Quantum Dynamics

Aiichiro Nakano

Collaboratory for Advanced Computing & Simulations
Department of Computer Science
Department of Physics & Astronomy
Department of Chemical Engineering & Materials Science
Department of Quantitative & Computational Biology
University of Southern California

Email: anakano@usc.edu

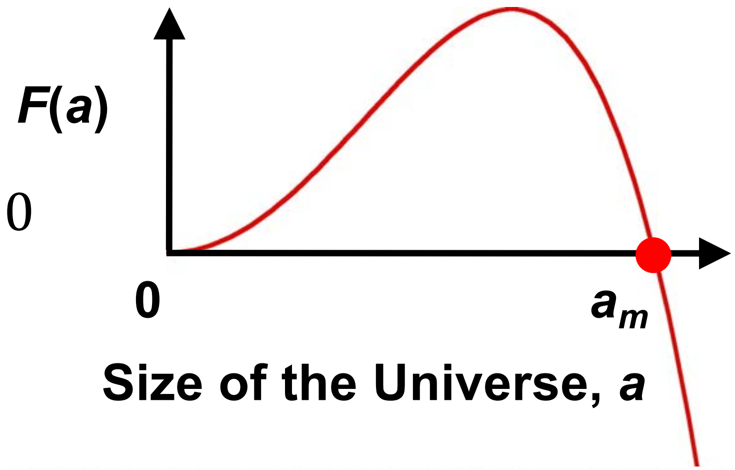
Goals:

1. **Partial differential equation**
2. **Spectral method**
   (Fourier transform)

Quantum Universe

- **Wheeler-deWitt equation**

$$\left[-\hbar^2 \frac{d^2}{da^2} + \underbrace{\left(\frac{3\pi c^3}{2G}\right)^2\left(a^2 - \frac{a^4}{a_m^2}\right)}_{F(a)}\right]\psi(a) = 0$$



CREATION OF UNIVERSES FROM NOTHING

Alexander VILENKIN
Physics Department, Tufts University, Medford, MA 02155, USA

Received 11 June 1982

A cosmological model is proposed in which the universe is created by quantum tunneling from literally nothing into a de Sitter space. After the tunneling, the model evolves along the lines of the inflationary scenario. This model does not have a big-bang singularity and does not require any initial or boundary conditions.

Phys. Lett. **117B**, 25 ('82)

IS IT POSSIBLE TO CREATE A UNIVERSE IN THE LABORATORY BY QUANTUM TUNNELING?

Edward FARHI*

Center for Theoretical Physics, Laboratory for Nuclear Science and Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Alan H. GUTH*,**

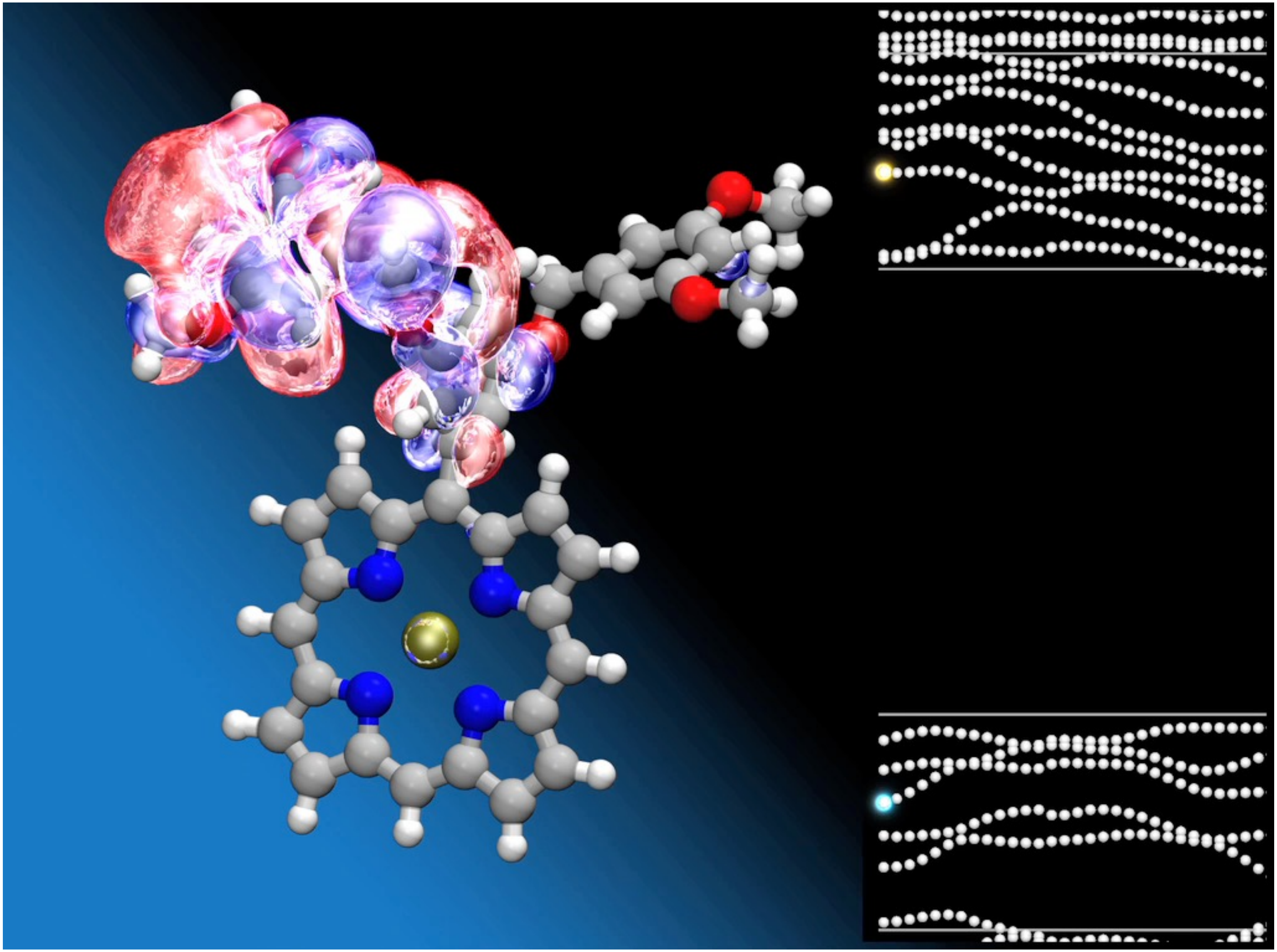
Center for Theoretical Physics, Laboratory for Nuclear Science and Department of Physics, Massachusetts Institue of Technology, Cambridge, MA 02139, USA
and
Harvard–Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138, USA

