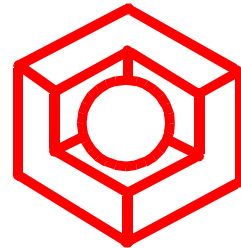


BremHLR

Competence Center of
High Performance Computing Bremen



amatos: parallel adaptive mesh generator for atmospheric and oceanic simulation

Jörn Behrens^{*},
Natalja Rakowsky[‡], Wolfgang Hiller[‡],
Dörthe Handorf[†], Mathias Läuter[†], Jürgen Pöpke[†],
Klaus Dethloff[†]
□□

BremHLR Technical Report 02-03

^{*} Technical University of Munich, Germany

[‡] Alfred-Wegener-Institute for Polar and Marine Research, Bremerhaven

[†] Alfred-Wegener-Institute for Polar and Marine Research, Potsdam

amatos: parallel adaptive mesh generator for atmospheric and oceanic simulation

Jörn Behrens^a, Natalja Rakowsky^{b,*}, Wolfgang Hiller^b,
Dörthe Handorf^c, Matthias Läuter^c, Jürgen Pöpke^c,
Klaus Dethloff^c

^a*Technische Universität München, Center for Mathematical Sciences, Garching, Germany*

^b*Alfred Wegener Institute for Polar and Marine Research, Computing Center, Bremerhaven, Germany*

^c*Alfred Wegener Institute for Polar and Marine Research, Potsdam, Germany*

Abstract

The grid generator **amatos** has been developed for adaptive modelling of ocean and atmosphere circulation. It features adaptive control of planar, spherical, and volume grids with triangular or tetrahedral elements refined by bisection. The user interface GRID API, a Fortran 90 module, shields the application programmer from the technical aspects of mesh adaption like **amatos**' hierarchical data structure, the OpenMP parallelization, and the effective calculation of a domain decomposition by a space filling curve (SFC) approach.

This article presents the structure of **amatos**, the user interface GRID API, the powerful SFC ordering and decomposition of data, and two applications of **amatos**, namely the modelling of tracer advection in the polar vortex and the development of the adaptive finite element atmosphere model PLASMA (parallel large scale model of the atmosphere).

Key words: Grid generation, finite elements, parallelization, ocean and atmosphere modelling

* Corresponding author. Tel.: +49-471-4831-1582; fax: +49-471-4831-1590.
Email address: nrakowsky@awi-bremerhaven.de

1 Introduction

The grid generator `amatos` (Behrens, 1996, 2003) was developed to make adaptive mesh generation available for dynamic time dependent flow problems as arising in climate simulations.

The philosophy of `amatos` is to hide away all nontrivial tasks concerning mesh generation and adaptation from the application programmer. The complete mesh generation process can be controlled by approx. 20 Fortran 90 subroutines of the application programming interface (GRID API), which also provides additional variables, data structures and constants.

The generic field of application for `amatos` is atmosphere and ocean circulation modelling with semi-Lagrangian advection schemes. The time stepping scheme, however, is not part of `amatos`. The semi-Lagrangian background is merely reflected by the choice of service routines (e.g., interpolation) implemented in the GRID API, which may be extended for further needs. At the time being, `amatos` provides

- a F90-interface based on modular software techniques,
- mesh adaptation (i.e., refinement and coarsening corresponding to a given error criterion) of planar, spherical, or volume grids with triangular or tetrahedral elements,
- adaptivity by hierarchical data structures (refinement by bisection method),
- support of arbitrary finite element (FEM) basis functions (to be defined by the user, linear and quadratic functions are preinstalled),
- fast mesh partitioning by a space filling curve (SFC) approach (for parallelization by domain decomposition and as cache efficient ordering for local calculations),
- service routines e.g., for interpolation,
- shared memory parallelisation with OpenMP (under construction).

2 Using `amatos`

2.1 *Setup*

2.1.1 *Compiling*

Before compiling the library, the user has to choose between the planar, spherical, or volume version of `amatos`. OpenMP parallelization is toggled by the corresponding compiler directives. `amatos` has been successfully executed on

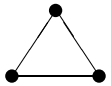

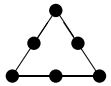
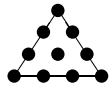
different hardware and operating systems, including SGI Origin, SUN Fire, IBM Regatta, Linux with Intel's Fortran90 compiler ifc, and comes with corresponding Makefiles.

2.1.2 Declaring Finite Element Types

If a required finite element type is not yet predefined, a signature data structure describing this type has to be added in the `amatos` source code. At the time being, `amatos` includes the definition of linear (unknowns defined at vertices) and quadratic (unknowns defined at vertices and edges) Lagrange elements. Table 1 shows the components of the Fortran 90 data structure `fem_signatur` and values for some finite element types.

Table 1

Components of the data structure `fem_signatur` and examples for commonly used finite element types.

element type	linear Lagr.	lin. non- conform.	quadr. Lagr.	cubic Lagrange
				
<code>i_order:</code> order	1	1	2	3
<code>i_unknowns:</code> total #variables	3	3	6	10
<code>i_npoints:</code> #var. per vertex	1	0	1	1
<code>i_gpoints:</code> #var. per edge	0	1	1	2
<code>i_epoints:</code> #var. per element	0	0	0	1
<code>r_gweights:</code> var. pos. on edge	–	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{3}, \frac{2}{3}), (\frac{2}{3}, \frac{1}{3})$
<code>r_eweights:</code> var. pos. on elem.	–	–	–	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

Once the library has been compiled and an application uses `amatos`, each variable in the application has to be registered with its finite element type in the initialization phase before any mesh items are created. Now, the appropriate memory will be allocated for each unknown position corresponding to that finite element type.

