Journal of Computational Physics 326 (2016) 188–196

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Short note

A multiresolution method for solving the Poisson equation using high order regularization

Mads Mølholm Hejlesen [a], Jens Honoré Walther [a,b,*]

[a] Department of Mechanical Engineering, Technical University of Denmark, Building 403, DK-2800 Kgs. Lyngby, Denmark
[b] Computational Science and Engineering Laboratory, ETH Zürich, Clausiusstrasse 33, CH-8092 Zürich, Switzerland

ARTICLE INFO

Article history:
Received 15 March 2016
Received in revised form 28 August 2016
Accepted 31 August 2016
Available online 7 September 2016

Keywords:
The Poisson equation
Regularization methods
Particle–mesh methods
Green's function solution

ABSTRACT

We present a novel high order multiresolution Poisson solver based on regularized Green's function solutions to obtain exact free-space boundary conditions while using fast Fourier transforms for computational efficiency. Multiresolution is a achieved through local refinement patches and regularized Green's functions corresponding to the difference in the spatial resolution between the patches. The full solution is obtained utilizing the linearity of the Poisson equation enabling super-position of solutions. We show that the multiresolution Poisson solver produces convergence rates that correspond to the regularization order of the derived Green's functions.

1. Introduction

Solving the Poisson equation is essential to many physical systems and frequently appears in fields such as astrophysics, electrodynamics and fluid mechanics. For many such systems, the elementary particles tend to condense in local regions (e.g. galaxies in astrophysics or the vortex structures of fluid dynamics). Hence, the computational method for solving the Poisson equation may significantly increase its efficiency by applying a localized mesh refinement scheme here referred to as *multiresolution* methods. Multiresolution methods have been implemented in particle–mesh simulation for astrophysics [1] and represents the state-of-the-art vortex methods [2–5].

For simulating strict free-space boundary conditions there are two widely used methods, both based essentially on the same principle. The particles are merged into a new set of data points which conserves a desired number of moments of the original particles and furthermore offers a favorable data-structure for a more effective evaluation of the interaction. One of these methods is the multipole method [6] based on tree algorithms [7]. Here the number of interactions is reduced by evaluating the long range interactions by multipole expansions of groups of particles. The other method is the particle–mesh method. Here moment-conserving schemes are used to interpolate the particle quantities to structured data points of a uniform mesh. The interpolation can be viewed as the counterpart to calculating the multipoles in multipole methods. The number of new data points in the particle–mesh method is however not reduced significantly but the equi-spaced structure of the data points enables the use of fast recursive algorithms such as the fast Fourier transform (FFT). Using a mesh to calculate the particle interactions prevents any interactions at lengths shorter than the mesh cell length. However a nearby particle–particle correction scheme can be applied to recover contribution from any unresolved short-range inter-

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

http://dx.doi.org/10.1016/j.jcp.2016.08.053
0021-9991/

action potentials. This method is known as the particle–particle particle–mesh (P^3M) method [8–11]. A similar concept of recovering sub-mesh scales is in this work combined with a regularization method to obtain a high order multiresolution Poisson solver.

The presented regularization method was introduced and developed in references [12–14] for the Green's function based Poisson solver to achieve an increased order of convergence for unbounded domains. In this work the method is extended using a novel approach formulated in Fourier space to achieve a multiresolution solver by combining the method of particle–particle corrections from the P^3M method with this regularization method. Here a number of local refinement patches are used to retrieve the solution at a higher band of wave-numbers after which the super-position principle is used to obtain the full solution. The multiresolution implementation is shown to maintain a high order converging solution for different configurations of the refinement patches.

For a full implementation of the presented multiresolution method in a particle–mesh method, the reader is referred to the approach of Rasmussen et al. [5]. Here a mesh-based Poisson solver and computational particles are combined in a multiresolution implementation using local refinement patches.

2. Methodology

2.1. The Green's function solution to the Poisson equation in unbounded domains

For a given vector field $\boldsymbol{v}(\boldsymbol{x})$ which has at least two continuous derivatives (i.e. $\boldsymbol{v}(\boldsymbol{x}) \in C^2$) we may use the Helmholtz decomposition to express the field in terms of two potential functions $\phi(\boldsymbol{x})$ and $\boldsymbol{\psi}(\boldsymbol{x})$ by:

$$\boldsymbol{v}(\boldsymbol{x}) = \nabla \times \boldsymbol{\psi}(\boldsymbol{x}) - \nabla\phi(\boldsymbol{x}) \qquad \text{where} \qquad \nabla \cdot \boldsymbol{\psi}(\boldsymbol{x}) = 0 \tag{1}$$

The fundamental operations describing the conservation of the flux and circulation of the vector field $\boldsymbol{v}(\boldsymbol{x})$ is the divergence: $\vartheta(\boldsymbol{x}) = \nabla \cdot \boldsymbol{v}(\boldsymbol{x})$, and the curl: $\boldsymbol{\omega}(\boldsymbol{x}) = \nabla \times \boldsymbol{v}(\boldsymbol{x})$, respectively. From the Helmholtz decomposition it follows that these may be expressed by the potential functions:

$$\vartheta(\boldsymbol{x}) = -\nabla^2\phi(\boldsymbol{x}) \qquad \text{and} \qquad \boldsymbol{\omega}(\boldsymbol{x}) = -\nabla^2\boldsymbol{\psi}(\boldsymbol{x}) \tag{2}$$

