then**
 - 23: $\mathbf{s}^{LP} = \mathbf{s}^M$. $\delta^{NEW} = \delta^{r-1}$.
 - 24: **end if**
 - 25: **end if**
 - 26: Odredi duljinu spusta λ^* , $\lambda^* = \text{DULJINASPUSTAMRN}(\mathbf{x}^{(k)}, \mathbf{s}^{LP})$.
 - 27: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^* \mathbf{s}^{LP}$.
 - 28: δ^{NEW} je granica područja za idući korak.
-

ima dosta slučajeva kada metoda LP-M daje dobre rezultate.

Algoritam bi se mogao popraviti boljim načinom određivanja područja Δ na kojem se određuje smjer spusta, a brzina bi se mogla popraviti i paralelizacijom rješavanja problema linearnog programiranja (3.26) na područjima Δ^{r-1} , Δ^r i Δ^{r+1} (koraci 2, 5, 11, odnosno 18 u algoritmu 3.10).

Tablica 3.5: Uspješnost primjene metoda LP-M u usporedbi s NWT-M, GN-M, BGN-M i GS-M na odabranim primjerima polinomnih sustava reda 2.

	NWT-M	GN-M	BGN-M	GS-M	LP-M
freudenstein-roth	23,1	99,5	83,5	100,0	27,6
leary	40,1	68,6	70,3	90,3	35,9
mickey	63,2	100,0	100,0	100,0	51,4
morgan	61,6	62,2	61,4	38,4	41,3
reimer2	55,6	58,1	67,2	80,2	8,7
rosenbrock	13,6	16,0	17,2	19,3	14,2
sendra	72,9	62,7	79,6	98,1	46,1

3.5 Smjerovi koordinatnih osi

Ideja izbora koordinatnih osi za smjerove spusta spomenuta je u samom uvodu u problem rješavanja polinomnih sustava metodom najdublje spusta (poglavlje 2.2) i tamo je spomenuto da metoda ne postiže naročito dobre rezultate, ali ovdje ćemo je spomenuti radi potpunosti.

Prednosti metode su jednostavno određivanje smjera spusta i prikaza polinoma f_i u standardnoj bazi. Nedostatak je što za određivanje pravog smjera, treba gledati u smjeru svih koordinatnih osi, vidjeti gdje se postiže najveći napredak i zatim smjer u kojem je postignut najveći napredak proglasiti pravim smjerom. Cijeli postupak je već opisan u algoritmu 2.3 za određivanje duljine spusta minimizacijom kvadrata euklidske norme.

Radi potpunosti, ovdje će se opisati varijanta algoritma koja za određivanje duljine spusta koristi minimizaciju max-norme reziduala, a cijela metoda se dobije uklapanjem algoritma u opći algoritam 2.8 (zamjena koraka 4–6). Metode koje koriste smjerove koordinatnih osi za smjer spusta nazivaju se KO-E i KO-M.

U tablici 3.6 prikazana je uspješnost metoda koje koriste minimizaciju kvadrata euklidske norme reziduala za određivanje duljine spusta (*-E metode). Metoda KO-E se pokazala izvrsnom na primjeru freudenstein-roth, ali na ostalim primjerima nije pokazala neku uspješnost. Dapače, na tri primjera uspješnost je zanemariva (morgan, rosenbrock i sendra).

U tablici 3.7 prikazana je uspješnost metoda koje koriste minimizaciju max-norme reziduala za određivanje duljine spusta (*-M metode). U usporedbi s ostalim *-M metodama, KO-M se pokazala boljom nego li se KO-E metoda pokazala u usporedbi s ostalim *-E metodama. KO-M se pokazala jako dobrom na primjeru freudenstein-roth, solidnom na primjeru mickey, a zanemariva uspješnost je bila na primjerima morgan i rosenbrock.

Algoritam 3.11 Jedan korak metode najdublje spusta KO-M za rješavanje polinomnog sustava.

Pretpostavke: Zadana početna točka $\mathbf{x}^{(k)}$.

- 1: Odredi $\lambda^* = \text{DULJINASPUSTAMRN}(\mathbf{x}^{(k)}, \mathbf{e}_1)$
- 2: $\mathbf{s}^{KO} = \lambda^* \mathbf{e}_1$
- 3: $f^{KO} = \|\mathbf{F}(\mathbf{x}^{(k)} + \lambda^* \mathbf{e}_1)\|_\infty$.
- 4: **for** $i=2$ **to** n
- 5: Odredi $\lambda^* = \text{DULJINASPUSTAMRN}(\mathbf{x}^{(k)}, \mathbf{e}_i)$
- 6: $f^i = \|\mathbf{F}(\mathbf{x}^{(k)} + \lambda^* \mathbf{e}_i)\|_\infty$.
- 7: **if** $f^i < f^{KO}$ **then**
- 8: $\mathbf{s}^{KO} = \lambda^* \mathbf{e}_i$.
- 9: $f^{KO} = f^i$.
- 10: **end if**
- 11: **end for**
- 12: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{KO}$.

Tablica 3.6: Uspješnost primjene metode KO-E u usporedbi s NWT-E, GN-E, BGN-E i GS-E na odabranim primjerima polinomnih sustava reda 2.

	NWT-E	GN-E	BGN-E	GS-E	KO-E
freudenstein-roth	23,4	69,9	87,7	81,1	97,2
leary	64,9	56,5	80,5	45,5	31,3
mickey	63,2	96,1	100,0	99,2	65,3
morgan	61,0	22,5	61,5	37,4	0,0
reimer2	75,4	61,9	95,1	99,6	27,5
rosenbrock	13,1	1,5	66,0	41,0	0,0
sendra	77,2	45,5	84,4	90,7	1,1

3.6 Metoda najdublje spusta u \mathbb{C}^n

U ovom potpoglavlju promatra se mogućnost proširenja metode najdublje spusta tako da se traže i kompleksna rješenja realnih polinomnih sustava, tj. polinomnih sustava kojima su svi koeficijenti realni brojevi.

Neka je s $\mathbf{F}(\mathbf{z}) = \mathbf{0}$, $\mathbf{F} = (f_1, \dots, f_n)$, zadan realni polinomni sustav. Naš cilj je pokušati proširiti metodu najdublje spusta tako da možemo dobiti i kompleksna rješenja sustava. U tom smislu potrebno je vidjeti što treba napraviti da bi se metoda mogla proširiti, odnosno, koje prepravke je potrebno napraviti na ključnim koracima metode najdublje spusta da bi metoda

Tablica 3.7: Uspješnost primjene metode KO-M u usporedbi s NWT-M, GN-M, BGN-M, GS-M i LP-M na odabranim primjerima polinomnih sustava reda 2.

	NWT-M	GN-M	BGN-M	GS-M	LP-M	KO-M
freudenstein-roth	23,1	99,5	83,5	100,0	27,6	90,1
leary	40,1	68,6	70,3	90,3	35,9	32,0
mickey	63,2	100,0	100,0	100,0	51,4	65,6
morgan	61,6	62,2	61,4	38,4	41,3	1,0
reimer2	55,6	58,1	67,2	80,2	8,7	16,4
rosenbrock	13,6	16,0	17,2	19,3	14,2	0,0
sendra	72,9	62,7	79,6	98,1	46,1	28,7

radila i u kompleksnom području.

Određivanje smjera samo po sebi nije problematično jer je Newtonov smjer dobro definiran i u slučaju kompleksne funkcije. Rješavanje linearnog sustava s koeficijentima iz polja kompleksnih brojeva isto nije problematično.

Nakon što je određen smjer spusta s iz neke početne točke \mathbf{z}^0 , potrebno je odrediti polinome $f_{i,\mathbf{z}^0,\mathbf{s}}$. Ti polinomi su kompleksne funkcije realne varijable $f_{i,\mathbf{z}^0,\mathbf{s}} : \mathbb{R} \rightarrow \mathbb{C}$, koji predstavljaju restrikciju polinoma f_i na odgovarajući pravac u kompleksnom prostoru \mathbb{C}^n koji prolaz kroz točku \mathbf{z}^0 i ima smjer \mathbf{s} . Točke tog pravca su opisane jednadžbom $\mathbf{z} = \mathbf{z}^0 + \lambda \mathbf{s}$ pa su polinomi $f_{i,\mathbf{z}^0,\mathbf{s}}$ definirani s

$$f_{i,\mathbf{z}^0,\mathbf{s}}(\lambda) = f_i(\mathbf{z}^0 + \lambda \mathbf{s}).$$

Koeficijenti polinoma $f_{i,\mathbf{z}^0,\mathbf{s}}$ općenito mogu biti i kompleksni brojevi.

Kao i u realnom slučaju, potrebno je odrediti njihov prikaz u standardnoj bazi polinoma $1, \lambda, \lambda^2, \dots$. Za to se mogu koristiti metode simboličkog računanja, analogne onima za realni slučaj, koje su opisane u potpoglavlju 2.4. Pri tome treba uvažiti da su koeficijenti kompleksni brojevi i s njima računati na odgovarajući način.

Dosadašnji koraci su bili istovjetni koracima za realni slučaj, samo je računanje s realnim brojevima zamijenjeno računanjem s kompleksnim brojevima, no kod određivanja duljine koraka to se mijenja. Umjesto zbroja kvadrata reziduala (funkcije RSS) koristit će se funkcija $\text{CSS}_{\mathbf{z}^0,\mathbf{s}}$ koja predstavlja zbroj kvadrata apsolutnih vrijednosti reziduala, ali samo za točke na pravcu $\mathbf{z} = \mathbf{z}^0 + \lambda \mathbf{s}$,

$$\text{CSS}_{\mathbf{z}^0,\mathbf{s}}(\lambda) = \sum_{i=1}^n |f_{i,\mathbf{z}^0,\mathbf{s}}(\lambda)|^2 = \sum_{i=1}^n f_{i,\mathbf{z}^0,\mathbf{s}}(\lambda) \overline{f_{i,\mathbf{z}^0,\mathbf{s}}(\lambda)}.$$

U realnom slučaju, ove funkcije su jednake, ali u kompleksnom slučaju nisu, jer za kompleksni broj z vrijedi $z^2 \in \mathbb{C}$, dok je $|z|^2 = z \cdot \bar{z}$ realan broj. Nadalje, z^2 je neprekidno diferencijabilna na skupu kompleksnih brojeva, dok $|z|^2$ nije diferencijabilna u 0. Srećom, restrikcija od $|z|^2$ na pravac u kompleksnoj ravnini je diferencijabilna funkcija u skupu realnih brojeva, pa je tako i funkcija $\text{CSS}_{\mathbf{z}^0, \mathbf{s}}$.

Nakon što je definiran polinom $\text{CSS}_{\mathbf{z}^0, \mathbf{s}}$, postupak se nastavlja slično kao i u realnom slučaju, određivanjem nultočka derivacije polinoma $\text{CSS}_{\mathbf{z}^0, \mathbf{s}}$. Neka je skup $N_0 = \{z_1, \dots, z_k\}$ skup svih nultočka, realnih i kompleksnih, derivacije polinoma $\text{CSS}_{\mathbf{z}^0, \mathbf{s}}$.

Nova točka $\mathbf{z}^{(1)}$ se definira s

$$\mathbf{z}^{(1)} = \mathbf{z}^{(0)} + \lambda^* \mathbf{s},$$

pri čemu je $\lambda^* \in \mathbb{C}$ određena s

$$\lambda^* = \arg \min_{z \in N_0} |\text{CSS}_{\mathbf{z}^0, \mathbf{s}}(\lambda)|.$$

Geometrijski, na slučaj kada je $\lambda^* \in \mathbb{C}$ može se gledati kao na zaokretanje vektora spusta \mathbf{s} . Kako se λ^* može zapisati u polarnom obliku $\lambda^* = |\lambda^*|e^{i\phi}$, vrijedi da je $\lambda^* \mathbf{s} = |\lambda^*|e^{i\phi} \mathbf{s}$. Uz definiciju $\mathbf{s}^* = e^{i\phi} \mathbf{s}$, za novu točku $\mathbf{z}^{(1)}$ vrijedi

$$\mathbf{z}^{(1)} = \mathbf{z}^{(0)} + \lambda^* \mathbf{s} = \mathbf{z}^{(0)} + |\lambda^*|e^{i\phi} \mathbf{s} = \mathbf{z}^{(0)} + |\lambda^*| \mathbf{s}^*.$$

Dakle, $\mathbf{z}^{(1)}$ se ne nalazi na pravcu određenom točkom $\mathbf{z}^{(0)}$ i vektorom spusta \mathbf{s} , već se nalazi na pravcu određenom točkom $\mathbf{z}^{(0)}$ i vektorom spusta \mathbf{s}^* koji se dobije rotacijom vektora \mathbf{s} za kut ϕ .

Proširenje algoritma najdublje spusta na kompleksne nultočke je otvoreno pitanje, ali početni rezultati obećavaju. Na određenim problemima na kojima su najbolje realne metode najdublje spusta imale uspješnost oko 60% (potpoglavlje 4.2), kompleksna metoda najdublje spusta koja koristi Newtonov smjer je imala uspješnost veću od 90%.

Sam algoritam je realiziran u programu *Mathematica*.

Poglavlje 4

Rezultati

U ovom poglavlju se daje prikaz rezultata rada metoda najdublje spusta na nizu polinomnih sustava. U prvom potpoglavlju se na nekoliko primjera daje uvid u fenomene koji nastaju prilikom rješavanja polinomnih sustava. U drugom potpoglavlju opisan je numerički eksperiment koji je proveden za određivanje uspješnosti metoda. Dan je pregled polinomnih sustava koji su korišteni u eksperimentu te uspješnost metoda pri rješavanju sustava. U posljednjem potpoglavlju opisan je način određivanja indeksa pomoću kojih se može na bolji način usporediti metode i odrediti koja je metoda najbolja.

4.1 Područja konvergencije različitih metoda

U ovom poglavlju prikazuje se ponašanje različitih metoda na primjerima polinomnih sustava mickey, leary i himmelbaum. Cilj je pokazati da područja na kojima metode konvergiraju ne trebaju imati nikakvu specijalnu strukturu, u kakvom su odnosu područja konvergencije različitih nultočaka te ukazati na važnost izbora smjera.

4.1.1 Problem mickey

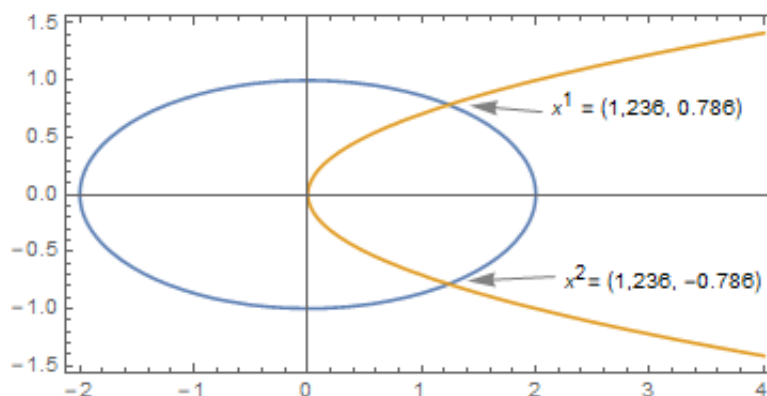
U ovom primjeru promatra se ponašanje metoda za rješavanje polinomnog sustava na problemu mickey. Na slici 4.1 dane su jednadžbe i geometrijska interpretacija sustava.

Problem mickey ima dva realna rješenja $\mathbf{x}^1 = (1,236, 0,786)$ i $\mathbf{x}^2 = (1,236, -0,786)$.

Numerički eksperiment je napravljen na području $D = [-1, 4] \times [-3, 3]$ kako bi se za svaku metodu utvrdilo kolika je njezina uspješnost, koliki udjel iterativnih procesa konvergira prema pojedinom rješenju te koliko je prosječno iteracija napravljeno dok se iterativni proces nije zaustavio. Područje D je odabrano tako da po x -osi nultočke budu približno po sredini, a kako su nultočke simetrične obzirom na y -os, onda je logično da se druga komponenta početnog vektora bira iz simetričnog intervala oko y -osi.

$$f_1(x, y) = x^2 + 4y^2 - 4$$

$$f_2(x, y) = 2y^2 - x$$



Slika 4.1: Polinomni sustav Mickey i njegova geometrijska interpretacija.

Početne točke su odabrane prema standardnoj ekvidistantoj podjeli područja D

$$\mathbf{x}_{i,j}^{(0)} = (x_i, y_j), \quad i, j = 0, \dots, 200;$$

$$x_i = -1 + i \cdot h_x, \quad h_x = (4 - (-1))/200, \quad i = 0, \dots, 200;$$

$$y_j = -3 + j \cdot h_y, \quad h_y = (3 - (-3))/200, \quad i = 0, \dots, 200.$$

Pokus je proveden za ukupno $201^2 = 40.401$ različitu početnu točku.

U tablici 4.1 prikazani su rezultati vezani uz uspješnost iterativnog postupka te njegova završetka (konvergirao prema rješenju \mathbf{x}^1 ili \mathbf{x}^2 , odnosno nije konvergirao prema rješenju).

Na promatranom području D , metode NWT*, GN*, BGN* i GS* imaju uspješnost skoro 100%. Ako se usporede ti rezultati s rezultatima iz tablica 3.6 i 3.7, vidi se da su takvi rezultati očekivani za sve navedene metode osim za metode NWT*. Ovaj rezultat sugerira da je izbor početne točke važan, jer ako je početna točka bliže rješenju veća je vjerojatnost da će iterativni proces i konvergirati prema rješenju.

Jacobijeva matrica problema Mickey je

$$J\mathbf{F}(\mathbf{x}) = \nabla\mathbf{F}(\mathbf{x}) = \begin{bmatrix} 2x & 8y \\ -1 & 4y \end{bmatrix},$$

a $\det J\mathbf{F}(\mathbf{x}) = 8xy + 8y$ pa problemi mogu nastati tamo gdje je Jacobijeva matrica singularna, što je slučaj za $y = 0$, odnosno $x = -1$. Upravo u točkama za koje vrijedi $y = 0$, odnosno $x = -1$ su i započeli iterativni postupci koji nisu konvergirali prema rješenju.

Kako je područje napravljeno tako da je jednak broj početnih točaka kojima je bliže rješenje \mathbf{x}^1 i početnih točaka kojima je bliže rješenje \mathbf{x}^2 , za očekivati bi bilo da će podjednaki broj iteracijskih procesa konvergirati prema rješenju \mathbf{x}^1 , odnosno prema \mathbf{x}^2 . To je istina za metode NWT*, GN*, BGN* i LP-M, ali u slučaju metoda GS* i KO*, vidljivo je da je u većini slučajeva

Tablica 4.1: Prikaz rezultata rješavanja problema mickey različitim metodama. Za svaku metodu je prikazano koliki udjel iterativnih procesa završi u nultočki $\mathbf{x}^1 = (1,236, 0,786)$ (stupac 1), u nultočki $\mathbf{x}^2 = (1,236, -0,786)$ (stupac 2) te koliko iterativnih procesa ne konvergira prema rješenjima.

Metoda	1	2	NE	Metoda	1	2	NE
NWT-E	49,5	49,5	1,0	NWT-M	49,5	49,5	1,0
GN-E	49,7	49,7	0,5	GN-M	49,8	49,8	0,5
BGN-E	49,5	49,5	1,0	BGN-M	49,8	49,8	0,5
GS-E	32,2	67,0	0,8	GS-M	36,0	63,5	0,5
KO-E	8,0	74,3	17,7	KO-M	13,9	72,7	13,4
				LP-M	36,9	36,4	26,7

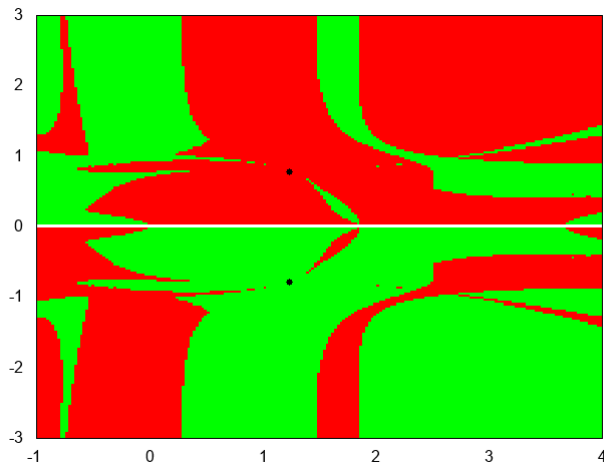
iterativni proces konvergira prema rješenju \mathbf{x}^2 .

Da bi se ilustriralo ponašanje metoda, za svaku metodu je napravljena slika koja pokazuje gdje završava iteracijski proces koji započinje u određenoj početnoj točki. Ako je boja točke crvena, onda je iteracijski proces koji je započeo u toj točki konvergira prema \mathbf{x}^1 , ako je boja zelena, onda je konvergira prema \mathbf{x}^2 , a ako ne boja bijela, onda proces nije konvergira. Dvije crne točke na svakoj slici naznačuju poziciju \mathbf{x}^1 i \mathbf{x}^2 .

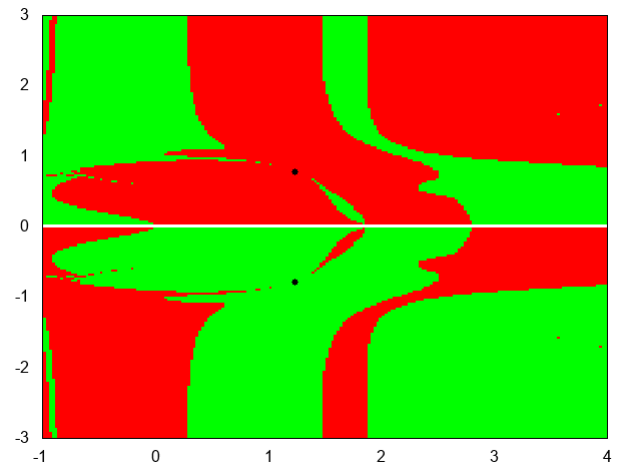
Kod metoda kod kojih podjednaki broj iteracijskih postupaka konvergira prema svakom rješenju, očekivalo bi se da kad postupak konvergira, da konvergira prema bližoj nultočki. No, to je jedino istina za metode NWT*, što se može vidjeti na slici 4.2.(i)*. Na slikama 4.2.(a)–(d) prikazano je ponašanje metoda BGN* i GN*. Primjetno je da postoje sličnosti između metoda koji koriste jednaki smjer spusta, ali različiti način izbora duljine spusta može voditi prema različitim nultočkama.

Zanimljivo je na slikama 4.2.(e)–(f) uočiti razliku između područja konvergencije za metode GS*. Kod metode GS-E postoje nekakva područja konvergencije, dok se kod GS-M metode čini da je riječ o slučajnom odabiru. Sličan zaključak je i kod metode LP-M, s time da je tu jasno uočljivo područje u kojem nema konvergencije.

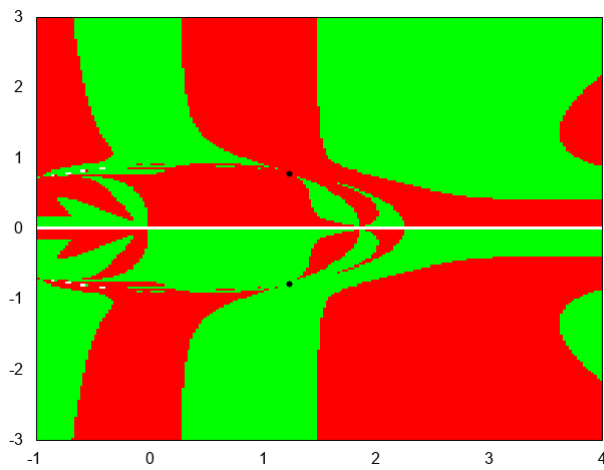
*Slika za metodu NWT-M je potpuno ista pa zato nije uključena



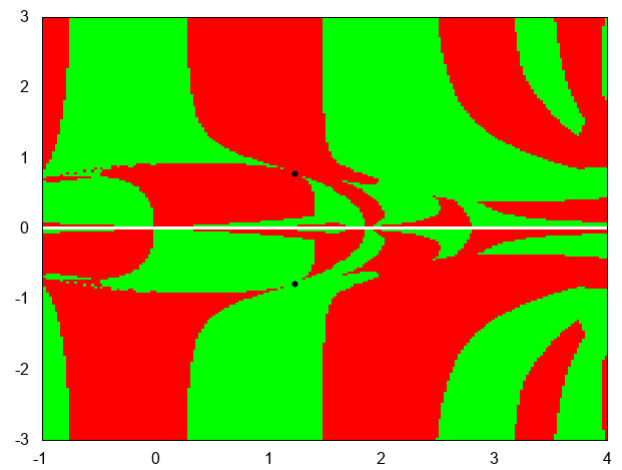
(a) metoda BGN-E



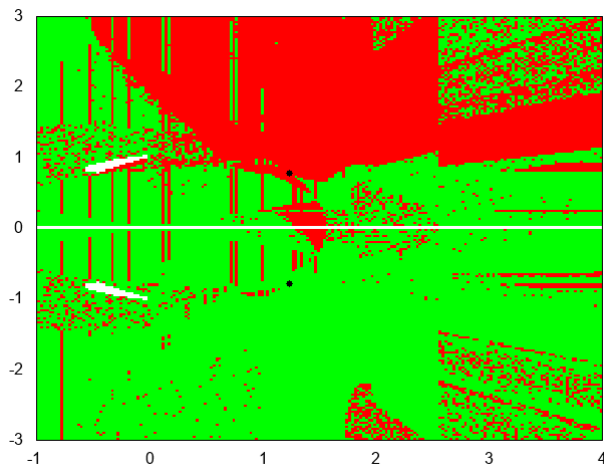
(b) metoda BGN-M



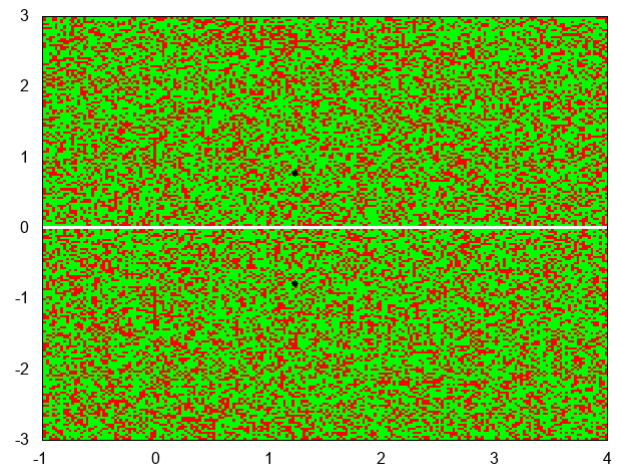
(c) metoda GN-E



(d) metoda GN-M

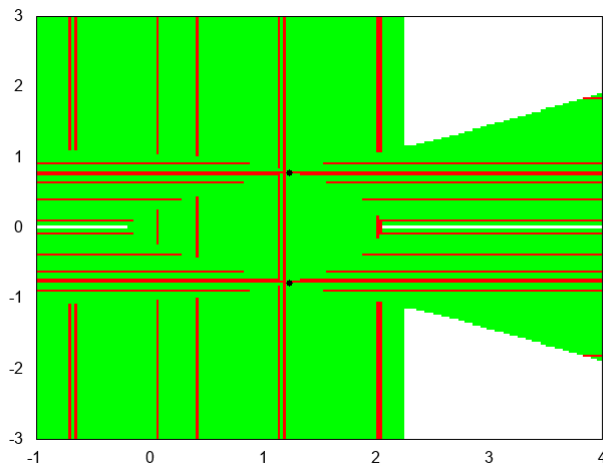


(e) metoda GS-E

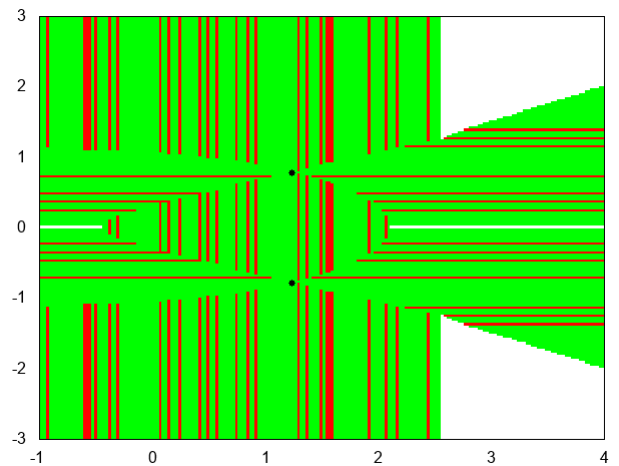


(f) metoda GS-M

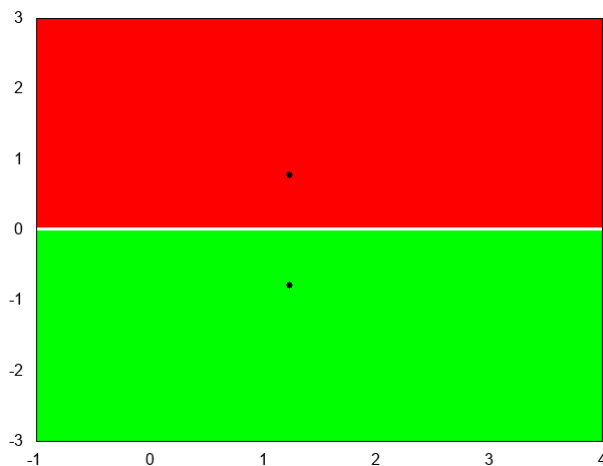
Slika 4.2: Prikaz područja konvergencije za pojedinu nultočku problema mi ckey. Crvena boja označava da iterativni proces konvergira nultočki $\mathbf{x}^1 = (1,236, 0,786)$, dok zelena boja označava da iterativni proces konvergira nultočki $\mathbf{x}^2 = (1,236, -0,786)$. Crne točke označavaju poziciju nultočaka \mathbf{x}^1 i \mathbf{x}^2 .



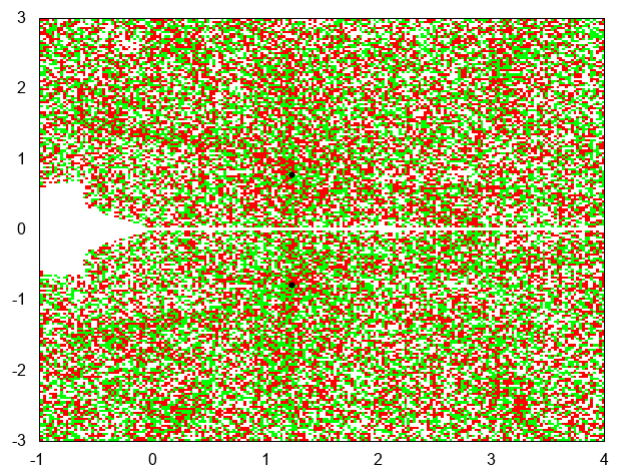
(g) metoda KO-E



(h) metoda KO-M



(i) metoda NWT-E



(j) metoda LP-M

Slika 4.2: (nastavak) Prikaz područja konvergencije za pojedinu nultočku problema mickey. Crvena boja označava da iterativni proces konvergira nultočki $\mathbf{x}^1 = (1,236, 0,786)$, dok zelena boja označava da iterativni proces konvergira nultočki $\mathbf{x}^2 = (1,236, -0,786)$. Crne točke označavaju poziciju nultočaka \mathbf{x}^1 i \mathbf{x}^2 .

U tablici 4.2 prikazano je koliki je prosječan broj iteracija potreban da bi iterativni proces konvergirao prema rješenju \mathbf{x}^1 , rješenju \mathbf{x}^2 , odnosno koliko je prosječno iteracija potrebno da bi se proces zaustavio ako ne konvergira prema rješenju. Metode NWT*, BGN* i KO* imaju približno jednak broj iteracija bez obzira na način određivanja duljine spusta, metoda GN-M ima manje iteracija od metode GN-E, što se može objasniti time da češće koristi Newtonov smjer jer smjer najstrmijeg spusta nije dao rezultat. S druge strane, kod metode GS-E je potrebno dvostruko manje iteracija u odnosu na metodu GS-M.

Najbrže su metode bazirane na Newtonovu smjeru što se uklapa u teoriju koja govori da Newtonova metoda, kada konvergira, konvergira jako brzo (kvadratična konvergencija).

Zanimljivo je uočiti da sve metode relativno brzo ustanovljavaju da nema mogućnosti na-

Tablica 4.2: Prikaz broja iteracija kod rješavanja problema *mickey* različitim metodama. Za svaku metodu je prikazano koliko je prosječno iteracija potrebno da bi proces konvergirao prema nekom od rješenja $\mathbf{x}^1 = (1,236, 0,786)$ (stupac 1) ili $\mathbf{x}^2 = (1,236, -0,786)$ (stupac 2), odnosno da bi se proces zaustavio bez da konvergira rješenjima.

Metoda	1	2	NE	Metoda	1	2	NE
NWT-E	4,5	4,5	1,0	NWT-M	4,6	4,6	1,0
GN-E	27,5	27,5	4,1	GN-M	10,9	10,9	3,0
BGN-E	8,9	8,9	2,0	BGN-M	9,3	9,3	4,0
GS-E	7,1	7,1	3,9	GS-M	14,0	14,2	2,5
KO-E	20,0	20,0	3,0	KO-M	19,6	19,5	3,0
				LP-M	13,7	13,7	3,8

pretka, što se vidi po tome da je najveći prosječan broj iteracija za slučaj kada proces ne konvergira samo 4,1 (tablica 4.2). Da razlog tome nije samo činjenica što ima relativno malo točaka za koje iterativni proces ne konvergira, pokazuju podatci za metode KO-E, KO-M i LP-M: premda je broj neuspjeha između 13% i 26%, prosječan broj iteracija u takvim slučajevima je 3 (za metode KO*), odnosno, 3,8 (za metodu LP-M).

4.1.2 Problem *leary*

U ovom primjeru promatrat će se ponašanje metoda za rješavanje polinomnog sustava na primjeru problema *leary*. Jednadžbe polinoma f_1 i f_2 koji čine sustav i njegova geometrijska interpretacija prikazani su na slici 4.3. Ako se rješenje sustava promatra kao skup točaka u kojima se sijeku krivulje $f_1(x, y) = 0$ i $f_2(x, y) = 0$, važno je uočiti da krivulje nisu jednostavne. Krivulja $f_1(x, y) = 0$ sastoji se od tri krivulje, obje koordinatne osi te pravca $x + 2y + 1 = 0$, dok se krivulja $f_2(x, y) = 0$ sastoji od dva dijela. Ipak, ove dvije krivulje imaju samo jednu zajedničku točku, $\mathbf{x}^* = (-2,679, 0,8397)$.

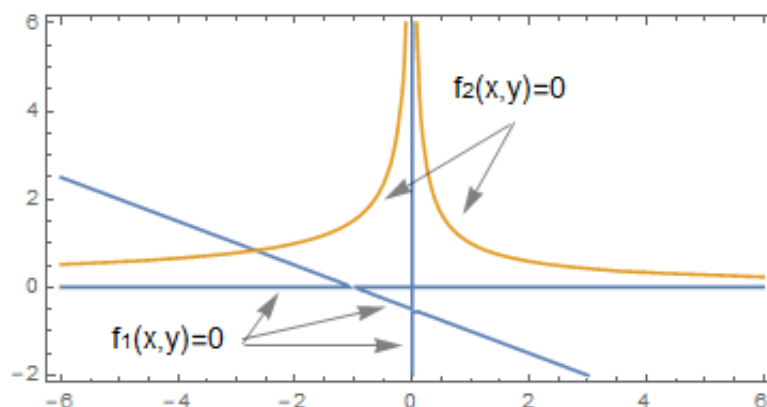
Slično kao i kod problema *mickey*, numerički eksperiment je napravljen na pravokutnom području

$$D = [-5,68, 0,32] \times [-2,16, 3,84] = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x} - \mathbf{x}^*\|_\infty \leq 3\}.$$

da bi se za svaku metodu ispitala uspješnost metode te iz kojih početnih točaka iterativni proces konvergira rješenju. Pokus je napravljen uz dva različita ograničenja najvećeg broja iteracija, $M_{IT} = 20$ i $M_{IT} = 100$, da bi se vidjelo hoće li veći najveći broj iteracija utjecati na uspješnost metoda.

$$f_1(x, y) = x^2y + 2xy^2 + xy$$

$$f_2(x, y) = x^2y^3 + xy - 2$$



Slika 4.3: Polinomni sustav Leary i njegova geometrijska interpretacija.

Početne točke su odabrane prema standardnoj ekvidistantoj podjeli područja D

$$\begin{aligned} \mathbf{x}_{i,j}^{(0)} &= (x_i, y_j), & i, j &= 0, \dots, 200; \\ x_i &= -1 + i \cdot h_x, & h_x &= 6/200, & i &= 0, \dots, 200; \\ y_j &= -3 + j \cdot h_y, & h_y &= 6/200, & j &= 0, \dots, 200. \end{aligned}$$

Pokus je proveden za ukupno $201^2 = 40.401$ različitu početnu točku.

U tablici 4.3 prikazana je uspješnost pojedinih metoda ovisno o najvećem dozvoljenom broju iteracija. Kod većine metoda rast uspješnosti je bio zanemariv osim kod metoda GN-E i GS-M. Dapače, uz povećani najveći broj iteracija, metoda GS-M je metoda koja ima najveću uspješnost.

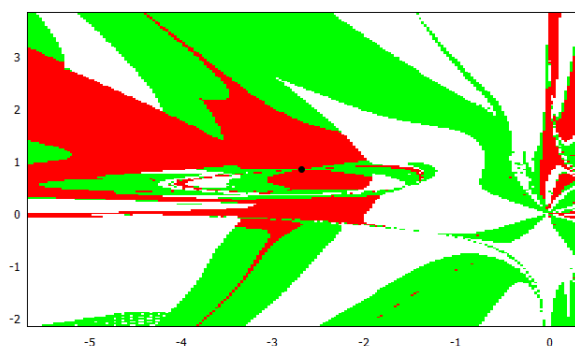
Kada se dobiveni rezultati usporede s rezultatima iz tablica 3.6 i 3.7, može se očitati da primjena metoda na području D ima znatno bolju uspješnost, što je posljedica činjenice da su sve početne točke iz primjera D blizu rješenja problema.

Na slici 4.4 prikazana su područja konvergencije za metode GN-E i GS-M za slučaj kada je maksimalni broj iteracija 20, odnosno 100. Crvenom bojom označene su one početne točke iz kojih je iterativni proces konvergirao prema rješenju u najviše 20 iteracija, dok su zelenom bojom označene one početne točke iz kojih je iterativni proces konvergirao prema rješenju u najviše 100 iteracija. U slučaju metode GS-M, područje početnih točaka iz kojih je iteracijski proces konvergirao u rasponu od 21 do 100 iteracija, prirodno se nadovezuje na područje točaka iz kojih je proces konvergirao u najviše 20 iteracija, a kod metode GN-E ponašanje je čudnije.

