Aplikace matematiky

# Algoritmy. 1. Matrixinvert. 2. Matrixperm. 3. Gauss

*Aplikace matematiky*, Vol. 12 (1967), No. 1, 81--85,86

Persistent URL: http://dml.cz/dmlcz/103070

**Terms of use:**

© Institute of Mathematics AS CR, 1967

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.

This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* http://dml.cz

# ALGORITMY

Tímto ročníkem počínaje zřizujeme v časopise Aplikace matematiky rubriku ALGORITMY*), v níž budeme publikovat algoritmy zapsané v ALGOLu podobně jako se publikují např. v časopise CACM. Nebudeme v ní uveřejňovat algoritmy určené k řešení nějaké konkrétní úlohy a použitelné jen pro ni (např. pro výpočet *tabulek* nějaké speciální funkce), nýbrž algoritmy dílčích úloh (např. algoritmy pro jednotlivé numerické metody, pro výpočet speciálních funkcí) resp. i algoritmy pro řešení technických úloh dostatečně obecného charakteru.

Na rozdíl od příspěvků tohoto druhu v CACM, kde bývají publikovány i algoritmy s minimálními vysvětlivkami nebo i bez nich, budou algoritmy v této rubrice doprovázeny velmi stručným popisem (nikoliv odvozením) použité metody (třebas i jen pomocí vysvětlivek vložených do programu). Jen ve výjimečných případech může být tento popis obsáhlejší než sám algoritmus. Do této rubriky budou přijímány jen algoritmy, které jsou nejen vyzkoušené, nýbrž i dostatečně ověřené v praxi při řešení většího počtu úloh a to pro řešení takových úloh, pro něž dosud algoritmy v ALGOLu publikovány nebyly, nebo algoritmy, které jsou v nějakém směru dokonalejší než algoritmy dosud publikované. Redakce si vyhražuje právo přetiskovat v této rubrice osvědčené algoritmy, které byly již publikovány jinde.

Každý příspěvek (výhradně v jazyce českém) se skládá z těchto částí:

1. Název algoritmu (krátký a výstižný). Obsáhlejší nadpis, charakterizující algoritmus přesněji.

2. Jméno a adresa pracoviště autora.

3. Pokud nejde o metodu všeobecně známou velmi stručný popis užité metody, v každém případě však odkaz na literaturu. (Redakce si vyhražuje krácení.)

4. Algoritmus: Algoritmus musí být zapsán v revidovaném ALGOLu-60 (viz CACM, vol. 6, Jan. 1963), typografická úprava je obdobná jako v časopise CACM, polotučný tisk se modře zakroužkuje. (Redakce si vyhražuje zásady do typografické úpravy.) Autor popíše algoritmus s jasnými avšak stručnými vysvětlivkami ve tvaru deklarace procedury bez nelokálních identifikátorů a se všemi formálními parametry specifikovanými. Vypracování algoritmu musí být velmi pečlivé, bez syntaktických chyb. Algoritmus bude zpravidla při recenzi ověřen na počítači. Není však možné jej zkoušet a opravovat v něm chyby.

5. Stručné vyjádření jak jsou výsledky přesné, pro jaký obor úloh je algoritmus použitelný, popřípadě s jakou účinností (např. algoritmus pro výpočet speciální funkce „je velmi pomalý pro $x > 100$").

6. Kontrolní příklad: Pokud to charakter úlohy dovoluje a neklade to přílišné nároky na rozsah příspěvku, je třeba připojit i vstupní a výstupní údaje kontrolního příkladu voleného tak, aby napomáhal při ověření správné funkce algoritmu na stroji.

---

*) Zodpovědným členem redakční rady pro příspěvky, uveřejňované v této rubrice je Doc. dr. Jiří Raichl C.Sc. z Centra numerické matematiky MFF UK.

7. Jak byl algoritmus zkoušen a užíván: zda byl skutečně překládán překladačem z ALGOLu nebo původně naprogramován v jiném jazyku a ve kterém a do ALGOLu pouze přepsán.

