# Multiresolution Simulations using Particles

Michael Bergdorf[1] and Petros Koumoutsakos[1]

Computational Science & Engineering Laboratory
ETH Zürich
CH-8092, Switzerland
{petros,bergdorf}@inf.ethz.ch

**Abstract.** We present novel multiresolution particle methods with extended dynamic adaptivity in areas where increased resolution is required. In the framework of smooth particle methods we present two adaptive approaches: one based on globally adaptive mappings and one employing a wavelet-based multiresolution analysis to guide the allocation of computational elements. Preliminary results are presented from the application of these methods to problems involving the development of sharp vorticity gradients. The present particle methods are employed in large scale parallel computer architectures demonstrating a high degree of parallelization and enabling state of the art large scale simulations of continuum systems using particles.

## 1 Approximations using particles

The development of particle methods is based on the integral representation of functions and differential operators. The integrals are discretized using particles as quadrature points.

### 1.1 Function approximation

The approximation of continuous functions by particle methods starts with the equality

$$q(\boldsymbol{x}) \equiv \int q(\boldsymbol{x}-\boldsymbol{y})\,\delta(\boldsymbol{y})\,d\boldsymbol{y}\,. \tag{1}$$

Using $N$ particles we discretize above equality by numerical quadrature and get the "point-particle" approximation of $q$:

$$q^h(\boldsymbol{x}) = \sum_p Q_p(t)\,\delta(\boldsymbol{x}-\boldsymbol{x}_p)\,. \tag{2}$$

Point particle methods based on the apprimation (2) yield exact weak solutions of conservation laws. A drawback of point particle approximations is that the function $q^h$ can only be reconstructed on particle locations $\boldsymbol{x}_p$.

This shortcoming is addressed by mollifying the Dirac delta function in (1) resulting on a mollified approximation:

$$q^\varepsilon(\boldsymbol{x}) = \int q(\boldsymbol{x}-\boldsymbol{y})\,\zeta^\varepsilon(\boldsymbol{y})\,d\boldsymbol{y}\,, \tag{3}$$

where $\zeta^\varepsilon = \varepsilon^{-d}\zeta(\boldsymbol{x}/\varepsilon)$, $\boldsymbol{x} \in \mathbb{R}^d$, and $\varepsilon$ being a characteristic length scale of the kernel. For consistency of the approximation the kernel $\zeta$ has to fulfill the following moment conditons:

$$\int \zeta\, \boldsymbol{x}^{\boldsymbol{\alpha}}\, d\boldsymbol{x} = 0^{\boldsymbol{\alpha}} \text{ for } 0 \leq |\boldsymbol{\alpha}| < r\,. \tag{4}$$

The kernel $\zeta$ is of order $r$ and the following error bound holds:

$$\|q - q^\varepsilon\| \leq C\varepsilon^r \|q\|_\infty\,. \tag{5}$$

Now again, we get a discrete but smooth function approximation by approximating the integral in (3) by a mid point quadrature rule yielding

$$q^{\varepsilon,h}(\boldsymbol{x}) = \sum_p Q_p\, \zeta^\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p)\,, \tag{6}$$

The error of (6) can be assessed by splitting $\|q - q^{\varepsilon,h}\|$ into

$$\begin{aligned} \|q - q^{\varepsilon,h}\| \leq &\quad \|q - q^\varepsilon\| + &\quad \|q^\varepsilon - q^{\varepsilon,h}\| \\ \leq &\quad C_1\varepsilon^r\|q\|_\infty + &\quad C_2\left(\tfrac{h}{\varepsilon}\right)^m \|q\|_\infty\,. \end{aligned} \tag{7}$$

We conclude from this, that $(h/\varepsilon)$ must be smaller that 1, *i.e.* smooth particles *must* overlap[1].

### 1.2 Differential Operator approximation

In smooth particle methods differential operators can be approximated by discrete integral operators. Degond & MasGallic developed an integral representation of the diffusion operator - isotropic and anisotropic - which was later extended to differential operators of arbitrary degree in [6]. The integral operator for the 1D Laplacian for instance takes the form

$$\Delta^\varepsilon q = \frac{1}{\varepsilon^2}\int \left[q(y) - q(x)\right] \eta^\varepsilon(x - y)\, dy\,, \tag{8}$$

where the kernel $\eta(x)$ has to fulfill $\int x^2\, \eta(x)\, dx = 2$. This integral is discretized by particles using their locations as quadrature points:

$$\left(\Delta^{\varepsilon,h} q\right)(x_{p'}) = \varepsilon^{-2}\sum_p \left[q_p - q_{p'}\right]\, \eta^\varepsilon(x_{p'} - x_p)\, v_p\,. \tag{9}$$

