



An integral equation–based numerical method for the forced heat equation on complex domains

Fredrik Fryklund¹  · Mary Catherine A. Kropinski² · Anna-Karin Tornberg¹

Received: 18 July 2019 / Accepted: 19 June 2020 /
Published online: 20 August 2020
© The Author(s) 2020

Abstract

Integral equation–based numerical methods are directly applicable to homogeneous elliptic PDEs and offer the ability to solve these with high accuracy and speed on complex domains. In this paper, such a method is extended to the heat equation with inhomogeneous source terms. First, the heat equation is discretised in time, then in each time step we solve a sequence of so-called modified Helmholtz equations with a parameter depending on the time step size. The modified Helmholtz equation is then split into two: a homogeneous part solved with a boundary integral method and a particular part, where the solution is obtained by evaluating a volume potential over the inhomogeneous source term over a simple domain. In this work, we introduce two components which are critical for the success of this approach: a method to efficiently compute a high-regularity extension of a function outside the domain where it is defined, and a special quadrature method to accurately evaluate singular and nearly singular integrals in the integral formulation of the modified Helmholtz equation for all time step sizes.

Keywords Heat equation · Boundary integral method · Modified Helmholtz equation · Yukawa potential · Quadrature · Complex domains · Function extension · Rothe’s method

Mathematics Subject Classification (2010) 65M70 · 65M80 · 65R20

Communicated by: Leslie Greengard

✉ Fredrik Fryklund
ffry@kth.se

Mary Catherine A. Kropinski
mkropins@sfu.ca

Anna-Karin Tornberg
akto@kth.se

¹ Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden

² Department of Mathematics, Simon Fraser University, Burnaby, Canada

1 Introduction

In this paper, we present a highly accurate numerical method for solving the forced isotropic heat equation with Dirichlet data on complex multiply connected domains in two dimensions. We adapt the solution methodology introduced by Kropinski and Quaife in [1], but extend and generalise their work to allow for fewer restrictions on the forcing term, including nonlinear forcing, with improved discretisation in time and uniform accuracy all the way up to the boundary. First, the heat equation is discretised in time with an implicit treatment of the diffusion term, an approach that is sometimes referred to as elliptic marching or Rothe's method [2, 3]. This results in a sequence of modified Helmholtz equations, also known as the linearised Poisson-Boltzmann equation, to be solved at each time step. Doing so advances the solution to the parabolic heat equation in time. A relaxed definition of the modified Helmholtz equation reads $\alpha^2 u - \Delta u = f$, with α^2 inversely proportional to the time step. Utilising the linearity, this equation is further split into two: one that finds a particular solution for the specific right-hand side without enforcing the boundary conditions, and a homogeneous problem that ensures that the sum of the two solutions solves the original problem. The homogeneous problem is solved with a boundary integral method with a panel-based Nyström quadrature scheme, as introduced in [4] by Kropinski and Quaife. The particular solution is written as a volume potential with the free space Green's function for the modified Helmholtz equation, also known as the Yukawa-or screened Columb potential. To avoid constructing quadrature methods for the evaluation of this volume potential over complex domains, an extension of the right-hand side f is introduced, allowing for integration over a simple rectangular domain. In [1] the authors demonstrated the potential of developing an efficient and accurate general boundary integral solver for the heat equation on complex domains. Moreover, they list the major remaining issues that require further investigation. At that time only examples for which a continuous extension of f could be constructed by hand were considered, thus excluding complex geometries and general data. Another impediment was the loss of accuracy for evaluating layer potentials close to their sources. Their solution was to over-resolve the boundary, but the loss of accuracy is still significant as a target point approaches the boundary. In this paper, we introduce the following developments:

- High-order adaptive methods for time evolution.
- A method to efficiently compute a high-regularity extension of a function f to an enclosing and geometrically simple domain, given only its values at discrete locations in Ω .
- A special purpose quadrature method to avoid loss of accuracy when evaluating layer potentials close to the boundary where the kernel becomes nearly singular.

Two main groups of semi-implicit time stepping methods are Runge-Kutta methods [5] and spectral deferred correction methods [6–8]. We use the former, but the approach we propose is general with respect to the choice of semi-implicit time stepper.

It is not a simple problem to construct a high regularity extension of a function, for which only its values are known at discrete points inside the original domain Ω . In

[1], Kropinski and Quaife considered only examples for which a continuous extension could be constructed by hand. We use a partition of unity extension technique (PUX) by Fryklund, Lehto, and Tornberg in [9]. They solve the Poisson equation with the abovementioned split into a particular and an homogeneous problem. We now use this method for function extension in the context of the modified Helmholtz equation with excellent results and can consider a larger class of forcing terms compared with [1]. An alternative approach for function extension is given in [10], where the function to be extended outside of Ω sets the boundary Dirichlet data on $\partial\Omega$ for the Laplace equation in $\mathbb{R}^2 \setminus \Omega$. The solution to this problem is computed with an integral equation–based method, and defines a continuous function extension. See [9] and the references therein for other extension techniques, such as Fourier continuation methods or extending the unknown solution or solution from previous time step [11–13].

When evaluating a layer potential close to a boundary, the kernel becomes nearly singular. A well-known challenge with boundary integral–based methods is accurate numerical integration of singular (for evaluation on the boundary) and nearly singular kernels. The comparative study [14] complemented with [15] gives an overview of state of the art methods. The latter includes panel-based explicit kernel-split schemes with product integration, pioneered by Helsing and Ojala [16] for the Laplace equation. This methodology is applicable to a large class of linear elliptic PDEs, and achieves excellent results also for e.g. the Helmholtz [15] and Stokes equations [17]. However, for the modified Helmholtz equation, product integration may fail altogether for sufficiently large α , i.e., for small time steps in our setting. The quadrature rule will in this case require an unfeasibly high resolution of the boundary, which is not motivated by the geometry nor the resolution requirement for the layer density. This spurred the development of a quadrature scheme to solve this problem. The Yukawa potential decays as $\exp(-\alpha) \sqrt{\pi}/(2\alpha)$; thus, the kernel becomes more localised as α increases. In this process, the product integration requires an increasing amount of upsampling, but only over a decreasing interval, and hence only local upsampling is needed. In a separate paper [18], we present an adaptive quadrature scheme in the spirit of [19] that lifts the previous restriction on α . This scheme is also used to solve the modified Stokes equation in [20].

A parallel development of a boundary integral–based solver for the heat equation is based on direct approximation of the heat kernel, thus avoiding discretisation of the differential operator with respect to time. In the initial work [21], it was observed that to achieve the desired accuracy for domains with high curvature the time step must be considerably smaller than the formal rate of convergence would suggest. The authors refer to this as *geometrically induced stiffness*, a phenomenon specific to this framework. In recent work towards solving the heat equation with said method, Wang et al. have developed a *hybrid method* that allows for evaluation of the boundary and volume potentials including the space-time heat kernel without the constraints from geometric stiffness [22].

Efforts to solve the heat equation with boundary integral equation–based techniques are not only motivated by that specific task. Surely, there are other methods to solve the heat equation on a complex domain, such as finite element methods. However, the algorithmic development in these efforts is essential to increase

the applicability of integral equation–based numerical methods which sport several attractive features, including that complex geometry naturally enters the problem and generation of an unstructured mesh is redundant, ill-conditioning associated with discretising the operators is avoided, high accuracy can be attained, and boundary data and far-field conditions are simple to incorporate. Developments for the heat equation are also related to extension from Stokes to Navier-Stokes equations [20].

The focus of this paper is on the heat equation. However, fast integral equations for the modified Helmholtz equation are of interest for the many applications that equation applies to. These include, but are not limited to, electrostatic interactions in protein and related biological functions, macroscopic electrostatics, Debye–Huckel theory for dilute electrolytes, water wave problems, in the linearisation of the Poisson–Boltzmann equation, and approximation of surfaces [23–28]. Consequently, there is active research on solution methods and analysis thereof for the modified Helmholtz equation. In [29], the method of fundamental solution is used, while in [30] it is solved by plane wave functions.

1.1 Overview of the paper

The mathematical problem is formulated in Section 2, both for the heat equation and the modified Helmholtz equation. Section 3 contains the numerical methods for solving the homogeneous problem and the particular problem for the modified Helmholtz equation, including an introduction to the partition of unity extension technique (PUX). It is assumed that the heat equation has been appropriately discretised in time. Thereafter, we present the numerical results in Section 4, for the modified Helmholtz equation, the heat equation, and a reaction-diffusion type problem. Finally, we present our conclusions and an outlook in Section 5. See Appendix A.1 for more details on how IMEX Runge-Kutta methods reduce the heat equation to a sequence of modified Helmholtz equations. There are simple examples, Butcher tableaux, and a note on adaptivity. In Appendix B, we present a graphical overview of the solution procedure for the modified Helmholtz equation.

2 Formulation

Consider the forced isotropic heat equation:

$$\frac{\partial U(t, \mathbf{x})}{\partial t} - \Delta U(t, \mathbf{x}) = F(t, \mathbf{x}), \quad t_0 < t, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^2, \quad (1)$$

$$U(t_0, \mathbf{x}) = U_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (2)$$

$$U(t, \mathbf{x}) = g(t, \mathbf{x}), \quad \mathbf{x} \in \Gamma, \quad (3)$$

subject to initial data U_0 and Dirichlet boundary data g . To fix notation, let Ω be a time-independent, compact $(N_\Gamma + 1)$ -ply connected region in \mathbb{R}^2 with a boundary Γ consisting of $(N_\Gamma + 1)$ closed curves. These form the set $\Gamma = \{\Gamma_n\}_{n=0}^{N_\Gamma}$, where Γ_0 is the outer boundary of the region Ω ; see Fig. 1. The component curves are individually

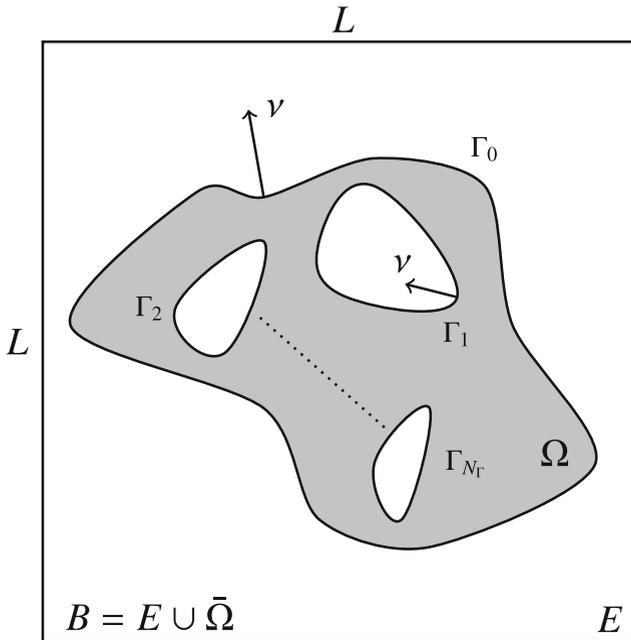


Fig. 1 The heat equation (1)–(3) is defined in Ω . It is enclosed in a box $B = [L, L]^2$ and E is the complement to $\bar{\Omega}$ relative B . The boundaries are denoted $\Gamma_n, n = 0, \dots, N_\Gamma$. The outer boundary is Γ_0 and the outward directed normal is denoted by ν

smooth and a parametrisation for each is assumed to be known. The outward directed normal at $\mathbf{y} \in \Gamma$ is denoted $\nu(\mathbf{y}) = \nu_{\mathbf{y}}$ and $\kappa(\mathbf{y})$ denotes the curvature at $\mathbf{y} \in \Gamma$.

2.1 Discretising in time and the modified Helmholtz equation

The heat equation (1) is first discretised in time, an approach known as elliptic marching or Rothe’s method. To prevent severe time step restrictions, an implicit-explicit (IMEX) scheme is used. It consists of using an implicit discretisation of the stiff terms and an explicit one for the nonstiff terms [31]. Regardless of the specifics of the IMEX scheme, to advance the solution U in discrete time, a sequence of modified Helmholtz equations is solved. The modified Helmholtz equation is stated as:

$$\alpha^2 u(\mathbf{x}) - \Delta u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{4}$$

$$u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \tag{5}$$

with u unknown in Ω . The scalar parameter α^2 is inversely proportional to the time step δt ; its explicit form along with f and g is given by the specific IMEX scheme. We use an adaptive IMEX Runge-Kutta method of fourth order in this paper (see Appendix A.1), as we have found it to be very effective. However, what follows holds for any IMEX scheme.

Using the linearity of the differential operator $\alpha^2 - \Delta$, the solution u to (4)–(5) is decomposed into a homogeneous solution u^H and a particular solution u^P , such that

$u(\mathbf{x}) = u^H(\mathbf{x}) + u^P(\mathbf{x})$ for $\mathbf{x} \in \Omega$. First, the particular solution is acquired by solving a free space problem:

$$\alpha^2 u^P(\mathbf{x}) - \Delta u^P(\mathbf{x}) = f^e(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2, \tag{6}$$

$$u(\mathbf{x}) \rightarrow 0, \quad |\mathbf{x}| \rightarrow \infty, \tag{7}$$

assuming the existence of an extension $f^e \in C^k(\mathbb{R}^2)$, for some $k \geq 0$, of the right-hand side f from (4) such that:

$$f^e(\mathbf{x}) = f(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \tag{8}$$

$$\text{supp}(f^e) \subset B = [-L, L]^2, \tag{9}$$

for some finite L . The boundary condition, given by the Dirichlet data g in (5), is satisfied by u if u^H is a solution to:

$$\alpha^2 u^H - \Delta u^H = 0, \quad \mathbf{x} \in \Omega, \tag{10}$$

$$u^H = \tilde{g}(\mathbf{x}) = g(\mathbf{x}) - u^P(\mathbf{x})|_{\Gamma}, \quad \mathbf{x} \in \Gamma. \tag{11}$$

In short, first solve the free space problem (6) to obtain the boundary data for the homogeneous problem (10)–(11). The solution to the modified Helmholtz equation is the sum of the two solutions, $u(\mathbf{x}) = u^H(\mathbf{x}) + u^P(\mathbf{x})$ for $\mathbf{x} \in \Omega$. See the flowchart in Appendix B for a graphical overview.

