ROYAL HOLLOWAY UNIVERSITY OF LONDON

DOCTORAL THESIS

3-D spherical high-resolution modelling of South Atlantic rifting-related mantle flow

Author:
Jorge M. TARAMÓN GÓMEZ

Supervisors:
Prof. Jason P. MORGAN
Dr. Jörg HASENCLEVER
Prof. Marta PÉREZ-GUSSINYÉ

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

August 10, 2018

COMPASS
Department of Earth Sciences
Royal Holloway University of London
Egham, United Kingdom

Declaration of Authorship

I, Jorge M. TARAMÓN GÓMEZ hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed: ______________________________

Date: ______________________________

"The most beautiful experience we can have is the mysterious. It is the fundamental emotion that stands at the cradle of true art and true science. Whoever does not know it and can no longer wonder, no longer marvel, is as good as dead, and his eyes are dimmed."

Albert Einstein

Abstract

The treatment of far-field boundary conditions (BCs) is one of the most poorly resolved issues for regional modelling of geodynamic processes. The mantle velocity field, along the side-walls and base of a modelling region, is typically much more poorly known than the geometry of past global motions of the surface plates, as constrained by global plate motion reconstructions. In this thesis, I have developed numerical tools that allow the study of 3-D spherical regional models with no fictitious internal boundaries using an embedded high resolution region within a global spherical coarse mesh.

A key piece in any numerical simulation using the Finite Element Method (FEM) is the mesh. I have developed an algorithm to generate high-quality unstructured meshes with embedded high resolution regions within 2-D and 3-D Cartesian, 2-D cylindrical and 3-D spherical shell domains. The mesh nodes are treated as if they were linked by virtual springs and the FEM is used to solve iteratively for the optimal nodal positions for the static equilibrium of this spring system. A 'guide-mesh' is incorporated to easily define preferred element sizes throughout the mesh.

A new technique, the 'Double Jacobian', is presented for more accurate solution in cylindrical or spherical geometries. This approach combines the advantages of working simultaneously in both Cartesian and polar or spherical coordinates. On the one hand, the governing matrix equations are kept in Cartesian coordinates, preserving their symmetry. On the other hand, the element geometry is described in 'straight-sided' polar or spherical coordinates, preserving the appropriate curved boundary surfaces and interfaces. These 'straight-sided' polar or spherical elements allow search routines to rapidly find arbitrary points in polar or spherical coordinates.

The tools described above have been applied to study the influence of the Tristan da Cunha plume during the early rifting and break-up of the South Atlantic. Global plate motion BCs are applied through time using GPlates. Models show a migration of hotter and weaker plume material towards the rifting region before the break-up, influenced by the lateral thickness variations in the initial structure of the lithosphere. Once the plume material reaches the rifting region, it is found to preferentially migrate southwards. This migration appears to be due to the presence of thicker São Francisco and conjugate Congo cratonic roots in the North combined with a ridge 'suction' force due to stretching of non-cratonic lithosphere in the South. This mechanism could explain the observed preferential southward formation of early-rifting-related Seaward Dipping Reflectors (SDRs) along South Atlantic margins with respect to their Tristan Plume progenitor.

Acknowledgements

First, I would like to acknowledge Jason P. Morgan for his guidance over the last four years of this project. He was always willing to interesting discussions and gave me invaluable advice. I would also like to thank him the opportunity he gave me to keep working on numerical modelling.

I would also like to acknowledge Jörg Hasenclever. During the week I spent at GEOMAR and the week he visited RHUL he provided a boost in my project through enlightening discussions about coding. Also thank to him for his helpful reviews of the papers.

I am grateful to Marta Pérez Gussinyé for her valuable comments on the third paper. I would also like to acknowledge Saswata Hier-Majumder for his helpful feedback during the annual reviews and upgrade meeting. Also for giving me the opportunity to demonstrate with him. Thanks to COMPASS group for the financial support and especially to Jürgen Adam. Thanks also to Kevin D'Souza and Frank Lehane for being always helpful.

My gratitude to Ana M. Negredo and Juan Rodríguez González. They opened the door of numerical modelling.

On a more personal level I would like to acknowledge my parents for their incommensurable love and support, no matter the distance. Thanks for always believe in me, I would have not gotten that far and would not become the person I am without you.

Thanks to Miguel, Elena, Silvia, Albert, Arnaud, Dami, Camilla, Nathaniel, Pablo and María for sharing so many good moments both inside and outside the university. Without you the last four years would have been really boring.

Thank also to "Staff football", especially to Sameer, Vinay, Eugenio, Pedro, Kevin and Iñaki. I immensely enjoyed playing football with you as well as talking about the highlights of the game we had just played while having a pint in a pub.

I would like to acknowledge the support and encouragement from my friends I left in Spain. Thanks to Javi, Fati, Jose, Óscar, Mick, Susi, Pablo, Zule, Lucía, Fran and Judit. You always made me feel special every time I came back to Madrid.

Lastly and more important, thanks to Iria for her endless support, patience and love.

Contents

List of Figures

List of Tables

To my parents

Chapter 1

Introduction

The modern view of the Earth's dynamics is based on plate tectonics, where large rigid lithospheric plates move slowly relative to each other over tens of millions of years [Wilson, 1963; McKenzie and Parker, 1967; Morgan, 1968]. There are three types of plate boundary that depend on the relative motion between plates: (1) convergent, where one plate subducts below another one; (2) divergent, where two plates spread apart creating new oceanic crust; and (3) transform, where two plates move parallel to the plate boundary in opposite directions. The extension and stretching of continental lithosphere is known as rifting. If rifting continues over time, it may lead eventually to continental break-up and the creation of a new divergent margin. This type of margin is also known as a passive margin and is often host to economic petroleum systems. The slow rate of motion, the temperature of the source rock, and the deposition of sediments are important factors to create the necessary conditions to potentially produce hydrocarbon reservoirs. These processes are all linked to the rifting process itself, which remains poorly understood. Depending on the amount of magmatism, passive margins can be considered non-volcanic or volcanic, also known as magma-poor and magma-rich margins, respectively (see Figure 1.1 for the global distribution of passive margins).

1.1 Volcanic rifted margins and mantle plumes

Volcanic rifted margins are associated with syn-rift magmatism that occurs during the last stage of continental break-up and the beginning of sea-floor spreading. Diagnostic features of volcanic rifted margins are offshore Seaward Dipping Reflectors (SDRs), a high-velocity lower crust, and an initially thicker oceanic crust [e.g. Mutter et al., 1982; White et al., 1987; Gladczenko et al., 1997; Franke, 2013]. SDRs are thick wedges of volcanic flows that have been widely accepted to be a clear indicator of a volcanic margin. Volcanic rifted margins are usually associated with continental flood basalts, in which tremendous volumes of basalts are extruded within a short geological time interval [Morgan, 1971; White and McKenzie, 1989; Coffin and Eldholm, 1994; Courtillot et al., 1999]. Flood basalt emplacement is often thought to be related to the arrival of a mantle plume to lithospheric depths [Morgan, 1971]. Flood basalts are usually associated with

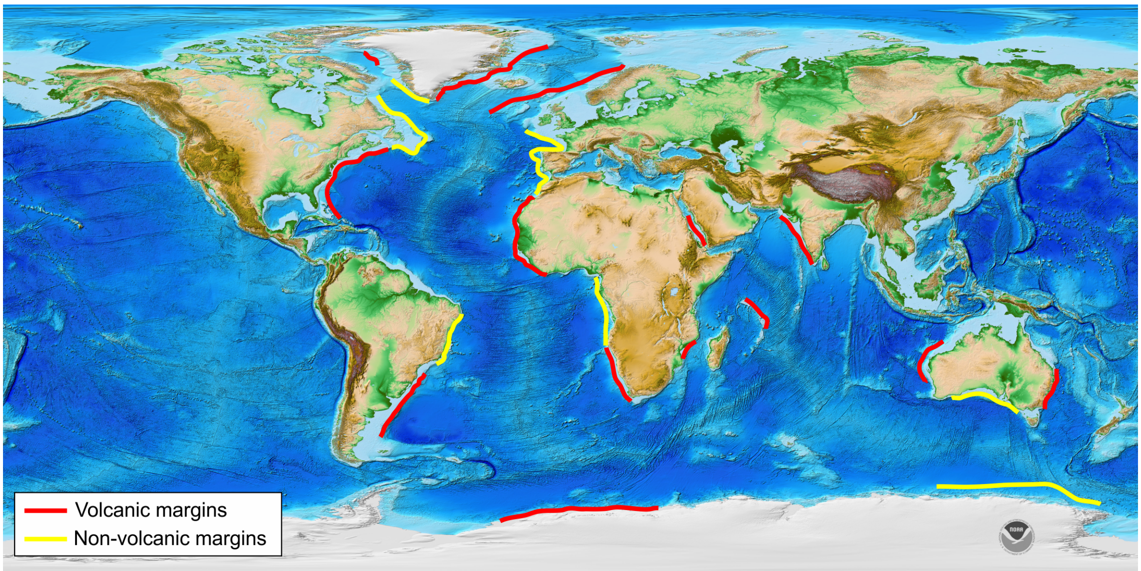


FIGURE 1.1: Global distribution of passive margins in a topographic map from ETOPO1 [Amante and Eakins, 2009]. Type of passive margin data from Geoffroy [2005], Blaich et al. [2011], Andres-Martinez [2016], and Taposeea [2017].

continental break-up and volcanic rifted margins. For example in the Atlantic Ocean, the North Atlantic Igneous Province (NAIP) is associated with the conjugate margins East Greenland-Norway and East Greenland-Great Britain, the Central Atlantic Magmatic Province (CAMP) is associated with the conjugate margins East USA-NW Africa, and the Parana-Etendeka flood basalt is associated with the conjugate margins SE South America-SW Africa. In the Indian Ocean, the Karoo flood basalt is associated with the conjugate margins Africa-Madagascar and South Africa-East Antarctica, the Madagascar flood basalt is associated with conjugate margins Madagascar-Seychelles and the Deccan Traps are associated with the conjugate margins Seychelles-India. However, not all flood basalts need to be associated with continental break-up, for example the Siberian Traps.

The magmatism observed at volcanic rifted margins requires an anomalously warmer mantle than in the case of magma-poor margins [White and McKenzie, 1989]. In the North Atlantic for example, the proximity between the Iceland hotspot and the volcanic rifted margins of East Greenland-Norway and East Greenland-Great Britain led to the idea of mantle plumes as the origin of this local mantle temperature increase. Mantle plumes, considered to be thermal anomalies, are usually thought to arise from the core-mantle boundary carrying warm low density material that should eventually pond at the base of the lithosphere to create a volcanic chain as the plate moves over the relatively fixed plume [Morgan, 1971]. Near a ridge axis, plume material might drain towards regions of thinner lithosphere producing melting by decompression of the asthenosphere [Morgan et al., 1995; Sleep, 1996].

The volcanic conjugate rifted margins in the North Atlantic have been interpreted to show the 'classic' plume-head scenario in which SDRs are distributed symmetrically

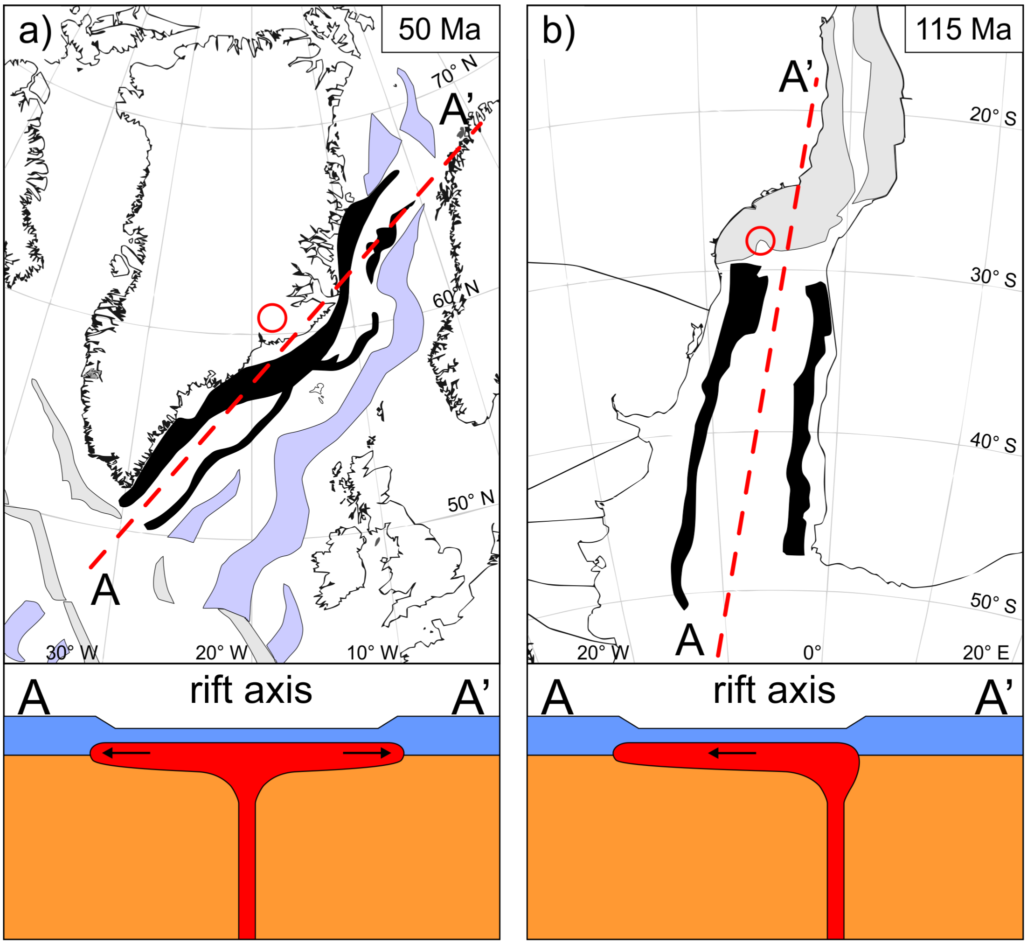


FIGURE 1.2: Differences between (a) symmetrical plume-head scenario (reconstruction of the North Atlantic at 50 Ma) and (b) asymetric lateral plume-flow scenario (reconstruction of the South Atlantic at 115 Ma). Black colour shows the position of SDRs (volcanic margins) and grey and purple colours represent non-volcanic margins and non-volcanic basins, respectively. Structures for (a) from Lundin and Doré [2011] and for (b) from Gladczenko et al. [1997], Moulin et al. [2010], Blaich et al. [2011], and Stica et al. [2014]. Reconstructions made using GPlates and plate kinematic reconstructions given by Gurnis et al. [2012]. Red circle represents the location of the mantle plume. Bottom insets show along-rift axis sketches for each scenario.

along the ridge with respect to their Iceland mantle plume progenitor (Figure 1.2a). However, there are other examples where the interaction between mantle plumes and volcanic rifted margins does not fit this exact pattern. In the South Atlantic there is a sharp transition between volcanic rifted margins south of the Rio Grande Rise and Walvis Ridge and non-volcanic rifted margins in the central part of the South Atlantic (Figure 1.2b). This asymmetry has kept controversial the role of the Tristan da Cuhna plume in the continental break-up of the South Atlantic since it departs from the conventional 'symmetrical' plume-head scenario.

1.2 Numerical modelling

The exponential growth of computational capabilities during the last two decades has allowed numerical modelling to become an important tool to study the processes of the Earth's interior in regions where direct observations are not possible. Numerical modelling of geodynamic processes allows us to test different hypothesis about their applicability to the Earth. For example in 2-D, numerical models have been applied to explore the role of the opening speed during rifting, showing that it controls the symmetry of the margin and mantle exhumation [Pérez-Gussinyé et al., 2006; Brune et al., 2014] or suggesting a strength-velocity feedback that leads to a plate acceleration before break-up [Brune et al., 2016]. Numerical models in 3-D can study, for example, the interaction between a mantle plume and cratonic lithosphere varying the position of the plume beneath the craton and the rheology of the lithosphere [Koptev et al., 2016].

A key piece in any numerical simulation is the mesh that divides the continuum domain into a discrete assemble of elements in order to solve the Partial Differential Equations (PDEs) using the Finite Element Method (FEM). A similar but more regular mesh is used in Finite Difference based approaches . Mesh generation and refinement become essential to create good quality meshes in which well shaped elements improve the performance of iterative methods like the conjugate gradient method [Shewchuk, 2002]. One of the many applications of FEM is its ability to use a formulation to generate unstructured meshes on which a thermo-mechanical finite element calculation can be made by solving a set of equations arising from the conservation of mass, momentum and energy.

In this thesis I have carried out several 3-D spherical numerical modelling experiments in order to answer some questions related to the uncertainties that surround the role of the Tristan da Cunha plume in the initial rifting and break-up of the South Atlantic such as: can we explain the along-ridge asymmetry of SDRs in a plume related flow scenario?, and how much could sublithospheric topography influence near-surface plume flow? In order to do these experiments, I constructed and explored 3-D flow and melting models. The choice of far-field boundary conditions in planetary viscous flow problems often strongly shapes the large-scale flow structure in this kind of geosimulation. The treatment of far-field boundary conditions remains one of the most poorly resolved issues for regional modelling of geodynamic processes, since the mantle velocity field along the side-walls and base of a model region is typically poorly known. I improve on previous models by using global models in which the better known geometry of past global motions of the surface plates, as constrained by fairly well-agreed upon global plate motion reconstructions, is used as the surface boundary condition. Throughout this thesis work, I have developed numerical tools that allow the study of this type of 3-D spherical regional model that contains no fictitious internal boundaries by using an embedded high resolution region within a global spherical coarse mesh.

1.3 Thesis outline

Chapter 2 describes an algorithm to generate high-quality meshes with embedded high resolution regions within 2-D and 3-D cartesian, 2-D cylindrical and 3-D spherical shell domains. The mesh nodes are treated as if they were linked by virtual springs and the FEM is used to solve iteratively for the optimal nodal positions for the static equilibrium of system. A 'guide-mesh' approach is implemented to easily define preferred element sizes throughout the mesh. The algorithm also includes routines to locally improve the quality of the mesh and avoid poorly shaped 'sliver-like' tetrahedra.

Chapter 3 presents a new technique, the 'Double Jacobian', for the improved solution of finite element problems in cylindrical or spherical geometries. This approach combines the advantages of working simultaneously in both Cartesian and polar or spherical coordinates. On the one hand, the governing matrix equations are still formulated in Cartesian coordinates, thereby preserving their simplest (symmetric) algebraic form. On the other hand, the element geometry is described in 'straight/linearly varying' polar or spherical coordinates that preserve the appropriate curved boundary surfaces and cylindrical/spherical internal interfaces. These 'linear-edge' polar or spherical elements allow search routines to rapidly find arbitrary points in the resulting triangle or tetrahedron in polar or spherical coordinates, respectively. The Double Jacobian consists in computing the local to Cartesian mapping as a two-stage process: (1) from local to polar/spherical coordinates (and back), where the mapping is to a straight-edged polar triangle or spherical tetrahedron. In these cases the Jacobian partial derivatives are constant within the element leading to a straightforward analytical matrix expressions. (2) A second analytical mapping from polar/spherical to Cartesian coordinates (and back). The net Jacobian from local coordinates to a cylindrical or spherical element in Cartesian geometry is simply the matrix product of these two easy-to-compute inverse Jacobian matrices.

Chapter 4 studies the influence of the Tristan da Cunha plume in the break-up of the South Atlantic using the tools described in Chapter 2 and Chapter 3, solving for the evolution of thermo-mechanical mantle flow in this region. Models differ in varying the plume flux, the initial position of the plume, and whether to include melting. Model results suggest a possible explanation for the observed early-rifting-related SDRs along South Atlantic margins within a southward plume flow scenario.

Chapter 5 presents a critical evaluation and future prospects.

Chapter 6 presents a final overview of the results.

Chapter 2

Generation of unstructured meshes in 2-D, 3-D, and spherical geometries with embedded high resolution sub-regions

Jorge M. Taramón, Jason P. Morgan, Chao Shi, Jörg Hasenclever. Generation of unstructured meshes in 2-D, 3-D, and spherical geometries with embedded high resolution sub-regions. Manuscript planned for submission at Computers & Geosciences, 2018a.

Authors contribution

JMT and JPM designed the research. JMT, CS, JPM and JH programmed the 2-D rectangular mesh generator. JMT programmed the 2-D cylindrical mesh generator and 3-D spherical mesh generator in discussion with JPM. JMT designed the tests and analysed the results in discussion with JPM. JMT wrote the manuscript in collaboration with JPM and JH.

Received ; | Revised ; | Accepted

DOI: xxx/xxxx

ARTICLE TYPE

Generation of unstructured meshes in 2-D, 3-D, and spherical geometries with embedded high resolution sub-regions

Jorge M. Taramón*[1] | Jason P. Morgan[1,2] | Chao Shi[2] | Jörg Hasenclever[3]

[1]Department of Earth Sciences, Royal Holloway University of London, Egham, Surrey, UK
[2]EAS Dept., Cornell University, Ithaca, NY, USA
[3]Institute of Geophysics, Hamburg University, Hamburg, Germany

Correspondence
*Jorge M. Taramón, Department of Earth Sciences, Royal Holloway University of London Egham, Surrey TW20 0EX, United Kingdom. Email: jorge.taramongomez.2014@live.rhul.ac.uk

Present Address
Department of Earth Sciences, Royal Holloway University of London Egham, Surrey TW20 0EX, United Kingdom

Abstract

We present 2-D, 3-D, and spherical mesh generators for the Finite Element Method (FEM) using triangular and tetrahedral elements. The mesh nodes are treated as if they were linked by virtual springs that obey Hooke's law. Given the desired length for the springs, the FEM is used to solve for the optimal nodal positions for the static equilibrium of this spring system. A 'guide-mesh' approach allows the user to create embedded high resolution sub-regions within a coarser mesh. The method converges rapidly. For example, the algorithm is able to refine within a few iterations a specific region embedded in an unstructured tetrahedral spherical shell so that the edge-length factor $l_{0r}/l_{0c} = 1/33$ where l_{0r} and l_{0c} are the desired spring length for elements inside the refined and coarse regions respectively. One application for this type of mesh is in regional numerical models, where a high-resolution region can be embedded in a global mesh, thereby avoiding fictitious domain boundaries at only a small additional computational cost. The algorithm also includes routines to locally improve the quality of the mesh and to avoid ill-shaped 'sliver-like' tetrahedra.

KEYWORDS:
Optimization, Finite Element Method, Adaptivity, Mesh generation

1 | INTRODUCTION

Mesh generation and (adaptive) refinement are essential ingredients for computational modelling in various scientific and industrial fields. A particular design metric or goal is the quality of the generated mesh, because low-quality meshes can potentially lead to larger numerical approximation errors. A high-quality mesh would consist of triangles (in 2-D) or tetrahedra (in 3-D) that have aspect ratios near 1, i.e. their sides should have similar lengths. The techniques to generate meshes can be crudely classified into three groups: (1) The advancing front method [*Löhner and Parikh*, 1988; *Schöberl*, 1997; *Choi et al.*, 2003; *Ito et al.*, 2004] starts from the boundary of the domain. New elements are created one-by-one from an existing front of elements towards the interior until the region is filled. Advancing front methods generally create high-quality meshes close to the domain boundaries but can have difficulties in regions where advancing fronts merge. (2) Octree-based methods [*Mitchell and Vavasis*, 1992; *Labelle and Shewchuk*, 2007; *Ito et al.*, 2009] produce graded meshes through recursive subdivision of the domain. The simplicity of these methods makes them very efficient. However, poorly shaped elements can be introduced near region boundaries. (3) Delaunay Triangulation ensures that the circumcircle/circumsphere associated to each triangle/tetrahedron does not contain any other point in its interior. This feature makes Delaunay-based methods [*Chew*, 1989; *Ruppert*, 1995; *Chew*, 1997; *Shewchuk*, 1998] robust and efficient. However, in 3-D they can generate very poorly shaped tetrahedra with four almost coplanar vertex nodes. These so-called 'sliver' elements have a volume near zero. Several techniques to remove slivers have been proposed [*Cheng et al.*, 2000; *Li and Teng*, 2001; *Cheng and Dey*, 2002] although some slivers near the boundaries can typically persist [*Edelsbrunner and Guoy*, 2002]. Any 'good' mesh should be able to meet the following requirements [*Bern et al.*, 1994]: (1) It conforms to the boundary; (2) It is fine enough in those regions where the problem to be solved demands higher accuracy; (3) Its total number of elements is as small as possible to reduce the size of the problem and the computational costs to solve it; (4) It has well-shaped elements to improve the performance of iterative methods such as the conjugate gradient method [*Shewchuk*, 2002].

Current mesh generation algorithms oriented to engineering such as Netgen [*Schöberl*, 1997], GiD (https://www.gidhome.com) or TetGen [*Si*, 2015] are based on the methods described above. Variational methods [*Alliez et al.*, 2005] rely on energy minimization to optimize the mesh during the generation procedure in order to create higher-quality meshes. A widely used open access community-code for 2-D mesh generation is Triangle [*Shewchuk*, 1996]. DistMesh [*Persson and Strang*, 2004] is an elegant and simple spring-based method that allows the user to create 2D and 3D unstructured meshes based on the distance from any point to the boundary of the domain. However this algorithm is often slow, requiring many steps to converge,

Frequently used mesh generators in 3-D geodynamic problems are the ones included in the ASPECT [*Kronbichler et al.*, 2012], Rhea [*Burstedde et al.*, 2008] and Fluidity [*Davies et al.*, 2011] codes. ASPECT and Rhea are written in C++ with adaptive mesh refinement (AMR). However their regular hexahedral elements create so-called "hanging nodes" in regions where the resolution changes and cannot be directly applied to create well-formed tetrahedral elements. Fluidity is another example of AMR for a tetrahedral mesh. However it has very limited mesh generation capabilities, and in this context mesh-generation should not be confused with mesh adaptivity.

Here we present a new unstructured mesh generator that is based on a finite element implementation of the DistMesh approach using virtual springs between nodes and solving for the static equilibrium positions of the nodes. We have modified the Distmesh solution procedure to directly solve for static equilibrium, making it considerably faster than the DistMesh code. It also allows the user to create tetrahedral meshes without hanging nodes. The user can also create embedded high resolution sub-regions within a much coarser mesh. This approach becomes very useful when the goal is to create a mesh that minimizes the number of fictitious internal boundaries within a computational problem. Throughout the algorithm, a built-in MATLAB (http://www.mathworks.com) 'delaunay' function is called to generate the spring connectivity matrix that relates nodes to triangles or tetrahedra. We have developed and tested techniques for adding or rejecting nodes in regions where the mesh resolution is too high or too low respectively. A smooth variation in the element size between high resolution and low resolution regions is achieved by using a guide-mesh approach. These local operations improve the quality of poorly shaped elements that can potentially result from the ficticious spring algorithm to determine good nodal locations. The mesh-generation code is written in vectorized MATLAB and can be easily used within the MATLAB working environment.

The motivation to build this computational tool was twofold: (1) For our research we wished to perform numerical experiments on 3-D spherical shell meshes with embedded high resolution regions, and found no available open source code that could readily and efficiently construct this type of mesh; (2) We wished to eventually add the capability to adaptively remesh high-resolution regions during a time-dependent solution, and preferred to have an easy-to-modify mesh-generation code that we could further customize for this task.

We will present this approach first in its simplest form for making a mesh in a well-defined rectangular 2-D region (Section 2). In Section 3 we show how a 2-D cylindrical annulus mesh can be generated with small modifications to the previous rectangular mesh generator algorithm. In Section 4 we present the modifications needed to create the 3-D spherical shell mesh that we are using to solve for mantle flow.

2 | 2-D RECTANGULAR WORK FLOW

This mesh generation algorithm has its simplest form as a program to create a 2-D rectangular mesh with an embedded high resolution sub-region. The white and yellow boxes in Figure 1 show the flowchart that describes this algorithm.

Step 1: Definition of preferred nodal distances and initial placement of the nodes

The first step in this recipe is to define the preferred nodal distances within the refined (l_{0r}) and coarse (l_{0c}) regions as well as the dimensions of the regions. In order to avoid poor quality elements, an appropriate smooth transition for the mesh refinement should be specified. Here we choose a preferred spring-length function that is defined on a so-called 'guide-mesh'. This approach

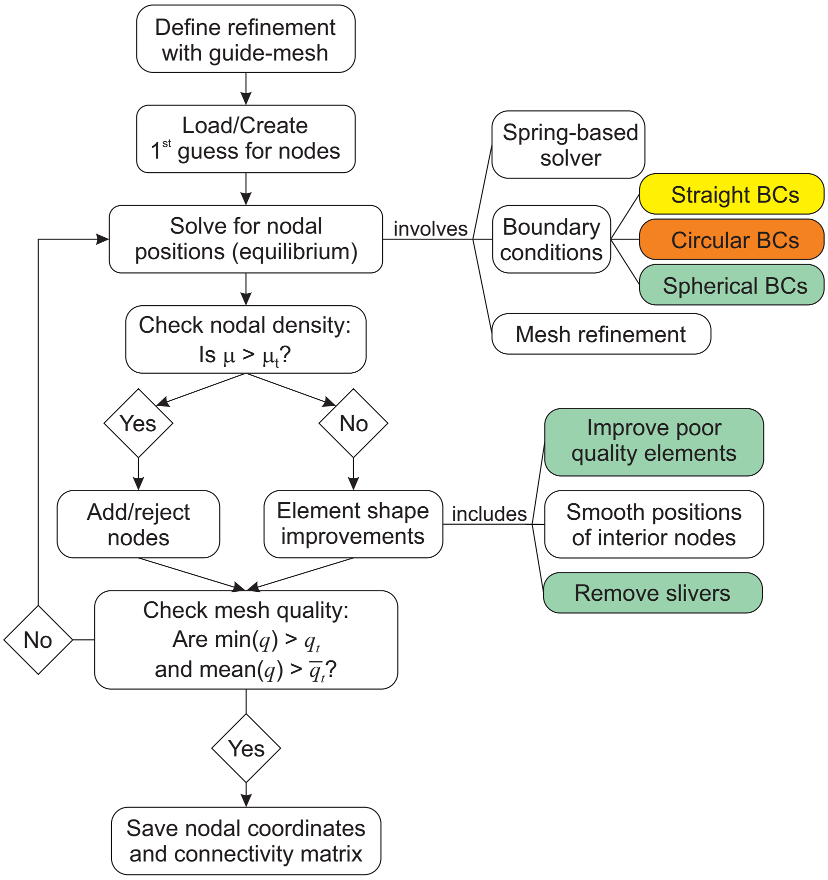

Define refinement with guide-mesh
Load/Create 1st guess for nodes
Solve for nodal positions (equilibrium)
involves
Spring-based solver
Boundary conditions
Straight BCs
Circular BCs
Spherical BCs
Mesh refinement
Check nodal density: Is μ > μt?
Yes
No
Add/reject nodes
Element shape improvements
includes
Improve poor quality elements
Smooth positions of interior nodes
Remove slivers
Check mesh quality: Are min(q) > qt and mean(q) > q̄t?
No
Yes
Save nodal coordinates and connectivity matrix

FIGURE 1 Flow chart for the mesh generator iterative process. Yellow, orange and green boxes represent the routines exclusively used for creating 2-D rectangular meshes, 2-D cylindrical annulus meshes and 3-D spherical shell meshes, respectively. White boxes represent the shared routines to all mesh generators. μ is the mean of the misfit spring lengths (equation (16)) and q is the quality factor of the elements (equations (13) and (31) for triangular and tetrahedral elements respectively). Tolerance parameters μ_t, q_t and $\bar{q}_t$ are listed in Table 1.

is very similar to the background grid approach created by *Löhner and Parikh* [1988]. The generation of a refined rectangular mesh using the guide-mesh approach involves the following steps. First, create a (coarse) mesh to serve as a guide-mesh with only a small number of nodes defining the boundaries of the domain and the internal boundaries of the embedded high resolution and transition sub-regions. Second, create the design function $l_0(x, y)$ for each node of the guide-mesh. This function defines the desired length for the springs around those points. Third, the function $l_0(x, y)$ is evaluated at the midpoint of all springs using linear Finite Element shape functions. We find that a coarse guide-mesh is a simple and flexible way to control nodal spacing during the generation of a Finite Element mesh. Figure 2a shows the guide-mesh for a rectangular mesh example whose parameters are listed in Table 1. Red and blue dots represent nodes in the guide-mesh with defined l_{0r} and l_{0c}, respectively. The red region represents the refined region of the mesh with spring length approximately equal to l_{0r}. The green region defines the transition region where the length of the springs smoothly varies from l_{0r} to l_{0c}. The blue region represents the coarse region of the mesh with a approximate spring length of l_{0c}.

The next step is to create a starting guess for the locations of the nodes. Computational work is reduced considerably with a good initial guess for the density of the nodes. Nodes on the boundary and within the domain are created taking into account both the location of the refined region and the desired springs length for elements inside the refined and coarse regions. Boundary nodes in the refined and coarse regions are created using l_{0r} and l_{0c} respectively for the spacing between the nodes. The interior nodes within the refined and coarse regions are created using a circle packing lattice with radius equal to $l_{0r}/2$ and $l_{0c}/2$ respectively. This fills each region with an equilateral triangular tiling. In the transition region the size of the elements is expected to change smoothly between l_{0r} and l_{0c}. The initial placement for boundary and interior nodes in the transition region is created using l_{0r} as explained above. After this step, the rejection method described in *Persson and Strang* [2004] is used to discard

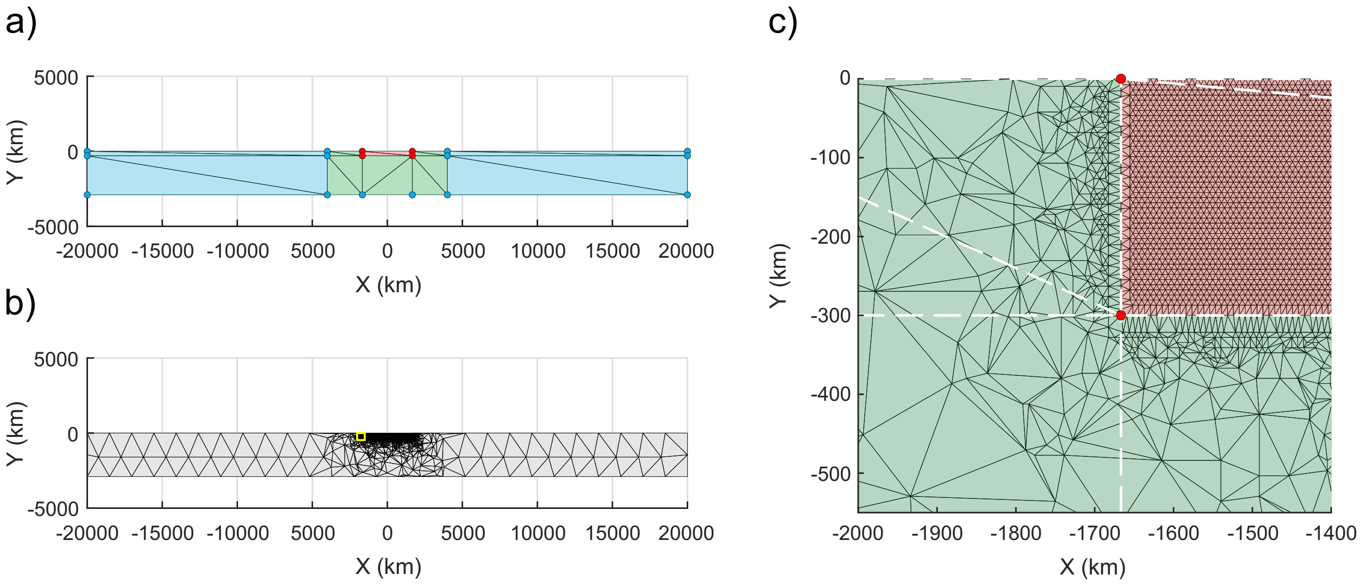


FIGURE 2 (a) Guide-mesh defined by a few nodes in Cartesian coordinates for a rectangular mesh. The parameters for this mesh are listed in Table 1. Each node is assigned a value for the desired spring length, being l_{0r} for red dots and l_{0c} for blue dots. The length of the springs within the refined region (in red) is approximately equal to l_{0r}. The length of the springs within the transition region (in green) varies smoothly from l_{0r} to l_{0c}. The length of springs within the coarse region (in blue) is approximately equal to l_{0c}. (b) Initial guess for the rectangular mesh. (c) Zoom around the left boundary of the refined region for the initial guess (yellow line in (b)). The guide-mesh defining refined (red) and transition (green) regions is shown in white dashed lines.

points and create a 'balanced' intitial distribution of nodes. After performing a Delaunay triangulation, a quasi-regular mesh of triangles within the refined and coarse regions, with a poorly structured transition region between them is created (Figure 2b). Figure 2c shows a zoom of the initial mesh with the guide-mesh also shown.

Step 2: Spring-based solver

Inspired by *Persson and Strang* [2004], to generate an unstructured mesh we link the future locations of finite element nodes with virtual elastic springs. The spring length is used to define the desired nodal distance within any mesh region, i.e. short springs lead to mesh regions with higher resolution and longer springs lead to lower resolution mesh regions. Nodal positions are solved for so that the global network of virtual springs is in static equilibrium. The behaviour of each ficticious spring is described by Hooke's law

$$F = -k\delta s\,, \tag{1}$$

where F is the force acting at each end of spring, k is the stiffness of the spring, and δs is the distance the spring is stretched or compressed from its equilibrium length l_0. Forces and nodal positions are expressed in x, y coordinates in 2-D (Figure 3a). Because Hooke's law is formulated along the spring direction it is necessary to introduce the X' axis as the local 1-D reference system to solve for the nodal positions. Hooke's law for each spring in the local 1-D reference system is given by

$$f_1{}' = k\delta s = k(x_2{}' - x_1{}' - l_0)\,, \tag{2a}$$

$$f_2{}' = -k\delta s = -k(x_2{}' - x_1{}' - l_0)\,, \tag{2b}$$

where f' and x' are the force and position of the ends of the spring given by the subscripts 1 and 2, respectively. Writing equations (2a) and (2b) in matrix form, and moving the force terms to the left hand side yields

$$\begin{pmatrix} f_1{}' \\ f_2{}' \end{pmatrix} + k\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\begin{pmatrix} 0 \\ l_0 \end{pmatrix} = k\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\begin{pmatrix} x_1{}' \\ x_2{}' \end{pmatrix}\,. \tag{3}$$

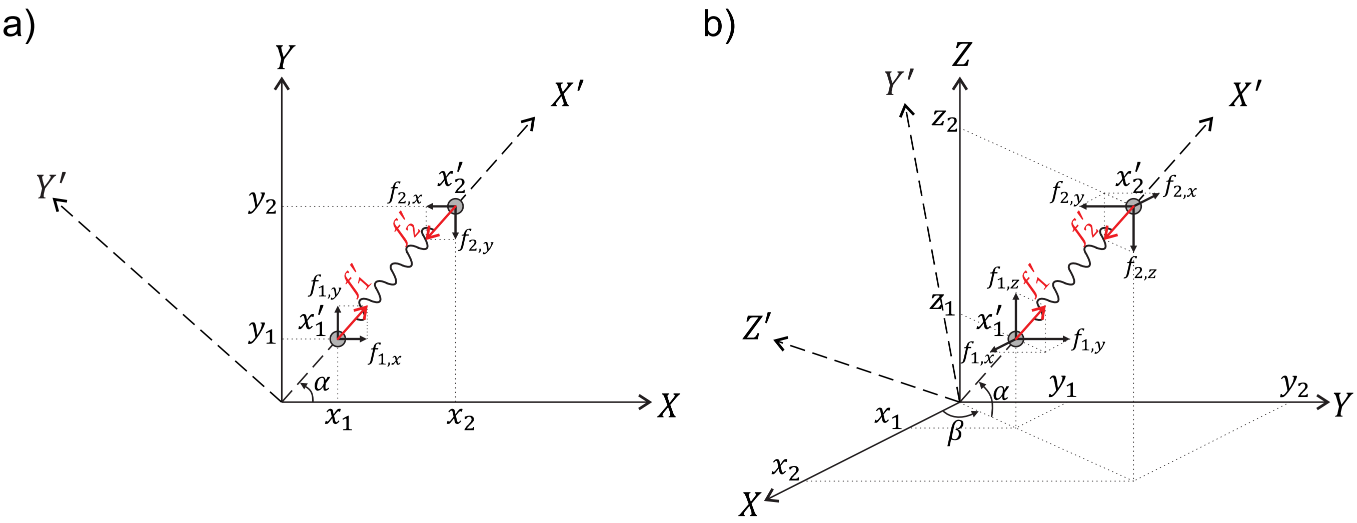


FIGURE 3 (a) Virtual spring in the 2-D space. Both global reference system (X, Y) and local reference system (X', Y') are shown. (b) Virtual spring in the 3-D space. Both global reference system (X, Y, Z) and local reference system (X', Y', Z') are shown. Grey dots represent two nodes linked by the virtual spring. Red arrows represent the forces acting at each end of the spring.

In order to solve for the nodal positions in 2-D, a change from local coordinates ($x_1{}'$, 0; $x_2{}'$, 0) to global coordinates (x_1, y_1; x_2, y_2) is needed. This change of coordinates is described in matrix form as

$$\boldsymbol{R_{2D}} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ 0 & 0 & \cos\alpha & \sin\alpha \end{bmatrix}, \tag{4}$$

where α is the angle of the X' axis measured from the X axis in the counterclockwise direction (Figure 3a). Applying equation (4) to equation (3) (see Appendix A for further details), equation (3) becomes

$$k \begin{bmatrix} -c_\alpha{}^2 & -s_\alpha c_\alpha & c_\alpha{}^2 & s_\alpha c_\alpha \\ -s_\alpha c_\alpha & -s_\alpha{}^2 & s_\alpha c_\alpha & s_\alpha{}^2 \\ c_\alpha{}^2 & s_\alpha c_\alpha & -c_\alpha{}^2 & -s_\alpha c_\alpha \\ s_\alpha c_\alpha & s_\alpha{}^2 & -s_\alpha c_\alpha & -s_\alpha{}^2 \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} f_{1,x} \\ f_{1,y} \\ f_{2,x} \\ f_{2,y} \end{pmatrix} + kl_0 \begin{pmatrix} c_\alpha \\ s_\alpha \\ -c_\alpha \\ -s_\alpha \end{pmatrix}, \tag{5}$$

where $s_\alpha \equiv \sin\alpha$ and $c_\alpha \equiv \cos\alpha$. Equation (5) can be written in the matrix form as

$$\boldsymbol{Kx} = \boldsymbol{f_r} + \boldsymbol{f_{l_0}}, \tag{6}$$

where $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{x}$ is the nodal displacement vector, $\boldsymbol{f_r}$ is the residual force and $\boldsymbol{f_{l_0}}$ is the force-term created by the fact that the springs would have zero-force at their desired length. Whwn solved for the equilibrium state, $\boldsymbol{f_r} = 0$. A vectorized 'blocking' technique based on the MATLAB methodology described in the MILAMIN code [*Dabrowski et al.*, 2008] is employed to speed up the assembly of the stiffness matrix. The solution to this problem is the 'optimal' position of each node obtained from the inversion of the system of static force equilibrium equations

$$\boldsymbol{x} = \boldsymbol{K}^{-1}\boldsymbol{f_{l_0}}. \tag{7}$$

Straight line Boundary Conditions

Boundary conditions are necessary to constrain the mesh to the desired domain boundaries, and to differentiate between boundary and interior nodes. In the simple case of a rectangular mesh, a boundary node is free to slide along a domain edges parallel to the X- or Y-axis. We achieve this by setting one of its y_i or x_i values to be fixed and letting the other value vary so that the node is free to move along the boundary segment. In the case of a general line that is not parallel to the X- or Y-axes, this requires a transformation from global coordinates to a new local coordinate system in which the constraint direction is parallel to a local coordinate axis. In other words, the new local axes have to be parallel to and perpendicular to the boundary segment. For simplicity, the mathematical implementation is shown for one triangle (Figure 4). Node 2 is free to slide along the tilted segment

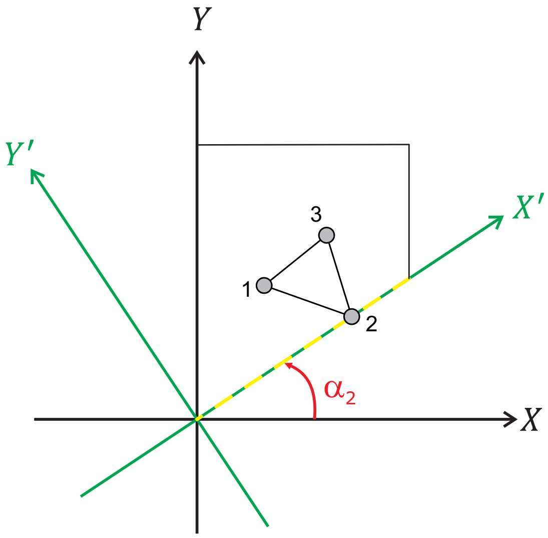


FIGURE 4 Implementation of boundary conditions along a straight tilted segment (yellow dashed line) for one triangle. A rotation is needed for the node 2 in order to pass from the global reference system (X, Y) to the local reference system (X', Y') where $y_2{}' = 0$ is the constrained boundary condition.

(yellow dashed line in Figure 4) since $y_2{}' = 0$ defines the boundary constraint. The boundary condition is imposed by a rotation of coordinate system for node 2 given by the transformation matrix $\boldsymbol{T}$ that relates global coordinates $\boldsymbol{x}$ to local coordinates $\boldsymbol{x}'$ by

$$\underbrace{\begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \end{pmatrix}}_{\boldsymbol{x}} = \underbrace{\begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \cos\alpha_2 & -\sin\alpha_2 & & \\ & & \sin\alpha_2 & \cos\alpha_2 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}}_{\boldsymbol{T}} \underbrace{\begin{pmatrix} x_1 \\ y_1 \\ x_2{}' \\ 0 \\ x_3 \\ y_3 \end{pmatrix}}_{\boldsymbol{x}'} . \tag{8}$$

Applying the transformation matrix to the stiffness matrix and force vector

$$\boldsymbol{K}' = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{T} , \tag{9}$$

$$\boldsymbol{f}_{l_0}{}' = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{f}_{l_0} , \tag{10}$$

the new system of equations is given by

$$\boldsymbol{K}'\boldsymbol{x}' = \boldsymbol{f}_{l_0}{}' , \tag{11}$$

which is solved for $\boldsymbol{x}'$. When desired, the original global coordinates are recovered through the transformation matrix

$$\boldsymbol{x} = \boldsymbol{T}\boldsymbol{x}' . \tag{12}$$

Step 3: Mesh refinement

In this algorithm we refine a mesh by decreasing the element size in the region of interest. One common issue in the refinement process arises from the size contrast between large and small elements within a short spatial interval so that poorly-shaped elements with short and long edges may form. In order to mitigate this issue a transition region surrounding the refined region is defined using the guide-mesh approach described above (see Figure 2a).

Quality factor for triangles

The ’quality’ of a mesh is determined by assessing the quality of its individual elements. This usually involves measures of angles, edge lengths, areas (in 2-D), volumes (in 3-D), or the radius of its inscribed and circumscribed circles/spheres [e.g.

Dompierre et al., 1998; *Shewchuk*, 2002]. Here we use a normalized quality factor, which in 2-D is given by

$$q_{_{2D}} = \frac{2r_c}{R_c}\,, \tag{13}$$

where r_c is the radius of the element's inscribed circle and R_c is the radius of its circumscribed circle. R_c and r_c can be expressed as

$$r_c = \frac{1}{2}\sqrt{\frac{(b+c-a)(c+a-b)(a+b-c)}{a+b+c}}\,, \tag{14}$$

$$R_c = \frac{abc}{\sqrt{(a+b+c)(b+c-a)(c+a-b)(a+b-c)}}\,, \tag{15}$$

where a, b and c are the side lengths of the triangle. A fair criteria to evaluate the quality of a mesh is to provide the minimum and mean values of the quality factor [cf. *Alliez et al.*, 2005]. Here both are used as control parameters to determine when the iterative algorithm has reached the desired mesh quality tolerances (Figure 1).

Step 4: Local mesh improvements

So far the above algorithm would only move nodes within the domain to meet the desired spring lengths/internodal distances. However, in general we do not know a priori how many nodes are needed for a mesh. Therefore we use algorithms to locally add and remove nodes where the spacing is too loose or tight in the equilibrium configuration. After solving for nodal positions, we check if the mesh has reached the expected nodal density by determining the mean of the misfit in spring lengths (Figure 1). This is given by

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{l_i - l_{0i}}{l_{0i}}\right|\,, \tag{16}$$

where l is the actual spring length, l_0 is the desired spring length and N is the total number of springs in the mesh. Nodes are added or rejected (see below) if $\mu \geq \mu_t$. When $\mu < \mu_t$ the expected nodal density is achieved and element shape improvements (see below) are applied to obtain higher quality elements. After some experimentation we found it appropriate to use $0.02 < \mu_t < 0.05$ for 2-D meshes.

Add/reject nodes

In the iterative process of mesh generation the possibility to either add or reject nodes plays an important local role. This feature is especially relevant when the goal is to create a global coarse mesh with an embedded high resolution sub-region. The logic for adding or rejecting nodes is based on the relative length change of the springs connecting nodes

$$\epsilon = \frac{l - l_0}{l_0}\,, \tag{17}$$

indicating whether springs are stretched ($\epsilon > 0$) or compressed ($\epsilon < 0$) with respect to their desired lengths. A new node is created at the midpoint of those springs with $\epsilon > 0.5$, i.e. springs stretched more than 50% greater than their desired length. One node at the end of a spring is rejected when $\epsilon < -0.5$, i.e. springs compressed more than 50% below their desired length. In order to save computational time, the add/reject nodes routine is called as a sub-iteration within the main iteration in which nodal positions are found. Sub-iterations are performed until the percentage of springs with $|\epsilon| > 0.5$ in the sub-iteration $j+1$ is higher than in the sub-iteration j. This implementation is especially useful when a large fraction of nodes need to be either added or rejected within a particular region of the mesh, e.g. when a relatively poor initial guess is used.

Smooth positions of the interior nodes

Good quality meshes are directly related to the generation of isotropic elements [*Alliez et al.*, 2005]. A Laplacian smoothing criteria [cf. *Choi et al.*, 2003] is used to improve the shape of poorly shaped elements, i.e. to make elements as close to a equilateral triangles or regular tetrahedra as possible. This method is only applied to interior nodes. The routine repositions interior nodes towards the mean of the barycentres of their surrounding elements, i.e.

$$\boldsymbol{x}_s = \frac{\sum_{i=1}^{N}\boldsymbol{x}_{bi}}{N}\,, \tag{18}$$

where $\boldsymbol{x}_s$ are the new coordinates of the interior node, N is the number of elements surrounding the interior node and $\boldsymbol{x}_{\boldsymbol{b}_i}$ are the barycentre coordinates of the i-th surrounding element. Figure S1 shows an example of smoothing positions of interior nodes for a 2-D mesh.

Example: Rectangular mesh with an embedded high resolution region

Several tests have been performed with the above implementations in order to demonstrate the robustness of this mesh-generation recipe. As an example, we show the results for a rectangular box with an embedded high-resolution sub-region (code available in Supporting information). The input parameters that control the algorithm are listed in Table 1. The algorithm created the mesh in 9 s (purple dot in Figure 5) after 8 outermost loop iterations (cf. Figure 1). All tests in this study have been performed using MATLAB R2015a (8.5.0.197613) on a 3.2 GHz Intel Core i5 (MacOSX 10.12.5) with 24 GB of 1600 MHz DDR3 memory. Figure 6a shows the final mesh (top) and a zoom around the left boundary of the refined region (bottom) for the iteration 8 (see Figure S2 for iterations 0 (initial mesh) and 1). The final mesh has 22000 nodes forming 43000 triangles with an edge-length factor $l_{0r}/l_{0c} = 1/200$. The percentage of triangles within the coarse, transition and refined regions is 0.3%, 6.3% and 93.4% respectively. The lowest quality factor for an element is 0.51 (red line in Figure 6b) and the mean quality factor for all elements is 0.99 (blue line in Figure 6b). Only 0.12% of the triangles have a quality factor lower than 0.6 (green line in Figure 6b). Figure 6c shows the fraction of elements as a function of quality factor for the final mesh.

TABLE 1 Mesh Parameters.

Symbol	Meaning	Rectangular box	Cylindrical annulus	Spherical shell
d	Depth	2900 km	-	-
l	Length	40000 km	-	-
r_i	Inner radius	-	3471 km	3471 km
r_o	Outer radius	-	6371 km	6371 km
x_0	x-coordinate centre of refined region	0 km	-	-
z_0	z-coordinate centre of refined region	0 km	-	-
θ_0	Colatitude centre of refined region	-	90°	90°
ϕ_0	Longitude centre of refined region	-	-	90°
r_0	Radial distance centre of refined region	-	6371 km	6371 km
l_{0c}	Desired spring length for elements inside the coarse region	1500 km	2000 km	2000 km
l_{0r}	Desired spring length for elements inside the refined region	7.5 km	10 km	60 km
d_t	Transition region depth	2900 km	2900 km	2900 km
l_t	Transition region length	8000 km	8000 km	6800 km
w_t	Transition region width	-	-	9600 km
d_r	Refined region depth	300 km	300 km	300 km
l_r	Refined region length	3333 km	3333 km	2200 km
w_r	Refined region width	-	-	5000 km
q_t	Tolerance for minimum quality factor	0.45	0.30	0.23
$\bar{q}_t$	Tolerance for mean quality factor	0.89	0.93	0.80
μ_t	Tolerance for mean misfit spring length	0.025	0.04	0.13

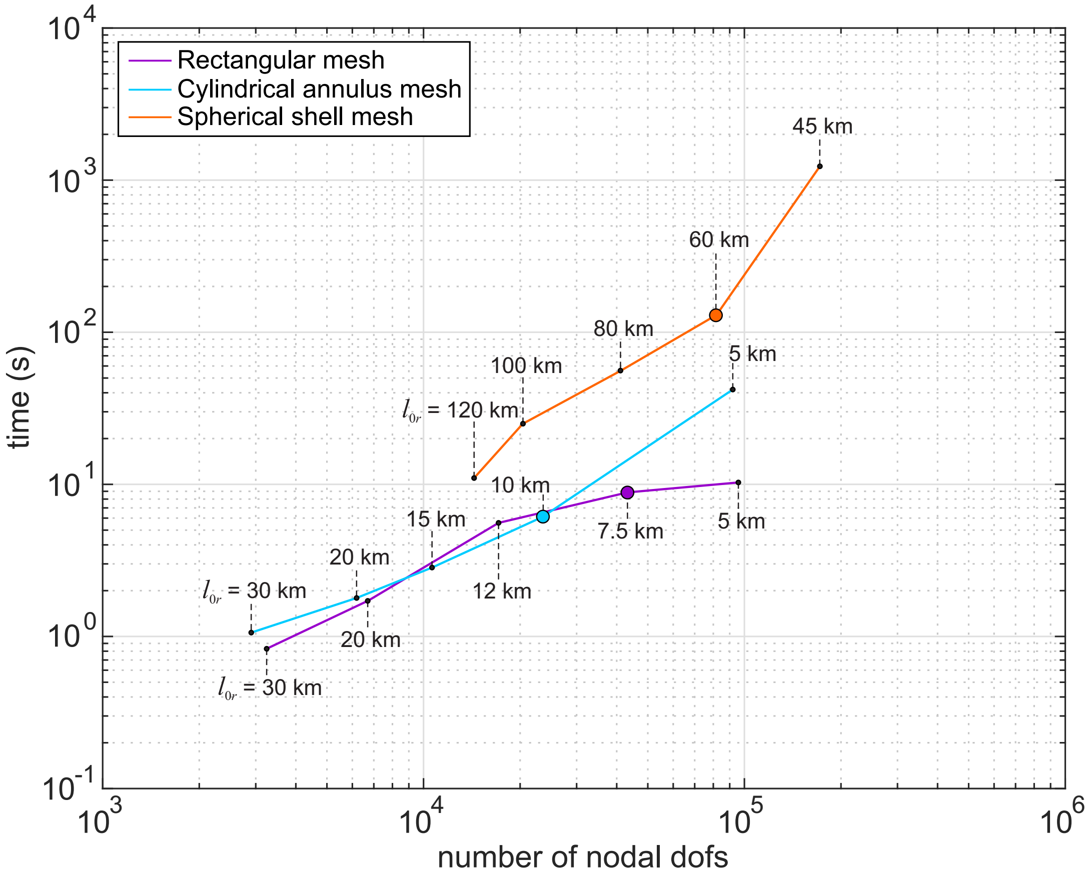


FIGURE 5 Computational time as a function of the number nodal degrees of freedom (dofs). All meshes have been generated using the parameters listed in Table 1; only l_{0r} is varied. The purple dot (7.5 km-rect. mesh) is for the example shown in Figure 6, the blue dot (10 km-cyl. mesh) is for the example shown in Figure 9, and the orange dot (60 km-3D sph. mesh) is for the example shown in Figure 13. The non-linear increase in solution time for the highest-resolution spherical shell mesh occurs because a direct matrix-inversion algorithm is currently used to solve the finite element equations. Changing the current direct solver to a standard iterative FE solver would allow the solution time to continue to scale linearly with the number of mesh-points, while also resulting in an easy-to-parallelize code-tool.

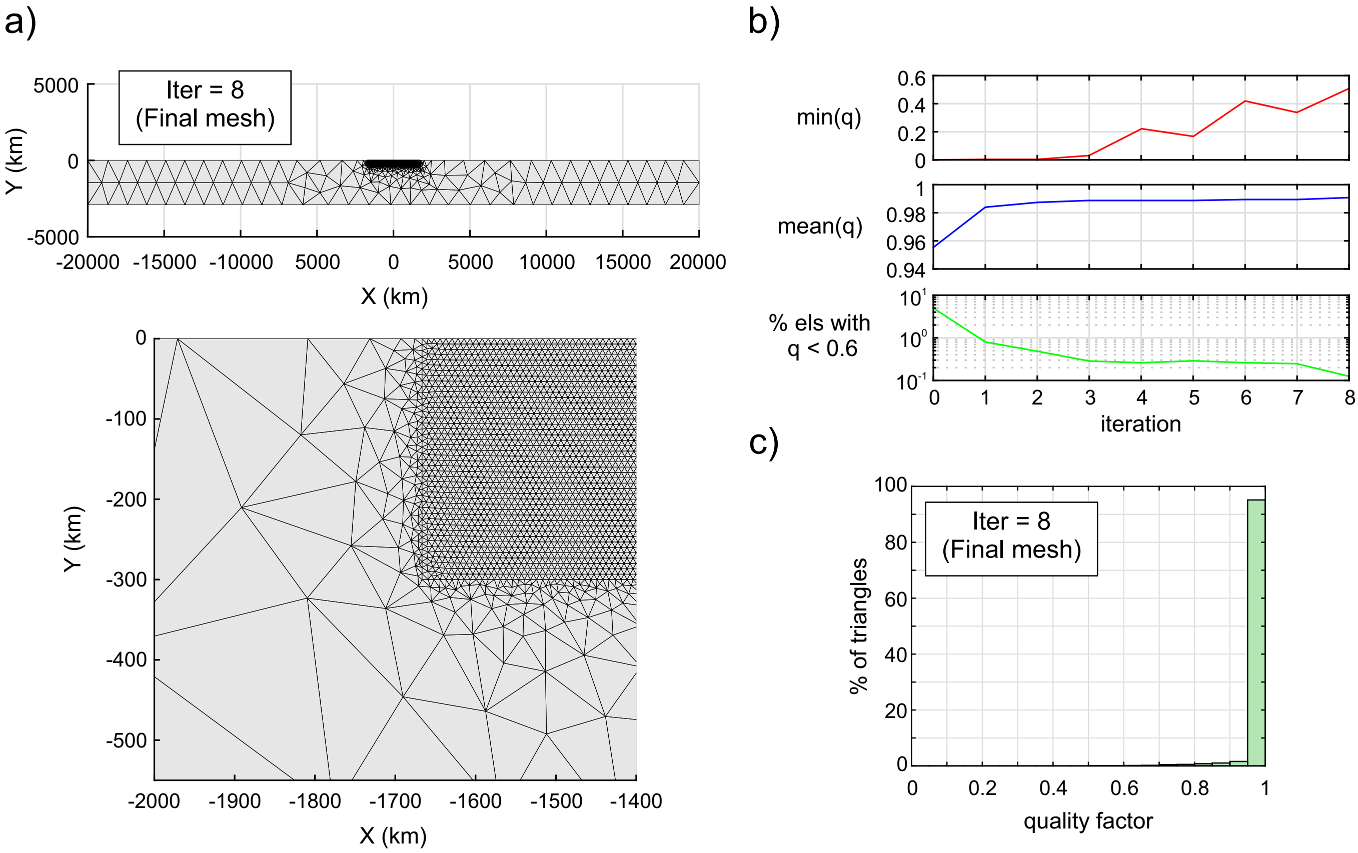


FIGURE 6 (a) Final mesh (top) for a rectangular box with an embedded high resolution sub-region and a zoom around the left boundary of the refined region (bottom). (b) Minimum quality factor (red line), mean quality factor for all elements (blue line) and percentage of elements having a quality factor lower than 0.6% (green line) as a function of iteration number. (c) Histogram of the fraction of elements as a function of quality factor for the final mesh. The lowest quality factor for an element is 0.51.

3 | 2-D CYLINDRICAL ANNULUS WORK FLOW

The algorithm presented above needs to be slightly modified to generate a cylindrical annulus mesh. The white and orange boxes in Figure 1 show the flowchart that describes this modified algorithm. Since the general algorithm is the same, in this section we only discuss the parts that differ from the rectangular mesh generator described previously.

Cylindrical annulus guide-mesh

The generation of a refined cylindrical annulus mesh using the guide-mesh involves the same steps as for a rectangular mesh except that the function $l_0(x, y)$ becomes $l_0(\theta, r)$. In this case the guide-mesh is a coarse cylindrical annulus mesh defined in polar coordinates. Figure 7a shows the guide-mesh (white dashed lines) defining the refined (red), transition (green) and coarse (blue) regions and the parameters are listed in Table 1. Red and blue dots represent l_{0r} and l_{0c} respectively. The initial triangulation is shown in black solid lines. Figure 7c shows a zoom of the guide-mesh defined in polar coordinates. Green dots represent the points where the function $l_0(\theta, r)$ is interpolated. The use of a guide-mesh defined in polar coordinates (white dashed lines in Figure 7a and Figure 7c) instead of Cartesian coordinates (white dashed lines in Figure 7b and Figure 7d) takes advantage of higher precision when l_0 values are interpolated in points both close and on the boundaries (green dots in Figure 7c). This is because the shape of the outer and inner boundaries of any cylindrical annulus mesh defined in Cartesian coordinates is not perfectly circular (Figure 7b). Therefore, it may occur that some boundary points (magenta dots in Figure 7d) may lay outside of the boundaries of a Cartesian guide-mesh (which can be a very coarse mesh) preventing accurate interpolation for the desired length at those points. Furthermore, the fact that both boundaries – the cylindrical annulus mesh and its guide-mesh – would not overlap in a Cartesian geometry would reduce the precision of the interpolated l_0 values (yellow dots in Figure 7d).

