

## Parallel Algorithm of Monte Carlo Method for Computing Infrared Transmittance in Particle Cloud

Wang Zhenhua<sup>1,a</sup>, He Zhihong<sup>1,b</sup>, Liang Hong<sup>1,c</sup>, Dong Shikui<sup>1,d\*</sup>

<sup>1</sup>School of Energy Science and Engineering, Harbin Institute of Technology, Harbin 150001, China

<sup>A</sup> email: 12B902005@hit.edu.cn, <sup>d\*</sup> email: dongsk@hit.edu.cn

**Keywords:** Particle Cloud; Infrared Radiation; Monte Carlo Method; Parallel Computing

**Abstract.** In this paper, the random number was generated by skip algorithm from parallel computation, so each thread can have independent random function distribution, and then can track different emission direction, to achieve the parallel computing by using Monte Carlo Method (MCM). Calculation results show that the parallel computing by using MCM can not only hold high calculation accuracy, but also reduce computation time and achieve high computation efficiency, when computing the infrared transmittance of particle cloud.

### Introduction

Due to the stochastic nature of emission rays, there is very little association among different ray directions, so the MCM for solving radiation transfer has very good parallel characteristics. In recent years, with the development of parallel computing tools of MPI (Message Passing Interface) and GPU (Graphics Processing Unit), the efficient parallel computing for solving radiation heat transfer began develop rapidly. In 2010, Hadjidoukas [1] developed a high performance version of the Monte Carlo simulation code MC4 to simulate the radiation particle transport, the speedup results are almost linear on all systems. In 2011, Frezzotti [2] coupled the flow using Finite Difference Method (FDM) with radiation heat transfer using MCM with parallel computing, makes the proposed method of solution a viable alternative to particle simulations for studying unsteady low Mach number flows. In 2013, Sudheer [3] studied the load balance of parallel computing using MCM, in the same year, Ishii [4] parallelized the MCM code to obtain solutions of radiation coupling with flow field in the combustor of double nozzle jet engine, and obtained good efficiency, too. In 2014, Liu [5] parallel computed the radiation transfer of CT imaging based on GPU using MCM, and compared the advantages and disadvantages between MPI and GPU.

In the study of infrared transmission of particle cloud, the computation time always is ten thousands seconds, there is an urgent need to adopt an effectively algorithm to reduce the computation time. So in this paper, the parallel computing using MCM will be introduced into the particle cloud infrared transmittance research of particle cloud, to improve the research efficiency.

### MCM for solving radiation heat transfer

The basic idea of MCM for solving radiation heat transfer is: The transfer process is broken up to a series of independent sub-processes, such as emission, transmission, refraction, absorption and scattering and so on. Each process is in the formation of random problem. Each unit (surface element or body element) emits a certain amount of rays (beam or photon emission), then the trail of each ray is tracked (absorbed by medium or interface, radiate into or out from the medium), then the statistical results of the unit obtain the radiation heat transfer distribution [6]. When solving the infrared transmittance of particle cloud, each light ray is without energy, the radiation transfer factor  $RD_{i,j}$  is calculated by random probabilistic model. The radiation transfer factor is defined as: In a certain detection direction, the radiation ray emitted from face element  $i$  into the cloud, then radiation out from face element  $j$  after a series of absorption and scattering, the proportion of the ray out from element  $j$  is radiation transfer factor  $RD_{i,j}$ . Due to the detection direction is determined when computing cloud transmittance, so the direction from target to surface element. But the

incident point of the face element need a random probability simulation, and the transfer length and the scattering direction when the ray is scatted also need random probability simulation.

