

A parallel implementation of an MHD code for the simulation of mechanically driven, turbulent dynamos in spherical geometry

K. Reuter^{a,*}, F. Jenko^a, C.B. Forest^b, R.A. Bayliss^b

^a *Max-Planck-Institut für Plasmaphysik, EURATOM Association, D-85748 Garching, Germany*

^b *Department of Physics, University of Wisconsin–Madison, 1150 University Ave., Madison, WI 53706, USA*

Received 24 July 2007; accepted 20 February 2008

Available online 26 February 2008

Abstract

A parallel implementation of a nonlinear pseudo-spectral MHD code for the simulation of turbulent dynamos in spherical geometry is reported. It employs a dual domain decomposition technique in both real and spectral space. It is shown that this method shows nearly ideal scaling going up to 128 CPUs on Beowulf-type clusters with fast interconnect. Furthermore, the potential of exploiting single precision arithmetic on standard x86 processors is examined. It is pointed out that the MHD code thereby achieves a maximum speedup of 1.7, whereas the validity of the computations is still granted. The combination of both measures will allow for the direct numerical simulation of highly turbulent cases ($1500 < \text{Re} < 5000$), which have been previously impractical, given today's computational speed.

© 2008 Elsevier B.V. All rights reserved.

PACS: 47.65.Md

Keywords: Magnetohydrodynamics; Dynamo theory; Parallelization

1. Introduction

One fundamental issue in geophysics and astrophysics is the question how the kinetic energy in a flow of an electrically conducting fluid or plasma is converted to magnetic energy. This mechanism is usually referred to as the self-excited dynamo process or magnetohydrodynamic (MHD) dynamo, and it is accepted to be the cause of the magnetic fields of the Earth and of many astrophysical objects [1]. In these systems, rotational, thermal, and (gravitational) potential energy serve as sources for the generation of large scale magnetic fields. Briefly described, the process by which these fields are generated is as follows. When an electrically conducting fluid flows through a region interspersed with a magnetic field, an electromotive force (EMF) is induced. This EMF causes currents in the medium which in turn generate a magnetic field. If the latter

is oriented in a way that it reinforces the initial (seed) field, and if the amplification is sufficiently strong to overcome Ohmic losses, then the magnetic field will grow in time.

To study self-excitation, several experiments have been carried out using liquid metal, at first prescribing laminar flows by pipes [2,3]. With the advent of modern computing resources, three-dimensional direct numerical simulations of geo-, astrophysical and laboratory dynamos also became possible, targeted at the investigation of the dynamo effect in more realistic, bounded geometries [4–7]. At the same time, theoretical understanding of MHD turbulence has been advanced by simulations in periodic box geometry [8,9]. Turbulence plays an important role in the onset and saturation of the dynamo process. Recent investigations focus on the role of turbulent fluctuations in magnetic field generation. Experimentalists turned towards less restricted flows in simply connected geometries [10–14], whereas theorists examined the effects of different scales of fluctuations on the dynamo onset [15].

In the case of an electrically conducting, incompressible fluid, magnetic (\mathbf{B}) and velocity (\mathbf{v}) fields are governed by the

* Corresponding author.

E-mail address: khr@ipp.mpg.de (K. Reuter).

following MHD equations

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times \mathbf{v} \times \mathbf{B} + \frac{1}{\mu_0 \sigma} \nabla^2 \mathbf{B}, \quad (1)$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho (\mathbf{v} \cdot \nabla) \mathbf{v} = \mathbf{j} \times \mathbf{B} + \rho \nu \nabla^2 \mathbf{v} - \nabla p + \mathbf{F}, \quad (2)$$

$$\nabla \cdot \mathbf{B} = \nabla \cdot \mathbf{v} = 0, \quad (3)$$

where σ is the conductivity, ρ is the mass density, ν is the viscosity, p is the pressure, $\mathbf{j} = \mu_0^{-1} \nabla \times \mathbf{B}$ is the current density, and \mathbf{F} is a driving term. Eqs. (1)–(3) constitute a non-linear system of partial differential equations which can only be tackled numerically. Fluid turbulence is governed by the Reynolds number $\text{Re} = LV/\nu$, while magnetic self-excitation is in addition governed by the magnetic Reynolds number $\text{Rm} = \mu_0 \sigma LV$, where L (V) is a characteristic length (velocity) of the system. Magnetic field generation can occur if the condition $\text{Rm} > \text{Rm}_{\text{crit}}$ is satisfied, where Rm_{crit} is a threshold which depends on the flow geometry as well as on the level of turbulent fluctuations expressed by Re . To find and explain the shape of the curve $\text{Rm}_{\text{crit}}(\text{Re})$ is an active field of research in theory and of practical relevance for magnetic field generation in laboratory experiments [16].

