

A new efficient algorithm for multi-scale rough surfaces contact mechanics

Jun Zhang and Yiyong Huang

Aerospace Science and Engineering College, National University of Defense Technology, 41000 Changsha Hunan, China

Abstract. Traditional multi-scale rough surfaces contact algorithms face both computation accuracy and efficiency problems. In this paper, we present a new efficient algorithm, Sliding Window based Gauss-Seidel Algorithm under BEM framework to solve the problems. We slide a window of small size on the contact region to reduce linear equation dimensions. After investigating the influence coefficient matrix, we find that the deformation is only influenced by the pressures around, so our algorithm will be very accurate. By taking the Persson Model results as the baseline to compare all kinds of numerical methods, our algorithm performs better than the conventional algorithms in both accuracy and efficiency.

1 Introduction

Contact mechanics is a basic problem in mechanical engineering, whether the contact performance is good or not directly affects the normal use of seal components, brake parts and so on. Real surfaces are never ideally flat and roughness is present at different scales, from the specimen size down to the interatomic distance. Hence, when two bodies are pressed against each other, contact takes place at the asperities and the real contact area is a fraction of the nominal one. Therefore, the analysis of two surfaces contact problems must be taken into account when scale effect.

The earliest study of contact mechanics can be traced back to 1882, Hertz proposed a contact model between the elastic sphere, and gives the corresponding analytical solutions. In 1956, the multi scale effect should be considered when the elastic contact of rough surface is studied by Archard. In 1966, Greenwood and Williamson proposed the GW model, the model took rough surface description into a series of randomly distributed spherical bulge, they assumed that each asperity deformation independently of each other and used Hertz theory to solve single asperity contact problem. [1] In 1990, Majumdar and Bhushan proposed a model based on fractal contact, with fractal dimension and fractal roughness parameter to characterize a surface roughness, the MB model predicted that the total load was linear with the contact area when load is small. [2] There is an important issue in the models above that the interaction between asperities is not considered. In 2001, Persson proposed a kind of multi scale statistical model, and gave the analytical model of the full contact problem and the approximate solution of the contact problem. Persson theory regarded surface roughness as a stochastic process to analyze, used power spectral density to characterize

surfaces, and concluded the probability density function of load and resolution should conform to the diffusion equation, given the diffusion equation of the boundary conditions and initial conditions can be calculated analytical solution. In a very wide range of loads, the load was linear with the contact area, but the ratio was not the same as the result of the multi asperities model and the numerical method. [3]

The comparison and verification of the existing model results contribute to the progress in understanding the limits of the current approaches and further research. Experimental study on the measurement precision of the problem, it is difficult to make accurate approximation. Therefore, numerical simulation is very important in the study of contact problem. In 2004, Hyun used the Finite Element Method to study the elastic contact problem between fractal rough surfaces. Pei extended the Hyun model to the elasto-plastic contact problem in 2005. In 2006, Persson and Yang verified the contact stress along with the variation of the observed scale in the Persson theory by numerical methods. The key lies in the Boundary Element Method for solving large-scale linear equations, Polonsky and Keer in 1999, MLMS algorithm is proposed to accelerate the solving of equations. In 2001, Borri-Brunetto proposed the ICARUS algorithm based on greedy Gauss-Seidel algorithm and the idea of the real contact area changes with the resolution. [4] The same year, Karpenko, Polonsky and Keer applied Fast Fourier Transform technology to the Boundary Element Method, and achieved good results.

With the increase of the number of boundary element grid, the dimension of the linear system of equations is increasing exponentially, which leads to the ill conditioned system of equations and can not be solved. Therefore, it is necessary to reduce the dimension of the equations. In 2008, Xia put forward a scheme based on

block Gauss Seidel method for study on the contact problems, but the resolution improved to a high level, the algorithm lost its computation efficiency and calculation accuracy. [5] In this paper, a Gauss-Seidel algorithm based on sliding window is proposed to solve the problem of multi scale elastic contact problem.

