Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2016-0075

Vol. **22**, No. 2, pp. 460-472
August 2017

Precorrected-FFT Accelerated Singular Boundary Method for Large-Scale Three-Dimensional Potential Problems

Weiwei Li[1,2], Wen Chen[1,2,*] and Zhuojia Fu[1,2]

[1] *State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering, International Center for Simulation Software in Engineering and Sciences, College of Mechanics and Materials, Hohai University, Nanjing 210098, P.R. China.*
[2] *State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, P.R. China.*

Received 23 May 2016; Accepted (in revised version) 20 December 2016

Abstract. This study makes the first attempt to accelerate the singular boundary method (SBM) by the precorrected-FFT (PFFT) for large-scale three-dimensional potential problems. The SBM with the GMRES solver requires $\mathcal{O}(N^2)$ computational complexity, where N is the number of the unknowns. To speed up the SBM, the PFFT is employed to accelerate the SBM matrix-vector multiplication at each iteration step of the GMRES. Consequently, the computational complexity can be reduced to $\mathcal{O}(N\log N)$. Several numerical examples are presented to validate the developed PFFT accelerated SBM (PFFT-SBM) scheme, and the results are compared with those of the SBM without the PFFT and the analytical solutions. It is clearly found that the present PFFT-SBM is very efficient and suitable for 3D large-scale potential problems.

AMS subject classifications: 65M70, 65N35, 65T50

Key words: Singular boundary method, precorrected-FFT, fast matrix algorithm, three-dimensional potential problems, large-scale.

1 Introduction

The singular boundary method (SBM) [1] is a recent strong-form technique free of numerical integration to solve the boundary value problems. As a boundary-type meshless method, it can obtain the same accuracy of the boundary element method (BEM) without the mesh-generation. Since the original intensity factors (OIFs) are introduced, the singularities are removed [2]. This method can be viewed as a modified version of the method of fundamental solutions (MFS). However, unlike the MFS, the SBM distributes

*Corresponding author. *Email addresses:* liweiwei_hhu@163.com (W. Li), chenwen@hhu.edu.cn (W. Chen), paul212063@hhu.edu.cn (Z. Fu)

 ©2017 Global-Science Press

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

the source points coinciding with the collocation points on the physical boundary to avoid the perplexing fictitious boundary issue. Therefore, the SBM is truly meshless, free of integration, and easy-to-implement. Due to these advantages, the SBM is considered a competitive alternative method for some certain problems.

However, the SBM will run into difficulties when large-scale problems are encountered. Like in the BEM, the SBM generates a dense $N\times N$ resultant interpolation matrix, where N is the number of the unknowns [3]. Computing this kind of matrix equation is computationally very expensive and requires $\mathcal{O}(N^2)$ operations. Otherwise, if the direct solver is used, for solving the system of equations, $\mathcal{O}(N^3)$ operations are required. Thanks to costly CPU time and huge memory demands, the standard SBM is largely restricted to the small and medium-scale applications.

The acceleration techniques, when combined with the iterative linear system solver GMRES [4], is an effective strategy to speed up the SBM as in the case of the BEM. Recently, the fast multipole method (FMM) [5, 6] has been successfully applied to the SBM solution of potential [3] and Helmholtz problems [7].

As an alternative accelerating technique, this paper makes the first attempt to present the development of a precorrected-FFT (PFFT) [8] accelerated SBM approach for large-scale three-dimensional (3D) potential problems. The PFFT technique is at best an $\mathcal{O}(N\log N)$ algorithm, and the advantage of the PFFT is that it is relatively kernel independent [9], and easier to implement. It has been successfully applied to electromechanical systems [10–12], elastodynamics [9, 13], composite materials [14, 15], nonlinear wave dynamics [16, 17], acoustic problems [18, 19], just to mention a few.

The rest of this paper is organized as follows. We describe the SBM formulation for 3D potential problems followed by coupling of the SBM with the PFFT. Next, three benchmark examples are investigated to valid the developed the PFFT accelerated SBM (PFFT-SBM) strategy. The PFFT-SBM is compared with those of the SBM without the PFFT and the analytical solutions. The conclusions are drawn lastly.

2 The SBM formulation for 3D potential problems

Consider the Laplace equation governing 3D potential problems

$$\nabla^2 u(x)=0, \quad x\in\Omega, \tag{2.1}$$

subjected to boundary conditions

$$u(x)=\bar{u}(x), \quad x\in\Gamma_D, \tag{2.2}$$

$$q(x)=\frac{\partial u}{\partial n}(x)=\bar{q}(x), \quad x\in\Gamma_N, \tag{2.3}$$

where u is the potential field, ∇^2 represents the Laplacian operator, n indicates the unit outward normal, and $\Gamma=\Gamma_D\cup\Gamma_N=\partial\Omega$ denotes the whole boundary of the domain Ω.

