We now look at <u>Errors in vortex methods</u>.

An outline of various topics: ADI

* Errors in 2D vortex methods
* The flow map & its appearance in the quadrature
* Components of the error: Consistency (discretization + smoothing) + Stability

Strategies to overcome problems

* Remeshing / rezoning
* Adaptive quadrature - Free Lagrange methods.

Before we get there we note common problems solved in the literature.

- The vortex sheet roll-up problem discussed earlier is one. Hoeijmakers & ~~Vaastra~~ Vaastra AIAA 1983 simulates this with a 2nd order panel method.
- The vortex merging problem. Consider two co-rotating vortex patches and study their evolution. Christiansen (1973, JCP, vol 13 no. 3) was one of the first to simulate these to good effect.
- The method of contour dynamics by Zabusky et. al (1979, JCP, v30 no. 1, pp. 96-106) is one such seminal approach to solving the merging problem. The idea is that one tracks the contour separating the region of vorticity from the rest of the fluid. Produces very high res. solutions
- The Perlman test cases. Perlman in 1985 introduced a ~~test~~ set of test problems with a known solution against which the vortex method simulation could be compared. These vortex patches were radially symmetric. One example is

$$\omega(z) = \begin{cases} (1-|z|^2)^7 & \|z\| \le 1 \\ 0 & \|z\| > 1 \end{cases}$$

This is radially symmetric & has an exact solution. Using this and other test problems he studied the higher order accuracy of vortex methods. M. Perlman (1985, JCP, vol 59 p.p. 200-223)

Rough notes for AE 625

Given this background let us consider the errors A02
in vortex method simulation.

First we must elaborate one detail we did not mention earlier.

As seen before the motion of the vortices describes the evolution of the vorticity consider a function or a mapping ~~ϕ~~ X such that $z \longrightarrow X(z,t)$ i.e. if a particle is initially at z then its position at time 't' is given as $X(z,t)$, this is called the flow map such that

$$\dot{X}(z,t) = u(X(z,t),t) \quad ; \quad X(z,0) = z.$$

Clearly vorticity is conserved along particle paths. i.e.

$$\omega(X(z,t),t) = \omega(z,0).$$

To find u from ω we have

$$u(z,t) = \int K(z-z')\,\omega(z',t)\,dx'\,dy'$$

Using the flow map we can write this as.

$$\dot{X}(z,t) = \int K(X(z,t)-z')\,\omega(z',t)\,dx'\,dy'$$

Clearly, we can ~~take~~ see that for our Lagrangian simulation, $z' = X(z',t)$. Since the flow is incompressible the Jacobian of X is $\underline{1}$. Thus we may write

$$\dot{X}(z,t) = \int K(X(z,t) - X(z',t))\,\omega(z',0)\,dx'\,dy'.$$

Thus for the quadrature we require ω only at the initial points and continue to perform the quadrature using the values of $X(z',t)$.

Note that these points may be in a complete disarray with no ordered distribution.

Now given this we may start discussing the details of the ~~two~~ errors involved in vortex methods.

Let f_δ be a smoothing function s.t. $f_\delta(z) = \frac{1}{\delta^2} f\left(\frac{|z|}{\delta}\right)$.

Rough notes for AE 625

For higher order convergence of the method we must have the following conditions satisfied on f. A03

(i) $\int f(z)\,dx\,dy = 1$

(ii) $\int x^\alpha y^\beta f(z) = 0 \quad 1 \le \alpha+\beta \le m-1$

(iii) $\int |z|^m |f(z)| < \infty$

& $f \in C^L$ & $f(z) = 0$ for $z \ge 1$; This compact support may be weakened.

$w \in C^M$ has compact support (or w dies to zero at ∞ sufficiently fast)

Given this we have the following theorem.

Theorem: Given $L \ge 3$; $M \ge \max(L+1, m+2)$ and $m \ge 4$. Let $\delta = c h^a$ where $0 < a < 1$. Suppose h is large enough so

$$L > (m-1)a / 1-a$$

Then the computed $X_{h,\delta}$ satisfies

$$\| X - X_{h,\delta} \| \le O(h^{ma})$$ as h & δ go to zero.

where the discrete norm $\|g\|_h = \left(h^2 \sum_i |g(z_i)|^2 \right)^{1/2}$.

There are similar bounds for the velocity & vorticity fields.

The theorem lets us pick $a \simeq 1$ & $\delta = O(h)$ only for smooth flows.

To see how this comes about consider the error in the velocity

error $= \| V(z, t) - \hat{V}_{h\delta}(\tilde{z}, t) \|$

where $V(z,t)$ is the exact velocity & $\hat{V}_{h\delta}(\tilde{z},t)$ is the computed one.