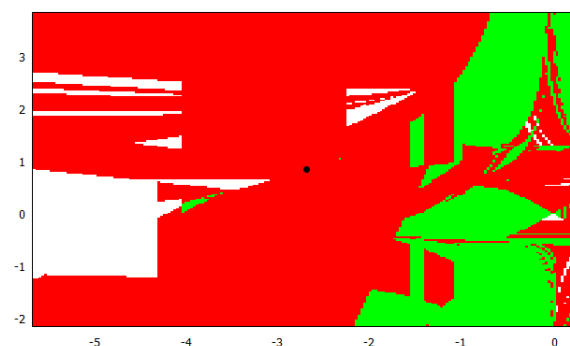
Podatci dobiveni eksperimentom su iskorišteni i za pokušaj određivanja kvalitete početne točke. Početna točka se smatra kvalitetnijom ako je više metoda koji su počeli s njom konvergiralo prema rješenju. U promatranom slučaju, svakoj početnoj točki iz promatrane mreže pridružen je broj između 0 i 11, ovisno o tome koliko je metoda konvergiralo prema rješenju. Prema podacima iz tablice 4.4, za slučaj kada je maksimalni broj iteracija $M_{IT} = 20$, iz 0,3%

Tablica 4.3: Prikaz uspješnosti metoda pri rješavanju problema *leary* u ovisnosti o maksimalnom broju iteracija. Za svaku metodu je prikazano kolika je uspješnost ako je broj iteracija za doseganje rješenja ograničen na 20 (stupac $M_{IT} = 20$), odnosno 100 (stupac $M_{IT} = 100$)

Metoda	$M_{IT} = 20$	$M_{IT} = 100$	Metoda	$M_{IT} = 20$	$M_{IT} = 100$
NWT-E	69,37	69,44	NWT-M	46,39	46,49
GN-E	20,30	57,14	GN-M	81,93	82,08
BGN-E	75,72	75,72	BGN-M	87,42	87,42
GS-E	51,82	51,84	GS-M	70,19	89,53
KO-E	53,44	53,47	KO-M	46,37	46,42
			LP-M	45,50	45,51



(a) metoda GN-E



(b) metoda GS-M

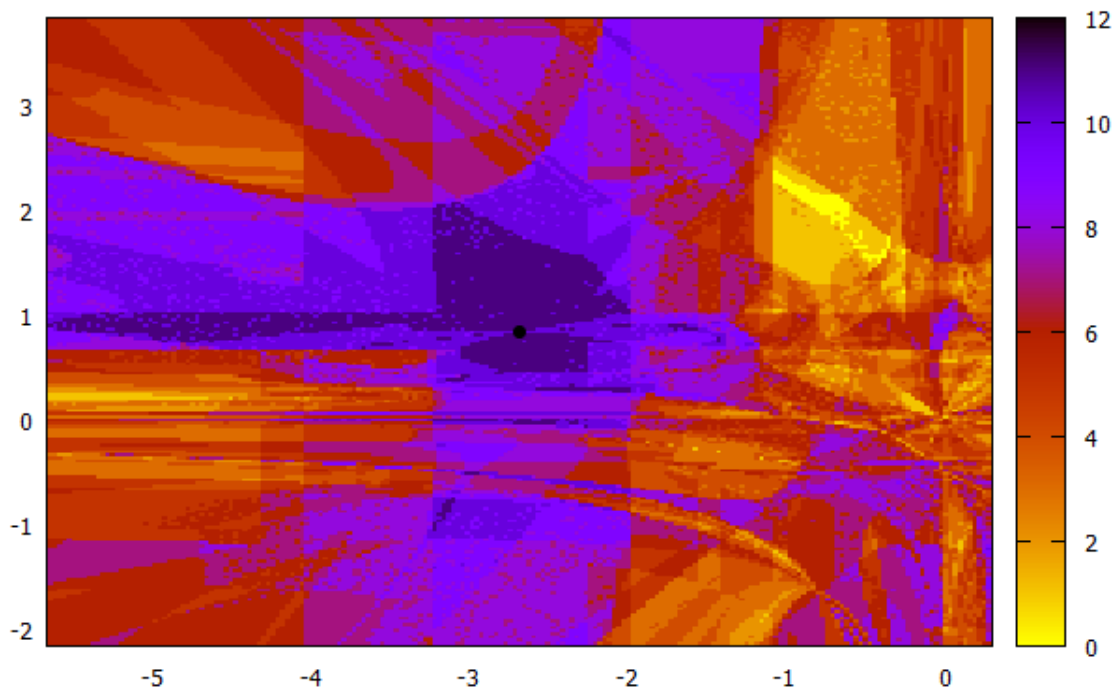
Slika 4.4: Prikaz područja konvergencije metoda GN-E (slika (a)) i GS-M (slika (b)) pri rješavanju problema *leary*. Crvenom bojom su označene početne točke iz kojih je iterativni proces konvergirao rješenju uz najviše 20 iteracija, dok su zelenom bojom označene početne točke iz kojih je iterativni proces konvergirao prema rješenju u najviše 100 iteracija. Crni krug u sredini označava rješenje.

početnih točaka u području D prema rješenju nije konvergirala niti jedna metoda. S druge strane, iz 4,7% početnih točaka prema rješenju su konvergirale sve metode. Kada je $M_{IT} = 100$, tada je broj početnih točaka iz kojih ne konvergira niti jedna metoda zanemariv (4 od 40401).

Na slici 4.5 prikazana je raspodjela početnih točaka na području D prema broju metoda koje su konvergirale prema rješenju \mathbf{x}^* , za slučaj kada je maksimalni broj iteracija bio 20. Zanimljivo je da postoji područje relativno blizu rješenja (označeno žutom bojom) iz kojeg niti jedna od metoda nije uspjela konvergirati. Područje iz kojeg sve metode konvergiraju prema rješenju je očekivano blizu rješenja, ali ima nepravilan oblik, što pokazuje da rješavanje polinomnih sustava čak i kada se započne s početnom točkom blizu rješenja ne mora uvijek konvergirati rješenju.

Tablica 4.4: Raspodjela početnih točaka unutar područja D prema broju metoda koje su konvergirale prema rješenju za maksimalan broj iteracija $M_{IT} = 20$, odnosno $M_{IT} = 100$.

Broj metoda	0	1	2	3	4	5	6	7	8	9	10	11
% ($M_{IT} = 20$)	0,3	1,6	2,4	9,4	8,9	11,9	14,4	15,1	15,3	8,4	7,7	4,7
% ($M_{IT} = 100$)	0,0	0,3	1,4	4,9	10,0	12,0	11,9	13,0	17,5	13,4	8,4	7,3



Slika 4.5: Početne točke unutar područja D prikazane prema broju metoda za koje je iteracijski postupak konvergirao rješenju problema `leary`. Maksimalni broj iteracija 20.

4.1.3 Problem `himmelbaum`

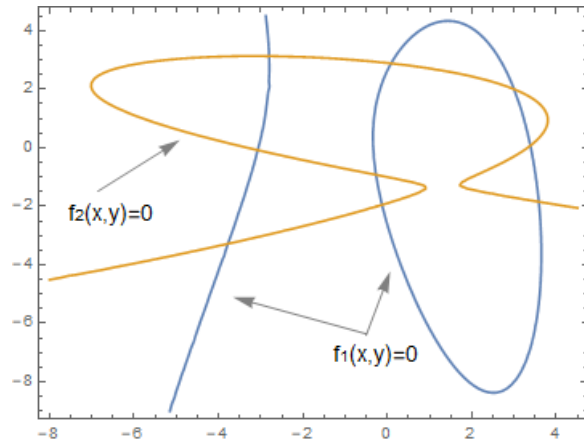
U ovom odjeljku promatra se ponašanje metoda na rješavanju polinomnog sustava `himmelbaum`. Jednadžbe polinoma f_1 i f_2 koji čine sustav i njegova geometrijska interpretacija prikazani su na slici 4.6. Kako se vidi sa slike, krivulje $f_1(x, y) = 0$ i $f_2(x, y) = 0$ se sijeku u 9 točaka. Neka je zadana početna točka $\mathbf{x}^{(0)} = (4, 3)$. Sve promatrane metode najdublje spusta konvergiraju prema nekom rješenju sustava, ali ne istom:

- prema $(3, 2)$ konvergiraju metode GN-E i GN-M,
- prema $(0,087, 2,884)$ konvergiraju metode GS-E, GS-M i KO-E,
- prema $(-0,128, -1,954)$ konvergira metoda KO-M i
- prema $(-0,271, -0,923)$ konvergiraju metode NWT-E, NWT-M, BGN-E, BGN-M i LP-M.

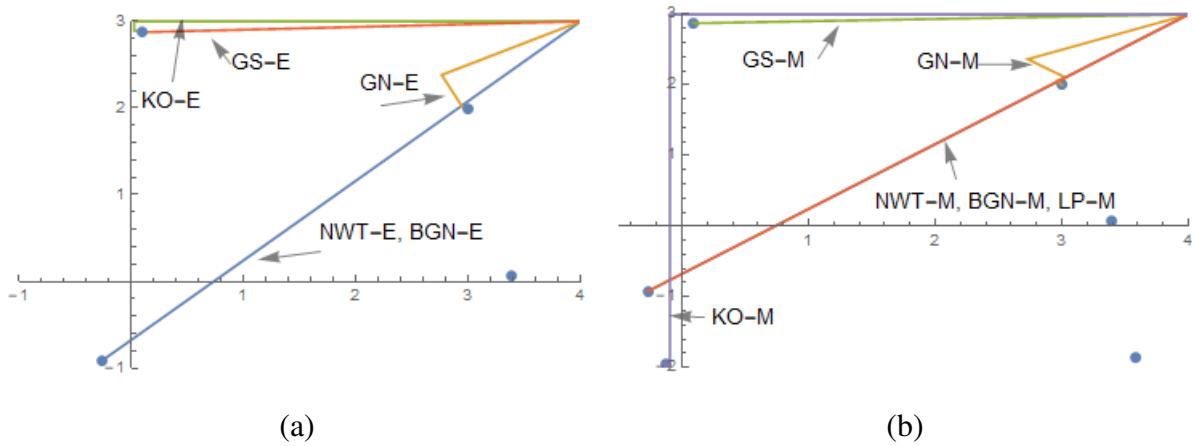
Na slici 4.7 prikazana je putanja po kojoj se kreću iterativni postupci za sve metode najdub-

$$f_1(x,y) = -42x + 2y^2 + 4xy + 4x^3 - 14$$

$$f_2(x,y) = -26y + 2x^2 + 4xy + 4y^3 - 22$$



Slika 4.6: Polinomni sustav himmelbaum i njegova geometrijska interpretacija.



Slika 4.7: Putanje kretanja metoda najdublje spusta na problemu himmelbaum s početnom točkom $\mathbf{x}^{(0)} = (4, 3)$. Na slici (a) prikazane su putanje za metode *-E, dok su na slici (b) prikazane putanje za metode *-M.

ljevog spusta i to tako da su na slici 4.7.a prikazane putanje za metode koje koriste minimizaciju kvadrata euklidske norme reziduala za određivanje duljine spusta (metode *-E), dok su na slici 4.7.b prikazane putanje za metode koje koriste minimizaciju max-norme reziduala za određivanje duljine spusta (metode *-M).

Iz slike se može primijetiti da metode koje koriste isti izbor smjera uglavnom konvergiraju prema istom rješenju polinomnog sustava. Iznimku čine metode koje za smjer spusta koriste smjer koordinatnih osi. Točka $\mathbf{x}^{(1)}$ izračunata metodom KO-E je $\mathbf{x}_E^{(1)} = (0,022, 3)$, dok je prva točka izračunata metodom KO-M $\mathbf{x}_M^{(1)} = (-0,096, 3)$. Razlika nije velika, ali je dovoljna da iterativni postupci konvergiraju svaki prema svojoj nultočki.

4.2 Pregled rezultata eksperimenta

U ovom potpoglavlju daje se pregled rezultata numeričkog eksperimenta u kojem se uspješnost metoda najdubljega spusta uspoređuje s uspješnošću dvije dostupne biblioteke za rješavanje nelinearnih sustava varijantom Newtonove metode: algoritmom MinPack i algoritmom NLEQ, koje se obje smatraju vrlo kvalitetnim bibliotekama za rješavanje nelinearnih sustava.

Biblioteka Minpack [37] je nastala koncem 70-tih godina dvadesetog stoljeća u Argonne National Laboratoriju u SAD-u i sastoji se od potprograma namijenjenih rješavanju nelinearnih sustava odnosno nelinearnog problema najmanjih kvadrata. Nelinearni sustavi se rješavaju korištenjem modifikacije Powellove metode, dok se za rješavanje nelinearnog problema najmanjih kvadrata koristi Levenberg-Marquardtova metoda. Korisnik treba osigurati potprogram za računanje nelinearne funkcije, a može, ali ne mora, osigurati i potprogram za izračun gradijenta funkcije. Ako korisnik ne može osigurati takav potprogram, gradijent se računa numeričkim putem. Kvaliteta programa je potvrđena od strane niza autora (npr. [11]), a programski paket Octave za rješavanje sustava nelinearnih funkcija koristi funkciju `fsolve`, koja je bazirana na potprogramima iz biblioteke MinPack [38]. MinPack je napisan u programskom jeziku Fortran, a postoje i implementacije u drugim programskim jezicima.

Biblioteka NLEQ nastala je desetak godina nakon biblioteke MinPack i dio je biblioteke numeričkih programa CodeLib, razvijene u Konrad Zuse Zentrum u Berlinu, otkuda se može preuzeti ([39]). Kao i MinPack, napisana je u programskom jeziku Fortran. Sam algoritam je zasnovan na afino invarijantnoj prigušenoj Newtonovoj metodi, opisanoj u [12]. Razlika u odnosu na standardne lokalne tehnike Newtonove metode je korištenje specijalne strategije prigušenja u cilju stvaranja globalno konvergentne Newtonove sheme. Ključno svojstvo algoritma NLEQ je afina invarijantnost i ona se nastoji što dalje očuvati u samom algoritmu [40].

Obzirom da su obje biblioteke napravljene na principu *crne kutije*, odnosno dovoljno je pozvati glavni potprogram s odgovarajućim parametrima, a on će vratiti eventualno rješenje problema i još neke dodatne informacije, za potrebe testiranja je napravljen jednostavan program u programskom jeziku Fortran koji služi kao sučelje iz kojeg se poziva glavni potprogram pojedine biblioteke. Promjene originalnog koda bile su minimalne i odnosile su se na način na koji se rutini dostavlja funkcija u kojoj je zapisan problem koji se rješava, odnosno gradijent tog problema.

Usporedba različitih metoda najdubljega spusta i metoda iz biblioteka MinPack i NLEQ bazirana je na uspješnosti rješavanja. Uspješnost rješavanja je testirana na nizu od 103 polinomna sustava koji predstavljaju standardne testove metoda za rješavanje polinomnih sustava.

Većina problema je preuzeta sa stranice [41] na kojoj se nalazi cijela zbirka polinomnih sustava. Na stranici se mogu naći i polinomni sustavi s kompleksnim koeficijentima koji nisu tema ovog rada. Ostali problemi su preuzeti iz razne literature ili iz drugih internetskih izvora,

Tablica 4.5: Raspodjela problema korištenih za testiranje prema dimenziji.

Dimenzija	2	3	4	5	6	7	8	9	10	12
Broj problema	9	14	16	13	16	6	14	7	6	2

a mahom je riječ o raznim test problemima ili primjerima na kojima su ukazani nedostaci pojedinih metoda za rješavanje polinomnih sustava. U tablici 4.7 dan je kratki opis svakog problema i njegova dimenzija. Detaljniji opis problema može se naći u dodatku. Potrebno je napomenuti da je određeni broj testova (pogotovo oni iz PoSSo kolekcije) zapravo namijenjen rješavanju polinomnih sustava metodama računalne algebre (određivanje Gröbnerove baze) i zato mogu biti vrlo teški za razne metode spusta.

Prema dimenziji problema, najviše promatranih problema je bilo dimenzije 4 i 6 (po 16), a najmanje problema je bilo dimenzije 12. Cijela raspodjela problema korištenih za testiranje prema dimenziji prikazana je u tablici 4.5.

Izračun uspješnosti metoda najdublje spusta i metoda iz biblioteka MinPack i NLEQ napravljena je tako da se postupak za rješavanje pojedinog sustava proveo veliki broj puta s različitim početnim točkama. Preciznije, neka je

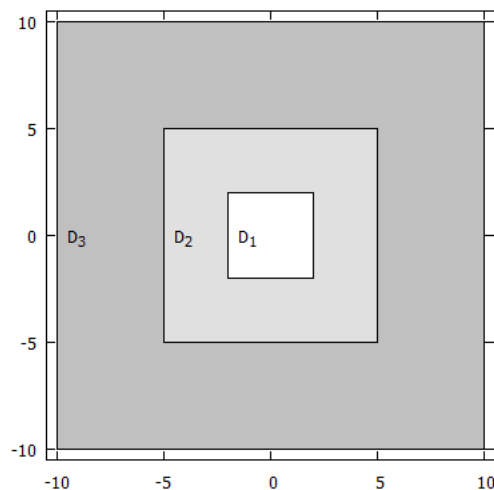
$$K_a = \{\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n : x_i \in [-a, a], \text{ za } i = 1, \dots, n\}$$

kugla oko ishodišta radijusa a u max-normi. Za svaku od dimenzija $n = 2, \dots, 10$ generirane su tri datoteke s točkama iz područja $D_1 = K_2$, $D_2 = K_5 \setminus K_2$ i $D_3 = K_{10} \setminus K_5$ (slika 4.8). Geometrijski, područja D_2 i D_3 su prstenovi oko ishodišta u max-normi s unutarnjim radijusima 2 (za D_2) odnosno 5 (za D_3), te vanjskim radijusima 5 (za D_2) i 10 (za D_3). Jedina iznimka je napravljena za dimenziju $n = 12$, za koju su generirane datoteke iz samo dva područja: $D_1^{12} = K_1$ i $D_2^{12} = K_2 \setminus K_1$. Razlog tome je što u oba primjera za dimenziju 12 rješenja trebaju biti kosinusi i sinusi određenih kutova, što znači da trebaju biti u K_1 .

Detaljan prikaz broja točaka po datotekama dan je u tablici 4.6.

Područja D_1 , D_2 i D_3 su odabrana zato jer u većini primjera je rješenje negdje blizu ishodišta, a cilj je bio utvrditi kako će se metoda ponašati i kada početna točka nije blizu rješenja. U svakom slučaju, svi problemi iste dimenzije su testirani na istom nizu početnih točaka, što je kod nekih problema možda i dovelo do veće uspješnosti rješavanja, ali važno je napomenuti da su sve metode bile u jednakom položaju.

Mjerenje uspješnosti za zadani par (metoda, problem) je napravljeno tako da se zadani problem rješavao zadanom metodom N puta, svaki put za drugu početnu točku iz unaprijed zadanih datoteka. Na kraju je prebrojano koliko puta je iterativni postupak doveo do rješenja, a uspješ-



Slika 4.8: Područja D_1 , D_2 i D_3 na kojima se provodi testiranje za dimenziju $n = 2$.

Tablica 4.6: Broj generiranih točaka po područjima i dimenzijama.

Dimenzija	D_1	D_2	D_3
2	4.000	3.000	3.000
3	2.000	2.000	2.000
4	2.000	1.000	1.000
5,6,7,8	1.000	1.000	1.000
9, 10	500	500	500

nost je definirana kao broj uspješno završenih iterativnih procesa podijeljena s ukupnim brojem započetih iterativnih procesa.

Uspješnost rješavanja svih problema prikazana je u tablicama 4.8–4.15. Rezultati pokazuju da za većinu problema u pravilu uvijek postoji metoda koja problem rješava s uspješnošću većom od 90%. Iznimke se mogu svrstati u nekoliko skupina.

Jednu skupinu čine sustavi koji su posebno dizajnirani da stvaraju probleme. U tu skupinu se mogu ubrojiti problemi graspa i monfroy koji imaju singularna rješenja, kao i problem Rosenbrock, koji je čuveni test problem za razne metode rješavanja nelinearnih sustava, odnosno bezuvjetne minimizacije funkcija više varijabli.

Drugu skupinu čine sustavi nastali u praksi, za čije uspješno rješavanje je potrebno napraviti odgovarajuće skaliranje. Takvo ponašanje pokazuju sustavi combo3000, pb601 i rbp124, kod kojih se uočava drastična razlika u uspješnosti u odnosu na skaliranu verziju.

Treću skupinu čine sustavi za koje nisu dane dobre početne točke. Tipičan primjer je sustav assur44 kod kojeg najbolja metoda ima uspješnost 23,5%. No, kako su rješenja sustava

Tablica 4.7: Kratki opis problema korištenih za uspoređivanje uspješnosti metode najdublje spusta i algoritama za usporedbu. Dio 1 od 3.

Naziv problema	Dim.	Kratki opis
assur44	8	Mehanički problem paralelnih manipulatora
bifurcation	3	Test problem za rješavanje nelinearnih sustava.
bifurcation2	3	Transformirani problem bifurcation
boon	6	Neurofiziološki problem.
brown	5	Brownov 5 dimenzionalni skoro linearni sustav
butcher7	7	Sustav iz test kolekcije PoSSo dimenzije 8, autor Butcher
butcher8	8	Sustav iz test kolekcije PoSSo dimenzije 7, autor Butcher
camera1s	6	Premještanje kamere između dvije pozicije
caprasse	4	Sustav iz test kolekcije PoSSo, autor Caprasse
cassou	4	Sustav iz test kolekcije PoSSo, autori Cassou-Nogue
chandra4	4	Chandrasekhar H-jednadžba za $n=4$
chemequ	5	Kemijska ravnoteža sagorijevanja ugljikovodika
chemequs	5	Problem chemequ skaliran
chemical	3	Kemijski sustav ravnoteže
chemkin	10	Stacionarni problem kemijske kinetike
cohn2	4	Cohn-ove modularne jednadžbe za specijalna polja algebarskih brojeva
cohn3	4	Cohn-ove modularne jednadžbe za specijalna polja algebarskih brojeva
combo3000	10	Model sagorijevanja na temperaturi od 3000 stupnjeva (kemija)
combo3000s	10	Skalirani model sagorijevanja na temperaturi od 3000s stupnjeva (kemija)
cpdm5	5	5-dimenzionalni sustav autora Caprasse i Demaret
cyclic5	5	Ciklički problem 5 korijena
cyclic6	6	Ciklički problem 6 korijena
cyclic7	7	Ciklički problem 7 korijena
cyclic8	8	Ciklički problem 8 korijena
d1	12	Rijetki sustav autora Hong i Stahl
des18_3	8	Problem ""crtež djeteta""
des22_24	10	Problem ""crtež djeteta""
discrete3	8	Sustav iz test kolekcije PoSSo
eco5	5	5-dimenzionalni ekonomski problem
eco6	6	6-dimenzionalni ekonomski problem
eco7	7	7-dimenzionalni ekonomski problem
eco8	8	8-dimenzionalni ekonomski problem
filter9	9	Problem dizajna filtera
fourbar	4	Problem 4 šipke (mehanika)
freudenstein_roth	2	Test problem za rješavanje nelinearnih sustava.
game4two	4	Problem Nashove ravnoteže za 4 igrača s dvije strategije
game5two	5	Problem Nashove ravnoteže za 5 igrača s dvije strategije
game6two	6	Problem Nashove ravnoteže za 6 igrača s dvije strategije
geneig	6	Generalizirani problem svojstvenih vrijednosti
graspa	3	Test problem za rješavanje nelinearnih sustava.
heart	8	Problem srčanog dipola
himmelbaum	2	Test problem za rješavanje nelinearnih sustava.
ipp	8	Problem šest-zglobnog okretnog mehanizma
katsura5	6	Problem magnetizma u fizici reda 5
katsura6	7	Problem magnetizma u fizici reda 6
katsura7	8	Problem magnetizma u fizici reda 7
kearfoot16	6	Test problem za rješavanje nelinearnih sustava.

Tablica 4.7: (nastavak) Kratki opis problema korištenih za uspoređivanje uspješnosti metode najdublje spusta i algoritama za usporedbu. Dio 2 od 3.

Naziv problema	Dim.	Kratki opis
kin1	12	Inverzna kinematika lakat manipulatora.
kinema	9	Problem kinematike robota.
kotsireas	6	Kotsireasov problem iz nebeske mehanike
ku10	10	10-dimenzionalni sustav autora Ku
leary	2	Test problem za rješavanje nelinearnih sustava.
lorentz	4	Točka ekvilibrija 4-dimenzionalnog Lorentzova atraktora
mickey	2	Test problem za rješavanje nelinearnih sustava.
mladenov	10	Nelinearni problem iz elektrotehnike.
monfroy	3	Test problem za rješavanje nelinearnih sustava.
moreno	3	Test problem za rješavanje nelinearnih sustava.
morgan	2	Test problem za rješavanje nelinearnih sustava.
nashequilibria	8	Nashov ekvilibrij za problem iz teorije igara.
noon3	3	Neuralna mreža modelirana pomoću adaptivna Lotka-Volterra sustava
noon4	4	Neuralna mreža modelirana pomoću adaptivna Lotka-Volterra sustava
noon5	5	Neuralna mreža modelirana pomoću adaptivna Lotka-Volterra sustava
pb601	3	Problem 601, izvor iz elektrokemije.
pb601vs	3	Skalirani problem pb601
proddeco	4	Sustav sa strukturom dekompozicije produkta
puma	8	Pozicija ruke i orijentacija PUMA robota
quadfor2	4	Gaussove kvadraturne formule s dva čvora i dvije težine na [-1,1]
quadgrid	5	Interpolacijska kvadraturna formula za funkciju definiranu na mreži
rabmo	9	Optimalna višedimenzionalna kvadraturna formula
rbpl	6	Generička pozicija paralelnog robota
rbpl24	9	Paralelni robot s 24 realna rješenja
rbpl24es	9	Problem rbpl24 skaliran
rbpl24s	9	Problem rbpl24 skaliran
redcyc5	4	Reducirani problem cyclic5
redcyc6	5	Reducirani problem cyclic6
redcyc7	6	Reducirani problem cyclic7
redcyc8	7	Reducirani problem cyclic8
redeco5	5	Reducirani 5-dimenzionalni ekonomski problem
redeco6	6	Reducirani 6-dimenzionalni ekonomski problem
redeco7	7	Reducirani 7-dimenzionalni ekonomski problem
redeco8	8	Reducirani 8-dimenzionalni ekonomski problem
rediff3	3	Trodimenzionalni reakcijsko-difuzijski problem
reimer2	2	2-dimenzionalni sustav čiji je autor Reimer
reimer3	3	3-dimenzionalni sustav čiji je autor Reimer
reimer4	4	4-dimenzionalni sustav čiji je autor Reimer
reimer5	5	5-dimenzionalni sustav čiji je autor Reimer
rose	3	Opći ekonomski problem ravnoteže
rosenbrock	2	Test funkcija za nelinearne sustave i problem optimizacije
s9_1	8	Mali sustav iz područja konstruktivne Galoisove teorije
sendra	2	Sustav iz Posso test kolekcije
solotarev	4	Sustav iz Posso test kolekcije
stewgou40	9	Stewart–Goughova platforma s 40 realnih stanja

Tablica 4.7: (nastavak) Kratki opis problema korištenih za uspoređivanje uspješnosti metode najdublje spusta i algoritama za usporedbu. Dio 3 od 3.

Naziv problema	Dim.	Kratki opis
tangents0	6	Određivanje tangente na 4 jedinične sfere
tangents1	6	Određivanje tangente na 4 jedinične sfere
tangents2	6	Određivanje tangente na 4 jedinične sfere
toms1	2	Test problem za rješavanje nelinearnih sustava.
toms12	3	Test problem za rješavanje nelinearnih sustava.
trinks	6	Sustav iz Posso test kolekcije
utbikker	4	Sustav autora Uteshev i Bikker
virasoro	8	Konstrukcija Virasoro algebri
winkler	3	Test za Grobnerovu bazu
wood	4	Sustav izveden iz optimizacije Woodove funkcije
wright	5	Sustav čiji je autor Wright

assur44 kosinusi i sinusi, logično bi bilo pokušati s početnom točkom čije su komponente u intervalu $[-1, 1]$. Kada se pogleda uspješnost samo za početne točke iz područja K_2 , onda se vidi značajan rast uspješnosti, kod najbolje metode do 30,4%.

Tablica 4.8: Uspješnost rješavanja odabranih polinomnih sustava dimenzije 2 različitim metodama. Najveća uspješnost rješavanja svakog pojedinog sustava je uokvirena.

Problem	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M	KO-E	KO-M	LP-M	MinPack	NLEQ
freudenstein-roth	23,4	23,1	69,9	99,5	87,7	83,5	81,1	100,0	97,2	90,1	27,6	28,5	23,3
himmelbaum	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	51,3	59,8	99,5	96,0
leary	64,9	40,1	56,5	68,6	80,5	70,3	45,5	90,3	31,3	32,0	35,9	6,8	21,9
mickey	63,2	63,2	96,1	100,0	100,0	100,0	99,2	100,0	65,3	65,6	51,4	64,0	63,2
morgan	61,0	61,6	22,5	62,2	61,5	61,4	37,4	38,4	0,0	1,0	41,3	98,3	83,5
reimer2	75,4	55,6	61,9	58,1	95,1	67,2	99,6	80,2	27,5	16,4	8,7	40,6	47,2
rosenbrock	13,1	13,6	1,5	16,0	66,0	17,2	41,0	19,3	0,0	0,0	14,2	31,6	82,5
sendra	77,2	72,9	45,5	62,7	84,4	79,6	90,7	98,1	1,1	28,7	46,1	50,0	37,2
toms1	100,0	100,0	97,6	100,0	100,0	100,0	100,0	100,0	100,0	55,6	66,7	99,4	99,9

Tablica 4.9: Uspješnost rješavanja odabranih polinomnih sustava dimenzije 3 različitim metodama. Najveća uspješnost rješavanja svakog pojedinog sustava je uokvirena.

Problem	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M	KO-E	KO-M	LP-M	MinPack	NLEQ
bifurcation	59,3	65,1	68,7	74,0	65,9	68,3	52,7	96,3	50,2	0,0	57,0	23,8	11,9
bifurcation2	52,2	53,5	69,2	72,3	64,8	65,3	83,3	93,5	55,7	0,2	50,6	37,8	35,2
chemical	90,3	98,6	0,0	93,8	98,2	96,6	93,9	11,4	0,0	0,0	85,2	3,5	26,4
graspa	33,6	35,6	4,2	23,5	42,9	32,5	26,3	25,6	0,0	0,0	29,7	19,1	36,8
monfroy	6,5	7,0	0,0	8,9	6,2	6,3	4,4	5,8	0,0	0,0	3,7	4,0	8,2
moreno	64,5	64,5	82,8	81,1	91,6	79,3	97,4	100,0	95,1	38,1	86,3	68,3	63,9
noon3	86,3	82,3	99,9	99,2	100,0	99,2	93,1	100,0	79,5	0,0	73,5	77,6	78,5
pb601	0,1	3,4	0,0	5,6	0,0	15,3	0,0	0,2	0,0	0,0	8,6	0,0	18,8
pb601vs	22,3	20,2	0,0	19,8	20,6	20,2	11,8	13,4	0,0	0,0	24,1	2,4	28,0
rediff3	76,1	68,8	87,5	100,0	100,0	100,0	91,3	100,0	1,7	0,0	65,6	100,0	60,6
reimer3	33,1	33,5	7,0	39,1	50,4	43,3	57,2	70,1	8,0	0,0	61,5	19,8	27,1
rose	87,0	88,9	19,9	85,2	100,0	86,3	98,5	100,0	25,9	0,0	98,7	52,2	80,1
toms12	89,3	71,9	43,0	78,9	93,2	100,0	96,2	100,0	41,3	1,1	92,2	89,0	31,5
winkler	59,4	54,8	0,2	89,9	76,1	90,6	78,4	77,8	67,2	0,0	85,9	65,3	36,8

Tablica 4.10: Uspješnost rješavanja odabranih polinomnih sustava dimenzije 4 različitim metodama. Najveća uspješnost rješavanja svakog pojedinog sustava je uokvirena.

Problem	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M	KO-E	KO-M	LP-M	MinPack	NLEQ
caprasse	95,2	94,3	33,7	52,3	99,9	96,3	99,5	100,0	35,8	0,1	95,3	80,7	85,3
cassou	3,5	2,4	0,0	0,3	7,4	0,8	1,3	0,2	0,0	0,0	2,6	0,0	0,3
chandra4	99,7	99,9	84,3	94,7	99,5	99,4	99,9	99,2	30,9	0,0	91,6	98,6	99,1
cohn2	96,2	85,0	15,2	73,0	94,1	92,2	98,7	99,6	49,5	0,4	78,4	60,4	7,7
cohn3	90,4	85,3	0,1	25,4	74,6	79,7	82,4	90,9	0,5	0,0	55,2	51,5	37,7
fourbar	9,7	10,5	5,0	42,3	59,5	44,4	35,0	40,2	35,8	13,0	55,7	29,4	4,6
game4two	33,2	34,4	0,4	46,8	35,7	38,2	37,0	67,2	30,5	0,0	52,5	45,7	26,6
lorentz	92,5	90,4	100,0	99,9	100,0	100,0	99,8	100,0	100,0	0,0	98,8	99,7	57,5
noon4	89,2	73,3	98,9	96,3	100,0	93,2	96,2	99,8	78,7	0,0	59,7	83,3	80,4
proddeco	99,0	99,4	62,6	99,1	100,0	99,8	87,1	94,8	61,6	21,2	96,0	90,1	84,3
quadfor2	57,0	65,4	34,9	39,2	72,4	63,3	63,1	75,2	43,7	0,0	80,9	41,3	82,3
redcyc5	99,6	98,7	50,3	99,7	100,0	99,8	99,7	100,0	28,0	0,0	99,1	87,7	56,6
reimer4	11,4	15,3	0,4	14,4	24,9	18,6	34,3	69,4	0,0	0,0	30,8	4,3	11,3
solotarev	100,0	99,9	49,2	100,0	100,0	100,0	100,0	99,9	45,3	0,0	82,7	82,9	66,1
utbikker	70,7	70,4	11,1	71,9	86,1	75,9	71,0	77,1	5,5	0,0	81,0	67,6	63,2
wood	66,4	61,6	29,7	47,5	76,8	60,3	94,8	61,4	2,3	0,0	34,0	54,3	95,5

Tablica 4.11: Uspješnost rješavanja odabranih polinomnih sustava dimenzije 5 različitim metodama. Najveća uspješnost rješavanja svakog pojedinog sustava je uokvirena.

Problem	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M	KO-E	KO-M	LP-M	MinPack	NLEQ
brown	99,4	99,5	27,4	99,6	99,7	99,9	99,8	100,0	7,6	0,0	61,2	86,2	61,8
chemequ	82,7	88,4	0,0	68,3	82,8	81,5	56,5	1,5	0,0	0,0	0,0	0,0	57,0
chemequs	97,5	99,2	0,0	90,8	99,7	96,1	92,6	64,9	0,0	0,0	0,4	0,0	28,1
cpdm5	97,2	97,9	69,2	99,8	100,0	99,8	99,6	100,0	19,0	0,0	90,4	92,2	82,4
cyclic5	77,6	86,4	38,6	69,1	80,8	79,1	58,4	88,1	53,0	0,0	84,0	36,3	36,9
eco5	60,5	50,4	34,4	55,3	74,2	56,2	76,3	78,5	51,4	0,0	27,3	23,7	36,2
game5two	31,5	31,3	1,3	35,0	34,8	36,3	31,7	37,6	7,0	0,0	35,6	18,1	22,4
noon5	88,8	60,6	58,3	81,7	100,0	76,8	93,0	94,3	52,7	0,0	25,8	89,0	81,1
quadgrid	44,5	30,1	0,0	24,2	22,3	22,4	37,2	50,3	0,0	0,0	12,2	7,0	11,7
redcyc6	84,8	86,7	30,5	92,5	96,8	93,6	92,8	96,1	22,5	6,6	98,3	63,8	43,2
redeco5	95,6	88,6	100,0	93,8	100,0	94,6	99,3	100,0	100,0	0,0	84,9	94,5	56,0
reimer5	5,3	5,8	0,1	7,7	13,3	9,4	17,2	30,7	0,0	0,0	16,0	0,0	4,4
wright	100,0	100,0	100,0	100,0	100,0	100,0	97,2	96,9	100,0	0,0	80,9	100,0	97,8

Tablica 4.12: Uspješnost rješavanja odabranih polinomnih sustava dimenzije 6 različitim metodama. Najveća uspješnost rješavanja svakog pojedinog sustava je uokvirena.

Problem	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M	KO-E	KO-M	LP-M	MinPack	NLEQ
boon	92,2	74,8	0,7	81,2	97,2	94,9	97,9	99,3	0,0	0,0	18,0	32,5	76,0
camera1s	28,5	13,5	0,0	7,4	39,4	13,8	6,1	1,3	0,0	0,0	0,1	0,0	38,5
cyclic6	42,6	47,2	1,0	38,6	52,1	42,7	28,4	65,6	37,8	0,0	66,1	9,3	27,0
eco6	38,7	33,9	9,8	37,9	42,4	39,1	47,0	65,3	18,1	0,0	12,7	11,7	24,2
game6two	53,6	50,1	12,5	56,9	77,0	61,4	67,9	83,5	48,5	0,0	62,4	12,3	19,4
geneig	100,0	99,7	33,7	97,1	100,0	99,9	100,0	98,5	52,0	0,0	94,5	62,4	84,4
katsura5	66,5	58,3	76,3	97,8	98,3	97,6	96,6	97,7	23,0	14,5	94,8	77,8	44,0
kearfoot16	72,4	61,9	100,0	97,3	100,0	96,1	100,0	100,0	100,0	0,0	59,3	74,3	26,1
kotsireas	45,1	41,9	0,0	44,1	74,7	55,6	71,7	85,3	1,0	0,0	22,4	23,7	25,1
rbpl	31,1	30,0	2,5	49,6	68,6	49,1	51,9	70,0	0,0	0,0	49,8	18,1	15,3
redcyc7	33,6	38,6	2,1	49,6	55,7	50,0	30,7	33,4	1,7	0,1	96,8	21,5	29,1
redeco6	72,4	68,6	94,1	76,2	100,0	74,5	90,9	98,5	99,9	0,0	68,3	67,4	41,7
tangents0	93,1	93,3	4,0	93,5	98,5	93,0	97,7	99,9	49,0	0,0	90,8	16,5	74,2
tangents1	43,4	41,4	2,0	94,3	100,0	88,3	98,5	99,9	0,3	0,0	47,0	10,3	8,6
tangents2	39,9	38,5	30,3	94,8	100,0	88,8	96,0	100,0	0,2	0,0	55,7	47,4	7,4
trinks	96,3	92,4	0,0	89,4	100,0	95,2	98,4	99,1	0,0	0,0	65,0	2,0	51,8

Tablica 4.13: Uspješnost rješavanja odabranih polinomnih sustava dimenzije 7 različitim metodama. Najveća uspješnost rješavanja svakog pojedinog sustava je uokvirena.

Problem	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M	KO-E	KO-M	LP-M	MinPack	NLEQ
redco7	86,7	81,4	77,8	84,9	100,0	85,4	96,9	99,9	91,1	0,0	67,2	68,6	44,1
redcyc8	13,1	16,1	7,2	17,3	20,5	17,9	11,1	11,1	22,2	2,6	92,8	33,6	25,3
katsura6	51,8	42,4	41,1	82,0	76,8	77,5	71,1	68,1	9,3	17,0	64,9	80,9	30,6
eco7	51,6	42,9	18,0	42,8	68,4	46,0	63,9	68,8	10,5	0,0	9,4	8,8	28,0
cyclic7	19,2	33,9	0,0	26,5	25,5	31,4	13,0	26,7	3,9	0,0	36,6	1,6	16,9
butcher7	2,8	2,6	0,0	4,2	10,4	5,0	4,0	6,5	2,5	0,0	20,5	5,0	6,5

Tablica 4.14: Uspješnost rješavanja odabranih polinomnih sustava dimenzije 8 različitim metodama. Najveća uspješnost rješavanja svakog pojedinog sustava je uokvirena.

Problem	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M	KO-E	KO-M	LP-M	MinPack	NLEQ
assur44	10,7	10,5	0,0	12,4	23,5	15,5	17,0	4,4	0,0	0,0	0,8	0,0	11,9
butcher8	2,7	2,8	0,0	4,7	9,3	5,7	4,4	6,7	0,0	0,0	17,4	2,8	8,4
cyclic8	10,2	19,1	0,0	15,2	17,5	21,6	6,8	25,8	1,2	0,0	32,6	0,2	11,6
des18_3	23,1	22,9	0,0	19,2	37,1	21,4	12,4	15,7	0,0	0,0	7,3	0,0	12,8
discrete3	99,9	99,7	69,4	92,8	99,9	99,4	98,3	95,0	0,0	0,0	66,1	98,0	99,3
eco8	31,4	28,6	9,4	30,6	37,9	31,3	36,2	50,5	0,6	0,0	4,4	4,2	19,6
heart	95,1	85,4	17,4	74,4	92,6	78,7	92,7	61,8	12,4	0,0	25,5	3,3	100,0
ipp	42,9	41,9	31,5	53,0	60,5	53,9	52,5	51,0	20,7	0,0	18,0	30,6	40,3
katsura7	37,5	31,6	37,0	73,4	64,6	65,8	53,6	59,2	5,9	12,6	49,6	37,8	22,4
nashequilibria	63,8	63,9	0,6	23,9	62,1	65,8	72,1	90,5	15,1	0,0	1,6	0,3	21,6
puma	62,6	64,3	47,6	90,5	100,0	89,3	88,7	94,9	39,3	0,0	13,8	17,5	65,4
redeco8	61,6	58,1	53,6	63,3	83,4	61,8	78,0	94,0	86,2	0,0	48,2	38,5	31,7
s9_1	97,4	97,2	1,2	98,3	100,0	99,3	99,1	99,9	0,0	0,0	43,2	16,9	66,9
virasoro	92,2	53,2	96,0	99,1	97,4	83,6	96,8	99,9	71,0	0,0	84,8	99,4	98,8

Tablica 4.15: Uspješnost rješavanja odabranih polinomnih sustava dimenzije 9, 10 i 12 različitim metodama. Najveća uspješnost rješavanja svakog pojedinog sustava je uokvirena.

