15th World Congress on Computational Mechanics (WCCM-XV)
8th Asian Pacific Congress on Computational Mechanics (APCOM-VIII)
Virtual Congress: 31 July – 5 August 2022
S. Koshizuka (Ed.)

MULTIPLE SOLVERS FOR IMPLICIT TEMPERATURE CALCULATION OF HEAT CONDUCTION WITH THE MPS METHOD

SIJUN LI[1], YUBAO ZHONG[1], QINGHANG CAI[1], XINKUN XIAO[1], KAILUN GUO[1*], RONGHUA CHEN[1], WENXI TIAN[1], SUIZHENG QIU[1], GUANGHUI SU[1]

[1]School of Nuclear Science and Technology, Shaanxi Key Laboratory of Advanced Nuclear Energy And Technology, State Key Laboratory of Multiphase Flow in Power Engineering, Xi'an Jiaotong University,
Xi'an 710049, China
Corresponding author: E-mail: kailunguo@xjtu.edu.cn; Tel & Fax: +86-029-82665607

Key words: Temperature calculation, MPS method, Computational cost, Solvers

Abstract. *In this study, an implicit algorithm and different solvers are applied to the Moving Particle Semi-Implicit (MPS) method for temperature calculation. The original MPS method uses an explicit method for temperature calculation and is limited by time increment due to diffusion number. In this paper, the heat conduction of plates with Dirichlet boundary condition and Neumann boundary condition is studied. The accuracy of explicit and implicit calculation of plate heat transfer cases are compared. In addition, different solvers are compared. Consequently, it is shown that the error of the implicit algorithm is not much different from the original one. And CG is still the better solver, CGS is also a superior solver. The implicit algorithm increases the size of a single time step by a maximum of 50 times (original diffusion number is 0.2), while the calculation time of a single time step does not increase Substantially, so it has a significant effect on the acceleration of calculation.*

1 INTRODUCTION

Heat exchange phenomena appear in natural and in various engineering fields in large numbers of ways. Temperature is also the most basic property of the material, which affects the physical properties of the material and then affects the calculation of other aspects like modeling fluid flow. Therefore, no matter what kind of numerical simulation method, the calculation of heat transfer is an important part. At present, there are mesh methods based on the Euler equation and meshless methods (particle method) based on the Lagrange equation in the discrete aspect of the computing domain. After a long time of development, grid-based methods have shown great success in the simulations of heat transfer. But there are still many problems with the particle method.

Lagrangian particle methods are superior for the simulation of deformation problems including free surface flows [1], fluid-structure interactions [2], multiphase flows [3], and bubble dynamics [4]. There are two widely used particle methods for numerical simulation, Smoothed Particle Hydrodynamics (SPH) [5] and the Moving Particle Semi-Implicit (MPS) method. SPH and MPS are similar while some differences still exist. One is the difference between discrete schemes, the other is the explicit algorithm used in SPH while the semi-implicit algorithm used in MPS. In the original SPH, the pressure is directly calculated by the state equation. In addition, in the semi-implicit method, the pressure field is solved globally from Pressure Poisson Equation (PPE). In the heat transfer part of both methods [6][7], the temperature of each particle is solved directly by retrieving the particles in the neighbor domain and then calculating the interaction.

MPS was first introduced by Koshizuka and Oka [8] for single-phase flow. In the earliest MPS method, the calculation of heat transfer was not considered. Subsequently, temperature calculations were added to the MPS as a separate module [9]. An explicit calculation model and the diffusion number criterion are used to limit each time step to obtain a stable solution. However, due to the limitation of the time step, effective results cannot be obtained in a limited time in some working conditions (section 2.3). Therefore, it is necessary to reduce the limitation of each time step, which is the same motivation as the implicit calculation of viscosity term [10]. There has been some literature [11][12] on implicit thermal conductivity, but few on the optimal algorithm.

In this paper, the Krylov subspace solvers are used to solve the temperature solution matrix: Conjugate gradient method (CG) [13], Conjugate Residual method (CR) [14], Bi-Conjugate Gradient Stabilized method (BICGSTAB) [15], Bi-Conjugate Residual Stabilized method (BICRSTAB) [16], Conjugate Gradient Squared method (CGS) [17], Direct Lanczos algorithm (D-Lanczos) [18]. Meanwhile, the preconditioner Diagonal-scaling which is simple and easy to parallelize is used to Preconditioned Conjugate Gradient (PCG) and Preconditioned Bi-Conjugate Gradient Stabilized method (PBICGSTAB). Of particular interest in this work is to investigate a better solver.

