УДК 004.2

O. Syniavsky, V. Kyslooky, O. Khomenko

A PARALLEL ALGORITHM FOR SOLVING THE FIRST BORDER VALUE PROBLEM OF ELASTICITY THEORY IN 3D SPACE BY MONTE CARLO METHOD

This work concerns the solution of the first border value problem of elasticity theory for 3-dimensional objects of any form and connectivity.

Our approach lies in the application of random walk on spheres method for finding the solution of the differential equations system of the 6-th order which corresponds to this problem. The suggested algorithm is parallel and was designed to be run on high-performance computing clusters.

Keywords: Random walk on spheres, Monte Carlo method, first border value problem, elasticity theory, parallel algorithm.

Introduction

A parallel algorithm for solving the first boundary value problem of elasticity theory is suggested. The problem exists in the case of an elastic body restrained over a part of its surface or closed within a rigid shell.

A continual variant of Monte Carlo method, namely the Random Walk on Spheres method, is used. This method has several features and benefits. It is an independent numerical method. Within this method, discretization is achieved not by replacing the continuous space with the discrete one, but by replacing the general population of trajectories of random walks with a partial selection of the trajectories. Monte Carlo method can be used to solve both deterministic and stochastic boundary value problems. Estimates for functionals of the solution (e.g. the functional of reliability) can also be evaluated. Simultaneously with the evaluation of the solution, its stochastic accuracy can be estimated. The significant benefits of Monte Carlo method also include the ease and efficiency of code parallelization when implementing on modern cluster architectures.

But the method of Random Walk on Spheres is well known only for boundary value problems of second order and is hard in generalization for higher-order boundary value problems.

To construct an algorithm for solving problems of elasticity theory by the Monte Carlo method, we have generalized the mean value theorem, well known in the theory of harmonic functions. Thus, in our case the Lame equations have a correspondent integral relation between the displacement vector in the center of a sphere and the displacements on its surface [1]. The relation was obtained from the solution of the elastic sphere deformation problem, in which a unit force is focused in the center of the sphere with restrained boundary.

Based on the relation, the solution of the problem has been obtained and can be estimated by means of the Random Walk on Spheres process.

1. Analytic PROBLEM STATEMENT

Let $G \in R^3$ be a limited finitely connected domain in 3-dimensional Euclidean space. Let symbol Γ denote the boundary of the given domain. A point in the space will be marked as x, where x has coordinates of (x_1, x_2, x_3).

The state of some point of the domain in the absence of mass forces is described by the classical static Lame equation [3]:

$$\Delta \mathrm{u}(x) + \frac{1}{1-2\nu}\, grad\ div\, \mathrm{u}(x) = 0, x \in G, \qquad (1)$$

where:

– $\mathrm{u}(x) = (u_1(x_1, x_2, x_3), u_2(x_1, x_2, x_3), u_3(x_1, x_2, x_3))$ is the displacement vector consisting of regular real-valued function;

– ν is Poisson's ratio, which characterizes the deformed material.

The first boundary value problem of elasticity theory for Lame equation lies in finding the vector function $\mathrm{u} \in \mathrm{C}^2(G) \cap \mathrm{C}(\bar{G})$ which satisfies the boundary condition:

$$\mathrm{u}(x) = \mathrm{g}(x), \quad x \in \Gamma, \qquad (2)$$

where $\mathrm{g} \in \mathrm{C}(G)$ is the given vector function whose values are the displacements on the domain boundary.

2. Integral representation of the solution

Represent the displacement vector in a point of elastic space through a surface integral over the displacements on the sphere, drawn around the point as the center. This representation is the basis of our stochastic scheme for solving spatial problems of elasticity theory by the method of statistical experiments.

Denote

– $u = (u_1, u_2, u_3)$ – displacement vector;

– $\sigma_n = (\sigma_{n1}, \sigma_{n2}, \sigma_{n3})$ – tension on the plain which is tangent to a sphere S with radius R;

– V – domain limited by sphere S;

– $X = (X_1, X_2, X_3)$ – vector of volumetric forces.

The target mode of deformation in the absence of mass forces is defined by the boundary conditions of the problem (2) and by Lame equation (1). Call this mode the main one. To obtain the integral representation of the displacement vector, consider an auxiliary mode of deformation. The latter will be the deformation mode of a sphere restrained over its surface. Let the auxiliary mode be generated by a unit force focused in the center of the sphere.

