

1 Introduktion

Vid numerisk lösning av partiella differentialekvationer är det av stor vikt att veta vilken metod som är mest effektiv för ett givet problem. Egenskaper hos metoden i fråga kan exempelvis vara noggrannhetsordning, stabilitetsvillkor samt hur komplicerad den är att implementera. Den tidsberoende Schrödingerekvationen (SE) är i detta sammanhang ytterst intressant då den är en ekvation som ofta behöver lösas. Dessutom har den egenskaper som gör den intressant att lösa med numeriska metoder.

Metoderna som ska undersökas är

1. Leap-Frog-metoden (explicit)
2. Fjärde ordningens Runge-Kutta (explicit)
3. Crank-Nicholson (implicit)

tillsammans med SBP-SAT-metoden (Summation By Parts - Simultaneous Approximation Term) för rumsdiskretisering. För att lösningsmetoden ska vara stabil måste Courant-Friedrichs-Lewy-villkoret (CFL-villkoret) vara uppfyllt. CFL gäller då en explicit metod används. Detta villkor ger den övre gränsen för tidssteget, vilket för SE är att tidssteget k är proportionellt mot rumssteget h som $k \sim h^2$. Detta innebär att tidssteget kommer att vara mycket mindre än rumssteget. Detta kan leda till att noggrannhetsordningen i tiden blir mindre viktig. Detta eftersom tidssteget kanske ändå blir så litet att trunkeringsfelet i tiden ligger under givna toleranser.

De två första metoderna är explicita medan den tredje är implicit, vilket innebär att ett ekvationssystem måste lösas för varje tidssteg. Detta försvårar implementerandet men gör den också ovillkorligt stabil (CFL-villkoret gäller alltså inte i detta fall). Leap-Frog och Crank-Nicholson leder också till energikonservering vilket är av vikt i många sammanhang (t.ex. vid simulering av vågrörelse).

Problemet består alltså i huvudsak i att finna den metod som kan sägas ge en bra avvägning mellan noggrannhet, exekveringshastighet och ett enkelt implementerande.

2 Teori

Den tidsberoende Schrödingerekvationen ges i allmänhet av

$$i\hbar \frac{d}{dt} \Psi(\vec{\mathbf{r}}, t) = \left(-\frac{\hbar^2}{2m} + V(\vec{\mathbf{r}}, t) \right) \Psi(\vec{\mathbf{r}}, t)$$

Vi ska titta på den endimensionella tidsberoende SE

$$u_t = iau_{xx}, \quad 0 \leq x \leq 1, \quad t \geq 0$$

med Dirichletvillkor och en rektangulär potential. För att problemet ska vara stabilt lösbart måste både den kontinuerliga modellen uppfylla villkoren för välställdhet och det diskretiserade problemet vara stabilt.

2.1 Det kontinuerliga problemet

Den fullständiga kontinuerliga modellen ges (på intervallet $0 \leq x \leq 1$) av

$$\begin{cases} u_t = iau_{xx}, & 0 \leq x \leq 1, & t \geq 0, & a \in \Re \\ u = f(x), & 0 \leq x \leq 1, & t = 0 \end{cases} \quad (1)$$

där den rektangulära potentialen är konstant $= 0$ på intervallet $0 \leq x \leq 1$ och oändlig annars. Detta problem har den (normaliserade) analytiska lösningen

$$u(x, t) = \sqrt{2} \sin(\pi x) e^{ia\pi^2 t} \quad (2)$$

Vilket ger ett begynnelsevillkor som ser ut som

$$f(x) = \sqrt{2} \sin(\pi x) \quad (3)$$

Vi kan visa att egenvärdena till detta problem är rent imaginära genom att fouriertransformera den givna PDE i rummet. Låt \hat{P} vara en andra ordningens differentialoperator i rummet och \hat{u} vara fouriertransformen av u i rummet. Detta ger en första ordningens ODE

$$\hat{u}_t = ia\hat{P}\hat{u} = -ia\omega^2\hat{u} \quad (4)$$

För att kunna säkerställa stabilitet måste problemet vara välställt

Definition 1 Ett begynnelse- och randvärdesproblem sägs vara **välställt** om en unik lösning u existerar och om uppskattningen

$$\|u\| \leq e^{bt} \|f(\cdot)\|$$

uppfylls då randdata och ev. tvingande funktioner sätts $= 0$ och konstanten b ej beror av data[1].

Detta kan förenklas till att villkoren för välstäldhet uppfylls om energiuppskattningen för modellen visar på konstant eller avtagande energi. För att öka läsbarheten inför vi en notation för den inre produkten samt motsvarande norm (där en upphöjd $(*)$ betecknar det komplexa konjugatet).

$$(u, w) = \int_0^1 u^* w dx, \quad \|u\|^2 = \int_0^1 u^* u dx$$

Energiuppskattningen görs genom att multiplicera den givna PDE med u^* och integrera den partiellt, samt addera det komplexa konjugatet.

$$\begin{aligned} (u, u_t) &= ia(u, u_{xx}) = ia u^* u_x \Big|_0^1 - ia(u_x, u_x) \\ + (u_t, u) &= -ia(u_{xx}, u) = -ia u_x^* u \Big|_0^1 + ia(u_x, u_x) \end{aligned}$$

