FACTA UNIVERSITATIS
Series: **Architecture and Civil Engineering** Vol. 6, N° 2, 2008, pp. 199 - 205
DOI:10.2298/FUACE0802199L

PROGRAM FOR CALCULATION OF A DEFLECTION OF A UNIFORM LOADED SQUARE PLATE USING GAUSS-SEIDEL METHOD FOR SOLUTION OF POISSON DIFFERENTIAL EQUATION

UDC 624.073.1:624.042.3(045)=111

Jana Lipkovski[1], Davorin Penava[2]

[1]University of Belgrade, Faculty of Civil Engineering, Serbia
[2]University of Osijek, Faculty of Civil Engineering, Croatia

Abstract. *In this work we will consider a simply supported square plate of side L that is subjected to a load q per unit area. The deflection w in the z-direction is the solution of the Poisson differential equation. For simplicity, we suppose that the loading is uniform so that q is constant. The program is written for computing the deflections w at a set of points with n intervals along each side of the square plate. In this case, we employ a program that uses the Gauss-Seidel method to approximate the solution of Poisson's equation. The program is written for kmax applications through all interior grid points.*

Key words: *Square plate, uniform load, deflection, program, Poisson's equation, Gauss-Seidel method, grid*

1. INTRODUCTION

At this time practically all finite element programs use some form of Gauss elimination to solve the equilibrium equations $KU = R$. However, it is interesting to note that during the initial developments of the finite element method, iterative solution algorithms have been employed extensively, and much research has been spent on improving various iterative solution schemes. A basic disadvantage of an iterative solution is that the time of solution can be estimated very approximately, because the number of iterations required for convergence depends on the condition number of the matrix K and whether effective acceleration factors are used.

The objective in this section of $KU = R$, it is necessary to use an initial estimate for the displacements U, say $U^{(1)}$, which is no better value is known, may be a null vector. In the Gauss-Seidel iteration we than evaluate for $s = 1,2,\ldots$:

$$U_i^{(s+1)} = k_{ii}^{-1}\left\{R_i - \sum_{j=1}^{i-1} k_{ij}U_j^{(s+1)} - \sum_{j=i+1}^{n} k_{ij}U_j^{(s+1)}\right\} \tag{1}$$

Received September 21, 2008

where $U_i^{(s)}$ and R_t are the ith component of U and R, and s indicates the cycle of iteration. Alternatively, we may write in matrix form,

$$U^{(s+1)} = K_D^{-1}\{R - K_L U^{(s+1)} - K_L^T U^{(s)}\} \tag{2}$$

where K_D is a diagonal matrix, $K_D = \mathrm{diag}(k_{ii})$, and K_L is a lower triangular matrix with the elements k_{ij} such that:

$$K = K_D + K_L + K_L^T \tag{3}$$

The iteration is continued until the change in the current estimate of the displacement vector is small enough, i.e., until

$$\frac{\left\| U^{(s+1)} - U^{(s)} \right\|_2}{\left\| U^{(s+1)} \right\|_2} < \varepsilon \tag{4}$$

where ε is the convergence tolerance. The number of iterations depends on the "quality" of starting vector $U^{(1)}$ and on the conditioning of the matrix K. But it is important to note that the iteration will always converge, provided that K is positive definite. Furthermore, the rate of convergence can be increased using overrelaxation, in which case the iteration is as follows,

$$U^{(s+1)} = U^{(s)} + \beta K_D^{-1}\{R - K_L U^{(s+1)} - K_D U^{(s)} - K_l^T U^{(s)}\} \tag{5}$$

where β is the overrelaxation factor. The optimum value β depends on the matrix K but is usually between 1,3 an 1,9.

All trough direct solution methods are used almost exclusively in finite element analysis programs, it is important to recognize some advantages of the Gauss-Seidel method. The solution scheme might be used effectively in the problem areas of reanalysis and optimization, all trough a form of Gauss elimination may still be more efficient. Namely, if in reanalysis a structure is changed only slightly, the previous solution is a good starting vector for the Gauss-Seidel iterative solution f the new structure.

