Параллельные реализации разностных схем для решения уравнений агрегационной кинетики

Сергей Матвеев

9 октября 2019 г.

Skolkovo Institute of Science and Technology

Aggregation and fragmentation models

- Smoluchowski aggregation equations in discrete form [1914-1916]

$$\frac{dn_k(t)}{dt} = \frac{1}{2}\sum_{i+j=k} K_{i,j}n_in_j - n_k\sum_{j=1}^{\infty} K_{j,k}n_j$$

- Smoluchowski coagulation equation in continuous form [H. Muller, 1928]

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

$$n(t,v)\int_0^\infty K(u,v)f(t,x,u)du.$$

How to evaluate sums and integrals in modest time?

$$\sum_{j=1}^{N} K_{i,j} n_j = \begin{bmatrix} K_{2,2} & K_{2,3} & \dots & K_{2,N} \\ K_{3,2} & \dots & \dots & K_{3,N} \\ \dots & \dots & \dots & \dots \\ K_{N,2} & K_{N,3} & \dots & K_{N,N} \end{bmatrix} \times \begin{bmatrix} n_1 \\ n_2 \\ \dots \\ n_N \end{bmatrix} \approx$$

$$\approx UV^T \times \begin{bmatrix} n_1 \\ n_2 \\ \dots \\ n_N . \end{bmatrix}$$

How to evaluate sums and integrals in modest time?[1]

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

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

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

[1] *M., Smirnov, Tyrtyshnikov, A fast method for the Cauchy problem for Smoluchowski coagulation equation, JCP, 2015*

Aggregation and fragmentation models

- Advection-coagulation equation

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

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

$$f(t=0,x,v) = f_0(x,v)$$

$$f(t,x=0,v) = f_b(t,x).$$

Numerical method for advection-coagulation equation [2]

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

- Total Variation Diminishing (TVD) scheme for advection part
- Fast evaluation of Smoluchowski integrals for coagulation part
- Perfectly Matching Layers (PML) at the second boundary

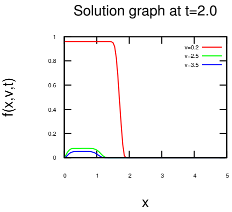
Total complexity is $O(NMR \log N)$ operations instead of $O(N^2 M)$

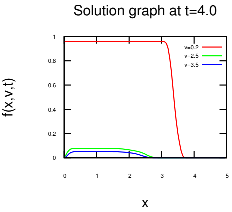
[2] *Zagidullin, Smirnov, M., Tyrtyshnikov, An Efficient Numerical Method for a Mathematical Model of a Transport of Coagulating Particles, Moscow State Univ. Bulletin, 2017*

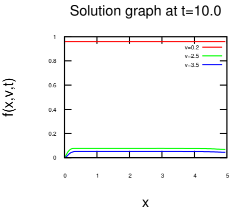
Numerical tests of the proposed approach. $\Delta h = \Delta v = 0.05$, $\Delta t = 0.01$, $v, x \in [0, 5]$, $t \in [0, 10]$

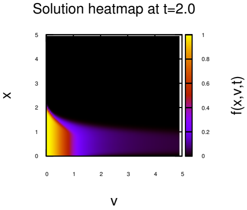
Kernel $K(u, v)$	Velocity $c(v)$	Grid parameters			
		$2\Delta h, 2\Delta t$	$\Delta h, \Delta t$	$\frac{\Delta h}{2}, \frac{\Delta t}{2}$	$\frac{\Delta h}{4}, \frac{\Delta t}{4}$
Constant	Constant	0.737	4.190	35.213	228.024
Constant	Decreasing	0.707	4.346	34.718	248.944
Ballistic	Constant	2.643	21.176	301.308	1779.738
Ballistic	Decreasing	2.522	21.332	270.761	2087.567