Detta ger oss den slutgiltiga energiuppskattningen. Högerledet består av endast randtermer.

$$\frac{d}{dt} \|u\|^2 = ia(u^* u_x - u_x^* u) \Big|_0^1 \leq 0$$

Villkoret att energin är konstant eller avtagande kan uppfyllas genom att sätta det homogena Dirichletvillkoret

$$u = u^* = 0$$

på båda ränderna. Vi kan också uppfylla välställdhetskraven genom att sätta homogena Neumannvillkor på ränderna.

$$u_x = u_x^* = 0$$

2.2 Den diskreta approximationen

Den semi-diskreta formuleringen av SBP-SAT-metoden[2] ser i vårt fall ut som

$$v_t = iaD_2v + SAT$$

där D_2 är SBP-operatoren för approximering av andra ordningens partialderivator i rummet och SAT är termerna som svarar mot den svaga hanteringen av randvillkoren. Med homogena Dirichlet-villkor blir detta

$$v_t = iaH^{-1}(-M - e_1S_1 + e_mS_m)v + \tau_1H^{-1}S_1^T e_1^T v + \tau_mH^{-1}S_m^T e_m^T v$$

där vi har satt två sk. 'penalty'-faktorer, τ_1 och τ_m . Dessa faktorer används för att finjustera vår diskreta approximation för att säkerställa stabilitet. För att genomföra energiuppskattningen multiplicerar vi denna formulering från vänster med v^*H där v^* är det *hermiteska transponatet*. Detta ger efter lite förenkling

$$v^*Hv_t = iav^*(-M - e_1S_1 + e_mS_m)v + \tau_1v^*S_1^T e_1^T v + \tau_mv^*S_m^T e_m^T v \quad (5)$$

Därefter adderas på liknande sätt som för det kontinuerliga fallet det hermiteska transponatet av 5

$$v_t^*H^T v = -iav^*(-M - e_1S_1 + e_mS_m)^*v + \tau_1^*v^*e_1S_1v + \tau_m^*v^*e_mS_mv$$

till ekvation 5, vilket ger oss den slutgiltiga energiuppskattningen för den diskreta approximationen.

$$\frac{d}{dt}\|v\|_H^2 = (\tau_1^* - ia)v^*e_1S_1v + (\tau_m^* + ia)v^*e_mS_mv + (\tau_1 + ia)v^*S_1^T e_1^T v + (\tau_m - ia)v^*S_m^T e_m^T v$$

Vi ser att för att säkerställa stabilitet krävs att penalty-faktorerna uppfyller villkoren

$$\begin{cases} \tau_1 & = -ia \\ \tau_m & = ia \end{cases}$$

Vi väljer $\tau_1 = -ia$ och $\tau_m = ia$ för implementationen av den diskreta approximationen. Detta leder till att uttrycket för den semi-diskreta approximationen blir

$$v_t = iaH^{-1} \underbrace{(-M - e_1 S_1 + e_m S_m + S_1^T e_1^T + S_m^T e_m^T)}_A v \quad (6)$$

För att underlätta notationen väljer vi att kalla matrisen $-M - e_1 S_1 + e_m S_m + S_1^T e_1^T + S_m^T e_m^T$ för A .

2.3 Leap-Frog-metoden

Leap-Frog-metoden (LF) består av centraldifferensen

$$v_t \approx \frac{v^{n+1} - v^{n-1}}{2k}$$

där v är lösningsvektorn. Vi kan visa genom taylorutveckling att denna metod är av noggrannhetsordning $O(k^2)$. Låt oss ställa upp uttrycket för den fullständiga PDE-diskretiseringen med hjälp av ekvation 6

$$\frac{v^{n+1} - v^{n-1}}{2k} = iaH^{-1} A v^n \quad (7)$$

Vi ser av detta att en implementation kräver två initieringsvärden, v^0 och v^1 . Vi tar fram v^1 genom att ta ett steg med en fjärde ordningens Runge-Kutta-metod.

Stabilitetsområdet för LF utgörs i det komplexa talplanet av hela den imaginära axeln. Detta kan visas genom att lösa karakteristiska ekvationen för diskretiseringsschemat ekvation 7. Detta villkor gör LF till synes väl lämpad för att lösa SE, då egenvärdena är rent imaginära enligt ekvation 4. Metoden är också *symplektisk*, dvs. energibevarande.

Noggrannhetsordningen i tiden för LF kan ökas genom att taylorutveckla de ingående tidsskiftade termerna på detta sätt

$$v^{n\pm 1} = v^n \pm kv_t^n + \frac{k^2}{2} v_{tt}^n \pm \frac{k^3}{6} v_{ttt}^n + \frac{k^4}{24} v_{tttt}^n \pm \dots$$

Detta ger en PDE-diskretisering som är noggrann till fjärde ordningen i tiden.

$$\frac{v^{n+1} - v^{n-1}}{2k} = iaH^{-1} A v^n - ia^3 \frac{k^2}{6} A^3 v^n + O(k^4) \quad (8)$$

2.4 Fjärde ordningens Runge-Kutta

Som jämförelse mot övriga metoder används den klassiska Runge-Kutta-metoden, också kallad RK4. Denna metod är noggrann till fjärde ordningen. Till skillnad mot de övriga metoderna (LF och CN) är den inte symplektisk vilket kan göra den olämplig i sammanhang då energikonservering är viktig.