Jemal GUVEN

Instituto de Ciencias Nucleares, Universidad Nacional Autonoma de Mexico, Circuito Exterior C.U. A. Postal 70-543, 04510 Mexico D.F. Mexico

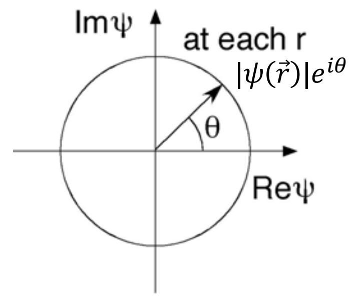
Nucl. Phys. **B339**, 417 ('90)

Photoexcited Electron Dynamics



Wave Equation

- **Complex wave function**



$$\psi(\vec{r},t) = \mathrm{Re}\psi(\vec{r},t) + i\mathrm{Im}\psi(\vec{r},t) \in \mathbb{C}\ \left(i = \sqrt{-1}\right)$$

- **Probability**

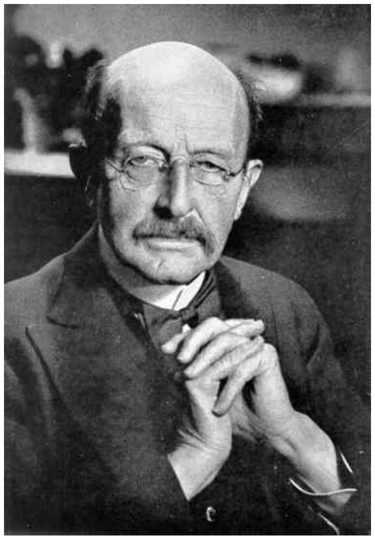
$$P(\vec{r},t) = \psi^*(\vec{r},t)\psi(\vec{r},t) = |\psi(\vec{r},t)|^2 = |\mathrm{Re}\psi(\vec{r},t)|^2 + |\mathrm{Im}\psi(\vec{r},t)|^2$$

$$\psi^*\psi = (\psi_0 - i\psi_1)(\psi_0 + i\psi_1) = \psi_0^2 + \psi_1^2 \geq 0$$

- **Normalization**

$$\int dx \int dy \int dz\, |\psi(\vec{r},t)|^2 = 1$$

- **Schrödinger (partial differential) equation**



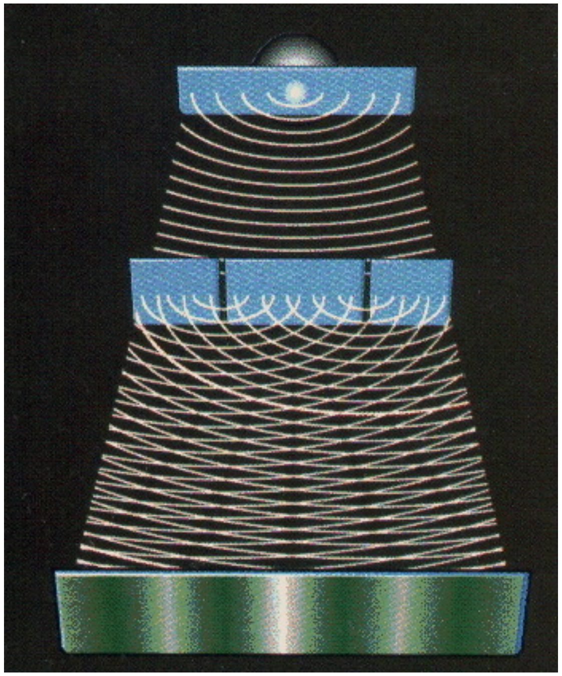
$$i\hbar \frac{\partial}{\partial t}\psi(\vec{r},t) = \left[-\frac{\hbar^2}{2m}\nabla^2 + V(\vec{r})\right]\psi(\vec{r},t)$$

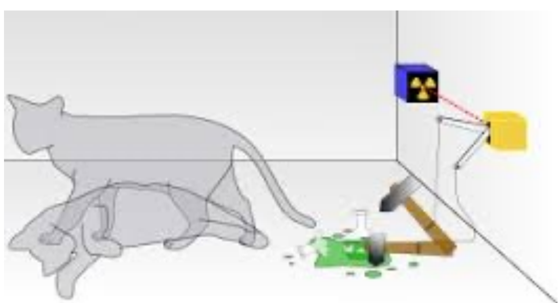
Laplacian: $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$

Planck constant: $\hbar = 1.05457 \times 10^{-27}$ g•cm^2/s

Single-Electron Double-Slit Experiment

What wave?





http://rdg.ext.hitachi.co.jp/rd/moviee/doubleslite-n.mpeg

Akira Tonomura (Hitachi, Ltd.)

Atomic Unit

Length, energy & time in atomic unit

$$\frac{e^2}{r} = \frac{\hbar^2}{mr^2}$$

$$E = \frac{e^2}{r}$$

$$E = \frac{\hbar}{t}$$

CGS
Gaussian unit

$$\begin{cases} \vec{r} = \dfrac{\hbar^2}{me^2}\vec{r}\,' & \dfrac{\hbar^2}{me^2} = 0.529177\ \text{Å} \\[2ex] V = \dfrac{me^4}{\hbar^2}V' & \dfrac{me^4}{\hbar^2} = 27.2116\ \text{eV} \\[2ex] t = \dfrac{\hbar^3}{me^4}t' & \dfrac{\hbar^3}{me^4} = 0.0241889\ \text{fs} \end{cases}$$

Bohr

Hartree

Time-dependent Schrödinger equation in atomic unit

$$ih\frac{\partial}{\partial t}\psi(\vec{r},t) = \left[-\frac{\hbar^2}{2m}\nabla^2 + V(\vec{r})\right]\psi(\vec{r},t)$$

$$\Downarrow$$

$$i\frac{\partial}{\partial t'}\psi(\vec{r}\,',t') = \left[-\frac{\nabla'^2}{2} + V(\vec{r}\,')\right]\psi(\vec{r}\,',t')$$

