Numerical Methods in Geophysics:
High-Order Operators

Why do we need higher order operators?

Taylor Operators

One-sided operators

High-order Taylor Extrapolation

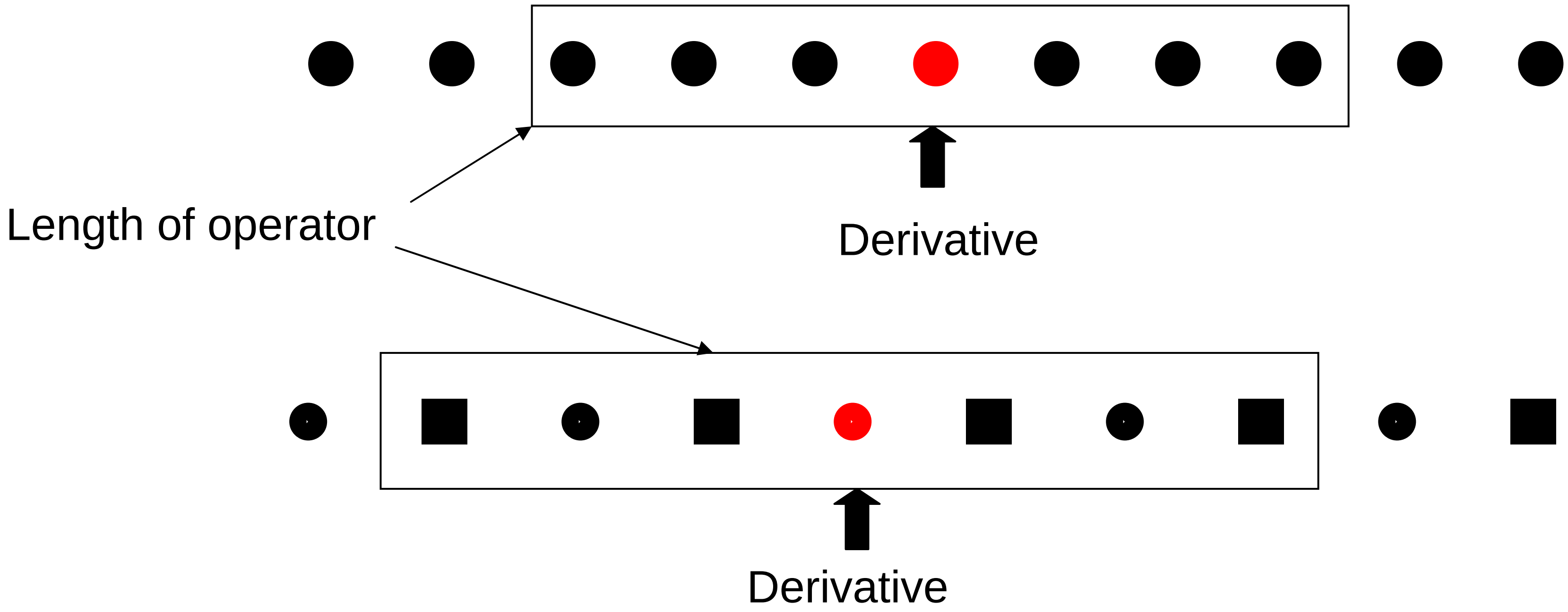
Runge-Kutta Method

Truncated Fourier Operators - Derivative and Interpolation

Accuracy of high-order schemes derivatives

Why do we need higher-order operators?

For realistic problems the first and second order methods are not accurate enough. So far we only used information from the nearest neighbouring grid points.
Could we improve the accuracy of the derivative operators by using more information (on both sides)?



Taylor Operators

... like so often we look at Taylor series ...
Remember how we derived the second-order scheme

$$af^+ \approx af + af'dx$$

$$bf^- \approx bf - bf'dx$$

$$\Rightarrow af^+ + bf^- \approx (a+b)f + (a-b)f'dx$$

the solution to this equation for a and b leads to
a system of equation which can be cast in matrix form

Interpolation

$$a + b = 1$$

$$a - b = 0$$

Derivative

$$a + b = 0$$

$$a - b = 1/dx$$

... in matrix form ...

Interpolation

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Derivative

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 0 \\ 1/dx \end{pmatrix}$$

... so that the solution for the *weights* is ...

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{-1}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{-1}\begin{pmatrix} 0 \\ 1/dx \end{pmatrix}$$

... and the result ...

Interpolation

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$$

Derivative

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{2\,dx} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Can we generalise this idea to longer operators?

Let us start by extending the Taylor expansion beyond f(x±dx):

$$*a \,|\qquad f(x-2dx) \approx f - (2dx) f' + \frac{(2dx)^2}{2!} f'' - \frac{(2dx)^3}{3!} f'''$$

$$*b \,|\qquad f(x-dx) \approx f - (dx) f' + \frac{(dx)^2}{2!} f'' - \frac{(dx)^3}{3!} f'''$$

$$*c \,|\qquad f(x+dx) \approx f + (dx) f' + \frac{(dx)^2}{2!} f'' + \frac{(dx)^3}{3!} f'''$$

$$*d \,|\qquad f(x+2dx) \approx f + (2dx) f' + \frac{(2dx)^2}{2!} f'' + \frac{(2dx)^3}{3!} f'''$$

... again we are looking for the coefficients a,b,c,d with which the function values at x±(2)dx have to be multiplied in order to obtain the interpolated value or the first (or second) derivative!

... Let us add up all these equations like in the previous case ...

Taylor Operators

$$af^{--} + bf^{-} + cf^{+} + df^{++} \approx$$

$$f\,(a + b + c + d) +$$

$$dxf\,'(-2a - b + c + 2d) +$$

$$dx^2 f\,''(2a + \frac{b}{2} + \frac{c}{2} + 2d) +$$

$$dx^3 f\,'''(-\frac{8}{6}a - \frac{1}{6}b + \frac{1}{6}c + \frac{8}{6}d)$$

... we can now ask for the coefficients a,b,c,d, so that the left-hand-side yields either f,f',f'',f''' ...

... if you want the interpolated value ...

$$a + b + c + d = 1$$

$$-2a - b + c + 2d = 0$$

$$2a + \frac{b}{2} + \frac{c}{2} + 2d = 0$$

$$-\frac{8}{6}a - \frac{1}{6}b + \frac{1}{6}c + \frac{8}{6}d = 0$$

... you need to solve the matrix system ...

Taylor Operators

... Interpolation ...

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -2 & -1 & 1 & 2 \\ 2 & 1/2 & 1/2 & 2 \\ -8/6 & -1/6 & 1/6 & 8/6 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

... with the result after inverting the matrix on the lhs ...

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} -1/6 \\ 2/3 \\ 2/3 \\ -1/6 \end{pmatrix}$$

... first derivative ...

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -2 & -1 & 1 & 2 \\ 2 & 1/2 & 1/2 & 2 \\ -8/6 & -1/6 & 1/6 & 8/6 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 1/dx \\ 0 \\ 0 \end{pmatrix}$$

... with the result ...

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \frac{1}{2dx} \begin{pmatrix} 1/6 \\ -4/3 \\ 4/3 \\ -1/6 \end{pmatrix}$$

... third derivative ...

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -2 & -1 & 1 & 2 \\ 2 & 1/2 & 1/2 & 2 \\ -8/6 & -1/6 & 1/6 & 8/6 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1/dx^3 \end{pmatrix}$$

... with the result ...

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \frac{1}{dx^3} \begin{pmatrix} -1/2 \\ 1 \\ -1 \\ 1/2 \end{pmatrix}$$

... why did it not work for the 2nd derivative ?

Taylor Operators

$$*a \mid \quad f(x-2dx) \approx f - (2dx) f' + \frac{(2dx)^2}{2!} f'' - \frac{(2dx)^3}{3!} f''' + \frac{(2dx)^4}{4!} f''''$$

$$*b \mid \quad f(x-dx) \approx f - (dx) f' + \frac{(dx)^2}{2!} f'' - \frac{(dx)^3}{3!} f''' + \frac{(dx)^4}{4!} f''''$$

$$*c \mid \quad \boxed{f(x) = f}$$

$$*d \mid \quad f(x+dx) \approx f + (dx) f' + \frac{(dx)^2}{2!} f'' + \frac{(dx)^3}{3!} f''' + \frac{(dx)^4}{4!} f''''$$

$$*e \mid \quad f(x+2dx) \approx f + (2dx) f' + \frac{(2dx)^2}{2!} f'' + \frac{(2dx)^3}{3!} f''' + \frac{(2dx)^4}{4!} f''''$$

… note that we had to add the 4th derivatives which will give us the required constraints on the coefficients a,b,c,d,e

Taylor Operators

$$
\begin{aligned}
af^{--} + bf^{-} + cf + df^{+} + ef^{++} \approx \\
f(a+b+c+d+e) + \\
dxf'(-2a - b + 0 + d + 2e) + \\
dx^2 f''(2a + \frac{b}{2} + 0 + \frac{d}{2} + 2e) + \\
dx^3 f'''(-\frac{8}{6}a - \frac{1}{6}b + 0 + \frac{1}{6}d + \frac{8}{6}e) + \\
dx^4 f''''(\frac{2}{3}a + \frac{1}{24}b + 0 + \frac{1}{24}d + \frac{2}{3}e)
\end{aligned}
$$

... so finally we end up with the system ...

