UPPSALA UNIVERSITET

Numerical simulation of the nonlinear Korteweg-de Vries equation using SBP in time

Christoffer Zakrisson, Erik Hedlund

Project in Computational Science: Report

January 31, 2015

PROJECT REPORT

Department of Information Technology

Abstract

We implement a novel method for time integration of PDE:s, using summation-by-parts (SBP) finite difference operators in time and space. We show that the method is highly accurate as well as fast, especially for stiff problems. Using SBP also means proving stability is straight forward. The method requires large amounts of memory, however, even when using sparse data structures. This problem can be somewhat mitigated however by using the 'multi block formulation' (or 'restart method') outlined in Nordström and Lundquist [2].

Contents

1 Introduction

The dominating method for solving initial boundary value problems (IBVP) is to use local methods, which work by calculating the solution for new time levels from previously calculated time levels. Global methods, conversely, provide a means to obtain the solution at all time levels of interest at once.

Finite difference methods, based on the summation-by-parts and simultaneous-approximation-term (SBP-SAT) technique offer a robust, well-proven approach to discretising IBVP (SBP) [4] and for weakly imposing boundary conditions (SAT). These methods have traditionally been used to discretise the spatial dependencies. Recently, however, two papers [1] [2] by Nordström and Lundquist explored the use of SBP operators to discretise IBVP in time, thereby creating a global method. This method is referred to in this paper as 'SBP in time'.

Results by Lundquist and Nordström show high orders of accuracy and unconditional stability for this method. The unconditional stability of this method makes it a good candidate for time integration of stiff problems (which forces the use of small time steps) since a larger time step can be used for these problems compared to an explicit, local method.

This paper will apply SBP in time to the Korteweg de-Vries (KdV) equation, which is a stiff, nonlinear wave equation that is of interest when studying ocean waves (among other kinds). The soliton solution to this equation is particularly intriguing since it corresponds to a tsunami wave. We will compare SBP in time against a specially derived implicit (hence, unconditionally stable) fourth order Crank-Nicholson method (CN4) and the traditional explicit fourth order Runge-Kutta method (RK4).

2 Theory

What makes the Korteweg-de Vries equation an interesting problem for this study are the difficulties arising when attempting to solve it. The stiffness and non-linearity put special demands on the solution method and make it a good problem for testing different solution methods.

2.1 Linearisation

The KdV equations is given by

$$u_t = u_{xxx} + 6uu_x, \tag{1}$$

where the solition solution

$$u(x,t) = \frac{1}{2}c\,\mathrm{sech}^2\left(\frac{\sqrt{c}}{2}(x - ct - a)\right) \tag{2}$$

is an analytical solution to the KdV equation. Due to the need to limit the scope of this paper a linearised version of (1) is introduced as

$$u_t + \alpha u_x + \beta u_{xxx} = F(x,t), \tag{3}$$

where $F(x,t)$ is a forcing function, which ensures that (3) has the same solution as (1).

2.2 Well-posedness

Proving the stability of the SBP-SAT method is straight forward for well-posed problems. Wellposedness can be proved for the continous problem by using the energy method, which consists of showing that the l_2-norm of the solution is bounded. The continuous problem (without boundary conditions) is given by the following set of equations

$$\begin{cases} u_t + \alpha u_x + \beta u_{xxx} = F(x,t), & L \leq x \leq R, \quad t \geq 0, \\ u = f(x), & L \leq x \leq R, \quad t = 0. \end{cases} \tag{4}$$

Before analysing the well-posedness of the problem we define an inner product and a corresponding norm

$$(u,v) = \int_L^R u^* v \, dx, \quad (u,u) = \int_L^R u^* u \, dx = \|u\|^2 . \tag{5}$$

The continuous energy estimate is obtained by multiplying equation 4 by u, integrating by parts and adding the transpose. Since we are only interested in well-posedness we can disregard the forcing function in the following analysis. Doing this gives us

$$\begin{aligned}(u,u_t) + (u_t,u) = -\beta \left(2uu_{xx} - u_x^2\right)\Big|_L^R - \underbrace{\beta(u_{xx},u_x) + \beta(u_{xx},u_x)}_{=0} \\ - \alpha \, u^2\Big|_L^R + \underbrace{\alpha(u_x,u) - \alpha(u_x,u)}_{=0} .\end{aligned} \tag{6}$$