Dimenzija	Problem	NWT-E	NWT-M	GN-E	GN-M	BGN-E	BGN-M	GS-E	GS-M	KO-E	KO-M	LP-M	MinPack	NLEQ
9	filter9	3,3	3,1	0,3	34,5	91,6	46,7	51,1	56,0	0,0	0,0	0,8	0,2	1,6
9	kinema	45,8	47,3	0,0	40,9	49,6	59,7	58,1	55,3	0,0	0,0	2,7	0,9	49,3
9	rabmo	0,4	0,5	0,0	0,1	0,9	0,2	0,3	0,7	0,0	0,0	0,5	0,0	0,5
9	rbpl24	0,5	0,9	0,0	0,7	0,4	0,5	0,0	0,0	0,0	0,0	0,0	0,0	0,0
9	rbpl24es	72,1	69,8	2,3	90,5	100,0	91,2	93,9	98,2	0,0	0,0	22,8	14,1	57,2
9	rbpl24s	80,1	76,7	0,3	83,0	99,9	87,4	90,3	95,9	0,0	0,0	22,4	5,2	67,3
9	stewgou40	15,7	15,9	17,0	77,9	99,9	82,2	75,1	92,3	0,0	0,0	24,9	4,7	14,9
10	chemkin	22,1	23,6	22,1	57,6	75,0	57,0	55,8	73,1	0,0	0,0	3,4	5,5	15,6
10	combo3000	0,0	0,0	0,0	0,0	0,0	0,0	35,7	33,8	0,0	0,0	14,9	0,3	100,0
10	combo3000s	89,8	85,3	0,1	4,0	91,1	88,9	64,3	61,3	0,0	0,0	6,3	2,4	54,5
10	des22_24	74,7	69,2	0,0	61,9	90,9	66,3	74,5	76,2	0,0	0,0	11,1	0,0	35,5
10	ku10	1,9	4,1	0,0	9,4	10,0	8,7	74,4	99,4	0,0	0,0	0,9	0,1	4,6
10	mladenov	48,1	46,7	1,1	16,0	50,3	22,7	6,2	6,1	0,0	0,0	7,6	8,1	0,2
12	d1	28,0	31,6	7,6	83,2	100,0	81,2	72,8	82,4	0,0	0,0	1,1	0,7	25,5
12	kin1	28,7	29,2	9,2	84,8	100,0	83,8	73,5	81,5	0,0	0,0	1,6	1,7	24,4

Tablica 4.16: Prikaz uspješnosti rješavanja po metodama. Za svaku metodu je dan broj problema kod kojih je uspješnost rješavanja bila u zadanih granicama.

Metoda	Uspješnost rješavanja					
	$\geq 90\%$	70% - 90%	50% - 70%	30% - 50%	10% - 30%	$< 10\%$
NWT-E	24	19	18	17	14	12
NWT-M	18	17	23	21	14	11
GN-E	11	5	11	14	12	51
GN-M	30	20	13	15	14	12
BGN-E	48	16	15	8	10	7
BGN-M	31	23	18	12	11	9
GS-E	43	17	14	12	8	10
GS-M	49	14	16	6	8	11
KO-E	11	4	9	13	12	55
KO-M	1	0	4	2	7	90
LP-M	15	12	21	13	21	22
MinPack	12	12	13	12	17	38
NLEQ	9	14	16	19	32	14

U tablici 4.16 za svaku metodu je prikazano na koliko problema je ostvarila određenu uspješnost. Iz tablice se može vidjeti da su po ovom kriteriju daleko najuspješnije metode BGN-E, GS-E i GS-M koje su na više od 40 problema ostvarile uspješnost rješavanja veću od 90%. Drugu grupu po ovom kriteriju čine metode BGN-M i GN-M koje obje imaju oko 30 problema kod kojih je uspješnost rješavanja veća od 90%. To zapravo i nije iznenađenje jer metoda GN-M često uzima Newtonov smjer za smjer spusta (potpoglavlje 3.1.3). U usporedbi s referentnim algoritmima MinPack i NLEQ, i metode kojima je smjer spusta Newtonov smjer također pokazuju bolju uspješnost, dok su ostale metode uglavnom nešto lošije od referentnih metoda. Zanimljivo je primijetiti i da je metoda KO-E, u kojoj je smjer spusta jednak smjeru neke koordinatne osi, po ovom kriteriju usporediva s metodom GN-E. S druge strane, metoda KO-M se pokazala daleko najlošijom.

Gledano prema načinu određivanja duljine spusta, određivanje pomoću kvadrata euklidske norme reziduala daje bolje rezultate u slučaju korištenja Newtonova smjera, kombinacije Newtonova smjera i smjera najstrmijeg spusta te smjera koordinatnih osi. S druge strane, određivanje duljine spusta pomoću max-norme reziduala je bolje kod Gauss-Seidelova smjera. Bolji rezultat metode GN-M u odnosu na GN-E više je posljedica načina izbora smjera nego načina određivanja duljine spusta. Činjenica da je metoda GS-M uspješnija od metode GS-E ukazuje na opravdanost uvođenja max-norme, iako je određivanje duljine spusta pomoću nje složenije.

Detaljnija usporedba metoda je napravljena u sljedećem potpoglavlju.

4.3 Određivanje indeksa uspješnosti

U tablicama 4.8–4.15 prikazana je uspješnost svake metode na svakom problemu. U tablici 4.16 je za svaku metodu prikazano koliko problema rješava s određenim postotkom uspješnosti. No, da bi bolje mogli usporediti dvije metode potrebno je razviti novi pristup. Zbog toga je potrebno razviti metodologiju kojom će se svakoj metodi dodijeliti jedan broj – indeks uspješnosti metode, koji će se kasnije koristiti za uspoređivanje kvalitete različitih metoda.

Obzirom na prirodu problema, svaki od indeksa uspješnosti je nužno povezan s kolekcijom polinomnih sustava na kojima je mjerena uspješnost te metodama koje će se uspoređivati. Formalno, neka je S označena kolekcija polinomnih sustava (problema), te neka je M skup metoda za koje je poznata uspješnost na problemima iz kolekcije S . Neka je $s_{mt,pr}$ označena uspješnost metode mt na problemu pr .

Razvijena su tri indeksa za određivanje uspješnosti metoda na zadanoj kolekciji problema S . Sva tri indeksa bazirana su na odgovarajućim funkcijama čija vrijednost ovisi o uspješnosti metode na odabranom problemu, i to obzirom na sve ostale metode iz skupa M .

Indeksi su:

- indeks maksimalne uspješnosti, I_M , koji broji koliko puta je zadana metoda bila najuspješnija među svim metodama na odabranoj kolekciji problema S ;
- indeks međusobne uspješnosti, I_B , kojom se za svaku metodu broji koliko je puta bila uspješnija od svih drugih metoda na svim problemima iz odabrane kolekcije S ;
- indeks poretka, I_R , koji se dobije zbrajanjem vrijednosti indeksa poretka metode za svaki probleme iz kolekcije S . Indeks poretka metode za određeni problem korespondira s rangom uspješnosti metode u odnosu na ostale metode iz M .

4.3.1 Indeks maksimalne uspješnosti

Za izračun indeksa maksimalne uspješnosti metode mt na kolekciji S u odnosu na metode iz skupa M , potrebno je prvo definirati funkciju $i_M(mt; pr)$ koja ima vrijednost 1, ako je uspješnost metode mt na problemu $pr \in S$, $r_{mt,pr}$, jednaka maksimalnoj uspješnosti za sve metode na problemu pr :

$$i_M(mt; pr) = \begin{cases} 1, & r_{mt,pr} = \max\{r_{m,pr} : m \in M\}, \\ 0, & \text{inače.} \end{cases}$$

Za funkciju i_M vrijedi $\sum_{m \in M} i_M(m; pr) \geq 1$, odnosno moguće je da više metoda imaju uspješnost jednaku maksimalnoj uspješnosti za neki problem. U tom slučaju za sve te metode vrijednost od i_M je 1.

Indeks maksimalne uspješnosti definiran je kao zbroj vrijednosti funkcije i_M za sve pro-

Tablica 4.17: Indeks maksimalne uspješnosti svih metoda na kolekcijama $S_{2,3}$, $S_{4,5}$, $S_{6,7,8}$, $S_{9,10,12}$ i $S_{sve} = S_{2,3} \cup S_{4,5} \cup S_{6,7,8} \cup S_{9,10,12}$, pri čemu je $S_{i,j}$ skup svih problema reda i i j . Broj u zagradama u zaglavlju stupca predstavlja ukupan broj problema u promatranoj kolekciji.

	$S_{2,3}$ (23)	$S_{4,5}$ (29)	$S_{6,7,8}$ (36)	$S_{9,10,12}$ (15)	S_{sve} (103)
BGN-E	7	13	15	11	46
GS-M	14	15	13	1	43
BGN-M	6	3	0	1	10
GN-M	5	2	2	0	9
LP-M	0	1	7	0	8
NWT-M	3	3	0	1	7
GS-E	3	2	2	0	7
NLEQ	3	2	1	1	7
NWT-E	2	2	2	0	6
KO-E	2	3	1	0	6
GN-E	1	3	1	0	5
MinPack	2	1	0	0	3
KO-M	0	0	0	0	0

bleme iz kolekcije S :

$$I_M(mt) = \sum_{p \in S} i_M(mt; p).$$

Pregled indeksa maksimalne uspješnosti dan je u tablici 4.17. Indeks je dan po kolekcijama $S_{2,3}$, $S_{4,5}$, $S_{6,7,8}$, $S_{9,10,12}$ te $S_{sve} = S_{2,3} \cup S_{4,5} \cup S_{6,7,8} \cup S_{9,10,12}$, gdje $S_{2,3}$ označava skup svih problema reda 2 i 3, $S_{4,5}$ skup svih problema reda 4 i 5, itd. Skup S_{sve} označava skup svih problema.

Prema indeksu maksimalne uspješnosti na S_{sve} najbolja metoda je BGN-E ($I_M(\text{BGN-E}) = 46$), a slijedi je metoda GS-M koja na kolekciji S_{sve} ima $I_M(\text{GS-M}) = 43$. Ostale metode imaju bitno manji indeks maksimalne uspješnosti. Ako se gledaju problemi manjeg reda (kolekcije $S_{2,3}$ i $S_{4,5}$) uspješnija je metoda GS-M, a kod problema višeg reda (kolekcije $S_{6,7,8}$ i $S_{9,10,12}$) bolji rezultat postiže metoda BGN-E. Vrijedi istaknuti da je $I_M(\text{LP-M}) = 7$ na $S_{6,7,8}$, ali se čini da je to prvenstveno zato jer se metoda LP-M dobro ponaša na grupi problema `cyclic*`, `redcyc*` i `butcher*`. Ovaj rezultat ukazuje na slabost indeksa maksimalne uspješnosti, pogotovo ako je primijenjen na kolekciji u kojoj je mali broj problema, a među njima ih nekoliko ima sličnu strukturu. Nadalje, indeks maksimalne uspješnosti je dobar u situaciji kada postoje jedna ili dvije dominantne metode, ali ako bi bilo više dominantnih metoda, indeks nije dovoljno fin za usporedbu, stoga je potrebno promijeniti pristup i koristiti druge indekse.

4.3.2 Indeks međusobne uspješnosti

Indeks međusobne uspješnosti, I_B , za zadanu metodu mt se definira tako da se za svaki problem u kolekciji i u odnosu na ostale metode iz M pregleda u koliko slučajeva je metoda mt imala bolju uspješnost. Za izračun indeksa međusobne uspješnosti uvodi se pomoćna funkcija $i_B(mt; pr)$ kojom se utvrđuje od koliko drugih metoda metoda mt ima bolju uspješnost pri rješavanju problema pr :

$$i_B(mt; pr) = \sum_{\substack{m \in M \\ m \neq mt}} \begin{cases} 1, & r_{mt,pr} > r_{m,pr}, \\ 0, & \text{inače.} \end{cases} \quad (4.1)$$

Sada je indeks međusobne uspješnosti za kolekciju S definiran kao zbroj vrijednosti funkcije i_B za sve probleme iz S :

$$I_B(mt) = \sum_{p \in S} i_B(mt; p).$$

Ako je n_M broj metoda u skupu svih metoda koje se uspoređuju, a n_S je broj svih problema u kolekciji S , tada je

$$0 \leq I_B(mt) \leq (n_M - 1) \cdot n_S,$$

jer je $(n_M - 1) \cdot n_S$ ukupan broj usporedbi metode mt s drugim metodama iz M na svim problemima iz S . Također, iz definicije i_B slijedi

$$0 \leq i_B(mt; pr) \leq n_M - 1 \quad \text{i} \quad \sum_{m \in M} i_B(m; pr) \leq \frac{n_m \cdot (n_m - 1)}{2},$$

s time da se jednakost postiže samo ako među metodama nema jednako uspješnih na problemu pr .

Vrijednosti indeksa međusobne uspješnosti za slučaj kada se promatraju svi problemi dana je u prvom stupcu tablice 4.18.a. U odnosu na indeks maksimalne uspješnosti (tablica 4.17) može se uočiti da je došlo do dosta promjena. Prva tri i posljednje mjesto su isti kao i za slučaj indeksa maksimalne uspješnosti, ali sredina tablice se dosta promijenila: na četvrtom mjestu se nalazi metoda GS-E, a na petom mjestu je metoda GN-M. Kako je već spomenuto u poglavlju 3.1.3, metoda GN-M češće koristi za smjer spusta Newtonov smjer pa je očekivano da daje bolje rezultate nego li metoda GN-E koja je treća po uspješnosti od kraja, a od nje su neuspješnije samo metode koje koriste koordinatne osi kao smjer spusta. Metode NWT* i LP-M su se pokazale boljima od metode NLEQ, koja se pokazala znatno boljom od metode MinPack.

Indeks međusobne uspješnosti također ima svoje nedostatke. Npr. za problem morgan, uspješnost metode NWT-M je za 0,6% bolja od uspješnosti metode NWT-E i stoga je NWT-M bolja od NWT-E. Naravno, postavlja se pitanje je li to zbog izbora točaka, jer različite metode imaju različita područja konvergencije (poglavlje 4.1) tako da bi drugi izbor točaka možda donio prevagu

Tablica 4.18: Indeks međusobne uspješnosti metoda za slučaj usporedbe na kolekciji svih problema (tablica (a)), odnosno samo na kolekciji problema reda 2 (tablica (b)).

Metoda	I_B	I_B^1	I_B^3	Metoda	I_B^E	I_B	I_B^1	I_B^3
BGN-E	1010	924	878	GS-M	76	76	70	67
GS-M	954	889	822	BGN-E	75	77	71	70
BGN-M	847	767	706	BGN-M	65	66	61	55
GS-E	820	773	722	GS-E	65	65	62	61
GN-M	758	697	637	GN-M	62	66	59	52
NWT-E	687	616	554	NWT-E	45	46	41	40
NWT-M	648	589	524	MinPack	40	43	39	38
LP-M	567	519	485	GN-E	39	39	38	36
NLEQ	489	460	426	NWT-M	37	40	33	32
MinPack	374	335	295	NLEQ	37	38	36	35
GN-E	279	245	217	KO-E	27	28	25	20
KO-E	230	211	183	LP-M	21	23	21	19
KO-M	27	24	18	KO-M	19	21	18	13

(a)

(b)

metodi NWT-E.

Stoga se koristi i modificirani indeks međusobne uspješnosti I_B^d , u čijem se izračunu podrazumijeva da su dvije metode jednako uspješne ako se njihova uspješnost na zadanom problemu razlikuje manje od d . Za računanje I_B^d koristi se modificirana funkcija i_B^d definirana s

$$i_B^d(mt; pr) = \sum_{\substack{m \in M \\ m \neq mt}} \begin{cases} 1, & r_{mt,pr} > r_{m,pr} + d, \\ 0, & \text{inače.} \end{cases} \quad (4.2)$$

Naravno, sada se nameće pitanje odabira vrijednosti d . Najjednostavnija varijanta je odabir vrijednosti $d = 1$ ako se uspješnost mjerila na velikom broju početnih točaka (barem 3000), a ako je broj početnih točaka manji, bolje bi bilo uzeti veću vrijednost, npr. $d = 3$.

U drugom i trećem stupcu tablice 4.18.a prikazani su indeksi I_B^1 i I_B^3 . U odnosu na primjenu indeksa I_B , nema razlika osim što je kod I_B^1 i I_B^3 na trećem mjestu metoda GS-E, a na četvrtom BGN-M. Zbog načina na koji su formirani, treba vrijediti $I_B^{d_1}(mt) > I_B^{d_2}(mt)$ kada je $d_1 < d_2$, što se može vidjeti i u tablici.

Pravi način za određivanje je li jedna metoda statistički uspješnija od druge na nekom promatranom problemu bi bilo provođenje McNemarova testa za razliku proporcija zavisnih uzoraka [42]. Test se sastoji od ponavljanja iteracijskog postupka za jednake početne točke različiti-

tim metodama mt_A i mt_B , a početne točke se onda razvrstavaju u četiri skupine:

- obje metode uspjele doći do rješenja,
- obje metode nisu uspjele doći do rješenja,
- samo je metoda mt_A došla do rješenja,
- samo je metoda mt_B došla do rješenja.

Iz podataka o broju elemenata u svakoj od skupina formira se 2×2 tablica

	Metoda mt_A uspjela	Metoda mt_B uspjela
Metoda mt_B uspjela	n_{11}	n_{12}
Metoda mt_B nije uspjela	n_{21}	n_{22}

a test veličina je $z^2 = \frac{(n_{12} - n_{21})^2}{n_{12} + n_{21}}$, koja se ponaša kao χ^2 distribucija s jednim stupnjem slobode. Pomoćna funkcija za izračun indeksa i_B^E se definira s

$$i_B^E(mt; pr) = \sum_{\substack{m \in M \\ m \neq mt}} \begin{cases} 1, & r_{mt,pr} \text{ je statistički značajno veći od } r_{m,pr}, \\ 0, & \text{inače,} \end{cases} \quad (4.3)$$

a sam indeks I_B^E se računa kao $I_B^E(mt) = \sum_{p \in S} i_B^E(mt; p)$.

U tablici 4.18.b prikazani su indeksi međusobne uspješnosti metoda na kolekciji problema dimenzije 2. Prikazani su indeksi I_B^E , I_B , I_B^1 i I_B^3 . Razlike u vrijednosti indeksa su očekivane, ali bitno je primijetiti da je poredak metoda uglavnom isti, bez obzira koja se metoda koristi, a to i je bio cilj.

Bez obzira koji se indeks međusobne uspješnosti koristi, ako se želi usporediti razlika između dvije metode m_1 i m_2 indeksom međusobne uspješnosti, umjesto apsolutne razlike indeksa $I_B(m_1) - I_B(m_2)$ bolje je koristiti normiranu razliku

$$\frac{I_B(m_1) - I_B(m_2)}{(n_M - 1) \cdot n_S},$$

pogotovo ako se radi usporedba na dvije različite kolekcije problema S_1 i S_2 . Npr. ako se žele usporediti metode BGN-E i GS-M indeksom I_B na kolekciji svih problema (tablica 4.18.a, ukupno 103 problema) i na kolekciji problema reda 2 (tablica 4.18.b, ukupno 9 problema), onda je razlika između njih na kolekciji svih problema $(1010 - 954)(12 \cdot 103) = 0,0453$, a na kolekciji problema reda 2 razlika iznosi $(77 - 76)(12 \cdot 9) = 0,0093$, pa se može reći da je BGN-E malo bolja od GS-M na problemima reda 2, ali gledano na svim problemima BGN-E je znatno bolja od GS-M.

4.3.3 Indeks poretka

Posljednji indeks koji se promatra je indeks poretka. Ideja indeksa poretka je da naglasi razliku između metoda u slučaju kada postoji više metoda koje dijele jednaki rezultat, što indeks međusobne uspješnosti ne radi u dovoljnoj mjeri, a to se pogotovo vidi kada se za uspoređivanje uspješnosti koristi statistička razlika, odnosno modificirano uspoređivanje uz zadanu vrijednost d .

Indeks poretka se računa se, kao i druga dva indeksa, prvo računa za određeni problem, a onda se zbroje vrijednosti indeksa za sve probleme iz kolekcije. Vrijednost indeksa poretka metode mt za određeni problem pr se dobije koristeći sljedeći postupak:

- Prvo se izračuna vrijednost $i_M(m; pr)$ za sve metode i promatrani sustav pr . Za računanje i_M se mogu koristiti sve varijante (statistička razlika uspješnosti, obična razlika uspješnosti, modificirana razlika uspješnosti).
- Potom se metode sortiraju silazno prema vrijednosti $i_M(m; pr)$.
- Neka je n_M ukupan broj metoda koje se uspoređuju. Za najbolju metodu je i_R jednak $n_M - 1$, za drugu najbolju metodu je i_R jednak $n_M - 2$, ..., za najlošiju metodu je i_R jednak 0.
- Ako postoji više metoda koje imaju jednaku vrijednost i_M na promatranom problemu, onda se i_R odredi metodom usrednjavanja.

Npr. za problem *mickey* (tablica 4.8), postoje četiri metode koje imaju uspješnost 100% i tri metode koje imaju uspješnost 63,2%. Cijeli postupak određivanja i_R se može predstaviti tablično:

Metode	i_M	Poredak	i_R	Metode	i_M	Poredak	i_R
GN-M, BGN-E, BGN-M, GS-M	9	1.-4.	10,5	KO-E	5	8.	5
GS-E	8	5.	8	MinPack	4	9.	4
GN-E	7	6.	7	NWT-E, NWT-M, NLEQ	1	10.-12.	2
KO-M	6	7.	6	LP-M	0	13.	0

Vrijednost od i_R za četiri najbolje metode je 10,5, odnosno srednja vrijednost brojeva 12, 11, 10 i 9, koji se dodjeljuju za četiri najbolje metode u slučaju kada je ukupan broj metoda 13. Slično, metodama NWT-E, NWT-M i LP-M se dodjeljuje vrijednost 2, kao srednja vrijednost brojeva 3, 2 i 1, koji odgovaraju poretku metoda.

Indeks poretka za metodu mt se odredi kao zbroj vrijednosti funkcije i_R za metodu mt i sve probleme iz promatrane kolekcije,

$$I_R(mt) = \sum_{p \in S} i_R(mt; p).$$

Tablica 4.19: Usporedba uspješnosti različitih metoda najdubljega spusta za rješavanje polinomnih sustava pomoću indeksa poretka. Usporedba je rađena na tri kolekcije sustava: S_M je skup svih sustava čija je dimenzija manja ili jednaka 5, kolekciju S_V čine sustavi čija je dimenzija veća ili jednaka 6, a kolekcija $S = S_M \cup S_V$ predstavlja sve sustave. Usporedba je rađena s indeksima I_R i I_R^1 .

Metode	I_R			I_R^1		
	S_M	S_V	S	S_M	S_V	S
BGN-E	498,5	543,5	1042,0	486,5	531,5	1018,0
GS-M	491,5	487,0	978,5	481,5	488,0	969,5
BGN-M	442,5	429,5	872,0	432,0	428,5	860,5
GS-E	416,0	424,0	840,0	416,5	429,0	845,5
GN-M	386,5	393,5	780,0	390,5	390,0	780,5
NWT-E	354,5	353,0	707,5	356,0	343,5	699,5
NWT-M	340,0	327,0	667,0	345,5	321,0	666,5
LP-M	297,0	276,5	573,5	296,0	265,0	561,0
NLEQ	228,5	267,0	495,5	232,0	268,5	500,5
MinPack	217,0	183,0	400,0	230,5	181,5	412,0
GN-E	183,5	141,0	324,5	182,0	146,5	328,5
KO-E	161,5	124,0	285,5	163,0	138,0	301,0
KO-M	39,0	29,0	68,0	44,0	47,0	91,0

Za indeks poretka vrijedi

$$0 \leq I_R(mt) \leq (n_M - 1) \cdot n_S \quad \text{i} \quad \sum_{m \in M} I_R(m) = n_S \cdot \frac{(n_M - 1)(n_M - 2)}{2}.$$

Kako je već spomenuto, izračun indeksa poretka ovisi o načinu na koji se određuje koja je od dvije metode uspješnija na promatranom problemu. U tom kontekstu, ako se koristi obična usporedba uspješnosti (4.1), indeks poretka označava se s I_R , ako se koristi statistička usporedba (4.3), oznaka je I_R^E , a ako se koristi usporedba s odmakom d (4.2) onda se koristi oznaka I_R^d .

U tablici 4.19 dan je pregled indeksa poretka I_R i I_R^1 na kolekcijama polinomnih sustava S_M , S_V i $S = S_M \cup S_V$. S_M je skup svih sustava čija je dimenzija manja ili jednaka 5, skup S_V čine sustavi čija je dimenzija veća ili jednaka 6, a S je skup svih sustava. Rezultati iz tablice su u skladu s rezultatima dobivenih i s prijašnjim indeksima i pokazuju da je metoda koja u pravilu ima najveću uspješnost BGN-E, a slijedi je metoda GS-M. U usporedbi s metodama MinPack i NLEQ, većina metoda najdubljega spusta pokazuje bolje rezultate, a iznimke su metode koje koriste koordinatne osi kao smjer spusta te metoda GN-E koja koristi smjer najstrijemijeg spusta, koja je problematična zbog spore konvergencije.

Poglavlje 5

Zaključak

Polinomni sustavi se pojavljuju i imaju važnu ulogu u brojnim područjima znanosti. Problem traženja rješenja polinomnih sustava se smatra numerički izazovnim za opće metode rješavanja nelinearnih sustava i stoga su razvijene brojne metode za njihovo rješavanje.

U ovom radu je predstavljena metoda najdublje spusta za rješavanje polinomnih sustava. Osnovna novost u metodi najdublje spusta u odnosu na standardne metode spusta je da se kod nje duljina smjera spusta određuje kao točka u kojoj odgovarajuća funkcija (u radu se promatra ili kvadrat euklidske norme reziduala ili max-norma reziduala) poprima najmanju vrijednost za razliku od metoda spusta kod kojih se duljina spusta određuje pomoću neke aproksimativne metode (najčešće linijskog pretraživanja). Polinom je funkcija pomoću koje se određuje duljina spusta za slučaj polinomnog sustava pa se za nju relativno jednostavno može odrediti točka u kojoj postiže najmanju vrijednost.

Metoda najdublje spusta je, zapravo, skup metoda koji se razlikuju po odabiru smjera spusta te načina na koji se određuje duljina spusta. Smjer spusta može biti krajnje jednostavan, kao što je smjer koordinatnih osi (metode KO-E i KO-M), odnosno neki od standardnih smjerova spusta koji se koriste kod običnih metoda spusta kao što su Newtonov smjer (metode NWT-E i NWT-M) ili smjer najstrmijeg spusta (metode GN-E i GN-M) smjerova. Obzirom da rješavanje korištenjem i Newtonova smjera i smjera najstrmijeg spusta ima određene slabosti, razvijena je metoda koja u svakom koraku iterativnog postupka odabire onaj smjer koji daje veći napredak prema rješenju (metode BGN-E i BGN-M). Od nestandardnih smjerova, prikazano je određivanje smjera na način koji oponaša Gauss-Seidelovu iterativnu metodu za rješavanje linearnih sustava (metode GS-E i GS-M) te metoda u kojoj se smjer određuje rješavanjem zadatke linearnog programiranja.

Kako su linearni sustavi jedna podklasa polinomnih sustava, metode najdublje spusta se mogu primijeniti i na njih, ali jedino primjena metode KO-E ima smisla, jer je za nju lako odrediti i smjer spusta i duljinu spusta. Za varijaciju KO-E metode u kojoj se kao smjerovi spusta ciklički biraju smjerovi koordinatnih osi dokazano je da konvergira ako sustav ima rješenja. Sama

metoda se ne može mjeriti sa specijaliziranim metodama za rješavanje linearnih sustava, ali postoje naznake da bi na rijetkim sustavima mogla biti efikasna.

Korištenje max–norme za određivanje duljine spusta je jedna od novosti ovog rada. Premda rezultati ne sugeriraju da je njezino korištenje za određivanje duljine spusta bolje od korištenja euklidske norme, uspješnost metode GS–M pokazuje ispravnost njezina korištenja. Ipak, glavna prepreka korištenju max–norme je složen postupak izračuna točke u kojoj se postiže minimum.

Da bi se ispitala kvaliteta rješavanja polinomnih sustava metodom najdublje spusta, napravljeno je testiranje. Stvorena je kolekcija od više od 100 polinomnih sustava koji se pojavljuju u raznim kolekcijama test problema za numeričke ili optimizacijske metode, a potom su sustavi rješavani veliki broj puta na unaprijed zadanom skupu početnih točaka. Rezultati rješavanja sustava metodama najdublje spusta su uspoređeni s rezultatima koje su postigli dva dobro poznata algoritma za rješavanje nelinearnih sustava, MinPack i NLEQ na istim problemima i s istim početnim točkama. Dobiveni rezultati pokazuju da većina metoda najdublje spusta ima znatno veću uspješnost u rješavanju polinomnih sustava u odnosu na referentne algoritme MinPack i NLEQ, a to se posebno vidi u slučaju kada početna točka nije blizu rješenja.

Određivanje smjera spusta rješavanjem problema linearnog programiranja (metoda LP–M) je još jedna novina koju donosi ovaj rad. Trenutno je njezin najveći nedostatak je brzina, ali je zanimljivo da je po uspješnosti, gledano po svim problemima, bolja od oba referentna algoritma. Osobito je uspješnija od algoritma Minpack, dok je od algoritma NLEQ znatno bolja na polinomnim sustavima višeg reda.

Metode najdublje spusta pronalaze jedno realno rješenje polinomnog sustava. Ako je poznato da sustav ima više realnih rješenja, dobra taktika za pronalaženje svih rješenja može biti rješavanje sustava s različitim početnim točkama i ponavljanje pokusa dok se ne dođe do svih potrebnih rješenja. Obzirom na visoku uspješnost metoda, čak i u slučaju kada početne točke nisu prilagođene problemu, za očekivati je da se sva realna rješenja mogu naći jako brzo. Veći problem predstavlja situacija kada je potrebno naći sva realna rješenja polinomnog sustava, a nije unaprijed poznato koliko ih ima. U tom slučaju je najbolje koristiti metodu najdublje spusta u kompleksnom području i prebrojiti sva različita rješenja dok se ne dođe do broja rješenja koje predviđaju Bezoutov i Bronsteinov teorem.

Najbolje metode BGN–E i BGN–M bi se mogle ubrzati paralelizacijom, jer su postupci za određivanja smjera spusta potpuno neovisni. Time bi se postiglo značajno ubrzanje jer preostali dijelovi algoritma nisu računski zahtjevni. Dapače, metode bi se mogle proširiti i s još nekim smjerom pretraživanja čime bi se isto mogla popraviti uspješnost.

Rezultati primjene metoda najdublje spusta pokazuju dobre rezultate u odnosu na referentne metode spusta. S druge strane, usporedba metoda najdublje spusta s metodom homotopnog kontinuiteta, odnosno algoritmima koji koriste prikaz polinomnog sustava u Gröbnerovoj bazi, nije direktno moguća. Metoda homotopije pronalazi sva rješenja polinomnog

sustava, realna i kompleksna, što u nekim situacijama može biti previše, pogotovo ako je broj realnih rješenja značajno manji od broja kompleksnih rješenja. Metode zasnovane na prikazu polinomnog sustava u Gröbnerovoj bazi također daju sva rješenja, ali njihova implementacija u aritmetici pokretnog zarez je nepouzdana, osim u slučaju kada se rješavaju polinomni sustavi s racionalnim koeficijentima u aritmetici visoke preciznosti.

Primijenjene na odgovarajućim problemima, metode najdublje spusta pokazuju izvrsne rezultate i mogu se koristiti u praksi.

Literatura

- [1] Meintjes, K., Morgan, A. P., “A methodology for solving chemical equilibrium systems”, *Applied Mathematics and Computation*, vol. 22, no. 4, 1987, str. 333–361.
- [2] Tsai, L.-W., Morgan, A. P., “Solving the kinematics of the most general six-and five-degree-of-freedom manipulators by continuation methods”, *Journal of Mechanical Design*, vol. 107, no. 2, 1985, str. 189–200.
- [3] Morgan, A., Sommese, A., “Computing all solutions to polynomial systems using homotopy continuation”, *Applied Mathematics and Computation*, vol. 24, no. 2, 1987, str. 115–138.
- [4] Verschelde, J., Verlinden, P., Cools, R., “Homotopies exploiting Newton polytopes for solving sparse polynomial systems”, *SIAM Journal on Numerical Analysis*, vol. 31, no. 3, 1994, str. 915–930.
- [5] Van Hentenryck, P., McAllester, D., Kapur, D., “Solving Polynomial Systems Using a Branch and Prune Approach”, *SIAM Journal on Numerical Analysis*, vol. 34, no. 2, 1997, str. 797–827.
- [6] Zarowski, C. J., *An introduction to numerical analysis for electrical and computer engineers*. John Wiley & Sons, 2004.
- [7] Morgan, A., *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*, ser. *Classics in Applied Mathematics*. Philadelphia: SIAM, 2009.
- [8] Dickenstein, A., Emiris, I. Z., (ur.), *Solving polynomial equations: foundations, algorithms, and applications*, ser. *Algorithms and computation in mathematics*. Berlin; New York: Springer, 2005, vol. 14.
- [9] Sommese, A. J., Wampler II, C. W., *The Numerical Solution of Systems of Polynomials: Arising in Engineering And Science*. World Scientific, 2005.
- [10] Manocha, D., “Solving Systems of Polynomial Equations”, *IEEE Computer Graphics and Applications*, vol. 14, 1994, str. 46–55.

- [11] Dennis Jr, J. E., Schnabel, R. B., Numerical methods for unconstrained optimization and nonlinear equations, ser. Classics in Applied Mathematics. SIAM, 1996, vol. 16.
- [12] Deuffhard, P., Newton methods for nonlinear problems: affine invariance and adaptive algorithms, ser. Springer Series in Computational Mathematics. Springer Science & Business Media, 2011, vol. 35.
- [13] Hirsch, M. J., Meneses, C., Pardalos, P. M., Resende, M. G., “Global optimization by continuous grasp”, Optimization Letters, vol. 1, no. 2, 2007, str. 201–212.
- [14] Grosan, C., Abraham, A., Snasel, V., “Solving polynomial systems using a modified line search approach”, Int. J. Innov. Comput. I, vol. 8, no. 1, 2012.
- [15] Allgower, E. L., Georg, K., “Numerical path following”, Handbook of numerical analysis, vol. 5, 1997, str. 3–207.
- [16] Li, T.-Y., “Numerical solution of multivariate polynomial systems by homotopy continuation methods”, Acta Numerica, vol. 6, 1997, str. 399–436.
- [17] Kearfott, R. B., “Empirical Evaluation of Innovations in Interval Branch and Bound Algorithms for Nonlinear Systems”, SIAM J. Sci. Comput., vol. 18, no. 2, 1997, str. 574–594.
- [18] Kearfott, R. B., “An interval branch and bound algorithm for bound constrained optimization problems”, Journal of Global Optimization, vol. 2, no. 3, 1992, str. 259–280.
- [19] Kearfott, R. B., Rigorous global search: continuous problems, ser. Nonconvex Optimization and Its Applications. Dordrecht: Kluwer Academic Publishers, 1996, vol. 13.
- [20] Wilkinson, J. H., “The evaluation of the zeros of ill-conditioned polynomials. Part II”, Numerische Mathematik, vol. 1, no. 1, 1959, str. 167–180.
- [21] Möller, H., “Gröbner bases and numerical analysis”, Grobner bases and applications (Linz, 1998), vol. 251, 1998, str. 159–178.
- [22] Nocedal, J., Wright, S., Numerical Optimization, 2nd ed. New York: Springer Science & Business Media, 2006.
- [23] Boyer, C., A History of Mathematics. Wiley, 1991.
- [24] McNamee, J. M., Numerical Methods for Roots of Polynomials – Part I. Elsevier, 2007.
- [25] McNamee, J. M., Pan, V., Numerical Methods for Roots of Polynomials – Part II. Elsevier, 2013.

- [26] Jenkins, M. A., “Algorithm 493: Zeros of a real polynomial [c2]”, *ACM Transactions on Mathematical Software (TOMS)*, vol. 1, no. 2, 1975, str. 178–189.
- [27] Werner, W., “On the simultaneous determination of polynomial roots”, in *Iterative solution of nonlinear systems of equations*. Springer, 1982, str. 188–202.
- [28] Bini, D. A., Daddi, F., Gemignani, L., “On the shifted QR iteration applied to companion matrices.”, *ETNA. Electronic Transactions on Numerical Analysis [electronic only]*, vol. 18, 2004, str. 137–152.
- [29] Bulirsch, R., Stoer, J., *Introduction to numerical analysis*. Springer, 2002.
- [30] Hlupić, N., Beroš, I., Basch, D., “A derivative-free algorithm for solving quasi-linear systems”, in *Proceedings of the ITI 2013 35th International Conference on Information Technology Interfaces*. IEEE, 2013, str. 297–302.
- [31] Hlupić, N., Beroš, I., Basch, D., “A derivative-free algorithm for finding least squares solutions of quasi-linear and linear systems”, *Journal of computing and information technology*, vol. 21, no. 2, 2013, str. 125–135.
- [32] Golub, G. H., Van Loan, C. F., *Matrix computations*, 4th ed. JHU Press, 2012.
- [33] Beroš, I., Hlupić, N., Brčić, M., “Computational aspects of efficient estimation of harmonically related sine-waves”, *Automatika*, vol. 56, no. 3, 2015, str. 359–366.
- [34] Kelley, C. T., *Solving nonlinear equations with Newton’s method*, ser. *Fundamentals of Algorithms*. SIAM, 2003, vol. 1.
- [35] Moré, J. J., Garbow, B. S., Hillstom, K. E., “Testing Unconstrained Optimization Software”, *ACM Trans. Math. Softw.*, vol. 7, no. 1, 1981, str. 17–41.
- [36] Ortega, J. M., Rheinboldt, W. C., *Iterative solution of nonlinear equations in several variables*, ser. *Classics in Applied Mathematics*. SIAM, 2000, vol. 30.
- [37] Moré, J. J., Garbow, B. S., Hillstom, K. E., “User guide for MINPACK-1. Argonne National Laboratory”, Argonne, IL, USA, 1980.
- [38] “Octave solvers”, dostupno na: <https://octave.org/doc/v5.2.0/Solvers.html> (14. 01. 2019.).
- [39] “Numerical mathematics-newtonlib”, dostupno na: <http://elib.zib.de/pub/elib/codelib/NewtonLib> (21. listopada 2019.).
- [40] Nowak, U., Weimann, L., “A family of newton codes for systems of highly nonlinear equations-algorithm, implementation”, *Konrad-Zuse-Zentrum für Informationstechnik, Tech. Rep.*, 1990.

- [41] Verschelde, J., “The database of polynomial systems”, dostupno na: <http://homepages.math.uic.edu/~jan/> (28. 11. 2018.).
- [42] Agresti, A., An introduction to categorical data analysis. John Wiley & Sons, 2018.

