

Mat1062: Introductory Numerical Methods for PDE

Mary Pugh

April 9, 2009

1 Ownership

These notes are the joint property of Rob Almgren and Mary Pugh.

2 Let's compute!

In the following, we will use Richardson Extrapolation. This is a method of accelerating a computation that's done on an equal-spaced mesh. In its most general form, assume that $A(k)$ is the output that you get if you do the computation with time-step k and assume that $A(k/2)$ is the output that you get with time-step $k/2$. If the error is $\mathcal{O}(k^n)$ then we use n to create a new output:

$$\frac{2^n}{2^n - 1}A(k/2) - \frac{1}{2^n - 1}A(k) \quad (1)$$

This output will be more accurate than either of the parts that went into it. Why should this work? Assume that

$$A(k) = \text{Truth} + Ck^n + Dk^{n+1} + \dots \quad (2)$$

Then

$$A(k/2) = \text{Truth} + C\frac{k^n}{2^n} + D\frac{k^{n+1}}{2^{n+1}} + \dots$$

and so

$$\begin{aligned}
 \frac{2^n}{2^n-1}A(k/2) - \frac{1}{2^n-1}A(k) &= \frac{2^n}{2^n-1} \left[\text{Truth} + C\frac{k^n}{2^n} + D\frac{k^{n+1}}{2^{n+1}} + \dots \right] \\
 &\quad - \frac{1}{2^n-1} \left[\text{Truth} + Ck^n + Dk^{n+1} + \dots \right] \\
 &= \text{Truth} + C\frac{k^n}{2^n-1} + D\frac{k^{n+1}}{2(2^n-1)} - C\frac{k^n}{2^n-1} - D\frac{k^{n+1}}{2^n-1} + \dots \\
 &= \text{Truth} - D\frac{k^{n+1}}{2(2^n-1)}.
 \end{aligned}$$

That is, the expression (1) is $\mathcal{O}(k^{n+1})$ close to the true solution even though it's built out of pieces that are only $\mathcal{O}(k^n)$ close to the true solution. Note that I assumed that the next order in the expansion (2) was of order k^{n+1} . If it were of higher order then (1) would be more accurate than $\mathcal{O}(k^{n+1})$.

We consider three time-stepping schemes:

1. The first scheme is as described above: given u^n , take one time step of size k to find u^* , the solution of $iu_t = u_{xx}$ with initial data u^n . Then take a time step of size k to find u^{n+1} , the solution of $iu_t = -|u|^n u$ with initial data u^* . This is the most basic splitting method.
2. The second scheme is a Strang splitting. Given u^n , take one time step of size $k/2$ to find u^* , the solution of $iu_t = u_{xx}$ with initial data u^n . Then take a time step of size k to find u^{**} , the solution of $iu_t = -|u|^n u$ with initial data u^* . Then take one time step of size $k/2$ to find u^{n+1} , the solution of $iu_t = u_{xx}$ with initial data u^{**} .
3. The third scheme is accelerated Strang splitting. Let u_c^{n+1} be what you get by applying the Strang splitting to u^n . Specifically, you took three time steps $k/2$, k , and $k/2$ to get from u^n to u_c^{n+1} . Now, let u_f^{n+1} be what you get by applying the Strang splitting twice. Specifically, you take steps $k/4$, $k/2$, and $k/4$ to get from u^n to $u^{n+1/2}$ and then take steps $k/4$, $k/2$, and $k/4$ to get from $u^{n+1/2}$ to u_f^{n+1} . We use Richardson extrapolation to blend the coarse and fine computations:

$$u^{n+1} = \frac{4}{3}u_f^{n+1} - \frac{1}{3}u_c^{n+1}$$

By construction, the first scheme will conserve the discrete particle number

$$\sum_{j=0}^{N-1} |u_j^n|^2.$$

It will be first-order accurate in time. The second scheme will also conserve the particle number and will be second-order accurate in time. The third scheme will be fourth-order accurate in time but it will not preserve the particle number. Indeed, we know that

$$\sum_{j=0}^{N-1} |u_{f,j}^{n+1}|^2 = \sum_{j=0}^{N-1} |u_j^n|^2 \quad \text{and} \quad \sum_{j=0}^{N-1} |u_{c,j}^{n+1}|^2 = \sum_{j=0}^{N-1} |u_j^n|^2.$$