As seen, both potentials may thus be obtained by solving a Poisson equation. By utilizing the linearity of the Poisson equation and considering the equations by their Fourier transform we obtain the algebraic equations:

$$\begin{aligned} -k^2\widehat{\phi}(\boldsymbol{k}) &= -\widehat{\vartheta}(\boldsymbol{k}) \\ -k^2\widehat{\boldsymbol{\psi}}(\boldsymbol{k}) &= -\widehat{\boldsymbol{\omega}}(\boldsymbol{k}) \end{aligned} \quad \Rightarrow \quad \begin{aligned} \widehat{\phi}(\boldsymbol{k}) &= \frac{\widehat{\vartheta}(\boldsymbol{k})}{k^2} \\ \widehat{\boldsymbol{\psi}}(\boldsymbol{k}) &= \frac{\widehat{\boldsymbol{\omega}}(\boldsymbol{k})}{k^2} \end{aligned} \tag{3}$$

Here the $\widehat{\cdot}$ denotes a Fourier transformed variable where $\boldsymbol{k} = (k_x, k_y, k_z)$ is the angular wave-number of Fourier space, and $k = |\boldsymbol{k}|$ is the magnitude of the angular wave-number vector. For unbounded domains we obtain the solution to Eq. (3) by defining Green's functions. Here the homogeneous solution with appropriate boundary conditions (i.e., the Green's function) is derived analytically and the particular solution is then found by convolving the Green's function with the right-hand-side of the Poisson equation. Hence, we invert the Poisson equation by convolving with a Green's function (G), and utilize the commutative properties of the convolution operation with respect to differentiation to obtain:

$$\phi(\boldsymbol{x}) = G(\boldsymbol{x}) * \vartheta(\boldsymbol{x}) \qquad \text{and} \qquad \boldsymbol{\psi}(\boldsymbol{x}) = G(\boldsymbol{x}) * \boldsymbol{\omega}(\boldsymbol{x}) \tag{4}$$

where $*$ denotes linear convolution. In Fourier space the Green's function is given by:

$$-k^2\widehat{G}(k) = -1 \quad \Rightarrow \quad \widehat{G}(k) = \frac{1}{k^2} \tag{5}$$

The corresponding singular Green's functions for 2D and 3D in real space thus become:

$$\nabla^2 G(\boldsymbol{x}) = -\delta(\boldsymbol{x}) \qquad \Rightarrow \qquad G(\boldsymbol{x}) = \begin{cases} -\dfrac{1}{2\pi}\ln(r) & \text{in 2D,} \\ \dfrac{1}{4\pi r} & \text{in 3D,} \end{cases} \tag{6}$$

where $r = |\boldsymbol{x}|$.

2.2. The Green's function solution to the Poisson equation using a discretized approximation with high order regularization

The discrete equivalent to the field quantities is obtained by defining a set of N discrete mesh points $\boldsymbol{x}_i$ where $i = \{1, \ldots, N\}$. The mesh points are assigned the field values ϑ_i and $\boldsymbol{\omega}_i$ such that the approximated continuous field values $\vartheta(\boldsymbol{x})$ and $\boldsymbol{\omega}(\boldsymbol{x})$ become:

$$\vartheta(\boldsymbol{x}) = \sum_{i=1}^{N} \zeta(\boldsymbol{x}-\boldsymbol{x}_i)\vartheta_i \qquad \text{and} \qquad \boldsymbol{\omega}(\boldsymbol{x}) = \sum_{i=1}^{N} \zeta(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{\omega}_i \tag{7}$$

Here ζ is a regularization function which represents the density distribution of the discrete element. Using this representation the convolution equations take the semi-discrete form:

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{N} G_\zeta(\boldsymbol{x}-\boldsymbol{x}_i)\vartheta_i \qquad \text{and} \qquad \boldsymbol{\psi}(\boldsymbol{x}) = \sum_{i=1}^{N} G_\zeta(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{\omega}_i \tag{8}$$

and the regularized homogeneous equation then becomes:

$$\nabla^2 G_\zeta(\boldsymbol{x}-\boldsymbol{x}_i) = -\zeta(\boldsymbol{x}-\boldsymbol{x}_i) \tag{9}$$

For a discrete approximation as is used in numerical simulations, singularities cannot be handled in Fourier space. Due to the discretization of the domain in physical space ($\boldsymbol{x}_i$), the Fourier space, represented by the wave-numbers $k_i = \{-k_s/2, \ldots, k_s/2\}$, is bounded and can therefore only reproduce a singularity by a finite approximation. Here $k_s = 2\pi/h$ is the angular sampling wave-number corresponding to the discrete mesh cell length (h), and $k_s/2$ is the highest resolved angular wave-number due to the Nyquist–Shannon sampling theorem. Attempts of defining a finite value to replace the singularity $G(\mathbf{0})$ have been presented in [8,15,16] however with a convergence rate limited to $\mathcal{O}(h^2)$.

A method to achieve a higher order convergence rate is to use a spectrally compact regularization function which makes the integral of the inverse Fourier transform convergent [13,14]. Hence, the regularization function must fulfill:

$$\widehat{\zeta}(k) = 1 \quad \text{for} \quad k = 0 \qquad \text{and} \qquad \widehat{\zeta}(k) = 0 \quad \text{for} \quad k \geq k_s/2 \tag{10}$$

An m-th order regularization is obtained by nullifying the $m-2$ order derivatives of the spectral function at $k=0$. This is equivalent to conserving the higher order moments in real space which is summarized by the following additional requirements:

$$\left.\frac{d^\beta \widehat{\zeta}}{dk^\beta}\right|_{k=0} = \int_{-\infty}^{\infty} r^\beta \zeta(r)\, d\boldsymbol{x} = 0^\beta \quad \text{for} \quad \beta = \{0, 2, \ldots, m-2\} \tag{11}$$

We note that m is an even number since the moments for odd integers of β are automatically satisfied due to the symmetry of the regularization function (ζ).

