ACADEMIC
PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 184 (2003) 476–497

JOURNAL OF
COMPUTATIONAL
PHYSICS

www.elsevier.com/locate/jcp

# A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials

L. Moresi *, F. Dufour, H.-B. Mühlhaus

*CSIRO Exploration and Mining, Australia Resources Research Centre, P.O. Box 1130, Bentley, Western Australia, Australia*

Received 19 October 2001; received in revised form 13 September 2002; accepted 10 October 2002

**Abstract**

We review the methods available for large deformation simulations of geomaterials before presenting a Lagrangian integration point finite element method designed specifically to tackle this problem. In our Ellipsis code, the problem domain is represented by an Eulerian mesh and an embedded set of Lagrangian integration points or particles. Unknown variables are computed at the mesh nodes and the Lagrangian particles carry history variables during the deformation process. This method is ideally suited to model fluid-like behavior of continuum solids which are frequently encountered in geological contexts. We present benchmark examples taken from the geomechanics area.

*Keywords:* Finite element method; Lagrangian integration points; Large deformation viscoelasticity; Multigrid

## 1. Introduction

A long-term goal of geological modeling is to achieve the degree of simulation capability currently enjoyed by the engineering community. The routine ability to recreate the evolution of geological structures during deformation and simultaneously compute stresses, temperatures, fluid flow vectors, and chemical evolution would greatly enhance our ability to understand the Earth. Many questions in geology are formulated in terms of inverse analysis because of the very nature of the science but there is little hope of making significant progress until a reliable forward modeling capability is developed [18].

Unlike most engineering simulations, however, the geometry of the geological model is a result of the continuously evolving non-linear interaction of the structure and the rheology at extremely large strains [13]. Initially the geometry might be relatively simple, e.g., flat layering, but during the course of the simulation very intricate patterns will develop and need to be accurately resolved by the numerical method.

* Corresponding author. Present address: School of Mathematical Sciences, Monash University, P.O. Box 28M, Clayton, Victoria 3800, Australia.

*E-mail addresses:* louis.moresi@sci.monash.edu.au (L. Moresi), Frederic.Dufour@ec-nantes.fr (F. Dufour), hans.muhlhaus@csiro.au (H.-B. Mühlhaus).

PII: S0021-9991(02)00031-1

In many cases the complexity reflected in the geometry results from post-failure deformation of the material.

To make matters worse, the rheological laws which govern particular geological materials at a given scale are often poorly determined—this is partly due to the difficulty in measuring behavior at realistic strain rates, and partly due to the microstructural complexity of the materials which make up a given rock suite. The fact that the materials also undergo metamorphic transformations and phase changes during the evolution of a geological structure is another complicating factor which makes simulation difficult.

As an example, on a global scale, the continents drift as an integral part of the surface thermal boundary layer of the convecting mantle. They have retained a distinct identity within the mantle flow for billions of years while developing a strong physical and chemical fabric along the way. Motions in the mantle are described by the equations of fluid dynamics for very large deformation. The rheology needed to describe deformation in the lithosphere is highly non-linear, and near the surface where temperatures are less than approximately 600 °C it becomes necessary to consider the role of elasticity [17]. The strong correlation between seismicity and plate boundaries (e.g., [1]) makes it seem likely that plate motions are associated with localization of deformation occuring when stresses reach the yield strength of the lithosphere.

From a modeling point of view, it is necessary to consider the fluid convection of the mantle and the history-dependent viscoelastic/brittle behavior of the continental crust as a single coupled system. At the same time, the precise structure and composition of the deep continental crust is not well known. The requirements for a geological simulation code are therefore an ability to track boundaries and interfaces through extremely large deformation, including fluid convection, of non-linear history-dependent materials. The wide range of physical and temporal scales, and the many coupled physical processes also impose a need for computational efficiency. The code should also be very flexible in the rheological laws which it can treat.

Many different numerical methods have been devised for mechanical simulations of this kind. Some derive from standard engineering methods, while others were developed to handle specific problems in the physical sciences. We summarize a number of these methods in order to illustrate the difficulties involved in creating convincing, realistic simulations. We next present the concepts behind Ellipsis, a Lagrangian Integration Point code. The method derives from the dynamic Material Point Method (e.g., [16]) modified for simulating creeping flow of viscoelastic-brittle materials in the limit of very high strains, including thermal convection.

We discuss the implementation in the context of standard finite element methods, concentrating on the important modifications which are needed to a standard Eulerian flow code. For example, it is necessary to develop an integration scheme based on an arbitrary set of given positions of integration points in an element and to split Lagrangian particles close to stagnation points in the flow.

We then move on to the specific formulation for a creeping Maxwell viscoelastic material with a yield stress which is a suitable representation for our target application: global simulations of planetary deformation.

The method is demonstrated on a number of very simple benchmark cases where the accuracy can otherwise be studied in large deformation.

### *1.1. Modeling techniques*

The key to all these problems is to find a way to deal efficiently with the most general case: finite strain viscoelastic/plastic deformation of fluids with material parameters which can depend arbitrarily on history. This is a specialized problem lying between classical structural engineering and fluid dynamics, and, in general, requires specialized numerical methods. A great deal of research has been done in developing numerical methods suited for large strain problems in different application areas.

The principal difficulty in finite strain modeling is the tendency for any computational or logical connecting mesh between material points to become arbitrarily tangled during the simulation.

In fluid dynamics where very high strains are always expected, the approach is often to cast the equations of motion in Eulerian form, in which there is no need to update computational points to account for material deformation. Additional advection terms are needed to handle the transport of material history information. The fact that the computational points have no special relationship to the flow usually means that computation of the advection terms involves an operation equivalent to interpolation, and hence introduces some spurious diffusion. Specialized schemes to moderate this effect exist, but require significant tuning (e.g., [8,14]).

With Lagrangian formulations, the deformation of the computational mesh is used to compute strain gradients. Severe distortion or tangling renders the mesh useless for this purpose. It is important to remesh, or reorganize the connectivity of the mesh to maintain a usable grid (see for example [4], (NEM, [5])). However, this regridding/reconnection represents some kind of smoothing operation on the recorded deformation history.

One further difficulty with Lagrangian finite element models is the fact that shape functions are discontinuous at the nodal points which means that strains/strain rates cannot be obtained directly from the nodal point displacement/velocity data at the material points, only in the element interiors.

Meshless methods exist in which overlapping smoothing kernels local to each computational point are used to create a differentiable interpolation of the nodal point unknowns (e.g., Smoothed Particles Hydrodynamics (SPH, [10]), Element Free Galerkin Methods (EFG, [2])). These methods have no difficulty in dealing with arbitrary relative motions of the computational points, but the need for overlapping kernels makes it difficult to apply boundary conditions at computational points.

An alternative to pure meshless methods is the Material Point Method (MPM) developed by Sulsky, Schreyer, and their co-workers (e.g., [16]) who use an Eulerian mesh of computational points with Lagrangian material points which record the history information. In this case the mesh supplies boundary information, a framework for building the discretization, and incremental solutions for updating information at the material points and for updating the locations of material points. In this method there is also an unavoidable smoothing which comes from the coarseness of the computational mesh relative to the material point density. The MPM can be regarded either as a particle method in which particle–particle interactions are mediated by a grid, or as a finite element method with moving integration points. We rely upon this latter view in developing a version of MPM which is suited to convection problems in geodynamics.

## 2. Finite elements with moving integration points

In this section we describe the similarities and principle differences between a standard finite element method and one with moving integration points. Principally these differences are in the updating of integration points themselves, and in the formulation of a weak form based on material property matrices which are tied to integration points. In later sections we discuss efficient grid-based solution methods which can be used with our formulation, and some implementation details.

### *2.1. Updating integration points*

In a particle-in-cell method the grid generally remains fixed whereas particles move through the mesh during computations. Unlike most fixed grid methods, however, the existence of a Lagrangian reference frame allows the equations to be formulated in a Lagrangian sense (equating the material derivative with the time derivative provided all rates of change are computed particle-by-particle and updated appropriately during the advection phase. This eliminates the difficulty with numerical diffusion of advected quantities. Particle velocities are interpolated from nodal velocities and then particle position is updated using a suitable integration scheme such as:

$$\boldsymbol{x}_p^{t+\Delta t} = \boldsymbol{x}_p^t + \Delta t \sum_{\text{node}} \boldsymbol{v}_n N_n(x_p), \tag{1}$$

where $\boldsymbol{v}$ is the nodal velocity and $N$ are the shape functions associated with the nodes of the element in which the particle currently resides. In practice, a higher order scheme such as the fourth order Runge–Kutta scheme gives a more accurate result. Particle updates can be done in a predictor–corrector fashion, initially updating particle locations to obtain velocity solutions and then repeatedly correcting the final locations to obtain a converged velocity. We have found that the improvements in accuracy from iterating on the particle locations are generally too small to justify the increase in complexity, and, for non-linear rheological laws, the iterative scheme may become unstable. Our preference is therefore to use a second order Runge–Kutta scheme (mid-point method) and reduce the timestep whenever higher accuracy in the time integration is required.

