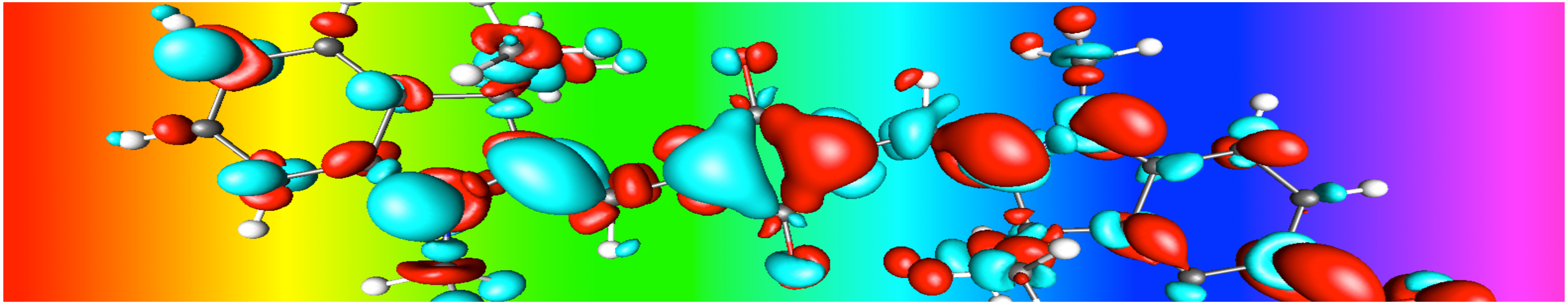
ICTP-SISSA Master in High Performance Computing
Trieste, 2016-17

Fourier Transforms and Fast Fourier Transforms



Ralph Gebauer

ICTP
The Abdus Salam
International Centre
for Theoretical Physics

Monday, April 10th, 2017

What is a Fourier transform?

Given a c-valued function f(x), with

$$\int_{-\infty}^{\infty} |f(x)|\, dx \quad < \quad \infty$$

Fourier transform:

$$F(q) = \int_{-\infty}^{\infty} e^{-iqx} f(x)\, dx$$

Inverse Fourier transform:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{iqx} F(q)\, dq$$

Properties of FT:

