Journal of Computational Physics 252 (2013) 458–467

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A high order solver for the unbounded Poisson equation

Mads Mølholm Hejlesen [a], Johannes Tophøj Rasmussen [a], Philippe Chatelain [b], Jens Honoré Walther [a,c,*]

[a] Department of Mechanical Engineering, Technical University of Denmark, Building 403, DK-2800 Kgs. Lyngby, Denmark
[b] Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain, B-1348, Belgium
[c] Computational Science and Engineering Laboratory, ETH Zürich, Clasiusstrasse 33, CH-8092 Zürich, Switzerland

ARTICLE INFO

Article history:
Received 21 July 2012
Received in revised form 24 February 2013
Accepted 31 May 2013
Available online 27 June 2013

Keywords:
Poisson solver
Elliptic solver
Unbounded domain
Infinite domain
Isolated system
Green's function solution
Numerical integration
Vortex methods
Particle-mesh methods

ABSTRACT

A high order converging Poisson solver is presented, based on the Green's function solution to Poisson's equation subject to free-space boundary conditions. The high order convergence is achieved by formulating regularised integration kernels, analogous to a smoothing of the solution field. The method is extended to directly solve the derivatives of the solution to Poisson's equation. In this way differential operators such as the divergence or curl of the solution field can be solved to the same high order convergence without additional computational effort. The method, is applied and validated, however not restricted, to the equations of fluid mechanics, and can be used in many applications to solve Poisson's equation on a rectangular unbounded domain.

1. Introduction

The solution of Poisson's equation on an unbounded domain is essential in many physical problems and appears among others in the field of fluid mechanics [1], molecular dynamics [2], and astrophysics [3]. The Poisson equation often describes the energy or potential of a physical system, and is either directly or indirectly governing the dynamics of the system. Particle Methods, techniques that have been extensively applied and validated in the past two decades [4–6], are typically used for simulating such problems.

Particle methods generally provide a solution to Poisson's equation by applying a Green's function solution in the form of the Biot–Savart equation [7] where each particle exerts a force onto every other particle in the system. The direct evaluation of the mutual particle interaction is an N_p-body problem which scales as $\mathcal{O}(N_p^2)$, and tree algorithms such as the Barnes–Hut [8] or the fast multipole method (FMM) [9] have been used to improve the computational scaling of particle methods. The FMM is capable of achieving an optimal scaling of $\mathcal{O}(N_p)$ [9], however, it is subject to a large pre-factor when constructing and traversing the tree data structure. This reduces the efficiency of the FMM and makes it less attractive, particularly for three-dimensional simulations where the translation of multipole expansions becomes expensive. Recent development in large-scale particle methods, such as fluid dynamics applications, has been focused on hybrid particle-mesh methods such as the Vortex-In-Cell (VIC) method by Christiansen [10]. Through interpolation schemes the VIC method applies a mixed

* Corresponding author at: Department of Mechanical Engineering, Technical University of Denmark, Building 403, DK-2800 Kgs. Lyngby, Denmark. Tel.: + 45 4525 4327; fax: + 45 4588 4325.
E-mail address: jhw@mek.dtu.dk (J.H. Walther).

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.05.050

particle-mesh discretisation of the governing equations and thereby exploits the advantages of both a Lagrangian and an Eulerian frame of reference. Specifically, the vorticity–velocity relation which leads to a Poisson equation, is handled on a uniform mesh. This enables the use of highly efficient fast Fourier transform (FFT) based Poisson solvers that offer a direct solution at a reduced computational cost relative to most iterative solvers. Given the right conditions, the Poisson solver on a square/cubic domain has the capability to scale as $\mathcal{O}(dN^d \log(N))$ where N is the number of grid points in each direction, and d is the number of dimensions.

There are two main strategies for using FFTs to solve Poisson's equation. The first is to exploit the sine/cosine formulated eigenvalues of a finite difference approximation to Poisson's equation, subject to either Dirichlet or Neumann boundary conditions. The order of convergence in this method depends on the finite difference scheme and on the accurate determination of boundary conditions. Methods for accurately determining Dirichlet boundary conditions that corresponds to free-space boundary conditions have been investigated by James [11], Serafini et al. [12] Colonius and Taira [13], and Cocle et al. [14].

The second strategy, which is the focus of the presented work, is to construct the Green's function solution through an integral formulation carried out in a discrete sense by convolving the field with an integration kernel. The convolution is performed efficiently in the Fourier domain by multiplying the discrete Fourier coefficients of the field with the Fourier coefficients of the integration kernel. In a periodic domain, spectral differentiating is obtained in Fourier space simply by multiplying the coefficients with their respective wavenumber. Furthermore, the periodic boundary condition is naturally treated by the periodicity of the Fourier series and a spectral convergence is therefore easily obtained as shown by Rasmussen [15]. For the unbounded free-space problem, the integration kernel cannot be represented exactly in Fourier space due to the inherent periodicity of the Fourier transform. On the same ground the domain must be extended to twice the size, to avoid the periodicity of the discrete convolution operator. Hockney and Eastwood [4] showed that it is possible to impose free-space boundary conditions by zero-padding the vorticity field and convolving this with a free-space integration kernel of equal size.

At first sight, extending the domain size to $2N$ in each unbounded direction increases the computational scaling to $\mathcal{O}(2^d dN^d \log(2N))$ and requires an allocated memory of $(2N)^d$ for a fully unbounded domain. However, an FFT based convolution algorithm, proposed by [4], significantly reduces the computational scaling and the memory needed by exploiting the fact that a multidimensional Fourier transform consists of series of one-dimensional transforms carried out in each direction, sequentially.

The method of [4] is widely used in the field of particle-mesh methods and has been extended in recent work. Mixed periodic and free-space boundary conditions were obtained by Chatelain and Koumoutsakos [16] who solved the extended Poisson equation in the free-space directions leading to a Helmholtz type equation in the periodic directions. A multi-resolution method was developed by Rasmussen et al. [17]. This method exploits the linearity of the problem to obtain local mesh refinement by applying rectangular patches using a buffer-zone to ensure a continuous solution and is similar to the method of Villumsen [18].

The convolution of the density field and the integral kernels are inherently based on a mid-point numerical quadrature and is therefore limited to a second order convergence as shown by [15]. A high order converging, Green's function based Poisson solver has, to this point, not been presented in the literature. Qiang [19] however, used a similar convolution integral method to find the electric potential of a charge density field and reached $\mathcal{O}(h^4)$ convergence by enforcing a quadrature rule correction onto the charge density field. Here, h denotes the discretisation length such as the mesh cell size or particle spacing. Qiang applied the method to cases where the integration kernel is non-singular and does not address the problems of singular integration kernels, which are the most common when solving Poisson's equation by a Green's function.

