The Fast multipole method

This was discussed in class and good material for this is available in my PhD thesis from pg 233 – 242. A01

Apart from body-cell tree codes as discussed above the following were also discussed in class

- ~~Bod~~ Cell-Cell tree codes
  - Local expansions
  - Shifting of local expansion
  - Handling of cells of different sizes
- The Adaptive FMM due to Carrier, Greengard & Rokhlin
- Other extensions
  - Dealing with passive particles
  - Anderson's method : FMM without Multipoles
  - Using the FMM to accelerate the panel method calculations.
- Using the tree code or a tree data structure for organizing particles and finding neighbors efficiently.

Helmholtz Laws for vorticity

For an inviscid flow

I No element of fluid which was not originally in rotation is made to rotate

II The elements that are at any time belonging to one vortex line, however they may be translated, remain on one vortex line.

III $\omega\, dA$ for a vortex filament is constant through its whole length and retains its value during all displacements of the filament.

The diff. eqn for vortex motion/vorticity is $\frac{d\vec{\omega}}{dt} = \vec{\omega}\cdot\nabla\bar{V}$

A 02

The first law is easy to see since $\omega = 0$ implies that this will continue to remain so for all time.

Now consider a vortex filament ~~at~~ an element $\delta \vec{r}_i$ of this would be governed by the equation

$$\frac{d}{dt}(\delta \vec{r}_i) = \frac{d}{dt}\left(\frac{\partial \vec{r}_i}{\partial x_i}\right) d x_i = (\delta \vec{r}_i \cdot \nabla \vec{v})$$

Consider $\delta \vec{r}_i - \varepsilon \vec{\omega}$, the evolution of this is

$$\frac{d}{dt}(\delta \vec{r} - \varepsilon \vec{\omega}) = (\delta \vec{r}_i - \varepsilon \vec{\omega}) \cdot \nabla \vec{v}$$

~~This is the same as~~ $\delta \vec{r}_i$ - can be thought of as the displacement along the vortex line. So if $\delta \vec{r}_i - \varepsilon \vec{\omega}$ is initially zero then the vortex line is along the displacement vector initially and from the I law proof we see that $\delta \vec{r}_i - \varepsilon \vec{\omega}$ has to be zero for all time thus the second law is proved.

The third is easily seen as a special case of Kelvin's theorem.

Kelvins theorem is $\frac{d}{dt}\Gamma_{\text{material}} = \frac{d}{dt}\left[\oint_C v_i \, dx_i\right] = \int \nu \, \partial_j \partial_j v_i \, dx_i$

Proof:

$$\Gamma = \oint v_i \, dx_i = \int_{m_1}^{m_2} v_i \frac{\partial x_i}{\partial m} dm \qquad m \text{ - along material curve}$$

$$\frac{d\Gamma}{dt} = \oint_{m_1}^{m_2} \frac{\partial v_i}{\partial t} \frac{\partial x_i}{\partial m} dm + \oint_{m_1}^{m_2} v_i \frac{\partial}{\partial t}\left(\frac{\partial x_i}{\partial m}\right) dm$$

$$= \oint_{m_1}^{m_2} \frac{\partial v_i}{\partial t} dx_i + \underbrace{\oint_{m_1}^{m_2} v_i \, dv_i}_{}$$

$$\text{''} \oint_{m_1}^{m_2} \frac{1}{2} d(v_i v_i) = 0.$$

$$= \oint_{m_1}^{m_2} \frac{\partial v_i}{\partial t} dx_i = \oint \frac{D v_i}{D t} dx_i = \oint \left[-\partial_i\left(\frac{p}{\rho}\right) + \nu \partial_j \partial_j v_i\right] dx_i$$

$$= \oint \nu \, \partial_j \partial_j v_i \, dx_i$$

Using Kelvins theorem we see that for an inviscid fluid, $\frac{D\Gamma}{Dt} = 0$ for a material region. A03

This along with Helmholtz second law and the fact that $\mathrm{div}(\underset{\sim}{\omega}) = 0$ clearly let us establish Helmholtz third law.