Prilog A. Popis polinomnih sustava

Problem assur44

Dimenzija problema:8

Varijable: $\mathbf{x} = (c_a, s_a, c_b, s_b, c_c, s_c, c_d, s_d)$

$$f_1(\mathbf{x}) = c_a^2 + s_a^2 - 1, \quad f_2(\mathbf{x}) = c_b^2 + s_b^2 - 1$$

$$f_3(\mathbf{x}) = c_c^2 + s_c^2 - 1, \quad f_4(\mathbf{x}) = c_d^2 + s_d^2 - 1$$

$$\begin{aligned} f_5(\mathbf{x}) = & 8 \cdot c_c \cdot c_b \cdot c_a + 8 \cdot c_c \cdot c_b \cdot s_a - 8 \cdot c_c \cdot s_b \cdot c_a + 8 \cdot c_c \cdot s_b \cdot s_a - 8 \cdot s_c \cdot c_b \cdot c_a \\ & + 8 \cdot s_c \cdot c_b \cdot s_a - 8 \cdot s_c \cdot s_b \cdot c_a - 8 \cdot s_c \cdot s_b \cdot s_a + 24 \cdot c_c \cdot c_b - 24 \cdot c_c \cdot s_b - 24 \cdot s_c \cdot c_b \\ & - 24 \cdot s_c \cdot s_b - 16 \cdot c_b \cdot c_a + 12 \cdot c_b \cdot s_a - 12 \cdot s_b \cdot c_a - 16 \cdot s_b \cdot s_a - 4 \cdot c_c + 28 \cdot s_c \\ & - 48 \cdot c_b - 36 \cdot s_b + 24 \cdot c_a + 48 \end{aligned}$$

$$\begin{aligned} f_6(\mathbf{x}) = & -8 \cdot c_d \cdot c_b \cdot c_a + 16 \cdot c_d \cdot c_b \cdot s_a - 16 \cdot c_d \cdot s_b \cdot c_a - 8 \cdot c_d \cdot s_b \cdot s_a + 16 \cdot s_d \cdot c_b \cdot c_a \\ & + 8 \cdot s_d \cdot c_b \cdot s_a - 8 \cdot s_d \cdot s_b \cdot c_a + 16 \cdot s_d \cdot s_b \cdot s_a - 32 \cdot c_d \cdot c_b + 86 \cdot c_d \cdot s_b - 86 \cdot s_d \cdot c_b \\ & - 32 \cdot s_d \cdot s_b + 8 \cdot c_b \cdot c_a + 12 \cdot c_b \cdot s_a - 12 \cdot s_b \cdot c_a + 8 \cdot s_b \cdot s_a + 16 \cdot c_d + 28 \cdot s_d \\ & - 73 \cdot c_b + 12 \cdot s_b - 28 \cdot c_a - 30 \cdot s_a + 112 \end{aligned}$$

$$\begin{aligned} f_7(\mathbf{x}) = & -980 \cdot c_d \cdot c_b \cdot c_a - 1960 \cdot c_d \cdot c_b \cdot s_a + 1960 \cdot c_d \cdot s_b \cdot c_a - 980 \cdot c_d \cdot s_b \cdot s_a \\ & - 1960 \cdot s_d \cdot c_b \cdot c_a + 980 \cdot s_d \cdot c_b \cdot s_a - 980 \cdot s_d \cdot s_b \cdot c_a - 1960 \cdot s_d \cdot s_b \cdot s_a \\ & + 280 \cdot c_c \cdot c_b \cdot c_a - 1400 \cdot c_c \cdot c_b \cdot s_a + 1400 \cdot c_c \cdot s_b \cdot c_a + 280 \cdot c_c \cdot s_b \cdot s_a \\ & + 1400 \cdot s_c \cdot c_b \cdot c_a + 280 \cdot s_c \cdot c_b \cdot s_a - 280 \cdot s_c \cdot s_b \cdot c_a \\ & + 1400 \cdot s_c \cdot s_b \cdot s_a + 6300 \cdot c_d \cdot c_c - 4900 \cdot c_d \cdot s_c + 6860 \cdot c_d \cdot c_b + 8330 \cdot c_d \cdot s_b \\ & - 4900 \cdot s_d \cdot c_c - 6300 \cdot s_d \cdot s_c - 8330 \cdot s_d \cdot c_b + 6860 \cdot s_d \cdot s_b + 6860 \cdot c_c \cdot c_b \\ & + 1540 \cdot c_c \cdot s_b + 1540 \cdot s_c \cdot c_b - 6860 \cdot s_c \cdot s_b + 56 \cdot c_b \cdot c_a + 2268 \cdot c_b \cdot s_a \\ & - 2268 \cdot s_b \cdot c_a + 56 \cdot s_b \cdot s_a - 11480 \cdot c_d + 5390 \cdot s_d - 8060 \cdot c_c + 1820 \cdot s_c \\ & - 10094 \cdot c_b - 4788 \cdot s_b + 784 \cdot c_a - 1764 \cdot s_a + 19798 \end{aligned}$$

$$f_8(\mathbf{x}) = 280 \cdot c_d \cdot c_c - 280 \cdot s_d \cdot s_c - 336 \cdot c_d - 240 \cdot c_c + 391$$

Problem bifurcation

Dimenzija problema:3

Varijable: $\mathbf{x} = (x, y, z)$

$$f_1(\mathbf{x}) = x^2 + y^2 - 0,265625$$

$$f_2(\mathbf{x}) = 5x^8 - 6x^4y + y^4 + 2z$$

$$f_3(\mathbf{x}) = 2x^2y^2 - 2x^6 + 2z$$

Problem bifurcation2

Dimenzija problema:3

Varijable: $\mathbf{x} = (x, y, z)$

$$f_1(\mathbf{x}) = x + y^2 - 0,265625$$

$$f_2(\mathbf{x}) = 5x^4 - 6x^2y + y^4 + 2z$$

$$f_3(\mathbf{x}) = 2xy^2 - 2x^3 + 2z$$

Problem boon

Dimenzija problema:6

Varijable: $\mathbf{x} = (s_1, g_1, s_2, g_2, C_1, C_2)$

$$f_1(\mathbf{x}) = s_1^2 + g_1^2 - 1$$

$$f_2(\mathbf{x}) = s_2^2 + g_2^2 - 1$$

$$f_3(\mathbf{x}) = C_1 \cdot g_1^3 + C_2 \cdot g_2^3 - 1,2$$

$$f_4(\mathbf{x}) = C_1 \cdot s_1^3 + C_2 \cdot s_2^3 - 1,2$$

$$f_5(\mathbf{x}) = C_1 \cdot g_1^2 \cdot s_1 + C_2 \cdot g_2^2 \cdot s_2 - 0,7$$

$$f_6(\mathbf{x}) = C_1 \cdot g_1 \cdot s_1^2 + C_2 \cdot g_2 \cdot s_2^2 - 0,7$$

Problem brown

Dimenzija problema:5

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$

$$f_1(\mathbf{x}) = 2x_1 + x_2 + x_3 + x_4 + x_5 - 6$$

$$f_2(\mathbf{x}) = x_1 + 2x_2 + x_3 + x_4 + x_5 - 6$$

$$f_3(\mathbf{x}) = x_1 + x_2 + 2x_3 + x_4 + x_5 - 6$$

$$f_4(\mathbf{x}) = x_1 + x_2 + x_3 + 2x_4 + x_5 - 6$$

$$f_5(\mathbf{x}) = x_1 x_2 x_3 x_4 x_5 - 1$$

Problem butcher7

Dimenzija problema:7

Variable: $\mathbf{x} = (x, y, z, t, u, v, w)$

$$f_1(\mathbf{x}) = z \cdot u + y \cdot v + t \cdot w - w^2 - \frac{1}{2} \cdot w - \frac{1}{2}$$

$$f_2(\mathbf{x}) = z \cdot u^2 + y \cdot v^2 - t \cdot w^2 + w^3 + w^2 - \frac{1}{3} \cdot t + \frac{4}{3} \cdot w$$

$$f_3(\mathbf{x}) = x \cdot z \cdot v - t \cdot w^2 + w^3 - \frac{1}{2} \cdot t \cdot w + w^2 - \frac{1}{6} \cdot t + \frac{2}{3} \cdot w$$

$$f_4(\mathbf{x}) = z \cdot u^3 + y \cdot v^3 + t \cdot w^3 - w^4 - \frac{3}{2} \cdot w^3 + t \cdot w - \frac{5}{2} \cdot w^2 - \frac{1}{4} \cdot w - \frac{1}{4}$$

$$f_5(\mathbf{x}) = x \cdot z \cdot u \cdot v + t \cdot w^3 - w^4 + \frac{1}{2} \cdot t \cdot w^2 - \frac{3}{2} \cdot w^3 + \frac{1}{2} \cdot t \cdot w - \frac{7}{4} \cdot w^2 - \frac{3}{8} \cdot w - \frac{1}{8}$$

$$f_6(\mathbf{x}) = x \cdot z \cdot v^2 + t \cdot w^3 - w^4 + t \cdot w^2 - \frac{3}{2} \cdot w^3 + \frac{2}{3} \cdot t \cdot w - \frac{7}{6} \cdot w^2 - \frac{1}{12} \cdot w - \frac{1}{12}$$

$$f_7(\mathbf{x}) = -t \cdot w^3 + w^4 - t \cdot w^2 + \frac{3}{2} \cdot w^3 - \frac{1}{3} \cdot t \cdot w + \frac{13}{12} \cdot w^2 + \frac{7}{24} \cdot w + \frac{1}{24}$$

Problem butcher8

Dimenzija problema:8

Variable: $\mathbf{x} = (b_1, b_2, b_3, a, b, c_2, c_3, a_{32})$

$$f_1(\mathbf{x}) = -a - b + b_1 + b_2 + b_3$$

$$f_2(\mathbf{x}) = -\frac{1}{2} - \frac{1}{2} \cdot b + a \cdot b - b^2 + b_2 \cdot c_2 + b_3 \cdot c_3$$

$$f_3(\mathbf{x}) = -\frac{1}{3} \cdot a + \frac{4}{3} \cdot b + b^2 - a \cdot b^2 + b^3 + b_2 \cdot c_2^2 + b_3 \cdot c_3^2$$

$$f_4(\mathbf{x}) = -\frac{1}{6} \cdot a + \frac{2}{3} \cdot b - \frac{1}{2} \cdot a \cdot b + b^2 - a \cdot b^2 + b^3 + a_{32} \cdot b_3 \cdot c_2$$

$$f_5(\mathbf{x}) = -\frac{1}{4} - \frac{1}{4} \cdot b + a \cdot b - \frac{5}{2} \cdot b^2 - \frac{3}{2} \cdot b^3 + a \cdot b^3 - b^4 + b_2 \cdot c_2^3 + b_3 \cdot c_3^3$$

$$f_6(\mathbf{x}) = -\frac{1}{8} - \frac{3}{8} \cdot b + \frac{1}{2} \cdot a \cdot b - \frac{7}{4} \cdot b^2 + \frac{1}{2} \cdot a \cdot b^2 - \frac{3}{2} \cdot b^3 + a \cdot b^3 - b^4 + a_{32} \cdot b_3 \cdot c_2 \cdot c_3$$

$$f_7(\mathbf{x}) = -\frac{1}{12} - \frac{1}{12} \cdot b + \frac{2}{3} \cdot a \cdot b - \frac{7}{6} \cdot b^2 + a \cdot b^2 - \frac{3}{2} \cdot b^3 + a \cdot b^3 - b^4 + a_{32} \cdot b_3 \cdot c_2^2$$

$$f_8(\mathbf{x}) = \frac{1}{24} + \frac{7}{24} \cdot b - \frac{1}{3} \cdot a \cdot b + \frac{13}{12} \cdot b^2 - a \cdot b^2 + \frac{3}{2} \cdot b^3 - a \cdot b^3 + b^4$$

Problem camera1s

Dimenzija problema:6

Varijable: $\mathbf{x} = (d_1, d_2, d_3, q_1, q_2, q_3)$

$$f_1(\mathbf{x}) = -d_1q_1 - d_2q_2 - d_3q_3 + 1$$

$$f_2(\mathbf{x}) = -3,6d_1q_1 + 4,1d_1q_2 + 2,0d_1q_3 + 0,1d_1 + 4,1d_2q_1 + 1,8d_2q_2 + 3,7d_2q_3 - 0,2d_2 \\ + 2,0d_3q_1 + 3,7d_3q_2 - 4,0d_3q_3 + 0,3d_3 + 0,1q_1 - 0,2q_2 + 0,3q_3 + 5,8$$

$$f_3(\mathbf{x}) = -2,140796d_1q_1 - 3,998792d_1q_2 + 3,715992d_1q_3 - 0,2828d_1 - 3,998792d_2q_1 \\ - 1,575196d_2q_2 - 3,998792d_2q_3 + 3,715992d_3q_1 - 3,998792d_3q_2 \\ - 2,140796d_3q_3 + 0,2828d_3 - 0,2828q_1 + 0,2828q_3 + 5,856788$$

$$f_4(\mathbf{x}) = 0,3464d_1q_1 + 0,1732d_1q_2 - 5,999648d_1q_3 - 0,1732d_1 + 0,1732d_2q_1 - 5,999648d_2q_2 \\ - 0,1732d_2q_3 + 0,3464d_2 - 5,999648d_3q_1 - 0,1732d_3q_2 - 0,3464d_3q_3 \\ - 0,1732d_3 - 0,1732q_1 + 0,3464q_2 - 0,1732q_3 + 5,999648$$

$$f_5(\mathbf{x}) = -5701,3d_1q_1 - 2,9d_1q_2 + 3796,7d_1q_3 - 1902,7d_1 - 2,9d_2q_1 - 5698,7d_2q_2 \\ + 1897,3d_2q_3 + 3803,3d_2 + 3796,7d_3q_1 + 1897,3d_3q_2 + 5703,1d_3q_3 + 0,7d_3 \\ - 1902,7q_1 + 3803,3q_2 + 0,7q_3 + 5696,9$$

$$f_6(\mathbf{x}) = -6,8d_1q_1 - 3,2d_1q_2 + 1,3d_1q_3 + 5,1d_1 - 3,2d_2q_1 - 4,8d_2q_2 - 0,7d_2q_3 - 7,1d_2 \\ + 1,3d_3q_1 - 0,7d_3q_2 + 9,0d_3q_3 - d_3 + 5,1q_1 - 7,1q_2 - q_3 + 2,6$$

Problem caprasse

Dimenzija problema:4

Varijable: $\mathbf{x} = (x, y, z, t)$

$$f_1(\mathbf{x}) = y^2 \cdot z + 2 \cdot x \cdot y \cdot t - 2 \cdot x - z$$

$$f_2(\mathbf{x}) = -x^3 \cdot z + 4 \cdot x \cdot y^2 \cdot z + 4 \cdot x^2 \cdot y \cdot t + 2 \cdot y^3 \cdot t + 4 \cdot x^2 - 10 \cdot y^2 + 4 \cdot x \cdot z - 10 \cdot y \cdot t + 2$$

$$f_3(\mathbf{x}) = 2 \cdot y \cdot z \cdot t + x \cdot t^2 - x - 2 \cdot z$$

$$f_4(\mathbf{x}) = -x \cdot z^3 + 4 \cdot y \cdot z^2 \cdot t + 4 \cdot x \cdot z \cdot t^2 + 2 \cdot y \cdot t^3 + 4 \cdot x \cdot z + 4 \cdot z^2 - 10 \cdot y \cdot t - 10 \cdot t^2 + 2$$

Problem cassou

Dimenzija problema:4

Varijable: $\mathbf{x} = (b, c, d, e)$

$$f_1(\mathbf{x}) = 15b^4cd^2 + 6b^4c^3 + 21b^4c^2d - 144b^2c - 8b^2c^2e - 28b^2cde - 648b^2d \\ + 36b^2d^2e + 9b^4d^3 - 120,0$$

$$\begin{aligned}
 f_2(\mathbf{x}) &= 30c^3b^4d - 32de^2c - 720db^2c - 24c^3b^2e - 432c^2b^2 + 576ec - 576de + 16cb^2d^2e \\
 &+ 16d^2e^2 + 16e^2c^2 + 9c^4b^4 + 5184 + 39d^2b^4c^2 + 18d^3b^4c - 432d^2b^2 \\
 &+ 24d^3b^2e - 16c^2b^2de - 240c
 \end{aligned}$$

$$\begin{aligned}
 f_3(\mathbf{x}) &= 216db^2c - 162d^2b^2 - 81c^2b^2 + 5184 + 1008ec - 1008de + 15c^2b^2de - 15c^3b^2e \\
 &- 80de^2c + 40d^2e^2 + 40e^2c^2
 \end{aligned}$$

$$f_4(\mathbf{x}) = 261 + 4db^2c - 3d^2b^2 - 4c^2b^2 + 22ec - 22de$$

Problem chandra4

Dimenzija problema:4

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4)$

$$f_1(\mathbf{x}) = -8 + 7,48766x_1 - 0,25617x_1^2 - 0,17078x_1x_2 - 0,128085x_1x_3$$

$$f_2(\mathbf{x}) = -8 + 7,48766x_2 - 0,34156x_1x_2 - 0,25617x_2^2 - 0,204936x_2x_3$$

$$f_3(\mathbf{x}) = -8 + 7,48766x_3 - 0,384255x_1x_3 - 0,307404x_2x_3 - 0,25617x_3^2$$

$$f_4(\mathbf{x}) = -8 + 7,48766x_4 - 0,409872x_1x_4 - 0,34156x_2x_4 - 0,2927657142857143x_3x_4$$

Problem chemequ

Dimenzija problema:5

Varijable: $\mathbf{x} = (y_1, y_2, y_3, y_4, y_5)$

$$f_1(\mathbf{x}) = y_1 \cdot y_2 + y_1 - 3 \cdot y_5$$

$$\begin{aligned}
 f_2(\mathbf{x}) &= 2 \cdot y_1 \cdot y_2 + y_1 + 1,9230E-06 \cdot y_2^2 + y_2 \cdot y_3^2 + 5,4518E-04 \cdot y_2 \cdot y_3 \\
 &+ 3,4074E-05 \cdot y_2 \cdot y_4 + 4,4975E-07 \cdot y_2 - 10 \cdot y_5
 \end{aligned}$$

$$f_3(\mathbf{x}) = 2 \cdot y_2 \cdot y_3^2 + 5,4518E-04 \cdot y_2 \cdot y_3 + 3,8600E-01 \cdot y_3^2 + 4,1062E-04 \cdot y_3 - 8 \cdot y_5$$

$$f_4(\mathbf{x}) = 3,4074E-05 \cdot y_2 \cdot y_4 + 2 \cdot y_4^2 - 40 \cdot y_5$$

$$\begin{aligned}
 f_5(\mathbf{x}) &= y_1 \cdot y_2 + y_1 + 9,6150E-07 \cdot y_2^2 + y_2 \cdot y_3^2 + 5,4518E-04 \cdot y_2 \cdot y_3 + 3,4074E-05 \cdot y_2 \cdot y_4 \\
 &+ 4,4975E-07 \cdot y_2 + 1,930E-01 \cdot y_3^2 + 4,1062E-04 \cdot y_3 + y_4^2 - 1
 \end{aligned}$$

Problem chemequs

Dimenzija problema:5

Varijable: $\mathbf{x} = (y_1, y_2, y_3, y_4, y_5)$

$$f_1(\mathbf{x}) = 4,00649105280831y_1y_2 + 0,526610813156747y_1 - 0,473964757724582y_5$$

$$\begin{aligned}
 f_2(\mathbf{x}) &= 7,43133120490877y_2y_3^2 + 5,31324863453933y_1y_2 + 4,31363490603044y_2^2 \\
 &+ 0,806989849429520y_2y_3 + 0,149987275886495y_2y_4 + 0,349185128215561y_1 \\
 &+ 0,132605295558012y_2 - 1,04758859585902y_5
 \end{aligned}$$

$$\begin{aligned}
 f_3(\mathbf{x}) &= 18,8743868807074 y_2 y_3^2 + 1,02481225819107 y_2 y_3 + 0,478801780852764 y_3^2 \\
 &\quad - 1,06428266510309 y_5 + 0,101454213888243 y_3 \\
 f_4(\mathbf{x}) &= 6,77344896993456E-02 y_2 y_4 + 7,80160463417002 y_4^2 - 1,89237063040101 y_5 \\
 f_5(\mathbf{x}) &= 3,89976792151467 y_2 y_3^2 + 1,39412684031101 y_1 y_2 + 1,13184129248280 y_2^2 \\
 &\quad + 0,423487130504208 y_2 y_3 + 0,0787093928408356 y_2 y_4 + 0,098928555270968 y_3^2 \\
 &\quad + 4,53284262319986 y_4^2 + 0,183243206921733 y_1 + 0,069587784958167 y_2 \\
 &\quad + 0,041924316940676 y_3 - 20337,0256289491
 \end{aligned}$$

Problem chemical

Dimenzija problema:3

Varijable: $\mathbf{x} = (z_1, z_2, z_3)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= 14 z_1^2 + 6 z_1 z_2 + 5 z_1 - 72 z_2^2 - 18 z_2 - 850 z_3 + 0,000000002 \\
 f_2(\mathbf{x}) &= 0,5 z_1 z_2^2 + 0,01 z_1 z_2 + 0,13 z_2^2 + 0,04 z_2 - 40000 \\
 f_3(\mathbf{x}) &= 0,03 z_1 z_3 + 0,04 z_3 - 850
 \end{aligned}$$

Problem chemkin

Dimenzija problema:10

Varijable: $\mathbf{x} = (y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4, y_5, z_5)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= 9 \cdot y_2^2 - 5,656854249492381 \cdot y_2 + z_2 \\
 f_2(\mathbf{x}) &= x_3^2 + y_3^2 + z_3^2 - 1 \\
 f_3(\mathbf{x}) &= x_4^2 + y_4^2 + z_4^2 - 1 \\
 f_4(\mathbf{x}) &= y_5^2 + z_5^2 - 8/9 \\
 f_5(\mathbf{x}) &= x_3 - 2,828427124746190 \cdot y_2 \cdot x_3 + y_2 \cdot y_3 + z_2 \cdot z_3 - 1/3 \\
 f_6(\mathbf{x}) &= x_3 \cdot x_4 + y_3 \cdot y_4 + z_3 \cdot z_4 - 1/3 \\
 f_7(\mathbf{x}) &= 1/3 \cdot x_4 + y_4 \cdot y_5 + z_4 \cdot z_5 - 1/3 \\
 f_8(\mathbf{x}) &= 8/3 - 2,828427124746190 \cdot y_2 + x_3 + x_4 \\
 f_9(\mathbf{x}) &= y_2 + y_3 + y_4 + y_5 + 8/9 \\
 f_{10}(\mathbf{x}) &= z_2 + z_3 + z_4 + z_5
 \end{aligned}$$

Problem cohn2

Dimenzija problema:4

Varijable: $\mathbf{x} = (x, y, z, t)$

$$f_1(\mathbf{x}) = x^3 y^2 + 4x^2 y^2 z - x^2 y z^2 + 288x^2 y^2 + 207x^2 y z + 1152xy^2 z + 156xy z^2 + xz^3 - 3456x^2 y + 20736xy^2 + 19008xyz + 82944y^2 z + 432xz^2 - 497664xy + 62208xz + 2985984x$$

$$f_2(\mathbf{x}) = y^3 t^3 + 4y^3 t^2 - y^2 z t^2 + 4y^2 t^3 - 48y^2 t^2 - 5y z t^2 + 108y z t + z^2 t + 144z t - 1728z$$

$$f_3(\mathbf{x}) = -x^2 z^2 t + 4xz^2 t^2 + z^3 t^2 + x^3 z + 156x^2 z t + 207xz^2 t + 1152xz t^2 + 288z^2 t^2 + 432x^2 z + 19008x z t - 3456z^2 t + 82944x t^2 + 20736z t^2 + 62208xz - 497664z t + 2985984z$$

$$f_4(\mathbf{x}) = y^3 t^3 - xy^2 t^2 + 4y^3 t^2 + 4y^2 t^3 - 5xy^2 t - 48y^2 t^2 + x^2 y + 108xy t + 144xy - 1728x$$

Problem cohn3

Dimenzija problema:4

Varijable: $\mathbf{x} = (x, y, z, t)$

$$f_1(\mathbf{x}) = -x^3 y^2 + 2x^2 y^2 z - x^2 y z^2 - 144x^2 y^2 - 207x^2 y z + 288xy^2 z + 78xyz^2 + xz^3 - 3456x^2 y - 5184xy^2 - 9504xyz - 432xz^2 - 248832xy + 62208xz - 2985984x$$

$$f_2(\mathbf{x}) = -x^3 z t^2 + x^2 z^2 t^2 - 6x^3 z t + 4x^2 z^2 t + 32x^3 t^2 - 72x^2 z t^2 - 87xz^2 t^2 - z^3 t^2 - 8x^3 z - 432x^2 z t - 414xz^2 t + 2592xz t^2 + 864z^2 t^2 - 1728x^2 z - 20736x z t + 3456z^2 t - 186624z t^2 - 124416xz - 1492992z t - 2985984z$$

$$f_3(\mathbf{x}) = x^2 y t^3 - 2xy^2 t^3 + y^3 t^3 + 8x^2 y t^2 - 12xy^2 t^2 + 4y^3 t^2 - 24xy t^3 + 24y^2 t^3 + 20x^2 y t - 20xy^2 t - 160xy t^2 + 96y^2 t^2 + 128x t^3 + 16x^2 y + 96xy t + 2304x t^2 + 1152xy + 13824x t + 27648x$$

$$f_4(\mathbf{x}) = y^3 t^3 - y^2 z t^3 + 4y^3 t^2 - 2y^2 z t^2 + 72y^2 t^3 + 71y z t^3 + 288y^2 t^2 + 360y z t^2 + 6z^2 t^2 + 1728y t^3 - 464z t^3 + 432y z t + 8z^2 t + 6912y t^2 - 4320z t^2 + 13824t^3 + z^2 - 13824z t + 55296t^2 - 13824z$$

Problem combo3000

Dimenzija problema:10

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$

$$f_1(\mathbf{x}) = x_2 + 2 \cdot x_6 + x_9 + 2 \cdot x_{10} - 1,0E-5$$

$$f_2(\mathbf{x}) = x_3 + x_8 - 3,0E-5$$

$$f_3(\mathbf{x}) = x_1 + x_3 + 2 \cdot x_5 + 2 \cdot x_8 + x_9 + x_{10} - 5,0E-5$$

$$f_4(\mathbf{x}) = x_4 + 2 \cdot x_7 - 1,0E-5$$

$$f_5(\mathbf{x}) = 0,5140437E-7 \cdot x_5 - x_1^2$$

$$\begin{aligned}
 f_6(\mathbf{x}) &= 0,1006932\text{E-}6 \cdot x_6 - x_2^2 \\
 f_7(\mathbf{x}) &= 0,7816278\text{E-}15 \cdot x_7 - x_4^2 \\
 f_8(\mathbf{x}) &= 0,1496236\text{E-}6 \cdot x_8 - x_1 \cdot x_3 \\
 f_9(\mathbf{x}) &= 0,6194411\text{E-}7 \cdot x_9 - x_1 \cdot x_2 \\
 f_{10}(\mathbf{x}) &= 0,2089296\text{E-}14 \cdot x_{10} - x_1 \cdot x_2^2
 \end{aligned}$$

Problem combo3000s

Dimenzija problema:10

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$

$$\begin{aligned}
 f_1(\mathbf{x}) &= 1,01815483301669\text{E-}01 x_2 + 9,89610100506422\text{E-}01 x_6 + 1,34637048100730 x_9 \\
 &+ 3,46970317210432 x_{10} - 2,12454115933396 \\
 f_2(\mathbf{x}) &= 5,76739795357135\text{E-}01 x_3 + 7,89949754301577\text{E-}01 x_8 - 2,19492968782850 \\
 f_3(\mathbf{x}) &= 7,19621954936988\text{E-}01 x_9 + 9,27261335190857\text{E-}01 x_{10} + 8,95128807036246\text{E-}01 x_3 \\
 &+ 2,45208250541714 x_8 + 8,73159766974099\text{E-}02 x_1 + 1,37722259176202 x_5 \\
 &- 5,67773314994310 \\
 f_4(\mathbf{x}) &= 2,50030218520604\text{E-}03 x_4 + 1,99987913687438\text{E+}01 x_7 - 1,99987913687440\text{E+}01 \\
 f_5(\mathbf{x}) &= -1,37722259176203 x_1^2 + 7,26099038733160\text{E-}01 x_5 \\
 f_6(\mathbf{x}) &= -9,89610100506425\text{E-}01 x_2^2 + 1,01049898287039 x_6 \\
 f_7(\mathbf{x}) &= -1,99987913687438\text{E+}01 x_4^2 + 5,00030217607503\text{E-}02 x_7 \\
 f_8(\mathbf{x}) &= -1,93702197268146 x_3 x_1 + 5,16256404988361\text{E-}01 x_8 \\
 f_9(\mathbf{x}) &= -9,68877757611935\text{E-}01 x_2 x_1 + 1,03212194948595 x_9 \\
 f_{10}(\mathbf{x}) &= -3,21732159608141 x_2^2 x_1 + 3,10817545009477\text{E-}01 x_{10}
 \end{aligned}$$

Problem cpdm5

Dimenzija problema:5

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= 4x_1^3 + 3x_1^2 x_2 + 3x_1^2 x_3 + 3x_1^2 x_4 + 3x_1^2 x_5 + 3x_1 x_2^2 + 3x_1 x_3^2 + 3x_1 x_4^2 + 3x_1 x_5^2 \\
 &+ x_2^3 + x_3^3 + x_4^3 + x_5^3 + 2x_1^2 + 3x_1 x_2 + 3x_1 x_3 + 3x_1 x_4 + 3x_1 x_5 - x_2^2 - x_3^2 - x_4^2 - x_5^2 - 6x_1 \\
 f_2(\mathbf{x}) &= x_1^3 + 3x_1^2 x_2 + 3x_1 x_2^2 + 4x_2^3 + 3x_2^2 x_3 + 3x_2^2 x_4 + 3x_2^2 x_5 + 3x_2 x_3^2 + 3x_2 x_4^2 + 3x_2 x_5^2 \\
 &+ x_3^3 + x_4^3 + x_5^3 - x_1^2 + 3x_1 x_2 + 2x_2^2 + 3x_2 x_3 + 3x_2 x_4 + 3x_2 x_5 - x_3^2 - x_4^2 - x_5^2 - 6x_2 \\
 f_3(\mathbf{x}) &= x_1^3 + 3x_1^2 x_3 + 3x_1 x_3^2 + x_2^3 + 3x_2^2 x_3 + 3x_2 x_3^2 + 4x_3^3 + 3x_3^2 x_4 + 3x_3^2 x_5 + 3x_3 x_4^2 + 3x_3 x_5^2 \\
 &+ x_4^3 + x_5^3 - x_1^2 + 3x_1 x_3 - x_2^2 + 3x_2 x_3 + 2x_3^2 + 3x_3 x_4 + 3x_3 x_5 - x_4^2 - x_5^2 - 6x_3
 \end{aligned}$$

$$\begin{aligned}
 f_4(\mathbf{x}) &= x_1^3 + 3x_1^2x_4 + 3x_1x_4^2 + x_2^3 + 3x_2^2x_4 + 3x_2x_4^2 + x_3^3 + 3x_3^2x_4 + 3x_3x_4^2 + 4x_4^3 + 3x_4^2x_5 + 3x_4x_5^2 \\
 &+ x_5^3 - x_1^2 + 3x_1x_4 - x_2^2 + 3x_2x_4 - x_3^2 + 3x_3x_4 + 2x_4^2 + 3x_4x_5 - x_5^2 - 6x_4 \\
 f_5(\mathbf{x}) &= x_1^3 + 3x_1^2x_5 + 3x_1x_5^2 + x_2^3 + 3x_2^2x_5 + 3x_2x_5^2 + x_3^3 + 3x_3^2x_5 + 3x_3x_5^2 + x_4^3 + 3x_4^2x_5 + 3x_4x_5^2 \\
 &+ 4x_5^3 - x_1^2 + 3x_1x_5 - x_2^2 + 3x_2x_5 - x_3^2 + 3x_3x_5 - x_4^2 + 3x_4x_5 + 2x_5^2 - 6x_5
 \end{aligned}$$

Problem cyclic5

Dimenzija problema:5

Varijable: $\mathbf{x} = (abcdx)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= a + b + c + d + x \\
 f_2(\mathbf{x}) &= a \cdot b + b \cdot c + c \cdot d + d \cdot x + x \cdot a \\
 f_3(\mathbf{x}) &= a \cdot b \cdot c + b \cdot c \cdot d + c \cdot d \cdot x + d \cdot x \cdot a + x \cdot a \cdot b \\
 f_4(\mathbf{x}) &= a \cdot b \cdot c \cdot d + b \cdot c \cdot d \cdot x + c \cdot d \cdot x \cdot a + d \cdot x \cdot a \cdot b + x \cdot a \cdot b \cdot c \\
 f_5(\mathbf{x}) &= a \cdot b \cdot c \cdot d \cdot x - 1
 \end{aligned}$$

Problem cyclic6

Dimenzija problema:6

Varijable: $\mathbf{x} = (z_0, z_1, z_2, z_3, z_4, z_5)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= z_0 + z_1 + z_2 + z_3 + z_4 + z_5 \\
 f_2(\mathbf{x}) &= z_0z_1 + z_1z_2 + z_2z_3 + z_3z_4 + z_4z_5 + z_5z_0 \\
 f_3(\mathbf{x}) &= z_0z_1z_2 + z_1z_2z_3 + z_2z_3z_4 + z_3z_4z_5 + z_4z_5z_0 + z_5z_0z_1 \\
 f_4(\mathbf{x}) &= z_0z_1z_2z_3 + z_1z_2z_3z_4 + z_2z_3z_4z_5 + z_3z_4z_5z_0 + z_4z_5z_0z_1 + z_5z_0z_1z_2 \\
 f_5(\mathbf{x}) &= z_0z_1z_2z_3z_4 + z_1z_2z_3z_4z_5 + z_2z_3z_4z_5z_0 + z_3z_4z_5z_0z_1 + z_4z_5z_0z_1z_2 + z_5z_0z_1z_2z_3 \\
 f_6(\mathbf{x}) &= z_0z_1z_2z_3z_4z_5 - 1
 \end{aligned}$$

Problem cyclic7

Dimenzija problema:7

Varijable: $\mathbf{x} = (z_0, z_1, z_2, z_3, z_4, z_5, z_6)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= z_0 + z_1 + z_2 + z_3 + z_4 + z_5 + z_6 \\
 f_2(\mathbf{x}) &= z_0z_1 + z_1z_2 + z_2z_3 + z_3z_4 + z_4z_5 + z_5z_6 + z_6z_0 \\
 f_3(\mathbf{x}) &= z_0z_1z_2 + z_1z_2z_3 + z_2z_3z_4 + z_3z_4z_5 + z_4z_5z_6 + z_5z_6z_0 + z_6z_0z_1 \\
 f_4(\mathbf{x}) &= z_0z_1z_2z_3 + z_1z_2z_3z_4 + z_2z_3z_4z_5 + z_3z_4z_5z_6 + z_4z_5z_6z_0 + z_5z_6z_0z_1 + z_6z_0z_1z_2
 \end{aligned}$$

$$f_5(\mathbf{x}) = z_0 z_1 z_2 z_3 z_4 + z_1 z_2 z_3 z_4 z_5 + z_2 z_3 z_4 z_5 z_6 + z_3 z_4 z_5 z_6 z_0 + z_4 z_5 z_6 z_0 z_1 + z_5 z_6 z_0 z_1 z_2 + z_6 z_0 z_1 z_2 z_3$$

$$f_6(\mathbf{x}) = z_0 z_1 z_2 z_3 z_4 z_5 + z_1 z_2 z_3 z_4 z_5 z_6 + z_2 z_3 z_4 z_5 z_6 z_0 + z_3 z_4 z_5 z_6 z_0 z_1 + z_4 z_5 z_6 z_0 z_1 z_2 + z_5 z_6 z_0 z_1 z_2 z_3 + z_6 z_0 z_1 z_2 z_3 z_4$$

$$f_7(\mathbf{x}) = z_0 z_1 z_2 z_3 z_4 z_5 z_6 - 1$$

Problem cyclic8

Dimenzija problema:8

Varijable: $\mathbf{x} = (z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7)$

$$f_1(\mathbf{x}) = z_0 + z_1 + z_2 + z_3 + z_4 + z_5 + z_6 + z_7$$

$$f_2(\mathbf{x}) = z_0 z_1 + z_1 z_2 + z_2 z_3 + z_3 z_4 + z_4 z_5 + z_5 z_6 + z_6 z_7 + z_7 z_0$$

$$f_3(\mathbf{x}) = z_0 z_1 z_2 + z_1 z_2 z_3 + z_2 z_3 z_4 + z_3 z_4 z_5 + z_4 z_5 z_6 + z_5 z_6 z_7 + z_6 z_7 z_0 + z_7 z_0 z_1$$

$$f_4(\mathbf{x}) = z_0 z_1 z_2 z_3 + z_1 z_2 z_3 z_4 + z_2 z_3 z_4 z_5 + z_3 z_4 z_5 z_6 + z_4 z_5 z_6 z_7 + z_5 z_6 z_7 z_0 + z_6 z_7 z_0 z_1 + z_7 z_0 z_1 z_2$$

$$f_5(\mathbf{x}) = z_0 z_1 z_2 z_3 z_4 + z_1 z_2 z_3 z_4 z_5 + z_2 z_3 z_4 z_5 z_6 + z_3 z_4 z_5 z_6 z_7 + z_4 z_5 z_6 z_7 z_0 + z_5 z_6 z_7 z_0 z_1 + z_6 z_7 z_0 z_1 z_2 + z_7 z_0 z_1 z_2 z_3$$

$$f_6(\mathbf{x}) = z_0 z_1 z_2 z_3 z_4 z_5 + z_1 z_2 z_3 z_4 z_5 z_6 + z_2 z_3 z_4 z_5 z_6 z_7 + z_3 z_4 z_5 z_6 z_7 z_0 + z_4 z_5 z_6 z_7 z_0 z_1 + z_5 z_6 z_7 z_0 z_1 z_2 + z_6 z_7 z_0 z_1 z_2 z_3 + z_7 z_0 z_1 z_2 z_3 z_4$$

$$f_7(\mathbf{x}) = z_0 z_1 z_2 z_3 z_4 z_5 z_6 + z_1 z_2 z_3 z_4 z_5 z_6 z_7 + z_2 z_3 z_4 z_5 z_6 z_7 z_0 + z_3 z_4 z_5 z_6 z_7 z_0 z_1 + z_4 z_5 z_6 z_7 z_0 z_1 z_2 + z_5 z_6 z_7 z_0 z_1 z_2 z_3 + z_6 z_7 z_0 z_1 z_2 z_3 z_4 + z_7 z_0 z_1 z_2 z_3 z_4 z_5$$

$$f_8(\mathbf{x}) = z_0 z_1 z_2 z_3 z_4 z_5 z_6 z_7 - 1$$

Problem d1

Dimenzija problema:12

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12})$

$$f_1(\mathbf{x}) = x_1^2 + x_2^2 - 1$$

$$f_2(\mathbf{x}) = x_3^2 + x_4^2 - 1$$

$$f_3(\mathbf{x}) = x_5^2 + x_6^2 - 1$$

$$f_4(\mathbf{x}) = x_7^2 + x_8^2 - 1$$

$$f_5(\mathbf{x}) = x_9^2 + x_{10}^2 - 1$$

$$f_6(\mathbf{x}) = x_{11}^2 + x_{12}^2 - 1$$

$$f_7(\mathbf{x}) = 3x_3 + 2x_5 + x_7 - 3,9701$$

$$f_8(\mathbf{x}) = 3x_1 x_4 + 2x_1 x_6 + x_1 x_8 - 1,7172$$

$$f_9(\mathbf{x}) = 3x_2 x_4 + 2x_2 x_6 + x_2 x_8 - 4,0616$$

$$f_{10}(\mathbf{x}) = x_3 x_9 + x_5 x_9 + x_7 x_9 - 1,9791$$

$$f_{11}(\mathbf{x}) = x_2 x_4 x_9 + x_2 x_6 x_9 + x_2 x_8 x_9 + x_1 x_{10} - 1,9115$$

$$f_{12}(\mathbf{x}) = -x_3 x_{10} x_{11} - x_5 x_{10} x_{11} - x_7 x_{10} x_{11} + x_4 x_{12} + x_6 x_{12} + x_8 x_{12} - 0,4077$$

