Josef Čermák

Use of the method of factorization for solving differential equations in a rectangular region.

Aplikace matematiky, Vol. 15 (1970), No. 4, 235–244

Persistent URL: http://dml.cz/dmlcz/103292

USE OF THE METHOD OF FACTORIZATION FOR SOLVING DIFFERENTIAL EQUATIONS IN A RECTANGULAR REGION

JOSEF ČERMÁK

(Received October 10, 1968)

INTRODUCTION

When solving partial differential equations of elliptic type, very often we are compelled to use the method of nets. The obtained systems of difference equations are usually very extensive. Their solution by Gauss elimination demands a large capacity of the store of the computer. Then the iteration methods of solving are often used. But their use, when greater exactness is required, makes heavy demands on the computing time. As the system contains a large number of zero coefficients, it is suitable to use the factorization method for the solution. This method is one of direct methods, but it employes the store of the computer minimally. The volume of calculation work is substantially minor than for the above mentioned methods.

1. DESCRIPTION OF SOLUTION

Consider the linear partial differential equation of the second order of elliptic type in the form

$$a(x, y) \frac{\partial^2 u}{\partial x^2} + c(x, y) \frac{\partial^2 u}{\partial y^2} + d(x, y) \frac{\partial u}{\partial x} + e(x, y) \frac{\partial u}{\partial y} = f(x, y) \tag{1}$$

with the boundary condition

$$r(x, y) \frac{\partial u}{\partial n} = s(x, y)\, u + t(x, y) \tag{2}$$

at the boundary of a rectangular region $(0 \leqq x \leqq p, 0 \leqq y \leqq q)$. $\partial/\partial n$ signifies the derivative in the direction of outer normal.

In the direction x let us choose m inner nodal points of the net, in the direction y let it be n, then their coordinates are determined by

$$x_j = (j + \tfrac{1}{2})\,h\,, \quad y_i = (i + \tfrac{1}{2})\,k \tag{3}$$

where $h = p/m$, $k = q/n$.

Consider the notation $a_{ij}, \ldots, a(x_j, y_i)$, etc. for further functions. Then the difference equations obtained from the equation (1) can be written:

$$a_{i,j}\frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} + c_{i,j}\frac{u_{i+1,j} - 2u_{i,j} + u_{i+1,j}}{k^2} + \tag{4}$$

$$+ d_{i,j}\frac{u_{i,j+1} - u_{i,j-1}}{2h} + e_{i,j}\frac{u_{i+1,j} - u_{i-1,j}}{2k} = f_{i,j}$$

where $i = 0, 1, 2, \ldots, n - 1$; $j = 0, 1, 2, \ldots, m - 1$.

Consequently the system contains $m \, . \, n$ equations of $m \, . \, n + 2m + 2n$ unknowns. Further equations, which are resulting from the equation (2), can be grouped together with the equations (4). For $y = 0, q$ it can be written

$$r(x_j, 0)\frac{u_{0,j} - u_{-1,j}}{k} = s(x_j, 0)\frac{u_{0,j} + u_{-1,j}}{2} + t(x_j, 0)\,, \tag{5}$$

$$r(x_j, q)\frac{u_{n,j} - u_{n-1,j}}{k} = s(x_j, q)\frac{u_{n,j} + u_{n-1,j}}{2} + t(x_j, q)$$

for $j = 0, 1, \ldots, m - 1$

(6) Let us interpret $\{u_{i,0}, u_{i,1}, \ldots, u_{i,m-1}\}$ as the vector $\boldsymbol{U}_i$,

then the equations (5) can be written in the form

$$R_0\frac{\boldsymbol{U}_0 - \boldsymbol{U}_{-1}}{k} = S_0\frac{\boldsymbol{U}_{-1} + \boldsymbol{U}_0}{2} + \boldsymbol{T}_0\,, \tag{7}$$

$$R_n\frac{\boldsymbol{U}_n - \boldsymbol{U}_{n-1}}{k} = S_n\frac{\boldsymbol{U}_n + \boldsymbol{U}_{n-1}}{2} + \boldsymbol{T}_n\,. \tag{8}$$

R_0, S_0, R_n, S_n are diagonal matrices, $\boldsymbol{T}_0$, $\boldsymbol{T}_n$ are vectors with the elements $r(x_j, 0)$, $s(x_j, 0)$, $r(x_j, q)$, $s(x_j, q)$, $t(x_j, 0)$, $t(x_j, n)$, respectively.