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

In the SBM, the approximate solutions of the Laplace equation can be expressed by [20]

$$u(x_i)=\sum_{j=1,j\neq i}^{N}\alpha_j G(x_i,s_j)+\alpha_i A_{ii},\quad i=1,\cdots,N, \tag{2.4}$$

$$q(x_i)=\sum_{j=1,j\neq i}^{N}\alpha_j \frac{\partial G(x_i,s_j)}{\partial n_x}+\alpha_i B_{ii},\quad i=1,\cdots,N, \tag{2.5}$$

where $\{\alpha_i\}$, $i=1,\cdots,N$ are the unknown coefficients, x_i and s_j denote the i-th collocation and the j-th source points, respectively, N is the number of source points, n_x the outward unit normal on the collocation points x_i, and

$$G(x_i,s_j)=\frac{1}{4\pi\|x_i-s_j\|_2} \tag{2.6}$$

is the fundamental solution of the Laplace equation. A_{ii} and B_{ii} are the origin intensity factors to replace the singular terms $G(x_i,s_i)$ and $\partial G(x_i,s_i)/\partial n_x$. There are several techniques to calculate the origin intensity factors, such as inverse interpolation technique (IIT) [1], semi-analytical technique [2], and empirical formulas. In this paper, the semi-analytical technique with boundary IIT [2] is proposed. By adopting a subtracting and adding-back technique, the origin intensity factors on Neumann boundary conditions B_{ii} can be derived as

$$B_{ii}=-\frac{1}{L_i}\sum_{j=1,j\neq i}^{N}L_j\frac{\partial G(x_i,s_j)}{\partial n_s}, \tag{2.7}$$

where L_j denotes the area of the curved surface surrounding the source point s_j, as shown in Fig. 1. The source intensity factors on Dirichlet boundary can be calculated by the inverse interpolation technique. We choose a sample solution $\bar{u}_0=x+y+z+c$ for 3D problems, and then $2N+1$ linear equations are obtained

$$\bar{u}_0(x_i)=\sum_{j=1,j\neq i}^{N}\beta_j G(x_i,s_j)+\beta_i A_{ii}+c,\quad x_i=s_j, \tag{2.8}$$

$$\frac{\partial\bar{u}_0(x_i)}{\partial n_x}=\sum_{j=1,j\neq i}^{N}\beta_j\frac{\partial G(x_i,s_j)}{\partial n_x}+\beta_i B_{ii},\quad x_i=s_j, \tag{2.9}$$

$$\bar{u}_0(x_I)=\sum_{j=1}^{N}\beta_j G(x_I,s_j)+c,\quad x_I\neq s_j, \tag{2.10}$$

where x_I represents an inner point, and (A_{ii},β_j,c) are $2N+1$ unknowns which can be solved.

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

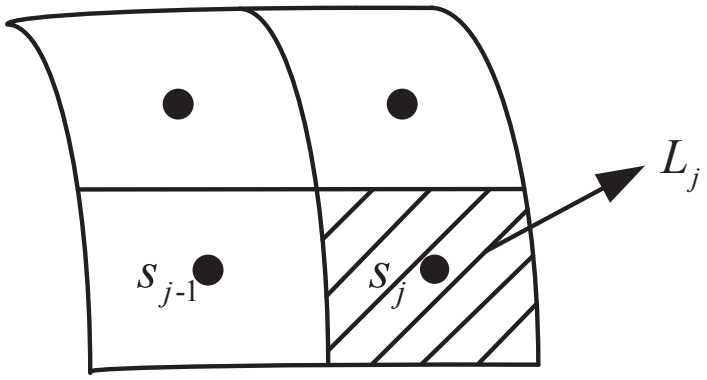


Figure 1: Nodal integration domain for 3D problems.

