THE MAINSTREAMING OF INTERVAL ARITHMETIC

John L. Gustafson
Sun Microsystems, Inc.

Abstract Interval arithmetic and validated arithmetic methods are almost unknown in the United States, and are absent in the federally funded High Performance Computing efforts of the last twenty years. The focus has been on floating-point operations per second (FLOPS) to the exclusion of any concern for the correctness of the result. However, the treaty-mandated need to validate nuclear weapons without physical experiments (the ASCI program) may prove to be the key to changing this. Radiation transport provides an example where bounded intervals can provide much more useful answers than existing point methods, whether they are used for modeling nuclear reactions or for computer-generated graphics. This example, and others, can be used to illustrate a general strategy that will allow us to move interval arithmetic into the mainstream of high-speed computing.

1. INTRODUCTION

Why hasn't interval arithmetic become a widely adopted technique?

Much of the effort of the last several decades of interval arithmetic research has focused on the making interval bounds tighter for mathematical problems. It has not yet addressed what is required to replace *physical simulation* point algorithms in common use with interval algorithms. In the area of technical computing, as compared to business use of computers, the results of point algorithms are generally regarded as advice to an engineer or scientist and not the proof of a result.

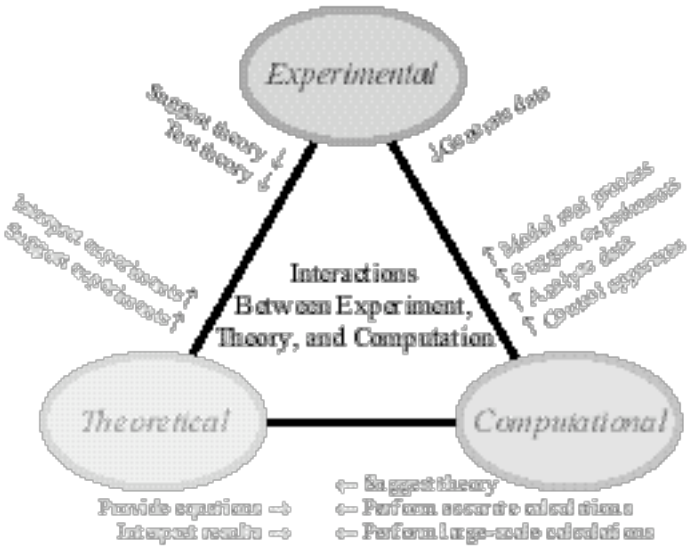

Figure 1. Computational Science as a Third Branch of Science

While only a very small percentage of technical computer users have knowledge of numerical analysis, most understand that rounding, discretization, and coding errors are ever-present hazards. Because of the other vertices of the triangle, experiment and theory, the error-prone nature of computational science has not prevented it from being a useful approach.

The ASCI (Accelerated Strategic Computing Initiative) program initially seemed like it might provide a watershed for this attitude. In 1996, international treaties demanded cessation of all experimental testing of nuclear weapons [6], so the ASCI program was begun to provide a way to *completely* replace experiments with computations. This decision potentially represented as great a change in the business of science as was the rise of the experimental method during the Renaissance.

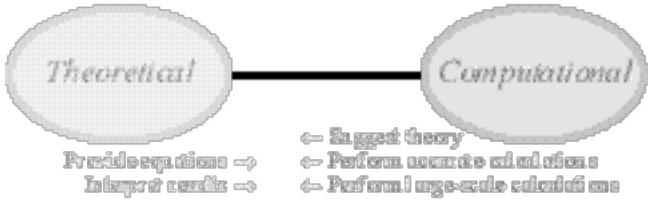

Figure 2 "Experimentless Science"

Unfortunately, instead of making use of the well-established body of knowledge of interval arithmetic and verifiable computing, the ASCI program has retreated to comparing conventional point algorithms to historical experiments and some near-critical nuclear experiments, instead of directly

addressing the issue of how to obtain certainty from computer simulations alone. Perhaps verifiable computing techniques will be invoked later in the ASCI program, or perhaps the independent software vendors (ISVs) of technical software will lead the process of adoption; here we discuss some of the fundamental obstacles that must be overcome in either case.

