NTNU
Department of Mathematical Sciences

To be filled in

Sindre Eskeland

December 4, 2015

MASTER

Department of Mathematical Sciences

Norwegian University of Science and Technology

Supervisor: Professor Elena Celledoni

Preface

To be filled in!

Acknowledgment

To be filled in.

Summary and Conclusions

To be filled in!

Contents

Preface i
Acknowledgment ii
Summary and Conclusions iii

1 Introduction **2**

2 Explonation **3**

2.1 Projection methods 3
2.2 Zero initial condition 4
2.3 Discretization 4
2.4 Lingo 6

3 Compare time integration methods **8**

4 Energy preservation for SLM **10**

Chapter 1

Introduction

Skrive litt om problemet, layouten og slikt.

!!!!!!!!!!!!!!!!!!!!!!cite rapport og SLM !!!!!!!!!!!!!!!

!!Cite hamiltonian!!!!!!!!!!!!!!!!!!!!

The equation

$$\begin{aligned} \dot{u}(t) &= Au(t) + F(t) \\ u(0) &= u_0 \end{aligned} \tag{1.1}$$

often makes an appearance when solving partial differential equations with numerical methods. The author has earlier observed how the heat equation, discretized with finite difference methods to be on the form of equation (1.1) can be solved with the use of the Krylov projection method(KPM) !!!!Cite rapporten min!!!!. This note will continue on the same track, with more focus on the wave equation, and energy preservation. It will also feature a comparison between Symplectic Lanzcos method(SLM) !!!Cite!!! and KPM. SLM is a projection technique that only works on Hamiltonian matrices. Due to this, SLM (claim to) preserve energy better than KPM.

Chapter 2

Explonation

There will here be a short explanation of all solvers, constants, outdata and expressions used in this text. Matlab notation is used where applicable. Fell free to go straight to next chapter, and use this as a reference later in the note.

2.1 Projection methods

As mentioned earlier, two projection methods will be presented, KPM and SLM. They are very similar in nature, with the only difference being the orthogonalisation method. KPM uses Arnoldi's algorithm, given in Algorithm **??**, and SLM uses the symplectic lanczos method, given in Algorithm **??**. The framework is given in Algorithm **??**. !!!!!!!!!!!!!Dette kommer snart!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!kopiere inn algoritmene!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!Skrive litt om de!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!Skrive litt om hvor de kommer fra!!!!!!
Algorihms:

Arnoldi

Symplectic Lanzcos method

forklare hva u_n er!

Må også forklare dirmet

2.2 Zero initial condition

For both KPM and SLM it is important that the initial conditions are zero. Equation (1.1) can be transformed so that it has zero initial conditions in the following way:

$$\hat{u}(t) = u(t) - u_0$$

The original equation can then be written as

$$\begin{aligned} \hat{\hat{u}}(t) &= A\hat{u}(t) + Au_0 + F(t) \\ \hat{u}(0) &= 0 \\ u(t) &= \hat{u} + u_0 \end{aligned} \quad . \tag{2.1}$$

All test problems with a non-zero initial condition will be transformed in this way before they are solved.

2.3 Discretization

!!!!!!!!!!!!!!!!!Figurer over simulert tid må være med ekte simulert tid fra 0 til 1(tror dette er fikset!)!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!Mange av bildene skal ha k og ikke time på x-aksen(tror dette er fikset)!!!!!!!!!!!!!!!!!!!!!!!!

The number of points in each spacial direction is m, this gives makes the step size $h_s = 1/m$. The number of steps in time is k, this makes the step size $h_t = 1/k$.

Let the matrix I be the identity matrix of appropriate size, and let

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \tag{2.2}$$

of appropriate size.

Equation (1.1) can be the result of the discretization of several equation. Since SLM needs a Hamiltonian matrix, this is the main focus. One of the two implemented matrices is the discretization of the 2 dimensional wave equation,

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial \mathbf{x}} + f(t, \mathbf{x}). \tag{2.3}$$

This can be discretized to be on the form of equation (1.1), with the matrix

$$\begin{aligned} \tilde{A} &= \frac{2}{h_s^2} \text{gallery}('\text{poisson}', m-2) \\ A &= \begin{bmatrix} 0 & I \\ -\tilde{A} & 0 \end{bmatrix} \end{aligned} \quad . \tag{2.4}$$

The matrix $\tilde{A}$ is also known as the five-point stencil. This matrix will be referred to as `wave` when used. The second implemented matrix is a random Hamiltonian matrix, given by

$$\hat{A} = \text{rand}(2 * (m-2)^2) \, A = \frac{1}{2} J(\hat{A} + \hat{A}^\top + m^2 I). \tag{2.5}$$

Since we are interested in comparing the different projection methods to each other, the matrix will be saved and reused when necessary. This matrix will be referred to as `semirandom`. The part $m^2 I$ is added to make JA diagonally dominant, there would be no way of knowing if any of the methods would converge without this part.

