Requirements and problems in parallel model development at DWD

Ulrich Schättler, Günther Doms and
Jürgen Steppeler
*Deutscher Wetterdienst, Postfach 100465, 63004
Offenbach, Germany
Tel.: + 49 69 8062 2739; Fax: + 49 69 8062 3721;
E-mail: ulrich.schaettler@dwd.de*

Nearly 30 years after introducing the first computer model for weather forecasting, the *Deutscher Wetterdienst* (DWD) is developing the 4th generation of its numerical weather prediction (NWP) system. It consists of a global grid point model (GME) based on a triangular grid and a non-hydrostatic *Lokal Modell* (LM). The operational demand for running this new system is immense and can only be met by parallel computers.

From the experience gained in developing earlier NWP models, several new problems had to be taken into account during the design phase of the system. Most important were portability (including efficieny of the programs on several computer architectures) and ease of code maintainability. Also the organization and administration of the work done by developers from different teams and institutions is more complex than it used to be.

This paper describes the models and gives some performance results. The modular approach used for the design of the LM is explained and the effects on the development are discussed.

1. Introduction

In 1996 DWD started to develop the 4th generation of its NWP system. The current 3rd generation operational system consists of a spectral *Global Modell* (GM), a regional grid point model for the synoptic and meso-α scale covering the Northern Atlantic and Europe (the *Europa-Modell* EM) and a high resolution meso-β scale *Deutschland-Modell* (DM). EM and DM are running the same code but with different domain sizes and horizontal and vertical resolutions.

In the new system, GM and EM are replaced by a global grid point model GME with physical packages based on the EM/DM. It must produce global forecasts for up to seven days which either match or surpass the quality of the EM. The hydrostatic DM will be replaced by a nonhydrostatic *Lokal Modell* (LM), which will be used for numerical weather prediction on the meso-β and on the meso-γ scale as well as for the evaluation of local climate and for various scientific applications covering a wide range of spatial scales (down to grid spacings of about 100 m). The weather forecasts include clouds, fog, precipitation, local wind systems and also severe weather events. The whole system will be used as a simulation and research tool for applications such as parameterizations, data assimilation and climate investigations. For the development of both models collaborations have been started with several national and international research institutes and universities.

The initial resolutions of the models for NWP ($\sim$ 55 km horizontal for GME with 31 levels and $\sim$ 8 km for LM with 35 levels) will be increased in the next years (to $\sim$ 25 km for GME with 40 levels and $\sim$ 2–3 km for LM with 50 levels) demanding a computational power of about 300×10^{12} floating point operations for a 24 hour forecast for each model. To meet these requirements, GME and LM have been parallelized and implemented for distributed memory parallel computers using Standard Fortran 90 and the Message Passing Interface (MPI) as a parallel library. But they can still be executed on conventional scalar and vector computers where MPI is not available.

At these performance levels, efficiency of the models is extremely dependent on the underlying hardware. Changes to computer and processor architectures in the past have forced model developers to a complete restructuring and recoding of their codes. With the rapid development of computers in mind it can be foreseen that the frequency of such updates will increase in the future. On the other hand it is not clear today which computer or processor architecture will be the most promising or affordable in roughly 3–5 years time. Beyond the well known requirements of code maintainability and efficiency on a particular computer system, portability and efficiency on a wide range of different

ISSN 1058-9244 / $8.00 © 2000, IOS Press. All rights reserved

computer systems and architectures must be accounted for. At the same time the program design should also allow for easy code modifications to react not only to changes in computer hardware but also to new scientific developments.

This paper reports on the development progress achieved so far at the DWD. Section 2 gives the basic features and parallelization strategies of both models as well as some performance results. The modular approach used for the design of the LM is described in Section 3. The effect of the modularity on the development work is discussed. Future requirements for running higher resolution models and problems regarding computer architecture and programming style are presented in Section 4.

2. Description of the models

Detailed scientific documentation is available for both models [1–3]. Therefore, only some basic features will be given here. A more comprehensive summary can be found in [4].

2.1. The nonhydrostatic regional model LM

2.1.1. Equations, algorithms and grid structure

The model is based on the set of governing equations for a nonhydrostatic fully compressible atmosphere. Introducing a spherical coordinate system (λ, ϕ) with a rotated pole and a generalized terrain-following vertical coordinate ζ the prognostic equations for momentum (u, v, w), perturbation pressure p', temperature T and moisture q take the form

$$\begin{aligned}
&\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u - \frac{uv}{a} \tan\varphi - fv = \\
&\qquad -\frac{1}{\rho a \cos\varphi} \left(\frac{\partial p'}{\partial \lambda} + \frac{J_\lambda}{\sqrt{G}} \frac{\partial p'}{\partial \zeta} \right) + M_u, \\
&\frac{\partial v}{\partial t} + \mathbf{v} \cdot \nabla v + \frac{u^2}{a} \tan\varphi + fu = \\
&\qquad -\frac{1}{\rho a} \left(\frac{\partial p'}{\partial \varphi} + \frac{J_\varphi}{\sqrt{G}} \frac{\partial p'}{\partial \zeta} \right) + M_v, \\
&\frac{\partial w}{\partial t} + \mathbf{v} \cdot \nabla w = \frac{1}{\rho \sqrt{G}} \frac{\partial p'}{\partial \zeta} + B + M_w, \qquad (1) \\
&\frac{\partial p'}{\partial t} + \mathbf{v} \cdot \nabla p' - g\rho_0 w = -(c_{pd}/c_{vd}) pD \\
&\qquad +(c_{pd}/c_{vd} - 1) \rho c_{pd} Q_T, \\
&\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T = \frac{1}{\rho c_{pd}} \left(\frac{\partial p'}{\partial t} + \mathbf{v} \cdot \nabla p' - g\rho_0 w \right) \\
&\qquad +Q_T, \\
&\frac{\partial q^v}{\partial t} + \mathbf{v} \cdot \nabla q^v = -(S^l + S^f) + M_{q^v}, \\
&\frac{\partial q^{l,f}}{\partial t} + \mathbf{v} \cdot \nabla q^{l,f} + \frac{1}{\rho \sqrt{G}} \frac{\partial P_{l,f}}{\partial \zeta} = S^{l,f} + M_{q^{l,f}},
\end{aligned}$$