2. MOORE'S LAW AND PRECISION

Pressure to confront issues of computing veracity comes from another source: Moore's law. While not stated explicitly until 1967, the doubling of part density and performance every 18 to 22 months can be backdated at least to the late 1930s with the first binary computers of Atanasoff or Zuse [4]. The Zuse Z1 used 22-bit floating-point arithmetic in 1939, and used six bits to address 64 memory locations. In the year 2000, a large server like a Sun Starfire uses 38 bits to address 128 billion bytes in a linear address space. This represents 38 – 6 = 32 doublings per Moore's law, over a period of 61 years.

Does precision also follow Moore's law? In fits and starts, the size of floating-point numbers has increased along with the number of address bits. The 1977 Cray-1 made 64-bit floating-point arithmetic available with full hardware support, and at the time, it seemed like the 47-bit mantissa (14-decimal) precision would make worries about numerical precision unnecessary. The IEEE 754 floating-point standard extended the mantissa to 53 bits (almost 16 decimals), and both Intel and Motorola have offered processors that have floating-point registers with extra precision for register-to-register scratch calculations.

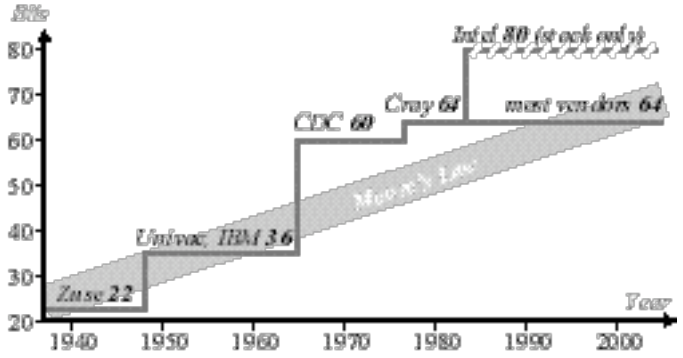

Figure 3. Moore's Law and Floating-Point Precision

With computers that can perform 10^{12} operations per second becoming common, it would appear that we are now due for another step up in precision. A biased loss of only 0.5 ULP per operation means an entire 53-bit mantissa can lose all meaning in only half an hour's worth of arithmetic. However, to change to 128-bit precision is a daunting prospect. Besides halving the number of floating-point data we can hold in a given amount of

memory, the use of 128-bit data doubles the burden on the memory bus and caches, which are already the limiting parts of modern computer architectures. This creates some very practical pressures on computer manufacturers to revisit the issue of veracity in technical computing and explore alternatives.

For example, it may be that intervals with 32-bit precision numbers are much more powerful than point values with 64-bit precision, eliminating the need to increase the number of bits per datum. In trying to persuade users that interval arithmetic might actually be faster than conventional arithmetic, the following are useful starting points for discussions:

- Intervals form a closed algebra that eliminates the need for complicated error trapping; this can save both time and chip area.
- Strassen matrix multiplication becomes numerically safe, allowing substantial speedups for certain applications.
- Fast Fourier Transforms (FFTs) do not require extra precision for the "twiddle factors."
- Sums and dot products can be performed without need for sorting to guard against rounding errors; the interval confidence might suffice.
- Vendors such as Sun are now supporting interval arithmetic in compilers, and to a growing extent, in processor designs.
- Argument reduction for functions like $sin(x)$, $cos(x)$, $exp(x)$, and $log(x)$ is simplified via intervals.
- Compilers can avoid unnecessary precision effort by recognizing imprecise inputs.

These are offered not as theorems, but more as points for discussion. In practice, they may or may not hold. These issues must be explored if interval arithmetic is to enter the mainstream of scientific computing.

3. INTERVAL PHYSICS

Consider the usual sequence by which physical problems are converted to computer programs:

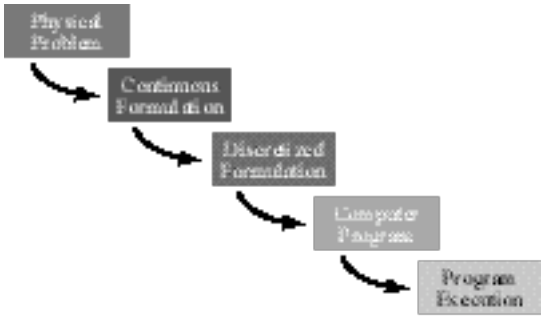

Figure 4. Where to Apply Interval Concepts

The last step shown in Figure 4, between the Computer Program and the Program Execution, is where most of the effort to improve veracity in computing has been. Either by using extra guard bits in temporary calculations or using high-precision mantissas, vendors have sought to improve answer quality without requiring any effort on the part of programmer or user.

