

Mantle Convection in a 2D Spherical Shell

Ana-Catalina Plesa
 Dept. of Planetary Physics,
 Joint Planetary Interior Physics Research Group
 University of Münster and IfP DLR Berlin
 Berlin, Germany
 e-mail: ana.plesa@dlr.de

Abstract—To get a closer look inside the planets and evaluate their mantle dynamics, numerical simulation of thermal convection has proved itself to be a powerful tool. In order to achieve a high resolution, to obtain faster results and to study a larger parameter range, the simulation of mantle convection in a two-dimensional spherical geometry is more efficient than a full three-dimensional spherical shell. In this work, we show the performance and results of a 2D version of GAIA, a mantle convection spherical code with strongly temperature dependent rheology.

Keywords—mantle convection; 2D spherical code; performance; domain decomposition.

I. INTRODUCTION

Numerical simulations have been used to model mantle convection, which may take different forms depending on the planet. On Earth, mantle convection involves recycling of the surface or oceanic lithosphere and results in plate tectonics. Because the lithosphere is relatively cold, recycling the lithosphere represents an extremely efficient way to remove the heat and cool the mantle. On other terrestrial planets, the so-called one-plate planets like Venus and Mars, mantle convection does not involve the outer layers. Instead it occurs below a stagnant lid where heat is transported by conduction. The different characteristics of mantle convection have also a strong influence on the resources needed to simulate the interior dynamics of a planet.

Mantle convection is a highly non-linear process which can be modeled using the conservation equations of the mass, energy and momentum [15]. The simulation time depends on various factors. The size of the grid used for space discretization plays an important role but also the number of time steps needed to reach a solution of the conservation equations is crucial. For realistic calculations a high resolution and small time steps are needed for the convergence of the solution. The computational time increases considerably and thus 2D models are more suited to perform such resource demanding simulations. To run a simulation with a reasonable resolution, the code must work with more than one CPU in parallel.

In the next section several approaches for the domain decomposition of a 2D spherical grid are presented. We also introduce a formula to calculate the overhead of data

exchange between the domains for the two-dimensional spherical grid. In Section III, we illustrate the speedup obtained for 2D grids with up to 128 CPUs on different supercomputer centers. Another performance test consists in the amount of time the simulation needs depending on the size of the grid to reach a stable (steady-state) solution. We present a comparison of the required computational time for both 2D and 3D grids. Section IV illustrates results obtained by using the GAIA framework with two and three-dimensional spherical grids. In order to compare these results, the Nusselt number and plots showing temperature distribution for each case have been computed and compared. We also compare our results obtained with the 2D version of the GAIA code to other published results. In the final section we present our conclusions and give an outlook on future works.

II. DOMAIN DECOMPOSITION

The space discretization is based on the finite-volume method with the advantage of utilizing fully irregular grids in three [9], [10] and two dimensions [13]. The grid contains Voronoi cell information, obtained by computing the Voronoi diagram after performing the Delaunay triangulation of the computational points as shown in Figure 1. To run the code

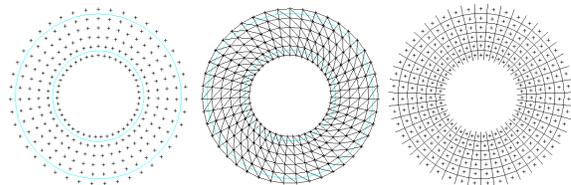


Figure 1. Left: grid with computational nodes, center: Delaunay triangulation of the grid points, right: Voronoi diagram of the 2D spherical grid.

with more than one CPU in parallel a domain decomposition of the grid is applied, which results in an optimal breakdown of the grid into p equal surfaces, where p specifies the amount of domains and processors. An efficient domain decomposition minimizes the area between these sections, leading to a minimized overhead of data exchange between the processors.