Problem des18_3

Dimenzija problema:8

Varijable: $\mathbf{x} = (a_{10}, a_{20}, a_{21}, a_{22}, a_{30}, a_{31}, a_{32}, a_{33})$

$$f_1(\mathbf{x}) = 6 \cdot a_{33} \cdot a_{10} \cdot a_{20} + 10 \cdot a_{22} \cdot a_{10} \cdot a_{31} + 8 \cdot a_{32} \cdot a_{10} \cdot a_{21} - 162 \cdot a_{10}^2 \cdot a_{21} + 16 \cdot a_{21} \cdot a_{30} + 14 \cdot a_{31} \cdot a_{20} + 48 \cdot a_{10} \cdot a_{30}$$

$$f_2(\mathbf{x}) = 15 \cdot a_{33} \cdot a_{10} \cdot a_{21} - 162 \cdot a_{10}^2 \cdot a_{22} - 312 \cdot a_{10} \cdot a_{20} + 24 \cdot a_{10} \cdot a_{30} + 27 \cdot a_{31} \cdot a_{21} + 24 \cdot a_{32} \cdot a_{20} + 18 \cdot a_{22} \cdot a_{10} \cdot a_{32} + 30 \cdot a_{22} \cdot a_{30} + 84 \cdot a_{31} \cdot a_{10}$$

$$f_3(\mathbf{x}) = -240 \cdot a_{10} + 420 \cdot a_{33} - 64 \cdot a_{22} + 112 \cdot a_{32}$$

$$f_4(\mathbf{x}) = 180 \cdot a_{33} \cdot a_{10} - 284 \cdot a_{22} \cdot a_{10} - 162 \cdot a_{10}^2 + 60 \cdot a_{22} \cdot a_{32} + 50 \cdot a_{32} \cdot a_{10} + 70 \cdot a_{30} + 55 \cdot a_{33} \cdot a_{21} + 260 \cdot a_{31} - 112 \cdot a_{20}$$

$$f_5(\mathbf{x}) = 66 \cdot a_{33} \cdot a_{10} + 336 \cdot a_{32} + 90 \cdot a_{31} + 78 \cdot a_{22} \cdot a_{33} - 1056 \cdot a_{10} - 90 \cdot a_{21}$$

$$f_6(\mathbf{x}) = 136 \cdot a_{33} - 136$$

$$f_7(\mathbf{x}) = 4 \cdot a_{22} \cdot a_{10} \cdot a_{30} + 2 \cdot a_{32} \cdot a_{10} \cdot a_{20} + 6 \cdot a_{20} \cdot a_{30} - 162 \cdot a_{10}^2 \cdot a_{20} + 3 \cdot a_{31} \cdot a_{21} \cdot a_{10}$$

$$f_8(\mathbf{x}) = 28 \cdot a_{22} \cdot a_{10} \cdot a_{33} + 192 \cdot a_{30} + 128 \cdot a_{32} \cdot a_{10} + 36 \cdot a_{31} \cdot a_{20} + 36 \cdot a_{33} \cdot a_{20} - 300 \cdot a_{10} \cdot a_{21} + 40 \cdot a_{32} \cdot a_{21} - 648 \cdot a_{10}^2 + 44 \cdot a_{22} \cdot a_{31}$$

Problem des22_24

Dimenzija problema:10

Varijable: $\mathbf{x} = (a_{20}, a_{21}, a_{22}, a_{23}, a_{30}, a_{31}, a_{32}, a_{33}, a_{34}, a_{35})$

$$f_1(\mathbf{x}) = 16 a_{20} a_{32} + 18 a_{21} a_{31} + 20 a_{22} a_{30}$$

$$f_2(\mathbf{x}) = -80 a_{23} + 180 a_{34} + 855 a_{35}$$

$$f_3(\mathbf{x}) = 7 a_{20} a_{31} + 8 a_{21} a_{30}$$

$$f_4(\mathbf{x}) = 210 a_{35} - 210$$

$$f_5(\mathbf{x}) = 40 a_{20} a_{34} + 44 a_{21} a_{33} + 48 a_{22} a_{32} + 52 a_{23} a_{31} + 280 a_{30}$$

$$f_6(\mathbf{x}) = 27 a_{20} a_{33} + 30 a_{21} a_{32} + 33 a_{22} a_{31} + 36 a_{23} a_{30}$$

$$f_7(\mathbf{x}) = 55 a_{20} a_{35} + 60 a_{21} a_{34} + 65 a_{22} a_{33} + 70 a_{23} a_{32} + 80 a_{30} + 375 a_{31}$$

$$f_8(\mathbf{x}) = 78a_{21}a_{35} + 84a_{22}a_{34} + 90a_{23}a_{33} - 170a_{20} + 102a_{31} + 480a_{32}$$

$$f_9(\mathbf{x}) = 136a_{23}a_{35} - 114a_{22} + 152a_{33} + 720a_{34}$$

$$f_{10}(\mathbf{x}) = 105a_{22}a_{35} + 112a_{23}a_{34} - 144a_{21} + 126a_{32} + 595a_{33}$$

Problem discrete3

Dimenzija problema:8

Variable: $\mathbf{x} = (y, z, t, u, v, s, a, b)$

$$\begin{aligned} f_1(\mathbf{x}) = & -1,37539569915948E-01y^2 - 9,16930466106320E-02yz - 6,87697849579740E-02yt \\ & - 5,50158279663792E-02yu - 4,58465233053160E-02yv - 3,92970199759852E-02ys \\ & - 3,43848924789870E-02ya + 4,12618709747844E+00y - 4,40126623731034E+00 \end{aligned}$$

$$\begin{aligned} f_2(\mathbf{x}) = & -1,78264047725719E-01yz - 1,33698035794289E-01z^2 - 1,06958428635431E-01zt \\ & - 8,91320238628593E-02zu - 7,63988775967365E-02zv - 6,68490178971445E-02zs \\ & - 5,94213492419062E-02za + 4,01094107382867E+00z - 4,27833714541725E+00 \end{aligned}$$

$$\begin{aligned} f_3(\mathbf{x}) = & -1,96868164299851E-01yt - 1,57494531439881E-01zt - 1,31245442866567E-01t^2 \\ & - 1,12496093885629E-01tu - 9,84340821499253E-02tv - 8,74969619110448E-02ts \\ & - 7,87472657199403E-02ta + 3,93736328599701E+00t - 4,19985417173015E+00 \end{aligned}$$

$$\begin{aligned} f_4(\mathbf{x}) = & -2,07217047148772E-01yu - 1,72680872623977E-01zu - 1,48012176534837E-01tu \\ & - 1,29510654467983E-01u^2 - 1,15120581749318E-01uv - 1,03608523574386E-01us \\ & - 9,41895668858055E-02ua + 3,88531963403948E+00u - 4,14434094297544E+00 \end{aligned}$$

$$\begin{aligned} f_5(\mathbf{x}) = & -2,13678947124996E-01yv - 1,83153383249996E-01zv - 1,60259210343747E-01tv \\ & - 1,42452631416664E-01uv - 1,28207368274997E-01v^2 - 1,16552152977270E-01vs \\ & - 1,06839473562498E-01va + 3,84622104824992E+00v - 4,10263578479991E+00 \end{aligned}$$

$$\begin{aligned} f_6(\mathbf{x}) = & -2,18035916725452E-01ys - 1,90781427134770E-01zs - 1,69583490786463E-01ts \\ & - 1,52625141707816E-01us - 1,38750128825288E-01vs - 1,27187618089847E-01s^2 \\ & - 1,17403955159859E-01sa + 3,81562854269541E+00s - 4,07000377887510E+00 \end{aligned}$$

$$\begin{aligned} f_7(\mathbf{x}) = & -2,21139931193274E-01ya - 1,96568827727354E-01za - 1,76911944954619E-01ta \\ & - 1,60829040867835E-01ua - 1,47426620795516E-01va - 1,36086111503553E-01sa \\ & - 1,26365674967585E-01a^2 + 3,79097024902755E+00a - 4,04370159896272E+00 \end{aligned}$$

$$\begin{aligned} f_8(\mathbf{x}) = & -2,23445119966269E-01yb - 2,01100607969642E-01zb - 1,82818734517856E-01tb \\ & - 1,67583839974702E-01ub - 1,54692775361263E-01vb - 1,43643291406887E-01sb \\ & - 1,34067071979761E-01ab + 3,77063639943078E+00b - 4,02201215939284E+00 \end{aligned}$$

Problem eco5

Dimenzija problema:5

Variable: $\mathbf{x} = (x_1, x_2, x_3, x_4, u_5)$

$$f_1(\mathbf{x}) = x_1 \cdot x_5 + x_1 \cdot x_2 \cdot x_5 + x_2 \cdot x_3 \cdot x_5 + x_3 \cdot x_4 \cdot x_5 - 1$$

$$f_2(\mathbf{x}) = x_2 \cdot x_5 + x_1 \cdot x_3 \cdot x_5 + x_2 \cdot x_4 \cdot x_5 - 2$$

$$f_3(\mathbf{x}) = x_3 \cdot x_5 + x_1 \cdot x_4 \cdot x_5 - 3$$

$$f_4(\mathbf{x}) = x_4 \cdot x_5 - 4$$

$$f_5(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + 1$$

Problem eco6

Dimenzija problema:6

Variable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$

$$f_1(\mathbf{x}) = x_1 \cdot x_6 + x_1 \cdot x_2 \cdot x_6 + x_2 \cdot x_3 \cdot x_6 + x_3 \cdot x_4 \cdot x_6 + x_4 \cdot x_5 \cdot x_6 - 1$$

$$f_2(\mathbf{x}) = x_2 \cdot x_6 + x_1 \cdot x_3 \cdot x_6 + x_2 \cdot x_4 \cdot x_6 + x_3 \cdot x_5 \cdot x_6 - 2$$

$$f_3(\mathbf{x}) = x_3 \cdot x_6 + x_1 \cdot x_4 \cdot x_6 + x_2 \cdot x_5 \cdot x_6 - 3$$

$$f_4(\mathbf{x}) = x_4 \cdot x_6 + x_1 \cdot x_5 \cdot x_6 - 4$$

$$f_5(\mathbf{x}) = x_5 \cdot x_6 - 5$$

$$f_6(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + x_5 + 1$$

Problem eco7

Dimenzija problema:7

Variable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$

$$f_1(\mathbf{x}) = x_1 \cdot x_7 + x_1 \cdot x_2 \cdot x_7 + x_2 \cdot x_3 \cdot x_7 + x_3 \cdot x_4 \cdot x_7 + x_4 \cdot x_5 \cdot x_7 + x_5 \cdot x_6 \cdot x_7 - 1$$

$$f_2(\mathbf{x}) = x_2 \cdot x_7 + x_1 \cdot x_3 \cdot x_7 + x_2 \cdot x_4 \cdot x_7 + x_3 \cdot x_5 \cdot x_7 + x_4 \cdot x_6 \cdot x_7 - 2$$

$$f_3(\mathbf{x}) = x_3 \cdot x_7 + x_1 \cdot x_4 \cdot x_7 + x_2 \cdot x_5 \cdot x_7 + x_3 \cdot x_6 \cdot x_7 - 3$$

$$f_4(\mathbf{x}) = x_4 \cdot x_7 + x_1 \cdot x_5 \cdot x_7 + x_2 \cdot x_6 \cdot x_7 - 4$$

$$f_5(\mathbf{x}) = x_5 \cdot x_7 + x_1 \cdot x_6 \cdot x_7 - 5$$

$$f_6(\mathbf{x}) = x_6 \cdot x_7 - 6$$

$$f_7(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + 1$$

Problem eco8

Dimenzija problema:8

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$

$$f_1(\mathbf{x}) = x_1 \cdot x_8 + x_1 \cdot x_2 \cdot x_8 + x_2 \cdot x_3 \cdot x_8 + x_3 \cdot x_4 \cdot x_8 + x_4 \cdot x_5 \cdot x_8 + x_5 \cdot x_6 \cdot x_8 + x_6 \cdot x_7 \cdot x_8 - 1$$

$$f_2(\mathbf{x}) = x_2 \cdot x_8 + x_1 \cdot x_3 \cdot x_8 + x_2 \cdot x_4 \cdot x_8 + x_3 \cdot x_5 \cdot x_8 + x_4 \cdot x_6 \cdot x_8 + x_5 \cdot x_7 \cdot x_8 - 2$$

$$f_3(\mathbf{x}) = x_3 \cdot x_8 + x_1 \cdot x_4 \cdot x_8 + x_2 \cdot x_5 \cdot x_8 + x_3 \cdot x_6 \cdot x_8 + x_4 \cdot x_7 \cdot x_8 - 3$$

$$f_4(\mathbf{x}) = x_4 \cdot x_8 + x_1 \cdot x_5 \cdot x_8 + x_2 \cdot x_6 \cdot x_8 + x_3 \cdot x_7 \cdot x_8 - 4$$

$$f_5(\mathbf{x}) = x_5 \cdot x_8 + x_1 \cdot x_6 \cdot x_8 + x_2 \cdot x_7 \cdot x_8 - 5$$

$$f_6(\mathbf{x}) = x_6 \cdot x_8 + x_1 \cdot x_7 \cdot x_8 - 6$$

$$f_7(\mathbf{x}) = x_7 \cdot x_8 - 7$$

$$f_8(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + 1$$

Problem filter9

Dimenzija problema:9

Varijable: $\mathbf{x} = (a, b, m_1, m_2, m_3, m_4, m_5, m_6, m_7)$

$$f_1(\mathbf{x}) = m_2 m_4 m_6 - 0,01$$

$$f_2(\mathbf{x}) = a m_4 b - 0,014$$

$$f_3(\mathbf{x}) = a^2 + m_1^2 - 0,08$$

$$f_4(\mathbf{x}) = b^2 + m_7^2 - 0,74$$

$$f_5(\mathbf{x}) = m_3^2 + m_5^2 + m_4^2 + m_2^2 + m_6^2 - 0,9401$$

$$f_6(\mathbf{x}) = m_4^2 m_6^2 + m_2^2 m_4^2 + m_3^2 m_6^2 + m_2^2 m_5^2 + m_3^2 m_5^2 + m_2^2 m_6^2 - 0,038589$$

$$f_7(\mathbf{x}) = m_1 m_3 m_5 m_7 - m_6 m_1 m_3 b + m_2 m_6 a b - m_2 a m_5 m_7 + 0,0081$$

$$f_8(\mathbf{x}) = -m_1 m_2 m_3 m_4 b - a m_4 m_5 m_6 m_7 + a m_4 b m_6^2 + a m_2^2 m_4 b + 0,00156$$

$$f_9(\mathbf{x}) = -m_4^2 m_7^2 - m_3^2 m_7^2 + 2 m_5 m_6 b m_7 - m_2^2 m_7^2 - m_5^2 m_7^2 - m_3^2 b^2 - b^2 m_6^2 - m_2^2 b^2 + 0,679325$$

Problem freudenstein-roth

Dimenzija problema:2

Varijable: $\mathbf{x} = (x, y)$

$$f_1(\mathbf{x}) = x - y^3 + 5y^2 - 2y - 13$$

$$f_2(\mathbf{x}) = x + y^3 + y^2 - 14y - 29$$

Problem fourbar

Dimenzija problema:4

Varijable: $\mathbf{x} = (X_1, X_2, Y_1, Y_2)$

$$\begin{aligned}
f_1(\mathbf{x}) = & 0,01692601 \cdot X_1^2 \cdot Y_1^2 - 0,888509280014 \cdot X_1^2 \cdot Y_2^2 + 0,0411717692438 \cdot X_2^2 \cdot Y_1^2 \\
& - 0,00437457395884 \cdot X_2^2 \cdot Y_2^2 + 0,331480641249 \cdot X_1 \cdot X_2 \cdot Y_1^2 - 1,38036964668 \cdot X_1 \cdot X_2 \cdot Y_2^2 \\
& - 0,270492270191 \cdot X_1^2 \cdot Y_1 \cdot Y_2 + 1,44135801774 \cdot X_2^2 \cdot Y_1 \cdot Y_2 + 0,859888946812 \cdot X_1 \cdot X_2 \cdot Y_1 \cdot Y_2 \\
& + 0,0791489659197 \cdot X_1^2 \cdot Y_1 - 0,00336032777032 \cdot X_1^2 \cdot Y_2 - 0,0620826738427 \cdot X_1 \cdot Y_1^2 \\
& + 0,501879647495 \cdot X_1 \cdot Y_2^2 + 0,647156236961 \cdot X_2^2 \cdot Y_1 + 0,0926311741907 \cdot X_2^2 \cdot Y_2 \\
& - 0,255000006226 \cdot X_2 \cdot Y_1^2 - 0,0896892386081 \cdot X_2 \cdot Y_2^2 - 0,568007271041 \cdot X_1 \cdot X_2 \cdot Y_2 \\
& + 0,095991501961 \cdot X_1 \cdot X_2 \cdot Y_1 + 0,165310767618 \cdot X_1 \cdot Y_1 \cdot Y_2 - 0,563962321337 \cdot X_2 \cdot Y_1 \cdot Y_2 \\
& - 0,0784871167595 \cdot X_1 \cdot Y_1 - 0,0784871167595 \cdot X_2 \cdot Y_2 + 0,011807283256 \cdot X_1 \cdot Y_2 \\
& - 0,011807283256 \cdot X_2 \cdot Y_1 + 0,0422876985355 \cdot X_1^2 + 0,0422876985355 \cdot X_2^2 \\
& + 0,0372427422943 \cdot Y_1^2 + 0,0372427422943 \cdot Y_2^2
\end{aligned}$$

$$\begin{aligned}
f_2(\mathbf{x}) = & 0,518178672335 \cdot X_1^2 \cdot Y_1^2 - 0,0414464807343 \cdot X_1^2 \cdot Y_2^2 + 2,63600135179 \cdot X_2^2 \cdot Y_1^2 \\
& - 0,799490472298 \cdot X_2^2 \cdot Y_2^2 + 0,29442805494 \cdot X_1^2 \cdot Y_1 \cdot Y_2 + 1,46551534655 \cdot X_2^2 \cdot Y_1 \cdot Y_2 \\
& - 0,631878110759 \cdot X_1 \cdot X_2 \cdot Y_1^2 - 1,80296540237 \cdot X_1 \cdot X_2 \cdot Y_2^2 - 2,87586667102 \cdot X_1 \cdot X_2 \cdot Y_1 \cdot Y_2 \\
& - 0,987856648177 \cdot X_1^2 \cdot Y_1 - 0,530579106676 \cdot X_1^2 \cdot Y_2 - 0,0397576281649 \cdot X_1 \cdot Y_1^2 \\
& + 0,317719102869 \cdot X_1 \cdot Y_2^2 - 1,93710490787 \cdot X_2^2 \cdot Y_1 + 0,00127693327315 \cdot X_2^2 \cdot Y_2 \\
& - 0,581380074072 \cdot X_2 \cdot Y_1^2 - 0,0672137066743 \cdot X_2 \cdot Y_2^2 + 0,531856039949 \cdot X_1 \cdot X_2 \cdot Y_1 \\
& + 0,949248259696 \cdot X_1 \cdot X_2 \cdot Y_2 + 0,514166367398 \cdot X_1 \cdot Y_1 \cdot Y_2 - 0,357476731033 \cdot X_2 \cdot Y_1 \cdot Y_2 \\
& + 0,140965913657 \cdot X_1 \cdot Y_1 + 0,140965913657 \cdot X_2 \cdot Y_2 - 0,153347218606 \cdot X_1 \cdot Y_2 \\
& + 0,153347218606 \cdot X_2 \cdot Y_1 + 0,283274882058 \cdot X_1^2 + 0,283274882058 \cdot X_2^2 \\
& + 0,0382903330079 \cdot Y_1^2 + 0,0382903330079 \cdot Y_2^2
\end{aligned}$$

$$\begin{aligned}
 f_3(\mathbf{x}) = & 0,0233560008057 \cdot X_1^2 \cdot Y_1^2 - 0,00428427501149 \cdot X_1^2 \cdot Y_1 \cdot Y_2 - 0,792756311827 \cdot X_1^2 \cdot Y_2^2 \\
 & + 0,0492185850289 \cdot X_2^2 \cdot Y_2^2 + 0,0759264856293 \cdot X_1 \cdot X_2 \cdot Y_1^2 + 1,14839711492 \cdot X_1 \cdot X_2 \cdot Y_1 \cdot Y_2 \\
 & - 0,283066217262 \cdot X_2^2 \cdot Y_1^2 + 0,460041521291 \cdot X_2^2 \cdot Y_1 \cdot Y_2 - 0,388399310674 \cdot X_1 \cdot X_2 \cdot Y_2^2 \\
 & - 0,0561169736293 \cdot X_1 \cdot Y_1^2 + 0,485064247792 \cdot X_1 \cdot Y_2^2 + 0,0689567235492 \cdot X_1^2 \cdot Y_1 \\
 & - 0,115620658768 \cdot X_1^2 \cdot Y_2 - 0,13286905328 \cdot X_2 \cdot Y_1^2 - 0,084375901147 \cdot X_2 \cdot Y_2^2 \\
 & + 0,639964831612 \cdot X_2^2 \cdot Y_1 + 0,101386684276 \cdot X_2^2 \cdot Y_2 + 0,217007343044 \cdot X_1 \cdot X_2 \cdot Y_1 \\
 & - 0,571008108063 \cdot X_1 \cdot X_2 \cdot Y_2 + 0,0484931521334 \cdot X_1 \cdot Y_1 \cdot Y_2 - 0,541181221422 \cdot X_2 \cdot Y_1 \cdot Y_2 \\
 & - 0,00363197918253 \cdot X_1 \cdot Y_2 + 0,00363197918253 \cdot X_2 \cdot Y_1 - 0,0781302968652 \cdot X_1 \cdot Y_1 \\
 & - 0,0781302968652 \cdot X_2 \cdot Y_2 + 0,0471311092612 \cdot X_1^2 + 0,0471311092612 \cdot X_2^2 \\
 & + 0,0324495575052 \cdot Y_1^2 + 0,0324495575052 \cdot Y_2^2
 \end{aligned}$$

$$\begin{aligned}
 f_4(\mathbf{x}) = & 0,393707415641 \cdot X_1^2 \cdot Y_1^2 + 0,59841456862 \cdot X_1^2 \cdot Y_2^2 + 0,0735854940135 \cdot X_2^2 \cdot Y_1^2 \\
 & + 0,0548997238169 \cdot X_2^2 \cdot Y_2^2 + 0,0116156836985 \cdot X_1^2 \cdot Y_1 \cdot Y_2 + 0,0699694273575 \cdot X_2^2 \cdot Y_1 \cdot Y_2 \\
 & - 0,305757340849 \cdot X_1 \cdot X_2 \cdot Y_1^2 - 0,364111084508 \cdot X_1 \cdot X_2 \cdot Y_2^2 - 0,223392923175 \cdot X_1 \cdot X_2 \cdot Y_1 \cdot Y_2 \\
 & + 0,0996725944534 \cdot X_1^2 \cdot Y_1 + 0,0113936468426 \cdot X_1^2 \cdot Y_2 - 0,381205205249 \cdot X_1 \cdot Y_1^2 \\
 & - 0,473402150235 \cdot X_1 \cdot Y_2^2 - 0,0213613191759 \cdot X_2^2 \cdot Y_1 - 0,0372595571271 \cdot X_2^2 \cdot Y_2 \\
 & + 0,148904552394 \cdot X_2 \cdot Y_1^2 + 0,142408744984 \cdot X_2 \cdot Y_2^2 - 0,0486532039697 \cdot X_1 \cdot X_2 \cdot Y_1 \\
 & + 0,121033913629 \cdot X_1 \cdot X_2 \cdot Y_2 - 0,00649580741066 \cdot X_1 \cdot Y_1 \cdot Y_2 + 0,092196944986 \cdot X_2 \cdot Y_1 \cdot Y_2 \\
 & - 0,0483106652705 \cdot X_1 \cdot Y_1 - 0,0483106652705 \cdot X_2 \cdot Y_2 - 0,00316794272326 \cdot X_1 \cdot Y_2 \\
 & + 0,00316794272326 \cdot X_2 \cdot Y_1 + 0,00634952598374 \cdot X_1^2 + 0,00634952598374 \cdot X_2^2 \\
 & + 0,0922886309144 \cdot Y_1^2 + 0,0922886309144 \cdot Y_2^2
 \end{aligned}$$

Problem game4two

Dimenzija problema:4

Varijable: $\mathbf{x} = (p_1, p_2, p_3, p_4)$

$$\begin{aligned}
 f_1(\mathbf{x}) = & -1,122525605 \cdot p_3 - 0,1202590019 \cdot p_3 \cdot p_2 + 1,078803537 \cdot p_2 + 0,7696590455 \cdot p_4 \\
 & + 0,1628390188 \cdot p_4 \cdot p_3 \cdot p_2 - 0,1163105815 \cdot p_4 \cdot p_3 - 0,5333928492 \cdot p_4 \cdot p_2 \\
 & - 0,6823118878
 \end{aligned}$$

$$\begin{aligned}
 f_2(\mathbf{x}) = & -0,4266067331 \cdot p_3 + 0,9756757996 \cdot p_4 + 0,646954510E-01 \cdot p_4 \cdot p_3 \\
 & - 0,2142068753 \cdot p_3 \cdot p_1 + 2,814522341 \cdot p_1 + 0,3980853635 \cdot p_4 \cdot p_3 \cdot p_1 \\
 & - 2,546266040 \cdot p_4 \cdot p_1 - 1,15791820
 \end{aligned}$$

$$\begin{aligned}
 f_3(\mathbf{x}) &= 2,023260100 \cdot p_2 - 0,1342402078 \cdot p_4 - 0,3904810713 \cdot p_4 \cdot p_2 + 0,4274147938 \cdot p_1 \\
 &+ 0,5269506245 \cdot p_4 \cdot p_1 - 3,069473137 \cdot p_2 \cdot p_1 + 0,3284270487 \cdot p_4 \cdot p_2 \cdot p_1 \\
 &- 0,9128973443
 \end{aligned}$$

$$\begin{aligned}
 f_4(\mathbf{x}) &= -0,6256820158 \cdot p_3 - 0,1614530476 \cdot p_3 \cdot p_2 + 1,080721123 \cdot p_2 \\
 &+ 1,366746822 \cdot p_3 \cdot p_1 - 2,347725424 \cdot p_1 + 0,5941402017 \cdot p_2 \cdot p_1 \\
 &- 2,233884240 \cdot p_3 \cdot p_2 \cdot p_1 + 1,026687422
 \end{aligned}$$

Problem game5two

Dimenzija problema:5

Varijable: $\mathbf{x} = (p_1 p_2 p_3 p_4 p_5)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= -1,390350657 p_3 + 0,4641393136 p_3 p_2 - 0,6266605268 p_2 - 1,400171891 p_4 \\
 &- 2,090683800 p_4 p_3 p_2 + 4,089263882 p_4 p_3 + 1,129827638 p_4 p_2 + 1,881614464 p_5 p_4 p_3 p_2 \\
 &+ 0,4716169661 p_5 p_3 p_2 - 0,8625849122 p_5 p_4 p_2 - 1,398871056 p_5 p_2 + 0,9599693844 p_5 \\
 &+ 0,714025397E-01 p_5 p_4 + 0,1073802376 p_5 p_3 - 0,9259664538 p_5 p_4 p_3 - 0,2067814278
 \end{aligned}$$

$$\begin{aligned}
 f_2(\mathbf{x}) &= -1,196136754 p_3 - 0,9249804195 p_4 + 0,3188761009 p_4 p_3 - 1,045301323 p_3 p_1 \\
 &- 0,0306661782 p_1 + 0,5987012929 p_4 p_3 p_1 - 0,4448182692 p_4 p_1 - 0,3908068031 p_5 p_4 p_3 p_1 \\
 &- 1,212939725 p_5 + 2,586129779 p_5 p_4 - 0,1180169224 p_5 p_3 - 1,051519507 p_5 p_4 p_3 \\
 &+ 2,134979375 p_5 p_3 p_1 - 1,337061849 p_5 p_4 p_1 - 0,2961272671 p_5 p_1 + 0,7316111016
 \end{aligned}$$

$$\begin{aligned}
 f_3(\mathbf{x}) &= 2,272943163 p_2 - 0,4131564265 p_4 - 1,920446680 p_4 p_2 + 0,80509234E-2 p_1 \\
 &+ 1,342851102 p_4 p_1 - 2,979502184 p_2 p_1 + 3,391571834 p_4 p_2 p_1 - 0,5975693742 p_5 p_4 p_2 \\
 &+ 0,3002794716 p_5 p_2 - 0,7893445350 p_5 + 1,276948001 p_5 p_4 - 4,601376311 p_5 p_4 p_1 \\
 &+ 2,356804322 p_5 p_1 + 3,498840190 p_5 p_4 p_2 p_1 - 1,355375015 p_5 p_2 p_1 - 1,231070236
 \end{aligned}$$

$$\begin{aligned}
 f_4(\mathbf{x}) &= -2,206336116 p_3 + 2,318673689 p_3 p_2 - 1,267478048 p_2 + 1,110654516 p_3 p_1 \\
 &+ 1,533592098 p_1 - 1,872504375 p_2 p_1 + 0,3299103675 p_3 p_2 p_1 - 3,400750472 p_5 p_3 p_2 \\
 &+ 2,093674516 p_5 p_2 - 1,772874182 p_5 + 2,993821915 p_5 p_3 - 1,356762392 p_5 p_3 p_1 \\
 &+ 0,637534233E-01 p_5 p_1 + 0,5870371377 p_5 p_2 p_1 + 1,018269743 p_5 p_3 p_2 p_1 + 1,400431557
 \end{aligned}$$

$$\begin{aligned}
 f_5(\mathbf{x}) &= -2,522718869 p_3 + 0,8323646978 p_3 p_2 - 1,375039881 p_2 - 0,3055443755 p_4 \\
 &+ 0,6760632172 p_4 p_3 p_2 - 0,4262974456 p_4 p_3 + 1,268255245 p_3 p_1 + 0,5352674901 p_4 p_2 \\
 &- 1,024495558 p_1 + 1,818275404 p_4 p_3 p_1 - 1,354832512 p_4 p_1 - 1,595112039 p_2 p_1 \\
 &+ 2,237956242 p_4 p_2 p_1 + 3,370102170 p_3 p_2 p_1 - 3,465040669 p_4 p_3 p_2 p_1 + 2,132631128
 \end{aligned}$$

Problem game6two

Dimenzija problema:6

Variable: $\mathbf{x} = (p_1, p_2, p_3, p_4, p_5, p_6)$

$$\begin{aligned}
f_1(\mathbf{x}) &= 0,5629821171 p_6 p_4 p_3 + 0,8266692450 p_3 - 0,6202580689 p_3 p_2 + 2,111265523 p_2 \\
&- 0,9348508860 p_4 + 0,4061841780 p_4 p_3 p_2 - 0,214857892E-01 p_4 p_3 - 0,6625588651 p_4 p_2 \\
&- 2,74225814 p_5 p_4 p_3 p_2 + 2,692695392 p_5 p_3 p_2 + 1,50496027 p_5 p_4 p_2 - 1,99954916 p_5 p_2 \\
&+ 1,285858413 p_5 - 0,2407183338 p_5 p_4 - 3,252304180 p_5 p_3 + 4,312145706 p_5 p_4 p_3 \\
&+ 2,736883996 p_6 p_5 p_3 + 1,86041124 p_6 p_5 p_4 + 0,615948718 p_6 - 0,42589966 p_6 p_5 p_4 p_3 p_2 \\
&- 1,734729605 p_6 p_5 - 0,4835556072 p_6 p_4 - 0,7430261699 p_6 p_3 + 1,339997538 p_6 p_4 p_3 p_2 \\
&- 0,9550267 p_6 p_5 p_3 p_2 + 1,04648616 p_6 p_3 p_2 - 1,63036709 p_6 p_5 p_4 p_2 + 1,37680566 p_6 p_4 p_2 \\
&+ 1,658908057 p_6 p_5 p_2 - 2,594596197 p_6 p_5 p_4 p_3 - 2,139964727 p_6 p_2 - 0,7493708055 \\
f_2(\mathbf{x}) &= 1,7714586 p_4 p_3 p_1 + 2,6157015 p_5 p_4 p_3 p_1 + 0,3725029 p_5 p_1 p_6 p_3 - 0,7954528 p_5 p_1 p_6 p_4 \\
&+ 0,2151125848 p_4 + 0,6865208531 p_3 - 0,4038474650 p_1 p_6 + 0,1737585616 p_5 \\
&+ 2,808937532 p_6 p_5 p_4 p_3 + 0,6722570098 p_6 p_5 - 0,9994809109 p_3 p_1 - 0,8097227494 p_1 \\
&+ 0,389424373 p_6 p_5 p_4 p_3 p_1 + 2,06959403 p_1 p_6 p_4 + 0,02699805 p_6 - 3,457070843 p_1 p_6 p_4 p_3 \\
&+ 0,2875509765 p_5 p_1 p_6 + 1,207766011 p_6 p_4 p_3 - 1,445086856 p_5 p_4 p_3 - 2,00985221 p_5 p_1 p_4 \\
&+ 1,4265185 p_1 p_6 p_3 - 1,925890289 p_6 p_5 p_3 - 0,3335314847 p_6 p_5 p_4 - 1,223483720 p_6 p_4 \\
&+ 0,1347934066 p_6 p_3 + 0,7709280197 p_5 p_3 + 0,6006729886 p_1 p_4 - 1,880629955 p_4 p_3 \\
&+ 0,1851751562 p_5 p_4 - 0,7295830375 p_5 p_3 p_1 + 0,5936630617 p_5 p_1 + 0,03949729 \\
f_3(\mathbf{x}) &= 5,808829230 p_5 p_1 p_6 p_4 - 2,494428392 p_4 - 2,044053089 p_2 - 0,1032723202 p_1 p_6 \\
&+ 0,7060916957 p_2 p_1 - 1,620286322 p_5 + 1,326107436 p_6 p_5 + 0,1912503521 p_4 p_2 p_1 \\
&- 1,505297482 p_1 - 2,235082597 p_1 p_6 p_4 - 0,4180766069 p_6 - 4,811635696 p_6 p_5 p_4 p_2 p_1 \\
&+ 1,273203178 p_5 p_4 p_2 p_1 + 1,134504692 p_5 p_1 p_6 p_2 + 0,4679302881 p_1 p_6 p_4 p_2 \\
&+ 0,8865485848 p_6 p_2 p_1 + 2,509841675 p_6 p_5 p_4 p_2 - 3,186351822 p_5 p_4 p_2 \\
&- 0,7092981953 p_6 p_4 p_2 - 1,067452138 p_6 p_5 p_2 + 2,331785278 p_4 p_2 \\
&+ 0,1605853472 p_6 p_2 + 1,852990230 p_5 p_2 + 1,340919744 - 2,276984784 p_5 p_1 p_6 \\
&- 2,878039506 p_5 p_1 p_4 - 0,2889306883 p_5 p_1 p_2 - 2,389199972 p_6 p_5 p_4 \\
&+ 1,200120596 p_6 p_4 + 1,310550265 p_1 p_4 + 2,929240922 p_5 p_4 + 1,455547595 p_5 p_1
\end{aligned}$$

$$\begin{aligned}
 f_4(\mathbf{x}) &= -1,193974442 p_5 p_3 p_2 - 0,1372075624 p_5 p_1 p_6 p_3 + 0,1747445127 p_3 + 1,332629531 p_2 \\
 &+ 0,3196306546 p_1 p_6 + 1,795174628 p_2 p_1 + 1,671472037 p_5 - 1,149350540 p_6 p_5 \\
 &- 0,2005768432 p_3 p_1 + 0,8455955098 p_1 - 0,2439006599 p_6 - 1,935808119 p_5 p_1 p_6 p_2 \\
 &+ 0,5691530995 p_6 p_2 p_1 + 3,002609648 p_6 p_5 p_2 - 0,6252462145 p_6 p_2 - 1,682232741 p_5 p_2 \\
 &+ 3,852486370 p_5 p_3 p_2 p_1 - 1,228434777 p_3 p_2 p_1 + 0,9334203703 p_6 p_5 p_3 p_2 p_1 \\
 &+ 0,946202549 p_1 p_6 p_3 p_2 - 1,97028668 p_6 p_5 p_3 p_2 - 0,662968924 + 0,0439447 p_6 p_2 p_3 \\
 &- 0,051164686 p_3 p_2 + 0,4527026 p_5 p_1 p_6 + 1,70048581 p_5 p_1 p_2 - 1,564344832 p_1 p_6 p_3 \\
 &+ 1,02840262 p_6 p_5 p_3 + 0,5192817 p_6 p_3 - 0,733612559 p_5 p_3 \\
 &- 0,0013157604 p_5 p_3 p_1 - 2,650253107 p_5 p_1 \\
 f_5(\mathbf{x}) &= -1,874451936 p_4 p_3 p_1 + 1,539768136 p_4 + 0,8522057726 p_3 - 0,3598665646 p_2 \\
 &- 0,9929167089 p_1 p_6 - 0,2118178799 p_2 p_1 + 1,275233727 p_4 p_2 p_1 + 1,867803736 p_3 p_1 \\
 &+ 1,992250869 p_1 + 2,25522893 p_1 p_6 p_4 + 1,69271425 p_6 - 2,885311582 p_1 p_6 p_4 p_3 \\
 &- 1,825516 p_1 p_6 p_4 p_2 + 2,2968428 p_6 p_2 p_1 - 0,19097805 p_6 p_4 p_2 - 0,874856287 p_4 p_2 \\
 &- 1,695886109 p_6 p_2 - 3,110424395 p_3 p_2 p_1 - 5,099122276 p_1 p_6 p_3 p_2 \\
 &+ 1,826037898 p_4 p_3 p_2 p_1 + 3,192081622 p_6 p_2 p_3 + 0,3022528926 p_3 p_2 \\
 &+ 2,470462821 p_6 p_4 p_3 + 3,841265503 p_6 p_4 p_3 p_2 p_1 + 0,5003690270 p_4 p_3 p_2 \\
 &+ 0,8618106168 p_1 p_6 p_3 - 1,513768682 - 1,326021072 p_6 p_4 - 1,960674929 p_6 p_3 \\
 &- 0,6415141099 p_1 p_4 - 1,724428781 p_4 p_3 - 2,224289853 p_6 p_4 p_3 p_2 \\
 f_6(\mathbf{x}) &= 1,521838778 p_5 p_3 p_2 + 0,9229492072 p_4 p_3 p_1 - 1,914447458 p_5 p_4 p_3 p_1 \\
 &- 0,6995253769 p_4 - 0,2695877097 p_3 - 1,700516904 p_2 + 1,932658121 p_2 p_1 \\
 &- 0,8722575175 p_5 + 2,605004255 p_4 p_2 p_1 - 1,557559724 p_3 p_1 - 2,125003114 p_1 \\
 &- 2,462133076 p_5 p_4 p_2 p_1 - 0,1032684102 p_5 p_4 p_2 - 1,158260603 p_4 p_2 \\
 &+ 0,4944815018 p_5 p_2 - 3,951044578 p_5 p_3 p_2 p_1 + 3,128536400 p_3 p_2 p_1 \\
 &+ 6,215920367 p_4 p_3 p_2 p_1 - 2,198227650 p_3 p_2 - 0,2069484004 p_5 p_4 p_3 \\
 &- 0,4061084714 p_5 p_1 p_4 + 4,179409895 p_4 p_3 p_2 + 1,562160649 p_5 p_1 p_2 \\
 &+ 0,06300378 p_5 p_3 - 0,2843519282 p_1 p_4 + 0,8789383114 p_4 p_3 + 1,022462973 p_5 p_4 \\
 &+ 1,327290709 p_5 p_3 p_1 + 0,091083445 p_5 p_1 + 8,000548513 p_5 p_4 p_3 p_2 p_1 \\
 &- 3,392952256 p_5 p_4 p_3 p_2 + 2,475255958
 \end{aligned}$$