2 METHODS

2.1 The MPS method

For the heat conduction problem with constant physical properties, the Lagrangian governing equations in the MPS method are given as follows:

$$\frac{D\rho}{Dt} = 0 \tag{1}$$

$$\frac{Du}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2 u + g \tag{2}$$

$$\rho c\frac{dT}{dt} = k\nabla^2 T + \dot{Q} \tag{3}$$

Where ρ is the density; u the velocity; P the pressure; ν is the laminar kinematic viscosity; g is the gravity; c the heat capacity; T the temperature; t the time; k the thermal conductivity; and $\dot{Q}$ is the heat source per particle. In this study, only the energy conservation equation will be solved.

In the MPS method, the computational domain is discretized into particles and uses the kernel function to calculate the interaction with particles. the kernel function adopted in this study is from Ref. [19]

$$w_{ij} = w(r_{ij}, r_e) = \begin{cases} \left(1 - \frac{r_{ij}}{r_e}\right)^2 & r_{ij} \le r_e \\ 0 & r_{ij} > r_e \end{cases} \tag{4}$$

Where the r_{ij} is the distance between particle i and the neighbor particle j; r_e is the interaction radius ($r_e = 3.1 \cdot l_0$ in 2 dimensions), and is the constant initial particle distance. The particle number density (PND) which is proportional to particle density is defined from kernel function as:

$$n_i = \sum_{j \ne i} w(r_{ij}, r_e) \tag{5}$$

In the MPS method, differential operators are used to discretize different particle interaction models based on the kernel function, and in the energy conservation equation, gradient and Laplacian operator is given by [8]:

$$\nabla \varphi = \frac{d}{n_0} \sum_{j \ne i} \frac{(\varphi_j - \varphi_i)(r_j - r_i)}{\left| r_j - r_i \right|^2} w(r_{ij}, r_e) \tag{6}$$

$$\nabla^2 \varphi = \frac{2d}{\lambda n^0} \sum_{j \ne i} (\varphi_j - \varphi_i) w(r_{ij}, r_e) \tag{7}$$

Where φ is the parameter using the Laplacian operator; d is the number of dimensions; n^0 is the initial PND decided by initial uniform particle configuration; λ the constant parameter dependent on particle size, defined in the following:

$$\lambda = \frac{\sum_{j \ne i} w(r_{ij}, r_e) r_{ij}^2}{n^0} \tag{8}$$

The original MPS method [20] utilizes a semi-implicit algorithm to solve the equations. In each time step, the solution of governing equations is spilt into an explicit part and an implicit part. The energy equation is solved explicitly. To correct the deviation of PND, the pressure field is solved from Pressure Poisson Equation (PPE):

$$\nabla^2 P_i^{n+1} = -\frac{\rho}{\Delta t^2} \left\{ \frac{n^* - n^0}{n^0} - a P_i^{n+1} \right\} \tag{9}$$

Where, P is the pressure, ρ is the density, Δt is the time step, n^* is the temporal PND, a is an artificial compressibility coefficient to smooth the pressure field. After obtaining the pressure field distribution, the pressure gradient term is used to solve the correction value of velocity to ensure the conservation of PND.

2.2 Boundary treatment

In this study, we use the boundary treatment [21] which has been applied to SPH to arrange the thermal boundary to obtain higher accuracy. For the Neumann boundary condition, the temperature value of dummy particles is the same as the internal particle like the original MPS method. The temperature value of dummy particles is calculated before solving the implicit temperature equation, as shown in the following:

$$T_{Dummy} = \left(1+\frac{d_j}{d_i}\right)T_{Boundary} - \frac{d_j}{d_i}T_{Internal} \tag{10}$$

2.3 Time integration

To improve the stability of the algorithm and prevent pseudo-oscillatory solutions or non-physical solutions, the time step in MPS is severely constrained. Whether solving the PPE equation, energy conservation equation, or viscosity term, there is a stable condition to constrain each time step. The minimum time step is selected in the calculation to ensure that all stability conditions are met. [22], respectively by:

$$\Delta t \le C\frac{l_0}{u_{\max}}, 0 < C \le 1 \tag{11}$$

and for explicit temperature calculation:

$$\Delta t \le S\frac{l_0^2}{2\alpha_{\max}}, 0 < S \le 1 \tag{12}$$

Where C is the courant number, l_0 the initial distance between particles, u_{max} is the maximum velocity among all particles, S the diffusion number, α_{max} is the maximum thermal diffusivity ($\alpha = k/\rho c$ The above two conditions are necessary conditions to maintain the accuracy of numerical simulation.