2.1.3 Initial Mesh

The user has to provide an initial coarse mesh of the computational domain in a special input file. The triangular or tetrahedral mesh must resolve the computational domain and should be as coarse as possible. Two examples for an initial triangulation and resulting refined grids for regular geometries are shown in fig. 1 and 2, more complex examples are subject of section 4.3, see fig. 11.

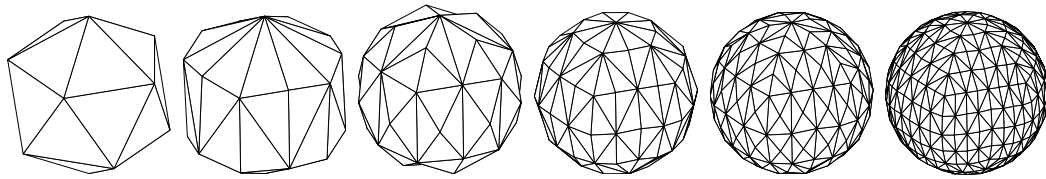


Fig. 1. Initial triangulation (icosahedron) and uniform refinement of a spherical grid.

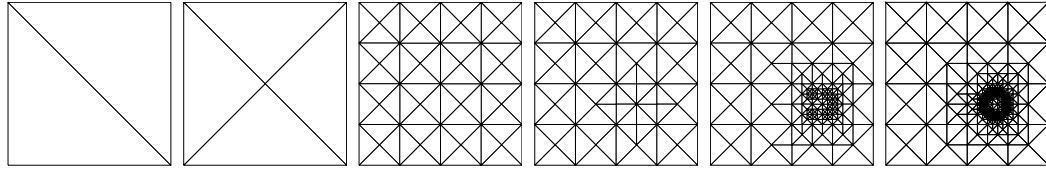


Fig. 2. Initial triangulation and subsequent uniform and adaptive refinement of a square grid.

The file is constructed with keywords preceding the corresponding values. It contains the vertex coordinates and the mesh topology with information on boundary edges. To enhance the performance of some calculations, the vertices of a (triangular) element should be given in counterclockwise order. If this is not the case, **amatos** issues a warning and reorders the vertices.

For each element, one edge has to be marked for the first refinement step. One full refinement step (one bisection performed on all elements) needs not to result in a conforming mesh, because **amatos** checks for hanging vertices and refines the neighbouring elements until the mesh is conforming.

The list of keywords and the data types in **amatos** can be extended to handle further properties of the mesh. E.g., in oceanographic applications one could distinguish between different coastlines or open boundaries by tagging the boundary edges or vertices.

2.2 The User Interface GRID API

The application programming interface GRID API provides routines, data, and parameters to the programmer of a model code, but hides the underlying complicated data structures and methods of grid adaption in **amatos** as illustrated in fig. 3. The routines in the GRID API can be classified according to two characteristics.

- (1) Think of the adaptive algorithm as a two phase procedure: First, the mesh is generated/adapted and each mesh item keeps associated data. In the second phase, all required data is gathered from the mesh items into vectors and numerical calculations are performed on vectors (the consecutive data storage allows for efficient pipelined or vectorized execution).

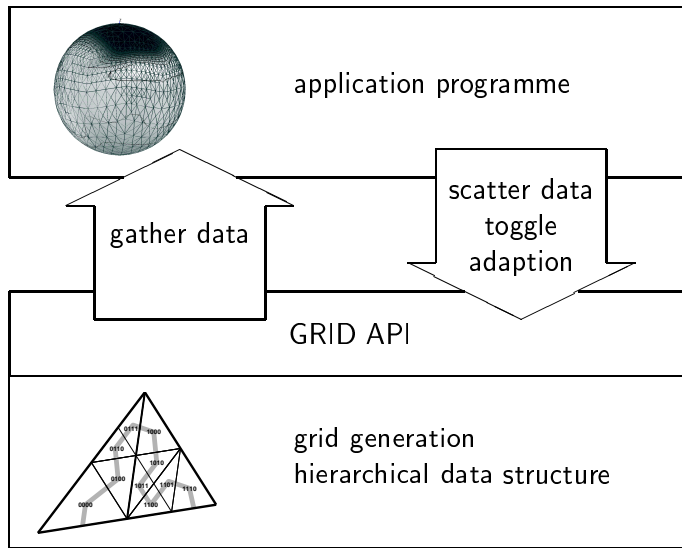


Fig. 3. The grid application programming interface

Finally, the results are scattered back to mesh item storage positions.

- (2) Think of the program as a data flow with methods acting on and manipulating the data. A data structure called `grid_handle` represents a specific instance of the mesh. Methods act on the instance, manipulating it.