A pseudo-spectral code which solves the MHD equations in the spherical geometry of an impeller-driven liquid metal dynamo experiment was developed at the University of Wisconsin–Madison [16]. Using the MHD code, a parameter scan in Re – Rm space is currently performed to shed light on the dynamo threshold. The program was initially designed to run on a single processor which limited turbulent simulations to $\text{Re} < 1500$, given today's computational speed. In the experiment using liquid sodium, the magnetic Reynolds number is of the order $\sim 10^2$, whereas the Reynolds number is of the order $\sim 10^7$ [11]. While the magnetic field can be resolved well, fluid turbulence at such high Re is far beyond the scope of direct numerical simulations. Considering Kolmogorov's theory of turbulence, it can be estimated that the required number of grid points scales like $\text{Re}^{9/4}$ [17]. However, by choosing $\text{Re} \gg \text{Rm}$ in the simulations it is possible to asymptotically capture the dominant effects the turbulence has on the magnetic field, giving a unique opportunity to compare MHD simulations with experiments.

In this paper, we report the parallelization of the code using a domain decomposition technique in real and spectral space. Furthermore, we show that a significant speedup is gained by using IEEE single precision arithmetic, without affecting the validity of the computations. The combination of both measures permits the numerical exploration of the regime $1500 < \text{Re} < 5000$. The method presented here is of potential use to speed up similar pseudo-spectral codes.

This paper is organized as follows. In Section 2, a brief review of the mathematical/physical model is given, as it is implemented in the code. In Section 3, the parallelization strategy is discussed. The potential gain of using single precision arithmetic in the code is discussed in Section 4. The benefit of both measures is quantified in terms of scaling tests in Section 5. Finally, Section 6 closes with a summary.

2. Mathematical/physical model

In order to substantiate the parallelization strategy presented in the next section, we give an abridgment of the physical model and the numerical method. In a spherical geometry, it is favorable to split magnetic and velocity fields into poloidal and toroidal components of the form

$$\mathbf{v} = \nabla \times \nabla \times s \mathbf{r} + \nabla \times t \mathbf{r}, \quad (4)$$

$$\mathbf{B} = \nabla \times \nabla \times S \mathbf{r} + \nabla \times T \mathbf{r}, \quad (5)$$

where s , t and S , T are complex scalar functions of the coordinates r , θ and ϕ [18]. One advantage of this representation is that the curl of a vector field effectively reduces to a Laplacian applied to its poloidal scalar function. One easily derives from Eq. (4) that the vorticity reads

$$\boldsymbol{\omega} = \nabla \times \nabla \times t \mathbf{r} + \nabla \times (-\nabla^2 s) \mathbf{r}. \quad (6)$$

The scalar functions are expanded in normalized spherical harmonics $Y_{l,m}(\theta, \phi)$, the analogue of a Fourier decomposition on the surface of a sphere. Taking the curl of Eq. (2) and substituting Eq. (4) gives two equations which govern the temporal evolution of the radial parts $s_{l,m}(r, t)$ and $t_{l,m}(r, t)$ for each mode l, m . In a dimensionless form, they read

$$\left(\frac{\partial}{\partial \tau} + \frac{\text{Rm}_0}{\text{Re}_0} \nabla^2 \right) \nabla^2 s = \text{Rm}_0 [G]_T + [\nabla \times \mathbf{F}]_T, \quad (7)$$

$$\left(\frac{\partial}{\partial \tau} - \frac{\text{Rm}_0}{\text{Re}_0} \nabla^2 \right) t = \text{Rm}_0 [G]_S + [\nabla \times \mathbf{F}]_S, \quad (8)$$

where τ is the time variable, G is the sum of the nonlinear advection term and the Lorentz force, and the subscripts “ S ” and “ T ” signify the corresponding poloidal and toroidal scalar functions. The driving term \mathbf{F} models the counter-rotating impellers as they are used in the Madison Dynamo Experiment. Applying the same method on Eq. (1), evolution equations for the magnetic scalars S and T are obtained. For further details, as well as for a discussion of boundary conditions the reader is referred to [16].