Within the Interval Arithmetic community, most of the effort has been one or two levels up from this. For example, one might explore how to perform Cholesky factorization using interval arithmetic, and under what conditions the bounds remain reasonable. An ordinary differential equation with bounded coefficients might be shown to have a bounded solution.

When such accomplishments are shown to technical ISVs, the reaction is predictably unenthusiastic. The ISV developer might have questions like the following:

- Does a particular car design protect the passengers in a 20 m/s crash?
- Will a jet engine vibrate enough to destroy itself at some operating RPM?
- Does the steady-state heat flow in a circuit design permit it to operate correctly?
- What will the weather be, three days from now?

There doesn't seem to be any obvious way to proceed from questions like these to the solutions provided by interval arithmetic. What is needed is not so much interval arithmetic, but what I call *interval physics*. This is an attack on the first two steps of the sequence shown in Figure 4.

We are taught, early in our technical careers, that physical problems are continuum problems; they therefore map to partial differential equations (PDEs), and a solution of those PDEs will predict behavior in a manner that is as accurate as our understanding of the physics. Since closed-form solutions of PDEs are rare, the differential equations are approximated by finite differences that are discrete and amenable to calculation. However, this is exactly the point at which verifiability is lost in the model. Any use of interval arithmetic after this is useless if the desired result is to provably contain the true answer for a physical simulation. No matter how clever we are about controlling the expansion of intervals in an algorithm, we will never convince the mainstream scientific computing community that interval arithmetic is a good idea until we can bound the *physical* behavior of a system.

3.1 Measuring Physical Answer Quality

Suppose the result of a physical simulation is a scalar quantity F that is a function of space and time variables. Suppose further that one can establish by physical and mathematical reasoning that it is impossible for F to be larger than F^+ or smaller than F^-. We can then define the *total error*, E, as

$$E = \iiiint_D (F^+ - F^-)\, dx\, dy\, dz\, dt \tag{1}$$

and the answer quality, Q, as

$$Q = 1 / E, \tag{2}$$

where D represents the domain of interest in space-time.

These two definitions permit scientific comparison of algorithms and a way to assess performance rigorously. Two very different computers using very different approaches to solving a problem can be compared based on the answer quality achieved in a given amount of time. The trick now is to establish $F^+ - F^-$ for practical situations.

3.2 A Radiation Example

Some physical problems can be expressed as *integral equations* instead of differential equations. Integral equations of the second kind are particularly amenable to interval techniques:

$$f(x) - \int_D K(x, s)\, f(s)\, ds = g(x). \tag{3}$$

The $g(x)$ term makes the integral equation inhomogeneous. This equation can be restated in a form that suggests an iterative technique based on initial values for f:

$$f(x) = g(x) + \int_D K(x, s)\, f(s)\, ds. \tag{4}$$

Suppose we can construct bounds $f^- < f < f^+$ based on physical reasoning (such as causality, conservation of energy, etc.) The K and g functions must also be bounded above and below. These all can be very wide intervals initially, corresponding to very low answer quality (as defined in the previous section) if the action of the right-hand side operator above is a contractive mapping. With the bounds initialized, iterate the above equation until the answer quality Q stops improving.

The domain D is now subdivided. It could be subdivided in space or in time; what matters is that once D is split, bounds can be improved on K and g. With the subdivided geometry, the interval representing f can be again refined successively until Q stops improving. Each part of the subdivided domain has a measurable contribution to the error, E. What seems most efficient is to repeatedly subdivide the part of the domain that makes the largest contribution to E, maintaining a sorted queue of subdomains to be split.

Radiosity provides a specific example of this technique [5]. Radiosity problems arise both in radiative heat transfer and in computer graphics where

surfaces are assumed diffuse reflectors. The equation, using conventional variable names, is

$$b(r) = e(r) + \rho(r) \int_S F(r, r')\, b(r')\, dr', \tag{5}$$

where b is the radiation given off at a point, r and r' are points in the set of all reflecting surfaces S, e is the radiation that is emitted (as a heat or light source, not simply a reflector), ρ is the reflectivity at a point, and $F(r, r')$ is the *form factor* that determines how much radiation from point r' is received at point r.