with the buoyancy term

$$\begin{aligned}
&B = g \frac{\rho_0}{\rho} \\
&\left\{ \frac{T - T_0}{T} - \frac{p' T_0}{p_0 T} + \left(\frac{R_v}{R_d} - 1 \right) q^v - q^l - q^f \right\},
\end{aligned} \qquad (2)$$

the advection operator

$$\begin{aligned}
\mathbf{v} \cdot \nabla = &\tfrac{1}{a\,\cos\varphi} \left(u \tfrac{\partial}{\partial \lambda} + v \cos\varphi \tfrac{\partial}{\partial \varphi} \right) \\
&+ \dot{\zeta} \tfrac{\partial}{\partial \zeta},
\end{aligned} \qquad (3)$$

the contravariant vertical velocity

$$\dot{\zeta} = \frac{1}{\sqrt{G}} \left(\frac{J_\lambda}{a \cos\varphi} u + \frac{J_\varphi}{a} v - w \right), \qquad (4)$$

and the three-dimensional wind divergence

$$\begin{aligned}
D = &\frac{1}{a \cos\varphi \sqrt{G}} \left\{ \frac{\partial}{\partial \lambda} \left(\sqrt{G} u \right) + \frac{\partial}{\partial \zeta} (J_\lambda u) \right. \\
&\left. + \frac{\partial}{\partial \varphi} \left(\sqrt{G} v \cos\varphi \right) + \frac{\partial}{\partial \zeta} (J_\varphi v \cos\varphi) \right\} (5) \\
&- \frac{1}{\sqrt{G}} \frac{\partial w}{\partial \zeta}.
\end{aligned}$$

Here subscript 0 denotes the reference state and primes the deviations from the reference state. The constants represent radius of the earth (a), heat capacity of dry air at constant pressure (c_{pd}), Coriolis parameter (f), acceleration due to gravity (g), gas constants for vapour (R_v) and dry air (R_d) and the ratio of heat capacities for air at constant pressure to constant volume (c_{vd}). The remaining terms describe the diabatic heating rates, the cloud microphysical sources for the moisture components q^k and the divergence of the turbulent and the diffusive fluxes.

Spatial discretization is by standard second order finite difference schemes on a C-/Lorenz-grid. The time integration is performed with the leapfrog-method using the Klemp and Wilhelmson [5] time splitting scheme including extensions proposed by Skamarock and Klemp [6] to solve for the sound and gravity wave terms. The basic idea behind time-splitting is to treat the fast terms describing sound and gravity wave propagation with small time steps Δt_s while using a large step Δt for the slow terms (advection, physics). The terms responsible for the fast modes are integrated with a small time step Δt_s whereas the slow mode tendencies are integrated using the large leapfrog time step. Thus, the algorithm can be more efficient than a fully explicit time-stepping scheme.

The split-explicit method is implemented with an implicit Crank-Nicolson method in the vertical and an explicit forward-backward scheme in the horizontal. As an alternative to this scheme, the new forward-in-time splitting method proposed by Wicker and Skamarock [7] has been implemented in the LM by Wicker. This new scheme is based on second-order Runge-Kutta time integration combined with third-order upwind-biased advection and with the forward-backward scheme for the fast modes. A semi-implicit scheme is currently being tested for the LM by Thomas [9] with very promising first meteorological results. The elliptic PDE for this scheme is solved with a generalized minimal residual algorithm. Whether this method is a computational alternative must be further investigated. Saito [8] has investigated similar questions with the split-explicit version of LM and the mesoscale nonhydrostatic model of the Meteorological Research Institute (Japan).

The physics package of LM has been adapted from the operational hydrostatic DM and thus only applies on the meso-β but not on smaller scales. Work on new parameterization schemes to upgrade the physics for model applications on smaller scales is in progress. A diabatic digital filtering initialization scheme (Lynch, [10]) has been implemented to reduce noise and spin-up resulting from non-balanced interpolated data, which are used to drive the LM.

2.1.2. *Parallelization*

The parallelization strategy for the LM is the 2D domain or data decomposition (grid partitioning) which is well suited for grid point models using finite differences. This strategy also is used and described by several other authors [11–14]. Each processor gets an appropriate part of the data to solve the model equations on its own subdomain. These subdomains are arranged in a two-dimensional array of rectangular tiles. The local data structure of every processor contains additional rows and columns to store the values of grid points belonging to neighboring processors (see Fig. 1). During the integration step each processor updates the values of its local subdomain; grid points on the edges are exchanged using explicit message passing.

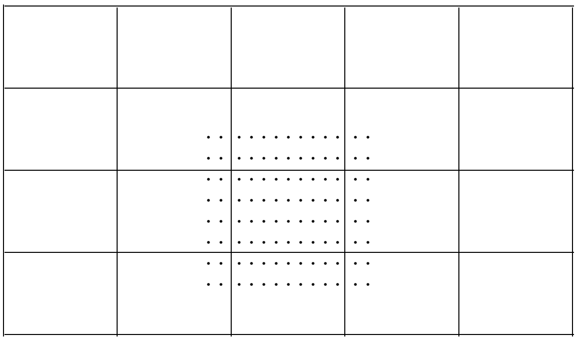


Fig. 1. 2D domain decomposition with local data structure.

All algorithms described above only require a nearest-neighbor exchange, i.e. local communications. The only exception is the minimal residual algorithm to solve the PDE in the semi-implicit scheme. For the computation of global dot-products, a global summation over all subdomains is necessary in addition to the local communication.

