SIAM J. Sci. Comput.
Vol. 37, No. 6, pp. A2630–A2655

CONTOUR INTEGRAL SOLUTION OF ELLIPTIC PDEs IN CYLINDRICAL DOMAINS[*]

NICHOLAS HALE[†] AND J. A. C. WEIDEMAN[†]

Abstract. The solutions of certain elliptic PDEs can be expressed as contour integrals of Dunford type. In this paper efficient contours and quadrature rules for the approximation of such integrals are proposed. The trapezoidal and midpoint rules are used in combination with a conformal mapping that fully exploits the analyticity of the integrand, leading to rapidly converging quadrature formulas of double exponential type. In addition to optimizing the step size of the quadrature formula, implementation aspects such as the solution of the resulting shifted linear systems at each quadrature node are discussed. Numerical examples involving Laplace's equation in a rectangle, a box, and an annular cylinder are presented. Timing and accuracy comparisons of the various implementations are given.

Key words. contour integral, Laplace's equation, Dunford integral, double exponential quadrature, matrix functions

AMS subject classifications. 65D30, 65F60, 65N35

DOI. 10.1137/15M1032764

1. Introduction.

Contour integral methods, in which the operator exponential is approximated by some quadrature scheme, have been studied extensively in the context of time integration of parabolic partial differential equations (PDEs); see [7, 16, 21, 27]. Elliptic PDEs posed in right cylindrical domains can be treated in an analogous manner, as outlined in the work of Gavrilyuk and coauthors [8, 9, 10]. Here a cylindrical domain is defined in the differential geometry sense of the word, namely, a ruled surface spanned by a one-parameter family of parallel lines [5, section 3-5.A]. (This includes rectangular prisms and annular cylinders, two of the test examples we shall consider below.) The space coordinate in the direction of the rulings, here called x, plays a role similar to the time variable in the parabolic problem.

Let such a cylinder be defined by $(x, y) \in [0, 1] \times \Omega$, and let A be an elliptic operator defined on the closed domain $\Omega \in \mathbb{R}^d$. Our interest here is the elliptic PDE in $\mathbb{R}^{d+1}$:

$$\frac{\partial^2 u}{\partial x^2} + Au = 0, \quad u(0, y) = 0, \quad u(1, y) = f(y), \quad y \in \Omega \tag{1.1}$$

(with other types of boundary conditions considered below). For computational work, A will be a matrix, a discrete approximation to the elliptic operator.

The PDE (1.1) is solved, at least formally, by $u(x, y) = E(x; A)f(y)$, where

$$E(x; z) = \big(\sin(\sqrt{z})\big)^{-1} \sin(x\sqrt{z}), \qquad x \in [0, 1], \tag{1.2}$$

[*]Submitted to the journal's Methods and Algorithms for Scientific Computing section July 27, 2015; accepted for publication (in revised form) September 23, 2015; published electronically November 10, 2015.

http://www.siam.org/journals/sisc/37-6/M103276.html

[†]Department of Mathematical Sciences (Applied Mathematics), Stellenbosch University, Stellenbosch 7600, South Africa (nickhale@sun.ac.za, weideman@sun.ac.za). The first author was supported by a Postdoctoral Fellowship from Stellenbosch University. The second author was supported by Incentive Funding from the National Research Foundation in South Africa.

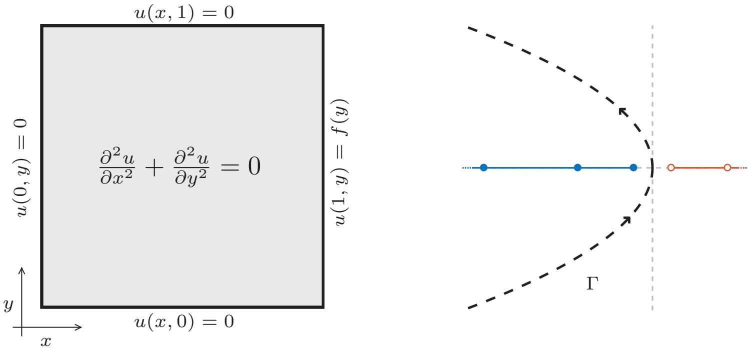

$u(x,1) = 0$
$u(0,y) = 0$
$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$
$u(1,y) = f(y)$
y
$u(x,0) = 0$
x
Γ

FIG. 1. *Left: Domain and boundary conditions for the textbook problem* (1.4)–(1.5). *Right: Example contour, Γ, in the z-plane for the evaluation of the integral* (1.3). *The contour encloses the eigenvalues of A (solid dots), but should keep a safe distance from the poles of $E(x;z)$ (open dots).*

a function analytic in $z \in \mathbb{C} \setminus \{(k\pi)^2, k = 1, 2, \ldots\}$. Evaluating this function with an operator or matrix argument can be done by a Cauchy integral

$$u(x,y) = \frac{1}{2\pi i} \int_\Gamma E(x;z)(zI - A)^{-1} f(y)\, dz, \tag{1.3}$$

sometimes known as the Dunford or Dunford–Taylor formula. (For the operator case, see [6, VII.3.9] or [28, VIII.7], and for the matrix case see [13, section 1.2].) Here Γ is any positively oriented contour in the domain of analyticity of $E(x;z)$ that encloses the spectrum of A. The evaluation of the integral (1.3) on a suitable contour by an efficient quadrature rule forms the basis of the solution procedure proposed in [8, 9, 10].

The simplest example is the case where A is the second derivative operator with homogeneous Dirichlet boundary conditions on $\Omega = [0,1]$, so that (1.1) becomes

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \tag{1.4}$$

subject to

$$u(x,0) = u(x,1) = 0,\ \ 0 < x < 1; \quad u(0,y) = 0,\ \ u(1,y) = f(y),\ \ 0 < y < 1. \tag{1.5}$$

This problem, summarized in Figure 1 (left), is the textbook example of Laplace's equation on a square. Generalizations to higher dimensions ($y \to (y_1, y_2, \ldots, y_d)$) and more general domains Ω are straightforward, however, as is the treatment of nonconstant coefficients (i.e., $A = A(y)$) and more general boundary conditions (such as Neumann or Robin) on the operator A.

It might be instructive to connect the integral (1.3) to better known formulas for the solution of (1.1) by means of the following informal argument. Assume, for example, that the operator A is negative definite with eigenvalues $\lambda_j < 0$ and corresponding eigenfunctions $v_j(y)$. Assume further the expansion

$$f(y) = \sum_j c_j v_j(y), \tag{1.6}$$

and insert this into (1.3), to obtain

$$u(x,y) = \frac{1}{2\pi i} \sum_j c_j \left(\int_\Gamma \frac{\sin(x\sqrt{z})}{\sin(\sqrt{z})} \frac{dz}{z - \lambda_j} \right) v_j(y). \tag{1.7}$$

When the integral in parentheses is evaluated by residues (e.g., on a contour Γ as shown in Figure 1), it follows that

$$u(x,y) = \sum_j c_j \frac{\sin(x\sqrt{\lambda_j})}{\sin(\sqrt{\lambda_j})} v_j(y). \tag{1.8}$$

For example, in the case of the problem (1.4)–(1.5) the eigenvalues are $\lambda_j = -j^2\pi^2$ with corresponding eigenfunctions $v_j(y) = \sin(j\pi y)$. Making use of the identity $\sin i\theta = i\sinh\theta$, one obtains the Fourier series solution to (1.4)–(1.5), namely,

$$u(x,y) = \sum_j c_j \frac{\sinh(j\pi x)}{\sinh(j\pi)} \sin(j\pi y), \tag{1.9}$$

as derived in most textbooks by the method of separation of variables.

Rather than using residue calculus, the approach of [8, 9, 10] and the present paper is to compute instead the integral (1.3) by numerical quadrature. This involves choosing a suitable contour, as well as an efficient integration rule to apply on this contour. A key factor in choosing these is the decay properties of the integrand in (1.3) and in particular the function $E(x;z)$ defined by (1.2). A straightforward calculation yields

$$\left|E(x;re^{i\phi})\right| \sim e^{-(1-x)\sqrt{r}|\sin\frac{1}{2}\phi|}, \qquad r\to+\infty, \quad \phi\neq 0, \tag{1.10}$$

which represents rapid decay away from the positive real axis. This rapid decay makes the simple trapezoidal or midpoint rules natural candidates for the numerical integration [18, 25]. A good contour would aim to maximize the decay, and therefore favor the left half-plane over the right, but would stay away from the spectrum of A where the resolvent norm becomes large.

After choosing the contour and quadrature rule, the approximation to (1.3) can then be expressed as

$$u(x,y) \approx \sum_k w_k(x)\,(z_k(x)I - A)^{-1} f(y). \tag{1.11}$$

Although the weights w_k and nodes z_k will continue to depend on x, for brevity we hereafter drop the x dependence from our notation.

The advantage of (1.11) over series such as (1.8) is that no eigenvalues or eigenfunctions need to be computed, only the resolvent. When A is a matrix, this is equivalent to solving complex shifted linear systems of the form

$$(z_k I - A)\,\boldsymbol{v} = \boldsymbol{f}, \tag{1.12}$$

where the right-hand side is the vector representation of $f(y)$. These systems are independent and can therefore be solved in parallel, which is a major advantage of this class of methods.

Another advantage is the fact that these methods lift the curse of dimensionality by one degree. In particular, (1.1) is a PDE in $\mathbb{R}^{d+1}$ but because $A\in\mathbb{R}^d$ the problems (1.12) are d-dimensional. A two-dimensional (2D) PDE can therefore be reduced to solving only a handful of one-dimensional problems, while a three-dimensional (3D) Laplace equation leads to the solution of complex Helmholtz equations in two dimensions, etc.

On the other hand, in practical computations where A is large, solving the systems (1.12) may still be costly. It is therefore essential that the quadrature rule for computing (1.3) is optimized so that the number of resolvent solves can be kept to a minimum. This issue has received attention in the case of parabolic PDEs [16, 27], but in the elliptic case the only results available are presented in [8, 9, 10]. While this collection of papers provides powerful convergence proofs in a very general setting, implementation aspects are not emphasized. We therefore consider the present paper a practical extension of [8, 9, 10] and summarize the connections as follows.

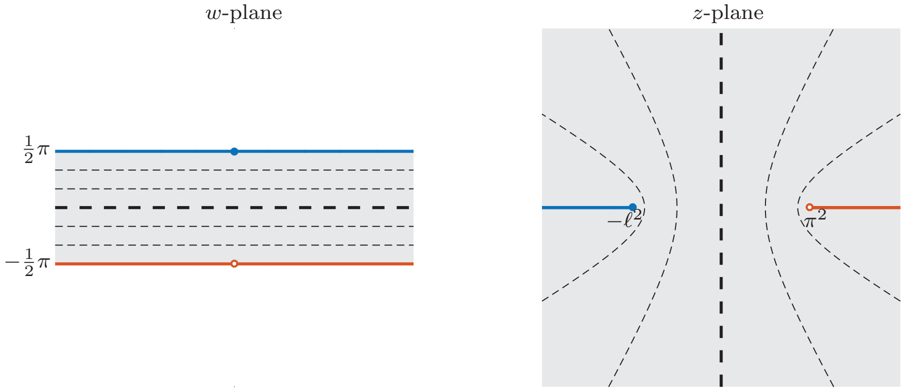
First, we point out that in [8, 9, 10] results were formulated in terms of sinc quadrature rather than the trapezoidal or midpoint rules, but these rules are all equivalent in the present situation. In [8] a parabolic contour was used, and a hyperbolic contour was used in [9, 10]. Here we use a hyperbolic contour as well, which follows naturally from a conformal mapping argument. Its parameters differ from the mapping used in [9, 10], however, in that we exploit here more of the analyticity offered by the function (1.2).

Second, the numerical linear algebra involved in solving the discretized versions of (1.12) received no attention in [8, 9, 10]. Here we focus on spectral collocation for the discretization of A and show that if it has a Kronecker product formulation, the use of specialized Sylvester or Lyapunov solvers can lead to a major speedup in execution time.

The present paper was inspired by a problem solved in [25], namely, the last of the 10 problems in Trefethen's hundred-dollar, hundred-digit challenge [2]. The problem involved a particle at the center of a 10×1 rectangle that undergoes Brownian motion until it hits one of the edges. The challenge was to compute, to 10 significant digits, the probability that it hits one of the shorter rather than one of longer edges. One of several solution methods outlined in [2] involves solving (1.4)–(1.5) with the domain (x, y) rescaled to the rectangle $[0, 1] \times [0, 10]$ and $f(y) = 1$. By symmetry, the required probability is then twice the solution value at the center of the rectangle. By using a Fourier series similar to (1.8) the probability can be computed as $3.837587979 \times 10^{-7}$. Not all the solution methods sketched in [2] carry over from the 2D rectangle to a 3D box, but the formulation as a Laplace PDE survives and hence also the Fourier series solution. For a box of dimension $[0, 1] \times [0, 1] \times [0, 10]$ the probability drops to $7.298817657 \times 10^{-10}$. A solution method based on (1.11) was used in [25] to solve this "particle-in-a-box" problem. We present here an improved version of the method used in [25], and fill in the details omitted in that paper. The solution values just cited will serve as reference values for our numerical tests below, where we shall use the acronyms PIAS, PIAR, and PIAB for particle-in-a-square/rectangle/box, respectively.

Although analytical solutions for all these examples are available in the form of series, it should be stressed that the technique and parameter choices described here are directly applicable to far more general operators A. The only restriction is that the spectrum of A (or that of its discretization) must lie on or near the negative real axis.

The outline of the paper is as follows. In section 2 we select a suitable contour, Γ, and quadrature rule for the approximation of (1.3) with an asymptotically optimal step size, h_*. In section 3 we validate this by comparing the series (1.11) resulting from the quadrature approximation to the classical Fourier series solution (1.9) for the problem (1.4)–(1.5). Section 4 focuses on various techniques for the efficient solution of the resolvent systems (1.12) when the operator A is discretized by spectral collocation in some 2D and 3D problems. In section 5 we briefly discuss some extensions to the



(a) Infinite strip

(b) Double slit-plane

FIG. 2. *Left: Infinite strip of half-width $\pi/2$ about the real axis. Right: The slit-plane $\mathbb{C} \setminus \big\{(-\infty, -\ell^2] \cup [\pi^2, \infty)\big\}$. Horizontal lines in the w-plane (left) are mapped to the hyperbolas in the z-plane (right) by the conformal map* (2.1)*. In particular, the real axis is mapped to the vertical (dashed) line with real part $(\pi^2 - \ell^2)/2$, which defines our contour of integration, Γ. (Different contours are considered in section* 5*.)*

method, including how to deal with the case $x \to 1$ (where the rapid decay in (1.10) is lost) and how to incorporate boundary conditions other than those given in (1.1).

2. Discretization of the contour integral.

The numerical approximation of the integral (1.3) consists of two steps: choose an appropriate contour, then discretize the contour integral by a quadrature rule.

2.1. Choosing a contour.

The simplest choice of contour would be $\Gamma : z = i\theta$, $-\infty < \theta < \infty$, but this fails to take into account (a) the decay and (b) the region of analyticity of the integrand in (1.3), and hence can yield a poor rate of convergence. Instead, following [11, 25], we determine the contour by means of a conformal map from a horizontal strip in the complex plane to the domain of analyticity of the integrand. Such a map is optimal for the trapezoidal and midpoint quadrature rules [17].

The domain of analyticity is determined by the function $E(x; z)$ and the resolvent $(zI - A)^{-1}$. The function $E(x; z)$ has poles at $(k\pi)^2, k = 1, 2, \ldots$, but for simplicity we assume analyticity only in $\mathbb{C} \setminus [\pi^2, \infty)$. The resolvent is analytic at all z not in the spectrum of A, which we assume is confined to the interval $(-\infty, -\ell^2]$ on the negative real axis with $\ell \geq 0$. This assumption includes many cases of interest, including (1.4). In section 5.4 we discuss extensions to the case where the spectrum is located in a wedge-like region around the negative real axis. The plan is therefore to construct a conformal map that takes a horizontal strip in the complex w-plane to the double slit-plane $\mathbb{C} \setminus \big\{(-\infty, -\ell^2] \cup [\pi^2, \infty)\big\}$ (see Figure 2). It is well known that the mapping $z = i \sinh w$ maps the strip $|\mathrm{Im}\, w| < \frac{1}{2}\pi$ conformally to the slit-plane $\mathbb{C} \setminus \big\{(-\infty, -1] \cup [1, \infty)\big\}$, and adjusting for the nonsymmetric location of the slit tips one obtains

$$z = \frac{1}{2}(\pi^2 - \ell^2) + \frac{i}{2}(\pi^2 + \ell^2) \sinh w, \qquad |\mathrm{Im}\, w| < \frac{1}{2}\pi. \tag{2.1}$$

Let $w = \theta + i\alpha$ with $(\theta, \alpha) \in (-\infty, \infty) \times (-\frac{1}{2}\pi, \frac{1}{2}\pi)$. For fixed α, the curve $z = z(\theta)$ is a member of the family of concentric hyperbolas with foci at $z = -\ell^2$

and $z = \pi^2$. For the moment, we shall use the particular member of this family that corresponds to $\alpha = 0$ as the contour of integration in (1.3), because then the strips of analyticity in the upper and lower halves of the w-plane have equal widths; see Figure 2 (left). That is, the contour is defined by

$$\Gamma: \ z = \frac{1}{2}(\pi^2 - \ell^2) + \frac{i}{2}(\pi^2 + \ell^2)\sinh\theta, \qquad -\infty < \theta < \infty. \tag{2.2}$$

Note that this is in fact a parametrization of the vertical line that passes through the midpoint of the interval $[-\ell^2, \pi^2]$; see Figure 2 (right). As discussed in section 5.2, accuracy can sometimes be improved by exploiting the decay in the left half-plane of the factor $E(x; z)$ in (1.3). In this case $\alpha > 0$ is used, and the contour is a hyperbola that begins in the third quadrant, winds around the spectrum, and terminates in the second quadrant, much like the diagram in Figure 1 (right). The analogous situation for parabolic PDEs was considered in [15, 16]. In both the $\alpha = 0$ and $\alpha > 0$ cases the contour is closed at infinity and thus encloses the spectrum in (1.3) as required.

An alternative mapping that could be considered is the one in [11], where the contour is determined via a conformal map not from the infinite strip to the double slit-plane, but from a periodic strip to the region $\mathbb{C}\setminus\{[-L^2, -\ell^2]\cup[\pi^2, \infty)\}$. Although there may be some small benefit from the stronger assumption that the spectrum is contained in $[-L^2, -\ell^2]$, the focus in [11] was on operators/matrix functions like A^α and $\log(A)$ that do not exhibit the rapid decay of (1.10). As we shall see in the next section, mapping from the infinite strip allows one to take advantage of this decay in determining an optimal step size in the quadrature discretization and, as we see in (2.2), results in a much simpler map.

2.2. Trapezoidal rule: Error estimates and step size.

When the change of variable defined by (2.2) is made in (1.3), one obtains

$$u(x, y) = \frac{1}{2\pi i}\int_{-\infty}^{\infty} E(x; z(\theta))\big((z(\theta)I - A)^{-1} f(y)\big) z'(\theta)\, d\theta. \tag{2.3}$$

By inserting (2.2) into (1.10), one sees that the factor $E(x; z(\theta))$ in the integrand exhibits double exponential decay as $\theta \to \pm\infty$, $x \in [0, 1)$ (cf. also (2.19) below). This rapid decay accounts for the efficiency of the trapezoidal/midpoint rule when applied to (2.3). In fact, one of the members in the family of double exponential integration rules is the sinh-transformation [20, section 8.2.3], and it is noteworthy that in the present context this rule appeared naturally from the conformal map (2.1). For discussions of double exponential quadrature we refer to [18, 20, 25].

To complete the discretization of (2.3) by the trapezoidal (resp., midpoint) rule, we introduce an evenly spaced grid with spacing h:

$$\theta_k = kh \qquad (\text{resp., } \theta_k = (k + \tfrac{1}{2})h), \qquad k = 0, \pm 1, \pm 2, \dots. \tag{2.4}$$

The approximation to (2.3) is then

$$u(x, y) \approx \frac{h}{2\pi i}\sum_{k=-\infty}^{\infty} E(x; z(\theta_k))\Big((z(\theta_k)I - A)^{-1} f(y)\Big) z'(\theta_k). \tag{2.5}$$

When the symmetry of the integrand as well as the contour (2.2) is taken into account, followed by a truncation to n terms, the approximation can be computed as

$$u(x, y) \approx \operatorname{Im}\Big\{\sum_{k=0}^{n-1}{}' w_k (z_k I - A)^{-1} f(y)\Big\}, \quad z_k := z(\theta_k), \quad w_k := \frac{h}{\pi} E(x; z(\theta_k)) z'(\theta_k), \tag{2.6}$$

```
 1 function [z, w, h] = nodesandweights(n, x, ell2)
 2 %NODESANDWEIGHTS  Optimal nodes and weights for contour integral method.
 3 % Inputs:   n  - Number of points
 4 %           x  - Height of required slice
 5 %           ell2 - Largest (most positive) eigenvalue of A is at -ell2
 6 % Ouputs:   z  - Quadrature nodes
 7 %           w  - Quadrature weights
 8 %           h  - Optimal step size
 9 % For wapr.m see www.mathworks.com/matlabcentral/fileexchange/3644/
10 if ( nargin < 3 ), ell2 = 0; end                % Default to ell2 = 0
11 h = 2/n*wapr(sqrt(2/(ell2+pi^2))*pi^2*n/(1-x)); % Optimal step size h_*: Eqn (2.21)
12 t = ((0:n-1).'+0.5)*h;                          % Equispaced t_k (midpoint): Eqn (2.4)
13 z = (pi^2-ell2)/2+1i*(pi^2+ell2)/2*sinh(t);     % Mapped nodes z(t_k): Eqn (2.2)
14 dzdt = 1i*(pi^2+ell2)/2*cosh(t);                % z'(t_k)
15 E = sin(x*sqrt(z))./sin(sqrt(z));               % E(x;z(t_k)): Eqn (1.3)
16 indx = imag(sqrt(z)) > asinh(realmax);          % Deal with overflow ...
17 E(indx) = exp(1i*(1-x)*sqrt(z(indx)));          %  when im(z) >> 1.
18 w = (h/pi)*dzdt.*E;                             % Weights: Eqn (2.6)
19 end
```

FIG. 3. *MATLAB code for computing quadrature nodes and weights with the optimal step size, h_*, in* (2.21). *Lines* 16–17 *prevent overflow in the E function when* $\mathrm{Re}(z)$ *is large and negative. This code, along with codes to reproduce all the results in this paper, can be found online at* [12].

where the prime indicates that in the case of the trapezoidal rule (but not the midpoint rule), the $k = 0$ term is halved. MATLAB code for generating the weights and nodes of the midpoint rule version of this formula is given in Figure 3.

To determine a good choice of step size h we follow a line of analysis similar to that presented in [14, 27]. To this end, consider the absolutely convergent integral

$$I = \int_{-\infty}^{\infty} g(\theta)\, d\theta \tag{2.7}$$

and its infinite and finite trapezoidal or midpoint rule approximations

$$I_h = h \sum_{k=-\infty}^{\infty} g(\theta_k), \qquad I_{h;n} = h \sum_{k=-n+1}^{n-1} g(\theta_k) \tag{2.8}$$

(where the upper limit is replaced by $n-2$ in the case of the midpoint rule). In the remainder of this section we derive accurate approximations to the discretization and truncation errors,

$$D = |I - I_h|, \qquad T = |I_h - I_{h;n}|, \tag{2.9}$$

associated with the quadrature approximation to (2.3). As we shall see, the discretization error can be made small by letting $h \to 0$, but that increases the truncation error (assuming a fixed value of n). The optimal value of h follows from an asymptotic balance of these two opposing effects.

THEOREM 2.1 (see [14, 17, 25]). *Let $w = \theta + i\psi$ with θ and ψ real. Suppose $g(w)$ is analytic in the strip $-c_- < \psi < c_+$ for some $c_- > 0$, $c_+ > 0$, with $g(w) \to 0$ uniformly as $|w| \to \infty$ in that strip. Suppose further that the function $g(w)$ satisfies*

$$\int_{-\infty}^{\infty} |g(\theta + ic_+)|\, d\theta \le M_+(c_+), \qquad \int_{-\infty}^{\infty} |g(\theta - ic_-)|\, d\theta \le M_-(c_-), \tag{2.10}$$

then

$$|I - I_h| = D_+ + D_-, \tag{2.11}$$

where

$$D_+ \leq \frac{M_+(c_+)}{e^{2\pi c_+/h}-1}, \qquad D_- \leq \frac{M_-(c_-)}{e^{-2\pi c_-/h}-1}. \tag{2.12}$$

For the integral (2.3) we require

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} \left|E(x,z_\pm)z_\pm'\right| \|(z_\pm I - A)^{-1}\boldsymbol{f}\| \, d\theta \leq M_\pm(c_\pm), \tag{2.13}$$

where

$$z_\pm = z(\theta \pm ic_\pm), \quad z_\pm' = z'(\theta \pm ic_\pm). \tag{2.14}$$

The conformal map (2.1) involves a strip of analyticity of half-width $\frac{1}{2}\pi$ in the w-plane, and therefore it follows that $c_\pm < \frac{1}{2}\pi$ with $c_\pm \to \frac{1}{2}\pi$ maximizing the decay rate of the exponential factors in (2.12). At the same time, however, $M_+(c_+)$ grows unboundedly as $c_+ \to \frac{1}{2}\pi$, where the shift z_+ in the resolvent approaches the negative real axis and hence the spectrum of A. Similarly, $M_-(c_-)$ is unbounded as $c_- \to \frac{1}{2}\pi$, as z_- then approaches the positive real axis and the poles of $E(x;z)$. The remedy is to take $c_\pm = \frac{1}{2}\pi - \varepsilon_\pm$ for some $0 < \varepsilon_\pm \ll 1$ and find the values of ε_+ and ε_- that minimize the right-hand side of

$$D_\pm \leq \min_{0<\varepsilon_\pm<\frac{1}{2}\pi} \frac{M_\pm(\frac{1}{2}\pi - \varepsilon_\pm)}{e^{2\pi(\pi/2-\varepsilon_\pm)/h}-1}. \tag{2.15}$$

Rather than attempting to characterize rigorously the behavior of $M_\pm(\frac{1}{2}\pi - \varepsilon_\pm)$ as $\varepsilon_\pm \to 0$ we offer the following heuristic argument that the dependence on $\varepsilon_\pm$ is at worst algebraic and can in practice be ignored. For M_- the algebraic behavior is a consequence of the fact that $E(x;z)$ has only simple poles. For M_+ and diagonalizable matrix A we have [24]

$$\|(z_+ I - A)^{-1}\|_2 \leq \frac{\kappa(V)}{\operatorname{dist}(z_+, \sigma(A))}, \tag{2.16}$$

where $\kappa(V)$ is the condition number of the matrix of eigenvectors of A. This shows that the resolvent norm also behaves like a simple pole as z_+ approaches the spectrum of A . Furthermore, the numerical experiments in Figure 4 suggest that in practice the growth is only logarithmic. In either case, choosing the values of $\varepsilon_\pm$ that minimize (2.15) leads to $D_\pm \ = \mathscr{O}(h^{-1}e^{-\pi^2/h})$ or $D_\pm \ = \mathscr{O}(\log(h^{-1})e^{-\pi^2/h})$, respectively. When h is small, the exponential factor $e^{-\pi^2/h}$ dominates both h^{-1} and $\log(h^{-1})$ and so we ignore them. Thus, we assume the two discretization errors are comparable and combine them into one, namely,

$$D = \mathscr{O}(e^{-\pi^2/h}). \tag{2.17}$$

The exception is when ℓ is large. Here the fast decay in $E(x;z)$ as z approaches the negative real axis leads to $M_+ = \mathscr{O}(e^{-(1-x)\ell}\log(\varepsilon_+^{-1}))$, while M_- remains $\mathscr{O}(\log(\varepsilon_-^{-1}))$. The two discretization errors can then be significantly different, which justifies shifting the contour of integration off the real axis in the θ-plane to balance these terms. We explore this in section 5.2.

If the series (2.5) could be summed to infinitely many terms, the convergence rate (2.17) would be very attractive. In practice, however, the infinite series is truncated

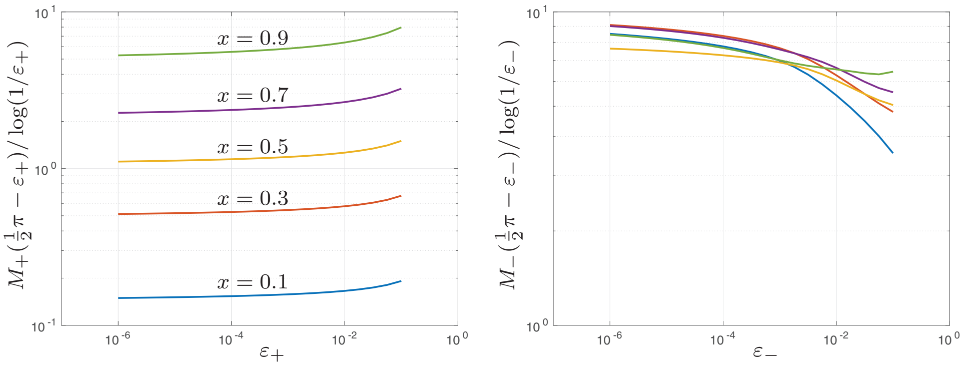


FIG. 4. *Numerical verification of the behavior of $M_+(c_+)$ (left) and $M_-(c_-)$ (right) as $c_\pm \to \frac{1}{2}\pi$ for the problem* (1.4)–(1.5) *with $f(y) = 1$ and various choices of x. The left figure shows that the growth is logarithmic in ε_+ and roughly linear in x (at least, away from the limits $x \to 0$ and $x \to 1$). The figure on the right, which shows curves for the same values of x, suggests that the growth is slightly faster than logarithmic in ε_- and essentially independent of x. This logarithmic growth is negligible compared to the exponential decay in the denominators of* (2.12).

as in (2.6). We estimate the truncation error as the magnitude of the first omitted term in (2.6), which is justified by the double exponential decay of the weights w_k in the series. Hence we shall estimate the truncation error as $T = \mathscr{O}(E(x; z(\theta_n))$.

From (2.2) we deduce that if h is fixed, then

$$z(\theta_n) \sim \frac{i}{2}(\pi^2 + \ell^2)\sinh\theta_n \sim \frac{i}{4}(\pi^2 + \ell^2)\,e^{\theta_n}, \qquad n \to \infty. \tag{2.18}$$

Accordingly, we substitute $r = \frac{1}{4}(\pi^2 + \ell^2)\,e^{\theta_n}$ and $\phi = \frac{1}{2}\pi$ into (1.10) and use the fact that $\theta_n = nh$ (resp., $\theta_n \sim nh$) for the trapezoidal (resp., midpoint) rule. This yields an estimate for the truncation error

$$T = \mathscr{O}(e^{-(1-x)\sqrt{\pi^2+\ell^2}\,e^{nh/2}/(2\sqrt{2})}), \tag{2.19}$$

valid when x is bounded away from 1.

An estimate for the optimal step size h can be derived by matching the exponents in the discretization and truncation error estimates, namely, (2.17) and (2.19). That is,

$$\frac{\pi^2}{h} = \frac{(1-x)}{2\sqrt{2}}\sqrt{\pi^2 + \ell^2}\,e^{hn/2}, \tag{2.20}$$

the solution of which can be expressed in terms of the Lambert W-function[1]

$$h_* = \frac{2}{n}W\left(\frac{\sqrt{2}\,\pi^2\,n}{(1-x)\sqrt{\pi^2+\ell^2}}\right). \tag{2.21}$$

The corresponding estimate for the total error is

$$T + D = \mathscr{O}\big(e^{-\pi^2/h_*}\big), \qquad n \to \infty. \tag{2.22}$$

[1] Most computer algebra systems make provision for the computation of $w = W(t)$, which is defined implicitly by $we^w = t$. For the computations here, for example, in the code in Figure 3, we used the MATLAB function `wapr` available from [1].

Using the fact that the principal branch of the Lambert function obeys $W(t) \sim \log t$, $t \to \infty$, one concludes that the optimal step size and total error satisfy, respectively,

$$h \sim c_1 \frac{\log n}{n}, \qquad T + D = \mathscr{O}\big(e^{-c_2 n/\log n}\big), \qquad n \to \infty, \tag{2.23}$$

for constants c_1 and c_2. The latter formula is consistent with the typical convergence rate for double exponential quadrature [18, 25]. The estimates (2.23) agree with the formulas of [9, 10] but are not very useful in practice. We have found that the formula (2.21), for example, gives a much better estimate for h when n is only moderately large. A similar observation was made in [25].

In the above analysis we assumed $x \in [0, 1)$ but with x bounded away from 1, and we have also assumed that $\ell = \mathscr{O}(1)$. We look at the modifications necessary when $x \to 1$ and $\ell \gg 1$ in sections 5.1 and 5.2, respectively.

3. The contour integral series versus the classical series.

We start with a numerical check on the validity of the optimal step size formula (2.21). For this purpose we consider the textbook Dirichlet problem on the square, namely, (1.4)–(1.5) with $f(y) = 1$ on $(0, 1)$. The explicit series solution, which is of the form (1.9), will be used to compute the error.

To enable us to focus purely on the accuracy of the discretization of the contour integral as described in section 2, we shall not use a discrete version of the operator $A = \partial^2/\partial^2 y$ but rather use exact formulas. This particular problem is simple enough for the resolvent equation $v(z; y) = (zI - A)^{-1} f(y)$ to be solved analytically, which in the case $f(y) = 1$ is

$$-\frac{d^2 v}{dy^2} + zv = 1, \qquad v(0) = v(1) = 0, \tag{3.1}$$

with solution

$$v(z; y) = \frac{1}{z}\Big(1 - \frac{\cosh((\frac{1}{2} - y)\sqrt{z})}{\cosh(\frac{1}{2}\sqrt{z})}\Big). \tag{3.2}$$

The series (2.6) resulting from the contour integral approximation becomes

$$u(x, y) \approx \operatorname{Im}\Big\{ {\sum_{k=0}^{n-1}}' w_k v(z_k; y) \Big\}. \tag{3.3}$$

In our first test we establish the validity of the step size formula (2.21). In this simple problem the spectrum of A can be computed explicitly (the eigenvalues are given below (1.8)) and hence we take $\ell = \pi$.

In the left panel of Figure 5, we show the numerically computed errors in (3.3) as contour plots in the (n, h) parameter domain. In the right panel, we do the same for the total error estimate $D+T$, with the discretization and truncation errors defined by (2.17) and (2.19), respectively. In each of the two panels, we have superimposed the theoretical estimate of the step size (2.21) as a dashed curve. The striking agreement between the two contour plots and the fact that the dashed curve follows the minimum error path in the (n, h) plane lends support to our error analysis. We note that in each plot two distinct regions can be identified. Above the dashed curve the discretization error D dominates, and below it the truncation error T dominates. The contour

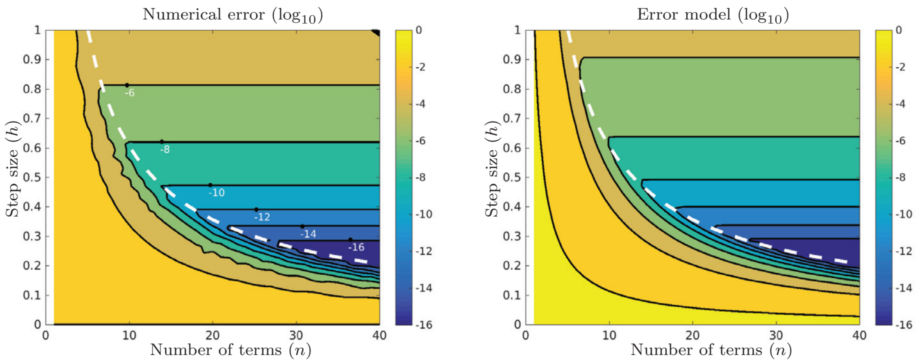


FIG. 5. *Contour plots of the error in the series approximation* (3.3) *to the Dirichlet problem* (1.4)–(1.5) *with* $f(y) = 1$*, as a function of the number of terms,* n*, and the step size,* h*. The labels indicate the* $\log_{10}$ *value of the maximum relative error computed at* $(x, y) = (0.5, \mathbf{y})$*, where* $\mathbf{y}$ *are a* 100 *equally spaced points in* $(0, 1)$*. On the left, numerically computed errors are shown. On the right, the theoretical error as defined by the sum of the terms in* (2.17) *and* (2.19)*. Experiment and theory are shown to be in excellent agreement. The dashed curve represents the optimal step size* h_* *from* (2.21) *as a function of* n*.*

curves in these two regions are consistent with the theoretical error formulas (2.17) and (2.19), respectively, namely, $h =$ constant (horizontal lines) and $hn =$ constant (hyperbolas).

We have not seen the quadrature series (3.3) in the literature. It can be considered an alternative to the classical series solution (1.9), which by contrast can be found in almost any undergraduate textbook on engineering mathematics. It is well known that the latter series converges slowly when $x \approx 1$, and in these cases the new series can in fact yield higher accuracy for the same number of terms. This is demonstrated in Figure 6, where we show the absolute error (the max-norm on $y \in (0, 1)$, approximated using 100 evenly spaced values) in each of the two series as a function of the number of terms. For small and intermediate values of x (left panel) the classic series is more accurate, while for x close to 1 (right panel) the contour integral series is more accurate, with the break-even point around $x \approx 0.7$ (middle panel). We hasten to add that technically the classical series will always converge faster as $n \to \infty$, as it converges exponentially while the contour integral series converges only subexponentially (recall (2.23)). In a finite precision (i.e., finite n) computation, however, the contour integral series can converge faster in a practical sense.

It is further possible to compare the accuracy of the classical and the contour integral series by estimating the remainder of each when the series is truncated to n terms. Using (1.8) and (2.21)–(2.22), one can show that the contour integral is superior for values of x in the range $(x_c, 1)$, with $x_c \sim 1 - \pi/(2 \log(2n))$, $n \gg 1$. For example, with $n = 32$ the value of x_c is approximately 0.71, which is consistent with the break-even case shown in the middle panel of Figure 6.

Returning to the striking correspondence between theory and computation shown in Figure 5, it is only a single numerical experiment and one has to question how it will hold up in other situations.

First, we have chosen only a single value of x in the center of the domain, and measuring the error at other values of $x \in (0, 1)$ will certainly change the quality of approximation (as we saw in Figure 6, the errors increase with increasing x). However, if one repeats Figure 5 for other values of x the degree of correspondence between the empirical and theoretical contour plots (left and right panels) remains excellent

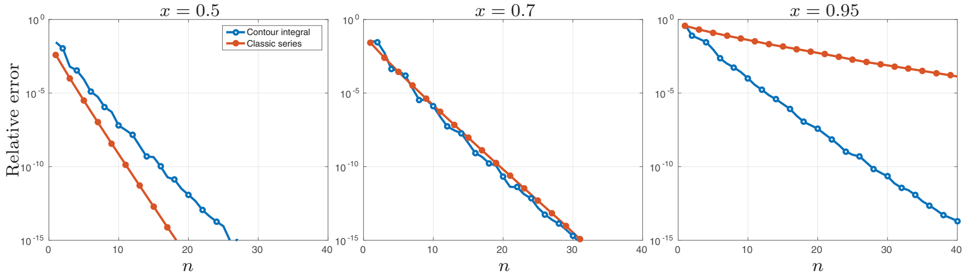


Fig. 6. *Convergence of the classical series* (1.8) *and the contour integral series* (3.3) *for the problem* (1.4)–(1.5) *with* $f(y) = 1$*. For large* x *the contour integral series is better. In particular, to reach* 10 *digits of accuracy when* $x = 0.95$*, fewer than* 30 *terms are required in* (3.3)*, while the series* (1.8) *requires over* 120 *terms. The limit* $x \to 1$ *is discussed in further detail in section* 5.1.

(figures are omitted for brevity). The exception is when $x \approx 1$, because then the double exponential decay in (2.3) breaks down as mentioned in the last paragraph of section 2. This situation will be looked at separately in section 5.1.

Second, we have used spectral information on A to obtain a value for ℓ in (2.21), but what if no such information is available or is hard to come by? In the case of the square domain $[0, 1] \times [0, 1]$ it makes no great difference, at least when $A = \partial^2/\partial y^2$, if one simply sets $\ell = 0$. In fact, by doing so one overestimates the optimal step size only slightly, which is tantamount to erring on the safe side, as can be seen in Figure 5.

If the spectrum of A is shifted further along the negative real axis, however, having a good estimate for ℓ^2 can improve the accuracy by some orders of magnitude. To demonstrate this, consider changing the square domain to the rectangle $(x, y) \in [0, 1] \times [0, b]$. In the case $b = 0.1$ and $f(y) = 1$, the problem (1.4)–(1.5) is then identical to the 10-digit problem discussed in the introduction. This modification changes the exact value of ℓ^2 from π^2 to π^2/b^2. We continue to use the series approximation (3.3), with the obvious modifications to (3.2). If ℓ is large, then further advantage can be obtained by moving the contour of integration off the real axis in Figure 2(a). We explore this in section 5.2.

In Figure 7 we plot error contours for the case $b = 0.1$. In contrast to Figure 5, where we plotted errors as a function of (n, h), we plot errors here as a function of (n, ℓ^2). The value of h used at any point in the (n, ℓ^2) plane is the optimal step size formula (2.21). The dashed curve represents the theoretical value of ℓ^2, namely, $(\pi/b)^2$. Below this curve the error is just the total error, defined by (2.22). Again there is a reasonable agreement between theory and experiment. In this part of the error plots we can see the effect of using an inaccurate estimate of ℓ^2. Compare, for example, the error levels along the dashed curve $\ell^2 = (\pi/b)^2$, which represent exact spectral information, with the error level along the line $\ell^2 = 0$, which would be a reasonable estimate if no spectral information were available. In this example one can see a reduction of about half the number of terms in the series if exact spectral information is used.

Above the dashed curve in Figure 7 we see the effect of overestimating ℓ^2. To understand what is happening it is informative to return to Figure 2. Suppose the right-hand figure of Figure 2 depicts our overestimated value of ℓ^2 and that the true smallest eigenvalue of A lies on, say, the dashed hyperbola to the left of the vertical line at $(\pi^2 - \ell^2)/2$. The preimage of this point in the w-plane (Figure 2, (left)) under

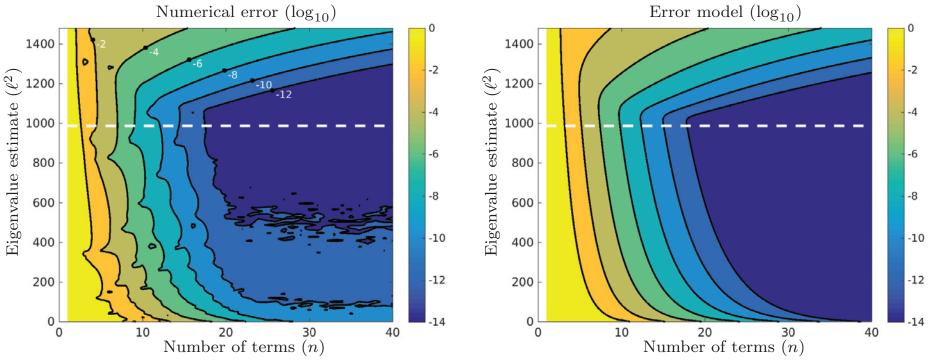


FIG. 7. *Contour plots of the error in the series approximation* (3.3) *to the Dirichlet problem* (1.4)–(1.5) *with* $f(y) = 1$*, but defined on the rectangular domain* $[0, 1] \times [0, 0.1]$ *as a function of the number of terms,* n*, and the approximation to the dominant eigenvalue,* $-\ell^2$*. The labels indicate the* $\log_{10}$ *value of the maximum relative error computed at* $(x, y) = (0.5, \boldsymbol{y})$*, where* $\boldsymbol{y}$ *represents* 100 *equally spaced points in* $[0, 1]$*. The left figure shows the numerical error, whereas the right shows the level curves of the model error* $D + T$*. Again, experiment and theory are shown to be in excellent agreement, except for the lower right of the domain, where roundoff errors contaminate the results.*

the conformal map (2.1) then lies on the first dashed line above the imaginary axis. This reduces the width of analyticity in the strip and so, by Theorem 2.1, the rate of convergence is reduced.

We conclude then from Figure 7 that having a good approximation to ℓ^2 can significantly improve the convergence rate of the contour integral method and that it is better to underestimate the value than to overestimate it.

4. Discretizing in space and numerical linear algebra.

In the numerical results presented above we used an exact eigenfunction representation of the operator to focus specific attention on the quadrature error. In practice such a representation is typically not available and some discrete approximation to the operator will be required. Once the operator is discretized, the computation of the resolvent in the sum (2.6) becomes equivalent to the solution of n shifted linear systems of the form (1.12). The efficient solution of such systems is therefore a key factor in the success of the contour integral method, and we discuss this here along with the question of the discretization of A.

4.1. Example 1: Laplace equation in a square.

Although one could use the methods of finite differences or finite elements, we shall use Chebyshev spectral collocation for the discretization here. That is, a problem such as (1.4)–(1.5) is discretized by

$$\frac{\partial^2 \boldsymbol{u}}{\partial x^2} + A\boldsymbol{u} = 0, \quad \boldsymbol{u}(0) = 0, \quad \boldsymbol{u}(1) = \boldsymbol{f}, \tag{4.1}$$

where the y-variable has been discretized on an $(m+1)$-point Chebyshev grid of the second kind, scaled to $[0, 1]$. The vectors $\boldsymbol{u} = \boldsymbol{u}(x)$ and $\boldsymbol{f}$ have entries

$$\boldsymbol{u}_j(x) \approx u(x, y_j), \quad \boldsymbol{f}_j = f(y_j), \quad \text{where} \quad y_j = \sin^2\left(\frac{\pi j}{2(m+1)}\right), \quad j = 1, \ldots, m. \tag{4.2}$$

In the case of (1.4)–(1.5), A is the second derivative spectral collocation matrix of order $m \times m$ with homogeneous Dirichlet conditions enforced by basis recombination [23, 26].

The resolvent systems to be solved are

$$(4.3) \qquad \big(z_k I - A\big)\boldsymbol{v}_k = \boldsymbol{f}, \quad k = 0, \ldots, n-1.$$

Although the matrices A are dense, they are typically of moderate size. For example, to reproduce the error curves in Figure 6 differentiation matrices of sizes 32×32, 38×38, and 83×83 are sufficient, respectively, for values of $x = 0.5$, 0.7, and 0.95. The increasing number of degrees of freedom is required to resolve the corner singularity at $(x, y) = (1, \pm 1)$. One of the particular advantages of the contour integral method is precisely the fact that accurate solutions can be obtained in the interior of the domain (i.e., far away from the corners) without having to resolve these kinds of singularities.

The more challenging case where A arises from discretization on a two- or higher-dimensional domain will be discussed below, but for this one-dimensional problem the systems (4.3) can simply be solved directly and in parallel, for example, with the MATLAB `backslash` and `parfor` functions. When A is large, improved efficiency can be obtained by computing an a priori Hessenberg factorization, $A = PHP^T$, or Schur factorization, $A = USU^*$. (Note that an LU factorization is not enough.) The coefficient matrix $z_k I - A$ in (4.3) can thus be transformed into $z_k I - H$ or $z_k I - S$, which can then be solved efficiently, and likewise in parallel. Alternatively, one could use the ultraspherical formulation of Chebyshev spectral methods [19], which in the current setting effectively reduces to the factorization $A = 4\mathscr{D}S^{-1}D_2\mathscr{D}^{-1}$, where $S = S_0 S_1$, and D_2 are given in [19, section 3], and $\mathscr{D}$ is a diagonally scaled discrete cosine transform. In this case the resolvents $z_k I - A$ are transformed to $z_k S - D_2$ (plus boundary conditions) which, due to the banded structure of D and S, can be solved in linear time.

Figure 8 shows the time taken to solve (4.1) using each of these different factorizations as the matrix size, m, is increased using $n = 1, 20$, and 100 nodes, respectively. The $n = 1$ case corresponds to a single linear solve, where it is no surprise that forming a Hessenberg or Schur factorization is more expensive. When $n = 20$—a more typical case—we see that although the asymptotic complexity remains the same, the Hessenberg factorizations can significantly reduce the computation time for all m. The Schur factorization is typically more expensive to compute than the Hessenberg (in fact a Hessenberg reduction is usually the first step in a Schur factorization) but the upper triangular matrix S will allow each linear solve to be performed faster. The break-even value for these two competing factors is around $n \approx 100$. For large m (in this case around 50 or so) the ultraspherical discretization eventually wins because of its linear complexity. However, for nonconstant coefficient problems the bandwidth of the matrices in the ultraspherical discretization will increase, and this break-even point will become larger.

It is worth noting that the discussion above is closely related to our choice of a spectral discretisation. Had we chosen to use finite differences or finite elements, then the matrices in question would instead be large and sparse, and typically one would employ iterative (rather than direct) methods to solve (4.3) [4, 14].

A complication arises when the solution is required at many values of x. Since the optimal step size, and hence the nodes $z_k = z(\theta_k)$ in (4.3), depend on x through (2.21), each value of x requires the solution of a separate set of resolvent systems. The factorizations above can alleviate this problem somewhat, but an alternative strategy is to ignore the fact that the optimal step size depends on x and simply fix h via (2.21) for a specific value of x in $(0, 1)$, say, x_*. Then the equations of (4.3) are solved only

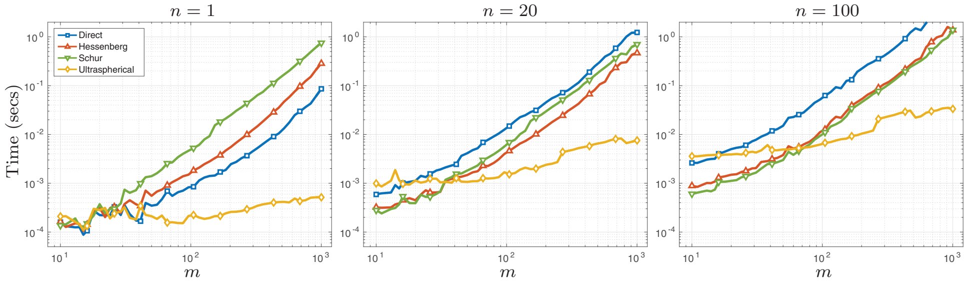


FIG. 8. *Time taken to solve the PIAS problem* (4.1) *using various factorizations of the* $m \times m$ *discretization in space and* $n = 1, 20, 100$ *quadrature nodes, respectively. Caching effects present an unfair advantage to the Schur factorization if one averages over a number of runs, so the results shown are from just a single run (and hence are not very smooth).*

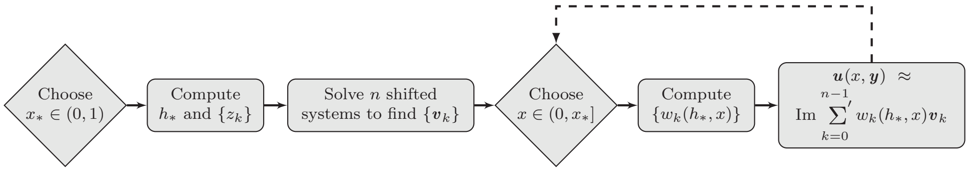

Choose $x_* \in (0,1)$
Compute h_* and $\{z_k\}$
Solve n shifted systems to find $\{\boldsymbol{v}_k\}$
Choose $x \in (0, x_*]$
Compute $\{w_k(h_*, x)\}$
$\boldsymbol{u}(x, \boldsymbol{y}) \approx \mathrm{Im} \sum_{k=0}^{n-1}{}' w_k(h_*, x)\boldsymbol{v}_k$

FIG. 9. *Algorithm for the efficient computation of the solution at multiple values of* x*. The main computational effort lies in the third box, where the linear systems are solved. After this, evaluating at any* $x \in (0, x_*]$ *simply requires taking a linear combination of the solution vectors.*

once, and the resulting vectors $\boldsymbol{v}_k$ are used for evaluating the quadrature sum (2.6) for all other values of x, just with different weights (as summarized in Figure 9). For most of these x values the step size h will not be optimal, but because the dependence of h on x is only logarithmic in (2.21) the loss of accuracy will be compensated for by the fact that far fewer systems have to be solved. Figure 10 shows the accuracy achieved by fixing such a value of x_* and solving a single set of resolvent systems. The results show that for $x \in (0, x_*)$ there is a nominal loss of accuracy because of the nonoptimal step size, but when $x \in (x_*, 1)$ the loss can be significant. Hence, when solving for multiple values of x, the optimal strategy is to choose x_* to be the largest x value required.

4.2. Example 2: Laplace equation in a box. Going to higher-dimensional problems, the numerical linear algebra becomes more challenging. Using the factorizations outlined above can help, but when the operator A can be expressed in Kronecker product form, Sylvester or Lyapunov solvers offer an even more efficient computational procedure.

As an illustration, we consider the PIAB problem mentioned in the introduction. That is, we solve

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y_1^2} + \frac{\partial^2 u}{\partial y_2^2} = 0, \qquad (x, y_1, y_2) \in [0,1] \times [0, 0.1] \times [0, 0.1], \tag{4.4}$$

subject to $u(x, 0, 0) = 1$ for $x \in (0,1)$ and $u = 0$ on all other sides (Figure 11 (left)). Taking advantage of symmetry, the 2D problem we must solve for each term in the sum (2.6) is therefore

$$z_k u - \left(\frac{\partial^2 u}{\partial y_1^2} + \frac{\partial^2 u}{\partial y_2^2}\right) = 2, \qquad (y_1, y_2) \in [0.05, 0.1] \times [0.05, 0.1], \tag{4.5}$$

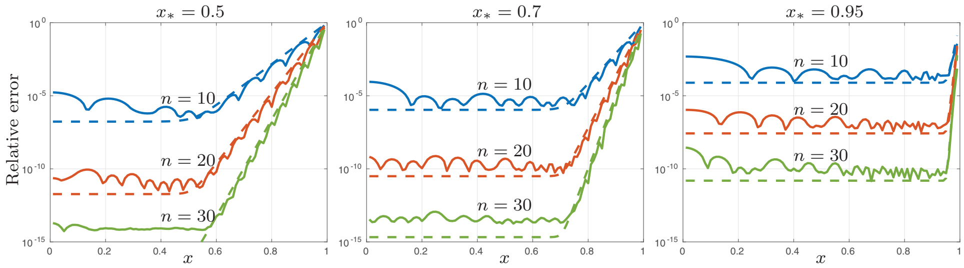


FIG. 10. *Relative error for the PIAS problem when using x_* to determine the step size h_* in* (2.21). *The solid lines denote the numerical errors (using spectral collocation with $m = 32$, 38, and 83, respectively), while the dashed lines show the theoretical errors $T + D$ from* (2.22). *The error is computed as the maximum error at the collocation nodes. Note that each of the nine solid curves is computed from a single set of solutions of the resolvent systems* (4.3)*, as summarized in Figure* 9. *When solving for multiple slices the optimal strategy is to pick x_* as the largest slice.*

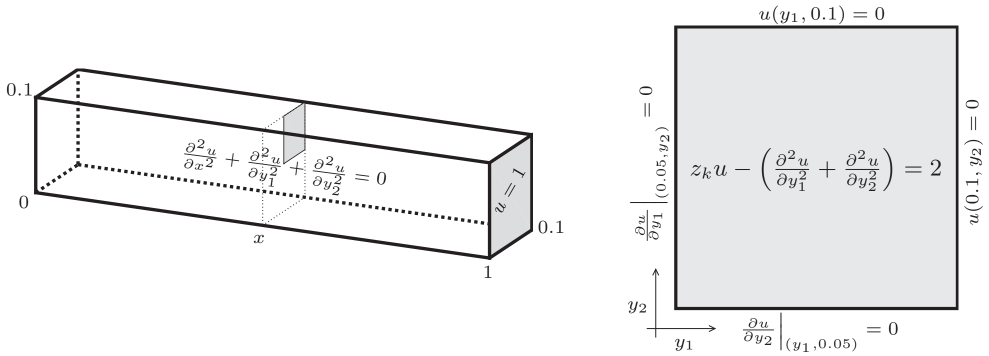

0.1
0
x
1
0.1
$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y_1^2} + \frac{\partial^2 u}{\partial y_2^2} = 0$
$u = 1$
$u(y_1, 0.1) = 0$
$\left.\frac{\partial u}{\partial y_1}\right|_{(0.05, y_2)} = 0$
$z_k u - \left(\frac{\partial^2 u}{\partial y_1^2} + \frac{\partial^2 u}{\partial y_2^2}\right) = 2$
$u(0.1, y_2) = 0$
y_2
y_1
$\left.\frac{\partial u}{\partial y_2}\right|_{(y_1, 0.05)} = 0$

FIG. 11. *Left: Domain for the PIAB problem* (4.4). *The boundary conditions are homogeneous Dirichlet on all boundaries except the shaded face, where $u(1, y_1, y_2) = 1$. The shaded region in the interior of the box depicts the location of the reduced* 2*D problem shown in Figure* 11 *(right). Right: Domain and boundary conditions for the symmetrized* 2*D problem* (4.5)–(4.6).

subject to

$$u(0.1, y_2) = u(y_1, 0.1) = \left.\frac{\partial u}{\partial y_1}\right|_{(0.05, y_2)} = \left.\frac{\partial u}{\partial y_2}\right|_{(y_1, 0.05)} = 0, \tag{4.6}$$

as shown in Figure 11 (right).

In the code shown in Figure 12 we use `DMSUITE` [26] to compute the $m \times m$ second-order Chebyshev differentiation matrix D_C^2 with the boundary conditions (4.6) incorporated via basis recombination [3, section 6.5]. The discretized version of (4.5) can then be expressed as

$$\left(\frac{1}{2} z_k I - D_C^2\right) U + U \left(\frac{1}{2} z_k I - D_C^2\right)^T - \mathit{2} = 0, \tag{4.7}$$

where I is the $m \times m$ identity matrix and $\mathit{2}$ is the $m \times m$ matrix of twos. This matrix equation takes the form of a Sylvester equation and we employ the MATLAB function

```
 1  % Solve the particle in a box problem.
 2  b = 0.1;                                % Aspect ratio of box.
 3  x = 0.5;                                % Evaluate in center of box.
 4  n = 20;                                 % Number of resolvent solves.
 5  m = 20;                                 % Space discretization size.
 6  [~, DC2] = cheb2bc(m, [0 1 0 ; 1 0 0]); % Neumann condition one side
 7                                          %   and Dirichlet at the other.
 8  DC2 = (4/b)^2*DC2;                      % Scaling.
 9  U1 = 2*ones(m - 1);                     % RHS.
10  [S, DC2] = schur(DC2, 'complex');       % Compute Schur factorisation only once.
11  U1 = S'*U1*S;                           % Transform the RHS.
12  l2 = 2*(pi/b)^2;                        % Smallest eigenvalue of operator.
13  [z, w] = nodesandweights(n, x, l2);     % Quadrature nodes and weights.
14  % Solve resolvent systems:
15  U = 0; I = eye(m-1);
16  for k = 1:n                             % <-- This can be changed to parfor.
17      vk = lyap( z(k)/2*I - DC2, z(k)/2*I - DC2.', -U1 );
18      U = U + w(k)*vk;
19  end
20  U = imag(S*U*S');                       % Undo the Schur factorization.
21  uApprox = U(1)                          % Computed solution (center of box).
22  uExact = 7.2988176570485260889e-10;     % Know solution (for x = 0.5, b = 0.1).
23  relErr = abs((uApprox - uExact)/uExact) % Relative Error.
```

FIG. 12. *MATLAB code to solve the PIAB problem* (4.4). *The function* cheb2bc *is part of the* DMSUITE *package* [26]. *The code takes around* 0.01 *seconds on a desktop computer and returns a relative error of* 8.7×10^{-13} *compared to the known solution* $7.2988176570485260889 \times 10^{-10}$ [25].

lyap to solve it efficiently (see line 17 of Figure 12).[2] Since we are interested in solving (4.7) for a number of different z_k, we can further increase efficiency by computing the Schur factorization of D_C^2 required by the Sylvester solver only once (see lines 10-11 and 20).

Figure 13 (left) shows just how much more efficient the solution procedure becomes when we use the Sylvester solver and Schur factorization. Figure 13 (right) shows the rate of convergence as we modify the code from Figure 12 to run with $n = 1, 2, \ldots, 20$. The dashed line shows the estimated convergence rate from (2.22), which agrees well with the numerical results. The plateau in the convergence curve is due to the ill-conditioning of the spectral collocation matrix.

4.3. Example 3: Laplace equation in an annular cylinder. For a more interesting domain than the rectangle or box configurations considered above, consider Laplace's equation in cylindrical coordinates

$$\frac{\partial^2 u}{\partial x^2} + Au = 0, \qquad A = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2}, \tag{4.8}$$

which we will solve in the domain $(x, r) \in [0, 1] \times [1, 3]$ (Figure 14). For boundary conditions, let $u = 0$ on all sides except for the annular face at $x = 1$, where

$$u(1, r, \theta) = u_1(r, \theta) = (r - 1)(3 - r)(1 - \sin(\theta)). \tag{4.9}$$

The operator A may be written in rank 2 Kronecker form as

$$A = \left(D_r^2 + R^{-1}D_r\right) \otimes I + R^{-2} \otimes D_\theta^2, \tag{4.10}$$

[2] lyap is part of the MATLAB Control System Toolbox, but a free alternative is available [22]. If z_k were real, then (4.7) would take the form of a continuous Lyapunov equation, which can be solved more efficiently than a general Sylvester equation. For complex z_k the system is only Lyapunov-like (as the transpose in (4.7) is not the Hermitian transpose), and although in principle one could exploit this to solve (4.7) more efficiently, the MATLAB implementation of lyap does not do so.

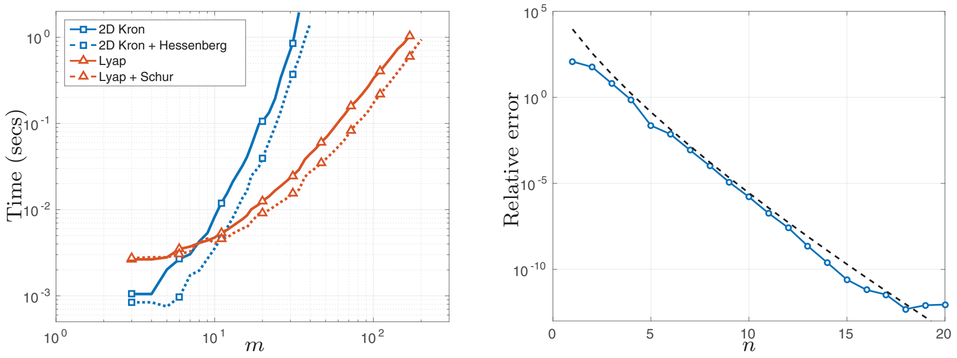


FIG. 13. *Left: Computation times as the matrix size, m, in the code in Figure* 12 *is increased* (n *is fixed at* 20). *The different curves represent the different algorithms and factorizations as described in the text. When $m > 10$ it becomes significantly faster to use the Sylvester solver* `lyap` *than to solve directly (for example, using the MATLAB* `kron` *and* `backslash` *functions) and faster still to precompute the Schur factorizations (as implemented in Figure* 12). *Right: Convergence to the solution of the PIAB problem as the number of resolvent solves (i.e., number of quadrature nodes), n, in Figure* 12 *is increased* (m *is fixed at* 20). *The solid line shows the numerical results, whereas the dashed line is the slope predicted by* (2.22).

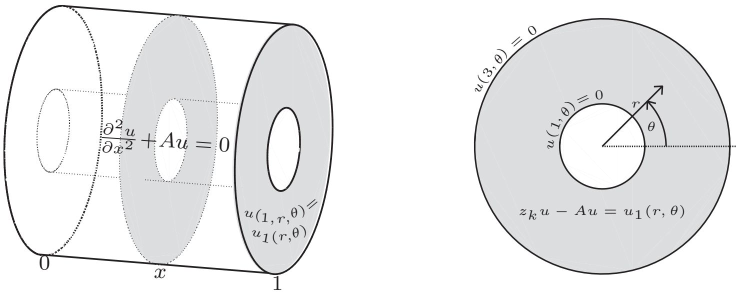

$\frac{\partial^2 u}{\partial x^2} + Au = 0$
$u(1,r,\theta) = u_1(r,\theta)$
0
x
1
$u(3,\theta) = 0$
$u(1,\theta) = 0$
r
θ
$z_k u - Au = u_1(r,\theta)$

FIG. 14. *Left: Domain for the problem* (4.8). *The boundary conditions are zero on all boundaries except the shaded face, where $u(1,r,\theta) = u_1(r,\theta)$. Right: Slice in x of the domain. The angular and radial coordinates, θ and r, are discretized using Fourier and Chebyshev collocation, respectively.*

as may the shifted operator

$$zI - A = \left(zI - D_r^2 - R^{-1}D_r\right) \otimes I - R^{-2} \otimes D_\theta^2, \tag{4.11}$$

where D_r, D_r^2, and D_θ are differential operators, R is a multiplication operator, and I is the identity. For our discretization we again use spectral collocation, but this time using Chebyshev expansion in the r direction and Fourier expansion in the θ direction ($0 \le \theta < 2\pi$) to account for the periodicity. Discretizing the operators in (4.11) the resolvent systems from (2.6) then become

$$\left(z_k I - D_C^2 - R^{-1}D_C\right)U - R^{-2}U{D_F^2}^T - U_1 = 0, \tag{4.12}$$

where D_C and D_C^2 are the first- and second-order Chebyshev differentiation matrices with Dirichlet boundary conditions enforced by basis recombination, R is the diagonal matrix whose entries are the interior Chebyshev points shifted to $(1,3)$, D_F^2 is the second-order Fourier differentiation matrix, and $[U_1]_{ij} = u_1(r_j, \theta_i)$.

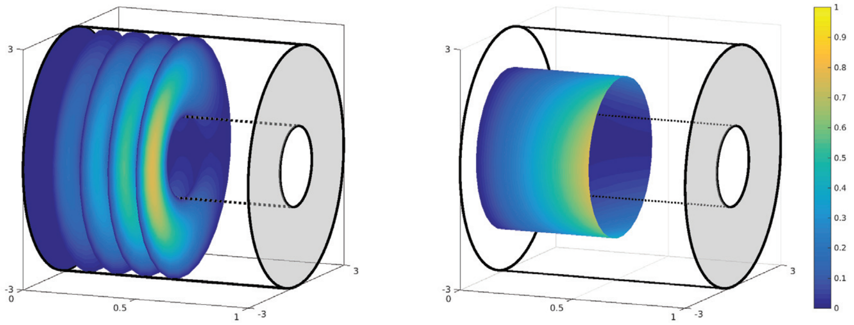


FIG. 15. *Solution to* (4.8)–(4.9) *at* 5 *slices below* $x = 0.5$ *(left) and at* $r = 2, x \leq 0.5$ *(right) obtained using the approach outlined in Figure* 9. *Using* 21 *quadrature nodes and* 21 *collocation points in each direction yields an accuracy of around* 10 *digits and takes around* 0.02 *seconds. By contrast, using a full* 3D *approximation using an additional* 21*-point Chebyshev discretization in the* x *direction takes around* 3 *seconds. This difference in timing becomes even more pronounced as the number of collocation points is increased.*

Here (4.12) is a more general Sylvester form than (4.7), but we may multiply through by $-R^2$ to obtain

$$(4.13) \qquad \left(R^2D_C^2 + RD_C - z_kR^2\right)U + U{D_F^2}^T + R^2U_1 = 0,$$

which can be solved by `lyap`. Similar to the previous example, we can save some time by computing the Schur factorizations of D_F^2 only once. (In fact, in this instance the factorization is known explicitly in terms of the unitary DFT matrix.) However, since the shifts in the term multiplying U from the left in (4.13) are no longer constant diagonals, their Schur factorizations must be computed for each z_k.

Rather than repeat the timing and convergence plots of Figure 13 we offer Figure 15, in which we solve the problem (4.8)–(4.9) for all $x \leq x_* = 0.5$ following the approach outlined in Figure 9. In particular, using $n = 21$ quadrature nodes and $m = 21$ collocation points in both the r and θ directions is sufficient to provide a relative accuracy of around 10 digits and takes around 2 hundredths of a second. By contrast, using a full 3D approximation using an additional 21-point Chebyshev discretization in the x direction takes around 3 seconds. Changing to $m = 31$ these times become around 4–5 hundredths of a second for the contour integral approach and over 3 minutes for the full 3D discretization. Increasing m further eventually leads to memory problems in the full 3D case.

5. Extensions.

5.1. The case $x \to 1$. As pointed out in the last paragraph of section 2, the contour integral method and its error analysis break down when $x \to 1$. The double exponential decay in the integrand of (2.3) is lost and the trapezoidal rule loses its efficiency. A remedy suggested in [10, section 4.3.2] is to rewrite (1.3) as

$$(5.1) \quad u(x, y) = \frac{1}{2\pi i}\int_{-\infty}^{\infty} E(x; z(\theta))\Big((z(\theta)I - A)^{-1} - (z(\theta) - \pi^2)^{-1}I\Big)f(y)z'(\theta)\,d\theta,$$

which is permitted by Cauchy's theorem. To be precise, in that reference a term $z(\theta)^{-1}$ is subtracted. Here we shift the location of this pole so that it lies on the boundary of the strip of analyticity (Figure 2), not within it. Now, as $|z| \to \infty$ the quantity involving the resolvent (i.e., inside the larger brackets) satisfies $O(z^{-1})$ in (2.3) but $O(z^{-2})$ in (5.1). Because of the exponential behavior of the mapping (2.2) as $\theta \to \pm\infty$, exponential decay has therefore been recovered when $x \to 1$, but the double exponential decay has been forfeited. We now consider optimal parameters and the associated convergence rate for this situation.

The estimate for the discretization error (2.17) remains unchanged, but a new estimate for the truncation error (2.19) is required since $E(x, z(\theta)) \to 1$ as $x \to 1$. Making use of (2.18), we can combine the $z(\theta)^{-2}$ decay mentioned above with the growth of $z'(\theta)$ to estimate

$$T = \mathscr{O}(e^{-\theta_n}). \tag{5.2}$$

Matching this with $D = \mathscr{O}(e^{-\pi^2/h})$ requires $nh_* = \pi^2/h_*$, and so

$$h_* = \pi/\sqrt{n}, \qquad T + D = \mathscr{O}(e^{-\pi\sqrt{n}}). \tag{5.3}$$

Comparing (5.3) with (2.21)–(2.22) we see the break-even point occurs when

$$\pi\sqrt{n} = 2W\left(\frac{\sqrt{2}\pi^2 n}{(1-x)\sqrt{\pi^2+\ell^2}}\right), \tag{5.4}$$

from which it follows that the break-even values of x for a given value of n, $x_b(n)$, satisfy

$$1 - x_b(n) = 2\pi e^{-\pi\sqrt{n}/2}\sqrt{\frac{2n}{\pi^2+\ell^2}}. \tag{5.5}$$

Alternatively, solving (5.4) for n and substituting this to $T + D$ in (5.3) we define

$$(T + D)_b(x) := e^{2W_{-1}\left(-(1-x)\sqrt{2(\pi^2+\ell^2)}/8\right)}, \tag{5.6}$$

where W_{-1} denotes the lower branch of the W-function. For a given x, this expression indicates approximately where the errors from using h_* defined by (2.21) and (5.3) are equal.

Now, as the error is uniform in x, we can solve for multiple x slices in much the same way as described in section 4.1. Figure 16 shows how the first two steps in the optimal strategy for solving single or multiple slices is adapted to account for the case when $x \approx 1$. If x is large enough, or if the tolerance is small enough, then the modified integral (5.1) with the modified step size h_* from (5.3) should be used; otherwise it remains more efficient to use the algorithm as described in sections 2–4. In Figure 17 (right) we see that for a tolerance of 10^{-10} in the PIAS problem (1.4)–(1.5) the break-even value of n is around 60. Setting $\ell = 0$ and solving (5.6) equal to 10^{-15} we find that for any $\ell > 0$ it is not until $x_* > 0.999999$ that the modified formula obtains 15 digits before the one given in section 2.

5.2. The case $\boldsymbol{\ell \gg 1}$. When $\ell \gg 1$ the implied constant $e^{-(1-x)\ell}$ in the formula for M_+ mentioned in section 2 becomes significant. To exploit this additional decay it becomes advantageous to replace the vertical line of integration as defined by (2.2) with a hyperbola that begins and ends in the left half-plane. That is, we consider for $0 < \alpha < \frac{1}{2}\pi$ the contour

$$\Gamma:\ z = \frac{1}{2}(\pi^2 - \ell^2) + \frac{i}{2}(\pi^2 + \ell^2)\sinh(\theta + i\alpha), \qquad -\infty < \theta < \infty. \tag{5.7}$$

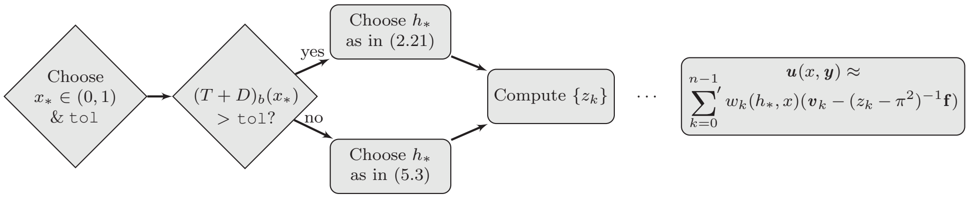


FIG. 16. *The required modifications of Figure* 9 *to account for the case* $x \approx 1$ *by using the quadrature approximation to* (5.1)*. The boxes on the left deal with deciding which step size to use and the box on the right accounts for the* $(z-\pi^2)^{-1}$ *term.*

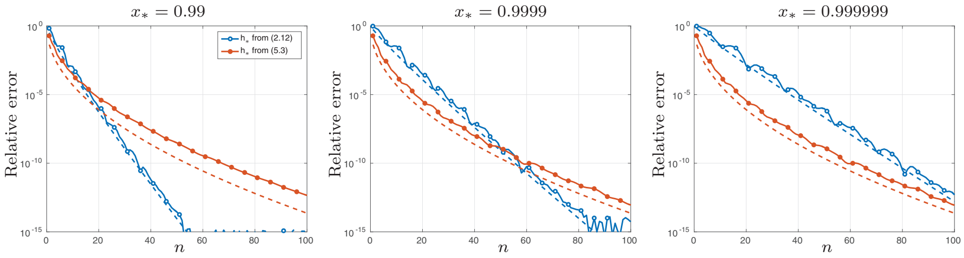


FIG. 17. *Convergence of the quadrature approximation to* (5.1) *using the optimal step sizes,* h_**, defined by* (2.21) *(open circles) and* (5.3) *(solid circles) for the PIAS square problem* (1.4)–(1.5)*. As* $x \to 1$ *the constant in the (almost) geometric rate obtained when using the former decreases but the* $\mathscr{O}(e^{-\pi\sqrt{n}})$ *convergence from* (5.3) *is essentially independent of* x*. However, we see that unless* x *is very close to* 1 *or the accuracy requirements are not particularly stringent, then the step size given by* (2.21) *is typically more efficient.*

It is straightforward to extend the error analysis of section 2 to this contour. The main point to keep in mind is that the contour of integration in the w-plane is no longer the real θ-axis but rather the horizontal line $\theta + i\alpha$ in the upper half-plane. This has the consequence that the strip of analyticity is no longer symmetric but has half-widths $\frac{1}{2}\pi + \alpha$ and $\frac{1}{2}\pi - \alpha$ in the the lower and upper half-planes, respectively. A similar computation was done for parabolic PDEs in [27].

Adjusting for the nonsymmetric strip as in the latter reference, one obtains two discretization errors, namely,

$$D_- = \mathscr{O}(e^{-2\pi(\pi/2+\alpha)/h}), \qquad D_+ = \mathscr{O}(e^{-2\pi(\pi/2-\alpha)/h-(1-x)\ell}). \tag{5.8}$$

The formula for the truncation error (5.10) should be modified as well. From (5.7)

$$z(\theta_n) \sim \frac{i}{4}(\pi^2+\ell^2)e^{\theta_n+i\alpha}, \quad n \to \infty, \tag{5.9}$$

and therefore we put $r = \frac{1}{4}(\pi^2+\ell^2)e^{\theta_n}$, $\phi = \frac{1}{2}\pi + \alpha$ into (1.10) to obtain

$$T = \mathscr{O}(e^{-\frac{1}{2}(1-x)\sqrt{\pi^2+\ell^2}\,e^{nh/2}\sin(\alpha/2+\pi/4)}). \tag{5.10}$$

A comparison of the exponential arguments of the three error terms leads to a system of two equations for solving for the unknowns h and α in terms of x, ℓ, and n. A general closed form solution appears out of reach, but the two equations can be reduced to a single scalar equation in α that can then be solved numerically. Specifically, by comparing the exponential arguments of the error terms above, it is possible to obtain

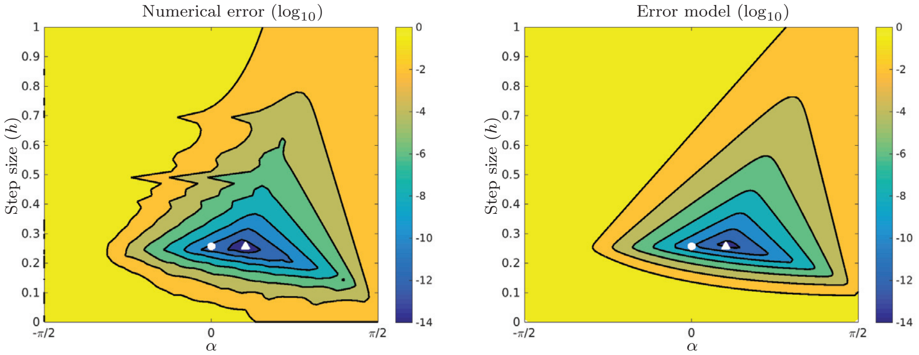


FIG. 18. *Solution of the PIAB problem of section* 4.2 *with* $b = 0.1$*,* $x = 0.5$*, and* $n = 15$*. On the left, numerically computed errors are shown. On the right, the theoretical error as defined by the sum of the terms in* (5.8) *and* (5.10) *is shown. The white dot represents the contour parameters of section* 2*, i.e., the vertical contour is used* ($\alpha = 0$) *with step size defined by* (5.11)*. The white triangle represents an improvement of about two or three orders of magnitude if the vertical contour is replaced by a hyperbola in the left half-plane (with* α *defined by* (5.12)*).*

the following expressions for h:

$$(5.11) \qquad h = \frac{4\,\pi\,\alpha}{(1-x)\,\ell}, \qquad h = \frac{2}{n} W\left(\frac{n\pi(\pi + 2\alpha)}{(1-x)\sqrt{\pi^2+\ell^2}\,\sin(\alpha/2+\pi/4)}\right).$$

The first expression follows from a comparison of D_- and D_+. The second expression, which is identical to (2.21) in the case $\alpha = 0$, follows from a comparison of D_- and T. By setting the two expressions for h equal one obtains the single equation for α mentioned above.

Assuming the $\alpha > 0$ case is a small perturbation of the $\alpha = 0$ case (which would be the situation for moderate ℓ), the second expression in (5.11) can be expanded as a series in α. When only leading-order terms are retained one finds

$$(5.12) \qquad \alpha_* \approx \frac{(1-x)\ell}{2\pi n} W\left(\frac{\sqrt{2}\,\pi^2 n}{(1-x)\sqrt{\pi^2+\ell^2}}\right).$$

Note that the corresponding step size, given by the first formula in (5.11), is then identical to the value used before, namely, (2.21).

As validation of (5.12) we offer Figure 18, in which we solve the PIAR problem from section 3 with $b = 0.1$ for fixed value of $n = 15$, but with varying step size, h, and shift, α. The left panel of Figure 18 shows the numerical error, while the right panel shows the error model resulting from (5.8) and (5.10). We see that, at least in the area of interest, our model is a good approximation to the error and that the choice of α_* in (5.12) is near optimal.

5.3. Other boundary conditions. Our focus in this paper has been on the PDE (1.1) with solution $u(x, y) = E(x; A)f(y)$ with $E(x; z)$ as in (1.2). However, as summarized in Table 1, one can express the solution to (1.1) with different boundary conditions in a similar manner by using a different function $E(x; z)$.

The analysis presented in section 2 for the particular case of (1.2) can be readily extended to these other boundary conditions. In particular, each of the $E(x; z)$ functions in Table 1 satisfies the asymptotic behavior (1.10). The main difference is

Table 1

Functions $E(x;z)$ corresponding to alternative boundary conditions in (1.1) and the location of their poles, p_k, on the positive real axis. The table can be extended to include mixed or Robin conditions, but to avoid clutter these are omitted. The situation where an inhomogeneous constraint appears on the left boundary is readily handled by the use of symmetry and the principle of superposition.

Left boundary	Right boundary	$E(x;z)$	$p_k, k=1,2,\ldots$
$u(0,y)=0$	$u(1,y)=f(y)$	$\big(\sin(\sqrt{z})\big)^{-1}\sin(x\sqrt{z})$	$(k\pi)^2$
$u_x(0,y)=0$	$u(1,y)=f(y)$	$\big(\cos(\sqrt{z})\big)^{-1}\cos(x\sqrt{z})$	$\frac{1}{4}(k\pi)^2$
$u(0,y)=0$	$u_x(1,y)=f(y)$	$\big(\sqrt{z}\cos(\sqrt{z})\big)^{-1}\sin(x\sqrt{z})$	$\frac{1}{4}(k\pi)^2$
$u_x(0,y)=0$	$u_x(1,y)=f(y)$	$\big(\sqrt{z}\sin(\sqrt{z})\big)^{-1}\cos(x\sqrt{z})$	$((k-1)\pi)^2$

the location of their singularities on the positive real axis, given as p_k in the final column of Table 1. Adjusting the conformal map (2.1) to account for p_1, the singularity closest to the origin, we obtain

$$z(\theta) = \frac{1}{2}(p_1 - \ell^2) + \frac{i}{2}(p_1 + \ell^2)\sinh\theta \tag{5.13}$$

and, after matching the truncation and discretization errors as in section 2, the optimal step sizes

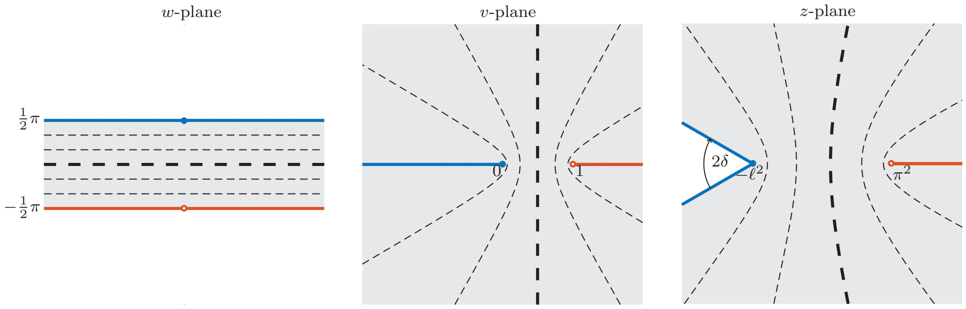
$$h_* = \frac{2}{n}W\left(\frac{\sqrt{2}\,\pi^2\,n}{(1-x)\sqrt{p_1+\ell^2}}\right). \tag{5.14}$$

Since, for $x \geq 0$, $W(x)$ is monotonically increasing, substituting (5.14) to the formula (2.22) for the total error we find that, for a fixed value of ℓ, the contour integral method using the conformal map (5.13) and step size (5.14) will converge fastest for the case of Dirichlet–Dirichlet boundary conditions and slowest for Neumann–Neumann (with the Neumann–Dirichlet and Dirichlet–Neumann cases being somewhere in between). If $\ell^2 \gg p_1$, then this difference will be hardly noticeable. However, if A is singular, i.e., $\ell = 0$, then in the Neumann–Neumann case the spectrum of A coincides with p_1 at the origin and we cannot find a suitable contour for the integral (1.3). However, since the problem is ill-posed in this instance (i.e., the solution is defined only up to a constant) this is to be expected.

5.4. Imaginary spectrum. We now briefly consider the case of operators whose spectra lie within the wedge

$$S_{\ell,\delta}:\ z = -\ell^2 + re^{i\phi}, \quad r \geq 0, \quad |\phi| \geq \pi - \delta, \quad \delta \leq \tfrac{1}{2}\pi, \tag{5.15}$$

shown as the unshaded region in Figure 19(c). When Gavrilyuk and coworkers consider sectorial operators in [9, 10] they continue to use a hyperbolic map but modify their parameters to take account of the wedge. We take a different approach by choosing the conformal map from the infinite strip in Figure 19(a) to precisely the region $S_{\ell,\delta}$. To derive this we observe, as in section 2, that $\sinh(w)$ maps the infinite strip of half-width $\pi/2$ to the double-slit region with slit tips at ±1, and hence $v = (i\sinh(w)+1)/2$ maps to the double-slit region with tips at $v=0$ and $v=1$ shown in Figure 19(b). Choosing the branch cut to be the negative real axis, $v^{(1-\delta/\pi)}$ will open up a wedge of the appropriate half-angle, δ, and so, combined with a suitable



(a) Infinite strip (b) Double slit-plane (c) Sectorial region

FIG. 19. *Conformal map from infinite strip to the exterior of the wedge-like region, $S_{\ell,\delta}$.*

shift and scaling, we have that $z = (\ell^2 + \pi^2)v^{(1-\delta/\pi)} - \ell^2$ will map from the double-slit region to $S_{\ell,\delta}$. Combining the two stages gives

$$\text{(5.16)} \qquad \Gamma: \; z = (\pi^2 + \ell^2)\left(\tfrac{1}{2} + \tfrac{1}{2}i\sinh(\theta + i\alpha)\right)^{1-\delta/\pi} - \ell^2, \qquad -\infty < \theta < \infty.$$

Note that if $\delta = 0$, then (5.16) reduces to (5.7). Figure 19(c) shows the images of horizontal lines in the w plane under the transformation defined by (5.16). Unlike in Figure 2 these will not be hyperbolas for any $\delta \neq 0$. Furthermore, the image of the center line is no longer a vertical line, even if $\alpha = 0$.

We now reproduce the results of section 2 and section 5.2 for this case. Details of the derivation are omitted as these are almost identical to the $\delta = 0$ case. All the formulas given below reduce precisely to those of section 5.2 when $\delta = 0$ and those of section 2 when both $\delta = 0$ and $\alpha = 0$.

Considering (5.16) we find that if $z = re^{i\phi}$, then

$$\text{(5.17)} \qquad r \sim (\pi^2 + \ell^2)\left(\tfrac{1}{2}e^{\theta}\right)^{1-\delta/\pi}, \quad \phi \sim (\tfrac{1}{2}\pi + \alpha)(1 - \delta/\pi), \quad \theta \to \infty,$$

which, substituted into (1.10), gives a truncation error

$$\text{(5.18)} \qquad T = \mathscr{O}\left(e^{-(1-x)\sqrt{\pi^2+\ell^2}\sin\left(\alpha\frac{\pi+\delta}{2\pi} + \frac{\pi-\delta}{4}\right)e^{nh(1-\delta/\pi)/2}/2^{1-\delta/\pi}}\right).$$

For $\delta \leq \frac{1}{2}\pi$ the discretization errors are essentially the same as in (5.8); thus, comparing T and D_- gives

$$\text{(5.19)} \qquad h = \frac{2}{n(1-\delta/\pi)}W\left(\frac{n\pi 2^{-\delta/\pi}(\pi + 2\alpha)}{(1-x)\sqrt{\ell^2+\pi^2}\sin\left(\frac{\pi-\delta}{4} + \alpha\frac{\pi-\delta}{2\pi}\right)}\right).$$

Comparing with the first expression in (5.11), one can solve numerically for the optimal values of α_* and h_*.

6. Conclusion.

There were two distinct parts to this paper. In sections 2–3 we revisited the contour integral method proposed by Gavrilyuk and coworkers [8, 9, 10] for the solution of elliptic PDEs posed on cylindrical domains and investigated its use in a more practical setting. We derived explicit formulas for the optimal step size and compared the resulting quadrature approximation to the standard series solution for the textbook problem of Laplace's equation on a square. The techniques used were similar to those for parabolic PDEs discussed in [14, 27, 16], but because of additional singularities in the integrand in the elliptic case the optimal convergence rate is only subgeometric and not fully geometric, as is the case for parabolic PDEs.

In the second part, section 4, we focused on efficient techniques for solving the resolvent systems in the case when the operator A is discretized by spectral collocation. We demonstrated that execution speed can be improved significantly by a priori factorization strategies, such as the Hessenberg and Schur forms. When the operator can be expressed in Kronecker product form, a major additional advantage can be gained from the use of specialized Sylvester or Lyapunov solvers. Timing comparisons have shown that such an implementation is faster by orders of magnitude than the standard approach of a full 3D tensor product formulation. These same factorizations are also applicable in the case of parabolic PDEs, but as far as we are aware they have received little attention in the literature.

The optimal contour and parameters derived in the first part are independent of the discretization of the operator A and hence are equally applicable in the case of finite difference or finite element methods, or even off-the-shelf software for the lower-dimensional subproblems. In many such instances the resulting matrices will be sparse and hence iterative methods are used to solve the linear systems, making the factorizations discussed above less suitable. Investigating preconditioning strategies for iterative methods is beyond the scope of this paper, but this has been explored in the case of parabolic PDEs [4, 14].

Finally, it should be emphasized that the method we discuss is not universally applicable to all elliptic PDEs. In particular, the method is intrinsically linked to PDEs posed on cylindrical domains, with the variable x playing the role of the time variable in the parabolic case. In addition, the operator A should be close to normal, with eigenvalues on or near the negative real axis, so that its resolvent norm is bounded away from this axis. On the other hand, much more general PDEs than those used as test examples in this paper can be solved by this method. This includes problems with different kinds of boundary conditions, a force term, or nonconstant coefficients in the operator A.

Contour integral methods are now well established as a means of solving parabolic PDEs. We believe that with the combination of effectively reducing the problem dimension, the trivial parallelization, and the possibility of avoiding corner singularities, they also provide a powerful tool for solving elliptic PDEs.

Acknowledgment. We thank the referees for numerous helpful suggestions.

REFERENCES

[1] D. A. Barry, *wapr: Real Values of the Lambert W Function*, http://www.mathworks.com/matlabcentral/fileexchange/3644 (25 June 2003).

[2] F. Bornemann, D. Laurie, S. Wagon, and J. Waldvogel, *The SIAM* 100-*Digit Challenge*, SIAM, Philadelphia, 2004.

[3] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Dover, Mineola, New York, 2001.

[4] K. Burrage, N. Hale, and D. Kay, *An efficient implicit FEM scheme for fractional-in-space reaction-diffusion equations*, SIAM J. Sci. Comput., 34 (2012), pp. A2145–A2172.

[5] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ, 1976.

[6] N. Dunford and J. T. Schwartz, *Linear Operators.* I. *General Theory*, Interscience Publishers, New York, 1958.

[7] I. P. Gavrilyuk, *Approximation of the operator exponential and applications*, Comput. Methods Appl. Math., 7 (2007), pp. 294–320.

[8] I. P. Gavrilyuk, W. Hackbusch, and B. N. Khoromskij, *$\mathscr{H}$-matrix approximation for elliptic solution operators in cylinder domains*, East-West J. Numer. Math., 9 (2001), pp. 25–58.

[9] I. P. Gavrilyuk, V. L. Makarov, and V. B. Vasylyk, *Exponentially convergent approximation to the elliptic solution operator*, Comput. Methods Appl. Math., 6 (2006), pp. 386–404.

[10] I. P. Gavrilyuk, V. L. Makarov, and V. B. Vasylyk, *Exponentially Convergent Algorithms for Abstract Differential Equations*, Front. Math., Birkhäuser/Springer Basel AG, Basel, Switzerland, 2011.

[11] N. Hale, N. J. Higham, and L. N. Trefethen, *Computing $\mathbf{A}^{\alpha}$, $\log(\mathbf{A})$, and related matrix functions by contour integrals*, SIAM J. Numer. Anal., 46 (2008), pp. 2505–2523.

[12] N. Hale and J. A. C. Weideman, *Companion Code for Hale & Weideman paper "Contour Integral Solution of Elliptic PDEs in Cylindrical Domains,"* `https://github.com/nickhale/contour2015` (24 September 2015).

[13] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.

[14] K. J. in 't Hout and J. A. C. Weideman, *A contour integral method for the Black–Scholes and Heston equations*, SIAM J. Sci. Comput., 33 (2011), pp. 763–785.

[15] M. López-Fernández and C. Palencia, *On the numerical inversion of the Laplace transform of certain holomorphic mappings*, Appl. Numer. Math., 51 (2004), pp. 289–303.

[16] M. López-Fernández, C. Palencia, and A. Schädle, *A spectral order method for inverting sectorial Laplace transforms*, SIAM J. Numer. Anal., 44 (2006), pp. 1332–1350.

[17] E. Martensen, *Zur numerischen Auswertung uneigentlicher Integrale*, Z. Angew. Math. Mech., 48 (1968), pp. T83–T85.

[18] M. Mori, *Discovery of the double exponential transformation and its developments*, Publ. Res. Inst. Math. Sci., 41 (2005), pp. 897–935.

[19] S. Olver and A. Townsend, *A fast and well-conditioned spectral method*, SIAM Rev., 55 (2013), pp. 462–489.

[20] H.-R. Schwarz, *Numerical Analysis*, Wiley, Chichester, UK, 1989.

[21] D. Sheen, I. H. Sloan, and V. Thomée, *A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature*, IMA J. Numer. Anal., 23 (2003), pp. 269–299.

[22] A. Townsend and N. Hale, *freeLYAP*, `https://github.com/ajt60gaibb/freeLYAP` (7 June 2015).

[23] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, Philadelphia, 2000.

[24] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra. The Behavior of Nonnormal Matrices and Operators*, Princeton University Press, Princeton, NJ, 2005.

[25] L. N. Trefethen and J. A. C. Weideman, *The exponentially convergent trapezoidal rule*, SIAM Rev., 56 (2014), pp. 385–458.

[26] J. A. C. Weideman and S. C. Reddy, *A MATLAB differentiation matrix suite*, ACM Trans. Math. Software, 26 (2000), pp. 465–519.

[27] J. A. C. Weideman and L. N. Trefethen, *Parabolic and hyperbolic contours for computing the Bromwich integral*, Math. Comp., 76 (2007), pp. 1341–1356.

[28] K. Yosida, *Functional Analysis*, 2nd ed., Grundlehren Math. Wiss. 123, Springer-Verlag, New York, 1968.