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

Louis Moresi, Frédéric Dufour, Hans Muhlhaus

▶ **To cite this version:**

Louis Moresi, Frédéric Dufour, Hans Muhlhaus. A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. Journal of Computational Physics, 2003, 184, pp.476-497. hal-00333883

HAL Id: hal-00333883
https://hal.science/hal-00333883

Submitted on 27 Oct 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Journal Of Computational Physics, 184:476-497, 2003

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

L. Moresi, F. Dufour, H.-B. Mühlhaus

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.

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 (Wijns et al., 2001 [20]).

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 (Mühlhaus et al, 2001 [15]). 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. 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 behaviour 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 (Watts et al, 1980 [19]). The strong correlation between seismicity and plate boundaries (e.g. Barazangi & Dorman, 1969 [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 behaviour 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 ELLIPSIS Lagrangian Integration Point code: a method for simulating viscoelastic-brittle materials in fluid-like deformation. The method is tested on a number of very simple benchmark cases where analytic solutions are known, or where the accuracy can otherwise be quantified in large deformation.

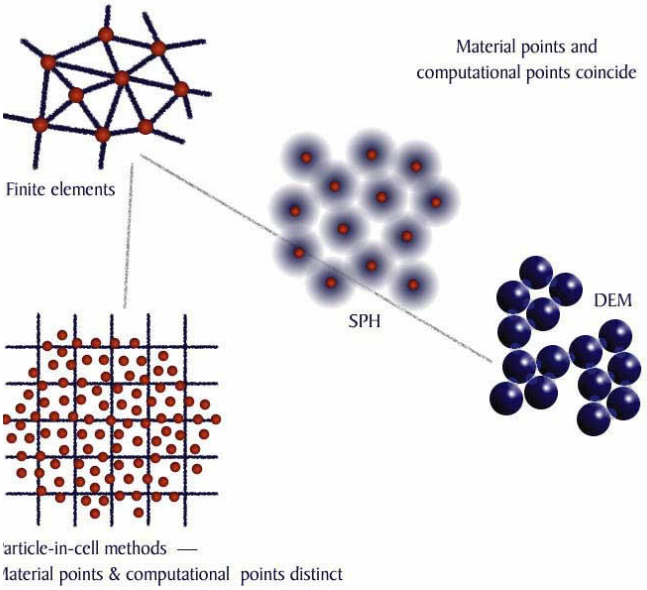

Figure 1: Different ways to discretize a problem provide a natural representation for systems with different controlling physics. In fully Lagrangian FEM, SPH and DEM all the computational points are also material points. In the Lagrangian integration point method, computational points are not material points but a set of material points is also tracked. In ALE and Eulerian FEM there is no tracking of material points.

modeling techniques

The key to all these problems is to find a way to deal efficiently with the most general case: finite strain viscoelasticity. This is a specialized problem: most engineering codes are optimized to study the modest strains which accumulate prior to failure. Nevertheless, an enormous amount 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.

Broadly speaking there have been two approaches to dealing with this problem. One approach is to retain the mesh, but to reorganize it periodically and, perhaps, modify the method to cope with a less-than-optimal distribution of discretization points, and the other is to dispense with the mesh entirely for computing mechanical behavior and derive a formulation where the material points interact with each other directly. The approach used in Ellipsis is distinct from these methods in relaxing the requirement that material points and computational points need to be the same (see Figure 1)

Highly deforming mesh methods

Mesh based Lagrangian methods are not particularly well suited to very large strain fluid flow applications as the mesh is subject to tangling and considerable effort is required to regrid or reconnect the mesh to prevent the computation of derivatives and shape functions from becoming inaccurate. The Arbitrary Lagrangian Eulerian method (Huerta and Liu 1987 [9]) avoids mesh tangling by allowing computational points to move independently from the underlying material. This strategy makes it possible to prevent mesh points from approaching one another too closely, while still preserving important interface details. However, additional advection terms are required to handle transport of quantities relative to the mesh and these terms are dispersive.

The Dynamical Lagrangian Remeshing method (Braun and Sambridge 1994 [6]) reconnects all mesh points after any deformation to ensure the mesh always remains optimally configured (in DLR this means the mesh is always a Delaunay triangulation). Stress histories can be compute at integration points and therefore need to be interpolated when new element connections occur. The extension to 3D is non-trivial and the method is limited to linear triangles/tetrahedra.

An alternative approach, the Natural Element Method (NEM, Braun and Sambridge 1995 [7]) uses a natural-neighbor interpolation scheme over a Delaunay triangulation to develop finite-element-like shape functions associated with each computational, material point. The Natural Element Method (NEM) does not suffer from the same difficulties as FEM when elements are highly distorted and, although it uses a triangulated mesh, has the advantage that the interpolation functions are continuously differentiable away from the computational points. NEM shares with DLR the difficulty associated with preserving tensor information at integration points after reconnection of the mesh. Braun and Sambridge ([7]) point out that the use of a standard gaussian integration scheme in NEM is not exact for natural neighbor interpolation functions. Another complication is that the shape functions overlap element boundaries, and therefore descriptions of flux boundary conditions is not trivial and may not be accurate.

