Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems

Tianbing Chen, John Strain [*]

Department of Mathematics, University of California, 970 Evans Hall #3840, Berkeley, CA 94720-3840, United States

Abstract

A new piecewise-polynomial interface method (PIM) for discretizing elliptic problems with complex interfaces between high-contrast materials is derived, analyzed and tested. A Krylov-accelerated interface multigrid approach (IMG) solves the discretization efficiently. Stability and convergence are proved in one dimension, while an extensive array of numerical experiments with complex interfaces and large coefficient transitions demonstrate the accuracy, efficiency and robustness of the method in two dimensions.
© 2008 Elsevier Inc. All rights reserved.

1. Introduction

Elliptic interface problems with discontinuous coefficients and singular sources are found in many applications and simulations [12,16,19]. Solutions of these problems are usually non-smooth or discontinuous across interfaces. For example, consider the following elliptic partial differential equation

$$\nabla \cdot (\beta \nabla u) - \kappa u = f \quad \text{in } \Omega^+ \cup \Omega^- \tag{1}$$

with Dirichlet boundary condition

$$u = g \quad \text{on } \partial \Omega,$$

where $\beta(x,y) \geqslant \beta_{\min} > 0, \kappa \geqslant 0$. Here Ω is a regular domain separated by an interface Γ into subdomains Ω^+ and Ω^- (see Fig. 1). Both the coefficient β and the source term f are typically discontinuous across the interface Γ, where the following jump conditions are prescribed:

$$[u] := u^+ - u^- = w, \quad [\beta u_n] := \beta^+ u_n^+ - \beta^- u_n^- = v$$

---

[*] Corresponding author.
E-mail addresses: ctbing@math.berkeley.edu (T. Chen), strain@math.berkeley.edu (J. Strain).
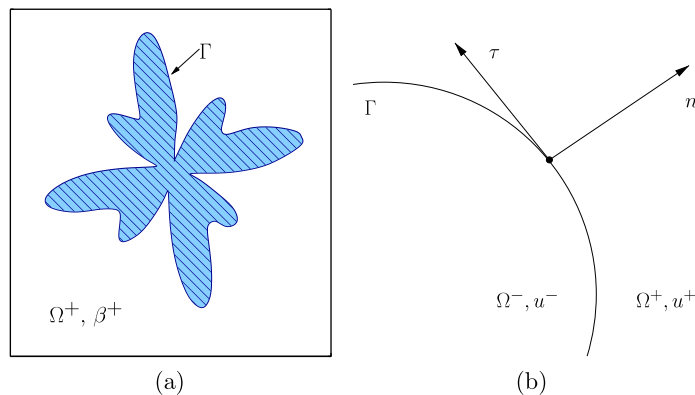
Fig. 1. (a) An irregular interface Γ partitions a regular domain: $\Omega = \Omega^+ \cup \Gamma \cup \Omega^-$ where β^+/β^- can be either small or large. (b) Normal and tangential directions along the interface Γ. Here $u^\pm = u|_{\Omega_\pm}$.

with w, v being functions defined along the interface Γ. In the special case that $w = v = 0$, the solution u is continuous but its normal derivative u_n has jumps across Γ whenever β^+ is different from β^-.

The challenge in solving elliptic interface problems is that interfaces can be very complex while the coefficients can be high-contrast. Consequently, it is difficult to use body-fitted unsteady grid to fit the evolving interfaces. A fixed Cartesian grid, where the interface can cut through the grid lines, is often used. A variety of methods have been proposed to deal with the grid-interface interaction [4,7,9,10,16,13–15,28,32,29–31,33,34].

LeVeque and Li proposed the immersed interface method (IIM) for solving elliptic equations with discontinuous coefficients and singular sources [13]. Global $O(h^2)$ accuracy is achieved by using the conventional $O(h^2)$ central scheme for regular points and a local $O(h)$ scheme for irregular points. A Taylor series expansion at the interface yields a set of linear equations for the undetermined coefficients and the correction term. A local $O(h)$ approximation requires jump conditions involving second derivatives. Various extensions and improvements have been considered in the literature. For interface problems with piecewise constant coefficients, a fast IIM was constructed by introducing an unknown jump condition $[u_n]$, to be solved numerically together with the elliptic equation [14]. The success of the fast IIM is based on the fact that the IIM produces the standard finite difference scheme with a correction term in 2D when $\beta^+ = \beta^- = 1$. As a result, standard fast Poisson solvers can be applied. For general elliptic problems, a maximum principle preserving immersed interface method (MIIM) forces the system matrix to be an M-matrix [15]. The coefficients of the finite difference scheme are found by solving a constrained optimization problem, guaranteeing both stability and global second-order accuracy.

Motivated by the fast IIM, the explicit jump IIM (EJIIM) introduces the high-order jumps at the intersections of the interface and the coordinate directions as auxiliary unknowns [28]. The interpolation equation for these high-order jumps is derived via a one-sided local polynomial approximation and the jump data. Numerical examples show that it may be critical to choose between exterior and interior points for interpolation accuracy.

Another second-order method for elliptic interface problems is the decomposed immersed interface method (DIIM), which decomposes the jump data along coordinate directions [4]. The method uses the standard central finite difference scheme for the left-hand side and introduces a correction term from jumps to the right-hand side, where high-order one-sided interpolation is used on both sides of the interface. The advantage is the coefficient matrix remains symmetric and diagonally dominant and thus most standard solvers can be applied. However, due to the fact that the right-hand side may involve large correction terms, a small parameter for successive under-relaxation is required to reach convergence.

A related high-order approach is the matched interface and boundary method (MIB) [34]. In each dimension, a high-order finite difference equation using grid data and jump data is derived through the help of fictitious points. In multiple dimensions, the jump data $[\beta u_x]$ in each coordinate direction is expressed in terms of

$[\beta u_n]$ and $[\beta u_\tau]$. The latter is obtained by a combination of the prescribed $[u_\tau]$ and one-sided interpolation of nearby grid values. Since this one-sided interpolation has to be second-order and would involve many grid points on one side, the method is limited to simple interfaces. For general irregular interfaces, the MIB was improved by disassociating the discretization and the domain extension [33]. To deal with sharp-edged interfaces, the concepts of primary and secondary fictitious values are introduced in [31], where second-order convergence is confirmed through numerical examples. However, the resulting scheme has a critical acute angle limitation because the scheme depends on a priori calculation of secondary fictitious values. This restriction is removed by using two sets of interface jump conditions in [29]. Interfaces with corner points are also addressed in [10], but the resulting method is only first-order for Lipschitz-continuous interfaces.

Recently a coupling interface method (CIM) has been introduced for solving elliptic interface problems [7]. It takes a dimension splitting approach and is derived from a linear/quadratic approximation on both sides of the interfaces in 1D. For higher dimensions, a coupled equation for the first-order derivatives is derived through the jump conditions in each coordinate direction. However, the CIM requires either that the interface intersects each grid segment at most once (first-order accurate) or that the interface does not intersect two adjacent grid segment simultaneously (second-order accurate). This restriction limits the application of the CIM to complex moving interface problems.

Another challenge of elliptic interface problems is to design a fast solver for the resulting matrix, which is typically unsymmetric [7,13,34] so standard fast solvers are not applicable. In [1–3], multigrid methods were designed specifically for interface problems discretized by immersed interface methods [13,15]. For interfaces with moderate curvature, the method proposed in [1] produced satisfactory performance. AMG is employed in [7] to solve the resulting linear system. It is observed that convergence becomes worse when the problem has high-contrast coefficients and thus many more iterations are needed to achieve reasonable accuracy [1].

Thus an interface method should have the following properties:

- The method produces reasonably accurate solutions on a given mesh, when interfaces are complex and/or there exist high-contrast coefficients. Typical examples are two interface points moving towards each other in 1D or interfaces developing acute angles in higher dimensions.
- There exists an associated fast solver, which incorporates jump conditions and converges independent of mesh size. The complexity of interfaces, as well as the ratio β^+/β^- has minimum effect on convergence rates.

This paper introduces a new piecewise-polynomial interface method (PIM) for elliptic problems. For 1D cases, we incorporate all possible jump conditions and differential equations in local approximations. As a result, the PIM employs the minimum number of grid points while enjoying second-order accuracy, even with multiple intersections. Explicit formulas are derived and rigorous analysis of general cases confirms stability and rate of convergence.

To extend the idea for higher dimensions, we introduce a least squares approach to determine the unknown coefficients for piecewise polynomials, due to its flexibility for complex interfaces. Various cases with multiple intersections are carefully addressed. Extensive numerical examples show that the PIM produces high-quality solutions for complex interfaces and high-contrast coefficients represented on coarse grids with multiple intersections. The interface may have corner points and/or high curvature. The method naturally extends to treat problems with multiple interfaces, general jump conditions and mixed boundary conditions.

We further apply the idea of the PIM to design a second-order accurate interpolator and thus a new interface multigrid solver (IMG). We provide spectral analysis to investigate convergence properties of the IMG. To enhance its robustness for problems with high-curvature interfaces and high-contrast coefficients, the IMG is used as preconditioner for Krylov subspace iterations. Among Krylov iterations, GMRES [22] and BiCG-STAB [26] are appropriate since our matrix is not symmetric. Numerical examples show that the IMG-preconditioned Krylov solver is very stable and fast. It usually takes less than 6 iterations for the relative residual to go below 10^{-10}, despite the fact that β^+/β^- can move between 10^3 and 10^{-3}. The computational time grows linearly in the number of unknowns.

Similar discretization approaches can be found in [32,33], in that all methods use Taylor series expansion for local approximation and use two physical jumps only. We note that [32,33] use a wider stencil in both 1D and higher dimensions and thus have trouble dealing with complex interfaces. For higher-dimensional cases,

[32,33] uses dimension splitting approaches with one-sided interpolation and have to refine the mesh when multiple intersections occur. On the other hand, the PIM uses higher-dimensional polynomials for approximation and produces accurate results without refining the mesh for complex interfaces with multiple intersections. Least squares techniques are also employed in [15], but the method there uses quadratic optimization techniques to solve an under-determined system with the sign property of the discrete maximum principle. The PIM proposes piecewise polynomials according to local geometries and then imposes at least as many equations, thus solving an overdetermined system, which has been shown to have full column rank. Thus the existence of solutions is guaranteed as well as the order of accuracy.

The paper is organized as follows: we describe the piecewise-polynomial interface method for 1D in Section 2. Section 3 extends the approach to 2D. Section 4 provides stability and convergence analysis for the PIM in 1D. We derive and analyze the new multigrid approach in Section 5. Section 6 presents GMRES(m) preconditioned by the multigrid solver. Numerical examples in Section 7 confirm the accuracy of the PIM and the efficiency of the new Krylov-accelerated multigrid solver on problems with high-contrast coefficients and complex interfaces. Comparisons with previous methods are also presented.

2. The piecewise-polynomial interface method in 1D

The PIM is most conveniently derived in one-dimensional geometry. Consider the 1D elliptic equation

$$Lu := (\beta u_x)_x = f(x), \quad 0 < x < 1 \tag{2}$$

along with the jump conditions

$$[u](\alpha_i) = w_i, \quad [\beta u_x](\alpha_i) = v_i, \quad i = 1, \ldots, n_\Gamma$$

on the domain $\Omega = (0, 1)$ divided into subdomains Ω^+ and Ω^- by a set of interface points $\Gamma := \{\alpha_1, \ldots, \alpha_{n_\Gamma}\}$. Dirichlet boundary conditions are imposed on $\partial\Omega$. We define the subdomains

$$\Omega^+ := \{x \in [0, 1] : \phi(x) > 0\}, \quad \Omega^- := \{x \in [0, 1] : \phi(x) < 0\},$$

where the phase function $\phi(x) := \prod_{i=1}^{n_\Gamma} (\alpha_i - x)$. A uniform grid on the interval $[0, 1]$ is given by

$$x_i = (i - 1)h, \quad 1 \leqslant i \leqslant N,$$

where $h = 1/(N - 1)$. We say x_i is a *regular point* if the interface does not separate any points in the standard three-point stencil $\{x_{i-1}, x_i, x_{i+1}\}$ centered at x_i. Otherwise, we say x_i is an *irregular point*.

Assume that (2) is approximated at each interior point x_i with $1 < i < N$ by

$$L_h U_i := \gamma_{i,1} U_{i-1} + \gamma_{i,2} U_i + \gamma_{i,3} U_{i+1} = F_i + C_i,$$

where $F_i = f(x_i)$ and C_i is a correction term which vanishes except near interfaces. The standard three-point stencil at an interior regular point x_i has

$$\gamma_{i,1} = \frac{\beta_{i-1/2}}{h^2}, \quad \gamma_{i,2} = \frac{-\beta_{i-1/2} - \beta_{i+1/2}}{h^2}, \quad \gamma_{i,1} = \frac{\beta_{i+1/2}}{h^2}, \quad C_i = 0.$$

At irregular points we use the modified stencil described below. For simplicity of notation, we shall assume that $\beta(x)$ is piecewise constant. The same approach works when $\beta(x)$ is piecewise smooth.

2.1. Single intersection

Suppose that α is the only interface point between x_{j-1} and x_{j+1}, and $x_j \leqslant \alpha = x_j + \theta h < x_{j+1}$ (see Fig. 2). We use a piecewise quadratic polynomial
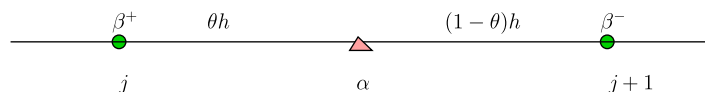

Fig. 2. A single-intersection case: $\alpha = x_j + \theta h$.

$$p(x) = \begin{cases} p_1(x) := a_1x^2 + a_2x + a_3 & x < \alpha, \\ p_2(x) := a_4x^2 + a_5x + a_6 & x > \alpha \end{cases} \tag{3}$$

to approximate the solution $u(x)$ near x_j, where $a_1, \ldots, a_6$ are undetermined coefficients. We assume that the following values are provided:

$$\begin{aligned} u(x_j) &= U_j, \quad u(x_{j+1}) = U_{j+1}, \\ [u](\alpha) &= w, \quad [\beta u_x](\alpha) = v, \\ Lu^+(\alpha) &= f^{(1)}, \quad Lu^-(\alpha) = f^{(2)}. \end{aligned} \tag{4}$$

Substituting (3) into (4) gives a linear system for the unknown coefficients:

$$\begin{bmatrix} x_j^2 & x_j & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{j+1}^2 & x_{j+1} & 1 \\ \alpha^2 & \alpha & 1 & -\alpha^2 & -\alpha & -1 \\ 2\beta^+\alpha & \beta^+ & 0 & -2\beta^-\alpha & -\beta^- & 0 \\ 2\beta^+ & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\beta^- & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} U_j \\ U_{j+1} \\ w \\ v \\ f^{(1)} \\ f^{(2)} \end{bmatrix}. \tag{5}$$

After solving for the coefficients in terms of the right-hand side, we obtain approximate solution values as linear combinations of data:

$$\widehat{U}_{j+1} \approx p_1(x_{j+1}) = a_1 x_{j+1}^2 + a_2 x_{j+1} + a_3 =: \lambda_{1,1}U_j + \lambda_{1,2}U_{j+1} + +\lambda_{1,3}w + \lambda_{1,4}v + \lambda_{1,5}f^{(1)} + \lambda_{1,6}f^{(2)}$$

and

$$\widehat{U}_j \approx p_2(x_j) = a_4 x_j^2 + a_5 x_j + a_6 =: \lambda_{2,1}U_j + \lambda_{2,2}U_{j+1} + \lambda_{2,3}w + \lambda_{2,4}v + \lambda_{2,5}f^{(1)} + \lambda_{2,6}f^{(2)}.$$

Symbolic computation gives explicit formulas for the coefficients $\lambda_{i,j}$:

$$\begin{aligned} \lambda_{1,1} &= 1 - \frac{\beta^-}{\hat{\beta}}, \quad \lambda_{2,1} = \frac{\beta^+}{\hat{\beta}}, \\ \lambda_{1,2} &= \frac{\beta^-}{\hat{\beta}}, \quad \lambda_{2,2} = 1 - \frac{\beta^+}{\hat{\beta}}, \\ \lambda_{1,3} &= \frac{\beta^-}{\hat{\beta}}, \quad \lambda_{2,3} = -\frac{\beta^+}{\hat{\beta}}, \\ \lambda_{1,4} &= \frac{(1-\theta)}{\hat{\beta}}h, \quad \lambda_{2,4} = \frac{\theta}{\hat{\beta}}h, \\ \lambda_{1,5} &= \frac{(1-\theta)(\hat{\beta}-\theta\beta^+)}{2\beta^+\hat{\beta}}h^2, \quad \lambda_{2,5} = -\frac{\theta^2}{2\hat{\beta}}h^2, \\ \lambda_{1,6} &= -\frac{(1-\theta)^2}{2\hat{\beta}}h^2, \quad \lambda_{2,6} = \frac{\theta(\hat{\beta}-(1-\theta)\beta^-)}{2\beta^-\hat{\beta}}h^2, \end{aligned} \tag{6}$$

where $\hat{\beta}$ is the weighted average

$$\hat{\beta} := \theta\beta^- + (1-\theta)\beta^+ > 0.$$

The modified stencils for j and $j+1$ are then

$$\begin{aligned} (\beta u_x)_x(x_j) &\approx \frac{\beta^+}{h^2}(U_{j-1} + \widehat{U}_{j+1} - 2U_j), \\ (\beta u_x)_x(x_{j+1}) &\approx \frac{\beta^-}{h^2}(\widehat{U}_j + U_{j+2} - 2U_{j+1}). \end{aligned}$$

As a result, we have

$$L_h U_j := \frac{\beta^+}{h^2}(\lambda_{1,2}U_{j+1} + U_{j-1} - (2 - \lambda_{1,1})U_j) = F_j + C_j,$$

$$L_h U_{j+1} := \frac{\beta^-}{h^2}(U_{j+2} + \lambda_{2,1}U_j - (2 - \lambda_{2,2})U_{j+1}) = F_{j+1} + C_{j+1},$$

(7)

where

$$C_j = -\frac{\beta^+}{h^2}(\lambda_{1,3}w + \lambda_{1,4}v + \lambda_{1,5}f^{(1)} + \lambda_{1,6}f^{(2)}),$$

$$C_{j+1} = -\frac{\beta^-}{h^2}(\lambda_{2,3}w + \lambda_{2,4}v + \lambda_{2,5}f^{(1)} + \lambda_{2,6}f^{(2)}).$$

(8)

Note 1. When $\beta^+ = \beta^-$, we have $\lambda_{1,1} = 0$ and $\lambda_{1,2} = 1$. As a result, we have regular stencils at irregular points when β is continuous.

