INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 24, 1253–1270 (1997)

A THREE-DIMENSIONAL EXPLICIT ELEMENT-FREE GALERKIN METHOD

T. BELYTSCHKO,[1,]* P. KRYSL[2] AND Y. KRONGAUZ[3]

[1]*Civil and Mechanical Engineering,* [2]*Civil Engineering,* [3]*Theoretical and Applied Mechanics, Northwestern University,*
Evanston, IL 60208, U.S.A.

SUMMARY

The formulation and implementation of a three-dimensional meshless method, the element-free Galerkin (EFG) method, are described. The formulation is intended for dynamic problems with geometric and material nonlinearities solved with explicit time integration, but some of the developments are applicable to other solution methods. The mechanical formulation is posed in the reference configuration so that the shape functions and their derivatives need to be computed only once. A method for speeding up the calculation of shape functions and their derivatives is presented. Results are presented for sloshing problems and Taylor bar impact problems, including an impact problem in which the bar impacts with an angle of obliquity. ©1997 by John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **24**: 1253–1270, 1997

No. of Figures: 12. No. of Tables. 0: No. of References: 24.

KEY WORDS: meshless methods; sloshing; computational fluid dynamics; impact

1. INTRODUCTION

The implementation and application of a meshless method, the element-free Galerkin (EFG) method, for three-dimensional dynamic problems with explicit time integration are described. EFG is a meshless method in the sense of the definition given by Oñate *et al.*:[1] the approximation is constructed entirely in terms of a set of scattered nodes without recourse to any elements or zones. The advantages of meshless methods are manifold: (i) the need to generate a mesh of nodes and elements is eliminated—only nodes need to be scattered in the solid, which is generally much easier; (ii) the treatment of moving discontinuities such as cracks and shocks is facilitated, since no new mesh needs to be constructed as in finite element methods.[2–7] Current work in these methods also shows many potential advantages: (i) certain types of locking tend not to be as pronounced as in low-order finite elements, e.g. volumetric locking seldom appears; (ii) adaptivity is far easier to implement; (iii) it is easier to enrich approximations with closed-form solutions, such as near-tip crack fields. For these reasons, considerable interest has developed in meshless methods in recent years. Reviews of these methods have recently been written by Duarte and Oden[8] and Belytschko *et al.*,[7] so we will not summarize the literature here.

This paper is concerned with the implementation of EFG for three-dimensional dynamic problems with explicit time integration. Since the simulation of problems with explicit time integration usually

*Correspondence to: T. Belytschko, Civil and Mechanical Engineering, Northwestern University, Evanston, IL 60208, U.S.A.

CCC 0271–2091/97/121253–18 $17.50

Received 8 May 1996
Revised 18 October 1996

requires many time steps, the cost per time step should be kept as low as possible. We describe several steps we have taken to minimize the computation time: (i) the implementation of the equations in a material setting so that most of the shape functions are only computed at the beginning of the calculation; (ii) the use of a background mesh for volume integrations in combination with a low order of quadrature; (iii) an empirical optimization of the domains of influence of the weight functions so that a reasonable trade-off is achieved between accuracy and speed.

The paper is organized as follows. In Section 2 we describe the governing equations in a reference (or material) setting. Section 3 describes the generation of the approximation functions and their derivatives in the EFG procedure. A method for constructing the approximation functions is described. Although the EFG approximations are usually derived from the concept of a moving least squares approximation, we will here describe a method which is based on ensuring the consistency of the approximations. The consistency is imposed by modifying a kernel by a linear combination of four vectors which exactly satisfy linear consistency. The approach yields an approximation which is identical with that generated by moving least squares but leads to more efficient formulae for the derivatives.

In Section 4 we describe the implementation of the EFG procedure: the generation of the element background mesh, coupling to finite elements, etc. Section 5 presents some results of three-dimensional calculations. Two problems have been selected: the Taylor impact problem and fluid sloshing. For the latter some two-dimensional calculations are also reported. In the Taylor problem a new variant is proposed in which an angle of obliquity is included in the impact so that the response is not rotationally symmetric. We conclude with some remarks on the performance of the method and suggestions for future work.

2. GOVERNING EQUATIONS

Kinematics

We consider a three-dimensional body $\mathscr{B}$ which is an open set in the Euclidean space $\mathbb{R}^3$. The body consists of material points X. The material points can be identified with co-ordinates in a fixed Cartesian system, with basis vectors $\mathbf{e}_k, k=1,2,3$, in a reference configuration $\mathscr{B}^0$, i.e. the material point X is identified with the position vector $\mathbf{X}=\sum_k X_k\mathbf{e}_k$. The Cartesian co-ordinate system $\mathbf{e}_k$ will be used exclusively, both for the reference and for the current configurations. Because of the fact, the distinction between covariant and contravariant components is superfluous.

The motion of the body $\mathscr{B}$ is described by the mapping $\boldsymbol{\chi}$, $\mathbf{x}=\boldsymbol{\chi}(\mathbf{X},t)$, where $\mathbf{x}$ is the co-ordinate of the material point X in the current configuration, $\mathbf{x}=x_k\mathbf{e}_k$.

The velocity and acceleration of a material point are obtained by material differentiation with respect to time as $\mathbf{v}(\mathbf{X},t)=\partial\boldsymbol{\chi}(\mathbf{X},t)/\partial t$ and $\mathbf{a}(\mathbf{X},t)=\partial^2\boldsymbol{\chi}(\mathbf{X},t)/\partial t^2$. The deformation gradient $\mathbf{F}$ is defined by using the operator $\boldsymbol{\nabla}^0=(\partial(\cdot)/\partial X_k)\mathbf{e}_k$ as $\mathbf{F}=\boldsymbol{\nabla}^0\otimes\boldsymbol{\chi}=(\partial\chi_i/\partial X_j)\mathbf{e}_i\otimes\mathbf{e}_j$.

Conservation equations

We are dealing with a purely mechanical theory of continuous media. The basic equations are the conservation of mass,

$$\rho^0=\rho\det\mathbf{F}, \tag{1}$$

and energy conservation,

$$\rho^0\frac{\partial\mathscr{E}}{\partial t}=\mathbf{P}:\frac{\partial\mathbf{F}}{\partial t}, \tag{2}$$

where $\mathscr{E}$ is the internal energy per unit mass, **P** is the first Piola–Kirchhoff stress tensor and the operator ‘**:**’ is a simultaneous contraction on both indices, defined for tensors $\mathbf{p} = p_{ij}\mathbf{e}_i \otimes \mathbf{e}_j$ and $\mathbf{q} = q_{ij}\mathbf{e}_i \otimes \mathbf{e}_j$ as $\mathbf{p} : \mathbf{q} = p_{ij}q_{ij}$. The conservation of linear momentum gives

$$\rho^0 \mathbf{a} = \rho^0 \mathbf{b} + \boldsymbol{\nabla}^0 \cdot \mathbf{P}, \tag{3}$$

where **b** is the body force per unit mass, and the conservation of angular momentum yields

$$\mathbf{F} \cdot \mathbf{P}^{\mathrm{T}} = \mathbf{P} \cdot \mathbf{F}^{\mathrm{T}}. \tag{4}$$

The governing system of equations is completed by constitutive equations with appropriate strain measures.

Boundary and initial conditions

The boundary conditions can be specified as any combination of prescribed displacements on $\partial^1\mathscr{B}$,

$$\mathbf{u}(\mathbf{X}, t) = \bar{\mathbf{u}}(\mathbf{X}, t), \quad \mathbf{X} \in \partial^1\mathscr{B}, \tag{5}$$

and prescribed tractions on $\partial\mathscr{B}$,

$$\mathbf{P}(\mathbf{X}, t) \cdot \mathbf{n}^0(\mathbf{X}) = \bar{\mathbf{t}}(\mathbf{X}), \quad \mathbf{X} \in \partial\mathscr{B}, \tag{6}$$

where $\partial^1\mathscr{B}$ and $\partial\mathscr{B}$ correspond to disjoint subsets of the boundary $\partial\mathscr{B}, \partial\mathscr{B} = \partial^1\mathscr{B} \bigcup \partial\mathscr{B}$.

The solution is sought given the initial conditions, which specify the initial velocity and stresses in $\mathscr{B}$, i.e.

$$\mathbf{v}(\mathbf{X}, 0) = \bar{\mathbf{v}}(\mathbf{X}), \quad \mathbf{X} \in \mathscr{B}, \tag{7}$$

$$\mathbf{P}(\mathbf{X}, 0) = \bar{\mathbf{P}}(\mathbf{X}) = \boldsymbol{\sigma}(\mathbf{X}, 0), \quad \mathbf{X} \in \mathscr{B}, \tag{8}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, which in the reference configuration is identical with the first Piola–Kirchhoff stress tensor.

