

UNIVERSITY OF CENTRAL FLORIDA

MAP 4384: NUMERICAL METHODS FOR COMPUTATIONAL SCIENCES
FINAL PROJECT

The Non-Linear Schrödinger Equation

Chris Marsh, Daniel Mandragona, Michael Groff

facilitated by
Dr. Alvaro Islas

April 30, 2018

Contents

1	Abstract	2
2	Notation	2
3	Derivation of Constants	2
3.1	Constant M	2
3.2	Constant N	3
4	NLSE Schemes	4
4.1	NLS1	5
4.2	NLS2	5
4.3	Crank-Nicholson	6
5	Experiment 1 - Sech(x)	7
5.1	NLS1	7
5.2	NLS2	8
5.3	Crank-Nicholson	9
5.4	Conclusion of Experiment 1	9
6	Experiment 2 - One Bump	10
6.1	NLS1	10
6.2	NLS2	11
6.3	Crank-Nicholson	12
6.4	Conclusion - One Bump	12
7	Experiment 3 - Competing Waves	13
7.1	Competing Waves with Crank-Nicholson	14
7.2	Competing Waves with NLS2	14
7.3	Competing Waves with NLS1	15
7.4	Conclusion - Two Bump	16
8	Conclusion	16

1 Abstract

We will be examining a special case of the Schrödinger equation known as the non-linear Schrödinger equation and different approaches to solving this equation numerically with periodic boundary conditions given by:

$$iu_t + u_{xx} + 2|u|^2u = 0 \quad (1)$$

With periodic boundary conditions:

$$u\left(\frac{-L}{2}, t\right) = u\left(\frac{L}{2}, t\right) \quad (2)$$

And initial conditions:

$$\begin{aligned} u(x, 0) &= \text{sech}(x) \\ u(x, 0) &= a(1 + r\cos(\mu x)) \\ u(x, 0) &= a(1 + r_1\cos(\mu x) + r_2\cos(2\mu x)) \end{aligned}$$

We chose the first initial condition intentionally, as it generates an exact, stable solution. We will use this initial condition to develop our code, and then test our code with the more realistic one-bump and two-bump initial conditions. The purpose of this is to compare our three schemes in terms of approximation error, and CPU runtime cost. Finally in our third experiment, using the two-bump initial conditions, we will see the *rogue wave* phenomenon, and compare the three scheme's ability to approximate this phenomenon.

2 Notation

Throughout this document we will use the following notation.

1. If u , is a complex number of the form $a + bi$, then \bar{u} will denote the complex conjugate of u , such that \bar{u} is of the form $a - bi$.
2. $u(x_i, t_j) := u_i^j$
3. If we are comparing $u(x_i, t_j)$ with $u(x_{i-1}, t_j)$ we will use u_1^j and u_0^j respectively.

3 Derivation of Constants

We will begin by deriving two of the constants. We will call them both M and N respectively. To obtain both of our constants M and N we obtain the complex conjugate of equation (1).

$$-i\bar{u}_t + \bar{u}_{xx} + 2|u|^2\bar{u} = 0 \quad (3)$$

3.1 Constant M

To obtain our constant M , Multiply equation (1) by \bar{u} , and equation (3) by u .

$$\bar{u}(iu_t + u_{xx} + 2|u|^2u) = 0 \quad (4)$$

$$u(-i\bar{u}_t + \bar{u}_{xx} + 2|u|^2\bar{u}) = 0 \quad (5)$$

Subtracting equation (5) from equation (4), we obtain the following,

$$\begin{aligned}
& i(\bar{u}u_t + u\bar{u}_t) + \bar{u}u_{xx} - u\bar{u}_{xx} + 2|u|^2u\bar{u} - 2|u|^2u\bar{u} \\
& = i(\bar{u}u_t + u\bar{u}_t) + \bar{u}u_{xx} - u\bar{u}_{xx} \\
& = i(u\bar{u}_t) + \bar{u}u_{xx} - u\bar{u}_{xx} = i(|u|^2)_t + \bar{u}u_{xx} - u\bar{u}_{xx} \\
& = i(|u|^2)_t + ((\bar{u}u_x)_x - u_x\bar{u}_x) - ((u\bar{u}_x)_x - u_x\bar{u}_x) \\
& = i(|u|^2)_t + ((\bar{u}u_x)_x - (u\bar{u}_x)_x) - (u_x\bar{u}_x - u_x\bar{u}_x) \\
& = i(|u|^2)_t + (\bar{u}u_x - u\bar{u}_x)_x = 0
\end{aligned}$$

Therefore we have,

$$i(|u|^2)_t + (\bar{u}u_x - u\bar{u}_x)_x = 0 \quad (6)$$

Then by integrating equation (6) with respect to x and using periodic boundary conditions (2), we obtain,

$$\begin{aligned}
& i \int_{-L/2}^{L/2} i(|u|^2)_t + (\bar{u}u_x - u\bar{u}_x)_x dx = 0 \\
& \Rightarrow \int_{-L/2}^{L/2} i(|u|^2)_t dx + \int_{-L/2}^{L/2} (\bar{u}u_x - u\bar{u}_x)_x dx = 0 \\
& \Rightarrow \int_{-L/2}^{L/2} i(|u|^2)_t dx + \int_{-L/2}^{L/2} (\bar{u}u_x - u\bar{u}_x)_x dx = 0 \\
& \Rightarrow \int_{-L/2}^{L/2} i(|u|^2)_t dx + (\bar{u}u_x - u\bar{u}_x)_x \Big|_{-L/2}^{L/2} = 0 \\
& \Rightarrow i \int_{-L/2}^{L/2} (|u|^2)_t dx = 0
\end{aligned}$$

Therefore,

$$M = \int_{-L/2}^{L/2} (|u|^2) dx \quad (7)$$

Note that M is constant with respect to time.

