

Large-scale gyrokinetic particle simulation of microturbulence in magnetically confined fusion plasmas

S. Ethier
W. M. Tang
R. Walkup
L. Oliker

As the global energy economy makes the transition from fossil fuels toward cleaner alternatives, nuclear fusion becomes an attractive potential solution for satisfying growing needs. Fusion, the power source of the stars, has been the focus of active research since the early 1950s. While progress has been impressive—especially for magnetically confined plasma devices called tokamaks—the design of a practical power plant remains an outstanding challenge. A key topic of current interest is microturbulence, which is believed to be responsible for the unacceptably large leakage of energy and particles out of the hot plasma core. Understanding and controlling this process is of utmost importance for operating current devices and designing future ones. In addressing such issues, the Gyrokinetic Toroidal Code (GTC) was developed to study the global influence of microturbulence on particle and energy confinement. It has been optimized on the IBM Blue Gene/L™ (BG/L) computer, achieving essentially linear scaling on more than 30,000 processors. A full simulation of unprecedented phase-space resolution was carried out with 32,768 processors on the BG/L supercomputer located at the IBM T. J. Watson Research Center, providing new insights on the influence of collisions on microturbulence.

Introduction

Plasmas are ionized gases that are often referred to as the *fourth state of matter*, and they comprise more than 99% of the visible matter in the universe. They exhibit rich complex collective phenomena and are subjects of major areas of research including plasma astrophysics and fusion energy science. Fusion is the power source of the sun and other stars, and it occurs when certain isotopes of the lightest atom, hydrogen, combine to make helium in a very hot (100 million degrees centigrade) plasma. The development of fusion as a secure and reliable energy system that is environmentally and economically sustainable is a truly formidable scientific and technological challenge in the twenty-first century. As such, progress toward this goal requires the acquisition of the basic scientific understanding to enable the innovations that are still needed to make fusion energy a

practical realization. Research in plasma science requires the accelerated development of computational tools and techniques that enhance the scientific understanding needed to develop predictive models that are superior to extrapolations of experimental results. This is made possible by the rapid advances in high-performance computing technology that will allow simulations of increasingly complex phenomena with greater physics fidelity. Accordingly, advanced computational codes, properly benchmarked with theories and experiments, are now recognized to be powerful new tools for scientific discovery. In the key area of turbulent transport of the plasma, the development of the gyrokinetic (GK) formalism [1–3] and its subsequent implementation in advanced simulations have enabled major progress. By averaging the phase information of the fast gyrating motion of the particles out of the kinetic equation, the

©Copyright 2008 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied by any means or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/08/\$5.00 © 2008 IBM

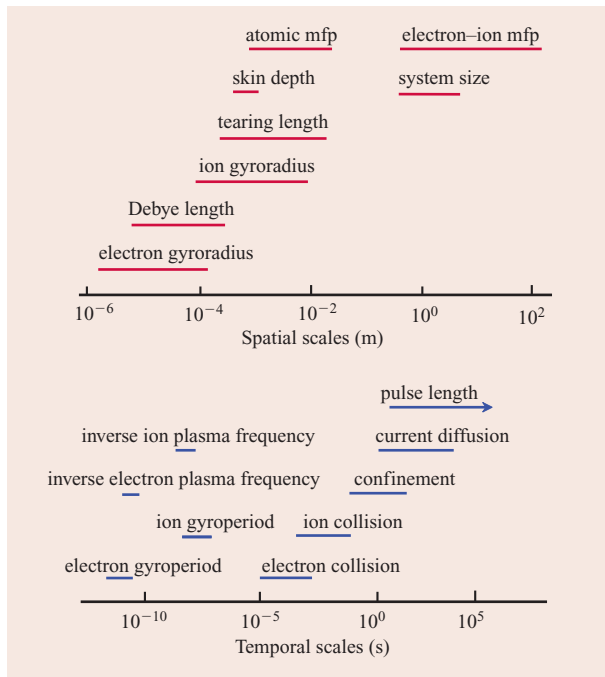


Figure 1

Very large ranges in spatial and temporal scales present major challenges to plasma theory and simulation. In the plasma, the distance covered by electrons and ions between collisions, or the mean free path (mfp), is of the same order as the system size; thus, simulations of phenomena faster than the collision time can be considered collisionless. The pulse length (see arrow), which is the time during which the plasma is confined, continues to grow as we gain better control over the plasma instabilities. (The positioning of terms along the y -axis direction on each graph is arbitrary.)

GK formalism focuses on the physics relevant to the low-frequency plasma microturbulence in magnetic confinement fusion experiments. Our intent in this paper is to provide the readers with a concise background description of the significance of fusion research as a whole and to then focus on the global GK particle-in-cell (PIC) approach to address the plasma microturbulence problem—a very complex, nonlinear phenomenon that is generally believed to play a key role in determining the efficiency of magnetic confinement of fusion-grade plasmas. A more detailed, comprehensive review, which cites other effective approaches including Eulerian and continuum methods, can be found, for example, in Reference [4].

In a magnetically confined plasma, the interplay between the complex trajectories of individual charged particles and the collective effects arising from the long-range nature of electromagnetic forces lead to a wide range of waves, oscillations, and instabilities characterizing the medium. As a result, an enormous

range of temporal and spatial scales are involved in plasmas of interest. As illustrated in **Figure 1**, the relevant physics can span over ten orders of magnitude in time and space. Associated processes include the turbulence-driven (anomalous) transport of energy and particles across a confining magnetic field, the abrupt rearrangements (disruptions) of the plasma caused by large-scale instabilities that reconfigure the magnetic field lines (tearing modes), and the interactions involving the plasma particles with electromagnetic waves (at a distance referred to as the *skin depth*) and also with atoms from the device wall. Many of these phenomena involve short length scales and timescales (microns and nanoseconds), while others occur on long timescales (seconds and minutes in the case of the tokamak pulse length) and length scales on the order of the device size, which is about a meter for most existing tokamaks. Although the fundamental laws that determine the behavior of plasmas, such as Maxwell's equations and equations of classical statistical mechanics, are well known, obtaining their solution under realistic conditions is a scientific problem of extraordinary complexity. Effective prediction of the properties of energy-producing fusion plasma systems depends on the successful integration of many complex phenomena, which, as mentioned, span vast space scales and timescales. This is a formidable challenge that can be met only with advanced scientific computation properly cross-validated with experiment and analytic theory.