Taylor Operators

... if you want the second derivative ...

$$a + b + c + d + e = 0$$

$$-2a - b + 0 + d + 2e = 0$$

$$2a + \frac{b}{2} + 0 + \frac{d}{2} + 2e = 1$$

$$-\frac{8}{6}a - \frac{1}{6}b + 0 + \frac{1}{6}d + \frac{8}{6}e = 0$$

$$\frac{2}{3}a + \frac{1}{24}b + 0 + \frac{1}{24}d + \frac{2}{3}e = 0$$

... you need to solve the matrix system ...

... could we find interpolation weights like this ?

... second derivative ...

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \\ 2 & 1/2 & 0 & 1/2 & 2 \\ -8/6 & -1/6 & 0 & 1/6 & 8/6 \\ 2/3 & 1/24 & 0 & 1/24 & 2/3 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1/dx^2 \\ 0 \\ 0 \end{pmatrix}$$

... with the result ...

$$\begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \frac{1}{dx^2} \begin{pmatrix} -1/12 \\ 4/3 \\ -5/2 \\ 4/3 \\ -1/12 \end{pmatrix}$$

Taylor Operators

… Fornberg[1] (1996) gives a closed-form expression for
the first derivative weights …

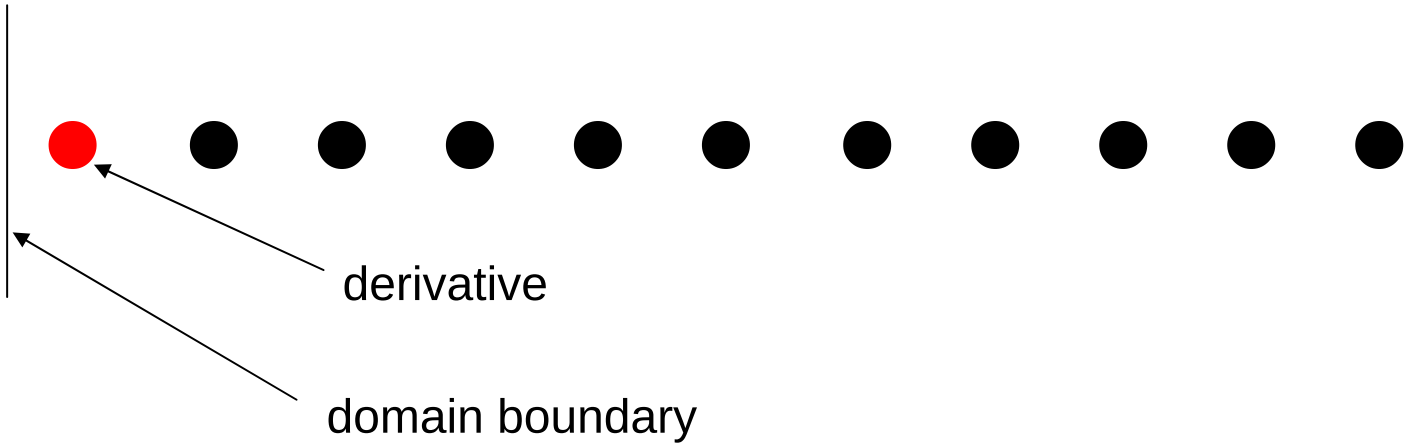
$$w^1_{p,j} = \begin{cases} \dfrac{(-1)^{j+1}(p/2)!^2}{j(p/2+j)!(p/2-j)!} & \text{if j= ✐1, ✐2,...,✐p/2} \\ 0 & \text{if j= 0} \end{cases}$$

... where p(even) is the order of accuracy, j is the x-position of the weight.

[1]Fornberg, B., A practical guide to pseudospectral methods, Cambridge University Press.

One-sided operators

Before we look at how the operators look like as they grow longer
we investigate whether we can approximate a derivative near
a physical boundary ....



Let us follow the same route as before and use Taylor series.
Let's start with a first order scheme.

One-sided operators

... so we have to look for information on one side only

$$af^{+} \approx af + af'dx$$

$$bf^{++} \approx bf + bf'(2dx)$$

$$\Rightarrow af^{+} + bf^{++} \approx (a+b)f + (a+2b)f'dx$$

the solution to this equation for a and b leads to
a system of equations which can be cast in matrix form

Derivative

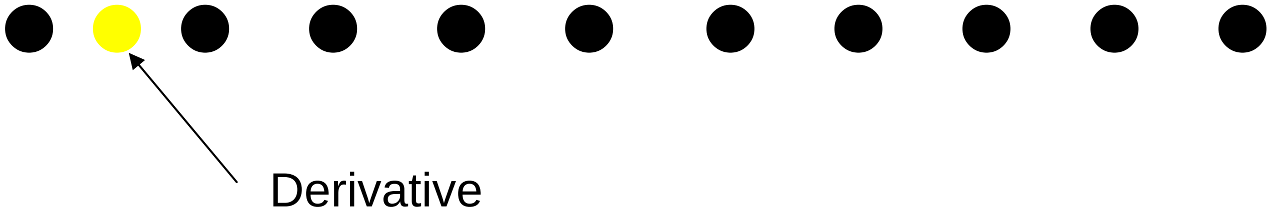
$$\begin{aligned} a + b &= 0 \\ a + 2b &= 1/dx \end{aligned} \quad \Rightarrow \quad \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

... and the solution is ...

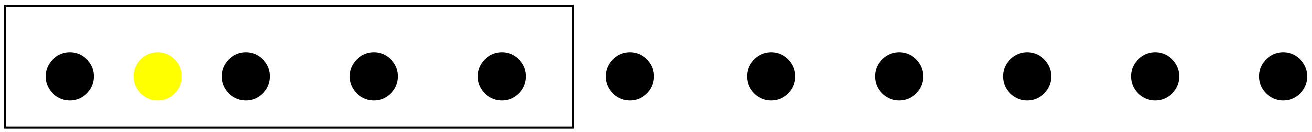
One-sided operators

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{dx} \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

This is our well known definition of the centered derivative, but it will be defined not right at the boundary but dx/2 away from it!



Let us extend this to the right and find higher-order operators



One-sided operators

$$*a\ |\qquad f = f$$

$$*b\ |\qquad f^{+} \approx f + (dx) f' + \frac{(dx)^2}{2!} f'' + \frac{(dx)^3}{3!} f'''$$

$$*c\ |\qquad f^{++} \approx f + (2dx) f' + \frac{(2dx)^2}{2!} f'' + \frac{(2dx)^3}{3!} f'''$$

$$*d\ |\qquad f^{+++} \approx f + (3dx) f' + \frac{(3dx)^2}{2!} f'' + \frac{(3dx)^3}{3!} f'''$$

... again we multiply by our coefficients and add everything up ...

One-sided operators

$$
\begin{aligned}
af + bf^{+} + cf^{++} + df^{+++} \approx \\
f\,(a + b + c + d) + \\
dxf\,'(0 + b + 2c + 3d) + \\
dx^{2} f\,''(0 + \frac{1}{2}b + 2c + \frac{9}{2}d) + \\
dx^{3} f\,'''(0 + \frac{1}{6}b + \frac{4}{3}c + \frac{27}{6}d)
\end{aligned}
$$

... to obtain the derivatives we have to solve the system ...

One-sided operators

... if you want the first derivative ...

$$a + b + c + d = 0$$

$$0 + b + 2c + 3d = 1$$

$$0 + \frac{1}{2} b + 2c + \frac{9}{2} d = 0$$

$$0 + \frac{1}{6} b + \frac{4}{3} c + \frac{27}{6} d = 0$$

... you need to solve the matrix system ...