These two matrices also has some test problems that satisfies the condition: $u(t,0,y) = u(t,1,y) = u(t,x,0) = u(t,x,1) = 0$.

In the case when the energy is constant and `wave` is used, the test problem is

$$\begin{aligned} u(t,x,y) &= \sin(\pi x)\sin(\pi y)\cos(\sqrt{2}\pi t) \\ u_0 &= \sin(\pi x)\sin(\pi y) \\ f(t,x,y) &= 0, \end{aligned} \tag{2.6}$$

and

$$\begin{aligned} u(t,x,y) &= \text{unknown} \\ u_0 &= \text{rand}(2(m-2)^2, 1) \\ f(t,x,y) &= 0 \end{aligned} \tag{2.7}$$

for `semirandom`. This test problem is kept with the same conditions as A.

The source term $f(t,x,y)$ is chosen to be zero because it is easier to work with a constant energy. Some problems with non-constant energy will be presented later.

2.4 Lingo

!!!!!!!!!!!!!!!!!!!!Skriv noe her!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!Forklare forskjellen på u og U?!!!!!!!!!!!!!!!!

There will here be a short explanation of the labels on the figures you will see later.

iterations	number of restarts performed by Arnoldi or symplectic lanzcos method.
time	time elapsed when solving the problem (y-axis)
error1	Difference in error between analytical solution, and estimated solution.
energy2	Difference between the largest and smallest energy during the simulated time.

Table 2.1: Explanation of the labels on the y-axis.

!!!!!!!!!!!!!!!!!!!!Skriv noe her!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!Lag en bedre tabell!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!Fyll inn etterhvert som du trenger det!!!!!!!!!!!!!!!!!!!!

m	number of point in each spacial direction
n	restart variable
k	number of points in time
t	simulated time
restart	a boolean value. If restart == 1, Arnoldi or symplectinc lanzcos method will restart.
convergence criterion	if restart is true, restarting will commence until the change in the solution is less than the convergence criteria

Table 2.2: Explanation of the labels on the x-axis.

Chapter 3

Compare time integration methods

!!!!!!!!!!!!!!!!!!!!!!!!!!Endre y(u eller U) til noe annet!!!!!!!!!!!!!!!!!!!!!

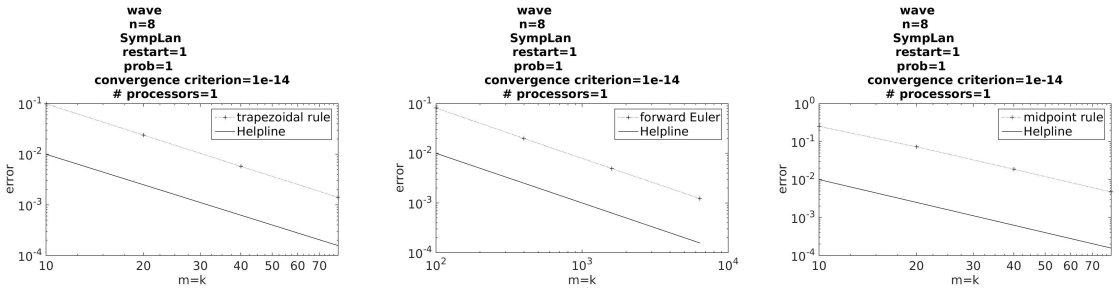
This section will be concerned with comparing the integrators. How well they estimate the error and energy will be the primary concern.

The relevant integrators are trapezoidal rule, forward euler, and midpoint rule. The definition, and the iteration scheme, of the different methods are given in the table below.

