Gammie 11/19/96

COMPUTATIONAL APPROACHES TO STAR FORMATION

I. Introduction and Motivation

In star formation generally, we are trying to find out how interstellar gas turned into the solar system we know today. Some of the big questions:

1. What is the initial mass function?

2. What is the star formation rate?

3. How do binaries form?

4. How do planets form?

5. What is the engine that drives protostellar outflows?

There is a long list of physical processes going into each of these problems, but many of the processes (like hydrodynamics and magnetohydrodynamics, which go into the formation of molecular clouds), share some common attributes:

1. Nonlinearity. [In the hydrodynamic equations, the term $(v \cdot \nabla)v$.]

2. Nonstationary (time-dependent). [e.g. large velocity fluctuations in molecular clouds]

3. Combinations of multiple, equally important effects. [e.g. magnetic fields + gravity + pressure + random motions]

Some are even nonlocal, like radiative transfer! These problems will not be solved by hand in any generality. (There are not even many nonlinear ordinary differential equations that can be solved by hand.)

Occasionally by making drastic assumptions, such as $\partial_t \rightarrow 0$, or self-similarity, one can reduce a particular problem to solving an ordinary differential equation numerically.

So we are driven to numerical simulations.

II. What makes a numerical paper interesting?

A. Accuracy. Discussion of code tests, or reference *in astronomical literature* containing code tests. In particular:

1. linear theory tests.
2. tests on exactly solvable problem.
3. *Convergence* tests on the problem at hand.

But convergence cannot be blindly applied: all relevant scales must be resolved. Example: isothermal collapse calculations in star formation.

In all cases, look for *quantitative* comparison, and *quantitative* controls on errors.

B. Fidelity. Does the approximate problem solved ever correspond to reality in any limit?

C. Relevance. Is there anything that is abstracted from the numerical solution that will ever be used by another researcher? Avoid the implicationless simulation.

A final word of warning: the "I have a dream" speech.

III. An Example: MHD Turbulence in Molecular Clouds.

Gammie & Ostriker 1996 (ApJ 466, 814) studied the dynamics of molecular clouds using a numerical model. Ask: Can nonlinear hydromagnetic waves support clouds against collapse along field lines?

Model is "1 2/3" D; slab-symmetric, but transverse velocities and magnetic fields allowed. Simulation done along the direction of the mean field.

Model contained the following essential features of cloud dynamics:

1. Magnetic fields strong enough that $V_A/c_s \simeq 10$.

2. Random velocities with $\delta V/c_s \simeq 10$.

3. Self-gravity.

These features put "protostellar" turbulence outside regimes that are usually studied, e.g., incompressible, homogeneous turbulence.

The simulation parameters are V_A/c_s, $\delta V/c_s$, and $n_J =$ number of Jeans lengths in simulation.

Can nonlinear hydromagnetic waves support clouds against collapse? Yes.

Now going on to more complicated two and three dimensional models of the ISM.

IV. Numerical Fundamentals: Hydro.

Momentum equations:

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\frac{\nabla p}{\rho}.$$

Mass conservation:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v}).$$

Isothermal equation of state:

$$p = c_s^2 \rho.$$

Can rewrite the momentum equations, using the continuity eqtn., in "conservative form":

$$\partial_t(\rho v_i) = -\partial_j(p\delta_{ij} + \rho v_i v_j).$$

Much of the work in solving this equation goes into dealing with the final term on the right.

Eulerian and *Lagrangian* schemes. Eulerian schemes discretize on spatial grid, while Lagrangian resolution elements follow the fluid. Examples: ZEUS and SPH.

Courant Condition. Stability condition:

$$\Delta t \lesssim \frac{\Delta x}{v_c},$$

where v_c is a characteristic velocity (the sound speed when v_x/c_s is small, and v_x in the opposite limit) and Δx is a characteristic distance (e.g. zone size). Signal cannot propagate across more than 1 zone / timestep. MHD: similar condition, w/ Alfvén speed.

Implication: consider a 3D Eulerian hydro simulation with fixed spatial size over a fixed time interval.

$$\text{CPU time} \sim (\Delta x)^{-4}.$$

Roundoff error and *Truncation error.* Roundoff error is caused by the finite accuracy of numerical representations of real numbers. It is unavoidable.
Truncation error is scheme-dependent. Should be uniform across the grid.

Numerical Dissipation All schemes must also have numerical dissipation. Nonlinear terms in the hydro equations naturally transfer energy to small scales. Cannot allow it to accumulate! Can be handled by *artificial viscosity* or other, more indirect methods.

See Numerical Recipes, 2nd ed., §19.1, for an introduction and references.