In the present work the Green's function based Poisson solver is extended to achieve a high order of convergence of the convolution integral for a continuous field by formulating regularised integration kernels. These are based on the work of [20–24] using smoothed particles to achieve high order regularised integration kernels for mesh-free vortex methods. However, unlike mesh-free methods, the mesh-based Poisson solver requires a finite and continuous integration kernel to ensure an accurate numerical solution. Therefore, a great effort is put into defining the centre value of the regularised integration kernels as this is identified, in the present study, as a crucial factor for achieving high order convergence.

As mentioned, Poisson's equation is often encountered when solving a scalar or a vector field potential. The potential is introduced to simplify the mathematical problem, however, it is often the scalar or the vector field itself which is the essential value. On this ground, the presented method is extended to directly calculate the derivatives of the solution of Poisson's equation to an equally high order as the Poisson solver. Further, the evaluation of differential operators such as the gradient, divergence, or the curl of the solution is discussed with consideration to the lowest computational cost.

2. Vorticity–velocity equations and Fourier-based convolution for their solution

In fluid mechanics, the kinematics is defined through the vorticity–velocity equations. The vorticity $\boldsymbol{\omega}$ is defined as the curl of the velocity $\boldsymbol{u}$:

$$\boldsymbol{\omega} \equiv \nabla \times \boldsymbol{u}. \tag{1}$$

Eq. (1) can be inverted to find $\boldsymbol{u}$ by defining a solenoidal vector potential ($\boldsymbol{\psi}$: $\nabla \cdot \boldsymbol{\psi} = 0$) referred to as the stream function

$$\boldsymbol{u} \equiv \nabla \times \boldsymbol{\psi}. \tag{2}$$

This leads to Poisson equations for the stream function

$$\nabla^2 \boldsymbol{\psi} = -\boldsymbol{\omega} \tag{3}$$

and the velocity field

$$\nabla^2 \boldsymbol{u} = -\nabla \times \boldsymbol{\omega}. \tag{4}$$

Using a Green's function approach, Poisson's equation (Eq. (3) or Eq. (4)) can be solved on the unbounded domain $\mathbb{R}^d$ through the convolutions

$$\boldsymbol{\psi}(\boldsymbol{x}) = \int_{\mathbb{R}^d} G(\boldsymbol{x}-\boldsymbol{x}')\boldsymbol{\omega}(\boldsymbol{x}')\,d\boldsymbol{x}' \quad \text{or} \quad \boldsymbol{u}(\boldsymbol{x}) = \int_{\mathbb{R}^d} \boldsymbol{K}(\boldsymbol{x}-\boldsymbol{x}')\boldsymbol{\omega}(\boldsymbol{x}')\,d\boldsymbol{x}' \tag{5}$$

with the Green's functions

$$G(\boldsymbol{x}) = \begin{cases} -\frac{1}{2\pi}\log|\boldsymbol{x}| & \text{in 2D,} \\ \frac{1}{4\pi|\boldsymbol{x}|} & \text{in 3D,} \end{cases} \tag{6}$$

where $|\boldsymbol{x}|$ denotes the length of $\boldsymbol{x}$. The corresponding velocity kernels $\boldsymbol{K}(\boldsymbol{x}) = \nabla G(\boldsymbol{x})\times$ are given by

$$\boldsymbol{K}(\boldsymbol{x}) = \begin{cases} -\frac{1}{2\pi|\boldsymbol{x}|^2}\boldsymbol{x}\times & \text{in 2D,} \\ -\frac{1}{4\pi|\boldsymbol{x}|^3}\boldsymbol{x}\times & \text{in 3D,} \end{cases} \quad \text{where } \boldsymbol{x}\times = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}. \tag{7}$$

Eqs. (5) can be solved in an unbounded, discrete sense by the discrete linear convolution of the field values of the integration kernel and the vorticity field

$$\boldsymbol{\psi} = G * \boldsymbol{\omega} \quad \text{or} \quad \boldsymbol{u} = \boldsymbol{K} * \boldsymbol{\omega}. \tag{8}$$

Here $*$ denotes the discrete linear convolution operator which is obtained by zero-padding the vorticity field to twice the domain size and applying the discrete cyclic convolution operator to the now zero-padded vorticity field and the integration kernel of equal size. The convolution is performed efficiently in Fourier space whereas the linearity of the problem also enables the curl of Eq. (4) to be carried out in Fourier space, thus reducing the number of FFTs needed. The convolution integral equations in Fourier space are written as

$$\widehat{\boldsymbol{\psi}} = \widehat{G}\widehat{\boldsymbol{\omega}} \quad \text{or} \quad \widehat{\boldsymbol{u}} = \widehat{\boldsymbol{K}}\widehat{\boldsymbol{\omega}}, \tag{9}$$

where $\widehat{\cdot}$ denotes the Fourier transform. For the 2D case $\boldsymbol{\omega} = \{0, 0, \omega_3\}$ and $\boldsymbol{\psi} = \{0, 0, \psi_3\}$. This reduces Eq. (9) to one equation for $\boldsymbol{\psi}$ and two equations for $\boldsymbol{u}$.

The domain doubling technique has a severe impact on the convolution footprint, however it can be reduced from $(2N)^d$ to $2N^d + (2N)^{d-1}$. Hockney and Eastwood [4] propose to carry out a transform in one direction and then proceed plane by plane i.e. Fourier-transform along the remaining dimension(s), multiply by $\widehat{G}$ and inverse transform. This method, however, still requires the integration kernels G or $\boldsymbol{K}$ to be calculated initially and stored in their full size $(2N)^d$. It is evident, that for solving the velocity directly in Eq. (9) a number of d kernel functions are needed instead of the one needed for solving the stream function. Alternatively, the curl operator can be implemented by spectral differentiating such that

$$\widehat{\boldsymbol{u}} = \iota\boldsymbol{k} \times \widehat{\boldsymbol{\psi}}, \tag{10}$$

where ι is the imaginary unit and $\boldsymbol{k} = \{k_x, k_y, k_z\}$ is the wavenumber corresponding to the Fourier coefficients.

3. High order methods: Regularised integration kernels

We now look into the construction of the integration kernels and how to derive high order integration kernels by regularising the given scalar or vector field. By convolving Poisson's equation (Eq. (3)) with a kernel function G and exploring the linear properties of convolution we arrive at the Green's function equation

$$\nabla^2 G(\boldsymbol{x}) = -\delta(\boldsymbol{x}-\boldsymbol{x}'), \tag{11}$$

where $\delta(\boldsymbol{x}-\boldsymbol{x}')$ is the Dirac delta function. The delta function causes the integration kernel G to be singular at $|\boldsymbol{x}| = 0$ as seen in Eq. (6). The singularity of the kernels poses two problems for representing the kernel functions in the Fourier domain. First, in order to represent a singularity in Fourier space an infinite number of Fourier coefficients are needed. Due to the discretisation of the domain in the physical space $\boldsymbol{x}$, the Fourier space, represented by the wavenumbers $k_i = \{-k_s/2, \ldots, k_s/2\}$, is bounded and can therefore only approximate a singularity by a finite estimation. Here $k_s = 1/h$ is the sampling wavenumber and $k_s/2$ is the highest resolved wavenumber due to the Nyquist–Shannon sampling theorem.