Note 2. The closed formulas in (6) are useful for convergence analysis. In practice, numerical solutions of the local systems are faster and more stable than closed formulas.

2.2. Multiple intersections

The single-intersection approach naturally extends to multiple-intersection cases. A typical case is shown in Fig. 3, where α_1 and α_2 are the only interface points between x_{j-1} and x_{j+1} and $x_j \leqslant \alpha_1 < \alpha_2 < x_{j+1}$. As in the single-intersection case, we use a piecewise quadratic polynomial

$$p(x) = \begin{cases} p_1(x) := a_1 x^2 + a_2 x + a_3 & x < \alpha_1, \\ p_2(x) := a_4 x^2 + a_5 x + a_6 & \alpha_1 < x < \alpha_2, \\ p_3(x) := a_7 x^2 + a_8 x + a_9 & x > \alpha_2 \end{cases}$$

(9)

to approximate the solution $u(x)$ near x_j. We assume that the following values are provided:

$$u(x_j) = U_j, \quad u(x_{j+1}) = U_{j+1},$$

$$[u](\alpha_i) = w_i, \quad [\beta u_x](\alpha_i) = v_i, \quad i = 1, 2,$$

$$Lu^+(\alpha_1) = f^{(1)}, \quad Lu^-\left(\frac{\alpha_1 + \alpha_2}{2}\right) = f^{(2)}, \quad Lu^+(\alpha_2) = f^{(3)}.$$

(10)

Substituting (9) into (10), a linear system for the unknown coefficients can be set as above, yielding approximate solution values

$$\widehat{U}_{j+1} \approx p_1(x_{j+1}) =: \lambda_{1,1}U_j + \lambda_{1,2}U_{j+1} + \lambda_{1,3}w_1 + \lambda_{1,4}w_2 + \lambda_{1,5}v_1 + \lambda_{1,6}v_2 + \lambda_{1,7}f^{(1)} + \lambda_{1,8}f^{(2)} + \lambda_{1,9}f^{(3)}$$

and

$$\widehat{U}_j \approx p_3(x_j) =: \lambda_{2,1}U_j + \lambda_{2,2}U_{j+1} + \lambda_{2,3}w_1 + \lambda_{2,4}w_2 + \lambda_{2,5}v_1 + \lambda_{2,6}v_2 + \lambda_{2,7}f^{(1)} + \lambda_{2,8}f^{(2)} + \lambda_{2,9}f^{(3)}.$$

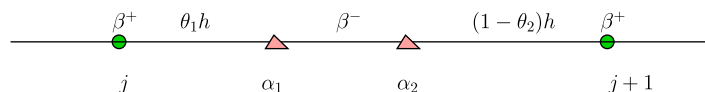Explicit formulas for $\lambda_{i,j}$ are provided in A.1. As a result, we have for the irregular points

Fig. 3. A double-intersection case: $\alpha_1 = x_j + \theta_1 h$, $\alpha_2 = x_j + \theta_2 h$.

$$L_h U_j := \frac{\beta^+}{h^2}(\lambda_{1,2}U_{j+1} + U_{j-1} - (2 - \lambda_{1,1})U_j) = F_j + C_j,$$

$$L_h U_{j+1} := \frac{\beta^+}{h^2}(U_{j+2} + \lambda_{2,1}U_j - (2 - \lambda_{2,2})U_{j+1}) = F_{j+1} + C_{j+1},$$

(11)

where

$$C_j = -\frac{\beta^+}{h^2}(\lambda_{1,3}w_1 + \lambda_{1,4}w_2 + \lambda_{1,5}v_1 + \lambda_{1,6}v_2 + \lambda_{1,7}f^{(1)} + \lambda_{1,8}f^{(2)} + \lambda_{1,9}f^{(3)}),$$

$$C_{j+1} = -\frac{\beta^+}{h^2}(\lambda_{2,3}w_1 + \lambda_{2,4}w_2 + \lambda_{2,5}v_1 + \lambda_{2,6}v_2 + \lambda_{2,7}f^{(1)} + \lambda_{2,8}f^{(2)} + \lambda_{2,9}f^{(3)}).$$

(12)

Note 3. When $\beta^+ = \beta^-$, we have $\lambda_{1,1} = 0$ and $\lambda_{1,2} = 1$ and thus a regular stencil at the irregular points when β is continuous.

Note 4. The sign conditions

$$\lambda_{1,5}, \lambda_{2,5} > 0, \quad \lambda_{1,6}, \lambda_{2,6} < 0$$

(13)

turn out to be the key property in the convergence proof below.

Note 5. For the coefficients $\beta(x)$, we assume the same constant β^+ for both $x < \alpha_1$ and $x > \alpha_2$. More generally, we can assume $\beta(x) = \beta_1$ for $x < \alpha_1$, $\beta(x) = \beta_2$ for $\alpha_1 < x < \alpha_2$ and β_3 for $x > \alpha_2$. A similar conclusion holds.

2.3. Why multiple intersections

We design this example to justify the consideration of multiple-intersection cases. It also arises in the solution of moving interface problems when interfaces collide. The coefficient is

$$\beta(x) = \begin{cases} 1.0 & \text{if } x \in \Omega^+, \\ 2.0 & \text{otherwise.} \end{cases}$$

The source term is

$$f(x) = \begin{cases} -18\sin(3x) & \text{if } x \in \Omega^+, \\ -2\cos(x) & \text{otherwise.} \end{cases}$$

We propose the homogeneous jump conditions

$$[u] = [\beta u_x] = 0.0$$

as well as Dirichlet boundary conditions $u(0) = 0$, $u(1) = 2\sin 3$. The phase function is $\phi(x) = (0.38 - x)(0.4 - x)$. Since an analytical solution is not available, we compare solutions on coarse grids with the solution on the finest grid ($N = 609$). Fig. 3 displays the relative positions of the interface and grid points ($j = 8$) in the case $N = 20$. We compare

(1) the PIM discretization at irregular points x_8 and x_9 and
(2) standard stencils at x_8 and x_9, ignoring the interface.

Fig. 4 shows a comparison of the numerical results obtained from both approaches when $N = 20$. The maximum error from the approach (2) is of order 10^{-1} while the PIM approach (1) reduces it to 10^{-3}. Therefore, interface points may cause jumps in both the solution and its derivatives, which affect the process of discretizing differential equations for high accuracy. Of course multiple intersections are even more unavoidable and problematic in two- or three-dimensional problems.
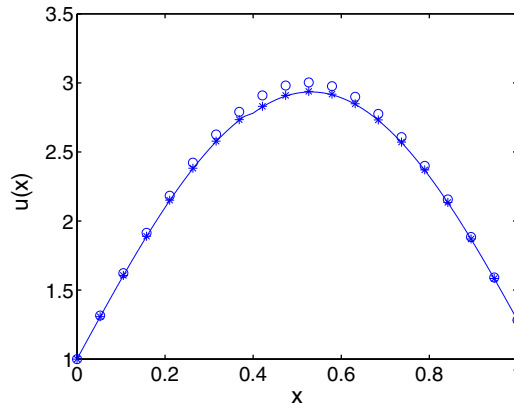
Fig. 4. A comparison of the computed solutions for $N = 20$. The solid line is the exact solution. The asterisk denotes the result of the PIM approach (1). The circle denotes the result of the approach (2).

2.4. Rank analysis

In this section we show that the local system determining the unknown stencil always has full rank, first for piecewise constant and then for variable coefficients.

2.4.1. Piecewise constant coefficients

Theorem 2.1. *Given two points $x_j < x_{j+1}$, and two arrays of points $\alpha_1, \ldots, \alpha_k$ and $\gamma_1, \ldots, \gamma_{k-1}$ satisfying $x_j \leqslant \alpha_1 < \gamma_1 < \alpha_2 < \gamma_2 < \cdots < \gamma_{k-1} < \alpha_k \leqslant x_{j+1}$ (see Fig. 5). Assume without loss of generality that $k \geqslant 2$. Define a piecewise quadratic polynomial $p(x)$ by*

$$p(x) = \begin{cases} p_1(x) & x < x_j, \\ p_2(x) & x_j < x < \alpha_1, \\ p_3(x) & \alpha_1 < x < \alpha_2, \\ \cdots \\ p_{k+1}(x) & x > x_{j+1}, \end{cases}$$

where $p_i(x)$ are quadratic polynomials. If $\beta(x)$ is piecewise constant, then $p(x)$ is uniquely determined by the following $3k + 3$ values:

$$p(x_j), \quad p(x_{j+1}),$$
$$[p](\alpha_i), \quad [\beta u_x](\alpha_i), \quad i = 1, \ldots, k,$$
$$Lp(x_j), \quad Lp(x_{j+1}), \quad Lp(\gamma_i), \quad i = 1, \ldots, k-1,$$
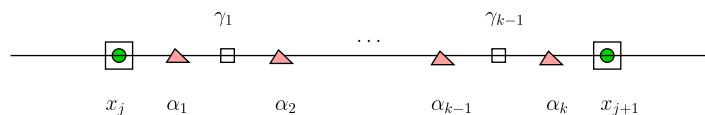
where L is the differential operator of (2).

Fig. 5. A general multiple-intersection case. We impose function values at x_j, x_{j+1} (circles), jump conditions at α_i (triangles) and partial differential equations at γ_i and x_j, x_{j+1} (squares).

Proof. It suffices to show that if the $3k + 3$ values provided are all zero, then $p(x) \equiv 0$. If

$$Lp(x_j) = 0, \quad Lp(x_{j+1}) = 0, \quad Lp(\gamma_i) = 0 \quad i = 1, \dots, k - 1,$$

then $p(x)$ must be piecewise linear since $\beta(x)$ is piecewise constant. The homogeneous jump conditions

$$[p](\alpha_i) = 0, \quad [\beta p_x](\alpha_i) = 0, \quad i = 1, \dots, k$$

show that $p(x)$ is actually continuous and each linear piece has the same sign of slope since $\beta(x) > 0$. Therefore $p(x)$ is monotonic. On the other hand we have $p(x_j) = p(x_{j+1}) = 0$. The conclusion follows immediately. $\square$

2.4.2. Variable coefficients

The same conclusion holds for variable-coefficient cases provided that the step size h is small enough. For simplicity we consider the single-intersection case (see Fig. 2).

Theorem 2.2. *Suppose that α is the only interface point between x_{j-1} and x_{j+1}, and $x_j \leqslant \alpha = x_j + \theta h < x_{j+1}$. If $\max\{|\beta_x^+|, |\beta_x^-|\} > 0$, then the local system determining unknown coefficients has full rank if*

$$h < \frac{\min\{\beta^- / \beta^+, \beta^+ / \beta^-\}}{\max\{|\beta_x^+|, |\beta_x^-|\}}, \tag{14}$$

where $\beta^{\pm}, \beta_x^{\pm}$ are evaluated at α.

Proof. Using the same notations in Section 2.1, we obtain a local linear system

$$Aa := \begin{bmatrix} x_j^2 & x_j & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{j+1}^2 & x_{j+1} & 1 \\ \alpha^2 & \alpha & 1 & -\alpha^2 & -\alpha & -1 \\ 2\beta^+\alpha & \beta^+ & 0 & -2\beta^-\alpha & -\beta^- & 0 \\ 2\beta^+ + 2\alpha\beta_x^+ & \beta_x^+ & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\beta^- + 2\alpha\beta_x^- & \beta_x^- & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} U_j \\ U_{j+1} \\ w \\ v \\ f^{(1)} \\ f^{(2)} \end{bmatrix} =: b.$$

Some calculations shows that the determinant of the matrix is

$$\det(A) = 2h\Big(2\beta^+\beta^-(\theta\beta^- + (1-\theta)\beta^+) - (1-\theta)^2\beta^+\beta^+\beta_x^- h + \theta^2\beta^-\beta^-\beta_x^+ h\Big)$$
$$\geqslant 2h\big(2\beta^+\beta^-(\theta\beta^- + (1-\theta)\beta^+) - (\beta^+\beta^+ + \beta^-\beta^-)\max\{|\beta_x^+|, |\beta_x^-|\}h\big).$$

A sufficient condition to ensure $\det(A) \neq 0$ is that

$$h < \frac{2\beta^+\beta^-(\theta\beta^- + (1-\theta)\beta^+)}{(\beta^+\beta^+ + \beta^-\beta^-)\max\{|\beta_x^+|, |\beta_x^-|\}}.$$

After simplification, it suffices to assume

$$h < \frac{\min\{\beta^- / \beta^+, \beta^+ / \beta^-\}}{\max\{|\beta_x^+|, |\beta_x^-|\}}. \quad \square$$

Note 6. If $\beta(x)$ is piecewise constant, then $\max\{|\beta_x^+|, |\beta_x^-|\} = 0$ and thus the inequality (14) is always satisfied.

2.5. Numerical examples for the 1D PIM

We present several examples to confirm the accuracy and robustness of the 1D PIM. Example 1 considers the case when $\beta(x)$ is highly oscillating. Example 2 explores the effect of interface locations on the global accuracy in solution.

2.5.1. Example 1

The exact solution is

$$u(x) = \begin{cases} e^x \cos(x) & \text{if } x \in \Omega^+, \\ 0 & \text{otherwise} \end{cases}$$

and the coefficient is

$$\beta(x) = \begin{cases} 1.0 + 0.6 \sin(kx) & \text{if } x \in \Omega^+, \\ 1 & \text{otherwise.} \end{cases}$$

The phase function is $\phi(x) = (0.4 - x)(0.404 - x)(0.6 - x)(0.604 - x)$. As a result, we have two pairs of clustered interface points on coarse grids, which are then well separated on fine grids. Table 1 shows the numerical results for $k = 20$ and 80. The convergence rate is very smooth and not affected by the transition from multiple intersections to single intersection as N increases.

For comparison purpose, we also present the numerical results for the same problem without interfaces, i.e. $\Omega^+ = (0, 1)$. The results are very similar. Therefore, the 1D PIM works well for highly oscillating coefficients in terms of both accuracy and convergence rates.

2.5.2. Example 2

The exact solution is

$$u(x) = \begin{cases} \sin(x) + 1 & \text{if } x \in \Omega^+, \\ \cos(x) - 1 & \text{otherwise} \end{cases}$$

and the coefficient is

$$\beta(x) = \begin{cases} 1.0 & \text{if } x \in \Omega^+, \\ 2.0 & \text{otherwise.} \end{cases}$$

We fix the grid to be $N = 40$. The phase function is $\phi(x) = (a - x)(b - x)$ where

$$a = x_{20} + (1 - d)h/2, \quad b = x_{20} + (1 + d)h/2$$

with d being an parameter. We vary d from 0.01 to 0.99

$$d = 0.01, 0.02, \ldots, 0.99,$$

so that the interface points move around inside the interval (x_{20}, x_{21}). Fig. 6 demonstrates uniform accuracy with respect to the location of interface points.

3. The piecewise-polynomial interface method in 2D

The 1D approach extends naturally to 2D. We assume the domain $\Omega = (0, 1) \times (0, 1)$ and cover the square with N grid points in each direction, so that

Table 1
Example 1: β is highly oscillating

N	$k = 20$				$k = 80$			
	$\|E_N\|_\infty$	Order	$\|e_N\|_\infty$	Order	$\|E_N\|_\infty$	Order	$\|e_N\|_\infty$	Order
20	3.361e−03		3.431e−03		5.549e−02		8.762e−02	
80	2.266e−04	1.9	1.914e−04	2.0	1.859e−03	2.4	2.412e−03	2.5
320	1.311e−05	2.0	1.172e−05	2.0	1.188e−04	2.0	1.444e−04	2.0
1280	7.428e−07	2.1	7.292e−07	2.0	7.186e−06	2.0	8.968e−06	2.0

Here E_N is the maximum error $\|u - U\|_\infty$ for the interface problem with a mesh size N while e_N is the maximum error for the problem $\Omega^+ = (0, 1)$, containing k wavelengths of the coefficient β.
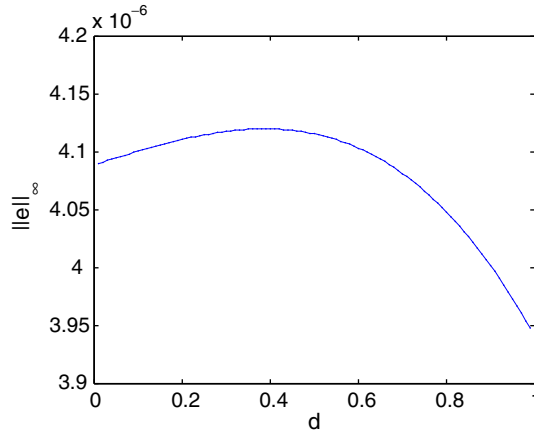
Fig. 6. The effect of interface location on global accuracy. Note the small scale in *y*-direction. Here $\|e\|_\infty$ is the maximum error $\|u - U\|_\infty$ and $N = 40$.

$$x_i = (i - 1)h, \quad y_j = (j - 1)h, \quad i, j = 1, \ldots, N,$$

where $h = 1/(N - 1)$. We say that (x_i, y_j) is a *regular point* if the interface does not separate any points in the standard five-point stencil centered at (x_i, y_j). Otherwise, we say that (x_i, y_j) is an *irregular point*.

