Bull. Nov. Comp. Center, Num. Anal., 12 (2003), 13–20
© 2003 NCC Publisher

The 3D thermoelasticity program

Y.L. Gurieva*

The main objective of the paper is to present a program to solve the 3D BVP for the problem of linear thermoelasticity. Numerical algorithms for data structures, element-by-element finite volume approximations, and iterative solution are described. Results of numerical experiments are given.

The purpose of the present program is to solve a mixed boundary value problem (BVP) for a system of equations of the linear thermoelasticity theory in the computational domain composed of parallelepipeds, using finite-volume approximation, element-by-element approach to get global matrix, and fast incomplete factorization solvers.

1. The BVP statement

The problem is to find $\overline{u} = (u_1, u_2, u_3) \equiv (u, v, w)$ – a displacement vector, which os a solution to the Lamé equation for the equilibrium state of the deformed body

$$-2\,\mathrm{div}(\mu\varepsilon(\overline{u})) - \mathrm{grad}(\lambda\,\mathrm{div}\,\overline{u}) = \overline{f} - \mathrm{grad}(\alpha(2\mu + 3\lambda)T), \tag{1}$$

where $\overline{u}$ is a displacement vector, $\varepsilon(\overline{u})$ is the deformation tensor with the elements

$$\varepsilon_{i,j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \quad i, j = 1, 2, 3,$$

$\overline{x} = (x_1, x_2, x_3) \equiv (x, y, z)$ is a point in the body, $\overline{f}$ is a load vector, T is temperature, α is the heat extension coefficient, λ, μ are the Lamé coefficients. The latter depend on the Poisson coefficient ν and the Jung modulus E in the following way:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}.$$

The solution is sought for in the domain Ω composed of a set of rectangular subdomains, where the given λ, μ are positive constants and the functions $\overline{f}$, T are smooth in each subdomain.

On the external boundary of Ω, the Dirichlet or the Neumann boundary conditions for the displacements hold.

*Supported by the Russian Foundation for Basic Research under Grants 02-01-01176, 01-07-90367, and NWO–RFBR under Grant 047-008-007.

2. Algorithms

The domain Ω is discretized by a non-uniform parallelepipedoidal grid with the nodes (x_i, y_j, z_k) in such a way that the boundary parts and the boundary edges belong to the coordinate planes and grid lines, respectively.

Note that equation (1) is derived from the condition of the equilibrium state of the linear elasticity theory:

$$\operatorname{div}\sigma + \overline{f} = \overline{0}, \tag{2}$$

where $\overline{0}$ is the null vector and

$$\sigma = 2\mu\varepsilon + \lambda \operatorname{div}\overline{u}I - \gamma TI \tag{3}$$

is a symmetric stress tensor, $\gamma = \alpha(2\mu + 3\lambda)$, and I is a unit tensor with the components δ_{ij}.

We apply the finite volume approach to obtain an approximation of (1). This means that we integrate equation (2) over a finite volume

$$\begin{aligned} V_{i,j,k} = \Big\{ &(x_i + x_{i-1})/2 \le x \le (x_i + x_{i+1})/2, \\ &(y_j + y_{j-1})/2 \le y \le (y_j + y_{j+1})/2, \\ &(z_k + z_{k-1})/2 \le z \le (z_k + z_{k+1})/2 \Big\} \end{aligned}$$

around a grid node and come to

$$-\int\limits_{S_{i,j,k}} \sigma\overline{n}\, ds = \int\limits_{V_{i,j,k}} \overline{f}\, dV, \tag{4}$$

where $S_{i,j,k}$ is the surface of the finite volume $V_{i,j,k}$. Using (3), we can write down three components of the product $\sigma\overline{n}$ in the following form:

$$\sigma n_x = \left\{ \begin{array}{c} 2\mu\dfrac{\partial u}{\partial x} + \lambda\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} + \dfrac{\partial w}{\partial z}\right) - \gamma T \\ \mu\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right) \\ \mu\left(\dfrac{\partial u}{\partial z} + \dfrac{\partial w}{\partial x}\right) \end{array} \right\},$$

$$\sigma n_y = \left\{ \begin{array}{c} \mu\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right) \\ 2\mu\dfrac{\partial v}{\partial y} + \lambda\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} + \dfrac{\partial w}{\partial z}\right) - \gamma T \\ \mu\left(\dfrac{\partial v}{\partial z} + \dfrac{\partial w}{\partial y}\right) \end{array} \right\},$$

$$\sigma n_z = \left\{ \begin{array}{c} \mu\left(\dfrac{\partial u}{\partial z} + \dfrac{\partial w}{\partial x}\right) \\ \mu\left(\dfrac{\partial v}{\partial z} + \dfrac{\partial w}{\partial y}\right) \\ 2\mu\dfrac{\partial w}{\partial z} + \lambda\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} + \dfrac{\partial w}{\partial z}\right) - \gamma T \end{array} \right\}.$$

