DOI: 10.14529/jsfi180204

Parallel Numerical Algorithm for Solving Advection Equation for Coagulating Particles

Sergey A. Matveev[1,3], *Rishat R. Zagidullin*[2], *Alexander P. Smirnov*[2,3], *Eugene E. Tyrtyshnikov*[2,3]

In this work we present a parallel implementation of numerical algorithm solving the Cauchy problem for equation of advection of coagulating particles. This equation describes time-evolution of the concentration $f(x, v, t)$ of particles of size v at the point x at the time-moment t. Our numerical algorithm is based on use of total variation diminishing (TVD) scheme and perfectly matching layers (PML) for approximation of advection operator along spatial coordinate x and utilization of the fast numerical method for evaluation of coagulation integrals exploiting low-rank decomposition of coagulation kernel coefficients and fast FFT-based implementation of convolution operation along particle size coordinate v. In our work we exploit one-dimensional domain decomposition approach along spatial coordinate x because it allows to avoid use of parallel FFT implementations which are very expensive in terms of data exchanges and have poor parallel scalability. Moreover, locality of finite-difference operator from TVD-scheme along x coordinate allows to obtain good scalability even for computing clusters with slow network interconnect due to modest volumes of data necessary for synchronization exchanges between times integration steps.

Keywords: aggregation equations, parallel algorithms, low-rank matrices, convolution.

Introduction

Coagulation and fragmentation processes stand in the basement of a wide class of physical phenomena starting from micro-polymer chains growth [6, 17] and finishing at the scale of stars formation from interstellar dust [5, 8]. The very first model of aggregation was suggested by Smoluchowski in 1916 [24] and generalized by Hans Muller [18] into the form of the following integro-differential equation:

$$\frac{\partial f(t,v)}{\partial t} = \frac{1}{2}\int\limits_0^v K(u, v-u) f(t,u) f(t, v-u) du - f(t,v) \int\limits_0^\infty K(v,u) f(t,u) du \qquad (1)$$

This equation describes dynamics of concentration $f(t, v)$ of the particles of size v per unit volume. The first term in the right-hand side corresponds to growth of concentration due to coalescence of the aggregates of sizes v and $v - u$. The second term describes decrease of the concentration due to their coalescence of the particles of size v with other particles. Kernel coefficients

$$K(v, u) = K(u, v) \geq 0$$

correspond to rates of aggregation process and have to be derived for each concrete application. If the initial conditions $f(t = 0, v)$ are known, then Cauchy problem for the coagulation equation is defined and can be solved numerically.

Under natural assumptions [8] the solution of the Cauchy problem for semi-infinite size domain $v \in [0, \infty)$ can be approximated by its finite part $v \in [0, V_{\max}]$. In fact, the class

[1] Skolkovo Institute of Science and Technology, Moscow, Russian Federation

[2] Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Moscow, Russian Federation

[3] Marchuk Institute of Numerical Mathematics RAS, Moscow, Russian Federation

of mathematical models of potential interest is extremely wide and may include description of various physical effects: inception and sinks of particles of concrete sizes [2], unary fragmentation process [1] due to instabilities of big clusters and binary collisional fragmentation process [5], and many others [3, 7, 12, 14, 20]. Even though the list of phenomena for application of spatially homogeneous aggregation and fragmentation equations is really huge [25, 26], there is an even broader class of spatially inhomogeneous models [19, 22]. In this work we concentrate on analysis of advection-coagulation equation [28]:

$$\begin{aligned} \frac{\partial f(t,x,v)}{\partial t} + c(v)\frac{\partial f(t,x,v)}{\partial x} = \frac{1}{2}\int\limits_0^v K(u,v-u)f(t,x,u)f(t,x,v-u)du - \\ -f(t,x,v)\int\limits_0^\infty K(u,v)f(t,x,u)du, \end{aligned} \tag{2}$$

where function $f(t, x, v)$ corresponds to concentration per unit volume of particles of size v at the point with coordinate x at moment t. Coagulating particles are transported along the axis of nonnegative coordinate x with velocity $c(v)$. The velocities of the coalescence processes are again determined by the values of function $K(u, v)$. In the similar manner with classical coagulation equation one needs initial conditions $f(t = 0, x, v)$ and one boundary condition e.g. $f(t, x = 0, v)$ for definition of the Cauchy problem for advection-coagulation equation and its further numerical investigation.

