Discrete Wavelet transform.

DWT

Ingrid Daubechies.

Daubechies wavelet basis.

DAUB-4 is the simplest.

DAUB-4 filter coefficients : (c_0, c_1, c_2, c_3).

Transformation matrix:

$$\begin{pmatrix} c_0 & c_1 & c_2 & c_3 & 0 & 0 & & & \\ c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & & \bigcirc & \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & & & \\ 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & & & \\ & & & & c_0 & c_1 & c_2 & c_3 & \\ & & & & c_3 & -c_2 & c_1 & -c_0 & \\ & \bigcirc & & & & & \ddots & & \\ & & & & & & & & \end{pmatrix}$$

Smoothening function h.

$= \overleftrightarrow{C}^0$

picks out information that is lost in smoothening process.

$$h(t) \rightarrow h(t_k) = h_k$$

$$\vec{h} = \begin{Bmatrix} \vdots \\ h_k \\ \vdots \end{Bmatrix}$$

Smoothening:

$$c_0 h_k + c_1 h_{k+1} + c_2 h_{k+2} + c_3 h_{k+3}$$

Retaining lost information.

$$c_3 h_k - c_2 h_{k+1} + c_1 h_{k+2} - c_0 h_{k+3}$$

gives zero if function already has been smoothened.

Say:

$$\vec{h} = \begin{Bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{Bmatrix}$$

$$\overleftrightarrow{C} \circ \vec{h} = \begin{pmatrix} s_1' \\ d_1' \\ s_2' \\ d_2' \\ s_3' \\ d_3' \\ s_4' \\ d_4' \end{pmatrix}$$

smoothened h (s_2')

details of h. (d_2')

$$s_1' = c_0 h_1 + c_1 h_2 + c_2 h_3 + c_3 h_4$$

$$d_1' = c_3 h_1 - c_2 h_2 + c_1 h_3 - c_0 h_4$$

Reorganizing this vector.

$$\begin{Bmatrix} s_1' \\ d_1' \\ s_2' \\ d_2' \\ s_3' \\ d_3' \\ s_4' \\ d_4' \end{Bmatrix} \rightarrow \begin{Bmatrix} s_1' \\ s_2' \\ s_3' \\ s_4' \\ \hline d_1' \\ d_2' \\ d_3' \\ d_4' \end{Bmatrix}$$

s_1', s_2', s_3', s_4': Smoothened h $= \vec{h}^1$

d_1', d_2', d_3', d_4': details on lowest scale of h.

We now define a $\overleftrightarrow{C}^1$:

$$\begin{pmatrix} c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 \\ c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 \\ 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \overleftrightarrow{C}^1$$

$$\overleftrightarrow{C}^1 \overleftrightarrow{h}^1 = \left(\begin{array}{c} s_1^2 \\ d_1^2 \\ s_2^2 \\ d_2^2 \\ \hline d_1^1 \\ d_2^1 \\ d_3^1 \\ d_4^1 \end{array}\right) \xrightarrow[\text{Rearranging}]{} \left(\begin{array}{c} s_1^2 \\ s_2^2 \\ \hline d_1^2 \\ d_2^2 \\ \hline d_1^1 \\ d_2^1 \\ d_3^1 \\ d_4^1 \end{array}\right)$$

Before we proceed, let us find values for $c_0, \ldots, c_3$.

We wish to be able to reconstruct h_k from $s_1', s_2', \ldots, s_{N/2}', d_1', \ldots, d_{N/2}'$.

(If we can do this for s_j' and d_j', we can do it for any level s_j^i, d_j^i).

We will <u>demand</u> that $\overleftrightarrow{C}^0$ is orthogonal : $\boxed{\overleftrightarrow{C}^{0\,T} \cdot \overleftrightarrow{C}^0 = 1.}$

$$\Downarrow$$

$$\boxed{\begin{array}{ll} (1) & c_0^2 + c_1^2 + c_2^2 + c_3^2 = 1 \\ (2) & c_2 c_0 + c_3 c_1 = 0 \end{array}}$$

We will <u>demand</u> that $c_3, -c_2, c_1, -c_0$ gives zero for a "smooth image" function.

"Smooth enough": h is a polynomial of order $p = 0, 1$:

$h = \text{const} \leftarrow p = 0 \quad \rightarrow h_k = 0$

$h = a t \leftarrow p = 1 \quad \rightarrow h_k = a k.$

$\Rightarrow$

$$\boxed{\begin{array}{ll} (3) & c_3 - c_2 + c_1 - c_0 = 0 \\ (4) & 0 \cdot c_3 - 1 \cdot c_2 + 2 \cdot c_1 - 3 c_0 = 0 \end{array}}$$

Solving eqs. (1) – (4):

$$\boxed{\begin{aligned} c_0 &= (1+\sqrt{3})/4\sqrt{2} \\ c_1 &= (3+\sqrt{3})/4\sqrt{2} \\ c_2 &= (3-\sqrt{3})/4\sqrt{2} \\ c_3 &= (1-\sqrt{3})/4\sqrt{2} \end{aligned}}$$

Back to our example:

$$\vec{h} = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{pmatrix} \rightarrow \vec{h}^2 = \begin{pmatrix} s_1^2 \\ s_2^2 \\ \hline d_1^2 \\ d_2^2 \\ \hline d_1^1 \\ d_2^1 \\ d_3^1 \\ d_4^1 \end{pmatrix}$$

These are the wavelet coefficients.

