Parallel Computational Magneto-Fluid Dynamics

R. Keppens, FOM-Institute for Plasma-Physics Rijnhuizen,
P.O. Box 1207, 3430 BE Nieuwegein, The Netherlands,
keppens@rijnh.nl

G. Tóth, Department of Atomic Physics, Eötvös University,
Pázmány Péter sétány 1, 1117 Budapest, Hungary,
gtoth@hermes.elte.hu

J.P. Goedbloed, FOM-Institute for Plasma-Physics Rijnhuizen,
P.O. Box 1207, 3430 BE Nieuwegein, The Netherlands,
goedbloed@rijnh.nl

June 25, 1999

Abstract

In a Priority Program on Massively Parallel Computing, eight Dutch research groups (`http://www.phys.uu.nl/~mpr/`) cooperate since 1995 to develop new algorithms and software to simulate the non-linear dynamics of thermonuclear, geophysical, and astrophysical plasmas. As one result of this collaboration, the general-purpose **Versatile Advection Code** (`http://www.phys.uu.nl/~toth/`) has demonstrated excellent scaling properties across a variety of shared and distributed memory architectures.

VAC uses various shock capturing numerical methods with explicit, semi-implicit, or fully implicit time stepping, on 1, 2, or 3 dimensional finite volume grids. Portability to different hardware platforms is achieved by preprocessors that can translate the code from Fortran 90 both forwards to High Performance Fortran and backwards to Fortran 77.

With this code, we simulate complicated magnetized plasma phenomena where simultaneous large-scale and small-scale structures evolve on different timescales. We investigate fundamental hydromagnetic instabilities in multi-dimensional, fully non-linear regimes, as well as astrophysically oriented applications. An overview of recently investigated phenomena includes: the role of the magnetic field in Rayleigh-Taylor and Kelvin-Helmholtz instabilities, stellar wind models and solar coronal mass ejections, and accretion flows onto compact objects (black holes).

1 Introduction

The cluster project on Parallel Computational Magneto-Fluid Dynamics of the Dutch Science Organization (NWO) has developed and applied powerful numerical tools to CPU-intensive research applications. The eight groups involved are collaboratively progressing in nonlinear magnetohydrodynamical (MHD) simulations of astrophysical, thermonuclear and geophysical interest, in tailoring parallel eigenvalue solvers to MHD spectroscopy, and in implementing sophisticated parallel ocean circulation codes. A series of annual reports document our achievements [5]. The coding efforts of the past three years resulted in various optimized, scalable codes. Both the data parallel and the message passing paradigms have led to successful implementations on various vector and parallel platforms.

A particular highlight in this collaboration is the continued development of the Versatile Advection Code (VAC) [16, 19] to simulate nonlinear fluid or plasma dynamics. The code is versatile with respect to (1) applications (shallow water, Euler, Navier-Stokes, or magnetohydrodynamic equations), (2) geometry (1D, 1.5D, 2D, 2.5D or 3D), (3) computer platforms (PC, workstation, vector to distributed memory platform [7, 20]), (4) high resolution spatial [17] and (5) temporal discretizations [21, 10]. This state-of-the-art software package has grown out of intense collaboration between researchers from the astrophysics and thermonuclear physics groups with specialists in numerical analysis and computer science. A wide variety of applications has been or can be addressed, thanks to VAC's versatility for choosing a problem-specific optimal solution strategy.

The philosophy behind VAC is using a single software with options and switches for various problems, rather than developing a different method or version for each problem separately. The advantage of such a general approach is a reduction of overall time for software development, easier maintenance, compatibility of different parts, automatic extension of new features to all existing applications. The price of the general approach is some added complexity in the source code.

2 Numerical and parallel aspects

VAC aims to *solve a set of conservation laws with source terms.* Implemented are: the simple transport equation for test purposes, the Euler equations of compressible hydrodynamics with adiabatic or full energy equation, and the resistive MHD equations with isothermal or full energy equation. Source terms for external gravity, heat conduction, and viscosity can be included as library subroutines, e.g. the Euler equations with the viscosity source terms are the compressible Navier-Stokes equations. Other source terms can be defined in user written subroutines. We refer to [3] for a recent introductory textbook discussing these various systems of equations.

