ADVANCED COMPUTING TOOLS AND MODELS FOR ACCELERATOR PHYSICS *

Robert D. Ryne, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, U.S.A.

Abstract

This paper is based on a transcript of my EPAC'08 presentation on advanced computing tools for accelerator physics. Following an introduction I present several examples, provide a history of the development of beam dynamics capabilities, and conclude with thoughts on the future of large scale computing in accelerator physics.

INTRODUCTION

To begin I want to take you back 37 years to the 8th International Conference on High Energy Accelerators, held in Geneva in 1971. The opening address by Viktor Weisskopf was titled "The Past and Future of High-Energy Physics." Weisskopf wrote [1],

"…something new entered the picture–in this period from the thirties to the fifties, a new type of physicist appeared. No longer do we have only the experimental physicists and the theoretical physicists, but we have a new group which, for lack of a better word, I shall call the machine physicists. Of course we know who is the great, the first one who created that job: it was Ernest Lawrence. And there were many others. I would not like to leave out Stanley Livingston, nor McMillan and Veksler."

At the end there was a question and answer session and, as was customary at the time, it was included in the proceedings. In particular there was a comment from Lew Kowarski. Kowarski played a key role in the founding of CERN, and was named its first Director of Scientific and Technical Services in 1954. He created the Data Handling Division which he directed until his retirement in 1972. He died in 1979. I think his comment after Weisskopf's talk is a fitting opening to this presentation:

"I would like to comment on your three kinds of physicists in a perspective somewhat more extended in time. Early experimentalists worked with their hands: Galileo's legendary tossing of stones from the Tower of Pisa, or the alchemists mixing by hand the ingredients in their mixing bowls. In a similar way the theoreticians manipulated their numerical quantities and symbols by their unaided brain-power. Then came the machines to extend the experimenter's manual skill and to open whole new worlds of things to be handled in ways nobody could predict or even imagine before they really got going. Now we are at the beginning of a new kind of extension by machine: the computer comes to supplement the theoretician's brain. We cannot foresee what this fourth kind of creativity in physics will bring, but we may expect that, just as Ernest Lawrence's contribution was decisive to the development of nuclear machines, the name of John von Neumann will be remembered in connection with the origins of computational physics."

Kowarsky's comments were made in 1971, at a time when the fastest computer in the world was generally viewed to be the CDC 7600. Codes on the CDC 7600 could achieve performance of around 10 million floating point operations per second, or 10 Mflops. Moving forward 37 years to just 2 weeks ago, we see the announcement that the petaflops barrier – the quadrillion flops barrier – has now been broken, by a computer called "Roadrunner" developed by IBM in collaboration with the US DOE and Los Alamos. Roadrunner has roughly 6000 dual-core AMD Opteron chips and 12000 Cell GPU's based on Sony's Playstation 3. Compared to the CDC7600 at the time of Kowarsky, Roadrunner is 100 million times more powerful than the CDC7600.

When Kowarsky said, "*we cannot foresee what this fourth kind of creativity in physics will bring*" he probably did not imagine that such computational power would be available in just 3 decades.

So, with that as motivation, let me present some examples of what is being done now in accelerator modelling on current supercomputers.

LARGE-SCALE MODELLING

Since we are in Italy, I'll begin with an example related to an Italian accelerator project, the Fermi@Elettra project to design and construct a 4th generation light source. Recently simulations have been performed by Ji Qiang of LBNL, using the IMPACT-Z code, with up to 5 billion macroparticles. These simulations showed that it takes around a billion macroparticles to compute the uncorrelated energy spread in such a way that the result does not change significantly with macroparticle number. While huge numbers of macroparticles are not needed in all beam dynamics simulations or to compute all quantities (rms envelopes, for example, generally require a much smaller number), there are some problems, like this one, for which very large scale modelling in essential. Parallel beam dynamics codes such as IMPACT-Z, IMPACT-T, ELEGANT, OPAL, ORBIT, and SYNERGIA, are now able to perform simulations with hundreds of millions, and in some cases billions, of macroparticles. In many cases this is equivalent to the actual number of physical particles. Consider that a nanoCoulomb bunch in an electron linac contains about 6 billion particles.

