Computational Plasma Physics: Powerful New Tools of Scientific Discovery

Greg Hammett, Princeton Plasma Physics Lab (PPPL)
http://w3.pppl.gov/~hammett

DOE National Undergraduate Fellowship Program
In Plasma Physics and Fusion Sciences, 1-week course, June 2012

Acknowledgments:

Center for the Study of Plasma Microturbulence (General Atomics, PPPL, LLNL, U. Maryland, MIT)

DOE Scientific Discovery Through Advanced Computing
http://fusion.gat.com/theory/pmp

J. Candy, R. Waltz (General Atomics)
W. Dorland (Maryland) W. Nevins (LLNL), S. Jardin, D. Keyes, J.L. Peterson, et al.

(Not all of these slides were shown. Typos fixed 2013.08.01)
Computational Plasma Physics: Powerful New Tools of Scientific Discovery

• Brief intro to computational science

• The importance of good numerical algorithms
  Pitfall of naive algorithms for paradigm advection equation

• Examples of cutting edge computational plasma physics:
  – Simulating 5-dimensional plasma turbulence in fusion devices
  – MHD simulations of edge plasma instabilities

• Computational methods providing powerful tools for searching for ways to improve fusion reactor designs and our confidence in them.
But first another topic:

Brief Review:

Progress is being made in Fusion Energy research, well worth continuing.
Progress in Fusion Energy has Outpaced Computer Speed

Some of the progress in computer speed can be attributed to plasma science.

Princeton’s TFTR made 10 MW for 1 sec, enough for ~5000 Americans.
Fusion can’t be criticized for being behind schedule, because we never got the budget needed.

Cumulative Funding

~$30B development cost tiny compared to >$100 Trillion energy needs of 21st century and potential costs of global warming. Still 40:1 payoff after discounting 50+ years.

(This 2003 figure needs updating, but main point is still true.)
Fusion can’t be criticized for being behind schedule, because we never got the budget needed.

As Einstein taught us, time is relative, and for large projects, it is often important to measure time in units of $. Time is money.
Interesting Ideas To Try To Improve Fusion

* Liquid lithium coatings on walls: (1) protects solid wall from erosion, ELMs (2) absorbs incident plasma, reduces recycling of cold neutrals back to plasma, raises edge temperature & improves global performance. TFTR: ~2 keV edge temperature. NSTX, LTX: more lithium is better, where is the limit?

* Spherical Tokamaks (STs) appear to be able to suppress much of the ion turbulence: PPPL & Culham upgrading 1 --> 2 MA to test scaling

* Advanced tokamaks, studies of methods to controls Edge Localized Modes, alternative regimes (Hybrid scenarios with flattish q profiles) to improve performance

* Tokamaks spontaneously spin, and this sheared flow can reduce background turbulence and improve MHD stability. Can we enhance with updown-asymmetric tokamaks or non-stellarator-symmetric stellarators with quasi-toroidal symmetry?

* Josephine Proll, Per Helander, et al. (Germany) recently discovered a “quasi-isodynamic” stellarator configuration in which all trapped particles have averaged good curvature (PRL 20120). Shuts off trapped particle modes. Combine with Lithium to completely shut off turbulence?
Fusion performance depends sensitively on confinement

Sensitive dependence on turbulent confinement causes some uncertainties, but also gives opportunities for significant improvements, if methods of reducing turbulence extrapolate to larger reactor scales.

\[
\frac{dW}{dt} = P_{\text{ext}} + P_{\text{fusion}} - \frac{W}{\tau_E}
\]

Caveats: best if MHD pressure limits also improve with improved confinement. Other limits also: power load on divertor & wall, …
Improved Stellarators Being Studied

- Originally invented by Spitzer (’51), the unique idea when fusion declassified (’58)
- Mostly abandoned for tokamaks in ’69. But computer optimized designs now much better than slide rules. Now studying cost reductions.
- Breakthrough: Quasi-symmetry discovered in late 90’s: don’t need vector $B$ symmetric exactly toroidally, $|B|$ symmetric in field-aligned coordinates sufficient to be as good as tokamak.
- Magnetic field twist & shear provided by external coils, inherently steady-state. Stellarator can exceed Greenwald density limit, don’t have hard beta limit & don’t disrupt. Quasi-symmetry allows plasma spin to reduce turbulence? Other ways to reduces turbulence?
- Robotics breakthroughs could reduce costs for large complex devices that can’t be mass-produced.
turbulence (1/H) & MHD stability limits ($\beta$) could significantly improve fusion


(Relative Cost of Electricity (COE) estimates in Galambos et al. study, see ARIES reactor studies for more detailed costs estimates.)
Introduction to Scientific Computing and the Importance of Good Numerical Algorithms
Computing has become a powerful 3rd way of Scientific Discovery

- Traditional view: Experiments or Theory
  Scientific method is all about the interaction between the two:

  **Experiments**
  - Reality
  - observation
  - hypothesis testing

  **Theory**
  - hypothesis formulation
  - Insights, systematic laws

- New view: Experiments and Theory and Computing:

  **Experiments**
  - Reality

  **Theory**
  - Exact solutions to approximate Eqs.
  - (or exact Eqs. for simplified systems)

  **Computational Science**
  - Approximate solutions to exact Eqs.
  - (or more complicated systems)
  - numerical experiments / computational theory (oversimplified)
Types of Computational Physics Thesis:

1. Solve a new set of (approximate) equations that describe some physical phenomena of interest

2. Solve an existing set of equations with an improved algorithm and faster computer to study increased range of time and space scales: new phenomena.