[1] for certain kernels, an $r$-th order approximation can be achieved even with $\varepsilon = h$ [17]

## 2 Solving transport problems with Particle Methods

Particle methods discretize the Lagrangian form of the governing equation,

$$\frac{\partial q}{\partial t} + \nabla \cdot (\mathbf{u}\, q) = \mathcal{L}(q, \boldsymbol{x}, t)\,, \tag{10}$$

resulting in the following set of ODEs:

$$\begin{aligned} \frac{d\boldsymbol{x}_p}{dt} &= \mathbf{u}(\boldsymbol{x}_p, t)\,, && \text{positions} \\ \frac{dv_p}{dt} &= v_p\ (\nabla \cdot \mathbf{u})\,(\boldsymbol{x}_p, t)\,, && \text{volumes} \\ \frac{dQ_p}{dt} &= v_p\, \mathcal{L}^{\varepsilon,h}(q, \boldsymbol{x}_p, t)\,. && \text{weights} \end{aligned} \tag{11}$$

Particle positions are usually initialized as a regular lattice with spacing $h$, volumes are thus initially set to $v_p = h^d$ and $Q_p = q_o(\boldsymbol{x}_p)\, h^d$. The ODES (11), are now advanced using a standard explicit time stepper and the transported quantity $q$ can be reconstructed as

$$q(\boldsymbol{x}, t) = \sum_p Q_p(t)\, \zeta^\varepsilon\left(\boldsymbol{x} - \boldsymbol{x}_p(t)\right)\,. \tag{12}$$

However, as the particles follow the flow map $\mathbf{u}(\boldsymbol{x}, t)$ their positions eventually become irregular and distorted, and the function approximation (12) ceases to be well-sampled. To ascertain convergence, it is therefore necessary to periodically regularize the particle locations; this process is called “remeshing”.

### 2.1 Remeshing

Remeshing involves interpolation of particle weights from irregular particle locations onto a regular lattice. New particles are then created on the lattice, replacing the old particles. This interpolation process takes the form

$$Q_{p'}^{\text{new}} = \sum_p W(\boldsymbol{x}_{p'} - \boldsymbol{x}_p)\, Q_p^{\text{old}}\,, \tag{13}$$

where $Q_{p'}^{\text{new}}$ are the new particle weights, and $\boldsymbol{x}_{p'}$ are located on a regular lattice. The interpolation function $W(\boldsymbol{x})$ is commonly chosen to be a tensor product of one-dimensional interpolation function which for accuracy have to be sufficiently smooth and moment-conserving. The $M_4'$ function [14] is commonly used in the context of particle methods; it is in $C^1(\mathbb{R})$ and of third order.

The introduction of a grid clearly detracts from the meshless character of particle methods. The use of a grid in the context of particle methods does not restrict the adaptive character of the method and provides the basis for a new class of “hybrid” particle methods with several computational and methodological benefits

### 2.2 Hybrid particle methods

The introduction of a grid enables fast evaluation of differential operators using compact PSE kernels, enables the use of fast grid-based Poisson solvers [8], facilitates parallelization and is a key component in adaptive particle methods, which we will present in section 3. Hybrid particle methods make heavy use of these computational advantages [19, 3, 11].

Recently, we have developed a generic hybrid particle method framework [16], enabling efficient, parallel simulations of large-scale transport problems as diverse as the DNS of turbulent flows and diffusion processes in complex biological organelles. Figure 1 shows visualizations of the Crow instability and the elliptic instability of two counter-rotating vortex tubes employing a maximum of 33 million particles. The simulations were performed on a 16 cpu Opteron cluster. One time step for 1 million particles took less than 30 seconds. Current implementations using the fast multipole method which retain the meshless character of the particle method require approximately 2400 seconds per time step [20]. This clearly demonstrates the advantages of hybrid methods.