Another advantage of the Gauss-Seidel method can be that the assembled stiffness matrix need not to be formed and thus out-of-force solution is avoided, because all matrix multiplications can be carried out on the element level. For example, instead of calculating $K_L U^{(s-1)}$, we can evaluate $\Sigma_m K_L^{(m)} U^{(s+1)}$, where the summation goes over all finite elements and $K_L^{(m)}$ contribution of the mth matrix to K_L. The same procedure is generally used in the implementation of the central difference scheme for transient analysis.

Finally, we should also recognize that there is the possibility of combining the direct and iterative solution schemes discussed above.

2. Problem Consideration

Consider a simply supported square plate of side L that is subjected to a load q per unit area, as shown in Fig. 2.1 The deflection w in the z-direction is the solution of biharmonic equation:

$$\nabla^4 w = \frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^4 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{q}{D}$$

The boundary conditions are $w = 0$ and $\frac{\partial^2 w}{\partial \eta^2} = 0$ on its four edges, where η denotes the normal to the boundary. D is the flexural rigidity of the plate, given with:

$$D = \frac{E \cdot t^3}{12(1 - \nu^2)} \tag{6}$$

E – Young's modulus, t – Plate thickness, ν – Poisson's ratio

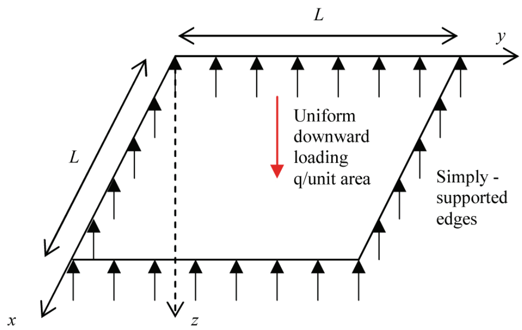


Fig. 2.1 Loaded square plate

For simplicity, suppose the loading is uniform so that q is constant. Now we are approaching the next faze of the procedure and that is, writing a program for computing the deflections w at a set of points with n intervals along each side of the square.

3. METHOD OF SOLUTION

By introducing the variable $u = \nabla^2 w$, the problem amounts to solving Poisson's equation twice in succession:

$$\nabla^2 u = \frac{q}{D} \tag{7}$$

$$\nabla^2 w = u \tag{8}$$

For this purpose, we employ a formula that uses the Gauss-Seidel method to approximate the solution of Poisson's equation:

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = \Psi(x, y) \tag{10}$$

The finite difference approximation of equation (10) is:

$$\frac{\Phi_{i-1,j} - 2\Phi_{i,j} + \Phi_{i+1,j}}{(\Delta x)^2} + \frac{\Phi_{i-1,j} - 2\Phi_{i,j} + \Phi_{i+1,j}}{(\Delta y)^2} = \Psi_{i,j} \tag{11}$$

Thus, for $\Delta x = \Delta y$, the Gauss-Seidel method amounts to repeated application of

$$\Phi_{i,j} = \frac{1}{4}[\Phi_{i-1,j} + \Phi_{i+1,j} + \Phi_{i,j-1} + \Phi_{i,j+1} - (\Delta x)^2 \Psi_{i,j}] \tag{12}$$

at every interior grid point. The program is written for *kmaks* applications of (12) through interior grid points.

Here, the matrix $\Psi_{i,j}$ would contain the known right side values. By using appropriate arguments, the program will solve equations (7) and (8).