### Random probability simulation of incident point

In a three dimensional rectangular volume under Cartesian Coordinates  $(X, Y, Z)$ , the coordinate figure range of the grid volume of  $i^{\text{th}}$  along  $X$  coordinate,  $j^{\text{th}}$  along  $Y$  coordinate,  $k^{\text{th}}$  along  $Z$  coordinate is defined as  $(x_{i,\min}, x_{i,\max})$ ,  $(y_{j,\min}, y_{j,\max})$ ,  $(z_{k,\min}, z_{k,\max})$ ,  $R_x, R_y, R_z$  are distributed random number which are in the interval of  $[0,1]$ , then the coordinate figure of the incident point is:

$$\begin{aligned} x_{\text{rad}} &= R_x (x_{i,\max} - x_{i,\min}) + x_{i,\min} \\ y_{\text{rad}} &= R_y (y_{j,\max} - y_{j,\min}) + y_{j,\min} \\ z_{\text{rad}} &= R_z (z_{k,\max} - z_{k,\min}) + z_{k,\min} \end{aligned} \quad (1)$$

### Random probability simulation of radiation direction

In the coordinates  $(X, Y, Z)$  of the incident direction, the zenith angle  $\theta$  and circumference angle  $\varphi$  of the scattering direction are defined as:

$$\theta = \arccos(\sqrt{1 - R_\theta}) \quad (2)$$

$$\varphi = 2\pi R_\varphi \quad (3)$$

where,  $R_\theta, R_\varphi$  are random numbers of zenith angle and circumference angle, respectively.

### Random probability simulation of transfer length

When the radiation energy transmits into the absorption and scattering medium, it will be attenuated. In the MCM, the probable propagation length of the ray is used to simulate the attenuation, as Eq.(4):

$$s = \frac{1}{\beta} \ln(1 - R_s) \quad (4)$$

where,  $S$  is probable propagation length,  $\beta$  is attenuation coefficient,  $R_s$  is the random number of the propagation length.

### Parallel computing of Monte Carlo method

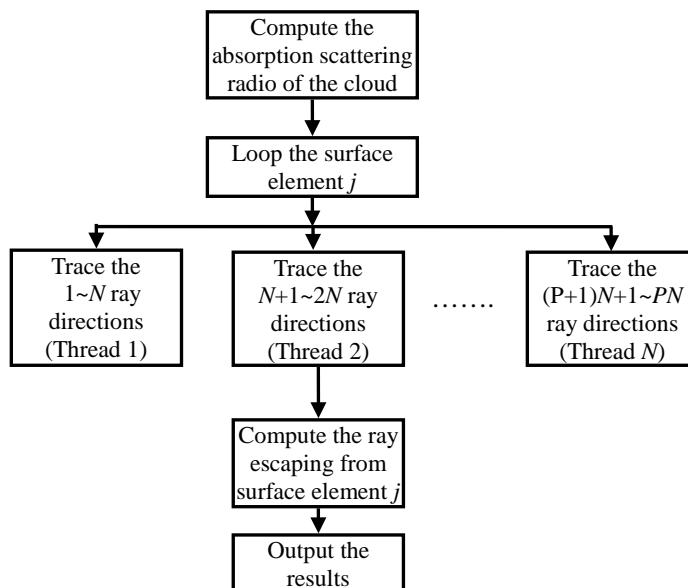


Fig.1 Parallel algorithm of MCM

Computing infrared transmittance of cloud using MCM has good inherent parallel calculation

conditions, mainly includes two points: 1) Large parallel grain, each surface element will emit million ray to meet the random simulation accuracy, so each thread will have to be assigned  $N$  rays, and track them with many different light directions, so the MCM has large parallel computing characteristics. 2) The calculated data is independence, when tracing ray, each tread emits incident ray randomly, and traces them in the medium independence. There is little data association between each thread. So solving the cloud transmission using MCM is very suitable for parallel computing, the parallel algorithm is shown in Fig. 1.

When parallel computing is used for MCM, there are two problems need be to be solved, one is the uncorrelated random number in each thread, the other is statistical problem of the computed results. The former problem mainly involves the mathematical problem of random number generation in parallel computing, the latter one mainly involves the data passing in parallel computing, these two issues will be two major difficulties which need to be addressed in the article.