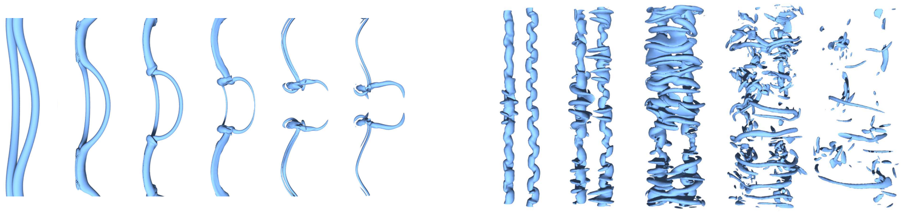


**Fig. 1.** Crow (left) and short-wave or elliptic instability (right)

## 3 Adaptive Particle Methods

The accuracy of smooth particle methods is determined by the core size $\varepsilon$ of the kernel $\zeta^{\varepsilon}(\boldsymbol{x})$. For computational efficiency this core size needs to be spatially variable to resolve small scales in different parts of the flow, such as the boundary layer and the wake of bluff body flows. As particles need to overlap, varying core sizes imply spatially varying particle spacings. This can be achieved in two ways:

- remeshing particles on a regular grid corresponding to variable size particles in a mapped using a global (adaptive) mapping
- remeshing particles by combining several simple local mappings in a domain decomposition frame.

In the context of vortex methods, Hou [10] first introduced spatially varying particle sizes and proved the convergence of the method in the case of the 2D

Euler equations. This proof was extended in [15] to the viscous case and the method was used for the simulation of wakes with stetched particle resolution. In [2] Cottet, Koumoutsakos, and Ould-Salihi formulated a convergent variable core size vortex method for the Navier-Stokes equations by using mappings from a reference space $\hat{\Omega} \subseteq \mathbb{R}^d$ with uniform core size $\hat{\varepsilon}$ to the "physical" space $\Omega \subseteq \mathbb{R}^d$ with cores of varying size $\varepsilon(\boldsymbol{x})$ in conjunction with an anisotropic diffusion operator, *i.e.*

$$\boldsymbol{x} = \boldsymbol{f}(\hat{\boldsymbol{x}})\,, \qquad \hat{\boldsymbol{x}} = \mathbf{g}(\boldsymbol{x})\,, \qquad \left\{\underline{\boldsymbol{\Phi}}\right\}_{ij} = \frac{\partial \hat{x}_i}{\partial x_j} \qquad \text{and} \qquad |\underline{\boldsymbol{\Phi}}| = \det \underline{\boldsymbol{\Phi}} \tag{14}$$

Like in the uniform core size method (11), we convect the particles in physical space, but diffusion is performed in reference space, so that with $N$ particles, located in $\{\boldsymbol{x}_j(t)\}_{j=1}^N = \{\boldsymbol{f}(\hat{\boldsymbol{x}}_j)\}_{j=1}^N$ we find an approximate solution to (10) by integrating the following set of ODEs:

$$\begin{aligned} \frac{d\boldsymbol{x}_j}{dt} &= \mathbf{u}(\boldsymbol{x}_j, t)\,, \\ \frac{dQ_j}{dt} &= \frac{\nu}{\hat{\epsilon}^2} \sum_k \psi_{pq}^{\hat{\epsilon}}(\hat{\boldsymbol{x}}_j - \hat{\boldsymbol{x}}_k) \Big(\frac{m_{pq}(\hat{\boldsymbol{x}}_j) + m_{pq}(\hat{\boldsymbol{x}}_k)}{2}\Big) [\hat{v}_j \hat{Q}_k - \hat{v}_k \hat{Q}_j]\,, \\ \frac{d\hat{v}_j}{dt} &= \hat{\nabla} \cdot \big(\underline{\boldsymbol{\Phi}}\mathbf{u}\big)(\boldsymbol{x}_j, t)\, \hat{v}_j\,. \end{aligned} \tag{15}$$

In the above equation $Q_j$ and $\hat{Q}_j$ denote the particle strength in physical and reference space, respectively, related by

$$\hat{Q}_j = Q_j |\underline{\boldsymbol{\Phi}}|(\boldsymbol{x}_j)\,,$$

and $m_{pq} = b_{pq} - \frac{1}{d+2}\delta_{pq}\delta_{p'q'}b_{p'q'}$, with

$$b_{p'q'} = \frac{1}{\underline{\boldsymbol{\Phi}}} \frac{\partial(\hat{\boldsymbol{x}})_{p'}}{\partial(\boldsymbol{x})_r} \frac{\partial(\hat{\boldsymbol{x}})_{q'}}{\partial(\boldsymbol{x})_r}$$

and $\psi_{pq}(\boldsymbol{x}) = (\boldsymbol{x})_p\,(\boldsymbol{x})_q\,\rho(\boldsymbol{x})$, $\rho(\boldsymbol{x})$ being a radially symmetric kernel with suitable moment properties [4].