And so

$$\begin{aligned} \sum_{j=0}^{N-1} \left| \frac{4}{3} u_{f,j}^{n+1} - \frac{1}{3} u_{c,j}^{n+1} \right|^2 &= \frac{16}{9} \sum_{j=0}^{N-1} |u_{f,j}^{n+1}|^2 + \frac{1}{9} \sum_{j=0}^{N-1} |u_{c,j}^{n+1}|^2 - \frac{8}{9} \sum_{j=0}^{N-1} \operatorname{Re} \left(u_{f,j}^{n+1} \overline{u_{c,j}^{n+1}} \right) \\ &= \frac{17}{9} \sum_{j=0}^{N-1} |u_j^n|^2 - \frac{8}{9} \sum_{j=0}^{N-1} \operatorname{Re} \left(u_{f,j}^{n+1} \overline{u_{c,j}^{n+1}} \right) \\ &= \sum_{j=0}^{N-1} |u_j^n|^2 + \frac{8}{9} \sum_{j=0}^{N-1} |u_j^n|^2 - \operatorname{Re} \left(u_{f,j}^{n+1} \overline{u_{c,j}^{n+1}} \right) \end{aligned}$$

The second sum would need to equal zero for the particle number to be preserved.

For all three schemes in addition to compute the discrete particle number at each time step t_n , we compute a discretization of the Hamiltonian $H(t_n)$:

$$H(t_n) = \int_0^{2\pi} |u_x(x, t_n)|^2 - \frac{1}{2} |u|^4 \, dx.$$

We approximate this integral via the trapezoidal rule. None of the three schemes conserve the Hamiltonian and so we will be able to use the Hamiltonian as another measure of accuracy. For this reason, whatever quadrature rule we use to approximate the particle number and the Hamiltonian should be very accurate — we want to be testing the accuracy of the scheme, not of the quadrature rule. In general, the trapezoidal rule is a second-order accurate quadrature rule, however it's spectrally accurate for integrals of

periodic functions.

We want to compare the schemes against exact solutions. The defocussing equation $iu_t = u_{xx} - |u|^2u$ has the plane wave solution

$$Ae^{i(Bx+t(A^2+B^2)+C)} \quad (3)$$

and the focussing equation $iu_t = u_{xx} + |u|^2u$ has the plane wave solution:

$$Ae^{i(Bx+t(B^2-A^2)+C)}, \quad (4)$$

where A , B , C , and D are any real number.

In addition, the focussing equation has soliton solutions

$$u(x, t) = \sqrt{2\alpha} \frac{\exp\left(i\left(\frac{U}{2}x - (\alpha - U^2/4)t + \phi_0\right)\right)}{\cosh(\sqrt{\alpha}(x + Ut - x_0))} \quad (5)$$

The parameters $\alpha > 0$, U , ϕ_0 , and x_0 are arbitrary. These solutions are localised in space — the plot of $|u(x, t)|$ looks somewhat Gaussian. The maximum of $|u|$ is initially at $x = x_0$ and moves with speed U .

We start by testing the first time-stepping scheme for the defocussing equation with the plane wave solution:

$$u(x, t) = 4e^{i(3x-7t)} \quad \text{on } [0, 2\pi] \quad (6)$$

That is, we're taking the parameters in the exact solution (4) to be $A = 4$, $B = 3$, $C = 0$. We take $h = 2\pi/32$ and compute with the initial time step $k = 1/2$. We divide the time steps by 2 at each point. We monitor the L^∞ norm of the error at time $t_f = 5$ as well as how well the particle number $|N(5) - N(0)|$ and Hamiltonian $|H(5) - H(0)|$ are preserved. We find:

k	$\ \text{err}\ _{L^\infty}$	$ M(5) - M(0) $	$ H(5) - H(0) $
1/2	4.4274e-03	4.2633e-14	6.4328e-04
1/4	1.1285e+01	1.9895e-13	8.5858e+03
1/8	1.0087e+01	4.4054e-13	9.0883e+03
1/16	9.9500e+00	8.5265e-14	7.4889e+03
1/32	8.9609e+00	1.9185e-12	8.6124e+03
1/64	2.9204e-12	2.1032e-12	5.2751e-11
1/128	6.1997e-12	3.9506e-12	9.8908e-11

