Assignment no. 2

Implementation of a Parallel N-Body Solver

DCAMM PhD Course
Scientific Computing

DTU

January 2008

1 Problem setting

Computational Fluid Dynamics (CFD) has matured as an important tool in many fields of science and engineering. The time and length scales considered in the CFD simulations range from million of years and kilometres in astrophysics, to days and hundreds of kilometres in meteorology, to meters and seconds in aerodynamics, and to Ångstrøms (10^{-10} m) and pico seconds (10^{-12} s) in the emerging field of nanofluidics. At the extreme scales of this spectra the systems are most naturally described by their discrete elements — *their particles* – e.g., planets in astrophysics and atoms and molecules in nanofluidics. At the intermediate scale, continuum models such as the Navier-Stokes equations present an efficient average description of the underlying molecular system which may be augmented with continuum or discrete models for sub-grid scale physics e.g., turbulence, pollutant transport, sprays and granular flow cf. Figure 1.

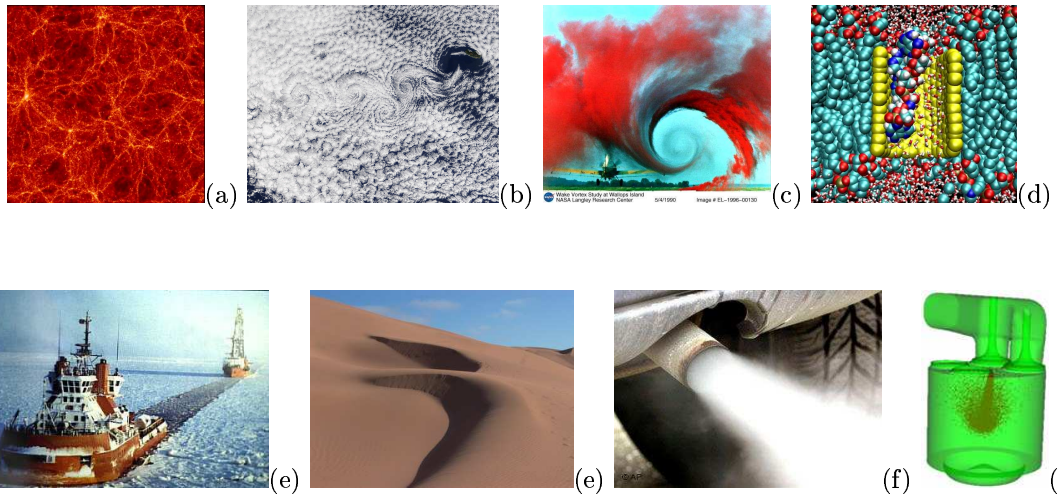

Figure 1: Computational Fluid Dynamics (CFD) is employed for simulations in (a) astrophysics, (b) meteorology, (c) aerodynamics, (d) nanofluidics, (e) ice dynamics, (f) granular flow, (g) pollutant transport, and (h) sprays. CFD tools based on particle methods provide a unified approach to both the continuum (b-c) and discrete systems (a+d, e-h).

Particle Methods provide a unifying framework for the numerical simulation of these systems through a Lagrangian formulation and discretization of the underlying governing equations. The

computational efficiency of these methods is however limited by the mutual interaction of the N particles — an N-body problem:

$$\frac{d\vec{x}_p}{dt} = \vec{v}_p, \tag{1}$$

$$\vec{v}_p = \sum_{q,q\neq p}^{N} \vec{K}(\vec{x}_p - \vec{x}_q)\alpha_q. \tag{2}$$

Here $\vec{x}_p = \begin{pmatrix} x_p \\ y_p \end{pmatrix}$ and $\vec{v}_p$ denote the position and velocity of the p-th particle ($p = 1, \ldots, N$), and α_p is the *strength* of the particle. $\vec{K}$ is the velocity *kernel*, which for the two-dimensional Navier-Stokes equations reads:

$$\vec{K}^{2D}(\vec{x}_p - \vec{x}_q) = \frac{\begin{pmatrix} y_q - y_p \\ x_p - x_q \end{pmatrix}}{2\pi|\vec{x}_p - \vec{x}_q|^2}. \tag{3}$$

In astrophysics $K^{2D} = -Gm_p\frac{(\vec{x}_p-\vec{x}_q)}{|\vec{x}_p-\vec{x}_q|^2}$, where m_p is the mass of the planet and G is the gravitation constant. In nanofluidics $K^{2D} = -\frac{q_p}{2\pi\epsilon_0}\frac{(\vec{x}_p-\vec{x}_q)}{|\vec{x}_p-\vec{x}_q|^2}$ is the Coulomb potential, where q_p is the charge of the atom and ϵ_0 is the vacuum permittivity.
The goal for the present exercise is to parallelize an existing implementation of the N-body problem (Eqs. (1–3)) as it occurs in particle vortex methods used for the simulation of two-dimensional, inviscid (zero viscosity) fluid flow

