Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁRSKA PRÁCA

Miroslav Jakubík

Vizualizace algoritmů lineární algebry

Katedra aplikované matematiky

Vedúci bakalárskej práce: RNDr. Ondřej Pangrác, Ph.D.

Študijní program: Informatika, Obecná informatika

2008

Chcel by som poďakovať všetkým, ktorí mi pomohli s touto prácou. V prvom rade môjmu vedúcemu ročníkového projektu a bakalárskej práce RNDr. Ondřejovi Pangrácovi, Ph.D., za trpezlivosť a dobré rady.

Prehlasujem, že som svoju bakalársku prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce a jej zverejňovaním.

V Prahe dňa Miroslav Jakubík

OBSAH

Názov práce: Vizualizace algoritmů lineární algebry
Autor: Miroslav Jakubík
Katedra (ústav): Katedra aplikované matematiky
Vedúci bakalárskej práce: RNDr. Ondřej Pangrác, Ph.D.
e-mail vedúceho: pangrac@kam.mff.cuni.cz

Abstrakt: Predložená práca sa zameriava na problematiku implementácie algoritmov lineárnej algebry. Tieto algoritmy nie sú príliš zložité, problém nastáva v tom, že počítač môže počas výpočtu pracovať s extrémnymi hodnotami (veľké čísla, zlomky s veľkým menovateľom, …) aj keď zadanie takéto hodnoty neobsahuje. Preto, ak chceme prezentovať medzivýsledky výpočtu, je vhodné, aby počítač postupoval ľudským postupom počítania a snažil sa takýmto extrémnym hodnotám vyhnúť.

Kľúčové slova: Matica, Gauss, Gauss-Jordan, Inverzná matica, Determinant

Title: Visualization of linear algebra algorithms
Author: Miroslav Jakubík
Department: Department of Applied Mathematics
Supervisor: RNDr. Ondřej Pangrác, Ph.D.
Supervisor's e-mail: pangrac@kam.mff.cuni.cz

Abstract: This text focuses on problems of implementation of basic linear algebra algorithms. These algorithms are not too complicated, but there is a certain problem of using extremely huge or extremely small numbers which are not comprehensible to the human. Thus, in case of displaying intermediate outcomes, it is better when calculations are made human-like.

Keywords: Matrix, Gauss, Gauss-Jordan, Inverse matrix, Determinant

KAPITOLA 1

Úvod

Táto bakalárska práca nadväzuje na ročníkový projekt, ktorého úlohou bolo naimplementovať prostredie pre maticové výpočty. Hlavným zámerom tohto projektu je vizualizovať priebeh týchto operácií na matici spôsobom, ktorý by pripomínal ľudský spôsob počítania.
Z tohto dôvodu priložený program nedosahuje taký výkon ako iné programy, napr. Mathematica alebo Matlab.

Program pracuje s maticami, ktoré môžu byť zadané celými číslami, číslami v desatinnom tvare, prípadne zlomkami.

Program je vytvorený tak, aby bol jednoduchý na ovládanie pre každého. Vstupné dáta (zadanie matice, zadanie operácií, ktoré sa prevedú na matici) sú zadávané do textového okna, kde je zároveň možné ovládať formu výsledkov. Tým sa minimalizovalo použitie pomocných okien.

Program prevádza nasledujúce operácie na maticiach:

- Gaussová eliminačná metóda
- Gauss-Jordanova eliminačná metóda
- inverzná matica
- determinant
- mocnina matice
- sčítanie, odčítanie, násobenie matíc

Najčastejšie používaná operácia je Gaussová eliminačná metóda, ktorá sa využíva za istých okolností vo všetkých týchto uvedených maticových operáciách.

Základne pojmy a definície

Sústava lineárnych rovníc
Nech $x_1, x_2, \ldots x_n$ – je n premenných (neznámych),
$a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, a_{22}, \ldots a_{mn}$; $b, \ldots b_m$ – sú reálne čísla
potom sústavou m lineárnych rovníc o n neznámych s koeficientmi $a_{11}, \ldots a_{mn}$ a vektorom pravých strán $b, \ldots b_n$ rozumieme

$$a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots a_{mn}x_n = b_m$$

Pod maticou A typu m x n, kde $m,n \in N$, rozumieme obdĺžnikovú schému reálnych čísiel, ktoré sú zapísané v m riadkoch a n stĺpcoch.

Každej sústave lineárnych rovníc vieme priradiť práve jednu maticu A typu m x n

$$A=\begin{pmatrix} a11 & a12 & \cdots & a1n \\ a21 & a22 & \cdots & a2n \\ \vdots & \vdots & & \vdots \\ am1 & am2 & \cdots & amn \end{pmatrix}$$

Rozšírená matica sústavy typu m x (n+1)

$$(A/b) = \begin{pmatrix} a11 & a12 & \cdots & a1n & b1 \\ a21 & a22 & \cdots & a2n & b2 \\ \vdots & \vdots & & \vdots & \vdots \\ am1 & am2 & \cdots & amn & bm \end{pmatrix}$$, kde b je vektor pravých strán $b = \begin{pmatrix} b1 \\ b2 \\ \vdots \\ bm \end{pmatrix}$

Ekvivalentné úpravy
Pri riešení sústav lineárnych rovníc sa používajú nasledujúce elementárne úpravy (elementárne riadkové úpravy)

1. vynásobenie i-tého riadku číslom t, $t \neq 0$
2. pripočítanie j-tého riadku k i-tému riadku

ďalšie 2 úpravy sa dajú odvodiť z predchádzajúcich úprav

- pripočítanie t-násobku j-tého riadku k i-tému riadku
- zámena dvoch riadkov

Tvrdenie:
Elementárne úpravy sústavy lineárnych rovníc nemenia množinu riešení danej sústavy.

Odstupňovaný tvar matice

Hovoríme, že matica A typu m x n je v odstupňovanom tvare, ak existuje číslo r, $0 \leq r \leq m$ také, že r+1, r+2, m-tý riadok je nulový a 1, ... r-tý riadok je nenulový a každý z týchto riadkov ma dlhší úsek počiatočných núl ako jeho predchodca.

Veta:

Každú maticu je možno riadkovými elementárnymi úpravami previezť na odstupňovanú maticu.

Hlavná diagonála

Pozostáva z prvkov matice a_{11}, a_{22}, ... a_{nn}.

