Partial Differential Equations

Yubo Tao

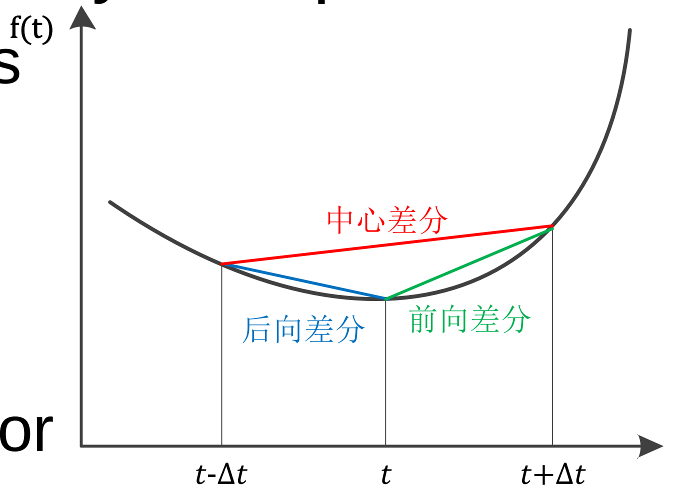
May 28, 2019

Outline

- Overview
- Ordinary differential equations (ODE)
  - Initial value problem (初值问题)
    - Euler, Runge-Kutta, Adams methods
  - Error analysis (误差分析)
    - Discretization error, Round-off, Stability
  - Boundary value problem (边值问题)
    - Shooting method, direct solve, relaxation
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

Ordinary Differential Equations

- Numerical methods – Based on Taylor expansions
  - $O(\Delta t^2)$ center-difference schemes
  - $f'(t) = \dfrac{[f(t+\Delta t) - f(t-\Delta t)]}{2\Delta t}$
  - $f''(t) = \dfrac{[f(t+\Delta t) - 2f(t) + f(t-\Delta t)]}{\Delta t^2}$
  - Accuracy(正确性) - truncated Taylor
  - Stability(稳定性) - difference schemes



$$f'(t) = [f(t+\Delta t) - f(t-\Delta t)]/2\Delta t$$
$$f''(t) = [f(t+\Delta t) - 2f(t) + f(t-\Delta t)]/\Delta t^2$$
$$f'''(t) = [f(t+2\Delta t) - 2f(t+\Delta t) + 2f(t-\Delta t) - f(t-2\Delta t)]/2\Delta t^3$$
$$f''''(t) = [f(t+2\Delta t) - 4f(t+\Delta t) + 6f(t) - 4f(t-\Delta t) + f(t-2\Delta t)]/\Delta t^4$$

Ordinary Differential Equations

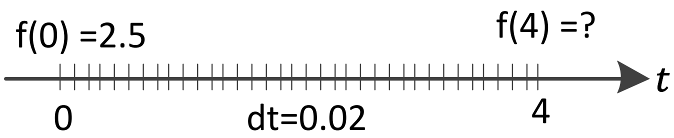
- Euler Method (forward or backward)
- 4th order Runge-Kutta Method
- Initial value problems

$$\frac{d\boldsymbol{y}}{dt} = f(\boldsymbol{y}, t)$$

$$y_{n+1} = y_n + \Delta t \cdot f(\boldsymbol{y}, t)$$

```
T = 4;
N = 200;
dt = T / N;
t0 = 0;
y0 = 2.5;
y(1) = y0;

for i = 1 : N
   t  = t0 + dt * i;
   y1 = y0 + f(y, t) * dt;
   y0 = y1;
   y(i + 1) = y1;
end
```



Outline

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
    - Basic time and space stepping schemes
    - Examples
      - Poisson equations
      - Heat diffusion equations
      - Wave equations
      - Advection-diffusion equation
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

Partial Differential Equations

偏微分方程类型	椭圆方程 **Elliptic equation**	抛物线方程 **Parabolic equation**	双曲线方程 **Hyperbolic equation**
典型方程 $u = u(x, y, t)$	$\Delta u = f$ 泊松方程	$u_t - k\Delta u = f$ 热传导方程	$u_{tt} - c^2\Delta u = f$ 波动方程
初值/边值问题	边值问题	初值问题	初值问题

- Numerical methods
  - 1 PDE → ODE (空间域展开)
    - Finite difference methods (有限差分法)
    - Spectral methods (频谱法)
  - 2 Algebra
    - $Ax = \mathrm{b}$
    - Iterative scheme $\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \boldsymbol{A} \cdot \mathrm{dt}$ (methods of lines)

$$\begin{tikzcd} \text{PDE} \arrow[r, "1"] & \text{ODE} \arrow[r, "2"] & \text{Algebra*} \arrow[ll, bend left, "3"] \end{tikzcd}$$

Undo

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$$

$$\Delta = \nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Finite Difference Methods

- Basic steps
  - Discretize in space and time
  - Solve a large linear system of equations or manipulate large, sparse matrices
- Accuracy and stability
  - Accuracy from the space-time discretization
  - Numerical stability as the solution is propagated in time
- Advantages
  - Easy to implement
  - Handle fairly complicated boundary conditions
  - Explicit calculations of the computational error

Finite Difference Methods

- Typical PDEs

典型PDE求解	$\Delta u = f$ 泊松方程	$u_t = k\Delta u$ 热传导方程	$u_t = \boldsymbol{c}\boldsymbol{u}_x$ 波动方程
空间离散差分近似 $\delta = \Delta x = \Delta y$	$Ax = \mathrm{b}$	$\frac{d\boldsymbol{u}}{\mathrm{dt}} = \frac{k}{\delta^2}\boldsymbol{A}\boldsymbol{u}$	$\frac{d\boldsymbol{u}}{\mathrm{dt}} = \frac{c}{2\delta}\boldsymbol{A}\boldsymbol{u}$
求解	(Linear Algebra) LU decomposition Conjugate gradient	(ODE) Euler method Runge-Kutta	(ODE) Euler method Runge-Kutta

$$-4u_{mn} + u_{(m-1)n} + u_{(m+1)n} + u_{m(n-1)} + u_{m(n+1)} = \delta^2 f$$

$$\frac{\partial u}{\partial t} = \frac{k}{\delta^2}[u(x+\Delta x) - 2u(x,y) + u(x-\Delta x, y)$$

$$\frac{\partial u}{\partial t} = \frac{k}{\delta^2}[-4u(x,y) + u(x-\Delta x, y) + u(x+\Delta x, y) + u(x, y-\Delta y) + u(x, y-\Delta y)]$$

$$\frac{\partial u}{\partial t} = \frac{k}{2\delta}[u(x+\Delta x) - u(x-\Delta x, y)$$

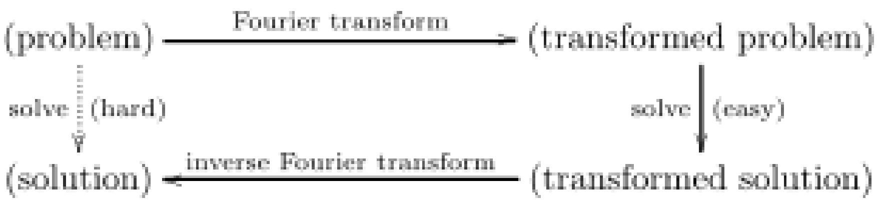
Finite Difference Methods

典型PDE求解	$\Delta u = f$ 泊松方程	$u_t = k\Delta u$ 热传导方程	$u_t = \boldsymbol{c u_x}$ 波动方程
空间离散差分近似 $\delta = \Delta x = \Delta y$	$Ax = \mathrm{b}$	$\frac{d\boldsymbol{u}}{\mathrm{dt}} = \frac{k}{\delta^2}\boldsymbol{Au}$	$\frac{d\boldsymbol{u}}{\mathrm{dt}} = \frac{c}{2\delta}\boldsymbol{Au}$
求解	(Linear Algebra) LU decomposition Conjugate gradient	(ODE) Euler method Runge-Kutta	(ODE) Euler method Runge-Kutta

$$
\mathbf{A} = \begin{bmatrix}
-4 & 1 & 0 & 1 & 1 \\
1 & -4 & 1 & 0 & 0 \\
0 & 1 & -4 & 1 & 0 \\
1 & 0 & 1 & -4 & 0 \\
1 & 0 & 0 & 0 & -4 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
-2 & 1 & 0 & \cdots & 0 & 1 \\
1 & -2 & 1 & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \ddots & & \\
\vdots & & & & & \vdots \\
 & & & & & 0 \\
\vdots & \cdots & 0 & 1 & -2 & 1 \\
1 & 0 & \cdots & 0 & 1 & -2
\end{bmatrix}
\begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & -1 \\
-1 & 0 & 1 & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \ddots & & \\
\vdots & & & & & \vdots \\
 & & & & & 0 \\
\vdots & \cdots & 0 & -1 & 0 & 1 \\
1 & 0 & \cdots & 0 & -1 & 0
\end{bmatrix}
$$

Outline

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
    - Fourier transform
    - Examples
      - Poisson equations
      - Heat diffusion equations
      - Filtered pseudo-spectral
    - Comparison of spectral and finite difference methods
  - Finite elements (有限元法)



J. Nathan Kutz, Data-Driven Modeling & Scientific Computation, Methods for Complex Systems & Big Data, Oxford, 2013

Spectral Methods

- The best known example - Fourier transform
- General expansion basis (functions)
  - Cosines and sines
    - $O(N^2) \rightarrow O(NlogN)$, periodic, pinned, no-flux
  - Bessel functions: radial, 2D problem
  - Legendre polynomials: 3D Laplaces equation
  - Hermite-Gauss polynomials: Schrödinger with harmonic potential
  - Spherical harmonics: radial, 3D problem
  - Chbychev polynomials: bounded 1D domains
    - Related to discrete cosines transform
- Application: Phase-based video motion processing
  - http://people.csail.mit.edu/nwadhwa/phase-video/

Basics of Fourier Series

- From Taylor series to a trigonometric series of sines and cosines
  - A uniformly convergent series

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx) \quad x \in (-\pi, \pi]$$

$$f(x) \cos mx = \frac{a_0}{2} \cos mx + \sum_{n=1}^{\infty} (a_n \cos nx \cos mx + b_n \sin nx \cos mx)$$

$$\int_{-\pi}^{\pi} f(x) \cos mx dx = \frac{a_0}{2} \int_{-\pi}^{\pi} \cos mx dx$$

$$+ \sum_{n=1}^{\infty} \left(a_n \int_{-\pi}^{\pi} \cos nx \cos mx dx + b_n \int_{-\pi}^{\pi} \sin nx \cos mx dx \right)$$

Basics of Fourier Series

- Orthogonality properties of sine and cosine

$$\int_{-\pi}^{\pi} \sin nx \cos mx dx = 0 \quad \forall\ n, m$$

$$\int_{-\pi}^{\pi} \cos nx \cos mx dx = \begin{cases} 0 & n \neq m \\ \pi & n = m \end{cases}$$

$$\int_{-\pi}^{\pi} \sin nx \sin mx dx = \begin{cases} 0 & n \neq m \\ \pi & n = m \end{cases}$$

- Coefficients a_n and b_n

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx dx \quad n \geq 0$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx dx \quad n > 0$$

Basics of Fourier Series

- Fourier series on the domain $x \in [-L, L]$
  - c_n is a complex, but $f(x)$ is still real

$$f(x) = \sum_{-\infty}^{\infty} c_n e^{in\pi x/L} \quad x \in [-L, L]$$

$$c_n = \frac{1}{2L} \int_{-L}^{L} f(x) e^{-in\pi x/L} dx$$

1. j
2. i = -j
3. i = j

- Properties
  - c_0 is real and $c_{-n} = c_n^*$
  - If $f(x)$ is even, all c_n are real
  - If $f(x)$ is odd, $c_0 = 0$ and all c_n are purely imaginary

Fourier Transform

- Fourier transform

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

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

1. j
2. i = -j
3. i = j

  - Entire real line $x \in [-\infty, \infty]$, but computational domain $x \in [-L, L]$
  - Kernel of the transform, $\exp(\mp ikx)$, describes oscillatory behavior

Fourier Transform

- Fourier transform
  - An eigenfunction expansion over all continuous wavenumbers k
- On a finite domain $x \in [-L, L]$
  - A discrete sum of eigenfunctions and associated wavenumbers (eigenvalues)

Fourier Transform

- From continuous to discrete on a finite domain
  - An expansion in a basis of cosine and sine functions
  - If $f(x)$ is discretized in N, F(k) has N data points

$$F[k] = \sum_{n=0}^{N-1} f[\mathrm{n}] e^{-j\frac{2\pi}{N}nk} \qquad k = 0,1,\dots,N-1$$

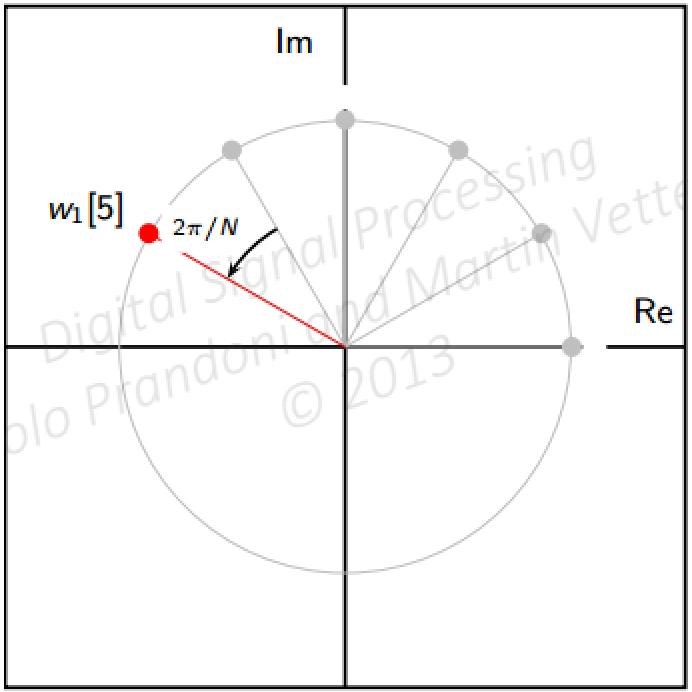
$$f[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}nk} \qquad n = 0,1,\dots,N-1$$

$$w^{nk} = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin\left(\frac{2\pi}{N}nk\right)$$

$$F_n = \sum_{k=0}^{N-1} w^{nk} f_k \qquad n = 0,1,\dots,N-1$$

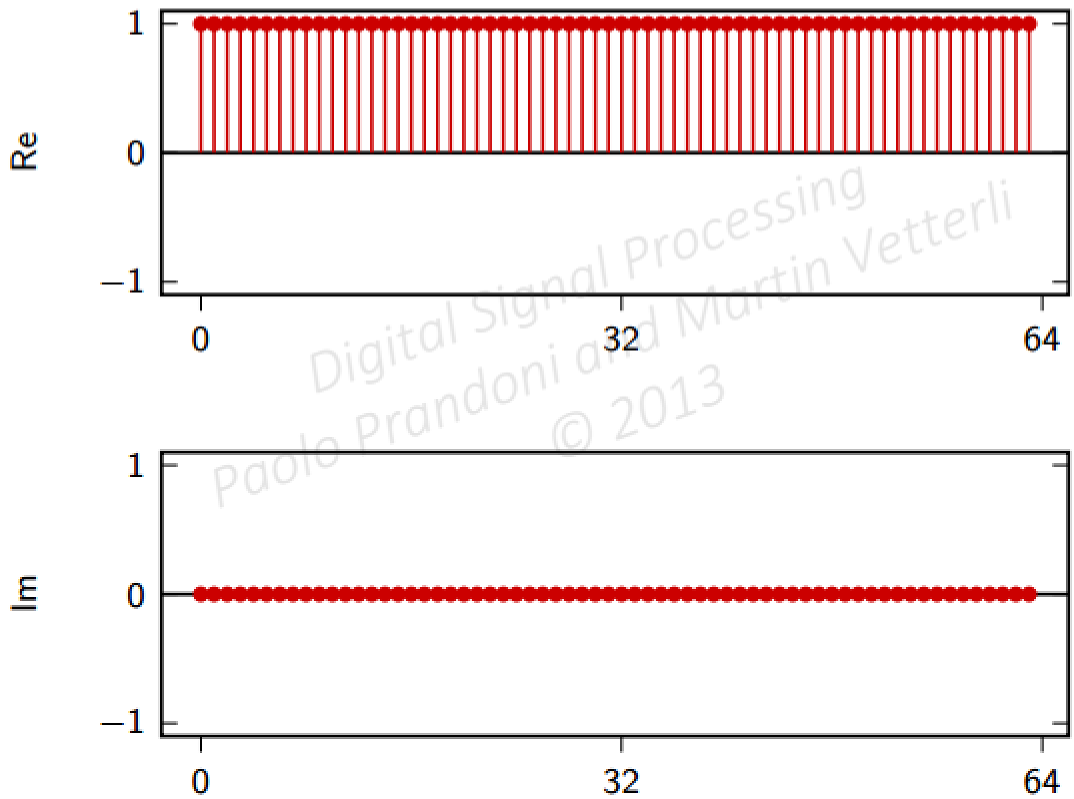
Fourier Transform

- Fourier analysis is a simple change of basis
- The set in C^N, $w_k[n] = e^{j\frac{2\pi}{N}nk}$, $\mathrm{n,k} = 0,1,\dots,\mathrm{N}-1$ is an orthogonal basis in C^N



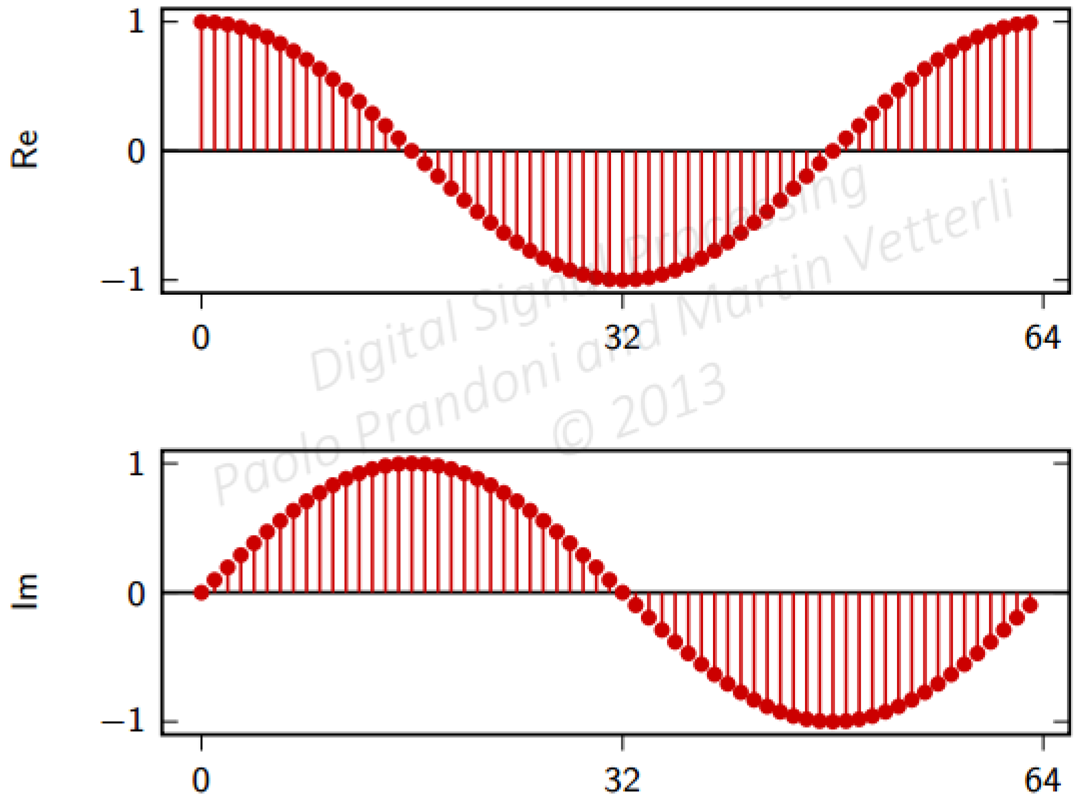
Fourier Transform

- $w_k[n] = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin(\frac{2\pi}{N}nk)$, basis vector $w_0 \in C^{64}$



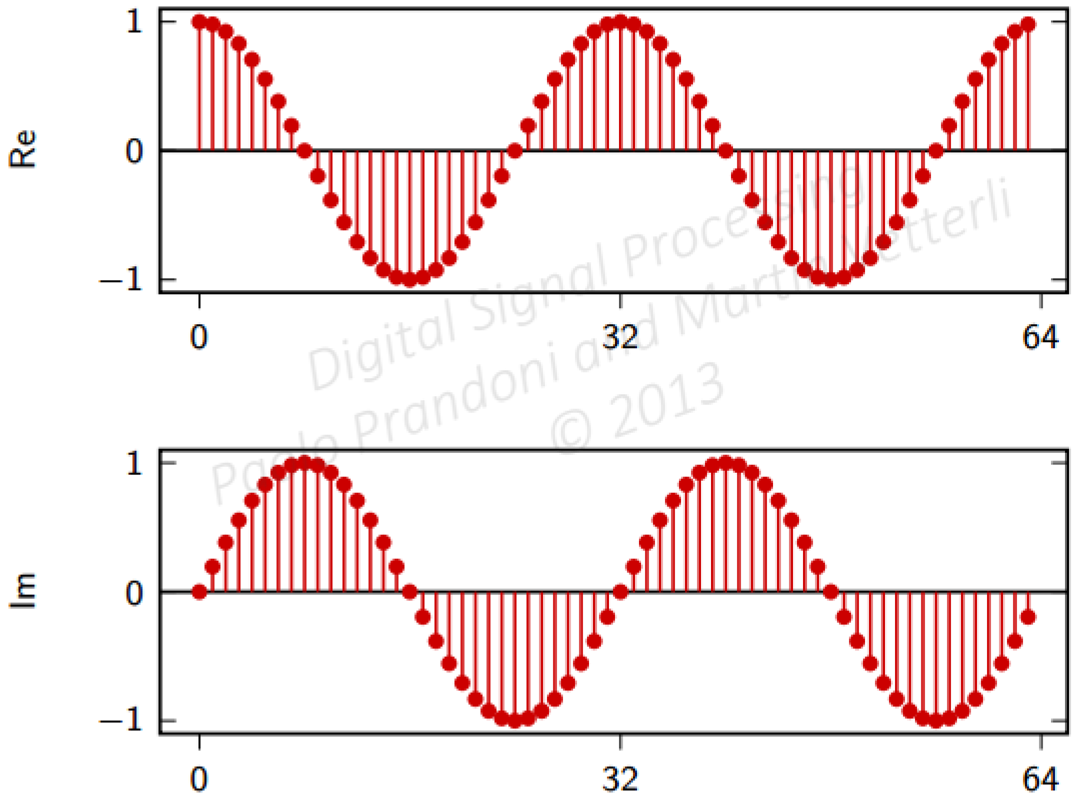
Fourier Transform

- $w_k[n] = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin(\frac{2\pi}{N}nk)$, basis vector $w_1 \in C^{64}$



