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:

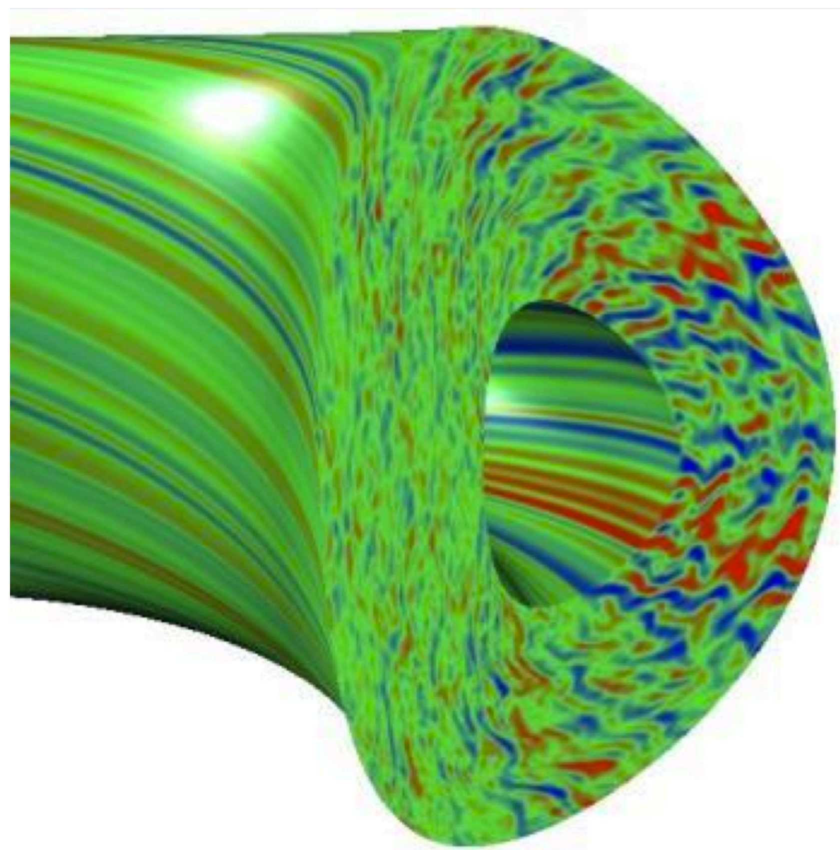
Center for the Study of Plasma Microturbulence (General Atomics, PPPL, LLNL, U. Maryland, MIT)

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.



Candy, Waltz (General Atomics)

(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

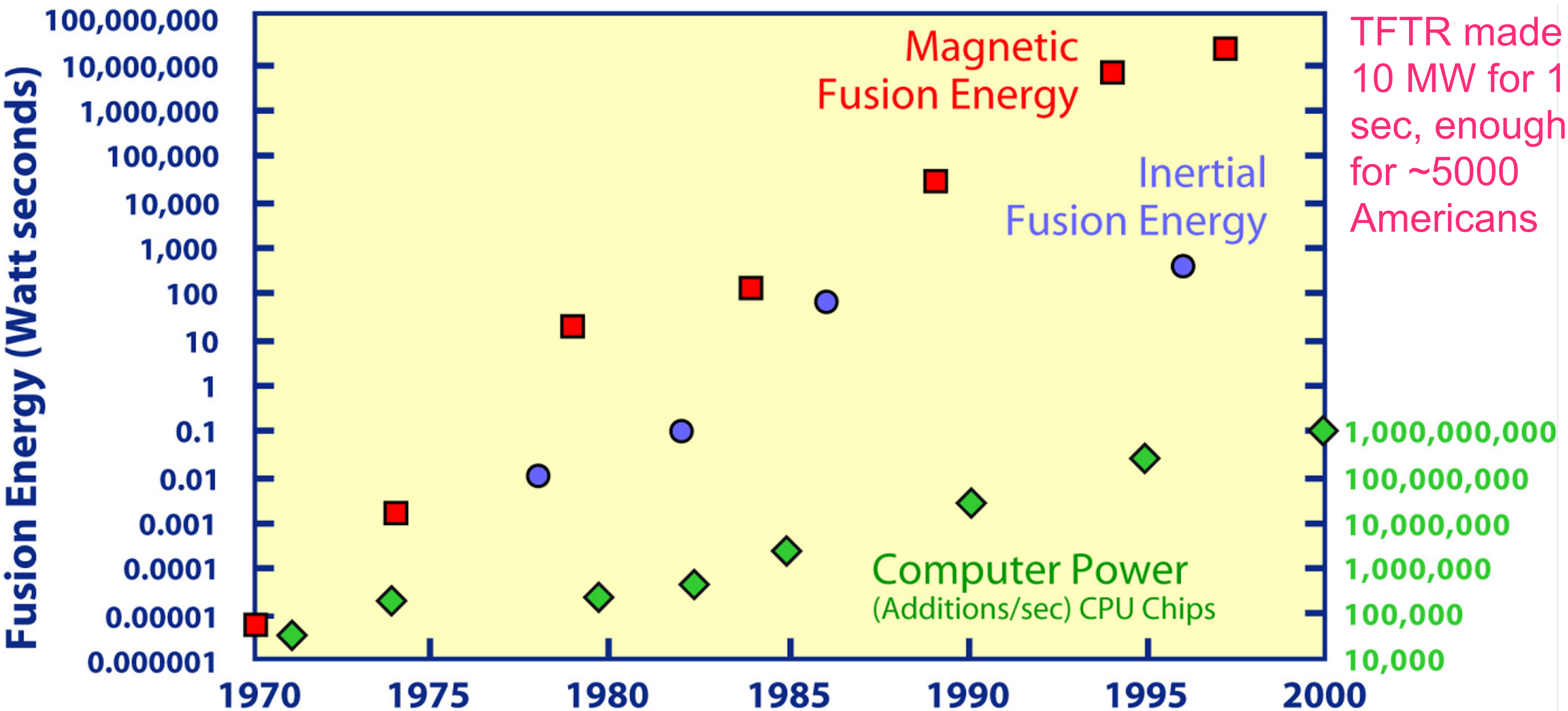
- Computational methods providing powerful tools for searching for ways to improve fusion reactor designs and our confidence in them.

But first another topic:

Brief Review:

Progress is being made in Fusion Energy research, well worth continuing.

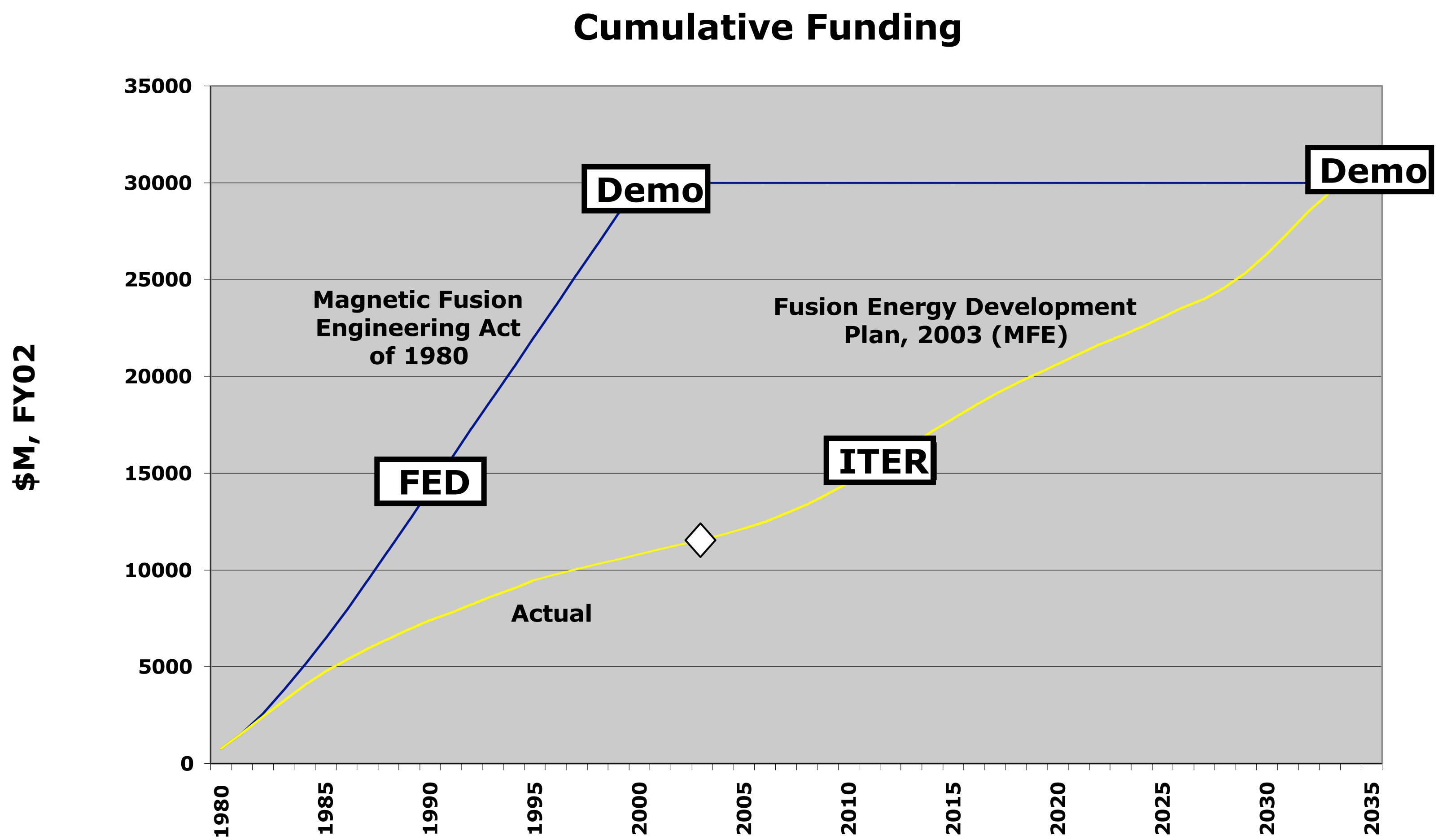
Progress in Fusion Energy has Outpaced Computer Speed



Princeton's TFTR made 10 MW for 1 sec, enough for ~5000 Americans

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

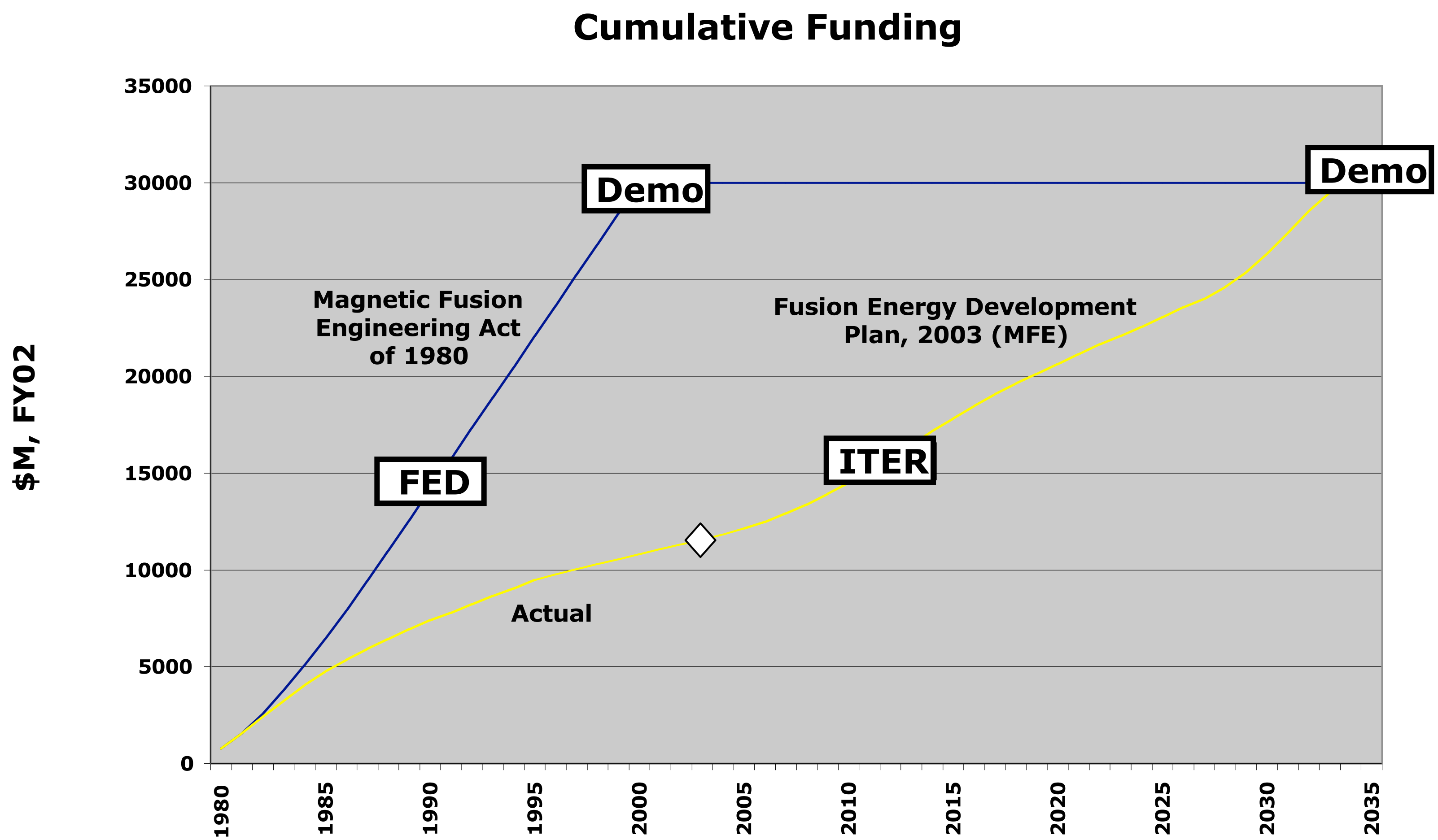
Fusion can't be criticized for being behind schedule, because we never got the budget needed



(This 2003 figure needs updating, but main point is still true.)

~$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.

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

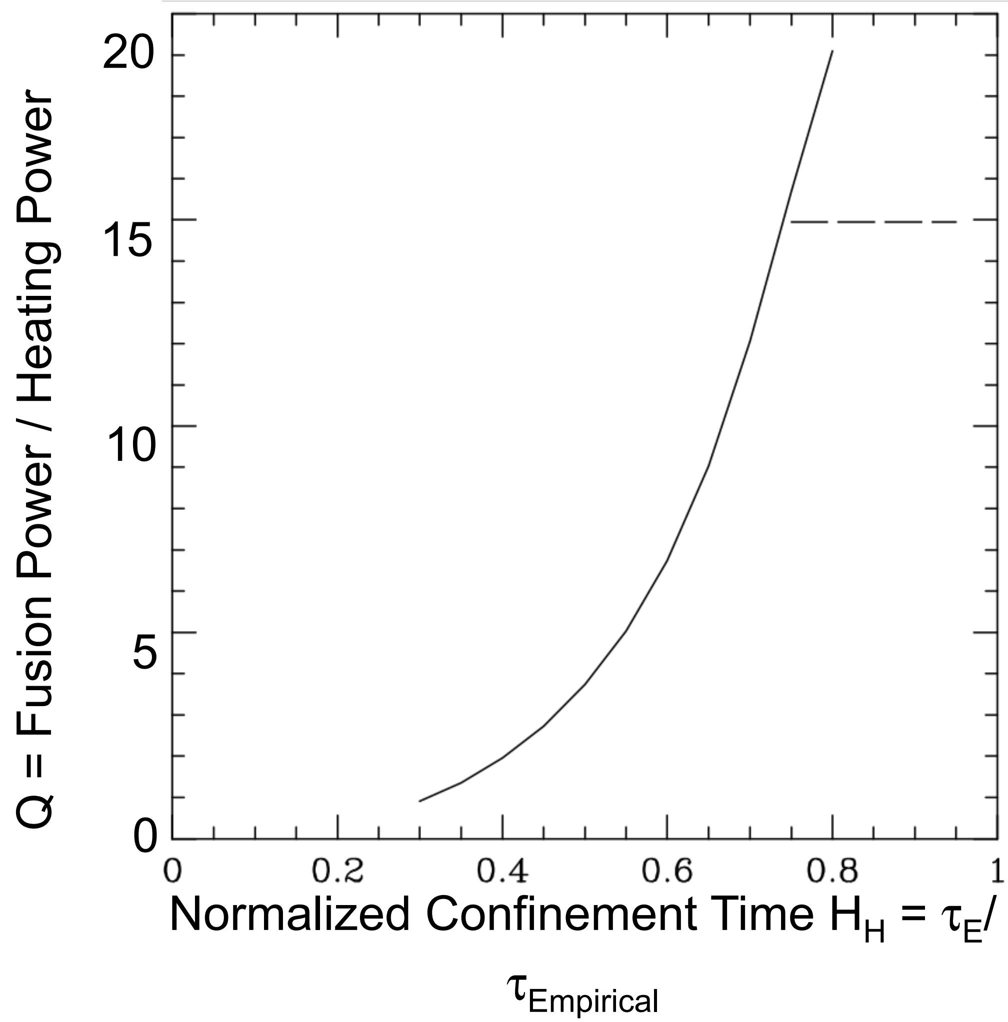
* Advanced tokamaks, studies of methods to controls Edge Localized Modes, alternative regimes (Hybrid scenarios with flattish q profiles) to improve performance

* 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.

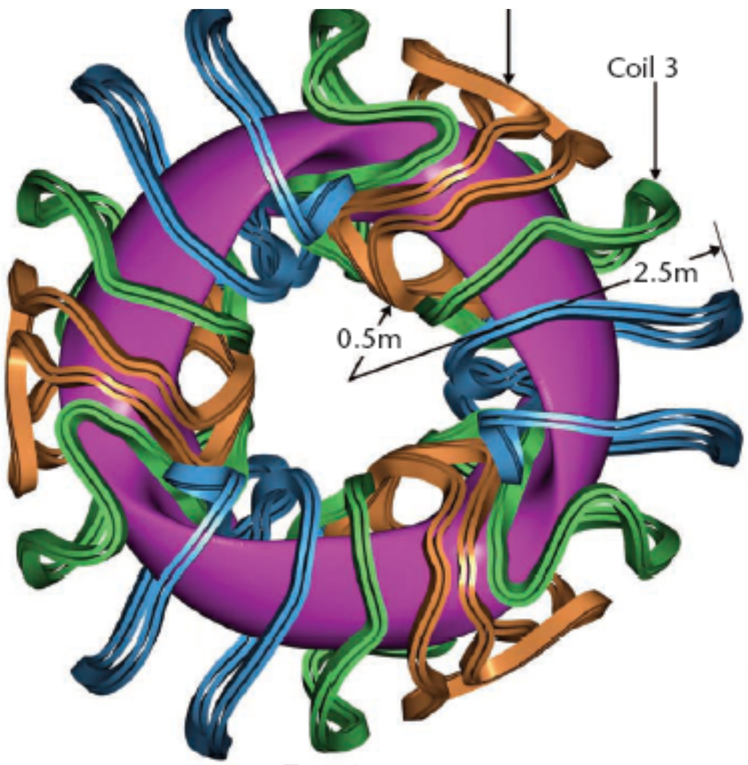


$$\frac{dW}{dt} = P_{ext} + P_{fusion} - \frac{W}{\tau_E}$$

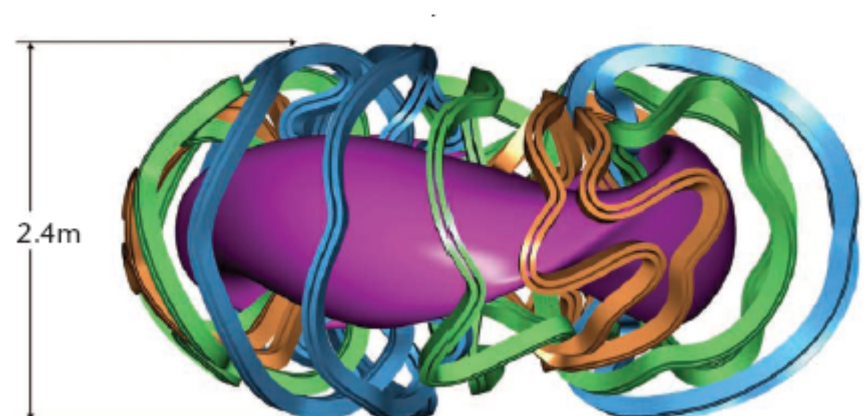
Caveats: best if MHD pressure limits also improve with improved confinement.
Other limits also: power load on divertor & wall, …

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.