Jednotková matica

Jednotkovou maticou rádu n nazveme štvorcovú maticu typu n x n , ktorá má na hlavnej diagonále jednotky a nuly na zvyšných pozíciách matice. Označenie I_n.

Regulárna, singulárna matica

Štvorcová matica sa nazýva regulárna, ak hodnosť matice je rovná svojmu rádu, v opačnom prípade, t.j. keď hodnosť matice je menšia ako rád matice, sa nazýva singulárna matica.

Horná trojuholníková matica

Matica, ktorá má pod diagonálou iba nulové prvky. ($a_{ij} = 0$, pre $i>j$)

KAPITOLA 2

Gaussová eliminačná metóda

Metóda riešenia sústav lineárnych algebrických rovníc.

- zostavíme rozšírenú maticu sústavy (A/b)
- pomocou elementárnych riadkových úprav prevedieme maticu na odstupňovaný tvar
- pomocou spätnej substitúcie popíšeme všetky riešenia

Počet riešení

- vo výslednej matici nesmie posledný stĺpec matice A obsahovať pivot, inak sústava nemá riešenie.
- ak počet nenulových riadkov vo výslednej matici je rovný počtu premenných, potom sústava ma jediné riešenie
- ak počet nenulových riadkov vo výslednej matici je menší ako počet premenných , potom sústava ma nekonečne veľa riešení

Postup Gaussovej eliminačnej metódy nie je jednoznačný, v jednotlivých krokoch eliminácie môžme voliť rôzne elementárne úpravy, ktoré rôzne pozmenia maticu. Preto výsledná matica po prevedení eliminácie nie je pri rôznych postupoch rovnaká. Môžu sa líšiť v hodnotách jednotlivých prvkov, no množina riešení danej matice však zostava rovnaká.

Pod označením jeden krok Gaussovej eliminácie rozumejme prevedenie takých elementárnych úprav na matici, po ktorých dostaneme v prvom stĺpci matice pivota (prvý riadok matice začína nenulovým prvkom, zvyšné riadky matice majú na prvej pozícii nulu).

Naznačenie algoritmu Gaussovej eliminačnej metódy

1. Zjednodušíme prvky matice.
2. Usporiadame riadky matice podľa dĺžky počiatočných úsekov núl.
3. Odstránime prípadne nulové stĺpce matice na začiatku matice.
4. Zameníme za prvý riadok matice riadok s najvhodnejším kandidátom na pivota. Ak matica neobsahuje riadok začínajúci prvkom +/-1, pokúsime sa ekvivalentnými úpravami riadkov matice takýto riadok vytvoriť (bod 5). Ak matica obsahuje takýto riadok, pokračujeme bodom 7.

5. Pomocou ekvivalentných úprav riadkov matice vytvoríme riadky začínajúce prvkom +/-1, ohodnotíme vykonané úpravy
6. Vytvoríme zoznam matíc, pričom každá matica má za prvý riadok zvolený jeden riadok z tých, ktoré sme dostali v predchádzajúcom bode 5. Na každej matici prevediem jeden krok Gaussovej eliminácie (bod 7, 8, 9) a zo získaných hodnôt matíc vyberiem tú najlepšiu, ktorá ďalej pokračuje v eliminácií.

7. Zistíme možné elementárne úpravy matice, ktorými nulujeme vytvoríme pivota v prvom stĺpci, ohodnotíme vykonané úpravy
8. Vytvoríme zoznam všetkých matíc, ktoré dokážeme vytvoriť prevedením nájdených elementárnych úprav jednotlivých riadkov matice, ohodnotíme vzniknuté matice.
9. Vyberieme najlepšie ohodnotenú maticu, ktorá ďalej pokračuje v eliminácií.

10. Odstránime prvý riadok a prvý stĺpec matice.
11. Ak je počet riadkov matice väčší ako 1, prevedieme opakovaný výpočet na zmenšenej matici. (pokračujeme bodom 1)

1. Zjednodušíme prvky matice

Pre každý riadok matice:

- hľadám najväčšieho spoločného deliteľa všetkých prvkov. Ak nájdem deliteľa, ktorého hodnota je rôzna od 1, predelím všetky prvky daného riadku týmto deliteľom.
- skontrolujem, či všetky zlomky daného riadku sú v základnom tvare.

Zjednodušovanie vykonávam hneď na začiatku algoritmu pretože, aj človek pri prevádzaní eliminácie sa snaží udržiavať maticu v čo najjednoduchšom tvare.

Ak by som pri hľadaní vhodných ekvivalentných úprav uvažoval pôvodný riadok a aj jeho zjednodušený tvar, rýchlo narastá počet možných ekvivalentných úprav, ktoré sa dajú na matici vykonať a tým aj počet matíc, ktoré môžu prevedením týchto úprav vzniknúť. Tým by sa veľmi zvýšila časová náročnosť programu. (viď 7. bod algoritmu)

Zjednodušený riadok má samozrejme nižšiu hodnotu prvého prvku. To môže v istých prípadoch znamenať, že pomocou násobkov tohto riadku môžem upraviť viacero zvyšných riadkov matice ako s pôvodným nezjednodušeným.

V nutnom prípade prenásobením zjednodušeného riadku opäť dostanem na pôvodný riadok.

2. Usporiadame riadky matice podľa dĺžky počiatočných úsekov núl
Usporiadanie riadkov podľa počtu núl na začiatku riadkov je operácia, ktorá v podstate žiadnym spôsobom neovplyvňuje priebeh výpočtu. Pre počítač je to zanedbateľná úprava matice. Je tu však nutná z dôvodu vypisovania medzivýsledkov eliminácie v „peknom“ tvare. Človek počas výpočtu eliminácie sa snaží udržiavať riadky matice v tomto usporiadaní.

3. Odstránime prípadné nulové stĺpce matice na začiatku matice
Táto operácia prechádza maticu po stĺpcoch, postupuje od prvého až po taký stĺpec, ktorý neobsahuje samé nuly. Všetky takéto nulové stĺpce sa odstraňujú. Týmto odpadá potreba pomocného indexu určujúceho, s ktorým stĺpcom sa aktuálne pracuje. A preto v každom kroku eliminácie vytváram pivota v prvom stĺpci.