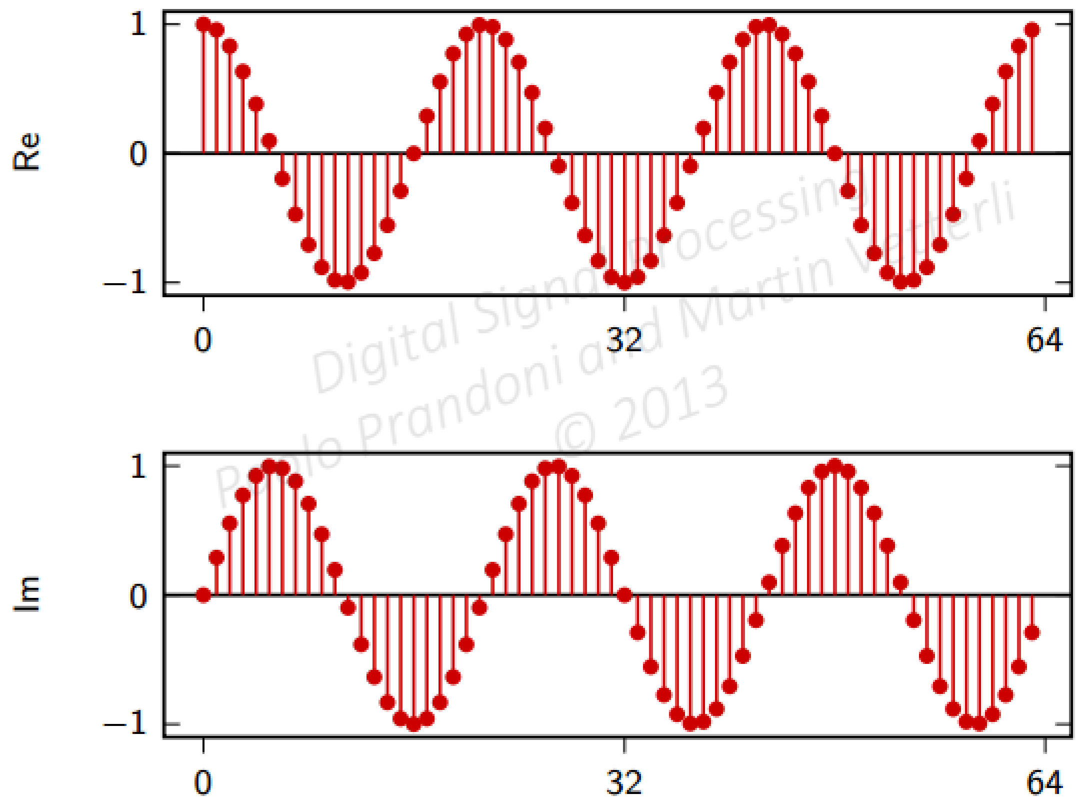
Fourier Transform

- $w_k[n] = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin(\frac{2\pi}{N}nk)$, basis vector w_2



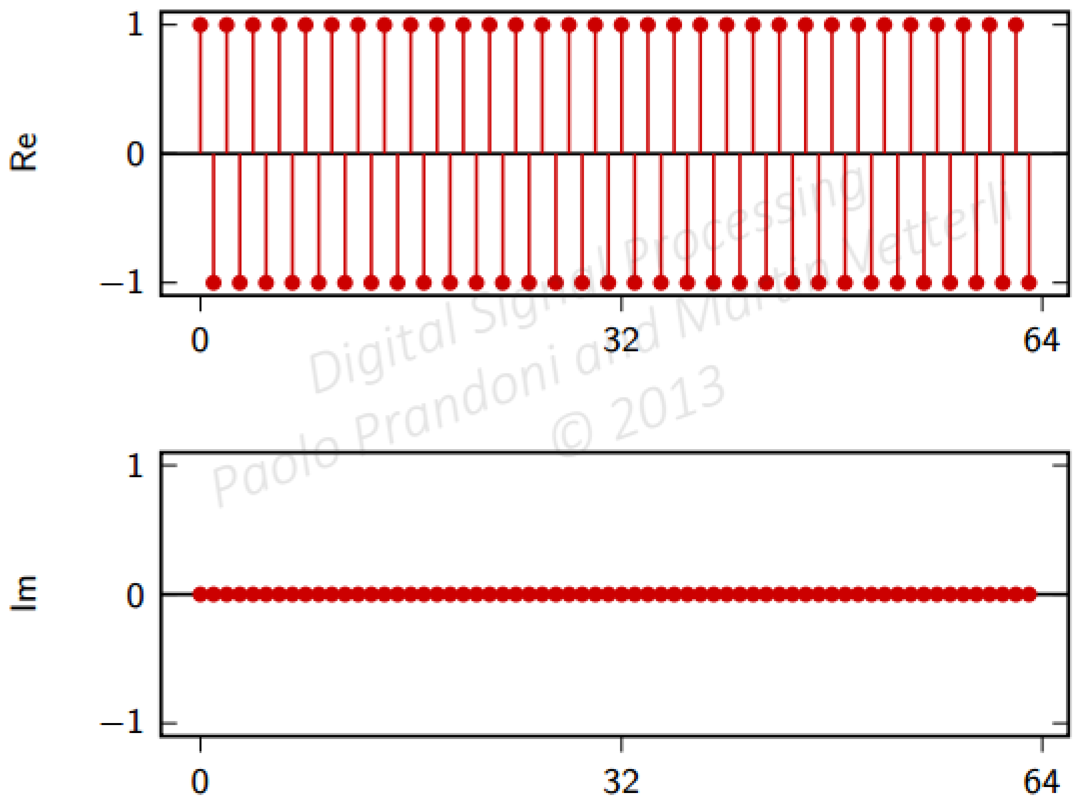
Fourier Transform

- $w_k[n] = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin(\frac{2\pi}{N}nk)$, basis vector w_3



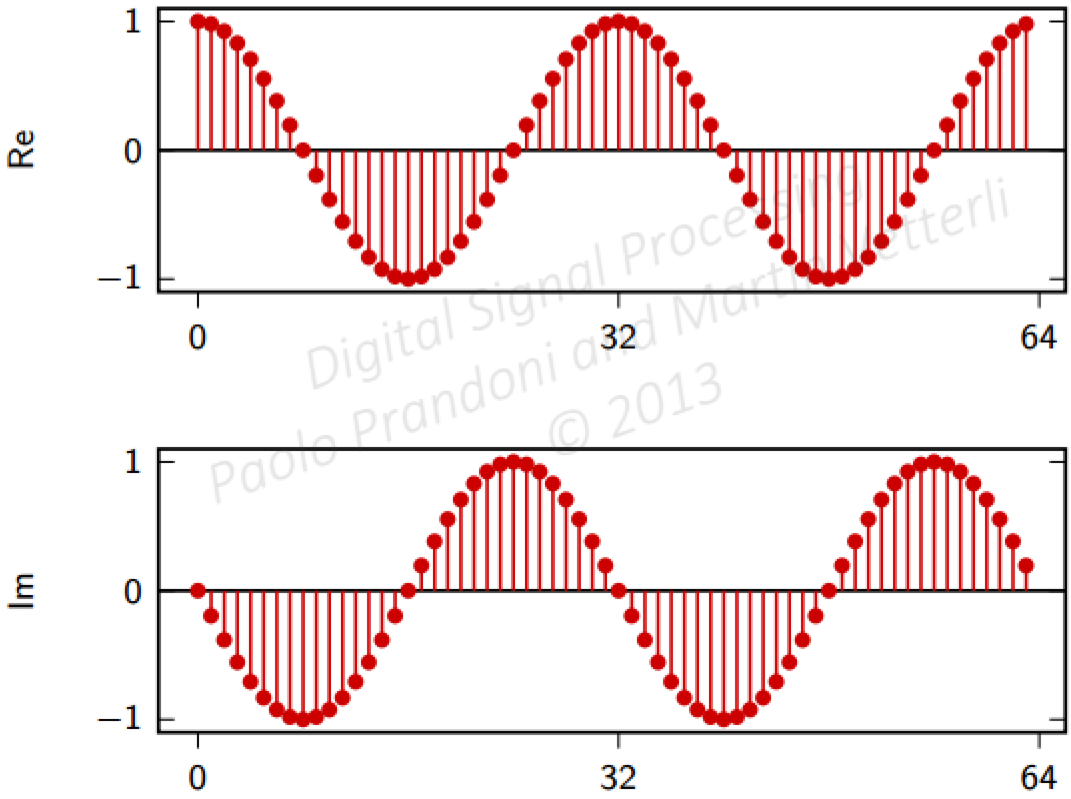
Fourier Transform

- $w_k[n] = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin(\frac{2\pi}{N}nk)$, basis vector w_{32}



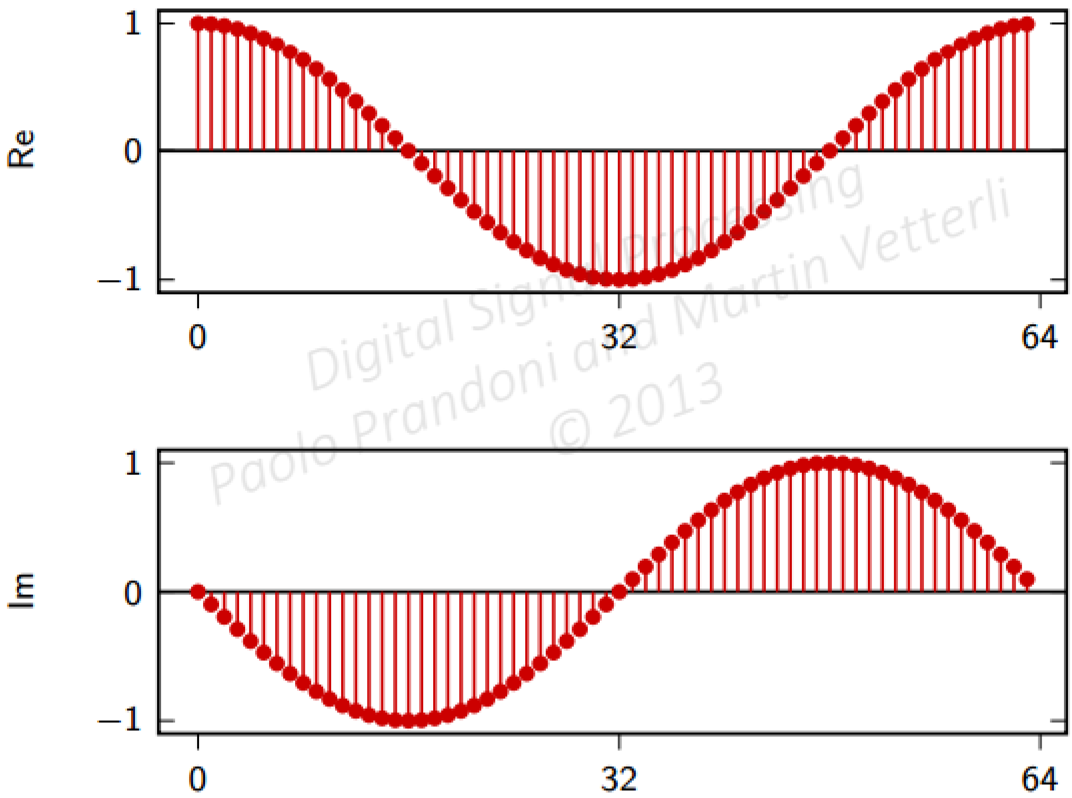
Fourier Transform

- $w_k[n] = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin(\frac{2\pi}{N}nk)$, basis vector w_{62}



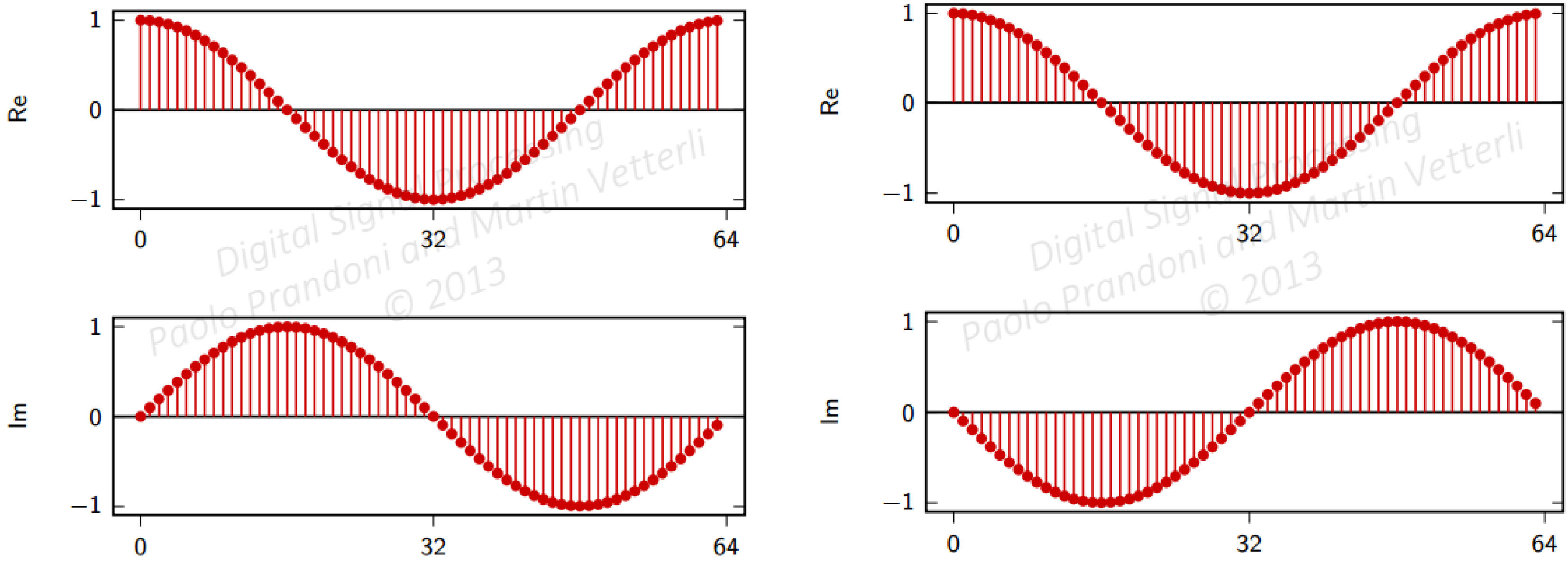
Fourier Transform

- $w_k[n] = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin(\frac{2\pi}{N}nk)$, basis vector w_{63}



Fourier Transform

- $w_k[n] = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin(\frac{2\pi}{N}nk)$
- basis vectors w_1 and w_{63}



Fourier Transform

- $< w_k, w_h > = \sum_{n=0}^{N-1} (e^{j\frac{2\pi}{N}nk})^* e^{j\frac{2\pi}{N}nh}$

$$= \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}n(h-k)}$$

$$= \begin{cases} N \quad for\ h = k \\ \frac{1-e^{j2\pi(h-k)}}{1-e^{j\frac{2\pi}{N}(h-k)}} = 0 \quad otherwise \end{cases}$$

- Vectors are not orthonormal. Normalization factor would be $1/\sqrt{N}$

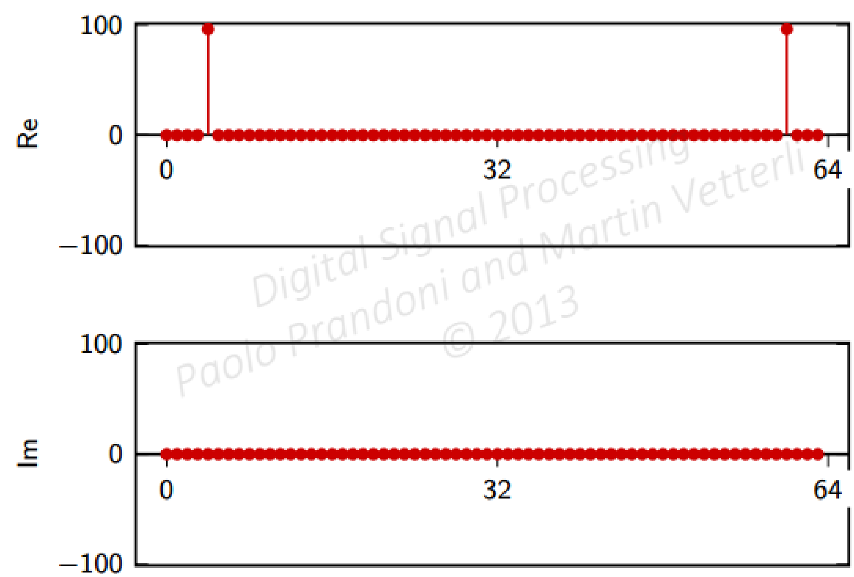
Fourier Transform

- $w_k[n] = e^{j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) + j\sin\left(\frac{2\pi}{N}nk\right)$
- DFT of $x[n] = 3\cos\left(\frac{2\pi}{16}n\right), x[n] \in C^{64}$
- $x[n] = 3\cos\left(\frac{2\pi}{16}n\right)$

$$= 3\cos\left(\frac{2\pi}{64}4n\right)$$

$$= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{-j\frac{2\pi}{64}4n}\right]$$

$$= \frac{3}{2}\left[w_4[n] + w_{60}[n]\right]$$

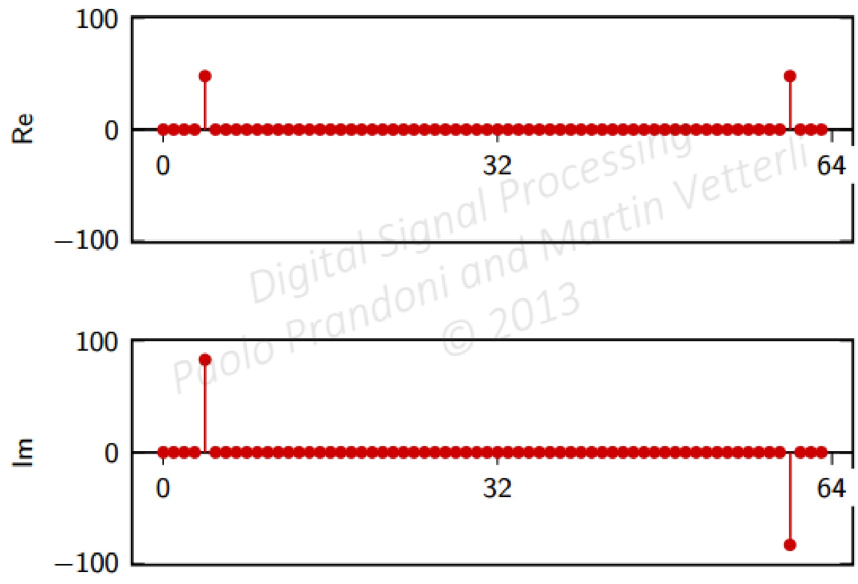


- $X[k] = \langle w_k[n], x[n] \rangle = \begin{cases} 96 & for\ k = 4, 60 \\ 0 & otherwise \end{cases}$

Fourier Transform

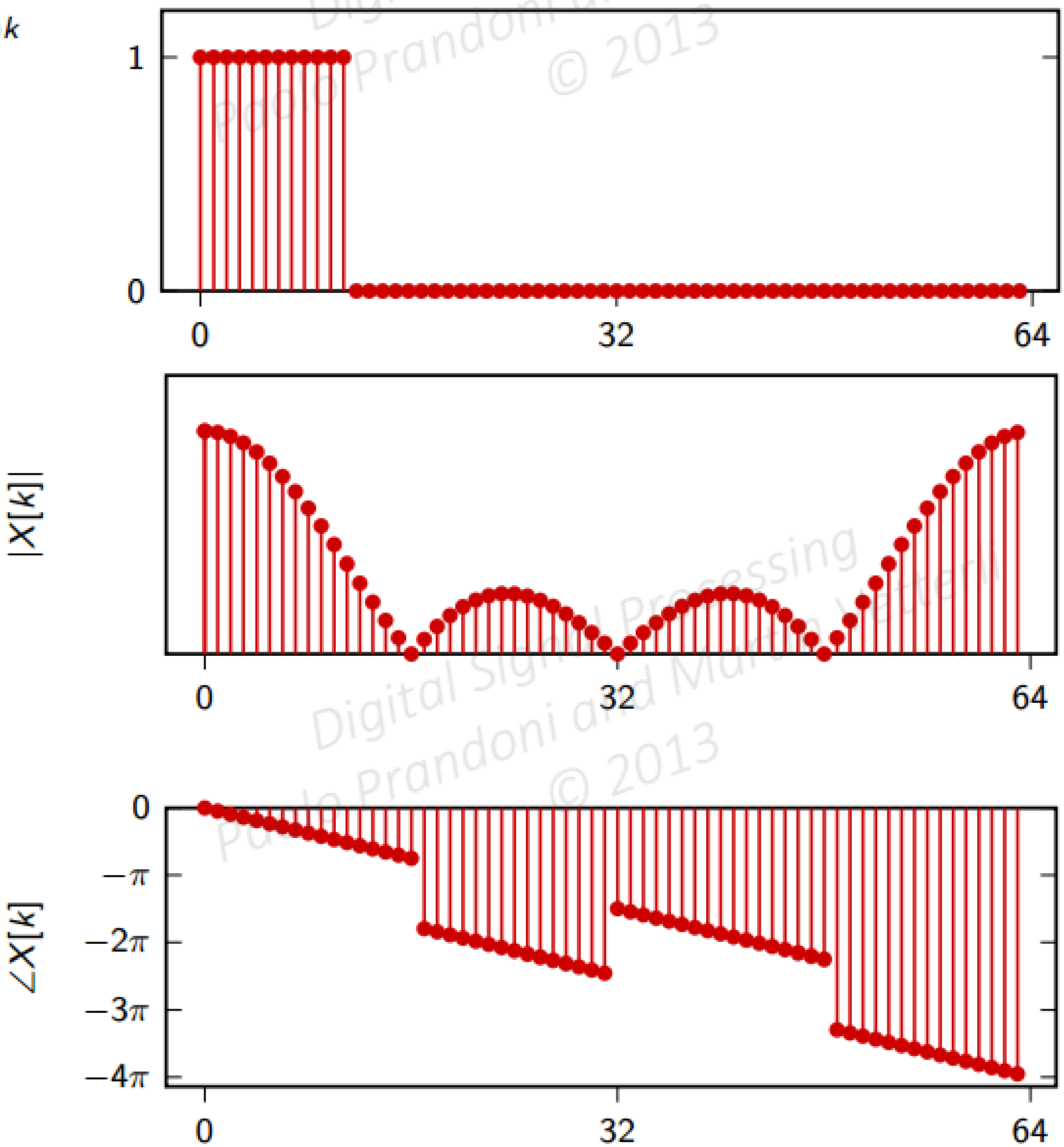
- DFT of $x[n] = 3\cos\left(\frac{2\pi}{16}n + \frac{\pi}{3}\right), x[n] \in C^{64}$

