# THE FINITE ELEMENT METHOD FOR SOLVING THE POISSON PARTIAL DIFFERENTIAL EQUATION

***C. I. Popovici***
Technical University ″Gh. Asachi" Iassy

## INTRODUCTION

Further, we will present the Poisson equations with mixed conditions on frontier solution, using a method of finite element.

## 1. THE PROBLEM

Let be $D \subset \mathbf{R}^2$ a bordered domain, having the regular frontier **S** and an operator A of Laplace type with next expression:

$$A = -\nabla\ K\ \nabla, \qquad (1)$$

where K = ($k_{ij}$ (x, y)), I, j=1, 2 is a symmetrical matrix of continuous functions on **D**, express of physical anisotropy of the domain **D** on which we study the described phenomena by Poisson equations (3).

We will noted with $D_A$ the whole set of the functions definite on $\overline{D}$ that satisfy the following conditions:

$$u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \quad \textit{continuous functions}$$

$$\frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x \partial y}, \frac{\partial^2 u}{\partial y^2} \quad \textit{continuous functions on portions}$$

They satisfy a mixed condition on the frontier S with this expression:

$$K\nabla u * \overline{n} + p(s)u - h(s) = 0 \qquad (2)$$

where:

- $\overline{n}$ is the normal unit vector toward the exterior frontier S;
- $\overline{n}$ p(s), h(s) are continuous functions on S.

Further, we will determine a function $u_0 \in D_A$, solution for the equation:

$$Au = g, \qquad (3)$$

where the function g named in engineering problems as "source function" is at least integrable on **D**.

It is demonstrated that the operator A definite through (3) is symmetrical and positive definite and the solution of the equation (3) with the frontier condition (2) is equivalence on $D_A$ with the functional minimisation

$$F(u) = <-Au, u> - 2 <g, u> = \iint_D [(\nabla u) K (\nabla u) - 2gu]\,dxdy - \int_S \left(p(s)u^2 - 2h(s)u\right)ds \qquad (4)$$

For the function $u_0 \in D_A$ determination that accomplish minimum of (4) further we will use the known Riesz method.

## 2. THE DISCRETE PROBLEM

To the formulate problem is associated a discrete problem. We will build a minifying sequence of functions $w_1$, $w_2$, … with imposed form, minimise sequence for the functional (4), that is:

$$\lim_{M \to \infty} \|W_M\|_E = \|u_0\|_E,$$

where $\|\cdot\|_E$ is the energetic norm.

The idea of finite element method consists in the domain decomposing into disjoint subsets reunion $T_i$ (named elements) satisfying the conditions:

$$\bigcup_{i=1}^{M} \overline{T}_i = D;$$

$$\overline{T}_i \cap \overline{T}_j = \begin{Bmatrix} a\ point; \\ a\ curve; \\ \Phi \end{Bmatrix} \qquad i,j = \overline{1, M} \qquad (5)$$

Corresponding to this divisions the functional F(u) will be:

$$F(u) = \sum_{i=1}^{M} F_i(u), \qquad (6)$$

where: $F_i(u)$ express the energy on the element $T_i$.

To every element $T_m$ of the division $T_1, T_2, \ldots, T_M$ will be associated a number of characteristically points $P_m$ named nodal points, obtaining N nodes on the whole domain. We will associate to every interior node "i" of ($x_i$, $y_i$) co-ordinates, a function $f_i(x,y)$ with the following proprieties:

a. $f_i$ continuous on $\overline{D}$, i=1,2,…,N,
b. $f_i(x_i, y_i) = \delta_{ij}$, (7)

c. $<f_i, f_j> = 0$, for $i \neq j$,

d. $\forall i \in \overline{1, M}$, the function $f_i$ satisfies the condition (2) on **S**.