3 The precorrected-FFT algorithm

As mentioned in the introduction, the SBM results in a system of equations having a dense influence matrix. In this paper, the iterative solver GMRES is employed to solve this system equation instead of the direct approach, e.g., Gaussian elimination. This reduces the computing operations from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$. The most time-consuming part of the GMRES is the matrix-vector product. To further accelerate the computation, the precorrected-FFT (PFFT) algorithm is applied. The main idea of the PFFT algorithm is to separate the computing interactions into far- and near-field parts [8]. The far-field interactions are computed by projecting onto, and interpolating from, a uniform 3D coarse grid, and the near-field interactions are calculated directly in the standard fashion. More specifically, after the nodes are collocated on the boundary of the problem domain, a parallelepiped containing the computational domain is constructed. Then this parallelepiped could be subdivided into an array of small cubes, so that every small cube contains a few nodes. The potentials at the uniform grid points formed by the cubes can be quickly computed by the FFT. Thus, the far-field interactions can be obtained with the aid of the uniform grid.

The main steps of the PFFT algorithm are summarized as follows:

1. Discretization of the problem domain and construction of the 3D uniform grid, as show in Fig. 2;
2. Separation of the near- and far-fields for each boundary node using the grid, as show in Fig. 3;
3. Projection of the quantities of the boundary nodes onto the surrounding grid points, as show in Fig. 3;
4. Computation of the grid-to-grid interaction using the FFT, as show in Fig. 3;
5. Interpolation of the grid interaction to the boundary nodes, as show in Fig. 3;

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

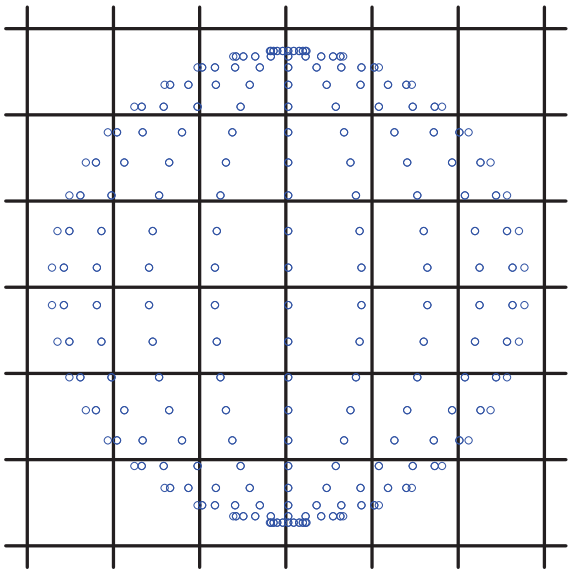


Figure 2: Discretization of the problem domain and construction of the 3D uniform grid.

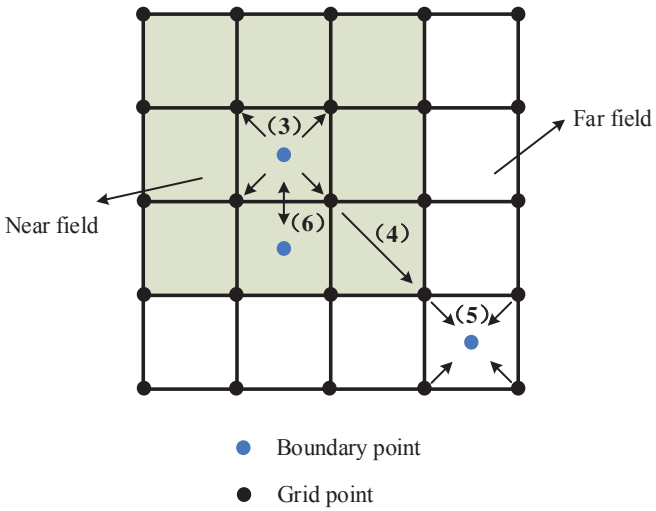


Figure 3: A pictorial representation of the steps (2)-(7) in the PFFT algorithm.

6. Computation of the near-field interaction directly, as show in Fig. 3;
7. Substitution of the approximated near-field interaction by the accurate near-field interaction.

The projection in step 3 and the interpolation in step 5 are performed using polynomial interpolation. The size of the near-field part will also affect the accuracy and the computational cost. In comparison with PFFT acceleration BEM, the main advantage of the

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

PFFT-SBM is that the PFFT-SBM avoids the mesh-generation and the expensive numerical integration in the projection, interpolation and computation of the near-field interaction.

Due to the requirement of the FFT, the number of the cubes in each dimensional direction should be 2^k-1, where k is an positive integer. It should be noted that the number of the cubes is determined according to the balance between efficiency and accuracy. Theoretically, as the number of the cubes grows, the computation cost and accuracy of the far-field interaction will both increase. In this paper, the number of the cubes is controlled by checking the maximum number of the boundary nodes in every cube. If the maximum number is larger than a threshold value P, more cubes are generated.