- $X[k] = < w_k[n], x[n] > = \begin{cases} 96e^{j\frac{\pi}{3}} & for\ k = 4 \\ 96e^{-j\frac{\pi}{3}} & for\ k = 60 \\ 0 & otherwise \end{cases}$



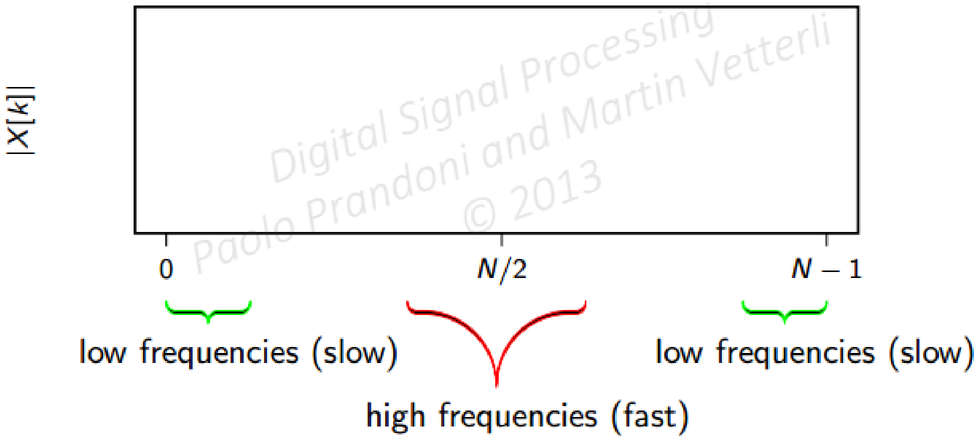
Fourier Transform

- $X[k] = \frac{\sin\left(\frac{\pi}{N}Mk\right)}{\sin\left(\frac{\pi}{N}k\right)} e^{-j\frac{\pi}{N}(M-1)k}$

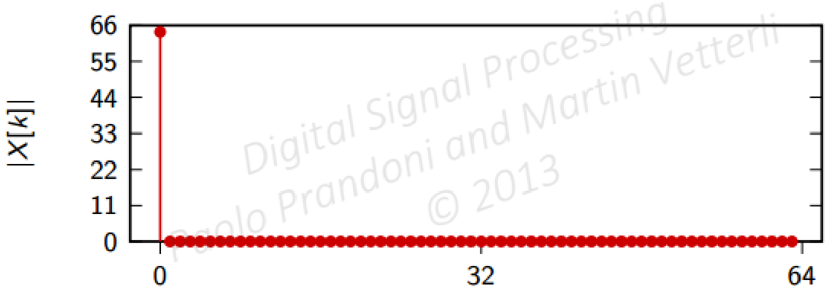


Fourier Transform

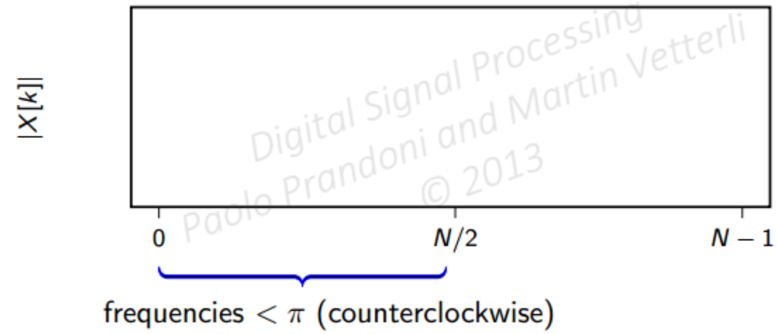
- Interpreting a DFT plot

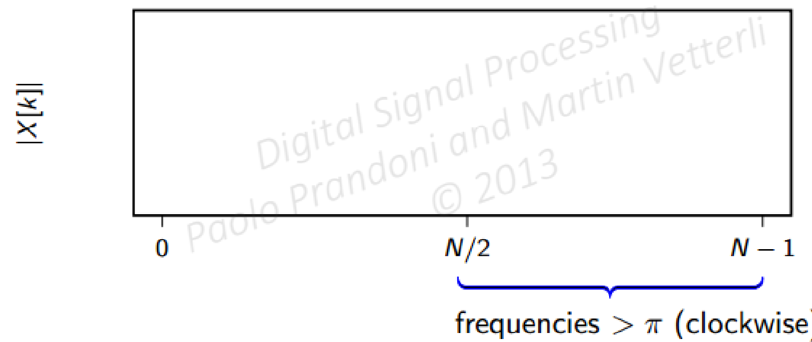


$x[n] = 1$ (slowest signal)

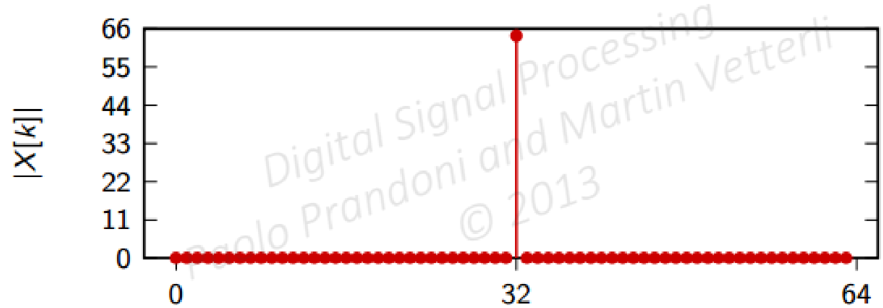


only lowest frequency





$x[n] = \cos \pi n = (-1)^n$ (fastest signal)

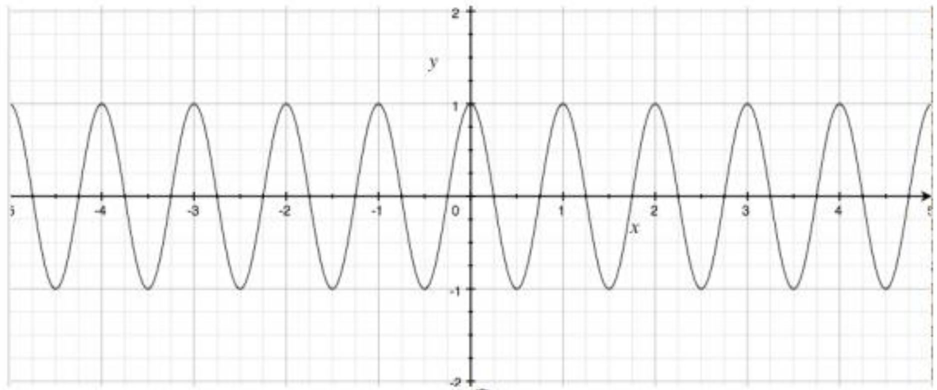


only highest frequency

Fourier Transform

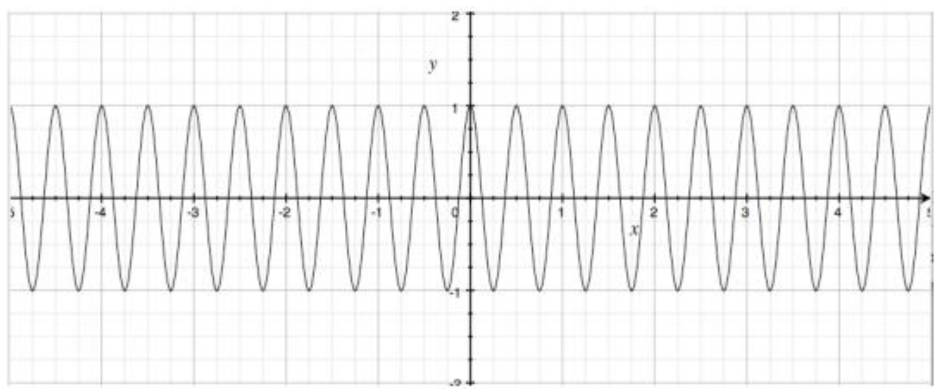
- $cos2\pi fx$

$$f = \frac{1}{T}$$



$$f = 1$$

$$\cos 2\pi x$$

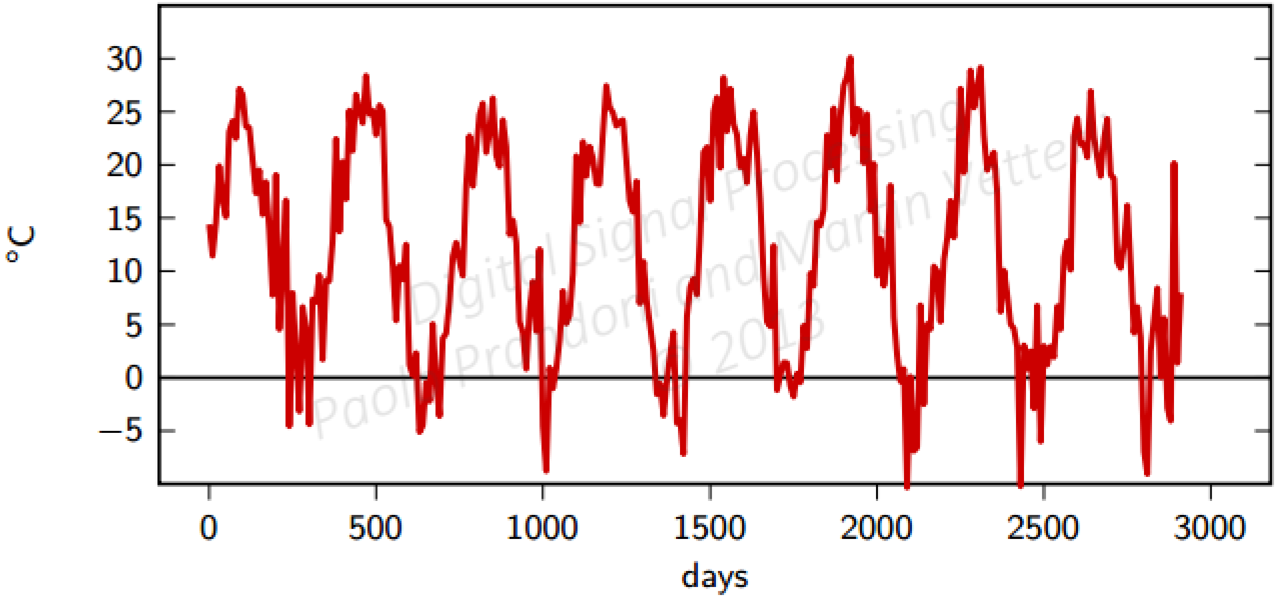


$$f = 2$$

$$\cos 4\pi x$$

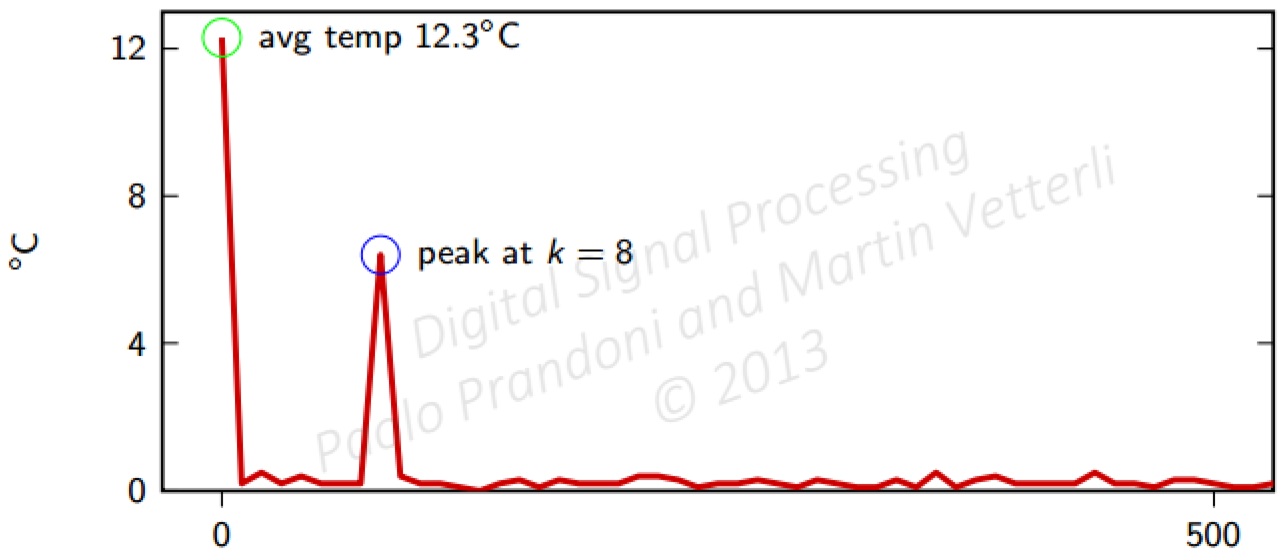
Fourier Transform

- Daily temperature (2920 days)



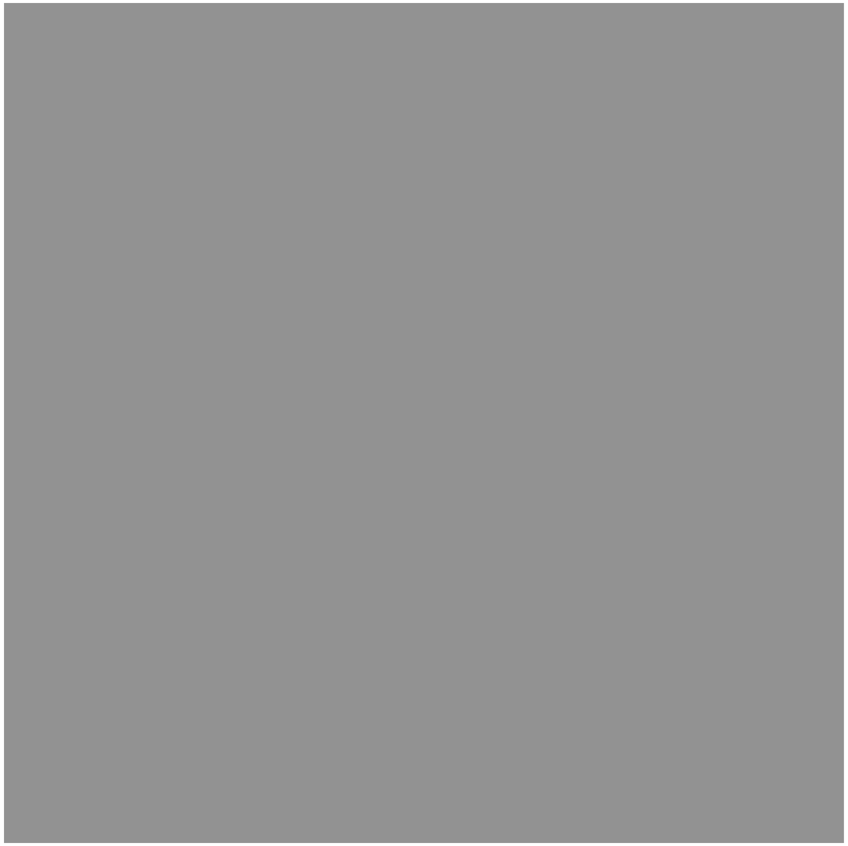
Fourier Transform

- Daily temperature (2920 days)
  - Average value (0-th DFT coefficient): 12.3
  - DFT main peak for k = 8, value 6.4
  - 8 cycles over 2920 days, period: 2920 /8 = 365 days



first few hundred DFT coefficients
(in magnitude and normalized by the length of the temperature vector)

