Fys4150
Project 3

Peter Killingstad and Karl Jacobsen

https://github.com/kaaja/fys4150

October 10, 2017

Note to instructurs about Github repository

If the above Github-link does not work, it is eighter because you have not yet accepted our invite to the repository, or you have not yet provided us with an e-mail adress available at Github so that we can invite you. If the latter applies to you, please send us an e-mail with an e-mailadress available in Github or your Github username so that we can send you an invite. Our e-mailadresses: peter.killingstad@hotmail.com, karljaco@gmail.com.

Abstract

1 Introduction

2 Theory

3 Results

3.1 Sun-Earth Forward Euler

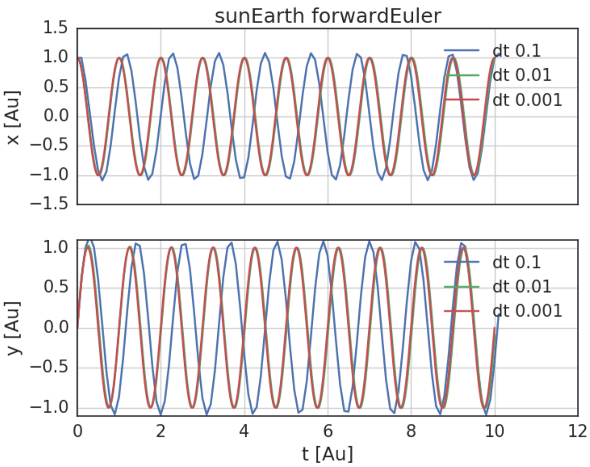

Figure 1: Sun-Earth system. Effect of Δt over a 10 year period.
The Forward Euler method seems to converge for the two smallest Δt

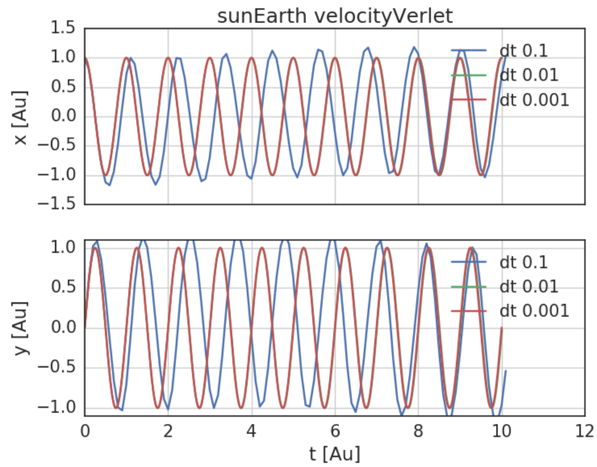

Figure 2: Sun-Earth system. Effect of Δt over a 10 year period.
The Velocity verlet method seems to converge faster than Forward Euler

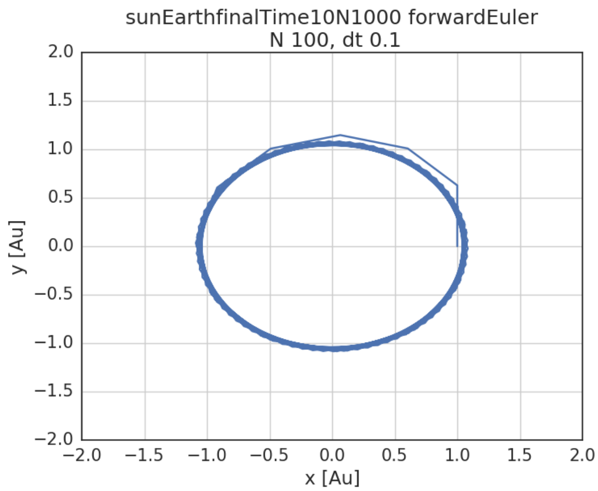

Figure 3: Sun-Earth system. Forward Euler. 10 years
Non-circulat orbits when time step is large.