Some odds and ends. (or not quite)

Conserved quantities

2D: In the case where there are no boundaries (both viscous inviscid)

$$\Omega_2 = \int_{R^2} \omega \, d\underset{\sim}{x} \qquad \text{circulation}$$

Linear impulse : $I_x = \int_{R^2} x\,\omega \, dx\,dy$; $I_y = \int_{R^2} y\,\omega\, dx\,dy$

For inviscid fluids $M_2 = \int_{R^2} r^2 \omega \, dx\,dy$

3D: No boundaries

$$\Omega_3 = \iiint \underset{\sim}{\omega} \, dx\,dy\,dz$$

$$\text{Helicity} = H_3 = \iiint_{\mathbb{R}^3} v \cdot \omega \, dx\,dy\,dz$$

Linear impulse $I_3 = \iiint_{\mathbb{R}^3} \underset{\sim}{x} \times \underset{\sim}{\omega} \, dx\,dy\,dz$

Inviscid case

$$M_3 = \iiint_{\mathbb{R}^3} \underset{\sim}{x} \times (\underset{\sim}{x} \times \underset{\sim}{\omega}) \, dx\,dy\,dz.$$

These are useful quantities to calculate and estimate accuracy of a simulation.

Blobs of different core radii:

If you have two blobs of core radius δ_i, δ_j respectively, then if you perform computations with these blobs the conserved quantities will not be preserved unless one uses a common δ for mutual interactions.

For example when computing using K_δ do not use K_{δ_i} & K_{δ_j} instead use K_δ with $\delta = \sqrt{\left(\frac{\delta_i^2+\delta_j^2}{2}\right)}$. When this is done the conserved quantities will remain conserved.

3D vortex methods: a very brief introduction.

The nature of the equations change from 2D to 3D. We have

$$\frac{D\underset{\sim}{\omega}}{Dt} = \underset{\sim}{\omega} \cdot \text{grad}(\underset{\sim}{V}) \quad ; \quad \underset{\sim}{\omega} = \text{curl}(\underset{\sim}{V})$$

The addition of the stretching term is important to note and changes the nature of the equations.

Fortunately we may use similar ideas that we used for 2D in 3D. For example it can be shown that we may use a "vector potential" $\underset{\sim}{\psi}$ such that

$$\nabla^2 \underset{\sim}{\psi} = -\underset{\sim}{\omega} \quad \text{and} \quad \underset{\sim}{V} = \text{curl}(\underset{\sim}{\psi})$$

If this is given we may use the same approach as in 2D.

Consider the Greens function for the ∇^2 operator in 3D

$$G = \frac{1}{4\pi r}$$

Thus $\underset{\sim}{V} = \text{curl}(\underset{\sim}{\psi}) = \text{curl}(G * \omega) = K * \omega$

Thus we can compute

$$\underset{\sim}{V}(\underset{\sim}{x}, t) = \iiint K(\underset{\sim}{x} - \underset{\sim}{x}') \, \underset{\sim}{\omega}(\underset{\sim}{x}', t) \, d\underset{\sim}{x}'$$

K is given by $K = -\frac{1}{4\pi r^3} \begin{pmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{pmatrix}$

This is singular and may be de-singularized using the same approach by convolving this with a smoothing kernel f_δ. Once this is done we may compute V as

$$V(\underset{\sim}{x}, t) = \sum_j K_\delta(\underset{\sim}{x} - \underset{\sim}{x}_j) \, \underset{\sim}{\omega}_j \, h^3.$$

Using these equations we may advect the vorticity with the local velocity. However, as per $\frac{D\underset{\sim}{\omega}}{Dt} = \underset{\sim}{\omega} \cdot \text{grad} \, \underset{\sim}{V}$, clearly the $\underset{\sim}{\omega}$ of each particle must change over time. This is done by

$$\frac{d\underset{\sim}{\omega}_i}{dt} = (\underset{\sim}{\omega}_i \cdot \nabla_x)(\underset{\sim}{V}(x_i, t))$$

where the gradient of the velocity is computed as

$$\nabla_x(\underset{\sim}{V}(x, t)) = \sum \nabla_x K_\delta(\underset{\sim}{x} - \underset{\sim}{x}_j) \, \omega_j \, h^3$$

By taking the dot product of this with ω_i we may solve the ODE numerically to find the change in the individual blob's vorticity. In this manner we may simulate a 3D flow. In 3D though the computations become more expensive. The FMM may be implemented but typically requires the use of more complicated mathematics, spherical harmonics.
One may use vortex filaments also for such simulation.
For more details see Leonard, Vortex methods for flow simulation. JCP, 37, 289-335 (1980)

Other good reviews are

G. S. Winckelmans & A. Leonard, "Contribution to vortex particle methods for the computation of 3D, incompressible unsteady flows", JCP 109, 247-273 (1993).