We see that the particle number is well-preserved for all choices of time step. This is unsurprising — conservation of the particle mass was built in to the scheme. However, the L^∞ error of the solution at time $t = 5$ is very poor for time steps $1/2 \dots 1/32$ and is excellent for time steps $1/64$ and $1/128$. We observe a similar thing for the Hamiltonian. This sharp transition from “lousy” to “fantastic” leads us to suspect there is an instability, which we observe in Figure 1. Here, we see that for time step $k = 1/32$ there is a

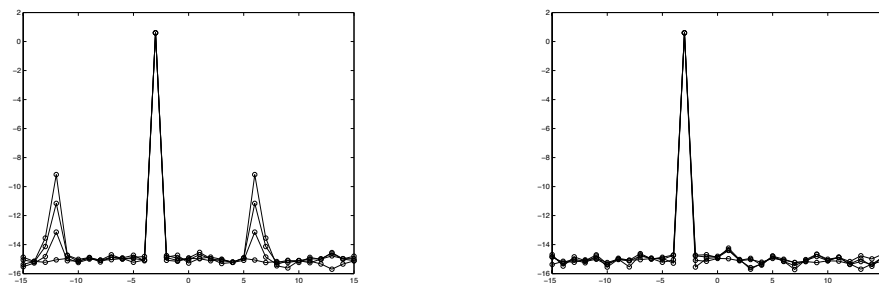


Figure 1: The spectrum is shown for the numerical approximation of the solution (6) at times $t = 0$, $t = 10/32$, $t = 20/32$, and $t = 30/32$. Left plot: The time step is $k = 1/32$. Right plot: The time step is $k = 1/64$.

numerical instability. The exact solution would have a single peak located at wave number -3 . This instability is gone for time-step $k = 1/64$. Because this stability constraint on the time-step is in the first time-stepping scheme (#1 on the above list) it is also in the other two — they are based on the first scheme.

We now test the scheme for the focussing equation with the plane wave solution:

$$u(x, t) = 4e^{i(3x+25t)} \quad \text{on } [0, 2\pi] \quad (7)$$

That is, we’re taking the parameters in the exact solution (3) to be $A = 4$, $B = 3$, $C = 0$. We take the exact same parameters as for Figure 1. Here we observe something quite different when looking at the same time-steps as in Figure 1. In the left plot of Figure 2 we see the main peak in the spectrum centered at $\ell = -3$. We also see some of the wave numbers near $\ell = -3$ growing up from the level of round-off. And we see the wave numbers $\ell = -14, 8$ are also growing. In the right plot of Figure 2, we see that the growth at wave numbers $\ell = -14, 8$ has been suppressed but the growth in the wave numbers flanking the plane wave frequency $\ell = -3$ are still

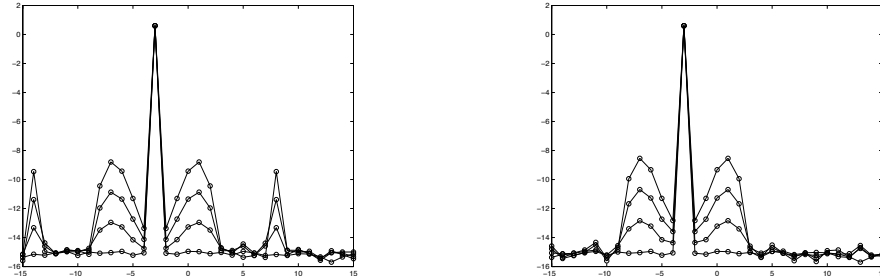


Figure 2: The spectrum is shown for the numerical approximation of the solution (7) at times $t = 0$, $t = 10/32$, $t = 20/32$, and $t = 30/32$. (Left plot: The time step is $k = 1/32$. Right plot: The time step is $k = 1/64$.)

growing. In Figure 3, we plot the spectrum at a fixed time $t = 480/512$

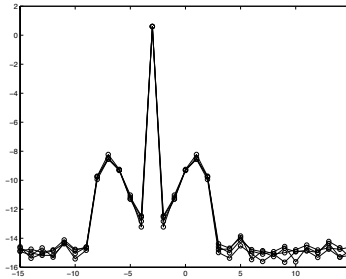


Figure 3: The spectrum is shown for the numerical approximation of the solution (7) at time $t = 480/512$. The spectra are shown from four runs with time steps $k = 1/64$, $k = 1/128$, $k = 1/256$, and $k = 1/512$. The graphs overlap sufficiently that we don't try to distinguish them.