2.1.1 The inhomogeneous modified Helmholtz equation

Let $\hat{u}^P = \hat{u}^P(\xi)$ and $\hat{f}^e = \hat{f}^e(\xi)$ denote the Fourier transforms for u^P and f^e , respectively. Here $\xi = [\xi_1, \xi_2] \in \mathbb{R}^2$ with $\xi = |\xi|$. Then under the Fourier transform (6) is:

$$\alpha^2 \hat{u}^P(\xi) + \xi^2 \hat{u}^P(\xi) = \hat{f}^e(\xi), \quad \xi \in \mathbb{R}^2 \tag{12}$$

and we obtain

$$\hat{u}^P(\xi) = \frac{\hat{f}^e(\xi)}{\alpha^2 + \xi^2}, \quad \xi \in \mathbb{R}^2. \tag{13}$$

Note that the above expression is free of singularities, since $\alpha^2 \neq 0$. The solution to (6)–(7) is given by the inverse Fourier transform:

$$u^P(\mathbf{x}) = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \hat{u}^P(\xi) e^{i\xi \cdot \mathbf{x}} d\xi. \tag{14}$$

For it to be well defined, the extension f^e must be in $L^1(\mathbb{R}^2)$. How to construct said extension and compute an approximation of u^P is presented in 3.1.

2.1.2 The homogeneous modified Helmholtz equation

Consider the homogeneous modified Helmholtz equation (10)–(11). The free-space Green’s function $G(\mathbf{x}, \mathbf{y})$ for the operator $\alpha^2 - \Delta$ is:

$$G(\mathbf{x}, \mathbf{y}) = \frac{\alpha^2}{2\pi} K_0(\alpha \|\mathbf{y} - \mathbf{x}\|), \tag{15}$$

where K_0 denotes the zeroth-order modified Bessel function of the second kind. In other contexts, the kernel $G(\mathbf{x}, \mathbf{y})$ is also referred to as the Yukawa or screened Coulomb potential. As in [1, 4], we seek the solution $u^H(\mathbf{x})$ for $\mathbf{x} \in \Omega$ in the form of a double layer potential:

$$u^H(\mathbf{x}) = \frac{\alpha^2}{2\pi} \int_{\Gamma} M(\mathbf{x}, \mathbf{y}) \mu(\mathbf{y}) ds_{\mathbf{y}}, \quad \forall \mathbf{x} \in \Omega, \tag{16}$$

with the kernel

$$M(\mathbf{x}, \mathbf{y}) = \frac{\partial}{\partial \nu_{\mathbf{y}}} K_0(\alpha \|\mathbf{y} - \mathbf{x}\|) = -\alpha K_1(\alpha \|\mathbf{y} - \mathbf{x}\|) \frac{\mathbf{y} - \mathbf{x}}{\|\mathbf{y} - \mathbf{x}\|} \cdot \nu_{\mathbf{y}}, \tag{17}$$

where K_1 denotes the first-order modified Bessel function of the second kind. The limiting value as \mathbf{x} goes to \mathbf{y} along a boundary segment Γ_n is well defined:

$$\lim_{\mathbf{x} \rightarrow \mathbf{y}} M(\mathbf{x}, \mathbf{y}) = -\frac{1}{2} \kappa(\mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \Gamma_n, \tag{18}$$

where $\kappa(\mathbf{y})$ is the curvature of Γ_n at $\mathbf{y} \in \Gamma_n, n = 0, 1, \dots, N_{\Gamma}$. The double layer density $\mu : \Gamma \rightarrow \mathbb{R}$ is not known a priori; it is found through the solution of a boundary integral equation. Such an equation of the second kind for μ can be formulated as:

$$\mu(\mathbf{x}) - \frac{1}{\pi} \int_{\Gamma} M(\mathbf{x}, \mathbf{y}) \mu(\mathbf{y}) ds_{\mathbf{y}} = -\frac{2}{\alpha^2} \tilde{g}(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma. \tag{19}$$

For a derivation, see e.g. [32]. For $\tilde{g} \equiv 0$ only the trivial solution $\mu \equiv 0$ along Γ satisfies (19). Thus, by the Fredholm alternative, the solution μ exists and is unique for any integrable \tilde{g} , for both simply and multiply connected domains [33]. This property is inherited by the corresponding discretised systems as well, introduced in Section 3.2.

Each contour Γ_n is split into $N_{P,n}$ intervals, referred to as panels, where $\Gamma_{n,k}$ is the k th panel on the n th contour and N_P the total number of panels over Γ . A panel $\Gamma_{n,k}$ is represented by a known parametrisation $\boldsymbol{\gamma}_{n,k}$, such that:

$$\Gamma_{n,k} = \{\boldsymbol{\gamma}_{n,k}(t) \mid t \in [-1, 1]\}. \tag{20}$$

By introducing a speed function $s_{n,k}(t) = |\boldsymbol{\gamma}'_{n,k}(t)|$ and $\mu_{n,k}(t) = \mu(\boldsymbol{\gamma}_{n,k}(t))$ the layer potential (16) can be written as:

$$u^H(\mathbf{x}) = \frac{\alpha^2}{2\pi} \sum_{n=0}^{N_{\Gamma}} \sum_{k=1}^{N_{P,n}} \int_{-1}^1 M(\mathbf{x}, \boldsymbol{\gamma}_{n,k}(t)) \mu_{n,k}(t) s_{n,k}(t) dt, \quad \forall \mathbf{x} \in \Omega \tag{21}$$

and analogously for the boundary integral equation (19):

$$\mu(\mathbf{x}) - \frac{1}{\pi} \sum_{n=0}^{N_{\Gamma}} \sum_{k=1}^{N_{P,n}} \int_{-1}^1 M(\mathbf{x}, \boldsymbol{\gamma}_{n,k}(t)) \mu_{n,k}(t) s_{n,k}(t) dt = -\frac{2}{\alpha^2} \tilde{g}(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma. \tag{22}$$

3 Discretisation

This section covers the numerical treatment of the modified Helmholtz equation. Note that two different methods are needed, one for the inhomogeneous problem and one for the homogeneous problem. We assume some suitable IMEX scheme has been chosen for temporal discretisation of the heat equation (1)–(3), e.g., the Runge-Kutta methods presented in Appendix A.1.

Consider a box $B = [-L, L]^2$ in \mathbb{R}^2 that contains $\bar{\Omega}$. The complement of $\bar{\Omega}$ relative to B is denoted by E . Denote the *grid* by \mathbf{X} , which is a set of N_u^2 elements \mathbf{x} , referred to as nodes or points. They are uniformly distributed with spacing δx over B . Let subscripts indicate subsets of \mathbf{X} , such as $\mathbf{X}_\Omega = \{\mathbf{x} \in \mathbf{X} | \mathbf{x} \in \Omega\}$ and $\mathbf{X}_E = \{\mathbf{x} \in \mathbf{X} | \mathbf{x} \in E\}$.

The solution to the modified Helmholtz equation is computed at all grid points that fall inside Ω , i.e., the elements of \mathbf{X}_Ω . First, we present how to find this solution for the free space problem (6)–(7). This involves extending the function f , based on the data at $\mathbf{X}_\Omega = \{\mathbf{x} \in \mathbf{X} | \mathbf{x} \in \Omega\}$ to $\mathbf{X}_E = \{\mathbf{x} \in \mathbf{X} | \mathbf{x} \in E\}$ such that it satisfies (8)–(9). Thereafter, we consider the homogeneous problem (10)–(11), formulated as a boundary integral equation on Γ . The solution is computed at the locations $\mathbf{X}_\Omega = \{\mathbf{x} \in \mathbf{X} | \mathbf{x} \in \Omega\}$ in a post-processing step.

3.1 The inhomogeneous problem and function extension

An approximate solution to the free-space problem (6)–(7) is computed by discretising the integral in (14) with the trapezoidal rule. It is evaluated efficiently with FFTs

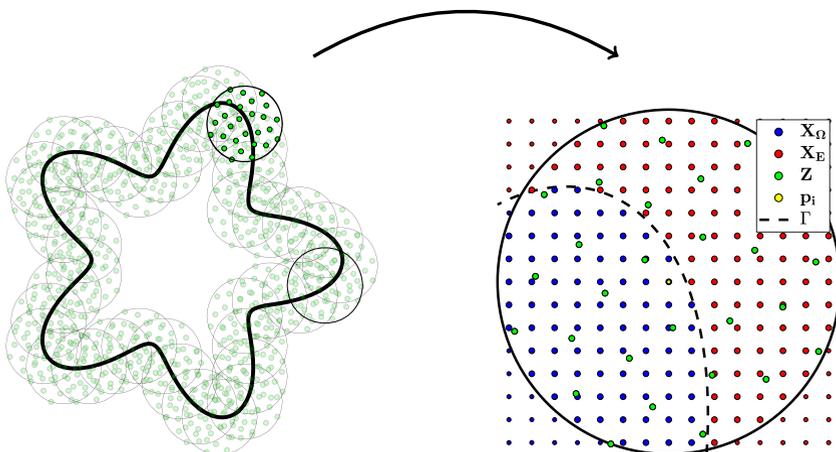


Fig. 2 Left: Schematic figure of distribution of extension partitions along Γ for a complex domain. The green markers correspond to RBF centres \mathbf{Z} , generated by (39), and the distribution is repeated for every partition. Right: Classification of points in \mathbf{X} as inside or outside Ω . The larger markers denote points in $\mathbf{X}_{i,\Omega}$ and $\mathbf{X}_{i,E}$

on the regular grid \mathbf{X} in B , thus in \mathbf{X}_Ω as well, and on the boundary Γ with a non-uniform inverse FFT. The latter is used to modify the given Dirichlet boundary data (5) for the homogeneous modified Helmholtz equation.

If the compactly supported f^e in (6) is smooth, then the coefficients in the Fourier series expansion decay exponentially fast with the wave number, and this procedure would be specially accurate. With limited regularity, the Fourier coefficients instead decay algebraically, with one additional order for each continuous derivative. Numerically, it is constructed with PUX, which is briefly reviewed in this subsection. The basic concept is to blend local extensions by a partition of unity into a global extension with compact support, enforced by weight functions. The global regularity of the extension is directly related to the construction of said partition of unity. This is achieved by distributing overlapping partitions along the boundary Γ of Ω . In each partition the local values of f are used to extend it to the points in the partition that fall outside Ω . For a more extensive treatment, see the original work [9].

3.1.1 Partition of unity

Let $\{\psi_i^k\}_{i=1}^{N_\psi}$ be a collection of N_ψ compactly supported radial functions such that $\psi_i^k(\mathbf{x}) = \psi^k(\mathbf{x} - \mathbf{p}_i)$ for some choice of centres $\{\mathbf{p}_i\}_{i=1}^{N_\psi}$. The superscript k indicates the highest regularity subset C_0^k of C_0 that ψ^k is a member of. Define a partition Ω_i as the support of ψ_i^k , i.e. $\Omega_i = \text{supp}(\psi_i^k)$, which is a disc with radius R . The choice of ψ^k is discussed in Section 3.1.3. Note that all partitions have the same radius. The number of partitions N_ψ , the location of the partition centres $\{\mathbf{p}_i\}_{i=1}^{N_\psi}$ and radius R are chosen such that the partitions cover Γ and that the partitions overlap with approximately a radius. The following notation will be useful. Each partition Ω_i has a set of points on the uniform grid within R of \mathbf{p}_i , which we denote \mathbf{X}_i , rather than \mathbf{X}_{Ω_i} . It can be split into two disjoint subsets: $\mathbf{X}_{i,\Omega} = \{\mathbf{x} \in \Omega_i \cap \Omega\}$ and $\mathbf{X}_{i,E} = \{\mathbf{x} \in \Omega_i \cap E\}$. Let N_i denote the number of elements in \mathbf{X}_i . Analogously, let $N_{i,\Omega}$ and $N_{i,E}$ denote the number of elements in $\mathbf{X}_{i,\Omega}$ and $\mathbf{X}_{i,E}$, respectively. See Fig. 2 for a graphical example. Given a function $f : \Omega \rightarrow \mathbb{R}$, the function

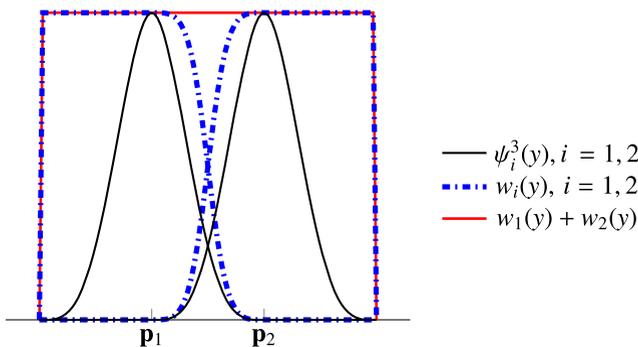


Fig. 3 Plot of weight functions (23) and their sum

values at the locations $\mathbf{X}_{i,\Omega}$ are used to create a local extension f_i^e . We will return to the construction of the local extensions in Section 3.1.2, but for now assume their existence.