### Random number generation algorithm in parallel computation

Because the random number of MCM is the pseudo-random number generated by recursive formula in computer, the random number series in each thread will be same when parallel computing is used. As shown in Fig.1, the ray directions computed in thread 1 may be same as them in thread 2, both of them just calculate the  $1\sim N$  ray directions, but the  $N+1\sim PN$  ray directions are not been calculated, so the random statistical characteristic will lose, that will makes parallel computing lost its practical significance. In order to solve this problem, the random number is generated by skip parallel computing of Multiple Recursive Generator (MRG) [7]. The recursive formula for the serial random number generator (SRNG) is:

$$R_{n+1} = k \cdot R_n \pmod{M}, \quad n = 0, 1, \dots \quad (5)$$

Where  $k$  is multiplier,  $M$  is modulus. If the initial value  $R_0$  is given, the random number sequence  $\{R_n\}$  can be calculated by Eq. (7). The recursive formula for skip parallel random number generator (PRNG) is:

$$R_{p,n+1} = A \cdot R_{p,n} \pmod{M}, \quad p = 0, 1, \dots, P \quad n = 0, 1, \dots \quad (6)$$

Where  $R_{p,0}$  is initial value in thread  $P$ . We can generate different seeds of different random number sequence according to different thread number, thus obtains different  $R_{p,0}$ ;  $A$  is generalized multipliers, and its expression is shown as Eq.(7).

$$A \equiv k^D \pmod{M} \quad (7)$$

Where  $D$  is prime number, different threads us different prime to eliminate the correlation of random number in different thread, which can generate high quality random numbers from original random number sequence.

### Data statistics in parallel computing

As shown in Fig.1, the absorption/scattering coefficient of the cloud are computed by thread 1, while other threads waiting, after that, the broadcast function of MPI, MPI\_BCAST, will be used to send the data computed by thread 0 to the other child threads, so that all threads can finish them initialization at the same time, then different thread can trace different ray directions emitted from the same surface element  $i$ . When all threads complete the tracing, the send and receive functions of MPI, MPI\_SEND and MPI\_RECV, will be used to send the ray number  $L_p$ , passing from surface element  $j$ , to the main thread 0, then the transmittance  $RD_{i,j}$  is calculated by Eq. (8):

$$RD_{i,j} = \sum_{p=1}^P L_p / (P \cdot N) \quad (8)$$

When the cloud transmittance is parallel computed using MCM, the data transfer only be needed at the time of initialization and statistical, so the data transfer, which is the main factor affecting the parallel efficiency [8], take a little proportion in the calculation, so the parallel computing can obtain high efficiency.

## Parallel computing simulation performance

To test the computing performance of parallel computing using MCM for computing cloud infrared transmittance, the infrared transmittance of a spherical cloud is calculated, as shown in Fig. 2.

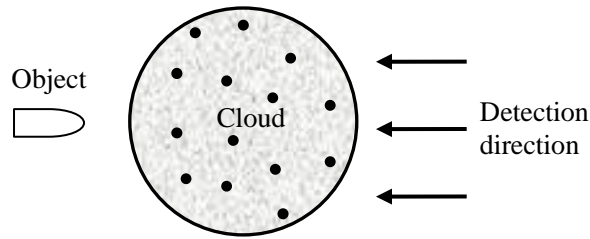


Fig.2 model of spherical cloud

The radius of the particles cloud is 100m, the particles is water droplet, the radius of the water droplet is  $2\mu\text{m}$ , the complex refractive of it at the wavelength of  $1.06\mu\text{m}$  is  $m=1.326-i4.18\text{E-}6$ , the absorbs and scattering cross section can be calculated by Mie scattering theory [9]. Each surface element emits  $10^6$  rays. Meanwhile, 10 sets of dual core PC are connected together by router, MPICH2 is used as parallel transmission tool. Then a small parallel workstation is set up. In this paper, the infrared transmittance is parallel computed by 1, 5, 10, 15, 20 CPUs respectively.

### Analysis of parallel computing accuracy

In this paper, PRNG is validated at first. The transmittance is parallel computed by PRNG and SRNG respectively to verify their influence on the calculation results, as shown in Fig.3 ( $P=20$ ).