Attempts of defining a finite value to replace the singularity $G(\mathbf{0})$ have been presented in [4,16,15] however no better than $\mathcal{O}(h^2)$ convergence is reported. Secondly, the singularity of the kernel functions introduces a discontinuity to the integration kernel and thereby reduces the accuracy of the Fourier transform. Generally, the error of a Fourier series estimation of a function has an upper bound which depends on the number of continuous derivatives of the function [25].

The singularity of the Green's function at the origin can be handled in several ways. A first technique [16] consists in replacing the mid-point quadrature by an exact formula for the integral of the singularity over the origin cell. We note that, while this suggests at spreading the treatment of the singularity to all the cells, it does not however change the rate of convergence of the method.

In the present work, we handle the singularity and achieve higher order by using smoothing functions. Effectively, a smoothed version of the right-hand side of Poisson's equation is considered

$$\boldsymbol{\omega}_\epsilon = \zeta_\epsilon * \boldsymbol{\omega}, \tag{12}$$

where ζ_ϵ is a radially-symmetric filter function with a smoothing radius ϵ. For conservation properties we have $\int_{\mathbb{R}^d} \zeta_\epsilon(\boldsymbol{x})\, d\boldsymbol{x} = 1$.

The application of this filter causes an error which is of the order

$$\|\boldsymbol{\omega}_\epsilon - \boldsymbol{\omega}\| < C_1 \epsilon^m \|\boldsymbol{\omega}\|, \tag{13}$$

where m is the order of the filter, i.e. the number of satisfied moments. An m-order accurate radially-symmetric filter will satisfy

$$0^\beta = \begin{cases} 2\pi \int_0^\infty r^\beta \zeta_\epsilon(r) r\, dr & \text{in 2D,} \\ 4\pi \int_0^\infty r^\beta \zeta_\epsilon(r) r^2\, dr & \text{in 3D,} \end{cases} \tag{14}$$

where $\beta = \{0, 2, \ldots, m-2\}$. We note that m is an even number since the moment conditions for odd integers β are automatically satisfied due to the symmetry of the kernels.

The convolution of this smoothed source term with the Green's function can be written as

$$\boldsymbol{\psi}_\epsilon = G * (\zeta_\epsilon * \boldsymbol{\omega}) = (G * \zeta_\epsilon) * \boldsymbol{\omega} = G_\epsilon * \boldsymbol{\omega}. \tag{15}$$

The substitution of G by its mollified version G_ϵ thus leads to an error which, similarly to the right-hand side, is governed by the order of the filter [26–29]. For the stream function the error is bounded by

$$\|\boldsymbol{\psi}_\epsilon - \boldsymbol{\psi}\| < C_2 \epsilon^m \|\boldsymbol{\omega}\|. \tag{16}$$

Note that the discretisation of the convolution above leads to a second error contribution. It is due to the underlying quadrature of the integral at the grid points

$$\left\|\boldsymbol{\psi}_\epsilon - \boldsymbol{\psi}_\epsilon^h\right\| < C_3 \epsilon \left(\frac{h}{\epsilon}\right)^l \|\boldsymbol{\omega}\|, \tag{17}$$

where l depends on the smoothness of the smoothing function ζ_ϵ. Error bounds can be estimated in a similar fashion for the velocity field

$$\|\boldsymbol{u}_\epsilon - \boldsymbol{u}\| < C_4 \epsilon^m \|\boldsymbol{\omega}\|, \tag{18}$$

$$\left\|\boldsymbol{u}_\epsilon - \boldsymbol{u}_\epsilon^h\right\| < C_5 \epsilon^2 \left(\frac{h}{\epsilon}\right)^l \|\boldsymbol{\omega}\|, \tag{19}$$

where one gains an order due to the null integral of the velocity kernel.

This is a well-known behaviour in particle methods [26–29,5]: the particle kernel must overlap other particle locations. In the present context, the regularisation kernel must be resolved by enough grid points. As it will be shown in Section 5, this is not without consequences for our problem.

For a given filter function ζ_ϵ, it is straightforward to derive the corresponding smoothed G_ϵ. For simplicity this is performed in polar (r, θ) and spherical (r, θ, ϕ) coordinates for 2D and 3D, respectively. The smoothed Green's function equation analogous to Eq. (11) is given by

$$-\zeta_\epsilon(r) = \nabla^2 G_\epsilon(r) = \begin{cases} \frac{1}{r}\frac{d}{dr}\left(r \frac{dG_\epsilon}{dr}\right) & \text{in 2D,} \\ \frac{1}{r^2}\frac{d}{dr}\left(r^2 \frac{dG_\epsilon}{dr}\right) & \text{in 3D.} \end{cases} \tag{20}$$

A weight function q_ϵ is defined from the smoothing function as

$$q_\epsilon(r) = \int_0^r \zeta_\epsilon(t)\, dt = \begin{cases} 2\pi \int_0^r \zeta_\epsilon(t) t\, dt & \text{in 2D,} \\ 4\pi \int_0^r \zeta_\epsilon(t) t^2\, dt & \text{in 3D,} \end{cases} \tag{21}$$

whereas combining Eqs. (20) and (21) renders an expression for the regularised integration kernel

$$G_\epsilon(r) = \begin{cases} -\frac{1}{2\pi}\int \frac{q_\epsilon(r)}{r}\,dr & \text{in 2D,} \\ -\frac{1}{4\pi}\int \frac{q_\epsilon(r)}{r^2}\,dr & \text{in 3D.} \end{cases} \tag{22}$$

The construction of an m-order kernel thus requires an m-order smoothing function. In addition to the moment conditions above, this smoothing function will also have to exhibit high order continuity as well. This will guarantee a fast relaxation of G_ϵ to the unfiltered G for radii larger than ϵ. This continuity can be obtained by working with kernels based on Gaussians.