Recently, we proposed an efficient numerical method solving the Cauchy problem for this equation [28] and demonstrated its efficiency for the concrete examples of modelling problems. We revisit description of this algorithm in Section 1.

Nevertheless, even modest simulations presented in our previous report required quite a lot of CPU-time. In this work we propose a parallel implementation of this algorithm. We should also emphasize, that there exists a special parallel algorithm allowing to perform evaluation of aggregation and fragmentation sums of the similar structure as coagulation integrals. Even though, a reasonable speedup of calculations was presented at [16], the parallel scalability of the algorithm is relatively poor [16]. This fact lies in motivation to exploit one-dimensional domain decomposition approach along spatial coordinate x for advection-coagulation equation but not along axis v and not two-dimensional domain decomposition.

In Section 2 we present a detailed description of novel parallel algorithm. Section 3 is devoted to tests of performance of the proposed algorithm at different computing clusters and *Lomonosov* supercomputer. We obtain speedup of calculations more than by 300 times. This allows us consider a broader class of problems of potential interest for mathematical modelling and studies of advection-coagulation processes. In conclusions we discuss the presented results and possible directions for further generalization of the presented ideas.

1. Numerical Method, Revisited

In this section we revisit description of the fast numerical algorithm for advection-coagulation equation from work [28]. First of all, we use an explicit Euler time-integration scheme with time step Δt for solution of the Cauchy problem:

$$\frac{f^{n+1}-f^n}{\Delta t} = A(f^n) + S(f^n),$$

where $A(f)$ is an approximation of the advection operator and $S(f)$ approximates Smoluchowski integrals. Let us denote M as number of grid nodes along particle size axis v with grid step Δv and parameters $v_1 = 0$, $v_M = M \cdot \Delta v$ and N as number of spatial grid nodes along x with step Δx, $x_1 = 0$, $x_N = N \cdot \Delta x$. Finally, we denote $f^n_{i,j}$ as value of numerical solution at grid node (x_j, v_i) at the moment $n \cdot \Delta t$.

For $S(f)$ we use skeleton decomposition of coagulation kernel and fast algorithms of linear algebra with overall algorithmic complexity of solution scheme to be $O(MR \log M)$ at each grid node along x axis [13]. Whereas without utilization of these ideas the complexity becomes $O(M^2)$. We discuss this method with more details in Section 2.2. We get significant acceleration for evaluation of Smoluchowski integrals if the rank R of the coagulation kernel is a modest number. For approximation of the advection part $A(f)$ we exploit well-known TVD scheme [10] and PML layers [4] allowing us to keep monotonicity of numerical method. Detailed relations for advection part are presented in Section 2.3.

1.1. Handling Smoluchowski Integrals

Skeleton decomposition of coagulation kernel

$$K(u,v) = \sum_{\alpha=1}^{R} a_\alpha(u) b_\alpha(v)$$

helps accelerate computations by reduction of the algorithmic complexity of the numerical evaluation of Smoluchowski integrals [27, 29].

With use of skeleton decomposition we can perform transformation of the first integral:

$$\int_0^v K(u, v-u) f(t,x,u) f(t,x,v-u) du = \int_0^v \sum_{\alpha=1}^{R} a_\alpha(u) b_\alpha(v-u) f(t,x,u) f(t,x,v-u) du =$$
$$= \sum_{\alpha=1}^{R} \int_0^v a_\alpha(u) f(t,x,u) b_\alpha(v-u) f(t,x,v-u) du.$$

Following this idea we also transform the second integral:

$$\int_0^\infty K(u,v) f(t,x,u) du \approx \int_0^{V_{\max}} \sum_{\alpha=1}^{R} a_\alpha(u) b_\alpha(v) f(t,x,u) du = \sum_{\alpha=1}^{R} b_\alpha(v) \int_0^v a_\alpha(u) f(t,x,u) du.$$

Transformation of integrals written as sum of R lower-triangular convolutions, hence, with the use of FFT algorithm, total algorithmic complexity of the numerical integration using the relations above yields to $O(MR \log M)$ operations at each from N points of spatial grid. Hence, total cost of evaluation of Smoluchowski integrals is $O(NMR \log M)$

1.2. Approximation of Advection Part

For approximation of the advection equation we use the following TVD-scheme.

