Symmetric Individual Timestep

Jun Makino and Piet Hut

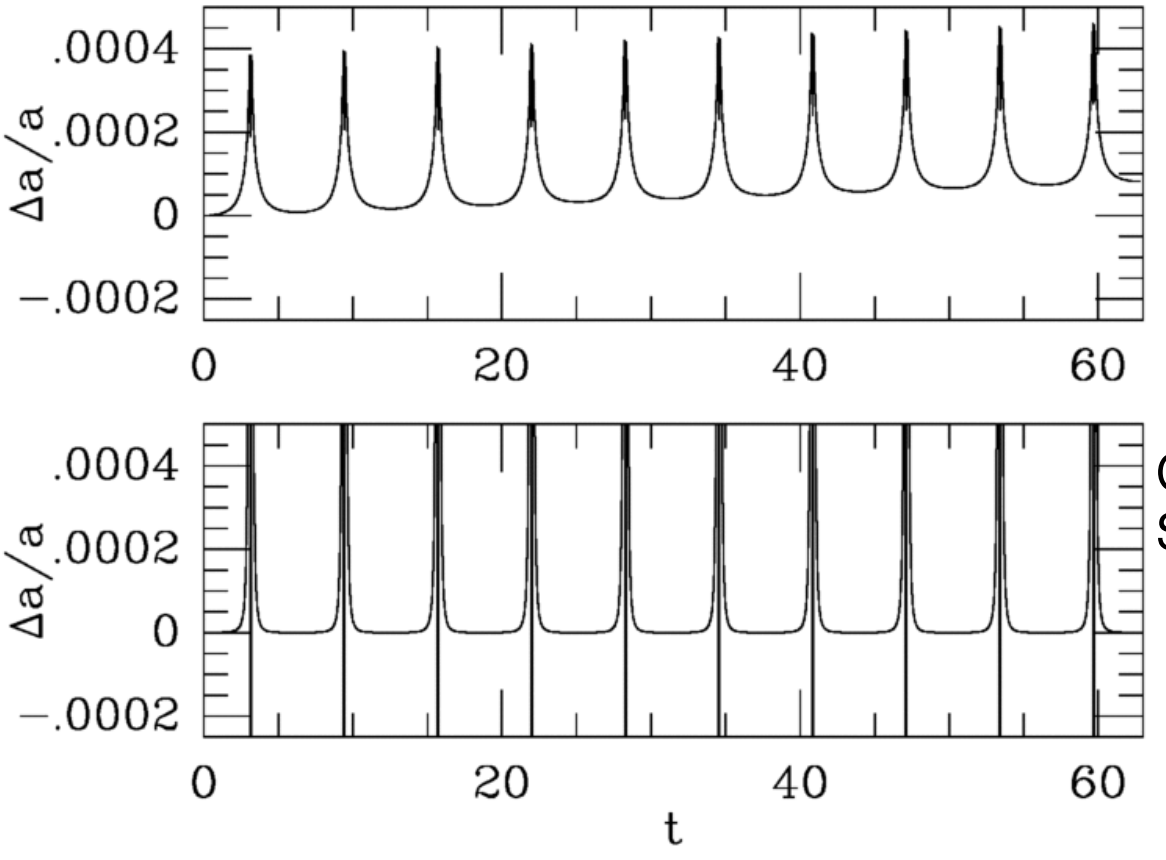
Talk overview

- **Symmetric timestep**
- **Block-Symmetric timestep**
- **Symmetric individual timestep**

Symmetric timestep

- **Time-symmetric integration scheme (such as leapfrog) does not have long-term error in energy.**
- **With usual variable timestep, however, it shows linear error**

