Some thoughts about modelling aggregation

Matveev S. A.

July 24, 2019

General properties of considered systems

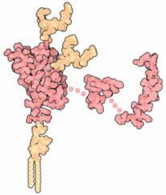
- Pavel Krapivsky, Sidney Redner, Eli Ben-Naim. A kinetic view of statistical physics. Cambridge University Press, 2010.
- Фукс, Н. А. "Механика аэрозолей." 1955.
- Leyvraz Francois. Scaling theory and exactly solved models in the kinetics of irreversible aggregation. Physics Reports, 2003.
- Галкин В. А. "Уравнение Смолуховского", ФизМатЛит, 2001

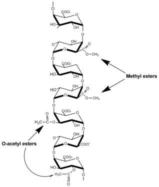
General properties of considered systems

We consider system of chaotically moving particles which

- Collide inelastically
- Fill space homogeneously and uniformly
- Collide by pairs, triplets etc
- Differ in sizes

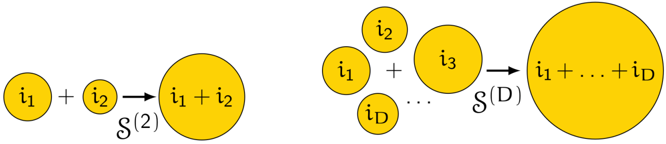
Aggregation and fragmentation processes can be described with Smoluchowski-type equations





- Aerosol dynamics
- Reversible polymerization
- Aggregation and fragmentation in planetary rings

General properties of considered systems



General structure of models

In compact notations for the discrete case:

$$\begin{cases} \dfrac{d\mathbf{n}}{dt} = \mathbf{S}(\mathbf{n(t)}), \\ \mathbf{n}(0) = \mathbf{n}_0 \end{cases} .$$

$$\mathbf{n}(t) = \begin{bmatrix} n_1(t) & n_2(t) & \dots & n_M(t) & \dots \end{bmatrix}$$

$$\mathbf{n}_0 = \begin{bmatrix} n_{1_0} & n_{2_0} & \dots & n_{M_0} & \dots \end{bmatrix},$$

and continuous

$$\begin{cases} \dfrac{\partial n(v,t)}{\partial t} = \mathbf{S}(n(t)), \\ n(v, t=0) = n_0(v) \end{cases} .$$

Multi-component case:

$$\overline{v} = (v_1, v_2, \dots, v_d)$$

Pairwise models

- Continuous Smoluchowski equation

$$\frac{\partial n(v,t)}{dt} = \frac{1}{2}\int_0^v C(u, v-u)n(u,t)n(v-u,t)du - \\ -n(v,t)\int_0^\infty C(v,u)n(u,t)du.$$

- Discrete Smoluchowski equation

$$\frac{dn_k(t)}{dt} = \frac{1}{2}\sum_{i+j=k} C_{i,j}n_i n_j - n_k \sum_{j\geq 1} C_{j,k}n_j \ , \ \ k = \overline{1,\infty}\,.$$

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

Pairwise models

- Irreversible coagulation with source and sink

$$\frac{dn_k(t)}{dt} = \frac{1}{2}\sum_{i+j=k} C_{i,j} n_i n_j - n_k \sum_{j=1}^{N} C_{j,k} n_j + J\delta_{k,1} \ , \ k = \overline{1, N} \, .$$

[Ball, Connaughton, Jones, Rajesh, Zaboronski, *Collective oscillations in irreversible coagulation driven by monomer inputs and large-cluster outputs*, PRL (2012)]

- Aggregation-fragmentation equations in planetary rings

$$\frac{dn_k(t)}{dt} = \frac{1}{2}\sum_{i+j=k} C_{i,j} n_i n_j - (1+\lambda) n_k \sum_{j\geq 1} C_{j,k} n_j \ , \ k = \overline{2, \infty} \, .$$

$$\frac{dn_1}{dt} = -n_1 \sum_{j\geq 1} C_{1,j} n_j + \frac{\lambda}{2} \sum_{i,j\geq 2} C_{i,j}(i+j) n_i n_j + \lambda n_1 \sum_{j\geq 2} j C_{1,j} n_j \, .$$

[Brilliantov, Krapivsky, Bodrova, ... *Size distribution of particles in Saturn's rings from aggregation and fragmentation*, PNAS (2015)]