Circular Boundary Conditions

Boundary conditions for a cylindrical annulus mesh are a generalization to the treatment for a straight-sided boundary line-segment. We denote the inner and outer boundaries Σ of the cylindrical annulus mesh as radii $r = r_{inner}$ and $r = r_{outer}$ respectively. Ω is the interior region confined between both boundaries. A useful boundary condition is to prescribe nodes on Σ that are free to move along the circular boundary. This nodal motion is generated by two independent steps (Figure 8a): 1) The node is allowed to move along the tangent line to the circle at its current location, and 2) the node is placed onto the circle by projecting its new location in the radial direction. This approximation assumes that the radial distance needed to put the node back onto the circle is small compared to the distance moved along the tangent line. For simplicity, the mathematical implementation is presented here only for one triangle (Figure 8b). The boundary condition for node 2 is that it slides along its tangent line (dashed line in Figure 8b) since $y_2{}' = |r|$, where r is the radial distance from the centre of the cylindrical annulus mesh to the boundary. The boundary condition is imposed by a rotation of the coordinate system for node 2 given by the transformation matrix $\boldsymbol{T}$ that relates global coordinates $\boldsymbol{x}$ with local coordinates $\boldsymbol{x}'$ (local surface-parallel reference system (X', Y') in green in Figure 8b) by

$$\underbrace{\begin{pmatrix} x_1\\ y_1\\ x_2\\ y_2\\ x_3\\ y_3\end{pmatrix}}_{\boldsymbol{x}} = \underbrace{\begin{bmatrix} 1 &&&&& \\ &1&&&& \\ &&\cos\theta_2 & \sin\theta_2 && \\ && -\sin\theta_2 & \cos\theta_2 && \\ &&&&1& \\ &&&&&1\end{bmatrix}}_{\boldsymbol{T}} \underbrace{\begin{pmatrix} x_1\\ y_1\\ x_2{}'\\ |r|\\ x_3\\ y_3\end{pmatrix}}_{\boldsymbol{x}'}, \tag{19}$$

where θ_2 is the angle of the node 2 measured from the Y axis in the clockwise direction. After applying the transformation matrix to the stiffness matrix and force vector

$$\boldsymbol{K}' = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{T}\,, \tag{20}$$

$$\boldsymbol{f}_{\boldsymbol{l}_0}{}' = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{f}_{\boldsymbol{l}_0}\,, \tag{21}$$

the new system of equations is given by

$$\boldsymbol{K}'\boldsymbol{x}' = \boldsymbol{f}_{\boldsymbol{l}_0}{}'\,, \tag{22}$$

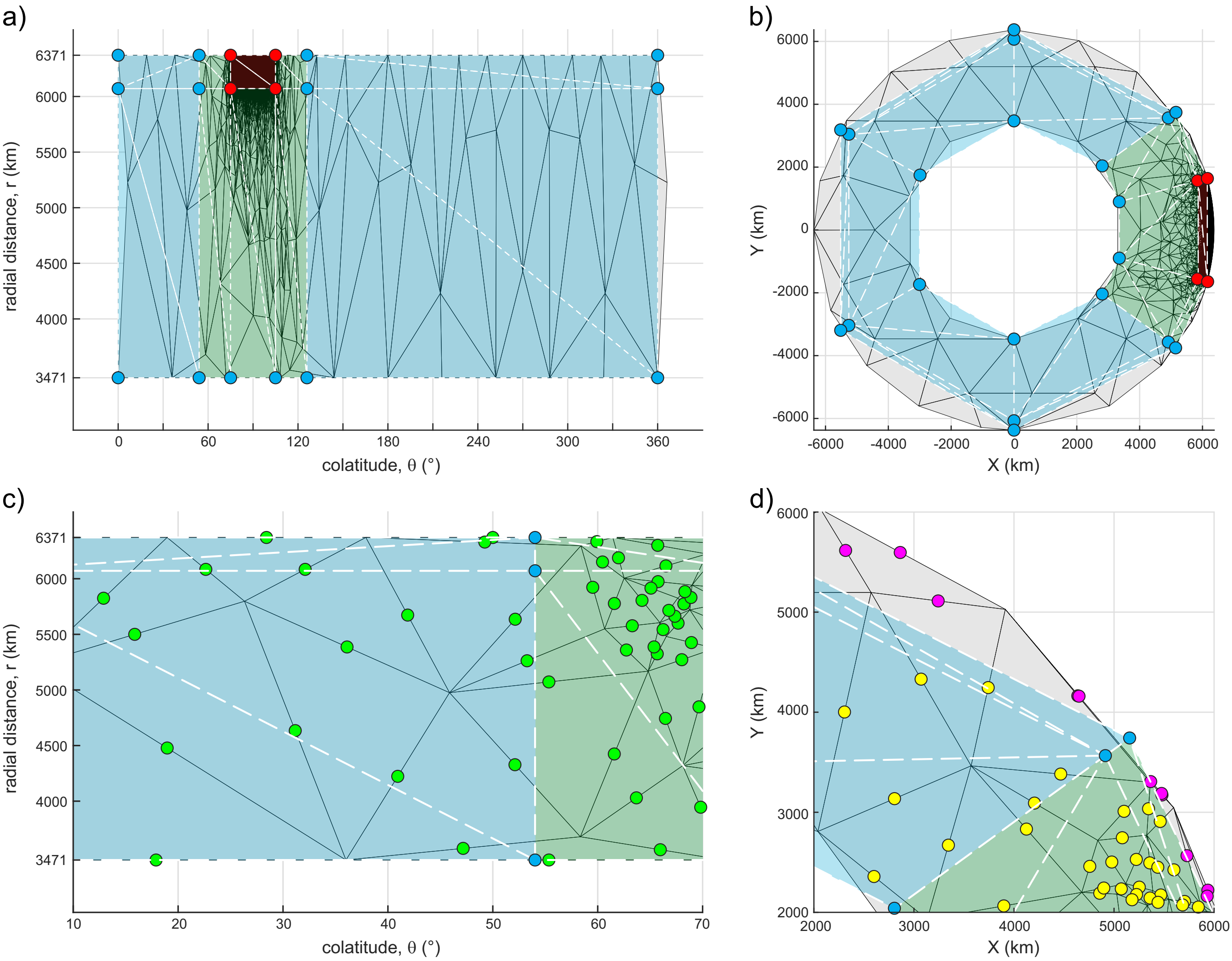


FIGURE 7 (a) Guide-mesh (white dashed lines) defined by a few nodes (red and blue dots represent l_{0r} and l_{0c} respectively) in polar coordinates for a cylindrical annulus mesh (initial guess is shown in black solid lines). Red, green and blue colours represent the refined, transition and coarse regions respectively. (b) Guide-mesh defined in Cartesian coordinates. Same colours as in (a). (c) Zoom around an edge of the transition region in polar coordinates. The function $l_0(\theta, r)$ can be interpolated at green dots with maximum precision since both boundaries – the cylindrical annulus mesh and its guide-mesh – are overlapping. (d) Zoom around an edge of the transition region in Cartesian coordinates. The function $l_0(x, y)$ cannot be interpolated at magenta dots since they lay outside of the outer boundary of a Cartesian guide-mesh. The precision of the interpolated l_0 values at yellow dots is reduced since both boundaries – the cylindrical annulus mesh and its guide-mesh – do not overlap.

which is then solved for $\boldsymbol{x}'$. Global coordinates are recovered through the transformation matrix

$$\boldsymbol{x} = \boldsymbol{T}\boldsymbol{x}' . \tag{23}$$

Add/reject nodes in cylindrical annulus meshes

The routine to add or reject nodes for a cylindrical annulus mesh works like the one explained above for a rectangular mesh. The only difference appears when a new node is added on a boundary spring. In this case, the new boundary node needs to be projected onto the surface along the radial direction.

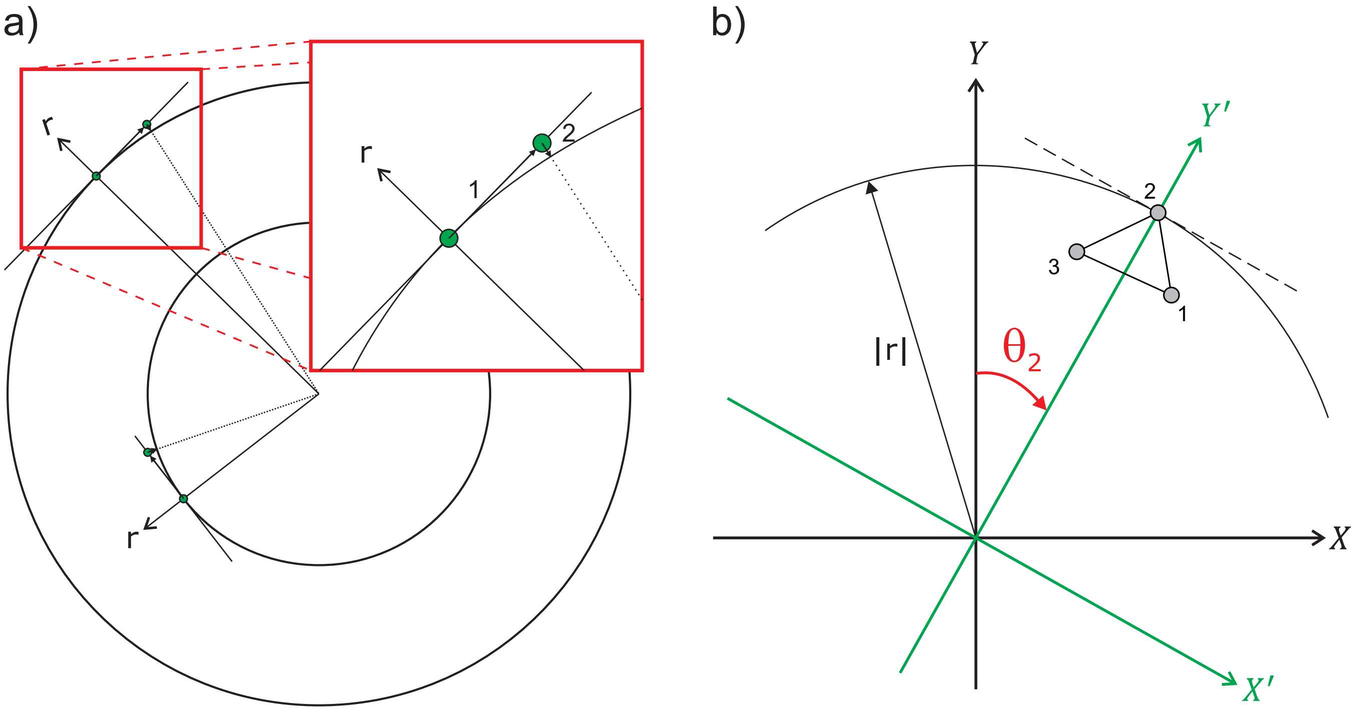


FIGURE 8 (a) Conceptual diagram for circular boundary conditions. The motion of boundary nodes is first restricted to be along the tangent line to the circle. Then they are 'pulled back' to the circle by projecting in the radial direction. (b) Implementation of circular boundary conditions for one triangle. A rotation is needed for the node 2 in order to pass from the global reference system (X, Y) to the local surface-parallel reference system (X', Y') where $y_2{}' = |r|$ is the constrained boundary condition.

Example: Cylindrical annulus mesh with an embedded high resolution region

We show the results for a cylindrical annulus mesh with an embedded high-resolution sub-region (code available in Supporting information). The input generation parameters are listed in Table 1. The algorithm created the mesh in 6 s (blue dot in Figure 5) after 7 iterations. Figure 9a shows the final mesh for iteration 7 (see Figure S3 for iterations 0 (initial mesh) and 1). Figure 9b shows a zoom around an edge of the refined region. The final mesh has 12000 nodes forming 23000 triangular elements with an edge-length factor $l_{0r}/l_{0c} = 1/200$. The percentage of triangles within the coarse, transition and refined regions is 0.2%, 6.1% and 93.7% respectively. The worst quality factor for an element is 0.40 (red line in Figure 9c) and the mean quality factor of all elements is 0.98 (blue line in Figure 9c). Only 0.26% of the triangles have a quality factor lower than 0.6 (green line in Figure 9c). Figure 9d shows the fraction of elements as a function of their quality factor for the final mesh.

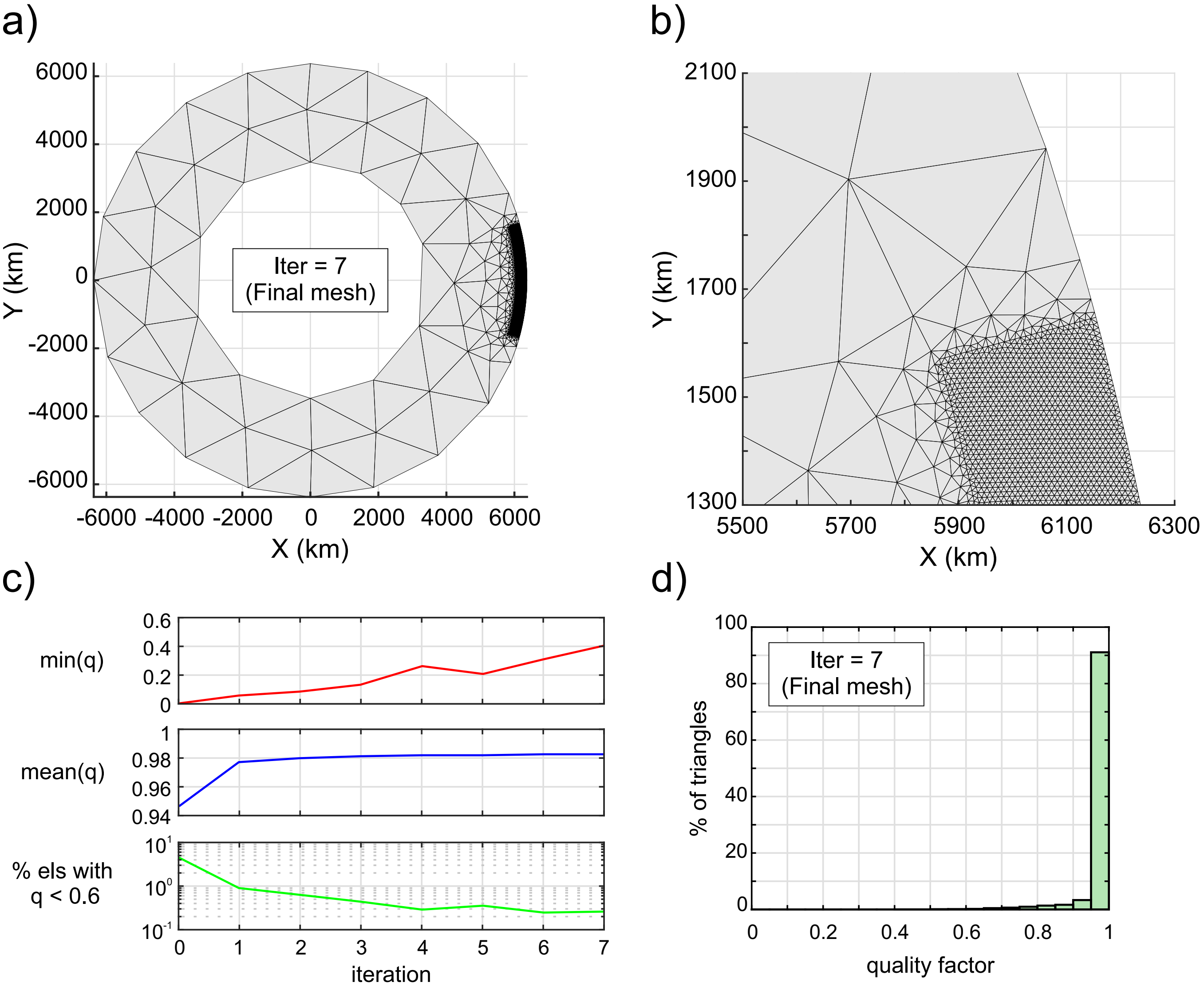


FIGURE 9 (a) Final mesh for a cylindrical annulus with an embedded high resolution sub-region. (b) Zoom around an edge of the refined region. (c) Minimum quality factor (red line), mean quality factor for all elements (blue line) and percentage of elements having a quality factor lower than 0.6% (green line) as a function of iteration number. (d) Histogram of the fraction of elements as a function of quality factor for the final mesh. The lowest quality factor for an element is 0.40.

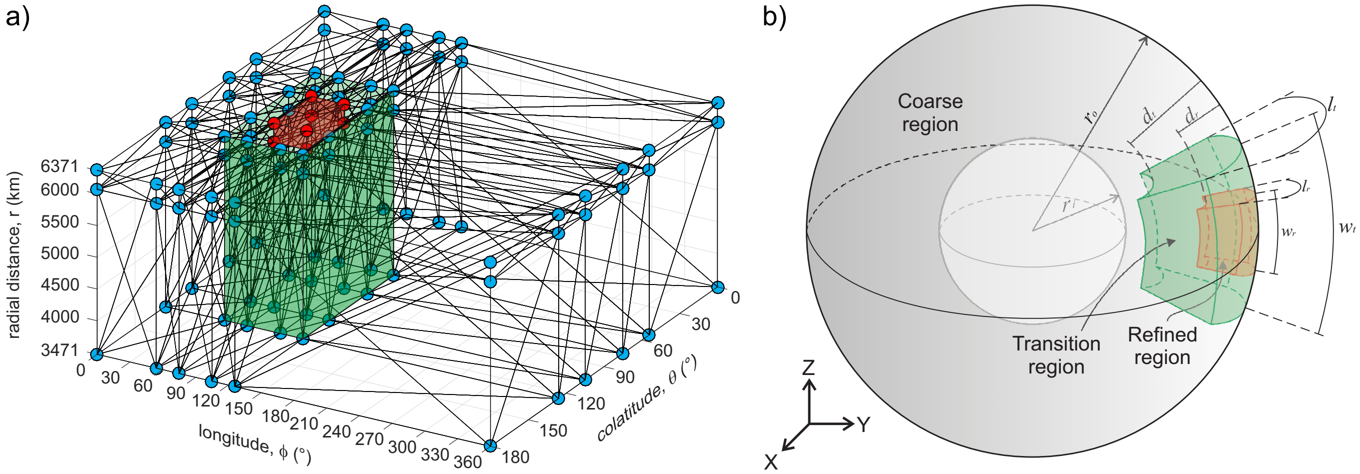


FIGURE 10 (a) Guide-mesh defined by a few nodes (red and blue dots represent l_{0r} and l_{0c} respectively) in spherical coordinates for a spherical shell. The length of the springs within the refined region (red) is approximately equal to l_{0r}. The length of the springs within the transition region (green) smoothly varies from l_{0r} to l_{0c}. Outside the transition region the length of the springs is approximately equal to l_{0c}. (b) Model domain representing a 3-D spherical shell with an embedded high resolution sub-region.

4 | 3-D SPHERICAL SHELL WORK FLOW

The algorithm presented above was developed as an intermediate step towards the generation of 3-D spherical shell meshes that include an embedded high resolution sub-region. The white and green boxes in Figure 1 show the flowchart that describes the 3-D spherical algorithm. In this section we discuss those parts of the algorithm that differ from the cylindrical annulus mesh generator.

Initial placement of the nodes in 3-D

The boundary nodes in the refined and coarse regions are created by recursively splitting an initial dodecahedron according to l_{0r} and l_{0c} respectively. This gives a uniform distribution of equilateral triangles on the spherical surface. In contrast to equilateral triangles in 2-D, which are able to fill up the plane, regular tetrahedra do not fill up the entire space. However, there do exist some compact lattices, e.g. the hexagonal close packing (hcp) lattice, that create a distribution of nodes that leads to well shaped tetrahedra. The interior nodes within the refined and coarse regions are created by a close-packing of equal spheres with radii equal to $l_{0r}/2$ and $l_{0c}/2$ respectively. The initial placement for boundary and interior nodes in the transition region is created using l_{0r} as explained above. Then the rejection method described in *Persson and Strang* [2004] is used to discard points and create a weighted distribution of nodes.

Spherical shell guide-mesh

The generation of a refined spherical shell mesh using the guide-mesh involves steps similar to those described above except that the preferred length function $l_0(\theta, r)$ is now $l_0(\theta, \phi, r)$. In this case the guide-mesh is a coarse spherical shell mesh defined in spherical coordinates (Figure 10a).

Spring-based solver in 3-D

The spring-based solver described above naturally extends to 3-D. Forces and nodal positions are expressed in x, y and z coordinates (Figure 3b). In order to solve for nodal positions in 3-D, a change from local coordinates $(x_1{}', 0, 0; x_2{}', 0, 0)$ to global coordinates $(x_1, y_1, z_1; x_2, y_2, z_2)$ is needed. This change of coordinates consists of a 3-D rotation described by the rotation

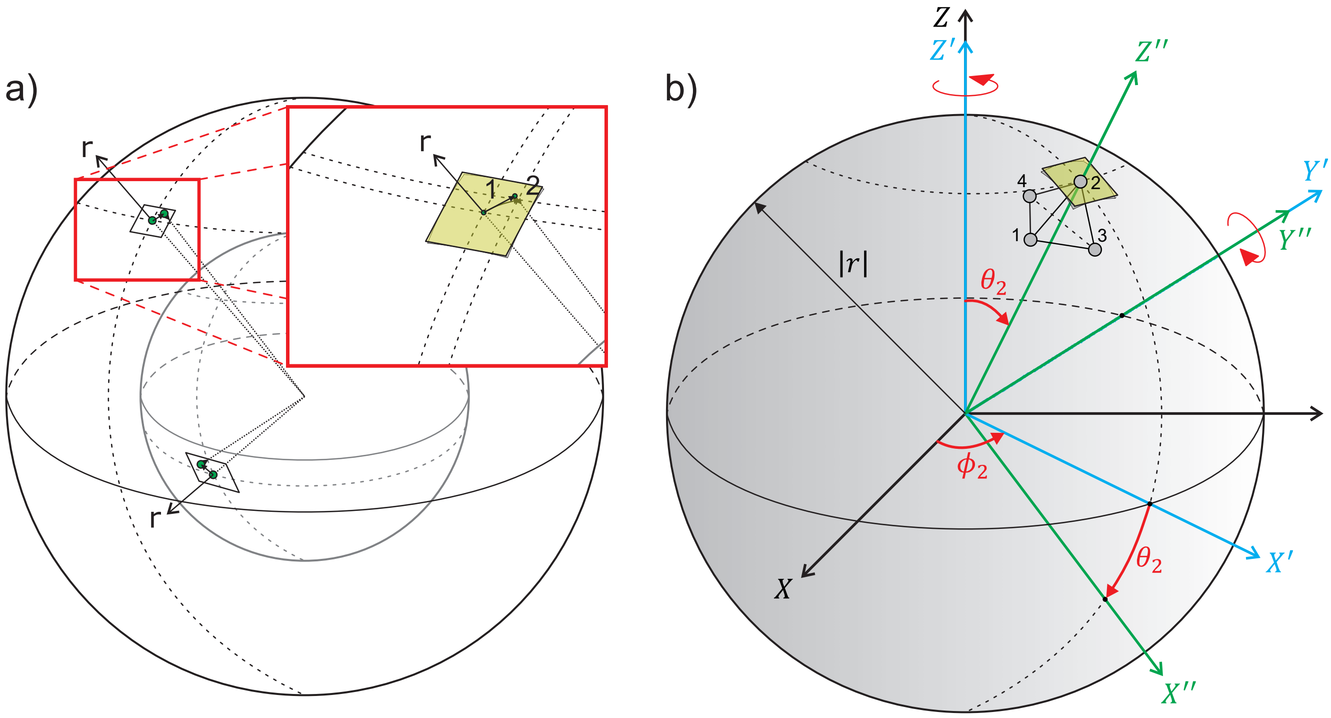


FIGURE 11 (a) Conceptual diagram for spherical boundary conditions. The motion of boundary nodes is first restricted to be along the tangent plane to the sphere. Then, they are 'pulled back' to the sphere's surface by projecting in the radial direction. (b) Implementation of spherical boundary conditions for one tetrahedron. Two rotations are needed for node 2 to pass from the global reference system (X, Y, Z) to the local reference system (X'', Y'', Z''), where $z_2{}'' = |r|$ is the boundary condition.

matrix

$$\boldsymbol{R_{3D}} = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta & \sin\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\alpha\cos\beta & \cos\alpha\sin\beta & \sin\alpha \end{bmatrix}, \tag{24}$$

where α and β are angles equivalents to latitude and longitude, respectively (Figure 3b). Applying equation (24) to equation (3) (see Appendix B for details), equation (3) becomes

$$k\begin{bmatrix} -{c_\alpha}^2{c_\beta}^2 & -{c_\alpha}^2 s_\beta c_\beta & -s_\alpha c_\alpha c_\beta & {c_\alpha}^2{c_\beta}^2 & {c_\alpha}^2 s_\beta c_\beta & s_\alpha c_\alpha c_\beta \\ -{c_\alpha}^2 s_\beta c_\beta & -{c_\alpha}^2{s_\beta}^2 & -s_\alpha c_\alpha s_\beta & {c_\alpha}^2 s_\beta c_\beta & {c_\alpha}^2{s_\beta}^2 & s_\alpha c_\alpha s_\beta \\ -s_\alpha c_\alpha c_\beta & -s_\alpha c_\alpha s_\beta & -{s_\alpha}^2 & s_\alpha c_\alpha c_\beta & s_\alpha c_\alpha s_\beta & {s_\alpha}^2 \\ {c_\alpha}^2{c_\beta}^2 & {c_\alpha}^2 s_\beta c_\beta & s_\alpha c_\alpha c_\beta & -{c_\alpha}^2{c_\beta}^2 & -{c_\alpha}^2 s_\beta c_\beta & -s_\alpha c_\alpha c_\beta \\ {c_\alpha}^2 s_\beta c_\beta & {c_\alpha}^2{s_\beta}^2 & s_\alpha c_\alpha s_\beta & -{c_\alpha}^2 s_\beta c_\beta & -{c_\alpha}^2{s_\beta}^2 & -s_\alpha c_\alpha s_\beta \\ s_\alpha c_\alpha c_\beta & s_\alpha c_\alpha s_\beta & {s_\alpha}^2 & -s_\alpha c_\alpha c_\beta & -s_\alpha c_\alpha s_\beta & -{s_\alpha}^2 \end{bmatrix}\begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} f_{1,x} \\ f_{1,y} \\ f_{1,z} \\ f_{2,x} \\ f_{2,y} \\ f_{2,z} \end{pmatrix} + kl_0\begin{pmatrix} c_\alpha c_\beta \\ c_\alpha s_\beta \\ s_\alpha \\ -c_\alpha c_\beta \\ -c_\alpha s_\beta \\ -s_\alpha \end{pmatrix}, \tag{25}$$

where $s_\alpha \equiv \sin\alpha$, $c_\alpha \equiv \cos\alpha$, $s_\beta \equiv \sin\beta$ and $c_\beta \equiv \cos\beta$. The system of equations is solved as described above (see equation (7)).

Spherical Boundary Conditions

For 3-D applications, we currently focus on developing unstructured spherical meshes. Using a notation similar to that for 2-D circular boundary conditions, we denote by Σ the inner and outer boundaries of the spherical shell with radii $r = r_{inner}$ and $r = r_{outer}$ respectively. Ω is the interior region between the boundaries. A useful boundary condition consists in prescribing boundary nodes that are free to slide along the local tangent plane to the spherical surface. Nodal sliding is generated in two independent steps (Figure 11a): 1) The node is allowed to move along the local tangent plane to the sphere, and 2) the node is returned to the sphere's surface by projecting in the radial direction. This approximation assumes that the radial distance needed to pull the node back to the surface of the sphere is small compared to the distance moved along the tangent plane. For simplicity, the mathematical implementation of the spherical boundary conditions is presented here only for one tetrahedron (Figure 11b). Node 2 is free to slide along the tangent plane since the boundary condition is $z_2{}'' = |r|$, where r is the radial distance from the centre of the sphere to the surface. This boundary condition is imposed by two rotations of the coordinate system for node 2. The first rotation is around the Z axis by an angle ϕ_2, which is the longitude of node 2 (local reference system (X', Y', Z') in blue in Figure 11b). The second rotation is around the Y' axis by an angle θ_2, which is the colatitude for node 2 (local reference system (X'', Y'', Z'') in green in Figure 11b). The complete rotation is given by the transformation matrix $\boldsymbol{T}$ that relates global coordinates $\boldsymbol{x}$ with local coordinates $\boldsymbol{x}''$ as follows

$$\underbrace{\begin{pmatrix} x_1\\ y_1\\ z_1\\ x_2\\ y_2\\ z_2\\ x_3\\ y_3\\ z_3\\ x_4\\ y_4\\ z_4 \end{pmatrix}}_{\boldsymbol{x}} = \underbrace{\begin{bmatrix} 1 &&&&&&&&&&& \\ & 1 &&&&&&&&&& \\ && 1 &&&&&&&&& \\ &&& \cos\phi_2\cos\theta_2 & -\sin\phi_2 & \cos\phi_2\sin\theta_2 &&&&&& \\ &&& \sin\phi_2\cos\theta_2 & \cos\theta_2 & \sin\phi_2\sin\theta_2 &&&&&& \\ &&& -\sin\theta_2 & 0 & \cos\theta_2 &&&&&& \\ &&&&&& 1 &&&&& \\ &&&&&&& 1 &&&& \\ &&&&&&&& 1 &&& \\ &&&&&&&&& 1 && \\ &&&&&&&&&& 1 & \\ &&&&&&&&&&& 1 \end{bmatrix}}_{\boldsymbol{T}} \underbrace{\begin{pmatrix} x_1\\ y_1\\ z_1\\ x_2''\\ y_2''\\ |r|\\ x_3\\ y_3\\ z_3\\ x_4\\ y_4\\ z_4 \end{pmatrix}}_{\boldsymbol{x}''} . \tag{26}$$

This transformation matrix contains a θ and ϕ angle for each node on the spherical boundary. Applying the transformation matrix to stiffness matrix and force vector

$$\boldsymbol{K}'' = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{T}\,, \tag{27}$$

$$\boldsymbol{f}_{l_0}'' = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{f}_{l_0}\,, \tag{28}$$

the new system of equations is given by

$$\boldsymbol{K}''\boldsymbol{x}'' = \boldsymbol{f}_{l_0}''\,, \tag{29}$$

which is solved for $\boldsymbol{x}''$. Global Cartesian coordinates are recovered through the transformation matrix

$$\boldsymbol{x} = \boldsymbol{T}\boldsymbol{x}''\,. \tag{30}$$

Quality factor for tetrahedra

The 3-D quality factor for a tetrahedron is defined by

$$q_{_{3D}} = \frac{3r_s}{R_s}\,, \tag{31}$$

where r_s is the radius of the tetrahedron's inscribed sphere and R_s is the radius of its circumscribed sphere. R_s and r_s are given by

$$r_s = \frac{|\boldsymbol{a}\cdot(\boldsymbol{b}\times\boldsymbol{c})\,|}{(|\boldsymbol{a}\times\boldsymbol{b}| + |\boldsymbol{b}\times\boldsymbol{c}| + |\boldsymbol{c}\times\boldsymbol{a}| + |\,(\boldsymbol{a}\times\boldsymbol{b}) + (\boldsymbol{b}\times\boldsymbol{c}) + (\boldsymbol{c}\times\boldsymbol{a})\,|)}\,, \tag{32}$$

$$R_s = \frac{|\boldsymbol{a}^2\cdot(\boldsymbol{b}\times\boldsymbol{c}) + \boldsymbol{b}^2\cdot(\boldsymbol{c}\times\boldsymbol{a}) + \boldsymbol{c}^2\cdot(\boldsymbol{a}\times\boldsymbol{b})\,|}{2|\boldsymbol{a}\cdot(\boldsymbol{b}\times\boldsymbol{c})\,|}\,, \tag{33}$$

where $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are vectors pointing from one node, O, to the three other nodes of the tetrahedron A, B and C respectively (Figure 12a). This quality factor is normalized to be 0 for degenerate tetrahedra and 1 for regular tetrahedra. Note that different definitions for normalized aspect ratios can lead to different estimators for the global quality of a mesh. For example, *Anderson et al.* [2005] define a shape measure s that depends on tetrahedral volume and the lengths of its edges. Computing $q_{_{3D}}$ and s for the same mesh gives differences of up to 0.1 for the worst element (Figure 12b). The quality factor $q_{_{3D}}$ that we choose to use is a more restrictive aspect ratio than the shape factor measure s.

Element shape improvements

In 3-D, even when the expected nodal density is achieved ($\mu < \mu_t$) by adding or rejecting nodes, a considerable number of poorly shaped tetrahedra can still persist. Local improvements are needed to ensure that the mesh is robust enough to perform optimal FEM calculations. After some experimentation, we found it appropriate to use $\mu_t = 0.13$ although this can vary from 0.1 to 0.2 depending on the degree of mesh refinement. The value of μ_t for 2-D meshes is smaller than for 3-D meshes due to the shape compactness that can be achieved on a 2-D planar surface.

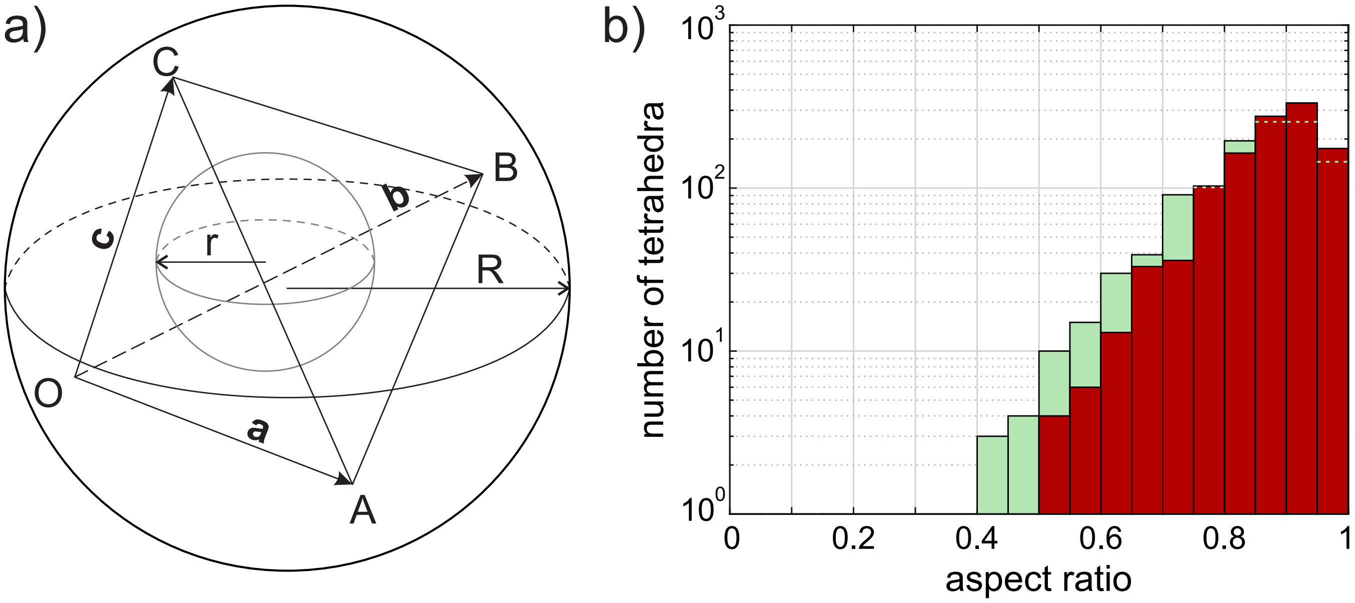


FIGURE 12 (a) Tetrahedron with vertices OABC. *R* and *r* are the radius of the circumscribed and inscribed spheres respectively. (b) Number of tetrahedra as a function of the quality factor $q_{_{3D}}$ (green) and the shape measure s (red) for the same mesh.

Methods based on swapping edges or faces to improve element quality can possibly generate non-Delaunay triangulations, which will cause problems in algorithms that rely on a mesh created by a Delaunay triangulation (e.g. point search algorithms). Hence, as an alternative and in addition to smoothing the position of interior nodes, we recommend two additional operations to improve the quality of tetrahedral elements.

Improvement of badly shaped tetrahedra

Unstructured 3-D meshes are composed of irregular tetrahedra. Some may be quite poor in terms of their shape and quality factor (see *Cheng et al.* [2000] for a complete categorization of badly shaped tetrahedra). The first improvement for tetrahedral shapes acts locally and only modifies one node of each badly shaped tetrahedron. For each badly shaped tetrahedron, identified by $q_{_{3D}} < q_{bad}$, where $0.2 \leq q_{bad} \leq 0.3$, we select the spring with the maximum distortion, i.e. $max(|\epsilon|)$. If $\epsilon > 0$, a new node is created in the midpoint of the selected spring, while a node at one end of the selected spring is removed if $\epsilon < 0$. A new connectivity is then created by another Delaunay triangulation. The new connectivity is only modified in the surroundings of nodes that have been added or removed, keeping the rest of the connectivity to be the same as the old triangulation. Figure S4 illustrates a simple example that improves badly shaped tetrahedra when meshing the unit cube.

Removing slivers

Slivers are degenerate tetrahedra whose vertices are well-spaced and near the equator of their circumsphere, hence their quality factor and enclosed volume are close to zero. We define a sliver as a tetrahedron with $q_{_{3D}} < 0.1$. Our routine for removing slivers is purely geometrical, i.e. it does not take into account the actual or desired length of the springs. The four vertices of each sliver are replaced by the three mesh points of the best potential triangle that can be generated from all permutations of its vertices and potential new nodes created at the midpoints of its springs (Figure S5). Delaunay triangulation is called afterwards to create the connectivity matrix around the changed nodes.

Example: Spherical shell mesh with an embedded high resolution region

We show the results for a spherical shell mesh with an embedded high-resolution sub-region (code available in Supporting information). The input mesh parameters are listed in Table 1. We recommend to set the point around which the refined region

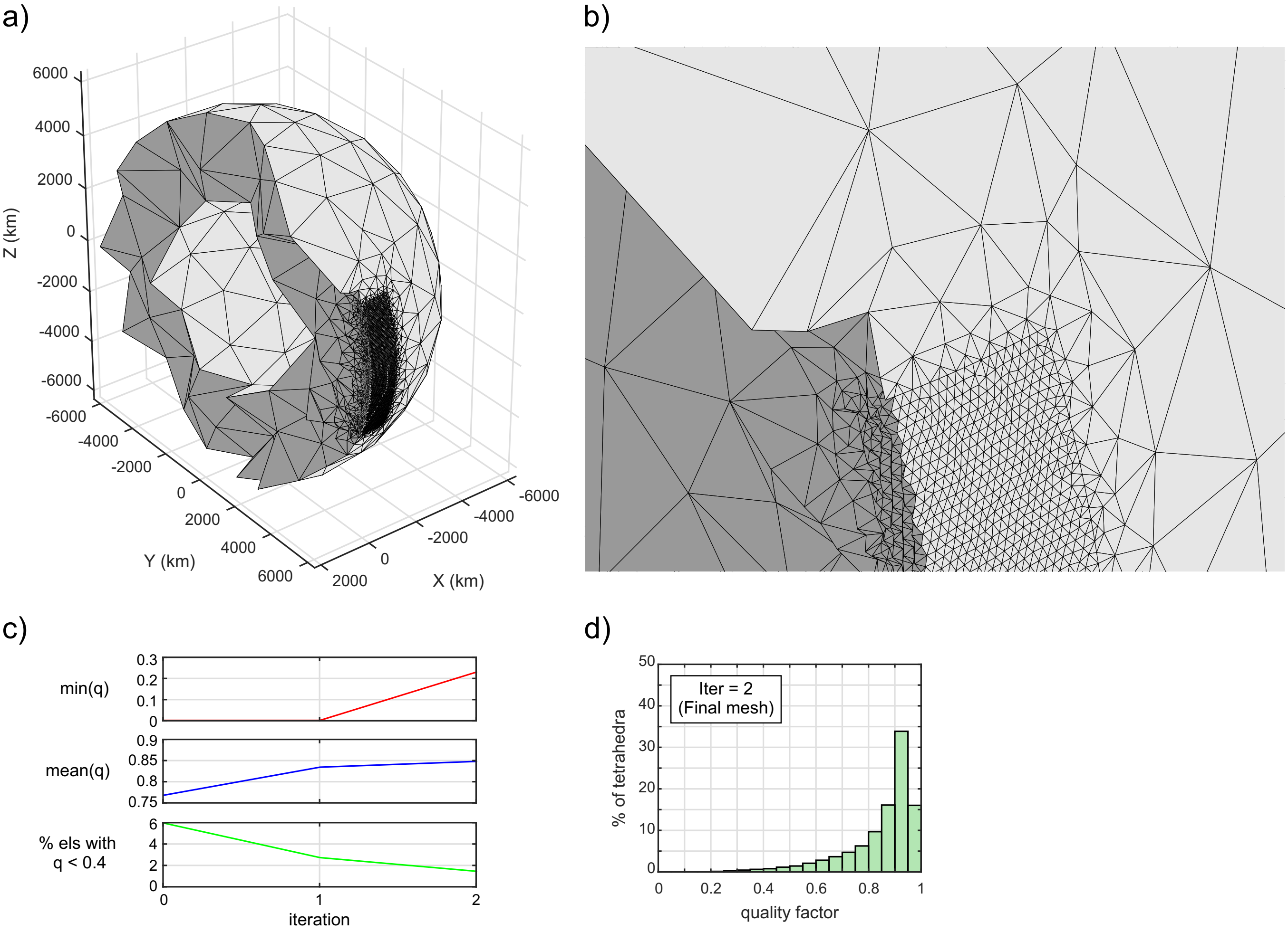


FIGURE 13 (a) Cross section of the final mesh with an embedded high resolution sub-region after refinement using the guide-mesh. (b) Zoom around the boundary of the refined region. (c) Minimum quality factor (red line), mean quality factor for all elements (blue line) and fraction of elements having a quality factor lower than 0.4% (green line) as a function of iteration number. (d) Histogram of the fraction of elements as a function of quality factor for the final mesh. The lowest quality factor for an element is 0.23.

is created far from the polar axis since the guide-mesh can have difficulties in interpolating the desired spring lengths near the polar axis.

For this example, the domain of the mesh is a spherical shell whose boundaries represent the core-mantle boundary and the Earth's surface (Figure 10b). The smallest tetrahedra with quasi-uniform size lie inside the high resolution region (red tesseroid in Figure 10b). This region is embedded within a coarser global mesh. A transition region (green tesseroid in Figure 10b) guarantees a gradual change in tetrahedral size from the high resolution region to the coarse region. The algorithm created the mesh in 130 s (orange dot in Figure 5) after 2 iterations (see Figure 13a for a cross section of the final mesh). Figure 13b shows a detail of the mesh around the northern boundary of the refined region. The mesh has 27000 nodes forming 150000 tetrahedra with an edge-length factor $l_{0r}/l_{0c} = 1/33$. The fraction of tetrahedra within the coarse, transition and refined regions is 0.7%, 21.6% and 77.7% respectively (Figure S6). The worst quality factor for an element is 0.23 (red line in Figure 13c) and the mean of the quality factor for all elements is 0.85 (blue line in Figure 13c). Only 1.5% of the tetrahedra have a quality factor lower than 0.4. Figure 13d shows the fraction of elements as a function of their quality factor for the final mesh.

5 | MESH COMPARISON WITH OTHER ALGORITHMS

In 2-D, there are many open source and commercial mesh generators that are flexible and work well, e.g. Triangle [*Shewchuk*, 1996], so we will not discuss the 2-D version of our code here. In general, comparing meshes created by different algorithms can be a difficult task because typically each algorithm creates a mesh with desirable characteristics for a specific problem. For this reason, we will test our 3-D algorithm by creating a simple geometry that can be easily reproduced by all compared algorithms. We chose three algorithms ADP3D [*Dompierre et al.*, 1998], DistMesh [*Persson and Strang*, 2004] and Netgen [*Schöberl*, 1997] to compare when creating a unit-radius sphere with a preferred nodal distance $l_0 = 0.2$. DistMesh, Netgen and the algorithm presented in this study were run on the same machine. For ADP3D we only have the benchmark published by *Dompierre et al.* [1998]. Table 2 shows the number of nodal degrees of freedom (dofs), the number of mesh elements, the computational time (in sec on our 3.2 GHz Intel Core i5 (MacOSX 10.12.5) machine with 24 GB of 1600 MHz DDR3 memory) and several tetrahedron shape-quality measures [cf. *Dompierre et al.*, 1998] for each algorithm:

- Quality factor, q_{3D}, given by equation (31) and also known as radius ratio, ρ.
- Aspect ratio, γ, given by
$$\gamma = 2\sqrt{6}\frac{r_s}{l_{max}}, \tag{34}$$
where l_{max} is the length of the longest edge of the tetrahedron.
- Mean ratio, η, given by
$$\gamma = \frac{12\sqrt[3]{9v^2}}{\sum_{i=1}^{6} l_i^2}, \tag{35}$$
where l_i is the length of each edge of the tetrahedron.
- Solid angle, θ_{min}, given by
$$\theta_{min} = \beta \min\left[\sin\left(\frac{\theta_i}{2}\right)\right], \tag{36}$$
$$\sin(\theta_i/2) = 12v\left(\prod_{\substack{j,k\neq i\\ 0\leq j<k\leq 3}} \left(\left(l_{ij}+l_{ik}\right)^2 - l_{jk}^2\right)\right)^{-1/2}, \tag{37}$$
where $\beta^-1 = \sqrt{6}/9$.

TABLE 2 Statistical data.

		ADP3D [1]	**DistMesh** [2]	**Netgen** [3]	**This study**
Number of nodal dofs		3132	3207	3453	3483
Number of elements		4905	5177	4942	5230
Computational time (s)		- -	16.51	3.46	3.75
Quality factor q_{3D}	min	0.324	0.045	0.501	0.486
	mean	0.873	0.898	0.793	0.881
Aspect ratio γ	min	0.317	0.038	0.457	0.405
	mean	0.772	0.796	0.684	0.773
Mean ratio η	min	0.501	0.112	0.624	0.560
	mean	0.898	0.915	0.832	0.899
Solid angle θ_{min}	min	0.178	0.044	0.185	0.232
	mean	0.665	0.737	0.525	0.686

[1][*Dompierre et al.*, 1998] [2][*Persson and Strang*, 2004] [3][*Schöberl*, 1997]

The histograms corresponding to these data are shown in Figure 14. Our approach creates a comparable mesh in about the same time as the compiled Netgen code, and more than four times faster than Distmesh (Table 2). Distmesh produces slightly higher mean values for all the shape measures. However, since it does not include routines to deal with slivers, it also produces the lowest minimum value for all the shape measures, e.g. it produces a few sliver-like 'bad' elements. Netgen gives the highest minimum value for all shape measures except for the solid angle. However, its mean values are the lowest. ADP3D produces comparable, but slightly lower minimum and mean values for all the shape measures to our MATLAB codetool, although we do not know the computational time for this algorithm. A specific feature that is missing from our code is that there is no graphical user interface (GUI). For the meshes we wish to make it is relatively easy to define a guide-mesh with a few lines of MATLAB code. As we plan to use this algorithm in an adaptive finite element code it was not a high priority to create an associated GUI. Since Distmesh is also written in MATLAB and was the original inspiration to our algorithm, a user could straightforwardly modify the original Distmesh code to include our guide-mesh approach. The other algorithms would require larger modifications to similarly control the mesh refinement associated with a mesh generation including an embedded high resolution sub-region.

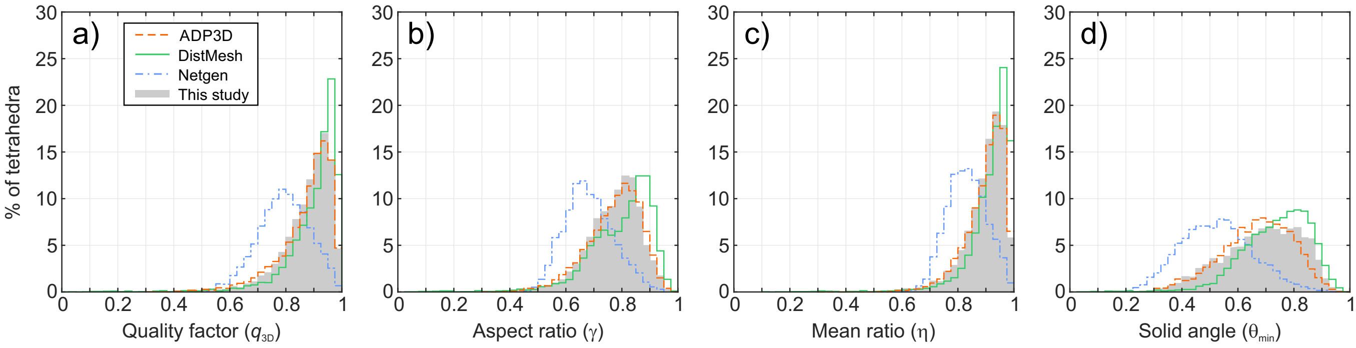


FIGURE 14 Histogram of the fraction of elements as a function of: (a) quality factor, (b) aspect ratio, (c) mean ratio and (d) solid angle.

6 | SUMMARY

We have developed the tools for generating unstructured meshes in 2-D, 3-D, and spherical geometries that can contain embedded high resolution sub-regions. While we do not discuss the recipe for the generation of a Cartesian 3-D mesh, only small modifications to the 3-D spherical code are needed to place boundary points along linear boundary edges and planar boundary surfaces rather than spherical shell boundaries. The algorithm employs the FEM to solve for the optimal nodal positions of a spring-like system of preferred nodal positions. Straight line, circular and spherical boundary conditions are imposed to constrain the shape of the mesh. We use a guide-mesh approach to smoothly refine the mesh around regions of interest. Methods for achieving the expected nodal density and improving the element shape and quality have been presented to ensure a high quality of the generated mesh, which is a prerequisite for high computational accuracy and fast iterative convergence. We have compared our algorithm to other open source mesh generators. This comparison shows that our algorithm generates the highest quality mesh, i.e. the highest minimum and mean value for all the shape measures, within a computational time comparable to or better than other open source algorithms. The presented mesh generator can be easily used for adaptive mesh refinement by varying the desired spring length depending on variables of interest. Since an adaptive refinement (or coarsening) will only change node positions in regions where the spatial resolution is changed, most nodes of the spring system remain in equilibrium so that only very few iterations are required to obtain the updated mesh.

SUPPORTING INFORMATION

The following supporting information is available as part of the online article:

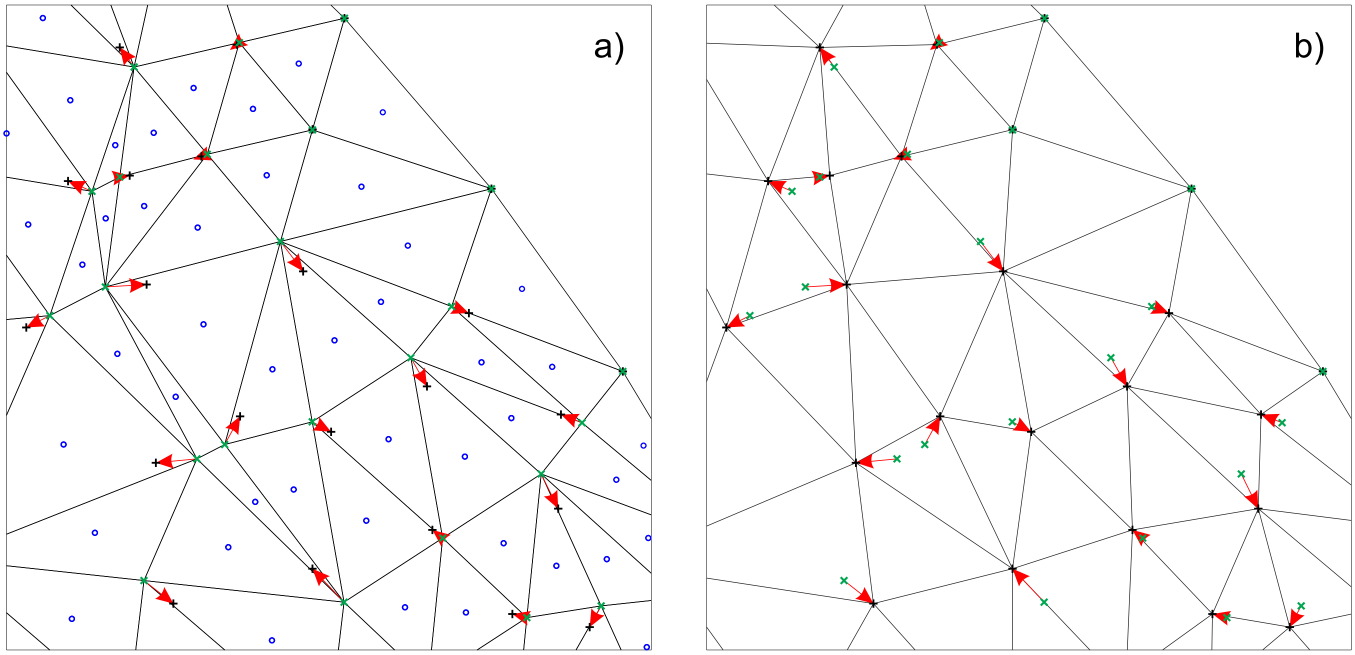


FIGURE S1 (a) Initial 2-D mesh. (b) Mesh after applying the Laplacian correction to smooth positions of its interior nodes. Blue points are the barycentres of the triangles. Green and black crosses are the nodal positions before and after smoothing, respectively. Red arrows indicate the motions of interior nodes.

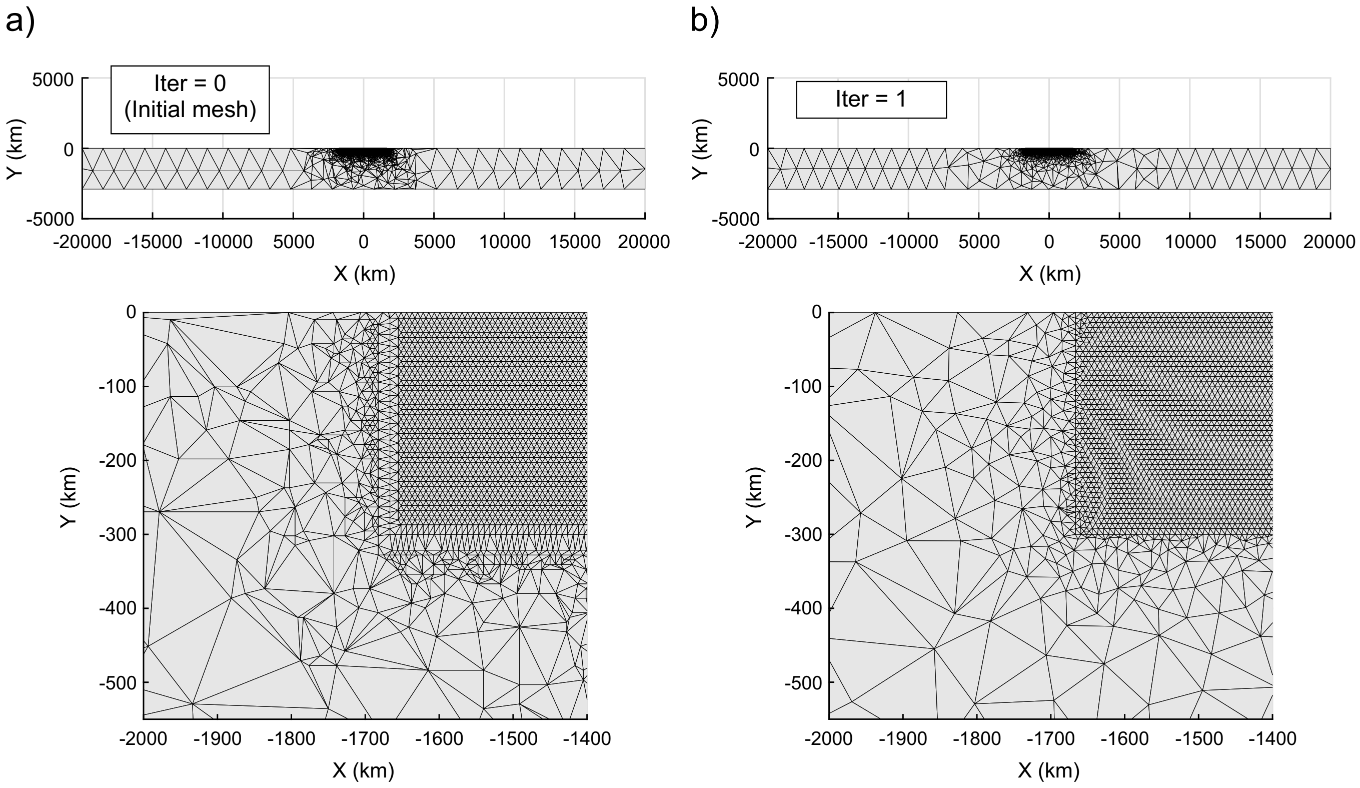


FIGURE S2 (a) Initial mesh (top) for a rectangular box with an embedded high resolution sub-region and a zoom around the left boundary of the refined region (bottom). (b) Mesh (top) and zoom (bottom) after the first iteration.

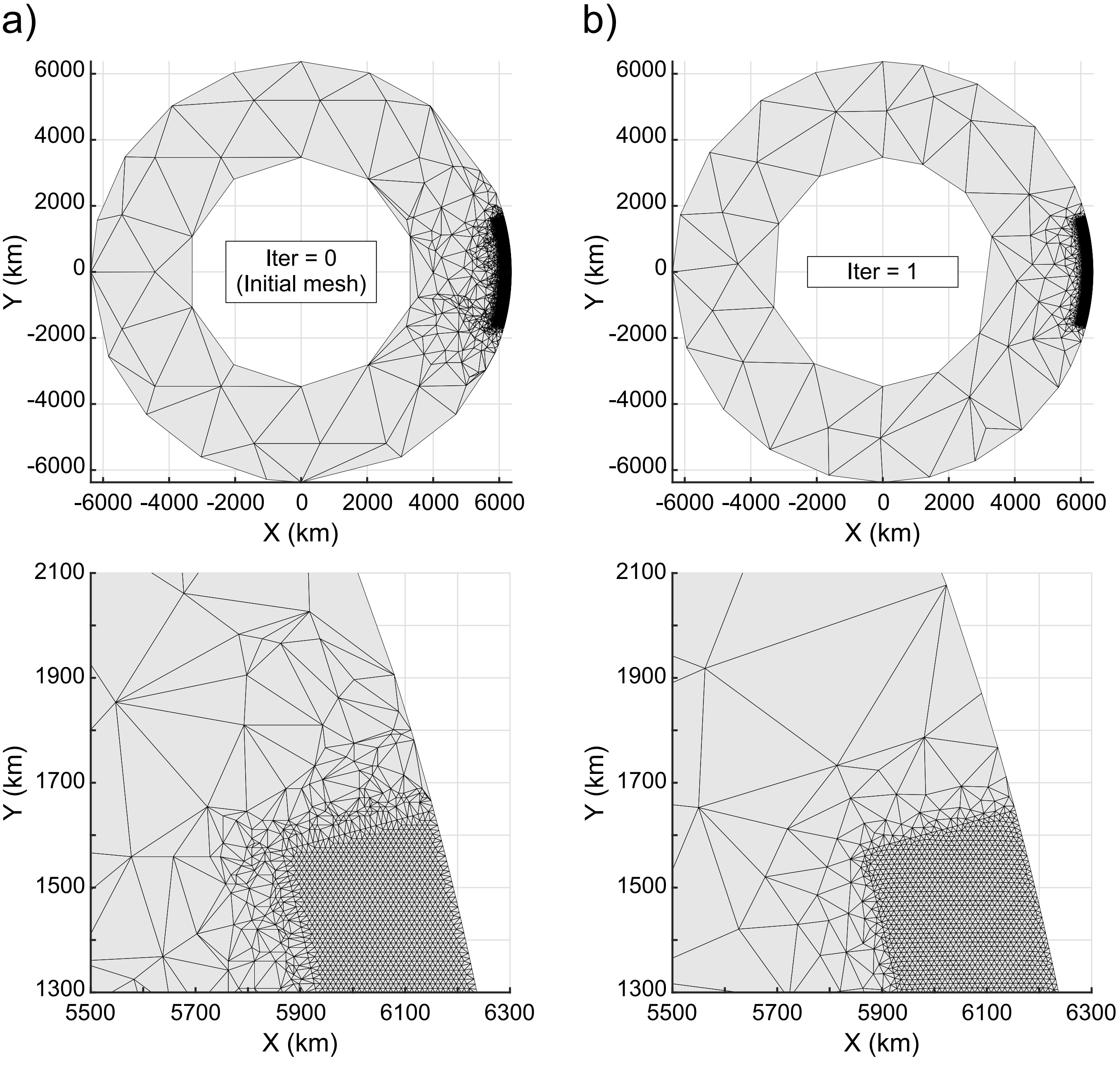


FIGURE S3 (a) Initial mesh (top) for a cylindrical annulus with an embedded high resolution sub-region and a zoom around an edge of the refined region (bottom). (b) Mesh (top) and zoom (bottom) after the first iteration.

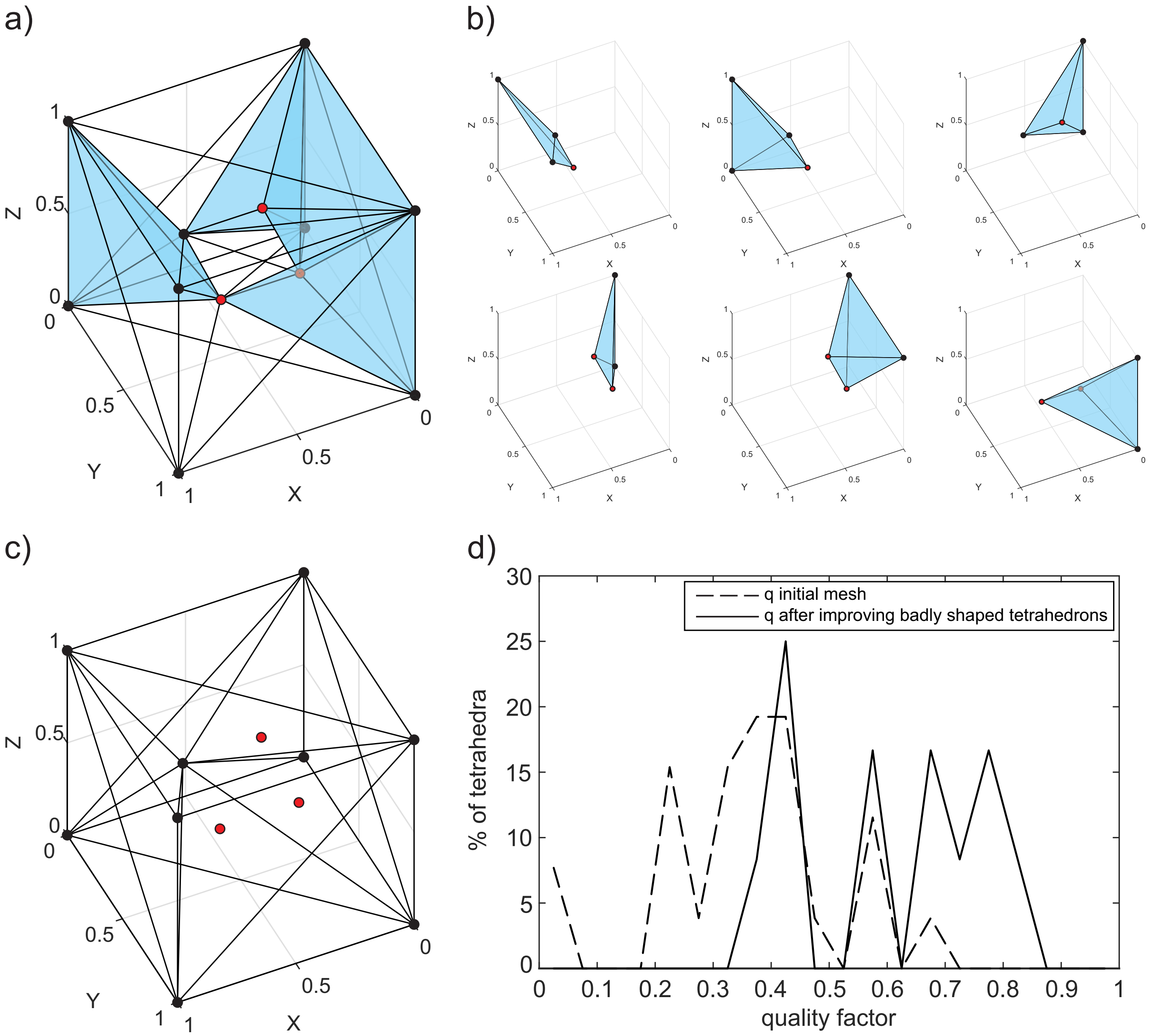


FIGURE S4 (a) Initial mesh with badly shaped tetrahedra (in blue). Rejected nodes in red. (b) Badly shaped tetrahedra. (c) Mesh after improving badly shaped tetrahedra contains no badly shaped tetrahedra. (d) Fraction of tetrahedra for a given quality factor for both before (dashed line) and after (solid line) local improvements to the shape of badly shaped tetrahedra. The minimum quality factor for the initial mesh is 0.04 and for the final mesh is 0.39.

