In view of constructing numerical integrators for the Klein–Gordon equation

\[
\frac{\partial u(t, x)}{\partial t} = \frac{\partial^2}{\partial x^2} \left[ \Delta + c^2 \right] u(t, x) + 2c^2 f(u(t, x)), \quad u(0, x) = \varphi(x), \quad \frac{\partial u(0, x)}{\partial t} = c^2 \gamma(x), \tag{1}
\]

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

\[
z(t, -\pi) = z(t, +\pi), \quad \frac{\partial^m}{\partial x^m} z(t, -\pi) = \frac{\partial^m}{\partial x^m} z(t, +\pi), \quad \text{for } m \in \mathbb{N} \quad \text{for all } t \in [0, T],
\]

the aim of this exercise sheet is to discuss the discretization of the spatial operators \( \frac{\partial^m}{\partial x^m} \) and in particular

\[
\langle \nabla \rangle_c := \sqrt{-\Delta + c^2} \tag{2}
\]

with so-called Fourier pseudospectral (FP) methods. We look for solutions \( z(t, \cdot) \) of (1) in Sobolev spaces \( H^r(\mathbb{T}) \), \( r \geq 0 \) equipped with the norm

\[
\| u \|_{H^r} = \sum_{k \in \mathbb{Z}} \left( 1 + |k|^2 \right)^{r} \| \hat{u}_k \|, \quad \text{for all } u \in H^r(\mathbb{T}), \quad (\text{the coefficients } \hat{u}_k \text{ are defined below}).
\]

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 \( u \in H^r(\mathbb{T}) \) have the Fourier series representation

\[
u(x) = \sum_{k \in \mathbb{Z}} \hat{u}_k e^{ikx}, \quad \hat{u}_k = \int_{\mathbb{T}} \mu(x)e^{-ikx}dx = \int_{-\pi}^{\pi} \mu(x)e^{-ikx}dx. \tag{3}
\]

**Exercise 6:** (Klein–Gordon in Fourier space)

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

\[
z(t, x) := \sum_{k \in \mathbb{Z}} \hat{z}_k(t) e^{ikx} \tag{4}
\]

solves the Klein–Gordon equation (1) for fixed \( c \geq 1 \), i.e. we set \( f(z) = z \).

a) From the ansatz (4), derive an ordinary differential equation for each Fourier coefficient \( \hat{z}_k(t) \) depending on time \( t \) for all \( k \in \mathbb{Z} \).

b) Exploit the results from part a) in order to motivate the definition of \( \langle \nabla \rangle_c = \sqrt{-\Delta + c^2} \) via its Fourier representation

\[
\langle \nabla \rangle_c u(x) := \sum_{k \in \mathbb{Z}} \langle k \rangle_c \hat{u}_k e^{ikx}, \quad \langle k \rangle_c = \sqrt{|k|^2 + c^2}, \quad u \in H^1(\mathbb{T}).
\]

c) **Advanced:** Now let \( f(z) = |z|^2 z \) be a cubic nonlinearity. Exploit ansatz (4) and determine the \( k \)-th Fourier coefficient \( \hat{f}_k(t) \) of \( f(z(t, x)) \) by considering

\[
f(z(t, x)) = f \left( \sum_{k \in \mathbb{Z}} \hat{z}_k(t)e^{ikx} \right) = \sum_{k \in \mathbb{Z}} \hat{f}_k(t)e^{ikx}.
\]

Note that for \( k \in \mathbb{Z} \) the coefficient \( \hat{f}_k(t) \) depends on the coefficients \( \hat{z}_\ell(t), \ell \in \mathbb{Z} \), of \( z(t, x) \).
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

\[
    x_j = jh, \quad j = -N/2, \ldots, N/2, \quad \text{with} \quad h = (2\pi) / N.
\]

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.

\[
    t_N(x_j) = u(x_j), \quad j = -N/2, \ldots, N/2.
\]

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

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

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

\[
    \hat{U}_k := \frac{1}{N} \sum_{j=-N/2+1}^{N/2} U_j e^{-ijx} \quad \text{for all} \quad k = -N/2 + 1, \ldots, N/2.
\]

**Remark:** Note that \( \hat{U}_k \) can also be interpreted as an approximation to the Fourier coefficient \( \hat{u}_k \) via the application of the trapezoidal quadrature rule with nodes \( x_j \) to the integral in (3).