As shown in equation (12), the time step in explicit temperature calculation is limited by particle size and physical properties. In some multiscale simulations, the time step is set by the smallest particle size, or under simulation with high thermal diffusivity, the time step of a single calculation can even reach the order of 10^{-7} s/step, it is difficult to get effective results in limited time. Therefore, it is necessary to increase the time step through the implicit solution to reduce the total calculation steps and then reduce the calculation time.

2.4 Solvers for implicit temperature calculation

In the original MPS method, only the PPE is solved globally. The PPE and energy equation are discretized by the Laplacian operator, for each particle i the discrete PPE equation and the explicit energy equation can be translated into:

$$\left\{\frac{2d}{\lambda n_0}\sum_{j\neq i} w(r_{ij}, r_e) + a\right\}P_i^{n+1} - \sum_{j\neq i}\left[\frac{2d}{\lambda n_0}w(r_{ij}, r_e)P_j^{n+1}\right] = \frac{\rho}{\Delta t^2}\frac{n^* - n^0}{n^0} \tag{13}$$

$$\rho c(T_i^{n+1} - T_i^n) = \frac{2kd\Delta t}{\lambda n^0}\sum_{j\neq i}(T_j^{\,n} - T_i^n)w(r_{ij}, r_e) \tag{14}$$

Where the superscripts n and $n+1$ of the above two formulas represent the previous time step and current time step respectively. Similarly, we can also solve the temperature implicitly like grid-based methods. The equation becomes as follows:

$$\begin{aligned}&\left\{\frac{2dk}{\lambda n_0}\sum_{j\neq i} w(r_{ij},r_e)+\frac{\rho c}{\Delta t}\right\}T_i^{n+1}-\sum_{j\neq i}\left[\frac{2dk}{\lambda n_0}w(r_{ij},r_e)T_{j,Internal}^{n+1}\right]\\&=\frac{\rho T_i^n}{\Delta t}+\sum_{j\neq i}\left[\frac{2dk}{\lambda n_0}w(r_{ij},r_e)T_{j,Dummy}^{n}\right]\end{aligned}\tag{15}$$

Such a formula can be written for every i particle that needs to be solved, and the temperature value of dummy particles is moved to the right of the equation as part of the source term. Due to the different forms of solving equations, it is necessary to find the optimal algorithm for implicit temperature calculation. Meanwhile, the convergence condition of implicit temperature calculation is explored in the same way as those in Ref [23]:

$$\|\mathbf{r}\|_2\,\Delta t=\rho c\Delta T_i^n<\rho c\varepsilon\tag{16}$$

The residual vector $\mathbf{r}$ is defined as the difference between the source term and the product of the sparse coefficient matrix and the approximate solution after an iteration. The convergence value ε (10^{-9}) which is the same as the PPE solution in the implicit temperature calculation.

The coefficient matrix of implicit temperature solution is also a symmetric sparse matrix like the PPE. Therefore, similar to Duan's work [23] in comparing different PPE solvers, the solvers used in the paper in most algorithms in the Krylov subspace. Including CG and D-Lanczos method in Ritz-Galerkin method, CR method in orthogonal method; BICGSTAB, BICRSTAB, and CGS in Petrov-Galerkin method. Meanwhile, the PCG and PBICG which use the preconditioner Diagonal-scaling are applied to implicit temperature calculation.

The CG method is the most common method to solve symmetric positive definite matrices. D-Lanczos is based on the Lanczos algorithm and solves the tridiagonal matrix by using Gaussian elimination with the nonselective main element. It is equivalent to the Lanczos algorithm, but the method with no principal element may cause calculation interruption. CR method is based on the Orthogonalization (minimal residual) method in Krylov subspace. GMRES [24] is an extension of its asymmetric matrix solution. Another extension is GCR [25]. Since only symmetric matrices need to be solved in temperature solution, CR is used in the orthogonalization method to compare with other algorithms. The Petrov-Galerkin method is based on the orthogonality of subspace and constraint space. It is an extension of the Ritz-Galerkin method to asymmetric matrices. Since Lanczos and BICG method was proposed in 1952 [26], there have been many biorthogonal methods based on BICG or BICG biorthogonal idea, including CGS, Bi-Conjugate Residual Stabilized method (BICR) and other methods with higher stability such as BICGSTAB and BICRSTAB.

In this paper, a one-dimensional heat conduction example and a square plate heat conduction example are used to compare the error of implicit and explicit temperature calculation. Subsequently, all the above solvers will be analyzed in terms of convergence, solution error, and computing resource consumption.

3 NUMERICAL EXAMPLES

One-dimension heat conduction, as well as the square plate heat conduction, is studied as two typical cases with different boundary conditions in this paper to evaluate the performance of different solvers. The boundary condition with fixed temperature is named as Dirichlet boundary condition, while the one given a function is called the Neumann boundary condition.