2.3. Regularization by Gaussian smoothing for unbounded domains

One approach to obtain an m-th order Gaussian regularization (ζ_m) is by performing an approximate deconvolution [17, 13,14] on the 2nd order Gaussian regularization function which effectively conserves higher order moments of the original field. An m-th order deconvolution function is thus obtained by a truncated series, after which the m-th order regularization function is given by:

$$\widehat{\zeta}_m(s) = \sum_{q=0}^{m/2-1} \frac{(s^2/2)^q}{q!} \exp\left(-\frac{s^2}{2}\right) \tag{12}$$

Here $s = k\sigma$ is the angular wave-number normalized with σ, which here represents a smoothing radius. The regularization functions for $m = \{2, 4, 6, 8, 10\}$ are shown in Fig. 1.

Using the high order regularization functions of Eq. (12) we find that the Gaussian regularized Green's functions G_m of order m take the general form previously presented in [12,13]:

$$G_m(\rho) = \begin{cases} -\dfrac{1}{2\pi}\left(\ln(\sigma\rho) - P_m(\rho)\exp\left(-\dfrac{\rho^2}{2}\right) + \dfrac{1}{2}\mathrm{E}_1\left(\dfrac{\rho^2}{2}\right)\right) & \text{in 2D} \\ \dfrac{1}{4\pi\sigma\rho}\left(\mathrm{Q}_m(\rho)\exp\left(-\dfrac{\rho^2}{2}\right) + \mathrm{erf}\left(\dfrac{\rho}{\sqrt{2}}\right)\right) & \text{in 3D} \end{cases} \tag{13}$$

where $\rho = r/\sigma$ is the normalized radial coordinate, $\mathrm{E}_1(z)$ is the first exponential integral function and $\mathrm{erf}(z)$ is the regular error function. P_m is the resulting polynomials for the 2D function which for $m = \{2, 4, 6, 8, 10\}$ are:

$$\begin{aligned} P_2(\rho) &= 0 & P_8(\rho) &= \frac{11}{12} - \frac{7}{24}\rho^2 + \frac{1}{48}\rho^4 \\ P_4(\rho) &= \frac{1}{2} & P_{10}(\rho) &= \frac{25}{24} - \frac{23}{48}\rho^2 + \frac{13}{192}\rho^4 - \frac{1}{384}\rho^6 \\ P_6(\rho) &= \frac{3}{4} - \frac{1}{8}\rho^2 \end{aligned} \tag{14}$$

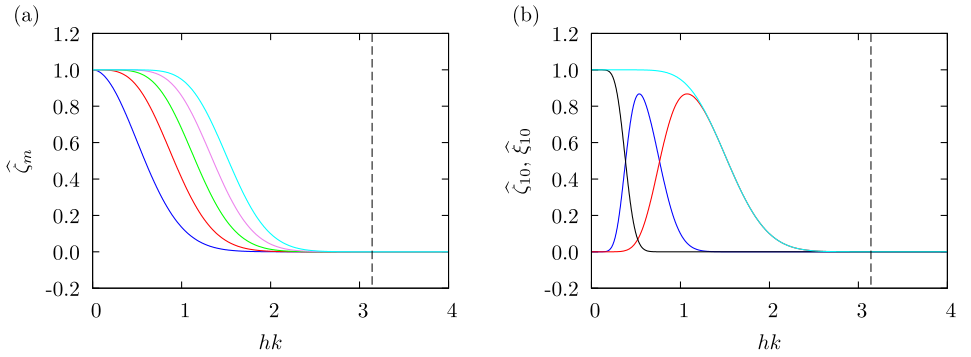

Fig. 1. (a) The regularization function in radial Fourier space with $\sigma = 2h$ for $m = 2$: (——); $m = 4$: (——); $m = 6$: (——); $m = 8$: (——); $m = 10$: (——). (b) The multiresolution regularization functions in Fourier space for $m = 10$ and $\sigma = 2h$ using two levels ($M = 2$) of mesh refinements (h_0, h_1 and h_2) with ratios $a_1 = h_0/h_1 = 2$ and $a_2 = h_1/h_2 = 2$ as are also sketched in Fig. 3. The regularization function for: the base mesh: (——); the level 1 refinement mesh: (——); the level 2 refinement mesh: (——); and the resulting regularization by super-position: (——) The highest wave-number is indicated by: (– – – –). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

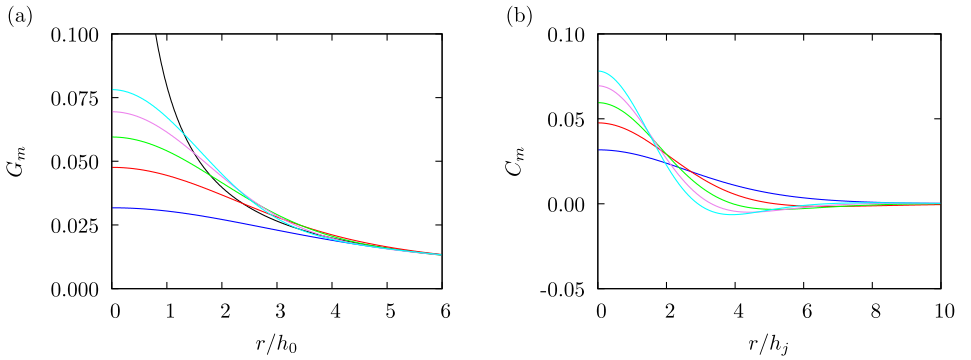

Fig. 2. The high order regularized Green's functions for 3D with $\sigma = 2h$ and $a_1 = 2$: (a) for the base mesh (G_m), and (b) for the refinement mesh (C_m). $m = 2$: (——); $m = 4$: (——); $m = 6$: (——); $m = 8$: (——); $m = 10$: (——). The singular Green's function G: (——). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Q_m is the resulting polynomials for the 3D function which for $m = \{2, 4, 6, 8, 10\}$ are:

$$
\begin{aligned}
Q_2(\rho) &= 0 & Q_8(\rho) &= \frac{1}{\sqrt{2\pi}}\left(\frac{19}{8}\rho - \frac{2}{3}\rho^3 + \frac{1}{24}\rho^5\right) \\
Q_4(\rho) &= \frac{1}{\sqrt{2\pi}}(\rho) & Q_{10}(\rho) &= \frac{1}{\sqrt{2\pi}}\left(\frac{187}{64}\rho - \frac{233}{192}\rho^3 + \frac{29}{192}\rho^5 - \frac{1}{192}\rho^7\right) \\
Q_6(\rho) &= \frac{1}{\sqrt{2\pi}}\left(\frac{7}{4}\rho - \frac{1}{4}\rho^3\right)
\end{aligned}
\tag{15}
$$

The finite value of $G_m(\mathbf{0})$ must be derived explicitly as:

$$
G_m(0) = \lim_{\rho \to 0}(G_m) = \begin{cases} \dfrac{1}{2\pi}\left(\dfrac{\gamma}{2} - \ln(\sqrt{2}\sigma) + Q_m(0)\right) & \text{in 2D} \\ \dfrac{\sqrt{2}}{8\pi^{3/2}\sigma}\left(\dfrac{R_m(\rho)}{\rho}\right)_{\rho=0} + \dfrac{\sqrt{2}}{4\pi^{3/2}\sigma} & \text{in 3D} \end{cases}
\tag{16}
$$

where $\gamma = 0.5772156649$ is Euler's constant. The derived regularized Green's functions G_m for $m = \{2, 4, 6, 8, 10\}$ (shown in Fig. 2) are consistent with those presented in [18–21,12,13].

2.4. The regularization method for a multiresolution Poisson solver

The mesh effectively introduces a highest resolved scale characterized by the Nyquist–Shannon sampling theorem. However due to the linearity of the Poisson equation we may decompose the problem into a multiscale formulation which can be solved independently and use a super-position of the solutions to solve the full problem. This strategy is used in the particle–particle particle–mesh method (P^3M) [8–10] to perform singular particle interactions using a particle–mesh method

for the mesh resolved scales and a particle–particle interaction for recovering the unresolved part of the potentials. A similar approach is used here to formulate a multiresolution method for the vector field representation. However, instead of using a particle–particle correction we now use a number of particle–mesh solvers at a higher resolution in order to recover refined scales of the solution. This is achieved by making a decomposition of the regularization function using M number of "nested" regularization functions which effectively serves a filter-bank of band-pass filters. Hence, we may decompose the highest resolved filter at level M into the sum of the coarsest (low-pass) filter at level 0 and the M multiresolution (band-pass) filters ξ_m for the refinement meshes of cell length h_j where $j = \{0, 1, \ldots, M\}$:

$$\zeta_m(\rho_M) = \zeta_m(\rho_0) + \sum_{j=1}^{M} \xi_m(\rho_j) \qquad \text{with} \qquad \rho_j = \frac{r}{\sigma_j} \quad \text{and} \quad \sigma_j = \alpha h_j \tag{17}$$

The band-pass regularization functions are then given by:

$$\xi_m(\rho_j) = \zeta_m(\rho_j) - \zeta_m\left(\frac{\rho_j}{a_j}\right) \qquad \text{with} \qquad a_j = \frac{h_{j-1}}{h_j} > 1 \tag{18}$$

The Green's function is decomposed in a similar fashion:

$$\nabla^2 G_\zeta(\rho_M) = \nabla^2 G_m(\rho_0) + \sum_{j=1}^{M} \nabla^2 C_m(\rho_j, a_j) \tag{19}$$

Here C_m is the Green's function for the band of scales between the two levels of refinement. The decomposition yields two equations for the regularized Green's function solutions on the meshes:

$$\nabla^2 G_m(\rho_0) = -\zeta_m(\rho_0) \qquad \text{and} \qquad \nabla^2 C_m(\rho_j, a_j) = -\xi_m(\rho_j) \tag{20}$$

The former is the regularized equation whose Green's function was derived in Sec. 2.2 and thus represents the regularized solution of the coarser scales that can be calculated using the particle–mesh method on the base mesh h_0. The latter equation represents each of the M refinement levels used to recover the finer scales of the solution which must be superpositioned with the solution of the coarser levels. The Green's functions for the refined scales are given in 2D by:

$$C_m(\rho_j, a_j) = \frac{1}{2\pi}\left(P_m(\rho_j)\exp\left(-\frac{\rho_j^2}{2}\right) - P_m\left(\frac{\rho_j}{a_j}\right)\exp\left(-\frac{\rho_j^2}{2a_j^2}\right) - \frac{1}{2}\mathrm{E}_1\left(\frac{\rho_j^2}{2}\right) + \frac{1}{2}\mathrm{E}_1\left(\frac{\rho_j^2}{2a_j^2}\right)\right) \tag{21}$$

and in 3D by:

$$C_m(\rho_j, a_j) = \frac{1}{4\pi\sigma\rho_j}\left(\mathrm{Q}_m(\rho_j)\exp\left(-\frac{\rho_j^2}{2}\right) - \mathrm{Q}_m\left(\frac{\rho_j}{a_j}\right)\exp\left(-\frac{\rho_j^2}{2a_j^2}\right) + \mathrm{erf}\left(\frac{\rho_j}{\sqrt{2}}\right) - \mathrm{erf}\left(\frac{\rho_j}{\sqrt{2}a_j}\right)\right) \tag{22}$$

An easy way to arrive at this result is to use the linearity of the problem by which:

$$C_m(\rho_j, a_j) = G_m\left(\rho_j\right) - G_m\left(\frac{\rho_j}{a_j}\right) \tag{23}$$

The obtained Green's functions are seen in Fig. 2. Here it is noticed that the Green's functions for recovering the refined scales converge to zero relatively fast. As will be shown later, the fast convergence towards zero can be utilized to reduce the computational cost of the numerical implementation.