Consider what is happening. At $t=0$ we have some particle positions z_i. These particles are moved as per the velocity field. The integral of the vorticity with the velocity kernel over these points produces the computed velocity field. Let $z_i(t)$ be the exact positions of the particles at time t and let $\tilde{z}_i(t)$ be the computed values.

Rough notes for AE 625

The error in the velocity field is given as

$$\text{error} = \| \underbrace{V(z,t)}_{\text{Exact}} - \underbrace{\hat{V}(z,t)}_{\text{computed velocity field.}} \|$$

Let $V_h(z,t)$ be the velocity field computed due to the blobs at the exact positions $z_i(t)$. That is, the initial distribution of vorticity is discretized into several particles carrying the vorticity. If the positions of those vortices $z_i(t)$ at time t were known exactly, what would be the ~~exact~~ computed velocity due to those? This would be termed $V_h(z,t)$. Thus from the triangle inequality we have

$$\text{error} = \| V(z,t) - \hat{V}(z,t) \| \leq \| V(z,t) - V_h(z,t) \| + \| V_h(z,t) - \hat{V}(z,t) \| .$$

~~Expanding these first of~~ $\| V(z,t) - V_h(z,t) \|$ is called the <u>consistency error</u> and $\| V_h(z,t) - \hat{V}(z,t) \|$ is called the <u>stability error</u>.

Clearly we must have that the consistency & stability errors should go $\to 0$ as h & $\delta \to 0$.

It turns out that the ~~stability err~~ consistency error can be further split into two parts, the smoothing & discretization error. To see this we expand the consistency error.

$$\| V(z,t) - V_h(z,t) \| = \left\| \int K(z-z') \, \omega(z') \, dx' dy' - \sum_i K_\delta(z - z_i^t) \, \Gamma_i \right\| .$$

$$\leq \underbrace{\left\| \int K(z-z') \, \omega(z') \, dx' dy' - \int K_\delta(z-z') \, \omega(z') \, dx' dy' \right\|}_{\text{smoothing error}}$$

$$+ \underbrace{\left\| \int K_\delta(z-z') \, \omega(z') \, dx' dy' - \sum_i K_\delta(z - z_i) \, \omega_i \, h^2 \right\|}_{\text{discretization error.}}$$

Clearly, the smoothing error is the difference introduced due to the introduction of the smoothing. The discretization error represents the error due to quadrature of the smoothed integral by the use of the trapezoid rule.

Rough notes for AE 625

For a smoothing function satisfying the properties mentioned in pg A03 it may be shown that A05

the smoothing error $\leq C_m \delta^m$.

That is, the smoothing function is said to be having order 'm'.

To outline a proof we see that

e_δ = smoothing error = $| K_\delta * w - K * w |$

$= |(K_\delta - K) * w| = | \int (K_\delta - K)(z - z') \, w(z') \, dx' dy' |$

$K_\delta = K * f_\delta$

$\therefore \; e_\delta = | \int\int [K * (f_\delta - \delta)](z - z') \; w(z') \, dx' dy' |$

We may crudely argue that this quantity is mainly controlled by the error in $\int\int (f_\delta - \delta) \, dx' dy'$

By the conditions on f_δ we have that the first ~~moment~~ $(m-1)$ moments of f_δ are zero under the integral. Similarly $\int\int f_\delta = 1$. + $f_\delta(z) = 0 \; \|z\| > 1$

Thus expanding f_δ in a Taylor series about zero we see that

$f_\delta = f_\delta(0) + f_\delta'(0) \, r + \ldots$

We can intuit that since f_δ is an approximate δ function and satisfies $(m-1)$ moments going to zero that the remainder term in the Taylor series will be $O(\delta^m)$. If the w is sufficiently smooth an its integral bounded we can see that

$$e_\delta \leq C \, O(\delta^m).$$

This is no proof. A formal proof is given in C. Anderson and C. Greengard's 1985 paper "On Vortex methods", SIAM Journal on Numerical Analysis, Vol 22. no. 3, pp 413 - 440.

We reproduce the following from there.

From Parseval's theorem we may find e_δ by transforming to the Fourier domain. Thus $(\overline{K_\delta * w} - \overline{K * w})$ where $\bar{\ }$ denotes Fourier Transform ~~...~~ must be found.