For every partition Ω_i and its associated basis function ψ_i^k , define the corresponding weight function w_i as:

$$w_i(\mathbf{x}) = \frac{\psi_i^k(\mathbf{x})}{\sum_{j=1}^{N_\psi} \psi_j^k(\mathbf{x})}, \tag{23}$$

which belongs to the space C_0^k . By construction, the set of weights $\{w_i\}_{i=1}^{N_\psi}$ forms a partition of unity. That is:

$$\sum_{i=1}^{N_\psi} w_i(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in \bigcup_{i=1}^{N_\psi} \bar{\Omega}_i, \tag{24}$$

which is referred to in the literature as Shepard’s method [34]. See Fig. 3 for a visualisation. This construction is used to combine the local extension $\{f_i^e\}_{i=1}^{N_\psi}$ into a global one:

$$f^e(\mathbf{x}) = \sum_{i=1}^{N_\psi} w_i(\mathbf{x}) f_i^e(\mathbf{x}). \tag{25}$$

However, (25) is not used, as we want an extension that it is continuous or of higher regularity as it is extended by zero outside its support. Refer to the set of partitions $\{\Omega_i\}_{i=1}^{N_\psi}$ as *extension* partitions and now introduce also the *zero* partitions $\{\Omega_i^0\}_{i=1}^{N_\psi}$. They are included in the partition of unity definition (24) and distributed such that they overlap the extension partitions, but do not intersect $\bar{\Omega}$. The associated local extension f_i^e is set to be identically equal to zero for $i = 1, \dots, N_\psi^0$. Hence, as the zero partitions are blended with the local extensions in the first layer of partitions

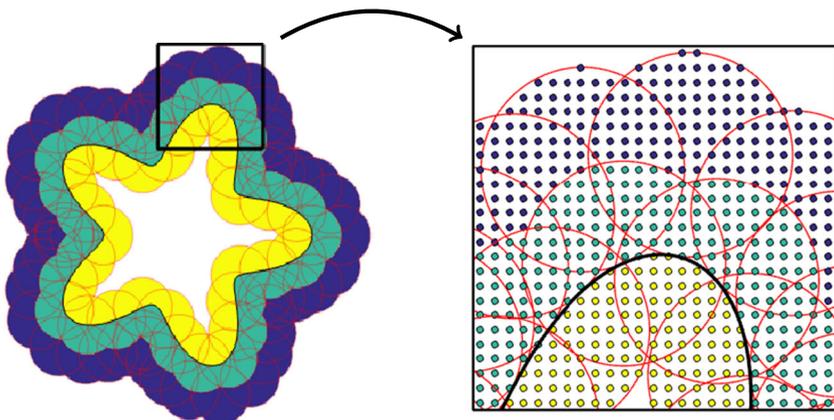


Fig. 4 Schematic image for function extension from a star-shaped domain Ω given by the black border. Observe that in this figure the partitions are not centred at uniform grid points. The red overlapping circles are the partitions. The yellow section corresponds to the uniform data points used for creating the local extension f_i^e , the blue section to points where $f^e = 0$, and the green sector is a blend of the two

$\{\Omega_i\}_{i=1}^{N_\psi}$, the global extension will be forced to zero over the overlapping region. Therefore, zero partitions should be placed such that f^e has a controlled decay to zero and the size of the overlap with extension partitions are about the same; see Fig. 4. Thus, the global extension will in these parts have the same regularity as w^p , as given by the regularity of the compactly supported basis function ψ^k . The extension f^e of f is given by:

$$f^e(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \mathbf{x} \in \mathbf{X}_\Omega, \\ \sum_{i=1}^{N_\psi+N_\psi^0} w_i(\mathbf{x})f_i^e(\mathbf{x}), & \mathbf{x} \in \bigcup_{i=1}^{N_\psi} \mathbf{X}_{i,E}, \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

As ψ^k , we use one of the compactly supported Wu-functions, which are tabulated after their regularity k ; see Table 1 or [35]. There are other options, but the Wu-functions have compact support and are simple to implement. Note that they have lower regularity at the origin, e.g., the Wu-function listed as C^4 is only C^2 at that point. Moreover, the $(k + 1)$ th derivative of ψ^k is of bounded variation. The partition centres $\{\mathbf{p}_i\}_{i=1}^{N_\psi}$ are set to be nodes on the regular grid that are the closest to be boundary, yet still in $\mathbf{X}_{i,\Omega}$. Thus, evaluation of weight functions at the origin is omitted and higher regularity is maintained. With this, we have described how local extensions are combined into a global one. It remains to construct the local extensions $\{f_i^e\}_{i=1}^{N_\psi}$.

3.1.2 Local extensions

We now return to the construction of the local extensions f_i^e for each extension partition $i = 1, \dots, N_\psi$. The local extension f_i^e is created as a weighted sum of radial basis functions that interpolates the values of f at $\mathbf{x} \in \mathbf{X}_{i,\Omega}$ and is evaluated at $\mathbf{x} \in \mathbf{X}_{i,E}$. The radial basis functions (RBFs) are denoted $\phi_j(\mathbf{x}) = \phi(\|\mathbf{z}_j - \mathbf{x}\|)$. The elements of the set $\mathbf{Z} = \{\mathbf{z}_j\}_{j=1}^{N_\phi} \subset \text{supp}(\psi_i^k)$ are the N_ϕ centres for the RBFs, whose distribution for now is left unspecified. The standard form of an RBF interpolant at a point \mathbf{x} is:

$$f_i^e(\mathbf{x}) = \sum_{j=1}^{N_\phi} \lambda_j \phi(\|\mathbf{z}_j - \mathbf{x}\|), \tag{27}$$

Table 1 Wu-functions $\psi^k \in C_0^k$, with compact support in $r \in (0, 1)$ [35]

Regularity	$\psi^k(r)$
$\psi^1 \in C_0^1$	$(1 - r)_+^2(2 + r)$
$\psi^2 \in C_0^2$	$(1 - r)_+^3(8 + 9r + 3r^2)$
$\psi^3 \in C_0^3$	$(1 - r)_+^4(4 + 16r + 12r^2 + 3r^3)$
$\psi^4 \in C_0^4$	$(1 - r)_+^5(8 + 40r + 48r^2 + 25r^3 + 5r^4)$
$\psi^5 \in C_0^5$	$(1 - r)_+^6(6 + 36r + 82r^2 + 72r^3 + 30r^4 + 5r^5)$

Here, $(\cdot)_+ = \max(0, \cdot)$. The listed regularity excludes evaluation at the origin

where λ_j are unknown coefficients to be determined. We use:

$$\phi(\|\mathbf{z}_j - \mathbf{x}\|) = e^{-(\varepsilon\|\mathbf{z}_j - \mathbf{x}\|)^2}, \tag{28}$$

where ε is a shape parameter setting the width of the Gaussian. The smallest interpolation error is obtained when ε is small, yet nonzero, but no general value can be given [36].

With some abuse of notation let \mathbf{X}_i and \mathbf{Z} refer to vectors with the members of respective set as elements. Then, following the outline of [37], let $\Phi(\mathbf{X}_i, \mathbf{Z})$ denote an $N_i \times N_\phi$ -matrix with elements $\Phi(\mathbf{X}_i, \mathbf{Z})_{m,n} = \phi(\|\mathbf{x}_m - \mathbf{z}_n\|)$, for $m = 1, \dots, N_i$ and $n = 1, \dots, N_\phi$. Furthermore, let $\Lambda = (\lambda_1 \lambda_2 \dots \lambda_{N_\phi})^T$. Consider a scenario when f is known for all nodes in \mathbf{X}_i , then the associated interpolation problem to (27) can be written as:

$$\Phi(\mathbf{X}_i, \mathbf{Z})\Lambda = f_{\mathbf{X}_i}, \tag{29}$$

with $f_{\mathbf{X}_i} = f(\mathbf{X}_i)$. If $N_i \geq N_\phi$ then Λ can be solved for in a least squares sense. However, this is an unstable problem for several reasons. First, the conditioning of the problem is heavily dependent on the shape parameter ε . For small ε , the interpolation weights Λ oscillate between positive and negative numbers of large magnitude [36]. Furthermore, it is not uncommon for the condition number for the interpolation matrix to be of order 10^{18} or more. These characteristics are common for interpolation with radial basis functions. Additionally, the data is represented on a uniform grid; collocating at these locations is the worst possible setting for interpolation, as with polynomials. These shortcomings can be circumvented by avoiding collocation and considering a least squares problem instead. Note that all problems mentioned above are purely numerical artifacts. The function space spanned by Gaussians is indeed a good approximation space.

Decouple the centres \mathbf{Z} of the radial basis functions from \mathbf{X}_i and assume they are distributed in a near optimal way with respect to minimising the interpolation error. We wish to omit explicit use of the interpolation coefficients Λ in (29). It can be achieved by formally solving for Λ by collocating at the centres \mathbf{Z} :

$$\Phi(\mathbf{Z}, \mathbf{Z})\Lambda = f_{\mathbf{Z}} \Leftrightarrow \Lambda = \Phi(\mathbf{Z}, \mathbf{Z})^{-1} f_{\mathbf{Z}}. \tag{30}$$

Here, $f_{\mathbf{Z}}$ are the values of f at the locations \mathbf{Z} , which are unknown. Due to the choice (28), the matrix $\Phi(\mathbf{Z}, \mathbf{Z})$ is symmetric and positive definite, thus the inverse $\Phi(\mathbf{Z}, \mathbf{Z})^{-1}$ is well defined. We can now reformulate (29) as:

$$\Phi(\mathbf{X}_i, \mathbf{Z})\Phi(\mathbf{Z}, \mathbf{Z})^{-1} f_{\mathbf{Z}} = f_{\mathbf{X}_i}. \tag{31}$$

Henceforth, we use the shorthand notation $A(\mathbf{X}_i, \mathbf{Z}) = \Phi(\mathbf{X}_i, \mathbf{Z})\Phi(\mathbf{Z}, \mathbf{Z})^{-1}$. For the purpose of function extension, sort the data points in Ω_i such that:

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{X}_{i,\Omega} \\ \mathbf{X}_{i,E} \end{bmatrix}, \tag{32}$$

where the components are of length $N_{i,\Omega}$ and $N_{i,E}$, respectively. Consequently, A can also be rearranged and split into two block matrices:

$$A = \begin{bmatrix} A_{i,\Omega} \\ A_{i,E} \end{bmatrix}, \tag{33}$$

with $A_{i,\Omega} = \Phi(\mathbf{X}_{i,\Omega}, \mathbf{Z})\Phi(\mathbf{Z}, \mathbf{Z})^{-1}$ of size $N_{i,\Omega} \times N_\phi$ and $A_{i,E} = \Phi(\mathbf{X}_{i,E}, \mathbf{Z})\Phi(\mathbf{Z}, \mathbf{Z})^{-1}$ of size $N_{i,E} \times N_\phi$. Since f is known at $\mathbf{X}_{i,\Omega}$, it can replace the corresponding entries in $f_{\mathbf{X}_i}$ (31) with $f_{i,\Omega} = f(\mathbf{X}_{i,\Omega})$. For each partition, we obtain the system:

$$\begin{bmatrix} A_{i,\Omega} \\ A_{i,E} \end{bmatrix} f_{\mathbf{Z}} = \begin{bmatrix} f_{i,\Omega} \\ f_{i,E} \end{bmatrix}, \tag{34}$$

with $f_{i,E} = f(\mathbf{X}_{i,E})$ unknown. The values $f_{i,\Omega}$ are mapped to the nodes \mathbf{Z} to obtain $f_{\mathbf{Z}}$. Thereafter, we obtain $f_{i,E}$, which is the local extension. That is:

1. Solve the least squares problem $A_{i,\Omega}f_{\mathbf{Z}} = f_{i,\Omega}$ for $f_{\mathbf{Z}}$.
2. Obtain the local extension $f_i^e(\mathbf{X}_{i,E}) = f_{i,E} = A_{i,E}f_{\mathbf{Z}}$.

This approach allows us to use a non-uniform distribution of RBF centres which significantly improves the stability, but still lets the data be represented on the uniform grid. We also avoid explicit use of the interpolation weights Λ . It remains to address the notorious ill-conditioning of Φ , associated with the shape parameter ε set small. This is achieved by applying the algorithm *RBF-QR*. It is intended for a formulation as (34), since it computes A , rather than Φ^{-1} , which acts as a mapping of data from non-uniformly to uniformly distributed locations. Said algorithm performs a change of basis for A , and in the process the condition number is reduced; see [38]. By the use of RBF-QR, the restrictions of choosing ε are lifted.

3.1.3 Properties of PUX

Three parameters need to be set for the PUX algorithm: the shape parameter ε for the width of Gaussians (28) used as interpolation basis, the partition radius R , and N_u , where N_u^2 is the number of uniformly distributed nodes over B . The remaining parameters can be set based on these values. For a complete discussion, see [9].

Due to the stabilising effect of RBF-QR, the shape parameter can be set small without risk of suffering from ill-conditioning. A good value is $\varepsilon = 2$, but the error in solving the modified Helmholtz equation is relatively insensitive. The parameter L is set by the partition radius R , as the box $[-L, L]^2$ only needs to be large enough to contain the support of f^e .

Let P be the number of uniform grid points per partition radius, denoted as:

$$P = \frac{N_u}{2L} R. \tag{35}$$

This measure is used to choose ψ^k from Table 1, and the number N_ϕ of basis functions (28) per partition. To see how P relates to ψ^k , consider the convergence of the error in solving the modified Helmholtz equation (4)–(5), assuming that only resolving u^P limits the accuracy. If f^e is smooth, then the error has asymptotically spectral convergence. However, the extension inherits the regularity of the weight function w . Recall that by construction $w \in C_0^k$ for a fixed ψ^k (23). Consequently, the error has an asymptotic convergence of $4 + k$: 2 orders for solving the modified Helmholtz equation and $k + 2$ orders since the $(k + 1)$ th derivative of f^e is of bounded variation [39]. A Wu-function of high regularity is harder to resolve than one of lower regu-

larity. This implies that given a resolution P , the error in resolving the Wu-function may hamper the convergence. As in [9], we use the heuristic relation:

$$k = \min \left(\left\lfloor \sqrt{P} - 0.9 \right\rfloor, 5 \right) \tag{36}$$

for choosing ψ^k . In Section 4, we confirm that (36) yields a satisfactory choice for ψ^k given P .

Creating a local extension involves solving the least squares problem $A_{i,\Omega} f_{\mathbf{Z}} = f_{i,\Omega}$ for $f_{\mathbf{Z}}$ for some i . It should be sufficiently overdetermined in order to be a well-posed problem, but too much is potentially numerically unstable. Given a P , the number of unknowns N_ϕ should be set accordingly to obtain a certain ratio of knowns and unknowns. We use:

$$N_\phi = \lfloor 3P \rfloor \tag{37}$$

to set the number of radial basis functions per partition. The data available for the least squares problem is downsampled with the sampling parameter c , defined as:

$$c = \max \left(\left\lfloor \sqrt{\frac{P}{8}} \right\rfloor, 1 \right). \tag{38}$$

If $c = 1$, then all points are used, $c = 2$ means that every other point is removed, etc. For each local extension, we will solve a least squares problem of size $(\beta N_\phi) \times N_\phi$. The factor β depends also on how large portion of the partition that belongs to the interior. In most instances, $2 \leq \beta \leq 5$.

We should remark that the benefits of increased resolution in resolving the Wu functions, as discussed above, still remain even if not all the data is used for the least squares problem.

The distribution of RBF centres \mathbf{Z} can be chosen freely, and we use the quasi-uniform *Vogel* node distribution defined as:

$$\mathbf{z}_j = \sqrt{\frac{j}{N_\phi}} \left(\cos \left(j\pi \left(3 - \sqrt{5} \right) \right), \sin \left(j\pi \left(3 - \sqrt{5} \right) \right) \right), \quad j = 1, \dots, N_\phi, \tag{39}$$

in a unit disc. See Fig. 2 for a visualisation. The distribution (39) is near optimal and RBF-QR performs well up to about 400 nodes. The locality of the weight functions guarantees that the least squares systems are of moderate size, which can be solved in parallel.

Constructing A (33) with RBF-QR is a computationally expensive operation, so employing it for every partition is undesirable. However, the matrix is the same for all partitions since \mathbf{p}_i is centred at a grid point from the uniform distribution. Thus, the pairwise distances for the elements in \mathbf{X}_i are independent of i . Therefore, a single matrix A can be precomputed with RBF-QR and reused for all extension partitions. The only difference between them in terms of A is the decomposition of \mathbf{X}_i into $\mathbf{X}_{i,\Omega}$ and $\mathbf{X}_{i,E}$, as it depends on how the boundary Γ intersects the partition. Note that the zero partitions may individually have a radius different from R in order to conform to the geometry of Ω and to overlap the extension partitions properly. All weights

$\{w_i\}_{i=1}^{N_\psi+N_\psi^0}$ are computed once in a precomputation step and are reused. After pre-computations, the PUX procedure only involves solving a least squares problem and a matrix vector multiplication for each partition, followed by a blending of the extended data.