Problem geneig

Dimenzija problema:6

Variable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$

$$f_1(\mathbf{x}) = -10 \cdot x_1 \cdot x_6^2 + 2 \cdot x_2 \cdot x_6^2 - x_3 \cdot x_6^2 + x_4 \cdot x_6^2 + 3 \cdot x_5 \cdot x_6^2 + x_1 \cdot x_6 + 2 \cdot x_2 \cdot x_6 + x_3 \cdot x_6 \\ + 2 \cdot x_4 \cdot x_6 + x_5 \cdot x_6 + 10 \cdot x_1 + 2 \cdot x_2 - x_3 + 2 \cdot x_4 - 2 \cdot x_5$$

$$f_2(\mathbf{x}) = 2 \cdot x_1 \cdot x_6^2 - 11 \cdot x_2 \cdot x_6^2 + 2 \cdot x_3 \cdot x_6^2 - 2 \cdot x_4 \cdot x_6^2 + x_5 \cdot x_6^2 + 2 \cdot x_1 \cdot x_6 + x_2 \cdot x_6 + 2 \cdot x_3 \cdot x_6 \\ + x_4 \cdot x_6 + 3 \cdot x_5 \cdot x_6 + 2 \cdot x_1 + 9 \cdot x_2 + 3 \cdot x_3 - x_4 - 2 \cdot x_5$$

$$f_3(\mathbf{x}) = -x_1 \cdot x_6^2 + 2 \cdot x_2 \cdot x_6^2 - 12 \cdot x_3 \cdot x_6^2 - x_4 \cdot x_6^2 + x_5 \cdot x_6^2 + x_1 \cdot x_6 + 2 \cdot x_2 \cdot x_6 - 2 \cdot x_4 \cdot x_6 \\ - 2 \cdot x_5 \cdot x_6 - x_1 + 3 \cdot x_2 + 10 \cdot x_3 + 2 \cdot x_4 - x_5$$

$$f_4(\mathbf{x}) = x_1 \cdot x_6^2 - 2 \cdot x_2 \cdot x_6^2 - x_3 \cdot x_6^2 - 10 \cdot x_4 \cdot x_6^2 + 2 \cdot x_5 \cdot x_6^2 + 2 \cdot x_1 \cdot x_6 + x_2 \cdot x_6 - 2 \cdot x_3 \cdot x_6 \\ + 2 \cdot x_4 \cdot x_6 + 3 \cdot x_5 \cdot x_6 + 2 \cdot x_1 - x_2 + 2 \cdot x_3 + 12 \cdot x_4 + x_5$$

$$f_5(\mathbf{x}) = 3 \cdot x_1 \cdot x_6^2 + x_2 \cdot x_6^2 + x_3 \cdot x_6^2 + 2 \cdot x_4 \cdot x_6^2 - 11 \cdot x_5 \cdot x_6^2 + x_1 \cdot x_6 + 3 \cdot x_2 \cdot x_6 - 2 \cdot x_3 \cdot x_6 \\ + 3 \cdot x_4 \cdot x_6 + 3 \cdot x_5 \cdot x_6 - 2 \cdot x_1 - 2 \cdot x_2 - x_3 + x_4 + 10 \cdot x_5$$

$$f_6(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + x_5 - 1$$

Problem graspa

Dimenzija problema:3

Variable: $\mathbf{x} = (x_1, x_2, x_3)$

$$f_1(\mathbf{x}) = x_3^3 - x_3 x_1 x_2$$

$$f_2(\mathbf{x}) = x_1^2 - x_3 x_2$$

$$f_3(\mathbf{x}) = 10x_3 x_2 + x_1 - x_3 - 0,1$$

Problem heart

Dimenzija problema:8

Variable: $\mathbf{x} = (a, b, c, d, t, u, v, w)$

$$f_1(\mathbf{x}) = a + b - 0,63254$$

$$f_2(\mathbf{x}) = c + d + 1,34534$$

$$f_3(\mathbf{x}) = ta + ub - vc - wd + 0,8365348$$

$$f_4(\mathbf{x}) = va + wb + tc + ud - 1,7345334$$

$$f_5(\mathbf{x}) = at^2 - av^2 - 2ctv + bu^2 - bw^2 - 2duw - 1,352352$$

$$f_6(\mathbf{x}) = ct^2 - cv^2 + 2atv + du^2 - dw^2 + 2buw + 0,843453$$

$$f_7(\mathbf{x}) = at^3 - 3atv^2 + cv^3 - 3cvt^2 + bu^3 - 3buw^2 + dw^3 - 3d wu^2 + 0,9563453$$

$$f_8(\mathbf{x}) = ct^3 - 3ctv^2 - av^3 + 3avt^2 + du^3 - 3duw^2 - bw^3 + 3bwu^2 - 1,2342523$$

Problem himmelbaum

Dimenzija problema:2

Varijable: $\mathbf{x} = (x_1, x_2)$

$$f_1(\mathbf{x}) = -42x + 2y^2 + 4xy + 4x^3 - 14$$

$$f_2(\mathbf{x}) = -26y + 2x^2 + 4xy + 4y^3 - 22$$

Problem ipp

Dimenzija problema:8

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$

$$f_1(\mathbf{x}) = x_1^2 + x_2^2 - 1$$

$$f_2(\mathbf{x}) = x_3^2 + x_4^2 - 1$$

$$f_3(\mathbf{x}) = x_5^2 + x_6^2 - 1$$

$$f_4(\mathbf{x}) = x_7^2 + x_8^2 - 1$$

$$\begin{aligned} f_5(\mathbf{x}) = & -2,4915068E-01 \cdot x_1 \cdot x_3 + 1,6091354 \cdot x_1 \cdot x_4 + 2,7942343E-01 \cdot x_2 \cdot x_3 \\ & + 1,4348016 \cdot x_2 \cdot x_4 + 4,0026384E-01 \cdot x_5 \cdot x_8 - 8,0052768E-01 \cdot x_6 \cdot x_7 \\ & + 7,4052388E-02 \cdot x_1 - 8,3050031E-02 \cdot x_2 - 3,8615961E-01 \cdot x_3 \\ & - 7,5526603E-01 \cdot x_4 + 5,0420168E-01 \cdot x_5 - 1,0916287 \cdot x_6 \\ & + 4,0026384E-01 \cdot x_8 + 4,9207290E-02 \end{aligned}$$

$$\begin{aligned} f_6(\mathbf{x}) = & 1,2501635E-01 \cdot x_1 \cdot x_3 - 6,8660736E-01 \cdot x_1 \cdot x_4 - 1,1922812E-01 \cdot x_2 \cdot x_3 \\ & - 7,1994047E-01 \cdot x_2 \cdot x_4 - 4,3241927E-01 \cdot x_5 \cdot x_7 - 8,6483855E-01 \cdot x_6 \cdot x_8 \\ & - 3,7157270E-02 \cdot x_1 + 3,5436896E-02 \cdot x_2 + 8,5383482E-02 \cdot x_3 \\ & - 3,9251967E-02 \cdot x_5 - 4,3241927E-01 \cdot x_7 + 1,3873010E-02 \end{aligned}$$

$$\begin{aligned} f_7(\mathbf{x}) = & -6,3555007E-01 \cdot x_1 \cdot x_3 - 1,1571992E-01 \cdot x_1 \cdot x_4 - 6,6640448E-01 \cdot x_2 \cdot x_3 \\ & + 1,1036211E-01 \cdot x_2 \cdot x_4 + 2,9070203E-01 \cdot x_5 \cdot x_7 + 1,2587767 \cdot x_5 \cdot x_8 \\ & - 6,2938836E-01 \cdot x_6 \cdot x_7 + 5,8140406E-01 \cdot x_6 \cdot x_8 + 1,9594662E-01 \cdot x_1 \\ & - 1,2280342 \cdot x_2 - 7,9034221E-02 \cdot x_4 + 2,6387877E-02 \cdot x_5 \\ & - 5,7131430E-02 \cdot x_6 - 1,1628081 \cdot x_7 + 1,2587767 \cdot x_8 + 2,1625750 \end{aligned}$$

$$\begin{aligned}
 f_8(\mathbf{x}) &= 1,4894773 \cdot x_1 \cdot x_3 + 2,3062341\text{E-}01 \cdot x_1 \cdot x_4 + 1,3281073 \cdot x_2 \cdot x_3 \\
 &- 2,5864503\text{E-}01 \cdot x_2 \cdot x_4 + 1,1651720 \cdot x_5 \cdot x_7 - 2,6908494\text{E-}01 \cdot x_5 \cdot x_8 \\
 &+ 5,3816987\text{E-}01 \cdot x_6 \cdot x_7 + 5,8258598\text{E-}01 \cdot x_6 \cdot x_8 - 2,0816985\text{E-}01 \cdot x_1 \\
 &+ 2,6868320 \cdot x_2 - 6,9910317\text{E-}01 \cdot x_3 + 3,5744413\text{E-}01 \cdot x_4 + 1,2499117 \cdot x_5 \\
 &+ 1,4677360 \cdot x_6 + 1,1651720 \cdot x_7 + 1,10763397 \cdot x_8 - 6,9686809\text{E-}01
 \end{aligned}$$

Problem katsura5

Dimenzija problema:6

Varijable: $\mathbf{x} = (x, y, z, t, u, v)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= 2 \cdot x^2 + 2 \cdot y^2 + 2 \cdot z^2 + 2 \cdot t^2 + 2 \cdot u^2 + v^2 - v \\
 f_2(\mathbf{x}) &= x \cdot y + y \cdot z + 2 \cdot z \cdot t + 2 \cdot t \cdot u + 2 \cdot u \cdot v - u \\
 f_3(\mathbf{x}) &= 2 \cdot x \cdot z + 2 \cdot y \cdot t + 2 \cdot z \cdot u + u^2 + 2 \cdot t \cdot v - t \\
 f_4(\mathbf{x}) &= 2 \cdot x \cdot t + 2 \cdot y \cdot u + 2 \cdot t \cdot u + 2 \cdot z \cdot v - z \\
 f_5(\mathbf{x}) &= t^2 + 2 \cdot x \cdot v + 2 \cdot y \cdot v + 2 \cdot z \cdot v - y \\
 f_6(\mathbf{x}) &= 2 \cdot x + 2 \cdot y + 2 \cdot z + 2 \cdot t + 2 \cdot u + v - 1
 \end{aligned}$$

Problem katsura6

Dimenzija problema:7

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= 1 \cdot x_1 + 2 \cdot x_2 + 2 \cdot x_3 + 2 \cdot x_4 + 2 \cdot x_5 + 2 \cdot x_6 + 2 \cdot x_7 - 1 \\
 f_2(\mathbf{x}) &= 2 \cdot x_4 \cdot x_3 + 2 \cdot x_5 \cdot x_2 + 2 \cdot x_6 \cdot x_1 + 2 \cdot x_7 \cdot x_2 - 1 \cdot x_6 \\
 f_3(\mathbf{x}) &= 1 \cdot x_3^2 + 2 \cdot x_4 \cdot x_2 + 2 \cdot x_5 \cdot x_1 + 2 \cdot x_6 \cdot x_2 + 2 \cdot x_7 \cdot x_3 - 1 \cdot x_5 \\
 f_4(\mathbf{x}) &= 2 \cdot x_3 \cdot x_2 + 2 \cdot x_4 \cdot x_1 + 2 \cdot x_5 \cdot x_2 + 2 \cdot x_6 \cdot x_3 + 2 \cdot x_7 \cdot x_4 - 1 \cdot x_4 \\
 f_5(\mathbf{x}) &= 1 \cdot x_2^2 + 2 \cdot x_3 \cdot x_1 + 2 \cdot x_4 \cdot x_2 + 2 \cdot x_5 \cdot x_3 + 2 \cdot x_6 \cdot x_4 + 2 \cdot x_7 \cdot x_5 - 1 \cdot x_3 \\
 f_6(\mathbf{x}) &= 2 \cdot x_2 \cdot x_1 + 2 \cdot x_3 \cdot x_2 + 2 \cdot x_4 \cdot x_3 + 2 \cdot x_5 \cdot x_4 + 2 \cdot x_6 \cdot x_5 + 2 \cdot x_7 \cdot x_6 - 1 \cdot x_2 \\
 f_7(\mathbf{x}) &= 1 \cdot x_1^2 + 2 \cdot x_2^2 + 2 \cdot x_3^2 + 2 \cdot x_4^2 + 2 \cdot x_5^2 + 2 \cdot x_6^2 + 2 \cdot x_7^2 - 1 \cdot x_1
 \end{aligned}$$

Problem katsura7

Dimenzija problema:8

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= -x_1 + 2 \cdot x_8^2 + 2 \cdot x_7^2 + 2 \cdot x_6^2 + 2 \cdot x_5^2 + 2 \cdot x_4^2 + 2 \cdot x_3^2 + 2 \cdot x_2^2 + x_1^2 \\
 f_2(\mathbf{x}) &= -x_2 + 2 \cdot x_8 \cdot x_7 + 2 \cdot x_7 \cdot x_6 + 2 \cdot x_6 \cdot x_5 + 2 \cdot x_5 \cdot x_4 + 2 \cdot x_4 \cdot x_3 + 2 \cdot x_3 \cdot x_2 + 2 \cdot x_2 \cdot x_1 \\
 f_3(\mathbf{x}) &= -x_3 + 2 \cdot x_8 \cdot x_6 + 2 \cdot x_7 \cdot x_5 + 2 \cdot x_6 \cdot x_4 + 2 \cdot x_5 \cdot x_3 + 2 \cdot x_4 \cdot x_2 + 2 \cdot x_3 \cdot x_1 + x_2^2
 \end{aligned}$$

$$f_4(\mathbf{x}) = -x_4 + 2 \cdot x_8 \cdot x_5 + 2 \cdot x_7 \cdot x_4 + 2 \cdot x_6 \cdot x_3 + 2 \cdot x_5 \cdot x_2 + 2 \cdot x_4 \cdot x_1 + 2 \cdot x_3 \cdot x_2$$

$$f_5(\mathbf{x}) = -x_5 + 2 \cdot x_8 \cdot x_4 + 2 \cdot x_7 \cdot x_3 + 2 \cdot x_6 \cdot x_2 + 2 \cdot x_5 \cdot x_1 + 2 \cdot x_4 \cdot x_2 + x_3^2$$

$$f_6(\mathbf{x}) = -x_6 + 2 \cdot x_8 \cdot x_3 + 2 \cdot x_7 \cdot x_2 + 2 \cdot x_6 \cdot x_1 + 2 \cdot x_5 \cdot x_2 + 2 \cdot x_4 \cdot x_3$$

$$f_7(\mathbf{x}) = -x_7 + 2 \cdot x_8 \cdot x_2 + 2 \cdot x_7 \cdot x_1 + 2 \cdot x_6 \cdot x_2 + 2 \cdot x_5 \cdot x_3 + x_4^2$$

$$f_8(\mathbf{x}) = -1 + 2 \cdot x_8 + 2 \cdot x_7 + 2 \cdot x_6 + 2 \cdot x_5 + 2 \cdot x_4 + 2 \cdot x_3 + 2 \cdot x_2 + x_1$$

Problem kearfoot16

Dimenzija problema:6

Variable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$

$$f_1(\mathbf{x}) = x_1^2 - 0,2x_1 + x_2 - 0,09$$

$$f_2(\mathbf{x}) = x_2^2 - 0,2x_2 + x_3 - 0,09$$

$$f_3(\mathbf{x}) = x_3^2 - 0,2x_3 + x_4 - 0,09$$

$$f_4(\mathbf{x}) = x_4^2 - 0,2x_4 + x_5 - 0,09$$

$$f_5(\mathbf{x}) = x_5^2 - 0,2x_5 + x_6 - 0,09$$

$$f_6(\mathbf{x}) = x_6^2 - 0,2x_6 + x_1 - 0,09$$

Problem kin1

Dimenzija problema:12

Variable: $\mathbf{x} = (s_1, c_1, s_2, c_2, s_3, c_3, s_4, c_4, s_5, c_5, s_6, c_6)$

$$f_1(\mathbf{x}) = s_1^2 + c_1^2 - 1$$

$$f_2(\mathbf{x}) = s_2^2 + c_2^2 - 1$$

$$f_3(\mathbf{x}) = s_3^2 + c_3^2 - 1$$

$$f_4(\mathbf{x}) = s_4^2 + c_4^2 - 1$$

$$f_5(\mathbf{x}) = s_5^2 + c_5^2 - 1$$

$$f_6(\mathbf{x}) = s_6^2 + c_6^2 - 1$$

$$f_7(\mathbf{x}) = s_2 c_5 s_6 - s_3 c_5 s_6 - s_4 c_5 s_6 + c_2 c_6 + c_3 c_6 + c_4 c_6 - 0,4077$$

$$f_8(\mathbf{x}) = c_1 c_2 s_5 + c_1 c_3 s_5 + c_1 c_4 s_5 + s_1 c_5 - 1,9115$$

$$f_9(\mathbf{x}) = s_2 s_5 + s_3 s_5 + s_4 s_5 - 1,9791$$

$$f_{10}(\mathbf{x}) = c_1 c_2 + c_1 c_3 + c_1 c_4 + c_1 c_2 + c_1 c_3 + c_1 c_2 - 4,0616$$

$$f_{11}(\mathbf{x}) = s_1 c_2 + s_1 c_3 + s_1 c_4 + s_1 c_2 + s_1 c_3 + s_1 c_2 - 1,7172$$

$$f_{12}(\mathbf{x}) = s_2 + s_3 + 9s_4 + s_2 + s_3 + s_2 - 3,9701$$

Problem kinema

Dimenzija problema:9

Varijable: $\mathbf{x} = (z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9)$

$$f_1(\mathbf{x}) = z_1^2 + z_2^2 + z_3^2 - 12 \cdot z_1 - 68$$

$$f_2(\mathbf{x}) = z_4^2 + z_5^2 + z_6^2 - 12 \cdot z_5 - 68$$

$$f_3(\mathbf{x}) = z_7^2 + z_8^2 + z_9^2 - 24 \cdot z_8 - 12 \cdot z_9 + 100$$

$$f_4(\mathbf{x}) = z_1 \cdot z_4 + z_2 \cdot z_5 + z_3 \cdot z_6 - 6 \cdot z_1 - 6 \cdot z_5 - 52$$

$$f_5(\mathbf{x}) = z_1 \cdot z_7 + z_2 \cdot z_8 + z_3 \cdot z_9 - 6 \cdot z_1 - 12 \cdot z_8 - 6 \cdot z_9 + 64$$

$$f_6(\mathbf{x}) = z_4 \cdot z_7 + z_5 \cdot z_8 + z_6 \cdot z_9 - 6 \cdot z_5 - 12 \cdot z_8 - 6 \cdot z_9 + 32$$

$$f_7(\mathbf{x}) = 2 \cdot z_2 + 2 \cdot z_3 - z_4 - z_5 - 2 \cdot z_6 - z_7 - z_9 + 18$$

$$f_8(\mathbf{x}) = z_1 + z_2 + 2 \cdot z_3 + 2 \cdot z_4 + 2 \cdot z_6 - 2 \cdot z_7 + z_8 - z_9 - 38$$

$$f_9(\mathbf{x}) = z_1 + z_3 - 2 \cdot z_4 + z_5 - z_6 + 2 \cdot z_7 - 2 \cdot z_8 + 8$$

Problem kotsireas

Dimenzija problema:6

Varijable: $\mathbf{x} = (b, d, B, D, F, f)$

$$f_1(\mathbf{x}) = b \cdot B - d \cdot B - b \cdot D + d \cdot D - 2 \cdot F + 2$$

$$f_2(\mathbf{x}) = b \cdot B - d \cdot B + b \cdot D - d \cdot D - 2 \cdot b \cdot F + 2 \cdot d \cdot F + 2 \cdot B - 2 \cdot D$$

$$f_3(\mathbf{x}) = b \cdot b - 2 \cdot b \cdot d + d \cdot d - 2 \cdot b - 2 \cdot d + f + 1$$

$$f_4(\mathbf{x}) = B^2 \cdot b^3 - 1$$

$$f_5(\mathbf{x}) = D^2 \cdot d^3 - 1$$

$$f_6(\mathbf{x}) = F^2 \cdot f^3 - 1$$

Problem ku10

Dimenzija problema:10

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$

$$f_1(\mathbf{x}) = 5 \cdot x_1 \cdot x_2 + 5 \cdot x_1 + 3 \cdot x_2 + 55$$

$$f_2(\mathbf{x}) = 7 \cdot x_2 \cdot x_3 + 9 \cdot x_2 + 9 \cdot x_3 + 19$$

$$f_3(\mathbf{x}) = 3 \cdot x_3 \cdot x_4 + 6 \cdot x_3 + 5 \cdot x_4 - 4$$

$$f_4(\mathbf{x}) = 6 \cdot x_4 \cdot x_5 + 6 \cdot x_4 + 7 \cdot x_5 + 118$$

$$f_5(\mathbf{x}) = x_5 \cdot x_6 + 3 \cdot x_5 + 9 \cdot x_6 + 27$$

$$f_6(\mathbf{x}) = 6 \cdot x_6 \cdot x_7 + 7 \cdot x_6 + x_7 + 72$$

$$f_7(\mathbf{x}) = 9 \cdot x_7 \cdot x_8 + 7 \cdot x_7 + x_8 + 35$$

$$f_8(\mathbf{x}) = 4 \cdot x_8 \cdot x_9 + 4 \cdot x_8 + 6 \cdot x_9 + 16$$

$$f_9(\mathbf{x}) = 8 \cdot x_9 \cdot x_{10} + 4 \cdot x_9 + 3 \cdot x_{10} - 51$$

$$f_{10}(\mathbf{x}) = 3 \cdot x_1 \cdot x_{10} - 6 \cdot x_1 + x_{10} + 5$$

Problem leary

Dimenzija problema:2

Varijable: $\mathbf{x} = (x, y)$

$$f_1(\mathbf{x}) = x^2 \cdot y^3 + x \cdot y - 2$$

$$f_2(\mathbf{x}) = 2 \cdot x \cdot y^2 + x^2 \cdot y + x \cdot y$$

Problem lorentz

Dimenzija problema:4

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4)$

$$f_1(\mathbf{x}) = x_1 \cdot x_2 - x_1 \cdot x_3 - x_4 + 1$$

$$f_2(\mathbf{x}) = x_2 \cdot x_3 - x_2 \cdot x_4 - x_1 + 1$$

$$f_3(\mathbf{x}) = -x_1 \cdot x_3 + x_3 \cdot x_4 - x_2 + 1$$

$$f_4(\mathbf{x}) = x_1 \cdot x_4 - x_2 \cdot x_4 - x_3 + 1$$

Problem mickey

Dimenzija problema:2

Varijable: $\mathbf{x} = (x, y)$

$$f_1(\mathbf{x}) = x^2 + 4 \cdot y^2 - 4$$

$$f_2(\mathbf{x}) = 2 \cdot y^2 - x$$

Problem mladenov

Dimenzija problema:10

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$

$$f_1(\mathbf{x}) = 2,5x_1^3 - 10,5x_1^2 + 11,8x_1 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 1$$

$$f_2(\mathbf{x}) = 2,5x_2^3 - 10,5x_2^2 + 11,8x_2 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 2$$

$$f_3(\mathbf{x}) = 2,5x_3^3 - 10,5x_3^2 + 11,8x_3 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 3$$

$$f_4(\mathbf{x}) = 2,5x_4^3 - 10,5x_4^2 + 11,8x_4 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 4$$

$$f_5(\mathbf{x}) = 2,5x_5^3 - 10,5x_5^2 + 11,8x_5 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 5$$

$$f_6(\mathbf{x}) = 2,5x_6^3 - 10,5x_6^2 + 11,8x_6 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 6$$

$$f_7(\mathbf{x}) = 2,5x_7^3 - 10,5x_7^2 + 11,8x_7 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 7$$

$$f_8(\mathbf{x}) = 2,5x_8^3 - 10,5x_8^2 + 11,8x_8 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8$$

$$f_9(\mathbf{x}) = 2,5x_9^3 - 10,5x_9^2 + 11,8x_9 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 9$$

$$f_{10}(\mathbf{x}) = 2,5x_{10}^3 - 10,5x_{10}^2 + 11,8x_{10} + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 10$$

Problem monfroy

Dimenzija problema:3

Varijable: $\mathbf{x} = (x, y, z)$

$$f_1(\mathbf{x}) = zx^2 - 0,5x - y^2$$

$$f_2(\mathbf{x}) = zy^2 + 2x + 0,5$$

$$f_3(\mathbf{x}) = z - 16x^2 - 4xy^2 - 1$$

Problem moreno

Dimenzija problema:3

Varijable: $\mathbf{x} = (x, y, z)$

$$f_1(\mathbf{x}) = 6y^2 + 20y + 2x + 44z - 170$$

$$f_2(\mathbf{x}) = 3y^3 - 43y - 7x - 6z + 100$$

$$f_3(\mathbf{x}) = z^3 - 79z + 6x^2 - 10y + 4$$

Problem morgan

Dimenzija problema:2

Varijable: $\mathbf{x} = (x, y)$

$$f_1(\mathbf{x}) = 15,05x^2y + 4,365xy^2 - 25,45xy + 2,007y^3 - 63,84y^2 + 488,3y - 2096$$

$$f_2(\mathbf{x}) = -4,398x^2y + 39,75x^2 - 1,039xy + 11,21x - 6,738y$$

Problem nash-equilibria

Dimenzija problema:8

Varijable: $\mathbf{x} = (a, b, c, d, e, f, g, h)$

$$f_1(\mathbf{x}) = 27bcf - 9bc - 9bf - 9cf + 3b + 3c + 3f - 1,1$$

$$f_2(\mathbf{x}) = 27adg - 9ad - 9ag - 9dg + 3a + 3d + 3g - 1,1$$

$$f_3(\mathbf{x}) = 75adh - 25ad - 15ah - 15dh + 5a + 5d + 3h - 1,1$$

$$f_4(\mathbf{x}) = 75bcj - 25bc - 15bj - 15cj + 5b + 5c + 3j - 1,1$$

$$f_5(\mathbf{x}) = 175agh - 35ag - 35ah - 25gh + 7a + 5g + 5h - 1,1$$

$$f_6(\mathbf{x}) = 175bfj - 35bf - 35bj - 25fj + 7b + 5f + 5j - 1,1$$

$$f_7(\mathbf{x}) = 343cfj - 49cf - 49cj - 49fj + 7c + 7f + 7j - 1,1$$

$$f_8(\mathbf{x}) = 343dgh - 49dg - 49dh - 49gh + 7d + 7g + 7h - 1,1$$

Problem noon3

Dimenzija problema:3

Varijable: $\mathbf{x} = (x_1, x_2, x_3)$

$$f_1(\mathbf{x}) = x_1 \cdot x_2^2 + x_1 \cdot x_3^2 - 1,1 \cdot x_1 + 1$$

$$f_2(\mathbf{x}) = x_2 \cdot x_1^2 + x_2 \cdot x_3^2 - 1,1 \cdot x_2 + 1$$

$$f_3(\mathbf{x}) = x_3 \cdot x_1^2 + x_3 \cdot x_2^2 - 1,1 \cdot x_3 + 1$$

Problem noon4

Dimenzija problema:4

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4)$

$$f_1(\mathbf{x}) = x_1 \cdot x_2^2 + x_1 \cdot x_3^2 + x_1 \cdot x_4^2 - 1,1 \cdot x_1 + 1$$

$$f_2(\mathbf{x}) = x_2 \cdot x_1^2 + x_2 \cdot x_3^2 + x_2 \cdot x_4^2 - 1,1 \cdot x_2 + 1$$

$$f_3(\mathbf{x}) = x_3 \cdot x_1^2 + x_3 \cdot x_2^2 + x_3 \cdot x_4^2 - 1,1 \cdot x_3 + 1$$

$$f_4(\mathbf{x}) = x_4 \cdot x_1^2 + x_4 \cdot x_2^2 + x_4 \cdot x_3^2 - 1,1 \cdot x_4 + 1$$

Problem noon5

Dimenzija problema:5

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$

$$f_1(\mathbf{x}) = x_1 \cdot x_2^2 + x_1 \cdot x_3^2 + x_1 \cdot x_4^2 + x_1 \cdot x_5^2 - 1,1 \cdot x_1 + 1$$

$$f_2(\mathbf{x}) = x_2 \cdot x_1^2 + x_2 \cdot x_3^2 + x_2 \cdot x_4^2 + x_2 \cdot x_5^2 - 1,1 \cdot x_2 + 1$$

$$f_3(\mathbf{x}) = x_3 \cdot x_1^2 + x_3 \cdot x_2^2 + x_3 \cdot x_4^2 + x_3 \cdot x_5^2 - 1,1 \cdot x_3 + 1$$

$$f_4(\mathbf{x}) = x_4 \cdot x_1^2 + x_4 \cdot x_2^2 + x_4 \cdot x_3^2 + x_4 \cdot x_5^2 - 1,1 \cdot x_4 + 1$$

$$f_5(\mathbf{x}) = x_5 \cdot x_1^2 + x_5 \cdot x_2^2 + x_5 \cdot x_3^2 + x_5 \cdot x_4^2 - 1,1 \cdot x_5 + 1$$

Problem pb601

Dimenzija problema:3

Varijable: $\mathbf{x} = (x_1, x_2, x_3)$

$$f_1(\mathbf{x}) = 5,3835466340508E+08x_2^6 + 5,03135199444E+08x_2^5 + 8,95258362E+07x_2^4 - 6,25E+13x_1^2x_3 + 5,77586E+06x_2^3 + 107358x_2^2 + 617x_2 + 1$$

$$f_2(\mathbf{x}) = -5,03135199444E+08x_2^5 - 1,790516724E+08x_2^4 + 6,25E+13x_1^2x_2 + 1,875E+14x_1^2x_3 - 1,73275812E+07x_2^3 + 20250000x_1x_2 - 429432x_2^2 - 3085x_2 - 6$$

$$f_3(\mathbf{x}) = -5,555555555555555E+15x_1^2 + 1,111111111111111E+16x_1x_3 + 1800000000x_2 + 1$$

Problem pb601vs

Dimenzija problema:3

Varijable: $\mathbf{x} = (x_1, x_2, x_3)$

$$f_1(\mathbf{x}) = 1,25623491196412E-02x_2^6 + 5,62718762857620E-01x_2^5 + 4,79909553837465x_2^4 + 1,48399354804415E+01x_2^3 - 1,45832692729029E-01x_1^2x_3 + 1,32206878780498E+01x_2^2 + 3,64174153281780x_2 + 2,82896953297879E-01$$

$$f_2(\mathbf{x}) = -1,00037203701072E-01x_2^5 - 1,70631629738197x_2^4 - 7,91449875082614x_2^3 + 5,38347462575619x_2x_1^2 + 7,77761241586725E-02x_1^2x_3 - 9,40121946251430x_2^2 + 1,92514482107687E-01x_2x_1 - 3,23704898069416x_2 - 3,01751660098453E-01$$

$$f_3(\mathbf{x}) = -3,97516381945663x_1^2 + 8,81655270550827E+01x_1x_3 + 3,27346103274571E+02x_2 + 8,71643679865540E-06$$

Problem proddeco

Dimenzija problema:4

Varijable: $\mathbf{x} = (z_1, z_2, z_3, z_4)$

$$f_1(\mathbf{x}) = 64,632945216 \cdot z_1^2 \cdot z_4^2 + 55,661869232 \cdot z_1^2 \cdot z_4 - 129,265890432 \cdot z_1 \cdot z_2 \cdot z_3 \cdot z_4 - 55,661869232 \cdot z_1 \cdot z_2 \cdot z_3 + 12,201006656 \cdot z_1 \cdot z_2 \cdot z_4 + 48,21149907 \cdot z_1 \cdot z_3 \cdot z_4 + 50,7965557025 \cdot z_1 \cdot z_3 + 71,3989540236 \cdot z_1 \cdot z_4^2 + 62,3536335897 \cdot z_1 \cdot z_4 + 64,632945216 \cdot z_2^2 \cdot z_3^2 - 12,201006656 \cdot z_2^2 \cdot z_3 - 48,21149907 \cdot z_2 \cdot z_3^2 - 71,3989540236 \cdot z_2 \cdot z_3 \cdot z_4 + 9,14136787 \cdot z_2 \cdot z_3 + 17,5403018476 \cdot z_2 \cdot z_4$$

$$\begin{aligned}
f_2(\mathbf{x}) &= 81,3704061375 \cdot z_1^2 \cdot z_4^2 + 55,3637563425 \cdot z_1^2 \cdot z_4 - 162,740812275 \cdot z_1 \cdot z_2 \cdot z_3 \cdot z_4 \\
&- 55,3637563425 \cdot z_1 \cdot z_2 \cdot z_3 + 25,032598875 \cdot z_1 \cdot z_2 \cdot z_4 + 21,781944477 \cdot z_1 \cdot z_3 \cdot z_4 \\
&+ 24,7967262078 \cdot z_1 \cdot z_3 + 55,073652285 \cdot z_1 \cdot z_4^2 + 41,688210899 \cdot z_1 \cdot z_4 \\
&+ 81,3704061375 \cdot z_2^2 \cdot z_3^2 - 25,032598875 \cdot z_2^2 \cdot z_3 - 21,781944477 \cdot z_2 \cdot z_3^2 \\
&+ 55,073652285 \cdot z_2 \cdot z_3 \cdot z_4 + 9,56199577 \cdot z_2 \cdot z_3 + 20,93013785 \cdot z_2 \cdot z_4 \\
f_3(\mathbf{x}) &= 35,2817031945 \cdot z_1^2 \cdot z_4^2 + 42,2584220358 \cdot z_1^2 \cdot z_4 - 70,563406389 \cdot z_1 \cdot z_2 \cdot z_3 \cdot z_4 \\
&- 42,2584220358 \cdot z_1 \cdot z_2 \cdot z_3 + 23,8847040258 \cdot z_1 \cdot z_2 \cdot z_4 + 8,0797827915 \cdot z_1 \cdot z_3 \cdot z_4 \\
&+ 13,5030253426 \cdot z_1 \cdot z_3 + 23,1780987025 \cdot z_1 \cdot z_4^2 + 37,033791391 \cdot z_1 \cdot z_4 \\
&+ 35,2817031945 \cdot z_2^2 \cdot z_3^2 - 23,8847040258 \cdot z_2^2 \cdot z_3 - 8,0797827915 \cdot z_2 \cdot z_3^2 \\
&- 23,1780987025 \cdot z_2 \cdot z_3 \cdot z_4 + 10,3467518726 \cdot z_2 \cdot z_3 + 16,946539941 \cdot z_2 \cdot z_4 \\
f_4(\mathbf{x}) &= 6,8993630265 \cdot z_1^2 \cdot z_4^2 + 43,1035038882 \cdot z_1^2 \cdot z_4 - 13,798726053 \cdot z_1 \cdot z_2 \cdot z_3 \cdot z_4 \quad (1) \\
&- 43,1035038882 \cdot z_1 \cdot z_2 \cdot z_3 + 56,0575143198 \cdot z_1 \cdot z_2 \cdot z_4 + 2,379925759 \cdot z_1 \cdot z_3 \cdot z_4 \\
&+ 16,2879242092 \cdot z_1 \cdot z_3 + 6,40051015 \cdot z_1 \cdot z_4^2 + 42,97692982 \cdot z_1 \cdot z_4 \\
&+ 6,8993630265 \cdot z_2^2 \cdot z_3^2 - 56,0575143198 \cdot z_2^2 \cdot z_3 - 2,379925759 \cdot z_2 \cdot z_3^2 \\
&- 6,40051015 \cdot z_2 \cdot z_3 \cdot z_4 + 22,5597322388 \cdot z_2 \cdot z_3 + 55,41365098 \cdot z_2 \cdot z_4
\end{aligned}$$

Problem puma

Dimenzija problema:8

Variable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$

$$\begin{aligned}
f_1(\mathbf{x}) &= x_1^2 + x_2^2 - 1 \\
f_2(\mathbf{x}) &= x_3^2 + x_4^2 - 1 \\
f_3(\mathbf{x}) &= x_5^2 + x_6^2 - 1 \\
f_4(\mathbf{x}) &= x_7^2 + x_8^2 - 1 \\
f_5(\mathbf{x}) &= 0,004731 \cdot x_1 \cdot x_3 - 0,3578 \cdot x_2 \cdot x_3 - 0,1238 \cdot x_1 - 0,001637 \cdot x_2 - 0,9338 \cdot x_4 + x_7 - 0,3571 \\
f_6(\mathbf{x}) &= 0,2238 \cdot x_1 \cdot x_3 + 0,7623 \cdot x_2 \cdot x_3 + 0,2638 \cdot x_1 - 0,07745 \cdot x_2 - 0,6734 \cdot x_4 - 0,6022 \\
f_7(\mathbf{x}) &= x_6 \cdot x_8 + 0,3578 \cdot x_1 + 0,004731 \cdot x_2 \\
f_8(\mathbf{x}) &= -0,7623 \cdot x_1 + 0,2238 \cdot x_2 + 0,3461
\end{aligned}$$

Problem quadfor2

Dimenzija problema:4

Variable: $\mathbf{x} = (w_1, w_2, x_1, x_2)$

$$f_1(\mathbf{x}) = w_1 + w_2 - 1$$

$$f_2(\mathbf{x}) = w_1 \cdot x_1 + w_2 \cdot x_2$$

$$f_3(\mathbf{x}) = w_1 \cdot x_1^2 + w_2 \cdot x_2^2 - \frac{2}{3}$$

$$f_4(\mathbf{x}) = w_1 \cdot x_1^3 + w_2 \cdot x_2^3$$

Problem quadgrid

Dimenzija problema:5

Variable: $\mathbf{x} = (w_0, w_1, w_2, w_3, b)$

$$f_1(\mathbf{x}) = w_0 + w_1 + w_2 + w_3 - 1$$

$$f_2(\mathbf{x}) = w_0 b + w_1 b + w_2 b + w_3 b + 0,5 w_1 + w_2 + 1,5 w_3 - 0,63397459621556$$

$$f_3(\mathbf{x}) = w_0 b^2 + w_1 b^2 + w_2 b^2 + w_3 b^2 + w_1 b + 2 w_2 b + 3 w_3 b + 0,25 w_1 + w_2 + 4,5 w_3 - 0,40192378864668$$

$$f_4(\mathbf{x}) = w_0 b^3 + w_1 b^3 + w_2 b^3 + w_3 b^3 + 1,5 w_1 b^2 + 3 w_2 b^2 + 4,5 w_3 b^2 + 0,75 w_1 b + 3 w_2 b + 6,75 w_3 b + 0,125 w_1 + w_2 + 3,375 w_3 - 0,13109155679036$$

$$f_5(\mathbf{x}) = w_0 b^4 + w_1 b^4 + w_2 b^4 + w_3 b^4 + 2 w_1 b^3 + 4 w_2 b^3 + 6 w_3 b^3 + 1,5 w_1 b^2 + 6 w_2 b^2 + 13,5 w_3 b^2 + 0,5 w_1 b + 4 w_2 b + 13,5 w_3 b + 0,0625 w_1 + w_2 + 5,0625 w_3 + 0,30219332850656$$