3.2 Constant N

To obtain the second constant N , we multiply equation (1) by \bar{u}_t , and equation (3) by u_t .

$$\bar{u}_t(iu_t + u_{xx} + 2|u|^2u) = 0 \quad (8)$$

$$u_t(-i\bar{u}_t + \bar{u}_{xx} + 2|u|^2\bar{u}) = 0 \quad (9)$$

Adding equation (8) to equation (9), we obtain the following,

$$\begin{aligned}
& i(\bar{u}_t u_t - u_t \bar{u}_t) + (\bar{u}_t u_{xx} + u_t \bar{u}_{xx}) + 2(|u|^2 u \bar{u}_t + |u|^2 \bar{u} u_t) \\
&= ((u_t u_x)_x - \bar{u}_{xt} u_x + (u_t \bar{u}_x)_x - u_{xt} \bar{u}_x) + 2(|u|^2 u \bar{u}_t + |u|^2 \bar{u} u_t) \\
&= ((u_t u_x + u_t \bar{u}_x)_x - [(\bar{u}_x u_x)_t - \bar{u}_x u_{xt} + u_{xt} \bar{u}_x]) + 2(|u|^2 u \bar{u}_t + |u|^2 \bar{u} u_t) \\
&= ((u_t u_x + u_t \bar{u}_x)_x - (|u_x|^2)_t) + 2(|u|^2 u \bar{u}_t + |u|^2 \bar{u} u_t) \\
&= ((u_t u_x + u_t \bar{u}_x)_x - (|u_x|^2)_t) + 2|u|^2 (u \bar{u}_t + \bar{u} u_t) \\
&= ((u_t u_x + u_t \bar{u}_x)_x - (|u_x|^2)_t) + 2|u|^2 (|u|^2)_t \\
&= (u_t u_x + u_t \bar{u}_x)_x - (|u_x|^2)_t + \frac{1}{2}(|u|^4)_t = 0
\end{aligned}$$

Therefore,

$$(u_t u_x + u_t \bar{u}_x)_x - (|u_x|^2)_t + \frac{1}{2}(|u|^4)_t = 0 \quad (10)$$

Then by integrating equation (10) with respect to x and using periodic boundary conditions (10), we obtain,

$$\begin{aligned}
& \int_{-L/2}^{L/2} (u_t u_x + u_t \bar{u}_x)_x + \int_{-L/2}^{L/2} \frac{1}{2}(|u|^4)_t - (|u_x|^2)_t dx = 0 \\
& \Rightarrow \frac{1}{2} \int_{-L/2}^{L/2} (|u|^4)_t - (|u_x|^2)_t dx = 0 \\
& \Rightarrow \int_{-L/2}^{L/2} (|u|^4 - |u_x|^2)_t dx = 0
\end{aligned}$$

Letting $N_t = \int_{-L/2}^{L/2} (|u|^4 - |u_x|^2)_t dx$, we obtain the following,

$$\begin{aligned}
N_t &= \int_{-L/2}^{L/2} (|u|^4 - |u_x|^2)_t dx = 0 \\
\Rightarrow N_t &= \frac{d}{dt} \int_{-L/2}^{L/2} (|u|^4 - |u_x|^2) dx = 0
\end{aligned}$$

Therefore,

$$N = \int_{-L/2}^{L/2} (|u|^4 - |u_x|^2) dx = 0 \quad (11)$$

Note that N is constant with respect to time.

4 NLSE Schemes

In this paper we will cover three methods for numerical approximation of the NLSE as time varies. For the following methods we seek approximation of NLSE from $x = -L/2$ to $x = L/2$, where x varies in steps of $h = L/N$. In addition to the dependence on x , we also record $u(x, t)$ at integer values of t only.

4.1 NLS1

This first scheme computes $u(x, t = i)$ from $u(x, t = i - 1)$. In order to do this we compute $u(x, t)$ at time increments of $k = \frac{1}{4}h^2$. Since we are only *recording* $u(x, t)$ at integer values of t , we only save the values when $k * m \in \mathbb{Z}^+$. Let $n_t = \frac{1}{k}$ then this is how many times we must compute $u(x, t)$ before reaching an integer, therefore the m from above will be some multiple of n_t .

In order to obtain an approximation for NLSE at time t from time $t - 1$ we must first substitute the following approximations into NLSE.

$$u_{xx}(x_i, t_i) = \frac{u(x_{i-1}, t_i) - 2u(x_i, t_i) + u(x_{i+1}, t_i)}{h^2} \quad (12)$$

$$u_t(x_i, t_i) = \frac{u(x_i, t_i) - u(x_i, t_{i-1})}{k} \quad (13)$$

Now plugging these into (4) and using notations for $u(x_i, t_i)$ we get:

$$\begin{aligned} iu_t + u_{xx} + 2|u|^2u &\approx i(9) + (8) + 2|u|^2u \\ &= i\left(\frac{u_x^1 - u_x^0}{k}\right) + \frac{u_0^t - 2u_1^t + u_2^t}{h^2} + 2|u_x^t|^2u_x^t \end{aligned}$$

Solving for $u_x^1 = u_x^0 + ik\left[\frac{1}{h^2}(u_0^t - 2u_1^t + u_2^t) + 2(u_x^t|u_x^t|^2)\right]$ where $x = 1$ and $t = 0$ (14)

We can convert this into matrix form:

$$u(x, t) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N+1} \end{bmatrix}^{t-1} + i \left(\frac{k}{h^2} \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 & 1 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 1 & 0 & \cdots & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N+1} \end{bmatrix}^{t-1} + 2k \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N+1} \end{bmatrix}^{t-1} \right)$$

Now written more compactly:

$$U^t = U^{t-1} + ik\left(\frac{1}{h^2}AU^{t-1} + 2kU^{t-1}\right) \quad (15)$$

$$\text{where } U^t = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N+1} \end{bmatrix}^t \text{ and } A = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 & 1 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 1 & 0 & \cdots & 0 & 1 & -2 \end{bmatrix}$$

4.2 NLS2

The second scheme is an implicit approximation of NLSE where we find $U(x)$ at a specific time t by starting with an initial vector, potentially very different from the accurate vector, and then apply an operator on

this vector iteratively where the result is a vector that will converge to the accurate vector. The process is as follows:

$$\begin{aligned}
\text{To find } V_\ell := U(x, t) \text{ let } V_1 = U(x, t - k) \text{ then} \\
V_2 = G(V_1, V_1) \\
V_3 = G(V_1, V_2) \\
\downarrow \qquad \qquad \downarrow \\
V_\ell = G(V_1, V_\ell)
\end{aligned}$$

The idea behind this is that this process will converge to the accurate vector as you iterate towards infinity. Clearly it is not feasible to iterative an infinite number of times so we will terminate the process if either of the following two conditions are met:

1. $\|V_i - V_{i-1}\| < \epsilon$
2. We have iterated some *levelMax* many times.