Schemat för RK4 applicerat på ekvation 6 blir

$$\begin{cases} v^{n+1} = v^n + \frac{1}{6}k(c_1 + 2c_2 + 2c_3 + c_4) \\ c_1 = iaH^{-1} A v^n \\ c_2 = iaH^{-1} A(v^n + \frac{1}{2}k c_1) \\ c_3 = iaH^{-1} A(v^n + \frac{1}{2}k c_2) \\ c_4 = iaH^{-1} A(v^n + k c_3) \end{cases} \quad (9)$$

Stabilitetsområdet för RK4 innefattar den imaginära axeln vilket innebär att den klarar av att fånga in egenvärdena för vårt problem givet ett tillräckligt litet tidssteg. På grund av dess explicita karaktär är denna metod bunden av CFL-villkoret.

2.5 Crank-Nicholson-metoden

Schemat för Crank-Nicholson-metoden (CN) ser ut som

$$\frac{v^{n+1} - v^n}{k} = iaH^{-1}A \left(\frac{v^{n+1} + v^n}{2} \right) \quad (10)$$

Vi ser av högerledet att denna metod är implicit. Den är också noggrann till andra ordningen i tiden. Dess implicita natur gör den ovillkorligt stabil (dvs. vi kan ignorera CFL-villkoret). Dock leder det också till att vi tvingas lösa ett ekvationssystem där antalet obekanta är proportionerligt mot antalet beräkningspunkter m i rummet. Detta ekvationssystem ges av

$$\left(I - \frac{1}{2}iaH^{-1}A \right) v^{n+1} = \left(I + \frac{1}{2}iaH^{-1}A \right) v^n$$

där I är $m \times m$ -identitetsmatrisen. För att minska exekveringstiden görs en LU-uppdelning av matrisen i vänsterledet initialt.

$$\frac{v^{n+1} + v^n}{2} = v^{n+\frac{1}{2}} + O(k^2) \quad (11)$$

Noggrannheten kan ökas till fjärde ordningen genom att taylorutveckla v^{n+1} och v^n kring $v^{n+\frac{1}{2}}$ och utnyttja ekvation 11. Detta ger det nya ekvationssystemet

$$\left(I - \frac{1}{2}iaH^{-1}A + \frac{1}{48}ia^3(H^{-1}A)^3 \right) v^{n+1} = \left(I + \frac{1}{2}iaH^{-1}A - \frac{1}{48}ia^3(H^{-1}A)^3 \right) v^n$$

2.6 Energikonservering

Energikonservering är viktigt vid simulering av system där den totala energin är en rörelsekonstant, t.ex. kvantmekaniska system. Vissa metoder för tidsintegrering (tidsstegning) leder till artefakter i energin (minskande, ökande eller oscillerande) över tid på grund av diskretiseringen i tiden. Metodernas energikonserverande (symplektiska) egenskaper undersöktes genom att simulera modellen med $m = 51$ och $k = 1 \cdot 10^{-4}$ under en lång tid, fram till $t = 200$. Energin hos systemet som modelleras beräknas med normen

$$E = v^* H v = \|v\|_h^2$$

2.7 Stabilitet

För att fastställa ett korrekt CFL-villkor för metoderna analyserades egenvärdena hos matrisen $iaH^{-1}A$ för olika tidssteg och rumssteg. Spektralradien ρ (beloppet av det största egenvärdet) hos denna matris är av stor vikt för stabiliteten hos en explicit metod.

m	$k \cdot \rho = 1$ vid k
26	$1,91 \cdot 10^{-4}$
51	$4,796 \cdot 10^{-5}$
101	$1,198 \cdot 10^{-5}$
201	$2,98 \cdot 10^{-6}$
401	$7,3 \cdot 10^{-7}$

Tabell 1: Tidssteg k för vilka spektralradien $\rho = 1$, för olika antal beräkningspunkter m

Då $\rho \leq 1$ kan metoden antas vara stabil. Undersökningen gjordes genom att beräkna spektralradien för matrisen $iaH^{-1}A$ för ett givet antal beräkningspunkter m i rummet och sedan se när kvantiteten $k \cdot \rho$ blir lika med ett. Resultaten visas i tabell 1. Om ett andrafradspolynom anpassas till dessa data fås (om första- och nollteordningenstermerna försummas) CFL-villkoret

$$k \sim 0,1188h^2$$

3 Resultat

3.1 Konvergens (SBP4)

Första uppgiften är att undersöka metodernas konvergensordningar då rumsdiskretiseringen sker med fjärde ordningens SBP-operatorer (SBP4)[2]. Konvergensordningen för en metod beräknas med formeln

$$q = \log_{10} \left(\frac{\|u - v^{(m_1)}\|_h}{\|u - v^{(m_2)}\|_h} \right) / \log_{10} \left(\frac{m_2}{m_1} \right) \quad (12)$$

där m är antalet beräkningspunkter i rummet och $\|u - v^{(m)}\|_h$ är den diskreta l_2 -normen av felet mellan den analytiska lösningen u (se ekvation 2, avsnitt 2.1) och den numeriska lösningen v .

