Karlsruher Institut für Technologie

Department of Mathematics
Institute for Applied and
Numerical Mathematics

JProf. Dr. Katharina Schratz
Dr. Patrick Krämer
Dr. Simon Baumstark

Numerical Integrators for Nonlinear Dispersive Equations — Exercise Sheet 02 November 27, 2018

In view of constructing numerical integrators for the nonlinear Schrödinger equation

$$i\partial_t u = -\Delta u + f(u), \quad u(0,x) = u^0(x) \in \mathbb{C}, \quad (t,x) \in [0,T] \times \mathbb{T}^d, \qquad \text{(NLS)}$$

for $t \in [0,T]$ and $x \in [-\pi, \pi] =: \mathbb{T}$ with polynomial nonlinearity $f(u)$, equipped with periodic boundary conditions, i.e.

$$u(t,-\pi) = u(t,+\pi), \quad \partial_x^m u(t,-\pi) = \partial_x^m u(t,+\pi), \quad \text{for } m \in \mathbb{N} \qquad \text{for all } t \in [0,T],$$

the aim of this exercise sheet is to discuss the discretization of the spatial operators ∂_x^m with so-called Fourier pseudospectral (FP) methods. We look for solutions $u(t,\cdot)$ of (NLS) in Sobolev spaces $H^r(\mathbb{T})$, $r \geq 0$ equipped with the norm

$$\|u\|_{H^r}^2 = \sum_{k\in\mathbb{Z}} |\,(1+|k|)^r\, \widehat{u_k}\,|^2, \quad \text{for all } u \in H^r(\mathbb{T}), \quad \text{(the coefficients } \widehat{u_k} \text{ are defined below).}$$

Note that all $u \in H^r(\mathbb{T})$ satisfy periodic boundary conditions on $\mathbb{T}$. It is well known that the set $\{e^{ikx}, k \in \mathbb{Z}\}$ forms an infinite basis in the spaces $H^r(\mathbb{T})$, $r \geq 0$, such that all $w \in H^r(\mathbb{T})$ have the Fourier series representation

$$w(x) = \sum_{k\in\mathbb{Z}} \widehat{w_k} e^{ikx}, \qquad \widehat{w_k} = \frac{1}{2\pi}\int_{\mathbb{T}} w(x)e^{-ikx}dx = \frac{1}{2\pi}\int_{-\pi}^{\pi} w(x)e^{-ikx}dx. \qquad (1)$$

Exercise 3: (Lie Splitting scheme for Nonlinear Schrödinger)

In the following, assume that for all $t \in [0,T]$ the function

$$u(t,x) := \sum_{k\in\mathbb{Z}} \widehat{u_k}(t) e^{ikx} \qquad (2)$$

solves the nonlinear Schrödinger equation (NLS) with cubic nonlinearity $f(u) = |u|^2 u$.

a) From the ansatz (2), derive a representation of the Laplace operator Δ applied to u.

In order to construct splitting schemes for solving (NLS), we split the problem into the subproblems

$$i\partial_t w = -\Delta w, \quad w(0,x) = w^0(x) \quad (S1), \qquad \text{and} \qquad i\partial_t z = |z|^2 z, \quad z(0,x) = z^0(x) \quad (S2).$$

b) Exploit the results from part a), in order to derive an ordinary differential equation from $(S1)$ for each Fourier coefficient $\widehat{w_k}(t)$ depending on time t for all $k \in \mathbb{Z}$. From these ODEs deduce an explicit Fourier representation for the exact flow of $(S1)$

$$\varphi_T^\tau(w^0(x)) = w(t,x).$$

c) In $(S2)$ compute the time derivative $\partial_t |z(t,x)|^2$ of the modulus of z. From your result, deduce an explicit representation for the exact flow of $(S2)$

$$\varphi_P^\tau(z^0(x)) = z(t,x).$$

d) Let $r > \frac{d}{2}$. Show that for $u, v \in H^r$ we have $\|uv\|_r \leq K(r,d)\,\|u\|_r\,\|v\|_r$, where $K(r,d)$ only depends on r and d.
Hint: *Exploit (without proof) that for all* $k \in \mathbb{Z}^d$ *we have* $\sum_{\ell\in\mathbb{Z}^d} \frac{(1+|k|^2)^r}{(1+|k-\ell|^2)^r(1+|\ell|^2)^r} \leq K(r,d)$.

