Other CFD Methods

Grétar Tryggvason
Spring 2010

Finite Volume Methods are used in the vast majority of CFD codes. However, there are other ways to compute the motion of fluids. Some of the more common alternatives include:

- Spectral Methods
- Finite Element Methods
- Lattice Boltzmann Methods
- Vortex Methods and other particle methods

Here we will briefly examine those.

Consider the initial-boundary value problem for the one dimensional heat equation

\[ \frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2}, \quad 0 < x < \pi, \quad t \geq 0 \]

\[ f(0,t) = f(\pi,t) = 0 \]

\[ f(x,0) = f(x) \]

By standard technique (Fourier sine transform) the solution is

\[ f(x,t) = \sum_{n=1}^{\infty} a_n(t) \sin nx \]

By standard technique

\[ a_n(t) = f_n e^{-n^2 t} \]

\[ f_n = \frac{2}{\pi} \int_0^{\pi} f(x) \sin nx \, dx \]

Assuming that the time integration is exact, the error by using only N terms is

\[ f - f_N = e^{-n^2 t} \]

for all time

This is faster than any power of N. Recall that the error for finite difference/finite volume methods is

\[ O(h^k) = \frac{1}{N^k} \]

Spectral methods are said to have “spectral” or “infinite order” accuracy

This property is not retained if the solution has discontinuities

A spectral approximation to the heat equation is now simply

\[ f(x,t) = \sum_{n=1}^{N} a_n(t) \sin nx \]

Sum to N only!

Substituting into the heat equation we get:

\[ \frac{da_n}{dt} = -n^2 a_n; \quad a_n(0) = f_n \quad (n = 1, 2, \ldots N) \]
For nonlinear equations the use of spectral methods becomes much more involved. 

\[ \frac{\partial f}{\partial t} + f \frac{\partial f}{\partial x} = \nu \frac{\partial^2 f}{\partial x^2} \]

Expand the solution in a Fourier series:

\[ f(x,t) = \sum a_k(t) e^{\frac{-2\pi i}{L} kx} \]

We get the following equation for the coefficients:

\[ \frac{da_k}{dt} + \frac{i \pi L}{2} a_k = -\nu \left( \frac{2\pi}{L} \right)^2 a_k \]

Computing the nonlinear term can become expensive, particularly for three-dimensional systems. By using the Fast Fourier Transform, the computation can be made faster by taking the inverse transform and do the multiplication in real space. Then Fourier transform the product back into spectral space. The multiplication can, however, lead to aliasing problems where the resolved wave numbers are contaminated by unresolved ones.

The fast Fourier transform:

\[ f_j \equiv \hat{f}_l e^{\frac{2\pi i}{N} lj} \]

inverse

\[ f_j = \frac{1}{N} \sum_{l=1}^{N} \hat{f}_l e^{\frac{-2\pi i}{N} lj} \]

Can be evaluated in \( 2N \log_2 N \) operations.

The Cooley-Tukey algorithm

Although spectral methods can have very high accuracy, they can be difficult to apply to nonlinear problems, problems with complex boundary conditions, and complicated geometries.

In addition to Fourier series, several other basis function can be used for other geometries and to control the resolution near boundaries.

Spectral elements methods, where a complex domain is broken into large blocks and each block is treated spectrally can be used treat somewhat complex boundaries.

Finite element methods are similar to spectral methods in that we expand the solution in terms of a known basis function. Unlike spectral methods, where the basis functions are defined globally over the whole computational domain, in the finite element method the basis functions are defined locally on each element.

\[ f(x) = \sum f_i \phi_i(x) \]

\[ \frac{\partial f}{\partial x} = \sum f_i \frac{\partial \phi_i}{\partial x} \]
Example: Elliptic Problem

\[
\frac{\partial}{\partial x} \left( k \frac{\partial f}{\partial x} \right) = q
\]

The residual is:

\[
R = \frac{\partial}{\partial x} \left( k \frac{\partial f}{\partial x} \right) - q
\]