3. Take an existing code (and possibly modify it) and use it to perform new physics studies and validation with experiment
## Recent PPPL Graduate Student Computational Thesis

<table>
<thead>
<tr>
<th>Year</th>
<th>Name</th>
<th>Institution (Laboratory)</th>
<th>Thesis Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>Lukin, Vyacheslav</td>
<td>NRL</td>
<td>Computational Study of the Internal Kink Mode Evolution and Associated Magnetic Reconnection Phenomena (#2)</td>
</tr>
<tr>
<td></td>
<td>Ferraro, Nathaniel M.</td>
<td>General Atomics</td>
<td>Non-ideal effects on the stability and Transport of magnetized plasmas (#2)</td>
</tr>
<tr>
<td>2010</td>
<td>Smith, Sterling</td>
<td>General Atomics</td>
<td>Magnetohydrodynamic Stability Spectrum With Flow and a Resistive Wall (#1)</td>
</tr>
<tr>
<td></td>
<td>Raburn, Daniel</td>
<td>Japan</td>
<td>Efficient Numerical Calculation of MHD Equilibria with Magnetic Islands, with Particular Application to Saturated Neoclassical Tearing Modes (#2)</td>
</tr>
<tr>
<td></td>
<td>Peterson, Jayson D.L.</td>
<td>LLNL, NIF</td>
<td>Relating gyrokinetic electron turbulence to plasma confinement in the NSTX (#3)</td>
</tr>
<tr>
<td>2012</td>
<td>Baumgaertel, Jessica</td>
<td>LANL, NIF</td>
<td>Gyrokinetic studies of turbulence in stellarators (#3)</td>
</tr>
</tbody>
</table>
As computers get faster, the types of problems we can address changes:
more time + space scales in a single simulation
Computational Physics & Numerical Algorithms are Interesting and Rich Fields

• Amazing exponential growth in computer power means we can now solve many problems that were thought impossible 30 years ago. Computers being applied to many problems of human importance and interest (physics, astrophysics, biology, climate modelling, engineering…) (Careful: there are also many problems that can never be directly solved on computers…)

• Computational work very interesting: sometimes you don’t really understand equations until you get up close & personal with them to solve them numerically. Boundary conditions, conservation laws, other properties that should be preserved…

• Study of Numerical algorithms is a large and rich field: Huge bag of numerical tricks: Many different algorithms highly optimized for different applications. Choice of which features of the original equations you want to preserve most accurately in the discrete numerical approximations:
  – Highly accurate solutions in some regimes but large errors in other regimes vs.
  – Fast and Robust algorithms that have somewhat larger but manageable errors over a wider range of parameters
  – Preserve exact conservation laws or other important properties of real solution? Conserve momentum or energy to round-off error (but sometimes can’t conserve both). Preserve \( \nabla \cdot \mathbf{B} = 0 \) exactly?
Richness of Study of Algorithms

• Vast zoo of algorithms, in part because developed for different applications in different fields (difficulties translating jargon between fields)

• Different algorithms are optimal for different applications, tradeoffs in accuracy, speed, complexity, conservation properties, bounds on solutions (positivity, non-oscillatory, etc.), preservation of other properties, efficiency on different computer architectures, overall robustness.

• Independent codes and/or different algorithms can be useful cross-checks.

• Get deeper insight into equations when trying to actually solve them on a computer
“Robustness”: a Useful Quality for Algorithms

• $\exp(-x^2) \approx 1 - x^2$

  Taylor series expansion, rigorous in certain limits, but behaves poorly in other regimes

• $\exp(-x^2) \approx 1 / (1 + x^2)$

  “Pade approximation” / rational function approximation, just as accurate, but more “robust”, i.e., bounded errors or preserves certain important features of the solution (positivity)

• illustrate with figure...

• (This may seem like a trivial example, but illustrates issues when trying to find good approximations for an exponential of a large matrix, which may arise from discretizing an operator.)
Paradigm Problem on Algorithm Subtleties: Simple passive advection

\[
\frac{\partial}{\partial t} f(x, t) = -\frac{\partial}{\partial x} (v f)
\]

Thousands of papers written on algorithms for this equation (and extensions: \(v\) a nonlinear function of \(f\), shocks, multiple fields \(f \rightarrow \vec{f}\), multidimensional \(\vec{x}\), generic hyperbolic equations and conservation laws, etc.). (Apps: weather and climate modelling, airplanes, architecture, astrophysics, nano-technology & MicroElectroMechanical Systems, ...) For \(v=\text{const}\), simple exact solution:

\[
f(x, t) = f_0(x - vt)
\]