Halo-cells, sometimes called ghost-cells, arise in domain decomposition as additional cells in each domain, which form an overlapping zone where data is exchanged. These cells border each domain and are on the same position as their active cells on the neighboring domain. The ratio between halo-cells and grid cells is a first measure of efficiency for parallelization. This ratio is important to determine the amount of data exchanged between the domains.

Using the GAIA Framework with a 2D spherical grid a circle surface will be decomposed into sectors with the same area by equally distributing p potential points on the circle circumference. Thus the coordinates of the i -th potential point can be calculated as follows:

$$p_i.x = \cos(2i\pi/p), p_i.y = \sin(2i\pi/p) \quad (1)$$

The nodes within each sector is assigned to a single processor. The processor assigned to a sector is the one corresponding to the point on the circle circumference closest to the grid nodes in that sector. In Figure 2, we show three possible domain decomposition approaches with the resulting halo-zone: To achieve similar lateral and radial resolution, the

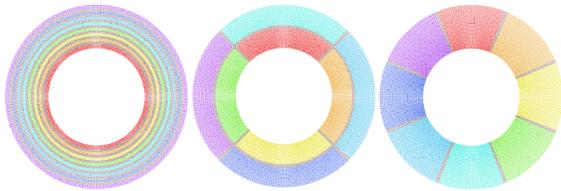


Figure 2. Domain decomposition: left: only radial, center: radial and lateral (RadialSplit), right: only lateral (LateralSplit); each color shows one domain and thus one processor; the gray regions show the Halo-zone between the domains.

number of points per shell is always greater than the number of shells. Keeping this in mind, the first approach for the domain decomposition, where the domains are divided only radial, is the one with the highest number of halo-cells, since in this case the amount of halo-cells scales with the number of points per shell times the number of domains. In the following, we will investigate the performance of the other two approaches (RadialSplit and LateralSplit).

III. PERFORMANCE

The code was tested using four supercomputer centers: HLRN (North-German Supercomputing Alliance), PF-CLUSTER1 (German Aerospace Centre, DLR Berlin), HP XC4000 (Steinbuch Centre for Computing, SCC Karlsruhe) and Itasca (Minnesota Supercomputing Institute for Advanced Computational Research). At HLRN, the HLRN-II SGI Altix ICE 8200 Plus cluster with computational nodes containing two quad-core sockets each for Intel Xeon Harpertown processors with 3 GHz and 16 GB memory per node has been used. The computational nodes on PF-CLUSTER1 have each two quad-core AMD Opteron(tm)

processors running at 2.3 GHz and 16 GB memory per node. On the XC2-Karlsruhe we tested the code with four-way computational nodes each containing two AMD Opteron Dual Cores running at 2.6 GHz with 32 GB per node. On the Itasca cluster we used computational nodes having each two quad-core Intel Xeon X5560 processors running at 2.8 GHz and 24 GB memory per node. In Figure 3, we show the speedup using a 128 shells grid with 152064 computational points which is a typical resolution for our mantle convection simulations. For the runs we used 8 CPUs, 16 CPUs, 32 CPUs, 64 CPUs and 128 CPUs in parallel.

To evaluate the performance, the same initial setup was tested on several node counts. The ratio of the execution time determines the speed-up. This speed-up is therefore the factor that determines the acceleration of the code for the same problem on various CPU counts. The speed-up has been calculated by averaging the time needed for a time-step over 20 time-steps. The "speed-up" factor in Figure 3 is calculated by dividing the amount of time needed with eight CPUs by the amount of time needed with the parallel code. The dotted line in Figure 3 shows the optimal speed-up. Up to 64 CPUs the speedup increases. However, increasing the number of CPUs the number of halo-cells also increases and the performance will eventually drop due to communication overhead.