Two-Dimensional Electron

- **Schrödinger equation (in atomic unit)**

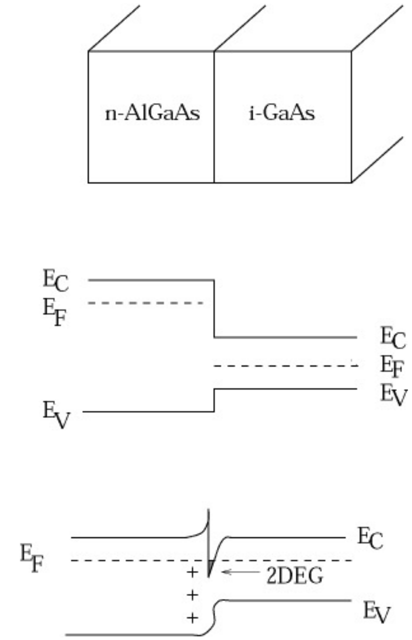
$$i\frac{\partial}{\partial t}\psi(x,y,t) = H\psi(x,y,t)$$

- **Hamiltonian operator**

$$H = -\frac{1}{2}\frac{\partial^2}{\partial x^2} - \frac{1}{2}\frac{\partial^2}{\partial y^2} + V(x,y)$$
$$= T_x + T_y + V$$

- **Periodic boundary condition**

$$\begin{cases}\psi(x+L_x,y) = \psi(x,y)\\ \psi(x,y+L_y) = \psi(x,y)\end{cases}$$



The Nobel Prize in Physics 1985

"for the discovery of the quantized Hall effect"

Klaus von Klitzing

Federal Republic of Germany

Max-Planck-Institut für Festkörperforschung
Stuttgart, Federal Republic of Germany

b. 1943

The Nobel Prize in Physics 1998

"for their discovery of a new form of quantum fluid with fractionally charged excitations"

Robert B. Laughlin

1/3 of the prize

USA

Stanford University
Stanford, CA, USA

b. 1950

Horst L. Störmer

1/3 of the prize

Federal Republic of Germany

Columbia University
New York, NY, USA

b. 1949

Daniel C. Tsui

1/3 of the prize

USA

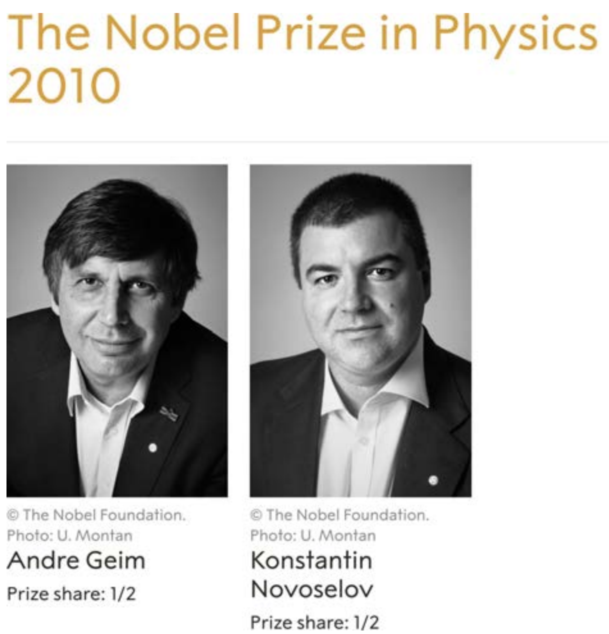
Princeton University
Princeton, NJ, USA

b. 1939
(in Henan, China)

Layered Materials Genome

- **Atomically-thin layered materials will dominate materials science in this century**

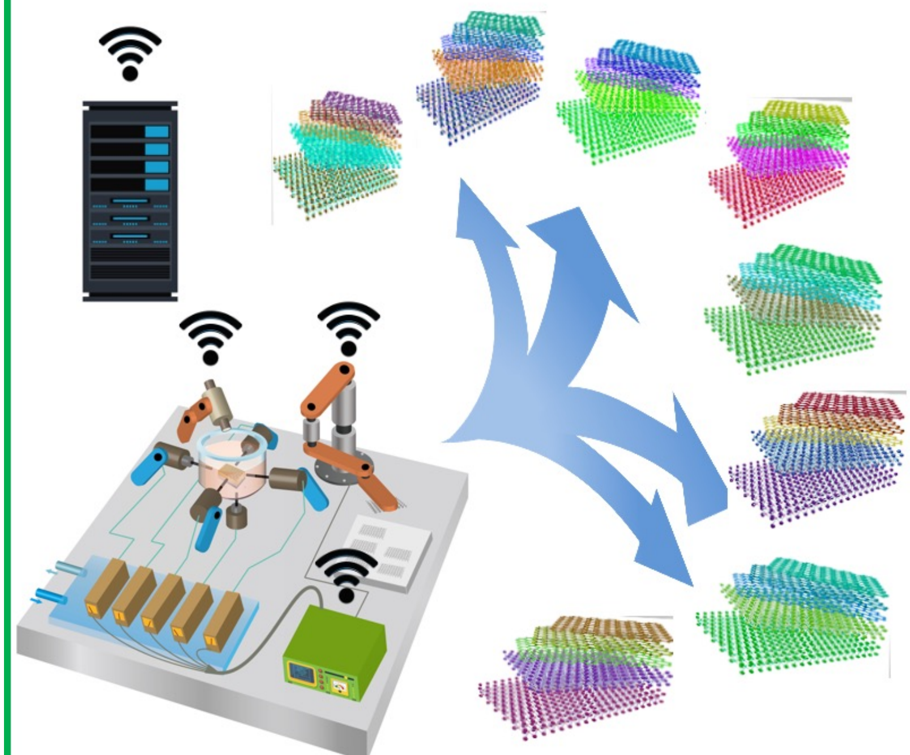
  Geim & Grigorieva, *Nature* **499**, 419 ('13)



- **Tuning material properties in desired ways by building heterostructures composed of unlimited combinations of atomically thin layers in a way similar to genomics**