Meshless methods

Meshless methods include the Discrete Element Method (DEM, see for example Sakaguchi and

Mühlhaus 1997 [16]), Smoothed Particles Hydrodynamics (SPH, Monaghan 1992 [12]), Element Free Galerkin (EFG, Belytschko et al 1994 [2]), and the Point Integration Method (PIM, Liu & Gu, 2001 [11]).
In the DEM, computational points are associated with a finite size and shape. Interaction between points (particles) occurs only where they are in contact or close neighbors - according to the specified interaction rules. This is a genuinely discrete method with no homogenization of the particle-particle interactions through a continuum description. The DEM is very well suited to modeling of fracture since particle interactions can take the form of breakable bonds. The interaction of a particle with it's neighbours varies with angle and time, since it is a function of the neighbours' coordinates, and the interaction history.
SPH and EFG are continuum methods in which the material points are associated with a basis function which can be used as an interpolant for field variables, and to develop an expansion for PDEs. The basis functions overlap other particles, making it more complicated to apply constraints to values at individual material points but allowing particles to pass arbitrarily close to one another during simulations of fluid motion with stagnation points.
PIM is also a continuum method. In PIM, global basis functions are defined which can be combined to produce non-overlapping shape functions. The method therefore eliminates one of the principal obstacles in efficient formulation of meshless solvers. However, very little in the way of practical applications for the method has been published to date so the promise of the method is largely unexplored. SPH, EFG and PIM are formulated in such a way that implicit solution methods or explicit methods can be used.

Lagrangian integration points

The method used in Ellipsis is a hybrid mesh / particle code —- the idea being to retain the generality and robustness of mesh-based FEM, and capture the geometrical flexibility of a fully-Lagrangian set of particles for tracking material deformation. The computational points are a set of nodes fixed in space, connected by a mesh of finite elements. An independent set of material points which carry material properties and the solution history is embedded in the mesh. Since the material and computational points have been formally separated, a strategy for coupling the two representations is needed. The usual finite element interpolation of nodal point values to element interiors is used to update the locations of particles and history variables. The particle properties are coupled to the computational mesh through a non-standard element quadrature in which the particles which happen to be in a given element are used as integration points. Boundary conditions constrain unknowns directly and therefore belong with the mesh points.
This formulation is very close to the Material Point Method (MPM) developed by Sulsky, Schreyer and their coworkers (e.g. Sulsky & Schreyer, 1996 [18]) who use an Eulerian mesh with Lagrangian particles. Their formulation was derived from problems where momentum dominates: impacts between elastoviscoplastic materials, suspensions in fast-moving fluids, and fast granular flows. The strength of the method comes from the incremental way in which particle quantities (including momentum) are updated from the mesh. Incompressible creeping flow can be treated by introducing a ficticious inertial term, but this is not ideal for the long-duration experiments we wish to simulate. The implicit formulation in our approach requires considerable modification to the method. However, when we introduce elasticity, there are strong parallels with the MPM.

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 particle-derived material property matrices. In later sections we discuss efficient grid-based solution methods which can be used with our formulation, and some implementation details.

Updating integration points

On one hand, we aim to build a finite element like formulation then we need integration points to integrate some quantities. On the other hand material particles are used to track history variables and deformation. The essence of our formulation (like in PIC method) is to use particles as 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, see (1)) provided all rates of change are computed particle-by-particle and updated appropriately during the advection phase. This eliminates the difficulty with numerical diffusion / dispersion of advected quantities.

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} \tag{1}$$

Particle velocities are interpolated from nodal velocities and then particle position is updated using a suitable integration scheme such as:

$$\boldsymbol{x}_p^{t+\Delta t} = \boldsymbol{x}_p^t + \Delta t \sum_{node} \boldsymbol{v}_n N_n(x_p) \tag{2}$$

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 4^{th} 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 2^{nd} order Runge-Kutta scheme - also known as the mid-point method- and reduce the timestep whenever higher accuracy in the time integration is required.

Variational form

The governing equations in domain Ω bounded by Γ are standard for creeping fluids, a conservation equation for momentum:

$$\tau_{ij,j} - \delta_{ij} p_{,j} + f_i = 0, \tag{3}$$

subject to a continuity requirement

$$v_{i,i} = 0, \tag{4}$$

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 $p_{,j}$ denotes differentiation of p with respect to x_j. The boundary conditions are given as follows:

$$\boldsymbol{\tau} \cdot n - p = \bar{t} \quad \text{on the natural boundary} \quad \Gamma_t \tag{5}$$

$$u = \bar{u} \quad \text{on the essential boundary} \quad \Gamma_u \tag{6}$$

in which the superposed bar denotes prescribed boundary values and n is the unit outward normal to domain Ω.
Inertial terms are not considered in (3); since the viscosity associated with creeping flow in rocks is enormous 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. The latter is not addressed in this paper.

$$\boldsymbol{f} = (0, -g\rho) \tag{7}$$

where g is the acceleration due to gravity (downwards) and ρ 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 (3) with the test functions N_i and (4) with Q. Integration over the volume and application of Gauss' theorem yield:

$$\int_\Omega \tau_{ij} N_{i,j} d\Omega - \int_\Omega p Q_{i,i} d\Omega = \int_\Omega f_i N_i d\Omega + \int_{\Gamma_t} \bar{t}_i N_i d\Gamma \tag{8}$$

and

$$\int_\Omega Q v_{i,i} d\Omega = 0 \tag{9}$$

Equation (8) must hold for all admissible N_i vanishing on boundary Γ_u where displacements are prescribed. The notation $N_{i,j}$ indicates the symmetrical form of the derivative: $\partial N_i/\partial x_j + \partial N_j/\partial x_i$.