Virtual work principle

The discrete formulation is obtained from the weak form, the principle of virtual work. It can be stated on the reference configuration $\mathscr{B}$ (i.e. in the material setting) in terms of the dependent variable $\mathbf{u} \in \mathscr{U}$ and the first Piola–Kirchhoff stress tensor **P** as

$$\int_{\mathscr{B}} \rho^0 \mathbf{a} \cdot \delta\mathbf{u} \, \mathrm{d}V = \int_{\mathscr{B}} \rho^0 \mathbf{b} \cdot \delta\mathbf{u} \, \mathrm{d}V - \int_{\mathscr{B}} \mathbf{P} : \boldsymbol{\nabla}^0 \otimes \delta\mathbf{u} \, \mathrm{d}V + \int_{\partial\mathscr{B}} \delta\mathbf{u} \cdot \bar{\mathbf{t}} \, \mathrm{d}A, \tag{9}$$

where $\mathscr{B}$ is the reference domain, ρ^0 is the mass density in the reference domain, **a** is the material acceleration and $\delta\mathbf{u} \in \mathscr{U}_0$ is the virtual displacement, with $\mathscr{U} = \{\mathbf{u} | \mathbf{u}(\mathbf{X}, t) \in C^0, \mathbf{u}(\mathbf{X}, t) = \bar{\mathbf{u}}$ for $\mathbf{X} \in \partial^1\mathscr{B}\}$ and $\mathscr{U}_0 = \{\mathbf{u} | \mathbf{u}(\mathbf{X}) \in C^0, \mathbf{u}(\mathbf{X}) = \mathbf{0}$ for $\mathbf{X} \in \partial^1\mathscr{B}\}$.

Element-free Galerkin approximation

The motion parameters of the material point X, i.e. the current co-ordinate **x** (or displacements $\mathbf{u} = \mathbf{x} - \mathbf{X}$), velocity **v** and acceleration **a**, are approximated in the EFG method by using the moving least squares[7] shape functions $\phi_I(\mathbf{X})$ as

$$\mathbf{u}(\mathbf{X}, t) = \sum_I \phi_I(\mathbf{X})\mathbf{u}_I(t), \tag{10a}$$

$$\mathbf{v}(\mathbf{X}, t) = \sum_I \phi_I(\mathbf{X})\mathbf{v}_I(t), \tag{10b}$$

$$\mathbf{a}(\mathbf{X}, t) = \sum_I \phi_I(\mathbf{X})\mathbf{a}_I(t), \tag{10c}$$

We wish to stress that $\mathbf{u}_I, \mathbf{v}_I$ and $\mathbf{a}_I$ are not the nodal values of displacements (velocities, etc.), but rather nodal parameters without a direct physical interpretation, because the shape functions $\phi_I(\mathbf{X})$ produce approximation, not interpolation.

The partial derivatives with respect to the referencing co-ordinates X_k are obtained simply as

$$\frac{\partial \mathbf{x}(\mathbf{X}, t)}{\partial X_k} = \sum_I \frac{\partial \phi_I(\mathbf{X})}{\partial X_k}\mathbf{x}_I(t). \tag{11}$$

Substitution of (10a) and (11) into (9) gives the relations for the mass matrix and internal forces as

$$\delta W^{\text{int}} = \int_{\mathscr{B}} P : \nabla^0 \otimes \delta\mathbf{u}\, \mathrm{d}V = \sum_I \delta\mathbf{u}_I \cdot \mathbf{f}_I^{\text{int}}, \tag{12a}$$

$$\delta W^{\text{ext}} = \int_{\mathscr{B}} \rho^0 \mathbf{b} \cdot \delta\mathbf{u}\, \mathrm{d}V + \int_{\partial\mathscr{B}} \delta\mathbf{u} \cdot \mathbf{t}^0\, \mathrm{d}A = \sum_I \delta\mathbf{u}_I \cdot \mathbf{f}_I^{\text{ext}}, \tag{12b}$$

$$\delta K = \int_{\mathscr{B}} \rho^0 \mathbf{a} \cdot \delta\mathbf{u}\, \mathrm{d}V = \sum_{I,J} \delta\mathbf{u}_I \cdot \mathbf{M}_{IJ} \cdot \mathbf{a}_J. \tag{12c}$$

3. CONSISTENCY APPROACH TO MESHLESS APPROXIMATIONS

Although the calculations reported here deal with homogeneous bodies with smooth solutions, the EFG implementation described here is eventually aimed at problems with moving discontinuities such as shocks or cracks. Therefore, even for a total Lagrangian formulation, recalculation of some of the shape functions and their derivatives is unavoidable. We describe here a methodology that streamlines the calculations.

The EFG approximation is formulated in terms of the moving least squares approximation (MLS), but for purposes of fast derivative evaluation it is convenient to describe it as a consistent version of a localized approximation; the final result is identical with the MLS approximation.

Consider a weight function with compact support $w(\mathbf{X} - \mathbf{X}_I) = w_I(\mathbf{X})$ associated with node I. We wish to compute the shape function $\phi_I(\mathbf{X})$ so as to be able to write an approximation to a function $u(\mathbf{X})$ in the form $u(\mathbf{X}) \approx \sum_I \phi_I(\mathbf{X})u_I$. Since we intend to use the shape functions in a Galerkin procedure for a second-order problem, we require the linear consistency equations to be satisfied. These consistency conditions are automatically satisfied if

$$\sum_I \phi_I(\mathbf{X}) = 1 \quad \text{and} \quad \sum_I \phi_I(\mathbf{X})\mathbf{X}_I = \mathbf{X}. \tag{13}$$

The sum in (13) is over those nodes whose domain of influence includes $\mathbf{X}$.

We will construct the shape function from the weight function $w_I(\mathbf{X})$ by multiplying the weight function by a correction function[7–9] and by enforcing the consistency conditions (13). Thus we choose for the shape function the form

$$\phi_I(\mathbf{X}) = C(\mathbf{X})w_I(\mathbf{X}), \tag{14}$$

where $C(\mathbf{X})$ is the correction function. The correction function will be sought in the form

$$C(\mathbf{X}) = \mathbf{a}(\mathbf{X})^{\mathrm{T}}\mathbf{g}(\mathbf{X}_I), \tag{15}$$

where $\mathbf{g}(\mathbf{X})$ is a column matrix of m linearly independent functions ($m \geqslant 4$ for 3D domains) and $\mathbf{a}$ is a column matrix of m coefficients. The functions $\mathbf{g}$ can be chosen to improve approximations; for example, Fleming *et al.*[10] have recently used functions corresponding to the asymptotic crack tip field. Since in generally there may be more functions g_i than there are linear consistency conditions, equations (13) will be completed to represent reproducibility conditions of the m functions $\mathbf{g}(\mathbf{X})$:

$$\sum_I \phi_I(\mathbf{X})\mathbf{g}(\mathbf{X}_I) = \mathbf{g}(\mathbf{X}). \tag{16}$$

It is obvious that $\mathbf{g}(\mathbf{X})$ should include $1, X, Y$ and Z to be able to satisfy the linear consistency requirements. The consistency conditions (13) can also be viewed as reproducibility conditions for $1, X, Y$ and Z.