Numerical solution often better in integral form, insures conservation properties:

\[
\frac{\partial}{\partial t} \int_{x_{i-1/2}}^{x_{i+1/2}} dx f = -vf_{i+1/2} + vf_{i-1/2}
\]

(illustrate with grid.) Define \(f_i\) as exact cell average, \(f_{i+1/2}\) as exact bdy value:

\[
\frac{\partial}{\partial t} \Delta x f_i = -vf_{i+1/2} + vf_{i-1/2}
\]
Centered 2cd order fluxes

\[ \frac{\partial}{\partial t} \Delta x f_i = -v f_{i+1/2} + v f_{i-1/2} \]

To evaluate fluxes through the boundary, try centered, 2cd order accurate interpolation \( f_{i+1/2} \approx (f_{i+1} + f_i)/2 \):

\[ \frac{\partial}{\partial t} f_i \approx \frac{-v f_{i+1} + v f_{i-1}}{2\Delta x} \]

Equivalent to finite-difference approx. to orig. eq.: \( \partial f/\partial t = -\partial(v f)/\partial x \). Preserves discrete analogs of conservation properties, density conservation

\[ \frac{\partial}{\partial t} \int dx f = \frac{\partial}{\partial t} \sum_i \Delta x f_i = 0 \]

and “entropy” conservation (or energy or enstrophy or vorticity conservation in various contexts involving Poisson bracket) like spectral or Arakawa algorithms:

\[ \frac{\partial}{\partial t} \int dx f^2 = \frac{\partial}{\partial t} \sum_i \Delta x f_i^2 = 0 \]
Relatively okay solution, will converge to exact solution as $\Delta x \rightarrow 0$. But disappointing it requires so many grid points, and that there are artificial oscillations and that $f<0$.  

**Simple test: Advection of Gaussian pulse in periodic box**
Harder test: Advection of Gaussian + Top Hat in periodic box

Gaussian + Step test, 1 period, CFL=0.1, 2cd order Centered

Paradigm for problems with shocks (near discontinuities) or other under-resolved features.
Upwind 1st order fluxes

\[ \frac{\partial}{\partial t} \Delta x f_i = -v f_{i+1/2} + v f_{i-1/2} \]

Physically, information should propagate only in one direction, i.e., for \( v > 0 \) the flux through the boundary is coming only from the left, so try approximating the fluxes using an “upwind flux”, \( f_{i+1/2} \approx f_i \) (and reverse for \( v < 0 \)).

\[ \frac{\partial}{\partial t} f_i \approx -v f_i + v f_{i-1} \]

Preserves that evolution of \( f_i \) depends only on \( f_j \) for \( j \leq i \).

Still have particle conservation but lose “entropy conservation”.
1st order upwind preserves positivity but very diffusive
Higher-order upwind Methods with clever monotonicity-preserving slope limiters

Reconstruct $f(x)$ in each cell, extrapolate to right boundary (for upwind flux if $v>0$):

$$f_{i+1/2} = f_i + s_i \Delta x / 2$$

Piecewise constant = 1st order upwind :

$$s_i = 0$$

Downwind slope (centered 2nd order flux):

$$s_i = s_{i+1/2} = \frac{f_{i+1} - f_i}{\Delta x}$$

Upwind slope (upwind-biased 2nd order flux):

$$s_i = s_{i-1/2} = \frac{f_i - f_{i-1}}{\Delta x}$$

Van Leer’s (MC) limiter: "Monotonized Central"

$$s_i = \text{minmod} \left( \frac{s_{i-1/2} + s_{i+1/2}}{2}, 2s_{i-1/2}, 2s_{i+1/2} \right)$$

($\text{minmod}(a,b,c) = \text{sign}(a) \times \text{min}(|a|,|b|,|c|)$ if $a$, $b$, and $c$ all have same sign, otherwise $\text{minmod}(a,b,c)=0$.)

In smooth regions, $s_{i+1/2} \approx s_{i-1/2}$, and $f_{i+1/2}$ is 2nd order accurate (with some upwind bias.) Near discontinuities or extrema, $s_{i+1/2}$ or $s_{i-1/2}$ is much smaller than other, $f_{i+1/2}$ switches to 1st order upwind with $s_i \approx 0$.

Godunov’s theorem: to guarantee avoiding artificial oscillations, a linear algorithm can only be 1st order (and very diffusive). This algorithm is nonlinear.
3rd order SSP-RK used here. Looks better at CFL=0.5 with 2nd order single-step time-space-coupled time advancement, (becomes exact at CFL=1), but for complex flows there will be regions at many different values of CFL=v*dt/dx, incl. CFL<<1.
Central differencing around boundary, $s_i = s_{i+1/2} = (f_{i+1} - f_i)/\Delta x$, corresponds to a reconstructed $f(x)$ that has overshoots. Even central differencing around a cell, $s_i = (f_{i+1} - f_{i-1})/(2 \Delta x)$ would give overshoots.

MC limiter never allows reconstructed $f(x)$ to overshoot average $f$ in adjacent cell. Much more robust.

Discontinuous Galerkin algorithms can generalize this approach.