So what is this G operator? To find G we begin with the scheme:

$$\begin{aligned}
\frac{U^{n+1} - U^n}{k} &= \frac{i}{h^2} A U^{n+1} + 2i|U^{n+1}|^2 U^{n+1} \\
U^{n+1} - \frac{ik}{h^2} A U^{n+1} &= U^n + 2ik|U^{n+1}|^2 U^{n+1} \\
B U^{n+1} &= U^n + 2ik|U^{n+1}|^2 U^{n+1} \\
U^{n+1} &= B^{-1}(U^n + 2ik|U^{n+1}|^2 U^{n+1}) \\
U^{n+1} &= B^{-1}F(U^n, U^{n+1})
\end{aligned}$$

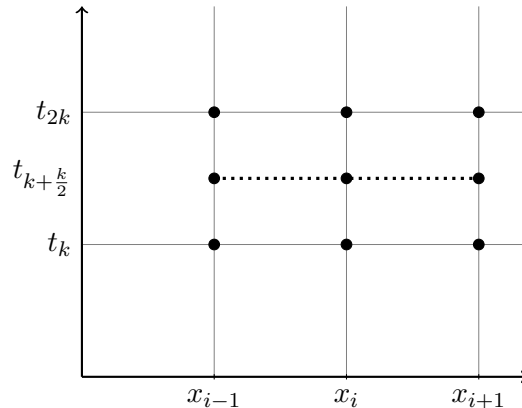
Therefore the G operator is $B^{-1}F(U^n, U^{n+1})$ for definitions:

$$\begin{aligned}
F(U^n, U^{n+1}) &:= U^n + 2ik|U^{n+1}|^2 U^{n+1} \\
B &:= I - \frac{ik}{h^2} A
\end{aligned}$$

4.3 Crank-Nicholson

The third scheme is again an implicit approximation of NLSE where we find our $U(x)$ at a time t and use the time-step prior approximate to half-step in order to decrease the error.

The half-steps are illustrated as:



We again use iteration of the implicit formula to obtain a result very close to the original vector, and again terminate the process once either of the two stopping conditions are met. To derive the implicit formula for the half time-steps first we solve for components of the NLSE in terms of the half time-steps:

$$\begin{aligned}
[U_i^{n+1/2}]_t &\approx \frac{U_i^{n+1} - U_i^n}{h} \\
[U_i^{n+1/2}]_{xx} &\approx \frac{U_{i-1}^{n+\frac{1}{2}} - 2U_i^{n+\frac{1}{2}} + U_{i+1}^{n+1}}{h^2} \\
[U_i^{n+1/2}]_{xx} &\approx \frac{U_{i-1}^n + U_{i-1}^{n+1} - 2(U_i^n + U_i^{n+1}) + (U_{i+1}^n + U_{i+1}^{n+1})}{2h^2} \\
[U_i^{n+1/2}]_{xx} &\approx \frac{1}{2h^2}(AU^{n+1} + AU^n) \\
U_i^{n+\frac{1}{2}}|U_i^{n+\frac{1}{2}}| &\approx \frac{1}{8}(U_i^n + U_i^{n+1})|U_i^n + U_i^{n+1}|
\end{aligned}$$

By using these second scheme with these new time steps we obtain:

$$U^{n+1} - \frac{ik}{2h^2}AU^{n+1} = U^n + \frac{ik}{2h^2}AU^n + \frac{ik}{4}(U_i^n + U_i^{n+1})|U_i^n + U_i^{n+1}|^2$$

We can then simplify this equation by allowing:

$$\begin{aligned}
B_- &= I - \frac{ik}{2h^2}A \\
B_+ &= I + \frac{ik}{2h^2}A \\
F(U^n, U^{n+1}) &= \frac{ik}{4}(U_i^n + U_i^{n+1})|U_i^n + U_i^{n+1}|^2
\end{aligned}$$

This results in the following simplification:

$$\begin{aligned}
B_-U^{n+1} &= B_+U^n + F(U^n, U^{n+1}) \\
U^{n+1} &= B_-^{-1}(B_+U^n + F(U^n, U^{n+1}))
\end{aligned}$$

Then using iteration on both sides by letting $V_o = U^n$ and $V = U^{n+1}$ and solving the following in each iteration:

$$V_{\ell+1} = B_-^{-1}(B_+U^n + F(U^n, V_\ell))$$


We continue until one of our two conditions are met and assign the final value for $V_{\ell+1}$ to U^{n+1} .

5 Experiment 1 - Sech(x)

In the following subsections we test our numerical schemes for NLSE on the initial condition $U(x, 0) = \text{sech}(x)$ which has been shown to be a very stable initial condition for the NLSE. This is because as time increases, the NLSE in theory will remain equal to the $\text{sech}(x)$ function.

5.1 NLS1

In this trial we use the initial condition that $U(x, 0) = \text{sech}(x)$ and x ranges from $-\frac{L}{2}$ to $\frac{L}{2}$ for $L = 20$ in increments of $h := L/N$ for $N = 2^7 = 128$. We evaluate in time step increments of $k = 1 \times 10^{-4}$.




nls1sechfinal.png

Figure 1: We can see that this scheme becomes inaccurate at $t = 15$.

It should be noted that for the following initial conditions, the average time this scheme took was 5.3106 seconds averaged over 30 runs. In the following experiments we will see that NLS1 is the worst of the three schemes. It is the most inaccurate, and as a consequence it requires smaller k values to produce a scheme comparable to that of NLS2 or Crank-Nicholson. This smaller k values translate to more work done by the CPU.

5.2 NLS2

In this trial we use all the initial conditions that we used in NLS1, but the only difference from the NLS1 experiment, is that instead of using time steps of $k = 1 \times 10^{-4}$ we will be using $k = \frac{1}{4}h^2$ which makes k a function of h .



nlse2sechfinal.png

Figure 2: We can see that this scheme is much more accurate, and is able to last for more than 30 seconds. It should be noted however that the maximum values of U suffer a slight decrease as the value of t increases.