In magnetic confinement fusion experiments, the plasma interacts directly with the electromagnetic fields, which can come from an external source or from currents produced within the plasma. This can lead to unstable behavior, in which the plasma rapidly rearranges itself and relaxes to a lower energy state. The resultant thermodynamically favored state is incompatible with the conditions needed for fusion systems, which require that more power output be generated than it takes to keep the hot plasma well confined. However, the hot plasma state is naturally subject to both large- and small-scale disturbances (instabilities) that provide the mechanisms for lowering its energy state. Therefore, it is necessary to first gain an understanding of these complex, collective phenomena, and then to devise the means to control them. The larger-scale (macro) instabilities can produce rapid topological changes in the confining magnetic field, resulting in a catastrophic loss of fusion power density. Even if these instabilities can be controlled or prevented, smaller-scale (micro) instabilities can remain and prevent efficient hot plasma confinement by causing the turbulent transport of energy and particles. In order to make progress in addressing these challenges, researchers in this field have effectively developed the requisite mathematical formalism embodied by GK theory to deal

with the complexity of the kinetic electromagnetic behavior of magnetically confined plasmas.

Plasma microturbulence

The scientific challenges related to magnetically confined plasmas can be categorized in four areas: macroscopic stability, wave–particle interactions, microturbulence and transport, and plasma–material interactions. In addition, the integrated modeling of the physical processes from all of these areas is needed to effectively harvest the physics knowledge from existing experiments and to design future devices. Because charged particles, momentum, and heat move more rapidly along the magnetic field than across it, magnetic fusion research has focused on magnetic traps in which the magnetic field lines wrap back on themselves to cover a set of nested toroidal surfaces, called *magnetic flux surfaces* because each surface encloses a constant magnetic flux.

Macroscopic stability is concerned with large-scale spontaneous deformations of magnetic flux surfaces. These major displacements (or macroinstabilities) are driven by the large electric currents flowing in the plasma and by the plasma pressure. Wave–particle interactions concern how particles and plasma waves interact. Detailed calculations of particle motions in background electromagnetic fields are needed to assess the application of waves to heat the plasma as well as address the dynamics of energetic particles resulting from intense auxiliary heating or alpha-particles from the fusion reactions. Microturbulence and the associated transport come from fine-scale turbulence, driven by inhomogeneities in the plasma density and temperature, which can cause particles, momentum, and heat to leak across the flux surfaces from the hot interior and to be lost at the plasma edge. Plasma–material interactions determine the ways in which high-temperature plasmas and material surfaces can coexist. Progress in the scientific understanding in all of these areas contributes to the interpretation and future planning of fusion systems. This demands significant advances in physics-based modeling capabilities—a formidable challenge that highlights the fact that advanced scientific codes indicate and measure the state of understanding of all natural and engineered systems.

The most fundamental theoretical description of a plasma comes from kinetic equations for the time-evolving distribution function within a six-dimensional phase space of each particle species. The equations are coupled to each other through self-consistent electric and magnetic fields. Velocity moments of these kinetic equations produce a hierarchy of fluid equations amenable to modeling. In general, the simulation techniques used in plasma physics fall into two broad categories: kinetic models and fluid models. Research in

plasma microturbulence has been active during the past decade, and many fluid and kinetic codes have been developed to study its impact on energy and particle confinement [5–15]. The kinetic approach used in this work is the PIC method, pioneered by Dawson and others ([16] and references therein). This method involves integrating a (possibly reduced) kinetic equation in time by advancing marker particles along a representative set of characteristics within the (possibly reduced) phase space. The method essentially involves a Lagrangian formulation in which the dynamics of an ensemble of gyro-averaged particles are tracked. Simulation techniques have been developed over the last 20 years that use finite-sized particles [17, 18] (to reduce the noise due to discrete marker particles), gyrokinetics [1–3] (a reduction of the full kinetic equation to a five-dimensional (5D) phase space that removes high-frequency motion that is not important to turbulent transport), and δf [19, 20] (a prescription for further reducing the discrete particle noise by separating the perturbed from the equilibrium part of the distribution function before integrating the GK equation along the appropriate characteristics). These advances have served to reduce the requirements on the number of particles necessary to faithfully represent the physics and have contributed to dramatically increasing the accuracy and realism of the PIC simulation technique.

Gyrokinetic Toroidal Code

The Gyrokinetic Toroidal Code (GTC) is a global, three-dimensional (3D) PIC code developed to study microturbulence in tokamak toroidal devices [9, 21]. The global capability of this code is unique in that it allows researchers to study the scaling of turbulence with the size of the tokamak and to systematically analyze important global dynamics such as turbulence spreading. When we use the term *global*, we refer to the fact that all of the effects of spatial variations in the plasma are taken into account in the full volumetric extent. Short- and long-scale correlations are calculated.

Particle-in-cell method

In order to carry out PIC simulations, the starting point is the Boltzmann equation—a nonlinear partial differential equation in Lagrangian coordinates. When dealing with plasmas, it is clear that the Coulomb potential for finite-sized particles is modified by Debye shielding (see Reference [16] and references cited therein). Thus, short-range interactions are reduced dramatically because there are equal numbers of electrons and ions within a Debye sphere. This leads to simplification of the expression for the short-range force on a single particle due to the electric-field generated by all of the other particles. Specifically, the point particles here are now effectively

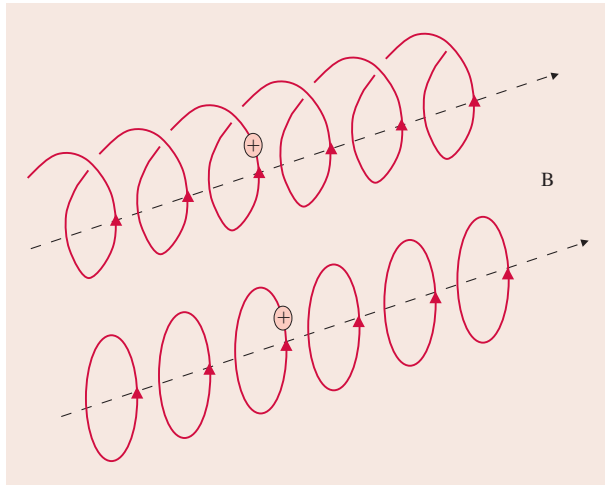


Figure 2

Helical motion (top) of a charged particle in a magnetic field (B) is modified by gyrokinetic approximation to a rotating charged ring (bottom). (Republished from [22], and used with permission from IOP Publishing Limited.)

uniformly charged spheres of Debye-length radius. Dealing with such spheres has a major benefit because instead of requiring N^2 calculations to represent binary interactions for N particles, PIC simulations require only N calculations. Although collisional dynamics are eliminated by this approximation, they can be recovered as subgrid phenomena via Monte Carlo methods with collision operators that can account for the scattering and diffusion of particles in velocity space (see Reference [16] and references cited therein).