Táto operácia má dôležitú funkciu pri maticiach typu m x n, kde m $\geq$ n. Matica takéhoto typu obsahuje väčší počet riadkov ako je potrebné k zisteniu riešenia pomocou Gaussovej eliminácie.
Po určitom počte krokov eliminácie matice takéhoto typu sa dostanem do tohto kroku algoritmu s maticou, ktorá bude tvorená buď len nulovými prvkami, prípadne nulovými prvkami s výnimkou niektorých prvkov v poslednom stĺpci. V takomto prípade končí algoritmus eliminácie, pretože matica sa nedá ďalej upravovať.
Je to z dôvodu, že po každom kroku eliminácie sa matica zmenší o prvý riadok a prvý stĺpec. (viď 10 bod algoritmu eliminácie)

Môžu nastať 2 prípady:

a) riadky matice sú lineárne závislé, vtedy sa po prevedení ekvivalentných úprav niektoré riadky matice prevedú na nulové

   napr. matica A= $\begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 2 & 4 \end{pmatrix}$ sa prevedie ekvivalentnými úpravami

   3. riadok deleno 2
   2. riadok mínus 3. riadok
   3. riadok mínus 1. riadok

na maticu $\begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

b) riadky matice nie sú lineárne závislé, po niekoľkých krokoch Gaussovej eliminácie dostávame pivota v poslednom stĺpci niektorého riadku matice, čiže matica nemá riešenie

napr. matica A= $\begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 4 \end{pmatrix}$ sa prevedie ekvivalentnými úpravami

2. riadok mínus 1. riadok
3. riadok mínus 2 riadok

na maticu $\begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

4. Zameníme za prvý riadok matice riadok s najvhodnejším kandidátom na pivota

Za prvý riadok matice sa snažíme vyberať taký riadok, ktorý sa dá najlepšie využiť pri ekvivalentných úpravách matice.

- jednou z najlepších možnosti je zvoliť riadok, ktorý začína prvkom +/-1, pretože prvky zvyšných riadkov v danom stĺpci sa dajú ľahko nulovať pomocou násobkov „jednotky“ (násobkov riadku s počiatočným prvkom 1)
- je to jediný riadok pomocou, ktorého sa môžu upravovať všetky zvyšné riadky.
- je to riadok, ktorý sa v ďalšom priebehu eliminácie už nemení.
- vyberáme len z tých riadkov, ktoré začínajú nenulovým prvkom.

Program pri výbere prvého riadku matice postupuje nasledovne:

a) prechádza postupne riadky matice, ak nájde riadok, ktorý začína prvkom +/-1, nastaví tento riadok ako prvý.
b) ak matica neobsahuje riadok, ktorý začína prvkom +/-1, pokúsime sa vytvoriť taký riadok (viď 5. bod algoritmu)
c) ak matica neobsahuje riadok, ktorý začína prvkom +/-1 a ani sa nedá taký riadok vhodnými úpravami vytvoriť. Zvolí sa riadok, ktorý začína najmenším prvkom v absolútnej hodnote. Pri viacerých riadkoch s rovnakým prvým prvkom sa ďalej vyberie podľa najmenšieho súčtu prvkov

5. Vytvoríme riadky matice, ktorý začínajú prvkom +/-1

Riadok začínajúci prvkom +/-1 sa snažím vytvoriť pomocou ekvivalentných úprav niektorých dvoch riadkov matice. Pomocou troch a viacerých riadkov sa už nepokúšam. Vo väčšine prípadov to nie je triviálne, a preto by tento postup často nepripomínal ľudský spôsob počítania.

Po nájdení vhodnej ekvivalentnej úpravy dvojice riadkov matice a následnom vytvorení nového riadku s počiatočným prvkom, ktorý je rovný +/-1, nahradzujem novým riadkom ten z dvojice riadkov, ktorý začína väčším prvkom.

Každá úprava, ktorá sa prevedie na matici, prejde vyhodnocovacou funkciou, ktorá podľa určitých vlastností ohodnotí prevedenú úpravu. (viď Ohodnotenie ekvivalentných úprav)

Aby sme dostali nejakou úpravou z dvoch čísiel jednotku, musí platiť, že tieto čísla musia byť nesúdeliteľné, inak z nich nikdy nevytvoríme jednotku.

Prípady, ktoré môžu nastať:

a) Hľadám ekvivalentnú úpravu medzi 2 riadkami, ktoré začínajú nesúdeliteľnými číslami

napr. A=$\begin{pmatrix} 3 & 1 & 4 \\ 7 & 1 & 8 \end{pmatrix}$ sa prevedie ekvivalentnými úpravami

2. riadok mínus 1. riadok * 2
zámena 2. riadku a 1. riadku

na maticu $\begin{pmatrix} 1 & -1 & 0 \\ 3 & 1 & 4 \end{pmatrix}$

b) Hľadám ekvivalentnú úpravu medzi 2 riadkami, ktoré začínajú zlomkami s rovnakými menovateľmi a nesúdeliteľnými čitateľmi. Keďže menovatele oboch zlomkov sú rovnaké, ani sa nebudú meniť. Upravuje sa len hodnota čitateľa.

Hľadám také násobky oboch riadkov, ktoré po odčítaní vytvoria riadok s prvým prvkom v tvare menovateľ/menovateľ, čo sa rovná 1.

Možnosť a) je špeciálny prípad s hodnotou menovateľa 1.

napr. A=$\begin{pmatrix} 7/3 & 5/3 & 4 \\ 2/3 & 1/3 & 1 \end{pmatrix}$ sa prevedie ekvivalentnou úpravou

1. riadok mínus 2. riadok * 2

na maticu $\begin{pmatrix} 1 & 1 & 2 \\ 2/3 & 1/3 & 1 \end{pmatrix}$

c) Hľadám ekvivalentnú úpravu medzi 2 riadkami, ktoré začínajú zlomkami s nesúdeliteľnými čitateľmi a aspoň jeden menovateľ je rovný 1.
V tomto prípade sa riadok, ktorý nezačína zlomkom s menovateľom 1, celý prenásobí hodnotou menovateľa prvého prvku. Takto odstránime zlomky na prvých pozíciách riadkov, a tým sa dostaneme na možnosť a).

d) Všetky zvyšné možnosti, ktoré môžu nastať sa už neskúšajú. Z toho dôvodu, že vhodnou úpravou niektorých riadkov by sme vytvorili riadok začínajúci prvkom +/-1 ale násobky týchto riadkov, ktoré by boli potrebné, nie sú vo väčšine prípadov „pekné" zlomky a napohľad nie je zrejmé ako sme k ním dospeli.