Figure 1 and Figure 2 show the calculating settings of these two cases. To meet the one-dimension requirement, the large length/width ratio is 10. Therefore, a plane with a height of 1.0 m and a width of 0.1 m is employed. The Dirichlet boundary condition is applied on the left edge. Meanwhile, we also analyze the influence of different boundary conditions on implicit heat transfer calculation. The square plate's side length is $L = 1.0$ m. In terms of boundary state, the Dirichlet boundary condition is applied on the right, left, and bottom edges while the upper edge is calculated with the Neumann boundary condition: $\partial T / \partial x = 0$. In both cases, no internal heat source involved there, the initial temperature of the one-dimension plate and the square plate is set to T_0, the temperature of the Dirichlet boundary is assumed T_w constantly.

In order to avoid the independence of particle average distance for the simulation similar to the mesh independent study in the mesh-based method. Different particle size is used to calculate these two, In the calculation of this paper, the particle sizes used in the one-dimension heat conduction are 0.001 m and 0.0005 m. Meanwhile, the initial particle spacing of 0.005 m and 0.002 m is used in the square plate heat conduction.

In the two simulations, all physical properties are constant, and the selected value of properties is within the range of those commonly used in projects. $\rho = 8000\ kg/m^3$, $c = 400\ J/(kg \cdot K)$, $k = 10\ W/(m \cdot K)$, In the following, we will analyze the difference between explicit and implicit in the two examples, and compare the calculation results under different algorithms, so as to find some better algorithm for implicit temperature calculation.

4 RESULTS AND DISCUSSION

4.1 Comparison of explicit and implicit results

In order to verify the accuracy of different algorithms, before comparing the performance of different algorithms, the analytical solution or the results of the original explicit calculation method are used to compare the results of different algorithms.

Figures 3 and 4 show the temperature distribution cloudy map of the implicit results using CG method in the simulation of one-dimension heat conduction and square plate conduction respectively. This result is consistent with that of literature [11].

The particle temperature value is derived to analyze the simulation results. In the two cases, the extracted temperature distribution line is in the middle of the y-axis and parallel to the x-axis. In the one-dimensional heat conduction simulation, the extracted temperature is located in the center of the one-dimensional plate. Since the length/width ratio is 10, the temperature distribution in the center of the plate can be regarded as one-dimensional heat conduction temperature distribution. The analytical solution is as follows:

$$\frac{T(x,t)-T_w}{T_0-T_w} = erf\left(\frac{x}{2\sqrt{\alpha t}}\right) \tag{9}$$

Where $T(x,t)$ denotes the temperature at time t and location x, and erf is error function. In the heat conduction simulation of square plates, the analytical solution is not used for comparison, but the explicit and implicit results are used for comparison. The one-dimensional heat conduction simulation results at 10 s, 20 s, and 30 s, and the square plate calculation results at 1000 s, 5000 s, and 10000 s are output, as shown in figures 5 and 6. As the simulation proceeds, the high-temperature region expands along the x-axis coordinate under the effect of temperature difference. Finally, the parabolic temperature distribution appears in the case of square plate heat conduction. The square plate heat conduction presents a symmetrical temperature distribution. In simulation, both the one-dimensional and square plate heat conduction, the results of the explicit, implicit, and analytical solutions are in good agreement. The temperature values of the simulation results almost coincide.

Fig. 7 shows the relative errors between the implicit results and the analytical solutions in the one-dimension heat conduction case. Fig. 8 shows the implicit and explicit relative errors of square plate heat conduction examples. In the simulation, the explicit calculation uses the diffusion number of 0.2, while the implicit one is 2.0. The implicit calculation can get a stable solution under this diffusion number, but can only get a divergent non-physical result if the same number original method is used. It can be seen that the error gradually decreases with the simulation. The error of the implicit method is larger than the original explicit result, but the overall error is in a small order of magnitude. The implicit results obtained by different algorithms are not very different, even though the relative error curves coincide.

In the one-dimensional heat conduction simulation results, the heat transfer phenomenon mainly depends on the temperature difference, and the temperature change before the 30s is mainly between the dimensionless distance of 0.4 and the fixed temperature wall, so the relative error of temperature is 0 when the dimensionless distance is greater than 0.4. Similar to the temperature distribution, the relative error distribution of square plate heat conduction also presents a symmetric form. The temperature distribution and error distribution on both sides of the square plate at the initial stage are similar to that of one-dimensional. The simulated temperature distribution by all the solvers agrees reasonably with the theoretical results, The performance of different solvers will be compared later.