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 Moresi & Solomatov (1995) [13].

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

$$\begin{pmatrix} \boldsymbol{K} & \boldsymbol{G} \\ \boldsymbol{G}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \\ \boldsymbol{0} \end{pmatrix} \tag{10}$$

where $\boldsymbol{K}$ is the global stiffness matrix, $\boldsymbol{G}$ 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.
In a mixed method, an arbitrary choice of shape functions for pressure and velocity can produce unreliable results. The particle-in-cell 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 lower order approximation than gaussian quadrature, the integration scheme based on particles 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

$$\boldsymbol{G}^T\boldsymbol{K}^{-1}\boldsymbol{K}\boldsymbol{v} + \boldsymbol{G}^T\boldsymbol{K}^{-1}\boldsymbol{G}\boldsymbol{p} = \boldsymbol{G}^T\boldsymbol{K}^{-1}\boldsymbol{F} \qquad (11)$$

and substituting the expression for $\boldsymbol{G}^T\boldsymbol{p}$ to obtain an expression entirely in the pressure:

$$\boldsymbol{G}^T\boldsymbol{K}^{-1}\boldsymbol{G}\boldsymbol{p} = \boldsymbol{G}^T\boldsymbol{K}^{-1}\boldsymbol{F} \qquad (12)$$

The procedure outlined in general by Cahouet & Chabard (1988 [8]) and for this specific system by Moresi & Solomatov (1995) [13] solves this pressure equation through a conjugate gradient series of solutions for velocities which can be accumulated to give the full solution. Hence the notion that we solve the pressure equation alone is an illusion.

Multigrid solver

The static, regular grid makes it possible to use (unmodified) a multigrid solver developed for purely Eulerian flow problems (Moresi & Solomatov, 1995 [13])
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.
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 Moresi & Solomatov (1998) [14]. In Ellipsis we use most of the time a full multigrid scheme with a direct solver scheme at the coarsest level.

Numerical implementation

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.

$$\int_{\Omega^e} \psi d\Omega^e \approx \sum_{p=1}^{P} w_p \psi(\boldsymbol{x}_p) \qquad (13)$$

where Ω^e is an element volume, ψ is a field quantity which is evaluated at a set of sample points with coordinates $\boldsymbol{x}_p$; 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 the particles notionally attached to 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 shall consider the one dimensional case in which ψ in (13) is a polynomial, and the integral is over the range -1 to 1 (a typical "master element").

$$\psi(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots \alpha_n x^n \qquad (14)$$

We integrate (14) algebraically over the domain, and equate coefficients with the quadrature expansion of the integral from (13) to obtain $n+1$ constraints on the set of w_p values:

$$\begin{aligned}
\sum_{p=1}^{\mathrm{n_{ep}}} w_p &= 2 && \text{(constant terms)} \\
\sum_{p=1}^{\mathrm{n_{ep}}} w_p x_p &= 0 && \text{(linear terms)} \\
\sum_{p=1}^{\mathrm{n_{ep}}} w_p x_p^2 &= \frac{2}{3} && \text{(quadratic terms)} \\
\sum_{p=1}^{\mathrm{n_{ep}}} w_p x_p^3 &= 0 && \text{(cubic terms)}
\end{aligned} \qquad (15)$$

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 would be very time-consuming, and may produce very poor results if the particles are unevenly distributed within an element, including 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 once the particle's positions evolve to a general configuration (the numbers of particles per element may vary), the accuracy of the integration scheme is relatively good for simulations where material strain is not

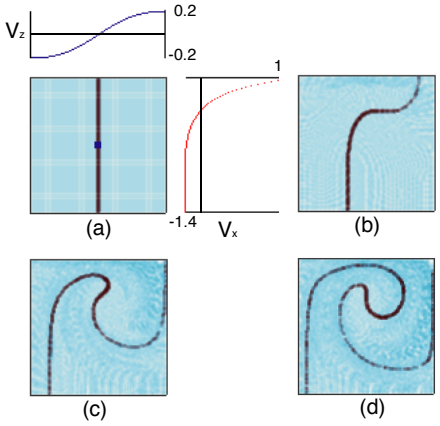

Figure 2: Geometry of the convection model at initial timestep (a) with velocity profiles, timestep = 150 (b), 500 (c) and 800 (d).

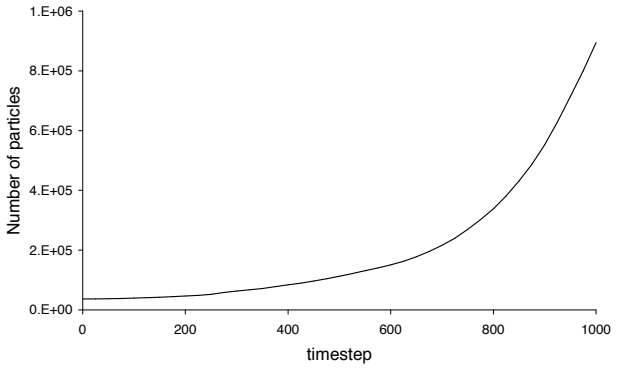

Figure 3: Total number of particles versus timestep for the driven cavity benchmark.

extreme (see benchmark section). With this in mind, we store the values of w_p determined from the local volume occupied by the particle in the initial configuration. These 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. , Hughes et al, 1987 [10]) .
To integrate a given order polynomial using particle locations, we generally 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 but are determined by where the particles happen to be within an element.

