Computational Physics

Shuai Dong

Rainbow: an arch of colors

Spectral analysis

- Fourier analysis and orthogonal functions
- Discrete Fourier transform
- Fast Fourier transform

Nature vs intuition

- Nature often behaves differently to how our intuition predicts it should.

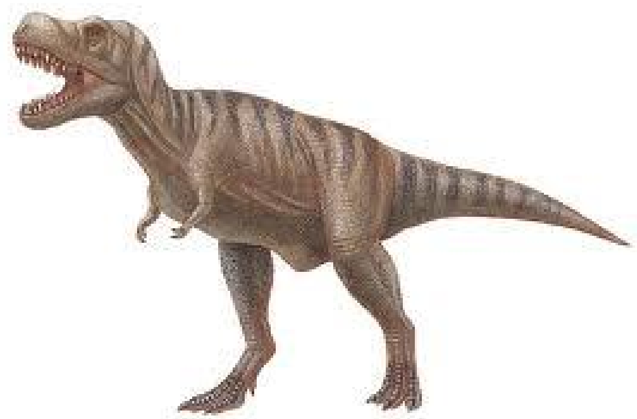
Seahorse

killer whale

Which is a fish?

Nature vs intuition

pterosaur

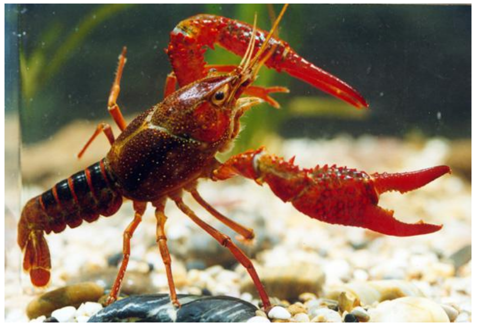


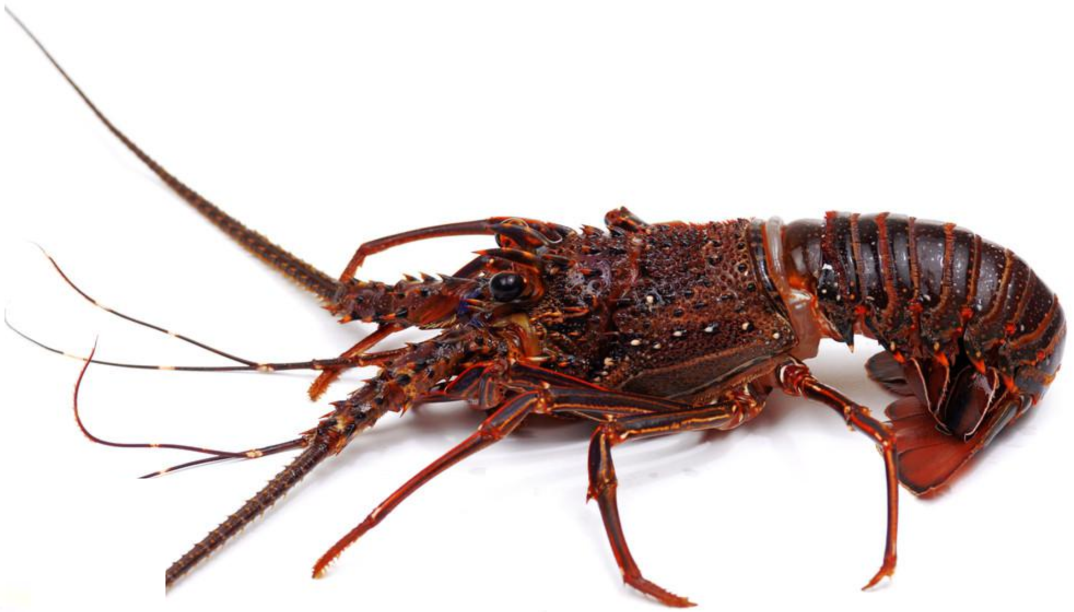
dinosaur

Which is the ancestor of birds?