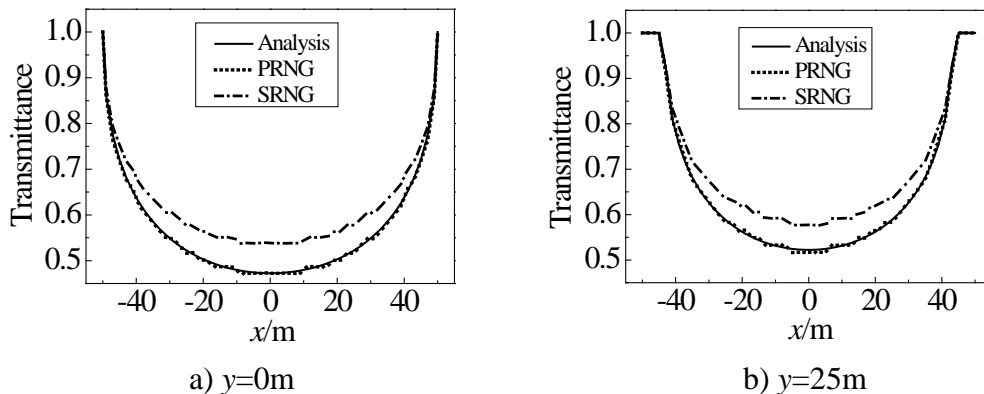


Fig.3 Calculation results comparison of SRNG and PRNG

It can be seen from Fig.3, if SRNG is used for MCM parallel computing, the calculation results will have large errors compared with them calculated by analysis method, but when the PRNG is used for MCM parallel computing, the calculation results will agree with analysis results very well, with very high accuracy. That is because, when there are  $P$  threads for parallel computing, the ray number assigned for each thread is only  $1/P$  of the original ray number, while each thread produces the same random number sequence if SRNG is used, so the ray directions calculated by random number will also be the same, that means only  $1/P$  of original rays are simulated, which case the inaccurate results. When PRNG is used for MCM parallel computing, each thread generates different random number sequences, so they will simulate different ray directions based on the different random number sequences, then each thread traces different  $L_N/P$  ray directions. Finally, the total traced ray directions are same as them of serial calculation, so the precision can be maintained compared with serial computing.

### Analysis of parallel computing efficiency

The purpose of parallel computing is to keep the original calculation accuracy and reduce the computation time to improve the computation efficiency. Therefore, the wall time of the parallel computation is needed to verify the parallel efficiency of parallel computing. The relationship between the run time  $T$  of parallel computing and the number of threads  $P$  is shown in Fig.4.

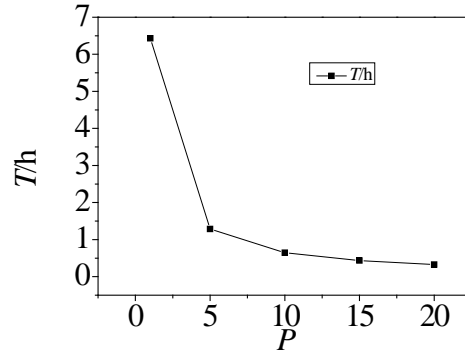


Fig.4 Relationship between run time and threads number

It can be seen from Fig.4, the total run time  $T$  decreases with the increase of threads number  $P$  exponentially, the total run time without parallel computing ( $P=1$ ) is about 6.5 hours, when 20 CPUs are used for parallel computing, the total run time reduces to 0.3 hours, the run time is about 1/20 of original level, so the MCM parallel computing can reduce the calculation time of infrared transmittance of cloud significantly. Write  $T(P)$  as the total run time with  $P$  threads, then  $T(1)$  is the total run time of serial computing. Usually, the parallel performance is characterized by parallel efficiency  $\eta$  [10]:

$$\eta = S / P \quad (12)$$

The total run time with different threads number is used to compute the parallel efficiency, as shown in Fig. 5.

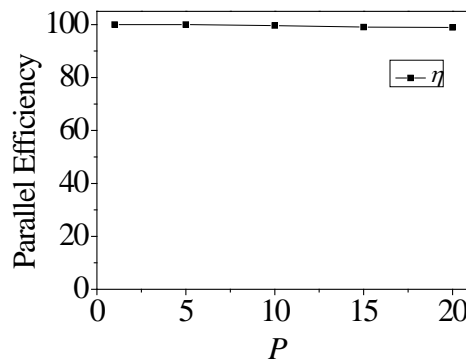


Fig.5 parallel efficiency with different threads number

It can be seen from Fig. 5, when the infrared transmittance of cloud is computed by MCM parallel computing, the speedup changes linearly with the threads number increases, when there are 20 CPUs used, the speedup reaches to 20, this shows its good scalability. Compared with the Spatial Domain Decomposition Parallelization (DDP) for solving radiation heat transfer [10], the parallel efficiency doesn't reduce with the threads number increases, is kept above 99%. That's because when MCM parallel computing is used for solving infrared transmittance of cloud, time consuming mainly occurs in the ray tracing calculations, the data transfer time which can influence the parallel efficiency shares a very small proportion, so the parallel efficiency can hold in a high rang.