Future Manufacturing Impact

Precisely aligned van der Waals hetero-structures to enable future twistronics

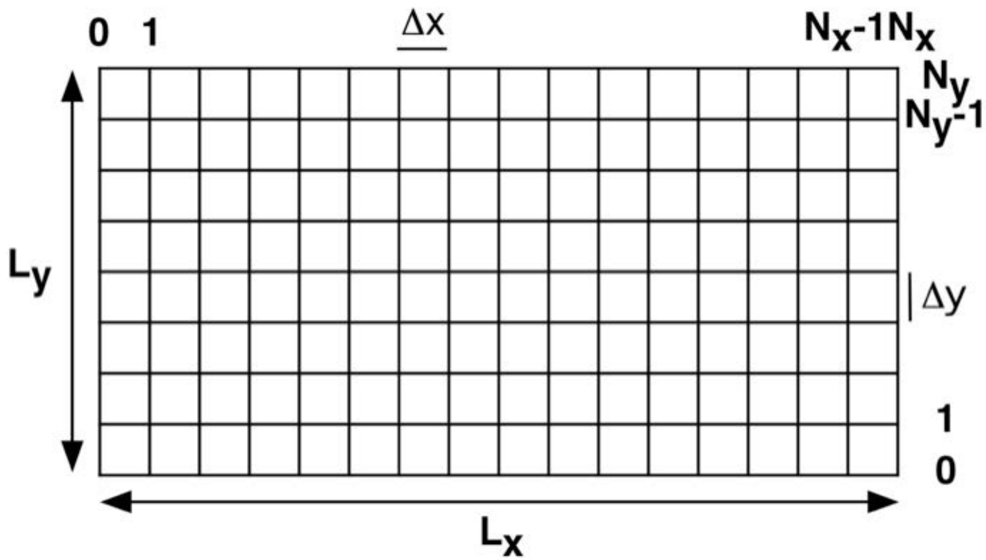


AI- & robotics-guided, scalable manufacturing platform

https://aiqma.netlify.app

Spatial Discretization

- **Regular 2D mesh:** $\psi_{jk} = \psi(j\Delta x, k\Delta y)$ $(\Delta x = L_x/N_x$ & $\Delta y = L_y/N_y)$



- **Finite differencing**

$$
\begin{cases}
(T_x\psi)_{j,k} = -\dfrac{1}{2}\dfrac{\psi_{j-1,k}-2\psi_{j,k}+\psi_{j+1,k}}{(\Delta x)^2} \\
(T_y\psi)_{j,k} = -\dfrac{1}{2}\dfrac{\psi_{j,k-1}-2\psi_{j,k}+\psi_{j,k+1}}{(\Delta y)^2} \\
(V\psi)_{j,k} = V_{j,k}\psi_{j,k}
\end{cases}
$$

$$
\begin{aligned}
&\frac{\partial^2}{\partial x^2}\psi(x,y,t) \\
&= \frac{\partial\psi(x+\Delta/2,y,t)/\partial x - \partial\psi(x-\Delta/2,y,t)/\partial x}{\Delta} \\
&= \frac{\frac{\psi(x+\Delta,y,t)-\psi(x,y,t)}{\Delta} - \frac{\psi(x,y,t)-\psi(x-\Delta,y,t)}{\Delta}}{\Delta} \\
&= \frac{\psi(x+\Delta,y,t)-2\psi(x,y,t)+\psi(x-\Delta,y,t)}{\Delta^2}
\end{aligned}
$$

Temporal Propagation

- **Formal solution to the Schrödinger equation:** $\frac{\partial}{\partial t}\psi(t) = -iH\psi(t)$

$$\psi(t+\Delta t) = \exp(-iH\Delta t)\psi(t)$$

$$\exp(-i\hat{H}t) = \sum_{n=0}^{\infty} \frac{(-it)^n}{n!}\hat{H}^n$$

- **Split-operator method (Trotter-expansion): unitary!**

$$\exp(-i(T_x+T_y)\Delta t) \overset{T_xT_y=T_yT_x}{\Longrightarrow} \exp(-iT_x\Delta t)\exp(-iT_y\Delta t)$$

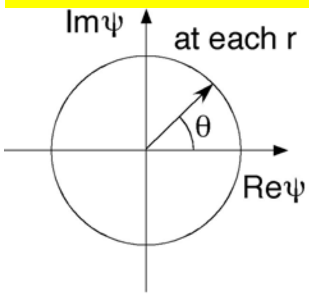
$$\begin{aligned}\psi(t+\Delta t) &= \exp\left(-i\left(T_x+T_y+V\right)\Delta t\right)\psi(t) \\ &= \exp(-iV\Delta t/2)\exp(-iT_x\Delta t)\exp\left(-iT_y\Delta t\right)\exp(-iV\Delta t/2)\psi(t) + O([\Delta t]^3)\end{aligned}$$

Split in a way each operator is easily exponentiated

- **Potential propagator (mesh point-by-point complex-number multiplications)**

$$(u_0+iu_1)(\psi_0+i\psi_1) = (u_0\psi_0 - u_1\psi_1) + i(u_1\psi_0 + u_0\psi_1)$$

$$(\exp(-iV\Delta t/2)\psi)_{jk} = \exp\left(-iV_{jk}\Delta t/2\right)\psi_{jk}$$



Rotation

$$\begin{aligned}&= \left[\cos\left(V_{jk}\Delta t/2\right) - i\sin\left(V_{jk}\Delta t/2\right)\right]\left[\mathrm{Re}\psi_{jk} + i\,\mathrm{Im}\psi_{jk}\right] \\ &= \left[\cos\left(V_{jk}\Delta t/2\right)\mathrm{Re}\psi_{jk} + \sin\left(V_{jk}\Delta t/2\right)\mathrm{Im}\psi_{jk}\right] \\ &\quad + i\left[-\sin\left(V_{jk}\Delta t/2\right)\mathrm{Re}\psi_{jk} + \cos\left(V_{jk}\Delta t/2\right)\mathrm{Im}\psi_{jk}\right]\end{aligned}$$

$$\exp(ia) = \sum_{n=0}^{\infty}\frac{(ia)^n}{n!} = \left(1 - \frac{a^2}{2!} + \frac{a^4}{4!} - \cdots\right) + i\left(a - \frac{a^3}{3!} + \frac{a^5}{5!} - \cdots\right) = \cos(a) + i\sin(a)$$

Kinetic Propagator

- **Mesh-point coupling**

$$T_x\psi_{j,k} = b\psi_{j-1,k} + 2a\psi_{j,k} + b\psi_{j+1,k}$$

- **Tridiagonal matrix representation**

$$T_x = \begin{bmatrix} 2a & b & & & & & b \\ b & 2a & b & & & & \\ & b & 2a & b & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & b & 2a & b & \\ & & & & b & 2a & b \\ b & & & & & b & 2a \end{bmatrix}$$

Note the periodic boundary condition

$$\begin{cases} a = 1/2(\Delta x)^2 \\ b = -1/2(\Delta x)^2 \end{cases}$$

Space Splitting Method (SSM)