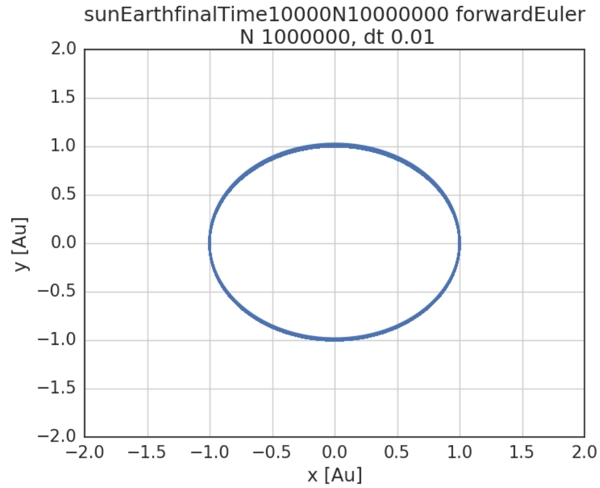

Figure 4: Sun-Earth system. Forward Euler. 10 000 years.
For $\Delta t = 0.01$, the forward Euler seems to give circular orbits, but we can see that the solution changes.

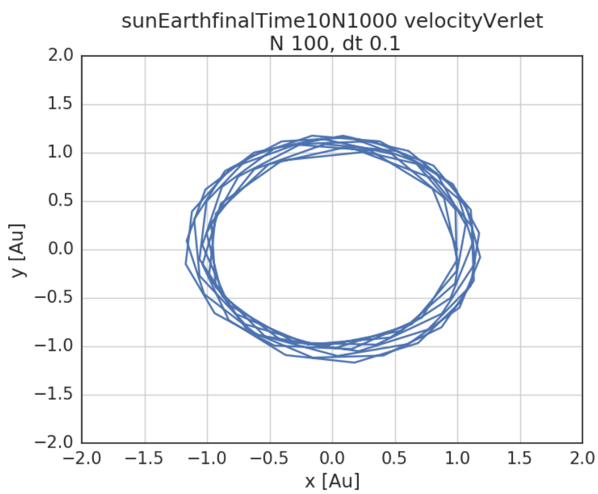

Figure 5: Sun-Earth system. Velocity Verlet. 10 years.
Large time step gives bad solutions also for Velocity Verlet.

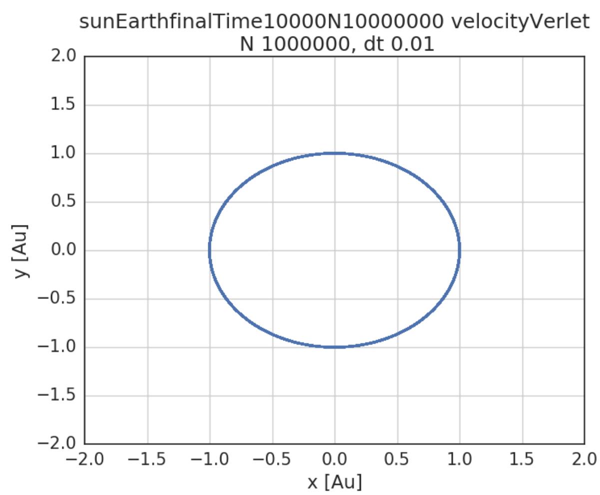

Figure 6: Sun-Earth system. Velocity Verlet. 10 000 years.
For velocity Verlet, the orbits seems to stay more circular compared to Forward Euler.

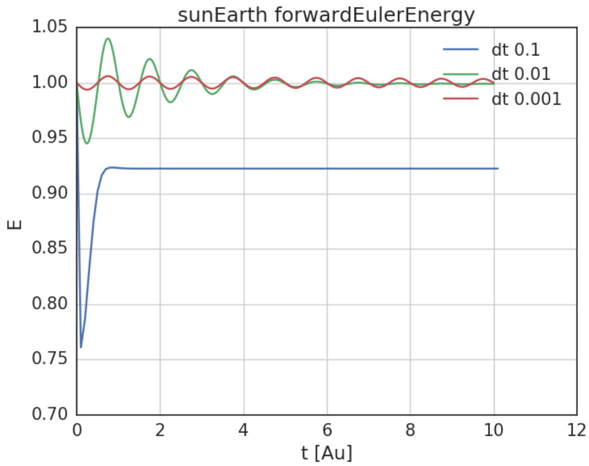

Figure 7: Sun-Earth system. Total Energy divided by total energy first time step. Forward Euler. 10 years. *Energy is not preserved with the Forward Euler method*