4 Numerical examples

The PFFT-SBM is implemented in C++ in this study. In order to measure the accuracy of the PFFT-SBM, the relative error defined as below is employed

$$e=\left[\frac{1}{M}\sum_{k=1}^{M}\left(\frac{I^k_{numerical}-I^k_{exact}}{I^k_{exact}}\right)^2\right]^{1/2}, \tag{4.1}$$

where M is the number of the test points, $I^k_{numerical}$ and I^k_{exact} denote the numerical and analytical solution at the k-th test point, respectively.

In our study, the iterative solver GMRES with a tolerance 1.0E-5 is used to both the SBM without the PFFT and the PFFT-SBM. A block diagonal matrix is employed as the preconditioner to accelerate the convergence of the GMRES solver. The threshold value P is set as 20. Computations are implemented on a desktop computer with an Intel Core i3-3110M 2.40 GHz Processor with 4G RAM running Windows 8.1.

4.1 A unit cube

First, we consider a unit cube to demonstrate the performance of the developed PFFT-SBM. The following exact solution is $u=-2x^2+y^2+z^2-xy+3$. Neumann boundary conditions are imposed on the upper and lower surfaces, and Dirichlet boundary conditions on the remaining surfaces. The relative error is evaluated over 36 points distributed uniformly on the surface of the sphere of radius 0.3, where the center coincides with that of the cube.

Fig. 4 shows the relative errors of u using the SBM with and without the PFFT. As can be seen from this figure, the accuracy of the PFFT-SBM is almost same as that of the SBM without the PFFT when using the same number of boundary nodes. However, due to the memory limit of the computer, the SBM without the PFFT is inapplicable for the large-scale case ($N>25000$) while the PFFT-SBM can keep working well. As the number of boundary nodes increases, the memory requirement and CPU time are shown in Fig. 5

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

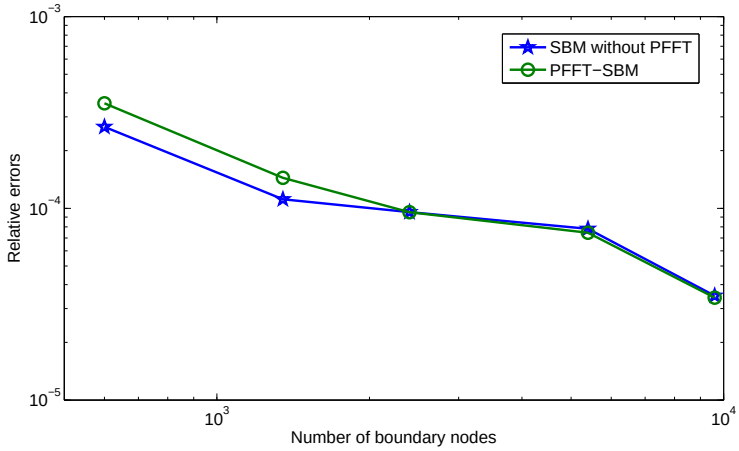


Figure 4: Relative error curves of u generated using the SBM with and without the PFFT.

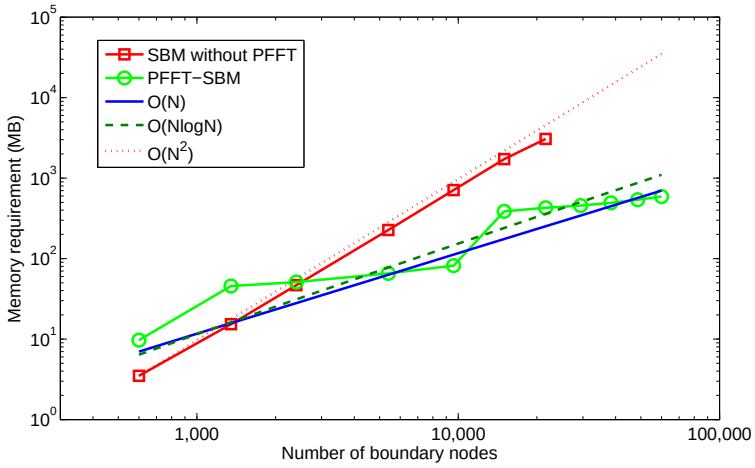


Figure 5: Comparison of memory requirements of the SBM with and without the PFFT.

and Fig. 6. We can observe that the SBM without the PFFT requires far more memory and CPU time than the PFFT-SBM, especially for large-scale simulation. For example, for the model with 21600 boundary nodes, more than 3GB memory and 2 hours are required in the SBM without the PFFT, while only 427.8MB memory and 172 seconds in the PFFT-SBM. This clearly shows that the PFFT-SBM has great advantage in efficiency over the SBM without the PFFT. Besides, the lines corresponding to $\mathcal{O}(N)$, $\mathcal{O}(N\log N)$ and $\mathcal{O}(N^2)$ are also plotted for reference. As excepted, the memory requirement and the CPU time of the PFFT-SBM oscillate along the line $\mathcal{O}(N\log N)$, while the SBM without the PFFT are of $\mathcal{O}(N^2)$. The results largely agree with the theoretical analysis.