The F90 structure `grid_handle` contains parameters describing an instance of the mesh. A grid handle for each time level that has to be held during a model run (`DEF_timesteps` time levels, typically two or three) is provided by the GRID API as a public variable `p_grid` of type `grid_handle` and dimension `DEF_timesteps`. For the application programmer, the most important components of `grid_handle` are the following integers

- `i_timetag`: value for current time tag,
- `i_enumfine`, `i_gnumfine`: number of elements or edges on the computational grid (in contrast to `i_enumber`, `i_gnumber`, i.e., total number of elements or edges on all refinement levels in the hierarchical structure),
- `i_nnumber`: number of vertices on the computational grid (for vertices, `i_nnumfine` would be equal to `i_nnumber`),
- `i_unknowns` of dimension `i_femtypes`: for each finite element type, the total number of unknowns on the computational grid.

With the `grid_handle`, the user can allocate the appropriate memory for data vectors, gather or scatter the desired mesh data via the GRID API, and toggle the mesh generation. Here, we can only give a very rough overview of the most important routines. For a complete description of the GRID API, we refer to `amatos`' user manual (Behrens, 2003).

Mesh creation and termination

`grid_initialize` initializes `amatos`,
`grid_registerfemvar` registers variables to a desired finite element type,
`grid_createinitial` reads the triangulation file (see section 2.1.3) and refines uniformly to the coarsest user-given level,
`grid_writesaveset`, `grid_readsaveset` write and read restart files
`grid_terminate` frees the memory.

Data retrieval

`grid_getinfo` gathers information from grid items into consecutive arrays,
`grid_getiteminfo` gathers information from an individual mesh item,
`grid_putinfo` scatters information from data arrays into the mesh items.

Numerical calculation

`grid_coordvalue` interpolates a variable at a given coordinate (bi-linear, bi-cubic spline, thin plate spline),
`grid_coordgradient`, `grid_nodegradient` estimates the gradient of a variable at a given coordinate, or at a given vertex, respectively,
`grid_integral` integrates a variable over the domain,
`grid_nodearea` calculates the area of influence for grid vertices,
`grid_polygridintersect` calculates the intersection of a given polygon with the mesh, i.e., determines the intersected elements and intersection area,
`grid_boundintersect` calculates the intersection of a line with the boundary.

Mesh control

`grid_adapt` adapts the grid according to the elements' markers set by `grid_putinfo`,
`grid_timeduplicate` duplicates the grid, typically for a new timestep,
`grid_domaincheck` checks if a given coordinate belongs to the mesh,
`grid_createdual` creates the dual mesh.

3 Grid Partitioning with a Space-Filling Curve Approach

The grid partitioning is a crucial part of the adaptive mesh generator. Sophisticated algorithms as e.g. provided by METIS (Karypis and Kumar, 1998) are not feasible for adaptive time stepping codes where the partitioning has to be recalculated over and over again. For `amatos`, Behrens and Zimmermann (2000) employ a space filling curve approach (SFC) that is implemented straight forward for a hierarchical triangular adaptive grid. This partitioning is very fast, guarantees optimal load balancing by construction and results in rather short (though not optimal) interface length. For the quality of SFC induced partitions see Zumbusch (2001). Furthermore, due to the fractal nature of the SFC the resulting ordering retains data locality on any level of

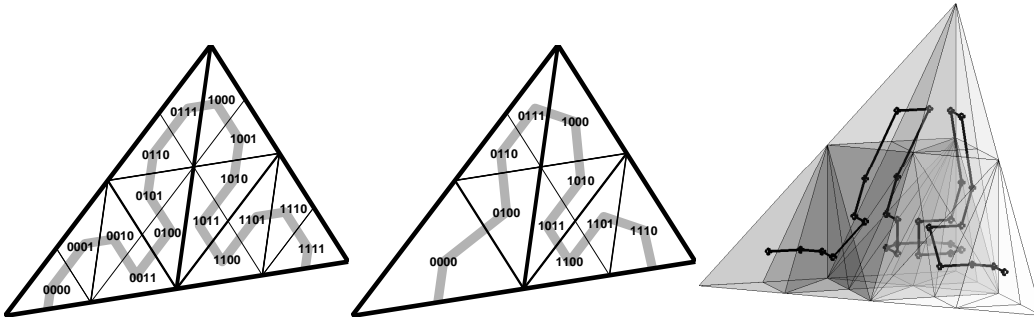


Fig. 4. Construction of the space filling curve (SFC) through the elements for a) uniform planar, b) unregularly refined, and c) uniform tetrahedral grids.

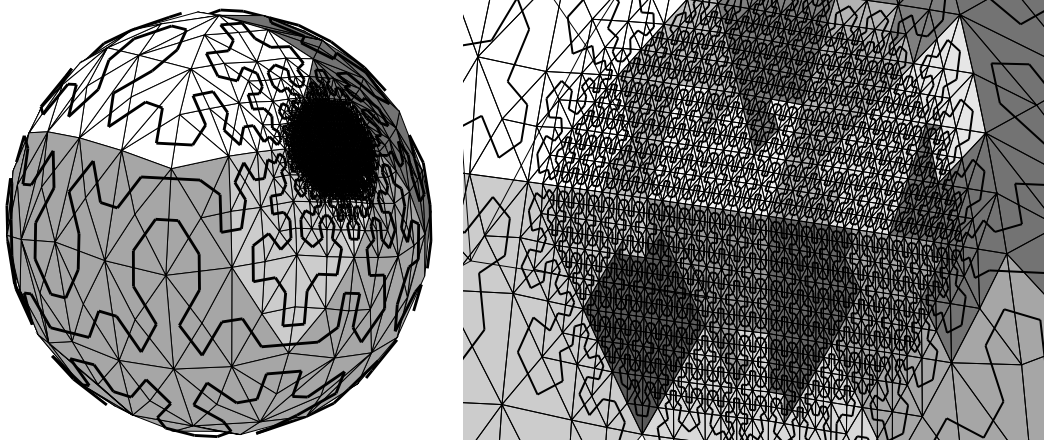


Fig. 5. SFC through the elements and a mesh partitioning into 14 subdomains

memory hierarchy from cache levels to main memory, making a further local renumbering on subdomains obsolete.

