Efficiency of Monte Carlo method to solve nonlinear coagulation equation when implemented on parallel computers (including GRID infrastructure)*

M. A. Marchenko
Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, Novosibirsk, Russia
Novosibirsk State University, Russia
e-mail: mam@osmf.sscc.ru

Рассмотрен параллельный алгоритм прямого статистического моделирования для решения нелинейного уравнения коагуляции в пространственно неоднородном случае. Исследована проблема минимизации вычислительных затрат алгоритма. Рассмотрена реализация параллельного алгоритма в инфраструктуре GRID.

1. Coagulation equation

We consider a Cauchy problem for a system of spatially inhomogeneous nonlinear equations of coagulation (the system is also referred to as **coagulation equation**) [5]:

$$\begin{aligned}
&\frac{\partial c_1}{\partial t} + \operatorname{div}(\overline{v}c_1) = -c_1 \sum_{j=1}^{\infty} K(1,j)c_j, \\
&\frac{\partial c_i}{\partial t} + \operatorname{div}(\overline{v}c_i) = \frac{1}{2}\sum_{j=1}^{i-1} K(i-j,j)c_{i-j}c_j - c_i \sum_{j=1}^{\infty} K(i,j)c_j, \quad i \geq 2, \\
&c_i(0,\overline{x}) = c_i^0(\overline{x}).
\end{aligned} \tag{1}$$

Here $c_i = c_i(t,\overline{x}),\ i = 1,2,\dots,$ is concentration of i-mers at time t and point $\overline{x}$, $\overline{v} = \overline{v}(\overline{x})$ is a spatially inhomogeneous velocity field, $K(i,j)$ is a coagulation kernel, $c_i^0(\overline{x})$ is a concentration of i-mers at $t = 0$. We consider the equation inside time-spatial domain $\Omega \times (0,T]$, where $\Omega \subset R^3,\ T < \infty$.

We evaluate the following functionals of the equation solution:

$$\varphi_i(t) = \int\limits_G c_i(t,\overline{x})d\overline{x},\ G \subseteq \Omega. \tag{2}$$

Also we consider the same space integrals of spectrum moments.

*The work was supported by RFBR (grants N 06-01-00586 and N 06-01-00046), President (grant N 587.2008.1) of "Leading scientific schools" program, INTAS (grant N 05-109-5267), Russian Science Support Foundation's grant.

2. Monte Carlo algorithm (single processor case)

To solve coagulation equation (1) we simulate sample values of test particles ensemble

$$\xi = \xi(T) = \{p_1, p_2, \ldots, p_N\},$$

where $N = N(T)$. The pair $p_k = (l_k, \overline{x}_k)$ is called a **test particle**. Here $l_k \geq 1$ is a size of the particle (integer value), $\overline{x}_k \in R^3$ is a position of the particle. Denote by N_0 an initial number of test particles at $t = 0$.

Note that a velocity variable is not included to the phase state of the test particle (cf. with the opposite case in [2]). The reason is that the velocity field is defined for the particle system a-priori (see (1)).

Lets us split the spatial domain Ω into sufficiently small non-overlapping subdomains $\Omega_1, \Omega_2, \ldots, \Omega_S$ (they will be referred to as **interaction subdomains**). Denote by ρ_s a volume of s-th interaction subdomain. To define Monte Carlo algorithm let us consider so called "regularized" coagulation kernel [2, 3]:

$$K^{\rho}(p_1, p_2) = \sum_{s=1}^{S} \rho_s^{-1} h_s(\overline{x_1}) h_s(\overline{x_2}) K(l_1, l_2), \tag{3}$$

where $h_s(\overline{x})$ is an indicator function of the domain Ω_s.

Assume that there exists a majorant for the coagulation kernel:

$$K(l_1, l_2) \leq \widehat{K} < \infty.$$

Then the majorant for the "regularized" kernel (3) is defined as follows:

$$\widehat{K}^{\rho}(p_1, p_2) = \widehat{K} \rho_{\min}^{-1}, \tag{4}$$

where $\rho_{\min} = \min_k \rho_k$.

Let us split the interval $[0, T]$ into subintervals of length $\triangle t$. According to the majorant frequency algorithm [2, 3] a sample value $\xi = \xi(T)$ is simulated as follows:

1. Simulating the initial distribution of particles according to the probability density $f(i, \overline{x}) = c_i^0(\overline{x})$. Thus we have initial state of the particles ensemble $\xi(0) = \{p_1, p_2, \ldots, p_{N_0}\}$. Set $t = 0,\ t_c = 0$.

