Glacial isostatic adjustment on 3-D Earth models:
a finite-volume formulation

Konstantin Latychev,¹ Jerry X. Mitrovica,¹ Jeroen Tromp,² Mark E. Tamisiea,³
Dimitri Komatitsch²,⁴ and Christina C. Christara⁵

¹Department of Physics, University of Toronto, Toronto, ON, Canada, MSS 1A7. E-mail: latychev@physics.utoronto.ca
²Seismological Laboratory, California Institute of Technology, Pasadena, CA 91125, USA
³Harvard-Smithsonian Center for Astrophysics (MS42), Cambridge, MA 02138, USA
⁴Geophysical Imaging Laboratory, Université de Pau et des Pays de l’Adour, 64013 Pau Cedex, France
⁵Department of Computer Sciences, University of Toronto, Toronto, ON, Canada, MSS 2E4

Accepted 2004 November 24. Received 2004 August 26; in original form 2003 August 28

SUMMARY
We describe and present results from a finite-volume (FV) parallel computer code for forward modelling the Maxwell viscoelastic response of a 3-D, self-gravitating, elastically compressible Earth to an arbitrary surface load. We implement a conservative, control volume discretization of the governing equations using a tetrahedral grid in Cartesian geometry and a low-order, linear interpolation. The basic starting grid honours all major radial discontinuities in the Preliminary Reference Earth Model (PREM), and the models are permitted arbitrary spatial variations in viscosity and elastic parameters. These variations may be either continuous or discontinuous at a set of grid nodes forming a 3-D surface within the (regional or global) modelling domain. In the second part of the paper, we adopt the FV methodology and a spherically symmetric Earth model to generate a suite of predictions sampling a broad class of glacial isostatic adjustment (GIA) data types (3-D crustal motions, long-wavelength gravity anomalies). These calculations, based on either a simple disc load history or a global Late Pleistocene ice load reconstruction (ICE-3G), are benchmarked against predictions generated using the traditional normal-mode approach to GIA. The detailed comparison provides a guide for future analyses (e.g. what grid resolution is required to obtain a specific accuracy?) and it indicates that discrepancies in predictions of 3-D crustal velocities less than 0.1 mm yr⁻¹ are generally obtainable for global grids with ~3 × 10⁶ nodes; however, grids of higher resolution are required to predict large-amplitude (>1 cm yr⁻¹) radial velocities in zones of peak post-glacial uplift (e.g. James bay) to the same level of absolute accuracy. We conclude the paper with a first application of the new formulation to a 3-D problem. Specifically, we consider the impact of mantle viscosity heterogeneity on predictions of present-day 3-D crustal motions in North America. In these tests, the lateral viscosity variation is constructed, with suitable scaling, from tomographic images of seismic S-wave heterogeneity, and it is characterized by approximately 2 orders of magnitude (peak-to-peak) lateral variations within the lower mantle and 1 order of magnitude variations in the bulk of the upper mantle (below the asthenosphere). We find that the introduction of 3-D viscosity structure has a profound impact on horizontal velocities; indeed, the magnitude of the perturbation (of order 1 mm yr⁻¹) is as large as the prediction generated from the underlying (1-D) radial reference model and it far exceeds observational uncertainties currently being obtained from space-geodetic surveying. The relative impact of lateral viscosity variations on predicted radial motions is significantly smaller.

Key words: crustal motions, finite volumes, glacial isostatic adjustment, 3-D viscoelastic Earth models.

1 INTRODUCTION
The Pleistocene period was marked by a long series of ice age cycles characterized by the growth and ablation of massive, high-latitude continental ice complexes. During the last glacial maximum (LGM), approximately 20 000 BP, grounded ice sheets covered most of Canada and the northeastern USA, Scandinavia, northern Britain, Siberia, the Arctic seas north of Eurasia, Chile, the
Antarctic and Greenland (Denton & Hughes 1981). The excess volume of ice at the LGM was sufficient to lower global sea levels by \( \sim 130 \text{ m} \) (Yokoyama et al. 2000) and the subsequent deglaciation event, which ended \( \sim 5000 \text{ BP} \), initiated a period of dramatic disequilibrium and adjustment in the Earth system that persists to the present day.

The analysis of data reflecting the so-called glacial isostatic adjustment (GIA) of the Earth has a long history within the geophysical sciences. In the earliest research, site-specific and regional sea level observations were used to constrain the viscosity structure of the Earth (e.g. Haskell 1935; McConnell 1968; Cathles 1975; Peltier & Andrews 1976) and/or the space–time history of ice cover since the LGM (e.g. Peltier & Andrews 1976; Clark 1980; Wu & Peltier 1983). These applications also commonly incorporated anomalies in the regional gravity field (e.g. Walcott 1973; Wu & Peltier 1983) and rotational state of the Earth (e.g. O’Connell 1971; Nakiboglu & Lambeck 1980; Sabadini & Peltier 1981; Sabadini et al. 1982; Yuen et al. 1982; Wu & Peltier 1984). Over the last decade, the GIA data set applied to these classic applications has expanded to include space–geodetic measurements, using very-long-baseline interferometry (e.g. James & Lambert 1993; Mitrovica et al. 1993), the global Positioning System (GPS; e.g. Johansson et al. 2002; Milne et al. 2001) and a suite of satellite-derived long-wavelength gravity harmonics (e.g. Vermeersen et al. 1998).

During the last decade or so, the spectrum of GIA applications has broadened significantly and this has led to renewed interest in the study of ice age dynamics. These applications include, for example:

1. estimates of ongoing global and regional sea level rise, derived by correcting tide gauge records of present-day sea level change for the contaminating effects of GIA (e.g. Peltier & Tushingham 1989; Douglas 1991; Davis & Mitrovica 1996; Milne et al. 2001; Mitrovica et al. 2001);
2. constraints on excess ice volume at the LGM as input into global climate models (e.g. Peltier 1994; Yokoyama et al. 2000; Milne et al. 2002);
3. study of ancient land bridge development and destruction and the impact of this evolution on human migration routes (e.g. Lambeck 1996a,b);
4. the connection between deglaciation-induced lithospheric stress fields and seismicity (e.g. Wu 1997; Wu et al. 1999); and
5. long-term perturbations in the orbital parameters of the Earth driven by GIA and the signature of these effects on palaeoclimate observations were used to constrain the viscosity structure of the Earth.

The Peltier–Wu theory expresses the deformational and gravitational impulse response of a spherically symmetric, self-gravitating, Maxwell viscoelastic Earth model, at each spherical harmonic degree, as a combination of an instantaneous elastic response and a set of viscoelastic modes of pure exponential decay. The decay times of the individual modes are associated with eigenvalues of the homogeneous problem (Peltier 1974). The modal amplitudes are ultimately derivable by assuming a set of simple poles in the Laplace transform domain and then applying Cauchy’s residue theorem (Wu 1978). Physically, each of the viscoelastic modes originates from some rheological discontinuity in the Earth models, for example the density jumps at the surface, 670 km depth and the core–mantle boundary (CMB), or the viscosity interface at the base of the elastic lithosphere. While recent work has further elucidated the potentially complex transform domain structure of the impulse response (e.g. Fang & Hager 1994; Han & Wahr 1995), including the possibility of continuous singularities, quasi-analytical calculations have demonstrated that predictions based on the Peltier–Wu method are robust (e.g. Boschi et al. 1999).

Following Parsons (1972), Peltier (1976) derived kernels relating arbitrary perturbations in the radial viscosity profile to consequent perturbations in the modal decay times (eigenfrequencies). This derivation was completed only recently by Tromp & Mitrovica (1999b), who presented analogous kernels for the perturbation in the modal amplitudes. Tromp & Mitrovica (1999b) furthermore demonstrated that these amplitudes were weighted versions of the eigenfunctions of the viscoelastic Earth model and derived a normal-mode methodology, independent of residue theory, for computing the weighting associated with an impulse loading. More importantly, for the present study, Tromp & Mitrovica (1999a, 2000) derived a perturbation theory for computing the effect on the normal-mode response of general, 3-D perturbations in viscoelastic structure. With suitable parametrization (for details, see Tromp & Mitrovica 2000) these expressions remain reasonably accurate for a 1 order of magnitude variation in viscosity. Unfortunately, lateral heterogeneities in Earth structure must be more significant than the range encompassed by this perturbation theory. Continents are characterized by old, thick lithosphere, while oceanic regions roughly satisfy the well-known square-root age dependence for the thickness. At plate margins, the elastic thickness locally vanishes. Furthermore, plate tectonic motions reflect a slow thermal convection within the Earth, implying the existence of large lateral variations in temperature, and therefore density and viscosity. Indeed, these variations have been increasingly well imaged by seismic tomography (e.g. Ekström & Dziewonski 1998).

In this article, we present in detail a finite-volume (FV) method for computing the response of a 3-D Maxwell viscoelastic model to an arbitrary surface mass loading. The primary function of this numerical tool will be to help establish a new generation of GIA predictions for the suite of geophysical applications discussed above. This goal is reflected in the example calculations described herein. However, the code may also be applied directly to problems in tectonic deformations on 3-D Earth models or, with minor changes, to the study of coseismic and post-seismic deformations. Several groups have recently described numerical schemes for tackling the GIA

© 2005 RAS, GJI, 161, 421–444
problem on 3-D Earth models (Wu & van der Wal 2003; Zhong et al. 2003). Our method is entirely independent of these approaches and also involves an improvement in the complexity of the adopted Earth model. Specifically, the methods of both Wu & van der Wal (2003) and Zhong et al. (2003) involve fully incompressible Earth models. This simplification introduces errors in the prediction of 3-D crustal motions as a result of GIA (Mitrovica et al. 1994a,b) that may, for example, complicate the analysis of space-geodetic observations.

The FV methodology presented here is ultimately developed for a distributed (parallel) computer environment, e.g. a PC cluster, although it is possible to run the code in the serial mode, i.e. on a single desktop computer. We solve the coupled, time-domain system of equations governing loading-induced displacement and gravity perturbations on a self-gravitating, Maxwell viscoelastic, aspherical Earth model in either a global or regional geometry. Following Wu & Peltier (1982), our governing equations treat a viscoelastic material that is incompressible in the long-term (so-called fluid) limit, but that is compressible in the short-term (elastic) limit.

The paper is separated into three main parts. First, we outline the new FV methodology. This development involves sections in the main text dealing with the governing equations, the FV method, sparse matrix solver and key issues related to parallelization. Technical details associated with grid generation, domain decomposition, linear interpolation, etc., are also required for implementation, but are not crucial for a basic understanding of the FV formulation; accordingly, these are compiled within a technical Appendix. Next, we outline a systematic benchmark comparison of predictions, on a spherically symmetric, non-rotating Earth model, of a suite of GIA data sets generated using the FV methodology and the traditional normal-mode approach. These comparisons involve both a simplified, single disc surface mass load, as well as a more complex, global model of the Late Pleistocene ice history. Our focus, in these tests, will be to verify the accuracy of the FV and the traditional normal-mode approach. These comparisons in-
where \( k = \mu \Delta t / \eta \). Note that \( k \) is the ratio of the time step to the Maxwell time, \( t_{\text{Maxwell}} = \eta / \mu \), which is a fundamental parameter describing the time response of a viscoelastic system. For reasons of numerical stability, one has to ensure \( k \leq 1 \) or, equivalently, \( \Delta t \leq t_{\text{Maxwell}} \).

Using representation (7) and the mass conservation eq. (3), we can now rewrite the momentum eq. (1) as

\[
\nabla \cdot \tau = - \nabla (\rho \nabla \Phi) + \rho \nabla \Phi_{1.0} + (\nabla \Phi) \cdot (\rho \nabla s) = - \nabla \cdot \xi \nabla \xi - 1,
\]

(9)

whose right-hand side (RHS) has become a function of the previous deformation history. Substituting the RHS of eq. (3) for \( \rho^0 \), we rewrite eq. (2) in the following form:

\[
\nabla \cdot (\nabla \Phi_{1.0} + 4\pi G \rho^0 s) = 0,
\]

(10)

to be solved for \( \{s, \Phi_{1.0}\} \).

The governing equations are solved as a boundary value problem within a finite modelling domain \( \Omega \) enclosed by a surface \( S \). Therefore, appropriate boundary conditions are required for both the displacement and the perturbed gravitational potential or their derivatives on \( S \) as well as on all internal interfaces of discontinuities in the material properties.