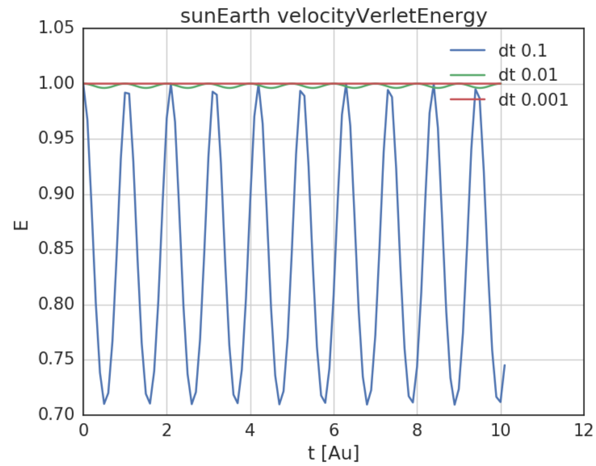

Figure 8: Sun-Earth system. Total Energy divided by total energy first time step. Velocity Verlet. 10 years. *Energy is preserved in Velocity Verlet provided fine enough time step.*

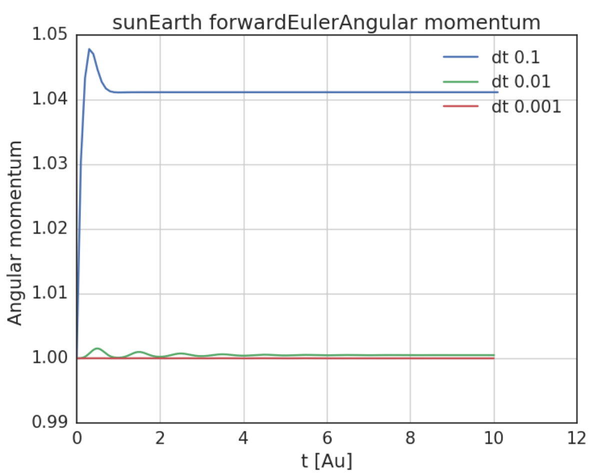

Figure 9: Sun-Earth system. Angular momentum divided by angular momentum first time step. Forward Euler. 10 years.
Angular momentum seems to be conserverd for the finest time step.

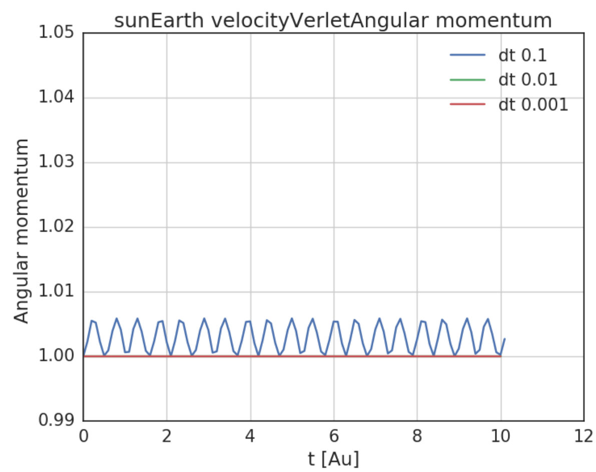

Figure 10: Sun-Earth system. Angular momentum divided by angular momentum first time step. Velocity Verlet. 10 years.
Angular momentum is conserverd given suffiently fine time steps. Conservation achieved faster than with Forward Euler.