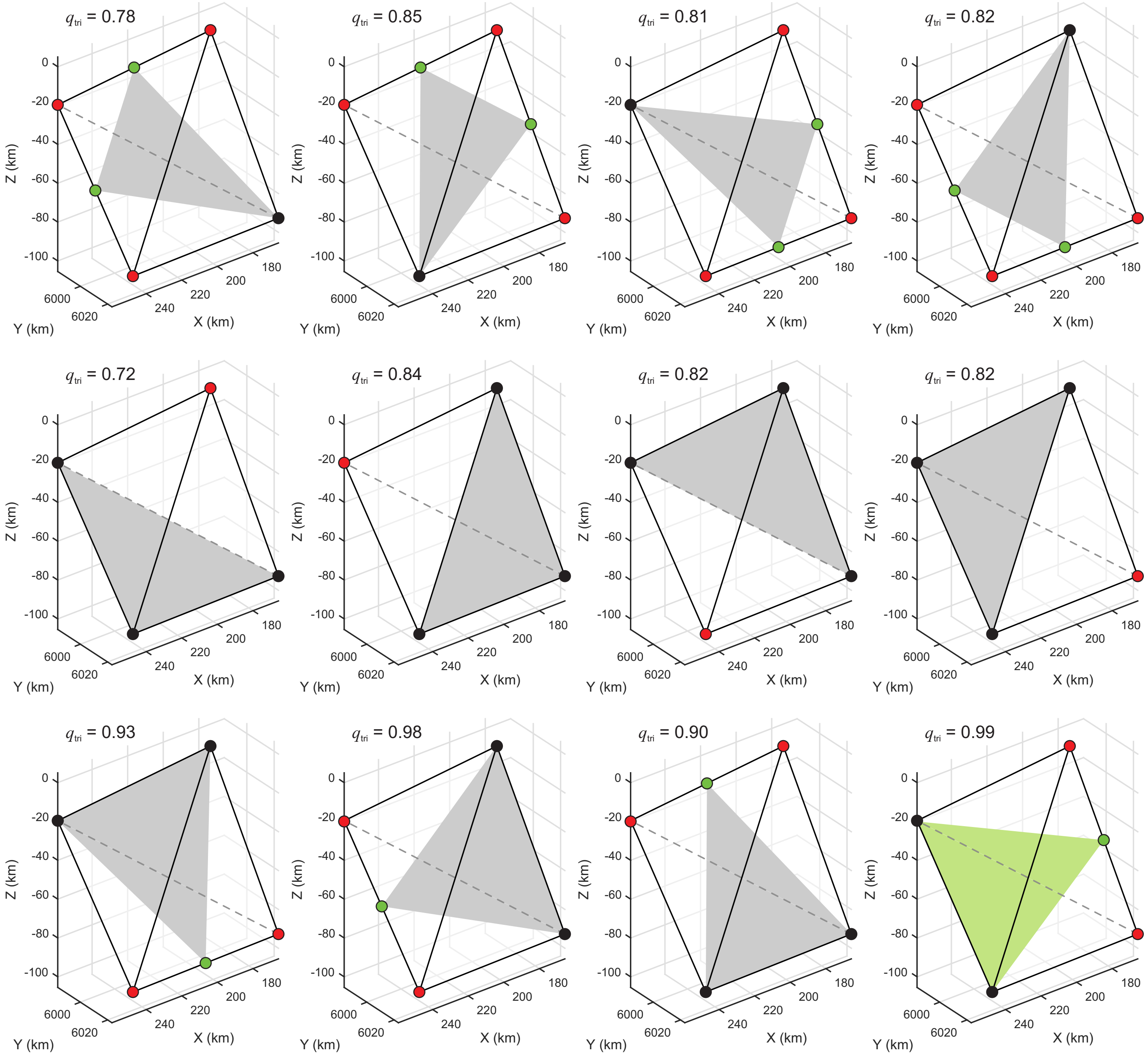


FIGURE S5 Removing a sliver (represented by black lines and dashed grey line for hidden edge). Possible triangles (grey and green colours) created from permutations of the vertices and midpoints of the edges of a sliver. Black, red and green points represent unaltered, removed and added nodes, respectively. q_{tri} is the quality factor for each triangle. The four vertices of the sliver are replaced by the three mesh points of the potential triangle with the best quality factor (green colour).

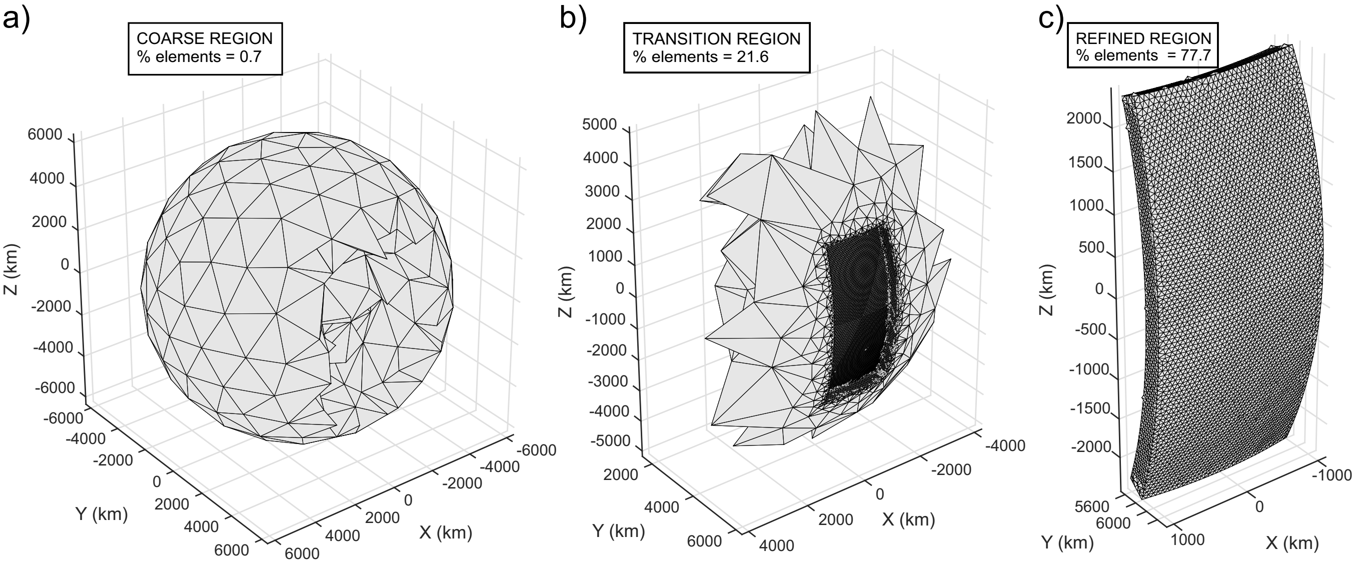


FIGURE S6 (a) Tetrahedra within the coarse region. (b) Tetrahedra within the transition region. (c) Tetrahedra within the refined region.

How to cite this article: Taramón, J. M., Morgan, J. P., Shi, C., and Hasenclever, J. (2018), Generation of unstructured meshes in 2-D, 3-D, and spherical geometries with embedded high resolution sub-regions.

APPENDIX

A. DERIVATION OF EQUATION (5)

The 2-D development of equation (3), rewritten here for convenience

$$\begin{pmatrix} f_1{}' \\ f_2{}' \end{pmatrix} + k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} 0 \\ l_0 \end{pmatrix} = k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} x_1{}' \\ x_2{}' \end{pmatrix}, \tag{A1}$$

is given by two steps. First, develop the right hand side of equation (A1) by writing local coordinates as a function of global coordinates (see Figure 3a)

$$\begin{aligned} k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} x_1{}' \\ x_2{}' \end{pmatrix} \\ &= k \begin{bmatrix} x_2{}' - x_1{}' \\ -(x_2{}' - x_1{}') \end{bmatrix} \\ &= k \begin{bmatrix} \left[(x_2 - x_1)c_\alpha + (y_2 - y_1)s_\alpha\right] \\ -\left[(x_2 - x_1)c_\alpha + (y_2 - y_1)s_\alpha\right] \end{bmatrix} \\ &= k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 c_\alpha + y_1 c_\alpha \\ x_2 c_\alpha + y_2 c_\alpha \end{bmatrix} \\ &= k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ 0 & 0 & \cos\alpha & \sin\alpha \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{pmatrix}, \end{aligned} \tag{A2}$$

where $s_\alpha \equiv \sin\alpha$ and $c_\alpha \equiv \cos\alpha$. Second, express the global coordinates of the force vector as a function of local coordinates (see Figure 3a)

$$\begin{pmatrix} f_{1,x} \\ f_{1,y} \\ f_{2,x} \\ f_{2,y} \end{pmatrix} = \begin{bmatrix} c_\alpha & 0 \\ s_\alpha & 0 \\ 0 & c_\alpha \\ 0 & s_\alpha \end{bmatrix} \begin{pmatrix} f_1{}' \\ f_2{}' \end{pmatrix}. \tag{A3}$$

Combining equations (A1) and (A2) gives

$$\begin{pmatrix} f_1{}' \\ f_2{}' \end{pmatrix} = k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ 0 & 0 & \cos\alpha & \sin\alpha \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{pmatrix} - k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} 0 \\ l_0 \end{pmatrix}. \tag{A4}$$

Substituting equation (A4) into equation (A3) and reordering gives

$$k \begin{bmatrix} c_\alpha & 0 \\ s_\alpha & 0 \\ 0 & c_\alpha \\ 0 & s_\alpha \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ 0 & 0 & \cos\alpha & \sin\alpha \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} f_{1,x} \\ f_{1,y} \\ f_{2,x} \\ f_{2,y} \end{pmatrix} + k \begin{bmatrix} c_\alpha & 0 \\ s_\alpha & 0 \\ 0 & c_\alpha \\ 0 & s_\alpha \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} 0 \\ l_0 \end{pmatrix}, \tag{A5}$$

which is equivalent to equation (5).

B. DERIVATION OF EQUATION (25)

The 3-D development of equation (3), rewritten here for convenience

$$\begin{pmatrix} f_1{}' \\ f_2{}' \end{pmatrix} + k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} 0 \\ l_0 \end{pmatrix} = k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} x_1{}' \\ x_2{}' \end{pmatrix}, \tag{B6}$$

also involves two steps. First, develop the right hand side of equation (B6) by writing local coordinates as a function of global coordinates (see Figure 3b)

$$\begin{aligned} k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} x_1{}' \\ x_2{}' \end{pmatrix} \\ &= k \begin{bmatrix} x_2{}' - x_1{}' \\ -(x_2{}' - x_1{}') \end{bmatrix} \\ &= k \begin{bmatrix} \left(\left[(x_2 - x_1)c_\beta + (y_2 - y_1)s_\beta\right]c_\alpha + (z_2 - z_1)s_\alpha \right) \\ -\left(\left[(x_2 - x_1)c_\beta + (y_2 - y_1)s_\beta\right]c_\alpha + (z_2 - z_1)s_\alpha \right) \end{bmatrix} \\ &= k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 c_\alpha c_\beta + y_1 c_\alpha s_\beta + z_1 s_\alpha \\ x_2 c_\alpha c_\beta + y_2 c_\alpha s_\beta + z_2 s_\alpha \end{bmatrix} \\ &= k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta & s_\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & c_\alpha c_\beta & c_\alpha s_\beta & s_\alpha \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \end{pmatrix}, \end{aligned} \tag{B7}$$

where $s_\alpha \equiv \sin\alpha$, $c_\alpha \equiv \cos\alpha$, $s_\beta \equiv \sin\beta$ and $c_\beta \equiv \cos\beta$. Second, express the global coordinates of the force vector as a function of local coordinates (see Figure 3b)

$$\begin{pmatrix} f_{1,x} \\ f_{1,y} \\ f_{1,z} \\ f_{2,x} \\ f_{2,y} \\ f_{2,z} \end{pmatrix} = \begin{bmatrix} c_\alpha c_\beta & 0 \\ c_\alpha s_\beta & 0 \\ s_\alpha & 0 \\ 0 & c_\alpha c_\beta \\ 0 & c_\alpha s_\beta \\ 0 & s_\alpha \end{bmatrix} \begin{pmatrix} f_1{}' \\ f_2{}' \end{pmatrix}. \tag{B8}$$

Combining equations (B6) and (B7) gives

$$\begin{pmatrix} f_1{}' \\ f_2{}' \end{pmatrix} = k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta & s_\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & c_\alpha c_\beta & c_\alpha s_\beta & s_\alpha \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \end{pmatrix} - k \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} 0 \\ l_0 \end{pmatrix}. \tag{B9}$$

Substituting equation (B9) into equation (B8) and reordering gives

$$k \begin{bmatrix} c_\alpha c_\beta & 0 \\ c_\alpha s_\beta & 0 \\ s_\alpha & 0 \\ 0 & c_\alpha c_\beta \\ 0 & c_\alpha s_\beta \\ 0 & s_\alpha \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta & s_\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & c_\alpha c_\beta & c_\alpha s_\beta & s_\alpha \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} f_{1,x} \\ f_{1,y} \\ f_{1,z} \\ f_{2,x} \\ f_{2,y} \\ f_{2,z} \end{pmatrix} + k \begin{bmatrix} c_\alpha c_\beta & 0 \\ c_\alpha s_\beta & 0 \\ s_\alpha & 0 \\ 0 & c_\alpha c_\beta \\ 0 & c_\alpha s_\beta \\ 0 & s_\alpha \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} 0 \\ l_0 \end{pmatrix}, \tag{B10}$$

which is equivalent to equation (25).

References

Alliez, P., D. Cohen-Steiner, M. Yvinec, and M. Desbrun (2005), Variational tetrahedral meshing, *ACM Trans. Graph.*, *24*(3), 617–625, doi:10.1145/1073204.1073238.

Anderson, A., X. Zheng, and V. Cristini (2005), Adaptive unstructured volume remeshing - I: The method, *J. Comput. Phys.*, *208*(2), 616–625, doi:10.1016/j.jcp.2005.02.023.

Bern, M., D. Eppstein, and J. Gilbert (1994), Provably good mesh generation, *J. Comput. Syst. Sci.*, *48*(3), 384–409, doi:10.1016/S0022-0000(05)80059-5.

Burstedde, C., O. Ghattas, M. Gurnis, G. Stadler, E. Tan, T. Tu, L. C. Wilcox, and S. Zhong (2008), Scalable adaptive mantle convection simulation on petascale supercomputers, 2008 SC - Int. Conf. High Perform. Comput. Networking, Storage Anal., pp. 1–15, doi:10.1109/SC.2008.5214248.

Cheng, S., and T. Dey (2002), Quality meshing with weighted Delaunay refinement, *Proc. Thirteen. Annu. ACM-SIAM Symp. Discret. algorithms*, *33*(1), 137–146, doi:10.1137/S0097539703418808.

Cheng, S., T. Dey, H. Edelsbrunner, M. A. Facello, and S. Teng (2000), Sliver exudation, *J. ACM*, *47*(5), 883–904, doi:10.1145/355483.355487.

Chew, L. P. (1989), Guaranteed-Quality Triangular Meshes, *Tech. rep.*, Department of Computer Science, Cornell University, Ithaca, New York.

Chew, L. P. (1997), Guaranteed-Quality Delaunay Meshing in 3D (short version), *Proc. Thirteen. Annu. Symp. Comput. Geom.*, pp. 391–393, doi:10.1145/262839.263018.

Choi, W., D. Kwak, I. Son, and Y. Im (2003), Tetrahedral mesh generation based on advancing front technique and optimization scheme, *Int. J. Numer. Methods Eng.*, *58*(12), 1857–1872, doi:10.1002/nme.840.

Dabrowski, M., M. Krotkiewski, and D. W. Schmid (2008), MILAMIN: MATLAB-based finite element method solver for large problems, *Geochem. Geophys. Geosyst.*, *9*(4), doi:10.1029/2007GC001719.

Davies, D. R., C. R. Wilson, and S. C. Kramer (2011), Fluidity: A fully unstructured anisotropic adaptive mesh computational modeling framework for geodynamics, *Geochem. Geophys. Geosyst.*, *12*(6), doi:10.1029/2011GC003551.

Dompierre, J., P. Labbé, F. Guibault, and R. Camarero (1998), Proposal of benchmarks for 3D unstructured tetrahedral mesh optimization, 7th Int. Meshing Roundtable, pp. 525–537.

Edelsbrunner, H., and D. Guoy (2002), An Experimental Study of Sliver Exudation, *Eng. Comput.*, *18*, 229–240, doi:10.1007/s003660200020.

Ito, Y., A. M. Shih, and B. K. Soni (2004), Reliable Isotropic Tetrahedral Mesh Generation Based on an Advancing Front Method, 13th Int. Meshing Roundtable, pp. 95–105.

Ito, Y., A. M. Shih, and B. K. Soni (2009), Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates, *Int. J. Numer. Methods Eng.*, *77*, 1809–1833, doi:10.1002/nme.2470.

Kronbichler, M., T. Heister, and W. Bangerth (2012), High accuracy mantle convection simulation through modern numerical methods, *Geophys. J. Int.*, *191*(1), 12–29, doi:10.1111/j.1365-246X.2012.05609.x.

Labelle, F., and J. R. Shewchuk (2007), Isosurface stuffing: Fast Tetrahedral Meshes with Good Dihedral Angles, *ACM Trans. Graph.*, *26*(3), doi:10.1145/1276377.1276448.

Li, X., and S. Teng (2001), Generating well-shaped Delaunay meshed in 3D, 12th Annu. ACM-SIAM Symp. Discret. algorithms, pp. 28–37, Washington, D. C.

Löhner, R., and P. Parikh (1988), Generation of three-dimensional unstructured grids by the advancing-front method, *Int. J. Numer. Methods Fluids*, *8*(10), 1135–1149, doi:10.1002/fld.1650081003.

Mitchell, S. A., and S. A. Vavasis (1992), Quality mesh generation in three dimensions, Proc. Eighth Annu. Symp. Comput. Geom. ACM, pp. 212–221, doi:10.1145/142675.142720.

Persson, P., and G. Strang (2004), A Simple Mesh Generator in MATLAB, *SIAM Rev.*, *46*(2), 329–345, doi:10.1137/S0036144503429121.

Ruppert, J. (1995), A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation, *J. Algorithms*, *18*, 548–585, doi:10.1006/jagm.1995.1021.

Schöberl, J. (1997), An advancing front 2D/3D-mesh generator based on abstract rules, *Comput. Vis. Sci.*, *1*(1), 41–52, doi:10.1007/s007910050004.

Shewchuk, J. R. (1996), Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator, in *Lin, Ming C and Manocha, Dinesh*, *Lecture Notes in Computer Science*, vol. 1148, edited by A. C. G. T. G. Eng., pp. 203–222, Springer, Berlin, doi:10.1007/BFb0014497.

Shewchuk, J. R. (1998), Tetrahedral mesh generation by Delaunay refinement, 14th Annu. Symp. Comput. Geom. SCG '98, pp. 86–95, doi:10.1145/276884.276894.

Shewchuk, J. R. (2002), What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures, Elev. Int. Meshing Roundtable, pp. 115–126, doi:10.1.1.68.8538.

Si, H. (2015), TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator, *AMC Trans. Math. Softw.*, *41*(2), doi:10.1145/2629697.

Chapter 3

Shape-preserving finite elements in cylindrical and spherical geometries: The Double Jacobian

Jason P. Morgan, Jorge M. Taramón, Jörg Hasenclever. Shape-preserving finite elements in cylindrical and spherical geometries: The Double Jacobian. Manuscript planned for submission at International Journal for Numerical Methods in Fluids, 2018b.

Authors contribution

JPM designed the initial research, the 3-D research was developed by JMT and JPM. JPM developed the equations of the Double Jacobian approach in 2-D cylindrical and performed initial implementation. JMT programmed the 2-D cylindrical and 3-D spherical numerical standard version of the code in discussion with JPM and JH. JH implemented the optimized version of the code. JMT designed the tests and analysed the results in discussion with JPM. JMT wrote the manuscript in collaboration with JPM. JH reviewed the manuscript.

Received ; Revised ; Accepted

DOI: xxx/xxxx

ARTICLE TYPE

Shape-preserving finite elements in cylindrical and spherical geometries: The Double Jacobian approach

Jason P. Morgan*[1] | Jorge M. Taramón[1] | Jörg Hasenclever[2]

[1]Department of Earth Sciences, Royal Holloway University of London, Egham, Surrey, UK
[2]Institute of Geophysics, Hamburg University, Hamburg, Germany

Correspondence
*Jason P. Morgan, Department of Earth Sciences, Royal Holloway University of London Egham, Surrey TW20 0EX, United Kingdom. Email: Jason.Morgan@rhul.ac.uk

Present Address
Department of Earth Sciences, Royal Holloway University of London Egham, Surrey TW20 0EX, United Kingdom

Abstract

We present a new technique, the 'Double Jacobian', for solving problems in cylindrical or spherical geometries, for example the Stokes flow problem for studying convection in Earth's mantle. Our approach combines the advantages of working simultaneously in Cartesian and polar or spherical coordinates. The governing matrix equations are kept in Cartesian coordinates, thereby preserving their Cartesian symmetry. However, the element geometry is described as a linear simplex in polar or spherical coordinates, thereby preserving appropriate cylindrical or spherical surfaces and internal interfaces. Isoparametric representations can still be used to define complex surface shapes. Using linear polar or spherical elements allows search routines for triangular or tetrahedral simplexes to rapidly find arbitrary points in terms of their polar or spherical coordinates. The Double Jacobian approach becomes especially powerful when element sizes vary strongly within the mesh, while the exact cylindrical or spherical surfaces or internal interfaces have to be preserved, as happens in several geophysical applications.

KEYWORDS:

1 | INTRODUCTION

Finite element methods have the well-known virtue that they can accurately approximate complex geometries and internal interfaces using isoparametric or even superparametric elements. However, for problems whose natural form has a cylindrical or spherical geometry, other approaches can be more efficient and accurate. A frequently used approach is to directly write the governing equations in polar or spherical coordinates [cf. *Zhong et al.*, 2000]. A significant drawback to this approach is that the governing equations can have useful symmetries in their Cartesian form that are absent when expressed in polar or spherical coordinates. For example, the matrix equations describing incompressible Stokes (creeping) flow of the Earth's mantle in a cylindrical (polar) or spherical geometry are

$$\begin{aligned} Ku + Gp &= f\,, \\ Du \quad\;\; &= 0\,, \end{aligned} \tag{1}$$

or in matrix form:

$$\begin{bmatrix} K & G \\ D & 0 \end{bmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{2}$$

where K is the stiffness matrix, G is the gradient matrix, D is the divergence matrix, f is the force vector, u is the velocity and p is the pressure. In Cartesian coordinates $D = G^{\mathrm{T}}$, leading to a symmetric system of equations, but in both polar and spherical coordinates additional terms appear in the divergence matrix leading to a non-symmetric system of equations. Preserving as exact as possible a boundary shape is especially important in gravitationally influenced flow problems with a natural central point of gravitational attraction, e.g. towards the centre of a cylinder or sphere. In this case, inexact matching of boundary or interface shapes generates spurious flow associated with a nominally 'hydrostatic' state similar to the well-known spurious flow generated by the use of 'too-simple' elements to model hydrostatic loads in sloping box [*Pelletier et al.*, 1989]. Below, we will show examples of this spurious hydrostatic flow mode in cylindrical and spherical geometries, and its cure through use of an exact boundary shape. One may also wish to discretize a cylindrical or spherical volume with 'linear-order' triangles or tetrahedra (e.g. simplex) so that fast search techniques can be used to find tracer particles and/or traceback points for semi-Lagrange advection schemes. Point search techniques become significantly slower and more complex when searching within high-order curved isoparametric elements.

Here we describe a new approach, the 'Double Jacobian' (DJ) approach, that allows us to combine several advantages of working simultaneously in both polar/spherical and Cartesian coordinates. The governing matrix equations are still formulated in Cartesian coordinates, thereby preserving their simplest Cartesian form. However, element geometries are generally described as simplices in polar or spherical coordinates, thereby naturally preserving cylindrical or spherical boundary surfaces and internal interfaces. Note that isoparametric polar or spherical elements can still be used as needed.

The basic idea behind this approach is to compute the local to Cartesian mapping as a two-stage process, hence the name 'Double Jacobian'. The first stage maps from local to polar/spherical coordinates (and back). This mapping is typically to a straight-edged polar or spherical element, for which the Jacobian partial derivatives are constant within the element. The mapping and its inverse are given by straightforward analytical matrix expressions. The second stage maps from polar/spherical to Cartesian coordinates (and back), and is also a simple analytical mapping. The net Jacobian from local finite element coordinates to a cylindrical or spherical element in Cartesian geometry (or any other analytically mapped geometry) is simply the matrix product of two easy-to-compute inverse Jacobian matrices. Because the net Jacobian has an analytical form, it can also be more rapidly computed than a general isoparametric or superparametric finite element mapping.

We will first present the 2-D case to illustrate the basic idea. Afterwards the 3-D case is discussed, which requires an additional step but is otherwise straightforward.

2 | 2-D DOUBLE JACOBIAN

2.1 | First Jacobian

The classical way to construct the stiffness matrices needed for a finite element discretization is to compute the local-coordinate derivatives of the shape functions in a reference element and then calculate the derivatives with respect to global coordinates for each element of the mesh using a local to global Cartesian coordinate mapping [cf. *Hughes*, 1987]. These local and global derivatives may be related in matrix form as

$$\begin{pmatrix} \dfrac{dN_i}{dx} \\[2ex] \dfrac{dN_i}{dz} \end{pmatrix} = \underbrace{\begin{bmatrix} \dfrac{d\xi}{dx} & \dfrac{d\eta}{dx} \\[2ex] \dfrac{d\xi}{dz} & \dfrac{d\eta}{dz} \end{bmatrix}}_{J_{LC}} \begin{pmatrix} \dfrac{dN_i}{d\xi} \\[2ex] \dfrac{dN_i}{d\eta} \end{pmatrix}, \tag{3}$$

where N are the shape functions, i is the local node numbering of the element, (x, z) are the Cartesian coordinates and (ξ, η) are the local coordinates within the reference triangle (Figure 1). The matrix J_{LC} is the Jacobian of the transformation from local to Cartesian coordinates. The shape functions for a 3-node triangle are given by:

$$N_1 = 1 - \xi - \eta\,, \tag{4a}$$

$$N_2 = \xi\,, \tag{4b}$$

$$N_3 = \eta\,. \tag{4c}$$

The Double Jacobian approach uses the standard finite element approach to first map from local to linear polar coordinates. The first Jacobian for a cylindrical (polar) mapping is analogous to the standard Jacobian where x and z are now changed to θ and r respectively, θ being the angle measured from the positive Z axis in clockwise direction and r being the radius. The global

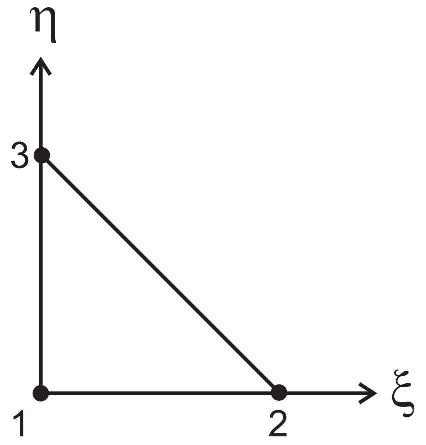


FIGURE 1 Nodal ordering for the reference triangle in the local coordinate frame.

derivatives may be expressed in matrix form as

$$\begin{pmatrix} \dfrac{dN_i}{d\theta} \\ \\ \dfrac{dN_i}{dr} \end{pmatrix} = \underbrace{\begin{bmatrix} \dfrac{d\xi}{d\theta} & \dfrac{d\eta}{d\theta} \\ \\ \dfrac{d\xi}{dr} & \dfrac{d\eta}{dr} \end{bmatrix}}_{J_{LP}} \begin{pmatrix} \dfrac{dN_i}{d\xi} \\ \\ \dfrac{dN_i}{d\eta} \end{pmatrix}, \tag{5}$$

where N_i are the shape functions and J_{LP} is the Jacobian of the transformation from local to polar coordinates. The derivatives of these shape functions with respect to local coordinates can be computed explicitly. However, the terms of the Jacobian J_{LP} cannot be directly computed since explicit expressions for $\xi(\theta, r)$ and $\eta(\theta, r)$ do not exist. A wonderful 'trick' in finite element programming (discovered by Bruce Irons in the mid 60s) is to make use of the inverse coordinate transformation

$$\begin{pmatrix} \dfrac{dN_i}{d\xi} \\ \\ \dfrac{dN_i}{d\eta} \end{pmatrix} = \underbrace{\begin{bmatrix} \dfrac{d\theta}{d\xi} & \dfrac{dr}{d\xi} \\ \\ \dfrac{d\theta}{d\eta} & \dfrac{dr}{d\eta} \end{bmatrix}}_{J_{PL}} \begin{pmatrix} \dfrac{dN_i}{d\theta} \\ \\ \dfrac{dN_i}{dr} \end{pmatrix}, \tag{6}$$

where J_{PL} is the Jacobian of the transformation from polar to local coordinates. From equations (5) and (6), $J_{LP} \equiv (J_{PL})^{-1}$. The inverse of the Jacobian from polar to local coordinates is given by

$$(J_{PL})^{-1} = \frac{1}{|J_{PL}|} \begin{bmatrix} \dfrac{dr}{d\eta} & -\dfrac{dr}{d\xi} \\ \\ -\dfrac{d\theta}{d\eta} & \dfrac{d\theta}{d\xi} \end{bmatrix}. \tag{7}$$

The polar coordinates for each element are related to local coordinates through the the shape functions:

$$\theta(\xi, \eta) = \sum_{i=1}^{3} N_i(\xi, \eta)\, \theta_i\,, \tag{8a}$$

$$r(\xi, \eta) = \sum_{i=1}^{3} N_i(\xi, \eta)\, r_i\,. \tag{8b}$$

Differentiating equations (8a) and (8b):

$$\frac{d\theta}{d\xi} = \theta_2 - \theta_1 \equiv \theta_{21}\,, \tag{9a}$$

$$\frac{d\theta}{d\eta} = \theta_3 - \theta_1 \equiv \theta_{31}\,, \tag{9b}$$

$$\frac{dr}{d\xi} = r_2 - r_1 \equiv r_{21}\,, \tag{9c}$$

$$\frac{dr}{d\eta} = r_3 - r_1 \equiv r_{31}\,. \tag{9d}$$

Finally, the inverse of the Jacobian from polar to local coordinates is given by combining equations (9a-d) and (7):

$$(J_{PL})^{-1} = \frac{1}{\theta_{21}r_{31} - r_{21}\theta_{31}} \begin{bmatrix} r_{31} & -r_{21} \\ -\theta_{31} & \theta_{21} \end{bmatrix}\,, \tag{10}$$

which is the standard finite element mapping between local coordinates and an arbitrary triangular element defined in linear polar coordinates.

2.2 | Second Jacobian

The second Jacobian in the Double Jacobian method is the analytical mapping from polar coordinates to Cartesian coordinates. The derivatives expressed in matrix form are given by

$$\begin{pmatrix} \frac{dN_i}{dx} \\ \\ \frac{dN_i}{dz} \end{pmatrix} = \underbrace{\begin{bmatrix} \frac{d\theta}{dx} & \frac{dr}{dx} \\ \\ \frac{d\theta}{dz} & \frac{dr}{dz} \end{bmatrix}}_{J_{PC}} \begin{pmatrix} \frac{dN_i}{d\theta} \\ \\ \frac{dN_i}{dr} \end{pmatrix}, \tag{11}$$

where J_{PC} is the Jacobian from polar to Cartesian coordinates. The analytical expressions for $\theta(x,z)$ and $r(x,z)$ are known, however, it is again easier to use the inverse transformation

$$\begin{pmatrix} \frac{dN_i}{d\theta} \\ \\ \frac{dN_i}{dr} \end{pmatrix} = \underbrace{\begin{bmatrix} \frac{dx}{d\theta} & \frac{dz}{d\theta} \\ \\ \frac{dx}{dr} & \frac{dz}{dr} \end{bmatrix}}_{J_{CP}} \begin{pmatrix} \frac{dN_i}{dx} \\ \\ \frac{dN_i}{dz} \end{pmatrix}, \tag{12}$$

where J_{CP} is the Jacobian from Cartesian to polar coordinates. From equations (11) and (12), $J_{PC} \equiv (J_{CP})^{-1}$. The inverse of the Jacobian from Cartesian to polar coordinates is given by

$$(J_{CP})^{-1} = \frac{1}{|J_{CP}|} \begin{bmatrix} \frac{dz}{dr} & -\frac{dz}{d\theta} \\ \\ -\frac{dx}{dr} & \frac{dx}{d\theta} \end{bmatrix}. \tag{13}$$

Cartesian coordinates are related to polar coordinates by:

$$x(\theta, r) = r \sin\theta\,, \tag{14a}$$

$$z(\theta, r) = r \cos\theta\,. \tag{14b}$$

Differentiating equations (14a) and (14b):

$$\frac{dx}{d\theta} = r\cos\theta\,, \tag{15a}$$

$$\frac{dx}{dr} = \sin\theta\,, \tag{15b}$$

$$\frac{dz}{d\theta} = -r\sin\theta\,, \tag{15c}$$

$$\frac{dz}{dr} = \cos\theta\,. \tag{15d}$$

The inverse of the Jacobian from Cartesian coordinates to polar coordinates can also be written as a function of polar coordinates by combining equations (15a-d) and (13):

$$(J_{CP})^{-1} = \frac{1}{r}\begin{bmatrix} \cos\theta & r\sin\theta \\ -\sin\theta & r\cos\theta \end{bmatrix}, \tag{16}$$

where θ and r are evaluated at each integration point. Making use of the matrix product of the two inverse Jacobians, global Cartesian derivatives can be expressed as a matrix product of the local derivatives in the local to polar and polar to Cartesian coordinate mappings. Substituting equation (5) into equation (11) yields

$$\begin{pmatrix} \dfrac{dN_i}{dx} \\ \\ \dfrac{dN_i}{dz} \end{pmatrix} = \underbrace{\begin{bmatrix} \dfrac{d\theta}{dx} & \dfrac{dr}{dx} \\ \\ \dfrac{d\theta}{dz} & \dfrac{dr}{dz} \end{bmatrix}}_{J_{PC}} \underbrace{\begin{bmatrix} \dfrac{d\xi}{d\theta} & \dfrac{d\eta}{d\theta} \\ \\ \dfrac{d\xi}{dr} & \dfrac{d\eta}{dr} \end{bmatrix}}_{J_{LP}} \begin{pmatrix} \dfrac{dN_i}{d\xi} \\ \\ \dfrac{dN_i}{d\eta} \end{pmatrix}, \tag{17}$$

which is equivalent to

$$\begin{pmatrix} \dfrac{dN_i}{dx} \\ \\ \dfrac{dN_i}{dz} \end{pmatrix} = (J_{CP})^{-1}(J_{PL})^{-1} \begin{pmatrix} \dfrac{dN_i}{d\xi} \\ \\ \dfrac{dN_i}{d\eta} \end{pmatrix}. \tag{18}$$

Substituting equations (10) and (16) into equation (18), the analytical expression for this mapping is

$$\begin{pmatrix} \dfrac{dN_i}{dx} \\ \\ \dfrac{dN_i}{dz} \end{pmatrix} = \frac{1}{\theta_{21}r_{31} - r_{21}\theta_{31}} \begin{bmatrix} r_{31}\dfrac{\cos\theta}{r} - \theta_{31}\sin\theta & -r_{21}\dfrac{\cos\theta}{r} + \theta_{21}\sin\theta \\ \\ -r_{31}\dfrac{\sin\theta}{r} - \theta_{31}\cos\theta & r_{21}\dfrac{\sin\theta}{r} + \theta_{21}\cos\theta \end{bmatrix} \begin{pmatrix} \dfrac{dN_i}{d\xi} \\ \\ \dfrac{dN_i}{d\eta} \end{pmatrix}. \tag{19}$$

Note that the 2-D Double Jacobian approach (DJ) will ensure a perfect mapping to the circular arc-edges of the elements of a cylindrical annulus mesh. In the mapping from local coordinates to polar coordinates (first Jacobian of the Double Jacobian), the elements are described in polar coordinates in which their edges are straight line segments (Figure 2a). Any point on the edge of a boundary element of the mesh is mapped to its true position along a circular arc (Figure 2b). Neither the linear Jacobian for straight edge elements (LIN) nor the isopametric Jacobian for quadratic elements (ISO) can exactly map to a circular arc-shape since they use Cartesian linear and quadratic polynomial approximations, respectively (Figure 2d and Figure 2f, respectively). For this reason a superparametric cubic or quartic mapping is sometimes chosen to better approximate curved boundaries. The third column in Figure 2 shows the distance in km between the edge shape computed in Cartesian coordinates (red line for DJ, blue line for LIN and green line for ISO) and the true shape of a circular arc edge. The element shown in Figure 2 has an average edge length of 2000 km. The boundary of the ISO mapping differs from the DJ mapping by a few kilometres (compare Figure 2g and Figure 2c). While seemingly small, this inaccuracy can negatively influence the solution for viscous (Stokes) flow within a cylindrical annulus mesh that contains coarsely meshed regions, as will be shown below.

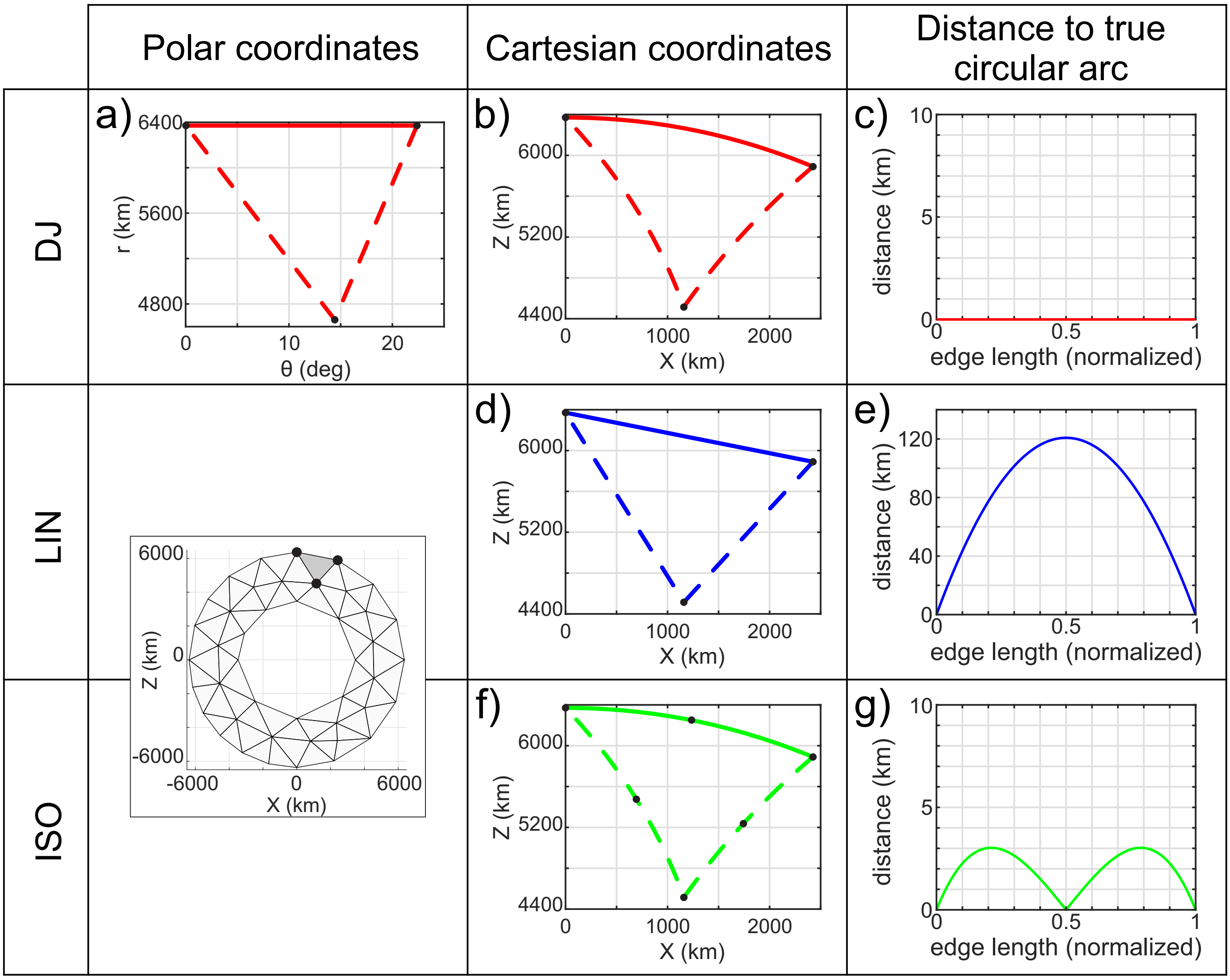


FIGURE 2 Cartesian and polar mapping for the edge shape of an arbitrary triangle along the surface of the region. The first two columns show the edge shape calculated by: 3-linear polar mapping (red line), which is the true circular arc shape (DJ), 3-node linear Cartesian mapping (LIN) (blue line) and 6-node isoparametric quadratic Cartesian mapping (ISO) (green line). The location of the triangle is shown in the inset in the first column. Black points represent the element nodes. The third column shows the distance between the edge shape given by the Cartesian mappings and the true circular arc shape.

2.3 | 2-D Standard Jacobian vs 2-D Double Jacobian

We first test the Double Jacobian formulation by comparing the stiffness matrices given by both the standard Jacobian (LIN and ISO) and Double Jacobian (DJ) formulations. Benchmarking has been done for the boundary elements of cylindrical annulus meshes with different resolutions, from an element edge length $l_0 = 2000$ km to $l_0 = 50$ km (Figure 3a). The domain of the Cartesian coordinate system is $\mathbb{R}^2$ whereas the domain of the polar coordinate system is $\theta \in [0, 2\pi)$ and $r \in [0, \infty)$. Due to the cyclic property of the polar coordinate system, elements that cross the positive Z axis have a discontinuity in the θ coordinate of their vertices. This issue is solved by a 180° rotation of all elements that cross $\theta = 2\pi$, computing the Jacobian and stiffness matrices in the rotated frame, and then performing a back-transformation into the unrotated frame. The general rule for applying a rotation matrix to vectors and tensors is:

$$v_{rot} = \boldsymbol{R} v \,, \tag{20a}$$

$$\sigma_{rot} = \boldsymbol{R} \sigma \boldsymbol{R}^{\mathsf{T}} \,, \tag{20b}$$

where v is a vector, σ is a tensor and $\boldsymbol{R}$ is the rotation matrix. The inverse transformation is given by:

$$v = \boldsymbol{R}^{\mathsf{T}} v_{rot} \,, \tag{21a}$$

$$\sigma = \boldsymbol{R}^{\mathsf{T}} \sigma_{rot} \boldsymbol{R} \,. \tag{21b}$$

The matrix defining a 180° rotation in 2-D Cartesian coordinates is

$$\boldsymbol{R}_{180} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} . \tag{22}$$

Considering a Stokes problem with Taylor-Hood elements, the sizes of the element matrices $\boldsymbol{K}_e$, $\boldsymbol{G}_e$ and vector $\boldsymbol{f}_e$ for each 6-node quadratic-velocity and 3-node linear-pressure triangular element ($\mathrm{P_2P_1}$) are 12x12, 12x3 and 12x1 respectively. In terms of applying a rotation, the $\boldsymbol{G}_e$ matrix is treated as three column vectors since it is not a square matrix. In order to recover the matrices $\boldsymbol{K}_e$, $\boldsymbol{G}_e$ and vector $\boldsymbol{f}_e$ from the rotated counterparts we apply the rotation matrix to each pair of two degrees of freedom (dof) for each node:

$$\boldsymbol{K}_e = \boldsymbol{R}_{180}^{\mathsf{T}} \boldsymbol{K}_e^{rot} \boldsymbol{R}_{180} \,, \tag{23a}$$

$$\boldsymbol{G}_e = \boldsymbol{R}_{180}^{\mathsf{T}} \boldsymbol{G}_e^{rot} \,, \tag{23b}$$

$$\boldsymbol{f}_e = \boldsymbol{R}_{180}^{\mathsf{T}} \boldsymbol{f}_e^{rot} \,. \tag{23c}$$

For simplicity, we show here the unrotated matrices and vector for one node with two dof. The rotated stiffness matrix, gradient matrix and force vector for velocity node one of an element are:

$$\boldsymbol{K}_e^{rot}(node\ 1) = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} , \tag{24a}$$

$$\boldsymbol{G}_e^{rot}(node\ 1) = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \end{bmatrix} , \tag{24b}$$

$$\boldsymbol{f}_e^{rot}(node\ 1) = \begin{pmatrix} f_{11} \\ f_{21} \end{pmatrix} . \tag{24c}$$

Substituting equations (24a) to (24c) in equations (23a) to (23c) respectively:

$$\boldsymbol{K}_e(node\ 1) = \boldsymbol{K}_e^{rot}(node\ 1) \,, \tag{25a}$$

$$\boldsymbol{G}_e(node\ 1) = -\boldsymbol{G}_e^{rot}(node\ 1) \,, \tag{25b}$$

$$\boldsymbol{f}_e(node\ 1) = -\boldsymbol{f}_e^{rot}(node\ 1) \,. \tag{25c}$$

Note that the stiffness matrix $\boldsymbol{K}_e$ remains the same due to the symmetry of this rotation. Figure 3b shows the difference between the stiffness matrix $\boldsymbol{K}_e$ computed with $\mathrm{P_2P_1}$LIN and $\mathrm{P_2P_1}$DJ elements. This difference is calculated as:

$$\epsilon_{STD-DJ} = \frac{max\left(\left|\boldsymbol{K}_e{}^{STD} - \boldsymbol{K}_e{}^{DJ}\right|\right)}{max\left(\left|\boldsymbol{K}_e{}^{DJ}\right|\right)} . \tag{26}$$

The stiffness matrices of both approaches become more similar when elements are smaller. For example, LIN applied to a mesh with $l_0 = 150$ km (cyan line in Figure 3b) gives stiffness matrices which differ from the ones given by DJ by less than

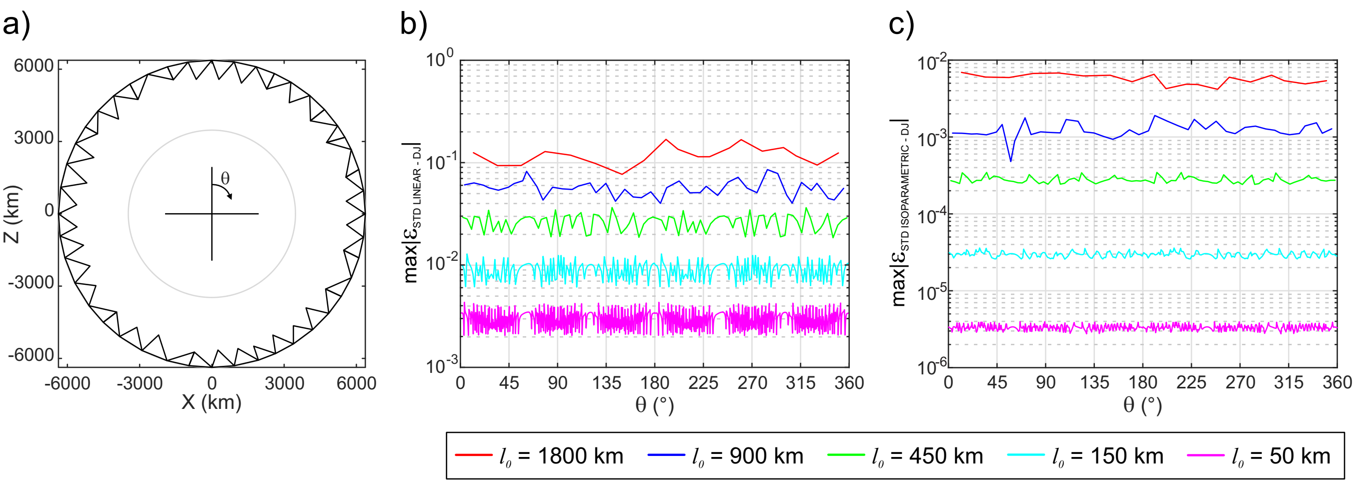


FIGURE 3 (a) Boundary triangular elements of an unstructured cylindrical annulus mesh with element edge length $l_0 = 900$ km. (b) The maximum relative difference between K_e computed with P_2P_1LIN elements and K_e computed with P_2P_1DJ elements. This is computed for different mesh resolutions (colours in legend) as a function of the colatitude θ of the barycentres of the boundary elements. (c) The maximum relative difference between K_e computed with P_2P_1ISO elements and K_e computed with P_2P_1DJ elements.

1%. However, in the case of a mesh with $l_0 = 1800$ km (red line in Figure 3b), the stiffness matrices given by both Jacobians differ by 12% on average. P_2P_1LIN elements provide a poor mapping to the curvature of the elements in coarse meshes, and the resulting stiffness matrices are poor approximations. For P_2P_1ISO elements, smaller differences remain when compared to P_2P_1DJ elements (Figure 3c). These smaller differences result from the better (but still imperfect) quadratic Cartesian polynomial approximation to the true circular arc used by P_2P_1ISO elements (Figure 2). Fluctuations in the lines of Figure 3b and Figure 3c are directly related to small variations in element size.

2-D quadratic elements with straight edges

The general quadratic-order triangular element in a Cartesian coordinate system has nodes added to the middle of each edge. The Cartesian coordinates of the mid-edge nodes m_i for a straight-edged element are given by:

$$\boldsymbol{m}_4 = \left(\frac{1}{2}(x_1 + x_2), \frac{1}{2}(z_1 + z_2)\right) , \tag{27a}$$

$$\boldsymbol{m}_5 = \left(\frac{1}{2}(x_2 + x_3), \frac{1}{2}(z_2 + z_3)\right) , \tag{27b}$$

$$\boldsymbol{m}_6 = \left(\frac{1}{2}(x_3 + x_1), \frac{1}{2}(z_3 + z_1)\right) . \tag{27c}$$

where subscripts are the local node numbering of the element, being 1, 2 and 3 for the vertices and 4, 5 and 6 for the mid-edge nodes. Calculating the mid-edge node positions of a quadratic isoparametric element using equations (27a) to (27c) will lead to quadratic triangular elements with straight edges in the Cartesian coordinate system.

2-D quadratic elements computed in polar coordinates

In cylindrical geometries it may be more useful to add mid-edge nodes using the polar coordinate system so that the mid-edge nodes of boundary edges lie along a cylindrical boundary. To do so, the Cartesian coordinates of the vertices of each element are transformed into polar coordinates, and then mid-edge nodes μ_i in polar coordinates for each element are computed as:

$$\boldsymbol{\mu}_4 = \left(\frac{1}{2}(\theta_1 + \theta_2), \frac{1}{2}(r_1 + r_2)\right) , \tag{28a}$$

$$\boldsymbol{\mu}_5 = \left(\frac{1}{2}(\theta_2 + \theta_3), \frac{1}{2}(r_2 + r_3)\right) , \tag{28b}$$

$$\boldsymbol{\mu}_6 = \left(\frac{1}{2}(\theta_3 + \theta_1), \frac{1}{2}(r_3 + r_1)\right) . \tag{28c}$$

Then the polar vertex and mid-edge coordinates are transformed back into Cartesian coordinates. Equations (28a) to (28c) can be applied to the whole domain of the cylindrical annulus mesh except for elements that cross $\theta = 2\pi$, for which the following recipe can be used (Figure 4). First, make a 180° rotation in Cartesian coordinates (Figure 4b). Second, transform to polar coordinates and compute mid-edge nodes following equations (28a) to (28c) (Figure 4c). Third, transform back to Cartesian coordinates (Figure 4d). Fourth, make a 180° rotation in Cartesian coordinates (Figure 4e).

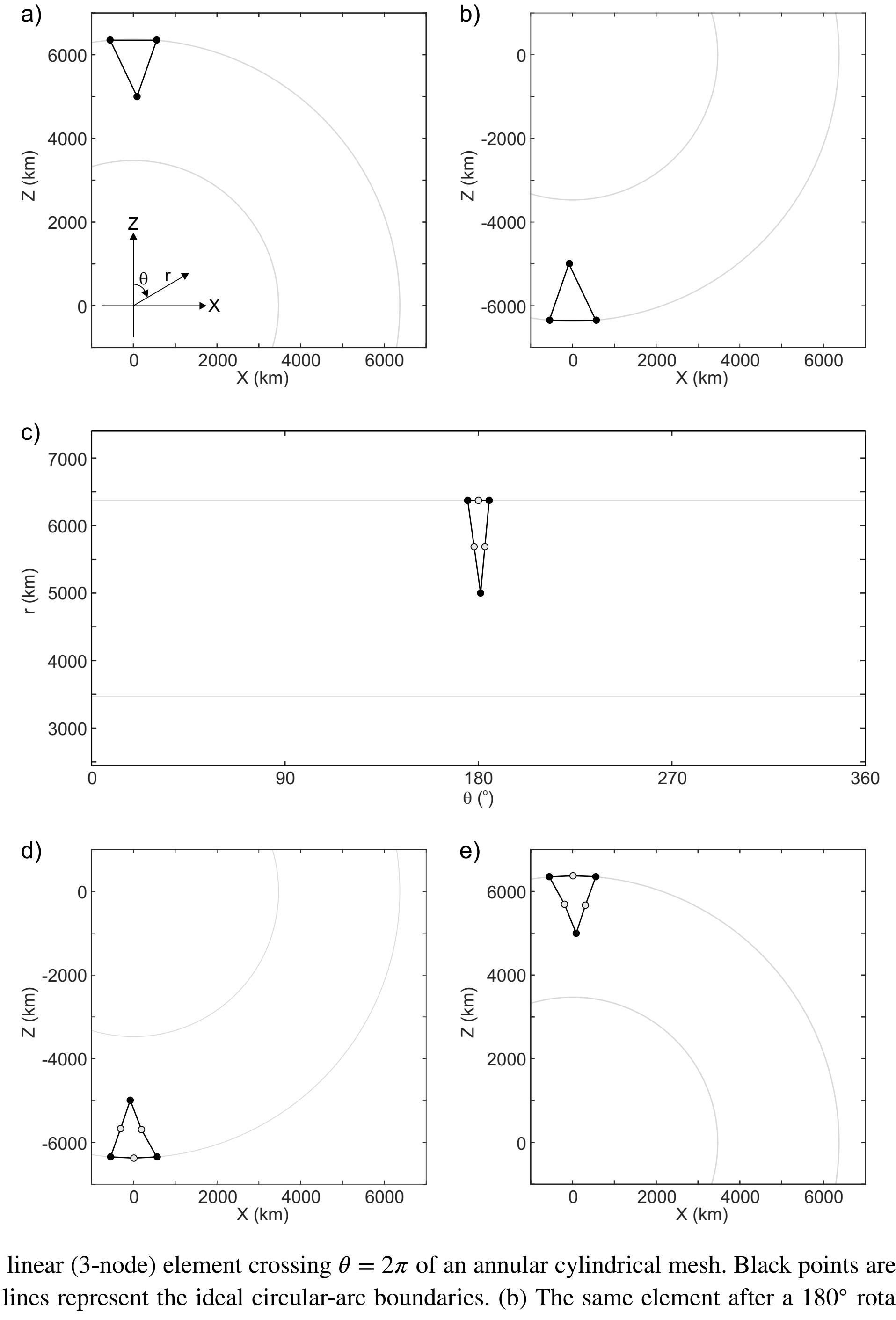


FIGURE 4 (a) A linear (3-node) element crossing $\theta = 2\pi$ of an annular cylindrical mesh. Black points are the vertices of the element and grey lines represent the ideal circular-arc boundaries. (b) The same element after a 180° rotation. (c) The same rotated element defined in a polar coordinate system. Mid-edge nodes (grey points) have been computed following equations (28a) to (28c) to obtain a quadratic-order element. (d) Rotated quadratic element shown in Cartesian coordinates, where the mid-edge node of the boundary edge lies along the annular boundary. (e) Quadratic element in the original position after undoing the 180° rotation used to determine its stiffnes matrix.

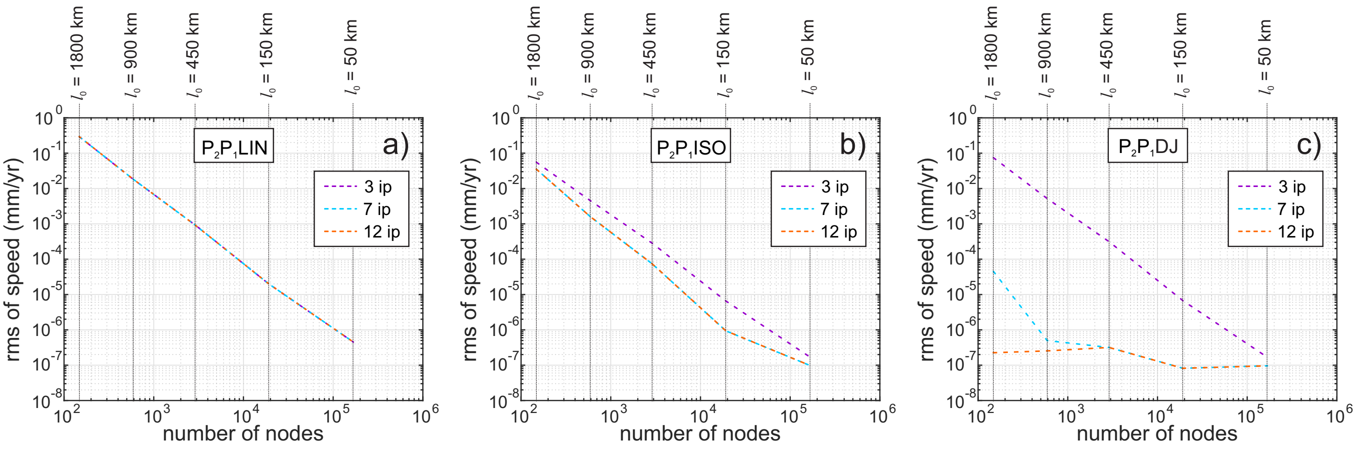


FIGURE 5 Root mean square of the speed for the 'no-flow' test in a cylindrical annulus using: (a) P_1P_1LIN, (b) P_2P_1ISO and (c) P_2P_1DJ elements. Results are shown as a function of the number of mesh nodes (X axis) and integration points (ip) per element (dashed colour lines in legend).

Numerical integration of Double Jacobian elements

Because the Jacobian for a polar/spherical to Cartesian mapping is not a low order polynomial function, care is needed to accurately evaluate these numerical integrations. In other words, the Jacobian mapping cannot be exactly integrated by any Gaussian integration so that element integrations benefit from high-order numerical integration. We will see this effect in the examples below. Note, however, that the time-savings resulting from the fact that shape-function derivatives can be analytically evaluated at each integration point more than compensates for the extra time consumed by the evaluation of more integration points.

Results

Differences between the standard Jacobian (LIN and ISO) and Double Jacobian (DJ) approaches are most apparent for coarse meshes due to their differently-shaped boundaries. We performed three tests to assess the accuracy of the 2-D Double Jacobian formulation: (1) the hydrostatic 'no-flow' test proposed by *Pelletier et al.* [1989], (2) a viscous flow problem solved for a simple case in which density and viscosity are uniform and (3) the Method of Manufactured Solutions (MMS) [*Salari and Knupp*, 2000; *Roache*, 2002] applied to the steady state diffusion equation. We test using M2TRI [*Hasenclever*, 2010; *Hasenclever et al.*, 2011], a numerical code written in MATLAB (http://www.mathworks.com) to solve for the 2-D thermo-mechanical viscous flow of the mantle using the standard Jacobian method in Cartesian coordinates (LIN and ISO). The 2-D Double Jacobian approach has been implemented in a cylinder-coordinate version of M2TRI. For performance reasons both codes use the vectorized 'blocking' technique for assembly the stiffness matrices described in *Dabrowski et al.* [2008].

A simple and surprisingly robust test for the potential accuracy of a finite element flow solver is to test the code's ability to predict zero flow for a 'hydrostatic' no-flow state [*Pelletier et al.*, 1989]. To assess general performance, this test is best conducted on a irregular mesh because regular 'finite-difference-like' grids can sometimes mask an element's poor performance [*Pelletier et al.*, 1989]. Cylindrical and spherical geometries with spatially varying gravity directions prove to be more difficult to obtain accurate no-flow solutions than Cartesian discretizations in which gravity only acts in the vertical direction. A 'cylindrical no-flow' test was performed using the P_2P_1LIN, P_2P_1ISO and P_2P_1DJ elements (Figure 5). The test has no-slip boundary conditions and uniform density and viscosity. The resolution of the meshes varies from l_0 = 1800 km to l_0 = 50 km and the number of integration points (ip) used are 3, 7 and 12, describing 2^{nd}, 5^{th} and 6^{th} order accurate polynomial integration, respectively. Figure 5 shows the root mean square of the speed as a function of the number of mesh nodes. P_2P_1DJ gives a more accurate zero no-flow solution than P_2P_1LIN or P_2P_1ISO, if a sufficiently accurate numerical integration rule is employed.

We have also performed a variant of the 'no-flow' test (Figure 6). The setup consists of two concentric layers with different densities, where the deeper layer is 100 kg/m^3 denser than the shallower layer. Again viscosity is uniform and no-slip boundary conditions are applied. Again for coarse meshes P_2P_1DJ with appropriate numerical integration gives a more accurate zero flow than do either P_2P_1LIN or P_2P_1ISO.

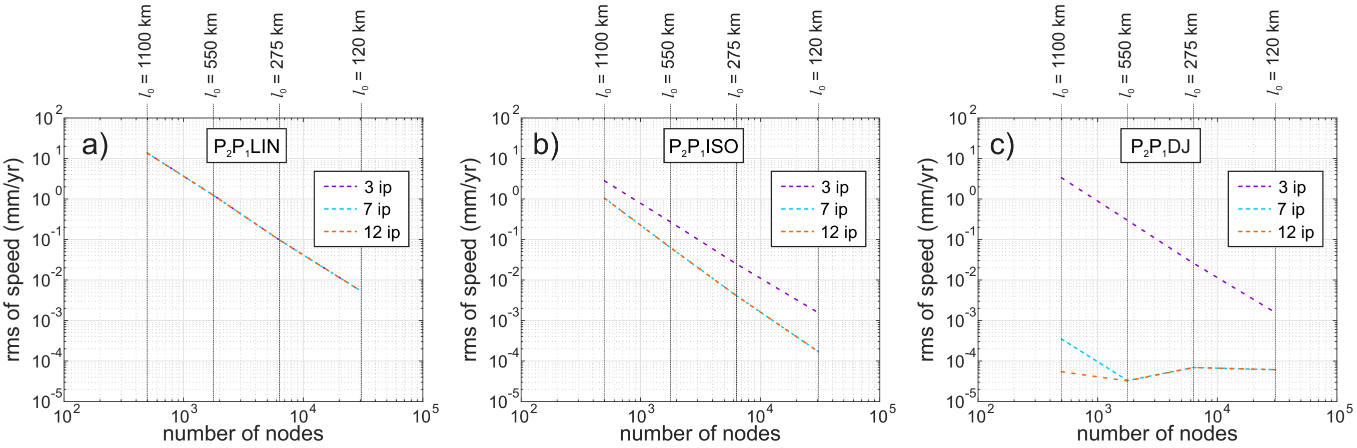


FIGURE 6 Same 'no-flow' test as in Figure 5 for two concentric layers with a uniform density layer that overlies a denser layer.

Figure 7 shows the no-flow solution using 12 integration points for the cases shown in Figure 6. The zero flow solution obtained using P_2P_1LIN elements is not well resolved for any mesh resolution. The zero flow solution obtained using P_2P_1ISO elements is not well resolved except for the highest resolution considered in the test (l_0 = 120 km ; Figure 7k). When using the Double Jacobian method (DJ) the zero flow solution is resolved independent of mesh resolution.