There has been remarkable improvement in the simulation of beam-beam effects in colliders in the last decade. In the 1990's such simulations were mainly weak-strong simulations (i.e. simulations that were not fully self-consistent) that used simplified models. Compare that with the first million-particle, million-turn simulation of beam-beam effects (in the LHC) performed

*This work was supported by the Director, Office of Science, Office of High Energy Physics, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

in 2003 using the BEAMBEAM3D code. Parallel codes such as BEAMBEAM3D and the NIMZOVICH package have been used to study the parameter space of many colliders to find improved working points and optimize luminosity. Modern beam-beam codes can model head-on collisions, crossing-angle collisions, long-range collisions, crab-crossing, and multiple bunches in a ring interacting at multiple collision points.

Two more important "firsts" occurred recently using parallel beam dynamics codes that illustrate the increasing complexity of modern beam dynamics models: in 2005, the first parallel simulation of multi-bunch injection from a linac to an accumulator was performed using the Synergia package, and this year the first parallel beam dynamics simulation of neighboring bunch effects in a cyclotron was performed using OPAL-CYCL.

Next consider electromagnetic modelling. In the early 1990's simulations usually assumed cyclindrical symmetry, had stair-step boundaries, and could compute cavity eigenfrequencies to around 0.1% accuracy. Today fully 3D electromagnetic simulations can be performed of structures with extremely complex geometry, and eignmode calculations can achieve accuracies of 10 parts per million. In fact, present-day calculations can now be performed, with accuracies that are, in many cases, better than fabrication tolerance. Some examples of parallel electromagnetic modelling codes are OMEGA3P, T3P, VORPAL, ACHERON3D, FEMAXX, GdfidL, and PBCI. As an example of what is now possible, consider that the code OMEGA3P was recently used to compute eigenmodes in a 3D model of an 8-cell cryomodule for a proposed linear collider; this simulation used 300 GB of memory and required 1 hr of CPU time on 1024 processors on a Cray XT4 at NERSC.

Parallel codes are also being used for high fidelity modelling of laser- and plasma-based accelerators. Some examples include OSIRIS, QuickPIC, and VORPAL. Recently OSIRIS simulations have been performed using a grid of size 4000x256x256 to study laser wakefield accelerators (LWFAs). Parallel simulations of laser- and plasma-based accelerators are being used to help understand the underlying physics of LWFAs in the blow-out regime, to design 100 GeV LWFA stages, to provide feedback to experiments, to design plasma wakefield accelerator (PWFA) afterburners, and to design stable, low energy spread, staged systems of LWFAs.

So far I've mentioned that parallel codes are being used to design and optimize linacs and colliders; to design and optimize complex electromagnetic structures; to explore the complex physics of advanced accelerator concepts based on lasers and plasmas, and to design and optimize the next generation of experiments based on these concepts. Before closing this section of my talk and want to briefly mention some other important applications. Parallel codes are being used to explore and understand multi-species instabilities, and to develop designs to control them. Examples include the simulation of the electron-proton instability in the SNS accumulator ring using ORBIT, and the simulation of electron-cloud effects at LHC using the WARP-POSINST code. Parallel simulations using VORPAL are being used to explore the phenomenon of electron cooling with emphasis on the proposed electron cooling system at RHIC. Parallel simulations using IMPACT-Z are being used to explore space-charge driven emittance exchange in rings. And the WARP code is being used in applications for Heavy Ion Fusion and High Energy Density Physics.

HISTORY

Obviously the remarkable advances that have taken place in computational accelerator physics could not have been made through advances in computer hardware alone. New physical models, computational methods, numerical algorithms, and tools for analyzing the huge amounts of data now being generated, have been crucial. Next I will present some history on the development of tools for computational accelerator physics, with a focus on beam dynamics.