The full solution of the Poisson equation is now found by the principle of super-position of solutions:

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{N_0} G_m\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{\sigma_0}\right)\vartheta_i + \sum_{j=1}^{M}\sum_{i=1}^{N_j} C_m\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{\sigma_j}, a_j\right)\vartheta_i = \phi_0(\boldsymbol{x}) + \sum_{j=1}^{M}\phi_j(\boldsymbol{x}) \tag{24}$$

and similarly for the vector potential:

$$\boldsymbol{\psi}(\boldsymbol{x}) = \sum_{i=1}^{N_0} G_m\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{\sigma_0}\right)\boldsymbol{\omega}_i + \sum_{j=1}^{M}\sum_{i=1}^{N_j} C_m\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{\sigma_j}, a_j\right)\boldsymbol{\omega}_i = \boldsymbol{\psi}_0(\boldsymbol{x}) + \sum_{j=1}^{M}\boldsymbol{\psi}_j(\boldsymbol{x}) \tag{25}$$

here N_j is the number of discrete mesh points on level j. As seen, the full solution consists of a super-position of filtered solutions which are calculated individually on meshes of different discretization refinement.

2.5. Numerical implementation of the regularized multiresolution Poisson solver

The data-structure of the mesh enables the use of FFT-based convolution. In order to circumvent the inherent periodicity of the FFT convolution for unbounded domains we adopt a zero-padding method introduced by Hockney and Eastwood [8,22,23].

The 1D discrete Fourier transform of an arbitrary discrete function (A_i) and its inverse transform are given by:

$$\widehat{A}_j = h\sum_{i=0}^{n-1} A_i \exp\left(-\frac{2\pi\imath ij}{n}\right) \quad \text{and} \quad A_i = \frac{1}{nh}\sum_{j=0}^{n-1} \widehat{A}_j \exp\left(\frac{2\pi\imath ij}{n}\right) \tag{26}$$

Here n is the discrete integration length of the Fourier transform, and $\imath$ is the imaginary unit. In order to obtain a linear convolution of a field $\mathcal{A}_i$ with $i = \{0, \ldots, N-1\}$, and an integral function $\mathcal{G}_j$ with $j = \{-L, \ldots, L-1\}$ where $L \leq N$, the discrete functions are extended with zeros to obtain the length of the Fourier integral $n = N + L$. Furthermore the indexes $\{-L, \ldots, -1\}$ of $\mathcal{G}_j$ are shifted to append the function in order to comply with the integration interval of the Fourier transform. We note that for $L = N$ the integral function ($\mathcal{G}_j$) extends to the entire length of $\mathcal{A}_i$, and hence only $\mathcal{A}_i$ is zero-padded to twice its original length.

The discrete Fourier transforms can be calculated efficiently by the recursive FFT algorithm which is one of the clear advantages of the presented method. It is here emphasized that the FFT-based convolution is a direct method for evaluating the discrete integral of the convolution and thus cannot introduce any additional artifacts that may otherwise be related to Fourier series e.g. Gibbs phenomena. Hence, the only error is the quadrature of the discrete integral which convergences in a rate proportional to the smoothness of the field.

The Poisson equations of Eq. (2) may thus be solved by:

$$\begin{aligned} \widehat{\phi}_0 &= \widehat{G}_m \widehat{\vartheta} \\ \widehat{\boldsymbol{\psi}}_0 &= \widehat{G}_m \widehat{\boldsymbol{\omega}} \end{aligned} \quad \text{and} \quad \begin{aligned} \widehat{\phi}_j &= \widehat{C}_m \widehat{\vartheta} \\ \widehat{\boldsymbol{\psi}}_j &= \widehat{C}_m \widehat{\boldsymbol{\omega}} \end{aligned} \tag{27}$$

The linearity of the Poisson equation enables the gradient to be calculated directly in Fourier space. Hence, the full solution of the vector field may be composed as:

$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{v}_0(\boldsymbol{x}) + \sum_{j=1}^{M} \boldsymbol{v}_j(\boldsymbol{x}) \quad \text{where} \quad \widehat{\boldsymbol{v}}_j = \imath\boldsymbol{k} \times \widehat{\boldsymbol{\psi}}_j - \imath\boldsymbol{k}\widehat{\phi}_j \tag{28}$$

The partial solution on the base mesh and the refinement patches is obtained accordingly to the previous Sec. 2.5 using the appropriate Green's functions. Here it is noted that the Green's functions of the refinement patches (Eqs. (21) and (22)) are approximately bounded functions cf. Fig. 2. This behavior may be utilized to reduce the computational footprint of solving the refinement meshes as the extent of the convolution kernel $\mathcal{G}_i$ becomes less than N after which the zero-padding is reduced together with the length of the Fourier transforms (n). In the presented study the extent of the Greens function L_j at the refinement level j is chosen to $10a_j$ by which the truncated values are below machine precision. The corresponding length of the Fourier integral thus becomes $n_j = N_j + 10a_j$.