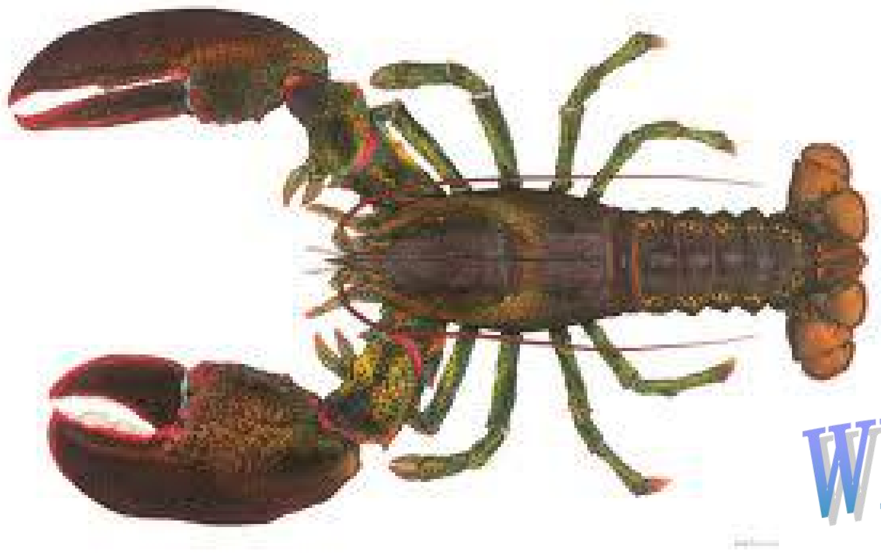
Nature vs intuition

Crayfish



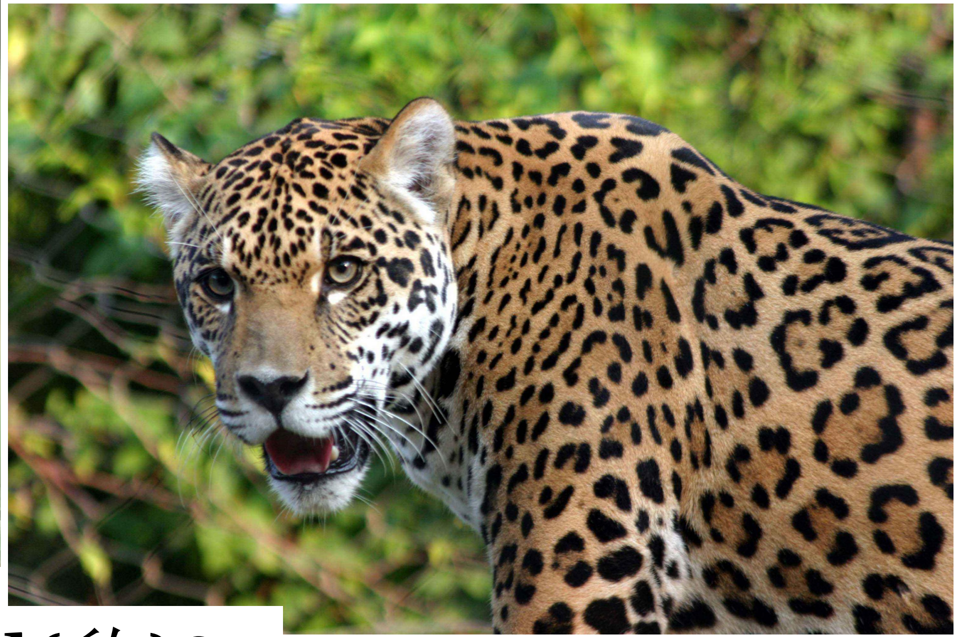


Spiny lobster



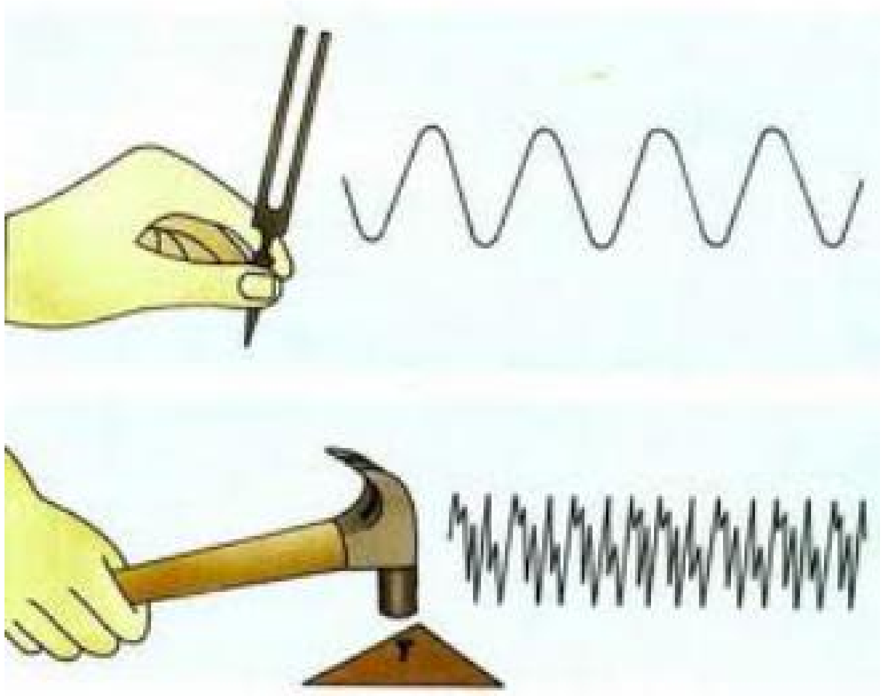
Lobster

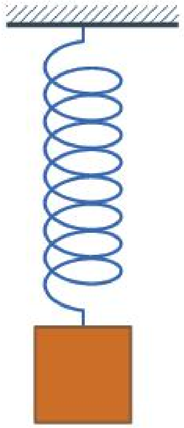
Which is 龙虾?



leopard(豹)?

Periodic motion





1/T, 2/T, 3/T, 4/T, ……

Fourier first pointed out that an arbitrary periodic function f(t), with a period T, can be decomposed into a summation of simple harmonic terms, which are periodic functions of frequencies that are multiples of the fundamental frequency 1/T of the function f(t).

Jean Baptiste Joseph Fourier

(21 March 1768 – 16 May 1830)

A

- He was a French mathematician and physicist born in Auxerre and best known for initiating the investigation of Fourier series and their applications to problems of heat transfer and vibrations.
- The Fourier transform and Fourier's Law are also named in his honor.
- Fourier is also generally credited with the discovery of the greenhouse effect.

Wide Applications

- Widely used in science & technology.
- Very very useful in signal processing!

南京大学学生凭拨号音破译周鸿祎手机号码
来源：北京晚报 (2012-09-01)

这位大学生是来自南京大学软件工程专业的刘靖康。奇虎360和百度发生搜索战后，优酷网上出现了一段采访周鸿祎的视频。在该段视频中，拨打周鸿祎电话时，画面上并没有出现拨号的过程，只呈现了拨号声。

- 刘靖康截取了这段10秒钟的拨号音，“我们平常所用的电话，是通过DTMF信号来向交换机传递命令的，我们每按下电话键盘上的一个键，就会同时发出两个不同频率的声音，转化成电流在对面解析。通过某些相关软件手段便可以还原号码按键音，进而解析出号码。”通过软件，刘靖康还原出周鸿祎的电话号码，然后自己拨打后进行了确认。