$\therefore \; e_\delta = | \hat{g}(t) | = | \overline{(K_\delta * w(t))}(\xi) - \overline{(K * w(t))}(\xi) |$

$= (\overline{K}_\delta(\xi) - \overline{K}(\xi)) \, \overline{w}(\xi, t) \, |$ [Convolution thm.]

$$= \left| \bar{K}(\xi) \left(\bar{f}_\delta(\xi) - 1 \right) \bar{\omega}(\xi) \right|$$

$$= \left| \bar{K}(\xi) \bar{\omega}(\xi) \left(\bar{f}_\delta(\xi) - \bar{f}(0) \right) \right|$$ ← condition on $\int f_\delta \, dx = 1$.

$$= \left| \bar{K}(\xi) \bar{\omega}(\xi) \left(\bar{f}(\delta \xi) - \bar{f}(0) \right) \right|$$

↑ Scaling theorem (division in real space becomes multiplication in F. space).

Now expanding $(\bar{f}(\delta\xi) - \bar{f}(0))$ about 0 in a Taylor series we see that since moment conditions are satisfied that the derivatives of $\bar{f}(0)$ ~~at~~ upto $m-1$ order are zero. The m'th term is bounded by the remainder which is $O(\delta^m)$.

Now by establishing bounds on $\bar{K}(\xi) = \frac{C}{|\xi|}$ and $\bar{w}(\xi)$ based on its smoothness we can show that

$$\boxed{e_s \leq C_s \delta^m}$$

For more details please refer the proof in the paper by Anderson & Greengard. With a much more involved proof they also show that the discretization error $e_d \leq C_d \left(\frac{h}{\delta}\right)^L \delta$ when $\delta \leq 1/2$ and constant C_d.

Clearly if f is infinitely differentiable $L \to \infty$ & $e_d \to 0$. However ~~we~~ this is true iff $h < \delta$ i.e. we may choose $\delta = C h^a$ $0 < a < 1$

Thus error $\leq e_s + e_d \leq C_s \delta^m \leq C h^{ma}$.

This clearly shows why we need the vortex particles to overlap such that $\delta > h$.

Numerical experiments by Perlman (JCP, 1985 vol 59, pp 200-223) show that it is often not sufficient to choose δ close to h but one must choose δ larger than h. Some suggest that one must use ~~h~~ $\approx \delta = \sqrt{h}$. This implies a very significant amount of overlap between particles.

Rough notes for AE 625

We now look at strategies used to overcome these accuracy problems. There are primarily two strategies to overcome the error. The major problem is the discretization error. A07

Since the quadrature nodes become disorganized the quadrature rule becomes increasingly inaccurate in time. This fact was brought out by the numerical results of Perlman. Thus if we require higher order accuracy with not too large a core-radius we may do one of two things.

* Employ some kind of "regridding" or "re-meshing".
* Improve the quadrature rule by changing the weights of the rule.

Note that using a large δ implies a larger smoothing error. It also reduces the efficiency of the computational method when a Fast Multipole method (FMM) is used. We shall see this later.

Remeshing

Beale & Majda first proposed this in their 1985 JCP paper. What they did was to periodically interpolate the vorticity back onto a regular mesh. Then they would form new particles from the grid and continue the computation. They called this "rezoning".

In 1991 H.O. Nordmark (JCP, 97, 366-397) proposed two algorithms for rezoning. He compared his results with the exact solution and showed that long time higher order accuracy is possible with this strategy. However the remeshing procedure does introduce an error of its own. Thus remeshing is not performed at every timestep but only after some error condition is satisfied. This paper looks at comparison with the Euler equation.

In 1996 Nordmark (JCP, 129, 41-56) performed a similar numerical experiment using the N.S. equations in 2D. Once again he shows that occasional rezoning produces much more long time accuracy than without. Similarly P. Koumoutsakos computed the inviscid axisymmetrization of an Elliptical vortex patch using a vortex method.

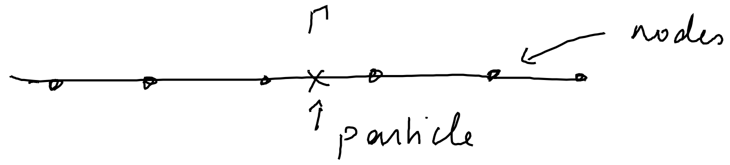
Rough notes for AE 625

He also employs periodic remeshing. Various quantities are studied in order to measure the effect of such remeshing. (JCP, 138, 821-857) 1997.

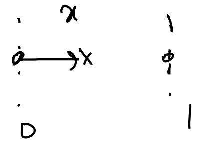
Subsequently this approach has been used by other authors. The idea involved is to consider a fixed grid and interpolate the vorticity onto the grid. The nodes of the grid then define new computational particles. Care is taken to the higher moments of the vorticity are preserved. For example Koumoutsakos consider up to 4^{th} order of moments. The approach introduces some error & viscosity (hyper-viscosity) but greatly reduces discretization error so makes up for this loss.

In subsequent work Ploumhans & Winckelmans (JCP, 2000, V165) propose an algorithm to perform remeshing in the presence of arbitrarily complex geometries.