The speedup calculation in Figure 3 shows a better performance for the Intel Xeon Harpertown processors on HLRN. PFCLUSTER1, Itasca and XC2-Karlsruhe show similar speedup for 128 CPUs although on XC2-Karlsruhe the number of cores per node is limited to 4 while on all other clusters we used 8 cores per node. For 128 CPUs a slower increase in the performance can be observed due to the communication overhead. Figure 4 shows the

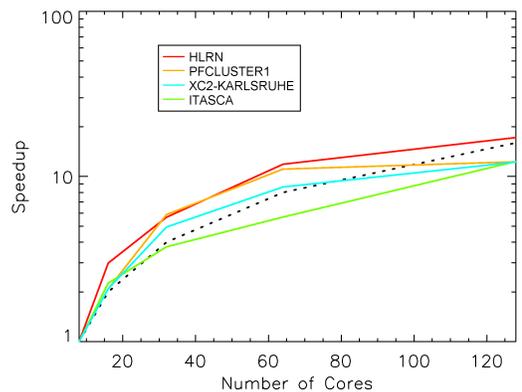


Figure 3. Speedup using a 2D 128 shells grid with 152064 computational nodes on 8, 16, 32, 64 and 128 CPUs on HLRN (red), PFCLUSTER1 (orange), XC2-Karlsruhe (blue) and Itasca (green).

performance obtained when using various grids and both only lateral divisions (LateralSplit) and lateral and radial

divisions (RadialSplit) for the domain decomposition. The RadialSplit shows in most of the cases better performance than the LateralSplit.

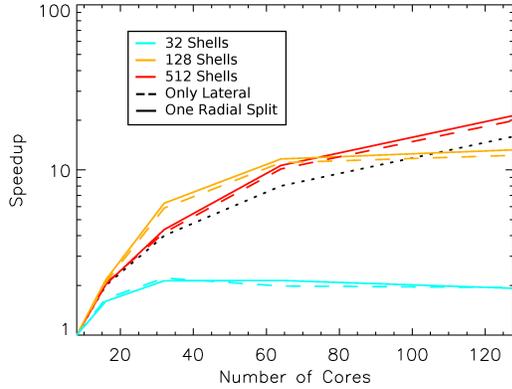


Figure 4. Speedup using three different 2D grids: 32 shells with 9056 computational points grid (blue), 128 shells with 152064 computational points grid (orange) and 512 shells with 2378752 grid (red) and both LateralSplit (dashed lines) and RadialSplit (full lines) domain decompositions.

For 2D grids one can use the following formula to calculate the number of halo-cells needed for both types of domain decomposition.

$$\begin{aligned} n_{cpus} &= lS \cdot (rS + 1) \\ haloCells &= 2 \cdot (ppS \cdot rS + s \cdot lS) \end{aligned} \quad (2)$$

where lS is the number of lateral domains, rS is the number of radial domains, ppS is the number of points per shell and s is the number of shells.

Using formula (2) we can calculate the number of halo-cells for a 2D grid knowing the the number of points per shell, the number of shells and the number of lateral and radial divisions. By increasing the grid's radial resolution

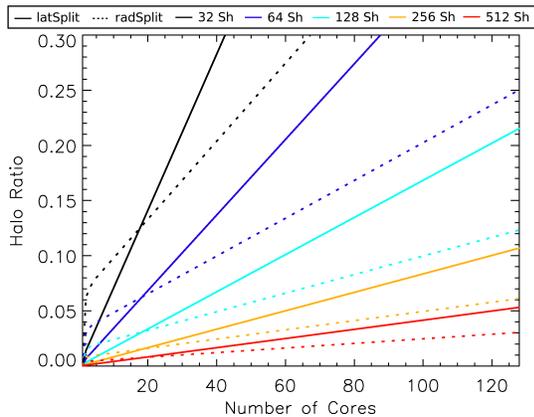


Figure 5. Ratio of halo-cells to computational cells depending on the grid resolution and the number of CPUs used.

and thus the number of shells or by increasing the number

of CPUs used, the RadialSplit shows less halo-cells than the LateralSplit. Therefore the amount of exchanged data is smaller for larger grids and/or larger number of cores used.