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

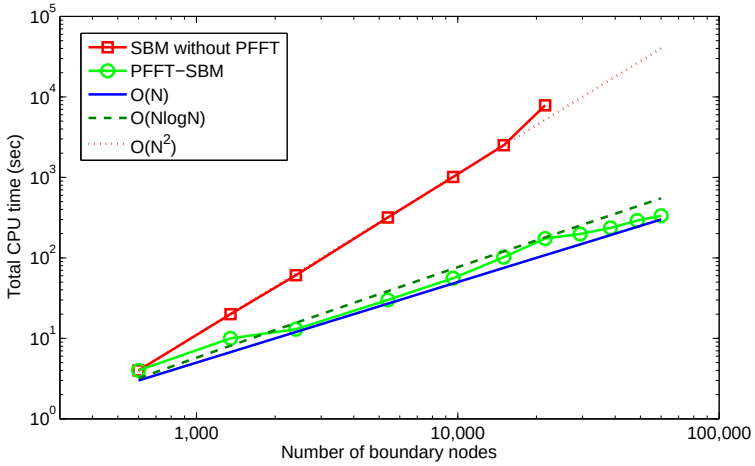


Figure 6: Comparison of total CPU time of the SBM with and without the PFFT.

4.2 A ring torus

A ring torus is considered as the second example. As shown in Fig. 7, the torus can be defined as

$$x(\phi,\theta)=\big((R+r\cos\phi)\cos\phi,(R+r\cos\phi)sin(\phi),r\sin\phi\big), \tag{4.2}$$

where R=2 and r=1 are the distance between the centers of the tube and the torus and the radius of the tube, respectively. The exact solution is $u=xy+yz+xz$. Dirichlet boundary is imposed on the whole surface. 20 points distributed uniformly on the curved center line of the torus are tested for the accuracy.

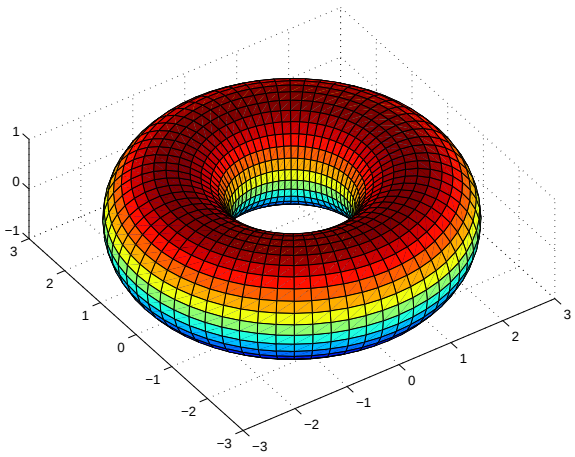


Figure 7: A ring torus ($R=2, r=1$).

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

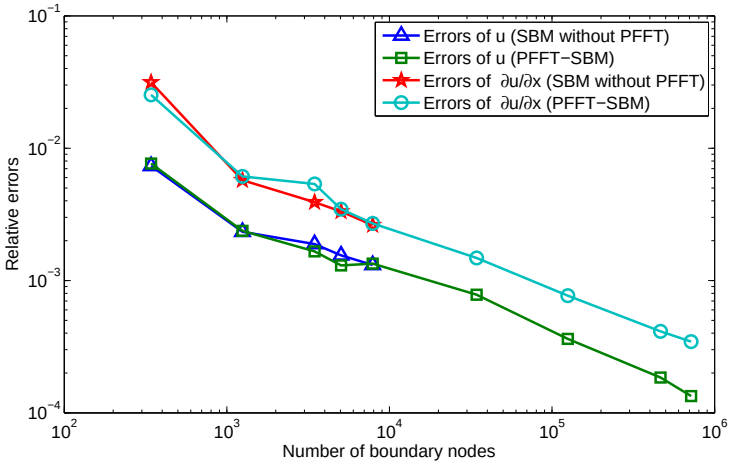


Figure 8: Relative error curves by the SBM with and without the PFFT.