In [2] analytic, invertible mappings have been employed. Albeit being a simple and robust way to efficiently resolve the range of length scales in the flow, this method requires prior knowledge about the flow physics. In [1] we extended this method by introducing two different approaches to dynamical adaptivity in particle methods; One approach makes use of a global adaptive mapping (AGM, see section 3.1), and one employing dynamically placed patches of smaller sized particles, reminiscent of adaptive mesh refinement in finite volume methods (AMR).

### 3.1 Particle method with adaptive global mappings

We introduce a transient smooth map $\mathbf{f} : \hat{\Omega} \times [0, T] \to \Omega$ represented by particles:

$$\boldsymbol{x}(\hat{\boldsymbol{x}}, t) = \mathbf{f}(\hat{\boldsymbol{x}}, t) = \sum_j \boldsymbol{\chi}_j(t)\, \zeta^{\hat{\varepsilon}}(\hat{\boldsymbol{x}} - \boldsymbol{\xi}_j)\,, \tag{16}$$

where $\boldsymbol{\xi}_p$ are fixed at grid point locations. The parameters in the map that are changed in the process of adaptation are the node values $\boldsymbol{\chi}_j$. As the map (16) is not easily invertible, we require it to be smooth in both space and time. Given this property, the governing equation (10) can be entirely cast into reference space, again yielding a transport equation:

$$\frac{\partial \hat{q}'}{\partial t} + \hat{\nabla} \cdot (\hat{q}' \, \tilde{\boldsymbol{u}}) = \hat{\mathcal{L}}(\hat{q}', \hat{\boldsymbol{x}}, t) \,, \tag{17}$$

where $\hat{q}' = (|\Phi|)^{-1} \hat{q}$ and

$$\tilde{\boldsymbol{u}} = \Phi(\hat{\boldsymbol{u}} - \boldsymbol{\mathcal{U}}) \,, \qquad \text{and} \qquad \boldsymbol{\mathcal{U}} = \frac{\partial \mathbf{f}}{\partial t} = \sum_j \frac{\partial \boldsymbol{\chi}_j}{\partial t} \, \zeta^{\hat{\varepsilon}}(\hat{\boldsymbol{x}} - \boldsymbol{\xi}_j) \,. \tag{18}$$

What remains is to chose a $\boldsymbol{\mathcal{U}}$, such that particle core sizes in physical space are small where small scale features are present in the flow. In [1] this was accomplished by setting $\boldsymbol{\mathcal{U}}$ to be the solution of a moving mesh partial differential equation (MMPDE),

$$\boldsymbol{\mathcal{U}} = \hat{\nabla} \cdot \left( M(\hat{\boldsymbol{x}}, t) \hat{\nabla} \mathbf{f}(\hat{\boldsymbol{x}}, t)^{\mathrm{T}} \right) \,, \tag{19}$$

where $M(\hat{\boldsymbol{x}}, t)$ is a so-called monitor function: a positive measure which takes great values where numerical resolution should be increased, *e.g.*

$$M(\hat{\boldsymbol{x}}, t) = \sqrt{1 + \alpha |\mathcal{B} \hat{q}|^2} \,, \tag{20}$$

$\mathcal{B}$ being a high-pass filter. We applied this method in [1] to the evolution of an elliptical vortex governed by the 2D Euler equations. Figure 2 depicts the adaptation of the underlying grid, and thus the particle core sizes $\varepsilon(\boldsymbol{x})$.

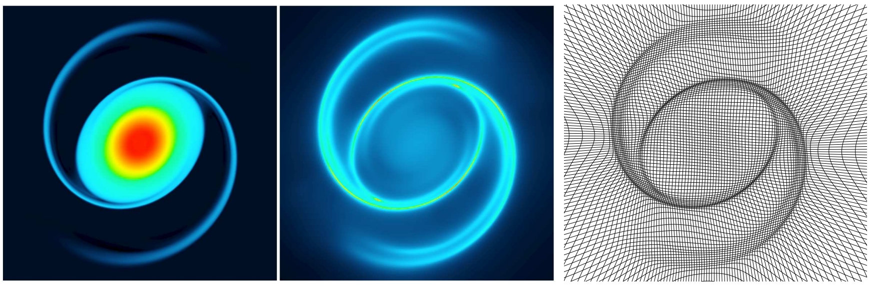


**Fig. 2.** Simulation of the evolution of an inviscid elliptical vortex using the AGM particle method: vorticity (left), particle sizes (middle, dark areas represent coarse particle sizes) and grid (right).