This paper is structured as follows: in Section 2, we describe the mathematical model of elastic contact problem. Section 3 introduces two conventional algorithms, the Direct Gauss-Seidel algorithm and Block Gauss-Seidel algorithm. Section 4 introduces our algorithm, Sliding Window based Gauss-Seidel algorithm. Section 5 designs examples to compare the algorithms.

2 Mathematical Model

2.1 Fractal surfaces

There are lots of algorithms to generate fractal surfaces, such as Random Midpoint Displacement(MPD), W-M Function Method, Fast Fourier Transform technology etc. Here we use MPD to get random rough surfaces.

MPD is often used in terrain modelling, through the fractal dimension D and the number of iterations n to control the roughness of the generated surface [6].

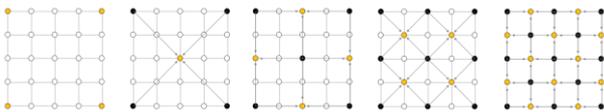


Figure 1. Yellow points mean the generating point in current iteration loop, black points represent the generated points in the iteration loop before.

By MPD we generate a three-dimensional fractal surface, as shown in Fig. 2.

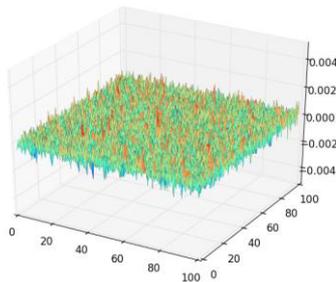


Figure 2. A fractal surface generated by MPD.

2.2 The model

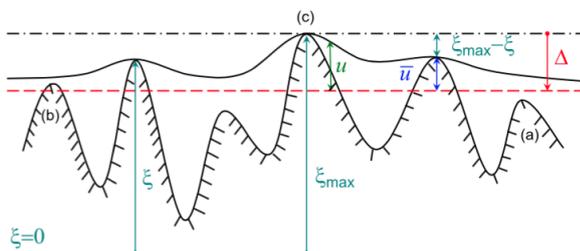


Figure3. Sketch of the contact problem between a rigid rough surface and an elastic half-plane. Its deformed configuration corresponding to the imposed far-field displacement Δ is

depicted with a black solid line. The red dashed line corresponding to a rigid-body motion of the half-plane identifies the heights to be included in the initial trial contact domain.

In this paper, we assume that one of the contact surfaces is a flat elastic one, another is a rough rigid surface with fractal characteristics.

By imposing a far-field closing displacement Δ to the bodies in contact, the displacement at each point of the contact area is related to the contact pressures as follows [7]:

$$u(x) = \int_S H(x, y)p(y)dS \quad (1)$$

where $u(x)$ is the displacement at the surface point defined by the position vector x , $H(x, y)$ is the displacement at x due to a unit pressure acting at y , and S is the apparent contact area. Assuming linear elastic isotropic materials, the influence coefficients are given by [7]

$$H(x, y) = \frac{1-\nu^2}{\pi E} \frac{1}{\|x-y\|} \quad (2)$$

where E and ν denote, respectively, the composite Young's modulus and Poisson's ratio of the materials of the bodies in contact, respectively.

Upon discretization of the surface as a grid where each nodal height defined by the indices (i, j) is modelled as a square punch with an elevation $\xi_{i,j}$ above a reference plane coincident with the level of the smallest surface height, the contact problem requires the simultaneous solution of the following set of equations and the satisfaction of a set of inequalities [7]:

$$u_{i,j} = \sum_{k=1}^{N_c} \sum_{l=1}^{N_c} H_{i-k,j-l} p_{k,l}, \quad 1 \leq i \leq N_c, \quad 1 \leq j \leq N_c$$

$$u_{i,j} = \bar{u}_{i,j} = \Delta - \xi_{i,j}, \quad \forall (i, j) \in I$$

$$p_{i,j} > 0, \quad \forall (i, j) \in I$$

$$u_{i,j} > \bar{u}_{i,j}, \quad \forall (i, j) \in \bar{I}$$

$$p_{i,j} = 0, \quad \forall (i, j) \in \bar{I}$$

where I is the domain of boundary elements in contact, \bar{I} is the domain of boundary elements not in contact, and Δ is the imposed far-field displacement.