### 2.2. Variational form

The governing equations in domain $\Omega$ bounded by $\Gamma$ are standard for creeping fluids, a conservation equation for momentum:

$$\boldsymbol{\tau}_{ij,j} - p_{,i} + \boldsymbol{f}_i = 0, \tag{2}$$

subject to a continuity requirement

$$\boldsymbol{v}_{i,i} = 0. \tag{3}$$

In the above, $\boldsymbol{v}$ is the velocity, $\boldsymbol{\tau}$ is the deviatoric Cauchy stress tensor, $p$ its volumetric component (i.e., pressure), and $\boldsymbol{f}$ is the specific body force. The notation $p_{,i}$ denotes differentiation of $p$ with respect to $x_i$. The boundary conditions are given as follows:

$$\boldsymbol{\tau} \cdot n - p = \bar{t} \quad \text{on the natural boundary } \Gamma_t, \tag{4}$$

$$u = \bar{u} \quad \text{on the essential boundary } \Gamma_u, \tag{5}$$

in which the superposed bar denotes prescribed boundary values and $n$ is the unit outward normal to domain $\Omega$.

Inertial terms are not considered in (2); since the viscosity associated with creeping flow in rocks is enormous ($10^{19}$–$10^{24}$ Pa s) we can assume the Prandtl number is infinite. This means that the strain rate instantaneously equilibrates with the applied boundary conditions and body forces—a factor which strongly influences the way we approach the solution of the equations.

Body forces are assumed to arise through density differences in the material. This may be due to compositional differences between adjacent parcels of material, or temperature variations within the fluid

$$\boldsymbol{f} = (0, -g\rho), \tag{6}$$

where $g$ is the acceleration due to gravity (downwards) and $\rho$ is the material density. We assume the Boussinesq approximation in only considering the density variations when they contribute to the body force term.

The weak form is obtained in the usual way multiplying (2) with the test functions $N_i$ and multiplying (3) with $Q$. Integration over the volume and application of Gauss' theorem yield:

$$\int_\Omega \tau_{ij} N_{i,j}\, \mathrm{d}\Omega - \int_\Omega p N_{i,i}\, \mathrm{d}\Omega = \int_\Omega f_i N_i\, \mathrm{d}\Omega + \int_{\Gamma_t} \bar{t}_i N_i\, \mathrm{d}\Gamma \tag{7}$$

and

$$\int_\Omega v Q_{,i}\, \mathrm{d}\Omega = 0. \tag{8}$$

Eq. (7) must hold for all admissible $N_i$ vanishing on boundary $\Gamma_u$ where displacements are prescribed. The notation $N_{i,j}$ indicates the symmetrical form of the derivative: $\partial N_i/\partial x_j + \partial N_j/\partial x_i$.

## 3. Solution strategy

The particle in cell formulation produces a set of matrix equations in nodal unknowns which have precisely the same form as the standard FEM formulation for the same mesh (though the coefficients in the matrices will be different in most cases). Our solution scheme is very similar to that detailed in [11].

### 3.1. Mixed formulation

We use a mixed formulation in which nodal unknowns are velocity and pressure, and the incompressibility constraint is considered as an independent equation. This gives

$$\begin{pmatrix} \boldsymbol{K} & \boldsymbol{G} \\ \boldsymbol{G}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \\ \boldsymbol{0} \end{pmatrix}, \tag{9}$$

where $\boldsymbol{K}$ is the global stiffness matrix and $\boldsymbol{G}$ is the discrete gradient operator. $\boldsymbol{K}$ is built up from particle derived material properties as well as shape function derivatives, whereas $\boldsymbol{G}$ is purely geometrical and is associated with the mesh. The requirement of symmetry in this equation forces a specific definition of $Q$ in Eq. (8).

With a mixed method, an arbitrary choice of discretization and interpolating functions for pressure and velocity can produce unreliable results. The moving integration point formulation is no different in this respect, and we have found it necessary to combine bilinear velocity shape functions with constant, discontinuous pressures, for example. In other words, despite generally being a somewhat less precise approximation than Gaussian quadrature, the integration scheme based on particle positions does not under-integrate the pressure field in a way which avoids locking of the elements.

Whereas in many fluid applications or elasticity formulations it is common to eliminate the pressure equations through a penalty method, we have found that this leads to very poor performance of iterative solvers and cannot be used in this case. Instead we follow the Uzawa scheme in which we apparently eliminate the velocity by writing

$$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{K}\boldsymbol{v} + \boldsymbol{G}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{G}\boldsymbol{p} = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{F} \tag{10}$$

and using the fact that $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{v} = 0$ to obtain an expression entirely in the pressure:

$$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{G}\boldsymbol{p} = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{F}. \tag{11}$$

The procedure outlined in general by Cahouet and Chabard [6] and for this specific system by Moresi and Solomatov [11] solves this pressure equation through a conjugate gradient series of solutions for velocities which can be accumulated to give the full solution.

### 3.2. Multigrid solver

The static, regular grid makes it possible to use (unmodified) a multigrid solver developed for purely Eulerian flow problems [11].

The basis of the method is to build an approximate solution to the problem on a coarse grid by computing the stiffness matrix, boundary conditions, and force terms at this resolution. The approximate solution is then interpolated to a finer mesh and used as a starting guess in the iteration. By repeated transfer between coarse and fine meshes, and using an iterative scheme which reduces errors on a length scale defined by the mesh (such as Gauss–Seidel relaxation), reduction of errors on all length-scales occurs at the same rate.

One difficulty in multigrid methods is to devise a consistent representation of the problem at different resolutions which maximizes the convergence rate, particularly when material properties are spatially variable. The construction of the stiffness matrices for each grid using a resolution-independent representation of the problem overcomes this difficulty. However, for maximum computational efficiency it would almost certainly be necessary to use the relationship between components of the stiffness matrices at different multigrid resolutions to compute the grid transfer operations (some form of operator-dependent transfer method such as [7]).

If carefully tuned, multigrid methods are capable of solving problems with $n$ unknowns in a time proportional to $n$, in contrast to $n \log n$ for conjugate gradient method and $n^2$ or even $n^3$ for direct solver.

For non-linear problems, the solution rate is also greatly enhanced by using the coarse levels for numerous fast iterations to approach the approximate solution before interpolating to finer meshes. For example, see [12]. In Ellipsis we use a full multigrid scheme with a direct solver at the coarsest level.

## 4. Numerical implementation

### 4.1. Numerical integration

At the heart of the finite element method is the concept that the weak form of the equations can be split up into a number of sub-integrals over individual elements, and that these integrals can be computed approximately using a suitable quadrature scheme.

$$\int_{\Omega^e} \phi \, d\Omega^e \approx \sum_{p=1}^{n_{ep}} w_p \phi(\boldsymbol{\xi}_p), \tag{12}$$

where $\Omega^e$ is an element volume, $\phi$ is a field quantity which is evaluated at a set of sample points with coordinates $\boldsymbol{\xi}_p = (\xi_p, \zeta_p, \ldots)$; $w_p$ are weights for each integration point.

In standard FEM, the locations of the quadrature points, and their weights, are chosen to optimize the integration accuracy for a given set of interpolation functions. The criterion for choosing the quadrature scheme is usually computational efficiency: the minimum number of locations required to achieve exact integration of a specific degree polynomial.

In our formulation, we assume the sample points coincide with material points embedded in the fluid which therefore move with respect to the mesh. The locations of the quadrature points are, as a consequence, given for each element, and it is necessary to vary the weights in order to obtain the correct integral for a given element. The procedure is similar to the determination of weights for any quadrature rule: deriving a set of constraints based on the requirement that polynomials of a certain order have to be integrated exactly, and then equating coefficients to obtain the set of $w_p$ values.

We first consider the one-dimensional case in which $\phi$ in (12) is a polynomial, and the integral is over the range −1 to 1 (a typical "master element")

$$\phi(\xi) = \alpha_0 + \alpha_1\xi + \alpha_2\xi^2 + \cdots + \alpha_n\xi^n. \tag{13}$$

We integrate (13) algebraically over the domain, and equate coefficients with the quadrature expansion of the integral from (12) to obtain $n+1$ constraints on the set of $w_p$ values:

$$\begin{aligned}
&\sum_{p=1}^{n_{\mathrm{ep}}} w_p = 2 \quad \text{(constant terms)},\\
&\sum_{p=1}^{n_{\mathrm{ep}}} w_p\xi_p = 0 \quad \text{(linear terms)},\\
&\sum_{p=1}^{n_{\mathrm{ep}}} w_p\xi_p^2 = \frac{2}{3} \quad \text{(quadratic terms)},\\
&\sum_{p=1}^{n_{\mathrm{ep}}} w_p\xi_p^3 = 0 \quad \text{(cubic terms)},
\end{aligned} \tag{14}$$

and so on.

In principle, we can imagine choosing a suitable number of constraints, and performing an inversion for the $w_p$ values, but in practice this can be very time-consuming, and is not guaranteed to converge for unfavorable material point distributions (e.g., points clumped towards one corner) and may produce negative $w_p$ values. If we wish to associate the value of $w_p$ attributed to a particle with the representative mass or volume of fluid which it occupies, then $w_p$ must be positive.

