Particle methods for micro and macroscale flow simulations

By P. Koumoutsakos

1. Introduction

Particle methods are powerful computational techniques to simulate phenomena ranging from protein formation to stellar cluster formation. In fluid dynamics, particle methods have been implemented for simulations of flows inside micro and nanotubes (via *molecular dynamics* simulations) as well as for flows around aerodynamic shapes (via *vortex methods*). The underlying principle of particle methods is the use of computational elements that automatically adapt to resolve the flow field.

In molecular dynamics simulations, the computational elements carry information about the material properties of the fluid while elements in vortex methods represent macroscale quantities such as the vorticity of the flow field. The method of molecular dynamics has been extensively developed in the last decade, and the reader is referred to the review article by Koplik and Banavar (1998) and references therein for an extensive survey of computational issues in flow simulations using this technique.

Vortex methods are based on the discretization of the vorticity field and the Lagrangian description of the governing equations, which when solved determine the evolution of the computational elements. In addition to automatically adapting to the solution, classical vortex methods enjoy advantages such as the use of computational elements only where the vorticity field is nonzero and the rigorous treatment of boundary conditions at infinity. Until recently, disadvantages such as the computational cost and the inability to treat accurately viscous effects had limited their application to modeling the evolution of the vorticity field of unsteady high Reynolds number flows using a few tens to a few thousands computational elements. These difficulties have been overcome with the advent of fast summation algorithms (multipole and particle-grid techniques) that have optimized the computational cost. Moreover, recent developments in numerical analysis allow for the accurate treatment of viscous effects. Vortex methods have today reached a level of maturity, offering an interesting alternative to finite difference and spectral methods for high resolution numerical solutions of the Navier-Stokes equations. In the last three decades research in numerical analysis aspects of vortex methods has provided a solid mathematical background for understanding the accuracy and stability of the method (see Cottet and Koumoutsakos 1999 - referred to as *CK99* from here on). At the same time vortex methods retain their appealing physical character that, we believe, was the motivation for their introduction.

In this article, we report some recent developments on the formulation of boundary conditions and the implementation of spatially varying smoothing functions for vortex methods (*CK99*). These developments have been implemented in Cloud In Cell and Fast Multipole algorithms for simulations of vortex ring-wall interactions and cylinders in rotational oscillations, revealing a drastic modification of the wake structure and significant drag reduction.

In the present line of work we exploit the common features of various particle methods in order to construct computational tools for flow simulations over a large range of scales. Hence the tree data structure that has been developed for the efficient velocity evaluation in vortex methods is being used to identify near neighbors for Lennard-Jones type interaction in Molecular Dynamics Simulations. Using this algorithm we conduct simulations of droplet evaporation and coalescence.

Our goal is a hybrid particle algorithm suitable for simulations of macroscale flows involving micro and nano devices. In such an algorithm molecular dynamics could be implemented for the microscale component simulations, thus providing us with boundary condition for the simulations of the macroscale flows, using vortex methods.

2. Vortex methods

The computationally intensive part of vortex methods is the evaluation of the velocity field on the computational elements (particles) from the vorticity field. As particles carry vorticity the straightforward implementation amounts to pairwise interactions of the computational elements. For N particles this is the classical N-body problem whose computational cost scales as $\mathcal{O}(N^2)$. Fast velocity evaluations can be realized either by employing Particle-Mesh (Cloud in Cell – CIC) techniques or by using multipole methods and efficient tree data structures. While the latter algorithms are the methods of preference as they do not require explicit far field conditions or any regularity of the domain, CIC algorithms are the method of choice for flows in regular and/or periodic domains due to the efficiency and speed of existing Fast Poisson solvers.

2.1 Cloud In Cell (CIC)

An excellent account of particle mesh techniques such as CIC can be found in the classic book by Hockney and Eastwood (1988). At each time step the mesh vorticity is constructed from the particle strength using higher order assignment/interpolation functions. A vector Poisson equation, $\nabla^2\boldsymbol{\Psi} = -\boldsymbol{\omega}$ is solved for the stream function $\boldsymbol{\Psi}$ subject to Dirichlet or Neumann boundary conditions. The particle velocity is subsequently interpolated from the mesh velocity $\mathbf{u} = \nabla \times \boldsymbol{\Psi}$ using the same interpolation procedures. We implement 4th order interpolation/assignment schemes that conserve up to the third moment of the vorticity flowfield invariants of the flow.

Simulations of vortex ring reconnection using a CIC code are currently being conducted in parallel computer architectures. Typical simulations use 60000 particles for the initial configuration and approximately 500000 particles during the reconnection phase. The calculations were performed on a Sun Ultra 2 workstation and on a NEC SX4. The CPU time per time step for 500000 particles and a 140^3 mesh