Evidently, in order to obtain an m-order integration kernel the smoothing function ζ_ϵ must satisfy $m/2$ equations. Further it is required that the Fourier transform of the smoothing function $\widehat{\zeta_\epsilon}$ is smooth and the weight function has a rapid convergence of $w(r) \to 1$ for $r \to \infty$. We construct the smoothing functions by considering the function of the dimensionless radius $\rho = r/\epsilon$

$$\zeta_\epsilon(r) = \frac{1}{\epsilon^d}\zeta\left(\frac{r}{\epsilon}\right), \tag{23}$$

and use a Gaussian kernel cf. [20–22,24] of the form

$$\zeta(\rho) = P_m(\rho)\exp\left(\frac{-\rho^2}{2}\right) \quad \text{where } P_m(\rho) = a_1 + a_2\rho^2 + a_3\rho^4 + \cdots + a_{m/2}\rho^{m-2}. \tag{24}$$

P_m is an even polynomial of order $m-2$ that supplies the $m/2$ degrees of freedom needed to satisfy the $m/2$ moment equations of Eq. (14). The coefficients $a_1, a_2, \ldots, a_{m/2}$ can be calculated by solving the system of equations consisting of the $m/2$ equations for the m moments.

Alternatively, one can follow the method of Chatelain and Leonard [30] and construct the kernels iteratively. They derived the generic extrapolation formula to increase the order of an interpolation kernel in the context of particle methods. An interpolation kernel $\tilde{W}$ of order $m+1$ is then obtained by recombining an m-order scheme W with its gradient

$$\tilde{W}(\boldsymbol{x}) = \left(1 + \frac{d}{m}\right)W(\boldsymbol{x}) + \frac{1}{m}\boldsymbol{x}\cdot\nabla W(\boldsymbol{x}), \tag{25}$$

under some conditions of smoothness for W (we refer to [30] for details). If W is built upon a radial function $\zeta_\epsilon(r)$, this equation becomes

$$\tilde{W}(\boldsymbol{x}) = \tilde{\zeta}_\epsilon(|\boldsymbol{x}|) = \left(1 + \frac{d}{m}\right)\zeta_\epsilon(|\boldsymbol{x}|) + \frac{1}{m}|\boldsymbol{x}|\frac{d\zeta_\epsilon}{dr}(|\boldsymbol{x}|). \tag{26}$$

The kernels of the form of Eq. (24) satisfy the required continuity conditions and their radial symmetry leads to the cancellation of odd order moments and error terms. The substitution in Eq. (26) then leads to the following recursive formula for the polynomials $P_m(\rho)$

$$P_{m+2}(\rho) = \left(1 + \frac{d}{m} - \frac{1}{m}\rho^2\right)P_m(\rho) + \frac{1}{m}\rho\frac{\partial P_m}{\partial\rho}(\rho). \tag{27}$$

For the Gaussian smoothing function of Eq. (24) the mollified Green's function kernel G_ϵ takes the general form

$$G_\epsilon(\boldsymbol{x}) = \begin{cases} -\frac{1}{2\pi}\left(\log(|\boldsymbol{x}|) - Q_m\left(\frac{|\boldsymbol{x}|}{\epsilon}\right)\exp\left(\frac{-|\boldsymbol{x}|^2}{2\epsilon^2}\right) + \frac{1}{2}\mathrm{E}_1\left(\frac{|\boldsymbol{x}|^2}{2\epsilon^2}\right)\right) & \text{in 2D,} \\ \frac{1}{4\pi|\boldsymbol{x}|}\left(\frac{1}{\sqrt{2\pi}}R_m\left(\frac{|\boldsymbol{x}|}{\epsilon}\right)\exp\left(\frac{-|\boldsymbol{x}|^2}{2\epsilon^2}\right) + \mathrm{erf}\left(\frac{|\boldsymbol{x}|}{\sqrt{2}\epsilon}\right)\right) & \text{in 3D,} \end{cases} \tag{28}$$

where Q_m and R_m are polynomials given in Appendix A, $\mathrm{E}_1(z)$ is the first exponential integral function and $\mathrm{erf}(z)$ is the regular error function given by

$$\mathrm{E}_1(z) = \int_1^\infty \frac{\exp(-z\rho)}{\rho}\,d\rho \quad \text{and} \quad \mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z \exp\left(-\rho^2\right)d\rho. \tag{29}$$

In order to define $G_\epsilon(\mathbf{0})$ the exponential integral function $\mathrm{E}_1(z)$ can be expanded to an infinite series [31] by

$$\mathrm{E}_1(z) = -\gamma - \log(z) - \sum_{n=1}^{\infty}\frac{(-1)^n z^n}{n\,n!}, \tag{30}$$

where $\gamma = 0.5772156649$ is Euler's constant. The finite value of $G_\epsilon(\mathbf{0})$ is now defined as

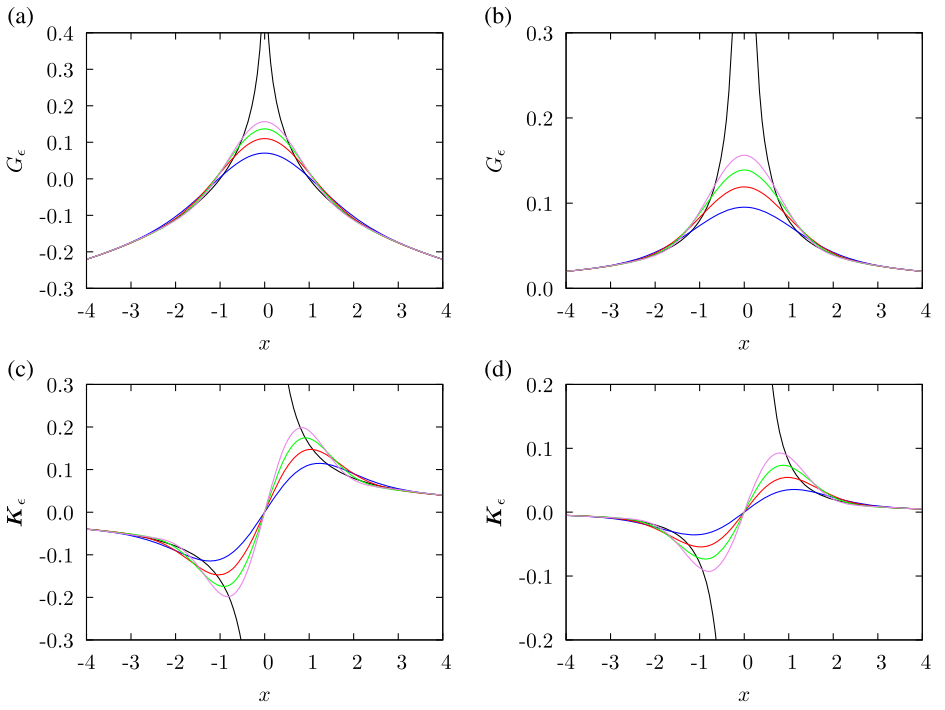

Fig. 1. Regularised integration kernels G_ϵ and $\boldsymbol{K}_\epsilon$ with $\epsilon = 1$ compared to the non-regularised kernels. (a) and (b) show the 2D and 3D integration kernels, respectively. The corresponding velocity kernels is shown in (c) for 2D and (d) for 3D. (———): G and $\boldsymbol{K}$. (———): $m = 4$; (———): $m = 6$; (———): $m = 8$; (———): $m = 10$.