For a given geometry, e is stated as an exact value and not an interval. The reflectivity ρ is between 0.05 and 0.95 in practice. Empirically, a perfectly "white" surface reflects no more than 95% of light that strikes it, and a perfectly "black" surface always reflects at least 5% of the light that strikes it. The form factors are between 0 (completely occluded) and 1 (completely visible) if integrated over r'. If we solve

$$b_{max} = e_{max} + \rho_{max}\,(1.0 \times b_{max}), \tag{6}$$

we obtain $b_{max} = e_{max} / (1 - 0.95)$ as a physical upper bound. In other words, the brightest that the radiation can be is if every surface has perfect visibility of the emitting surfaces and the maximum 0.95 reflectivity, and the multiple reflections create the infinite series $e_{max} \times (1 + 0.95 + 0.95^2 + 0.95^3 + \ldots) = e_{max} / (1 - 0.95)$. Similarly, the darkest that the radiation can be is zero, since it is possible for F to be exactly zero. This treats the entire geometry as a single point, so the integral equation becomes a simple scalar equation. Now the geometry can be subdivided, and this improves the bounds on all the quantities in the radiosity integral equation. A simple test geometry in two dimensions is shown in Figure 5.

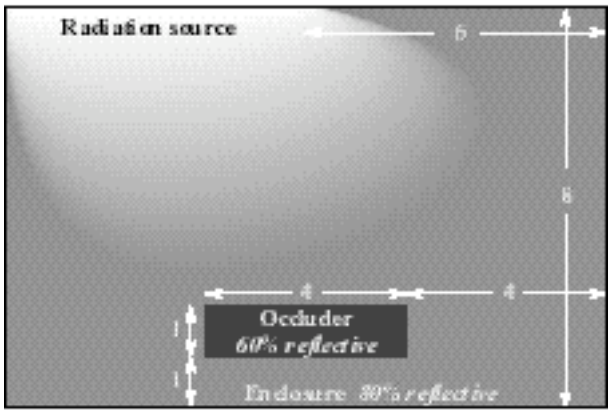

Figure 5. Simple Radiosity Test Geometry

Results after several iterations are shown in Figure 6 that produce a progressively better set of intervals that bound the radiation, as the *z* axis of Figure 5:

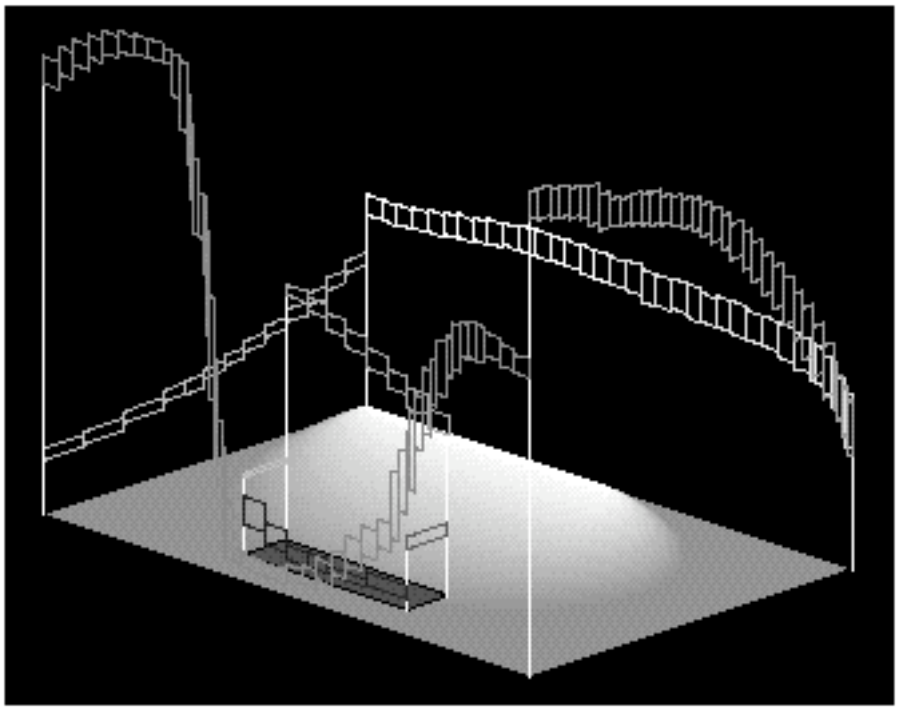

Figure 6. Rigorously Bounded Radiosity Solution

The bounds can be made arbitrarily tight with further iterations. Interestingly, this research revealed that the so-called "hierarchical radiosity" methods [3] for radiosity are very inefficient at obtaining high answer quality compared to the method described here. While the work done on a system of N patches is order N, this is accomplished not by minimizing work but by proliferating patches. The $Q = 1/E$ definition of answer quality is quite useful for resolving arguments about the superiority of any particular approach; the traditional $O(N)$ arguments can be misleading because they equate answer quality with the number of discrete variables, which may have nothing to do with the confidence in their computed value. Figure 7 shows the error bounds that result from the "hierarchical radiosity" method, using the same number of operations as was used to obtain Figure 6.

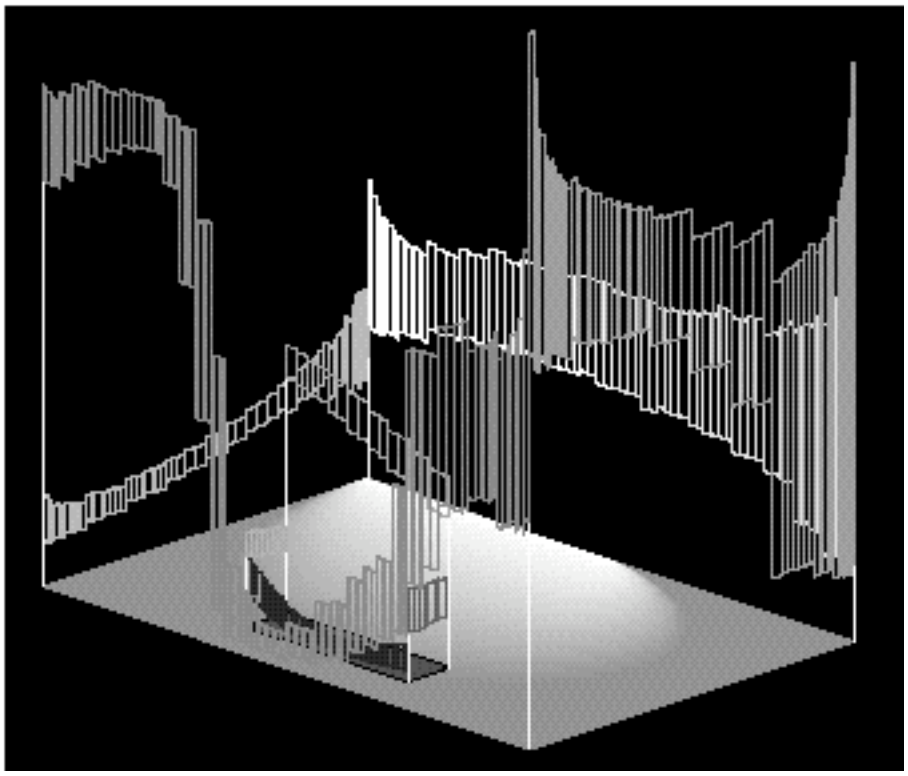

Figure 7. Poorer Bound via "Hierarchical Radiosity"

3.3 Bypassing PDEs

Consider the limiting process used in deriving a PDE and then a difference equation. The physics is assumed continuous, the rules of calculus are applied to the continuous functions that represent the variables and their derivatives, and then the difference equations are derived as approximations to the differential equations.

It may be better practice to bypass the PDE formulation and *discretize the physics directly*. Mass distributions can be regarded as collections of point masses with potential forces between them ("large atoms"). These can be made smaller as more computing power is available, moving the limiting process to the physics and not to the difference approximation of the PDE. Similarly, electromagnetic fields might be modeled as quanta, or fluid dynamics as interacting particles. In a way, this is what lattice-gauge theories do.

The advantage of this approach is that it permits provable bounds on physical simulations via interval physics. Interval arithmetic can then be used on the interval physics, which insures that rounding errors cannot spoil the containment of the answer. In creating an example to illustrate this approach

of “bypassing PDEs,” I stumbled on a surprisingly simple case where the two limiting approaches do not yield the same answer, and for which the traditional PDE approach is the one with more dubious approximations: A one-dimensional wave equation with an initial velocity given by a step function. That is, a stretched string set in motion by being struck with a hammer.