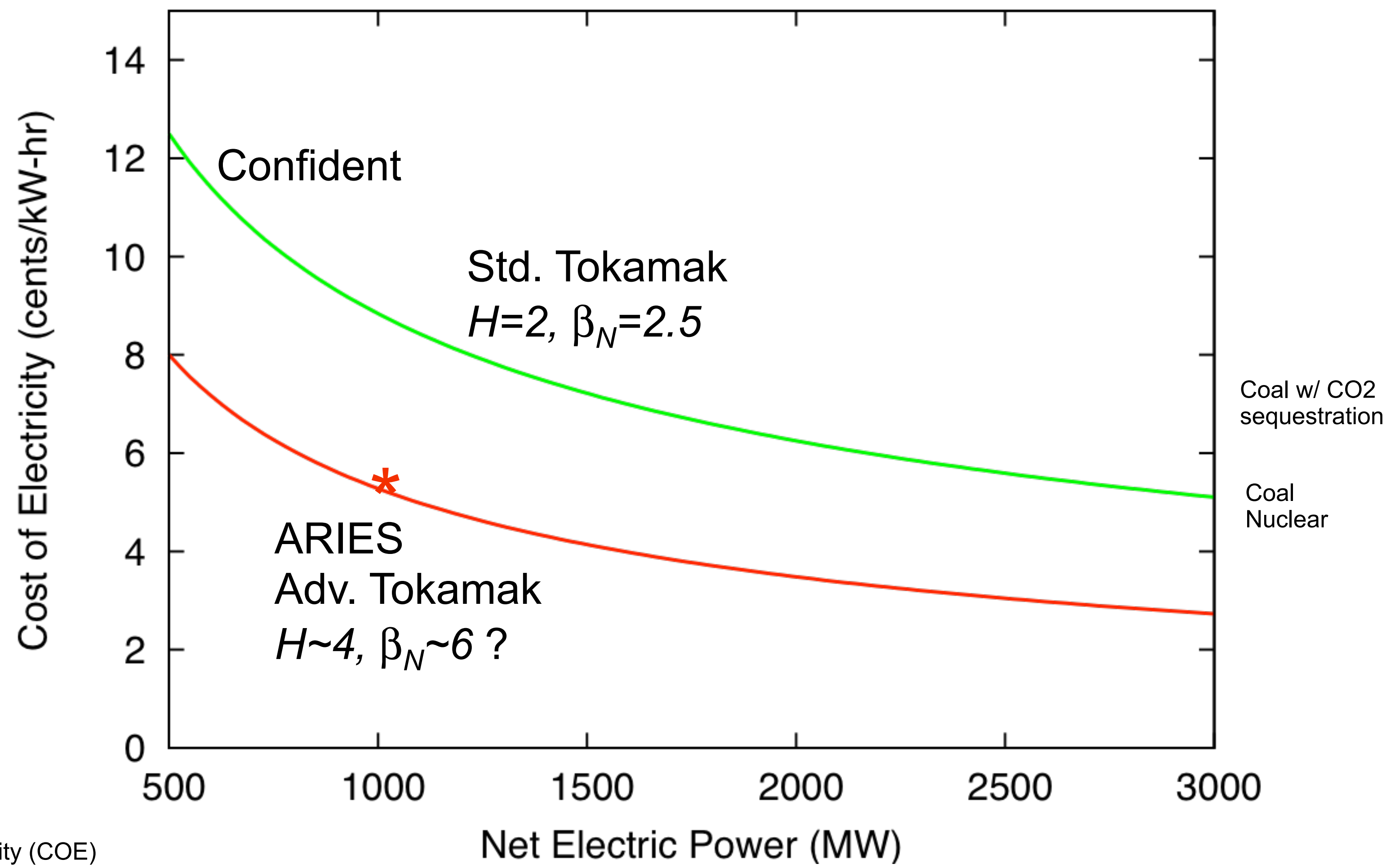


Top view



Front view

↓ turbulence (1/H) & ↑ MHD stability limits (β) could significantly improve fusion



(Relative Cost of Electricity (COE) estimates in Galambos et al. study, see ARIES reactor studies for more detailed costs estimates.)

From Galambos, Perkins, Haney, & Mandrekas 1995 Nucl.Fus. (very good), scaled to match ARIES-AT reactor design study (2001), http://aries.ucsd.edu/ARIES/

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

2008

Lukin, Vyacheslav (NRL)	Computational Study of the Internal Kink Mode Evolution and Associated Magnetic Reconnection Phenomena (#2)
Ferraro, Nathaniel M. (General Atomics)	Non-ideal effects on the stability and Transport of magnetized plasmas (#2)

2010

Smith, Sterling (General Atomics)	Magnetohydrodynamic Stability Spectrum With Flow and a Resistive Wall (#1)

2011

Raburn, Daniel (Japan)	Efficient Numerical Calculation of MHD Equilibria with Magnetic Islands, with Particular Application to Saturated Neoclassical Tearing Modes (#2)
Peterson, Jayson D.L. (LLNL, NIF)	Relating gyrokinetic electron turbulence to plasma confinement in the NSTX (#3)

2012

Baumgaertel, Jessica (LANL, NIF)	Gyrokinetic studies of turbulence in stellarators (#3)

From Wikipedia: Fastest supercomputer speeds

As computers get faster, the types of
problems we can address changes:
more time + space scales in a single simulation

exaflop →

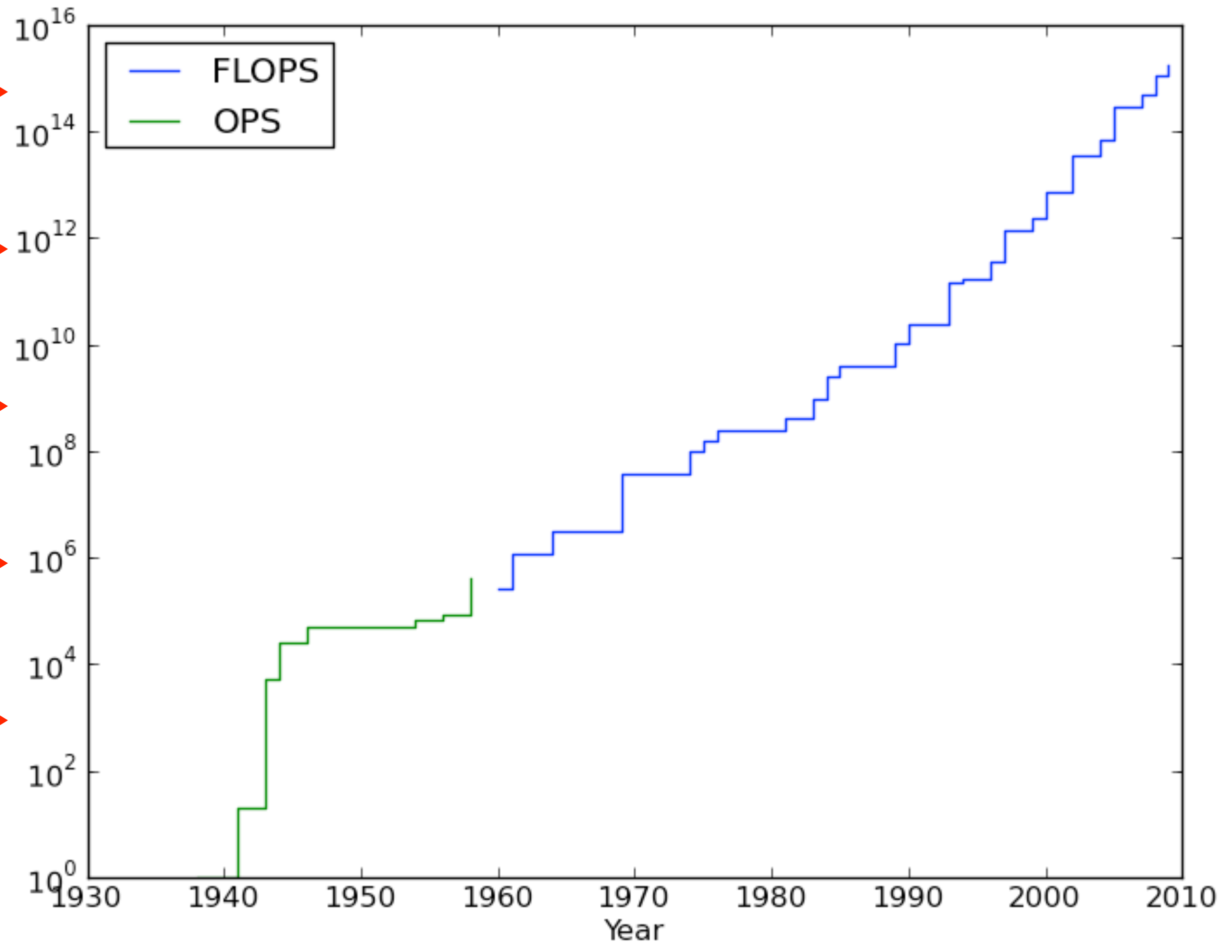
petaflop →

teraflop →

gigaflop →

megaflop →

kiloflop →



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

$$\frac{\partial}{\partial t} f(x,t) = -\frac{\partial}{\partial x}(vf)$$

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=const, simple exact solution:

$$f(x,t) = f_0(x - vt)$$

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

$$\frac{\partial}{\partial t} \int_{x_{i-1/2}}^{x_{i+1/2}} dx f = -v f_{i+1/2} + v f_{i-1/2}$$

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

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

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

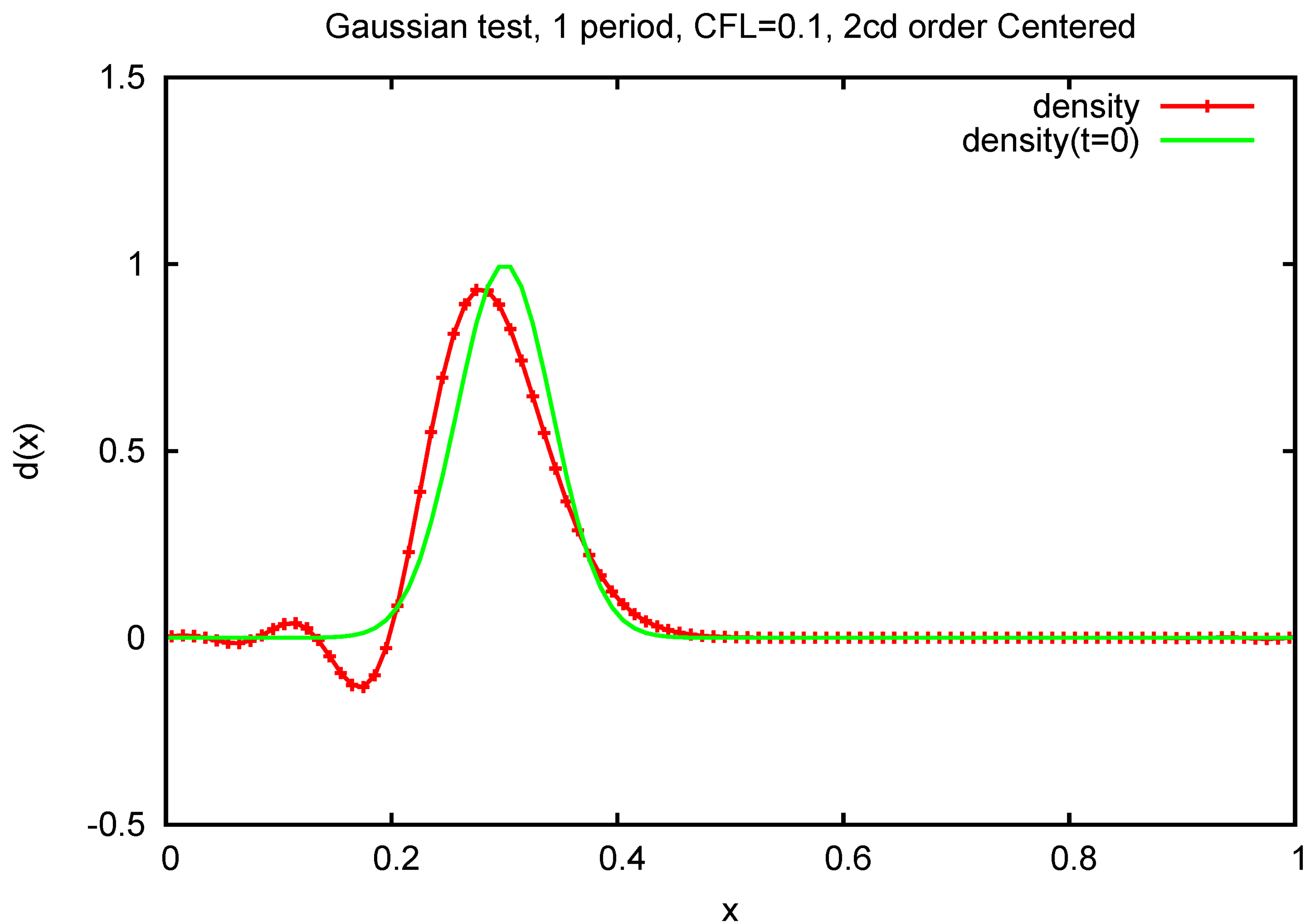
$$\frac{\partial}{\partial t}\int dx f = \frac{\partial}{\partial t}\sum_i \Delta x f_i = 0$$

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

$$\frac{\partial}{\partial t}\int dx f^2 = \frac{\partial}{\partial t}\sum_i \Delta x f_i^2 = 0$$

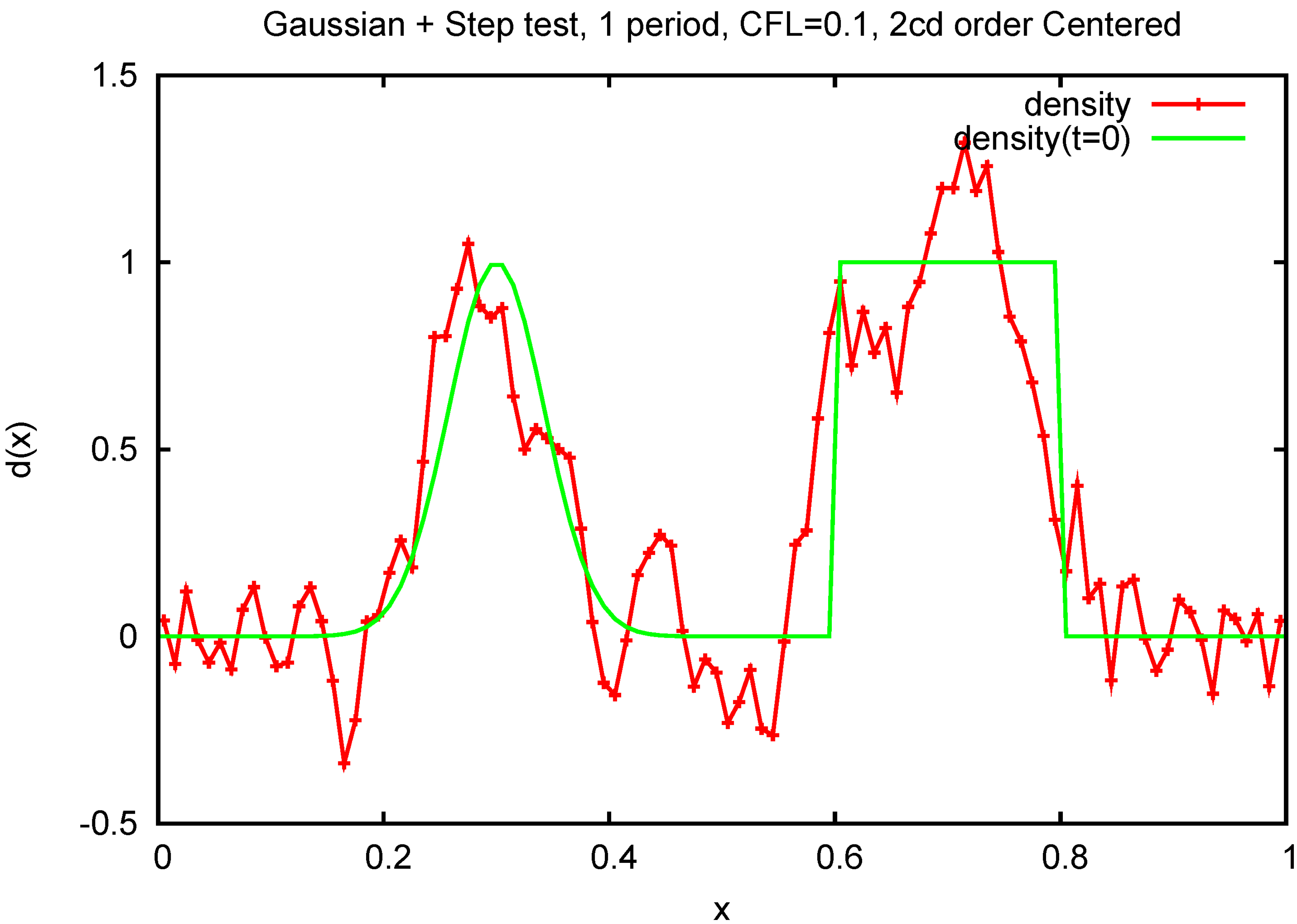
.

Simple test: Advection of Gaussian pulse in periodic box



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.

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



Paradigm for problems with shocks (near discontinuities) or other under-resolved features.