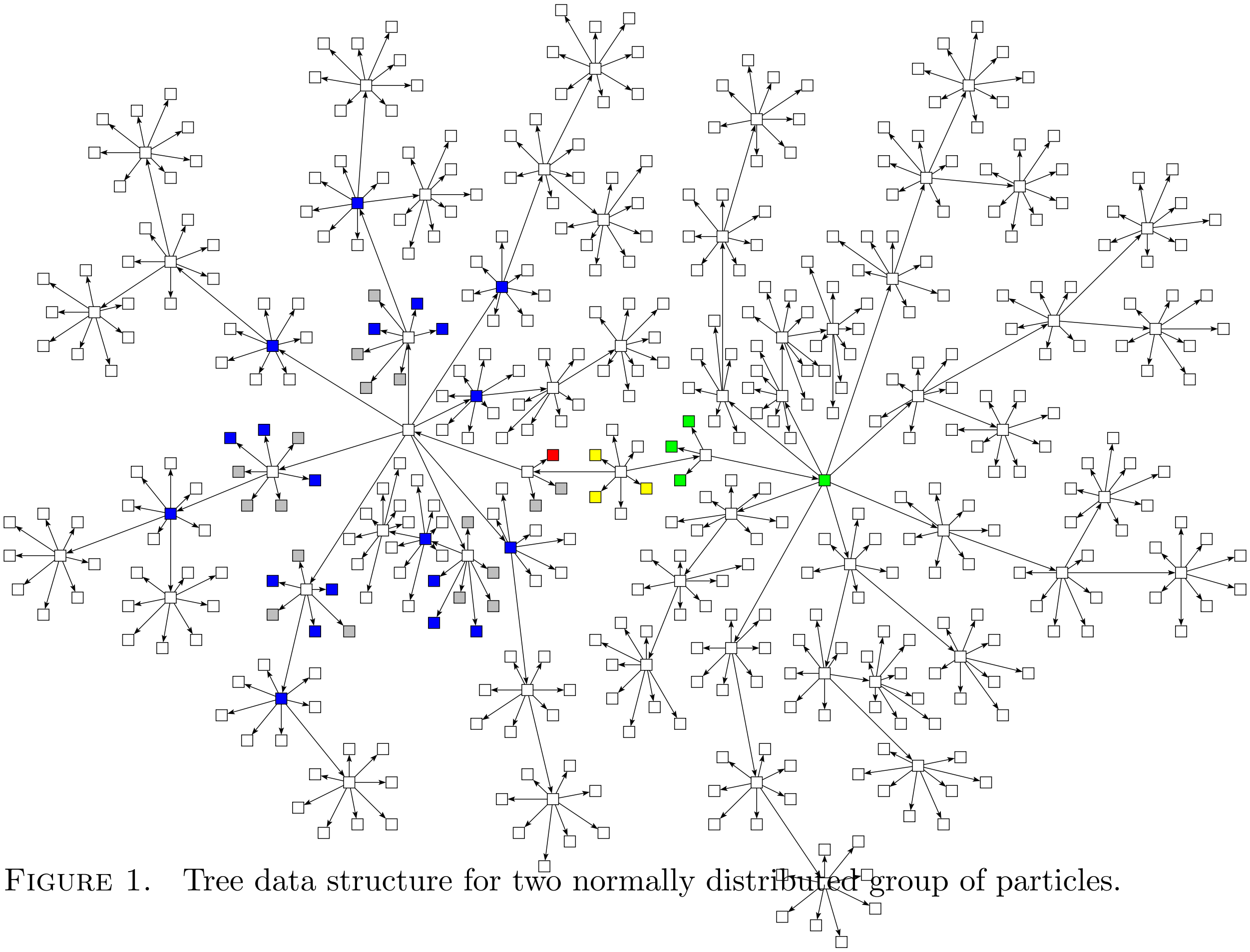

FIGURE 1. Tree data structure for two normally distributed group of particles.

is approximate 140 seconds for the Sun and 4.7 seconds for the SX4. The sustained performance on the SX4 is 1.0 GFlop, approximately 50% of peak performance.

2.2 Fast multipole methods

The CIC method is ideally suited to vortical flows in simple geometries, for which fast Poisson solvers are efficient. For complex geometries, the velocity field can be derived from the vorticity field using the Biot-Savart integral, enforcing at the same time the far field boundary conditions. However, when this integral is discretized using as quadrature points the locations of the vortex particles, the nominal cost of the method is proportional to the square of the computational elements, making it prohibitively expensive. A remedy to this situation, which unlike the CIC avoids the implementation of a grid, while maintaining the accurate treatment of the far field boundary condition is the Fast Multipole Method (FMM) first proposed by Greengard and Rohklin (1987) (see also a more recent review article by Greengard, 1997). It is based on the observation that the influence of a cluster of particles at a certain distance may be approximated by a finite series expansion. In order to exploit this observation, clusters of particles are spatially decomposed into a hierarchy of clusters formally represented by a tree data structure.

The tree is constructed by initially defining its root as the cubic box enclosing