### 3.2 Wavelet-based multiresolution particle method

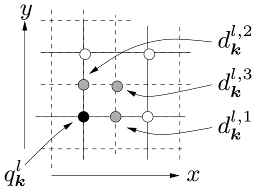
We employ a wavelet-based multiresolution analysis (MRA) using $L + 1$ levels of refinement to guide the creation of particles on the grid. The function $q(\boldsymbol{x}, t)$ can be represented as

$$q^L = \sum_{\boldsymbol{k}} q^0_{\boldsymbol{k}} \, \zeta^0_{\boldsymbol{k}} + \sum_{0 \le l < L} \sum_{\boldsymbol{k}} \boldsymbol{d}^l_{\boldsymbol{k}} \, \boldsymbol{\psi}^l_{\boldsymbol{k}} \,, \tag{21}$$

where $q^0_{\boldsymbol{k}}$ are the weights, $\zeta^0_{\boldsymbol{k}}$ are the scaling functions (or kernels) on the coarsest level, $\boldsymbol{d}^l_{\boldsymbol{k}}$ are the so-called detail coefficients and $\boldsymbol{\psi}^l_{\boldsymbol{k}}$ are the wavelets. The MRA here is based on an iterative interpolation scheme as introduced by Deslauriers and Dubuc [5], thus we do not have explicit scaling functions $\zeta$ and wavelets $\boldsymbol{\psi}$. In this scheme the scaling coefficients of two subsequent levels are related through

$$\begin{aligned} q^{l+1}_{2\boldsymbol{k}} &= q^l_{\boldsymbol{k}} \\ q^{l+1}_{2\boldsymbol{k}+1} &= d^l_{\boldsymbol{k}} + \sum_{\boldsymbol{j}} w^l_{\boldsymbol{j}-\boldsymbol{k}} \, q^l_{\boldsymbol{j}-\boldsymbol{k}} \,, \end{aligned} \tag{22}$$

where $w^l_{\boldsymbol{j}}$ are coefficients related to the polynomial interpolation of the scheme [5].



$y$
$d^{l,2}_{\boldsymbol{k}}$
$d^{l,3}_{\boldsymbol{k}}$
$d^{l,1}_{\boldsymbol{k}}$
$q^l_{\boldsymbol{k}}$
$x$

**Fig. 3.** Each detail coefficient $d^{l,m}_{\boldsymbol{k}}$, with $m = 1, \ldots, 2^d - 1$ corresponds to a specific grid point on the next higher level.

As illustrated in Figure 3 each detail coefficient is associated with a grid point on the next finer grid. Let $child(\boldsymbol{k}, m)$ be the grid point associated with $d^{l,m}_{\boldsymbol{k}}$ and let $ancs(\boldsymbol{k})$ denote the set of grid points $\boldsymbol{k}'$ needed to interpolate the value $q^l_{\boldsymbol{k}}$ from values $q^{l-1}_{\boldsymbol{k}'}$ and detail coefficients $\boldsymbol{d}^{l-1}_{\boldsymbol{k}'}$ of the next coarser level. Then an adapted grid is constructed by discarding all those grid points whose $|d^{l,m}_{\boldsymbol{k}}|$ are smaller as a prescribed threshold, i.e.

$$\mathcal{K}_> = \mathcal{K}^0 \cup \left\{ \boldsymbol{k}' = child(\boldsymbol{k}, m) \cup ancs(\boldsymbol{k}') \;\middle|\; |d^{l,m}_{\boldsymbol{k}}| > \varepsilon \,,\, l \in [0, L-1] \right\}. \tag{23}$$

Note that $\mathcal{K}^0_> \equiv \mathcal{K}^0$ and that $ancs(\boldsymbol{k}')$ are added to maintain proper nestedness of the grids (see for instance [18] for details). In order to be able to capture small scales that may emerge between two subsequent MRAs we follow the conservative approach of Liandrat and Tchamitchian [13] and additionally activate all children of the active grid points, *i.e.*

$$\boldsymbol{\mathcal{K}}^l_> \leftarrow \boldsymbol{\mathcal{K}}^l_> \cup \{\, \mathit{child}(\boldsymbol{k}, m) \ \mid\ \boldsymbol{k} \in \boldsymbol{\mathcal{K}}^{l-1}_>\,,\ m = 1, 2, 3 \,\} \ \text{ for } \ l = L, \ldots, 1\,. \tag{24}$$