Upwind 1st order fluxes

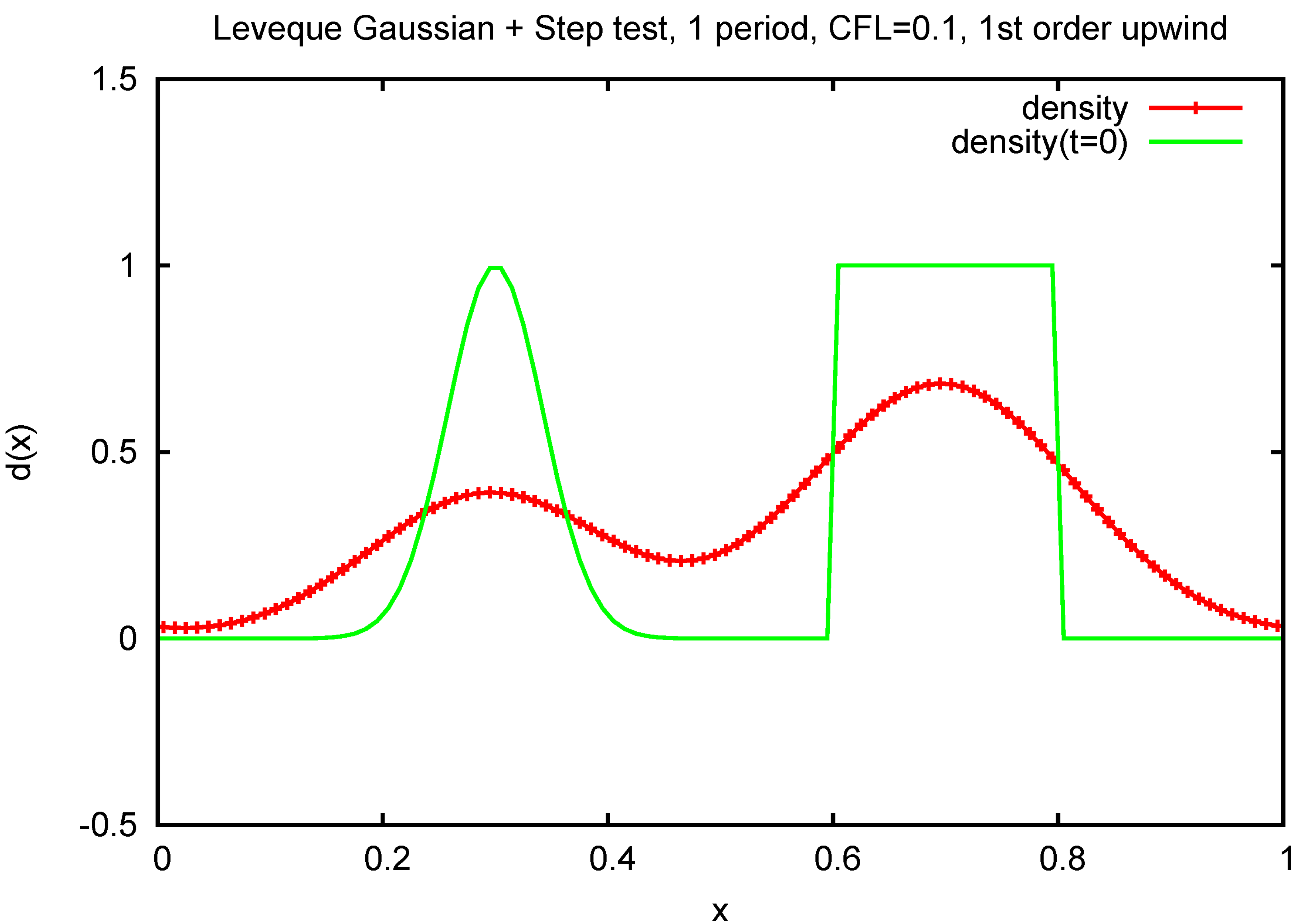
$$\frac{\partial}{\partial t}\Delta x f_i = -v f_{i+1/2} + v f_{i-1/2}$$

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 $v < 0$).

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

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 2cd order flux):

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

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

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

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

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

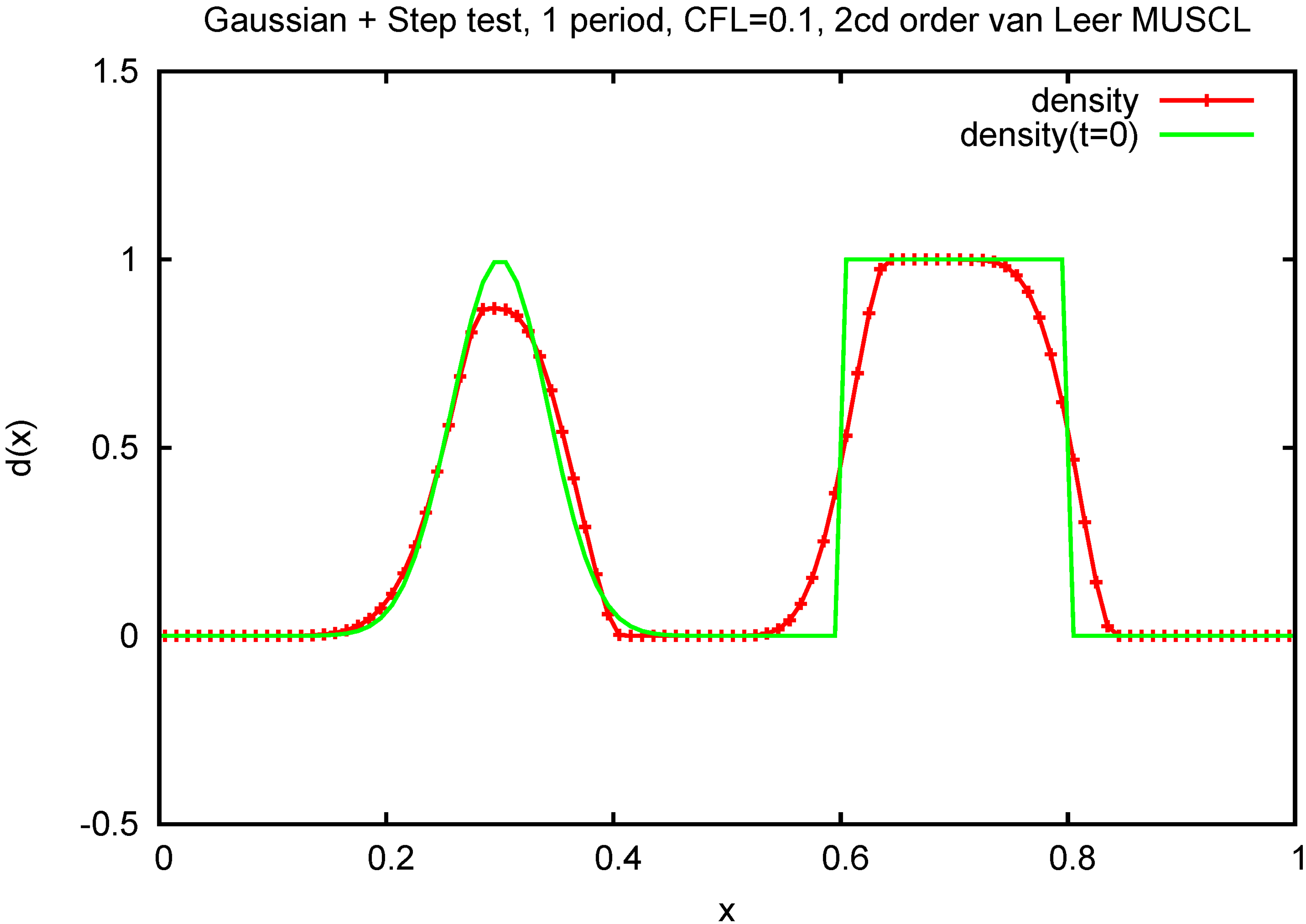
(minmod(a,b,c) = sign(a) * min(|a|,|b|,|c|) if a, b, and c all have same sign, otherwise minmod(a,b,c)=0.)

In smooth regions, $s_{i+1/2} \approx s_{i-1/2}$, and $f_{i+1/2}$ is 2cd 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 $s_i \approx 0$.

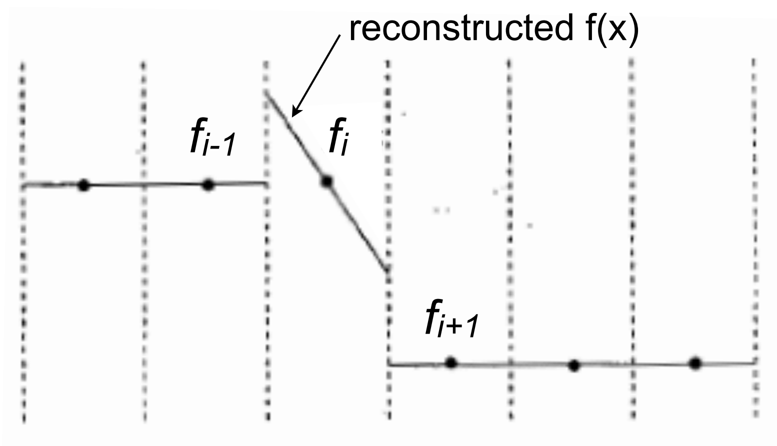
Godunov's theorem: to guarantee avoiding artificial oscillations, a linear algorithm can only be 1st order (and very diffusive). This algorithm is nonlinear.

High-Order Upwind Method with Limiters Do Fairly Well

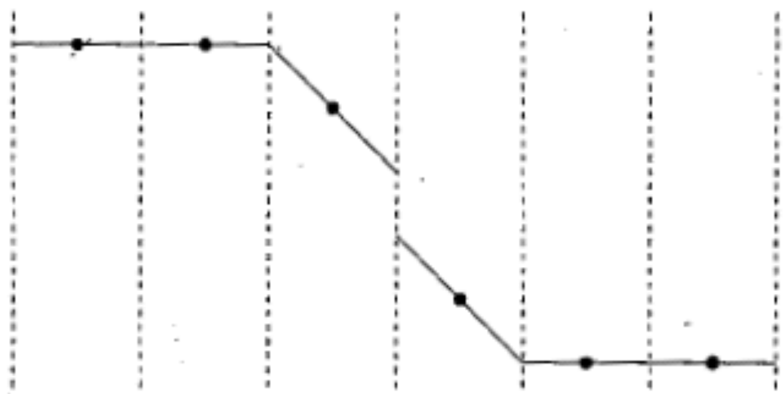
(except for some clipping of extrema)



3rd order SSP-RK used here. Looks better at CFL=0.5 with 2cd 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.



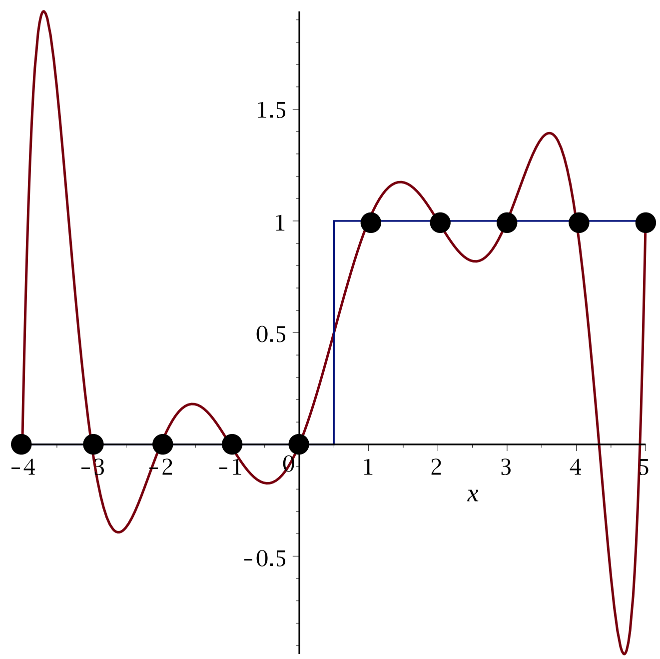
MC limiter never allows reconstructed $f(x)$ to overshoot average f in adjacent cell. Much more robust.

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.

From R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge Univ. Press (2002).

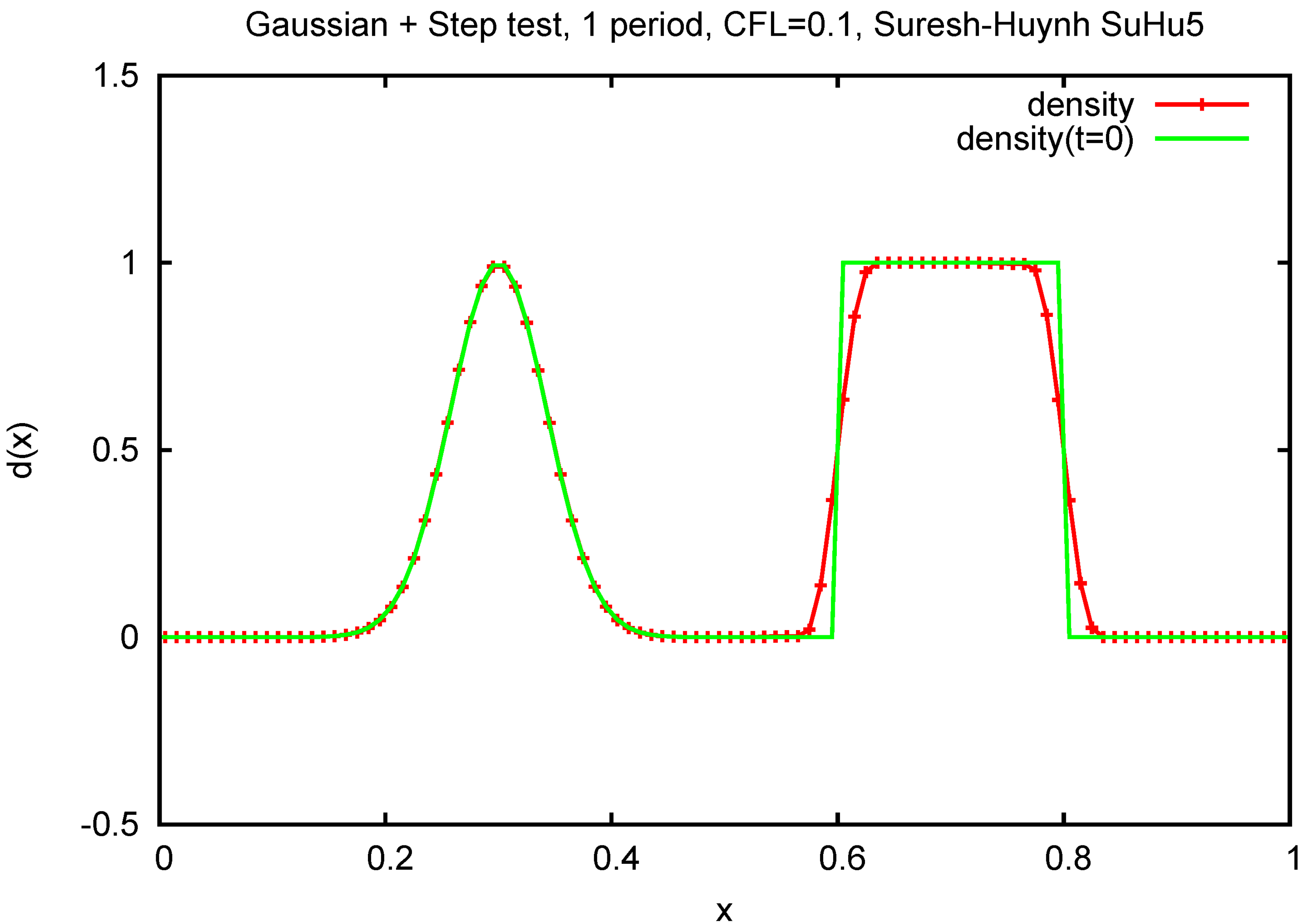
9th order polynomial fit to step function at 10 discrete points:



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

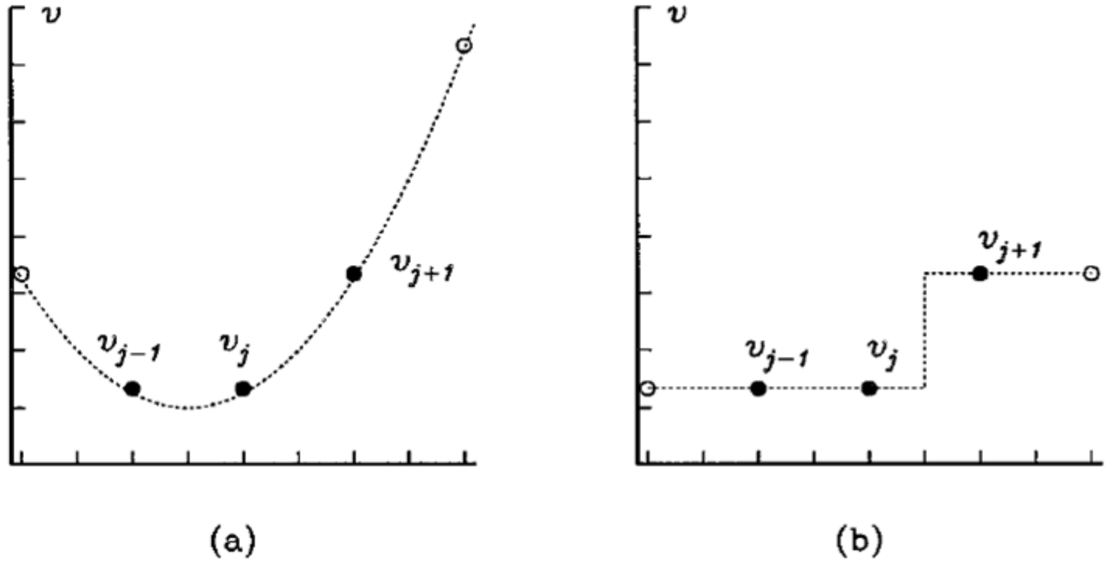


(Suresh-Huynh 1997, Colella-Sekora 2008 comparable)

(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 overshoots (2cd order in smooth regions, but const. near extrema, "clipping")
- **TVD** (Total Variation Diminishing) (variations of 2cd order van Leer)
- Colella & Woodward 84 **PPM** (Piecewise Parabolic Method) (4th order for smooth solutions, except const. near extrema) Widely-used gold-standard.
- **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-)

New limiters distinguish between smooth extrema & discontinuities



Suresh, Huynh '97

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).

