An Error Minimized Pseudospectral Penalty Direct Poisson Solver

by
Horng, Tzuy-Leng
Dept. of Applied Mathematics
Feng Chia University, Taichung, Taiwan
tlhorng123@gmail.com
http://newton.math.fcu.edu.tw/~tlhorng

and
Teng, Chun-Hao
Dept. of Applied Mathematics
National Chiao Tung University, Hsin-Chu, Taiwan
chunhao.teng@gmail.com
Abstract

This paper presents a direct Poisson solver based on an error minimized Chebyshev pseudospectral penalty formulation for problems defined on rectangular domains. The current method is based on eigenvalue-eigenvector matrix diagonalization methods developed long time before and elaborated by Chen, Su, and Shizgal (2000) [1]. However, this kind of fast Poisson solvers utilizing pseudospectral discretization is usually restricted to periodic or Dirichlet boundary conditions only for efficient implementation. Enforcement of Neumann or Robin boundary conditions requests messy row operations that reduces the easiness of operation. The current method can easily accommodate all three types of boundary conditions: Dirichlet, Neumann, and Robin boundary conditions. The reason for that is that it employs penalty method. Besides, the penalty parameters are determined analytically such that the discrete L2 error is minimized. 2D and 3D numerical experiments are conducted and the results show that the penalty scheme computes numerical solutions with better accuracy, compared to the traditional approaches with boundary conditions enforced strongly. The current method can be extended to generally linear 2nd order elliptic-type partial differential boundary value problems with arbitrary coefficients with only one restriction that those arbitrary coefficients must be able to be separable to coordinate-variable functions. Under this generalization, this method can be surely applied to solve famous Helmholtz equation too. As to computing efficiency, the asymptotic operation count for current method in 3D case is $2N_xN_yN_z (N_x + N_y + N_z)$, and its counterpart for 2D case is $2N_xN_y(N_x + N_y)$. Obviously, it is far superior to method expressing Laplace operator in tensor product. For an FFT-based method the asymptotic operation count is basically $2N_xN_yN_z \log(N_x) + \log(N_y) + \log(N_z)$ for 3D case, and $2N_xN_y\log(N_x) + \log(N_y)$ for 2D case. Indeed, the current method which relies on extensive matrix-matrix multiplications is inferior to FFT-based methods in theory. However, this inferiority also depends on hardware. For moderate $N_x$, $N_y$, and $N_z$, it is not necessarily inferior in computers nowadays. Of course, for large grid resolution, FFT-based methods remain the the best choice if handling boundary conditions is not a problem.

Approximation by Chebyshev polynomials

(1) Suppose that $u(x)$ is approximated by a truncated series of Chebyshev polynomials as

$$u(x) = \sum_{k=0}^{N} \hat{u}_k T_k(x),$$

where $T_k(x)$ denotes Chebyshev polynomial and $\hat{u}_k$ expansion coefficient.

(2) Choosing the Chebyshev Gauss-Lobatto collocations points:

$$x_j = \cos \frac{j\pi}{N}, \quad j = 0, 1, \ldots, N,$$

(3) We can form the following discrete transform and inverse transform

$$\tilde{u}_k = \frac{2}{\bar{c}_k N} \sum_{j=0}^{N} \frac{1}{\bar{c}_j} u(x_j) T_k(x_j),$$

$$u(x_j) = \sum_{k=0}^{N} \tilde{u}_k T_k(x_j),$$

where $\bar{c}_j = \begin{cases} 2, & j = 0, N \\ 1, & \text{otherwise} \end{cases}$ and $\bar{c}_k = \begin{cases} 2, & k = 0, N \\ 1, & \text{otherwise} \end{cases}$. 
(4) For pseudospectral method, interpolating $u(x)$ at the collocation points $x_j$, $j = 0, 1, \ldots, N$.

(5) Hence, $u(x)$ can be also expressed as

$$u(x) = \sum_{k=0}^{N} L_{N,k}(x)u(x_k),$$

where $L_{N,k}(x)$ is the Lagrange interpolating polynomial defined as

$$L_{N,k}(x) = \prod_{l=0 \atop l \neq k}^{N} \frac{x - x_l}{x_k - x_l} = (-1)^{N+k+1} \frac{(1-x)^2 T_N'(x)}{c_k N^2 (x - x_k)}.$$

(6) Then, the derivatives $\frac{du}{dx}(x_j)$ and $\frac{d^2u}{dx^2}(x_j)$ can be approximated as

$$\frac{du}{dx}(x_j) = \sum_{k=0}^{N} u(x_k) \frac{dL_{N,k}}{dx}(x_j) \quad \text{and} \quad \frac{d^2u}{dx^2}(x_j) = \sum_{k=0}^{N} u(x_k) \frac{d^2L_{N,k}}{dx^2}(x_j),$$

where $\frac{dL_{N,k}}{dx}(x_j)$ is usually referred to the Chebyshev collocation derivative matrix.
Chebyshev Collocation Derivative Matrix

(1) The entries of first-order Chebyshev collocation derivative matrix with respect to \( x \), based on Gauss-Lobatto collocation points, are given as below:

\[
(D_x)_{00} = \frac{2N_x^2 + 1}{6}, \quad (D_x)_{N_x,N_x} = -\frac{2N_x^2 + 1}{6},
\]

\[
(D_x)_{jj} = \frac{-x_j}{2(1-x_j^2)} , \quad j = 1, 2, \ldots, N_x - 1,
\]

where \( c_i = \begin{cases} 2 & \text{for } i = 0 \text{ or } N_x, \\ 1 & \text{otherwise} \end{cases} \)

\[
(D_x)_{ij} = \frac{c_i (-1)^{i+j}}{c_j (x_i - x_j)} , \quad i \neq j, \text{ and } i, j = 0, 1, 2, \ldots, N_x,
\]

here \( x_j = \cos \left( \frac{j\pi}{N_x} \right) , \quad j = 0, 1, 2, \ldots, N_x \) are Chebyshev Gauss-Lobatto collocation points. Also, there is a version for avoiding round-off error when \( N_x \) is large.