## Conclusion

In this paper, the MCM parallel computing is applied to the infrared transmittance calculation of particle cloud. Parallel random number generator (PRNG) is used to acquire independent random number sequence in different thread, and MPI is used to calculate the statistical results, as the two main points. The results show that the PRNG has a great effect on the result of MCM parallel computing.

At the same time, the MCM parallel computing can not only keep good calculation accuracy, but also can reduce the total run time significantly, when it is used to solve infrared transmittance of cloud. When there are 20 CPUs are used, the parallel speedup can reach to 20. It can be seen that,

the MCM parallel computing for solving infrared transmittance of particle cloud has good adaptability, and has good application value for the study of solving infrared transmittance of particle cloud.

## Acknowledgement

The authors are grateful to the anonymous reviewers for their critical and constructive review of the manuscript. This study was co-supported by The National Nature Science Foundation of China (No.51176039).

## References

- [1]. P.hadjidoukas, C.Bousis, D. Emfietzoglou. Parallelization of a Monte Carlo particle transport simulation code, *J, Comput Phys Commun.* 181(2010) 928-936
- [2]. A.Frezzotti, G.P.Ghiroldi, L.Gibelli. Solving the Boltzmann equation on GPUs, *J. Comput Phys Commun.* 182(2011) 2445-2453
- [3]. C.D.Sudheer, S.Krishnan, A.Srinivasan, P.R.C.Kent. Dynamic load balancing for petascale quantum Monte Carlo applications: The Alias method, *J. Comput Phys Commun.* 184(2013) 284-292
- [4]. Ayako Ishii, Naofumi Ohhishi, Hiroki Nagakura, Hirotaka Ito, Shoichi Yamada. Parallel computing of radiative transfer in relativistic jets using Monte Carlo method, *J. High Energ Dens Phys.* 2(2013) 280-287.
- [5]. T.Liu, X.G.Xu, C.D.Carothers. Comparison of two accelerators for Monte Carlo radiation transport calculations, Nvidia Tesla M2090 GPU and Xeon Phi 5110p coprocessor: A case study for X-ray CT imaging dose calculation, *J. Ann Nucl Energy.* 82(2014) 230-239.
- [6]. J.D. Densmore, H.Park, A.B. Wollaber, R.m. Rauenzahn. Monte Carlo Simulation Methods in Moment-Based Scale-Bridging Algorithms for Thermal Radiative-Transfer Problems, *J. Comput Phys.* 284(2015) 40-58.
- [7]. Ashok Srinivasan, Michael Mascagni, David Ceperley. Testing Parallel Random Number Generators, *Parallel Comput.* 29(2003) 69-94.
- [8]. Wang Zhenhua, He Zhihong, Mu Lei, Dong Shikui. Parallel algorithm and its convergence of spatial domain decomposition of discrete ordinates method for solving radiation heat transfer problem, *Chinese J Aeronaut.* 28(2015) 77-85.
- [9]. Janna M. Dlugach, Michale I. Mishchenko. Effects of Nonsphericity on the Behavior of Lorenz-Mie Resonances in Scattering Characteristics of Liquid-Cloud Droplets, *J Quant Spectrosc Ra.* 146(2014) 227-234.
- [10]. Bobby Ohilip, Mark A. Berrill, Srokanth Allu, Steven P.Hamilton, Rahul S.Sampath, Kevin T.Clarno, Gary A.Dilts. A parallel multi-domain solution methodology applied to nonlinear thermal transport problems in nuclear fuel pins, *J. Comput Phys.* 286(2015) 143-171.
- [11]. G.Colomer, R.Borrell, F.X.Trias, I.Rodriguez. Parallel algorithms for Sn transport sweeps on unstructured meshes, *J. Comput Phys.* 232(2013) 118-135.