We see that only the boundary terms remain after integrating. This gives the following estimate for the norm of the solution in time.

$$\frac{d}{dt}\|u\|^2 = \left[\beta u_x^2 - 2\beta u u_{xx} - \alpha u^2\right]_L^R \tag{7}$$

The remaining boundary terms can be regarded as a quadratic form, and can therefore be written as a vector-matrix-vector product

$$\frac{d}{dt}\|u\|^2 = \begin{bmatrix} u \\ u_x \\ u_{xx} \end{bmatrix}^* \underbrace{\begin{bmatrix} -\alpha & 0 & -\beta \\ 0 & \beta & 0 \\ -\beta & 0 & 0 \end{bmatrix}}_{\text{B}} \begin{bmatrix} u \\ u_x \\ u_{xx} \end{bmatrix} \Bigg|_L^R . \tag{8}$$

By examining the eigenvalues of the matrix B we can determine the number of boundary conditions to be specified at each boundary. The eigenvalues are given by

$$\lambda_1 = \beta, \quad \lambda_2 = \frac{1}{2}\left(-\alpha - \sqrt{\alpha^2 + 4\beta^2}\right), \quad \lambda_3 = \frac{1}{2}\left(-\alpha + \sqrt{\alpha^2 + 4\beta^2}\right) \tag{9}$$

where we assume $\alpha, \beta \neq 0$. We see that we have $\lambda_2 < 0, \quad \lambda_3 > 0$ (since $\alpha^2 + 4\beta^2 > 0$ for all $\alpha, \beta \in \mathbb{R}$) and the sign of $\lambda_1 = \beta$ is determined by the sign of β. In order to ensure well-posedness we need at least one boundary condition per boundary. We also need to specify one extra condition on one of the boundaries. The side for which we need to specify an extra boundary condition depends on the sign of β.

Choosing $\alpha = -\frac{3}{2}c,\ \beta = -\frac{1}{2}$ (solely because of easier computations) gives

$$\lambda_1 = -\frac{1}{2}, \quad \lambda_2 = \frac{1}{2}\left(\frac{3}{2}c - \sqrt{\frac{9}{4}c^2 + 1}\right), \quad \lambda_3 = \frac{1}{2}\left(\frac{3}{2}c + \sqrt{\frac{9}{4}c^2 + 1}\right). \tag{10}$$

Therefore, we have two negative eigenvalues and one positive eigenvalue, which implies that we need one boundary condition at $x = R$ and two at $x = L$. From (7) we see that we can simply set

$$u|^R = 0, \quad u_{xx}|^L = u_x|^L = 0 \tag{11}$$

for well-posedness. A set of strongly well-posed boundary conditions are

$$\begin{aligned} u_x|^L &= g_l(t), \quad \gamma_l u + \delta_l u_{xx}|^L = \tilde{g}_l(t), \\ \gamma_r u + \delta_r u_{xx}|^R &= g_r(t), \end{aligned} \tag{12}$$

where

$$\gamma_l = -\frac{\lambda_2\sigma_2}{\beta}, \quad \delta_l = \sigma_2, \quad \gamma_r = -\frac{\lambda_3\sigma_3}{\beta}, \quad \delta_r = \sigma_3,$$

where the normalisation factors σ_2 and σ_3 (obtained from orthogonalising B by $B = T^T\Lambda T$) are denoted by

$$\sigma_i = \frac{1}{\sqrt{1 + \left(\frac{\lambda_i}{\beta}\right)^2}}.$$

2.3 Semi-discretisation

The spatial discretisation is performed by applying the SBP-SAT method in space and stability is proved by the energy method. We proceed with the spatial discretisation of the problem in section 2.2 by dividing the interval $L \leq x \leq R$ into m points, with a spatial step size of $h = \frac{R-L}{m-1}$. First, some definitions of the SBP operators used (taken from K.Mattsons article [3]).