Assuming that the mass of the core remains constant during glacial cycling, Tromp & Mitrovica (1999a) derive the following boundary condition at the CMB:

\[
\hat{\nu} \cdot \mathbf{T} = \rho_0 (s^+ \cdot \nabla \Phi^0 + \Phi^1) \hat{\nu},
\]

(11)

where the variables superscripted with the plus sign are evaluated on the mantle side of the CMB, \( \rho_0 \) is the density of the liquid core just below the CMB and \( \hat{\nu} \) is a unit normal to the discontinuity. The direction of \( \hat{\nu} \) is chosen such that, if a discontinuity belongs to the domain boundary \( S \), it points outside the domain.

Generally, perturbations in density in the entire core would affect \( \Phi^1 \). Moreover, because the outer core is liquid, the notion of displacement does not exist there. The governing equations in the outer core would need to be formulated for the fluid pressure as in, for example, Komatitsch & Tromp (2002). However, following Zhong et al. (2003), the core is treated as an incompressible fluid of uniform density. This simplification allows us to restrict the simulation domain to the mantle only, observing that in this case the only contribution of the core to the perturbed gravitational potential \( \Phi^1 \) is the result of an apparent surface mass density, \( \sigma_{\text{CMB}} \). The latter may be associated with a change in the shape of the core and is given by

\[
\sigma_{\text{CMB}} = - \rho_0 s^+ \cdot \hat{\nu}.
\]

(12)

The external loads resulting from the ice and melt water are treated as surface mass sources, whose surface mass density \( \sigma \) is an arbitrary function of both time and position. If the surface is not loaded, the normal stress becomes zero. The corresponding Neumann boundary condition reads (Wu & Peltier 1982)

\[
\{ \mathbf{\nu} \cdot \mathbf{T} \} = - \sigma (\nabla \Phi^0) \mathbf{\nu}. \]

(13)

The gravity field exists in all of space and evaluation of the perturbed gravitational potential is a computationally challenging problem. There are two natural options for treating this problem: either extend the domain substantially outside the Earth or compute the surface values of the perturbed gravitational potential, \( \Phi^1_S = \Phi^1(r_S) \), directly from first principles. We take the latter approach and distinguish the following three contributions:

\[
\Phi^1_S = - G \int \frac{\rho^0(r)(s(r) \cdot (r_S - r))}{|r_S - r|^3} d\Omega - G \int r_S = r \frac{\sigma(r)}{|r_S - r|} dS - G \int \frac{\sigma_{\text{CMB}}(r)}{|r_S - r|} dS,
\]

(14)

where \( R \) is the radius of the Earth. The first term in eq. (14) is the result of the displacement from equilibrium at any point \( r \) within the domain. It is derived in Dahlen & Tromp (1998, p. 73) as a single volumetric integral. The volume is restricted to \( \Omega \), where we have non-trivial displacements \( s \). The second term is the direct impact of the load itself, which can be written by inspection as a standard solution to Poisson’s eq. (2). The integration is carried out over all of \( S \), but the non-trivial contributions come only from loaded patches on the top surface at \( r_S = R \). For definiteness, both radius vectors involved in the integration, \( r \) and \( r_S \), originate from the centre of the Earth. Finally, the third term is the contribution resulting from CMB deformation.

We allow discontinuities in material properties, such as \( \kappa, \mu, \eta \) and \( \rho \), on a set of prescribed surfaces within \( \Omega \). A simple example is a radial discontinuity in the Preliminary Reference Earth Model (PREM; Dziewonski & Anderson 1981), but there are no restrictions on the shape or position of such an interface. Following Tromp & Mitrovica (1999a), we require that the displacement, perturbed gravitational potential, both the ambient gravitational potential and the normal component of its gradient, and the normal component of stress all be continuous across a discontinuity. These conditions can be written formally as

\[
[\mathbf{s}]^+ = 0, \quad [\Phi^1]^+ = 0, \quad [\Phi^0]^+ = 0, \quad [\mathbf{\nu} \cdot \nabla \Phi^0]^+ = 0, \quad [\mathbf{\nu} \cdot \mathbf{T}]^+ = 0.
\]

(15)

The notation \([\cdot]^+\) denotes the jump in the enclosed quantity when going from the + side to the − side of a discontinuity. Integration of eq. (2) around a pillbox centred on a density discontinuity readily gives

\[
[\mathbf{\nu} \cdot \nabla \Phi^1]^+ = - 4\pi G(\mathbf{s} | [\rho^0]^+).
\]

(16)

where we have used the fact that \( s \) is continuous.

We treat the governing eqs (9) and (10) numerically as a single vector equation with four unknowns at each solution point, comprising the three independent displacement components and a scalar perturbed gravitational potential. Note that there is a time lag of order \( \Delta t \) between the displacement components and the perturbed gravitational potential, because evaluation of the boundary condition at time step \( n \) requires a knowledge of \( s^n \). In practice, two iterations per time step yields good results.

The time-marching procedure is as follows: we solve the system of eqs (9) and (10) at time step \( n \) for \( \{s^n, \Phi_{1,n}\} \), ensure mass conservation at the CMB, calculate the perturbed gravitational potential on the boundary as per eq. (14), solve the governing equations again, update the viscous memory term \( \xi \mathbf{\xi} = \xi^0 (\xi^{-1} \mathbf{\xi} - \mathbf{\xi}^0) \) according to expression (8) and the perturbed gravitational potential on the boundary according to expression (14) (second iteration), and then move to the next time step \( n + 1 \).

Although the simulations described below are based on global grids, we have also implemented calculations generated with regional grids in which side boundaries are introduced in the model domain. For the latter case, which is clearly useful when dealing with localized loads, we adopt a boundary condition of zero displacement on the side boundary connecting the surface and CMB of the Earth.

© 2005 RAS, GJI, 161, 421–444
The details of the grid generation scheme we ultimately adopted are shown in Fig. 1(a). The vertices are the grid nodes. Each grid node is defined by its four vertices \( \{ v^1, v^2, v^3, v^4 \} \), as shown. (b) The corresponding control volume constructed for node \( v^1 \), shown in (a). For the selected element, we label the vertices of the control volume that belong to the staggered grid. They are the mid-edges \( \{ e^{12}, e^{13}, e^{14} \} \) and the mid-faces \( \{ f^{123}, f^{134}, f^{124} \} \) of the original grid, as detailed further in Fig. 2. Point \( e^i \) is the centre of element \( \{ v^i, v^2, v^3, v^4 \} \).

### 2.2 Control volume-element method on tetrahedral grids

Both governing partial differential equations are second-order elliptic equations with respect to the coordinates. They are solved numerically on a tetrahedral grid, constructed to cover the entire simulation domain \( \Omega \). We favour tetrahedra because of their simplicity, flexibility and, not the least, because of the availability of a spherical tetrahedral grid generator in the early stages of the project. The details of the grid generation scheme we ultimately adopted are given in Appendix A1.

We adhere to a Cartesian coordinate system throughout, which is defined by the three mutually orthogonal unit vectors \( \{ \hat{x}, \hat{y}, \hat{z} \} \) in the \( x, y, z \) direction, respectively. Our basic grid element, the tetrahedron, is defined by its four vertices \( \{ v^1, v^2, v^3, v^4 \} \) as shown in Fig. 1(a). The vertices are the grid nodes. Each grid node is associated with four unknowns, \( \{ U, V, W, \Phi \} \), where \( U = (\mathbf{s} \cdot \hat{n}) \), \( V = (\mathbf{s} \cdot \hat{n}') \) and \( W = (\mathbf{s} \cdot \hat{n}^2) \). All the material properties, i.e. the elastic moduli, viscosity and density, are specified (sampled) at the grid nodes only.

If a grid node lies on a discontinuity, we allow multiple nodal values for \( \{ \kappa, \mu, \eta, \rho_0 \} \). In extreme cases, there may be as many such values as there are the elements sharing this particular node. However, for a given element, the four material properties for each of the four nodes are single valued and no discontinuities are allowed inside the element.

It is fundamental in finite-element (FE) methods (e.g. Zienkiewicz 1977; Huyakorn & Pinder 1983) that one obtains the weak form of the governing equations. This is achieved by multiplying the terms of the original equation by a test function and integrating by parts, using the Gauss theorem. The benefits of this action are two-fold. The order of the highest derivative to be discretized is reduced by 1 as a result of the integration and, in most practical cases, the boundary conditions can be readily incorporated as surface integrals. The volume used for the integration involves most commonly all the elements that share the node. This is illustrated in Fig. 1(a) for node \( v^1 \). For clarity, we highlight one tetrahedron and show its edges inside the shaded volumetric body as dashed lines.

The control volume method (e.g. Patankar 1980) is a strategy that can work with either finite-difference (FD; e.g. Nogotov 1978; Shashkov 1995) or FE (e.g. Baliga & Patankar 1980) techniques. It requires that the relevant conservation laws hold on average over a control volume, constructed around a grid node of interest. As a rule, the vertices of the faces, defining the boundary of the control volume, belong to the dual, staggered grid. The latter is usually constructed by taking the Voronoi volume, bounded by the perpendicular bisectors to the faces of the original grid. This conservative approach has been successfully applied to various problems, such as diffusion (Shashkov & Steinberg 1996), electromagnetics (e.g. Hyman & Shashkov 1996), porous medium convection and compliance modelling (e.g. Latychev 2000), etc. There exist further modifications of the control volume method that enforce conservation laws on the user-defined cells (e.g. Cai et al. 1997), but we follow the node-centred approach here.

We use linear shape functions, i.e. any field or property in the interior of the element is linearly interpolated in terms of its anchor values at the element vertices. Because the leading terms and the RHS of eq. (9) and the entire eq. (10) are already in divergence form, we use the simplest unit test functions for Gaussian integration with respect to the control volume. Generally, four linearly independent scalar equations are required at each gridpoint, one for each solution component. The first three are obtained by taking the dot product of eq. (9) with \( \hat{n}^1, \hat{n}^2 \) and \( \hat{n}^3 \), respectively, and integrating against the control volume, while the forth one is simply eq. (10) integrated against the control volume.

Let us now construct a control volume. For a grid node of interest, e.g. \( v^1 \), we select all the tetrahedral elements that contain that node as one of their four vertices. Now we focus on a single element \( \{ v^1, v^2, v^3, v^4 \} \), whose volume is \( \Omega^e \). Within each such element, as shown in Fig. 2(a), we identify the mid-edges \( \{ e^{12}, e^{13}, e^{14} \} \), which are located halfway between \( v^1 \) and the other three vertices, mid-faces \( \{ f^{123}, f^{134}, f^{124} \} \), which are the centres of gravity of the three triangular faces of the element, intersecting at node \( v^1 \), and the centre of the element \( e^i \). Introducing the Cartesian coordinates of the nodes as \( \{ x^i, y^i, z^i \} \), where \( i = \{ 1, 2, 3, 4 \} \), we can write, for...
example, for the $x$ coordinates of the, staggered nodes
\[ e^{ij}(x) = \frac{1}{2}(x^i + x^j), \quad f^{ij}(x) = \frac{1}{3}(x^i + x^j + x^k), \]
\[ c^e(x) = \frac{1}{4} \sum_{i=1}^{4} x^i, \]
where $i, j, l = [1, 2, 3, 4]$. Similar expressions can easily be written for the other two coordinates by substituting $y$ and $z$ for $x$ in the last formula. Note the intentional use of superscripts to enumerate various nodal values. For now, subscripts are reserved strictly for the last formula. Note the intentional use of superscripts to enumerate various nodal values. For now, subscripts are reserved strictly for the last formula.

Discontinuities are allowed across surfaces $S^{ij}$. A typical complete control volume for a selected grid node counts approximately 100–120 triangles on its outer surface. The resulting shape is usually quite complicated and may not be convex. We demonstrate one example in Fig. 1(b), which corresponds to the tetrahedral grid fragment centred around node $v^1$, as shown in Fig. 1(a).

It is important to realize that all possible neighbouring control volumes may only share the boundary $S^i$ located in the interior of the element. Therefore, the numerical continuity of gradients between neighbouring control volumes is automatically enforced. This is in contrast to a conventional FE discretization with linear shape functions, where all element surfaces may feature discontinuous derivatives.

