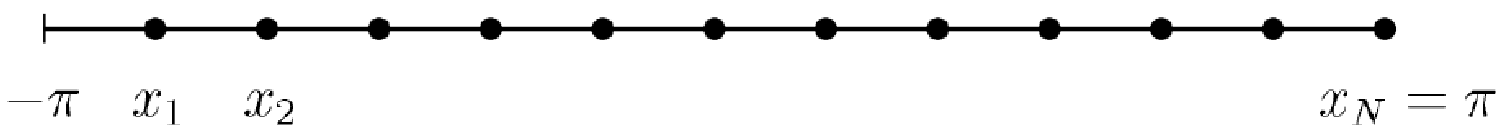
Second-order approximation

Let us calculate numerically the derivative of a periodic function $u(x)$ (such that $u(x) = u(x + 2\pi)$). Suppose that we know the values of this function $\{u_1, ..., u_N\}$ sampled on a uniform grid $\{x_1, ..., x_N\}$ with a constant step $x_{j+1} - x_j = h$.



The central finite difference scheme

$$w_j = \frac{u_{j+1} - u_{j-1}}{2h}. \tag{1}$$

with $u_0 = u_N$ and $u_1 = u_{N+1}$ for the periodicity, is second-order accurate:

$$u'(x_j) = w_j + Ch^2, \tag{2}$$

where C is a constant independent of h.

Second-order approximation

Using matrix notation,

$$w = Du,$$

$$D = h^{-1}\begin{pmatrix} 0 & \frac{1}{2} & & & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \ddots & & \\ & & \ddots & & \\ & & \ddots & 0 & \frac{1}{2} \\ \frac{1}{2} & & & -\frac{1}{2} & 0 \end{pmatrix}.$$

This matrix is circulant tridiagonal, its elements a_{ij} only depend on $(i-j) \mod N$. It can be derived using local interpolation,

For $j = 1, 2, ..., N$

- Let p_j be a polynomial of degree ≤ 2 such that $p_j(x_{j-1}) = u_{j-1}$, $p_j(x_j) = u_j$ and $p_j(x_{j+1}) = u_{j+1}$.
- Assign $w_j = p'_j(x_j)$.

Fourth-order approximation

Similarly, it is possible to derive higher-order finite-difference schemes. For example, the following scheme is 4th order accurate:

For $j = 1, 2, ..., N$

- Let p_j be a polynomial of degree ≤ 4 such that $p_j(x_{j\pm2}) = u_{j\pm2}$, $p_j(x_{j\pm1}) = u_{j\pm1}$ and $p_j(x_j) = u_j$.
- $w_j = p_j'(x_j)$.

We obtain a circulant five-diagonal matrix,

$$D = h^{-1} \begin{pmatrix} & & \ddots & & & \frac{1}{12} & -\frac{2}{3} \\ & & \ddots & -\frac{1}{12} & & & \frac{1}{12} \\ & & \ddots & \frac{2}{3} & \ddots & & \\ & & \ddots & 0 & \ddots & & \\ & & \ddots & -\frac{2}{3} & \ddots & & \\ -\frac{1}{12} & & & \frac{1}{12} & \ddots & & \\ \frac{2}{3} & -\frac{1}{12} & & & \ddots & & \end{pmatrix}.$$

Spectral method

Higher-order methods have wider stencil, i.e., their matrices have more non-zero elements. Spectral collocation methods use global interpolation, therefore, the matrices are full.

- Let p be a function (independent of j) such that $p(x_j) = u_j$, $\forall j$.
- $w_j = p'(x_j)$.

For example, Fourier spectral methods use trigonometric polynomials.
The derivative matrix is as follows:

$$D = \begin{pmatrix} & \vdots & \\ \ddots & \frac{1}{2}\cot\frac{3h}{2} & \\ \ddots & -\frac{1}{2}\cot\frac{2h}{2} & \\ \ddots & \frac{1}{2}\cot\frac{1h}{2} & \\ & 0 & \\ & -\frac{1}{2}\cot\frac{1h}{2} & \ddots \\ & \frac{1}{2}\cot\frac{2h}{2} & \ddots \\ & -\frac{1}{2}\cot\frac{3h}{2} & \ddots \\ & \vdots & \end{pmatrix}.$$

Example computer program

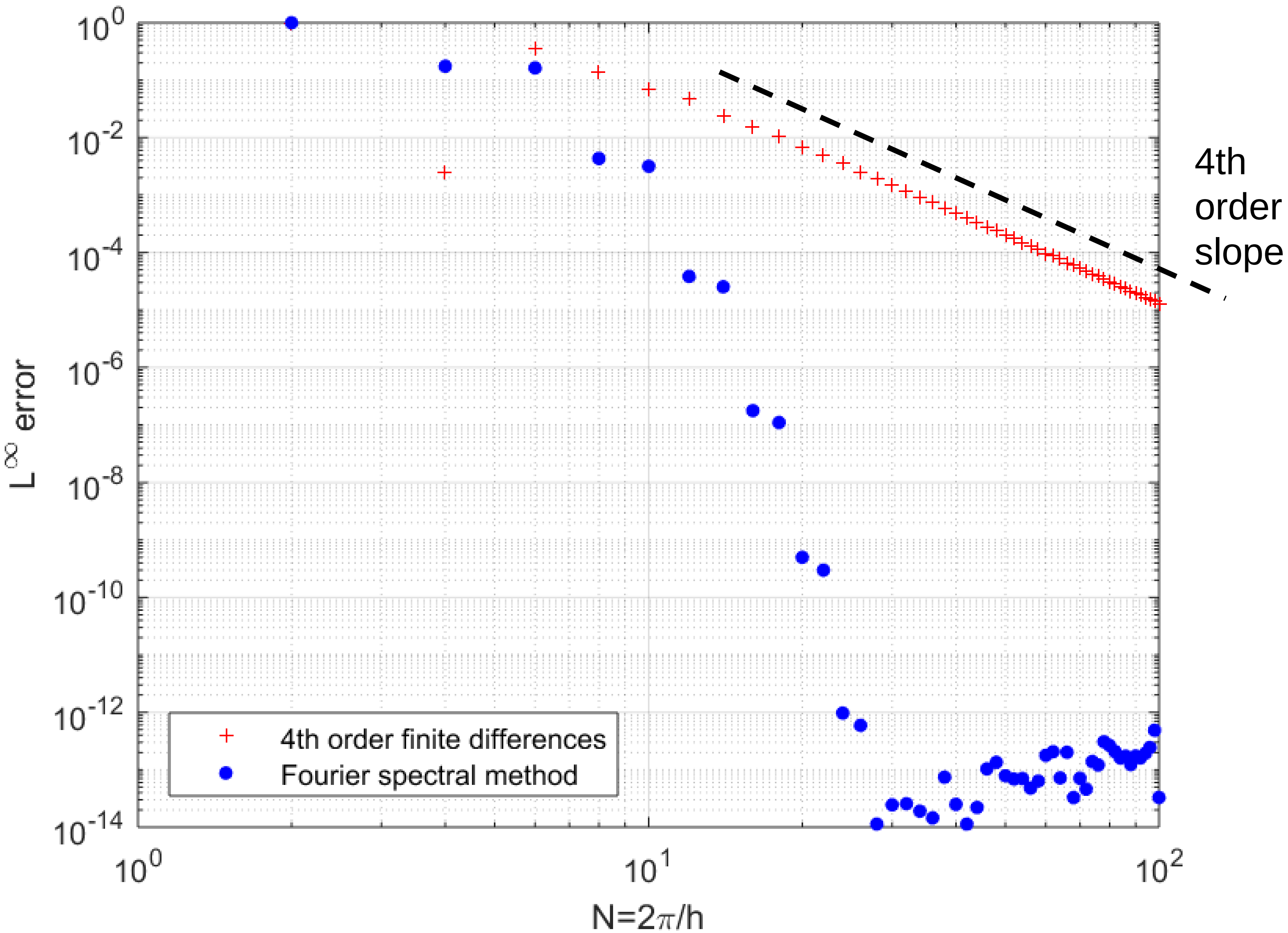
Example. $u(x) = e^{\sin x}$, $u'(x) = e^{\sin x}\cos x$.

```
clearvars;
Nvec = 2:2:100;
for it = 1:length(Nvec)
N = Nvec(it);
h = 2*pi/N; x = -pi + (1:N)'*h;
u = exp(sin(x)); uprime = cos(x).*u;

e = ones(N,1);
D = sparse(1:N,[2:N 1],2*e/3,N,N)...
- sparse(1:N,[3:N 1 2],e/12,N,N);
D4 = (D-D')/h;
error4(it) = norm(D4*u-uprime,inf);

column = [0 .5*(-1).^(1:N-1).*cot((1:N-1)*h/2)];
DF = toeplitz(column,column([1 N:-1:2]));
errorF(it) = norm(DF*u-uprime,inf);
end
loglog(Nvec,error4,'r+','markersize',4), hold on
loglog(Nvec,errorF,'b.','markersize',15), hold on
legend('4th order finite differences','Fourier spectral method', ...
'Location','SouthWest');
grid on, xlabel 'N=2\pi/h', ylabel 'L^\infty error'
```

Spectral convergence



Calculating the derivative using Fourier transform

In practice, the matrix-vector product Du is calculated using a **fast Fourier transform**. This allows reducing the computation complexity from N^2 to $N \log N$ operations.

First, let us consider the continuous Fourier transform of a function $u(x)$, $x \in \mathbb{R}$:

$$\hat{u}(k) = \int_{-\infty}^{\infty} e^{-ikx} u(x) dx, \quad k \in \mathbb{R}.$$

It is possible to reconstruct u from $\hat{u}$ using the inverse Fourier transform,

$$u(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} \hat{u}(k) dk, \quad x \in \mathbb{R}.$$

where x is the independent variable in the physical domain, k in the wave number domain. The following identity relates the Fourier transform of a function to its derivative:

$$\widehat{u'}(k) = ik\hat{u}(k).$$

Discrete Fourier transform

If u is sampled on a grid $x \in \{0, h, ..., (N-1)h\}$, the discrete counterpart of the above transform is called the discrete Fourier transform.

$$v_k = \sum_{j=1}^{N} u_j W_N^{(j-1)(k-1)}$$

and

$$u_j = \frac{1}{N} \sum_{k=1}^{N} v_k W_N^{-(j-1)(k-1)},$$

where $j = \overline{1, N}$, $k = \overline{1, N}$ and

$$W_N = e^{(-2\pi i)/N}.$$

Fast Fourier transform (FFT) is a computationally efficient algorithm to compute the discrete Fourier transform.

Discrete Fourier transform

The derivative u' is calculated using the following algorithm:

- Fast Fourier transform of u;
- Multiply the result by the wave number;
- Inverse transform.

Example. $u(x) = e^{\sin x}$

```
N = 32;
h = 2*pi/N;
u = exp(sin(-pi+(1:N)'*h));

v = fft(u);
w = 1i*[0:N/2-1 0 -N/2+1:-1]' .* v;
u1 = real(ifft(w));
```

Fourier pseudo-spectral method PDEs

- FFT / IFFT have complexity N log(N)
- Derivatives are computed in Fourier space:

$$\begin{aligned} \widehat{\nabla\phi} &= \mathrm{i}\underline{\mathrm{k}}\widehat{\phi} \\ \widehat{\nabla^2\phi} &= -\left|\underline{\mathrm{k}}\right|^2\widehat{\phi} \end{aligned}$$

- No numerical dispersion / diffusion
- Non-linear terms evaluated in physical space, since

$$\widehat{\phi\psi} = \widehat{\phi} \star \widehat{\psi}$$

- Natural boundary conditions are periodic