2.2. *The new global model GME*

2.2.1. *Equations and algorithms*

The system of equations solved in the GME is based on the hydrostatic primitive equations and is given in differential form for local spherical coordinates (η, χ) and a hybrid vertical coordinate ξ as follows.

$$\begin{aligned}
\frac{\partial u}{\partial t} - (\zeta + f)\, v + \dot{\xi}\frac{\partial u}{\partial \xi} &= -\frac{1}{a}\frac{\partial}{\partial \eta}(\Phi + K) \\
&\quad - \frac{RT_v}{a}\frac{\partial}{\partial \eta}(\ln p) + \left(\frac{\partial u}{\partial t}\right)_{sub} - K_4 \nabla^4 u, \\
\frac{\partial v}{\partial t} + (\zeta + f)\, u + \dot{\xi}\frac{\partial v}{\partial \xi} &= -\frac{1}{a}\frac{\partial}{\partial \chi}(\Phi + K) \\
&\quad - \frac{RT_v}{a}\frac{\partial}{\partial \chi}(\ln p) + \left(\frac{\partial v}{\partial t}\right)_{sub} - K_4 \nabla^4 v, \\
\frac{\partial T}{\partial t} + \frac{u}{a}\frac{\partial T}{\partial \eta} + \frac{v}{a}\frac{\partial T}{\partial \chi} + \dot{\xi}\frac{\partial T}{\partial \xi} &= \frac{\alpha\omega}{c_p} + \frac{L_v}{c_p} C_{vl} \\
&\quad + \left(\frac{\partial T}{\partial t}\right)_{sub} - K_4 \nabla^4 (T - T_{ref}), \qquad (6) \\
\frac{\partial p_s}{\partial t} &= -\frac{1}{a}\int_0^1 \left\{\frac{\partial}{\partial \eta}\left(u\frac{\partial p}{\partial \xi}\right) + \frac{\partial}{\partial \chi}\left(\frac{\partial p}{\partial \xi}\right)\right\} d\xi, \\
\frac{\partial q_v}{\partial t} + \frac{u}{a}\frac{\partial q_v}{\partial \eta} + \frac{v}{a}\frac{\partial q_v}{\partial \chi} + \dot{\xi}\frac{\partial q_v}{\partial \xi} &= -C_{vl}
\end{aligned}$$

$$+\left(\frac{\partial q_v}{\partial t}\right)_{sub} - K_4\nabla^4 q_v,$$

$$\frac{\partial q_l}{\partial t} + \frac{u}{a}\frac{\partial q_l}{\partial \eta} + \frac{v}{a}\frac{\partial q_l}{\partial \chi} + \dot{\xi}\frac{\partial q_l}{\partial \xi} = C_{vl}$$

$$+\left(\frac{\partial q_l}{\partial t}\right)_{sub},$$

where u and v are the zonal and meridional wind components, T is the temperature, p_s is the surface pressure, q_v is the specific water vapor content, q_l is the specific cloud liquid water content, ζ is the vorticity and K the specific kinetic energy. p is the pressure, T_v is the virtual temperature, T_{ref} is a reference temperature depending only on height. C_{vl} is the condensation rate and $(\ldots)_{sub}$ are the sub-grid scale tendencies due to parameterized processes.

2.2.2. Grid generation

For the horizontal discretization of the equations a triangular grid based on the icosahedron is introduced. It was first described by Sadourny et al. [15] and Williamson [16]. The approach outlined here is based on the work of Baumgardner [17]. The same grid also is used by Loft [18].

To construct the grid, the sphere is divided into 20 spherical triangles of equal size by placing a plane icosahedron into it. The 12 vertices of the icosahedron touch the sphere, one vertex coincides with the north pole and the opposite one with the south pole. The spherical triangles are defined by the great circles connecting two vertices respectively. Each of the 12 vertices then is surrounded by 5 spherical triangles. Two adjacent triangles are combined to form a “diamond”, i.e. a logically square block.

For further grid generation, the sides of the 20 main triangles are subdivided iteratively into ni equal parts to form subtriangles. Each point in a main triangle is surrounded by six triangles and accordingly is the center of a hexagon. However, the points which form the vertices of the icosahedron are surrounded by only five triangles and therefore are the centers of pentagons. Some resulting grids are illustrated in Fig. 2.

The derivation of the necessary numerical operators (e.g. for the gradient, the divergence or the Laplacian) for this triangular grid as well as a more detailed explanation of the grid generation can be found in the documentation of the GME [2].

2.2.3. Parallelization

The diamonds can be looked upon as logical square blocks and therefore can be implemented with normal data structures. In the sequential program a global two-dimensional field is stored as a three-dimensional array. The third dimension represents the 10 different diamonds covering the earth.

The parallelization strategy is by data decomposition again. But while this is straightforward for a regional model a more sophisticated strategy must be used here. A practical way for the parallelization is based on the viewpoint that every diamond can be regarded as a regional model and is related to an idea of John Baumgardner. Every diamond can be partitioned using a two-dimensional decomposition in the same way like the domain of a regional model. Since all diamonds are of equal size, their decomposition is identical. Every processor then gets a part of each diamond, i.e. it computes the forecast in a subdomain of all ten diamonds.

Other decompositions of this triangular grid that minimize the amount of data to be transferred have been investigated by GMD (German National Research Center for Information Technology) [19,20]. But within the decomposition described above the 10 parts of each diamond that a processor obtains are distributed regularly over the earth. From a statistical point of view there is a chance to get a rather balanced load distribution, regarding the computations in the physical packages (day-night radiation, land-water distribution).

2.2.4. Numerical solution

The two moisture equations for q_v and q_l are solved using a semi-Lagrangian advection scheme in the horizontal direction to allow for monotonicity and positive definiteness. For u, v, T and p_s a semi-implicit Eulerian method is applied which runs about 20% faster compared to the semi-Lagrangian version at the same time step.