Pairwise models

- Multicomponent Smoluchowski equation

$$\frac{\partial n(\overline{v}, t)}{\partial t} = \frac{1}{2}\int_0^{v_1} \dots \int_0^{v_d} C(\overline{v} - \overline{u}; \overline{u}) n(\overline{v} - \overline{u}, t) n(\overline{u}, t) du_1 \dots du_d -$$

$$- n(\overline{v}, t) \int_0^{\infty} \dots \int_0^{\infty} C(\overline{u}; \overline{v}) n(\overline{u}, t) du_1 \dots du_d .$$

Multi-particle kinetics

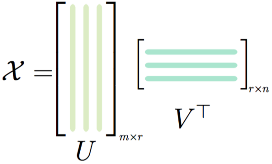
$$\frac{dn_k}{dt} = S_k^{(2)}(\mathbf{n}) + S_k^{(3)}(\mathbf{n}), \qquad k = \overline{1, \infty},$$

$$S_k^{(2)}(\mathbf{n}) = \frac{1}{2} \sum_{i_1+i_2=k} C_{i_1,i_2}^{(2)} \, n_{i_1} n_{i_2} - n_k \sum_{i_1=1}^{\infty} C_{i_1,k}^{(2)} \, n_{i_1},$$

$$S_k^{(3)}(\mathbf{n}) = \frac{1}{6} \sum_{i_1+i_2+i_3=k} C_{i_1,i_2,i_3}^{(3)} \, n_{i_1} n_{i_2} n_{i_3} - \frac{n_k}{2} \sum_{i_1,i_2=1}^{\infty} C_{i_1,i_2,k}^{(3)} \, n_{i_1} n_{i_2}.$$

Low-rank decompositions

Matrix skeleton decompositon



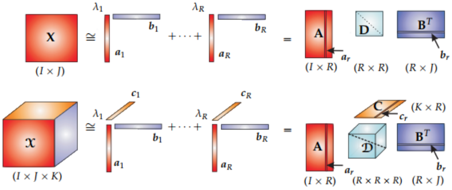
and rule for evaluation of the elements:

$$X(i_1, i_2) = \sum_{\alpha=1}^{R} U(i_1, \alpha) V(i_2, \alpha)$$

- We store just $2nR \ll n^2$. It is good when $R \ll n$
- There are robust methods

Low-rank decompositions

Canonical polyadic



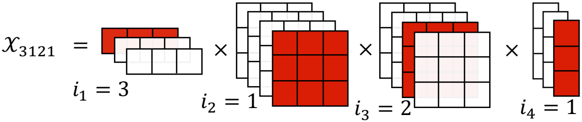
$$X(i_1, i_2, \dots, i_d) = \sum_{\alpha=1}^{R} U_1(i_1, \alpha) U_2(i_2, \alpha) \dots U_d(i_d, \alpha)$$

- Compression $O(N^d) \to O(dNR)$
- Evaluation of rank is – NP-full

Low-rank decompositions

Tensor train

$$X_{i_1,i_2,\dots,i_d} = \sum_{\alpha_i=1}^{R} G_1(\alpha_0, i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \dots G_d(\alpha_{d-1}, i_d, \alpha_d)$$



- Compression $O(N^d) \rightarrow O(dNR^2)$
- Complexity $O(dNR^3)$
- Robust methods
- Robust operations

Predictor-corrector time-integration

$$\frac{n^{t+\frac{1}{2}} - n^t}{0.5\tau} = S^{(2)}(n^t) + S^{(3)}(n^t) + \ldots + S^{(D)}(n^t)$$

$$\frac{n^{t+1} - n^t}{\tau} = S^{(2)}(n^{t+\frac{1}{2}}) + S^{(3)}(n^{t+\frac{1}{2}}) + \ldots + S^{(D)}(n^t),$$

Straight-forward complexity of evaluation of $S^{(D)}$ is $O(N^D)$. Even for two-component case it is too much!