Cottet & Koumoutsakos also will have more recent details on this.

We stop this very brief review here and proceed to look at handling viscous fluids in 2D (and perhaps 3D).

Viscous vortex methods

$$\frac{D\omega}{Dt} = \nu \nabla^2 \omega \quad , \ \nu \text{ is constant in 2D.}$$

We need to solve this

- In a Lagrangian framework.
- Apply Boundary condition accurately.

In vortex methods the general approach to solving this is to do viscous / operator splitting. At each time step we first solve the Euler equation & then the Heat equation. i.e.

$$\frac{D\omega}{Dt} = 0 \longrightarrow \text{advection}$$

$$\frac{\partial \omega}{\partial t} = \nu \nabla^2 \omega \longrightarrow \text{diffusion}$$

Clearly this introduces an error. Beale & Majda (1981, Math. Comp., v. 37, No. 156, pp 243-259) prove that the error in this is $O(\nu \Delta t)$ where Δt is the timestep. This may be easily improved to $O(\nu \Delta t^2)$ if Strang-type splitting is employed.

To see this we look at the Linear convection diffusion equation.

We follow the derivation in Cottet & Koumoutsakos (2000).

Consider, $\dfrac{\partial W}{\partial t} + \underset{\sim}{c} \cdot \text{grad}(W) = \nu \nabla^2 W$ $\qquad$ A0-1

Here 'C' is a known ~~constant~~ function but not a function of W
Since this equation is linear we may easily solve it.

Let $C \cdot \text{grad}(\ ) = -A$ (an operator)

& $\nu \nabla^2 (\ ) = B$ (another operator)

Note that if A & B were discretized (along with W) we may write them as Matrices.

Thus we have $\dfrac{\partial W}{\partial t} = AW + BW$

$$\Rightarrow \boxed{W = W_0 \, e^{(A+B)t}}$$

Now consider the time stepping process we write $W(n \Delta t) = W^n$.

$\therefore \quad W^{n+1} = e^{(A+B)\Delta t} W^n$

This is the exact solution given W^n.

With operator splitting we get:

$$W^{n+1/2} = e^{A \Delta t} W^n$$

$$W^{n+1} = e^{B \Delta t} W^{n+1/2} = e^{B\Delta t} e^{A \Delta t} W^n$$

Clearly the error arises because $e^{(A+B)\Delta t} \neq e^{B\Delta t} e^{A \Delta t}$ in the general case when $AB \neq BA$. ~~this may happen in a~~

To see this, Taylor expand each.

$$e^{B\Delta t} e^{A \Delta t} = \left(1 + \Delta t\, B + \frac{\Delta t^2}{2} B^2 + \ldots\right) \left(1 + \Delta t A + \frac{\Delta t^2}{2} A^2 + \ldots\right)$$

$$e^{(A+B)\Delta t} = \left(1 + (A+B)\Delta t + \frac{\Delta t^2}{2} (A+B)(A+B) + \ldots\right)$$

These are equal only when $AB = BA$. It can also be easily seen that the error in the above is $O(\nu \Delta t^2)$. Therefore the global error over several time steps $\nu \Delta t)$.

AOS

In Strang type splitting we instead do

$$W^{n+1} = \left(e^{B\Delta t/2}\, e^{A\Delta t}\, e^{B\Delta t/2}\right) W^n$$

This can be easily shown to generate an $O(\nu \Delta t^3)$ local and $O(\nu \Delta t^2)$ global error.

The Strang type splitting basically implies that the heat equation is first solved for a $\Delta t/2$ time step. The Euler equation then solved for a full Δt. The Heat ~~solution~~ equation is then solved for a $\Delta t/2$.

The above analysis was for a simpler equation. The results still hold though for the NS eqns. See Beale & Majda or Cottet & Koumoutsakos for details.

In this manner we may solve the NS equations using Lagrangian schemes.

In order to solve the heat equation ~~is a Lagrangia~~ there are several approaches.

- Traditional grid based schemes – we don't want this
- Grid free schemes
- "Some what mesh dependent" schemes.

We will discuss 4 schemes.