Integration scheme

In order to verify the integration scheme for dealing with large deformation problems, we model driven convection in an unit square box (Fig. 2). The default boundary condition is free-slip everywhere except on the top between $x = 0.4$ and $x = 0.6$ where a horizontal velocity is applied towards the right. We track with a darker line the motion of the material along the vertical mid-line through time (Fig. 2). The velocity solution is uniquely determined by the boundary conditions and is independent of time.
The flow carries particles around the box through time, however, which will stir any material interfaces, alter the distribution of particles within elements, and consequently may disrupt the integration scheme. 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.3). We treat the issue of particle splitting (a local remeshing step) in a subsequent section.
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% (Fig. 4.(a) and (c)), the remaining error is due to the integration scheme approximation.
The error is computed as follows:

$$\mathrm{Err} = \frac{\sqrt{|V_x - v_x|^2 + |V_z - v_z|^2}}{\sqrt{V_x^2 + V_z^2}} \qquad (16)$$

where V is the reference velocity field obtained with the fine mesh and v is a coarse-grid velocity field. 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 are recomputed to satisfy constant and linear terms in the constraint equations (15). Although the particle based integration scheme is never better than a four-point Gauss scheme, the accuracy is comparable through time, particularly when larger numbers of particles are used (Fig. 4)
Recomputing the particle weights is a computationally costly step not required in the Material Point Method which is a close ancestor of the scheme presented here. In Figure 5, 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 hundreds of timesteps with ini-

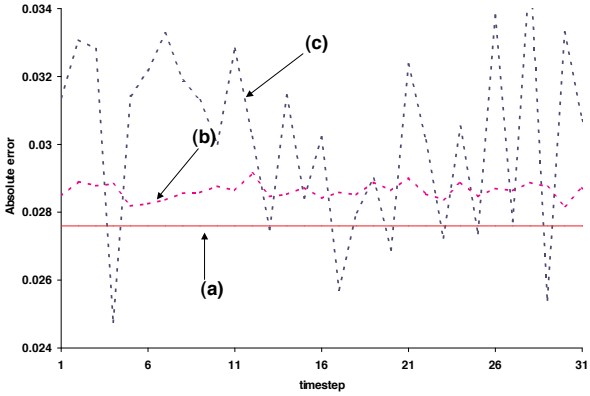

Figure 4: Error versus timestep for 4 Gaussian integration points (a), and for 4 (b) and 16 (c) particles initially evenly distributed within the element with equal weights.

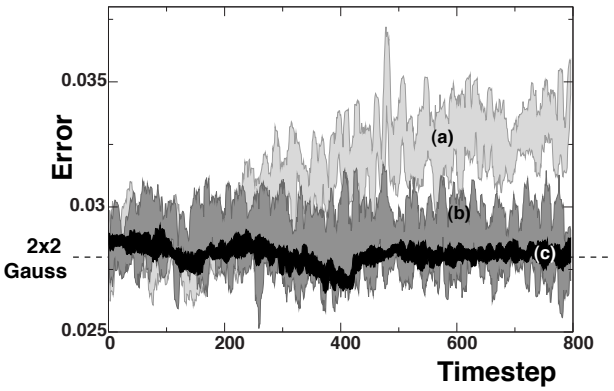

Figure 5: Error versus time for different particle weighting schemes: (a) particle weights are conserved exactly through time (except for particle splitting), (b) particle weights are adjusted to fit the constant terms of the constraints, (c) particle weights are adjusted to fit the constant and the linear terms of the constraints.

tially 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 point width to give the average error ($Ave(t)$) and its variance ($Var(t)$). Then we plotted for each case $Ave(t) + Var(t)$ and $Ave(t) - Var(t)$.

When we keep the particle weight constant so that the particle mass is conserved, an element mass 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. 5.(a)) despite an ever 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 (15) for both constant and linear terms is applied (Fig. 5(b,c)).

Another reason to include the constraints from (15) up to the linear terms comes from the typical problems we tackle in geological modeling. Invariably there are a strong vertical gravitational forces which 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.

The Gaussian integration is the most efficient of the schemes we have implemented, but is not suited to large deformation particle-in-cell methods. We obtain reasonable results using four particles (initially) per element and adjusting their weights as they are moved with the flow. Sixteen particles are needed before we have enough degrees of freedom to achieve the same integration as a four point Gauss scheme in 2D, allowing for the fact that some particles will be in the wrong position within an element to contribute to the integrals.

Element matrices and particle properties

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

$$k^e_{ab} = \int_{\Omega^e} \boldsymbol{B}^T_a \boldsymbol{D} \boldsymbol{B}_b d\Omega^e \tag{17}$$

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,

$$k^e_{ab} = \sum_{p=1}^{\mathrm{n_{ep}}} w_p \boldsymbol{B}^T_a(\boldsymbol{x}_p) \boldsymbol{D}_p \boldsymbol{B}_b(\boldsymbol{x}_p) \tag{18}$$

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

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 (Figure 6 (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 the element containing the tracer itself. In the worst case this may leave some elements entirely empty of tracers 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 Figure 6(b) where the heavily shaded particles sitting at the centroid of the salami-shaped local regions represent the original occupants of the volumes from Figure 6(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 Figure 6(c).
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 should occur when the distortion is relatively small.