This discussion will focus on advances from the 1970's onward, but of course the story of modelling, understanding, and predicting particle trajectories goes back much, much further than that. Some would say it goes all the way back to Maxwell, Lorentz, and Poincare. With the steady development of particle accelerators from the 1930's onward, people became very interested in understanding the linear and nonlinear properties of orbits, and in understanding the stability of orbits. While preparing for this presentation I went over many old proceedings from accelerator conferences. One person who was both incredibly prolific and arguably ahead of his time was L. Jackson Laslett, who was first at Ohio and then moved to Lawrence Berkeley Lab. In looking through old proceedings, Jackson seems to be one of the early people to emphasize accelerator computations. There is a paper in a 1956 proceedings in which he describes particle tracking calculations using the ILLIAC computer.

Arguably the most important of the very early beam dynamics codes was the TRANSPORT code developed by Karl Brown and his collaborators. Though the roots of the code date back to the 1960's and earlier, the 2nd order version was released around 1969.

In TRANSPORT, the particle motion was represented by a 2nd order power series around a reference particle. The power series was in 6 variables, namely 3 coordinates and 3 momenta.

But there's a problem with this power series. If truncated, it is not symplectic, that is, it does not have the Hamiltonian properties that are characteristic of the true dynamical system of a charged particle in electromagnetic fields. While some people appreciated very early the importance of being symplectic – it's clear from Jackson Laslett's papers that he did – I think most people did not appreciate it until the 80's. For example, I can remember being at Los Alamos and seeing simulations of trajectories of particles in the Proton Storage Ring. The simulated particle trajectories were unphysical, they spiraled into the origin, and it was due to numerics, not physics.

A sea change occurred in the accelerator community in the 1980's. Though many people played a role, Alex Dragt was clearly the pioneering figure in the development and establishment of Lie algebraic methods in accelerator physics. Trained as a particle physicist, Alex had a strong mathematical background and he had been working on these tools in the 1970's. A famous result was published in 1976, that of the Dragt-Finn factorization. In that paper, Alex and John Finn showed how an origin-preserving symplectic map could be uniquely represented as a product of Lie transformations. Then, in 1980 Alex spent a sabbatical working in Richard Cooper's accelerator physics group at Los Alamos; while there he realized that the tools he had been developing were ideally suited to analyzing and predicting beam dynamics in accelerators. It turned out that, if one expanded Alex's Lie product as an infinite series, one could identify a one-to-one correspondence between the tensors in Karl Brown's Taylor series and the Lie algebraic polynomials in Alex's expression. But the Lie product is symplectic even if the product is truncated. Furthermore the Lie algebraic approach used the minimal number of free parameters required to specify a symplectic map to a given order. Alex went on to find ways, using generating functions, to evaluate the action of Lie transformations without expanding them, thereby creating a transformation that had the correct linear and nonlinear properties to a specified order, but that was symplectic to *all* orders.

These tools became the foundation of a new code, the Maryland Lie algebraic beam dynamics code, or MaryLie. This was initially developed by Alex and has first graduate student, David Douglas, around 1980. Along with a 3rd order library of beamline elements in Lie algebraic form, it contained routines for symbolically computing Poisson brackets, for manipulating and combining maps, and for performing truncated power series algebra.

The 1980's also saw the development of the program "MAD" (Methodical Accelerator Design) by F. Christof Iselin and his collaborators at CERN. The MAD code also ushered in the notion of a standard input language for use by the accelerator modelling community to describe an accelerator lattice.

I mentioned the code TRANSPORT as being developed around 1970, and the Dragt-Finn factorization being published in 1976 – an interval of 7 years between them. Another key advance came 7 years later again, when, in 1983, Alex and his graduate student Etienne Forest published a paper showing the differential equations governing the Lie algebraic polynomials [2]. This opened the door to computing transfer maps for non-ideal beamline elements such as magnets with fringe fields. My initial contribution to accelerator physics came in 1985 when, as a graduate student in Alex Dragt's group, I implemented these equations in the GENMAP routines of MaryLie, and used them to compute transfer maps for rare earth cobalt quadrupole magnets and maps for dipole magnets with fringe fields.