Assume that at each interior point (x_i, y_j) with $1 < i, j < N$ the differential equation (1) is approximated by

$$L_h U_{i,j} := \sum_{(i_k, j_k) \in \mathbb{N}_{ij}} \gamma_k U_{i_k, j_k} = F_{ij} + C_{ij},$$

where $F_{ij} = f(x_i, y_j)$, C_{ij} is a correction term at (i, j) and $\mathbb{N}_{ij}$ is an index set of grid points neighboring (x_i, y_j). The standard five-point stencil at a regular point (x_i, y_j) has

$$\frac{1}{h}\left\{ \left(\beta_{i+1/2,j} \frac{U_{i+1,j} - U_{i,j}}{h} - \beta_{i-1/2,j} \frac{U_{i,j} - U_{i-1,j}}{h} \right) + \left(\beta_{i,j+1/2} \frac{U_{i,j+1} - U_{i,j}}{h} - \beta_{i,j-1/2} \frac{U_{i,j} - U_{i,j-1}}{h} \right) \right\} - \kappa_{i,j} U_{i,j}$$
$$= F_{ij},$$

where

$$\beta_{i,j} = \beta(x_i, y_j), \quad \beta_{i+1/2,j} = \beta\left(x_i + \frac{h}{2}, y_j\right)$$

and so on. Therefore, we have $C_{ij} = 0$ and

$$\mathbb{N}_{ij} = \{(i, j), (i + 1, j), (i - 1, j), (i, j + 1), (i, j - 1)\}.$$

At irregular points we use modified stencils described below. For simplicity of notation, we shall assume that $\beta(x, y)$ is piecewise constant and $\kappa \equiv 0$. The same approach works for general cases.

3.1. Single intersection

Consider the typical case shown in Fig. 7. We use a piecewise quadratic polynomial

$$p(x, y) = \begin{cases} p_1(x, y) = a_1 x^2 + a_2 xy + a_3 y^2 + a_4 x + a_5 y + a_6 & (x, y) \in \Omega^+, \\ p_2(x, y) = a_7 x^2 + a_8 xy + a_9 y^2 + a_{10} x + a_{11} y + a_{12} & (x, y) \in \Omega^- \end{cases} \tag{15}$$

to approximate the solution $u(x, y)$ near α_1. We assume that the following values are provided:

$$u(x_{i_k}, y_{j_k}) = U_{i_k, j_k}, \quad (i_k, j_k) \in \mathbb{N}_{ij},$$
$$[u](\alpha_i) = w_i, \quad [\beta u_n](\alpha_i) = v_i, \quad i = 1, 2, 3, \tag{16}$$
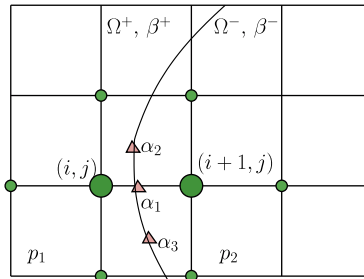$$Lu^+(\alpha_1) = f^{(1)}, \quad Lu^-(\alpha_1) = f^{(2)},$$

Fig. 7. A single-intersection case: α_1 is the only intersection point between (i, j) and $(i + 1, j)$. Here α_2 is an extra interface point nearby.

where $\mathbb{N}_{ij}$ is the index set of 8 grid points marked with circles in Fig. 7. As a result, we can substitute (15) into (16), set up a 16×12 linear system for the unknown coefficients and solve them via least squares, yielding approximate solution values

$$\widehat{U}_{i+1,j} \approx p_1(x_{i+1}, y_j), \quad \widehat{U}_{i,j} \approx p_2(x_i, y_j).$$

The modified stencils for (i, j) and $(i + 1, j)$ are

$$((\beta u_x)_x + (\beta u_y)_y)(x_i, y_j) \approx \frac{\beta^+}{h^2}(\widehat{U}_{i+1,j} + U_{i,j+1} + U_{i-1,j} + U_{i,j-1} - 4U_{i,j}),$$

$$((\beta u_x)_x + (\beta u_y)_y)(x_{i+1}, y_j) \approx \frac{\beta^-}{h^2}(U_{i+2,j} + U_{i+1,j+1} + \widehat{U}_{i-1,j} + U_{i+1,j-1} - 4U_{i+1,j}).$$

The rest follows from the 1D PIM approach.

Note 7. We could select fewer neighbor points, making the local system exactly determined. However, the local system might be singular if it is square (see Section 3.5 for an example).

Note 8. It is a problem-dependent question about how many neighbor points to select. For complex interfaces more aggression is appropriate but the standard nine-point stencil usually suffices.

3.2. Multiple intersections

When representing complex interfaces on coarse grids, more than one interface points may separate neighboring grid points. Consider the typical example shown in Fig. 8. Even though (x_i, y_j) and (x_i, y_{j+1}) are both in Ω^+, we should expect some discontinuity in the solution or its derivatives. Thus the standard stencil cannot be applied directly at these two irregular points. In the spirit of the 1D PIM, we propose a piecewise quadratic polynomial
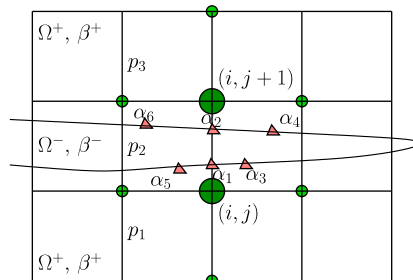

Fig. 8. Double-intersection case: α_1, α_2 are the interface points between (i, j) and $(i, j + 1)$. α_3 is an extra interface point near α_1 and α_4 is another interface point near α_2.

$$p(x,y) = \begin{cases} p_1(x,y) & (x,y) \in \Omega^+, \quad y \leqslant y_{\alpha_1}, \\ p_2(x,y) & (x,y) \in \Omega^-, \\ p_3(x,y) & (x,y) \in \Omega^+, \quad y \geqslant y_{\alpha_2} \end{cases} \tag{17}$$

to approximate the solution $u(x,y)$ near α_1, α_2 where $(x_{\alpha_i}, y_{\alpha_i})$ are the coordinates of α_i, $i = 1, 2$. We assume that the following values are provided:

$$\begin{aligned} u(x_{i_k}, y_{j_k}) &= U_{i_k, j_k}, \quad (i_k, j_k) \in \mathbb{N}_{ij}, \\ [u](\alpha_i) &= w_i, \quad [\beta u_n](\alpha_i) = v_i, \quad i = 1, \ldots, 6, \\ Lu^+(\alpha_1) &= f^{(1)}, \quad Lu^-(\alpha_{1/2}) = f^{(1/2)}, \quad Lu^+(\alpha_2) = f^{(2)}, \end{aligned} \tag{18}$$

where $\alpha_{1/2}$ is the middle point between α_1 and α_2 and $\mathbb{N}_{ij}$ is the index set of eight points marked with circles in Fig. 8. As a result, we can substitute (17) into (18), set up a 23×18 linear system for the unknown coefficients and solve for them via least squares, yielding approximate solution values

$$\widehat{U}_{i,j+1} \approx p_1(x_i, y_{j+1}), \quad \widehat{U}_{i,j} \approx p_3(x_i, y_j).$$

The modified stencils for (i, j) and $(i, j+1)$ are then

$$((\beta u_x)_x + (\beta u_y)_y)(x_i, y_j) \approx \frac{\beta^+}{h^2}(U_{i,j} + \widehat{U}_{i,j+1} + U_{i-1,j} + U_{i,j-1} - 4U_{i,j}),$$

$$((\beta u_x)_x + (\beta u_y)_y)(x_i, y_{j+1}) \approx \frac{\beta^+}{h^2}(U_{i+1,j+1} + U_{i,j+2} + U_{i-1,j+1} + \widehat{U}_{i,j-1} - 4U_{i,j+1}).$$

Note 9. Numerical experiments show that if we propose quadratic polynomials for p_1, p_3 but linear polynomial for p_2, only first-order convergence rates can be achieved.

3.3. Hard cases

Extremely complex interfaces on coarse grids may involve additional complications. Fig. 9 shows two cases.

(1) In Fig. 9a, there are enough neighbor points for $(i+1, j)$ but none for (i, j). As a result, the standard approach in Section 3.1 works but produces inaccurate representations.
(2) In Fig. 9b, there are too few neighbor points for (i, j) and $(i+1, j)$ and thus the standard approach will produce an under-determined local system.

To resolve these hard cases, we look beyond (i, j) and $(i+1, j)$ and proposes more polynomials as necessary. Fig. 9a and b shows the resulting stencils. We follow two criteria in designing new polynomial(s):
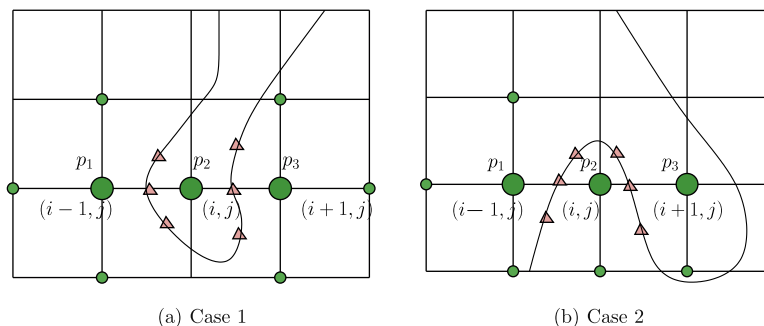

(a) Case 1   (b) Case 2

Fig. 9. Examples for hard cases.

- The search starts from a neighbor of the irregular point. In Fig. 9, $(i-1,j)$ is a neighbor of (i,j).
- The new polynomial should have enough supporting grid points. In Fig. 9, $(i-1,j)$ is thus a good candidate.

In almost all of our numerical examples, only one additional polynomial is needed.

3.4. Order of accuracy

One difference between the 2D PIM and the 1D version is that we use jump conditions at interface points off grid lines. The following analysis justifies the approach for smooth interfaces or fine grids.

Lemma 3.1. *Let $p(x,y)$ be a 2D quadratic polynomial satisfying the following six conditions*:

$$p(x_i, y_i) = p_i, \quad \frac{\partial p}{\partial \vec{n}}(x_i, y_i) = q_i, \quad i = 1, 2, 3, \tag{19}$$

where $\vec{n}$ is an arbitrary unit vector. If the points $\{(x_i, y_i)\}_{i=1}^3$ are not collinear and $\vec{n}$ is not parallel to any segment connecting two of the points, then the conditions (19) uniquely determine $p(x,y)$.

Proof. Let $p(x,y) = a_1 x^2 + a_2 xy + a_3 y^2 + a_4 x + a_5 y + a_6$. We assume without loss of generality that

$$\{(x_i, y_i)\} = \{(0,0), (1,r), (s,t)\}, \quad \vec{n} = (1,0),$$

where r, s, t are arbitrary real numbers. From conditions (19) we can set up the system for the coefficients $\{a_i\}$ as:

$$Aa := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & r & r^2 & 1 & r & 1 \\ 2 & r & 0 & 1 & 0 & 0 \\ s^2 & st & t^2 & s & t & 1 \\ 2s & t & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} p_1 \\ q_1 \\ p_2 \\ q_2 \\ p_3 \\ q_3 \end{bmatrix} =: b.$$

Then $p(x,y)$ is uniquely determined iff the matrix A is non-singular. A straightforward calculation shows that

$$\det(A) = 2rt(r-t)(t-rs).$$

If $rt = 0$ or $t = r$ then at least one of the segments connecting two of the points has the same direction as $\vec{n}$, while if $t = rs$ then three points are collinear. This completes the proof. $\square$

Note 10. By a perturbation analysis, we can show that the lemma still holds if conditions (19) are replaced by

$$p(x_i, y_i) = p_i, \quad \frac{\partial p}{\partial \vec{n}_i}(x_i, y_i) = q_i, \quad i = 1, 2, 3,$$

provided that $\max_{i,j}\{|\vec{n}_i - \vec{n}_j|\} \leqslant O(h)$.

Note 11. For the case shown in Fig. 7, Lemma 3.1 shows that the quadratic polynomial $p1 - p2$ is uniquely determined everywhere from the six jump conditions. Therefore if the number of grid points collected around (i,j) and $(i+1,j)$ is greater than or equal to six and the points are in general position, then the system (16) has full rank. The same conclusion holds for general cases.

Theorem 3.2. *If all the points involved in Eq. (16) satisfy*

$$|x_i - x_j| = O(h) \tag{20}$$

then the PIM local approximation has an accuracy of $O(h^3)$.

Proof. For simplicity, we consider the single-intersection case in Fig. 7. First we set up a linear system for the unknown coefficient as in (5), where the right-hand side contains either $u^\pm$ or its derivatives. The full column rank of this system is guaranteed by Lemma 3.1. Taylor expansion at α_1 gives

$$\widehat{U}_{i+1,j} - u_{i+1,j} = p_1(x_{i+1}, y_j) - u_{i+1,j} = \sum_k h^k \sum_{p+q=k} (C_{pq}^+ \partial_x^p \partial_y^q u^+ + C_{pq}^- \partial_x^p \partial_y^q u^-).$$

From dimensional analysis the coefficients $C_{pq}^\pm$ depends on $\beta^\pm$ and the relative positions of x_i and α_i, but not on $u^\pm$ or h due to (20). On the other hand, this approximation is exact if the true solution is indeed a piecewise quadratic polynomial. Therefore, $C_{pq}^\pm = 0$ for $p + q \leqslant 2$ and so

$$\widehat{U}_{i+1,j} - u_{i+1,j} = \mathrm{O}(h^3). \quad \square$$

Note 12. The same argument can be applied to multiple-intersection cases and thus $\mathrm{O}(h^3)$ accuracy is always achieved. Therefore, we have local truncation error of $\mathrm{O}(h)$ at irregular points and thus globally second-order accuracy is expected for computed solutions.

3.5. Why least squares: a singular case

Consider the typical situation shown in Fig. 10 where we take a naive approach of imposing one condition per unknown to get a square linear system:

$$u(x_{i_k}, y_{j_k}) = U_{i_k, j_k}, \quad (i_k, j_k) \in \mathbb{N}_{ij},$$
$$[u](\alpha) = w, \quad [\beta u_n](\alpha) = v,$$
$$Lu^+(\alpha) = f^{(1)}, \quad Lu^-(\alpha) = f^{(2)}.$$

However, the local 12×12 system has only rank 11 whenever α is on the same line as (i, j) and $(i + 1, j)$!

Alternatively we can replace α by some other interface point $\hat{\alpha}$ where $\hat{\alpha}$ is not on the same line as (i, j) and $(i + 1, j)$. Then the local system has full rank. But solving for the unknown coefficients gives

$$\widehat{U}_{i+1,j} = 4U_{i,j} - U_{i-1,j} - U_{i,j+1} - U_{i,j-1} + C,$$

where C is some constant. The global matrix then has the corresponding row for the point (i, j) all zeros!

Least squares provide the flexibility of adding more information appropriately to effectively remove these difficulties with minimal extra effort.

4. Convergence analysis for the 1D PIM

4.1. Single intersection

We assume that α is the only interface point between x_{j-1} and x_{j+1} with $x_j \leqslant \alpha = x_j + \theta h < x_{j+1}$ (see Fig. 2). Define
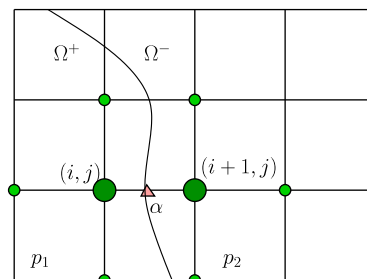

Fig. 10. A singular case.

$$\hat{\beta} := \theta\beta^- + (1-\theta)\beta^+, \quad \beta_{\max} := \max\{\beta^+, \beta^-\}, \quad \beta_{\min} := \min\{\beta^+, \beta^-\},$$

$$M_{xxx} := \max\{\max_{x<\alpha} |u'''(x)|, \max_{x>\alpha} |u'''(x)|\}, \quad M_{xxxx} := \max\{\max_{x<\alpha} |u''''(x)|, \max_{x>\alpha} |u''''(x)|\}.$$

4.1.1. Local truncation error

We define the truncation error at each interior point by the usual formula:

$$T_i := L_h u_i - f_i - C_i,$$

where $u_i = u(x_i)$. At regular points, Taylor expansion shows $|T_i| \leqslant O(h^2)$. At an irregular point x_j, we have

$$\widehat{U}_{j+1} = \lambda_{1,1} U_j + \lambda_{1,2} U_{j+1} + \lambda_{1,3} w + \lambda_{1,4} v + \lambda_{1,5} f^{(1)} + \lambda_{1,6} f^{(2)}.$$

As a result, the local truncation error

$$T_j = \frac{\beta^+ u_{j-1} - 2\beta^+ u_j + \beta^+ \hat{u}_{j+1}}{h^2} - f_j.$$

Plugging in all the expressions (6) and simplifying gives

$$T_j = \frac{\beta^+(1-\theta)(\theta^2 D_1 - \theta D_2 + D_3)h}{\hat{\beta}} + O(h^2),$$

where

$$D_1 = -\frac{\beta^+}{2} u_{xxx}^+ + \frac{\beta^-}{3} u_{xxx}^+ + \frac{\beta^-}{6} u_{xxx}^-,$$

$$D_2 = \frac{\beta^+}{2} u_{xxx}^+ - \frac{\beta^-}{6} u_{xxx}^+ - \frac{\beta^-}{3} u_{xxx}^-,$$

$$D_3 = -\frac{\beta^+}{6} u_{xxx}^+ + \frac{\beta^+}{6} u_{xxx}^-$$

and the derivatives are evaluated at α. An upper bound follows:

$$|T_j| \leqslant \frac{\beta^+(1-\theta)(3\theta^2 + 3\theta + 1)\beta_{\max} M_{xxx}}{3\beta_{\min}} h \leqslant \frac{7\beta_{\max}^2 M_{xxx}}{3\beta_{\min}} h. \tag{21}$$

The same bound can be derived for T_{j+1}.

4.1.2. Stability analysis

We prove a discrete maximum principle in the following lemma.

Lemma 4.1. *If $\beta(x)$ is piecewise constant, then the coefficients are bounded above and below by*

$$\frac{\beta_{\min}}{h^2} \leqslant |\gamma_{i,k}| \leqslant \frac{2\beta_{\max}}{h^2}, \quad 1 \leqslant i \leqslant n-1, \quad k = 1,2,3$$

and a discrete maximum principle is satisfied:

$$\gamma_{i,1} > 0, \quad \gamma_{i,3} > 0, \quad \gamma_{i,2} < 0, \quad \gamma_{i,1} + \gamma_{i,3} \leqslant |\gamma_{i,2}|, \quad 1 \leqslant i \leqslant n-1.$$

Proof. The conclusions are obvious for regular points. At irregular points x_j, x_{j+1} with $x_j \leqslant \alpha < x_{j+1}$, (7) and (6) imply

$$\gamma_{j,1} = \frac{\beta^+}{h^2}, \quad \gamma_{j,2} = -\frac{\beta^+(1 + \beta^-/\hat{\beta})}{h^2}, \quad \gamma_{j,3} = \frac{\beta^+ \beta^-/\hat{\beta}}{h^2},$$

$$\gamma_{j+1,1} = \frac{\beta^- \beta^+/\hat{\beta}}{h^2}, \quad \gamma_{j+1,2} = -\frac{\beta^-(1 + \beta^+/\hat{\beta})}{h^2}, \quad \gamma_{j+1,3} = \frac{\beta^-}{h^2}.$$

To prove the bounds, we distinguish two different cases. If $\beta^- \leqslant \beta^+$,

$$\frac{\beta^+ \beta^-}{\hat{\beta}} = \frac{\beta^+ \beta^-}{\theta\beta^- + (1-\theta)\beta^+} \leqslant \frac{\beta^+ \beta^-}{\theta\beta^- + (1-\theta)\beta^-} = \beta^+,$$

$$\frac{\beta^+ \beta^-}{\hat{\beta}} = \frac{\beta^+ \beta^-}{\theta\beta^- + (1-\theta)\beta^+} \geqslant \frac{\beta^+ \beta^-}{\theta\beta^+ + (1-\theta)\beta^+} = \beta^-.$$

If on the other hand $\beta^- > \beta^+$,

$$\frac{\beta^+ \beta^-}{\hat{\beta}} = \frac{\beta^+ \beta^-}{\theta\beta^- + (1-\theta)\beta^+} > \frac{\beta^+ \beta^-}{\theta\beta^- + (1-\theta)\beta^-} = \beta^+,$$

$$\frac{\beta^+ \beta^-}{\hat{\beta}} = \frac{\beta^+ \beta^-}{\theta\beta^- + (1-\theta)\beta^+} < \frac{\beta^+ \beta^-}{\theta\beta^+ + (1-\theta)\beta^+} = \beta^-.$$

Thus $\beta_{\min} \leqslant \beta^+ \beta^- / \hat{\beta} \leqslant \beta_{\max}$. The conclusion follows from (7) via simplification. $\quad\square$

4.1.3. Convergence proof

We start with the following lemma, which generalizes Theorems 6.1 and 6.2 of [17].

Lemma 4.2. *Given a difference scheme L_h defined on a discrete set of interior points J_Ω, we assume the following conditions hold*:

1. *J_Ω can be partitioned into a number of disjoint sets*

   $$J_\Omega = J_1 \cup J_2 \cup \cdots J_s, \quad J_i \cap J_k = \varnothing \quad \text{if } i \neq k.$$

2. *The truncation error of the difference scheme at a grid point p satisfies*

   $$|T_p| \leqslant T_i, \quad \forall p \in J_i, \quad i = 1, 2, \ldots, s.$$

3. *There exits a non-negative mesh function Φ defined on $J_\Omega \cup J_{\partial\Omega}$ satisfying*

   $$L_h \Phi_p \geqslant K_i > 0, \quad \forall p \in J_i, \quad i = 1, 2, \ldots, s.$$

Then the global error of the approximate solution from the difference scheme at mesh points is bounded by

$$||e_i||_\infty \leqslant \left(\max_{A \in J_{\partial\Omega}} \Phi_A \right) \max_{1 \leqslant i \leqslant s} \left\{ \frac{T_i}{K_i} \right\},$$

where $e_i = u_i - U_i$ and $J_{\partial\Omega}$ are the boundary points.

To prove the major theorem for convergence, the key is to construct the following comparison function:

$$\Phi(x) = \begin{cases} \frac{(x-\alpha)^2}{\beta^+} + \frac{(1-\alpha)(\alpha-x)\hat{\beta}}{\beta^+} + \frac{(\alpha-x_j)(x_j-x)}{h} & \text{if } x \leqslant \alpha, \\ \frac{(x-\alpha)^2}{\beta^-} + \frac{\alpha(x-\alpha)\hat{\beta}}{\beta^-} + \frac{(x_{j+1}-\alpha)(x-x_{j+1})}{h} & \text{if } x > \alpha. \end{cases} \tag{22}$$

The function $\Phi(x)$ is non-negative at mesh points and the last two terms in each expression above correspond to the Green function.

We apply the difference scheme (7) to the comparison function at j and $j + 1$:

$$L_h \Phi_j = \frac{\beta^+}{\hat{\beta}}(2 + \theta(\rho_1 - 1) + \theta^2(\rho_1 - 1)) + \frac{\beta^+(1-\theta)}{h} + \frac{\beta^+ \theta}{h},$$

$$L_h \Phi_{j+1} = \frac{\beta^-}{\hat{\beta}}(2\rho_2 + 3\theta(1 - \rho_2) + \theta^2(\rho_2 - 1)) + \frac{\beta^- \theta}{h} + \frac{\beta^-(1-\theta)}{h}$$

with $\rho_1 := \beta^-/\beta^+$, $\rho_2 := \beta^+/\beta^-$. We are now ready to provide the error estimate in the following theorem.

Theorem 4.3. *Assume $\beta(x)$ is piecewise constant, then the error of the approximate solution obtained from the PIM is bounded by*

$$\|u_i - U_i\|_{-\infty} \leqslant C(\beta^+, \beta^-, M_{xxx}, M_{xxxx})h^2.$$

Proof. In the notation of Lemma 4.2,

$$J_1 = \{1, 2, \ldots, j-1\}, \quad J_2 = \{j+2, \ldots, n-1\}, \quad J_3 = \{j, j+1\}, \quad J_{\partial\Omega} = \{0, n\}.$$

In regions J_1 and J_2 the standard difference equation is used: $L_h\Phi_i = 4$. Thus

$$T_i \leqslant \frac{\beta_{\max}M_{xxxx}}{12}h^2 \quad \text{and} \quad \frac{T_i}{K_i} \leqslant \frac{\beta_{\max}M_{xxxx}}{24}h^2, \quad i = 1, 2$$

for $K_1 = K_2 = 2$.

It's straightforward to obtain the following inequalities:

$$\frac{\beta^+}{\hat{\beta}}(2 + \theta(\rho_1 - 1) + \theta^2(\rho_1 - 1)) \geqslant \frac{\beta^+}{\hat{\beta}}(2 - \theta - \theta^2) \geqslant 0,$$

$$\frac{\beta^-}{\hat{\beta}}(2\rho_2 + 3\theta(1 - \rho_2) + \theta^2(\rho_2 - 1)) \geqslant \frac{\beta^-}{\hat{\beta}}\min\{2, 2\rho_2\} \geqslant 0.$$

Hence in region J_3

$$L_h\Phi_j \geqslant \frac{\beta^+(1 - \theta)}{h} + \frac{\beta^+\theta}{h} = \frac{\beta^+}{h}, \quad L_h\Phi_{j+1} \geqslant \frac{\beta^-\theta}{h} + \frac{\beta^-(1 - \theta)}{h} = \frac{\beta^-}{h}.$$

If we take $K_3 = \beta_{\min}/h$ and combine the local truncation error estimate (21), then

$$\frac{T_i}{K_i} \leqslant \frac{7\beta_{\max}^2 M_{xxx}}{3\beta_{\min}^2}h^2, \quad i = j, j+1.$$

Finally we estimate $\max_{A \in J_{\partial\Omega}} \Phi_A = \max\{\Phi(0), \Phi(1)\}$. From the definition of $\Phi(x)$, we have

$$\Phi(0) = \frac{\alpha^2}{\beta^+} + \frac{\alpha(1 - \alpha)\hat{\beta}}{\beta^+} + \frac{(\alpha - x_j)x_j}{h}, = \frac{\alpha^2}{\beta^+} + \frac{\alpha(1 - \alpha)\hat{\beta}}{\beta^+} + \theta x_j.$$

Similarly, we have

$$\Phi(1) = \frac{(1 - \alpha)^2}{\beta^-} + \frac{\alpha(1 - \alpha)\hat{\beta}}{\beta^-} + (1 - \theta)(1 - x_{j+1}).$$

Since these are both of order O(1), the conclusion follows from Lemmas 4.1 and 4.2. $\quad\square$

4.2. Multiple intersections

We employ a more general approach for multiple-intersection cases by using an implicit construction of comparison functions. Consider the typical case shown in Fig. 3 and define

$$\theta_1 := \frac{\alpha_1 - x_j}{h}, \quad \theta_2 := \frac{\alpha_2 - x_j}{h}, \quad \hat{\beta} := (1 + \theta_1 - \theta_2)\beta^- + (-\theta_1 + \theta_2)\beta^+.$$

Based on the formulas in A.1, we can show that Lemma 4.1 also applies to the double-intersection case. From (11) and (12), the local truncation error at irregular points is bounded by

$$|T_i| \leqslant C(\beta^+, \beta^-, M_{xxx})h, \quad i = j, j+1. \tag{23}$$

A detailed calculation is tedious and avoided here. We are now ready to provide the error estimate in the following theorem.

Theorem 4.4. *Assume $\beta(x)$ is piecewise constant, then the error of the approximate solution obtained from the* 1D *PIM is bounded by*

$$\|u(x_i) - u_h(x_i)\|_\infty \leqslant C(\alpha_1, \alpha_2, \beta^+, \beta^-, M_{xxx}, M_{xxxx})h^2. \tag{24}$$

Proof. The solution to the following interface problem:

$$\begin{aligned}
(\beta\phi_x)_x &= 1, \quad \phi|_{\partial\Omega} = 1, \\
[\phi](\alpha_1) &= 0, \quad [\phi](\alpha_2) = 0, \\
[\beta\phi_x](\alpha_1) &= -1, \quad [\beta\phi_x](\alpha_2) = 1
\end{aligned} \tag{25}$$

exits, and is unique, piecewise smooth, and bounded [6]. Let

$$\Phi(x) := \phi(x) + |\min_{x \in \Omega} \phi(x)| \geqslant 0.$$

Then $\Phi(x)$ also satisfies the differential equations and jump conditions in (25). In the notation of Lemma 4.2, let

$$J_1 = \{1, 2, \ldots, j-1\}, \quad J_2 = \{j+2, \ldots, n-1\}, \quad J_3 = \{j, j+1\}, \quad J_{\partial\Omega} = \{0, n\}.$$

In regions J_1 and J_2 the standard difference equation is used: $L_h\Phi_i = 1$. Thus

$$T_i \leqslant \frac{\beta_{\max}M_{xxxx}}{12}h^2, \quad \text{and} \quad \frac{T_i}{K_i} \leqslant \frac{\beta_{\max}M_{xxxx}}{12}h^2, \quad i = 1, 2$$

for $K_1 = K_2 = 1$.

At the irregular point j, the relation $T_j = L_h\Phi_j - C_j - f_j$ ensures that

$$L_h\Phi_j = T_j + C_j + f_j,$$

where $T_j = \mathrm{O}(h)$ and

$$C_j = -\frac{\beta^+}{h^2}(\lambda_{1,3}w_1 + \lambda_{1,4}w_2 + \lambda_{1,5}v_1 + \lambda_{1,6}v_2 + \lambda_{1,7}f^{(1)} + \lambda_{1,8}f^{(2)} + \lambda_{1,9}f^{(3)}) = \frac{\beta^+}{h^2}(\lambda_{1,5} - \lambda_{1,6}) + \mathrm{O}(1).$$

Hence ignoring higher-order terms for simplicity,

$$L_h\Phi_j = \frac{\beta^+}{h^2}(\lambda_{1,5} - \lambda_{1,6}) + \mathrm{O}(1) + \mathrm{O}(h) \approx \frac{2(1-\theta_2)\beta^- + (\theta_2 - \theta_1)\beta^+}{\hat{\beta}h} =: \frac{C}{h},$$

where C is some generic positive constant depending on $\beta^+, \beta^-, \theta_1, \theta_2$ but not h. Similarly, using the sign property in (13) we can show that $L_h\Phi_{j+1} \geqslant C/h$. Combining (23), we have $T_i/K_i \leqslant C/h^2, i = j, j+1$. Finally since both $\Phi(0)$ and $\Phi(1)$ are of order $\mathrm{O}(1)$, the conclusion follows from Lemma 4.2. $\square$

Note 13. Compared with the previous theorem, the relative location of the interface θ_1, θ_2 enters the generic constant in (24). This can be explained into two ways:

- In general, the solution $u(x)$ depends on the location of interface points and so do its derivatives. In this sense, θ enters implicitly into the generic constant in the previous theorem.
- Typically, the convergence rate for interface problems is experimentally not constant and depends on the relative positions. Thus we expect to see θ's in the generic constant.

4.3. Variable coefficients

We briefly discuss the case of variable coefficients. Following the same approach as in Section 2.1, we have approximations of the form

$$\begin{aligned}
\widehat{U}_{j+1} &\approx \lambda_{1,1}U_j + \lambda_{1,2}U_{j+1} + \lambda_{1,3}w + \lambda_{1,4}v + \lambda_{1,5}f^{(1)} + \lambda_{1,6}f^{(2)}, \\
\widehat{U}_j &\approx \lambda_{2,1}U_j + \lambda_{2,2}U_{j+1} + \lambda_{2,3}w + \lambda_{2,4}v + \lambda_{2,5}f^{(1)} + \lambda_{2,6}f^{(2)}
\end{aligned}$$

with

$$\lambda_{1,1} = 1 - \lambda_{1,2},$$

$$\lambda_{1,2} = \frac{\beta^- \beta^- (2\beta^+ + (-1 + 2\theta)\beta_x^+ h)}{\hat{\beta}},$$

$$\lambda_{1,4} = \frac{1 - \theta}{2}(2\beta^- - (1 - \theta)\beta_x^- h)(2\beta^+ - (1 - 2\theta)\beta_x^+ h)h,$$

$$\lambda_{2,1} = \frac{\beta^+ \beta^+ (2\beta^- + \beta_x^- (-s1 + 2\theta)h)}{\hat{\beta}},$$

$$\lambda_{2,2} = 1 - \lambda_{2,1},$$

$$\lambda_{2,4} = \frac{\theta h}{2}(2\beta^- - (1 - 2\theta)\beta_x^- h)(2\beta^+ + \theta\beta_x^+ h),$$

where

$$\hat{\beta} := 2\beta^+ \beta^- (\theta\beta^- + (1 - \theta)\beta^+) - (1 - \theta)^2 \beta^+ \beta^+ \beta_x^- h + \theta^2 \beta^- \beta^- \beta_x^+ h.$$

Only a partial list for $\lambda_{i,j}$ is provided for simplicity, which is to be used in the convergence proof. Here $\beta^\pm$ and its derivatives are evaluated at α. Thus at the irregular points

$$L_h U_j = \frac{1}{h^2}(\beta_{j+1/2}^+ \lambda_{1,2} U_{j+1} + \beta_{j-1/2}^+ U_{j-1} - ((1 - \lambda_{1,1})\beta_{j+1/2}^+ + \beta_{j-1/2}^+)U_j),$$

$$L_h U_{j+1} = \frac{1}{h^2}(\beta_{j+3/2}^- U_{j+2} + \beta_{j+1/2}^- \lambda_{2,1} U_j - ((1 - \lambda_{2,2})\beta_{j+1/2}^- + \beta_{j+3/2}^-)U_{j+1})$$

with

$$C_j = -\frac{\beta_{j+1/2}^+}{h^2}(\lambda_{1,3} w + \lambda_{1,4} v) + \lambda_{1,5} f^{(1)} + \lambda_{1,6} f^{(2)},$$

$$C_{j+1} = -\frac{\beta_{j+1/2}^-}{h^2}(\lambda_{2,3} w + \lambda_{2,4} v + \lambda_{2,5} f^{(1)} + \lambda_{2,6} f^{(2)}).$$

We summarize the result in the following theorem:

Theorem 4.5. *If there exists a constant $0 < C < 1$ with*

$$h \leqslant 2C \frac{\min_x \beta(x)}{\max_x |\beta_x(x)|},$$

then the 1D *PIM satisfies the discrete maximum principle and second-order convergence is achieved.*

Proof. With the upper bound for h, the $\lambda_{i,j}$ satisfies the same sign property as for the constant-coefficient case in Section 2.1. Thus the same approach as before yields the conclusion. □

4.4. Asymptotic error estimates

We extend the approach of [11,17] to provide estimates which show more precisely how the $O(h^2)$ error behaves in the limit as the mesh size h tends to zero.

For simplicity of notation, we consider the case when there is only one interface point α and $\beta^+ = \beta^- = 1$ (see Fig. 2). The extension to general cases is straightforward. Using the same notation as before, we know from previous calculations that the error satisfies

$$L_h e_i = T_i, \tag{26}$$

where the truncation error

$$T_i = \begin{cases} \frac{h^2}{12} u_{xxxx} + O(h^4) & i \neq j, \quad j+1, \\ -\frac{[u_{xxx}]}{3!}(1-\theta)^3 h + O(h^2) & i = j, \\ -\frac{[u_{xxx}]}{3!}\theta^3 h + O(h^2) & i = j+1. \end{cases}$$

We now define $\psi^{(1)}$ to be the solution of the problem

$$\psi_{xx}^{(1)} = \frac{1}{12} u_{xxxx},$$
$$\psi^{(1)}(0) = \psi^{(1)}(1) = 0,$$
$$[\psi^{(1)}](\alpha) = w_1, \quad [\psi_x^{(1)}](\alpha) = v_1$$

and $\psi^{(2)}$ to be the solution of the problem

$$\psi_{xx}^{(2)} = 0,$$
$$\psi^{(2)}(0) = \psi^{(2)}(1) = 0,$$
$$[\psi^{(2)}](\alpha) = w_2, \quad [\psi_x^{(2)}](\alpha) = v_2,$$

where w_1, v_1, w_2, v_2 are constants to be determined later. Then $\psi := \psi^{(1)} + h\psi^{(2)}$ satisfies

$$\psi_{xx} = \frac{1}{12} u_{xxxx}. \tag{27}$$

From (6)

$$L_h \psi_j^{(1)} = C_j + f_j + T_j = -\frac{1}{h^2}(\lambda_{1,3} w_1 + \lambda_{1,4} v_1 + \lambda_{1,5} f^{(1)} + \lambda_{1,6} f^{(2)}) + O(1) = -\frac{1}{h^2} w_1 - \frac{1-\theta}{h} v_1 + O(1).$$

Similarly, we have

$$L_h \psi_j^{(2)} = -\frac{1}{h^2} w_2 - \frac{1-\theta}{h} v_2 + O(1).$$

Therefore,

$$L_h \psi_j = L_h \psi_j^{(1)} + h L_h \psi_j^{(2)} = -\frac{1}{h^2} w_1 - \frac{(1-\theta)v_1 + w_2}{h} + O(1). \tag{28}$$

Repeating the steps at $j+1$ gives

$$L_h \psi_{j+1} = L_h \psi_{j+1}^{(1)} + h L_h \psi_{j+1}^{(2)} = \frac{1}{h^2} w_1 - \frac{\theta v_1 - w_2}{h} + O(1). \tag{29}$$

If we define w_1, v_1, w_2 by

$$w_1 = 0,$$
$$(1-\theta)v_1 + w_2 = \frac{[u_{xxx}]}{3!}(1-\theta)^3, \tag{30}$$
$$\theta v_1 - w_2 = \frac{[u_{xxx}]}{3!}\theta^3,$$

then

$$w_1 = 0, \quad v_1 = \frac{[u_{xxx}]}{3!}((1-\theta)^3 + \theta^3). \tag{31}$$

Combining (26)–(30) gives

$$\left| L_h \left(\frac{e_i}{h^2} - \psi_i \right) \right| \leqslant O(h^2) \quad i \neq j, \quad j+1,$$
$$\left| L_h \left(\frac{e_i}{h^2} - \psi_i \right) \right| \leqslant O(1) \quad i = j, \quad j+1. \tag{32}$$

Using Lemma 4.2 with the comparison function Φ in (22),

$$\left| \frac{e_i}{h^2} - \psi_i \right| \leqslant Ch, \tag{33}$$

so

$$e_i = \psi_i^{(1)} h^2 + \mathrm{O}(h^3). \tag{34}$$

We summarize the results in the following theorem:

Theorem 4.6. *Assume that $\beta^+ = \beta^- = 1$, the solution $u(x)$ is piecewise smooth and there is only one interface point α. Then the numerical solution U_i from the 1D PIM satisfies*

$$u_i - U_i = \psi^{(1)} h^2 + \mathrm{O}(h^3),$$

where $\psi^{(1)}$ is the solution of

$$\psi_{xx} = \frac{1}{12} u_{xxxx},$$
$$\psi(0) = \psi(1) = 0,$$
$$[\psi](\alpha) = 0,$$
$$[\psi_x](\alpha) = \frac{[u_{xxx}]}{3!} ((1 - \theta)^3 + \theta^3).$$

Note 14. With minor modification, the approach can be applied to piecewise constant coefficient cases. Compared with the discrete maximum principle, the asymptotic estimate gives a sharp error estimate and a basis for deferred correction.

5. The new multigrid solver

Our multigrid approach contains the usual components: an interpolation operator I_{2h}^h, a restriction operator I_h^{2h} and a relaxation scheme. Let the error equation on the fine grid be $A^h v^h = r^h$. The corresponding coarse grid equation is $A^{2h} v^{2h} = r^{2h}$, where the restriction and coarse-grid operators are defined as [25,5]:

$$I_h^{2h} := (I_{2h}^h)^T, \quad A^{2h} := I_h^{2h} A^h I_{2h}^h \tag{35}$$

and $r^{2h} := I_h^{2h} r^h$. Red–black Gauss–Seidel iteration [5] is used as our relaxation scheme, due to its efficiency and easy implementation. The algorithm recursively consider $A^{2h} v^{2h} = r^{2h}$ as the fine grid problem and telescopes down to the coarsest grid.

We employ a novel interpolation operator I_{2h}^h, which attains second-order accuracy even for complex interfaces on coarse grids. Higher order accuracy is attained similarly with a larger stencil.

5.1. Operator-dependent interpolation for regular points

For two-dimensional problems, we need to define the interpolation operator I_{2h}^h such that $v^h = I_{2h}^h v^{2h}$. Consider a generic nine-point stencil shown in Fig. 11. At coarse-grid points, the value of v^{2h} is simply copied to be the value of v^h. For fine-grid points that are on a vertical cell edge, we start with the discretization stencil (see Fig. 11) and assume the error residual is small:

$$\gamma_0 v_0 + \gamma_1 v_1 + \cdots + \gamma_8 v_8 \approx 0. \tag{36}$$

Away from the interface, v varies smoothly. Hence we approximate as

$$v_1, v_3 \approx v_0; \quad v_5, v_6 \approx v_2; \quad v_7, v_8 \approx v_4. \tag{37}$$

From (36), the interpolation scheme gives $v_0 = c_2 v_2 + c_4 v_4$ with

$$c_2 = -(\gamma_2 + \gamma_5 + \gamma_6)/(\gamma_0 + \gamma_1 + \gamma_3), \quad c_4 = -(\gamma_4 + \gamma_7 + \gamma_8)/(\gamma_0 + \gamma_1 + \gamma_3).$$

$$\begin{bmatrix} 6 & 2 & 5 \\ 3 & 0 & 1 \\ 7 & 4 & 8 \end{bmatrix} \qquad \begin{bmatrix} \gamma_6 & \gamma_2 & \gamma_5 \\ \gamma_3 & \gamma_0 & \gamma_1 \\ \gamma_7 & \gamma_4 & \gamma_8 \end{bmatrix}$$

Fig. 11. The numbering scheme for the nine-point stencil (left) and the discretization stencil (right).

The same strategy can be applied to fine-grid points that are on a horizontal cell edge. The interpolation scheme is $v_0 = c_1 v_1 + c_3 v_3$, where

$$c_1 = -(\gamma_1 + \gamma_5 + \gamma_8)/(\gamma_0 + \gamma_2 + \gamma_4), \quad c_3 = -(\gamma_3 + \gamma_6 + \gamma_7)/(\gamma_0 + \gamma_2 + \gamma_4).$$

The only fine-grid points to be interpolated are those in the center of the cell. Since all values of the neighboring grid points have been determined, we can solve (36) to obtain the fine-grid values.

5.2. Interpolation for irregular points

At irregular points near interfaces, the approximation (37) is no longer valid since the error after pre-relaxation steps can have a large jump in the normal derivative [3]. However, since the jump conditions have been explicitly incorporated into the PIM approach, it is natural to assume the following homogeneous jump conditions along the interface:

$$[v] = 0, \quad [\beta v_n] = 0, \quad [u_\tau] = 0. \tag{38}$$

The basic idea of the new interpolation is to construct a piecewise-polynomial approximation and then evaluate it at a fine-grid point as in the PIM. A piecewise linear polynomial is sufficient for second-order accuracy.

5.2.1. Single intersection in 1D

Consider the typical case shown in Fig. 12. To approximate $v(x)$ near the interface point α, we propose a piecewise linear polynomial

$$p(x) = \begin{cases} p_1(x) := a_1 x + a_2 & x < \alpha, \\ p_2(x) := a_3 x + a_4 & x > \alpha \end{cases}$$

satisfying

$$p(x_{i-1}^{2h}) = v_{i-1}^{2h}, \quad p(x_i^{2h}) = v_i^{2h}; \quad [p]_\alpha = 0, \quad [\beta p_x]_\alpha = 0.$$

The last two jump conditions come from (38). With four equations and four unknowns, we can solve for the coefficients $a_1, \ldots, a_4$ and thus the interpolated value $v_{2i-1}^h = p(x_{2i-1}^h)$ at the fine-grid point x_{2i-1}^h.

5.2.2. Multiple intersections in 1D

On a coarse grid, more than one interface point may separate neighboring grid points. A typical example is shown in Fig. 13. To approximate $v(x)$ near the interface points α_1 and α_2, we propose a piecewise linear polynomial

$$p(x) = \begin{cases} p_1(x) & x < \alpha_1, \\ p_2(x) & \alpha_1 < x < \alpha_2, \\ p_3(x) & x > \alpha_2 \end{cases}$$
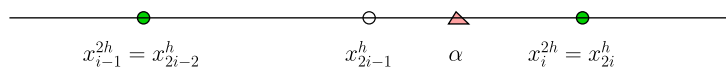
satisfying

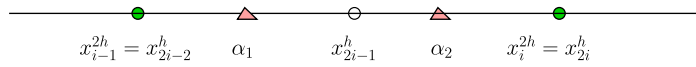Fig. 12. A single interface point α lies between coarse-grid points x_{i-1}^{2h} and x_i^{2h}.

Fig. 13. Two interface points α_1 and α_2 lie between coarse-grid points x_{i-1}^{2h} and x_i^{2h}.

$$p(x_{j-1}^{2h}) = v_{j-1}^{2h}, p(x_j^{2h}) = v_j^{2h}; \quad [p]_{\alpha_i} = 0, \quad [\beta p_x]_{\alpha_i} = 0, \quad i = 1, 2.$$

Solving for the coefficients gives the interpolated value $v_{2i-1}^h = p(x_{2i-1}^h)$ at the fine-grid point x_{2i-1}^h.

Note 15. The new interpolation in 1D is similar to [27]. For higher dimensions, however, [27] applies the 1D interpolation dimension-by-dimension by implicitly assuming that

$$[\beta u_\tau]_\Gamma \approx 0,$$

which may not be true if we have large jumps in β along the interface. Thus second-order accuracy can not be attained by the method of [27].

5.2.3. Single intersection in 2D

The 1D approach extends naturally to 2D. Consider the case shown in Fig. 14a. To approximate $v(x, y)$ near the interface point α, we propose a piecewise linear polynomial

$$p(x, y) = \begin{cases} p_1(x, y) := a_1 x + a_2 y + a_3 & (x, y) \in \Omega^+, \\ p_2(x, y) := a_4 x + a_5 y + a_6 & (x, y) \in \Omega^- \end{cases}$$

satisfying

$$\begin{aligned} & p(x_{i_k}^{2h}, y_{j_k}^{2h}) = v_{i_k, j_k}^{2h}, \quad (i_k, j_k) \in \mathbb{N}_{2i, 2j-1}, \\ & [p]_\alpha = 0, \quad [\beta p_n]_\alpha = 0, \end{aligned} \tag{39}$$

where $\mathbb{N}_{2i, 2j-1}$ is the index set of four coarse-grid points marked with filled circles in Fig. 14a. With six equations and six unknowns, we can solve for the coefficients and thus the interpolated value $v_{2i, 2j-1}^h = p(x_{2i}^h, y_{2j-1}^h)$ at the fine-grid point (x_{2i}^h, y_{2j-1}^h).

5.2.4. Multiple intersections in 2D

The same approach applies to the case when more than one interface point separate neighboring coarse-grid points. Consider the typical case shown in Fig. 14b. To approximate $v(x, y)$ near the interface points α_1, α_2, we propose a piecewise linear polynomial

$$p(x, y) = \begin{cases} p_1(x, y) := a_1 x + a_2 y + a_3 & (x, y) \in \Omega^+, \quad y \leqslant y_{\alpha_1}, \\ p_2(x, y) := a_4 x + a_5 y + a_6 & (x, y) \in \Omega^-, \\ p_3(x, y) := a_7 x + a_8 y + a_9 & (x, y) \in \Omega^+, \quad y \geqslant y_{\alpha_2} \end{cases}$$
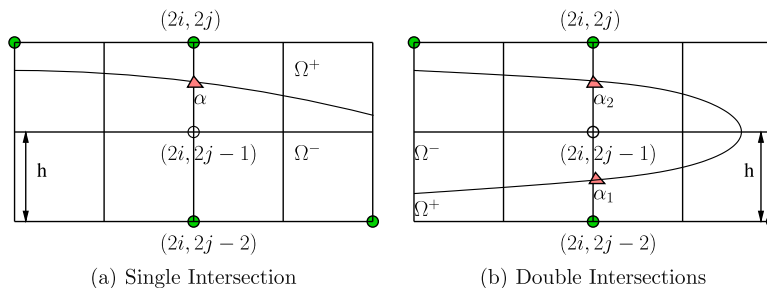

(a) Single Intersection   (b) Double Intersections

Fig. 14. Interpolation at (x_{2i}^h, y_{2j-1}^h). Filled circles are the coarse-grid points used for interpolation at the fine grid point(empty circle). Indices are for the fine mesh. Here $x_i^{2h} = x_{2i}^h$, and so forth.

satisfying

$$p(x_{i_k}^{2h}, y_{j_k}^{2h}) = v_{i_k, j_k}^{2h}, \quad (i_k, j_k) \in \mathbb{N}_{2i, 2j-1},$$

$$[p]_{\alpha_i} = 0, \quad [\beta p_n]_{\alpha_i} = 0, \quad [u_\tau] = 0, \quad i = 1, 2,$$

where $\mathbb{N}_{2i, 2j-1}$ is the index set of four coarse-grid points marked with filled circles in Fig. 14b and $(x_{\alpha_i}, y_{\alpha_i})$ is the coordinate of α_i, $i = 1, 2$. With 10 equations and 9 unknowns, we can solve for the coefficients via least squares and thus the interpolated value is $v_{2i, 2j-1}^h = p(x_{2i}^h, y_{2j-1}^h)$ at the fine-grid point (x_{2i}^h, y_{2j-1}^h).

Note 16. There are cases with too few neighboring points for interpolation due to high curvature of interfaces. As shown in Section 3.3, these cases can be resolved by adding more polynomials appropriately.

5.3. Multigrid cycles

The most commonly used iteration cycles in multigrid are V-cycle, F-cycle and W-cycle. The efficiency and robustness of them have been evaluated in our numerical experiments. We are particularly interested in F-cycle since it is cheaper than W-cycle and much more robust than V-cycle in our experiments. The algorithm is presented for convenience in a recursive format.

Algorithm 1: Multigrid F-cycle scheme

1: **procedure** MGF $(f^h, v^h, \mu_1, \mu_2, \mu_3)$
2:     **if** coarsest level **then**
3:         direct solver and return
4:     **else**
5:         Relax μ_1 times on $A^h u^h = f^h$ with initial guess v^h
6:         $f^{2h} = I_h^{2h}(f^h - A^h v^h)$
7:         $v^{2h} = 0$
8:         $v^{2h} = \text{MGF}(f^{2h}, v^{2h}, \mu_1, \mu_2, \mu_3)$
9:         Correct $v^h = v^h + I_{2h}^h v^{2h}$
10:         Relax μ_2 times on $A^h u^h = f^h$ with initial guess v^h
11:         **if** $\mu_3 > 0$
12:             $f^{2h} = I_h^{2h}(f^h - A^h v^h)$
13:             $v^{2h} = 0$
14:             $v^{2h} = \text{MGF}(f^{2h}, v^{2h}, \mu_1, \mu_2, 0)$
15:             Correct $v^h = v^h + I_{2h}^h v^{2h}$
16:             Relax μ_3 times on $A^h u^h = f^h$ with initial guess v^h
17:         **end if**
18:     **end if**
19. **end procedure**

5.4. Spectral analysis

In this section we investigate the eigenvalue spectra of iteration matrices. We are interested in exploring the following questions:

(1) How is the F-cycle compared with the V-cycle in terms of convergence rates?
(2) How is the spectrum of iteration matrices related to convergence rates?
(3) How does the spectra of iteration matrix change as the jump in coefficients increases?

We denote the discretized linear system as

$$Au = b.$$

To study the convergence rates of various multigrid cycles, we split the matrix in the form

$$A = K + (A - K)$$

with K being the iteration matrix. As a result, we can express one iteration of multigrid cycle as

$$Ku^{(i+1)} + (A - K)u^{(i)} = b$$

or, equivalently

$$u^{(i+1)} = (I - K^{-1}A)u^{(i)} + K^{-1}b.$$

Therefore, we have

$$r^{(i+1)} = (I - AK^{-1})r^{(i)}, \quad e^{(i+1)} = (I - K^{-1}A)e^{(i)}, \tag{40}$$

where $r^{(i)} = b - Au^{(i)}$, $e^{(i)} = u - u^{(i)}$. Eq. (40) displays the close relation between asymptotic convergence rates and the spectral radius of $I - AK^{-1}$. Note that $I - K^{-1}A$ has the same spectral radius as $I - AK^{-1}$. We consider the following model problem. The interface is $x^2 + y^2 = 0.50012563^2$ and the exact solution is

$$u(x,y) = \begin{cases} r^2 & \text{if } (x,y) \in \Omega^-, \\ \left(1 - \frac{1}{8b} - \frac{1}{b}\right)/4 + \left(\frac{r^4}{2} + r^2\right)/b + \log(2r)/(10b) & \text{otherwise,} \end{cases}$$

where $r = \sqrt{x^2 + y^2}$ and the diffusion coefficient is

$$\beta(x,y) = \begin{cases} x^2 + y^2 + 1 & \text{if } (x,y) \in \Omega^-, \\ b & \text{otherwise.} \end{cases}$$

First consider a moderate value such as $b = 10$. Fig. 15a and b presents the spectra for the V-cycle and F-cycle, respectively, where ρ is the spectral radius of the corresponding matrices. These graphs show two important features: (i) most eigenvalues are clustered around zero for both cases, and (ii) the spectra radius ρ from F-cycle (0.007) is much smaller than that from V-cycle (0.073).

Fig. 15c illustrates the progress of iterations for V-cycle and F-cycle. The graphs show $\log(\|r^{(n)}\|_2/\|r^{(0)}\|_2)$ versus iterations where $\|r^{(n)}\|_2$ is the 2-norm of the residual vector after n iterations. It is clear that the asymptotic convergence rate ρ_e of F-cycle (0.006) is much smaller than that of V-cycle (0.073), where we have $\rho_e := \|r^{(m)}\|_2/\|r^{(0)}\|_2$ with m being the total number of iterations. The F-cycle is thus preferred for our test problems even though it costs a little more than the V-cycle. Comparing Fig. 15a–c, we see that the spectral radius determines the convergence rate as expected.

In the second experiment, we let the value of b vary and study the eigenvalue spectra of the iteration matrices from the F-cycle. Fig. 15d shows the relation between b and ρ. Fig. 16 shows the detailed plots for selected values of b. We have the following observations:

- Most eigenvalues are clustered around zero for all cases, which is advantageous for the Krylov methods [18].
- As $b \to \infty$, the spectral radius increases but tends to converge to some value much smaller than 1.0. Part of the reason is that the corresponding continuous problem is always well-posed.
- As $b \to 0$, the spectral radius increases rapidly and eventually exceeds 1 for $b = 0.0001$ (see Fig. 15d). As a result, the F-cycle diverges as confirmed by numerical experiments.

As shown by the numerical examples in Section 7, the Krylov-accelerated multigrid approach effectively removes large eigenvalues, reduces the spectral radius and thus converges rapidly for all cases.

6. Preconditioned GMRES(m)

We accelerate multigrid by Krylov subspace iteration. In particular, we use multigrid as preconditioner for GMRES [22]. The motivation comes from the inefficiency of multigrid as a solver alone in the case of large-jump coefficients.
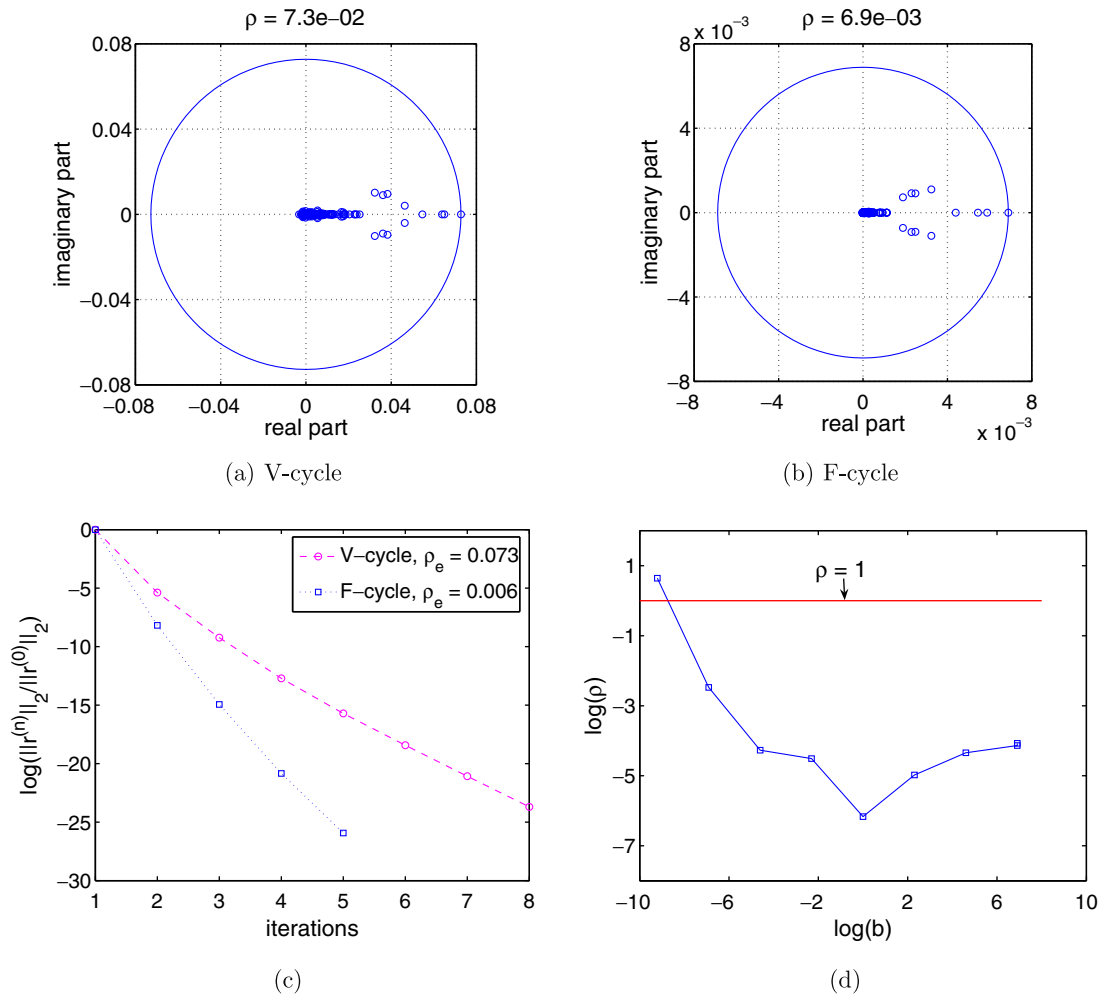
Fig. 15. Spectral analysis with $N = 33$: (a) and (b) eigenvalue spectrum plots for $b = 10$; (c) the convergence comparison for $b = 10$; and (d) spectral radius of the F-cycle with $b = 10^{-4}, 10^{-3}, \ldots, 10^4$.

Algorithm 2: GMRES(m) algorithm with F-cycle as a right preconditioner

1: **procedure** GMRES $(A, x, b, \mu_1, \mu_2, \mu_3)$
2:     $r_0 = b - Ax^{(0)}$, $\beta = \|r_0\|_2$, and $v_1 = r_0/\beta$
3:     **for** $j = 1, \ldots, m$ **do**
4:         $u = \mathrm{MGF}(v_j, 0, \mu_1, \mu_2, \mu_3)$
5:         $w = Au$
6:         **for** $i = 1, \ldots, j$ **do**
7:             $h_{i,j} = (w, v_i)$
8:             $w = w - h_{i,j}v_i$
9:         **end for**
10:        $h_{j+1,j} = \|w\|_2$, $v_{j+1} = w/h_{j+1,j}$
11:        Define $V_m = [v_1, \ldots, v_m]$, $H_m = \{h_{i,j}\}_{1 \leqslant i \leqslant j+1; 1 \leqslant j \leqslant m}$
12:    **end for**
13:    $y_m = \mathrm{argmin}_y \|\beta e_1 - H_m y\|_2$, $x_m = x_0 + \mathrm{MGF}(V_m y_m, 0, \mu_1, \mu_2, \mu_3)$
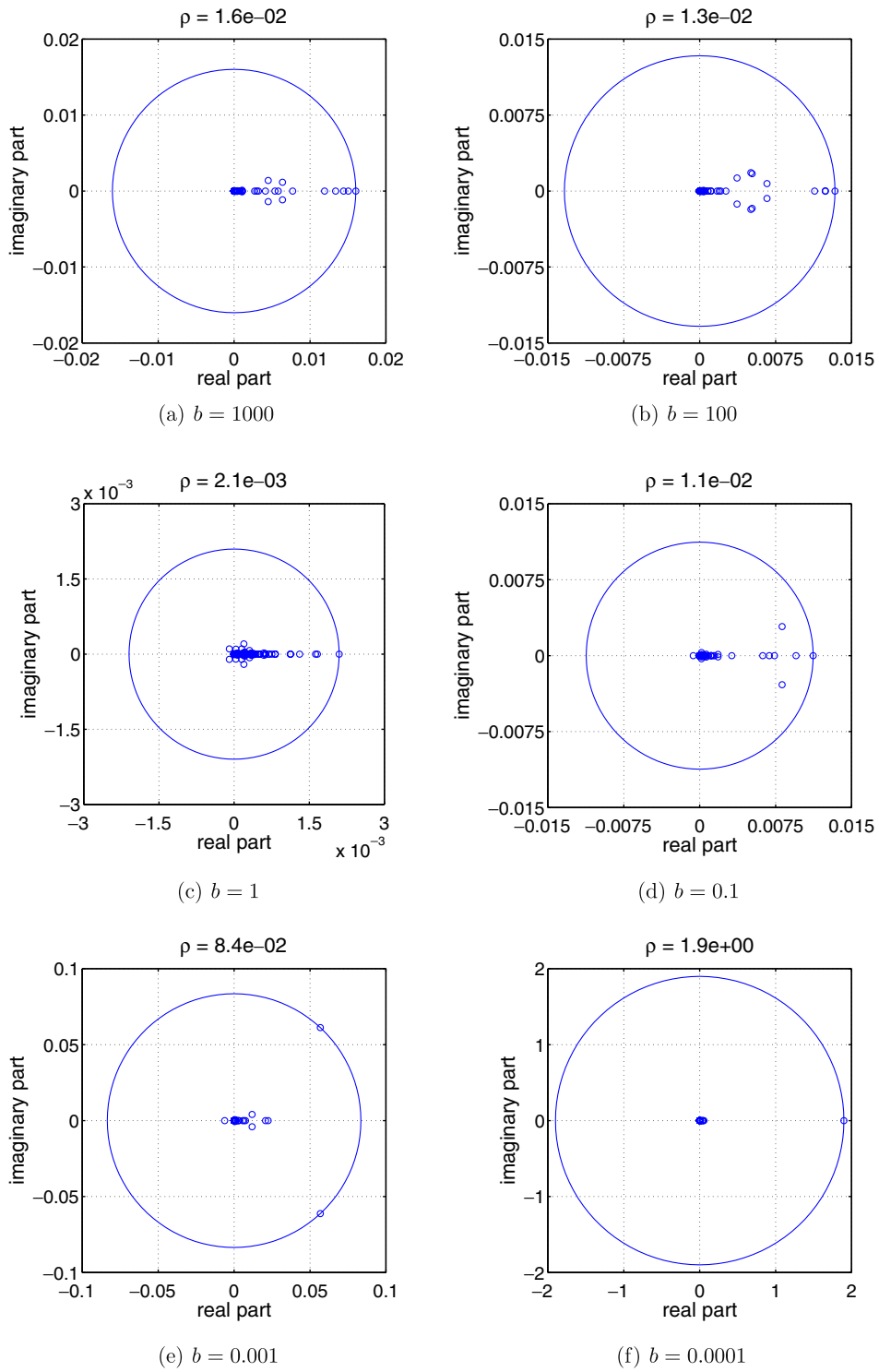14:   If satisfied stop, else set $x_0 = x_m$ and restart
15: **end procedure**

Fig. 16. Eigenvalue spectrum plots for the F-cycle with $N = 33$. Here ρ is the spectral radius.

The restarted version GMRES(m) is a robust solver for the linear system

$$Ax = b,$$

where A is a sparse, unsymmetric matrix. If used as a solver alone, GMRES(m) may not be efficient. A large number of iterations are expected, especially for the case when A is ill-conditioned. In many practical applications, people use the preconditioned GMRES(m) to accelerate the convergence. The right preconditioned GMRES(m) algorithm is based on solving

$$AM^{-1}u = b, \quad u = Mx.$$

As is shown in [21], GMRES(m) builds an orthogonal basis of the right-preconditioned Krylov subspace

$$\text{span}\{r_0, AM^{-1}r_0, \ldots, (AM^{-1})^{m-1}r_0\}.$$

As a result, the approximate solution minimizes the 2-norm residual $\|b - Ax\|_2$ among all vectors from the affine subspace

$$x_0 + \text{span}\{z_0, M^{-1}Az_0, \ldots, (M^{-1}A)^{m-1}z_0\}$$

in which z_0 is the preconditioned residual $z_0 = M^{-1}r_0$. Equivalently, the associated residual $r = b - Ax$ has the minimal 2-norm among all vectors belonging to

$$r_0 + \text{span}\{AM^{-1}r_0, (AM^{-1})^2 r_0, \ldots, (AM^{-1})^m r_0\}.$$

This observation is useful in constructing the GMRES minimal polynomial explicitly. The pseudocode for GMRES(m) right preconditioned with multigrid is provided in Algorithm 2.

7. Numerical examples

We have tested our PIM discretization and IMG solver with a number of 2D experiments. We investigate the accuracy of the computed solution, and the efficiency of the multigrid solver for problems with complex interfaces or high-contrast coefficients.

Our test problem is

$$Lu := \nabla \cdot (\beta \nabla u) = f \quad \text{in } \Omega^+ \cup \Omega^-,$$
$$[u] = w, \quad [\beta u_n] = v \quad \text{on } \Gamma,$$
$$u = g \qquad\qquad \text{on } \partial\Omega$$

with $\Omega := \Omega^+ \cup \Gamma \cup \Omega^- = [-1, 1] \times [-1, 1]$. A Cartesian grid is used with

$$x_i = -1 + (i-1)h, \quad y_j = -1 + (j-1)h,$$

where $h = 2/(N-1)$ and N varying from 20 to 600.

The order of the scheme is estimated as

$$\text{order} := \left| \frac{\log(\|E_{2N}\|_\infty / \|E_N\|_\infty)}{\log(2)} \right|,$$

where $\|E_N\|_\infty$ is the maximum error

$$\|E_N\|_\infty := \max_{i,j} |u(x_i, y_j) - U_{ij}|$$

on an $N \times N$ grid.

Interfaces are represented with linear line segments where we use 3000 control points for smooth interfaces with high curvature. The only exception is the pentagon example, where only 10 control points are needed. For the new multigrid solver, we use two pre- and two post-Gauss–Seidel smoothing steps. The coarsest grid is 9×9 and the iteration is stopped when the relative residual norm satisfies

$$\frac{\|L_h U - f_h\|_2}{\|f_h\|_2} < 10^{-10}, \tag{41}$$

where our discretization contributes the linear system $L_h U = f_h$.

We test four different solvers:

- GMRES-F: GMRES as solver right preconditioned with the F-cycle.
- BiCGSTAB-F: BiCGSTAB as solver right preconditioned with the F-cycle.
- The direct solver SuperLU [8].
- The AMG method using AMG1R6 [20,24].

To measure the performance of Krylov-accelerated multigrid solver, we define

n_{it} := number of iterations to satisfy 10-digit accuracy (41),

t_{it} := total CPU seconds for the new Krylov-accelerated multigrid solver.

All the tests are done on one processor of a Sun Java workstation W2100z with 3.2 GB memory and dual AMD 1.8 GHz Opteron CPUs. The code is written in C++ and compiled with gcc 4.0. The implementation of the methods is sequential and unoptimized.

7.1. The IMG solver

We compare the IMG solver with AMG and SuperLU and study the effect of high-contrast coefficients.

7.1.1. Example 1: solver comparison

Our initial motivation was to design a fast efficient solver. In this example, we would like to compare the performance, in terms of CPU seconds, of our iterative solver GMRES-F, algebraic multigrid (AMG) and direct solver SuperLU [8]. We test the AMG method using AMG1R6 written by Ruge, Stüben and Hempel with version date 1997. The interface is (see Fig. 1)

$$\begin{cases} X(\theta) = (a + b\cos(m\theta)\sin(n\theta))\cos\theta, \\ Y(\theta) = (a + b\cos(m\theta)\sin(n\theta))\sin\theta \end{cases}$$

with $a = b = 0.40178$ and $m = 2, n = 6$. The exact solution is

$$u(x,y) = \begin{cases} x - y^2 & \text{if } (x,y) \in \Omega^+, \\ e^x \cos \pi y & \text{otherwise} \end{cases}$$

and the diffusion coefficient is

$$\beta(x,y) = \begin{cases} b & \text{if } (x,y) \in \Omega^+, \\ 1 + x^2 + y^2 & \text{otherwise.} \end{cases}$$

Table 2 shows how the CPU time (in seconds) grows with respect to mesh size N. For the case $b = 1000$, timings of GMRES-F and AMG are comparable. Our experiments show that their performance slightly surpass SuperLU for $N \geqslant 400$ and we expect larger differences as N increases. SuperLU also requires a large amount of memory and easily goes beyond what is available for $N \geqslant 700$ or so.

However, for the case $b = 0.001$, the AMG solver slows down dramatically, while GMRES-F and SuperLU remain similar to the previous case. In both cases, GMRES-F performs very well and least-squares analysis shows a growth rate of 2.0 as expected.

7.1.2. Example 2: the effect of high-contrast coefficients

Most standard multigrid approaches perform poorly for elliptic problems with high-contrast coefficients. We would like to study how different jumps in coefficients affect the convergence of the IMG solver, as well

Table 2
Example 1: CPU time (in seconds) comparison

N	$b = 1000$			$b = 0.001$		
	GMRES-F	SuperLU	AMG	GMRES-F	SuperLU	AMG
33	0.1	0.1	0.1	0.1	0.1	0.9
65	0.2	0.1	0.2	0.2	0.1	1.4
129	0.8	0.3	0.8	1.2	0.3	10.2
257	3.4	3.0	3.5	6.8	3.0	96.5
513	13.9	23.7	15.8	24.4	22.4	556.3

as the accuracy of our PIM discretization. Both GMRES-F and BiCGSTAB-F are used to verify the robustness of the Krylov-accelerated multigrid solver.

The interface is $x^2 + y^2 = 0.50012563^2$. The exact solution is

$$u(x,y) = \begin{cases} \sin(x+y) + 1 & \text{if } (x,y) \in \Omega^+, \\ \cos(x+y) + 1 & \text{otherwise} \end{cases}$$

and the diffusion coefficient

$$\beta(x,y) = \begin{cases} b & \text{if } (x,y) \in \Omega^+, \\ 1 & \text{otherwise}. \end{cases}$$

Here b is a contrast parameter varying from 10^{-3} to 10^3 and introducing ill conditioning when small. Table 3 displays iteration counts and CPU seconds for GMRES(20) preconditioned with the IMG solver. Grid-independent convergence is confirmed for all values of b. CPU time grows linearly with the number of unknowns. As the jump in β along the interface becomes larger, the iteration number grows but remains small. Note that as b approaches 0, in addition to large jump in β, the continuous problem becomes more ill-conditioned (see Fig. 17b).

In Table 4 BiCGSTAB [26] is the solution method preconditioned with the IMG. A similar convergence pattern is evident. For each iteration, BiCGSTAB requires two matrix–vector products and two preconditioner solves, while GMRES needs one matrix–vector product and one preconditioner solve. This explains why iteration counts in BiCGSTAB-F are always less than those in GMRES-F. The CPU seconds between them are always comparable. Table 5 shows the convergence analysis.

7.2. Accuracy and convergence analysis

We investigate the accuracy of the computed solutions with complex interfaces and high-contrast coefficients.

7.2.1. Example 3: complex interfaces
The exact solution is

$$u(x,y) = \begin{cases} x + y + 1 & \text{if } (x,y) \in \Omega^+, \\ \sin(x+y) + \cos(x+y) + 1 & \text{otherwise} \end{cases}$$

Table 3
GMRES(20)-F: each entry represents $(n_{it})t_{it}$

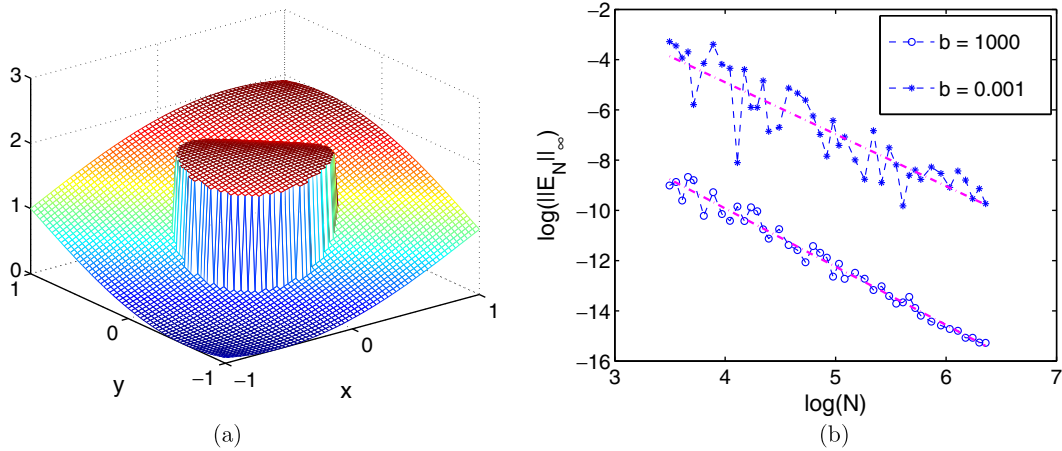| N	b						
	0.001	0.01	0.1	1	10	100	1000
33	(5)0.1	(5)0.1	(4)0.1	(3)0.0	(4)0.1	(4)0.1	(4)0.1
65	(6)0.3	(5)0.2	(4)0.2	(3)0.2	(4)0.2	(5)0.2	(5)0.2
129	(6)1.2	(6)1.2	(4)0.9	(3)0.8	(4)1.0	(5)1.1	(5)1.1
257	(6)5.5	(6)5.6	(4)4.3	(3)3.6	(4)4.3	(5)5.0	(5)4.9
513	(7)23.8	(6)21.7	(5)18.8	(4)16.4	(4)16.3	(5)18.8	(5)18.7

Fig. 17. Example 2: (a) the computed solution for $N = 65$, $b = 1000$; (b) the maximum error $\|E_N\|_\infty$ versus the mesh size N in log–log scale.

Table 4
BiCGSTAB-F: each entry represents $(n_{it})t_{it}$

N	b						
	0.001	0.01	0.1	1	10	100	1000
33	(3)0.1	(3)0.1	(2)0.1	(2)0.0	(2)0.1	(2)0.0	(2)0.0
65	(3)0.2	(5)0.3	(2)0.2	(2)0.2	(2)0.2	(3)0.2	(3)0.2
129	(4)1.3	(4)1.2	(2)0.8	(2)0.7	(2)0.8	(3)0.9	(3)1.0
257	(4)5.3	(4)5.0	(2)3.5	(2)2.9	(2)3.5	(3)4.0	(3)4.1
513	(4)20.6	(4)20.5	(5)25.3	(2)13.7	(2)13.9	(3)16.1	(3)16.1

Table 5
Example 2: high-contrast coefficients

| N	$b = 0.001$		$b = 0.1$		$b = 10$		$b = 1000$	
	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order
33	3.760e−02	–	9.155e−04	–	1.250e−04	–	1.224e−04	–
65	1.237e−02	1.6	1.896e−04	2.3	3.245e−05	1.9	2.974e−05	2.0
129	9.300e−04	3.7	1.200e−05	4.0	7.270e−06	2.2	8.397e−06	1.8
257	2.782e−04	1.7	3.206e−06	1.9	9.187e−07	3.0	1.110e−06	2.9
513	7.230e−05	1.9	8.242e−07	2.0	2.448e−07	1.9	2.851e−07	2.0

and the diffusion coefficient

$$\beta(x,y) = \begin{cases} (xy + 2)/5 & \text{if } (x,y) \in \Omega^+, \\ (x^2 - y^2 + 3)/7 & \text{otherwise.} \end{cases}$$

The interface is given by

$$\begin{cases} X(\theta) = (a + b\cos(m\theta)\sin(n\theta))\cos\theta, \\ Y(\theta) = (a + b\cos(m\theta)\sin(n\theta))\sin\theta. \end{cases}$$

For case I, $a = b = 0.40178$ and $m = 2, n = 6$ (see Fig. 1a). For case II, $a = 0.50012563$, $b = 0.250012563, m = 0$ and $n = 12$. GMRES-F is used to solve the discretized system. Fig. 18a and b plot the computed solutions for both cases on a 65×65 mesh. Fig. 18c and d are plots of errors versus the mesh
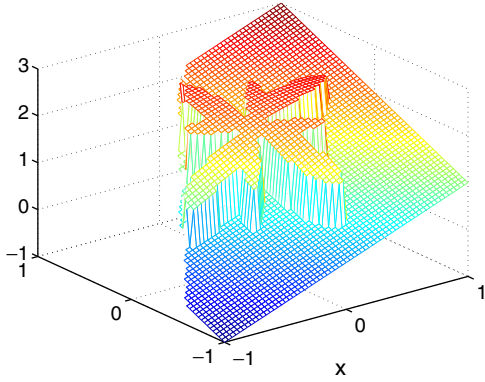
size N in log–log scale for both cases with N varying from 33 to 577. The least-square fit produces an asymptotic convergence rate of 2.39 for case I and 2.46 for case II.

Table 6 exhibits the computed errors and confirms second-order convergence rates. The grid-independence speed of the IMG solver is clearly displayed.

7.2.2. Example 4: complex interface with high-contrast coefficients

The interface is (see Fig. 19a)

$$\begin{cases} X(t) = r(t)\cos(\theta(t)), \\ Y(t) = r(t)\sin(\theta(t)) \end{cases}$$

with

$$\theta(t) = t + \sin(4t), \quad r(t) = 0.6012563 + 0.2401256\cos(4t + \pi/2).$$

Note that the curvature can be as large as 255 at certain points. The exact solution is

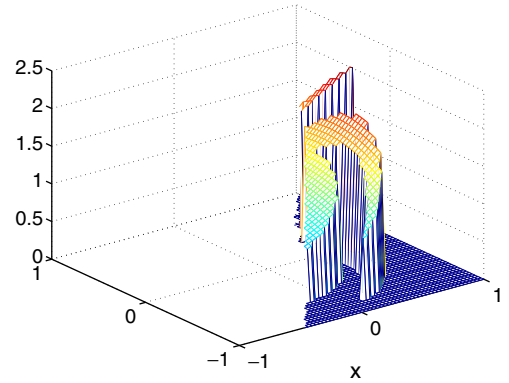$$u(x,y) = \begin{cases} 0 & \text{if } (x,y) \in \Omega^+, \\ e^x\cos y & \text{otherwise} \end{cases}$$

and the diffusion coefficient is

Table 6
Example 3: complex interfaces

N	Case I				Case II			
	n_{it}	t_{it}	$\|E_N\|_\infty$	Order	n_{it}	t_{it}	$\|E_N\|_\infty$	Order
33	3	0.1	1.967e−04	–	4	0.1	2.774e−04	–
65	4	0.2	4.237e−05	2.2	4	0.2	4.635e−05	2.6
129	3	0.9	9.462e−06	2.2	4	1.1	8.702e−06	2.4
257	3	3.9	1.707e−06	2.5	4	4.6	1.865e−06	2.2
513	4	16.5	4.393e−07	2.0	4	17.1	3.388e−07	2.5
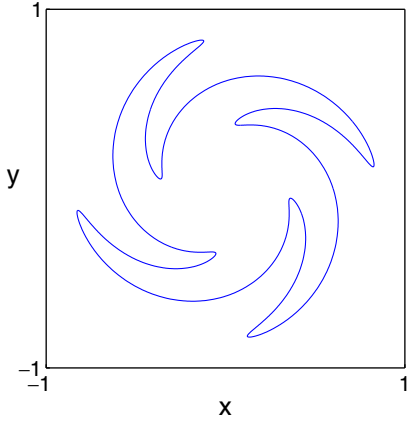
(a) Interface

$$\beta(x,y) = \begin{cases} b & \text{if } (x,y) \in \Omega^+, \\ 1 + x^2 + y^2 & \text{otherwise,} \end{cases}$$

where b is either 1000 or 0.001. Hence this example contains both a large-curvature interface and variable high-contrast coefficients.

GMRES-F is used to solve the discretized system. Fig. 19b plots the computed solution on a 65×65 mesh. Fig. 19c and d plot errors versus the mesh size N in log–log scale for both cases with N varying from 33 to 577. The least-square fit produces an asymptotic convergence rate of 2.2 for $b = 1000$ and 1.8 for $b = 0.001$.

Table 7 exhibits the computed errors and confirms second-order convergence rates. The grid-independence of the IMG solver is clearly displayed for $b = 1000$, while the iteration number grows a little but still is reasonable for the ill-posed case $b = 0.001$.

7.2.3. Example 5: interface with corners

The pentagon interface of Fig. 20a if given by

$$\begin{cases} X(\theta) = r(\theta) \cos \theta, \\ Y(\theta) = r(\theta) \sin \theta \end{cases}$$

with

$$r(\theta) = \begin{cases} \frac{R \sin(\theta_t/2)}{\sin(\theta_t/2 + \theta - \theta_r - 2\pi(i-1)/5)}, & \text{for } \theta_r + \pi(2i-3)/5 \leqslant \theta < \theta_r + \pi(2i-2)/5, \\ \frac{R \sin(\theta_t/2)}{\sin(\theta_t/2 - \theta + \theta_r + 2\pi(i-1)/5)}, & \text{for } \theta_r + \pi(2i-2)/5 \leqslant \theta < \theta_r + \pi(2i-1)/5. \end{cases}$$

The exact solution is

$$u(x,y) = \begin{cases} 0 & \text{if } (x,y) \in \Omega^+, \\ e^{\pi x} \cos(\pi y) + 5 & \text{otherwise} \end{cases}$$

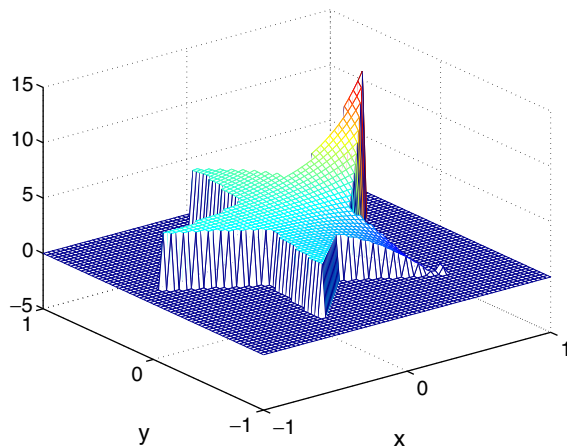and the diffusion coefficient is

Table 7
Example 4: complex interfaces with variable high-contrast coefficients

N	$b = 1000$				$b = 0.001$			
	n_{it}	t_{it}	$\|E_N\|_\infty$	Order	n_{it}	t_{it}	$\|E_N\|_\infty$	Order
33	7	0.1	2.867e−04	–	8	0.1	6.099e−03	–
65	8	0.4	6.887e−05	2.1	9	0.4	4.808e−03	0.3
129	8	1.6	1.147e−05	2.6	11	2.1	9.289e−04	2.4
257	8	6.8	3.166e−06	1.9	10	8.1	3.931e−04	1.2
513	7	24.6	5.321e−07	2.6	10	32.6	7.748e−05	2.3
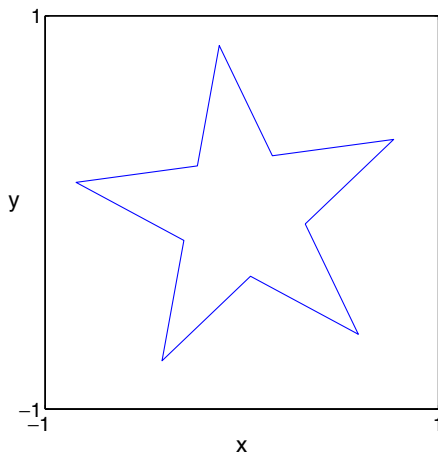
(a) Interface   (b) Computed Solution

Fig. 20. Example 5: (a) the interface Γ; (b) the absolute value of the computed solution on a 65×65 mesh for $b = 1000$.

$$\beta(x,y) = \begin{cases} b & \text{if } (x,y) \in \Omega^+, \\ 1 & \text{otherwise,} \end{cases}$$

where b is either 1000 or 1. Hence this example contains both a large-curvature interface and high-contrast coefficients.

This problem is taken from the MIB [29], but the case here is more challenging since b goes up to 1000 instead of 1. GMRES-F is used to solve the discretized system. Fig. 20b plots the computed solution on a 65×65 mesh.

Table 8 exhibits the computed errors and displays excellent second-order convergence rates. The accuracy is also slightly better than shown in [29], despite the high-contrast coefficients. The grid-independence of the IMG solver is clearly displayed for both $b = 1$ and $b = 1000$.

7.3. Comparison and accuracy study

We demonstrate the accuracy of our method when applied to problems considered by previous authors [4,13,14,28]. For interfaces with moderate curvature, it is never significantly worse and sometimes better.

7.3.1. Example 6

The interface is $x^2 + y^2 = 0.25$. The exact solution is

$$u(x,y) = \begin{cases} r^2 & \text{if } r \leqslant 0.5, \\ \left(1 - \frac{1}{8b} - \frac{1}{b}\right)/4 + \left(\frac{r^4}{2} + r^2\right)/b + C\log(2r)/b & \text{otherwise} \end{cases}$$

with the diffusion coefficient

$$\beta(x,y) = \begin{cases} x^2 + y^2 + 1 & \text{if } r \leqslant 0.5, \\ b & \text{otherwise.} \end{cases}$$

As a result, the jump conditions are

$$[u] = 0, \quad [\beta u_n] = 2C, \quad [u_n] = (2C + 5/4)/b - 1.$$

This problem has been studied with the IIM [13] and DIIM [4]. SuperLU is used to solve the discretized system. In our first test, $b = 10$ and $C = 0.1$. Fig. 21a plots the computed error on a 40×40 mesh and Fig. 21b exhibits a least-square fit of the convergence rate. Table 9 compares the computed errors to the results from [4,13]. We observed a second-order convergence rate and slightly improved accuracy from the PIM.

In our second test, we set $b = 1000$ to give a large contrast in the coefficients. Table 10 exhibits an error behavior comparable to the results of [4,13].

7.3.2. Example 7: composite material problem

We also consider a composite material problem with piecewise constant coefficients, typically generating large differences in material properties. The interface is $x^2 + y^2 = 0.25$. The exact solution is

Table 8
Example 5: sharp-edged interfaces with high-contrast coefficients

N	$b = 1$				$b = 1000$			
	n_{it}	t_{it}	$\|E_N\|_\infty$	Order	n_{it}	t_{it}	$\|E_N\|_\infty$	Order
33	3	0.0	1.847e−02	–	5	0.1	1.057e−02	–
65	3	0.1	4.645e−03	2.0	6	0.2	2.801e−03	1.9
129	3	0.6	5.191e−04	3.2	5	0.9	4.472e−04	2.6
257	3	2.4	1.295e−04	2.0	5	3.6	9.201e−05	2.3
513	4	13.6	3.184e−05	2.0	6	18.3	2.172e−05	2.1
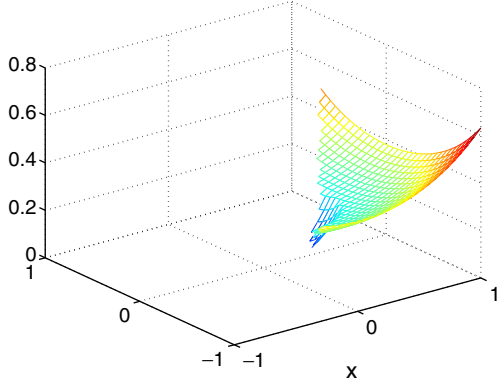
Table 9
Example 6: $b = 10$, $C = 0.1$

N	DIIM		IIM		PIM	
	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order
20	1.394e−03		3.520e−03		8.243e−04	
40	3.228e−04	2.1	7.561e−04	2.2	2.514e−04	1.7
80	7.857e−05	2.0	1.651e−04	2.2	6.404e−05	1.9
160	1.925e−05	2.0	3.600e−05	2.2	1.499e−05	2.1
320	4.774e−06	2.0	8.441e−06	2.1	3.910e−06	1.9

Table 10
Example 6: $b = 1000$, $C = 0.1$

N	DIIM		IIM		PIM	
	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order
32	2.083e−04		5.136e−04		2.401e−04	
64	5.296e−05	2.0	8.235e−05	2.8	5.726e−05	2.0
128	1.330e−05	2.0	1.869e−05	2.2	1.401e−05	2.0
256	3.330e−06	2.0	4.026e−06	2.2	3.461e−06	2.0

$$u(x,y) = \begin{cases} \frac{2x}{s+1+(s-1)/4} & \text{if } r \leqslant 0.5, \\ \frac{x(s+1)-(s-1)x/(4r^2)}{s+1+(s-1)/4} & \text{otherwise} \end{cases}$$

and the diffusion coefficient is

$$\beta(x,y) = \begin{cases} \beta^- & \text{if } r \leqslant 0.5, \\ \beta^+ & \text{otherwise}. \end{cases}$$

Here $s = \beta^-/\beta^+$ is the contrast ratio and $r = \sqrt{x^2 + y^2}$. Hence the jump conditions are

$$[u] = 0, \quad [\beta u_n] = 0.$$

This example has been studied with the FIIM [14], EJIIM [28] and DIIM [4]. SuperLU is used to solve the discretized system. A grid refinement analysis is summarized in Table 11 for $s = 5000$ and Table 12 for $s = 1/5000$. The results from [4,14,28] are also presented for comparison purpose. We observed robust performance from the PIM for both cases (see Fig. 22).

(a) $s = 5000$

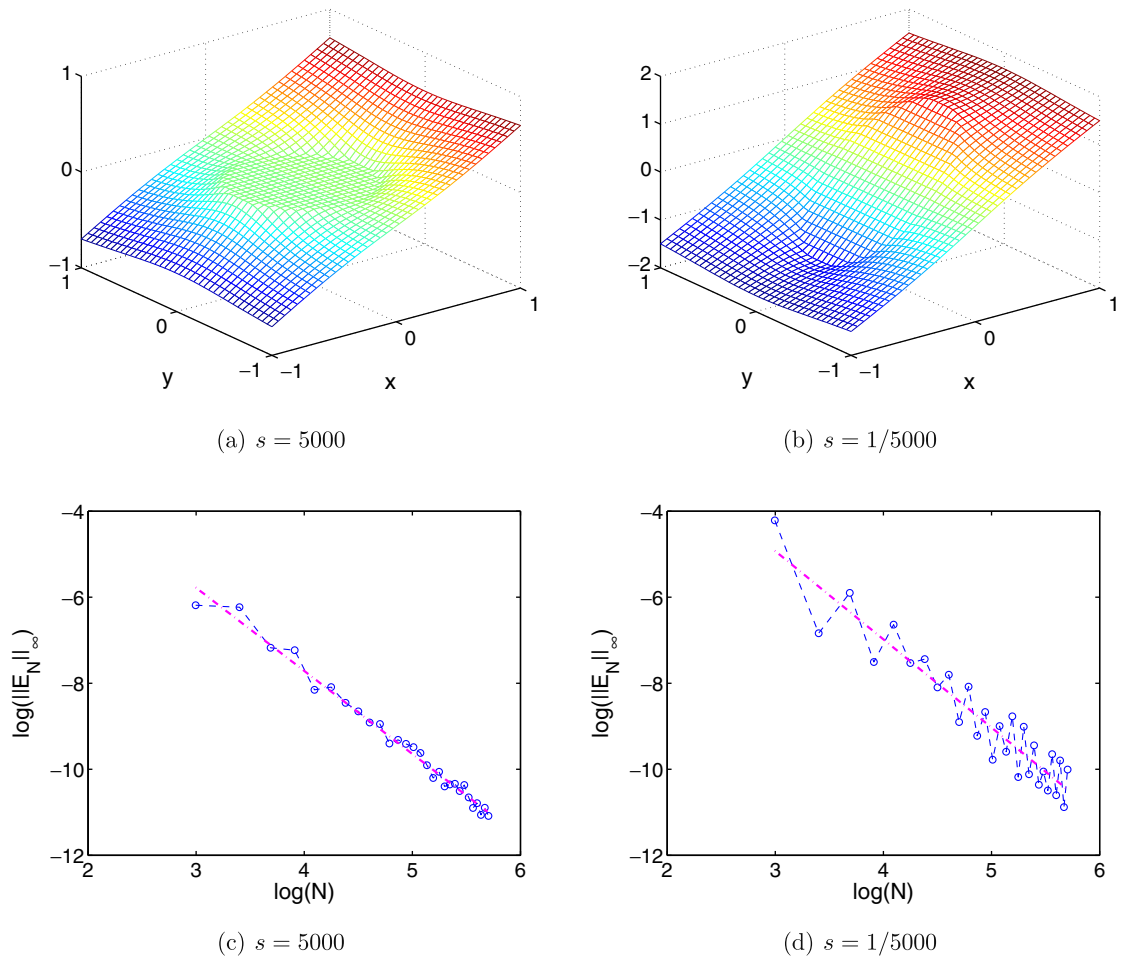(b) $s = 1/5000$

(c) $s = 5000$

(d) $s = 1/5000$

Fig. 22. Example 7: (a) and (b) the computed solutions on a 40×40 mesh; (c) and (d) the maximum error $\|E_N\|_\infty$ versus the mesh size N in log–log scale for $N = 20{:}10{:}300$.

Table 11
Example 7: $\beta^- = 5000$, $\beta^+ = 1$

N	DIIM		EJIIM		FIIM		PIM	
	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order
50	3.278e−04		3.5e−04		9.2e−02		9.686e−04	
100	5.277e−05	2.6	9.0e−05	2.0	5.9e−02	0.6	1.269e−04	2.9
200	1.371e−05	1.9	2.2e−05	2.0	7.7e−03	2.9	3.274e−05	1.9
400	3.653e−06	1.9	–	–	–	–	7.856e−06	2.1

Here – denotes unpublished data.

Table 12
Example 7: $\beta^- = 1$, $\beta^+ = 5000$

N	DIIM		EJIIM		FIIM		PIM	
	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order	$\|E_N\|_\infty$	Order
50	7.038e−03		5.5e−04		1.6e−03		5.559e−04	
100	1.934e−03	1.9	1.3e−04	2.1	2.3e−04	2.8	3.889e−04	0.5
200	5.209e−04	1.9	3.2e−05	2.0	5.0e−05	2.2	1.184e−04	1.7
400	1.346e−04	2.0	–	–	–	–	2.730e−05	2.1

Here – denotes unpublished data.

8. Conclusions

We have proposed an efficient geometric multigrid method for solving elliptic problems with complex interfaces separating high-contrast materials. The multigrid approach is further enhanced by Krylov subspace acceleration methods. The continuous problem is discretized by a new piecewise-polynomial approach. Theoretical analysis confirms second-order accuracy of the multigrid interpolation and spectrum analysis justifies the Krylov-accelerated multigrid approach. Numerical results for problems with complex interfaces and high-contrast coefficients demonstrate the efficiency and robustness of the new approach, which is also being applied to over-determined elliptic systems [23].

Appendix A

A.1. Formulas for the double-intersection case

$$\lambda_{1,1} = 1 - \frac{\beta^-}{\hat{\beta}}, \quad \lambda_{2,1} = \frac{\beta^-}{\hat{\beta}},$$

$$\lambda_{1,2} = \frac{\beta^-}{\hat{\beta}}, \quad \lambda_{2,2} = 1 - \frac{\beta^-}{\hat{\beta}},$$

$$\lambda_{1,3} = \frac{\beta^-}{\hat{\beta}}, \quad \lambda_{2,3} = -\frac{\beta^-}{\hat{\beta}},$$

$$\lambda_{1,4} = -\frac{\beta^-}{\hat{\beta}}, \quad \lambda_{2,4} = \frac{\beta^-}{\hat{\beta}},$$

$$\lambda_{1,5} = \frac{(1-\theta_2)\beta^- + \theta_3\beta^+}{\beta^+\hat{\beta}} h, \quad \lambda_{2,5} = \frac{\theta_1\beta^-}{\beta^+\hat{\beta}} h,$$

$$\lambda_{1,6} = -\frac{(1-\theta_2)\beta^-}{\beta^+\hat{\beta}} h, \quad \lambda_{2,6} = -\frac{\theta_1\beta^- + \theta_3\beta^+}{\beta^+\hat{\beta}} h,$$

$$\lambda_{1,7} = \frac{1 - 2\theta_1 + \theta_1^2\beta^-/\hat{\beta}}{2\beta^+} h^2, \quad \lambda_{2,7} = -\frac{\theta_1^2\beta^-}{2\beta^+\hat{\beta}} h^2,$$

$$\lambda_{1,8} = -\frac{\theta_3(2(1-\theta_2)\beta^- + \theta_3\beta^+)}{2\beta^+\hat{\beta}} h^2, \quad \lambda_{2,8} = -\frac{\theta_3(2\theta_1\beta^- + \theta_3\beta^+)}{2\beta^+\hat{\beta}} h^2,$$

$$\lambda_{1,9} = -\frac{(1-\theta_2)^2\beta^-}{2\beta^+\hat{\beta}} h^2, \quad \lambda_{2,9} = -\frac{-1 + 2\theta_2 + (1-\theta_2)^2\beta^-/\hat{\beta}}{2\beta^+} h^2,$$

where

$$\hat{\beta} := (1 + \theta_1 - \theta_2)\beta^- + (-\theta_1 + \theta_2)\beta^+, \quad \theta_3 := \theta_2 - \theta_1.$$

References

[1] L. Adams, T. Chartier, New geometric immersed interface multigrid solvers, SIAM J. Sci. Comput. 25 (2004) 1516–1533.
[2] L. Adams, T. Chartier, A comparison of algebraic multigrid and geometric immersed interface multigrid methods for interface problems, SIAM J. Sci. Comput. 26 (2005) 762–784.
[3] L. Adams, Z. Li, The immersed interface/multigrid methods for interface problems, SIAM J. Sci. Comput. 24 (2002) 463–479.
[4] P. Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, J. Comput. Phys. 197 (2004) 364–386.
[5] W. Briggs, V. Henson, S. McCormick, A Multigrid Tutorial, SIAM, Philadelphia, 2000.
[6] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math. 79 (1998) 175–202.
[7] I. Chern, Y. Shu, A coupling interface method for elliptic interface problems, J. Comput. Phys. 225 (2007) 2138–2174.

[8] J. Demmel, S. Eisenstat, J. Gilbert, X. Li, J. Liu, A supernodal approach to sparse partial pivoting, SIAM J. Matrix Anal. Appl. 20 (3) (1999) 720–755.

[9] A. Fogelson, J. Keener, Immersed interface methods for Neumann and related problems in two and three dimensions, SIAM J. Sci. Comput. 22 (2000) 1630–1654.

[10] S. Hou, X. Liu, A numerical method for solving variable coefficient elliptic equation with interfaces, J. Comput. Phys. 202 (2005) 411–445.

[11] H. Huang, Z. Li, Convergence analysis of the immersed interface method, IMA J. Numer. Anal. 19 (1999) 583–608.

[12] J. Langer, Instabilities and pattern formation in crystal growth, Rev. Mod. Phys. 52 (1980) 1–28.

[13] R. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.

[14] Z. Li, A fast iterative algorithm for elliptic interface problems, SIAM J. Numer. Anal. 35 (1998) 230–254.

[15] Z. Li, K. Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, SIAM.J. Sci. Comput. 23 (2001) 339–361.

[16] Z. Li, K. Ito, The Immersed Interface Method, SIAM, Philadelphia, 2006.

[17] K. Morton, D. Mayers, Numerical Solution of Partial Differential Equations, Cambridge, 2005.

[18] C. Oosterlee, T. Washio, An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems, SIAM J. Sci. Comput. 19 (1998) 87–110.

[19] C. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.

[20] J. Ruge, K. Stüben, Algebraic multigrid, Multigrid Methods, Front. Appl. Math., vol. 3, SIAM, Philadelphia, 1987.

[21] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, Philadelphia, 2003.

[22] Y. Saad, M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, J. Sci. Stat. Comput. 7 (1986) 856–869.

[23] J. Strain, Locally-corrected spectral methods and overdetermined elliptic systems, J. Comput. Phys. 224 (2007) 1243–1254.

[24] K. Stüben, A review of algebraic multigrid, J. Comput. Appl. Math. 128 (2001) 281–309.

[25] U. Trottenberg, C. Oosterlee, A. Schuller, Multigrid, Academic Press, Toronto, 2001.

[26] H. Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, J. Sci. Stat. Comput. 13 (1992) 631–644.

[27] J. Wan, X. Liu, A boundary condition-capturing multigrid approach to irregular boundary problems, SIAM J. Sci. Comput. 25 (6) (2004) 1982–2003.

[28] A. Wiegmann, K. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (2000) 827–862.

[29] S. Yu, G. Wei, Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces, J. Comput. Phys. 224 (2007) 729–756.

[30] S. Yu, G. Wei, Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities, J. Comput. Phys. 227 (2007) 602–632.

[31] S. Yu, Y. Zhou, G. Wei, Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces, J. Comput. Phys. 224 (2007) 729–756.

[32] X. Zhong, A new high-order immersed interface methods for solving elliptic equations with imbedded interface of discontinuity, J. Comput. Phys. 225 (2007) 1066–1099.

[33] Y. Zhou, G. Wei, On the fictitous-domain and interpolation formulations of the matched interface and boundary (MIB) method, J. Comput. Phys. 219 (2006) 228–246.

[34] Y. Zhou, S. Zhao, M. Feig, G. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. 213 (2006) 1–30.