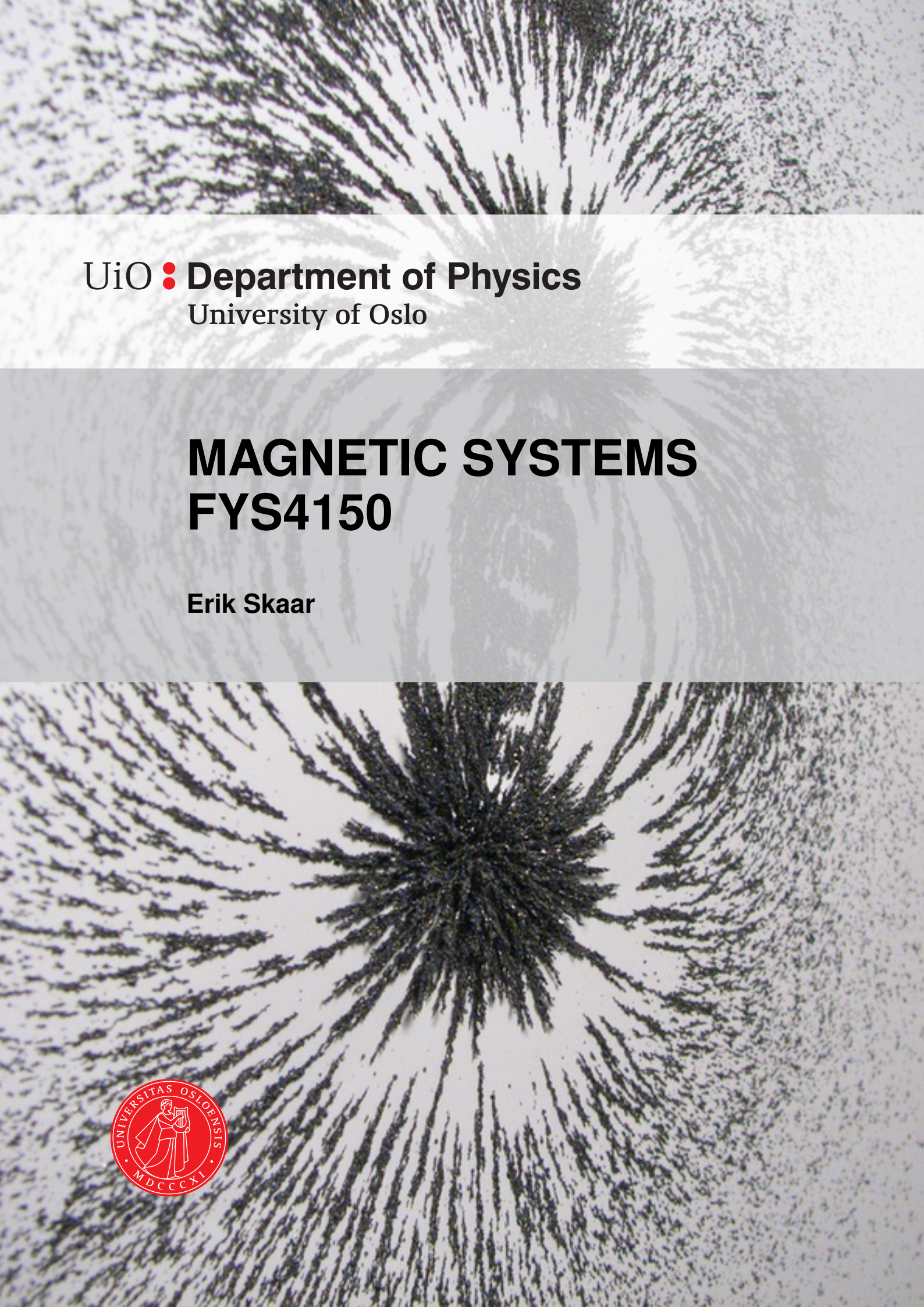


UiO : Department of Physics
University of Oslo

MAGNETIC SYSTEMS
FYS4150

Erik Skaar

Contents

Abstract

1 Introduction

These laws are not enough to solve the motion of the planets. From the laws one can derive differential equations for the motion, which are not trivial or even possible to solve analytically. This is where computational methods are useful. With the tools developed in computtational physics we can make a prediction to the motion of the planets in our solar system.[1] And because of our assignment we kind of have to do this to pass the course.[1]

[1] Semester page for FYS3150 - Autumn 2017.