Definition 1. *Let $D_1 = H^{-1}\left(Q - \frac{1}{2}e_1e_1^T + \frac{1}{2}e_me_m^T\right)$ be a first order, summation-by-parts finite difference operator where $H = H^T \geq 0$, $Q + Q^T = 0$ and*

$$\begin{aligned} e_1 &= \begin{bmatrix}1 & 0 & 0 & \dots & 0\end{bmatrix}^T, \\ e_m &= \begin{bmatrix}0 & \dots & 0 & 0 & 1\end{bmatrix}^T. \end{aligned}$$

Definition 2. *Let $D_3 = H^{-1}\left(R + \frac{1}{2}d_{1,1}^Td_{1,1} - \frac{1}{2}d_{1,m}^Td_{1,m} - e_1d_{2,1} + e_md_{2,m}\right)$ be a third order, summation-by-parts finite difference operator where $d_{1,1}$, $d_{1,m}$, $d_{2,1}$, $d_{2,m}$ are vectors that approximate the boundary derivatives*

$$\begin{aligned} d_{1,1}v &\approx u_x|^L, \quad d_{1,m}v \approx u_x|^R, \\ d_{2,1}v &\approx u_{xx}|^L, \quad d_{2,m}v \approx u_{xx}|^R. \end{aligned}$$

Applying the operators in definitions 1 and 2 to the problem defined in equation (4) gives us the formulation

$$v_t = -\alpha D_1 v - \beta D_3 v + SAT \tag{13}$$

$$v_j(t=0) = f(x_j), \quad x_j = L + jh, \quad j = 0, \ldots, m-1 \tag{14}$$

where SAT denotes the simultaneous approximation terms used to weakly imposed the boundary conditions. Continuing the analysis we first use the boundary conditions in equation (11). This gives us the terms

$$\begin{aligned} SAT \ &= \tau_{1,1} H^{-1} d_{1,1}^T \left(d_{1,1} v - g_l \right) \\ &+ \tau_{1,2} H^{-1} e_1 \left(\gamma_l e_1^T v + \delta_l d_{2,1} v - \tilde{g}_l \right) \\ &+ \tau_{2,1} H^{-1} e_m \left(\gamma_r e_m^T v + \delta_r d_{2,m} v - g_r \right). \end{aligned} \tag{15}$$

Combining equations (13) and (15) yields

$$\begin{aligned} v_t = &- \beta D_3 v - \alpha D_1 v \\ &+ \tau_{1,1} H^{-1} d_{1,1}^T \left(d_{1,1} v - g_l \right) \\ &+ \tau_{1,2} H^{-1} e_1 \left(\gamma_l e_1^T v + \delta_l d_{2,1} v - \tilde{g}_l \right) \\ &+ \tau_{2,1} H^{-1} e_m \left(\gamma_r e_m^T v + \delta_r d_{2,m} v - g_r \right). \end{aligned} \tag{16}$$

Before continuing, we note that the matrix H defines an inner product $v^T H w$ and a corresponding norm $\|v\|_H^2 = v^T H v$ analogous to equation (5). Therefore, to obtain the semi-discrete energy estimate, we multiply (16) by $v^T H$ and add the transpose, which gives us

$$\begin{aligned} v^T H v_t + v_t^T H v = &- \alpha v^T \left(Q + Q^T - e_1 e_1^T + e_m e_m^T \right) v \\ &- \beta v^T \left(R + R^T + d_{1,1}^T d_{1,1} - d_{1,m}^T d_{1,m} \right. \\ &\left. - e_1 d_{2,1} - d_{2,1}^T e_1^T + e_m d_{2,m} + d_{2,m}^T e_m^T \right) v \\ &+ \tau_{1,1} v^T \left(d_{1,1}^T d_{1,1} + d_{1,1} d_{1,1}^T \right) v \\ &+ \tau_{1,2} v^T \left(\gamma_l (e_1 e_1^T + e_1^T e_1) + \delta_l (e_1 d_{2,1} + d_{2,1}^T e_1^T) \right) v \\ &+ \tau_{2,1} v^T \left(\gamma_r (e_m e_m^T + e_m^T e_m) + \delta_r (e_m d_{2,m} + d_{2,m}^T e_m^T) \right) v \\ &- 2\tau_{1,1} d_{1,1} v g_l - 2\tau_{1,2} v_1 \tilde{g}_l - 2\tau_{2,1} v_m g_r. \end{aligned} \tag{17}$$