Once the individual solutions are obtained on all meshes, the solutions are super-positioned in order to obtain the full solution accordingly to Eqs. (24), (25) and (28). In order to study the convergence rate of the presented Poisson solver, the solution of the coarse levels may be super-positioned onto the refined mesh points by applying the shift theorem in Fourier space. This method does not introduce any additional error, however comes at a high computational cost. Thus, it is only used here to demonstrate the formal convergence rate of the multiresolution solver when neglecting truncation errors introduced by a lower order mesh-to-mesh interpolation.

For a more practical use, the super-position of the solution is performed by a moment conserving mesh-to-mesh interpolation from the coarse level mesh to the refined level. Using a B-spline interpolation proposed by [24] we may obtain a fourth order interpolation scheme using the M_6^* interpolation function of [3,25]. In the current implementations we use a staggered mesh refinement as shown in Fig. 3 where the mesh nodes of the refinement mesh do not coincide with the mesh nodes of its parent mesh. The staggered configuration is practical when using a particle–mesh method, as the particles represent an integral value of a density field and is then easily super-positioned.

3. Results

3.1. Convergence test for the multiresolution method

The regularized Green's functions together with the numerical implementation are now tested to validate the convergence rate of the Poisson solver. To investigate the convergence of the Poisson solver, the test function must be bounded within the computational domain. In the presented examples we test the Poisson solver for calculating the vector field ($\boldsymbol{v}(\boldsymbol{x})$) given a pre-defined distribution of the curl of the vector field ($\boldsymbol{\omega}(\boldsymbol{x})$) through Eqs. (1) and (2) with a zero divergence ($\vartheta(\boldsymbol{x}) = 0$).

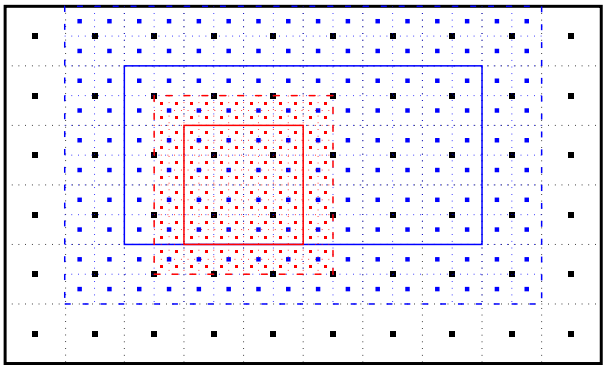

Fig. 3. Sketch of the numerical implementation of the multiresolution method using refinement patches with a staggered mesh configuration. The black mesh is the base mesh and the blue and red meshes are refinement meshes on level 2 and 3 respectively with a refinement ratio of $a_1 = a_2 = 2$. The edge of the refinement mesh is appended with a layer of ghost cells (marked with dashed lines) to ensure that the interpolation function is fully supported on the mesh. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Additionally, the test function must have at least two continuous derivatives which is a direct consequence of the mapping of the Laplace operator $\nabla^2 : C^k \mapsto C^{k-2}$ where $k \geq 2$ is needed to give a continuous solution for the Poisson equation. Furthermore, in order to make the quadrature error insignificant and thus allowing an arbitrary high convergence rate the test function must be infinitely continuous differentiable. To meet these requirements we propose the use of a bump function distribution which is of class C^∞ i.e., it has an infinite number of continuous derivatives [16,12]. The bump function is defined as:

$$f(q) = \begin{cases} \exp\left(-\dfrac{c}{1-q^2}\right) & \text{for} \quad |q| < 1 \\ 0 & \text{for} \quad |q| \geq 1 \end{cases} \tag{29}$$

where c is an arbitrary positive constant which for the presented cases $c = 10$ is chosen. In order to obtain bounded test functions, the bump function is used as the radial distribution of a circular patch in 2D and of a torus in 3D cf. Fig. 4. Here the location of the refinement meshes is also shown. Using centered polar coordinates (r, θ) in 2D and cylindrical coordinates (r, θ, z) in 3D case the vector potential may then be defined as:

$$\psi = f\left(\frac{r}{R}\right) \quad \text{in 2D} \qquad \text{and} \qquad \boldsymbol{\psi} = f\left(\frac{\sqrt{(r-R)^2 + z^2}}{R}\right)\boldsymbol{e}_\theta \quad \text{in 3D} \tag{30}$$

Here R is the radius of the radial distribution of the bump function and $\boldsymbol{e}_\theta$ is the azimuthal normal vector in 3D. The resulting distribution of $\boldsymbol{\omega}(\boldsymbol{x})$ and corresponding vector field $\boldsymbol{v}(\boldsymbol{x})$ are found analytically by Eqs. (1) and (2). We define the error of the calculated vector field $\boldsymbol{v}_i$ compared to the analytical solution $\boldsymbol{v}(\boldsymbol{x}_i)$ as:

$$\text{error} = \left(\frac{\sum_{i=1}^{N} |\boldsymbol{v}(\boldsymbol{x}_i) - \boldsymbol{v}_i|^2 h_i^d}{\sum_{i=1}^{N} |\boldsymbol{v}(\boldsymbol{x}_i)|^2 h_i^d} \right)^{1/2} \tag{31}$$

The obtained error of the 2D and 3D convergence tests are shown in Fig. 5 for the regularized Green's functions of order $m = \{2, 4, 6, 8, 10\}$ using a single refinement patch with a refinement ratio of $a_1 = 2$. As seen, the error of the 2D and the 3D convergence tests are coinciding, and the convergence rate corresponds to the respective design parameters of the Green's functions when super-positioning the solution using a shift in Fourier space. A similar convergence rate is observed with the maximum error (not shown). When super-positioning the solution by the fourth order interpolation M_6^* [3,25] it is seen that the convergence order is limited by the convergence rate of the interpolation error.