2. Simulating a random value τ — a time between subsequent coagulation events. The value τ is exponentially distributed with the parameter

$$\widehat{\nu} = \frac{1}{2N_0} \sum_{i \neq j} \widehat{K}^{\rho}(p_i, p_j). \tag{5}$$

Set $t_c = t_c + \tau$. If $t_c > \triangle t$ the algorithm switches to the stage 6.

3. Choosing a pair of coagulating particles (p_i, p_j) with the probability $(0.5N(N-1))^{-1}$.

4. Simulating a real or fictitious coagulation event for the chosen pair. A probability of fictitious event $P_f(p_i, p_j)$ is given as follows:

$$P_f(p_i, p_j) = 1,$$

provided both the particles belong to different interaction subdomains;

$$P_f(p_i, p_j) = 1 - \frac{K(l_i, l_j)\rho_{\min}}{\widehat{K}\rho_s},$$

provided both the particles belong to the same interaction subdomain Ω_s. When the real coagulation event occurs, the chosen pair of particles merges into one particle and the phase state of the ensemble changes as follows:

$$(p_i, p_j) = ((l_i, \overline{x}_i), (l_j, \overline{x}_j)) \to p_i' = (l_i + l_j, \overline{x}_i), \quad N = N - 1.$$

When the fictitious coagulation event occurs, the phase state of the ensemble doesn't change.

5. Switching to the stage 2.

6. Simulating the spatial transport of all particles according to Euler method with the step size $\triangle t$:

$$\overline{x}_i' = \overline{x}_i + \triangle t \overline{v}(\overline{x}_i).$$

Set $t_c = 0,\ t = t + \triangle t$.

7. If $t \le T$ the algorithm switches to the stage 2.

Then the functional φ can be estimated according to the formula

$$\varphi \approx \mathrm{E}_\xi \zeta \approx \frac{1}{L} \sum_{i=1}^{L} \zeta_i,$$

where $\zeta_i = \zeta(\xi_i)$, the notation E_ξ stands for the expectation with respect to distribution of ξ.

Denote by $\varepsilon_{\mathrm{det}}$ a deterministic error of the estimator ζ (i. e. the error of estimating φ with $\mathrm{E}_\xi \zeta$). Denote by $\varepsilon_{\mathrm{stat}}$ a statistical error of the estimator ζ (i. e. the error of estimating $\mathrm{E}_\xi \zeta$ with $1/L \sum\limits_{i=1}^{L} \zeta_i$). It is known that $\varepsilon_{\mathrm{stat}} = \gamma \sqrt{\mathrm{D}\zeta / L},\ \gamma = \mathrm{const}$.

3. Monte Carlo algorithm (multiprocessor case)

We consider a case when the number of particles N is so large that a simulation of a sample ξ must be carried out only on M processors of parallel computer:

$$\xi = \{\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(M)}\}.$$

To make parallel decomposition of the single processor algorithm, the computational domain $\Omega \subset R^3$ is splitted into M non-overlapping subdomains $\widehat{\Omega}_1, \widehat{\Omega}_2, \ldots, \widehat{\Omega}_M$, the particles being sorted out into subdomains. These subdomains will be referred to as **processors' subdomains**. Denote by n_m a number of particles in m-th subdomain. Each subdomain will be treated by a single processor.

Let us write the majorant frequency in the following way: $\widehat{\nu} = \sum\limits_{m=1}^{M} \widehat{\nu}_m$, where

$$\widehat{\nu}_m = \frac{1}{2N_0} \sum_{i \ne j} \widehat{K}^\rho(p_i, p_j), \tag{6}$$

the summation being taken over the particles belonging to $\widehat{\Omega}_m$ [2, 3]. Then a parallel simulation may be carried out as follows:

1. Each processor simulates initial condition independent of other processors (see stage 1):

$$\xi^{(m)}(0) = \{p_1^{(m)}, p_2^{(m)}, \ldots, p_{n_m}^{(m)}\},\ m = 1, 2, \ldots, M.$$

2. On m-th processor over the step size $\triangle t$ all coagulation events (real and fictitious) are simulated independent of other processors (see stages 2–5). In simulation instead of the parameter $\widehat{\nu}$ in (5) we use parameter $\widehat{\nu}_m$ from (6) and instead of N we use n_m. At the end of immediate interval $\triangle t$ we get

$$\xi^{(m)}(i\triangle t) = \{p_1^{(m)}, p_2^{(m)}, \ldots, p_{n_m}^{(m)}\},\ n_m = n_m(i\triangle t),\ m = 1, 2, \ldots, M,\ i = 1, 2, \ldots, T/\triangle t.$$

3. At the end of $\triangle t$ all processors exchange particles according to Euler method (see stage 6). Thus we get the updated ensembles $\xi^{(m)}(i\triangle t),\ m = 1, 2, \ldots, M$. Then the parallel algorithm switches to the stage 2.