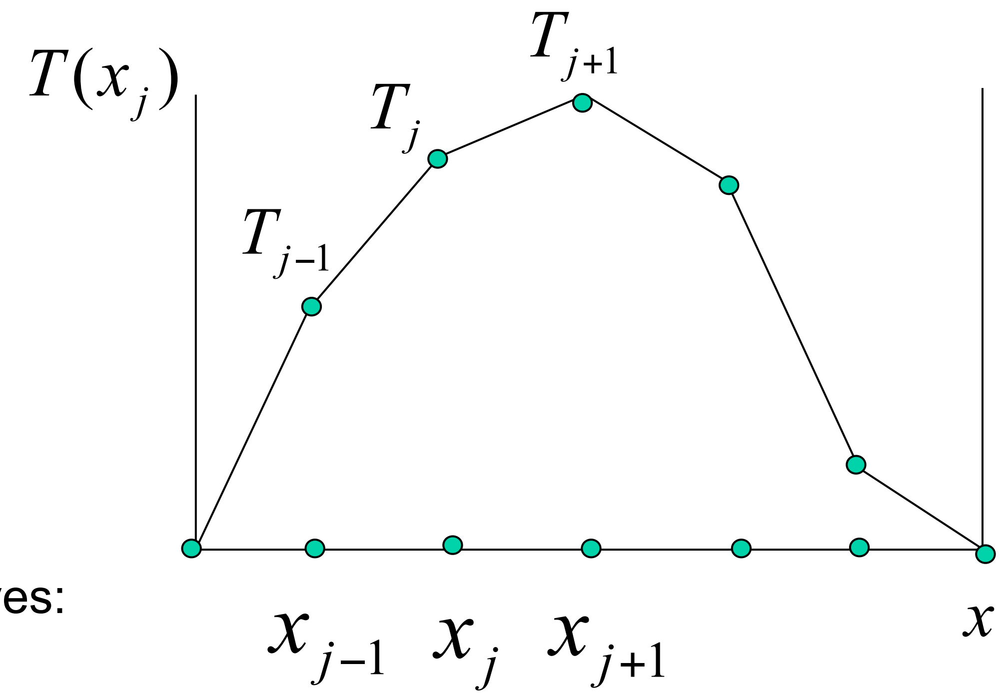
- Main idea behind these algorithms: detect discontinuities / under-resolved features, revert to lower-order polynomial in non-smooth regions, allow discontinuities (allowed for by hyperbolic eqs.), introduce minimum necessary numerical diffusion in non-smooth regions to preserve (or encourage) monotonicity, positivity.
- Suresh-Huynh ('97): relaxed previous limiters to allow higher order interpolations near smooth extrema, 5^{th} order in smooth regions, essentially a more efficient way to implement WENO
- Colella-Sekora ('08): alternate way to relax piecewise-constant assumption at extrema, 4^{th} order in smooth regions (even order = no numerical diffusion in smooth regions)

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:

$$T(x_j, t) = T_j(t)$$

$$x_j = j\,\Delta x$$

Discrete analogs of 1st & 2cd derivatives:

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

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

Simple Discretization of a Diffusion Eq.

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

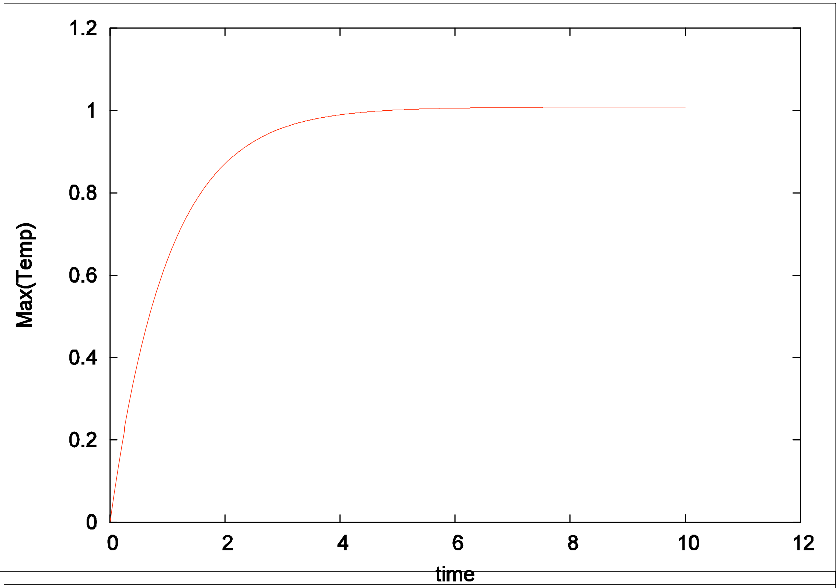
$$\frac{T_j(t+\Delta t) - T_j(t)}{\Delta t} = \frac{(T_{j+1} - T_j) - (T_j - T_{j-1})}{(\Delta x)^2} + \sin(x_j)$$

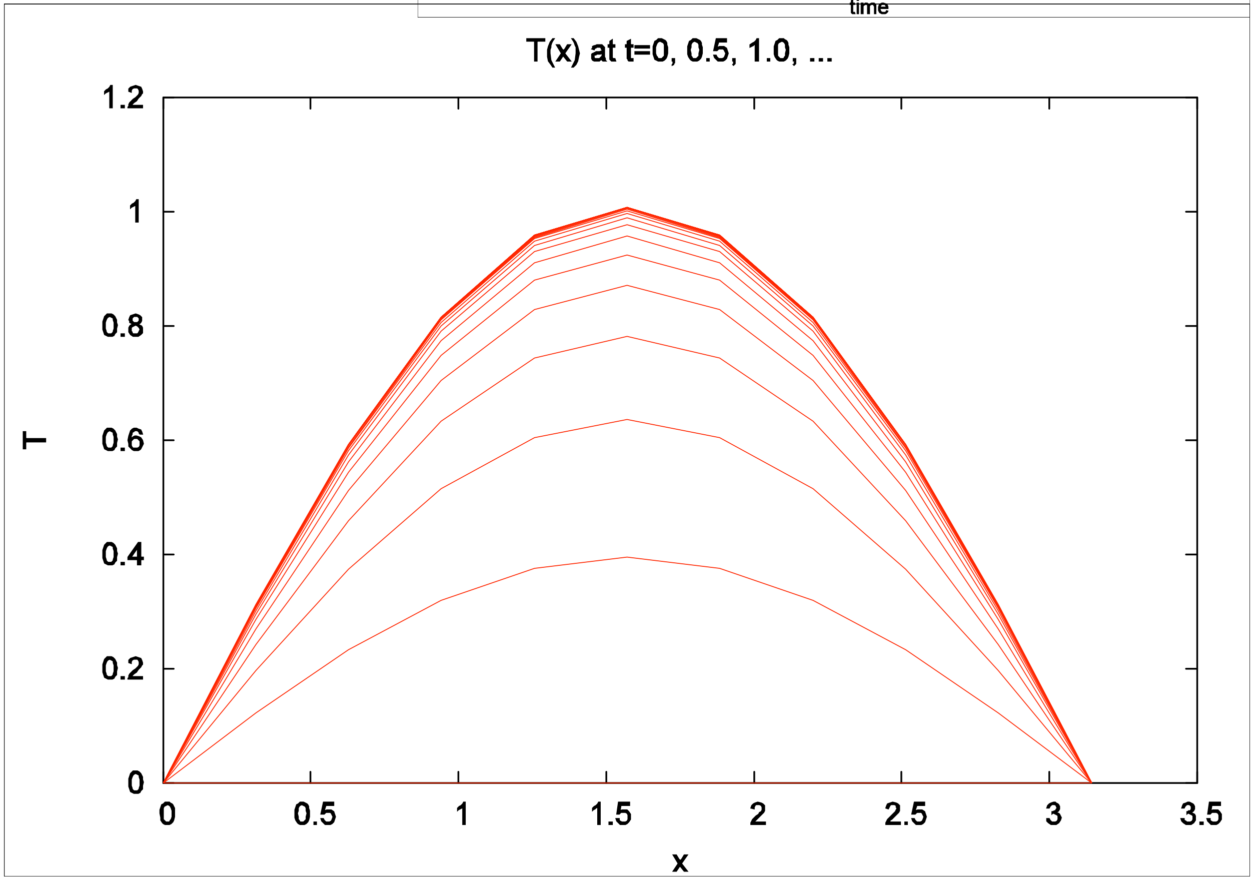
Given $T_j(t)$, loop over all positions j to get T_j (t+dt).
Repeat to find T(t+2dt) from T(t+dt)…

Test simple diffusion algorithm on a coarse mesh,
10 points for x=0 to x=pi,
dt=0.01

matches exact solution fairly well.

Everything "seems fine", just want to use finer spatial mesh…

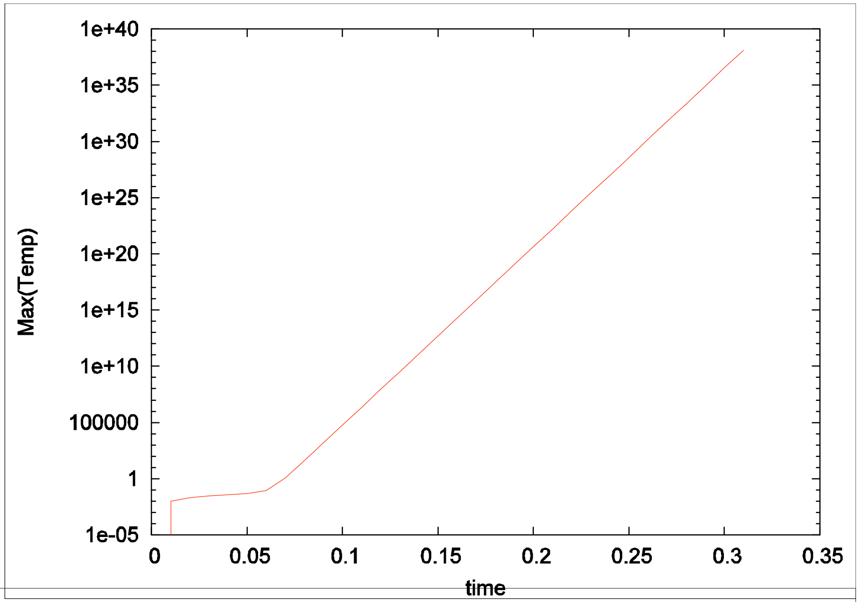


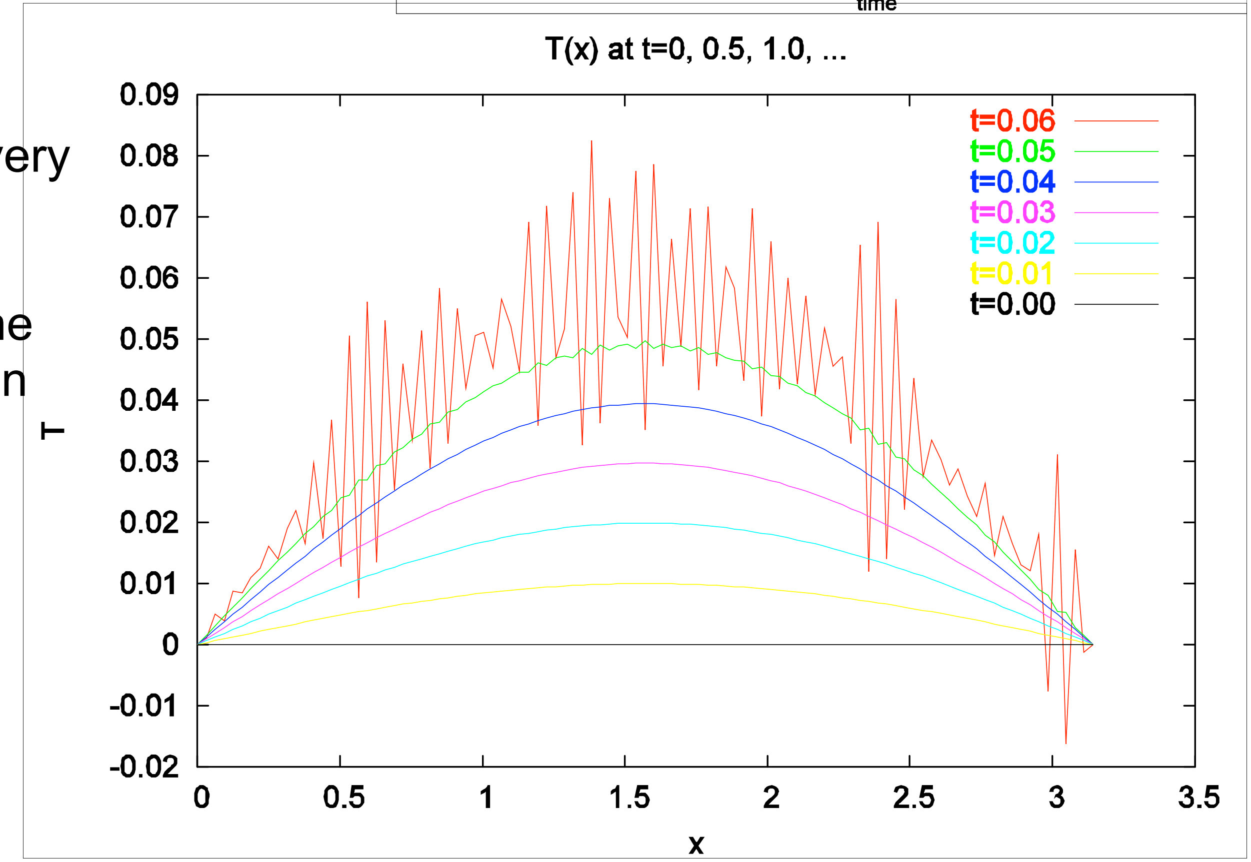


Test simple diffusion algorithm on a finer mesh,
increase from 10 to 100 points for x=0 to x=pi,
dt=0.01 (unchanged)

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 ≠ 1 modes

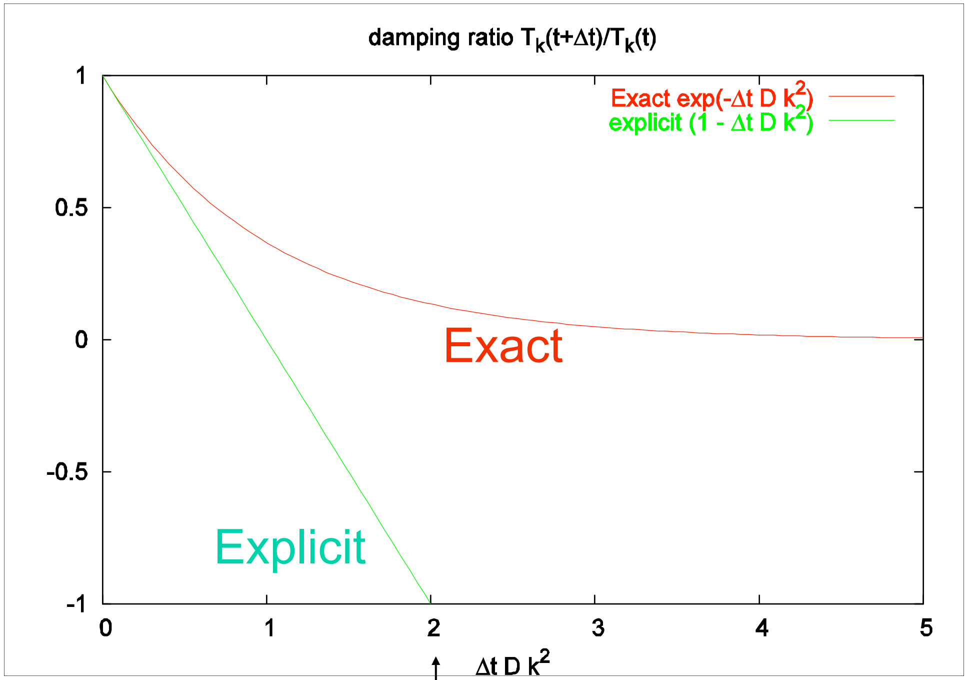
$$\frac{\partial}{\partial t}T_k = -Dk^2 T_k$$

Explicit integration (1st order "Euler"):

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

$$T_k(t+\Delta t) = \underbrace{\left(1 - \Delta t\, Dk^2\right)}_{\text{1st order Taylor series approx. to exact result } e^{-\Delta t\, D k^2}} T_k(t)$$

1st order Taylor
series approx. to
exact result $e^{-\Delta t\, D k^2}$



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"):

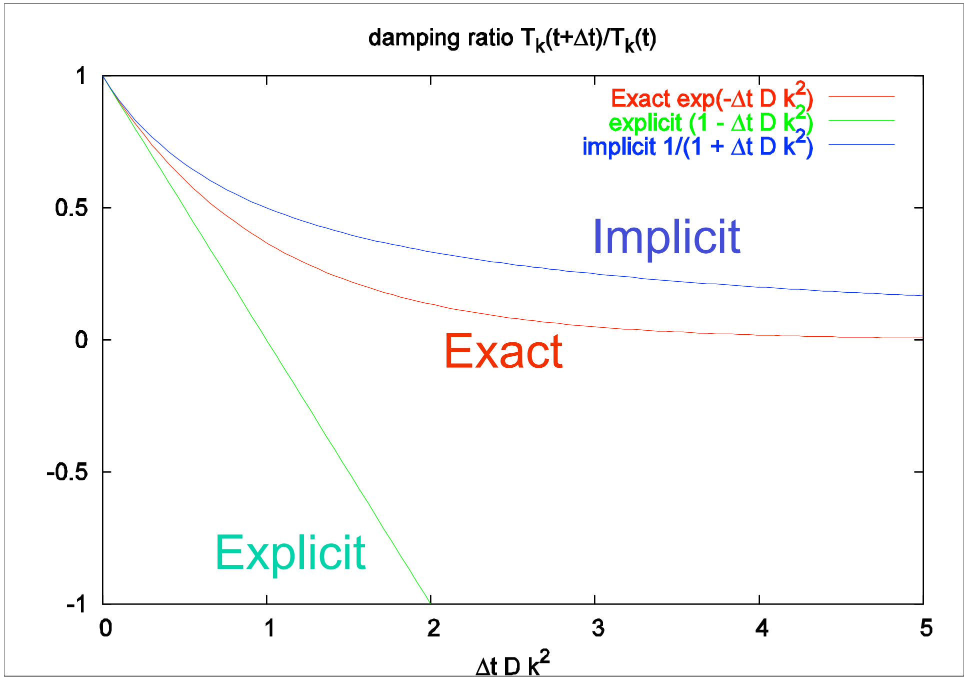
$$\frac{T_k(t+\Delta t) - T_k(t)}{\Delta t} = -Dk^2 T_k(t)$$

Implicit integration (1st order "Backwards Euler"):

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

$$T_k(t+\Delta t) = \frac{1}{\underbrace{\left(1 + \Delta t\, Dk^2\right)}_{\text{1st order Padé approx. to exact result } e^{-\Delta t\, D k^2}}} T_k(t)$$