All equation-modules are written in the original *Loop Annotation Syntax* (LASY) [18]. The LASY notation essentially generalizes the Fortran 90 array syntax with patterns that are expanded according to dimensionality. This source code is then translated to Fortran 90 by a Perl preprocessor for a D-dimensional simulation of the selected physics problem. As a result, the number of spatial dimensions $1 \leq D \leq 3$ and the number of components $D \leq C \leq 3$ for the vector variables (e.g. the momentum and the magnetic field) may have 6 different combinations. The equations can be solved on a 1, 2 or 3D structured grid with *conservative finite volume discretization.* Structured grids include Cartesian, polar and spherical grids as trivial examples, but they can be used in more complicated geometries as well. In 1D and 2D both slab and cylindrical symmetry may be assumed for the ignored dimension(s). Boundary conditions are implemented by two layers of *ghost cells* around the mesh. For each boundary and each variable the type of the boundary can be defined as periodic, symmetric, antisymmetric, open, fixed, etc. Time dependent or more complicated boundary conditions can be realized via user defined subroutines.

Several algorithms are available for solving the differential equations: The Flux Corrected Transport scheme (FCT) [2], two Total Variation Diminishing schemes with a

Roe-type Riemann solver (TVD [4] and TVD-MUSCL [24]), and a Lax-Friedrichs type TVD scheme (TVDLF [25]) with no Riemann solver. The TVD schemes maintain monotone profiles by appropriately limiting the slopes of the variables. Since they make use of an approximate Riemann solver, they resolve discontinuities without spurious oscillations, but the Riemann solver is specific to the equation module. The TVD Lax-Friedrichs scheme has proven to be more robust, but slightly more diffusive. All these schemes are *second order in space and are able to simulate shocks and other discontinuities* as well as smooth flows. Multidimensional equations are solved either by Strang-type dimensional splitting [15], or by adding the fluxes from all directions simultaneously. Source terms may be included in a time split fashion or added together with fluxes. An excellent introduction into these methods can be found in [12].

When the MHD equations are solved, the solenoidal condition on the magnetic field can be enforced in several ways. We only mention the possibility of using a projection scheme [1] which requires the solution of a Poisson equation. The Poisson equation is solved by the Conjugate Gradient (CG) and the stabilized Bi-Conjugate Gradient (BiCGSTAB [23]) methods, both implemented in LASY syntax.

Fully implicit and semi-implicit time integration schemes [10, 21] are also available, both for time accurate and steady state problems. The Jacobian matrix is then evaluated numerically. The resulting linear system is solved by a block tridiagonal solver in 1D, while in multidimensional simulations, preconditioned [22] iterative schemes are used.

In [20, 7], we compared the performance of the code on different computer platforms, including work stations and vector and parallel supercomputers. We make use of High Performance Fortran (HPF) to run on distributed memory architectures. Therefore, the Perl script `f90tohpf` inserts HPF distribution directives into the translated LASY source code for all arrays defined over the computational grid. This automatic insertion allows one to experiment with different distributions, like `(BLOCK,BLOCK)` or `(*,BLOCK)` for a 2D problem. The optimal distribution for the particular problem and computer platform is then easily determined from scaling experiments and then used for production simulations. We reported scaling for HPF execution on a Cray T3D, Cray T3E and IBM SP platforms. Timing results for the latter two platforms for a specific 3D MHD problem are given below, alongside the vector Cray C90 performance.

3 Research Achievements with VAC

VAC has been applied to investigate non-linear plasma dynamics, like Rayleigh-Taylor [11] and Kelvin-Helmholtz instabilities [8, 6], stellar winds [9] and accretion disks [14], and solar physics applications like prominences and waves in coronal loops. The number of applications is still increasing. Here, we briefly discuss some of these recent results.

- *Rayleigh-Taylor instability*

  In [11], we simulated the development of the Rayleigh-Taylor instability, causing a heavy compressible gas to mix into a lighter one underneath, in an external gravitational field. We compared the development of this instability in two and three spatial dimensions, without and with magnetic fields. Adding an initial horizontal magnetic field, the mixing process is slowed down and may suppress the development of secondary Kelvin-Helmholtz instabilities in shear flow regions entirely.

