

Aplikace matematiky

Josef Čermák

Algoritmy. 27. PSQRT. Řešení soustavy rovnic se symetrickou pozitivně definitní $(2m + 1)$ diagonální maticí

Aplikace matematiky, Vol. 17 (1972), No. 4, 321–324

Persistent URL: <http://dml.cz/dmlcz/103422>

Terms of use:

© Institute of Mathematics AS CR, 1972

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



This document 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

27. psqrt

ŘEŠENÍ SOUSTAVY ROVNIC SE SYMETRICKOU POZITIVNĚ DEFINITNÍ
($2m + 1$) DIAGONÁLNÍ MATICÍ

JOSEF ČERMÁK, CSc., VŠChT Pardubice

procedure *psqrt* (*n*, *m*, *p*, *a*, *signal*) result: (*y*);
value *m*, *n*, *p*; **integer** *m*, *n*, *p*; **real procedure** *a*;
comment *n* je počet rovnic, *m* šíře pásu nenulových koeficientů vedle hlavní diagonály, *p* počet vektorů pravých stran, *a* funkční procedura pro určení nenulových hodnot prvků $a(i, j)$ matice v pásu, *signal* návěstí, od kterého pokračuje program v případě, že matice soustavy není pozitivně definitní, $y[1 : n, 1 : p]$ je matice výsledku;
label *signal*; **array** *y*;

begin

real array *c* [$1 : n - m, 0 : m$], *cc* [$0 : m \times (m + 1)/2$];
integer *i*, *j*, *k*, *nm*, *ki*, *m2*, *knm*, *km*;
real *s*, *q*; *nm* := $n - m$; *m2* := $m \times 2 + 2$;
for *k* := 1 **step** 1 **until** *nm* **do**
 begin *s* := $a(k, k)$;
 for *i* := 1 **step** 1 **until** *m* **do**
 begin *ki* := $k - i$; **if** *ki* > 0 **then** *s* := $s - c[ki, i] \uparrow 2$ **end**;
 if *s* < 0 **then go to** *signal*; *s* := $\text{sqrt}(s)$; $c[k, 0] := s$;
 for *j* := 1 **step** 1 **until** *m* **do**
 begin *q* := $a(k, j + k)$;
 for *i* := 1 **step** 1 **until** $m - j$ **do**
 begin *ki* := $k - i$; **if** *ki* > 0
 then *q* := $q - c[ki, i] \times c[ki, j + i]$
 end;
 $c[k, j] := q/s$
 end;
 LI: **for** *j* := 1 **step** 1 **until** *p* **do**
 begin *q* := $a(k, n + j)$;

```

    for i := 1 step 1 until m do
        begin ki := k - i; if ki > 0 then
            q := q - c[ki, i] × y[ki, j]
        end;
        y[k, j] := q/s
    end
end;
for k := 1 step 1 until m do
    begin integer k1, k1i;
        knm := k + nm; km := m2 - k; k1 := k - 1;
        s := a(knm, knm);
        for i := 1 step 1 until m do if knm > i then
            s := s - (if k ≤ i then c[knm - i, i] ↑ 2 else cc[((k1 - i) × (km + i))
                ÷ 2 + i] ↑ 2);
            if s ≤ 0 then go to signal;
            s := sqrt(s); cc[(km × k1) ÷ 2] := s;
            for j := 1 step 1 until m - k do
                begin q := a(knm, knm + j);
                    for i := 1 step 1 until m - j do
                        begin ki := knm - i; if ki > 0 then
                            begin if k ≤ i then
                                q := q - c[ki, i] × c[ki, j + i] else
                                    begin k1i := ((k1 - i) × (km + i)) ÷ 2 + i;
                                        q := q - cc[k1i] × cc[k1i + j]
                                    end
                                end
                            end
                        end;
                    cc[(km × k1) ÷ 2 + j] := q/s
                end;
            for j := 1 step 1 until p do
                begin q := a(knm, n + j);
                    for i := 1 step 1 until m do
                        begin ki := knm - i; if ki > 0 then
                            q := q - (if k ≤ i then c[ki, i] else cc[((k1 - i) × (km + i))
                                ÷ 2 + i]) × y[ki, j]
                        end;
                    y[knm, j] := q/s
                end
            end;
        for k := m step -1 until 1 do
            begin knm := k + nm;
                km := ((k - 1) × (m2 - k)) ÷ 2;