1st order Padé approx. to exact result $e^{-\Delta t\, D k^2}$



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"):

$$y(\Delta t) = y(0) + \Delta t\, F(y(0))$$

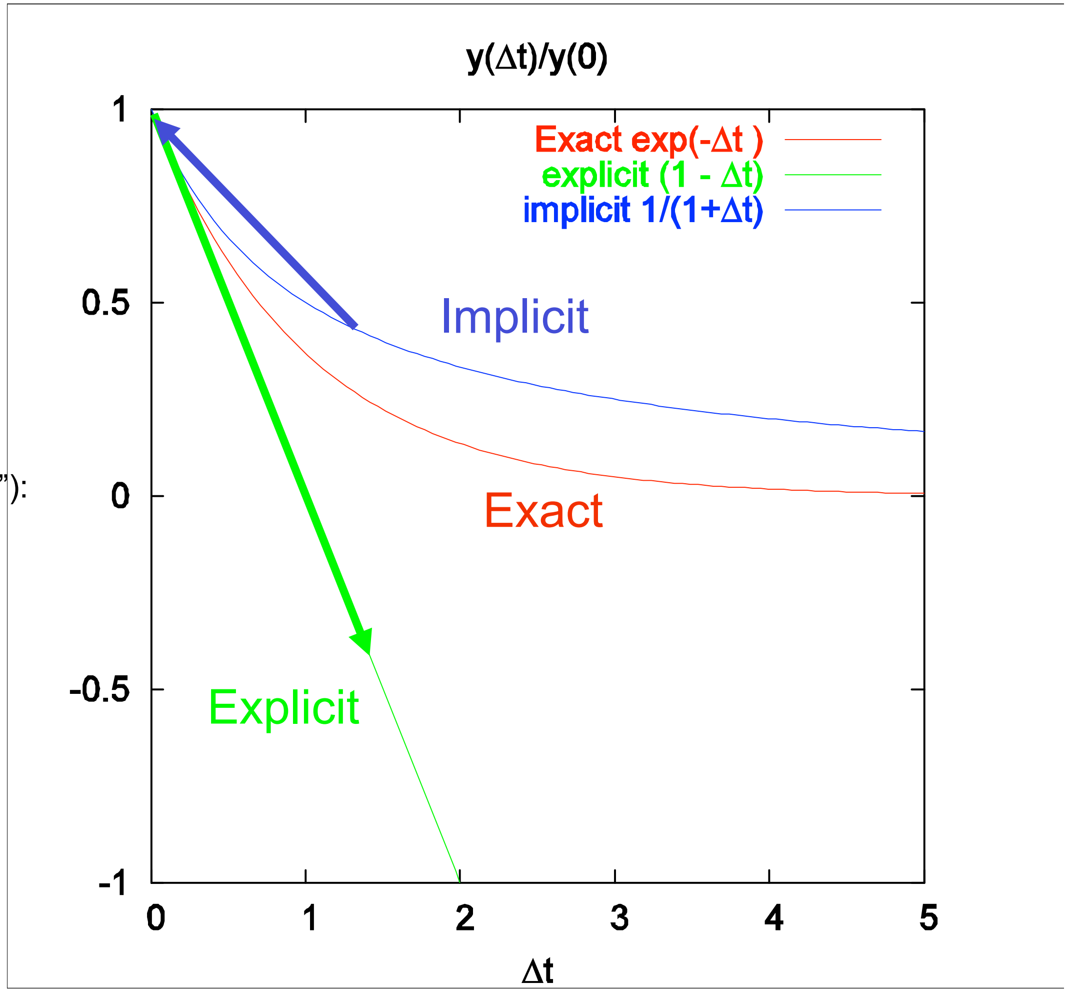
Implicit integration (1st order "Backwards Euler"):

$$y(\Delta t) = y(0) + \Delta t\, F(y(\Delta t))$$

Rearrange as:

$$y(0) = y(\Delta t) - \Delta t\, F(y(\Delta t))$$

Thus implicit algorithm
= integrating backwards in time,
to find what y(Δt) at the future time is
needed to give y(0) at the current time.
(Requires inverting operator 1-ΔtF …)



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:

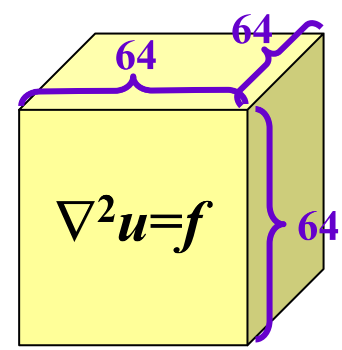
$$\frac{\partial \mathbf{Y}}{\partial t} = \mathbf{F}(\mathbf{Y})$$

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 $N=n^3$

Year	*Method*	*Reference*	*Storage*	*Flops*
1947	GE (banded)	Von Neumann & Goldstine	n^5	n^7
1950	Optimal SOR	Young	n^3	$n^4 \log n$
1971	CG	Reid Conjugate Gradients w/ Gustaffson's modified ILU preconditioner	n^3	$n^{3.5} \log n$
1984	Full MG	Brandt Multigrid	n^3	n^3



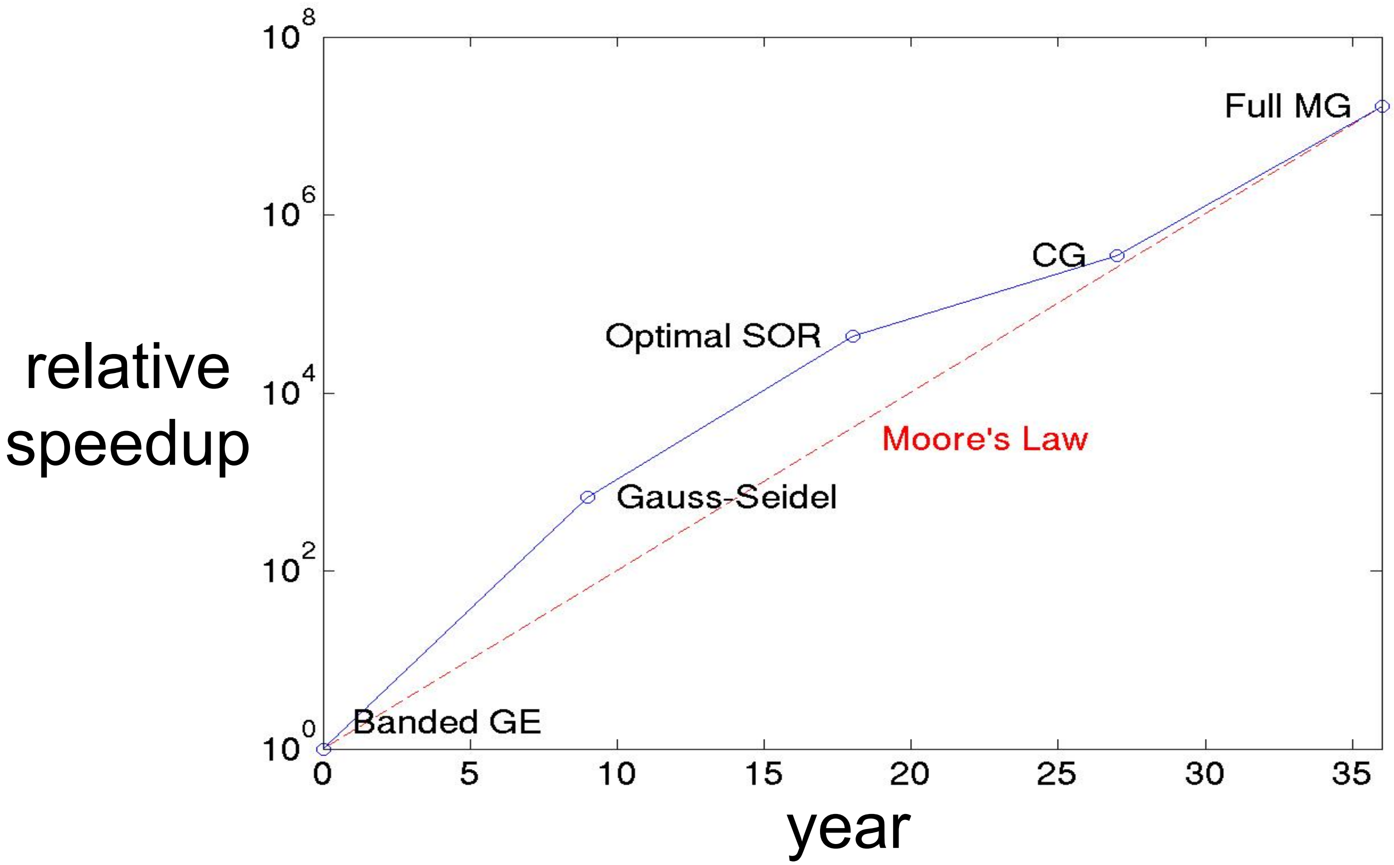
- If n=64, this implies an overall reduction in flops of ~16 million *

more recently: FMM Fast Multipole Method

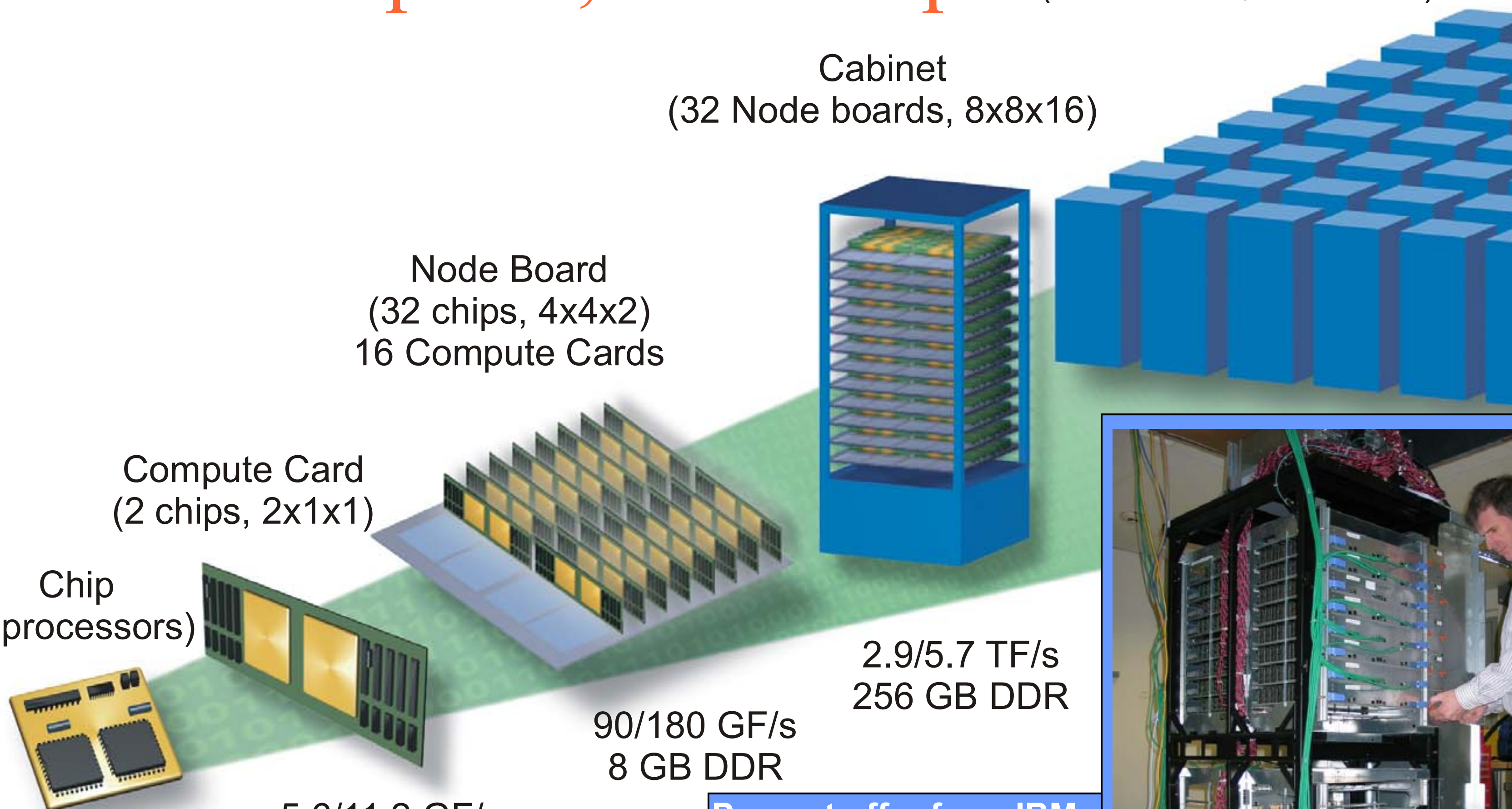
*Six-months is reduced to 1 s

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 $\Rightarrow$ the same as the factor from algorithms alone!



IBM’s BlueGene/L: 65536 dual procs, 180 Tflop/s



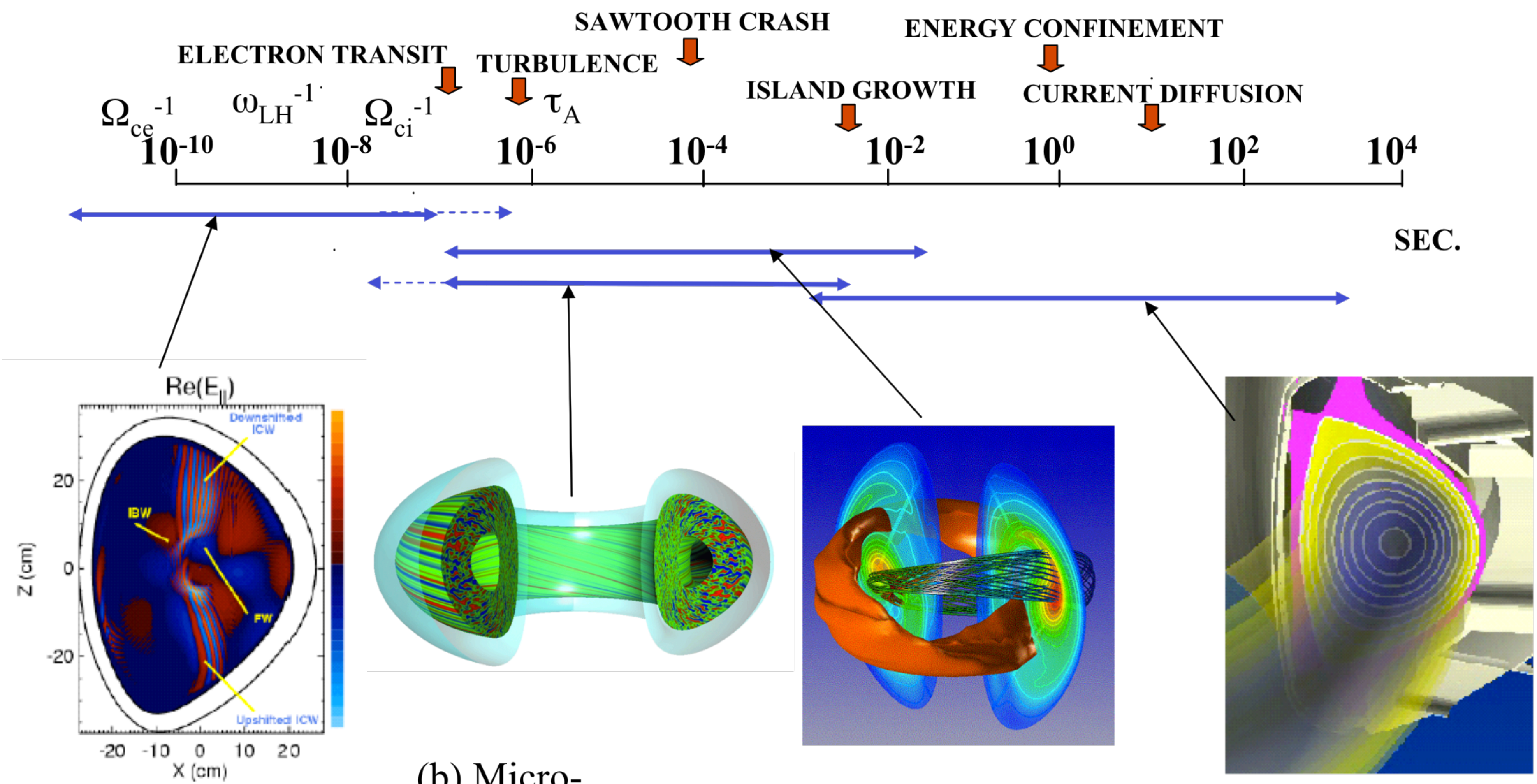
Present offer from IBM

Single cabinet
5.7 TFlop/s peak
$2M in acad. consortium

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



(a) RF codes

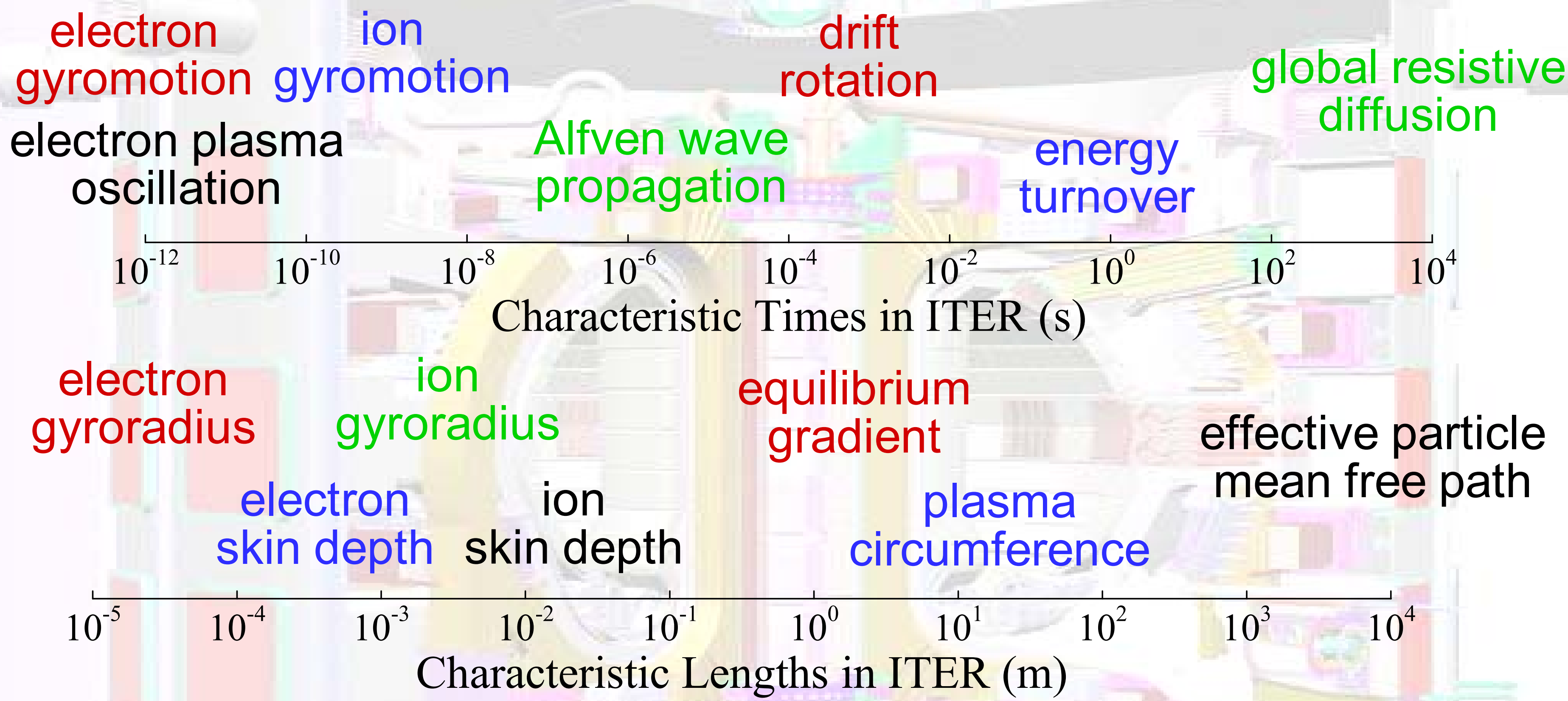
(b) Micro-turbulence codes

(c) Extended-MHD codes

(d) Transport Codes

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…

Carl Sovinec, http://www.csm.ornl.gov/workshops/SciDAC2005/SovinecTalk/scidac05_talk.pdf

