KYBERNETIKA — VOLUME *43* (2007), NUMBER 6, PAGES 767 – 776

EFFECTIVE COMPUTATION OF RESTORING FORCE VECTOR IN FINITE ELEMENT METHOD

Martin Balazovjech and Ladislav Halada

We introduce a new way of computation of time dependent partial differential equations using hybrid method FEM in space and FDM in time domain and explicit computational scheme. The key idea is quick transformation of standard basis functions into new simple basis functions. This new way is used for better computational efficiency. We explain this way of computation on an example of elastodynamic equation using quadrilateral elements. However, the method can be used for more types of elements and equations.

Keywords: FEM, stiffness matrix, restoring force vector, computational efficiency of algorithm, e-invariants

AMS Subject Classification: 65M60

1. INTRODUCTION

Modelling of seismic waves propagation and earthquake ground motion has advanced to a state when we want to calculate realistic 3D models. As a rule, elastic wave is considered as satisfactory approximation of these waves for computational modeling. Elastic wave propagation is governed by the elastodynamic equation. Due to complicated geometry of the boundary domain and heterogeneity of the area, modeling of seismic wave propagation can be realistic enough only if the computational domain is very large (computer memory more than 100 GB). One of the dominant method applicable to solving of the elastodynamic equation into such domain is the Finite Element Method (FEM).

The paper is organized as follows: Section 2 establishes the problem formulation. Section 3 presents the two formulation of local restoring force vector computation and the number of arithmetical operation required for both type of computation. Section 4 point out on the usefulness of e-invariants which save only the essential information needed for the computation.

2. FEM APPLIED TO THE EQUATION OF MOTION

Let us consider the wave propagation in two-dimensional perfectly elastic medium. Then the wave propagation in both x and y coordinates satisfy the equation of

motion and Hooke's law [3, 5]

$$\begin{aligned} \rho u_{x,tt} &= \tau_{xx,x} + \tau_{xy,y} + f_x \\ \rho u_{y,tt} &= \tau_{yx,x} + \tau_{yy,y} + f_y \end{aligned}, \quad \begin{bmatrix} \tau_{xx} \\ \tau_{yy} \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} u_{x,x} \\ u_{y,y} \\ u_{x,y} + u_{y,x} \end{bmatrix}, \tag{1}$$

where ρ is mass density, u_i and f_i are components of displacement and body force per unit volume, respectively, τ_{ij} are components of the stress tensor in $i, j = x, y$ direction. λ and μ are Lamé elastic coefficients. The computational domain is $\bar{\Omega} = \Omega \cup \Gamma$, where Ω represent the interior and Γ a the border of the domain. Usually, solution of such differential equations is subjected also to specified boundary and initial conditions. The weak formulation of equations (1) is based on the known standard three-step procedure with final equations

$$\begin{aligned} \int_\Omega (w_k \rho u_{x,tt} + w_{k,x}\, \tau_{xx} + w_{k,y}\, \tau_{xy} - w_k f_x)\, \mathrm{d}\Omega - \oint_\Gamma (w_k \tau_{xn})\, \mathrm{d}\Gamma = 0, \\ \int_\Omega (w_k \rho u_{y,tt} + w_{k,x}\, \tau_{yx} + w_{k,y}\, \tau_{yy} - w_k f_y)\, \mathrm{d}\Omega - \oint_\Gamma (w_k \tau_{yn})\, \mathrm{d}\Gamma = 0, \end{aligned} \tag{2}$$

where $w_k \in \{w_1, w_2, w_3, \ldots, w_\infty\}$ are basis functions defined in the computational domain. In sequel, we will use Galerkin weighted residual method and isoparametric form of numerical solution computed by decomposition of domain into quadrilateral elements. In this case the transformation between the actual element given by the coordinates $\mathbf{x} = (x_1, x_2, x_3, x_4)^T$, $\mathbf{y} = (y_1, y_2, y_3, y_4)^T$ and the master element has the form

$$x = \mathbf{b}^T \mathbf{x}\,, \qquad y = \mathbf{b}^T \mathbf{y}\,, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} (1-\eta)\,(1-\xi) \\ (1-\eta)\,(1+\xi) \\ (1+\eta)\,(1-\xi) \\ (1+\eta)\,(1+\xi) \end{bmatrix}, \tag{3}$$