The second test consists of solving a flow problem for a setup in which the velocity boundary conditions are free slip for the inner 'core-mantle' boundary and a prescribed horizontal velocity distribution along the outer surface. The prescribed velocity in the northern hemisphere is uniform in the counter-clockwise direction (v_θ = -40 mm/yr) whereas in the southern hemisphere the velocity is the same magnitude in the clockwise direction (v_θ = 40 mm/yr). This applied velocity distribution creates a divergence point at $\theta = 90°$ simulating rifting, and a convergence point at $\theta = 270°$ simulating subduction. In order to see how accurate the flow solutions are as a function of mapping method and mesh resolution, we would ideally compare them to an known analytical solution, however to our knowledge no closed-form analytical solution exists for this example. For comparison purposes, we consider the numerical solution for a mesh with l_0 = 10 km ($4.3 \cdot 10^6$ nodes) to be a good approximation to the exact solution. The implied solution as a function of mesh resolution is calculated by interpolating the flow field at all node locations of the high-resolution mesh. Figure 8 shows the root mean square (rms) of the deviation in velocity between these solutions and the high-resolution mesh solution as a function of mesh resolution, in all cases using 7 integration points per element. P_2P_1LIN elements gives deviations > 1 mm/yr unless the mesh has more than $2 \cdot 10^3$ nodes (black line in Figure 8). P_2P_1ISO and P_2P_1DJ elements yield the same deviation for meshes with more than 10^3 nodes (green line and red line respectively in Figure 8). DJ again works better for coarse meshes with resolutions of 1800 km because it can exactly map the circular-arc shape of the boundaries. This test was also conducted using 3 and 12 integration points per element (not shown). In the case of 3 integration points, P_2P_1DJ gives the same deviation as P_2P_1LIN and P_2P_1ISO for all mesh resolutions because this numerical integration is not accurate enough for the Double Jacobian method. For 12 integration points, the P_2P_1DJ elements gives results very similar to those shown in Figure 8.

Figure 9 shows the flow solution for three meshes with different resolutions (rows) using three different mapping methods (columns) and 7 integration points per element. As noted above, mesh resolution plays an important role in determining how well this flow can be resolved by these different mapping methods. The solutions obtained using P_2P_1LIN elements are not well resolved for the meshes with l_0 = 1800 km and l_0 = 900 km (Figures 9a and 9d), reaching maximum speeds of 210 mm/yr and 55 mm/yr, respectively. However, the solution for this large scale flow problem becomes well resolved for a mesh with l_0 = 450 km (Figure 9g). When using P_2P_1ISO and P_2P_1DJ elements, the solution is well resolved for meshes with l_0 = 900 km and l_0 = 450 km (Figures 9e, 9f, 9h and 9i). However, for the mesh with l_0 = 1800 km (Figures 9b and 9c) observable differences remain between the flows computed by P_2P_1ISO or P_2P_1DJ approaches near the inner boundary.

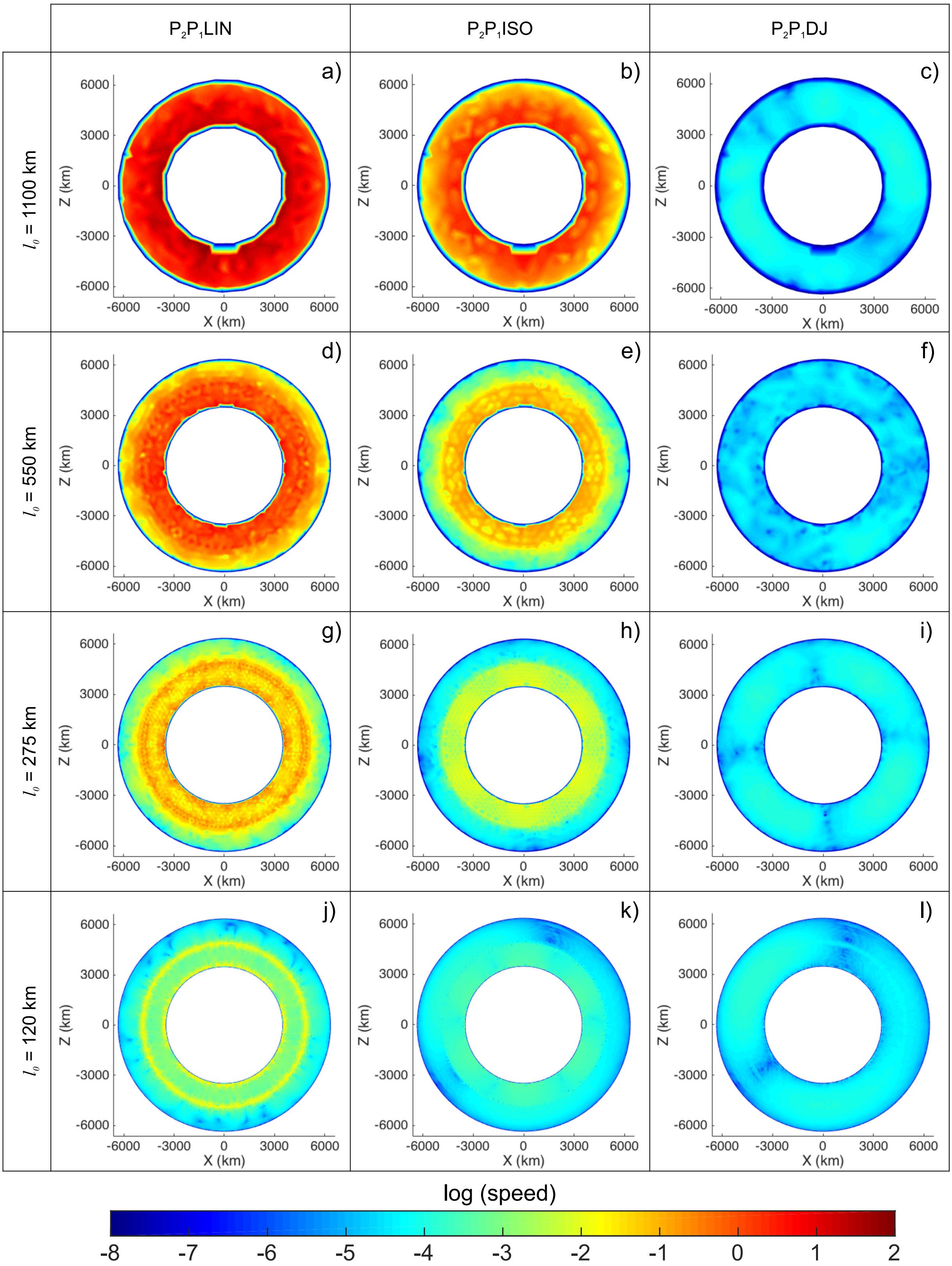


FIGURE 7 Viscous flow solution for a model with uniform viscosity and two concentric layers with the deeper layer being 100 kg/m^3 denser than the shallower layer. The boundary condition is no-slip. The flow is solved on four meshes with different resolutions (label on left side of each row). P$_2$P$_1$LIN, P$_2$P$_1$ISO, and P$_2$P$_1$DJ elements are used to solve the flow (label on the top of each column). Colour represents logarithm of speed.

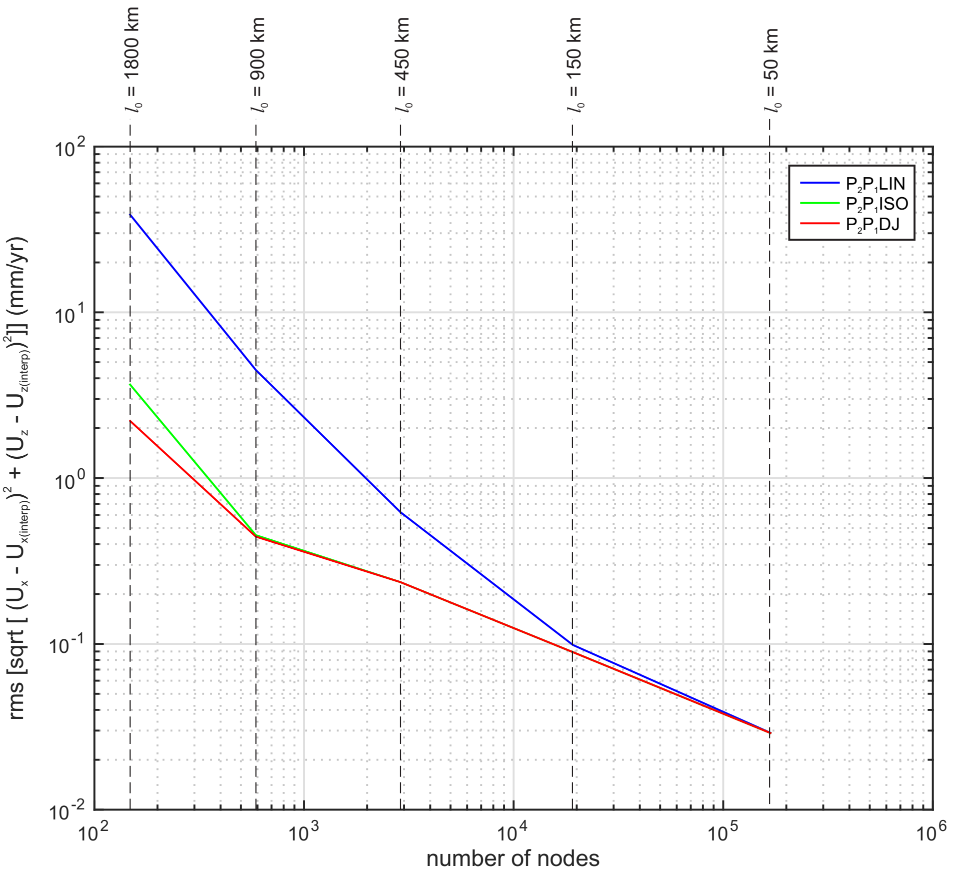


FIGURE 8 Root mean square (rms) of the deviation in velocity at each node between the solution given by coarser cylindrical annulus meshes and the high-resolution mesh solution using the P_2P_1LIN, P_2P_1ISO, and P_2P_1DJ elements as a function of mesh size. Flow solutions on these coarser meshes are interpolated to the node locations of the high-resolution mesh ($4.3 \cdot 10^6$ nodes) solution to compute the rms of the deviation in velocity. All calculations use 7 integration points per element.

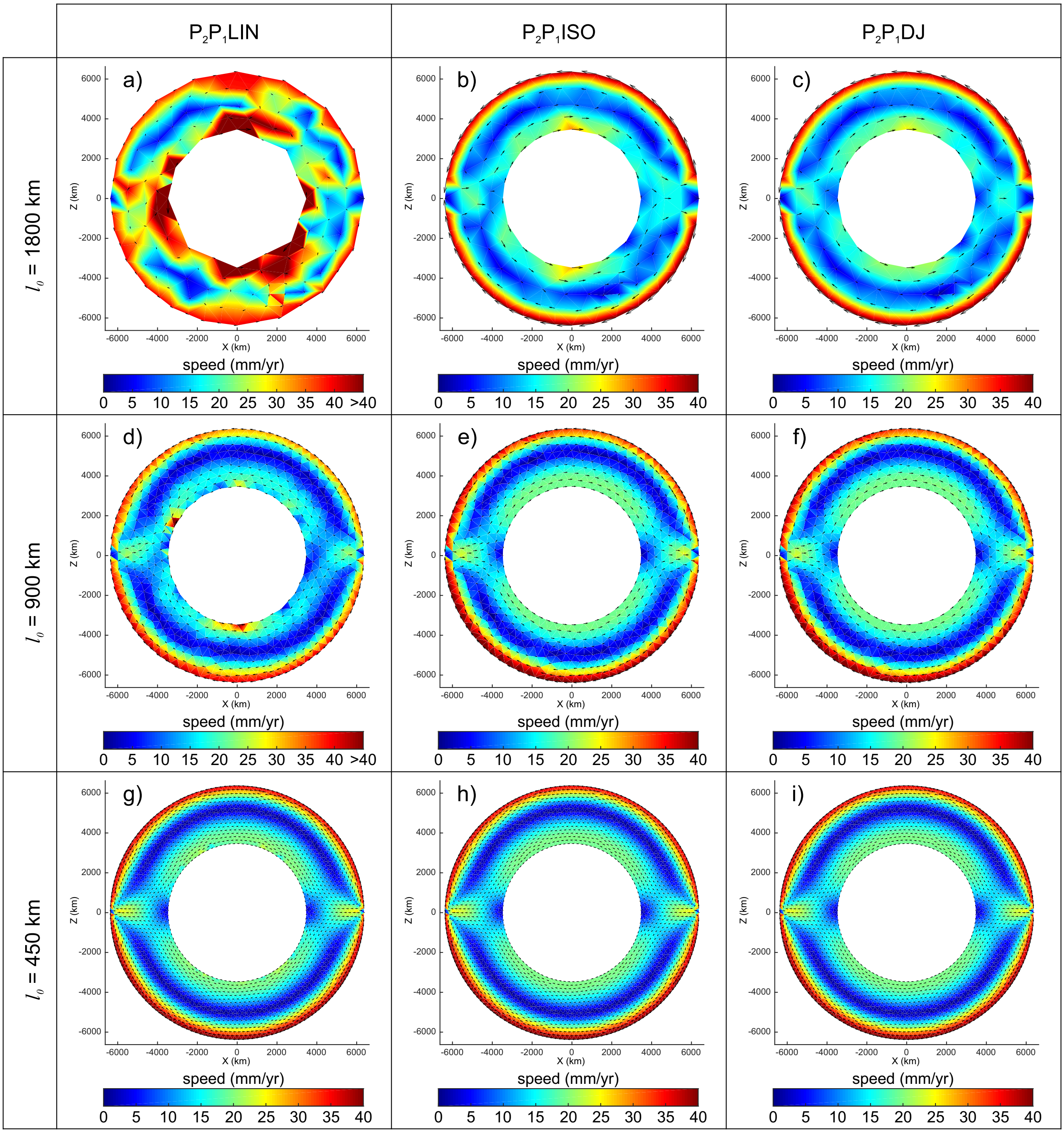


FIGURE 9 Viscous flow solution for a model with a uniform density and viscosity layer driven by imposed surface motion. The boundary conditions are free slip along the inner surface boundary and a horizontal prescribed velocity at the outer surface. The prescribed velocity in the northern hemisphere is in counter-clockwise direction (v_θ = -40 mm/yr) whereas in the southern hemisphere the velocity is in clockwise direction (v_θ = 40 mm/yr). The flow is solved on three meshes with different resolutions (label on left side of each row). The P_2P_1LIN, P_2P_1ISO, and P_2P_1DJ elements are used to solve the flow (label on the top of each column). Colour represents speed and black arrows show velocity vectors.

The third test consists of using the MMS to verify the order of accuracy of the 2-D code. This method provides a straightforward procedure to generate analytical solutions for benchmarking. The basic idea is to manufacture a non-trivial exact solution. In this test we applied the MMS to the steady state diffusion equation

$$\kappa \nabla^2 T = \Psi\,, \tag{29}$$

where κ is the thermal diffusivity ($\kappa = 1$), T is the tempreature field and Ψ is the source term. We use continuous Taylor-Hood triangular elements with a 6-node quadratic-Temperature interpolation (P_2). The domain is a cylindrical annulus with inner and outer radii equal to 0.5 and 1, respectively. The chosen solution with inherent radial symmetry is:

$$T_{mms} = e^{-\left(x^2+y^2\right)}\,. \tag{30}$$

This solution also defines the Dirichlet boundary conditions. The solution is passed through equation (29) to manufacture a compatible source term:

$$\Psi_{mms} = -\kappa \left[-4 + 4x^2 + 4y^2\right]\, e^{-\left(x^2+y^2\right)}\,. \tag{31}$$

Then we compute discrete solutions by solving

$$\kappa \nabla^2 T = \Psi_{mms}\,. \tag{32}$$

To verify the code and algorithm, we compute the L_2 norm given by:

$$L_2 = \sqrt{\int \left(T_{mms} - T\right)^2 dV}\,. \tag{33}$$

Equation (33) is discretized by computing the normalized global error or RMS error:

$$e_2 = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(T_i^{mms} - T_i\right)^2}\,, \tag{34}$$

where T_i^{mms} is the manufactured solution evaluated at $\left(x_i, z_i\right)$, i is the index of the discrete solution location, n is the total number of nodes and T_i is the discrete solution. Figure 10 shows the RMS error of temperature as a function of the average element size $< h >$ which in 2-D is defined by:

$$< h >= \left(\frac{A}{N}\right)^{1/2}\,, \tag{35}$$

where A is the area of the domain and N is the total number of elements. The error is seen to decrease quadratically for P_2LIN and cubically for P_2ISO and P_2DJ elements as the mesh is refined. Again the P_2DJ elements gives better results for coarser meshes as it exactly maps the circular-arc shape of boundaries. Note that the poor ability of the P_2LIN elements to map the boundaries of the region means they have a suboptimal h^2 vs. h^3 convergence rate even though they use the same second-order P_2 element interpolation as do the P_2ISO and P_2DJ elements.

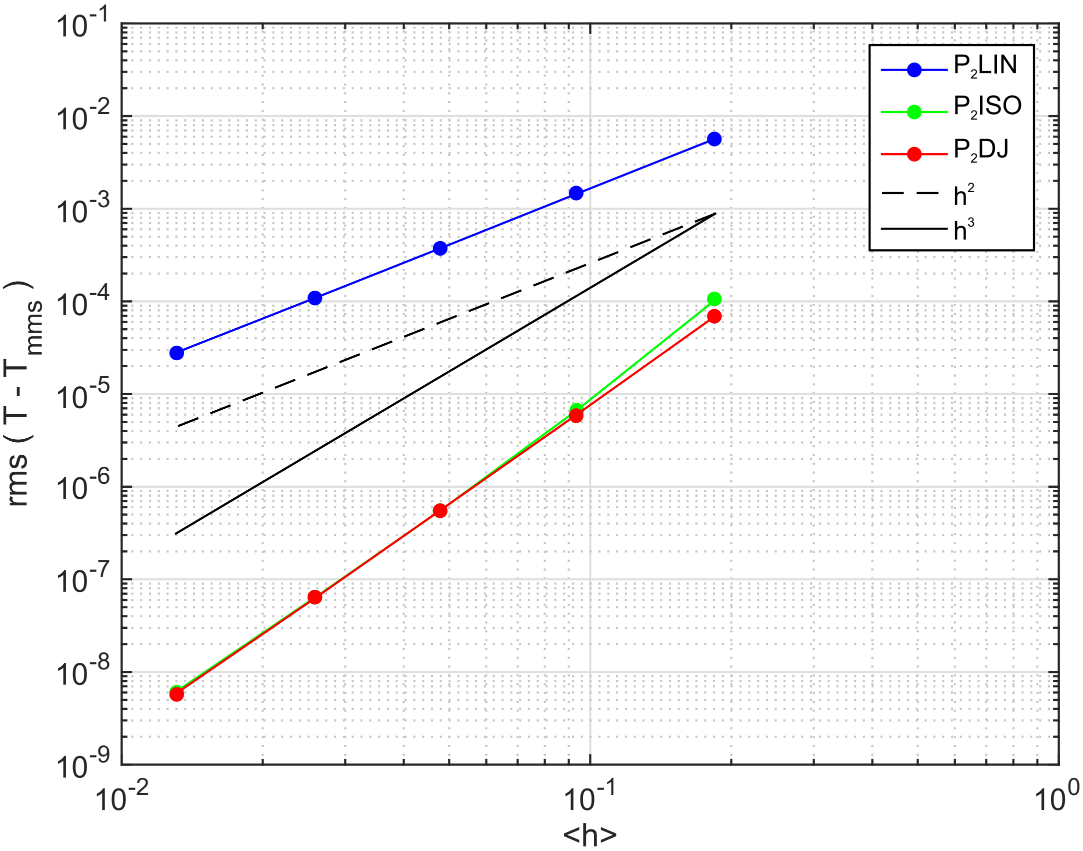


FIGURE 10 Root mean square error of temperature as a function of the average element size in the cylindrical annulus mesh after using the Method of Manufactured Solutions (MMS) to the steady state diffusion equation. Note that the P_2LIN element has a suboptimal h^2 convergence rate even though it uses the same second-order accurate interpolation functions as the P_2ISO element. The P_2DJ element performs best when large elements are used to define the circular boundaries of the problem region.

3 | 3-D DOUBLE JACOBIAN

For completeness, we also summarize the development of a 3-D spherical shell version of the Double Jacobian formulation. The one additional complication of the 3-D spherical formulation is the need to avoid integrating elements along the polar axis of the spherical coordinate system. We avoid this by rotating these elements to a equatorial location (and back) during their assembly, as we discuss below.

3.1 | First Jacobian

In a 3-D problem, derivatives are expressed in matrix form in terms of local ξ, η, ζ derivatives as:

$$\begin{pmatrix} \frac{dN_i}{dx} \\ \frac{dN_i}{dy} \\ \frac{dN_i}{dz} \end{pmatrix} = \underbrace{\begin{bmatrix} \frac{d\xi}{dx} & \frac{d\eta}{dx} & \frac{d\zeta}{dx} \\ \frac{d\xi}{dy} & \frac{d\eta}{dy} & \frac{d\zeta}{dy} \\ \frac{d\xi}{dz} & \frac{d\eta}{dz} & \frac{d\zeta}{dz} \end{bmatrix}}_{J_{LC}} \begin{pmatrix} \frac{dN_i}{d\xi} \\ \frac{dN_i}{d\eta} \\ \frac{dN_i}{d\zeta} \end{pmatrix}, \tag{36}$$

where N are the shape functions, i is the local node numbering of the element and J_{LC} is the Jacobian from local to Cartesian coordinates. The shape functions for a 4-node reference tetrahedron (Figure 11) are given by:

$$N_1 = \xi\,, \tag{37a}$$

$$N_2 = \eta\,, \tag{37b}$$

$$N_3 = \zeta\,, \tag{37c}$$

$$N_4 = 1 - \xi - \eta - \zeta\,. \tag{37d}$$

and their derivatives with respect to local coordinates can be explicitly computed.

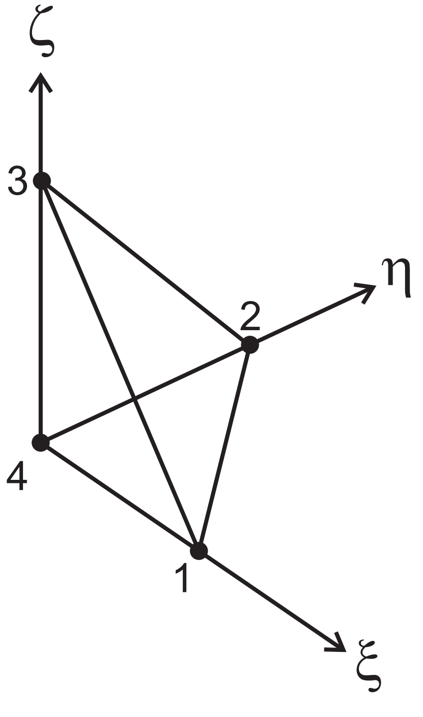


FIGURE 11 Nodal ordering for the reference tetrahedron in the local coordinate frame.

The first Jacobian for a spherical mapping is analogous to the standard Jacobian where x, y and z are replaced by θ, ϕ and r respectively, where θ is the colatitude, ϕ is the longitude, and r is the radius. Global derivatives expressed in matrix form are

$$\begin{pmatrix} \dfrac{dN_i}{d\theta} \\ \\ \dfrac{dN_i}{d\phi} \\ \\ \dfrac{dN_i}{dr} \end{pmatrix} = \underbrace{\begin{bmatrix} \dfrac{d\xi}{d\theta} & \dfrac{d\eta}{d\theta} & \dfrac{d\zeta}{d\theta} \\ \\ \dfrac{d\xi}{d\phi} & \dfrac{d\eta}{d\phi} & \dfrac{d\zeta}{d\phi} \\ \\ \dfrac{d\xi}{dr} & \dfrac{d\eta}{dr} & \dfrac{d\zeta}{dr} \end{bmatrix}}_{J_{LS}} \begin{pmatrix} \dfrac{dN_i}{d\xi} \\ \\ \dfrac{dN_i}{d\eta} \\ \\ \dfrac{dN_i}{d\zeta} \end{pmatrix}, \tag{38}$$

where J_{LS} is the Jacobian from local to spherical coordinates. As before, we do not have an analytical expression for $\xi(\theta, \phi, r)$, $\eta(\theta, \phi, r)$ and $\zeta(\theta, \phi, r)$. Since we do know the inverse relations, we use the inverse transformation of coordinates

$$\begin{pmatrix} \dfrac{dN_i}{d\xi} \\ \\ \dfrac{dN_i}{d\eta} \\ \\ \dfrac{dN_i}{d\zeta} \end{pmatrix} = \underbrace{\begin{bmatrix} \dfrac{d\theta}{d\xi} & \dfrac{d\phi}{d\xi} & \dfrac{dr}{d\xi} \\ \\ \dfrac{d\theta}{d\eta} & \dfrac{d\phi}{d\eta} & \dfrac{dr}{d\eta} \\ \\ \dfrac{d\theta}{d\zeta} & \dfrac{d\phi}{d\zeta} & \dfrac{dr}{d\zeta} \end{bmatrix}}_{J_{SL}} \begin{pmatrix} \dfrac{dN_i}{d\theta} \\ \\ \dfrac{dN_i}{d\phi} \\ \\ \dfrac{dN_i}{dr} \end{pmatrix}, \tag{39}$$

where J_{SL} is the Jacobian matrix for transforming from spherical to local coordinates. From equations (38) and (39) we see that $J_{LS} \equiv (J_{SL})^{-1}$. The inverse of the Jacobian from spherical to local coordinates is given by

$$(J_{SL})^{-1} = \frac{\begin{bmatrix} \phi,_\eta r,_\zeta - \phi,_\zeta r,_\eta & \phi,_\zeta r,_\xi - \phi,_\xi r,_\zeta & \phi,_\xi r,_\eta - \phi,_\eta r,_\xi \\ \\ \theta,_\zeta r,_\eta - \theta,_\eta r,_\zeta & \theta,_\xi r,_\zeta - \theta,_\zeta r,_\xi & \theta,_\eta r,_\xi - \theta,_\xi r,_\eta \\ \\ \theta,_\eta \phi,_\zeta - \theta,_\zeta \phi,_\eta & \theta,_\zeta \phi,_\xi - \theta,_\xi \phi,_\zeta & \theta,_\xi \phi,_\eta - \theta,_\eta \phi,_\xi \end{bmatrix}}{\theta,_\xi \phi,_\eta r,_\zeta + \theta,_\eta \phi,_\zeta r,_\xi + \theta,_\zeta \phi,_\xi r,_\eta - \theta,_\zeta \phi,_\eta r,_\xi - \theta,_\xi \phi,_\zeta r,_\eta - \theta,_\eta \varphi,_\xi r,_\zeta}, \tag{40}$$

where the compact notation $\theta,_\xi$ means the derivative of θ with respect to ξ. Spherical coordinates for each element are related to local coordinates through the shape functions:

$$\theta(\xi, \eta, \zeta) = \sum_{i=1}^{4} N_i(\xi, \eta, \zeta)\, \theta_i \,, \tag{41a}$$

$$\phi(\xi, \eta, \zeta) = \sum_{i=1}^{4} N_i(\xi, \eta, \zeta)\, \phi_i \,, \tag{41b}$$

$$r(\xi, \eta, \zeta) = \sum_{i=1}^{4} N_i(\xi, \eta, \zeta)\, r_i \,, \tag{41c}$$

Differentiating equations (41a) to (41c) yields:

$$\theta,_{\xi} = \theta_1 - \theta_4 \equiv \theta_{14}\,, \tag{42a}$$

$$\theta,_{\eta} = \theta_2 - \theta_4 \equiv \theta_{24}\,, \tag{42b}$$

$$\theta,_{\zeta} = \theta_3 - \theta_4 \equiv \theta_{34}\,, \tag{42c}$$

$$\phi,_{\xi} = \phi_1 - \phi_4 \equiv \phi_{14}\,, \tag{42d}$$

$$\phi,_{\eta} = \phi_2 - \phi_4 \equiv \phi_{24}\,, \tag{42e}$$

$$\phi,_{\zeta} = \phi_3 - \phi_4 \equiv \phi_{34}\,, \tag{42f}$$

$$r,_{\xi} = r_1 - r_4 \equiv r_{14}\,, \tag{42g}$$

$$r,_{\eta} = r_2 - r_4 \equiv r_{24}\,, \tag{42h}$$

$$r,_{\zeta} = r_3 - r_4 \equiv r_{34}\,. \tag{42i}$$

Finally, the inverse of the Jacobian from spherical to local coordinates is given by substituting equations (42a) to (42i) into equation (40):

$$(J_{SL})^{-1} = \frac{\begin{bmatrix} \phi_{24}r_{34} - \phi_{34}r_{24} & \phi_{34}r_{14} - \phi_{14}r_{34} & \phi_{14}r_{24} - \phi_{24}r_{14} \\ \theta_{34}r_{24} - \theta_{24}r_{34} & \theta_{14}r_{34} - \theta_{34}r_{14} & \theta_{24}r_{14} - \theta_{14}r_{24} \\ \theta_{24}\phi_{34} - \theta_{34}\phi_{24} & \theta_{34}\phi_{14} - \theta_{14}\phi_{34} & \theta_{14}\phi_{24} - \theta_{24}\phi_{14} \end{bmatrix}}{\theta_{14}(\phi_{24}r_{34} - \phi_{34}r_{24}) + \theta_{24}(\phi_{34}r_{14} - \phi_{14}r_{34}) + \theta_{34}(\phi_{14}r_{24} - \phi_{24}r_{14})}\,. \tag{43}$$

3.2 | Second Jacobian

The second Jacobian used in constructing the Double Jacobian is the analytical mapping from spherical coordinates to Cartesian coordinates. These derivatives expressed in matrix form are

$$\begin{pmatrix} \dfrac{dN_i}{dx} \\ \dfrac{dN_i}{dy} \\ \dfrac{dN_i}{dz} \end{pmatrix} = \underbrace{\begin{bmatrix} \dfrac{d\theta}{dx} & \dfrac{d\phi}{dx} & \dfrac{dr}{dx} \\ \dfrac{d\theta}{dy} & \dfrac{d\phi}{dy} & \dfrac{dr}{dy} \\ \dfrac{d\theta}{dz} & \dfrac{d\phi}{dz} & \dfrac{dr}{dz} \end{bmatrix}}_{J_{SC}} \begin{pmatrix} \dfrac{dN_i}{d\theta} \\ \dfrac{dN_i}{d\phi} \\ \dfrac{dN_i}{dr} \end{pmatrix}, \tag{44}$$

where J_{SC} is the Jacobian from spherical to Cartesian coordinates. Again, we make use of the inverse transformation between these coordinate systems

$$\begin{pmatrix} \dfrac{dN_i}{d\theta} \\ \dfrac{dN_i}{d\phi} \\ \dfrac{dN_i}{dr} \end{pmatrix} = \underbrace{\begin{bmatrix} \dfrac{dx}{d\theta} & \dfrac{dy}{d\theta} & \dfrac{dz}{d\theta} \\ \dfrac{dx}{d\phi} & \dfrac{dy}{d\phi} & \dfrac{dz}{d\phi} \\ \dfrac{dx}{dr} & \dfrac{dy}{dr} & \dfrac{dz}{dr} \end{bmatrix}}_{J_{CS}} \begin{pmatrix} \dfrac{dN_i}{dx} \\ \dfrac{dN_i}{dy} \\ \dfrac{dN_i}{dz} \end{pmatrix}, \tag{45}$$

where J_{CS} is the Jacobian matrix to transform from Cartesian to spherical coordinates, and $J_{SC} \equiv (J_{CS})^{-1}$. The inverse of the Jacobian from Cartesian to spherical coordinates is given by

$$(J_{CS})^{-1} = \frac{\begin{bmatrix} y,_{\phi}\, z,_{r} - y,_{r}\, z,_{\phi} & y,_{r}\, z,_{\theta} - y,_{\theta}\, z,_{r} & y,_{\theta}\, z,_{\phi} - y,_{\phi}\, z,_{\theta} \\ x,_{r}\, z,_{\phi} - x,_{\phi}\, z,_{r} & x,_{\theta}\, z,_{r} - x,_{r}\, z,_{\theta} & x,_{\phi}\, z,_{\theta} - x,_{\theta}\, z,_{\phi} \\ x,_{\phi}\, y,_{r} - x,_{r}\, y,_{\phi} & x,_{r}\, y,_{\theta} - x,_{\theta}\, y,_{r} & x,_{\theta}\, y,_{\phi} - x,_{\phi}\, y,_{\theta} \end{bmatrix}}{x,_{\theta}\,(y,_{\phi}\, z,_{r} - y,_{r}\, z,_{\phi}) + x,_{\phi}\,(y,_{r}\, z,_{\theta} - y,_{\theta}\, z,_{r}) + x,_{r}\,(y,_{\theta}\, z,_{\phi} - y,_{\phi}\, z,_{\theta})}\,, \tag{46}$$

where again $x,_{\theta}$ means the derivative of x with respect to θ. Cartesian coordinates within each element are related to spherical coordinates through:

$$x(\theta, \phi, r) = r \sin\theta \cos\phi\,, \tag{47a}$$

$$y(\theta, \phi, r) = r \sin\theta \sin\phi\,, \tag{47b}$$

$$z(\theta, \phi, r) = r \cos\theta\,. \tag{47c}$$

Differentiating equations (47a) to (47c) yields:

$$x,_{\theta} = r \cos\theta \cos\phi\,, \tag{48a}$$

$$x,_{\phi} = -r \sin\theta \sin\phi\,, \tag{48b}$$

$$x,_{r} = \sin\theta \cos\phi\,, \tag{48c}$$

$$y,_{\theta} = r \cos\theta \sin\phi\,, \tag{48d}$$

$$y,_{\phi} = r \sin\theta \cos\phi\,, \tag{48e}$$

$$y,_{r} = \sin\theta \sin\phi\,, \tag{48f}$$

$$z,_{\theta} = -r \sin\theta\,, \tag{48g}$$

$$z,_{\phi} = 0\,, \tag{48h}$$

$$z,_{r} = \cos\theta\,. \tag{48i}$$

The inverse of the Jacobian from Cartesian coordinates to spherical coordinates is obatined by substituting equations (48a) to (48i) into equation (46):

$$(J_{CS})^{-1} = \begin{bmatrix} \dfrac{\cos\theta\cos\phi}{r} & -\dfrac{\sin\phi}{r\sin\theta} & \sin\theta\cos\phi \\ \dfrac{\cos\theta\sin\phi}{r} & \dfrac{\cos\phi}{r\sin\theta} & \sin\theta\sin\phi \\ -\dfrac{\sin\theta}{r} & 0 & \cos\theta \end{bmatrix}, \tag{49}$$

where θ, ϕ and r are evaluated at each integration point within the spherical tetrahedron. Making use of the two Jacobians, global derivatives can be expressed as a function of local derivatives through a spherical coordinate mapping. In this way, the surfaces and edges of Cartesian elements are appropriately curved. Substituting equation (38) into equation (44) yields

$$\begin{pmatrix} \dfrac{dN_i}{dx} \\ \dfrac{dN_i}{dy} \\ \dfrac{dN_i}{dz} \end{pmatrix} = \underbrace{\begin{bmatrix} \dfrac{d\theta}{dx} & \dfrac{d\phi}{dx} & \dfrac{dr}{dx} \\ \dfrac{d\theta}{dy} & \dfrac{d\phi}{dy} & \dfrac{dr}{dy} \\ \dfrac{d\theta}{dz} & \dfrac{d\phi}{dz} & \dfrac{dr}{dz} \end{bmatrix}}_{J_{SC}} \underbrace{\begin{bmatrix} \dfrac{d\xi}{d\theta} & \dfrac{d\eta}{d\theta} & \dfrac{d\zeta}{d\theta} \\ \dfrac{d\xi}{d\phi} & \dfrac{d\eta}{d\phi} & \dfrac{d\zeta}{d\phi} \\ \dfrac{d\xi}{dr} & \dfrac{d\eta}{dr} & \dfrac{d\zeta}{dr} \end{bmatrix}}_{J_{LS}} \begin{pmatrix} \dfrac{dN_i}{d\xi} \\ \dfrac{dN_i}{d\eta} \\ \dfrac{dN_i}{d\zeta} \end{pmatrix}, \tag{50}$$

which is equivalent to

$$\begin{pmatrix} \dfrac{dN_i}{dx} \\[2ex] \dfrac{dN_i}{dy} \\[2ex] \dfrac{dN_i}{dz} \end{pmatrix} = (J_{CS})^{-1}(J_{SL})^{-1} \begin{pmatrix} \dfrac{dN_i}{d\xi} \\[2ex] \dfrac{dN_i}{d\eta} \\[2ex] \dfrac{dN_i}{d\zeta} \end{pmatrix} . \tag{51}$$

Substituting equations (43) and (49) into equation (51) yields

$$\begin{aligned} \begin{pmatrix} \dfrac{dN_i}{dx} \\[2ex] \dfrac{dN_i}{dy} \\[2ex] \dfrac{dN_i}{dz} \end{pmatrix} &= \frac{1}{\theta_{14}(\phi_{24}r_{34} - \phi_{34}r_{24}) + \theta_{24}(\phi_{34}r_{14} - \phi_{14}r_{34}) + \theta_{34}(\phi_{14}r_{24} - \phi_{24}r_{14})} \\ &\begin{bmatrix} \dfrac{\cos\theta\cos\phi}{r} & -\dfrac{\sin\phi}{r\sin\theta} & \sin\theta\cos\phi \\[2ex] \dfrac{\cos\theta\sin\phi}{r} & \dfrac{\cos\phi}{r\sin\theta} & \sin\theta\sin\phi \\[2ex] -\dfrac{\sin\theta}{r} & 0 & \cos\theta \end{bmatrix} \\ &\begin{bmatrix} \phi_{24}r_{34} - \phi_{34}r_{24} & \phi_{34}r_{14} - \phi_{14}r_{34} & \phi_{14}r_{24} - \phi_{24}r_{14} \\[2ex] \theta_{34}r_{24} - \theta_{24}r_{34} & \theta_{14}r_{34} - \theta_{34}r_{14} & \theta_{24}r_{14} - \theta_{14}r_{24} \\[2ex] \theta_{24}\phi_{34} - \theta_{34}\phi_{24} & \theta_{34}\phi_{14} - \theta_{14}\phi_{34} & \theta_{14}\phi_{24} - \theta_{24}\phi_{14} \end{bmatrix} \begin{pmatrix} \dfrac{dN_i}{d\xi} \\[2ex] \dfrac{dN_i}{d\eta} \\[2ex] \dfrac{dN_i}{d\zeta} \end{pmatrix} . \end{aligned} \tag{52}$$

The 3-D Double Jacobian (DJ) approach ensures an exact mapping to the spherical cap surfaces of the boundary elements in a spherical shell mesh. In the mapping from local coordinates to spherical coordinates (first Jacobian of the Double Jacobian), the element geometries are expressed in spherical coordinates in which their surfaces are the surfaces of tetrahedral 3-simplexes. Any point on the surface of a boundary element of the mesh is mapped to the true spherical shape. This contrasts with direct isoparametric or superparametric mapping from local to Cartesian coordinates. Neither the linear Jacobian (LIN) nor the isoparametric Jacobian (ISO) nor even the superparametric Jacobian (SUP) can exactly map to a spherical shape since these are polynomial approximations. Figure 12 shows the distance in km between the shape of tetrahedral faces computed using 4-node linear spherical mapping (DJ) and the face's shape computed using LIN, ISO, SUP Cartesian mappings. Note that the scale in the colour bar is different for each case. The element shown in Figure 12 has an average edge length of 2000 km. The shape of the tetrahedron faces given by ISO differ from DJ by a few kilometres. However, when solving for viscous flow within a spherical mesh containing coarse regions, results can be affected by these small boundary mismatches as will be shown below.

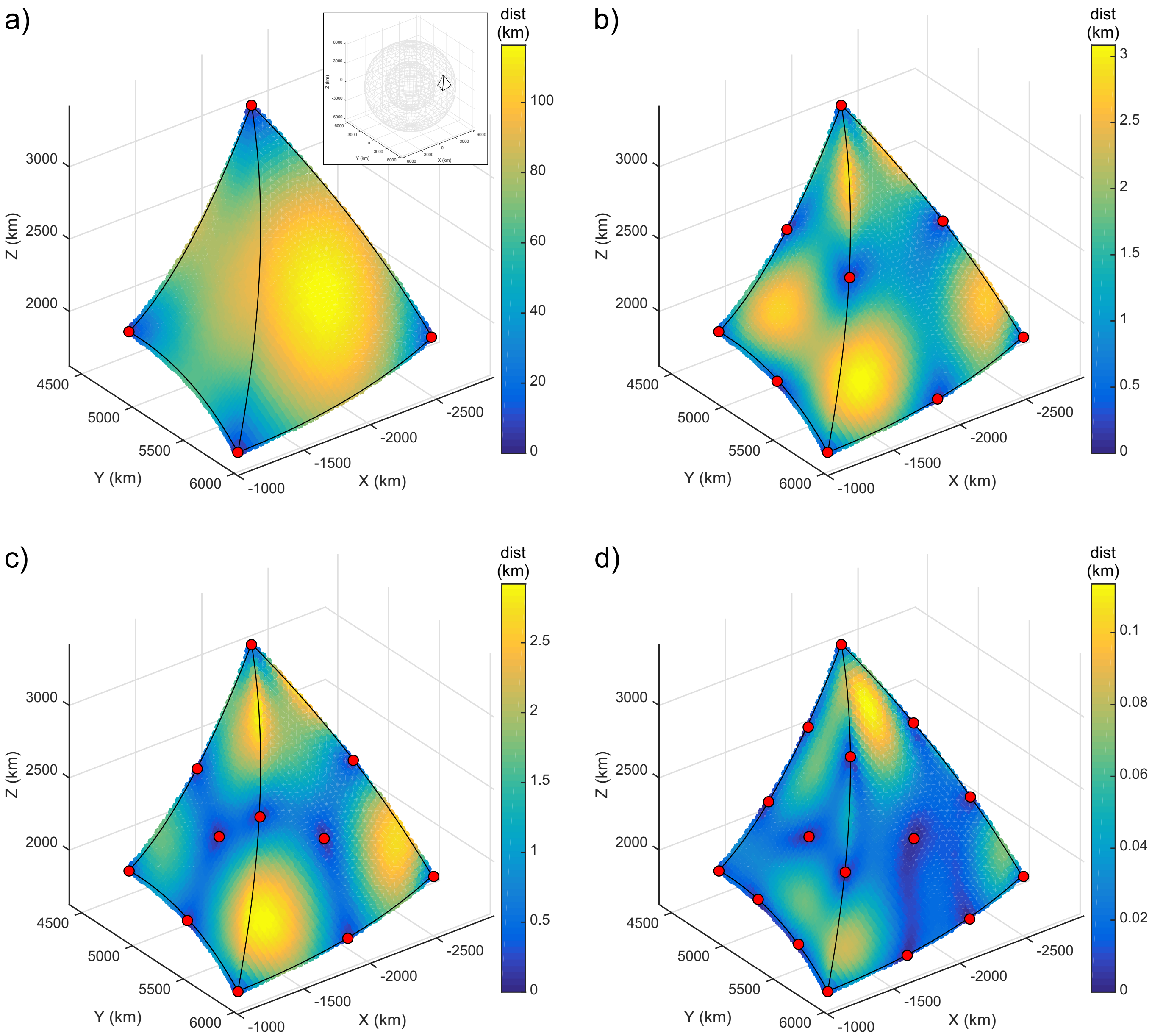


FIGURE 12 Distance in km (colours) between the shape of tetrahedral faces computed by 4-node linear spherical mapping (DJ), which preserves the true surface shape in spherical coordinates and the face's shape for the same tetrahedron computed by: (a) 4-node linear Cartesian mapping (LIN); (b) 10-node quadratic isoparametric Cartesian mapping (ISO); (c) 14-node quasi-cubic superparametric Cartesian mapping (Q-SUP); (d) 20-node cubic superparametric Cartesian mapping (SUP). The position of the tetrahedron in the mesh is shown in the top right inset in (a). Black lines represent the curved edges of the tetrahedron and red points represent the mapping element's nodal position. Note that the scale in the colour bar is different for each case.

3.3 | 3-D Standard Jacobian vs 3-D Double Jacobian

The 3-D spherical Double Jacobian formulation was first examined by comparing the stiffness matrices given by both Standard Jacobian (LIN and ISO) and Double Jacobian (DJ) formulations in a similar way to the above 2-D test. This comparison was done for the boundary elements of spherical shell meshes with different resolutions, from $l_0 = 2000$ km to $l_0 = 50$ km. We selected boundary elements along the plane YZ (inset of Figure 13b).

The domain of the Cartesian coordinate system is $\mathbb{R}^3$ whereas the domain of the spherical coordinate system is $\theta \in [0, \pi]$, $\phi \in [0, 2\pi)$ and $r \in [0, \infty)$. As in the 2-D case, the elements crossing the meridian at $\phi = 2\pi$ have a discontinuity in the ϕ coordinate of their vertices. Again as in 2-D, the 2π-crossing issue is resolved by doing a 180° rotation around the Z axis of the elements crossing the meridian $\phi = 2\pi$ and computing the Jacobian and stiffness matrices in this rotated frame. When assembling the stiffness matrices, transformations need to be done to the stiffness matrices of elements crossing $\phi = 2\pi$ in order to recover the correct matrices in the original frame. The matrix describing a 180° rotation around the Z axis is given by

$$R_{Z_{180}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{53}$$

We use continuous Taylor-Hood tetrahedral elements with a quadratic 10-node velocity interpolation (3 dof for each of the 10 nodes) and linear pressure interpolation (1 dof for each vertex node)(P_2P_1). The size of the matrices $\boldsymbol{K}_e$, $\boldsymbol{G}_e$ and vector $\boldsymbol{f}_e$ for each tetrahedral element are 30x30, 30x4 and 30x1 respectively. In terms of applying a rotation, the $\boldsymbol{G}_e$ matrix is handled as four column vectors. In order to recover the matrices $\boldsymbol{K}_e$, $\boldsymbol{G}_e$ and vector $\boldsymbol{f}_e$ from the rotated counterparts we apply the rotation matrix to the first, second and third dof of all nodes:

$$\boldsymbol{K}_e = \boldsymbol{R}^{\mathsf{T}}_{Z_{180}} \boldsymbol{K}^{rot}_e \boldsymbol{R}_{Z_{180}}\,, \tag{54a}$$

$$\boldsymbol{G}_e = \boldsymbol{R}^{\mathsf{T}}_{Z_{180}} \boldsymbol{G}^{rot}_e\,, \tag{54b}$$

$$\boldsymbol{f}_e = \boldsymbol{R}^{\mathsf{T}}_{Z_{180}} \boldsymbol{f}^{rot}_e\,. \tag{54c}$$

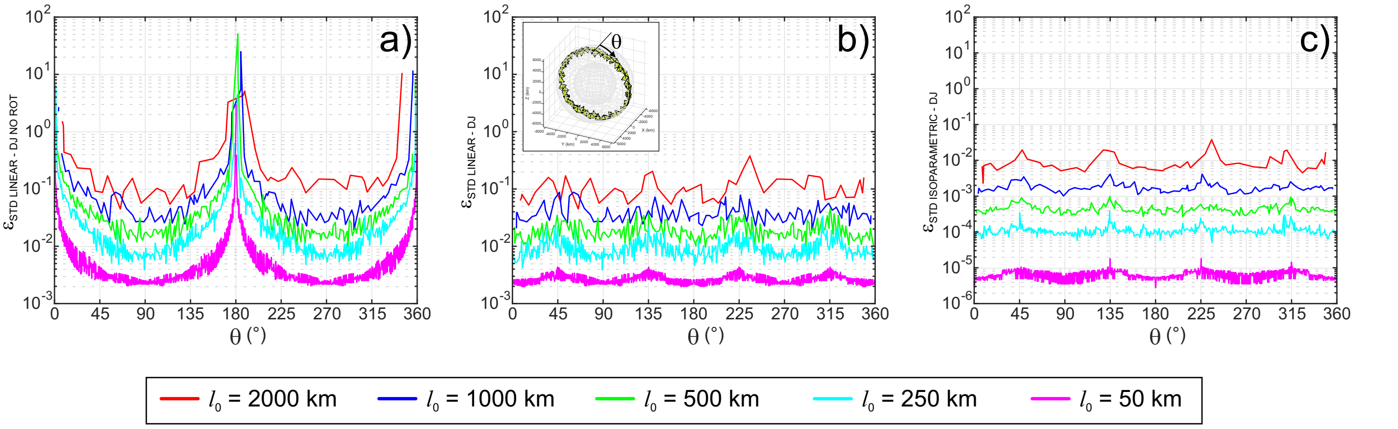


FIGURE 13 (a) The maximum relative difference between $\boldsymbol{K}_e$ computed with P_2P_1LIN elements and $\boldsymbol{K}_e$ computed with P_2P_1DJ elements without rotations. The data is computed for different mesh resolutions (colours in legend) as a function of the colatitude θ of the barycentres of the boundary elements. The inset in (b) shows an example ring of boundary elements with $l_0 = 1000$ km. (b) The maximum relative difference between $\boldsymbol{K}_e$ computed with P_2P_1LIN elements and $\boldsymbol{K}_e$ computed with P_2P_1DJ elements with rotations. (c) The maximum relative difference between $\boldsymbol{K}_e$ computed with P_2P_1ISO elements and $\boldsymbol{K}_e$ computed with P_2P_1DJ elements with rotations.

For simplicity we show the unrotated matrices and vector for one node with three dof. The rotated stiffness matrix, gradient matrix and force vector for one velocity node of one element are:

$$K_e^{rot}(node\ 1) = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}, \tag{55a}$$

$$G_e^{rot}(node\ 1) = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \end{bmatrix}, \tag{55b}$$

$$f_e^{rot}(node\ 1) = \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \end{pmatrix}. \tag{55c}$$

Substituting equations (55a) to (55c) into equations (54a) to (54c) respectively gives:

$$K_e(node\ 1) = \begin{bmatrix} k_{11} & k_{12} & -k_{13} \\ k_{21} & k_{22} & -k_{23} \\ -k_{31} & -k_{32} & k_{33} \end{bmatrix}, \tag{56a}$$

$$G_e(node\ 1) = \begin{bmatrix} -g_{11} & -g_{12} & -g_{13} & -g_{14} \\ -g_{21} & -g_{22} & -g_{23} & -g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \end{bmatrix}, \tag{56b}$$

$$f_e(node\ 1) = \begin{pmatrix} -f_{11} \\ -f_{21} \\ f_{31} \end{pmatrix}. \tag{56c}$$

Elements near the axis of the spherical coordinate system have an additional resolution issue. Their longitudinal coordinate will vary considerably within an element leading to less accurate numerical integration of these stiffness matrices using the Double Jacobian approach. To highlight the problem of numerical integration near the pole of the spherical geometry we calculated the difference between the stiffness matrix computed with P_2P_1LIN and P_2P_1DJ elements using equation (26) for a ring of boundary elements shown in the inset of Figure 13b. The large differences near the poles reflect the poor numerical integration of P_2P_1DJ elements in this region (Figure 13a). We choose to resolve this issue by doing a 90° counter-clockwise rotation around the X axis for all elements with an angular distance less than θ_c from the pole, where θ_c is the angle between the Z axis and the cone's surface (Figure 14 shows an example). This 90° rotation places them near the equator of this new spherical reference frame and their Jacobian, element matrices and vectors are computed in this rotated frame. After some experimentation, we choose $\theta_c = 45°$. Alternatively, one could try the approach of increasing the accuracy of the numerical integration scheme for elements near the pole of the spherical coordinate system. When assembling the matrices of the elements inside the cone, two transformations are needed to construct the corresponding matrices in the unrotated frame. The matrix for a 90° counter-clockwise rotation around the X axis is given by

$$R_{X_{90}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{57}$$

In order to recover the matrices K_e, G_e and vector f_e from the rotated counterparts we apply the rotation matrix to each group of three dof for each node:

$$K_e = R_{X_{90}}^{\mathsf{T}} K_e^{rot} R_{X_{90}}, \tag{58a}$$

$$G_e = R_{X_{90}}^{\mathsf{T}} G_e^{rot}, \tag{58b}$$

$$f_e = R_{X_{90}}^{\mathsf{T}} f_e^{rot}. \tag{58c}$$

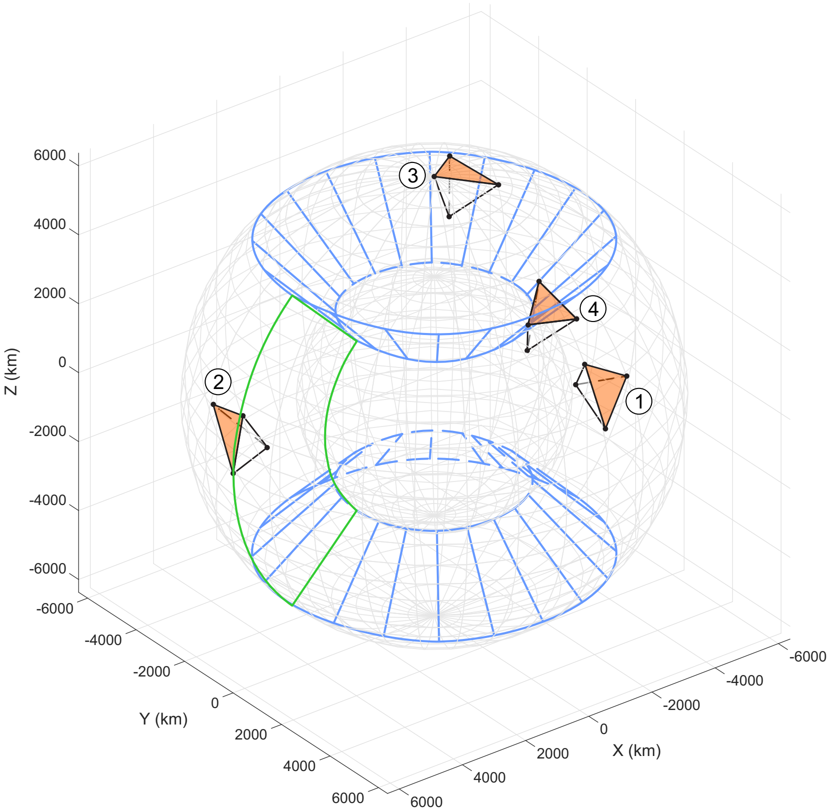


FIGURE 14 Classification of the spherical tetrahedral elements of a spherical shell mesh: (1) elements outside the cone that do not cross the meridian $\phi = 2\pi$ (Type 1), (2) elements outside the cone that cross the meridian $\phi = 2\pi$ (Type 2), (3) elements inside the cone (Type 3), (4) elements crossing the cone boundary (Type 4). Black points are the vertices of the element, grey lines represent the ideal spherical boundaries, orange colour represents the element face along the outer mesh boundary, green lines define the meridian $\phi = 2\pi$ and blue lines represent the double cone with $\theta_c = 45°$.

Substituting equations (55a) to (55c) into equations (58a) to (58c) respectively gives:

$$K_e(node\ 1) = \begin{bmatrix} k_{11} & k_{13} & -k_{12} \\ k_{31} & k_{33} & -k_{32} \\ -k_{21} & -k_{23} & k_{22} \end{bmatrix}, \tag{59a}$$

$$G_e(node\ 1) = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ -g_{21} & -g_{22} & -g_{23} & -g_{24} \end{bmatrix}, \tag{59b}$$

$$f_e(node\ 1) = \begin{pmatrix} f_{11} \\ f_{31} \\ -f_{21} \end{pmatrix}. \tag{59c}$$

The difference between the stiffness matrices for the boundary elements computed with P_2P_1LIN and P_2P_1DJ elements using these rotations is shown in Figure 13b. The effect of proximity to the pole does not appear when using the above rotations to calculate the element matrices. For simplicity, from now on we will refer to the Double Jacobian method including the rotations as 'the Double Jacobian method' (DJ). The two approaches yield more similar stiffness matrices for smaller element sizes. For example, P_2P_1LIN elements in a mesh with $l_0 = 250$ km give stiffness matrices which differ from the ones given by P_2P_1DJ elements by about 1% (cyan line in Figure 13b). However, in the case of a mesh with $l_0 = 2000$ km, the stiffness matrices given by both Jacobians differ by 10% on average (red line in Figure 13b). As in the cylindrical case, P_2P_1LIN elements provide a poor approximation to the boundary curvature for elements in coarse meshes, and its resulting stiffness matrices are imprecise.

Even using the Cartesian isoparametric P_2P_1ISO element, noticeable differences remain when compared to the Double Jacobian (Figure 13c).

3-D quadratic elements with straight edges

The general quadratic tetrahedral element in a Cartesian coordinate system has nodes added to the middle of each edge. Cartesian coordinates of the edge nodes for an element are given by:

$$\boldsymbol{m}_5 = \left(\frac{1}{2}(x_1+x_2), \frac{1}{2}(y_1+y_2), \frac{1}{2}(z_1+z_2)\right) , \tag{60a}$$

$$\boldsymbol{m}_6 = \left(\frac{1}{2}(x_2+x_3), \frac{1}{2}(y_2+y_3), \frac{1}{2}(z_2+z_3)\right) , \tag{60b}$$

$$\boldsymbol{m}_7 = \left(\frac{1}{2}(x_3+x_4), \frac{1}{2}(y_3+y_4), \frac{1}{2}(z_3+z_4)\right) , \tag{60c}$$

$$\boldsymbol{m}_8 = \left(\frac{1}{2}(x_4+x_1), \frac{1}{2}(y_4+y_1), \frac{1}{2}(z_4+z_1)\right) , \tag{60d}$$

$$\boldsymbol{m}_9 = \left(\frac{1}{2}(x_1+x_3), \frac{1}{2}(y_1+y_3), \frac{1}{2}(z_1+z_3)\right) , \tag{60e}$$

$$\boldsymbol{m}_{10} = \left(\frac{1}{2}(x_4+x_2), \frac{1}{2}(y_4+y_2), \frac{1}{2}(z_4+z_2)\right) , \tag{60f}$$

where subscripts are the local numbering of the element, being 1 to 4 for the vertices and 5 to 10 for the mid-edge nodes. Calculating the edge node positions using equations (60a) to (60f) leads to quadratic tetrahedral elements with straight edges in the Cartesian coordinate system. In isoparametric elements, the mid-edge node positions are shifted to better represent curved boundaries.

3-D quadratic elements computed in spherical coordinates

In spherical geometries it is useful to perform a recursive mesh refinement by splitting the Taylor-Hood elements in a spherical coordinate system in which mid-edge nodes of boundary edges lie along a spherical boundary. The larger the element size, the larger the improvement given by the use of curved edges. Quadratic elements with curved edges in Cartesian coordinates are computed in spherical coordinates using the Euclidean distance in a similar way to quadratic elements with straight edges. Note that linear edges in spherical coordinates will be curved edges in Cartesian coordinates, and vice versa. The process involves a coordinate transformation. First, the Cartesian coordinates of the vertices of each element are transformed to spherical coordinates. Second, mid-edge nodes in spherical coordinates for each element are computed through:

$$\boldsymbol{\mu}_5 = \left(\frac{1}{2}(\theta_1+\theta_2), \frac{1}{2}(\phi_1+\phi_2), \frac{1}{2}(r_1+r_2)\right) , \tag{61a}$$

$$\boldsymbol{\mu}_6 = \left(\frac{1}{2}(\theta_2+\theta_3), \frac{1}{2}(\phi_2+\phi_3), \frac{1}{2}(r_2+r_3)\right) , \tag{61b}$$

$$\boldsymbol{\mu}_7 = \left(\frac{1}{2}(\theta_3+\theta_4), \frac{1}{2}(\phi_3+\phi_4), \frac{1}{2}(r_3+r_4)\right) , \tag{61c}$$

$$\boldsymbol{\mu}_8 = \left(\frac{1}{2}(\theta_4+\theta_1), \frac{1}{2}(\phi_4+\phi_1), \frac{1}{2}(r_4+r_1)\right) , \tag{61d}$$

$$\boldsymbol{\mu}_9 = \left(\frac{1}{2}(\theta_1+\theta_3), \frac{1}{2}(\phi_1+\phi_3), \frac{1}{2}(r_1+r_3)\right) , \tag{61e}$$

$$\boldsymbol{\mu}_{10} = \left(\frac{1}{2}(\theta_4+\theta_2), \frac{1}{2}(\phi_4+\phi_2), \frac{1}{2}(r_4+r_2)\right) . \tag{61f}$$

Third, the spherical vertex and mid-edge coordinates are transformed back to Cartesian coordinates. Alternatively, boundary mid-edge nodes can be projected in the radial direction to the boundary. Again there is a special case for elements crossing the meridian $\phi = 2\pi$. While their vertices are nearby in Cartesian coordinates, some are nearly 2π radians from each other in spherical coordinates. Elements close to the spherical pole can also become very distorted when their mid-edge nodes are computed in spherical coordinates. This is mainly due to two facts. First, the closer the element to the Z axis the wider the longitude range of the vertices within the same element, leading to wrong mid-edge positions. Second, elements may have one or two vertices on the Z axis itself. Any node lying on the Z axis will not have a well defined longitude.

The problem of computing mid-edge nodes in spherical coordinates for elements crossing the meridian $\phi = 2\pi$ and elements lying closer than an angle θ_c from the pole is addressed through the identification of four types of elements as a function of their position within the mesh (Figure 14):

- Type 1 (tetrahedron 1 in Figure 14): Elements whose four vertices are outside of the double cone and do not cross the meridian $\phi = 2\pi$. These tetrahedral elements have linear edges and faces in the original spherical frame.
- Type 2 (tetrahedron 2 in Figure 14): Elements whose four vertices are outside of the double cone yet cross the meridian $\phi = 2\pi$. These tetrahedral elements have linear edges and faces in the 180°-rotated-around-the-Z-axis spherical frame.
- Type 3 (tetrahedron 3 in Figure 14): Elements having at least three vertices inside of the double cone. These tetrahedral elements have linear edges in the 90°-rotated-around-the-X-axis spherical frame.
- Type 4 (element 4 in Figure 14): Elements that cross the double cone boundary and have at least one edge outside the double cone. These elements act as a link between the spherical tetrahedral elements computed inside the cone, i.e. in the 90°-rotated-around-the-X-axis spherical frame, and the spherical tetrahedral elements computed outside the cone, i.e. in the original spherical frame. These elements have some curved edges and faces in the 90°-rotated-around-the-X-axis spherical frame.

Type 1 elements

The mid-edge nodes of Type 1 elements are computed as follows:

- Transform to spherical coordinates.
- Compute mid-edge nodes following equations (61a) to (61f) where edges are straight.
- Transform back to Cartesian coordinates.

Type 2 elements

Computing mid-edge nodes for Type 2 elements involves the following steps:

- Make a 180° rotation around the Z axis (Figure 15b).
- Transform to spherical coordinates and compute mid-edge nodes following equations (61a) to (61f) where edges are straight (Figure 15c).
- Transform back to Cartesian coordinates (Figure 15d).
- Make a 180° rotation around the Z axis (Figure 15e).

The mid-edge nodes of elements crossing the meridian $\phi = 2\pi$ computed in the 180° rotated spherical system match exactly with the mid-edge nodes of the shared edges with the neighbour elements computed in the original spherical system due to their symmetry with respect to the Z axis.

Type 3 elements

The mid-edge nodes of Type 3 elements are computed as follows:

- Make a 90° counter-clockwise rotation around X axis (Figure 16b).
- Transform to spherical coordinates and compute mid-edge nodes following equations (61a) to (61f) where edges are straight (Figure 16c).
- Transform back to Cartesian coordinates (Figure 16d).
- Make a 90° clockwise rotation around X axis (Figure 16e).