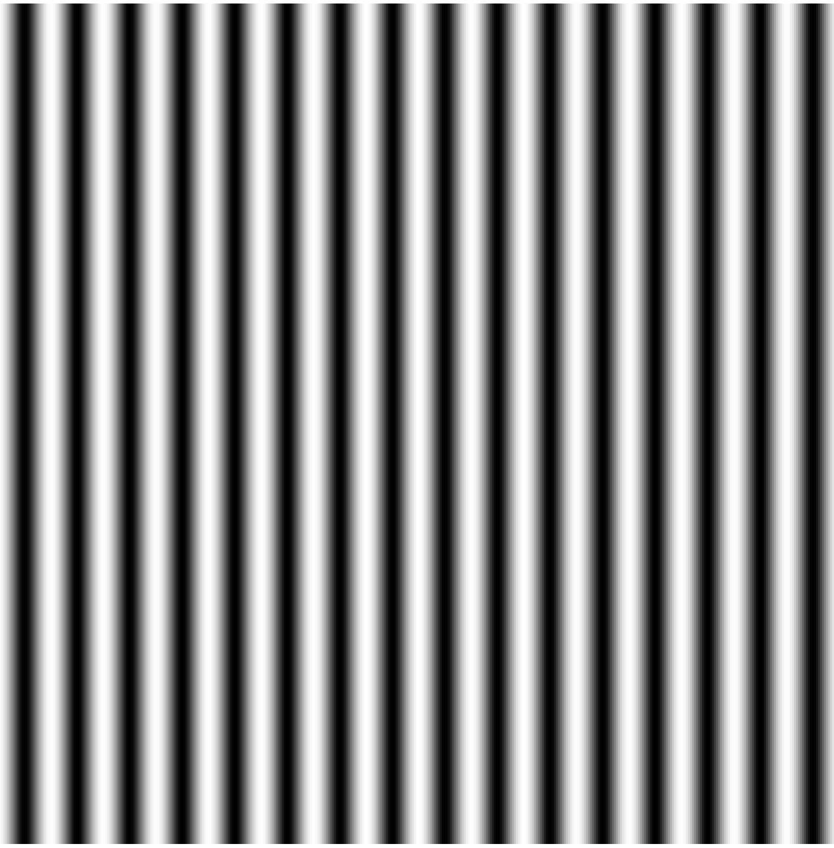
Fourier Transform of Constant



Spatial Domain

Frequency Domain

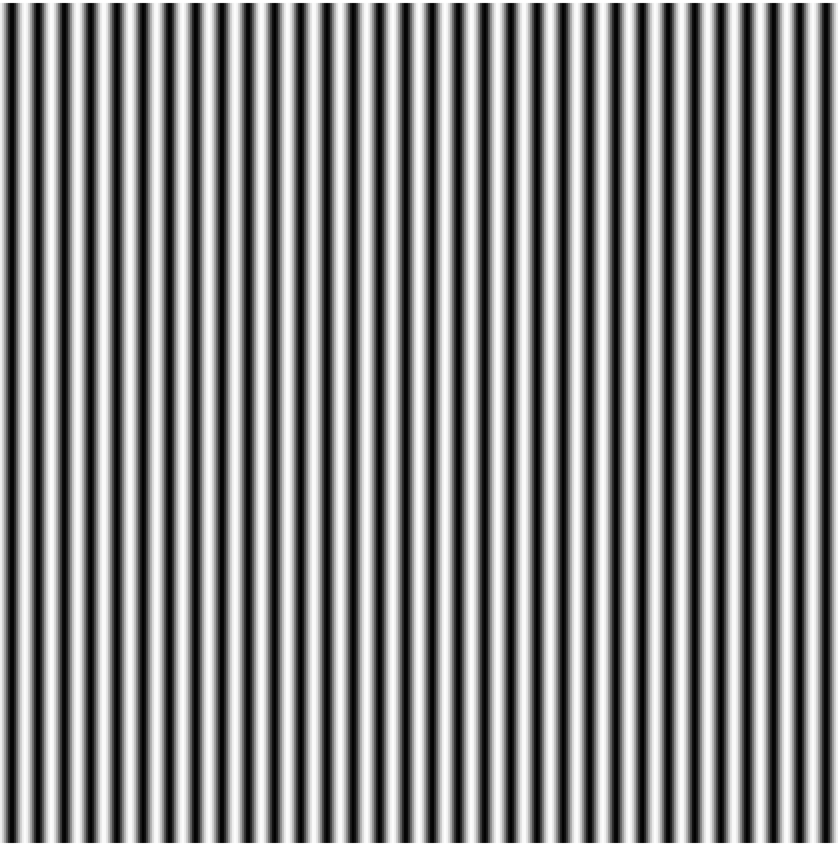
$\sin(2\pi/32)x$



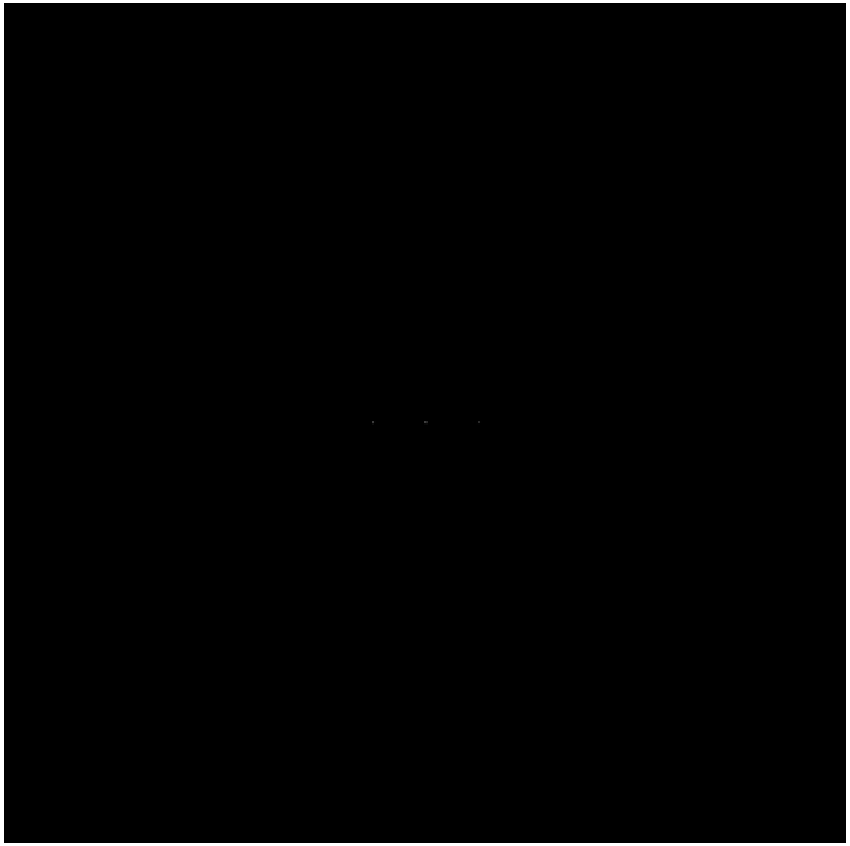
Spatial Domain **Frequency Domain**

Frequency = 1/32; 32 pixels per cycle

$\sin(2\pi/16)x$



Spatial Domain



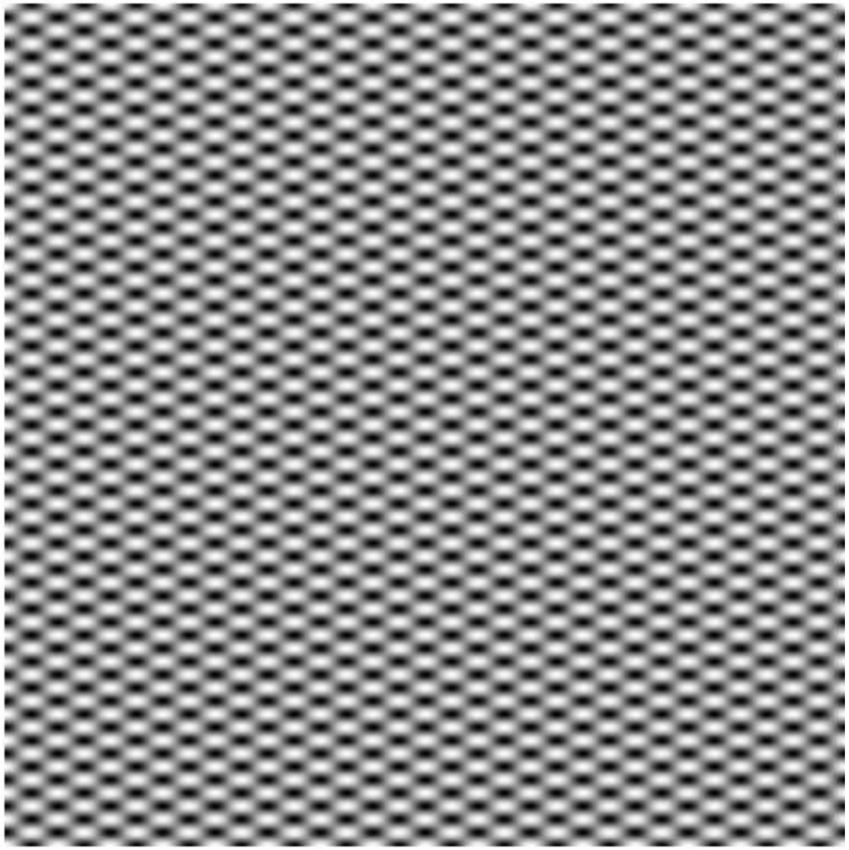
Frequency Domain

$$\sin\left(\frac{2\pi}{16}\right)y$$

Spatial Domain

Frequency Domain

$$\sin\left(\frac{2\pi}{32}\right)x \times \sin\left(\frac{2\pi}{16}\right)y$$

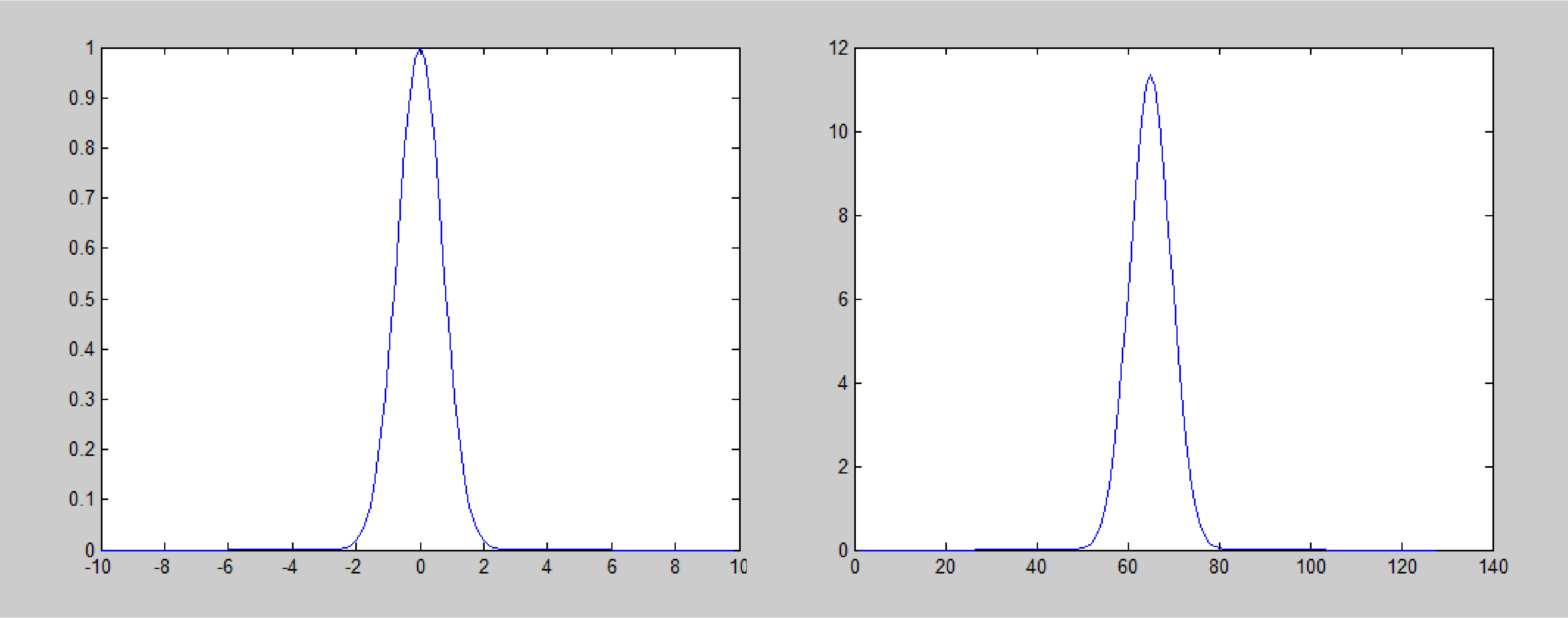


Spatial Domain

Frequency Domain

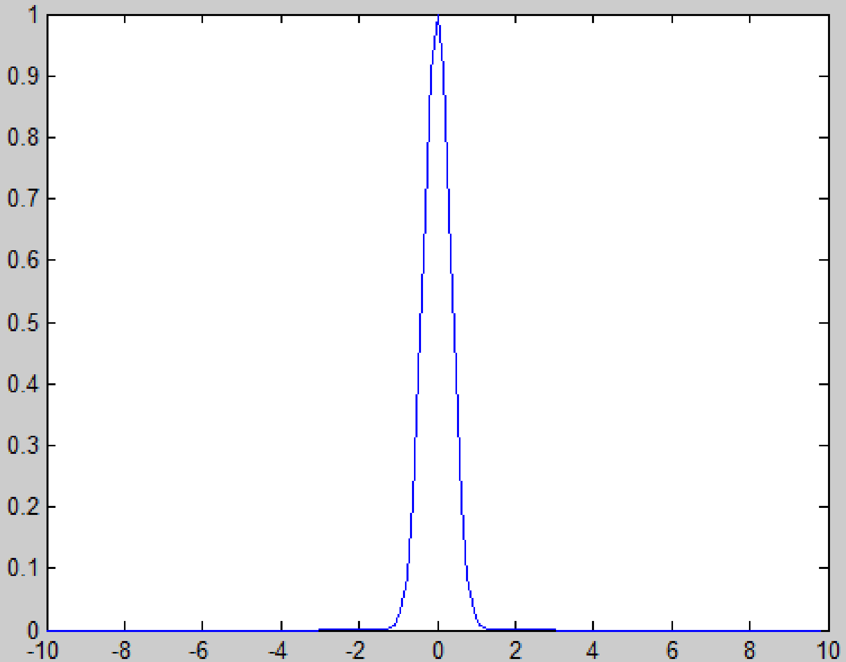
Fourier Transform of Gaussian

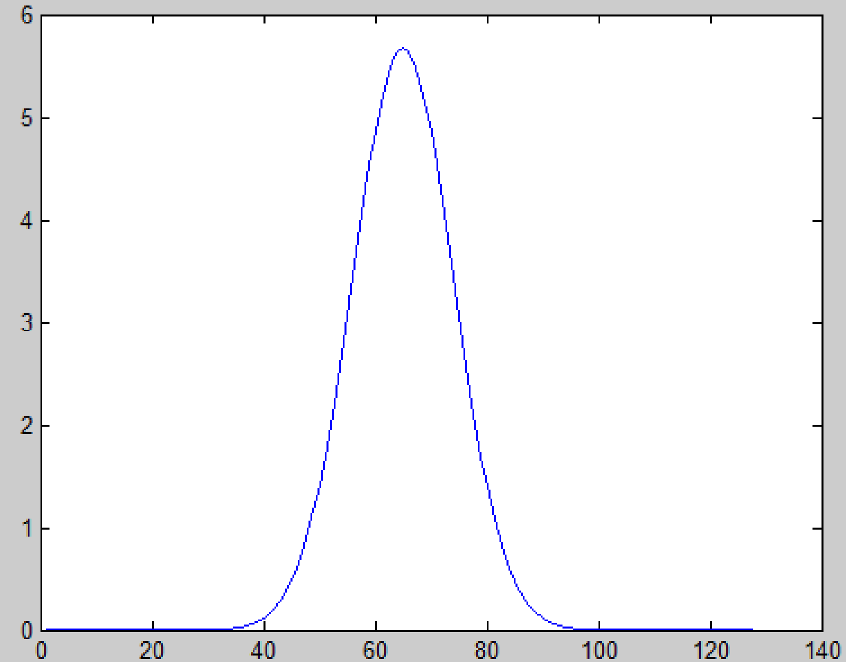
- a = 1 $$f(x) = \exp(-\alpha x^2) \quad \rightarrow \quad \widehat{f}(k) = \frac{1}{\sqrt{2\alpha}} \exp\left(-\frac{k^2}{4\alpha}\right)$$



Fourier Transform of Gaussian

- a = 4 $$f(x) = \exp(-\alpha x^2) \quad \rightarrow \quad \widehat{f}(k) = \frac{1}{\sqrt{2\alpha}} \exp\left(-\frac{k^2}{4\alpha}\right)$$

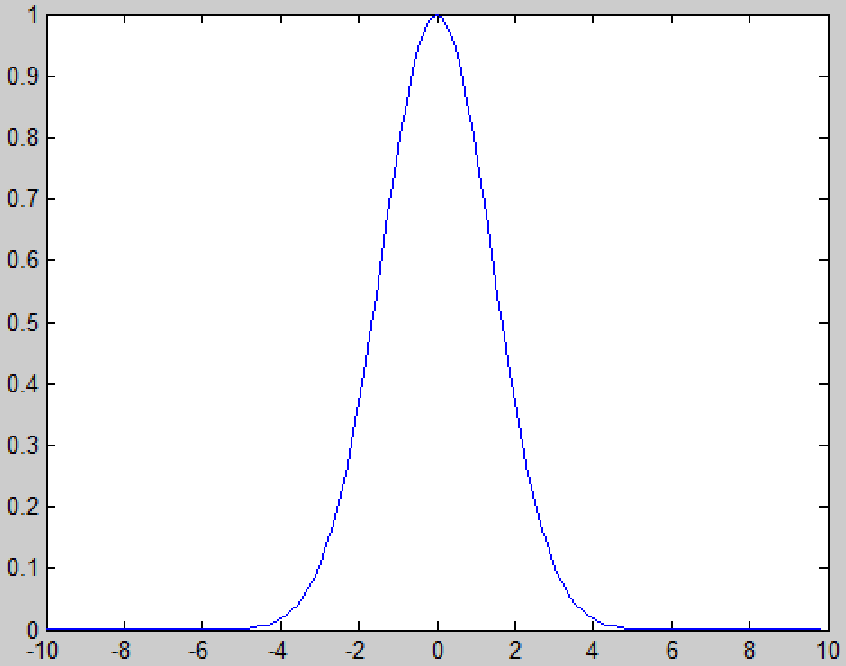


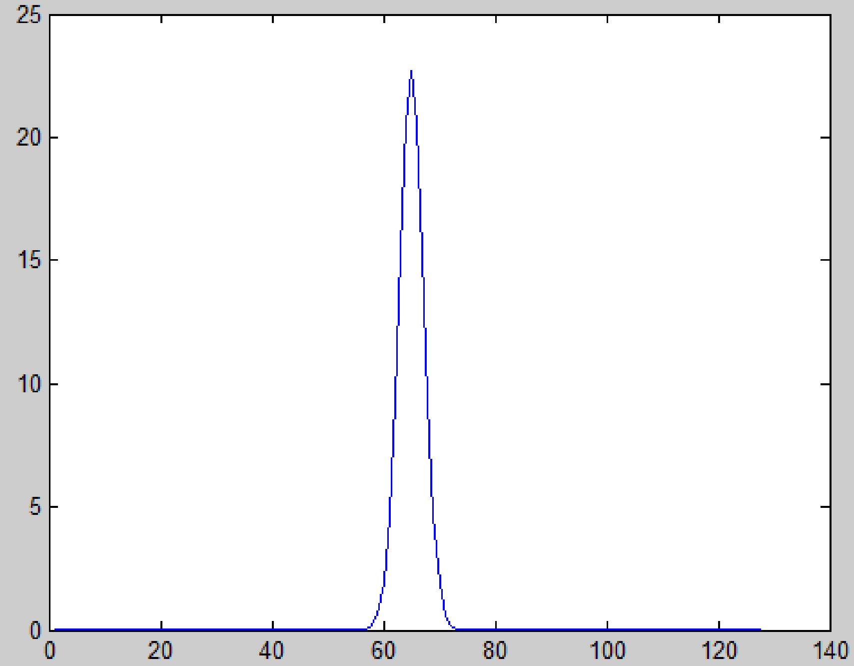


Fourier Transform of Gaussian

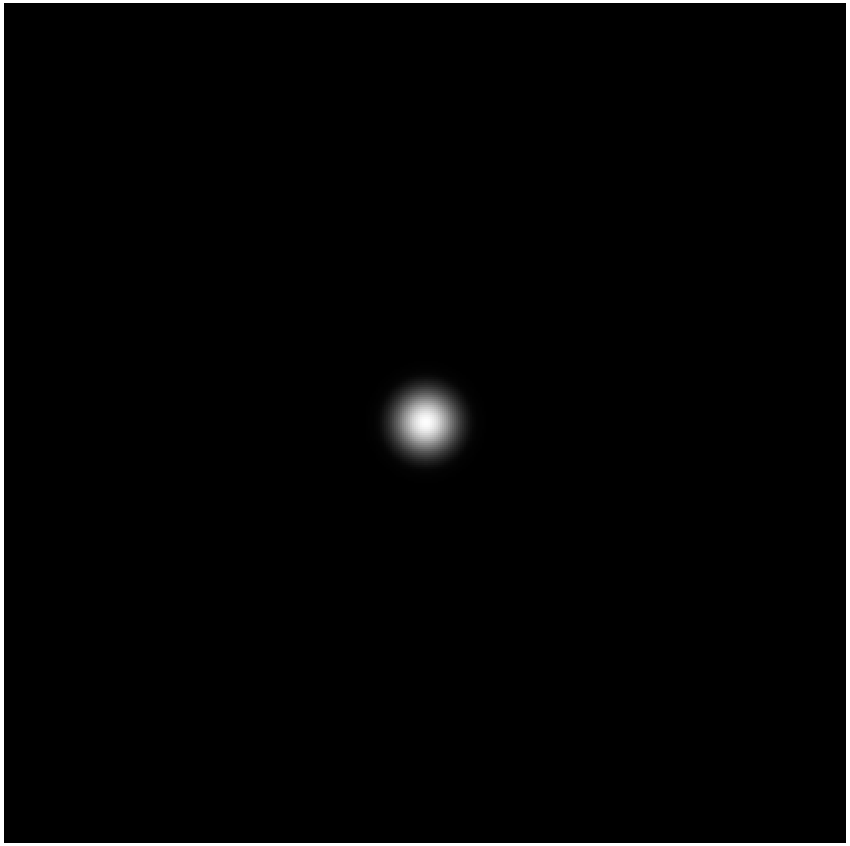
- a = 0.25

$$f(x) = \exp(-\alpha x^2) \quad \rightarrow \quad \widehat{f}(k) = \frac{1}{\sqrt{2\alpha}} \exp\left(-\frac{k^2}{4\alpha}\right)$$