2 Theory

2.1 the Ising model

2.1.1 Partition function

2.1.2 Energy

2.1.3 the mean magnetic moment |M|

2.1.4 the specific heat C_V

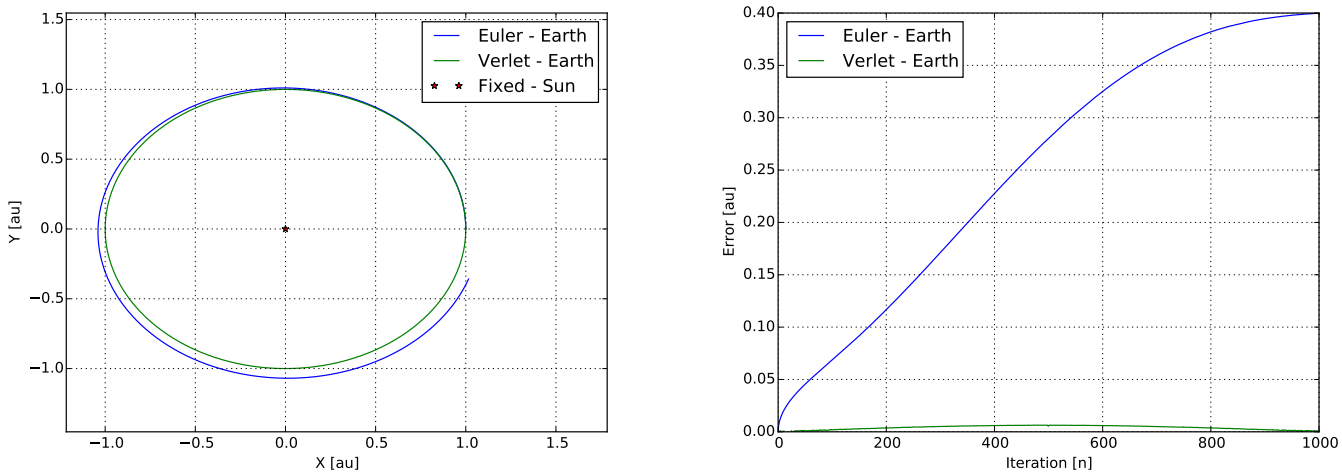
2.1.5 the susceptibility χ

3 Method

4 Result & Discussion

4.1 Earth-Sun system

4.1.1 Stability



(a) (b)

Figure 1: a) shows the orbit of earth around the sun. The intial velocity is set to 2π in y direction and the start position to 1 au in x direction. b) shows how the error develops. The intial values should give a perfect circular motion. So the error is calculated by $r_i - r_0$. It is apparent that the Verlet-Velocity method is a better approximation. This simulation was with 1000 points with the end time of 1 year. Both simulations was produced by plot_earth_sun.py

Table 1: Time table for the different algorithms. The algorithms use nearly the same time. This is not a shocker since the number of FLOPs for the algorithms are similar, see section (**??**). Note: this is only the result from one test, but several was done. Both algorithms were very close and it seems to be random which is fastest.

n	Forward-Euler	Verlet-Velocity	fastest	$\frac{slowest}{fastest}$
10	0.000136	0.000148	Euler	1.08823529412
100	0.000208	0.000179	Verlet	1.16201117318
1000	0.000392	0.000389	Verlet	1.00771208226
10000	0.002427	0.002426	Verlet	1.00041220115
100000	0.022931	0.022293	Verlet	1.02861884897
1000000	0.167022	0.175944	Euler	1.05341811258
10000000	1.58721	1.52666	Verlet	1.03966174525
100000000	15.1786	15.1176	Verlet	1.00403503202

5 Conclusion

6 References

References

[1] Morten Hjorth-Jensen. *Computational Physics*. Project-4. 2017. URL: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Projects/2017/Project4/pdf/Project4.pdf.