The unknown coefficients $\mathbf{a}$ can be solved for us by using the reproducibility conditions of the functions $\mathbf{g}(\mathbf{X})$, i.e. substituting (14) and (15) into (16) gives

$$\sum_I w_I(\mathbf{X})[\mathbf{g}(\mathbf{X}_I)\mathbf{g}^T(\mathbf{X}_I)]\mathbf{a}(\mathbf{X}) = \mathbf{g}(\mathbf{X}). \tag{17}$$

Equation (17) can be cast in a form identical with that arrived at by the moving least squares technique (compare e.g. with Reference 7), i.e.

$$\sum_I \sum_j w_I(\mathbf{X}) g_i(\mathbf{X}_I) g_j(\mathbf{X}_I) a_j(\mathbf{X}) = g_i(\mathbf{X}) \tag{18}$$

or

$$\mathbf{A}(\mathbf{X})\mathbf{a}(\mathbf{X}) = \mathbf{g}(\mathbf{X}), \tag{19}$$

where $\mathbf{A}$ is given by

$$A_{ij} = \sum_I w_I(\mathbf{X}) g_i(\mathbf{X}_I) g_j(\mathbf{X}_I). \tag{20}$$

The matrix $\mathbf{A}$ is symmetric and positive definite and is often called the moment matrix. The explicit expressions of (14) and (15) and of the matrix $\mathbf{A}$ can be written for a linear basis in one-, two- and three-dimensional problems respectively as

$$\phi_I(X) = [a_0(X) + a_1(X)X_I]w_I(x), \tag{21a}$$

$$\mathbf{A}(X) = \sum_I w_I(X)\begin{bmatrix} 1 & X_I \\ X_I & X_I^2 \end{bmatrix} = \sum_I w_I(x)\mathbf{A}_I, \tag{21b}$$

$$\mathbf{A}_{,X}(X) = \sum_I w_{I,X}(X)\mathbf{A}_I, \tag{21c}$$

$$\phi_I(\mathbf{X}) = [a_0(\mathbf{X}) + a_1(\mathbf{X})X_I + a_2(\mathbf{X})Y_I]w_I(\mathbf{X}), \tag{22a}$$

$$\mathbf{A}(\mathbf{X}) = \sum_I w_I(\mathbf{X})\mathbf{A}_I, \tag{22b}$$

$$\mathbf{A}_I = \begin{bmatrix} 1 & X_I & Y_I \\ X_I & X_I^2 & X_I Y_I \\ Y_I & X_I Y_I & Y_I^2 \end{bmatrix}, \tag{22c}$$

$$\mathbf{A}_{,i}(\mathbf{X}) = \sum_I w_{I,i}(\mathbf{X})\mathbf{A}_I \tag{22d}$$

and

$$\phi_I(\mathbf{X}) = [a_0(\mathbf{X}) + a_1(\mathbf{X})X_I + a_2(\mathbf{X})Y_I + a_3(\mathbf{X})Z_I]w_I(\mathbf{X}), \tag{23a}$$

$$\mathbf{A}(\mathbf{X}) = \sum_I w_I(\mathbf{X}) \begin{bmatrix} 1 & X_I & Y_I & Z_I \\ X_I & X_I^2 & X_I Y_I & X_I Z_I \\ Y_I & X_I Y_I & Y_I^2 & Y_I Z_I \\ Z_I & X_I Z_I & Y_I Z_I & Z_I^2 \end{bmatrix} = \sum_I w_I(\mathbf{X})\mathbf{A}_I, \tag{23b}$$

$$\mathbf{A}_{,i}(\mathbf{X}) = \sum_I w_{I,i}(\mathbf{X})\mathbf{A}_I. \tag{23c}$$

The shape function can be computed at a given point **X** by solving (19). The shape functions are then written by combining (19) with (15) and (14) as

$$\phi_I(\mathbf{X}) = [\mathbf{A}^{-1}(\mathbf{X})\mathbf{f}(\mathbf{X})]^{\mathrm{T}}\mathbf{a}(\mathbf{X}_I)w_I(\mathbf{X}). \tag{24}$$

The above derivation is closely related to the procedure of imposing reproducing conditions by means of a correction function as proposed by Liu *et al.*[9]

The matrix inversion in (24) is to be understood symbolically; the matrix **A** is factorized either by LU or by QR decomposition. The derivatives of the shape functions are calculated by an approach described in Reference 11 which speeds up the computations. The required equations are obtained by differentiating (14) and by noting that the differentiation of (19) yields

$$\mathbf{A}_{,i}\mathbf{a} + \mathbf{A}\mathbf{a}_{,i} = \mathbf{g}_{,i}, \tag{25}$$

where $g_{,i}$ denotes $\partial g/\partial X_i$. Thus we can obtain the derivatives of $\mathbf{a}_{,i}$ by solving

$$\mathbf{A}\mathbf{a}_{,i} = \mathbf{g}_{,i} - \mathbf{A}_{,i}\mathbf{a}. \tag{26}$$

For this purpose the factorization of **A** computed when solving (19) can be reused, so the computation of the derivatives involves little extra computation.

The savings in the computation times recorded by the authors for the construction of the EFG shape functions can be approximately 50% in two-dimensional problems and even more in 3D cases. The initial computation of the shape functions at all integration points, which include the search for the nodes which influence a given quadrature point, is about 15 times as expensive as an explicit update for an elastic material. The savings achieved by the presented technique are therefore significant.

4. IMPLEMENTATION OF THE EFG PROCEDURE

Central difference time stepping

To advance the solution in time, we have chosen the central difference scheme in the classical form.[12] In the *n*th step, given velocities $\dot{\mathbf{u}}_{n-1/2} = \dot{\mathbf{u}}(t_n - \Delta_n t/2)$ and displacements $\mathbf{u}_{n-1} = \mathbf{u}(t_{n-1})$ (with the variable time step $\Delta_n t = (\Delta_{n+1/2} t + \Delta_{n-1,2} t)/2)$:

1. Calculate velocities at time $t_n + \Delta_n t/2$:

$$\dot{\mathbf{u}}_{n+1/2} = \dot{\mathbf{u}}_{n-1/2} + \Delta_n t \mathbf{M}^{-1} \cdot (\mathbf{f}_n^{\mathrm{ext}} - \mathbf{f}_n^{\mathrm{int}}).$$

2. Calculate displacements at time $t_{n+1} = t_n + \Delta_n t$:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta_n t \dot{\mathbf{u}}_{n+1/2}.$$

The symbols in the above formulae are $\mathbf{M}$, the mass matrix (constant and diagonal), $\mathbf{u}_n$ and $\dot{\mathbf{u}}_n$, the vectors of displacements and velocities respectively, $\mathbf{f}_n^{\text{ext}}$, the external loads, and $\mathbf{f}_n^{\text{int}}$, the nodal forces corresponding to the stresses, all at time step n.

Stable time step

The stable time step can be computed for an elastic isotropic material under small deformations (this stability estimate is usually applied also to finite deformations) from the smallest distance between nodes, $d_{\min}$, over the speed of a dilatational wave, $c = \sqrt{[E/\rho(1-2\nu)]}$ (E is the Young elastic modulus and ν is the Poisson ratio), resulting in

$$\Delta t \leqslant c/d_{\min}. \qquad (27)$$

The penalty enforcement of contact which is used in the impact problems discussed below reduces the stable time step.

Coupling to finite elements

The EFG shape functions can be modified to mimic ordinary finite element shape functions on the boundary of the EFG domain.[14] Thus one can either enforce essential boundary conditions as with regular finite elements or, as will be demonstrated below, one can use this feature to couple EFG domains to finite elements in a single computational model.

The basic idea is to model the EFG domain as a finite element, or rather as a super-element since there are internal degrees of freedom involved. The EFG super-element interacts with the rest of the mesh through finite element nodes located at the interfaces between the EFG domain and the adjacent finite elements. An example of the incorporation of an EFG super-element is given in Figure 1. The coupling technique requires that EFG nodes be placed at the vertices of the integration cells at the interface between the super-element and other finite elements. These EFG nodes then have a dual identity: they are both EFG nodes (as seen from the point of view of the EFG domain) and FE nodes (as seen from the point of view of the adjacent finite elements) (they are marked in Figure 1 by both crosses and circles). In what follows, the EFG nodes which are not involved in the coupling are denoted as pure EFG nodes (these are marked by crosses only). The integration cells which have one or more dual FE/EFG nodes at their vertices are called interface cells. The shape functions are modified in the interface cells to be the regular FE shape functions at a part of the cell boundary. One

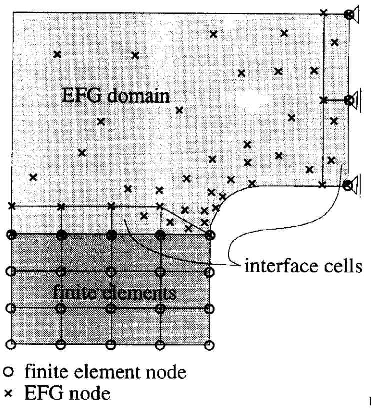


Figure 1. EFG super-element in an FE domain

important implication of this is that the shape functions in the interior of the interface cells are a blend of finite-element and EFG shape functions.