**Exercise 7:** (Trigonometric Interpolation Polynomial)

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

\[
    \hat{U}_{-N/2} e^{-ixN/2} = \hat{U}_{N/2} e^{ixN/2}.
\]

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

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

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

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

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

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

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

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

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

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

**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

\[
    \| \partial_x^m u - \partial_x^m t_N \|_{H^r} \leq \| u - t_N \|_{H^{r+s}} \leq K \cdot N^{-s} \| u \|_{H^{r+s}} \quad u \in H^{r+s+\delta}(\mathbb{T}^d).
\]

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


next page →
Programming Exercise 3: (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

\[ k = [0, \ldots, N_x/2 - 1, -N_x/2 + 1, \ldots, -1] \quad (\text{see also [Tref2000]}) \]

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

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

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

\[ \text{err}_\ell = \sqrt{\frac{h^2}{k}} \| u''_{\text{num}} - u''_{\text{exact}} \|_2. \]

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

Programming Exercise 4: (Implementation of an Exponential-type FP Time Integration Scheme for KG)

In this programming exercise, we construct an exponential integrator for the linear Klein–Gordon equation

\[
\partial_t^2 z(t,x) = -c^2 \left( \nabla \right)_c^2 z(t,x) + c^2 \lambda z(t,x), \quad z(0,x) = \varphi(x), \quad \partial_t z(0,x) = c \langle \nabla \rangle_c \gamma(x),
\]

with \( \lambda = \frac{1}{2} \) on the torus \( T = [-\pi, \pi] \) (i.e. periodic boundary conditions) and for \( t \in [0,T] \).

In the following let \( T = 1 \) and let
\[
\varphi(x) := \frac{\cos(x) - 2 \sin(2x)}{2 - \sin(x)} \quad \text{and} \quad \gamma(x) := \frac{\sin(x) - 2 \cos(2x)}{2 - \cos(x)}.
\]

For practical implementation issues we consider the discretization of time \( t_n = n \tau \), \( n = 0, 1, 2, \ldots \) \( |T/\tau| =: N_T \) and of space \( x_j = jh, \ j = -N_x/2 + 1, \ldots, N_x/2 \) with \( N_x = 128 \) and for
\[
\tau = \tau_m = \frac{T}{N_T}, \quad N_T^m := 2^m, \quad m = 1, 2, \ldots, 8 \quad \text{and} \quad h = \frac{2\pi}{N_T^m}.
\]

(a) Building up on your code from Programming Exercise 2, combine and implement the exponential integrator \( \Phi^\ell_{\text{exp}} \) from Exercise Sheet 2 with the FP space discretization scheme from Programming Exercise 3 in order to approximate the solution of the linear KG equation (6).

(b) Analogously to the previous exercise sheets, the exact solution of (6) is given by

\[
z(t,x) = \cos \left( t \Omega_c \right) \varphi(x) + \frac{\sin \left( t \Omega_c \right)}{t \Omega_c} c \langle \nabla \rangle_c \gamma(x), \quad \Omega_c = c \sqrt{\langle \nabla \rangle_c^2 - \lambda}
\]

Animate the numerical solution and the exact solution together in one plot for all times \( t_n \) corresponding to the time step \( \tau_0 \) and for \( c = 1 \) and \( c = 8 \).

(c) Proceed according to Programming Exercise 2b) and for \( c_{\ell} = 1.7^\ell, \ell = 0, 1, \ldots, 10 \) create order plots for both the convergence in the time step size \( \tau \) and in the parameter \( c \) of the scheme \( \Phi^\ell_{\text{exp}} \) corresponding to the time step sizes \( \tau_m \), \( m = 1, \ldots, 8 \) considering the maximal error in the discrete \( L^2 \) norm (cf. Programming Exercise 3c)), i.e.

\[
\text{err}_{\text{max}} = \max_{t_n \in [0,T]} \| z(t_n, \cdot) - z^\ell \|_{L^2}.
\]

**Hint:** For each combination of time step \( \tau_m \) and parameter \( c_{\ell} \) save the corresponding \( \text{err}_{\text{max}} \) in a two-dim. array.

Discussion in the problem class friday 11:30 am, in room 2.058 in the Kollegiengebäude Mathematik 20.30.