(2) \( D_{xx} = D_x^2 \).
Chebyshev Gauss-Lobatto collocation mesh

$N_x = N_y = 12$
Example: 1D Poisson equation (PE) with Dirichlet BC’s

(1) Consider an 1D Poisson equation with Dirichlet boundary conditions as follows,
\[ u_{xx} = f, \quad x \in [-1,1], \quad u(-1) = g_{-} \text{ and } u(1) = g_{+}. \]

(2) Discretization: (The first/last row/column of \( D_{xx} \) can be stripped due to Dirichlet boundary conditions.)

\[
\begin{bmatrix}
(D_{xx})_{ij}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
\vdots \\
u_{N_x-2} \\
u_{N_x-1}
\end{bmatrix}
= \begin{bmatrix}
f_1 - (D_{xx})_{1,0}g_{+} - (D_{xx})_{1,N_x}g_{-} \\
f_2 - (D_{xx})_{2,0}g_{+} - (D_{xx})_{2,N_x}g_{-} \\
\vdots \\
f_{N_x-2} - (D_{xx})_{N_x-2,0}g_{+} - (D_{xx})_{N_x-2,N_x}g_{-} \\
f_{N_x-1} - (D_{xx})_{N_x-1,0}g_{+} - (D_{xx})_{N_x-1,N_x}g_{-}
\end{bmatrix},
\]

where \( (D_{xx})_{ij} = (D_{xx})_{ij}, i, j = 1, 2, \ldots, N_x - 1. \)

(3) Exponential order of convergence.
**Example: 2D Poisson equation with Dirichlet BC’s**

1. Consider a 2D Poisson problem with Dirichlet boundary conditions in a rectangular domain as follows,

\[ u_{xx} + u_{yy} = f, \quad (x, y) \in [-1,1] \times [-1,1], \]
\[ u(-1, y) = g^-_x(y), \quad u(1, y) = g^+_x(y), \]
\[ u(x, -1) = g^-_y(x), \quad u(x, 1) = g^+_y(x). \]

2. Discretizing \((N_x + 1) \times (N_y + 1)\) collocation points in the \(x\) and \(y\) directions for \(u\) respectively, and keeping \(u\) in the shape of a rectangular matrix:

\[ D_{yy} u + \left( D_{xx} u^t \right)^t = D_{yy} u + uD_{xx}^t = f, \quad \text{with} \quad u, f \in R^{(N_y+1) \times (N_x+1)}. \]

3. The first and last rows and columns of \(D_{xx}\) and \(D_{yy}\) can be stripped due to Dirichlet BCs, then we can obtain

\[ \overline{D}_{yy} \overline{u} + \overline{u} \overline{D}_{xx}^t = \overline{f}, \quad \text{where} \quad \overline{u}, \overline{f} \in R^{(N_y-1) \times (N_x-1)}, \]

where \((\overline{D}_{yy})_{ij} = (D_{yy})_{ij}, i, j = 1, 2, \ldots, N_y - 1, (\overline{D}_{xx})_{ij} = (D_{xx})_{ij}, i, j = 1, 2, \ldots, N_x - 1, \]
\[ \overline{f}_{ij} = f_{ij} - (D_{yy})_{i0} \left( g^+_y(x_j) \right) - (D_{yy})_{iN_y} \left( g^-_y(x_j) \right) - (D_{xx})_{j0} \left( g^+_x(y_i) \right) - (D_{xx})_{jN_x} \left( g^-_x(y_i) \right), \quad \text{with} \]
\[ i = 1, 2, \ldots, N_y - 1; \quad j = 1, 2, \ldots, N_x - 1. \]
References for Diagonalization Method (Dirichlet BC’s)


(7) S. Zhao and M. J. Yedlin, A new iterative Chebyshev spectral method for solving the elliptic equation $\nabla \cdot (\sigma \nabla u) = f$, J. Comput. Phys. 113 (1994), 215-223. (2D)


(9) Y. Su, Collocation spectral methods in the solution of Poisson equation, MS thesis, University of British Columbia, Canada, 1998. (detailed review and 2D derivation)


Diagonalization via Eigenvector

(1) \( \bar{D}_{yy} e_{y_i} = \lambda_i e_{y_i} \Rightarrow \bar{D}_{yy} Y = Y \Lambda \), where \( \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots & \\ & & & \lambda_{N_y - 1} \end{bmatrix} \),

here \( \lambda_i \) and \( e_{y_i} \) represent the eigenvalue and eigenvector of \( \bar{D}_{yy} \) and \( Y = \begin{bmatrix} e_{y_1}, e_{y_2}, \ldots, e_{y_{N_y - 1}} \end{bmatrix} \).