6. Vytvoríme zoznam matíc

Vytvorím zoznam rôznych matíc, v ktorom každá matica má za prvý riadok dosadený jeden z riadkov, ktoré vznikli v predchádzajúcom 5. bode algoritmu eliminácie. Postupne na každej matici zo zoznamu prevediem jeden krok Gaussovej eliminácie (bod 7-9), a potom vyberiem najlepšiu maticu, s ktorou pokračujem ďalej v eliminácii.

7. Zistíme možné elementárne úpravy matice, ktorými vytvoríme pivota v prvom stĺpci

V tomto kroku algoritmu zistím všetky možnosti ekvivalentných úprav matice. Hľadám také úpravy matice, že po ich prevedení bude mať matica okrem prvého riadku vo všetkých zvyšných riadkoch v prvom stĺpci nulový prvok (vytvorím pivota v prvom stĺpci).

Keď nájdem ekvivalentnú úpravu i-tého riadku pomocou j-tého riadku, zapamätám si zároveň aj opačnú úpravu riadkov matice, čiže úprava j-tého riadku pomocou i-tého riadku. Ak nájdená úprava spočíva v odčítaní vhodných násobkov týchto riadkov, pri opačnej úprave sa zmení znamienko všetkým prvkom nového riadku, pri sčítaní sa znamienko nemení.

Každá úprava, ktorá sa prevedie na matici, prejde vyhodnocovacou funkciou, ktorá podľa určitých vlastností ohodnotí prevedenú ekvivalentnú úpravu. (viď Ohodnotenie ekvivalentných úprav)

Prípady, ktoré môžu nastať:

a) Hľadám ekvivalentnú úpravu medzi 2 riadkami, ktoré začínajú prirodzeným číslom. Nájdem najmenší spoločný násobok oboch čísiel, následne prenásobím oba riadky takou hodnotou, aby počiatočné prvky oboch riadkov boli rovné hodnote spoločného násobku.

napr. riadky matice $\begin{pmatrix} 2 & 1 & 3 \\ 3 & 1 & 4 \end{pmatrix}$, najmenší spoločný násobok je 6

ekvivalentnou úpravou prevedieme
2. riadok * 2 mínus 1. riadok * 3

na maticu $\begin{pmatrix} 2 & 1 & 3 \\ 0 & -1 & -1 \end{pmatrix}$

b) Hľadám ekvivalentnú úpravu medzi 2 riadkami, ktoré začínajú zlomkami s rovnakými menovateľmi. Najmenší spoločný násobok týchto zlomkov bude zlomok, ktorého menovateľ je rovnaký ako menovateľ zlomkov. Čitateľ tohto zlomku dostaneme ako najmenší spoločný násobok čitateľov oboch zlomkov. Možnosť a) je špeciálny prípad pre menovateľ rovný 1.

Napr. riadky matice $\begin{pmatrix} 3/2 & 1 & 5/2 \\ 1/2 & 1 & 3/2 \end{pmatrix}$, najmenší spoločný násobok je 3/2

ekvivalentnou úpravou prevedieme
2. riadok * 3 mínus 1. riadok

na maticu $\begin{pmatrix} 3/2 & 1 & 5/2 \\ 0 & 2 & 2 \end{pmatrix}$

c) Hľadám ekvivalentnú úpravu medzi 2 riadkami, ktoré začínajú zlomkami s rovnakými čitateľmi. Spoločný násobok zlomkov, bude zlomok, ktorého čitateľ je rovnaký ako čitateľ zlomkov a menovateľ bude rovný 1.
Je to z toho dôvodu, že ak by som hľadal najmenší spoločný násobok týchto zlomkov, pri väčších hodnotách zlomkov by mohlo byt náročné takýto násobok nájsť a často to už nie je „pekné“ číslo, a teda by to nezodpovedalo ľudskému postupu počítania.

napr. $\begin{pmatrix} 5/4 & 1 & 9/4 \\ 5/6 & 1 & 11/6 \end{pmatrix}$, najmenší spoločný násobok bude 5.

ekvivalentnou úpravou sa prevedie
2. riadok * 6 mínus 1. riadok * 4

na maticu $\begin{pmatrix} 5/4 & 1 & 9/4 \\ 0 & 2 & 2 \end{pmatrix}$

d) Vo zvyšných prípadoch môže byť nájdenie najmenšieho spoločného násobku netriviálne a následne násobky upravovaného a pomocného riadku môžu zdanlivo vyzerať absurdne. A navyše tento postup by často nepripomínal ľudský spôsob počítania. V takýchto prípadoch sa postupuje nasledovne. Upravovaný riadok sa prenásobí hodnotou prvého prvku pomocného riadku a pomocný riadok sa prenásobí hodnotou prvého prvku upravovaného riadku. Týmto sme dostali rovnaký prvý prvok u oboch riadkov, čiže ich môžeme následne odčítať.

8. Vytvoríme zoznam všetkých matíc, ktoré dokážeme vytvoriť prevedením nájdených ekvivalentných úprav jednotlivých riadkov matice

Vytvorím zoznam všetkých matíc, ktoré dokážem vytvoriť použitím rôznych ekvivalentných úprav jednotlivých riadkov matice. Počet týchto matíc závisí od počtu riadkov matice, ktoré sa musia upravovať.

Pre matice s väčším počtom riadkov, už je množstvo možných matíc, ktoré dokážeme vytvoriť, veľmi veľké, a preto treba postupne rušiť najnevhodnejšie ekvivalentné úpravy. Nájdene ekvivalentné úpravy pre každý riadok matice sa zoradia podľa ohodnotenia úpravy a tie najhoršie sa zrušia (počet úprav, ktoré sa zrušia, závisí na počte riadkov matice).
Toto obmedzovanie začína od matíc s počtom riadkov 7. Pre takúto maticu s nenulovými prvkami v prvom stĺpci by sa mohlo vytvoriť až takmer 30 tisíc rôznych matíc, čo by viedlo k obrovskej časovej zložitosti programu.
Pre maticu so siedmymi riadkami sa po tomto obmedzení necháva pre každý riadok maximálne 5 možných ekvivalentných úprav. Minimálne však ponechávam 2 ekvivalentné úpravy pre každý riadok matice.

Každá vytvorená matica prejde vyhodnocovacou funkciou, ktorá podľa určitých vlastnosti ohodnotí danú maticu (viď Ohodnotenie matice)

9,10,11. Výber najlepšej matice a príprava na ďalší krok Gaussovej eliminácie

Matice sa zoradia podľa celkového ohodnotenia, ktoré doposiaľ získali. Najlepšie ohodnotená matica pokračuje ďalším krokom Gaussovej eliminácie.