The convergence rate of the test using four bordering patches in 2D with a refinement ratio of $a_1 = 2$ (cf. Fig. 4) is seen in Fig. 6. For the four bordering patches, overlapping ghost regions of $10a_j$ cells are used according to the extent of the Green's function for the refinement patch C_m. Again the fourth order interpolation function M_6^* [3,25] is used for the super-position of the solution. It is seen in Fig. 6 that the obtained rate of convergence attains the order of the Green's functions for $m = \{2, 4\}$ however is limited by the interpolation scheme for higher order Green's function ($m > 4$). It is thus noted that the method is capable of handling bordering refinement patches without increasing the error when a sufficient amount of ghost cells is used.

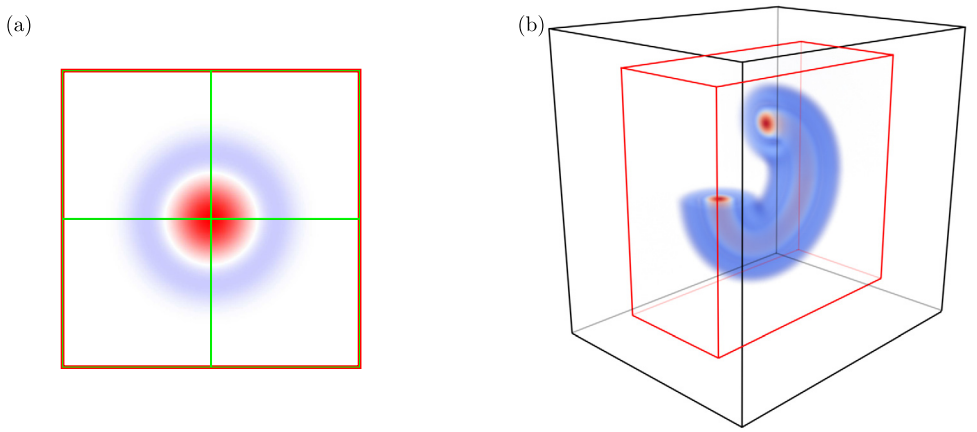

Fig. 4. The distribution of $\boldsymbol{\omega}(\boldsymbol{x})$ for the bump test function in 2D (a) and in 3D (b). The 3D test function is a closed torus but is shown in (b) without the upper left quadrant in order to make the core distribution visible. The location of the mesh refinement patches are indicated by the red box for the single patch configuration and green boxes for the configuration of four bordering patches in 2D. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

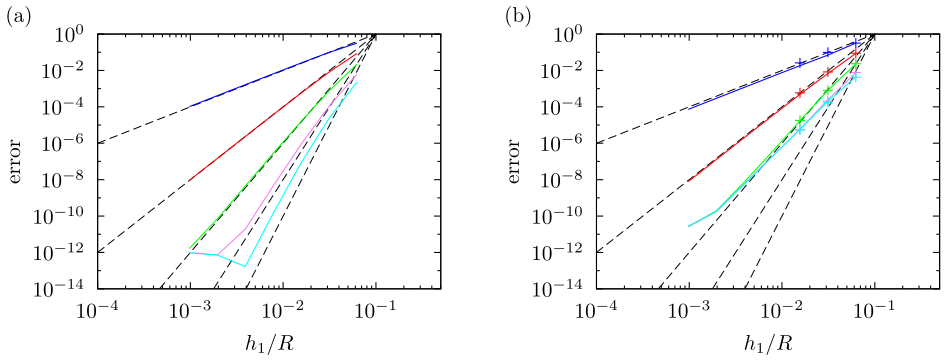

Fig. 5. The error of the calculated solutions obtained by regularized Green's functions using the multiresolution method with one patch and a mesh refinement ratio of $a_1 = 2$. The solution obtained using Fourier shift (a) and 4th order B-spline interpolation (b) to super-position the mesh solutions. Convergence order: (– – – –) from top $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$, $\mathcal{O}(h^6)$, $\mathcal{O}(h^8)$, $\mathcal{O}(h^{10})$; $m = 2$: (——); $m = 4$: (——); $m = 6$: (——); $m = 8$: (——); $m = 10$: (——). The lines represent the error for the 2D test cases and the points (+) in (b) represents the error for the 3D test cases. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

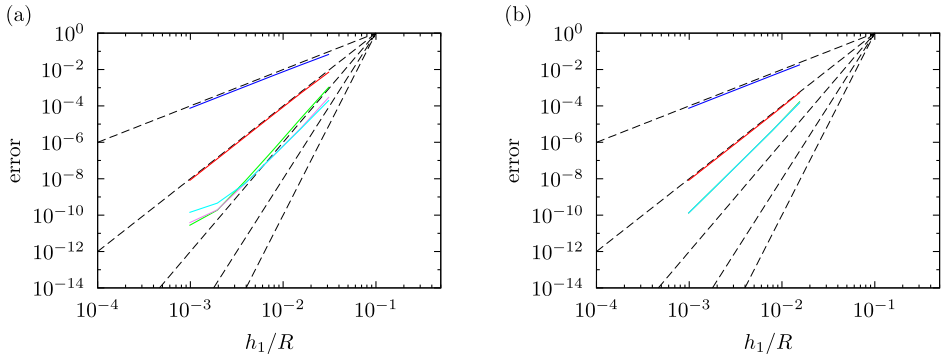

Fig. 6. The error of the calculated solutions obtained by regularized Green's functions using the multiresolution method with a mesh refinement ratio of $a_1 = 2$ with four patches (a) and $a_1 = 3$ with one patch (b). The solution is obtained using the 4th order B-spline interpolation to super-position the mesh solutions. Convergence order: (– – – –) from top $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$, $\mathcal{O}(h^6)$, $\mathcal{O}(h^8)$, $\mathcal{O}(h^{10})$; $m = 2$: (——); $m = 4$: (——); $m = 6$: (——); $m = 8$: (——); $m = 10$: (——). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The convergence rate obtained using a single refinement patch with a refinement ratio of $a_1 = 3$ is shown in Fig. 6. It is observed that for the larger mesh refinement ratio $a_1 = 3$ the error is larger due to an increased interpolation error.

