Intro DG (supporting slides)

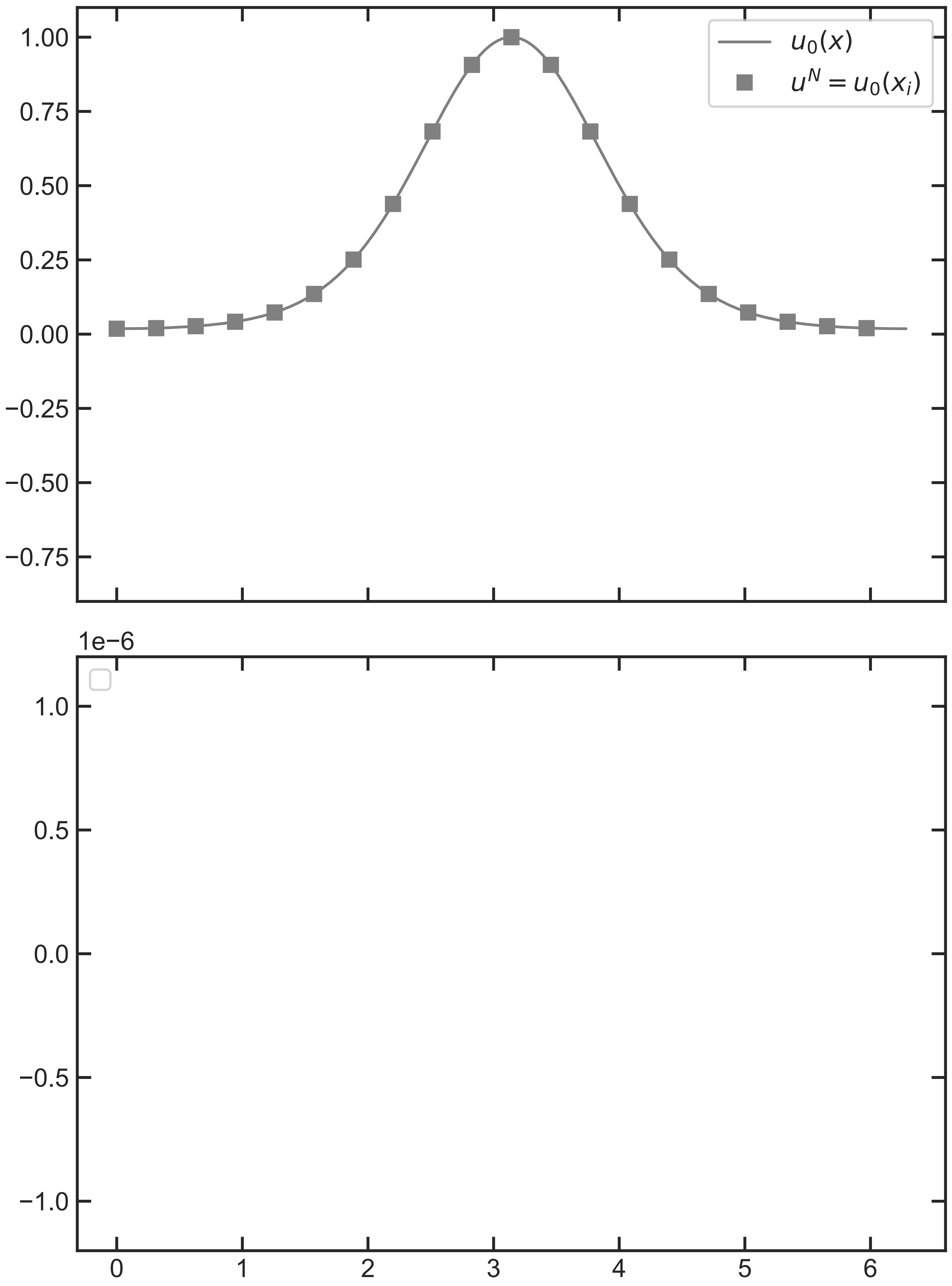
Harald Pfeiffer

ICERM Spectre workshop
Aug 6, 2024

Spectral derivatives

```python
def u0(x):
    u=np.exp(-2*(np.cos(x)+1))
    return u

def du0(x):
    du=2*np.sin(x)*np.exp(-2*(np.cos(x)+1))
    return du
```