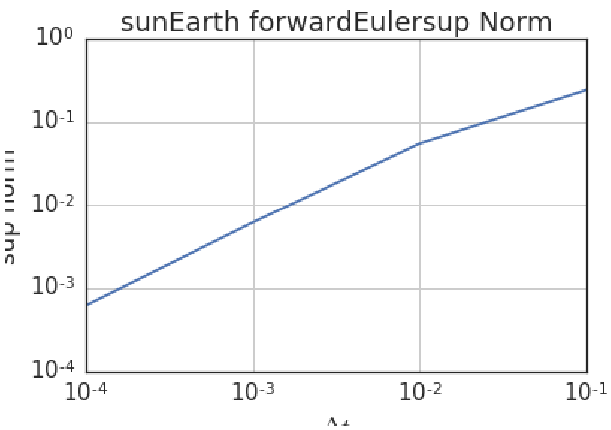

Figure 11: Sun-Earth system. Sup-norm total energy. Forward Euler.
Forward Euler's sup-norm goes like $\mathcal{O}(\Delta t)$

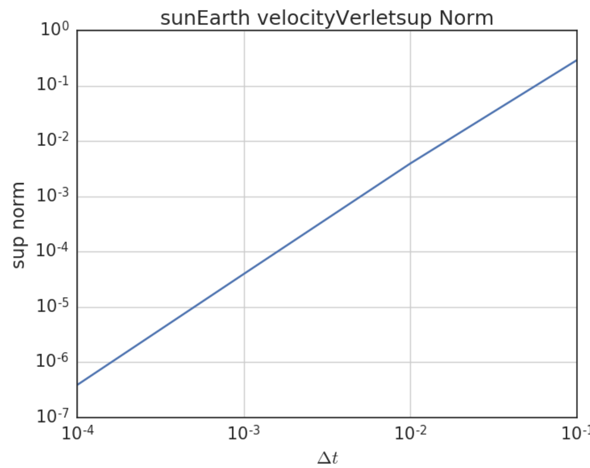

Figure 12: Sun-Earth system. Sup-norm total energy. Velocity Verlet
The sup-norm in energy for Velocity Verlet goes one higher order than Forward Euler

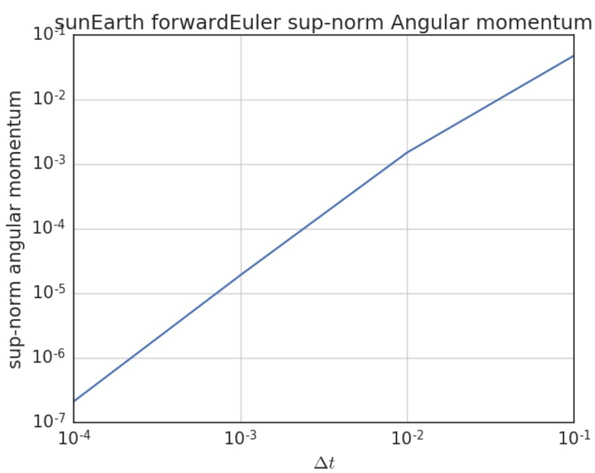

Figure 13: Sun-Earth system. Sup-norm Angular Momentum. Forward Euler
Forward Euler's sup-norm for angular momentum goes like $\mathcal{O}(\Delta t^2)$.

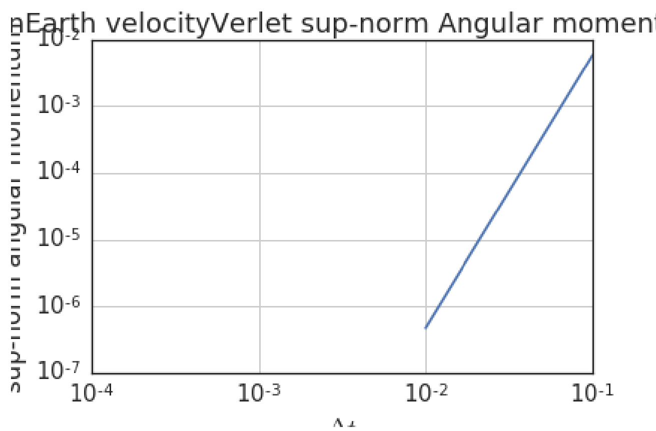

Figure 14: Sun-Earth system. Sup-norm Angular momentum. Velocity Verlet
Velocity Verlet's sup-norm error less than 10^{-12} like $\mathcal{O}(\Delta t^4)$.