Convergence of the different methods are shown in figure 4.1

Trapezoidal rule [1]	$y_{i+1} = y_i + h_t f\left(\frac{1}{2}(t_n + t_{n+1}), \frac{1}{2}(y_n + y_{n+1})\right)$	$y_{i+1} = (I - \frac{Ah_t}{2}) \backslash \left(y_i + \frac{h_t}{2}\left(Ay_i + (F_{i+1} + F_i)\right)\right)$
Forward Euler [2]	$y_{i+1} = y_i + h_t f(t_i, y_i)$	$y_{i+1} = y_i + h_t\left(Ay_i + F_i\right)$
Midpoint rule[3]	$y_{i+2} = y_i + h_t + f\left(t_{i+1}, \frac{1}{2}(y_i + y_{i+2})\right)$	$y_{i+2} = (I - Ah_t) \backslash \left(y_i + 2h_t\left(\frac{Ay_i}{2} + F_{i+1}\right)\right)$

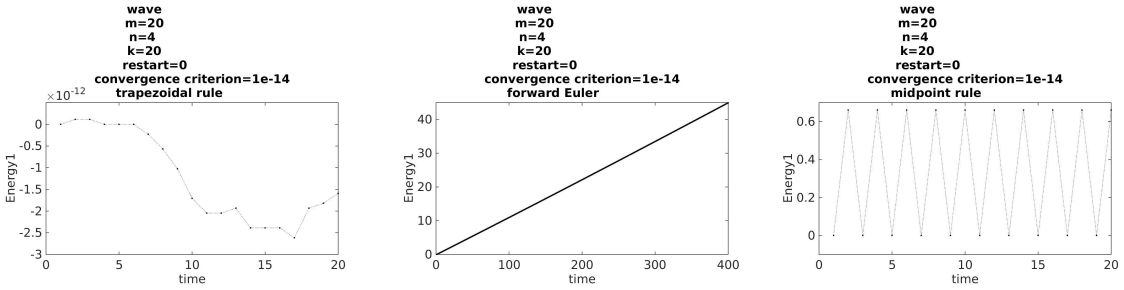
Table 3.1: Proposed methods for integrating in time. A problem that arises with the midpoint rule is the need to know $F_{i+\frac{1}{2}}$, this is solved by doubling the step size. Er denne tabellen fin nok?



(a) Help line decreases with m^2. (b) Help line decreases with m. (c) Help line decreases with m^2.

Figure 3.1: Figure of the convergence for the different integration methods. !!!!!!!Skrive inn det somstår på toppen av figuren her!!!!!

Figure 4.1 shows that all figures converge with the expected rate.



(a) (b) (c)

Figure 3.2: Figure of the difference in energy at a point in time. It is clear that only trapezoidal rule gives a suitable approximation of the error. Forward euler has an linearly increasing energy, while the midpoint rule gives periodic energy. Skrive inn det som står på toppen av figuren her

After looking at figure 3.2 it is easy to conclude that trapezoidal rule outperforms the other two integration methods.

Chapter 4

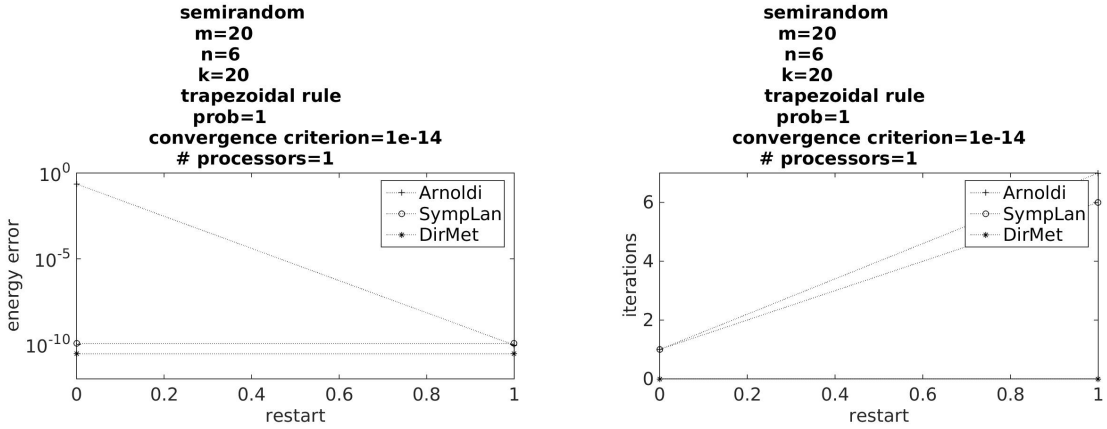
Energy preservation for SLM

It can be proved that a restart of SLM does not alter the energy. This chapter is devoted to showing that this still hold with numerical approximations.

The residual energy of the symplectic Lanzcos method is

$$\frac{1}{2}e_r^\top JAe_r + e_r^\top Jr_{n+1}e_{2n}^\top z \tag{4.1}$$

with $e_r = u - u_n$ (analytical solution minus approximated solution) and r_{n+1} is the residual vector given by the symplectic Lanzcos method.