e) Write down a Lie splitting time integration scheme Φ_{Lie}^τ for solving the cubic (NLS).

f) Sketch the proof for the first order convergence of Φ_{Lie}^τ in H^r, $r > d/2$, for initial data $u^0 \in H^{r+2}$.

next page →

The Fourier Pseudospectral (FP) Space Discretization (based on [Faou2012, Tref2000])

In the following, let u be smooth and periodic on the interval $[-\pi, \pi]$. Furthermore let $N \in \mathbb{N}$ be even and let a discretization of the interval $[-\pi, \pi]$ be defined through

$$x_j = jh, \; j = -N/2, \ldots, N/2, \qquad \text{with} \qquad h = (2\pi)/N.$$

In order to discretize spatial differential operators, the idea is to use a **trigonometric interpolation polynomial** $t_N(x)$ to approximate $u(x)$ with interpolation property in the grid points x_j, i.e.

$$t_N(x_j) = u(x_j), \qquad j = -N/2, \ldots, N/2.$$

Let $U \in \mathbb{C}^N$ with $U_j = u(x_j), j = -N/2+1, \ldots, N/2$. We define the following trigonometric polynomial

$$t_N(x) = \frac{1}{2N}\left(\widehat{U}_{-N/2}e^{-ixN/2} + \widehat{U}_{N/2}e^{ixN/2}\right) + \frac{1}{N}\sum_{k=-N/2+1}^{N/2-1} \widehat{U}_k e^{ikx} \quad \text{for all } x \in [-\pi, \pi]$$

where $\widehat{U} = \mathcal{F}_N U = \left(\widehat{U}_k\right)_{k=-N/2+1}^{N/2}$ is the **discrete Fourier transform** of U, defined via

$$\widehat{U}_k := \sum_{j=-N/2+1}^{N/2} U_j e^{-ijx_k} \quad \text{for all } k = -N/2+1, \ldots, N/2.$$

Remark: *Note that $\widehat{U}_k$ can also be interpreted as an approximation to the Fourier coefficient $\widehat{u_k}$ via the application of the trapezoidal quadrature rule with nodes x_j to the integral in* (1).

<u>**Exercise 4:**</u> (Trigonometric Interpolation Polynomial)

a) Show that for all $k = -N/2, \ldots, N/2$ we have that

$$\hat{U}_{-N/2}e^{-ix_kN/2} = \hat{U}_{N/2}e^{ix_kN/2}.$$

b) Exploiting part a), show that t_N satisfies the interpolation property, i.e. show that

$$t_N(x_j) = u(x_j) \qquad \text{for all } j = -N/2, \ldots, N/2.$$

In particular, with the aid of Exercise 7, we define the **inverse discrete Fourier transform** $\mathcal{F}_N^{-1}\hat{U}$ of $\hat{U}$ through

$$\left(\mathcal{F}_N^{-1}\hat{U}\right)_j := \frac{1}{N}\sum_{k=-N/2+1}^{N/2} \hat{U}_k e^{ikx_j} = t_N(x_j), \qquad j = -N/2+1, \ldots, N/2.$$

Based on the trigonometric polynomial t_N and the discrete Fourier transform above, we numerically approximate the m-th spatial derivative of u, i.e. $\partial_x^m u(x_j)$, in the grid points $x_j, j = -N/2+1, \ldots, N/2$ by

$$\partial_x^m t_N(x_j) = t_N^{(m)}(x_j) = \mathcal{F}_N^{-1}\left((i\tilde{k})^m \cdot \mathcal{F}_N U\right)_j, \quad j = -N/2+1, \ldots, N/2,$$

where (see remark in the footnote on the ordering in MATLAB and Python)

$$\tilde{k} = [-N/2+1, \ldots, N/2-1, \chi], \quad \chi = \begin{cases} N/2, & m \text{ even}, \\ 0, & m \text{ odd}. \end{cases}$$

Note that the theory on spectral derivatives also works for periodic functions on arbitrary intervals $[a, b] \subset \mathbb{R}$.

c) Determine a linear transformation of the interval $[-\pi, \pi]$ to the interval $[a, b] \subset \mathbb{R}$. Which **additional factor** in the coefficients $\tilde{k}$ is then necessary in order to correctly apply the spectral scheme as described?