Till att börja med undersöktes andra ordningens LF (LF2), fjärde ordningens Runge-Kutta (RK4) och andra ordningens Crank-Nicholson (CN2). I tabell 2 ser vi de logaritmerade l_2 -felen för ett antal simuleringar fram till $t = 0.2$. Enligt denna tabell är felen för RK4 och CN2 av samma storleksordning.

m	k	$\log l_2^{(LF2)}$	$\log l_2^{(RK4)}$	$\log l_2^{(CN2)}$
26	$1 \cdot 10^{-4}$	-2.7137	-3.0153	-3.0154
51	$4 \cdot 10^{-5}$	-3.4081	-6.6127	-6.5726
101	$1 \cdot 10^{-5}$	-4.0079	-7.7412	-7.7047
201	$2 \cdot 10^{-6}$	-4.7058	-8.9097	-8.8876
401	$4 \cdot 10^{-7}$	-5.4042	-10.1368	-10.099

Tabell 2: $\log(l_2 - fel)$ för LF2, RK4 och CN2 till $t = 0.2$ med SBP4 i rummet

Konvergensordningarna redovisas i tabell 3. LF2 ger ungefär andra ordningens konvergens medan RK4 och CN2 visar på fjärde ordningens konvergens. Dock används endast SBP4 vilket innebär att rumsfelet dominerar för dessa

$(m1, m2)$	$q^{(LF2)}$	$q^{(RK4)}$	$q^{(CN2)}$
(26, 51)	2.3735	12.2946	12.1575
(51, 101)	2.0211	3.8028	3.8148
(101, 201)	2.3349	3.9096	3.9579
(201, 401)	2.3284	4.0913	4.0388

Tabell 3: Konvergensordningar för LF2, RK4 och CN2 till $t = 0.2$ med SBP4 i rummet

m	k	$t_{(LF2)}$	$t_{(RK4)}$	$t_{(CN2)}$
26	$1 \cdot 10^{-4}$	0.019313	0.059713	0.03213
51	$4 \cdot 10^{-5}$	0.064032	0.19291	0.17308
101	$1 \cdot 10^{-5}$	0.52256	1.8552	2.5082
201	$2 \cdot 10^{-6}$	5.4896	20.3182	37.6456
401	$4 \cdot 10^{-7}$	108.0041	405.1416	798.3771

Tabell 4: Exekveringstider i sekunder för LF2, RK4 och CN2 till $t = 0.2$ med SBP4 i rummet

simuleringar och vi kan därmed inte förvänta oss högre konvergens än fjärde ordningen i detta fall.

Exekveringstiderna för metoderna mättes i sekunder med hjälp av kommandot *tic toc* i MATLAB. Tiderna för simuleringarna i tabell 2 visas i tabell 4. Vi ser att LF2 är överlägset snabbast för alla nyttjade tids- och rumssteg. För RK4 och CN2 ser tiderna ut att vara nära varandra för stora tids- och rumssteg medan tiden för CN2 växer till nära 2x tiden för RK4 vid det minsta tids- och rumsstegen.

Fjärde ordningens LF (LF4) borde innebära en förbättring i noggrannhet över LF2. Resultaten från simuleringar med samma tids- och rumssteg som för de i tabell 2 visas i tabell 5. Vi ser att LF4 inte ger någon skillnad från LF2. Detta kan bero på fel i implementeringen eller en felaktig härledning av metoden (se ekvation 8, avsnitt 2.3).

Resultaten från simuleringarna med fjärde ordningens Crank-Nicholson (CN4) visas i tabell 6. Vi ser av detta att skillnaden mellan C2 och CN4 är liten. På grund av att dessa metoder är ovillkorligt stabila valdes i detta fall tidssteget som $k \sim h$ vilket ger större tidssfel, men snabbare exekvering. Av denna anledning blir också konvergensen av andra ordningen, vilket vi ser i tabell 7.

Vi ser också att i vissa fall visar CN4 på ett större fel än CN2, vilket är märkvärdigt. Detta kan (som för LF4) bero på fel i implementeringen eller en felaktig härledning av metoden (se avsnitt 2.5).

m	k	$\log l_2^{(LF4)}$	$\log l_2^{(LF2)}$
26	$1 \cdot 10^{-4}$	-2.7137	-2.7137
51	$4 \cdot 10^{-5}$	-3.4039	-3.4081
101	$1 \cdot 10^{-5}$	-4.0079	-4.0079
201	$2 \cdot 10^{-6}$	-4.7058	-4.7058
401	$4 \cdot 10^{-7}$	-5.4042	-5.4042

Tabell 5: $\log(l_2 - fel)$ för LF4 och LF2 till $t = 0.2$ med SBP4 i rummet

m	k	$\log l_2^{(CN2)}$	$\log l_2^{(CN4)}$
26	0.04	-0.91091	-0.73734
51	0.02	-1.5011	-1.3255
101	0.01	-2.0991	-1.9231
201	0.005	-2.6996	-2.5235
401	0.0025	-1.6172	-1.6217
801	0.0013	-2.012	-2.0142

Tabell 6: $\log(l_2 - fel)$ för CN2 och CN4 till $t = 0.2$ med SBP4 i rummet

$(m1, m2)$	$q^{(CN2)}$	$q^{(CN4)}$
(26, 51)	2.0152	2.0137
(51, 101)	2.0152	2.0137
(101, 201)	2.0092	2.0089
(201, 401)	-3.6084	-3.0066
(401, 801)	1.3137	1.3061

Tabell 7: Konvergensordningar CN2 och CN4 till $t = 0.2$ med SBP4 i rummet

Avsaknaden av skillnad mellan felen för CN2 och CN4 visar sig även i konvergensordningarna.