One-sided operators

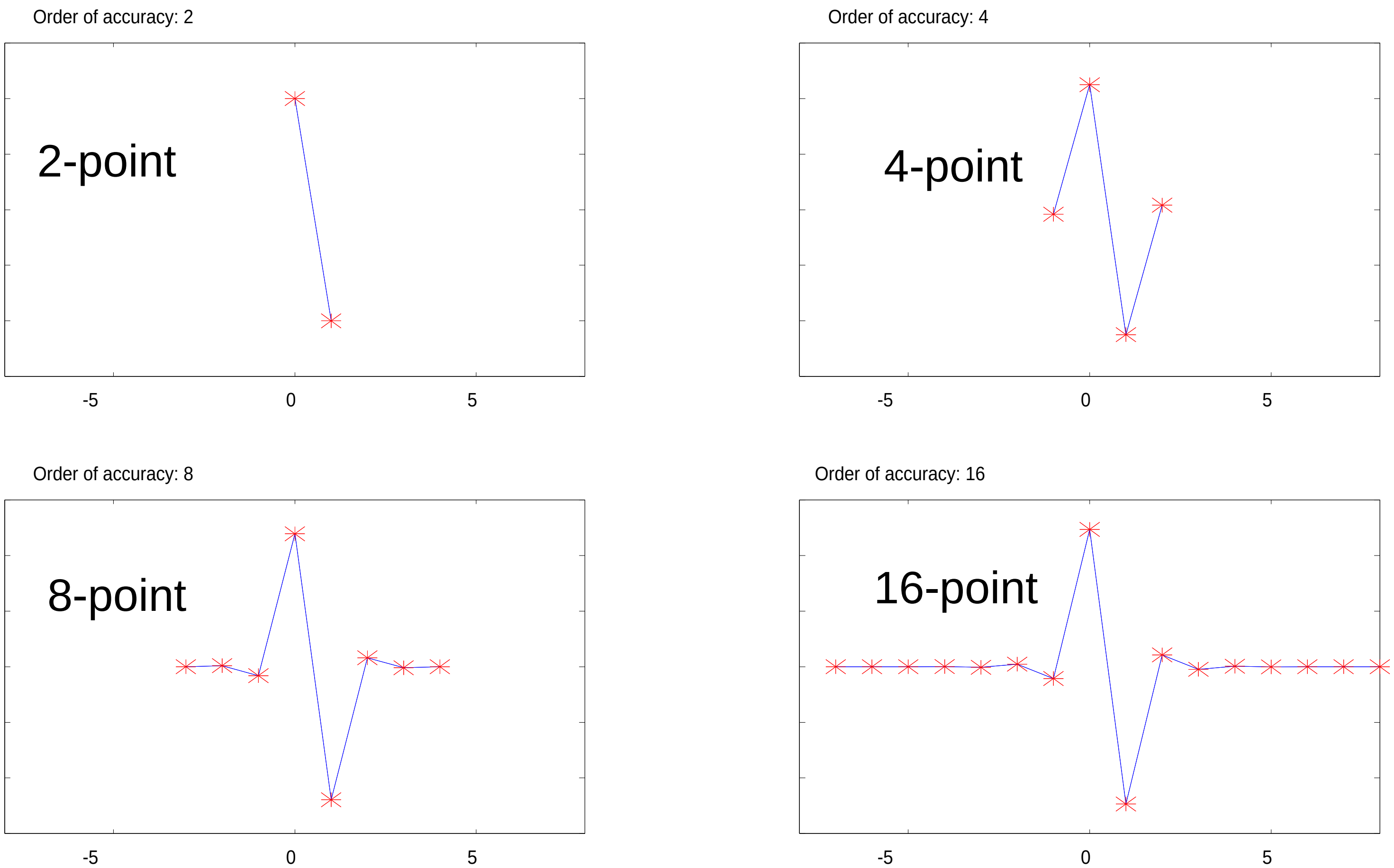
... Interpolation ...

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 1/2 & 2 & 9/2 \\ 0 & 1/6 & 4/3 & 27/6 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 1/dx \\ 0 \\ 0 \end{pmatrix}$$

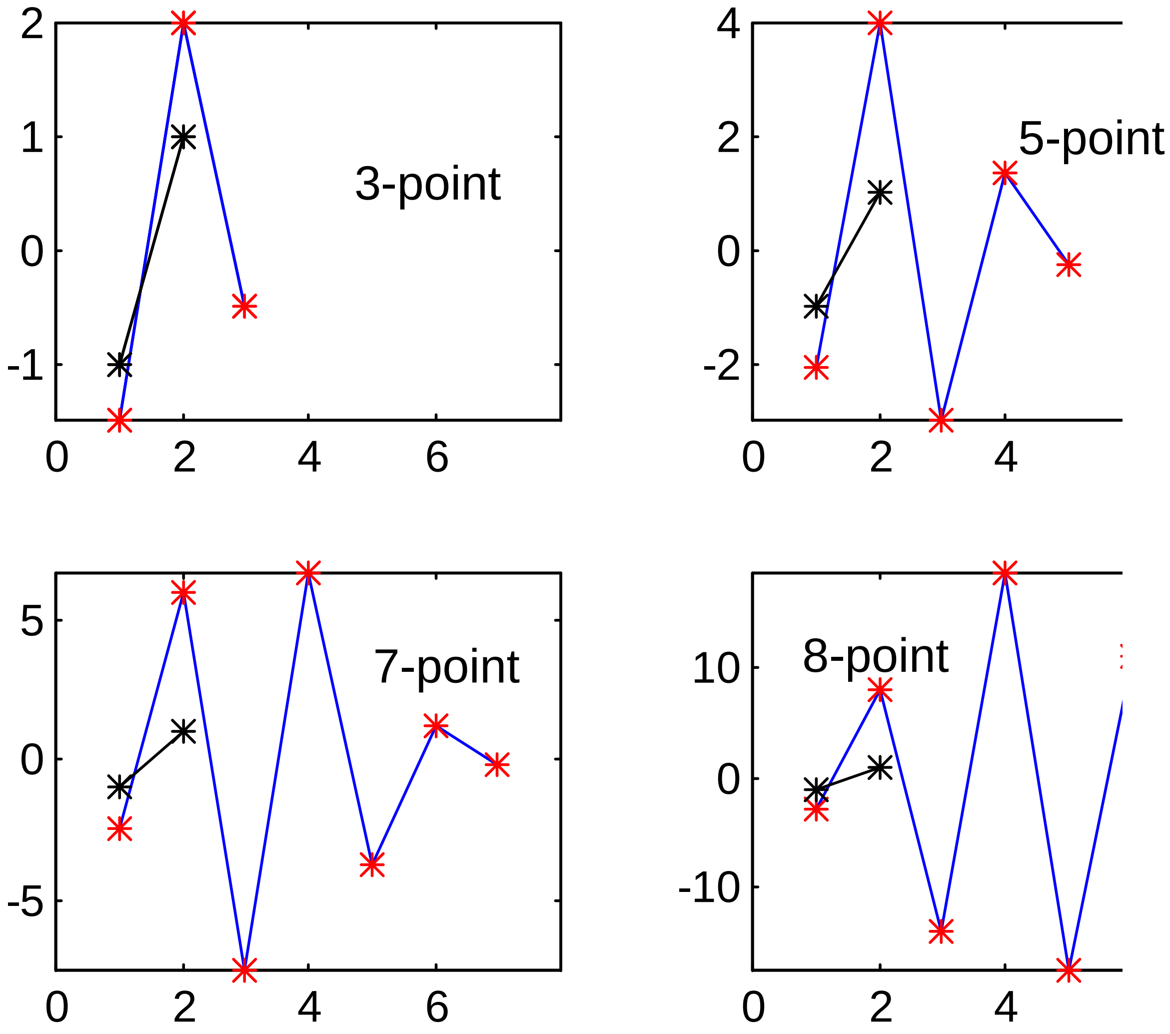
... with the result after inverting the matrix on the lhs ...

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \frac{1}{dx} \begin{pmatrix} -11/6 \\ 3 \\ -3/2 \\ 1/3 \end{pmatrix}$$

Taylor Operators



Taylor Operators - one sided



Note the exploding coefficients with increasing operator length

Taylor Operators - Summary

Finite-difference operators with high-order accuracy can be derived using Taylor series. For two-sided operators the coefficients rapidly decrease. For one-sided operators the coefficients get larger with increasing operator length.

Now that we improved the accuracy of the space derivatives, how can we improve the accuracy of the time extrapolation?

Let’s look at the Taylor scheme ...

High-order Taylor extrapolation

Let us look at the acoustic wave equation

$$\ddot{p} = c^2 (\partial_x^2 + \partial_z^2)\, p + c^2 S$$

.. we now know how to accurately calculate the r.h.s. of this equation...
our standard FD scheme for the time extrapolation yields

$$p(t+dt) = 2p(t) - p(t-dt) + dt^2\, \ddot{p}$$

... extending this to higher orders leads to the scheme ...

$$p(t+dt) = 2p(t) - p(t-dt) + 2\sum_{n=1}^{N} \frac{dt^{2n}}{(2n)!} p^{2n}$$

... this has interesting consequences as we only need the even orders of the time derivative, which we can easily calculate ...