Useful way to think about these types of algorithms: “REA”: Reconstruct, Evolve, Average. (See Leveque). Starting from cell averages at last time step, reconstruct continuum solution, evolve in time (simple shifting of solution), and average back to cell averages. As described in Leveque, if the reconstruction step doesn’t introduce artificial overshoots, than the later steps won’t either.

Just going to higher order doesn’t help near sharp gradient regions (Gibb’s phenomena), or near boundaries or outside radius of convergence (Runge’s phenomena).

Basis problem: calculus concepts and theorems about convergence apply to sufficiently smooth, well-resolved, functions, and thus break near discontinuities.
New Limiters even preserve accuracy at smooth Extrema

Gaussian + Step test, 1 period, CFL=0.1, Suresh-Huynh SuHu5

(Suresh-Huynh 1997, Colella-Sekora 2008 comparable)
(My incomplete understanding of) Historical Development of Shock-Capturing Fluid Algorithms

- Initial ideas from physicists (Boris, van Leer) & (applied) mathematicians: Phil Collela, Ami Harten, Stan Osher, Chi-Wang Shu, Bjorn Enquist, Eitan Tadmor, ...
- earliest numerical viscosity, simple upwind: von Neumann & Richtmeyer ('50), Courant, Isaacson, & Rees ('52), Rosenbluth.
- Godunov ('59): generalized upwind to multiple eqs. w/ shocks (Riemann solver), theorem: only 1st order near discontinuities; piecewise-constant reconstructions
- Two indep. breakthroughs (FCT, van Leer): nonlinear switches enhance diffusion only near discontinuities or under-resolved features
- FCT (Flux-Corrected Transport) (71-79), Boris, Book. Zalesak version (79)
- van Leer (72-79), MUSCL (Monotone Upstream-Centered Schemes for Conservation laws) piecewise linear interpolation with slope limiters to avoid overshots (2cd order in smooth regions, but const. near extrema, “clipping”)
- TVD (Total Variation Diminishing) (variations of 2cd order van Leer)
- ENO/WENO (Weighted Essentially Non-Oscillatory, '87/'94-'96) Elegant solution to long-standing Gibbs oscillation problem (related to fitting with a Sobolev norm?) but somewhat slow. Arbitrary order (3rd, 5th typical) [Local operations, parallelizes easier than splines, ...]
- Suresh-Huynh '97, Colella-Sekora '08, new limiters avoid clipping of smooth extrema
- DG (Discontinuous Galerkin, extends to higher-order information within each cell beyond cell average. Use Gaussian integration to achieve order 2p-1 with p points/cell. Good aspects of finite-volume + finite-element, can borrow limiter ideas. (Growing development & use 2000-)}
Main idea behind these algorithms: detect discontinuities / under-resolved features, revert to lower-order polynomial in non-smooth regions, allow discontinuities (allowed for by hyperbolic eqs.), introduce minimum necessary numerical diffusion in non-smooth regions to preserve (or encourage) monotonicity, positivity.

- Suresh-Huynh ('97): relaxed previous limiters to allow higher order interpolations near smooth extrema, 5th order in smooth regions, essentially a more efficient way to implement WENO.
- Colella-Sekora ('08): alternate way to relax piecewise-constant assumption at extrema, 4th order in smooth regions (even order = no numerical diffusion in smooth regions)

**FIG. 2.1.** An extrema and a discontinuity look the same over a stencil of three points. The values $v_{j-1}$, $v_j$ and $v_{j+1}$ in (a) are identical to the corresponding ones in (b).
Pitfalls of Naive Numerical Algorithms:
A Simple Diffusion Eq. Example

\[ \frac{\partial}{\partial t} T(x, t) = \frac{\partial^2 T}{\partial x^2} + \sin(x) \]
Discretizing a Diffusion Eq.

\[ \frac{\partial}{\partial t} T(x, t) = \frac{\partial^2 T}{\partial x^2} + \sin(x) \]

Discretize \( T(x) \) onto a grid:

\[ T(x_j, t) = T_j(t) \]

Discrete analogs of 1st & 2nd derivatives:

\[ \left( \frac{\partial T}{\partial x} \right)_{j+1/2} = \frac{T_{j+1} - T_j}{\Delta x} \]

\[ \left( \frac{\partial^2 T}{\partial x^2} \right)_{j} = \left( \frac{\partial}{\partial x} \frac{\partial T}{\partial x} \right) = \frac{(T_{j+1} - T_j) - (T_j - T_{j-1})}{(\Delta x)^2} \]

\[ x_j = j \Delta x \]
Simple Discretization of a Diffusion Eq.

\[
\frac{\partial}{\partial t} T(x, t) = \frac{\partial^2 T}{\partial x^2} + \sin(x)
\]

\[
\frac{T_j(t + \Delta t) - T_j(t)}{\Delta t} = \frac{(T_{j+1} - T_j) - (T_j - T_{j-1})}{(\Delta x)^2} + \sin(x_j)
\]