Leapfrog example

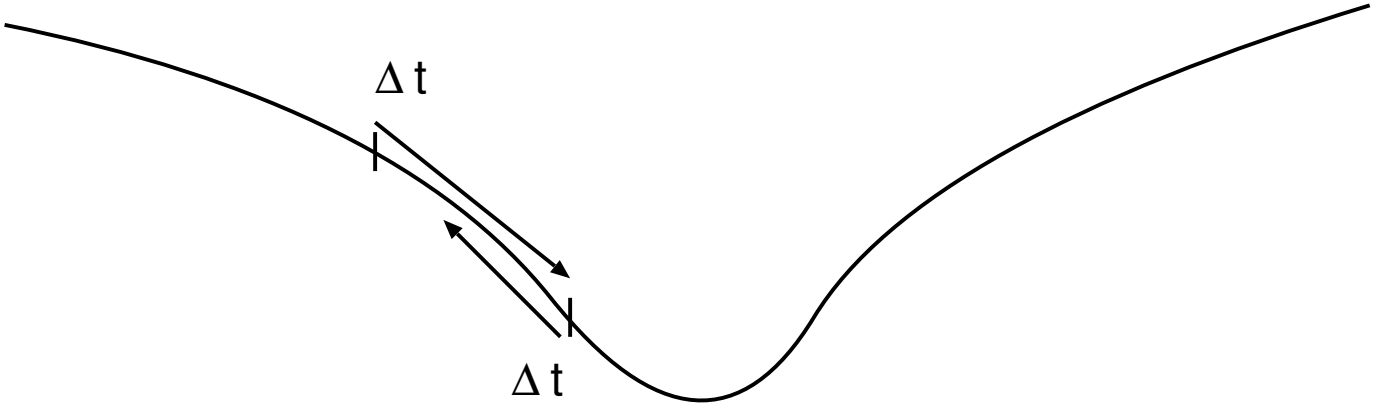


Variable Stepsize

Constant Stepsize

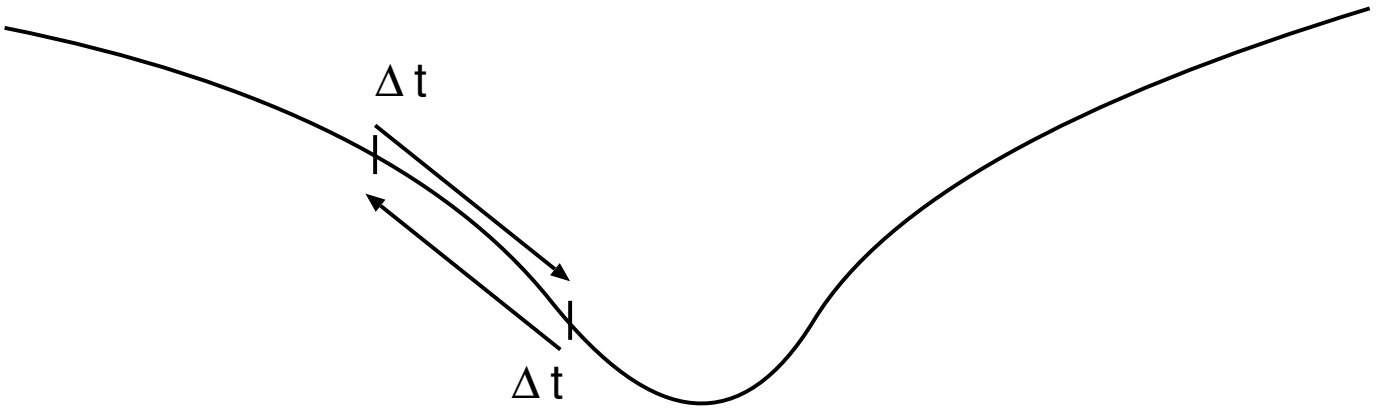
Constant step is better...

Non-symmetry of TIMESTEP



Timestep you calculate in forward path is generally not the same as that you calculate when you go backward.

Symmetrized TIMESTEP



You need to find some way to force the forward path and backward path to have exactly the same timestep.