The EFG super-element defines the momentum equations at the pure EFG nodes as local. In other words, it hides these equations from the rest of the computational model. These equations need to be advanced in time, however, and the simple loops over finite elements and finite element nodes to update the configuration, compute the internal forces, etc. must be augmented by loops over super-elements to allow them to perform the needed operations on their local (private) degrees of freedom. Thus, for example, the effective nodal forces in the central difference algorithm are in ordinary finite element programmes computed in an element loop such as

```
FOR(all loads L in domain D) {assemble_ext_loads(D, L);}
FOR(all elements E in domain D) {assemble_restoring_forces(D, E);}
```

Programmes implementing super-elements should augment the code of the above fragment by

```
...
FOR(all super-elements SE in domain D) {calc_eff_loads(D, SE);}
```

to allow the super-element to assemble external and restoring forces corresponding to the internal degrees of freedom of the super-element.

Discretization of the EFG domain

Geometry approximation. The geometry of the EFG domain in the present work is approximated by a collection of hexahedral cells. The FE/EFG coupling requires that cells placed along the boundaries shared with finite elements (so-called 'interface cells') be compatible with the adjacent finite elements. Otherwise the geometric subdivision into cells is quite arbitrary. Thus the cells may be generated e.g. by octrees, multiblock mesh generators or directly from the geometric components constituting a CAD model.

Placement of EFG nodes. The nodes are generated independently of the integration cells, with the exception of the interface cells where the EFG nodes must be present at the vertices to define the ramp function.[14]

Volume integrals. The volume integrals (mass matrix, restoring forces) are obtained by using either regular Gaussian quadrature or special two-point diagonal quadrature[15] at the Gauss points $\xi_i = -1/\sqrt{3}$ and $1/\sqrt{3}$. Our experience with the quadrature scheme suggests that a lower order-quadrature (one-point or $2 \times 2 \times 2$ in hexahedral cells) with smaller cells may be preferable to larger cells with high-order quadrature. The reason seems to be the quite complicated variation in the shape function derivatives, which makes it difficult to integrate them by the usual high-order Gaussian quadrature.

Surface integrals. The surface integrals (load, contacts) are evaluated independently of the cell structure.

Size of the domain of influence. The choice of the sizes of the supports (domains of influence) is quite crucial. From the point of view of the computation costs the supports should be chosen as small as possible, since the time to compute any kinematic quantity (deformation gradients, strain tensors, etc.) is proportional to the number of nodes affecting each integration point. For example, in a regular cubic grid with spacing *a* in each direction the radii of circular domains of influence are chosen to be a multiple of the length of the diagonal of the cube, $\sqrt{3}a$. The integration cells are the cubes having the nodes as vertices. The average number of nodes at an integration point for a $2 \times 2 \times 2$ quadrature is 20 for $d_{\mathrm{m}} = \sqrt{3}a$, 59 for $d_{\mathrm{m}} = 1{\cdot}5 \times \sqrt{3}a$ and 124 for $d_{\mathrm{m}} = 2 \times \sqrt{3}a$. The computing time within a step grows correspondingly in the ratios 1:2·95:6·2.

From the point of view of accuracy there is an optimum support size. Unfortunately, the optimum is not easy to compute and, in addition, it varies with the characteristics of the problem at hand; see the discussion for fourth-order problems by Krysl and Belytschko.[16] The reader should realize, however, that only absolute accuracy is involved, not the rate of convergence.

Mass matrix. To exploit the potential speed of the explicit time-stepping technique, the mass matrix should be diagonal. We have computed the mass matrix by the row sum technique. The mass matrix of the *I*th EFG node is a multiple of the unit matrix, $\mathbf{M}_I = M_I \mathbf{1}$. Because the shape functions of all EFG nodal points affecting a given point sum to one, i.e. $\sum_I \phi_I = 1$, the factors M_I can be computed as

$$M_I = \int_{\mathscr{B}} \rho^0 \phi_I \, \mathrm{d}V. \tag{28}$$

It should be noted that the factors M_I are independent of material deformation and consequently the mass matrix is constant in time.

Internal forces. The EFG method seems to be free of numerical locking both for bending and for almost incompressible materials such as large-strain plasticity and rubber hyperelasticity. This means that the kinematic relations need not be modified in order to avoid it as, for example, in finite elements integrated by a reduced or selective scheme (stabilized, finite elements),[17] enhanced strain finite elements,[18] etc. This makes the task of defining a material model for use with the EFG super-element appealingly simple. The kinematic quantities are computed by straightforward substitution of (10a) and (11) into the appropriate definition of the kinematic tensor (deformation gradient, rate of deformation, Green–Lagrange strain tensor, etc.). Thus the interface between the integration point and the material model is very simple and allows for any material.

The restoring forces are computed from equation (12a). As can be seen, in order to be able to use a particular material model with the EFG domain, the material model must be able to compute at a

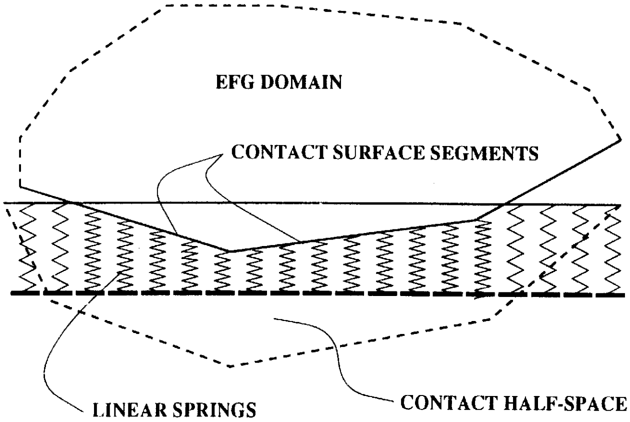


Figure 2. Definition of penetrated half-space as a Winkler foundation

given time the first Piola–Kirchhoff stress tensor. The resulting internal forces are assembled either to the internal equations for purely EFG nodes or to the global equations for EFG/FE nodes.

Material laws. The Taylor bar impacts simulations in this paper were based on the assumption that the material is elastoplastic with linear hardening, with the classical J_2 flow rule. The stress update procedure is based on the method of instantaneous final rotation of Nagtegaal and Veldpaus.[19]

Bending and incompressibility locking. It is a well-known fat that the eight-noded, fully integrated, isoparametric finite element locks for incompressible materials and also tends to produce over-stiff response in bending. Previous numerical experiments indicate that the EFG method is not susceptible to numerical locking for almost incompressible materials and is not excessively stiff in binding.[2]

The EFG shape functions are of relatively high order when the supports are larger. This makes the approximation less constrained and is probably the reason for the absence of locking. On the other hand, it is well known that the EFG shape functions resemble the regular finite element shape functions for smaller domains of influence, although the finite element shape functions are in general not recovered for more than one-dimensional spaces. As a consequence, one would expect difficulties with locking. However, as our numerical experiments show, this is not the case. It can probably be explained as follows. Considering again the above regular cubic grid, the smallest possible radius of a spherical domain of influence is $d_{\mathrm{m}} = \sqrt{3}a$; smaller domains of influence usually give badly conditioned shape functions and it becomes impossible to construct the shape functions at some locations. The resulting approximation is still much less constrained than for hexahedral, fully integrated finite elements, since an integration points still has 20 neighbouring nodes on average, whereas there are only eight neighbours for any integration point in the trilinear finite elements.

Locking for large-strain plasticity may actually appear in the interface cells between the EFG subdomain and the finite elements. The reason is that if a trilinear ramp is used to blend the finite element shape functions and the EFG shape functions, the shape functions in the interior of the interface cells are rather similar to the regular finite element shape functions, which are known to produce an over-stiff response.

Parallelization. The EFG super-element has been integrated into a finite element programme which has been parallelized by the authors for workstation and supercomputer networks as described in Reference 20. The domain decomposition used in Reference 20 is based on finite elements and the EFG super-element can be included in the finite element model without any modification. However, it is possible that a need might arise to assign the EFG domain to more than one computer, e.g. when it is too large to fit onto a single machine. The solution could be to split the original EFG super-element into two or more super-elements by introducing interfaces. This is not always feasible in large-strain plasticity with respect to locking, however; see the preceding section.

Nevertheless, since the EFG super-elements are currently more expensive than high-performance finite elements, especically one-point quadrature stabilized elements, it is assumed here that the EFG domains will be used sparingly, i.e. only where their high accuracy and special capabilities are needed. In that case the EFG domains can be assumed to be of limited size and the domain decomposition based on finite elements may be considered appropriate.

5. APPLICATIONS

Sloshing in two dimensions

We consider the problem of a tank filled with water that has an initial velocity $v_0 = 1\ \mathrm{m\ s^{-1}}$ in the positive *x*-direction at time zero. The configuration is shown in Figure 3.