Remark: *Note that also for higher spatial dimensions $d \geq 1$, i.e. $x \in \mathbb{T}^d$, this scheme satisfies error bounds which only depend on the regularity of u and on the number of grid points $N \in \mathbb{N}$, i.e. for $s' - s > d/2$ we have*

$$\|\partial_x^m u - \partial_x^m t_N\|_{H^r} \leq \|u - t_N\|_{H^{r+m}} \leq K \cdot N^{-s}\|u\|_{H^{r+m+s'}}, \qquad u \in H^{r+m+s'}(\mathbb{T}^d).$$

next page →

Note that the Fourier numbers are order differently in MATLAB and Python as follows $\tilde{k} = [0, \ldots, N/2-1, \chi, -N/2+1, \ldots, -1]$.

[Faou2012] - E. Faou, *Geometric numerical integration and Schrödinger equations*, EMS 2012
[Tref2000] - L. Trefethen, *Spectral methods in MATLAB*, SIAM 2000

Programming Exercise 2: (Implementation of the FP Discretization Scheme)

Consider $u(x) := \exp(\sin(x))$ on the interval $x \in [-\pi, \pi]$ and consider $N_x = 16$ grid points at first.

a) Implement the spectral space discretization scheme to compute an approximation to the second derivative $u''(x)$. Plot the numerical result together with the exact derivative.

   Hint: Use the MATLAB built-in functions `fft` and `ifft`, and the Python functions `numpy.fft.fft`, `numpy.fft.ifft` respectively. Note that MATLAB and also Python orders the Fourier numbers as follows

   $$\tilde{k} = [0, \dots, N_x/2 - 1, \chi, -N_x/2 + 1, \dots, -1] \qquad \text{(see also [Tref2000])}.$$

b) Add a finite difference (FD) approximation to u'' to your plot. What do you observe?

   ***Recall** that, a FD approximation to u'' in the grid points $x_j, j = -N_x/2+1, \dots, N_x/2$ is obtained by computing the matrix vector product AU, where the matrix A is defined via $A := \frac{1}{h^2}\operatorname{tridiag}(1, -2, 1)$, with additional nonzero entries $A_{N_x,1} = \frac{1}{h^2} = A_{1,N_x}$ due to p.b.c., and where the vector U is a vector with entries $U_j := u(x_j)$, $j = -N_x/2+1, \dots, N_x/2$.*

c) Create an order plot for the spatial accuracy of the FP and the FD method using $N_x^\ell = 8 \cdot 2^\ell$, $\ell = 0, \dots, 9$ grid points. Compute the corresponding errors in the approximate L^2 norm, i.e.

   $$\text{err}_\ell = \sqrt{h_\ell}\,\left\| u''_{num} - u''_{exact} \right\|_2 .$$

d) Repeat all the steps for the function $g(x) = 1/\cosh(x)$ on the interval $x \in [-\pi, \pi]$. What can you observe?

   How do your results change if we consider g on $x \in [-4\pi, 4\pi]$ instead? Can you give an explanation?

   Hint: $\cosh(x) = (e^x + e^{-x})/2$, $\quad \frac{d}{dx}\cosh(x) = \sinh(x)$, $\quad \cosh(x)^2 = 1 + \sinh(x)^2$.

Programming Exercise 3: (Implementation of a Lie Splitting scheme for NLS)

In this programming exercise, we construct a splitting scheme for solving the nonlinear Schrödinger equation with cubic nonlinearity, i.e.,

$$i\partial_t u(t,x) = -\Delta u(t,x) + \lambda |u(t,x)|^2 u(t,x), \quad u(0,x) = u^0(x), \quad \lambda \in \mathbb{R} \tag{4}$$

on the torus $\mathbb{T} = [-\pi, \pi]$ (i.e. periodic boundary conditions) and for $t \in [0, T]$.

In the following let $T = 1$ and let

$$u^0(x) := \cos(x).$$

For practical implementation issues, we consider the discretization of space $x_j = jh$, $j = -\frac{N_x}{2} + 1, \dots \frac{N_x}{2}$ with $N_x = 64$, i.e. $h = \frac{2\pi}{N_x}$, and of time $t_n = n\tau$, $n = 0, 1, 2, \dots \lfloor T/\tau \rfloor =: N_T$ for time step sizes

$$\tau = \tau_m = \frac{T}{N_T^m}, \quad \text{where} \quad N_T^m, m = 1, 2, \dots, m_{\max}$$

is the m-th divisor of the number $N_T^{m_{\max}} = 120$. All divisors of a number $N \in \mathbb{N}$ can be found in MATLAB (since version R2014b) via the function `divisors()`.