- 这篇帖子引发了广大网友的好奇心，许多人开始按照这篇帖子的指点自行实验。昨天下午，周鸿祎说：“我的手机号被曝光，现在接到无数个电话，打电话的人就说一句‘周总，认识一下’。”他在微博中说：“手机号被公布，很多好奇的童鞋打电话发短信，峰值时30秒一个电话，一分钟一条短信。感谢大家的关心和问候，恳请大家还是在微博粉我和我交流，有问题咨询发邮件私信均可，请别打电话发短信了，同时邀请南大那位能干的小朋友去360实习。”

- 创新工场的李开复也对这位能干的大学生产生了兴趣。他在微博中说：“这位同学：来创新工场吧！你学的信号处理，不知是否读过我的论文？我有合适项目供你考虑。请私信@创新工场微招聘，我两周后到南京，盼见面。”

- Assuming that we have a time-dependent function f(t) with a period T, that is, f(t+T) = f(t), the Fourier theorem can be cast as a summation

$$f(t) = \sum_{j=-\infty}^{\infty} g_j e^{-ij\omega t},$$

which is commonly known as the Fourier series. Here $\omega = 2\pi/T$ is the fundamental angular frequency and g_j's are the Fourier coefficients, which are given by

$$g_j = \frac{1}{T}\int_0^T f(t)e^{ij\omega t}\,dt.$$

The Fourier theorem can be derived from the properties of the exponential functions

$$\phi_j(t) = \frac{1}{\sqrt{T}} e^{-ij\omega t},$$

which form an orthonormal basis set in the region of one period of f(t):

$$\int_{t_0}^{t_0+T} \phi_j^*(t)\phi_k(t)\,dt = \langle j|k\rangle = \delta_{jk},$$

where t_o is an arbitrary starting point, ϕ^*_j (t) is the complex conjugate of ϕ_j (t), and δ_{jk} is the Kronecker function. We can always take t_o = 0 for convenience, because it is arbitrary.