Appendix A2 provides details on the numerical integration of the governing equations against control volumes. This operation yields a sparse matrix-linear system of the form
\[ A \cdot \psi = b \] (18)
for the vector of unknowns $\psi$ to be solved at each time step. Note that the system matrix $A$ is stationary (i.e. does not change in the course of simulations) and needs to be assembled only once. The RHS vector $b$ depends on the loads and viscous memory terms and, therefore, is time-dependent.

2.3 Sparse matrix solver

The development presented in Section 2.2 (see also Appendix A2) has led us to solving a sparse linear system, given by eq. (18), at each time step. The system matrix is non-symmetric because of the advection terms. The Generalized Minimal Residual (GMRES) method

Figure 2. The control volume geometry. (a) A tetrahedral element with vertices $\{v^1, v^2, v^3, v^4\}$ and centre $c^e$. For vertex $v^1$, we construct a control volume and show its intersection with the element, $\Omega^{cv}$, defined by $\{v^1, e^{123}, f^{123}, f^{134}, e^{134}, c^e\}$, where $e^{12}$, $e^{13}$ and $e^{14}$ are the mid-edges and $f^{123}$, $f^{134}$ and $f^{124}$ are the mid-faces of the original element. All vertices of $\Omega^{cv}$, except $v^1$, belong to the staggered grid, so that, for example, $e^{12}$ divides edge $\{v^1, v^2\}$ in half, while $f^{123}$ lies in the middle of the triangular face $\{v^1, v^2, v^3\}$, etc. The outer surface of $\Omega^{cv}$ is made up of two fully conforming cups with six triangles in each: $S^i = \bigcup_{l=1,6} S^{il, m}$, shown in (b), and $S^{ii} = \bigcup_{l=1,6} S^{ii, m}$, shown in (c).
of Saad & Schultz (1986) is particularly well suited to handle such matrices. GMRES belongs to a family of sparse iterative solvers generally termed Krylov subspace methods. One popular representative of this class of solvers is the Conjugate Gradient method for symmetric matrices. The basic idea of all these algorithms is to minimize a certain norm of the residual in an appropriately chosen subspace. We give only a very brief summary of GMRES in this section, referring the interested reader to detailed accounts on the subject, e.g. Khosla & Rubin (1981), van der Vorst (1981), Saad & Schultz (1986), Saad (1989) and Shadid & Timinario (1994).

If \( \psi^{(0)} \) is an initial approximation to the solution of eq. (18), we define a residual \( \mathbf{r}^{(0)} = b - \mathbf{A}\psi^{(0)} \) and the mth Krylov subspace

\[
\mathbf{K}_m := \text{span} \{ \mathbf{r}^{(0)}, \mathbf{A}\mathbf{r}^{(0)}, \ldots, \mathbf{A}^{m-1}\mathbf{r}^{(0)} \}.
\]

The GMRES approximation \( \psi^{(m)} \) is constructed so that \( \psi^{(m)} \in \mathbf{K}_m \) and

\[
\psi^{(m)} = \psi^{(0)} + \gamma_1 \vartheta^{(1)} + \ldots + \gamma_m \vartheta^{(m)},
\]

where \( \vartheta^{(1)}, \ldots, \vartheta^{(m)} \) form an orthonormal basis in \( \mathbf{K}_m \) and the coefficients \( \gamma_i \) are chosen to minimize the Euclidean norm \( \| \mathbf{b} - \mathbf{A}\psi^{(m)} \| \) of the residual among all vectors in \( \psi^{(0)} + \mathbf{K}_m \). If \( m \) is equal to the number of unknowns, the method should yield the exact solution in less than or equal to \( m \) steps. In practice, in order to reduce the storage requirement and the per iteration computational cost, the restarted version of GMRES is used, in which the number of steps is normally set between 5 and 20. If the last iterate \( \psi^{(m)} \) is still not satisfactory, it is copied into \( \psi^{(0)} \) and the process restarts. Thus, the restarted GMRES consists of two nested (inner and outer) iteration loops.

The convergence rate of iterative methods depends on the spectral properties of the system matrix, which may be improved by applying a preconditioner. That is, we pre-multiply the original matrix \( \mathbf{A} \) by another matrix, either from the left or from the right (or even both), attempting to make the original equation as diagonal as possible. We will then be solving a modified system, expressed as

\[
\mathbf{M}_1^{-1}\mathbf{A}\mathbf{M}_2^{-1}\chi = \mathbf{M}_1^{-1}\mathbf{b},
\]

where \( \mathbf{M}_1 \) is the left preconditioner, \( \mathbf{M}_2 \) the right preconditioner, the superscript \( -1 \) stands for the matrix inverse and \( \psi = \mathbf{M}_1^{-1}\chi \). Note that we can set, e.g. \( \mathbf{M}_1 = I \) or \( \mathbf{M}_2 = I \) at will. The incomplete lower-upper (ILU) preconditioner (e.g. van der Vorst 1981) makes a numerically robust pair with GMRES. It is known that a regular lower-upper (LU) decomposition leads to the exact solution (within machine precision) and is similar to Gaussian elimination with regard to computational cost, because the matrix associated with the complete LU is not sparse. The most common strategy in constructing the ILU decomposition is to follow the sparsity pattern of the system matrix in the preconditioning matrix, allowing a controlled amount of extra fill-in elements in each row. Large fill-ins result in a more accurate (but still approximate) inversion at a progressively higher computational cost.

In the course of the GMRES iterations, one has to perform several operations, of which the most time-consuming are: dot products to construct the orthonormal basis, matrix–vector products to control the residual and preconditioning operations. Note that in our case, the latter involve only a back-and-forward substitution at each iteration, which is equivalent to a matrix–vector product. The preconditioning matrices \( \mathbf{M} \) are to be set up only once, outside the iteration loop. This is because our grid is stationary and all the material properties are those at equilibrium, so that \( \mathbf{A} \) does not change with time.

We adopt the GMRES software package by Fraysse, Giraud & Gratton, available from http://www.cerfacs.fr/algor/. It is coded in reverse communication mode, which allows the user to perform the three operations mentioned in the previous paragraph outside the GMRES routine. This makes it extremely convenient for parallelization, which is the subject of the next section. Based on extensive numerical experimentation, we set \( m = 20 \), choose the right preconditioner only and select the iterative classical Gram–Schmidt orthogonalization out of the menu of four options to construct the orthonormal basis. The following convergence criterion is chosen to control the GMRES iterations:

\[
\frac{\| \mathbf{A}\chi^{(i)} - \mathbf{b} \|}{\| \mathbf{b} \|} \leq \varepsilon,
\]

where \( \chi^{(i)} \) is the last computed iterate and \( \varepsilon \) is the relative error, set to \( 10^{-6} \). The GMRES solver requires a real work array of length \( (m \times m) + [m \times (n_{\text{row}} + 5)] + [5 \times (n_{\text{row}} + 1)] \), where \( n_{\text{row}} \) is the row dimension of \( \mathbf{A} \). The number of the GMRES iterations is limited to 100.

The ILU routine is borrowed from the SPARSEKIT-II package (where it is called ilat), developed by Y. Saad and others (available from http://www-users.cs.umn.edu/~saad/). It uses a dual truncation mechanism to control the fill-in elements to allow either \( 2 \times n_{\text{max}} + 1 \) elements per row or to drop the element whose magnitude is less than \( \delta \times A_{ij} \) for row \( i \). One needs a real array of length \( n_{\text{row}} \times (2 \times n_{\text{max}} + 1) \) and one integer array of length \( 2 \times n_{\text{row}} \times (n_{\text{max}} + 1) \). We set empirically \( n_{\text{max}} = 50 \) and \( \delta = 0.001 \).

### 2.4 Parallel aspects

The FV method discussed in the previous section requires approximately \( \sim 800 \text{ MB} \) of random access memory (RAM) per \( 10^7 \) grid nodes in double precision arithmetic. The numerical resolution possible with this small number of nodes is inadequate for the GIA problem, particularly in the case of global simulations. However, even a modest grid refinement would move us quickly beyond the capacity of contemporary desktop hardware.

Beowulf PC clusters have recently become one of the preferred choices for large numerical simulations, provided that each of the CPUs of the cluster can work independently most of the time. We run our calculations on a 64-node cluster at the University of Toronto delivered in 2004 May. Each PC node has a dual Opteron processor, working at 1.6 GHz, equipped with 6 GB of RAM and connected via a Gigabit network. On the software side, the interprocessor communication is handled by the MPI library. We use the open source MPICH version of MPI distributed by Argonne National Laboratory.

The sparse matrix solver described in Section 2.3 requires that the user perform matrix–vector and dot products in the course of the GMRES iterations. In fact, these are the only operations that need to be parallelized explicitly, because of the reverse communication feature built into the GMRES package.

Each processor executes exactly the same set of instructions but on a different set of data. The global system matrix \( \mathbf{A} \) is large and must be distributed between the processors. The interprocessor communications are executed by sending and receiving messages. To handle this, we perform an equation-to-processor mapping at the pre-processing phase; that is, we assign certain rows of \( \mathbf{A} \) to a pre-specified CPU. We also identify nodes whose values are to be sent to or received from other CPUs. Appendix A3 explains how this is accomplished. Note that, for the sake of brevity, we will refer to nodal values, with the understanding that each grid node has either
three or four solution components. A computer node will be further referred to as a CPU to avoid confusion.

Our parallel grid generator divides the simulation domain into approximately equal, overlapping parts according to the desired number of CPUs, $n_p$. Each CPU deals only with its own subdomain. We require all subdomain nodes to be labelled as shown on the conceptual, 2-D example in Fig. 3. There are four types of nodes: external, overlapping, boundary and private (internal).

We need one layer of external nodes that belong to neighbouring subdomains but are connected to the nodes assigned to the current CPU. The external nodes are required to complete the discretization for the assigned nodes near the subdomain boundary. Matrix coefficients resulting from the external nodes must be stored in local memory, while intermediate solution values at these nodes must be communicated from all neighbouring CPUs as soon as they become available.

We generally allow one layer of overlap between neighbouring subdomains, which is known to reduce the number of global iterations (Bjorstad & Widlund 1989) and ensures that the same node in different subdomains has identical field values. The overlapping nodal values are exchanged between neighbouring CPUs to obtain an algebraic average, which is computed taking into account the valence (or weight) of the nodes.

The boundary nodes are connected to the overlapping nodes but are not connected to the external nodes. They are likely to make external nodes in the neighbouring subdomains, therefore their values may need to be communicated. The private nodes are those connected either to the boundary nodes or to other private nodes. Their valence can only be unity and they do not participate in the communications.

We now discuss the implementation of distributed matrix–vector and dot products. The system matrix $A$ is square in the serial case and rectangular in the parallel case because of the coefficients resulting from the external nodes. The parallel performance is usually improved if each CPU sorts its associated grid nodes such that the private nodes are listed first, while the external nodes are listed last. The local part of the system matrix will then have an arrow-head structure, with a square block of zeroes in the top-right quadrant, as shown in Fig. 4. The figure suggests a natural, local splitting

$$A = A^{\text{loc}} \oplus B,$$  \hspace{1cm} (23)

where $A^{\text{loc}}$ is a sparse, $n_{\text{loc}} \times n_{\text{loc}}$ matrix with a non-zero diagonal and $B$ is a sparse, rectangular matrix, whose entries are the coefficients of the external solution components. The number of rows in $B$, $n_{\text{ext}}$, equals the number of the external nodes.

To perform a distributed matrix–vector product, e.g. $A \cdot \psi$, each CPU must copy the relevant portion of the locally owned $\psi$ to the sending buffer, forward it to its neighbours and, at the same time, receive the required external and overlapping values of $\psi$ from the assigned neighbours. All values that are to be sent out are pre-weighted according to their valence on the sending CPUs; so at the receiving end, we simply sum up all the contributions to arrive at the required algebraic average. After all the local segments of $\psi$, which we call $\psi^{\text{loc}}$, have been set, on each CPU, $A^{\text{loc}}$ is multiplied by $\psi^{\text{loc}}$ (adjusted for the overlaps by now). Then, $B$ is multiplied by the local external components of $\psi$ and the resulting vector is added to the last $n_{\text{ext}}$ rows of $A^{\text{loc}} \cdot \psi^{\text{loc}}$. In the end, each CPU will have its local portion of the global vector $A \cdot \psi$.