V. Numerical Zoology

Most numerical simulation papers relevant to star formation are concerned with just a few pieces of physics: gravity, hydrodynamics, and magnetohydrodynamics.

Here is a resume of some of the codes and algorithms most commonly used in astrophysics.

1. Gravity

Solve

$$\nabla^2 \phi = 4\pi G \rho.$$

This can be done by

Fast Fourier Transform (FFT). See
  Hockney & Eastwood 1982, Computer Simulations Using Particles.
  "O(N log N)", meaning CPU time scales as $N \log N$
  Useful for periodic boundary conditions; other boundary conditions can be more difficult to implement, although there is a simple method for an isolated system.
  Important advantage: existence of highly optimized, canned FFT routines.

Other Basis methods. See, e.g.,
  Weinberg 1996, ApJ, 470, 715.
  Hernquist & Ostriker 1992, ApJ, 386, 375.
  O(N log N)
  Multipole for angular component plus some basis for radial components. Weinberg's method can be optimal for cer-

tain slowly evolving systems, e.g. near-equilibrium galaxy models.

Tree. See, e.g.
Barnes & Hut 1986, Nature, 324, 446.
Barnes & Hut 1989, ApJS, 70, 389.
Hernquist & Katz ApJS 70, 419.
O(N log N)
Works well for approximate systems with a large range in spatial scales; e.g. collapse problems such as galaxy formation.
Subtleties: building the tree, especially on vector or parallel machines, or in periodic geometry.

Fast Multipole Method. See
Greengard & Rokhlin 1987, JCP 73, 325.
O(N)
Essentially a variant of the tree method. Not yet used in astro. Fast in principle, but the overhead is so large that in practice it is not faster except for very large numbers of particles.

Multigrid potential solver (start w/ Numerical Recipes).
O(N)
Fast but complicated. Particularly useful when solution is desired on an irregular grid.

Direct summation.
O(N^2)
Easy to program in its naive form. This is sometimes competitive w/ other methods for specialized computers, e.g.

the GRAPE chip (see J. Makino papers).
In less naive forms this can be extremely complex.
Aarseth's NBODY5, for example, contains special algorithms for accurately evolving close encounters and evolving binary systems, and is designed to study small clusters.
Integration of planetary orbits is also a specialized problem.

Combination, e.g. P^3 M; see the nice review of Bertschinger 1991, in After the First Three Minutes, p. 297.

2. Hydrodynamics

Smoothed particle hydrodynamics. See
J.J. Monaghan, ARAA 30, 543.
Hernquist & Katz ApJS 70, 419.
Invention credited to Lucy, Monaghan, similar idea were floating around in 70s. N-body like method. Pressure gradients on particle i is:

$$\frac{d\mathbf{v}_i}{dt} = \ldots - \sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W(|\mathbf{x}_i - \mathbf{x}_j|, h)$$

W is a *smoothing kernel* and h is a *smoothing length.* "Lagrangian" method: resolution elements follow the fluid.

Problems:
Noisy: not appropriate for problems with nearly uniform density distributions, e.g. linear and mildly nonlinear evolution of modes in disks. Large numerical viscosity.

Poor shock resolution.
Don't believe 1D tests.

Advantages: *Galilean invariant.* Conservation properties. Capable of resolving many orders of magnitude in linear scale. Used to advantage in studies of stellar collisions (e.g. Rasio & Shapiro, Hernquist & Goodman).

Memory-intensive: 1 kbyte/SPH particle.

ZEUS, and similar codes. See
Stone & Norman 1992, ApJS, 80, 791 et seq.
Eulerian.
Operator-split: updates one piece of equation at a time.
Staggered mesh: velocities live on zone faces, while density lives at zone center.
Advection is generally done w/ "Van Leer" method. see Stone & Norman.

This is a widely tested, robust, and reliable method that is now publicly available from the National Center for Supercomputing Applications (NCSA), a supercomputer center at UIUC. See
http://www.ncsa.uiuc.edu/Apps/SoftwareListing.html

This method is second-order, meaning that the errors scale as Δx^2 for linear problems.

Constraints: fixed grid, so cannot resolve large variation in spatial scales. Largest simulations to date have $\simeq 512^3$ zones.

PPM, and similar codes. see

Woodward & Colella 1984 JCP, 54, 174–201.
Eulerian, Operator-split.
Zone-centered: all variables live at middle of zone.
Godunov-type scheme: Solves *Riemann problem* at each zone interface. Riemann problem is the flow away from a hydrodynamic discontinuity, and is exactly solvable.
Variables are representation is piece-wise parabolic.

Several publicly available versions of this code. See
http://www.icemcfd.com/cfd/CFD_codes_p.html
http://fermi.clas.virginia.edu/∼gl8f/VH-1.html.