Fourier analysis and orthogonal functions

- We can generalize the Fourier theorem to a nonperiodic function defined in a region of $x \in [a, b]$ if we have a complete basis set of orthonormal functions $\phi_k(x)$ with

$$\int_a^b \phi_j^*(x)\phi_k(x)\,dx = \langle j|k\rangle = \delta_{jk}$$

- For any arbitrary function f(x) defined in the region of $x \in [a, b]$, it can be written as:

$$f(x) = \sum_j g_j \, \phi_j(x)$$

if the function is square integrable, defined by

$$\int_a^b |f(x)|^2 \, dx < \infty.$$

the coefficients g_j are given by

$$g_j = \int_a^b \phi_j^*(x) f(x) \, dx = \langle j | f \rangle.$$

- The continuous Fourier transform is obtained if we restrict the series to the region of t ∈ [-T/2, T/2] and then extend the period T to infinity. Then the sum becomes an integral:

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(\omega) e^{-i\omega t}\, d\omega,$$

which is commonly known as the Fourier integral. The Fourier coefficient function is given by

$$g(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{i\omega t}\, dt.$$

The Fourier transform & the inverse Fourier transform

Three dimensions:

$$g(q) = \frac{1}{(2\pi)^{3/2}} \int f(r) e^{-iq\cdot r} dr$$

$$f(r) = \frac{1}{(2\pi)^{3/2}} \int g(q) e^{iq\cdot r} dq$$

The space defined by q is usually called the momentum space.

Discrete Fourier transform

- $f(x)$: a physical quantity obtained from measurements. The measurements are between $x=0$ and $x=L$.

- The data are measured at evenly spaced points with each interval $h=L/(N-1)$, where N is the total number of data points.

- Assume that the data repeat periodically outside the region of $x \in [0,L]$, which is equivalent to the periodic boundary condition on the finite system.

- The corresponding wavenumber in the momentum space is then discrete too, with an interval $\kappa=2\pi/L$.

Discrete Fourier transform

$$f_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} g_j e^{i2\pi jk/N}, \quad g_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} f_k e^{-i2\pi jk/N},$$

The exponential functions in the series form a discrete basis set of orthogonal functions:

$$\langle \phi_j | \phi_m \rangle = \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} e^{-i2\pi kj/N} \frac{1}{\sqrt{N}} e^{i2\pi km/N}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} e^{i2\pi k(m-j)/N} = \delta_{jm}.$$

Code example

$f_1(x)=\sin(2\pi x)+\sin(50\pi x+1)+\sin(104\pi x+2)$

$f_2(x)=-\sin(2\pi x)+\sin(50\pi x+2)+\sin(104\pi x+1)$

x within $[0,1]$.

6.1.Fourier.cpp

Homework

- To do a two-dimensional discrete Fourier transform:
- $f(x,y)= \sin(5x+7y)$ for x within $0,2\pi$, y within $0, 2\pi$

Fast Fourier transform

- The straightforward discrete Fourier transform algorithm is very inefficient, because the computing time needed is proportional to N^2.

- Speed: FFT (fast Fourier transform).
- by Cooley and Tukey (1965)
- Originally proposed by Gauss, 1822;

- Carl Friedrich Gauss

  1777-1855

- The simplest fast Fourier transform algorithm is accomplished with the observation that we can separate the odd and even terms in the discrete Fourier transform as

$$g_j = \sum_{k=0}^{N/2-1} f_{2k} e^{-i2\pi j(2k)/N} + \sum_{k=0}^{N/2-1} f_{2k+1} e^{-i2\pi j(2k+1)/N},$$

$$= x_j + y_j e^{-i2\pi j/N},$$

$$x_j = \sum_{k=0}^{N/2-1} f_{2k} e^{-i2\pi jk/(N/2)} \qquad y_j = \sum_{k=0}^{N/2-1} f_{2k+1} e^{-i2\pi jk/(N/2)}.$$

Here the factor 1/sqrt(N) have been ignored.