3 Conventional Methods

The focus and difficulty of solving the section 2 model is to solve the equations $Hp = \bar{u}$ rapidly and accurately. This section will introduce two conventional methods.

3.1 Direct gauss-seidel algorithm

The algorithm considers that the deformation of each point is affected by all the points in the plane, and the influence decreases with the increase of the distance from the point. [8]

Algorithm description: Direct Gauss-Seidel Algorithm

Input: coefficient matrix H , vector \bar{u} , initial pressure vector p , initial contact set I_0 , maximum number of iterations K_{max} , error limit ϵ .

Algorithm Procedure:

Step 1 $k := 0, I := I_0$

Step 2 while $\left(\begin{matrix} k \leq K_{max} \text{ and} \\ \min(p) < -\epsilon \text{ or} \\ k = 0 \end{matrix} \right)$ do:

(2.1) Gauss-Seidel algorithm is used to solve the linear equations $Hp = \bar{u}$ to get the solution vector p

(2.2) for $(i, j) \in I$ do:

(2.2.1) if $p_{i,j} < -\epsilon$ then

$p_{i,j} := 0$ and $I := I \setminus \{(i, j)\}$

Step 3 $p_{opt} := p$

Step 4 $\bar{I} := \{(1, \dots, i, \dots, N_c) \times (1, \dots, i, \dots, N_c)\} \setminus I$

Output: dimensionless contact area

The main steps of the contact algorithm are summarized above, where H denotes the matrix collecting the influence coefficients, the vector \bar{u} collects the imposed displacements, and the vector p collects the contact forces. The boundary elements subject to tensile forces are eliminated from the contact domain at the step (2.2.1). After convergence, the optimal solution is stored in the vector p_{opt} in the step 3. The set of boundary elements not in contact \bar{I} , is updated in the step 4.

3.2 Block gauss-seidel algorithm

The overall process of the Block Gauss-Seidel algorithm is consistent with 3.1, the difference is that the huge linear equations are divided into block processing, the dimension of the H matrix is reduced to $N_c \times N_c$, and then apply the Gauss-Seidel algorithm to solve. The algorithm considers that the deformation of each point is only related to all points of the row of the point, which is not related to the point of other rows. [5]

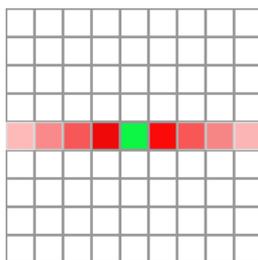


Figure 5. The influence of one point is only related to points in this row

4 Our Algorithm

Linear equations of the dimension disaster problem exist in the Direct Gauss-Seidel algorithm, when the dimension of the matrix reaches a high level, it can't be solved. The Block Gauss-Seidel algorithm using block calculation to reduce the dimension, but to each point on effects into consideration is not comprehensive, resulting in the decline of accuracy. This section will first analysis the characteristics of the coefficient matrix H , then describe Gauss-Seidel algorithm based on sliding window.