$$G_\epsilon(0) = \lim_{x\to 0}(G_\epsilon) = \begin{cases} \frac{1}{2\pi}\left(\frac{\gamma}{2} - \log(\sqrt{2}\epsilon) + Q_m(0)\right) & \text{in 2D,} \\ \frac{\sqrt{2}}{8\pi^{3/2}}\left(\frac{R_m(\frac{|\boldsymbol{x}|}{\epsilon})}{|\boldsymbol{x}|}\right)_{x=0} + \frac{\sqrt{2}}{4\pi^{3/2}\epsilon} & \text{in 3D.} \end{cases} \tag{31}$$

From the relation $\boldsymbol{K}(\boldsymbol{x}) = \nabla G(\boldsymbol{x})\times$ the corresponding velocity kernels $\boldsymbol{K}_\epsilon$ of m-th order are now easily found. The derived integration kernels G_ϵ and $\boldsymbol{K}_\epsilon$ up to $m = 10$ are listed in Appendix A and shown in Fig. 1 for 2D and 3D with $\epsilon = 1$. As seen, the regularised kernel functions are all smoothly converging towards a finite value at $|\boldsymbol{x}| = 0$ and towards the non-regularised kernel functions G and $\boldsymbol{K}$ for $|\boldsymbol{x}| \to \infty$. The derived integration kernels are consistent with the kernel functions presented in [20–22,24] and have been extended to higher order in Appendix A. Furthermore, $G_\epsilon(0)$ has been defined which allows for the integration kernels to be implemented in the FFT-based Poisson solver presented in Section 2.

4. Validation: Compact vortex blob

To investigate the convergence of the free-space Poisson solver, the test function, i.e. the vorticity distribution, must be enclosed within the computational domain, and have at least two continuous derivatives. This is a consequence of the mapping of the Laplace operator $\nabla^2 : C^k \to C^{k-2}$ whereas $k \geqslant 2$ is needed to give a continuous solution. These criteria are fulfilled by using a compact and sufficiently smooth vortex blob to form a vorticity patch in 2D and a torus shaped vortex ring in 3D. Commonly used compact vorticity patches in 2D are those presented by Beale and Majda [21] and Perlman [22] which has the general form

$$\omega(r) = \begin{cases} (1-r^2)^k & \text{for } r \leqslant 1, \\ 0 & \text{for } r > 1, \end{cases} \tag{32}$$

resulting in a test function is of class C^{k-1}. The radial profile of the vorticity and the corresponding velocity distributions for the patch where $k = 3$ [21] are shown in Fig. 2(a). It is noted in Eq. (32) that this class of patches has a limited number of continuous derivatives, and the highest achievable rate of convergence is therefore limited. This is seen in Fig. 2(b) where the normalised root-mean-squared (rms) error of the calculated velocity field is plotted as a function of the spatial resolution h. As seen, the rate of convergence does not exceed $\mathcal{O}(h^4)$ when using an integration kernel of higher order. However, it is noticed that the error is reduced when using an integration kernel of an order that exceeds the smoothness of the test function.

Besides only allowing a limited rate of convergence, the stream function is not easily obtained from Eq. (32) which disables the testing of the pure Poisson equation (Eq. (3)) when using the vorticity patches of Eq. (32). To circumvent these

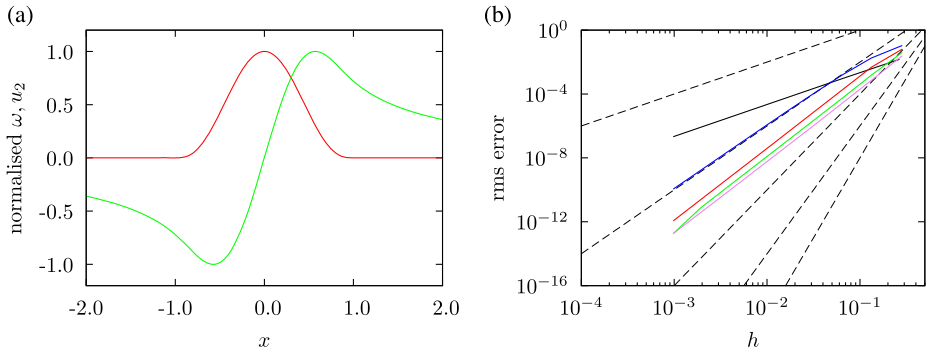

Fig. 2. (a) The normalised vorticity profile and the corresponding velocity distribution for the 2D vorticity patch of Eq. (32) with $k = 3$ at $(x, y) = (\cdot, 0)$. ( ): Vorticity $\omega/\max(\omega)$; ( ): Velocity $u_2/\max(u_2)$. (b) The normalised rms error of the resulting velocity field calculated by high order integration kernels with $\epsilon = 2h$. The error is virtually identical when calculated by the velocity kernels $\boldsymbol{K}_\epsilon$ in Eq. (9) as when using the integration kernel G_ϵ with spectral differentiating in Eq. (10). ( ): from top $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$, $\mathcal{O}(h^6)$, $\mathcal{O}(h^8)$, $\mathcal{O}(h^{10})$; ( ): non-regularised kernels G and $\boldsymbol{K}$; ( ): $m = 4$; ( ): $m = 6$; ( ): $m = 8$; ( ): $m = 10$.

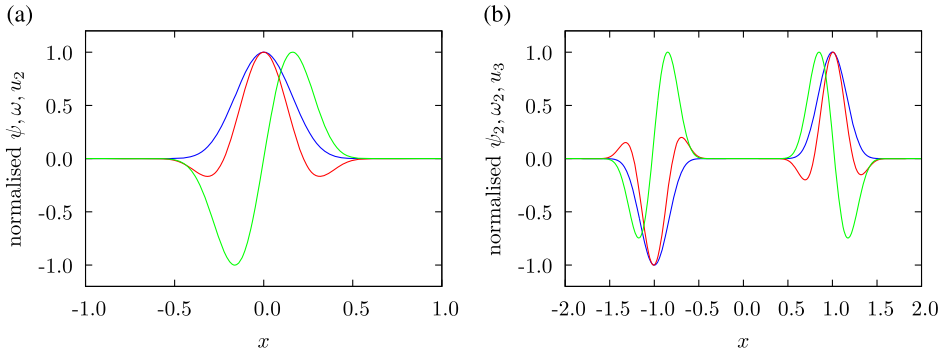

Fig. 3. The normalised test function and the corresponding solution for $c = 20$ and $R = 1$: (a) shows the 2D vorticity patch at $(x, y) = (\cdot, 0)$ and (b) shows the 3D vortex ring at $(x, y, z) = (\cdot, 0, 0)$. ( ): vorticity $\omega/\max(\omega)$; ( ): stream function $\psi/\max(\psi)$; ( ): velocity $u_i/\max(u_i)$.