The three-dimensional Helmholtz equation, that must be solved for the semi-implicit method, is decomposed into n_k (the number of vertical layers) two-dimensional equations by diagonalizing the system with the eigenvectors of the vertical structure matrix. Only the external and the first four internal modes are solved (split semi-implicit approach), the other $n_k - 5$ modes are treated explicitly because the corresponding phase velocities are smaller than the advection speed. The two-dimensional equations are solved by a Gauß-Seidel algorithm which takes about 15 iterations for the external mode and 3 iterations for the internal ones.

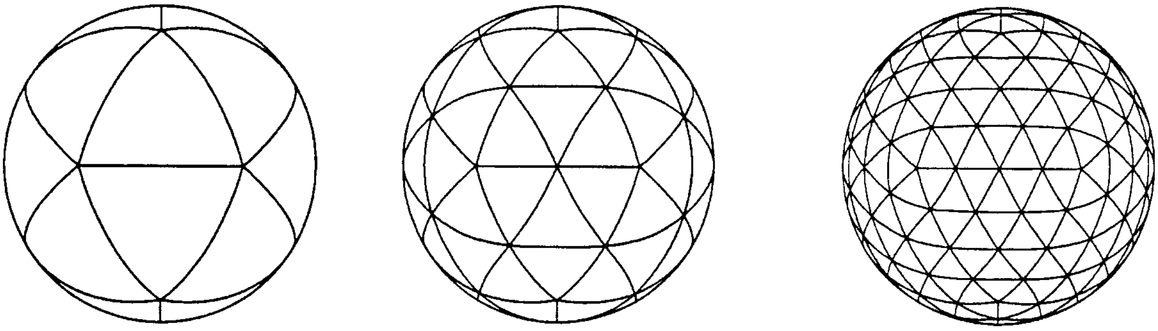


Fig. 2. Grids derived from the icosahedron.

The current version of GME is restricted to Courant numbers $C < 1$. Otherwise the parcel trajectory could depart outside the surrounding triangles and in the parallel program outside the subdomain of a processor. This would require a more complex communication pattern in the search algorithms in case of $C > 1$.

2.3. Parallel performance of the models

Numerical tests investigating the parallel performance of LM and GME have been performed on DWD's SGI/Cray T3E1200 with up to 256 processors. The T3E has DEC Alpha 21164 processors with 128 MByte memory running at 600 MHz (with 1200 MFlop/s peak performance). Each processor has 6 stream buffers which consist of additional hardware between the memory and the CPU to improve the memory bandwidth for vector-like data references. The processors are connected in a three-dimensional torus where each processor has a direct link to all of its neighbors, capable of sustaining a transferrate of 480 MByte/s.

The two aspects of performance that are of special interest for parallel programs are single node performance and the scaling, i.e. how the single nodes work together.

Regarding the computations, single node performance is a critical issue on the T3E due to the memory design of the processors (cache architecture) and some time has been spent to optimize LM and GME especially for exploiting the stream buffers. This was done by splitting computation intensive loops which would have used more than the 6 existing streams into smaller loops reducing the number of streams and avoiding stream "thrashing". The loop splitting was mostly done manually but also by using compiler directives. The compiler option `-O split2`, which tries to split every loop, was not used because it lowered the overall performance. The compiler options used for both models are `-O3 -O aggress,unroll2,split1,pipeline2`. Especially the unrolling and the pipelining were found to improve the performance significantly.

On the other hand, no optimizations were necessary for the communications due to the fast interconnection network. In the LM, only about 3% of the elapsed time is needed for the communication.

For the scaling the communication/computation ratio therefore plays only a minor role. More important are the distribution of the workload and whether there are parts of the program that are not or cannot be parallelized. In LM and GME input and output are sequential parts, because all data are routed through processor 0. Using striped disks and the Flexible File I/O (FFIO) from SGI/Cray, the achieved IO-rates (80 MByte/s for output, but only 20 MByte/s for input, depending on the workload of the machine) are sufficing for the moment. When the model size is increased in the next years, parallel asynchronous I/O will be implemented. The use of FFIO is encapsulated in a special I/O-library which also contains the routines for encoding and decoding GRIB files and therefore does not disturb the portability of the models.

The timings for both models are displayed in Fig. 3. For the LM a 6 hour forecast with a small grid size (109×109 grid points and 20 vertical layers) and a grid size corresponding to the initial resolution used for NWP (325×325 and 35 vertical layers) has been run with full I/O. For the big grid this means that 46 MByte have been read and 100 MByte have been written to disk per forecast hour. For the small grid the speedup only decreases for $\geqslant 64$ processors. This is partly because the subdomains become too small and the com-

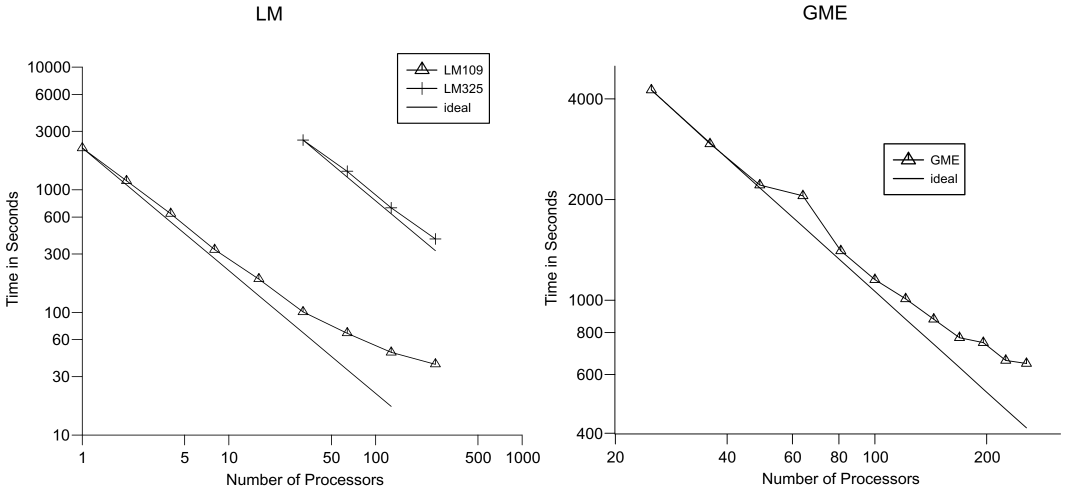


Fig. 3. Parallel performance of the models.