where the components of vector $\mathbf{b}$ are Lagrange family interpolation functions defined on the square $\langle -1, 1\rangle \times \langle -1, 1\rangle$. Approximation of the dependent variables u_x and u_y on the master element Ω^e is again expressed as

$$u_x = \mathbf{b}^T \mathbf{u}_x\,, \qquad u_y = \mathbf{b}^T \mathbf{u}_y \tag{4}$$

where $\mathbf{u}_x = (u_{x1}, u_{x2}, u_{x3}, u_{x4})^T$ and $\mathbf{u}_y = (u_{y1}, u_{y2}, u_{y3}, u_{y4})^T$ are values of dependent variable at the nodes of element Ω^e. Applying relation (4) to the integral equations (2) we obtain the matrix form of equation (2) for the element Ω^e

$$\begin{bmatrix} \mathbf{M} & 0 \\ 0 & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{x,tt} \\ \mathbf{u}_{y,tt} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} = \frac{1}{\rho} \begin{bmatrix} \mathbf{M} & 0 \\ 0 & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{f}_x \\ \mathbf{f}_y \end{bmatrix} + \begin{bmatrix} \mathbf{bc}_x \\ \mathbf{bc}_y \end{bmatrix}. \tag{5}$$

Vectors $\mathbf{f}$ and $\mathbf{bc}$ express external forces and boundary conditions, respectively. The matrices of equations (5) are the local mass matrix composed of

$$\mathbf{M} = \rho \int_{-1}^{1} \int_{-1}^{1} \left(\mathbf{b}\mathbf{b}^T \det \mathbf{J} \right) \mathrm{d}\eta\, \mathrm{d}\xi \tag{6}$$

and the local stiffness matrix composed of

$$\begin{aligned}
\mathbf{K}_{11} &= \int_{-1}^{1}\int_{-1}^{1} \left(\mathbf{b}_{,x}\,(\lambda+2\mu)\,\mathbf{b}_{,x}^{T} + \mathbf{b}_{,y}\,\mu\mathbf{b}_{,y}^{T}\right) \mathrm{Det}\,\mathbf{J}\,\mathrm{d}\eta\,\mathrm{d}\xi,\\
\mathbf{K}_{12} &= \int_{-1}^{1}\int_{-1}^{1} \left(\mathbf{b}_{,x}\qquad\quad \lambda\mathbf{b}_{,y}^{T} + \mathbf{b}_{,y}\,\mu\mathbf{b}_{,x}^{T}\right) \mathrm{Det}\,\mathbf{J}\,\mathrm{d}\eta\mathrm{d}\xi,\\
\mathbf{K}_{22} &= \int_{-1}^{1}\int_{-1}^{1} \left(\mathbf{b}_{,y}\,(\lambda+2\mu)\,\mathbf{b}_{,y}^{T} + \mathbf{b}_{,x}\,\mu\mathbf{b}_{,x}^{T}\right) \det\,\mathbf{J}\,\mathrm{d}\eta\mathrm{d}\xi,
\end{aligned} \tag{7}$$

where numerical integration has been calculated by Gauss quadrature. Naturally, equation (5) holds only for a quadrilateral element of the domain. To solve the problem on the whole domain, we have to make so called assembling of all elements in the domain. This way we obtain the system of equations

$$\mathbb{M}\mathfrak{u}_{tt} = \mathbb{K}\mathfrak{u} + \mathfrak{f}, \tag{8}$$

where $\mathbb{M}$ and $\mathbb{K}$ are global mass and stiffness matrices, respectively. $\mathfrak{f}$ is the source component. Now, such system of equations can be solved using approximation of the second time derivative by the central difference formula. As the result we obtain an explicit recurrent relation for the unknown variable $\mathfrak{u}^{m+1}$ in the time $t = (m+1)\,\Delta t$.

$$\mathfrak{u}^{m+1} = \Delta t^2\mathbb{M}^{-1}\left(\mathbb{K}\mathfrak{u}^{m} + \mathfrak{f}\right) - \mathfrak{u}^{m-1} + 2\mathfrak{u}^{m}. \tag{9}$$