- **2×2 block-diagonal decomposition & split-operator exponentiation**

$$T_x = \begin{bmatrix} 2a & b & & & & & b \\ b & 2a & b & & & & \\ & b & 2a & b & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & b & 2a & b & \\ & & & & b & 2a & b \\ b & & & & & b & 2a \end{bmatrix} \qquad \begin{cases} \varepsilon_n^+ = \dfrac{1}{2}\left[\exp\left(-\dfrac{i\Delta t}{n}(a+b)\right) + \exp\left(-\dfrac{i\Delta t}{n}(a-b)\right)\right] \\ \varepsilon_n^- = \dfrac{1}{2}\left[\exp\left(-\dfrac{i\Delta t}{n}(a+b)\right) - \exp\left(-\dfrac{i\Delta t}{n}(a-b)\right)\right] \end{cases}$$

$$= \frac{1}{2}\begin{bmatrix} a & b & & & & & \\ b & a & & & & & \\ & & a & b & & & \\ & & b & a & & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix} + \begin{bmatrix} a & & & & & & b \\ & a & b & & & & \\ & b & a & & & & \\ & & & \ddots & & & \\ & & & & a & b & \\ & & & & b & a & \\ b & & & & & & a \end{bmatrix} + \frac{1}{2}\begin{bmatrix} a & b & & & & & \\ b & a & & & & & \\ & & a & b & & & \\ & & b & a & & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix}$$

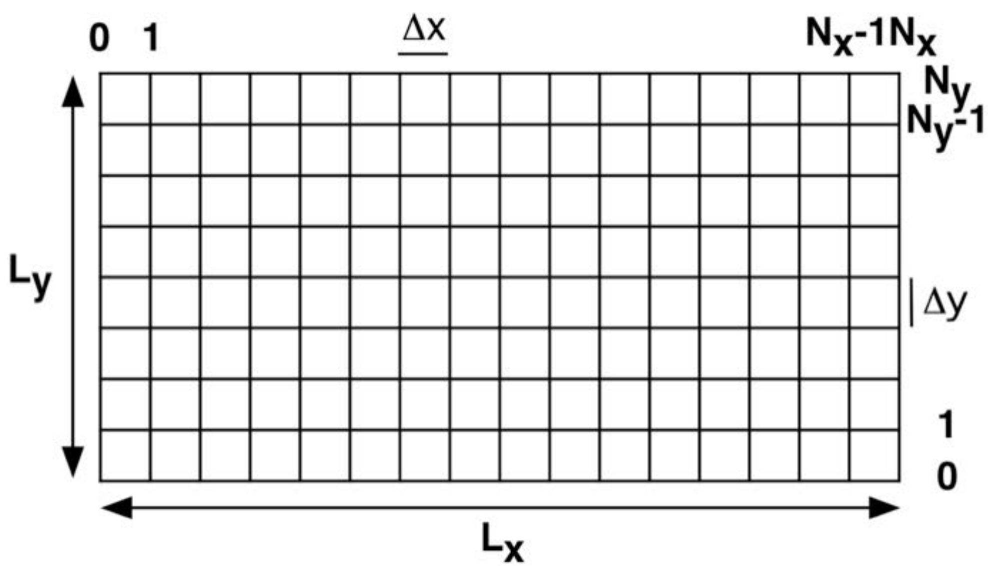
$$\exp(-i\Delta t T_x) = U_x^{(\mathrm{half})} U_x^{(\mathrm{full})} U_x^{(\mathrm{half})} + O([\Delta t]^3) =$$

$$\begin{bmatrix} \varepsilon_2^+ & \varepsilon_2^- & & & & & \\ \varepsilon_2^- & \varepsilon_2^+ & & & & & \\ & & \varepsilon_2^+ & \varepsilon_2^- & & & \\ & & \varepsilon_2^- & \varepsilon_2^+ & & & \\ & & & & \ddots & & \\ & & & & & \varepsilon_2^+ & \varepsilon_2^- \\ & & & & & \varepsilon_2^- & \varepsilon_2^+ \end{bmatrix} \begin{bmatrix} \varepsilon_1^+ & & & & & & \varepsilon_1^- \\ & \varepsilon_1^+ & \varepsilon_1^- & & & & \\ & \varepsilon_1^- & \varepsilon_1^+ & & & & \\ & & & \ddots & & & \\ & & & & \varepsilon_1^+ & \varepsilon_1^- & \\ & & & & \varepsilon_1^- & \varepsilon_1^+ & \\ \varepsilon_1^- & & & & & & \varepsilon_1^+ \end{bmatrix} \begin{bmatrix} \varepsilon_2^+ & \varepsilon_2^- & & & & & \\ \varepsilon_2^- & \varepsilon_2^+ & & & & & \\ & & \varepsilon_2^+ & \varepsilon_2^- & & & \\ & & \varepsilon_2^- & \varepsilon_2^+ & & & \\ & & & & \ddots & & \\ & & & & & \varepsilon_2^+ & \varepsilon_2^- \\ & & & & & \varepsilon_2^- & \varepsilon_2^+ \end{bmatrix}$$