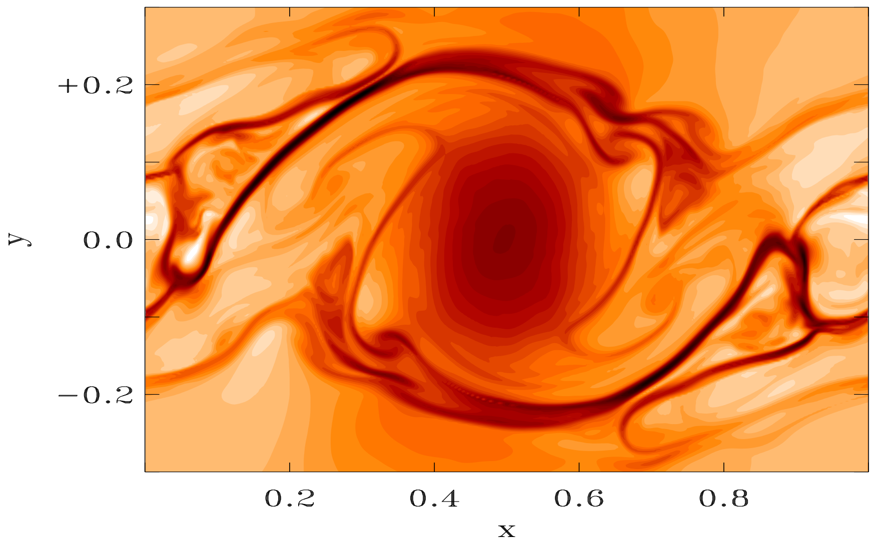

Figure 1: This figure displays the density structure in a two-dimensional current-vortex sheet evolution when it transits to magnetically modified turbulence. This occurs when a Kelvin-Helmholtz unstable shear flow triggers small-scale reconnection events.

- *Kelvin-Helmholtz instability*

  We performed a numerical study of the Kelvin-Helmholtz instability occuring at the interface of a shear flow configuration in compressible magnetohydrodynamics. The calculations cover two [8] and three dimensional [6] configurations.

  The 2D results considered an initial magnetic field aligned with the shear flow, and analyzed the differences between cases where the field is unidirectional everywhere and where the field changes sign at the interface. In the latter case, the initial current sheet gets amplified by the vortex flow and can become unstable to tearing instabilities forming magnetic islands. Thereby, the evolution of such a current-vortex sheet eventually transits to magnetically modified turbulence (Fig. 1).

  The 3D simulations considered shear flow in a cylindrical jet configuration, embedded in a uniform magnetic field. The growth of linear perturbations at specified poloidal and axial mode numbers demonstrated intricate non-linear coupling effects. We identified the physical mechanims leading to induced secondary Kelvin-Helmholtz instabilities at higher mode numbers (see Fig. 2). The initially weak magnetic field becomes locally dominant in the non-linear dynamics before and during saturation.

  In Table 1, we list execution times per time step for this 3D MHD problem (8 unknowns per grid point) on several platforms. The resolution was held fixed at $50 \times 100 \times 100$ (altogether 4 million unknowns), which is still a modest resolution. The code requires roughly 400 MB memory on the Cray C90, where we reach 388.4 Mflops on a single CPU which is 40 % of the theoretical peak performance. When we exploit autotasking on the shared memory C90 (using `f90 -O scalar3,vector3,task3,inline3`), we reach an efficiency (Speedup/Number of PEs) of 97 % on 2 CPUs and 91 % on 4 CPUs. For the distributed memory platforms IBM SP and Cray T3E, we find similarly good speedup (see Table 1). For