In Appendix B of my PhD thesis (pp 269 - 277) I discuss a slightly modified version of this algorithm in considerable detail. I describe the overall idea below. Consider a 1D case as seen in the figure. Lets say we wish to interpolate the particle vorticity to the given grid. We do this as follows. Consider the following

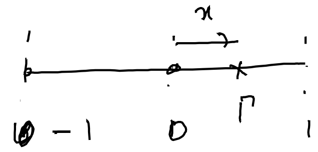


$$\Lambda_1 = \begin{cases} 1 - x & \text{to } 0 \\ x & \text{to } 1 \end{cases}$$



If we have a particle at x. We distribute a value of $(1-x)\Gamma$ to the point 0 and $x\Gamma$ to that at 1. This is a first order interpolation. Similarly we may do this in a second order interpolation as

$$\Lambda_2 = \begin{cases} x(x-1)/2 & \text{to } -1 \\ (1-x^2) & \text{to } 0 \\ x(1+x)/2 & \text{to } 1 \end{cases}$$

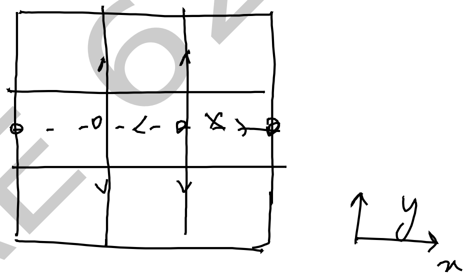


We may do this for higher orders also and for the case of boundaries

Rough notes for AE 625

We may define "de centered" Interpolation Kernels
See either the papers or my thesis for details.
For the use of each type of interpolation one A09
may compute a "penalty" and thereby decide upon the best interpolation
by minimizing the penalty.

In 2D this may be done as follows.
First interpolate along x axis to points
on the x = const lines of the grid.
Then interpolate those along the y = const
grid points.



This is called a Cartesian product approach.
Clearly this process can be used for rezoning.

Modifying the Quadrature

John Strain in his 1997 paper (JCP 132, ~~108-122~~ pp 108-122)
proposes a different approach. In this, the idea
is to change the weights of the quadrature rule such that high
accuracy is obtained. The flow map is not used and therefore
the difficulty is avoided. This approach leads to what is known
as a "Free-Lagrangian method".

The idea is to calculate

$$u(z_i, t) = \int K(z_i - z') \, \omega(z', t) \, dx' dy'$$

$$\approx \sum_{j=1}^{N} w_{ij}(t) \, K(z_i - z_j) \, \omega(z_j, t)$$

the idea is that one computes new weights w_{ij} at each time step to
obtain higher order accuracy for long times. In his paper Strain
presents a fast and adaptive way of doing this.

Rough notes for AE 625

A10

The idea of re doing the quadrature goes back even further and in 1988 Beale suggested in his paper (On the accuracy of vortex methods at large times) that this could be done. He proposed a simple scheme to solve for the following. At some time t let $z_i(t)$ be the particle positions. He creates a square grid and interpolates the vorticity using $w_i \approx \sum_j f_\delta(z_i - z_j)\, \gamma_j h^2$ and solves for the γ_j's given the known values of w_i.

What Strain does is to construct new quadrature rules based on the given nodes $z_i(t)$ at each time. To do this efficiently, he constructs a tree structure splitting the domain into rectangles containing p or $p+1$ particles. Suppose we desire q^{th} order accurate quadratures, then in 2D there are $m = q(q+1)/2$ monomials of the form $x^\alpha y^\beta$ $\alpha + \beta \le q-1$ for which we will need to satisfy conditions. Recall that we can construct quadrature rules by ensuring that the integral for polynomials up to a certain order are captured exactly. Thus he sets up in each cell $p, p+1$ points such that $p \geq q(q+1)/2$ particles and solves for

$$\sum_{z_j \in \text{cell}_i} P_\alpha(x_j)\, P_\beta(y_j)\, W_j^i = \int_{\text{cell}_i} P_\alpha(x)\, P_\beta(y)\, dx\, dy$$

$$= \delta_{\alpha 0}\, \delta_{\beta 0}\, |\text{Area of cell}_i|$$

for $0 \le \alpha + \beta \le q-1$

This system of equations in ~~P~~ P can be solved, where P are Legendre polynomials. This produces the weights W_j^i. Using these weights over the given points (which are the positions of the particles) one may perform the quadrature very accurately to order q.

This is the general approach that Strain uses to produce a fast adaptive free-Lagrangian vortex method. This method is capable of producing higher order accuracy at ~~long times~~ large times. Solving the system of equations does impose a CPU ~~constraint~~ time penalty but the higher order accuracy is a major gain.

Rough notes for AE 625