Element inverse mapping

It is usual to change variables in the element integrals such as (17) 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 (1999) [21] 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 Figure 7 for the 2D case — the extension to three dimensions is straightforward. $\boldsymbol{e}_\xi$ and $\boldsymbol{e}_\eta$ are unit vectors in the 'natural directions' of the distorted element which map to the ξ and η axes respectively in the master element. h_ξ and h_η 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 $\boldsymbol{x}_p^0$,

$$\begin{aligned} \boldsymbol{\xi}_p &= (0,0) \\ \boldsymbol{x}_p^0 &= \left(\sum_{n=1}^{\mathrm{n_{en}}} N_n(\boldsymbol{\xi}_p) x_n, \sum_{n=1}^{\mathrm{n_{en}}} N_n(\boldsymbol{\xi}_p) y_n \right) \end{aligned} \tag{19}$$

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

$$\begin{aligned} \xi_p &\leftarrow \xi_p + \beta \left(e_{\xi x} x_p^i + e_{\xi y} x_p^i \right) / h_\xi \\ \eta_p &\leftarrow \eta_p + \beta \left(e_{\eta x} x_p^i + e_{\eta y} x_p^i \right) / h_\eta \\ \boldsymbol{x}_p^{i+1} &= \left(\sum_{n=1}^{\mathrm{n_{en}}} N_n(\boldsymbol{\xi}_p) x_n, \sum_{n=1}^{\mathrm{n_{en}}} N_n(\boldsymbol{\xi}_p) y_n \right) \end{aligned} \tag{20}$$

i is the iteration index, β 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 $\boldsymbol{\xi}$ 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)

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 (1994) [17], 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 is 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 to avoid the boundary nodes overtaking the particles, for example.

Constitutive Relationships

Viscoelastic formulation

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

$$\frac{\overset{\triangledown}{\boldsymbol{\tau}}}{2\mu} + \frac{\boldsymbol{\tau}}{2\eta} = \hat{\boldsymbol{D}}_v + \hat{\boldsymbol{D}}_e = \hat{\boldsymbol{D}} \tag{21}$$

where $\overset{\triangledown}{\boldsymbol{\tau}}$ is the Jaumann corotational stress rate for an element of the continuum, μ is the shear modulus and

η is shear viscosity. $\hat{\boldsymbol{D}}$ is the deviatoric part of $\boldsymbol{D}$.

$$\overset{\triangledown}{\boldsymbol{\tau}} = \dot{\boldsymbol{\tau}} + \boldsymbol{\tau}\mathbf{W} - \mathbf{W}\boldsymbol{\tau} \tag{22}$$

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

$$W_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} - \frac{\partial V_j}{\partial x_i}\right) \tag{23}$$

The $\boldsymbol{W}$ 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 (21) by expressing the Jaumann stress-rate in a difference form:

$$\overset{\triangledown}{\boldsymbol{\tau}}^{t+\Delta t^e} \approx \frac{\boldsymbol{\tau}^{t+\Delta t^e} - \boldsymbol{\tau}^t}{\Delta t^e} - \mathbf{W}^t\boldsymbol{\tau}^t + \boldsymbol{\tau}^t\mathbf{W}^t \tag{24}$$

where the superscripts $t, t+\Delta t^e$ indicate values at the current and future timestep respectively. Δt^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.
(21) becomes

$$\boldsymbol{\tau}^{t+\Delta t^e} = \frac{2\eta\Delta t^e}{\alpha+\Delta t^e}\hat{\boldsymbol{D}}^{t+\Delta t^e} + \frac{\alpha}{\alpha+\Delta t^e}\boldsymbol{\tau}^t + \frac{\alpha\Delta t^e}{\Delta t^e+\alpha}(\mathbf{W}^t\boldsymbol{\tau}^t - \boldsymbol{\tau}^t\mathbf{W}^t) \tag{25}$$

where $\alpha = \eta/\mu$ is the shear relaxation time.
We can simplify the above equations by defining an effective viscosity η_{eff}:

$$\eta_{\text{eff}} = \eta\frac{\Delta t^e}{\Delta t^e+\alpha} \tag{26}$$

Then the deviatoric stress is given by

$$\boldsymbol{\tau}^{t+\Delta t^e} = \eta_{\text{eff}}\left(2\hat{\boldsymbol{D}}^{t+\Delta t^e} + \frac{\boldsymbol{\tau}^t}{\mu\Delta t^e} + \frac{\mathbf{W}^t\boldsymbol{\tau}^t - \boldsymbol{\tau}^t\mathbf{W}^t}{\mu}\right) \tag{27}$$

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 (26) 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.

$$F_i^{e,t} = -\frac{\eta_{\text{eff}}}{\mu\Delta t^e}\tau_{ij,j}^t \tag{28}$$

Therefore (3) becomes

$$\tau_{ij,j}^{t+\Delta t^e} - \delta_{ij}p_{,i} + f_i + F_i^{e,t} = 0, \tag{29}$$

which we solve by substituting for $\boldsymbol{\tau}^{t+\Delta t^e}$ to give a set of equations for velocity unknowns.
In choosing a material timescale (Δt^e) independent of the numerical advection timestep (Δt), 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:

$$\hat{\boldsymbol{\tau}}^{t+\Delta t} = \left(1 - \frac{\Delta t}{\Delta t^e}\right)\boldsymbol{\tau}^t + \frac{\Delta t}{\Delta t^e}\boldsymbol{\tau}^{t+\Delta t^e} \tag{30}$$