Data Structures in Program qd.c

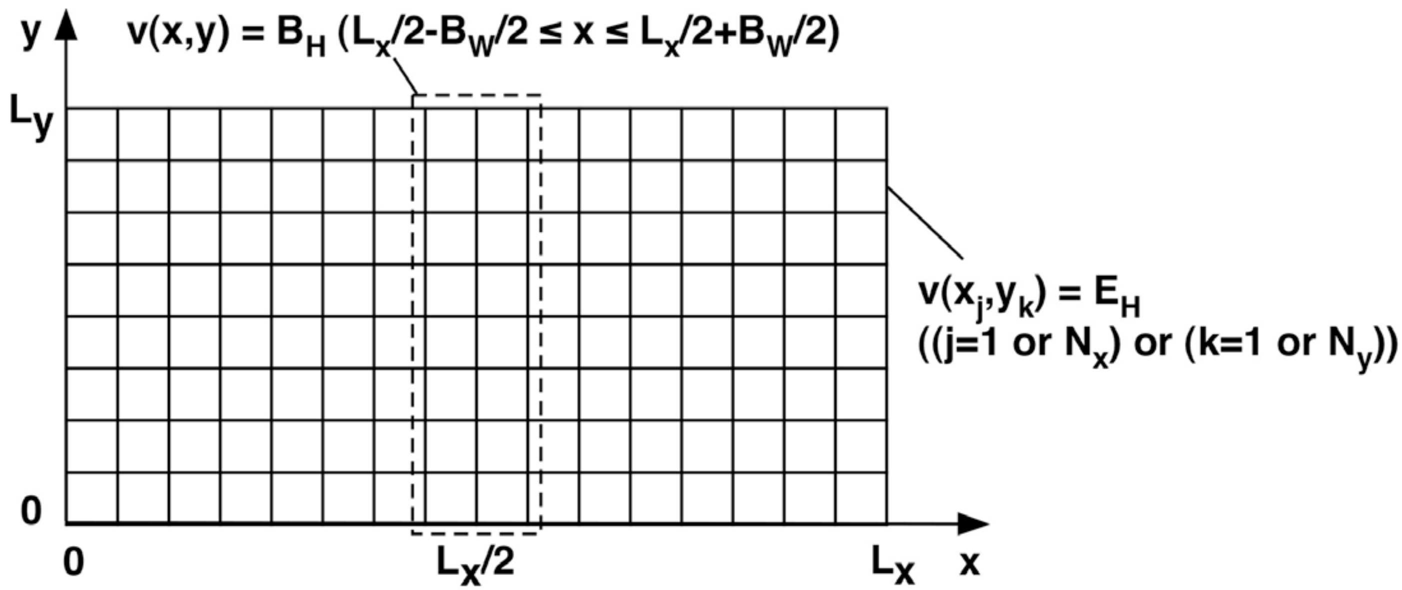
- Wave function: psi[NX+2][NY+2][2]
- Periodic boundary condition by auxiliary elements

```
for (sy=1; sy<=NY; sy++)
  for (s=0; s<=1; s++) {
    psi[0][sy][s] = psi[NX][sy][s];
    psi[NX+1][sy][s] = psi[1][sy][s];}
for (sx=1; sx<=NX; sx++)
  for (s=0; s<=1; s++) {
    psi[sx][0][s] = psi[sx][NY][s];
    psi[sx][NY+1][s] = psi[sx][1][s];}
```



Potential Propagator in qd.c

- **Potential barrier: v[NX+2][NY+2]**



- **Potential propagator: $\exp(-iV\Delta t/2)$, u[NX+2][NY+2][2]**

- **Potential propagation: $\psi \leftarrow \exp(-iV\Delta t/2)\,\psi$**

```
for (sx=1; sx<=NX; sx++)
  for (sy=1; sy<=NY; sy++) {
    wr=u[sx][sy][0]*psi[sx][sy][0]-u[sx][sy][1]*psi[sx][sy][1];
    wi=u[sx][sy][0]*psi[sx][sy][1]+u[sx][sy][1]*psi[sx][sy][0];
    psi[sx][sy][0]=wr;
    psi[sx][sy][1]=wi;}
```

Kinetic Propagator in qd.c

$$\left(U_x^{(\text{half})}\psi\right)_{i,j} = \varepsilon_2^- \delta_{\text{mod}(i,2),0}\psi_{i-1,j} + \varepsilon_2^+\psi_{i,j} + \varepsilon_2^- \delta_{\text{mod}(i,2),1}\psi_{i+1,j}$$

$$\left(U_x^{(\text{full})}\psi\right)_{i,j} = \varepsilon_1^- \delta_{\text{mod}(i,2),1}\psi_{i-1,j} + \varepsilon_1^+\psi_{i,j} + \varepsilon_1^- \delta_{\text{mod}(i,2),0}\psi_{i+1,j}$$

```
/* WRK|PSI holds the new|old wave function */
for (sx=1; sx<=NX; sx++)
  for (sy=1; sy<=NY; sy++) {
    wr=al[d][t][0]*psi[sx][sy][0]-al[d][t][1]*psi[sx][sy][1];
    wi=al[d][t][0]*psi[sx][sy][1]+al[d][t][1]*psi[sx][sy][0];
    if (d==0) {
      wr+=(blx[t][sx][0]*psi[sx-1][sy][0]-blx[t][sx][1]*psi[sx-1][sy][1]);
      wi+=(blx[t][sx][0]*psi[sx-1][sy][1]+blx[t][sx][1]*psi[sx-1][sy][0]);
      wr+=(bux[t][sx][0]*psi[sx+1][sy][0]-bux[t][sx][1]*psi[sx+1][sy][1]);
      wi+=(bux[t][sx][0]*psi[sx+1][sy][1]+bux[t][sx][1]*psi[sx+1][sy][0]);}
    else if (d==1) {
      wr+=(bly[t][sy][0]*psi[sx][sy-1][0]-bly[t][sy][1]*psi[sx][sy-1][1]);
      wi+=(bly[t][sy][0]*psi[sx][sy-1][1]+bly[t][sy][1]*psi[sx][sy-1][0]);
      wr+=(buy[t][sy][0]*psi[sx][sy+1][0]-buy[t][sy][1]*psi[sx][sy+1][1]);
      wi+=(buy[t][sy][0]*psi[sx][sy+1][1]+buy[t][sy][1]*psi[sx][sy+1][0]);}
    wrk[sx][sy][0]=wr;
    wrk[sx][sy][1]=wi;}
/* Copy the new wave function back to PSI */
for (sx=1; sx<=NX; sx++)
  for (sy=1; sy<=NY; sy++)
    for (s=0; s<=1; s++) psi[sx][sy][s]=wrk[sx][sy][s];
```

Initial Wave Function

- **Gaussian wave packet**

$$\psi(x, t=0) = C\exp\left(-\frac{(x-x_0)^2}{4\sigma^2}\right)\exp(ik_0 x)\sin\left(\frac{\pi y}{L_\mathrm{y}}\right)$$

Symbol	Variable in qd.c
x_0 (packet center)	X0
σ (packet spread)	S0
$k_0^2/2$ (energy)	E0

Quantum Dynamics—II
One Dimensional System

Aiichiro Nakano

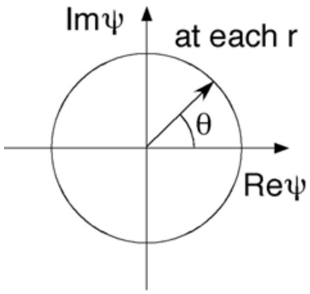
Collaboratory for Advanced Computing & Simulations
Department of Computer Science
Department of Physics & Astronomy
Department of Chemical Engineering & Materials Science
Department of Quantitative & Computational Biology
University of Southern California