The Material Point Method uses constant weights for the particles to ensure that mass is conserved exactly in the system. Although constant particle weights are not guaranteed to satisfy even the lowest order constraint (volume conservation) once the particle's positions evolve to a general configuration (since the numbers of particles per element may vary), the accuracy of the integration scheme is relatively good for simulations where material strain is not extreme (see benchmark section). We therefore use the MPM integration scheme as a starting point for computing the weights for a given element. The reference values are adjusted to best fit the constraints up to a certain degree, and subject to the further constraint that no tracer should have a negative weight.

For viscous flow using bilinear elements we can obtain optimal convergence rates, in the limit of an infinitely fine mesh, if only the constant and the linear constraint terms are used (e.g., [9]).

We define a misfit to the constant constraint as follows:

$$M_0 = \sum_p w_p \tag{15}$$

and misfits to the bilinear constraints as follows:

$$M_\xi = \sum_p w_p\xi_p; \quad M_\zeta = \sum_p w_p\zeta_p; \quad M_{\xi\zeta} = \sum_p w_p\xi_p\zeta_p \tag{16}$$

and define

$$L_\xi = \sum_p \xi_p^2; \quad L_\zeta = \sum_p \zeta_p^2; \quad L_{\xi\zeta} = \sum_p \xi_p^2\zeta_p^2. \tag{17}$$

An improvement to the particle weight, $w_p$ is then given by the following iterative procedure

$$w_p^{i+1} = \frac{1}{M_0^{i+1}}\left[w_p^i - \frac{M_\xi^i}{L_\xi^i} - \frac{M_\zeta^i}{L_\zeta^i} - \frac{M_{\xi\zeta}^i}{L_{\xi\zeta}^i}\right], \tag{18}$$

where $i$ is the iteration counter.

To integrate a given order polynomial using particle locations, we generally expect that we will need two to three times the number of particles per element (in 2D) than there would usually be Gaussian quadrature points. This is because the locations of the integration points cannot be optimized as they are in Gaussian or other integration schemes, so particle weights are the only free parameters.

### *4.2. Element matrices and particle properties*

In a standard finite element formulation, the stiffness matrix for an element, $\boldsymbol{k}^{\mathrm{e}}$ is built up in a segregated form, often written as:

$$k^{\mathrm{e}}_{ab} = \int_{\Omega^{\mathrm{e}}} \boldsymbol{B}_a^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B}_b \, \mathrm{d}\Omega^{\mathrm{e}}, \tag{19}$$

where $\boldsymbol{B}_a$ is a matrix comprising shape function derivatives (associated with node $a$) obtained from the constitutive relationship, and $\boldsymbol{D}$ is a matrix of material properties. For our formulation, the material property matrix is considered to be an attribute of an individual particle together with its current state and history and not a property of the mesh. Therefore, in the context of the numerical integration scheme discussed above,

$$k^{\mathrm{e}}_{ab} = \sum_{p=1}^{n_{\mathrm{ep}}} w_p \boldsymbol{B}_a^{\mathrm{T}}(\boldsymbol{x}_p) \boldsymbol{D}_p \boldsymbol{B}_b(\boldsymbol{x}_p), \tag{20}$$

where $n_{\mathrm{ep}}$ is the number of particles which happen to be in an element.

### *4.3. Particle splitting*

Fluid flow close to a stagnation point produces an elongation in one direction and a corresponding shortening in the perpendicular direction. This has the effect of distorting the original volume local to each tracer into a narrow filament (Figs. 1(a) and (b)). This may mean that the fluid initially associated with a particle lies in several different elements, while the integration scheme simply lumps the entire amount into

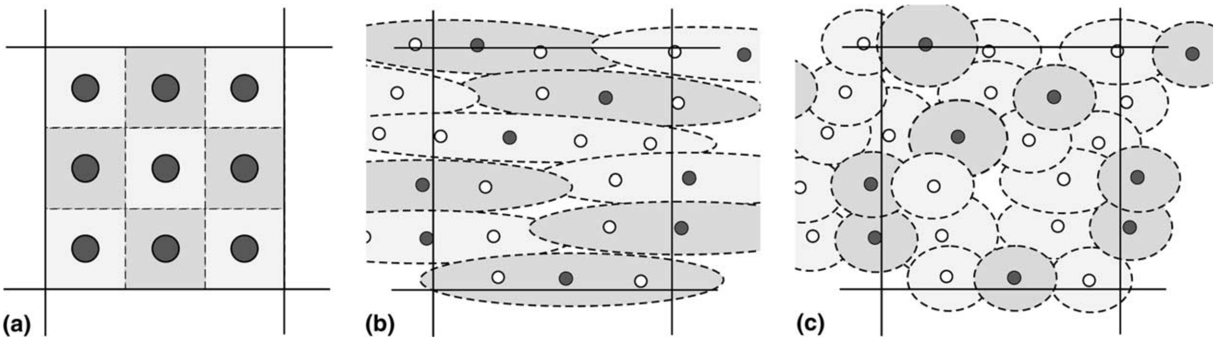

Fig. 1. Integration schemes become more complicated when large material strains produce elongated "local volumes" for particles. (a) Initial configuration, (b) flow near a stagnation point elongates domains: solid circles are original particle points, open circles indicate new locations within the original domain which have similar spacing to the original interparticle spacing, (c) remapping the domains to the solid and open circles again allows the particles to contribute to the integrals of the correct element. The fact that this procedure produces overlapping domains is an indication of the approximate nature of the integration scheme.

the element containing the tracer itself. In the worst case this may leave some elements entirely empty of integration points and others considerably over represented.

The remedy is to ensure that the volume of fluid associated with a particle never becomes too distorted. We keep track of a local measure of strain associated with each particle and use this to generate new particles nearby which occupy the extremities of the distorted local volume. This is illustrated in Fig. 1(b) where the heavily shaded particles sitting at the centroid of the salami-shaped local regions represent the original occupants of the volumes from Fig. 1(a). The material within the element is poorly represented by the particles which actually contribute to the element integrals. The lightly shaded particles are later additions which aim to correct this problem. Local volumes corresponding to the new particle distributions are indicated in Fig. 1(c).

The details of the particular splitting procedure used in these simulations are illustrated in Fig. 2. Lightweight ghost particles track the bounds of the volume associated with each of the material points and allow us to determine the local deformation at any later time. The location of the ghost particles is also used to determine where to place the copies of the original material particle. When splitting particles, we attribute the same history variables to the copies as were held on the original particle. To ensure that this approximation is a reasonable one, the particle splitting occurs when the distortion is relatively small. Accurate tracking of the deformation also ensures that the splitting procedure respects boundary locations and internal interfaces.

Generating new particles requires a complementary step to merge particles when their density becomes unnecessarily high. This is done through a proximity testing step for all particles within an element. When two particles *of the same material type* approach within a factor $A/2$ of the initial separation distance, they are merged to a new particle located at the average of their positions, and recording the average of each of their histories.

An example of how particle splitting and merging influences the number of particles in a simulation is given in the first benchmark example (Fig. 5).

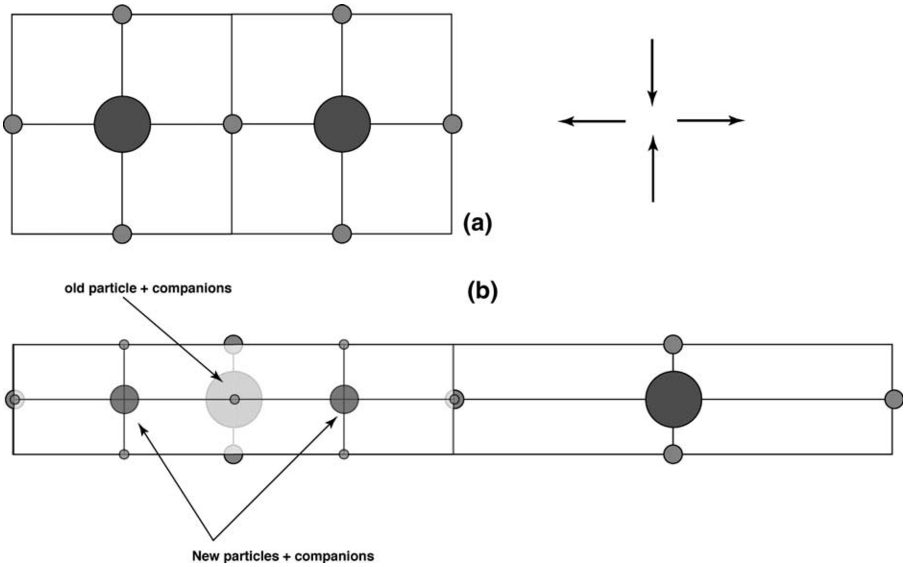

Fig. 2. Tracking the deformation measure around a particle can be done using lightweight ghost particles which measure the extent of the local volume around a material particle in the undeformed state (a). After stretching of the local volume, the ghost particles are used to determine the splitting direction and the location of the two new particles (b, left particle has been split).

### 4.4. Element inverse mapping

It is usual to change variables in the element integrals such as (19) to a regular master element. This greatly simplifies the computation, but in our case there is an additional step required before we can map to a master element. The particle positions are known in the global coordinate system and must first be mapped into the element local coordinate system. Zhao et al. [19] found an algebraic mapping for bilinear quadrilateral elements, but a more general approach is required for higher order elements and for 3D.