In the next table we calculate the amount of time needed to reach a stable (steady-state) solution by using different resolutions for both two- and three-dimensional grids. The

Table I
COMPUTATION TIME DEPENDING ON THE NUMBER OF GRID POINTS AND THE PROBLEM DIMENSION. WE USED THE ISOVISCOUS BENCHMARK-TEST FROM [9] AND 8 CPUS OF AN AMD OPTERON ARCHITECTURE.

Number of shells	Number of points (2D)	Computation time (s) (2D)	Number of points (3D)	Computation time (s) (3D)
16	3114	242.807	46116	2208.531
24	6760	399.519	266292	16797.99
32	11764	437.094	348228	30674.416
40	18186	836.693	430164	45135.616
48	25950	1355.142	512100	65438.916

2D grid is up to one order of magnitude faster than the three-dimensional grid with the same resolution. This is an major advantage when high resolution is needed or a larger parameter space has to be tested.

IV. APPLICATION TO MANTLE CONVECTION

A. Mantle Convection Model

We consider thermal convection in a 2D spherical shell using the GAIA code [9], [10]. The equations used are the equations of conservation of mass momentum and energy [15]. These equations are scaled with the thickness of the mantle as a length scale and with the thermal diffusivity as a time scale. Therefore the non-dimensional equations of a Boussinesq fluid assuming a Newtonian rheology and an infinite Prandtl number are [5]:

$$\nabla \cdot \vec{u} = 0 \quad (3)$$

$$\nabla \cdot [\eta(\nabla \vec{u} + (\nabla \vec{u})^T)] + RaT\vec{e}_r - \nabla p = 0 \quad (4)$$

$$\frac{\partial T}{\partial t} + \vec{u} \nabla T - \nabla^2 T - \frac{Ra_Q}{Ra} = 0 \quad (5)$$

The parameters in the above and following equations are non-dimensionalized using the relationships to physical properties presented in [3] where \vec{u} is the velocity field, η is the viscosity, T is the temperature, \vec{e}_r is the unity vector in radial direction, p is the pressure, t is the time, Ra is the thermal Rayleigh number and Ra_Q is the Rayleigh number for internal heat sources.

The viscosity is calculated using the Arrhenius law for diffusion creep [11]. The non-dimensional formulation of the Arrhenius viscosity law for only temperature dependent viscosity [14] is given by:

$$\eta(T) = \exp\left(\frac{E}{T + T_{surf}} - \frac{E}{T_{ref} + T_{surf}}\right) \quad (6)$$

where E is the activation energy, T_{surf} the surface temperature and T_{ref} the reference temperature.

We choose a fix surface temperature that will not change during the simulation. Depending on the problem, one can choose between no-slip and free slip boundary conditions. For this the velocity vector is decomposed into a lateral part projected onto the boundary and a radial part. In the free slip case the radial component of the velocity is set to zero while material can still move along the boundary whereas in the no-slip case both radial and lateral components are set to zero.

As mentioned earlier, the discretization of the governing equations is based on the finite-volume method with the advantage of utilizing fully irregular grids.

As space is discretized by a fixed grid, time must be discretized as well. For the temporal discretization a fully implicit second-order method, also called an implicit three-level scheme, as shown in [7] has been used. In contrast to spatial discretization, the temporal discretization is flexible and can adapt with a varying time step Δt to the situation. A method proposed by [2] and [12] called SIMPLE was adopted to solve the coupling of the continuity equation with the momentum equation.

Following quantities will be used for the comparison: the root mean square velocity, v_{rms} , and the volume averaged temperature $|T|$. Another value of interest is the Nusselt number, which is defined as the ratio of total heat flux to purely conductive heat-flux. Nu_{top} is the Nusselt number at the surface while Nu_{bottom} is the bottom Nusselt number. Nu_{avg} is the average between Nu_{top} and Nu_{bottom} .