Three more key developments occurred in the 1980's :

- There was the maturation of normal form techniques, developed by Alex Dragt, Etienne Forest, and others. Normal form methods represented a powerful means of analyzing and designing circular accelerators. The original, famous paper of Courant and Snyder, published in 1958, demonstrated the concept of linear lattice functions in accelerator physics. Though people had tried to generalize it to high order, with varying degrees of success, the development of Lie algebraic-based normal form techniques finally solved the problem completely.
- There was significant progress in the developed of symplectic integrators throughout the 1980's and into the 1990's. Ron Ruth published a paper with a 3rd order integration algorithm in 1983. Etienne Forest and Ron Ruth published a 4th order algorithm in 1990. Also in 1990, Yoshida and Suzuki published an algorithm for taking a time-reversible symplectic algorithm of order 2n and composing it with itself to produce an algorithm of order 2n+2.
- In the late 1980's Martin Berz introduced differential algebraic (DA) tools into accelerator physics. Using Martin's differential algebra package, it became possible to compute transfer maps to arbitrarily high order. Martin implemented this capability in his code COSY-infinity.

Now let me take a step backward in time and point out that, along with the tremendous advances in single particle dynamics, during the 1960's through 1980's there was also a separate activity aimed at modelling beams with space charge. In the USA the most widely used of these codes were developed at Los Alamos and went by the names PARMILA, PARMELA, and PARMTEQ. I was also interested in space-charge effects, in fact it was a major part of my Ph.D. dissertation. Being from those of the "Maryland school" who spent summers at Los Alamos, I can remember thinking about how much effort the space-charge code developers put into modelling space-charge effects, but how they seemed to be not fully aware of certain advances in high order optics. I can remember them working to put new capabilities into their codes, like fringe field effects, and my being surprised because the problem had essentially been solved by Alex and his colleagues. This problem of combining high order optics with space-charge effects in beam dynamics simulations would be solved in the 1990's.

Another activity that also took place in the 1990's was the beginning of the adoption of parallel computing techniques by the accelerator community. I got my start in parallel computing on the Connection Machine 5 at the Advanced Computing Laboratory at Los Alamos, initially developing a 2D space-charge code called IMPACT. This led to a DOE Grand Challenge and extension of IMPACT to 3D by the mid 1990's. To the best of my knowledge IMPACT was the first, production, parallel beam dynamics code in the accelerator community. Originally developed to perform large-scale simulations of space-charge effects in linacs, it used numerically computed transfer maps to describe rf cavities. It also used a (Lie algebra inspired) split-operator technique that made it possible for the two main bodies of work mentioned previously – beam optics and space-charge effects – to be combined into a single, coherent capability. This was the predecessor of the MaryLie/IMPACT code that combined the high-order optics capabilities of MaryLie with the 3D space-charge capabilities of IMPACT.

The Grand Challenge of the 1990's represented the first DOE-sponsored activity for parallel code development for accelerator modelling. It was performed in collaboration with my colleague Kwok Ko of SLAC along with researchers at UCLA and Stanford. This was followed by a DOE SciDAC project that was a major, community-wide effort in the USA. Currently a second SciDAC project is underway, the Community Petascale Project for Accelerator Science and Simulation (ComPASS), led by Panagiotis Spentzouris of Fermilab.

In this section I have described some of the history of the development of beam dynamics codes and methodologies. No doubt new approaches will be developed that, along with advances in computer hardware, will enable continued advances in computational accelerator physics. Here I will mention just 3 areas of current activity.

First, in 2007 Jean-Luc Vay of LBNL recognized that by a judicious choice of reference frame (i.e. choice of Lorentz transformation) certain calculations could be performed orders of magnitude more quickly than if performed in the beam frame [3]. To date, applications of this new approach have led to 1000x speedup in 3D calculations of the electron cloud instability, a 45000x speedup in a 2D free electron laser test problem, and 2-3 order of magnitude speedup in some laser-plasma acceleration simulations.