The notation is given in Fig. 3 for the 2D case—the extension to three dimensions is straightforward. $\boldsymbol{e}_\xi$ and $\boldsymbol{e}_\zeta$ are unit vectors in the 'natural directions' of the distorted element which map to the $\xi$ and $\zeta$ axes, respectively, in the master element. $h_\xi$ and $h_\zeta$ are characteristic dimensions of the element in the appropriate directions. We wish to map the coordinates, $\boldsymbol{x}_p$ of the particle $p$ to the coordinates in the master element, $\boldsymbol{\xi}_p$. The procedure takes the form of a predictor–corrector iteration: we first guess an initial value of $\boldsymbol{\xi}_p$ and use this to predict the global coordinates $\boldsymbol{x}_p^0$

$$\begin{aligned}
&\boldsymbol{\xi}_p = (0,0), \\
&\boldsymbol{x}_p^0 = \left( \sum_{n=1}^{n_{\mathrm{en}}} N_n(\boldsymbol{\xi}_p) x_n, \sum_{n=1}^{n_{\mathrm{en}}} N_n(\boldsymbol{\xi}_p) y_n \right),
\end{aligned} \tag{21}$$

where $n_{\mathrm{en}}$ is the number of nodes in the element, and $\boldsymbol{x}_n$ are their coordinates. We compute $\xi_p$ through a number of corrector steps:

$$\begin{aligned}
&\xi_p \leftarrow \xi_p + \beta\left(e_{\xi x} x_p^i + e_{\xi y} x_p^i\right)/h_\xi, \\
&\eta_p \leftarrow \xi_p + \beta\left(e_{\zeta x} x_p^i + e_{\zeta y} x_p^i\right)/h_\zeta, \\
&\boldsymbol{x}_p^{i+1} = \left( \sum_{n=1}^{n_{\mathrm{en}}} N_n(\boldsymbol{\xi}_p) x_n, \sum_{n=1}^{n_{\mathrm{en}}} N_n(\boldsymbol{\xi}_p) y_n \right),
\end{aligned} \tag{22}$$

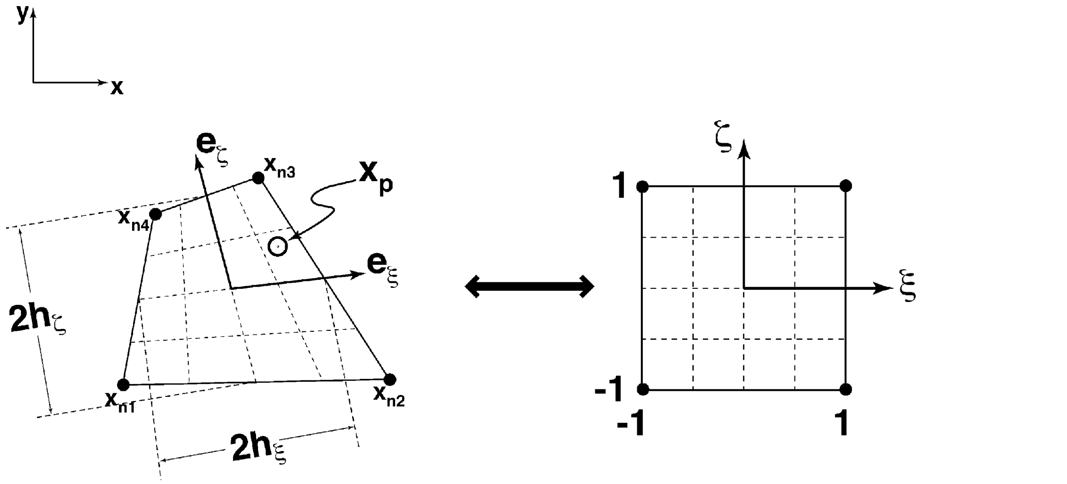

Fig. 3. Coordinate systems in global mesh, distorted element, and master element reference frames.

$i$ is the iteration index, $\beta$ is a relaxation parameter which we chose to be 1.0 for the first iteration and 0.9 for subsequent iterations. When $\boldsymbol{e}_\xi$ and $\boldsymbol{e}_\eta$ are orthogonal, the iteration completes in one step, otherwise it is necessary to repeat the correction step until the predicted $\boldsymbol{x}_p^i$ is within a satisfactory tolerance of the known value.

We use $\xi$ within the element search algorithm since the master element coordinates give an immediate determination of whether the particle lies inside or outside the element even for highly distorted elements (if the element is too distorted for this procedure to work, then it is also too distorted to use to create the element matrices).

### *4.5. Moving boundary conditions*

One major difficulty with the Eulerian mesh is in the application of velocity boundary conditions normal to the surface as the mesh does not automatically track the boundary position. If the sense of the boundary velocity is into the domain, then the method requires the generation of new particles to represent the incoming material. If the sense is out of the domain then particles, and their stored history, will be lost.

To avoid such difficulties, we recommend updating the mesh in between solution steps to follow any velocity boundary conditions normal to boundary surfaces. As pointed out by Sulsky et al. [15], the formulation is unaffected by changes to the mesh occurring after the particle positions have been updated since the mesh carries no information other than domain geometry and boundary conditions: material configuration and history information are entirely determined by the particles.

In general the mesh can be regenerated completely to have the optimal number of nodes and configuration for the updated geometry. In the compression and extension examples given here, however, the mesh topology remains unchanged—the mesh is simply scaled in the appropriate direction to minimize the recomputation of the connectivity matrices. It is important to use the same updating procedure for nodal points as particles even if the expression for the boundary location can be integrated exactly. If this is not done, then the integration error in the location of particles near to the boundary accumulated over many thousands of timesteps could result in a particle being overtaken by an edge and lost.

## 5. Constitutive relationships

### *5.1. Viscoelastic formulation*

We use a Maxwell viscoelastic constitutive relationship which assumes that the deformation rate is the sum of viscous and elastic parts

$$\frac{\overset{\triangledown}{\boldsymbol{\tau}}}{2\mu} + \frac{\boldsymbol{\tau}}{2\eta} = \hat{\boldsymbol{D}}_v + \widehat{\boldsymbol{D}_e} = \hat{\boldsymbol{D}}, \tag{23}$$

where $\overset{\triangledown}{\boldsymbol{\tau}}$ is the Jaumann corotational stress rate for an element of the continuum, $\mu$ is the shear modulus, and $\eta$ is shear viscosity. $\hat{\boldsymbol{D}}$ is the deviatoric part of $\boldsymbol{D}$.

$$\overset{\triangledown}{\boldsymbol{\tau}} = \dot{\boldsymbol{\tau}} + \boldsymbol{\tau}\mathbf{W} - \mathbf{W}\boldsymbol{\tau}, \tag{24}$$

where $\mathbf{W}$ is the material spin tensor,

$$W_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} - \frac{\partial V_j}{\partial x_i}\right). \tag{25}$$

The $\mathbf{W}$ terms account for material spin during advection which reorients the elastic stored-stress tensor.

As we are primarily interested in solutions where very large deformations may occur—such as buoyancy driven fluid convection, we prefer to work with a fluid-like system of equations from the outset.

Hence we obtain a stress/strain-rate relation from (23) by expressing the Jaumann stress-rate in a difference form:

$$\overset{\triangledown}{\boldsymbol{\tau}}{}^{t+\Delta t^{\mathrm{e}}} \approx \frac{\boldsymbol{\tau}^{t+\Delta t^{\mathrm{e}}} - \boldsymbol{\tau}^{t}}{\Delta t^{\mathrm{e}}} - \mathbf{W}^{t}\boldsymbol{\tau}^{t} + \boldsymbol{\tau}^{t}\mathbf{W}^{t}, \tag{26}$$

where the superscripts $t, t + \Delta t^{\mathrm{e}}$ indicate values at the current and future timestep, respectively. $\Delta t^{\mathrm{e}}$ is a timestep which captures the relevant timescales of the changes in elastic stresses. This timestep could, in fact, differ from that chosen for updating the particle positions.

Eq. (23) becomes

$$\boldsymbol{\tau}^{t+\Delta t^{\mathrm{e}}} = \frac{2\eta\Delta t^{\mathrm{e}}}{\alpha + \Delta t^{\mathrm{e}}}\hat{\boldsymbol{D}}^{t+\Delta t^{\mathrm{e}}} + \frac{\alpha}{\alpha + \Delta t^{\mathrm{e}}}\boldsymbol{\tau}^{t} + \frac{\alpha\Delta t^{\mathrm{e}}}{\Delta t^{\mathrm{e}} + \alpha}(\mathbf{W}^{t}\boldsymbol{\tau}^{t} - \boldsymbol{\tau}^{t}\mathbf{W}^{t}), \tag{27}$$

where $\alpha = \eta/\mu$ is the shear relaxation time.