We use the element-by-element approach [1] to obtain the global matrix of the resulting system of equations. This means that for the grid element

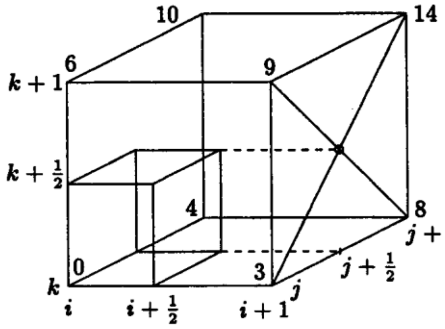
$$E_{i,j,k} = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}]$$

we calculate local additives for the resulting grid equations for eight vertex nodes of the element from the parts of the surface integrals over eight finite volumes around the vertex nodes which are contained in this element. The global matrix is assembled via the local matrices for all grid elements. The global matrix has 3 × 3 block structure

$$A = \left(\begin{array}{ccc} A_{uu} & A_{uv} & A_{uw} \\ A_{vu} & A_{vv} & A_{vw} \\ A_{wu} & A_{wv} & A_{ww} \end{array} \right)$$

and is a symmetric one, i.e., $A_{vu} = A_{uv}$, $A_{wu} = A_{uw}$, $A_{wv} = A_{vw}$, so we will take into account only six matrix blocks: three diagonal blocks and three above diagonal ones. Each block is of $N \times N$ dimension, where N is the number of unknowns for one component of the displacement vector.

A simple quadrature formula for the surface integral (4) in $E_{i,j,k}$ gives the following representation for, e.g., the surface integral for u-component over the surface $S_{0,3}$, which is a part of the surface of the finite volume $V_{i,j,k}$ around node number 0 restricted by the element $E_{i,j,k}$ and which is perpendicular to the element edge $(0,3)$ at the point with the coordinates $(x_{i+1/2}, y_j)$ (the figure):



Local numbering of the nodes in the element $E_{i,j,k}$

$$\Gamma^u_{0,3} = -S_{0,3}\sigma n_x^{(u)} = -\frac{h_z h_y}{4}\Bigg[-(2\mu+\lambda)\frac{u_3 - u_0}{h_x} + \lambda\left(\frac{v_4 + v_8 - v_0 - v_3}{2h_y} + \frac{w_6 + w_9 - w_0 - w_3}{2h_z}\right) - \gamma\frac{T_0 + T_3}{2}\Bigg].$$

Here we use the notations for unknowns according to the local element numbering presented in the figure.

For each element node, we should calculate three surface integrals for three components of the displacement vector, so the total number of integrals to be calculated is $8 \cdot 3 \cdot 3 = 72$. As an example, below we give approximations of the rest two quadratures for the surface integrals around node number 0 for u-component:

$$\begin{aligned} I^u_{0,4} &= -S_{0,4}\sigma n^{(u)}_y = -\frac{h_z h_x}{4}\mu\left(\frac{u_4 - u_0}{h_y} + \frac{v_8 + v_3 - v_0 - v_4}{2h_x}\right), \\ I^u_{0,6} &= -S_{0,6}\sigma n^{(u)}_z = -\frac{h_x h_y}{4}\mu\left(\frac{u_6 - u_0}{h_z} + \frac{w_9 + w_3 - w_6 - w_0}{2h_x}\right). \end{aligned}$$

The local matrix, representing the approximation of fluxes on the element, has the same 3×3 block structure as global matrix:

$$A^l = \begin{pmatrix} A^l_{uu} & A^l_{uv} & A^l_{uw} \\ A^l_{vu} & A^l_{vv} & A^l_{vw} \\ A^l_{wu} & A^l_{wv} & A^l_{ww} \end{pmatrix}.$$

Here each block is of 8×8 size and has four nonzero elements in each row.