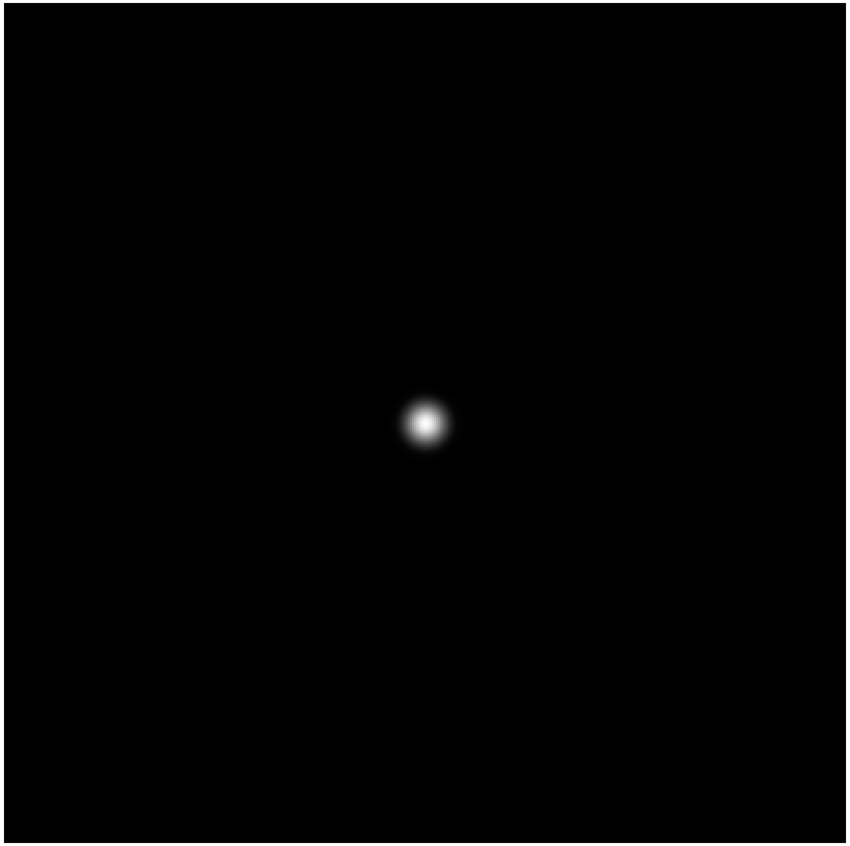




$$e^{-r^2/16^2}$$

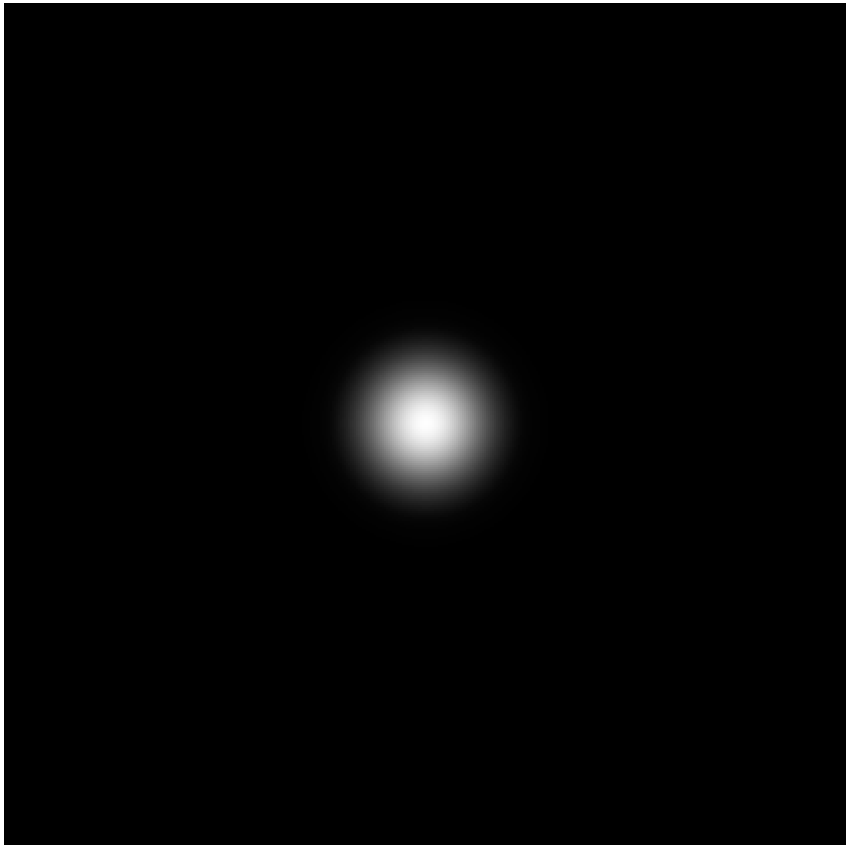


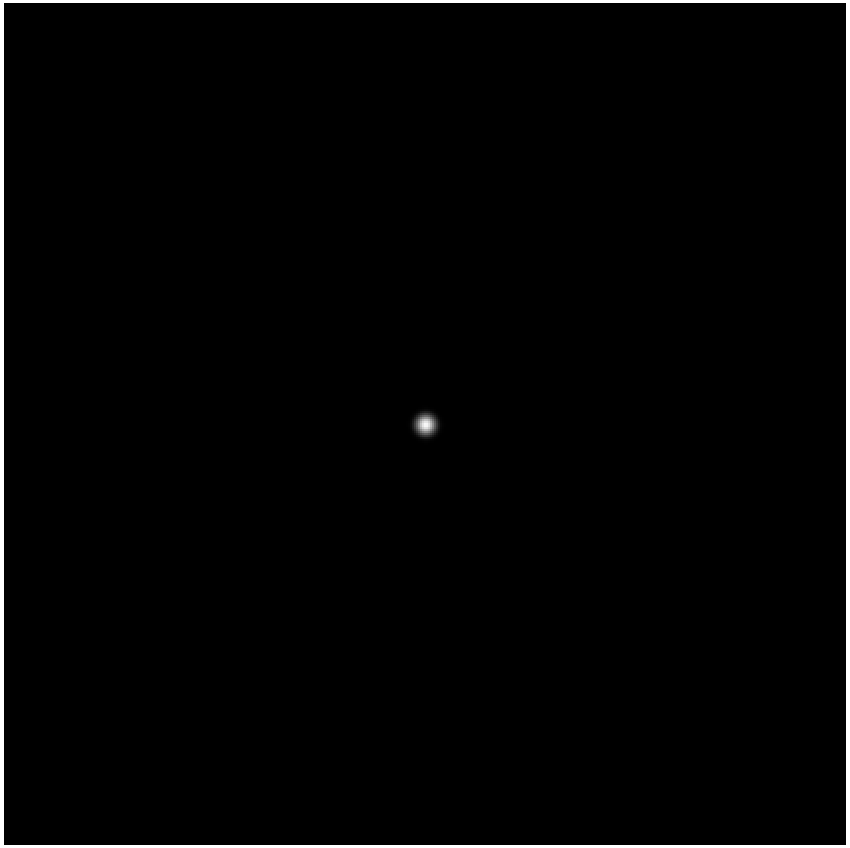
Spatial Domain



Frequency Domain

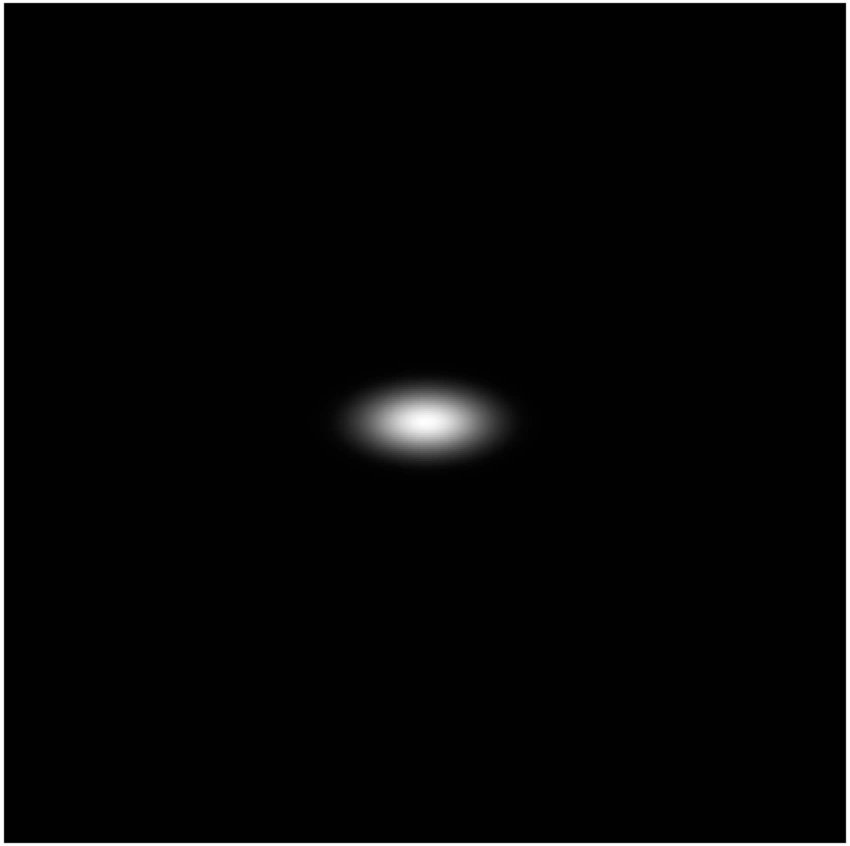
$e^{-r^2/32^2}$

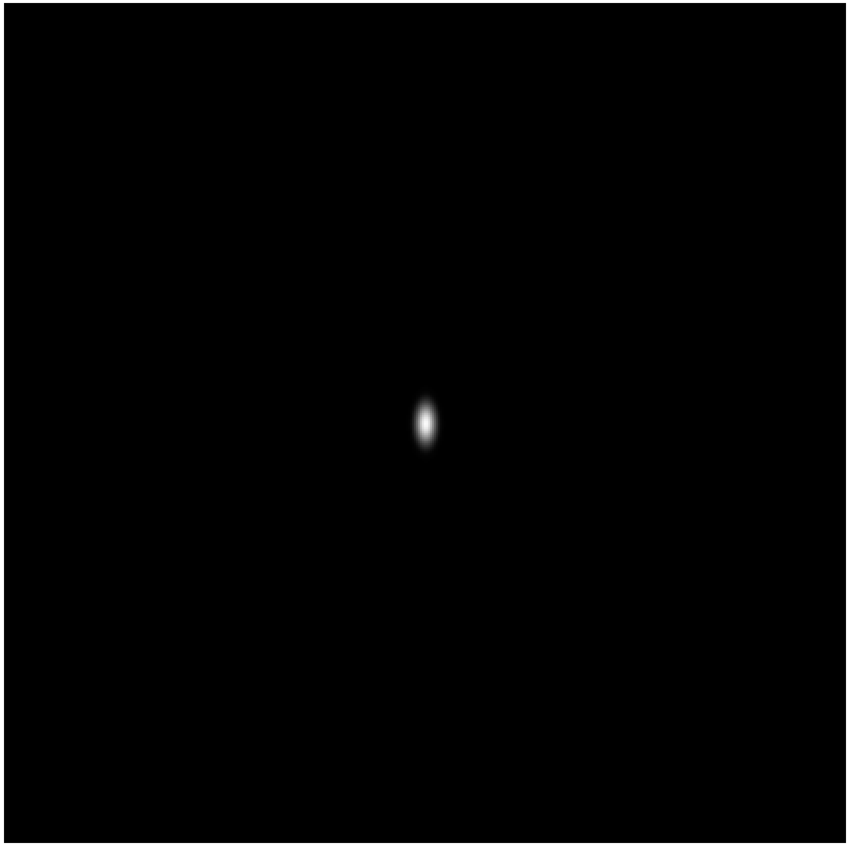




Spatial Domain **Frequency Domain**

$$e^{-x^2/32^2} \times e^{-y^2/16^2}$$

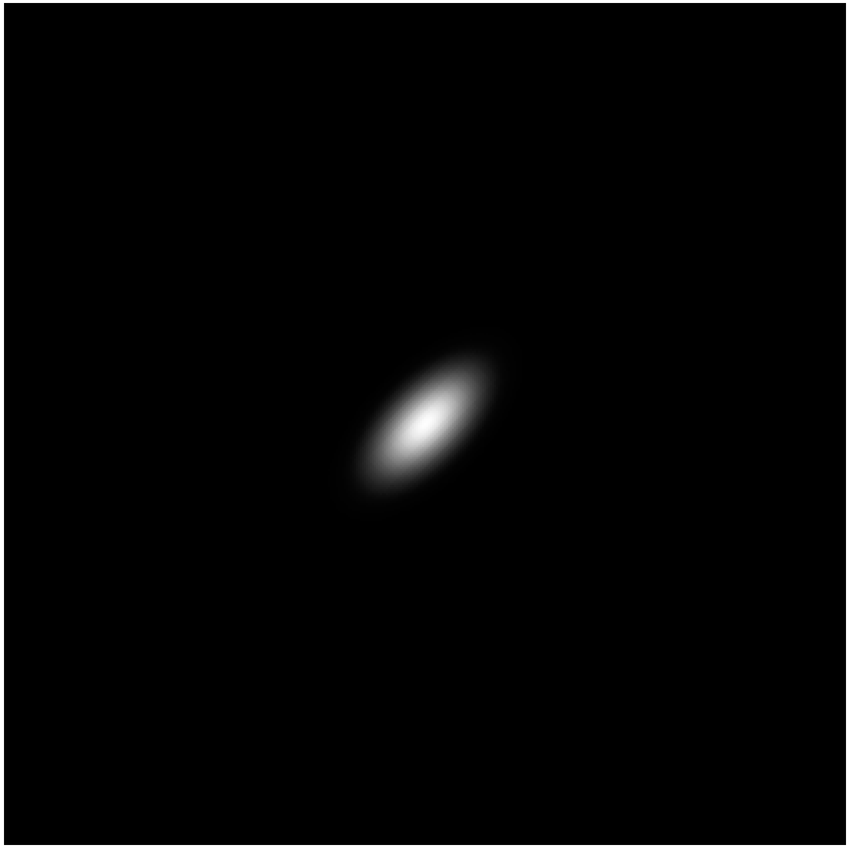




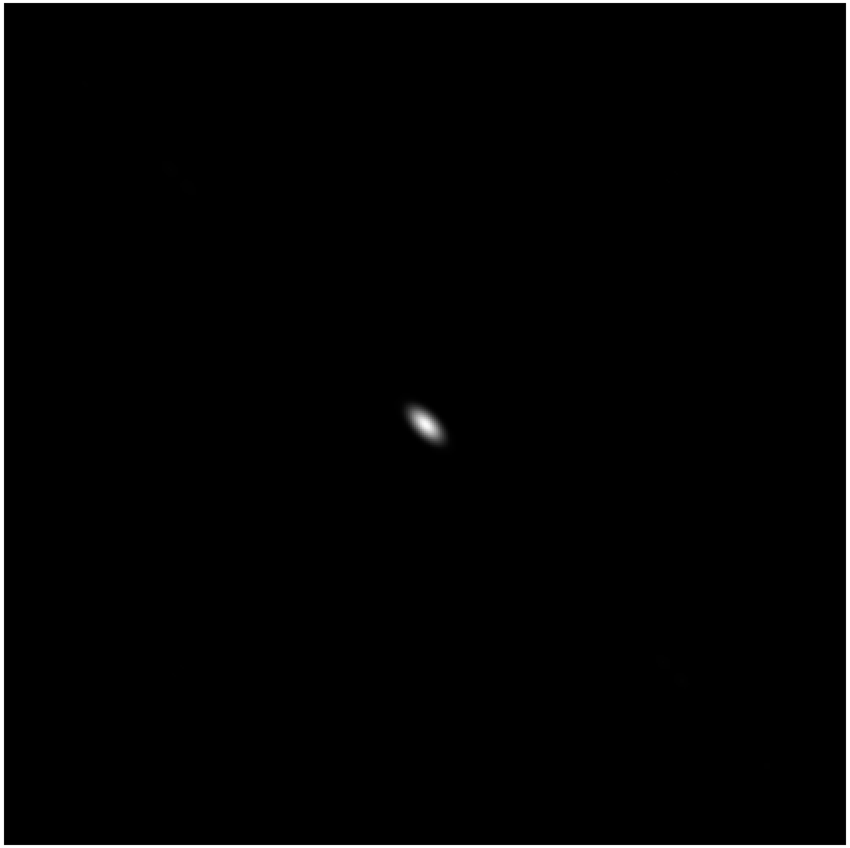
Spatial Domain

Frequency Domain

Rotate 45 $e^{-x^2/32^2} \times e^{-y^2/16^2}$

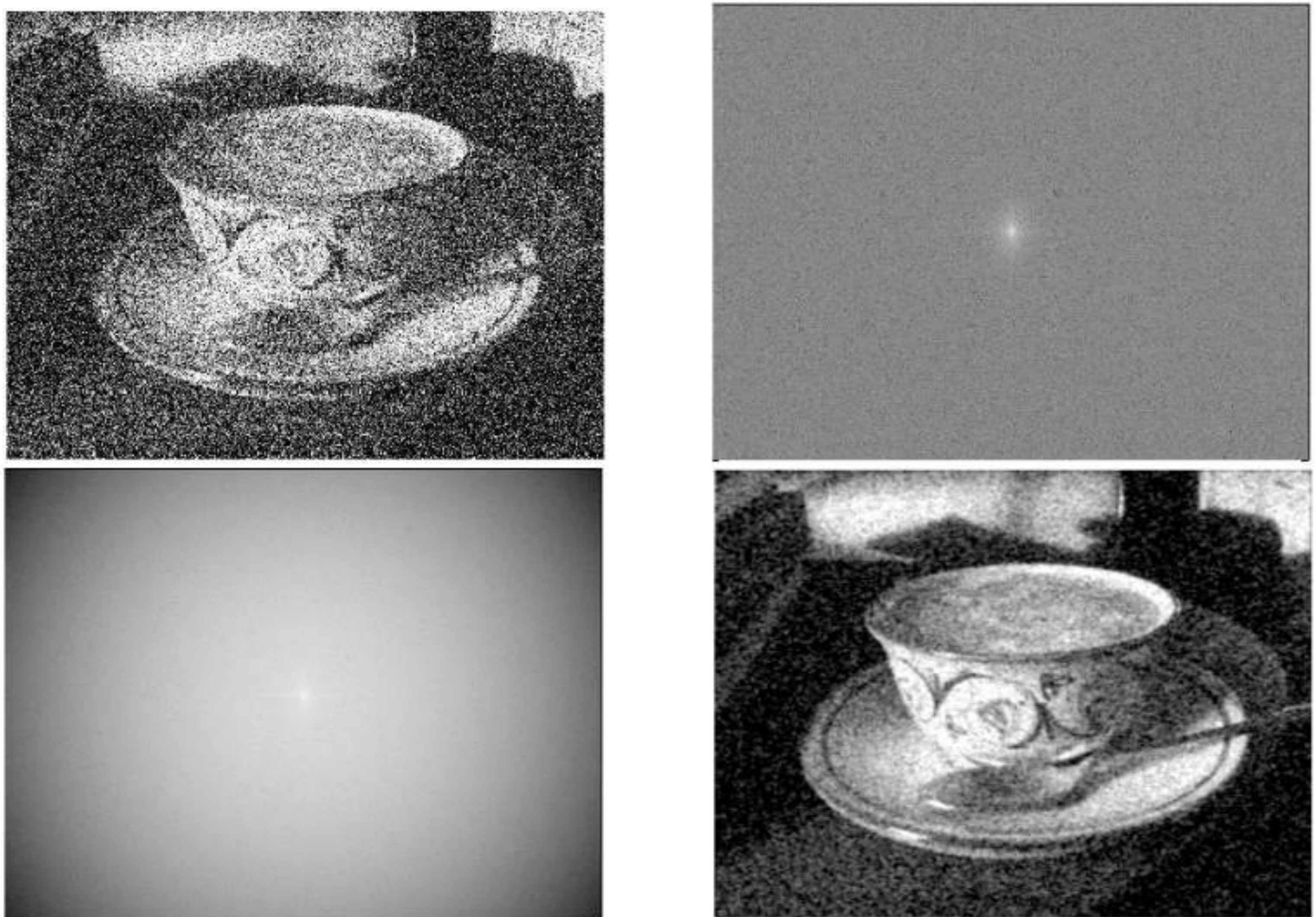


Spatial Domain

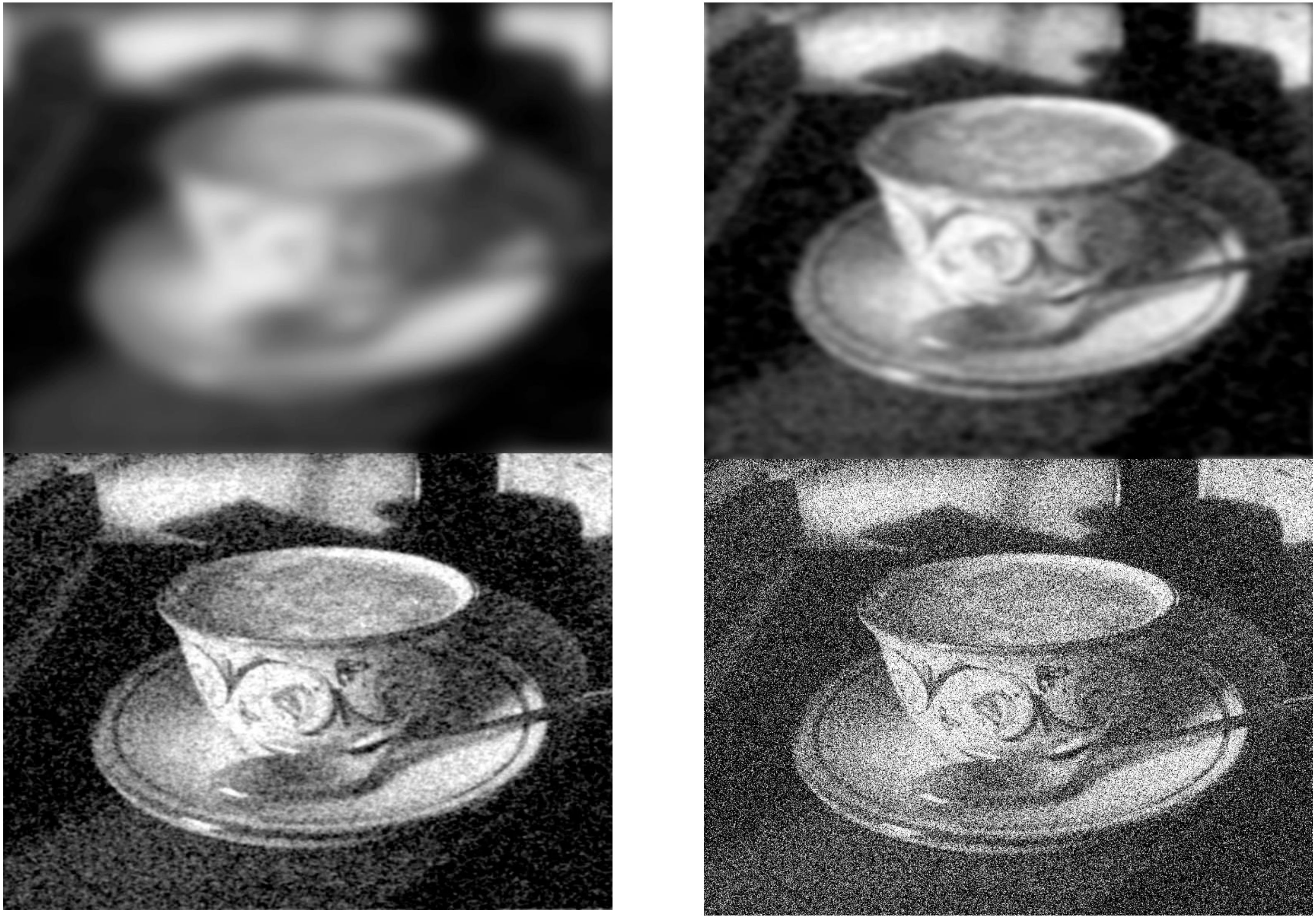


Frequency Domain

Linear Filtering for Image Denoising



Linear Filtering for Image Denoising



Fourier Transform

- Fourier transform

Define $W_N = e^{-j\frac{2\pi}{N}}$
(or simply W when N is evident from the context)

Analysis formula:

$$\mathbf{X} = \mathbf{W}\mathbf{x}$$

Synthesis formula:

$$\mathbf{x} = \frac{1}{N}\mathbf{W}^H\mathbf{X}$$

Change of basis matrix $\mathbf{W}$ with $\mathbf{W}[n,m] = W_N^{nm}$:

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W^1 & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{2(N-1)} \\ & & & \dots & & \\ 1 & W^{N-1} & W^{2(N-1)} & W^{3(N-1)} & \dots & W^{(N-1)^2} \end{bmatrix}$$

Fourier Transform

- Fourier transform $O(N^2)$
  - Same as LU decomposition in solving Ax = b
  - Local expansion in finite difference method
  - Global expansion
    - Every basis function is evaluated on the entire domain

$$F_n = \sum_{k=0}^{N-1} w^{nk} f_k \qquad 0 \le n \le N-1$$

command	expansion	boundary conditions
fft	$F_k = \sum_{j=0}^{2N-1} f_j \exp(i\pi jk/N)$	periodic: $f(0) = f(L)$
dst	$F_k = \sum_{j=1}^{N-1} f_j \sin(\pi jk/N)$	pinned: $f(0) = f(L) = 0$
dct	$F_k = \sum_{j=0}^{N-2} f_j \cos(\pi jk/2N)$	no-flux: $f'(0) = f'(L) = 0$

Fast Fourier Transform

$$F_n = \sum_{k=0}^{N-1} w^{nk} f_k \qquad 0 \le n \le N-1$$

- Fast Fourier transform $O(N^2) \rightarrow O(NlogN)$

$$(F_N)_{jk} = w_n^{jk} = \exp(i2\pi jk/N) \qquad \mathbf{y} = \mathbf{F}_N \mathbf{x}$$

$$w_{2n}^2 = \exp(i2\pi/(2n)) \exp(i2\pi/(2n)) = \exp(i2\pi/n) = w_n$$

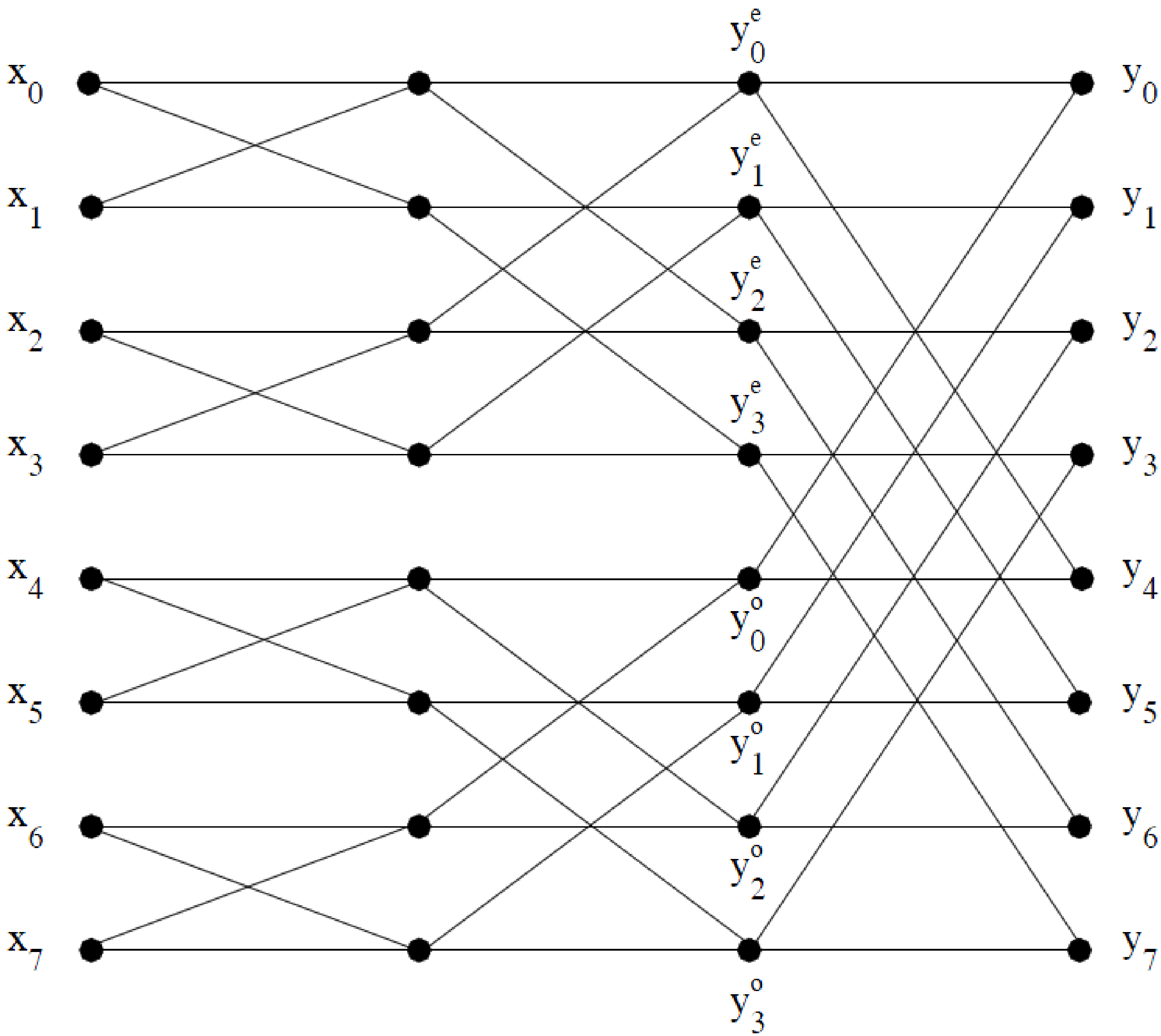
$$\mathbf{x}^e = \begin{pmatrix} x_0 \\ x_2 \\ x_4 \\ \vdots \\ x_{N-2} \end{pmatrix} \quad \text{and} \quad \mathbf{x}^o = \begin{pmatrix} x_1 \\ x_3 \\ x_5 \\ \vdots \\ x_{N-1} \end{pmatrix}$$

$$\mathbf{y}^e = \mathbf{F}_M \mathbf{x}^e \quad \text{and} \quad \mathbf{y}^o = \mathbf{F}_M \mathbf{x}^o$$

$$y_n = y_n^e + w_N^n y_n^o \quad n = 0, 1, 2, \ldots, M-1$$

$$y_{n+M} = y_n^e - w_N^n y_n^o \quad n = 0, 1, 2, \ldots, M-1$$

Fast Fourier Transform



Derivative Relations

- Fourier transform of $f'(x)$

$$\int u(x)v'(x)\,dx = u(x)v(x) - \int u'(x)v(x)\,dx$$

  - Fourier transform of $f(x)$ is $\widehat{f(x)}$

$$\widehat{f'(x)} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-ikx} f'(x)dx = f(x)e^{-ikx}|_{-\infty}^{\infty} + \frac{ik}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-ikx} f(x)dx$$