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

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 (Ben-Zion and Sammis, 2001 [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 (21) by introducing a Prandtl-Reuss flow rule for the plastic part of the stretching:

$$\frac{\overset{\triangledown}{\boldsymbol{\tau}}}{2\mu} + \frac{\boldsymbol{\tau}}{2\eta} + \lambda\frac{\boldsymbol{\tau}}{2\,|\boldsymbol{\tau}|} = \hat{\boldsymbol{D}}_e + \hat{\boldsymbol{D}}_v + \hat{\boldsymbol{D}}_p = \hat{\boldsymbol{D}} \tag{31}$$

where λ 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 (31), we again express the Jaumann stress rate in first order difference form (using the Lagrangian particle reference frame):

$$\boldsymbol{\tau}^{t+\Delta t^e}\left[\frac{1}{2\mu\Delta t^e} + \frac{1}{2\eta} + \frac{\lambda}{2\,|\tau|}+\right] = \hat{\boldsymbol{D}}^{t+\Delta t^e} + \frac{1}{2\mu\Delta t^e}\boldsymbol{\tau}^t + \frac{1}{2\mu}(\mathbf{W}^t\boldsymbol{\tau}^t - \boldsymbol{\tau}^t\mathbf{W}^t) \tag{32}$$

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

$$\boldsymbol{\tau}^{t+\Delta t^e} = \eta' \left[2\hat{\boldsymbol{D}}^{t+\Delta t^e} + \frac{1}{\mu \Delta t^e}\boldsymbol{\tau}^t + \frac{1}{\mu}(\mathbf{W}^t\boldsymbol{\tau}^t - \boldsymbol{\tau}^t\mathbf{W}^t) \right] \tag{33}$$

using an effective viscosity, η' given by

$$\eta' = \frac{\eta \tau_{\text{yield}} \mu \Delta t^e}{\eta \tau_{\text{yield}} + \tau_{\text{yield}} \mu \Delta t^e + \lambda \eta \mu \Delta t^e} \tag{34}$$

We determine λ by equating the value of $|\boldsymbol{\tau}^{t+\Delta t^e}|$ with the yield stress in (33). Alternatively, in this particular case, we can obtain η' directly as

$$\eta' = \tau_{\text{yield}} / \left| \hat{\mathbf{D}}_{\text{eff}} \right| \tag{35}$$

where

$$\hat{\mathbf{D}}_{\text{eff}} = 2\hat{\boldsymbol{D}}^{t+\Delta t^e} + \frac{1}{\mu \Delta t^e}\boldsymbol{\tau}^t + \frac{1}{\mu}(\mathbf{W}^t\boldsymbol{\tau}^t - \boldsymbol{\tau}^t\mathbf{W}^t) \tag{36}$$

and $|\boldsymbol{D}| = (2D_{ij}D_{ij})^{1/2}$.
The value of λ or η' 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.

Applications

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. Figure 8 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 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).
Figure 9 shows the progress of the experiment. Initially, deformation was uniform, resulting in gradual stretching of the sample ($t \leq 0.180$). 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 focussed all subsequent deformation until the sample failed entirely ($t \geq 0.2404$). The frames are not uniformly spaced in time since the post-failure behaviour occurred on a much shorter timescale than the gradual loading. Note, for example, that the necking 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 Figure 10a, 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 3×10^5. 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 Figure 10b. Before yielding, the displacement of each sample point increased monotonically. Once yielding occured, 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 would need to complement the yield criterion on the deviatoric stress with a suitable tension cutoff condition.

Folding of rock strata

Geologists are intrigued by the formation of folds in sedimentary and metamorphic rocks on a large range of scales. Different conditions are necessary to create such buckling but it is often associated with tectonic compression. Other parameters are rock mineralogy, temperature, pressure, water content etc. We are predominantly interested in studying buckling as a function of elastic and viscous properties of the different layers. As a first step to show the capability of this new formulation to handle large deformation, we present in this analysis a folding problem of a competent layer embedded into two weaker purely viscous semi- infinite layers.
The initial geometry (Fig. 11a) is perfectly linear which means that there is no trigger for the instability to start except numerical noise. An unit inward velocity boundary condition is set up on the right side of the mesh. All the others boundaries are free-slip. We model the strong layer as viscoelastic with a yield criteria to follow qualitatively the evolution of the folding and the failure. Figure 11b shows the evolution during the homogeneous shortening phase where numerical noise triggers the instability and internal elastic stresses are built up. Once the instability is triggered the folding of the competent layer starts (Fig. 11c) and show a wavelength which is in accordance to the Biot theory (1965) [4]. If we keep compressing then bending stresses are built up in the beam with the largest tensile stress in the outer part where the curvature is the highest. Then due to tensile failure the beam breaks into a number of pieces (Fig. 11d) and the internal elastic stresses are released which straighten the layer. The final numerical plot is to be compared with what geologists can observe in situ (Fig. 11e). Even if the reason of such a microstructure is not perfectly understood (brittle failure, perpendicular shearing, anisotropy ... etc) the model presented in this paper looks reasonably accurate.

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:

$$\frac{\partial T}{\partial t} + v_i \frac{\partial T}{\partial x_i} = \kappa \frac{\partial^2 T}{\partial x_j^2}, \tag{37}$$

where x_i are the coordinates, v_i is the velocity, T the temperature and κ is the thermal diffusivity.
This thermomechanical formulation is applied to the convection solution used as a benchmark by Blankenbach and coworkers in [[5]]
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, ρ is the reference density, ΔT is the temperature contrast across the layer, and η 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 β.