In practical applications the order of $\mathbb{M}$ and $\mathbb{K}$ can be quite large $(10^4 - 10^6)$ [4]. Especially, in such cases attention has to be paid to formulation of such a computational algorithm that economize storage and number of required computational operations. However, there are also situations, when the global mass and stiffness matrices are too large to be stored in computer memory and numerical solution of the original problem can not be found by finite element method. Therefore, efforts to overcome this problem can be found in the literature. For example, the global mass matrix can be approximated by a diagonal matrix (lumped mass matrix) and beside using the global stiffness matrix some authors suggest to use a global restoring force vector $\mathfrak{r}^m = \mathbb{K}\mathfrak{u}^m$. It means that the stiffness matrix is not computed in order to reduce the storage memory, but rather the restoring force vector $\mathfrak{r}^m$ is computed in every time level without saving stiffness matrix in memory. In other words we do not assemble the global stiffness matrix using local matrices but we assemble global restoring forces using local restoring forces. This procedure is described in [1] Moreover, direct computation of restoring forces has a further advantage in the case of nonlinear material response [2]. As the result, this computational process reduces the storage, but is very time consuming. In the following section we suggest how to formulate the computation of restoring force vector to be the most effective from the computational point of view.

3. COMPUTATION OF THE LOCAL RESTORING FORCE VECTOR USING NEW BASIS FUNCTIONS

The local restoring force vector for a quadrilateral element is given by

$$\begin{bmatrix}\mathbf{r}_x\\ \mathbf{r}_y\end{bmatrix} = \begin{bmatrix}\mathbf{K}_{11}\mathbf{u}_x + \mathbf{K}_{12}\mathbf{u}_y\\ \mathbf{K}_{12}^T\mathbf{u}_x + \mathbf{K}_{22}\mathbf{u}_y\end{bmatrix}. \tag{10}$$

Substituting (7) for $\mathbf{K}_{ij},\ i,j=1,2$ we obtain

$$\begin{aligned}\mathbf{r}_x &= \int_{-1}^{1}\int_{-1}^{1}\left(\mathbf{b}_{,x}\,(\lambda+2\mu)\,\mathbf{b}_{,x}^T\,\mathbf{u}_x + \mathbf{b}_{,y}\,\mu\left(\mathbf{b}_{,y}^T\,\mathbf{u}_x + \mathbf{b}_{,x}^T\,\mathbf{u}_y\right)\right.\\ &\qquad\left. + \mathbf{b}_{,x}\,\lambda\mathbf{b}_{,y}^T\,\mathbf{u}_y\right)\det\,\mathbf{J}\,\mathrm{d}\eta\mathrm{d}\xi, \qquad (11)\\ \mathbf{r}_y &= \int_{-1}^{1}\int_{-1}^{1}\left(\mathbf{b}_{,y}\,(\lambda+2\mu)\,\mathbf{b}_{,y}^T\,\mathbf{u}_y + \mathbf{b}_{,x}\,\mu\left(\mathbf{b}_{,y}^T\,\mathbf{u}_x + \mathbf{b}_{,x}^T\,\mathbf{u}_y\right)\right.\\ &\qquad\left. + \mathbf{b}_{,y}\,\lambda\mathbf{b}_{,x}^T\,\mathbf{u}_x\right)\det\,\mathbf{J}\,\mathrm{d}\eta\mathrm{d}\xi. \qquad (12)\end{aligned}$$

Now, we will proceed as suggested in [1]. First of all, let us express the first derivatives of the functions $\mathbf{u}_x$ and $\mathbf{u}_y$ from (4). It holds

$$\begin{aligned}u_{x,x} &= \mathbf{b}_{,x}^T\,\mathbf{u}_x, \quad u_{x,y} = \mathbf{b}_{,y}^T\,\mathbf{u}_x,\\ u_{y,x} &= \mathbf{b}_{,x}^T\,\mathbf{u}_y, \quad u_{y,y} = \mathbf{b}_{,y}^T\,\mathbf{u}_y.\end{aligned} \tag{13}$$

Using these values we can compute the stress in quadrature points

$$\begin{aligned}\tau_{xx} &= (\lambda+2\mu)\,u_{x,x} + \lambda u_{y,y}\,,\\ \tau_{xy} &= \mu\,(u_{x,y} + u_{y,x})\,,\\ \tau_{yy} &= (\lambda+2\mu)\,u_{y,y} + \lambda u_{x,x}\end{aligned} \tag{14}$$

and finally the restoring force in nodes

$$\begin{aligned}\mathbf{r}_x &= \textstyle\int_{-1}^{1}\int_{-1}^{1}(\tau_{xx}\mathbf{b}_{,x} + \tau_{xy}\mathbf{b}_{,y})\det\,\mathbf{J}\,\mathrm{d}\eta\mathrm{d}\xi,\\ \mathbf{r}_y &= \textstyle\int_{-1}^{1}\int_{-1}^{1}(\tau_{xy}\mathbf{b}_{,x} + \tau_{yy}\mathbf{b}_{,y})\det\,\mathbf{J}\,\mathrm{d}\eta\mathrm{d}\xi.\end{aligned} \tag{15}$$