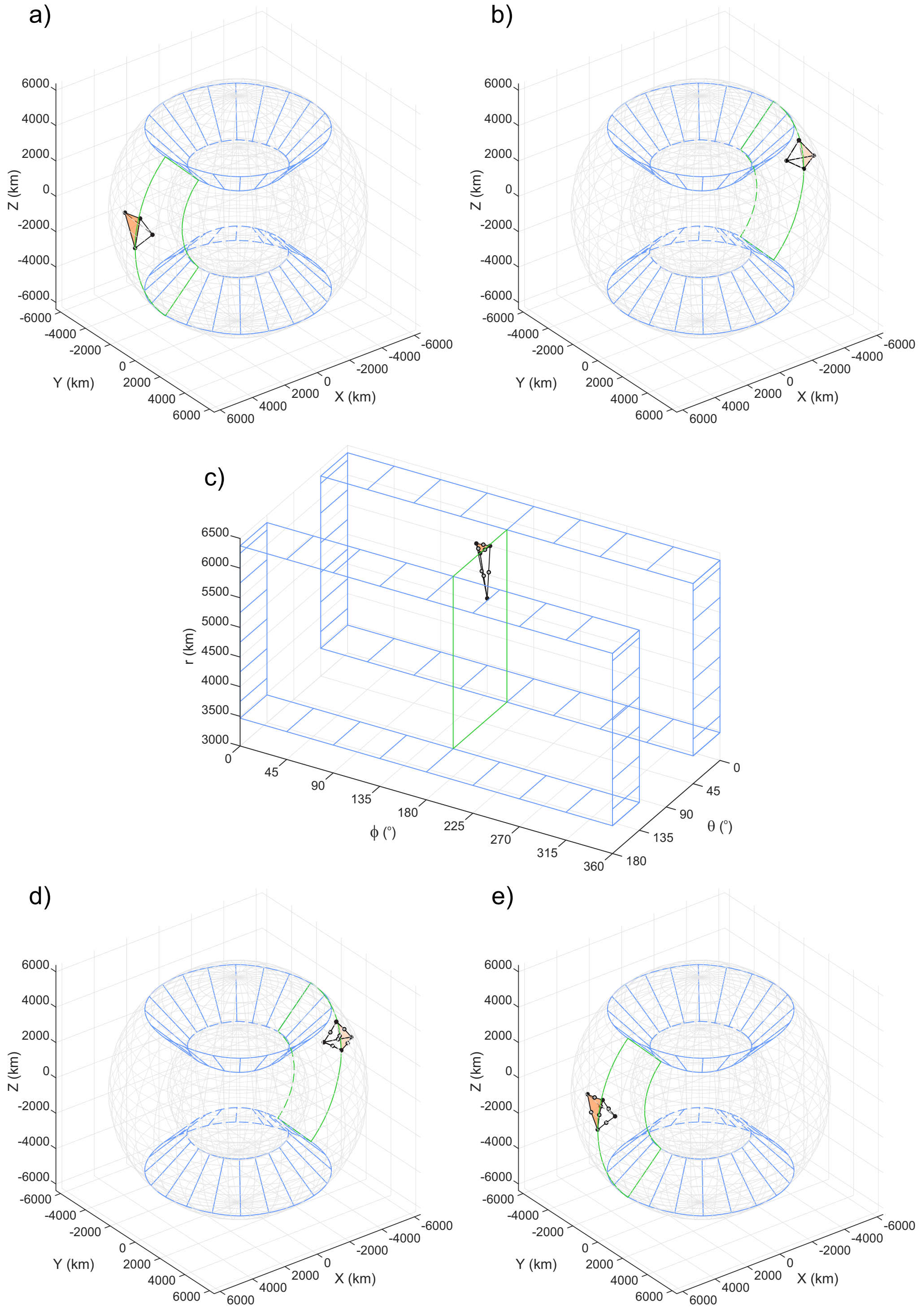


FIGURE 15 (a) Linear element crossing the meridian $\phi = 2\pi$ of a spherical shell mesh in Cartesian coordinate system. (b) Same element in a 180° rotated Cartesian coordinate system. (c) The same element in a 180° rotated spherical system in which it has straight edges. Mid-edge nodes (grey points) are computed following equations (61a) to (61f) to obtain a quadratic-order linear-edged element. (d) Rotated quadratic element in Cartesian coordinates, where the mid-edge nodes of the boundary edges lie along the boundary. (e) Quadratic element with curved edges in the original Cartesian coordinate system after undoing the 180° rotation. Black points are the vertices of the element, grey lines represent the ideal spherical boundaries, orange colour represents the element face which is on the outer mesh boundary, green lines define the meridian $\phi = 2\,\pi$ and blue lines represent the double cone with $\theta_c = 45°$.

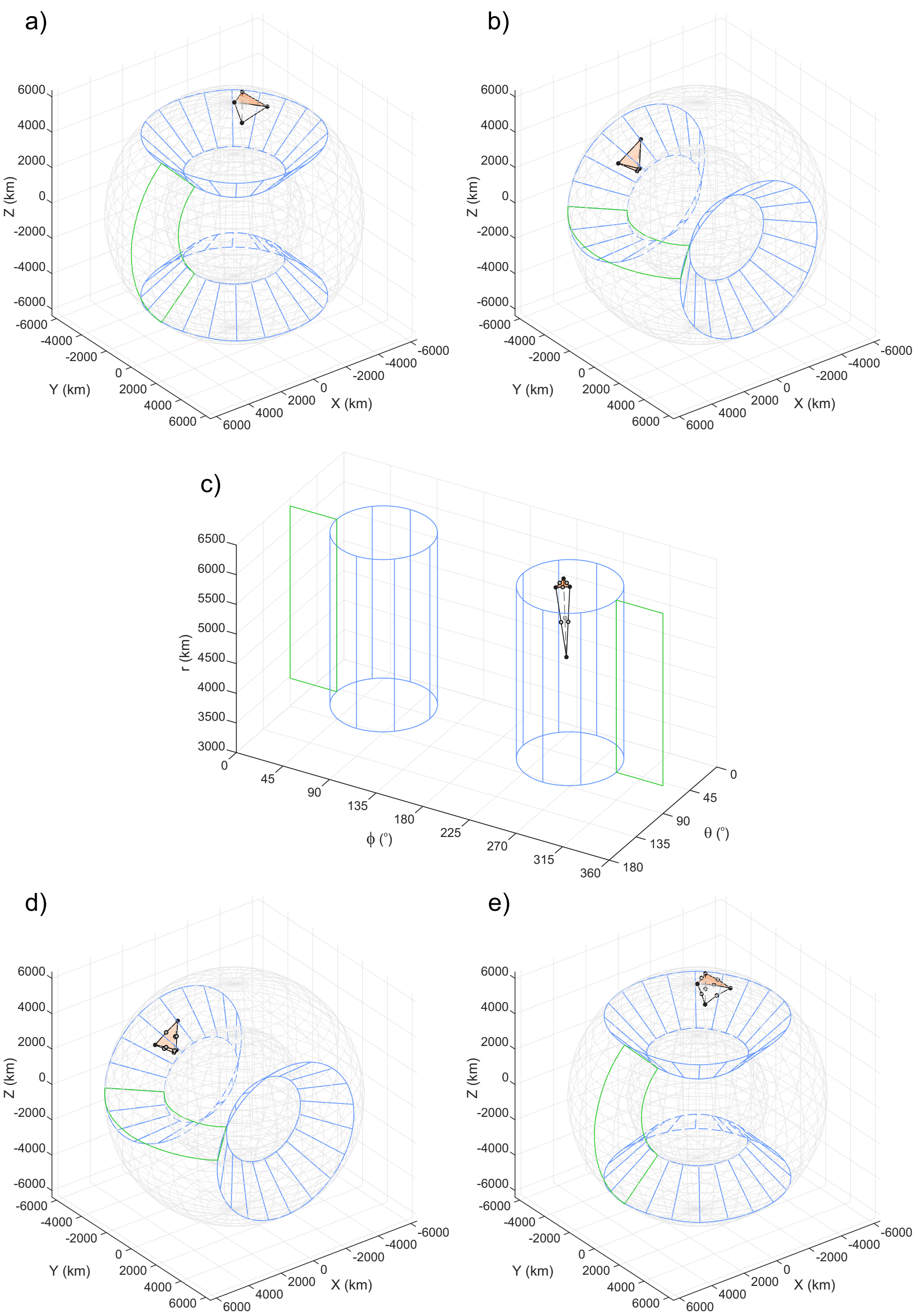


FIGURE 16 (a) Linear element inside the cone of a spherical shell mesh in Cartesian coordinate system. (b) Same element in a 90° counter-clockwise rotated Cartesian coordinate system. (c) The same element in a 90° rotated spherical system in which it has straight edges. Mid-edge nodes (grey points) are computed following equations (61a) to (61f) to obtain a quadratic-order element. (d) Rotated quadratic element in Cartesian coordinates, where the mid-edge nodes of the boundary edges lie along the boundary. (e) Quadratic element with curved edges in the original Cartesian coordinate system. Black points are the vertices of the element, grey lines represent the ideal spherical boundaries, orange colour represents the element face which is on the outer mesh boundary, green lines define the meridian $\phi = 2\pi$ and blue lines represent the double cone with $\theta_c = 45°$.

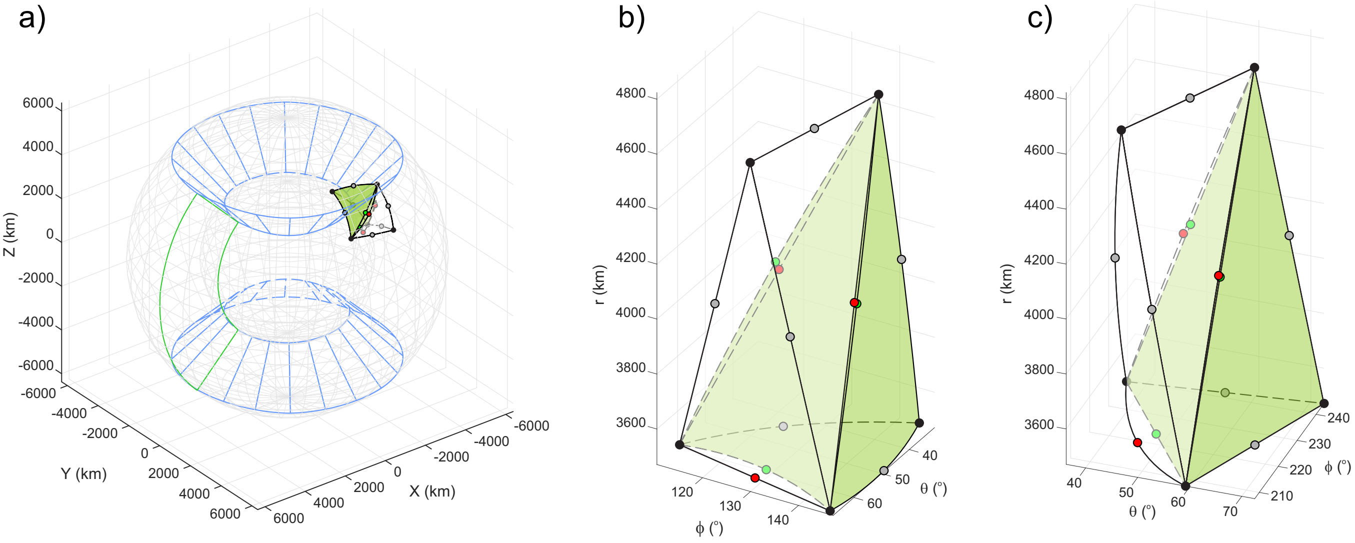


FIGURE 17 (a) Type 4 element (green) that shares a face with a Type 1 element (white) in Cartesian coordinate system. Black points are the vertices of the elements, grey points are the mid-edge nodes of the non shared edges, red points are the mid-edge nodes of the shared edges for the element outside the cone, i.e. linear mid-edge nodes computed in the original spherical frame, and green points are the mid-edge nodes of the shared face for the element crossing the cone boundary, i.e. linear mid-edge nodes computed in the 90° rotated spherical frame. (b) In the original spherical frame, the Type 1 element (white) has straight edges and its mid-edge nodes are computed following the equations (61a) to (61f). The Type 4 element (green) has curved edges since its mid-edge nodes are linear in the 90° rotated spherical frame. (c) In the 90° rotated spherical frame, the Type 4 element (green) has straight edges and its mid-edge nodes are computed following the equations (61a) to (61f). The Type 1 element (white) has curved edges since its mid-edge nodes are linear in the original spherical frame.

Type 4 elements

The mid-edge node position computed on an edge of an element crossing the cone boundary (computed in the 90° rotated spherical frame) is not congruent with the mid-edge node position computed on the same shared edge of a Type 1 element (computed in the original spherical frame; Figure 17). This distortion is assessed by measuring the distance between mid-edge nodes, for the same shared edge, computed in both 90° rotated spherical frame and original spherical frame. Figure 18a shows the mean distance of mid-edge nodes on shared edges between elements crossing the cone boundary and adjoined elements outside the cone as a function of the mesh node number for different cone angles. The distance between mid-edge nodes is directly related to the element size. The cone angle dependence of the mean distance is shown in Figure 18b for different mesh resolutions. Assuming a spherical shell mesh with an embedded high resolution subregion outside the cone, the coarse elements with element size around 2000 km cross the cone boundary. The best choice of θ_c to minimize the distance between mid-edge nodes of shared edges computed in both the 90° rotated spherical frame and the original spherical frame was empirically found to lie between 45° and 55°.

Isoparametric spherical elements link both the 90° rotated spherical frame and the original spherical frame in order to avoid the mismatch in the mid-edge nodes of the shared edges. The mid-edge nodes of the shared edges of the Type 4 elements are substituted by the mid-edge nodes of the shared edges of the Type 1 elements. In other words, some of the mid-edge nodes of the Type 4 elements are computed in the 90° rotated spherical frame and some other mid-edge nodes are computed in the original spherical frame. Therefore the Type 4 elements have curved edges in both spherical frames (Figure 19). In the mapping from local coordinates to spherical coordinates (first Jacobian of the Double Jacobian) for Type 4 elements, some faces are curved in spherical coordinates, whereas the faces in the local coordinate system are straight. In order to have a more accurate mapping of these elements, 20-node cubic elements are used as a superparametric elements to compute their first Jacobian of the Double Jacobian. Figure 20 shows the distance in km between a face's shape computed using 4-node linear spherical mapping and its shape computed in the 90° rotated spherical frame using its 4-node linear mapping, 10-node quadratic mapping, 14-node quasi-cubic mapping and 20-node cubic mapping. Note that the scale of colour bar is different for each case. The element shown in Figure 20 has an average edge length of 2000 km and its location in the mesh is shown in the inset in Figure 20a. For simplicity, from now on we will refer to the combination of DJ using 4-node linear mapping for Type 1-3 elements and DJ using 20-node cubic mapping for Type 4 elements as the DJ method. Note that for calculations on a spherical prism instead of the whole sphere, this additional element complexity is not needed in the DJ approach.

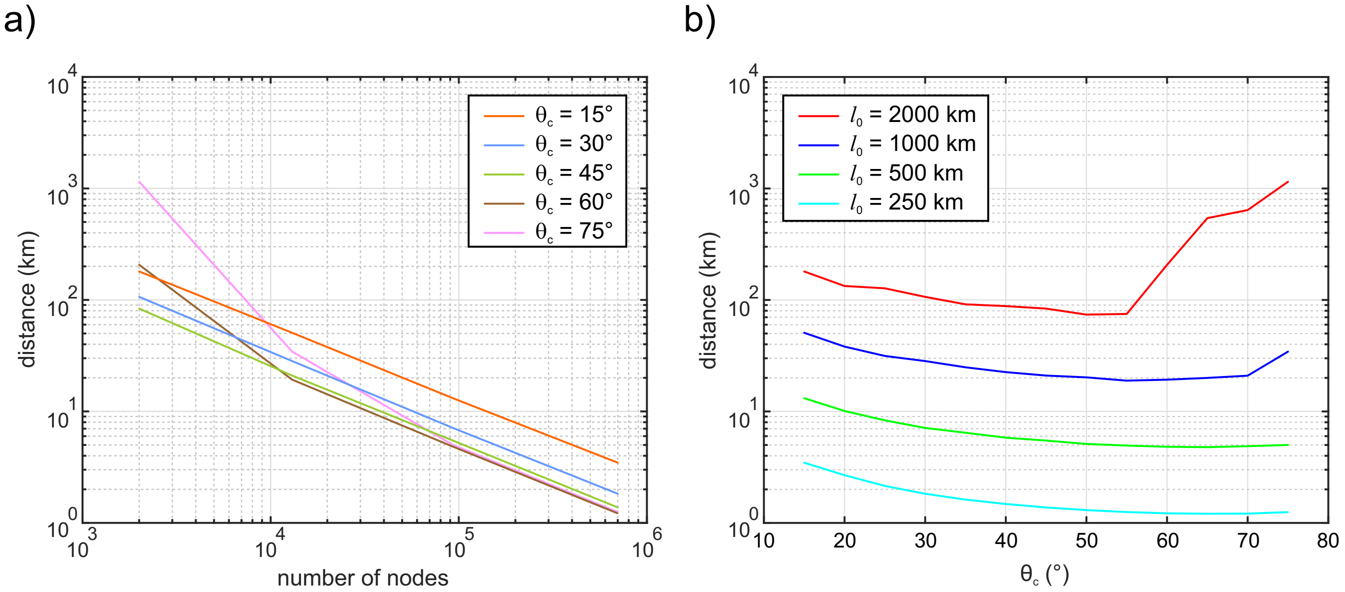


FIGURE 18 (a) Mean distance of mid-edge nodes on shared edges between elements crossing the cone boundary and adjoined elements outside the cone as a function of the mesh node number for different cone angles. (b) Mean distance of mid-edge nodes on shared edges between elements crossing the cone boundary and adjoined elements outside the cone as a function of the cone angle for different mesh resolutions.

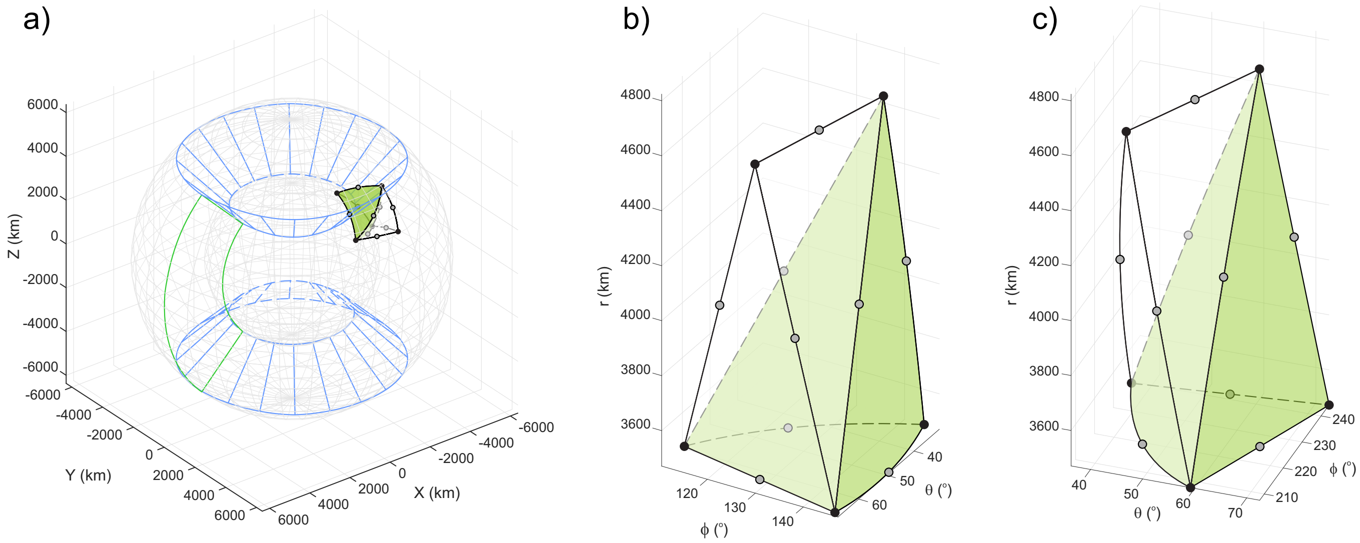


FIGURE 19 (a) Type 4 element (green) that shares a face with a Type 1 element (white) in Cartesian coordinate system. Black points are the vertices of the elements, grey points are the mid-edge nodes. (b) In the original spherical frame, the Type 1 element (white) has straight edges. The Type 4 element (green) has three curved edges and three straight edges. (c) In the 90° rotated spherical frame, the Type 4 element (green) has three curved edges and three linear edges. The Type 1 element (white) has curved edges since its mid-edge nodes are linear in the original spherical frame.

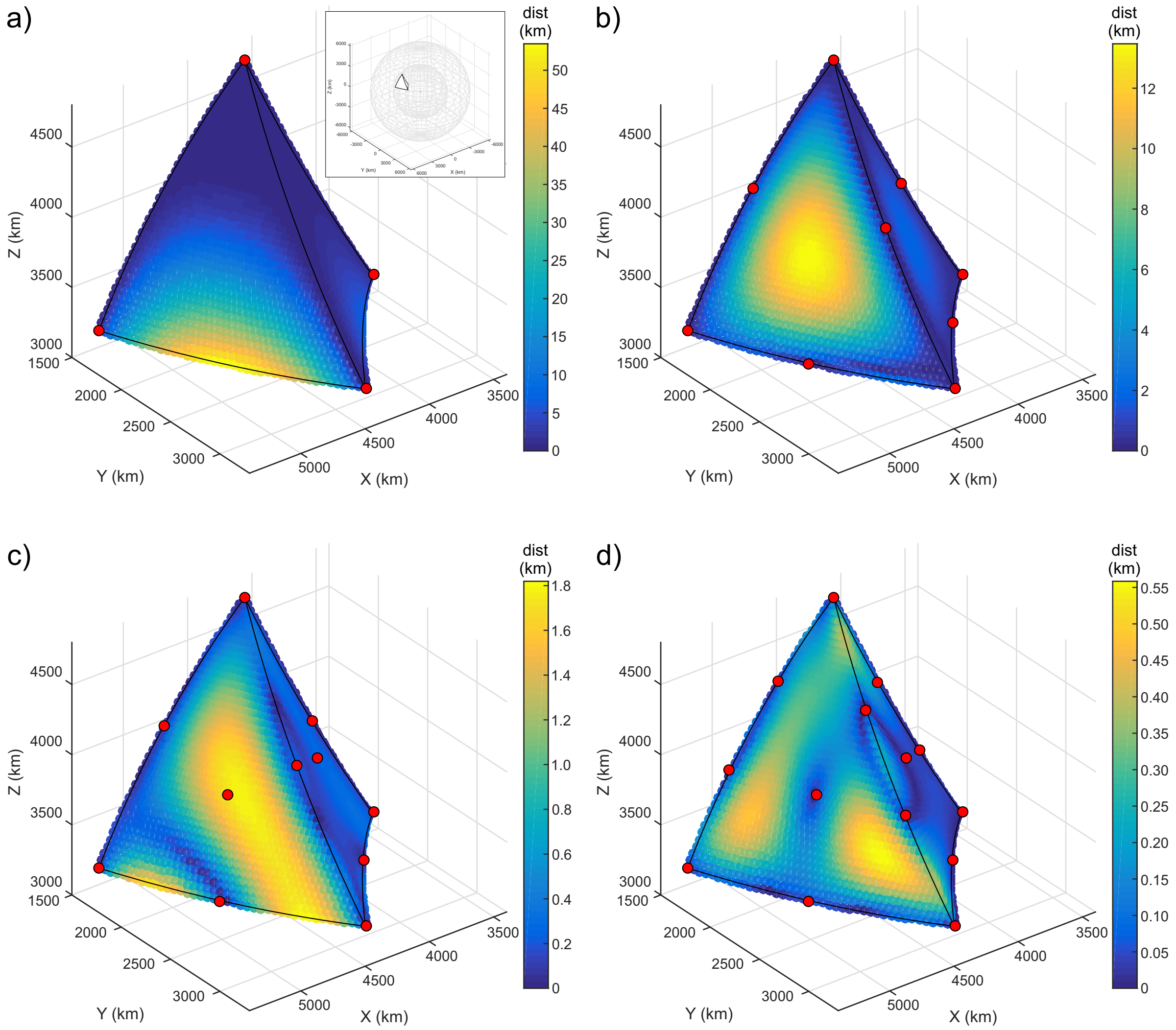


FIGURE 20 Distance in km (colours) between a face's shape computed by DJ using 4-node linear spherical mapping and its shape computed in the 90° rotated spherical frame using: (a) 4-node linear spherical mapping; (b) 10-node quadratic spherical isoparametric mapping; (c) 14-node quasi-cubic spherical superparametric mapping; (d) 20-node cubic spherical superparametric mapping. The position of the tetrahedron in the mesh is shown in the top right corner in (a). Black lines represent the curved edges of the tetrahedron and red points represent the mapping element's nodal position. Note that the scale in the colour bar is different for each case.

Application to multigrid meshes

One of the advantages of using quadratic elements with mid-edge nodes computed in spherical coordinates arises when a recursive refinement of the mesh is required, for example, when using adaptive mesh refinement or multigrid techniques. The multigrid method in combination with the Finite Element Method performs better when the fine mesh is nested within each coarser mesh [e.g. *Briggs et al.*, 2000]. Figure 21 shows three methods to recursively refine a Taylor-Hood element that has one of its faces on the surface of a spherical mesh. The first method splits the elements in Cartesian coordinates using the methodology described in Section *3-D quadratic elements with straight edges*. The 10-node tetrahedron is split into eight tetrahedra (3-simplices) and new mid-edge nodes are created (Figures 21a and 21b). The process is repeated recursively (Figure 21c). This method will keep the fine mesh nested within all coarser meshes (black dots are in the same position as dashed squares) but is not able to well approximate the face, which is a Cartesian plane, to the spherical surface (red dashed lines). The second method also splits the elements in Cartesian coordinates. However, the new vertices of the 'child' elements that are on the top surface are projected outwards to the actual spherical surface (Figures 21d, 21e and 21f). This method better approximates the top surface face to the true spherical surface as the number of level of multigrid increases, although the elements have straight Cartesian faces. A drawback is that the fine 'child' mesh is not completely nested within its coarser 'parent' mesh. The third method splits the elements in spherical coordinates using the methodology described in Section *3-D quadratic elements computed in spherical coordinates* (Figures 21g, 21h and 21i). This method creates true spherical faces for all 'parent' elements. Furthermore each fine mesh is nested within its coarser parent mesh. We employed this latter method in our mantle convection code.

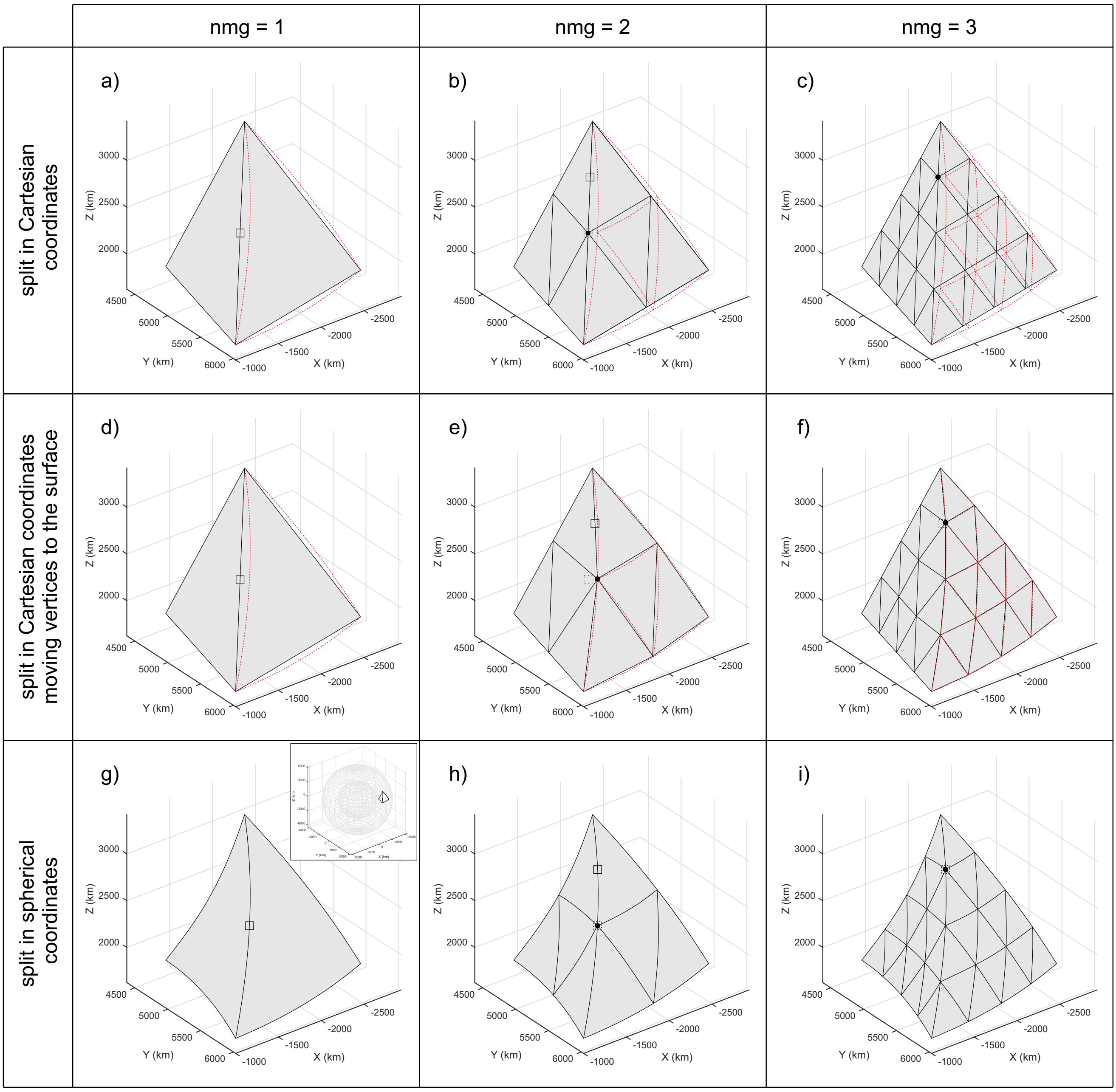

nmg = 1
nmg = 2
nmg = 3
split in Cartesian coordinates
split in Cartesian coordinates moving vertices to the surface
split in spherical coordinates
a)
b)
c)
d)
e)
f)
g)
h)
i)
Z (km)
3000
2500
2000
4500
5000
5500
6000
Y (km)
-1000
-1500
-2000
-2500
X (km)

FIGURE 21 Generation of multigrid meshes for three methods: (a), (b), and (c) split the element in Cartesian coordinates; (d), (e), and (f) split the element in Cartesian coordinates and project the new top surface vertices of 'child' elements towards the true spherical surface; (g), (h), and (i) split the element in spherical coordinates. The position of the tetrahedron in the mesh is shown in the top right inset in panel (g). The solid squares show the position one of the mid-edge nodes in the 'parent' element, the dashed squares represent the position of one of the mid-edge nodes of the 'parent' element in the 'child' element and the black dots show the position of one of the vertex of a 'child' element. Red dashed lines represent the spherical surface of the face.

Results

As in 2-D, differences between the standard Jacobian and Double Jacobian approaches are largest when large elements are used along the problem boundaries. The final 3D experiments are carried out in a spherical coarse mesh with an embedded high resolution subregion. Again, we perform three tests to assess the accuracy of the 3-D Double Jacobian formulation: (1) the hydrostatic 'no-flow' test proposed by *Pelletier et al.* [1989], (2) the Stokes problem solved for a small, dense, icosahedral sinker placed within a uniform density spherical shell and (3) the Method of Manufactured Solutions (MMS) [*Salari and Knupp*, 2000; *Roache*, 2002] applied to the steady state diffusion equation. We use M3TET [*Hasenclever*, 2010; *Hasenclever et al.*, 2011], a numerical code written in MATLAB to solve for the 3-D thermo-mechanical viscous flow and melting of the mantle using the standard Jacobian method in Cartesian coordinates. The matrix equations arising from the viscous flow problem are solved using a Conjugate Gradient algorithm [*Saad*, 2003] preconditioned with single V-cycles of geometrical multigrid [*Briggs et al.*, 2000]. To efficiently utilize multigrid, a recursive mesh refinement is performed as described above. The 3-D Double Jacobian approach has been implemented in M3TET_SPH, which is a spherical-coordinate version of M3TET. Both codes use the vectorized 'blocking' technique [*Dabrowski et al.*, 2008] for more efficient assembly of the finite element matrices in MATLAB.

A 'spherical no-flow' test is performed for P_2P_1LIN, P_2P_1ISO and P_2P_1DJ elements (Figure 22). The resolution of the meshes varies from l_0 = 2000 km to l_0 = 500 km and the number of integration points (ip) per element used are 5, 10 and 14, describing 3^{rd}, 3^{rd} and 5^{th} order accurate gaussian quadrature, respectively. For coarse meshes, the Double Jacobian gives a much more accurate zero flow solution for higher-order numerical integration (blue and orange dashed lines in Figure 22c) than for low-order integration. For example, for the mesh with l_0 = 2000 km (1940 nodes), the speed computed with P_2P_1DJ elements for the 'no-flow' test is between 0 and 0.02 mm/yr when using 10 or 14 integration points (orange bars in Figure 23b and Figure 23c). P_2P_1LIN elements are unable to produce 'no-flow' like results due to the size of the planar patches of irs boundary approximation to a spherical surface (purple bars in Figure 23).

The second test consists of solving the viscous flow for an icosahedral sinker placed within a spherical shell. The aim of this test is to check the reliability of the 3-D methods to reproduce realistic buoyancy forces within the high resolution region. The density contrast between the sinker and the surrounding mantle is 100 kg/m^3. The sinker is located at 100 km depth on the Z axis. The test has been performed for viscosity contrasts between the sinker and the surrounding mantle varying from 1 to 10^3, and the boundary condition is no-slip at both 'core-mantle' boundary and outer surface. The 3-D setup consists of a spherical shell with an embedded high resolution region centred on the polar axis. The sinker is an icosahedron with edge length l_0 = 25 km (Figure 24a). The purpose of this test is to check the buoyancy forces from a high viscosity non-deforming sinker. For such isolated buoyancy anomalies, the Double Jacobian approach (red dashed line in Figure 24b) yields very similar results to standard FE approaches (green dashed line and black solid line in Figure 24b) as all meshes have a close approximation to spherical boundaries above the region with the sinker. The 3-D methods have an accuracy that is most significantly shaped by mesh-discretization-related errors, i.e. mesh resolution is the controlling factor for these errors. We observe that at shallower depths within the high resolution region, the flow solution reproduces the falling speed better than when the sinker is located close to the transition to lower-resolution parts of the mesh (not shown). It is possible to improve the accuracy in regions within the high resolution near transitions to lower-resolution parts of the mesh by introducing a fictitious concentric internal boundary to the high-resolution portion of the mesh.

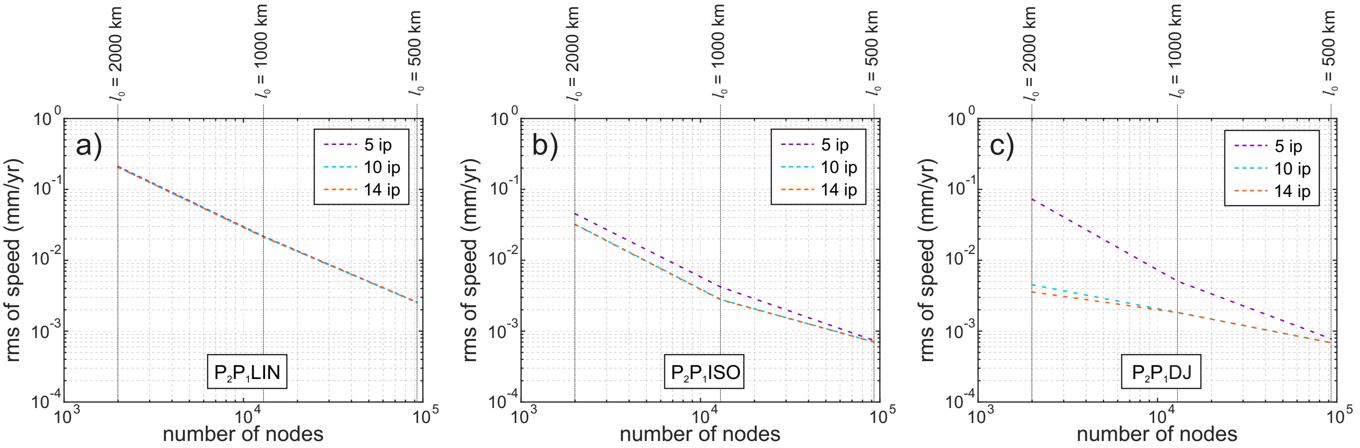


FIGURE 22 Root mean square of the speed for the 'no-flow' test in a spherical shell using: (a) P_2P_1LIN; (b) P_2P_1ISO; (c) P_2P_1DJ elements. Plots show accuracy (rms) as a function of the number of mesh nodes. Colour lines represent the number of integration points (ip) per element used.

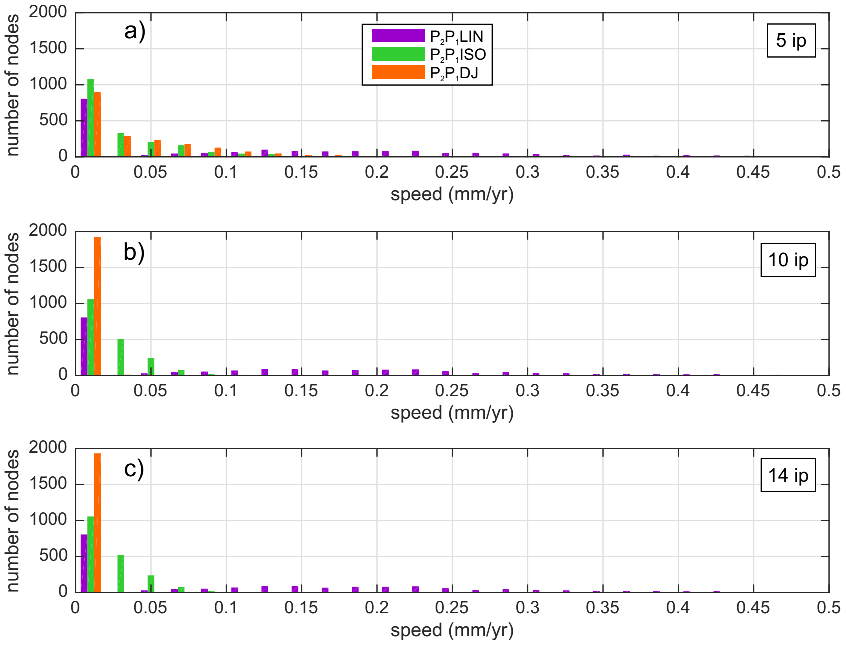


FIGURE 23 Histograms of speed for a spherical shell mesh with 1940 nodes and l_0 = 2000 km using: (a) 5 integration points (ip) per element (3^{rd} order accurate gaussian quadrature); (b) 10 integration points per element (3^{rd} order accurate gaussian quadrature); (c) 14 integration points per element (5^{th} order accurate gaussian quadrature). Purple, green and orange colours represent the element used: P_2P_1LIN, P_2P_1ISO and P_2P_1DJ, respectively. The bin size is 0.02.

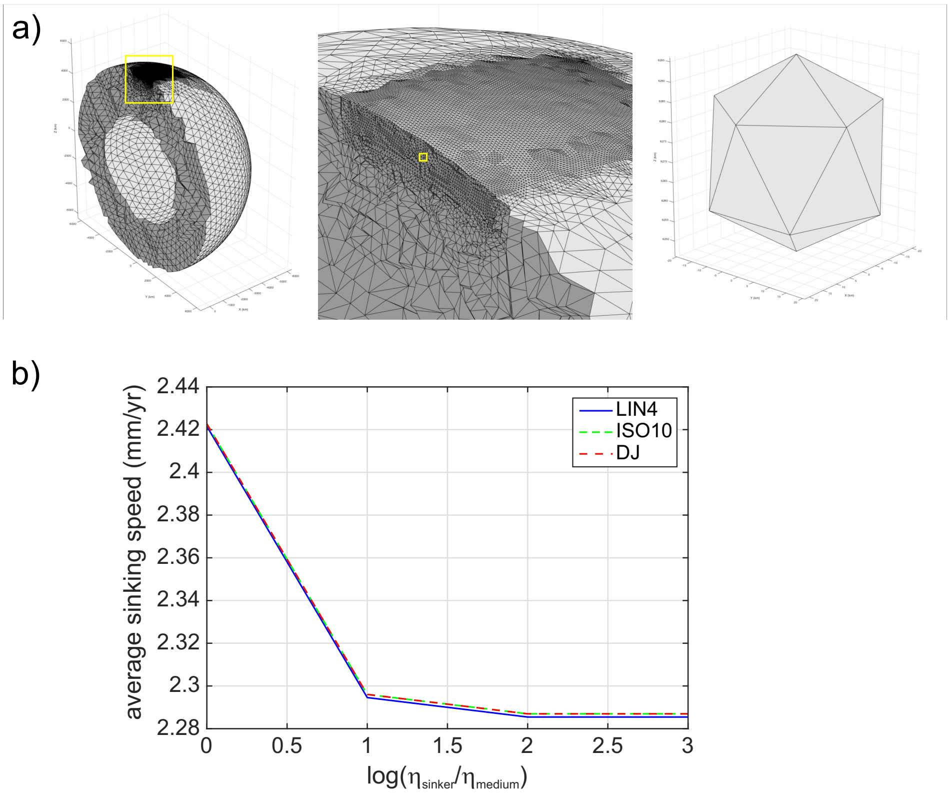


FIGURE 24 Test icosahedral sinker. (a) Spherical shell mesh with embedded high resolution region centred on its polar axis that has an embedded icosahedral sinker (right panel). The resolution within the high resolution region is $l_0 = 25$ km. Yellow lines represent the zoom region. (b) Average sinking speed within the icosahedral sinker as a function of the viscosity contrast between the sinker and its surrounding mantle.

The MMS test to verify the order of accuracy of the 3-D code is the 3-D analog to the 2-D MMS test. We use continuous Taylor-Hood tetrahedral elements with a 10-node quadratic-Temperature interpolation (P_2). The domain is a spherical shell with inner and outer radii equal to 0.5 and 1, respectively. The chosen 3-D solution is:

$$T_{mms} = e^{-(x^2+y^2+z^2)} \tag{62}$$

and also defines the Dirichlet boundary conditions. The solution, passed through equation (29), manufactures a compatible source term:

$$\Psi_{mms} = -\kappa\left(4x^2\, e^{-x^2-y^2-z^2} + 4y^2\, e^{-x^2-y^2-z^2} + 4z^2\, e^{-x^2-y^2-z^2} - 6\, e^{-x^2-y^2-z^2}\right). \tag{63}$$

Then we compute discrete solutions by solving equation (32). Figure 25 shows the RMS error (34) of temperature as a function of the average element size $< h >$:

$$< h >= \left(\frac{V}{N}\right)^{1/3}, \tag{64}$$

where V is the volume of the domain and N is the total number of elements. The error decreases suboptimally (quadratically) for P_2LIN elements and optimally (cubically) for P_2ISO and P_2DJ elements when refining the mesh. In this case, the P_2DJ element (red line in Figure 25) performs better than the P_2ISO element (green line in Figure 25) for all mesh resolutions as it can accurately match the spherical boundaries. Note the the error does not decrease when the geometry of the boundaries is kept fixed while refining the mesh (the cyan line in Figure 25 which uses meshes that are projected multigrid refinements of the coarsest mesh).

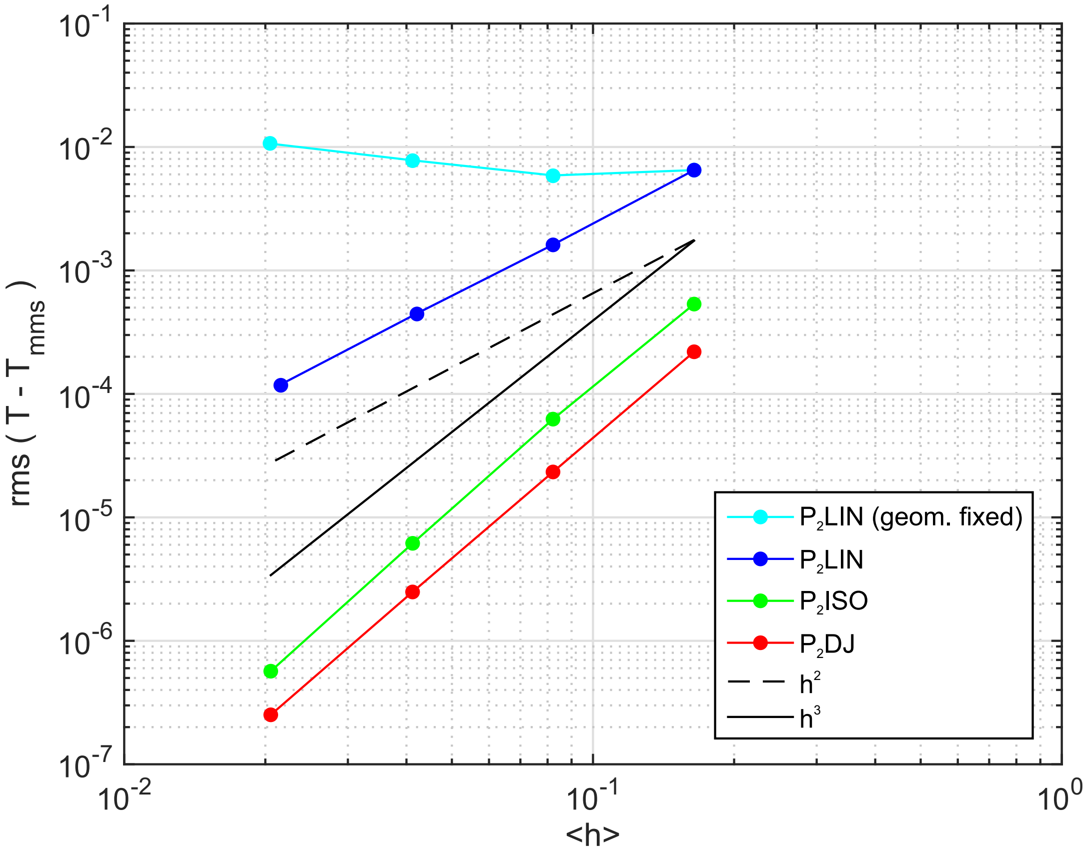


FIGURE 25 Root mean square error of temperature as a function of the average element size in the spherical shell mesh after using the MMS to the steady state diffusion equation. As in 2-D, the first-order planar patch approximation to the spherical shell boundary surfaces leads to a suboptimal h^2 convergence rate for the second-order P_2LIN element. Keeping the geometrical discretization fixed while increasing the number of elements even leads to non convergence of the solution (cyan line).

4 | SUMMARY

We have developed a new method, the Double Jacobian, to improve the efficient solution of finite element problems in cylindrical or spherical geometries. The method computes the local to Cartesian mapping as a two-stage process using analytical expressions that can be more rapidly computed than a general isoparametric or superparametric finite element. This approach is combined with linear-geometry (e.g. 'straight-edged') elements defined in polar or spherical coordinates. Their edges are all straight in polar or spherical coordinates with the exception (in 3-D) of a few elements that cross the cone (Type 4 elements). The use of linear simplex element geometries allows search routines to rapidly find arbitrary points in the resulting polar (2-D) triangular or spherical (3-D) tetrahedral elements. Of course, isoparametric or superparametric DJ elements can be used as needed to model deflections from cylindrical or spherical geometries.

We show that in cylindrical geometries, the DJ method is both more accurate and more efficient than Cartesian isoparametric or superparametric approaches. In spherical geometries, the DJ approach has to deal with the issue of the singular pole of the spherical coordinates system. To address this issue, we use an additional coordinate transformation during the assembly of element matrices near the pole. This approach has the advantage that it can exactly map the surfaces and internal spherical interfaces of the region. A somewhat simpler hybrid approach could combine the DJ method to assembly elements away from the pole with Cartesian superparametric assembly of elements near the pole.

The combination of the Double Jacobian formulation and quadratic Taylor-Hood elements ensures an effective method to study large 3-D problems with spherical geometry using multigrid techniques. Through different tests in both 2-D and 3-D we can confirm that, for relatively coarse meshes: (1) The DJ method is superior to conventional FE approaches in cylindrical and spherical geometries at not generating spurious flow associated with hydrostatic pressure fields; (2) The DJ method is equivalent to conventional methods at properly capturing the effects of local buoyancy force variations; (3) The DJ method is more accurate than conventional FE approaches at capturing the effects of internal shell-like layering with changes in internal density at layer interfaces.

References

Briggs, W. L., V. E. Henson, and S. F. McCormick (2000), *A Multigrid Tutorial*, 2nd ed., SIAM, https://doi.org/10.1137/1.9780898719505.

Dabrowski, M., M. Krotkiewski, and D. W. Schmid (2008), Milamin: Matlab-based finite element method solver for large problems, *Geochem. Geophys. Geosyst.*, *9*(4), doi: 10.1029/2007GC001719.

Hasenclever, J. (2010), Modeling mantle flow and melting processes at mid-ocean ridges and subduction zones development and application of numerical models, Ph.D. thesis, Universität Hamburg.

Hasenclever, J., J. P. Morgan, M. Hort, and L. H. Rüpke (2011), 2d and 3d numerical models on compositionally buoyant diapirs in the mantle wedge, *Earth Planet. Sci. Lett.*, *311*(1-2), 53–68, doi: 10.1016/j.epsl.2011.08.043.

Hughes, T. (1987), *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis*, 1st ed., Prentice-Hall.

Pelletier, D., A. Fortin, and R. Camarero (1989), Are fem solutions of incompressible flows really incompressible? (or how simple flows can cause headaches!), *Int. J. Numer. Methods Fluids*, *9*(1), 99–112, doi: 10.1002/fld.1650090108.

Roache, P. J. (2002), Code Verification by the Method of Manufactured Solutions, *J. Fluids Eng.*, *124*(1), doi: 10.1115/1.1436090.

Saad, Y. (2003), *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM.

Salari, K., and P. Knupp (2000), *Code Verification by the Method of Manufactured Solutions*, Sandia National Laboratories, Albuquerque.

Zhong, S., M. T. Zuber, L. Moresi, and M. Gurnis (2000), Role of temperature-dependent viscosity and surface plates in spherical shell models of mantle convection, *J. Geophys. Res.*, *105*(B5), 11063–11082, doi: 10.1029/2000JB900003.

Chapter 4

3-D modelling of the South Atlantic rifting

Jorge M. Taramón, Jason P. Morgan, Jörg Hasenclever, Marta Pérez-Gussinyé. 3-D modelling of South Atlantic rifting. Manuscript planned for submission at Nature Geoscience, 2018c.

Authors contribution

JPM and JMT designed the research. JMT programmed the code to implement the velocity boundary conditions extracted from GPlates. JMT developed the 3-D spherical geodynamic code (based on the 3-D Cartesian version developed by JPM and JH) in discussion with JPM and JH. JH implemented melting in the code. JMT designed the tests in discussion with JPM. JMT wrote the manuscript in collaboration with JPM and JH. MP reviewed the manuscript.

Received ; | Revised ; | Accepted

DOI: xxx/xxxx

ARTICLE TYPE

3-D modelling of South Atlantic rifting

Jorge M. Taramón*[1] | Jason P. Morgan[1] | Jörg Hasenclever[2] | Marta Pérez-Gussinyé[3]

[1]Department of Earth Sciences, Royal Holloway University of London, Egham, Surrey, UK
[2]Institute of Geophysics, Hamburg University, Hamburg, Germany
[3]MARUM - Center for Marine Environmental Sciences, University of Bremen, Bremen, Germany

Correspondence
*Jorge M. Taramón, Department of Earth Sciences, Royal Holloway University of London Egham, Surrey TW20 0EX, United Kingdom. Email: jorge.taramongomez.2014@live.rhul.ac.uk

Present Address
Department of Earth Sciences, Royal Holloway University of London Egham, Surrey TW20 0EX, United Kingdom

Abstract

We study the potential mantle flow associated with the early rifting evolution of the South Atlantic, exploring in particular the influence of Tristan da Cunha plume and initial lithospheric thickness variations on this system. Global plate motion boundary conditions and plate boundaries are updated using GPlates every 1 Myr. While surface plate motions are prescribed everywhere on the sphere, buoyancy forces are only considered within a 4200 km (N-S) x 1800 km (E-W) x 300 km (depth) high-resolution sub-region to avoid convective instabilities in the coarser mesh regions. Our models show the migration of hotter and weaker plume material towards the rifting region before break-up. This flow is influenced by the lateral thickness variations in the initial structure of the lithosphere. Once the plume material reaches the rifting region, it is found to preferentially migrate southwards. This preferential southward flow appears to be due to the presence of thicker São Francisco and conjugate Congo cratonic roots in the North combined with a ridge 'suction' force due to stretching of non-cratonic lithosphere in the South. Regions of plume-influenced rifting are found to have significant early uplift in comparison to 'non-volcanic' rifting sectors.

KEYWORDS:

1 | INTRODUCTION

Volcanic rifted margins are associated with large magmatic extrusion during the last stage of continental break-up and the beginning of sea-floor spreading. These margins are usually identified by offshore thick wedges of volcanic material called Seaward Dipping Reflectors (SDRs), a high-velocity lower crust and a thickened oceanic crust [e.g. *Mutter et al.*, 1982; *White et al.*, 1987; *Gladczenko et al.*, 1997; *Holbrook et al.*, 2001; *Franke*, 2013]. Volcanic margins are often associated with continental flood basalts that erupt in a short geological timelapse of a few million years [*Morgan*, 1971; *White and McKenzie*, 1989; *Coffin and Eldholm*, 1994; *Courtillot et al.*, 1999].

In the South Atlantic in particular, it is thought that the origin of the continental flood basalts located in the Paraná and Etendeka traps as well as the Walvis Ridge and Rio Grande Rise are associated with the Tristan da Cunha mantle plume [e.g. *Morgan*, 1981; *O'Connor and Duncan*, 1990]. Although mantle plumes may initiate continental rifting and break-up, also known as plume-induced rifting, the Tristan da Cunha plume was not thought to trigger the opening of the South Atlantic since there is evidence for sea-floor spreading starting in the south and propagating northwards [*Nürnberg and Müller*, 1991; *Storey*, 1995; *Fromm et al.*, 2015].

Mantle plumes, considered to be thermal anomalies, arise from the core-mantle boundary and carry warmer low density material that eventually ponds at the base of the lithosphere to create a volcanic chain as the plate moves over the fixed plume [*Morgan*, 1971]. Near a ridge axis, plume material might flow towards regions of thin lithosphere producing enhanced melting and volcanism by decompression of the asthenosphere [*Morgan et al.*, 1995; *Sleep*, 1996]. Plume material flowing to regions of thin lithosphere reduces the lithosphere strength and has been proposed to aid in the final stage of the continental rifting and break-up [*Buiter and Torsvik*, 2014]. Thermo-chemical plumes have been proposed to be an effective mechanism to erode the base of the lithosphere [*Sobolev et al.*, 2011] and have been also linked to large mass-extinctions events [*Courtillot et al.*, 1999]. Other mechanisms have been proposed to explain the offshore along-strike volcanism observed in the South Atlantic. For example, *Koopmann et al.* [2014] suggest that a segmentation in rifting is the main mechanism that controls the volcanism in that region. Other authors, following *Sleep*'s hypothesis, suggest that the sublithospheric topography may have played an important role in guiding plume material towards the south, where lithosphere is thin, producing the observed volcanism and producing much less volcanism towards the north where the lithosphere is thick [*Gladczenko et al.*, 1997; *Buiter and Torsvik*, 2014; *Taposeea et al.*, 2017].

The plume location and its arrival to the base of the lithosphere are not well constrained. Some authors suggest Africa was the place of the plume impact [*Duncan*, 1984; *Fromm et al.*, 2015]. Others locate it beneath the Paraná basin since most of the continental flood basalts are located on South America [e.g. *O'Connor and Duncan*, 1990; *Turner et al.*, 1994; *VanDecar et al.*, 1995]. The flood basalt emplacement within the Paraná and Etendeka traps located in Brazil and Namibia, respectively, was a fast process that lasted approximately 1 Myr starting at 134-132 Ma [*Renne et al.*, 1992, 1996; *Thiede and Vasconcelos*, 2010]. However, *Dodd et al.* [2015] suggest it took more than 4 Myr for the event, and used this to explain why the Paraná-Etendeka LIP is not associated with a mass-extinction like many other LIPs. The age of the opening of the South Atlantic Ocean is not well constrained either, varying from 126 Ma to 132 Ma [*Torsvik et al.*, 2009]. *Buiter and Torsvik* [2014] argue that flood basalt emplacement and break-up occurs almost simultaneously in several conjugate margins of the Atlantic and Indian oceans.

SDRs have been observed along more than 2000 km of the South American and South African margins with a maximum width of the SDRs located between the continental flood basalt province of Paraná and the Rio Grande Rise [*Gladczenko et al.*, 1997]. The African margin shows an asymmetry between volcanic margins south of the Walvis Ridge [e.g. *Bauer et al.*, 2000] and non-volcanic margins north of the Walvis Ridge [e.g. *Contrucci et al.*, 2004]. A recent study shows a sharp transition between volcanic and non-volcanic margin across the Florianópolis Fracture Zone and Walvis ridge along-strike [*Fromm et al.*, 2015]. In the South America conjugate margin there is also asymmetry between volcanic margins south of the Rio Grande Fracture Zone [e.g. *Stica et al.*, 2014] and non-volcanic margins north of the Campos basin [e.g. *Moulin et al.*, 2010]. The presence of Aptian salt along Santos and Campos basins makes it difficult to interpret seismic images to find possible SDRs. *Mohriak et al.* [2008] consider that the central segment of the South Atlantic is associated with SDRs. *Blaich et al.* [2011] argue that the magmatism experienced in that region was not sufficient to form SDRs. The contrast between volcanic margin south of the Rio Grande Rise and Walvis Ridge and non-volcanic margin to the north (Figure 1 and Figure 2b) keeps controversial the role of the Tristan da Cunha plume in the continental break-up since it departs from the conventional plume-head scenario, that predicts symmetrically enhanced volcanism along the plume head (Figure 2a).

The choice of far-field boundary conditions in planetary viscous flow problems often strongly shapes the large-scale structure of a geosimulation. For example, numerical models in 2-D have shown that rift speed controls the margin symmetry and mantle exhumation [*Pérez-Gussinyé et al.*, 2006; *Brune et al.*, 2014]. A recent study suggest a strength-velocity feedback during the continental rifting with a plate acceleration before break up [*Brune et al.*, 2016]. In 3-D, numerical modelling of regional rifting is usually done in Cartesian rectangular boxes with simplified 'plate spreading' or 'uniform rifting'. For example *Koopmann et al.* [2014] model the rift migration of North and South Atlantic with a sequential activation of segments using the same extension velocity for all the segments. Oblique extension has been proposed to aid the continental rifting and break-up [*Brune et al.*, 2012]. *Koptev et al.* [2016] use 3-D numerical models to show the interaction between mantle plumes and cratonic lithosphere applied to the Central-East African rift. One way to improve the capabilities of regional models is to include more accurate boundary conditions by using a nested modelling approach, in which a global model is used to determine a large-scale flow pattern that is then imposed as a constraint along the boundaries of the region to be modelled [e.g. *Mihalffy et al.*, 2008]. In a recent study, *Gassmöller et al.* [2016] use the nested approach to model in 3-D the long-term evolution of Tristan da Cunha plume and its interaction with the ridge.

Here we use a different approach by using 3-D spherical global numerical models with an embedded high-resolution subregion. Thereby, regional rifting processes can be modelled with no fictitious internal boundaries for which poorly-known velocities or stresses need to be prescribed. We use this approach to study the potential mantle flow associated with the first 30 Myr of rifting evolution in the South Atlantic. We explore the influence of Tristan da Cunha plume and initial lithospheric thickness variations on plume and mantle flow during rifting. Our results suggest a lateral plume flow scenario in which hotter and weak plume material preferentially migrates southwards. This preferential southward flow appears to be mainly due to the

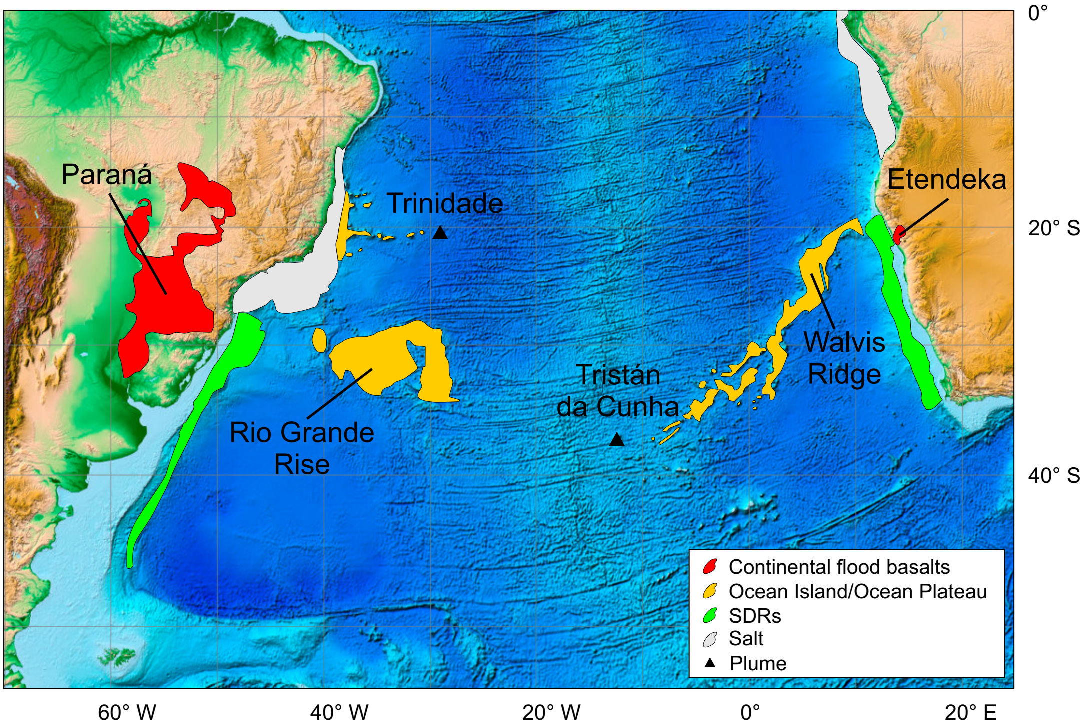


FIGURE 1 Map of the South Atlantic from ETOPO1 [*Amante and Eakins*, 2009] with onshore and offshore flood basalts and seaward dipping reflectors (SDRs). Structures from *Gladczenko et al.* [1997]; *Moulin et al.* [2010]; *Blaich et al.* [2011]; *Stica et al.* [2014].

presence in the North of thicker São Francisco and conjugate Congo cratonic roots during early rifting combined with 'suction' associated with early plate stretching in the non-cratonic regions. We propose this mechanism led to the observed preferential southward formation of early-rifting-related SDRs along South Atlantic margins with respect to their Tristan da Cunha plume progenitor.