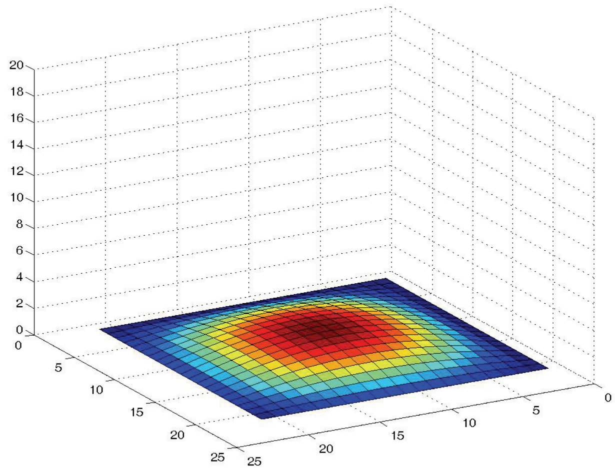


Fig. 2.2 Deflected square plate

5. MATLAB IMPLEMENTATION

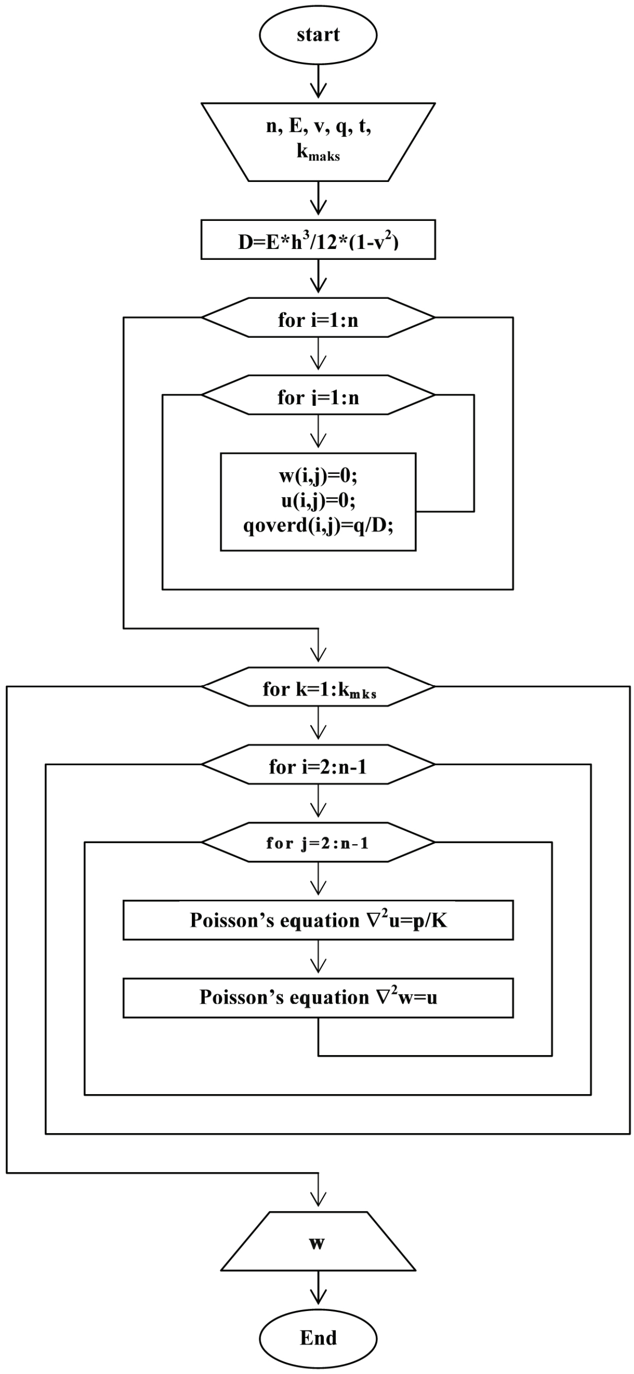
Table 1. List of Principal Variables and they symbols