We can simplify the above equations by defining an effective viscosity $\eta_{\mathrm{eff}}$:

$$\eta_{\mathrm{eff}} = \eta\frac{\Delta t^{\mathrm{e}}}{\Delta t^{\mathrm{e}} + \alpha}. \tag{28}$$

Then the deviatoric stress is given by

$$\boldsymbol{\tau}^{t+\Delta t^{\mathrm{e}}} = \eta_{\mathrm{eff}}\left(2\hat{\boldsymbol{D}}^{t+\Delta t^{\mathrm{e}}} + \frac{\boldsymbol{\tau}^{t}}{\mu\Delta t^{\mathrm{e}}} + \frac{\mathbf{W}^{t}\boldsymbol{\tau}^{t} - \boldsymbol{\tau}^{t}\mathbf{W}^{t}}{\mu}\right). \tag{29}$$

Our system of equations is thus composed of a quasi-viscous part with modified material parameters and a right-hand-side term depending on values from the previous timestep. This approach minimizes the modification to the viscous flow code. Instead of using physical parameters for viscosity we use an effective value (28) to take into account elasticity, then, during computations for the force term, we add elastic internal stresses from the previous timestep or from initial conditions

$$F_i^{\mathrm{e},t} = -\frac{\eta_{\mathrm{eff}}}{\mu\Delta t^{\mathrm{e}}}\tau_{ij,j}^{t}. \tag{30}$$

Therefore (2) becomes

$$\tau_{ij,j}^{t+\Delta t^{\mathrm{e}}} - p_{,i} + f_i + F_i^{\mathrm{e},t} = 0, \tag{31}$$

which we solve by substituting for $\boldsymbol{\tau}^{t+\Delta t^{\mathrm{e}}}$ to give a set of equations for velocity unknowns.

In choosing a material timescale ($\Delta t^{\mathrm{e}}$) independent of the numerical advection timestep ($\Delta t$), it is necessary for a given particle to carry with it a stress history of several advection timesteps (corresponding to several complete solutions for the stress field). This is achieved through an averaging procedure in which the stress tensor stored on a given particle is averaged with the newly calculated stress tensor ($\boldsymbol{\tau}^{t}$) at the same location:

$$\boldsymbol{\tau}^{t+\Delta t} = \left(1 - \frac{\Delta t}{\Delta t^{\mathrm{e}}}\right)\boldsymbol{\tau}^{t} + \frac{\Delta t}{\Delta t^{\mathrm{e}}}\boldsymbol{\tau}^{t+\Delta t^{\mathrm{e}}}. \tag{32}$$

This stress is then advected and rotated with the particle to give the updated stored stresses $\boldsymbol{\tau}^{t+\Delta t}$.

### 5.2. Yielding

In a geological context we frequently deal with situations where part of the system is subjected to very high stresses. Under such conditions the material fails, but, unlike many practical engineering simulations, we are interested in simulating in the post-failure behavior up to very large strains.

On the basis that the post yield deformation trends increasingly to dominant, simple structures with increasing strain [3], our geological modeling at scales of tens to thousands of kilometers uses very simple descriptions of yielding. Brittle behavior is parameterized using a non-linear effective viscosity which is introduced whenever the stress would otherwise exceed the yield value $\boldsymbol{\tau}_{\text{yield}}$. This approach ignores details of individual faults, and treats only the influence of fault systems on the large scale convective flow.

To determine the value of the effective viscosity at any point we extend (23) by introducing a Prandtl–Reuss flow rule for the plastic part of the stretching:

$$\frac{\overset{\triangledown}{\boldsymbol{\tau}}}{2\mu} + \frac{\boldsymbol{\tau}}{2\eta} + \lambda \frac{\boldsymbol{\tau}}{2|\boldsymbol{\tau}|} = \widehat{\boldsymbol{D}_e} + \hat{\boldsymbol{D}}_v + \widehat{\boldsymbol{D}_p} = \hat{\boldsymbol{D}}, \tag{33}$$

where $\lambda$ is a parameter to be determined such that the stress remains on the yield surface, and $|\boldsymbol{\tau}| \equiv (\tau_{ij}\tau_{ij}/2)^{(1/2)}$. The plastic flow rule introduces a non-linearity into the constitutive law which, in general, requires iteration to determine the equilibrium state.

The implementation is as follows, starting from equation (33), we again express the Jaumann stress rate in first order difference form (using the Lagrangian particle reference frame):

$$\boldsymbol{\tau}^{t+\Delta t^{\mathrm{e}}}\left[\frac{1}{2\mu\Delta t^{\mathrm{e}}} + \frac{1}{2\eta} + \frac{\lambda}{2|\tau|} + \right] = \hat{\boldsymbol{D}}^{t+\Delta t^{\mathrm{e}}} + \frac{1}{2\mu\Delta t^{\mathrm{e}}}\boldsymbol{\tau}^{t} + \frac{1}{2\mu}(\mathbf{W}^{t}\boldsymbol{\tau}^{t} - \boldsymbol{\tau}^{t}\mathbf{W}^{t}). \tag{34}$$

No modification to the isotropic part of the problem is required when the von Mises yield criterion is used. At yield we use the fact that $|\boldsymbol{\tau}| = \tau_{\text{yield}}$ to write

$$\boldsymbol{\tau}^{t+\Delta t^{\mathrm{e}}} = \eta'\left[2\hat{\boldsymbol{D}}^{t+\Delta t^{\mathrm{e}}} + \frac{1}{\mu\Delta t^{\mathrm{e}}}\boldsymbol{\tau}^{t} + \frac{1}{\mu}(\mathbf{W}^{t}\boldsymbol{\tau}^{t} - \boldsymbol{\tau}^{t}\mathbf{W}^{t})\right] \tag{35}$$

using an effective viscosity, $\eta'$ given by

$$\eta' = \frac{\eta\tau_{\text{yield}}\mu\Delta t^{\mathrm{e}}}{\eta\tau_{\text{yield}} + \tau_{\text{yield}}\mu\Delta t^{\mathrm{e}} + \lambda\eta\mu\Delta t^{\mathrm{e}}}. \tag{36}$$

We determine $\lambda$ by equating the value of $|\boldsymbol{\tau}^{t+\Delta t^{\mathrm{e}}}|$ with the yield stress in (35). Alternatively, in this particular case, we can obtain $\eta'$ directly as

$$\eta' = \tau_{\text{yield}}/\left|\hat{\boldsymbol{D}}_{\text{eff}}\right|, \tag{37}$$

where

$$\hat{\boldsymbol{D}}_{\text{eff}} = 2\hat{\boldsymbol{D}}^{t+\Delta t^{\mathrm{e}}} + \frac{1}{\mu\Delta t^{\mathrm{e}}}\boldsymbol{\tau}^{t} + \frac{1}{\mu}(\mathbf{W}^{t}\boldsymbol{\tau}^{t} - \boldsymbol{\tau}^{t}\mathbf{W}^{t}) \tag{38}$$

and $|\boldsymbol{D}| = (2D_{ij}D_{ij})^{1/2}$.

The value of $\lambda$ or $\eta'$ is iterated to allow stress to redistribute from points which become unloaded. The iteration is repeated until the velocity solution is unchanged to within the error tolerance required for the solution as a whole.

## 6. Applications

### 6.1. Stationary driven cavity flow

Our first application example is a driven viscous cavity flow problem with a time-independent velocity solutions. The positions of the integration point particles are continually updated by the flow, however, which gives an indication of the sensitivity of the solution to the integration scheme.

The box is a unit square with free-slip boundaries everywhere except on the top between $x = 0.4$ and $x = 0.6$ where a horizontal velocity of 1 is applied towards the right. Some of the particles are "dyed" to track the material motion through time (Fig. 4).

Intense deformation in the corners results in a need to introduce new particles resulting in an increasing number of particles as a function of time (Fig. 5).

We compute velocity errors by comparison with a very fine mesh solution. This solution is obtained at the first timestep for a regular square mesh of 192 × 192 elements using a Gaussian integration scheme of 4 integration points per element. Our study model is a 48 × 48 elements with either Gaussian integration points or evenly distributed and weighted points. The error due to the use of a coarser mesh is about 2.8%, the remaining error is due to the integration scheme approximation.

The error is computed as follows:

$$\mathrm{Err} = \frac{\sqrt{|\boldsymbol{V}_x - \boldsymbol{v}_x|^2 + |\boldsymbol{V}_z - \boldsymbol{v}_z|^2}}{\sqrt{\boldsymbol{V}_x^2 + \boldsymbol{V}_z^2}} \tag{39}$$

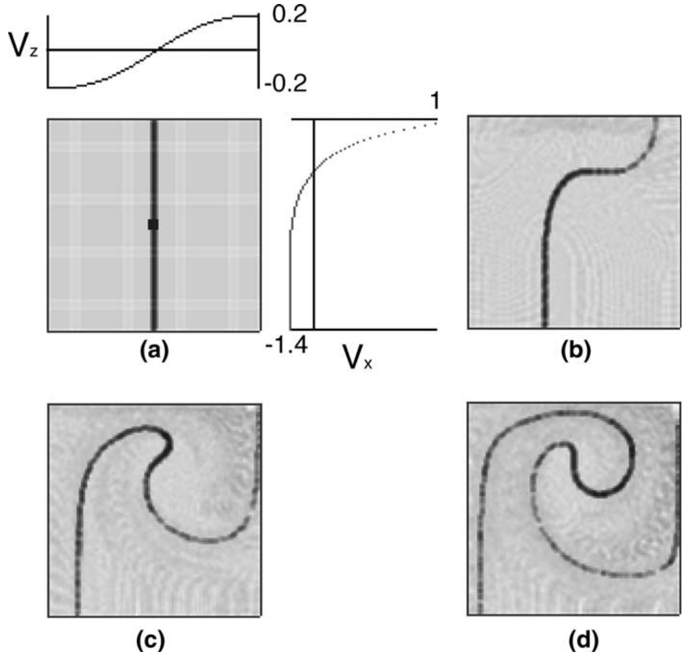

Fig. 4. Geometry of the convection model at initial timestep (a) with velocity profiles, timestep = 150 (b), 500 (c), and 800 (d).