Dot products are evaluated locally at first, applying weights to all overlapping terms. Then, the resulting local sums are added globally. Every single dot product requires one collective communication. The number of dot products per global iteration may be large, degrading the parallel performance substantially (e.g. Shadid & Timimaro 1994). The iterative classical Gram–Schmidt orthogonalization option, provided with the GMRES solver, allows the user to perform multiple dot products in one step, thus reducing the number of synchronization points.

The ILU preconditioner is constructed on each CPU for $A^{\text{loc}}$ only. As mentioned in the previous section, the preconditioning matrix is assigned once and for all.

As an illustration of the efficiency of the parallel implementation, we performed one-time-step runs of the FV code using a variable number of processors ($n_p$) and a global grid with $6 \times 10^5$ nodes. Fig. 5 shows the time elapsed as a function of $n_p$. In the case of perfect parallel efficiency, the results should follow a line with a slope of $-1$, as illustrated by the solid line on the figure. This ideal behaviour is evident for $n_p < 30$; at higher processor numbers, because of the increasing demand for communication between processors, a slight departure from this trend is evident and, indeed, expected. However, even at $n_p \sim 100$, the FV formulation maintains $\sim 80$ per cent efficiency (i.e. time elapsed in the ideal case over the actual time elapsed).

The performance shown in Fig. 5 is a strong function of the grid design and domain decomposition procedure, which impact the sparsity of the linear system associated with the FV discretization and the

![Figure 3](image-url)  
**Figure 3.** A simplified example of a triangulated subdomain in which we distinguish four types of nodes. The open circles, black circles, open squares and black squares are the external, overlapping, boundary and private nodes, respectively. The area occupied by the private and boundary nodes is shaded.

![Figure 4](image-url)  
**Figure 4.** The arrow-head structure of a typical local part of the global matrix $A$. This structure is achieved by renumbering the nodes associated with the subdomain, so that the private nodes are listed first and the external nodes are listed last. The local part of $A$ may be naturally split as $A = A^{\text{loc}} \oplus B$, where $A^{\text{loc}}$ is a sparse, square matrix of size $nn \times nn$ with a non-zero diagonal and $B$ is a sparse, rectangular matrix, whose entries are the coefficients of the external solution components. There is a zero block directly above matrix $B$. 

level of communication required between processors. We also note
that the improvement in speed of the simulation, as $n_p$ increases, is
just one of the major benefits of the parallel methodology; paral-
lelization permits one to adopt grids of much higher resolution and
presumably perform simulations of higher accuracy (see the next
section for a detailed discussion of this point).

3 BENCHMARK TESTS ON A SPHERICALLY SYMMETRIC EARTH MODEL

In this section, we adopt a specific spherically symmetric viscoelas-
tic Earth model and compare predictions for a suite of GIA data
sets generated using the FV formulation outlined above and the tra-
ditional, semi-analytical normal-mode methodology (Peltier 1974;

The density and elastic structure of the Earth model are given by
the seismic inference PREM (Dziewonski & Anderson 1981). The
simulations presented below are all run on a global grid, covering
the entire mantle, and a set of grid layers are positioned to reside
exactly at all major first- and second-order PREM discontinuities.
A linear interpolation is used to approximate the material properties
between any two layers of the PREM. Fig. 6 shows a projection of
the PREM on a 47-layer grid.

The radial viscosity profile of the Earth model is prescribed as
follows. In the top 120 km, an elastic lithosphere is assumed. This
is achieved, in practice, by setting the viscosity in this region to some
arbitrary high value, e.g. $10^{27}$ Pa s. Between the lithosphere and the
670-km discontinuity, viscosity is set to $5 \times 10^{20}$ Pa s. In the region
extending from this discontinuity to the CMB, the so-called lower
mantle, the viscosity is assigned a value of $\eta = 5 \times 10^{21}$ Pa s.
We note that the minimum Maxwell time for this model, $\text{Min}_{\eta/\mu}$, is approximately 108 yr. This is therefore an upper
bound on the time step for the explicit Euler scheme adopted here.

As discussed below, we will treat two different ice models in this
section. Regardless of this choice, it is assumed that an ice column
loads the spherical surface in the direction of the ambient, local
gravity field. Thus, eq. (13) gives for the surface traction

$$[\mathbf{\dot{u}} \cdot \mathbf{T}]_S = \rho^{\text{ice}} H \mathbf{\nabla} \Phi,$$

(24)

where $\rho^{\text{ice}}$ is the density of ice.

We have generated two regional and four global computational
grids for the simulations described below. Their key features are
provided in Table 1, which lists, in order, the number of grid nodes,
the number of spherical layers, the radial resolution at the surface, $\delta_0$
and the domain type (regional or global). The global grids G1–G4 are
characterized by progressively refined resolution. We will use
results based on these grids to assess the sensitivity of the numerical
error (as measured by the difference between the FV and normal-
mode calculations) to the spatial resolution of the grid.

The regional grids are centred at the north pole and extend an
angular distance of either $40^\circ$ (G0R) or $63.43^\circ$ (G2R). As described
above, the regional formulation assumes no displacement on the
side walls (a so-called clumped side boundary condition). The grid
G0R requires close to 800 MB of RAM if handled in serial and is
thus tractable with a single desktop PC. We use the grid to compare
predictions generated from serial versus parallel implementations
of the FV code. The regional grid G2R has a spatial resolution that
is identical in all respects to the global grid G2. We use the pair of
grids G2 and G2R to assess the impact of limiting the model
domain.

3.1 Simple disc ice load

We begin by considering the Earth model response to a simple ice
disc. The advantage of this preliminary calculation is that the sur-
face mass load may be defined without the ambiguity that is often
associated with global ice models, particularly with regard to the
timing of the load increments; it may thus serve as a good first test
in independent efforts to benchmark 3-D numerical codes.

![Figure 5](image1)

**Figure 5.** Total time elapsed versus number of processors, $n_p$, in a one-
time-step simulation using the finite-volume (FV) code (dashed line). All
calculations were performed on a global grid with $6 \times 10^5$ nodes. The solid
line shows the ideal scaling case (100 per cent parallel efficiency) in which
the time is proportional to $1/n_p$.

![Figure 6](image2)

**Figure 6.** Depth profiles of density and seismic velocities as given by the
Preliminary Reference Earth Model (PREM; Dziewonski & Anderson 1981).

<table>
<thead>
<tr>
<th>Grid</th>
<th>No. of nodes ($\times 10^{-6}$)</th>
<th>No. of layers</th>
<th>$\delta_0$ (km)</th>
<th>Domain type</th>
</tr>
</thead>
<tbody>
<tr>
<td>G0R</td>
<td>0.10</td>
<td>16</td>
<td>40</td>
<td>Regional</td>
</tr>
<tr>
<td>G2R</td>
<td>0.46</td>
<td>35</td>
<td>60</td>
<td>Regional</td>
</tr>
<tr>
<td>G1</td>
<td>1.25</td>
<td>29</td>
<td>60</td>
<td>Global</td>
</tr>
<tr>
<td>G2</td>
<td>1.80</td>
<td>35</td>
<td>60</td>
<td>Global</td>
</tr>
<tr>
<td>G3</td>
<td>3.84</td>
<td>40</td>
<td>30</td>
<td>Global</td>
</tr>
<tr>
<td>G4</td>
<td>12.30</td>
<td>60</td>
<td>15</td>
<td>Global</td>
</tr>
</tbody>
</table>
The ice disc is circular in horizontal cross-section, with a fixed diameter of 30°. The disc is centred at the north pole of the grid in order to take advantage of the grid pattern of concentric circles around the pole for a more accurate mapping of the circular disc boundary. The height variation across the disc (i.e. in vertical cross-section) is parabolic at any given time. If \( H \) denotes ice height, then

\[
H(\theta, t) = H_0(t) \left(1 - \frac{\theta^2}{\theta_{\text{max}}^2}\right),
\]

where \( \theta \) denotes colatitude, ranging from 0 to \( \theta_{\text{max}} = 15° \), and \( H_0 \) is the thickness of ice at \( \theta = 0 \) (Fig. 7a). The ice load is strictly zero outside the disc. The time dependence of the disc height is prescribed by the following relationship:

\[
H_0(t) = H_{\text{max}} \left\{\frac{1}{2} + \frac{1}{2} \text{erf} \left(\frac{\alpha}{t_0} \frac{t - t_0}{t_0}\right)\right\},
\]

where \( H_{\text{max}} = 1 \text{ km}, t_0 = 5000 \text{ yr} \) and \( \alpha = 2.628 \). This relationship is shown in Fig. 7(b).

We compared predictions generated from serial versus parallel (six CPU) implementations of the FV code run on grid G0R. The agreement in the two cases is excellent; peak values of radial displacement, tangential displacement and gravitational potential perturbation generally differ by less than 0.2 per cent. Ideally, the two predictions should, of course, be identical. However, in practice there is a difference in the algorithms applied in serial versus parallel applications that involves the application of the preconditioners, as mentioned in Section 2.4.

Next, we compare predictions generated using the FV code run on grids G2 and G2R. Fig. 8 shows both the radial and tangential displacement rates as a function of angular distance from the centre of the ice load \( t = 5000 \text{ yr} \) after the application of the disc load. The radial velocities computed using the two grids are in excellent agreement. The predicted tangential displacement is more sensitive to a reduction in the domain size. In this case, the agreement is excellent for \( \theta \leq \sim 30° \), or to an angular distance of twice the radius of the load. Beyond this zone, the regional grid underestimates the tail of the outward tangential displacement. This is not surprising given that the horizontal rates predicted by the global model extend, in contrast to radial displacements, well beyond the imposed location of the side boundary in grid G2R. It is important to note that one can obtain virtually the same accuracy in predictions of near-field 3-D motions using a regional rather than global grid (and, in the case we present here, the former requires ~4 times less computer memory than the latter). Conversely, accurate modelling of tangential deformations well beyond the periphery of the ice load requires the use of global scale grids.

Next, we turn to comparisons between results generated using the FV code and the traditional normal-mode formulation of the GIA problem. Fig. 9 shows predictions of radial and tangential crustal velocities \( t = 15 000 \text{ yr} \) after the onset of the test loading generated using the FV method on grid G3 (see Table 1). We choose to present results at \( t = 15 000 \text{ yr} \) because the time elapsed since the end of the main model loading phase (~5000 yr; see Fig. 7b) is, in this case, similar to the period between the end of the last deglaciation phase of the current ice age and the present day. The radial prediction shows sharply defined zones of post-glacial uplift and peripheral subsidence. Within 25° of the disc centre, the predictions are characterized by outward-directed tangential velocities; further afield, a broad zone of inward-directed horizontal motions is evident. The general form of these curves is consistent with earlier disc-load calculations based on the normal-mode approach (e.g. Mitrovica et al. 1994b).

In Fig. 10 we explore, in detail, the sensitivity of these discrepancies to the adopted FV grid resolution. In particular, we show, for each point where the radial (A, B) and tangential (C, D) velocities locally peak in Fig. 9, the relative (in per cent) and absolute discrepancy (in mm yr\(^{-1}\)) between the FV and normal-mode calculations as a function of the number of gridpoints in the former. The four
Figure 9. The radial (top) and tangential (bottom) crustal velocity at \( t = 15000 \) yr, as a function of angular distance from the disc load centre, computed using the finite-volume (FV) methodology on the numerical grid G3. The predictions at the specific points labelled A–D are used, in Fig. 10, to assess the level of discrepancy between the FV and independent normal-mode calculations as a function of the adopted grid.

We can conclude from Fig. 10 that obtaining predictions of the present-day 3-D crustal velocity field with an accuracy of \( \sim 0.1 \) mm yr\(^{-1}\) is generally achievable with an FV numerical grid of intermediate resolution (e.g. G3). However, the exception to this rule is the prediction of the large-amplitude (order 1 cm yr\(^{-1}\)) radial crustal velocity within zones of peak rebound (e.g. point A). In this case, accuracies better than 0.2 mm yr\(^{-1}\) require grids with relatively higher resolution, such as G4. In this regard, the results in Fig. 10 indicate that further, significant improvement in the accuracy of the predictions at site A, in contrast to sites B–D, is possible by adopting grids of higher spatial resolution than G4 (i.e. the decreasing trend in the numerical discrepancy for point A is not yet showing signs of convergence).