We cannot smoothen this further: any function on the form $h(t) = at + b$ will give zero d_k^j so the information on a and b must be kept in s_k^j at the level where only two remain ($s_k^{\frac{N}{2}}$).

Data filtering:

Remove d_j^i coefficients that correspond to structure we want removed.

Data compression.

Remove all d_j^i that has $|d_j^i| < \delta$. It is typically possible to remove a lot!

Wavelet acceleration of iterative methods for solving linear equation systems:

$$Ax = b$$

W is wavelet transform.

$$\tilde{A} = W A W^{-1}$$

$$\tilde{b} = W b$$

$\tilde{A}\tilde{x} = \tilde{b}$ ← solve this

$\downarrow$

$$\underline{x = W^{-1}\tilde{x}.}$$

Matrices with near-singular diagonal and "nice" behavior away from the diagonal:

$$A_{ij} = \frac{1}{\epsilon + |i-j|}$$; works well with wavelets.

<u>Power spectrum in wavelet basis:</u>

d^{i}_{j} ← scale (for i), ← position (for j)

Average over all positions.

$$\frac{1}{N_i} \sum_{j=1}^{N_i} |d^i_j|^2 = \Omega_i$$

Plot Ω_i vs. 2^i

} This is the equivalent of the power spectrum.

<u>Very</u> good for identifying scale-invariant information.

Quantum Mechanics

Path integral Monte Carlo

Schrödinger equation:

$$i\hbar \frac{\partial}{\partial t} \psi = H \psi$$

$$H = -\frac{\hbar^2}{2m} \nabla^2 + V$$

We introduce the propagator:

$(t > 0)$

$$\psi(x,t) = \int dy \underbrace{K(x,t;y,0)}_{} \psi(y,0)$$

We wish to find an expression for K.

R. P. Feynman, Rev. Mod. Phys. 20, 367 (1948).

$$t > t' > 0$$

$$\psi(z, t') = \int dy \; K(z, t'; y, 0) \, \psi(y, 0)$$

$$\psi(x, t) = \int dz \; K(x, t; z, t') \, \psi(z, t')$$

$$= \int dy \, dz \; K(x, t; z, t') \, K(z, t'; y, 0) \, \psi(y, 0)$$

$$\Rightarrow \quad \boxed{K(x, t; y, 0) = \int dz \; K(x, t; z, t') K(z, t'; y, 0)}$$

We repeat this construction:

$$t_0 < t_1 < \dots < t_N \quad ; \quad t_k = t_0 + k \Delta t$$

$$K(x_N; t_N; x_0, t_0) = \int \dots \int dx_1 \dots dx_{N-1}$$

$$K(x_N, t_N; x_{N-1}, t_{N-1}) \dots K(x_1, t_1; x_0, t_0)$$

We now assume Δt to be small:

$$i\hbar \frac{\partial}{\partial t} \psi(x,t) = H \psi(x,t)$$

$$= (T + V)\, \psi(x,t)$$

kinetic energy operator

potential energy operator.

$\Rightarrow$

$$\psi(x,t) = \psi(x, t-\Delta t) - \frac{i\Delta t}{\hbar} H \psi(x, t-\Delta t)$$

$$= \left(1 - \frac{i\Delta t}{\hbar} H\right) \psi(x, t-\Delta t)$$

$$\underline{\psi(x,t)} = \int dy \left(1 - \frac{i\Delta t}{\hbar} H\right) \delta(x-y)\, \psi(y, t-\Delta t)$$

$$= \int dy \left(1 - \frac{i\Delta t}{\hbar} H\right) \left\{ \sum_n \psi_n^0(x, t-\Delta t)\, \psi_n^{0\dagger}(y, t-\Delta t) \right\} \psi(y, t-\Delta t)$$

$$T \psi_n^0(x,t) = E_n^0 \psi_n^0(x,t)$$

Free particle wave functions.

$$\underline{\psi(x,t)} = \int dy \left(1 - \frac{i\Delta t}{\hbar} T - \frac{i\Delta t}{\hbar} V\right)$$

$$\sum_n{}' \psi_n^0(x, t-\Delta t)\, \psi_n^{0+}(y, t-\Delta t)\, \psi(y, t-\Delta t)$$

$$= \int dy \sum_n{}' \left(1 - \frac{i\Delta t}{\hbar} E_n^0 - \frac{i\Delta t}{\hbar} V\right)$$

$$\psi_n^0(x,t)\, e^{\frac{i\Delta t}{\hbar} E_n^0}\, \psi_n^{0+}(y, t-\Delta t)$$

$$\psi(y, t-\Delta t)$$

$$= \int dy \sum_n{}' \left(1 - \frac{i\Delta t}{\hbar} E_n^0 - \frac{i\Delta t}{\hbar} V\right)\left(1 + \frac{i\Delta t}{\hbar} E_n^0\right)$$