Gyrokinetic particle-in-cell method

Major progress in the simulation of the gyrophase-averaged Vlasov–Maxwell system of equations governing low-frequency microinstabilities followed the introduction of the GK methodology by Lee [2, 23]. This progress involved incorporating the ion polarization density into Poisson’s equation, and as illustrated in **Figure 2**, the effective separation of the particle gyromotion from its gyrocenter motion. Here, the actual helical motion of a charged particle in a magnetic field is modified to that of a rotating charged ring subject to guiding center electric field and magnetic drift motion together with parallel acceleration, that is, acceleration along the magnetic field line.

As with general PIC approaches, the GK PIC method consists of moving particles along the characteristics of the governing equation—here, the 5D GK equation. This involves solving the associated equations of motion for each particle in the Lagrangian frame of reference; these

equations are simple ordinary differential equations. The equation increases in complexity because the particles are subjected to forces from an externally imposed (equilibrium) magnetic field and from internal electromagnetic fields generated by the charged particles. A grid is used to map the charge density at each point arising from the particles in the vicinity. This is called the *scatter phase* of the PIC simulation. Maxwell’s equations that govern the fields (e.g., Poisson’s equation in electrostatic simulations) are then solved for the forces that are then gathered back (i.e., mapped back) to the positions of the particles during the gather phase of the simulation. This information is then used for advancing the particles in time by solving the equations of motion, which we refer to as the *push phase* of the simulation. We note that Maxwell’s equations are linear partial differential equations in Eulerian coordinates (lab frame). So, unlike the particle-pushing in the x and v (velocity) phase space, the field calculations are carried out in the lab frame.

As observed from computational results [2, 23] and supported by analytic studies [24], the noise level from the GK PIC simulations was found to be dramatically reduced compared with the noise level from using point particles that naturally include all of the very-short-range interactions (collisions). One interpretation of this property is that the Debye shielding previously mentioned is effectively replaced by the gyroradius shielding that is introduced by the presence of the ion polarization density in the GK Poisson’s equation [25, 26]. The GK particles in our PIC simulations can now be pictured as charged rings as they travel in the field. In order to deposit the charge of the rings on the grid, the GTC utilizes a four-point average method developed by Lee [23] that demonstrates that it is sufficient to divide the total charge between four points on the ring and deposit the four charges to their neighboring grid points. Another significant advance resulted from the introduction of the fully nonlinear delta- f prescription for further reducing the discrete particle noise via separation of the perturbed and the equilibrium part of the distribution function [20]. This is advantageous because the particles do not need to reproduce the change in density and temperature due to the equilibrium profiles. These profiles are included in the equilibrium distribution function and treated analytically. As a result, the particles can be uniformly loaded across the whole simulation volume, which greatly simplifies load balancing in a multiprocessor PIC code such as GTC. As a whole, modern GK methods have effectively speeded up computations by 3 to 6 orders of magnitude in timesteps and 2 to 3 orders of magnitude in spatial resolution. The accuracy and realism of the associated simulations have accordingly benefited from such advances.

GTC features summary

To briefly reiterate and summarize the essential features of our approach, we note that the geometry in GTC in real space is a 3D torus, which is the shape of the closed magnetic field equilibrium in fusion devices. Ion particles are followed in this volume using the gyrophase-averaged version of the collisionless kinetic equation known as *Vlasov's equation*. We solve the particle motion along the characteristics using a highly conservative Hamiltonian formulation in magnetic coordinates for the equations of the guiding center motion [27], effectively reducing the nonlinear PDE (partial differential equation in Lagrangian coordinates) GK equation to a set of linear ODEs (ordinary differential equations). As mentioned, we calculate the self-consistent electrostatic field generated by the particles at each timestep by depositing the charge of each particle on a grid and solving Poisson's equation, a linear PDE in Eulerian coordinates (lab frame). The value of the field at the position of the particles is then gathered back and used for advancing the particles in time. This process is referred to as the *gather-scatter process* in PIC codes. Note that depositing the charge of each particle on the grid is the scatter step, and mapping the forces back from the grid to the individual particle locations is the gather step. Once we know the total force, advancing the particles occurs, a process that is referred to as the *push phase*. GTC utilizes a highly specialized grid that follows the magnetic field lines as they twist around the torus (**Figure 3**). This allows us to use a small number of toroidal planes instead of a regular grid that does not follow field lines and a regular grid that has the same accuracy and number of toroidal modes. Two orders of magnitude in compute time are saved by using this optimized GTC grid. A more extensive description of GTC can be found in [22].

Parallel model

GTC includes three levels of parallelism, although only two are used in this study because of the lack of multithreading on the IBM Blue Gene/L* (BG/L) system. The original parallel scheme implemented in GTC is a one-dimensional (1D) domain decomposition in the toroidal direction (the long way around the torus). Each process is in charge of a domain, and communication is handled with Message Passing Interface (MPI) calls. Particles move from one domain to another while they travel around the torus. Only nearest-neighbor communication in a circular fashion is used to move the particles between domains, or processors. This method scales extremely well to a large number of processors but eventually becomes dominated by communications as more particles move in and out of the shrinking domains at each timestep. However, in practice, domination by communication is never reached because the number of