B. 2D-3D Comparison

First we compare the results obtained using a 2D grid to the results obtained using a 3D grid. A spherical harmonics disturbance pattern has been added to the initial temperature field to force the convection to establish a certain symmetry. We choose here the cubic pattern from [10], a pattern which is widely used in steady-state benchmark tests.

When comparing 2D and 3D cases there are some differences which arise from the disagreement in the ratio between inner and outer surface of the 2D and 3D grid respectively [8]. In [17], a scaling was proposed. The inner and outer surface areas of the 2D grid can be fitted the area ratio in the 3D geometry. However this scaling has as a result a smaller inner radius which leads to crowding of structures near the inner portion of the 2D spherical grid [8]. For the comparison we choose an isoviscouse bottom-heated convection with a Rayleigh number of $1e4$ and free-slip boundary conditions for the velocity.

In Figure 6, we present the temperature distribution for both two- and three-dimensional cases. In Table II, we present output values obtained with both the 2D and 3D version of the GAIA code.

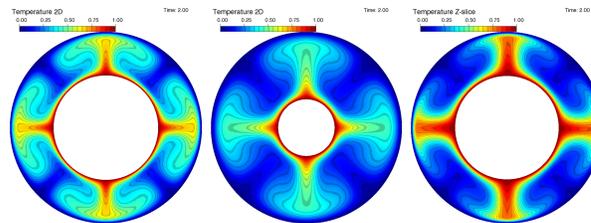


Figure 6. 2D-3D comparison for $Ra = 1e4$, $\Delta\eta_T = 1$: left 2D temperature distribution; center: 2D scaled radii temperature distribution; right: 3D temperature distribution.

Table II
COMPARISON OF THE RESULTS OBTAINED USING THE 2D VERSION AND 3D VERSION OF THE GAIA CODE; THE PARAMETERS USED ARE $Ra = 1e4$, $\Delta\eta_T = 1$ AND FREE SLIP BOUNDARIES.

Case	v_{rms}	$ T $	Nu_{top}	Nu_{bot}	Nu_{avg}
GAIA 2D	46.239	0.382	4.710	4.662	4.686
GAIA 2D scaled	29.817	0.277	3.721	3.633	3.677
GAIA 3D	39.243	0.208	4.029	4.021	4.025

C. Comparison with COMSOL Multiphysics © 3.5

Next we show a comparison with the commercial product COMSOL Multiphysics © 3.5. The sets of tests used for comparison with the COMSOL software [4] include one isoviscouse test with $Ra = 1e4$ and one temperature-dependent viscosity test with $Ra_{0.5} = 1e4$ at a non-dimensional reference temperature $T_{ref} = 0.5$. The activation energy and surface temperature from equation (6) are chosen in such a way that the viscosity contrast across the computational domain is $\Delta\eta_T = 1e6$. For the tests computed using the GAIA code, we use a projected 2D grid, while in COMSOL the mesh is fully irregular. While with GAIA uses finite volume discretization, the discretization scheme in COMSOL is finite element based. In contrast to the 2D-3D comparison tests from the last subsection, we use here no slip top boundary condition for the momentum equation to suppress unrealistic zero-mode from appearing in COMSOL. In Figure 7, we present the results for the isoviscouse case. Both the temperature slice and the velocity field indicate a good agreement between the two codes. The temperature distribution show the same convection structure with four thermal upwellings (plumes) on the axes. The maximum velocity is 49.54 for the GAIA case and 49.495 for the COMSOL run and exhibits identical distribution with high velocity in the areas where a thermal upwelling or a downwelling forms. The next comparison shows a temperature-dependent viscosity case using the Arrhenius viscosity law. In Figure 8, the temperature distribution show similar structure to the previous isoviscouse cases, however due to the temperature-dependent viscosity, the upwelling form changes since the bouyancy term in the conservation of momentum equation decreases with increasing viscosity.