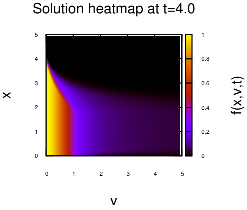
Example of the solution

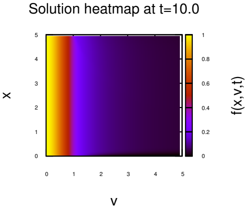










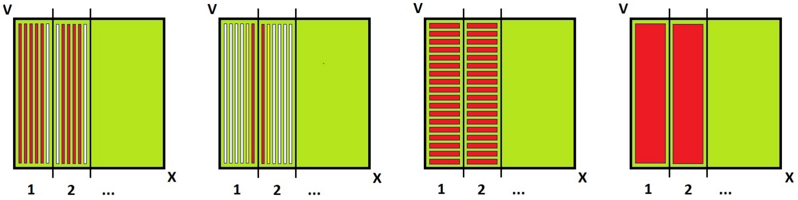


Parallel scalability of evaluation of Smoluchowski is poor[3]!
Let's avoid their parallelization

Number of cores	Time, sec.	Speedup
1	4600	1.0
2	2687	1.71
4	1578	3.54
8	1052	4.37
16	675	6.81
32	464	9.91
64	267	17.28
128	185	24.86
256	104	44.23

[3] *M. , A Parallel Implementation of a Fast Method for Solving the Smoluchowski-Type Kinetic Equations of Aggregation and Fragmentation Processes, 2015 (in Russian)*

Parallel algorithm[4]

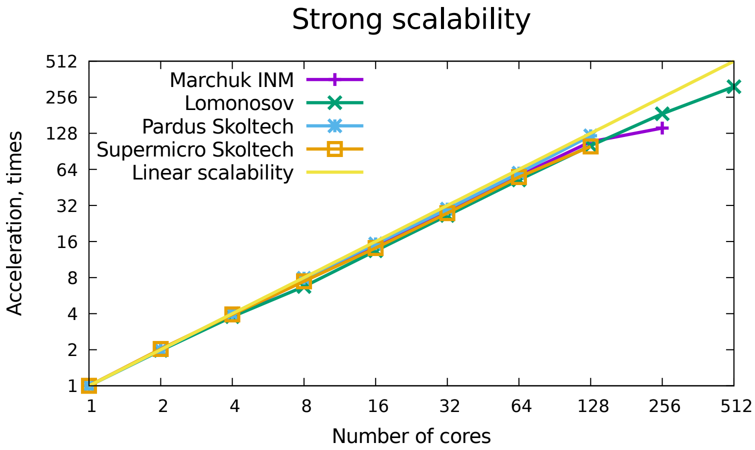


computation of $S(f^n)$ | data exchange | computation of $A(f^n)$ | calculation of f^{n+1}

One parallel time-integration step with use of the suggested parallel algorithm.

[4] *M. , Zagidullin, Smirnov, Tyrtyshnikov, A Parallel Numerical Algorithm Solving Equation of Advection of Coagulating Particles (submitted to SFRI), 2018*

Strong scalability

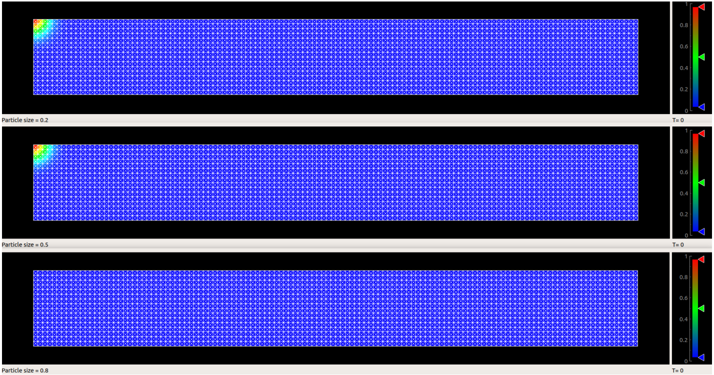


Strong scalability of the proposed parallel algorithm for different clusters. We obtain almost linear acceleration if number of processors is less or equal to 128.