7 Appendix

```
//FLOPs FOR ACCELERATION
// 1 FLOP * 3 directions
dx = x1 - x2
// 3 FLOPs
r = sqrt(dx*dx + dy*dy + dz*dz)
// 7 FLOPs
a = - (Gconst*m*M/(r*r)) / (m*r)
// 2 FLOPs * 3 directions
a = a + a*(x1-x2);
//TOTAL FLOPs = 19 FLOPs


//FLOPs FOR POSITION :: EULER
// 2 FLOPs * 3 directions
x = x + t_step*Vx
//TOTAL FLOPs = 6 FLOPs


//FLOPs FOR VELOCITY :: EULER
// 2 FLOPs * 3 directions
Vx = Vx + t_step*ax
//TOTAL FLOPs = 6 FLOPs


//FLOPs FOR POSITION :: Verlet
// 6 FLOPs * 3 directions
x = x + t_step*Vx + (0.5*t_step*t_step*a);
//TOTAL FLOPs = 21 FLOPs


//FLOPs FOR VELOCITY :: Verlet
// 4 FLOPs * 3 directions
Vx = Vx + (0.5*t_step*(Ax+Ax_old));
//TOTAL FLOPs = 12 FLOPs
```

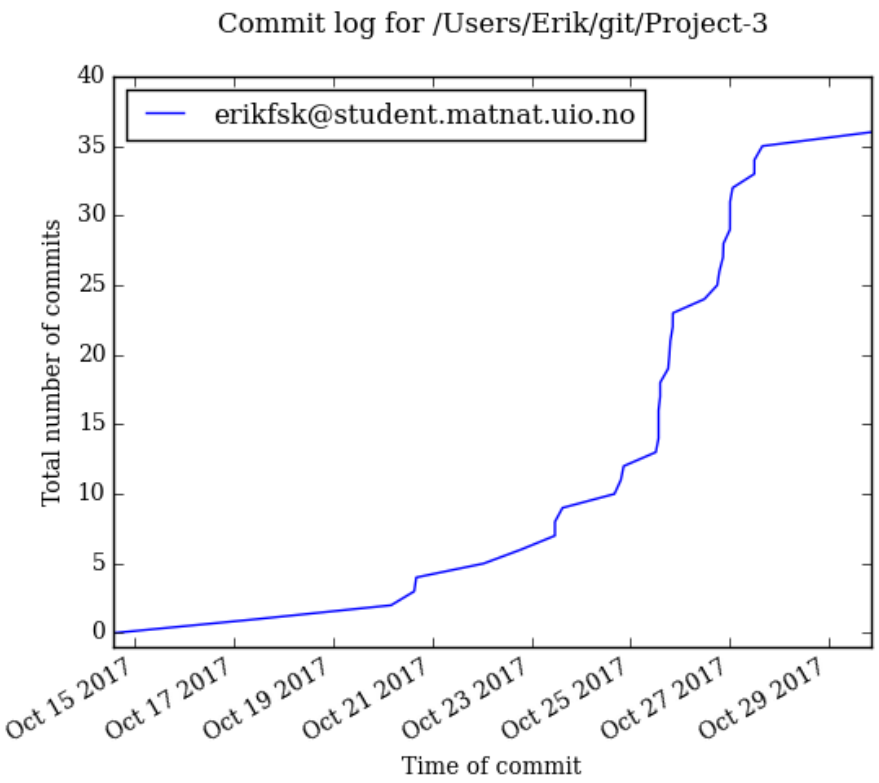


Figure 2: Our retarded workflow... Next time maybe it will be better?