Email: anakano@usc.edu

Goal: Understand `qd1.c`

Wave Equation

- **Complex wave function**



$$\psi(x,t) = \mathrm{Re}\psi(x,t) + i\mathrm{Im}\psi(x,t) \in \mathbb{C} \ \left(i = \sqrt{-1}\right)$$

- **Normalization**

$$\int dx |\psi(x,t)|^2 = 1$$

$$\psi^*\psi = (\psi_0 - i\psi_1)(\psi_0 + i\psi_1) = \psi_0^2 + \psi_1^2 \geq 0$$

- **Schrödinger equation (in atomic unit)**

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

- **Hamiltonian operator**

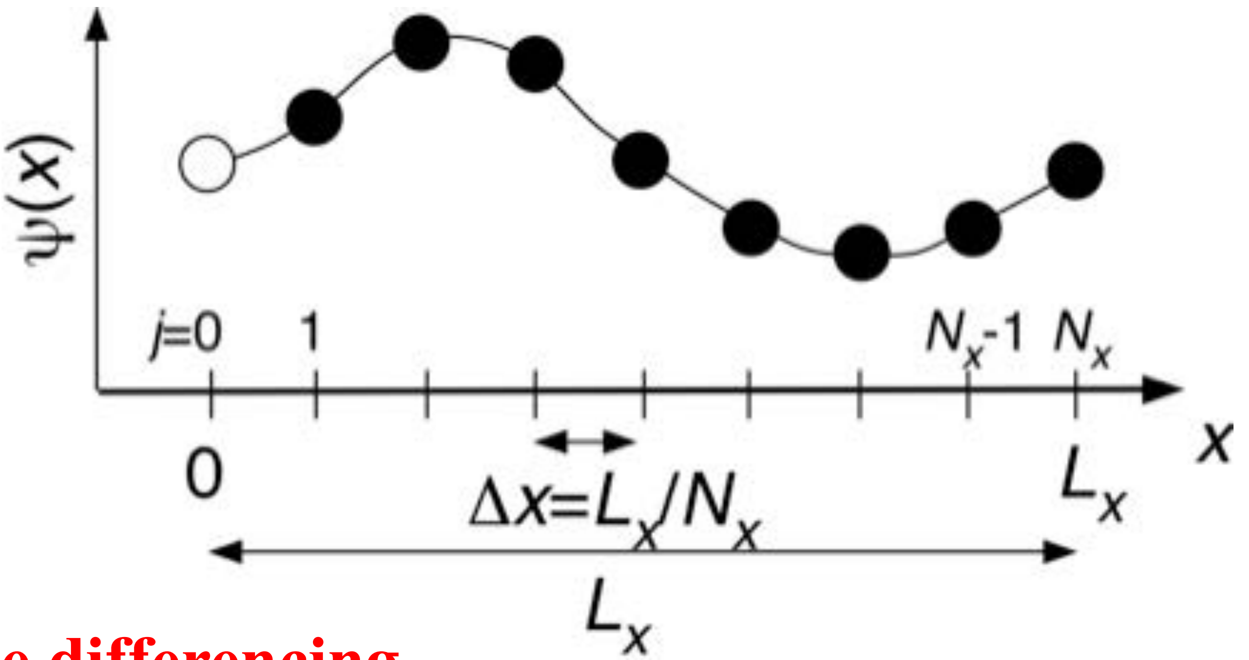
$$H = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + V(x) = T_x + V$$

- **Periodic boundary condition**

$$\psi(x + L_x) = \psi(x)$$

Spatial Discretization

- **Regular 1D mesh:** $\psi_j = \psi(j\Delta x)$ $(\Delta x = L_x/N_x)$



- **Finite differencing**

$$\begin{cases}(T_x\psi)_j = -\dfrac{1}{2}\dfrac{\psi_{j-1} - 2\psi_j + \psi_{j+1}}{(\Delta x)^2} \\ (V\psi)_j = V_j\psi_j\end{cases}$$

$$\frac{\partial^2}{\partial x^2}\psi(x,t) = \frac{\partial\psi(x+\Delta/2,t)/\partial x - \partial\psi(x-\Delta/2,t)/\partial x}{\Delta} = \frac{\frac{\psi(x+\Delta,t)-\psi(x,t)}{\Delta} - \frac{\psi(x,t)-\psi(x-\Delta,t)}{\Delta}}{\Delta} = \frac{\psi(x+\Delta,t)-2\psi(x,t)+\psi(x-\Delta,t)}{\Delta^2}$$

Temporal Propagation

- **Formal solution to the Schrödinger equation:** $\frac{\partial}{\partial t}\psi(t) = -iH\psi(t)$

$$\psi(t+\Delta t) = \exp(-iH\Delta t)\psi(t)$$

$$\exp(-\mathrm{i}\hat{H}t) = \sum_{n=0}^{\infty}\frac{(-it)^n}{n!}\hat{H}^n$$

- **Split-operator method: unitary!**

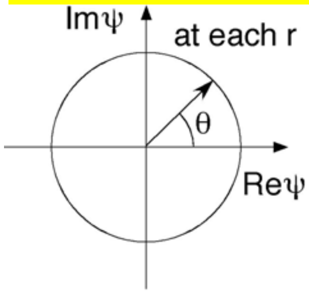
$$\begin{aligned}\psi(t+\Delta t) &= \exp(-i(T_x+V)\Delta t)\psi(t)\\ &= \exp(-iV\Delta t/2)\exp(-iT_x\Delta t)\exp(-iV\Delta t/2)\psi(t) + O([\Delta t]^3)\end{aligned}$$

Split in a way each operator is easily exponentiated

- **Potential propagator (mesh point-by-point complex-number multiplications)**

$$(u_0 + iu_1)(\psi_0 + i\psi_1) = (u_0\psi_0 - u_1\psi_1) + i(u_1\psi_0 + u_0\psi_1)$$

$$\begin{aligned}(\exp(-iV\Delta t/2)\psi)_j &= \exp(-iV_j\Delta t/2)\psi_j\\ &= [\cos(V_j\Delta t/2) - i\sin(V_j\Delta t/2)][\mathrm{Re}\psi_j + i\,\mathrm{Im}\psi_j]\\ &= [\cos(V_j\Delta t/2)\mathrm{Re}\psi_j + \sin(V_j\Delta t/2)\mathrm{Im}\psi_j]\\ &\quad + i[-\sin(V_j\Delta t/2)\mathrm{Re}\psi_j + \cos(V_j\Delta t/2)\mathrm{Im}\psi_j]\end{aligned}$$



Rotation

$$\exp(ia) = \sum_{n=0}^{\infty}\frac{(ia)^n}{n!} = \left(1 - \frac{a^2}{2!} + \frac{a^4}{4!} - \cdots\right) + i\left(a - \frac{a^3}{3!} + \frac{a^5}{5!} - \cdots\right) = \cos(a) + i\sin(a)$$

Kinetic Propagator: It's a Matrix!