3.1 SFC algorithm

The global SFC index of an element is calculated at the time of the elements' construction by bisection of the parent triangle. As depicted in fig. 4, a) and b), the algorithm is straight forward when regarding the indices as binary numbers starting with zero. The first bisection on the coarsest level divides the parent element into a child triangle on the left hand side of the bisecting edge receiving the leading bit 0, and a child triangle on the right with entry 1 as the first bit. The second bisection yields the second bit of the binary index, but the role of the left and right hand side is interchanged (second bit 1 on the left, second bit 0 on the right). Stepping down all levels of refinement and adding the corresponding bits according to the relative position in the bisected parent triangle gives the index in the SFC. For an unstructured grid, gaps in the numbering have to be removed after finishing this procedure.

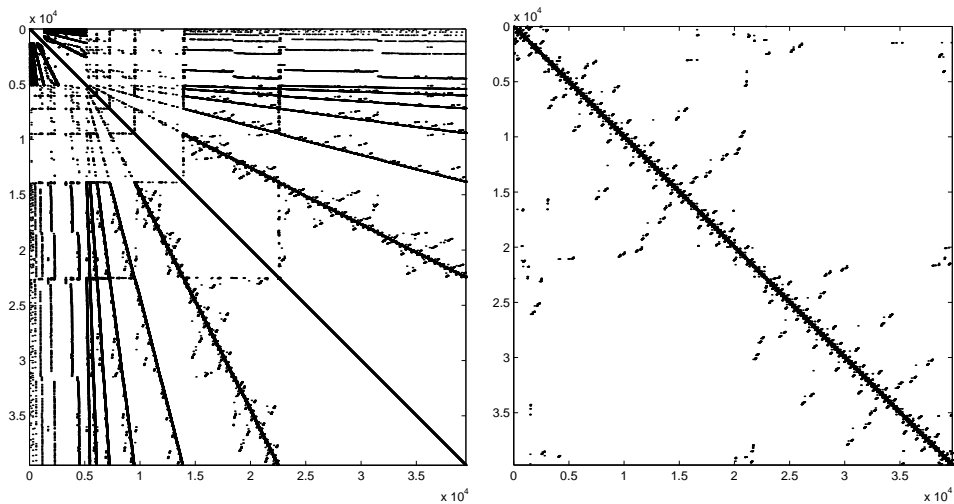


Fig. 6. Sparsity pattern of the stiffness matrix for an adaptively refined grid with original numbering (nodes in the order of creation) and rSFC numbering of nodes.

The SFC algorithm can easily be extended to tetrahedral grids with refinement by bisection (fig. 4, c). Figure 5 shows an adaptive grid on the sphere with an SFC through the elements and a zoom into the refined region. The partitioning is obtained by simply cutting the SFC into equal chunks.

In most applications, node wise calculations play a more important role than loops through the elements. More generally, one is ultimately interested in ordering the unknowns. Taking this into account, `amatos` sets up an ordering of all unknowns located at vertices, edges, and/or elements (depending on the order of the FE basis function) by collecting and counting them on the way along the elements' SFC. For matrix ordering, the reverse of this numbering should be taken to obtain a local arrow head down shape in the sparsity pattern. When the user retrieves data arrays from the GRID API, all arrays are sorted along this reverse SFC, abbreviated as rSFC.

The impact of the rSFC numbering of unknowns on the sparsity pattern of a typical stiffness matrix for linear FE can be studied in fig. 6 where most entries are gathered close to the diagonal. More complex matrices for coupled linear equations with several variables and mixed FE (e.g., in the context of the atmosphere model PLASMA, see section 4.2) can be set up according to the SFC numbering, too, which results in a very similar sparsity pattern.

In the context of ILU (incomplete LU-factorization) preconditioning, the rSFC ordering proves to be a powerful candidate among other well known fill-in reducing orderings like RCM (reverse Cuthill-McKee) and minimum degree. An example for a coupled linear equation in PLASMA with approx. 200 000 unknowns is given in table 2. We employ the FoSSI solver interface (Frickenhaus et al., 2003) and choose the PETSc solver library (Balay et al., 2003) with BiCGstab and level ILU(1) as preconditioner to reduce the residuum by a

Table 2

Comparison of the influence of matrix orderings on the iteration count and computation time for ILU(1)–preconditioned BiCGstab on one processor of a SunFire 15k: natural `amatos` numbering (original), reverse Cuthill-McKee (RCM), quotient minimum degree (QMD), reverse space filling curve (rSFC).

ordering	original	RCM	QMD	rSFC
# iterations	66	22	20	12
solution time	355,0 s	94,5 s	92,7 s	77,9 s

factor of 10^{-13} . Original and rSFC ordering are provided by the calling routine, while RCM and minimum degree ordering are calculated by the PETSc solver.