4. Conclusion

A regularization method for solving the Poisson equation using Green's functions in a patch-based multiresolution method has been presented. The multiresolution combines the idea of a particle–particle correction from the P^3M method with a regularization method in order to obtain a localized refinement of the solution. This has been achieved by deriving regularized Green's functions to solve the Poisson equation on a band of wave-numbers corresponding to the difference in the resolution between the meshes and then utilizing the linearity of the Poisson equation to super-position the solution.

It has been shown that the multiresolution Poisson solver is able to formally produce a convergence rate corresponding to the regularization order of the Green's function. When combined with a lower order interpolation function to super-position the solution the method assumes the convergence rate of the interpolation scheme. The method has been tested for different configurations of mesh refinement patches. Here it has been shown that the method is capable of handling bordering patches without introducing additional errors.

For a full implementation of the a multiresolution method in a particle–mesh method the reader is referred to the approach of Rasmussen et al. [5] in terms of combining a mesh-based Poisson solver and particles in a multiresolution implementation using local refinement patches.

Acknowledgements

We would like to acknowledge the helpful discussions with Henrik Juul Spietz, Johannes Tophøj Rasmussen, and Philippe Chatelain.

References

[1] J.V. Villumsen, A new hierarchical particle–mesh code for very large scale cosmological n-body simulations, Astrophys. J. Suppl. 71 (1989) 407–431.
[2] M. Bergdorf, G.-H. Cottet, P. Koumoutsakos, Multilevel adaptive particle methods for convection–diffusion equations, Multiscale Model. Simul. 4 (1) (2005) 328–357.
[3] M. Bergdorf, P. Koumoutsakos, A Lagrangian particle-wavelet method, Multiscale Model. Simul. 5 (3) (2006) 980–995.
[4] M. El Ossmani, P. Poncet, Efficiency of multiscale hybrid grid-particle vortex methods, Multiscale Model. Simul. 8 (5) (2010) 1671–1690.
[5] 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.
[6] J. Carrier, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, SIAM J. Sci. Stat. Comput. 9 (4) (1988) 669–686.
[7] J. Barnes, P. Hut, A hierarchical $O(N \log N)$ force-calculation algorithm, Nature 324 (4) (1986) 446–449.
[8] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, 2nd edition, Institute of Physics Publishing, Bristol, PA, USA, 1988.
[9] J.H. Walther, An influence matrix particle–particle particle–mesh algorithm with exact particle–particle correction, J. Comput. Phys. 184 (2003) 670–678.
[10] B.A. Luty, W. van Gunsteren, Calculating electrostatic interactions using the particle–particle particle–mesh method with nonperiodic long-range interactions, J. Phys. Chem. 100 (1996) 2581–2587.
[11] G. Morgenthal, J.H. Walther, An immersed interface method for the vortex-in-cell algorithm, Comput. Struct. 85 (2007) 712–726.
[12] M.M. Hejlesen, J.T. Rasmussen, P. Chatelain, J.H. Walther, A high order solver for the unbounded Poisson equation, J. Comput. Phys. 252 (2013) 458–467.
[13] M.M. Hejlesen, J.T. Rasmussen, P. Chatelain, J.H. Walther, High order Poisson solver for unbounded flows, Proc. IUTAM 18 (2015) 56–65.
[14] M.M. Hejlesen, A high order regularisation method for solving the Poisson equation and selected applications using vortex methods, Ph.D. thesis, Technical University of Denmark, February 2016.
[15] P. Chatelain, P. Koumoutsakos, A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions, J. Comput. Phys. 229 (2010) 2425–2431.
[16] J.T. Rasmussen, Particle methods in bluff body aerodynamics, Ph.D. thesis, Technical University of Denmark, October 2011.
[17] S. Stolz, N.A. Adams, An approximate deconvolution procedure for large-eddy simulation, Phys. Fluids 11 (7) (1999) 1699–1701.
[18] A. Leonard, Vortex methods for flow simulation, J. Comput. Phys. 37 (1980) 289–335.
[19] J.T. Beale, A. Majda, High order accurate vortex methods with explicit velocity kernels, J. Comput. Phys. 58 (1985) 188–208.
[20] M. Perlman, On the accuracy of vortex methods, J. Comput. Phys. 59 (1985) 200–223.
[21] 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.
[22] R.W. Hockney, The potential calculation and some applications, Methods Comput. Phys. 9 (1970) 136–210.
[23] J.W. Eastwood, D.R.K. Brownrigg, Remarks on the solution of Poisson's equation for isolated systems, J. Comput. Phys. 32 (1979) 24–38.
[24] J.J. Monaghan, Extrapolating B splines for interpolation, J. Comput. Phys. 60 (2) (1985) 253–262.
[25] W.M. van Rees, A. Leonard, D.I. Pullin, P. Koumoutsakos, A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers, J. Comput. Phys. 230 (2011) 2794–2805.