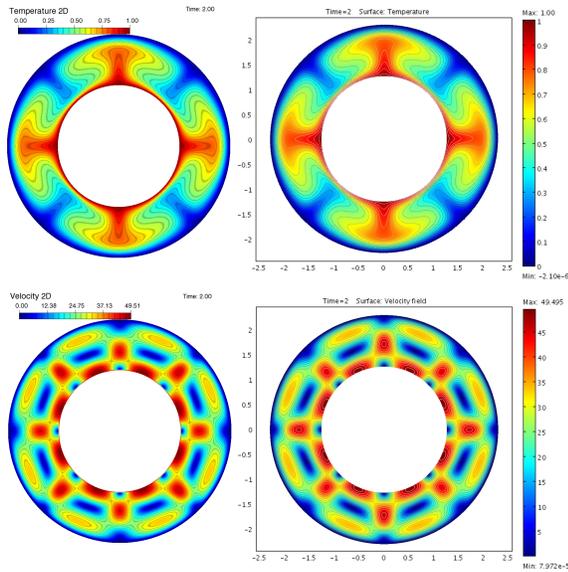


Figure 7. 2D comparison for $Ra = 1e4$, $\Delta\eta_T = 1$; left: 2D temperature (top) and velocity (bottom) distribution with GAIA; right: 2D temperature (top) and velocity (bottom) distribution with COMSOL Multiphysics © 3.5.

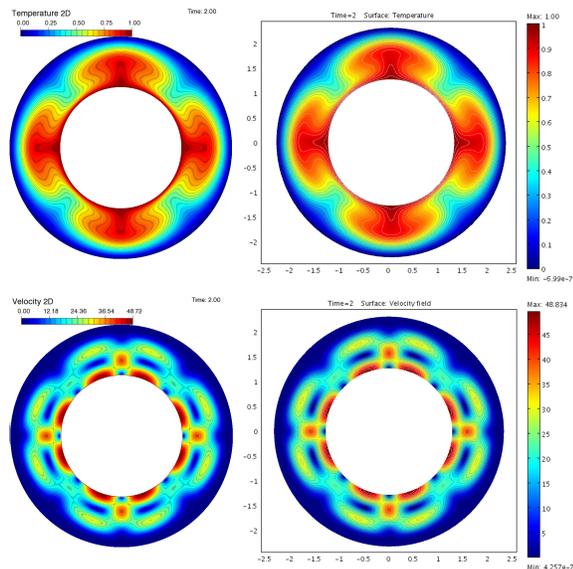


Figure 8. 2D comparison for $Ra_{0.5} = 1e4$, $\Delta\eta_T = 1e6$; left: 2D temperature (top) and velocity (bottom) distribution with GAIA; right: 2D temperature (top) and velocity (bottom) distribution with COMSOL Multiphysics © 3.5.

The highest velocity is in the regions of high temperature (thermal upwellings) since there the viscous forces are weaker than in the rest of the mantle. In these cases, due to the temperature-dependent viscosity, a stagnant lid forms on top of the convecting mantle.

In Table III, we show further output values for both the

isoviscous cases ($\Delta\eta_T = 1$) and the temperature-dependent viscosity case ($\Delta\eta_T = 1e6$) obtained with the 2D Version of the GAIA code and with the COMSOL Multiphysics © 3.5 software.

Table III
COMPARISON OF THE RESULTS OBTAINED USING THE GAIA CODE AND COMSOL MULTIPHYSICS © 3.5 SOFTWARE; THE PARAMETERS USED ARE $Ra = 1e4$, AND NO-SLIP BOUNDARIES.