Excellent resolution of discontinuities.
Convergence rate is more complicated.
Multidimensional, operator-split methods cannot formally do better than second order. See
Porter & Woodward 1994, ApJS, 93, 309.

Largest simulations to date have 1024^3 zones.

Spectral methods.

Operate by decomposing fluid variables into, e.g., Fourier components.

Not many astrophysical applications for these kinds of methods, since they do not handle discontinuities well (Gibbs phenomenon). Useful for studying flows with linear or quasilinear perturbations.

Often used for studies of incompressible flow, where they have excellent convergence properties.

Unstructured mesh.

This is a method often used in engineering applications for studying, e.g., flow around airfoils.

It has been adapted for astronomy by Guohong Xu. See http://xxx.lanl.gov/abs/astro-ph/9610061

Adaptive mesh.

e.g. Klein et al. adaptive mesh code.

Quasi-lagrangian schemes e.g.

Pen, preprint.

Gnedin & Bertschinger 1996, ApJ, 470, 115.

Pen's scheme is publicly available.

3. MHD

Solve hydro equations, with new force in momentum equations

$$\frac{D\mathbf{v}}{Dt} = \ldots - \frac{\nabla B^2}{8\pi\rho} + \frac{(\mathbf{B}\cdot\nabla)\mathbf{B}}{4\pi\rho},$$

and an equation for evolving the magnetic field (the "induction equation"):

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}).$$

Ambipolar diffusion is more difficult. Diffusion equations generally have a restrictive stability condition for explicit schemes:

$$\Delta t \lesssim \frac{\Delta x^2}{\nu}$$

where ν is a diffusion coefficient. Handle implicitly. See MacLow et al., 1995, ApJ 442, 726.

MOC-CT. "Method of characteristics" for evolving the transverse components of the field, together with "constrained transport" for evolving total field. This is the algorithm embodied in the ZEUS code.

MOC means that the transverse components of the field are evolved along characteristics.

CT means that the field is evolved in a clever way that automatically conserves $\nabla \cdot B = 0$. See
Evans & Hawley, 1988, ApJ 332, 659.

Memory requirements: for a 3D isothermal MHD simulation, 7 global arrays of 4 bytes $\times$ $256^3 \simeq 470$ Mbytes.

PPMHD. Extension of PPM scheme to MHD flows.
Ryu & Jones 1995, ApJ 442, 228,
Ryu, Jones, & Frank 1995, ApJ 452, 785,
Dai & Woodward 1994, JCP 115,485.

Difficulties: it is *much* more difficult to solve the MHD Riemann problem than the HD Riemann problem. Also, the solution is degenerate.

It is not clear how to preserve $\nabla \cdot B = 0$ in PPM type scheme, where variables are zone-centered. Various methods are used: flux cleaning, or see-no-evil approach. See
Brackbill & Barnes, 1980, JCP, 35, 426
for a discussion of the issue.

SPMH: based on SPH method, adds in terms for evolving magnetic field. Gammie, Katz.

Phillips and Monaghan 1985, MN 216, 883.

Problems: no way to ensure no monopoles. Noisy. Does not work.

VI. Hardware

National Supercomputing Centers:
  Pittsburgh (C90, T3D). http://www.psc.edu
  NCSA (CO2000, CM-5). http://www.ncsa.uiuc.edu
  San Diego (C90). http://www.sdsc.edu
  Cornell (IBM SP2). http://www.tc.cornell.edu/ctc.html

Vector and *Parallel* machines.

For a crude idea of current machine performance is, look under
Performance Database Server at
http://netlib2.cs.utk.edu/

For example, LINPACK benchmarks:
  C90, 16 proc: 11 Gflops
  T3D, 512 proc: 51 Gflops
  CM-5, 512 proc: 30 Gflops
  DEC Alpha 200MHz: 155 Mflops
  Sun Ultra: 63 Mflops
  Apple PowerMac 8500/132: 22 Mflops
  Apple Mac: 0.004 Mflops

Remember, you *never* get all 16 processors on a C-90, or all 512 nodes on a CM-5. All publicly accessible supercomputers are highly oversubscribed, so turnaround time is long. Also, these are not hydro codes. Real peak performance depends on code, optimization, load.

Allocations are applied for like grants. No money involved.

Small amounts of time, $\simeq$ 10 hr., are easy to come by. Maximum awards in the range of few $\times 10^3$ hr/yr to groups like Grand Challenge cosmology consortium.

Coding for a vector machine is relatively easy. Coding for a parallel machine can be easy or hard, depending on how it is done. Fortran 90, HPF, Message-passing libraries.