for four computations done with smaller and smaller time steps. We see that the spectra essentially overlap. This suggests that the behaviour near $l = -3$ reflects a real instability rather than a numerical instability. That is; if one were to linearize the focussing Schrödinger equation about the plane wave solution (7) then one will find unstable directions and eigenvalues with positive real parts. Indeed, one can do this and find there is an instability. This particular instability was discovered by Benjamin and Feir and is called the Benjamin-Feir instability or the modulation instability.

Now we would like to test the code against the soliton solutions (5).

Here, we will have some difficulties. First of all, the soliton solution is a “one-hump” solution on \mathbb{R} . It is not periodic in space — if you plot $|u|$ you do not see an infinite array of bumps. In this case, it’s natural to try a soliton with a “hump” at $x = 0$ on the domain $[-L/2, L/2]$ where the domain is much larger than the size of the bump. The logic is that because $|u|$ decays quickly to zero, if the domain is large enough then $|u|$ will be nearly zero at both ends of the interval (which would then make u on $[-L/2, L/2]$ look like it could be one period of a periodic function). And while the bump is moving with finite speed U , it would be a while before this is felt at the endpoints of the interval — for some period of time the solution would continue to “look periodic”. Let’s see how this works in practice.

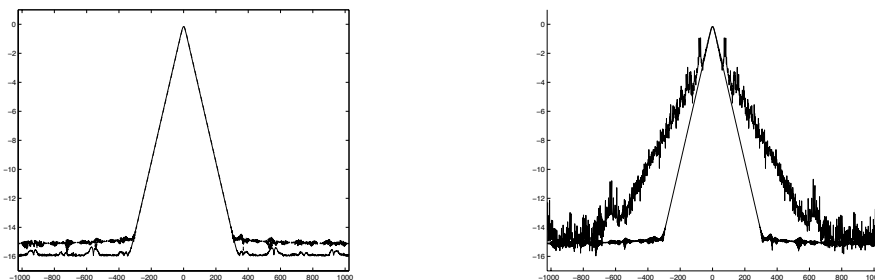


Figure 4: Left plot: the spectrum for the exact solution (8) shown at times $t = 0$ and $t = 1$. The solution has been sampled at 2048 points. The spectra overlap for amplitudes greater than $10e - 12$. The solution at time $t = 0$ has a spectrum that “flattens out” at $10e - 16$, the solution at time $t = 1$ has a spectrum that “flattens out” at $10e - 15$. Right plot: The three schemes are used to compute approximate solutions up to time $t = 1$. The time step is $k = 1/2000$. The spectrum of the exact solution at time $t = 1$ is shown as a guide line.

Consider the soliton

$$u_{\text{sol}}(x, t) = \sqrt{400} \frac{\exp(i(\frac{1}{2}x + (40000 - \frac{1}{4})t))}{\cosh(\sqrt{200}(x - t - \pi))} \quad (8)$$

This is the exact solution (5) with parameters $\alpha = 200$, $U = 1$, $x_0 = \pi$, and $\phi_0 = 0$. We consider this solution for times between $t = 0$ and $t = 1$. At time $t = 0$, we find at the endpoints $|u_{\text{sol}}(0, 0)| = 2.0e - 18$ and $|u_{\text{sol}}(2\pi, 0)| = 2.0e - 18$. At time $t = 1$, we find $|u_{\text{sol}}(2\pi, 1)| = 2.8e - 12$

and $|u_{\text{sol}}(2\pi, 1)| = 1.5e-24$. Note that at $x = 0$ the exact solution is already a little above the level of round-off. This shows up in the spectrum, see the left plot of Figure 4, where we see that the spectrum for the exact solution at time $t = 1$ flattens out at a slightly higher value. In the right plot of Figure 4, we present the spectra of the solutions as computed using the first scheme. The time step is $k = 1/2000$ and the solutions were computed up to time $t = 1$. We see that the spectrum of the approximate solution agrees fairly well with the exact solution for wave numbers $\ell = -60 \dots 60$ but then deviates markedly. In Figure 5 we present the analogous plots as in Figure 4 except we look at the plot of $|u|$. The left plot shows the magnitude of