f(x) is real $\Rightarrow$ $F(-q) = F(q)^*$

f(x) is imaginary $\Rightarrow$ $F(-q) = -F(q)^*$

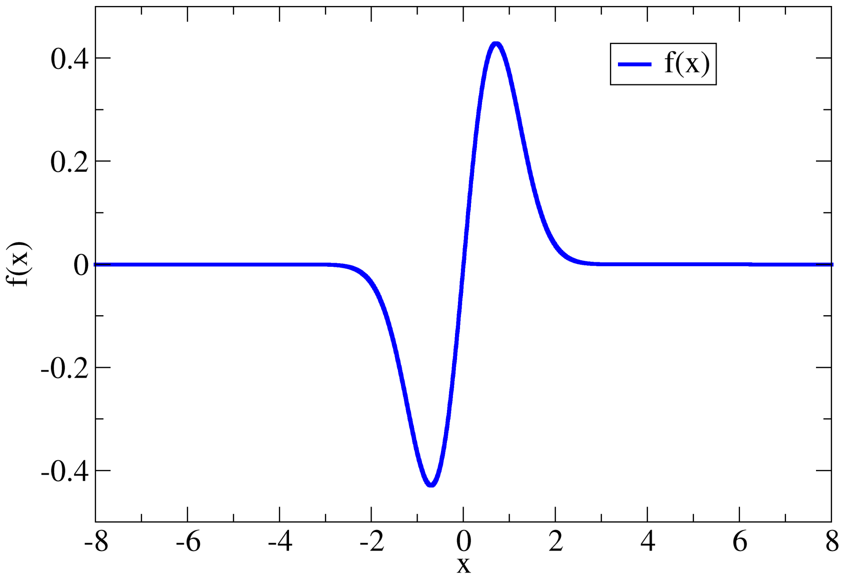
f(x) is even $\Rightarrow$ F(q) is even

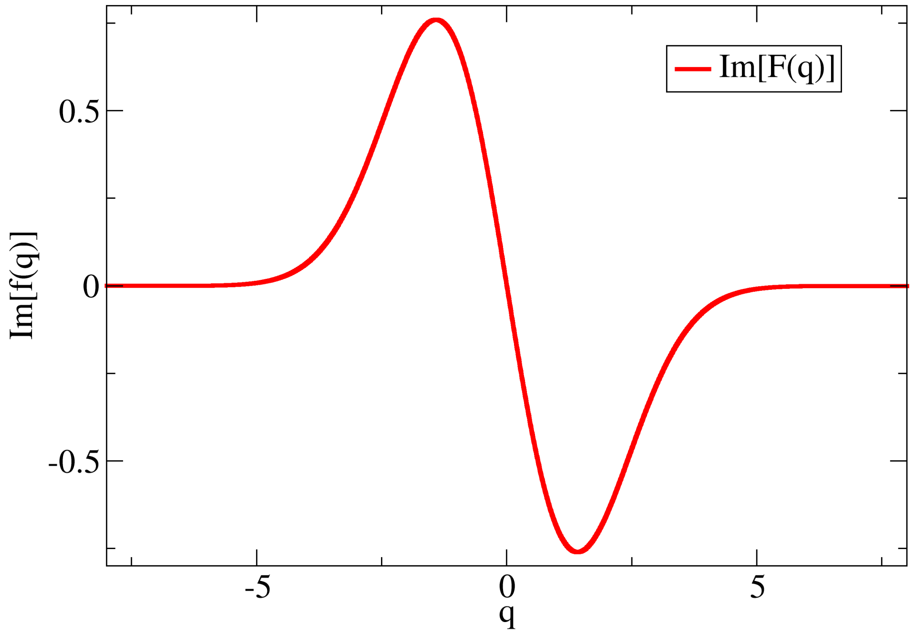
f(x) is odd $\Rightarrow$ F(q) is odd

Example 1:

$$F(x) = x\,e^{-x^2}$$

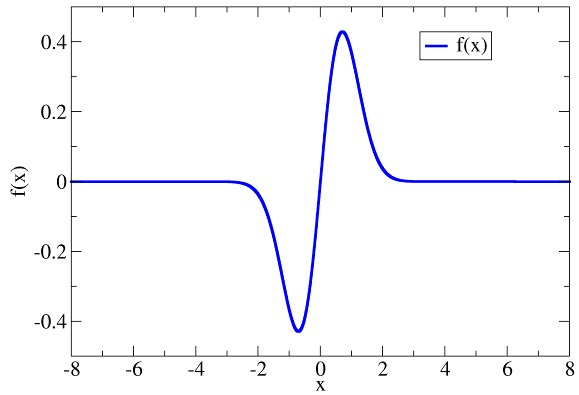
$$F(q) = -\frac{\sqrt{\pi}}{2}\,i\,q\,e^{-q^2/4}$$

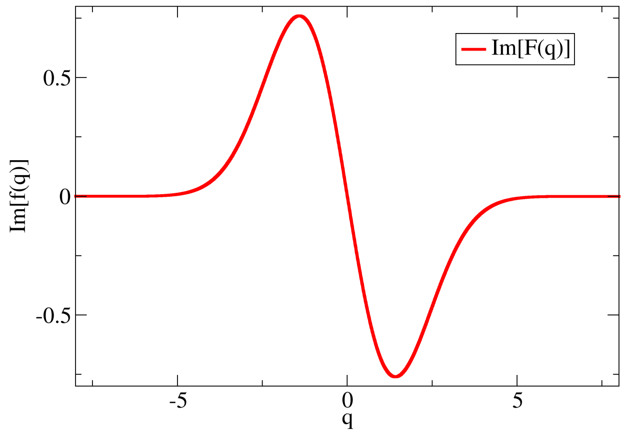




Observe here some general properties of FT's:

$$f(x) = x\, e^{iqx} \qquad\qquad F(q) = -\frac{\sqrt{\pi}}{2} i\, q\, e^{-q^2/4}$$