$$\frac{f_j^{n+1} - f_j^n}{dt} = A(f^n) = -\frac{c}{2dx}(C_j f_{j+1}^n + (2 - C_j - C_{j-1})f_j^n - (2 - C_{j-1})f_{j-1}^n) + \\ + \frac{c^2 dt}{2dx^2}(C_j f_{j+1}^n - (C_j + C_{j-1})f_j^n + C_{j-1} f_{j-1}^n),$$

where C_j is a flux limiter function that ensures that no artificial oscillations occur and has to satisfy the following restrictions

$$\frac{f_j^n - f_{j-1}^n}{f_{j+1}^n - f_j^n} \geq 0 \rightarrow 0 \leq C_j \leq 2$$

$$\frac{f_j^n - f_{j-1}^n}{f_{j+1}^n - f_j^n} < 0 \rightarrow C_j = 0.$$

For simplicity of notations we revisit this scheme ignoring volume index i because at each grid point along size coordinate v_i we perform these operations along spatial axis x. In fact, there are a lot of possible choices for C_j which were introduced by many authors [10, 11]. In our work, we use the one that is called “Monotonized Central Flux Limiter” [10].

$$C_j = \max(0, \min(2r_j, 0.5(1 + r_j), 2)), \text{where } r_j = \frac{f_j - f_{j-1}}{f_{j+1} - f_j}$$

The last thing that has to be done with respect to numerical solution of advection equation is incorporation of PML to emulate the absence of the right boundary condition in the grid. Hence, we modify the advection equation into following:

$$\frac{\partial f}{\partial t} = -c\frac{\partial f}{\partial x} - c\sigma(x)f,$$

where

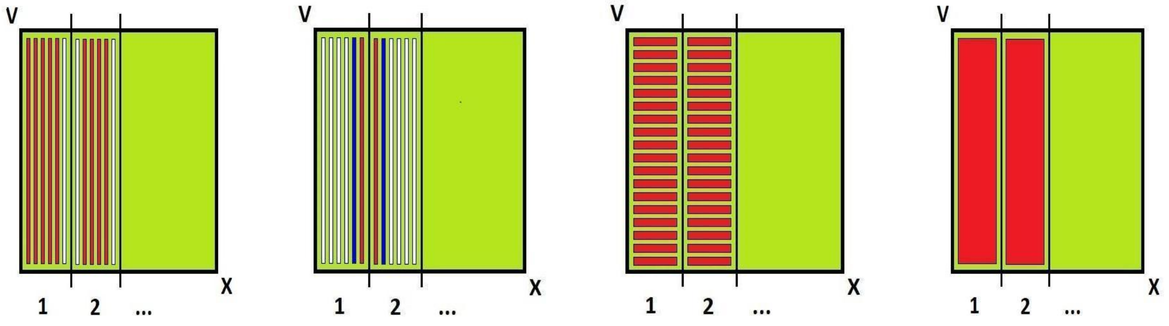
$$\sigma(x) = \begin{cases} \dfrac{(m+1)W \ln 10}{d}\left(\dfrac{x-l}{d}\right)^m & x \in [l; l+d], \\ 0 & \text{elsewhere,} \end{cases}$$

where l is width of calculation domain along axis x without PML, d is width of PML and W, m are PML parameters and formal assumption $f|_{x=l+d} \equiv 0$.

2. Parallel Algorithm

There are two possible ways of parallel implementation of our numerical method with use of p processors. First of all, we can perform simultaneous calculation by allocation of fixed particle size intervals to different processors. This is one-dimensional domain decomposition along axis v. But this option turns out to be bad due to quite poor parallel scalability of the algorithm for evaluation of the Smoluchoswki coagulation integrals what was reported in [16].

The reason lies in the fact that fast calculation of coagulation convolution typed integrals via FFT has a very limited resource of speedup on real parallel systems. Here we refer to work [9] where authors demonstrated that main problem in parallel execution of one-dimensional FFT is



(a) Computation of $S(f^n)$ (b) Data exchange (c) Computation of $A(f^n)$ (d) Calculation of f^{n+1}

Figure 1. One parallel time-integration step with use of the suggested parallel algorithm. Areas bordered by black lines represent the grid areas allocated to separated processors

problematic due to extremely high communication costs (comparable with computational). Such bad parallel scalability of evaluation of the coagulation integrals also leads to loss of motivation to exploit two-dimensional domain decomposition along both v and x coordinates in application to our problem.