Requirements for parallel random number streams being very strong, it is necessary to use well tested generator. It is recommended to use the generator introduced and tested in [4].

3.1. Optimal choice of parallel algorithm's parameters

While increasing the number of processors M it is reasonable to change other parameters of the algorithm in order to get better accuracy of computations. But it is necessary to change algorithm's parameters in an optimal way otherwise the computational cost may grow.

Note that

$$\xi = \xi(\overline{p}, N_0, \triangle t, \rho, M, \{\widehat{\Omega}_m\}_{m=1}^M),$$

where ρ is a typical volume of interaction subdomains, $\{\widehat{\Omega}_m\}_{m=1}^M$ is a set of processors' subdomains and $\overline{p}$ is a set of parameters corresponding to the coagulation equation (1) and the functional (2). Note that we neglect a dependence of ξ on interaction subdomains $\{\Omega_s\}_{s=1}^S$. It is possible to do it under some restrictions on the form of interaction subdomains.

An expectation of a computer time for Monte Carlo algorithm equals to $t_1 L$, where t_1 is an expectation of a computer time to simulate one sample. It follows from an equality of a deterministic error ε_{det} and a stochastic error $\varepsilon_{\text{stat}}$ that

$$L \sim \varepsilon_{\text{det}}^{-2} \mathrm{D}\zeta,$$

where $\mathrm{D}\zeta$ is a variance of the estimator. Therefore a **computational cost** of the algorithm is in direct proportion to the value

$$C(\zeta) = t_1 \varepsilon_{\text{det}}^{-2} \mathrm{D}\zeta.$$

Note that ε_{det} and $\varepsilon_{\text{stat}}$ don't depend on M and $\{\widehat{\Omega}_m\}_{m=1}^M$:

$$\varepsilon_{\text{det}} = \varepsilon_{\text{det}}(\overline{p}, N_0, \triangle t, \rho),\ \varepsilon_{\text{stat}} = \varepsilon_{\text{stat}}(\overline{p}, N_0, \triangle t, \rho, L).$$

Let us call the following function a **relative efficiency** of the parallel decomposition:

$$\Phi = \frac{C(\zeta)|_{M=1}}{C(\zeta)|_{M>1}} = \frac{t_1|_{M=1}}{t_1|_{M>1}}.$$

Here while simulating the values of $\overline{p}, N_0, \triangle t, \rho$ are the same for $M = 1$ and $M > 1$.

We assume a hypothesis that the variance of the estimator has the following dependence upon the parameters of the algorithm:

$$\mathrm{D}\zeta = D(\overline{p}, N_0, \triangle t, \rho) \sim N_0^{-1} D_1(\overline{p}),$$

the variance nearly not depending on ρ and $\triangle t$. It is possible to prove this hypothesis rigorously but this proof lies beyond the framework of this paper.

Also we assume that the deterministic error has the following order of magnitude [2, 3]:

$$\varepsilon_{\text{det}} = e(\overline{p}, N_0, \triangle t, \rho) \sim E_1(\overline{p})N_0^{-1} + E_2(\overline{p})\triangle t + E_3(\overline{p})\rho.$$

It follows from the last formula that

$$\triangle t \sim N_0^{-1},\ \rho \sim N_0^{-1}. \tag{7}$$

Therefore

$$\mathrm{D}\zeta \sim N_0^{-1},\ \varepsilon_{\text{det}} \sim N_0^{-1},\ L \sim \varepsilon_{\text{det}}^{-2}\mathrm{D}\zeta \sim N_0.$$

Let us investigate how the function t_1 depends on its parameters. It is evident that

$$t_1 = t_1(\overline{p}, N_0, \triangle t, \rho, M, \{\widehat{\Omega}_m\}_{m=1}^M).$$

It is clear that

$$t_1 = \mathrm{E}(t^{(i)} + t^{(c)} + t^{(e)}),$$

where $t^{(i)}$ corresponds to the simulation of the initial state of particles ensemble, $t^{(c)}$ corresponds to the independent sequential simulation of coagulation events on different processors, $t^{(e)}$ corresponds to the exchange of particles between processors. Let us derive orders of magnitude of the values $t^{(i)}, t^{(c)}, t^{(e)}$.

