TSC method using semi-implicit method for spring mass simulation

Ryo Sagayama †1 Akihiro Fujii †1 Teruo Tanaka †1 Takumi Washio †2 Takeshi Iwashita †3
†1Kogakuin University †2Tokyo University †3Hokkaido University

Introduction

What is time evolution simulation?

- Simulation of how an object moves in a given time

Parallelization in spatial direction

- A method of assigning a process to each mass point
- When the number of processes is larger than the number of masses, the number of parallelism is limited

Parallelization in time direction

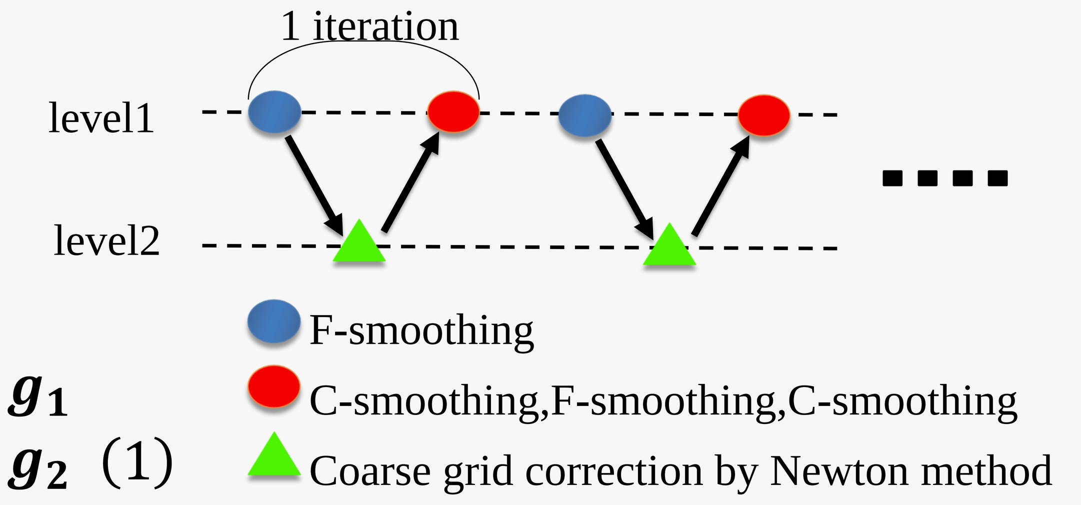
- Assigning a process to each time step
- An appropriate initial solution is given to each process
- The solution is updated iteratively until the overall residual is below a certain level
- No limit in the degree of parallelism in time-stepping simulation → Parallelize using the TSC method

TSC method

Overview

- It solves the simulation problem all at once using Newton method based on regarding the whole time steps as a nonlinear system
- Smoothing and Coarse grid correction composes one iteration of Newton update.

$$\begin{cases} \boldsymbol{u_0} = \boldsymbol{g_0} \\ A(\boldsymbol{u_1}, \boldsymbol{u_2}) = \boldsymbol{g_1} \\ A(\boldsymbol{u_2}, \boldsymbol{u_3}) = \boldsymbol{g_2} \quad (1) \\ \vdots \\ A(\boldsymbol{u_i}, \boldsymbol{u_{i-1}}) = \boldsymbol{g_i} \end{cases}$$