test restrictions, we propose the use of a bump function distribution which is of class C^∞ i.e. it has an infinite number of continuous derivatives. The bump function is defined as

$$f(t) = \begin{cases} \exp(-\frac{c}{1-t^2}) & \text{for } |t| < 1, \\ 0 & \text{for } |t| \geqslant 1, \end{cases} \tag{33}$$

where $c = 20$ is an arbitrary positive constant. Using centred polar (r, θ) and cylindrical (r, θ, z) coordinates for the 2D and 3D case, respectively, the stream functions are defined as

$$\psi = f\left(\frac{r}{R}\right) \quad \text{in 2D}, \qquad \boldsymbol{\psi} = f\left(\frac{\sqrt{(r-R)^2 + z^2}}{R}\right)\boldsymbol{e}_\theta \quad \text{in 3D}, \tag{34}$$

where R is the radius of the centre of the vorticity patch and $\boldsymbol{e}_\theta$ is the azimuthal normal vector. The initial normalised vorticity field and the resulting normalised velocity field are found analytically by Eq. (3) and Eq. (2), respectively, and is shown in Fig. 3.

The rms error of the 2D and 3D bump function test cases are shown in Fig. 4 for the regularised integration kernels G_ϵ and $\boldsymbol{K}_\epsilon$ with $m = 4, 6, 8, 10$. As seen, the 2D and the 3D convergence are virtually identical and the convergence rate corresponds to the respective design of the integration kernels. The same convergence rate is observed with the maximum error (not shown). Further, the solution of the velocity field is found to be practically identical, and with the same convergence properties, when using the velocity kernels $\boldsymbol{K}_\epsilon$ in Eq. (9) as when using the spectral differentiating in Eq. (10). Therefore, as Eq. (10) only uses a single integration kernel instead of the d kernels needed in Eq. (9), the method of spectral differentiation is considered the optimal choice in order to reduce the memory which is allocated for the integration kernels.

5. Influence of the smoothing radius ϵ

It is seen from Eqs. (16) and (19) that introducing a smoothing to the right-hand side of Poisson's equation induces a mollification error which is proportional to the size of the smoothing radius. The quadrature error of Eqs. (17) and (19)

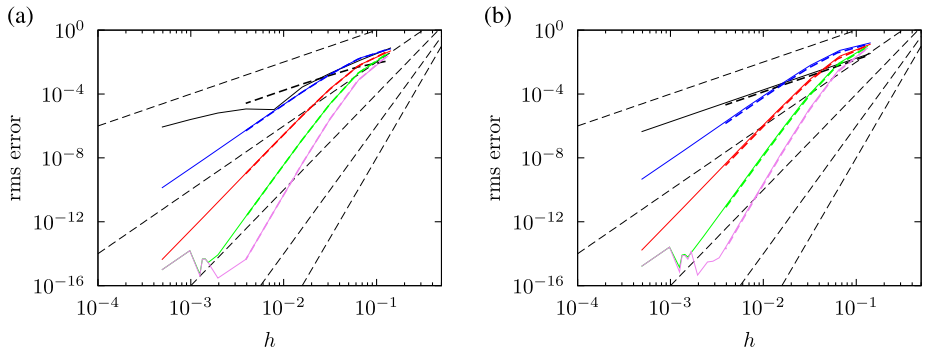

Fig. 4. The rms error of the calculated solutions of the bump function test case using regularised integration kernels G_ϵ and $\boldsymbol{K}_\epsilon$ with $\epsilon = 2h$. (a) The calculated stream function, (b) the calculated velocity field. The error for the velocity in (b) is virtually identical when calculated by the velocity kernels $\boldsymbol{K}_\epsilon$ in Eq. (9) as when using the integration kernel G_ϵ with spectral differentiating in Eq. (10). (– – – –): from top $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$, $\mathcal{O}(h^6)$, $\mathcal{O}(h^8)$, $\mathcal{O}(h^{10})$; (———): non-regularised kernels G and $\boldsymbol{K}$; (———): $m = 4$; (———): $m = 6$; (———): $m = 8$; (———): $m = 10$. Normal line: 2D; bold line: 3D.

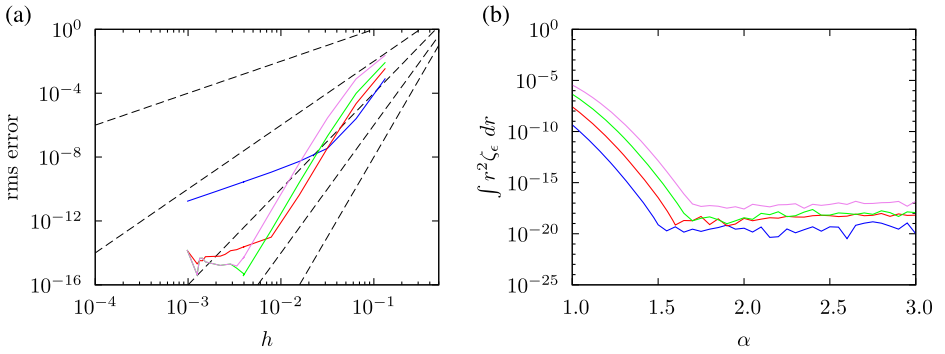

Fig. 5. (a) The rms error of the G_ϵ integration kernel with $m = 10$ for $\epsilon = \alpha h$ using different values of α. (– – – –): from top $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$, $\mathcal{O}(h^6)$, $\mathcal{O}(h^8)$, $\mathcal{O}(h^{10})$; (———): $\alpha = 1.0$; (———): $\alpha = 1.25$; (———): $\alpha = 1.5$; (———): $\alpha = 1.75$. (b) The calculated discrete moments of the smoothing function ζ_ϵ with $\beta = 2$ using different order kernels. (———): $m = 4$. (———): $m = 8$; (———): $m = 12$; (———): $m = 16$.

demands that the smoothing radius ϵ is resolved on the grid i.e. $\epsilon > h$. In the previous section, our validation relied on simply keeping $\epsilon = \alpha h$ where $\alpha = 2$, was chosen. A more rigorous approach, which is coherent with the error behaviour of the convolution, was suggested by Beale and Majda [29,28,21] in their convergence proof. They choose the smoothing radius as $\epsilon = h^q$ with $0 < q < 1$. The corresponding error then goes as $\mathcal{O}(h^{qm})$, which can be made close to $\mathcal{O}(h^m)$ for q close to 1.