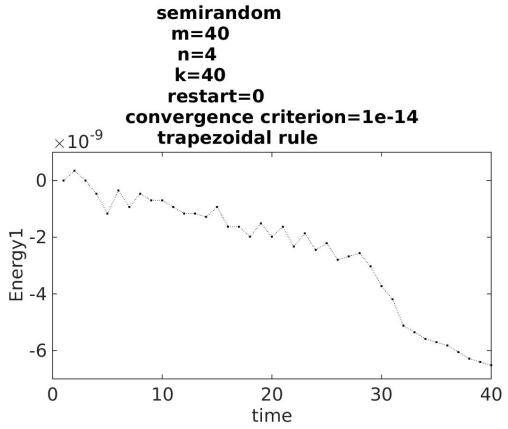


(a) The difference in energy with and without restart.

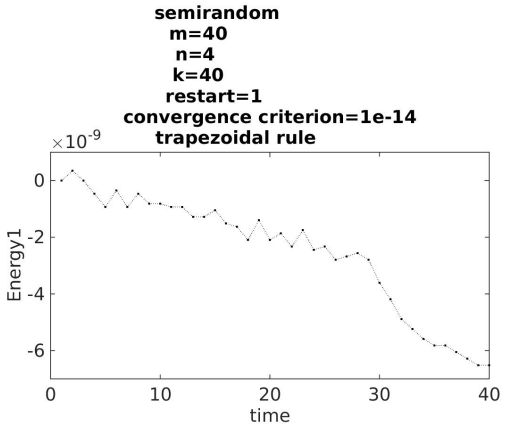
(b) The number of iterations performed with and without restarting.

Figure 4.1: The figure shows how the different methods change the energy with and without restarting.

The figures above implies that restarting the symplectic Lanzcos method does indeed not change the energy. But for Arnoldi's method it changes quite a bit.



(a) Without restart.

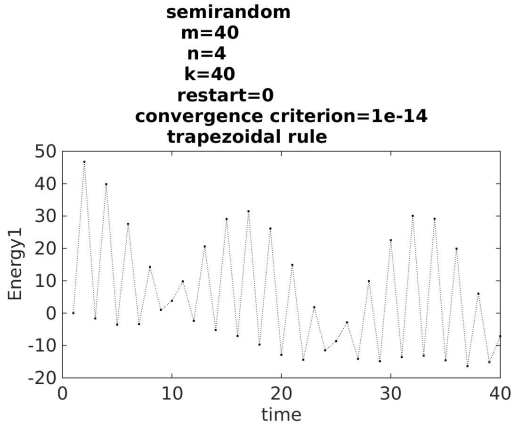


(b) With restart, the number of iterations = 47

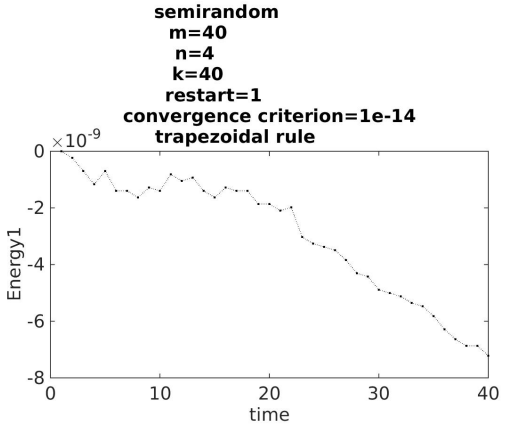
Figure 4.2: The figures shows the change in energy over time. Mer tekst

The figure above shows very little change in energy with SLM, with and without several restarts.

The figures below shows the same as figure 4.2, but with KPM instead of SLM.



(a) Without restart.



(b) With restart, the number of iterations = 48

Figure 4.3: Figures showing the change in energy over time. Mer tekst

When comparing figure 4.2a and 4.2b it becomes clear that SLM's does not alter the energy significantly when performing the restart. Figure 4.3a and 4.3b shows that this is not the case

for KPM.

!!!!!!!!!!!!!!!!!!!!!!ER dette overbevisende nok?!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Chapter 5

Some interesting results

We will now show some numerical experiments using a randomly generated Hamiltonian matrix(`semirandom`).

Bibliography

[1] Endre Süli and David F. Mayers. *An Introduction to Numerical Analysis*, chapter 7.2 Newton–Cotes formulae, pages 202–203. , 2003.

[2] Endre Süli and David F. Mayers. *An Introduction to Numerical Analysis*, chapter 12.2 One-step methods, page 317. , 2003.

[3] Endre Süli and David F. Mayers. *An Introduction to Numerical Analysis*, chapter Definition 10.1, page 286. , 2003.