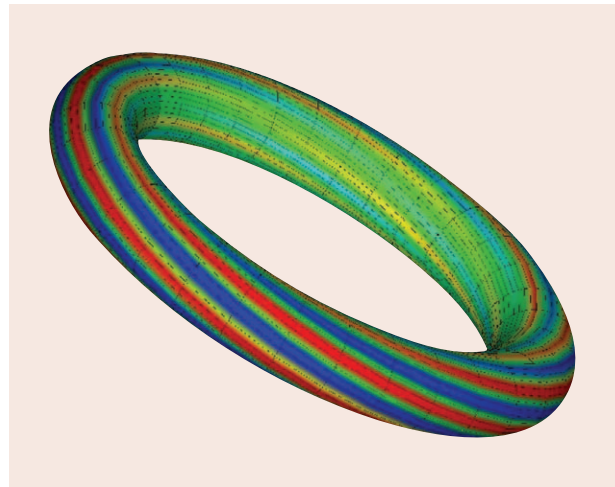


Figure 3

The highly optimized grid of the Gyrokinetic Toroidal Code follows the magnetic field lines as they twist around the torus. Also shown, in color, is the linear mode structure of the electrostatic potential, which also follows the field lines. A rainbow color map is used to represent the potential, with values going from negative (red) to positive (blue). (Republished from [22], and used with permission from IOP Publishing Limited).

domains is determined by the long-wavelength physics that we are studying. A toroidal grid having more than 64 or 128 planes introduces waves of shorter wavelengths in the system. These waves are damped by a physical collisionless damping process known as *Landau damping*. Using a higher toroidal resolution leaves the results unchanged, and hence, GTC generally uses 64 planes for production simulations.

GTC was originally developed and optimized on the large IBM SP* POWER3* computer at the National Energy Research Scientific Computing Center (NERSC) in Berkeley, California. We took advantage of the shared memory capability of that system by adding to the code a second level of parallelism at the loop level, implemented with OpenMP** (Open Multiprocessing) compiler directives. Although of limited scalability due to the serial sections between parallel loops, this method allows GTC to run in mixed-mode MPI/OpenMP on the IBM SP computer.

The advent of large-scale systems that do not support multithreaded parallelism, such as the BG/L system and the Cray XT3**, motivated us to add yet another level of parallelism to increase the concurrency of GTC. Within each toroidal domain, we now divide the particles between several MPI processes. Each process keeps a copy of the local grid, requiring the processes within a domain to sum their contribution to the total charge density on the grid at the end of the charge deposition

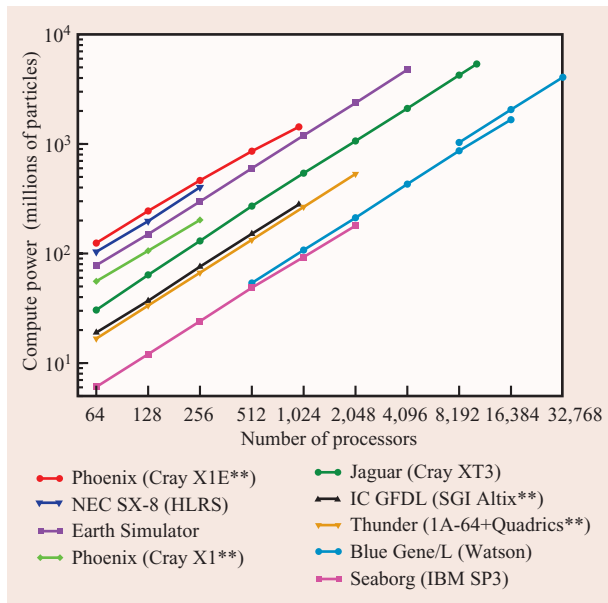


Figure 4

Weak scaling study of the Gyrokinetic Toroidal Code (GTC) on major high-performance computing platforms, including the Japanese Earth Simulator, the Cray XT3 at NCCS/ORNL, and the BG/L system at the IBM T. J. Watson Research Center. The device size is kept fixed while the number of particles is increased proportionally to the number of processors. On these three systems, GTC can move about five billion particles one step per second.

step. Thus, we introduced an *Allreduce* call, along with two additional communicators to handle the new communication patterns. An intradomain communicator links the processes within a common toroidal domain of the original 1D domain decomposition, while a toroidal communicator links in a ringlike fashion all the processes with the same intradomain rank. This extra level of parallelism allows GTC to scale to numerous processors and use many particles, which results in very high phase-space resolution and low statistical noise. GTC has been benchmarked and run on most of the largest computers worldwide, including the 5,120-processor Japanese Earth Simulator, the 40,960-core BG/L system at the IBM T. J. Watson Research Center, and the 12,824-core Cray XT3 at the National Center for Computational Sciences [28, 29]. **Figure 4** shows the results of an extensive benchmarking study of GTC on major high-performance computing platforms: the AMD Opteron** processor-based Cray XT** system (Jaguar), the vector computer Cray X1E (Phoenix) at the National Center for Computational Sciences (NCCS)/Oak Ridge National Laboratory (ORNL) in Oak Ridge, Tennessee (www.nccs.gov), the Earth Simulator vector computer at

the Earth Simulator Center in Yokohama, Japan (www.es.jamstec.go.jp/index.en.html), the BG/L system (BGW) at the IBM T. J. Watson Research Center in Yorktown Heights, New York (www.watson.ibm.com), the SGI Altix cluster (IC) at the Geophysical Fluid Dynamics Laboratory (GFDL) in Princeton, New Jersey (www.gfdl.gov), the Intel Itanium** cluster (Thunder) with a Quadrics interconnect at Lawrence Livermore National Laboratory in Livermore, California (www.llnl.gov), the NEC SX-8 vector computer at the High Performance Computing Center, Stuttgart (HLRS) of the University of Stuttgart, Germany (www.hlrs.de), and the IBM SP POWER3 system (Seaborg) at the NERSC in Berkeley, California (www.nersc.gov).

Optimizing GTC for the BG/L system

As mentioned in the previous section, GTC was originally developed and optimized on the IBM SP POWER3 system. Because the compiler on the BG/L system is essentially the same as the one on the SP system (from the standpoint of the application programmer), most compiler options were familiar and had already been used on the SP system. The original code port, carried out on the single-rack BG/L system at the Argonne National Laboratory, was straightforward, and the only challenge arose from the lack of a Fortran-95 module for the MPI library. This was easily addressed by using the include file `mpif.h` instead of the module.

The next level of optimization involved experimentation with different levels of compiler optimizations such as `-O2`, and `-O3` with options `-qarch=440` and `-qarch=440d`. The `440d` option instructs the compiler to generate parallel instructions for the BG/L PowerPC* dual floating-point unit (FPU) known as the *double hummer*. The combination of `-O3` with `-qarch=440d` gave the fastest code using our standard test case, but only slightly faster (by 1%) compared with `-O3` and `-qarch=440`. This indicates that the compiler was not able to find much fine-grained parallelism in the code, especially with the strict constraints of data alignment that are needed for using the dual FPU. The initial port of GTC to the BG/L system showed a highly scalable trend, with performance matching that of the IBM SP POWER3 system (see Figure 4).