Given \(T_j(t)\), loop over all positions \(j\) to get \(T_j(t+dt)\). Repeat to find \(T(t+2dt)\) from \(T(t+dt)\)…
Test simple diffusion algorithm on a coarse mesh,
10 points for $x=0$ to $x=\pi$,
d$t=0.01$

matches exact solution fairly well.
Everything “seems fine”, just want to use finer spatial mesh…
Test simple diffusion algorithm on a finer mesh,
increase from 10 to 100 points for x=0 to x=\pi,
dt=0.01 (unchanged)

Within 6 time steps the solution becomes garbage

Maximum temperature grows very quickly due to numerical instability, exceeds biggest number representable on the computer in just a few dozen iterations, T = “NaN”.
The root of the problem

\[ \frac{\partial}{\partial t} T(x, t) = D \frac{\partial^2 T}{\partial x^2} + \sin(x) \]

Fourier Transform, look at \( k \neq 1 \) modes

\[ \frac{\partial}{\partial t} T_k = -Dk^2 T_k \]

Explicit integration (1st order "Euler"):

\[ \frac{T_k(t + \Delta t) - T_k(t)}{\Delta t} = -Dk^2 T_k(t) \]

\[ T_k(t + \Delta t) = \left(1 - \Delta t \ Dk^2 \right) T_k(t) \]

1st order Taylor series approx. to exact result

\[ e^{-\Delta t \ Dk^2} \]

Stability limit: \( \Delta t \ D k^2 < 2 \) for all \( k \) modes in simulation
Fix with a more robust Implicit algorithm

\[ \frac{\partial}{\partial t} T_k = -Dk^2 T_k \]

Explicit integration (1st order “Euler”):

\[ \frac{T_k(t + \Delta t) - T_k(t)}{\Delta t} = -Dk^2 T_k(t) \]

Implicit integration (1st order “Backwards Euler”):

\[ \frac{T_k(t + \Delta t) - T_k(t)}{\Delta t} = -Dk^2 T_k(t + \Delta t) \]

\[ T_k(t + \Delta t) = \frac{1}{\left(1 + \Delta t \cdot Dk^2\right)} T_k(t) \]

1st order Padé approx. to exact result \( e^{-\Delta t Dk^2} \)

Implicit algorithm is accurate for modes with small \( \Delta t D k^2 \), while robustly stable for all \( \Delta t D k^2 \), giving qualitatively correct damping for all modes.

(higher order implicit algorithms exist: Crank-Nicolson, Backward Differentiation Formulas)
Geometrical Interpretation of Implicit Algorithm as Integrating Backwards in Time

\[
\frac{\partial y}{\partial t} = F(y) = -y
\]

Explicit integration (1st order “Euler”):

\[
y(\Delta t) = y(0) + \Delta t \cdot F(y(0))
\]

Implicit integration (1st order “Backwards Euler”):

\[
y(\Delta t) = y(0) + \Delta t \cdot F(y(\Delta t))
\]

Rearrange as:

\[
y(0) = y(\Delta t) - \Delta t \cdot F(y(\Delta t))
\]

Thus implicit algorithm = integrating backwards in time, to find what \(y(\Delta t)\) at the future time is needed to give \(y(0)\) at the current time.
(Requires inverting operator \(1 - \Delta t F\) …)
Implicit algorithms more complex, require inversions

\[
\frac{\partial}{\partial t} T(x, t) = \frac{\partial^2 T}{\partial x^2} + \sin(x)
\]

\[
\frac{T_j(t + \Delta t) - T_j(t)}{\Delta t} = \frac{T_{j+1}(t + \Delta t) - 2T_j(t + \Delta t) + T_{j-1}(t + \Delta t)}{(\Delta x)^2} + \sin(x_j)
\]

Can rearrange this in the form:

\[
M_{ij} T_j(t + \Delta t) = T_i(t) + \Delta t \sin(x_i)
\]

Requires inverting the matrix M to find the vector T at the future time

Sometimes this can be hard: General NxN matrix requires \(O(N^3)\) operations to invert. In this case, M is “tridiagonal” and solution can be found quickly in \(O(N)\) operations. For many PDE’s M is “sparse”, and fast solution methods exist… In general:

\[
\frac{\partial Y}{\partial t} = F(Y)
\]

Things can get very interesting, if \(F\) is a nonlinear integro-differential operator that has to be inverted. Recent Newton-Krylov algorithms?
The power of optimal algorithms

- Advances in algorithmic efficiency can rival advances in hardware architecture
- Consider Poisson’s equation on a cube of size $N=n^3$

<table>
<thead>
<tr>
<th>Year</th>
<th>Method</th>
<th>Reference</th>
<th>Storage</th>
<th>Flops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1947</td>
<td>GE (banded)</td>
<td>Von Neumann &amp; Goldstine</td>
<td>$n^5$</td>
<td>$n^7$</td>
</tr>
<tr>
<td>1950</td>
<td>Optimal SOR</td>
<td>Young</td>
<td>$n^3$</td>
<td>$n^4 \log n$</td>
</tr>
<tr>
<td>1971</td>
<td>CG</td>
<td>Reid Conjugate Gradients w/</td>
<td>$n^3$</td>
<td>$n^{3.5} \log n$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gustaffson’s modified ILU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1984</td>
<td>Full MG</td>
<td>Brandt Multigrid</td>
<td>$n^3$</td>
<td>$n^3$</td>
</tr>
</tbody>
</table>