In the traditional solution [1], the PDE is exactly solved by functions of the form $y = f(x+ct) + g(x–ct)$. This gives rise to the following familiar-looking time sequence:

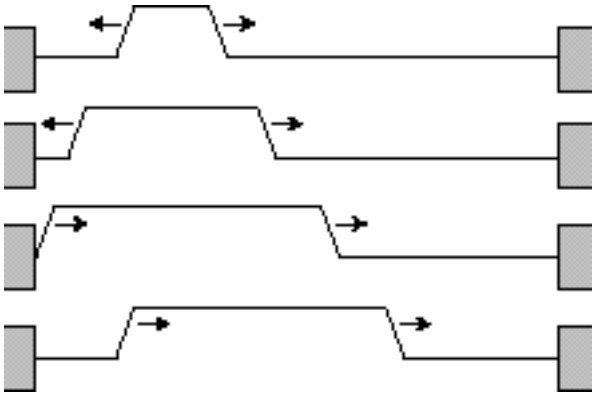

Figure 8. One-Dimensional Wave Equation by Traditional PDE

Suppose instead that we model the string as a set of point masses on springs. By bounding the behavior based on conservation of energy (spring potentials plus the kinetic energies of the masses), momentum (components in x and y), the string behaves in a strikingly different manner:

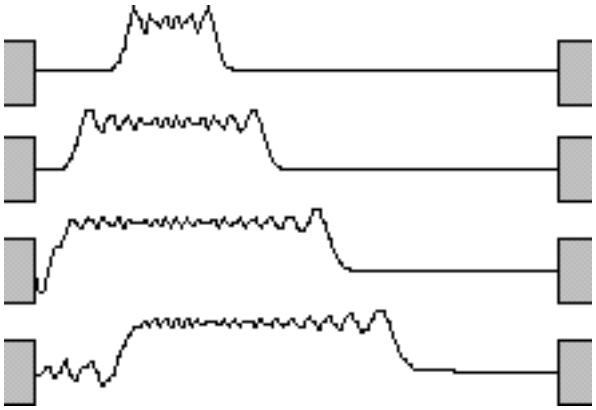

Figure 9. Wave Equation as the Limit of Point Mass Behavior

One might recognize this sort of "ringing" behavior as one resembling the Gibbs Overshoot Phenomenon, the high-frequency oscillations that never disappear when approximating a step function with a Fourier series. How can this be valid when the PDE shows something so different?

By taking the limiting process of turning the string physics into a PDE, one masks the error of using a forcing function that has discontinuities. Yet, it is reasonable that one could accelerate a string with a hammer that touches the string completely at one point but not at all at the adjacent point. Doing so produces "ringing" that does not disappear as the number of mass points increases. Of course, a realistic model would model the string as a cylinder of mass points with intermolecular forces that give rise to stiffness and other complications.

This example suggests that in addition to providing rigorous bounds for physical simulations, the interval physics approach can sometimes yield insights missed by approximations to PDE formulations.

3.4 The *N*-Body Problem

Interval physics can be applied to the *N*-body problem, though the method of getting an initial bound on the particle motion is not obvious. The difficulty arises in the fact that the potential function for inverse-square attractive forces has a singularity if any two particles contact each other. If that happens, their velocity goes to infinity. For any time step size, the possibility of infinite velocity exists, so there are no bounds. While there is ample literature for the bounding of ordinary differential equations (ODEs) by interval methods, it is usually assumed that the coefficients of the ODE can be bounded.

The key is to use *space steps* instead of time steps. Let r be the minimum separation between any pair of particles in the ensemble. Before any particles can move into contact with each other, they would first have to move a distance $r/3$. We can therefore put a bounding sphere of radius $r/3$ around every particle, and ask what is the shortest time t that would allow any particle to move that far. Since the forces on particles can be bounded within the range of those spheres, the acceleration of any particle is also bounded, and this can be used to establish the timestep. Time becomes the dependent variable, not the independent variable, if the calculations are to be rigorous. Because the minimum time must be greater than zero, the simulation will make progress. The amount of temporal progress will vary from one space step to the next.