3.2 The homogeneous problem

For simplicity, assume the number of contours to be one and write $\Gamma_{n,k} = \Gamma_k$, $\mathbf{y}_{n,k} = \mathbf{y}_k$, and $s_{n,k} = s_k$. We apply an N_Q -point, panel-based Nyström discretisation scheme based on the composite Gauss-Legendre quadrature rule, with nodes t_m^G and weights W_m^G , with $m = 1, \dots, N_Q$. Let $\mathbf{y}_{k,m} = \mathbf{y}_k(t_m^G)$, $s_{k,m} = s_k(t_m^G)$, and $\mu_{k,m} = \mu_k(t_m^G)$. An approximation of the solution μ to (22) is the solution of:

$$\mu_{i,j} - \frac{1}{\pi} \sum_{k=1}^{N_P} \sum_{m=1}^{N_Q} M(\mathbf{x}_{i,j}, \mathbf{y}_{k,m}) \mu_{k,m} s_{k,m} W_m^G = -\frac{2\tilde{g}_{i,j}}{\alpha^2}, \quad i = 1, \dots, N_P, \quad j = 1, \dots, N_Q, \tag{40}$$

and correspondingly for (21) we have:

$$u^H(\mathbf{x}) = \frac{\alpha^2}{2\pi} \sum_{k=1}^{N_P} \sum_{m=1}^{N_Q} M(\mathbf{x}, \mathbf{y}_{k,m}) \mu_{k,m} s_{k,m} W_m^G, \quad \mathbf{x} \in \Omega. \tag{41}$$

An important observation is that the kernel M (17) is not smooth and can contain singularities, depending on how \mathbf{x} approaches \mathbf{y} . Here, the Gauss-Legendre quadrature rule is insufficient, as the resulting loss of accuracy can be critical enough to render the result useless. We elaborate on this topic in Section 3.2.1.

In matrix notation, (40) can be written as $(\mathbf{I} + \mathbf{M})\boldsymbol{\mu} = \tilde{\mathbf{g}}$, where \mathbf{I} is the identity matrix and \mathbf{M} a compact operator. The density $\boldsymbol{\mu}$ can be efficiently obtained with GMRES, in terms of numbers of iterations. The condition number for $\mathbf{I} + \mathbf{M}$ is typically small or moderate and uniformly bounded. A fast multipole method (FMM) can be used for efficient computation of the involved potentials in (40) and (41) [40]. We use the point to point FMM for the two-dimensional Yukawa kernel presented in [4]. It is based on the volume equivalent in [41]. For the corresponding three-dimensional version, see [42].

Finally, a note on the restriction of the boundaries being smooth. For non-smooth boundaries, the integrand of (22) is not compact and the Fredholm alternative fails. While there are theoretical results on the solvability with Lipschitz continuous boundaries [43], they require sophisticated quadrature techniques, such as [44], which we have not implemented. These methods also allow cusps, i.e., non-Lipschitz boundaries, and mixed boundary conditions.

3.2.1 Special purpose quadrature

When solving for μ in (40) or evaluating the layer potential (41), several orders of accuracy may be lost, since the kernel M (17) is not smooth. Moreover, M can be singular, depending on if \mathbf{x} approaches some $\mathbf{y} \in \Gamma$ along Γ or from Ω . One of the most efficient methods to circumvent this loss of accuracy is explicit kernel-split

quadrature with product integration by Helsing (see [19]). However, for the modified Helmholtz equation with large α , i.e., for high temporal resolution, it can fail completely. Below we sketch the problem, its relation to α , and how to circumvent it.

We start by explaining product integration, which requires the involved integrals to be expressed in complex notation. To keep these paragraphs brief and simple, the reformulations are omitted. Consider a single panel $\Gamma_k \in \mathbb{C}$ with endpoints at -1 and 1 , but the panel does not have to follow the real axis. Let $\varphi : \Gamma_k \rightarrow \mathbb{R}$ be a smooth function and $S : \Gamma_k \times \mathbb{C} \rightarrow \mathbb{R}$ a non-smooth kernel that may be singular or nearly singular. The goal is to compute:

$$\int_{\Gamma_k} \varphi(\tau)S(\tau_0, \tau) d\tau \tag{42}$$

accurately for some fixed $\tau_0 \in \mathbb{C}$ arbitrarily close to or on Γ_k . To do this, approximate φ with a polynomial of degree $N_Q - 1$ in $\tau \in \Gamma_k$, such that:

$$\varphi(\tau_0) \approx \sum_{n=1}^{N_Q} c_n \tau^{n-1}, \tag{43}$$

with unknown coefficients $\{c_n\}$. Inserting this into (42) gives:

$$\int_{\Gamma_k} \varphi(\tau)S(\tau_0, \tau) d\tau \approx \sum_{n=1}^{N_Q} c_n \int_{\Gamma_k} \tau^{n-1} S(\tau_0, \tau) d\tau. \tag{44}$$

In our setting, the integrals on the right-hand side can be computed analytically through recursive formulas. The unknown coefficients $\{c_n\}$ are obtained by solving a Vandermonde system. If φ can be accurately represented as a $N_Q - 1$ degree polynomial over Γ_k , then product integration allows evaluation of integrals such as (44) without loss of accuracy as τ_0 and τ approach each other.

Kernel-split means that a kernel is decomposed into smooth and singular terms. Leaving complex notation, by [45, Section 10] the first-order modified Bessel function of the second kind K_1 , appearing in (17), can be decomposed as:

$$K_1(x) = \frac{1}{x} + I_1(x) \log(x) + K_1^S(x), \quad x \in \mathbb{R}^+. \tag{45}$$

This form is attractive since the singular terms are separated and can be studied individually. Here, I_1 is the modified Bessel function of the first kind of order one and K_1^S is a power series in x . For the kernel M (see (17)), the situation is slightly more involved, as the singularity structure depends on how \mathbf{x} approaches $\mathbf{y} \in \Gamma$. To distinguish between the two cases, for any $\mathbf{y} \in \Gamma$ denote $M(\mathbf{x}, \mathbf{y})$ as $M_\Gamma(\mathbf{x}, \mathbf{y})$ for $\mathbf{x} \in \Gamma$ and $M_\Omega(\mathbf{x}, \mathbf{y})$ for $\mathbf{x} \in \Omega$. We first study M_Γ . The decomposition (45) motivates the formulation:

$$M_\Gamma(\mathbf{x}, \mathbf{y}) = M_{\Gamma,0}(\mathbf{x}, \mathbf{y}) + \log(\|\mathbf{y} - \mathbf{x}\|)M_{\Gamma,L}(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \Gamma, \tag{46}$$

with $M_{\Gamma,L}$ identified as:

$$M_{\Gamma,L}(\mathbf{x}, \mathbf{y}) = \alpha I_1(\alpha \|\mathbf{y} - \mathbf{x}\|) \frac{\mathbf{y} - \mathbf{x}}{\|\mathbf{y} - \mathbf{x}\|} \cdot \nu_{\mathbf{y}}, \quad \mathbf{x}, \mathbf{y} \in \Gamma. \tag{47}$$

The term $M_{\Gamma,0}$ is smooth and by (18) we have:

$$M_{\Gamma,0}(\mathbf{y}, \mathbf{y}) = -\frac{1}{2}\kappa(\mathbf{y}), \quad \mathbf{y} \in \Gamma, \tag{48}$$

since the term $\log(\|\mathbf{x}-\mathbf{y}\|)M_{\Gamma,L}(\mathbf{x}, \mathbf{y})$ goes to zero in the limit $\mathbf{x} \rightarrow \mathbf{y}$. But in this limit the derivative of $\log(\|\mathbf{y} - \mathbf{x}\|)M_{\Gamma,L}(\mathbf{x}, \mathbf{y})$ has a log-type singularity. Thus, standard quadrature rules that rely on smoothness fail to be accurate. To maintain accuracy, product integration is needed, even though the limit is well-defined. In terms of (44) ϕ and S correspond to $\mu M_{\Gamma,L}$ and \log . This approach is used to compute the involved integrals in (22).

In the case $\mathbf{x} \in \Omega$, corresponding to computing (21), the kernel $M(\mathbf{x}, \mathbf{y})$ is singular in the limit $\mathbf{x} \rightarrow \mathbf{y}$ and product integration is required. We have:

$$M_{\Omega}(\mathbf{x}, \mathbf{y}) = M_{\Omega,0}(\mathbf{x}, \mathbf{y}) + \log(\|\mathbf{y}-\mathbf{x}\|)M_{\Omega,L}(\mathbf{x}, \mathbf{y}) + \frac{(\mathbf{y} - \mathbf{x}) \cdot \nu_{\mathbf{y}}}{\|\mathbf{y} - \mathbf{x}\|^2} M_{\Omega,C}(\mathbf{x}, \mathbf{y}), \quad \mathbf{x} \in \Omega, \mathbf{y} \in \Gamma, \tag{49}$$

where $M_{\Omega,0}$ is a smooth function, $M_{\Omega,L} = M_{\Gamma,L}$ and $M_{\Omega,C} = -1$. Again, we identify ϕ from (44) as μ multiplied with $M_{\Omega,L}$ or $M_{\Omega,C}$ and the singular function S corresponds to either $\log(\|\mathbf{y} - \mathbf{x}\|)$ or $(\mathbf{y} - \mathbf{x}) \cdot \nu_{\mathbf{y}}/\|\mathbf{y} - \mathbf{x}\|^2$. In complex notation, the latter is reduced to a Cauchy-type singularity.

There are two sources of error that may render the results of special purpose quadrature useless: interpolation error and cancellation error. Both $M_{\Gamma,L}$ and $M_{\Omega,L}$ contain the factor $I_1(\alpha\|\mathbf{x} - \mathbf{y}\|)$, which grows asymptotically like $e^{\alpha\|\mathbf{x}-\mathbf{y}\|}/\sqrt{\alpha\|\mathbf{x} - \mathbf{y}\|}$. The scaling with α can make I_1 grow too fast over a single panel to be accurately approximated by e.g. a 15th degree polynomial or even a 31st degree polynomial. The product integration relies on ϕ being well approximated by such a polynomial (43); otherwise, the result may be very inaccurate. Another complicating factor is that $M_{\Omega,0}$ and $M_{\Gamma,0}$ grow with signs opposing $M_{\Omega,L}$ and $M_{\Gamma,L}$. They have to cancel so that the total sum decreases, since K_1 is a decreasing function. For large α , this may lead to cancellation and a significant loss of precision. Recall that $\alpha \sim \delta t^{-1/2}$ as set by an adaptive time stepper.

This problem is not unique to the modified Helmholtz equation, but appears for the modified biharmonic equation and modified Stokes equations as well. One solution is an algorithm presented in a separate paper (see [18]). By local refinement of panels through adaptive recursive bisection, a kernel-split quadrature with product integration can be used successfully for a wide range of α . Based on error estimates, it is ensured that the new panels are of adequate size to accurately approximate ϕ with polynomial interpolation and that the cancellation is less than a set tolerance. Given a target point \mathbf{x} on the boundary, a new panel is created centered around it (see Fig. 5). The length is such that errors are below a set tolerance. Consequently, new neighbouring panels are created, whose layout depends on how the panel centred at \mathbf{x} cuts the original panel configuration. These panels are bisected recursively until a resulting panel does not need special purpose quadrature or can perform it successfully and avoid cancellation errors. If the target point is in the domain, then its projection onto the boundary is the centre on a new panel of appropriate length. Then, the procedure is analogous with on boundary evaluation.

The method is effective in terms of computations, as the increased cost scales as $\log(\alpha)$. Moreover, $K_1(\alpha\|\mathbf{x} - \mathbf{y}\|) \sim \sqrt{\pi}/(\alpha\|\mathbf{x} - \mathbf{y}\|) e^{-\alpha\|\mathbf{x}-\mathbf{y}\|}$ for large arguments,

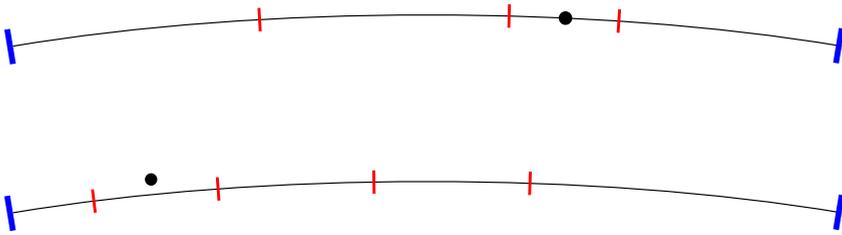


Fig. 5 Illustration of recursive bisection used when α is large. The black dot represents a target point on or near the boundary, represented by the black line. The blue lines are the boundaries of the original panel. The boundaries of the subpanels are indicated by the red lines. They are the result of recursive bisection

i.e. K_1 is very localised for large α and only a small portion of the boundary Γ needs to be upsampled.

Note that the algorithm merely refines the panels, as part of the quadrature method. No extra unknowns are introduced in the solution procedure.

3.3 Computational complexity

Here, we give account for the computationally costly steps. To solve the heat equation a sequence of modified Helmholtz equations is solved per time step and the number of such problems is set by that specific time stepping scheme. Thus, we focus on the complexity of solving the modified Helmholtz equation. The most demanding computational steps, numbered 1–5 in Fig. 16 in Appendix B, are:

1. Construct extension f^e of f with PUX.
2. Solve $\alpha^2 u^P - \Delta u^P = f^e$ in Ω with FFTs.
3. Compute u^P on Γ by NUFFT.
4. Solve $\alpha^2 u^H - \Delta u^H = 0$ by finding the double layer density μ on Γ .
5. Evaluate u^H in Ω .

We begin by discussing steps 2–5. Here, we will use the FFT, the NUFFT, and the FMM. For the FFT, it is solely the grid size that will determine the cost (for a specific implementation). The FMM for N sources and targets is reducing the cost of a $\mathcal{O}(N^2)$ matrix vector product to $\mathcal{O}(N)$, while introducing an approximation. Hence, the computational cost can be bounded by $C_{\text{FMM}}N$, but the constant will depend on the required accuracy. There is an approximation introduced also for the NUFFT; and hence, the constant will also here depend on the required accuracy. Keeping that in mind, we now go through the different steps. Step 2 is solving the free-space problem (6)–(7) and computing the particular solution u^P on the uniform grid. It involves two applications of an FFT, which is of cost $\mathcal{O}(N_u^2 \log N_u^2)$. If $N_{\Gamma, \mathcal{Q}}$ is the total number of discretisation points over all boundary segments, then computing the particular solution on these points by NUFFT, step 3, is of complexity $\mathcal{O}(N_u^2 \log(N_u^2)) + \mathcal{O}(N_{\Gamma, \mathcal{Q}})$. Solving for the double layer density (40) in step 4 is done with GMRES, where an FMM is used for the matrix vector multiplications. The number of GMRES iterations is independent of the resolution at a fixed tolerance, and so the total cost is $\mathcal{O}(N_{\Gamma, \mathcal{Q}})$. Evaluating the homogeneous solution u^H (41) on the uniform grid, step 5,

with an FMM is $\mathcal{O}(N_u^2)$. The special quadrature, that is applied both for solving and evaluating the solution, is a local correction and scales as $\mathcal{O}(N_\Gamma, \varrho)$.