Fig. 8 shows the relative errors of u and $\partial u/\partial x$ by the SBM with and without the PFFT as the number of boundary nodes varies from 600 to 716680. As shown in this figure, both the schemes obtain accurate results, and the PFFT-SBM can achieve the almost same accuracy of the SBM without the PFFT.

Fig. 9 and Fig. 10 illustrate the memory requirement and CPU time of the SBM with and without the PFFT, respectively. It can be noticed that the PFFT-SBM can solve the model with more than 400000 unknowns on a personal computer. For the model with 465000 boundary nodes, the PFFT-SBM used 2819.3 MB memory and 6751 seconds to obtain the numerical result. However, the SBM without the PFFT is only workable when

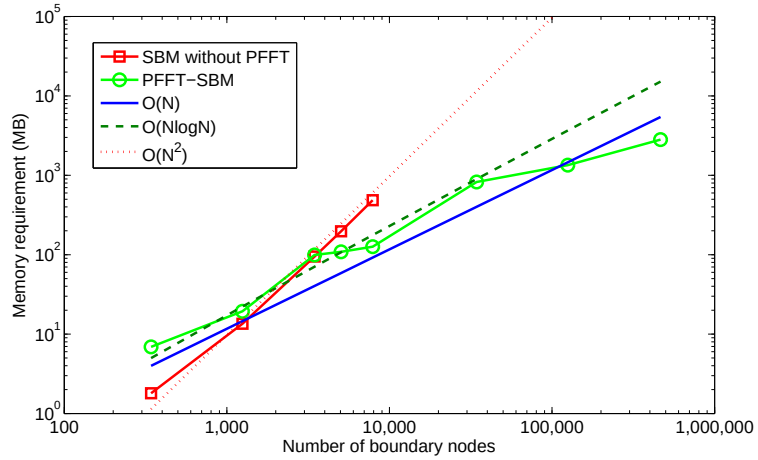


Figure 9: Comparison of memory requirements between the SBM with and without the PFFT.

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

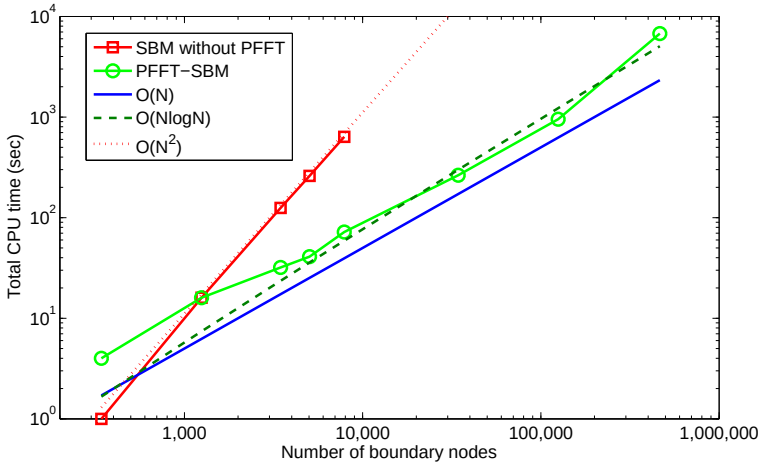


Figure 10: Comparison of total CPU time between the SBM with and without the PFFT.

the number of boundary nodes is less than 25000. If the scale of the computation is enlarged, the memory requirements of the SBM without the PFFT will approach or exceed the limit of the computer in this study. These figures clearly show that the PFFT-SBM improves computational efficiency dramatically.

The lines corresponding to $\mathcal{O}(N)$, $\mathcal{O}(N\log N)$ and $\mathcal{O}(N\log N)$ are also plotted in these figures. It is observed that the computing complexity of the PFFT-SBM is $\mathcal{O}(N\log N)$ against $\mathcal{O}(N^2)$ without the PFFT.

4.3 Porous model

In order to further test the PFFT-SBM to a complex case, the third example deals with a porous model, which is a cube including 100 equal cavities in the shape of capsules as shown in Fig. 11. The length of the cube is $L=100$, and the size of every capsule is shown in Fig. 12. The exact solution is $u=-2x^2+y^2+z^2-xy+3$. Dirichlet boundary is imposed on the whole surface. This model has very complicated geometries that require a large number of boundary nodes in the SBM. We should point out that it is almost impossible to solve this problem with the SBM without the PFFT in a desktop computer. Computations with the PFFT-SBM are showed for six nodal cases, namely, 145800 nodes, 187200 nodes, 231900 nodes, 313600 nodes, 411500 nodes, and 600000 nodes, respectively. The relative errors of u and $\partial u/\partial x$ are evaluated over 60 interior points on the surface whose distance to the outer boundary is 2, and presented in the second and third rows of Table 1, respectively. These errors demonstrate that high efficiency and accuracy can be obtained by the PFFT-SBM. The fourth and fifth rows list the CPU time and memory requirement. Although the computational cost grows with the increasing number of boundary nodes, it is still feasible in a personal computer. This example illustrates that the PFFT-SBM is capable to provide a fine solution of the problem with complicated geometry.