8. Základní literatura o použité metodě, popřípadě i odkazy na publikované algoritmy, řešící tutéž nebo obdobnou úlohu.

Žádáme autory, aby styl svých příspěvků přizpůsobili stylu příspěvků publikovaných v této rubrice dříve.

Vedle těchto částí, které budou publikovány, nechť autor na zvláštním listě uvede jeden, popřípadě i několik zkušebních příkladů pro recenzenta, jestliže musil být bod 6 příspěvku pro přílišný rozsah těchto údajů vynechán, které by napomáhaly při ověřování algoritmu.

Později budou do této rubriky k programům publikovaným dříve přijímány také:

1. „Certifikace", tj. vyjádření, zda určitý algoritmus byl (nebo nebyl) s úspěchem použit resp. které tiskové (i jiné) chyby je třeba v něm opravit.

2. „Poznámky", tj. zhodnocení algoritmu po stránce oboru použitelnosti, rychlosti a přesnosti výsledků na základě jeho použití na větším počtu případů.

Rubrika ALGORITMY bude uváděna na konci příslušného čísla časopisu a bude začínat vždy na liché stránce tak, aby si čtenáři mohli tuto část časopisu oddělit.

## 1. MATRIXINVERT

## INVERSE MATICE JORDANOVOU METODOU

Algoritmus byl převzat z časopisu CACM [3]. Komentářem a kontrolním příkladem doplnil JIŘÍ RAICHL, Centrum numerické matematiky MFF UK, Praha 1, Malostranské nám. 25.

Tato procedura invertuje matici eliminací (Jordanovou metodou [1], [2]). Eliminuje vždy neznámou, u níž je v odpovídající soustavě v absolutní hodnotě největší koeficient (*pivot*, o indexech *pivi*, *pivj*. Hledáme jej jak v řádcích, tak ve sloupcích). Výměny sloupců a řádků, které tento algoritmus předpokládá, však neprovádíme skutečnými výměnami prvků matice, nýbrž je pouze registrujeme pomocí vektorů o složkách $r_i$, $c_i$. Na počátku eliminace položíme $r_i = c_i = i$ a místo výměny sloupců, resp. řádků pouze vyměňujeme hodnoty těchto složek. Po skončení eliminace procedura *MATRIXPERM* (její deklaraci viz níže) provede skutečné výměny prvků v řádcích a sloupcích invertované matice. Je-li při některém eliminačním kroku absolutní hodnotou největší koeficient menší než dané $\varepsilon$, pokládáme danou matici za singulární a výpočtový postup přerušíme.

**procedure** *MATRIXINVERT* (*a*, *n*, *eps*, *singular*);
**comment** *a* je pole odpovídající před použitím procedury dané matici, po jejím použití matici invertované, *n* je stupeň matice, *eps* odpovídá $\varepsilon$, *singular* je cílový výraz udávající, kde pokračovat, pokládáme-li danou matici za singulární;

```
value n, eps;
array a; integer n; real eps; label singular;
begin integer i, j, k, l, pivi, pivj, p; real pivot; integer array r, c [1 : n];
for i: = 1 step 1 until n do r[i]: = c[i]: = i;
pivi: = pivj: = 1;
for i: = 1 step 1 until n do for j: = 1 step 1 until n do
  if abs (a[i, j]) > abs (a[pivi, pivj]) then begin pivi: = i; pivj: = j end;
for i: = 1 step 1 until n do
  begin l: = r[i]; r[i]: = r[pivi]; r[pivi]: = l;
        l: = c[i]; c[i]: = c[pivj]; c[pivj]: = l;
        if eps > abs (a[r[i], c[i]]) then goto singular;
        for j: = n step − 1 until i + 1, i − 1 step − 1 until 1 do
          a[r[i], c[j]]: = a[r[i], c[j]]/a[r[i], c[i]];
        a[r[i], c[i]]: = 1/a[r[i], c[i]];
        pivot: = 0;
        for k: = 1 step 1 until i − 1, i + 1 step 1 until n do
          begin for j: = n step − 1 until i + 1, i − 1 step − 1 until 1 do
            begin a[r[k], c[j]]: = a[r[k], c[j]] − a[r[i], c[j]] × a[r[k], c[i]];
              if k > i ∧ j > i ∧ abs (a[r[k], c[j]]) > abs (pivot)
                then begin pivi: = k; pivj: = j; pivot: = a[r[k], c[j]] end
            end konec cyklu zpracovávajícího prvky v řádku;
            a[r[k], c[i]]: = −a[r[i], c[i]] × a[r[k], c[i]]
          end konec cyklu zpracovávajícího řádky
  end konec eliminačního kroku;
MATRIXPERM (a[j, p], a[k, p], j, k, r, c, n, p);
MATRIXPERM (a[p, j], a[p, k], j, k, c, r, n, p)
end
```