3.2 Konvergens (SBP6)

För sjätte ordningens SBP-operatorer (SBP6)[2] ser vi att felen blir betydligt mindre för RK4 och CN2 enligt resultaten i tabell 8. LF krävde något kortare tidssteg för att uppnå stabilitet med dessa operatorer. Därför presenteras resultaten med LF separat från RK4 och CN2. Vi ser av resultaten i tabell 9 att LF2 inte uppvisar någon skillnad mellan SBP4 och SBP6.

I tabell 10 ser vi konvergensordningarna för CN2 och RK4 med SBP6-operatorerna (konvergensordningarna beräknade med formel 12, avsnitt 3.1). RK4 ser ut att ge femte och sjätte ordningens konvergens medan CN2 ger strax under femte ordningens konvergens.

Metoderna uppvisar liknande beteende för $m = 26$ och $m = 51$ som för SBP4 (se tabell 3, avsnitt 3.1) i tabell 10. Detta fenomen kan bero på att tidsintervallets längd inte är jämn delbart med tidssteget vilket leder till små skillnader i simuleringens sluttid. För $m = 201$ och $m = 401$ tappas CN2 nästan fyra konvergensordningar, vilket kan också bero på samma beteende. Liknande beteende observeras även i avsnitt 3.4 (se tabell 15, 16 och 17).

m	k	$\log l_2^{(RK4)}$	$\log l_2^{(CN2)}$
26	$1 \cdot 10^{-4}$	-3.0143	-3.0144
51	$4 \cdot 10^{-5}$	-7.4755	-7.2284
101	$1 \cdot 10^{-5}$	-9.1454	-8.7021
201	$2 \cdot 10^{-6}$	-10.8042	-10.1364
401	$4 \cdot 10^{-7}$	-12.4029	-10.3891

Tabell 8: $\log(l_2 - fel)$ för RK4 och CN2 till $t = 0.2$ med SBP6 i rummet

m	k	$\log l_2^{(LF2)}$	$\log l_2^{(LF2\ SBP6)}$
26	$1 \cdot 10^{-4}$	-2.7137	-2.7132
51	$2 \cdot 10^{-5}$	-3.4039	-3.4036
101	$6 \cdot 10^{-6}$	-4.0079	-4.0079
201	$1 \cdot 10^{-6}$	-5.0068	-5.0068
401	$4 \cdot 10^{-7}$	-5.4042	-5.4042

Tabell 9: $\log(l_2 - fel)$ för LF2 till $t = 0.2$ med SBP4 och SBP6 i rummet

$(m1, m2)$	$q^{(RK4)}$	$q^{(CN2)}$
(26, 51)	15.2469	14.4022
(51, 101)	5.6273	4.9659
(101, 201)	5.5501	4.799
(201, 401)	5.3299	0.84258

Tabell 10: Konvergensordningar för RK4 och CN2 till $t = 0.2$ med SBP6 i rummet

Exekveringstiderna (i sekunder) för CN2 och RK4 med SBP6-operatorerna visas i tabell 11

m	k	$t_{(RK4)}$	$t_{(CN2)}$
26	$1 \cdot 10^{-4}$	0.078885	0.038749
51	$4 \cdot 10^{-5}$	0.19133	0.19506
101	$1 \cdot 10^{-5}$	1.9368	2.4611
201	$2 \cdot 10^{-6}$	22.176	39.1553
401	$4 \cdot 10^{-7}$	518.2134	866.4234

Tabell 11: Exekveringstider i sekunder för RK4 och CN2 till $t = 0.2$ med SBP6 i rummet

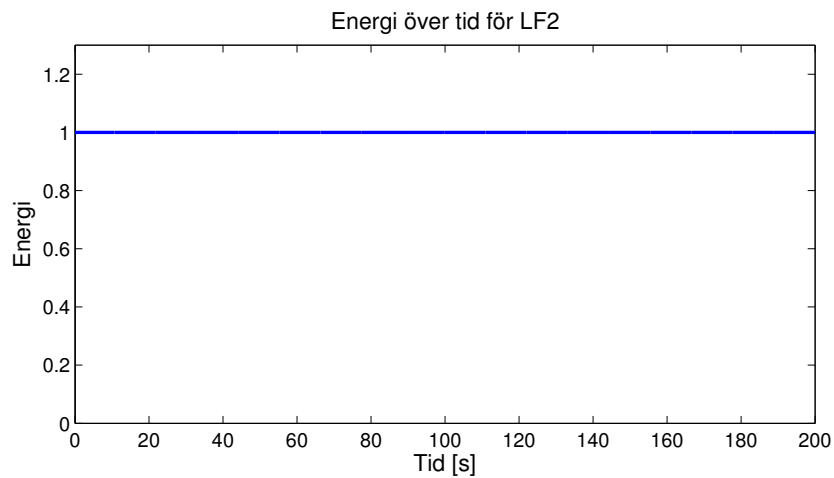
Metoden CN4 undersöktes också med SBP6-operatorerna. På grund av att dessa metoder är ovillkorligt stabila valdes även i detta fall tidssteget som $k \sim h$ vilket ger större tidssfel, men snabbare exekvering. Det ses i tabell 12 att skillnaden mellan CN4 och CN4 med SBP6 är försumbar.