2 | METHODOLOGY

2.1 | Numerical methods

We use M3TET_SPH, a numerical code written in MATLAB (http://www.mathworks.com) to solve for the thermo-mechanical viscous flow evolution of the mantle in spherical geometries. M3TET_SPH is based on the Cartesian verison, M3TET [*Hasenclever*, 2010; *Hasenclever et al.*, 2011]. The mantle is modelled as a 3-D incompressible fluid that satisfies the Boussinesq approximation. Using the index notation with the Einstein summation convention, the coupled equations of conservation of mass, force balance and energy are given by:

$$\frac{\partial u_i}{\partial x_i} = 0\,, \tag{1}$$

$$\frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j} - \rho g_i\,, \tag{2}$$

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x_i^2} - u_i \frac{\partial T}{\partial x_i}\,, \tag{3}$$

where u_i are the individual velocity components, x_i are the physical components, p is the pressure, τ_{ij} is the deviatoric stress tensor, ρ is the density, g_i are the components of the gravitational acceleration, T is the temperature, t is time and κ is the thermal diffusivity. The constitutive law is given by

$$\tau_{ij} = \eta \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)\,, \tag{4}$$

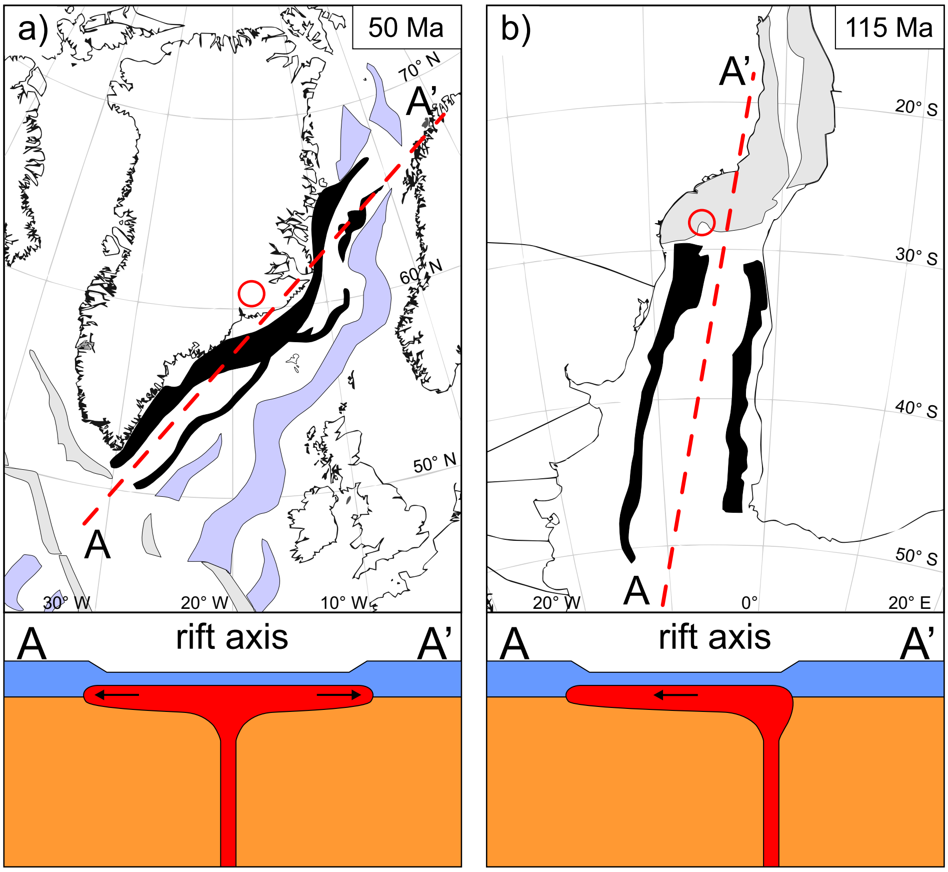


FIGURE 2 Differences between (a) symmetrical plume-head scenario (reconstruction of the North Atlantic at 50 Ma) and (b) asymetric lateral plume-flow scenario (reconstruction of the South Atlantic at 115 Ma). Black colour shows the position of SDRs (volcanic margins) and grey and purple colours represent non-volcanic margins and non-volcanic basins, respectively. Structures for (a) from *Lundin and Doré* [2011] and for (b) from *Gladczenko et al.* [1997]; *Moulin et al.* [2010]; *Blaich et al.* [2011]; *Stica et al.* [2014]. Reconstructions made using GPlates and plate kinematic reconstructions given by *Gurnis et al.* [2012]. Red circle represents the location of the mantle plume. Bottom insets show along-rift axis sketches for each scenario.

where η is the viscosity. The density, that controls the buoyancy-driven flow, is given by the simplified equation of state

$$\rho = \rho_0 \left(1 - \alpha \left(T - T_0\right)\right) , \tag{5}$$

where $\rho_0 = 3300$ kg m^{-3} is the density at reference mantle temperature $T_0 = 1300\,°C$, and $\alpha = 3$ x 10^{-5} K^{-1} is the thermal expansion coefficient. Tetrahedral P2P1 Taylor-Hood elements with continuous quadratic velocity and linear pressure interpolation functions are used to discretize the equations (1) to (3). The iterative solution technique constists of a conjugate gradient algorithm that is preconditioned by the multigrid method [*Hasenclever*, 2010; *Hasenclever et al.*, 2011]. The 3-D code is parallelized using the MPI-wrapper routines in MATLAB's *Parallel Computing Toolbox*. The Double-Jacobian approach (Chapter 3) is used to improve the efficiency when solving problems in a spherical geometry as well as to speed-up the particle search routines in curved-edge elements. The thermal advection-diffusion problem is solved by operator splitting [*Hasenclever*, 2010]. The diffusion part is discretized using quadratic-order elements and the equations are solved using a conjugate gradient algorithm. Temperature advection is done by a Semi-Lagrange advection scheme with second-order Predictor-Corrector back-tracking and cubic interpolation [*Shi*, 2012]. Melting is implemented following the approach in *Hasenclever* [2010]. The unstructured finite element spherical mesh is generated using MESH_3D_SPRING_SPH (Chapter 2).

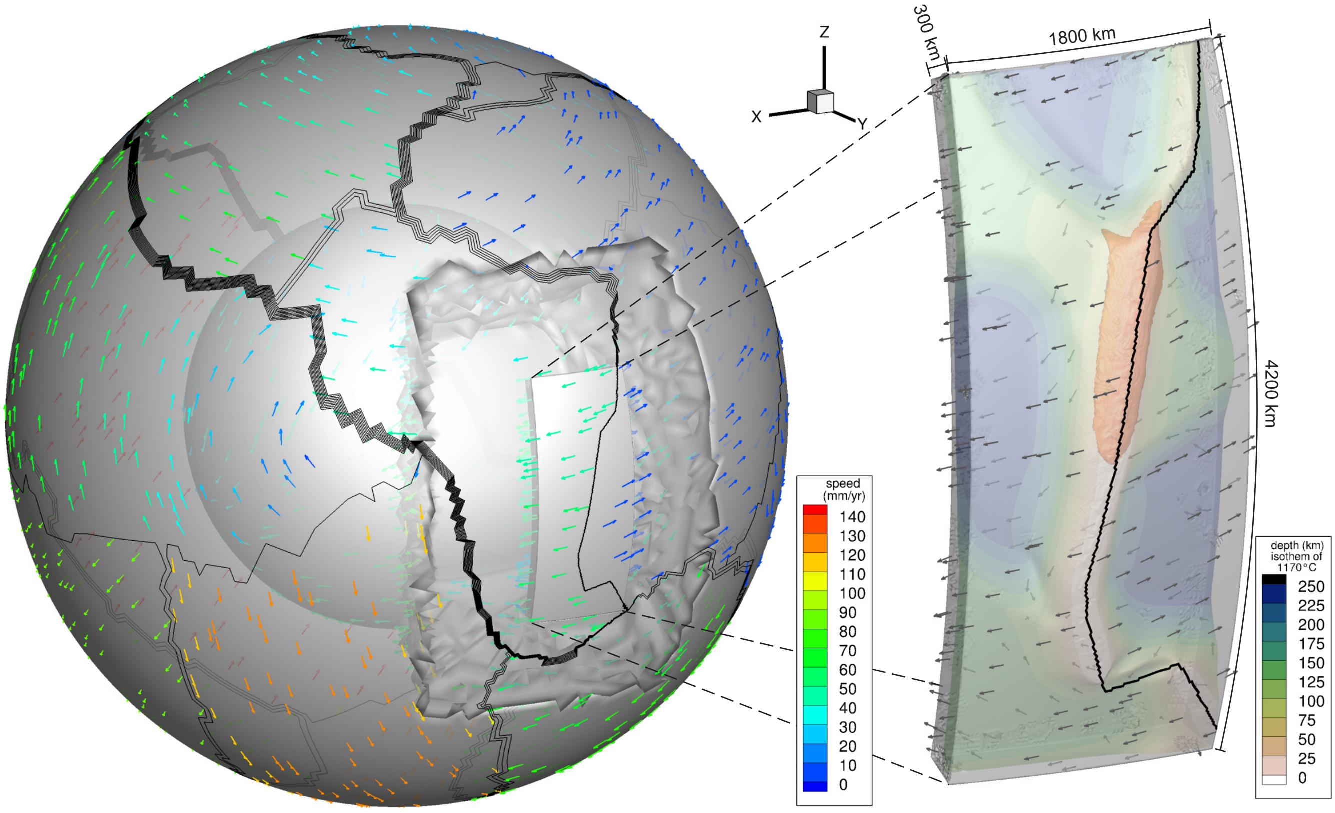


FIGURE 3 Model domain representing a 3-D spherical shell for the Earth's mantle (left) and zoom for the embedded high resolution region (right). Black lines represent the plate boundaries. Colour arrows are the speed for the plate motion (left). In the embedded high resolution region (right) the colour represents the depth of the isosurface of $T = 1170\,°C$ and black arrows are the velocity field obtained after imposing the plate motion boundary conditions. Red color shows plume material with the isosurface of $\log(\eta) = 18.2$.

2.2 | Model design

The 3-D model domain consists of a coarse spherical shell with the Earth's mantle dimensions that includes an embedded high resolution region around the area of interest (Figure 3). An embedded calculation avoids the 'edge' boundary conditions used in the nested modelling approach since they are replaced by global plate motion boundary conditions on the surface of the sphere. The embedded high resolution region is 300 km deep and extends 4200 km in latitudinal direction and 1800 km in longitudinal direction. A transition region surrounds the high resolution region in order to have a smooth spatial variation from the element size within the global coarse mesh (500 km) to the element size within the high resolution region (25 km). The mesh has 1.7 million elements (18 million degrees of freedom), with 66% of elements within the high resolution region. Buoyancy forces are only considered within the high resolution region to avoid convective instabilities in coarser mesh regions. The use of tetrahedral P2P1 Taylor-Hood elements allows a higher accuracy with a distance between mesh points of around 12.5 km.

2.3 | Initial conditions

The velocity boundary conditions are free slip along the core-mantle boundary and prescribed plate motion through time along the top surface. The simulation time for the early rifting and break-up of the South Atlantic ocean goes from 130 to 100 Ma. In order to impose the velocity boundary conditions numerically we use GPlates (https://www.gplates.org). The plate velocities are extracted every 1 Myr from the plate kinematic reconstructions given by *Gurnis et al.* [2012]. Velocities are linearly interpolated for each 0.1 Myr time step of the simulation.

The initial thermal structure is given by considering the lithosphere to be the thermal boundary layer of a half-space cooling model. Plate thickness variations from continental to cratonic lithosphere are simulated using ages of 100 and 350 Myr, respectively. This results in a depth for the isotherm of $1170\,°C$ of 130 km for continental lithosphere and 245 km for cratonic lithosphere. Outside the high resolution region we assume a constant lithosphere age of 80 Myr. The craton contours are digitised from *de Wit et al.* [2008]. The model contains a single 'hot Tristan Plume'. The initial geometry of the plume tail consists of a cylinder of radius 100 km extending from 670 km depth to the bottom of the high resolution region. The initial thermal structure for the plume is assumed to follow a Gaussian-shaped radial temperature profile with a maximum temperature anomaly of $150\,°C$ with respect to background mantle. The velocity boundary conditions for the plume are implemented by a parabolic-shaped radial velocity profile with the maximum velocity in the centre of the plume tail. The maximum ascent velocity is given by

$$V_{max} = \frac{2Q_p}{\pi R^2}, \tag{6}$$

where Q_p is the plume flux (km^{3}yr^{-1}) and R is the plume tail radius (km). We consider the effects of a range of plume fluxes from $5-20$ km^3 yr^{-1} (Table 1), which is consistent with ~$20-40$ mantle plumes supplying an upward return flow to the shallow mantle that balances the ~300 km^3 yr^{-1} downward flux associated with plate subduction [cf. *Yamamoto et al.*, 2007].

We use a temperature and pressure dependent upper mantle rheology given by

$$\eta(T,p) = \eta_0 exp\left[\frac{1}{RT_0}\left(E_a\left(\frac{T_0}{T}-1\right)+pV_a\right)\right], \tag{7}$$

where $\eta_0 = 2$ x 10^{18} Pa·s is the reference viscosity, R is the universal gas constant, $E_a = 400$ kJ/mol is the activation energy and $V_a = 4$ x 10^{-6} m^3/mol is the activation volume [*Hirth and Kohlstedt*, 2003]. The minimum and maximum cut-off viscosities are 10^{18} Pa·s and 10^{23} Pa·s, respectively. For simplicity, the lower mantle viscosity is considered to be uniform with a value of 5x10^{21} Pa·s.

3 | RESULTS

We have performed several numerical experiments by varying the initial position of the plume, the plume flux and whether to include melting during the numerical experiments (Table 1 and Figure 4). We found two plume flow scenarios depending on the plume location.

TABLE 1 Model parameters and output southward plume-flow migration.

Model	Plume location (colat, lon)	Plume flux (km^3 yr^{-1})	Melting	Southward plume-flow migration
L1F10	L1 (118°, 354°)	10	No	Yes
L1F15	L1 (118°, 354°)	15	No	Yes
L1F20	L1 (118°, 354°)	20	No	Yes
L1F15M	L1 (118°, 354°)	15	Yes	Yes
L1bF15	L1b (120°, 0°)	15	No	Yes
L2F10	L2 (121°, 351°)	10	No	Yes
L2F15	L2 (121°, 351°)	15	No	Yes
L3F15	L3 (124°, 351°)	15	No	No
L4F7.5	L4 (127°, 348°)	7.5	No	No

3.1 | Lateral southward plume-flow scenario

Models where the plume is located at L1, L1b and L2 (see Figure 4) develop a lateral southward plume flow migration. For comparison purposes we present here the results for the models L1F15, L1F15M, L1bF15 and L2F15 (see Supporting Information for models L1F10, L1F20 and L2F10).

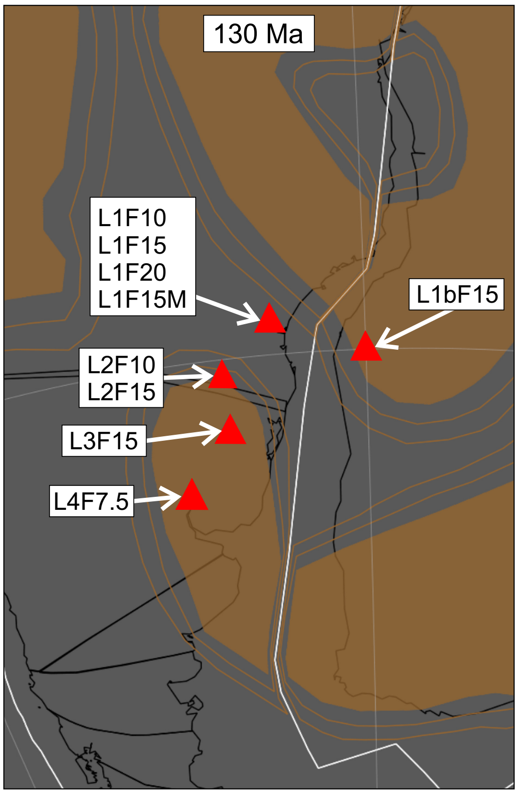


FIGURE 4 Plume location (red triangle) for each model listed in Table 1. Brown color represents the initial cratonic structure from *de Wit et al.* [2008]. Black and white lines are the reconstructed coastlines and the plate boundaries, respectively.

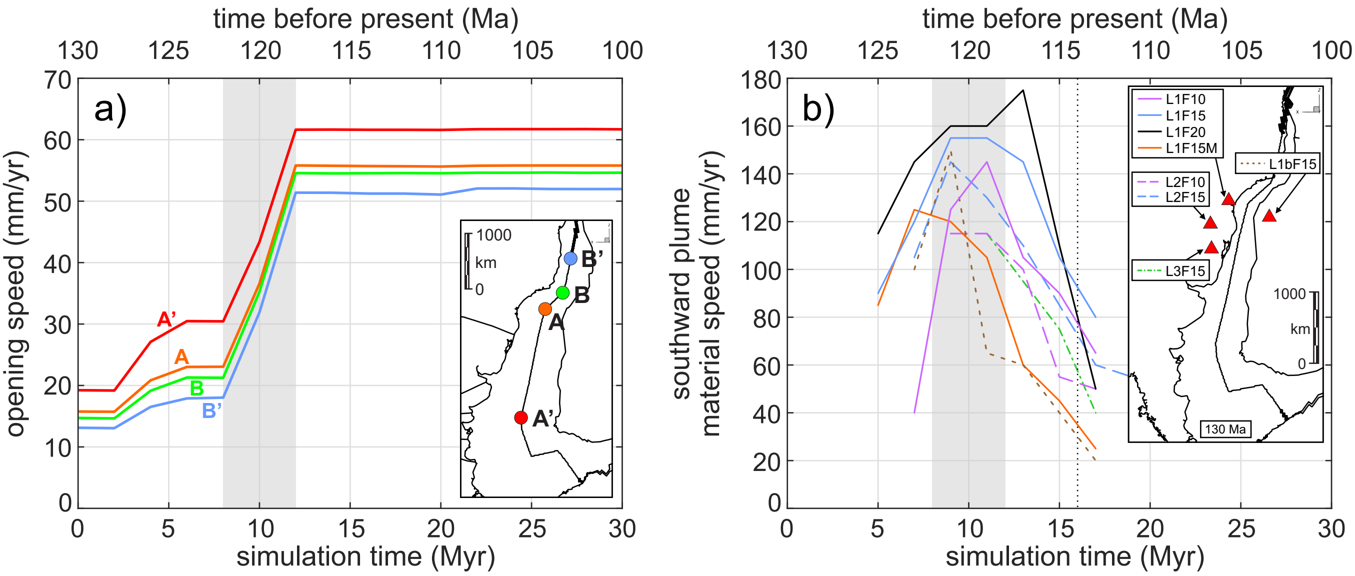


FIGURE 5 (a) Full opening speed in mm/yr of the South Atlantic between 130 Ma and 100 Ma. Kinematic reconstructions from *Gurnis et al.* [2012]. Grey colour represents an abrupt acceleration in the opening speed. (b) Average southward speed of plume material for every 2 Myr interval once the plume material reaches the rifting region. Vertical doted line represents when the North passage between São Francisco and Congo cratons opens allowing the northward migration of plume material.

Figure 6 and supporting information Movie S1 show the 3-D evolution of the model L1F15. At the start of the simulation there is a migration of hotter and weak plume material towards the rifting region before the break-up (Figure 6a). This migration is strongly influenced by the lateral thickness variations in the initial structure of the lithosphere (top inset Figure 6a). Once the plume material has reached the rifting region it flows preferentially southwards with much higher sub-rift temperatures in regions underlain by the plume material (Figure 6b). The southward migration is primarily due to 'suction' from the stretching region combined with a flow barrier of the thicker São Francisco and conjugate Congo cratonic roots in the North. After 16 Myr of rifting in the North, break-up finally occurs between São Francisco and Congo cratons. This opens a new 'path' for the plume material that starts to flow northwards but still in lesser amounts than flow to the south. At 18 Myr the plume material reaches its maximum southward influence (Figure 6c). However, faster opening and northwards flow lead to a shrinking of the region of the plume influence (Figure 6d). The top inset in Figures 6a-d shows the thermal thickness of the lithosphere, represented here by the depth of the isosurface of $1170\,°C$. The bottom inset in Figures 6a-d shows the plume contribution to the topography along the plate boundary between South America and Africa through the profiles AA' (south), AB (central) and BB' (north). The plume contribution to the topography is computed in two steps. First, the mantle contribution to isostatic topography for all plume models and for a plume free model is computed by an isostatic numerical integration [for a similar implementation see Appendix C of *Sparks et al.*, 1993]

$$\int_0^{d_c} (\rho(z) - \rho_m) g dz = \Delta\rho g h_{iso} \,, \tag{8}$$

where $d_c = 300$ km is the compensated depth, $\rho(z)$ is the depth dependent density, ρ_m is the mantle density, $\Delta\rho$ is the density contrast between the mantle and water, g is the gravity and h_{iso} is the isostatic topography. In order to discretize equation (8), $\rho(z)$ is sampled by interpolating the density in the radial direction along a column of points with a spatial interval $dz = 10$ km beneath each surface mesh point. Finally, the mantle contribution of a plume free model is subtracted from the mantle contribution of a plume model in order to compute the plume contribution to the topography. Figure 7 and supporting information Movie S2 show the evolution of the plume contribution to topography for the model L1F15. As the South America plate is moving over the plume, part of the plume's influence, as thermally-supported relief, is shaped by its motion and by rifting-related changes in lithosphere thickness. Regions of plume-influenced rifting are found to be associated with significant early uplift (green colour in Figure 7).

Figure 8 and supporting information Movie S3 show the 3-D evolution of the model L1F15M which explicitly includes the thermal effects of mantle melting. The behaviour of the plume material in this model with melting contribution is similar to the model without melting (L1F15). Note that the plume material is represented in the model L1F15M by an isosurface with $\log(\eta) = 18.5$ in comparison to $\log(\eta) = 18.2$ for model L1F15, e.g. it is roughly twice as viscous. The reason is because as melting consumes latent heat, the temperature of the melting plume material cools and therefore its temperature-dependent viscosity increases. The first 6 Myr of evolution are identical since no melting is produced at that time (Figure 8a). Similar to model L1F15, plume material starts to flow southwards once it reaches the rifting region. Melting occurs first along the profile AA', where the rifting velocities are higher than in the northern part. The region overlying the plume material shows a higher melt production rate per km upflow than in the southern end of the rifting region (black isosurface in Figure 8b). The isosurface for non-negligible melting rate (10^{-4} km^{-1}) shows the maximum extension of the melting region. After break-up between the São Francisco and Congo cratons melting appears in the northern part (Figure 8c). Regions with plume materal beneath the ridge present more melting rate per km upflow (a wider black isosurface above plume material in Figure 8c). Towards the end of the simulation, the width of the isosurface for melting rate per km upflow is still slightly wider in regions overlying the plume material (Figure 8d). Figure 9 and supporting information Movie S4 show the evolution of the plume contribution to topography for the model L1F15M. Again, there is a early uplift in regions of plume-influenced rifting. However, the plume contribution to the topography is lower in the model L1F15M than in model L1F15. This occurs because there is less plume material and because the mantle in the model with melting is cooler.

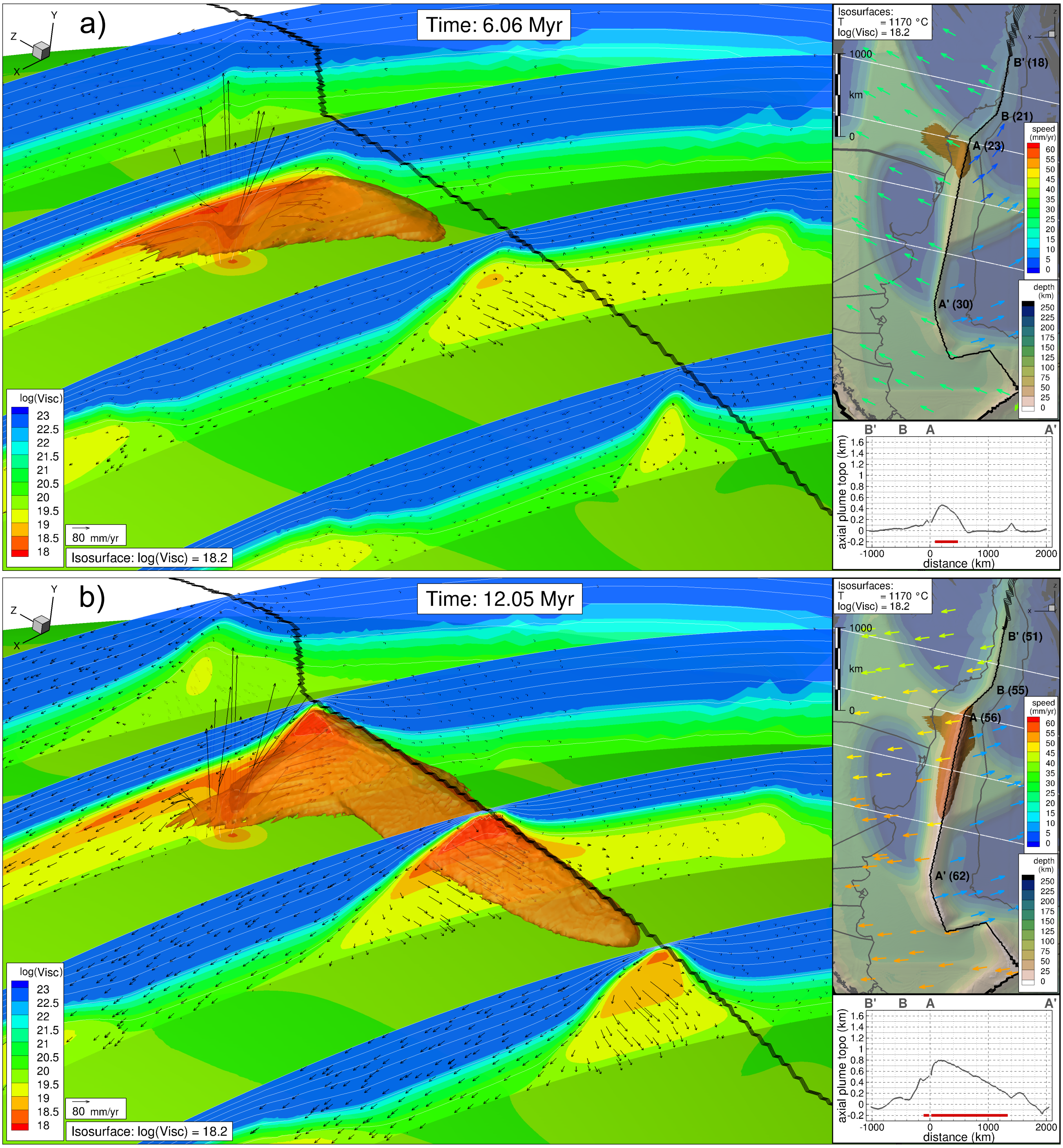


FIGURE 6 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1F15.

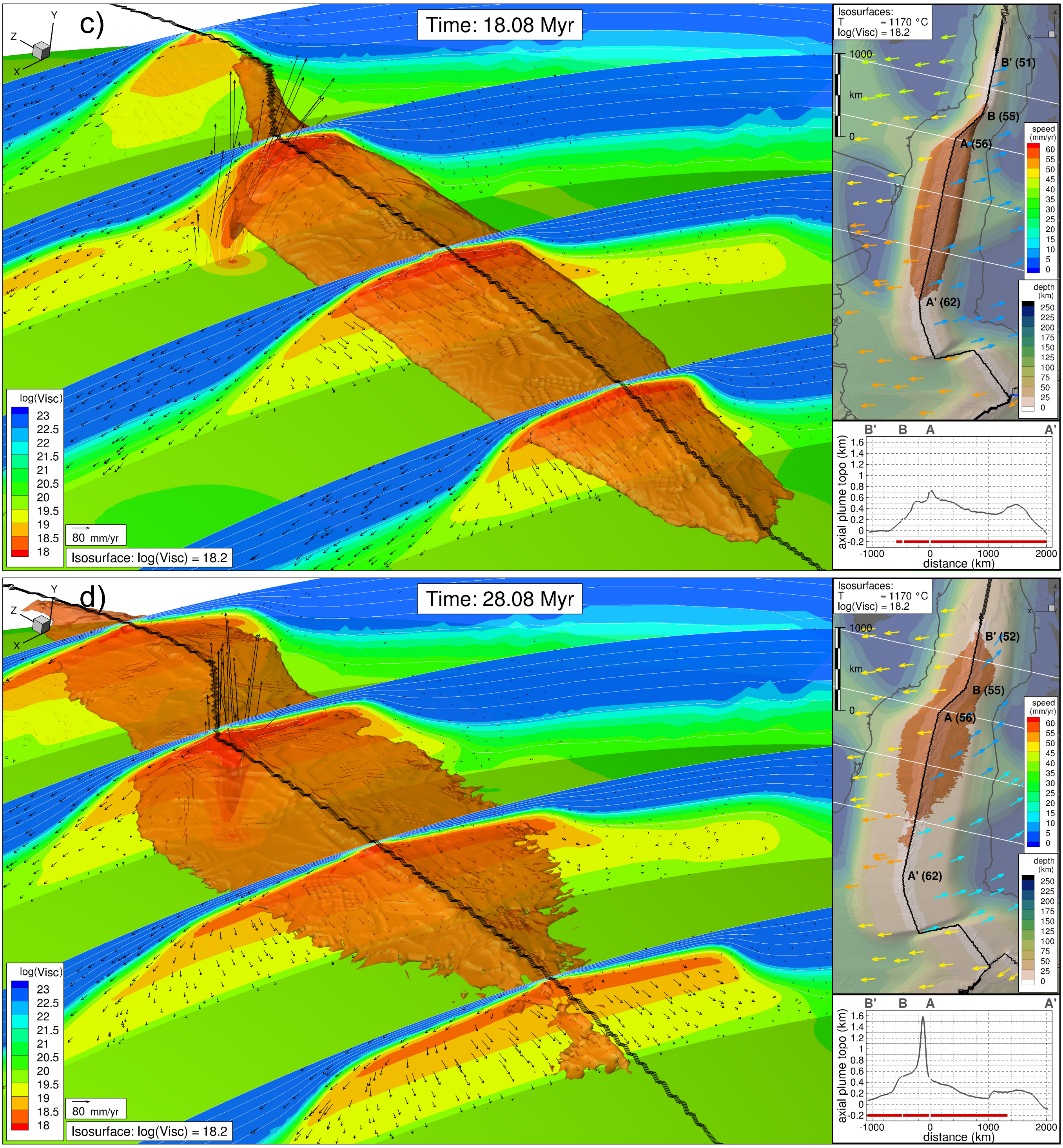


FIGURE 6 (Cont.) 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1F15. Main: 3-D images showing the geometry of the plume material at different times: (a) 6.06 Myr, (b) 12.05 Myr, (c) 18.08 Myr and (d) 28.08 Myr. Colour represents logarithm of viscosity. The velocity field and isotherms every 200 °C in the vertical cross sections are represented by arrows and white lines, respectively. Black line represents the plate boundary. The plume material is represented by the red isosurface with $\log(\eta) = 18.2$. Top inset: Top view of the 3-D evolution shown in the main inset. The isosurface with a temperature of 1170 °C is coloured with depth to show the lithospheric thickness variations. The plume material is represented by the red isosurface with $\log(\eta) = 18.2$. Colour arrows represent the top surface plate motion. Grey lines and black thick lines represent the reconstructed coastlines and the plate boundaries, respectively. Capital letters show the ends of the along ridge profiles shown in bottom inset. Numbers between parentheses show the full opening speed in mm/yr. Bottom inset: Plume contribution to axial topography in km. The red horizontal line represents the plume material beneath the ridge profile.

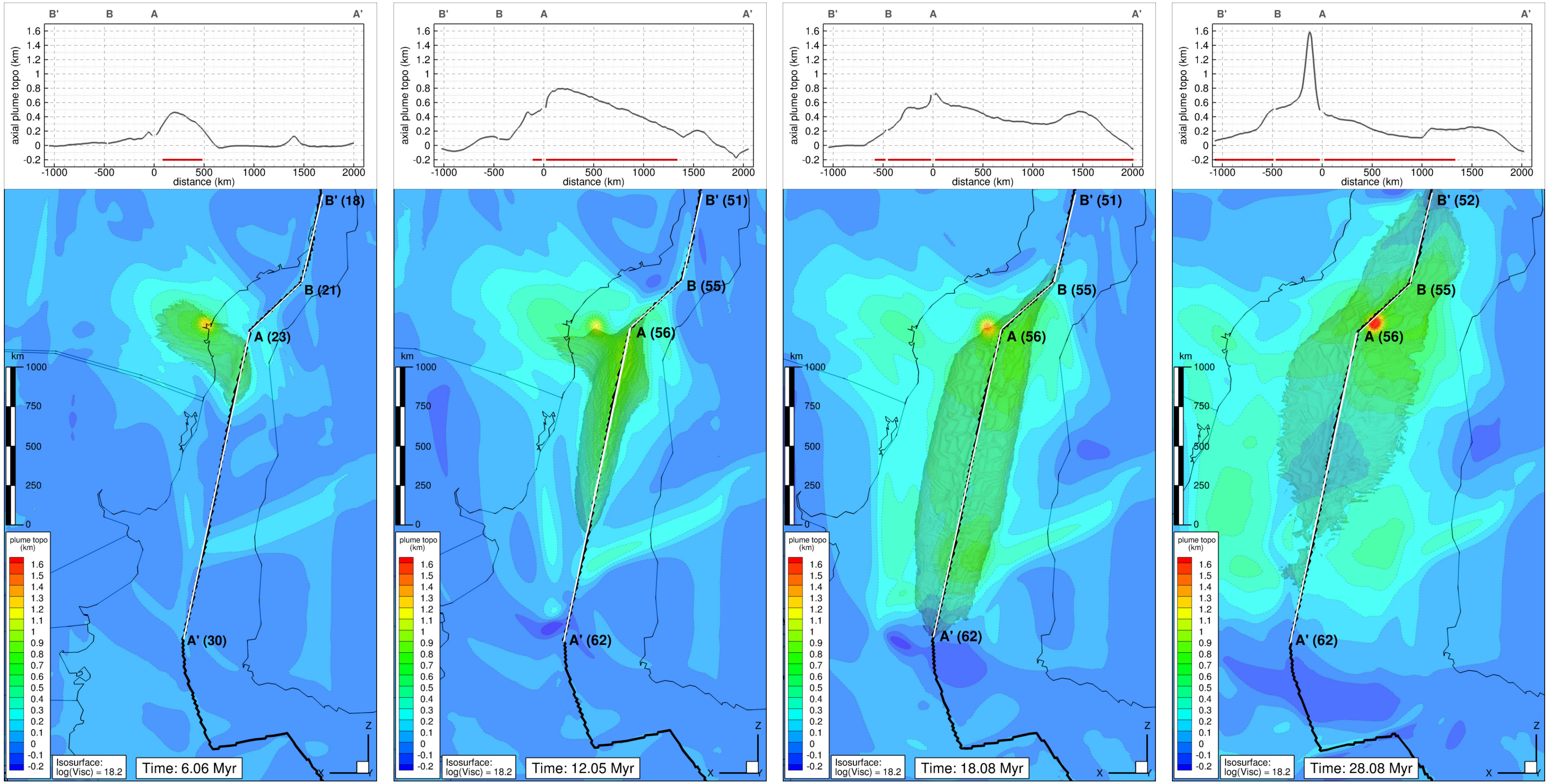


FIGURE 7 Evolution of the plume contribution to the topography for model L1F15. Colour represents the plume contribution to the topography in km. The plume material is represented by a isosurface with $\log(\eta) = 18.2$. Black thin and thick lines represent the reconstructed coastlines and the plate boundaries, respectively. Capital letters show the ends of the along ridge profiles shown in bottom inset. Numbers between parentheses show the full openning spped in mm/yr. Top inset: Plume contribution to axial topography in km. The red horizontal line represents the plume material beneath the ridge profile.

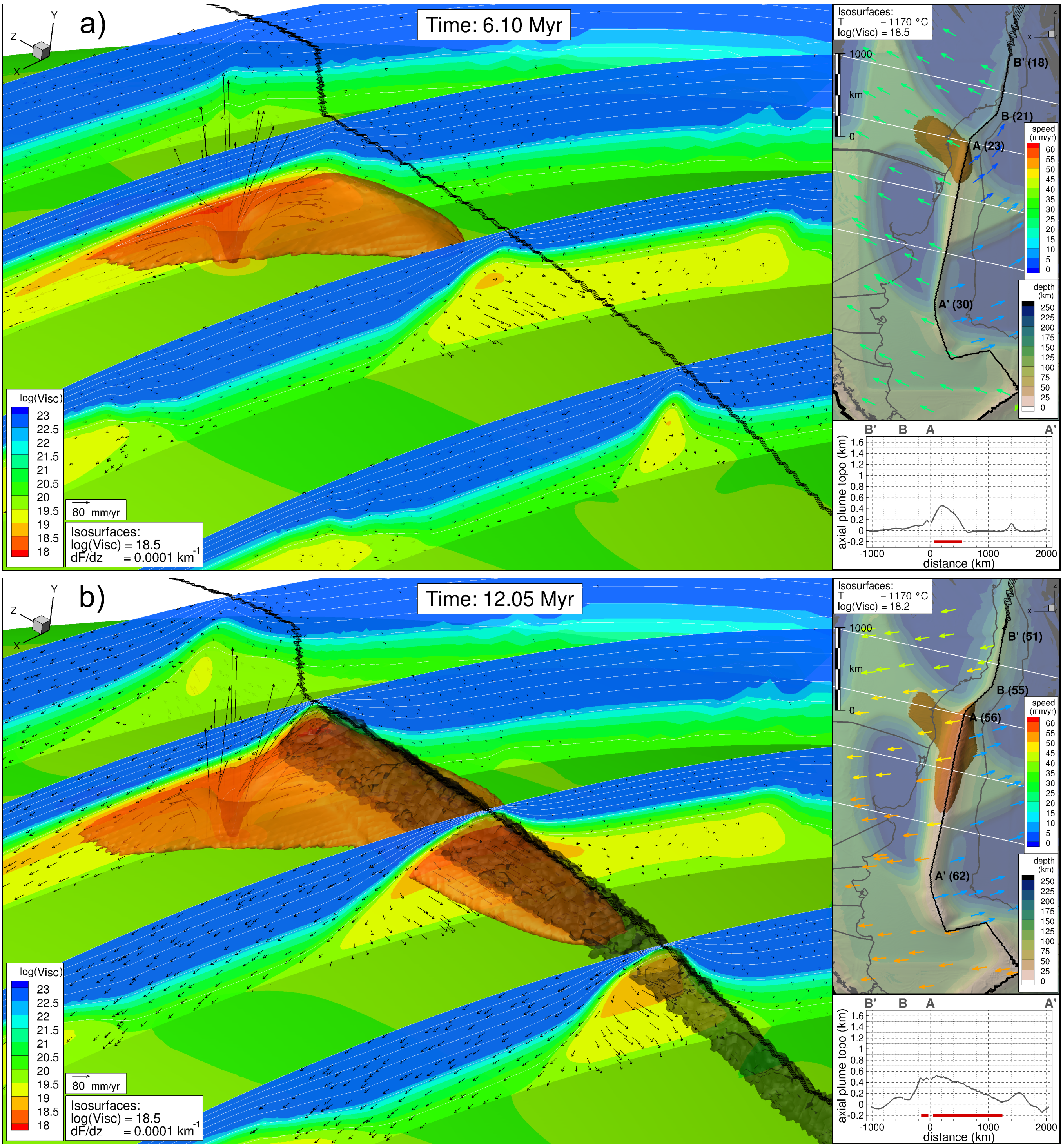


FIGURE 8 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1F15M.

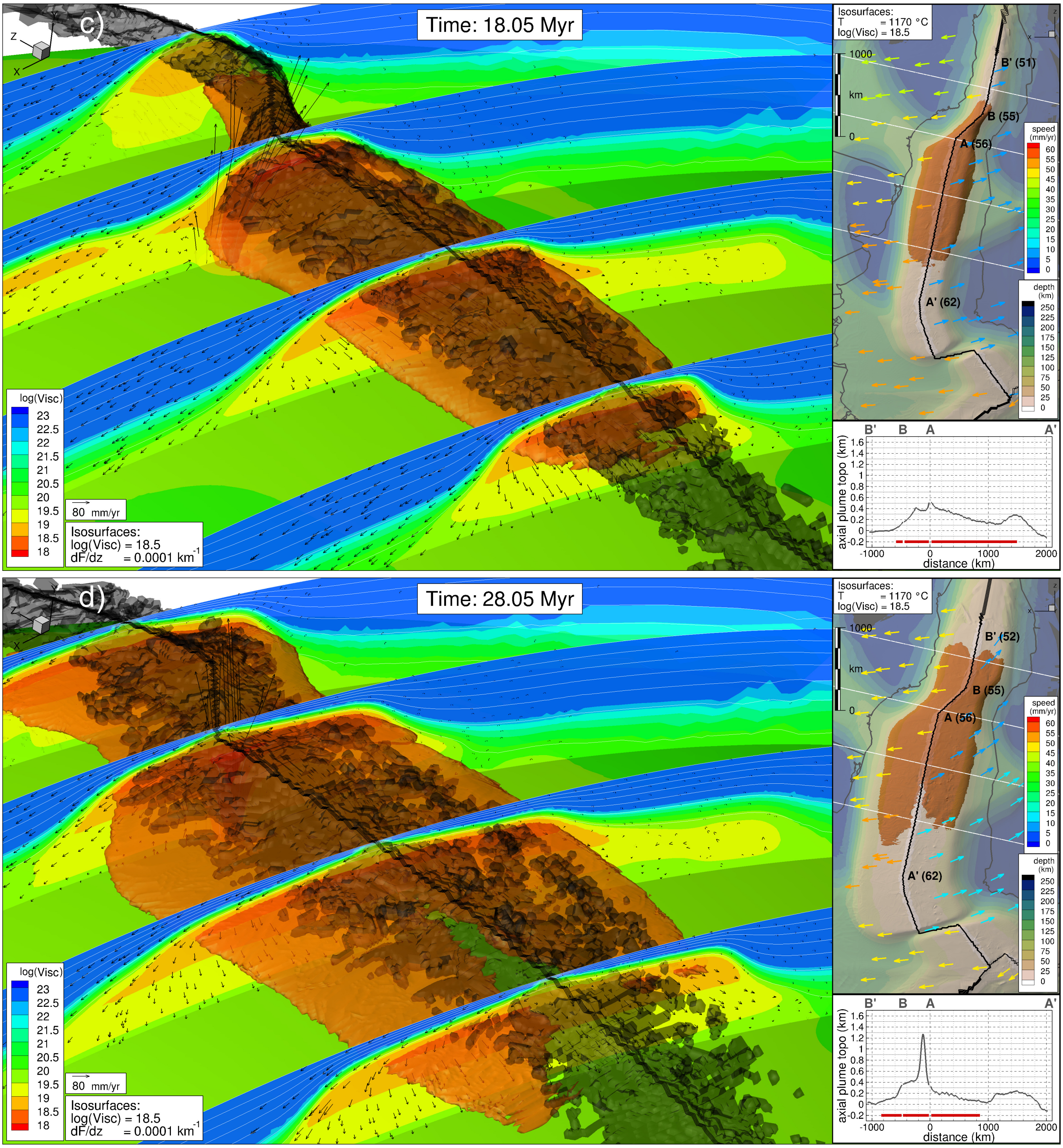


FIGURE 8 (Cont.) 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1F15M. Main: 3-D images showing the geometry of the plume material at different times: (a) 6.10 Myr, (b) 12.05 Myr, (c) 18.05 Myr and (d) 28.05 Myr. Colour represents logarithm of viscosity. The velocity field and isotherms every $200\,^{\circ}C$ in the vertical cross sections are represented by arrows and white lines, respectively. Black line represents the plate boundary. The plume material is represented by the red isosurface with $\log(\eta) = 18.5$. The melting rate per km upflow is represented by the black isosurface with $dF/dz = 0.0001\text{km}^{-1}$. Top inset: Top view of the 3-D evolution shown in the main inset. The isosurface with a temperature of $1170\,^{\circ}C$ is coloured with depth to show the lithospheric thickness variations. The plume material is represented by the red isosurface with $\log(\eta) = 18.5$. Colour arrows represent the top surface plate motion. Grey lines and black thick lines represent the reconstructed coastlines and the plate boundaries, respectively. Capital letters show the ends of the along ridge profiles shown in bottom inset. Numbers between parentheses show the full opening speed in mm/yr. Bottom inset: Plume contribution to axial topography in km. The red horizontal line represents the plume material beneath the ridge profile.

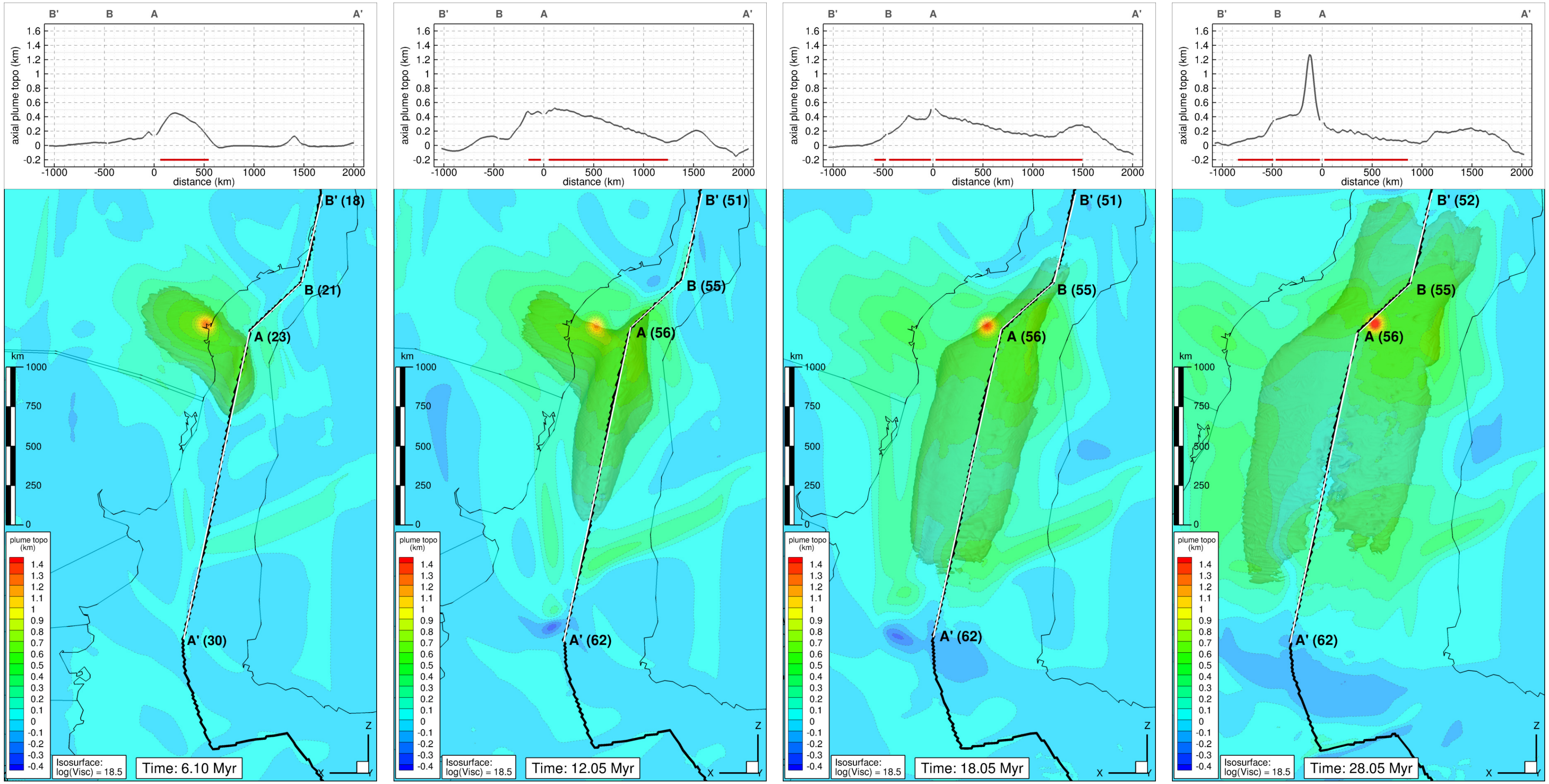


FIGURE 9 Evolution of the plume contribution to the topography for model L1F15M. Colour represents the plume contribution to the topography in km. The plume material is represented by a isosurface with $\log(\eta) = 18.5$. Black thin and thick lines represent the reconstructed coastlines and the plate boundaries, respectively. Capital letters show the ends of the along ridge profiles shown in bottom inset. Numbers between parentheses show the full openning spped in mm/yr. Top inset: Plume contribution to axial topography in km. The red horizontal line represents the plume material beneath the ridge profile.

Models L1F10 (see supporting information Figure S1, Figure S2, Movie S5 and movie S6) and L1F20 (see supporting information Figure S3, Figure S4, Movie S7 and movie S8) show the plume's influence beneath the rifting region when the only parameter changed with respect to the model L1F15 is the plume flux (See Table 1). The behaviour of the plume material is similar to the one observed in models L1F15 and L1F15M. The results for model L1F20 show plume material reaching further along the rifting region (firstly southwards and northwards towards the end of the simulation). On the other hand, model L1F10 shows less influence of the plume material in the rifting region.

Models where plume is located at L1 present similar behaviour of the plume material except for its extension along the ridge. We observe an increase of the southward plume material speed that is correlated with the abrupt increase in opening rate between 8 Myr and 12 Myr (Figure 5b) (The mean southward plume material speed is calculated measuring the travelled distance by the tip of the plume material for each 2 Myr interval during its southward motion). This is important, as it suggests that the 'suction' from the spreading region is the primary force that 'pulls' plume material towards the South. The southward plume material speed is directly proportional to the plume flux (compare purple line (L1F10), blue line (L1F15) and black line (L1F20) in Figure 5b). Model L1F15M (orange line in Figure 5b) presents the same initial southward plume material speed than model L1F15, although the isosurfaces for plume material differ by $\log(\eta) = 0.3$. Once melting starts, the southward plume material speed slows considerably due to the three-fold increase in viscosity.

Model L1bF15 presents a plume located beneath Congo craton close to its south-western edge. Figure 10 and supporting information Movie S9 show the 3-D evolution of the model L1bF15. During the first 4 Myr of simulation the plume material is impinging the base of the craton. At 6 Myr the plume material starts to drain towards the rifting region (Figure 10a). Once the plume material has reached the rifting region it starts to flow southwards (Figure 10b). The plume material reaches it maximum southward extent at 18 Myr (Figure 10c) and its region of influence shrinks towards the end of the simulation (Figure 10d). Figure 11 and supporting information Movie S10 show the evolution of the plume contribution to topography for model L1bF15.

Model L2F15 presents a plume located in the northern edge of the Rio de la Plata craton. Figure 12 and supporting information Movie S11 show the 3-D evolution of the model L2F15. When the plume material impinges the edge of the craton, it flows towards regions of thinner lithosphere and towards the rifting region (Figure 12a). Afterwards, it starts to migrate southwards (Figure 12b). At 18 Myr the plume material reaches it maximum southward influence (Figure 12c) and again, towards the end of the simulation, the region of plume influence shrinks (Figure 12d). Figure 13 and supporting information Movie S12 show the evolution of the plume contribution to the topography for model L2F15.

Model L2F10 (see supporting information Figure S5, Figure S6, Movie S13 and Movie S14) shows the plume's influence beneath the rifting region when the only parameter varied with respect to the model L2F15 is the plume flux (See Table 1). The behaviour of the plume material is similar to the one observed in models where the plume is located at L1 and model L2F15. However, the plume material does not reach the same southward distance as the plume material in model L1F10 (compare Figure S5c and Figure S2c) probably to the initial position of the plume and the initial plume flux. At the beginning of the simulation, the plume material needs around 4 Myr to reach a similar position to the initial position of the plume in the model L1F10. This delay produces that the plume material arrives to the rifting region when the abrupt acceleration of the southern part is too far to influence a rapid southward migration (purple dash line in Figure 5b) like observed for models where plume is located at L1 and for model L2F15. The southern end of the rifting region is then filled by upwelling mantle material instead of plume material. The plume contribution to the topography is lower in the model L2F10 than in the model L1F10. All models where plume location is at L1 and L2 experiment a considerably reduction of southward plume material speed once the North passage between São Francisco and Congo cratons opens (vertical dot line in Figure 5b).

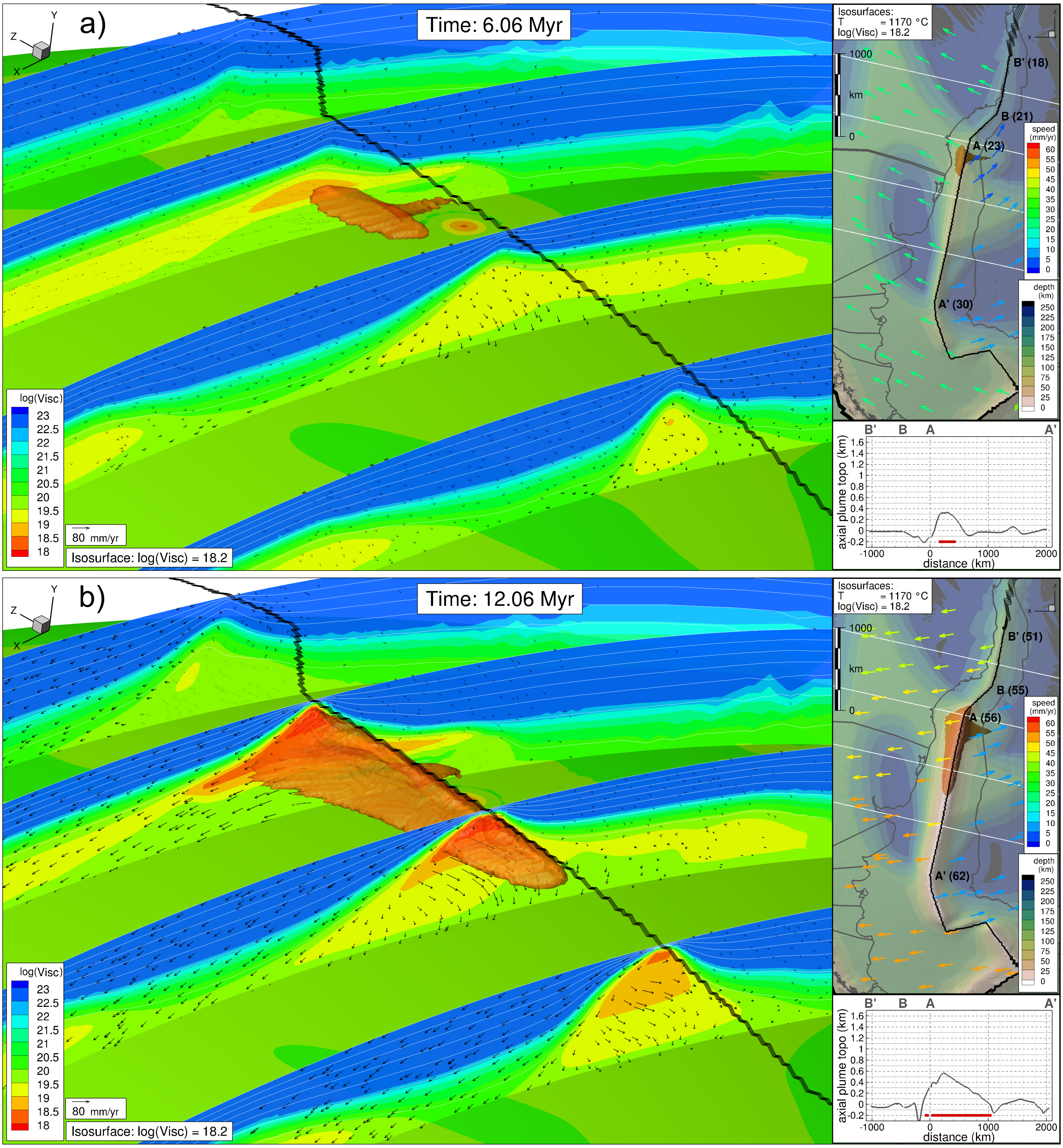


FIGURE 10 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1bF15.

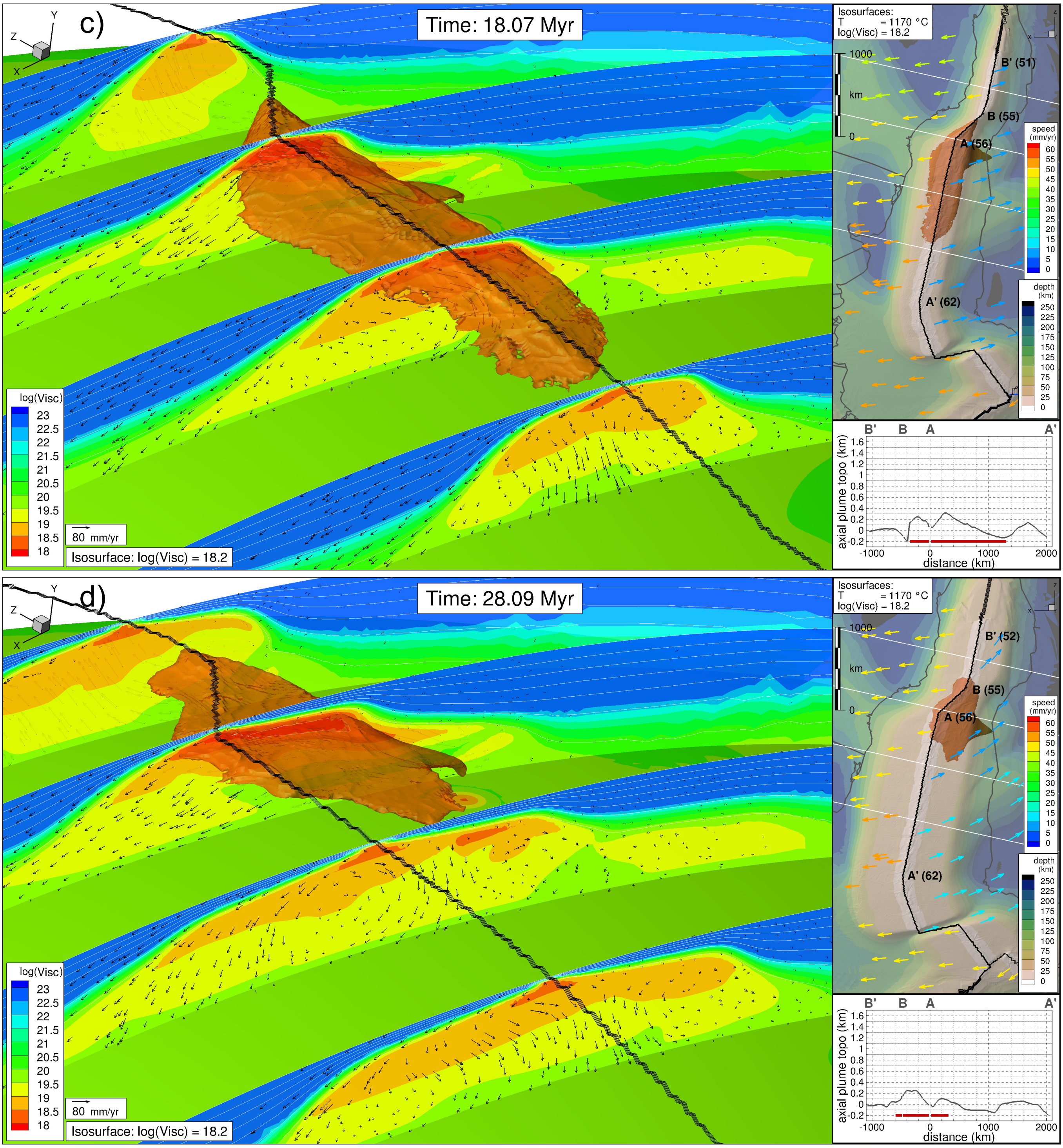


FIGURE 10 (Cont.) 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1bF15. Main: 3-D images showing the geometry of the plume material at different times: (a) 6.06 Myr, (b) 12.06 Myr, (c) 18.07 Myr and (d) 28.09 Myr. Same representation as in Figure 6.

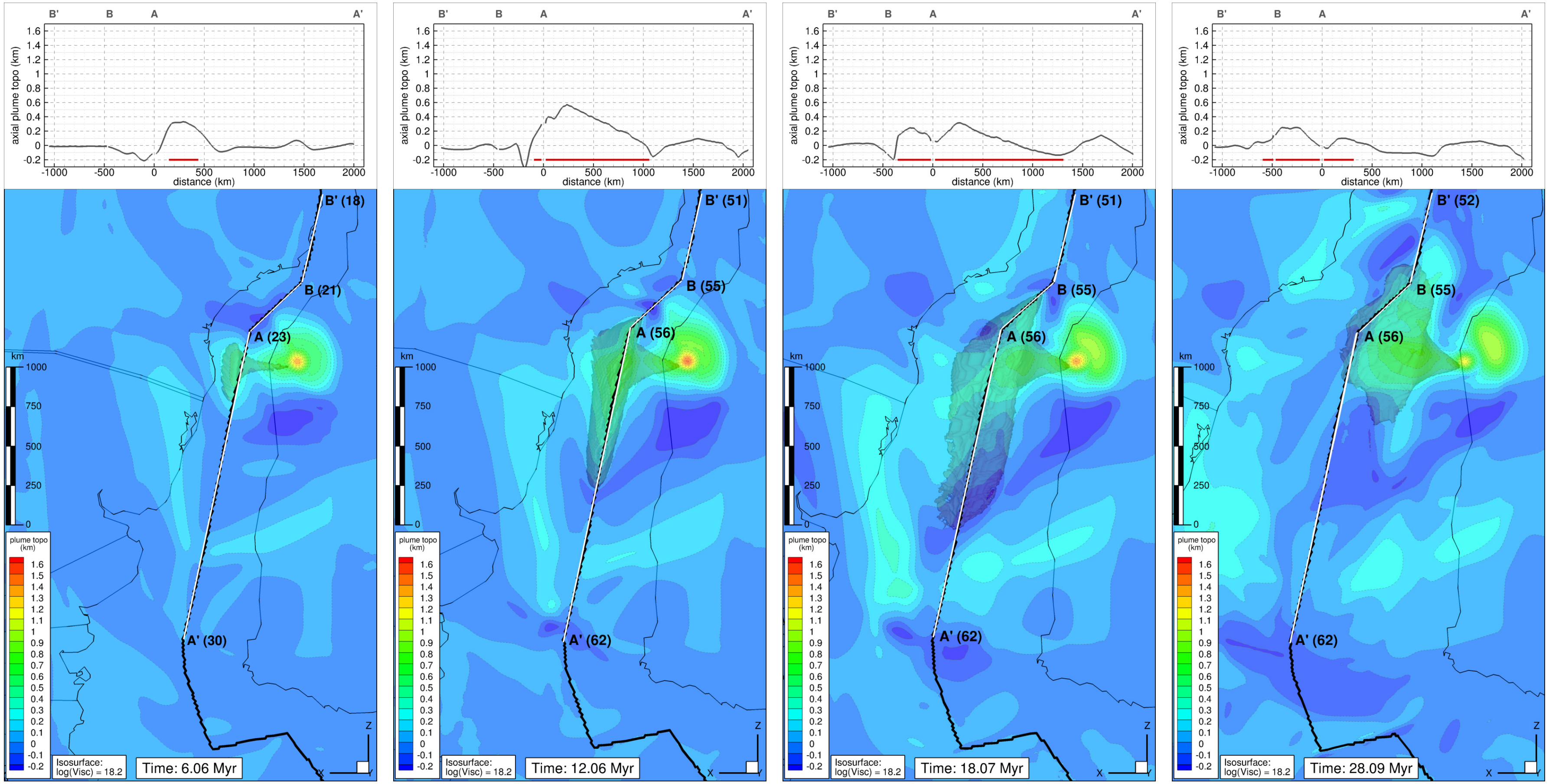


FIGURE 11 Evolution of the plume contribution to the topography for model L1bF15. Same representation as in Figure 7

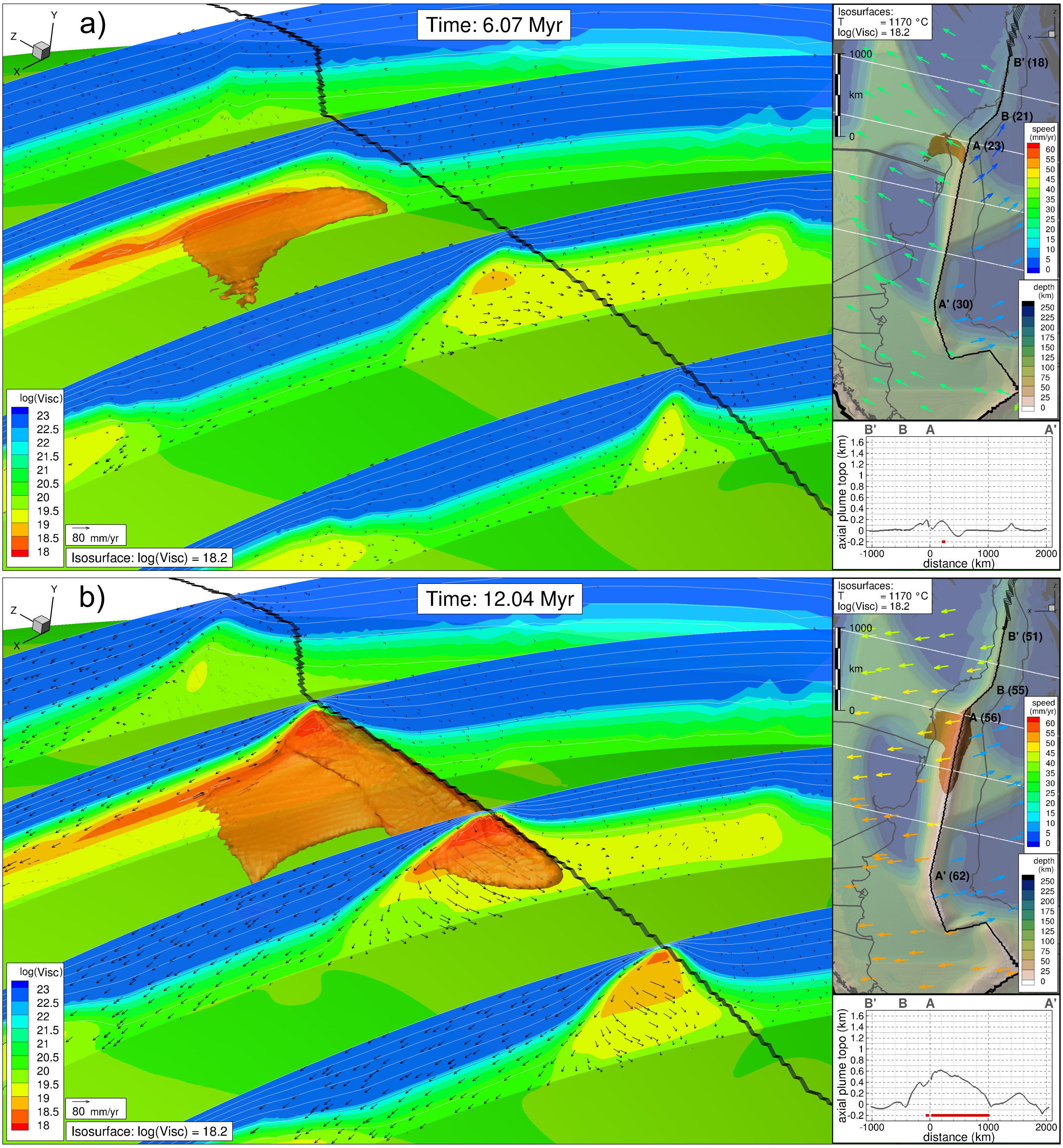


FIGURE 12 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L2F15.