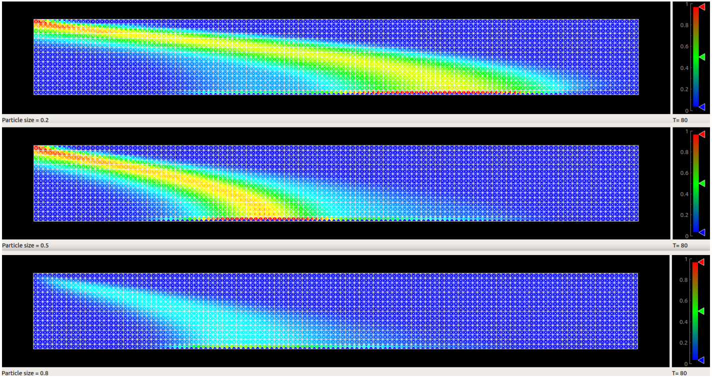
2D-advection case, model equations

$$\frac{\partial f(t,x,y,v)}{\partial t} + \nabla \cdot (\vec{c}(x,y,v) f(t,x,y,v)) =$$
$$= \frac{1}{2}\int_0^v K(u, v-u) f(t,x,y,u) f(t,x,y,v-u) du -$$
$$- f(t,x,y,v) \int_0^\infty K(u,v) f(t,x,y,u) du,$$

Solutions



Solutions


Particle size = 0.2
T= 80
Particle size = 0.5
T= 80
Particle size = 0.8
T= 80
1
0.8
0.6
0.4
0.2
0

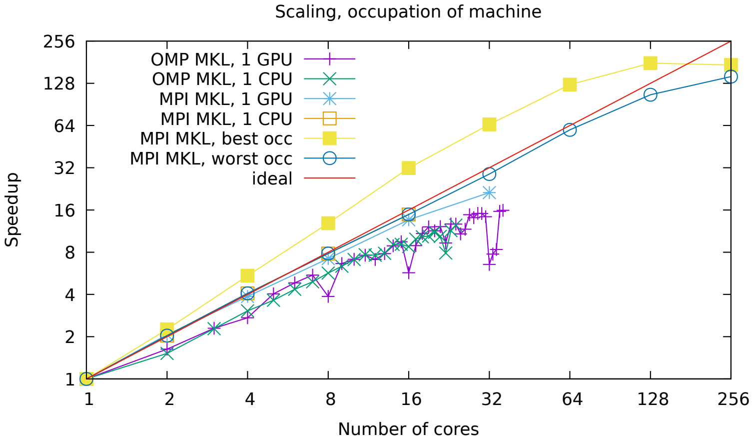
Speedups for pure CPU

P cores	time	speedup
1	159.312	1.0
2	74.6913	2.13
4	38.9939	3.98
8	21.54	7.4
16	10.5112	15.16
32	5.53312	28.81
64	2.94353	54.12
128	1.59911	99.63
256	0.931789	170.97
512	0.504464	315.8
1024	0.328353	485.19