However another option of execution of parallel calculations is to distribute them along the spatial coordinate x. Such distribution of data among processors corresponds exactly to uniform one-dimensional domain decomposition into the segments along spatial coordinate x. This approach allows to obtain very modest requirements in terms of number of communications – it is necessary just to synchronize the boundaries of spatial segments corresponding to neighboring processes for evaluation of advection operator with use of TVD-scheme stencil. In other words, we can allocate integrals at different points of the physical space to different processors and then transmit the data at the borders of the corresponding intervals. Thus, on the input each from p processes has set of grid nodes X and values of $f^n(x_i, v_j)$ at this domain $X \times (v_1, \ldots, v_M)$. Total complexity of time-integration step for each processor is $O\left(\frac{NRM \log M}{p}\right)$ operations and requires $O(pM)$ data elements for $O(p)$ data exchange operations.

We present our approach informally in Fig. 1. It shows the structure of the parallel numerical scheme and the areas of the grid which communicate with each other. Areas bordered by black lines represent the grid areas allocated to separated processors. Red stripes indicate how calculations are performed. Blue stripes pose organization of data transmission between the neighboring processes.

During stage corresponding to Fig. 1a each process evaluates the coagulation integrals along vertical axis v (lines 1-3 the pseudocode) and keeps two additional vectors for the boundaries assignment (presented as white stripes). Stage from Fig. 1b corresponds to synchronization of the particle size distributions along axis v (from blue stripes into red of the neighboring process) for the edge coordinates x between the processes (lines 4-5 of the pseudocode). It is necessary for the correct work of the TVD-scheme stencil. During stage from Fig. 1c the advection operator is evaluated along axis x for each particle size v (horizontal stripes corresponding to lines 6-10). All in all, during stage from Fig. 1d we provide calculation of the time-step result for both coordinates v and x (lines 11-16 of the pseudocode).

However, we should mention that grid parameter affects the parallel the efficiency of presented algorithm. Number of grid points along spatial coordinate x must be greater or equal to number of used processes. Otherwise there will be a lack of opportunity to distribute the data among the processors. We assume that the best parallel performance can be achieved in case of uniform domain decomposition when number of grid points along x can be divisible by a number of used processes. In the next section we present benchmarks of the proposed algorithm.

Algorithm 1 Operations performed by each process

```
for $i \in X$ do
    Compute $S(f^n(x_i; v_1 \dots, v_M))$ at $x_i$ along $v$    ▷ Requires $O\left(\frac{NRM \log M}{p}\right)$ operations
end for
Exchange data with left and right neighbors
and assign boundary conditions    ▷ Each process sends and receives $2M$ elements
for $i \in X$ do    ▷ loop among grid points corresponding to process
    for $j \in V$ do
        Apply advection operator $A(f^n(x_i; v_j))$    ▷ Requires $O\left(\frac{NM}{p}\right)$ operations
    end for
end for
for $i \in X$ do    ▷ loop among grid points corresponding to process
    for $j \in V$ do
        $f^{n+1}(x_i; v_j) = f^n(x_i; v_j) + \Delta t \cdot (\ S(f^n(x_i; v_j)) + A(f^n(x_i; v_j))\ )$    ▷ Requires $O\left(\frac{NM}{p}\right)$ operations
    end for
end for
return $f^{n+1}(x_i; v_j)$    ▷ defined for $X \times (v_1, \dots, v_M)$
```

3. Benchmarking Parallel Implementation

In our numerical tests we investigated the Cauchy problem for the advection-coagulation equation with zero-value initial conditions and the following left boundary condition:

$$f(t, x = 0, v) = e^{-v^2}. \tag{3}$$

Thus, we model a constant flow of particles from one side of the grid to the other.

Also, calculations differ with respect to coagulation kernels. Tests were conducted for constant kernel

$$K(u, v) \equiv 1,$$

which rank $R = 1$ and for ballistic coagulation kernel

$$K(u, v) = (u^{\frac{1}{3}} + v^{\frac{1}{3}})^2 \sqrt{\frac{1}{u} + \frac{1}{v}}$$

with greater values of ranks. Grid parameters were set as $\Delta x = \Delta v = 10^{-2}$ and $\Delta t = 0.005$.