m	k	$\log l_2^{(CN2)}$	$\log l_2^{(CN4)}$
26	0.04	-0.91095	-0.73737
51	0.02	-1.5011	-1.3256
101	0.01	-2.0991	-1.9231
201	0.005	-2.6996	-2.5235
401	0.0025	-1.6172	-1.6218
801	0.0013	-2.012	-2.0221

Tabell 12: $\log(l_2 - fel)$ för CN2 och CN4 till $t = 0.2$ med SBP6 i rummet

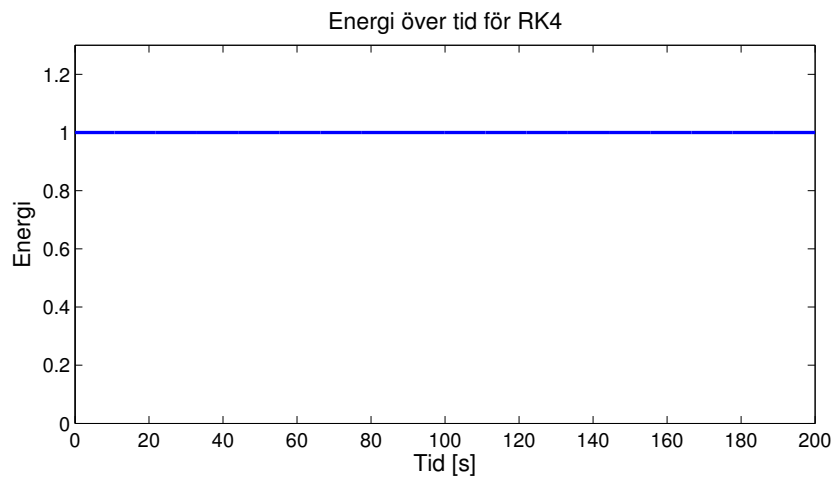
3.3 Energikonservering

Metodernas energikonserverande (symplektiska) egenskaper undersöktes genom att simulera modellen med $m = 51$ och $k = 1 \cdot 10^{-4}$ under en lång tid, fram till



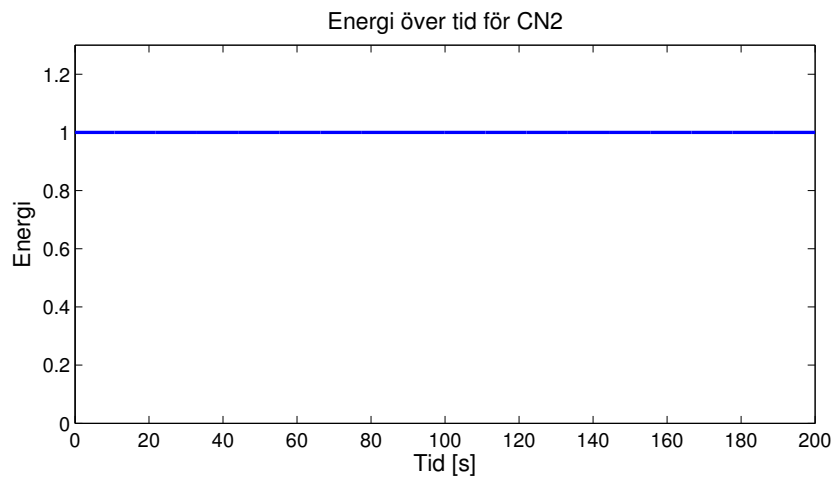
Figur 1: Energi över tid till $t = 200$ för LF2

$t = 200$. Vi ser i figur 1 att energin håller sig på en konstant nivå då modellen simuleras med LF2. Samma sak ser vi för RK4 i figur 2 och för CN2 i figur 3. Som det påpekas i avsnitt 2.4 bör RK4 uppvisa ett icke-energikonserverande beteende, men av figur 2 kan inget sådant beteende ses.

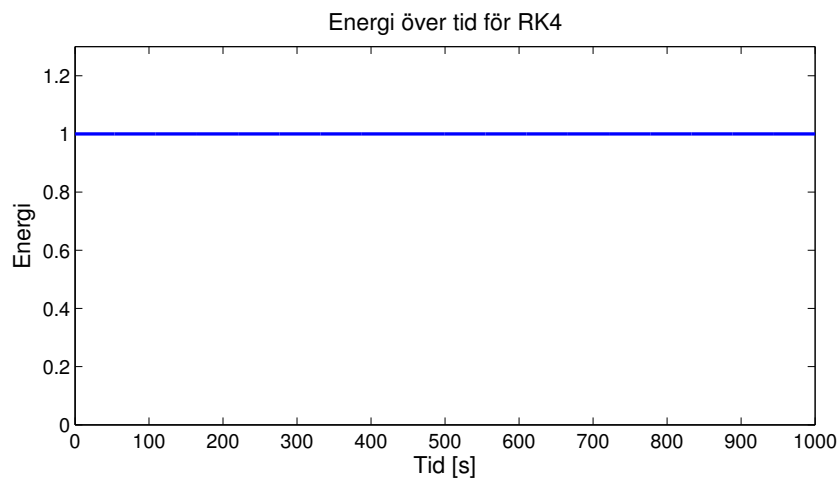


Figur 2: Energi över tid till $t = 200$ för RK4

För att undersöka detta vidare gjordes en simulering fram till $t = 1000$. Resultatet från denna simulering visas i figur 4 och vi ser att även för denna långa simuleringstid uppvisar RK4 energikonservering.