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.12 (a)), and let them settle (Fig.12 (b)). When the crystals had sunk to the bottom of the layer (Fig.12 (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.12 (c)) β 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 Fig.12 (d) and (e)). In the end particles reached a new equilibrium (Fig.12 (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

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 over other methods 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. Finally, while it is conceptually simple to retain the mesh, and computationally convenient, the storage requirement is very high.

References

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

[2] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free galerkin methods. *International Journal of Numerical Methods in Engineering*, 37:229–256, 1994.

[3] Y. Ben-Zion and C. Sammis. Characterization of fault zones. *Pure & Applied Geophysics*, 2001. Submitted.

[4] M.A. Biot. *Mechanics of incremental deformations*. John Wiley & Sons, New-York, 1965.

[5] B. Blankenbach, F. Busse, U. Christensen, L. Crespes, D. Gunkel, U. Hansen, H. Harder, G. Jarvis, K. Koch, G. Marquart, P. Moore, D. andOlson, H. Schmeling, and T. Schnaubelt. A benchmark comparison for mantle convection codes. *Geophysical Journal International*, 98:23–38, 1989.

[6] J. Braun and M. Sambridge. Dynamical lagrangian remeshing (dlr): A new algortihm for solving large strain deformation problems and its application to fault-propagation folding. *Earth and Planetary Sciences Letters*, 124:211–220, 1994.

[7] J. Braun and M. Sambridge. A numerical method for solving partial differential equations on highly irregular evolving grids. *Nature*, 379:655–660, 1995.

[8] J. Cahouet and J.-P. Chabard. Some fast 3d finite element solvers for the generalized stokes' problem. *International Journal for Numerical Methods in Fluids*, 8:869–895, 1988.

[9] A. Heurta and W. K. Liu. Viscous flow with large free surface motion. *Computer Methods in Applied Mechanics and Engineering*, 69:277–324, 1987.

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

[11] G. R. Liu and Y. T. Gu. A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering*, 50:937–951, 2001.

[12] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992.

[13] L. N. Moresi and V. S. Slolomatov. Numerical investigation of 2d convection with extremely large viscosity variations. *Physics of Fluids*, 7(9):2154–2162, 1995.

[14] L. N. Moresi and V. S. Slolomatov. Mantle convection with a brittle lithosphere: thoughts on the global tectonic styles of the earth and venus. *Geophysicical Journal International*, 133:669–682, 1998.

[15] H.-B. Mühlhaus, L. Moresi, F. Dufour, and B. Hobbs. The interplay of material and geometric instabilities in large deformations of viscous rock. In B. Karihaloo, editor, *Proc. of the IUTAM Symposium on analytical and computational fracture mechanics of non-homogeneous materials*, page in press. Kluwer, 2001.

[16] H. Sakaguchi and H.-B. Mühlhaus. Mesh free modelling of failure and localization in brittle materials. In *Deformation and Progressive Failure in Geomechanics*, pages 15–21. Pergamon, 1997.

[17] D. Sulsky, Z. Chen, and H. L. Schreyer. A particle method for history-dependent materials. *Computational Methods in Applied Mechanics and Engineering*, 118:179–196, 1994.

[18] D. Sulsky and H. Schreyer. Antisymmetric form of the material point method with applications to upsetting and taylor impact problems. *Computational Methods in Applied Mechanical Engineering*, 139:409–429, 1996.

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

[20] C. Wijns, F. Boschetti, and L. Moresi. Inversion in geology by interactive evolutionary computation. *J. Struct. Geol.*, page submitted, 2001.

[21] C. Zhao, B. E. Hobbs, H.-B. Mühlhaus, and A. Ord. A consistent point-searching algorithm for solution interpolation in unstructured meshes consisting of 4-node bilinear quadrilateral elements. *International Journal for Numerical Methods in Engineering*, 45:1509–1526, 1999.

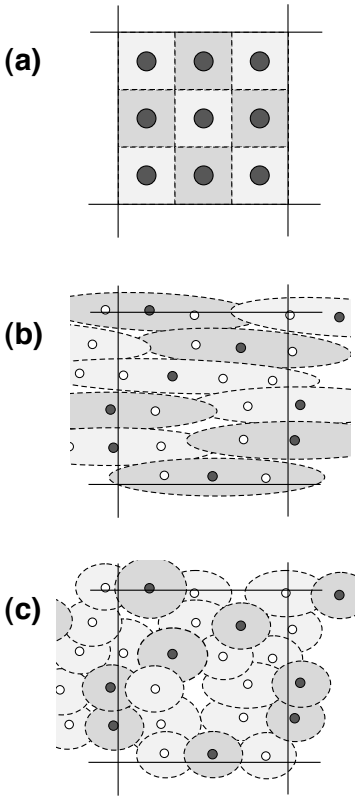

Figure 6: Integration schemes become more complicated when large material strains produce elongated "local volumes" for particles. (a) Initial configuration, (b) flow near a stagnation point elongates domains: solid circles are original particle points, open circles indicate new locations within the 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.

