FFT, wave numbers and IDL: IDL's `FFT()` function is pretty straight-forward to use, apart from the second argument which may be a bit confusing. E.g. `FFT(f)` and `FFT(f,-1)` return the forward transform of f, while `FFT(f,1)` will return the backward transform (which applies a normalization factor and is really the inverse of the forward transform). The ± 1 in the second argument represent the sign of the exponent $\pm 2\pi i \mathbf{k} \cdot \mathbf{x}$.

The trivial but somewhat tedious part to be still done by the user is to construct the appropriate grid of $\mathbf{k}$ values. Let $L_x \equiv N_x \delta x$ be the extent of the x interval (i.e. the period length in x). Then k changes in steps of $\delta k = 2\pi / L_x$. IDL treats k (in some sense) as starting at $k_{\min} = 0$ and increasing monotonically until $k_{\max} = 2\pi/\delta x$. However, k and $k - 2\pi/\delta x$ are equivalent and normally we want k to go from $-k_{\mathrm{Ny}}$ to $k_{\mathrm{Ny}} = \pi/\delta x$. This can be achieved by

```
dk = 2*!pi/Lx
k1 = (findgen(Nx)-Nx/2)*dk
k  = shift(k1, -Nx/2)
```

When plotting spectral information over k,

```
spect = abs(fft(f,-1))
plot, k, spect
```

the plot will be OK, apart from a gap around $k = 0$ and a spurious line connecting the leftmost and the rightmost point. This is because k is piecewise linear in two intervals and is discontinuous at the interface (`plot, k` to see this).

This (`k` and `spect` in the example) is the representation that is most useful for everything but plotting spectra (or anything else in Fourier space). E.g. filtering is easily done using k, and it often helps that we know that $k = 0$ is at position `k[0]`.

For plotting spectra, we use k_1, which is a continuous version of k. We then *must* shift the spectral information:

```
spect1 = shift(spect,Nx/2)
plot, k1, spect1
```

Note: The k and k_1 used here are *physical* wave numbers, and are thus not identical to the ones (unfortunately) introduced in the lecture. Using physical wave numbers, the Laplace equation $\Delta f = g$ becomes

$$-k^2 \tilde{f} = \tilde{g}$$

in Fourier space – without the factor $4\pi^2$.

Question 1 *Fourier transform*

Use IDL's `fft()` function to calculate the (discrete) Fourier transforms of the functions

$$\begin{aligned} f_0(x) &= 1 \,, \\ f_1(x) &= \sin 2\pi x \,, \\ f_2(x) &= \cos 2\pi x \,, \\ f_3(x) &= \frac{e^{-(x-x_0)^2/(2w^2)}}{\sqrt{2\pi w^2}} \,, \qquad x_0 = \frac{2}{3}, w = 0.03 \,, \\ f_4(x) &= \cos 2\pi x \,, \end{aligned}$$

on the interval $0 \le x < 1$.

(a) Plot modulus $|\tilde{f}_n(k)|$, real part $\Re\tilde{f}_n(k)$, imaginary part $\Im\tilde{f}_n(k)$ and phase $\arg\tilde{f}_n(k)$.

(b) Make sure that the normalization and range of k is correct. For one of the functions (not $f_0 \ldots$), change the interval length and verify that k is still correct.

(c) Apply a low-pass filter to f_4 to screen out any wave number $k > 50$, transform back, plot $f_4(x)$ and overplot the filtered f_4.

Question 2 *Spectral method in 1-d*

Use the spectral method to solve the one-dimensional stationary heat conduction equation

$$\frac{d^2T}{dx^2} = -q(x)$$

with

$$q(x) = \frac{e^{-(x-x_0)^2/(2w^2)}}{\sqrt{2\pi w^2}}$$

on the interval $0 \le x < 1$. Assume periodic boundary conditions $T(x) = T(1)$.

(a) Use the forward transform `fft(f,-1)` on the right hand side $-q$.

(b) Construct an array `k_2` that is equal to $1/k^2$ for $|k| \neq 0$ and is 0 for $k = 0$.

(c) Multiply $-\tilde{q}$ by `k_2` and transform back using `fft(f,1)`.

(d) Plot the solution $T(x)$. Does this match your expectations?

(e) Plot the second derivative of $T(x)$,

```
plot, x, wdderiv2(x,Temp)
```

and compare to $-q(x)$. Can you explain?

Question 3 *Tridiagonal solver*

(a) Use CVS to check out my tridiagonal solver in F90 from `wdobler/f90/lib`, and the corresponding test program and Makefile from `wdobler/f90/test`.

Hint: You can modify files in these directories and check the modifications in, but I will not like that (unless you really fixed a bug or added something really clever)! It is thus best if you check out the files into a place where you will not edit or compile, and copy them over to your playground.

(b) Try to understand the Makefile, compile and run the test. Try to understand the test.

(c) Use the solver to solve the system

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -\mathcal{C}/2 & 1+\mathcal{C} & -\mathcal{C}/2 & 0 & 0 & 0 \\ 0 & -\mathcal{C}/2 & 1+\mathcal{C} & -\mathcal{C}/2 & 0 & 0 \\ 0 & 0 & -\mathcal{C}/2 & 1+\mathcal{C} & -\mathcal{C}/2 & 0 \\ 0 & 0 & 0 & -\mathcal{C}/2 & 1+\mathcal{C} & -\mathcal{C}/2 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \vec{T} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

with $\mathcal{C} = 0.3$

(d) Repeat this in IDL, using ‘ `wdobler/idl/lib/diverse/solve_tridiag.pro`’.