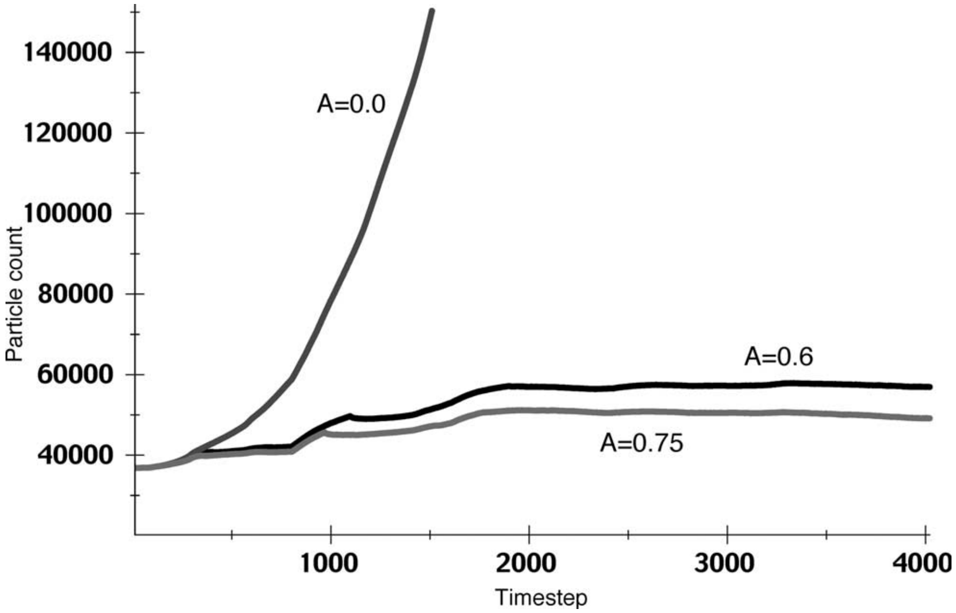

Fig. 5. Total number of particles versus timestep for the driven cavity benchmark. The free parameter is the "appetite" of the particles which controls when neighboring particles of the same type merge.

where $\boldsymbol{V}$ is the reference velocity field obtained with the fine mesh and $\boldsymbol{v}$ is a coarse-grid velocity field. The error is computed over all mesh points in the coarse grid. Firstly we compare results between a Gaussian integration scheme and a scheme with evenly distributed and weighted particles. The Gauss scheme does not allow for the particles to move, so is only applicable on the first timestep. The particle-based integration scheme was used over several tens of timesteps with two different numbers of integration points. At each step the particle weights were recomputed to satisfy constant and linear terms in the constraint equations (14). Although the particle based integration scheme was never better than a four-point Gauss scheme, the accuracy was comparable through time, particularly when larger numbers of particles were used.

Recomputing the particle weights is a computationally costly step not required in the Material Point Method. In Fig. 6, we show the accuracy of integration for the driven convection problem when the weights are fixed, when they satisfy the constant terms of the constraints, and when the constant and linear terms are satisfied.

We ran each model for several hundred timesteps with initially 16 particles per element and evaluated the accuracy of the results as well as the fluctuation of the solution. Fluctuation was computed using a moving window of 15 points width to give the average error ($\mathrm{Ave}(t)$) and its variance ($\mathrm{Var}(t)$). Then we plotted for each case $\mathrm{Ave}(t) + \mathrm{Var}(t)$ and $\mathrm{Ave}(t) - \mathrm{Var}(t)$.

When we keep the particle weight constant so that the particle mass (volume) is conserved, an *element* mass/volume may change when particles cross an element boundary. This simple scheme is associated with large errors with large fluctuations which becomes worse for very large deformation (Fig. 6(a)) despite an increasing number of particles.

Modifying the weights stabilizes the error through time with a mean value close to that of the Gaussian integration scheme. The variation in the error is considerably smaller when the constraint equation (14) for both constant and linear terms is applied (Figs. 6(b) and (c)).

Another reason to include the constraints from (14) up to the linear terms comes from the typical problems we tackle in geological modeling. Invariably there are a strong vertical gravitational forces which

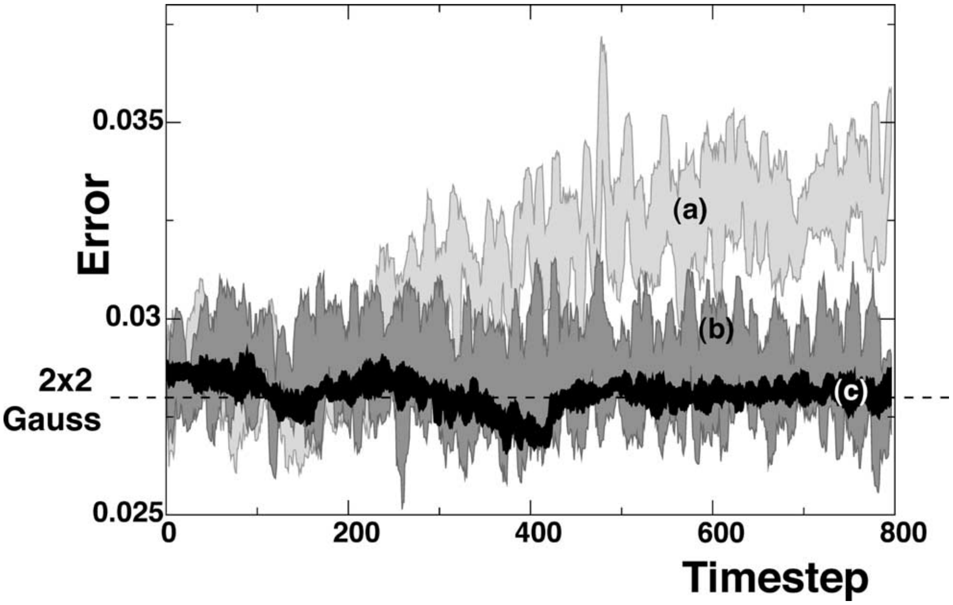

Fig. 6. Error versus time for different particle weighting schemes: (a) particle weights are conserved exactly through time (except for particle splitting), (b) particle weights are adjusted to fit the constant terms of the constraints, (c) particle weights are adjusted to fit the constant and the linear terms of the constraints. The accuracy of this mesh using the Gaussian integration scheme is shown by the dashed line.

are almost exactly balanced by vertical pressure gradients. These must be integrated very carefully to ensure that they do not produce a component of flow since even a small error contribution from these hydrostatic terms would easily dominate the solution.

### 6.2. Extension of a layer

The yielding algorithm is benchmarked by measuring the second invariant of the stress and displacement at points within a viscoelastic beam which was extended or compressed at a fixed rate, $v = 5$, by an imposed velocity boundary condition at one end. Fig. 7 indicates the geometry of the numerical experiment: the mesh was initially 3 units long by 1 unit high. The sample was 0.5 units thick, occupying the central half of

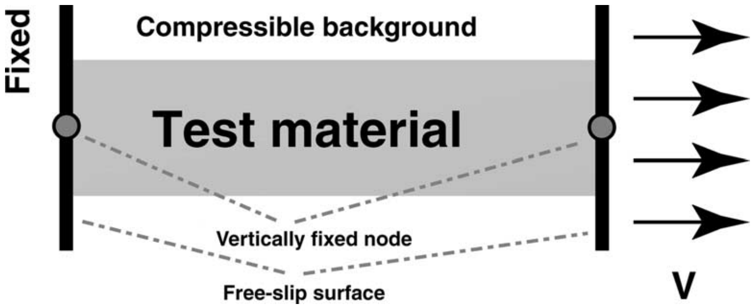

Fig. 7. Geometry for simulation of the extension of a viscoelastic bar with a yield stress.

the mesh, and was surrounded by a low viscosity, compressible material. Three sampling points (a,b,c) for recording the stress invariant and displacement were chosen within the sample initially placed along the mid-line at $x = 0.2, 0.5, 0.8$.

The material parameters of the sample ($\eta = 10^8$, $\mu = 10^6$) were chosen such that the relaxation time was long ($\alpha = \eta/\mu = 100$) compared to the duration of the experiment (0.25) so that the material behaved almost as an elastic solid (high Deborah number, *De*, defined as relaxation time/observation time).