Kontrolní příklad:

$\varepsilon = 10^{-6}$

$$\boldsymbol{A} = \begin{Vmatrix} 1 & 1{,}2 & 1{,}3 \\ 1{,}2 & 1 & 0 \\ 1{,}3 & 0 & 1 \end{Vmatrix} \qquad \boldsymbol{A}^{-1} = \begin{Vmatrix} -0{,}4694835681 & 0{,}5633802817 & 0{,}6103286385 \\ 0{,}5633802817 & 0{,}3239436620 & -0{,}7323943662 \\ 0{,}6103286385 & -0{,}7323943662 & 0{,}2065727700 \end{Vmatrix}$$

Tento algoritmus byl užíván v systému ALGOL - GENIUS na počítači SAAB.


[1] *O. Slavíček a kol.*: Základní numerické metody. SNTL, Praha 1964.
[2] *D. K. Faddějev, V. N. Faddějeva*: Num. metody lin. algebry. SNTL, Praha 1964.
[3] *J. Boothroyd:* Algorithm 231. CACM 1964, vol. 7, str. 347.

## 2. MATRIXPERM

### ALGORITMUS PRO VÝMĚNU ŘÁDKŮ RESP. SLOUPCŮ V MATICI

Tento algoritmus je pomocnou procedurou pro proceduru *MATRIXINVERT* (viz výše). Byl převzat z časopisu CACM, 1964, vol. 7, str. 347, J. Boothroyd.

```
procedure MATRIXPERM (a, b, j, k, s, d, n, p);
comment a, b prvky řádků, resp. sloupců, které se mají vyměňovat, j, k indexy
  vyměňovaných sloupců resp. řádků, s, d vektory, udávající výměny, n stupeň
  matice, p řádkový resp. sloupcový index ve vyměňovaných sloupcích resp. řádcích;
value n;
real a, b; integer array s, d; integer j, k, n, p;
begin integer i, t; real w; integer array tag, loc [1 : n];
for i: = 1 step 1 until n do tag[i]: = loc[i]: = i;
for i: = 1 step 1 until n do
  begin t: = s[i]; j: = loc[t]; k: = d[i];
        if j ≠ k then begin for p: = 1 step 1 until n do
                                begin w: = a; a: = b; b: = w end;
                            tag[j]: = tag[k]; tag[k]: = t;
                            loc[t]: = loc[tag[j]]; loc[tag[j]]: = j
end end                end
```

## 3. GAUSS

### ROZŠÍŘENÁ GAUSSOVA METODA

JAROSLAV MORÁVEK, Matematický ústav ČSAV, Praha 1, Žitná 25
VLADIMÍR ŠPIRKO, Vysoká škola chemicko-technologická, Praha 6, Technická 1905

Procedura GAUSS umožňuje řešit maticovou rovnici tvaru $\boldsymbol{AX} = \boldsymbol{B}$ (k rozměrům matic $\boldsymbol{A}$, $\boldsymbol{X}$, $\boldsymbol{B}$ viz comment). Procedura GAUSS speciálně umožňuje rozhodnout, zdali maticová rovnice $\boldsymbol{AX} = \boldsymbol{B}$ je řešitelná, invertovat matici a určit jisté číslo DET (viz procedura), které v případě čtvercové regulární matice je determinantem této matice; tímto způsobem lze již též snadno určit matici adjungovanou k dané regulární matici.

```
procedure GAUSS (A, B, m, na, nb, DET, SINGULAR, EPS);
value m, na, nb, EPS; array A, B; integer m, na, nb;
real DET, EPS; label SINGULAR;
comment GAUSS transformuje uspořádanou dvojici matic A[m × n_a], B[m × n_b]
na dvojici A'[m × n_a], B'[m × n_b]. Používá se přitom elementárních řádkových
```