This results in the energy estimate

$$
\begin{aligned}
\frac{d}{dt}\|v\|_H^2 =& \beta\left(d_{1,m}v\right)^2 + 2\left(\beta+\tau_{1,2}\delta_l\right)v_1 d_{2,1}v + 2\left(-\beta+\tau_{2,1}\delta_r\right)v_m d_{2,m}v \\
&+ \left(2\tau_{1,1}-\beta\right)\left(d_{1,1}v\right)^2 - 2\tau_{1,1}d_{1,1}vg_l \\
&+ \left(2\tau_{1,2}\gamma_l+\alpha\right)v_1^2 - 2\tau_{1,2}v_1\tilde{g}_l \\
&+ \left(2\tau_{2,1}\gamma_r-\alpha\right)v_m^2 - 2\tau_{2,1}v_m g_r \\
=& \beta\left(d_{1,m}v\right)^2 + 2\left(\beta+\tau_{1,2}\delta_l\right)v_1 d_{2,1}v + 2\left(-\beta+\tau_{2,1}\delta_r\right)v_m d_{2,m}v \\
&+ \left(2\tau_{1,1}-\beta\right)\left(d_{1,1}v - \frac{\tau_{1,1}g_l}{\left(2\tau_{1,1}-\beta\right)}\right)^2 - \frac{\left(\tau_{1,1}g_l\right)^2}{\left(2\tau_{1,1}-\beta\right)} \\
&+ \left(2\tau_{1,2}\gamma_l+\alpha\right)\left(v_1 - \frac{\tau_{1,2}\tilde{g}_l}{\left(2\tau_{1,2}\gamma_l+\alpha\right)}\right)^2 - \frac{\left(\tau_{1,2}\tilde{g}_l\right)^2}{\left(2\tau_{1,2}\gamma_l+\alpha\right)} \\
&+ \left(2\tau_{2,1}\gamma_r-\alpha\right)\left(v_m - \frac{\tau_{2,1}g_r}{\left(2\tau_{2,1}\gamma_r-\alpha\right)}\right)^2 - \frac{\left(\tau_{2,1}g_r\right)^2}{\left(2\tau_{2,1}\gamma_r-\alpha\right)}. \qquad (18)
\end{aligned}
$$

Choosing $\tau_{1,1} = \beta, \quad \tau_{1,2} = -\frac{\beta}{\delta_l}, \quad \tau_{2,1} = \frac{\beta}{\delta_r}$ results in a non-growing solution in time, thus producing an energy-stable semi-discrete problem.

2.4 Full discretisation

In this section a fully discrete energy estimate is derived using the SBP-SAT method in both time and space. Firstly, the semi-discrete problem (13) can be rewritten on the form

$$
\begin{aligned}
v_t &= H_x^{-1}Av + H_x^{-1}G(t) + F(t), \quad t \geq 0 \qquad (19)\\
v|_{t=0} &= f, \quad t = 0 \qquad (20)
\end{aligned}
$$

where, for the semi-discrete problem with characteristic boundary conditions in equation (15) (with derived penalty parameters)

$$
\begin{aligned}
A = -\alpha Q - \beta R + \left(\frac{\alpha}{2} - \frac{\beta}{\delta_l}\gamma_l\right)e_1e_1^T + \left(\frac{\beta}{\delta_r}\gamma_r - \frac{\alpha}{2}\right)e_me_m^T \\
+ \frac{\beta}{2}d_{1,1}^T d_{1,1} + \frac{\beta}{2}d_{1,m}^T d_{1,m} \qquad (21)
\end{aligned}
$$

and

$$
\begin{aligned}
G(t) &= -\beta d_{1,1}g_l(t) - \frac{\beta}{\delta_l}e_1\tilde{g}_l(t) - \frac{\beta}{\delta_r}e_m g_r(t) \qquad (22)\\
F(t) &= \left[F(x_1,t), F(x_2,t), \ldots, F(x_m,t)\right]^T \qquad (23)\\
f &= \left[f(x_1), f(x_2), \ldots, f(x_m)\right]^T \qquad (24)
\end{aligned}
$$

Before continuing we will define a convention for representing the problem, and introduce the Kronecker product.