- If $n=64$, this implies an overall reduction in flops of ~16 million*

*Six-months is reduced to 1 s
Algorithms and Moore’s Law

- This advance took place over a span of about 36 years, or 24 doubling times for Moore’s Law.
- $2^{24} \approx 16$ million ⇒ the same as the factor from algorithms alone!
IBM’s BlueGene/L: 65536 dual procs, 180 Tflop/s

System (64 cabinets, 64x32x32)

Cabinet (32 Node boards, 8x8x16)

Node Board (32 chips, 4x4x2) 16 Compute Cards

Compute Card (2 chips, 2x1x1)

Chip (2 processors)

2.8/5.6 GF/s 4 MB

5.6/11.2 GF/s 0.5 GB DDR

90/180 GF/s 8 GB DDR

2.9/5.7 TF/s 256 GB DDR

Present offer from IBM

Single cabinet 5.7 TFlop/s peak

$2M in acad. consortium

David Keyes, Columbia Univ.
Whimsical remarks on simulation progress, 1988-2005

• If similar improvements in speed ($10^5$) had been realized in the airline industry, a 3-hour flight would require one-tenth of a second today

• If similar improvements in storage ($10^4$) had been realized in the publishing industry, our office bookcases could hold the book portion of the collection of the Library of Congress (15M volumes)

• If similar reductions in cost ($10^4$) had been realized the higher education, tuition room and board would cost about $2 per year
For magnetic confinement, there are 4 classes of major simulation codes, each addressing different phenomena. Many examples of interesting computational plasma physics work could be shown...
Fusion plasmas exhibit enormous ranges of temporal and spatial scales.

- Nonlinear MHD-like behavior couples many of the time- & length-scales.

Even with the most powerful computers expected in the next 20 years, there are many problems with such an extreme range of scales that they can’t be directly solved…

Center for the Study of Plasma Microturbulence

- A DOE, Office of Fusion Energy Sciences, SciDAC (Scientific Discovery Through Advanced Computing) Project

- devoted to studying plasma microturbulence through direct numerical simulation

- National Team (& 2 main codes):
  - GA (Waltz, Candy)
  - U. MD (Dorland)
  - MIT (D. Ernst)
  - LLNL (Nevins, Cohen, Dimits)
  - PPPL (Hammett, …)

- They’ve done all the hard work …
Fairly Comprehensive 5-D Gyrokinetic Turbulence Codes Have Been Developed

• Solve for the particle distribution function \( f(r, \theta, \alpha, E, \mu, t) \) (avg. over gyration: 6D \( \rightarrow \) 5D)
• 500 radii x 32 complex toroidal modes (96 binormal grid points) x 10 parallel points along half-orbits x 8 energies x 16 \( v_{||}/v \)
  12 hours on ORNL Cray X1E with 256 MSPs
• Realistic toroidal geometry, kinetic ions & electrons, finite-\( \beta \) electro-magnetic fluctuations, collisions. Sophisticated algorithms.
• 3 most widely used comprehensive codes all use “continuum”/Eulerian algorithms:
  GS2 (Dorland et al.)
  GYRO (Candy et al.)
  GENE (Jenko et al.)
The gyrokinetic code **GENE**

Gene is a physically comprehensive Vlasov code:
- allows for kinetic electrons & electromagnetic fluctuations, collisions, and external ExB shear flows
- is coupled to various MHD codes and the transport code TRINITY
- can be used as **initial value** or **eigenvalue** solver
- supports **local** (flux-tube) and **global** (full-torus), gradient- and flux-driven simulations

**GENE** is well benchmarked and hyperscalable

Goerler, Jenko, et al.
Major Theoretical & Algorithmic Speedups

relative to simplest brute force PIC algorithm, fully resolved ($\Delta t \sim 1/\omega_{pe}$, $\Delta x \sim \lambda_{De}$), for ITER $1/\rho_* = a/\rho_0 \sim 700$

- Nonlinear gyrokinetic equation
  - ion polarization shielding eliminates plasma freq. $\omega_{pe}/\Omega_{ci} \sim m_i/m_e$ $x10^3$
  - ion polarization eliminates $\rho_e$ & Debye scales $(\rho_i/\lambda_{De})^3$ $x10^5$
  - average over fast ion gyration (& field-aligned), $\Omega_{ci}/\omega_* \sim 1/\rho_*$. $x10^3$
- Continuum or $\delta f$ PIC, reduces noise, $(f_0/\delta f)^2 \sim 1/\rho_*^2$ $x10^6$
- Field-aligned coordinates (nonlinear extension of ballooning coord.)
  \[ \Delta_\parallel / (\Delta_\perp q R / a) \sim a / (q R \rho_*) \]  $x70$
- Flux-tube / Toroidal annulus wedge, $\downarrow$ simulation volume
  - $k_q \rho_i = 0, 0.05, 0.1, \ldots, 1.0$
    $n = 0, 15, 30, \ldots, 300$ (i.e., 1/15 of toroidal direction) $x15$
  - $L_r \sim a/5 \sim 140 \rho \sim 10$ correlation lengths $x5$
- High-order / spectral algorithms in 5-D, $2^5 \times 2$ $x64$
- Implicit electrons $x5-50$
- Total combined speedup of all algorithms $x10^{23}$
- Massively parallel computers (Moore’s law 1982-2007) $x10^5$
Edge region very difficult