Finally, we now cover step 1. The number of partitions scale as $N_\psi \sim |\Gamma|/R$, where R is the partition radius. We will have N_ψ least squares problems of size $(\beta N_\phi) \times N_\phi$ to solve, where N_ϕ is the number of basis functions in one partition and β is the ratio of available data to unknowns. In most instances $2 \leq \beta \leq 5$, as discussed below (38). The total cost is hence $\mathcal{O}(N_\psi N_\phi^3)$.

The number of basis functions N_ϕ is proportional to P (see (37)), and hence proportional to $N_u R$ (see (35)). Assume that we have a fixed geometry ($|\Gamma|$ is constant) and are increasing the resolution of the uniform grid. With $R \sim 1/\sqrt{N_u}$, we have both N_ψ and N_ϕ proportional to $\sqrt{N_u}$ which yields a total cost of $\mathcal{O}(N_u^2)$, i.e. linear in the number of gridpoints. Linear complexity can hence be achieved if we reduce the partition radius R as the grid resolution is increased. If we keep R constant, then the number of partitions N_ψ will stay fixed, but N_ϕ will grow proportional to N_u . This yields an asymptotic computational cost of $\mathcal{O}(N_u^3) = \mathcal{O}((N_u^2)^{3/2})$, which grows faster than linear. However, since the partitions are small compared with the full domain, the system sizes are moderate, typically only up to a few hundred unknowns for the finest grid resolutions. At this point, the expected asymptotic scaling is not visible yet as discussed in [9, Section 7.4], where timings of the PUX procedure are presented. In practice, the increase in computational time is at most approximately linear. In this paper, we have therefore chosen to keep R constant when we consider convergence with respect to N_u .

4 Numerical results

In this section, we present the numerical results, starting with a study of the modified Helmholtz equation to confirm that the parameters for PUX can be set as in [9] for the Poisson equation. It forms the basis for the second numerical experiment, where the modified Helmholtz equation is solved on a more complex domain. The heat equation is solved on the same domain, for a range of set tolerances with an adaptive time stepper for different grid resolutions. Finally, the Allen-Cahn equation, a reaction-diffusion problem, is solved with randomised initial data.

To compute the errors, we consider an evaluation grid. It consists of N_{eval}^2 uniformly distributed nodes over the computational domain B . We evaluate the numerical solution and an analytical or computed reference solution on the nodes that fall inside Ω . The cardinality of this set of nodes as $N_{\text{eval}, \Omega}$. Two different errors are computed: the relative ℓ_2 -error and the relative discrete ℓ_∞ -error, defined as $\|\mathbf{u}_{\text{solution}} - \mathbf{u}_{\text{numerical}}\|_{\ell_p} / \|\mathbf{u}_{\text{solution}}\|_{\ell_p}$ where:

$$\|\mathbf{u}\|_{\ell_2} = \frac{1}{N_{\text{eval}, \Omega}} \sqrt{\sum_{i=1}^{N_{\text{eval}, \Omega}} |u_i|^2} \tag{50}$$

and

$$\|\mathbf{u}\|_{\ell_\infty} = \max |u_i|, \quad i = 1, \dots, N_{\text{eval}, \Omega}, \tag{51}$$

for a vector \mathbf{u} of length $N_{\text{eval}, \Omega}$. When referring to the errors, we mean both of them.

The following parameters are user specified in the numerical experiments: the length L for the computation domain $B = [-L, L]^2$, the resolution N_u , the number $N_{P,n}$ of Gauss-Legendre panels for each component curve Γ_n and partition radius R . We set the shape parameter $\varepsilon = 2$ for all numerical experiments and set the number of Gauss-Legendre nodes $N_Q = 16$.

4.1 Example 1: Study of weight functions

We now solve the modified Helmholtz equation (4)–(5) for:

$$u(x, y) = \sin(2\pi x) \sin(2\pi y) \exp(-(x^2 + y^2)), \tag{52}$$

to confirm that the parameters N_ϕ and c and the function ψ^k can be set by (35), (37), and (38), as in [9] for the Poisson equation. To reduce the complexity of the problem, assume the corresponding right-hand side to be known in all of \mathbb{R}^2 , not just Ω . To isolate the influence of the choice of weight function ψ (see Table 1), the actual values of f are used as values for the local extensions $f_{i,E}$, instead of the extrapolated ones $A_{i,E} f_{X_i}$. Compact support is still enforced via PUX, where the blending with the zero partitions reduces the regularity of f^e to k .

The computational domain is the unit circle centred at $(17/701, 5/439)$, contained in the box $B = [-L, L]^2$, with $L = 1.5$. The resolution N_u attains values between 40 and 500 and for the evaluation grid use $N_e = 1000$. The partition radius and the number of panels are set such that only the resolution of the uniform grid \mathbf{X} limits the accuracy. In this case, the partition radius is $R = 0.4$ and the number of panels $N_P = 32$. This means that the rate of convergence is only dependent on the regularity of the extension and we can study the influence of choice of Wu-function. Furthermore, we set $\alpha^2 = 10$.

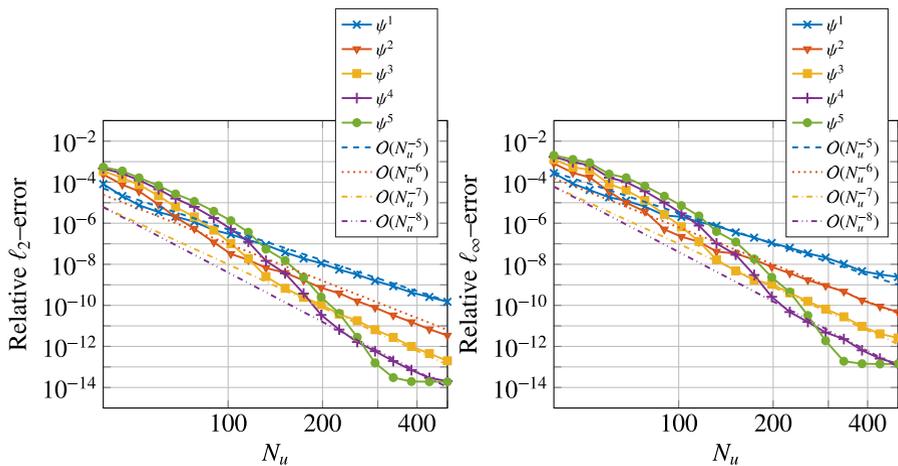


Fig. 6 Error in numerical solution for the modified Helmholtz equation with (52) and $\alpha^2 = 10$, but with local extensions given by analytic expression. Errors are plotted as a function of N_u in loglog-scale for ψ^k , where $k = 1, 2, 3, 4, 5$. See Table 1. Left: Relative ℓ_2 -error. Right: Relative ℓ_∞ -error

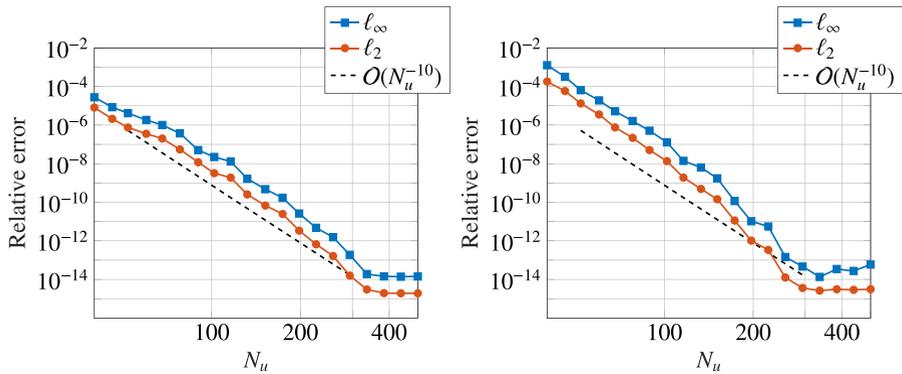


Fig. 7 Error in numerical solution for the modified Helmholtz equation with (52) and $\alpha^2 = 10$, but with local extensions given by analytic expression. The errors are plotted as functions of N_u in loglog-scale with ψ^k chosen according to (36) for each different value of N_u

In Fig. 6, the errors for solving the modified Helmholtz equation are plotted as functions of the number of grid points for different Wu-functions. The behaviour of the errors is as for the Poisson equation in [9]: ψ^k with few continuous derivatives requiring fewer points to be represented than ψ^k with a larger k . Consequently, high regularity can increase the error, since ψ^k is not sufficiently resolved. Compare the errors for using ψ^1 and ψ^5 in Fig. 6 for $N_u \sim 40$. As the grid is refined, the decay is spectral until the error is limited by an algebraic tail. The algebraic tail has a slope of $4 + k$, as expected. The ℓ_∞ -error is about one to two digits less accurate than the ℓ_2 -error, which is consistent for all numerical experiments in this paper. The reason is that there is almost always some target points close to the boundary for which the special quadrature does not give optimal results, e.g. at the intersection of two panels.

We now solve the modified Helmholtz equation in the same numerical setting, but let ψ^k be set automatically by (36). The result is presented in Fig. 7 and the lines follow the corresponding lowest errors in Fig. 6. Thus, (36) indeed chooses ψ^k correctly for a given N_u and we can set the PUX parameters for the modified Helmholtz equation as for the Poisson equation. This holds for α^2 from 10 to 10^5 as well, as is shown in the following numerical experiment. Moreover, the error decreases as that of a tenth-order method.

For the subsequent numerical experiments ψ^k , N_ϕ and c are set by (36), (37), and (38).

4.2 Example 2: The modified Helmholtz equation on a multiply connected domain

We now study the modified Helmholtz equation with a more complex setup for $\alpha^2 = 10^n$, $n = 1, 2, 3, 4, 5$. We take the solution to be:

$$u(x, y) = \cos\left(20\sqrt{x^2 + y^2}\right), \tag{53}$$

on the multiply connected domain shown in Fig. 8 and evaluate the right-hand side in (4) accordingly. The corresponding extension by PUX is shown in Fig. 9, where

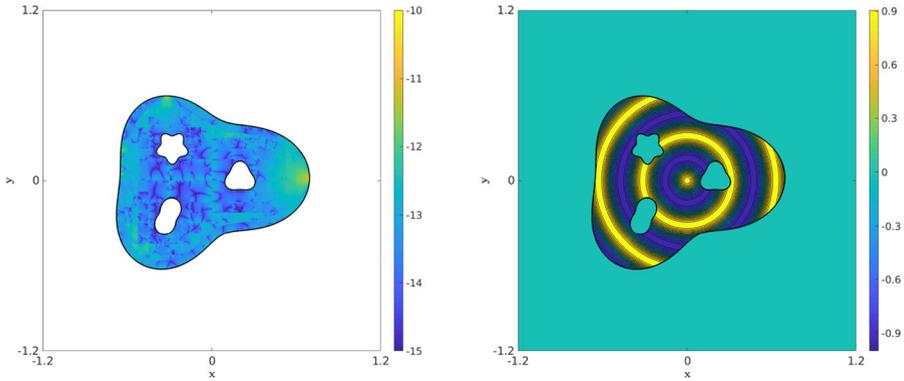


Fig. 8 For $N_u = 1000$ and $\alpha^2 = 10$: the left image shows the pointwise relative ℓ_2 -error for solving the modified Helmholtz equation for (53). The right image shows the solution (53)

$N_u = 1000$ and $k = 5$. The outer boundary is discretised into 80 panels, and the boundaries of the cavities are discretised with 20 panels each. Again, all parameters are set such that only N_u sets the bound for the error. We set $R = 0.23$ and $L = 1.2$. The parameters ψ^k , N_ϕ , and c are set by (36), (37), and (38). The evaluation grid has a resolution of $N_{eval} = 1000$.

The results in Fig. 10 suggest that (36) is a good estimate for setting ψ^k for more complex problems as well. We obtain 10th-order convergence with grid refinement. Note that slightly better results can be achieved; the same parameters are used for the entire range of α and are therefore potentially not optimal. As in the previous example, the relative ℓ_∞ -error is about two orders of magnitude larger than the relative ℓ_2 -error. In Fig. 8, the largest error is by the rightmost point in Ω , at the intersection of two panels. The special purpose quadrature is known to struggle with maintaining full accuracy in such situations.

The modified Helmholtz equation becomes significantly harder to solve for increasing α^2 . This is due to the rapid decay of the kernel (17), which requires a very fine resolution of the boundary to be resolved. We also suffer from cancellation

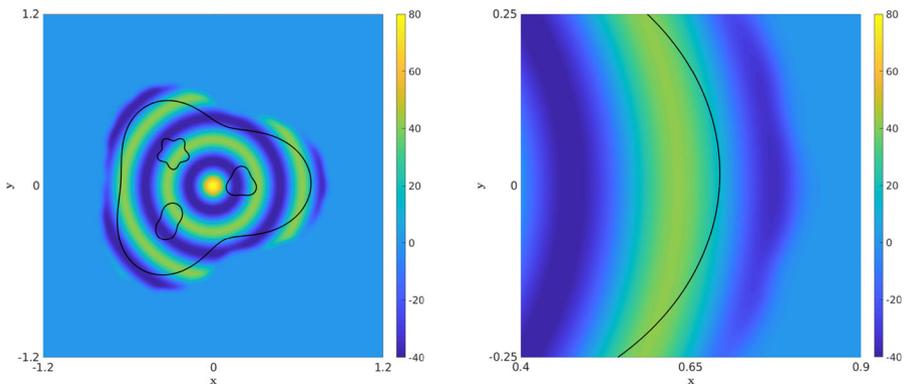


Fig. 9 Left: The right-hand side of the modified Helmholtz equation for (53), extended with PUX. Right: Magnification of the left image. For both $N_u = 1000$ and $\alpha^2 = 10$

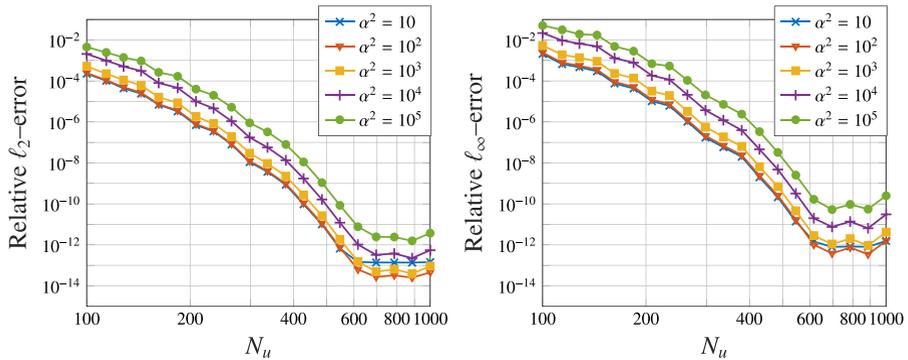


Fig. 10 The errors for example 2 for various resolutions of the uniform grid, over a range of values for α errors due to the scaling of terms with α or α^{-1} . Still, this is not alarming, as a relative ℓ_∞ -error of about 10^{-10} can still be obtained for $\alpha^2 = 10^5$. In terms of the heat equation, this corresponds to a time step of about 10^{-5} .