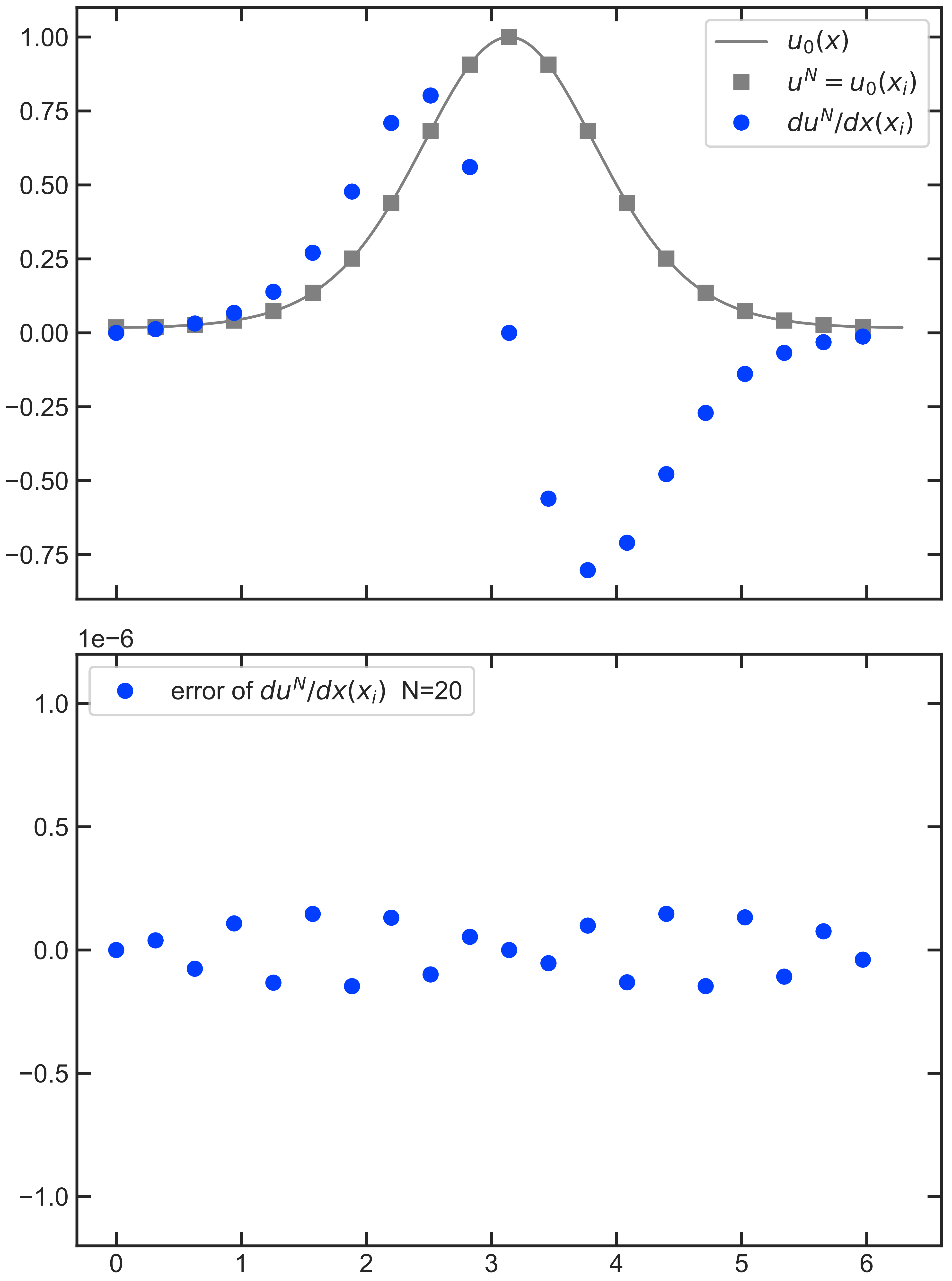
```python
# INIT:  'spectral' grid
N=20
x=np.linspace(0, 2*np.pi, num=N, endpoint=False)

# INIT:  fine grid for plotting & FD derivatives
xfine=np.linspace(0, 2*np.pi, 5000)

ax0.plot(xfine, u0(xfine), lw=1.2, c='grey', label="$u_0(x)$")
ax0.plot(x,     u0(x),     's',    c='grey', label="$u^N=u_0(x_i)$")
```



Spectral derivatives

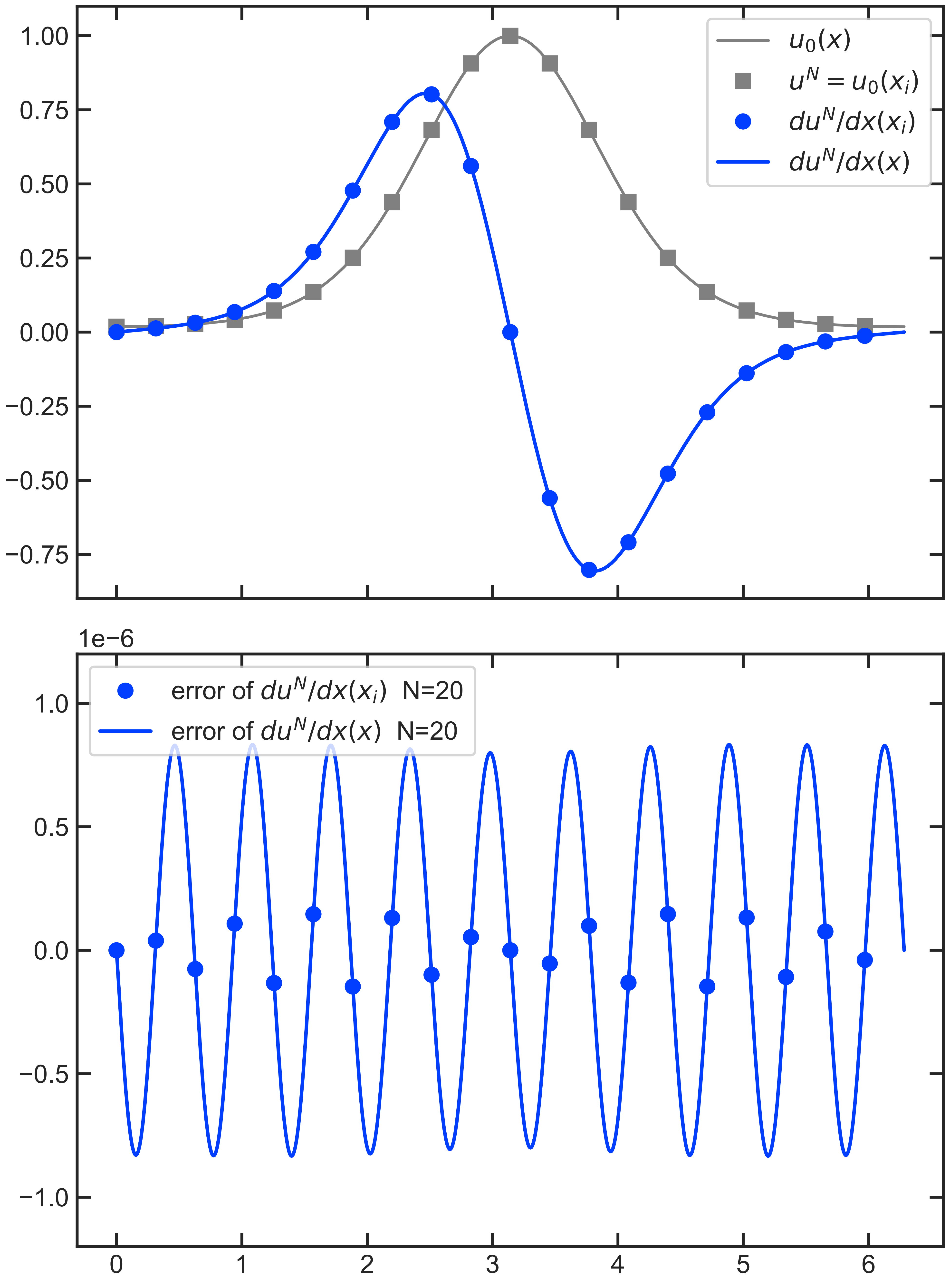
```python
def RealToFourier(u):
    """Given data u, at uniformly distributed grid-points
    0<=x<2pi, calculates Fourier coefficients A[k] such that

    u = 0.5*A[0] + Sum_k(  A[k].real * cos(k x)
                         - A[k].imag * sin(k x)
                        )

    returns complex array A
    """
    A=np.fft.rfft(u)
    A *= (2./len(u))
    return A
```

```python
# DIFFERENTIATION
ucoef = RealToFourier(u0(x))
k=np.arange(N/2+1,dtype=float)  # note N/2+1 -- complex coef array is shorter
ducoef = 1j*k*ucoef

# sum up series for derivative
duNdx = 0.5*np.ones(N)*np.real(ducoef[0])
for j in range(1,int(N/2)+1):
    duNdx += np.real(ducoef[j] * np.exp(1j*j*x))
ax0.plot(x, duNdx,         'o', c='b', label='$du^N/dx(x_i)$');
ax1.plot(x, duNdx-du0(x), 'o', c='b', label=f'error of $du^N/dx(x_i)$  N={N}'
```



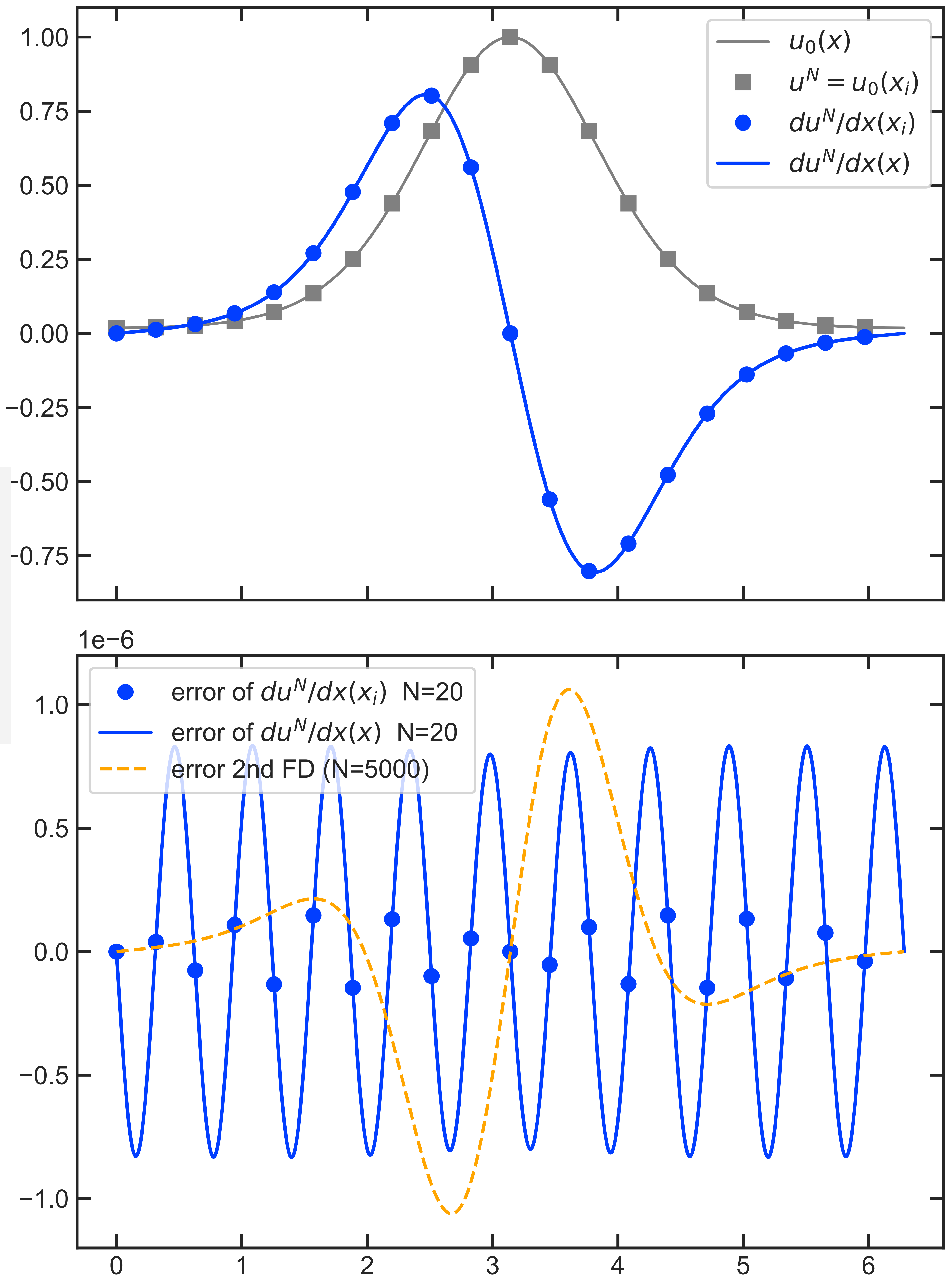
Spectral derivatives

```python
# INTERPOLATION:  sum up spectal series on fine grid
duNfine=  0*xfine + 0.5*np.real(ducoef[0])
for j in range(1,int(N/2)+1):
    duNfine += np.real(ducoef[j] * np.exp(1j*j*xfine))
ax0.plot(xfine, duNfine, lw=1.5, c='b', label='$du^N/dx(x)$');
SAVE(fig,ax0,ax1,"SpectralDeriv3.pdf")

ax1.plot(xfine, duNfine-du0(xfine), lw=1.5, c='b', label=f'error of
```



Spectral derivatives

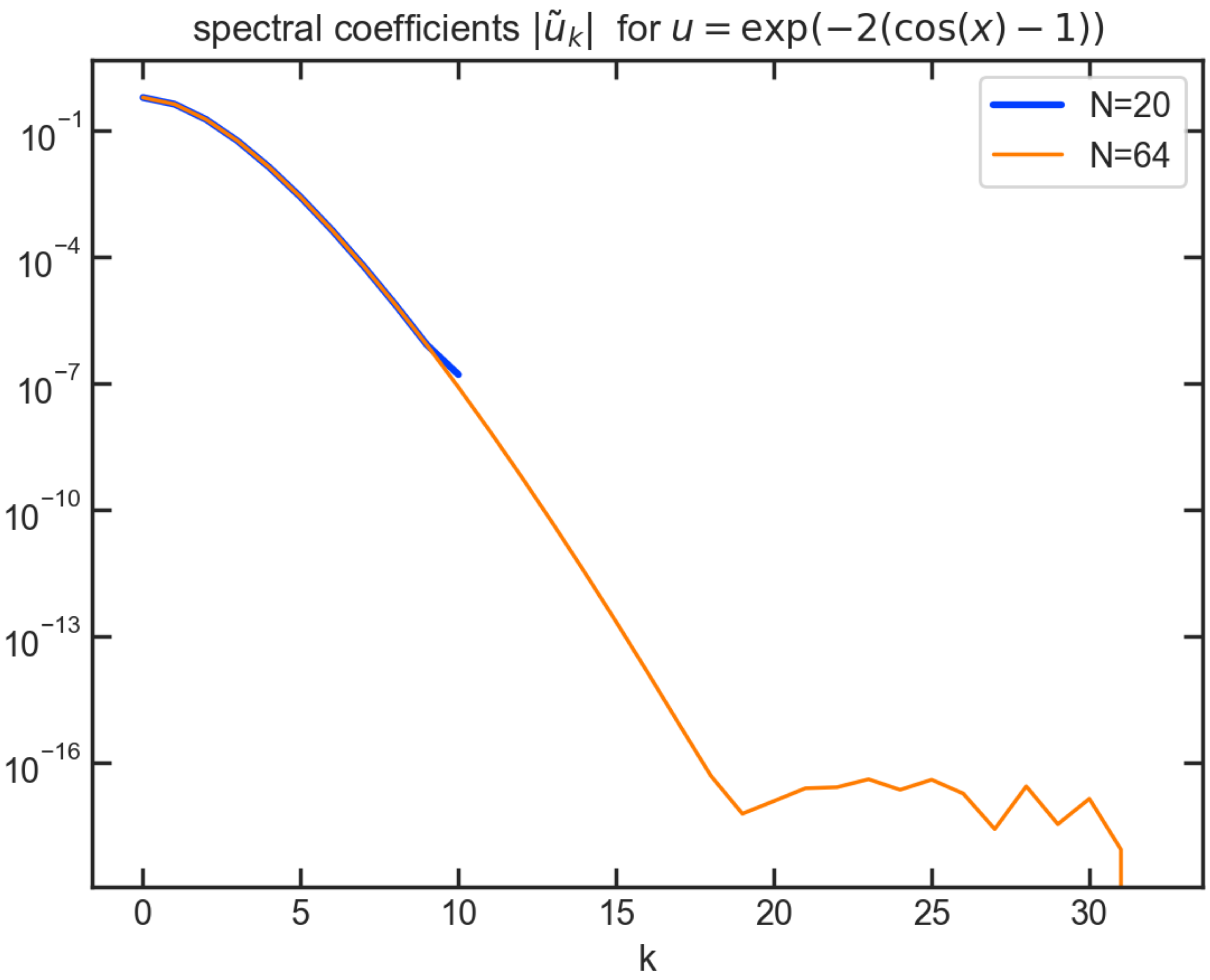
```python
# FINITE-DIFFERENCE DERIVATIVE
ufine=u0(xfine)
du_FD = (ufine[2:]-ufine[:-2])/(xfine[2]-xfine[0])
ax1.plot(xfine[1:-1], du_FD-du0(xfine[1:-1]),
         '--', lw=1.4, color='orange', label="error 2nd FD (N=5000)")
```



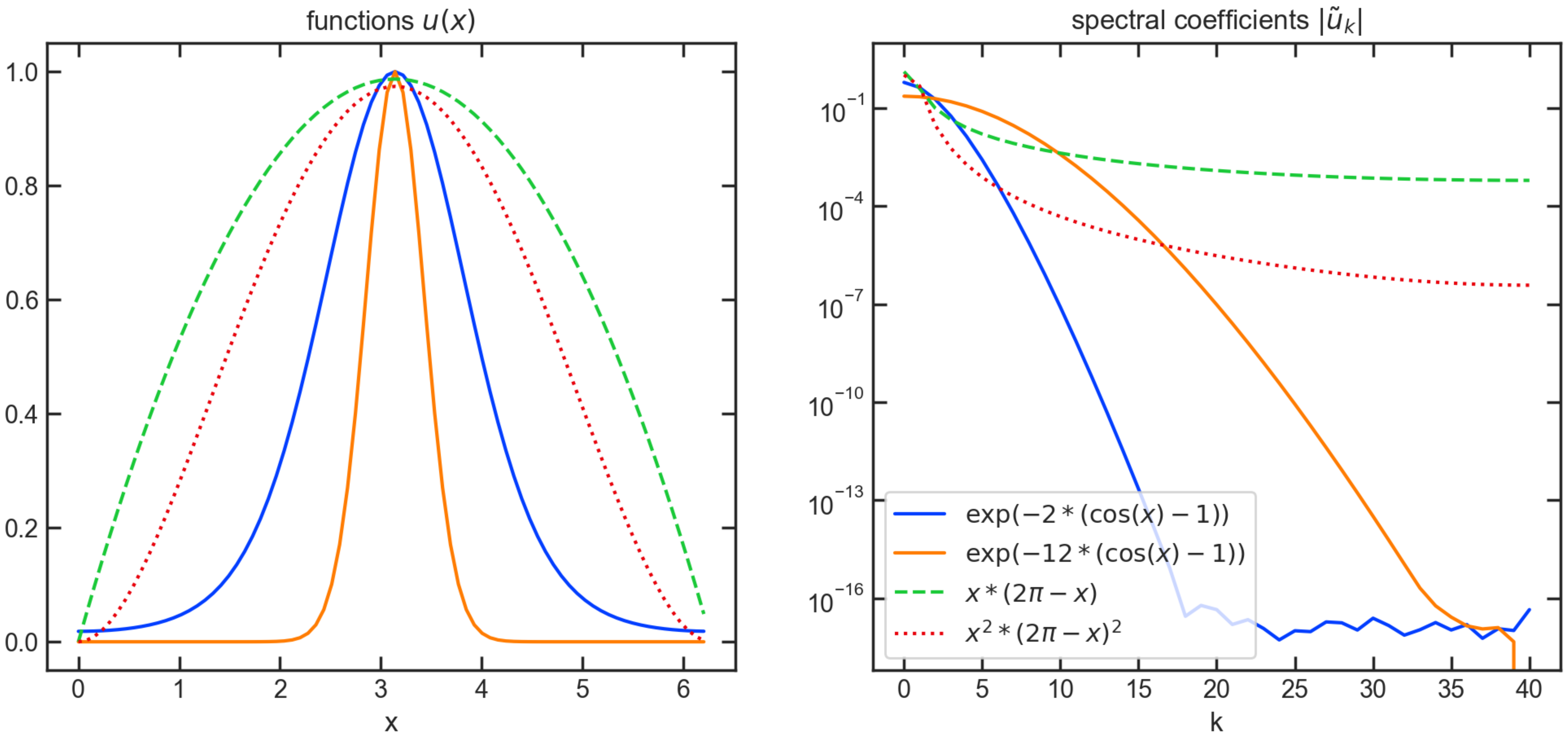
Spectral with N=20 is as accurate as 2nd order FD with N=5000 !!

exponential decay of spectral coefficients

```
for N in 20,64:
    x=np.linspace(0, 2*np.pi, num=N, endpoint=False)
    ucoef=RealToFourier(u0(x))
    plt.semilogy(np.abs(ucoef), lw=10/N**0.5, label=f"N={N}")
plt.xlabel('k')
plt.title('spectral coefficients $|\\tilde u_k|$  for $u=\exp(
plt.legend();
```



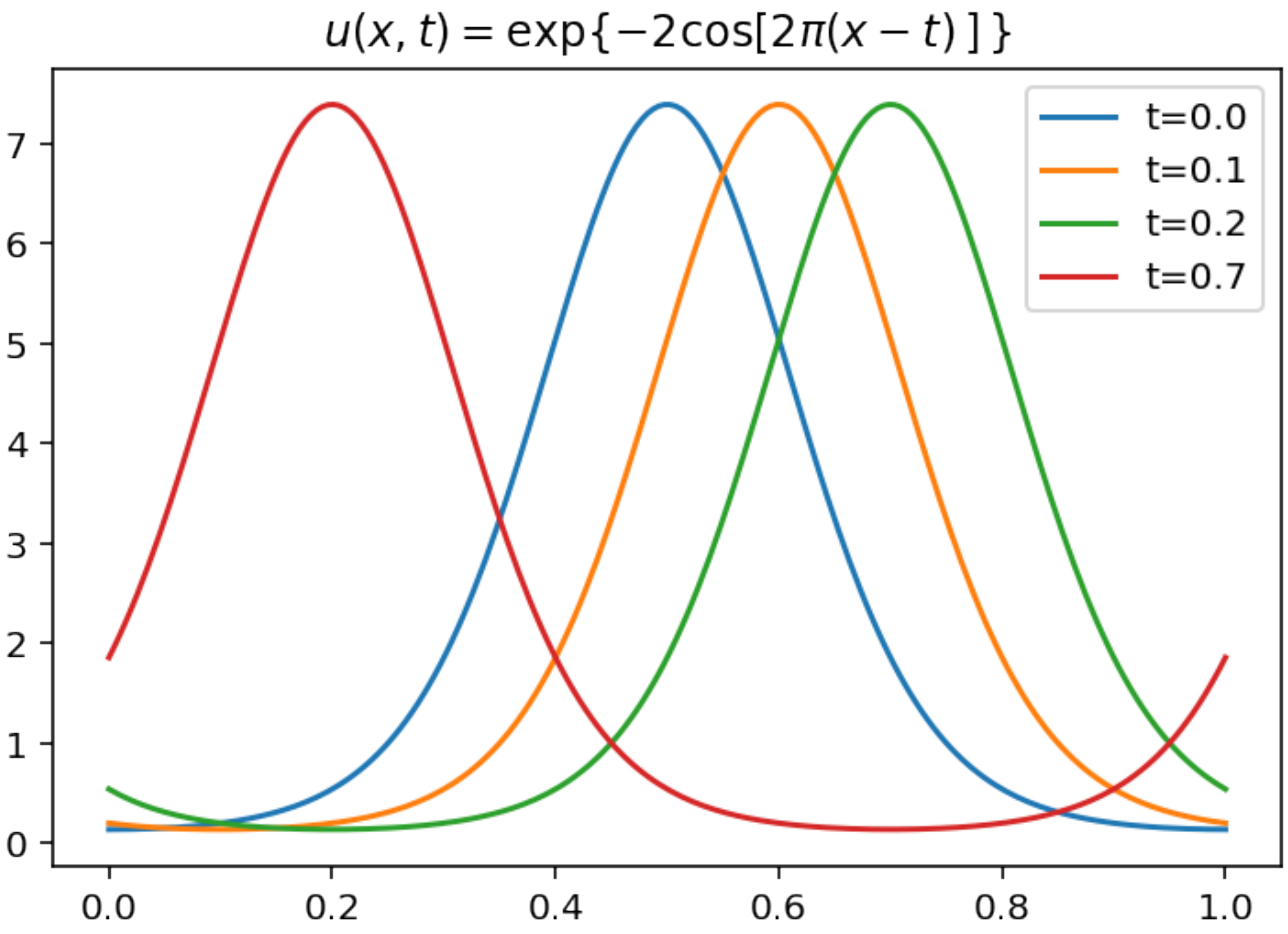
Convergence rate vs. *shape* of function



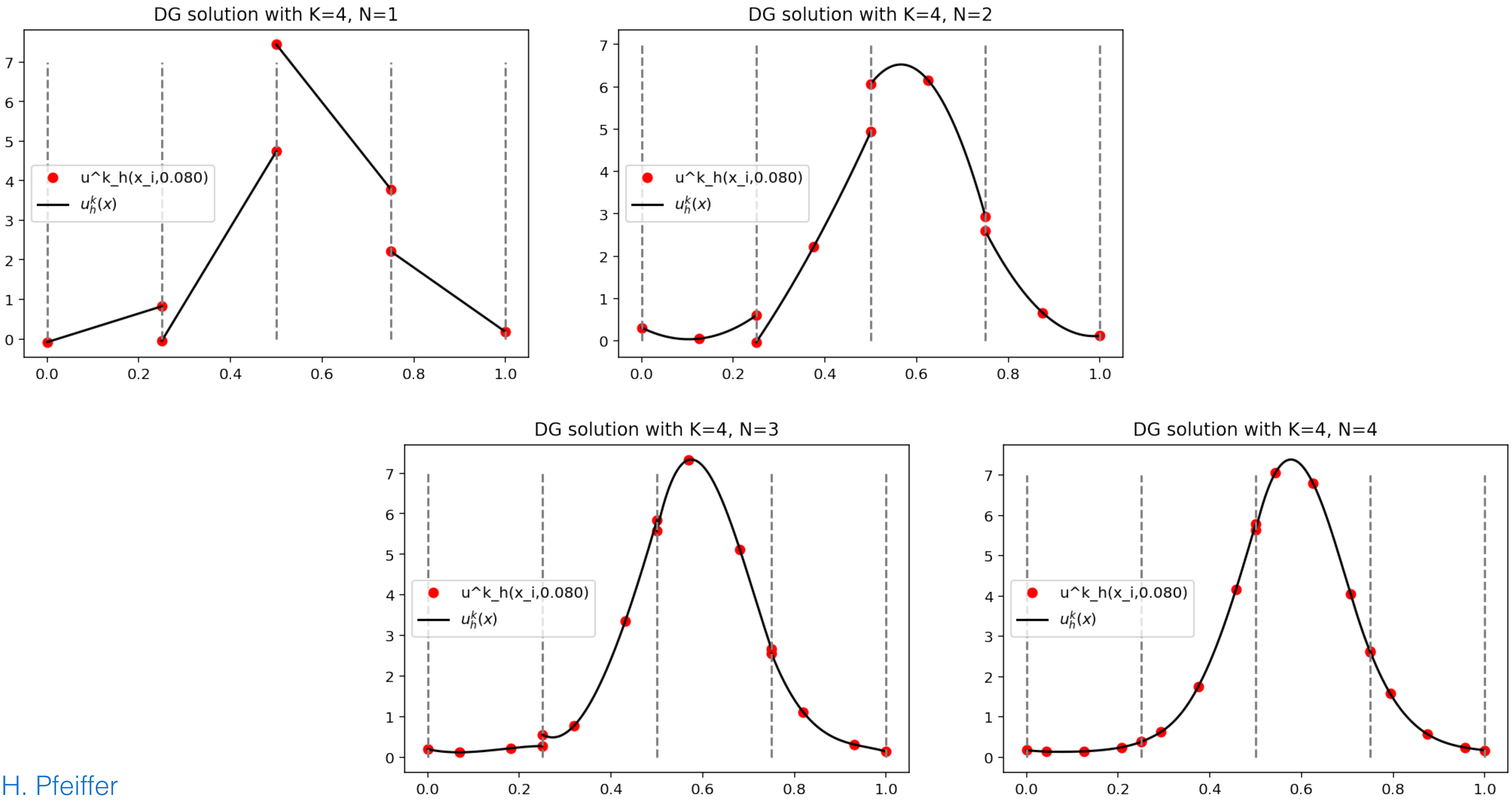
Solving the Advection Equation with DG

Example solution, advection eqn

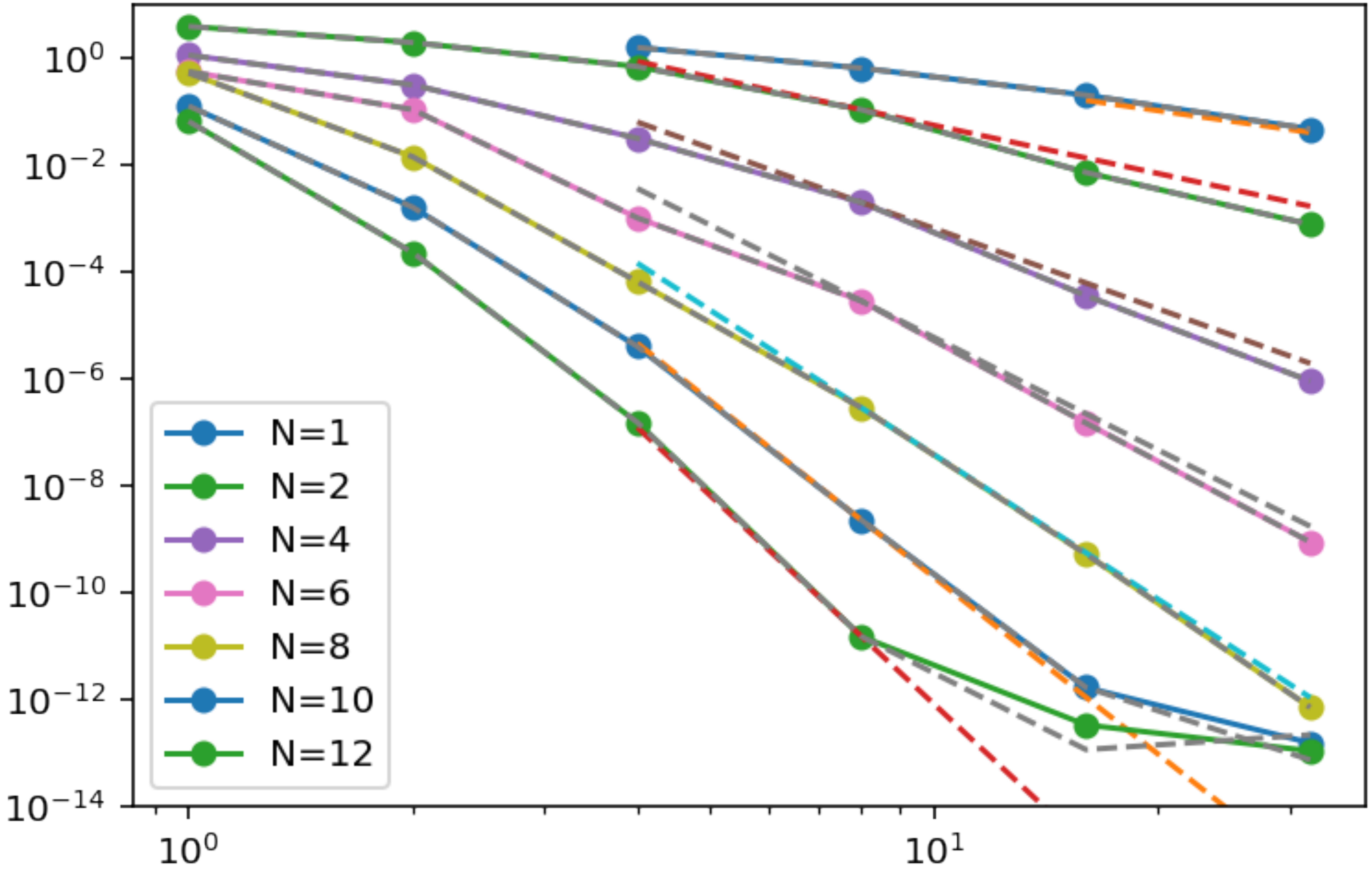
$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$$



DG solution at different orders N



DG: Convergence



# of elements K

solid-colored:
error

dashed-colored:
power-law $\propto K^{-N-1}$

grey-dashed:
2x Δt
(check time-stepping)

CF=2^{-7}=0.008