4.1 Analysis of influence coefficient matrix

Investigate the influence coefficient matrix H , $H_{i,j}^{k,l}$ denotes the influence between point (i, j) and point (k, l) , which are known from the literature [7]

$$H_{i,j}^{k,l} = \alpha \left\{ \begin{aligned} & (\bar{x}+a) \ln \left[\frac{(\bar{y}+a) + \sqrt{(\bar{y}+a)^2 + (\bar{x}+a)^2}}{(\bar{y}-a) + \sqrt{(\bar{y}-a)^2 + (\bar{x}+a)^2}} \right] \\ & + (\bar{y}+a) \ln \left[\frac{(\bar{x}+a) + \sqrt{(\bar{y}+a)^2 + (\bar{x}+a)^2}}{(\bar{x}-a) + \sqrt{(\bar{y}+a)^2 + (\bar{x}-a)^2}} \right] \\ & + (\bar{x}-a) \ln \left[\frac{(\bar{y}-a) + \sqrt{(\bar{y}-a)^2 + (\bar{x}-a)^2}}{(\bar{y}+a) + \sqrt{(\bar{y}+a)^2 + (\bar{x}-a)^2}} \right] \\ & + (\bar{y}-a) \ln \left[\frac{(\bar{x}-a) + \sqrt{(\bar{y}-a)^2 + (\bar{x}-a)^2}}{(\bar{x}+a) + \sqrt{(\bar{y}-a)^2 + (\bar{x}+a)^2}} \right] \end{aligned} \right\} \quad (4)$$

where $\alpha = \frac{1-v^2}{\pi E}$,

a is a single grid length, $\bar{x} = x_i - x_k, \bar{y} = y_j - y_l$
 Inspection point (6,6) influence coefficient change law, make k, l from 1 to 11, get the figure below

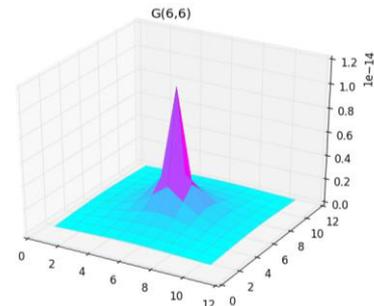


Figure 6. The distribution of influence coefficient of Point (6,6) on the plane.

From the figure can be drawn: the deformation is mainly related to the pressure near the point of several points on a certain point, the influence of further points is very small, based on the results, our algorithm is proposed.

4.2 Sliding window based gauss-Seidel algorithm

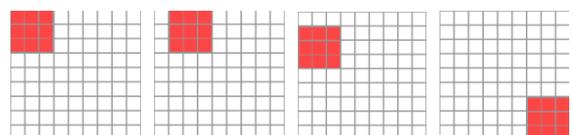


Figure 7. 3*3 sliding window demo.

Algorithm description: Sliding Window Based Gauss-Seidel Algorithm

Input: coefficient matrix H , window size w , vector \bar{u} , initial pressure vector p , initial contact set I_0 , maximum number of iterations K_{max} , error limit ϵ .

Algorithm Procedure:

Step 1 $k := 0, I := I_0$

Step 2 while $\left(\begin{matrix} k \leq K_{max} \text{ and} \\ \min(p) < -\epsilon \text{ or} \\ k = 0 \end{matrix} \right)$ do:

(2.1) Using Gauss-Seidel algorithm to solve the linear equations of dimension $w \times w$, the solution is the vector p within the window

(2.2) for $(i, j) \in I$ do:

(2.2.1) if $p_{i,j} < -\epsilon$ then

$p_{i,j} := 0$ and $I := I \setminus \{(i, j)\}$

Step 3 $p_{opt} := p$

Step 4 $\bar{I} := \{(1, \dots, i, \dots, N_c) \times (1, \dots, i, \dots, N_c)\} \setminus I$

Output: dimensionless contact area

Direct method in each iteration need solving $N_c \times N_c$ groups of $N_c \times N_c$ dimension linear equations, block method in each iteration requires solving N groups of $N_c \times N_c$ dimension linear equations, compared to them, the algorithm in this paper uses a sliding window of a small size sliding solution on the surface, in each iteration requires solving $(N_c - w + 1) \times (N_c - w + 1)$ groups of $w \times w$ dimension linear equations. When w is much smaller than N_c , our algorithm can approximate solution $N_c \times N_c$ groups of $w \times w$ dimension linear equation group. Not only to reduce the dimension of the purpose, but also to consider the deformation of each point by the window range of the impact of each point, which greatly improves the computational efficiency, but also the accuracy of the calculation.