We note that while we have only plotted comparisons between the two numerical schemes at \( t = 15000 \) yr, we have investigated discrepancies over the full 15 000-yr window subsequent to the onset of loading. We have found that accurate predictions at later times are more demanding in terms of the required spatial resolution of the FV grid than those at earlier times in the loading phase. Thus, choosing a grid resolution on the basis of the results in Fig. 10 in order to obtain a specified level of accuracy ensures the same level of accuracy (or better) at all earlier times in the simulation.

3.2 ICE-3G calculations

Next, we turn our attention to benchmark comparisons of predictions based on the global ICE-3G deglaciation model (Tushingham & Peltier 1991). The ICE-3G load history provides a model of the space–time history of ice cover since the LGM (which was assumed to occur at 18 000 BP). At the LGM, major ice complexes covered most of Canada and the northeastern USA, the Arctic, Fennoscandia, East and West Siberia, the Kara and Barents sea, and Scotland; furthermore, both the Antarctic and Greenland ice complexes were
significantly larger than at present day. From 18 000 to 5000 BP (the assumed end of the deglaciation phase), a meltwater volume equivalent to \( \sim 100 \) m of eustatic (i.e. geographically uniform) sea level rise entered the ocean according to the ICE-3G model.

Caution is necessary when implementing the ICE-3G model into GIA loading calculations. The spatial geometry of the ice volume snapshots is discretized into a series of circular discs of various radii. This representation is routinely rediscrized into ice elements bounded by latitude/longitude pairs in order to avoid holes that exist between neighbouring discs. For the purposes of our benchmarks, both the FV code and the normal-mode calculations preserve the exact circular geometries described by Tushingham & Peltier (1991). The temporal discretization of the ICE-3G model is also a common source of confusion. While the model is defined at integer times from 18 000 to 5000 BP, any change in ice volume between successive time steps is assumed to entirely occur at the earlier time boundary. That is, the difference in ice volumes at, for example, 12 000 and 11 000 BP, is assumed to melt in a step-like fashion at 12 000 BP (rather than linearly across the time interval). Our benchmarks adopt the sequence of step changes in ice volume implicit in the ICE-3G model representation. They also assume a complementary, eustatic ocean (meltwater) load.

![Maps of present-day radial and tangential crustal velocities over North America predicted using the finite-volume (FV) code run on grid G3 (Table 1). The predictions are based on the ICE-3G deglaciation history, with a complementary, eustatic water load, and the reference spherically symmetric, viscoelastic Earth model introduced above. These maps show the basic regional character of the 3-D velocity fields evident in recent predictions derived from the traditional normal-mode approach to GIA (e.g. James & Lambert 1993; Mitrovica et al. 1994b). In particular, the radial motions are characterized by a zone of post-glacial uplift encircled by subsidence within peripheral bulges. The peak uplift rate, for this ice/Earth model combination, reaches over \( 13 \) mm yr\(^{-1}\), while subsidence rates reach several mm yr\(^{-1}\). Within the near field, horizontal crustal velocities are directed outwards from the zone of maximum uplift (where they trend towards zero), but this pattern reverses in the far field to yield a broad pattern of inward (toward the ancient zone of glaciation) motions. The near-field horizontal rates reach amplitudes of close to \( \sim 2 \) mm yr\(^{-1}\), while the far-field rates peak under \( \sim 1 \) mm yr\(^{-1}\). Similar patterns were evident in our calculations based on the simple disc load model (Fig. 9).

It is important to demonstrate that accuracies achieved by the FV code in the benchmark calculations described in the last section, based on the parabolic disc ice model, are also obtainable when a more complex surface mass load such as ICE-3G is adopted. To this end, in Fig. 12 we compare numerical predictions based on the FV code (intermediate grid G3) with normal-mode calculations for points along a James bay—Florida great circle profile.

Once again, there is excellent agreement between the FV and normal-mode approaches. We have chosen to show results for grid G3 in Fig. 12 to reinforce an observation made in the context of the parabolic disc benchmarks: namely, the intermediate grid G3 is of sufficient resolution to yield accuracies of order 0.1 mm yr\(^{-1}\) or less in all locations, except zones of peak radial rebound (point A). As an example, the absolute discrepancy between the two predictions at the points of peak radial subsidence (point B) and outward-directed horizontal motion (point C) on the profile are \( \sim 0.1 \) and 0.02 mm yr\(^{-1}\), respectively, which is consistent with the G3 results in Fig. 10. However, the discrepancy reaches \( \sim 0.6 \) mm yr\(^{-1}\) at point A. As an indication of the sensitivity of this value to the grid size, the

Figure 11. Maps of present-day radial and tangential crustal velocities over North America predicted using the finite-volume (FV) code run on grid G3 (Table 1). The predictions are based on the ICE-3G deglaciation history, with a complementary, eustatic water load, and the reference spherically symmetric Earth model described in the text. Profiles along the great circle extending from James bay to Florida are shown in Fig. 12.

Figure 12. Predictions of present-day radial (top) and southward tangential (bottom) crustal velocity along a great circle profile extending from James bay to Florida (see Fig. 11 for the location of this profile). The two lines on each frame distinguish calculations based on the finite-volume (FV) methodology on the intermediate resolution grid G3 (dashed curve) and normal-mode approach (solid curve). Predictions at the specific points labelled A, B and C are discussed within the text. All calculations are based on the ICE-3G deglaciation history, with a complementary, eustatic water load, and the reference spherically symmetric Earth model defined in the text.
discrepancy at point A increases to ~2.0 mm yr\(^{-1}\) for the coarse grid G1 and it will decrease to ~0.4 mm yr\(^{-1}\) for grid G4. We note that the apparent (and small) shift in the near-field horizontal velocity predictions in Fig. 12 is of the order of the surface grid resolution.

One of the most important sets of observables treated in the GIA literature are the present-day (secular) rates of change in the long-wavelength zonal harmonics of the zonal geopotential of the Earth, the so-called \(J_l\) coefficients (for small \(l\)). Indeed, a comparison of numerical predictions of these harmonics (in particular, for \(l = 2\)) with satellite-derived constraints has been used in a large number of studies as a basis for the inference of deep mantle viscosity and/or recent cryospheric mass balance (for a recent example, see Sabadini et al. 2002). To complete our benchmarks, we show, in Table 2, predictions of \(J_2\) up to \(l = 8\) based on the traditional normal-mode approach and the FV code on grid G3 (as used in Fig. 12). The relative discrepancies range from less than 1 per cent (e.g. degree 2) to ~4 per cent (degree 7). In absolute terms, the difference in the \(J_2\) predictions is \(0.02 \times 10^{11}\) yr\(^{-1}\), which is significantly smaller than the observational uncertainty (\(0.2 \times 10^{11}\) yr\(^{-1}\); e.g. Sabadini et al. 2002). The current observational error at higher degrees (\(l = 3–6\)) ranges from 0.3–1.0 \(\times 10^{11}\) yr\(^{-1}\); once again, this is larger than the discrepancies evident in Table 2. We have also compared FV predictions of \(J_2\) generated from the complete sequence of grids G1–G4 in Table 1; in this case, the results show less than 2 per cent change from the G3 grid predictions given in Table 2), even for the FV grid G1. We conclude that a spatial resolution consistent with the coarse grid G1 is sufficient to ensure that predictions of \(J_2\) based on our FV formulation have an accuracy smaller than the current observational uncertainty.

<table>
<thead>
<tr>
<th>Degree</th>
<th>Normal mode (\times 10^{11})</th>
<th>Finite volume—G3 (\times 10^{11})</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-4.81</td>
<td>-4.84</td>
</tr>
<tr>
<td>3</td>
<td>1.53</td>
<td>1.50</td>
</tr>
<tr>
<td>4</td>
<td>-4.61</td>
<td>-4.53</td>
</tr>
<tr>
<td>5</td>
<td>4.23</td>
<td>4.14</td>
</tr>
<tr>
<td>6</td>
<td>-1.98</td>
<td>-1.93</td>
</tr>
<tr>
<td>7</td>
<td>4.02</td>
<td>3.87</td>
</tr>
<tr>
<td>8</td>
<td>-0.77</td>
<td>-0.75</td>
</tr>
</tbody>
</table>

**4 FIRST RESULTS FROM 3-D EARTH MODELS**

Space-geodetic surveying using the GPS is providing unprecedented constraints on 3-D crustal deformation rates, including within major centres of Late Pleistocene ice cover (e.g. Johansson et al. 2002). Analyses of these data have heretofore assumed spherically symmetric Earth models (e.g. Milne et al. 2001). In this section, as a first application of the FV code to 3-D Earth models, we examine the impact of lateral variations in mantle viscosity on predictions of present-day, GIA-induced crustal velocities in North America.

As in the last section, our predictions will be based on the ICE-3G deglaciation history, a eustatic meltwater load, and the radial elastic and density structure given by PREM. We also assume a geographically uniform elastic lithosphere of thickness 120 km. The geometry of the mantle viscosity field is constrained from seismic tomographic images of mantle structure. In particular, our 3-D viscosity field is prescribed through a sequence of steps that may be summarized by the following equations:

\[
\delta \ln \rho(r, \theta, \phi) = \frac{\partial \ln \rho}{\partial \ln v_s} \delta v_s(r, \theta, \phi), \tag{27}
\]

\[
\delta T(r, \theta, \phi) = -\frac{1}{\alpha(r)} \ln \rho(r, \theta, \phi), \tag{28}
\]

\[
\eta(r, \theta, \phi) = \eta_0(r) e^{-\epsilon \delta T(r, \theta, \phi)}, \tag{29}
\]

where \(r, \theta, \phi\) represent the position coordinates of radius, co-latitude and east longitude.

To begin, we adopt the recent seismic shear wave heterogeneity model S20RTS of Ritsema et al. (1999). This model provides relative variations in seismic wave speed throughout the mantle, \(\delta v_s(r, \theta, \phi)\) in eq. (27) and it is based on a large suite of seismic data sets, including normal modes, travel times and surface wave data. We next convert this heterogeneity to a model of the relative variation in density, the left-hand side (LHS) of eq. (27), using a depth-dependent velocity-to-density scaling profile. For this purpose, we adopt the profile of Forte & Woodward (1997), which was modified, on the basis of convection-related geodynamic constraints, from a profile of Karato (1993). The 3-D density field is then converted to a temperature field (eq. (28)) using the depth-dependent variation in the coefficient of thermal expansion, \(\alpha(r)\), derived by Chopelas & Boehler (1992). Finally, we assume an exponential dependence of the viscosity field, \(\eta(r, \theta, \phi)\), on the temperature variation. This dependence is shown in eq. (29), where \(\eta_0(r)\) refers to the reference radial profile of mantle viscosity. The strength of this dependence and thus the peak-to-peak lateral variation in viscosity are prescribed by the free parameter \(\epsilon\). A further, radially dependent scaling is applied to the RHS of eq. (29) in order to ensure that the mean value of \(\eta(r, \theta, \phi)\) at any depth is the same as the reference model adopted in the benchmark calculations: a sublithospheric upper-mantle viscosity of \(5 \times 10^{20}\) Pa s and lower-mantle viscosity of \(5 \times 10^{21}\) Pa s. When \(\epsilon = 0\) the system collapses to the spherically symmetric case; the relevant predictions of 3-D crustal velocities in this special case are given in Fig. 11.

Fig. 13 provides a measure of the lateral variation in viscosity introduced when we adopt a value of \(\epsilon = 2.5\) in the algorithm defined by eqs (27)–(29). For this choice, the standard deviation of the grid values of \(\delta \log \eta\) for layers within the lower mantle varies from...
∼0.4–0.7; this indicates that ∼50 per cent of the grid elements have a viscosity that falls within a 1 order of magnitude band around the reference lower-mantle value of 5 × 10^{21} \text{ Pa s} and ∼95 per cent fall within a 2 order of magnitude band. Our scaling procedure yields a slightly more limited (by a factor of ∼2 in log-space) lateral variation in viscosity over most of the upper mantle; however, the variation is significantly larger within the ∼200-km zone below the lithosphere (i.e. the asthenosphere).

In Fig. 14, we show maps of the impact on predictions of present-day radial and tangential crustal velocity of introducing this lateral variation in mantle viscosity into the FV calculations (the plots show the 3-D Earth model results minus the spherically symmetric reference results). As a companion to this figure, Fig. 15 samples the differential predictions of radial and southward tangential velocity along the James bay—Florida great circle profile introduced in the context of Fig. 12.