The derivative of the basis functions can be computed by help of the relation

$$\begin{bmatrix}b_{i,x}\\ b_{i,y}\end{bmatrix} = \frac{1}{\det\,\mathbf{J}}\begin{bmatrix}y_{,\eta} & -y_{,\xi}\\ -x_{,\eta} & x_{,\xi}\end{bmatrix}\begin{bmatrix}b_{i,\xi}\\ b_{i,\eta}\end{bmatrix}, \tag{16}$$

where Jacobian $\mathbf{J}$ of the spatial transformation is

$$\mathbf{J} = \begin{bmatrix}x_{,\xi} & x_{,\eta}\\ y_{,\xi} & y_{,\eta}\end{bmatrix} = \frac{1}{4}\begin{bmatrix}x_1 & x_2 & x_3 & x_4\\ y_1 & y_2 & y_3 & y_4\end{bmatrix}\begin{bmatrix}-\eta m & -\xi m\\ \eta m & -\xi p\\ -\eta p & \xi m\\ \eta p & \xi p\end{bmatrix} \tag{17}$$

$$= \frac{1}{4}\begin{bmatrix}(x_2-x_1)\,\eta m + (x_4-x_3)\,\eta p & (x_3-x_1)\,\xi m + (x_4-x_2)\,\xi p\\ (y_2-y_1)\,\eta m + (y_4-y_3)\,\eta p & (y_3-y_1)\,\xi m + (y_4-y_2)\,\xi p\end{bmatrix}, \tag{18}$$

whereby

$$\begin{aligned}\eta m &= 1-\eta, \quad \eta p = 1+\eta,\\ \xi m &= 1-\xi, \quad \xi p = 1+\xi\end{aligned} \tag{19}$$

and the derminant of Jacobian of the spatial transformation (4) is equal to

$$\det\,\mathbf{J} = x_{,\xi}\,y_{,\eta} - x_{,\eta}\,y_{,\xi}\,. \tag{20}$$

A new process of the local restoring force vector computation is based on the expression of values x, y, u_x, u_y be help of a new basis functions and consequently on using computational e-invariants composed from quadrilateral element parameters. Let $\mathbf{b}^{(\mathrm{inv})}$ be a new vector of basis functions on the unit square and $\mathbf{T}$ be a transform matrix

$$\mathbf{b}^{(\mathrm{inv})} = \begin{bmatrix}1\\ \xi\\ \eta\\ \eta\xi\end{bmatrix} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix}+1 & +1 & +1 & +1\\ -1 & +1 & -1 & +1\\ -1 & -1 & +1 & +1\\ +1 & -1 & -1 & +1\end{bmatrix} \tag{21}$$

with the property

$$\mathbf{b}^{(\mathrm{inv})} = \mathbf{T}\mathbf{b} \quad \text{and} \quad \tfrac{1}{4}\mathbf{T}\mathbf{T}^T = \mathbf{I}, \tag{22}$$

where $\mathbf{I}$ is unit matrix. These properties are very important because, as we will see, they facilitate to express the value of the function in nodal points using the new basis functions. The standard isoparametric formulation of the function values in the grid points of the element Ω^e is

$$\mathbf{c} = \begin{bmatrix}x\\ y\\ u_x\\ u_y\end{bmatrix} = \begin{bmatrix}x_1 & x_2 & x_3 & x_4\\ y_1 & y_2 & y_3 & y_4\\ u_{x1} & u_{x2} & u_{x3} & u_{x4}\\ u_{y1} & u_{y2} & u_{y3} & u_{y4}\end{bmatrix}\begin{bmatrix}b_1\\ b_2\\ b_3\\ b_4\end{bmatrix} = \mathbf{V}\mathbf{b}. \tag{23}$$