úprav matice ($\boldsymbol{A}$, $\boldsymbol{B}$) a vybírá se řádka s vedoucím (maximálním v absolutní hodnotě) elementem. Matici – řešení $\boldsymbol{X}(n_a \times n_b)$ dostaneme po skončení výpočtu na místě prvních $n_a$ řádků matice $\boldsymbol{B}$, tj. v poli $\boldsymbol{B}[1 : n_a, 1 : n_b]$. Na místě zbývajících řádků matice $\boldsymbol{B}$, tj. v poli $\boldsymbol{B}[n_a + 1 : m, 1 : n_b]$ budou v případě řešitelné maticové rovnice nuly (ve smyslu přesnosti použitého počítače).

```
begin integer i, j, k, piv; real t, g;
DET: = 1;
  for k: = 1 step 1 until na do
      begin
            piv: = k;
            for i: = k + 1 step 1 until m do
            if abs(a[i, k]) > abs(a[piv, k]) then piv: = i;
            if abs(a[piv, k]) < EPS then go to SINGULAR;
            q: = a[piv, k]; DET: = if piv = k then DET × q else – DET × q;
            for j: = k step 1 until na do
            begin
                  t: = a[k, j]; a[k, j]: = a[piv, j];
                  a[piv, j]: = t; a[k, j]: = a[k, j]/q
            end;
            for j: = 1 step 1 until nb do
            begin
                  t: = b[k, j]; b[k, j]: = b[piv, j];
                  b[piv, j]: = t; b[k, j]: = b[k, j]/q
            end;
            for i: = 1 step 1 until m do
            if i ≠ k then
            begin
                  for j: = k + 1 step 1 until na do
                  a[i, j]: = a[i, j] – a[i, k] × a[k, j];
                  for j: = 1 step 1 until nb do
                  b[i, j]: = b[i, j] – a[i, k] × b[k, j];
            end of Gauss arithmetic;
      end;
end of GAUSS;
```

Numerický příklad [1]:

$$\boldsymbol{A} = \begin{pmatrix} 3 & 5 & 1 & 0 \\ 2 & 1 & 4 & 5 \\ 1 & 7 & 4 & 2 \\ -3 & 5 & 1 & 1 \end{pmatrix}, \quad \boldsymbol{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\boldsymbol{X} = \boldsymbol{A}^{-1}:$$

$$\boldsymbol{A}^{-1} = \begin{pmatrix} 2{,}116 \times 10^{-1}, & 5{,}394 \times 10^{-2}, & -7{,}884 \times 10^{-2}, & -1{,}120 \times 10^{-1} \\ 1{,}701 \times 10^{-1}, & 4{,}149 \times 10^{-3}, & -8{,}229 \times 10^{-2}, & 1{,}452 \times 10^{-1} \\ -4{,}855 \times 10^{-1}, & -1{,}826 \times 10^{-1}, & 6{,}515 \times 10^{-1}, & -3{,}900 \times 10^{-1} \\ 2{,}697 \times 10^{-1}, & 3{,}237 \times 10^{-1}, & -4{,}730 \times 10^{-1}, & 3{,}278 \times 10^{-1} \end{pmatrix}$$

$$DET(\boldsymbol{A}) = -2{,}410 \times 10^{2}\,.$$

Procedura byla napsána v jazyce ALGOL ELLIOTT. Výpočty se prováděly na samočinných počítačích NATIONAL ELLIOTT 803 B a ELLIOTT 503. Algoritmu bylo použito k řešení systémů lineárních algebraických rovnic, které vznikají při analýze spekter N.M.R.[2].


[1] *В. И. Смирнов:* Курс высшей математики, т. III., 1. часть, стр. 22.

[2] *V. Špirko, J. Morávek*: Analysis of NMR spectra by differentiation with respect to a parameter. Coll. Czech. Chem. Comm. Vol. 31, p. 4057 (1966).