Following the first port just described, one of us (Dr. Walkup) carried out a full performance analysis of GTC on the much larger BGW system. A straightforward MPI trace obtained by linking the code with `libmpitrace_f.a` showed that GTC has a small ratio of communication to computation ($\sim 4\%$) for the test case used in this work. However, this MPI trace was performed on 512 nodes; thus, it may not reflect the communication ratio at much higher concurrency. An indication of this was found as we experimented with process mapping, as described

below. Significant improvement in runtime is achieved on 16,384 and 32,768 cores by optimizing process placement. Nevertheless, a communication overhead of even 10% is considered relatively minor and indicates that performance improvements should first focus on the most computationally intensive parts of the code.

Profiling the code with the performance-analysis tool gprof, again on 512 nodes, and examining the results with the IBM tool Xprofiler revealed that the performance issues were mainly in two routines: the charge deposition and the gather-push phase, in which approximately 89% of the time was spent, nearly equally divided between the two routines. The results also indicate that a significant amount of time was spent in intrinsic functions and intensive conversions from floating-point to integer numbers. By examining the statement-level timing data that Xprofiler provides as clock ticks tied to the source lines, it was clear that the push phase extensively used computationally intensive operations such as *sqrt*, *cos*, *sin*, and *exp*. By default, these functions come from the relatively slow GNU libm.a library. However, IBM analysts recommend the use of the highly tuned functions from the MASS (Mathematical Acceleration Subsystem) (libmass.a) instead. Vectorized versions of these mathematical routines also exist in a vector MASSV library that can exploit the SIMD (single-instruction, multiple-data) instructions on the BG/L processor and use the second (double-hammer) FPU. We inserted hand-coded calls to the vector MASSV routines in GTC to replace the default Fortran intrinsic calls in the main loop of the push routine. This change alone gave a performance boost of more than 30% to the overall runtime.

In the charge deposition routine, the analysis showed the existence of potential bottlenecks in computationally intensive conversions, register spills, and pipelining. The spills were detected by examining the assembler section of the routine. Investigation revealed that a code region makes several calls to the *aint()* type-conversion intrinsic function, which involves a slow function call in comparison to the one that is implemented in hardware. Replacing *aint()* with *real(int())* eliminated the unnecessary function call and improved pipelining, giving a significant performance boost to the routine and the code as a whole. Hand-coded loop unrolling was also carried out to optimize register usage and avoid stalls. All of the optimizations described above together improved GTC performance by 60% on the BG/L system.

Finally, it is worthwhile to mention that version 9.1 of the BG/L x1f90 (Fortran) compiler produces a GTC executable that runs 15% faster than the newer version 10 compiler using the same options. Thus, most of the simulations for this work were carried out using the faster executable.

Dual-core performance

Initial GTC experiments were conducted in coprocessor mode, in which one BG/L core is used for computation and the second is dedicated to communication. Additional experiments were then conducted in virtual node mode, in which both cores participate in computation and communication. Impressive results were obtained in virtual node mode, with a per-core efficiency of more than 96%. This indicates that indirect addressing of the gather-scatter PIC algorithm is more limited by memory latency than by memory bandwidth, because the DRAM (dynamic RAM) bandwidth is shared between the two cores. Future work will focus on understanding and optimizing GTC in multicore environments.

Process mapping

The software on the BG/L system offers an easy method for assigning processes to specific processors on the system. This process mapping has been shown to lead to significant performance improvements [30]. For large concurrency GTC simulations, we experimented with process placement, which resulted in significant performance gains. When particles move from one toroidal domain to another, nearest-neighbor communications in a ringlike fashion are used to move those particles to their new processor location. Fortunately, our simulation geometry and size are a perfect match for the BG/L 3D torus network used in point-to-point communication. The layout of a BG/L partition is given in terms of *XYZT*, and although different partitions can be used on the BGW, the *Y* direction can always be chosen to correspond to 32 nodes (assuming that enough nodes are used). Here, the *T* makes reference to the two cores on a BG/L node.

The torus network ensures that node 1 in the *Y* row is connected directly to node 32. Since GTC production simulations are carried out almost exclusively with 64 toroidal domains, a virtual-node mode calculation uses exactly 32 nodes to cover the whole toroidal circuit. By default, GTC assigns its processes by domains so that successive process IDs are in the same domain and have a copy of the same plane. Once the prescribed number of processes per domain is reached, the assignment proceeds to the next domain. This layout was an optimization chosen for systems with large shared memory nodes such as the IBM POWER3 and POWER4* processor-based systems, as well as the Japanese Earth Simulator. Using this layout on larger symmetric multiprocessors allows the expensive *Allreduce* operation in the charge deposition step to be confined within each node.

However, the BG/L system has only two cores per node and uses a different network for collective communications. We can, thus, optimize the process

placement layout by either modifying the source code or setting the environment variable *BGLMPI_MAPPING* to the desired mapping. The ideal mapping was determined to be *XZYT*, or the equivalent *XZTY*, which required an explicit mapping file since there are no predefined strings for these mapping configurations. In our case, all the processes in a GTC toroidal domain were assigned to an *XZ* plane so that processes exchanging particles would be nearest neighbors in the torus network. This optimization led to a surprising 30% improvement in GTC wall-clock (elapsed) time for calculations with 8,192 to 32,768 cores. This indicates that a significant load imbalance was present in GTC with the default mapping. Logically nearest-neighbor processes were most probably mapped to processors that were far apart. This imbalance can be difficult to detect with simple timers because BG/L processors seem to be accumulating CPU time even during a communication transaction. These impressive performance gains highlight the importance of process placement when running on such large processor counts.

Production simulation on BGW

As part of the Blue Gene Watson Day event organized by IBM and the Blue Gene Consortium [31], a 4-hour time slot on the largest BGW queue was allotted to GTC to run a production simulation. As would be appropriate for engaging such a significant allocation, we chose to study the effect of collisions on steady-state microturbulence, a challenging problem that demands use of a very high phase-space resolution. In the actual simulation runs, 1,024 particles per cell (PPC) were used to simulate a laboratory-size tokamak of 0.932-m major radius and 0.334-m minor radius. A total of 2.1 billion particles covered the phase-space volume of the system and deposited their charges on two million grid points divided in 64 planes.