(2) Similarly,

\( \bar{D}_{xx}^t e_{x_j} = \sigma_j e_{x_j} \Rightarrow \bar{D}_{xx}^t X = X \Sigma \), where \( \Sigma = \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots & \\ & & & \sigma_{N_x - 1} \end{bmatrix} \).
\[ \bar{D}_{yy}\bar{u} + \bar{u}\bar{D}_{xx}' = \bar{f} \]
\[ \bar{D}_{yy}\bar{u}X + \bar{u}\bar{D}_{xx}' X = \bar{f}X \]
\[ \bar{D}_{yy}\bar{u}X + \bar{u}X\Sigma = \bar{f}X \]
\[ \bar{D}_{yy}YY^{-1}\bar{u}X + \bar{u}X\Sigma = \bar{f}X \]
\[ Y\Lambda Y^{-1}\bar{u}X + \bar{u}X\Sigma = \bar{f}X \]
\[ \Lambda Y^{-1}\bar{u}X + Y^{-1}\bar{u}X\Sigma = Y^{-1}\bar{f}X \]
\[ \Lambda V + V\Sigma = B \quad \text{where } \ V = Y^{-1}\bar{u}X \text{ and } B = Y^{-1}\bar{f}X \]
i.e. \[ \left( \lambda_i v_{ij} + v_{ij} \sigma_j \right) = b_{ij}, \ i = 1, 2, \ldots, N_y - 1 \text{ and } j = 1, 2, \ldots, N_x - 1, \]
\[ \therefore v_{ij} = \frac{b_{ij}}{\lambda_i + \sigma_j} \quad \text{where } v_{ij} \text{ and } b_{ij} \text{ are the elements of } V \text{ and } B, \text{ respectively.} \]

Once \( V \) is solved, \( \bar{u} \) can be recovered from \( \bar{u} = YVX^{-1} \). The diagonalization method so far only works easily for Dirichlet BC's. For non-Dirichlet BC's, such as Neumann and Robin BC's, it requests messy row operations.
Example: 1D Poisson equation with BC’s involving derivative

(1) Consider an 1D Poisson equation with Dirichlet/Neumann boundary conditions as follows,
\[ u_{xx} = f, \quad x \in [-1,1], \quad u(-1) = g_{-} \quad \text{and} \quad u_x(1) = g_{+}. \]

(2) Stripping off first/last row/column of \( D_{xx} \) no longer works.

(3) Discretization:

\[
\begin{bmatrix}
\hat{D}_{xx}
\end{bmatrix}_{ij}
\begin{bmatrix}
u_0 \\
u_1 \\
\vdots \\
u_{N_x-1} \\
u_{N_x}
\end{bmatrix}
=
\begin{bmatrix}
g_+ \\
f_1 \\
\vdots \\
f_{N_x-1} \\
g_{-}
\end{bmatrix}, \quad \text{where} \quad \hat{D}_{xx} =
\begin{bmatrix}
(D_x)_{0,j} \\
\vdots \\
(D_x)_{N_x-1,j} \\
I_{N_x,j}
\end{bmatrix}_{(N_x+1)\times(N_x+1)}
\]

here \( I_{N_x,j} \) stands for the last row of the \((N_x+1)\times(N_x+1)\) identity matrix.
Example: 2D Poisson equation with BC’s involving derivative

(1) Consider a 2D Poisson equation with Dirichlet/Neumann boundary conditions as follows,

\[ u_{xx} + u_{yy} = f, \quad (x, y) \in [-1, 1] \times [-1, 1], \]

\[ u(-1, y) = g_x^-(y), \quad u_x(1, y) = g_x^+(y), \]

\[ u(x, -1) = g_y^-(x), \quad u_y(x, 1) = g_y^+(x). \]

(2) Following the idea of 1D case, discretizing \((N_x + 1) \times (N_y + 1)\) collocation points in \(x\) and \(y\) directions respectively for \(u\), and keeping \(u\) in the form of a rectangular matrix:

\[ \hat{D}_{yy} u + u \hat{D}_{xx}^t = \hat{f}, \quad u, \hat{f} \in R^{(N_x + 1) \times (N_y + 1)}, \]

where \(\hat{D}_{yy} = \begin{bmatrix} \left(D_y\right)_{0,j} & \cdots & \left(D_y\right)_{N_y,j} \\ \left(D_{yy}\right)_{ij} & \cdots & \left(D_{yy}\right)_{N_y,j} \\ I_{N_y,j} & \cdots & I_{N_y,j} \end{bmatrix}_{(N_y + 1) \times (N_y + 1)}\), 

\[ \hat{D}_{xx} = \begin{bmatrix} \left(D_x\right)_{0,j} & \cdots & \left(D_x\right)_{N_x,j} \\ \left(D_{xx}\right)_{ij} & \cdots & \left(D_{xx}\right)_{N_x,j} \\ I_{N_x,j} & \cdots & I_{N_x,j} \end{bmatrix}_{(N_x + 1) \times (N_x + 1)}\],

\[ \hat{f} = \begin{bmatrix} g_x^+ \text{ or } g_y^- \\ g_x^+ \\ \vdots \\ g_x^+ \\ g_y^+ \text{ or } g_y^- \\ g_y^+ \\ \vdots \\ g_y^+ \end{bmatrix}_{(N_y + 1) \times (N_x + 1)} \]

\[ f_{ij} = \begin{bmatrix} g_x^- \\ g_x^- \\ \vdots \\ g_x^- \\ g_y^+ \text{ or } g_y^- \\ g_y^+ \\ \vdots \\ g_y^+ \end{bmatrix}_{(N_y + 1) \times (N_x + 1)} \], \quad \text{here } \begin{cases} i = 1, 2, \ldots, N_y - 1 \\ j = 1, 2, \ldots, N_x - 1 \end{cases}.
(3) However, using $5 \times 5$ matrix as an example, $\hat{D}_{yy} u + u \hat{D}_{xx}^t = \hat{f}$ will become