Now, it is possible to write the following integral equation which expresses the principle of reciprocal actions [5]:

$$\sum_{i=1}^{3}[\int_{s}(\sigma_{ni}^{*}u_i - \sigma_{ni}u_i^{*})dS + \\ + \int_{V}(X_i^{*}u_i - X_i u_i^{*})dV] = 0 \quad (3)$$

(asterisk denotes quantities relating to the auxiliary mode).

Expressions for the components of displacement and tension vectors inside ball V differ from the known fundamental solution of equations of the spatial problem of elasticity theory with certain summands that provide the absence of displacement on the surface of the ball. The following tensions arise on the restrained ball surface when a unit force is applied to its center x (x_1, x_2, x_3) along a coordinate axis j (j = 1, 2, 3):

$$\sigma_{nj}^{(j)} = \frac{3}{8-12\nu}\left(5\frac{x_j^2}{R^2} + 1 - 4\nu\right); \\ \sigma_{ni}^{(j)} = \frac{3}{8-12\nu}\left(5\frac{x_i x_j}{R^2}\right), \ (i \neq j); \quad (4)$$

From equation (3), having set $X_i^{(j)} = \sigma_{nj}\sigma(x)$, obtain:

$$\mathrm{u}(x) = \frac{1}{4\pi R^2}\int_{S_x} A(x,y)\,\mathrm{u}(y)dS_x, \quad (5)$$

where:

– S_x – sphere with center in point x (x_1, x_2, x_3);

– $y = y$ (y_1, y_2, y_3) – a point on sphere S_x.

Elements of matrix $A = (a_{ij})$ $(i, j = 1, 2, 3)$ are of the following form:

$$a_{ii} = \frac{3}{8-12\nu}\left[5\frac{(x_i - y_i)^2}{R^2} + 1 - 4\nu\right]; \\ a_{ij} = \frac{3}{8-12\nu}\left[5\frac{(x_i - y_i)(x_j - y_j)}{R^2}\right], \ (i \neq j). \quad (6)$$

Thus, we have obtained the integral representation which links the displacement in the center of the ball to the displacement on its surface. It expresses the mean value theorem for the spatial problem of elasticity theory. The matrix is the Green's function of the first boundary problem for the ball.

3. Algorithm for finding the solution

3.1. *Random Walk on Spheres process definition*

Before constructing the algorithm for finding the solution of the problem it is appropriate to outline the concept of Random Walk on Spheres process in three-dimensional space [2].

3.1.1. *Definition 1*

Having a certain domain G ∈R3 with boundary Γ and a point $x \in G \cup \Gamma$, we will call Sx the maximal three-dimensional sphere with center x and radius R if $R = inf_{y\in\Gamma} \| y - x \|$, where Sx is empty if $x \in \Gamma$.

3.1.2. *Definition 2*

Having a certain domain $G \in R^3$ with boundary Γ and a point $x \in G \cup \Gamma$, we will call $\Phi(x)$ Random Walk on Spheres process starting at point x if:

– $\Phi(x) = \{K(x,\varphi), 0 \le x \le 1\}$, i.e. $\Phi(x)$ is the set of all the sequences $K(x,\varphi)$ of points in three-dimensional space, where

– each value of φ defines a sequence of points $K(x,\varphi) = \{P_{i+1}(x,\varphi), i = 0,1,...\}$ generated in the following manner:

– around the point $P_0(x,\varphi) = x$ as the center, draw a maximum sphere S_{P_0};

– select a random point $P_1(x,\varphi)$ from a uniform distribution on sphere S_{P_0};

– point $P_{i+1}(x,\varphi) = x$ is defined recursively from point $P_i(x,\varphi)$ and sphere S_{P_i} in the same manner as point $P_1(x,\varphi)$ was defined after $P_0(x,\varphi)$.

3.2. *Building the algorithm*

Using the definition of spherical process and equation (5), which expresses the mean value theorem in spatial elasticity theory, construct the solution of the first boundary value problem for an internal point Q of domain G when the displacements on its boundary Γ are given.