2 Governing Equations

The dynamic behaviour of an inviscid fluid is described by the incompressible Euler equations:

$$\frac{\partial\omega}{\partial t} + \vec{v}\cdot\nabla\omega = \frac{D\omega}{Dt} = 0, \tag{4}$$

where $\omega \equiv \nabla \times \vec{v}$ is the fluid vorticity and $\vec{v}$ the fluid velocity. The velocity is related to the vorticity through a Poisson equation

$$\nabla^2\psi = -\omega, \tag{5}$$

where $\psi : \vec{v} = \nabla \times (\psi\vec{e}_z)$ is the streamfunction, and $\vec{e} = (0, 0, 1)^T$. The two-dimensional Green's function to ∇^2 (Eq. (5)) yields the Biot-Savart relation:

$$\vec{v}(\vec{x}) = \frac{1}{2\pi}\int\int \frac{(\vec{x}-\vec{y})\times\omega(\vec{y})\vec{e}_z}{(\vec{x}-\vec{y})^2}d\vec{y}. \tag{6}$$

3 Discretisation

The present implementation discretises the vorticity field ($\omega(x)$) using N vortex particles initially located on a regular mesh cf. Fig. 2 with circulation/strength $\Gamma_p \equiv \alpha_p = \omega(\vec{x}_p)h^2$, and h is the mesh spacing. The discrete form of the Biot-Savart relation (Eq. (6)) is an N-body problem (Eq. (2)), where the singular kernel $\vec{K}$ is replaced by a smooth 4th order kernel:

$$\begin{aligned} \vec{K}_\delta^{(4)}(\vec{x}) &= \frac{(-y,x)^T}{2\pi r^2}\left(1 - 2e^{-|\vec{x}|^2/\delta^2} + e^{-|\vec{x}|^2/2\delta^2}\right), \\ &= \frac{(-y,x)^T}{2\pi r^2}\left(1 - e^{-|\vec{x}|^2/2\delta^2}\right)\left(1 + 2e^{-|\vec{x}|^2/2\delta^2}\right). \end{aligned} \tag{7}$$

To ensure convergence of the discretisation, the smoothing length (δ) should satisfy: $\delta \geq h$.

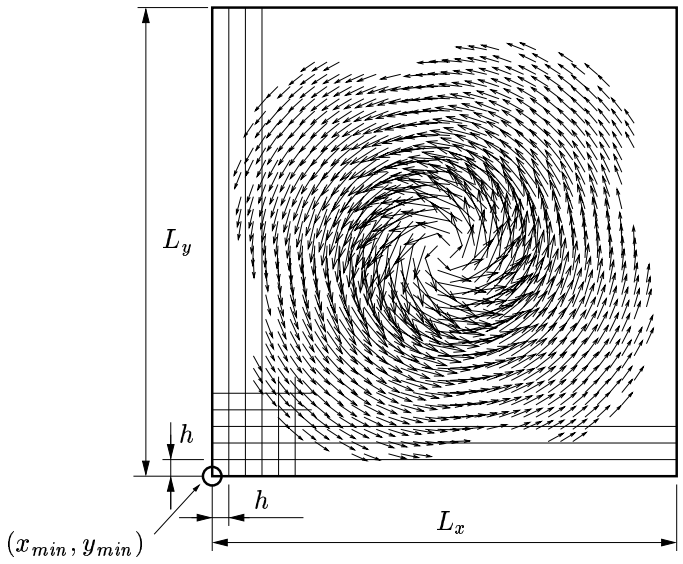

Figure 2: Shapshot of the initial condition.

4 Benchmark: Perlman vorticity patch

Simulate the inviscid evolution of the Perlman vorticity patch [6]:

$$\omega(z, t=0) = \begin{cases} \left(1-|z|^2\right)^7 & |z| \le 1 \\ 0 & |z| > 1, \end{cases} \tag{8}$$

were $\omega(z)$ is the fluid vorticity and $z = (x^2+y^2)^{-1/2}$.
The corresponding (exact) velocity field is time independent:

$$\vec{u}(z,t) = f(|z|) \begin{pmatrix} y \\ -x \end{pmatrix}, \tag{9}$$

where

$$f(|z|) = \begin{cases} -\dfrac{1}{16|z|^2}\left(1-(1-|z|^2)^8\right) & |z| \le 1, \\ -\dfrac{1}{16|z|^2} & |z| > 1. \end{cases} \tag{10}$$

Initialize the particles on a (M, M) Cartesian mesh with a constant mesh spacing (h) is both directions. The particle are placed at the centre of each grid box $(\vec{x}_p)$ with a strength $\Gamma_p \approx \omega(\vec{x}_p, t=0)h^2$. Note, particles are only required where $\omega \neq 0$!