Scalability of parallel implementation was tested on several different computing clusters, so we assume that the results are robust. Moreover, our calculations also differ with respect to mesh and coagulation kernel. Surprisingly, even for a system with slow interconnect (see Tab. 4

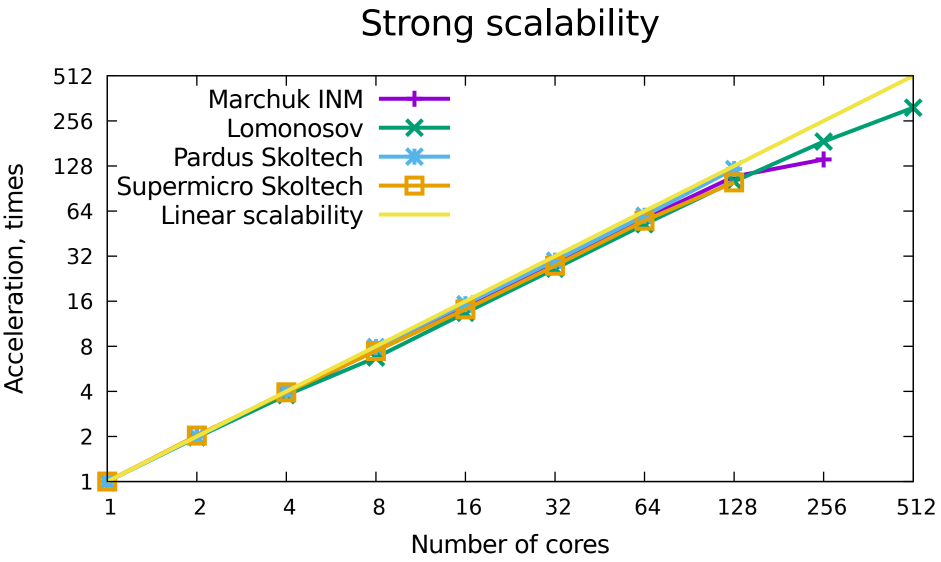


Figure 2. Strong scalability of the proposed parallel algorithm for different clusters. Visualized data corresponds to results presented in Tab. 1 – 4. We obtain almost linear speedup if number of processors is less than or equal to 128

Table 1. Supercomputer *Lomonosov*, tests for *regular4* partition with *40 Gb/s QDR* interconnect. Constant coagulation kernel, $R = 1$, $N = M = 8192$, 32 time-integration steps

p	Time, s	Speedup
1	34.476	1
2	17.468	1.97
4	9.126	3.778
8	5.124	6.728
16	2.576	13.384
32	1.306	26.398
64	0.66	52.236
128	0.342	100.807
256	0.185	186.357
512	0.11	313.418

with results at *Supermicro* cluster of Skoltech *1 Gb/s Ethernet*) and for heterogeneous cluster of Marchuk Institute of Numerical Mathematics (see Tab. 2) we obtain a reasonable speedup of calculations. Besides the fact that we obtain a good scalability of the algorithm for different clusters, we also see its good performance for different problem sizes (for each of Tab. 1 – 4 we tried a different problem size).

Tables 1 – 4 and Fig. 2 indicate that the scalability of parallel program is almost linear at the amount of nodes per dimension lower than or equal to 128. If the number of processors goes beyond that number, the payoffs from parallel implementation slowly decrease due to

Table 2. Cluster of Marchuk Institute of Numerical Mathematics RAS; test for heterogeneous cluster consisting of many different models of CPUs with *32 Gb/s QDR* interconnect. Ballistic coagulation kernel, $R = 12$, $N = M = 2048$, 32 time-integration steps

p	Time, s	Speedup
1	133.43	1
2	66.112	2.018
4	34.49	3.869
8	17.775	7.507
16	8.997	14.83
32	4.618	28.893
64	2.353	56.706
128	1.227	108.745
256	0.943	141.495

Table 3. *Pardus* cluster of Skoltech with *56 Gb/s FDR* interconnect. Ballistic coagulation kernel, $R = 14$, $N = M = 4992$, 32 time-integration steps

p	Time, s	Speedup
1	1202.947	1
2	604.34	1.99
4	301.304	3.992
8	152.223	7.903
16	78.681	15.289
32	40.3	29.85
64	20.175	59.626
128	9.848	122.151

Table 4. *Supermicro* cluster of Skoltech with *1 Gb/s Ethernet* interconnect. Ballistic coagulation kernel, $R = 14$, $N = M = 8064$, 32 time-integration steps

p	Time, s	Speedup
1	2373.189	1
2	1168.678	2.03
4	601.246	3.947
8	318.474	7.452
16	168.127	14.115
32	85.922	27.62
64	42.952	55.252
128	23.838	99.555

increasing costs of data exchanges between processors after each time integration step. Speedup also saturates with satisfactory results when the size of problems is not large enough (see Tab. 2).