$$
\begin{bmatrix}
  u_y + u_x & u_{xx} + u_y & u_{xx} + u_y & u_{xx} + u_y & u_y + u \\
  u_{yy} + u_x & \Delta u & \Delta u & \Delta u & u_{yy} + u \\
  u_{yy} + u_x & \Delta u & \Delta u & \Delta u & u_{yy} + u \\
  u_{yy} + u_x & \Delta u & \Delta u & \Delta u & u_{yy} + u \\
  u + u_x & u_{xx} + u & u_{xx} + u & u_{xx} + u & u + u
\end{bmatrix}
$$

\[
\begin{bmatrix}
g_+^x \text{ or } g_-^y \\
g_+^x \\
g_+^x \\
g_+^x \text{ or } g_-^y
\end{bmatrix} \begin{bmatrix}
f \\
f \\
f \\
f
\end{bmatrix} = \begin{bmatrix}
g_+^x \\
g_+^x \\
g_+^x \\
g_+^x \text{ or } g_-^y
\end{bmatrix},
\]

which is wrong at BC's. How to fix this? Resorting to Kronecker product $I \otimes \hat{D}_{yy} + \hat{D}_{xx} \otimes I$ is both time and memory consuming.
(1) Consider the 2D Poisson equation with all kind of boundary conditions in a rectangular domain, i.e.,

\[ B^x_{\pm} u(\pm1, y) = g^x_{\pm}(y), \quad B^y_{\pm} u(x, \pm1) = g^y_{\pm}(x), \]

where

\[ B^x_{\pm} = \alpha^x_{\pm} \pm \beta^x_{\pm} \partial_x, \quad B^y_{\pm} = \alpha^y_{\pm} \pm \beta^y_{\pm} \partial_y, \]

and \( \alpha^x_{\pm}, \beta^x_{\pm}, \alpha^y_{\pm}, \) and \( \beta^y_{\pm} \) are specified constants. At least one of \( \alpha^x_{\pm} \) and \( \alpha^y_{\pm} \) must be non-zero for a unique solution to exist.

(2) At this time, we can not strip first/last row/column of \( D_{y\gamma} \) and \( D_{\gamma x} \) as we did for the case of Dirichlet BCs. We follow the previous idea but modify the substitution by penalty method shown as belows.
(3) Replace $D_{xx}$ and $D_{yy}$ by the following $\tilde{D}_{xx}$ and $\tilde{D}_{yy}$, respectively

$$
\begin{align*}
\tilde{D}_{xx} &= \\
&= \begin{bmatrix}
(D_{xx})_{0,j} - \tau^x_+ \left( \alpha^x_+ I_{0,j} + \beta^x_+ (D_x)_{0,j} \right) \\
&- (D_{xx})_{ij} \\
&- (D_{xx})_{N_x,j} - \tau^-_x \left( \alpha^-_x I_{N_x,j} - \beta^-_x (D_x)_{N_x,j} \right)
\end{bmatrix}_{(N_x+1) \times (N_x+1)} \\
\tilde{D}_{yy} &= \\
&= \begin{bmatrix}
(D_{yy})_{0,j} - \tau^y_+ \left( \alpha^y_+ I_{0,j} + \beta^y_+ (D_y)_{0,j} \right) \\
&- (D_{yy})_{ij} \\
&- (D_{yy})_{N_y,j} - \tau^-_y \left( \alpha^-_y I_{N_y,j} - \beta^-_y (D_y)_{N_y,j} \right)
\end{bmatrix}_{(N_y+1) \times (N_y+1)}
\end{align*}
$$

where $\tau^x_+$, $\tau^-_x$, $\tau^y_+$, and $\tau^-_y$ are the penalty weights, which are usually large numbers.
(4) Using $5 \times 5$ matrix as an example, $\tilde{D}_{yy} u + u \tilde{D}_{xx}' = \tilde{f}$ will become

$$
\begin{bmatrix}
\Delta u - \tau_x^x B_x^x u - \tau_y^y B_y^y u & \Delta u - \tau_x^y B_x^y u & \Delta u - \tau_y^y B_y^y u & \Delta u - \tau_x^y B_x^y u & \Delta u - \tau_y^x B_x^x u - \tau_y^y B_y^y u \\
\Delta u - \tau_x^x B_x^x u & \Delta u & \Delta u & \Delta u & \Delta u - \tau_x^y B_x^y u \\
\Delta u - \tau_x^x B_x^x u & \Delta u & \Delta u & \Delta u & \Delta u - \tau_y^x B_y^x u \\
\Delta u - \tau_x^x B_x^x u & \Delta u & \Delta u & \Delta u & \Delta u - \tau_y^x B_y^x u \\
\Delta u - \tau_x^x B_x^x u - \tau_y^y B_y^y u & \Delta u - \tau_y^y B_y^y u & \Delta u - \tau_y^y B_y^y u & \Delta u - \tau_y^y B_y^y u & \Delta u - \tau_x^x B_x^x u - \tau_y^y B_y^y u
\end{bmatrix}
$$

$$
\begin{bmatrix}
f - \tau_+^x g_+^x - \tau_+^y g_+^y & f - \tau_+^y g_+^y & f - \tau_+^y g_+^y & f - \tau_+^y g_+^y & f - \tau_-^x g_-^x - \tau_+^y g_+^y \\
f - \tau_+^x g_+^x & f & f & f & f - \tau_-^x g_-^x \\
f - \tau_+^x g_+^x & f & f & f & f - \tau_-^x g_-^x \\
f - \tau_+^x g_+^x & f & f & f & f - \tau_-^x g_-^x \\
f - \tau_+^x g_+^x - \tau_-^y g_-^y & f - \tau_-^y g_-^y & f - \tau_-^y g_-^y & f - \tau_-^y g_-^y & f - \tau_-^x g_-^x - \tau_-^y g_-^y
\end{bmatrix}
$$

which well approximates $\Delta u = f$ with the specified BC's when $\tau$'s are large enough.
Diagonalization via Eigenvector again

(1) \( \tilde{D}_{yy} e_{y_i} = \lambda_i e_{y_i} \Rightarrow \tilde{D}_{yy} Y = Y \Lambda \) where \( \Lambda = \begin{bmatrix} \lambda_0 & & \\ & \lambda_1 & \\ & & \ddots \\ & & & \lambda_{N_y} \end{bmatrix} \)

here \( \lambda_i \) and \( e_{y_i} \) represent the eigenvalue and eigenvector of \( \tilde{D}_{yy} \)

and \( Y = \begin{bmatrix} e_{y_0}, e_{y_1}, \ldots, e_{y_{N_y}} \end{bmatrix} \).

(2) Similarly,

\( \tilde{D}_{xx}^t e_{x_j} = \sigma_j e_{x_j} \Rightarrow \tilde{D}_{xx}^t X = X \Sigma \) where \( \Sigma = \begin{bmatrix} \sigma_0 & & \\ & \sigma_1 & \\ & & \ddots \\ & & & \sigma_{N_x} \end{bmatrix} \).
\[ \tilde{D}_{yy} u + u \tilde{D}_{xx}^t = \tilde{f} \]
\[ \tilde{D}_{yy} uX + u \tilde{D}_{xx}^t X = \tilde{f}X \]
\[ \tilde{D}_{yy} uX + uX \Sigma = \tilde{f}X \]
\[ \tilde{D}_{yy} YY^{-1} uX + uX \Sigma = \tilde{f}X \]
\[ Y \Lambda Y^{-1} uX + uX \Sigma = \tilde{f}X \]
\[ \Lambda Y^{-1} uX + Y^{-1} uX \Sigma = Y^{-1} \tilde{f}X \]
\[ \Lambda V + V \Sigma = B \] where \( V = Y^{-1} uX \) and \( B = Y^{-1} \tilde{f}X \)

i.e. \( \left( \lambda_i v_{ij} + \sigma_j v_{ij} \right) = b_{ij}, \ i = 0,1,\ldots, N_y \) and \( j = 0,1,\ldots, N_x \),

\[ : v_{ij} = \frac{b_{ij}}{\lambda_i + \sigma_j} \] where \( v_{ij} \) and \( b_{ij} \) are the elements of \( V \) and \( B \), respectively.

Once \( V \) is solved, \( u \) can be recovered from \( u = YYX^{-1} \).
3D Case

\[
\tilde{D}_{xx} = \begin{bmatrix}
(D_{xx})_{0,j} - \tau_{+}^{x} \left( \alpha_{+}^{x} I_{0,j} + \beta_{+}^{x} (D_{x})_{0,j} \right)

\end{bmatrix}
\]

\[
\tilde{D}_{yy} = \begin{bmatrix}
(D_{yy})_{0,j} - \tau_{+}^{y} \left( \alpha_{+}^{y} I_{0,j} + \beta_{+}^{y} (D_{y})_{0,j} \right)

\end{bmatrix}
\]

\[
\tilde{D}_{zz} = \begin{bmatrix}
(D_{zz})_{0,j} - \tau_{+}^{z} \left( \alpha_{+}^{z} I_{0,j} + \beta_{+}^{z} (D_{z})_{0,j} \right)

\end{bmatrix}
\]
(2)

$$\sum_{l=0}^{N_y} (\tilde{D}_{yy})_{il} Y_{lm} = \lambda_m Y_{im} \Rightarrow \sum_{i=0}^{N_y} \sum_{l=0}^{N_y} Y_{mi}^{-1} (\tilde{D}_{yy})_{il} Y_{lm} = \lambda_m,$$

$$\sum_{j=0}^{N_x} (\tilde{D}_{xx}^t)_{lj} X_{jn} = \sigma_n X_{ln} \Rightarrow \sum_{l=0}^{N_x} \sum_{j=0}^{N_x} X_{nl}^{-1} (\tilde{D}_{xx}^t)_{lj} X_{jn} = \sigma_n,$$

$$\sum_{k=0}^{N_z} (\tilde{D}_{zz}^t)_{lk} Z_{ko} = \omega_o Z_{lo} \Rightarrow \sum_{l=0}^{N_z} \sum_{k=0}^{N_z} Z_{ol}^{-1} (\tilde{D}_{zz}^t)_{lk} Z_{ko} = \omega_o.$$
\begin{equation}
\sum_{l=0}^{N_y} (\tilde{D}_{yy})_{il} u_{ijk} + \sum_{l=0}^{N_x} u_{ilk} (\tilde{D}_{xx})_{lj} + \sum_{l=0}^{N_z} u_{ijl} (\tilde{D}_{zz})_{lk} = F_{ijk},
\end{equation}

\begin{equation}
\sum_{k=0}^{N_z} \sum_{j=0}^{N_x} \sum_{l=0}^{N_y} (\tilde{D}_{yy})_{il} u_{ljk} X_{jn} Z_{ko} + \sum_{k=0}^{N_z} \sum_{l=0}^{N_x} u_{ilk} \sigma_n X_{ln} Z_{ko} + \sum_{j=0}^{N_x} \sum_{l=0}^{N_z} u_{ijl} \omega_o Z_{lo} X_{jn} = \sum_{k=0}^{N_z} \sum_{j=0}^{N_x} F_{ijk} X_{jn} Z_{ko},
\end{equation}

\begin{equation}
\sum_{k=0}^{N_z} \sum_{j=0}^{N_x} \sum_{l=0}^{N_y} \sum_{m=0}^{N_y} (\tilde{D}_{yy})_{il} \ Y_{lm} Y_{ml}^{-1} u_{ljk} X_{jn} Z_{ko} + \sum_{k=0}^{N_z} \sum_{l=0}^{N_x} u_{ilk} X_{ln} Z_{ko} \sigma_n + \sum_{j=0}^{N_x} \sum_{l=0}^{N_z} u_{ijl} X_{jn} Z_{lo} \omega_o = \sum_{k=0}^{N_z} \sum_{j=0}^{N_x} F_{ijk} X_{jn} Z_{ko},
\end{equation}
\[
\sum_{k=0}^{N_z} \sum_{j=0}^{N_x} \sum_{l=0}^{N_y} Y_{im} \lambda_m Y^{-1} u_{ijk} X_{jn} Z_{ko} + \sum_{k=0}^{N_z} \sum_{l=0}^{N_x} u_{ilk} X_{ln} Z_{ko} \sigma_n
\]

\[
+ \sum_{j=0}^{N_x} \sum_{l=0}^{N_z} u_{ijl} X_{jn} Z_{lo} \omega_o = \sum_{k=0}^{N_z} \sum_{l=0}^{N_x} F_{ijk} X_{jn} Z_{ko},
\]

\[
\sum_{k=0}^{N_z} \sum_{j=0}^{N_x} \sum_{l=0}^{N_y} \lambda_m Y^{-1} u_{ijk} X_{jn} Z_{ko} + \sum_{i=0}^{N_y} \sum_{k=0}^{N_z} \sum_{l=0}^{N_x} Y^{-1} u_{ilk} X_{ln} Z_{ko} \sigma_n
\]

\[
+ \sum_{i=0}^{N_y} \sum_{l=0}^{N_z} \sum_{j=0}^{N_x} Y_{mi} u_{ijl} X_{jn} Z_{lo} \omega_o = \sum_{i=0}^{N_y} \sum_{k=0}^{N_z} \sum_{j=0}^{N_x} Y^{-1} F_{ijk} X_{jn} Z_{ko},
\]
\[ \lambda_m \sum_{i=0}^{N_y} \sum_{k=0}^{N_z} \sum_{j=0}^{N_x} Y^{-1}_{ml} u_{ijk} X_{jn} Z_{ko} + \sigma_n \sum_{i=0}^{N_y} \sum_{k=0}^{N_z} \sum_{l=0}^{N_x} Y^{-1}_{mi} u_{ilk} X_{ln} Z_{ko} \]

\[ + \omega_o \sum_{i=0}^{N_y} \sum_{l=0}^{N_z} \sum_{j=0}^{N_x} Y^{-1}_{mi} u_{ijl} X_{jn} Z_{ko} = \sum_{i=0}^{N_y} \sum_{k=0}^{N_z} \sum_{j=0}^{N_x} Y^{-1}_{mi} F_{ijk} X_{jn} Z_{ko}, \]

Hence, \((\lambda_m + \sigma_n + \omega_o) W_{mno} = B_{mno} \Rightarrow W_{mno} = \frac{B_{mno}}{\lambda_m + \sigma_n + \omega_o}, \ \forall m, n, o,\]

where \( W_{mno} = \sum_{i=0}^{N_y} \sum_{k=0}^{N_z} \sum_{l=0}^{N_x} Y^{-1}_{mi} u_{ilk} X_{ln} Z_{ko}, \)

\( B_{mno} = \sum_{i=0}^{N_y} \sum_{k=0}^{N_z} \sum_{j=0}^{N_x} Y^{-1}_{mi} F_{ijk} X_{jn} Z_{ko} \).

\[ \therefore u_{ijk} = \sum_{m=0}^{N_y} \sum_{o=0}^{N_z} \sum_{n=0}^{N_x} Y_{im} W_{mno} X^{-1}_{nj} Z^{-1}_{ok}. \]
Error Minimization for Penalty Parameter

(1) Taking 1D PE as example, $u''(x) = f(x)$.

(2) $f(x) = -16\pi^2 \sin(4\pi x)$, so exact solution $u(x) = \sin(4\pi x)$.

(3) BC's: (a) $u(-1) = 0$, $u'(1) = 4\pi$, (b) $u'(-1) = 4\pi$, $u(1) + u'(1) = 4\pi$.

Figure 1: BC (a), $N=16$.

Figure (2): BC (b), $N=16$. 
Error-Minimized Tau vs. Strongly Enforcement

Strongly-enforced BC is equivalent to \( \tau \rightarrow \infty \).

**Table 1: BC (a)**

<table>
<thead>
<tr>
<th>( N )</th>
<th>Present method</th>
<th>Strongly enforced BC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( L_2 ) error</td>
<td>( L_\infty ) error</td>
</tr>
<tr>
<td>16</td>
<td>7.5861e–03</td>
<td>8.2458e–03</td>
</tr>
<tr>
<td>20</td>
<td>8.4171e–05</td>
<td>8.9049e–05</td>
</tr>
<tr>
<td>24</td>
<td>4.2296e–07</td>
<td>4.4719e–07</td>
</tr>
<tr>
<td>28</td>
<td>1.169e–09</td>
<td>1.2001e–09</td>
</tr>
<tr>
<td>32</td>
<td>1.5770e–12</td>
<td>1.6215e–12</td>
</tr>
</tbody>
</table>

**Table 2: BC (b)**

<table>
<thead>
<tr>
<th>( N )</th>
<th>Present method</th>
<th>Strongly enforced BC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( L_2 ) error</td>
<td>( L_\infty ) error</td>
</tr>
<tr>
<td>16</td>
<td>7.7124e–03</td>
<td>8.3458e–03</td>
</tr>
<tr>
<td>20</td>
<td>8.7012e–05</td>
<td>9.1708e–05</td>
</tr>
<tr>
<td>24</td>
<td>4.4364e–07</td>
<td>4.7063e–07</td>
</tr>
<tr>
<td>28</td>
<td>1.1853e–09</td>
<td>1.2869e–09</td>
</tr>
<tr>
<td>32</td>
<td>1.7921e–12</td>
<td>1.9433e–12</td>
</tr>
</tbody>
</table>
Error-Minimized Tau Can Be Derived

\[ \tau_+ = \frac{-\left[ \alpha_-^2 + 2(\alpha_- + \beta_-)^2 \right]}{\left[ \alpha_-^2 + 2(\alpha_- + \beta_-)^2 \right]G_+ + \left[ 2(\alpha_- + \beta_-)(\alpha_+ + \beta_+) - \alpha_- \alpha_+ \right]H_+}, \]

\[ \tau_- = \frac{(-1)^N \left[ \alpha_+^2 + 2(\alpha_+ + \beta_+)^2 \right]}{\left[ 2(\alpha_+ + \beta_+)(\alpha_- + \beta_-) - \alpha_+ \alpha_- \right]G_- + \left[ \alpha_+^2 + 2(\alpha_+ + \beta_+)^2 \right]H_-}, \]

\[ G_+ = \frac{\alpha_+}{N^2(N^2 - 4)} - \frac{\beta_+ (2N^2 - 1)}{2N^2(N^2 - 1)}, \quad G_- = -\frac{3\alpha_-}{N^2(N^2 - 1)(N^2 - 4)} + \frac{\beta_-}{2N^2(N^2 - 1)}, \]

\[ H_+ = -\frac{(-1)^{N-1} 3\alpha_-}{N^2(N^2 - 1)(N^2 - 4)} + \frac{(-1)^{N-1} \beta_-}{2N^2(N^2 - 1)}, \quad H_- = \frac{(-1)^{N-1} \alpha_-}{N^2(N^2 - 4)} - \frac{(-1)^{N-1} \beta_- (2N^2 - 1)}{2N^2(N^2 - 1)}. \]

Error-Minimized Tau Can Be Extended to Higher-Dimension PE

Taking 2D PE as an example,
\[ \Delta u = u_{xx} + u_{yy} = f(x, y), \]  
with BC's
\[ B^x_\pm u(\pm 1, y) = g^x_\pm(y), \quad B^y_\pm u(x, \pm 1) = g^y_\pm(x). \]

There exists a unique set \( \{u^x(x, y), u^y(x, y), f^x(x, y), f^y(x, y)\} \) such that
\[ u(x, y) = u^x(x, y) + u^y(x, y), \]  
and \( f(x, y) = f^x(x, y) + f^y(x, y), \) with
\[ \partial_{xx} u^x = f^x, \quad \text{subject to} \quad B^x_\pm u(\pm 1, y) = g^x_\pm(y), \]
\[ \partial_{yy} u^y = f^y, \quad \text{subject to} \quad B^y_\pm u(x, \pm 1) = g^y_\pm(x), \]

So we can calculate error-minimized \( \tau_- \) and \( \tau_+ \) in each direction alone according to previous formula.
Numerical experiments for 2D PE

(1) \( \Delta u = f(x, y) \).

(2) \( f(x, y) = -32\pi^2 \sin(4\pi x) \sin(4\pi y) \), so exact solution is \( u(x, y) = \sin(4\pi x) \sin(4\pi y) \).

(3) BC's: (a) \( u(\pm 1, y) \pm \frac{\partial u(\pm 1, y)}{\partial x} = \pm 4\pi \sin(4\pi y) \),

\[ \frac{\partial u(x, \pm 1)}{\partial y} = 4\pi \sin(4\pi x). \]

<table>
<thead>
<tr>
<th>( N )</th>
<th>Present method</th>
<th>Strongly enforced BC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( L_2 ) error</td>
<td>( L_\infty ) error</td>
</tr>
<tr>
<td>16</td>
<td>7.73e-03</td>
<td>5.20e-03</td>
</tr>
<tr>
<td>20</td>
<td>9.13e-05</td>
<td>6.89e-05</td>
</tr>
<tr>
<td>24</td>
<td>4.96e-07</td>
<td>3.82e-07</td>
</tr>
<tr>
<td>28</td>
<td>2.22e-12</td>
<td>1.95e-12</td>
</tr>
<tr>
<td>32</td>
<td>7.30e-14</td>
<td>9.41e-14</td>
</tr>
<tr>
<td>48</td>
<td>4.06e-14</td>
<td>5.65e-14</td>
</tr>
<tr>
<td>64</td>
<td>8.73e-14</td>
<td>6.63e-14</td>
</tr>
</tbody>
</table>
BC's: \((b) \; u(\pm1, y) = 0, \; u(x, \pm1) = 0.\)

<table>
<thead>
<tr>
<th>N</th>
<th>Present method</th>
<th>CT</th>
<th>CC</th>
<th>CG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(L^2) error</td>
<td>(L^\infty) error</td>
<td>(L^\infty) error</td>
<td>(L^\infty) error</td>
</tr>
<tr>
<td>16</td>
<td>6.89e-03</td>
<td>5.25e-03</td>
<td>3.33e-02</td>
<td>5.25e-03</td>
</tr>
<tr>
<td>20</td>
<td>8.54e-05</td>
<td>6.26e-05</td>
<td>7.52e-05</td>
<td>4.05e-07</td>
</tr>
<tr>
<td>24</td>
<td>4.75e-07</td>
<td>3.76e-07</td>
<td>6.89e-06</td>
<td>4.05e-07</td>
</tr>
<tr>
<td>32</td>
<td>2.17e-12</td>
<td>1.78e-12</td>
<td>4.77e-11</td>
<td>2.87e-12</td>
</tr>
<tr>
<td>40</td>
<td>1.24e-14</td>
<td>2.05e-14</td>
<td>1.90e-12</td>
<td>3.63e-12</td>
</tr>
<tr>
<td>48</td>
<td>9.40e-15</td>
<td>1.03e-14</td>
<td>8.67e-13</td>
<td>3.90e-12</td>
</tr>
<tr>
<td>64</td>
<td>3.15e-14</td>
<td>5.05e-14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


BC's: (c) \( u(-1, y) = 0, \quad \frac{\partial u(1, y)}{\partial x} = 4\pi \sin(4\pi y), \)

\( u(x, -1) = 0, \quad u(x, 1) + \frac{\partial u(x, 1)}{\partial y} = 4\pi \sin(4\pi x). \)

<table>
<thead>
<tr>
<th>( N )</th>
<th>Present method</th>
<th>Strongly enforced BC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( L_2 ) error</td>
<td>( L_\infty ) error</td>
</tr>
<tr>
<td>16</td>
<td>7.30e-03</td>
<td>5.28e-03</td>
</tr>
<tr>
<td>20</td>
<td>8.84e-05</td>
<td>6.75e-05</td>
</tr>
<tr>
<td>24</td>
<td>4.87e-07</td>
<td>3.80e-07</td>
</tr>
<tr>
<td>28</td>
<td>2.21e-12</td>
<td>1.93e-12</td>
</tr>
<tr>
<td>32</td>
<td>1.59e-14</td>
<td>1.94e-14</td>
</tr>
<tr>
<td>48</td>
<td>2.08e-14</td>
<td>2.74e-14</td>
</tr>
<tr>
<td>64</td>
<td>3.75e-14</td>
<td>4.24e-14</td>
</tr>
</tbody>
</table>
Numerical experiment for 3D PE

(1) $\Delta u = f(x, y, z)$.

(2) $f(x, y, z) = -48\pi^2 \sin(4\pi x)\sin(4\pi y)\sin(4\pi z)$, so exact solution is $u(x, y) = \sin(4\pi x)\sin(4\pi y)\sin(4\pi z)$.

(3) BC's: $u(-1, y, z) = 0$, \( \frac{\partial u(1, y, z)}{\partial x} = 4\pi \sin(4\pi y)\sin(4\pi z) \),
\( \frac{\partial u(x, -1, z)}{\partial y} = 4\pi \sin(4\pi x)\sin(4\pi z) \), \( u(x, 1, z) + \frac{\partial u(x, 1, z)}{\partial y} = 4\pi \sin(4\pi x)\sin(4\pi z) \),
\( u(x, y, -1) - \frac{\partial u(x, y, -1)}{\partial z} = -4\pi \sin(4\pi x)\sin(4\pi y) \), \( \frac{\partial u(x, y, 1)}{\partial z} = 4\pi \sin(4\pi x)\sin(4\pi y) \).

<table>
<thead>
<tr>
<th>$N$</th>
<th>Present method</th>
<th>Strongly enforced BC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_2$ error</td>
<td>$L_\infty$ error</td>
</tr>
<tr>
<td>16</td>
<td>9.0627e−03</td>
<td>5.4835e−03</td>
</tr>
<tr>
<td>20</td>
<td>1.1572e−04</td>
<td>8.7926e−05</td>
</tr>
<tr>
<td>24</td>
<td>6.5795e−07</td>
<td>5.0986e−07</td>
</tr>
<tr>
<td>28</td>
<td>3.0727e−12</td>
<td>2.6552e−12</td>
</tr>
<tr>
<td>32</td>
<td>2.0128e−14</td>
<td>2.8533e−14</td>
</tr>
<tr>
<td>48</td>
<td>2.5304e−14</td>
<td>3.2713e−14</td>
</tr>
<tr>
<td>64</td>
<td>8.4110e−14</td>
<td>1.1653e−13</td>
</tr>
</tbody>
</table>
Conclusion

(1) Asymptotic operation account:

2D: \(2N_xN_y(N_x + N_y)\), 3D: \(2N_xN_yN_z(N_x + N_y + N_z)\), compared with FFT-based method,
2D: \(2N_xN_y(\log(N_x) + \log(N_y))\), 3D: \(2N_xN_yN_z(\log(N_x) + \log(N_y) + \log(N_z))\).

(2) This method can be easily extended to the following type of elliptic problem subjected to all kinds of BC's in a rectangular domain:

\[
(L_x + L_y + L_z)u = f(x, y, z),
\]

where \(L_x = a_1(x)\frac{\partial^2}{\partial x^2} + b_1(x)\frac{\partial}{\partial x} + c_1(x)\), \(L_y = a_2(y)\frac{\partial^2}{\partial y^2} + b_2(y)\frac{\partial}{\partial y} + c_2(y)\), \(L_z = a_3(z)\frac{\partial^2}{\partial z^2} + b_3(z)\frac{\partial}{\partial z} + c_3(z)\).

Sure, it includes Helmholtz equation.

(3) So far, analytic formula for error-minimized tau can be only derived for PE.

(4) For all BC's being Neumann in PE, which has infinitely many solutions and the difference of any two of them will be a constant, the current method will give you one of the solutions.

(5) The basis function is not limited to Chebyshev polynomials. It can be replaced by Legendre polynomials or any other basis function as far as there is an associated collocation derivative matrix. It can be applied to finite difference too.

Thank you for your attention.
Any question?