  - Assuming that $f(x) \to 0$ as $x \to \mp\infty$

$$\widehat{f'(x)} = \frac{ik}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{ikx} f(x)dx = ik\widehat{f(x)}$$

$$\widehat{f'} = ik\widehat{f}$$

$$\widehat{f^{(n)}} = (ik)^n\widehat{f}$$

i 复数符号a+ib, i * i = -1
k 波数, 频率, $f(x)$包含128个点, $k \in [-64,64]$

Derivative Relations

- Example

$$\widehat{f^{(n)}} = (ik)^n \widehat{f}$$

$$y'' - \omega^2 y = -f(x) \quad x \in [-\infty, \infty]$$

$$\begin{aligned}
\widehat{y''} - \omega^2 \widehat{y} &= -\widehat{f} \\
-k^2 \widehat{y} - \omega^2 \widehat{y} &= -\widehat{f} \\
(k^2 + \omega^2)\widehat{y} &= \widehat{f} \\
\widehat{y} &= \frac{\widehat{f}}{k^2 + \omega^2}
\end{aligned}$$

$$y(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{ikx} \frac{\widehat{f}}{k^2 + \omega^2} dk$$

Spectral Methods

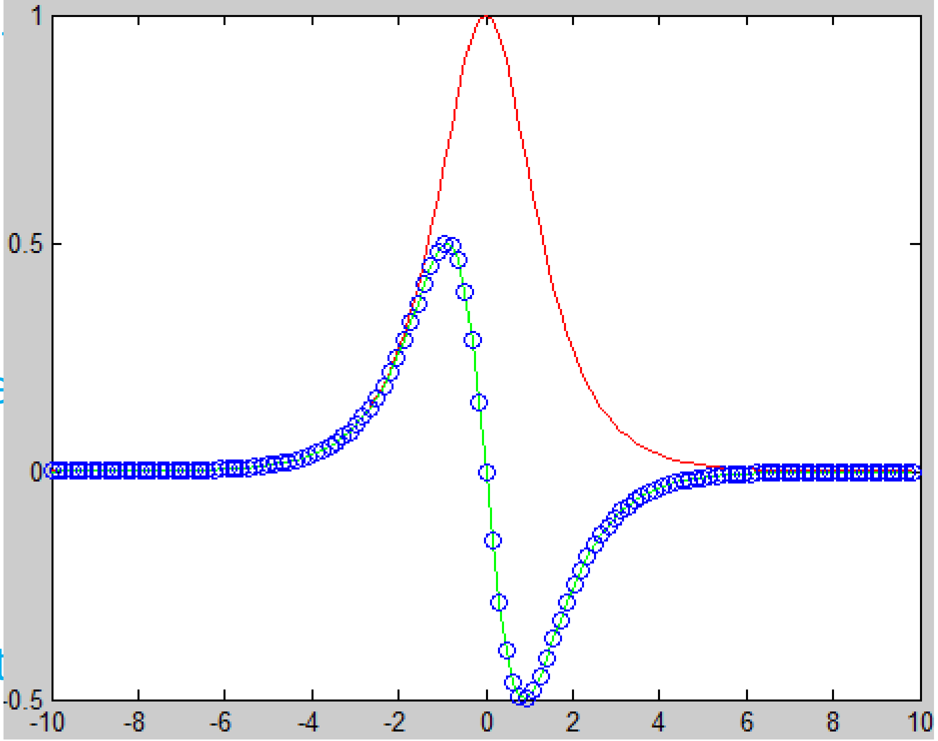
- $f(x) = \operatorname{sech}(x),\ f'^{(x)} = -\operatorname{sech}(x)\tanh(x)$

```
L=20; % define the computational domain [-L/2,L/2]
n=128; % define the number of Fourier modes 2^n
x2=linspace(-L/2,L/2,n+1); % define the domain discretization
x=x2(1:n); % consider only the first n poin

u=sech(x); % function to take a derivative
ut=fft(u); % FFT the function

k=(2*pi/L)*[0:(n/2-1) (-n/2):-1]; % k rescale
ut1=i*k.*ut; % first derivative
u1=ifft(ut1); % inverse transform

u1exact=-sech(x).*tanh(x); % analytic first
```



Spectral Methods

- General PDE $\frac{\partial u}{\partial t} = Lu + N(u)$
  - *L* is a linear, constant coefficient operator

$$L = ad^2/dx^2 + bd/dx + c$$

  - *N* includes the nonlinear and non-constant coefficient terms

$$N(u) = u^3 + f(x)u + g(x)d^2u/dx^2$$

- Apply Fourier transform (PDE → ODE)

$$\frac{d\widehat{u}}{dt} = \alpha(k)\widehat{u} + \widehat{N(u)}$$

Spectral Methods

- Linear terms

$$\frac{d\widehat{u}}{dt} = \alpha(k)\widehat{u} + \widehat{N(u)}$$

$$Lu = a\frac{d^2u}{dx^2} + b\frac{du}{dx} + cu$$

$$\begin{aligned}(ik)^2 a\widehat{u} + b(ik)\widehat{u} + c\widehat{u} \\ &= (-k^2 a + ibk + c)\widehat{u} \\ &= \alpha(k)\widehat{u}\end{aligned}$$

Spectral Methods

- Nonlinear terms $N(u) = u^3 + f(x)u + g(x)d^2u/dx^2$

1. $f(x)du/dx$

    - determine $du/dx \rightarrow \widehat{du/dx} = ik\hat{u}, du/dx = FFT^{-1}(ik\hat{u})$
    - multiply by $f(x) \rightarrow f(x)du/dx$
    - Fourier transform $FFT(f(x)du/dx)$

3. $u^3 d^2u/dx^2$

    - determine $d^2u/dx^2 \rightarrow d^2\widehat{u/d}x^2 = (ik)^2\hat{u}, d^2u/dx^2 = FFT^{-1}(-k^2\hat{u})$
    - multiply by $u^3 \rightarrow u^3 d^2u/dx^2$
    - Fourier transform $FFT(u^3 d^2u/dx^2)$

Outline

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
    - Fourier transform
    - Examples
      - Poisson equations
      - Heat diffusion equations
      - Filtered pseudo-spectral
    - Comparison of spectral and finite difference methods
  - Finite elements (有限元法)

Poisson Equations

- Streamfunction equation $$\nabla^2\psi = \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = \omega$$

$$\widehat{\frac{\partial^2\psi}{\partial x^2}} + \widehat{\frac{\partial^2\psi}{\partial y^2}} = \widehat{\omega} \quad \rightarrow \quad -k_x^2\widehat{\psi} + \frac{\partial^2\widehat{\psi}}{\partial y^2} = \widehat{\omega}$$

$$-k_x^2\widetilde{\widehat{\psi}} + \widetilde{\frac{\partial^2\widehat{\psi}}{\partial y^2}} = \widetilde{\widehat{\omega}} \quad \rightarrow \quad -k_x^2\widetilde{\widehat{\psi}} + -k_y^2\widetilde{\widehat{\psi}} = \widetilde{\widehat{\omega}}$$

$$\widetilde{\widehat{\psi}} = -\frac{\widetilde{\widehat{\omega}}}{k_x^2 + k_y^2}$$

- Periodic boundary conditions → no unique solution

$$\psi(-L, -L, t) = 0$$

Poisson Equations

- Streamfunction equation
  - $k_x = k_y = 0$ at the zero mode

$$\widetilde{\widehat{\psi}} = -\frac{\widetilde{\widehat{\omega}}}{k_x^2 + k_y^2}$$

- Option 1

$$\widetilde{\widehat{\psi}} = -\frac{\widetilde{\widehat{\omega}}}{k_x^2 + k_y^2 + eps}$$

- Option 2

```
kx(1)=10^(-6);
ky(1)=10^(-6);
```

Heat Diffusion Equations

- Diffusion equation $u_t = div(D \cdot \nabla u)$
- $D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\nabla u = (u_x, u_y)^T$
- $u_t = div\left((u_x, u_y)^T\right) = u_{xx} + u_{yy}$
- $\widehat{u_t} = -({k_x}^2 + {k_y}^2)\hat{u}$
- $\widehat{u_t} = -(k_x, k_y)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}(k_x, k_y)^T\hat{u}$
- $\hat{u} = \widehat{u_0}e^{-(k \cdot D \cdot k^T)t}$, $k = (k_x, k_y)$

Heat Diffusion Equations

- Heat diffusion equation $u_t = D\nabla^2 u$

$$\hat{u}_t = -D(k_x^2 + k_y^2)\hat{u} \quad \longrightarrow \quad \hat{u} = \hat{u}_0 e^{-D(k_x^2+k_y^2)t}$$

- The wavenumbers (spatial frequencies) decay according to a Gaussian function
- Linear filtering with a Gaussian is equivalent to a linear diffusion of the image for periodic boundary conditions

Heat Diffusion Equations

- Linear diffusion = Gaussian filter $u_t = D\nabla^2 u$
- Heat diffusion equation $u_t = \nabla \cdot (D(x,y)\nabla u)$
  - D(x, y) is a spatial diffusion coefficient
  - Particular regions in the spatial figure can be targeted by modification of the diffusion coefficient D(x, y)
- Nonlinear filtering $u_t = \nabla \cdot (D(u, \nabla u)\nabla u)$
  - Enhance certain features of the image

Heat Diffusion Equations

- Hyper-diffusion $$\frac{\partial u}{\partial t} = -\frac{\partial^4 u}{\partial x^4}$$

$$\frac{d\widehat{u}}{dt} = -(ik)^4\widehat{u} = -k^4\widehat{u}$$

$$\widehat{u} = \widehat{u}_0 \exp(-k^4 t)$$

- Difficulty for time-stepping scheme
  - N=1024, $k_{\max}^4 = 6.8 \times 10^{10}$
  - Even if the solution is small at the high wavenumbers

$$\frac{d\widehat{u}}{dt} = -(10^{10})(10^{-6}) = -10^4$$

Numerical stiffness

Filtered Pseudo-Spectral

- Example $$\frac{dy}{dt} + p(t)y = g(t)$$

$$\mu \frac{dy}{dt} + \mu p(t)y = \mu g(t) \qquad (\mu y)' = \mu' y + \mu y'$$

$$\frac{d}{dt}(\mu y) = \mu g(t) \qquad d\mu/dt = \mu p(t)$$

$$y = \frac{1}{\mu}\left[\int \mu(t)g(t)dt + c\right] \qquad \mu(t) = \exp\left(\int p(t)dt\right)$$

Filtered Pseudo-Spectral

- General PDE $$\frac{\partial u}{\partial t} = Lu + N(u)$$

$$\frac{d\widehat{u}}{dt} = \alpha(k)\widehat{u} + \widehat{N(u)}$$

$$\frac{d\widehat{u}}{dt} - \alpha(k)\widehat{u} = \widehat{N(u)}$$

$$\frac{d\widehat{u}}{dt}\exp(-\alpha(k)t) - \alpha(k)\widehat{u}\exp(-\alpha(k)t) = \exp(-\alpha(k)t)\widehat{N(u)}$$

$$\frac{d}{dt}[\widehat{u}\exp(-\alpha(k)t)] = \exp(-\alpha(k)t)\widehat{N(u)}$$

$$\frac{d\widehat{v}}{dt} = \exp(-\alpha(k)t)\widehat{N(u)} \qquad \widehat{v} = \widehat{u}\exp(-\alpha(k)t)$$

$$\widehat{u} = \widehat{v}\exp(\alpha(k)t)$$

Filtered Pseudo-Spectral

- Fisher-Kolmogorov Equation $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u^3 + cu$

$$\frac{d\widehat{u}}{dt} = (ik)^2 \widehat{u} + \widehat{u^3} + c\widehat{u}$$

$$\frac{d\widehat{u}}{dt} + (k^2 - c)\widehat{u} = \widehat{u^3} \qquad \alpha(k) = c - k^2$$

$$\frac{d\widehat{v}}{dt} = \exp[-(c - k^2)t]\widehat{u^3}$$

$$\widehat{u} = \widehat{v}\exp[(c - k^2)t]$$

Filtered Pseudo-Spectral

- When solving the general equation $\frac{\partial u}{\partial t} = Lu + N(u)$

  it is important to only step forward ∇t in time with

$$\frac{d\widehat{v}}{dt} = \exp(-\alpha(k)t)\widehat{N(u)}$$
$$\widehat{u} = \widehat{v}\exp(\alpha(k)t)$$

before the nonlinear term $\widehat{N(u)}$ is updated

- The computational savings for this method generally does not manifest itself unless there are more than two spatial derivatives in the highest derivative of Lu

Outline

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
    - Fourier transform
    - Examples
      - Poisson equations
      - Heat diffusion equations
      - Filtered pseudo-spectral
    - Comparison of spectral and finite difference methods
  - Finite elements (有限元法)

Comparison - Accuracy

- Finite differences
  - Determined by ∇x and ∇y chosen in the discretization , and fairly easy to compute
  - Generally much worse than spectral methods
- Spectral method
  - Rely on a global expansion, called spectrally accurate
  - Have infinite order accuracy, generally of much higher accuracy than finite differences

Comparison - Implementation

- Finite differences
  - Most difficulty is to generate the correct sparse matrices, Complicated with higher order schemes and in higher dimensions
  - $Ax = b$, check $\det(A) = 0$
- Spectral method
  - Difficulty is the continual switching between the time or space domain and the spectral domain

Comparison – Computational Efficiency

- Finite differences
  - Determined by the size of the matrices and vectors in solving Ax = b
    - Generally $O(N^2)$ efficiency by using LU decomposition
    - Iterative schemes can lower the operation count, but there are no guarantees about this
- Spectral method
  - FFT is $O(NlogN)$, faster than finite difference

Comparison – Boundary Conditions

- Finite differences
  - Generic boundary conditions
  - More complicated computational domains
- Spectral method
  - FFT method – only periodic boundary condition
  - Discrete sine – pinned boundary conditions
  - Discrete cosine – no-flux boundary conditions

Comparison Spectral and Finite Difference Methods

	有限差分法	频谱法 (Fourier Transform)
近似方式	Taylor expansions (Local)	Fourier expansions (Global)
Accuracy 正确性	Determined by ∇x and ∇y Easy to analysis	infinite order spectrally accurate Not easy to analysis
Implementation 实现难点	Generate correct spare matrices with higher order schemes and in higher dimensions	Continual switching between the time or space domain and the spectral domain
计算效率	Generally $O(N^2)$	Generally $O(NlogN)$
Boundary Conditions 边界条件	Generic boundary conditions Complicated computational domains	FFT (periodic) DST (pinned) DCT (non-flux)
Stability 稳定性	$Ax = b$ (A singular matrix) $\frac{d\boldsymbol{u}}{\mathrm{dt}} = \frac{k}{\delta^2}\boldsymbol{Au}$ (CFL number)	$\hat{u} = -\hat{f}/k^2 \ \ (k^2 != 0)$ $\frac{d\hat{\boldsymbol{u}}}{\mathrm{dt}} = -\boldsymbol{k}^2\hat{\boldsymbol{u}}$

Outline

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
  - Finite elements (有限元法)
    - Basic method
    - Calculus of variations

Finite Element Methods

- Finite element method is ideally suited for
  - Complex boundaries
  - Very general boundary conditions
  - Perhaps not as fast as finite difference and spectral
- Basic steps (similar to the above methods)
  - Discretize computational domain
  - Solve the matrix problem $Ax = b$

Finite Element Methods

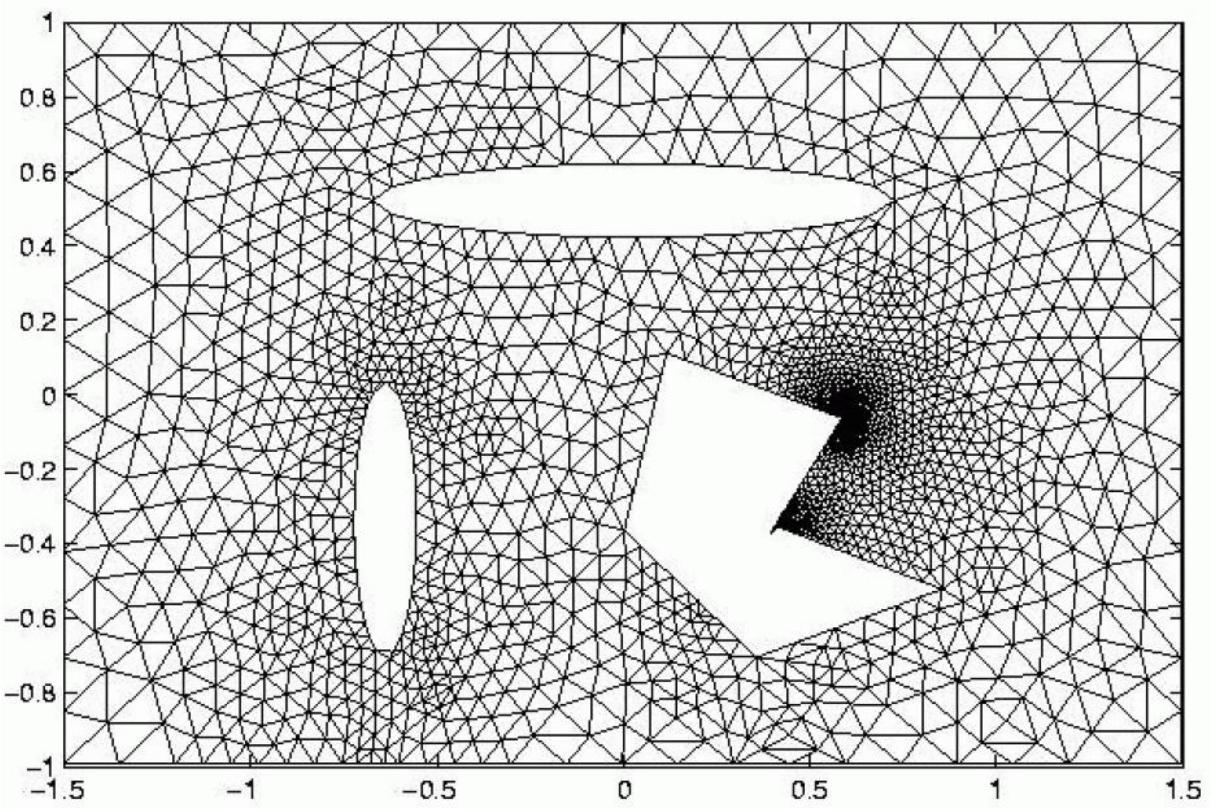
- Finite difference methods
  - Taylor series to locally expand the solution
- Spectral methods
  - Sine and cosine basis functions to globally expand the solution
- Finite elements are another form of finite difference
  - Interpolate over patches of the solution region to approximate the solution (local)

Essential Steps in FEM

- Discretization of the computational domain
- Selection of the interpolating functions
- Derivation of characteristic matrices and vectors
- Assembly of characteristic matrices and vectors
- Solution of the matrix problem $Ax = b$

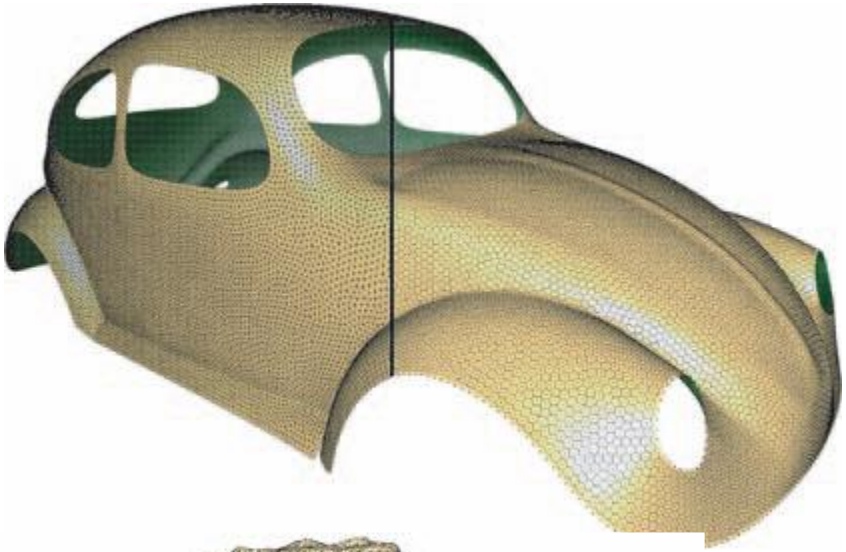
Domain Discretization

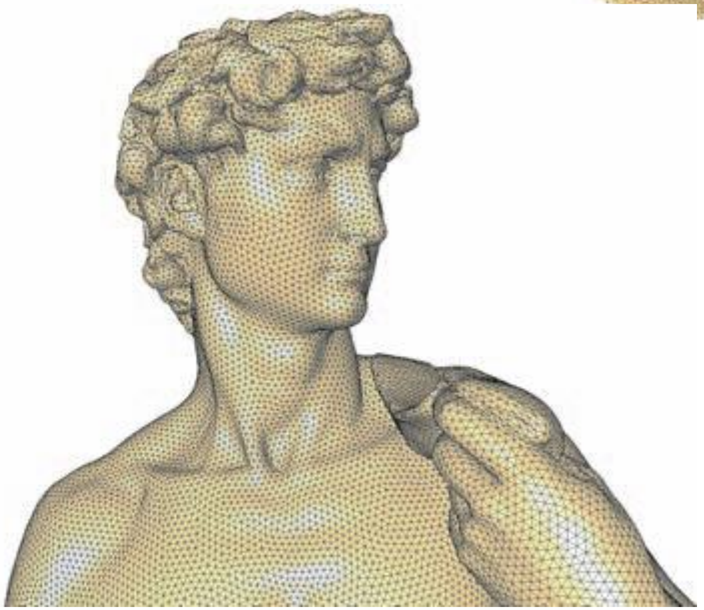
- Discretize the domain with triangular elements (or another appropriate element function)
  - Adaptive size