The bounding spheres can be reduced in size iteratively for that timestep, since each particle's range of motion can be made progressively more precise which in turn allows the forces to be bounded more precisely for all the other particles. The error is the sum of the volumes of all the bounding spheres for all the time steps, so once again we can precisely define the answer quality of the calculation. It is worth noting that doubling the answer quality of the N-body problem seems to require quadrupling the amount of work. The arguments of Greengard, Barnes-Hut, Appel *et al.* regarding clustering methods that reduce the force computations from $O(N^2)$ to $O(N\log N)$ or even

$O(N)$ [2] do not take the time stepping error into account; they only state an error bound for a static force calculation. Again, we see that the use of interval physics and rigorous a definition of simulation quality shed insight, on the relative merits of algorithms that may be contrary to commonly held beliefs.

The solution of the N-body problem by rigorous methods points the way to interval physics methods for computational fluid dynamics, structural analysis, and material property calculations. They missing piece is a definition of the potential function. Complicated chemical interactions make the simulation, say, of an ensemble of water molecules a very challenging proposition. If firm bounds can be found on the potential functions, it might be possible to do accurate fluid simulations without the need for the Navier-Stokes equations. The uncertainty of the potential would make clear what the limits of predictability in the fluid simulation are.

3.5 "Conservation Laws"

In trying to find interval physics formulations, every situation is different. Here are some examples of "conservation laws" that can be useful in bounding physical behavior:

- Energy, momentum, and mass are conserved (non-relativistic)
- Surfaces cannot reflect more radiant energy than they receive.
- Before a particle can move a distance x, it must first move distance $x/2$.
- Probabilities are greater than zero and less than one.
- The time to do a task cannot be negative.
- Group wave velocity cannot be greater than c.
- Local minima or maxima cannot occur in a Laplacian interior
- Effect cannot precede cause.
- The energy at one point cannot exceed the energy of a closed system.
- The Mean Value Theorem holds for continuous physical variables.
- Rigid bodies cannot occupy the same region of space.
- Coefficients of friction cannot be negative.

…and so forth. These indicate the style of thinking that is needed to formulate simulation problems in a manner that will allow the application of interval physics, and thus interval arithmetic. The use of interval physics controls the discretization errors and other uncertainties, while the interval arithmetic controls the rounding error to preserve containment of the answer as the computation progresses.

4. SUMMARY

The main hurdle to the mainstream use of interval arithmetic is the adoption of the methods by commercial vendors of scientific simulation

software. While Moore's law and the market for experimentless computing are applying pressure to find verifiable ways of computing, and vendors like Sun have supplied the tools in language compilers, there has not been a way to replace conventional PDEs with interval methods, and thus interval computing methods have remained confined to a small but ardent group of followers.

If we elevate the point at which interval-type thinking is introduced when a physical problems is being converted into a computer algorithm, we may find that the PDE formulation can be avoided altogether and "interval physics" used instead. The success of the small-scale programs discussed here suggests that it is time to attempt a simulation code that is on the scale of a NASTRAN or an LS-DYNA or a FLUENT, with the goal of determining if the problems these programs address can instead be rigorously solved using methods that provably contain the answer. Accomplishing this will be as significant as the introduction of the experimental method, since it will elevate computational science to the same standing as experimental and theoretical science as a means of obtaining scientific truths.

Acknowledgments

While at Ames Laboratory, the patient work of post-doctoral researchers Quinn Snell and Mary Oman, and research assistant Cassandra Biggerstaff, made possible many of the conceptual breakthroughs for the examples presented here. The constant encouragement of Sun colleague William Walster to develop and present these ideas is what led to the current effort, and for his unwavering zealotry, I am grateful.

References

1. Boyce, W.E., and DePrima, R.C., *Elementary Differential Equations and Boundary Value Problems*, Second Edition. New York: John Wiley & Sons, 1969.

2. Greengard, L. and Rokhlin, V., A fast algorithm for particle simulations. Journal of Computational Physics, 73, 1988.

3. Hanrahan, P., Salzman, D., and Aupperle, L. A rapid hierarchical radiosity algorithm. Computer Graphics 25, 1991; 4:197–206.

4. Rojas, R., and Hashagen, U., editors, *The First Computers: History and Architectures*. Cambridge: MIT Press, 2000.

5. Sillion, F. and Puech, C. *Radiosity and Global Illumination.* San Francisco: Morgan Kaufmann Publishers, Inc., 1994.

6. U.S. Department of Senate, *Daily Press Briefing*, DOT #147, Sept. 11, 1999.