One can compare Fig. 14 with Fig. 11 and Fig. 15 with Fig. 12. It is clear from this comparison that incorporating the 3-D mantle viscosity field prescribed above has a relatively moderate impact on the predictions of radial crustal rates. In zones of peak post-glacial rebound, the effect is of order 1 mm yr^{-1}, or approximately 10 per cent of the total signal; the impact is less than ∼0.5 mm yr^{-1} in the peripheral bulge regions. Along the profile extending from James bay to Florida, the 3-D Earth model shows peak uplift rates (James bay) approximately 0.7 mm yr^{-1} (or 6 per cent) higher than the reference, spherically symmetric predictions; subsidence rates within the peripheral bulge are approximately 0.2 mm yr^{-1} lower than the reference calculation.

In contrast, the incorporation of lateral variations in mantle viscosity (while maintaining the same globally averaged radial variation) has a significantly larger (relative) impact on predictions of horizontal motions. Indeed, the differential rate vectors in the near field of Fig. 14 have magnitudes that are comparable to the predictions generated by the spherically symmetric Earth reference model (Fig. 11). In particular, introducing the mantle viscosity variation summarized in Fig. 13 acts to significantly weaken the amplitude of the near field, outward-directed crustal velocities predicted using the reference model (Fig. 11); that is, the 3-D minus reference result shows a pronounced, inward-directed tangential motion perturbation of peak amplitude ∼1 mm yr^{-1} (Fig. 14). We note that this signal exceeds, by approximately an order of magnitude, the current observational uncertainty obtained from surveying using GPS (e.g. Johansson et al. 2002; Davis et al. 2003). In any event, the net effect is a profound change in the predicted, large-scale pattern of horizontal motions.

The perturbation to the horizontal crustal velocity field associated with the introduction of lateral variations in mantle viscosity is a longer wavelength signal than the analogous perturbation to the radial motions. What feature of the imposed 3-D viscosity field gives rise to this long-wavelength, large-amplitude signal over North America? As we have discussed, the construction of the 3-D viscosity field is constrained so as to preserve the globally averaged depth profile of mantle viscosity. However, the question arises: what is the regional depth profile of viscosity below Laurentia? Fig. 16 (top frame) shows this profile in the case where the averaging of the viscosity field at any depth in the model is limited to a spatial window centred around Hudson bay (see caption). This regional
depth profile differs from the globally averaged (reference) profile in two significant ways: first, the viscosity in the bottom half of the mantle is a factor of 2–4 stiffer than the reference value; secondly, the asthenospheric viscosity reflects the presence of a stiff, thick (∼200 km) sublithospheric continental root.

We used the FV code to compute the deformation of a spherically symmetric Earth model that was identical in all respects to our reference 1-D Earth model, with the exception that the simple two-layer viscosity profile in the latter was replaced by the regional viscosity depth profile shown in the top frame of Fig. 16. The difference between the southward tangential crustal velocity predicted for this case and the prediction based on the original reference model is shown by the solid line in Fig. 16 (bottom frame). For the purposes of comparison, this frame also shows the difference between the 3-D Earth model prediction and the original reference 1-D Earth model calculation (dashed line, taken directly from Fig. 15, bottom frame). The two curves show similar trends, indicating that the predictions over North America in the 3-D Earth model case are sampling some type of spatially weighted local depth profile of viscosity; in this case, the local depth profile involves the presence of a thick continental root and a stiff deep mantle. Of course, the difference in the amplitude of the two curves in the bottom frame of Fig. 16 indicates that this type of exercise has limitations and, more importantly, that obtaining accurate predictions of tangential crustal velocities requires that the complexity of the full 3-D Earth model be modelled.

5 Final Remarks

We have presented an FV parallel computer code for forward modelling the viscoelastic response within the mantle and crust of a 3-D, self-gravitating Earth to an arbitrary surface load. Our formulation assumes a Maxwell rheology and includes elastic compressibility. The conservative, control volume discretization of the governing equations is implemented on a tetrahedral grid in Cartesian geometry, using a low-order (linear) interpolation. We have developed and described here a suitable parallel grid generator for the problem, with a flexible overlapping domain decomposition scheme. A basic starting grid is constructed to honour all major radial discontinuities in PREM. The models are allowed to have arbitrary spatial variations in the elastic parameters and viscosity: these variations may be either continuous or discontinuous at the grid nodes, where the actual interface is a prescribed 3-D surface within the modelling domain. The modelling domain can either be global or regional. In the latter case, a set of clumping lateral boundary conditions is assumed. While we have focused on applications and testing related to the GIA problem, the formulation (and code) can also be easily adapted to yield predictions of post-seismic relaxation.

The FV code has been benchmarked by comparing a suite of predictions based on a spherically symmetric test model of the Earth with results generated using the traditional, normal-mode approach to GIA problems. We have found that an FV numerical grid with intermediate spatial resolution, e.g. grid G3 with a total of ∼4 × 10⁶ nodes, is generally sufficient to yield absolute discrepancies (compared to the normal-mode calculations) of order 0.1 mm yr⁻¹ or less in predictions of present-day 3-D crustal velocities. The exception to this basic rule involves the prediction of radial rates in zones of peak post-glacial rebound; in this case, achieving absolute discrepancies of order 0.1–0.2 mm yr⁻¹ (for total signals of order 1 cm yr⁻¹) requires numerical grids with significantly higher spatial resolution (e.g. G4). In contrast, we find that very accurate predictions of long-wavelength zonal harmonics in the rate of change of the geopotential of the Earth, the so-called J₄ coefficients, may be obtained with relatively coarse grids (e.g. G1).

We are currently working on extending our grid generation scheme to permit local grid refinement that can be applied selectively near surface loads. In this way, absolute accuracies less than a pre-defined tolerance may be achieved in all regions of post-glacial adjustment (near, medium and far field) with less computational effort than would be necessary with numerical grids characterized by (more or less) globally uniform surface spatial resolution (e.g. G1–G4).

The present study includes the first application of our FV formulation to the GIA problem on 3-D Earth models. Specifically, we investigated the effect of lateral variations in mantle strength on predictions of present-day, 3-D crustal velocities in North America. In this investigation, the geometry of the mantle viscosity variation at each depth was constrained using images from seismic tomography and the amplitude was tuned to yield an ∼2 order of magnitude peak-to-peak variation. We find that the introduction of lateral variations in viscosity has a more significant (relative) impact on horizontal velocities than on radial velocities. Indeed, the former is sufficient to yield a profound change in the pattern of GIA-induced horizontal velocities over Laurentia. We conclude that future analyses of space-geodetic data obtained from previously glaciated regions should consider the impact of 3-D viscosity structure on the GIA predictions.

As a final point, the surface mass load associated with GIA involves both an ice and a complementary global ocean load. The
former is generally prescribed a priori from regional or global models of Late Pleistocene ice cover, while an accurate prediction of the latter requires the solution of the so-called sea level equation (e.g. Farrell & Clark 1976). The sea level equation governs the gravitationally self-consistent redistribution of ocean mass; it is an integral equation because the redistribution influences the gravitational field of the planet and is, in turn, governed by this field. Milne et al. (1999) have derived a sea level equation for spherically symmetric Earth models that is revised from the standard form (as in Farrell & Clark 1976) to accurately account for the evolution of shorelines (as a result of both ocean height variations and local changes in the ice sheet perimeter) and perturbations in Earth rotation. Mitrovica & Milne (2003) have recently derived a generalized sea level equation that takes proper account of shore line evolution on any global Earth model: as long as a method exists for computing load-induced perturbations in the solid surface and gravitational potential in the model. In practice, our GIA predictions on global, 3-D Earth models will be performed in two stages. First, the FV numerical formulation described below would be combined with the Mitrovica & Milne (2003) sea level theory to iteratively compute the gravitationally self-consistent water load history. Next, the combined ice plus water load would be applied to the numerical model to predict any geophysical observables of interest such as 3-D crustal motions, gravity anomalies, etc. In the present paper, our benchmarks were defined by a known surface mass load; in future work, we will outline methods we have developed for coupling these results to the generalized sea level theory.

ACKNOWLEDGMENTS

We thank Yanick Ricard and two anonymous referees for their constructive criticism and a number of valuable comments which have led to a much improved final version of the manuscript.

The authors acknowledge research support from NSERC, CIAR, the Miller Institute for Basic Research in Science and the National Science Foundation grant EAR-0309576.

REFERENCES


Haskell, N.A., 1935. The motion of a fluid under a surface load, 1, Physics, 6, 265–269.


© 2005 RAS, GJI, 161, 421–444
Glacial isostatic adjustment on 3-D Earth models


van der Vorst, H., 1981. Iterative solution methods for certain sparse linear


Wu, P. & Peltier, W.R., 1984. Pleistocene deglaciation and the Earth’s rota-
Wu, P. & van der Wal, W., 2003. Postglacial sealevels on a spherical, self-


A1 Grid generation and domain decomposition

A good quality grid is a very important component in FV modelling. The grid generator must be able to mesh either the whole globe or subregions, support a layered structure that honours all major PREM radial discontinuities and feature grid refinement towards the surface of the Earth so that, for example, topographic variations and ocean–continent boundaries can be adequately mapped. For benchmarking purposes, it is prudent to have a single grid for the entire simulation domain, suitable for a serial solver. However, the principal output should be a distributed grid with all necessary communication information between the subdomains, such as the nodal types and global numbering, discussed in Section 2.4. To facilitate the discretization of the original boundary conditions (not to be mixed with the conditions on the artificial subdomain boundaries), we require yet another set of nodal attributes, which identifies the nodes on the lower, upper and side boundaries of $S$, as well as the inner nodes, $\Omega$–$S$. A suitable domain decomposition must ensure a minimum

© 2005 RAS, GJI, 161, 421–444
Figure A1. Spherical grid enrichment algorithm. An initial tetrahedron \{v^1, v^2, v^3, v^4\} (a) produces four tetrahedra, \{v^1, e_{12}, e_{13}, e_{14}\}, \{v^2, e_{12}, e_{23}, e_{24}\}, \{v^3, e_{13}, e_{23}, e_{34}\} and \{v^4, e_{14}, e_{24}, e_{34}\}, and an octahedron \{e_{12}, e_{13}, e_{23}, e_{24}, e_{34}\} (b). In this example, the octahedron is subdivided further into four tetrahedra by a new edge \{e_{14}, e_{23}\}. The bars above some of the vertices (in the notations) indicate spherical projection.

possible scatter between the subdomains with regard to the number of nodes, connections (particularly at the boundary) and neighbouring subdomains. Note that from a parallel efficiency perspective, if one of the subdomains must be different, it should have fewer nodes, connections and neighbours than the rest, otherwise load balancing would become an issue.

Our approach is to generate an initial grid on a single computer, perform domain decomposition subject to the specified number of CPUs and grow the subdomains to the desired size. The last two operations are done in parallel. This section discusses the steps involved in the grid generation and domain decomposition.

We have adopted a routine to refine a tetrahedral grid from a spherical grid generator package, coded by M. Everett and kindly provided to KL. For a more detailed description of this algorithm see Everett (1997) and Liu & Joe (1996). In short, the routine takes a tetrahedron \{v^1, v^2, v^3, v^4\} and adds six new gridpoints to the mid-edges, such that we obtain four new tetrahedra at the corners of the old element plus an octahedron in the middle, as shown in Fig. A1. The octahedron is subdivided further into four new tetrahedra by creating a new edge, joining the best pair of vertices of the octahedron. The best pair of vertices is chosen to maximize the sum of quality factors of the newly produced elements. The quality factor of a single tetrahedron is defined in Everett (1997) as

\[
Q = \frac{12 (\Omega^2)^2}{\sum_{i \neq j} \Lambda_{ij}},
\]

where \(\Omega^2\) is the volume of the tetrahedron and \(\Lambda_{ij}\) is the length of the edge joining vertices \(i\) and \(j\). To enforce a spherically layered structure, if a new node is produced by a pair of old nodes that lie on the same spherical surface, the new node is projected radially outwards to that surface. Note that the new nodes in Fig. A1, \(e_{23}\), \(e_{24}\) and \(e_{34}\), are radially projected. This fact is reflected by attaching bars to the notations. Thus, the old surface \(\{v^2, v^3, v^4\}\) produces a new surface \(\{v^2, \bar{v}^{23}, v^3, \bar{v}^{34}, v^4, \bar{v}^{24}\}\), which is a refined approximation of a sphere. The octahedron \(\{e_{12}, e_{13}, e_{14}, e_{23}, e_{24}, e_{34}\}\) in this example is subdivided into four tetrahedra by the edge \(\{e_{14}, e_{23}\}\). If both generating nodes belong to the same spherical surface and, in addition, their colatitudes are the same, the new node is optionally forced to have that colatitude by moving the new node along the spherical surface. This transformation results in a geographical grid.
Figure A3. Sample cross-section of part of a meshed spherical shell obtained from the shape shown in Fig. A2(a) after two refinement levels. The region near the origin is cut out.

on the surface from the poles to approximately 50° latitude, which may be useful in, for example, mapping model circular loads.