Figur 3: Energi över tid till $t = 200$ för CN2



Figur 4: Energi över tid till $t = 1000$ för RK4

3.4 Effektivitet

En effektiv metod definieras i denna rapport som den metod som snabbast uppnår en given tolerans. Vi ska undersöka exekveringstiderna för att uppnå toleranserna 10^{-4} , 10^{-7} och 10^{-10} . För att underlätta arbetet att finna ett gott val av tids- och rumssteg undersöktes rumsfelet. Detta gjordes genom att hålla rumssteget konstant och successivt minska tidssteget till dess att tidsfelet blivit försumbart. Observera att, för att slippa göra fler simuleringar än nödvändigt, användes inte LF2 för att undersöka rumsfelet (men dess effektivitet analyserades ändå).

Vi ser av tabell 13 att felet för $m = 16$ lägger sig till rätta vid ca -4.3 (vid tidssteget $k = 1 \cdot 10^{-6}$), vilket kan antas bestå till största del av rumsfelet. För

k	$\log l_2^{(RK4)}$	$\log l_2^{(CN2)}$
$1 \cdot 10^{-3}$	-3.0328	-3.0335
$1 \cdot 10^{-5}$	-4.1202	-4.1193
$1 \cdot 10^{-6}$	-4.3295	-4.3295
$1 \cdot 10^{-7}$	-4.335	-4.335

Tabell 13: $\log(l_2 - fel)$ för RK4 och CN2 vid $m = 16$ till $t = 1$ med SBP6 i rummet

k	$\log l_2^{(RK4)}$	$\log l_2^{(CN2)}$
$1 \cdot 10^{-4}$	-3.0147	-3.0151
$1 \cdot 10^{-5}$	-4.0192	-4.0194
$1 \cdot 10^{-6}$	-5.5767	-5.5732
$4 \cdot 10^{-7}$	-5.5767	-5.5761
$1 \cdot 10^{-7}$	-5.6194	-5.6193
$5 \cdot 10^{-8}$	-5.5767	-5.5767
$1 \cdot 10^{-8}$	-5.5767	-5.5767

Tabell 14: $\log(l_2 - fel)$ för RK4 och CN2 vid $m = 26$ till $t = 1$ med SBP6 i rummet

$m = 26$ ser vi i tabell 14 att rumsfelet är ungefär -5.5 . Det minsta tidssteget för vilket detta uppnås är $k = 1 \cdot 10^{-6}$. Observera att tidsstegen valts så att en tydlig trend mot ett visst fel framträder, vilket är det intressanta. Att de valda tidsstegen är olika för olika m har ingen betydelse.

I tabell 15 visas felen för $m = 51$. I detta fall är det svårt att se någon trend, dvs. felet minskar inte tydligt med minskande tidssteg. Detta kan bero på fenomenet som observeras i avsnitt 3.1 (se t.ex. tabell 3). Det som kan ses är dock att det minsta felet som uppnås är ca -7.3 , vilket som tidigare kan antas bestå till största del av rumsfelet.

Vi ser i tabell 16 att dessa resultat uppvisar samma beteende som för $m = 51$. Det minsta felet som uppnås är ca -9.1 för RK4 och ca -8.9 för CN2. Detta uppnås då då tidssteget är som störst $k = 1 \cdot 10^{-6}$, samma som för $m = 51$ och $m = 26$.

Slutligen undersöktes $m = 201$ vid ett antal tidssteg. Resultaten från dessa simuleringar ses i tabell 17. Vi ser i denna tabell att minsta felet för Rk4 är -11

k	$\log l_2^{(RK4)}$	$\log l_2^{(CN2)}$
$1 \cdot 10^{-4}$	-3.0147	-3.0151
$1 \cdot 10^{-5}$	-4.01	-4.0101
$1 \cdot 10^{-6}$	-7.3212	-7.2625
$1 \cdot 10^{-7}$	-6.0142	-6.0142
$8 \cdot 10^{-7}$	-7.3212	-7.2469
$5 \cdot 10^{-7}$	-5.312	-5.312
$4 \cdot 10^{-7}$	-7.3213	-7.2673
$5 \cdot 10^{-8}$	-5.5767	-5.5767

Tabell 15: $\log(l_2 - fel)$ för RK4 och CN2 vid $m = 51$ till $t = 1$ med SBP6 i rummet

k	$\log l_2^{(RK4)}$	$\log l_2^{(CN2)}$
$1 \cdot 10^{-4}$	-3.0147	-3.0151
$1 \cdot 10^{-5}$	-4.01	-4.0101
$2 \cdot 10^{-6}$	-4.7068	-4.7068
$1 \cdot 10^{-6}$	-9.1874	-8.9038
$8 \cdot 10^{-7}$	-9.1491	-8.9232
$5 \cdot 10^{-7}$	-5.3089	-5.3089
$4 \cdot 10^{-7}$	-9.122	-8.9787
$3 \cdot 10^{-7}$	-5.7069	-5.7069
$2 \cdot 10^{-7}$	-9.1203	-8.938
$1 \cdot 10^{-7}$	-6.0079	-6.0079
$9 \cdot 10^{-8}$	-6.1048	-6.1048
$5 \cdot 10^{-8}$	-5.5767	-5.5767
$1 \cdot 10^{-8}$	-5.5767	-5.5767