4 Examples

4.1 Advection Model for Tracer Transport

First experiments with adaptivity in atmospheric modelling (Behrens et al., 2000) allowed to model tracer advection in the polar vortex with high resolution on a workstation computer. Snapshots of the adaptive grid are depicted in fig. 7.

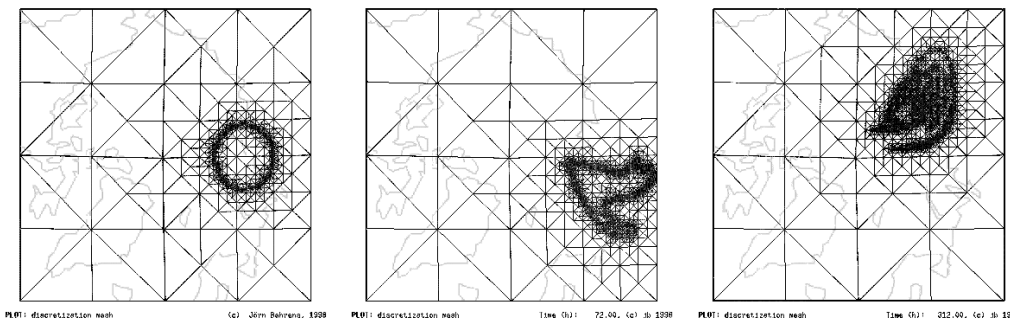


Fig. 7. Snapshots from an adaptive model (5 km minimal resolution) for the advection of tracer in a simulated wind field of the polar vortex.

4.2 PLASMA

In order to achieve reliable assessments of future climate development and the impact of anthropogenic influence an improved understanding of natural climate variability on time-scales from seasons to decades is required. For this, nonlinear interactions between processes at different spatial and temporal scales play a fundamental role. The realistic representation of these multi-scale

interactions is important not only for climate modelling but also for numerical weather prediction. To study them, the application of adaptive modelling techniques in atmospheric flow appears natural. Therefore, we applied these techniques by using the grid generator `amatos` for the development of the atmospheric model PLASMA (Parallel Large-scale Self-adaptive Model of the Atmosphere).

As a first step, a barotropic version of the model has been implemented, based on the shallow-water equations (SWE). The shallow water equations describe many of the physical phenomena that are represented in the full set of primitive equations and commonly serve as a test case for new numerical methods, because they present the major difficulties found in the horizontal discretization of three-dimensional equations.

The SWE describe the horizontal flow within a thin atmospheric layer of an hydrostatic gas with constant density. In the current implementation of PLASMA we use a scalar formulation of the SWE. This formulation has been derived from the vector formulation of the SWE on the sphere $S \subset \mathbb{R}^3$, as given in Côté (1988) by applying rot_S and div_S to the momentum equation. By defining vorticity $\zeta := \text{rot}_S \mathbf{u}$, divergence $\delta := \text{div}_S \mathbf{u}$, and the static underlying orography $\Phi_0 : S \rightarrow \mathbf{R}$, combining with the Helmholtz decomposition for time T , we obtain the following scalar formulation of the SWE

$$\begin{aligned}
\partial_t \zeta + \mathbf{u} \cdot \nabla_S \zeta + \zeta \delta + f \delta &= -\mathbf{u} \cdot \nabla_S f, \\
\partial_t \delta + \mathbf{u} \cdot \nabla_S \delta + \Delta_S \Phi - f \zeta &= -(\mathbf{n} \times \mathbf{u}) \cdot \nabla_S f - J(\mathbf{u}), \\
\partial_t \Phi + \mathbf{u} \cdot \nabla_S \Phi + \Phi \delta - \Phi_0 \delta &= \mathbf{u} \cdot \nabla_S \Phi_0, \\
-\Delta_S \psi &= \zeta, \\
\Delta_S \chi &= \delta, \\
\mathbf{rot}_S \psi + \nabla_S \chi &= \mathbf{u}
\end{aligned} \tag{1}$$

The function $\mathbf{u} : S \times (0, T) \rightarrow \mathbf{R}^3$ with $\mathbf{u} \cdot \mathbf{n} = 0$ denotes the horizontal velocity field and $\Phi : S \times (0, T) \rightarrow \mathbf{R}$ the geopotential height. The spherical differential operators $\nabla_S, \Delta_S, \text{rot}_S, \text{div}_S, \mathbf{rot}_S$ on S have been defined according to Dziuk (1988). The functions $\psi, \chi : S \times (0, T) \rightarrow \mathbf{R}$ are the streamfunction and the velocity potential respectively and the functional J is defined as

$$J(\mathbf{u}) := \sum_{i,j,k,l=1}^3 (\nabla_S u_i)_k \delta_{i,l} \delta_{j,k} (\nabla_S u_j)_l + u_i u_j \text{div} \text{div} (\partial_i n_j \mathbf{n}).$$

For the numerical implementation of these spherical SWE the Lagrange-Galerkin method, a combination of two discretization methods, has been applied. The first is the semi-Lagrangian method, the discretization of the material derivative along trajectories, which has good stability properties (Süli, 1988). The second is the finite element method on the sphere (Dziuk, 1988) realized on an approximating polyhedron S_h consisting of a triangular grid whose nodes