We will call Γ_ε a subset of points in G which are at a distance of not more than ε from Γ. The displacement of points of ε-neighborhood Γ_ε is approximately equal to the displacement of the nearest boundary points. Since the maximum sphere is tangent to the boundary of the domain, usually only at a few points, the probability of transition from the center of the sphere to these boundary points is close to zero. Therefore, the ε-neighborhood of boundary has been introduced.

The proper value of ε depends on the gradient of function given on Γ and on the desired solution accuracy. Getting the analytical relationship between these quantities for an arbitrary spatial problem is very difficult. Therefore, in practice the value of ε is chosen as a part of the basic geometric size of the investigated domain.

To start with, draw the maximum sphere S_q

around the point Q. If the values of $u(Q_i)$ where known for all the $Q_i \in S_q$ the overall calculation would turn into integrating a known function over sphere S_q. The method of calculating definite integrals using statistical sampling is implemented in this case as the following procedure. Generate N random points $Q_i^0 (i = 1..N)$ uniformly distributed over the sphere S_q. Vector $u(Q)$, according to equation (5), can be calculated as an average of products $A(Q,Q_i^0)\,u(Q_i^0)$:

$$u(Q) \approx \frac{1}{N}\sum_{i=1}^{N}\xi_i, \text{ where } \xi_i = A(Q,Q_i^0)\,u(Q_i^0). \quad (7)$$

Actually, only the summands for such points:

$$Q_i^0 \in S_Q^\varepsilon = S_Q \bigcap \Gamma_\varepsilon$$

are defined in the sum (7). Therefore, if $Q_i^0 \in \Gamma_\varepsilon$ the value of random vector $\xi_i = A(Q,Q_i^0)\,u(Q_i^0)$ will be the approximate estimate of the i-th term of series (7). If, on the contrary, random point $Q_i^0 \notin \Gamma_\varepsilon$, then draw a new maximum sphere $S_{Q_i^0}$ and generate a random point Q_i^1 from a uniform distribution over sphere $S_{Q_i^0}$. If $Q_i^1 \in \Gamma_\varepsilon$ then set the respective i-th term of series (7) $\xi_i = A(Q,Q_i^0)\,A(Q_i^0,Q_i^1)\,u(Q_i^1)$. Proceed the process for point $Q_i^1 \notin \Gamma_\varepsilon$.

Thereby, trajectories which get to ε-neighborhood of the boundary on the k-th step would produce the following functional:

$$\xi_i = A(Q,Q_i^0)\,A(Q_i^0,Q_i^1)...A(Q_i^{k-2},Q_i^{k-1})\,u(Q_i^{k-1}).$$

In general, a trajectory of a random walk can be long, but with a probability equal to one, it ends in Γ_ε [2] after a finite number of steps. When building algorithms of Random Walk on Spheres, trajectory length should be limited with some number k. This number should be large enough so that with probability close to one the contributions of all trajectories that are possible for this area be taken into account.

3.3. *Sequential algorithm scheme*

The algorithm built above can be schematically presented in the form of such a sequence of steps:

A. GET the target point Q on the entry.

B. UNTIL the SUFFICIENT number N of trajectories not longer than k is achieved:

a. call SPHERICAL PROCESS starting at point Q and get result R;

b. ADD R to U;

c. INCREMENT the counter of valid trajectories.

C. RETURN U/N.

SPHERICAL PROCESS consists of the following sequence of steps:

A. GET Q as the starting point of trajectory on the entry.

B. If the maximum length of trajectory has been EXCEEDED then RETURN an empty result.

C. INCREMENT trajectory length.

D. If the given point is CLOSE to boundary (i.e. $Q \in \Gamma_\varepsilon$) then:

a. FIND the point $P \in \Gamma_\varepsilon$ which is closest to the given Q;

b. RETURN the given boundary value $g(P)$ as the result.

E. If the given point Q is FAR from the boundary (i.e. $Q \notin \Gamma_\varepsilon$) then:

a. draw maximum sphere S_Q around point Q;

b. GENERATE a uniformly random point T on S_Q;

c. Call SPHERICAL PROCESS recursively on point T and get its result R;

d. RETURN $A(Q,T)\cdot R$.