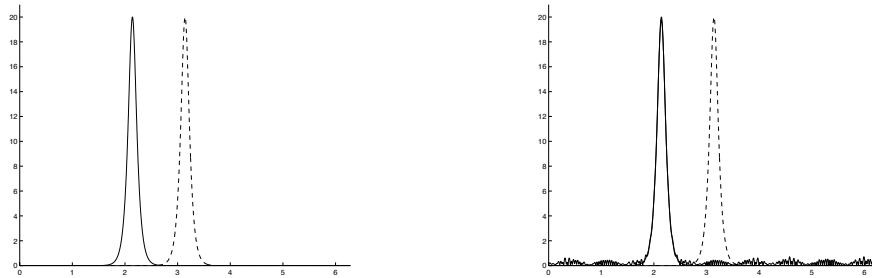


Figure 5: Left plot: The dashed line is the magnitude of the exact solution $|u_{\text{sol}}|$ plotted at time $t = 0$. The solid line is the magnitude at time $t = 1$. Right plot: We use the first scheme to compute up to time $t = 1$ using $N = 2048$ points and time steps of size $k = 1/2000$. The dashed line is the magnitude of the initial data ($|u_{\text{sol}}|$) at time $t = 0$. The solid line is the magnitude of the approximate solution at time $t = 1$.

the exact solution (8) at time $t = 0$ and the solid line shows this magnitude at time $t = 1$. The right plot shows the result of the computation using the first scheme and $N = 2048$ points and time steps $k = 1/2000$. The magnitude of the initial data is plotted with the dashed line and the magnitude of the approximate solution at time $t = 1$ is plotted with a solid line. Note that there are oscillatory pulses in the regions where the exact solution is very close to zero. These pulses do get smaller as the time step is refined however it becomes very CPU-intensive. I'm not an expert in computing solitons, but the above numerical experiments make me think that naive spectral methods are not the way to go. I would consult with an expert before proceeding any further.

None of the above was testing the three schemes head-to-head. We now do this with initial data for which we don't have an exact solution. We consider the focussing case of the equation and take the initial data

$$u_0(x) = \exp(e^{ix}).$$

We take $h = 2\pi/128$ and compute up to time $t_f = 1/10$. The coarsest time-step is $k = 1/100$ and we compute seven approximate solutions, dividing k by two with each computation. Call these solutions u_1, u_2, \dots, u_7 . As before, we monitor the particle number $N(t)$ and the Hamiltonian $H(t)$:

$$\|u_j(\cdot, 1/10) - u_{j+1}(\cdot, 1/10)\|_{L^\infty}, \quad |N(1/10) - N(0)|, \quad |H(1/10) - H(0)|$$

For the first scheme we find:

k	$\ u_j - u_{j+1}\ _{L^\infty}$	ratios	$ N(5) - N(0) $	$ H(5) - H(0) $	ratios
1/100	6.3721e-03	2.0015	-8.8818e-15	2.1151e-01	2.0218
1/200	3.1837e-03	2.0003	2.4869e-14	1.0461e-01	2.0107
1/400	1.5916e-03	2.0000	1.4211e-14	5.2026e-02	2.0053
1/800	7.9581e-04	2.0000	7.1054e-15	2.5944e-02	2.0027
1/1600	3.9791e-04	2.0000	1.7586e-13	1.2955e-02	2.0013
1/3200	1.9896e-04		8.8818e-15	6.4731e-03	2.0007
1/6400			1.5632e-13	3.2355e-03	

We see that the ratios of $\|u_j - u_{j+1}\|$ are going to 2 as expected. The particle number is conserved, as expected. The Hamiltonian is not conserved but as the time step gets smaller the Hamiltonian is more approximately conserved. Note that the Hamiltonian can also be used to test the convergence of the scheme; the ratios of the amount of drift are going to 2.