Cyclone is a Department of Energy initiative to provide information on the physics basis and reliability of transport models. The parameters for the well-documented Cyclone base case [10] were used at the mid-radius of the tokamak cross section, $r = 0.5a$ for the temperature- and density-gradient profiles ($R/L_T = 6.9$ and $R/L_N = 2.2$, where R is the major radius of the tokamak, and L_T and L_N are the temperature- and density-gradient scale lengths, respectively). The variable a is the radius of the cross section of the torus. We also used the magnetic shear parameters of [10] [$q = 1.4$ and $(r/q)(dq/dr) = 0.78$, where q relates to the amount of twist in the magnetic field as a function of the radius]. We used the collision model that was the energy-conserving Lorentz operator implemented in a Monte Carlo algorithm. In the PIC approach, the charged particles actually scatter off of the bulk equilibrium plasma instead of the individual particles. We considered an effective

collision time of $\tau_{\text{eff}} = 0.01\tau_{ii}$, where τ_{ii} is the Braginsky ion-ion collision time. The velocity-space nonlinearity term (usually ignored as a higher-order correction) was also included. Retaining this term has been shown to be essential to maintain energy conservation [2, 3]. This simulation was run for 12,000 timesteps in virtual node mode on 32,768 BGW cores. This is the largest number of MPI processes used for a GTC production run. This high-concurrency simulation requires us to maintain a balanced load during this simulation, which is a challenge. After preliminary test runs of a few hundred steps used to estimate the duration, extra time was allocated to guard against the degradation of load balance. For the present studies, the extra time proved unnecessary because the simulation ended very close to the estimated time of 3.5 hours, indicating a near-perfect load balance during the entire simulation.

All GTC simulations start with a very small perturbation in the distribution function of the particles. In practice, this is accomplished by giving each particle a very small non-zero random statistical weight on the order of ± 0.001 . The weight quantifies the departure from the equilibrium distribution [$\delta f/f = (f - f_0)/f$, where f_0 is the equilibrium distribution function]. Under the pressure-gradient drive and the collective motion of the ions, turbulence develops in the form of unstable electrostatic (drift) waves that grow exponentially in the linear phase, during which the heat flux also grows exponentially. Associated finger-like eddy structures form during the nonlinear saturation phase to enable more rapid cross-field migration of thermal energy. This is followed by the possible appearance of turbulence-generated zonal flows, which can develop as a result of nonlinear mode-coupling dynamics and act to shear apart or break the finger-like eddy structures [9, 32, 33]. As the zonal flows continue to grow, the heat flux decreases until both reach a relative steady state. This continuous sequence of events is shown in **Figure 5**, where the time evolution of the heat flux and the zonal flow are shown for three simulations: 1) BGW high-resolution simulation with collisions using 1,024 PPC, and two lower-resolution simulations using 10 PPC to compare results 2) with collisions and 3) without collisions.

Until about 2 years ago, GTC production simulations did not have the compute capability to perform longer temporal runs; that is, modest-resolution cases with 10 PPC were run for about half the number of timesteps of the present studies. These calculations focused more on gaining an understanding of the saturation mechanism during the earlier nonlinear phase rather than on the long-time steady-state phase. In particular, more particles are needed for longer simulations in order to keep the numerical noise low at late times. With today's much faster computers and computers with more processors, we

can afford to run longer simulations with a much higher resolution (i.e., with many more particles). Note that the numerical noise due to the finite number of particles is apparent at late times in the 10-PPC simulations in Figure 5(a) and manifests itself as a high-frequency, low-level component on the heat-flux curves. The zonal flow curves for the same simulations exhibit a stronger oscillation, although the oscillation has the characteristic frequency of the geodesic acoustic mode (GAM). Although this is a physical wave in the system, it is theoretically understood to be either stable or damped. However, in the presence of unphysical dissipation generated by numerical noise, the GAM is observed to grow in amplitude. It is quite significant to observe that our high-resolution, very-low-noise BGW calculation does not display a growing trend for the GAM [blue curve in Figure 5(b)]. This result verifies that the spurious growth of the GAM, caused by numerical noise, can be eliminated in sufficiently high-resolution simulations. The effect of collisions appears during the nonlinear saturation phase, as observed on the zonal flow curves. Without collisions, the zonal flow will grow until it reaches a steady-state value. However, when collisions are taken into account, they act to dampen the flow and keep it at a lower level (i.e., rate). Instead of remaining flat at steady state, the zonal flow now oscillates slowly. This lower level of zonal flow caused by collisional damping [34] has a direct influence on the heat flux since these flows regulate the turbulence observed in the simulations described previously. Instead of settling at a constant low level (red curve), the heat flux now remains higher and exhibits some structure. Further high-resolution simulations will be pursued to complete these important current studies.

Conclusion

In order to further verify, validate, and understand the physics of turbulent plasma behavior on realistic timescales characteristic of actual experimental observations, higher-resolution simulations are needed. This is best achieved by employing many more processors, which will allow the introduction of a much greater number of particles in these simulations. In the longer temporal scale simulations, previously neglected collisional dynamics can become important. Even in the very long mean-free-path thermonuclear plasmas of interest, collisions can, in principle, eventually have a significant impact on the diffusion of particles and energy. The current large BGW capabilities have enabled us to begin to gain realistic insights into these key issues. Petascale resources [i.e., computers capable of delivering a quadrillion (10^{15}) floating-point operations per second] in the near future can be expected to enable a unique and systematic examination of the influence of collisions