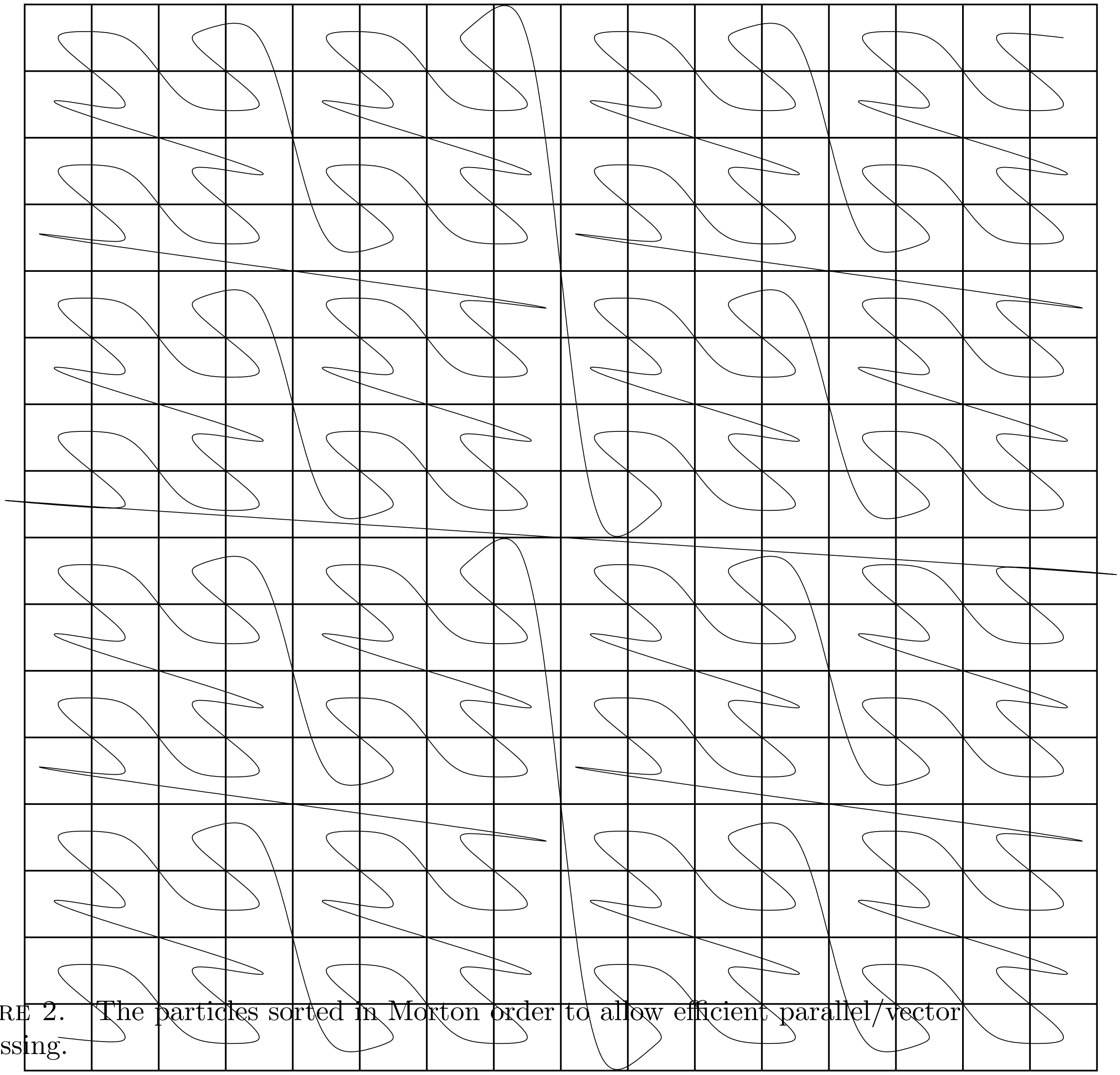

FIGURE 2. The particles sorted in Morton order to allow efficient parallel/vector processing.

all the computational elements. The root and its descendants are recursively subdivided into eight identical boxes until each of the boxes contains only a certain maximum number of particles or the maximum allowable levels of subdivisions has been reached. A genealogical list of parent child relations and four interaction lists of each box in the tree effectively determine the validity of the expansions.

Figure 1 shows an example of an imploded adaptive oct-tree for two normal distributed groups of particles. The tree consists of 430 boxes in 6 levels.

The contribution of a cluster of particles to the potential of a given particle can be computed to desired accuracy if the particle is sufficiently far from the cluster in proportion to the size of the cluster and a sufficiently large number of terms in the multipole expansions is taken. Since the creation of the p-term expansions requires $\mathcal{O}(Np^2)$ operations for each of the $\log_8 N$ levels of the tree, the total amount of work scales as $\mathcal{O}(\mathcal{N} \log \mathcal{N})$.

Further improvements are possible if well separated boxes are allowed to interact. These interactions are in the form of shifting the expansions of a certain cluster with the desired accuracy to the center of another cluster. Then these expansions are used to determine the velocities of the particles in the second cluster. The cost associated with the translation of a p-term multipole expansion is $\mathcal{O}(p^4)$ or $\mathcal{O}(p^2)$ if the multipole expansions are converted into exponential expansions (Greengard, 1997). The box-box interaction minimizes the tree traversal for the individual particles, making the algorithm formally $\mathcal{O}(N)$. A parallel version of the FMM is currently being developed using the exponential expansions. Other techniques for promoting vectorization and parallelization (data locality) is to sort the particles according to their position in the tree using Morton ordering. Particles in the same childless box are mapped consecutively in memory, securing an efficient stride im memory (Fig. 2). The spatial relations described by the tree are further utilized during load balancing of the particles and the boxes.

The tree data structure is used to identify clusters of particles and to identify for a certain particle its near-neighbor list and the clusters for which far-field expansions will be implemented. Naturally this property is exploited for other types of particle simulations such as the molecular dynamics presented in the following section.

2.3 Boundary conditions for 3d viscous vortex methods

The formulation of vorticity boundary conditions in terms of integral equations linking boundary terms and vorticity in the flow is presented in *CK99*. Following this work, we report here the implementation of boundary conditions in a three-dimensional viscous splitting algorithm.

Vorticity boundary conditions for three-dimensional viscous flows have, compared to the two-dimensional case, two additional difficulties. First, since vorticity is a vector, one needs 3 instead of 1 boundary condition. Secondly, vorticity created at the boundary must be divergence-free, and this constraint must enter the boundary conditions. To simplify the exposition, we will assume that the boundary is a flat plate located at $x_3 = 0$. The general case follows by using local coordinate axis parallel and orthogonal to the wall. We will also assume a velocity vanishing at the wall.