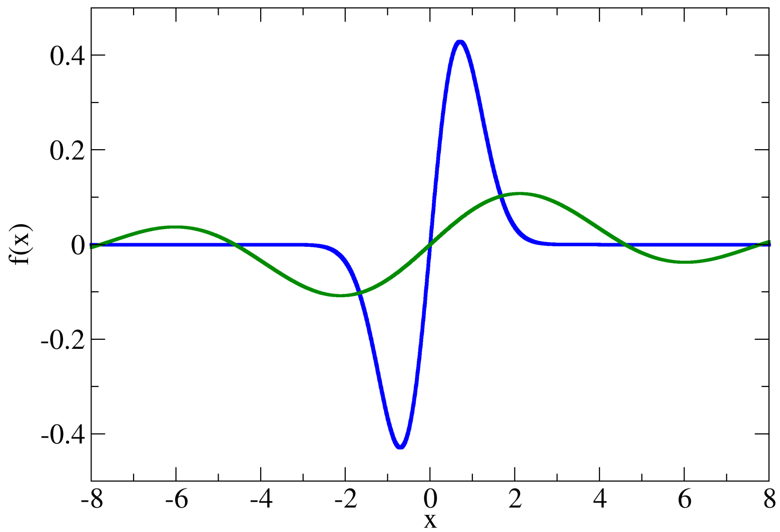


1) F(q=0) is average of f.
2) f real ⇔ F(-q) = F(q)*

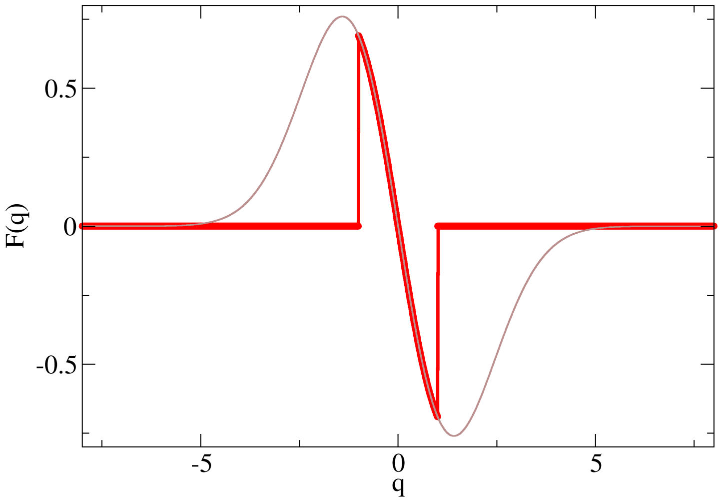
3) f odd ⇔ F imaginary

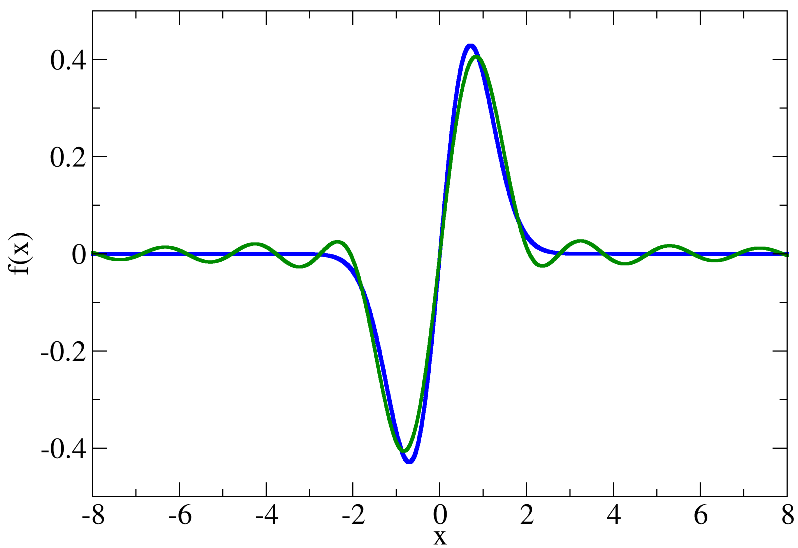
Restricting the information from the FT:

$f(x)$ $F(q)$

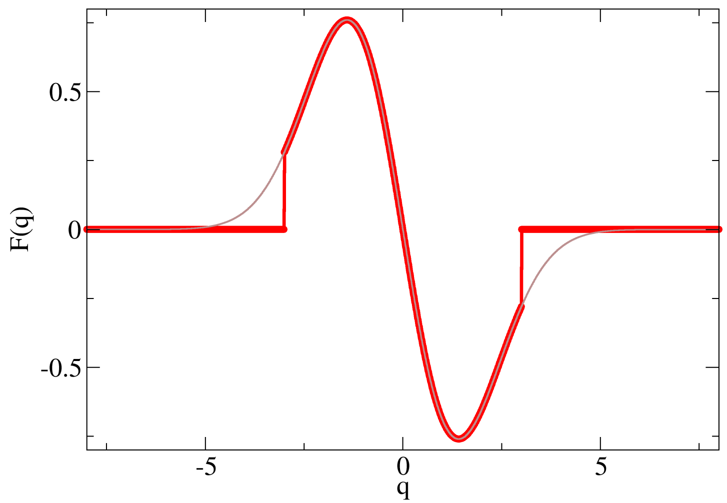


$$|q| < 1$$

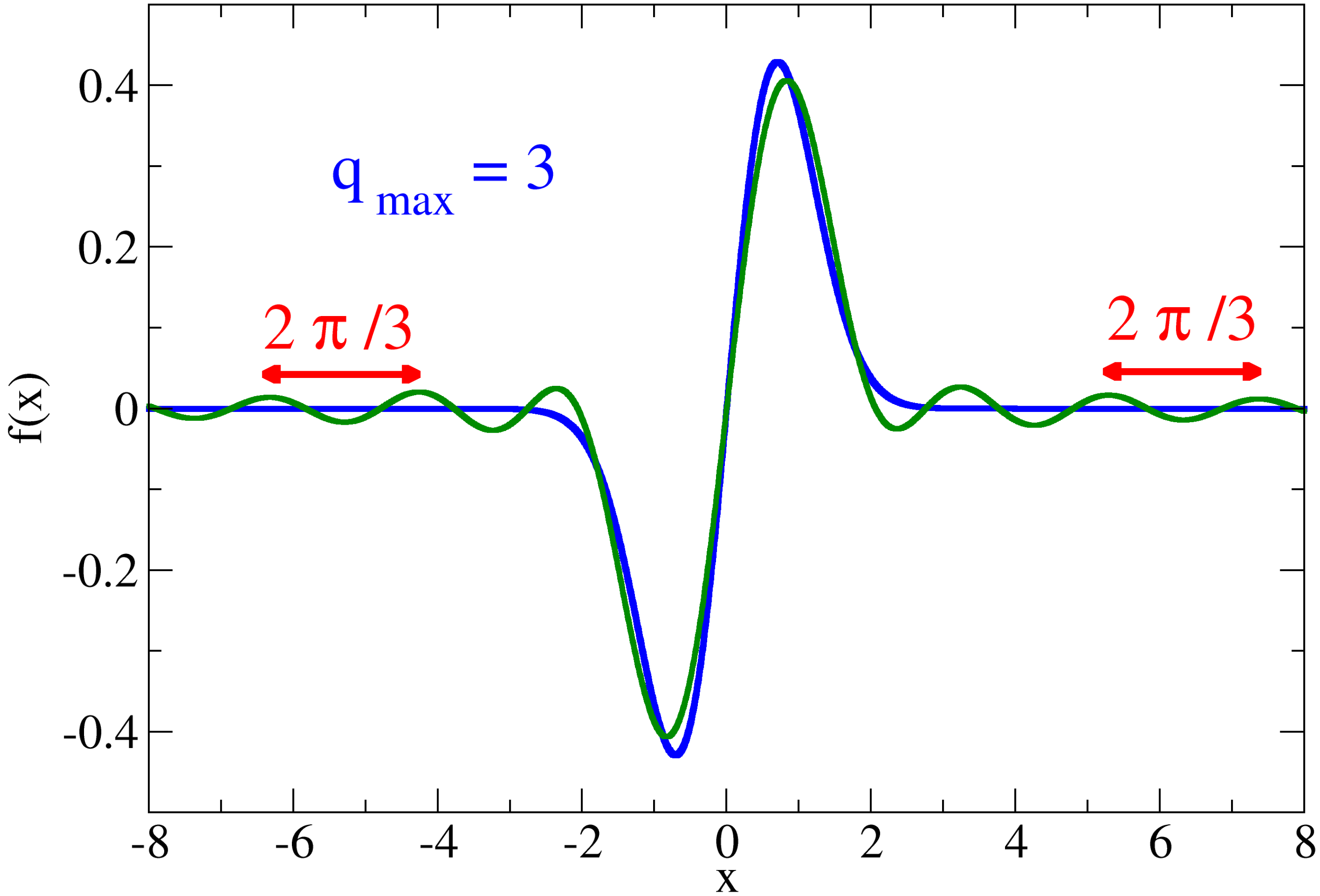




$$|q| < 3$$



FT bandwidth and spacial resolution



7

Fourier transforms and derivatives

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{iqx}\, F(q)\, dq$$

$$\begin{aligned} \frac{d}{dx} f(x) &= \frac{1}{2\pi} \frac{d}{dx} \int_{-\infty}^{\infty} e^{iqx}\, F(q)\, dq \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{iqx}\, (iq)\, F(q)\, dq \end{aligned}$$

$$\frac{d}{dx} \Longleftrightarrow iq$$

Fourier transforms and convolutions