- TT-based algorithm $O(NDR^2 \log N)$
- CP-based algorithm $O(NDR \log N)$

Main idea for pairwise case

Re-write **S** as:

$$\frac{1}{2}\sum_{i=2}^{N}\sum_{j=2}^{N} C_{i,j}(i+j)n_i n_j = \frac{1}{2}\sum_{i=2}^{N} i n_i \sum_{j=2}^{N} C_{i,j} n_j + \frac{1}{2}\sum_{i=2}^{N} n_i \sum_{j=2}^{N} C_{i,j} j n_j =$$

$$= \begin{bmatrix} n_2 & n_3 & \dots & n_N \end{bmatrix} \times \begin{bmatrix} C_{2,2} & C_{2,3} & \dots & C_{2,N} \\ C_{3,2} & \dots & \dots & C_{3,N} \\ \dots & \dots & \dots & \dots \\ C_{N,2} & C_{N,3} & \dots & C_{N,N} \end{bmatrix} \times \begin{bmatrix} 2n_2 \\ 3n_3 \\ \dots \\ Nn_N \end{bmatrix} \approx$$

$$\approx \begin{bmatrix} n_2 & n_3 & \dots & n_N \end{bmatrix} \times UV^T \times \begin{bmatrix} 2n_2 \\ 3n_3 \\ \dots \\ Nn_N\ . \end{bmatrix}$$

Main idea for pairwise case

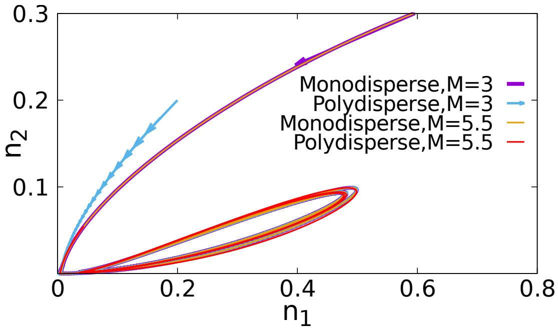
$$\sum_{i=1}^{k-1} C_{i,j} n_i n_{k-i} \approx \sum_{i=1}^{k-1} \sum_{\alpha=1}^{R} U_\alpha(i) V_\alpha(k-i) n_i n_{k-i} =$$

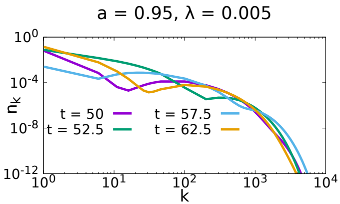
$$= \sum_{i=1}^{k-1} \sum_{\alpha=1}^{R} \widehat{U_\alpha}(i) \widehat{V_\alpha}(k-i)$$

$$\widehat{U_\alpha}(i) \equiv U_\alpha(i) n_i; \ \widehat{V_\alpha}(i) \equiv V_\alpha(i) n_i \,.$$

Hence complexity of time-step becomes $O(NR \log N)$ operations.
Then we need estimetes for R.

Oscillations in aggregation shattering processes





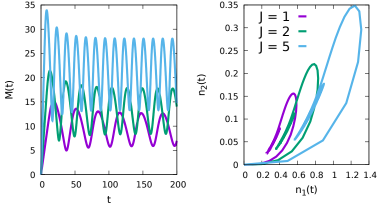
Oscillations in aggregation with source and sink

$$\frac{dn_k(t)}{dt} = \frac{1}{2}\sum_{i+j=k} C_{i,j} n_i n_j - n_k \sum_{j=1}^{N} C_{j,k} n_j + J\delta_{k,1} \ , \ \ k = \overline{1, N} \, .$$

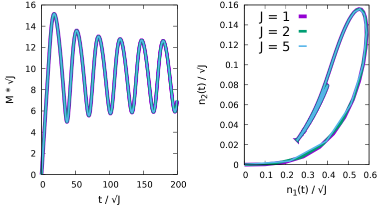
There is trivial scaling rule allowing to normalize sink:

$$n_i(t)|_J = n_i\left(t \cdot \sqrt{J}\right)|_{J=1} \cdot \sqrt{J}.$$

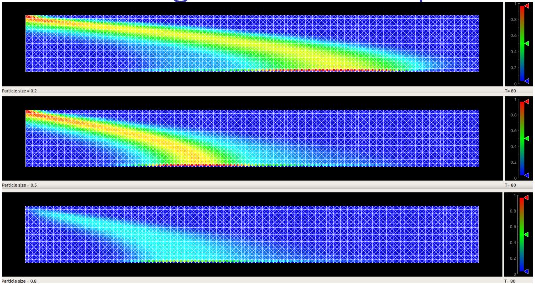
$C_{i,j} = i^{3/4}\, j^{3/4} + j^{3/4}\, i^{3/4}$