Conclusions

In this work we propose a parallel implementation of the numerical method solving advection-coagulation equation from work [28]. The numerical scheme is parallelized along spatial axis x which promotes a good speedup of calculations. In the presented benchmarks we show that our scheme of finding the numerical solution of the problem might be accelerated by hundreds of times giving an almost linear speedup with respect to the amount of used CPU cores.

There is also an opportunity for additional increase of performance of the presented method. As far as coalescence integrals are evaluated sequentially at each processor their calculation can be additionally accelerated by use of GPU and Cuda technology. We will apply our ideas to more intricate multicomponent coagulation models [7, 15, 23] requiring special implementations of multidimensional convolutions based on utilization of low-rank tensor decompositions [15, 21].

Acknowledgements

This work is supported by the Program 1 of the Presidium of the Russian Academy of Sciences “Fundamental mathematics and its applications” and by Program 26 “Fundamental foundations of design of algorithms and software for prospective and high-performance computing systems”. We also would like to acknowledge Center for Computational and Data-Intensive Science and Engineering of Skolkovo Institute of Science and Technology for comprehensive support of their Pardus cluster which we used during preparation of our benchmarks.

References

1. Aloyan, A.: Dynamics and kinematics of gas impurities and aerosols in the atmosphere. A Textbook (2002)

2. Ball, R., Connaughton, C., Jones, P., Rajesh, R., Zaboronski, O.: Collective oscillations in irreversible coagulation driven by monomer inputs and large-cluster outputs. Physical review letters 109(16), 168304 (2012), DOI: 10.1103/PhysRevLett.109.168304

3. Ball, R., Connaughton, C., Stein, T.H., Zaboronski, O.: Instantaneous gelation in Smoluchowskis coagulation equation revisited. Physical Review E 84(1), 011111 (2011), DOI: 10.1103/PhysRevE.84.011111

4. Berenger, J.P.: A perfectly matched layer for the absorption of electromagnetic waves. Journal of Computational Physics 114(2), 185–200 (1994), DOI: 10.1006/jcph.1994.1159

5. Brilliantov, N., Krapivsky, P., Bodrova, A., Spahn, F., Hayakawa, H., Stadnichuk, V., Schmidt, J.: Size distribution of particles in saturns rings from aggregation and fragmentation. Proceedings of the National Academy of Sciences 112(31), 9536–9541 (2015), DOI: 10.1073/pnas.1503957112

6. Brilliantov, N., Bodrova, A., Krapivsky, P.: A model of ballistic aggregation and fragmentation. Journal of Statistical Mechanics: Theory and Experiment 2009(06), P06011 (2009), DOI: 10.1088/1742-5468/2009/06/P06011

7. Chaudhury, A., Oseledets, I., Ramachandran, R.: A computationally efficient technique for the solution of multi-dimensional PBMs of granulation via tensor decomposition. Computers & Chemical Engineering 61, 234–244 (2014), DOI: 10.1016/j.compchemeng.2013.10.020

8. Galkin, V.: Smoluchowski equation. Fizmatlit, Moscow (2001), (in Russian)

9. Gupta, A., Kumar, V.: The scalability of fft on parallel computers. Parallel and Distributed Systems, IEEE Transactions on 4(8), 922–932 (1993), DOI: 10.1109/71.238626

10. Leer, B.V.: Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection. Journal of Computational Physics 23(3), 276–299 (1977), DOI: 10.1016/0021-9991(77)90095-X

11. Lyra, P.R.M., Morgan, K., Peraire, J., Peir, J.: TVD algorithms for the solution of the compressible euler equations on unstructured meshes. International Journal for Numerical Methods in Fluids 19(9), 827–847, DOI: 10.1002/fld.1650190906

12. Matveev, S.A., Krapivsky, P.L., Smirnov, A.P., Tyrtyshnikov, E.E., Brilliantov, N.V.: Oscillations in aggregation-shattering processes. Phys. Rev. Lett. 119, 260601 (Dec 2017), DOI: 10.1103/PhysRevLett.119.260601