The equation (7) replaces the equations of the nodal points of the first line, and the equation (8) – those of the n-th line. The equations of other remaining lines are treated similarly. The equations for the nodal points of the i-th line can be written as follows:

$$r(0, y_i)\frac{u_{i,0} - u_{i,-1}}{h} = s(0, y_i)\frac{u_{i,0} + u_{i,-1}}{2} + t(0, y_i), \tag{9}$$

$$r(p, y_i)\frac{u_{i,m} - u_{i,m-1}}{h} = s(p, y_i)\frac{u_{i,m} + u_{i,m-1}}{2} + t(p, y_i). \tag{10}$$

The equation (4) refers to the inner point of this line. By applying the equation (9), u_{i-1} can be eliminated out of the system, by applying the equation (10) we can eliminate the unknown $u_{i,m}$.

The equations (9) and (10) have been treated in the form

$$u_{i,-1} = u_{i,0}\frac{2r(0, y_i) - h\, s(0, y_i)}{2r(0, y_i) + h\, s(0, y_i)} - \frac{2h\, t(0, y_i)}{2r(0, y_i) + h\, s(0, y_i)}, \tag{11}$$

$$u_{i,m} = u_{i,m1}\frac{2r(p, y_i) + h\, s(p, y_i)}{2r(p, y_i) - h\, s(p, y_i)} + \frac{2h\, t(p, y_i)}{2r(p, y_i) - h\, s(p, y_i)}. \tag{12}$$

After introducing (11) and (12) into the equation (4) and after rearranging them, it can be written

$$M^D_{i,j}u_{i,j-1} + M_{i,j}u_{i,j} + M^H_{i,j}u_{i,j+1} + D_{i,j}u_{i-1,j} + H_{i,j}u_{i+1,j} = F_{i,j}. \tag{13}$$

The values of the coefficients for $0 < j < m - 1$ are as follows:

$$M^D_{i,j} = \frac{a_{i,j}}{h^2} - \frac{d_{i,j}}{2h}, \quad D_{i,j} = \frac{c_{i,j}}{k^2} - \frac{e_{i,j}}{2k}, \tag{14}$$

$$M^H_{i,j} = \frac{a_{i,j}}{h^2} + \frac{d_{i,j}}{2h}, \quad H_{i,j} = \frac{c_{i,j}}{k^2} + \frac{e_{i,j}}{2k},$$

$$M_{i,j} = -2\left(\frac{a_{i,j}}{h^2} + \frac{c_{i,j}}{k^2}\right), \quad F_{i,j} = f_{i,j}.$$

Coefficients for $j = 0$:

$$M^D_{i,0} = 0, \tag{15}$$

$$M_{i,0} = -2\left(\frac{a_{i,0}}{h^2} + \frac{c_{i,0}}{k^2}\right) + \left(\frac{a_{i,0}}{h^2} - \frac{d_{i,0}}{2h}\right)\frac{2r(0, y_i) - h\, s(0, y_i)}{2r(0, y_i) + h\, s(0, y_i)},$$

$$F_{i,0} = f_{i,0} + \left(\frac{a_{i,0}}{h^2} - \frac{d_{i,0}}{2h}\right)\frac{2h\, t(0, y_i)}{2r(0, y_i) + h\, s(0, y_i)}.$$

The remaining coefficients are coincident with those of (14). The coefficients for $j = m - 1$:

(16)
$$M^H_{i,m-1} = 0\,,$$

$$M_{i,m-1} = -2\left(\frac{a_{i,m-1}}{h^2} + \frac{c_{i,m-1}}{k^2}\right) + \left(\frac{a_{i,m-1}}{h^2} + \frac{d_{i,m-1}}{2h}\right)\frac{2r(p, y_i) + h\, s(p, y_i)}{2r(p, y_i) - h\, s(p, y_i)}\,,$$

$$F_{i,m-1} = f_{i,m-1} - \left(\frac{a_{i,m-1}}{h^2} + \frac{d_{i,m-1}}{2h}\right)\frac{2h\, t(p, y_i)}{2r(p, y_i) - h\, s(p, y_i)}\,.$$

The remaining coefficients are coincident with the expressions (14) for $j = m - 1$.

If we apply the vector notation for $u_{i,j}$ according to (6), all equations for nodal points of the i-th line can be expressed by one vector equation

(17)
$$D_i\boldsymbol{U}_{i-1} + M_i\boldsymbol{U}_i + H_i\boldsymbol{U}_{i+1} = \boldsymbol{F}_i$$

where $\boldsymbol{F}_i$ is the vector $\boldsymbol{F}_i = \{F_{i,0}, F_{i,1}, \ldots, F_{i,m-1}\}$,

D_i – the diagonal matrix of elements $D_{i,0}, D_{i,1}, \ldots, D_{i,m-1}$,

H_i – the diagonal matrix of elements $H_{i,0}, H_{i,1}, \ldots, H_{i,m-1}$,

M_i is a tridiagonal matrix. The elements of its main

diagonal are $M_{i,j}$, the elements under the diagonal are succesively $M^D_{i,1}, M^D_{i,2}, \ldots$ $\ldots, M^D_{i,m-1}$. The elements above the diagonal are $M^H_{i,0}, M^H_{i,1}, \ldots, M^H_{i,m-2}$. By grouping the equations (7), (8), (17) together we get a quasitridiagonal system (18)

(18)
$$M_{-1}\boldsymbol{U}_{-1} + H_{-1}\boldsymbol{U}_0 = \boldsymbol{F}_{-1}\,,$$
$$D_i\boldsymbol{U}_{i-1} + M_i\boldsymbol{U}_i + H_i\boldsymbol{U}_{i+1} = \boldsymbol{F}_i \quad \text{for} \quad i = 0, 1, \ldots, n - 1\,,$$
$$D_n\boldsymbol{U}_{n-1} + M_n\boldsymbol{U}_n = \boldsymbol{F}_n\,,$$

where

(19)
$$M_{-1} = \frac{S_0}{2} + \frac{R_0}{k}\,, \quad H_{-1} = \frac{S_0}{2} - \frac{R_0}{k}\,; \quad \boldsymbol{F}_{-1} = -\boldsymbol{T}_0\,,$$

$$D_n = \frac{S_n}{2} + \frac{R_n}{k}\,, \quad M_n = \frac{S_n}{2} - \frac{R_n}{k}\,; \quad \boldsymbol{F}_n = -\boldsymbol{T}_n\,.$$

The solution can be carried out as follows:

The given equation of the system (16) has been treated in the form

(20)
$$\boldsymbol{U}_{-1} = M'_{-1}\boldsymbol{U}_0 + \boldsymbol{F}_{-1}$$

where

$$M'_{-1} = -(M_{-1})^{-1}\,H_{-1}\,, \quad \boldsymbol{F}_{-1} = (M_{-1})^{-1}\,\boldsymbol{F}_{-1}\,.$$

After the substitution into the second equation and after the treatment we get

$$M'_0 \boldsymbol{U}_0 + H_0 \boldsymbol{U}_1 = \boldsymbol{F}_0 \tag{21}$$

where

$$M'_0 = D_0 M'_{-1} + M_0 \,, \quad \boldsymbol{F}'_0 = \boldsymbol{F}_0 - D_0 \boldsymbol{F}'_{-1} \,.$$

Further the process is repeated according to (20), (21) for the indices increasing by one.