Problem rabmo

Dimenzija problema:9

Variable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$

$$f_1(\mathbf{x}) = x_1 + x_3 + x_5 + 2x_7 - 1$$

$$f_2(\mathbf{x}) = x_1 x_2 + x_3 x_4 + 2x_5 x_6 + 2x_7 x_8 + 2x_7 x_9 - \frac{2}{3}$$

$$f_3(\mathbf{x}) = x_1 x_2^2 + x_3 x_4^2 + 2x_5 x_6^2 + 2x_7 x_8^2 + 2x_7 x_9^2 - 0,4$$

$$f_4(\mathbf{x}) = x_1 x_2^3 + x_3 x_4^3 + 2x_5 x_6^3 + 2x_7 x_8^3 + 2x_7 x_9^3 - \frac{2}{7}$$

$$f_5(\mathbf{x}) = x_1 x_2^4 + x_3 x_4^4 + 2x_5 x_6^4 + 2x_7 x_8^4 + 2x_7 x_9^4 - \frac{2}{9}$$

$$f_6(\mathbf{x}) = x_5 x_6^2 + 2x_7 x_8 x_9 - \frac{1}{9}$$

$$f_7(\mathbf{x}) = x_5 x_6^4 + 2x_7 x_8^2 x_9^2 - 0,04$$

$$f_8(\mathbf{x}) = x_5 x_6^3 + x_7 x_8 x_9^2 + x_7 x_8^2 x_9 - \frac{1}{15}$$

$$f_9(\mathbf{x}) = x_5 x_6^4 + x_7 x_8 x_9^3 + x_7 x_8^3 x_9 - \frac{1}{21}$$

Problem rbp1

Dimenzija problema:6

Varijable: $\mathbf{x} = (a, b, c, u, v, w)$

$$f_1(\mathbf{x}) = u^2 + v^2 + w^2 - 1$$

$$f_2(\mathbf{x}) = -12 \cdot b - 4 \cdot a + 4 \cdot c + 4 \cdot a \cdot b + 4 \cdot a \cdot c + 4 \cdot b \cdot c + a^2 + 5 \cdot b^2 + 13 \cdot c^2 + 8 \cdot w + 2 \cdot u + 9 \\ + 8 \cdot a \cdot c \cdot u + 8 \cdot a \cdot c \cdot w + 8 \cdot b \cdot a \cdot v + 8 \cdot b \cdot c \cdot v + 2 \cdot u \cdot a^2 - 6 \cdot u \cdot b^2 - 6 \cdot u \cdot c^2 + 2 \cdot v \cdot a^2 \\ + 2 \cdot v \cdot b^2 + 2 \cdot v \cdot c^2 + 8 \cdot w \cdot c^2 + 8 \cdot b \cdot u + 8 \cdot c \cdot v - 8 \cdot a \cdot v - 8 \cdot b \cdot w + 2 \cdot v$$

$$f_3(\mathbf{x}) = 8,86 - 6 \cdot b + 2 \cdot a + 4 \cdot a \cdot b \cdot u + 6 \cdot c - 6 \cdot a \cdot b - 6 \cdot a \cdot c + 6 \cdot b \cdot c + 2,86 \cdot a^2 + 4,86 \cdot b^2 \\ + 6,86 \cdot c^2 + 4 \cdot w - 3 \cdot u + 4 \cdot b \cdot c \cdot w + 4 \cdot a \cdot w - 4 \cdot c \cdot u + 4 \cdot a \cdot c \cdot u + 4 \cdot b \cdot c \cdot v - 3 \cdot u \cdot a^2 \\ - 3 \cdot u \cdot b^2 - 3 \cdot u \cdot c^2 - v \cdot a^2 + 3 \cdot v \cdot b^2 - v \cdot c^2 + 4 \cdot w \cdot c^2 + 4 \cdot b \cdot u - 4 \cdot a \cdot v + 3 \cdot v$$

$$f_4(\mathbf{x}) = -2 \cdot b + 2 \cdot a + 2 \cdot c + 2 \cdot a \cdot b - 10 \cdot a \cdot c - 2 \cdot b \cdot c + 3,5 \cdot a^2 - 2,5 \cdot b^2 + 1,5 \cdot c^2 + 4 \cdot w \\ - 5 \cdot u + 7,5 + 4 \cdot a \cdot c \cdot u - 4 \cdot a \cdot c \cdot w - 4 \cdot b \cdot a \cdot v + 4 \cdot b \cdot c \cdot v - 5 \cdot u \cdot a^2 - u \cdot b^2 - u \cdot c^2 \\ - v \cdot a^2 - v \cdot b^2 - v \cdot c^2 + 4 \cdot w \cdot c^2 + 4 \cdot b \cdot u - 4 \cdot c \cdot v - 4 \cdot a \cdot v + 4 \cdot b \cdot w - v$$

$$f_5(\mathbf{x}) = -4 \cdot a - 4 \cdot a \cdot b \cdot u - \frac{4}{3} \cdot c + \frac{4}{3} \cdot a \cdot b - 4 \cdot b \cdot c - \frac{59}{36} \cdot a^2 + \frac{13}{36} \cdot b^2 - \frac{59}{36} \cdot c^2 + 2 \cdot w - \frac{2}{3} \cdot u \\ - 4 \cdot b \cdot c \cdot w - 4 \cdot a \cdot w + \frac{13}{36} + 2 \cdot w \cdot b^2 + 2 \cdot w \cdot a^2 + 4 \cdot c \cdot u - \frac{2}{3} \cdot u \cdot a^2 - \frac{2}{3} \cdot u \cdot b^2 - \frac{2}{3} \cdot u \cdot c^2 \\ + v \cdot a^2 - 3 \cdot v \cdot b^2 + v \cdot c^2 + 2 \cdot w \cdot c^2 - 3 \cdot v$$

$$f_6(\mathbf{x}) = -8 \cdot b + 4 \cdot a \cdot b \cdot u + 8 \cdot c - 8 \cdot a \cdot b - 8 \cdot a \cdot c + 8 \cdot b^2 + 8 \cdot c^2 + 2 \cdot w - 2 \cdot u + 4 \cdot b \cdot c \cdot w \\ + 4 \cdot a \cdot w - 2 \cdot w \cdot b^2 - 2 \cdot w \cdot a^2 - 4 \cdot c \cdot u + 4 \cdot a \cdot c \cdot u + 4 \cdot a \cdot c \cdot w + 4 \cdot b \cdot a \cdot v + 4 \cdot b \cdot c \cdot v \\ - 2 \cdot u \cdot a^2 - 6 \cdot u \cdot b^2 - 6 \cdot u \cdot c^2 - 2 \cdot v \cdot a^2 + 2 \cdot v \cdot b^2 - 2 \cdot v \cdot c^2 + 2 \cdot w \cdot c^2 + 4 \cdot b \cdot u + 4 \cdot c \cdot v \\ - 4 \cdot a \cdot v - 4 \cdot b \cdot w + 2 \cdot v$$

Problem rbp124

Dimenzija problema:9

Varijable: $\mathbf{x} = (x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3)$

$$f_1(\mathbf{x}) = 62500 \cdot x_1^2 + 62500 \cdot y_1^2 + 62500 \cdot z_1^2 - 74529$$

$$f_2(\mathbf{x}) = 625 \cdot x_2^2 + 625 \cdot y_2^2 + 625 \cdot z_2^2 - 1250 \cdot x_2 - 2624$$

$$f_3(\mathbf{x}) = 12500 \cdot x_3^2 + 12500 \cdot y_3^2 + 12500 \cdot z_3^2 + 2500 \cdot x_3 - 44975 \cdot y_3 - 10982$$

$$f_4(\mathbf{x}) = 400000 \cdot x_1 \cdot x_2 + 400000 \cdot y_1 \cdot y_2 + 400000 \cdot z_1 \cdot z_2 - 400000 \cdot x_2 + 178837$$

$$f_5(\mathbf{x}) = 1000000 \cdot x_1 \cdot x_3 + 1000000 \cdot y_1 \cdot y_3 + 1000000 \cdot z_1 \cdot z_3 + 100000 \cdot x_3 - 1799000 \cdot y_3 - 805427$$

$$f_6(\mathbf{x}) = 3200 \cdot x_2 + 1271$$

$$f_7(\mathbf{x}) = 2000000 \cdot x_2 \cdot x_3 + 2000000 \cdot y_2 \cdot y_3 + 2000000 \cdot z_2 \cdot z_3 - 2000000 \cdot x_2 + 200000 \cdot x_3 \\ - 3598000 \cdot y_3 - 1403$$

$$f_8(\mathbf{x}) = 11380000000000 \cdot x_3 \cdot y_2 \cdot z_1 - 11380000000000 \cdot x_2 \cdot y_3 \cdot z_1 - 11380000000000 \cdot x_3 \cdot y_1 \cdot z_2 \\ + 11380000000000 \cdot x_1 \cdot y_3 \cdot z_2 + 11380000000000 \cdot x_2 \cdot y_1 \cdot z_3 - 11380000000000 \cdot x_1 \cdot y_2 \cdot z_3 \\ - 20688840000000 \cdot x_2 \cdot y_1 + 20688840000000 \cdot x_3 \cdot y_1 + 20688840000000 \cdot x_1 \cdot y_2 \\ - 20688840000000 \cdot x_3 \cdot y_2 - 20688840000000 \cdot x_1 \cdot y_3 + 20688840000000 \cdot x_2 \cdot y_3 \\ - 2014260000000 \cdot x_2 \cdot z_1 + 2014260000000 \cdot x_3 \cdot z_1 - 6190720000000 \cdot y_2 \cdot z_1 \\ + 6190720000000 \cdot y_3 \cdot z_1 + 2014260000000 \cdot x_1 \cdot z_2 - 2014260000000 \cdot x_3 \cdot z_2 \\ + 6190720000000 \cdot y_1 \cdot z_2 - 6190720000000 \cdot y_3 \cdot z_2 - 2014260000000 \cdot x_1 \cdot z_3 \\ + 2014260000000 \cdot x_2 \cdot z_3 - 6190720000000 \cdot y_1 \cdot z_3 + 6190720000000 \cdot y_2 \cdot z_3 \\ - 362960716800000 \cdot x_1 + 38025201600000 \cdot x_2 + 292548849600000 \cdot x_3 \\ + 11809567440000 \cdot y_1 + 1475978220000 \cdot y_2 - 825269402280000 \cdot y_3 \\ - 1212982689600000 \cdot z_1 - 151600474800000 \cdot z_2 \\ + 825859951200000 \cdot z_3 - 19295432410527$$

$$f_9(\mathbf{x}) = -777600000000 \cdot x_3 \cdot y_2 \cdot z_1 + 777600000000 \cdot x_2 \cdot y_3 \cdot z_1 + 777600000000 \cdot x_3 \cdot y_1 \cdot z_2 \\ - 777600000000 \cdot x_1 \cdot y_3 \cdot z_2 - 777600000000 \cdot x_2 \cdot y_1 \cdot z_3 + 777600000000 \cdot x_1 \cdot y_2 \cdot z_3 \\ - 1409011200000 \cdot x_2 \cdot y_1 + 1409011200000 \cdot x_3 \cdot y_1 + 1409011200000 \cdot x_1 \cdot y_2 \\ - 1409011200000 \cdot x_3 \cdot y_2 - 1409011200000 \cdot x_1 \cdot y_3 + 1409011200000 \cdot x_2 \cdot y_3 \\ - 1065312000000 \cdot x_2 \cdot z_1 + 1065312000000 \cdot x_3 \cdot z_1 - 805593600000 \cdot y_2 \cdot z_1 \\ + 805593600000 \cdot y_3 \cdot z_1 + 1065312000000 \cdot x_1 \cdot z_2 - 1065312000000 \cdot x_3 \cdot z_2 \\ + 805593600000 \cdot y_1 \cdot z_2 - 805593600000 \cdot y_3 \cdot z_2 - 1065312000000 \cdot x_1 \cdot z_3 \\ + 1065312000000 \cdot x_2 \cdot z_3 - 805593600000 \cdot y_1 \cdot z_3 + 805593600000 \cdot y_2 \cdot z_3 \\ + 235685027200 \cdot x_1 + 398417510400 \cdot x_2 + 158626915200 \cdot x_3 \\ - 311668424000 \cdot y_1 - 268090368000 \cdot y_2 + 72704002800 \cdot y_3 \\ + 412221302400 \cdot z_1 + 354583756800 \cdot z_2 + 307085438400 \cdot z_3 + 282499646407$$

Problem rbpl24es

Dimenzija problema:9

Varijable: $\mathbf{x} = (x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3)$

$$f_1(\mathbf{x}) = 9,54092867583359\text{E-}01 x_1^2 + 9,54092867583359\text{E-}01 y_1^2 \\ + 9,54092867583359\text{E-}01 z_1^2 - 1,13772139724992$$

$$f_2(\mathbf{x}) = 0,543572795268742 x_2^2 + 0,543572795268742 y_2^2 + 0,543572795268742 z_2^2 \\ - 1,08714559053748 x_2 - 2,28213602365629$$

$$f_3(\mathbf{x}) = 7,81600091707744\text{E-}01 x_3^2 + 7,81600091707744\text{E-}01 y_3^2 + 7,81600091707744\text{E-}01 z_3^2 \\ + 1,56320018341549\text{E-}01 x_3 - 2,81219712996446 y_3 - 6,86682576570756\text{E-}01$$

$$f_4(\mathbf{x}) = 1,12433011006630 x_1 x_2 + 1,12433011006630 y_1 y_2 + 1,12433011006630 z_1 z_2 \\ - 1,12433011006630 x_2 + 5,02679559734816\text{E-}01$$

$$f_5(\mathbf{x}) = 1,05181466955401 x_1 x_3 + 1,05181466955401 y_1 y_3 + 1,05181466955401 z_1 z_3 \\ + 1,05181466955401\text{E-}01 x_3 - 1,89221459052767 y_3 - 8,47159933854882\text{E-}01$$

$$f_6(\mathbf{x}) = 1,18650070685780 x_2 x_3 + 1,18650070685780 y_2 y_3 + 1,18650070685780 z_2 z_3 \\ - 1,18650070685780 x_2 + 1,18650070685780\text{E-}01 x_3 \\ - 2,13451477163718 y_3 - 8,32330245860744\text{E-}04$$

$$f_7(\mathbf{x}) = -0,639591378383803 x_1 y_2 z_3 + 0,639591378383803 x_1 z_2 y_3 + 0,639591378383803 y_1 x_2 z_3 \\ - 0,639591378383803 y_1 z_2 x_3 - 0,639591378383803 z_1 x_2 y_3 + 0,639591378383803 z_1 y_2 x_3 \\ + 1,16277712590175 x_1 y_2 + 0,0113207673973933 x_1 z_2 - 1,16277712590175 x_1 y_3 \\ - 0,0113207673973933 x_1 z_3 - 1,16277712590175 y_1 x_2 + 0,347937709840789 y_1 z_2 \\ + 1,16277712590175 y_1 x_3 - 0,347937709840789 y_1 z_3 - 0,113207673973933 z_1 x_2 \\ - 0,347937709840789 z_1 y_2 + 0,011320767397393 z_1 x_3 + 0,347937709840789 z_1 y_3 \\ + 1,16277712590175 x_2 y_3 + 0,011320767397393 x_2 z_3 - 1,16277712590175 y_2 x_3 \\ + 0,347937709840789 y_2 z_3 - 0,011320767397393 z_2 x_3 - 0,347937709840789 z_2 y_3 \\ - 2,03995206640848 x_1 + 0,0663734403960113 y_1 - 6,81733981016658 z_1 \\ + 0,213713454346801 x_2 + 8,29545645161926\text{E-}03 y_2 - 0,852041798251063 z_2 \\ + 1,64421548295483 x_3 - 4,63827060230442 y_3 + 4,64158967082591 z_3 \\ - 0,108446328751850$$

$$\begin{aligned}
 f_8(\mathbf{x}) = & 0,973909033873207 x_1 y_2 z_3 - 0,973909033873207 x_1 z_2 y_3 - 0,973909033873207 y_1 x_2 z_3 \\
 & - 0,973909033873207 y_1 z_2 x_3 + 0,973909033873207 z_1 x_2 y_3 - 0,973909033873207 z_1 y_2 x_3 \\
 & + 1,76472316937825 x_1 y_2 + 1,33425537640629 x_1 z_2 - 1,76472316937825 x_1 y_3 \\
 & - 1,33425537640629 x_1 z_3 - 1,76472316937825 y_1 x_2 + 1,00896975909264 y_1 z_2 \\
 & + 1,76472316937825 y_1 x_3 - 1,00896975909264 y_1 z_3 - 1,33425537640629 z_1 x_2 \\
 & - 1,00896975909264 z_1 y_2 + 1,33425537640629 z_1 x_3 + 1,00896975909264 z_1 y_3 \\
 & + 1,76472316937825 x_2 y_3 + 1,33425537640629 x_2 z_3 - 1,76472316937825 y_2 x_3 \\
 & + 1,00896975909264 y_2 z_3 - 1,33425537640629 z_2 x_3 - 1,00896975909264 z_2 y_3 \\
 & + 0,295184898583761 x_1 - 0,390350686351112 y_1 + 0,516288645013296 z_1 \\
 & + 0,499000016244640 x_2 - 0,335771130773653 y_2 + 0,444100210921065 z_2 \\
 & + 0,198673078354834 x_3 + 0,0910584942459657 y_3 \\
 & + 0,384610702968974E-01 z_3 + 0,353818104040334 \\
 f_9(\mathbf{x}) = & 1,43144710355625 x_2 + 0,568552896443749
 \end{aligned}$$

Problem rbp124s

Dimenzija problema:9

Varijable: $\mathbf{x} = (x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3)$

$$\begin{aligned}
 f_1(\mathbf{x}) = & 0,902353177111246 x_1^2 + 0,506753420513951 y_1^2 \\
 & + 1,15409618294195 z_1^2 - 1,89489327629418 \\
 f_2(\mathbf{x}) = & 0,547653292007868 x_2^2 + 0,401593017487710 y_2^2 + 0,914600572619008 z_2^2 \\
 & - 1,37399826809834 x_2 - 3,6181826772427 \\
 f_3(\mathbf{x}) = & 1,26480638249588 x_3^2 + 0,299953816740793 y_3^2 + 1,20375754734237 z_3^2 \\
 & + 0,346367554661357 x_3 - 3,03447482544876 y_3 - 2,08334787313530 \\
 f_4(\mathbf{x}) = & 1,01542688946772 x_1 x_2 + 0,651626823874782 y_1 y_2 + 1,48403542964494 z_1 z_2 \\
 & - 1,34750418563480 x_2 + 0,755749719256342 \\
 f_5(\mathbf{x}) = & 1,49129677811074 x_1 x_3 + 0,544238574067073 y_1 y_3 + 1,64533483741959 z_1 z_3 \\
 & + 0,197899885395127 x_3 - 1,73377157331563 y_3 - 2,18250318558661 \\
 f_6(\mathbf{x}) = & 3,63356727880623 x_2 x_3 + 1,51526503944521 y_2 y_3 + 4,58092916621510 z_2 z_3 \\
 & - 4,97526669091479 x_2 + 0,455809836346864 x_3 \\
 & - 3,99328243934042 y_3 - 4,37818922423922E-03
 \end{aligned}$$

$$\begin{aligned}
 f_7(\mathbf{x}) = & -1,50354584074328x_1y_2z_3 + 1,13265335620253x_1z_2y_3 + 1,31578980431683y_1x_2z_3 \\
 & - 1,74297802916645y_1z_2x_3 - 0,991212703684361E-01z_1x_2y_3 + 1,74297802916645z_1y_2x_3 \\
 & + 3,83650902835643x_1y_2 + 0,0563687325659283x_1z_2 - 1,99884215008463x_1y_3 \\
 & - 0,0389852390643551x_1z_3 - 3,35742305082321y_1x_2 + 1,29829675554826y_1z_2 \\
 & + 3,07590838122797y_1x_3 - 0,897916399527476y_1z_3 - 0,0493296589852189z_1x_2 \\
 & - 1,29829675554826z_1y_2 + 0,0451934442633164z_1x_3 + 0,676419697992925z_1y_3 \\
 & + 2,11450869353619x_2y_3 + 0,0412411890141876x_2z_3 - 3,71821525452185y_2x_3 \\
 & + 1,08541804248264y_2z_3 - 0,0546306758971674z_2x_3 - 0,817668710448484z_2y_3 \\
 & - 9,85984231145044x_1 + 0,240411038512531y_1 - 37,2647039170064z_1 \\
 & + 1,09272995041099x_2 + 0,0363213093186832y_2 - 5,62995295895362z_2 \\
 & + 7,70206259423188x_3 - 10,5808258067104y_3 \\
 & + 21,2115567191798z_3 - 0,695578779258571
 \end{aligned}$$

$$\begin{aligned}
 f_8(\mathbf{x}) = & 0,846362685065152x_1y_2z_3 - 0,637583178261945x_1z_2y_3 - 0,740672722829929y_1x_2z_3 \\
 & + 0,981141728306483y_1z_2x_3 + 0,557964661021749z_1x_2y_3 - 0,981141728306483z_1y_2x_3 \\
 & + 2,15248618665415x_1y_2 + 2,45598327418739x_1z_2 - 1,12145705524443x_1y_3 \\
 & - 1,69858520360145x_1z_3 - 1,88369340101545y_1x_2 + 1,39179304392241y_1z_2 \\
 & + 1,72574865667482y_1x_3 - 0,962579466940477y_1z_3 - 2,14929113844038z_1x_2 \\
 & - 1,39179304392241z_1y_2 + 1,96907643938610z_1x_3 + 0,725131774700529z_1y_3 \\
 & + 1,18635215524219x_2y_3 + 1,79687684671605x_2z_3 - 2,08611707028712y_2x_3 \\
 & + 1,16358396092368y_2z_3 - 2,38025621924476z_2x_3 - 0,876552774681028z_2y_3 \\
 & + 0,527434073456527x_1 - 0,522683827930675y_1 + 1,04327631315449z_1 \\
 & + 0,943203889658043x_2 - 0,543486300980384y_2 + 1,08479802518778z_2 \\
 & + 0,344041927386078x_3 + 0,0767905870025835y_3 \\
 & + 0,649757354494466z_3 + 0,838949001135772
 \end{aligned}$$

$$f_9(\mathbf{x}) = 1,41669698735990x_2 + 0,705867245375851$$

Problem redcyc5

Dimenzija problema:4

Varijable: $\mathbf{x} = (y_1, y_2, y_3, y_4)$

$$f_1(\mathbf{x}) = 1 + y_1 + y_2 + y_3 + y_4$$

$$f_2(\mathbf{x}) = y_1 + y_1 \cdot y_2 + y_2 \cdot y_3 + y_3 \cdot y_4 + y_4$$

$$f_3(\mathbf{x}) = y_1 \cdot y_2 + y_1 \cdot y_2 \cdot y_3 + y_2 \cdot y_3 \cdot y_4 + y_3 \cdot y_4 + y_4 \cdot y_1$$

$$f_4(\mathbf{x}) = y_1 \cdot y_2 \cdot y_3 + y_1 \cdot y_2 \cdot y_3 \cdot y_4 + y_2 \cdot y_3 \cdot y_4 + y_3 \cdot y_4 \cdot y_1 + y_4 \cdot y_1 \cdot y_2$$

Problem redcyc6

Dimenzija problema:5

Varijable: $\mathbf{x} = (y_1, y_2, y_3, y_4, y_5)$

$$f_1(\mathbf{x}) = 1 + y_1 + y_2 + y_3 + y_4 + y_5$$

$$f_2(\mathbf{x}) = y_1 + y_1 y_2 + y_2 y_3 + y_3 y_4 + y_4 y_5 + y_5$$

$$f_3(\mathbf{x}) = y_1 y_2 + y_1 y_2 y_3 + y_2 y_3 y_4 + y_3 y_4 y_5 + y_4 y_5 + y_5 y_1$$

$$f_4(\mathbf{x}) = y_1 y_2 y_3 + y_1 y_2 y_3 y_4 + y_2 y_3 y_4 y_5 + y_3 y_4 y_5 + y_4 y_5 y_1 + y_5 y_1 y_2$$

$$f_5(\mathbf{x}) = y_1 y_2 y_3 y_4 + y_1 y_2 y_3 y_4 y_5 + y_2 y_3 y_4 y_5 + y_3 y_4 y_5 y_1 + y_4 y_5 y_1 y_2 + y_5 y_1 y_2 y_3$$

Problem redcyc7

Dimenzija problema:6

Varijable: $\mathbf{x} = (y_1, y_2, y_3, y_4, y_5, y_6)$

$$f_1(\mathbf{x}) = 1 + y_1 + y_2 + y_3 + y_4 + y_5 + y_6$$

$$f_2(\mathbf{x}) = y_1 + y_1 y_2 + y_2 y_3 + y_3 y_4 + y_4 y_5 + y_5 y_6 + y_6$$

$$f_3(\mathbf{x}) = y_1 y_2 + y_1 y_2 y_3 + y_2 y_3 y_4 + y_3 y_4 y_5 + y_4 y_5 y_6 + y_5 y_6 + y_6 y_1$$

$$f_4(\mathbf{x}) = y_1 y_2 y_3 + y_1 y_2 y_3 y_4 + y_2 y_3 y_4 y_5 + y_3 y_4 y_5 y_6 + y_4 y_5 y_6 + y_5 y_6 y_1 + y_6 y_1 y_2$$

$$f_5(\mathbf{x}) = y_1 y_2 y_3 y_4 + y_1 y_2 y_3 y_4 y_5 + y_2 y_3 y_4 y_5 y_6 + y_3 y_4 y_5 y_6 + y_4 y_5 y_6 y_1 + y_5 y_6 y_1 y_2 + y_6 y_1 y_2 y_3$$

$$f_6(\mathbf{x}) = y_1 y_2 y_3 y_4 y_5 + y_1 y_2 y_3 y_4 y_5 y_6 + y_2 y_3 y_4 y_5 y_6 + y_3 y_4 y_5 y_6 y_1 + y_4 y_5 y_6 y_1 y_2$$

$$+ y_5 y_6 y_1 y_2 y_3 + y_6 y_1 y_2 y_3 y_4$$

Problem redcyc8

Dimenzija problema:7

Varijable: $\mathbf{x} = (y_1, y_2, y_3, y_4, y_5, y_6, y_7)$

$$f_1(\mathbf{x}) = 1 + y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7$$

$$f_2(\mathbf{x}) = y_1 + y_1 y_2 + y_2 y_3 + y_3 y_4 + y_4 y_5 + y_5 y_6 + y_6 y_7 + y_7$$

$$f_3(\mathbf{x}) = y_1 y_2 + y_1 y_2 y_3 + y_2 y_3 y_4 + y_3 y_4 y_5 + y_4 y_5 y_6 + y_5 y_6 y_7 + y_6 y_7 + y_7 y_1$$

$$f_4(\mathbf{x}) = y_1 y_2 y_3 + y_1 y_2 y_3 y_4 + y_2 y_3 y_4 y_5 + y_3 y_4 y_5 y_6 + y_4 y_5 y_6 y_7 + y_5 y_6 y_7 + y_6 y_7 y_1 + y_7 y_1 y_2$$

$$f_5(\mathbf{x}) = y_1 y_2 y_3 y_4 + y_1 y_2 y_3 y_4 y_5 + y_2 y_3 y_4 y_5 y_6 + y_3 y_4 y_5 y_6 y_7 + y_4 y_5 y_6 y_7 + y_5 y_6 y_7 y_1$$

$$+ y_6 y_7 y_1 y_2 + y_7 y_1 y_2 y_3$$

$$f_6(\mathbf{x}) = y_1 y_2 y_3 y_4 y_5 + y_1 y_2 y_3 y_4 y_5 y_6 + y_2 y_3 y_4 y_5 y_6 y_7 + y_3 y_4 y_5 y_6 y_7 + y_4 y_5 y_6 y_7 y_1 + y_5 y_6 y_7 y_1 y_2 \\ + y_6 y_7 y_1 y_2 y_3 + y_7 y_1 y_2 y_3 y_4$$

$$f_7(\mathbf{x}) = y_1 y_2 y_3 y_4 y_5 y_6 + y_1 y_2 y_3 y_4 y_5 y_6 y_7 + y_2 y_3 y_4 y_5 y_6 y_7 + y_3 y_4 y_5 y_6 y_7 y_1 + y_4 y_5 y_6 y_7 y_1 y_2 \\ + y_5 y_6 y_7 y_1 y_2 y_3 + y_6 y_7 y_1 y_2 y_3 y_4 + y_7 y_1 y_2 y_3 y_4 y_5$$

Problem redeco5

Dimenzija problema:5

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, u_5)$

$$f_1(\mathbf{x}) = x_1 + x_1 \cdot x_2 + x_2 \cdot x_3 + x_3 \cdot x_4 - u_5$$

$$f_2(\mathbf{x}) = x_2 + x_1 \cdot x_3 + x_2 \cdot x_4 - 2 \cdot u_5$$

$$f_3(\mathbf{x}) = x_3 + x_1 \cdot x_4 - 3 \cdot u_5$$

$$f_4(\mathbf{x}) = x_4 - 4 \cdot u_5$$

$$f_5(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + 1$$

Problem redeco6

Dimenzija problema:6

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, u_6)$

$$f_1(\mathbf{x}) = x_1 + x_1 \cdot x_2 + x_2 \cdot x_3 + x_3 \cdot x_4 + x_4 \cdot x_5 - u_6$$

$$f_2(\mathbf{x}) = x_2 + x_1 \cdot x_3 + x_2 \cdot x_4 + x_3 \cdot x_5 - 2 \cdot u_6$$

$$f_3(\mathbf{x}) = x_3 + x_1 \cdot x_4 + x_2 \cdot x_5 - 3 \cdot u_6$$

$$f_4(\mathbf{x}) = x_4 + x_1 \cdot x_5 - 4 \cdot u_6$$

$$f_5(\mathbf{x}) = x_5 - 5 \cdot u_6$$

$$f_6(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + x_5 + 1$$

Problem redeco7

Dimenzija problema:7

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, u_7)$

$$f_1(\mathbf{x}) = x_1 + x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_5 + x_5 x_6 - u_7$$

$$f_2(\mathbf{x}) = x_2 + x_1 x_3 + x_2 x_4 + x_3 x_5 + x_4 x_6 - 2 u_7$$

$$f_3(\mathbf{x}) = x_3 + x_1 x_4 + x_2 x_5 + x_3 x_6 - 3 u_7$$

$$\begin{aligned}
 f_4(\mathbf{x}) &= x_4 + x_1 x_5 + x_2 x_6 - 4 u_7 \\
 f_5(\mathbf{x}) &= x_5 + x_1 x_6 - 5 u_7 \\
 f_6(\mathbf{x}) &= x_6 - 6 u_7 \\
 f_7(\mathbf{x}) &= x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + 1
 \end{aligned}$$

Problem redeco8

Dimenzija problema:8

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, u_8)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= x_1 + x_1 \cdot x_2 + x_2 \cdot x_3 + x_3 \cdot x_4 + x_4 \cdot x_5 + x_5 \cdot x_6 + x_6 \cdot x_7 - 1 \cdot u_8 \\
 f_2(\mathbf{x}) &= x_2 + x_1 \cdot x_3 + x_2 \cdot x_4 + x_3 \cdot x_5 + x_4 \cdot x_6 + x_5 \cdot x_7 - 2 \cdot u_8 \\
 f_3(\mathbf{x}) &= x_3 + x_1 \cdot x_4 + x_2 \cdot x_5 + x_3 \cdot x_6 + x_4 \cdot x_7 - 3 \cdot u_8 \\
 f_4(\mathbf{x}) &= x_4 + x_1 \cdot x_5 + x_2 \cdot x_6 + x_3 \cdot x_7 - 4 \cdot u_8 \\
 f_5(\mathbf{x}) &= x_5 + x_1 \cdot x_6 + x_2 \cdot x_7 - 5 \cdot u_8 \\
 f_6(\mathbf{x}) &= x_6 + x_1 \cdot x_7 - 6 \cdot u_8 \\
 f_7(\mathbf{x}) &= x_7 - 7 \cdot u_8 \\
 f_8(\mathbf{x}) &= x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + 1
 \end{aligned}$$

Problem rediff3

Dimenzija problema:3

Varijable: $\mathbf{x} = (x_1, x_2, x_3)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= -2 \cdot x_1 + x_2 + 0,835634534 \cdot x_1 - 0,835634534 \cdot x_1^2 \\
 f_2(\mathbf{x}) &= x_1 - 2 \cdot x_2 + x_3 + 0,835634534 \cdot x_2 - 0,835634534 \cdot x_2^2 \\
 f_3(\mathbf{x}) &= x_2 - 2 \cdot x_3 + 0,835634534 \cdot x_3 - 0,835634534 \cdot x_3^2
 \end{aligned}$$

Problem reimer 2

Dimenzija problema:2

Varijable: $\mathbf{x} = (x, y)$

$$\begin{aligned}
 f_1(\mathbf{x}) &= x^2 - y^2 - 0,5 \\
 f_2(\mathbf{x}) &= x^3 - y^3 - 0,5
 \end{aligned}$$

Problem reimer3

Dimenzija problema:3

Varijable: $\mathbf{x} = (x, y, z)$

$$f_1(\mathbf{x}) = x^2 - y^2 + z^2 - 0,5$$

$$f_2(\mathbf{x}) = x^3 - y^3 + z^3 - 0,5$$

$$f_3(\mathbf{x}) = x^4 - y^4 + z^4 - 0,5$$

Problem reimer4

Dimenzija problema:4

Varijable: $\mathbf{x} = (x, y, z, t)$

$$f_1(\mathbf{x}) = x^2 - y^2 + z^2 - t^2 - 0,5$$

$$f_2(\mathbf{x}) = x^3 - y^3 + z^3 - t^3 - 0,5$$

$$f_3(\mathbf{x}) = x^4 - y^4 + z^4 - t^4 - 0,5$$

$$f_4(\mathbf{x}) = x^5 - y^5 + z^5 - t^5 - 0,5$$

Problem reimer5

Dimenzija problema:5

Varijable: $\mathbf{x} = (x, y, z, t, u)$

$$f_1(\mathbf{x}) = -1 + 2x^2 - 2y^2 + 2z^2 - 2t^2 + 2u^2$$

$$f_2(\mathbf{x}) = -1 + 2x^3 - 2y^3 + 2z^3 - 2t^3 + 2u^3$$

$$f_3(\mathbf{x}) = -1 + 2x^4 - 2y^4 + 2z^4 - 2t^4 + 2u^4$$

$$f_4(\mathbf{x}) = -1 + 2x^5 - 2y^5 + 2z^5 - 2t^5 + 2u^5$$

$$f_5(\mathbf{x}) = -1 + 2x^6 - 2y^6 + 2z^6 - 2t^6 + 2u^6$$

Problem rose

Dimenzija problema:3

Varijable: $\mathbf{x} = (x, y, z)$

$$f_1(\mathbf{x}) = y^4 - 2,85714285714286x^2$$

$$f_2(\mathbf{x}) = x^2z^4 + 0,7xz^4 + 0,145833333333333z^4 - 1,85185185185185x^2 - 1,2962962962963x - 0,226851851851852$$

$$\begin{aligned}
 f_3(\mathbf{x}) = & 0,6x^6y^2z + x^5y^3 + 0,428571428571429x^5y^2z + 1,4x^4y^3 - 0,35x^4yz^2 - 0,15x^4z^3 \\
 & + 0,609x^3y^3 + 0,315x^3y^2z - 0,616x^3yz^2 - 0,42x^3z^3 + 0,0392x^2y^3 + 0,0735x^2y^2z \\
 & - 0,3977166666667x^2yz^2 - 0,2275x^2z^3 - 0,03423875xy^3 + 0,00517125xy^2z \\
 & - 0,1149866666667xyz^2 - 0,029435xz^3 - 0,00643125y^3 \\
 & - 0,0126838541666667yz^2 - 0,0026796875z^3
 \end{aligned}$$

Problem rosenbrock

Dimenzija problema:2

Varijable: $\mathbf{x} = (x,y)$

$$f_1(\mathbf{x}) = 400x^3 - 400xy + 2x - 2$$

$$f_2(\mathbf{x}) = 200y - 200x^2$$

Problem s9_1

Dimenzija problema:8

Varijable: $\mathbf{x} = (a,b,c,d,e,f,g,h)$

$$f_1(\mathbf{x}) = -e \cdot g - 2 \cdot d \cdot h$$

$$f_2(\mathbf{x}) = 9 \cdot e + 4 \cdot b$$

$$f_3(\mathbf{x}) = -4 \cdot c \cdot h - 2 \cdot e \cdot f - 3 \cdot d \cdot g$$

$$f_4(\mathbf{x}) = -7 \cdot c + 9 \cdot a - 8 \cdot f$$

$$f_5(\mathbf{x}) = -4 \cdot d \cdot f - 5 \cdot c \cdot g - 6 \cdot h - 3 \cdot e$$

$$f_6(\mathbf{x}) = -5 \cdot d - 6 \cdot c \cdot f - 7 \cdot g + 9 \cdot b$$

$$f_7(\mathbf{x}) = 9 \cdot d + 6 \cdot a - 5 \cdot b$$

$$f_8(\mathbf{x}) = 9 \cdot c - 7 \cdot a + 8$$

Problem sendra

Dimenzija problema:2

Varijable: $\mathbf{x} = (x,y)$

$$f_1(\mathbf{x}) = -270x^4y^3 - 314xy^4 - 689xy^3 + 1428$$

$$\begin{aligned}
 f_2(\mathbf{x}) = & 36x^7 + 417x^6y - 200x^6 - 422x^5y^2 - 174x^5y - 270x^4y^3 - 966x^4y^2 \\
 & + 529x^4y + 36x^4 + 1428x^3y^4 + 529x^3y^3 - 1475x^2y^5 + 269x^2y^4 + 1303x^2y^3 \\
 & - 788x^2y^2 + 510xy^6 + 49xy^5 - 314xy^4 - 689xy^3 - 267y^6 + 262y^5 + 177y^4
 \end{aligned}$$