4.3 Example 3: Adaptive time stepper

We now test the solver for the heat equation (1)–(3) by setting a tolerance for the time stepping error and investigate if it can be maintained for different resolutions N_u . For this purpose, we use the IMEXRK34 scheme with an adaptive time stepper; see Appendix A.1.2 and Appendix A.1.3. The smaller the time step, the harder the modified Helmholtz equation is to solve, as concluded above. Thus, a high-order time stepping scheme, such as IMEXRK34 of fourth order, is a suitable choice since larger time steps can be used. However, other time marching methods can be used as well.

The domain and all parameters are set as for the previous example. The heat equation (1)–(3) is solved with the right-hand side F , initial condition, and Dirichlet boundary data prescribed by the analytical solution:

$$U(t, x, y) = \exp(-t) \sin((x \cos(\pi/4) + y \sin(\pi/4))) + \cos\left(20\sqrt{x^2 + y^2}\right), \quad (54)$$

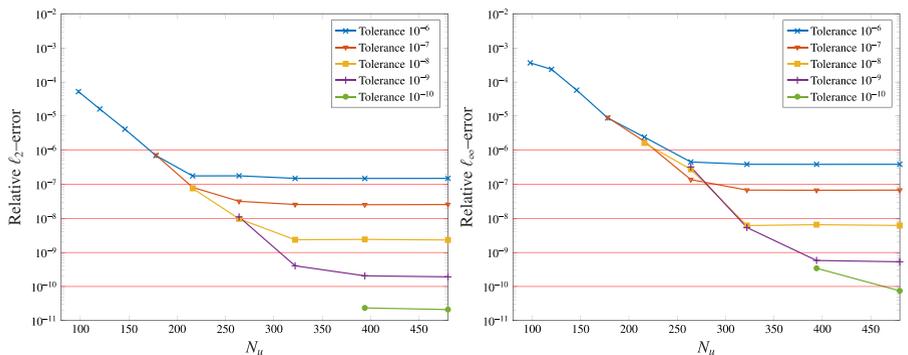


Fig. 11 The errors for example 3 for different resolutions of the uniform grid, at terminal time $t = 1$. The red lines are the set tolerances for the relative ℓ_2 -error, used by adaptive time stepper

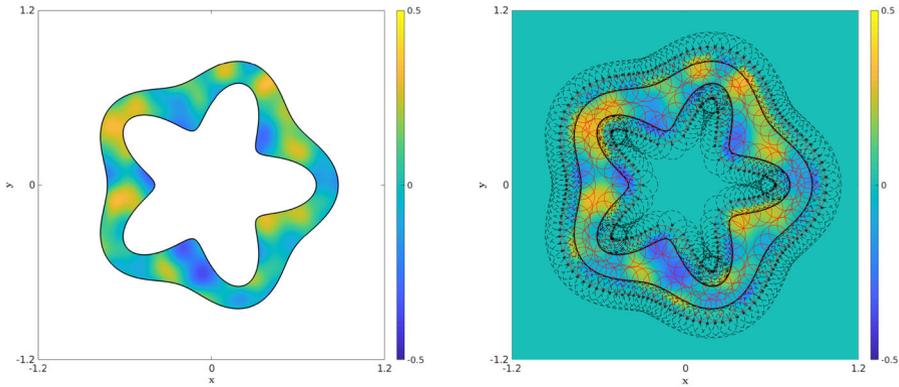


Fig. 12 Left: The initial data U_0 (56). Right: The right-hand side of (55) at t_0 , extended with PUX. Black corresponds to zero partitions and red to interpolation partitions. Note that to increase visibility of the field a different scaling is used than for 13a – f

where the time ranges from 0 to 1. For the evaluation grid, we set $N_{\text{eval}} = N_u$ and measure the error at terminal time $t = 1$.

In Fig. 11, the red lines correspond to set tolerances. It is clear that the adaptive time stepper works as intended, even for tolerances down to 10^{-10} . The relative ℓ_∞ -error is more sensitive to the resolution and exceeds the set tolerance earlier in terms of spatial resolution, roughly with one order in magnitude.

4.4 Example 4: The Allen-Cahn equation, a reaction diffusion problem

The Allen-Cahn equation is stated as:

$$\frac{\partial U(t, \mathbf{x})}{\partial t} - C \Delta U(t, \mathbf{x}) = U(t, \mathbf{x})(1 - U(t, \mathbf{x})^2), \quad t_0 < t, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^2, \quad (55)$$

$$U(t_0, \mathbf{x}) = U_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (56)$$

$$U(t, \mathbf{x}) = e^{-t/2} U_0(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \quad (57)$$

with $C = 10^{-3}$. The right-hand side of (55) is nonlinear and has three stationary points: $U = -1, 0, 1$. For randomised initial data, the solution creates over time patterns with zones attaining these values. The initial data we use is not entirely randomised, since we need smoothness to discuss convergence and accuracy. Instead, we create smooth data by uniformly distributing 50 Gaussians (28) with $\varepsilon = 10$ over the computational domain with $L = 1.2$. Each Gaussian is assigned a coefficient drawn randomly from a uniform distribution over -0.5 to 0.5 . The partition size R is set to 0.1 ; the domain, the extended right hand, and the distribution of partitions are shown in Fig. 12. Each boundary component is discretised with 80 panels.

We create a reference solution by solving the Allen-Cahn equation with tolerance 10^{-6} and $N_u = 800$, from time 0 to 6. The errors are measured on grids with $N_{\text{eval}} = 200, 400$ at terminal time $t = 6$. Snapshots of this solution are shown in Fig. 13a to f. Indeed, the solution forms a pattern of patches with the values $-1, 0,$ and 1 . The results are shown in Table 2. For $N_u = 400$, the relative ℓ_2 -error stays under

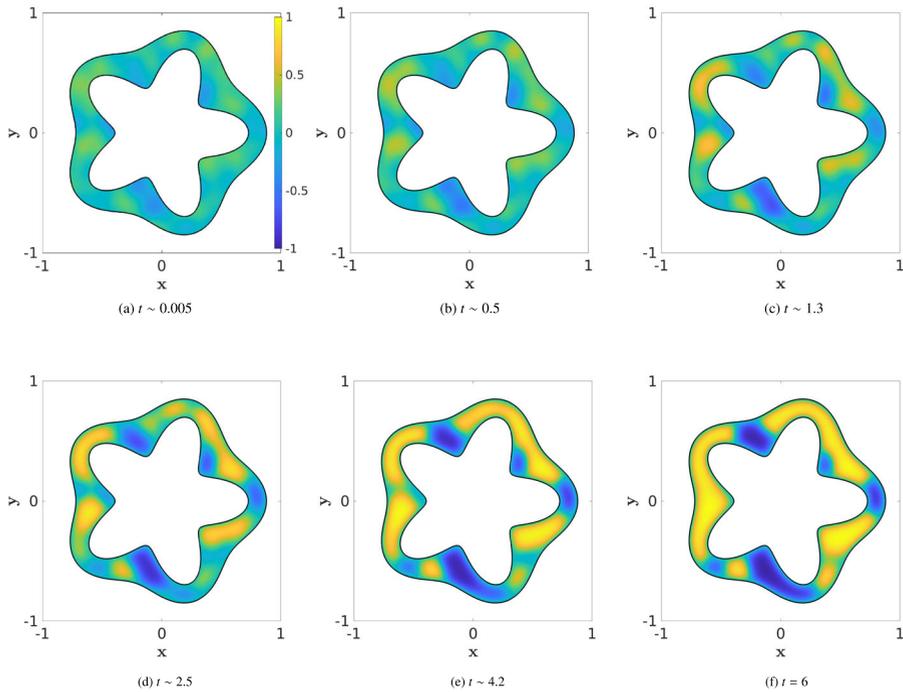


Fig. 13 Numerical solution to (55) with $N_u = 800$ and tolerance 10^{-6} at $t \sim 0.005, 0.5, 1.3, 2.5, 4.2$ and terminal time $t = 6$

the set tolerance. However, unlike example 3, the relative ℓ_∞ -error is always slightly larger than the set tolerance. For $N_u = 200$, only the tolerance 10^{-3} can be obtained. Clearly, this resolution is insufficient to resolve the spatial problem more accurately than that. The error at the terminal time $t = 6$ for $N_u = 400$ with tolerance 10^{-5} is shown in Fig. 14. In this figure, the evolution of the time step is also shown; as the solution advances in time the time step becomes larger. Initially it grows faster, compared with later, as the initial time step was intentionally set small (Fig. 15).

Table 2 Errors at terminal time $t = 6$ for solving (55)–(57) for resolutions $N_u = 200, 400$ and set tolerances for the adaptive time stepper

Tolerance	Relative ℓ_∞ error		Relative ℓ_2 error	
	$N_u = 200$	$N_u = 400$	$N_u = 200$	$N_u = 400$
10^{-3}	3.7314×10^{-3}	3.6123×10^{-3}	7.2035×10^{-4}	7.0625×10^{-4}
10^{-4}	2.6619×10^{-2}	3.6519×10^{-4}	6.6734×10^{-3}	6.8990×10^{-5}
10^{-5}	2.1136×10^{-3}	3.9993×10^{-5}	3.1405×10^{-4}	9.3987×10^{-6}

The reference solution is computed with a tolerance of 10^{-6} with $N_u = 800$. The resolution $N_u = 200$ is insufficient to reach errors below 10^{-3} , while the relative ℓ_2 -error for $N_u = 400$ satisfies tolerances 10^{-3} , 10^{-4} , and 10^{-5}

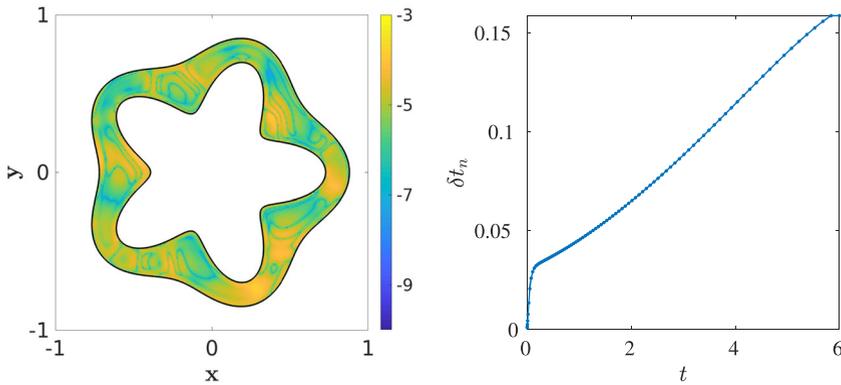


Fig. 14 Left: Pointwise relative error for $N_u = 400$, tolerance 10^{-5} . Right: Evolution of time step δt over time

5 Conclusions

We present a framework built around a panel-based Nyström boundary integral method for solving the forced isotropic heat equation in two dimensions, on multiply connected complex domains. We have addressed several of the issues listed in [1], thereby increasing the class of solvable problems as well as the accuracy in the solutions.

We show how any IMEX method can be applied as a time stepping scheme, and employ an adaptive fourth-order Runge-Kutta scheme in our examples, to accurately solve the heat equation as well as the Allen-Cahn equation, a reaction-diffusion problem with a nonlinear forcing term. Regardless of the specific details of the chosen method, a time step in solving the heat equation is reduced to solving one or a sequence, for a multi-stage method, of modified Helmholtz equations.

As in [4], we formulate the modified Helmholtz equation as a boundary integral problem. Utilising the linearity of the differential operator, the solution is split into a particular and homogeneous problem. Solving the former to high accuracy relies on extending the given right-hand side from the domain it is given on to the entire plane. It is achieved with a partition of unity extension (PUX) that only requires known data at uniform point locations inside the domain. The extension that is computed on a uniform grid in a rectangular domain has compact support and a specified global regularity, making spectral methods very efficient and simple to use. We confirm that the various parameters for PUX, in the context of the modified Helmholtz equation, indeed can be set as for the Poisson equation in [9]. This yields an automated selection for the global regularity to balance different errors, leading to a method which converges with an order 10 in the grid size.

A panel-based Nyström boundary integral method is used to solve the homogeneous problem with modified Dirichlet data, such that the total solution is the sum of the particular and homogeneous solution. Extending to problems with Neumann boundary data is straightforward. The boundary values of the particular solution are computed using a non-uniform FFT. For evaluation of singular and nearly

singular integrals, we have introduced a methodology based on product integration and an explicit kernel split that has given highly accurate results for the Helmholtz [15] and Stokes equations [17]. For large α (small time steps), the method in its original form would however fail completely if an unfeasibly high upsampling of the boundary was not applied. We however realised that this upsampling is only needed very locally, and developed an adaptive approach [18] to achieve a computationally efficient method with high accuracy.

In total, these developments yield a method for very accurately solving the heat equation on complex domains. The highest attainable accuracy in the solution of the modified Helmholtz equation does show a weak dependence on α , but even for the largest values, solutions can typically be attained with at least ten correct digits, meaning that strict time stepping tolerances for the heat equation can be satisfied.

In terms of future developments, it would be useful for some problems to replace the uniform grids and FFT-based method for the particular solution with a volume potential evaluation based on an adaptive FMM. This would however need an integration of the PUX method into the adaptive procedure. In its current form, a uniform grid is required for the precomputation of the interpolation matrix and for the overall efficiency of the PUX algorithm. Another development is to consider the solution of the heat equation and the closely related advection-diffusion equation in time-dependent domains. We have yet to investigate what the limitations are with elliptic marching in this context and will do so in a forthcoming project. Promising results can be found in e.g. [46]. The advection-diffusion equation in time-dependent domains can be used to model the concentration of surfactants in the oil-phase of a micro-system with water drops in oil. These surfactants, or surface active agents, have an exchange with surfactants on the drop surfaces, that alters the surface tension of the drop. Numerical methods for simulating surfactant advection and diffusion on the boundary of drops have been understood and implemented successfully (see [47, 48]). An important extension would be to allow also for surfactants in the oil-phase. One strength of these methods is the accurate treatment of interface conditions, something that is absolutely essential at these small scales where the interface dynamics is of key importance.