The weighted residual is:

\[
\int_\Omega W(x)R(x)dx = 0 \quad \text{with } W(x): \text{weighting function}
\]

Or:

\[
\int_\Omega W \frac{\partial}{\partial x} \left( k \frac{\partial f}{\partial x} \right) dx = \int_\Omega q W dx
\]

Galerkin Method

Take weighting function equal to the interpolation function

\[
\int_a^b k \frac{\partial f_j}{\partial x} \frac{\partial W}{\partial x} dx = \int_a^b q W dx
\]

For node i:

\[
\sum_{j=1}^{\Delta x} f_j \frac{\partial N_j}{\partial x} \frac{\partial W}{\partial x} dx - \int_{i-1}^{i} q W_i dx = 0
\]

Computational Fluid Dynamics I

Linear basis functions

\[
N_1^{(i)}(x) = \frac{x-x_i}{\Delta x_i}; \quad N_2^{(i)}(x) = \frac{x-x_i}{\Delta x_i}
\]

For element 1:

\[
N_1^{(i)}(x) = x - x_i; \quad N_2^{(i)}(x) = x - x_i
\]

For element 2:

\[
N_1^{(i)}(x) = \frac{x_i - x}{\Delta x_i}; \quad N_2^{(i)}(x) = \frac{x_i - x}{\Delta x_i}
\]

Computational Fluid Dynamics I

For element 1:

\[
N_1^{(i)}(x) = x - x_i; \quad N_2^{(i)}(x) = x - x_i
\]

\[
\frac{\partial N_1^{(i)}}{\partial x} = \frac{1}{\Delta x_i}; \quad \frac{\partial N_2^{(i)}}{\partial x} = \frac{1}{\Delta x_i}
\]

For element 2:

\[
N_1^{(i)}(x) = \frac{x_i - x}{\Delta x_i}; \quad N_2^{(i)}(x) = \frac{x_i - x}{\Delta x_i}
\]

\[
\frac{\partial N_1^{(i)}}{\partial x} = \frac{-1}{\Delta x_i}; \quad \frac{\partial N_2^{(i)}}{\partial x} = \frac{1}{\Delta x_i}
\]

Computational Fluid Dynamics I

\[
\int_a^b W \frac{\partial}{\partial x} \left( k \frac{\partial f}{\partial x} \right) dx = \int_a^b q W dx
\]

Integrate by parts

\[
\int_a^b \left[ \frac{\partial W}{\partial x} \frac{\partial f}{\partial x} \right] dx + \int_a^b \frac{\partial W}{\partial x} \frac{\partial f}{\partial x} dx = \int_a^b q W dx
\]

Evaluating the second term:

\[
\int_a^b \left[ \frac{\partial W}{\partial x} \frac{\partial f}{\partial x} \right] dx + \int_a^b \left[ W \frac{\partial f}{\partial x} \right] dx - \int_a^b W \frac{\partial f}{\partial x} \frac{\partial W}{\partial x} dx = \int_a^b q W dx
\]

Cancel for adjacent elements

Computational Fluid Dynamics I

\[
\sum_{j=1}^{\Delta x} f_j \frac{\partial N_j}{\partial x} \frac{\partial W}{\partial x} dx - \int_{i-1}^{i} q N_i (x) dx = 0
\]

integrating

\[
k_{i+1/2} \frac{f_i - f_{i-1}}{\Delta x^2} - k_{i-1/2} \frac{f_{i+1} - f_i}{\Delta x^2} - \frac{q_{i+1} + 4q_i + q_{i-1}}{6} = 0
\]

where

\[
k_{i+1/2} = \frac{1}{\Delta x} \int_{i}^{i+1} k dx
\]

Notice that a finite difference method would give the same result except with \(q_i\) on the RHS
Finite Element approximation of an unsteady problem:

\[
\frac{\partial f}{\partial t} + U \frac{\partial f}{\partial x} = 0
\]

Weak formulation

\[
\int \frac{\partial f}{\partial t} W \, dx + \int U \frac{\partial f}{\partial x} W \, dx = 0
\]

Introduce the finite element representation

\[
f(x) = \sum_i f_i N_i(x)
\]

The mass matrix makes it complex to use simple explicit time integrators. This can be overcome by "lumping" the matrix.