Section 5 will design examples to compare the algorithms.

5 Example and Analysis

5.1 Fractal surfaces with different parameters

In this paper, the MPD algorithm is used to generate fractal surface, and the surface is generated by setting the fractal dimension D and the resolution m . The relationship between the resolution and the number of iterations of the generated surface in section 2 is $m = n - 1$

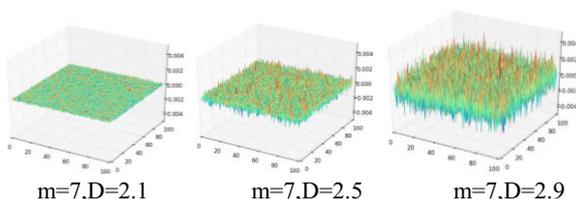


Figure 8. fractal surfaces where m=7

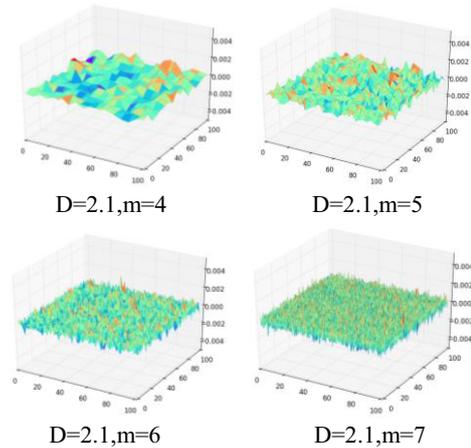


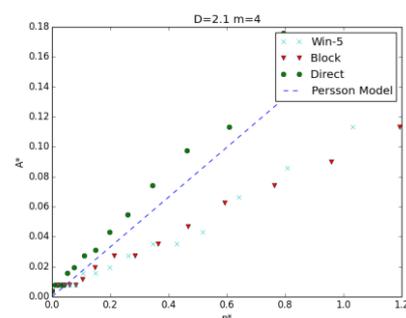
Figure 9. fractal surfaces where D=2.1

We investigate both the effect of the fractal dimension D , see examples in Figure 8, and the effect of the surface resolution by varying the generation parameter m , see Figure 9. The larger m and the larger D will make the surface rougher.

5.2 Comparison of algorithms

The simulation process is controlled by the imposed far-field displacement Δ , and the max Δ is the difference between ξ_{max} and the reference height see Figure 3. This paper selects a range of Δ , and is divided into 15 steps to carry out simulation. Definition of dimensionless pressure $p^* = \frac{pL}{\sigma E}$ and dimensionless contact area $A^* = A_r/A_0$, here p is the total pressure, L is the side of the contact surface, σ is r.m.s. of the distribution of surface heights, E is Young's modulus of elasticity, A_r is the real contact area, A_0 is the nominal contact area.

To investigate the effectiveness of our algorithm, we run groups of simulations on different parameters including sets of m and D , to compare Direct method, Block method and our algorithm (here we let window size=5) with the theoretical solution of Persson Model [4], see Figure 10, the results of $D=2.1$ $m=4,5,6$. Intuitively, when $m=4$, in a narrow range of p^* , errors of three numerical methods are very small, when p^* reach larger, Direct method performances best. But when m reaches larger to 5 and 6, our algorithm gets state-of-the-art results, outperforms the conventional algorithms.