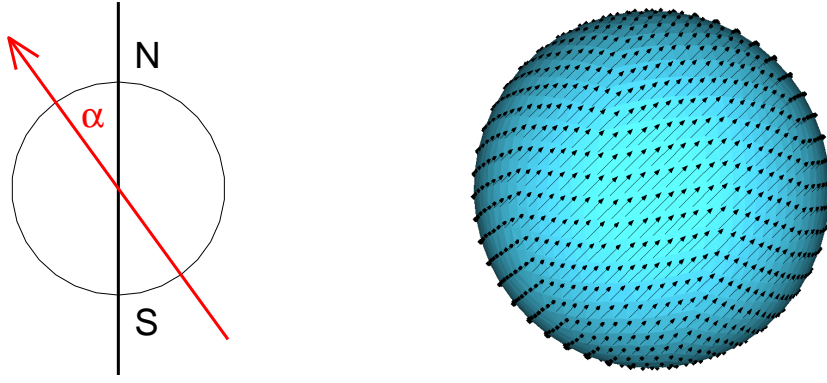


Fig. 8. Validation of PLASMA. Inclination angle α and initial condition for velocity field \mathbf{u} for the analytical solution (solid body rotation with $\alpha = \frac{\pi}{4}$)

lie on S . A detailed description of the numerical methods can be found in Lauter (2003).

Using amatos

For a first validation of the model an analytical solution is convenient, because the numerical error can be computed exactly. Here we consider two experiments, the solid body rotations with different rotation axes ($\alpha = 0$, $\alpha = \frac{\pi}{4}$). The initial conditions depending on the angle α , the inclination of the flows rotation axis to the earth's axis, see Fig. 8, are given by

$$\begin{aligned} \mathbf{u}(x) &= c \mathbf{a}(\alpha) \times x \quad \Leftrightarrow \quad \zeta(x) = c 2 \mathbf{a}(\alpha) \cdot x, \quad \delta(x) = 0, \quad \forall x \in S \\ \Phi(x) &= -\frac{1}{2} \left[(c \mathbf{a}(\alpha) \cdot x)^2 + 2 c \mathbf{a}(\alpha) \cdot x \boldsymbol{\Omega} \cdot x \right] + C, \quad \forall x \in S \end{aligned}$$

where

$$c := \frac{u_0}{R}, \quad \mathbf{a}(\alpha) := (-\sin(\alpha), 0, \cos(\alpha))^T.$$

$u_0 = 30 \frac{m}{s}$ is the maximal velocity, R the earth radius and C an arbitrary constant. With the underlying orography

$$\Phi_0(\mathbf{x}) = -\frac{\Omega^2}{2} + \frac{(\boldsymbol{\Omega} \cdot \mathbf{x})^2}{2}$$

we obtain a time periodic quasistationary flow.

Fig. 9 illustrates the relative $L^2(S_h)$ discretization error and shows, that the error decreases for increasing spatial and time resolution. Thus we have shown for these two experiments the experimental convergence of the numerical method.

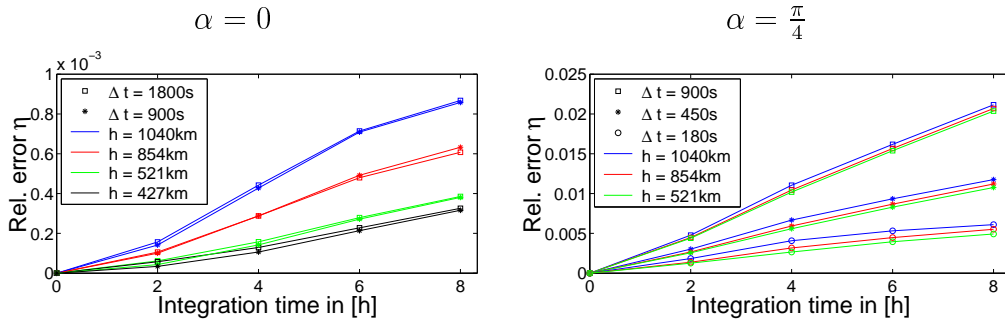


Fig. 9. Simulation with PLASMA. Relative discretization error $L^2(S_h)$ for the geopotential Φ

We have obtained first insights into the capability of the model to cope with more realistic atmospheric flow situations by studying the development of quasi-stationary planetary Rossby-waves, forced by orography. Here, we intended to reproduce quasi-stationary Rossby-waves by forcing a tropospheric westerly current with one localized mountain. The mountain of $480m$ height is situated at $30^\circ N, 0^\circ W$. The westerly current has its maximum of $30m/s$ at $30^\circ N$ with a meridional extension of 30° .

In Fig. 10, upper row, the orography and the initial geopotential height field for this test case is shown. Geopotential height fields after 20 day simulation time are displayed in the lower row of Fig. 10 for the simulation with a fixed grid and for the simulation with adaptive grid refinement. The grid refinement criteria is determined for every time step by the gradients of the vorticity and divergence.

Both simulations show the development of long Rossby waves forced by the isolated mountain. Due to the rather low height of the mountain, pronounced perturbations on smaller scales do not occur and the differences between the simulations with fixed and adaptive grids are rather small, but non-negligible.