The operation (20) is repeated $(n + 2)$ times, the operation (21) is repeated $(n + 1)$ times. In the last operation (20) for $i = n$, H_n is the zero matrix. By the given process $\boldsymbol{U}_n$ is determined during the last step. By the backward process the remaining $\boldsymbol{U}_i$ can be determined.

2. PROGRAMMING OF THE PROBLEM

The problem has been programmed in the reference language ALGOL 60 in the form of procedure. Solving the given problem it is necessary to introduce the function procedures *funa* (i, j), *func* (i, j), *fund* (i, j), *fune* (i, j), *funf* (i, j), *funr* (x, y), *funs* (x, y), *funt* (x, y). By these procedures the coefficients of the equation (1) eventually (2) can be determined. It is necessary to know $\boldsymbol{F}'_i$, M'_i, $\boldsymbol{U}_i$ for the backward process of the computation. Therefore it is not necessary to introduce any other arrays. In the first part of the program the values $F_{i,j}$ are placed in the cells $F'_{i,j}$, the values $M_{i,j}$ are placed in the cells $M'_{i,j}$. The matrix M_{-1} is diagonal as well as the matrix M'_{-1}. In the backward process we do not need to know the values of the elements of this matrix, therefore we do not introduce it. The number of the elements of the matrix M_i is $m^2(n + 1)$. Therefore it is convenient to apply the program so that $m < n$. As $\boldsymbol{U}_i$ is needed only in the backward process of the computation in the place where it is not necessary to hold in store $f_{i,j}$, it is possible to place individual elements $u_{i,j}$ into the cells $f_{i,j}$.

In the program we have employed the following notation of values:

m	number of inner points in the direction x,
n	number of inner points in the direction y,
p	magnitude of the area in the direction x,
q	magnitude of the area in the direction y,
eps	if the element of the main diagonal is less then *eps* during the inversion, the program continues from the label *OUT*,
step h	magnitude of the step in the direction $x(h)$,
step k	magnitude of the step in the direction $y(k)$,
mat $[i, j, k]$	elements of the matrix M_i, j the line index, k the column index,
$f[i, j]$	array of the right sides, array of function values at the nodal points $[0 : n, 0 : m - 1]$,

funa (i, j)	value of the coefficient $a(x, y)$ of the equation (1) at the point $x = x_j$, $y = y_i$, the name of the function procedure $(x_j = (j + \frac{1}{2})\, h;\ y_i = = (i + \frac{1}{2})\, k)$,
func (i, j)	value of the coefficient $c(x, y)$ of the equation (1) at the point $x = x_j$, $y = y_i$, the name of the function procedure,
fund (i, j)	value of the coefficient $d(x, y)$ of the equation (1) at the point $x = x_j$, $y = y_i$, the name of the function procedure,
fune (i, j)	value of the coefficient $e(x, y)$ of the equation (1) at the point $x = x_j$, $y = y_i$, the name of the function procedure,
funf (i, j)	value of the coefficient $f(x, y)$ of the equation (1) at the point $x = x_j$, $y = y_i$, the name of the function procedure,
funr (x, y)	value of the coefficient $r(x, y)$ of the boundary condition (2), the name of the function procedure,
funs (x, y)	value of the coefficient $s(x, y)$ of the boundary condition (2), the name of the function procedure,
funt (x, y)	value of the coefficient $t(x, y)$ of the boundary condition (2), the name of the function procedure,
OUT	the label for the case that the element of the main diagonal $<$ *eps*.

The diagonal matrices D_i, H_i are not introduced, the values of their elements can be determined always when needed.