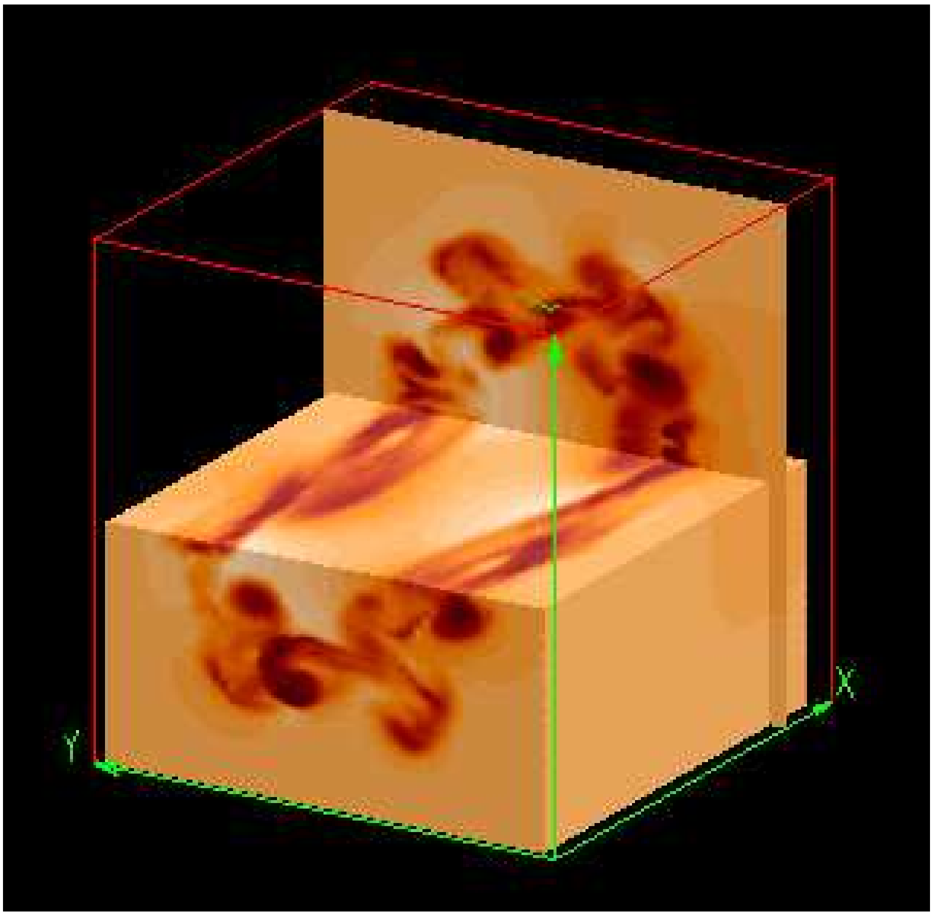

Figure 2: The three-dimensional density structure for a magnetized, jet-like shear flow. Kelvin-Helmholtz instabilities disrupt the originally circular jet cross-section, and cause localized density depletions (dark) and enhancements (bright). At the time shown, non-linear magnetic effects play a dominant role, with the strongest fields coinciding with low density regions.

Platform	1PE	2PE	4PE	8PE	16PE
Cray C90	8.40 19.5%	4.32 [$\times$1.9] 19.4%	2.27 [$\times$3.7] 19.4%	– –	– –
Cray T3E	– –	– –	– –	24.3 16.4%	12.9 [$\times$1.9] 16.6%
IBM SP	– –	– –	14.3 15.2%	7.7 [$\times$1.8] 15.6%	4.2 [$\times$3.4] 16.6%

Table 1: Timings on various platforms and number of PEs, in seconds per time step for the Kelvin-Helmholtz 3D ($50 \times 100 \times 100$) MHD simulation. Speedup factors are given between brackets and the percentage of time spent on the Poisson solver is indicated.

these HPF runs, the distribution was set to `(*,BLOCK,BLOCK)`. On both platforms, the Portland Group `pghpf 2.4-4` compiler was used with optimization level `-O3`. The $\nabla \cdot \mathbf{B} = 0$ condition was enforced through a projection scheme using 20 iterations of the Conjugate Gradient method to solve the Poisson equation. The table lists the percentage of time spent on the Poisson problem as well: it clearly scales both in shared and distributed memory execution.

- *Stellar wind and accretion flow*

  In [9], we simulated stellar winds, while in [14], accretion flows were modeled. Current modeling efforts include both steady-state and time-dependent transonic outflows from magnetized, rotating stars. For steady-state models, we efficiently exploit implicit time stepping. Specifically, Fig. 3 presents an axisymmetric, stationary solar wind solution that takes into account both the solar rotation and the presence of an equatorial dead zone.

4 Conclusions

We have tackled various hydrodynamic and magnetohydrodynamic physics issues where powerful numerical tools are a necessity. The examples presented here, all calculated with the Versatile Advection Code, show how we succesfully modeled axisymmetric solar and stellar winds, exploiting implicit time stepping strategies, or how we use parallel computing to investigate the non-linear evolution of Kelvin-Helmholtz instabilities in 3D MHD. We are extending our stellar wind simulations to fully 3D, time-dependent phenomena, along the lines of recent work by Linker *et al.* [13], and we can now study Kelvin-Helmholtz behaviour in driven plasma loops. Large-scale computing for plasma physics and astrophysics research promises to yield enormous assets in the coming millenium.