High-order Taylor extrapolation

... since ...

$$\partial_t^2 p = c^2(\partial_x^2 + \partial_z^2)p + c^2 S$$

... we have also ...

$$\partial_t^4 p = c^2(\partial_x^2 + \partial_z^2)\partial_t^2 p + c^2 \partial_t^2 S$$

... or ...

$$\partial_t^6 p = c^2(\partial_x^2 + \partial_z^2)\partial_t^4 p + c^2 \partial_t^4 S$$

... so we can loop through our algorithm as long as we want (N times) to achieve higher-order accuracy in the time-extrapolation scheme ...

$$p(t+dt) = 2p(t) - p(t-dt) + \boxed{2\sum_{n=1}^{N} \frac{dt^{2n}}{(2n)!} p^{2n}}$$

.. however we have to be careful how the spatial and temporal operators behave and whether the accuracy of the solution to the pde actually improves!

High-order extrapolation

... often we have to extrapolate a first order system ...

$$\partial_t T = f(T, t)$$

... or ...

$$\partial_t \dot{u} = \frac{1}{\rho(x)} \partial_x \tau$$

... and we initially used a simple scheme like ...

$$T_{j+1} \approx T_j + \mathrm{dt} f(T_j, t_j)$$

... this scheme is also known as the *Euler* scheme and is of little practical use ...

... how about predicting a value and
then averaging ....

$$T^*{}_{j+1} \approx T_j + \mathrm{dt} f(T_j, t_j)$$

this is our first guess (equivalent to the Euler scheme) and now we use this value to improve our solution ...

$$T_{j+1} = T_j + \frac{1}{2} dt \left[f(T_j, t_j) + f(T^*_{j+1}, t_{j+1}) \right]$$

... leading to a general algorithm like ...

For n=0,1,2,3,...N-1

$$x_{n+1} = x_n + dx$$

$$k_1 = dxf\,(x_n, y_n)$$

$$k_2 = dxf\,(x_{n+1}, y_n + k_1)$$

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$$

End

predictor

corrector

called predictor-corrector or modified Euler or .... scheme ...
... how does this apply to our cooling problem ?

High-order extrapolation

$$\frac{dT}{dt} = -T/\tau$$

$$T_{n+1} = T_n - \frac{dt}{\tau} T_n$$

... this was the simplest scheme ...
with the modified Euler scheme we get

For n=0,1,2,3,...N-1

$$t_{n+1} = t_n + dt$$

$$k_1 = -\frac{dt}{\tau} T_n$$

$$k_2 = -\frac{dt}{\tau}(T_n + k_1)$$

$$T_{n+1} = T_n + \frac{1}{2}(k_1 + k_2)$$

End

For n=0,1,2,3,...N-1

$$x_{n+1} = x_n + dx$$

$$k_1 = dxf\ (x_n, y_n)$$

$$k_2 = dxf\ (x_{n+1}, y_n + k_1)$$

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$$

End

... the next more accurate scheme is the fourth order Runge-Kutta method, an extension of the predictor-corrector scheme ...

For n=0,1,2,3,...N-1

$$x_{n+1} = x_n + dx$$
$$k_1 = dxf\ (x_n, y_n)$$
$$k_2 = dxf\ (x_{n+1/2}, y_n + k_1/2)$$
$$k_3 = dxf\ (x_{n+1/2}, y_n + k_2/2)$$
$$k_4 = dxf\ (x_{n+1}, y_n + k_3)$$
$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

End

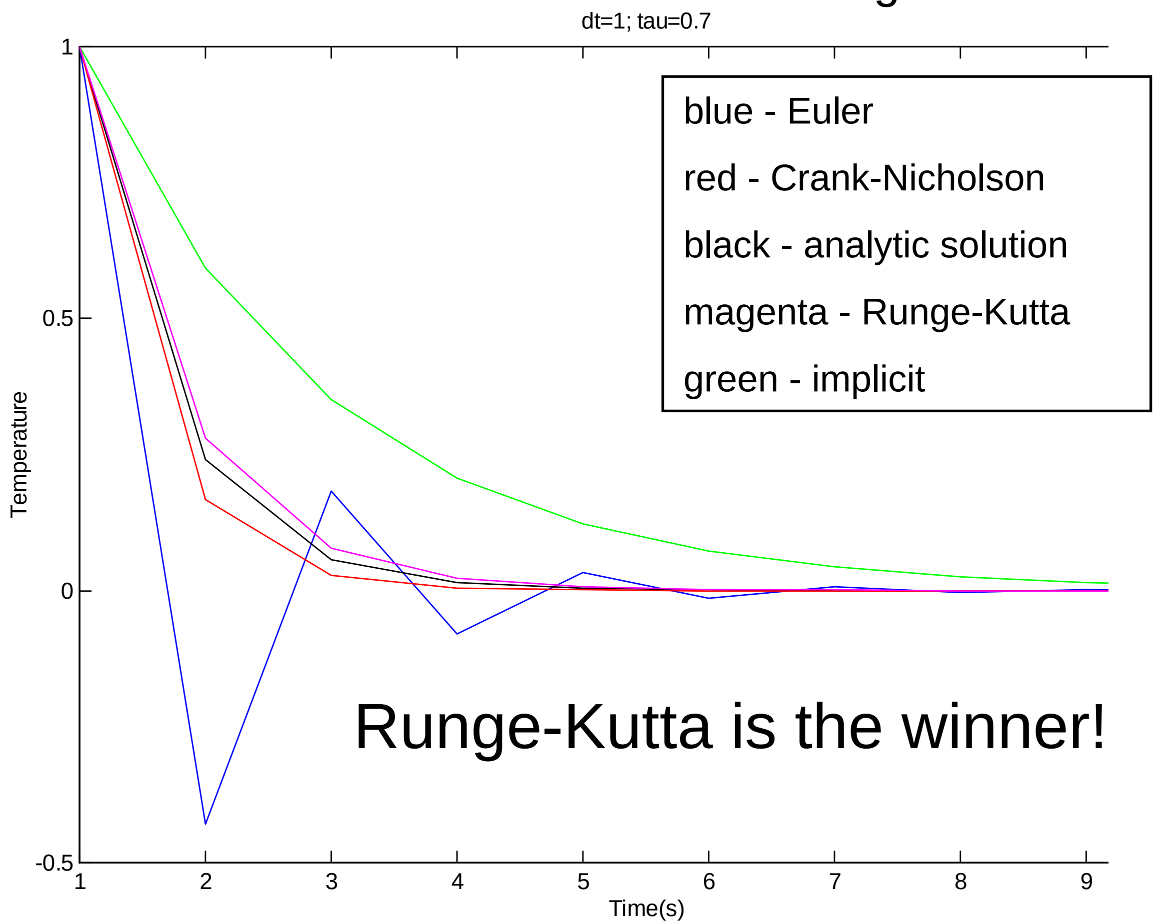
High-order extrapolation

... Matlab sample code ...

```
for i=1:nt,

t(i)=i*dt;
T(i+1)=T(i)-dt/tau*T(i);                        % Euler
Ta(i+1)=exp(-dt*i/tau);                         % Analytical solution
Ti(i+1)=T(i)*(1+dt/tau)^(-1);                   % implicit
Tm(i+1)=(1-dt/(2*tau))/(1+dt/(2*tau))*Tm(i);    % mixed implicit-explicit

k1=-dt/tau*Te(i);
k2=-dt/tau*(Te(i)+k1);
Te(i+1)=Te(i)+1/2*(k1+k2);                      % predictor-corrector

k1=-dt/tau*Tr(i);
k2=-dt/tau*(Tr(i)+k1/2);
k3=-dt/tau*(Tr(i)+k2/2);
k4=-dt/tau*(Tr(i)+k3);
Tr(i+1)=Tr(i)+1/6*(k1+2*k2+2*k3+k4);            % Runge-Kutta

end
```