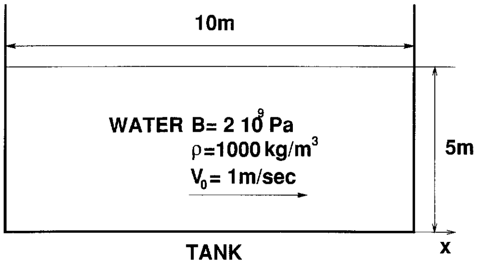


Figure 3. Tank filled with water with initial velocity

The water was modelled as inviscid, so the Lagrangian equation of motion describing the problem is

$$-p_{,i}+\rho b_i=\rho a_i, \tag{29}$$

where $b_x=0$ and $b_y=-g$. The material model used is

$$\dot{p}=\frac{\kappa}{\rho}\dot{\rho}, \tag{30}$$

where

$$\dot{\rho}=-\rho\nabla\cdot\mathbf{v}. \tag{31}$$

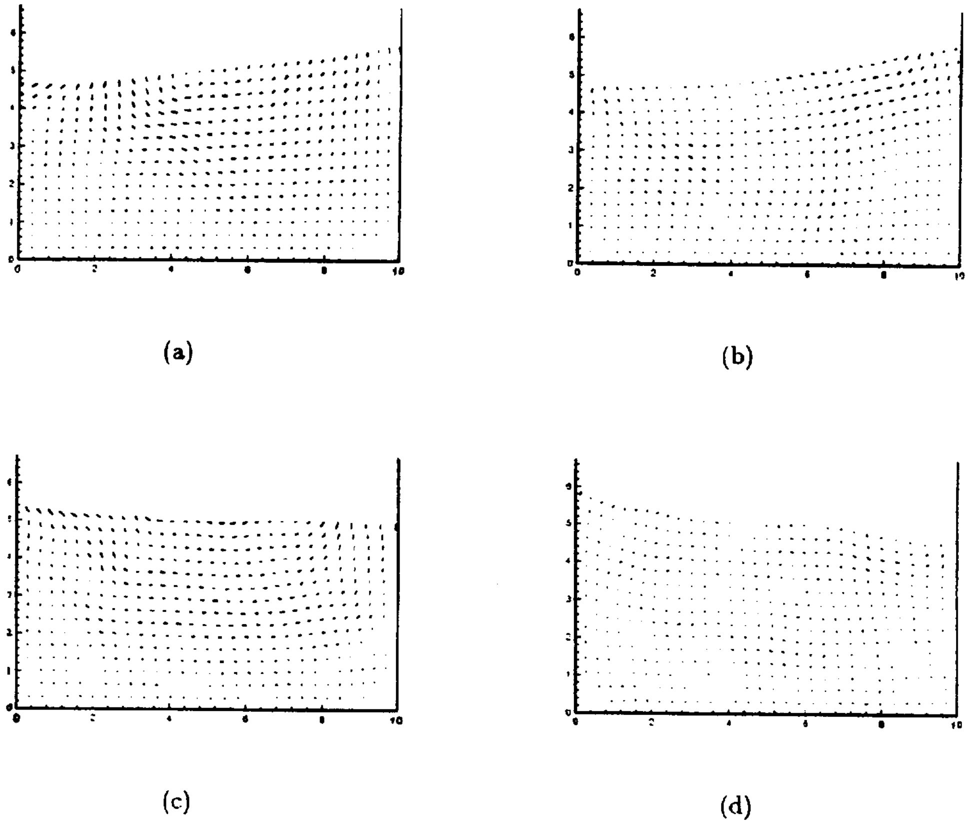
The bulk modulus κ is 2 GPa. The density of the water is taken to be 1000 kg/m^3.

The problem was solved using 496 nodes and 1800 quadrature points. An explicit central difference scheme was used for integration in time. The time step was therefore limited by conditional stability; a time step of 0·2 ms was used. The solution was advanced to 2·8 s by taking 14,000 time steps. Some of the various intermediate states are plotted in Figure 4.

Taylor bar impact

The EFG method was applied to 3D dynamic problems of finite strain elastoplastic deformation. The Taylor bar is a classical benchmark which consists of a cylindrical bar impacting a rigid, frictionless wall.[21,22] The contacting surfaces are assumed to remain in contact throughout the simulation. The material data corresponds to copper and the computational model is classical J_2 flow with linear isotropic hardening: $E=117$ GPa, $\nu=0{\cdot}35$, $\rho=8930$ kg m^{-3}, $h=100$ MPa, $\sigma_{y0}=400$ MPa. The radius of the cylinder is 3·2 mm and its length is 32·4 mm. The cylinder moves with a uniform initial velocity of 227 m s^{-1}. The response is computed for 80 μs; at the end of this interval the kinetic energy is almost entirely dissipated. The monitored data are the radius on the plane of impact, the cylinder length and the equivalent plastic strain.

Normal impact. We have solved two problems. In the first the impact is symmetric with respect to the impact plane. Thus the model consists of two bars impacting each other with the same velocity in opposite directions. The grid for this simulation was quite coarse. Figure 5 shows the cross-section with integration cells and EFG nodes (as X-markers). Twenty layers of the cells and nodes constituted half of the symmetric grid, giving 800 nodes (2400 degrees of freedom) and 480 hexahedral integration cells for the whole model. Spherical supports with a uniform support size of $d_{\mathrm{m}}=4$ mm were used. The material response was computed by the instantaneous final rotation method of Section 4 at the Gauss points. One-point quadrature was used, with the exception of five



(a)

(b)

(c)

(d)

Figure 4. Velocity plots in 2D sloshing problem at $t=$(a) 0·4, (b) 1·2, (c) 2·0 and (d) 2·8 s

layers of cells on each side of the impact plane where 2 × 2 × 2 quadrature was applied to avoid hour-glassing. The time step was 0·1 μs, leading to a total of 800 time steps.

The results are compared here at $t=80$ μs with the 3D solution of Hallquist[21] computed by DYNA3D. The radius at the plane of impact was 6·96 mm (DYNA3D: 7·03 mm) and the length of the bar was 21·63 mm (DYNA3D: 21·47 mm). Plate 1 shows the deformed grid at $t=80$ μs, with contours of the equivalent plastic strain in an axial cut, and Figure 6 documents the evolution of the radius at the impact plane in time.

The second grid for the normal impact was finer; see Figure 7, which shows the cross-section with integration cells and EFG nodes (as X-markers). Thirty-six layers of the cells and nodes constituted the full model, leading to 1728 nodes (5184 degrees of freedom) and 1728 hexahedral integration cells. Two quadrature schemes were used, the first bring the 'full' 2 × 2 × 2 Gaussian quadrature and the second the two-point diagonal quadrature (see Section 4) with 240 integration cells next to the plane of impact being integrated by 2 × 2 × 2 Gaussian quadrature. Spherical supports with a uniform radius of $d_{\mathrm{m}} = 2$ mm were used, leading to an average of 42 (37) EFG nodes involved at an integration point for eightpoint (two-point diagonal) quadrature. The contact condition at the impact plane was enforced using a simple penalty technique. The contact type was that of bilaterial frictionless sliding to emulate the desired no-lift condition. The time step was $3{\cdot}03 \times 10^{-8}$ s, leading to a total of 2640 time steps.

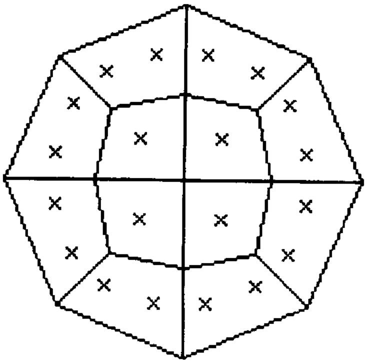


Figure 5. Coarse grid for symmetric variant of normal impact

The results (almost identical for the two quadrature rules) are compared here at $t = 80$ μs with the 3 D solution of Hallquist[21] computed by DYNA3D. The radius at the plane of impact was 7·19 mm (DYNA3D: 7·03 mm) and the length of the bar was 21·36 mm (DYNA3D: 21·47 mm). The maximum of the equivalent plastic deformation was 3·22 (DYNA3D: 2·96). As can be seen, the finer grid seems too flexible. However, when the present results are compared with all the available solutions for this problem, such as those collected by Ponthot,[23] it seems that (i) the Hallquist solutions are rather on the low side (e.g. the mean of the results for the radius from Reference 23 is 7·11 mm) and (ii) the penalty enforcement of the contact condition at the impact plane generated vibrations, which produced higher plastic deformations (all solutions from Reference 23 were for finite element models with directly specified sliding boundary conditions).