Lattice Boltzmann Methods

Lattice Boltzmann methods use "particles" on hexagonal grids where the particles move according to discrete rules. It can be shown that the macroscopic motion of the particles resembles the Navier-Stokes equations. It is now believed that these methods can yield results comparable in accuracy to other numerical methods.
**Computational Fluid Dynamics I**
Lattice Boltzmann methods

Update the velocity distributions at each node by:
\[
\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = \frac{1}{\tau} (f_i^{eq}(x,t) - f_i(x,t)), \quad \gamma = 0, 1, \ldots, M
\]
where
\[
f_i^{eq} = \rho_i \left[ 1 + 3(\epsilon_i \cdot \mathbf{u}) + \frac{3}{2} (\mathbf{u} \cdot \mathbf{u}) - \frac{3}{2} (\mathbf{u} \cdot \mathbf{u})^2 \right]
\]
and at each node
\[
\rho = \sum_{i=1}^{M} f_i, \quad \mathbf{u} = \sum_{i=1}^{M} f_i \mathbf{e}_i
\]

**Proof**

Set \( \mathbf{u} = \frac{\rho}{\Delta \mathbf{x}} = \alpha \mathbf{u} \) \( \Delta \mathbf{x} = \Delta \mathbf{y} = \mathbf{e} \)

Replace \( x \) by \( x + \Delta \mathbf{x} \)

\[
f_i(x + \Delta \mathbf{x}, t + \Delta t) - f_i(x, t) = \frac{\gamma}{\tau} (f_i^{eq}(x,t) - f_i(x,t))
\]

taking \( \Delta t = \Delta x = \Delta \mathbf{y} = \mathbf{e} \)

And crossing out the common terms yields
\[
f_i(x + \Delta \mathbf{x}, t + \Delta t) - f_i(x, t) = \frac{\Delta t}{\tau} (f_i^{eq}(x,t) - f_i(x,t))
\]

**Summary:**

Update the velocity distributions at each node by:
\[
f_i(x + \Delta \mathbf{x}, t + \Delta t) - f_i(x, t) = \frac{\Delta t}{\tau} (f_i^{eq}(x,t) - f_i(x,t))
\]
where
\[
f_i^{eq} = \rho_i \left[ 1 + 3(\epsilon_i \cdot \mathbf{u}) + \frac{3}{2} (\mathbf{u} \cdot \mathbf{u}) - \frac{3}{2} (\mathbf{u} \cdot \mathbf{u})^2 \right]
\]
and at each node
\[
\rho = \sum_{i=1}^{M} f_i, \quad \mathbf{u} = \sum_{i=1}^{M} f_i \mathbf{e}_i
\]

Advantages

- Very easy to program—parallelization easy
- Fast in part because incompressibility is not enforced

Disadvantages

- Difficult to implement new physics—must find a discrete equation that corresponds to the new physics
- The flow is not completely incompressible
By now it is understood that LBM are give results comparable to second order finite difference/volume methods.

LBM has been extended to more complex flow, including multiphase flows with sharp interfaces.

The figure shows a comparison of the unsteady rise of a buoyant bubble computed by LBM (left) and front tracking finite volume method (right).


Particle methods cover a large range of physical situations, some overlapping with more conventional methods. Particle methods include:

- Point vortex/vortex blob methods
- Particle-in-Cell codes
- Smoothed particle hydrodynamics
- Dissipative particle methods
- Molecular dynamics

and many more

In many cases a grid is used also, either for efficiency or to account for some of the physics.