Second, integrated Green functions (IGF's) have great potential for improving the accuracy of electrostatic PIC codes that utilize convolution-based Poisson solvers. The IGF approach improves the accuracy of the convolution integral by using basis functions. It maintains high accuracy even when the beam has a large geometrical aspect ratio, a situation which is problematic for non-IGF convolution approaches. Depending on the choice of basis functions the 3D IGF equations may be extremely complicated and time consuming to evaluate; however, they are also easily amenable to parallel computation. The IGF approach is now available in some beam-beam codes and in the codes IMPACT-Z, IMPACT-T, and MaryLie/IMPACT.

Lastly, significant progress continues to be made in the numerical computation of transfer maps. In particular, there is much progress in the computation of maps using field data on a surface the surrounds the beam. A number of people have been involved in this research area including Alex Dragt, Marco Venturini, Chad Mitchell, Peter Walstrom, Dan Abell, and others. Techniques have been developed in which the cross section of the surface may be rectangular, circular, elliptical, and for which the 3D geometry is straight or toroidal or a combination of these stitched together. Ever since the pioneering work of Dragt and Forest already mentioned, we have known how to calculate transfer maps when the on-axis gradients are known. However, it has also been recognized that, to compute high order maps, one cannot simply tabulate the on-axis lowest-order gradient and numerically differentiate the data because of numerical errors associated the procedure. Using surface data and calculating from the outside inward (i.e. from the surface inward to the axis) it is possible to develop a procedure for which one can compute numerically stable, accurate on-axis gradients to high order. This in turn allows the accurate computation of realistic transfer maps including nonlinear effects. Given that modern accelerator projects typically cost hundreds of millions to billions of dollars, it is hard to imagine that a major new accelerator will ever be built without first performing detailed beam dynamics simulations that include the linear and nonlinear properties of realistic beamline elements using transfer maps generated from surface data.

THE FUTURE

To conclude, let me speculate about the future. As I mentioned at the start of this talk, the last time I spoke at EPAC was in 1998 in Stockholm. At that time the number one computer on the top500 list [4] was the Intel ASCI Red system at Sandia National Laboratory, with a performance of 1.3 Tflops using 9000 processors. Last week at the 2008 International Supercomputing Conference in Dresden, Germany, the IBM/DOE "Roadrunner" system claimed the #1 spot at 1 petaflop. Already people are thinking about the future. Soon after the Roadrunner announcement of achieving a petaflop, headlines appeared [5,6] including "Welcome to the Post-Petaflop Era," and "All hail RoadRunner's petaflop record – now, what about the exaflop?"

This brings up the question, how will we, as accelerator modelers, use the supercomputers of the future? Since clock speeds have levelled off there has been a trend toward multi-core architectures. High end computing systems now have a larger total number of processors than they used to. Many supercomputers now have 10's of thousands of processors, and systems like IBM's BlueGene are already in use that have more than 100 thousand processors. A computer has already been proposed for climate modelling that would utilize 20 million embedded microprocessors [7].

I believe that the accelerator community will use future petascale and exascale systems in several ways. First, there is a clear need for parameters scans and design optimization. I expect that accelerator designers will perform studies in which a single accelerator system or accelerator component simulation might require a few thousand processors to a few 10's of thousands of processors, and we will make simultaneous use of 100's of thousands or millions of processors by running multiple single-point simulations within a single executable binary. Second, future peta- and exascale systems will open the door to using new algorithms and models that are deemed too challenging for terascale systems. For example, a 3D direct Vlasov code (i.e. a code in which a 6D phase space grid is used) could be run on a petascale or exascale system. Consider that a 6D grid with a grid size of 128 in each dimension has 4 trillion grid points. I would argue that when PIC codes are run with around a trillion macroparticles, at that point (and probably sooner) 3D Vlasov simulations will begin to look attractive. On the topic of new algorithms, I expect that we will also see new models and new codes for modelling coherent synchrotron radiation in 3D. Lastly, given the extreme power of even modest platforms of the future – soon it will be possible to purchase a teraflop of computing power for around 1000 Euros – I believe that there will be new opportunities for using advanced simulation models in accelerator control rooms and control systems.