13. Matveev, S., Smirnov, A., Tyrtyshnikov, E.: A fast numerical method for the cauchy problem for the Smoluchowski equation. Journal of Computational Physics 282, 23–32 (2015), DOI: 10.1016/j.jcp.2014.11.003

14. Matveev, S., Stadnichuk, V., Tyrtyshnikov, E., Smirnov, A., Ampilogova, N., Brilliantov, N.: Anderson acceleration method of finding steady-state particle size distribution for a wide class of aggregationfragmentation models. Computer Physics Communications 224, 154–163 (2018), DOI: 10.1016/j.cpc.2017.11.002

15. Matveev, S., Zheltkov, D., Tyrtyshnikov, E., Smirnov, A.: Tensor train versus Monte Carlo for the multicomponent Smoluchowski coagulation equation. Journal of Computational Physics 316, 164–179 (2016), DOI: 10.1016/j.jcp.2016.04.025

16. Matveev, S.A.: A parallel implementation of a fast method for solving the smoluchowski-type kinetic equations of aggregation and fragmentation processes. Vychislitel'nye Metody i Programmirovanie 16(3), 360–368 (2015), (in Russian)

17. Mirzaev, I., Byrne, E.C., Bortz, D.M.: An inverse problem for a class of conditional probability measure-dependent evolution equations. Inverse Problems 32(9), 095005 (2016), DOI: 10.1088/0266-5611/32/9/095005

18. Müller, H.: Zur allgemeinen theorie ser raschen koagulation. Fortschrittsberichte über Kolloide und Polymere 27(6), 223–250 (1928), DOI: 10.1007/BF02558510

19. Okuzumi, S., Tanaka, H., Kobayashi, H., Wada, K.: Rapid coagulation of porous dust aggregates outside the snow line: A pathway to successful icy planetesimal formation. The Astrophysical Journal 752(2), 106 (2012), DOI: 10.1088/0004-637X/752/2/106

20. Piskunov, V.: Analytical solutions for coagulation and condensation kinetics of composite particles. Physica D: Nonlinear Phenomena 249, 38–45 (2013), DOI: 10.1016/j.physd.2013.01.008

21. Rakhuba, M.V., Oseledets, I.V.: Fast multidimensional convolution in low-rank tensor formats via cross approximation. SIAM Journal on Scientific Computing 37(2), A565–A582 (2015), DOI: 10.1137/140958529

22. Sabelfeld, K.: A random walk on spheres based kinetic monte carlo method for simulation of the fluctuation-limited bimolecular reactions. Mathematics and Computers in Simulation (2016), DOI: 10.1016/j.matcom.2016.03.011

23. Smirnov, A., Matveev, S., Zheltkov, D., Tyrtyshnikov, E.: Fast and accurate finite-difference method solving multicomponent Smoluchowski coagulation equation with source and sink terms. Procedia Computer Science 80, 2141–2146 (2016), DOI: 10.1016/j.procs.2016.05.533

24. von Smoluchowski, M.: Drei vortrage uber diffusion, brownsche bewegung und koagulation von kolloidteilchen. Zeitschrift fur Physik 17, 557–585 (1916)

25. Sorokin, A., Strizhov, V., Demin, M., Smirnov, A.: Monte-Carlo modeling of aerosol kinetics. Atomic Energy 117(4), 289–293 (2015), DOI: 10.1007/s10512-015-9923-7

26. Stadnichuk, V., Bodrova, A., Brilliantov, N.: Smoluchowski aggregation–fragmentation equations: Fast numerical method to find steady-state solutions. International Journal of Modern Physics B 29(29), 1550208 (2015), DOI: 10.1142/S0217979215502082

27. Tyrtyshnikov, E.E.: Incomplete cross approximation in the mosaic–skeleton method. Computing 64(4), 367–380 (2000), DOI: 10.1007/s006070070031

28. Zagidullin, R.R., Smirnov, A.P., Matveev, S.A., Tyrtyshnikov, E.E.: An efficient numerical method for a mathematical model of a transport of coagulating particles. Moscow University Computational Mathematics and Cybernetics 41(4), 179–186 (Oct 2017), DOI: 10.3103/S0278641917040082

29. Zheltkov, D.A., Tyrtyshnikov, E.E.: A parallel implementation of the matrix cross approximation method. Vychislitel'nye Metody i Programmirovanie 16(3), 369–375 (2015), (in Russian)