Domain Discretization

- Triangulation
- Quadrangulation
- Remeshing





Domain Discretization

- Key features
  - The width and height of all discretization triangles should be similar
  - All shapes used to span the computational domain should be approximated by polygons
  - A commercial package is almost always used to generate the grid unless you are doing research in this area

Interpolating Functions

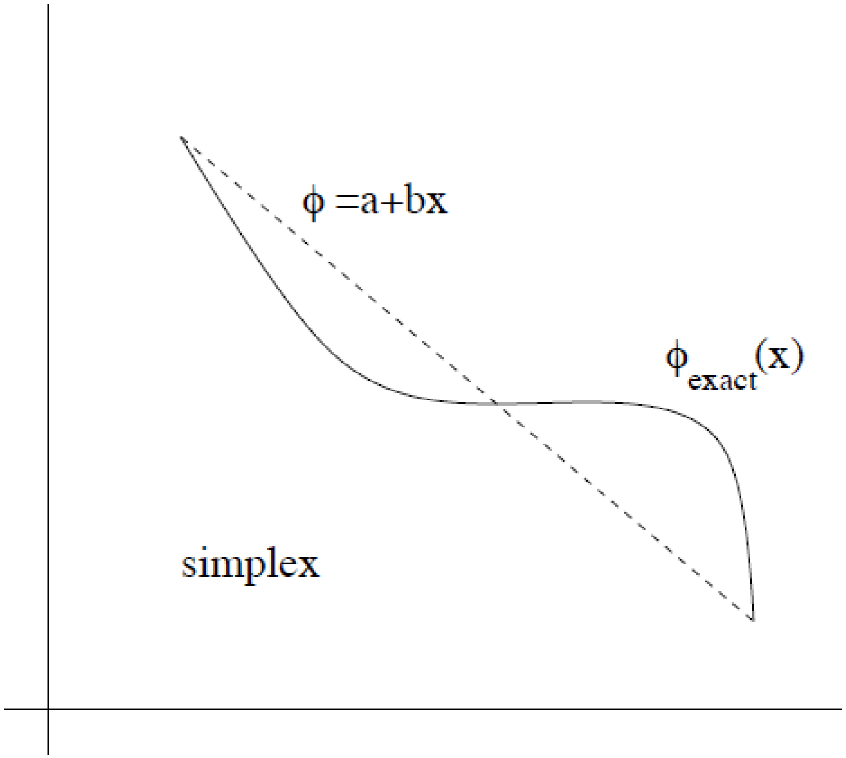
- Each element in the finite element basis relies on a piecewise approximation
- The solution is approximated by a simple function in each triangular (or polygonal) element
- The accuracy of the scheme depends upon the choice of the approximating function chosen

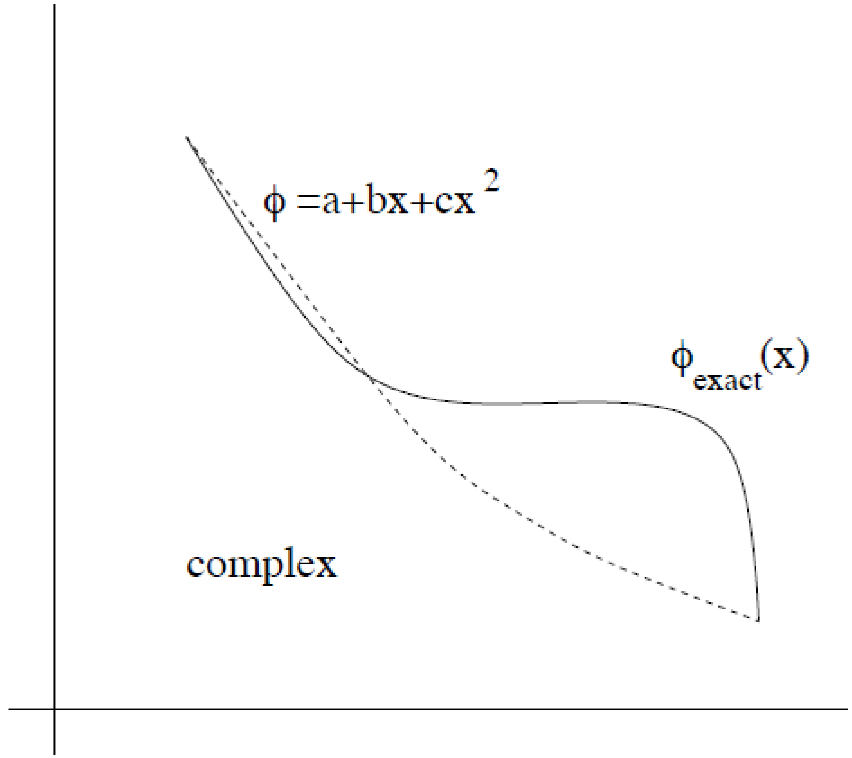
Interpolating Functions

- Polynomials
  - Low order polynomials → less computational cost
  - High order polynomials → more accuracy
- Three groups of elements
  - Simplex elements: linear polynomials
  - Complex elements: higher order polynomials
  - Multiplex elements: rectangles instead of triangles

Interpolating Functions

- Simplex and complex elements

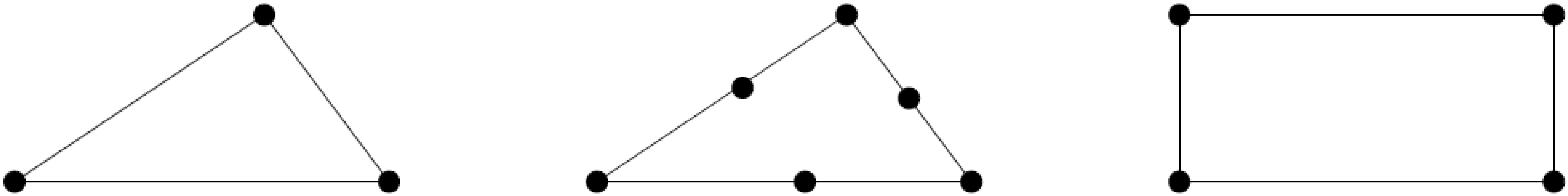




Interpolating Functions

- Simplex, complex and multiplex elements

$$\phi = a + bx + cy \qquad \phi = a + bx + cy + dx^2 + exy + fy^2 \qquad \phi = a + bx + cy + dxy$$



simplex complex multiplex

1D Simplex

- The approximation of the function

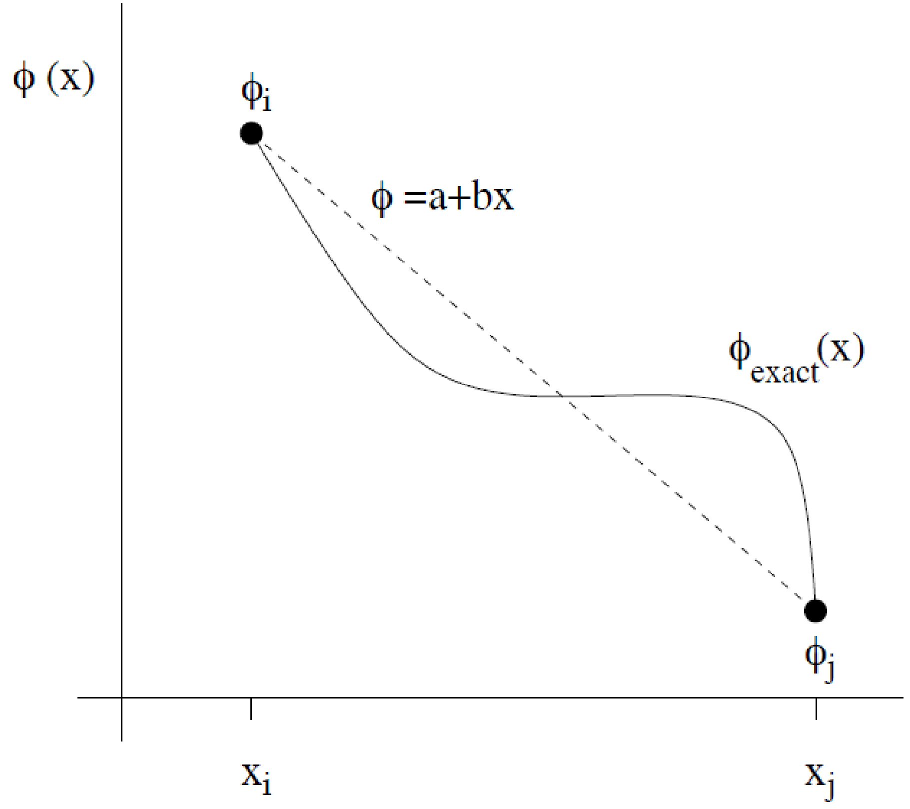
$$\emptyset(x) = a + bx = (1\ x)\begin{pmatrix} a \\ b \end{pmatrix}$$

- Coefficients a and b

$$\emptyset_i = a + bx_i$$

$$\emptyset_j = a + bx_j$$

- $\emptyset = \boldsymbol{Aa}$



1D Simplex

- $\emptyset = \boldsymbol{A}\boldsymbol{a}$

$$\emptyset = \begin{pmatrix} \emptyset_i \\ \emptyset_j \end{pmatrix}, \boldsymbol{A} = \begin{pmatrix} 1 & x_i \\ 1 & x_j \end{pmatrix}, \boldsymbol{a} = \begin{pmatrix} a \\ b \end{pmatrix}$$

$$\mathbf{a} = \mathbf{A}^{-1}\phi = \frac{1}{x_j - x_i}\begin{pmatrix} x_j & -x_i \\ -1 & 1 \end{pmatrix}\begin{pmatrix} \phi_i \\ \phi_j \end{pmatrix} = \frac{1}{l}\begin{pmatrix} x_j & -x_i \\ -1 & 1 \end{pmatrix}\begin{pmatrix} \phi_i \\ \phi_j \end{pmatrix}$$

- $l = x_j - x_i, \emptyset = (1\ x)\boldsymbol{a}$

$$\phi = \frac{1}{l}(1 \quad x)\begin{pmatrix} x_j & -x_i \\ -1 & 1 \end{pmatrix}\begin{pmatrix} \phi_i \\ \phi_j \end{pmatrix}$$

- $\emptyset(x) = N_i(x)\ \emptyset_i + N_j(x)\ \emptyset_j$

1D Simplex

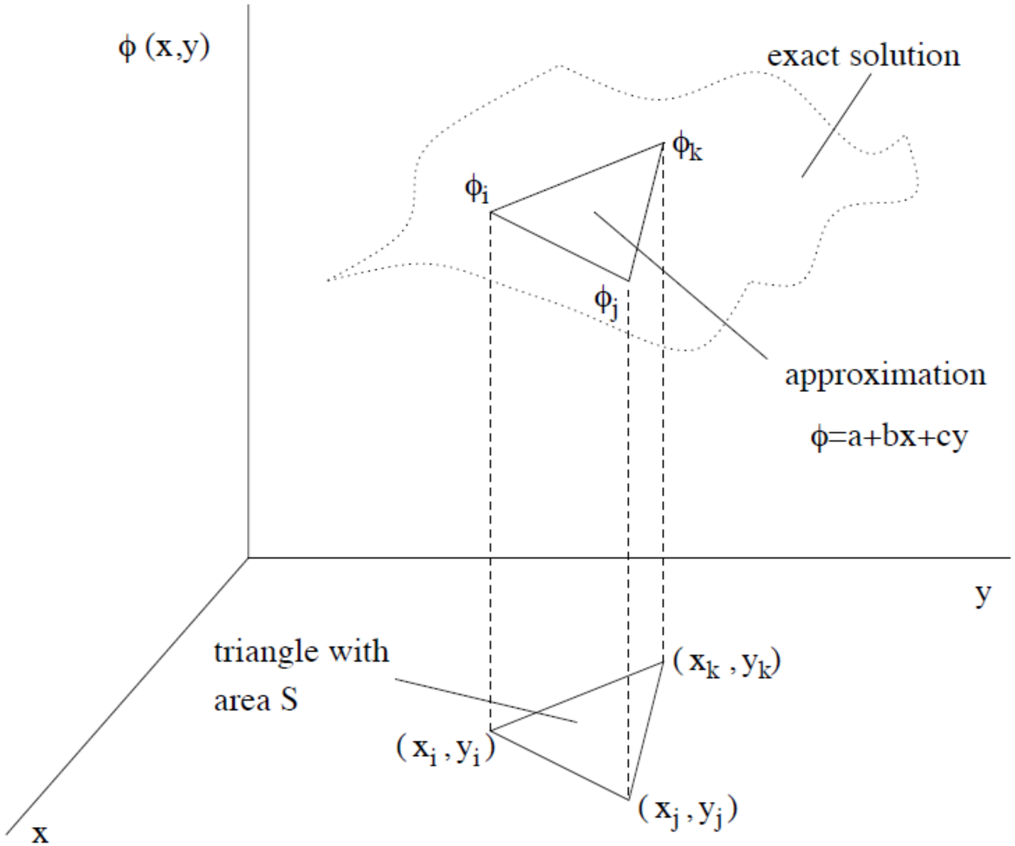
- $\emptyset(x) = N_i(x)\,\emptyset_i + N_j(x)\,\emptyset_j$
- $N_i(x)$ and $N_j(x)$ are Lagrange Polynomial coefficients (shape functions)

$$N_i(x) = \frac{1}{l}(x_j - x)$$
$$N_j(x) = \frac{1}{l}(x - x_i)$$

- $N_i(x_i) = 1, N_i\left(x_j\right) = 0$
- $N_j(x_i) = 0, N_j\left(x_j\right) = 1$

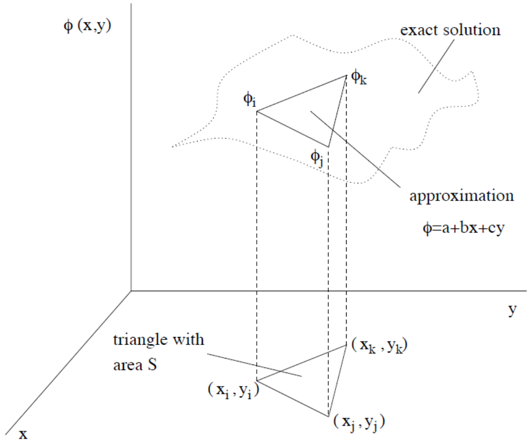
2D Simplex

- Approximate the solution over a region with a plane
  $\emptyset(x) = a + bx + cy$



2D Simplex

- Approximate the solution over a region with a plane
  $\emptyset(x, y) = a + bx + cy$
  - $\emptyset_i = a + bx_i + cy_i$
  - $\emptyset_j = a + bx_j + cy_j$
  - $\emptyset_k = a + bx_k + cy_k$



$$\phi = \mathbf{A}\mathbf{a} \rightarrow \phi = \begin{pmatrix} \phi_i \\ \phi_j \\ \phi_k \end{pmatrix}, \mathbf{A} = \begin{pmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{pmatrix}, \mathbf{a} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$\mathbf{a} = \mathbf{A}^{-1}\phi = \frac{1}{2S}\begin{pmatrix} x_j y_k - x_k y_j & x_k y_i - x_i y_k & x_i y_j - x_j y_i \\ y_j - y_k & y_k - y_i & y_i - y_j \\ x_k - x_j & x_i - x_k & x_j - x_i \end{pmatrix}\begin{pmatrix} \phi_i \\ \phi_j \\ \phi_k \end{pmatrix}$$

2D Simplex

- $\emptyset = \boldsymbol{Aa}$

$$\phi(x) = N_i(x,y)\phi_i + N_j(x,y)\phi_j + N_k(x,y)\phi_k$$

$$N_i(x,y) = \frac{1}{2S}[(x_j y_k - x_k y_j) + (y_j - y_k)x + (x_k - x_j)y]$$
$$N_j(x,y) = \frac{1}{2S}[(x_k y_i - x_i y_k) + (y_k - y_i)x + (x_i - x_k)y]$$
$$N_k(x,y) = \frac{1}{2S}[(x_i y_j - x_j y_i) + (y_i - y_j)x + (x_j - x_i)y]$$

- $N_i(x_i, y_i) = 1, N_i(x_j, y_j) = N_i(x_k, y_k) = 0$
- $N_j(x_j, y_j) = 1, N_j(x_i, y_i) = N_j(x_k, y_k) = 0$
- $N_k(x_k, y_k) = 1, N_k(x_i, y_i) = N_k(x_j, y_j) = 0$

Problems of Simplex Elements

- 1D Simplex $\emptyset(x) = a + bx$
- 2D Simplex $\emptyset(x, y) = a + bx + cy$
- Second or higher order PDE
  - $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$
  - $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$
- Solutions
  - Complex elements
  - Weak formulation

The Variational Principle

- Weak formulation - The integral formulation
  - Linear interpolation pieces whose derivatives don't necessarily match across elements
- Express the PDE as an integral which is to be minimized
  - Differential form → Integral form

$$I(\phi) = \iiint_V F\left(\phi, \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z}\right) dV + \iint_S g\left(\phi, \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z}\right) dS$$

  - F is the PDE, and g is the boundary condition

The Variational Principle

- Euler-Lagrange equations for minimizing variations

$$I(\phi) = \iiint_V F\left(\phi, \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z}\right) dV + \iint_S g\left(\phi, \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z}\right) dS$$

$$\Updownarrow$$

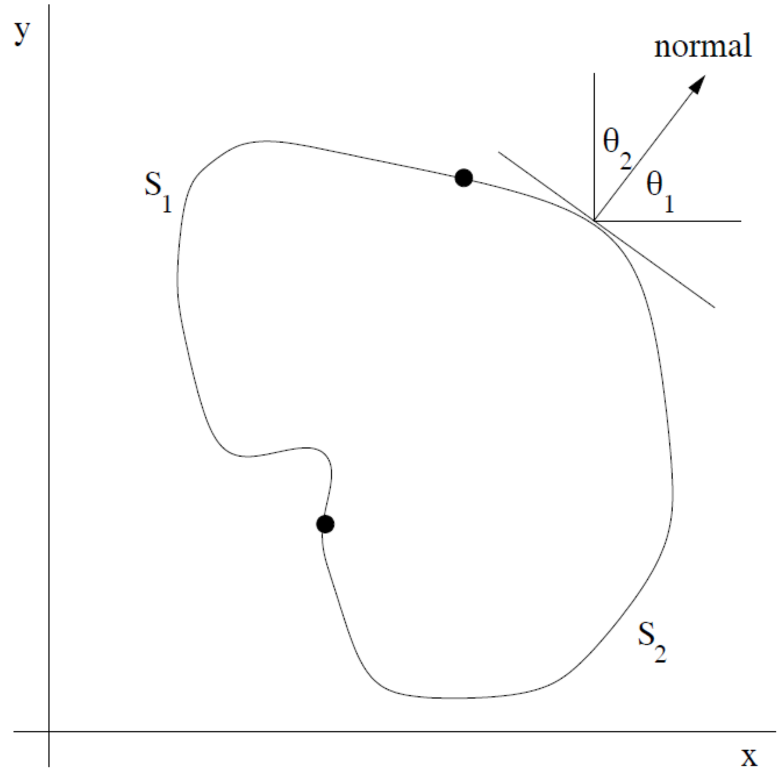
$$\frac{\delta F}{\delta \phi} = \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial(\phi_x)}\right) + \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial(\phi_y)}\right) + \frac{\partial}{\partial z}\left(\frac{\partial F}{\partial(\phi_z)}\right) - \frac{\partial F}{\partial \phi} = 0$$

The Variational Principle

$$\frac{\partial}{\partial x}\left(p(x,y)\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right)+r(x,y)u=f(x,y)$$

$$u(x,y)=g(x,y) \quad \text{on } S_1$$

$$p(x,y)\frac{\partial u}{\partial x}\cos\theta_1+q(x,y)\frac{\partial u}{\partial y}\cos\theta_2+g_1(x,y)u=g_2(x,y) \quad \text{on } S_2$$



The Variational Principle

$$\frac{\partial}{\partial x}\left(p(x,y)\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right)+r(x,y)u=f(x,y)$$

$$u(x,y)=g(x,y)\quad \text{on } S_1$$

$$p(x,y)\frac{\partial u}{\partial x}\cos\theta_1+q(x,y)\frac{\partial u}{\partial y}\cos\theta_2+g_1(x,y)u=g_2(x,y)\quad \text{on } S_2$$

$$I(\phi)=\iiint_V F\left(\phi,\frac{\partial \phi}{\partial x},\frac{\partial \phi}{\partial y},\frac{\partial \phi}{\partial z}\right)dV+\iint_S g\left(\phi,\frac{\partial \phi}{\partial x},\frac{\partial \phi}{\partial y},\frac{\partial \phi}{\partial z}\right)dS$$

$$I(u)=\frac{1}{2}\iint_D dxdy\left[p(x,y)\left(\frac{\partial u}{\partial x}\right)^2+q(x,y)\left(\frac{\partial u}{\partial y}\right)^2-r(x,y)u^2+2f(x,y)u\right]$$

$$+\int_{S_2} dS\left[-g_2(x,y)u+\frac{1}{2}g_1(x,y)u^2\right]$$

$$I_D=\frac{1}{2}\left[p(x,y)\left(\frac{\partial u}{\partial x}\right)^2+q(x,y)\left(\frac{\partial u}{\partial y}\right)^2-r(x,y)u^2+2f(x,y)u\right]$$

注意：上述边界条件仅基于S_2，实际上需要基于S_1和S_2的边界条件

The Variational Principle

- Variational calculus as the minimization over the functional space of interest

$$I_D = \frac{1}{2}\left[p(x,y)\left(\frac{\partial u}{\partial x}\right)^2 + q(x,y)\left(\frac{\partial u}{\partial y}\right)^2 - r(x,y)u^2 + 2f(x,y)u\right]$$

$$\frac{\delta F}{\delta \phi} = \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial(\phi_x)}\right) + \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial(\phi_y)}\right) + \frac{\partial}{\partial z}\left(\frac{\partial F}{\partial(\phi_z)}\right) - \frac{\partial F}{\partial \phi} = 0$$

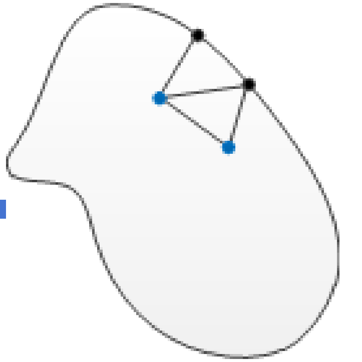
$$\begin{aligned}
\frac{\delta I_D}{\delta u} &= \boxed{\frac{\partial}{\partial x}\left(\frac{\partial I_D}{\partial(u_x)}\right)} + \boxed{\frac{\partial}{\partial y}\left(\frac{\partial I_D}{\partial(u_y)}\right)} - \frac{\partial I_D}{\partial u} = 0 \\
&= \frac{\partial}{\partial x}\left(\frac{1}{2}p(x,y)\cdot 2\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{2}q(x,y)\cdot 2\frac{\partial u}{\partial y}\right) - \left(-\frac{1}{2}r(x,y)\cdot 2u + f(x,y)\right) \\
&= \frac{\partial}{\partial x}\left(p(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right) + r(x,y)u - f(x,y) = 0
\end{aligned}$$

$$\boxed{\frac{\partial}{\partial x}\left(p(x,y)\frac{\partial u}{\partial x}\right)} + \boxed{\frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right)} + r(x,y)u = f(x,y)$$