4.2 Convergence of different solvers

The convergence properties of different solvers are discussed as follows. Different initial particle sizes are used to simulate one-dimensional heat conduction and square plate heat conduction in the two cases, to eliminate the influence of the time step, the same time step is used in the same example. The selection principle of the time step is to select the time step under the same stable dissipation number of smaller particles, so as to ensure the stability of the calculation results of smaller particles.

All the typical convergence profiles by different solvers of two cases are shown in Figures 9 and 10, where the values of the ordinates represent the residual refers to $\|\boldsymbol{r}\|_2 \Delta t$ (left side of equation (16). It can be seen that the number of iterations required for implicit calculation is much less than that for PPE. The reason is that the value in the implicit heat transfer equation $\rho c/\Delta t$ is much larger than a in PPE, which is also consistent with the results in the literature [23].

The convergence patterns are different among these solvers. The convergence curves of CG, CR, and D-Lanczos are stable and smooth. CG and Lanczos are mathematically equivalent, which prove to converge absolutely. D-Lanczos is the direct form of Lanczos which has the risk of infrequent breakdowns. In the calculation of the square plate case with a spacing of 0.002m, D-Lanczos suffered from infrequent breakdowns, but it also converged after reusing some of the principal elements. In the Petrov-Galerkin method, they need fewer iteration steps to converge. The effect of BICRSTAB's convergence is slightly lower than that of BICGSTAB. The convergence curve of BICGSTAB has small fluctuations, while CGS has no fluctuations. At the same time, CGS has fewer iteration steps and smoother curves before convergence. In theory, CGS will perform better in solving the equation with uniform eigenvalue distribution, which is different from solving the PPE. In addition, PCG and PBICGSTAB with Diagonal preconditioning are also used to solve the equation. The results indicate that diagonal preconditioning cannot accelerate convergence, but increases the operation in iteration then increase the consumption of computing resources.

The following table lists the number of component operations required by various solvers for every single iteration. Since multiplication consumes the most computing resources during the program operation, Matrix-Vector (MV) multiplication, Vector Update (VU), and Inner Product (IP) are considered in this paper, The component operations under the same time step are shown in figure 11. The number of MV required by the Petrov-Galerkin method in a single iteration is 2, while most other solvers are 1. However, since the number of iterations required for convergence is also less than other solvers, most solvers use roughly the same number of MV operations, except that CR is twice that of other solvers. For the VU, D-Lanczos needs the most VU operations in each time step, while CGS and CG have fewer VU operations. CGS has a minimum number of IP operations.

5 CONCLUSIONS

This paper discusses an implicit heat calculation module in the MPS method. Comparing the implicit module with the original one. Six Krylov subspace solvers, CG, PCG, CR, D-Lanczos, CGS, BICGSTAB, PBICGSTAB, and BICRSTAB, are examined in terms of accuracy, convergence, and computing resource consumption, and the following conclusions are drawn:

Implicit heat transfer calculation will not introduce large calculation errors and reduce the time step limit under the stability requirements. Due to the difference between PPE and implicit heat transfer equation, the number of iterations to advance a single time step is much less than that of the PPE equation, which will not bring more time consumption.

For the comparison of different algorithms, CG is a recommended solver, CGS has the least IP operations, and other component operations are similar to CG. Therefore, CGS solver can be considered in the subsequent MPI parallel.

REFERENCES

[1] Duan GT and Sakai M. An enhanced semi-implicit particle method for simulating the flow of droplets with free surfaces. *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING* (2022) **389**:1879-2138

[2] Duan GT, A. Yamaji, and M. Sakai. A multiphase MPS method coupling fluid-solid interaction/phase-change models with application to debris remelting in reactor lower plenum. *Annals of Nuclear Energy* (2022) **166**:108697-.

[3] Guo Kailun, et al. Numerical simulation of Rayleigh-Taylor Instability with periodic boundary condition using MPS method. *Progress in Nuclear Energy* (2018) **109.NOV.**:130-144.

[4] Chen R. H. et al. Numerical investigation on coalescence-solid of bubble pairs rising in a stagnant liquid. *Chemical Engineering Science* (2011) **66.21**:5055-5063.

[5] Gingold R. A., and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* **3**:375-389.

[6] Takabatake K.et al. Numerical study on a heat transfer model in a Lagrangian fluid dynamics simulation. *International Journal of Heat and Mass Transfer* (2016) **103**. dec.:635-645.

[7] Monaghan Cjj. Conduction Modelling Using Smoothed Particle Hydrodynamics. *Journal of Computational Physics* (1999).

[8] Koshizuka S. and Y. Oka. Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid. *Nuclear Science & Engineering* (1996) **123.3**:421-434.