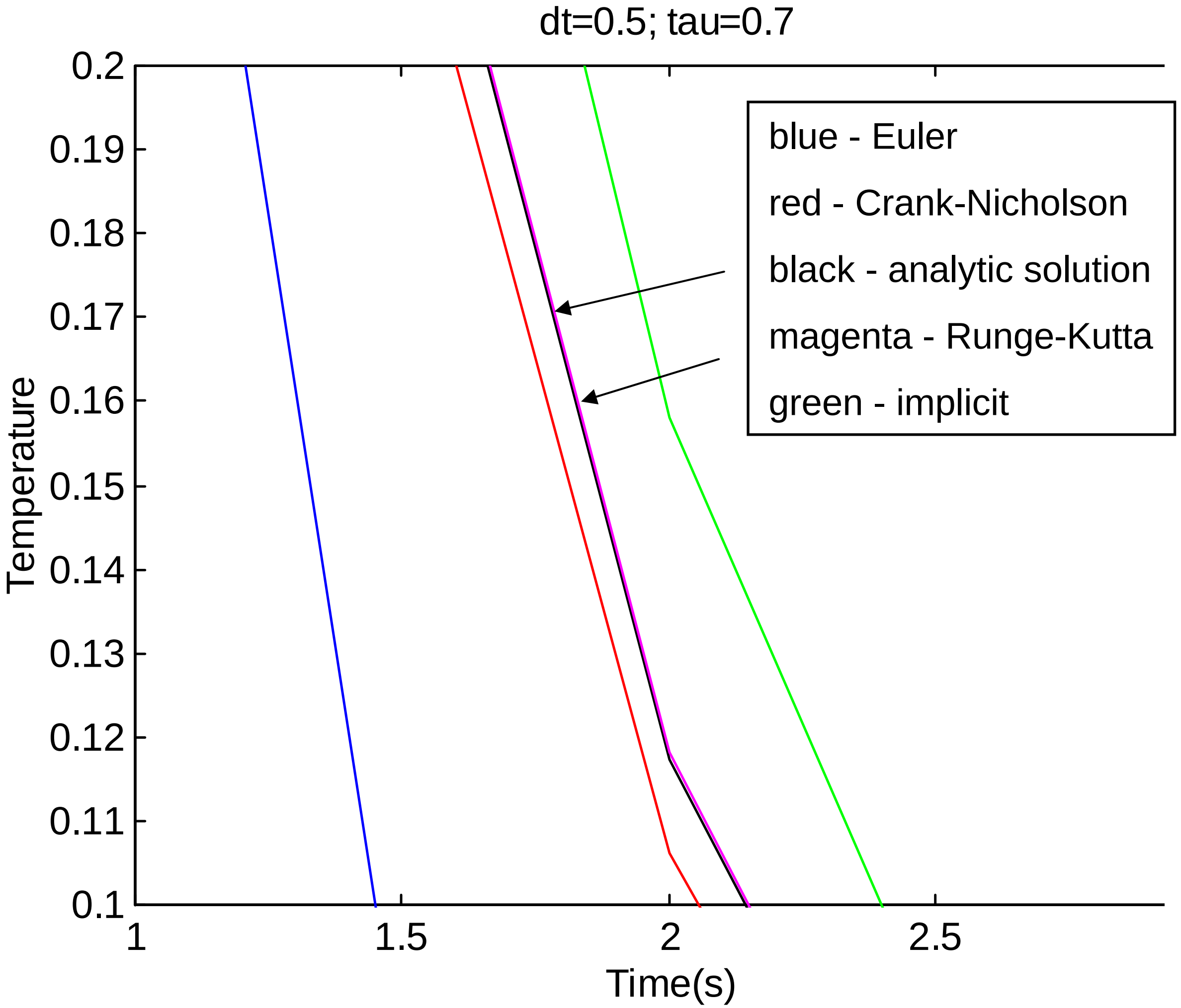
... with the results ...

Comparison of low order implicit, mixed implicit-explicit (Crank-Nicholson), modified Euler (predictor-corrector), Runge-Kutta (fourth order) for Newtonian Cooling



High-order extrapolation

Comparison of low order implicit, mixed implicit-explicit (Crank-Nicholson), modified Euler (predictor-corrector), Runge-Kutta (fourth order) for Newtonian Cooling



Fourier Coefficients

We will now approach the problem of finding high-order space operators from a completely different viewpoint: Fourier Integrals.

Let us recall

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(k) e^{-ikx} dk$$

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{ikx} dx$$

... where f(x) is an arbitrary function and F(k) is its Fourier spectrum. Note that there are several different definitions, which distinguish themselves through normalisation constants and the sign convention in the exponent.

Fourier Coefficients

... how can we express the derivative of a function using these expressions?

$$\partial_x f(x) = \partial_x \left(\int_{-\infty}^{\infty} F(k) e^{-ikx} dk \right)$$

$$= -\int_{-\infty}^{\infty} ikF(k) e^{-ikx} dk$$

... because F(k) clearly does not depend on x. Let us define ...

$$P(k) = -ik$$

... note that we use capital letters to denote fields in the wavenumber domain ...

Fourier Coefficients

... so that ...

$$\partial_x f(x) = \int_{-\infty}^{\infty} P(k)F(k)e^{-ikx}dk$$

... now a bell rings ... and we remember the Convolution Theorem which says "*a multiplication in the wavenumber domain is a convolution in the space domain*" which can be expressed as

$$\partial_x f(x) = \int_{-\infty}^{\infty} p(x-x')f(x')dx$$

... note the small letters as we are now in the space domain!

In the discrete and band-limited world this integral turns into a convolution sum. This is the most general way of describing a differential operator. It comprises all the cases from two-point, local operators up to the *exact* spectral operator.

Fourier Coefficients

.. we now want to express the operator p(x) in the space domain ...
we first have to get rid of the infinities as we are in a discrete domain where we have a maximum frqeuency (wavenumber), the **Nyquist frequency.**
This band-limitation can be expressed using Heaviside functions.

$$H(x)=\begin{cases}1 & \text{if } x>0\\ 0 & \text{if } x<0\end{cases}$$

in our example the limitation in *k* can be expressed as

$$P(k)=ik\left(H(k+k_{Ny})+H(k-k_{Ny})\right)$$

we now have to transform this back into the space domain

$$p(x)=\int_{-\infty}^{\infty}P(k)e^{-ikx}dk$$

Fourier Coefficients

... to obtain ...

$$p(x) = \frac{1}{\pi x^2}\left[\sin(k_{Ny}x) - k_{Ny}x\cos(k_{Ny}x)\right]$$

... in a staggered scheme we need to discretise space like ...

$$\begin{cases} x_{n+1/2} = (n+1/2)dx \\ k_{Ny} = \pi / dx \end{cases}$$

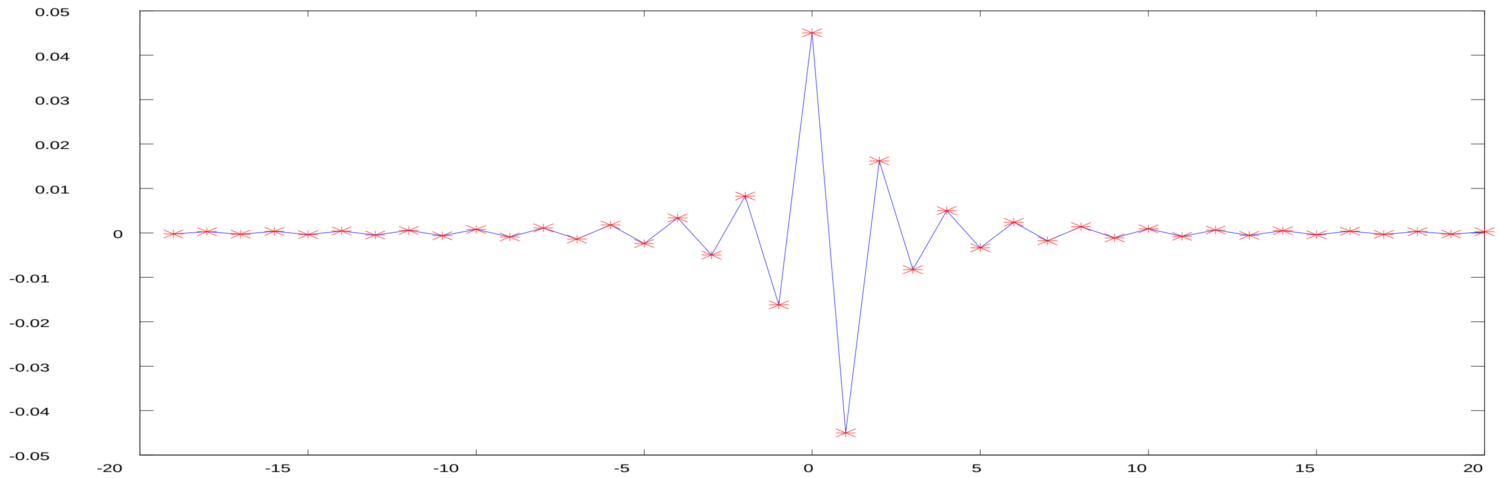
... leading to ...