The Variational Principle

- Differential form → integral form
- The solution in the weak (integral) form
  - The linear (simplex) interpolating functions give that the second derivatives (∂^2) are zero
  - The elliptic operator would have no contribution from a finite element of the form $\emptyset = a + bx + cy$
  - In the integral formulation, the second derivative terms are proportional to $(\partial x)^2 + (\partial y)^2$, which give $\emptyset = b^2 + c^2$

The Variational Principle



- A solution is sought which takes the form

$$u(x,y) = \sum_{i=1}^{m} \gamma_i \phi_i(x,y)$$

  - $\emptyset_i(x,y)$ are linearly independent piecewise-linear polynomials, and γ_i are constants
  - $\gamma_1, .. \gamma_n$ ensure that the integrand $I(u)$ in the interior of the computational domain is minimized, i.e.$\frac{\partial x}{\partial \gamma_i} = 0$
  - $\gamma_{n+1}, .. \gamma_m$ ensure that the boundary conditions are satisfied, i.e. those elements which touch the boundary must meet certain restrictions

Solution Method

$$u(x,y)=\sum_{i=1}^{m}\gamma_i\phi_i(x,y)$$

$$I(u)=\frac{1}{2}\iint_D dxdy\left[p(x,y)\left(\frac{\partial u}{\partial x}\right)^2+q(x,y)\left(\frac{\partial u}{\partial y}\right)^2-r(x,y)u^2+2f(x,y)u\right]$$
$$+\int_{S_2}dS\left[-g_2(x,y)u+\frac{1}{2}g_1(x,y)u^2\right]$$

$$\begin{aligned}
I(u) &= I\left(\sum_{i=1}^{m}\gamma_i\phi_i(x,y)\right)\\
&= \frac{1}{2}\iint_D dxdy\left[p(x,y)\left(\sum_{i=1}^{m}\gamma_i\frac{\partial\phi_i}{\partial x}\right)^2+q(x,y)\left(\sum_{i=1}^{m}\gamma_i\frac{\partial\phi_i}{\partial y}\right)^2\right.\\
&\qquad\left.-r(x,y)\left(\sum_{i=1}^{m}\gamma_i\phi_i(x,y)\right)^2+2f(x,y)\left(\sum_{i=1}^{m}\gamma_i\phi_i(x,y)\right)\right]\\
&\quad+\int_{S_2}dS\left[-g_2(x,y)\sum_{i=1}^{m}\gamma_i\phi_i(x,y)+\frac{1}{2}g_1(x,y)\left(\sum_{i=1}^{m}\gamma_i\phi_i(x,y)\right)^2\right]
\end{aligned}$$

Solution Method

- $I(u)$ are differentiated with respect to all the interior points and minimized

$$
\begin{aligned}
\frac{\partial I}{\partial \gamma_j} \;=\; & \iint_D dxdy\left[p(x,y)\sum_{i=1}^{m}\gamma_i\frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x}+q(x,y)\sum_{i=1}^{m}\gamma_i\frac{\partial \phi_i}{\partial y}\frac{\partial \phi_j}{\partial y}\right. \\
& \left. -r(x,y)\sum_{i=1}^{m}\gamma_i\phi_i\phi_j+f(x,y)\phi_j\right] \\
& +\int_{S_2} dS\left[-g_2(x,y)\phi_j+g_1(x,y)\sum_{i=1}^{m}\gamma_i\phi_i\phi_j\right]=0
\end{aligned}
$$

$$
\begin{aligned}
& \sum_{i=1}^{m}\left\{\iint_D dxdy\left[p\frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x}+q\frac{\partial \phi_i}{\partial y}\frac{\partial \phi_j}{\partial y}-r\phi_i\phi_j\right]+\int_{S_2} dS g_1\phi_i\phi_j\right\}\gamma_i \\
& +\iint_D dxdy f\phi_j-\int_{S_2} dS g_2(x,y)\phi_j=0\,.
\end{aligned}
$$

Solution Method

- $Ax = b$

$$\mathbf{x} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_n \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix} \quad \mathbf{A} = (\alpha_{ij})$$

$$\beta_i = -\iint_D dxdy\, f\phi_i + \int_{S_2} dS g_2 \phi_i - \boxed{\sum_{k=n+1}^{m} \alpha_{ik}\gamma_k}$$

$$\alpha_{ij} = \iint_D dxdy \left[p\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x} + q\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y} - r\phi_i\phi_j \right] + \int_{S_2} dS g_1 \phi_i \phi_j$$

$$\phi_i = \sum_{i=1}^{3} N_j^{(i)}(x,y)\phi_j^{(i)} = \sum_{i=1}^{3} \left(a_j^{(i)} + b_j^{(i)} x + c_j^{(i)} y \right) \phi_j^{(i)}$$

$$\sum_{i=1}^{m} \left\{ \iint_D dxdy \left[p\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x} + q\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y} - r\phi_i\phi_j \right] + \int_{S_2} dS g_1 \phi_i \phi_j \right\} \gamma_i$$
$$+ \iint_D dxdy\, f\phi_j - \int_{S_2} dS g_2(x,y)\phi_j = 0\,.$$

Finite Element Methods

- Discretize the computational domain into triangles T_1, T_2, …, T_k are the interior triangles and T_{k+1}, T_{k+2}, …, T_m are triangles that have at least one edge which touches the boundary
- For $l = k+1, k+2, \ldots, m$, determine the values of the vertices on the triangles which touch the boundary

Finite Element Methods

- Generate the shape functions $N_j^{(i)} = a_j^{(i)} + b_j^{(i)}x + c_j^{(i)}y$
- Compute the integrals for matrix elements α_{ij} and vector elements β_j in the interior and boundary
- Construct the matrix A and b
- Solve $Ax = b$
- Plot the solution $u(x, y) = \sum_{i=1}^{m} \gamma_i \emptyset_i(x, y)$

Outline

- Overview
- Ordinary differential equations (ODE)
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
  - Finite elements (有限元法)
    - Basic method
    - Calculus of variations

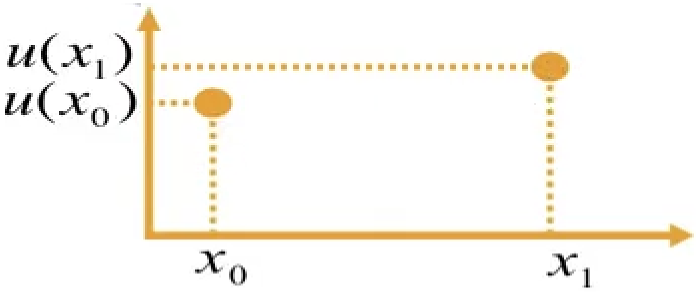
Calculus of Variations

- Generalization of Calculus that seeks to find the path, curve, surface, etc., for which a given Functional has a minimum or maximum
- Goal: find extrema values of integrals of the form $\int F(u, u_x) dx$
- It has an extremum only if the Euler-Lagrange Differential Equation is satisfied

$$\left(\frac{\partial}{\partial u} - \frac{d}{dx} \frac{\partial}{\partial u_x} \right) F(u, u_x) = 0$$

Calculus of Variations

- Example: Find the shape of the curve {x, u(x)} with shortest length $\int_{x_0}^{x_1} \sqrt{1+u_x^2}\,dx$ given u(x0), u(x1)



- Solution: A differential equation that u(x) must satisfy

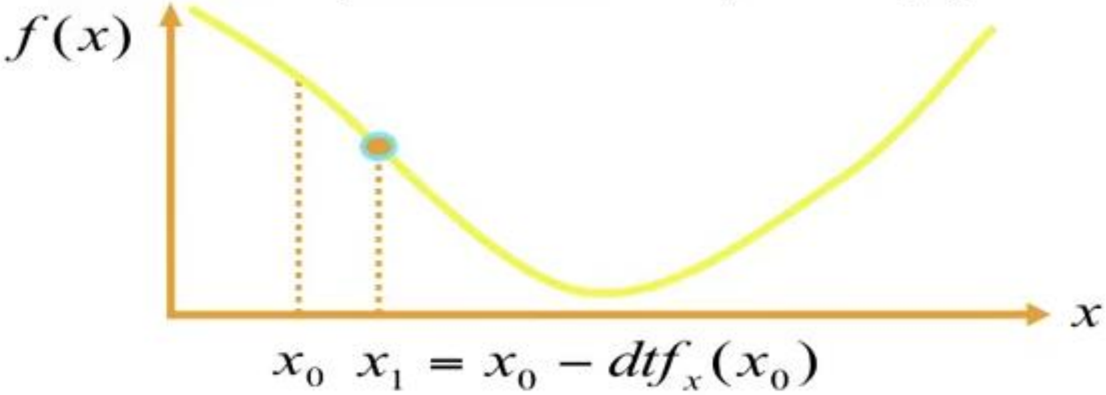
$$\left(\frac{\partial}{\partial u} - \frac{d}{dx}\frac{\partial}{\partial u_x}\right)F(u,u_x) = 0$$

$$\frac{u_{xx}}{\left(1+u_x^2\right)^{3/2}} = 0 \quad \Rightarrow \quad u_x = a \quad \Rightarrow \quad u(x) = ax + b$$

Extrema Points in Calculus

$$\forall \eta : \lim_{\varepsilon \to 0} \left(\frac{df(x + \varepsilon\eta)}{d\varepsilon} \right) = 0 \Leftrightarrow \forall \eta : f_x(x)\eta = 0 \Leftrightarrow f_x(x) = 0$$

Gradient descent process $x_t = -f_x$



Calculus of Variations

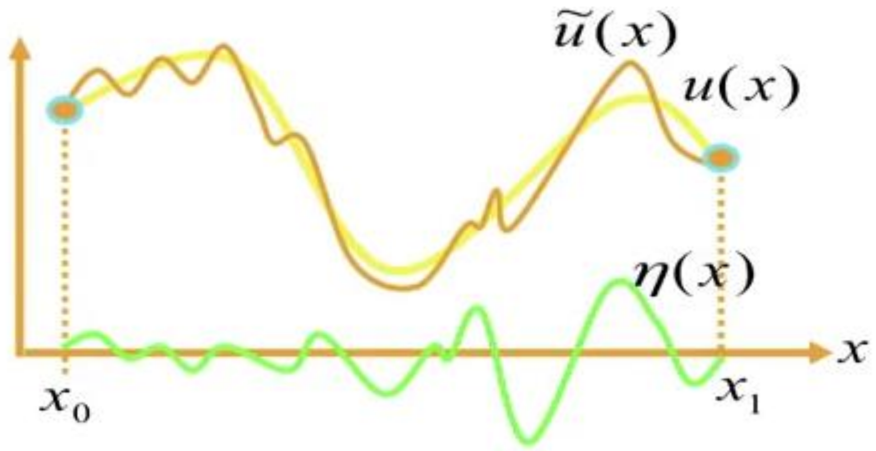
$$E(u(x)) = \int F(u, u_x)\,dx$$

$$\tilde{u}(x) = u(x) + \varepsilon\eta(x)$$

$$\forall \eta(x) : \lim_{\varepsilon \to 0}\left(\frac{d}{d\varepsilon}\int F(\tilde{u}, \tilde{u}_x)\,dx\right) \stackrel{?}{=} 0$$

$$\Downarrow$$

$$\frac{\delta E(u)}{\delta u} = \left(\frac{\partial}{\partial u} - \frac{d}{dx}\frac{\partial}{\partial u_x}\right)F(u, u_x)$$



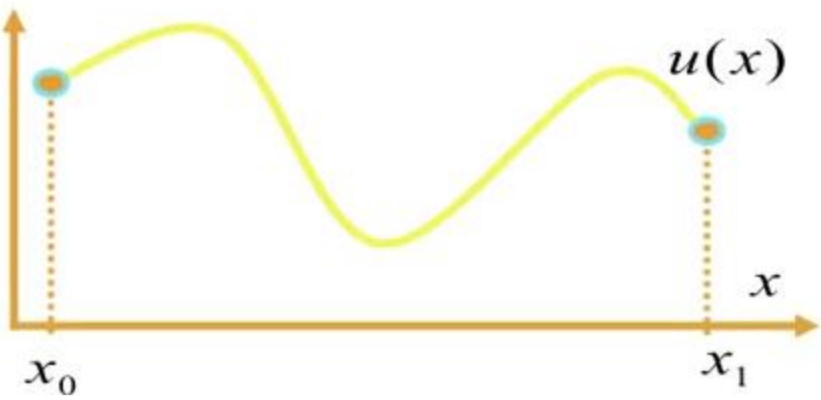
- Gradient descent process $u_t = -\dfrac{\delta E(u)}{\delta u}$

Calculus of Variations

- Gradient descent process

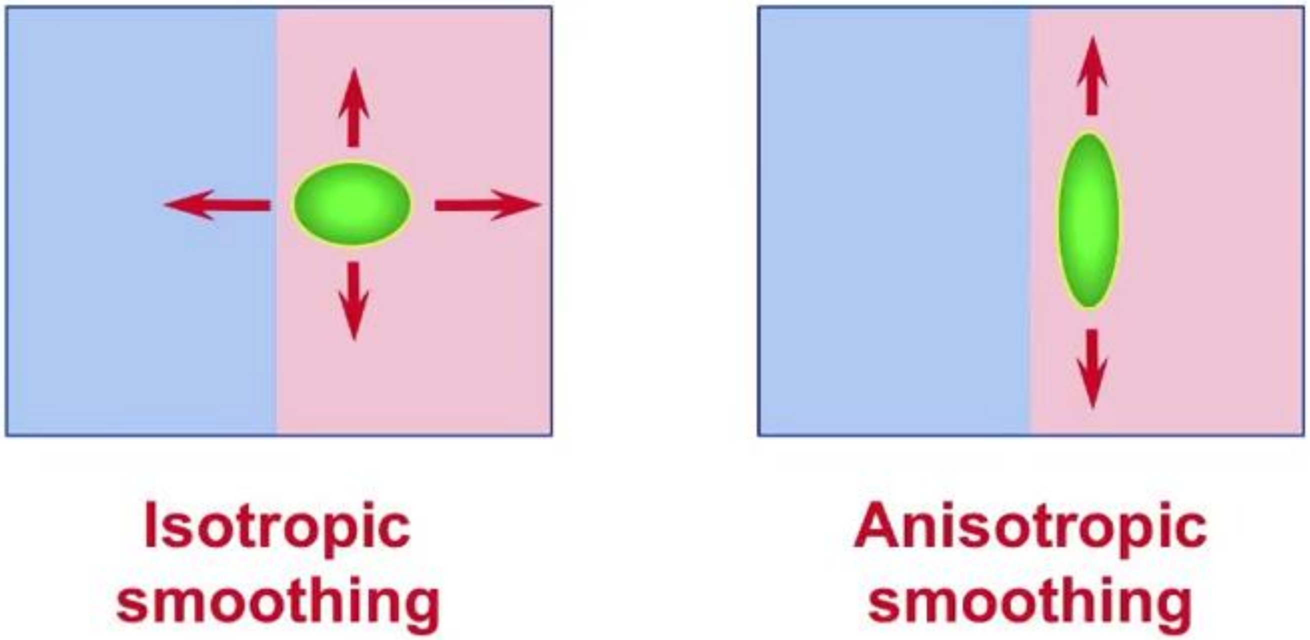
Calculus $\arg\min_{x} f(x) \Longrightarrow x_t = -f_x$

Calculus of variations $\arg\min_{u(x)} \int F(u, u_x)dx \Longrightarrow u_t = -\dfrac{\delta E(u)}{\delta u}$

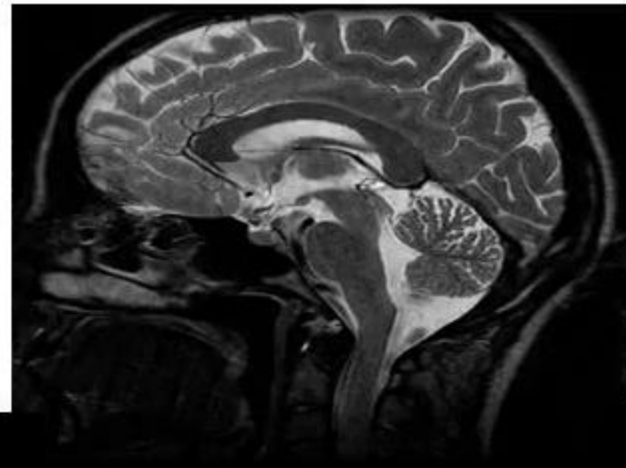


Anisotropic Diffusion

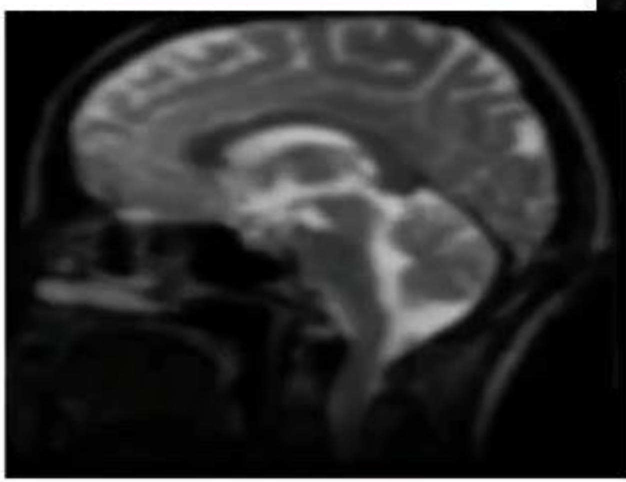
- Isotropic vs. Anisotropic smoothing



Anisotropic Diffusion

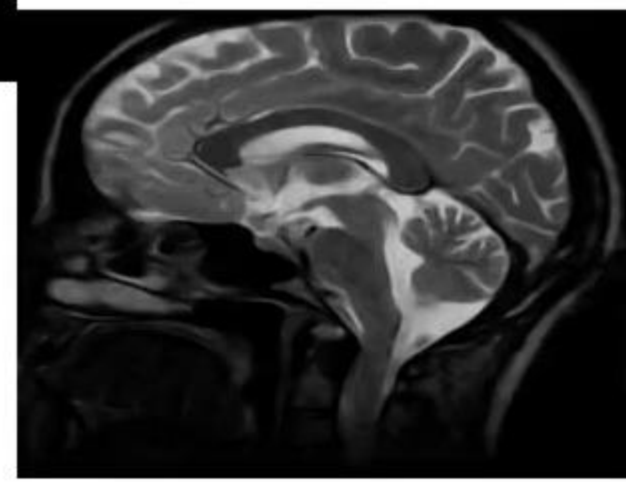


Isotropic
(Heat equation)



$$\frac{\partial I(x,y,t)}{\partial t} = \Delta I$$

Anisotropic



$$\frac{\partial I(x,y,t)}{\partial t} = div(g(|\nabla I|)\nabla I)$$

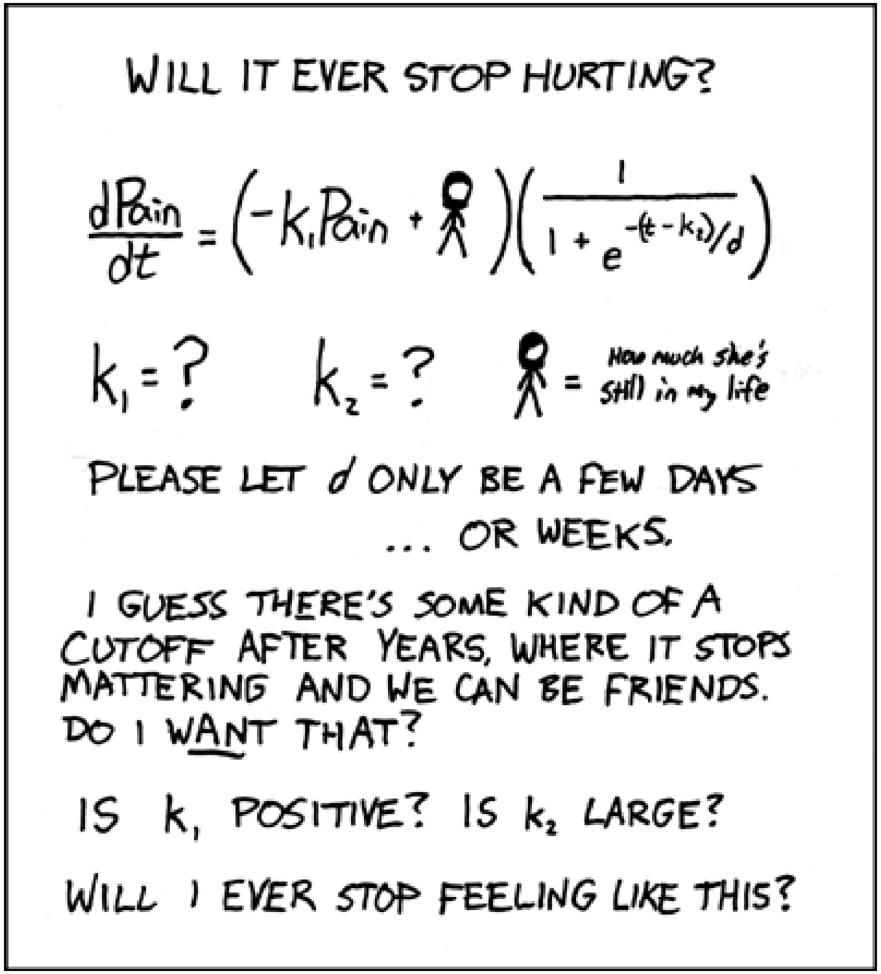
Anisotropic Diffusion

- $min_I \int_\Omega \rho(|\nabla I|)d\Omega \rightarrow \frac{\partial I(x,y,t)}{\partial t} = div(\rho' \frac{\nabla I}{|\nabla I|})$
- $\rho(a) = a^2 \rightarrow I_t = div\left(2|\nabla I| \frac{\nabla I}{|\nabla I|}\right) = 2\mathrm{div}(\nabla I)$
  - $\int |\nabla I|^2$
  - $I_t = 2\Delta I$, isotropic diffusion
- $\rho(a) = a \rightarrow I_t = div\left(\frac{\nabla I}{|\nabla I|}\right)$
  - Anisotropic diffusion
  - Total variation $\int |\nabla I|$
  - Relation with curve evolution, level sets

Summary

- Overview
- Ordinary differential equations (ODE)
  - Initial value problem (初值问题)
    - Euler, Runge-Kutta, Adams methods
  - Error analysis (误差分析)
    - Discretization error, Round-off, Stability
  - Boundary value problem (边值问题)
    - Shooting method, direct solve, relaxation
- Partial differential equations (PDE)
  - Finite differences (有限差分法)
  - Spectral methods (频谱法)
  - Finite elements (有限元法)

xkcd 128: dPain over dt



Title text: You laugh to keep from crying, you do math to keep from crying ...