Question A: Compute the root mean square (RMS) error of the particle velocity:

$$E_v(t) = \left(\frac{1}{N} \sum_p^N |\vec{v}_p - \vec{u}(\vec{x}_p)|^2 \right)^{1/2}. \tag{11}$$

Use four different mesh resolutions: $M = 10$, 20, 30, and 40 (corresponding to 80, 316, 716, and 1264 particles) to study the convergence rate of the vortex blob approximation at time $t = 0$.

Hint: Plot $E_v(t=0)$ versus N in a $\log - \log$ plot — the slope of the curve will indicate the order of the method. Compare the observed order with the theoretical prediction [2].

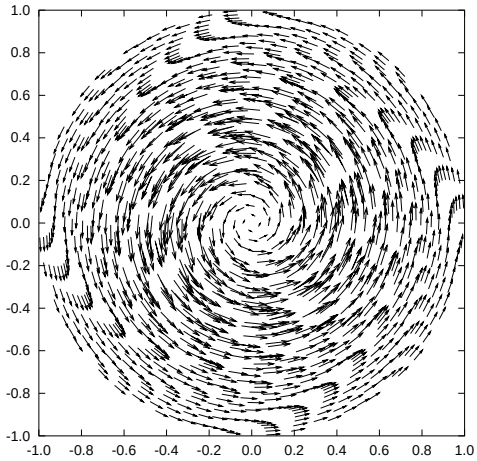

Figure 3: Shapshot of the particle position and velocity.

Question B: Simulate the evolution of the Perlman vortex from $t = 0$ to $t = 20$ using a time step (δt) of 0.2 (a total of 100 time steps). Use a second (or higher) order time integration scheme (this time step should be small enough, that the errors we observe are due to the spatial discretisation and not due to temporal errors).

Plot the time history of the RMS error for the four mesh resolutions. Based on this plot, is the initial order preserved during the time evolution ?

Hint: Does the difference (or ratio) between the different curves look constant in time ?

Question C: Plot the velocity of the particles for the four mesh resolutions cf. Fig. 3.

Question D: How does the computational cost depend on N ?

5 Parallelisation

Parallelize the N-body solver using the following decomposition strategies:

Replicate data:

- Add calls to `MPI_Init`, `MPI_Comm_Rank`, `MPI_Comm_Size`, and `MPI_Finalize`.
- Let root perform the output.
- Split the loops in the most time consuming subroutines across the processors.
- Gather the result using `MPI_AllReduce`.

Ring topology:

- Split the particles equally across the processors.
- Create a copy of the (local) particles and send this copy around the ring.
- Compute the interaction between the local particles and the copy.
- Compute the interaction between the local particles.

Verify that the parallel implementation is correct by re-doing the Perlman benchmark.
Report the parallel runtime (T_p), speedup (S), and efficiency (E).

6 Optional tasks

- Use symmetry: theoretically the computational cost can be reduced by a factor of two by using symmetry: i.e., $(\vec{x}_p - \vec{x}_q) = -(\vec{x}_q - \vec{x}_p)$. Re-arrange the DO loops to achive this. Report the speedup.
- Simulate the evolution on an elliptic vortex patch [5].
- Outline the principle of alternative, "fast" algorithms [1], [4], [3] and how you would parallelize them.

7 Writing a report on the results

The report has to have the following issues covered:

1. Problem description
2. Theoretical part
   (a) Description of the chosen partitioning strategy - why has this strategy been chosen?
3. Numerical experiments:
   (a) Give a description of the parallel implementation - algorithms and data structures. What kind of communications have been used and why?
   (b) Include table(s) of results for various problem (number of particles) sizes, varying number of processors, corresponding speedup and efficiency figures.
   (c) Present plots of the speedup (show also the ideal speedup on the plot).
4. Conclusions, ideas for possible optimizations.

The report must be written in English. A listing of the program code has to be attached to the report. Standard requirements are put on the design of the code, namely, structure and comments. You are encouraged to work in pairs. However, all topics in the assignment should be covered by each student, and you have to hand in individual reports!

8 Deadlines and credit points

The full report (paper copy!) must be submitted no later than January 28, 2008. Assignments submitted after this date will not be approved.

Jens Honoré Walther

References

[1] Josh Barnes and Piet Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324(4):446–449, 1986.

[2] J. T. Beale and A. Majda. High order accurate vortex methods with explicit velocity kernels. *J. Comput. Phys.*, 58:188–208, 1985.

[3] Charles K. Birdsall and Dieter Fuss. Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation. *J. Comput. Phys.*, 3:494–511, 1969.

[4] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[5] P. Koumoutsakos. Inviscid axisymmetrization of an elliptical vortex ring. *J. Comput. Phys.*, 138:821–857, 1997.

[6] Mirta Perlman. On the accuracy of vortex methods. *J. Comput. Phys.*, 59:200–223, 1985.