CONCLUSION

Since EPAC'08 is being held in Italy, I started this presentation with an example of state-of-the-art modelling for an Italian accelerator project (the Fermi@Elettra project). To conclude I want to present a quote from an Italian scientist, not an accelerator physicist, but a chemist named Enrico Clementi. He is quoted as having said [8],

"I know how to get four horses to pull a cart, but I don't know how to make 1024 chickens to it."

This was in support of a similar comment made by Seymore Cray who was not a fan of parallel computing. If one looks at the top500 list, one can see that massive parallelism now reigns supreme. But one cannot discount the comments of Clementi and Cray. It has been difficult to achieve high sustained performance on parallel computers, and it continues to be so. If one considers a machine like Roadrunner, the animal analogy is even more daunting: 6000 chickens (dual-core Opterons) are running the show, and 12000 horses (the PowerXCell 8i processors) are doing the heavy lifting!

Despite the challenges of extreme parallelism, I am optimistic. One simply has to look back a few years to see how far we have come and one cannot help but be excited about where we will be in a few year's time. When I spoke at the US Particle Accelerator Conference in 2001 I showed a photograph of the IBM Power3 system that was being installed at NERSC; the initial system had a peak performance of 3.4 teraflops and filled an entire room. By comparison, later this year it will be possible to buy a GPU-based floating point unit that fits in a person's hand, slides into a PC slot on a personal computer, and has a performance of a teraflop!

I already mentioned that when I spoke at my first EPAC in Stockholm in 1998 the fastest computer in the world achieved about a teraflop. Today, it is a petaflop. If this trend continues, and if people enjoyed this talk enough to invite me back in 10 years, I will be able to report that the exaflop barrier has been broken, and furthermore I'll be able to report on the many new things that computational accelerator physicists will have done on petaflop systems, which by that time will be ubiquitous. And if the trend continues, young people in this audience will see zettaflop and yottaflop computers in their lifetimes.

To quote Kowarski's words, "*We cannot foresee what this … kind of creativity in physics will bring,*" but it is certain to be exciting.

ACKNOWLEDGEMENTS

Many people provided me with material that was presented in my talk. I want to thank A. Adelman (PSI), J. Cary and D. Bruwhiler (Tech-X), A. Friedman (LLNL), C. Geddes, D. Grote, J. Qiang and M. Venturini (LBNL), E. Gjonaj and T. Weiland (TEMF), L.-Q. Lee and C. Ng (SLAC), W. Mori, C.-K. Huang, and M. Tzoufras (UCLA), E. Stern, J. Amundson, and P. Spentzouris (FNAL). Much of the work presented here from US laboratories and universities was performed with support and resources of the US DOE, Office of Science, Office of High Energy Physics, Office of Nuclear Physics, Office of Basic Energy Sciences, and Office of Advanced Scientific Computing Research. Much of the work was performed under the auspices of the US DOE SciDAC program, including the current project, Community Petascale Project for Accelerator Science and Simulation (ComPASS). Many of the results made use of the computer Franklin and other resources of the National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Laboratory. Results were also presented that made use of the computer Phoenix at the National Center for Computational Sciences (NCCS) at Oak Ridge National Laboratory and the ATLAS cluster at Lawrence Livermore National Laboratory.

REFERENCES

[1] Proc. 8th International Conference on High-Energy Accelerators, M. Hildred Blewett, Editor (1971).
[2] A. Dragt & E. Forest, J. Math. Phys. 24, 2734 (1983)
[3] J.-L. Vay, Phys. Rev. Lett. 98, 130405 (2007)
[4] http://www.top500.org/
[5] http://www.hpcwire.com/blogs/19752629.html
[6]http://www.computerworld.com/action/article.do?articleId=9095279&command=viewArticleBasic
[7] http://www.lbl.gov/CS/Archive/news050508.html
[8] http://softwareblogs.intel.com/2006/10/27/some-concurrent-beasties/