Fig. 8 shows the progress of the experiment. Initially, deformation was uniform, resulting in gradual stretching of the sample ($t \leqslant 0.180$). The entire sample reached the yield point at the same time ($t = 0.212$) and initially deformed uniformly with all points yielding. However, the deformation soon localized to a number of shear bands ($t = 0.220$), then to two places along the sample ($t = 0.227$), and finally to a single location ($t = 0.2359$) which focused all subsequent deformation until the sample failed entirely ($t \geqslant 0.2404$). The frames are not uniformly spaced in time since the post-failure behavior occurred on a much shorter timescale than the gradual loading. Note, for example, that the necking

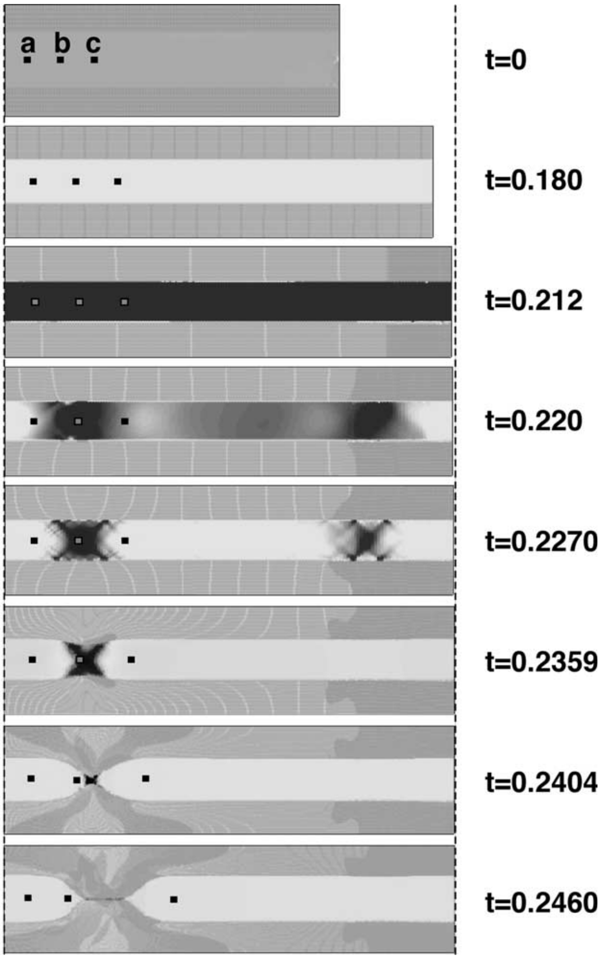

Fig. 8. Simulation of the extension of a viscoelastic bar with yield stress. Black shading indicates regions deforming at yield. Embedded marker points which follow the material deformation are indicated by a,b,c.

and separation of the two parts of the sample occurred with barely any movement of the end boundary.

Even with a material which has no strain softening, there is a tendency for deformation to localize in a particle-in-cell representation of the sample. This occurs because the sample boundary is never perfectly flat (as in real life) due to numerical fluctuations in the particle locations, and to a mild interference (moiré) effect between the array of particles and the underlying grid. These effects produce small fluctuations in the stress field which can result in early failure at certain points. Once nucleated, shear bands can propagate from these points—ultimately resulting in necking and complete separation of the two halves of the sample.

In general, we can force imperfections into the sample by introducing tiny perturbations to the material properties (e.g., in this case, some slight initial damage). This has the advantage that the perturbations have a physical expression which is not mesh dependent, and that the influence of mesh density or particle density on the solution can be examined independently.

In Fig. 9(a), we plot the stress at each of the sample points in the material as a function of time for a fixed end velocity. The evolution of stress within the beam was close to linear—apart from the influence of the changing of the beam thickness during deformation. The yield stress of the material was $3 \times 10^5$. The stress increased within the sample at the same rate for all the sample points until the yield stress was reached. At this stage, the stress was not able to increase any further, and the material deformed uniformly at the yield stress. Once localization had occurred, however, points outside the necking area begin to unload, and the stress dropped dramatically. The rate at which stress drops from yield back to zero is governed by the viscous part of the rheology, and the presence of a low viscosity background material.

The unloading is more clearly seen in the plots of the displacement of the sample points through time in Fig. 9(b). Before yielding, the displacement of each sample point increased monotonically. Once yielding occurred, and the deformation localized, the sample points on the left of the break (a,b), under the action of stored elastic stresses, rapidly retreated towards their original locations. The sample point on the right of the break (c) moved rapidly to the right as the elastic deformation relaxed.

It is worth discussing at this point a consequence of the fact that the yield criterion only applies to the deviatoric stress. During the separation of the layer, the pressure becomes enormous at the constriction, which obviously could not occur in a real material. To model this situation in a more realistic manner we

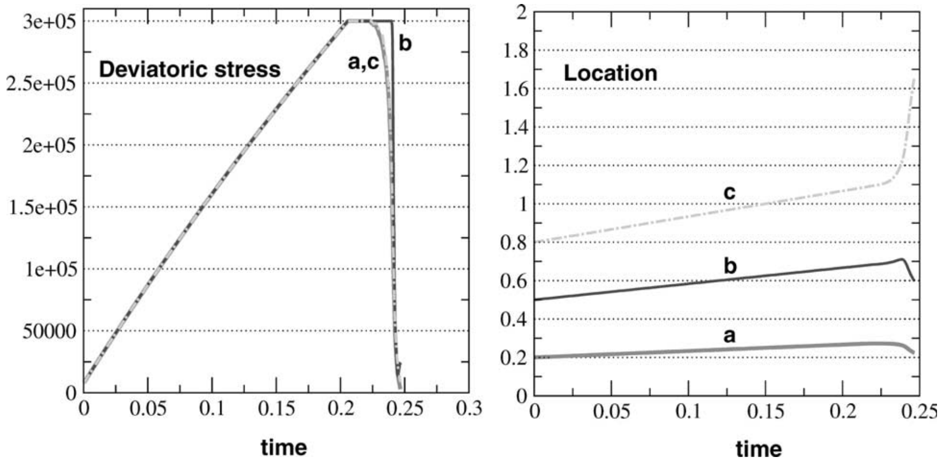

Fig. 9. Stress, and displacement at sample points a,b,c as a function of time for the extension experiment of Fig. 8.

would need to complement the yield criterion on the deviatoric stress with a suitable tension cutoff condition.

### 6.3. Thermal convection with suspended elliptical crystals

In addition to the set of equations described so far, an energy equation can be solved explicitly in conjunction with the time-marching scheme:

$$\frac{\partial T}{\partial t} + v_i \frac{\partial T}{\partial x_i} = \kappa \frac{\partial^2 T}{\partial x_j^2}, \tag{40}$$

where $x_i$ are the coordinates, $v_i$ is the velocity, $T$ the temperature, and $\kappa$ is the thermal diffusivity.

The convection solution in the absence of suspended particles is uniquely described by a single dimensionless number: the Rayleigh number, $Ra = g\rho\alpha\Delta T/\eta\kappa$, where $g$ is the gravitational acceleration, $\rho$ is the reference density, $\Delta T$ is the temperature contrast across the layer, and $\eta$ is viscosity. The density contrast in the fluid due to temperature variations is $(\alpha\rho\Delta T)$, and the ratio of density between fluid and solid at equivalent temperature is denoted by $\beta$.

We suspended a set of solid crystals with $\beta = 10$ in a steady-state convecting viscous fluid with $Ra = 10^5$ in an initially regular pattern (see Fig. 10(a)), and let them settle (Fig. 10(b)). When the crystals had sunk to the bottom of the layer (Fig. 10(c)), the convective circulation was mainly confined to the upper part of the layer where there were no crystals. Within the lower part of the layer, fluid percolated between the densely packed solid crystals.

Once particles had settled (see Fig. 10(c)) $\beta$ was arbitrarily changed to 0.5 which immediately reversed the density contrast between fluid and crystals and resulted in an full-scale overturn of the whole layer (see Figs. 10(d) and (e)). In the end particles reached a new equilibrium (Fig. 10(f)) on top of the convecting fluid.

This simple demonstration shows the promise of the method in simulating the dynamics of crystal-rich magmas, potentially accounting for evolving composition which would alter the density contrast between liquid and solid during the course of the simulation.

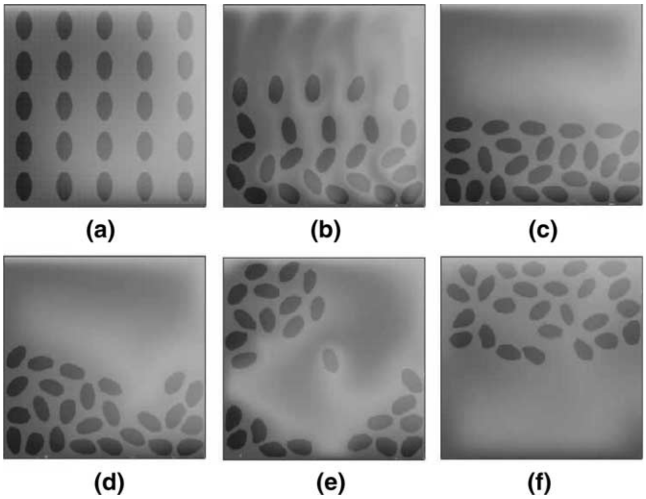

Fig. 10. Numerical simulation of the settling of dense crystals and the uplift of light crystals in a convecting viscous fluid. (a) Initial setup, (b) settled crystals, (c) time at which density is decreased, (d) and (e) light crystals rising, and (f) steady-state.