A convolution: $$(f \star g)(x) = \int_{-\infty}^{\infty} f(x')\, g(x - x')\, dx'$$

FT of a convolution: $$FT\left[(f \star g)\right](q) = F(q) \cdot G(q)$$

Important example: $$v(\vec{r}) = \int d^3\vec{r'}\, \frac{\rho(\vec{r'})}{\left|\vec{r} - \vec{r'}\right|}$$

Note that $$FT\left[\frac{1}{|\vec{r}|}\right] = \frac{4\pi}{|\vec{q}|^2}$$

Therefore: $$V(\vec{q}) = 4\pi \frac{\tilde{\rho}(\vec{q})}{|\vec{q}|^2}$$

(Very effective to solve Poisson equation!)

Fourier transforms of periodic functions

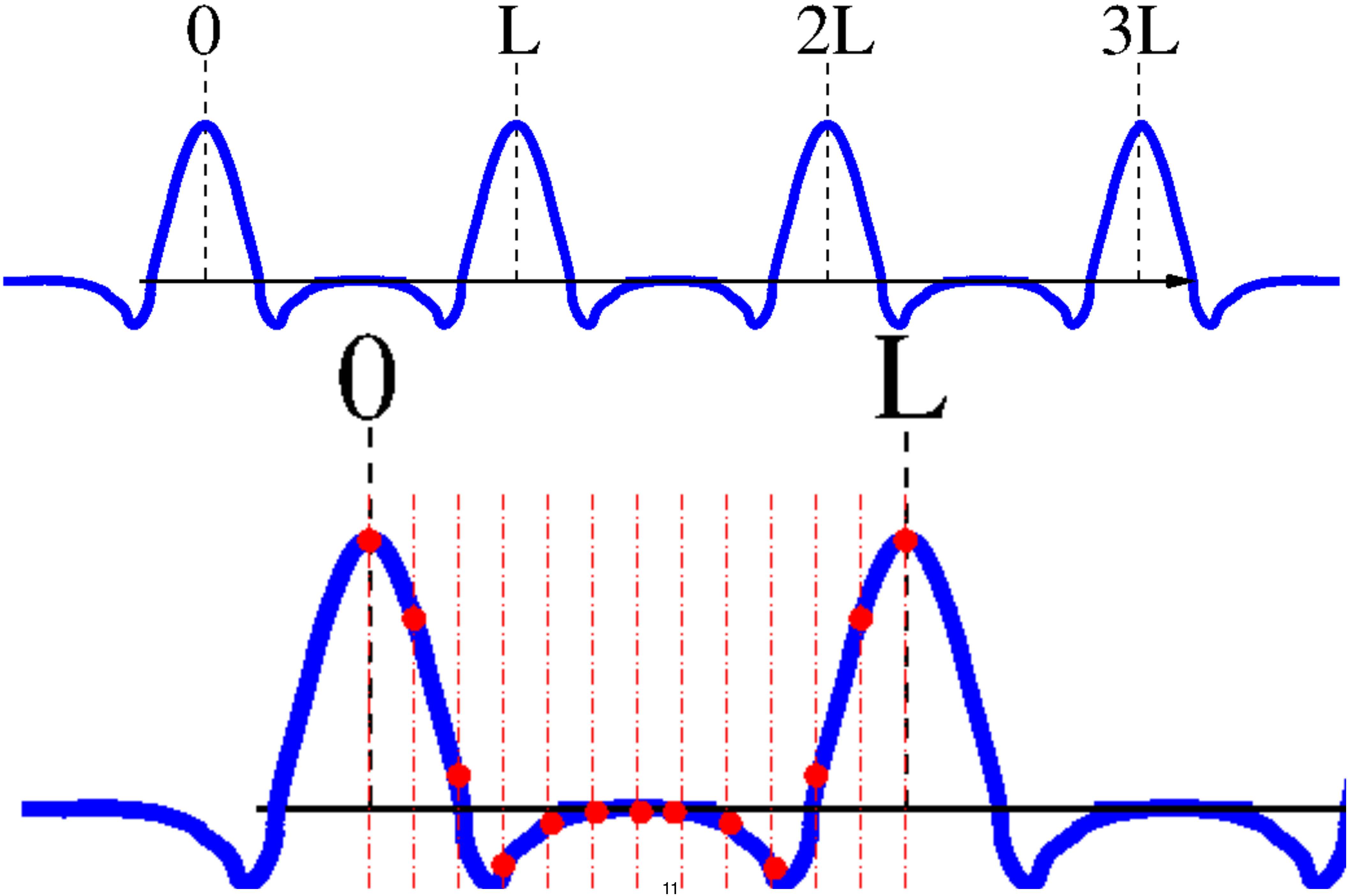
If f(x) is periodic with periodicity L, then f(x+L) = f(x).