Pre zvolenú maticu platí:
- má pevne zvolený prvý riadok, ktorý sa už v ďalšom priebehu eliminácie nebude meniť
- prvý stĺpec obsahuje pivota, tento stĺpec sa už v ďalšom priebehu eliminácie nebude meniť
Keďže v ďalšom priebehu eliminácie prvý riadok a prvý stĺpec matice sa už nebude meniť, môžem ich „odstrániť". Tým sa matica zmenší a aj zjednoduší.
Tým odpadá potreba používania indexov na určovanie riadku a stĺpca, s ktorými sa aktuálne pracuje. Na zmenšenú maticu rozmerov (m-1) x (n-1) opäť zavolám Gaussovú elimináciu.
Tato rekurzia skončí, keď má matica len 1 riadok. Ten už nie je možné ďalej upravovať.

Ohodnotenie ekvivalentných úprav

Riadky matice sa dajú upravovať rôznymi ekvivalentnými úpravami. Problém je v tom, že počet týchto úprav býva často pomerne veľký, a preto je potrebné tieto úpravy rozlišovať. Niektoré sú jednoduchšie na prevedenie ako iné, prípadne prvky matice po prevedení jednej úpravy budú menšie ako pri prevedení inej úpravy.

Z tohto dôvodu každá ekvivalentná úprava matice, ktorá sa nájde, získa hodnotu vyjadrujúcu výhodnosť jej použitia. Čím nižšia výsledná hodnota je, tým výhodnejšie je použiť danú ekvivalentnú úpravu.
Ak niektorá úprava získala vysoké ohodnotenie, ešte to neznamená, že nebude použitá v eliminácii. Po prevedení nájdených ekvivalentných úprav, ešte dochádza k ohodnoteniu celkovej matice. Obe tieto hodnotenia sa spolu sčítavajú, takže až vtedy sa skutočne preukáže výhodnosť použitia nájdených ekvivalentných úprav.

Pri maticiach s väčším počtom riadkov (7 a viac) býva veľký počet matíc, ktoré môžeme dostať použitím rôznych ekvivalentných úprav. Preto sa úpravy pre každý riadok zotriedia podľa ohodnotenia a najhoršie sa rušia. Počet úprav, ktoré sa zrušia závisí od veľkosti matice.

Niekoľko pozorovaní pri počítaní ľudským spôsobom:
- upravujeme hlavne dvojice riadkov, ktoré majú rovnaké prvé prvky, takéto riadky sa jednoducho odčítajú
- všímame si počet núl, ktoré nám po prevedení danej úpravy vzniknú. Najvýhodnejšie nuly, sú tie, ktoré vznikajú na začiatku riadkov. (mimo prvku, ktorý sme nulovali)
- upravujeme dvojice riadkov, u ktorých najmenší spoločný násobok prvých prvkov sa dá jednoducho spočítať
- vyhýbame sa úpravám, pri ktorých treba násobiť prvky veľkými číslami
- vyhýbame sa použitiu ekvivalentných úprav, pri ktorých treba násobiť/deliť prvky zlomkom, pretože počítanie so zlomkami je pre človeka náročnejšie ako s celými číslami

Upravujme i-tý riadok pomocou j-tého riadku. Riadok, ktorý touto úpravou dostaneme označme r.

Program hodnotí:

a) nutnosť násobiť upravovaný riadok i
   ak sa upravovaný riadok pri svojej úprave musí násobiť celým číslom, zvyšuje sa hodnota úpravy o 4, ak sa násobí zlomkom zvýši sa hodnota dvojnásobne, navyše ak je čitateľ tohto zlomku rôzny od 1, zvyšuje sa ešte o 2.
b) nutnosť násobiť pomocný riadok j
   rovnaký postup ako v bode a), ale hodnotí sa pomocný riadok j
c) vznik zlomkov
   po prevedení danej úpravy sa skontrolujú prvky upravovaného riadku i a vzniknutého riadku r, ak vznikol zlomok na pozícii, kde predtým nebol zvyšuje sa hodnota úpravy o 1, naopak ak sa odstránil, znižuje sa hodnota úpravy o 1
d) vznik nulového prvku
   po prevedení úpravy sa skontrolujú prvky vzniknutého riadku r, každý nulový prvok, ktorý sa v tomto riadku vyskytuje, znižuje hodnotu úpravy o 1.
e) zväčšenie prvkov
   po prevedení úpravy sa skontrolujú prvky upravovaného riadku i a vzniknutého riadku r, ak sa zväčšil počet cifier prvkov na jednotlivých pozíciách riadkov zvyšuje sa hodnota úpravy o 1 za každú cifru navyše, naopak ak sa počet cifier znížil, znižuje sa hodnota úpravy o 1 za každú cifru

Ohodnotenie matice

Z nájdených ekvivalentných úprav jednotlivých riadkov matice sa ich skombinovaním vytvoria rôzne matice. Počet týchto matíc býva pomerne veľký, preto treba vybrať tie najvýhodnejšie. Z tohto dôvodu je každej matici priradená hodnota, ktorá predstavuje výhodnosť danej matice a teda použitých ekvivalentných úprav, ktorými sme ju vytvorili. Táto hodnota vznikne ako súčet hodnotení použitých ekvivalentných úprav.
Na matici sa prevedie ešte jedno ohodnotenie, ktoré už môže len zlepšiť celkové hodnotenie danej matice. Toto ohodnotenie vyjadruje výhodnosť použitia tejto matice do ďalšieho kroku Gaussovej eliminácie. Je zamerané hlavne na vlastnosti druhých prvkov riadkov, ktoré sa budú ďalej upravovať.

Hodnotím všetky riadky okrem prvého, pretože tento sa už v ďalšom priebehu eliminácie nebude meniť. Tieto riadky už majú nulový prvý prvok, preto sa pozerám na druhý prvok, prípadne ďalšie.