Definition 3. *Let A be a $n \times m$ matrix and let B be a $p \times q$ matrix. The* Kronecker product *is defined as*

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \dots & a_{1,m}B \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,m}B \end{bmatrix}$$

The result therefore has dimensions $np \times qm$. The product is bilinear and associative. The mixed product rule for the Kronecker product is defined by the relation

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

Definition 4. *Let $t = i\Delta t$ for $i = 1, 2, \dots, n$ where $n\Delta t = T$ define a temporal grid on $[0, T]$, and let $x = L + j\Delta x$ for $j = 1, 2, \dots, m$ where $m\Delta x = R$ define a spatial grid on $[L, R]$.*

Let

$$\bar{v} = [v_1, v_2, \dots, v_n]^T, \quad v_i = [v_{i,1}, v_{i,2}, \dots, v_{i,m}]^T$$

be the $nm \times 1$ solution vector on these grids. v_i are subvectors defining the solution in space at time $t = i\Delta t$.

$$\begin{aligned} \bar{G} &= [G_1, G_2, \dots, G_n]^T, \quad G_i = G(i\Delta t) \\ \bar{F} &= [F_1, F_2, \dots, F_n]^T, \quad F_i = F(i\Delta t) \\ \bar{f} &= [f, v_2, v_3, \dots, v_n]^T \end{aligned}$$

Definition 5. *Let the SBP operator for the first derivative in time be denoted by $D_{1,t} = H_t^{-1}C_t$, where the included matrices are $n \times n$. Let*

$$E_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix}$$

be an $n \times n$ matrix.

Now that the full discretisation has been defined, we can formulate the SBP-SAT method in time from the semi-discretisation outlined in equation (19). The fully discrete SBP-SAT formulation is

$$\begin{aligned} \left(H_t^{-1}C_t \otimes I_x\right)\bar{v} = \left(I_t \otimes H_x^{-1}A\right)\bar{v} + \left(I_t \otimes H_x^{-1}\right)\bar{G} \\ + \sigma_t \left(H_t^{-1}E_1 \otimes I_x\right)\left(\bar{v} - \bar{f}\right) + \bar{F}. \end{aligned} \tag{25}$$

The next step is to derive the fully discrete energy estimate. Multiplying by $\bar{v}^T (H_t \otimes H_x)$ and adding the transpose we obtain

$$\bar{v}^T \left(\left(C_t + C_t^T\right) \otimes H_x \right) \bar{v} = \bar{v}^T \left(H_t \otimes \left(A + A^T\right)\right) \bar{v} + \bar{v}^T \left(H_t \otimes I_x\right) \bar{G} + \bar{G}^T \left(H_t \otimes I_x\right) \bar{v}$$
$$+ \sigma_t \bar{v}^T \left(E_1 \otimes H_x\right) \left(\bar{v} - \bar{f}\right) + \sigma_t \left(\bar{v} - \bar{f}\right)^T \left(E_1 \otimes H_x\right) \bar{v}$$
$$+ \bar{v}^T \left(H_t \otimes H_x\right) \bar{F} + \bar{F}^T \left(H_t \otimes H_x\right) \bar{v}.$$

Simplifying this, using the identity $(x-y)^T M(x-y) = x^T Mx - x^T My - y^T Mx + y^T My$, recognising that $\bar{v}^T \left(H_t \otimes I_x\right) \bar{G}$ can be written as $(v^1)^T H_t G^1 + (v^m)^T H_t G^2$ and setting the forcing function $\bar{F} = 0$ (this does not affect stability) leads to the fully discrete energy estimate