Non-zero Fourier coefficients only for:

$$q_n = \frac{2\pi}{L} n, \qquad n \in \mathbb{Z}$$

Therefore, in periodic functions, we obtain a Fourier series.

Sampling periodic functions



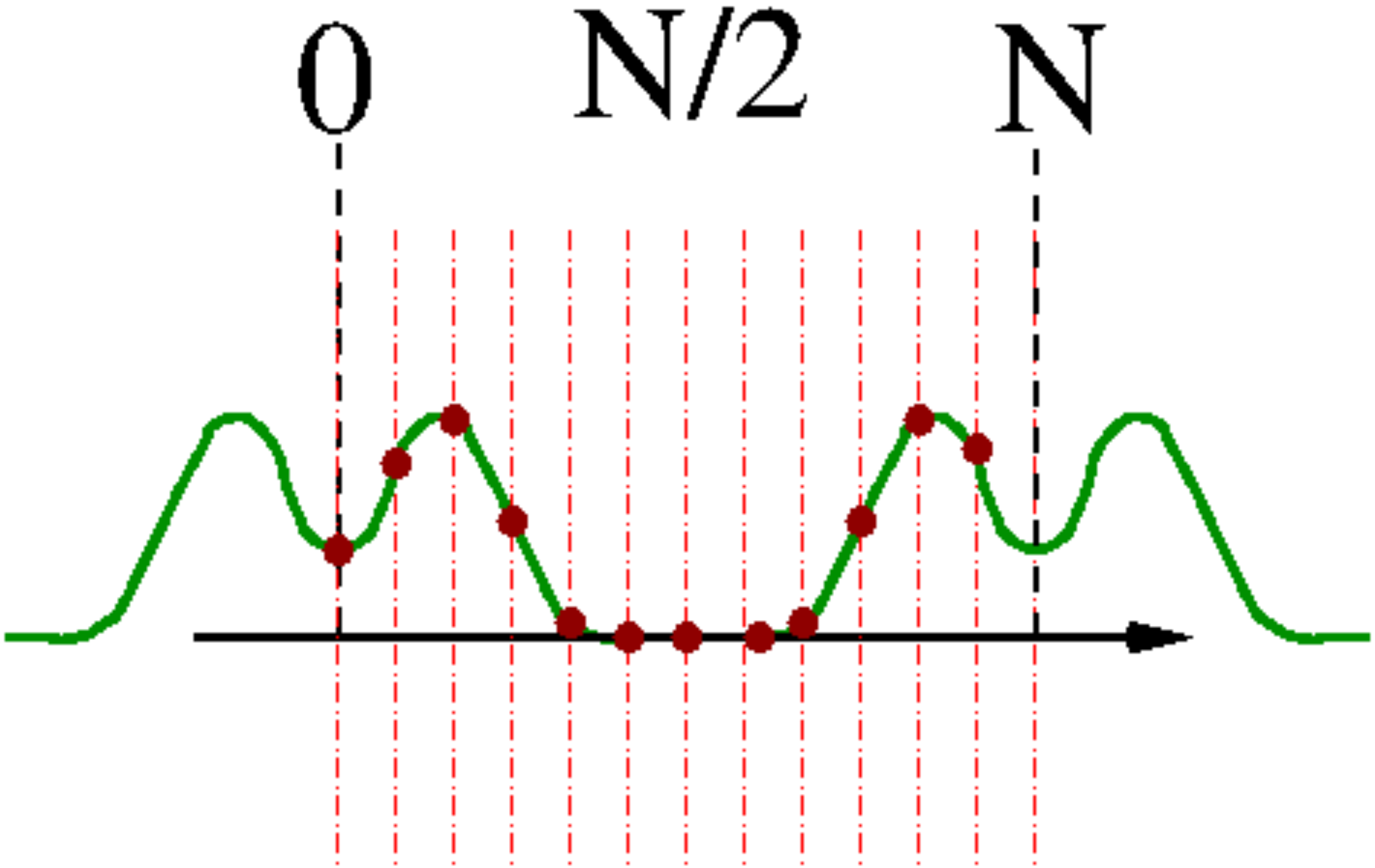
Fourier series

$$F(k) = \sum_{l=0}^{N-1} e^{-2\pi i \frac{k \cdot l}{N}} f(l)$$

$$f(l) = \frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi i \frac{k \cdot l}{N}} F(k)$$

Periodicity of coefficients

$$F(k+N) = F(k)$$



13

How are discrete Fourier transformations computed?

$$F(k) = \sum_{l=0}^{N-1} f(l)\, e^{-2\pi i \frac{k \cdot l}{N}}$$

Looks like an O(N^2) task:

→ N values of k,
→ for each value of k sum over N terms

Fast Fourier transformations (Cooley–Tukey FFT algorithm)

(In the following, we assume that $N=2^m$, i.e. a power of 2)

$$
\begin{aligned}
F(k) &= \sum_{l=0}^{N-1} f(l)\, e^{-2\pi i \frac{k \cdot l}{N}} \\
&= \sum_{l=0}^{N/2-1} f(2l)\, e^{-2\pi i \frac{k \cdot 2l}{N}} + \sum_{l=0}^{N/2-1} f(2l+1)\, e^{-2\pi i \frac{k\cdot(2l+1)}{N}} \\
&= \sum_{l=0}^{N/2-1} f(2l)\, e^{-2\pi i \frac{k \cdot l}{N/2}} + e^{-2\pi i \frac{k}{N}} \sum_{l=0}^{N/2-1} f(2l+1)\, e^{-2\pi i \frac{k \cdot l}{N/2}}
\end{aligned}
$$

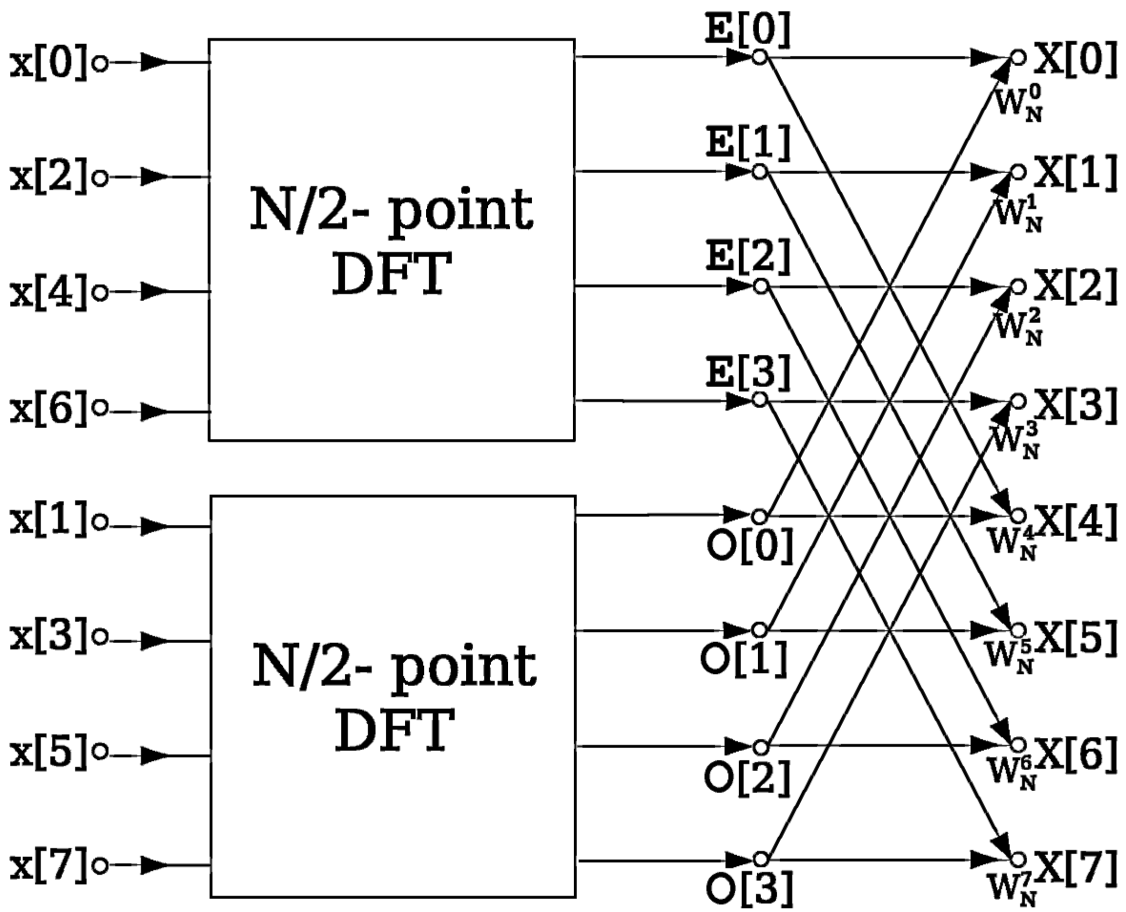
Fast Fourier transformations (Cooley–Tukey FFT algorithm)