Radial derivatives of fourth-order accuracy are computed in terms of finite differences on a non-uniform radial grid, which allows mesh packing near the outer boundary. The nonlinear terms G are evaluated using a pseudo-spectral method. This approach requires efficient routines which convert vector fields from real space to spectral space, and vice versa. Aliasing error is avoided in a standard way by truncating the upper third of the spectrum. The number of spectral modes is given by the equation $n = (l + 1)(l + 2)/2$, where l is chosen as $2/3$ of the number of points in poloidal direction for dealiasing reasons. Despite the fact that more than 75% of the computational time is used for these spectral transforms, the pseudospectral method pays off in terms of simplicity. Explicit second-order Adams–Bashforth predictor–corrector steps are used to advance the nonlinear terms, whereas the implicit Crank–Nicolson method is used to compute the temporal evolution of the linear terms. The latter requires the solution of linear band-diagonal systems of equations, the corresponding coefficient matrices are constructed from the discretized form of the differential operators

on the left-hand side of Eqs. (7) and (8), analogously for the equations which govern the magnetic field in terms of S and T . Since these matrices only change during runtime when the time step has to be reduced, they are precomputed and stored. To check the validity of the simulation, the balance of power is computed, which monitors the energy flow between the velocity field and the magnetic field.

3. Parallelization strategy

The geometry of the problem suggests a radial domain decomposition into spherical shells which are distributed across nodes. Computations are performed at the discrete radii in parallel, either in real space for all discrete angles θ and ϕ , or in spectral space for all modes l, m . The spectral transforms operate independently in radial direction, and thus are parallelized easily. To satisfy the requirements of the fourth-order radial derivatives, ghost cells with a width of 2 points have to be kept up to date at the domain boundaries. Thus, the right-hand side of the discretized form of Eqs. (7) and (8) can be computed using the radial decomposition.

However, the implicit Crank–Nicolson method demands a full radial range of points for each spectral mode. This requirement is contrary to the domain decomposition in real space, but at the same time particularly suitable for a second decomposition in spectral space. To connect both decompositions, global transposes of arrays which contain the right-hand side of the discretized form of Eqs. (7) and (8) are needed. The approach was implemented using the Message Passing Interface (MPI). Despite the fact that the global matrix transposes require expensive all-to-all communication, this method proves proper scaling on Beowulf-type clusters with fast interconnect, as it is shown in Section 5.

4. Exploiting single precision arithmetic

Some of the recent microprocessors perform significantly better in IEEE single precision arithmetic than they do in double precision. Examples include Intel's Pentium IV and AMD's Opteron processors which achieve twice the theoretical peak performance in single precision compared to double precision, provided that Streaming SIMD Extensions (SSE2) are used [19]. These CPUs are widely found in Beowulf-type clusters, which makes it an interesting feature to exploit. IBM's Cell architecture is even 10 times faster in single precision, whereas certain 64-bit microprocessors like the IBM POWER series or Intel's Itanium 2 do not profit from single precision arithmetic [20]. Apart from architecture-specific aspects, single precision arithmetic halves memory consumption compared to double precision, thus improves caching and effectively doubles bandwidth associated with I/O and inter-process communication in parallel computing.

However, only certain classes of scientific codes are suitable for operation in reduced precision. The present MHD code falls into that category. The dynamics of nonlinear systems, like the MHD dynamo, are sensitive to initial conditions, a property

which is usually referred to as deterministic chaos. A numerical simulation intrinsically works with a limited number of decimal places, thus permanently introducing numerical errors due to truncation. Consequently, time traces during turbulent phases in nonlinear MHD simulations have to be considered in terms of averages. While time traces may differ significantly regarding their short-term characteristics, averages of the system's dynamical variables over a comparable and sufficiently long phase should be independent of the precision used in the simulation.

Originally, the code had been designed to operate in double precision. It was successfully adapted to run in single precision arithmetic, except for one critical region which demands double precision for reasons of numerical stability. This critical region restricts itself to the call of a library routine which solves the band-diagonal systems during the Crank–Nicolson steps, and is now fixed to double precision in the implementation. Overall, the desired mode can be chosen at compile time, provided that suitable numerical libraries are supplied.