4. PaRALLEL version of the algorythm

The accuracy of the algorithm constructed above directly depends on the value of N. It specifies the number of random walks, i.e. the number of independent experiments that should be simulated. The result of each of these experiments gives a contribution to estimate the solution.

Achieving the sufficient accuracy of the solution requires the number N to be a large value. However, each of the N walks requires hundreds of arithmetic operations, as in all the steps of a random trajectory it is needed to find the distance from a point to the boundary, to generate a random point on a sphere, and to multiply a matrix by a vector. The above algorithm performs all of the N walks sequentially and spends a considerable amount of time to do that.

Applying parallel programming technologies and advanced computer capabilities will significantly improve the speed and efficiency of the proposed algorithm. Algorithms based on Random Walk on Spheres method have excellent opportunities for parallelization. All the N walks are independent of one another, so it is possible to simulate them simultaneously on multiple processors.

4.1. *Problem decomposition*

When constructing a parallel version of the above algorithm the most appropriate strategy is to apply functional decomposition of the problem. On the one hand, each of the N walks does not depend on other walks, and all of the experiments can be distributed evenly among the available processors (denote their number by P), according to decomposition by data. But on the other, the result of the algorithm depends on the successful completion of all of the N experiments and will not be ready until each processor performs its part of the task. In a parallel environment where the involved processors are not equal in the terms of speed or load, decomposition by data will cause significant delays and inefficiency. For this reason it is expedient to apply functional decomposition approach.

Thus, the N experiments can be divided into groups of n in each. Then the tasks for each proces-

sor will be sent in portions of size n, and will be substituted with new ones during the execution until there are N experiments done in total. P processors can be divided into two groups: 1 senior processor and $P - 1$ executive processors. The senior processor manages the work of the executive ones, loads them with tasks and performs aggregation of their results.

This strategy of decomposition allows to balance the load on processors optimally and to achieve the optimal performance.

4.2. *Parallel algorithm scheme*

Hence, we propose to introduce two different types of threads in the parallel algorithm: senior processor thread and executive processor thread. Their schemes are given below.

4.2.1. *Senior thread*

A. GET the target point Q on the entry.

B. SEND point Q to executive processes.

C. SEND the executives the first PORTION of n experiments.

D. UNTILL N experiments are done:

a. RECEIVE from an executive Ex an interim result T;

b. SEND Ex the next portion of n experiments;

c. STORE the interim result T ($U = U + T$);

d. INCREMENT the counter of ready experiments by n.

E. TERMINATE the executive processes by sending them an empty task.

F. RETURN the result of U/N.

4.2.2. *Executive thread*

A. RECEIVE the target point Q from the senior process.

B. RECEIVE a task t from the senior process.

C. If t is EMPTY then TERMINATE.

D. Run SPHERICAL PROCESS t times on point Q adding its result to R.

E. SEND the senior process the interim result R.

F. JUMP to STEP B.

5. PERFORMANCE ANALYSIS and time complexity

Analysis of the sequential algorithm time complexity T is presented in terms of the number of steps made by the spherical process, because the amount of operations t executed at each step is fixed and does not affect the asymptotic estimates.

6.1. *Dependence on the size of boundary ε-neighborhood*

Number T_f of trajectory steps on its way from the given point to the domain boundary is a random function of distance from this point to the boundary and of the ε-neighborhood size. In paper [4], it has been shown that the above spherical process has the time the following complexity:

$$T_f(\varepsilon) = O(|\ln\varepsilon|) \tag{8}$$

Hence, $T(\varepsilon) = O(T_f t) = O(|\ln\varepsilon|\, t) = O(|\ln\varepsilon|)$, (9)

where t is a constant time complexity of one step and doesn't depend on ε.

This means that the overall execution time of the algorithm is proportional to $|\ln\varepsilon|$ and goes to infinity when approaching ε to zero:

$$\lim_{\varepsilon\to 0} T(\varepsilon) = \lim_{\varepsilon\to 0} |\ln\varepsilon| = \infty \tag{10}$$

5.2. *Dependence on the number of experiments N*

Increasing the number N of experiments significantly improves the accuracy of the results. But we should determine how the parameter N affects the complexity of the algorithm. Denote by t the time complexity of an experiment, i.e. of drawing a random walk on the rules of the spherical process. As seen from the structure of the algorithm, time complexity of a walk does not depend on the total number of walks, i.e.

$$t(N) = const \tag{11}$$

Then, giving the time complexity of the whole algorithm as

$$T(N) = O(N \cdot t(N)) = O(N), \tag{12}$$

we come to the obvious conclusion that its complexity is a linear function of the number of experiments.