Case	v_{rms}	$ T $	Nu_{top}	Nu_{bot}	Nu_{avg}
GAIA 2D $\Delta\eta_T = 1$	27.347	0.488	3.255	3.222	3.2385
COMSOL 2D $\Delta\eta_T = 1$	24.966	0.488	3.238	3.139	3.1885
GAIA 2D $\Delta\eta_T = 1e6$	19.396	0.535	2.143	2.121	2.132
COMSOL 2D $\Delta\eta_T = 1e6$	15.257	0.535	2.124	2.087	2.1055

D. Comparison with published results

Next we compare 2D results with other published results. The following table lists benchmark results for an isoviscous case and a temperature-dependent viscosity case (BL stands for [1] and Ha for [6]). Ra_1 is the bottom Rayleigh number. In Figure 9 and Figure 10 we show the temperature distribution and the Nusselt number. The results listed in

Table IV
COMPARISON WITH PUBLISHED RESULTS; BL STANDS FOR BLANKENBACH ET AL.[1] AND HA STANDS FOR HANSEN ET AL.[6].

Case	Grid	Ra	$\Delta\eta_T$	Nu_{top}	Nu_{bot}	Nu_{avg}
Bl	18x18	$1e7$	$1e3$	—	—	9.57
	24x24	$1e7$	$1e3$	—	—	9.63
This	32x325	$1e7$	$1e3$	9.5	9.6	9.55
Ha	60x180	$1e6$	1	—	—	17.20
This	96x1017	$1e6$	1	16.83	17.17	17.0

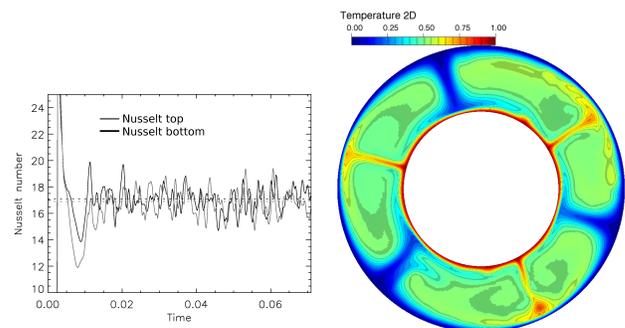


Figure 9. Left: Nusselt numbers as a function of time and right: 2D Temperature slice corresponding to the case from Blankenbach et al. [1].

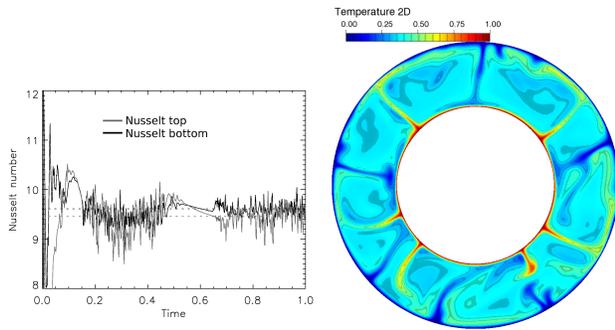


Figure 10. Left: Nusselt numbers as a function of time and right: 2D Temperature slice corresponding to the case from Hansen et al. [6].

Table IV show a good agreement between the two cases computed using GAIA 2D framework and the corresponding cases from [1] and [6]. Although the models from [1] and [6] use 2D box grids, the differences in the Nusselt numbers between [1], [6] and GAIA are within 1.2%. The two cases presented in Figure 10 and Figure 11 show a time-dependent behavior. Unlike the previous cases, the Nusselt number plotted in the right part of the figure varies with time until a quasi-steady-state is reached where the Nusselt number oscillates around a mean value.

V. CONCLUSION AND FUTURE WORKS

In this paper, we presented several methods for domain decomposition of 2D spherical grids and a formula for computing the resulting number of halo-cells and therefore the communication overhead. The performance tests show a super-linear speedup for the 2D simulations and a computation time needed to reach a steady-state solution at least one order of magnitude smaller than the one needed for the three dimensional cases. Further, comparison of 2D and 3D results show a good agreement in convection mode (number of thermal upwellings and downwellings) and Nusselt number. The validation with the commercial product COMSOL Multiphysics © 3.5 yielded satisfying results for both isoviscous and temperature-dependent viscosity cases. Comparison with other published results showed similar results for the Nusselt numbers.