Multilevel remeshing interpolates particles created on a set of grid points $\boldsymbol{\mathcal{K}}^l_>$ onto a set of grid points $\boldsymbol{\mathcal{K}}^l_\times$. This is accomplished in the following way: let $M$ denote the kernel used for remeshing the particles, then (i) horizontally extend the set of source grid points $\boldsymbol{\mathcal{K}}^l_>$ by $\boldsymbol{\mathcal{B}}^l$ , where

$$\boldsymbol{\mathcal{B}}^l = \left\{ \boldsymbol{k}' \mid \min_{\boldsymbol{k} \in \boldsymbol{\mathcal{K}}^l_>} |\boldsymbol{k}' - \boldsymbol{k}| \leq \lceil \tfrac{1}{2}\, \mathit{supp}(M) + \mathrm{LCFL} \rceil \right\}, \tag{25}$$

where the “Lagrangian CFL” LCFL $\equiv \delta t\, \|\nabla \otimes \boldsymbol{u}\|_\infty$, (ii) create particles on $\boldsymbol{\mathcal{K}}^l_> \cup \boldsymbol{\mathcal{B}}^l_1$, *i.e.*

$$Q^l_p = c^l_{\boldsymbol{k}}\, (h^l)^d\,, \qquad v^l_p = (h^l)^d\,, \qquad \boldsymbol{x}^l_p = \boldsymbol{x}^l_{\boldsymbol{k}}\,,$$

(iii) after convection, interpolate these particles onto a new set of grid points $\boldsymbol{\mathcal{K}}^l_\times$. Clearly, for consistency $\boldsymbol{\mathcal{K}}^l_\times$ cannot be chosen arbitrarily. We propose the following method: Introduce and indicator function $\chi^l$ defined as

$$\chi^l_{\boldsymbol{k}} = \begin{cases} 1\,, & \boldsymbol{k} \in \boldsymbol{\mathcal{K}}^l_> \\ 0\,, & \boldsymbol{k} \in \boldsymbol{\mathcal{B}}^l\,, \end{cases} \tag{26}$$

and convect the particles, *i.e.* solve the following set of equations

$$\frac{dQ^l_p}{dt} = \mathcal{L}(q, \boldsymbol{x}, t)\,, \qquad \frac{d\chi^l_p}{dt} = 0\,, \qquad \frac{d\boldsymbol{x}^l_p}{dt} = \boldsymbol{u}(\boldsymbol{x}^l_p, t)\,, \qquad \frac{dv^l_p}{dt} = v^l_p\ (\nabla \cdot \boldsymbol{u})\, (\boldsymbol{x}^l_p, t)\,. \tag{27}$$

The particle weights and the indicator are then interpolated onto the grid and grid points with $\tilde{\chi}^l_{\boldsymbol{k}} > 0$ are selected to consitute $\boldsymbol{\mathcal{K}}^l_\times$, where $\tilde{\chi}^l$ denotes the remeshed indicator function. Using this technique, the scale distribution $\{\boldsymbol{\mathcal{K}}^l_>\}^L_{l=0}$ is naturally convected with the flow and we obtain and adaptation mechanism which is independent of the CFL number.

To demonstrate the Lagrangian character of the adaptation we considered the convection of a passive scalar in 2D, subject to a vortical velocity field [12]. The problem involves strong deformation of a initial circular “blob” which at the end of the simulation returns to the initial condition. The remeshing function and particle kernel were both chosen as

$$W(\boldsymbol{x}) = \zeta(\boldsymbol{x}) = \prod_{l=1}^{d} M_6'''\big((\boldsymbol{x})_l\big)\,,$$

where the fourth-order accurate interpolating function $M_6'''$ is of higher order than the $M_4'$ function at the expense of a larger support. The wavelets employed were also fourth-order accurate. Figure 4 illustrates the adaptation of the grid/particles at two different times. We measure the $L^2$ and $L^\infty$ error of the

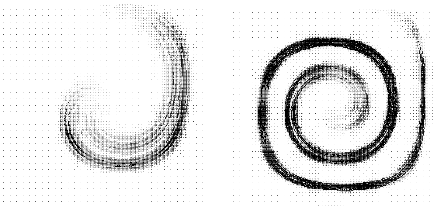


**Fig. 4.** Active grid points/particles at two different times of the simulation of a passive scalar subject to a single vortex velocity field.

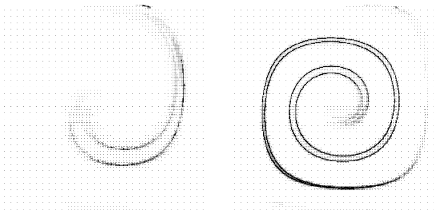


**Fig. 5.** Active grid points/particles at two different times of the simulation of a propagating interface subject to a single vortex velocity field.

final solution for different choices of $\epsilon$ and observe second order convergence, corresponding to fourth order convergence in $h$, as depicted in Figure 6. The maximum CFL measured during the course of the simulation was 40.7.