[9] Koshizuka S. et al. NUMERICAL ANALYSIS OF CRUST FORMATION IN MOLTEN CORE-CONCRETE INTERACTION USING MPS METHOD.

[10] Yamada, D. et al. Application of improved multiresolution technique for the MPS method to fluid lubrication. *Computational Particle Mechanics* (2021):1-21.

[11] Liang, Y. et al. Numerical models for heat conduction and natural convection with symmetry boundary condition based on particle method. *International Journal of Heat & Mass Transfer* (2015) **88**:433-444.

[12] Zw A, et al. Consistent Robin boundary enforcement of particle method for heat transfer problem with arbitrary geometry - ScienceDirect. *International Journal of Heat and Mass Transfer* 157.

[13] Hestenes M. R. and E. Steifel . Method of Conjugate Gradients for Solving Linear Systems. *J.res.nat.standards* (1952).

[14] Toh K. C. and M. Kojima. Solving Some Large-Scale Semidefinite Programs via the Conjugate Residual Method. *Siam Journal on Optimization* (2002) **12.3**:669-691.

[15] Vorst Henk A. Van Der. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing* (1992) **13**:631.

[16] Sogabe T. and S. Zhang. Extended Conjugate Residual Methods For Solving Nonsymmetric Linear Systems. *Numerical Linear Algebra and Optimization Department of Applied Physics* (2004).

[17] Sonneveld P. CGS, A Fast Lanczos-Type Solver for Nonsymmetric Linear systems. *SIAM Journal on Scientific and Statistical Computing* (1989).

[18] Paige C. C. and M. A. Saunders. Solutions of sparse indefinite systems of linear equations. *Siam J.numer.anal* (1975) **12**.4:617-629.

[19] Masahiro et al. Improvement of stability in moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids* (2011).

[20] Tanaka M. and T. Masunaga. Stabilization and smoothing of pressure in MPS method by Quasi-Compressibility. *Journal of Computational Physics* (2010) **229**.11:4279-4290.

[21] Jwa B et al. Modeling heat transfer subject to inhomogeneous Neumann boundary conditions by smoothed particle hydrodynamics and peridynamics. *International Journal*

of Heat and Mass Transfer (2019) **139**:948-962.

[22] Zhang S. and Morita K. et al. An improved MPS method for numerical simulations of convective heat transfer problems. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS* (2006).

[23] Duan GT and B. Chen. Comparison of parallel solvers for Moving Particle Semi-Implicit method. *Engineering Computations* (2015).

[24] Saad Y. , and M. H. Schultz . "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems." SIAM Journal on Scientific and Statistical Computing 7(1986):856-869.

[25] Yie H. , X. Cai , and J. White. Convergence properties of relaxation versus the surface-Newton generalized-conjugate residual algorithm for self-consistent electromechanical analysis of 3-D micro-electro-mechanical structures. *International Workshop on Numerical Modeling of Processes & Devices for Integrated Circuits IEEE*, 1994.

[26] Lanczos C. Solution of Systems of Linear Equations by Minimized Iterations1. *Journal of research of the National Bureau of Standards* (1952) 49.1.

Table 1: Number of different component operations for different solvers

Solver structure	MV (Matrix-Vector)	VU (Vector Update)	IP (Inner Product)
CG	1	3	2
CR	2	4	2
D-Lanczos	1	6	2
CGS	2	6	2
BICRSTAB	2	9	4
BICGSTAB	2	6	4