Center for the Study of Plasma Microturbulence

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

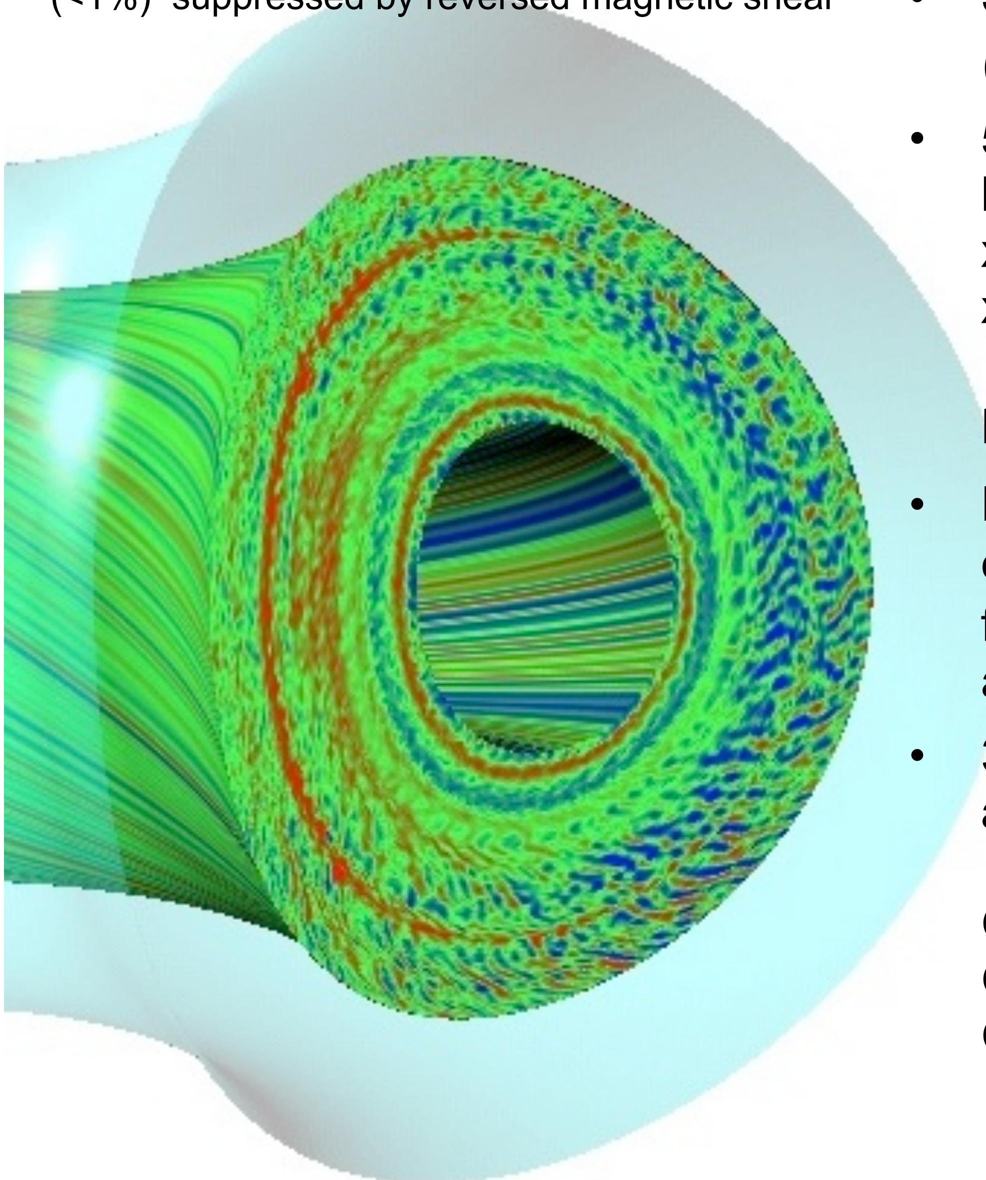
- devoted to studying plasma microturbulence through direct numerical sumulation

- 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

small scale, small amplitude density fluctuations (<1%) suppressed by reversed magnetic shear



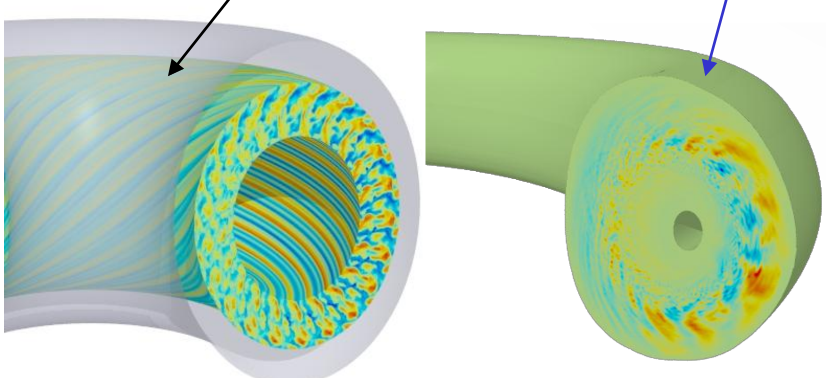
- Solve for the particle distribution function *f* $(r,\theta,\alpha,E,\mu,t)$ (avg. over gyration: 6D → 5D)
- 500 radii x 32 complex toroidal modes (96 binormal grid points)
  x 10 parallel points along half-orbits
  x 8 energies x 16 $v_{||}/v$
  12 hours on ORNL Cray X1E with 256 MSPs
- Realistic toroidal geometry, kinetic ions & electrons, finite-β 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.)

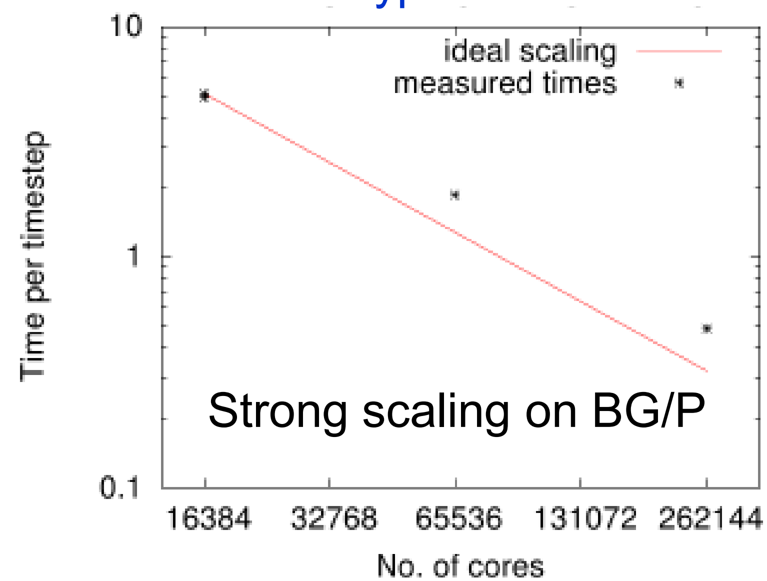
The gyrokinetic code GENE (gene.rzg.mpg.de)

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.

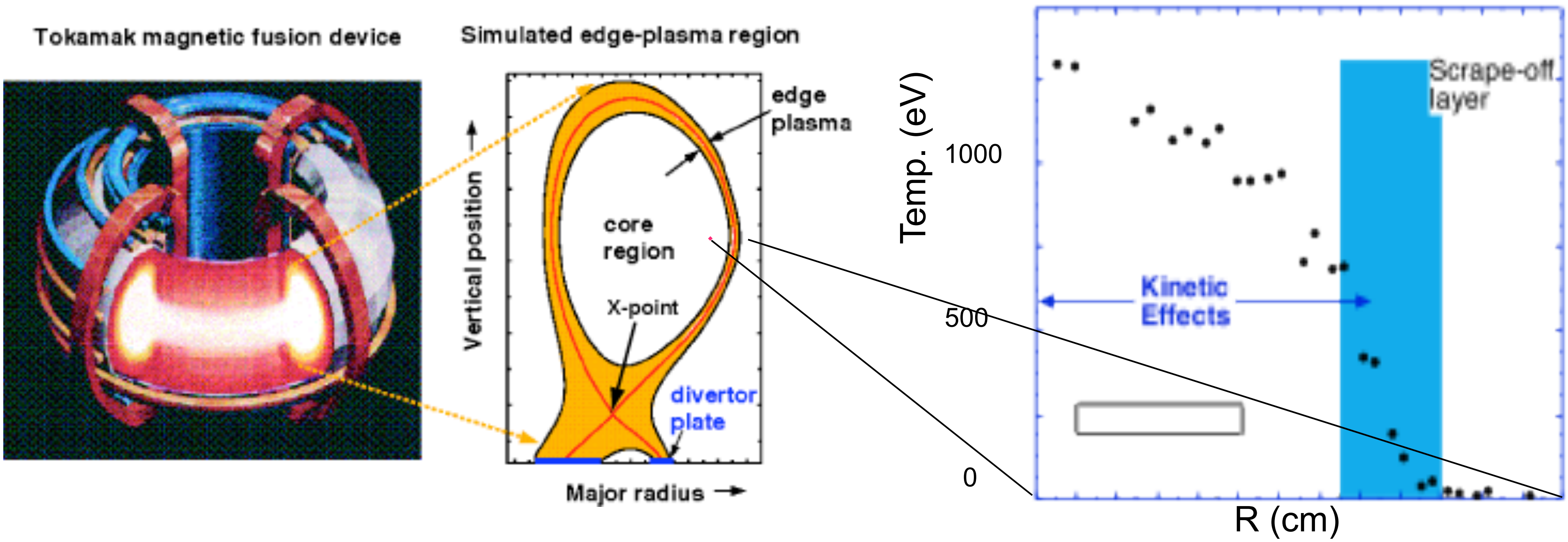
Major Theoretical & Algorithmic Speedups

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

- Nonlinear gyrokinetic equation
  - ion polarization shielding eliminates plasma freq. $\omega_{pe}/\Omega_{ci} \sim m_i/m_e$ — $x10^3$
  - ion polarization eliminates ρ_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 δf PIC, reduces noise, $(f_0/\delta f)^2 \sim 1/\rho_*^2$ — $x10^6$
- Field-aligned coordinates (nonlinear extension of ballooning coord.)

  $\Delta_{||} / (\Delta_{\perp} \, q \, R / a) \sim a / (q \, R \, \rho_*)$ — x70
- Flux-tube / Toroidal annulus wedge, ↓ simulation volume
  - $k_\theta \rho_i$ = 0, 0.05, 0.1, …, 1.0
    n = 0, 15, 30, …, 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 x 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

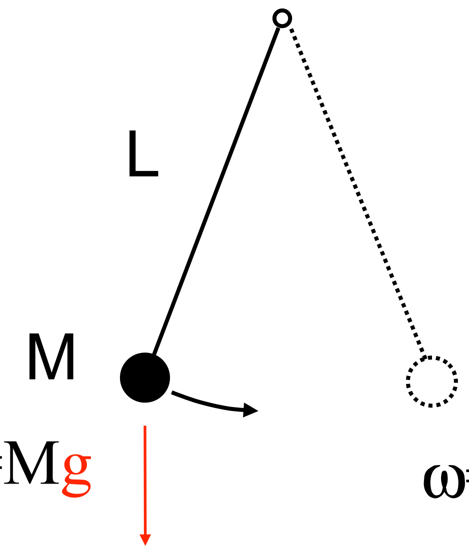


Edge pedestal temperature profile near the edge of an H-mode discharge in the DIII-D tokamak. [Porter*2000*]. Pedestal is shaded region.

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.

Stable Pendulum

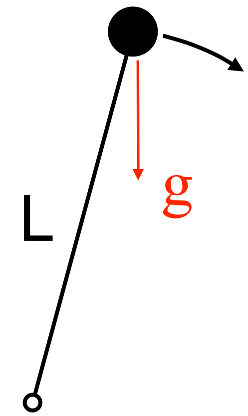


$F=Mg$

$\omega=(g/L)^{1/2}$

Unstable Inverted Pendulum

(rigid rod)

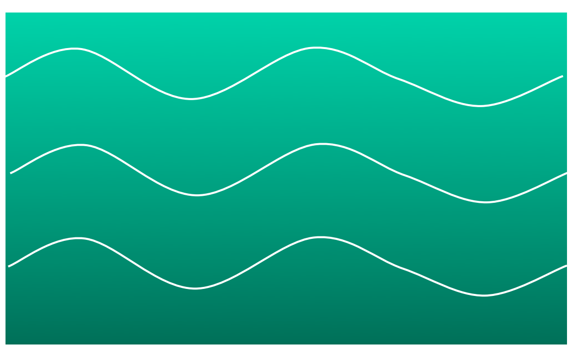


$\omega= (-g/|L|)^{1/2} = i(g/|L|)^{1/2} = i\gamma$

Instability

Density-stratified Fluid

$\rho=\exp(-y/L)$

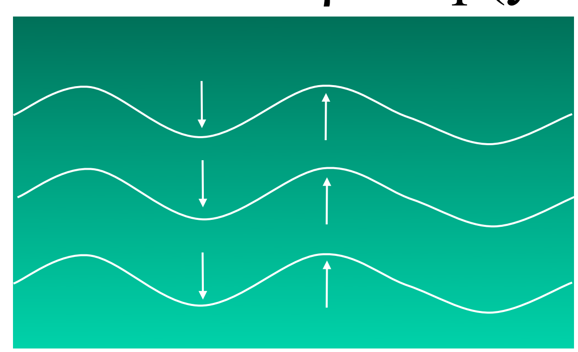


stable $\omega=(g/L)^{1/2}$

Inverted-density fluid

$\Rightarrow$Rayleigh-Taylor Instability

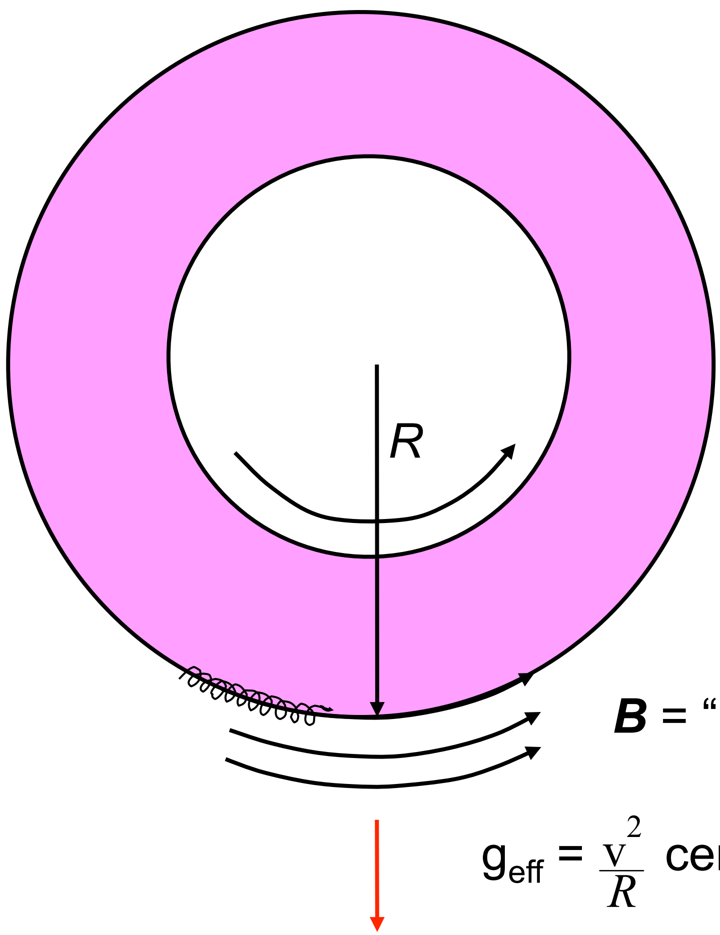
$\rho=\exp(y/L)$



Max growth rate $\gamma=(g/L)^{1/2}$

"Bad Curvature" instability in plasmas
≈ Inverted Pendulum / Rayleigh-Taylor Instability

Top view of toroidal plasma:



plasma = heavy fluid

B = "light fluid"

$g_{eff} = \frac{v^2}{R}$ centrifugal force

Growth rate:

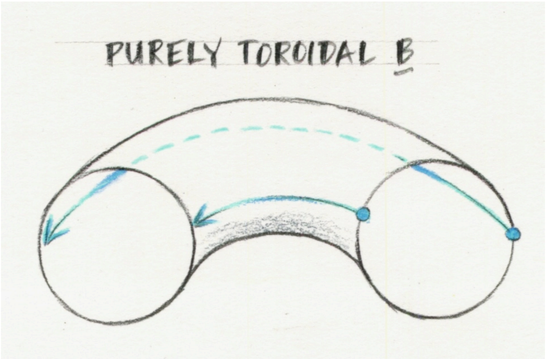
$$\gamma = \sqrt{\frac{g_{eff}}{L}} = \sqrt{\frac{v_t^2}{RL}} = \frac{v_t}{\sqrt{RL}}$$

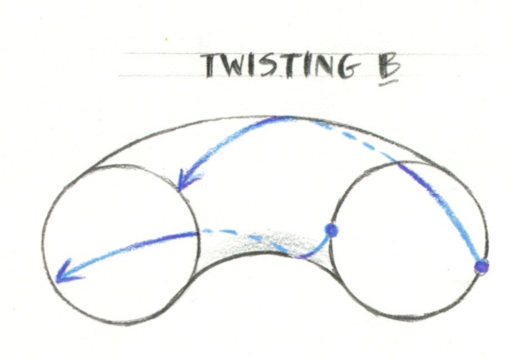
Similar instability mechanism
in MHD & drift/microinstabilities

$1/L = \nabla p/p$ in MHD,
$\propto$ combination of ∇n & ∇T
in microinstabilities.