Solution to

\[ \nabla^2 \phi = \sigma \]

The solution in a two-dimensional unbounded domain is

\[ \phi(x) = \frac{1}{2\pi} \int_{\sigma} \sigma(x') \ln r da' \]

The solution in a three-dimensional unbounded domain is

\[ \phi(x) = -\frac{1}{4\pi} \int_{\sigma} \frac{\sigma(x')}{r} d\nu, \quad r = |x - x'| \]

\[ \psi(x) = \frac{-1}{2\pi} \int_{\sigma} \sigma(x') \ln r da' \]

Velocity

\[ u(x) = \frac{\partial \psi}{\partial y} = \frac{1}{2\pi} \int_{\sigma} \sigma(x') \left( -\frac{\partial \ln r}{\partial y} \frac{\partial \sigma}{\partial x} + \frac{\partial \ln r}{\partial x} \frac{\partial \sigma}{\partial y} \right) da' \]

\[ u(x) = \frac{-1}{2\pi} \int_{\sigma} \sigma(x') \left( -(y - y')(x - x') \right) d\nu \]

since

\[ \frac{\partial \ln r}{\partial y} = \frac{\partial}{\partial y} \ln \left( \sqrt{(x - x')^2 + (y - y')^2} \right) = \frac{(y - y')}{(x - x')^2 + (y - y')^2}. \]
Euler Equation for two-dimensional inviscid incompressible flow

\[ \frac{d\omega}{dt} = 0 \]

The vorticity of each material particle is constant

\[ \nabla \psi = -\omega \]

\[ u = \nabla \times (\psi k) \]

where \( K \) is the appropriate velocity kernel

The vorticity of each material particle is constant

Vortex Methods

Use point vortices to approximate a smooth distribution of vorticity

\[ u(x) = \frac{1}{2\pi} \int K(x,x')\omega(x')dA' \]

Point vortices in an unbounded domain. The motion of each point is determined by

\[ \frac{dx_i}{dt} = \frac{1}{2\pi} \sum_{j=1}^{N} K(x_i,x_j)\Gamma_j \]

\[ \frac{d}{dt}(x_i,y_i) = \frac{1}{2\pi} \sum_{j=1}^{N} \Gamma_j \frac{(-y_i - y_j)(x_i - x_j)}{(x_i - x_j)^2 + (y_i - y_j)^2 + \delta^2} \]

Generally, the point vorticies are too singular to make a practical numerical method and the point vortices must be regularized. The simplest way is to find the velocity by

\[ \frac{d}{dt}(x_i,y_i) = \frac{1}{2\pi} \sum_{j=1}^{N} \Gamma_j \frac{(-y_i - y_j)(x_i - x_j)}{(x_i - x_j)^2 + (y_i - y_j)^2} \]

Numerical parameter
Small viscosity can be added to vortex methods either by “random walk” or localized averaging.

In three-dimension it is necessary to account also for stretching and tilting of vortex lines, but the basic methodology still works.

The $N^2$ problem: To find the velocity of each vortex, it is necessary to sum over all the other vortices. This leads to large computational times when the number of vortices, $N$, is large.

Remedies:
- Grid based Vortex-In-Cell methods
- Fast summation methods

Vortex-In-Cell
Advect point vortices, but solve
$$\nabla^2 \psi = -\omega$$
to find the velocities.

Fast summation method
A vortex far away from a group of vortices “sees” the group as a single large vortex.

By grouping the vortices together in an intelligent way, it is possible to reduce the operation count significantly.

Example
High Reynolds number density current

Related Methods
- Panel and boundary integral method for flow over solid bodies
- Boundary Integral Methods for free surface flows
- Contour dynamics methods for “patches” of constant vorticity
Other Particle Methods

**Smooth Particle Hydrodynamics**: the fluid mass is lumped into smoothed blobs that are moved using Newton’s second law directly, without an underlying mesh.

**Dissipative Particles Dynamics**: particles represent molecules or fluid blobs but while the method has similarities with SPH, it is generally used to model mesoscale behavior of complex fluids and more liberties are taken in modeling the interactions of the particles.

**Molecular Dynamics**: Not really a CFD technique but can be used to simulate small (really small) fluid regions.

There is no doubt that new solution strategies will continue to be developed. However, incremental advances of current approaches are likely to be the main vehicle for future advances in the computations of complex flows. The finite volume approach currently at the core of CFD has, in particular, proven to be exceedingly robust and versatile.