Acknowledgements We are grateful for the support from the Natural Science and Engineering Research Council of Canada.

Funding information Open access funding provided by Royal Institute of Technology. We received support of the Swedish Research Council under Grant No. 2015-04998 and funding from the Göran Gustafsson Foundation for Research in Natural Sciences and Medicine.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A. Adaptive time stepping with IMEX Runge-Kutta methods

A.1 Adaptive discretisation in time

This appendix shows how applying implicit-explicit Runge-Kutta (IMEXRK) schemes from [5] to the heat equation reduces it to a sequence of modified Helmholtz equations to solve at each time step. Formulate the heat equation (1)–(3) as:

$$\frac{\partial U(t, \mathbf{x})}{\partial t} = F^I(t, \mathbf{x}, U) + F^E(t, \mathbf{x}, U), \quad \mathbf{x} \in \Omega, \tag{A.1}$$

$$F^I(t, \mathbf{x}, U) = \Delta U(t, \mathbf{x}), \quad F^E(t, \mathbf{x}, U) = F(t, \mathbf{x}), \tag{A.2}$$

where the superscripts denote implicit and explicit, referring to the term being classified as stiff or nonstiff, respectively.

Let t_N denote an instance in time that is the sum of previous discrete time steps $\{\delta t_i\}_{i=1}^{N-1}$ that may be of different size:

$$t_N = \sum_{i=1}^{N-1} \delta t_i + t_0, \quad N \geq 1, \tag{A.3}$$

for some initial time t_0 . Let U_N be the approximation of $U(t_N)$, then at time t_{N+1} it is:

$$U_{N+1} = U_N + \delta t_N \sum_{\sigma \in \{I, E\}} \sum_{j=1}^{N_S} b_j^\sigma k_j^\sigma, \tag{A.4}$$

where N_S is the number of stages for k^σ , $\sigma \in \{I, E\}$, computed as

$$k_j^\sigma = F^\sigma \left(t_N + \delta t_N c_j^\sigma, \bar{U}_j \right), \quad j = 1, \dots, N_S. \tag{A.5}$$

The second argument of F^σ in (A.5) is defined as:

$$\bar{U}_i = U_N + \delta t_N \sum_{\sigma \in \{I, E\}} \sum_{j=1}^i a_{i,j}^\sigma k_j^\sigma = U_N + \delta t_N \sum_{\sigma \in \{I, E\}} \sum_{j=1}^{i-1} a_{i,j}^\sigma k_j^\sigma + \delta t_N a_{i,i}^\sigma k_i^\sigma, \quad i > 1, \tag{A.6}$$

Table 3 Coefficients for an IMEXRK scheme, where $\sigma \in \{I, E\}$, denoting implicit or explicit, applied to the stiff and nonstiff terms, respectively

0	0			
c_2^σ	a_{21}^σ	a_{22}^σ		
\vdots	\vdots		\ddots	
$c_{N_S}^\sigma$	$a_{N_S,1}^\sigma$	\dots	\dots	a_{N_S, N_S}^σ
	b_1^σ	\dots	\dots	$b_{N_S}^\sigma$

In general $a_{ij}^E = 0$ for $i \leq j$ and $a_{ij}^I \neq 0$ for $i = j$, excluding $i = 1$

and $\bar{U}_1 = U_N$. The coefficients $\{a_{i,j}^\sigma\}_{i,j=1}^{N_S}$, $\{b_j^\sigma\}_{j=1}^{N_S}$ and $\{c_j^\sigma\}_{j=1}^{N_S}$ are tabulated in the two associated Butcher tableaus for $\sigma = I$ and $\sigma = E$; see Table 3 for a general IMEXRK scheme. The principal difference between the coefficients for implicit and explicit methods is that $a_{i,j}^E = 0$ for $i \leq j$ while $a_{i,j}^I \neq 0$ for $i = j$, excluding $i = 1$. The quantity \bar{U}_i is unknown for every $i = 2, \dots, N_S$, since the corresponding implicit stage k_i^I is unknown.

The implicit stage at i is $k_i^I = \Delta \bar{U}_i$ by definition (A.2). To avoid approximating the differential operator, replace k_i^I in (A.6) with $\Delta \bar{U}_i$ and reformulate as:

$$\frac{1}{\delta t_N a_{i,i}^I} \bar{U}_i - \Delta \bar{U}_i = \frac{1}{\delta t_N a_{i,i}^I} U_N + \sum_{\sigma \in \{I, E\}} \sum_{j=1}^{i-1} \frac{a_{i,j}^\sigma}{a_{i,i}^I} k_j^\sigma. \tag{A.7}$$

The (A.7) has the form of the modified Helmholtz equation (4)–(5): $f(\mathbf{x})$ corresponds to the right-hand side, $u(\mathbf{x}) = \bar{U}_i(\mathbf{x})$ and $\alpha^2 = (\delta t_N a_{i,i}^I)^{-1}$. We stress that the larger α^2 is the harder (4)–(5) is to solve accurately in terms of numerics (see Section 3.2.1). The associated boundary condition g is (3) evaluated at time $t_N + \delta t_N c_i^I$.

To obtain the next stage k_i^I , equation (A.7) must be solved for \bar{U}_i in Ω . Once \bar{U}_i is known, reformulate (A.7) and compute:

$$k_i^I = F^I(t_N + \delta t_N c_i^I, \bar{U}_i) = \Delta \bar{U}_i = \frac{1}{\delta t_N a_{i,i}^I} \bar{U}_i - \frac{1}{\delta t_N a_{i,i}^I} U_N + \sum_{\sigma \in \{I, E\}} \sum_{j=1}^{i-1} \frac{a_{i,j}^\sigma}{a_{i,i}^I} k_j^\sigma. \tag{A.8}$$

With k_i^I known the stage k_i^E , that is F^E , can be computed explicitly. Note that for (1)–(3) $F^E = F(t, x)$, so the explicit stage k_i^E is independent of the implicit stages; thus, it is computed directly.

To summarise, the approximate solution U^{N+1} at time t^{N+1} is given by (A.4). The stages k_i^I , for $i = 1, \dots, N_S$ are obtained by solving (4)–(5), corresponding to (A.7), and explicit computation of (A.8). Once \bar{U}_i is known, $k_i^E = F^E(t_N + \delta t_N c_i^E, \bar{U}_i)$ is computed explicitly. See the flowchart in Appendix B for a graphical overview.

A.1.1 IMEXRK2

This scheme is never used in this paper, but serves as a simple example of applying an IMEX Runge-Kutta scheme. The stencil for IMEXRK2, with coefficients tabulated in Table 5, involves taking a half time step $\delta t_N/2$ and solving for \bar{U}_2 satisfying:

$$\frac{2}{\delta t_N} \bar{U}_2(\mathbf{x}) - \Delta \bar{U}_2(\mathbf{x}) = \frac{2}{\delta t_N} U^N(\mathbf{x}) + F^E(t_N, \mathbf{x}, \bar{U}_2), \quad \mathbf{x} \in \Omega. \tag{A.9}$$

By (A.4) the solution at the next time step, $t_{N+1} = \delta t_N + t_N$ for every $\mathbf{x} \in \Omega$ is:

$$U_{N+1} = U_N + \delta t_N \left(k_2^I + k_2^E \right) = U_N + \delta t_N \left(\Delta \bar{U}_2 + F^E \left(t_N + \frac{\delta t_N}{2}, \mathbf{x}, \bar{U}_2 \right) \right) \tag{A.10}$$

$$= U_N + \delta t_N \left(\frac{2}{\delta t_N} \bar{U}_2 - \frac{2}{\delta t_{N+1}} U^N - F^E(t_N, \mathbf{x}, \bar{U}_1) + F^E \left(t_N + \frac{\delta t_N}{2}, \mathbf{x}, \bar{U}_2 \right) \right) \tag{A.11}$$

$$= 2U_2^I - U_N + \delta t_N \left(F^E \left(t_N + \frac{\delta t_N}{2}, \mathbf{x}, \bar{U}_2 \right) - F^E(t_N, \mathbf{x}, U_N) \right). \tag{A.12}$$

An important aspect of IMEXRK2 is that we obtain a second-order method by only solving (4)–(5) once, i.e. only one intermediate stage is required.

An adaptive time stepper can be constructed by coupling IMEXRK2 with a method of lower order. A simple IMEX scheme of first order is the Forward-Backward Euler scheme, with coefficients given in Table 4. Applied to the heat equation (1), we have:

$$\frac{U_{N+1}(\mathbf{x})}{\delta t_N} - \Delta U_{N+1}(\mathbf{x}) = G(t_N, \mathbf{x}) + \frac{U_N(\mathbf{x})}{\delta t_N}. \tag{A.13}$$

A.1.2 IMEXRK34

The IMEXRK34 scheme is a coupled third- and fourth-order scheme; see Tables 6 and 7 for the associated Butcher tableaus. It has two sets of six stages $\{k_i^\sigma\}_{i=1}^6$ for $\sigma = I, E$, but only five implicit stages need to be solved for every iterate in time [5]. This is due to k_6^I at t_N is equal to k_1^I at t_{N+1} for $N > 1$, a property sometimes referred to as *first same as last*, or FSAL. Note that the explicit stages do not have this property.

For $N = 0$, the first stage must be given by supplementary initial data ΔU_0 . Otherwise, the procedure is exactly as described in Appendix A.1: for a given i solve (A.7) for \bar{U}_i . Once known extract $k_i^I = \Delta \bar{U}_i$ from (A.7) and compute $k_i^E = F^E(t_N + \delta t_N c_i^E, \bar{U}_i)$ explicitly and start over for $i + 1$ until all six stages are known. An approximate solution U_{N+1} at t_{N+1} is given by (A.4), which is a fourth-order approximation. The third-order approximation \tilde{U}_{N+1} is given by (A.4) as well, but with the coefficients $\{\tilde{b}_j^\sigma\}_{j=1}^{N_S}$ instead of $\{b_j^\sigma\}_{j=1}^{N_S}$ (Table 6).

A.1.3 Adaptivity

Denote the solution given by Forward-Backward Euler or the third-order method in IMEXRK34 as $\tilde{U}(\mathbf{x})$. At each discrete time instance $t_{N+1} = \delta t_N + t_N$ for some δt_N , we compute $U_{N+1}(\mathbf{x})$ and $\tilde{U}_{N+1}(\mathbf{x})$. The relative temporal error is approximated by:

$$r = \frac{\|U_{N+1} - \tilde{U}_{N+1}\|}{\|U_{N+1}\|}, \tag{A.14}$$

where $\|\cdot\|$ is the standard discrete ℓ_2 -norm (50). If r is less than some tolerance TOL, then $U_{N+1}(\mathbf{x})$ is accepted as solution at time t_{N+1} . The quantity r is used to produce a new time step $\delta t_{N,NEW}$ by:

$$\delta t_{N,NEW} = \delta t_{N,OLD} * (0.9 * TOL/r)^{\frac{1}{p+1}}, \tag{A.15}$$

Algorithm 1

Adaptive time stepping

```

1: procedure STEP IN TIME(step size  $\delta t_N$ )
2:   top:
3:   for  $i = 1 : N_S$  do
4:     Solve for  $k_i^I$ , compute  $k_i^E$ 
5:   Compute  $U_{N+1}$  and  $\tilde{U}_{N+1}$ 
6:    $r \leftarrow \frac{\|U - \tilde{U}\|}{\|U\|}$ 
7:    $\delta t_{N,NEW} \leftarrow \delta t_{N,OLD} * (0.9 * TOL/r)^{\frac{1}{p+1}}$ 
8:   if  $r < TOL$  then
9:     Keep solution  $U_{N+1}$ , set  $U_N = U_{N+1}$ ,
10:     $t_N \leftarrow t_N + \delta t_N$ 
11:     $\delta t_N \leftarrow \delta t_{N,NEW}$ 
12:   else
13:     Discard  $U_{N+1}$ 
14:     $\delta t_N \leftarrow \delta t_{N,NEW}$ 
15:   goto top.
    
```

Table 4 Coefficients for the IMEX scheme Forward-Backward Euler

0	0	0	0	0	0
1	0	1	1	1	0
	0	1		1	0

The left and right tables correspond to $\sigma = I$ and to $\sigma = E$, respectively

Table 5 Coefficients for the IMEXRK2 scheme

0	0	0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
	0	1		0	1

The left and right tables correspond to $\sigma = I$ and to $\sigma = E$, respectively

Table 6 The coefficients $\{a_{i,j}^E\}_{i,j=1}^6$, $\{b_j^E\}_{j=1}^6$, $\{\tilde{b}_j^E\}_{j=1}^6$ and $\{c_i^E\}_{i=1}^6$ for IMEXRK34

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0
$\frac{83}{250}$	$\frac{13861}{62500}$	$\frac{6889}{62500}$	0	0	0	0
$\frac{31}{50}$	$-\frac{116923316275}{2393684061468}$	$-\frac{2731218467317}{15368042101831}$	$\frac{9408046702089}{11113171139209}$	0	0	0
$\frac{17}{20}$	$-\frac{451086348788}{2902428689909}$	$-\frac{2682348792572}{7519795681897}$	$\frac{12662868775082}{11960479115383}$	$\frac{3355817975965}{11060851509271}$	0	0
1	$\frac{647845179188}{3216320057751}$	$\frac{73281519250}{8382639484533}$	$\frac{552539513391}{3454668386233}$	$\frac{3354512671639}{8306763924573}$	$\frac{4040}{17871}$	0
b_i^E	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
\tilde{b}_i^E	$\frac{4586570599}{29645900160}$	0	$\frac{178811875}{945068544}$	$\frac{814220225}{1159782912}$	$-\frac{3700637}{11593932}$	$\frac{61727}{225920}$

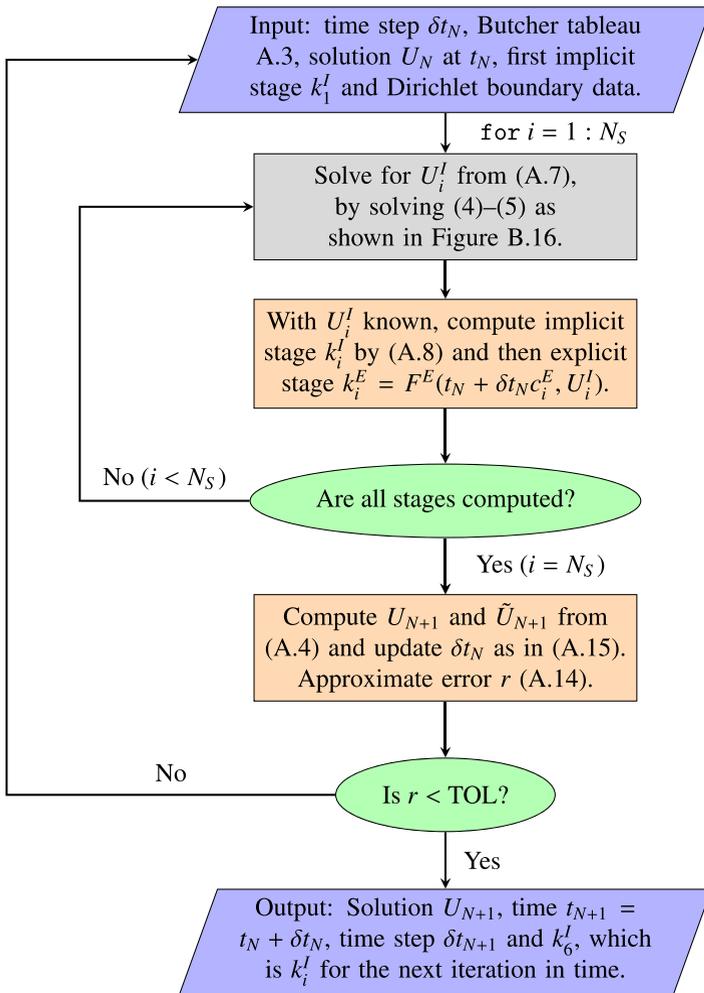
where $p = 2$ from the order of the IMEXRK2 scheme and $p = 4$ for IMEXRK34. The value 0.9 is a safety factor. If the solution is accepted, then $\delta t_{N+1} = \delta t_{N,NEW}$; otherwise, the computations start over at t_N with $\delta t_N = \delta t_{N,NEW}$. Thus, even if the solution is accepted, the step size is updated by the scheme (A.15), meaning growth is possible if appropriate. See the flowchart in Appendix B for a graphical overview.