The vector $\mathbf{c}$ can be expressed in the new basis functions as follows:

$$\mathbf{c} = \mathbf{V}\mathbf{b} = \mathbf{V}\mathbf{I}\mathbf{b} = \mathbf{V}\left(\tfrac{1}{4}\mathbf{T}^T\mathbf{T}\right)\mathbf{b} = \left(\mathbf{V}\tfrac{1}{4}\mathbf{T}^T\right)(\mathbf{T}\mathbf{b}) = \mathbf{V}^{(\mathrm{inv})}\mathbf{b}^{(\mathrm{inv})} \tag{24}$$

and it means

$$\mathbf{c} = \begin{bmatrix}x\\ y\\ u_x\\ u_y\end{bmatrix} = \begin{bmatrix}x^{(0)} & x^{(1)} & x^{(2)} & x^{(12)}\\ y^{(0)} & y^{(1)} & y^{(2)} & y^{(12)}\\ u_x^{(0)} & u_x^{(1)} & u_x^{(2)} & u_x^{(12)}\\ u_y^{(0)} & u_y^{(1)} & u_y^{(2)} & u_y^{(12)}\end{bmatrix}\begin{bmatrix}1\\ \xi\\ \eta\\ \eta\xi\end{bmatrix}, \tag{25}$$

where components $x^{(k)}, y^{(k)}, u_x^{(k)}, u_y^{(k)}, \quad k = 0, 1, 2, 12$ of matrix $\mathbf{V}^{(\mathrm{inv})}$ are some e-invariants with suitable properties from the computational point of view.

Let us express the restoring force vector using the new basis functions, too. We will proceed step by step according to relations (13) – (15). It holds

$$u_{x,x} = \mathbf{b}_{,x}^T\,\mathbf{u}_x = \mathbf{b}_{,x}^T\left(\tfrac{1}{4}\mathbf{T}^T\mathbf{T}\right)\mathbf{u}_x = \left(\mathbf{b}_{,x}^T\,\mathbf{T}^T\right)\left(\tfrac{1}{4}\mathbf{T}\mathbf{u}_x\right) = \mathbf{b}_{,x}^{(\mathrm{inv})T}\,\mathbf{u}_x^{(\mathrm{inv})}. \tag{26}$$

In analogy with the way of the formulation (26) we obtain

$$\begin{aligned} u_{x,y} &= \mathbf{b}_{,y}^{(\mathrm{inv})T}\,\mathbf{u}_{x}^{(\mathrm{inv})}, \\ u_{y,x} &= \mathbf{b}_{,x}^{(\mathrm{inv})T}\,\mathbf{u}_{y}^{(\mathrm{inv})}, \\ u_{y,y} &= \mathbf{b}_{,y}^{(\mathrm{inv})T}\,\mathbf{u}_{y}^{(\mathrm{inv})}. \end{aligned} \tag{27}$$

Using these values we can compute the stress in quadrature points int the same way as in (14) and for $\mathbf{b}_{,\mathbf{x}}$ we can derive from (22)

$$\mathbf{b}_{,x} = \mathbf{T}^{-1}\mathbf{b}_{,x}^{(\mathrm{inv})} = \tfrac{1}{4}\mathbf{T}^{T}\mathbf{b}_{,x}^{(\mathrm{inv})}\,. \tag{28}$$

If these values are substituted to (14) and then to (15), we obtain

$$\begin{aligned} \mathbf{r}_x &= \tfrac{1}{4}\mathbf{T}^T \int_{-1}^{1}\int_{-1}^{1} \left(\tau_{xx}\mathbf{b}_{,x}^{(\mathrm{inv})} + \tau_{xy}\mathbf{b}_{,y}^{(\mathrm{inv})}\right) \det\,\mathbf{J}\,\mathrm{d}\eta\mathrm{d}\xi, \\ \mathbf{r}_y &= \tfrac{1}{4}\mathbf{T}^T \int_{-1}^{1}\int_{-1}^{1} \left(\tau_{xy}\mathbf{b}_{,x}^{(\mathrm{inv})} + \tau_{yy}\mathbf{b}_{,y}^{(\mathrm{inv})}\right) \det\,\mathbf{J}\,\mathrm{d}\eta\mathrm{d}\xi. \end{aligned} \tag{29}$$