Tabell 16: $\log(l_2 - fel)$ för RK4 och CN2 vid $m = 101$ till $t = 1$ med SBP6 i rummet

k	$\log l_2^{(RK4)}$	$\log l_2^{(CN2)}$
$2 \cdot 10^{-6}$	-4.7058	-4.7058
$1 \cdot 10^{-6}$	-10.7957	-10.0688
$8 \cdot 10^{-7}$	-10.7948	-10.1681
$4 \cdot 10^{-7}$	-10.7412	-9.4739
$2 \cdot 10^{-7}$	-11.0397	-9.6903
$1 \cdot 10^{-7}$	-6.0068	-6.0068

Tabell 17: $\log(l_2 - fel)$ för RK4 och CN2 vid $m = 201$ till $t = 1$ med SBP6 i rummet

vid $k = 2 \cdot 10^{-7}$ och ca -10.1 vid $k = 8 \cdot 10^{-7}$ för CN2. Av dessa studier kan slutsatser dras om vad som är effektiva tidssteg för att uppnå givna toleranser.

I tabell 18 ses den minsta exekveringstiden tillsammans med tids- och rumsstegen som krävdes för att uppnå toleranserna $1 \cdot 10^{-4}$, $1 \cdot 10^{-7}$ och $1 \cdot 10^{-10}$ vid tiden $t = 1$. Metoderna som användes var LF2, RK4 och CN2. För toleransen $1 \cdot 10^{-10}$ utslöts LF2 på grund av tidsåtgången. Eftersom denna metod inte konvergerar lika snabbt som CN2 och RK4 (se tabell 3, avsnitt 3.1) krävdes ett betydligt mindre tidssteg för att uppnå givna toleranser.

tol	$t^{(LF2)}$	$t^{(RK4)}$	$t^{(CN2)}$
$1 \cdot 10^{-4}$	0.6573 (14, $8 \cdot 10^{-6}$)	0.6110 (14, $4 \cdot 10^{-5}$)	0.2763 (14, $4 \cdot 10^{-5}$)
$1 \cdot 10^{-7}$	892.7 (51, $1 \cdot 10^{-8}$)	0.7883 (42, $4 \cdot 10^{-5}$)	1.3700 (46, $2 \cdot 10^{-5}$)
$1 \cdot 10^{-10}$	-	141.7 (141, $1 \cdot 10^{-6}$)	261.1 (161, $1 \cdot 10^{-6}$)

Tabell 18: Exekveringstid i sekunder vid (m, k) för att uppnå toleranserna $1 \cdot 10^{-4}$, $1 \cdot 10^{-7}$ och $1 \cdot 10^{-10}$ vid $t = 1$ med metoderna LF2, RK4 och CN2.

4 Slutsatser

Leapfrog-metoden (LF2) presterar betydligt sämre vid lösning av Schrödingerekvationen än både fjärde ordningens Runge-Kutta (RK4) och Crank-Nicholson (CN2). Den gör detta trots till synes goda teoretiska förutsättningar, ett enkelt implementerande och snabb stegning. På grund av sin långsamma konvergensthastighet kräver den för korta tidssteg för att vara användbar i detta fall. Av resultaten ser vi att RK4 ger den snabbaste lösningen för toleranserna $1 \cdot 10^{-7}$ och $1 \cdot 10^{-10}$ medan CN2 verkar ha ett övertag för den högre toleransen $1 \cdot 10^{-4}$ trots sin implicita karaktär. Under arbetets gång undersöktes också fjärde ordningens korrekationer för Leapfrog och Crank-Nicholson (LF4 och CN4). Dessa gav dock inte förväntad konvergens. Samtliga metoder (LF2, RK4 och CN2) uppvisade energikonservering för tidsskalorna 200s trots att RK4 inte anses vara symplektisk. Energikonservering undersöktes därför för RK4 vid 1000s. RK4 uppvisade dock samma beteende som för 200s.

Eventuella framtida undersökningar kan fokusera på att jämföra RK4 med andra populära metoder för att lösa Schrödingerekvationen t.ex. exponentialintegratorer. Dessa undersökningar kan också studera andra randvillkor än Dirichletvillkor. Andra undersökningar av intresse är att försöka repetera resultaten för LF4 och CN4 som erhöles i denna rapport. I denna rapport observerades även ett beteende hos tidsfelet som författaren inte kunnat förklara, vilket också är något som framtida undersökningar bör vara uppmärksam på och titta närmare på om möjligt.

Referenser

- [1] Gustafsson, Bertil *High Order Difference Methods for Time Dependent PDE* Springer 2008
- [2] Mattsson, Ken *Erratum to: Summation by Parts Operators for Finite Difference Approximations of Second-Derivatives with Variable Coefficients*, Journal of Scientific Computing, vol. 51, nr. 3, s.754–754, 2012, Springer.