The parallel algorithm is synchronized at $i = 1, 2, \ldots, T/\triangle t$, namely, before data exchange stage and after having finished it. Therefore

$$t^{(c)} = \sum_{i=1}^{T/\triangle t} t_i^{(c)},\quad t_i^{(c)} = \max_{m=1,\ldots,M} t_{i,m}^{(c)},$$
$$t^{(e)} = \sum_{i=1}^{T/\triangle t} t_i^{(e)},\quad t_i^{(e)} = \max_{m=1,\ldots,M} t_{i,m}^{(e)},$$

where $t_{i,m}^{(c)}$ is a computer time corresponding to sequential simulation of coagulation events on m-th processor over i-th time interval $\triangle t$, $t_{i,m}^{(e)}$ is a computer time corresponding to the data exchange. It is clear that in an optimal case the following relationships for each sample must hold:

$$t_{i,1}^{(c)} \approx t_{i,2}^{(c)} \approx \ldots t_{i,M}^{(c)}, \tag{8}$$
$$t_{i,1}^{(e)} \approx t_{i,2}^{(e)} \approx \ldots t_{i,M}^{(e)} \tag{9}$$

at each step $i = 1, 2, \ldots, T/\triangle t$.

In what follows we describe some requirements for $\{\widehat{\Omega}_m\}_{m=1}^M$ to approximate relationships (8), (9). According to [2, 3] the value of $t_{i,m}^{(c)}$ has the following order of magnitude:

$$t_{i,m}^{(c)} \sim p_m \triangle t n_m, \tag{10}$$

where the constant p_m corresponds to the performance of the computer. Therefore if $\{\widehat{\Omega}_m\}_{m=1}^M$ are chosen such that

$$n_1(0) \approx n_2(0) \approx \ldots \approx n_M(0), \tag{11}$$
$$n_1(i\triangle t) \approx n_2(i\triangle t) \approx \ldots \approx n_M(i\triangle t),\quad i = 1, 2, \ldots, T/\triangle t$$

then the relationship (8) holds. Therefore a computational load of the parallel algorithm is quite well balanced provided $\mathrm{E}t^{(e)} \approx \mathrm{E}t^{(c)}$.

We consider a case when (11) holds automatically. It means that the coagulation equation has specific properties and the parameters of the parallel algorithm are being chosen in a specific way.

While simulating the initial condition, each processor makes the same amount of computational work. Namely, m-th processor uses the same random numbers as other processors use getting test particles in turn and choosing particles belonging to $\widehat{\Omega}_m$. Therefore

$$\mathrm{E}t^{(i)} = C_i N_0. \tag{12}$$

It follows from (11) that

$$\max_{m=1,\ldots,M} t_{i,m}^{(c)} \approx t_{i,m^*}^{(c)},\ i = 1, 2, \ldots, T/\triangle t$$

for some processor number m^*. A computational time it takes to simulate the fictitious coagulation events equals to $\dfrac{N_0}{M^2}\dfrac{C_f}{\rho}\triangle t$, where $C_f = \mathrm{const}$. For the real coagulation events a computational time equals to $\dfrac{N_0}{M} C_r \triangle t$, where $C_r = \mathrm{const}$. Therefore

$$t_{i,m^*}^{(c)} \approx \frac{N_0}{M}\Big(\frac{C_f}{M\rho} + C_r\Big)\triangle t, \quad \mathrm{E}t^{(c)} \sim \frac{N_0}{M}\Big(\frac{C_f}{M\rho} + C_r\Big). \tag{13}$$

A dependence of $t_{i,m}^{(e)}$ on the parameters $N_0, \triangle t, M, \{\widehat{\Omega}_m\}_{m=1}^M$ is obviously quite complicated. Also it is necessary to take into account a technology of processing the requests to send and receive data by a network software. Assume that the following relationship holds:

$$t_i^{(e)} \le C_e N_0 M^r \triangle t,\ r \ge 0,\ i = 1, 2, \ldots, T/\triangle t,$$

where $C_e = \mathrm{const}$. Considering the upper bound as the worst case of data exchange contribution we have

$$\mathrm{E}t^{(e)} \sim C_e N_0 M^r. \tag{14}$$

Let us specify the dependence of N_0 upon M in the following way:

$$N_0 = N_0' M^d,\ 0 \le d \le 1. \tag{15}$$

If we change the variables N_0, ρ for M taking into account (7) and (15) then for the case $M > 1$ we get the following relationship:

$$t_1|_{M>1} \sim C_i N_0 + \frac{N_0}{M}\Big(\frac{C_f}{M\rho} + C_r\Big) + C_e N_0 M^r \sim C_i M^d + C_f M^{2(d-1)} + C_r M^{d-1} + C_e M^{d+r}.$$