The elements are packed deterministically into the modelling domain by starting from initial bounding shapes constructed from a few tetrahedra. The most useful of those are shown in Fig. A2, where N is the north pole, S is the south pole and O denotes the centre of the Earth. Areas around the north pole are shaded for clarity. For a regional grid, we take the north pole and the origin, add either five or six nodes regularly spaced along a specified colatitude and join them as shown in Fig. A2(a). The side surface would approximate a cone if we employ the equal colatitude option. The initial shapes for the entire globe are either an icosahedron (Fig. A2b) or a dodecahedron (Fig. A2c). Fig. A3 shows a meshed spherical shell, obtained from the shape shown in Fig. A2(a) after two refinement levels. We identify six types of nodes with respect to their position in the domain: upper surface, upper rim, side surface, lower rim, lower surface and inner nodes. It is critical to update this information as the grid grows. Note that, in the case of a whole mantle domain, we would have the upper surface, lower surface and inner nodes only.

To add some flexibility to the building process, we introduce the following three operations on the grid: to cut out any specified selection of elements, to attach any number of spherical layers to either the top or the bottom of an existing shell and to attach spherical layers with grid refinement. In fact, we have used the cut operation to produce Fig. A3 by removing the central core.

The layer construction deserves further explanation. In building a spherical layer, we generally follow the approach of Everett (1997), which is to project the originating, triangulated spherical surface to the new desired radial location by constructing triangular barrels from each of the surface triangles and subdividing each such barrel into three tetrahedra. One needs three new edges on the three side faces of the barrel. There are eight independent ways to create the required three new edges, as illustrated in Fig. A4(a)–(h), but two of them (d) and (h) do not produce tetrahedra and must be discarded. The algorithm picks up the originating surface triangles, subject to their ordering in the current grid. The three new elements are then constructed, using the first available valid partitions out of six, which delivers the maximum sum of quality factors for the new elements. Special care must be exercised in order to avoid edge crossings between the shared rectangular sides of the neighbouring barrels. An example of such a crossing is shown in Fig. A5. If the cut operation is performed too frequently or the nodes are renumbered in a certain deterministic fashion while building the grid, the crossing problem may occasionally result in a deadlock, where the only available partitions for the current barrel are those of type (d) or (h).

The layer construction with grid refinement is a well-defined operation, shown in Fig. A6, free of the edge-crossing problem. It allows us to achieve better lateral resolution toward the surface of the Earth, which is naturally degraded by the spherical divergence.
...and a set of concentric circles of fixed latitude in the interim, di-
of boundary divisors consists of two circles, centred at the poles, in the manner shown in Fig. A7 for the whole globe. The pattern equal square area but not necessarily similar shape. This is done therefore solid angles bounded on the top surface by regularly shaped patches of equal square area would contain approximately Therefore, solid angles bounded on the top surface by regularly shaped patches of equal square area would contain approximately the same number of nodes and elements. We use this fact in perfor-
...the domain decomposition. The latter is reduced to a 2-
...4, which is shown separately in Fig. A8(b), does not contain node \(v^4\). The sum of \(\Omega^{P,4}\) gives the volume of the element, denoted earlier by \(\Omega^2\). The non-dimensional volumetric coordinates are then defined as follows:

\[
L^{P,l} = \frac{\Omega^{P,l}}{\Omega}, \quad l = 1, \ldots, 4.
\]  

(A4) 

To approximate the first derivative of any field, sampled at the nodes, in terms of \(L^{P,l}\), we simply differentiate eq. (A3) with respect to \(x, y\) or \(z\), as required, e.g.

\[
\partial_x F^P = F^1 \partial_x L^{P,1} + F^2 \partial_x L^{P,2} + F^3 \partial_x L^{P,3} + F^4 \partial_x L^{P,4}.
\]  

(A5) 

The explicit expressions for \(L^{P,l}\) and its derivatives are most easily obtained from geometrical considerations. We derive here the form for \(L^{P,4}\) only, as the other three can be completed by inspection, simply permuting indices. Consider again Fig. A8(b) and recall that volume \(\Omega^2\) can be computed as one third of the triangular base area \(\{v^1, v^2, v^3\}\) times the length of the normal to the base, erected from point \(P\). Let \(v'\) be a vector connecting vertex \(v^j\) to vertex \(v^j\) and \(p'\) be a vector connecting vertex \(v^j\) to point \(P\), then

\[
\Omega^2 = \frac{1}{3} \left[ \frac{1}{2} (v^{12} \times v^{13}) \cdot p' \right].
\]  

(A6)
Expanding the vector product in the above formula as a determinant in component form and using expression (A4), one obtains

\[
L^{P,4}(x, y, z) = (x^P - x^1) \partial_x L^{P,4} + (y^P - y^1) \partial_y L^{P,4} + (z^P - z^1) \partial_z L^{P,4},
\]

where

\[
\partial_x L^{P,4} = \frac{1}{\Delta \omega} [(y^2-y^1)(z^3-z^1)-(z^2-z^1)(y^3-y^1)],
\]

\[
\partial_y L^{P,4} = \frac{1}{\Delta \omega} [(x^2-x^1)(z^3-z^1)-(z^2-z^1)(x^3-x^1)],
\]

and

\[
\partial_z L^{P,4} = \frac{1}{\Delta \omega} [(x^2-x^1)(y^3-y^1)-(y^2-y^1)(x^3-x^1)].
\]

It is worth mentioning that because we adopt a linear interpolation, the gradient of any function or divergence of any field within an element becomes effectively constant. Thus, because \( \nabla L \) does not depend on the location of point \( P \), we shall drop the symbol \( P \) from the superscripts of the components of \( \nabla L \).

A completely analogous 2-D linear interpolation formalism exists. One simply replaces \( \Omega^e \) by a suitable surface area and the volumetric coordinates by the surface coordinates. The surface coordinates are constructed as per Fig. A8(a), placing point \( P \) on one of the triangular surfaces.

In the course of the control volume integration, we will encounter both surface and volume integrals. The discretized kernels of these integrals are products of linear functions obtained from formulae (A3) or (A5). Gaussian quadrature is a common technique for evaluating such integrals. For the volumetric integration we have

\[
\int F d\Omega = \Omega \sum_{q=1, m} F^q \omega^q + O^{n+1}(m),
\]

where the integrand is to be sampled at \( m \) quadrature points \( q \) inside the volume with weights \( \omega^q \), so that \( \sum_{q=1, m} \omega^q = 1 \). The size of the residual \( O^{n+1} \) is determined by the order of the quadrature rule, \( n \). If \( F \) is a polynomial of degree \( \leq n \), the integration is exact. The corresponding expression for the surface integration has the same generic form, where \( \Omega \) is replaced by \( S \).

The following is a complete summary of all the applicable quadrature rules for our FV integration, adopted from Huyakorn & Pinder (1983, pp. 97–98), where the sampling points are tabulated directly in the volumetric or surface coordinates. If \( F \) is a linear function, it is sufficient to sample it at the centre of either a volume or a surface and take \( \omega = 1 \). A second-order rule for triangles is obtained by sampling \( F \) at the mid-edges with \( \omega = 1/3 \) for each. A fifth-order quadrature for volumes requires \( F^1 = (1, 1, 1)/4, F^2 = (1/2, 1/6, 1/6, 1/6), F^3 = (1/6, 1/6, 1/6, 1/2) \) and \( F^4 = (1/6, 1/6, 1/6, 1/2) \) with \( \omega^1 = -16/20 \) and \( \omega^2 = 9/20 \) for \( k = [2, 3, 4, 5] \), where volumetric coordinates inside the brackets are used. A fifth-order surface integration involves seven sampling points \( F^1 = (1, 1, 1)/3, F^2 = (1, 1, 1)/3, F^3 = (1, 1, 1)/3, F^4 = (1, 1, 1)/3, F^5 = (1, 1, 1)/3, F^6 = (1, 1, 1)/3, F^7 = (1, 1, 1)/3 \) and \( F^8 = (1, 1, 1)/3 \), where \( \omega_1 = 0.05961587, \omega_2 = 0.470140266, \omega_3 = 0.797472699 \) and \( \omega_4 = 0.10128651 \) with \( \omega^1 = 0.225, \omega^2 = 0.13239415 \) for \( l = [2, 3, 4] \) and \( \omega^1 = 0.1259318 \) for \( l = [5, 6, 7] \).
Summation over all such subvolumes $\Omega^c$ sharing the grid node of interest is implicit in the above integrals to avoid clutter. Each of the $J$ integrals for the displacement components (eq. A15) is further a sum of six terms according to expression (A12):

$$J^\alpha = \sum_{l=1,6} J_{l,\alpha},$$

$$J_{l,\alpha} = \int \hat{n}_l \mu (\partial_s x_j + \hat{v}_j) dS^l,$$

where $J_{l,\alpha} = \int \hat{n}_l \mu l \ d\Omega^c$ and $\alpha = \{x, y, z\}$. There is a total of 19 integrals.

Consider integrals $J_{l,\alpha}$, where $\alpha = \{x, y, z\}$ and $l = \{1, \ldots, 6\}$, and $J^\Phi$, given by expressions (A15)-(A17). Because $\kappa$ and $\mu$ are both interpolated as linear functions and, in our equations, always appear together as $\kappa - 2\mu / 3$, we introduce a shorthand notation $\lambda = \kappa - 2\mu / 3$ and apply linear interpolation to $\lambda$. The first nine terms for $\lambda = \{x, y, z\}$ and $l = \{1, 2, 3\}$ describe a purely elastic response. The notation $\hat{v}_j$, where $\alpha = \{x, y, z\}$, will be used to indicate a projection of $\nu$ onto the Cartesian axes (i.e. directional cosines). Integrating $J^{r,1}$, we obtain

$$J^{r,1} = \int \hat{n}_l \lambda (\partial_s x_j + \hat{v}_j) dS^l,$$

$$= \int (\partial_s U + \partial_s V + \partial_s W) \lambda \hat{v}_j dS^l.$$

Note that the surface of integration includes $S^l$ only, because $J^{r,1}$ cancel out on surfaces $S^l$ inside the control volume, as per condition (13). In the case when $S^l$ belongs to the domain boundary, there will be a contribution to the RHS that we will discuss separately. Applying a linear interpolation to the terms in the above integral, we observe that the kernel is a linear function because the divergence is a constant. Thus, the first-order quadrature rule yields

$$J^{r,1} = \sum_{q,s} \int \hat{n}_l \lambda (\partial_s x_j + \hat{v}_j) dS^l,$$

$$= \int (\partial_s U + \partial_s V + \partial_s W) \lambda \hat{v}_j dS^l.$$

where $\lambda$ is to be interpolated at the centres of the six triangular faces $S^l$ shown in Fig. 2(b). The relevant interpolation rules can be derived in a straightforward manner. For example, for face $S^{1,4}$, we have at each vertex $\lambda (e^{12}) = (\lambda^1 + \lambda^2) / 2$, $\lambda (e^{13}) = (\lambda^1 + \lambda^3) / 2$, and $\lambda (f^{123}) = (\lambda^1 + \lambda^2 + \lambda^3) / 3$. If we denote the value of $\lambda$ at the centre of face $S^{1,4}$ by $\lambda (S^{1,4})$, the mid-point formula gives $\lambda (S^{1,4}) = \{\lambda (e^{12}) + \lambda (e^{13}) + \lambda (f^{123})\} / 3$ or, after grouping the coefficients,

$$\lambda (S^{1,4}) = \frac{13}{36} \lambda^1 + \frac{13}{36} \lambda^2 + \frac{7}{36} \lambda^3 + \frac{3}{36} \lambda^4.$$

The components of $\nu$ expressed in terms of the nodal values $\{U^j, V^j, W^j\}, l = \{1, 2, 3, 4\}$ via formula (A5) are

$$\partial_s U = \sum_{l=1,4} U^j \partial_s L^j, \quad \partial_s V = \sum_{l=1,4} V^j \partial_s L^j,$$

$$\partial_s W = \sum_{l=1,4} W^j \partial_s L^j.$$

Inspecting eqs (A8)-(A10), we recognize that the expressions in the square brackets are the doubles of the Cartesian projections of the area of triangle $\{v_1, v_2, v_3\}$. We can use these formulae for any triangular face within the control volume, substituting the appropriate values for the vertex coordinates, which are found via expressions such as eq. (A20).