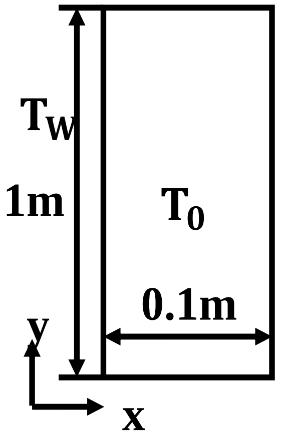

Figure 1: Model of one-dimensional heat conduction

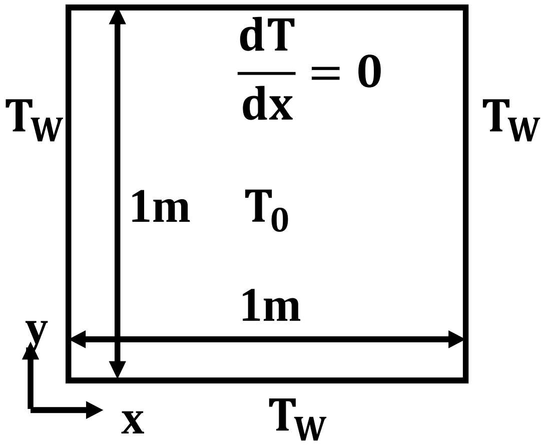

Figure 2: Model of square plate heat conduction

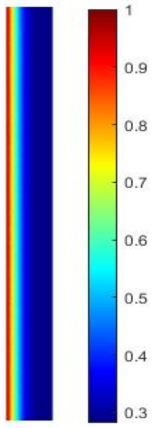

Figure 3: Temperature field in the one-dimensional heat conduction case

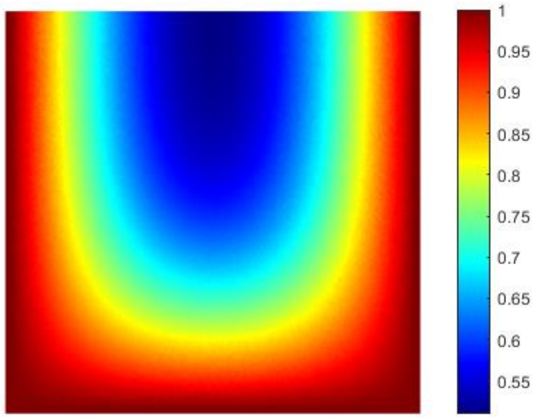

Figure 4: Temperature field in the square plate heat conduction case

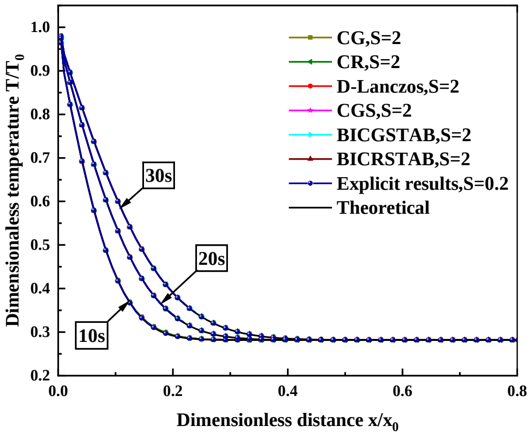

Figure 5: Temperature distribution in the one-dimensional heat conduction case

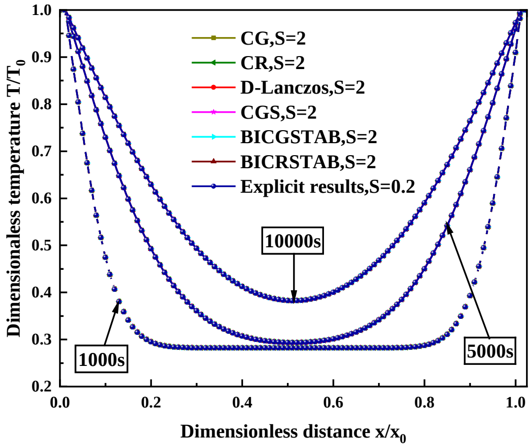

Figure 6: Temperature distribution in the square plate heat conduction case

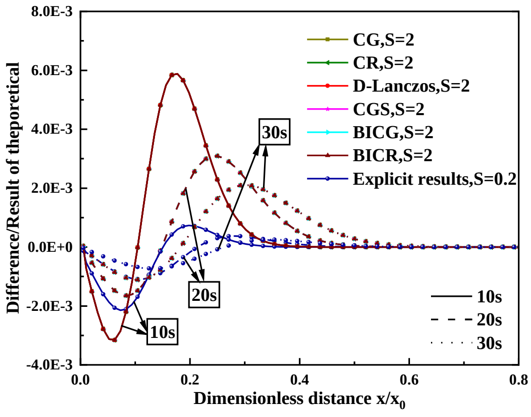