$$\begin{aligned} v_n^T H_x v_n = & (1+a)(v^1)^T H_t v^1 + (1+b)(v^m)^T H_t v^m \\ & + (\sigma_t + 1) v_1^T H_x v_1 + \sigma_t f_1^T H_x f_1 \\ & + \sigma_t \left(v_1 - f_1\right)^T H_x (v_1 - f_1) - \left(v^1 - G^1\right) H_t \left(v^1 - G^1\right) \\ & - \left(v^m - G^2\right)^T H_t \left(v^m - G^2\right) + (G^1)^T H_t G^1 + (G^2)^T H_t G^2 \\ & + \beta \bar{v}^T \left(H_t \otimes d_{1,1}^T d_{1,1}\right) \bar{v} + \beta \bar{v}^T \left(H_t \otimes d_{1,m}^T d_{1,m}\right) \bar{v}. \end{aligned} \quad (26)$$

where $a = 2(\frac{\alpha}{2} - \frac{\beta}{\delta_l}\gamma_l)$ and $b = 2(\frac{\beta}{\delta_r}\gamma_r - \frac{\alpha}{2})$. We see that setting $\sigma_t = -1$ leads to stability if $a, b \leq -1$.

3 Results

The analytical soliton solution of the KdV equation (2) was used in the efficiency- and convergence studies of the time integration methods. The soliton solution is shown in figure 1.

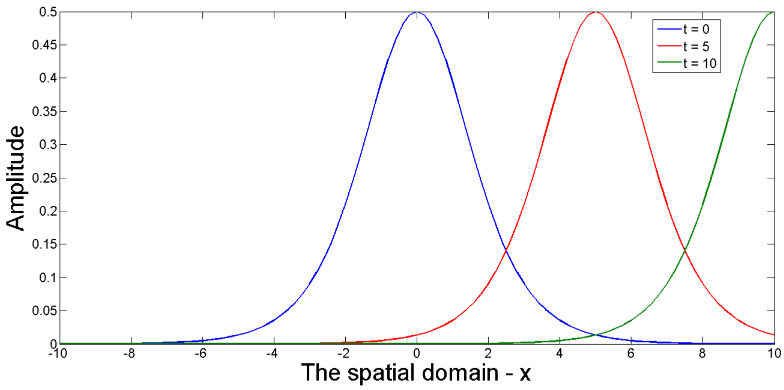

Figure 1: A single wave moving to the right, illustrated by being plotted for three different times.

The constants for equation (2) is $a = 0$, $c = 1$. The intervals used were $-10 \leq x \leq 10$ and $0 \leq t \leq T$, where $T = 10$. The same settings were used for all calculations.

3.1 Convergence

In order to verify that the methods deliver the expected convergence we present convergence plots in the following three subsections for all three methods used. The SBP operators used have different orders of accuracy on the boundary and the interior. The spatial derivative operators SBP2 and SBP4 have 2nd and 4th order accuracy respectively in the interior and 1 on the boundary, while SBP6 is 6th order accurate in the interior and 2nd order accurate on the boundary in space.

3.1.1 Convergence for SBP-SAT using 4th order Crank Nicolson for time discretisation

The number of spatial discretisation points used was $m = 21, 31, 41, 51, 101, 201, 301, 401, 501, 601, 701, 801, 901, 1001$. The spatial step size used was $\Delta x = \frac{R-L}{m-1}$. Since CN4 is an implicit method no CFL condition is required. However, due to the physical properties involved, the solution exhibited unstable behaviour unless the time step was constrained to $\Delta t = 0.1\Delta x$.

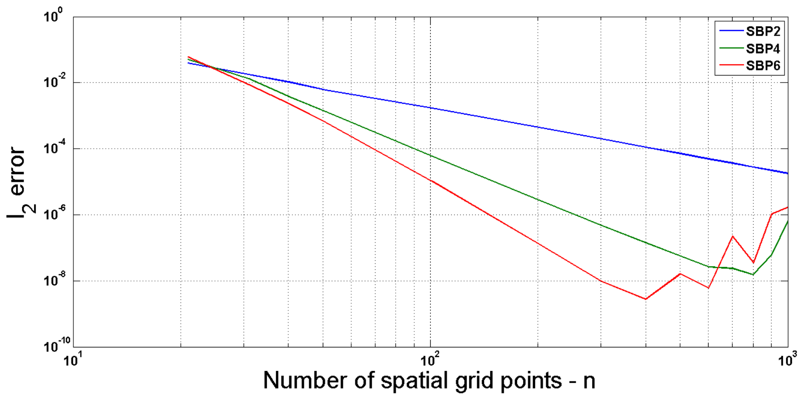

Figure 2: A logscale plot of the pointwise error against the number of discretisation points for CN4

The convergence for SBP(2,1) is 1.9828, SBP(4,1) has 4.3952 (excluding datapoints for numbers of spatial points greater than 601) and SBP(6,2) shows a convergence of 5.8983 (excluding datapoints for numbers of spatial points greater than 401). As can be seen in Figure 2 the method stops converging for large numbers of discretisation points. This is due to the condition number of the matrix involved increasing rapidly at the same time as the time step decreases which leads to a greater accumulation of the error due to ill-conditioning.