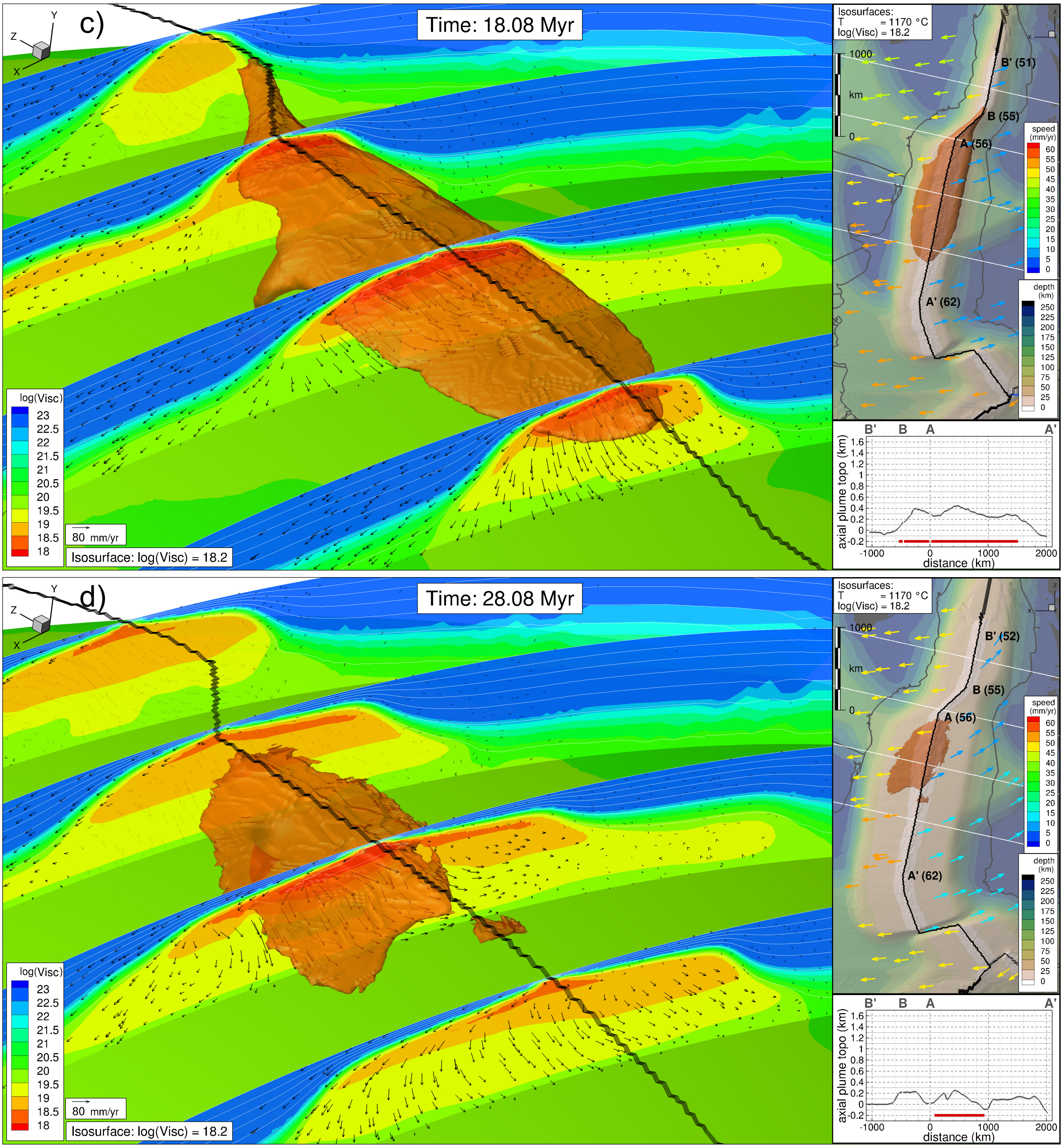


FIGURE 12 (Cont.) 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L2F15. Main: 3-D images showing the geometry of the plume material at different times: (a) 6.07 Myr, (b) 12.04 Myr, (c) 18.08 Myr and (d) 28.08 Myr. Same representation as in Figure 6.

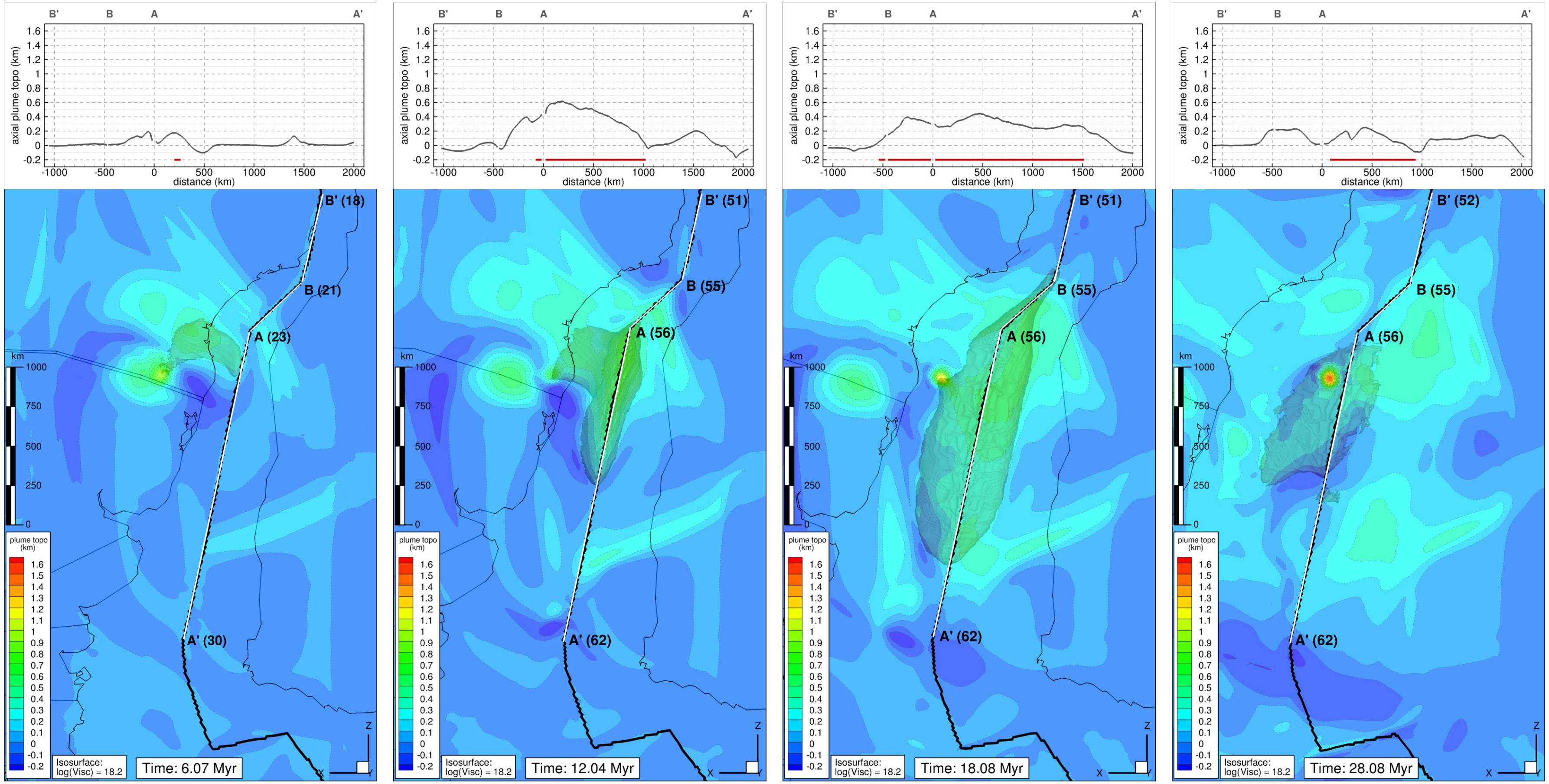


FIGURE 13 Evolution of the plume contribution to the topography for model L2F15. Same representation as in Figure 7

3.2 | Other lateral plume-flow scenarios

We have explored other possible plume flow scenarios by moving the plume location further southwards. Model L3F15 presents a plume beneath the Rio de la Plata craton close to its north-eastern edge. Figure 14 and supporting information Movie S15 show the 3-D evolution of the model L3F15. During the first 6 Myr of simulation, the model L3F15 shows a plume head flow scenario in which the plume material is impinging the base of the craton (top inset in Figure 14a). Around 8 Myr, the westward motion of the South America plate over the plume opens a new path for the plume material, which starts to drain towards the ridge (Figures 14b and 14c). Once plume material reaches the ridge, it starts to flow along the ridge with no preferred direction. Towards the end of the simulation, the region of the plume influence shinks due to faster opening rate (Figure 14d). Figure 15 and supporting information Movie S16 show the evolution of the plume contribution to topography for the model L3F15. The thermal uplift of the plume beneath the craton is clearly visible along the simulation.

Model L4F7.5 (see supporting information Figure S7, Figure S8, Movie S17 and movie S18) presents a plume beneath the western part of the Rio de la Plata craton. During the first 14 Myr of simulation, the model L4F7.5 shows a plume head flow scenario with the plume material impinging the base of the craton and creating a dome. At 18 Myr, the plume material starts to drain towards the ridge.

4 | DISCUSSION

The lateral southward plume flow migration observed in the models where plume is located at L1, L1b and L2 is potentially due to a combination of two main factors: (1) a 'suction' from regions of stretching in the South and (2) the presence of thicker São Francisco and conjugate Congo cratonic roots in the North. The southward plume material speed-up observed in models where plume is located at L1 suggest that the 'suction' from the spreading region is the dominant factor that 'pulls' plume material towards the South.

Although we cannot compare directly the models L1F15 and L1F15M since melting consumes energy leading to a decrease on temperature of the plume material, the results obtained in both models are quite similar, with a southward migration of the plume material. However, the southward migration of the plume material in model L1F15M is limited by the increase of viscosity due to the cooling from the latent heat of melting.

Models where plume is located at L1 show a correlation between a southward speed-up of plume material and abrupt acceleration in the opening speed (Figure 5). In general, this behaviour is not observed for models where the plume is at different locations. It appears that for models with the plume located at L1, the plume material has enough time to accumulate in the rifting region and start to migrate southwards before the abrupt acceleration in the opening speed occurs. When the opening speed increases dramatically, plume material migrates southwards faster than the mantle upwelling below the southern part of the ridge. This provides a possible alternative explanation for the observation of a plate acceleration before the break-up [*Brune et al.*, 2016]. In this case, the widespread development of SDRs is linked to rapid southward extrusion of plume material, with the SDR volcanism contributing to weakened lithosphere and enhanced ridge push over a wide section of ridge that leads to breakup. On the other hand, plume material in model L2F10 arrives with aproximately 4 Myr of delay to the rifting region when begins the abrupt plate acceleration. Plume material starts to flow southwards but it does not speed-up. This could be due to that mantle upwelling in the southern part inhibits the rapid southward motion of the plume material, or that the 'pond' of available mobile plume material is too small to be pulled southwards. Future models with a plume at L1 and a temporal delay are needed to make a better comparison.

Our models do not account for power law creep. However, if the rheology would consist of a composite viscosity including both dislocation and diffusion creep mechanisms it would decrease the viscosity of the upper part of the mantle in the stretching and plume regions. This would lead to models where plume material would flow faster than in our models, which could be more realistic. An additional simplification of the models is that a composition-dependent rheology is not included. Including a composition-dependent rheology would also tend to create a more viscous asthenosphere, at depths where water easily partitions into a melt.

The initial structure of the base of the craton above the plume in models L3F15 and L4F7.5 seems to play an important role in the initial evolution of the plume material by inhibiting its drainage to thinner lithosphere regions. A caveat of the numerical implementation of the cratons is that their base is exactly flat, and does not include a small regional tilt. This may cause plume material to accumulate at the base of the craton especially when plate motions are slow. Making the base of the craton slightly tilted would allow the plume material to more easily drain (cratons could have a more realistic initial shape by smoothing their lower boundaries by, for example, applying diffusion for tens of millions of years before the start of the experiment).

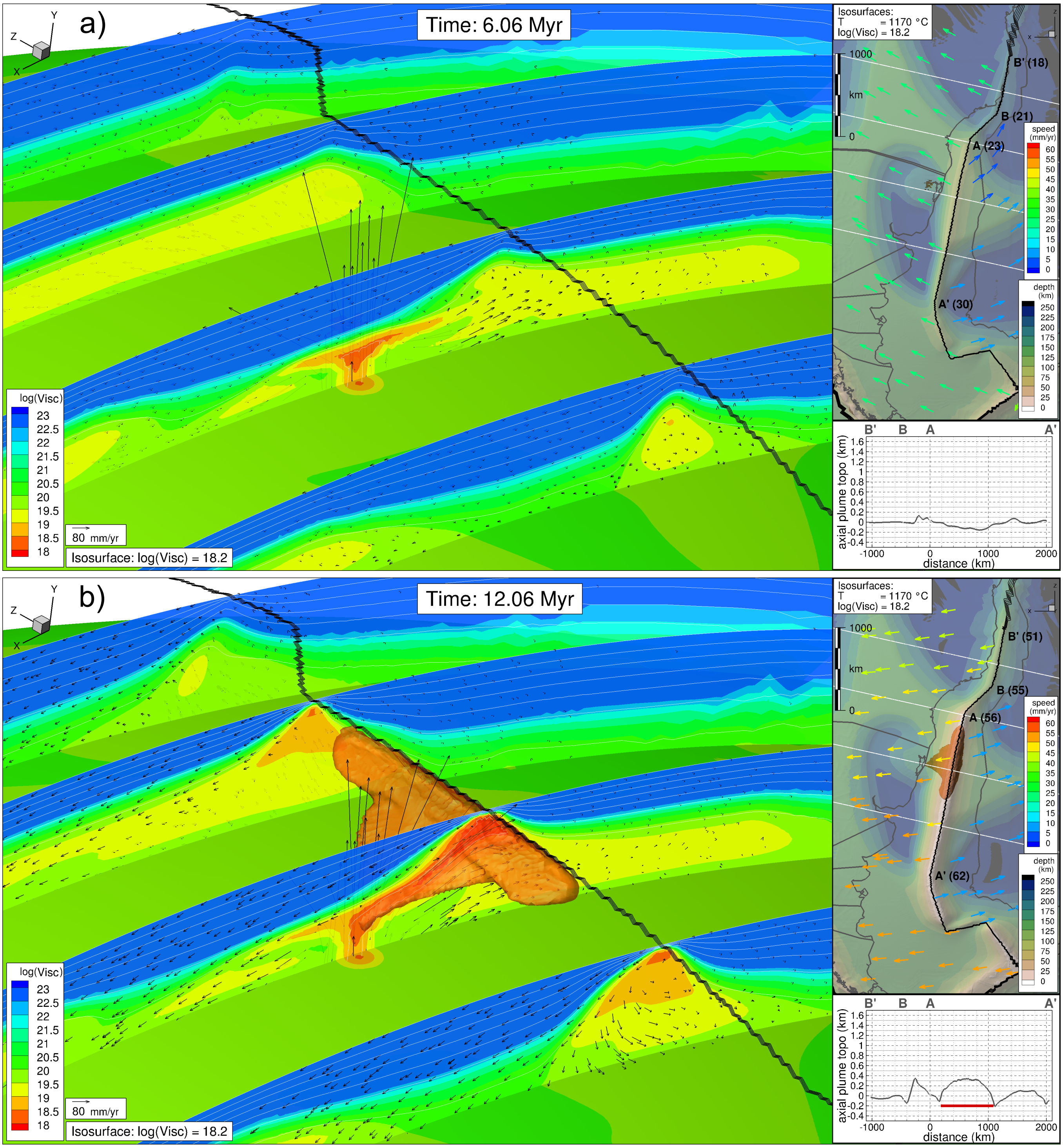


FIGURE 14 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L3F15.

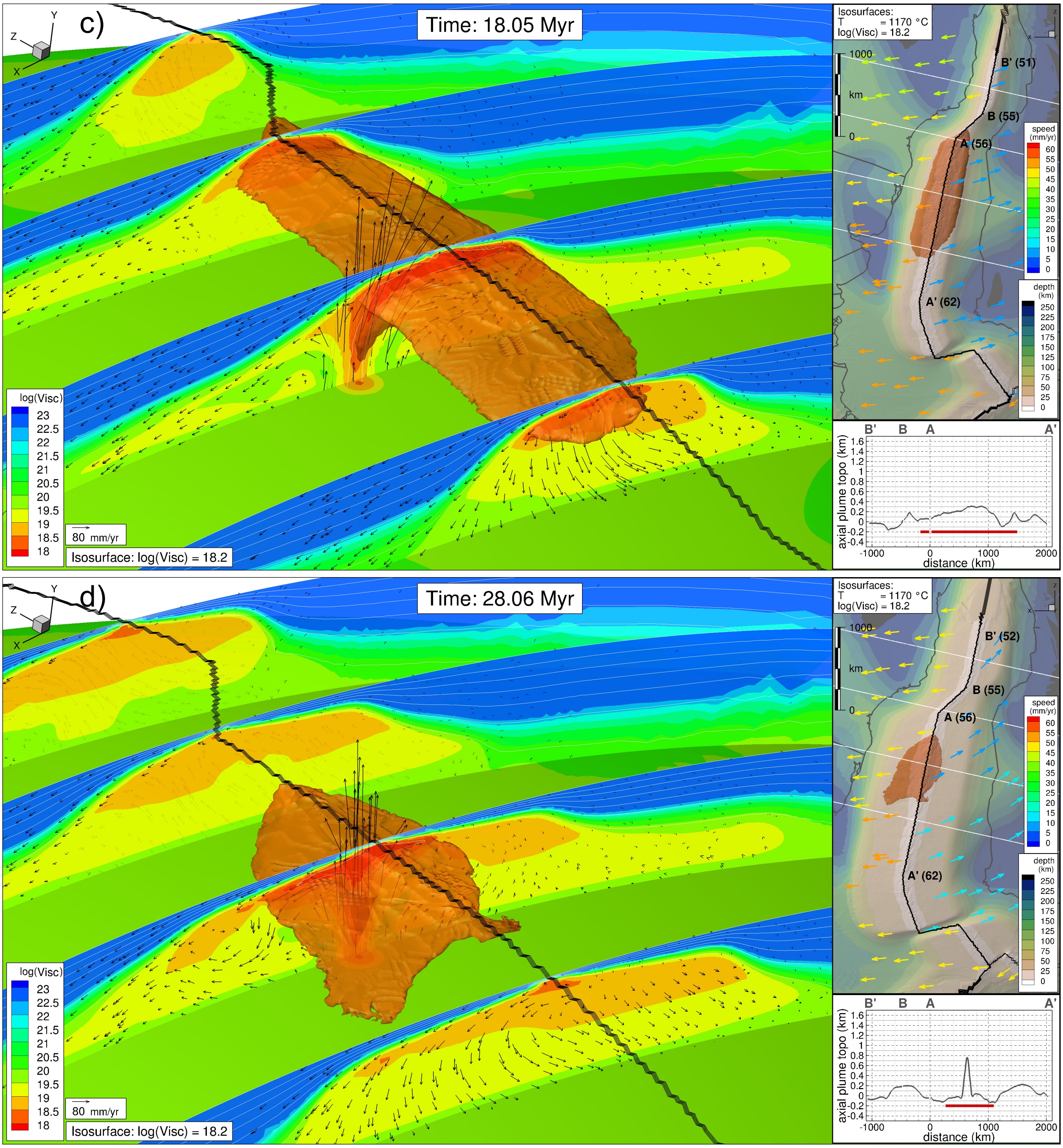


FIGURE 14 (Cont.) 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L3F15. Main: 3-D images showing the geometry of the plume material at different times: (a) 6.06 Myr, (b) 12.06 Myr, (c) 18.05 Myr and (d) 28.06 Myr. Same representation as in Figure 6.

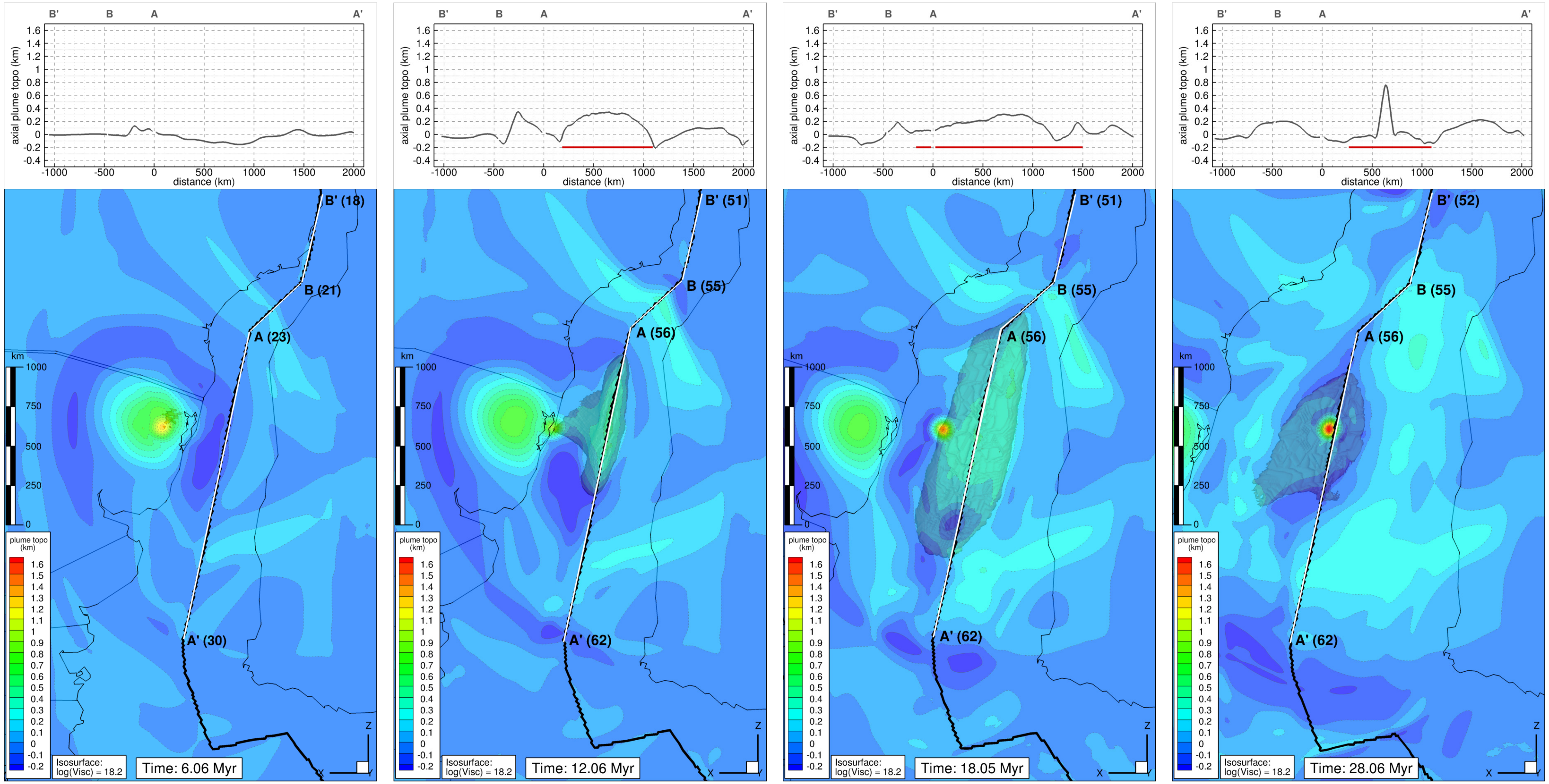


FIGURE 15 Evolution of the plume contribution to the topography for model L3F15. Same representation as in Figure 7.

4.1 | Implications for SDRs observations

Seward Dipping Reflectors (SDRs) are an accepted indicator for volcanic rifted margins. In the South Atlantic, they have been observed offshore South of Brasil, Uruguay and the central-north of Argentina in the South America margin and offshore Namibia and South Africa in the Africa margin. The distribution of the SDRs in the South Atlantic shows an asymmetry that is not observed in some other volcanic rifted margins with plume influence, for example in Greenland where there is a symmetric distribution of the volcanic rifted margins with respect to the plume position (classic plume head flow scenario). Our models predict a scenario in which plume material preferentially migrates southwards (Figure 6). This migration would lead to the emplacement of offshore thick wedges of volcanic material (SDRs) along the southern part of the South Atlantic, in agreement with the observed distribution of SDRs.

5 | CONCLUSIONS

We have studied the influence of Tristan da Cunha plume and initial lithosphere thickness in the early rifting evolution of the South Atlantic is a series of 3-D numerical models. Models consistent with the location of the Tristan plume track where plume is located at L1 and L2 show a lateral southward plume flow migration with a maximum extent proportional to the plume flux. This preferential southward flow is mainly due to a combination of suction associated with stretching in the South and the presence of thicker São Francisco and conjugate Congo cratonic roots in the North. In these models, plume material experiences a southward speed-up that is related to the abrupt acceleration in the opening speed. The duration of SDR emplacement is predicted to take 5-10 Myr. This rapid southward migration of the plume material could explain the formation of the observed rifting-related SDRs along the South Atlantic margins.

SUPPORTING INFORMATION

The following supporting information is available as part of the online article:

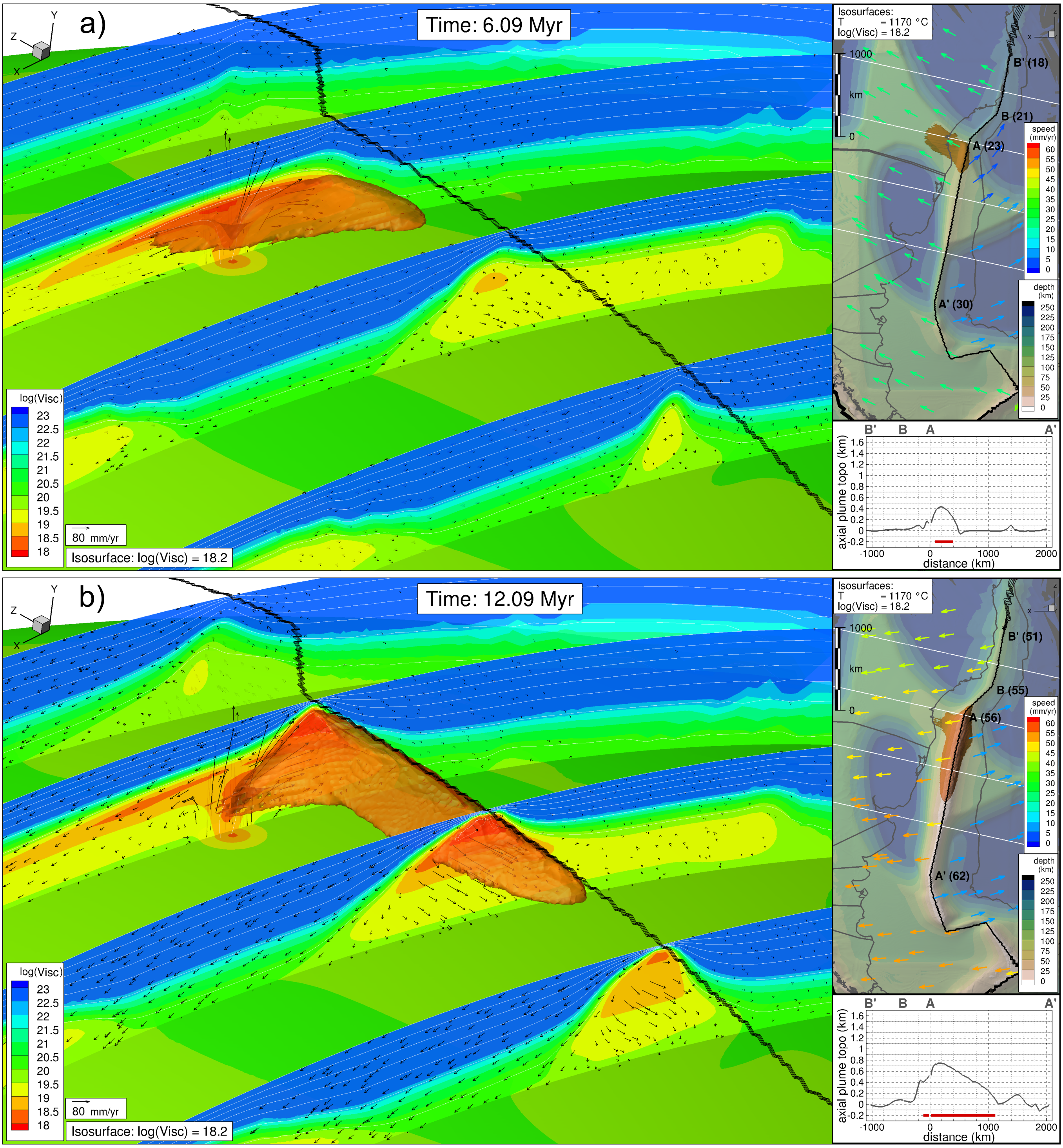


FIGURE S1 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1F10.

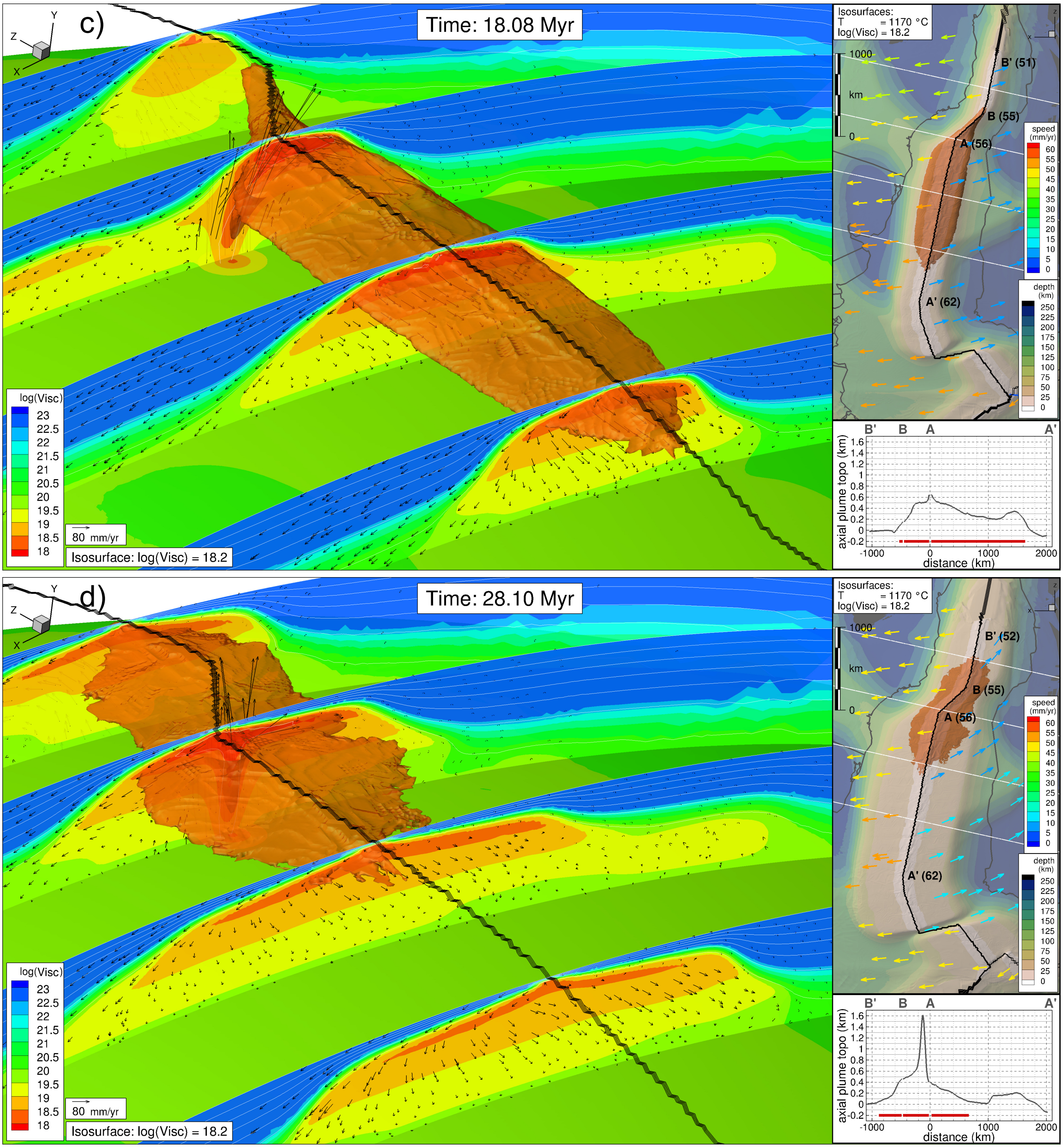


FIGURE S1 (Cont.) 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1F10. Main: 3-D images showing the geometry of the plume material at different times: (a) 6.09 Myr, (b) 12.09 Myr, (c) 18.08 Myr and (d) 28.10 Myr. Same representation as in Figure 6.

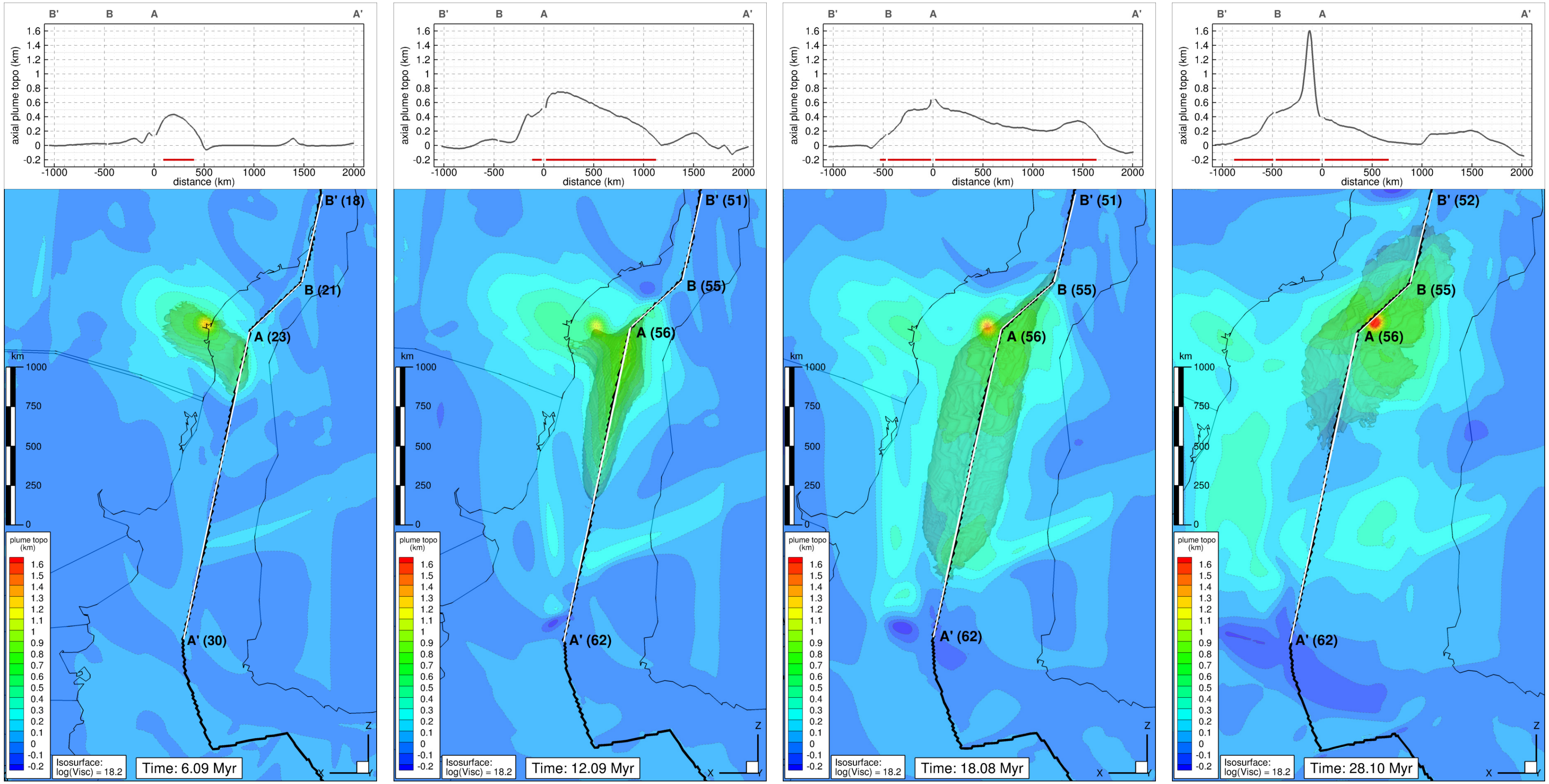


FIGURE S2 Evolution of the plume contribution to the topography for model L1F10. Same representation as in Figure 7.

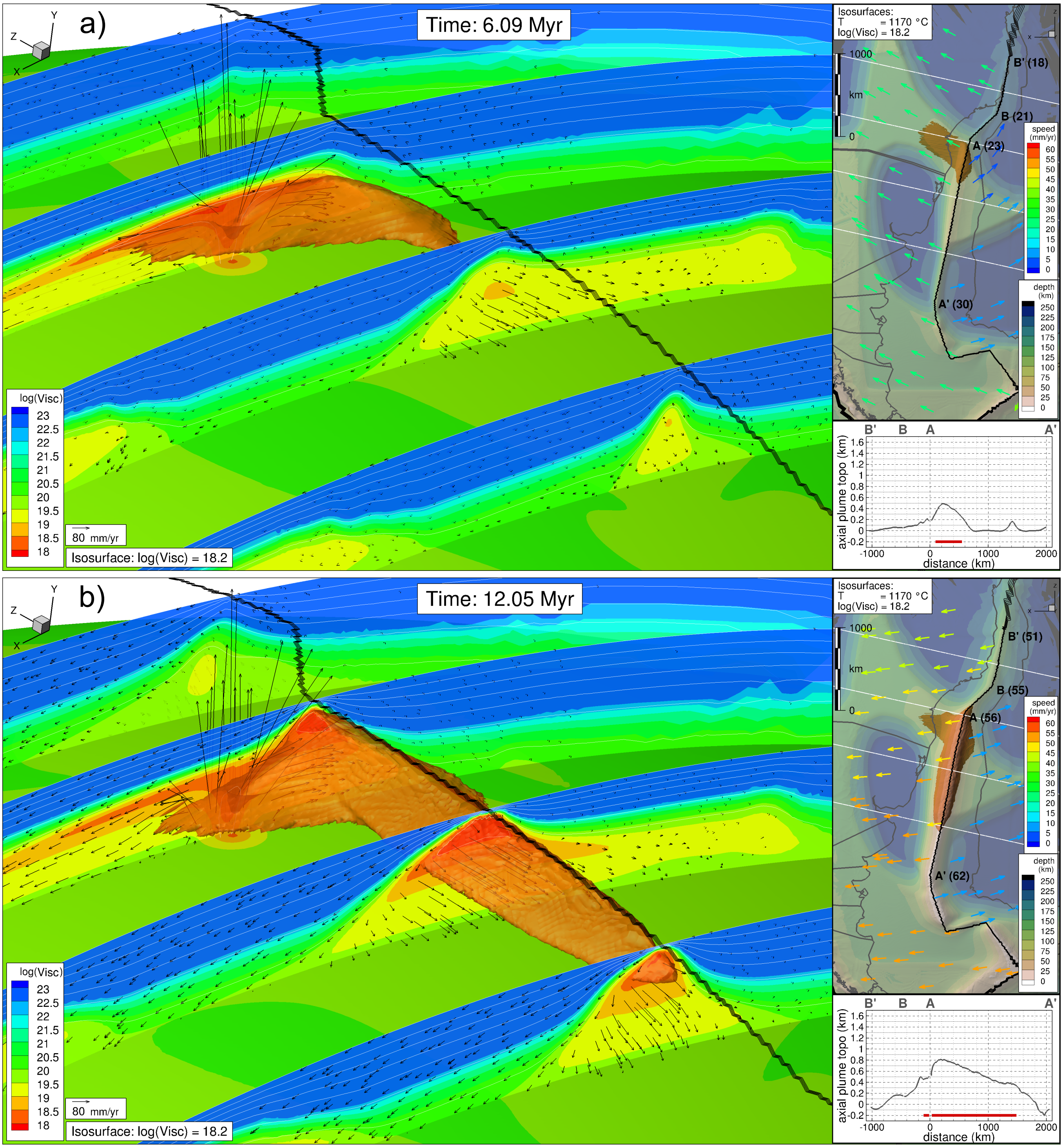


FIGURE S3 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1F20.

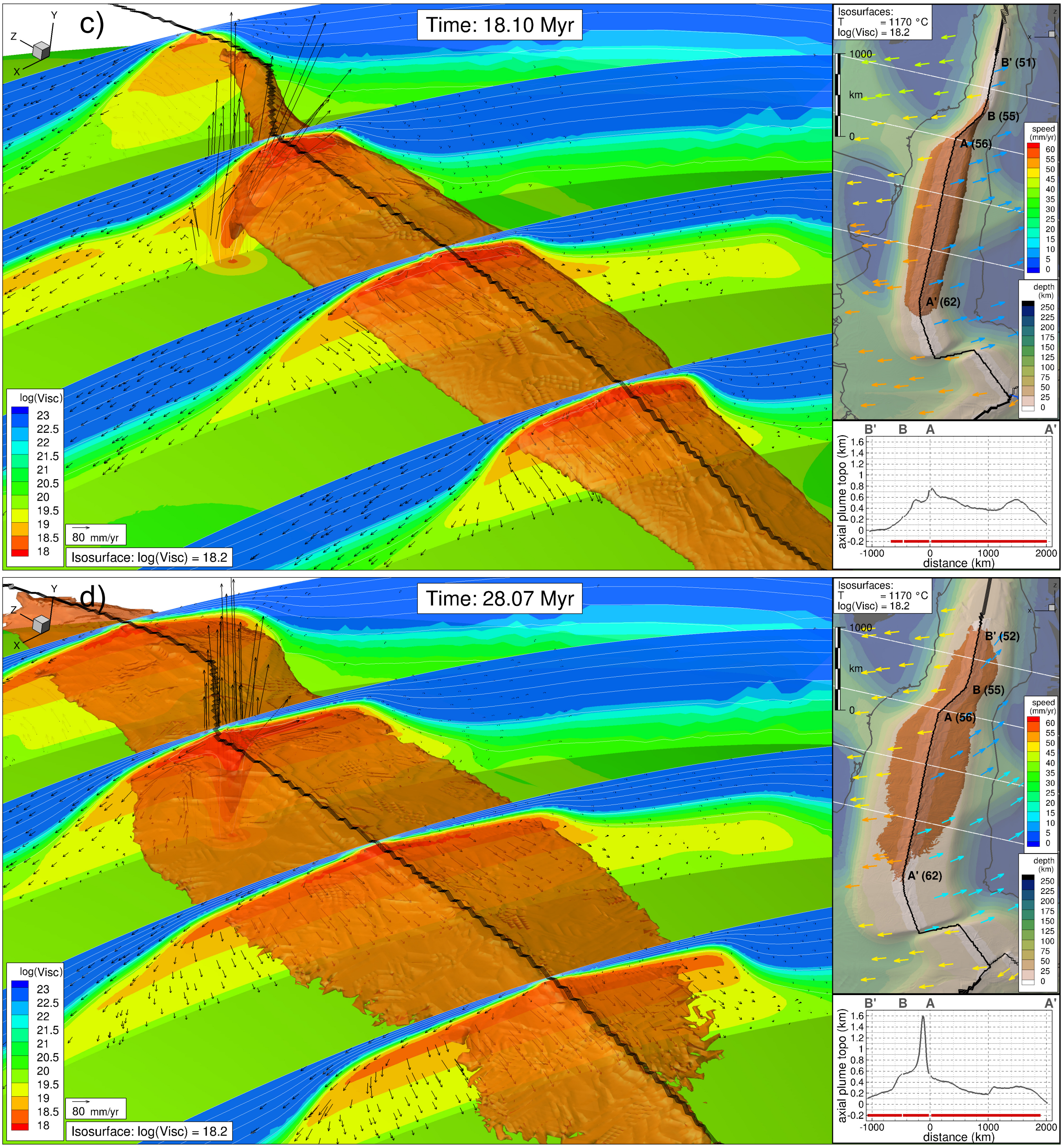


FIGURE S3 (Cont.) 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L1F20. Main: 3-D images showing the geometry of the plume material at different times: (a) 6.09 Myr, (b) 12.05 Myr, (c) 18.10 Myr and (d) 28.07 Myr. Same representation as in Figure 6.

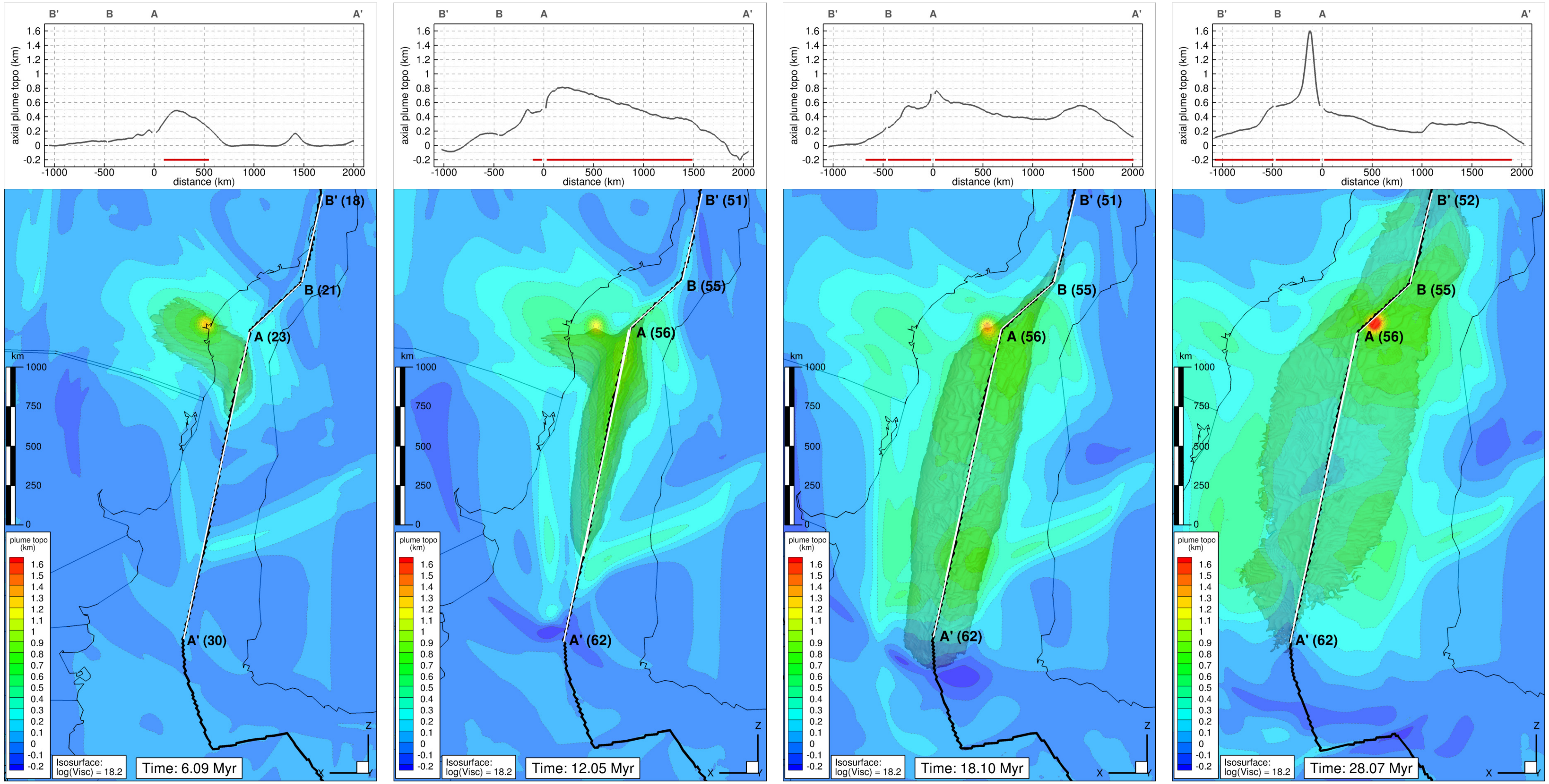


FIGURE S4 Evolution of the plume contribution to the topography for model L1F20. Same representation as in Figure 7.

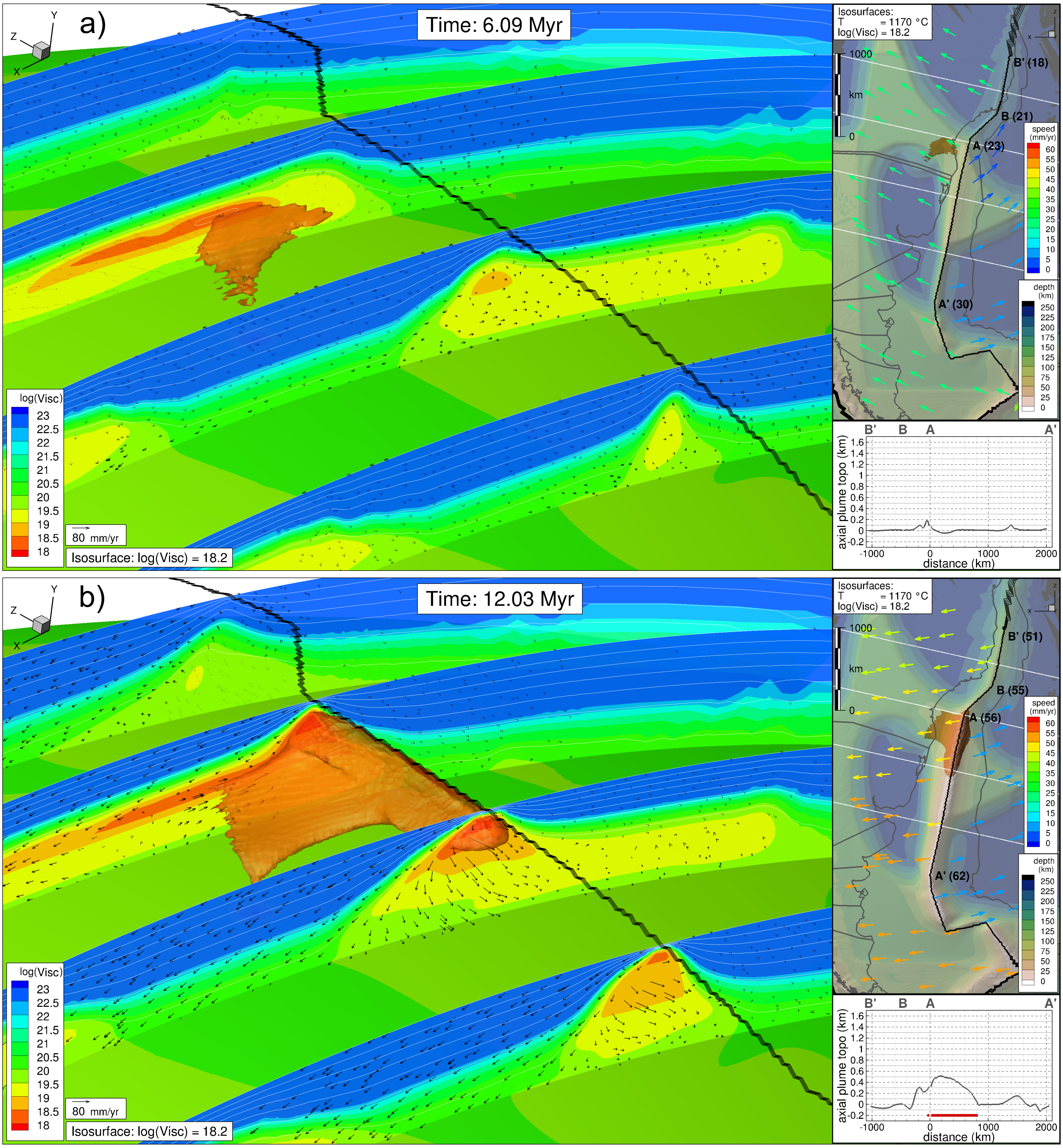


FIGURE S5 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L2F10.

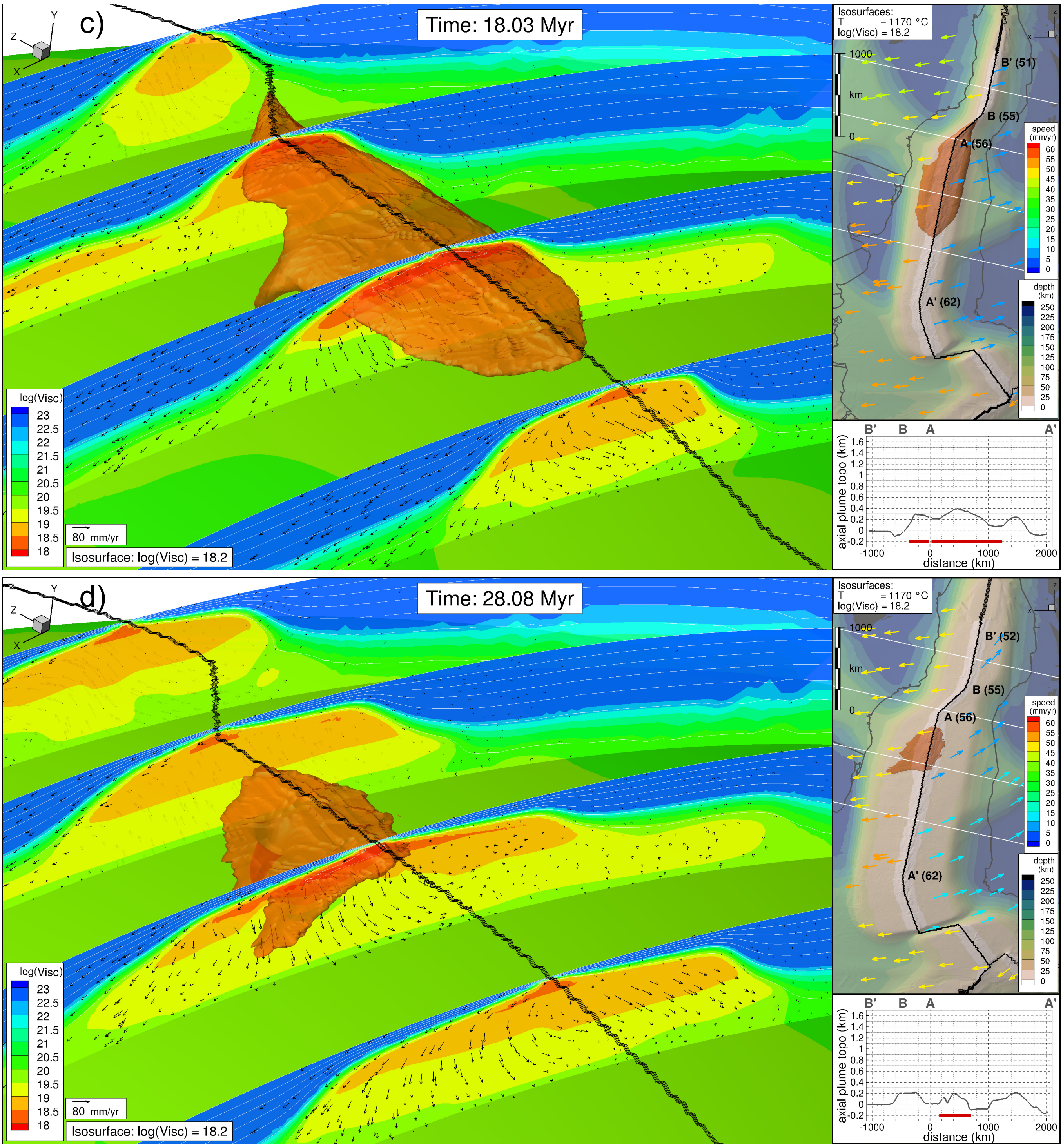


FIGURE 5 (Cont.) 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L2F10. Main: 3-D images showing the geometry of the plume material at different times: (a) 6.09 Myr, (b) 12.03 Myr, (c) 18.03 Myr and (d) 28.08 Myr. Same representation as in Figure 6.

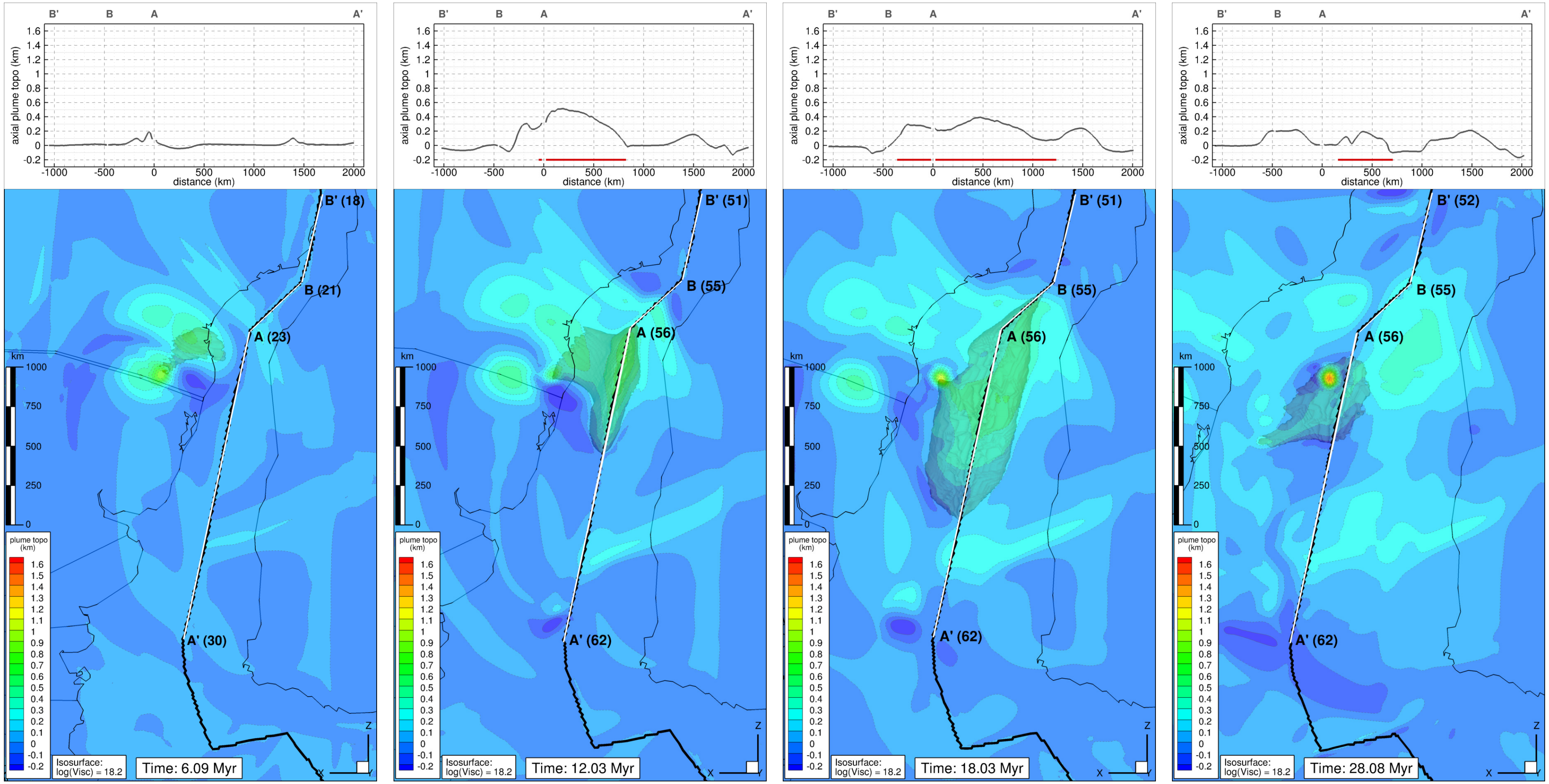


FIGURE 6 Evolution of the plume contribution to the topography for model L2F10. Same representation as in Figure 7

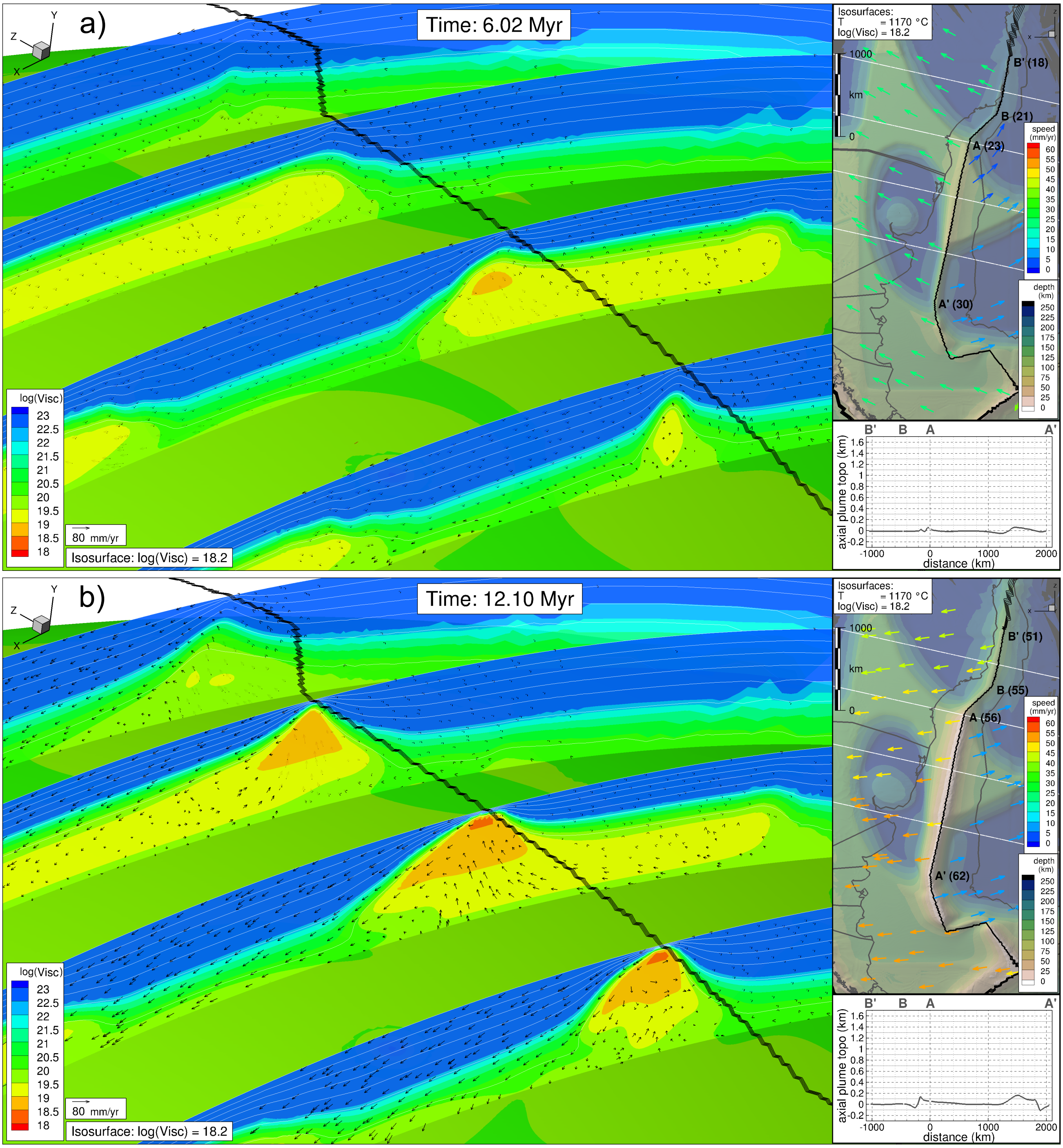


FIGURE S7 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L4F7.5.