Program symbol	List of Principal Variables Definition
n	Number of grid spacings along a side of the square, n
E	Young's modulus, E [kN/m^2]
v	Poisson's ratio, v
L	Length of a side of square, L [m]
q	Load per unit area of the plate, q [kN/m^2]
t	Plate thickness, t [m]
k_{maks}	Number of Gauss-Seidel iterations, k_{maks}
D	Flexural rigidity, D [kN/m]
i,j	Grid-point subscripts, i, j
u	Matrix of intermediate variable $\nabla^2 u = q/K$ at each grid point
w	Matrix of downward deflection $\nabla^2 w = u$ [m], at each grid point
qoverd	Matrix with values qoverd(i,j) = q/D at each grid point

Poisson's equation $\nabla^2 u = p/K$: $$u_{i,j} = \frac{1}{4}\left[u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - \left(\frac{L}{n}\right)^2 \frac{q}{k}\right]$$

Poisson's equation $\nabla^2 w = u$: $$w_{i,j} = \frac{1}{4}\left[w_{i-1,j} + w_{i+1,j} + w_{i,j-1} + w_{i,j+1} - \left(\frac{L}{n}\right)^2 u_{i,j}\right]$$

4. FLOW DIAGRAM



6. LOGICAL DECISION MATRIX

Table 2 Logical decision matrix regarding methods for solution of given problem

Method (programming language)	User friendly for engineers (weight: 0.3)	Application spectrum of our program (weight: 0.2)	Time of execution (weight: 0.3)	Output / Grpahical Presentation (weight: 0.2)	Total Score
Gauss-Seidel (Matlab)	60 pt	10 pt	70 pt	80 pt	57 pt
Jacobi (Matlab)	50 pt	10 pt	50 pt	80 pt	48 pt
Gauss-Seidel (Mathematica)	50 pt	10 pt	70 pt	90 pt	56 pt
Jacobi (Mathematica)	40 pt	10 pt	50 pt	90 pt	47 pt
Gauss-Seidel (Pascal)	50 pt	10 pt	70 pt	60 pt	50 pt
Jacobi (Pascal)	40 pt	10 pt	50 pt	60 pt	41 pt

7. CONCLUSION

In this work we consider a simply supported square plate of side L that is subjected to a load q per unit area. The deflection w in the z- direction is the solution of the biharmonic differential equation. For simplicity, we suppose that the loading is uniform so that q is constant. We also use two solution steps, by reducing the problem to the solution of two following Poisson's differential equations. The program is written for computing the deflections w at a set of points with n intervals along each side of the square plate. In this case, we employ a program that uses the Gauss-Seidel method to calculate approximately the solution of Poisson's equation.

The program has been written for one sort of problems, but can be improved by implementation of subprograms by replacing the discrete equations by changing subprograms. Every subprogram would refer to a different equation. Otherwise, we could use different numerical "solvers", by implementing them in different subprograms.

REFERENCES

1. Sekulovic, Miodrag: *Metod konacnih elemenata*, IRO "Gradjevinska knjiga", Beograd, 1988
2. Bathe, Klaus-Juergen: *Finite element precedures in engineering analysis*, Prentice Hall Inc., Englewood Cliffs, New Jersey 1982

PROGRAM ZA PRORAČUN DEFORMACIJA KVADRATNE PLOČE OPTEREĆENE JEDNAKOPODELJENIM OPTEREĆENJEM POMOĆU GAUS-SAJDELOVE METODE ZA SLUČAJ POASONOVE DIFERENCIJALNE JEDNAČINE

Jana Lipkovski, Davorin Penava

U ovom radu se razmatra slobodno oslonjena kvadratna ploča stranice L opterećena opterećenjem q prko cele površine. Pomeranje w u z-pravcuje je rezultat poasonove diferencijalne jednačine. Pretpostavlja se da je optercenje q konstantno. Program je napisan za proračun pomeranja w za skup tačaka u n intervalu duž svake strane kvadratne ploče. U tom slučaju se koristi program sa Gaus-Sajdelovom metodom za aproksimaciju Poasonove diferencijalne jednačine. Program je pisan za kmax za sve tačke u unutrašnjoj mreži.