3.1.2 Convergence for SBP-SAT using RK4 for time discretisation

The number of discretisation points used for RK4 was $m = 21, 31, 41, 51, 101, 201, 301, 501, 1001$. The spatial steps used was the length of the interval divided by the number of discretisation points minus one. The temporal steps where obtained from the CFL condition $\Delta t = 0.1\Delta x^3$.

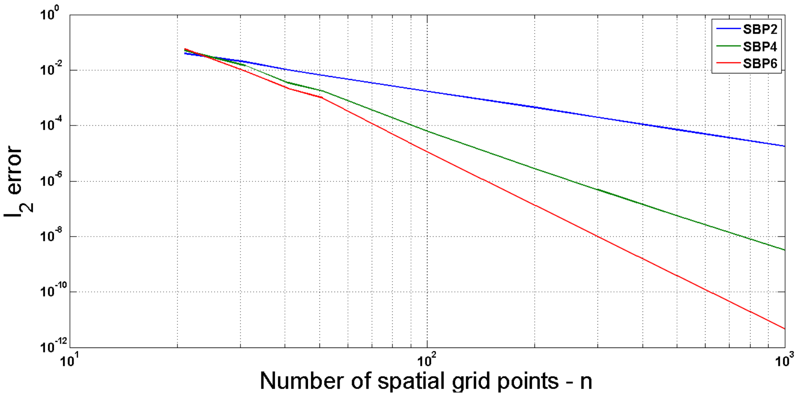

Figure 3: A logscale plot of the pointwise error against the number of discretisation points for RK4

The convergence for SBP(2,1) is 1.9967, SBP(4,1) has 4.3976 and SBP(6,2) shows a convergence of 6.1241.

3.1.3 Convergence for SBP-SAT in time

For the SBP in time method the numbers of discretisation points were $m = 21, 31, 41, 51, 101$ in time, together with the condition $\Delta x = 0.8\Delta t^2$ for the spatial steps in order to make sure that the spatial error did not dominate.

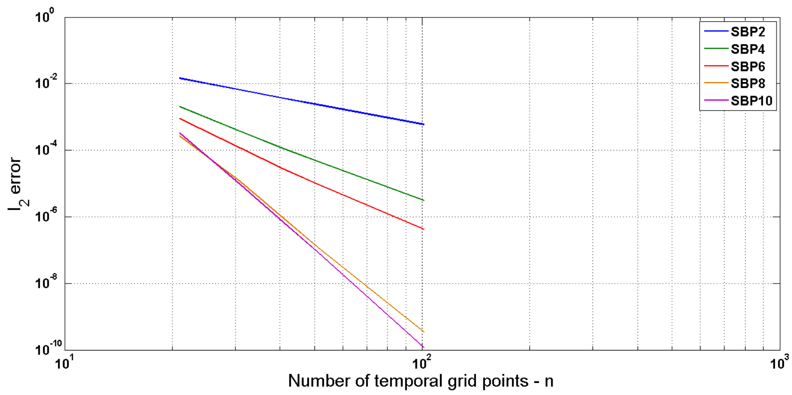

Figure 4: A logscale plot of the pointwise error against the number of discretisation points for SBP in time

The convergence for SBP2 is 2.0332, SBP4 has 4.1195, SBP6 shows 4.8585, SBP8 shows 8.6808 and SBP10 shows a convergence of 9.4612, which is in line with the expected order of accuracy.

3.2 Efficiency

In order to show the efficiency of the different methods the execution time for each run was calculated and and compared against the l_2-error. Thereby it is possible to see the execution time needed to reach a certain error.