These results encourage us to further investigate how the development and the variability of the quasi-stationary waves are influenced by the height of orography and especially by the two-way feedbacks between global and regional scales obtained with the adaptive mesh-refinement.

Here we demonstrated the successful application of the adaptive mesh generator `amatos` for the development and implementation of the global atmospheric model PLASMA which therefore allows an adequate representation of two-way feedbacks between global and regional scales. Applying new numerical methods and parallelisation techniques ensure efficient integrations of PLASMA on state-of-the-art high performance computing architectures. We have demonstrated the numerical convergence of the model and its capability to simulate the most important feature of the extratropical large scale circulation, the

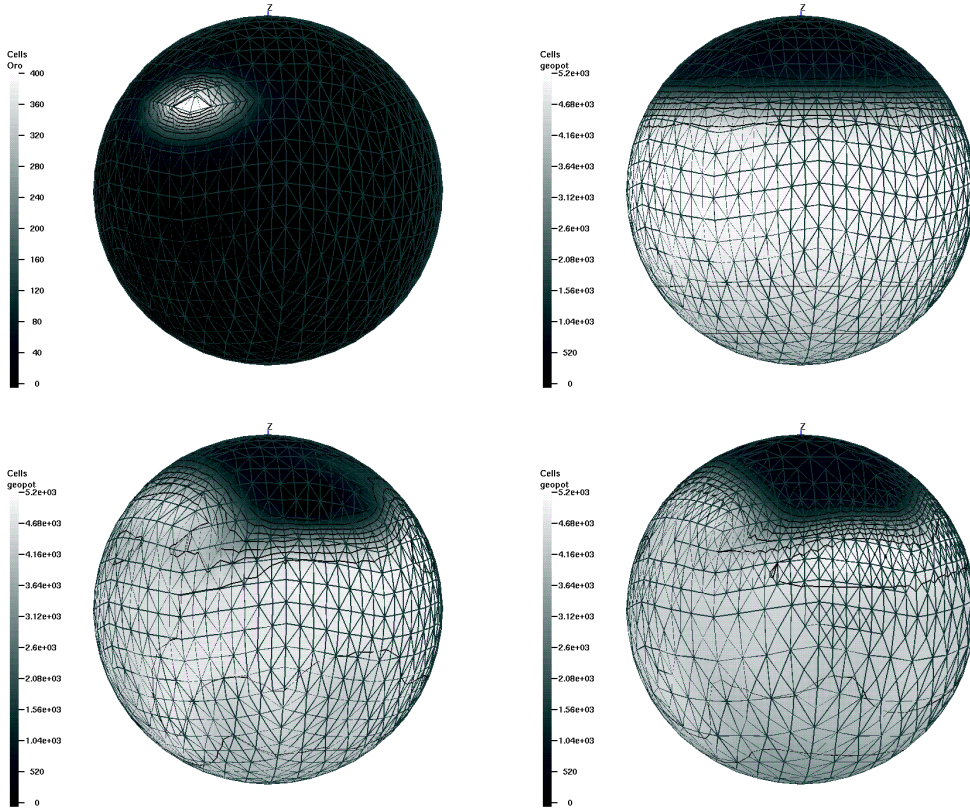


Fig. 10. Simulation of Rossby-waves with PLASMA. Upper row: Orography (left) and initial geopotential height field (right). Lower row: Geopotential height fields after 20 days for fixed mesh, mesh width $\approx 750\text{km}$ (left) and for adaptive mesh, mesh widths between 400 and 1700km (right).

quasi-stationary Rossby waves. In future, special emphasis will be put on the influence of the representation of two-way feedbacks on the variability of the quasi-stationary waves and therefore on climate variability on time-scales from seasons to decades.

4.3 *amatos* in Oceanography

From the grid generation point of view, the difference between atmospheric and oceanographic modelling is the highly irregular geometry of ocean basins.

amatos refines a given grid by bisecting the triangles without correcting elements at the boundary with respect to a given coastline, i.e., if a curved coastline is approximated by one element edge on the coarsest grid level, the new vertices on finer grid levels will all lie on this chord and will not be moved to the curved coast. It is planned to extend *amatos* for smooth geometries, such that new vertices at the boundary can be moved to a given coastline,