$$
\begin{aligned}
F(k) &= \sum_{l=0}^{N/2-1} f(2l)\, e^{-2\pi i \frac{k\cdot l}{N/2}} + e^{-2\pi i \frac{k}{N}} \sum_{l=0}^{N/2-1} f(2l+1)\, e^{-2\pi i \frac{k\cdot l}{N/2}} \\
&= \begin{cases} E_k + e^{-2\pi i \frac{k}{N}}\, O_k & \text{for } 0 \le k < N/2 \\ E_{k-N/2} + e^{-2\pi i \frac{k}{N}}\, O_{k-N/2} & \text{for } N/2 \le k < N \end{cases}
\end{aligned}
$$

Therefore, for 0 ≤ k < N/2, we have:

$$
\begin{aligned}
F(k) &= E_k + e^{-2\pi i \frac{k}{N}} O_k \\
F(k+N/2) &= E_k - e^{-2\pi i \frac{k}{N}} O_k
\end{aligned}
$$

Fast Fourier transformations (Cooley–Tukey FFT algorithm)



The issue with real vs. complex FTs

The FT of f(x) is complex, even if f(x) is a real function.

This seems to imply that more information is contained in F(q) than in f(x).

This is not the case: $F(q) = F(-q)^*$

However, the computational load of a real FT is the same as of a complex FT. Also the memory requirements are those of a complex function.

What can we do about this?

The issue with real vs. complex FTs

Solution 1: Ignore the "problem"

Solution 2: Two real FFTs in one shot:

f(x) and g(x) are real functions.
Define the auxiliary function a(x) = f(x) + i g(x)

$$
\begin{aligned}
A(q) &= F(q) + iG(q) \\
A(-q) &= F(-q) + iG(-q) \\
A(-q)^* &= F(q) - iG(q) \\
F(q) &= \frac{1}{2}\left(A(q) + A(-q)^*\right) \\
G(q) &= \frac{1}{2i}\left(A(q) - A(-q)^*\right)
\end{aligned}
$$