It follows from the quadratures $I^u_{0,3}$, $I^u_{0,4}$, $I^u_{0,6}$ that the corresponding elements of the local matrix have the form

$$\begin{aligned} uu_{0,3} &= -\frac{h_z h_y}{4h_x}(2\mu + \lambda), & uu_{0,4} &= -\frac{h_z h_x}{4h_y}\mu, \\ uu_{0,6} &= -\frac{h_x h_y}{4h_z}\mu, & uu_{0,0} &= uu_{0,3} + uu_{0,4} + uu_{0,6}, \\ uv_{0,0} &= \frac{h_z}{8}(\lambda + \mu), & uv_{0,3} &= \frac{h_z}{8}(\lambda - \mu), \\ uv_{0,4} &= \frac{h_z}{8}(-\lambda + \mu), & uv_{0,8} &= \frac{h_z}{8}(-\lambda - \mu), \\ uw_{0,0} &= \frac{h_y}{8}(\lambda + \mu), & uw_{0,3} &= \frac{h_y}{8}(\lambda - \mu), \\ uw_{0,4} &= \frac{h_y}{8}(-\lambda + \mu), & uw_{0,8} &= \frac{h_y}{8}(-\lambda - \mu). \end{aligned}$$

Hence, the approximation stencil for the diagonal blocks in the global matrix is a usual 7-point stencil, the stencil for A_{uv} block being a 9-point stencil in the plane which is perpendicular to z-axis, and the stencil for A_{uw} block is also a 9-point stencil in the plane perpendicular to y-axis. From similar considerations for the integrals for v-component, we obtain an approximation stencil of the last off-diagonal block A_{vw}: it lies in the plane perpendicular to x-axis and is a 9-point stencil.

Taking into account symmetry of the global matrix, its row, in general, has $3 + 9 + 9 = 21$ off-diagonal and diagonal non-zero elements. So the resulting grid equations are, in general, of 21-point type. The final system consists of equations of the form

$$p_0 u_0 - \sum_{k=1}^{21} p_k u_k = q_k,$$

where q_k is an approximation of the right-hand side of (4) for a certain component. Note that the dimension of the global matrix is $3N$, where N is the number of unknowns for one component of the displacement vector. Hence, the global matrix is a sparse matrix.

System (5) is iteratively solved by the EXIFMR method [2] which is a modification of the well-known minimum residuals method. The process stops when a fraction of the norms of the current and the initial residuals is less than a given small value.

3. Storage scheme and brief code description

To minimize the storage, we use a special sparse format for the global matrix representation. This storage scheme has only "calculatable" nodes and the nodes on the boundaries with the Dirichlet boundary conditions are eliminated from consideration and hence from the storage scheme. Every stored node has a thorough number in addition to its grid index (i, j, k). This numbering is made before carrying out approximation and is in excluding the Dirichlet nodes from consideration and associating every grid index with a new one thorough the number of a calculatable node. The sparse format has information for every calculatable node which consists of three groups: the number of calculatable neighbors (the integer array NE), their thorough numbers (the integer array NEIB), and coefficients of the link with these neighbors (a real diagonal array D and a real array for the off-diagonal entries AU). In addition, the format has a real array F for the values of the right-hand sides for the nodal equations.

In the program, this sparse format scheme excluding the values of the links with the neighbors is built before the approximation process starts using information about the boundary conditions in the domain. In doing so, the array with coefficients is zero and only its length is calculated.

After the approximation and assembling, many entries in the global matrix are zero ones (almost half the entries). To save the storage, the condensation procedure is applied to the sparse format. It is in excluding the information with zero values from the scheme.

The presented algorithms were implemented into the code using the programming language Fortran-90. The subroutines to read the input boundary value problem data, to construct the internal grid structure were taken from the 3D program package [3] and rewritten for dynamic Fortran-90 arrays, and subroutines implementing the presented approximation and forming the sparse matrix format we developed.

4. Numerical experiments

Test 1. The BVP for equation (1) with the right-hand side and side for a given substance concentration distribution for the constant temperature $T = T_0$ is considered. Here, instead of (3), we have

$$\sigma = 2\mu\varepsilon + \lambda \operatorname{div} \overline{u} I - \frac{2\mu + 3\lambda}{3}(\alpha(T - T_0) + N - 1).$$

The objective is to calculate a hydrostatic pressure value

$$H = \frac{2\mu + 3\lambda}{3}(\operatorname{div} \overline{u} - \alpha(T - T_0) - (N - 1))$$

having the displacement field.

For this test, the exact solution for the displacement field is known:

$$u = 1.31 \cdot 10^{-3} - \frac{2.62x}{2.5}, \quad v = 1.31 \cdot 10^{-3} - \frac{2.62y}{2.5}, \quad w = -\frac{2.62z \cdot 10^{-3}}{2.5},$$

and for this field $H = 0$.