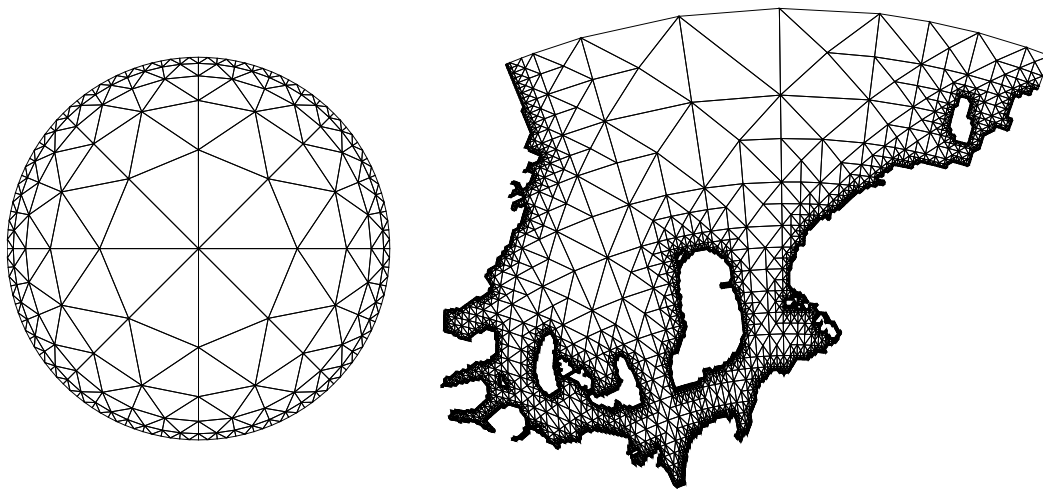


Fig. 11. Examples for initial grids that resolve the boundary

but a more general approach for arbitrary coastlines is not feasible. A remedy is to provide `amatos` with an initial grid that already resolves the coastline (and bottom topography in the 3D case) with the finest desired resolution as shown in the examples in fig. 11. This approach also avoids the question of how to deal with the computational domain changing during a model run.

5 Outlook

An ongoing project is the concise validation and code optimization of `amatos` version 2.0. The support of arbitrary finite element types introduced more data nodes which in turn made considerable extensions of the SFC support and the routines for data retrieval necessary.

The further plans reflect the needs of applications carried out with `amatos`. In the range of the PLASMA project, error estimators based on mathematical and physical criteria are developed.

A major concern are sparse linear solvers. As `amatos` is parallelized with OpenMP while state of the art parallel sparse solver packages are based on MPI, we had to find a way to establish efficient communication between OpenMP-threads and MPI-tasks (Rakowsky et al., 2003). However, a pure OpenMP library of sparse solvers remains desirable and we have started to implement the most promising domain decomposition solvers in OpenMP (Frickenhaus et al., 2003). Another promising approach is to make use of `amatos`' hierarchical data structure in a multigrid solver, e.g., a cascading multigrid solver (Bornemann and Krause, 1998).

Acknowledgement

The authors would like to thank Thomas Heinze for fruitful discussions and hints regarding the spherical version of `amatos`.

The PLASMA project and the development of `amatos` in this context is funded by the German Ministry for Education and Research in the range of the german climate research programme DEKLIM.

References

- Balay, S., Buschelman, K., Eijkhout, V., Gropp, W. D., Kaushik, D., Knep-
ley, M., McInnes, L. C., Smith, B. F., Zhang, H., 2003. PETSc users man-
ual. Tech. Rep. ANL-95/11 – Revision 2.1.6, Argonne National Laboratory,
<http://www.mcs.anl.gov/petsc>.
- Behrens, J., 1996. Adaptive Semi-Lagrange-Finite-Elemente-Methode zur
Lösung der Flachwassergleichungen: Implementierung und Parallelisierung.
Berichte zur Polarforschung 217, Alfred-Wegener-Institut, Bremerhaven,
PhD thesis, Universität Bremen.
- Behrens, J., July 2003. `amatos` Documentation. TU München, [http://www-
m3.mathematik.tu-muenchen.de/m3/software/amatos/index.html](http://www-m3.mathematik.tu-muenchen.de/m3/software/amatos/index.html).
- Behrens, J., Dethloff, K., Hiller, W., Rinke, A., 2000. Evolution of small scale
filaments in an adaptive advection model for idealized tracer transport.
Mon. Wea. Rev. 128, 2976–2982.
- Behrens, J., Zimmermann, J., 2000. Parallelizing an unstructured grid genera-
tor with a space-filling curve approach. In: Bode, A., Ludwig, T., Karl, W.,
Wismüller, R. (Eds.), Euro-Par 2000 Parallel Processing, 6th International
Euro-Par Conference, Munich, Germany, August/September 2000, Proceed-
ings. Vol. 1900 of Lecture Notes in Computer Science. Springer-Verlag, pp.
815–823.
- Bornemann, F. A., Krause, R., 1998. Classical and cascadic multigrid — a
methodological comparison. In: Bjørstadt, P., Espedal, M., Keyes, D. (Eds.),
Proceedings of the 9th International Conference on Domain Decomposition
Methods 1996, Ullensvang, Norway. Domain Decomposition Press, Bergen,
pp. 64–71.
- Frickenhaus, S., Hiller, W., Best, M., 2003. FoSSI: family of simplified solver
interfaces for parallel sparse solvers in numerical atmosphere and ocean
modeling. *Ocean Modelling* Submitted.
- Karypis, G., Kumar, V., 1998. METIS a software package for partitioning
unstructured graphs, partitioning meshes, and computing fill-reducing or-
derings of sparse matrices, version 4.0. User guide, University of Minnesota,
Department of Computer Science / Army HPC Research Center, Minneapo-
lis, <http://www-users.cs.umn.edu/~karypis/metis>.

- Rakowsky, N., Frickenhaus, S., Hiller, W., Läuter, M., Handorf, D., Dethloff, K., 2003. A self-adaptive finite element model of the atmosphere. In: Zwiefelhofer, W., Kreitz, N. (Eds.), ECMWF Workshop on the Use of High Performance Computing in Meteorology: Realizing TeraComputing, 4–8 November 2002, Reading UK. ECMWF, World Scientific, Singapore, pp. 279–293.
- Zumbusch, G. W., 2001. On the quality of space-filling curve induced partitions. *Z. Angew. Math. Mech.* 81 (Suppl. 1), 25–28, also as report SFB 256, Universität Bonn, no. 674.