Oblique impact. The oblique impact differs from the normal case in that the former is a true 3D problem whereas the latter is usually modelled as axisymmetric. The present simulation was run for an inclined anvil with slope 1:13, depicted in Figure 8. The contact is unilateral and frictionless,

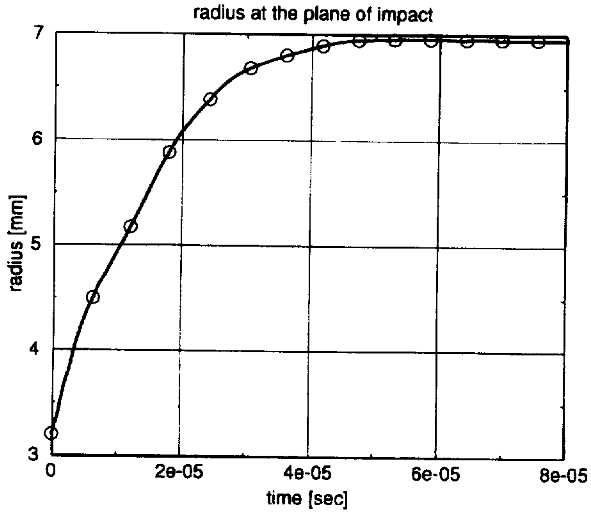


Figure 6. Evolution of radius at plane of impact for coarse grid

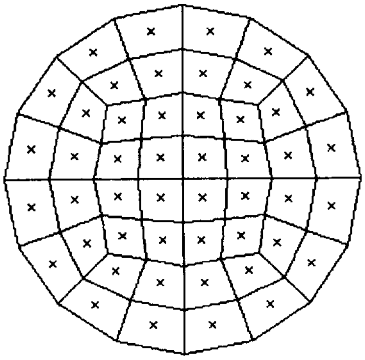


Figure 7. Finer grid

enforced by a penalty technique. The geometric and material data are, with the exception of the anvil, identical with those used above.

The response was again computed for 80 μs, but in this case the kinetic energy is not zero at this time, since part of the kinetic energy remains in the sliding 'down' the slope. The deformed shape of the bar is shown in Plate 2 and the distribution of the equivalent plastic strain is depicted in Plate 3.

Sloshing in a water tank

The final example deals with the sloshing in a rectangular free surface tank. The horizontal dimensions of the tank are 0·2 × 0·2 m^2 and the depth is 0·1 m. The liquid is assumed to be perfect, inviscid and compressible. The constitutive equation is used in the hypoelastic form

$$p(t+\Delta t)=p(t)-\kappa \det(\ln \mathbf{U}_{\mathrm{rel}}(t)), \qquad (32)$$

where p is the pressure, κ is the bulk modulus, which was adopted as constant, $\kappa = 2{\cdot}2$ GPa, and $\mathrm{U}_{\mathrm{rel}}(t)$ is the relative Hencky tensor.[19]

The initial conditions were selected to correspond to a tank moving with a constant uniform velocity $\bar{\mathrm{v}} = \{-0{\cdot}5, -0{\cdot}2, 0\}$ m s^{-1} (it is assumed that the contents of the tank are in static equilibrium corresponding to gravity loading), which is stopped suddenly at $t = 0$. The boundary conditions were enforced at the vertical walls and at the bottom using bilateral penalty tractions. The

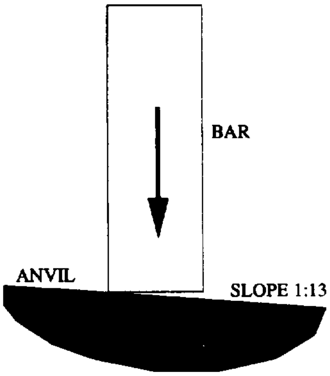


Figure 8. Schematic diagram of oblique impact simulation

grid was regular, with a spacing of 0·02 m in each direction. Spherical supports were used with a uniform radius of 0·05 m. The numerical integrations were performed at 10 × 10 × 10 hexahedral integration cells by 2 × 2 × 2 Gaussian quadrature. The time step was $\Delta t = 0{\cdot}00012$ s; the simulation was run for 0·6 s.

The computation is documented in Plate 4, which shows the displaced contents of the tank at intervals of 0·05 s during the motion.

6. COMPARISON OF COMPUTATION COSTS

Computational cost is one of the major factors affecting the acceptance of a given numerical technique. The cost should be measured for a prescribed accuracy for a given problem (both in terms of the rate of convergence and in terms of the absolute error). When finite elements are compared, this is typically not the case; the costs are usually measured for a given number of degrees of freedom. While this is acceptable if the accuracy per degree of freedom is of the same order, it becomes unsubstantiated if this is not the case. The EFG models are in many problems more accurate than finite element models with the same number of degrees of freedom.

The comparison of EFG and FE techniques based on accuracy measures is currently under investigation. While there are some preliminary results for 3D elastostatics,[7] the dynamic case has not yet been explored. Therefore we find it instructive to compare the techniques in terms of the degrees of freedom, although, as pointed out above, the comparison is not necessarily fair to the EFG method.

We have timed solutions to the normal impact of the Taylor bar. The EFG method was compared with the well-known explicit finite element programme LS-DYNA3D.[24] The FE and EFG models did not contain the same number of finite elements (integration cells), the contact conditions on the impact plane were enforced differently (deleted degrees of freedom in the finite element case; the contact algorithm in the EEG model), time steps differed and different material models were used. These differences were taken into account by appropriate scaling of the measured time as described below. The measurements were performed on an HP/9000 series 715 workstation.

The LS-DYNA3D computation kernel was written in Fortran 77. The Fortran 77 optimizer on the workstation used can perform vectorization; the extent to which vectorization has been performed for LS-DYNA3D is, however, unknown. The programme has been hand-tuned for performance for the last two decades and is reputed for speed. The solid elements used in the computation are one-point integrated, stabilized eight-noded hexahedra. Both the volume integration and the material constitutive law are quite efficient.[24] The finite element model for LS-DYNA3D consisted of 972 elements and 3551 degrees of freedom. The target time of 80 μs was reached by 2513 steps in 135 s.

The EFG programme was written in the C-language in an object-oriented manner. No hand-tuning of performance has yet been undertaken, with the exception of a careful programming style to avoid major inefficiencies. The programme was compiled with the default (conservative) optimization of the compiler. The performance of optimized C programmes on the workstation is typically 10%–30% worse than that of equivalent optimized (but not vectorized) Fortran 77 programmes. This is partly due to a more conservative optimization, which takes into account memory access uncertainties involved in unrestricted pointer arithmetic of the C-language.

The EFG model was the finer grid for normal impact described in Section 5, i.e. 1728 hexahedral integration of cells and 5184 degrees of freedom. Two quadrature schemes were considered as stated in Section 5, namely (i) 2 × 2 × 2 Gaussian quadrature and (ii) the special two-point quadrature rule of Section 4. Because of the differences in the characteristics of the FE and EFG models (number of finite elements versus number of integration cells, number of time steps, different formulations of the constitutive equations), the measured times need to be adjusted. The time measured for the LS-DYNA3D programme is thus compared with the time measured for the EFG model multiplied by these factors: $972/1728 = 0{\cdot}5625$ (to account for the different size of the model), $2513/2640 = 0{\cdot}95$ (to account for the different number of time steps) and 0·75 (the material model used in the EFG programme is more expensive than the model used by LS-DYNA3D, since more kinematic quantities need to be computed; the reduction factor is only a guess, since the material model of LS-DYNA3D has not been implemented yet).

The adjusted time used by the EFG programme for the eight-point integration scheme was 7330 s, i.e. 54 times the LS-DYNA3D timing. The adjusted time used by the EFG programmes for the two-point diagonal integration scheme was 1477 s, i.e. 11 times the LS-DYNA3D timing. In evaluating these numbers, let us note the following.

(a) The EFG method is intrinsically more expensive than the FE method. One reason lies in the fact that typically more nodes are involved in the Galerkin procedure at any integration point. However, this also makes the EFG method more accurate and removes some unpleasant conditions of overconstraining, e.g. volumetric locking. Another reason lies in the fact that connectivity for the EFG method varies from point to point, so that in contrast with the FE method some compiler optimizations are not available (e.g. loop unrolling).

(b) The EFG programme has not yet been streamlined for performance, e.g. by compiler directives allowing aggressive optimization or by rewriting critical sections of code in Fortran 77 or Fortran 90.