We also applied the presented method to the simulation of a propagating interface using a level set formulation. A "narrow band" formulation is easily accomplished with the present method by truncating the detail coefficients that are far from the interface. We consider the well-established 2D deformation test case which amounts to the propagation of a circle subject to the same velocity field as above. Figure 5 depicts the grid adaptation and comparing to Figure 4, one can clearly see the restriction of the refinement to a small neighborhood around the interface. We measure the error of the area encompassed by the interface at the final time and compare it against a non-adaptive particle level set method [9] and against the "hybrid particle level set method" of Enright *et al.* [7]. Figure 7 displays this comparison and we find that our adaptive approach performs favorably, which may be attributed in part to the adaptive character and in part to the high order of the method.

### 3.3 Parallelization

Recently we have developed a Parallel Particle-Mesh (PPM) software library [16] that facilitates large-scale calculations of transport and related problems using particles. The library provides the mechanisms necessary to achieve good parallel efficiency and load balancing in these situations where both meshes and particles operate as computational elements. Figure 8 and Figure 9 depict the parallel performance of the library for a Navier-Stokes solver based on the vortex method. The calculations were run on the Cray XT3 at the Swiss National Supercomputing Centre (CSCS). Our current work aims at implementing the adaptive techniques described herein into the parallel framework of the PPM library.

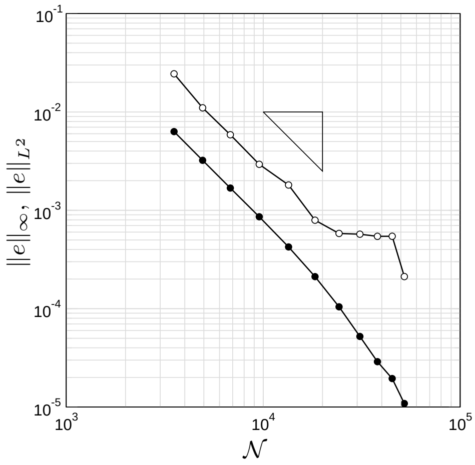


**Fig. 6.** $\varepsilon$-refinement study; the data points correspond to $\varepsilon = 2^{-p} \times 10^{-3}$ for $p = 0, \ldots, 10$. The triangle represents 2nd-order convergence. $\mathcal{N}$ is the number of active grid points/particles.

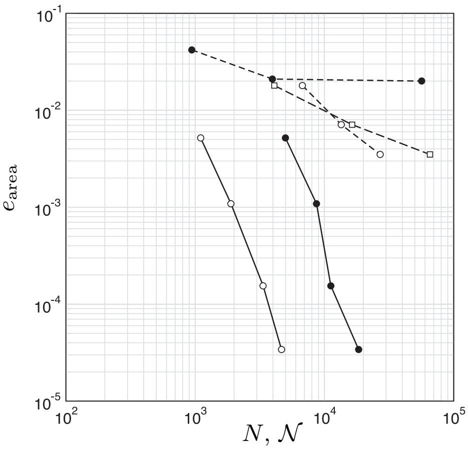


**Fig. 7.** Plot of relative error of the area enclosed by the interface against degrees of freedom: Hieber & Koumoutsakos [9] (●- -●, particles at time t=0), Enright *et al.*[7] (○- -○, auxiliary particles at time t=0 and □–□, grid points) and present method (○–○, active grid points at time t=0, ●–●, active grid points at the final time).

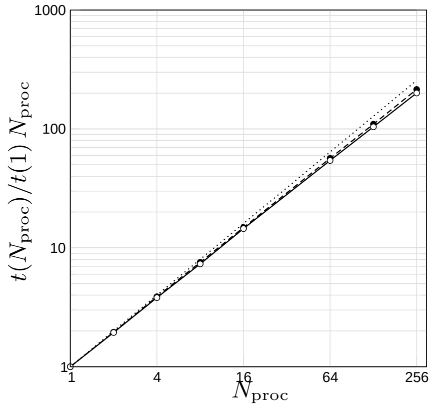


**Fig. 8.** Parallel scaling of a particle-based Navier-Stokes solver built using the PPM library. The calculations were performed on the Cray XT3, with 524,288 particles per CPU. Curves denote double precision (○–○), and single-precision (●- -●) results, respectively.