The decrease in the maximum values of $|U(x, t)|$ with increasing time mentioned in Figure 2, can also be seen in the graphs of constants M and N , in Figure 4a, and Figure 4b respectively. Despite the slight decrease in the maximum values for $|U(x, t)|$ with respect to increasing time, we conclude that NLS2 is a fairly good approximation of NLSE. It is also important to note that the average time that the script took (averaged over 200 runs) is 1.4907 seconds which is significantly shorter than how long NLS1 took.

5.3 Crank-Nicholson

We now examine the Crank-Nicholson scheme for approximating the NLSE. The average time that this scheme took over 200 trials is 1.84566, this shows that it takes slightly longer than the NLS2 scheme.

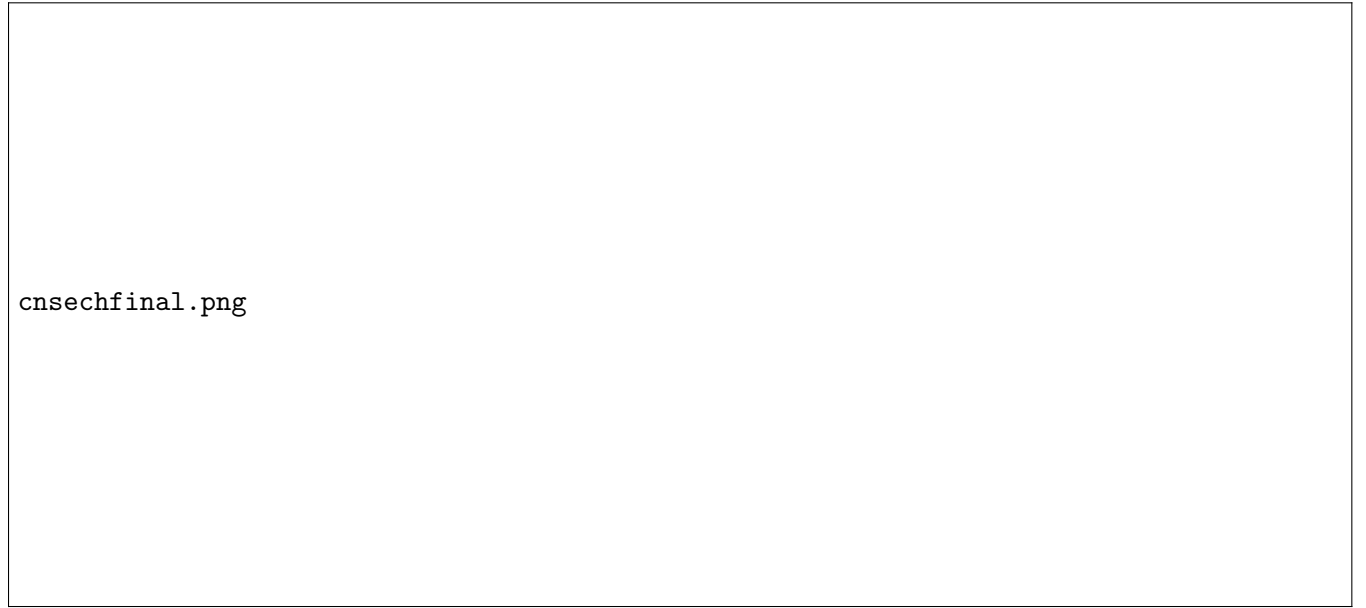
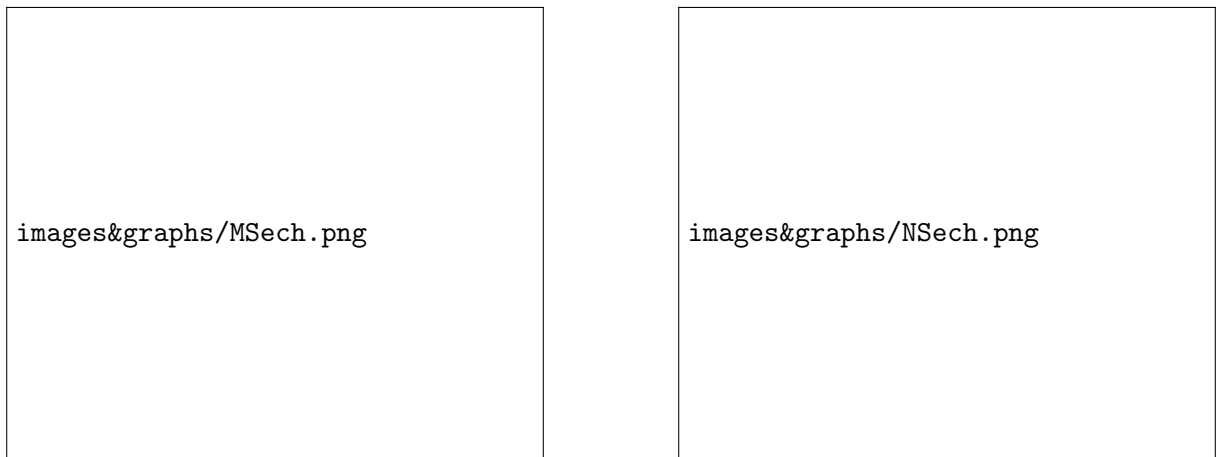


Figure 3: We see that this scheme is very accurate, and does not feature the steady decreasing that appeared in the NLS2 scheme.

5.4 Conclusion of Experiment 1

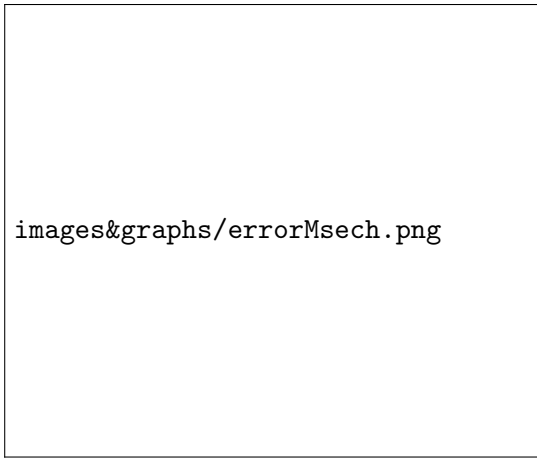
Below is the data for the Constants of Motion using the Sech(x) initial conditions.