- This process can be repeated further and further until eventually we have only two terms in each summation if $N = 2^M$, where M is an integer.

- There is one more symmetry between g_j for $j < N/2$ and g_j for $j \geq N/2$:

$$g_j = x_j + w^j y_j,$$

$$g_{j+N/2} = x_j - w^j y_j$$

- where $w = e^{-i2\pi/N}$ and $j = 0, 1, \dots, N/2-1$.

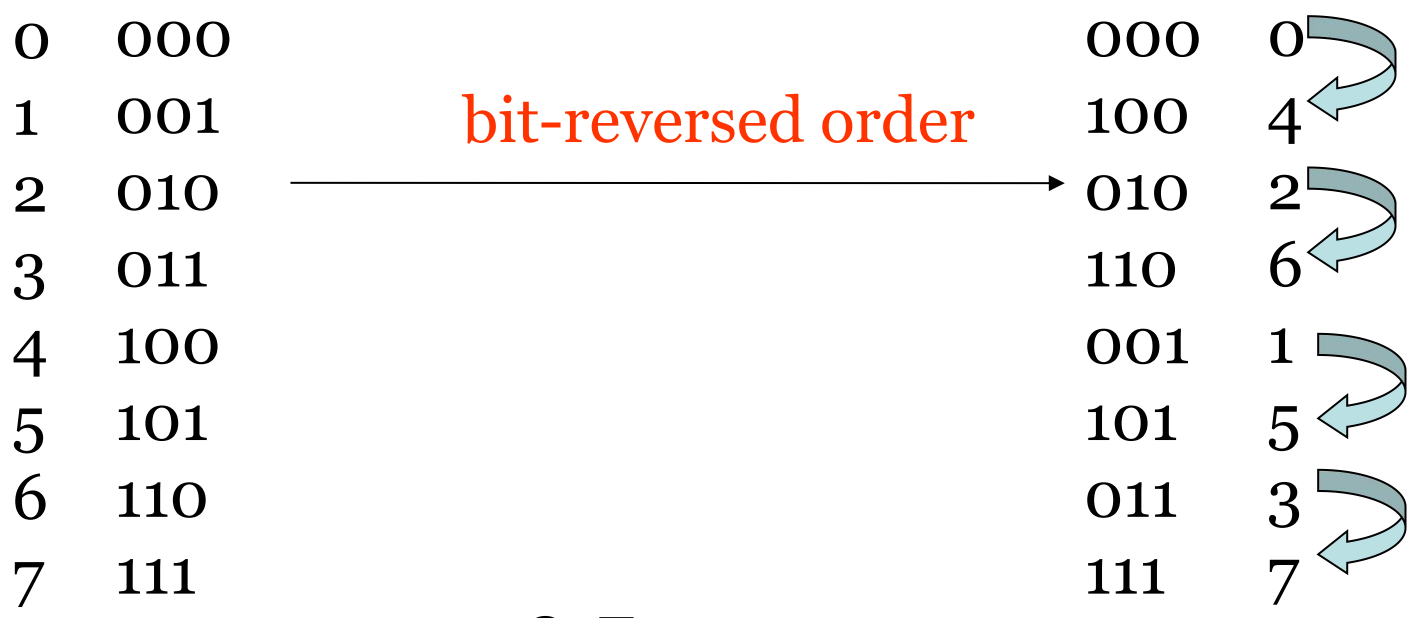
- After the summation is decomposed M times, we need to add individual data points in pairs.

- However, due to the sorting of the odd and even terms in every level of decomposition, the points in each pair at the first level of additions can be very far apart in the original data string.

- For example, if N=8, the decomposition is like:

```
              0-7
       0 2 4 6       1 3 5 7
      0:4   2:6      1:5   3:7
```

- Cooley and Tukey (1965) found that if we record the data string index with a binary number, a bit-reversed order will put each pair of data points next to each other for the summations at the first level.