Algorithm

If timestep depends only on instantaneous physical quantities:

$\Delta t_0 = \xi(x_0)$

$\Delta t_1 = \xi(x_1)$

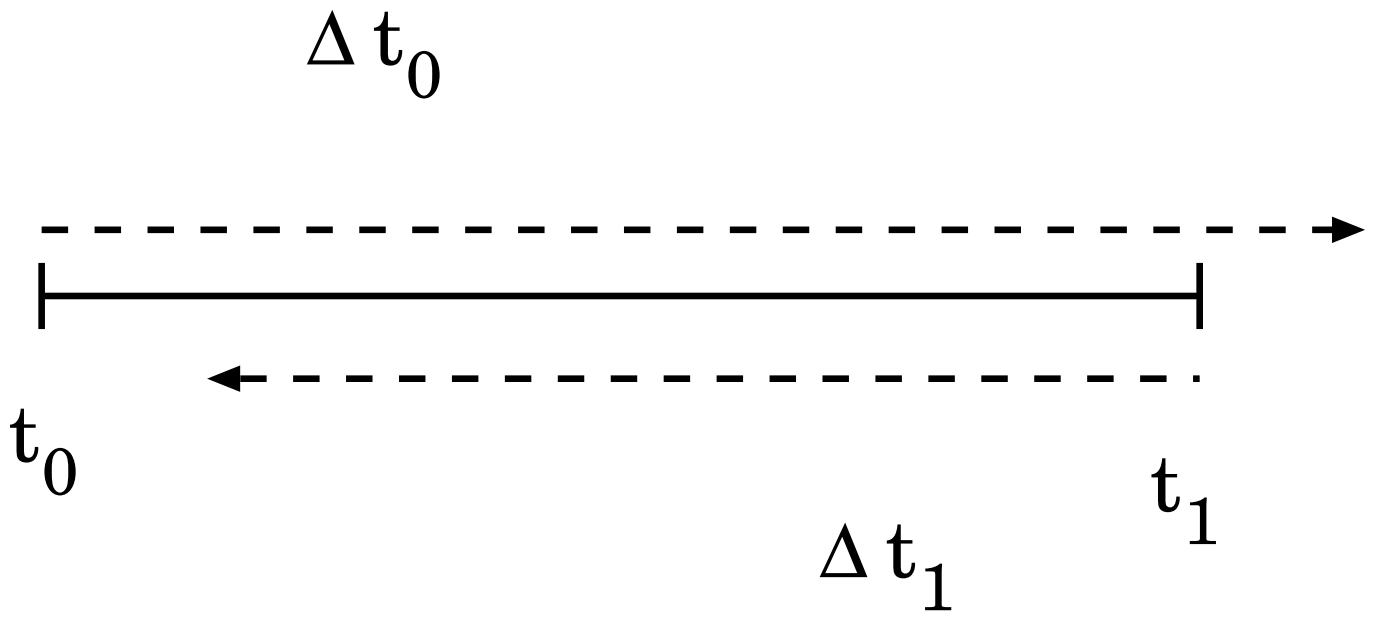
Some symmetric function of two values,
$f(t_0, t_1) = f(t_1, t_0)$**, would do the work.**

Example:

$f(x, y) = (x + y)/2$

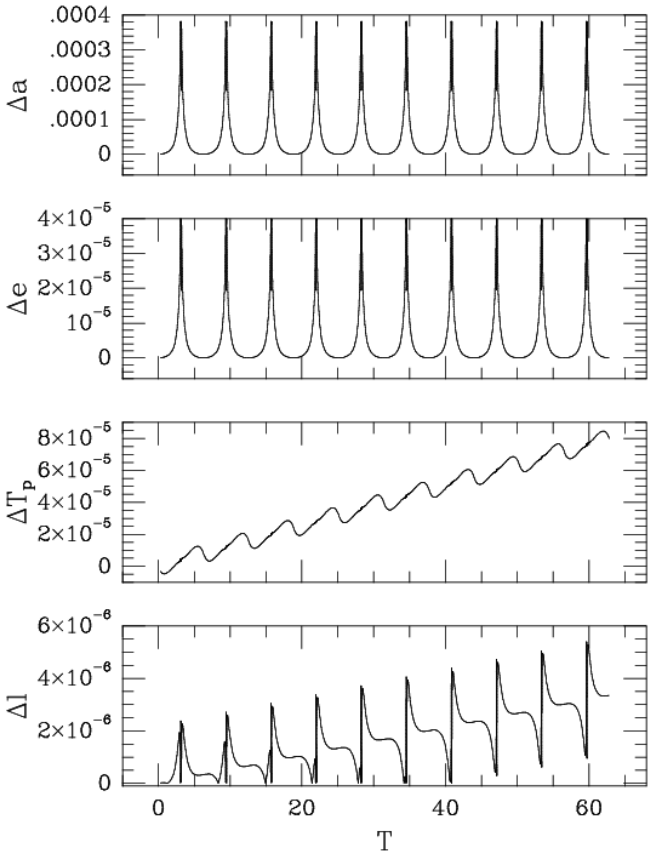
$f(x, y) = \min(x, y)$

Iterative solution

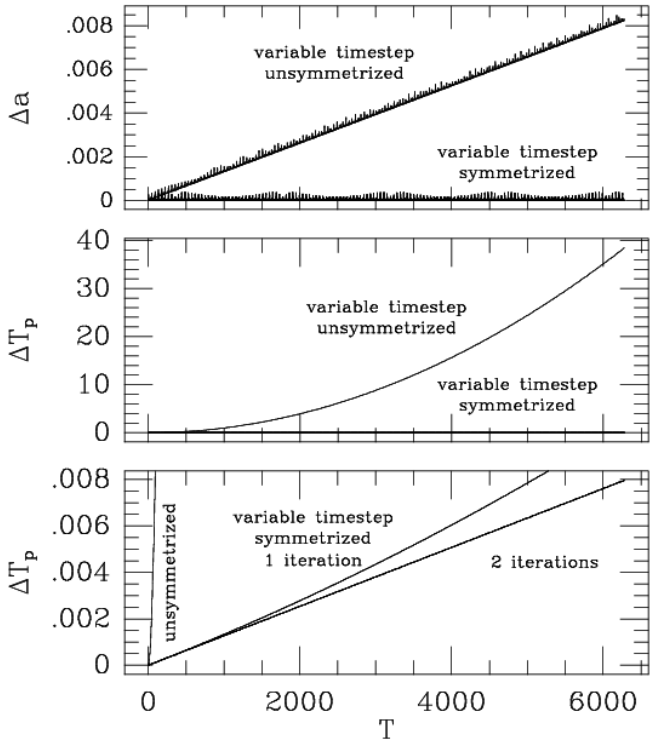


calculate t_1 using some guess for t_1.
Clever iteration converges pretty fast (2-3 iterations)

Time-symmetric timestep: result



Time-symmetric timestep: result(2)

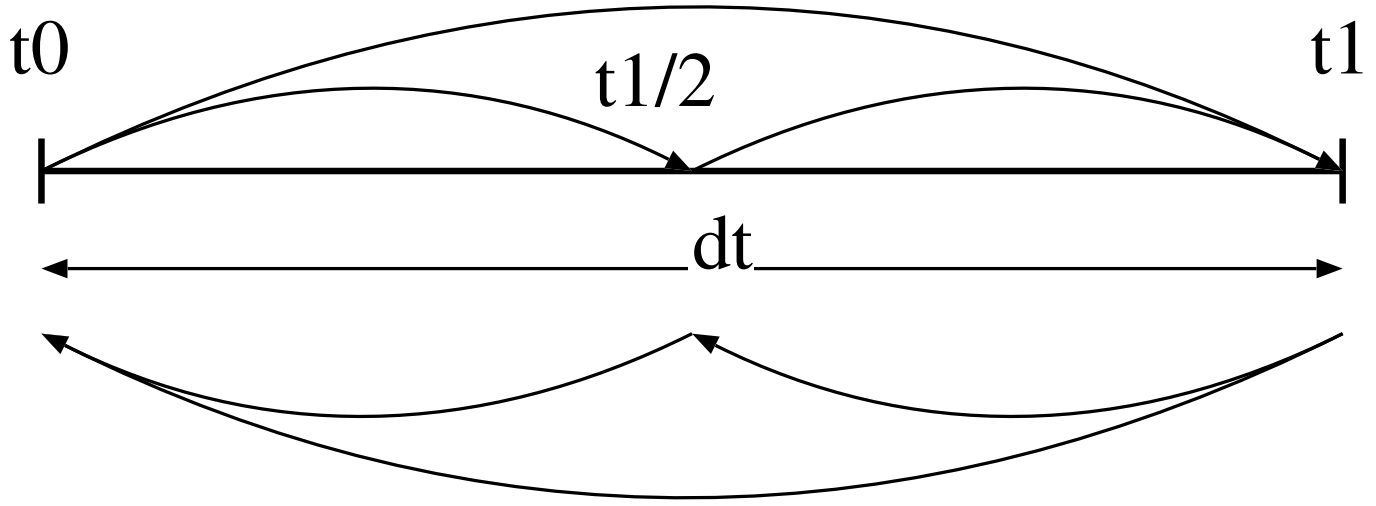


Block/individual timestep

Two problems:

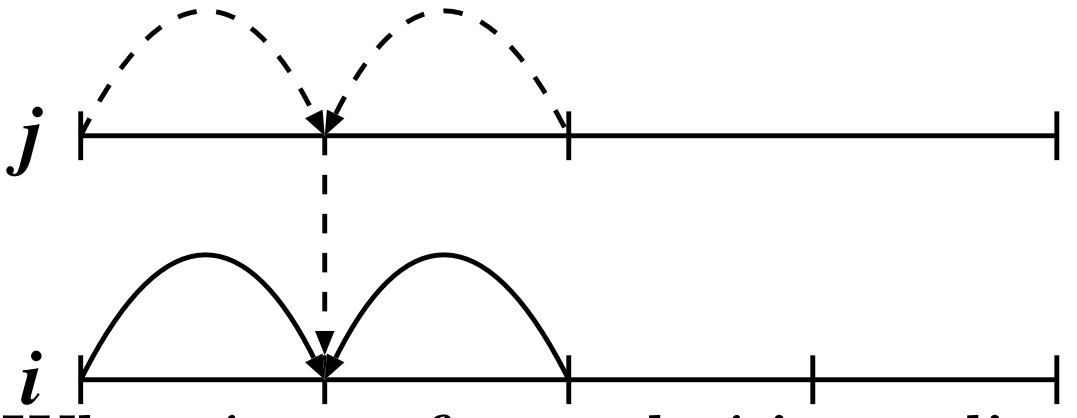
- **timesteps should be 2^k**
- **Different particles have different times and timesteps**

Blockstep



If you can go to t_0 to t_1, you can go back.
If you can go to t_0 to $t_{1/2}$, and $t_{1/2}$ to t_1, you can go back.
You are not allowed to go from $t_{1/2}$ to $t_{1+1/2}$
Time symmetry is okay.

Individual blockstep



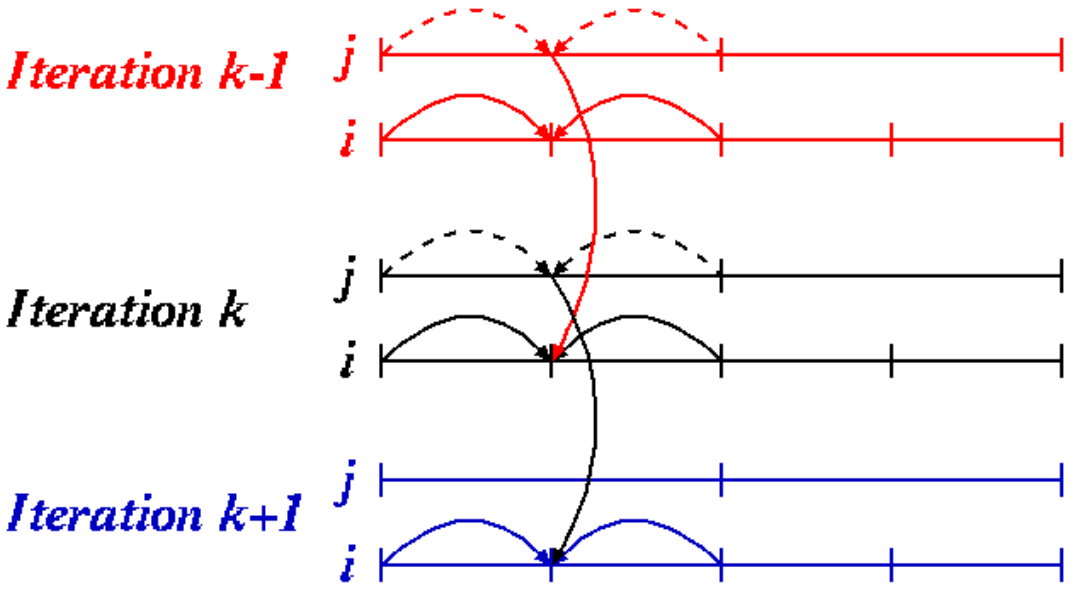
When i goes forward, j is predicted forward

When i goes backward, j is predicted backward

How forward and backward prediction can give the same result?