For the derivative of vector elements $\mathbf{b}^{(\mathrm{inv})}$ with respect to x and y we have

$$\mathbf{b}_{,x}^{(\mathrm{inv})} = \begin{bmatrix} 0 \\ \xi_{,x} \\ \eta_{,x} \\ (\eta\xi)_{,x} \end{bmatrix}, \quad \mathbf{b}_{,y}^{(\mathrm{inv})} = \begin{bmatrix} 0 \\ \xi_{,y} \\ \eta_{,y} \\ (\eta\xi)_{,y} \end{bmatrix}, \tag{30}$$

where

$$\begin{bmatrix} \xi_{,x} & \eta_{,x} \\ \xi_{,y} & \eta_{,y} \end{bmatrix} = \frac{1}{\det\,\mathbf{J}} \begin{bmatrix} y_{,\eta} & -y_{,\xi} \\ -x_{,\eta} & x_{,\xi} \end{bmatrix} \tag{31}$$

and for the Jacobian we now have

$$\mathbf{J} = \begin{bmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \end{bmatrix} = \begin{bmatrix} x^{(0)} & x^{(1)} & x^{(2)} & x^{(12)} \\ y^{(0)} & y^{(1)} & y^{(2)} & y^{(12)} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ \eta & \xi \end{bmatrix} = \begin{bmatrix} x^{(1)} + \eta x^{(12)} & x^{(2)} + \xi x^{(12)} \\ y^{(1)} + \eta y^{(12)} & y^{(2)} + \xi y^{(12)} \end{bmatrix}. \tag{32}$$

Thus, we have two possibilities how to compute the local restoring force vector. They are represented by the relations (15) and (29). Note that application of relation (29) is very simple for the case of a unit square element. As an example, let us consider a mesh of unit squares into a part of the computational domain. Let this mesh be generated be the translation of unit square. Then the spatial transformation between the master element and arbitrary square element becomes

$$\begin{aligned} x &= \tfrac{1}{4}\left(x_1 + x_2 + x_3 + x_4\right) + \xi = x^{(0)} + \xi, \\ y &= \tfrac{1}{4}\left(y_1 + y_2 + y_3 + y_4\right) + \eta = y^{(0)} + \eta. \end{aligned} \tag{33}$$

It can be seen easily that for the Jacobian and its determinant

$$\mathbf{J} = \begin{bmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \det\,\mathbf{J} = 1 \tag{34}$$

hold, and the derivative of the vector $\mathbf{b}^{(\mathrm{inv})}$ with respect to x and y has a very simple result

$$\mathbf{b}^{(\mathrm{inv})}_{,x} = \begin{bmatrix} 0 \\ \xi_{,x} \\ \eta_{,x} \\ (\eta\xi)_{,x} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \eta \end{bmatrix} \qquad \mathbf{b}^{(\mathrm{inv})}_{,y} = \begin{bmatrix} 0 \\ \xi_{,y} \\ \eta_{,y} \\ (\eta\xi)_{,y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \xi \end{bmatrix}. \tag{35}$$

Thus, using the following formulations (36) and (37) we can compute derivatives of displacements and stress components for the square element

$$\begin{aligned} u_{x,x} &= u_x^{(1)} + \eta u_x^{(12)}, \quad u_{x,y} = u_x^{(2)} + \xi u_x^{(12)}, \\ u_{y,x} &= u_y^{(1)} + \eta u_y^{(12)}, \quad u_{y,y} = u_y^{(2)} + \xi u_y^{(12)}, \end{aligned} \tag{36}$$

$$\begin{aligned} \tau_{xx} &= (\lambda + 2\mu)\left(u_x^{(1)} + \eta u_x^{(12)}\right) + \lambda\left(u_y^{(2)} + \xi u_y^{(12)}\right), \\ \tau_{xy} &= \mu\left(u_x^{(2)} + \xi u_x^{(12)} + u_y^{(1)} + \eta u_y^{(12)}\right), \\ \tau_{yy} &= (\lambda + 2\mu)\left(u_y^{(2)} + \xi u_y^{(12)}\right) + \lambda\left(u_x^{(1)} + \eta u_x^{(12)}\right). \end{aligned} \tag{37}$$