3. THE PROGRAM

```
Procedure factor (m, n, p, q, eps, funa, func, fund, fune, funf, funr,
  funs, funt, OUT) result : (f);
  value m, n, p, q; integer m, n; real p, q, eps; array f;

real procedure funa, func, fune, funf, funr, funs, funt;
  label OUT;

begin integer i, j, k; real step h, step k, help;
  array x [0 : m − 1], y[0 : n − 1], mat [0 : n, 0 : m − 1, 0 : m − 1];
  step h := p/m; step k := q/n; x[0] := step h/2; y[0] := step k/2;
  for j := 1 step 1 until m − 1 do x[j] := x[j − 1] + step h;
  for i := 1 step 1 until n − 1 do y[i] := y[i − 1] + step k;
  for k := 0 step 1 until n do
    for j := 0 step 1 until m − 1 do
    for $i$ := 0 step 1 until $m - 1$ do $mat\,[k, i, j] := 0$;
for $i$ := 0 step 1 until $n - 1$ do
  begin real $help\ 1$, $help\ 2$, $help\ 3$;
    for $j$ := 0 step 1 until $m - 1$ do
      begin $mat\,[i, j, j] := -2\times(funa\,(i, j)/step\ h \uparrow 2 + func\,(i, j)/step\ k \uparrow 2)$;
        $f[i, j] := funf\,(i, j)$;
        if $j > 0$ then $mat\,[i, j, j - 1] := funa\,(i, j)/step\ h \uparrow 2 -$
                                     $fund\,(i, j)/(2 \times step\ h)$;
        if $j < m - 1$ then $mat\,[i, j, j + 1] := funa\,(i, j)/step\ h \uparrow 2 +$
                                     $fund\,(i, j)/(2 \times step\ h)$
      end $j$;
    $help := funa\,(i, 0)/step\ h \uparrow 2 - fund\,(i, 0)/(2 \times step\ h)$;
    $help\ 1 := 2 \times funr\,(0, y[i]) + step\ h \times funs\,(0, y[i])$;
    $help\ 2 := funa\,(i, m - 1)/step\ h \uparrow 2 + fund\,(i, m - 1)/(2 \times step\ h)$;
    $help\ 3 := 2 \times funr\,(p, y[i]) - step\ h \times funs\,(p, i[i])$;
    $mat\,[i, 0, 0] := mat\,[i, 0, 0) + help \times (2 \times funr\,(0, y[i]) - step\ h \times$
                  $\times funs\,(0, y[i]))/help\ 1$;
    $mat\,[i, m - 1, m - 1] := mat\,[i, m - 1, m - 1] + help\,2 \times$
                       $\times (2 \times funr\,(p, y[i]) + step\ h \times$
                       $\times funs\,(p, y[i]))/help\ 3$;
    $f[i, 0] := f[i, 0] + help \times 2 \times step\ h \times funt\,(0, y[i])/help\ 1$;
    $f[i, m - 1] := f[i, m - 1] - help\,2 \times 2 \times step\,h \times funt\,(p, y, [i])/help\ 3$
  end $i$;
for $j$ := 0 step 1 until $m - 1$ do
  begin $mat\,[n, j, j] := funs\,(x[j], q)/2\text{-}funr\,(x[j], q)/step\ k$;
    $f[n, j] := -funt\,(x[j], q)$;
    comment here expressing in, numbers of matrix coefficients
    ends, the coefficients $D$, $H$ will be expressed in numbers
    during the course;
    $help := 2 \times step\ k/(funs\,(x[j], 0) \times step\ k + 2 \times funr\,(x[j], 0)) \times$
      $(func\,(0, j)/step\ k \uparrow 2 - fune\,(0, j)/(2 \times step\ k))$;
    $mat\,[0, j, j] := mat\,[0, j, j] - help \times (funs\,(x[j], 0)/2\text{-}funr\,(x[j], 0)$
      $/step\ k)$;
    $f[0, j] := f[0, j] - help \times funt\,(x[j], 0)$
  end $j$;
```

```
comment elimination of the vector U minus one has been done,
from here the direct process of solving starts;
for k := 0 step 1 until n do
  begin integer k 1; array help f[0 : m − 1];
    for i := 0 step 1 until m − 1 do
      begin
        if abs (mat [k, i, i]) ≦ eps then go to OUT;
        help := 1/mat [k, i, i]; mat [k, i, i] := 1;
        for k 1 := 0 step 1 until m − 1 do mat [k, i, k1] := mat [k, i, k1] ×
          × help;
        for j := 0 step 1 until m − 1 do
          if i ≠ j then
            begin help := mat [k, j, i]; mat [k, j, i] := 0;
              for k1 := 0 step 1 until m − 1 do
                mat [k, j, k1] := mat [k, j, k1] − help × mat [k, i, k1]
            end j
      end i;
    for i := 0 step 1 until m − 1 do
      begin help := 0;
        for j := 0 step 1 until m − 1 do
          help := help + mat [k, i, j] × f[k, j];
        help f[i] := help
      end i;
    for i := 0 step 1 until m − 1 do f[k, i] := help f[i];
    if k = n then go to s;
    for j := 0 step 1 until m − 1 do
      begin help := func (k, j)/step k ↑ 2 + fune (k, j)/(2 × step k);
        for i := 0 step 1 until m − 1 do mat [k, i, j] := −mat [k, i, j] × help;
      end j;
    for i := 0 step 1 until m − 1 do
```

```
        begin help := if $k = n - 1$ then funs $(x[i], q)/2$ + funr $(x[i], q)/step\ k$
        else func $(k + 1, i)/step\ k \uparrow 2$ − fune $(k + 1, i)/(2 \times step\ k)$;
          for $j := 0$ step 1 until $m - 1$ do
            mat $[k + 1, i, j]$ := mat $[k + 1, i, j]$ + help × mat $[k, i, j]$;
          $f[k + 1, i] := f[k + 1, i] - help \times f[k, i]$
        end $i$;
s: end $k$;
  comment end the direct process;
  for $k := n - 1$ step $-1$ until 0 do
    for $i := 0$ step 1 until $m - 1$ do
      begin help := 0;
        for $j := 0$ step 1 until $m - 1$ do
          help := help + mat $[k, i, j]$ + $f[k + 1, j]$;
        $f[k, i]$ := help + $f[k, i]$
      end $i$
end factor;
```