Let be $E_N$ the subspace of $D_A$, generated by the function $f_1, f_2, \ldots, f_N$.
A certain function $w_N \in E_N$ will have the next form:

$$w_N = \sum_{i=1}^{N} c_i f_i \tag{8}$$

The function $u(x, y) \in D_A$ will be approximated on every $T_i$ element with a continuous on D function w(x,y) with continuous on portions partial derivatives of first order.
For a certain element $T_m$ of the proposed division, the approximation function can be writhe as:

$$w_m(x,y) = \sum_{i=1}^{P_m} f_{m_i}(x,y) w_{m_i}, \tag{9}$$

Where $f_{m_i}$ is the function $f_1, f_2, \ldots f_N$ restriction to $T_m$ element which contains the node "i" and $w_{m_i}$ is the w(x, y) function value in the node "i" and have part of variations parameter .
We will insert the matric notations:

$$F_m = \lfloor f_{m_1}, f_{m_2}, \ldots f_{p_m} \rfloor, \tag{10}$$

$$W_m^T = \lfloor w_{m_1}, w_{m_2}, \ldots w_{p_m} \rfloor, \tag{11}$$

the relation (9) became:

$$w_m(x,y) = f_m w_m \tag{12}$$

At every element $T_m$ it is associated an matrix $L_m$ with $P_m$ lines and N columns, which put into biunique correspondence the $T_m$ element nodes and obtained nodes on D through performing the divisions in the elements $T_1, \ldots, T_M$.

Making the notations:

$$W^T = [W_1, W_2, \ldots, W_N] \tag{13}$$

and using the matrix $L_m$ we will obtain the link between the relations (12) and (13):

$$W_m = L_m W, \tag{14}$$

and (7) became:

$$w_m(x,y) = W^T L_m^T F_m^T \tag{15}$$

By means of formula (15) we obtained a interpolation on $T_m$ for the function from $D_A$.
If in (6) the function u is replaced with $w_m$ it is obtained:

$$F(W) = \sum_{m=1}^{M} \left( \iint_{T_m} (\nabla W_m K \nabla W_m - 2 W_m g_m) dxdy + \right.$$

$$\left. + \int_{S_m} \left( 2 h_m W_m - p_m W_m^2 \right) ds \right), \tag{16}$$

where:
$g_m$, $h_m$, $p_m$ are the functions g, h, p restrictions at the elements $T_m$ on the frontier $S_m$.
Taking into consideration (15) the expresion of $\nabla W_m(x,y)$ is:

$$\nabla W_m(x,y) = B_m L_m W_m,$$

where:

$$B_m = \begin{pmatrix} \dfrac{\partial f_{m1}}{\partial x} \cdots \dfrac{\partial f_{mP_m}}{\partial x} \\ \dfrac{\partial f_{m1}}{\partial y} \cdots \dfrac{\partial f_{mP_m}}{\partial y} \end{pmatrix}$$

The similar expressions with (12) can be obtain also for the functions g, h, p:
g (x,y) = $F_m g_m$,
h (x,y) = $F_m h_m$,
p (x,y) = $F_m p_m$,
where :
$g_m$, $h_m$, $p_m$ are vectors that contain functions values g, h, p in the element $T_m$ nodes.

The expresion of the functional (16) became:

$$F(W) = W^T \left[ \sum_{m=1}^{M} \left( L_m^T G_m L_m \right) W - 2 \sum_{m=1}^{M} L_m^T H_{gm} + \right.$$

$$\left. + 2 \sum_{m=1}^{M} L_m^T H_{hm} - \sum_{m=1}^{M} L_m^T H_{p_m} W \right] \tag{17}$$

where we made the notations:

$$H_{g_m} = \left( \iint_{T_m} F_m^T F_m dxdy \right) g_m,$$

$$H_{h_m} = \left( \iint_{S_m} F_m^T F_m ds \right) h_m,$$

$$H_{p_m} = \left( \iint_{S_m} F_m^T F_m ds \right) p_m,$$

The function (17) minimisation relative with $W_1, W_2, \ldots, W_N$ leads at the linear system solution:

$$\begin{cases} \dfrac{\partial F}{\partial w_1} = 0 \\ \ldots\ldots\ldots\ldots\ldots \\ \dfrac{\partial F}{\partial w_n} = 0 \end{cases} \qquad (18)$$

which can be also write as:

$$GW = H_g - H_h + H_p, \qquad (19)$$

where we noted with:

$$G = \sum_{m=1}^{M} L_m^T G_m L_m,$$

$$H_g = \sum_{m=1}^{M} L_m^T H_{g_m},$$

$$H_h = \sum_{m=1}^{M} L_m^T H_{h_m},$$

$$H_p = \sum_{m=1}^{M} L_m^T H_{p_m}.$$

The system (19) unknowns are: functions values $u_0$ that minimises the energy functional and so, the problem (3) solution in net nodes formed through the domain D division into finite elements.

This system solutions is quite comfortable because in case of the definite and symmetrical positive operators, the system (19) matrix is symmetrical having moreover an band structure that means an substantial advantage in numerical solution of system (19) by computer way.

## 3. THE COMPUTER PROGRAM

The computer program in C++ language solves an undetermined compatible system of n linear equations with n unknowns. For round-off errors reduction which are made by computer, before system and subsystems processing it is replace the equation that must to contain the one equation pivot from subsystem so that the obtained pivot to have the maximum absolute value.

The terms of problem are the system dimension, the coefficients and the free terms.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
int n, i, j, k, p, t;
double a [10] [11], x[10], max, s;
void main (void)
{cout << “Dimensiunea sistemului n=”;
cin >> n ;
cout << “Introduceţi coeficienţii : “ <<endl ;
for (i=1 ; i<=n ; i++)
{cout  << ” b ( “ << i <<”) = ” ;
cin >> a[i] [n+1] ; }
k=1;
t=0;
do
{max = fabs (a [k] [k]);
p = k ;
for ( i = k+1 ; i < = n ; i + + )
if (max < fabs (a [i] [k]))
{ max = fabs (a [i] [k]);
p = i ; }
if (max = = 0)
t = 1;
else
{ if (p|=k)
for (j = k;  j< n+1; j + + )
{ s = a [i] [k] ;
a [i] [k] = a [p] [j];
a [p] [j] = s; }
for ( i = k+1 ; i<= n; i + + )
for ( j = k+1 ; j<= n+1;  j + + )
a [i] [j] - = a [i] [k] * a [k] [j] / a [k] [k];
k + + ;  }}
while (( k < n) & & (t = = 0 ) )
cout << “Sistemul nu este compatibil determinat ! ″
<< endl ;
else
{ for (i = n ; i>0 ; i - - )
{ s = 0;
for ( j = i+1 ;  j<=n; j + + )
s + = a [i] [j] * x[j];
x[i] = (a [i] [n+1] –s / a[i] [i]; }
cout << “Soluţia sistemului:” << endl;
for ( i = 1 ; i<= n ; i + +)
cout << “x ( “ << i << “ ) = “ << x[i] << endl;}
getch ( ) ; }
```

## Bibliography


1. **Dragos, L.** *Mecanica fluidelor. **I**, Editura Academiei Române, Bucurşeti, 1999*
2. **Marcov, N.** *Metoda elementelor finite. În: Matematici Clasice şi Moderne, vol.IV (sub redacţia Caius Iacob), Editura Tehnică, Bucureşti, 1984.*
3. **Titus, P., Gheorghiu, C. I.** *Metode element finit şi aplicaţii, Editura Academiei R.S.R., Bucureşti, 1987.*
4. **Popovici, C.I.** *The finite element method for solving the two-*dimensional Stefan *problem with two phases (II), Mathematical Reports, Tom 3(53), ( 2001),3, 273-286, Bucureşti.*