The local restoring force vector is obtained by exact evaluation of integrals (29). The result is very simple

$$\mathbf{r}_x = \mathbf{T}^T \begin{bmatrix} 0 \\ (\lambda + 2\mu)\, u_x^{(1)} + \lambda u_y^{(2)} \\ \mu\left(u_x^{(2)} + u_y^{(1)}\right) \\ \frac{1}{3}(\lambda + 3\mu)\, u_x^{(12)} \end{bmatrix}, \quad \mathbf{r}_y = \mathbf{T}^T \begin{bmatrix} 0 \\ \mu\left(u_x^{(2)} + u_y^{(1)}\right) \\ (\lambda + 2\mu)\, u_y^{(2)} + \lambda u_x^{(1)} \\ \frac{1}{3}(\lambda + 3\mu)\, u_y^{(12)} \end{bmatrix}. \tag{38}$$

Algorithm efficiency is often measured by the number of arithmetical operations required by the algorithm. In the following table the number of required operations is given for computation of the local restoring force vector for a quadrilateral element.

Case of algorithm	dividing	multiplying	add/substr.	assigning
Standard	4	236	236	196
Proposed (Q4 element)	4	169	150	127
Proposed (square element)	0	11	32	29

In the next section we briefly describe the reason why suggested algorithm is more effective.

4. e-INVARIANTS OF THE QUADRILATERAL GEOMETRY

e-invariants saved only essential information needed for the computation of the local restoring force vector. Two groups of invariants were used: e-invariants of the quadrilateral geometry and e-invariants of the set of unknown values. e-invariants of the quadrilateral geometry uniquely determine convex quadrilateral in the plane.

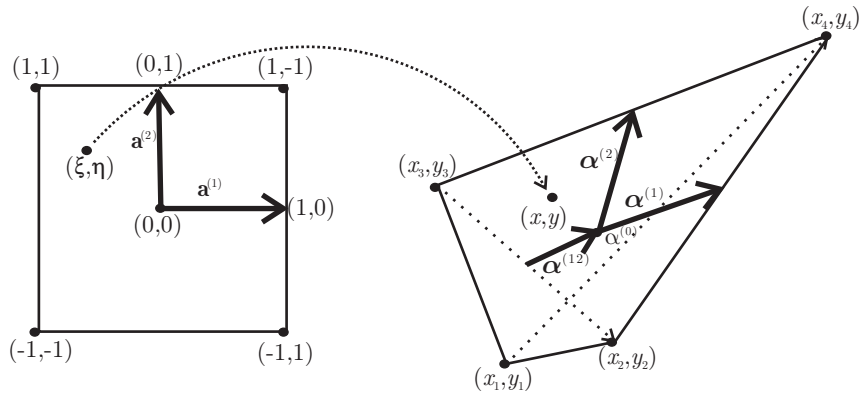

Fig 1. Bilinear transformation.

The first e-invariants $x^{(0)}$ and $y^{(0)}$ generate the point $\alpha^{(0)}$ in the quadrilateral domain. This point correspond with $(0,0)$ point into the master element in bilinear transformation shown in Figure 1. Coordinates of point $\alpha^{(0)}$ are

$$x^{(0)} = \tfrac{1}{4}\left(x_1 + x_2 + x_3 + x_4\right), \quad y^{(0)} = \tfrac{1}{4}\left(y_1 + y_2 + y_3 + y_4\right). \tag{39}$$

$\alpha^{(0)}$ determine the location of the quadrilateral element in computational domain independently on the rotation and change of shape of the element. Location $\alpha^{(0)}$ of the element is not needed for the computation of the restoring force vector.

Another two e-invariants are vectors $\boldsymbol{\alpha}^{(1)} = \left(x^{(1)}, y^{(1)}\right)$ and $\boldsymbol{\alpha}^{(2)} = \left(x^{(2)}, y^{(2)}\right)$. These vectors correspond with the vectors $\mathbf{a}^{(1)} = (1,0)$ and $\mathbf{a}^{(2)} = (0,1)$ into the master element. Their coordinates are as follows

$$\begin{aligned} x^{(1)} &= \tfrac{1}{4}\left(-x_1 + x_2 - x_3 + x_4\right), \quad y^{(1)} = \tfrac{1}{4}\left(-y_1 + y_2 - y_3 + y_4\right), \\ x^{(2)} &= \tfrac{1}{4}\left(-x_1 - x_2 + x_3 + x_4\right), \quad y^{(2)} = \tfrac{1}{4}\left(-y_1 - y_2 + y_3 + y_4\right). \end{aligned} \tag{40}$$