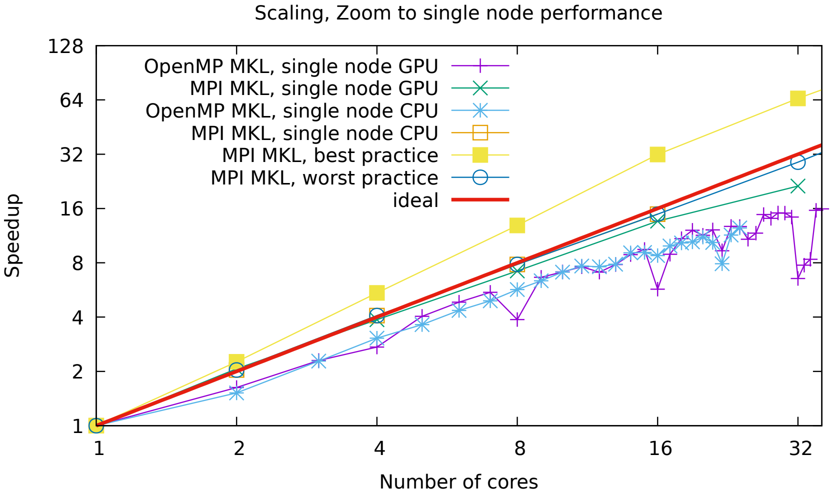
Hybrid CPU-GPU performance

P cores	no GPU, sec	speepdup	with GPU, sec	speedup
1	396.512	1.00	186.47	2.13
2	219.966	1.8	101.12	3.92
4	123.606	3.2	65.33	6.07
8	73.52	5.39	35.32	11.22
16	42.51	9.33	18.53	21.41
32	36.06	11.0	12.18	32.55

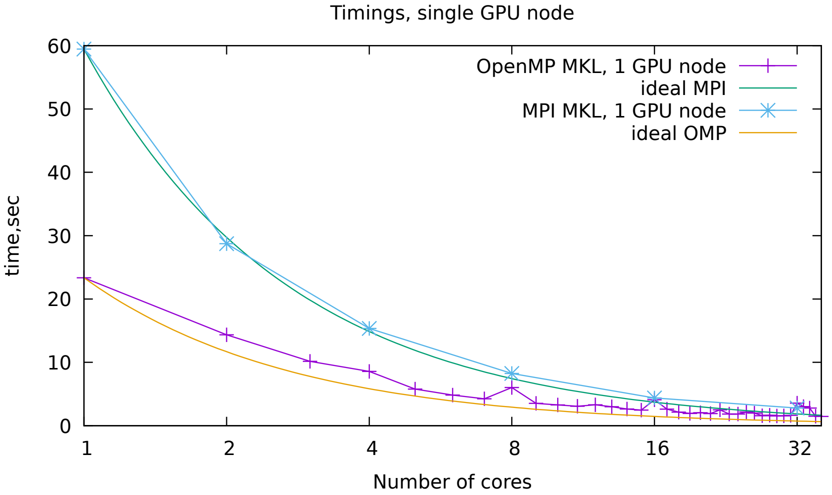
Testing parallel FFT



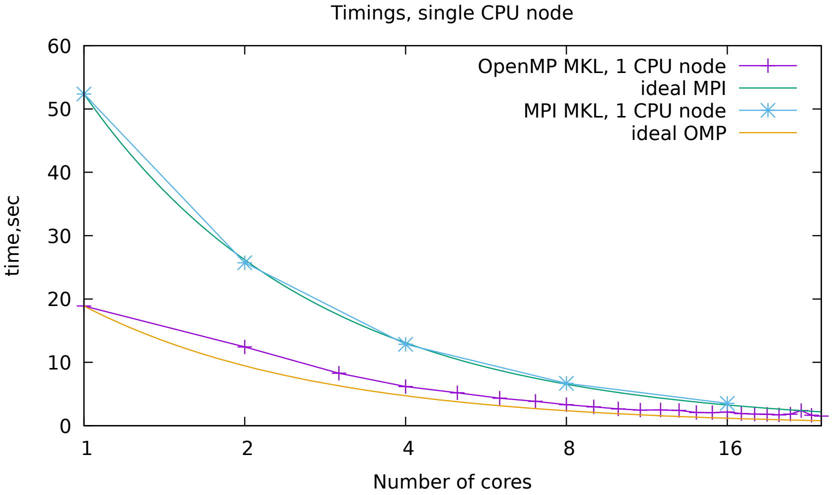
Testing parallel FFT



Testing parallel FFT



Testing parallel FFT



Thank you for your attention!