5. Numerical results

5.1. Single vs. double precision arithmetic

To justify the use of single precision, identical test cases were computed in both precisions, and the results were compared subsequently. As an example, time traces depicting the magnetic energy and the maximum velocity during a dynamo onset are shown in Fig. 1. While the graphs differ during the initial nonlinear phase due to the effects mentioned above, one finds that the system reaches its quasi-stationary state at the same level. In the run which uses single precision the dynamo onset takes place slightly earlier. Averaging over the interval between 2.5 and 5.0, the periodic time of the oscillations is in both cases 0.262, rounded to three decimal places. The average amplitude of the magnetic energy is 0.182 (sp), and 0.183 (dp), respectively. These numbers show that it is legitimate to speed-up simulations by exploiting single precision arithmetic, especially during the time-consuming exploration of Re–Rm-space, where the question is if self-excitation occurs at all. The physical process which is shown in Fig. 1 is briefly explained as follows. During the initial phase, the fluid is driven to a turbulent flow, only a weak seed magnetic field is present, which is of the order 10^{-6} in terms of the magnetic energy. The velocity field allows the growth of magnetic eigenmodes, which rise exponentially at first and then nonlinearly saturate. At this point the magnetic field reacts back on the velocity field and suppresses the turbulence.

Fig. 2 shows the dependence of the observed speedup on the grid size, measured with 2 processes on a single 2-way Pentium IV node. At a coarse resolution of $50 \times 09 \times 18$ points (where $N_r \times N_\theta \times N_\phi$ characterizes the spherical grid), a small speedup around 1.2 is achieved. When the resolution is successively increased, the speedup also goes up. At $350 \times 16 \times 32$ points, a maximum speedup of 1.72 is measured. Going to higher resolutions reduces the observed speedup. At $400 \times 24 \times 48$ points, it is still at a significant level around

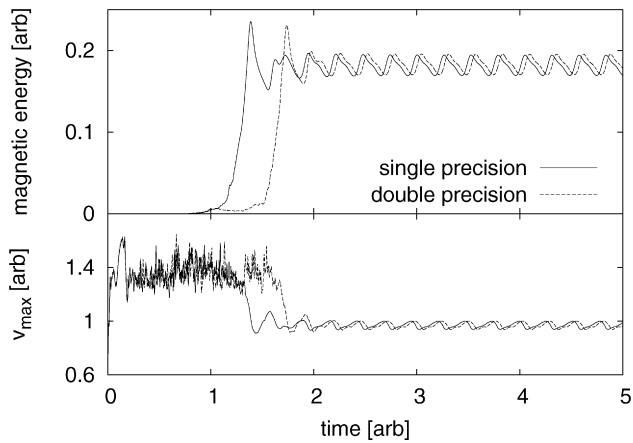


Fig. 1. Simulation of a dynamo onset using single and double precision, respectively, for an identical test case ($Re \approx 520$, $Rm \approx 205$). The observed speedup is $T_{dp}/T_{sp} \approx 1.69$.

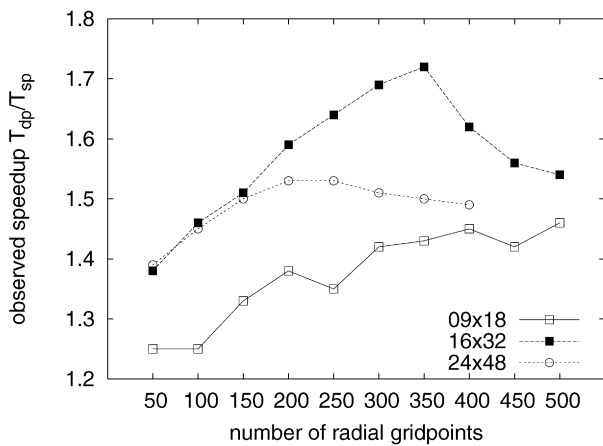


Fig. 2. Observed speedup against number of radial points, gained by exploiting single precision arithmetic on a 2-way Intel Pentium IV node. Angular mesh sizes are as indicated.

1.5. This behavior is likely to be caused by CPU cache effects. At low resolutions, both cases show proper cache usage, which makes the advantage of single precision marginal. An increase of the resolution leads to the situation, where arrays involved in at least a part of the computations still fit into the cache in single precision, while in double precision cache memory is already exceeded. At even finer resolutions, array sizes reach the cache size in both cases, which leads to the observed decrease of the speedup. These numbers can be taken as guiding values to optimize the size of domains in parallel computing. However, they vary between different types of CPUs.