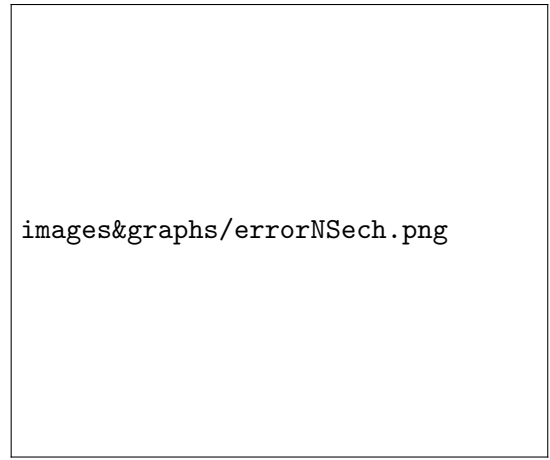
(a) Constant of Motion M

(b) Constant of Motion: N

Figure 4: Constants of Motion for Sech(x) Initial Condition



(a) Error of Constant: M



(b) Error of Constant: N

Figure 5: Error of Constants of Motion for Sech(x) Initial Condition

For the log error of the constants we see that both the M and N constants stay roughly near their initial errors, and that for the Crank-Nicholson scheme the error of the N constant is around 3 orders of magnitude higher than the M constant's. This is because of the error introduced from our central difference approximation of U 's x -derivative. Although all of our schemes use this central difference approximation, it can only be viewed with the Crank-Nicholson scheme because of how accurate this scheme is, the other schemes mask this error.

We conclude that the substantial differences in error makes up for the slightly longer CPU time required to run Crank-Nicholson in comparison to the NLS2 scheme, and so Crank-Nicholson should perform the best out of the three schemes in the experiments to come.

6 Experiment 2 - One Bump


In this experiment we test our schemes on the initial function:

$$u(x, 0) = a[1 + r\cos(\mu x)]$$

Where we let $a = 0.5$, $r = 0.01$, $\mu = 2\pi/L$, $L = 2\sqrt{2}\pi$.

6.1 NLS1

Testing the first scheme under these conditions we found that we had to drastically increase the number of time steps between integer times from $k = 10^{-4}$ to $k = 4 \times 10^{-6}$ just to get a scheme that lasted for 15 seconds. We will see that this must also be done for the two bump initial condition as well.



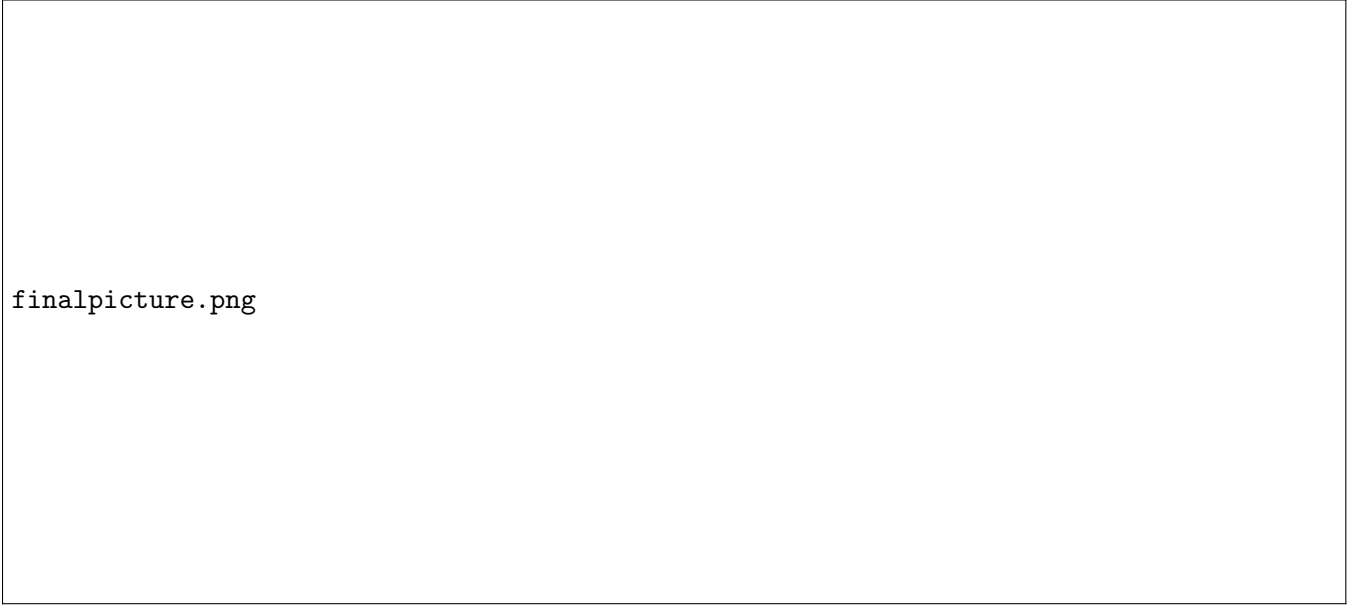
nls1onebumpfinal.png

Figure 6: We can see that the scheme becomes inaccurate at $t = 15$.

It should be noted that due to the drastically increased number of time steps that the run-time of this scheme increased as well. From only taking 5.3106 seconds in experiment one, NLS1 with $k = 4 * 10^{-6}$ takes a staggering 125.26 seconds.

6.2 NLS2

For the NLS2 scheme we go back to using $k = \frac{1}{4}h^2$ for the the time step constant. For reference this comes out to be $k = .0012047$ roughly 300 times the size of the time constant for NLS1.



finalpicture.png

Figure 7: Here we can see that our second scheme is much more stable for the One Bump initial condition. NLS2 succeeds in lasting the full 30 seconds.

This scheme allows us a better time constant, and as a result only takes 5.195 seconds averaged for a 100 trials. So not only is the scheme accurate for twice as long (independent trials showed that it could last for 500 seconds without ever incurring error greater than order 5×10^{-1}), but the run-time needed for the scheme is nearly 20 times less than that needed for NLS1.

While this scheme does reach a large error it returns to an extremely low error at approximately $t = 10$ seconds. As the scheme continues it again creates a large error before again becoming a good approximation at around $t = 30$ seconds.