It is easy to verify that these coordinate fulfill

$$\begin{aligned} x^{(1)} &= \tfrac{1}{2}\left(x_2 + x_4\right) - x^{(0)}, \qquad y^{(1)} = \tfrac{1}{2}\left(y_2 + y_4\right) - y^{(0)}, \\ x^{(2)} &= \tfrac{1}{2}\left(x_3 + x_4\right) - x^{(0)}, \qquad y^{(2)} = \tfrac{1}{2}\left(y_3 + y_4\right) - y^{(0)}. \end{aligned} \tag{41}$$

Positions of the vectors $\boldsymbol{\alpha}^{(1)}$ and $\boldsymbol{\alpha}^{(2)}$ are shown in Figure 1.

Both vectors are independent on the location of the quadrilateral element. Their lengths

$$\left|\boldsymbol{\alpha}^{(1)}\right| = \sqrt{x^{(1)}x^{(1)} + y^{(1)}y^{(1)}}, \qquad \left|\boldsymbol{\alpha}^{(2)}\right| = \sqrt{x^{(2)}x^{(2)} + y^{(2)}y^{(2)}} \tag{42}$$

and angel

$$\boldsymbol{\beta} = \operatorname{Arcos}\left(\frac{x^{(1)}y^{(1)} + x^{(2)}y^{(2)}}{\left|\boldsymbol{\alpha}^{(1)}\right|\left|\boldsymbol{\alpha}^{(2)}\right|}\right) \tag{43}$$

are independent on the rotation. The last invariants are $x^{(12)}$ and $y^{(12)}$. They generate the vector $\boldsymbol{\alpha}^{(12)}$, which is always non-zero for a general case of convex quadrilateral. This vector correspond with the zero vector in the master element. The coordinates of this vector are

$$x^{(12)} = \tfrac{1}{4}\left(x_1 - x_2 - x_3 + x_4\right), \quad y^{(12)} = \tfrac{1}{4}\left(y_1 - y_2 - y_3 + y_4\right) \tag{44}$$

or

$$x^{(12)} = x^{(0)} - \tfrac{1}{2}\left(x_2 + x_3\right), \quad y^{(12)} = y^{(0)} - \tfrac{1}{2}\left(y_2 + y_3\right). \tag{45}$$

Size of this vector is

$$\left|\boldsymbol{\alpha}^{(12)}\right| = \sqrt{x^{(12)}x^{(12)} + y^{(12)}y^{(12)}}. \tag{46}$$

This value is independent of element location and can be used as a rate of how much differs general convex quadrilateral element from parallelogram.

5. CONCLUSION

We presented a new way of restoring force vector computation in FEM. The proposed method differs from the standard algorithms using such basis functions which enable to compute this vector more effective. It means that the computation of the local restoring force vector for quadrilateral element using the 4-points numerical quadrature reduce the number of the required arithmetical operation approximately about $\frac{1}{3}$. This method can be easily extended to 3D case. The result of this generalization will be the subject of our next paper.

ACKNOWLEDGEMENT

The authors thank Peter Moczo for useful comments and valuable discussions regarding effective and accurate numerical computation and modeling of seismic waves propagation.

(Received November 30, 2006.)

REFERENCES

[1] R. J. Archuleta: Experimental and Numerical Three-dimensional Simulations of Strike-slip Earthquakes. Ph.D. Thesis. University of California, San Diego 1976.

[2] T. Belytscho, W. K. Liu, and B. Moran: Nonlinear Finite Elements for Continua and Structures. Wiley, New York 2000

[3] C. A. Felippa: Introduction to Finite Element Methods (ASEN 5007). Lecture notes. University of Colorado, Boulder 2005.

[4] P. Moczo, J. Kristek, and E. Bystrický: Efficiency and optimization of the 3D finite-difference modeling of seismic ground motion. J. Comp. Acoustics *9(2)* (2004), 593–609.

[5] J. N. Reddy: An Introduction to the Finite Element Method, McGraw-Hill, New York 1993.

Martin Balazovjech, Department of Astronomy Physics of the Earth and Meteorology, Comenius University, Mlynská dolina, 842 48 Bratislava. Slovak Republic.
e-mail: balazovjech@fmph.uniba.sk

Ladislav Halada, Geophysical Institute, Slovak Academy of Sciences, Dubravská cesta, 842 28 Bratislava. Slovak Republic.
e-mail: ladislav.halada@savba.sk