- RVM · Random Vortex Method (mesh free & but stochastic)
- Core spreading method (CSM) also mesh-free
- PSE – Particle Strength Exchange, requires remeshing.
- VRT – Vorticity Redistribution Technique, Mesh free.

RVM

The Random Vortex Method is probably the oldest of the vortex diffusion schemes. It was introduced first by A.J. Chorin in 1973 (JFM, 57, 785-796). A09

To see how it works consider the general solution of

$$\frac{\partial w}{\partial t} = \nu \nabla^2 w$$

The Greens function for this is $G(\vec{r},t) = \frac{1}{4\pi\nu t} e^{-\left(\frac{r^2}{4\nu t}\right)}$.

Therefore the solution is

$$w(\vec{r},t) = \frac{1}{4\pi\nu t} \iint_{\mathbb{R}^2} e^{-\frac{(\bar{r}-\bar{r}')^2}{4\nu t}} w(\vec{r}')\, dx'\, dy'.$$

Now write $w(\vec{r}') = \sum_{j=0}^{N} f_\delta(\vec{r}-(\vec{r}_j+\vec{r}'))\, \Gamma_j$

$\therefore$ we have by changing the order of the ~~kernel &~~ convolution

$$w(\vec{r},t) = \frac{1}{4\pi\nu t} \iint_{\mathbb{R}^2} e^{-\frac{r'^2}{4\nu t}} \sum_{j=0}^{N} f_\delta(\vec{r}-(\vec{r}_j+\vec{r}'))\,\Gamma_j\, dx'dy'$$

Clearly this integral may be evaluated using a Monte-Carlo approach. That is, the vorticity w is the expected value of the integral taken over Gaussian random variables $\vec{r}'$ with 0 mean and variance $2\nu t$.

Thus at each time step Gaussian random numbers (or random numbers distributed as per a Gaussian) are generated with zero mean & variance $2\nu\Delta t$. These displacements are added to the vortices.

Note that Gaussian deviates may be generated from uniform deviates using the Box Muller method from:

$$y_1 = \sqrt{-2\ln(x_1)}\cos(2\pi x_2) \;;\; y_2 = \sqrt{-2\ln x_1}\,\sin(2\pi x_2)$$

Implementation of the BC

To satisfy the no-slip B.C we introduce vortex blobs or sheets just above the surface of the body such that the no-slip BC is satisfied. This can be implemented in various ways.

* Find the slip, assume that locally a vortex induces only a local velocity. Consider the image of the vortex just inside the body and set Γ_{blob} such that the slip velocity is zero.
* Solve a system of equations for each blob such that the strengths are found to exactly cancel the slip.

Other researchers suggest a different approach to solve the BC. but we shall not discuss that currently.

The algorithm to be used for a vortex method used to simulate the incompressible N.S equations can now be given as follows.

Algorithm

1. Compute the slip velocity on control points
2. Release vorticity suitably to satisfy the no-slip BC.
3. Convect existing blobs and sheets (not the newly created ones)
4. Diffuse all the particles using a suitable scheme.
5. Back to step 1.

Thus by using such a scheme we may simulate an incompressible 2D, viscous problem.

We next discuss some of the other diffusion schemes for vortex methods.

Core Spreading Model (CSM)

Consider a smoothing function given as

$$f_\delta(|x - x_i|, t) = \frac{1}{2\pi\sigma(t)} e^{-|x-x_i|^2/2\sigma^2}$$

The equation $\frac{\partial w}{\partial t} = \nu \nabla^2 w$ is satisfied when

$$\frac{d\sigma^2}{dt} = 2\nu$$

Thus if we change the smoothing radius such that

$$\sigma^2(t+\Delta t) = \sigma^2(t) + 2\nu\Delta t$$

will solve the heat equation exactly if the blob is a Gaussian. The trouble is that σ increases indefinetely and ends up solving the incorrect equation (or there is a huge error). The original method was first proposed by Leonard in 1980 (JCP, v.37, pp 289-335). In 1996 & 1997 Rossi proposed a modification to split the blobs after the reach a certain size and this eliminates the problem. See Rossi, SISC v 17(2), 370-397.

For more details on how best to simulate Problems with the core spreading method see the PhD. thesis of D. Shiels (1998, CalTech). The core spreading model is naturally grid free.