munication/computation ratio increases and partly because of a more imbalanced distribution of the workload in the physics. However, doubling the number of processors 3 or 4 times, the execution time decreases almost linearly in Fig. 3 for the bigger grid. On this problem an overall performance of 60 MFlop/s (using 64 processors) and 52 MFlop/s (using 256 processors) was achieved per processor. Regarding only the computations, these rates range between 68 and 61 MFlop/s for the dynamics and between 90 and 83 MFlop/s for the physics.

The GME has been run with $ni = 128$ and 31 vertical layers, also corresponding to the initial resolution used for NWP. A 24 hour forecast has been performed starting with real data but without writing GRIB files. For these tests the number of processors employed is always a power of two, giving the best communication performance at the boundary of the diamonds. For less than 100 processors an almost linear speedup can be realized for the GME. When using $n^2 = 64$ processors, the size of the largest subdomain is not signifanctly smaller than when using $(n - 1)^2$ processors. This results in a bad distribution of the workload and the reduction of the execution time is not very significant. Similar effects can also be seen for $\geqslant$ 180 processors.

3. Code design for the LM

3.1. Modular design

Modularity is a basic attribute of NWP models, but in programming languages such as Fortran 77 this is difficult to express in the program design. Fortran 90 supports a modular development approach by grouping together variable declarations and subprograms into `MODULE`s.

The LM uses `MODULE`s in three different ways:

- The data modules form the data pool of the model (meteorological as well as organizational variables). With the Fortran 90 `USE`-statement these data are available for other modules. The data modules replace the `COMMON`-blocks used in Fortran 77.
- The second group of modules provide utility routines that handle small tasks which need not be model specific. Examples are the time measurement, the determination of the actual date and time or the computation of meteorological variables derived from the prognostic variables. All routines necessary for parallel programming (i.e. routines containing calls to the message passing library) are also placed in utility modules.
- All routines belonging to a model specific task (or package) are combined in a source module. “Package” is a term defined by Kalnay [21] referring to the physical packages, i.e. the parameterization of the atmospheric subgrid-scale physical processes such as radiation or convection. More generally, other parts of the model (dynamics, input and output of data) can be viewed as packages. By using the data and utility modules, the source modules belonging to these packages can be written in such a way that they are independent from each other.

Every module has to list the data and the routines used from other modules. These lists define clearly the interface of the module. Figure 4 shows the modules used in the LM and their dependencies. The top level of the model is the main program `lmorg`. It manages all tasks of the forecast by using the source modules.

The clear modular formulation facilitates concurrent work on different (source) modules. For the ongoing development of the LM, this is very important, because most physical parameterizations have to be adapted to very high resolution in the near future. The work on different schemes can be done in parallel without slowing down other projects. At the same time, different numerical schemes for the dynamics can be investigated and tested.

3.2. *Portability*

One of the main goals for the source code development of the LM is portability. First of all this means that the same code must run on different computer platforms without having to change the source itself. This is achieved by using only standard Fortran 90 and MPI. MPI is adopted as a standard by nearly all computer vendors and efficient implementations are available for their parallel machines. For sequential platforms having no MPI implementation, dummy interfaces for the MPI routines are provided for the LM.

A second aspect of portability is that the program should also be efficient on different machines. The efficiency of the LM on vector processors is very good, because the code is written in the same way as former highly vectorized models (the inner-most loop is horizontal east-west direction). The coding style used for vectorization has some limitations for cache based scalar RISC processors. These can partly be overcome on the T3E by the streams and the optimizations described in Section 2.3. Other optimizations performed on the LM code so far are not hardware specific, but in a way that every processor architecture will benefit (avoid duplicate computations in different routines by providing more memory; avoid divisions, etc). With more optimizations the efficiency especially on the T3E can be enhanced, but as DWD's T3E will be replaced by a successor system in 2001, only little effort will be put in such a work. Also no optimizations will be done that degrade the performance on other machines severly.

Up to now the LM has been tested on Cray PVP machines, Cray T3E, SGI Origin 2000, IBM SP2, Fujitsu VPP700, NEC SX-4, a cluster of LINUX PCs and several workstations.

3.3. *Parallel programming*

As mentioned above, a portable parallel version of the LM is already available. Research and development is going on in the parameterizations, the dynamics, the assimilation scheme and related areas. The problem now faced is that most of the programmers involved in this work do not have much experience in parallel programming. Therefore, a strategy has been developed to enable them to work on the parallel LM and in a parallel computing environment.

The basic idea is that the computations in a subdomain are organized in the same way as the ones in the total domain, if working on a shared memory computer. The total domain includes a user-specified number of boundary lines (= `nboundlines`) surrounding the computational domain, where data are provided by a driving model. The forecast is computed only in the interior of the total domain. The same holds for every subdomain, with the exception that the values on the boundary lines (also `nboundlines` at each side) are provided by the neighboring processors via message passing.

Three different kinds of calculations have to be considered for the programming:

- Loop organization: The horizontal size of a subdomain is (`1`...`ie`,`1`...`je`). Start- and end-indices are provided, if values have to be calculated only in the interior part (`istart`...`iend`,`jstart`...`jend`). If values have to be calculated also on the boundaries, the loops range from `1`...`ie` and `1`...`je`, respectively. These values are set at the beginning of the program, according to the number of processors and the decomposition. Therefore, for most loop calculations there is no difference between the sequential and the parallel program.
- Grid point calculations: To perform computations on certain grid points, routines are provided to determine the local indices and the number of the subdomain in which a grid point is located from the global indices of the total domain and vice versa.
- Elemental parallel operations: Routines for special operations needing message passing are included in the utility modules. These are tools for computing e.g. mean or extreme values of the total domain as well as distributing values to or collecting them from the nodes.