5.2. Parallel computations

The parallel code was first benchmarked on a standard Linux cluster with 2-way Intel Pentium IV nodes, connected via Gigabit Ethernet. While the speedup is nearly ideal going from 1 to 2 processors on a single node, it breaks down dramatically when more than one node is used. This is mainly to be caused by the relatively slow interconnect in combination with

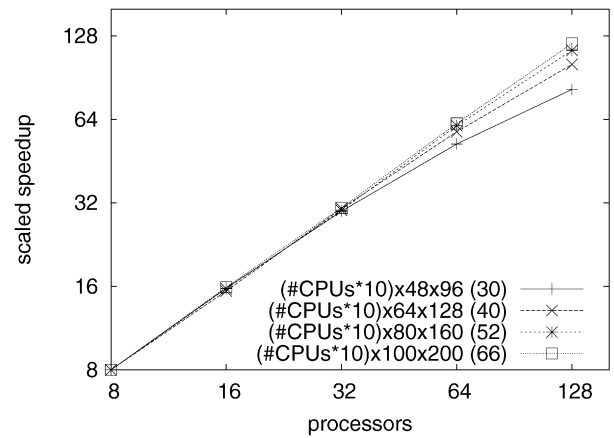


Fig. 3. Weak scaling of the parallel code in 4 different cases. In addition to the grid resolution in real space, the degree l of the spherical harmonics used in the transforms is given in parentheses. The number of radial gridpoints is increased linearly.

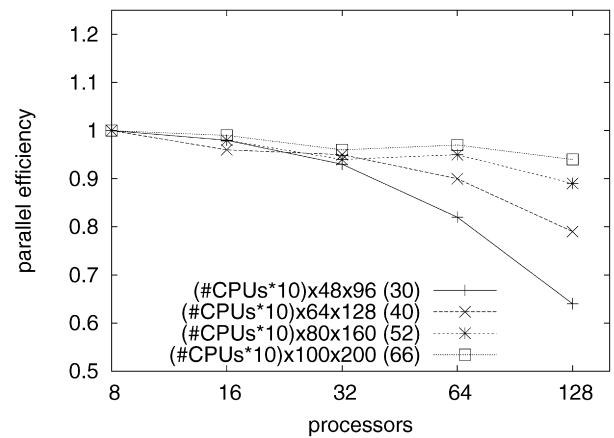


Fig. 4. Parallel efficiency of the code in the cases presented in Fig. 3.

the global matrix transpose which implies expensive all-to-all communication, as it can be shown by measurement. On one processor without the transpose, the Crank–Nicolson steps take about 10% of overall runtime, as it was measured in a test run at $400 \times 32 \times 64$ grid points. Going to 2 CPUs, this proportion is 13%, whereas it rises to 45% (65%, 71%) with 4 (8, 16) processors. To allow for better performance on Beowulf-type clusters with slow interconnect, a change from the Crank–Nicolson method to a fully explicit scheme should be considered.

In addition, the scaling of the parallel code was examined on a Linux cluster with 8 Intel Xeon CPUs per node and InfiniBand interconnect. Consequently, we have rescaled the measurements presented below in the way that 8 CPUs correspond to a speedup of 8, and an efficiency of 1. Fig. 3 shows the scaled speedup (weak scaling) where the resolution is increased linearly with the number of CPUs. It is defined in the standard way as $S = Pt(N_0, 1)/t(PN_0, P)$, where P is the number of CPUs in use, N_0 is the base grid resolution and t is the wallclock time. Four test cases are examined, where the resolution and also the degree l of the spherical harmonics is given in Fig. 3. Fig. 4 shows the corresponding parallel efficiency

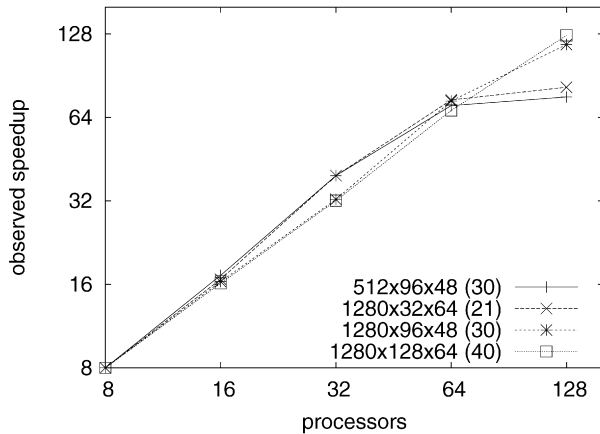


Fig. 5. Strong scaling of the parallel code in 4 different cases.