We now test the second scheme, which is simply the splitting scheme made more accurate via Strang splitting:

k	$\ u_j - u_{j+1}\ _{L^\infty}$	ratios	$ N(5) - N(0) $	$ H(5) - H(0) $	ratios
1/100	4.4332e-04	4.0047	0	2.5412e-03	4.0088
1/200	1.1070e-04	4.0012	2.6645e-14	6.3391e-04	4.0022
1/400	2.7667e-05	4.0003	2.1316e-14	1.5839e-04	4.0005
1/800	6.9163e-06	4.0001	1.2790e-13	3.9592e-05	4.0001
1/1600	1.7290e-06	4.0000	1.7764e-14	9.8977e-06	4.0000
1/3200	4.3226e-07		1.8296e-13	2.4744e-06	4.0000
1/6400			1.3323e-13	6.1860e-07	

We see that the ratios of $\|u_j - u_{j+1}\|$ are going to 4 as expected. The particle number is conserved, as expected. Again, the Hamiltonian is only approximately conserved. As before, the Hamiltonian can also be used to test the convergence of the scheme; the ratios of the amount of drift are going to 4.

Finally, we test the third scheme which is Strang splitting accelerated with Richardson Extrapolation:

k	$\ u_j - u_{j+1}\ _{L^\infty}$	ratios	$ N(5) - N(0) $	ratios	$ H(5) - H(0) $	ratios
1/100	2.8869e-07	16.432	1.2370e-08	32.236	1.3443e-06	15.977
1/200	1.7568e-08	16.197	3.8372e-10	31.609	8.4141e-08	15.988
1/400	1.0846e-09	16.083	1.2140e-11	56.479	5.2628e-09	15.979
1/800	6.7440e-11	16.172	2.1494e-13	5.5000	3.2936e-10	18.646
1/1600	4.1701e-12	11.489	3.9080e-14	.21569	1.7664e-11	56.500
1/3200	3.6295e-13		1.8119e-13	1.4783	3.1264e-13	.061111
1/6400			1.2257e-13		5.1159e-12	

We see that the ratios of $\|u_j - u_{j+1}\|$ are going to 16. These ratios “get bad” once the differences are near round-off error. The particle number is not conserved but as the time step gets smaller the particle number is more approximately conserved. When we use the particle number to test the convergence we see ratios that are close to 32. These ratios “get bad” once the drift is near round-off error. Again, the Hamiltonian is only approximately conserved. As before, the Hamiltonian can also be used to test the convergence of the scheme; the ratios of the amount of drift are going to 16.

3 An Invented Nonlinear Equation

Consider the following integro-differential equation

$$u_t = u_{xx} + 10 \mathcal{H}[u_x] + \frac{1}{100} u^3 u_x \quad (9)$$

on $[0, 2\pi]$ with periodic boundary conditions. Here, \mathcal{H} the Hilbert transform:

$$\mathcal{H}[v](x) := \frac{1}{\pi} \text{P.V.} \int_{-\infty}^{\infty} \frac{v(x-y)}{y} dy.$$

That is, $\mathcal{H}[v]$ is v convolved with $1/(\pi x)$ where the integral is taken in the sense of principal values. In frequency space,

$$u(x) = e^{i\ell x} \implies \mathcal{H}[u](x) = -i \operatorname{sgn}(\ell) e^{i\ell x}$$

where

$$\operatorname{sgn}(\ell) = \begin{cases} 1 & \ell > 0 \\ 0 & \ell = 0 \\ -1 & \ell < 0 \end{cases}$$

This is an equation that I pulled out of thin air. I don't know if it has any exact solutions that I could use to test a code with. So how can I test my code? Checking that I get reasonable ratios when I refine the space step h and time step k isn't sufficient — reasonable ratios suggest that the approximate solutions are converging to something. This is completely different from having evidence that they're converging to the right thing. Where "the right thing" is the solution of equation (9).