The features described above allow programmers to work on special modules of the LM in a parallel en-

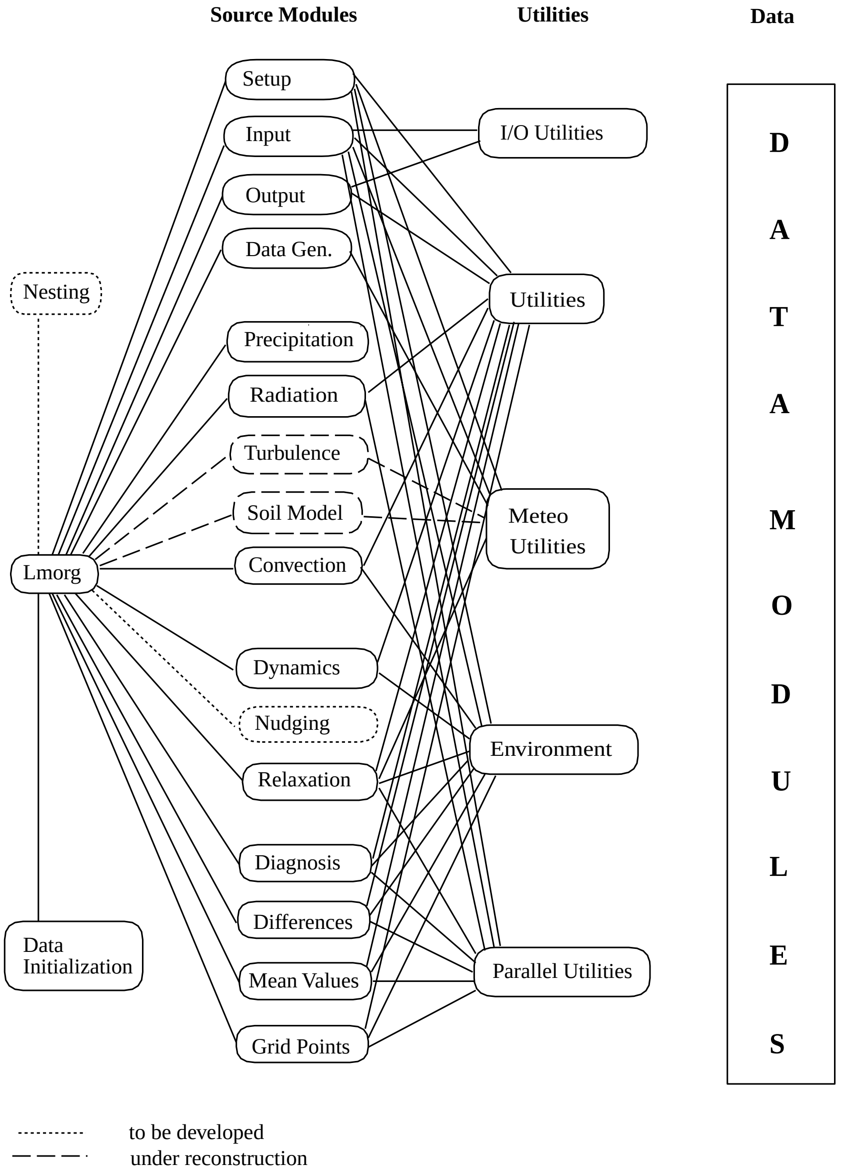


Fig. 4. Modular structure of the LM.

vironment without having much knowledge of parallelization. They are able to get the code running, but an experienced programmer might have to optimize the modules later on.

4. Future development issues

The LM will be the main forecast tool of DWD in the next decade. In 2001 it should run with a grid size of $800 \times 800 \times 50$ points and a time step of $\Delta t = 10s$. A 24 hour forecast must be completed during 30 minutes of wallclock time. For that purpose, the computing power at DWD will increase over the next years. The current SGI/Cray T3E1200 with 456 application processors will be expanded to a system with about 800 processors in 1999 and later on by a successor system, the architecture of which is not clear today.

A current trend on the hardware sector is the clustering of SMP (symmetric multi processing) systems. DWD now is concerned about the performance of the LM on machines with $\geqslant$ 1000 processors and about such SMP clusters. Also it is very important to know whether the programming style must be changed

Table 1
Predicted timings for different processor speeds

Processor	runtime (h)	dynamics %	physics %	MPI %	Efficiency
T3E600	4.78	64.83	33.64	1.53	0.86
T3E900	3.20	64.63	33.53	1.84	0.86
T3E4000	0.72	61.20	31.75	7.05	0.81

to fully exploit the two different connection systems (inter- and intranode communication) of clusters. Similar problems have been studied e.g. by [22] and [23].

These questions have been investigated on behalf of DWD by GMD and the software engineering company Pallas [24,25]. They constructed a run time model for the LM and predicted the performance on several partly non-existing computer architectures.

Table 1 shows predicted runtimes of the LM in the size described above for a 24 hour forecast on a 1024 processor T3E with different processor power (T3E600 with 600 MFlop/s peak performance, T3E900 and a fictitious T3E4000). Given are the runtime in hours and the percentages for the computations (in the dynamics and in the physics) and for the communications together with the parallel efficiency. The same interconnection network has been assumed for all processor types, therefore the percentage of the communication is higher for faster processors resulting in a decreased efficiency. Table 1 shows that the processor speed must be about 7 times faster than that of the T3E600 to compute a 24 hour forecast in half an hour.