Acknowledgements. The Versatile Advection Code was developed as part of the project on 'Parallel Computational Magneto-Fluid Dynamics', funded by the Dutch Science Foundation (NWO) Priority Program on Massively Parallel Computing, and coordinated by JPG. Computer time on the Cray C90, the IBM SP (SARA, Amsterdam), and the Cray T3E (HPαC, Delft) was sponsored by the Dutch 'Stichting Nationale Computerfaciliteiten' (NCF). GT is supported by the Hungarian Science Foundation (OTKA D 25519) and the Bolyai Fellowship.

References

[1] J. U. Brackbill and D. C. Barnes, 1980, J. Comput. Phys. 35, 426.

[2] J. P. Boris and D. L. Book, 1973, J. Comput. Phys. 11, 38.

[3] A. R. Choudhuri, 'The physics of fluids and plasmas: an introduction for astrophysicists', Cambridge University Press, 1998.

[4] A. Harten, 1983, J. Comput. Phys. 49, 357.

[5] Parallel Computational Magneto-Fluid Dynamics: third annual report, MPP-CMFD team, eds. R. Keppens and J.P. Goedbloed, 1998, Rijnhuizen Report 98-234, also available at `http://www.phys.uu.nl/~mpr/mpr98.ps.gz`

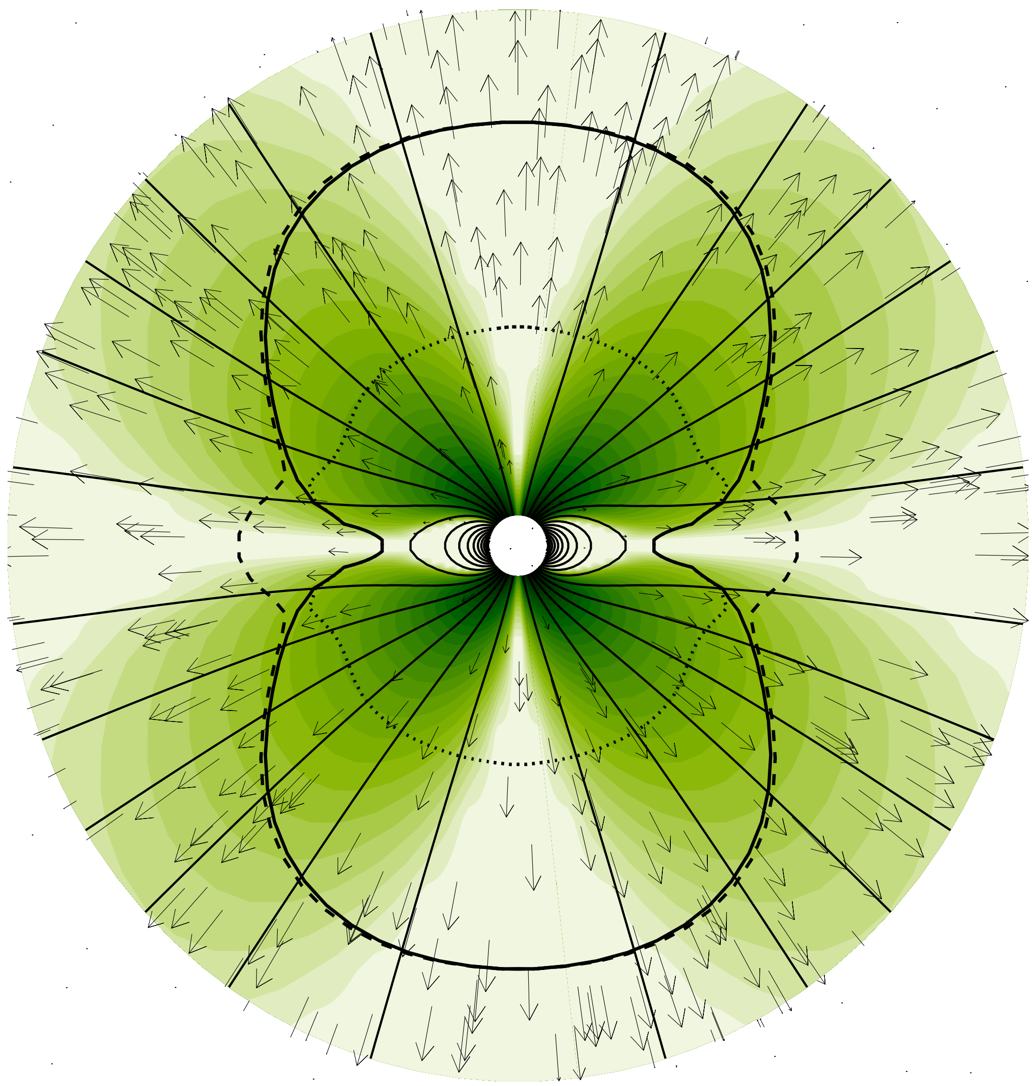

Figure 3: An example of a stellar wind calculated with the Versatile Advection Code. Solid lines indicate the coronal magnetic field structure, with both open and closed field line regions. Thermally driven plasma escapes freely (vectors) along the open field lines, thereby continuously accelerating from subslow to superfast speeds. The three MHD transonic critical curves are indicated (hourglass curves). The grey-scale gives the absolute value of the toroidal magnetic field, built up by the stellar rotation.

[6] R. Keppens and G. Tóth, 1999, Phys. of Plasmas 6(5), 1461.

[7] R. Keppens and G. Tóth, 'Using High Performance Fortran for Magnetohydrodynamic simulations', 1998, submitted to Parallel Computing.

[8] R. Keppens, G. Tóth, R. H. J. Westermann, and J. P. Goedbloed, 1999, J. Plasma Phys. 61(1), 1.

[9] R. Keppens and J. P. Goedbloed, 1999, Astron. Astrophys. 343, 251.