Major extensions to gyrokinetic codes needed to handle additional complications of edge region of tokamaks (& stellarators):

open & closed field lines, steep gradients near beta limit, electric & magnetic fluctuations, strong shear-flow layers, steep-gradients and large amplitude fluctuations, positivity constraints, wide range of collisionality, non-axisymmetric RMP coils, plasma-wall interactions, strong sources and sinks in atomic physics.

Edge pedestal temperature profile near the edge of an H-mode discharge in the DIII-D tokamak. [Porter2000]. Pedestal is shaded region.
Stable Pendulum

\[ F = Mg \]
\[ \omega = \left(\frac{g}{L}\right)^{1/2} \]

Unstable Inverted Pendulum

\[ \omega = \left(-\frac{g}{|L|}\right)^{1/2} = i \left(\frac{g}{|L|}\right)^{1/2} = i\gamma \]

(rigid rod)

Density-stratified Fluid

\[ \rho = \exp\left(-\frac{y}{L}\right) \]

stable \( \omega = \left(\frac{g}{L}\right)^{1/2} \)

Inverted-density fluid

\[ \Rightarrow \text{Rayleigh-Taylor Instability} \]

\[ \rho = \exp\left(\frac{y}{L}\right) \]

Max growth rate \( \gamma = \left(\frac{g}{L}\right)^{1/2} \)
“Bad Curvature” instability in plasmas
≈ Inverted Pendulum / Rayleigh-Taylor Instability

Top view of toroidal plasma:

Growth rate:
\[ \gamma = \sqrt{\frac{g_{\text{eff}}}{L}} = \frac{V_t}{\sqrt{RL}} = \frac{V_t}{\sqrt{RL}} \]

Similar instability mechanism in MHD & drift/microinstabilities

1/L = \( \frac{\nabla p}{p} \) in MHD,
\( \propto \) combination of \( \nabla n & \nabla T \) in microinstabilities.

plasma = heavy fluid

B = “light fluid”

\[ g_{\text{eff}} = \frac{v^2}{R} \] centrifugal force
The Secret for Stabilizing Bad-Curvature Instabilities

Twist in $B$ carries plasma from bad curvature region to good curvature region:

Unstable

Stable

Similar to how twirling a honey dipper can prevent honey from dripping.
Spherical Torus has improved confinement and pressure limits (but less room in center for coils)
Understanding Turbulence That Affects the Performance of Fusion Device

Central temp $\sim 10$ keV $\sim 10^8$ K

Large temperature gradient $\rightarrow$ turbulent eddies $\rightarrow$ cools plasmas $\rightarrow$ determines plasma size needed for fusion ignition

Major progress in last decade: detailed nonlinear simulations (first 3-D fluid approximations, now 5-D $f(x,v_\parallel,v_\perp,t)$) & detailed understanding

(Candy & Waltz, GA 2003)
Continuum/Eulerian Approach to Electromagnetic Gyrokinetic Turbulence

GS2 (Dorland & Kotschenreuther) http://gs2.sourceforge.net
GENE (Jenko) http://www.ipp.mpg.de/~fsj/
GYRO (Candy & Waltz) http://fusion.gat.com/comp/parallel/

These codes widely used by many to study plasma turbulence in fusion devices. GYRO & GENE currently the most comprehensive gyrokinetic codes available:

- Gyrokinetic ions (multiple species) & adiabatic/drift-kinetic/gyrokinetic electrons
- Trapped and passing electrons (and ions) for Trapped Electron Mode
- Pitch-angle scattering collision operator (TEM / neoclassical effects)
- Finite beta magnetic fluctuations as well as electrostatic fluctuations (important for kinetic-ballooning modes, magnetic flutter contribution to transport)
- General shaped tokamak geometry
- Equilibrium ExB and parallel velocity shear
- Finite-$\rho_*$ effects (profile shear stabilization, nonlocal transport)…

Nevertheless, a lot of interesting work remains to be done: more tests against experiments, particle transport, transport barrier formation, shaping effects, understand scalings, couple to transport codes for complete predictive ability, &:

edge simulations (new codes needed to do gyrokinetics in the edge, challenging…)

Complex 5-dimensional Computer Simulations being developed

- Solving gyro-averaged kinetic equation to find time-evolution of particle distribution function
  \[ f(\vec{x}, E, v_{||}/v, t) \]
- Gyro-averaged Maxwell’s Eqs. determine Electric and Magnetic fields
- “typical” grid 96x32x32 spatial, 10x20 velocity, x 3 species for \(10^4\) time steps.
Gyrokinetic Eq. Summary