One way to reach the desired speed within one processing element is the utilization of SMP nodes. As programming models for SMP clusters there are two major alternatives:

- Only message passing on all processors:
  This will be efficient, if the MPI implementation can fully exploit the speed of the shared-memory communication within one SMP-node. For the LM this model has the advantage, that no changes are necessary.
- Message passing on the cluster level and shared-memory programming within one node:
  The shared-memory programming could be done with automatic parallelization, which most compilers provide on loop level. The code could also be taken as it is today, but normally this is not very efficient. By using compiler directives, the efficiency will be better, but major changes to the code are necessary then. Another problem of this approach is the portability, but OpenMP could be a new standard for the parallelization with directives.

Again, the modular design of the LM would facilitate the adaptation to SMP-clusters using the shared-memory model, because an incremental parallelization is possible, starting with the most computing intensive modules.

Acknowledgements

The development of the models described in this article is the work of two groups at DWD. The authors, who are responsible for the development of the LM would like to thank the GME group (Dörte Liermann, Detlev Majewski, Peter Prohl, Bodo Ritter) for providing their material. The work on both models was enriched by several visits of international scientists at DWD.

References

[1] G. Doms and U. Schättler, The Nonhydrostatic Limited-Area Model LM (Lokal Modell) of DWD – Part I: Scientific Documentation, *Technical Report*, DWD, March 1997.

[2] D. Majewski, Documentation of the New Global Model GME, Deutscher Wetterdienst, 1996.

[3] D. Majewski, The New Global Icosahedral-hexagonal Grid Point Model GME of the Deutscher Wetterdienst, *ECMWF Seminar on Numerical Methods in Atmospheric Models* (to appear), 1998.

[4] U. Schättler and E. Krenzien, Model Development for Parallel Computers at DWD, *Making its Mark – Proceedings of the Seventh ECMWF Workshop on the Use of Parallel Processors in Meteorology*, G.-R. Hoffmann and N. Kreitz, eds., World Scientific, 1997, pp. 83–100.

[5] J.B. Klemp and R.B. Wilhelmson, The Simulation of Three-dimensional Convective Storm Dynamics, *Journal of the Atmospheric Sciences*, **35** (1978), 1070–1096.

[6] W. Skamarock and J.B. Klemp, The Stability of Time-splitting Methods for the Hydrostatic and Nonhydrostatic Elastic Systems. *Monthly Weather Review* **120** (1992), 2109–2127.

[7] L. Wicker and W. Skamarock, A Time Splitting Scheme for the Elastic Equations Incorporating Second-Order Runge-Kutta Time Differencing. Submitted to *Monthly Weather Review*.

[8] Kazuo Saito, Günther Doms, Ulrich Schättler and Jürgen Steppeler, 3-D Mountain Waves by the Lokal-Modell of DWD and the MRI Mesoscale Nonhydrostatic Model. *Papers in Meteorology and Geophysics* **49**(1) (1998), 7–19.

[9] S. Thomas and C. Girard, Semi-implicit Scheme for the DWD LM Model, *Technical Report*, DWD, August 1998.

[10] P. Lynch, The Dolphy-Chebyshev Window: A Simple Optimal Filter. *Monthly Weather Review* **125** (1997), 655–660.

[11] A. Dickinson, P. Burton, J. Parker and R. Baxter, Implementation and Initial Results from a Parallel Version of the Meteorological Office Atmosphere Prediction Model, *Coming of Age – Proceedings of the Sixth ECMWF Workshop on the Use of Parallel Processors in Meteorology,* G.-R. Hoffmann and N. Kreitz, eds., World Scientific, 1995, pp. 177–194.

[12] I. Foster and J. Michalakes, MPMM: A Massively Parallel Mesoscale Model, *Parallel Supercomputing in Atmospheric Science – Proceedings of the Fifth ECMWF Workshop on the Use of Parallel Processors in Meteorology,* G.-R. Hoffmann and T. Kauranne, eds., World Scientific, 1993, pp. 354–363.

[13] T. Kauranne, J. Oinonen, S. Saarinen, O. Serimaa and J. Hietaniemi, The Operational HIRLAM 2 Model on Parallel Computers, *Coming of Age – Proceedings of the Sixth ECMWF Workshop on the Use of Parallel Processors in Meteorology,* G.-R. Hoffmann and N. Kreitz, eds., World Scientific, 1995, pp. 63–74.

[14] U. Schättler and E. Krenzien, The Parallel "Deutschland-Modell" – A Message-Passing Version for Distributed Memory Computers, *Parallel Computing* **23** (1997), 2215–2226.

[15] R. Sadourny, A. Arakawa and Y. Mintz, Integration of the Nondivergent Barotropic Vorticity Equation with an Icosahedral-Hexagonal Grid for the Sphere, *Monthly Weather Review* **96** (1968), 351–356.

[16] D. Williamson, Numerical Integration of Fluid Flow over Triangular Grids, *Monthly Weather Review* **97** (1969), 885–895.

[17] J. Baumgardner, A Three-Dimensional Finite Element-Model for Mantle Convection, Ph. D. thesis, The University of California at Los Angeles, 1983.

[18] Richard D. Loft, A Modular 3-D Dynamical Core Testbed, *Making its Mark – Proceedings of the Seventh ECMWF Workshop on the Use of Parallel Processors in Meteorology,* G.-R. Hoffmann and N. Kreitz, eds., World Scientific, 1997, pp. 270–283.

[19] O. Bröker, K. Cassirer, R. Hess, C. Jablonowski, W. Joppich and S. Pott, Forschungs- und Entwicklungsarbeiten im Rahmen des neuen Global-Modells (GME) des DWD, Gesellschaft für Mathematik und Datenverarbeitung, Internes Arbeitspapier, 28. November 1996.

[20] O. Bröker, Laufzeitvorhersagen für parallele Versionen des globalen Wettermodells GME, Diplomarbeit, Rheinische Friedrich-Wilhelms-Universität Bonn, März 1998.