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

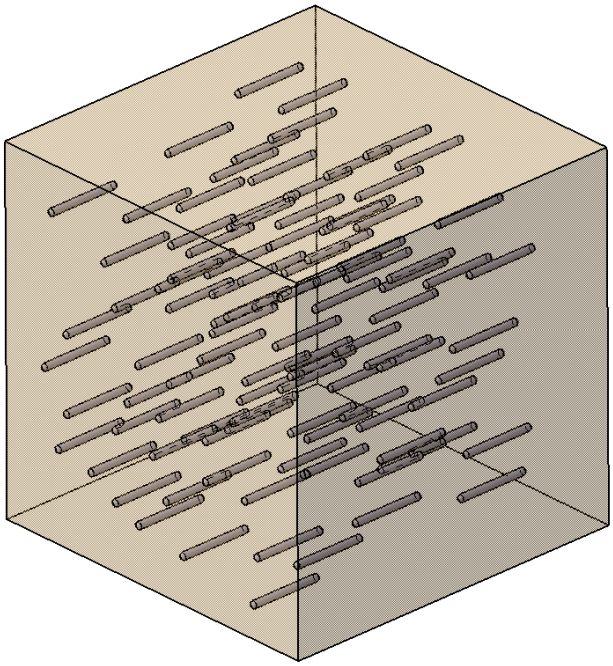


Figure 11: A cube including many cavities in the shape of capsules.

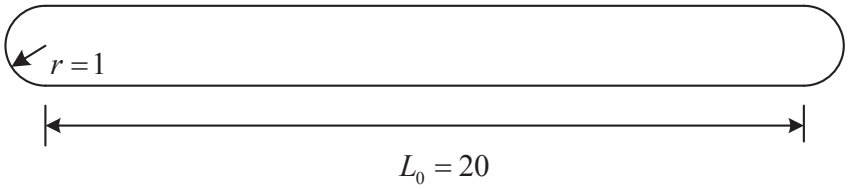


Figure 12: The size of every capsule.

Table 1: The results of porous model using PFFT-SBM.

Nodes number	Errors of u	Errors of $\partial u/\partial x$	CPU time(s)	Memory(Mb)
145800	1.39627E-3	2.15784E-3	653	602.7
187200	1.11619E-3	1.00687E-3	844	803.4
231900	8.29500E-4	7.36562E-4	1058	1064.0
313600	4.31786E-4	6.16614E-4	1533	1331.1
411500	3.33205E-4	5.38353E-4	2652	1796.5
600000	1.03576E-4	3.96784E-4	4262	2046.8

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

5 Conclusions

A fast SBM has been developed for 3D potential problems based on the PFFT algorithm in this study. As a boundary-type strong-form meshless technique, the SBM can achieve the good numerical accuracy and efficiency without the mesh-generation, the singularity and the numerical integration in the BEM. The SBM is truly meshless, mathematically simple, and easy to program. As the iterative solver GMRES and acceleration technique PFFT are employed, the CPU time and memory requirements can be reduced to $\mathcal{O}(N\log N)$ in the SBM procedure. The three numerical examples are presented to illustrate the accuracy and efficiency of the developed PFFT-SBM code for large-scale potential problems.

In comparison with the PFFT accelerated BEM, the PFFT-SBM avoids the mesh-generation and the expensive numerical integration in the projection, interpolation and computation of the near-field interaction.

From the results of the first two numerical examples, it can be found that the PFFT-SBM can achieve the almost same accuracy of the SBM without the PFFT when using the same number of boundary nodes. However, the SBM without the PFFT requires far more memory and CPU time than the PFFT-SBM, especially for large-scale simulation. The SBM without the PFFT is inapplicable for the large-scale case on a personal computer while the PFFT-SBM can keep working well.

The third example deals with a porous model having very complicated inner holes. This example demonstrates that high efficiency and accuracy can be obtained as well for such complex-shaped problem by the PFFT-SBM. The results of the numerical experiments are also shows that the PFFT-SBM is at best an $\mathcal{O}(N\log N)$ algorithm.