$$p(x_n) = \frac{(-1)^n}{\pi((n+1/2)dx)^2}$$

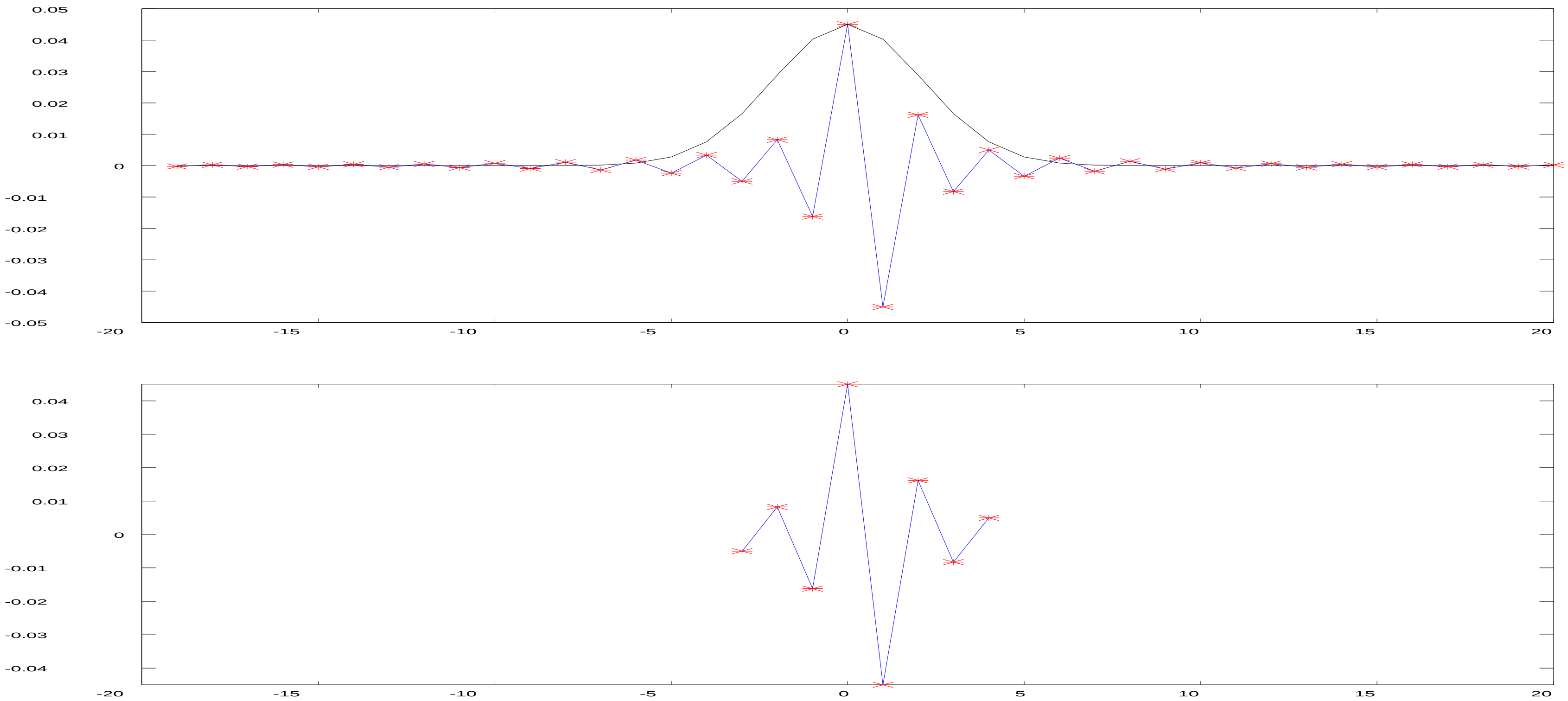
Fourier Coefficients

$$p(x_n) = \frac{(-1)^n}{\pi((n+1/2)dx)^2}$$

... these are our differential weights ...



to shorten the operator we taper with a Gaussian function



Fourier Coefficients - Interpolation

... the same approach can be applied to the problem of interpolation ...

$$f(x+dx/2)=\int_{-\infty}^{\infty}F(k)e^{-ik(x+dx/2)}dk$$

$$=\int_{-\infty}^{\infty}F(k)e^{-ikdx/2}e^{-ikx}dk$$

$$=\int_{-\infty}^{\infty}F(k)I(k)e^{-ikx}dk$$

$$I(k)=e^{-ikdx/2}$$

i.e. I(k) is now our interpolation operator expressed in the wavenumber domain. Again we are looking for the equivalent representation in the space domain, which we get by inverse Fourier Transform

Fourier Coefficients - Interpolation

... in the band-limited world our operator is ...

$$I(k) = e^{-ikdx/2}\left(H(k + k_{Ny}) + H(k - k_{Ny})\right)$$

... which in the space domain yields ...

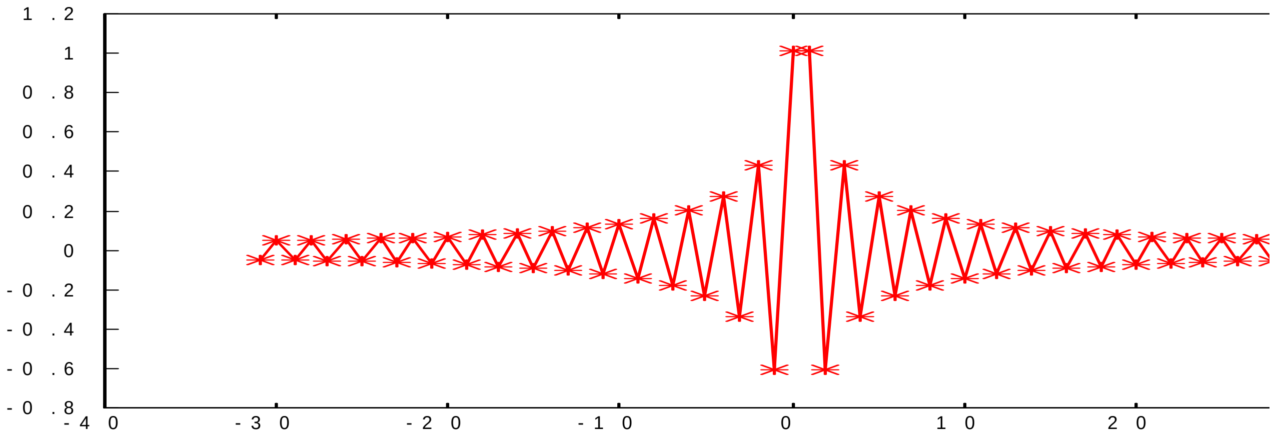
$$i(x) = \frac{\sin(k_{Ny}(x + dx/2))}{\pi(x + dx/2)}$$