Choosing the smoothing radius ϵ is not trivial as a too conservative choice of α might lead to a break-down of the convergence rate, as seen in Fig. 5(a), for the G_ϵ kernel of $m = 10$. The same behaviour is also found using $\epsilon = h^q$ when q is close to 1 (not shown). It is seen that keeping ϵ proportional to h will produce an $\mathcal{O}(h^{10})$ error, up to a certain point. However, eventually the discretisation error overwhelms the smoothing error which causes the convergence rate to break-down. This crossover point can be offset by increasing the overlap α, albeit at the cost of a higher smoothing error.

In Fig. 5(b) it is seen to what extent the calculated discrete moments satisfy Eq. (14) when using different values of α. We consider the second-order ($\beta = 2$) moment, shown for the smoothing functions with $m = 4, 8, 12, 16$. It clearly shows that in order for the discrete moments to be satisfied to the accuracy of the quadrature i.e. at the kinks of the curves, the smoothing radius must be greater than the grid size by a margin or overlap α which is related to the order of the smoothing function m. We also observe that a higher order kernel will require a larger overlap to achieve the same moment error as a lower order one: as expected, higher order polynomials (Eq. (24)) require more points to be captured accurately.

6. Conclusion

A convolution integral method was presented for calculating the solution to Poisson's equation of a continuous field in an unbounded domain to an arbitrary high order of convergence. The smoothing method applied in mesh-free vortex methods to produce vortex blobs was combined with the FFT-based Poisson solver of particle-mesh methods to achieve a high rate of convergence. The high order is achieved without any additional computational cost other than that needed to initially calculate the high order integration kernel.

The method was shown to be able to calculate the derivative of the solution explicitly, either by direct spectral differentiating or by analytically differentiating the integration kernel. The two methods of calculating the derivative were found to give practically identical results and convergence behaviour, and as using spectral differentiating only requires a single integration kernel, the allocated memory can be reduced by choosing this method. It was also shown that by performing

the differentiation explicitly, differential operators such as the curl of a vector field could be performed in Fourier space, due to the linearity of the problem. This eliminates the need for an additional numerical scheme and maintains the specified order of convergence.

Acknowledgements

The research has been supported by the Danish Research Council of Independent Research (Grant No. 274-08-0258). We would also like to acknowledge the helpful discussions with Thomas Beale, Diego Rossinelli, Wim van Rees and Petros Koumoutsakos.

Appendix A. Kernel functions

To simplify the implementation of the presented method, the integration kernels G_ϵ and the corresponding velocity kernels $\boldsymbol{K}_\epsilon$ are here given explicitly for $m = 4, 6, 8, 10$. Our web page[1] provides maple scripts for the derivation of the high order integration kernels and the analytical solution to the bump function test case. Furthermore, simple Matlab implementations of the 2D and 3D Poisson solver as applied to the test cases, are also available.

A.1. Regularised integration kernels

The regularised integration kernel of order m is written on the general form

$$G_\epsilon(\boldsymbol{x}) = \begin{cases} -\frac{1}{2\pi}(\log(|\boldsymbol{x}|) - Q_m(\frac{|\boldsymbol{x}|}{\epsilon})\exp(\frac{-|\boldsymbol{x}|^2}{2\epsilon^2}) + \frac{1}{2}\mathrm{E}_1(\frac{|\boldsymbol{x}|^2}{2\epsilon^2})) & \text{in 2D,} \\ \frac{1}{4\pi|\boldsymbol{x}|}(\frac{1}{\sqrt{2\pi}}R_m(\frac{|\boldsymbol{x}|}{\epsilon})\exp(\frac{-|\boldsymbol{x}|^2}{2\epsilon^2}) + \mathrm{erf}(\frac{|\boldsymbol{x}|}{\sqrt{2}\epsilon})) & \text{in 3D,} \end{cases} \tag{A.1}$$

where the first four polynomials Q_m and R_m with $m = 4, 6, 8, 10$ are given by

$$Q_4(\rho) = \frac{1}{2}, \qquad R_4(\rho) = \rho, \tag{A.2}$$

$$Q_6(\rho) = \frac{3}{4} - \frac{1}{8}\rho^2, \qquad R_6(\rho) = \frac{7}{4}\rho - \frac{1}{4}\rho^3, \tag{A.3}$$

$$Q_8(\rho) = \frac{11}{12} - \frac{7}{24}\rho^2 + \frac{1}{48}\rho^4, \qquad R_8(\rho) = \frac{19}{8}\rho - \frac{2}{3}\rho^3 + \frac{1}{24}\rho^5, \tag{A.4}$$

$$Q_{10}(\rho) = \frac{25}{24} - \frac{23}{48}\rho^2 + \frac{13}{192}\rho^4 - \frac{1}{384}\rho^6, \qquad R_{10}(\rho) = \frac{187}{64}\rho - \frac{233}{192}\rho^3 + \frac{29}{192}\rho^5 - \frac{1}{192}\rho^7. \tag{A.5}$$

The centre value $G_\epsilon(0)$ given by Eq. (31) is

$$m = 4: \quad G_\epsilon(0) = \frac{1}{2\pi}\left(\frac{\gamma}{2} - \log(\sqrt{2}\epsilon) + \frac{1}{2}\right) \text{ in 2D}, \qquad G_\epsilon(0) = \frac{3}{8}\frac{\sqrt{2}}{\pi^{3/2}\epsilon} \text{ in 3D}, \tag{A.6}$$

$$m = 6: \quad G_\epsilon(0) = \frac{1}{2\pi}\left(\frac{\gamma}{2} - \log(\sqrt{2}\epsilon) + \frac{3}{4}\right) \text{ in 2D}, \qquad G_\epsilon(0) = \frac{15}{32}\frac{\sqrt{2}}{\pi^{3/2}\epsilon} \text{ in 3D}, \tag{A.7}$$

$$m = 8: \quad G_\epsilon(0) = \frac{1}{2\pi}\left(\frac{\gamma}{2} - \log(\sqrt{2}\epsilon) + \frac{11}{12}\right) \text{ in 2D}, \qquad G_\epsilon(0) = \frac{35}{64}\frac{\sqrt{2}}{\pi^{3/2}\epsilon} \text{ in 3D}, \tag{A.8}$$

$$m = 10: \quad G_\epsilon(0) = \frac{1}{2\pi}\left(\frac{\gamma}{2} - \log(\sqrt{2}\epsilon) + \frac{25}{24}\right) \text{ in 2D}, \qquad G_\epsilon(0) = \frac{315}{512}\frac{\sqrt{2}}{\pi^{3/2}\epsilon} \text{ in 3D}, \tag{A.9}$$

where $\gamma = 0.5772156649$ is Euler's constant.