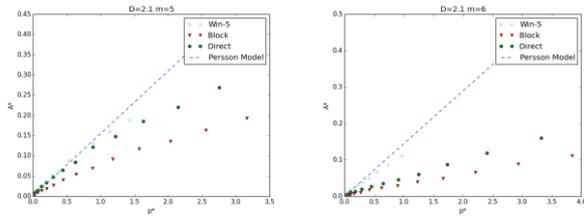


Figure 10. The comparison among algorithms where $D=2.1$ $m=4,5,6$

5.3 Effect of window size

The window size determines the accuracy and efficiency of our algorithm, if too large, it will affect the calculation speed; if too small, it will affect the calculation accuracy. It's meaningful to study the effect of window size. Here, we set the window size from 3 to 7, to compare the accuracy of different window sizes, we let the Persson Model and Direct Method as the baselines, to contrast the efficiency, we compare the log CPU time of different window sizes with the Direct Method.

5.3.1 Accuracy comparison

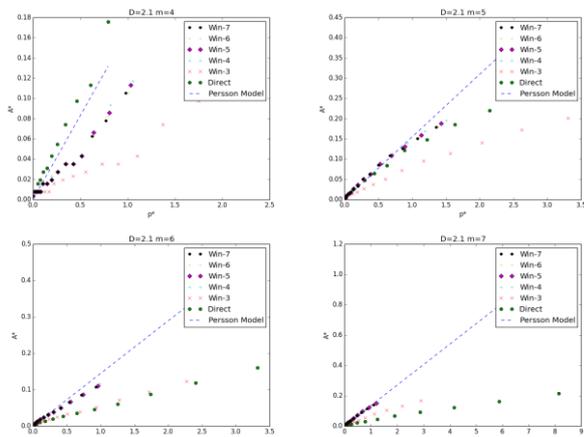


Figure 11. The accuracy comparison among different window sizes, Direct Method, Persson Model where $D=2.1$ $m=4,5,6,7$

See Fig. 11, the accuracy increases with the increase of the window size as expected, but the difference is not big when window size is more than 3, especially when m is larger than 5.

5.3.2 Efficiency comparison

See Fig. 12, the computation is faster when the window size is smaller. The Direct Method is computationally faster than our algorithm when window size is larger than 3 when m equals to 4, but with the growth of m , the

Direct Method computation time increases exponentially, is far more than our algorithm.

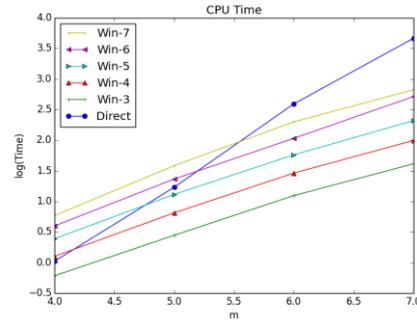


Figure 12. The efficiency comparison among different window sizes and Direct Method

5 Conclusions

We present a new efficient numerical algorithm for solving multi-scale rough surfaces contact mechanics problem under Boundary Element Method framework considering the drawbacks of both conventional methods, Direct Gauss-Seidel Method and Block Gauss-Seidel Method. In this paper we consider both the dimension reduction and accuracy, propose the Sliding Window based Gauss-Seidel algorithm. By taking the Persson Model results as the baseline to compare all kinds of numerical methods, our algorithm performs better than the conventional algorithms in both accuracy and efficiency.

References

1. J.A. Greenwood, J.B.P. Williamson, Proc. R. Soc. Lond. Ser. A **295**(1966)
2. A. Majumdar, B. Bhushan, ASME J. Tribol. **112** (1990)
3. B.N.J. Persson, J. Chem. Phys. **115** (2001)
4. M. Borri-Brunetto, A. Carpinteri, B. Chiaia, Int. J. Fract. **95** (1999)
5. Xia Boqian, Li Ying, Journal of Mechanical Strength, **30**(5) (2008)
6. H.O. Peitgen, D. Saupe, The Science of Fractal Images, Springer-Verlag, New York (1988)
7. J.R. Barber, Elasticity, 3rd edn., Springer, Dordrecht (2010)
8. M. Paggi, Q.-C. He Wear **336-337** (2015)