0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

000	0
100	4
010	2
110	6
001	1
101	5
011	3
111	7

0-7

0 2 4 6   1 3 5 7

0:4   2:6   1:5   3:7

$g_j\ (0, 1, 2, 3, 4, 5, 6, 7) = x_j\ (0, 2, 4, 6) + w_j\ \ y_j\ (1, 3, 5, 7)$

$\boxed{1}$

$w_j = \exp(-i2\pi j/8) \qquad (j=0\text{-}3)$

$g_0=x_0+w_0y_0 \quad g_1=x_1+w_1y_1 \quad g_2=x_2+w_2y_2 \quad g_3=x_3+w_3y_3$

$g_4=x_0-w_0y_0 \quad g_5=x_1-w_1y_1 \quad g_6=x_2-w_2y_2 \quad g_7=x_3-w_3y_3$

$\boxed{2}$

$x_j\ (0, 2, 4, 6) = xx_j\ (0, 4) + ww_j\ \ xy_j\ (2, 6)$

$y_j\ (1, 3, 5, 7) = yx_j\ (1, 5) + ww_j\ \ yy_j\ (3, 7)$

$ww_j = \exp(-i2\pi j/4) \qquad (j=0\text{-}1)$

$x_0=xx_0+ww_0xy_0 \quad x_1=xx_1+ww_1xy_1 \quad y_0=yx_0+ww_0yy_0 \quad y_1=yx_1+ww_1yy_1$

$x_2=xx_0-ww_0xy_0 \quad x_3=xx_1-ww_1xy_1 \quad y_2=yx_0-ww_0yy_0 \quad y_3=yx_1-ww_1yy_1$

$\boxed{3}$

$xx_j(0,4)=xxx_j(0)+www_j\ xxy_j(4) \quad xy_j(2,6)=xyx_j(2)+www_j\ xyy_j(6)$

$yx_j(1,5)=yxx_j(1)+www_j\ yxy_j(5) \quad yy_j(3,7)=yyx_j(3)+www_j\ yyy_j(7)$

$www_j = \exp(-i2\pi j/2) \qquad (j=0) \qquad www_0=1$

$xx_0=f_0+f_4 \quad xy_0=f_2+f_6 \quad yx_0=f_1+f_5 \quad yy_0=f_3+f_7$

$xx_1=f_0-f_4 \quad xy_1=f_2-f_6 \quad yx_1=f_1-f_5 \quad yy_1=f_3-f_7$

From bottom to top

$$\begin{tikzcd}
f_0 \arrow[d, red] \arrow[dr, blue] & f_4 \arrow[dl, red] \arrow[d, blue] & f_2 \arrow[d, red] \arrow[dr, blue] & f_6 \arrow[dl, red] \arrow[d, blue] & f_1 \arrow[d, red] \arrow[dr, blue] & f_5 \arrow[dl, red] \arrow[d, blue] & f_3 \arrow[d, red] \arrow[dr, blue] & f_7 \arrow[dl, red] \arrow[d, blue] \\
xx_0 \arrow[d, red] \arrow[drr, blue] & xx_1 \arrow[d, red] \arrow[drr, blue] & xy_0 \arrow[dll, red] \arrow[d, blue] & xy_1 \arrow[dll, red] \arrow[d, blue] & yx_0 \arrow[d, red] \arrow[drr, blue] & yx_1 \arrow[d, red] \arrow[drr, blue] & yy_0 \arrow[dll, red] \arrow[d, blue] & yy_1 \arrow[dll, red] \arrow[d, blue] \\
x_0 \arrow[d, red] \arrow[drrrr, blue] & x_1 \arrow[d, red] \arrow[drrrr, blue] & x_2 \arrow[d, red] \arrow[drrrr, blue] & x_3 \arrow[d, red] \arrow[drrrr, blue] & y_0 \arrow[dllll, red] \arrow[d, blue] & y_1 \arrow[dllll, red] \arrow[d, blue] & y_2 \arrow[dllll, red] \arrow[d, blue] & y_3 \arrow[dllll, red] \arrow[d, blue] \\
g_0 & g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7
\end{tikzcd}$$

翻花绳

Code example

$f_1(x)=\sin(2\pi x)+\sin(50\pi x+1)+\sin(104\pi x+2)$ within [0,1].

6.2.FFT.cpp

With the FFT algorithm, the computing time is reduced.

A careful analysis shows that the total computing time is proportional to $N \log_2 N$ instead of to N^2.

FFTW

- The Fastest Fourier Transform in the West (FFTW) is a software library for computing discrete Fourier transforms developed by Matteo Frigo and Steven G. Johnson at the Massachusetts Institute of Technology.
- Website: http://www.fftw.org
- Stable version: 3.3.8
- License: GPL, commercial

Homework

- To learn how to use the FFTW library
- 6.3.FFTW.cpp

- Try $f(x,y)= \sin(5x+7y)$ for x within $0,2\pi$, y within $0, 2\pi$