6.3 Crank-Nicholson

When we apply the Crank-Nicholson scheme to the same boundary conditions we obtain the following surface:

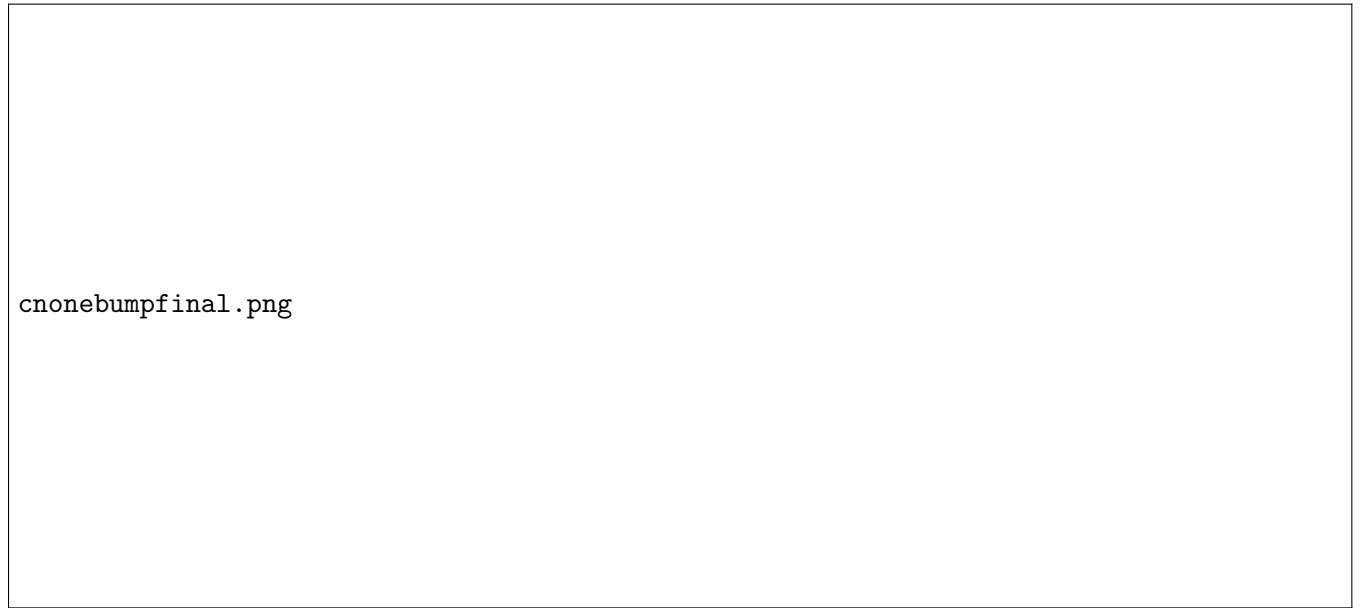
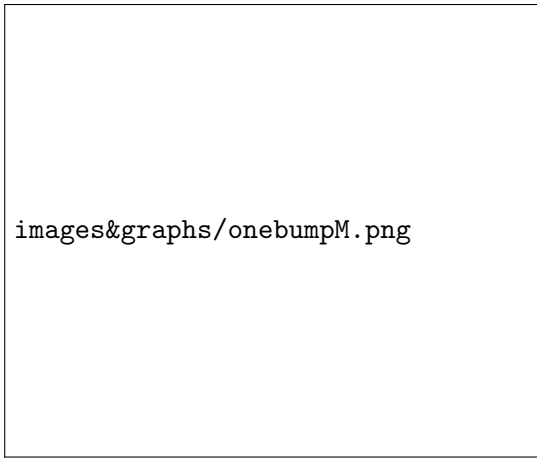


Figure 8: Notice that this surface only has one peak vs. the two peaks that appear in NLS2. It should be noted that the max value at the peak is approximately 1.2

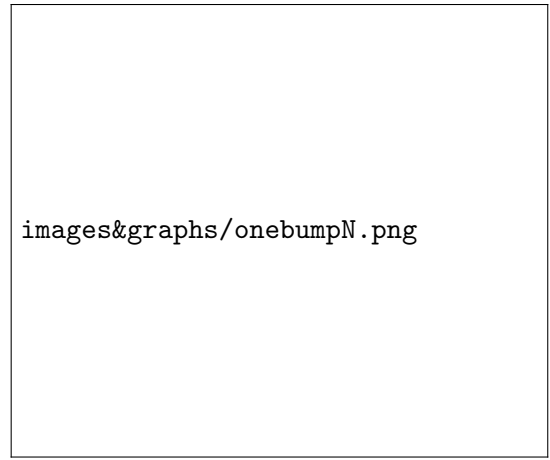
We find this to be the most accurate scheme. It features only slightly longer run-time requirements than NLS2 (6.5 seconds averaged over 100 runs), but offers a much more accurate surface.

6.4 Conclusion - One Bump

Below is the data for the Constants of Motion and their errors for the one bump initial condition. The error plots for the constants of motion confirm that the Crank-Nicholson scheme is the best of our three schemes.



(a) Constant of Motion: M

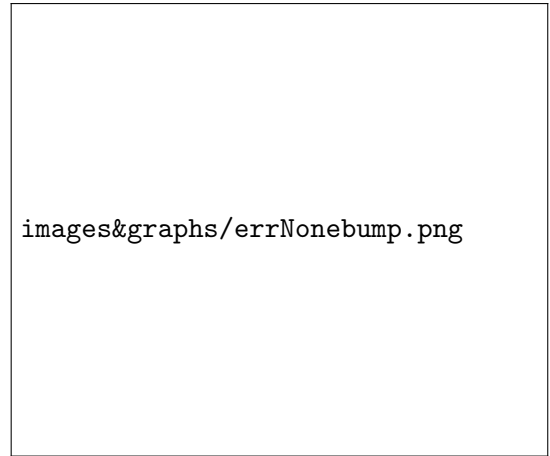


(b) Constant of Motion: N

Figure 9: Constants of Motion for the One Bump initial condition



(a) Log Error of M



(b) Log Error of N

Figure 10: Log Error of Constants for the One Bump initial condition.