We need to know the futre of j before we push i.

Iterative Symmetric blockstep

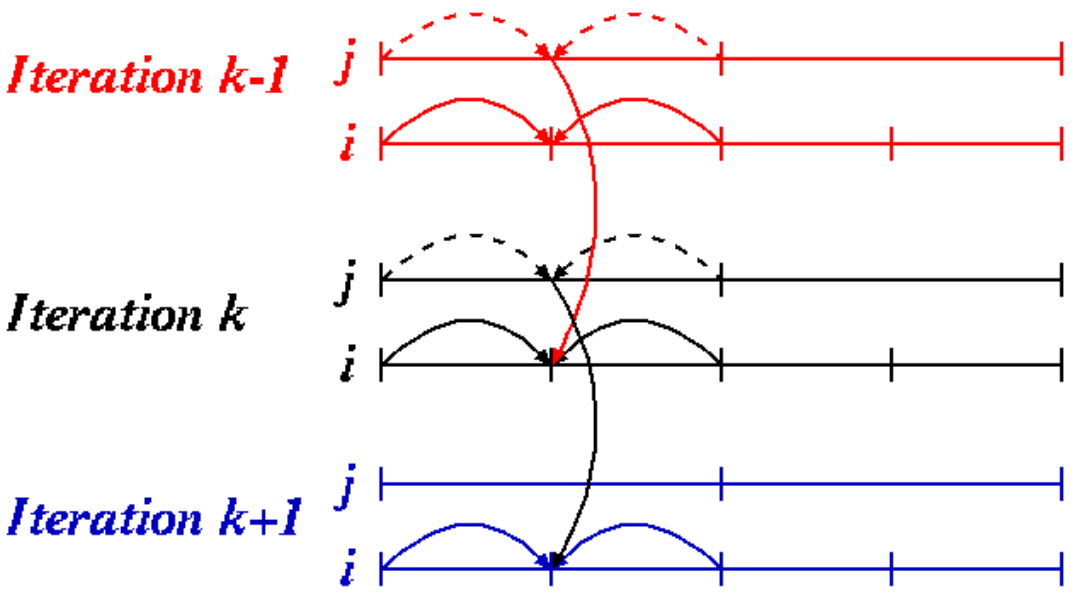


Integrate for some fixed period of time (for exmple, 0 to 1/32, 1/32 to 2/32 ...)

Store all the steps for that period of time.

Iterate the time integration, using the previous iteration to "predict"