```

```

L3:   for j := 1 step 1 until p do
       begin s := y[knm, j];
           for i := 1 step 1 until m - k do s := s - y[knm + i, j] × cc[km + i];
               y[knm, j] := s/cc[km]
           end j
       end k;
   for k := nm step -1 until 1 do
L4:   for j := 1 step 1 until p do
       begin s := y[k, j];
           for i := 1 step 1 until m do s := s - y[k + i, j] × c[k, i];
               y[k, j] := s/c[k, 0]
           end
       end
   end
end

```

Metoda předpokládá čtení, případně postupný výpočet hodnot prvků na hlavní diagonále a prvků v pásu šíře m nad diagonálou. Do paměti počítače se ukládá místo n^2 prvků matice jen neúplný pás šíře $m + 1$ odpovídající horní trojúhelníkové matici tj. $(n - m) \cdot (m + 1) + m \cdot (m + 1)/2$. Jedná se o vhodnou aplikaci metody odmocnin [1]. Publikovaná procedura [2] pro řešení soustav s pásovými symetrickými pozitivně definitními maticemi předpokládá před vstupem umístění koeficientů soustavy v matici $[1 : n, 0 : m]$. Předpokládaná verze klade menší nárok na kapacitu paměti, navíc však umožňuje hodnoty prvků matice postupně číst (je-li jádrem funkční procedury a pouhé čtení) nebo počítat. Formální parametry i, j procedury a mají běžný význam $i \dots$ řádkového indexu, $j \dots$ sloupcového. Hodnotu i -tého prvku k -tého vektoru pravých stran určí program vstupem do funkční procedury s parametry $i, n + k$.

Kontrolní příklad:

Je dána soustava rovnic

$$\begin{aligned}
 (*) \quad & 5x_1 + 3x_2 + 2x_3 & & = 17 \\
 & 3x_1 + 3x_2 + x_3 + 2x_4 & & = 20 \\
 & 2x_1 + x_2 + 10x_3 - 3x_4 + x_5 & = & 27 \\
 & 2x_2 - 3x_3 + 5x_4 + 4x_5 & = & 35 \\
 & x_3 + 4x_4 + 25x_5 & = & 144
 \end{aligned}$$

Funkční procedura programu pro řešení tohoto příkladu spočívá v přechtení jednoho čísla na pásce dat a dosazení za $a(i, j)$.

Páska dat obsahuje tyto hodnoty:

- 5 (n) počet rovnic
- 2 (m) poloviční počet nenulových vedlejších diagonál
- 1 (p) počet vektorů pravých stran.

Prvky pásu a pravé strany soustavy (*)

$$a(i, j) \begin{cases} 5 & 3 & 2 & 17 \\ 3 & 1 & 2 & 20 \\ 10 & -3 & 1 & 27 \\ & 5 & 4 & 35 \\ & & 25 & 144 \end{cases}$$

Výsledek: $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, $x_4 = 4$, $x_5 = 5$.

Při pravidelném využití jen pro matice s jedním vektorem pravé strany je možno algoritmus zjednodušit deklarací y jako jednorozměrného pole, dále je třeba vyloučit instrukce cyklu po návěstích $L1$, $L2$, $L3$, $L4$ s příslušnými závorkami **begin** a **end**. Uvnitř těchto závorek je třeba zrušit druhý index pole y a nahradit druhý parametr funkční procedury a po návěstích $L1$, $L2$ výrazem $n + 1$.

Literatura

- [1] Д. К. Фадеев: Вычислительные методы линейной алгебры. ФИЗМАТГИЗ, Москва 1963, Ленинград.
- [2] H. Rutishauer: Computing 1 (1966), 77–79.