Solution 3: use special real-complex FFT subroutines:
r2c & c2r

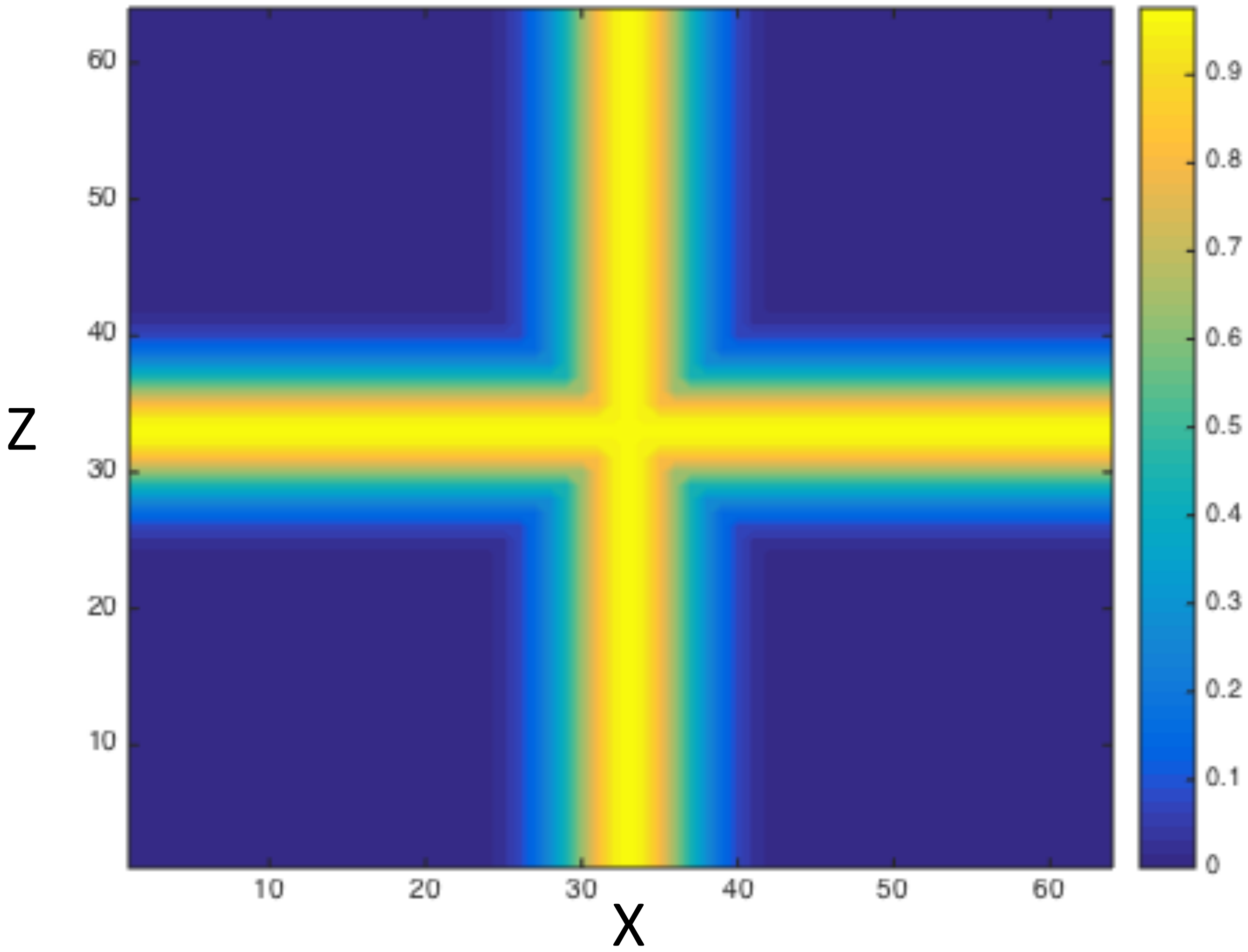
19

An example: the diffusion equation

$$\frac{\partial c(\mathbf{r},t)}{\partial t} = \nabla \cdot \left[D(\mathbf{r}) \nabla c(\mathbf{r},t) \right]$$

An example: the diffusion equation

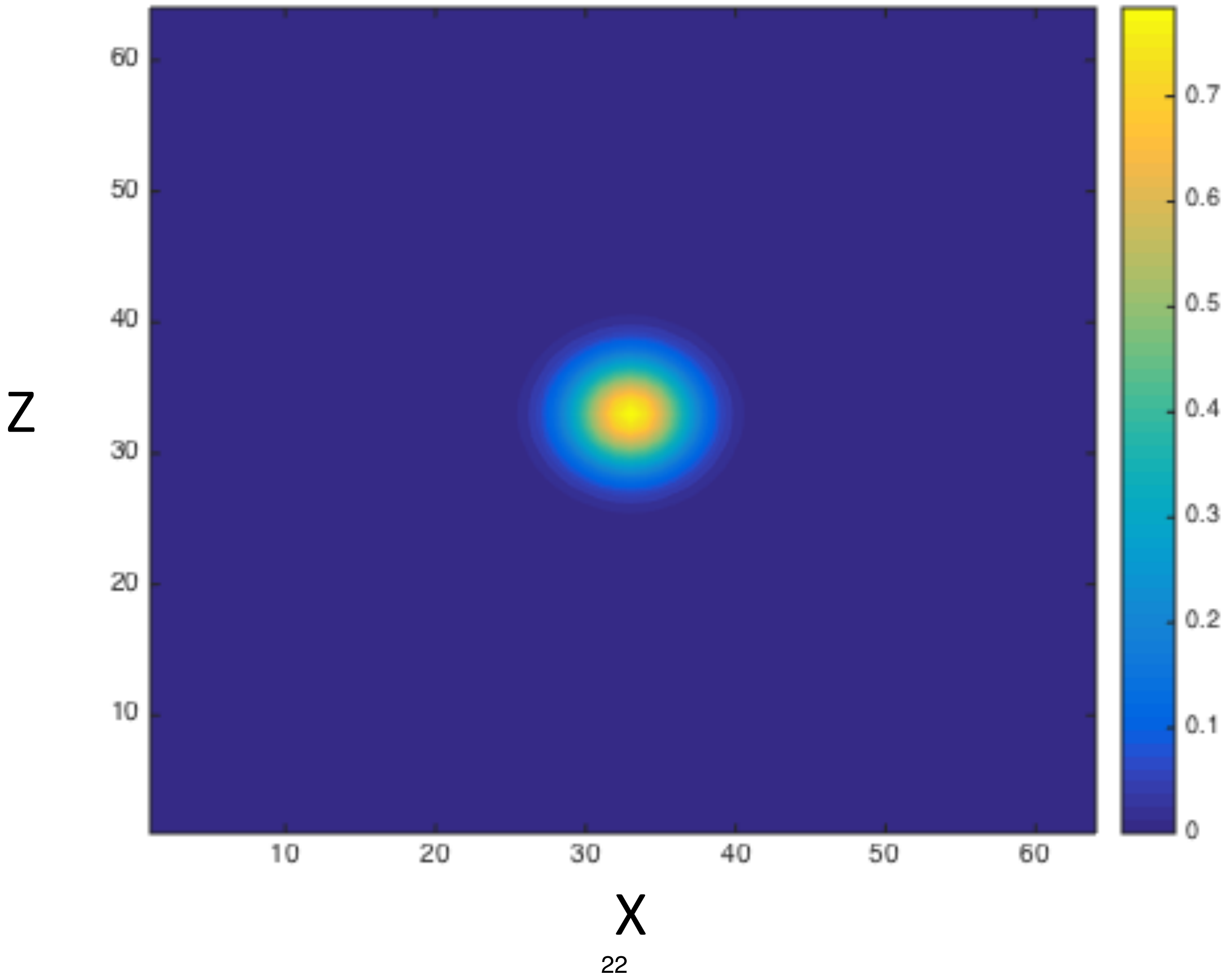
The diffusion coefficient:



21

An example: the diffusion equation

Starting concentration:



An example: the diffusion equation

Concentration after some time: