An Error Minimized Pseudospectral Penalty Direct Poisson Solver

by

Horng, Tzyy-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 2NxNyNz (Nx + Ny + Nz), and its counterpart for 2D case is 2NxNy(Nx + Ny). Obviously, it is far superior to method expressing Laplace operator in tensor product. For an FFT-based method the asymptotic operation count is basically 2NxNyNz (log(Nx) + log(Ny) + log(Nz)) for 3D case, and 2NxNy(log(Nx) + log(Ny)) 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 Nx, Ny, and Nz, it is not necessarily inferior in computers nowadays. Of course, for large grid resolution, FFTbased methods remain the the best choice if handling boundary conditions is not a problem.

[1] H. CHEN AND Y. SU AND B.D. SHIZGAL. A Direct Spectral Collocation Poisson Solver in Polar and Cylinder Coordinates. J. Comput. Phys. 160 (2000) 453-469.

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

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

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

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

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:

$$\left(D_x\right)_{00} = \frac{2N_x^2+1}{6}, \quad \left(D_x\right)_{N_x,N_x} = -\frac{2N_x^2+1}{6},$$

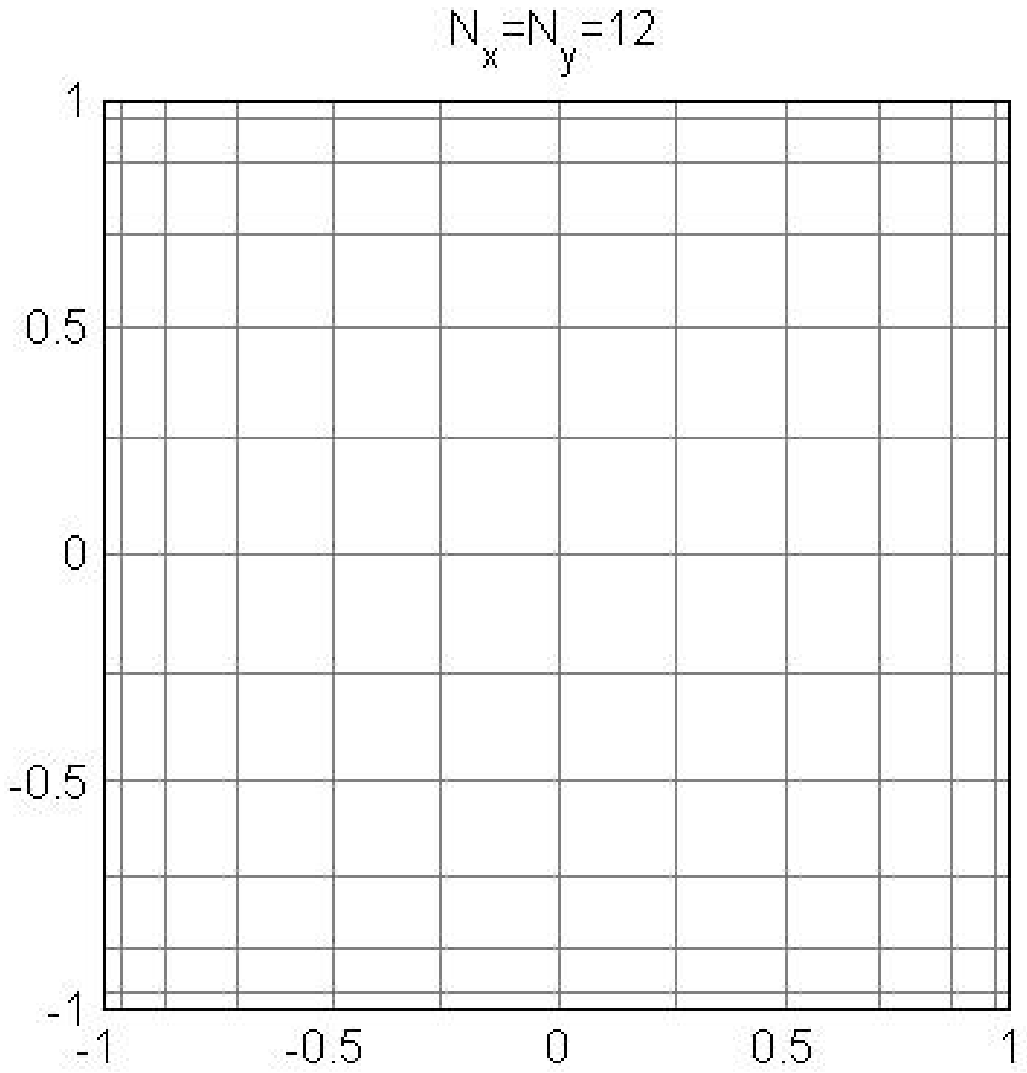
$$\left(D_x\right)_{jj} = \frac{-x_j}{2\left(1-x_j^2\right)}, \ j = 1,2,\ldots,N_x-1, \qquad \text{where } c_i = \begin{cases} 2 & \text{for } i = 0 \text{ or } N_x, \\ 1 & \text{otherwise} \end{cases}$$

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

here $x_j = \cos\left(\frac{j\pi}{N_x}\right)$, $j = 0,1,2,\ldots,N_x$ are Chebyshev Gauss-Lobatto collocation points. Also, there is a version for avoiding round-off error when N_x is large.
(W. Don and S. Solomonoff in SIAM J. Sci. Comp. Vol. 6, pp. 1253--1268 (1994))

(2) $D_{xx} = D_x^2$.

Chebyshev Gauss-Lobatto collocation mesh



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.)

$$\begin{bmatrix} & & \\ & \left(\bar{D}_{xx}\right)_{ij} & \\ & & \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N_x-2} \\ u_{N_x-1} \end{bmatrix} = \begin{bmatrix} f_1 - \left(D_{xx}\right)_{1,0} g_+ - \left(D_{xx}\right)_{1,N_x} g_- \\ f_2 - \left(D_{xx}\right)_{2,0} g_+ - \left(D_{xx}\right)_{2,N_x} g_- \\ \vdots \\ f_{N_x-2} - \left(D_{xx}\right)_{N_x-2,0} g_+ - \left(D_{xx}\right)_{N_x-2,N_x} g_- \\ f_{N_x-1} - \left(D_{xx}\right)_{N_x-1,0} g_+ - \left(D_{xx}\right)_{N_x-1,N_x} g_- \end{bmatrix},$$

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

(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 shown as follows,

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

(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:

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

(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

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

where $\left(\bar{D}_{yy}\right)_{ij}=\left(D_{yy}\right)_{ij}$, $i,j=1,2,\ldots,N_y-1$, $\left(\bar{D}_{xx}\right)_{ij}=\left(D_{xx}\right)_{ij}$, $i,j=1,2,\ldots,N_x-1$,

$\bar{f}_{ij}=f_{ij}-\left(D_{yy}\right)_{i0}\left(g_{+}^{y}(x_j)\right)-\left(D_{yy}\right)_{iN_y}\left(g_{-}^{y}(x_j)\right)-\left(D_{xx}\right)_{j0}\left(g_{+}^{x}(y_i)\right)-\left(D_{xx}\right)_{jN_x}\left(g_{-}^{x}(y_i)\right)$, with $i=1,2,\ldots,N_y-1;\quad j=1,2,\ldots,N_x-1.$

References for Diagonalization Method (Dirichlet BC's)

(1) D. B. Haidvogel, Thomas Zang, The accurate solution of Poisson's equation by expansion in Chebyshev polynomials, J. Comput. Phys. 30 (1979), 167-180. (2D diagonalization under spectral method)

(2) P. Haldenwang, G. Labrosse, and S. Abboudi, Chebyshev 3-D spectral and 2-D pseudospectral solvers for the Helmholtz equation, J. Comput. Phys., 55 (1984), 115-128.

(3) Hwar C. Ku, Richard S. Hirsh, and Thomas D. Taylor, A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations, J. Comput. Phys. 10 (1987), 439-462. (earliest, 2D)

(4) D. Funaro and D. Gottlieb, A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations, Mathematics of Computation, 51:184 (1988), 599-613. (penalty method analysis)

(5) U. Ehrenstein, R. Peyret, A Chebyshev collocation method for the Navier-Stokes equations with application to double-diffusive convection, International Journal for Numerical Methods in Fluids 9 (1989), 427-452. (2D Helmholtz)

(6) Henry H. Yang, Bernie Shizgal, Chebyshev pseudospectral multi-domain technique for viscous flow calculation, Comput. Methods Appl. Mech. Engrg. 118 (1994), 47-61. (2D Helmholtz, Neumann BC)

(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)

(8) H. B. Chen, K. Nandakumar, W. H. Finlay, and H. C. Ku, Three-dimensional viscous flow through a rotating channel: a pseudospectral matrix method approach, 23 (1996): 379-396. (3D Helmholtz, Dirichlet BC)

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

(10) Heli Chen, Yuhong Su, and B. D. Shizgal, A direct spectral collocation Poisson solver in polar and cylindrical coordinates, J. Comput. Phys. 160 (2000), 453-469. (2D/3D cylindrical coordinate and symbolic derivation)

(11) Roger Peyret, Spectral methods for incompressible viscous flow, Springer-Verlag, New York, 2002. (2D, p 88)

(12) S. Nguyen, C. Delcarte, A spectral collocation method to solve Helmholtz problems with boundary conditions involving mixed tangential and normal derivatives, J. Comput. Phys. 200 (2004), 34-49. (complicated Helmholtz solver, single/multi-block)

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 λ_i and e_{y_i} represent the eigenvalue and eigenvector of $\bar{D}_{yy}$

and $Y = \left[e_{y_1}, e_{y_2}, \ldots, e_{y_{N_y - 1}} \right].$

(2) Similarly,

$$\bar{D}^t_{xx} e_{x_j} = \sigma_j e_{x_j} \Rightarrow \bar{D}^t_{xx} X = X\Sigma, \text{ where } \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_{N_x - 1} \end{bmatrix}.$$

$$\bar{D}_{yy}\bar{u} + \bar{u}\bar{D}_{xx}^{t} = \bar{f}$$

$$\bar{D}_{yy}\bar{u}X + \bar{u}\bar{D}_{xx}^{t}X = \bar{f}X$$

$$\bar{D}_{yy}\bar{u}X + \bar{u}X\Sigma = \bar{f}X$$

$$\bar{D}_{yy}YY^{-1}\bar{u}X + \bar{u}X\Sigma = \bar{f}X$$

$$Y\Lambda Y^{-1}\bar{u}X + \bar{u}X\Sigma = \bar{f}X$$

$$\Lambda Y^{-1}\bar{u}X + Y^{-1}\bar{u}X\Sigma = Y^{-1}\bar{f}X$$

$\Lambda V + V\Sigma = B$ where $V = Y^{-1}\bar{u}X$ and $B = Y^{-1}\bar{f}X$

i.e. $\left(\lambda_i v_{ij} + v_{ij}\sigma_j\right) = b_{ij}$, $i = 1, 2, \ldots, N_y - 1$ and $j = 1, 2, \ldots, N_x - 1$,

$$\therefore v_{ij} = \frac{b_{ij}}{\lambda_i + \sigma_j}$$ where v_{ij} and b_{ij} are the elements of V and B, respectively.

Once V is solved, $\bar{u}$ 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,

$$u_{xx} = f, \quad x \in [-1,1], \quad u(-1) = g_- \text{ and } u_x(1) = g_+.$$

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

(3) Discretization:

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

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,

$$u_{xx}+u_{yy}=f,\quad (x,y)\in[-1,1]\times[-1,1],$$
$$u(-1,y)=g_{-}^{x}(y),\ u_x(1,y)=g_{+}^{x}(y),$$
$$u(x,-1)=g_{-}^{y}(x),\ u_y(x,1)=g_{+}^{y}(x).$$

(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:

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

where $\hat{D}_{yy}=\begin{bmatrix}(D_y)_{0,j}\\ \hline (D_{yy})_{ij}\\ \hline I_{N_y,j}\end{bmatrix}_{(N_y+1)\times(N_y+1)}$, $\hat{D}_{xx}=\begin{bmatrix}(D_x)_{0,j}\\ \hline (D_{xx})_{ij}\\ \hline I_{N_x,j}\end{bmatrix}_{(N_x+1)\times(N_x+1)}$,

$$\hat{f}=\left[\begin{array}{c|ccc|c} g_{+}^{x}\text{ or }g_{+}^{y} & g_{+}^{y} & \cdots & g_{+}^{y} & g_{-}^{x}\text{ or }g_{+}^{y}\\ \hline g_{+}^{x} & & & & g_{-}^{x}\\ \vdots & & f_{ij} & & \vdots\\ g_{+}^{x} & & & & g_{-}^{x}\\ \hline g_{+}^{x}\text{ or }g_{-}^{y} & g_{-}^{y} & \cdots & g_{-}^{y} & g_{-}^{x}\text{ or }g_{-}^{y}\end{array}\right]_{(N_y+1)\times(N_x+1)},\ \text{here}\ \begin{cases} i=1,2,\ldots,N_y-1\\ j=1,2,\ldots,N_x-1\end{cases}.$$

(3) However, using 5×5 matrix as an example, $\hat{D}_{yy}u+u\hat{D}^{t}_{xx}=\hat{f}$ will become

$$\left[\begin{array}{c|ccc|c} u_y+u_x & u_{xx}+u_y & u_{xx}+u_y & u_{xx}+u_y & u_y+u \\ \hline u_{yy}+u_x & \Delta u & \Delta u & \Delta u & u_{yy}+u \\ u_{yy}+u_x & \Delta u & \Delta u & \Delta u & u_{yy}+u \\ u_{yy}+u_x & \Delta u & \Delta u & \Delta u & u_{yy}+u \\ \hline u+u_x & u_{xx}+u & u_{xx}+u & u_{xx}+u & u+u \end{array}\right]$$

$$=\left[\begin{array}{c|ccc|c} g^x_+ \text{ or } g^y_+ & g^y_+ & g^y_+ & g^y_+ & g^x_- \text{ or } g^y_+ \\ \hline g^x_+ & f & f & f & g^x_- \\ g^x_+ & f & f & f & g^x_- \\ g^x_+ & f & f & f & g^x_- \\ \hline g^x_+ \text{ or } g^y_- & g^y_- & g^y_- & g^y_- & g^x_- \text{ or } g^y_- \end{array}\right],$$

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.

Diagonalization with Penalty Method

(1) Consider the 2D Poisson equation with all kind of boundary conditions in a rectangular domain, i.e.,

$$B_{\pm}^{x}u\left(\pm 1, y\right) = g_{\pm}^{x}(y), \quad B_{\pm}^{y}u\left(x, \pm 1\right) = g_{\pm}^{y}(x),$$

where

$$B_{\pm}^{x} = \alpha_{\pm}^{x} \pm \beta_{\pm}^{x}\partial_{x}, \qquad B_{\pm}^{y} = \alpha_{\pm}^{y} \pm \beta_{\pm}^{y}\partial_{y},$$

and $\alpha_{\pm}^{x}$, $\beta_{\pm}^{x}$, $\alpha_{\pm}^{y}$, and $\beta_{\pm}^{y}$ are specified constants. At least one of $\alpha_{\pm}^{x}$ and $\alpha_{\pm}^{y}$ must be non-zero for a unique solution to exist.

(2) At this time, we can not strip first/last row/column of D_{yy} and D_{xx} 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

$$\tilde{D}_{xx} = \begin{bmatrix} \dfrac{\left(D_{xx}\right)_{0,j} - \tau_+^x \left(\alpha_+^x I_{0,j} + \beta_+^x \left(D_x\right)_{0,j}\right)}{\left(D_{xx}\right)_{ij}} \\ \dfrac{}{\left(D_{xx}\right)_{N_x,j} - \tau_-^x \left(\alpha_-^x I_{N_x,j} - \beta_-^x \left(D_x\right)_{N_x,j}\right)} \end{bmatrix}_{\left(N_x+1\right)\times\left(N_x+1\right)}$$

$$\tilde{D}_{yy} = \begin{bmatrix} \dfrac{\left(D_{yy}\right)_{0,j} - \tau_+^y \left(\alpha_+^y I_{0,j} + \beta_+^y \left(D_y\right)_{0,j}\right)}{\left(D_{yy}\right)_{ij}} \\ \dfrac{}{\left(D_{yy}\right)_{N_y,j} - \tau_-^y \left(\alpha_-^y I_{N_y,j} - \beta_-^y \left(D_y\right)_{N_y,j}\right)} \end{bmatrix}_{\left(N_y+1\right)\times\left(N_y+1\right)},$$

where τ_+^x, τ_-^x, τ_+^y, and τ_-^y are the penalty weights, which are usually large numbers.

(4) Using 5×5 matrix as an example, $\tilde{D}_{yy}u+u\tilde{D}_{xx}^{t}=\tilde{f}$ will become

$$\begin{bmatrix}
\Delta u-\tau_+^x B_+^x u-\tau_+^y B_+^y u & \Delta u-\tau_+^y B_+^y u & \Delta u-\tau_+^y B_+^y u & \Delta u-\tau_+^y B_+^y u & \Delta u-\tau_-^x B_-^x u-\tau_+^y B_+^y u \\
\Delta u-\tau_+^x B_+^x u & \Delta u & \Delta u & \Delta u & \Delta u-\tau_-^x B_-^x u \\
\Delta u-\tau_+^x B_+^x u & \Delta u & \Delta u & \Delta u & \Delta u-\tau_-^x B_-^x u \\
\Delta u-\tau_+^x B_+^x u & \Delta u & \Delta u & \Delta u & \Delta u-\tau_-^x B_-^x u \\
\Delta u-\tau_+^x B_+^x u-\tau_-^y B_-^y u & \Delta u-\tau_-^y B_-^y u & \Delta u-\tau_-^y B_-^y u & \Delta u-\tau_-^y B_-^y u & \Delta u-\tau_-^x B_-^x u-\tau_-^y B_-^y u
\end{bmatrix}$$

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

which well approximates $\Delta u=f$ with the specified BC's when τ'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 λ_i and e_{y_i} represent the eigenvalue and eigenvector of $\tilde{D}_{yy}$

and $Y = \left[e_{y_0}, e_{y_1}, \ldots, e_{y_{N_y}} \right]$.

(2) Similarly,

$$\tilde{D}^t_{xx} e_{x_j} = \sigma_j e_{x_j} \Rightarrow \tilde{D}^t_{xx} X = X\Sigma \text{ where } \Sigma = \begin{bmatrix} \sigma_0 & & & \\ & \sigma_1 & & \\ & & \ddots & \\ & & & \sigma_{N_x} \end{bmatrix}.$$

$$\tilde{D}_{yy}u + u\tilde{D}_{xx}^{t} = \tilde{f}$$

$$\tilde{D}_{yy}uX + u\tilde{D}_{xx}^{t}X = \tilde{f}X$$

$$\tilde{D}_{yy}uX + uX\Sigma = \tilde{f}X$$

$$\tilde{D}_{yy}YY^{-1}uX + uX\Sigma = \tilde{f}X$$

$$Y\Lambda Y^{-1}uX + uX\Sigma = \tilde{f}X$$

$$\Lambda Y^{-1}uX + Y^{-1}uX\Sigma = Y^{-1}\tilde{f}X$$

$\Lambda V + V\Sigma = B$ 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 $j = 0,1,\ldots,N_x$,

$$\therefore v_{ij} = \frac{b_{ij}}{\lambda_i + \sigma_j}$$ where v_{ij} and b_{ij} are the elements of V and B,

respectively.

Once V is solved, u can be recovered from $u = YVX^{-1}$.

3D Case

(1)

$$\tilde{D}_{xx}=\left[\frac{\left(D_{xx}\right)_{0,j}-\tau_{+}^{x}\left(\alpha_{+}^{x}I_{0,j}+\beta_{+}^{x}\left(D_{x}\right)_{0,j}\right)}{\left(D_{xx}\right)_{ij}}\over{\left(D_{xx}\right)_{N_x,j}-\tau_{-}^{x}\left(\alpha_{-}^{x}I_{N_x,j}-\beta_{-}^{x}\left(D_{x}\right)_{N_x,j}\right)}}\right]_{\left(N_x+1\right)\times\left(N_x+1\right)},$$

$$\tilde{D}_{yy}=\left[\frac{\left(D_{yy}\right)_{0,j}-\tau_{+}^{y}\left(\alpha_{+}^{y}I_{0,j}+\beta_{+}^{y}\left(D_{y}\right)_{0,j}\right)}{\left(D_{yy}\right)_{ij}}\over{\left(D_{yy}\right)_{N_y,j}-\tau_{-}^{y}\left(\alpha_{-}^{y}I_{N_y,j}-\beta_{-}^{y}\left(D_{y}\right)_{N_y,j}\right)}}\right]_{\left(N_y+1\right)\times\left(N_y+1\right)},$$

$$\tilde{D}_{zz}=\left[\frac{\left(D_{zz}\right)_{0,j}-\tau_{+}^{z}\left(\alpha_{+}^{z}I_{0,j}+\beta_{+}^{z}\left(D_{z}\right)_{0,j}\right)}{\left(D_{zz}\right)_{ij}}\over{\left(D_{zz}\right)_{N_z,j}-\tau_{-}^{z}\left(\alpha_{-}^{z}I_{N_z,j}-\beta_{-}^{z}\left(D_{z}\right)_{N_z,j}\right)}}\right]_{\left(N_z+1\right)\times\left(N_z+1\right)},$$

(2)

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

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

$$\sum_{k=0}^{N_z}\left(\tilde{D}_{zz}^{t}\right)_{lk} Z_{ko} = \omega_o Z_{lo} \Rightarrow \sum_{l=0}^{N_z}\sum_{k=0}^{N_z} Z_{ol}^{-1}\left(\tilde{D}_{zz}^{t}\right)_{lk} Z_{ko} = \omega_o.$$

(3)

$$\sum_{l=0}^{N_y}\left(\tilde{D}_{yy}\right)_{il} u_{ljk} + \sum_{l=0}^{N_x} u_{ilk}\left(\tilde{D}^t_{xx}\right)_{lj} + \sum_{l=0}^{N_z} u_{ijl}\left(\tilde{D}^t_{zz}\right)_{lk} = F_{ijk},$$

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

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

$$\sum_{k=0}^{N_z}\sum_{j=0}^{N_x}\sum_{l=0}^{N_y} Y_{im}\lambda_m Y_{ml}^{-1} u_{ljk} X_{jn} Z_{ko} + \sum_{k=0}^{N_z}\sum_{l=0}^{N_x} u_{ilk} X_{ln} Z_{ko}\sigma_n$$

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

$$\sum_{k=0}^{N_z}\sum_{j=0}^{N_x}\sum_{l=0}^{N_y} \lambda_m Y_{ml}^{-1} u_{ljk} X_{jn} Z_{ko} + \sum_{i=0}^{N_y}\sum_{k=0}^{N_z}\sum_{l=0}^{N_x} Y_{mi}^{-1} u_{ilk} X_{ln} Z_{ko}\sigma_n$$

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

$$\lambda_m \sum_{i=0}^{N_y}\sum_{k=0}^{N_z}\sum_{j=0}^{N_x} Y_{ml}^{-1} u_{ljk} X_{jn} Z_{ko} + \sigma_n \sum_{i=0}^{N_y}\sum_{k=0}^{N_z}\sum_{l=0}^{N_x} Y_{mi}^{-1} u_{ilk} X_{ln} Z_{ko}$$

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

Hence, $\left(\lambda_m + \sigma_n + \omega_o\right) W_{mno} = B_{mno} \Rightarrow W_{mno} = \dfrac{B_{mno}}{\lambda_m + \sigma_n + \varpi_o}, \forall m, n, o,$

where $W_{mno} = \sum_{i=0}^{N_y}\sum_{k=0}^{N_z}\sum_{l=0}^{N_x} Y_{mi}^{-1} u_{ilk} X_{ln} Z_{ko}, \quad B_{mno} = \sum_{i=0}^{N_y}\sum_{k=0}^{N_z}\sum_{j=0}^{N_x} Y_{mi}^{-1} F_{ijk} X_{jn} Z_{ko}.$

$$\therefore u_{ijk} = \sum_{m=0}^{N_y}\sum_{o=0}^{N_z}\sum_{n=0}^{N_x} Y_{im} W_{mno} X_{nj}^{-1} Z_{ok}^{-1}.$$

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) $u'(-1) = 4\pi,\ u(1) + u'(1) = 4\pi.$

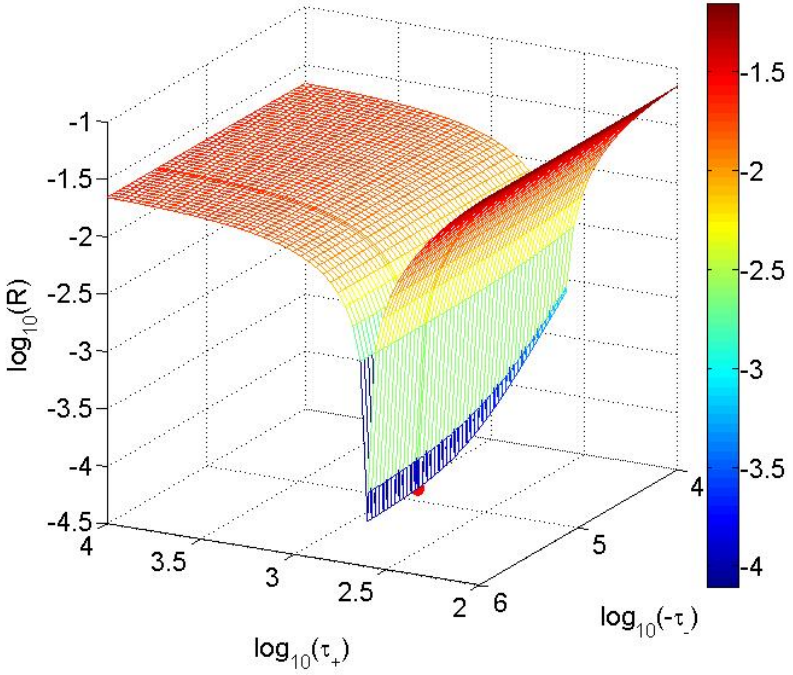


Figure 1: BC (a), N=16.

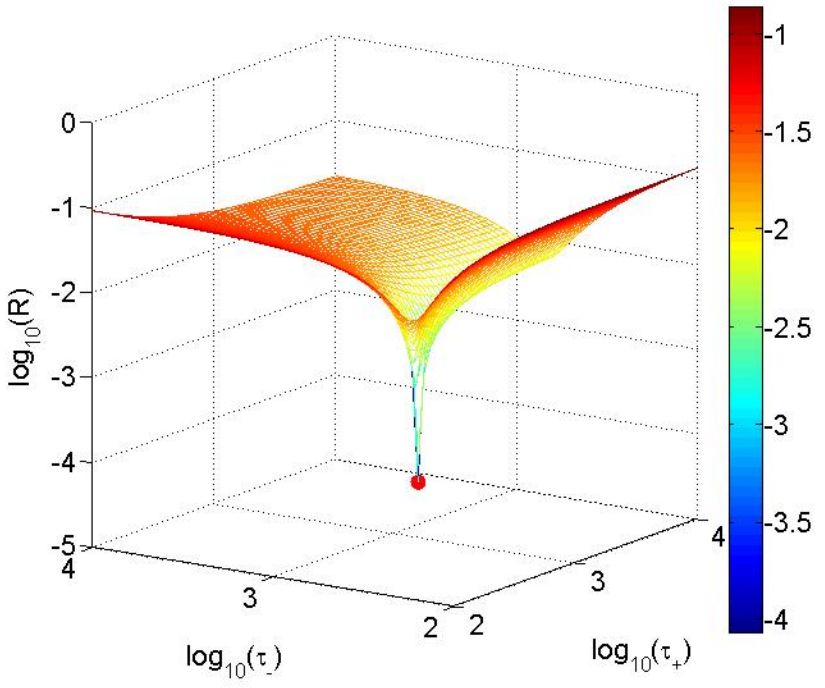


Figure (2): BC (b), N=16.

Error-Minimized Tau vs. Strongly Enforcement

Strongly-enforced BC is equivalen to $\tau \to \infty$.

Table 1: BC (a)

N	Present method		Strongly enforced BC	
	L_2 error	L_∞ error	L_2 error	L_∞error
16	7.5861e−03	8.2458e−03	8.2730e−01	7.6315e−01
20	8.4171e−05	8.9049e−05	2.2776e−02	2.1001e−02
24	4.2298e−07	4.4719e−07	2.1196e−04	1.9540e−04
28	1.1169e−09	1.2001e−09	8.9051e−07	8.2080e−07
32	1.5770e−12	1.6215e−12	1.9692e−09	1.8149e−09

Table 2: BC (b)

N	Present method		Strongly enforced BC	
	L_2 error	L_∞ error	L_2 error	L_∞error
16	7.7124e−03	8.3458e−03	8.2730e−01	7.6315e−01
20	8.7012e−05	9.1708e−05	2.2776e−02	2.1001e−02
24	4.4364e−07	4.7063e−07	2.1189e−04	1.9534e−04
28	1.1853e−09	1.2869e−09	8.9054e−07	8.2082e−07
32	1.7921e−12	1.9433e−12	1.9697e−09	1.8153e−09

Error-Minimized Tau Can Be Derived

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

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

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

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

Details see: Tzyy-Leng Horng and Chun-Hao Teng*, 2012, "An error minimized pseudospectral penalty direct Poisson solver," Journal of Computational Physics, 231(6):2498–2509.

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

$B_{\pm}^{x} u(\pm 1, y) = g_{\pm}^{x}(y), \quad B_{\pm}^{y} u(x, \pm 1) = g_{\pm}^{y}(x).$

There exists a unique set $\left\{u^{x}(x, y), u^{y}(x, y), f^{x}(x, y), f^{y}(x, y)\right\}$ 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

$\partial_{xx} u^{x} = f^{x},$ subject to $B_{\pm}^{x} u(\pm 1, y) = g_{\pm}^{x}(y),$

$\partial_{yy} u^{y} = f^{y},$ subject to $B_{\pm}^{y} u(x, \pm 1) = g_{\pm}^{y}(x),$

So we can calculate error-minimized τ_{-} and τ_{+} in each direction alone according to previous formula.

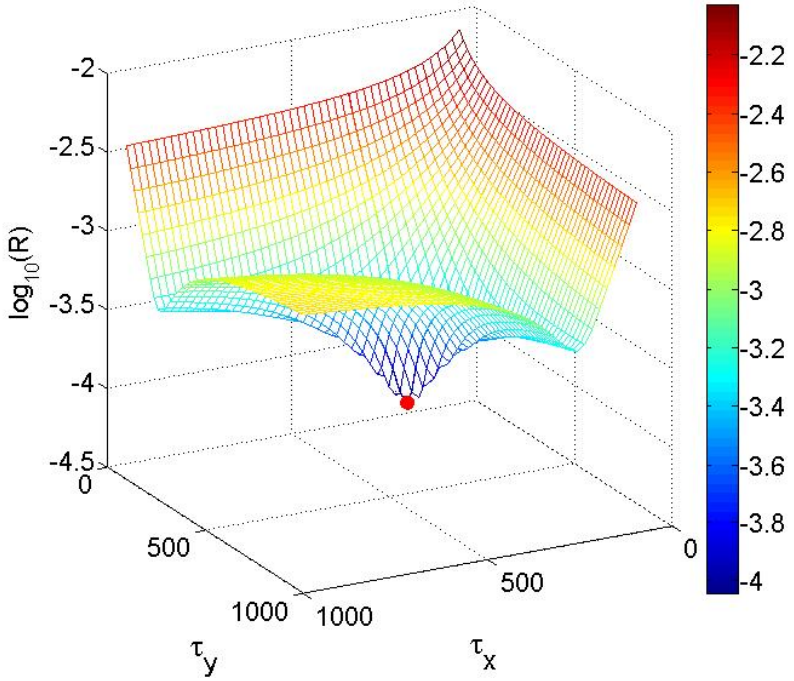
Numerical experiments for 2D PE

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

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

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

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



N	Present method		Strongly enforced BC	
	L_2 error	L_∞ error	L_2 error	L_∞ error
16	7.7325e−03	5.2042e−03	1.0911e−01	9.3090e−02
20	9.1325e−05	6.8995e−05	3.0619e−03	2.5662e−03
24	4.9615e−07	3.8162e−07	2.8567e−05	2.3930e−05
28	2.2247e−12	1.9539e−12	2.6613e−10	2.2226e−10
32	7.3045e−14	9.4147e−14	1.9322e−14	2.3648e−14
48	4.0532e−14	5.5681e−14	5.9653e−14	7.2664e−14
64	8.7292e−14	6.6391e−14	1.0678e−13	1.2695e−13

BC's: (b) $u(\pm 1, y) = 0, \quad u(x, \pm 1) = 0.$

N	Present method		CT	CC	CG
	L_2 error	L_∞ error	L_∞ error	L_∞ error	L_∞ error
16	6.89e−03	5.25e−03	3.33e−02	5.25e−03	5.22e−03
20	8.54e−05	6.26e−05		7.52e−05	
24	4.75e−07	3.76e−07	6.89e−06	4.05e−07	
32	2.17e−12	1.78e−12	4.77e−11	2.87e−12	2.17e−12
40	1.24e−14	2.05e−14		1.47e−12	
48	9.40e−15	1.03e−14	1.90e−12	3.63e−12	
64	3.15e−14	5.05e−14	8.67e−13	3.90e−12	6.11e−15

CT (Chebyshev tau): D.B. Haidvogel, T. Zang, The accurate solution of Poisson's equation by expansion in Chebyshev polynomials, J. Comput. Phys. 30 (1979) 167–180.

CC (Chebyshev collocation): U. Ehrenstein, R. Peyret, A Chebyshev collocation method for the Navier–Stokes equations with application to double-diffusive convection, Int. J. Numer. Methods fluids 9 (1989) 427–452.

CG (Chebyshev Galerkin): J. Shen, Efficient spectral-Galerkin method II. Direct solvers of second and fourth order equations by using Chebyshev polynomials, SIAM J. Sci. Comput. 16 (1995) 74–87.

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

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

N	Present method		Strongly enforced BC	
	L_2 error	L_∞ error	L_2 error	L_∞ error
16	7.30e−03	5.28e−03	6.75e−02	1.00e−01
20	8.84e−05	6.75e−05	1.87e−03	2.75e−03
24	4.87e−07	3.80e−07	1.74e−05	2.57e−05
28	2.21e−12	1.93e−12	1.68e−10	2.43e−10
32	1.59e−14	1.94e−14	1.94e−11	2.25e−11
48	2.08e−14	2.74e−14	6.12e−11	8.92e−11
64	3.75e−14	4.24e−14	1.43e−11	3.31e−11

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 eaxct solution is $u(x, y) = \sin(4\pi x)\sin(4\pi y)\sin(4\pi z).$

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

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

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

N	Present method		Strongly enforced BC	
	L_2 error	L_∞ error	L_2 error	L_∞error
16	9.0627e−03	5.4835e−03	4.5057e−02	5.6987e−02
20	1.1572e−04	8.7926e−05	1.3111e−03	1.6651e−03
24	6.5795e−07	5.0986e−07	1.2725e−05	1.5583e−05
28	3.0727e−12	2.6552e−12	1.2452e−10	1.4530e−10
32	2.0128e−14	2.8533e−14	1.1011e−14	1.3784e−14
48	2.5304e−14	3.2713e−14	1.5377e−14	1.5321e−14
64	8.4110e−14	1.1653e−13	3.0008e−14	3.7415e−14

Conclusion

(1) Asymptotic operation account:

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

2D: $2N_xN_y\left(\log\left(N_x\right)+\log\left(N_y\right)\right)$, 3D: $2N_xN_yN_z\left(\log\left(N_x\right)+\log\left(N_y\right)+\log\left(N_z\right)\right)$.

(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:

$$\left(L_x+L_y+L_z\right)u=f\left(x,y,z\right),$$

where $L_x=a_1\left(x\right)\frac{\partial^2}{\partial x^2}+b_1\left(x\right)\frac{\partial}{\partial x}+c_1\left(x\right)$, $L_y=a_2\left(y\right)\frac{\partial^2}{\partial y^2}+b_2\left(y\right)\frac{\partial}{\partial y}+c_2\left(y\right)$,

$$L_z=a_3\left(z\right)\frac{\partial^2}{\partial z^2}+b_3\left(z\right)\frac{\partial}{\partial z}+c_3\left(z\right).$$

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.

(6) Sample matlab code is available at http://newton.math.fcu.edu.tw/~tlhorng/index_eng.html.

Thank you for your attention.
Any question?