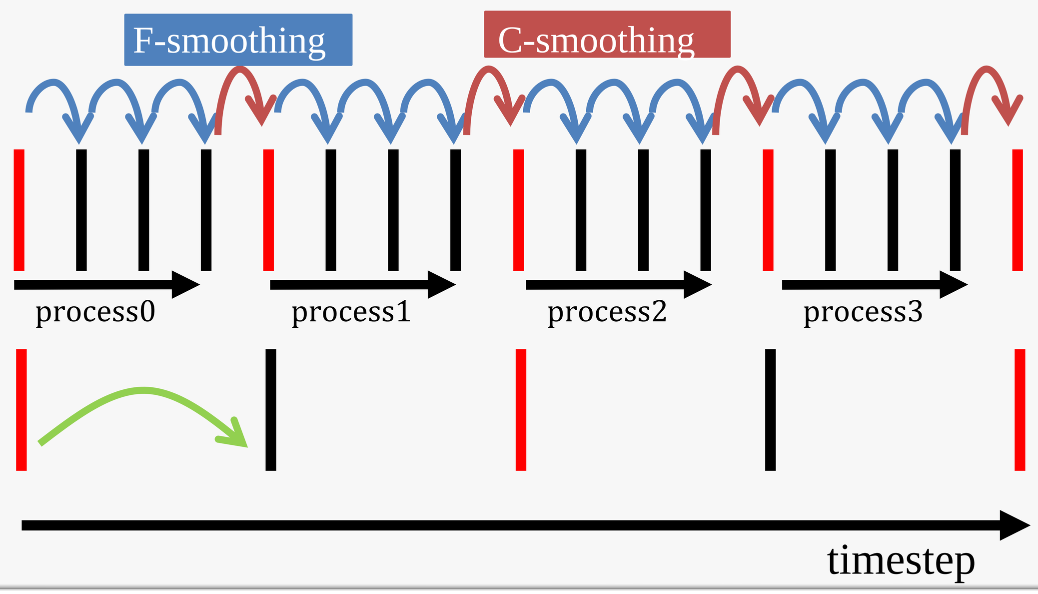
Smoothing

- F-smoothing corresponds blue arrows in right figure. Each sections are updated in parallel
- C-smoothing corresponds red arrows and passes the solution to the next process.

Coarse grid correction

- Solving Jacobi matrices for all time steps using Newton's method
- The linear problem becomes huge. The size of eq.(2) is space size x time steps
- Coarsening and then solving for correction

$$\begin{pmatrix} J_0 & & & \\ K_1 & J_1 & & \\ & & \ddots & \\ & & K_i & J_i \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{u_0} \\ \Delta\boldsymbol{u_1} \\ \Delta\boldsymbol{u_2} \\ \vdots \\ \Delta\boldsymbol{u_i} \end{pmatrix} = \begin{pmatrix} \Delta\boldsymbol{r_0} \\ \Delta\boldsymbol{r_1} \\ \Delta\boldsymbol{r_2} \\ \vdots \\ \Delta\boldsymbol{r_i} \end{pmatrix} \quad (2)$$



Semi implicit method

Outline of the method

- The time integration method calculates modified Jacobian matrix that has positive Eigen-values
- This allows the inverse matrix to be obtained stably when solving by the direct method
- In this study, only this Jacobi matrix generation method is applied to the TSC method
- P is elemental Jacobi matrix corresponding to one spring with two mass points

Parameter ω

- a parameter that determines the degree of correction
- The closer to 0, the more correction is applied, and the closer to 1, the less correction is applied

$$P = \max\left(-\omega \frac{\min(\gamma_i,\ \gamma_j)}{\Delta t}, 2\ddot{\varphi}(r_{ij})\right) N_{ij} + \max\left(-\omega \frac{\min(\gamma_i,\ \gamma_j)}{\Delta t}, 2\ddot{\varphi}(r_{ij})\right) M_{ij}$$

$$(0 \le \omega \le 1)$$

Numerical experiments

Target Physical Simulation

- Three mass-point system connected with springs subjects to random forces from the medium
- Number of time steps: 1024 to 131072
- Number of divisions (parallel number): 2 to 256