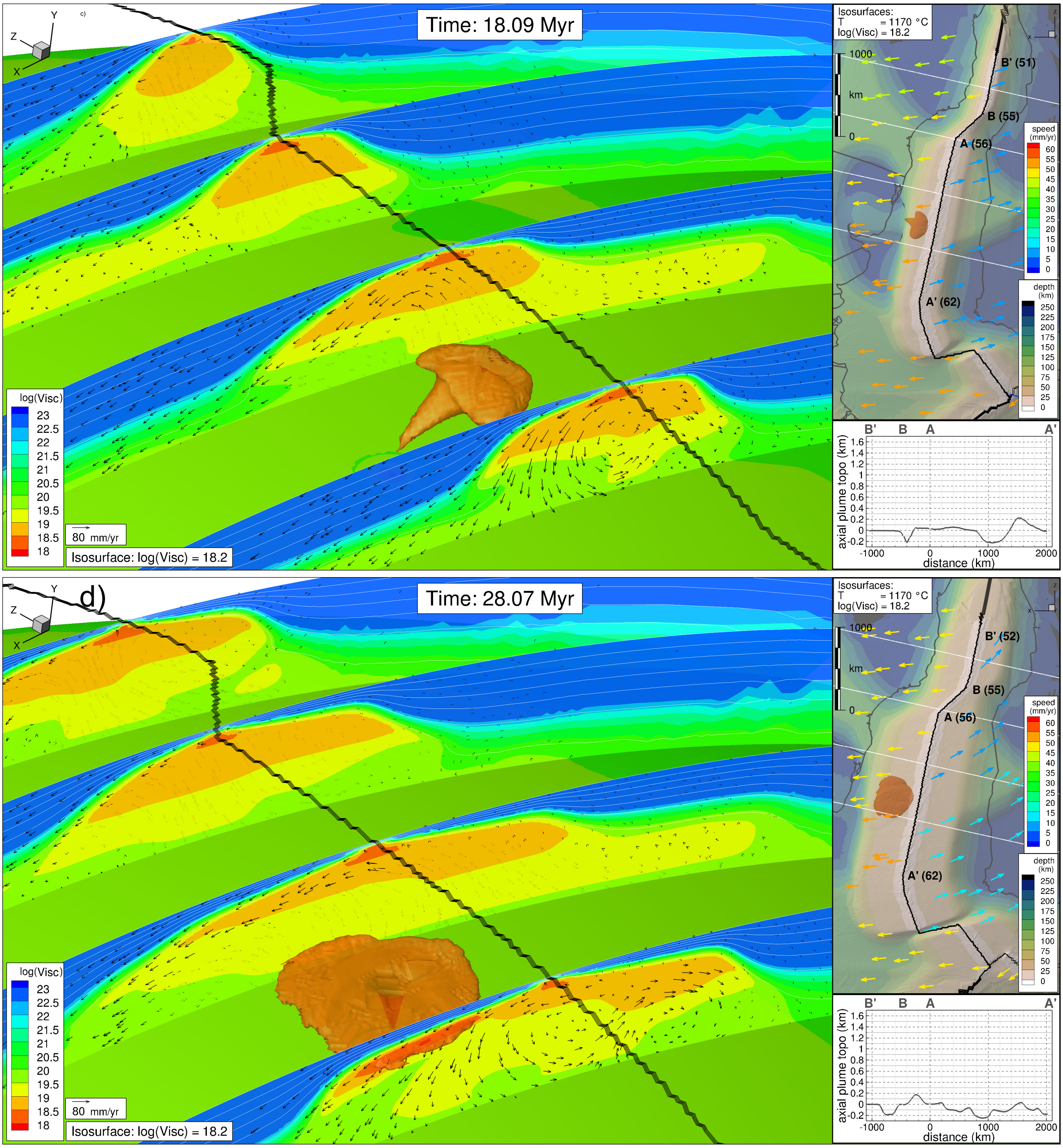


FIGURE S7 (Cont.) 3-D evolution of plume material during rifting and break-up of the South Atlantic for model L4F7.5. Main: 3-D images showing the geometry of the plume material at different times: (a) 6.02 Myr, (b) 12.10 Myr, (c) 18.09 Myr and (d) 28.07 Myr. Same representation as in Figure 6.

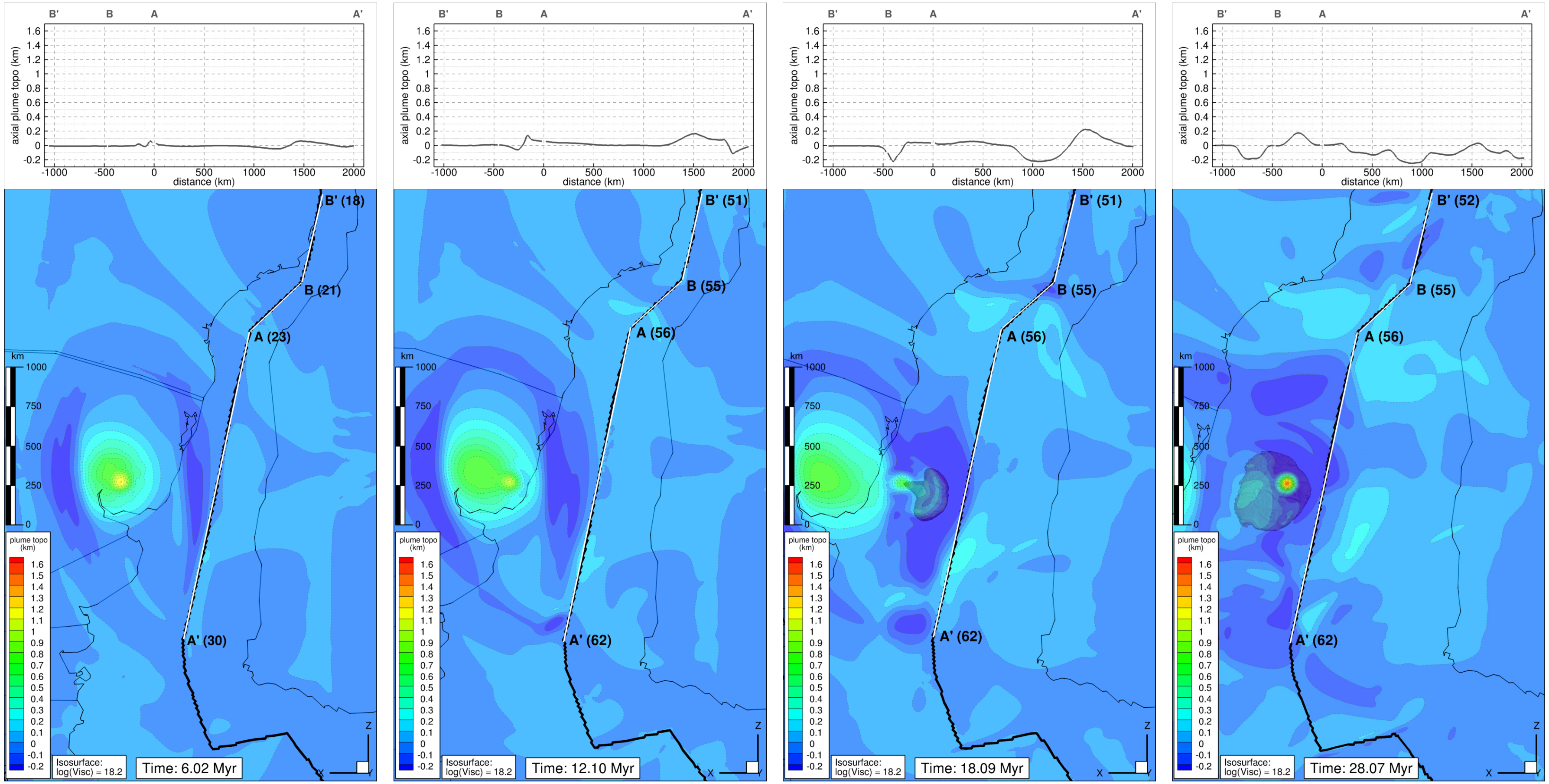


FIGURE S8 Evolution of the plume contribution to the topography for model L4F7.5. Same representation as in Figure 7.

References

Amante, C., and B. Eakins (2009), Etopo1 1 arc-minute global relief model: procedures, data sources and analysis, *Noaa technical memorandum nesdis ngdc-24*, National Geophysical Data Center, NOAA, doi: 10.7289/V5C8276M.

Bauer, K., S. Neben, B. Schreckenberger, R. Emmermann, K. Hinz, N. Fechner, K. Gohl, A. Schulze, R. B. Trumbull, and K. Weber (2000), Deep structure of the Namibia continental margin as derived from integrated geophysical studies, *J. Geophys. Res.*, *105*(B11), 25829–25853, doi: 10.1029/2000JB900227.

Blaich, O. A., J. I. Faleide, and F. Tsikalas (2011), Crustal breakup and continent-ocean transition at South Atlantic conjugate margins, *J. Geophys. Res. Solid Earth*, *116*(1), 1–38, doi: 10.1029/2010JB007686.

Brune, S., A. A. Popov, and S. V. Sobolev (2012), Modeling suggests that oblique extension facilitates rifting and continental break-up, *J. Geophys. Res. Solid Earth*, *117*(8), 1–16, doi: 10.1029/2011JB008860.

Brune, S., C. Heine, M. Pérez-Gussinyé, and S. V. Sobolev (2014), Rift migration explains continental margin asymmetry and crustal hyper-extension, *Nat. Commun.*, *5*, 1–9, doi: 10.1038/ncomms5014.

Brune, S., S. E. Williams, N. P. Butterworth, and R. D. Müller (2016), Abrupt plate accelerations shape rifted continental margins, *Nature*, *536*(7615), 201–204, doi: 10.1038/nature18319.

Buiter, S. J., and T. H. Torsvik (2014), A review of Wilson Cycle plate margins: A role for mantle plumes in continental break-up along sutures?, *Gondwana Res.*, *26*(2), 627–653, doi: 10.1016/j.gr.2014.02.007.

Coffin, M. F., and O. Eldholm (1994), Large igneous provinces: Crustal structure, dimensions, and external consequences, *Rev. Geophys.*, *32*(1), doi: 10.1029/93RG02508.

Contrucci, I., L. Matias, M. Moulin, L. Géli, F. Klingelhofer, H. Nouzé, D. Aslanian, J. L. Olivet, J. P. Réhault, and J. C. Sibuet (2004), Deep structure of the West African continental margin (Congo, Zaïre, Angola), between 5 S and 8 S, from reflection/refraction seismics and gravity data, *Geophys. J. Int.*, *158*(2), 529–553, doi: 10.1111/j.1365-246X.2004.02303.x.

Courtillot, V., C. Jaupart, I. Manighetti, P. Tapponnier, and J. Besse (1999), On causal links between flood basalts and continental breakup, *Earth Planet. Sci. Lett.*, *166*(3-4), 177–195, doi: 10.1016/S0012-821X(98)00282-9.

de Wit, M. J., J. Stankiewicz, and C. Reeves (2008), Restoring pan-african-brasiliano connections: more gondwana control, less trans-atlantic corruption, *Geol. Soc. London, Spec. Publ.*, *294*(1), 399–412, doi: 10.1144/SP294.20.

Dodd, S. C., C. Mac Niocaill, and A. R. Muxworthy (2015), Long duration (>4 Ma) and steady-state volcanic activity in the early Cretaceous Paraná-Etendeka Large Igneous Province: New palaeomagnetic data from Namibia, *Earth Planet. Sci. Lett.*, *414*, 16–29, doi: 10.1016/j.epsl.2015.01.009.

Duncan, R. A. (1984), Age progressive volcanism in the New England Seamounts and the opening of the central Atlantic Ocean, *J. Geophys. Res. Solid Earth*, *89*(B12), 9980–9990, doi: 10.1029/JB089iB12p09980.

Franke, D. (2013), Rifting, lithosphere breakup and volcanism: Comparison of magma-poor and volcanic rifted margins, *Mar. Pet. Geol.*, *43*, 63–87, doi: 10.1016/j.marpetgeo.2012.11.003.

Fromm, T., L. Planert, W. Jokat, T. Ryberg, J. H. Behrmann, M. H. Weber, and C. Haberland (2015), South Atlantic opening: A plume-induced breakup?, *Geology*, *43*(10), 931–935, doi: 10.1130/G36936.1.

Gassmöller, R., J. Dannberg, E. Bredow, B. Steinberger, and T. H. Torsvik (2016), Major influence of plume-ridge interaction, lithosphere thickness variations, and global mantle flow on hotspot volcanism-The example of Tristan, *Geochem. Geophys. Geosyst.*, pp. 1454–1479, doi: 10.1002/2015GC006177.

Gladczenko, T. P., K. Hinz, O. Eldholm, H. Meyer, S. Neben, and J. Skogseid (1997), South Atlantic volcanic margins, *J. Geol. Soc. London.*, *154*, 465–470, doi: 10.1144/gsjgs.154.3.0465.

Gurnis, M., M. Turner, S. Zahirovic, L. DiCaprio, S. Spasojevic, R. Müller, J. Boyden, M. Seton, V. C. Manea, and D. J. Bower (2012), Plate tectonic reconstructions with continuously closing plates, *Comput. Geosci.*, *38*(1), 35–42, doi: 10.1016/j.cageo.2011.04.014.

Hasenclever, J. (2010), Modeling mantle flow and melting processes at mid-ocean ridges and subduction zones development and application of numerical models, Ph.D. thesis, Universität Hamburg.

Hasenclever, J., J. P. Morgan, M. Hort, and L. H. Rüpke (2011), 2d and 3d numerical models on compositionally buoyant diapirs in the mantle wedge, *Earth Planet. Sci. Lett.*, *311*(1-2), 53–68, doi: 10.1016/j.epsl.2011.08.043.

Hirth, G., and D. Kohlstedt (2003), Rheology of the upper mantle and the mantle wedge: A view from the experimentalists, in *Insid. subduction Fact.*, edited by J. Eiler, pp. 83–105, American Geophysical Union, Washington, D. C., doi: 10.1029/138GM06.

Holbrook, W., H. Larsen, J. Koneraga, T. Dahl-Jensen, I. Reid, P. Kelemen, J. Hopper, G. Kent, D. Lizarralde, S. Bernstein, and R. Detrick (2001), Mantle thermal structure and active upwelling during continental breakup in the North Atlantic, *Earth Planet. Sci. Lett.*, *190*(3-4), 251–266, doi: 10.1016/S0012-821X(01)00392-2.

Koopmann, H., S. Brune, D. Franke, and S. Breuer (2014), Linking rift propagation barriers to excess magmatism at volcanic rifted margins, *Geology*, *42*(12), 1071–1074, doi: 10.1130/G36085.1.

Koptev, A., E. Burov, E. Calais, S. Leroy, T. Gerya, L. Guillou-Frottier, and S. Cloetingh (2016), Contrasted continental rifting via plume-craton interaction: Applications to Central East African Rift, *Geosci. Front.*, *7*(2), 221–236, doi: 10.1016/j.gsf.2015.11.002.

Lundin, E. R., and A. G. Doré (2011), Hyperextension, serpentinization, and weakening: A new paradigm for rifted margin compressional deformation, *Geology*, *39*(4), 347–350, doi: 10.1130/G31499.1.

Mihalffy, P., B. Steinberger, and H. Schmeling (2008), The effect of the large-scale mantle flow field on the Iceland hotspot track, *Tectonophysics*, *447*(1-4), 5–18, doi: 10.1016/j.tecto.2006.12.012.

Mohriak, W., M. Nemčok, and G. Enciso (2008), South Atlantic divergent margin evolution: rift-border uplift and salt tectonics in the basins of SE Brazil, *Geol. Soc. London, Spec. Publ.*, *294*(1), 365–398, doi: 10.1144/SP294.19.

Morgan, J. P., W. J. Morgan, Y.-S. Zhang, and W. H. F. Smith (1995), Observational hints for a plume-fed, suboceanic asthenosphere and its role in mantle convection, *J. Geophys. Res. Solid Earth*, *100*(B7), 12753–12767, doi: 10.1029/95JB00041.

Morgan, J. W. (1981), *The Oceanic Lithosphere*, vol. 7, chap. 13. Hotspot tracks and the opening of the Atlantic and Indian Oceans, pp. 443–487, Harvard University Press.

Morgan, W. J. (1971), Convection Plumes in the Lower Mantle, *Nature*, *230*(5288), 42–43, doi: 10.1038/230042a0.

Moulin, M., D. Aslanian, and P. Unternehr (2010), A new starting point for the South and Equatorial Atlantic Ocean, *Earth-Science Rev.*, *98*(1-2), 1–37, doi: 10.1016/j.earscirev.2009.08.001.

Mutter, J. C., M. Talwani, and P. L. Stoffa (1982), Origin of seaward-dipping reflectors in oceanic crust off the Norwegian margin by 'subaerial sea-floor spreading'., *Geology*, *10*(7), 353–357, doi: 10.1130/0091-7613(1982)10<353:OOSRIO>2.0.CO;2.

Nürnberg, D., and R. Müller (1991), The tectonic evolution of the south atlantic from late jurassic to present, *Tectonophysics*, *191*(1), 27–53, doi: 10.1016/0040-1951(91)90231-G.

O'Connor, J. M., and R. A. Duncan (1990), Evolution of the Walvis Ridge-Rio Grande Rise Hot Spot System: Implications for African and South American Plate motions over plumes, *J. Geophys. Res.*, *95*(B11), 17475–17502, doi: 10.1029/JB095iB11p17475.

Pérez-Gussinyé, M., J. P. Morgan, T. J. Reston, and C. R. Ranero (2006), The rift to drift transition at non-volcanic margins: Insights from numerical modelling, *Earth Planet. Sci. Lett.*, *244*(1-2), 458–473, doi: 10.1016/j.epsl.2006.01.059.

Renne, P. R., M. Ernesto, I. G. Pacca, R. S. Coe, J. M. Glen, M. Prevot, and M. Perrin (1992), The Age of Parana Flood Volcanism, Rifting of Gondwanaland, and the Jurassic-Cretaceous Boundary, *Science*, *258*(5084), 975–979, doi: 10.1126/science.258.5084.975.

Renne, P. R., J. M. Glen, S. C. Milner, and A. R. Duncan (1996), Age of Etendeka flood volcanism and associated intrusions in southwestern Africa, *Geology*, *24*(7), 659–662, doi: 10.1130/0091-7613(1996)024<0659:AOEFVA>2.3.CO;2.

Shi, C. (2012), The dynamics of a mantle with plume-fed asthenosphere: Method development and numerical experimental studies, Ph.D. thesis, Cornell University.

Sleep, N. H. (1996), Lateral flow of hot plume material ponded at sublithospheric depths, *J. Geophys. Res.*, *101*(B12), 28065–28083, doi: 10.1029/96JB02463.

Sobolev, S. V., A. V. Sobolev, D. V. Kuzmin, N. A. Krivolutskaya, A. G. Petrunin, N. T. Arndt, V. A. Radko, and Y. R. Vasiliev (2011), Linking mantle plumes, large igneous provinces and environmental catastrophes, *Nature*, *477*(7364), 312–316, doi: 10.1038/nature10385.

Sparks, D. W., E. M. Parmentier, and J. P. Morgan (1993), Three-dimensional mantle convection beneath a segmented spreading center: Implications for along-axis variations in crustal thickness and gravity, *J. Geophys. Res. Solid Earth*, *98*(B12), 21977–21995, doi: 10.1029/93JB02397.

Stica, J. M., P. V. Zalán, and A. L. Ferrari (2014), The evolution of rifting on the volcanic margin of the Pelotas Basin and the contextualization of the Paraná-Etendeka LIP in the separation of Gondwana in the South Atlantic, *Mar. Pet. Geol.*, *50*, 1–21, doi: 10.1016/j.marpetgeo.2013.10.015.

Storey, B. C. (1995), The role of mantle plumes in continental breakup: Case histories from Gondwanaland, *Nature*, *377*(6547), 301–308, doi: 10.1038/377301a0.

Taposeea, C. A., J. J. Armitage, and J. S. Collier (2017), Asthenosphere and lithosphere structure controls on early onset oceanic crust production in the southern South Atlantic, *Tectonophysics*, *716*, 4–20, doi: 10.1016/j.tecto.2016.06.026.

Thiede, D. S., and P. M. Vasconcelos (2010), Paraná flood basalts: Rapid extrusion hypothesis confirmed by new 40Ar/39Ar results, *Geology*, *38*(8), 747–750, doi: 10.1130/G30919.1.

Torsvik, T. H., S. Rousse, C. Labails, and M. A. Smethurst (2009), A new scheme for the opening of the South Atlantic Ocean and the dissection of an Aptian salt basin, *Geophys. J. Int.*, *177*(3), 1315–1333, doi: 10.1111/j.1365-246X.2009.04137.x.

Turner, S., M. Regelous, S. Kelley, C. Hawkesworth, and M. Mantovani (1994), Magmatism and continental break-up in the South Atlantic: high precision40Ar-39Ar geochronology, *Earth Planet. Sci. Lett.*, *121*(3-4), 333–348, doi: 10.1016/0012-821X(94)90076-0.

VanDecar, J. C., D. E. James, and M. Assumpção (1995), Seismic evidence for a fossil mantle plume beneath south america and implications for plate driving forces, *Nature*, *378*(6552), 25–31, doi: 10.1038/378025a0.

White, R., and D. McKenzie (1989), Magmatism at rift zones: The generation of volcanic continental margins and flood basalts, *J. Geophys. Res.*, *94*(B6), 7685–7729, doi: 10.1029/JB094iB06p07685.

White, R. S., G. D. Spence, S. R. Fowler, D. P. McKenzie, G. K. Westbrook, and A. N. Bowen (1987), Magmatism at rifted continental margins, *Nature*, *330*, 439–444, doi: 10.1038/330439a0.

Yamamoto, M., J. P. Morgan, and W. J. Morgan (2007), Global plume-fed asthenosphere flow–I: Motivation and model development, in *Plates, Plumes Planet. Process.*, vol. 430, edited by D. M. Foulger, G. R. Jurdy, pp. 165–188, Spec. Pap. Geol. Soc. Am.

How to cite this article: Taramón, J. M., Morgan, J. P., Shi, C., and Hasenclever, J. (2018), 3-D modelling of South Atlantic rifting.

Chapter 5

Critical evaluation and future prospects

The aim of this thesis is to study the influence of a mantle plume on the initial rifting and break-up of the South Atlantic. This type of regional study is usually done in a 3-D Cartesian box. Using a spherical geometry helps to create a more realistic model of a region. However, implementing realistic boundary conditions along the surfaces of a spherical prism is challenging. My new approach avoids this by using a computational framework consisting of an embedded approach in which regional rifting processes are modelled within a global spherical mesh on which global plate motions are imposed as boundary conditions. For this purpose, I have built upon several pre-existing numerical codes in order to create new algorithms for the demands of the project.

5.1 Mesh Generator

The first stage of this thesis consisted in creating an algorithm to generate 3-D unstructured spherical shell mesh with an embedded high resolution region. The numerical code for solving the thermo-mechanical viscous flow (M3TET_SPH) uses an Eulerian scheme. For this reason, I focused my effort in creating high-quality meshes that contain high-resolution regions rather than optimizing the algorithm to adaptive mesh refinement.

One of the drawbacks of this algorithm is that is not able (yet) to adaptively refine the mesh. However, since an adaptive refinement (or coarsening) will only change node positions in regions where the spatial resolution is changed, most nodes of the spring system remain in equilibrium so that only very few iterations are required to obtain the updated mesh. Future work will consist in merging the codes MESH_3D_SPRING_SPH (mesh generator) and M3TET_SPH (solver) to include an adaptive mesh refinement scheme, for example, coupling the spring constants or the preferred spring length l_0 to strain rates or temperature gradients.

Another caveat of the 3-D mesh generator algorithm is that although it successfully removes the slivers, there is still a small fraction of tetrahedra (~1%) that remains with a quality factor between 0.23 and 0.4. However, most mesh algorithms usually create meshes with a small fraction of elements with relatively low quality [e.g. Alliez et al., 2005; Dardenne et al., 2009]. Although Dompierre et al. [1998] made a proposal for benchmarking 3-D unstructured tetrahedral meshes, the wide variety of shape measure definitions and the lack of more benchmarking makes it difficult to evaluate and compare different mesh optimization algorithms.

5.2 Double Jacobian

The second tool implemented through this thesis consists in a new method to improve the solution of finite element problems in cylindrical or spherical geometries. In a cylindrical geometry, the 2-D Double Jacobian solves the issue of elements crossing $\theta = 2\pi$ by rotating them 180°. The mid-edge nodes of elements crossing $\theta = 2\pi$ computed in the rotated polar system match exactly with the mid-edge nodes of the shared edges with the neighbour elements computed in the original polar system due to their symmetry of the rotation. Hence the 2-D DJ only needs linear elements in polar coordinates to preserve the exact curved boundary.

In a spherical geometry, the issue for elements crossing $\phi = 2\pi$ (Type 2) is solved similarly to the 2-D case by doing a 180° counter-clockwise rotation around the Z axis. The mid-edge nodes of Type 2 elements computed in the 180° rotated spherical system match exactly with the mid-edge nodes of the shared edges with the neighbour elements computed in the original spherical system due to their symmetry with respect to the Z axis. The poor numerical integration near the polar axis is solved by doing a 90° counter-clockwise rotation around the X axis of the elements inside a double cone (Type 3) which places them near the equator in this new reference frame. For elements crossing the cone boundary (Type 4), i.e., elements that link both the 90° rotated spherical frame and the original spherical frame, cubic 20-node elements are used to compute their first Jacobian in the Double Jacobian. This is done because the mid-edge node position computed on an edge of a Type 4 element, i.e., computed in the 90° rotated spherical frame, is not congruent with the mid-edge node position computed on the same shared edge of a Type 1 element, i.e., computed in the original spherical frame. Hence the Double Jacobian in 3-D uses linear elements for Type 1, Type 2 and Type 3 elements and cubic 20-node elements for Type 4 elements to preserve the exact curved boundary.

Even when using 20-node elements for the first Jacobian, DJ results faster than the standard Jacobian mappings due to its analytical mappings for the first and second Jacobians. Since elements in the Double Jacobian method are defined in their polar/spherical coordinates, in which their edges are straight, search routines can easlily find arbitrary points for semi-Lagrange advection schemes.

5.3 South Atlantic experiments

In the last part of the thesis, the tools described in Chapter 2 and Chapter 3 are applied to model the regional influence of the Tristan da Cunha plume in the rifting and break-up of the South Atlantic. The plume flux is a not well constrained parameter. For example, Davies [1988], Turcotte and Schubert [2002] and Sleep [1990] estimate the buoyancy plume flux to 500 kg s^{-1}, 1100 kg s^{-1} and 1700 kg s^{-1}, respectively. They made these estimations assuming that the relief of a hotspot swell reflects the whole upwelling flux of a mantle plume. Yamamoto et al. [2007], within the scenario of a plume-fed asthenosphere [Morgan et al., 1995], estimate the global plume flux to be of order 300 km^3yr^{-1}, divided into ~20-40 mantle plumes, which could imply a typical 7.5 km^3yr^{-1} plume flux for a '40 plume Earth'. In our models we vary the plume flux from 5 km^3yr^{-1} to 20 km^3yr^{-1}. We find that the best plume flux value that simulates a southward migration of the plume material that matches with the SDRs emplacement along the margins is 15 km^3yr^{-1}, close to the value proposed by Yamamoto et al. [2007]. Our models do not account for a power law creep. A composite viscosity including both diffusion and dislocation creep mechanisms would decrease the viscosity of the upper part of the mantle. This would imply that we would need a lower plume flux to get similar southward plume material transport to model L1F15. On the other hand, an additional simplification of the models is that a composition-dependent rheology is not included. Including the potential effect that melting would increase the viscosity of the residue would tend to create a more viscous asthenosphere. It is important to note that although the effects of a power law creep and a composition-dependent rheology act in opposite directions, they do not necessary cancel each other. Future models are needed to assess the roles of these effects.

The initial structure of the lithosphere appears to play an important role in the initial steps of the simulation as plume material drains towards thinner lithosphere regions [Sleep, 1996]. The base of the cratons in our models is completely flat without any small regional tilt. When the plume is located beneath a craton, as for example in models L3F15 and L4F7.5, the plume material ponds and creates a dome at the base of the craton, especially when the plate motions are slow. Making the base of the craton slightly tilted would aid to plume drainage. Smoothing the lower boundaries of the cratons by applying diffusion for tens millions of years before the start of the experiment would also help to create cratons with a more realistic lower surface.

Models in which the plume is located at L1, L1b and L2 present a lateral plume flow migration that is potentially due to a combination of two factors: (1) a 'suction' from regions of greater stretching in the South and (2) the presence of thicker São Francisco and conjugate Congo cratonic roots in the North. Models where the plume is located at L1 show a correlation between a southward speed-up of plume material and abrupt acceleration in the opening speed. For these models, the plume material has enough time to accumulate in the rifting region and start to migrate southwards before the abrupt acceleration in the opening speed occurs. When the opening speed increases abruptly, plume material migrates southwards faster than the mantle upwells below the southern part of the ridge. This behaviour is not observed in models in which the plume is at different locations. For example, plume material in model L2F10 arrives with approximately 4 Myr of delay to the rifting region with respect to model L1F10. At that moment begins the abrupt plate acceleration. Plume material starts to flow southwards but it does not speed-up. This could be due to that mantle upwelling in the southern part inhibits a rapid southward motion of the plume material. Creating a model with the same parameters as model L1F15 but with a delay of the plume arrival of, for example, 8 Myr would test the influence of the abrupt acceleration in the rapid southward migration of the plume material.

The plume contribution to the topography (thermal relief) shows uplift in regions influenced by the plume material. The dynamic topography would also show the contribution from viscous stresses from deeper flow in the mantle. A mesh with more resolution would be required to compute accurate stresses.

One of the limitations of the model resolution does not come from the mesh generator itself but from the algorithm to solve the velocity subproblem. The iterative solution technique consists of a conjugate gradient algorithm preconditioned by a single V-cycle of a geometric multigrid solver, which has been shown to improve the performance considerably compared to other solution algorithms [Hasenclever, 2010; Hasenclever et al., 2011]. A Cholesky forward-backwards substitution is performed on the coarsest level. For this reason, especially in 3-D, the number of nodes of the coarsest multigrid mesh should not exceed ~60000. Table 5.1 presents a comparison of mesh resolution, number of nodes and average running time on a large workstation for the mesh used in the experiments shown in Chapter 4. When creating the mesh, a compromise between the number of nodes and the maximum resolution must be achieved. Creating elements in the refined region with a resolution of 80 km, keeping the same size of the refined region, would double the number of nodes than using a resolution of 100 km for the refined region (Table 5.1).

TABLE 5.1: Comparison of multigrid levels.

multigrid level	resolution of coarse region l_{0c} (km)	resolution of refined region l_{0r} (km)	number of nodes	running time
1 (mesh given by the mesh generator)	2000	100	37000	7 h
		90	48000	11 h
		80	72000	18 h
2	1000	50	290000	3.5 d
		45	375000	5.5 d
		40	560000	9.5 d
3	500	25	2.3 million	24 d
		22.5	2.9 million	–
		20	4.4 million	–
4	250	12.5	17.5 million	–
		11.2	22.6 million	–
		10	34.3 million	–

Chapter 6

Conclusions

- An algorithm for generating unstructured meshes with embedded high resolution regions within 2-D and 3-D Cartesian, 2-D cylindrical and 3-D spherical shell domains has been developed .
- The algorithm employs the FEM to solve for the optimal nodal positions of a spring-like system of preferred nodal positions. A guide-mesh approach is used to smoothly refine the mesh around regions of interest.
- Methods for achieving the expected nodal density and improving the element shape and quality have been presented to ensure a high quality of the generated mesh, which is a prerequisite for high computational accuracy and fast iterative convergence.
- A new technique, the 'Double Jacobian', is presented for more accurate solution of finite element problems in cylindrical and spherical geometries, in particular when element sizes vary strongly within the mesh.
- The Double Jacobian approach computes the local to Cartesian mapping as a two-stage process using analytical expressions that can be more rapidly computed than a general isoparametric or superparametric finite element.
- The use of elements computed in polar or spherical coordinates in which their edges vary linearly (e.g. are 'straight'), allows standard search routines to rapidly find arbitrary points. These elements naturally preserve the appropriate circular or spherical boundary surfaces as well as cylindrical/spherical interfaces in Cartesian coordinates.
- The combination of the Double Jacobian formulation and quadratic finite elements ensures an effective method to study large 3-D problems with spherical geometry using multigrid techniques.
- The influence of Tristan da Cunha plume and initial lithosphere thickness in the early rifting evolution of the South Atlantic has been studied in a series of 3-D numerical models.

- Models consistent with the location of the Tristan plume track where the plume is located at L1, L1b and L2 (Figure 4 in Chapter 4) show a lateral southward plume flow migration with a maximum extent proportional to the assumed plume flux.
- The preferential southward flow observed in the models is mainly due to a combination of suction associated with stretching in the sourthern rifting region and the presence of thicker São Francisco and conjugate Congo cratonic roots in the North, that inhibit northwatd flow.
- In models where plume is located at L1 (see Figure 6 in Chapter 4 for model L1F15), consistent with the early location of the Tristan plume track, plume material experiences a southward speed-up that is related to the abrupt acceleration in the opening speed between 8 Myr and 12 Myr (Figure 5 in Chapter 4).
- The duration of SDR emplacement is predicted to take 5-10 Myr. This rapid southward migration of the plume material could explain the formation of the observed rifting-related SDRs along the South Atlantic margins.

Appendix A

Implementing Velocity BCs in a spherical shell mesh with an embedded high resolution sub-region: South Atlantic application

A.1 Introduction

In this appendix, we describe the tools and steps to implement the velocity boundary conditions from plate kinematic reconstructions in a sphercal shell. We present a particular case for the South Atlantic region.

A.2 Generation of a spherical shell with an embedded hi-resolution region

The first step is to create a mesh according to the purpose of the study. We provide a 3-D spherical mesh generator (springmesh_3d) that is able to create a spherical shell with an embedded high resolution subregion (see Chapter 2 for further information about the mesh generator). For example, to study the South Atlantic region we create a global coarse mesh with a rectangular embedded high resolution subregion (Figure A.1). This mesh is generated by setting a few parameters as e.g., the edge-length of elements within the high resolution region and whithin the coarse region, the quality tolerance of the elements and the mean quality of the elements. The rectangular shape of the high resolution region can also be modified (length, width, depth and position) in order to satisfy the requirements of the region of interest of a particular study case. By default, the mesh generator creates a mesh centering the high resolution region around a point P_0 with colatitud $\theta = 90°$ and longitude $\phi = 90°$, althought these coordinates can be changed. We

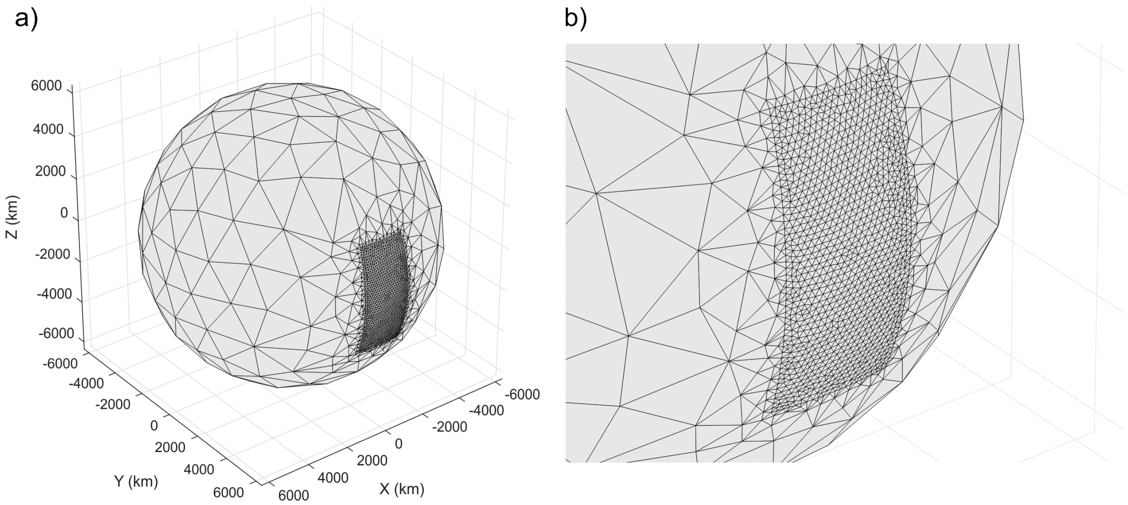


FIGURE A.1: (a) Spherical shell mesh with a high resolution embedded region. (b) Detail of the high resolution region.

recomend to create the high resolution region far from the spherical polar axis since the guide-mesh used in the mesh generator can have dificuties in interpolating the desired element edge length near the polar axis. Then, the high resolution region can be shifted to the region of interest (e.g., South Atlantic) using Finite Rotations. We only have to provide the centre point (colatitude and longitude) of the region of interest.

The output mesh can also be split in multigrid levels to further increment of the resolution. The mesh generator creates an output file (.gpml) compatible with GPlates containing the top surface nodes of the mesh (latitude, longitude).

A.3 Velocities at surface nodes from plate kinematic reconstructions

In this section we explain the basic steps to export velocities at the top surface nodes of the mesh using GPlates. For further details the reader is referred to the Tutorial 5.3: Velocity Fields. Global plate kinematic reconstructions through time are required to generate and export plate velocities. The input files needed to export plate velocities are a rotation file (.rot) to reconstuct the geometries through time, a dinamically-close plate polygons file (.gpml) to define the plates geometry and a mesh file (.gpml). Optionally we can load a coastline file (.dat). Gurnis et al. [2012] provide here the geometry and rotation files for the last 140 Ma. Matthews et al. [2016] also provide here global rotation model from the last 410 Ma. The steps for exporting top surface velocities from plate kinematic reconstructions are:

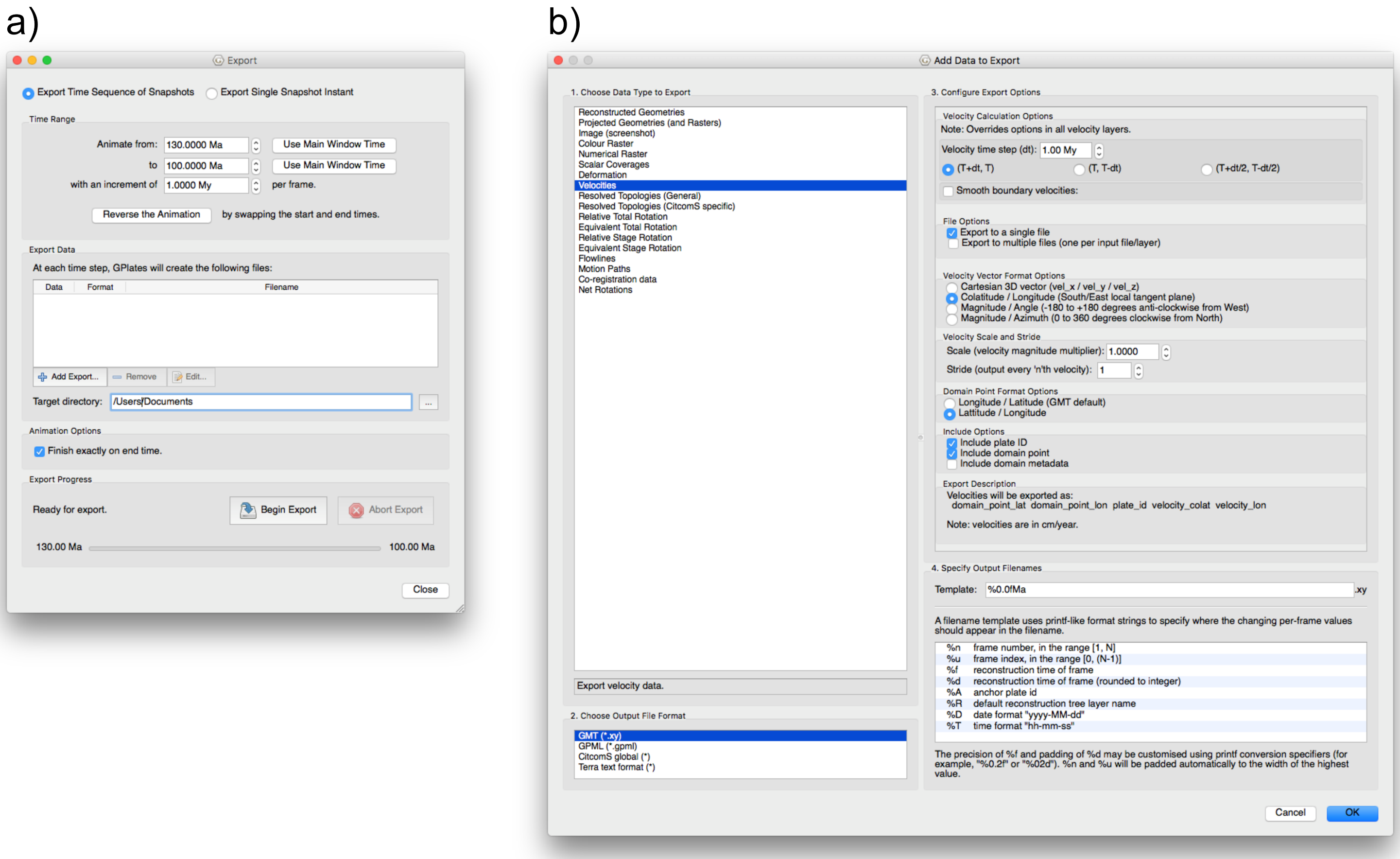
- Click *File* and then *Manage Feature Collections*.

- Click *Open File...* and load the rotation (.rot), geometry (.gpml) and mesh (.gpml) files.
- Click *Reconstruction* and then *Export....*
- Select time in the Time Range box. We export velocities from 130 Ma to 100 Ma with an increment of 1 Ma per frame (Figure A.2a).
- Click *Add export...* in the Export Data box. In the new window select the folowing (Figure A.2b):
  - Box 1: Select Velocities
  - Box 2: Select GMT(*.xy)
  - Box 3: Select the following:
    - Velocity Calculation Options: (T+dt,T)
    - File Options: Export to a single file
    - Velocity Vector Format Options: Colatitude/Longitude
    - Domain Point Format Options: Latitude/Longitude
    - Include Options: Include plate ID and Include domain point
  - Box 4: Type in Template: %0.0fMa
- Select the output directory and click *Begin Animation*

Figure A.3 shows snapshots of the top surface velocities every 6 Ma cenetered in the South Atlantic region. The velocities (arrows) are exported from the plate kinematic reconstructions given by Gurnis et al. [2012] global model. White and black lines represent the plate boundaries and present day coastlines. Yellow dots are the top surface points of the mesh. Brown colour represents the contour of the South America and Africa cratons. Red triangle is the position of the plume.

A.4 Velocity field for the viscous flow

The exported files from GPlates are read and loaded in a finite element convection code (M3TET_SPH) written in MATLAB. Figure shows the global mantle flow solution after solving the viscous flow for the first time step (130 Ma). Black lines represent the plate boundaries. Colour represents the speed in mm/yr and arrows are the velocity vectors.



Export

● Export Time Sequence of Snapshots ○ Export Single Snapshot Instant

Time Range

Animate from: 130.0000 Ma [Use Main Window Time]

to 100.0000 Ma [Use Main Window Time]

with an increment of 1.0000 My per frame.

[Reverse the Animation] by swapping the start and end times.

Export Data

At each time step, GPlates will create the following files:

Data	Format	Filename

[Add Export...] [Remove] [Edit...]

Target directory: /Users/Documents [...]

Animation Options

☑ Finish exactly on end time.

Export Progress

Ready for export. [Begin Export] [Abort Export]

130.00 Ma 100.00 Ma

[Close]

Add Data to Export

1. Choose Data Type to Export

- Reconstructed Geometries
- Projected Geometries (and Rasters)
- Image (screenshot)
- Colour Raster
- Numerical Raster
- Scalar Coverages
- Deformation
- Velocities
- Resolved Topologies (General)
- Resolved Topologies (CitcomS specific)
- Relative Total Rotation
- Equivalent Total Rotation
- Relative Stage Rotation
- Equivalent Stage Rotation
- Flowlines
- Motion Paths
- Co-registration data
- Net Rotations

Export velocity data.

2. Choose Output File Format

- GMT (*.xy)
- GPML (*.gpml)
- CitcomS global (*)
- Terra text format (*)

3. Configure Export Options

Velocity Calculation Options

Note: Overrides options in all velocity layers.

Velocity time step (dt): 1.00 My

● (T+dt, T) ○ (T, T-dt) ○ (T+dt/2, T-dt/2)

□ Smooth boundary velocities:

File Options

☑ Export to a single file
□ Export to multiple files (one per input file/layer)

Velocity Vector Format Options

○ Cartesian 3D vector (vel_x / vel_y / vel_z)
● Colatitude / Longitude (South/East local tangent plane)
○ Magnitude / Angle (-180 to +180 degrees anti-clockwise from West)
○ Magnitude / Azimuth (0 to 360 degrees clockwise from North)

Velocity Scale and Stride

Scale (velocity magnitude multiplier): 1.0000

Stride (output every 'n'th velocity): 1

Domain Point Format Options

○ Longitude / Latitude (GMT default)
● Lattitude / Longitude

Include Options

☑ Include plate ID
☑ Include domain point
□ Include domain metadata

Export Description

Velocities will be exported as:
domain_point_lat domain_point_lon plate_id velocity_colat velocity_lon

Note: velocities are in cm/year.

4. Specify Output Filenames

Template: %0.0fMa .xy

A filename template uses printf-like format strings to specify where the changing per-frame values should appear in the filename.

%n	frame number, in the range [1, N]
%u	frame index, in the range [0, (N-1)]
%f	reconstruction time of frame
%d	reconstruction time of frame (rounded to integer)
%A	anchor plate id
%R	default reconstruction tree layer name
%D	date format "yyyy-MM-dd"
%T	time format "hh-mm-ss"

The precision of %f and padding of %d may be customised using printf conversion specifiers (for example, "%0.2f" or "%02d"). %n and %u will be padded automatically to the width of the highest value.

[Cancel] [OK]

FIGURE A.2: (a) Export animation window. (b) Window with settings to export the data.

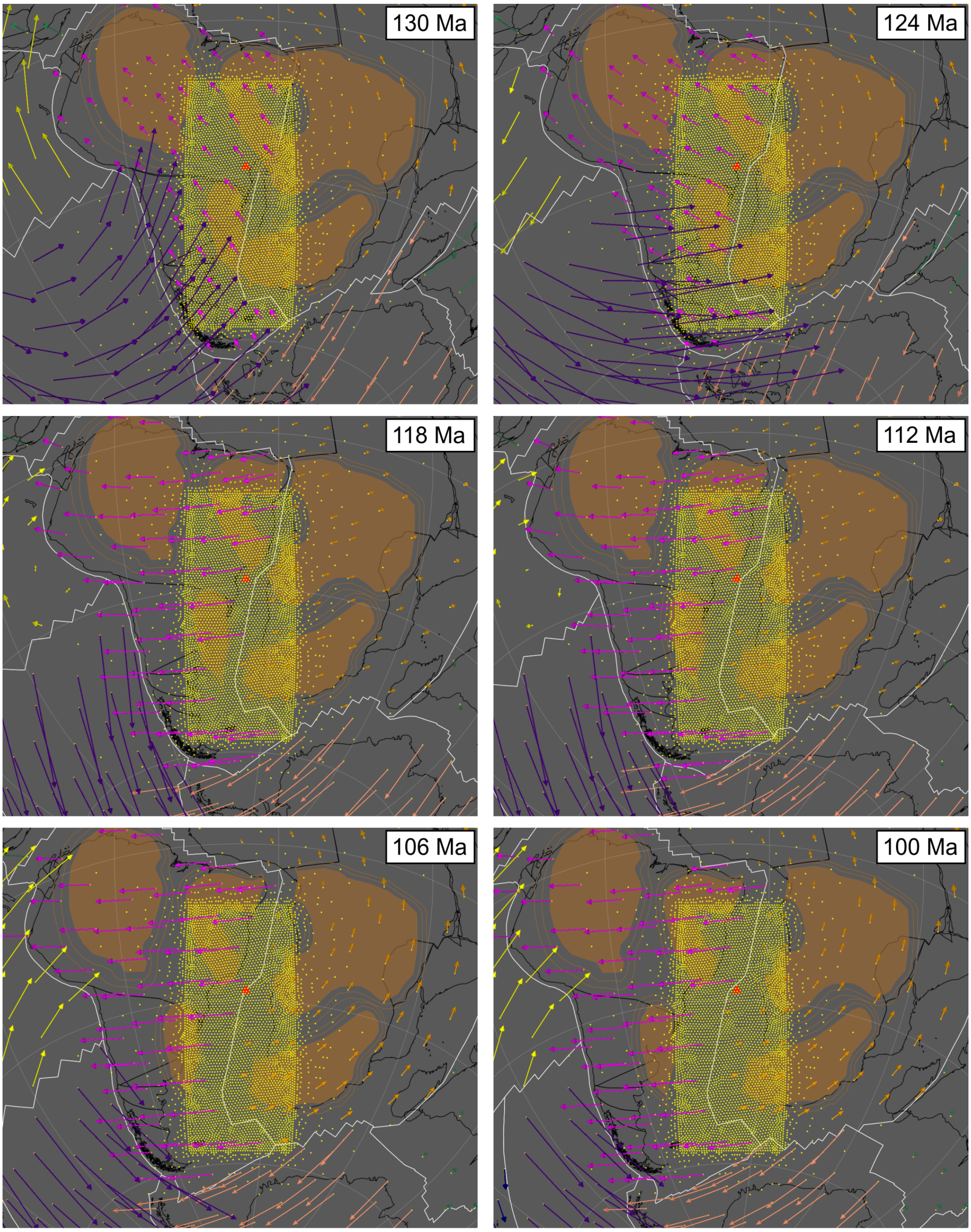


FIGURE A.3: Snapshots for the top surface velocities derived from plate kinematic reconstructions by Gurnis et al. [2012]. White lines represent the plate boundaries. Black lines are the present day coastlines reconstructed back in time. Yellow dots are the top surface nodes of the mesh obtained with the mesh generator. Arrows represent the reconstructed plate velocities. Note that GPlates only plot some arrows to do the image more legible, but the velocity exported data is done for each top surface node. Brown colour represents the contour of the South America and Africa cratons. Red triangle sets the position of the plume.

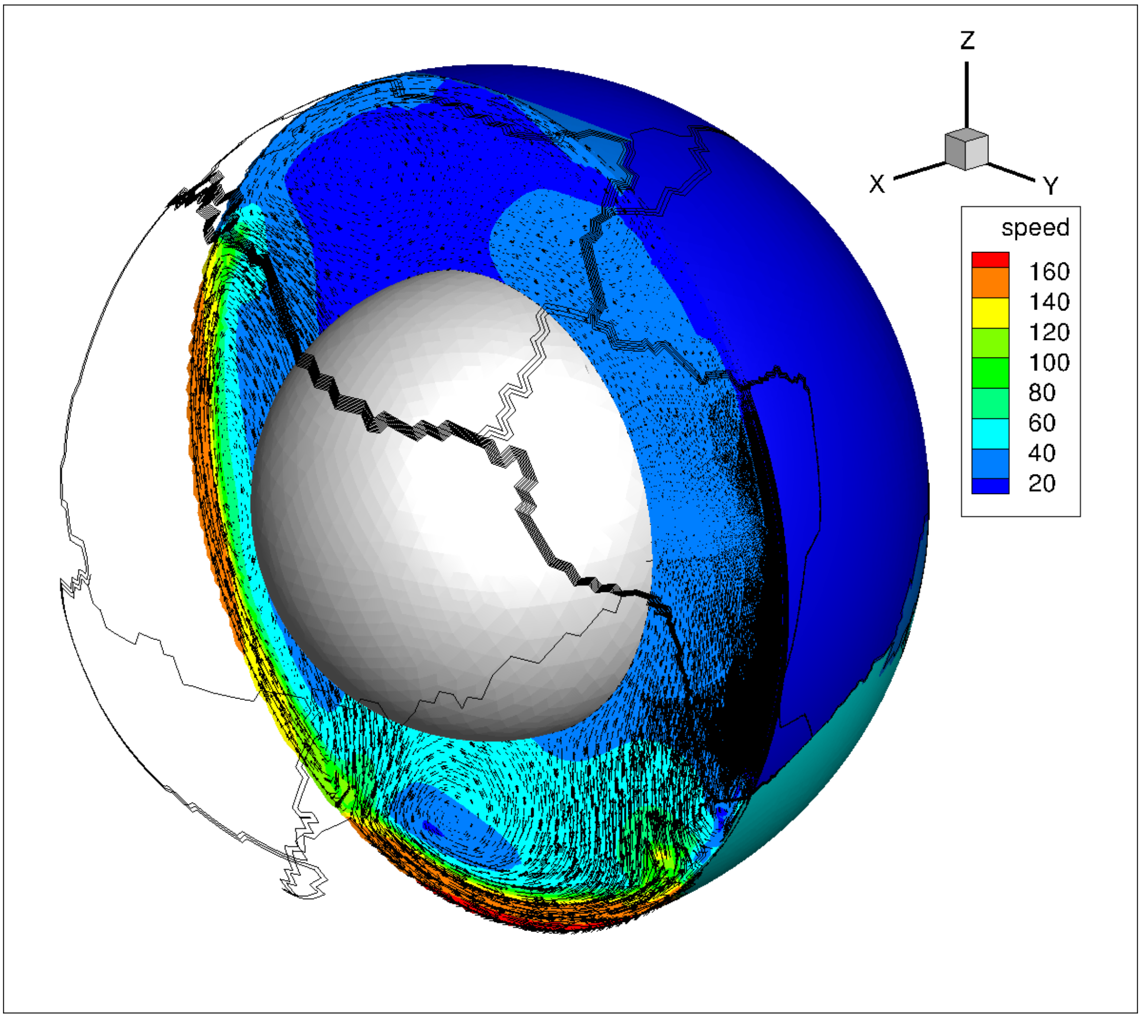


FIGURE A.4: Global mantle flow solution after solving the viscous flow for the first time step (130 Ma). Black lines represent the plate boundaries. Colour is the speed in mm/yr and arrows are the velocity vectors.

Bibliography

Alliez, P., D. Cohen-Steiner, M. Yvinec, and M. Desbrun (2005). Variational tetrahedral meshing. *ACM Trans. Graph.* 24 (3), 617–625. DOI: 10.1145/1073204.1073238.

Amante, C. and B. Eakins (2009). *ETOPO1 1 arc-minute global relief model: procedures, data sources and analysis*. NOAA Technical Memorandum NESDIS NGDC-24. National Geophysical Data Center, NOAA. DOI: 10.7289/V5C8276M.

Andres-Martinez, M. (2016). Rifted margin architecture and the interplay between mantle, crustal and surface processes from geodynamic numerical experiments. PhD thesis. Royal Holloway University of London.

Blaich, O. A., J. I. Faleide, and F. Tsikalas (2011). Crustal breakup and continent-ocean transition at South Atlantic conjugate margins. *J. Geophys. Res. Solid Earth* 116 (1), 1–38. DOI: 10.1029/2010JB007686.

Brune, S., C. Heine, M. Pérez-Gussinyé, and S. V. Sobolev (2014). Rift migration explains continental margin asymmetry and crustal hyper-extension. *Nat. Commun.* 5, 1–9. DOI: 10.1038/ncomms5014.

Brune, S., S. E. Williams, N. P. Butterworth, and R. D. Müller (2016). Abrupt plate accelerations shape rifted continental margins. *Nature* 536 (7615), 201–204. DOI: 10.1038/nature18319.

Coffin, M. F. and O. Eldholm (1994). Large igneous provinces: Crustal structure, dimensions, and external consequences. *Rev. Geophys.* 32 (1). DOI: 10.1029/93RG02508.

Courtillot, V., C. Jaupart, I. Manighetti, P. Tapponnier, and J. Besse (1999). On causal links between flood basalts and continental breakup. *Earth Planet. Sci. Lett.* 166 (3-4), 177–195. DOI: 10.1016/S0012-821X(98)00282-9.

Dardenne, J., S. Valette, N. Siauve, N. Burais, and R. Prost (2009). Variational tetrahedral mesh generation from discrete volume data. *Vis. Comput.* 25 (5-7), 401–410. DOI: 10.1007/s00371-009-0323-7.

Davies, G. F. (1988). Ocean bathymetry and mantle convection: 1. Large-scale flow and hotspots. *J. Geophys. Res. Solid Earth* 93 (B9), 10467–10480. DOI: 10.1029/JB093iB09p10467.

Dompierre, J., P. Labbé, F. Guibault, and R. Camarero (1998). Proposal of benchmarks for 3D unstructured tetrahedral mesh optimization. 7th Int. Meshing Roundtable, p. 525–537.

Franke, D. (2013). Rifting, lithosphere breakup and volcanism: Comparison of magma-poor and volcanic rifted margins. *Mar. Pet. Geol.* 43, 63–87. DOI: 10.1016/j.marpetgeo.2012.11.003.

Geoffroy, L. (2005). Volcanic passive margins. *Comptes Rendus - Geosci.* 337 (16), 1395–1408. DOI: 10.1016/j.crte.2005.10.006.

Gladczenko, T. P., K. Hinz, O. Eldholm, H. Meyer, S. Neben, and J. Skogseid (1997). South Atlantic volcanic margins. *J. Geol. Soc. London.* 154, 465–470. DOI: 10.1144/gsjgs.154.3.0465.

Gurnis, M. et al. (2012). Plate tectonic reconstructions with continuously closing plates. *Comput. Geosci.* 38 (1), 35–42. DOI: 10.1016/j.cageo.2011.04.014.

Hasenclever, J. (2010). Modeling Mantle Flow and Melting Processes at Mid-Ocean Ridges and Subduction Zones Development and Application of Numerical Models. PhD thesis. Universität Hamburg. URL: http://www.sub.uni-hamburg.de/opus/volltexte/2010/4873/.

Hasenclever, J., J. P. Morgan, M. Hort, and L. H. Rüpke (2011). 2D and 3D numerical models on compositionally buoyant diapirs in the mantle wedge. *Earth Planet. Sci. Lett.* 311 (1-2), 53–68. DOI: 10.1016/j.epsl.2011.08.043.

Koptev, A., E. Burov, E. Calais, S. Leroy, T. Gerya, L. Guillou-Frottier, and S. Cloetingh (2016). Contrasted continental rifting via plume-craton interaction: Applications to Central East African Rift. *Geosci. Front.* 7 (2), 221–236. DOI: 10.1016/j.gsf.2015.11.002.

Lundin, E. R. and A. G. Doré (2011). Hyperextension, serpentinization, and weakening: A new paradigm for rifted margin compressional deformation. *Geology* 39 (4), 347–350. DOI: 10.1130/G31499.1.

Matthews, K. J., K. T. Maloney, S. Zahirovic, S. E. Williams, M. Seton, and R. D. Müller (2016). Global plate boundary evolution and kinematics since the late Paleozoic. *Glob. Planet. Change* 146, 226–250. DOI: 10.1016/j.gloplacha.2016.10.002.

McKenzie, D. P. and R. L. Parker (1967). The North Pacific: An example of tectonics on a sphere. *Nature* 216 (5122), 1276–1280. DOI: 10.1038/2161276a0.

Morgan, J. P., W. J. Morgan, Y.-S. Zhang, and W. H. F. Smith (1995). Observational hints for a plume-fed, suboceanic asthenosphere and its role in mantle convection. *J. Geophys. Res. Solid Earth* 100 (B7), 12753–12767. DOI: 10.1029/95JB00041.

Morgan, W. J. (1971). Convection Plumes in the Lower Mantle. *Nature* 230 (5288), 42–43. DOI: 10.1038/230042a0.

Morgan, W. J. (1968). Rises, trenches, great faults, and crustal blocks. *J. Geophys. Res.* 73 (6), 1959–1982. DOI: 10.1029/JB073i006p01959.

Moulin, M., D. Aslanian, and P. Unternehr (2010). A new starting point for the South and Equatorial Atlantic Ocean. *Earth-Science Rev.* 98 (1-2), 1–37. DOI: 10.1016/j.earscirev.2009.08.001.

Mutter, J. C., M. Talwani, and P. L. Stoffa (1982). Origin of seaward-dipping reflectors in oceanic crust off the Norwegian margin by 'subaerial sea-floor spreading'. *Geology* 10 (7), 353–357. DOI: 10.1130/0091-7613(1982)10<353:OOSRIO>2.0.CO;2.

Pérez-Gussinyé, M., J. P. Morgan, T. J. Reston, and C. R. Ranero (2006). The rift to drift transition at non-volcanic margins: Insights from numerical modelling. *Earth Planet. Sci. Lett.* 244 (1-2), 458–473. DOI: 10.1016/j.epsl.2006.01.059.

Shewchuk, J. R. (2002). What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures. Elev. Int. Meshing Roundtable, p. 115–126. DOI: `doi:10.1.1.68.8538`.

Sleep, N. H. (1990). Hotspots and Mantle Plumes: Some Phenomenology. *Geology* 95, 6715–6736. DOI: `10.1029/JB095iB05p06715`.

Sleep, N. H. (1996). Lateral flow of hot plume material ponded at sublithospheric depths. *J. Geophys. Res.* 101 (B12), 28065–28083. DOI: `10.1029/96JB02463`.

Stica, J. M., P. V. Zalán, and A. L. Ferrari (2014). The evolution of rifting on the volcanic margin of the Pelotas Basin and the contextualization of the Paraná-Etendeka LIP in the separation of Gondwana in the South Atlantic. *Mar. Pet. Geol.* 50, 1–21. DOI: `10.1016/j.marpetgeo.2013.10.015`.

Taposeea, C. S. A. (2017). Numerical modelling of the continental break-up of the southern South Atlantic. PhD thesis. Imperial College London.

Turcotte, D. and G. Schubert (2002). *Geodynamics*. 2nd Edition. Cambridge University Press.

White, R. and D. McKenzie (1989). Magmatism at rift zones: The generation of volcanic continental margins and flood basalts. *J. Geophys. Res.* 94 (B6), 7685–7729. DOI: `10.1029/JB094iB06p07685`.

White, R. S., G. D. Spence, S. R. Fowler, D. P. McKenzie, G. K. Westbrook, and A. N. Bowen (1987). Magmatism at rifted continental margins. *Nature* 330, 439–444. DOI: `10.1038/330439a0`.

Wilson, J. T. (1963). Hypothesis of Earth's behaviour. *Nature* 198 (4884), 925–929. DOI: `10.1038/198925a0`.

Yamamoto, M., J. P. Morgan, and W. J. Morgan (2007). Global plume-fed asthenosphere flow–I: Motivation and model development. *Plates, Plumes Planet. Process.* Ed. by D. M. Foulger G. R. Jurdy. Vol. 430. Spec. Pap. Geol. Soc. Am., p. 165–188.