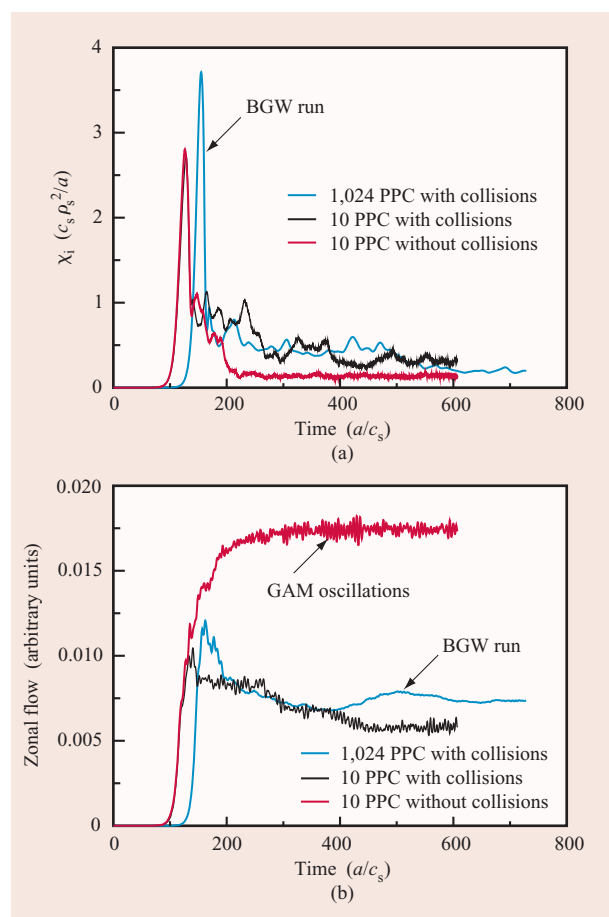


Figure 5

Time evolution of heat flux at mid-radius (a) and zonal flow (b) for the high-resolution BGW simulation with collisions and 1,024 PPC (blue), as well as for two simulations with 10 PPC, with (black) and without (red) collisions. (PPC: particles per cell.) The y-axis label variables include χ (the heat flux), c_s (the ion sound speed), ρ_s (the ion gyroradius at the characteristic temperature), and a (the radius of the cross section of the torus). The subscript “s” indicates that the quantities are calculated using the electron temperature instead of the ion temperature.

on the long-time steady-state plasma transport behavior most relevant to actual experimental observations.

Acknowledgments

Dr. Ethier and Professor Tang thank Professor Zhihong Lin of the University of California, Irvine, and Dr. W. W. Lee from the Princeton Plasma Physics Laboratory for fruitful discussions about gyrokinetic simulations and plasma microturbulence. We are grateful to IBM for hosting the BGW Day event and providing highly valued computer time on the BGW system. Dr. Ethier and Professor Tang are supported by the U.S. Department of Energy Office of Fusion Energy Sciences under contract

number DE-ACO2-76CH03073. Dr. Oliker is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-ACO2-05CH11231.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of the OpenMP Architecture Review Board, Cray, Inc., Silicon Graphics, Inc., Quadrics Ltd., Advanced Micro Devices, Inc., or Intel Corporation in the United States, other countries, or both.