Problem solotarev

Dimenzija problema:4

Variable: $\mathbf{x} = (x, a, b, y)$

$$f_1(\mathbf{x}) = 3 \cdot x^2 - 2 \cdot x - a$$

$$f_2(\mathbf{x}) = x^3 - x^2 - x \cdot a + a - 2 \cdot b - 2$$

$$f_3(\mathbf{x}) = 3 \cdot y^2 - 2 \cdot y - a$$

$$f_4(\mathbf{x}) = y^3 - y^2 - y \cdot a - a + 2$$

Problem stewgou40

Dimenzija problema:9

Variable: $\mathbf{x} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, n_1, n_2, n_3)$

$$f_1(\mathbf{x}) = n_1^2 + n_2^2 + n_3^2 - 1$$

$$f_2(\mathbf{x}) = h_{11}^2 + h_{12}^2 + h_{13}^2 - 1$$

$$f_3(\mathbf{x}) = h_{21}^2 + h_{22}^2 + h_{23}^2 - 1$$

$$f_4(\mathbf{x}) = h_{11} \cdot h_{21} + h_{12} \cdot h_{22} + h_{13} \cdot h_{23}$$

$$f_5(\mathbf{x}) = n_1^2 + 1,085610 \cdot n_1 \cdot h_{11} - 2,215830 \cdot n_1 + 0,2946372680 \cdot h_{11}^2 - 1,202763603 \cdot h_{11} \\ + 0,8110958214 + n_2^2 + 1,085610 \cdot n_2 \cdot h_{12} + 0,2946372680 \cdot h_{12}^2 + n_3^2 \\ + 1,085610 \cdot n_3 \cdot h_{13} + 0,2946372680 \cdot h_{13}^2$$

$$f_6(\mathbf{x}) = -1,050876963 h_{11} - 1,446982100 h_{12} + 0,5808481060 h_{21} + 0,7997861232 h_{22} \\ - 1,098188 n_1 - 1,512126 n_2 - 1,012257626 h_{12} h_{22} + 0,9156939726 h_{11}^2 \\ + 0,9156939726 h_{12}^2 + 0,2797510772 h_{23}^2 + 0,9156939726 h_{13}^2 + n_2^2 + n_3^2 \\ - 1,012257626 h_{11} h_{21} + n_1^2 - 1,012257626 h_{13} h_{23} + 0,2797510772 h_{22}^2 \\ + 0,2797510772 h_{21}^2 - 1,057830 n_3 h_{23} - 0,3068774482 + 1,913838 n_1 h_{11} \\ + 1,913838 n_2 h_{12} + 1,913838 n_3 h_{13} - 1,057830 n_1 h_{21} - 1,057830 n_2 h_{22}$$

$$\begin{aligned}
f_7(\mathbf{x}) = & 2,805076n_1 h_{12} h_{23} - 0,9789534962h_{11} + 0,2982299150h_{12} - 0,7004990340h_{13} \\
& + 0,5196729762h_{21} - 0,1583139833h_{22} + 0,3718567014h_{23} - 1,470154n_1 \\
& + 0,447870n_2 - 1,051982n_3 + 2,805076n_3 h_{11} h_{22} - 0,4707567232h_{12} h_{22} \\
& - 2,061946850h_{12} h_{23} - 0,6281546940h_{11} h_{23} - 1,475444730h_{11} h_{22} \\
& + 1,475444730h_{12} h_{21} - 2,805076n_1 h_{13} h_{22} + 2,805076n_2 h_{13} h_{21} \\
& - 2,805076n_2 h_{11} h_{23} + 0,4434028332h_{11}^2 + 0,4434028332h_{12}^2 \\
& + 0,1249495243h_{23}^2 + 0,4434028332h_{13}^2 + 0,6281546940h_{13} h_{21} \\
& + 2,061946850h_{13} h_{22} + n_2^2 + n_3^2 - 2,805076n_3 h_{12} h_{21} - 0,4707567232h_{11} h_{21} + n_1^2 \\
& - 0,4707567232h_{13} h_{23} - 1,393177215 + 0,1249495243h_{22}^2 + 0,1249495243h_{21}^2 \\
& - 3,934225682h_{11} h_{22} h_{12} h_{21} + 1,967112841h_{11}^2 h_{22}^2 + 1,967112841h_{12}^2 h_{21}^2 \\
& - 0,706964n_3 h_{23} + 1,967112841h_{12}^2 h_{23}^2 + 1,967112841h_{13}^2 h_{22}^2 + 1,331770n_1 h_{11} \\
& + 1,331770n_2 h_{12} + 1,331770n_3 h_{13} - 3,934225682h_{12} h_{23} h_{13} h_{22} \\
& + 1,967112841h_{13}^2 h_{21}^2 + 1,967112841h_{11}^2 h_{23}^2 - 3,934225682h_{13} h_{21} h_{11} h_{23} \\
& - 0,706964n_1 h_{21} - 0,706964n_2 h_{22} \\
f_8(\mathbf{x}) = & 0,215344n_1 h_{12} h_{23} - 0,4919329150h_{11} + 0,5032939412h_{12} + 0,3524721322h_{13} \\
& - 1,191622463h_{21} + 1,219142585h_{22} + 0,8538028204h_{23} - 1,028376n_1 \\
& + 1,052126n_2 + 0,736836n_3 + 0,215344n_3 h_{11} h_{22} + 1,108589329h_{12} h_{22} \\
& - 0,1107273007h_{12} h_{23} - 0,1132845107h_{11} h_{23} + 0,7933660580E-1 h_{11} h_{22} \\
& - 0,7933660580E-1 h_{12} h_{21} - 0,215344n_1 h_{13} h_{22} + 0,215344n_2 h_{13} h_{21} - 0,215344n_2 h_{11} h_{23} \\
& + 0,2288273329h_{11}^2 + 0,2288273329h_{12}^2 + 1,342683023h_{23}^2 \\
& + 0,2288273329h_{13}^2 + 0,1132845107h_{13} h_{21} + 0,1107273007h_{13} h_{22} + n_2^2 + n_3^2 \\
& - 0,215344n_3 h_{12} h_{21} + 1,108589329h_{11} h_{21} + n_1^2 + 1,108589329h_{13} h_{23} \\
& + 1,342683023h_{22}^2 + 1,342683023h_{21}^2 - 0,2318651916E-1 h_{11} h_{22} h_{12} h_{21} \\
& + 0,1159325958E-1 h_{11}^2 h_{22}^2 + 0,1159325958E-1 h_{12}^2 h_{21}^2 + 2,317484n_3 h_{23} \\
& + 0,1159325958E-1 h_{12}^2 h_{23}^2 + 0,1159325958E-1 h_{13}^2 h_{22}^2 + 0,956718n_1 h_{11} + 0,956718n_2 h_{12} \\
& + 0,956718n_3 h_{13} - 0,2318651916E-1 h_{12} h_{23} h_{13} h_{22} + 0,1159325958E-1 h_{13}^2 h_{21}^2 \\
& + 0,1159325958E-1 h_{11}^2 h_{23}^2 - 0,2318651916E-1 h_{13} h_{21} h_{11} h_{23} + 2,317484n_1 h_{21} \\
& + 2,317484n_2 h_{22} - 0,9664888140
\end{aligned}$$

$$\begin{aligned}
 f_9(\mathbf{x}) = & 0,707826n_1h_{12}h_{23} + 0,1618923443h_{11} + 0,2597332554E-1h_{12} - 0,5621060514E-1h_{13} \\
 & + 0,2776652044h_{21} + 0,4454743538E-1h_{22} - 0,9640807436E-1h_{23} \\
 & - 1,180946n_1 - 0,189466n_2 + 0,410036n_3 + 0,707826n_3h_{11}h_{22} \\
 & + 0,6446406506E-1h_{12}h_{22} - 0,4179521416h_{12}h_{23} + 0,6705448046E-1h_{11}h_{23} \\
 & + 0,1451170709h_{11}h_{22} - 0,1451170709h_{12}h_{21} - 0,707826n_1h_{13}h_{22} + 0,707826n_2h_{13}h_{21} \\
 & - 0,707826n_2h_{11}h_{23} + 0,1879284557E-1h_{11}^2 + 0,1879284557E-1h_{12}^2 \\
 & + 0,5528188464E-1h_{23}^2 + 0,1879284557E-1h_{13}^2 - 0,6705448046E-1h_{13}h_{21} \\
 & + 0,4179521416h_{13}h_{22} + n_2^2 + n_3^2 - 0,707826n_3h_{12}h_{21} + 0,6446406506E-1h_{11}h_{21} + n_1^2 \\
 & + 0,6446406506E-1h_{13}h_{23} + 0,5528188464E-1h_{22}^2 + 0,5528188464E-1h_{21}^2 \\
 & - 0,2505088232h_{11}h_{22}h_{12}h_{21} + 0,1252544116h_{11}^2h_{22}^2 + 0,1252544116h_{12}^2h_{21}^2 - 0,470242n_3h_{23} \\
 & + 0,1252544116h_{12}^2h_{23}^2 + 0,1252544116h_{13}^2h_{22}^2 - 0,274174n_1h_{11} - 0,274174n_2h_{12} \\
 & - 0,274174n_3h_{13} - 0,2505088232h_{12}h_{23}h_{13}h_{22} + 0,1252544116h_{13}^2h_{21}^2 - 0,470242n_2h_{22} \\
 & + 0,1252544116h_{11}^2h_{23}^2 - 0,2505088232h_{13}h_{21}h_{11}h_{23} - 0,470242n_1h_{21} - 0,1948854017
 \end{aligned}$$

Problem tangents0

Dimenzija problema:6

Varijable: $\mathbf{x} = (x_0, x_1, x_2, x_3, x_4, x_5)$

$$f_1(\mathbf{x}) = x_0^2 + x_1^2 + x_2^2 - 1$$

$$f_2(\mathbf{x}) = x_0x_3 + x_1x_4 + x_2x_5$$

$$f_3(\mathbf{x}) = 50x_0^2 - 2x_0x_4 + 14x_0x_5 + x_1^2 - 14x_1x_2 + 2x_1x_3 + 49x_2^2 - 14x_2x_3 + x_3^2 + x_4^2 + x_5^2 - 1$$

$$\begin{aligned}
 f_4(\mathbf{x}) = & 29x_0^2 - 10x_0x_1 - 4x_0x_2 - 4x_0x_4 + 10x_0x_5 + 5x_1^2 - 20x_1x_2 + 4x_1x_3 - 2x_1x_5 + 26x_2^2 \\
 & - 10x_2x_3 + 2x_2x_4 + x_3^2 + x_4^2 + x_5^2 - 1
 \end{aligned}$$

$$f_5(\mathbf{x}) = 9x_0^2 - 6x_0x_5 + 9x_2^2 + 6x_2x_3 + x_3^2 + x_4^2 + x_5^2 - 1$$

$$f_6(\mathbf{x}) = 9x_0^2 + 6x_0x_5 + 9x_2^2 - 6x_2x_3 + x_3^2 + x_4^2 + x_5^2 - 1$$

Problem tangents1

Dimenzija problema:6

Varijable: $\mathbf{x} = (x_0, x_1, x_2, x_3, x_4, x_5)$

$$f_1(\mathbf{x}) = x_0^2 + x_1^2 + x_2^2 - 1$$

$$f_2(\mathbf{x}) = x_0x_3 + x_1x_4 + x_2x_5$$

$$f_3(\mathbf{x}) = x_3^2 + x_4^2 + x_5^2 - 0,25$$

$$f_4(\mathbf{x}) = x_3^2 + x_4^2 - 2x_2x_4 + x_2^2 + x_5^2 + 2x_1x_5 + x_1^2 - 0,25$$

$$f_5(\mathbf{x}) = x_3^2 + \sqrt{3}x_2x_3 + 0,75x_2^2 + x_4^2 - x_2x_4 + 0,25x_2^2 + x_5^2 - \sqrt{3}x_0x_5 + x_1x_5 + 0,75x_0^2 - \frac{\sqrt{3}}{2}x_0x_1 + 0,25x_1^2 - 0,25$$

$$f_6(\mathbf{x}) = x_3^2 - \frac{2\sqrt{6}}{3}x_1x_3 + \frac{\sqrt{3}}{3}x_2x_3 + \frac{2}{3}x_1^2 - \frac{\sqrt{2}}{3}x_1x_2 + \frac{1}{12}x_2^2 + x_4^2 + \frac{2\sqrt{6}}{3}x_0x_4 - x_2x_4 + \frac{2}{3}x_0^2 - \frac{\sqrt{6}}{3}x_0x_2 + 0,25x_2^2 + x_5^2 - \frac{\sqrt{3}}{3}x_0x_5 + x_1x_5 + \frac{1}{12}x_0^2 - \frac{\sqrt{3}}{6}x_0x_1 + 0,25x_1^2 - 0,25$$

Problem tangents2

Dimenzija problema:6

Varijable: $\mathbf{x} = (x_0, x_1, x_2, x_3, x_4, x_5)$

$$f_1(\mathbf{x}) = x_0^2 + x_1^2 + x_2^2 - 1$$

$$f_2(\mathbf{x}) = x_0x_3 + x_1x_4 + x_2x_5$$

$$f_3(\mathbf{x}) = x_3^2 + x_4^2 + x_5^2 - 0,3$$

$$f_4(\mathbf{x}) = x_3^2 + x_4^2 - 2x_2x_4 + x_2^2 + x_5^2 + 2x_1x_5 + x_1^2 - 0,25$$

$$f_5(\mathbf{x}) = x_3^2 + \sqrt{3}x_2x_3 + 0,75x_2^2 + x_4^2 - x_2x_4 + 0,25x_2^2 + x_5^2 - \sqrt{3}x_0x_5 + x_1x_5 + 0,75x_0^2 - \frac{\sqrt{3}}{2}x_0x_1 + 0,25x_1^2 - 0,25$$

$$f_6(\mathbf{x}) = x_3^2 - \frac{2\sqrt{6}}{3}x_1x_3 + \frac{\sqrt{3}}{3}x_2x_3 + \frac{2}{3}x_1^2 - \frac{\sqrt{2}}{3}x_1x_2 + \frac{1}{12}x_2^2 + x_4^2 + \frac{2\sqrt{6}}{3}x_0x_4 - x_2x_4 + \frac{2}{3}x_0^2 - \frac{\sqrt{6}}{3}x_0x_2 + 0,25x_2^2 + x_5^2 - \frac{\sqrt{3}}{3}x_0x_5 + x_1x_5 + \frac{1}{12}x_0^2 - \frac{\sqrt{3}}{6}x_0x_1 + 0,25x_1^2 - 0,25$$

Problem toms1

Dimenzija problema:2

Varijable: $\mathbf{x} = (x_1, x_2)$

$$f_1(\mathbf{x}) = 4x_1^3 - 3x_1 - x_2$$

$$f_2(\mathbf{x}) = x_1^2 - x_2 = 0$$

Problem toms12

Dimenzija problema:3

Varijable: $\mathbf{x} = (x_1, x_2, x_3)$

$$f_1(\mathbf{x}) = 5x_1^9 - 6x_1^5x_2^2 + x_1x_2^4 + 2x_1x_3$$

$$f_2(\mathbf{x}) = -2x_1^6x_2 + 2x_1^2x_2^3 + 2x_2x_3$$

$$f_3(\mathbf{x}) = x_1^2 + x_2^2 - 0,265625$$

Problem trinks

Dimenzija problema:6

Varijable: $\mathbf{x} = (x, y, z, u, v, t)$

$$\begin{aligned} f_1(\mathbf{x}) &= 45 \cdot y + 35 \cdot u - 165 \cdot v - 36 \\ f_2(\mathbf{x}) &= 35 \cdot y + 25 \cdot z + 40 \cdot t - 27 \cdot u \\ f_3(\mathbf{x}) &= 25 \cdot y \cdot u - 165 \cdot v^2 + 15 \cdot x - 18 \cdot z + 30 \cdot t \\ f_4(\mathbf{x}) &= 15 \cdot y \cdot z + 20 \cdot t \cdot u - 9 \cdot x \\ f_5(\mathbf{x}) &= -11 \cdot v^3 + x \cdot y + 2 \cdot z \cdot t \\ f_6(\mathbf{x}) &= -11 \cdot u \cdot v + 3 \cdot v^2 + 99 \cdot x \end{aligned}$$

Problem utbikker

Dimenzija problema:4

Varijable: $\mathbf{x} = (t, x, y, z)$

$$\begin{aligned} f_1(\mathbf{x}) &= x^2 + xy + y^2 - 2xz - 4yz + 3z^2 - 3xt + 2yt + t^2 - 3x - 2y + 3z - 2t - 2 \\ f_2(\mathbf{x}) &= 2x^2 - xy + y^2 - xz - yz - 6z^2 - xt + yt - 5zt - 3t^2 - 5x + y + 5z + 2t + 5 \\ f_3(\mathbf{x}) &= -3 - 3xy + 2xz + xt^2 - 5xz^2 - 5z^2t - 3xt - 2zt + xyz + xyt - x^2z + x^2 - y^2 + 2z^2 \\ &\quad + 11z - 2t - x + y + x^3 + y^3 - 3z^3 + 2t^3 - 3t^2 - 5y^2z + 7yz^2 \\ f_4(\mathbf{x}) &= -15 + 2xy + 11xt^2 + 5xz^2 - zt - 4xyz + 6xyt - x^2z + 3x^2 + 2y^2 - z^2 + 4z - 10t - 35x \\ &\quad - 14y - x^3 + 6y^3 + 15z^3 + 4t^3 + 5t^2 + 6y^2z + 4yz^2 - xzt + 6x^2y - 12xy^2 - 7y^2t + 2yt \end{aligned}$$

Problem virasoro

Dimenzija problema:8

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$

$$\begin{aligned} f_1(\mathbf{x}) &= 8x_1^2 + 8x_1x_2 + 8x_1x_3 + 2x_1x_4 + 2x_1x_5 + 2x_1x_6 + 2x_1x_7 - 8x_2x_3 - 2x_4x_7 - 2x_5x_6 - x_1 \\ f_2(\mathbf{x}) &= 8x_1x_2 - 8x_1x_3 + 8x_2^2 + 8x_2x_3 + 2x_2x_4 + 2x_2x_5 + 2x_2x_6 + 2x_2x_7 - 2x_4x_6 - 2x_5x_7 - x_2 \\ f_3(\mathbf{x}) &= -8x_1x_2 + 8x_1x_3 + 8x_2x_3 + 8x_3^2 + 2x_3x_4 + 2x_3x_5 + 2x_3x_6 + 2x_3x_7 - 2x_4x_5 - 2x_6x_7 - x_3 \\ f_4(\mathbf{x}) &= 2x_1x_4 - 2x_1x_7 + 2x_2x_4 - 2x_2x_6 + 2x_3x_4 - 2x_3x_5 + 8x_4^2 + 8x_4x_5 + 2x_4x_6 + 2x_4x_7 \\ &\quad + 6x_4x_8 - 6x_5x_8 - x_4 \\ f_5(\mathbf{x}) &= 2x_1x_5 - 2x_1x_6 + 2x_2x_5 - 2x_2x_7 - 2x_3x_4 + 2x_3x_5 + 8x_4x_5 - 6x_4x_8 + 8x_5^2 + 2x_5x_6 \\ &\quad + 2x_5x_7 + 6x_5x_8 - x_5 \end{aligned}$$

$$f_6(\mathbf{x}) = -2x_1x_5 + 2x_1x_6 - 2x_2x_4 + 2x_2x_6 + 2x_3x_6 - 2x_3x_7 + 2x_4x_6 + 2x_5x_6 + 8x_6^2 + 8x_6x_7 \\ + 6x_6x_8 - 6x_7x_8 - x_6$$

$$f_7(\mathbf{x}) = -2x_1x_4 + 2x_1x_7 - 2x_2x_5 + 2x_2x_7 - 2x_3x_6 + 2x_3x_7 + 2x_4x_7 + 2x_5x_7 + 8x_6x_7 - 6x_6x_8 \\ + 8x_7^2 + 6x_7x_8 - x_7$$

$$f_8(\mathbf{x}) = -6x_4x_5 + 6x_4x_8 + 6x_5x_8 - 6x_6x_7 + 6x_6x_8 + 6x_7x_8 + 8x_8^2 - x_8$$

Problem winkler

Dimenzija problema:3

Varijable: $\mathbf{x} = (x, y, z)$

$$f_1(\mathbf{x}) = 4 \cdot x \cdot z - 4 \cdot x \cdot y^2 - 16 \cdot x^2 - 1$$

$$f_2(\mathbf{x}) = 2 \cdot y^2 \cdot z + 4 \cdot x + 1$$

$$f_3(\mathbf{x}) = 2 \cdot x^2 \cdot z + 2 \cdot y^2 + x$$

Problem wood

Dimenzija problema:4

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4)$

$$f_1(\mathbf{x}) = 200 \cdot x_1^3 - 200 \cdot x_1 \cdot x_2 + x_1 - 1$$

$$f_2(\mathbf{x}) = -100 \cdot x_1^2 + 110,1 \cdot x_2 + 9,9 \cdot x_4 - 20$$

$$f_3(\mathbf{x}) = 180 \cdot x_3^3 - 180 \cdot x_3 \cdot x_4 + x_3 - 1$$

$$f_4(\mathbf{x}) = -90 \cdot x_3^2 + 9,9 \cdot x_2 + 100,1 \cdot x_4 - 20$$

Problem wright

Dimenzija problema:5

Varijable: $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$

$$f_1(\mathbf{x}) = x_1^2 - x_1 + x_2 + x_3 + x_4 + x_5 - 10$$

$$f_2(\mathbf{x}) = x_2^2 + x_1 - x_2 + x_3 + x_4 + x_5 - 10$$

$$f_3(\mathbf{x}) = x_3^2 + x_1 + x_2 - x_3 + x_4 + x_5 - 10$$

$$f_4(\mathbf{x}) = x_4^2 + x_1 + x_2 + x_3 - x_4 + x_5 - 10$$

$$f_5(\mathbf{x}) = x_5^2 + x_1 + x_2 + x_3 + x_4 - x_5 - 10$$

Prilog B. Opis programa PolySolver

U ovom prilogu opisan je program PolySolver namijenjen rješavanju polinomnih sustava metodom najdublje spusta. Dan je pregled mogućnosti programa i opisane su glavne radnje koje program može napraviti.

U drugom dijelu dodatka opisane su glavne strukture podataka koje se u programu koriste za reprezentaciju polinomnog sustava.

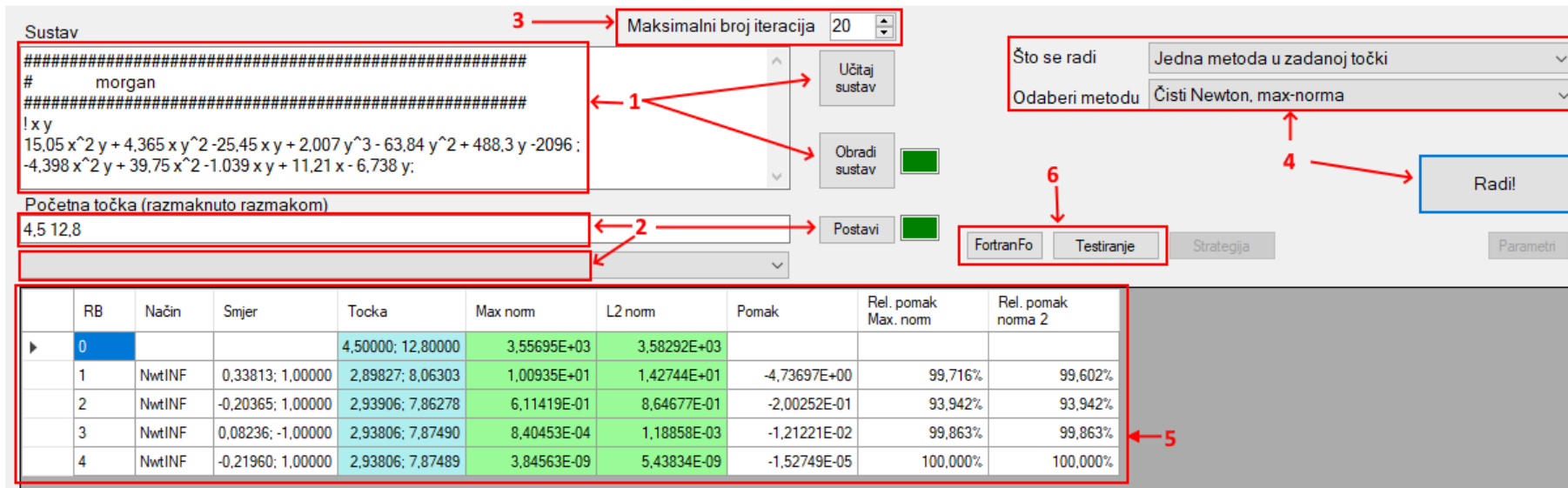
Pregled mogućnosti programa PolySolver

Program PolySolver je namijenjen rješavanju polinomnih sustava. Program je napisan u programskom jeziku Visual Basic, a korisniku omogućuje da upiše ili učitava određeni polinomni sustav, zada početne točke i vidi kako se pojedine metode najdublje spusta ponašaju. Također, moguće je testirati ponašanje metoda na zadanom problemu na nizu početnih točaka koje se nalaze u nekoj datoteci te spremite rezultate za daljnju obradu.

Sučelje programa je orijentirano korisniku i omogućuje mu da s minimalnom količinom rada dođe do željenog rezultata. Na slici 1 prikazan je izgled glavne forme i njezinih najvažnijih dijelova.

Dio 1 sačinjavaju dijelovi vezani uz zadavanje polinomnog sustava. Polinomni sustav se može ili učitati iz postojeće tekstualne datoteke (gumb Učitaj sustav) ili direktno upisati u za to predviđeno polje. Bez obzira na način zadavanja, treba poštovati sljedeća pravila:

- Jednadžbe sustava trebaju biti odvojene simbolom ; (točka-zarez) .
- Kod pisanja jednadžbi, umnožak se može pisati ili pomoću simbola * ili samo navođenjem razmaka. Na primjer, izrazi $2 x y$ i $2*x*y$ su jednaki.



Slika 1: Glavni dijalog programa PolySolver. 1 - dijelovi dijaloga povezani s odabirom sustava, 2 - dijelovi dijaloga povezani s odabirom početnih točaka, 3 - postavljanje maksimalnog broja iteracija, 4 - postavljanje akcije i parametara akcije, 5 - rezultati rada, 6 - dodatne mogućnosti.

	Metoda	Iteracija	T	MaxRes	Ukupno provjera
▶	NwtINF	4	2,93806; 7,87489	3,84563E-09	4
	GradNwtINF	9	-3,53111; 7,87558	1,36424E-12	14
	GradOrNwtINF	4	2,93806; 7,87489	3,84563E-09	8
	Nwt2	4	2,93806; 7,87489	4,54747E-13	4
	GradNwt2	21	-3,53636; 7,92726	2,82730E+00	20
	GradOrNwt2	4	2,93806; 7,87489	4,54747E-13	8
	GS_INF	16	-3,53111; 7,87558	4,54747E-13	16
	GS_2	6	2,93806; 7,87489	8,52651E-14	6
	LPadap_INF	2	2,51463; 10,81463	1,22338E+02	2
	CoorAx_INF	21	2,30226; 12,80000	1,78748E+02	2
	CoorAx2	21	2,78223; 8,89497	4,95823E+01	20

Slika 2: Prikaz rezultata rada kada je odabrana akcija Sve metode u zadanoj točki.

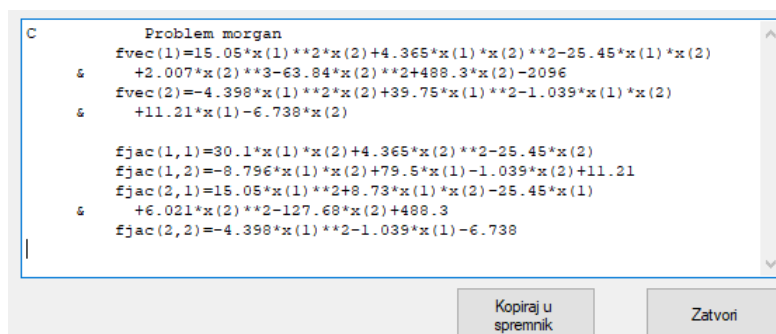
- Potencija se piše ili pomoću simbola \wedge ili pomoću simbola **, kao u programskom jeziku FORTRAN. Tako se izraz x^5 može zapisati kao $x\wedge 5$, odnosno kao $x**5$.
- Program ne može evaluirati izraze u zagradama, stoga jednadžbe sustava trebaju biti zapisane u raširenom obliku.
- Nazivi nepoznanica se mogu sastojati od više slova ili znamenaka, s time da na prvom mjestu treba biti slovo.
- Nije nužno na početku navesti koje se nepoznanice koriste u sustavu. Ako se to želi napraviti, tada prije sustava treba staviti liniju koja započinje sa simbolom ! i iza nje navesti nepoznanice u željenom redoslijedu. Ako poredak nije naveden, onda se smatra da je redoslijed nepoznanica jednak redoslijedu pojavljivanja u sustavu.
- Linije koje započinju simbolom # smatraju se komentarima i ne obrađuju se.
- Ako se sustavu želi dati ime pod kojim će ga program pamtit, tada prva linija sustava treba biti komentar oblika #*****, a u drugoj liniji se navodi naziv sustava, opet kao komentar. Primjer za sustav morgan prikazan je na slici 1.

Kada je sustav učitao ili upisan treba ga obraditi (gumb Obradi sustav) i ako je sve u redu mali pravokutnik desno od gumba Obradi sustav će pozeleniti.

Dio 2 sačinjavaju elementi dijaloga povezani s postavljanjem početne točke. Početna točka za sustav reda n sastoji se od n brojeva odijeljenih razmakom. Brojevi se zapisuju u standardnom decimalnom formatu. Nakon što je početna točka odabrana, treba je postaviti (gumb Postavi). Sve početne točke se čuvaju u padajućem izborniku i od tamo ih je moguće ponovno postaviti. Također, početna točka se može postaviti i iz rezultata izračuna.

Dio 3 predstavlja element u kojem korisnik može odabrati maksimalni broj iteracija koje će program izvesti, a u dijelu 4 se bira vrsta akcije i odgovarajući parametri. Korisnik može odabrati između rješavanja sustava jednom odabranom metodom ili rješavanja sustava svim metodama. Izlaz programa u prvom slučaju je prikazan na slici 1, dio 5, a u drugom slučaju na slici 2. U slučaju kada se promatra rad jedne metode u zadanoj početnoj točki (slika 1, dio 5), prikazuju se informacije o rednom broju iteracije, načinu na koji je odabran smjer spusta, vektor smjera spusta, normiran u max–normi (najveća komponenta ima vrijednost ± 1), točka u kojoj se iteracijski postupak trenutno nalazi, max–norma reziduala, euklidska (L_2) norma reziduala, kolika je bila duljina spusta (stupac pomak), te relativno smanjenje reziduala u max–normi i euklidskoj normi.

Kada se promatraju rezultati rada za sve metode (slika 2), tada se prikazuju podatci o kojoj je metodi riječ, broj napravljenih iteracija (ima smisla samo ako je metoda konvergirala rješenju), točka u kojoj se metoda zaustavila nakon zadanog broja iteracija, max–norma reziduala (stupac MaxRes) i koliko je bilo ukupno provjera smjera. Ako je iterativni proces za neku metodu konvergirao prema rješenju tada se mijenja i boja pozadine odgovarajućeg retka. Na slici 2 se može vidjeti kako je većina metoda konvergirala prema rješenju, s time da su dvije metode



```

C      Problem morgan
      fvec(1)=15.05*x(1)**2*x(2)+4.365*x(1)*x(2)**2-25.45*x(1)*x(2)
&      +2.007*x(2)**3-63.84*x(2)**2+488.3*x(2)-2096
      fvec(2)=-4.398*x(1)**2*x(2)+39.75*x(1)**2-1.039*x(1)*x(2)
&      +11.21*x(1)-6.738*x(2)

      fjac(1,1)=30.1*x(1)*x(2)+4.365*x(2)**2-25.45*x(2)
      fjac(1,2)=-8.796*x(1)*x(2)+79.5*x(1)-1.039*x(2)+11.21
      fjac(2,1)=15.05*x(1)**2+8.73*x(1)*x(2)-25.45*x(1)
&      +6.021*x(2)**2-127.68*x(2)+488.3
      fjac(2,2)=-4.398*x(1)**2-1.039*x(1)-6.738

```

Slika 3: Generirani kod u programskom jeziku FORTRAN za računanje vrijednosti polinomnog sustava i njegova gradijenta.

(GN-M i GS-M) konvergirale prema jednom rješenju, a preostale metode prema drugom rješenju problema.

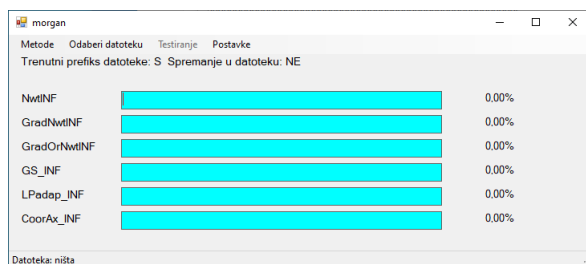
Nazivi metoda u programu razlikuju se od naziva metoda korištenih u radu, ali nazivi su dovoljno sugestivni. Na primjer, metoda GradNwtINF iz programa odgovara metodi GN-M, a GradOrNwt2 je BGN-E metoda iz rada.

U dijelu 6 na slici 1 nalaze se dva gumba. Prvi gumb, FortranForm, se koristi za generiranje koda u programskom jeziku FORTRAN koji služi za izračunavanje vrijednosti zadanog sustava, kao i za računanje vrijednosti gradijenta sustava. Taj kod se kasnije koristi u programima napravljenim za usporedbu efikasnosti metoda najdublje spusta s otprije poznatim metodama (MinPack i NLEQ). Izgled generiranog koda za problem Morgan prikazan je na slici 3. Vrijednosti sustava se nalaze u vektoru *fvec*, a vrijednosti gradijenta sustava u matrici *fjac*.

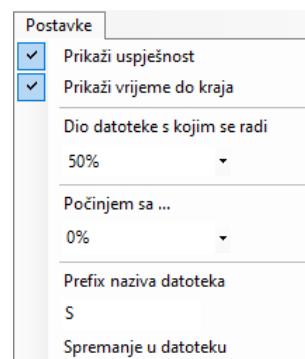
Drugi gumb, Testiranje, otvara novi dijalog u kojem se korisniku programa omogućuje da testira rad metoda na zadanom problemu na nizu točaka koje se nalaze u nekoj datoteci. Na slici 4 prikazan je dijalog i glavne mogućnosti testiranja. Korisnik može odabrati hoće li se dinamički prikazivati uspješnost rješavanja i koliko je preostalo vrijeme do završetka (zbog brzine je bolje isključiti), hoće li se testirati na svim točkama datoteke ili samo na dijelu točaka te hoće li se rezultati spremati u datoteku tako da bi se mogli kasnije obraditi (slika 4.b). Ime datoteke se generira automatski iz naziva problema i naziva datoteke, a rezultati se spremaju u formatu pogodnom za daljnju obradu. Moguće je odabrati više datoteka odjednom, a program će za svaku ulaznu datoteku generirati posebnu izlaznu datoteku s rezultatima testiranja.

Korisnik isto tako može odabrati koje metode će se testirati (slika 4.c), što je pogotovo zgodno za slučaj kada se želi napraviti brza usporedba nekoliko metoda.

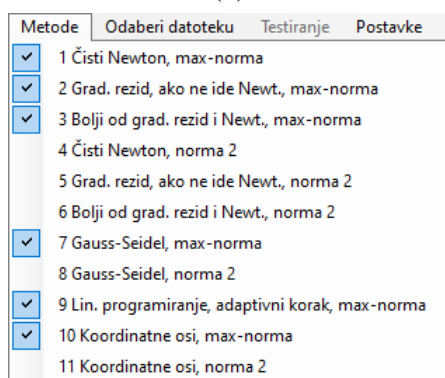
Kao prvi rezultat testiranja prikazuje se uspješnost metode u smislu za koliki postotak početnih točaka je iterativni postupak konvergirao prema rješenju. Na slici 4.d je prikazan završetak testiranja na problemu Bifurcation te uspješnost pojedinih metoda.



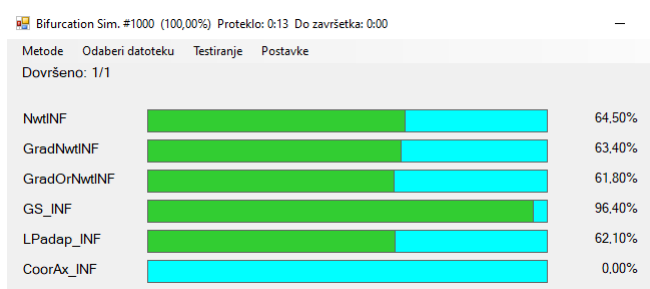
(a)



(b)



(c)



(d)

Slika 4: Dijalog u kojem se zadani problem rješava na nizu početnih točaka. (a) - početni izgled dijaloga, (b) - postavke testiranja na zadanom problemu, (c) - odabrane metode za testiranje, (d) - izgled dijaloga nakon testiranja.

Glavne strukture podataka

Za opisivanje polinomnog sustava u programu PolySover koriste se tri klase

- `PolynomialSystem` - klasa koja opisuje cijeli polinomni sustav.
- `Multinom` - klasa koja opisuje jednu jednadžbu sustava (polinom u više varijabli).
- `Monom` - klasa koja opisuje jedan monom koji se sastoji od više varijabli.

Klasa `PolynomialSystem` je glavna klasa u kojoj se nalaze podatci o nepoznicama (nazivi i interno kodiranje), jednadžbama sustava (polje `multinoma`) i njihovim parcijalnim derivacijama, koji se simbolički izračunavaju prilikom inicijalizacije sustava. Obzirom da jedan objekt tipa `PolynomialSystem` sadrži sve informacije o sustavu, on predstavlja jedan od ulaznih argumenata za potprograme koji implementiraju pojedine metode. Unutar klase su definirane metode za izračun vrijednosti sustava u zadanoj točki, izračun Jacobijeve matrice sustava u zadanoj točki te metode koji za zadanu točku \mathbf{x} i smjer \mathbf{s} određuju koeficijente polinoma $f_{i,\mathbf{x},\mathbf{s}}$.

Klasa `Multinom` je implementirana kao lista monoma, pri čemu je lista implementirana pomoću polja. Glavne metode implementirane na njoj su metode za izračun vrijednosti `multinoma` te određivanje najvećeg stupnja `multinoma`.

Klasa `Monom` služi za opis monoma, tj. izraza oblika $ax_1^{d_1} \cdot x_n^{d_n}$. Monom je opisan pomoću

cjelobrojnog polja (d_1, \dots, d_n) eksponenata. Najvažnija metoda ove klase je izračun vrijednosti monoma u zadanoj točki.

Za opis točaka i smjerova napravljen je klasa `Vektor`.

Sve metode najdublje spusta su implementirane na modularan način. Na prvoj razini, za sve metode su napravljeni potprogrami koji za zadani polinomni sustav, početnu točku i maksimalni broj iteracija provode cijeli postupak i vraćaju točku do koje je iterativni postupak stigao, informaciju o uspjehu postupka te broj napravljenih iteracija i provjera smjera. Svaki od tih potprograma, ovisno o metodi, poziva osnovni potprogram koji provodi jedan korak iterativnog postupka koristeći određeni smjer spusta i način određivanja duljine spusta. Ulazni parametri osnovnih potprograma su polinomni sustav i početna točka, a potprogrami vraćaju novu točku te koliki je bio pomak u smjeru. Ako je pomak bio 0, onda metoda nije uspjela napredovati.

Zahvaljujući ovakvoj realizaciji metoda, moguće je jednostavno nadograđivati program novim mogućnostima te dodavati i nove metode.

Životopis

Ivo Beroš rođen je 10.8.1972. u Splitu. Osnovnu školu završio je u Starome Gradu na Hvaru, a srednju školu je završio u Jelsi. Godine 1991. upisao je studij Matematike na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu. Diplomirao je 1996. s temom *Uvjetna aproksimacija polinomnim splineom*. Voditelj diplomskog rada je bio prof.dr. Miljenko Marušić. Poslijediplomski studij matematike upisao je 1995. godine, a magistarski rad s temom *Metoda kolokacija za rješavanje singularno perturbiranog rubnog problema za ODJ napetim splajnovima višeg reda* obranio je 2001. godine. Voditelj magistarskog rada je bio prof.dr. Miljenko Marušić.

Od 1997. do 2004. bio je zaposlen kao asistent na PMF-Matematički odjel u Zagreb gdje je održavao vježbe iz kolegija koji pokrivaju različita područja matematike i računarstva. U razdoblju od 2004. do 2019. bio je zaposlen na Veleučilištu VERN' kao predavač iz matematičkih i računarskih kolegija. Od rujna 2019. zaposlen je na Institutu za turizam, Zagreb, na radnom mjestu asistenta. Napisao je 10 znanstvenih i stručnih radova.

Popis objavljenih djela

Rad u časopisima

1. Beroš, I., Hlupić, N., Brčić, M., “Computational Aspects of Efficient Estimation of Harmonically Related Sine-Waves”, *Automatika Journal for Control, Measurement, Electronics, Computing and Communications*, Vol. 56 No. 3, veljača 2016., str. 359-366.
2. Beroš, I., Meter, J., “An integer programming model for assigning students to elective courses”, *Croatian Operational Research Review*, Vol. 6, No. 2, listopad 2015., str. 511–524.
3. Hlupic, N., Beroš, I., Basch, D., “A Derivative-free Algorithm for Finding Least Squares Solutions of Quasi-linear and Linear Systems”. *CIT. Journal of Computing and Information Technology*, vol.21 No. 2, 2013.,str. 125–135.
4. Rajsman, M., Badanjak, D., Beroš, I., “Prognostic trend model of passenger transport performance of the Croatian maritime traffic system”, *International Journal for Engineering Modelling*, Vol. 20 No. 1-4, 2007., str. 23–27.

5. Rajsman, M., Beroš, I., “Model razvitka putničke prijevozne potražnje u hrvatskom pomorskom prometnom sustavu”, *Naše more, Znanstveni časopis za more i pomorstvo*, Vol. 53 No. 3-4, listopad 2006., str. 112-117.
6. Beroš, I., Marušić, M., (1999). “Evaluation of tension splines”, *Mathematical Communications*, vol. 4 No. 1, lipanj 1999., str. 73–81.

Rad u zbornicima konferencija

1. Hlupić, N., Beroš, I., Basch, D., “A derivative-free algorithm for solving quasi-linear systems”, In *Proceedings of the ITI 2013 35th International Conference on Information Technology Interfaces (ITI)*. 2013., str. 297–302.
2. Beroš, I., Marušić, M., “Solving Parabolic Singularly Perturbed Problems by Collocation Using Tension Splines”, In Z. Drmač, M. Marušić, and Z. Tutek (Eds.), *Proceedings of the Conference on Applied Mathematics and Scientific Computing*, 2005., str. 131–140.
3. Beroš, I., Marušić, M., “Asymptotic Behaviour of Tension Spline Collocation Matrix”, In Z. Drmač, V. Hari, L. Sopta, Z. Tutek, and K. Veselić (Eds.), *Applied Mathematics and Scientific Computing*, 2003., str. 199–208.
4. Beroš, I., Marušić, M., “Collocation with High Order Tension Splines”, In M. Rogina, V. Hari, and Z. Tutek (Eds.), *Proceedings of the 1. Conference on Applied Mathematics and Computation Dubrovnik, Croatia*, 1999., str. 91-99.

Biography

Ivo Beroš was born in Split, Croatia on 10 August 1972. He graduated in mathematics from the University of Zagreb, Faculty of Science, Department of Mathematics in 1996. His graduation thesis was *Constrained Approximation By Polynomial Splines*. He received his M.Sc. degree in 2001 from the University of Zagreb, Faculty of Science, Department of Mathematics. The title of his thesis was *The Collocation Method for Solving Singularly Perturbed Problems for ODE with High Order Tension Splines*.

From 1997 to 2005 he worked as a teaching assistant at the University of Zagreb, Faculty of Science, Department of Mathematics. From 2005 to 2019, he worked as the lecturer at Vern' University of Applied Sciences, Zagreb. Since recently (2019), he has worked at the Institute for Tourism, Zagreb.

He has published 10 scientific articles. He speaks English.