Reference scheme:

Because for the cubic NLS on the torus $\mathbb{T}$, we do not have an exact solution available, we need to compute a reference solution in order to test our schemes. Thus, before performing the actual time stepping with step size τ_m, $m = 1, 2, \dots, m_{\max}$, we carry out the time integration over the full interval $t \in [0, T]$ with a suitable **reference scheme** using a smaller time step size $\tau_{\text{ref}} := \frac{\min_m \tau_m}{M_{\text{ref}}} \ll \tau_m$ for $M_{\text{ref}} \in \mathbb{N}$. Since we choose N_T^m to be a divisor of $N_T^{m_{\max}}$ and due to $\tau_m = \frac{T}{N_T^m}$, we have that

$$\frac{\tau_m}{\tau_{m_{\max}}} = \frac{\tau_m}{\min_m \tau_m} = \frac{N_T^{m_{\max}}}{N_T^m} \in \mathbb{N} \qquad \text{for all } m = 1, 1, 2, \dots, m_{\max}. \tag{5}$$

In the case of the (4), a suitable scheme in order to test the Lie splitting scheme Φ^τ_{Lie} from Exercise 3 is for example the so-called *Strang splitting* scheme which is defined as

$$u^{n+1} = \Phi^\tau_{\text{Strang}}(u^n) = \left(\varphi_T^{\tau/2} \circ \varphi_P^\tau \circ \varphi_T^{\tau/2}\right)(u^n).$$

(a) Building up on your code from Programming Exercise 1, combine and implement the Lie splitting scheme Φ^{τ}_{Lie} from Exercise 3 with the FP space discretization scheme from Programming Exercise 3 in order to approximate the solution of the nonlinear Schrödinger equation (4) with $\lambda = 1$.

(b) α) Animate your numerical solution over time for time step size $\tau = \tau_{m_{\max}} = \frac{1}{120}$, i.e., after each iteration with your scheme Φ^{τ}_{Lie} plot the spatial distribution u^n_{LIE} on the whole interval $\mathbb{T} = [-\pi, \pi]$ using the `plot` command and the command `drawnow`.

β) Now set $\lambda = 0$ and animate the Fourier *actions* of the numerical solution, i.e., the distribution of the modulus $|\left(\widehat{u^n_{\text{LIE}}}\right)_k|^2$ of the Fourier coefficients of the numerical solution at time t_n against the corresponding Fourier numbers $k = -N_x/2 + 1, \dots N_x/2$. How do your results change, if you set $\lambda = 1$ again?

In the following fix $\lambda = 1$.

(c) Implement the Strang splitting scheme $\Phi^{\tau_{\text{ref}}}_{\text{Strang}}$ with $\tau_{\text{ref}} := \frac{\min_m \tau_m}{M_{\text{ref}}}$ for $M_{\text{ref}} = 200$ as a reference method into the code from part a) and b). Note that you **only need to save** the numerical approximations u^n_{ref} at times $t_n = n\tau_{m_{\max}}$ due to the relation (5). The corresponding time grid then covers also the coarser time grids corresponding to τ_m for all $1 \le m \le m_{\max}$.

(d) Proceed as in Programming Exercise 1 in order to create order plots for the numerical approximations u^n_{Lie} at times $t_n = n\tau_m$, obtained with the scheme Φ^{τ}_{Lie}.
Note, that we measure the maximal errors over the whole time interval $[0, T]$ in the approximate H^r norm (choose $r = 1 > d/2$, $d = 1$), i.e., corresponding to the numerical solution obtained with time step size τ_m, we compute

$$\text{err}^m_{\text{Lie}} = \frac{\sqrt{2\pi}}{N_x} \max_{t_n \in [0,T]} \left\| (1 + |k|)^1 \cdot \texttt{fft}(u_{\text{ref}}(t_n) - u^n_{\text{LIE}}) \right\|_{\ell^2}.$$

Then, the time step τ_m is `loglog`-plotted against the corresponding error $\text{err}^m_{\text{Lie}}$.

(e) In order to test the reference Strang splitting scheme, also create order plots for $\Phi^{\tau}_{\text{Strang}}$ into the same plots from part d).

Discussion in the problem class monday 8:00 am, in room 3.061 in the Kollegiengebäude Mathematik 20.30.