Figure 7: Relative error in the one-dimensional heat conduction case

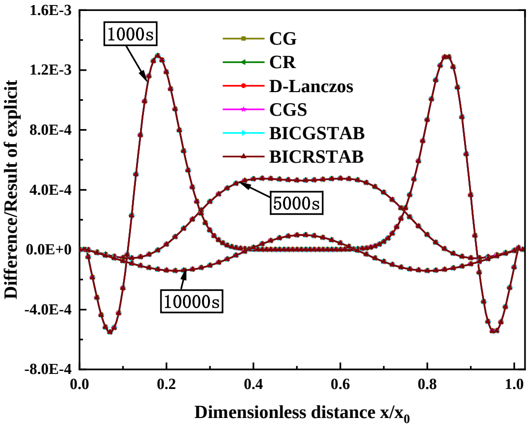

Figure 8: Relative error in the square plate heat conduction case

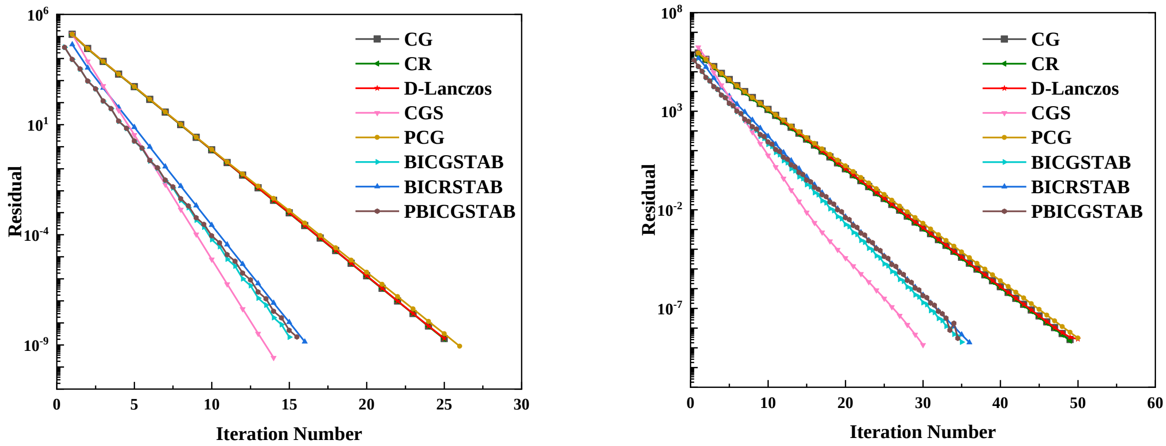

Figure 9: The convergence profiles by different solvers in one-dimensional heat conduction case. (a) initial particle spacing 0.001m (b) initial particle spacing 0.0005m

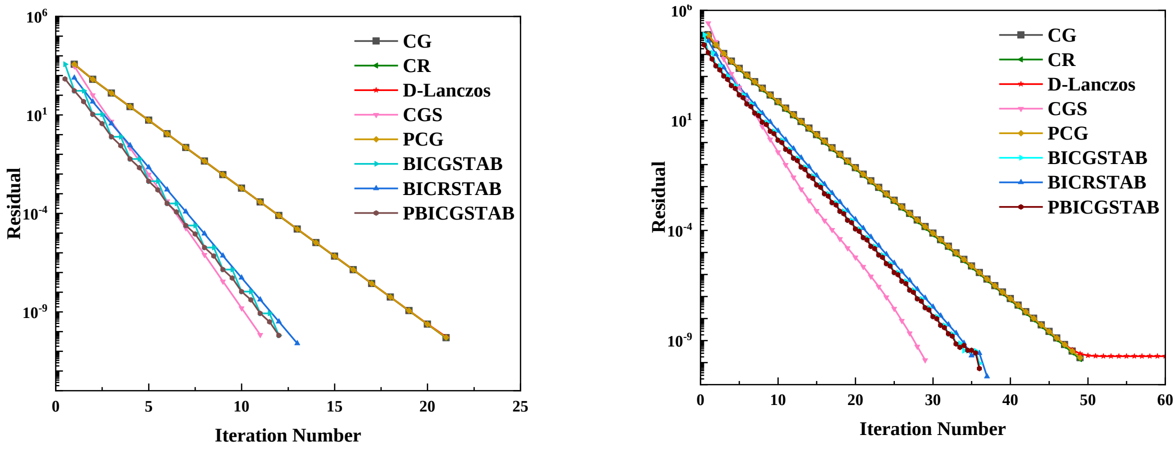

Figure 10: The convergence profiles by different solvers in square plate heat conduction case. (a) initial particle spacing 0.005m (b) initial particle spacing 0.002m

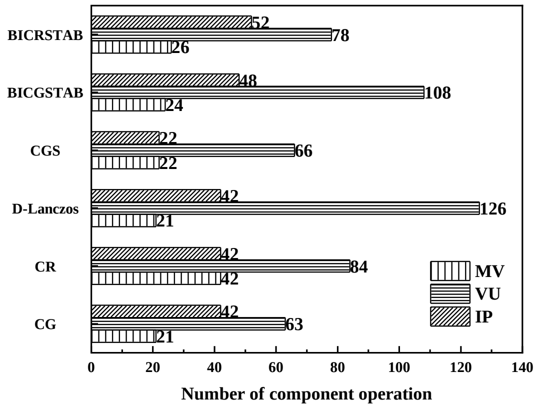

Figure 11: Number of component operation of different algorithms in the same time step