Acknowledgments

The authors are grateful to the National Science Funds of China (Grant Nos. 11572111, 11370297), the 111 Project (Grant No. B12032), the Jiangsu Province Graduate Students Research and Innovation Plan (Grant No. KYLX_0419), the Fundamental Research Funds for the Central Universities (Grant No. 2014B31514). Thanks Prof. Zaiyou Yan for the provision of the PFFT code.

References

[1] W. Chen, Singular boundary method: a novel, simple, meshfree, boundary collocation numerical method (in Chinese), ACTA MECH SOLIDA SIN 30 (2009), 592-599.

[2] Z. Fu, W. Chen, J. T. Chen, and W. Qu, Singular Boundary Method: Three Regularization Approaches and ExteriorWave Applications, CMES-COMP MODEL ENG 99 (2014), 417-443.

[3] Y. Gu, H. Gao, W. Chen, C. Liu, C. Zhang, and X. He, Fast-multipole accelerated singular boundary method for large-scale three-dimensional potential problems, INT J HEAT MASS TRAN 90 (2015), 291-301.

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press

[4] Y. Saad, and M. H. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, SIAM J Sci and Stat Comput 7 (1986), 856-869.
[5] G. L, and R. V, A fast algorithm for particle simulations, J COMPUT PHYS 72 (1987), 325-348.
[6] G. L, and R. V, A new version of the fast multipole method for the Laplace equation in three dimensions, ACTA NUMER 6 (1997), 229-269.
[7] W. Qu, W. Chen, and Y. Gu, Fast multipole accelerated singular boundary method for the 3D Helmholtz equation in low frequency regime, COMPUT MATH APPL 70 (2015), 679-690.
[8] P. J. R, and W. J. K, A precorrected-FFT method for electrostatic analysis of complicated 3-d structures, IEEE T COMPUT AID D 16 (1997), 1059-1072.
[9] Z. Y. Yan, J. Zhang, and W. Ye, Rapid solution of 3-D oscillatory elastodynamics using the pFFT accelerated BEM, ENG ANAL BOUND ELEM 34 (2010), 956-962.
[10] J. Ding, and W. Ye, A fast integral approach for drag force calculation due to oscillatory slip Stokes flows, INT J NUMER METH ENG 60 (2004), 1535-1567.
[11] N. Masters, and W. Ye, Fast BEM Solution for Coupled 3D Electrostatic and Linear Elastic Problems, ENG ANAL BOUND ELEM 28 (2004), 1175C1186.
[12] W. Ye, X. Wang, W. Hemmert, D. Freeman, and J. White, Air damping in laterally oscillating micro-resonators: A Numerical and Experimental Study, J MICROELECTROMECH S 12 (2003), 557-566.
[13] J. Xiao, W. Ye, Y. Cai, and J. Zhang, Precorrected FFT accelerated BEM for large-scale transient elastodynamic analysis using frequency-domain approach, INT J NUMER METH ENG 90 (2012), 116-134.
[14] Z. Chen, S. Chai, H. Yang, and J. Mao, Precorrected-FFT method for EM scattering from composite metallicdielectric objects, CHINESE SCI BULL 55 (2010), 656-663.
[15] Y. Liua, N. Nishimurab, and Y. Otanib, Large-scale modeling of carbon-nanotube composites by a fast multipole boundary element method, COMP MATER SCI 34 (2005), 173-187.
[16] S.-c. Jiang, B. Teng, Y. Gou, and D.-z. Ning, A precorrected-FFT higher-order boundary element method for waveCbody problems, ENG ANAL BOUND ELEM 36 (2012), 404-415.
[17] H. Yan, and Y. Liu, An efficient high-order boundary element method for nonlinear waveCwave and wave-body interactions, J COMPUT PHYS 230 (2011), 402-424.
[18] Z. Yan, Simulation of acoustic scattering by the fast BEM approach, J HYDRODYN 22 (2010), 948-953.
[19] Z. Yan, and X. Gao, The development of the pFFT accelerated BEM for 3-D acoustic scattering problems based on the Burton and Miller's integral formulation, ENG ANAL BOUND ELEM 37 (2013), 409-418.
[20] W. Chen, and Z. Fu, A method of fundamental solutions without fictitious boundary, ENG ANAL BOUND ELEM 34 (2010), 530-532.

https://doi.org/10.4208/cicp.OA-2016-0075 Published online by Cambridge University Press