Other integrals $J^{r,l}$ for $\alpha = \{x, y, z\}$ and $l = 1, 2, 3$ are evaluated similarly:

$$J^{r,l} = \sum_{q,s} \int \hat{n}_l \lambda (\partial_s x_j + \hat{v}_j) dS^l, \quad \alpha = \{x, y, z\}.$$
The integration over all inner surfaces \( S^{\text{II}} \) of the Earth would only be a Dirichlet boundary for the perturbed components should be removed. However, a node on the surface of the Earth would only be a Dirichlet boundary for \( \Phi^{1} \), so all four components should be removed. However, a node on the surface of the Earth would only be a Dirichlet boundary for the perturbed gravitational potential. As a result, there will be either three or four components associated with one node in the final vector of unknowns.

Discretized equation for the corresponding solution component makes a row in the system matrix with the row entries being the coefficients of the nodal values of the displacement, \( \{ U, V, W \} \), and the perturbed potential, \( \Phi^{1} \). On average, each node is connected to 12 neighbours in our tetrahedral grid, therefore a typical number of non-zero entries is approximately 48 per row. We adopt the compressed sparse row (CSR) format to store the matrix, which requires all the non-zero matrix coefficients be appended to each other on a row-by-row basis.

For a given matrix \( A \), if \( n_{\text{row}} \) is the number of rows (row dimension) and \( n_{\text{nz}} \) is the number of non-zero matrix coefficients (contained in all rows), the CSR format requires one real and two integer arrays to store the matrix: \( a (1 : n_{\text{nz}}) \), \( ia (1 : n_{\text{row}} + 1) \) and \( ja (1 : n_{\text{nz}}) \), where array \( a \) holds the matrix coefficients, array \( ia \) contains pointers to the beginning of each row in \( a \) and array \( ja \) contains the column indices for each entry in array \( a \). As an example, the number of entries in row \( l \) is \( ia (l + 1) - ia (l) \) for \( l = 1, \ldots, n_{\text{row}} \). It is customary to generate a pointer to a non-existing row such as \( ia (n_{\text{row}} + 1) \equiv n_{\text{nz}} + 1 \). In the code, the matrix assembly is carried out directly in the CSR format. Apart from the obvious memory savings, this storage scheme also renders a fast execution of the matrix–vector product.

After the assembly process is completed, the matrix is scaled to improve the condition number. For each row, all entries are normalized relative to the value of the diagonal term with a negative sign, so that all diagonal terms become equal to \(-1\). This procedure is used in place of the more traditional dimension-independentization. Note that the scaling factors must be stored in memory in order to make proper adjustments to the RHS vector of the system.

Let us now turn our attention to assembling the RHS of eq. (18). There are three contributions to the RHS vector \( b \): Dirichlet nodes, external surface loads and viscous memory terms. The true Dirichlet nodes are handled during the LHS assembly. We identify Dirichlet nodes connected to the node of interest, e.g. \( l \), multiply the appropriate matrix coefficients by the Dirichlet value at those nodes and add the result to \( b_{l} \) with the opposite sign.

Recall that the original momentum eq. (1) contains \( \nabla \cdot T \). This term, dotted with the constant vector \( \mathbf{\hat{n}} \), where \( \alpha = \{ x, y, z \} \), and integrated over the volume, produces

\[
\int (\nabla \cdot T) \cdot \mathbf{\hat{n}} d\Omega = \int \mathbf{\hat{n}} \cdot T \cdot \mathbf{\hat{n}} dS.
\]  \hspace{1cm} (A31)

The integration over all inner surfaces \( S^{\text{I}} \) has been performed during the LHS assembly, therefore we only need to consider element surfaces of type \( S^{\text{II}} \) and then only those that belong to the domain surface \( S \). Substituting for \( \mathbf{\hat{n}} \cdot T \) from the boundary condition (13), we obtain

\[
\int_{S} (-\sigma \nabla \Phi^{0}) \cdot \mathbf{\hat{n}} dS = \int_{S^{\text{II}}} (-\sigma \nabla \Phi^{0}) \cdot \mathbf{\hat{n}}^{\text{II}} dS^{\text{II}}. \hspace{1cm} (A32)
\]

Therefore, to account for the external surface loads, we loop over \( S \) and consider triangles of type \( S^{\text{II}} \) on which \( \sigma \neq 0 \). It is assumed that both \( \nabla \Phi^{0} \) and \( \sigma \) are linear functions within the triangles, which makes the kernel a quadratic function. The second-order quadrature rule then applies, where the sampling points are taken at the mid-edges of the triangles, as shown in Fig. 9(a). The surface mass density \( \sigma \) must be available at each time step as a function of position and time.

Let us now consider the viscous memory term. Because the RHS of eq. (9) is a perfect divergence and \( \mathbf{\hat{n}} \) is a constant vector for \( \alpha = \{ x, y, z \} \), the control volume integration is straightforward:

\[
\xi^{\alpha} = \int \mathbf{\hat{n}} \cdot (\nabla \cdot \mathbf{\xi}) d\Omega^{\text{CV}} = \int \hat{v}_{\xi_{\alpha}} \mathbf{\hat{n}}^{\alpha} dS^{\text{II}}. \hspace{1cm} (A33)
\]

The explicit expressions are

\[
\xi^{\alpha} = \left\{ \begin{array}{ll}
\int (\xi_{xx} \hat{v}_{x}^{2} + \xi_{yy} \hat{v}_{y}^{2} + \xi_{zz} \hat{v}_{z}^{2}) dS \quad \text{for } \alpha = x, \\
\int (\xi_{xy} \hat{v}_{x} \hat{v}_{y} + \xi_{xz} \hat{v}_{x} \hat{v}_{z}) dS \quad \text{for } \alpha = y, \\
\int (\xi_{yz} \hat{v}_{y} \hat{v}_{z} + \xi_{zx} \hat{v}_{x} \hat{v}_{z}) dS \quad \text{for } \alpha = z.
\end{array} \right.
\]  \hspace{1cm} (A34)

Because the tensor \( \mathbf{\xi} \) is symmetric, we need to store and update only six (out of nine) independent components \( \{ \xi_{xx}, \xi_{yy}, \xi_{zz}, \xi_{xy}, \xi_{yz}, \xi_{xz}, \xi_{xz}, \xi_{xx}, \xi_{zz} \} \) at each quadrature point within an element. The amount of computer memory to be used for that storage may be large and strongly depends on the type and order of the quadrature rule. It follows from formula (8) that the kernels in integrals (A34) are proportional to \( \mu \). Strictly speaking, regular polynomials cannot approximate this product under the assumption that both the shear modulus and the viscosity are linear functions within the element, because \( k = \mu \Delta t / \eta \). A reasonable compromise may be achieved if we assume that \( k \) and \( \mu \) are linear within the element, which effectively means that \( \eta \) is forced to assume a constant value within the element. The integration in formula (A34) is carried out over the surfaces of type \( S^{\text{II}} \) only. Note that \( \xi \) is a part of the full stress tensor \( T \), whose integrals cancel out exactly on the inner surfaces \( S^{\text{II}} \), while the integrals over the boundary surfaces \( S^{\text{II}} \) have been included as the loading terms in eq. (A32).

The storage scheme for \( \mathbf{\xi} \) is grid-element based. In Fig. 2(b), we show six triangular surfaces \( S^{\text{II}}, l = [1, \ldots, 6] \), which form the boundary of \( \Omega^{\text{CV}} \) constructed for vertex \( v^{1} \). If the other three control volumes for vertices \( v^{2}, v^{3} \) and \( v^{4} \) are constructed in a similar fashion, we end up with a total of 12 independent triangular surfaces \( S^{\text{II}}, l = [1, \ldots, 12] \). For clarity, we show them in Figs A9(a)–(d) as four planar layouts, corresponding to surfaces \( S^{\text{II}} \) within the element, constructed for vertices \( v^{1} \) to \( v^{4} \), respectively. Each layout contains six triangles whose vertices involve the mid-edges, mid-faces and the centre of the element. The notational convention follows that of expressions (17) and is extended to nodes \( v^{1}, v^{3} \) and \( v^{4} \). The sampling quadrature points for the second-order rule are located on the mid-edges of \( S^{\text{II}} \). There are 22 such independent quadrature points per element, labelled as integers inside the large, opaque circles in Fig. A9. Therefore, to store the tensor \( \mathbf{\xi} \) within one grid element, we need a real vector of length \( 132 = 6 \times 22 \). Occasionally, it may be sufficient for some applications to sample \( \mathbf{\xi} \) in the centres of \( S^{\text{II}}, l = [1, \ldots, 12] \), which produces almost identical results, but reduces the length of the storage vector to \( 72 = 6 \times 12 \).
A3 Setting up the communication table

The local equations are set up for all nodes except the external nodes. Let us define communication nodes as the union of all local overlapping and boundary nodes. Clearly, both the external and communication nodes are involved in interprocessor communication. To set up addresses for the respective MPI buffers, we use the notion of the global number, which acts as a unique identifier for any node regardless of which subdomain(s) it belongs to. The global numbering, performed by the parallel grid generator, is apparently a non-unique operation, because it is only required that all nodes in the entire modeling domain be labelled continuously from 1 to $N$ (where $N$ is the total number of grid nodes in all subdomains) and the relevant part of this information be available locally. Now we launch a three-step algorithm. First, we set up the receiving buffers by broadcasting integer arrays of global numbers, corresponding to local communication nodes in a loop over all the CPUs from 0 to $np - 1$, where $n_p$ is the number of the available CPUs. Other CPUs look for matching global numbers among their local communication and external nodes. Step two of the algorithm is to broadcast the content of the receiving buffers in a loop from 0 to $np - 1$, so that other CPUs identify the nodes to be sent among local communication nodes, which match the buffer nodes. The redundancy built in this algorithm acts as a filter, so that in the end, for a given CPU, we obtain a list of neighboring processors ranging from 1 to $n_l \leq n_p$, plus $n_l$ arrays for the receiving buffer and $n_l$ arrays for the sending buffer. All elements in those buffer arrays are grid nodes in the global numbers. The number of neighboring CPUs must be kept to minimum as it has a major impact on the MPI performance, especially with a 100-MB Ethernet connection on the network. Typically, we have $n_l \sim 6–8$ for our grid. Note that we implicitly rely on the communication symmetry property; that is, if CPU $i$ sends a message to CPU $j$, then CPU $j$ must also send a message to CPU $i$. However, the lengths of these messages may be different. The last step of the algorithm is to ensure a proper correspondence between what is sent and what is received. Because the global numbers are invariant with respect to the CPUs, we simply sort all entries in both sending and receiving buffer arrays in ascending order and, finally, replace all global numbers by their local counterparts. At this point, all global information is no longer needed.

Figure A9. Layouts for the quadrature points corresponding to the second-order Gaussian rule sampled on all 12 inner triangular surfaces $S_l^i, l = [1, \ldots, 12]$ inside an element $\{u^1, v^2, v^3, v^4\}$. Layouts (a) to (d) correspond to the four triangulated cups that belong to the surfaces of the control volumes constructed for vertices $u^1$ to $v^4$, respectively. Each cup is a 3-D surface consisting of six triangles that have point $c_e$ in common. For simplicity all cups are drawn flat. There is a total of 22 quadrature points labelled from 1 to 22 inside large opaque circles. Note that the markup of layout (a) for the staggered nodes is in accord with that of Fig. 2(b).