A future goal is to include active compositional fields in our model. For this, further benchmarks as well as performance tests will be needed.

ACKNOWLEDGMENT

This research has been supported by the Helmholtz Association through the research alliance "Planetary Evolution and Life". The author thanks the anonymous reviewers for their comments, which helped improving the manuscript.

REFERENCES

[1] B. Blankenbach et al., *A benchmark comparison for mantle convection codes*, Geophys. J. Int., 1989, 98, 23-38.

[2] L.S. Caretto, A.D. Gosman, S.V. Patankar, and D.B. Spalding, *Two calculation procedures for steady, three-dimensional flows with recirculation*, Third Int. Conf. Numer. Methods Fluid Dyn., Paris, 1972.

[3] U. Christensen, *Convection with pressure- and temperature-dependent non-Newtonian rheology*, Geophysical Journal- Royal Astronomical Society, 1984, 77, 343-384.

[4] G., Couberbaisse, *A comparison between the software COMSOL © and a lattice Boltzmann code when simulating behaviour of viscous material*, European COST P19 project, CONFERENCE on Multiscale Modelling of Materials, Brno, Academy of Sciences - Masaryk University, Czech Republic.

[5] O. Grasset and E.M. Parmentier, *Thermal convection in a volumetrically heated, infinite Prandtl number fluid with strongly temperature-dependent viscosity: implications for planetary thermal evolution*, J. geophys. Res., 1998, 103, 18 17118 181.

[6] U. Hansen, D.A. Yuen, and A.V. Malevsky, *Comparison of steady-state and strongly chaotic thermal convection at high Rayleigh number*, Physical Review A, 1992, 46, 4742-4754.

[7] H. Harder and U. Hansen, *A finite-volume solution method for thermal convection and dynamo problems in spherical shells*, Geophysical Journal International, 2005, 161, 522- 532.

[8] J. W. Hernlund and P. J. Tackley, *Modeling mantle convection in the spherical annulus*, Phys. Earth Planet. Inter., 171 (1-4), 48-54. doi: 10.1016/j.pepi.2008.07.037.

[9] C. Huettig and K. Stemmer, *The spiral grid: A new approach to discretize the sphere and its application to mantle convection*, Geochim. Geophys. Geosyst., 9, Q02018, 2008, doi:10.1029/2007GC001581.

[10] C. Huettig and K. Stemmer, *Finite volume discretization for dynamic viscosities on Voronoi grids*, Phys. Earth Planet. Interiors, 2008, doi: 10.1016/j.pepi.2008.07.007.

[11] S. I., Karato and P. Wu, *Rheology of the upper mantle: a synthesis*, Science 260, 771-778, 1986.

[12] S.V. Patankar, *Numerical heat transfer and fluid flow*, edn, Vol., pp Pages, Mc-Graw-Hill, New York, 1980.

[13] A.-C. Plesa and C. Huettig, *Numerical Simulation of Planetary Interiors: Mantle Convection in a 2D Spherical Shell*, (Abstract), Workshop on Geodynamics 2008, Herz- Jesu-Kloster, Neustadt, Waldstr. 145 67434, Neustadt/Weinstrasse, Sept 30 - Oct 2, 2008.

[14] J. H., Roberts and S., Zhong, *Degree-1 convection in the Martian Mantle and the origin of the hemispheric dichotomy*, Journal of Geophysical Research E: Planets 111, 2006.

[15] G. Schubert, D.L. Turcotte, and P. Olson, *Mantle Convection in the Earth and planets*, Cambridge University Press, 2001.

[16] V.S. Solomatov and L.-N. Moresi, *Scaling of time-dependent stagnant lid convection: Application to small-scale convection on the earth and other terrestrial planets*, J. Geophys. Res., 2000,105, 21795-21818.

[17] P.E., van Keken, *Cylindrical scaling for dynamical cooling models of the earth*, Phys. Earth Planet. Interiors, 2001, 124, 119-130.