References

[1] Алгоритмы и алгоритмические языки, выпуск 4, Академия наук СССР вычислительный цэнтр (1967).

[2] *С. К. Годунов, В. С. Рябенький:* Введение в теорию разностных схем, Москва 1962.

Souhrn

UŽITÍ METODY FAKTORIZACE PŘI ŘEŠENÍ DIFERENCIÁLNÍCH ROVNIC V PRAVOÚHLÉ OBLASTI

Josef Čermák

Řešení lineárních parciálních diferenciálních rovnic eliptického typu metodou sítí vede pravidelně k rozsáhlým systémům diferenčních rovnic. Matice systému obsahuje velký počet nulových koeficientů. V publikaci je ukázáno, jak vhodnou interpretací

počítaných hodnot funkce v uzlových bodech obdélníkové sítě lze podstatně snížit potřebu paměti počítače, současně snížit i množství aritmetických operací. Program předpokládá oblast řešení obdélník omezený souřadnicovými osami a přímkami $y = q$, $x = p$. Uvedená oblast obsahuje ve směru x . m vnitřních bodů sítě, ve směru y . n vnitřních bodů. V jednotlivých směrech jsou vzdálenosti síťových bodů konstantní. Okrajové body jsou vzdáleny o půl kroku od okraje oblasti. Okrajovou podmínku lze vhodnou volbou funkčních procedur $r(x, y)$, $s(x, y)$, $t(x, y)$ v uživatelském programu upravit na podmínky I, II i III. druhu (viz rovnice (2)). Řešení rovnic s různými koeficienty je opět umožněno vhodnou volbou funkčních procedur a, c, d, e, f v uživatelském programu.

Author's address: Dr. *Josef Čermák*, CSc., Vysoká škola chemickotechnologická v Pardubicích, Leninovo n. 565, Pardubice.