The Secret for Stabilizing Bad-Curvature Instabilities

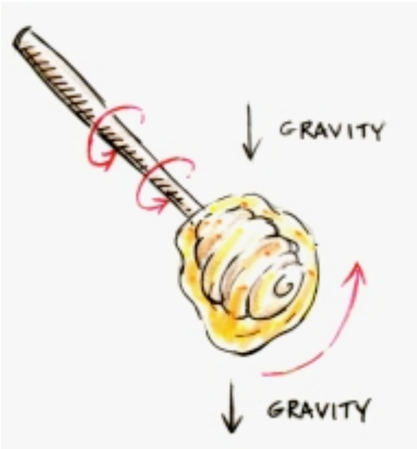
Twist in ***B*** carries plasma from bad curvature region to good curvature region:





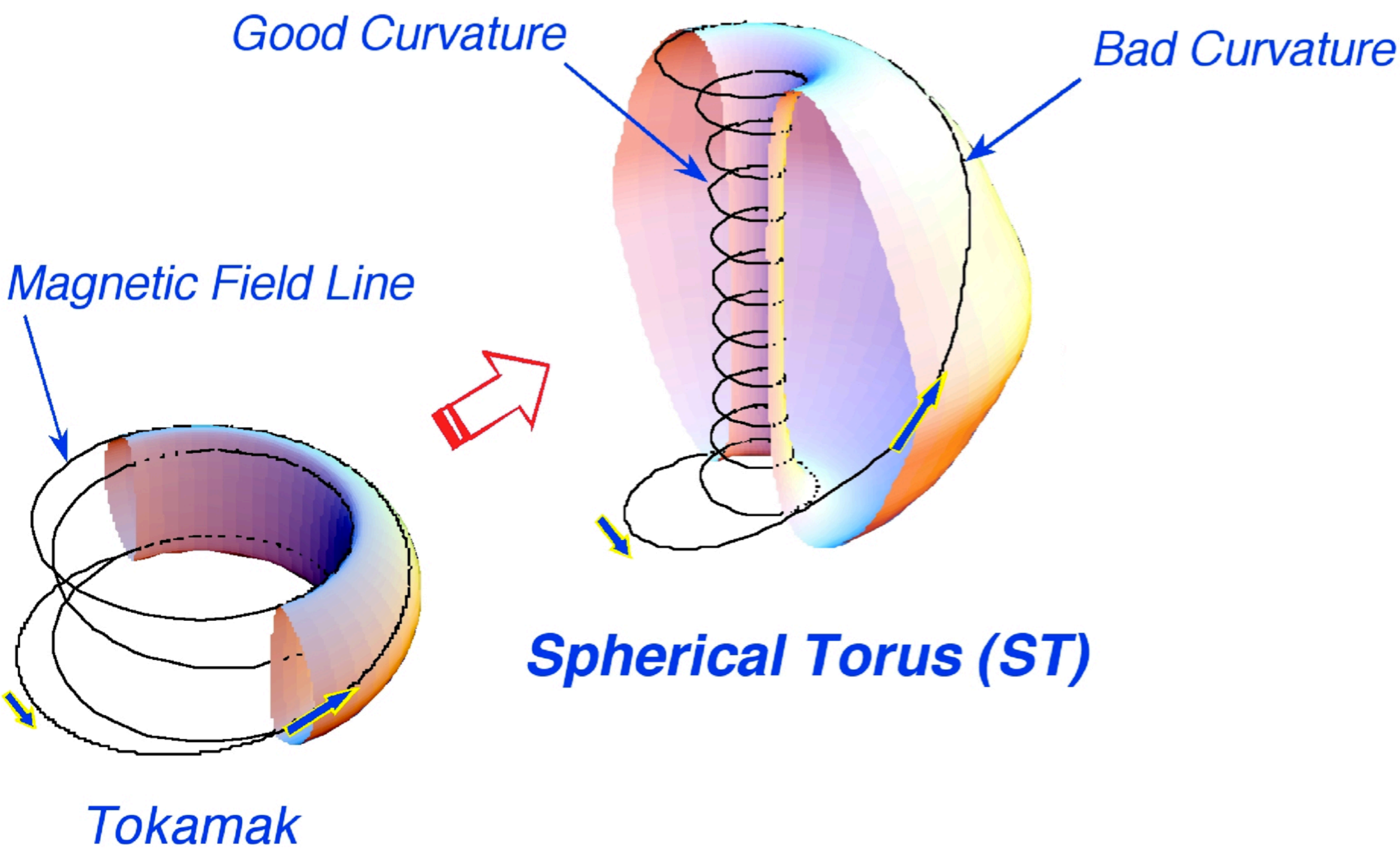
Unstable

Stable

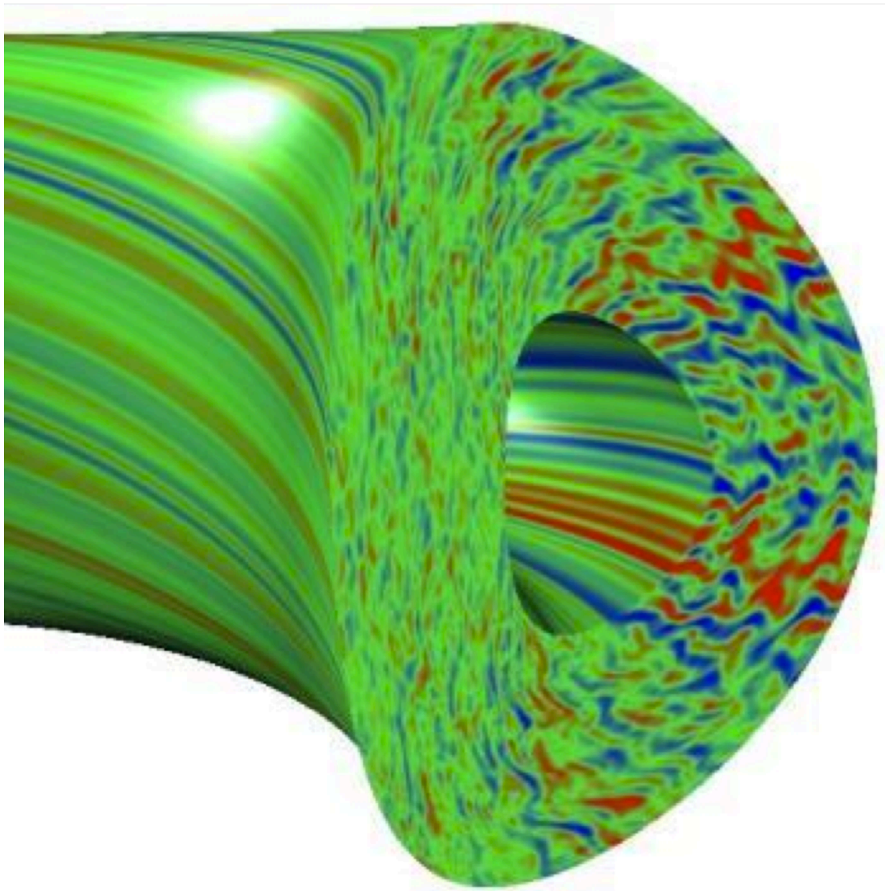


Similar to how twirling a honey dipper can prevent honey from dripping.

Spherical Torus has improved confinement and pressure limits (but less room in center for coils)



Understanding Turbulence That Affects the Performance of Fusion Device

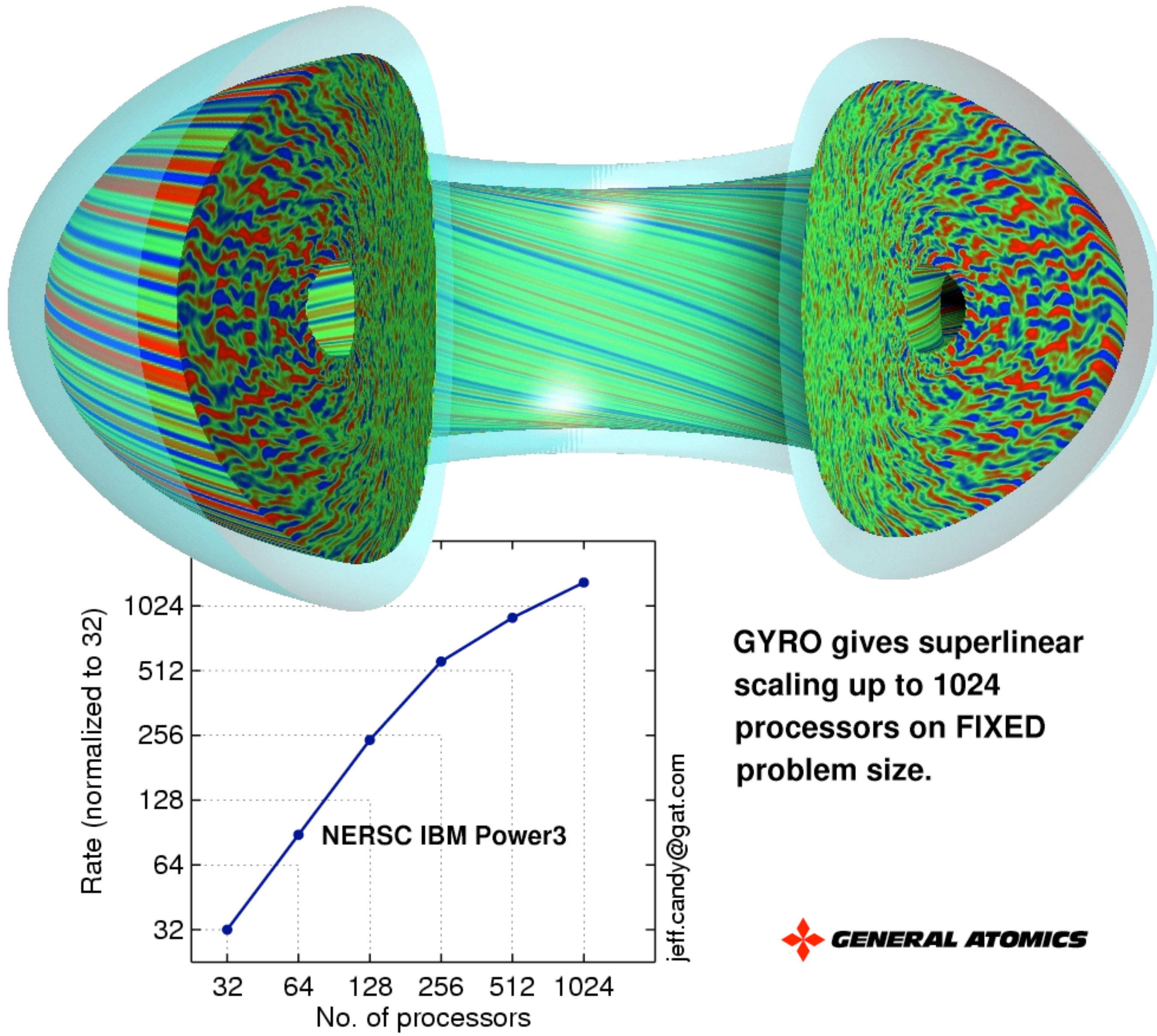


(Candy & Waltz, GA 2003)

Central temp ~ 10 keV ~ 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 $f(\vec{x}, v_{||}, v_{\perp}, t)$) & detailed understanding



GYRO gives superlinear scaling up to 1024 processors on FIXED problem size.

GENERAL ATOMICS

Comprehensive 5-D computer simulations of core plasma turbulence being developed by Plasma Microturbulence Project. Candy & Waltz (GA) movies shown: d3d.n16.2x_0.6_fly.mpg & supercyclone.mpg, from http://fusion.gat.com/comp/parallel/gyro_gallery.html (also at http://w3.pppl.gov/~hammett/refs/2004).

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-ρ_* 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

  $$f(\vec{\mathbf{x}}, E, v_{||}/v, t)$$

- Gyro-averaged Maxwell's Eqs. determine Electric and Magnetic fields
- "typical" grid 96x32x32 spatial, 10x20 velocity, x 3 species for 10^4 time steps.
- Various advanced numerical methods: implicit, semi-implicit, pseudo-spectral, high-order finite-differencing and integration, efficient field-aligned coordinates, Eulerian (continuum) & Lagrangian (particle-in-cell).

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):

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

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.

$$\chi = J_0\left(\frac{k_\perp v_\perp}{\Omega_c}\right)\left(\phi - \frac{v_{||}}{c}A_{||}\right) + \frac{J_1(k_\perp v_\perp/\Omega_c)}{k_\perp v_\perp/\Omega_c}\frac{m v_\perp^2}{q}\frac{\delta B_{||}}{B}$$

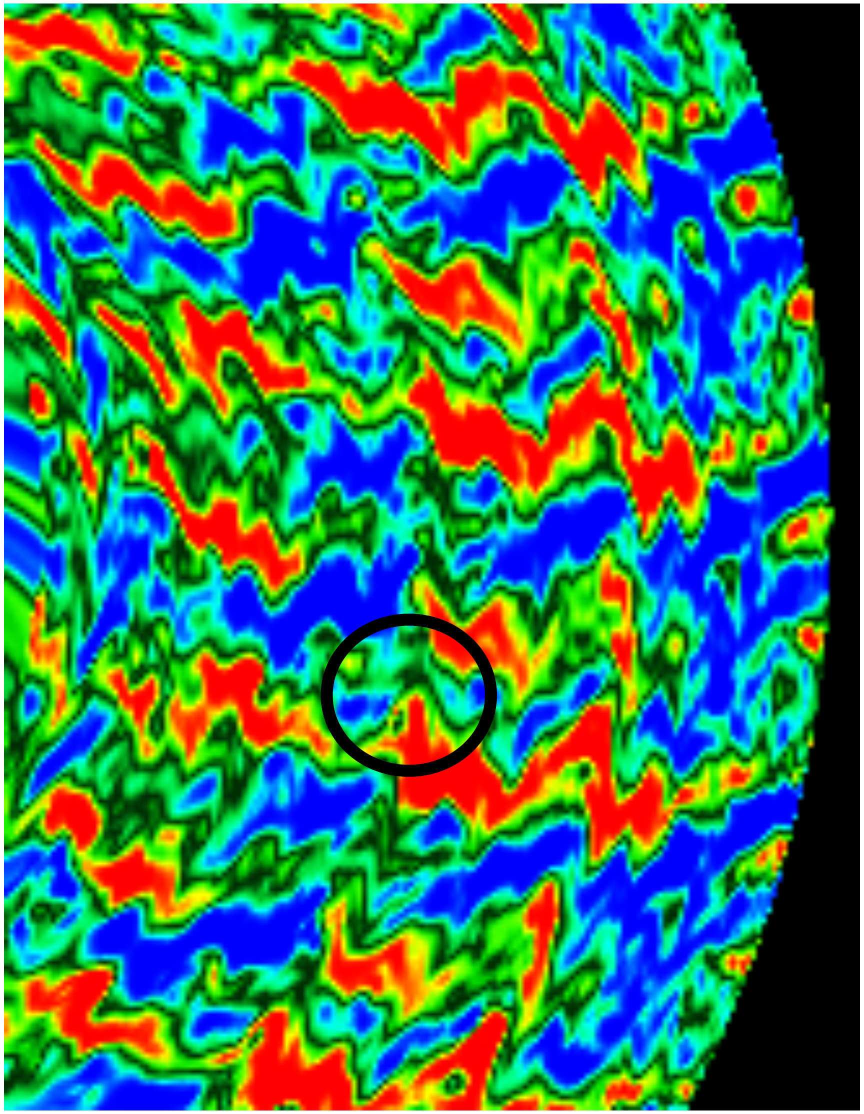
Bessel Functions represent averaging around particle gyro-orbit

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.

$$\chi = J_0(k_\perp \rho)\,\Phi$$

$$\chi(\vec{\mathbf{x}}) = \oint d\theta \;\; \Phi(\vec{\mathbf{x}} + \vec{\mathbf{n}}(\theta))$$

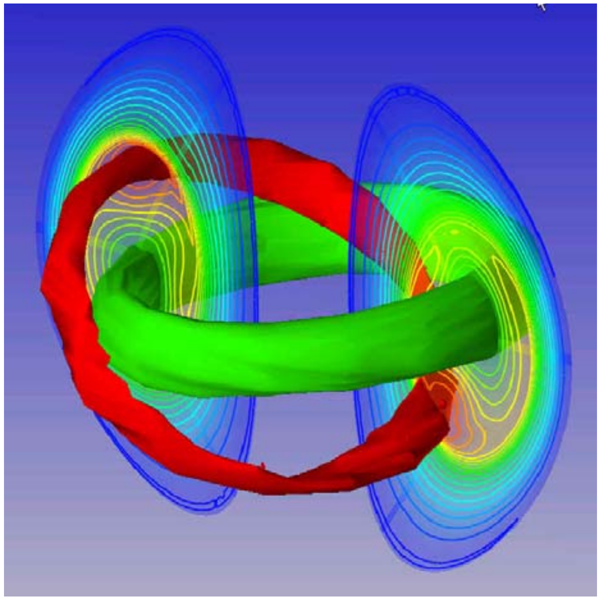


Examples of Nonlinear Macroscopic Simulation

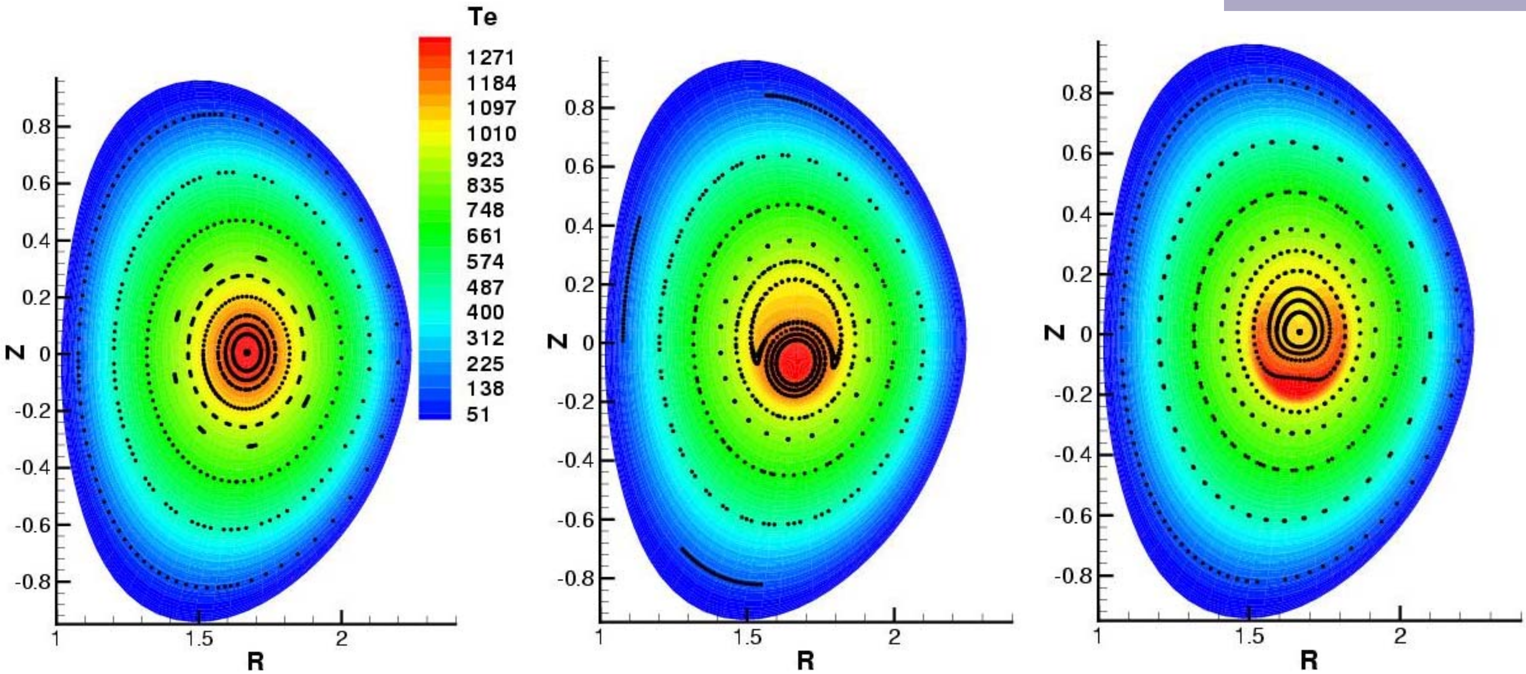
1) MHD evolution of the tokamak internal kink mode (m=1, n=1)

- Plasma core is exchanged with cooler surrounding plasma.