Gyro-averaged, non-adiabatic part of 5-D particle distribution function: \( h_s = h_s(\mathbf{x}, \varepsilon, \mu, t) \) determined by gyrokinetic Eq. (in deceptively compact form):

\[
\frac{\partial h}{\partial t} + \left( v_{\parallel} \hat{b} + \mathbf{v}_d \right) \cdot \nabla h + \hat{b} \times \nabla \chi \cdot \nabla (h + F_0) + q \frac{\partial F_0}{\partial \varepsilon} \frac{\partial \chi}{\partial t} = C[h]
\]

Generalized Nonlinear ExB Drift
Incl. Magnetic fluctuations

\( \chi(\mathbf{x}, t) \) is gyro-averaged, generalized potential. Electric and magnetic fields from gyro-averaged Maxwell’s Eqs.

\[
\chi = J_0 \left( \frac{k_{\perp} v_{\perp}}{\Omega_c} \right) \left( \phi - \frac{v_{\parallel}}{c} A_{\parallel} \right) + \frac{J_1(k_{\perp} v_{\perp}/\Omega_c)}{k_{\perp} v_{\perp}/\Omega_c} \frac{m v_{\perp}^2 \delta B_{\parallel}}{q \frac{k_{\perp} v_{\perp}/\Omega_c}} B
\]
Bessel Functions represent averaging around particle gyro-orbit

Gyroaveraging eliminates fast time scales of particle gyration (10 MHz - 10 GHz)

Easy to evaluate in pseudo-spectral codes. Fast multipoint Padé approx. in other codes.

\[ \chi = J_0(k_\perp \rho) \Phi \]

\[ \chi(\vec{x}) = \int d\theta \Phi(\vec{x} + \vec{n}(\theta)) \]
Examples of Nonlinear Macroscopic Simulation

1) MHD evolution of the tokamak internal kink mode (m=1, n=1)
   - Plasma core is exchanged with cooler surrounding plasma.

M3D simulation of NSTX [W. Park]

Evolution of pressure and magnetic topology from a NIMROD simulation of DIII-D
ITER mesh

Circle $f = 0.000$

Zoom $f = 0.000$
ITER ELM: pressure time evolution
ELM pressure: initial, mode growth, outflow
Calculations on the Cray XT-3 have allowed the first simulations of mode conversion in ITER

ITER with D:T:HE3 = 20:20:30 with $N_R = N_Z = 350$, $f = 53$ MHz, $n = 2.5 \times 10^{19}$ m$^{-3}$

(4096 processors for 1.5 hours on the Cray XT-3)

Future Work – Will extend this MC scenario to more ITER relevant densities ($\approx 7 \times 10^{19}$ m$^{-3}$)
Computational Plasma Physics: Powerful New Tools of Scientific Discovery

- Exponential growth of computer power means that a lot of important and interesting problems are becoming tractable by computer solutions. Will continue to be a growth field.

- The importance of good numerical algorithms
  Pitfall of naive algorithm for paradigm advection or diffusion equations

- Examples of cutting edge computational plasma physics, such as:
  - Simulating 5-dimensional plasma turbulence in fusion devices
  - MHD simulations of Edge Localized Modes (ELMs).
  - RF heating

- Computer simulations can be fun!
General Computational References

- S. Jardin, **Computational Methods in Plasma Physics** (used in the graduate computational plasma physics course he teaches at Princeton)
- Richard Fitzpatrick’s computational physics & other online physics lecture notes http://farside.ph.utexas.edu/teaching.html
- Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics
- LeVeque, Finite Volume Methods for Hyperbolic Problems
- Gershenfeld, The Nature of Mathematical Modeling

Useful web sites / high quality numerical software:
- www.netlib.org  Vast repository of high quality (& free) numerical software
- PETSC (library of optimized parallelized algorithms for scientific computing, PDEs and Linear solvers)
- FFTW (Fastest FFT in the West)
- www.scidac.org DOE Scientific Discovery Through Advanced Computing Initiative
Advection Algorithm References

• D. R. Durrant, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics
• "Introduction to "Flux-Corrected Transport: I. SHASTA, A Fluid Algorithm That Works", S. T. Zalesak 1997, JCP 135, 170. Nice 2-page review of historical place of FCT algorithm, and an introduction to the original FCT article, reprinted in this special issue of JCP celebrating its 30th anniversary.
Selected Further References

- Extensive fusion library, fusion history, reactor design studies, etc: http://fire.pppl.gov
- This talk: http://w3.pppl.gov/~hammet/talks
- Center for the Study of Plasma Microturbulence https://web.gat.com/theory/Cspm
- GYRO code and movies http://fusion.gat.com/comp/parallel/gyro.html
- GENE gyrokinetic turbulence code http://gene.rzg.mpg.de
- GS2 gyrokinetic code http://gs2.sourceforge.net
- Center for Multiscale Plasma Dynamics http://cmpd.umd.edu/
- My gyrofluid & gyrokinetic plasma turbulence references: http://w3.pppl.gov/~hammet/papers/