Table 7 The coefficients $\{a_{i,j}^I\}_{i,j=1}^6$, $\{b_j^I\}_{j=1}^6$, $\{\tilde{b}_j^I\}_{j=1}^6$ and $\{c_j^I\}_{j=1}^6$ for IMEXRK34

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{83}{250}$	$\frac{8611}{62500}$	$-\frac{1743}{31250}$	$\frac{1}{4}$	0	0	0
$\frac{31}{50}$	$\frac{5012029}{34652500}$	$-\frac{654441}{2922500}$	$\frac{174375}{388108}$	$\frac{1}{4}$	0	0
$\frac{17}{20}$	$\frac{15267082809}{155376265600}$	$-\frac{71443401}{120774400}$	$\frac{730878875}{902184768}$	$\frac{2285395}{8070912}$	$\frac{1}{4}$	0
1	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
b_j^I	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
\tilde{b}_j^I	$\frac{4586570599}{29645900160}$	0	$\frac{178811875}{945068544}$	$\frac{814220225}{1159782912}$	$-\frac{3700637}{11593932}$	$\frac{61727}{225920}$

Appendix B. Flowchart over solution procedure

Given U_N , the next solution U_{N+1} is obtained by (A.4). The implicit stages (A.5) must be solved for. Explicit stages are computed directly. The same stages are used to compute low order approximation \tilde{U}_{N+1} , used for adaptive time stepper.



Approximate solution U_{N+1} to the diffusion equation at time t_{N+1} .

Fig. 15 Flowchart over the procedure for updating the approximate solution U_N at t_N for the heat equation (1)–(3). Note that the grey block corresponds to the flowchart in Fig. 16

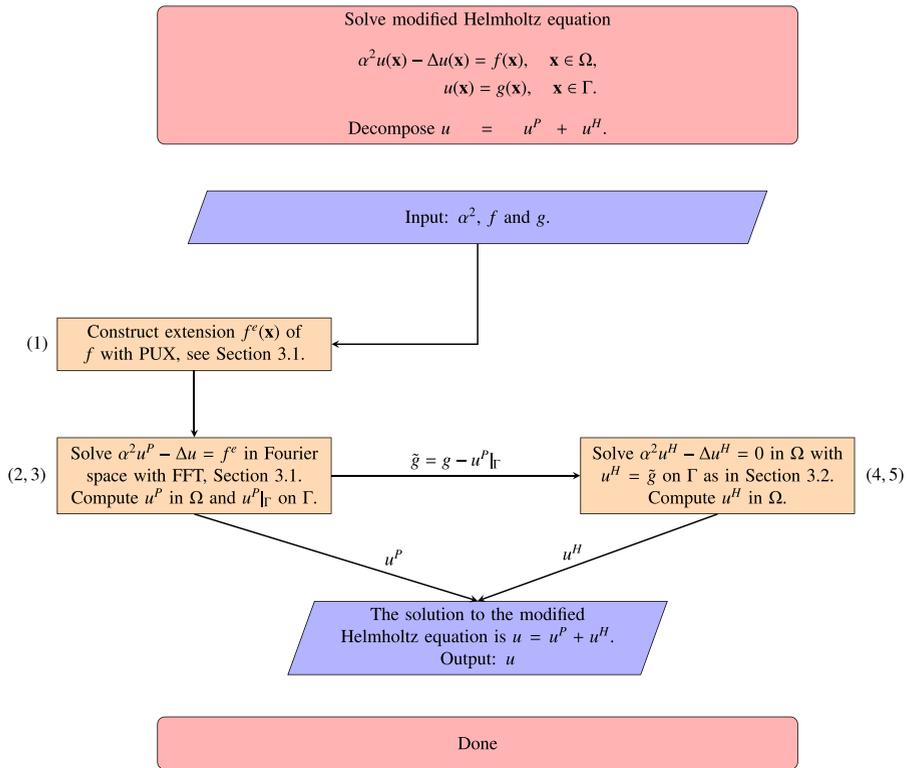


Fig. 16 Flowchart over the procedure for solving the modified Helmholtz equation (4)–(5). The numbers associated with the yellow boxes denote the major computational steps; see Section 3.3

References

1. Kropinski, M.C., Quaife, B.D.: Fast integral equation methods for Rothe’s method applied to the isotropic heat equation. *Comput. Math. Appl.* **61**(9), 2436–2446 (2011)
2. Chapko, R., Kress, R.: Rothe’s method for the heat equation and boundary integral equations. *J. Integral. Equ. Appl.* **9**(1), 47–69 (1997)
3. Chapko, R.: On the combination of Rothe’s method and boundary integral equations for the nonstationary Stokes equation. *J. Integral. Equ. Appl.* **13**(2), 99–116 (2001)
4. Kropinski, M.C., Quaife, B.D.: Fast integral equation methods for the modified Helmholtz equation. *J. Comput. Phys.* **230**(2), 425–434 (2011)
5. Kennedy, C.A., Carpenter, M.H.: Additive Runge–Kutta schemes for convection–diffusion–reaction equations. *Appl. Numer. Math.* **44**(1), 139–181 (2003). pg. 176
6. Dutt, A., Greengard, L., Rokhlin, V.: Spectral deferred correction methods for ordinary differential equations. *BIT* **40**(2), 241–266 (2000)
7. Jia, J., Huang, J.: Krylov deferred correction accelerated method of lines transpose for parabolic problems. *J. Comput. Phys.* **227**(3), 1739–1753 (2008)
8. Minion, M.L.: Semi-implicit spectral deferred correction methods for ordinary differential equations. *Commun. Math. Sci.* **1**(3), 471–500 (2003)
9. Fryklund, F., Lehto, E., Tornberg, A.-K.: Partition of unity extension of functions on complex domains. *J. Comput. Phys.* **375**, 57–79 (2018)
10. Askham, T., Cerfon, A.J.: An adaptive fast multipole accelerated Poisson solver for complex geometries. *J. Comput. Phys.* **344**, 1–22 (2017)

11. Bruno, O.P., Lyon, M.: High-order unconditionally stable FC-AD solvers for general smooth domains I. Basic elements. *J. Comput. Phys.* **229**(6), 2009–2033 (2010)
12. Stein, D.B., Guy, R.D., Thomases, B.: Immersed boundary smooth extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains. *J. Comput. Phys.* **335**, 155–178 (2017)
13. Shirokoff, D., Nave, J.-C.: A sharp-interface active penalty method for the incompressible Navier–Stokes equations. *J. Sci. Comput.* **62**(1), 53–77 (2015)
14. Hao, S., Barnett, A.H., Martinsson, P.G., Young, P.: High-order accurate methods for Nyström discretization of integral equations on smooth curves in the plane. *Adv. Comput. Math.* **40**(1), 245–272 (2014)
15. Helsing, J., Holst, A.: Variants of an explicit kernel-split panel-based Nyström discretization scheme for Helmholtz boundary value problems. *Adv. Comput. Math.* **41**(3), 691–708 (2015)
16. Helsing, J., Ojala, R.: On the evaluation of layer potentials close to their sources. *J. Comput. Phys.* **227**(5), 2899–2921 (2008)
17. Ojala, R., Tornberg, A.-K.: An accurate integral equation method for simulating multi-phase stokes flow. *J. Comput. Phys.* **298**, 145–160 (2015)
18. Klinteberg, L., Fryklund, F., Tornberg, A.-K.: An adaptive kernel-split quadrature method for parameter-dependent layer potentials. [arXiv:1906.07713](https://arxiv.org/abs/1906.07713) (2019)
19. Helsing, J.: Integral equation methods for elliptic problems with boundary conditions of mixed type. *J. Comput. Phys.* **228**(23), 8892–8907 (2009)
20. af Klinteberg, L., Askham, T., Kropinski, M.C.: A fast integral equation method for the two-dimensional navier-stokes equations. *J. Comput. Phys.* **409**, 109353 (2020)
21. Li, J., Greengard, L.: High order accurate methods for the evaluation of layer heat potentials. *SIAM J. Sci. Comput.* **31**(5), 3847–3860 (2009)
22. Wang, S., Jiang, S., Wang, J.: Fast high-order integral equation methods for solving boundary value problems of two dimensional heat equation in complex geometry. *J. Sci. Comput.* **79**(2), 787–808 (2019)
23. Zhou, H.-X., Pang, X.: Electrostatic interactions in protein structure, folding, binding, and condensation. *Chem. Rev.* **118**(4), 1691–1741 (2018). PMID: 29319301
24. Juffer, A.H., Botta, E.F.F., van Keulen, B.A.M., van der Ploeg, A., Berendsen, H.J.C.: The electric potential of a macromolecule in a solvent: A fundamental approach. *J. Comput. Phys.* **97**(1), 144–171 (1991)
25. Chen, K.H., Chen, J.T.: Adaptive dual boundary element method for solving oblique incident wave passing a submerged breakwater. *Comput. Method. Appl. M.* **196**(1), 551–565 (2006)
26. Vorobjev, Y.N.: Modeling of electrostatic effects in macromolecules, pp. 163–202. Springer International Publishing, Cham (2019)
27. Liang, J., Subramaniam, S.: Computation of molecular electrostatics with boundary element methods. *Biophys. J.* **73**(4), 1830–1841 (1997)
28. Kouibia, A., Pasadas, M., Reyah, L., Akhrif, R.: Approximation of surfaces by modified helmholtz splines. *J. Comput. Appl. Math.* **350**, 262–273 (2019)
29. Chen, C.S., Jiang, X., Chen, W., Yao, G.: Fast solution for solving the modified Helmholtz equation with the method of fundamental solutions. *Commun. Comput. Phys.* **17**(3), 867–886 (2015)
30. Li, X.: On solving boundary value problems of modified Helmholtz equations by plane wave functions. *J. Comput. Appl. Math.* **195**(1), 66–82 (2006). Special Issue: The International Symposium on Computing and Information (ISCI2004)
31. Ascher, U., Ruuth, S., Wetton, B.: Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.* **32**(3), 797–823 (1995)
32. Quaife, B.: Fast integral equation methods for the modified helmholtz equation, Ph.D. Thesis, Simon Fraser University (2011)
33. Atkinson, K.E.: The numerical solution of integral equations of the second kind. Cambridge Monographs on Applied and Computational Mathematics (Book 4). Cambridge University Press, Cambridge (1997)
34. Shepard, D.: A two-dimensional interpolation function for irregularly-spaced data, vol 23 (1968)
35. Fasshauer, G.F.: Meshfree Approximation Methods with MATLAB, World Scientific Publishing Co., Inc. River Edge, NJ, USA (2007)
36. Larsson, E., Fornberg, B.: Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Comput. Math. Appl.* **49**(1), 103–130 (2005)

37. Larsson, E., Shcherbakov, V., Heryudono, A.: A least squares radial basis function partition of unity method for solving PDEs. *SIAM J. Sci. Comput.* (2017)
38. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. *SIAM J. Sci. Comput.* **33**(2), 869–892 (2011)
39. Trefethen, L.: *Spectral methods in MATLAB*, Society for Industrial and Applied Mathematics (2000)
40. Carrier, J., Greengard, L., Rokhlin, V.: A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Stat. Comp.* **9**(4), 669–686 (1988)
41. Cheng, H., Huang, J., Leiterman, T.J.: An adaptive fast solver for the modified helmholtz equation in two dimensions. *J. Comput. Phys.* **211**(2), 616–637 (2006)
42. Greengard, L.F., Huang, J.: A new version of the fast multipole method for screened Coulomb interactions in three dimensions. *J. Comput. Phys.* **180**(2), 642–658 (2002)
43. Verchota, G.: Layer potentials and regularity for the Dirichlet problem for Laplace's equation in Lipschitz domains. *J. Funct. Anal.* **59**(3), 572–611 (1984)
44. Helsing, J.: Solving integral equations on piecewise smooth boundaries using the RCIP method: a tutorial, ArXiv e-prints (2012)
45. NIST: Digital Library of Mathematical Functions, Release 1.0.16 of 2017-09-18 <http://dlmf.nist.gov/>
46. Khatri, S., Tornberg, A.-K.: An embedded boundary method for soluble surfactants with interface tracking for two-phase flows. *J. Comput. Physics* **256**, 768–790 (2014)
47. Isson, S.P., Siegel, M., Tornberg, A.-K.: Simulation and validation of surfactant-laden drops in two-dimensional Stokes flow. *J. Comput. Phys.* **386**, 218–247 (2019)
48. Kropinski, M.C.A., Lushi, E.: Efficient numerical methods for multiple surfactant-coated bubbles in a two-dimensional Stokes flow. *J. Comput. Phys.* **230**(12), 4466–4487 (2011)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.