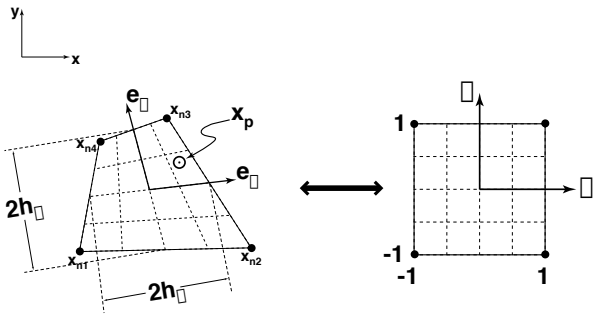

Figure 7: Coordinate systems in global mesh, distorted element, and master element reference frames

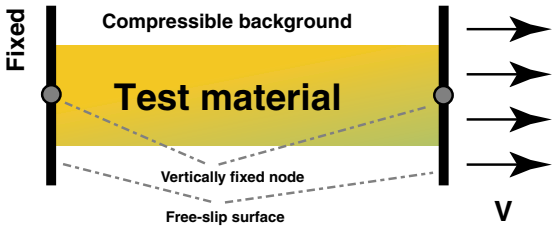

Figure 8: Geometry for simulation of the extension of a viscoelastic bar with a yield stress.

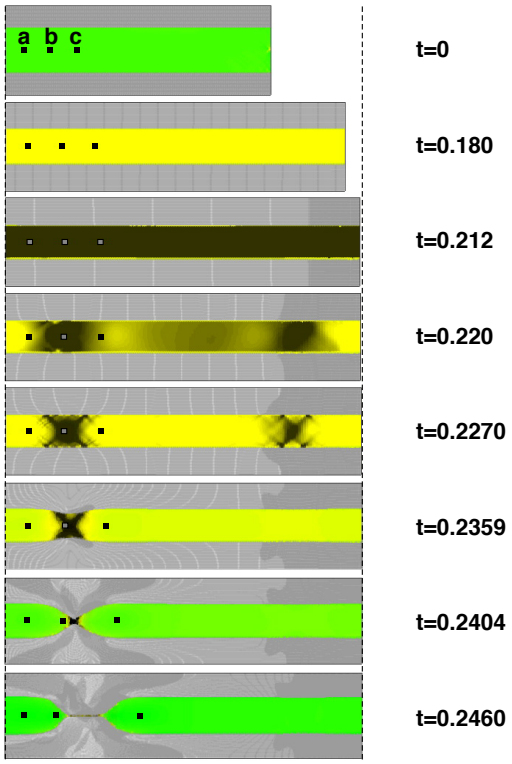

Figure 9: Simulation of the extension of a viscoelastic bar with yield stress. Black shading indicates regions deforming at yield. Embedded marker points which follow the material deformation are indicated by a,b,c.

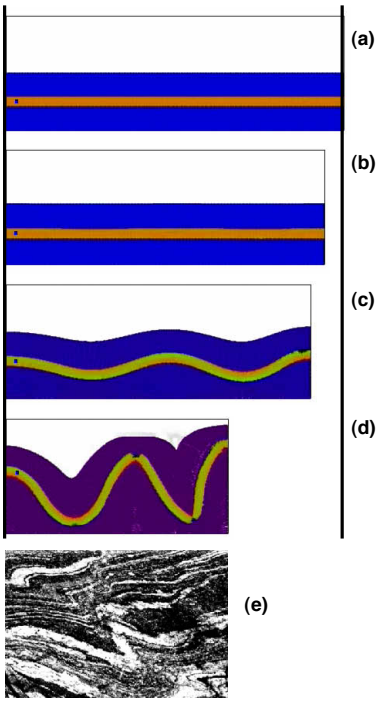

Figure 11: (a) Initial configuration of the layers, (b) homogeneous shortening, (c) viscous-like folding, (d) failure of the layer and (e) Quartzo-feldspathic layers (light colors) defining asymmetric folds in Archean migmatitic gneiss, Simo, Northern Finland.

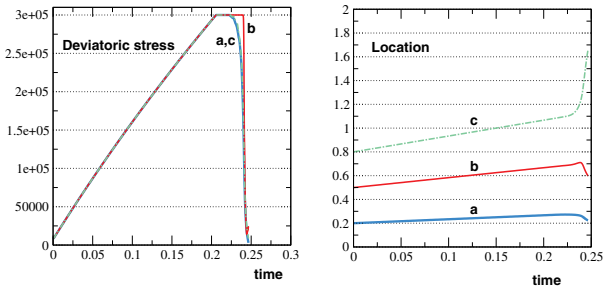

Figure 10: Stress, and displacement at sample points a,b,c as a function of time for the extension experiment of figure 9

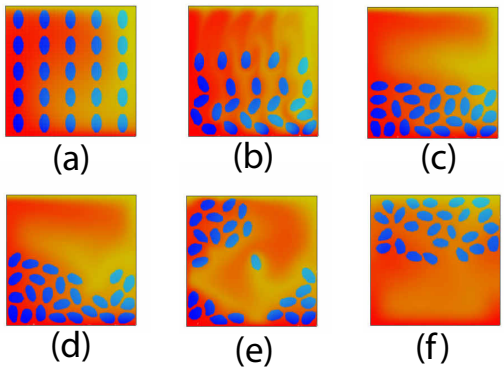

Figure 12: Numerical simulation of the settling of dense crystals and the uplift of light crystals in a convecting viscous fluid. (a) Initial setup, (b) settled crystals, (c) time at which density is decreased, (d) and (e) light crystals rising, (f) steady-state