We noticed that although both NLS2 and Crank-Nicholson are accurate for much longer than NLS1, that NLS2 produced a very different surface than Crank-Nicholson's. This is because that although NLS2 has small error (order 10^{-2}), it is still nearly 100 times that of Crank-Nicholson's at the later time periods. We might be led to believe that an error of order 10^{-2} is pretty good, but this experiment proves otherwise since this error produces a drastically different surface.

Another interesting thing that we see from the error plots is that the error increases at times, but then shrinks at later times. Our hypothesis is that this is caused from the surface peaks. We believe that the errors are masked by the function $|U(x, t)|$ decreasing after its peak. Since the NLS2 surface has an additional peaking time for $|U(x, t)|$ than the Crank-Nicholson scheme, that the NLS2 error does not benefit from this mask.

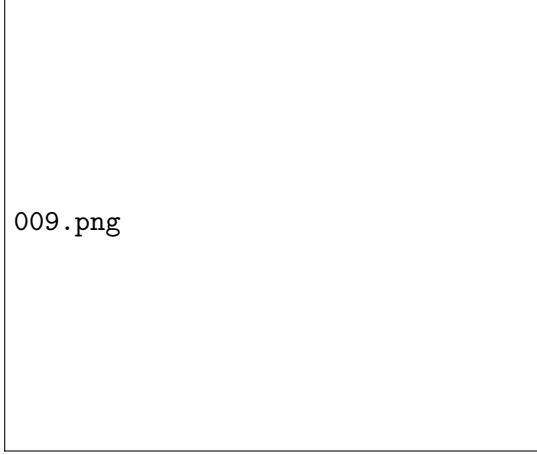
7 Experiment 3 - Competing Waves

For this experiment we test our schemes with the initial function of competing waves. This equation is:

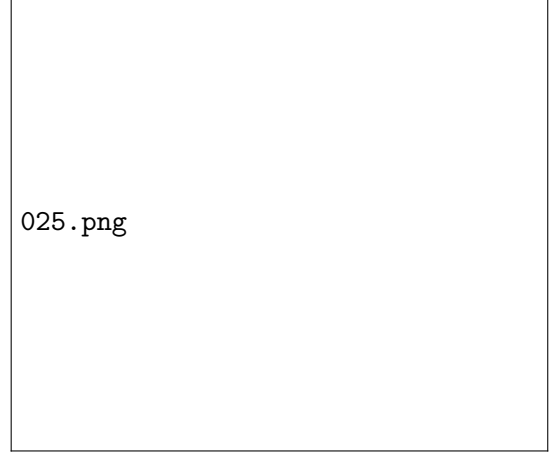
$$U(x, 0) = \frac{1}{2} \left[1 + r_1 \cos\left(\frac{2\pi}{L}x\right) + r_2 \cos\left(\frac{4\pi}{L}\left(x - \frac{L}{4}\right)\right) \right];$$

where $L = 4\pi\sqrt{2}$, $N = 128$, $h = \frac{L}{N}$, and $k = \frac{1}{4}h^2$. For the following trials we keep the constant $r_1 = .1$ fixed while we examine changing values for r_2 . The values of r_2 come from the set $S = \{.009, .025, .0375, .1\}$.

7.1 Competing Waves with Crank-Nicholson

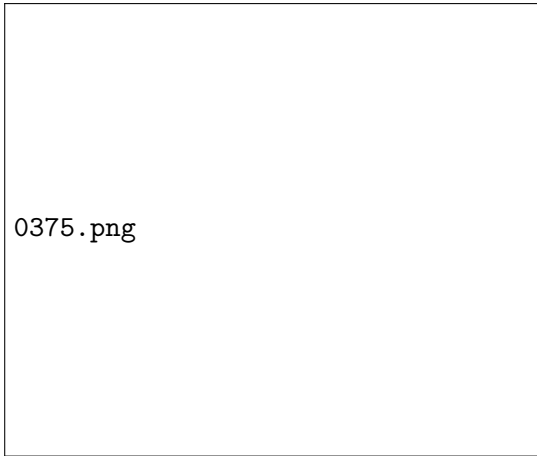


(a) Here we have set $r_2 = .009$. We examine the graph of $U(x,t)$ from the side and see that the *strong* wave is in front of the *weak* wave.

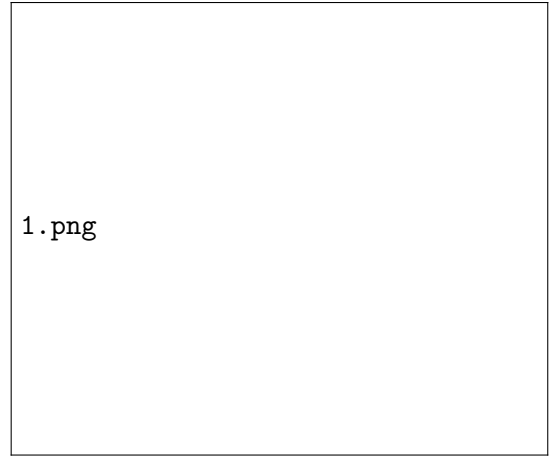


(b) We now set $r_2 = .025$, and see that the *strong* wave is now closer to the *weak* wave.

Figure 11: Competing Waves.



(a) Now we set $r_2 = .0375$ and see that the two waves have now coalesced and have gone through constructive interference.



(b) Finally we set $r_2 = .1$, and see that the *strong* wave has swapped positions with the *weak* wave.

Figure 12: Competing Waves(continued).

These four graphs exhibit the *rogue* wave phenomenon, where the nonlinear term in the Schrödinger Equation takes effect, and the *strong* wave takes in energy from the *weak* wave using this to bolster its own peak.

7.2 Competing Waves with NLS2

For this experiment we test NLS2 only with the $r_2 = .0375$. This is because $r_2 = .0375$ is the critical r_2 value that demonstrates the *rogue wave* phenomenon, and that Crank-Nicholson has proven to be a better

approximation of NLSE.