While we don't know exact solutions of (9), we do know exact solutions of its linearization. So let's linearize! Take

$$u(x, t) = \bar{u} + \epsilon v(x, t) \quad (10)$$

where \bar{u} is a constant. We know that \bar{u} is a solution of (9) and so if we put (10) into (9) and collect in orders of ϵ , we find

$$v_t = v_{xx} + 10 \mathcal{H}[v_x] + \frac{1}{10} \bar{u}^3 v_x + \mathcal{O}(\epsilon) \quad (11)$$

If we take initial data

$$u_0(x) = \bar{u} + \epsilon \cos(\ell x)$$

then, using the linearized equation (11), we find the solution

$$u(x, t) = \bar{u} + \epsilon e^{(-\ell^2 + 10|\ell|)t} \cos(\ell x + \bar{u}^3 \ell t / 100) + \mathcal{O}(\epsilon^2)$$

This means that if we define

$$u_L(x, t) = \bar{u} + \epsilon e^{(-\ell^2 + 10|\ell|)t} \cos(\ell x + \bar{u}^3 \ell t / 100) \quad (12)$$

then this function should be $\mathcal{O}(\epsilon^2)$ close to the approximate solution of (9).

We now have something we can test against our code. First, we choose values for \bar{u} and ℓ . Let's take $\bar{u} = 2$ and $\ell = 3$. Basically, we want to make sure sure that we choose something other than 1 because if $\bar{u} = 1$ then $\bar{u}^3 = \bar{u}^5$ and we wouldn't notice if there was a bug in the code that put the wrong exponent on \bar{u} .

3.1 Time-stepping

Before testing the code, a quick comment on what type of time-stepping we use. Equation (9) has two linear terms and one nonlinear term. As we know, if we treat the term u_{xx} explicitly, this will introduce a stability constraint on the time-step k . And so an easy first approach would be to treat the linear terms implicitly and the nonlinear term explicitly. This will lead to a local truncation error that is $\mathcal{O}(k^2)$. Even if we treated the linear terms via Crank-Nicolson, the local truncation error would still be $\mathcal{O}(k^2)$ because the nonlinear term is treated explicitly. If we wanted a higher-order scheme, we'd have to use a nonlinear solver at each step. And so, we're stuck with $\mathcal{O}(k^2)$ if we want a simple one-step scheme. To avoid stability constraints, we treat the linear terms implicitly and treat the nonlinear terms explicitly to avoid a nonlinear solve. This results in

$$\hat{u}_\ell^{n+1} = \hat{u}_\ell^n + k \left(-\ell^2 \hat{u}_\ell^{n+1} + 10|\ell| \hat{u}_\ell^{n+1} + \frac{1}{100} (\widehat{(u^n)^3 u_x^n})_\ell \right)$$

that is,

$$\hat{u}_\ell^{n+1} = \frac{\hat{u}_\ell^n + \frac{k}{100} (\widehat{(u^n)^3 u_x^n})_\ell}{1 + k(\ell^2 - 10|\ell|)} \quad (13)$$

And so we do the time-stepping as follows:

1. At time t_n we are in spectral space and have the discrete Fourier coefficients $\{\hat{u}_0^n, \dots, \hat{u}_{N-1}^n\}$. We use these Fourier coefficients to create

$$\hat{v}_k = \begin{cases} ik \frac{2\pi}{L} \hat{u}_k^n & 0 \leq k < N/2 \\ 0 & k = N/2 \\ -i(N-k) \frac{2\pi}{L} \hat{u}_k^n & N/2 < k \leq N-1 \end{cases}$$

which are a good approximation of the discrete Fourier coefficients of u_x^n (if N is large enough for u to be spectrally resolved).

2. Apply the inverse discrete Fourier transform (ifft) to $\{\hat{u}_\ell^n\}$ and $\{\hat{v}_\ell\}$ to recover u^n and u_x^n in physical space. Now apply the nonlinearity, creating

$$w_j = ((\text{ifft}(\hat{u}^n))_j)^3 (\text{ifft}(\hat{v}))_j, \quad j = 0, \dots, N-1$$

which is $(u^n)^3 u_x^n$ in physical space.

3. Apply the discrete Fourier transform (fft) to $\{w_j\}$ creating

$$((\widehat{(u^n)^3 u_x^n})_\ell = \text{fft}(w)_\ell \quad \ell = 0, \dots, N-1$$

Now that we have $((\widehat{(u^n)^3 u_x^n})_\ell$ we can take a time step via (13)