One vorticity boundary condition immediately follows from the no-slip condition: the wall-normal component of the vorticity vanishes at the wall:

$$\omega_3 = 0$$

It is worthwhile to notice that a consequence of this condition is that, at the wall, $(\mathbf{u} \cdot \nabla)\omega_3 = (\boldsymbol{\omega} \cdot \nabla)u_3 = 0$. Hence the normal component of the vorticity equation written at the wall yields

$$\frac{\partial^2 \omega_3}{\partial x_3^2} = 0$$

Let us now turn to the tangential components of the vorticity. A natural extension of the two-dimensional vorticity flux boundary conditions is to enforce no-slip for the components u_1 and u_2 of the velocity by creation of vorticity for the components ω_2 and ω_1 respectively

This leads to the following three-dimensional version of the algorithm:

I. solve the convection diffusion for the three components of the vorticity, with homogeneous Dirichlet boundary condition for the normal component and homogeneous Neumann boundary conditions for the tangential components:

$$\omega_3 = \frac{\partial \omega_1}{\partial x_3} = \frac{\partial \omega_2}{\partial x_3} = 0$$

II. compute the slip (u_1, u_2) at the boundary

III. repeat sub-step 1 for the tangential components, with the new Neumann boundary conditions

$$\frac{\partial \omega_1}{\partial x_3} = -\frac{u_2}{\Delta t} \, , \, \frac{\partial \omega_2}{\partial x_3} = \frac{u_1}{\Delta t}$$

It is easy to check (see *CK99*) that this procedure guarantees that the vorticity remains divergence-free for all times. Figure 3 shows the viscous interaction of a vortex ring impinging at an angle on a solid wall using the high order CIC method. The Reynolds number based on the circulation of the ring is 800. The ring is resolved using 500000 particles and a 140^3 mesh (*CK99*). These simulations were performed on a cluster of DEC-Alpha workstations with a performance of about 5000 particles per CPU second per processor.

2.4 Variable size vortex methods

In order to account for diffusion in vortex methods simulations, we employ the scheme of Particle Strength Exchange (PSE). As it is discussed in *CK99*, the PSE has enough flexibility in dealing with viscous effects to allow the treatment of variable viscous scales in the vorticity redistribution scheme. As we wish to use fewer particles and thus reduce the computational cost, in flow regions with relatively small vorticity gradients (such as a cylinder far wake), we are interested in developing diffusion and convection formulas for vortex methods with spatially varying blob sizes (ϵ). This can be done through merging of nearby vortices, or, more generally, by periodically remeshing the particle distribution on a variable size mesh.

A consistent treatment of the diffusion requires the overlapping of the particles on a scale given by the kernel used in the PSE formula. As a result, a locally coarser particle resolution must go with an increasing diffusion range ϵ. In the case of the Biot-Savart integral the incorporation of variable blobs consists of replacing ϵ by $\epsilon(\mathbf{y})$ in the otherwise unchanged quadrature formulas. However, this technique would not be consistent in the case of the integrals that are used to replace the Laplacian operator in the PSE formulation (see *CK99* for details). The correct way to implement a variable blob size in a PSE scheme is through a change of variables which maps the variable particle grid to a uniform one.

For simplicity let us focus on the one-dimensional case. We will denote locations in the physical space with variable grid size by x, y and locations in the mapped

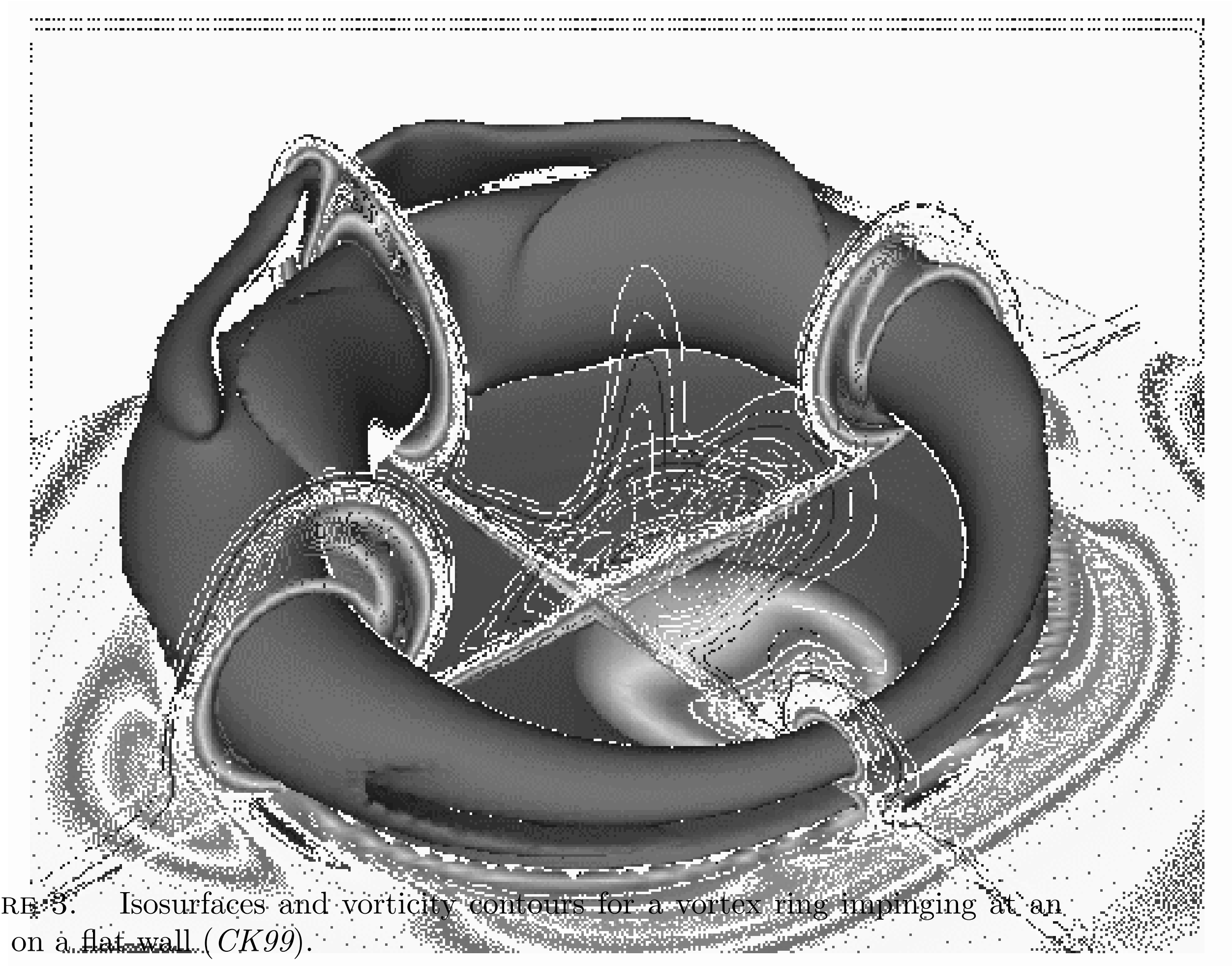

FIGURE 3. Isosurfaces and vorticity contours for a vortex ring impinging at an angle on a flat wall (*CK99*).

coordinates where the grid-size is uniform by $\hat{x}, \hat{y}$. We will assume that the mapping is given by the formulas

$$x = f(\hat{x}), \hat{x} = g(x), \omega(x) = \hat{\omega}(\hat{x}).$$

Writing derivatives in the mapped coordinates yields:

$$\frac{d^2\omega}{dx^2} = h(\hat{x}) \frac{d}{d\hat{x}} \left[h(\hat{x}) \frac{d\hat{\omega}}{d\hat{x}} \right]$$

where $h(\hat{x}) = g'(x)$. Next, we use the following integral approximation (see *CK99* for a proof):

$$\frac{d}{d\hat{x}} \left[h(\hat{x}) \frac{d\hat{\omega}}{d\hat{x}} \right] \simeq \epsilon^{-3} \int \frac{h(\hat{x}) + h(\hat{y})}{2} [\hat{\omega}(\hat{x}) - \hat{\omega}(\hat{y})] \eta(\frac{\hat{x} - \hat{y}}{\epsilon})\, d\hat{y}.$$

In the above formula, the kernel η satisfies the necessary moment properties and ϵ is a *constant* blob size. This leads to the following PSE scheme for the heat equation in one dimension with $\nu = 1$

$$\frac{d\omega_p}{dt} = \epsilon^{-3} h(\hat{x}_p) \sum_q \hat{v}_q \frac{h(\hat{x}_p) + h(\hat{x}_q)}{2} [\omega_q - \omega_p] \eta(\frac{\hat{x}_p - \hat{x}_q}{\epsilon})$$

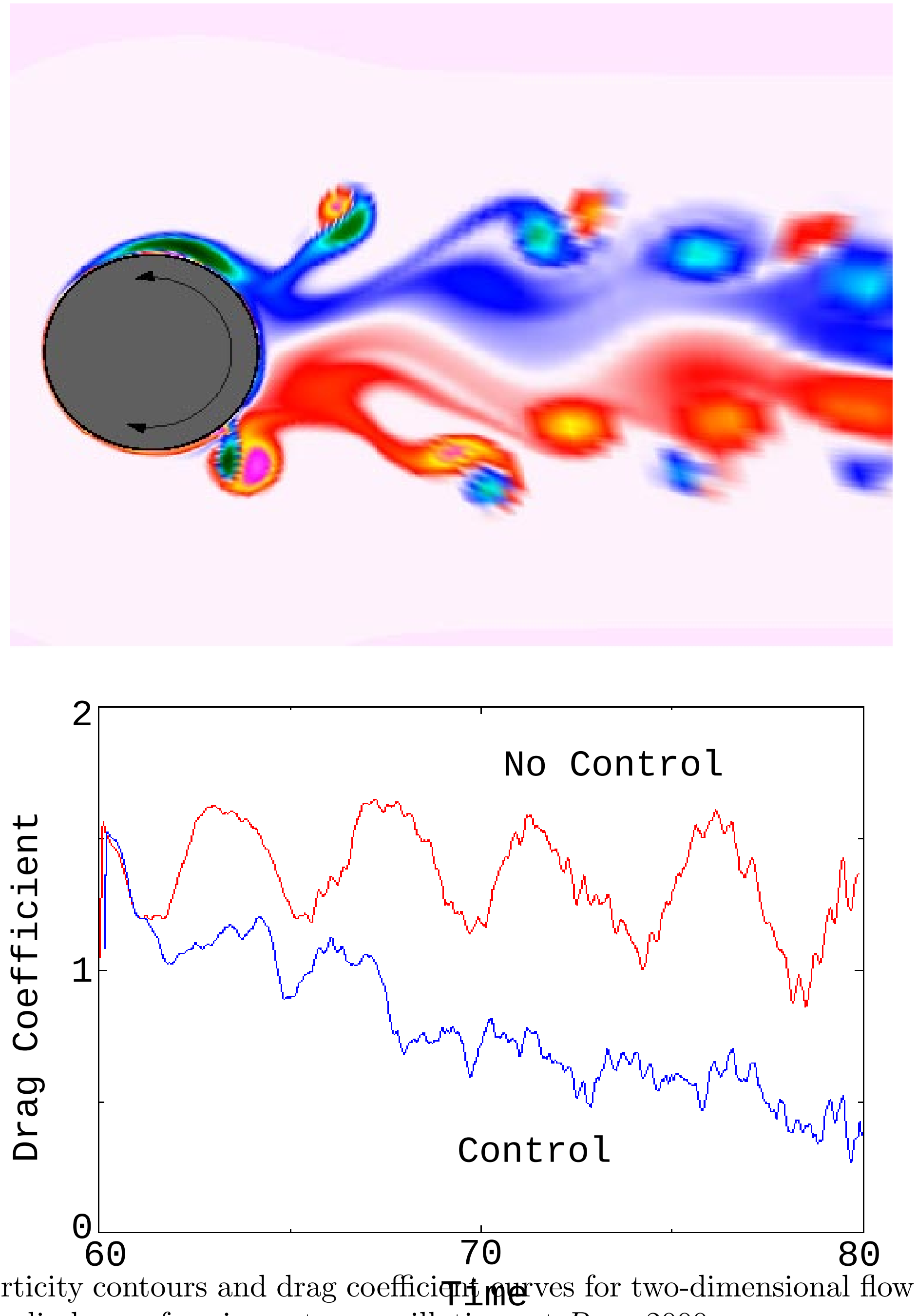

FIGURE 4. Vorticity contours and drag coefficient curves for two-dimensional flow past a circular cylinder performing rotary oscillations at $Re = 2000$.

where $\hat{v}_q$ denotes the volumes of the mapped particles. Notice that the volumes of the physical and mapped particles are related through the Jacobian of the mapping:

$$\hat{v}_q = v_q h(\hat{x}_q)$$

which establishes that the scheme is indeed conservative. To extend this approach to two or three-dimensional flows, one must handle integral approximations of differential terms involving diffusion tensors.

In convection-diffusion problems, the use of variable blob sizes must be combined with regridding techniques to ensure that the particle discretization is everywhere consistent with the local blob sizes and that particles maintain an adequate overlap.

The vortex method is ideally suited to simulations of external flows because it allows an exact treatment of the far field conditions, and their computational effort is restricted to vortical zones of the flow. They can be made even more efficient if one can take advantage of the decay of vorticity gradients in the wake to save computational elements. In the case of a wake behind a cylinder, the blob size is proportional to the distance to the center of the obstacle. Particles are mapped to a uniform mesh through a logarithmic mapping. In the mapped coordinates the diffusion operators allow the use of simple PSE formulas. The particle distributions in the wake of an impulsively started cylinder at $Re = 200$, using the uniform particle distribution and the the variable blob sizes according to the logarithmic mapping, were compared. The evolution of the drag coefficient obtained for the two simulations is identical while the variable blob simulation requires an order of magnitude less particles (about 5000), which translates to respective savings in the computational times.

Using these new vortex methods, we conducted two-dimensional simulations at $Re = 1000$, for a cylinder undergoing rotary oscillations. The motivation for these simulations was to investigate the experimental findings of Tokumaru and Dimotakis (1992), which have shown that at $Re = 15000$ rotary cylinder oscillations may result in up to 80% drag reduction. In the present simulations a similar drag reduction was observed (Fig. 4), and it is attributed to the destruction of the Karman wake and the formation and ejection of vortex dipoles from the surface of the body. Clearly one does not expect the flow to be purely two-dimensional at these Re numbers, and the results of the simulations should not be considered as direct numerical simulations (DNS). However the similar drag reduction between computations and experiments and the mechanism of dipole ejection from the cylinder surface merit further investigation as effective control mechanisms via three-dimensional DNS.

3. Molecular dynamics

Molecular Dynamics (MD) is the method of choice for the study of flow phenomena in micro and nanoscale flows. In the context of particle methods, MD algorithms may be easily constructed from tools already developed for vortex methods. The tree data structures developed for the identification of near-neighbors in vortex methods is implemented in MD simulations using the Lennard-Jones potential.

We present here results from the implementation of a molecular dynamics algorithm to the simulation of microdroplet evaporation and coalescence (see Walther and Koumoutsakos (1999) for further details).

3.1 Droplet evaporation

To study the evaporation of nano-droplets, the tree code was adapted for molecular dynamics (MD) simulation. The short range potential of the 12-6 Lennard-Jones system ($U(r) = 4\epsilon[(\sigma/r)^{12} - (\sigma/r)^6]$) allows a simple truncation of the potential at $r/\sigma = 2.5 - 10$. Hence, the tree effectively serves to identify neighboring particles.

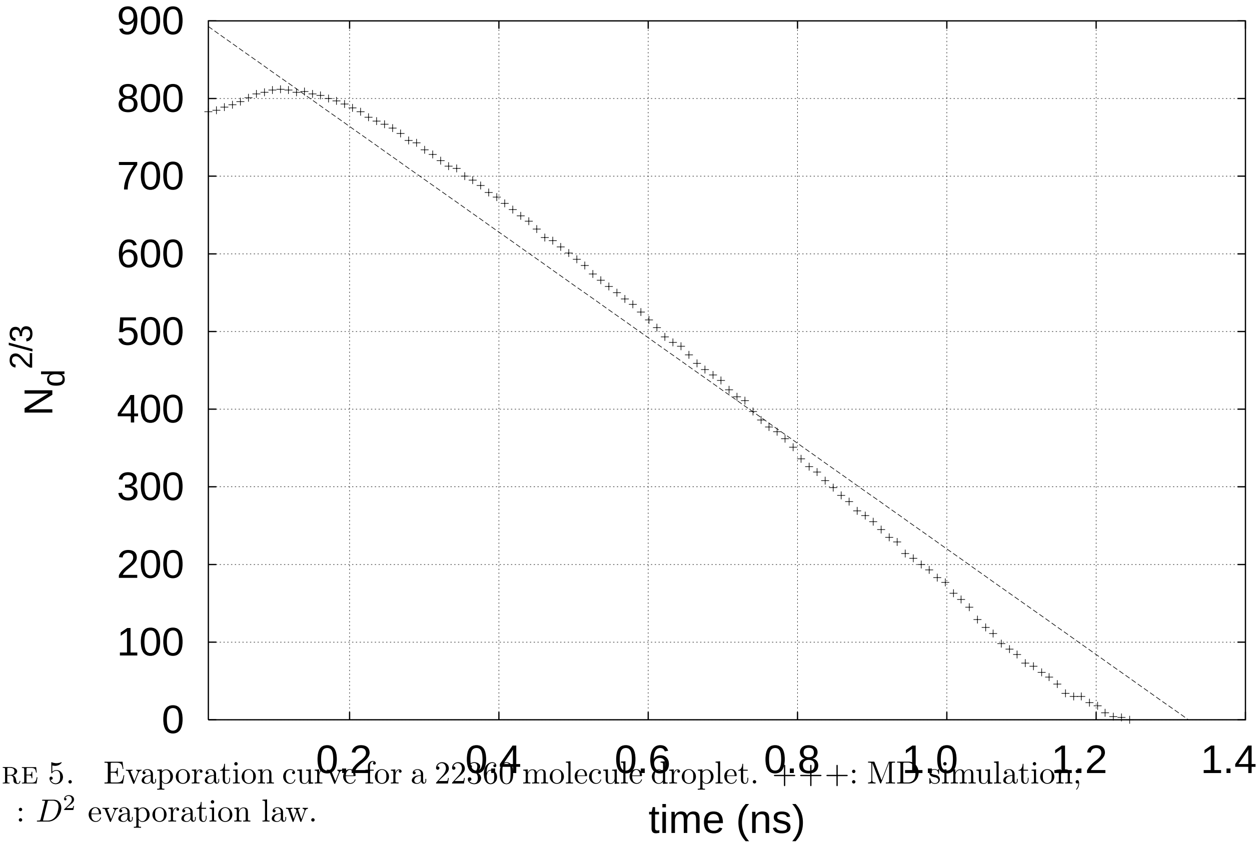

FIGURE 5. Evaporation curve for a 22360 molecule droplet. +++: MD simulation; ---- : D^2 evaporation law.

Standard leapfrog integration is used to advance the molecules in time subject to periodic boundary condition using a ghost-layer technique.

Simulation of the evaporation of an Argon droplet into Argon vapor has been conducted for systems involving 20000 - 150000 molecules. The far-field temperature boundary condition is enforced by heating the molecules in the far-field at regular intervals.

Initially the molecules are places on a face-centered-cubic (f.c.c.) lattice with the desired temperature. The system is then relaxed during 5000 to 10000 time steps, heating the complete system to the equilibrium temperature of $T^* = Tk_B/\epsilon = 0.83$. After the relaxation the temperature of the vapor is increased to $T^* = 2.50$, and as the droplet reaches the saturation temperature ($T^* \approx 1.0$), evaporation commences/begins/starts. The simulations were performed on Sun Ultra 2, allowing approximately 20000 particles per CPU seconds per time step.

Figure 5 compares the predicted evaporation coefficient, β_v, with the D^2 evaporation laws (Kuo, 1986) in terms of the enumeration of molecules in the droplet (N_d). The predicted value is within 10-20% of the D^2-law, which is considered to be a good agreement considering the approximations of the simulation.

3.2 Droplet coalescence

Another study currently being conducted involves the coalescence and breakup of nano droplets. The governing physical parameters are the Weber number, $We = \rho 2RU^2/\gamma$, Reynolds number $Re = \rho 2RU/\mu$, and the impact parameter $B = \chi/2R$,

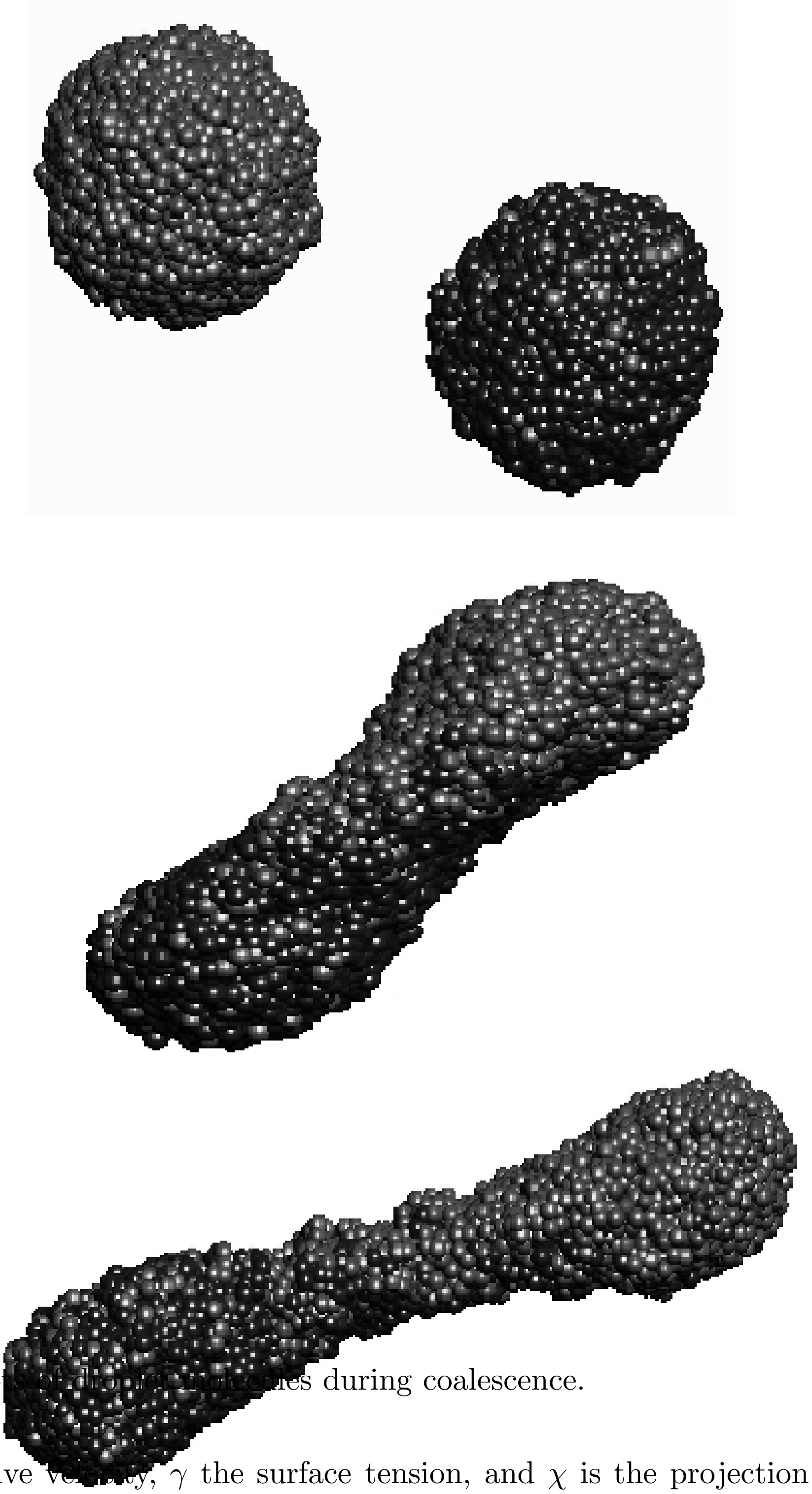

FIGURE 6. Snapshots of droplet molecules during coalescence.

where U is the relative velocity, γ the surface tension, and χ is the projection of the separation distance between the droplet centers in the direction normal to that of U.

The system consists of 2×5576 and 16556 molecules for the droplet and gas phase, respectively. Assuming a Lennard-Jones potential and a cutoff of 6.5σ, the simulations require 5 CPU seconds per time step on a Sun Ultra 2.

Figure 6 shows snapshots of the droplet molecules during coalescence and breakup for the parameters $(We, Re, B) = (210, 570, 0.5)$.

5. Summary and conclusions

We have presented some results from our ongoing investigations on particle methods (vortex methods and molecular dynamics). Having resolved in the last decade several issues concerning the accuracy and numerical capabilities of vortex methods, the key issue is the use of efficient computational techniques for the solution of the associated N-body problem. Particle-Mesh techniques and multipole summation algorithms coupled with efficient tree-data structures have being implemented, allowing for large scale simulations using millions of computational elements in two and three dimensions.

Looking ahead, particle methods are envisioned as a computational technique that could help bridge the gap in simulations of incompressible flows in micro and macroscales. The coupling may be achieved by using molecular dynamics algorithms to simulate the microscale phenomena and provide boundary conditions for macroscale simulations using vortex methods. This hybrid procedure may be further extended to other schemes, coupling particle and grid based methods in a formulation that takes advantage of the complementary advantages of each method. Such numerical algorithms may offer an accurate and viable alternative to existing multi-block schemes while offering the capability of extending simulations to the level of nano and microscale flow phenomena. Portability and efficiency of the numerical codes and the exploitation of emerging large scale parallel computer architectures will be crucial in these developments.

Acknowledgments

I am grateful for the collaboration of Georges-Henri Cottet and Jens Walther throughout the course of this work.

REFERENCES

Cottet, G.-H. and Koumoutsakos, P. 1999 *Vortex methods: theory and applications.* Cambridge University Press.

Greengard, L. and Rokhlin V. 1987 A fast algorithm for particle simulation. *J. Comp. Phys.* **73,** 325.

Greengard, L. and Rokhlin, V. 1997 A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica.* **7,** 229-269.

Hockney R. W. and Eastwood, J. W. 1988 *Computer simulation using particles.* IOP Publishing.

Kolpik J. and Banavar J. R. 1998 Physics of Fluids at Low reynolds Numbers-A molecular Approach. *Comput. Phys.* **12,** 424-431.

Kuo, K. K.-Y. 1986 *Principles of combustion.* John Wiley & Sons.

Tokumaru, P. and Dimotakis, P. 1991 Rotary Oscillation control of a cylinder wake. *J. Fluid Mech.* **224,** 77-90.

Walther, J. and Koumoutsakos, P. 1999 Large scale molecular dynamics simulations of droplet coalescence and evaporation. *(Submitted J. Chem. Phys.)*