(a) Mesh of NLS2 with $r_2 = .0375$

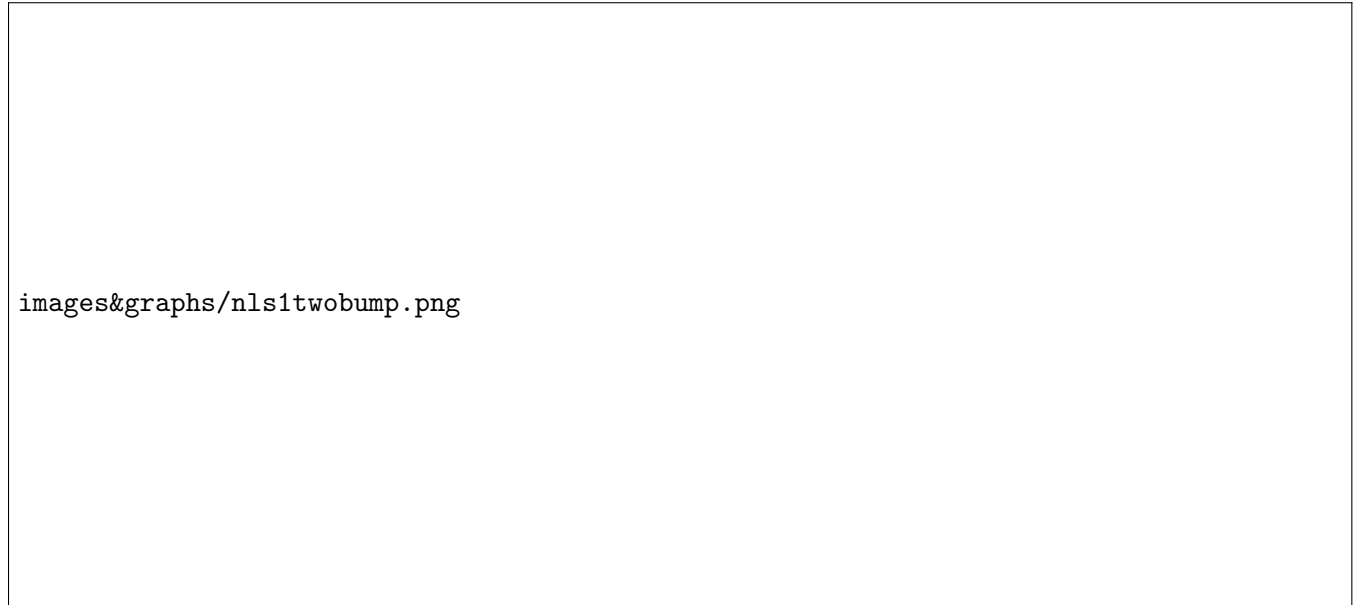


(b) Side view of (a). We see that NLS2 with the Two Bump initial condition follows the trend of decreasing with time.

Figure 13: NLS2 Mesh Side-View & Log Error

7.3 Competing Waves with NLS1

In this section we attempt to use the competing waves initial condition with the NLS1 scheme. We see that even by decreasing the size of the time steps and increments in the x value, that we still cannot get an accurate approximation with this scheme. $k = 5 \times 10^{-5}$, we get the following surface for $r_2 = 0.0375$:



Interestingly, the NLS1 scheme works better with the two bump initial condition than it does with the one bump initial condition. Although it only lasts for 15 seconds, it accomplishes this with an improved time constant. This improved time constant translates to an average time per trial of 9.2 seconds (averaged over 100 trials) compared to 125 seconds for the one bump initial condition.

7.4 Conclusion - Two Bump

The Constants of Motion and their errors for the two bump initial condition are displayed below:

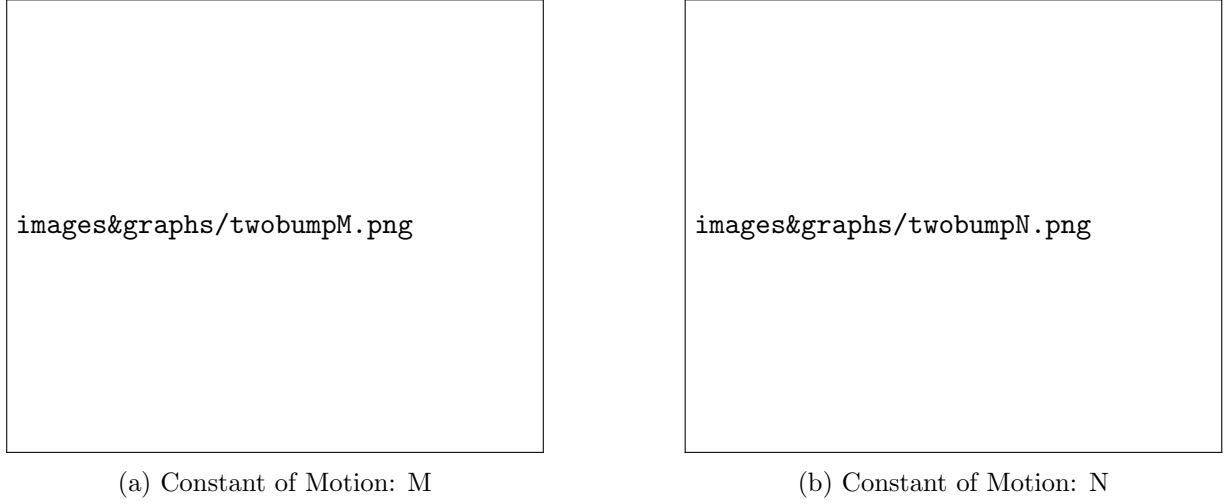


Figure 14: Constants of Motion for Two Bump initial condition.



Figure 15: Log Error of Constants of Motion

From these plots we see that again Crank-Nicholson is the best of the three schemes, and that the NLS2 scheme continues with this trend of shrinking over time. Furthermore we found that the NLS1 scheme performed surprisingly better than it had with the one bump initial condition. Finally we have seen that the *rogue wave* phenomenon appears in all three of these schemes, even if only for a moment (NLS1 blew up directly after).

8 Conclusion

From the application of these three schemes to varying initial conditions we can see just how important choosing a scheme can be. We found that the Crank-Nicholson scheme is the most accurate of our three schemes, while only coming at a marginal cost in terms of actual run-time. While there is still a certain degree of error in all of these schemes, the schemes themselves vary greatly in performance and CPU time.

Further study into derivation and testing of new schemes may yield better approximations in the future that can be more generally applied.