Similarly, for the case $M = 1$ we get the following relationship:

$$t_1|_{M=1} \sim C_i N_0 + N_0\Big(\frac{C_f}{\rho} + C_r\Big) \sim (C_i + C_r) M^d + C_f M^{2d}.$$

Therefore for the case $M > 1$ the computational cost has the following order of magnitude:

$$C(\zeta) \sim M^{2d+r}. \tag{16}$$

The relative efficiency of parallel decomposition has the following order of magnitude as $M \to \infty$:

- if $C_e = 0$ then $\Phi \sim M^d \to \infty$;
- if $C_e > 0,\ d < r$ then $\Phi \sim M^{d-r} \to 0$;
- if $C_e > 0,\ d = r$ then $\Phi \sim \text{const}$;
- if $C_e > 0,\ d > r$ then $\Phi \sim M^{d-r} \to \infty$.

In conclusion let us note that the foregoing results were obtained for the case when he numbers of particles $n_m,\ m = 1, 2, \ldots, M$ were almost equal during simulation according to (11). But we hope that in the case when the distribution of particle ensemble among processors is quite close to (11) one a behavior of relative efficiency and computational cost will be close to the above-mentioned relationships. Surely, the most complicated cases have to be investigated later on.

3.2. Implementation of parallel Monte Carlo algorithm on GRID infrastructure

In what follows we give some practical advices on the implementation of the DSMC algorithm on GRID infrastructure. We assume that computers forming GRID infrastructure have different performances. Also, underlying network is thought to be MPLS one, so we can order necessary network bandwidth. It is clear taht in the case of GRID implementation the computational cost of the data exchange is greater than the computational cost of sequential computations: $\mathrm{E}t^{(c)} \ll \mathrm{E}t^{(e)}$.

Here a main question is choosing the necessary network bandwidth. It is evident that in practice the number of processors M can not be increased infinitely. So it is not clear when the relative efficiency starts showing the asymptotic behavior (as described hereinbefore). Therefore a question arises which is a minimal network bandwidth under which

$$t_1|_{M=1} = t_1|_{M=M'}, \tag{17}$$

where M' is a number of available processors.

To determine minimal network bandwidth we make preliminary computations with $M = M'$ and necessary values of $\rho, \triangle t$ and $\{\widehat{\Omega}_m\}_{m=1}^{M}$. Actually the values of L and N_0 may be quite small. At the end we have the estimates for the values $t^{(i)}, t^{(c)}, t^{(e)}$. Actually instead of $t^{(e)}$ it is reasonable to evaluate $b^{(e)}$ — an amount of data transferred between processors (in bytes). If necessary we can scale the values of $t^{(i)}, t^{(c)}, b^{(e)}$ to the necessary value of N_0. It easy to do it because the values of $t^{(i)}, t^{(c)}, b^{(e)}$ are in direct proportion to N_0. Having all this information we can simulate a behavior of the network using special network simulator [1]. Such simulation enables to evaluate minimal network bandwidth to satisfy the condition (17).

The author thanks S.V. Rogazinskii for a helpful discussion of the paper.

References

[1] Adami D., Giordano S., Repeti M. et al. Design and implementation of a grid network-aware resource broker // Proc. PDCN: Parallel and Distributed Computing and Networks. IASTED: ACTA Press, Feb. 2006.

[2] Ivanov M.S., Rogazinskii S.V. Efiicient schemes of direct statistical simulation of rarified gas flows // Math. Modeling, 1989. Vol. 1, N 7. P. 130–145. (in Russian).

[3] Marchenko M.A. Majorant frequency principle for an approximate solution of a nonlinear spatially inhomogeneous coagulation equation by the Monte Carlo method // Russ. J. Numer. Anal. Math. Modelling. 2006. Vol. 21, N 3. P. 199–218.

[4] Marchenko M.A., Mikhailov G.A. Parallel realization of statistical simulation and random number generators // Russ. J. Numer. Anal. Math. Modelling. 2002. Vol. 17, N 1. P. 113–124.

[5] Voloshuk V.M., Sedunov Yu. S. Processes of coagulation in disperse systems. L.: Gidrometeoizdat, 1975.

Received for publication 20 February 2008