3.2 Convergence Test

We consider equation (9) with initial data

$$u_0(x) = 2 + \frac{1}{10} \cos(3x).$$

The initial time is $t_0 = 0$ and the final time is $t_f = 1e-6$. We take $h = 2\pi/64$ and $k = 5e-8$. Computing in this way, we find an approximate solution u_1 . We then divide k by 2 and compute again, creating approximate solution u_2 . In this way, we create six approximate solutions. We then find the L^∞ norm of their difference at the final time t_f :

	difference	ratios
$\ u_1(\cdot, t_f) - u_2(\cdot, t_f)\ _{L^\infty}$	5.5111e-13	1.9952
$\ u_2(\cdot, t_f) - u_3(\cdot, t_f)\ _{L^\infty}$	2.7622e-13	2
$\ u_3(\cdot, t_f) - u_4(\cdot, t_f)\ _{L^\infty}$	1.3811e-13	2.0065
$\ u_4(\cdot, t_f) - u_5(\cdot, t_f)\ _{L^\infty}$	3.4195e-14	2.2628
$\ u_5(\cdot, t_f) - u_6(\cdot, t_f)\ _{L^\infty}$	7.9936e-15	

We see that as k decreases, the ratios are going to 2. However, for k very small, the differences continue to decrease but the ratios aren't getting closer to 2 anymore. What we are observing is that we have the expected local truncation error $\mathcal{O}(k^2)$ but at some point round-off error is entering into things.

3.3 Linear Test

We now want to test the computed solutions against the linearization. To do this, we want to take h and k sufficiently small so that we can be confident that the computed solutions “have converged” to whatever it is that they are going to converge to. We now turn to the question of, “The thing they've converged to, is it $\mathcal{O}(\epsilon^2)$ close to the function (12)?”

We take the same initial data as above, the same initial time, and the same final time.

To see if h is small enough, we plot the spectrum at the final time. Doing this, we find that the physical spectrum is $[0, \ell_{\max}]$ where $\ell_{\max} = 12$. Whether or not this means h is small enough really involves getting into the questions of aliasing error caused by nonlinearities. These questions were raised and then dodged earlier. In any case, for our purposes here, h is small enough.

To see if k is small enough, we look at the differences above. We have two choices: take the smallest value of k because the differences are smallest or take $k = 1.25e - 08$ because this led to a ratio that was closest to 2. Which one we choose won't make a real difference; we'll take $k = 1.25e - 08$.

Now, we take initial data

$$u_0(x) = 2 + \epsilon \cos(3x)$$

for $\epsilon = 1, 1/2, 1/4, 1/8, 1/16, 1/32$. For each value of ϵ there is a corresponding, exact solution of the linearized problem (12); call it u_{exact} . For example, for $\epsilon = 1$ we generate u_1 and $u_{1\text{exact}}$. We will look at the L^∞ norm of the difference between the approximate solution at the final time and the corresponding, exact solution of the linearized problem:

$$\|u_1(\cdot, t_f) - u_{1\text{exact}}(\cdot, t_f)\|_{L^\infty}.$$

If the approximate solutions are converging to the correct solution then this difference should be $\mathcal{O}(\epsilon^2)$ as $\epsilon \rightarrow 0$.

	difference	ratios
$\ u_1(\cdot, t_f) - u_{1\text{exact}}(\cdot, t_f)\ _{L^\infty}$	2.5358e-07	4.7463
$\ u_2(\cdot, t_f) - u_{2\text{exact}}(\cdot, t_f)\ _{L^\infty}$	5.3426e-08	4.3526
$\ u_3(\cdot, t_f) - u_{3\text{exact}}(\cdot, t_f)\ _{L^\infty}$	1.2275e-08	4.1765
$\ u_4(\cdot, t_f) - u_{4\text{exact}}(\cdot, t_f)\ _{L^\infty}$	2.9390e-09	4.0880
$\ u_5(\cdot, t_f) - u_{5\text{exact}}(\cdot, t_f)\ _{L^\infty}$	7.1892e-10	4.0435
$\ u_6(\cdot, t_f) - u_{6\text{exact}}(\cdot, t_f)\ _{L^\infty}$	1.7780e-10	

We see that as ϵ decreases by factors of 2, the difference between the approximate solution of the nonlinear problem and the exact solution of the linearized problem decreases by factors of 4. This shows that the difference is $\mathcal{O}(\epsilon^2)$, as desired.

While this doesn't prove anything, it's strong evidence that the code is correct. This type of test is something you can apply to any nonlinear problem as long as you have a linear approximation of the problem that you can solve exactly.