which is given by the ratio S/P . The cases with $l = 30, 40$ scale efficiently up to 32 processors, whereas the cases with $l = 52, 66$ use 128 processors efficiently. Fig. 5 shows the observed speedup (strong scaling) in 4 test cases where the number of CPUs is increased linearly at a fixed problem size. The observed speedup is defined as the ratio $t(1)/t(P)$. Again, depending on the resolution, a nearly ideal scaling is observed.

6. Summary

In this paper, the parallelization of a nonlinear pseudo-spectral MHD code is reported. It is implemented using a dual domain decomposition technique in both real and spectral space. The parallel version shows a nearly ideal scaling going up to 128 processors. In addition, it is shown that computations on modern x86-architectures (e.g., Intel Pentium IV) can benefit from arithmetic in IEEE single precision by a factor of 1.7. The combination of parallel computing and single precision arithmetic is expected to speed up simulations of turbulent MHD dynamos significantly. It is now possible to explore the highly turbulent regime $1500 < Re < 5000$ in Re–Rm-space.

Acknowledgement

The authors would like to thank F. Merz for the helpful discussions on parallel programming.

References

- [1] H.K. Moffatt, Magnetic Field Generation in Electrically Conducting Fluids, first ed., Cambridge University Press, Cambridge, 1978.
- [2] A. Gailitis, et al., Phys. Rev. Lett. 84 (2000) 4365.
- [3] R. Stieglitz, U. Müller, Physics of Fluids 13 (2001) 561.
- [4] G. Glatzmaier, P. Roberts, Physics of the Earth and Planetary Interiors 91 (1995) 63.
- [5] A. Kageyama, T. Sato, Phys. Rev. E 55 (1997) 4617.
- [6] W. Kuang, J. Bloxham, Nature 389 (1997) 371.
- [7] U. Christensen, P. Olson, G.A. Glatzmaier, Geophys. J. Internat. 138 (1999) 393.
- [8] F. Cattaneo, D.W. Hughes, Phys. Rev. E 54 (1996) R4532.
- [9] W.-C. Müller, D. Biskamp, R. Grappin, Phys. Rev. E 67 (2003) 066302.
- [10] M.D. Nornberg, E.J. Spence, R.D. Kendrick, C.M. Jacobson, C.B. Forest, Phys. Rev. Lett. 97 (2006) 044503.
- [11] M.D. Nornberg, E.J. Spence, R.D. Kendrick, C.M. Jacobson, C.B. Forest, Phys. of Plasmas 13 (2006) 055901.
- [12] E.J. Spence, M.D. Nornberg, C.M. Jacobson, R.D. Kendrick, C.B. Forest, Phys. Rev. Lett. 96 (2006) 055002.
- [13] E.J. Spence, et al., Phys. Rev. Lett. 98 (2007) 164503.
- [14] R. Monchaux, et al., Phys. Rev. Lett. 98 (2007) 044502.
- [15] J.-P. Laval, P. Blaineau, N. Leprovost, B. Dubrulle, F. Daviaud, Phys. Rev. Lett. 96 (2006) 204503.
- [16] R.A. Bayliss, C.B. Forest, M.D. Nornberg, E.J. Spence, P.W. Terry, Phys. Rev. E 75 (2007) 026303.
- [17] U. Frisch, Turbulence. The Legacy of A.N. Kolmogorov, first ed., Cambridge University Press, Cambridge, 1995.
- [18] E. Bullard, H. Gellman, Roy. Soc. London Philos. Trans. Ser. A 247 (1954) 213.
- [19] A. Klimovitski, Intel DeveloperUPDATEMagazine, March 2001.
- [20] J. Langou, et al., Exploiting the performance of 32 bit floating point arithmetic in obtaining 64 bit accuracy (revisiting iterative refinement for linear systems), LAPACK Working Note 175, Department of Computer Science, University of Tennessee, Knoxville, TN 37996, USA, 2006.