$$\psi_n^0(x,t)\, \psi_n^{0+}(y, t-\Delta t)\, \psi(y, t-\Delta t)$$

$$= \int dy \sum_n{}' \psi_n^0(x,t) \left(1 - \frac{i\Delta t}{\hbar} V\right)$$

$$\psi_n^{0+}(y, t-\Delta t)\, \psi(y, t-\Delta t)$$

$$= \underline{\int dy \sum_n{}' \psi_n^0(x,t)\, \psi_n^{0+}(y, t-\Delta t)\, e^{-\frac{i\Delta t}{\hbar} V}}$$

$$\underline{\psi(y, t-\Delta t)}$$

$$\underline{K(x,t;\, y,\, t-\Delta t)}$$

$$= \underline{\sum_n{}' \psi_n^0(x,t)\, \psi_n^{0+}(y, t-\Delta t)\, e^{-\frac{i\Delta t}{\hbar} V}}$$

$$\psi_n^0(x,t) = \frac{1}{\sqrt{L}}\, e^{i k_n x - \frac{i}{\hbar} E_n^0 t}$$

$$k_n = \frac{2\pi n}{L}$$

$$E_n^0 = \frac{\hbar^2}{2m} k_n^2$$

$$\underline{\sum_n{}' \psi_n^0(x,t)\, \psi_n^{0+}(y,t')}$$

$$= \sum_n{}' \frac{1}{L}\, e^{2\pi i \frac{n}{L}(x-y) - i \frac{\pi^2 \hbar}{m} \frac{n^2}{L^2}(t-t')}$$

$$\longrightarrow 2\pi \int_{-\infty}^{+\infty} dp\; e^{2\pi i p (x-y) - i \frac{\pi^2\hbar}{m} p^2 (t-t')}$$

$$\underline{= \left(\frac{m}{2\pi i \hbar (t-t')}\right)^{1/2} e^{\frac{im}{2\hbar}\left(\frac{x-y}{t-t'}\right)^2 (t-t')}}$$

Hence,

$$\underline{K(x,t;y,t-\Delta t)}$$

$$\underline{= \left(\frac{m}{2\pi i \hbar \Delta t}\right)^{1/2} e^{\frac{im}{2\hbar}\left(\frac{x-y}{\Delta t}\right)^2 \Delta t - \frac{i}{\hbar} V(x)\Delta t}}$$

The phase may be written

$$\frac{i}{\hbar} \Delta t \left\{ \frac{m}{2} \left(\frac{x-y}{\Delta t} \right)^2 - V(x) \right\} = S(x,t\,;\,y,t-\Delta t)$$

This is the <u>classical action</u>.

We may now use the product rule on page 316 to calculate $K(x,t\,;\,y,t')$:

$$\begin{array}{cccc} x & t & y & t' \\ \shortparallel & \shortparallel & \shortparallel & \shortparallel \\ x_N & t_N & x_0 & t_0 \end{array}$$

$$\underline{K(x_N,t_N\,;\,x_0,t_0)}$$

$$= \int dx_1 \cdots dx_{N-1} \left(\frac{m}{2\pi i \hbar \Delta t} \right)^{\frac{N-1}{2}} e^{\frac{i}{\hbar} \sum_{k=1}^{N} \Delta t \left\{ \frac{m}{2} \left(\frac{x_k - x_{k-1}}{\Delta t} \right)^2 - V(x_k) \right\}}$$

Feynman path integral.

$$= \underline{\int \mathcal{D}x \; e^{\frac{i}{\hbar} S(x_N,t_N\,;\,x_0,t_0)}}$$

Sum over all paths!

Action along a given path.

This is an integral in function space

$$\boxed{K(x,t;y,t') = \int \mathcal{D}x(t)\, e^{\frac{i}{\hbar}\int_{t'}^{t} dt'' L[x(t'')]}}$$

$$L[x(t)] = \frac{m}{2}\dot{x}(t)^2 - V(x(t))$$

Classical Lagrangian

In order to implement these ideas on the computer, again go to imaginary time.

We are then in reality dealing with a thermal system

The partition function:

$$Z = \sum_{\text{conf}}{}' e^{-H(\text{conf})\beta}$$

$$\beta = \frac{1}{kT}$$

$$Z_1 = \sum_n{}' \psi_n^+(x)\, e^{-H\beta}\, \psi_n(x)$$

$$= \mathrm{Tr}\, e^{-H\beta}$$

We define the <u>density matrix</u>:

$$\rho(x, x'; \beta) = \sum_n{}' \psi_n^+(x)\, e^{-H\beta}\, \psi_n(x')$$

$$Z_1 = \rho(x, x, \beta)$$

<u>Density matrix for a free particle.</u>

$$H = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2}$$

$$\psi_n(x) = \frac{1}{\sqrt{L}}\, e^{i k_n x}$$

$$k_n = \frac{2\pi n}{L}$$

$$\rho(x, x'; \beta) = \left(\frac{m}{2\pi\beta\hbar^2} \right)^{1/2} e^{-m(x-x')^2/2\beta\hbar^2}$$