Since the CN4 method is ill-conditioned for big matrices and stops converging, it was run with different time steps to better show the tradeoff between execution time and error. During all efficiency measurements 6th order SBP was used for the spatial discretisation. SBP in time was run with the 10th order, block norm operators in time. The 10th order block norm operators are 10th order accurate in the interior and 9th order accurate on the boundary [1].

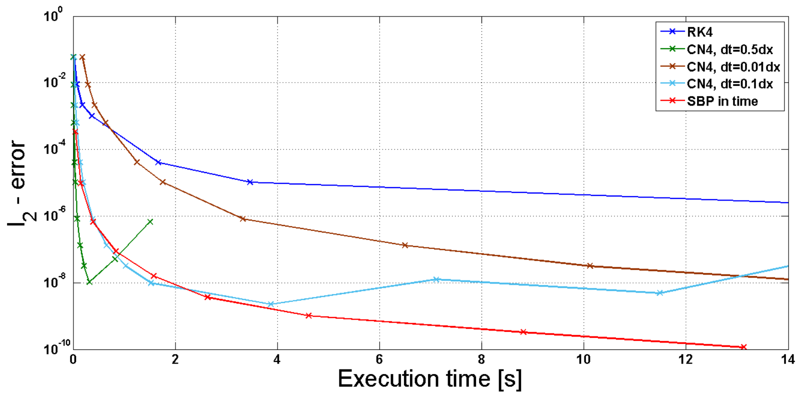

Figure 5: Plot of the pointwise error against execution time for CN4, RK4 and SBP in time

In Figure 5 we can see that RK4 is slow compared to the other methods. CN4 stops converging at different levels, depending on the size of the time steps, as can be seen. SBP in time shows consistent behaviour and converges quickly, with the smallest error of any method tested.

4 Discussion and Conclusions

SBP in time is a highy accurate, efficient and stable method that is relatively easy to implement. Comparing the method to RK4 we find that SBP in time has a large advantage for the KdV-equation due to the stiffness brought on by the third derivative term which requires us to choose the timestep as $k \sim h^3$. RK4 is also an inherently serial algorithm, which puts a hard limit on the amount of performance that can be extracted from modern computer hardware. The CN4 method has the advantage over RK4 of being an implicit method, which makes it significantly more efficient for stiff problem. However, CN4 is a difficult method to implement due to the complicated correction terms involved. It also requires the solution of a system of linear equations in each time step which compounds the error made due to ill-conditioning of the matrix (during our trials we saw condition numbers on the order of 10^6 when the number of points in space was large).

The disadvantage of using SBP in time is that, due to the Kronecker product, the matrix of the system of linear equations grows like $O(m^2n^2)$ where m is the number of point in space and n is the number of points in time. This problem can be mitigated somewhat by using sparse matrices (SBP matrices have a sparse structure), which we have done. A better method of reducing memory requirements would be to use the 'multi block formulation' (or 'restart'-method) outlined in *Lundquist et al* [2] which splits the time domain into p 'steps' containing n/p time levels and restarting the computations with the previous end time as the initial condition for the next 'step'. This reduces the memory requirements to $O((n/p)^2 m^2)$

Further studies of SBP in time may focus on analysing the performance of the method when using the restart method for various problem (for example the KdV-equation). Another important area of study is the application of SBP in time to truly nonlinear problems to see if the method is effective even when the system of equations is nonlinear.

References

[1] Jan Nordström, Thomas Lundquist: *Summation-by-parts in time.* Department of Mathematics, Computational Mathematics, Linköping University, 2013.

[2] Thomas Lundquist, Jan Nordström: *The SBP-SAT technique for initial value problems.* Department of Mathematics, Computational Mathematics, Linköping University, 2013.

[3] Ken Mattson: *Diagonal-norm summation by parts operators for finite difference approximations of third and fourth derivatives.* Division of Scientific Computing, Department of Information Technology, Uppsala University, 2013.

[4] H.-O. Kreiss and G. Scherer: *Finite element and finite difference meth- ods for hyperbolic partial differential equations. Mathematical Aspects of Finite Elements in Partial Differential Equations.*, Academic Press, Inc., 1974.