References

1. E. Frieman and L. Chen, "Nonlinear Gyrokinetic Equations for Low-Frequency Electromagnetic Waves in General Plasma Equilibria," *Phys. Fluids* **25**, No. 2, 502–508 (1982).
2. W. W. Lee, "Gyrokinetic Approach in Particle Simulation," *Phys. Fluids* **26**, No. 2, 556–562 (1983).
3. D. H. E. Dubin, J. A. Krommes, C. Oberman, and W. W. Lee, "Nonlinear Gyrokinetic Equations," *Phys. Fluids* **26**, No. 12, 3524–3535 (1983).
4. W. M. Tang and V. S. Chan, "Advances and Challenges in Computational Plasma Science," *Plasma Phys. Control. Fusion* **47**, No. 2, R1–R34 (2005).
5. S. Parker, W. Lee, and R. Santoro, "Gyrokinetic Simulation of Ion Temperature Gradient Driven Turbulence in 3D Toroidal Geometry," *Phys. Rev. Lett.* **71**, No. 13, 2042–2045 (1993).
6. M. Kotschenreuther, G. Rewoldt, and W. M. Tang, "Comparison of Initial Value and Eigenvalue Codes for Kinetic Toroidal Plasma Instabilities," *Comput. Phys. Commun.* **88**, No. 2–3, 128–140 (1995).
7. R. D. Sydora, V. K. Decyk, and J. M. Dawson, "Fluctuation-Induced Heat Transport Results from a Large Global 3D Toroidal Particle Simulation Model," *Plasma Phys. Control. Fusion* **38**, No. 12A, A281–A294 (1996).
8. M. A. Beer and G. W. Hammett, "Toroidal Gyrofluid Equations for Simulations of Tokamak Turbulence," *Phys. Plasmas* **3**, No. 11, 4046–4064 (1996).
9. Z. Lin, T. S. Hahm, W. W. Lee, W. M. Tang, and R. B. White, "Turbulent Transport Reduction by Zonal Flows: Massively Parallel Simulations," *Science* **181**, No. 5384, 1835–1837 (1998).
10. A. M. Dimits, G. Bateman, M. A. Beer, B. I. Cohen, W. Dorland, G. W. Hammett, C. Kim, et al., "Comparisons and Physics Basis of Tokamak Transport Models and Turbulence Simulations," *Phys. Plasmas* **7**, No. 3, 969–983 (2000).
11. F. Jenko, "Massively Parallel Vlasov Simulation of Electromagnetic Drift-Wave Turbulence," *Comput. Phys. Commun.* **125**, No. 1–3, 196–209 (2000).
12. J. Candy and R. Waltz, "An Eulerian Gyrokinetic-Maxwell Solver," *J. Comput. Phys.* **186**, No. 2, 545–581 (2003).
13. Y. Chen and S. E. Parker, "A δf Particle Method for Gyrokinetic Simulations with Kinetic Electrons and Electromagnetic Perturbations," *J. Comput. Phys.* **189**, No. 2, 463–475 (2003).
14. Y. Idomura, S. Tokuda, and Y. Kishimoto, "Global Gyrokinetic Simulation of Ion Temperature Gradient Driven Turbulence in Plasmas Using a Canonical Maxwellian Distribution," *Nucl. Fusion* **43**, No. 4, 234–243 (2003).
15. L. Villard, S. J. Allfrey, A. Bottino, M. Brunetti, G. L. Falchetto, V. Grandgirard, R. Hatzky, et al., "Full Radius Linear and Nonlinear Gyrokinetic Simulations for Tokamaks and Stellarators: Zonal flows, Applied $E \times B$ Flows, Trapped Electrons and Finite Beta," *Nucl. Fusion* **44**, No. 1, 172–180 (2004).
16. J. M. Dawson, "Particle Simulation of Plasmas," *Rev. Modern Phys.* **55**, No. 2, 403–447 (1983).
17. A. B. Langdon and C. K. Birdsall, "Theory of Plasma Simulation Using Finite-Size Particles," *Phys. Fluids* **13**, No. 8, 2115–2122 (1970).
18. H. Okuda and C. K. Birdsall, "Collisions in a Plasma of Finite-Size Particles," *Phys. Fluids* **13**, No. 8, 2123–2134 (1970).
19. M. Kotschenreuther, "Numerical Simulation," *Bull. Am. Phys. Soc.* **33**, No. 9, 2107–2108 (1988).
20. S. E. Parker and W. W. Lee, "A Fully Nonlinear Characteristic Method for Gyrokinetic Simulation," *Phys. Fluids B* **5**, No. 1, 77–86 (1993).
21. Z. Lin, T. S. Hahm, W. W. Lee, W. M. Tang, and R. B. White, "Gyrokinetic Simulations in General Geometry and Applications to Collisional Damping of Zonal Flows," *Phys. Plasmas* **7**, No. 5, 1857–1862 (2000).
22. S. Ethier, W. M. Tang, and Z. Lin, "Gyrokinetic Particle-in-Cell Simulations of Plasma Microturbulence on Advanced Computing Platforms," *J. Phys. Conf. Series* **16**, 1–15 (2005).
23. W. W. Lee, "Gyrokinetic Particle Simulation Model," *J. Comput. Phys.* **72**, 243–269 (1987).
24. J. A. Krommes, W. W. Lee, and C. Oberman, "Equilibrium Fluctuation Energy of Gyrokinetic Plasma," *Phys. Fluids* **29**, No. 8, 2421–2425 (1986).
25. W. W. Lee and H. Qin, "Alfvén Waves in Gyrokinetic Plasmas," *Phys. Plasmas* **10**, No. 8, 3196–3203 (2003).
26. W. W. Lee, "Theoretical and Numerical Properties of a Gyrokinetic Plasma: Issues Related to Transport Time Scale Simulation," *Computer Phys. Commun.* **164**, No. 1, 244–250 (2004).
27. R. B. White and M. S. Chance, "Hamiltonian Guiding Center Drift Calculation for Plasmas of Arbitrary Cross Section," *Phys. Fluids* **27**, No. 10, 2455–2467 (1984).
28. L. Oliker, J. Carter, M. Wehner, A. Canning, S. Ethier, A. Mirin, G. Bala, et al., "Leading Computational Methods on Scalar and Vector HEC Platforms," *SC '05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, Seattle, WA, IEEE Computer Society, 2005.
29. L. Oliker, A. Canning, J. Carter, C. Iancu, M. Lijewski, S. Kamil, J. Shalf, et al., "Scientific Application Performance on Candidate PetaScale Platforms," *Proceedings of the International Parallel and Distributed Processing Symposium*, Long Beach, CA, March 24–30, 2007; see <http://crd.lbl.gov/~oliker/papers/ipdps07.pdf>.
30. G. Bhanot, A. Gara, P. Heidelberger, E. Lawless, J. C. Sexton, and R. Walkup, "Optimizing Task Layout on the Blue Gene/L Supercomputer," *IBM J. Res. & Dev.* **49**, No. 2/3, 489–500 (2005).
31. BlueGene Consortium; see <http://www-fp.mcs.anl.gov/bgl2/>.
32. G. W. Hammett, M. A. Beer, W. Dorland, S. C. Cowley, and S. A. Smith, "Developments in the Gyrofluid Approach to Tokamak Turbulence Simulations," *Plasma Phys. Control. Fusion* **35**, No. 8, 973–985 (1993).
33. B. I. Cohen, T. J. Williams, A. M. Dimits, and J. A. Byers, "Gyrokinetic Simulations of EXB Velocity-Shear Effects on Ion-Temperature-Gradient Modes," *Phys. Fluids B* **5**, No. 8, 2967–2980 (1993).
34. Z. Lin, T. S. Hahm, W. W. Lee, W. M. Tang, and P. H. Diamond, "Effects of Collisional Zonal Flow Damping on Turbulent Transport," *Phys. Rev. Lett.* **83**, No. 18, 3645–3648 (1999).

Received March 19, 2007; accepted for publication May 2, 2007; Internet publication December 18, 2007

Stéphane Ethier *Princeton Plasma Physics Laboratory (PPPL), P.O. Box 451, Princeton, New Jersey 08543 (ethier@pppl.gov)*. Dr. Ethier is a computational physicist in the Computational Plasma Physics Group at PPPL. He received a Ph.D. degree from the Department of Energy and Materials of the Institut National de la Recherche Scientifique in Montreal, Canada. His current research involves high-performance computing and large-scale gyrokinetic particle-in-cell simulations of microturbulence in magnetic confinement fusion devices.

William M. Tang *Princeton Plasma Physics Laboratory (PPPL), P.O. Box 451, Princeton, New Jersey 08543 (tang@pppl.gov)*. Professor Tang is Chief Scientist at PPPL and Associate Director of the Princeton Institute of Computational Science & Engineering at Princeton University. After receiving his Ph.D. degree in physics from the University of California, Davis, in 1972, he advanced to the Principal Research Physicist rank at PPPL and Lecturer with rank of Professor in the Department of Astrophysical Sciences by 1979. He also became a Fellow of the American Physical Society at that time. From 1992 through 2004, he served as head of the PPPL Theory Department, generally recognized as the premier plasma science theory department in the world. He is currently the Director of the Plasma Science Advanced Computing Institute for the Fusion Energy Sciences Program of the Department of Energy. With respect to his research activities, he is internationally recognized for his leading role in the development of the requisite kinetic formalism as well as the associated computational applications dealing with electromagnetic plasma behavior in complex geometries.

Robert Walkup *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (walkup@us.ibm.com)*. Dr. Walkup is a Research Staff Member at the IBM Thomas J. Watson Research Center. He received his Ph.D. degree in physics from the Massachusetts Institute of Technology. He currently works on high-performance computing algorithms.

Leonid Oliker *Lawrence Berkeley National Laboratory (LBNL), 1 Cyclotron Road, Berkeley, California 94720 (oliker@lbl.gov)*. Dr. Oliker is a Computer Staff Scientist in the Future Technologies Group of the Computational Research Division at LBNL. He received his Ph.D. degree in computer Science from the University of Colorado, Boulder. His research interests involve optimization of irregular algorithms on advanced parallel systems, performance evaluation of scientific computations on emerging architectures, and job superscheduling in heterogeneous grid environments.