The test computational domain is $[0, 2.5]^3$ cube. The Dirichlet boundary conditions are given on the bottom face of the cube, the rest faces being "free" surfaces without boundary conditions. The equation coefficients are $\lambda = 2.9 \cdot 10^4$, $\mu = 1.92 \cdot 10^4$, $\alpha = 6.3 \cdot 10^{-6}$, $N = 1 - 0.524 \cdot 10^{-3} \cdot 3$. The grid is uniform with the step $h = 0.5$. The stopping criterion for the iterative process is 10^{-8}.

The test results are the following. The grid consists of 216 nodes with 36 Dirichlet nodes. The number of calculatable nodes for one component is 180, and 540 unknowns in total. A length of the AU array is 864. If the exact solution is taken as initial guess, then the solution is sought for by 0 iterations, i.e., approximation gives the exact solution and $H = 0$ at all grid nodes. If the initial solution is zero, then we arrive at the solution by 33 iterations and have H of order 10^{-7}.

The same test on the cubic grid with the number of nodes in one direction 41 has the following characteristics. The number of grid nodes is 68921, the Dirichlet nodes – 1681, calculatable nodes for one component – 67240, the total number of unknowns – 201720, a length of AU array is 1390433. For the PC with 1.7 GHz processor and 256 Mb RAM, for the initial guess in the form of the exact solution it takes 6.2 seconds to construct the approximation and 1 second for the iterative process (to calculate the initial residual). For zero initial guess, it takes 6.4 seconds for the approximation and 52.8 seconds for 176 iterations to attain H of order 10^{-6}.

So, an expected theoretical result is obtained with the iterative accuracy.

Test 2. The computational region is $[0,3]^3$ cube. The exact solution to equation (2) is known

$$u = \sin\frac{\pi x}{3}\sin\frac{\pi y}{3}\sin\frac{\pi z}{3}, \quad v = \frac{u}{2}, \quad w = \frac{u}{4}.$$

The right-hand side is also known and is calculated by substituting the exact solution into the equation. The boundary conditions are zero Dirichlet conditions on all the faces of the cube according to the exact solution.

The aim of the test is to investigate approximation order, so the equation be solved on three embedded cubic grids with N steps in one direction equal to 15, 30, and 60. To show the order, the truncation error $\delta_t = \|t - t_h\| = \max_{ijk}\|t(x_i, y_j, z_k) - t^h_{ijk}\|$ for three displacement components is calculated after the iteration process stops.

The results for the zero initial guess are given in Table 1.

Test 3. This test has the computational region Ω in the form of a cube with a cubic hole: $\Omega = [0,3]^3 \setminus [1,2]^3$. The exact solution is taken in the form

$$u = \sin(\pi x)\sin(\pi y)\sin(\pi z), \quad v = \frac{u}{2}, \quad w = \frac{u}{4}.$$

The boundary conditions are the exact solution on all the faces of the region. Three grids are the same as in Test 2. The right-hand side of (2) is calculated via the exact solution. The results are shown in Table 2.

Table 1

N	δ_u	δ_v	δ_w	Num. of iter.
15	0.014	0.0068	0.0034	15
30	0.0035	0.0017	0.0087	38
60	0.00087	0.00043	0.00022	72

Table 2

N	δ_u	δ_v	δ_w	Num. of iter.
15	0.11	0.062	0.038	14
30	0.034	0.017	0.010	27
60	0.0086	0.0044	0.0027	54

One can make several conclusions from the tables:

1. As the truncation error decreases by, approximately, factor four with doubling the number of grid steps, the truncation error for both tests, 2 and 3, has order $O(h^2)$.
2. From Table 2 it follows that the error for the displacement components differs by factors 2 and 4, respectively, thus corresponding to the exact solution.
3. The error in Table 2 is, approximately, 10 times bigger than that of Table 1 because the period of the solution is three times less than the period of Test 2, and so it lacks grid points to attain the error of the higher order.

References

[1] Gurieva Y.L., Il'in V.P., On the finite volume technology for mixed boundary value problems // Proceedings of the International Conference AMCA-95. – Novosibirsk: NCC Publisher, 1995. – P. 650–655.

[2] Il'in V.P., Itskovich E.A. Two explicit incomplete factorization methods // NCC Bulletin. Series Num. Anal. – Novosibirsk: NCC Publisher, 2002. – Iss. 11. – P. 51–59.

[3] Gurieva Y.L., Il'in V.P. Program package for 3d boundary-value elliptic problem // NCC Bulletin. Series Num. Anal. – Novosibirsk: NCC Publisher, 2002. – Iss. 11. – P. 33–50.