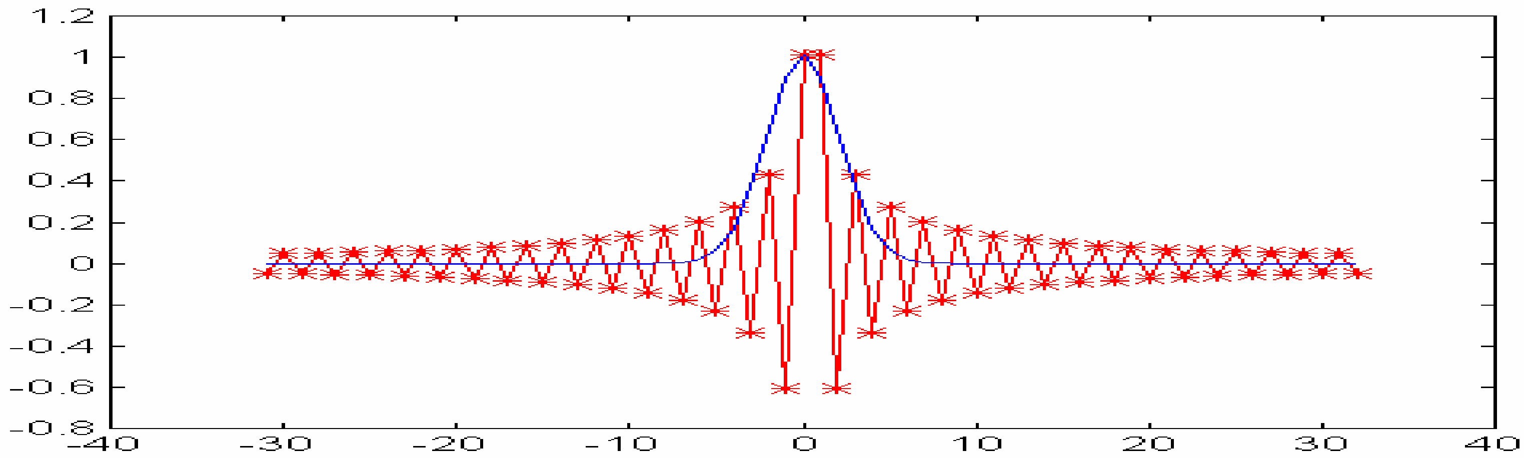
discretising with

$$\begin{cases} x_{n+1/2} = (n+1/2)dx \\ k_{Ny} = \pi/dx \end{cases}$$

we obtain

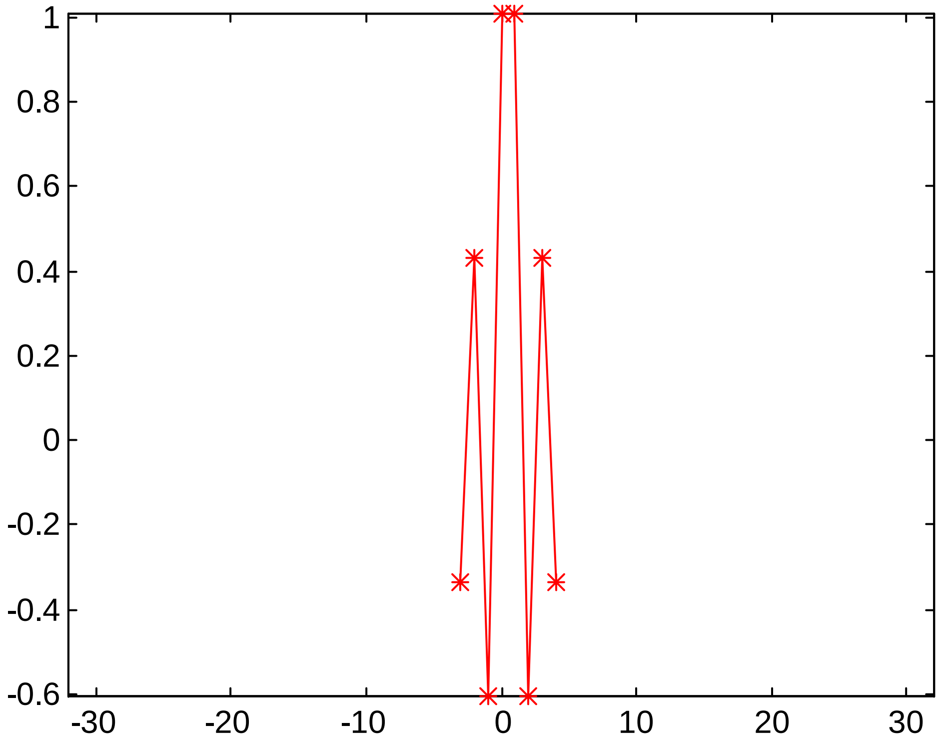
$$i(x_n) = \frac{(-1)^n}{\pi dx(n+1/2)}$$


1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-40
-30
-20
-10
0
10
20


1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-40
-30
-20
-10
0
10
20
30
40

Fourier Coefficients - Interpolation

the final 8-point operator



... as mentioned earlier the derivative operator in the wavenumber domain is

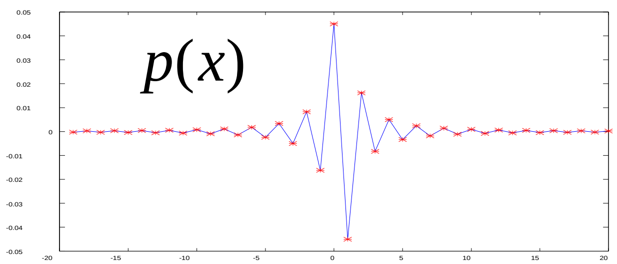
$$\partial_x f(x) = \partial_x \left(\int_{-\infty}^{\infty} F(k) e^{-ikx} dk \right)$$

$$= -\int_{-\infty}^{\infty} ikF(k) e^{-ikx} dk$$

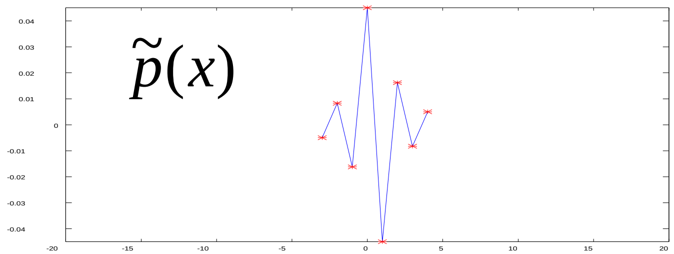
$$\boxed{P(k) = -ik}$$

... which in the space domain led to the convolutional operator p(x) looking like

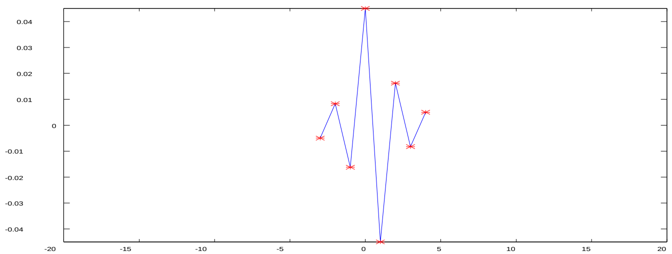
exact



shortened

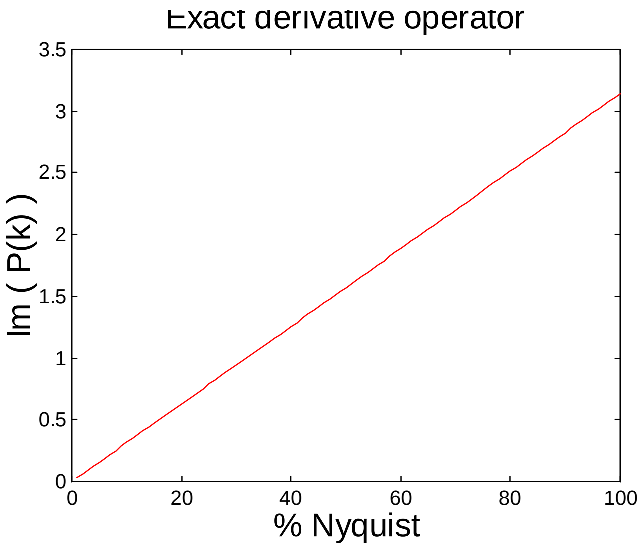


... this suggests that we can now Fourier transform our shortened operator and compare it with the exact one in the k-domain ...



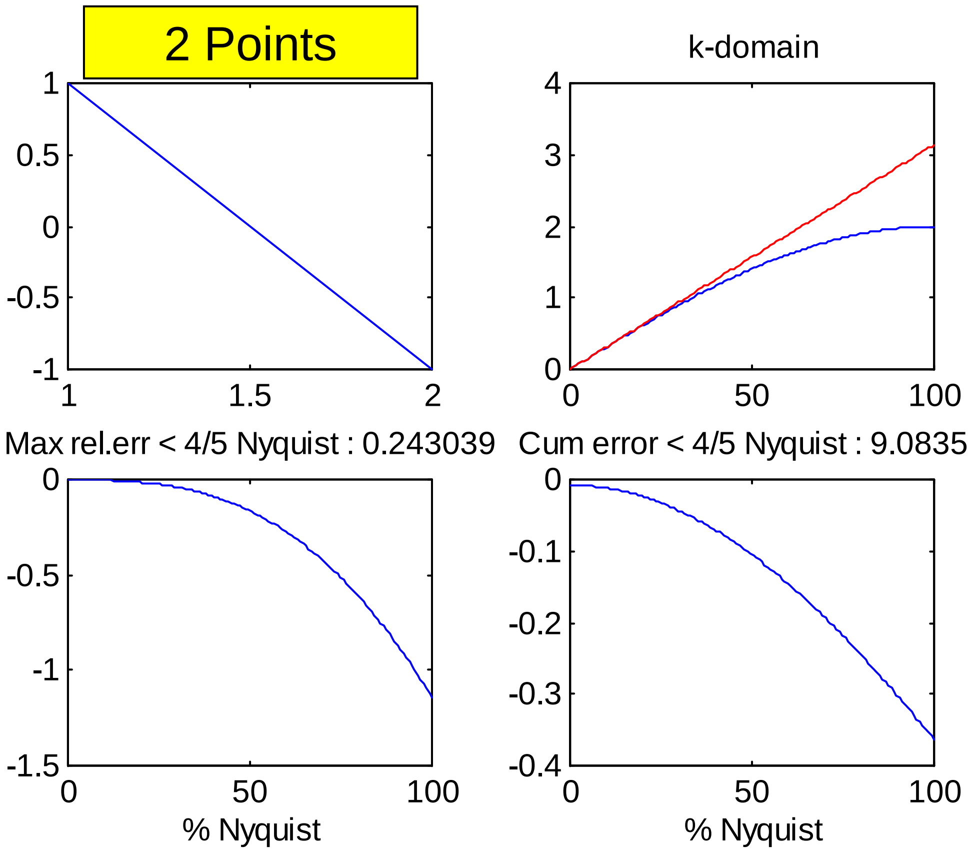
$$FFT(\tilde{p}(k)) = -i\tilde{k}$$

... which also means that we now have a *numerical* wavenumber as a function of the *exact* wavenumber we propagate in our system.