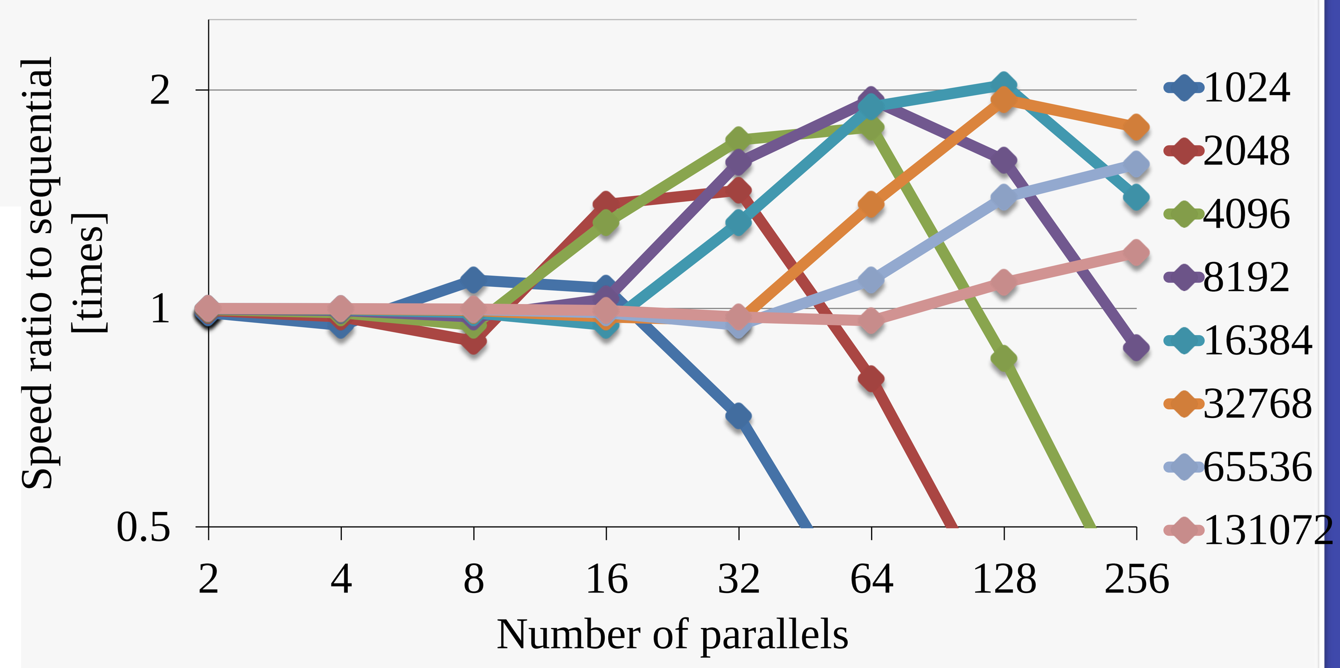
Experimental results

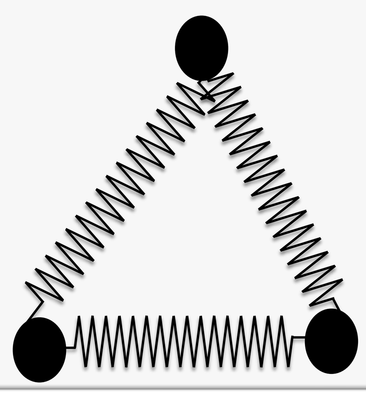
- When the TSC method is used, the maximum number of time steps is 16384 and the number of partitions is 128, the performance is about twice that of the Explicit Euler method
- When the semi-implicit method was applied, the number of iterations was improved at 16384 timesteps and 2048 divisions

When Omega was 1, the correction was too strong and the number of repetitions was too high

para\ ω	1	2	3	4	5	conv
	× 0.001					
2	1	1	1	1	1	1
4	2	2	2	2	2	2
8	4	4	4	4	4	4
16	8	8	8	8	8	8
32	13	12	11	10	10	10
64	16	12	10	10	10	10
128	17	11	9	8	8	8
256	17	10	7	6	6	6
512	18	10	7	6	6	6
1034	17	10	7	7	7	7
2048	17	10	9	9	10	10

In some places, as Omega was moved closer to 0 and the correction was weakened, the number of iterations decreased more than the original





Conclusion

Findings

- The TSC method can be used to achieve better performance in comparison with explicit Euler method
- Increasing the number of partitions and timesteps seems to improve the performance
- Convergence can be accelerated by using the semi-implicit method

Future Prospects

- Extend the number of partitions and time steps
- Experiment with semi-implicit methods on larger problem sizes
- Relax convergence conditions

Acknowledgments

This work was supported by JSPS KAKENHI Grant NumberJP22K12063 and JHPCN in Japan (Project ID jh220049).

Reference

[1] Akihiro Fujii, Shigeo Kaneko, Teruo Tanaka, Takeshi Iwashita, Time Segment Correction Method for Parallel Time Integration, Journal of Information Processing, Volume27, pp.822-830 (2019).

[2] Takumi Washio, Xiaoke Cui, Ryo Kanada, Jun-ichi Okada, Seiryo Sugiura, Yasushi Okuno, Shoji Takada, Toshiaki Hisada. Using incomplete Cholesky factorization to increase the time step in molecular dynamics simulations. Journal of Computational and Applied Mathematics 2022.