### 6.4. Plate tectonic simulation with viscoelastic rheology

The final example is a highly simplified form of the target problem (convection in the Earth's mantle with brittle, viscoelastic lithosphere). The model is a a two-dimensional box heated from below with a temperature- and pressure-dependent viscosity law, and constant elastic modulus.

$$\eta = \eta_0 \exp\left(\frac{e + qz}{T + T_0}\right), \tag{41}$$

where $\eta_0$ is a reference viscosity value, $e$ and $q$ are rheological parameters, $z$ is depth, $T$ is temperature, and $T_0$ is a reference temperature. The yield stress, $\tau_{\mathrm{y}}$, is given by

$$\tau_{\mathrm{y}} = (a + bp)(1 - c\epsilon), \tag{42}$$

where $a$, $b$, and $c$ are material parameters, $p$ is pressure, and $\epsilon$ is a scalar measure of strain obtained by integrating the second invariant of the strain-rate tensor with time. $\tau_{\mathrm{y}}$ is truncated at a minimum value to limit the weakening effect of accumulated strain.

In this model shown in Fig. 11 the Rayleigh number was $5 \times 10^6$, the elastic modulus was $10^6$, and the viscosity parameters were $\eta_0 = 0.1995$, $e = 1.773$, $q = 3.74$, $T_0 = 0.1$ which gave a surface viscosity of $10^7$, a viscosity change due to temperature of $10^7$, and a viscosity change due to depth of 30. Yielding parameters were set to $a = 2 \times 10^5$, $b = 0.05$, $c = 0.4$ with truncation on the yield stress such that $10^4 < \tau_{\mathrm{y}} < 5 \times 10^5$. The value of $\Delta t_e$ chosen was 0.01, with the timestep for the particle-position update chosen by a Courant condition determined from the shortest advection time across an element. The grid size was $40 \times 240$ elements, regularly spaced.

After starting from a perturbed conductive state, the system has reached a steady convecting state with three downwellings and three upwellings. The surface velocity, plotted at the top of the figure, is broken into six distinct regions with sharp boundaries. The movement of the surface is possible because failure occurs in the strong, viscoelastic cold thermal boundary layer (indicated by light bands in the temperature

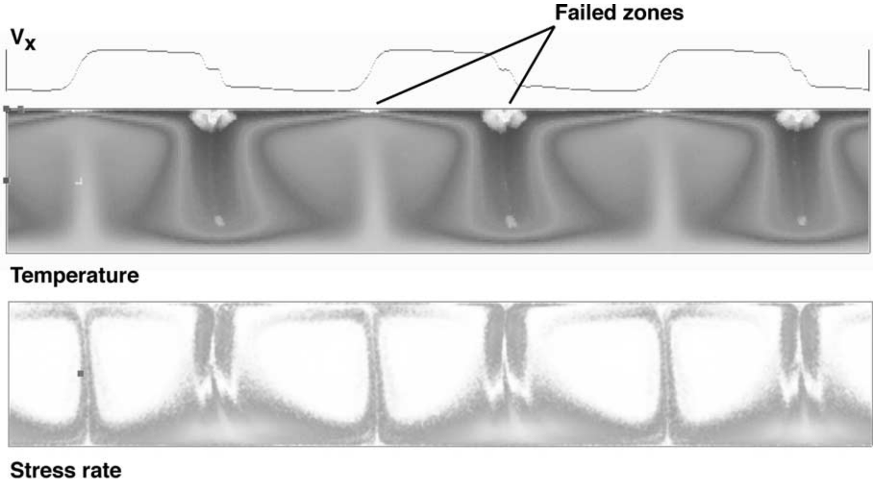

Fig. 11. Numerical simulation of convection in a fluid with viscoelastic rheology and a yield stress showing temperature (top) and the second invariant of stress rate (bottom). For a suitable range of Rayleigh number and yield stress the surface velocity, $v_x$, breaks up into a number of distinct 'plates'.

plot). The plot of the invariant of the stress-rate shows that the elastic stresses are only important in the cool parts of the system (the upper cold thermal boundary layer and the descending 'slabs').

## 7. Conclusions

We have presented a Lagrangian integration point finite element formulation designed to handle large deformation for viscoelastic materials. The scheme derives from the Material Point Method but differs in a number of important aspects, including the fact that it is based on a fast-implicit solution method, and that it includes various particle reweighting steps which improve accuracy in the fluid-deformation limit.

We have demonstrated that the particle-in-cell finite element scheme is comparable in accuracy with traditional Eulerian finite element methods for fluid dynamical problems.

The principal advantages of the Lagrangian Integration Point method are: the ability to continue to extremely large deformation without significant change in accuracy; the ability to track material history and interfaces through time with accuracy comparable to more specialized methods; the fact that a regular grid is preserved throughout the run which allows fast numerical solvers to be employed.

Disadvantages include the fact that the hybrid mesh/particle approach does not allow particles to communicate directly, and therefore lacks some of the flexibility of pure-particle codes. Secondly, the resolution is related to the grid point spacing, not the finer particle spacing. Thirdly, while it is conceptually simple to retain the mesh, and computationally convenient, the storage requirement is very high.

We also have not considered the situation of mesh refinement. When element sizes vary, our current assumption that accurate integration of every element will be satisfied by uniformly high particle densities is no longer valid.

## References

[1] M. Barazangi, J. Dorman, World seismicity maps compiled from essa, coast and geodetic survey, epicenter data 1961–1967, Bull. Seismol. Soc. Am. 59 (1969) 369–380.

[2] T. Belytschko, Y.Y. Lu, L. Gu, Element-free galerkin methods, Int. J. Numer. Methods Eng. 37 (1994) 229–256.

[3] Y. Ben-Zion, C. Sammis, Characterization of fault zones, Pure Appl. Geophys., 2001 (submitted).

[4] J. Braun, M. Sambridge, Dynamical lagrangian remeshing (dlr): a new algortihm for solving large strain deformation problems and its application to fault-propagation folding, Earth Planet. Sci. Lett. 124 (1994) 211–220.

[5] J. Braun, M. Sambridge, A numerical method for solving partial differential equations on highly irregular evolving grids, Nature 379 (1995) 655–660.

[6] J. Cahouet, J.-P. Chabard, Some fast 3d finite element solvers for the generalized stokes' problem, Int. J. Numer. Methods Fluids 8 (1988) 869–895.

[7] P. de Zeeuw, Matrix-dependent prolongations and restrictions in a blackbox multigrid solver, J. Comp. Appl. Math. 33 (1990) 1–27.

[8] P.M. Gresho, R.L. Lee, R.L. Sani, Advection-dominated flows, with emphasis on the consequences of mass lumping, in: Finite Elements in Fluids, vol. 3, Wiley and Sons, Chichester, 1978, pp. 335–350.

[9] T.J.R. Hughes, The Finite Element Method, Prentice-Hall, Inc., Englewood Cliffs, NJ 07632, 1987.

[10] J.J. Monaghan, Smoothed particle hydrodynamics, Annu. Rev. Astron. Astrophys. 30 (1992) 543–574.

[11] L.N. Moresi, V.S. Solomatov, Numerical investigation of 2d convection with extremely large viscosity variations, Phys. Fluids 7 (9) (1995) 2154–2162.

[12] L.N. Moresi, V.S. Solomatov, Mantle convection with a brittle lithosphere: thoughts on the global tectonic styles of the Earth and Venus, Geophys. J. Int. 133 (1998) 669–682.

[13] H.-B. Mühlhaus, L. Moresi, F. Dufour, B. Hobbs, The interplay of material and geometric instabilities in large deformations of viscous rock. In: B. Karihaloo (Ed.), Proceedings of the IUTAM Symposium on Analytical and Computational Fracture Mechanics of Non-Homogeneous Materials, Kluwer, 2001 (in press).

[14] P.K. Smoliarkiewicz, T.L. Clark, The multidimensional positive definite advection transport algorithm: further developments and applications, J. Comput. Phys. (1986).

[15] D. Sulsky, Z. Chen, H.L. Schreyer, A particle method for history-dependent materials, Comput. Methods Appl. Mech. Eng. 118 (1994) 179–196.

[16] D. Sulsky, H. Schreyer, Antisymmetric form of the material point method with applications to upsetting and taylor impact problems, Comput. Methods Appl. Mech. Eng. 139 (1996) 409–429.

[17] A.B. Watts, J.H. Bodine, N.M. Ribe, Observations of flexure and the geological evolution of the pacific basin, Nature 283 (1980) 532–537.

[18] C. Wijns, F. Boschetti, L. Moresi, Inversion in geology by interactive evolutionary computation. J. Struct. Geol, 2001 (submitted).

[19] C. Zhao, B.E. Hobbs, H.-B. Mühlhaus, A. Ord, A consistent point-searching algorithm for solution interpolation in unstructured meshes consisting of 4-node bilinear quadrilateral elements, Int. J. Numer. Methods Eng. 45 (1999) 1509–1526.