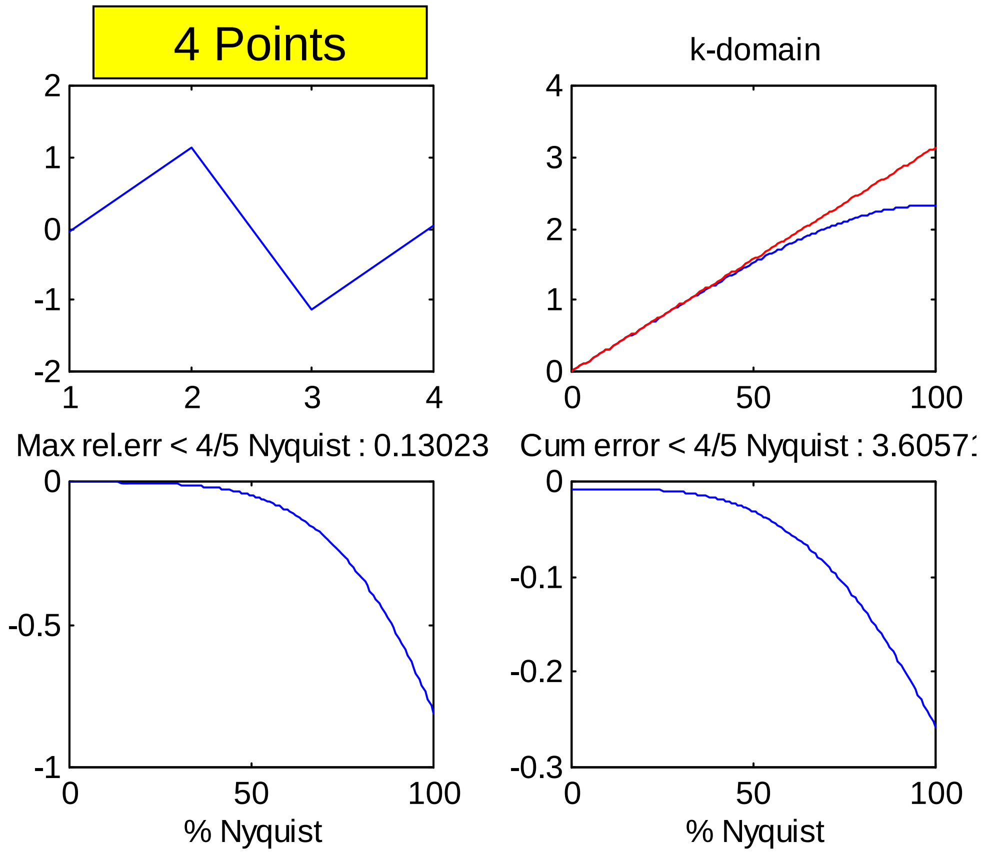
High-order operators - Accuracy

... we can now compare some of our high-order Taylor operators with the exact operator in the wavenumber domain ...

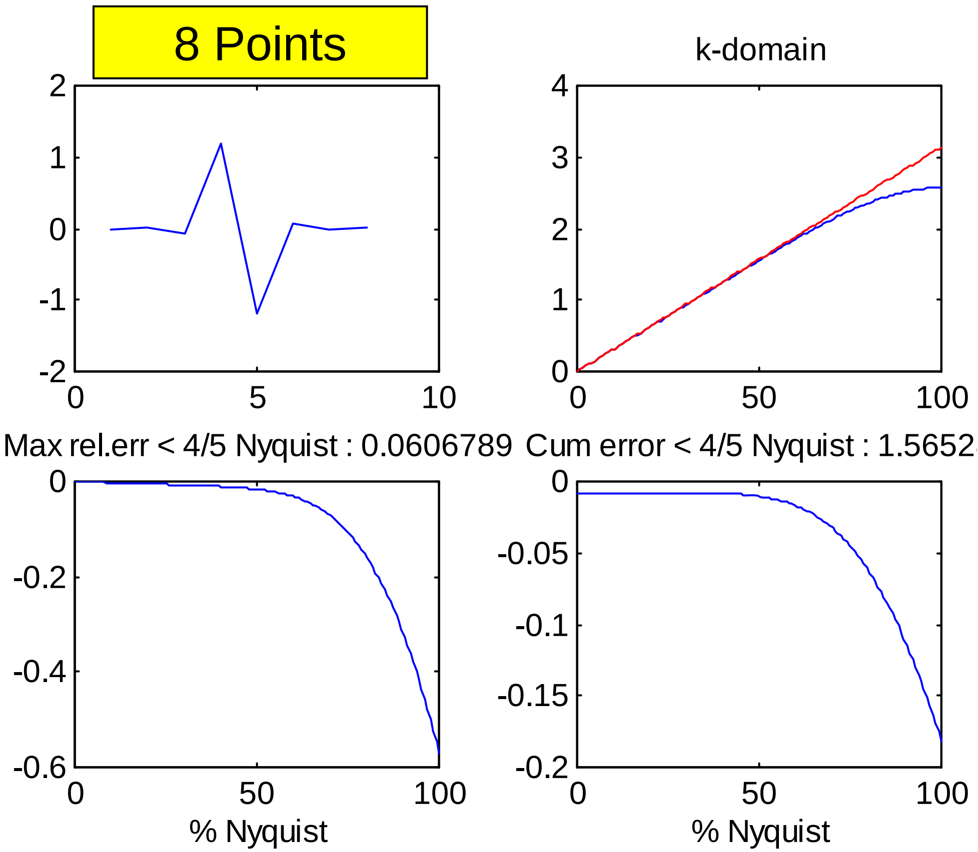


High-order operators - Accuracy

... we can now compare some of our high-order Taylor operators with the exact operator in the wavenumber domain ...

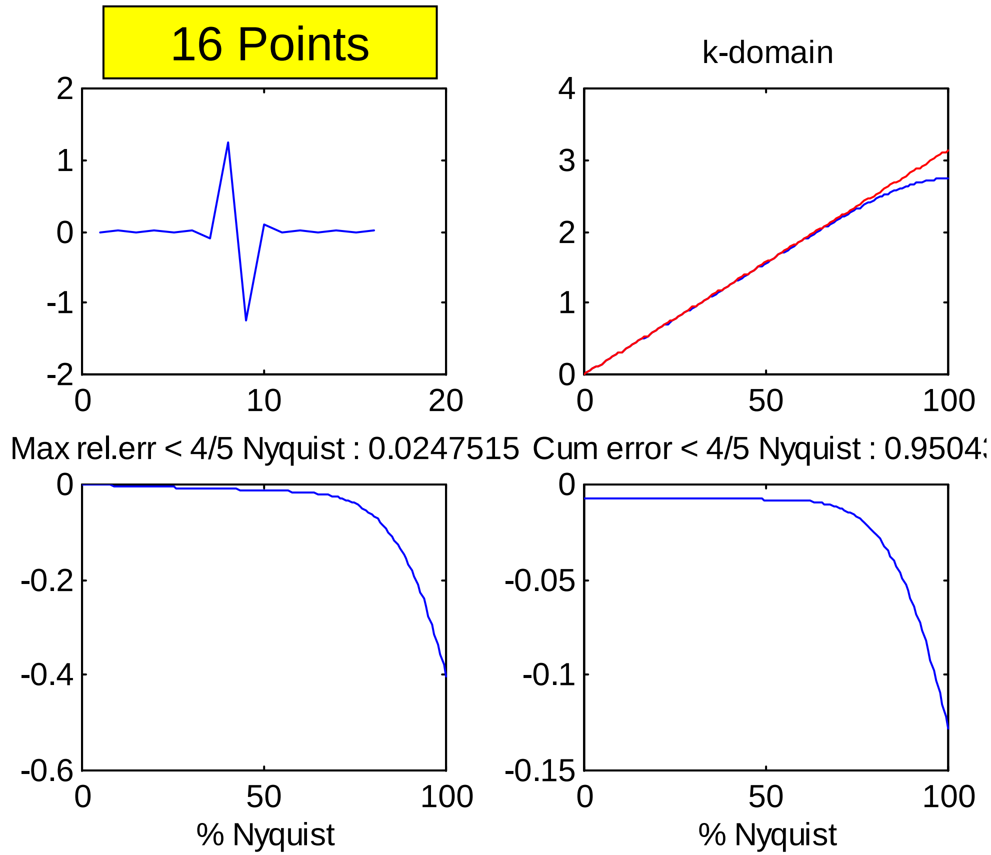


... we can now compare some of our high-order Taylor operators with the exact operator in the wavenumber domain ...



High-order operators - Accuracy

... we can now compare some of our high-order Taylor operators with the exact operator in the wavenumber domain ...



Fourier Coefficients

Summary

Finite-difference operators can be regarded as (truncated) spectral (global) operators in a general way. In pseudo-spectral methods, the space derivatives are calculated in the wavenumber domain.

The length of the operator determines its accuracy.

FD improving accuracy Spectral

2 increasing length of operator nx