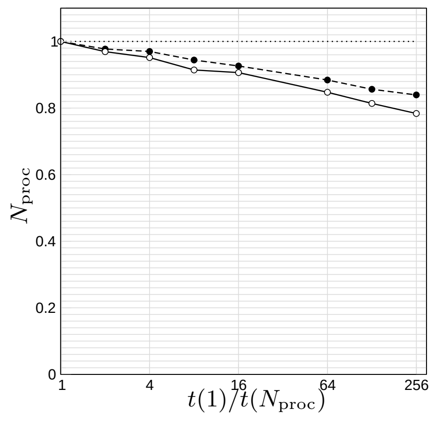


**Fig. 9.** Parallel Efficiency of a PPM-based Navier-Stokes solver. Curves denote double precision (○–○), and single-precision (●- -●) results, respectively.

## 4 Conclusions

We present multiresolution particle-mesh methods for simulating transport equations. We outline two methods introducing enhanced dynamic adaptivity and multiresolution capabilities for particle methods. The first method is based on an adaptive global mapping from a reference space to physical space for the particle locations; it has been successfully applied to the evolution of an elliptical vortex in an inviscid incompressible fluid. The second method is based on a wavelet multiresolution decomposition of the particle function representation. It is equipped with a Lagrangian adaptation mechanism that enables the simulation of transport problems and interface capturing problems independent of the CFL number. We have presented results of an interface tracking problem where the method has shown to have superior volume conservation properties. We are currently working on the application of this method to the Navier-Stokes equations.

## References

1. M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos. Multilevel adaptive particle methods for convection-diffusion equations. *Multiscale Modeling and Simulation*, 4(1):328–357, 2005.
2. G.-H. Cottet, P. Koumoutsakos, and M. L. Ould Salihi. Vortex methods with spatially varying cores. *Journal of Computational Physics*, 162:164–185, 2000.
3. G.-H. Cottet and P. Poncet. Advances in direct numerical simulations of 3d wall-bounded flows by vortex-in-cell methods. *Journal of Computational Physics*, 193:136–158, 2003.
4. P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations. part 2: The anisotropic case. *Mathematics of Computation*, 53(188):509–525, 1989.
5. G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constructive Approximation*, 5:49–68, 1989.
6. J. D. Eldredge, A. Leonard, and T. Colonius. A general deterministic treatment of derivatives in particle methods. *Journal of Computational Physics*, 180:686–709, 2002.
7. D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.
8. F. H. Harlow. Particle-in-cell computing method for fluid dynamics. *Methods in Computational Physics*, 3:319–343, 1964.
9. S. E. Hieber and P. Koumoutsakos. A Lagrangian particle level set method. *Journal of Computational Physics*, 210(1):342–367, 2005.
10. T. Y. Hou. Convergence of a variable blob vortex method for the euler and navier-stokes equations. *SIAM Journal on Numerical Analysis*, 27(6):1387–1404, 1990.
11. P. Koumoutsakos. Multiscale flow simulations using particles. *Annual Review of Fluid Mechanics*, 37(1):457–487, 2005.
12. R. J. Leveque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis*, 33(2):627–665, 1996.

13. J. Liandrat and P. Tchamitchian. Resolution of the 1D regularized burgers equation using a spatial wavelet approximation. ICASE Report 90-83, NASA Langley Research Center, 1990.
14. J. J. Monaghan. Extrapolating b-splines for interpolation. *Journal of Computational Physics*, 60:253–262, 1985.
15. P. Ploumhans and G. S. Winckelmans. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *Journal of Computational Physics*, 165:354–406, 2000.
16. I. F. Sbalzarini, J. H. Walther, M. Bergdorf, S. E. Hieber, E. M. Kotsalis, and P. Koumoutsakos. PPM – a highly efficient parallel particle-mesh library. *Journal of Computational Physics*, 215(2):566–588, 2006.
17. A.-K. Tornberg and B. Engquist. Numerical approximations of singular source terms in differential equations. *Journal of Computational Physics*, 200:462–488, 2004.
18. O. V. Vasilyev. Solving multi-dimensional evolution problems with localized structures using second-generation wavelets. *International Journal of Computational Fluid Dynamics*, 17(2):151–168, 2003.
19. J. H. Walther and P. Koumoutsakos. Three-dimensional particle methods for particle laden flows with two-way coupling. *Journal of Computational Physics*, 167:39–71, 2001.
20. Q. X. Wang. Variable order revised binary treecode. *Journal of Computational Physics*, 200(1):192 –210, 2004.