[10] R. Keppens, G. Tóth, M. A. Botchev, and A. van der Ploeg, 1999, Intern. Journ. for Numer. Meth. in Fluids 30, 335.

[11] R. Keppens and G. Tóth, in: J. M. L. M. Palma, J. Dongarra and V. Hernandez (eds.), VECPAR'98 - Third International Conference for Vector and Parallel Processing, Lecture Notes in Computer Science 1573 (Springer, Berlin, 1999), 680.

[12] R. J. LeVeque, D. Mihalas, E. A. Dorfi, and E. Müller, 'Computational Methods for Astrophysical Fluid Flow', Saas-Fee Advanced Course 27, Lecture Notes 1997, eds. O. Steiner and A. Gautschy (Springer, Berlin, 1998).

[13] J. A. Linker, Z. Mikic, D. A. Biesecker, R. J. Forsyth, S. E. Gibson, A. J. Lazarus, A. Lecinski, P. Riley, A. Szabo, B. J. Thompson, 1999, J. Geophys. Res. 104, 9809.

[14] D. Molteni, G. Tóth, and O. A. Kuznetsov, 'On the azimuthal stability of shock waves around black holes', 1999, Astrophys. J., in press.

[15] G. Strang, 1968, SIAM J. Numer. Anal. 5, 506.

[16] G. Tóth, 1996, Astrophys. Lett. & Comm. 34, 245.

[17] G. Tóth and D. Odstrčil, 1996, J. Comput. Phys. 128, 82.

[18] G. Tóth, 1997, J. Comput. Phys. 138, 981.

[19] G. Tóth, in: B. Hertzberger and P. Sloot (eds.), High Performance Computing and Networking, Lecture Notes in Computer Science 1225, Proceedings HPCN Europe 1997 (Springer, Berlin, 1997), 253.

[20] G. Tóth and R. Keppens, in: P. Sloot, M. Bubak, and B. Hertzberger (eds.), High Performance Computing and Networking, Lecture Notes in Computer Science 1401, Proceedings HPCN Europe 1998 (Springer, Berlin, 1998), 368.

[21] G. Tóth, R. Keppens, and M. A. Botchev, 1998, Astron. Astrophys. 332, 1159.

[22] A. van der Ploeg, R. Keppens, and G. Tóth, in: B. Hertzberger and P. Sloot (eds.), High Performance Computing and Networking, Lecture Notes in Computer Science 1225, Proceedings HPCN Europe 1997 (Springer, Berlin, 1997), 421.

[23] H. A. van der Vorst, 1992, SIAM J. Sci. Statist. Comput. 13, 631.

[24] B. van Leer, 1979, J. Comput. Phys. 32, 101.

[25] H. C. Yee, A class of high-resolution explicit and implicit shock-capturing methods, 1989, NASA TM-101088.