Hodnotí sa:

a) Ak ma riadok druhý a prípadne ďalšie prvky rovné 0, znížim za každú nulu hodnotu o 8. Ekvivalentná úprava, ktorá vytvorila tento riadok je jedna z najvýhodnejších a takmer isto sa bude vyskytovať vo vybranej matici, ktorou ďalej pokračuje eliminácia. Pretože každá vzniknutá nula znamená, že sa v ďalšom kroku eliminácie tento riadok nespracúva, a teda sa skracuje celková eliminácia.
b) Ak riadok nemá druhý prvok nulový, zisťujem, či zvyšné riadky matice od tohto dole, nezačínajú takým istým prvkom v druhom stĺpci. V takom prípade sa znižuje hodnota matice o 4. Za každý ďalší takýto riadok s rovnakým prvkom sa znižuje hodnota o dvojnásobok.
c) Ak nie sú splnené predchádzajúce 2 možnosti, tak zisťujem, či zvyšné riadky od tohto dole, nezačínajú prvkom v druhom stĺpci, ktorý je násobkom prvku skúmaného riadku. V takom prípade sa znižuje hodnota o 2, za každý takýto riadok.

Ukážkový príklad vyriešený pomocou priloženého programu

Matica bola náhodne vygenerovaná programom.

a=

1/2	1	1/3	1/4	2
3/2	1/2	4	1/2	3
4/3	1/3	3	2/4	5
5/2	1/2	1	7	11

Gauss [a]

4. riadok minus 2. riadok
zamena 4. riadku a 1. riadku
pom=

1	0	-3	13/2	8
3/2	1/2	4	1/2	3
4/3	1/3	3	1/2	5
1/2	1	1/3	1/4	2

2. riadok minus 4. riadok * 3
3. riadok * 3 minus 1. riadok * 4
4. riadok * 2 minus 1. riadok
pom=

1	0	-3	13/2	8
0	-5/2	3	-1/4	-3
0	1	21	-49/2	-17
0	2	11/3	-6	-4

zamena 3. riadku a 2. riadku
pom=

1	0	-3	13/2	8
0	1	21	-49/2	-17
0	-5/2	3	-1/4	-3
0	2	11/3	-6	-4

3. riadok * 4 plus 4. riadok * 5
4. riadok minus 2. riadok * 2
pom=

1	0	-3	13/2	8
0	1	21	-49/2	-17
0	0	91/3	-31	-32
0	0	-115/3	43	30

4. riadok * 91 plus 3. riadok * 115
pom=

1	0	-3	13/2	8
0	1	21	-49/2	-17
0	0	91/3	-31	-32
0	0	0	348	-950

4. riadok deleno 2
pom=

1	0	-3	13/2	8
0	1	21	-49/2	-17
0	0	91/3	-31	-32
0	0	0	174	-475

V prvom kroku Gaussovej eliminácie sa každý riadok môže upraviť pomocou troch zvyšných riadkov matice. Z týchto rôznych ekvivalentných sa vytvorí 18 rôznych matíc od najlepšieho ohodnotenia 10 (táto matica sa použila) až po najhoršie 52. Je vidieť rozdiel v náročnosti použitých ekvivalentných úprav a z toho vyplýva aj rozdielny počet zlomkov.

najlepšie ohodnotená

2. riadok minus 4. riadok * 3
3. riadok * 3 minus 1. riadok * 4
4. riadok * 2 minus 1. riadok

pom=

1	0	-3	13/2	8
0	-5/2	3	-1/4	-3
0	1	21	-49/2	-17
0	2	11/3	-6	-4

najhoršie ohodnotená

2. riadok * 2 minus 1. riadok * 3
3. riadok * 3/2 minus 2. riadok * 4/3
4. riadok * 4/3 minus 3. riadok * 1/2

pom=

1	0	-3	13/2	8
0	1	17	-37/2	-18
0	-1/6	-5/6	1/12	7/2
0	7/6	-19/18	1/12	1/6

Gauss-Jordanová eliminačná metóda

Táto operácia spočíva vtom, že Gaussovou eliminačnou metódou prevedieme maticu na hornú trojuholníkovu maticu. Túto maticu ďalej pomocou ekvivalentných úprav prevedieme až na diagonálny tvar.

Ak má matica počet riadkov rovný alebo vyšší ako počet premenných, z posledného stĺpca matice priamo zistíme hodnoty riešenia matice.

Výsledná matica je jednoznačná, použitie rôznych ekvivalentných úprav ju nemení. Maticu upravíme na odstupňovaný tvar, v tomto tvare už nie je príliš možnosti ekvivalentných úprav, pomocou ktorých prevedieme maticu na diagonálny tvar. Ukázalo sa výhodne previesť pivoty matice na 1. Následne nulovanie prvkov nad diagonálou matice sa prevádza pomocou násobkov „jednotky" (násobkov riadkov, ktoré majú v danom stĺpci pivot rovný 1)

Tento postup riešenia matíc sa využíva pri hľadaní inverznej matice.

Ukážkový príklad vyriešený pomocou priloženého programu

Matica bola náhodne vygenerovaná programom.

```
a=
 3   0  -3  -1  -5
 1  -2  -1  -5   4
-4   0   5   0   5
-2   0   5   3  -4

GaussJordan [a]
zamena 2. riadku a 1. riadku
pom=
 1  -2  -1  -5   4
 3   0  -3  -1  -5
-4   0   5   0   5
-2   0   5   3  -4

2. riadok * 2 plus 4. riadok * 3
3. riadok * 3 plus 2. riadok * 4
4. riadok * 2 minus 3. riadok
pom=
 1  -2  -1  -5    4
 0   0   9   7  -22
 0   0   3  -4   -5
 0   0   5   6  -13

4. riadok * 2 minus 2. riadok
pom=
 1  -2  -1  -5    4
 0   0   1   5   -4
 0   0   3  -4   -5
 0   0   5   6  -13

3. riadok minus 2. riadok * 3
4. riadok minus 2. riadok * 5
pom=
 1  -2  -1   -5   4
 0   0   1    5  -4
 0   0   0  -19   7
 0   0   0  -19   7

4. riadok minus 3. riadok
pom=
 1  -2  -1   -5   4
 0   0   1    5  -4
 0   0   0  -19   7
 0   0   0    0   0

Nastavenie pivotov na 1
3. riadok deleno -19
pom=
 1  -2  -1  -5   4
 0   0   1   5  -4
 0   0   0   1  -7/19
 0   0   0   0   0

1. riadok minus 2. riadok * -1
2. riadok minus 3. riadok * 5
pom=
 1  -2   0   0    0
 0   0   1   0  -41/19
 0   0   0   1   -7/19
 0   0   0   0    0
```

Inverzná matica

Definícia

Nech A je štvorcová matica rádu n. Pokiaľ existuje matica B taká, že súčin matíc A*B = I_n (jednotková matica), potom matica B sa nazýva inverznou maticou k matici A. Pokiaľ existuje inverzná matica k matici A, nazýva sa matica A regulárnou, v opačnom prípade singulárnou.