(c) The experience we have gained in 3D elastostatics[7] indicates that FE models attain the same level of accuracy with comparable computation times to EFG models, but with four to five times as many degrees of freedom. In an implicit method such as elastoplastics, stiffness assembly is a major cost. This operation is proportional to the square of the number of nodes involved at an integration point. In explicit computations the cost is linearly proportional to the number of nodes involved at an integration point. Thus the explicit EFG method can be expected to be less affected by the larger number of nodal unknowns per quadrature point than the implicit one.

Thus the costs of the EFG method appear comparable with those of finite element models of similar accuracy. On the other hand, once we consider problems with moving discontinuities, the EFG method becomes more cost-effective.

7. CONCLUSIONS

The formulation and implementation of a three-dimensional meshless method, the element-free Galerkin (EFG) method, have been described. The methodology is intended for dynamic problems with geometric and material non-linearities solved with explicit time integration. We have shown how to formulate the EFG approximation in the reference (material) setting, which achieves considerable savings because the EFG shape functions need not be recomputed for each time step.

Although the problems reported here deal with homogeneous bodies, the main field of application of the presented method is seen in problems with evolving discontinuities such as cracks. In such a case, some of the shape functions need to be recomputed to account for the developing discontinuity. Thus a method for speeding up the calculation of shape functions and their derivatives is needed. One such method, based on the consistency conditions, was presented above. The savings in recomputations can be more than 50%.

Solutions by the EFG method presented in this work have been reported for two sets of problems. The first were simulations of large-strain plastic deformation and the second set of results dealt with fluid sloshing.

The classical benchmark of Taylor bar impact was solved for both normal impact and oblique impact. The accuracy obtained was satisfactory when compared with reference solutions. The grid used for the unsymmetric case was chosen to be finer and a frictionless penalty contact technique was used. The accuracy of the solution compared well with reference solutions.

To test the EFG method on a truly three-dimensional problem, the Taylor bar formulation has been modified so that the bar impacts with an angle of obliquity. There are no reference solutions for this formulation.

All the impact simulations indicate that the EFG method is free of the volumetric locking that plagues large-strain formulations of finite elements for (almost) incompressible materials (J_2 plasticity, hyperelasticity, etc.), which agrees with the findings of Reference 2 for incompressible linear problems.

The second set of problems dealt with the sloshing of inviscid compressible fluids in tanks with free surfaces. We have presented a two-dimensional simulation in a large tank and a three-dimensional simulation in a small, hexahedral tank with oblique initial conditions.

The issue of computational cost of the EFG method was discussed in Section 6. It was shown that the EFG method can be programmed in such a manner as to be competitive with fast finite element programmes, especially when compared on the basis of accuracy versus cost.

The EFG method has been shown to be a versatile tool for explicit large-deformation three-dimensional simulations. This was demonstrated on the Taylor bar impact benchmark and both two- and three-dimensional free surface fluid motion (sloshing). The method shows promising characteristics, e.g. the absence of volumetric locking in incompressible materials. The results for problems of moving discontinuities such as cracks will be reported soon.

ACKNOWLEDGEMENTS

We gratefully acknowledge the support of the Office of Naval Research. J.-P. Ponthot is thanked for valuable comments and suggestions regarding the large-strain plasticity material law.

REFERENCES

1. E. Oñate, S. Idelsohn, O. C. Zienkiewicz, R. L. Taylor and C. Sacco, 'A stabilized finite point method for analysis of fluid mechanics problems', *Comput. Methods Appl. Mech Eng.*, **139**, 315–346 (1996).
2. T. Belytschko, Y. Y. Lu and L. Gu, 'Element-free Galerkin methods', *Int. j. numer. methods eng.*, **37**, 229–256 (1994).
3. T. Belytschko, Y. Y. Lu, L. Gu and M. Tabbara, 'Element-free Galerkin methods for static and dynamic fracture', *Int. J. Solids Struct.*, **32**(17/18), 2547–2570 (1995).
4. T. Belytschko, L. Gu and Y. Y. Lu, 'Fracture and crack growth by element-free Galerkin methods', *Model. Simul. Mater. Sci. Eng.*, **2**, 519–534 (1994).
5. T. Belytschko and M. Tabbara, 'Dynamic fracture using element-free Galerkin methods', *Int. j. numer. methods eng.*, **39**, 923–938 (1996).
6. Y. Y. Lu, T. Belytschko and M. Tabbara, 'Element-free Galerkin methods for wave propagation and dynamic fracture', *Comput. Methods Appl. Mech. Eng.*, **126**, 131–153 (1995).
7. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming and P. Krysl, 'Meshless methods: an overview and recent developments', *Comput. Methods Appl. Mech. Eng.*, **139**, 3–47 (1996).
8. C. A. Duarte, 'A review of some meshless methods to solve partial differential equations', *Tech. Rep. 95-06*, Texas Institute for Computational and Applied Mathematics, Austin, TX, 1995.
9. W. K. Liu, S. Jun and Y. F. Zhang, 'Reproducing kernel particle methods', *Int. j. numer. methods fluids*, **20**, 1081–1106 (1995).
10. M. Fleming, Y. A. Chu, B. Moran and T. Belytschko, 'Enriched element-free Galerkin methods for singular fields', *Int. j. numer. methods eng.*, submitted.
11. T. Belytschko, Y. Krongauz, M. Fleming, D. Organ and W. K. Liu, 'Smoothing and accelerated computations in the element-free Galerkin method', *J. Comput. Appl. Math.*, **74**, 111–126 (1996).
12. T. Belytschko, 'A survey of numerical methods and computer programs for dynamic structural analysis', *Nucl. Eng. Design*, **37**, 23–34 (1976).

13. T. Belytschko and M. O. Neal, 'Contact-impact by the pinball algorithm with penalty and Lagrangian methods', *Int. j. numer. methods eng.*, **31** 547–572 (1991).
14. T. Belytschko, D. Organ and Y. Krongauz, 'A coupled finite element–element-free Galerkin method', *Comput. Mech.*, **17**, 186–195 (1996).
15. W. K. Liu, Y.-K. Hu and T. Belytschko, 'Multiple quadrature underintegrated finite elements', *Int. j. numer. methods eng.*, **37**, 3263–3289 (1994).
16. P. Krysl and T. Belytschko, 'Analysis of thin plates by the element-free Galerkin method', *Comput. Mech.*, **17**, 26–35 (1996).
17. T. Belytschko and L. P. Bindeman, 'Assumed strain stabilization of the eight node hexahedral element', *Comput. Methods Appl. Mech. Eng.*, **105**, 225–260 (1993).
18. J. C. Simo and M. S. Rifai, 'A class of mixed assumed strain methods and the method of incompatible modes', *Int. j. numer. methods eng.*, **29**, 1595–1638 (1990).
19. J. C. Nagtegaal and F. E. Veldpaus, 'On the implementation of finite strain plasticity equations in a numerical model', in Pittman (ed.), *Numerical Anlaysis of Forming Processes, Wiley, New York, 1984, pp. 351–371.*
20. P. Krysl and T. Belytschko, 'Object-oriented parallization of explicit structural dynamics with PVM', in preparation.
21. J. O. Hallquist, DYNA2D and DYNA3D User's Manuals, Lawrence Livermore National Laboratories, Livermore, CA, 1980.
22. W. K. Liu, H. Chang, J. S. Chen and T. Belytschko, 'Arbitrary Lagrangian–Eulerian Petrov–Galerkin finite elements for non-linear continua', *Comput. Methods Appl. Mech. Eng.*, **68**, 259–310 (1988).
23. J.-P. Ponthot, 'Mechanics of continuous solids under large strains, and unified treatment by the finite element method', *Tech. Rep. TF-38*, University of Liège, 1995.
24. J. O. Hallquist, 'LS-DYNA3D theoretical manual', *LSTC Rep. 1018, Rev. 3*, Lawrence Livermore Software Technology Corporation, Livermore, CA, 1994.