Iterative Symmetric blockstep

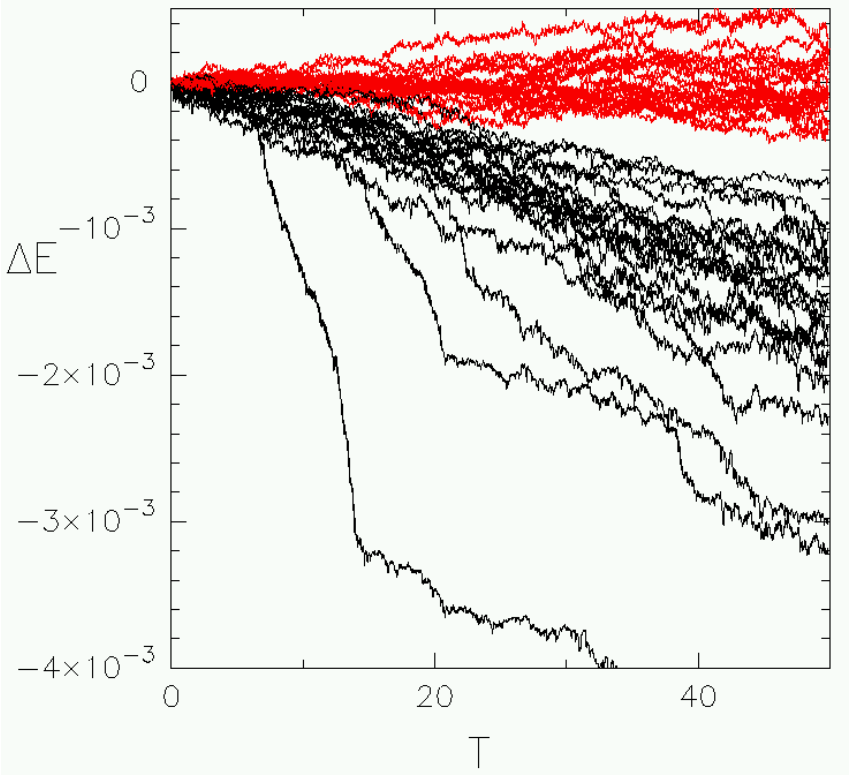


Integrate for some fixed period of time (for exmple, 0 to 1/32, 1/32 to 2/32 ...)
Store all the steps for that period of time.

Iterate the time integration, using the previous iteration to "predict"

Sounds too complicated?

Result(1) N=100

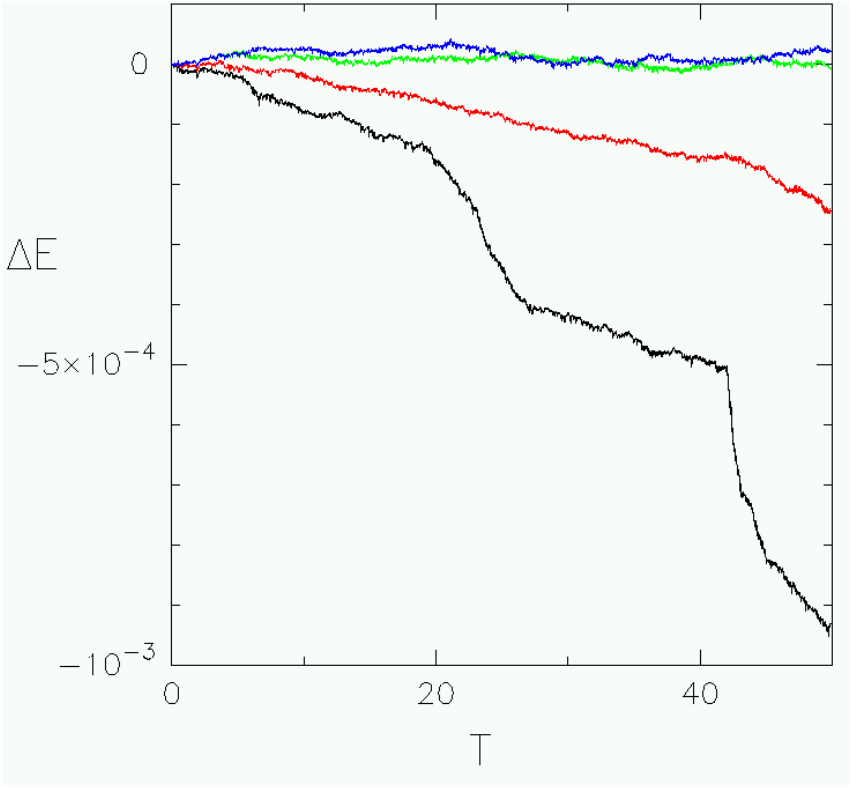


Implementation entirely in C.
(No Ruby, C++...)

$N = 100$ Plummer model
$\epsilon = 0.01$

Red curves: Symmetrize with 6 iterations
Black: No iteration

Result(2) N=512



$N = 512$ Plummer model

$\epsilon = 1/512$

0, 1, 2, 3 iterations

Summary

- **Fully-iterative block-individual time-symmetric timest algorithm can be implemented.**
- **It works.**
- **Convergence is fast even for a primitive iteration scheme.**
- **Improvement is much bigger for larger N.**