Postup hľadania inverznej matice A^{-1}

- rozšírime maticu A o jednotkovú maticu I_n $(A|\ I_n)$
- pokúsime sa previesť túto rozšírenú maticu ekvivalentnými úpravami na tvar $(I_n|A^{-1})$ (pomocou Gauss-Jordanovej eliminačnej metódy)
- ak sa nám podarí splniť bod 2, rozšírená časť matice priamo tvorí inverznú maticu A^{-1}

Ukážkový príklad vyriešený pomocou priloženého programu

z=

1	2	3
0	4	3
1	4	4

Inverse [z]

Rozšírenie matice o jednotkovu maticu

pom=

1	2	3	1	0	0
0	4	3	0	1	0
1	4	4	0	0	1

zamena riadkov matice podla poctu nul

pom=

1	2	3	1	0	0
1	4	4	0	0	1
0	4	3	0	1	0

2. riadok minus 1. riadok

pom=

1	2	3	1	0	0
0	2	1	-1	0	1
0	4	3	0	1	0

3. riadok minus 2. riadok * 2

pom=

1	2	3	1	0	0
0	2	1	-1	0	1
0	0	1	2	1	-2

Nastavenie pivotov na 1

2. riadok deleno 2

pom=

1	2	3	1	0	0
0	1	1/2	-1/2	0	1/2
0	0	1	2	1	-2

1. riadok minus 2. riadok * 2
2. riadok minus 3. riadok * 1/2

pom=

1	0	2	2	0	-1
0	1	0	-3/2	-1/2	3/2
0	0	1	2	1	-2

1. riadok minus 3. riadok * 2

pom=

1	0	0	-2	-2	3
0	1	0	-3/2	-1/2	3/2
0	0	1	2	1	-2

pom=

-2	-2	3
-3/2	-1/2	3/2
2	1	-2

Determinant

Definícia

Nech A je štvorcová matica rádu n. Determinant det(A) matice A definujem rovnosťou Det(A) = $\sum_{\pi \in S_n} \operatorname{sgn}(\pi) a_{1,\pi(1)} a_{2\pi(2)} \ldots a_{n,\pi(n)}$, kde sa sčítava cez všetky permutácie π množiny {1, 2, … n}. V každom z týchto súčinov vystupuje ako činiteľ práve jediný prvok z každého stĺpca a práve jediný prvok z každého riadku matice A.

- sgn(π) znamená znamienko permutácie π. Znamienko permutácie je vždy +1 alebo -1.

Definuje sa sgn(π) = $\prod_{1 \le i < j \le n} \frac{\pi(j) - \pi(i)}{j - i}$

Vlastnosti determinantov

- determinant matice, ktorá má nulový riadok (stĺpec) je rovný 0
- determinant hornej (dolnej) trojuholníkovej matice je rovný súčinu všetkých prvkov z hlavnej diagonály matice
- („elementárna riadková operácia I“) ak vynásobíme všetky prvky niektorého riadku (stĺpca) matice A číslom c, determinant vzniknutej matice je rovný c*det(A). Špeciálne, ak má matica nulový riadok (stĺpec), potom det(A) = 0.
- („elementárna riadková operácia II“) determinant sa nezmení, ak pripočítame k niektorému riadku (stĺpcu) matice A c-násobok niektorého riadku matice (stĺpca). Determinant vzniknutej matice je rovný determinantu pôvodnej matice A.
- prehodenie dvoch riadkov (stĺpcov) matice mení znamienko determinantu
- determinant matice A je rôzny od 0, nazývame maticu A regulárnou, v opačnom prípade nazývame maticu A singulárnou
- („rozvoj determinantu podľa riadku“) pre každý index i platí vzťah $\det(A) = \sum_{j=1}^{n} (-1)^{i+j} \det(A_{ij})$, kde A_{ij} je matica, ktorá vznikne z A vypustením i-tého riadku a j-tého stĺpca

Program počíta determinant matice dvomi spôsobmi

1. matica sa prevedie na hornú trojuholníkovu maticu Gaussovou eliminačnou metódou
    - spočíta sa súčin všetkých prvkov na hlavnej diagonále
    - pri každej zámene dvoch riadkov matice sa mení znamienko determinantu
    - ak sa niektorý riadok matice pri svojej úprave násobil číslom c, výsledný determinant sa delí číslom c

2. LaPlaceov rozvoj
- rozvoj matice sa robí podľa riadku (stĺpca), ktorý obsahuje najväčší počet núl
- znamienko subdeterminantu určíme podľa vzorca $(-1)^{i+j}\det(A_{ij})$
- táto metóda výpočtu determinantu je časovo veľmi náročná, preto je možné touto metódou počítať determinant matíc s maximálne šiestimi riadkami, v opačnom prípade sa hodnota determinantu počíta prvým spôsobom

Metóda, ktorou sa vypočíta hodnota determinantu matice sa volí v programe v ponuke Nastavenia -> Možnosti -> Determinant

Ukážkový príklad vyriešený pomocou priloženého programu

```
c=
 1   2   3   1
 0   4   3  -1
 1   3   2   1
 2   0   1   1

rozvoj podla 1. prvku 2. riadku = 0
0 * Det[pom] = 0
pom=
 2   3   1
 3   2   1
 0   1   1

rozvoj podla 2. prvku 2. riadku = 4
4 * Det[pom] = 4
pom=
 1   3   1
 1   2   1
 2   1   1

rozvoj podla 1. prvku 1. riadku = 1
1 * Det[pom] = 1
pom=
 2   1
 1   1

rozvoj podla 2. prvku 1. riadku = 3
-3 * Det[pom] = 3
pom=
 1   1
 2   1

rozvoj podla 3. prvku 1. riadku = 1
1 * Det[pom] = -3
pom=
 1   2
 2   1

rozvoj podla 3. prvku 2. riadku = 3
-3 * Det[pom] = 3
pom=
 1   2   1
 1   3   1
 2   0   1

rozvoj podla 1. prvku 3. riadku = 2
2 * Det[pom] = -2
pom=
 2   1
 3   1

rozvoj podla 2. prvku 3. riadku = 0
0 * Det[pom] = 0
pom=
 1   1
 1   1

rozvoj podla 3. prvku 3. riadku = 1
1 * Det[pom] = 1
pom=
 1   2
 1   3

rozvoj podla 4. prvku 2. riadku = -1
-1 * Det[pom] = 9
pom=
 1   2   3
 1   3   2
 2   0   1

rozvoj podla 1. prvku 3. riadku = 2
2 * Det[pom] = -10
pom=
 2   3
 3   2

rozvoj podla 2. prvku 3. riadku = 0
0 * Det[pom] = 0
pom=
 1   3
 1   2

rozvoj podla 3. prvku 3. riadku = 1
1 * Det[pom] = 1
pom=
 1   2
 1   3


Det[c] = 16
```

KAPITOLA 3

Program

V tejto kapitole si predstavíme priložený program.