$C_{i,j} = i^{3/4}\, j^{3/4} + j^{3/4}\, i^{3/4}$, scaling



Advection-coagulation in 2D spatial case



$$\frac{\partial f(t,x,y,v)}{\partial t} + \nabla \cdot (c(x,y,v) f(t,x,y,v)) =$$

$$= \frac{1}{2} \int_0^v K(u, v-u) f(t,x,y,u) f(t,x,y,v-u) du -$$

$$- f(t,x,y,v) \int_0^\infty K(u,v) f(t,x,y,u) du,$$

Issues in progress for advection-coaulation

- Add realistic evaluation of velocities $c(x, y, v)$
- Introduce some theoretical anlysis for convergence and stability
- Incorporate use of ready-made CFD packages and platforms

We are very interested in discussions and advice!

Main idea for milti-component case

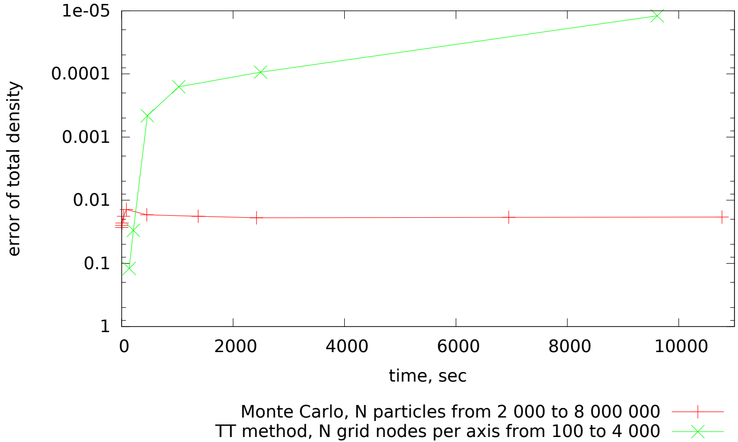
- Represent both kernel and soluton in TT-format and use TT-operations

$$\frac{\partial n(\overline{v},t)}{\partial t} = \frac{1}{2}\int_0^{v_1} \dots \int_0^{v_d} C(\overline{v}-\overline{u};\overline{u}) n(\overline{v}-\overline{u},t) n(\overline{u},t) du_1 \dots du_d -$$

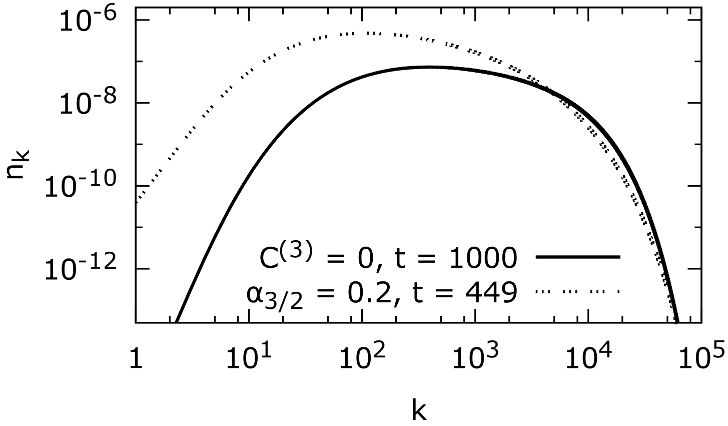
$$-n(\overline{v},t)\int_0^{\infty} \dots \int_0^{\infty} C(\overline{u};\overline{v}) n(\overline{u},t) du_1 \dots du_d.$$

- After some technical preparations we obtain complexity of time-step $O(dN\log NR^3 + d^2NR^6)$ instead of $O(N^{2d})$

Multicomponent case performance vs Monte Carlo



Numerical results for triple collisions



Speedup in parallel[1]

Number of CPU-cores	time, sec	Speedup
1	257.80	1.00
2	147.62	1.75
4	80.21	3.21
8	43.65	5.91
16	22.63	11.39
32	14.83	17.38
64	13.15	19.60
128	12.22	21.09

[1] "Zhores" supercomputer of Skoltech, 100 Gb/s EDR, 24 CPU cores per node

Спасибо за внимание!
Thank you for attention![2]

[2] Our codes are open, ask if you interested