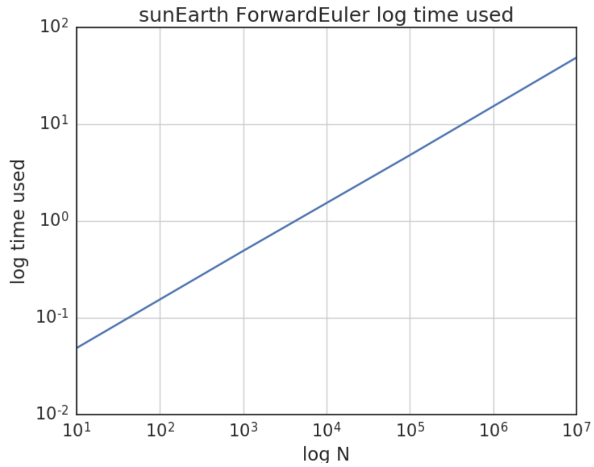

Figure 15: Sun-Earth system. Log time. Forward Euler *Forward Euler's $\mathcal{O}(N)$.*

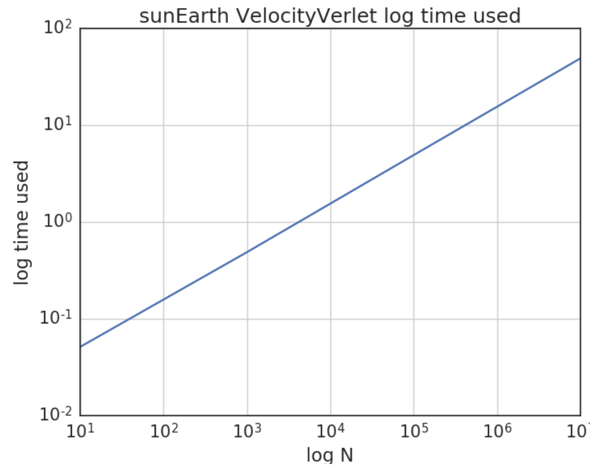

Figure 16: Sun-Earth system. Log time. Velocity Verlet *Velocity Verlet's log time is of the same order as Forward Euler.*

4 Conclusions

5 Feedback

5.1 Project 1

This project has been extremely educational. We learned about about c++, especially pointers and dynamic memory allocoation. Also which for us was a well forgotten subject, we learned about dangerous of numerical round-off errors.

We feel the size of the project is large, much larger than typical assignments in other courses. However, the quality and quantity of the teaching without a doubt made the workload managable. The detailed lectures, combined with the fast and good respones on Piazza helped a lot!

We think the project could have gone even smoother, if we on the 2nd lab-session had learned basic branching in Github. We used a considerable amount of time finding out of this.

All in all, two thumbs up!

5.2 Project 2

- catch: We ended up using a lot of time making this work properly. Still we have some problems with catch and Qt. We think we might had benefited from a demonstration at the lab.
- We were not able to understand the revised Sturm-Bisection algorithm from Barth et al.'s [1] paper on the revised Sturm-Bisection.
- Apart from the small details above, we are very happy about this project. How would have thought linear algebra could be fun?!

6 Bibliography

[1] Barth, Martin, Wilkinson (1967) Calculation of eigenvalues of a symmetric tridiagonal matrix by the method of bisection. *Numeriche mathematik 9, 386 - 393 (1967)*

[2] Hjorth-Jensen, M.(2015) Computational physics. Lectures fall 2015. `https://github.com/CompPhysics/ComputationalPhysics/tree/master/doc/Lectures`

[3] Hjorth-Jensen, M.(2017) Project 2, fys4150 2017. `https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Projects/2017/Project2/pdf/Project2.pdf`

[4] Kiusalaas, J.(2013) Numerical Methods in Engineering with Python 3. 3rd edition.

[5] Taut, M. (1993) Two electrons in an external oscillator potential: Particular analytic solutions of a Coulomb correlation problem *Phys. Rev. A 48, 3561 (1993).*