The Particle Strength Exchange Method (PSE)

The method was first proposed by Degond & Mas-Gallic in 1989 (Math. Comp. V188, p 485). The idea behind the method is to replace the ∇^2 operator by an integral operator.

$$\nabla^2 w \approx \iint w(\underset{\sim}{y}) - w(\underset{\sim}{x})\, K(\underset{\sim}{y}, \underset{\sim}{x})\, d\underset{\sim}{y}$$

where K is an integral operator such that

$$\iint \prod_{i=1}^{n} (\underset{\sim}{y} - \underset{\sim}{x})_i \, K(|\underset{\sim}{y} - \underset{\sim}{x}|) \, d\underset{\sim}{y} = 0 \qquad 1 \le n \le r+1$$

and

$$\iint (\underset{\sim}{y} - \underset{\sim}{x})_i (\underset{\sim}{y} - \underset{\sim}{x})_j \, K(|\underset{\sim}{y} - \underset{\sim}{x}|) \, d\underset{\sim}{y} = 2\delta_{ij}$$

To see how this is possible note that if we expand

$$\omega(\underset{\sim}{y}) = \omega(\underset{\sim}{x}) + \left[(\underset{\sim}{y} - \underset{\sim}{x}) \cdot \nabla\right] \omega(\underset{\sim}{x})$$
$$+ \frac{1}{2}\left[\left((\underset{\sim}{y} - \underset{\sim}{x}) \cdot \nabla\right)\left((\underset{\sim}{y} - \underset{\sim}{x}) \cdot \nabla\right)\right] \omega + \dots$$

The error in this is $O(\varepsilon^r)$ where ε is the "core radius" of the kernel K (or a length scale of K).

A detailed error analysis shows that the error is $O\left(\varepsilon^r + \frac{h^k}{\varepsilon^{k+1}}\right)$ where h is particle spacing and k is a measure of the smoothness of K. This clearly shows that some amount of particle overlap is necessary for good accuracy.

The method therefore requires periodic remeshing. This is one disadvantage of the method.

Note that one can use this approach to evaluate the different derivatives of a function by choosing a suitable kernel.

In general in implementations a Gaussian kernel is used. This gives second order accuracy.

More details on the method are available in Cottet & Koumoutsakos book.

The Vorticity Redistribution Technique (VRT)

This method was first proposed by Shankar & Van Dommelen in 1996 (JCP, 127, pp 88-109).

Let

$$\omega^n = \sum_i \Gamma_i^n f_\delta(\underset{\sim}{x} - \underset{\sim}{x}_i) \quad ; \quad n \text{ - denotes the time level.}$$

Solving the diffusion equation reduces to solving for some f_{ij}'s such that

$$\omega^{n+1} = \sum_i \sum_j f_{ij}^n \Gamma_i^n f_\delta(\underset{\sim}{x} - \underset{\sim}{x}_j)$$

by comparing the Fourier Transform of this & comparing with the exact solution we can establish condition on f_{ij} to solve the heat equation to any order of desired accuracy.

We introduce $\quad \underset{\sim}{\xi}_{ij} = (\underset{\sim}{x}_j - \underset{\sim}{x}_i)/h_\nu \quad ; \quad h_\nu = \sqrt{\nu \Delta t}$.

Then the following equations are solved.

$O(1) \quad : \quad \sum_j f_{ij}^n = 1$

$O(\Delta t^{1/2}) : \quad \sum_j f_{ij}^n \xi_{1\,ij} = 0 \quad ; \quad \sum_j f_{ij}^n \xi_{2\,ij} = 0$

$O(\Delta t) : \quad \sum_j f_{ij}^n \xi_{1\,ij}^2 = 2 \; ; \; \sum_j f_{ij}^n \xi_{1\,ij} \xi_{2\,ij} = 0 \; ; \; \sum_j f_{ij}^n \xi_{2\,ij}^2 = 2$

and so on.

In order to solve this given each particle, we ~~identify~~ create a ball of radius $R h_\nu \approx \sqrt{12 \nu \Delta t}$ inside which we find neighbouring particles and solve a minimization problem to satisfy the above equations. The additional condition $f_{ij} \geq 0$ is also applied to keep $\sum |\Gamma_i^n|$ bounded. If no solution is possible, new particles with zero strength are added at suitable locations. The method is completely grid free but computationally quite intense.