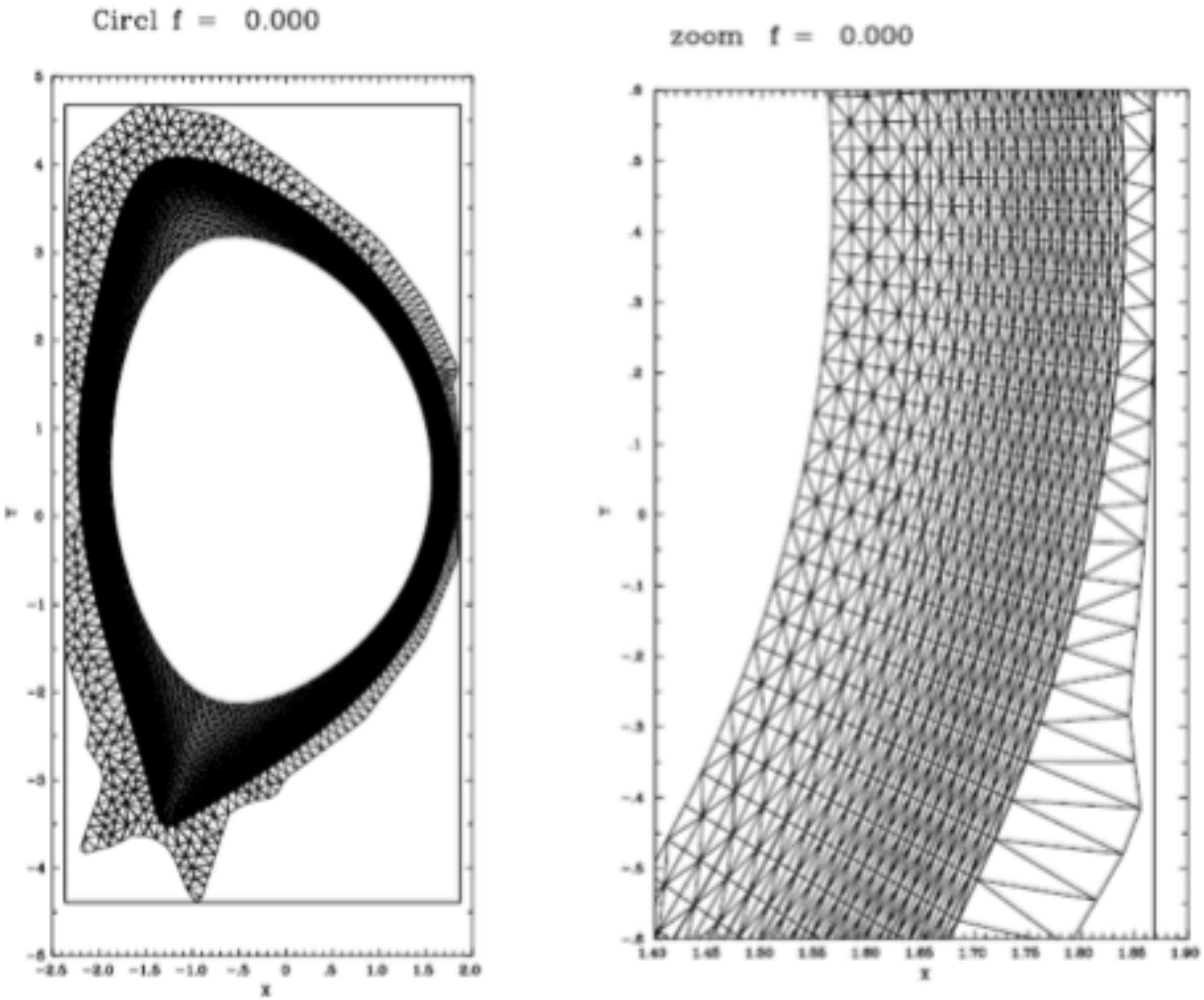
M3D simulation of NSTX [W. Park]



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

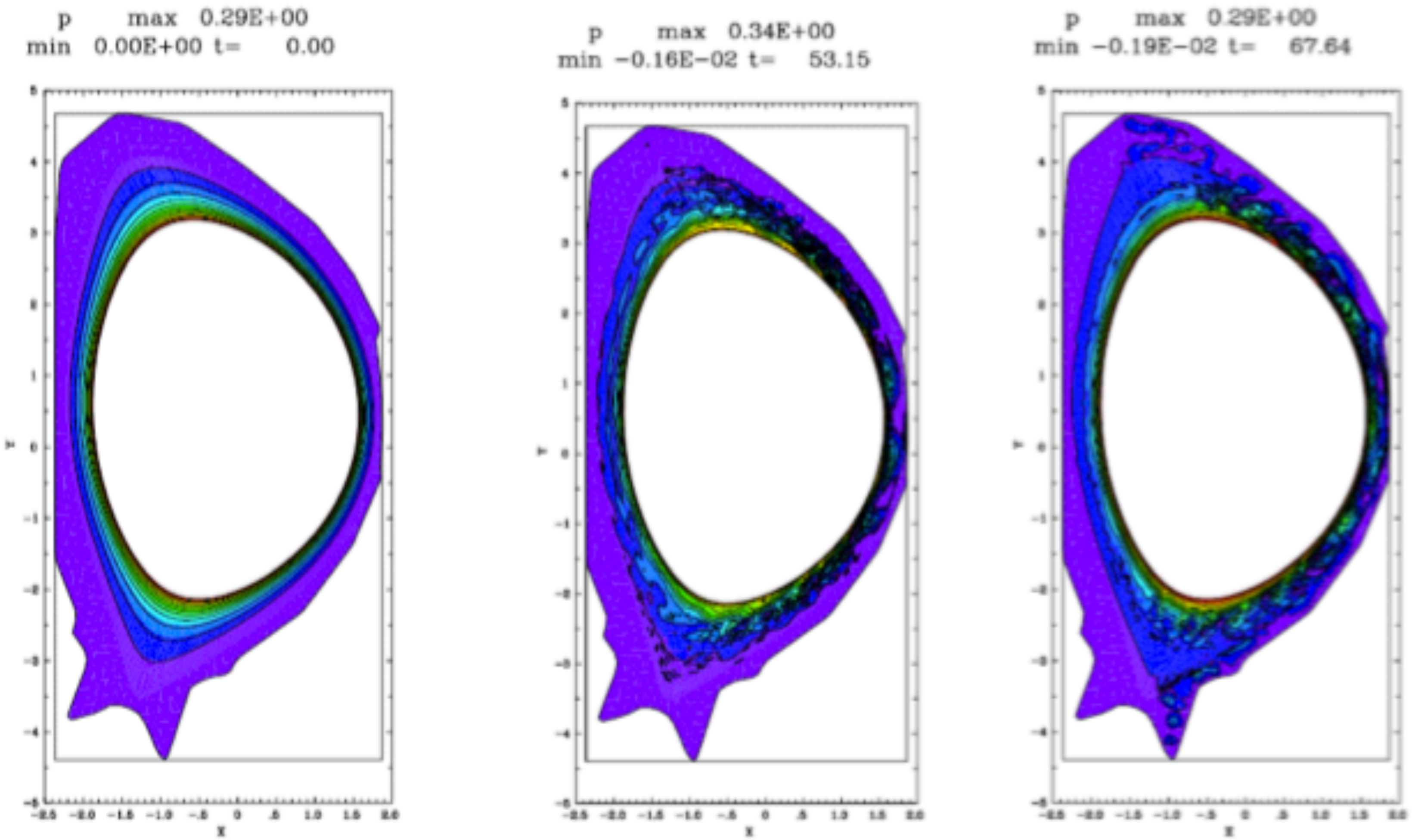


ITER mesh



Strauss

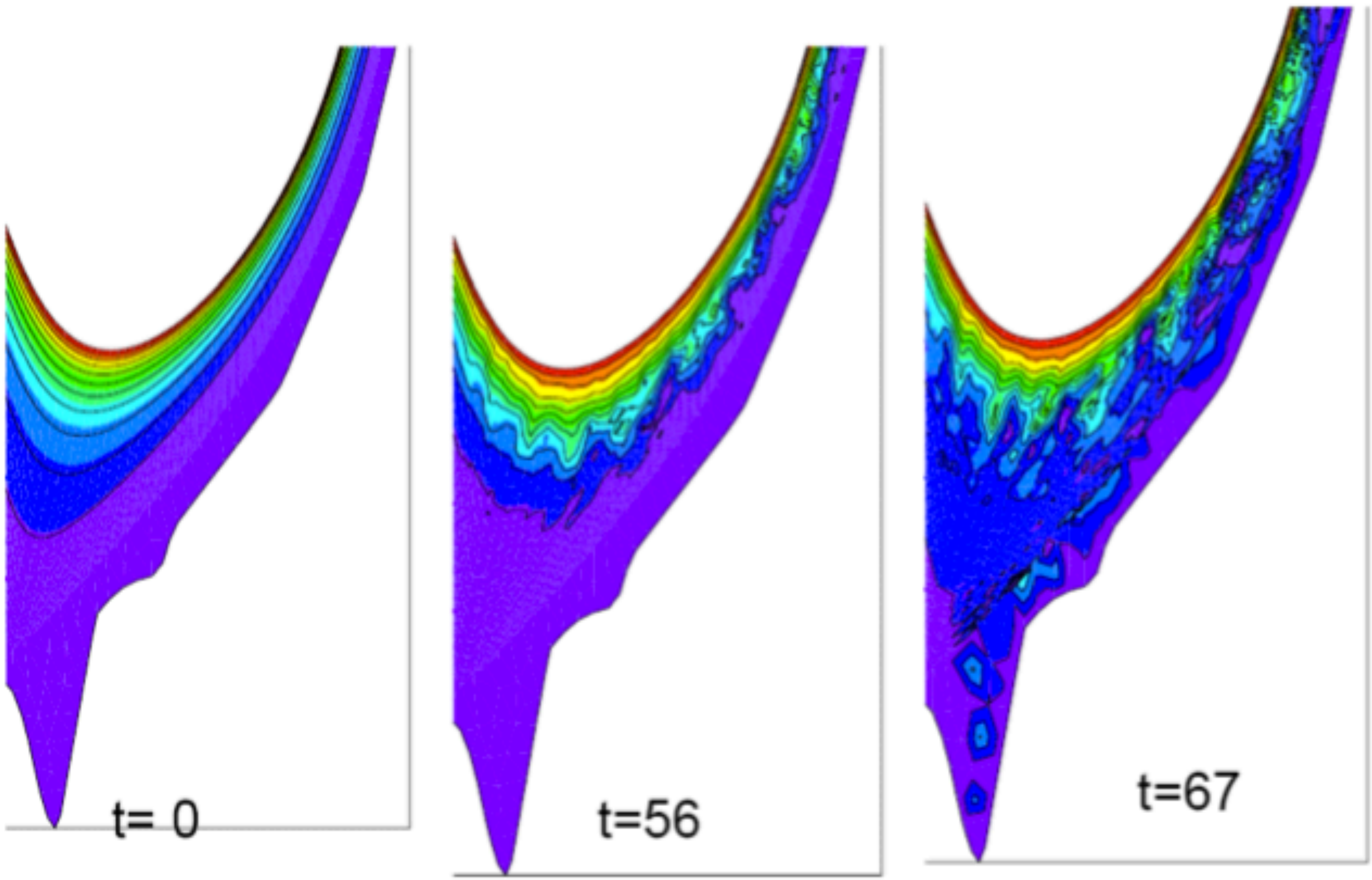
ITER ELM: pressure time evolution



Strauss

Center for Extended Magnetohydrodynamic Modelling (CEMM), S. Jardin et al., 2006 http://w3.pppl.gov/theory/PSACI%2006/Jardin%20CEMM(2006)-rev.pdf

ELM pressure: initial, mode growth, outflow



Strauss

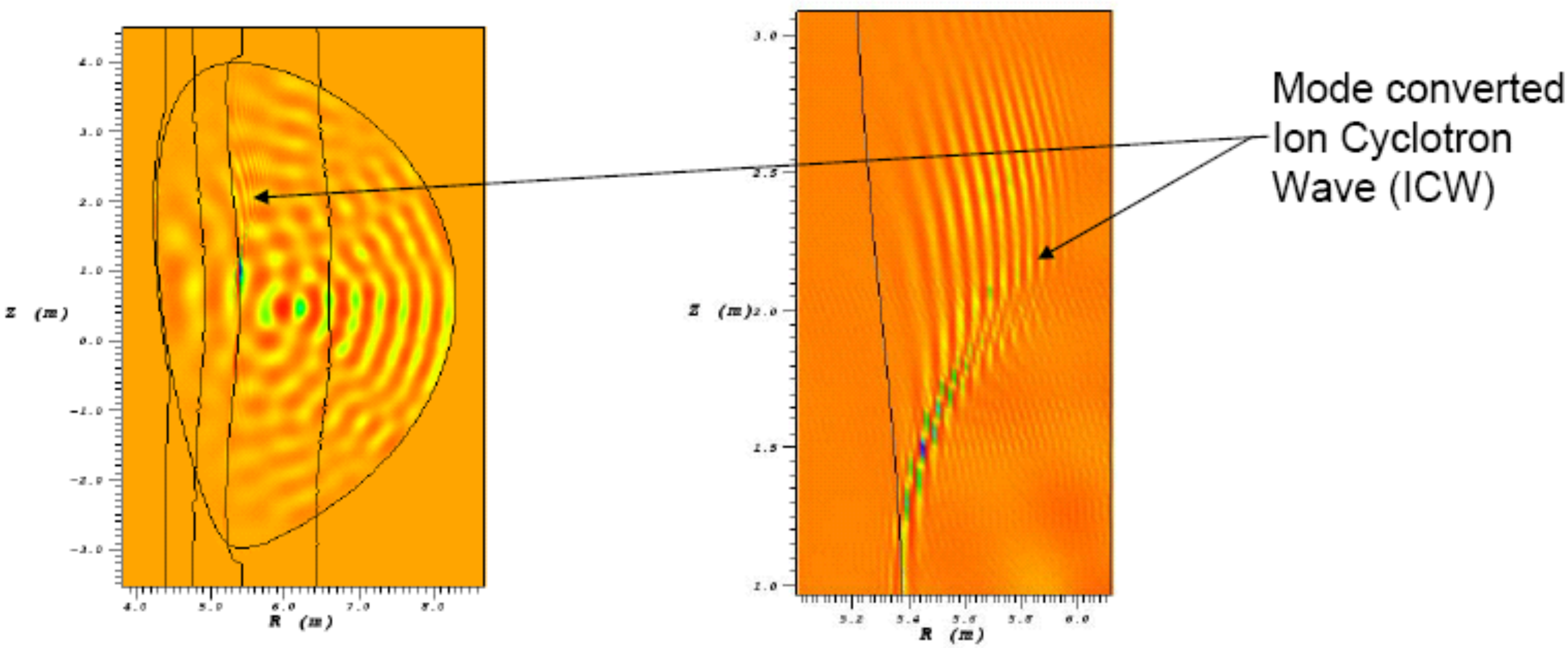
Center for Extended Magnetohydrodynamic Modelling (CEMM), S. Jardin et al., 2006 http://w3.pppl.gov/theory/PSACI%2006/Jardin%20CEMM(2006)-rev.pdf

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, $n = 2.5\text{x}10^{19}$ m^{-3}

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

E_perp | Blowup (E_parallel)



Future Work – Will extend this MC scenario to more ITER relevant densities ($\approx 7 \times 10^{19}$m^{-3})

Paul Bonoli et al., 2006, Center for Simulation of Wave-Plasma Interactions http://w3.pppl.gov/theory/PSACI%2006/Bonoli%20PSAC_RFSciDAC_Rev.pdf

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

- **Numerical Recipes**, Press, Teukolsky, Vetterling, and Flannery, http://www.nr.com (in C/C++, Fortran 77/90) entertainingly written, many insights, good place to start
- 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

- R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge Univ. Press (2002).
- D. R. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics
- Arakawa. *J. Comput. Phys.* **1**, 119-+ (1966).
- Colella and Sekora. *J. Comput. Phys.* **227**:15, 7069-7076 (2008).
- Liu, Osher and Chan. *J. Comput. Phys.* **115**, 200-212 (1994).
- Martin and Colella. *Private Communication* (2008).
- Naulin and Nielsen. *SIAM J. Sci. Comput.* **25**:1, 104-126 (2003).
- Very good tutorial on ENO/WENO: "Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws," Chi-Wang Shu, NASA/CR-97-206253, ICASE Report No. 97-65 (1997), http://library-dspace.larc.nasa.gov/dspace/jsp/handle/2002/14650
- Suresh and Huynh. *J. Comput. Phys.* **136**, 83-99 (1997).
- Zalesak. *J. Comput. Phys.* **31**, 335-362 (1979). Improved form of FCT.
- R Cockburn & Shu 2001, J. Sci. Comp. 16, 173 "Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems"
- Zhou, Li and Shu. *J. Sci. Comput.* **16**:2, 145-171 (2001).
- W. Rider, "A Very Brief History of Hydrodynamic Codes", Sandia talk, 2007, https://cfwebprod.sandia.gov/cfdocs/CCIM/docs/Rider_CSRI_June27_2007.pdf
- "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.
- "Review Article: Upwind and High-Resolution Methods for Compressible Flow: From Donor Cell to Residual-Distribution Schemes", Bram van Leer, Commun. Comput. Phys. (2006).

Selected Further References

- Extensive fusion library, fusion history, reactor design studies, etc: http://fire.pppl.gov
- This talk: http://w3.pppl.gov/~hammett/talks
- Center for the Study of Plasma Microturbulence https://web.gat.com/theory/Cspm
- DOE Scientific Discovery Through Advanced Computing http://www.scidac.gov
- GYRO code and movies http://fusion.gat.com/comp/parallel/gyro.html
  & http://w3.pppl.gov/~hammett/talks/2004
- 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/~hammett/papers/
- "ENDING THE ENERGY STALEMATE: A Bipartisan Strategy to Meet America's Energy Challenges", The National Commission on Energy Policy, December 2004. http://www.energycommission.org/
- “Anomalous Transport Scaling in the DIII-D Tokamak Matched by Supercomputer Simulation”, Candy & Waltz, Phys. Rev. Lett. 2003
- “Burning plasma projections using drift-wave transport models and scalings for the H-mode pedestal”, Kinsey et al., Nucl. Fusion 2003
- “Electron Temperature Gradient Turbulence”, Dorland, Jenko et al. Phys. Rev. Lett. 2000
- "Comparisons and Physics Basis of Tokamak Transport Models and Turbulence Simulations", Dimits et al., Phys. Plasmas 2000.