A.2. Regularised velocity kernels

The corresponding velocity kernels on the general form are

$$\boldsymbol{K}_\epsilon(\boldsymbol{x}) = \begin{cases} -\frac{1}{2\pi|\boldsymbol{x}|^2}(1 - Q_m(\frac{|\boldsymbol{x}|}{\epsilon})\exp(\frac{-|\boldsymbol{x}|^2}{2\epsilon^2}))\boldsymbol{x}\times & \text{in 2D,} \\ -\frac{1}{4\pi|\boldsymbol{x}|^3}(\frac{1}{\sqrt{2\pi}}R_m(\frac{|\boldsymbol{x}|}{\epsilon})\exp(\frac{-|\boldsymbol{x}|^2}{2\epsilon^2}) + \mathrm{erf}(\frac{|\boldsymbol{x}|}{\sqrt{2}\epsilon}))\boldsymbol{x}\times & \text{in 3D,} \end{cases} \tag{A.10}$$

where

[1] http://www.student.dtu.dk/~jhwa/poisson/scripts.tar.gz.

$$\boldsymbol{x}\times = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}, \tag{A.11}$$

where the polynomials Q_m and R_m for $m = 4, 6, 8, 10$ are

$$Q_4(\rho) = 1 - \frac{1}{2}\rho^2, \qquad R_4(\rho) = -2\rho + \rho^3, \tag{A.12}$$

$$Q_6(\rho) = 1 - \rho^2 + \frac{1}{8}\rho^4, \qquad R_6(\rho) = -2\rho + \frac{9}{4}\rho^3 - \frac{1}{4}\rho^5, \tag{A.13}$$

$$Q_8(\rho) = 1 - \frac{3}{2}\rho^2 + \frac{3}{8}\rho^4 - \frac{1}{48}\rho^6, \qquad R_8(\rho) = -2\rho + \frac{89}{24}\rho^3 - \frac{20}{24}\rho^5 + \frac{1}{24}\rho^7, \tag{A.14}$$

$$Q_{10}(\rho) = 1 - 2\rho^2 + \frac{3}{4}\rho^4 - \frac{1}{12}\rho^6 + \frac{1}{384}\rho^8, \qquad R_{10}(\rho) = -2\rho + \frac{1027}{192}\rho^3 - \frac{349}{192}\rho^5 + \frac{35}{192}\rho^7 - \frac{1}{192}\rho^9, \tag{A.15}$$

and the centre value of the velocity kernel is defined as $\boldsymbol{K}_\epsilon(0) = 0$.

References

[1] G. Morgenthal, J.H. Walther, An immersed interface method for the vortex-in-cell algorithm, Computers & Structures 85 (2007) 712–726.
[2] J. Shimada, H. Kaneko, T. Takada, Performance of fast multipole methods for calculating electrostatic interactions in biomacromolecular simulations, J. Comput. Chem. 15 (1) (1994) 28–43.
[3] R.D. Budiardja, C.Y. Cardall, Parallel FFT-based Poission solver for isolated three-dimensional systems, Comp. Phys. Commun. 182 (2011) 2265–2275.
[4] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, 2nd edition, Institute of Physics Publishing, Bristol, PA, USA, 1988.
[5] G.-H. Cottet, P. Koumoutsakos, Vortex Methods – Theory and Practice, Cambridge University Press, New York, 2000.
[6] M. Griebel, S. Knapek, G.W. Zumbusch, Numerical Simulation in Molecular Dynamics, Springer-Verlag, Berlin, 2007.
[7] A. Leonard, Vortex methods for flow simulation, J. Comput. Phys. 37 (1980) 289–335.
[8] J. Barnes, P. Hut, A hierarchical $O(N \log N)$ force-calculation algorithm, Nature 324 (4) (1986) 446–449.
[9] J. Carrier, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, SIAM J. Sci. Stat. Comput. 9 (4) (1988) 669–686.
[10] J.P. Christiansen, Numerical simulation of hydrodynamics by the method of point vortices, J. Comput. Phys. 13 (1973) 363–379.
[11] R.A. James, The solution of Poisson's equation for isolated source distributions, J. Comput. Phys. 25 (1977) 71–93.
[12] D.B. Serafini, P. McCorquodale, P. Colella, Advanced 3D Poisson solvers and particle-in-cell methods for accelerator modeling, J. Phys.: Conf. Ser. 16 (2005) 481–485.
[13] T. Colonius, K. Taira, A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions, Comp. Meth. Appl. Mech. & Engng. 197 (25) (2008) 2131–2146.
[14] R. Cocle, G. Winckelmans, G. Daeninck, Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations, J. Comput. Phys. 227 (2008) 9091–9120.
[15] J.T. Rasmussen, Particle methods in bluff body aerodynamics, PhD thesis, Technical University of Denmark, October 2011.
[16] P. Chatelain, P. Koumoutsakos, A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions, J. Comput. Phys. 229 (2010) 2425–2431.
[17] J.T. Rasmussen, G.-H. Cottet, J.H. Walther, A multiresolution remeshed vortex-in-cell algorithm using patches, J. Comput. Phys. 230 (17) (2011) 6742–6755.
[18] J.V. Villumsen, A new hierarchical particle-mesh code for very large scale cosmological n-body simulations, Astro. J. Supp. 71 (1989) 407–431.
[19] J. Qiang, A high-order fast method for computing convolution integral with smooth kernel, Comp. Phys. Commun. 181 (2010) 313–316.
[20] A. Leonard, Vortex methods for flow simulation, J. Comput. Phys. 37 (1980) 289–335.
[21] J.T. Beale, A. Majda, High order accurate vortex methods with explicit velocity kernels, J. Comput. Phys. 58 (1985) 188–208.
[22] M. Perlman, On the accuracy of vortex methods, J. Comput. Phys. 59 (1985) 200–223.
[23] O.H. Hald, Convergence of vortex methods for Euler's equations, III, SIAM J. Numer. Anal. 24 (3) (1987) 538–582.
[24] G.S. Winckelmans, A. Leonard, Contribution to vortex particle methods for the computation of three-dimensional incompressible unsteady flows, J. Comput. Phys. 109 (1993) 247–273.
[25] T.W. Körner, Fourier Analysis, Cambridge University Press, Cambridge, 1988.
[26] P.-A. Raviart, Particle approximation of first order systems, J. Comput. Math. 4 (1) (1986) 50–61.
[27] O.H. Hald, Convergence of vortex methods for Euler's equations, II, SIAM J. Numer. Anal. 16 (1979) 726–755, 5.
[28] J.T. Beale, A. Majda, Vortex methods. I: Convergence in three dimensions, Math. Comput. 39 (159) (1982) 1–27.
[29] J.T. Beale, A. Majda, Vortex methods. II: Higher order accuracy in two and three dimensions, Math. Comput. 39 (159) (1982) 29–52.
[30] P. Chatelain, A. Leonard, Isotropic compact interpolation schemes for particle methods, J. Comput. Phys. 227 (2008) 3244–3259.
[31] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables, Applied Mathematics Series, vol. 55, National Bureau of Standards, 1972.