Tento program bol vytvorený ako súčasť ročníkového projektu a bakalárskej práce na Matematicko – Fyzikálnej fakulte Univerzity Karlovej v Prahe. Vedúci tejto práce bol RNDr. Ondřej Pangrác, Ph.D.

Implementuje algoritmy lineárnej algebry spôsobom pripomínajúcim ľudský postup počítania s maticami.

Minimálne systémové požiadavky
Tento program funguje na operačnom systéme Windows, s nainštalovaným Microsoft .NET Framework 2.0. Miesto na pevnom disku : približne 1 MB.

Inštalácia a spustenie programu

1. Vložte CD do mechaniky
2. Skopírujte si na Váš pevný disk priečinok Matice
3. Otvorte skopírovaný priečinok
4. Spustite matice.exe

Užívateľské rozhranie programu
Hlavné okno programu je rozdelené na dve časti. Vľavo sa zapisuje zadanie matíc a operácie, ktoré sa majú previesť na maticiach. Vpravo sa zobrazujú výsledky. Rozdelenia okna nie je pevne určené, a teda sa dá meniť pomer šírky vstupnej časti k výsledkovej časti okna.

Vo vstupnej časti okna sa zároveň určuje spôsob výpisu výsledkov prevedených operácii s maticami. Výpis ovplyvňujeme pomocou použitia znaku >:

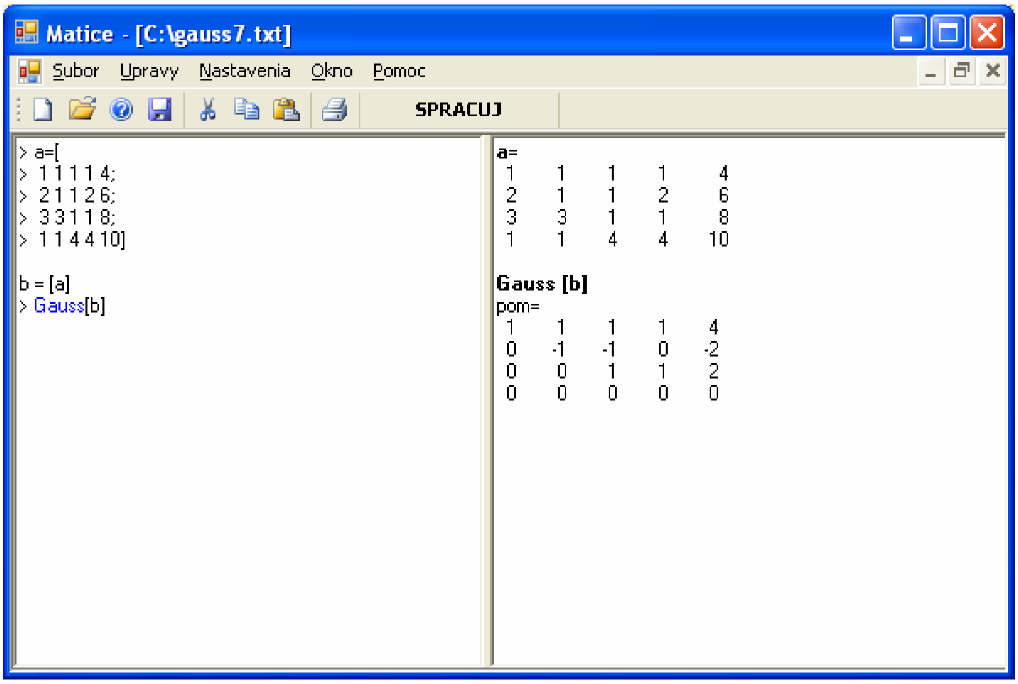
- ak napíšeme pred názov operácie znak >, vypíše sa výsledná matica po prevedení danej maticovej operácie (napr. po zadaní „> Gauss[a]“ sa vypíše výsledná matica po prevedení Gaussovej eliminácie)
- ak napíšeme pred názov operácie dva znaky >>, vypíše sa celý priebeh výpočtu danej maticovej operácie (napr. po zadaní „>> Inverse[a]“ sa vypíše celý postup výpočtu inverznej matice)
- ak pred názov operácie nenapíšeme znak >, daná operácia s maticou sa prevedie na pozadí ale na výstup nič sa nevypíše

Zvýrazňovanie textu

- vo vstupnej časti okna sa po zadaní maticovej operácie farebne zvýrazňuje text zadanej operácie, tým sa jednak sprehľadnia zadané dáta a zároveň je to výborná kontrola správnosti zadania danej maticovej operácie
- vo výsledkovej časti okna sa zvýrazňujú zadania jednotlivých maticových operácií, tým sa jednoducho rozlíši, ku ktorej maticovej operácii vypísaný text patrí

Program umožňuje mať súčasne otvorené viaceré okná v hlavnom okne programu, čiže môžeme naraz pracovať s viacerými dokumentmi.

Ukážka spusteného programu



Obr. 3.1

V položke menu Nastavenia -> Možnosti sa dajú nastaviť vlastnosti pre nasledujúce maticové operácie:

1. determinant, volí sa metóda výpočtu hodnoty determinanta matice:
- pomocou Gaussovej eliminačnej metódy
- Laplaceovým rozvojom

2. funkcia random, generuje náhodne prvky matice
- volí sa interval čísiel, z ktorého sa generujú prvky matice

Literatúra

[1] Jindřich Bečvář : Lineární algebra
3. vydání, MATFYZPRESS, Praha, 2005

[2] Jiří Matoušek, Jaroslav Nešetřil : Kapitoly z diskrétní matematiky,
2. vydání, Nakladatelství Karolinum, Praha, 2003

[3] Cormen, Leiserson, Rivest, Stein : Introduction to algorithms
2nd Edition, Mc Graw Hill 2001