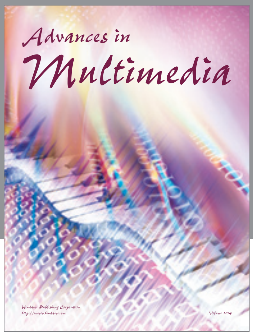
[21] E. Kalnay et.al., Rules for Interchange of Physical Parameterizations, *Bull. A.M.S.* **70** (1989), 620–622.

[22] Chris N. Hill and Andrew Shaw, Transitioning from MPP to SMP: Experiences with a Navier-Stokes solver, *Making its Mark – Proceedings of the Seventh ECMWF Workshop on the Use of Parallel Processors in Meteorology,* G.-R. Hoffmann and N. Kreitz, eds., World Scientific, 1997, pp. 250–269.

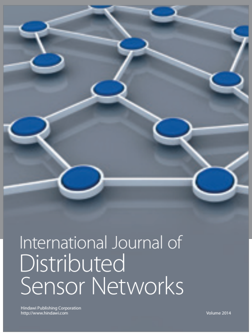
[23] Aaron C. Sawdey, Matthew T. O'Keefe and Wesley B. Jones, A General Programming Model for Developing Scalable Ocean Circulation Applications, *Making its Mark – Proceedings of the Seventh ECMWF Workshop on the Use of Parallel Processors in Meteorology,* G.-R. Hoffmann and N. Kreitz, eds., World Scientific, 1997, pp. 209–225.

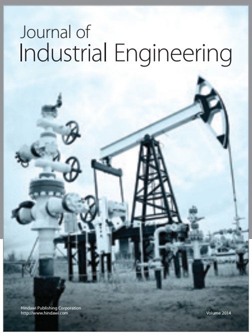
[24] Untersuchung und Modellierung des Lokalen Modells (LM) für Cluster paralleler Systeme mit gemeinsamem Speicher, GMD - Forschungszentrum Informationstechnik GmbH, Institut für Algorithmen und Wissenschaftliches Rechnen (SCAI), Schloß Birlinghoven, 53754 Sankt Augustin.

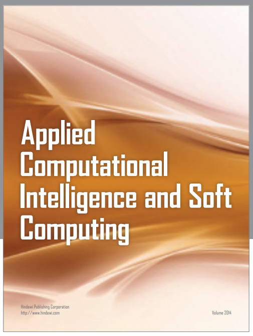
[25] Der DWD LM-Code für SMP-Cluster: Leistungsschätzung und Untersuchungen zur Programmierung, PALLAS GmbH, Hermülheimerstraße 10, 50321 Brühl.

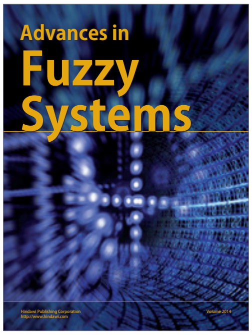

Advances in
Multimedia

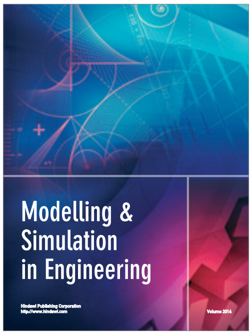
The Scientific World Journal

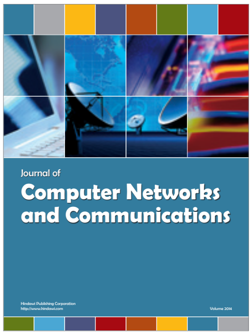

International Journal of
Distributed Sensor Networks

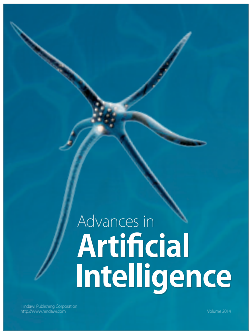

Journal of
Industrial Engineering

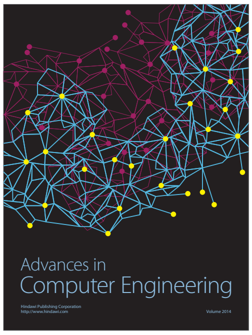

Applied Computational Intelligence and Soft Computing


Advances in
Fuzzy Systems

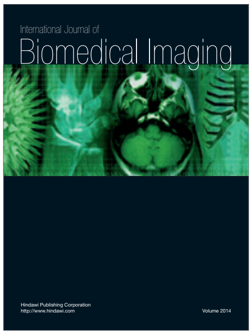

Modelling &
Simulation
in Engineering

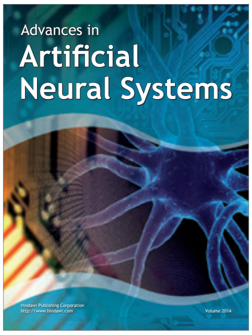

Journal of
Computer Networks and Communications

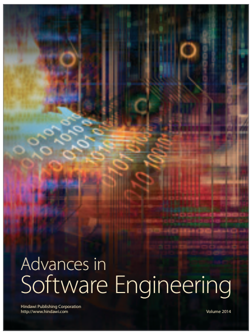

Advances in
Artificial Intelligence


Advances in
Computer Engineering

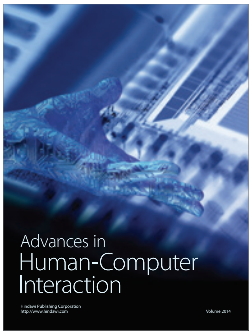
International Journal of
Computer Games Technology

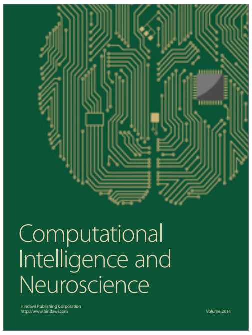

International Journal of
Biomedical Imaging


Advances in
Artificial Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer Interaction


Computational Intelligence and Neuroscience

International Journal of
Reconfigurable Computing

Journal of
Electrical and Computer Engineering