5.3. *Parallelization efficiency*

It is important to notice that the Random Walks on Spheres algorithm of Monte Carlo method is an algorithm of minimum connectivity (connectivity of an algorithm is the amount of data transmitted from the n-th step of the algorithm to the $(n + 1)$-th step). Due to this, the efficiency of its parallelization is very good in comparison with other methods of solving such problems.

Denote by P the number of processors involved in the algorithm calculation, by N – the size of the problem. Ideal (and therefore unattainable) parallelization efficiency of an algorithm whose sequential complexity is $T(N)$ leads to the complexity of

$$T(N,P) = \frac{T(N)}{P},$$

when running in parallel on P processors. Such performance is impossible, because a parallel implementation always adds a certain amount of time $S(N,P)$ spent on data exchange between involved processors.

In the case of the proposed parallel algorithm, $S(N,P)$ is a linear function of the size of the problem and the number of processors.

Denoting by n the size of a portion for one task, we can give such a time complexity estimate:

$$S(N,P) = \frac{N}{(P-1)\,n} c, \tag{14}$$

where c is time spent on one communication between two processes.

Indeed, given that, in this algorithm, tasks are

sent and received in portions of n walks each, the total of

$$\frac{N}{(P-1)\,n}$$

communications are needed for all of the N experiments to exchange messages between the senior and executive processes. Time c spent on one such exchange is a constant and is consumed for sending one integer value and three floating-point values.

Since work on portions of the tasks does not require any synchronization between the executive processes, the time consumed by the entire group of processors for parallel calculations is P – 1 times smaller than in the case of the sequential implementation. Thus, estimation for the time complexity of the parallel algorithm can be presented in the following form:

$$T(N,P) = \frac{T(N)}{(P-1)} + \frac{N}{(P-1)n}c \qquad (15)$$

This time complexity is quite good and demonstrates high parallelization efficiency, which fully justifies additional resources spent on development, implementation and launch of the respective software.

Література

1. Dekhtiaryuk, E.S. & Syniavsky, O.L. 1978. Solving elasticity theory problems by Monte Carlo method. 8-th International congress for applied mathematics in engineering, Weimar.
2. Muller M. E. 1956. Some continuous Monte Carlo method for the Dirichlet problem. Ann. Math. Stat., 27, (3).
3. Sabelfeld K. K. & Shalimova I. A. 1997. Spherical means for PDEs. VSP, Utrecht.
4. Sabelfeld, K. K. 1991. Monte Carlo Methods in Boundary Value Problems. Springer–Verlag, Berlin.
5. Varvaak, L. P., Dekhtiaryuk, E. S., Reutfarb, I. Z., Syniavsky, O. L. & Khimenko, V. V., 1973. Solving the spatial problem of elasticity theory by the method of statistical experiments. Resistance of materials and structural theory, 20, Kyiv.

Синявський О. Л., Кислоокий В. М., Хоменко О. І.

ПАРАЛЕЛЬНИЙ АЛГОРИТМ РОЗВ'ЯЗАННЯ ПЕРШОЇ КРАЙОВОЇ ЗАДАЧІ ТЕОРІЇ ПРУЖНОСТІ В 3-ВИМІРНОМУ ПРОСТОРІ ЗА МЕТОДОМ МОНТЕ-КАРЛО

Роботу присвячено розв'язанню першої крайової задачі теорії пружності для 3-вимірних тіл довільної форми та зв'язності.

Наш підхід полягає у розробленні методу блукання сферами для знаходження розв'язку системи диференційних рівнянь 6-го порядку, яка відповідає цій задачі. Запропонований алгоритм є паралельним і розрахованим на виконання на високопродуктивних обчислювальних кластерах.

Ключові слова: алгоритм блукання сферами, метод Монте-Карло, перша крайова задача, теорія пружності, паралельний алгоритм.

Матеріал надійшов 14 червня 2011 р.