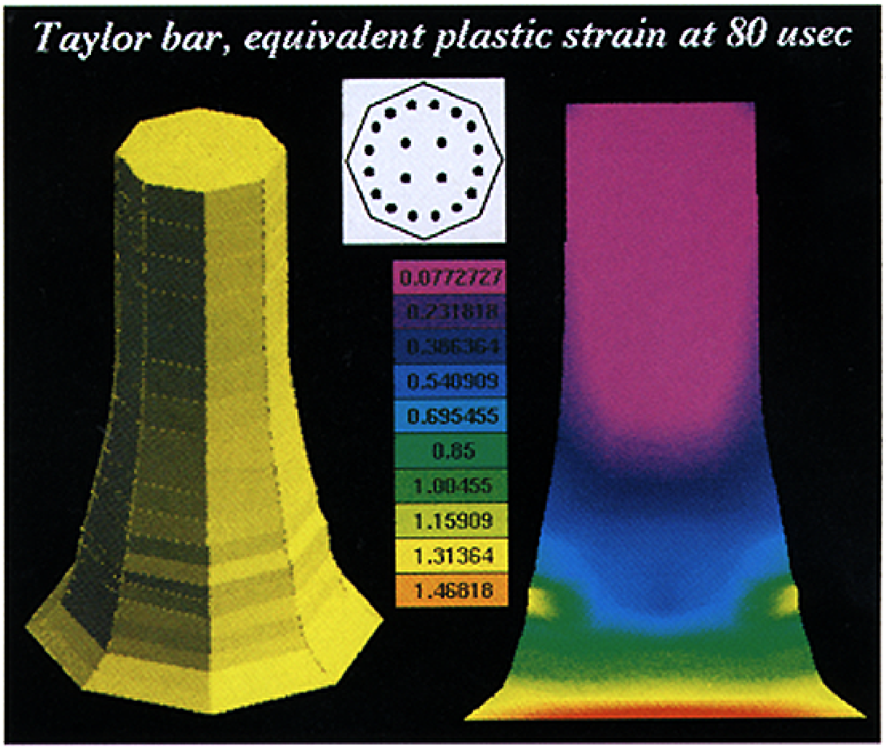


Plate 1. Coarse grid at 80 µseconds. Deformed shape, cut through the body with contours of equivalent plastic strain (only one half of the whole grid is plotted)

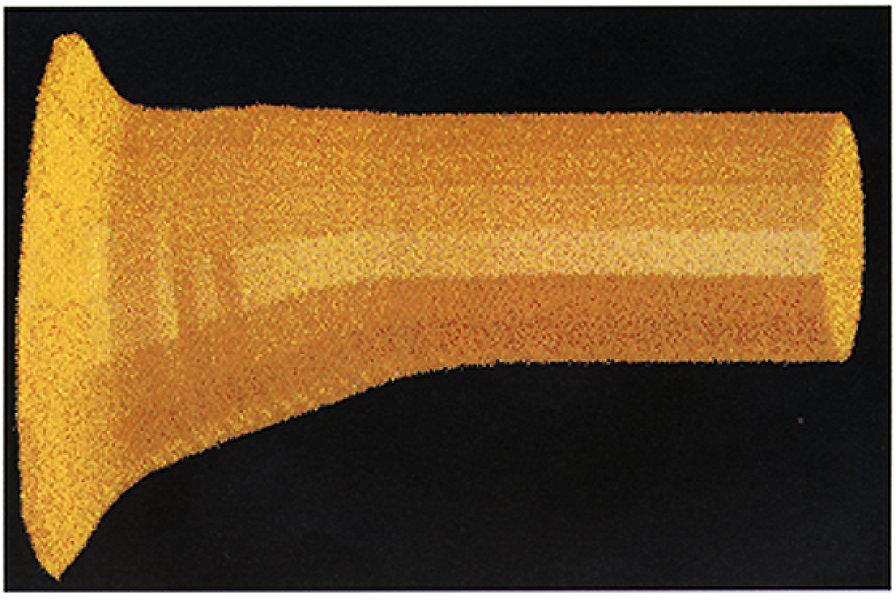


Plate 2. Deformed shape of the bar for the oblique impact at 80 µseconds

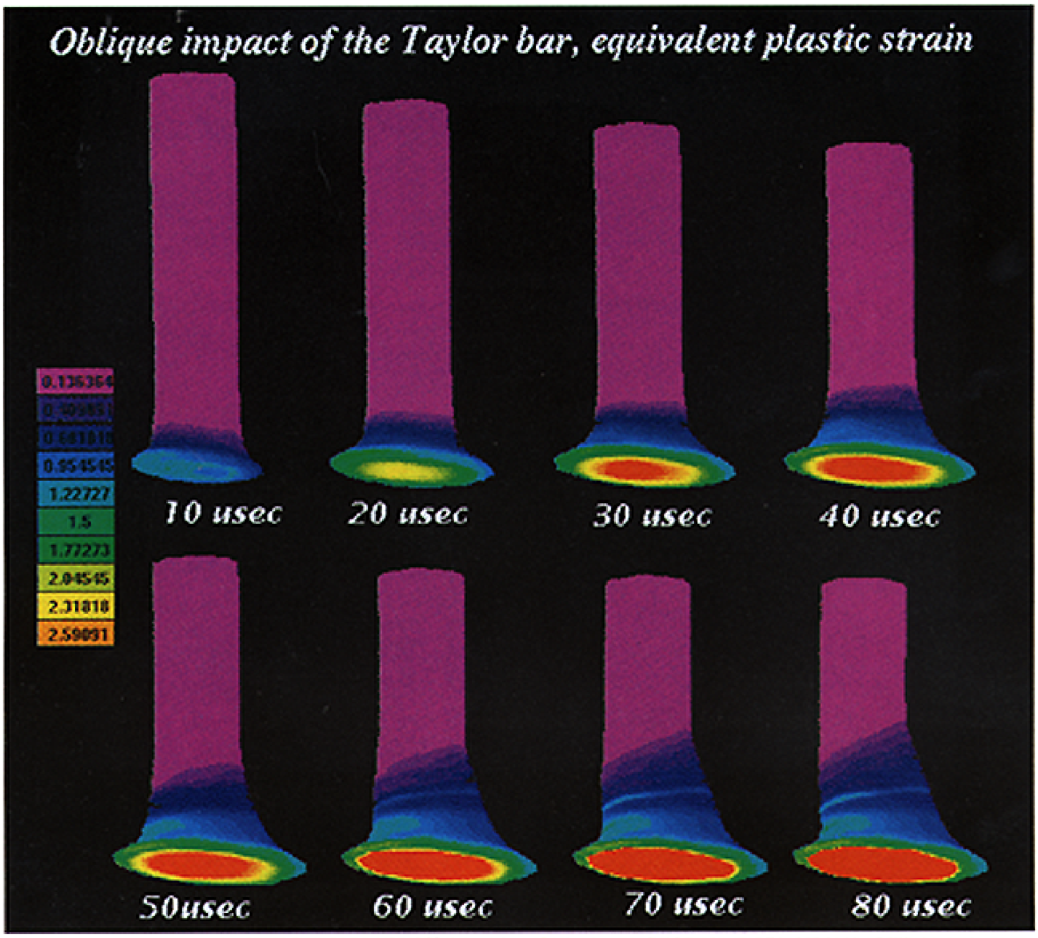


Plate 3. History of the distribution of the equivalent plastic strain for the oblique impact

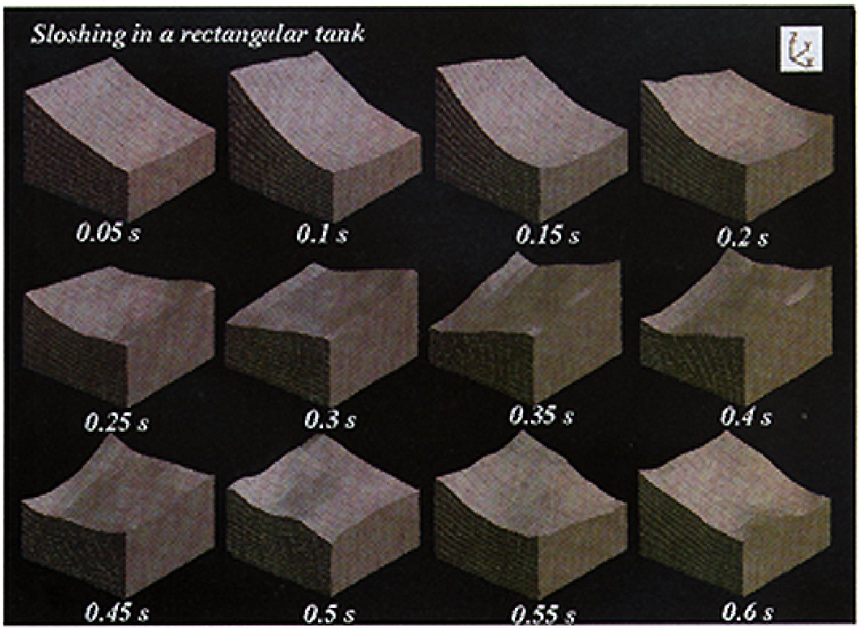


Plate 4. History of the liquid motion for the sloshing problem