- **Mesh-point coupling**

$$T_x\psi_j = b\psi_{j-1} + 2a\psi_j + b\psi_{j+1}$$

- **Tridiagonal matrix representation**

$$T_x = \begin{bmatrix} 2a & b & & & & & b \\ b & 2a & b & & & & \\ & b & 2a & b & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & b & 2a & b & \\ & & & & b & 2a & b \\ b & & & & & b & 2a \end{bmatrix}$$

Note the periodic boundary condition

$$\begin{cases} a = 1/2(\Delta x)^2 \\ b = -1/2(\Delta x)^2 \end{cases}$$

Space Splitting Method (SSM)

- **2×2 block-diagonal decomposition & split-operator exponentiation**

$$T_x = \begin{bmatrix} 2a & b & & & & & b \\ b & 2a & b & & & & \\ & b & 2a & b & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & b & 2a & b & \\ & & & & b & 2a & b \\ b & & & & & b & 2a \end{bmatrix}$$

Block-by-block exponentiation

$$\exp\begin{bmatrix} \blacksquare & & \\ & \blacksquare & \\ & & \blacksquare \end{bmatrix} = \sum_{n=0}^{\infty} \frac{1}{n!} \begin{bmatrix} \blacksquare & & \\ & \blacksquare & \\ & & \blacksquare \end{bmatrix}^n$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!} \begin{bmatrix} \blacksquare^n & & \\ & \blacksquare^n & \\ & & \blacksquare^n \end{bmatrix} = \begin{bmatrix} e^{\blacksquare} & & \\ & e^{\blacksquare} & \\ & & e^{\blacksquare} \end{bmatrix}$$

Split-operator (Trotter expansion) again

$$= \frac{1}{2}\begin{bmatrix} a & b & & & & & \\ b & a & & & & & \\ & & a & b & & & \\ & & b & a & & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix} + \begin{bmatrix} a & & & & & & b \\ & a & b & & & & \\ & b & a & & & & \\ & & & \ddots & & & \\ & & & & a & b & \\ & & & & b & a & \\ b & & & & & & a \end{bmatrix} + \frac{1}{2}\begin{bmatrix} a & b & & & & & \\ b & a & & & & & \\ & & a & b & & & \\ & & b & a & & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix}$$

$$\exp(-i\Delta t T_x) = U_x^{(\text{half})} U_x^{(\text{full})} U_x^{(\text{half})} + O([\Delta t]^3)$$

$$= \exp\left(-\frac{i\Delta t}{2}\begin{bmatrix} a & b & & & & & \\ b & a & & & & & \\ & & a & b & & & \\ & & b & a & & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix}\right) \exp\left(-i\Delta t\begin{bmatrix} a & & & & & & b \\ & a & b & & & & \\ & b & a & & & & \\ & & & \ddots & & & \\ & & & & a & b & \\ & & & & b & a & \\ b & & & & & & a \end{bmatrix}\right) \exp\left(-\frac{i\Delta t}{2}\begin{bmatrix} a & b & & & & & \\ b & a & & & & & \\ & & a & b & & & \\ & & b & a & & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix}\right)$$

How? Block diagonal → block-by-block exponentiation

Space Splitting Method (SSM)

$$\exp(-i\Delta t T_x) = U_x^{(\text{half})} U_x^{(\text{full})} U_x^{(\text{half})} + O([\Delta t]^3)$$

$$= \exp\left(-\frac{i\Delta t}{2}\begin{bmatrix} a & b & & & & \\ b & a & & & & \\ & & a & b & & \\ & & b & a & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix}\right) \exp\left(-i\Delta t \begin{bmatrix} a & & & & & b \\ & a & b & & & \\ & b & a & & & \\ & & & \ddots & & \\ & & & & a & b & \\ & & & & b & a & \\ b & & & & & & a \end{bmatrix}\right) \exp\left(-\frac{i\Delta t}{2}\begin{bmatrix} a & b & & & & \\ b & a & & & & \\ & & a & b & & \\ & & b & a & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix}\right)$$

$$= \begin{bmatrix} \varepsilon_2^+ & \varepsilon_2^- & & & & & \\ \varepsilon_2^- & \varepsilon_2^+ & & & & & \\ & & \varepsilon_2^+ & \varepsilon_2^- & & & \\ & & \varepsilon_2^- & \varepsilon_2^+ & & & \\ & & & & \ddots & & \\ & & & & & \varepsilon_2^+ & \varepsilon_2^- \\ & & & & & \varepsilon_2^- & \varepsilon_2^+ \end{bmatrix} \begin{bmatrix} \varepsilon_1^+ & & & & & & \varepsilon_1^- \\ & \varepsilon_1^+ & \varepsilon_1^- & & & & \\ & \varepsilon_1^- & \varepsilon_1^+ & & & & \\ & & & \ddots & & & \\ & & & & \varepsilon_1^+ & \varepsilon_1^- & \\ & & & & \varepsilon_1^- & \varepsilon_1^+ & \\ \varepsilon_1^- & & & & & & \varepsilon_1^+ \end{bmatrix} \begin{bmatrix} \varepsilon_2^+ & \varepsilon_2^- & & & & & \\ \varepsilon_2^- & \varepsilon_2^+ & & & & & \\ & & \varepsilon_2^+ & \varepsilon_2^- & & & \\ & & \varepsilon_2^- & \varepsilon_2^+ & & & \\ & & & & \ddots & & \\ & & & & & \varepsilon_2^+ & \varepsilon_2^- \\ & & & & & \varepsilon_2^- & \varepsilon_2^+ \end{bmatrix}$$

$$\begin{cases} \varepsilon_n^+ = \dfrac{1}{2}\left[\exp\left(-\dfrac{i\Delta t}{n}(a+b)\right) + \exp\left(-\dfrac{i\Delta t}{n}(a-b)\right)\right] \\ \varepsilon_n^- = \dfrac{1}{2}\left[\exp\left(-\dfrac{i\Delta t}{n}(a+b)\right) - \exp\left(-\dfrac{i\Delta t}{n}(a-b)\right)\right] \end{cases}$$

Just need 2×2 exponentiation

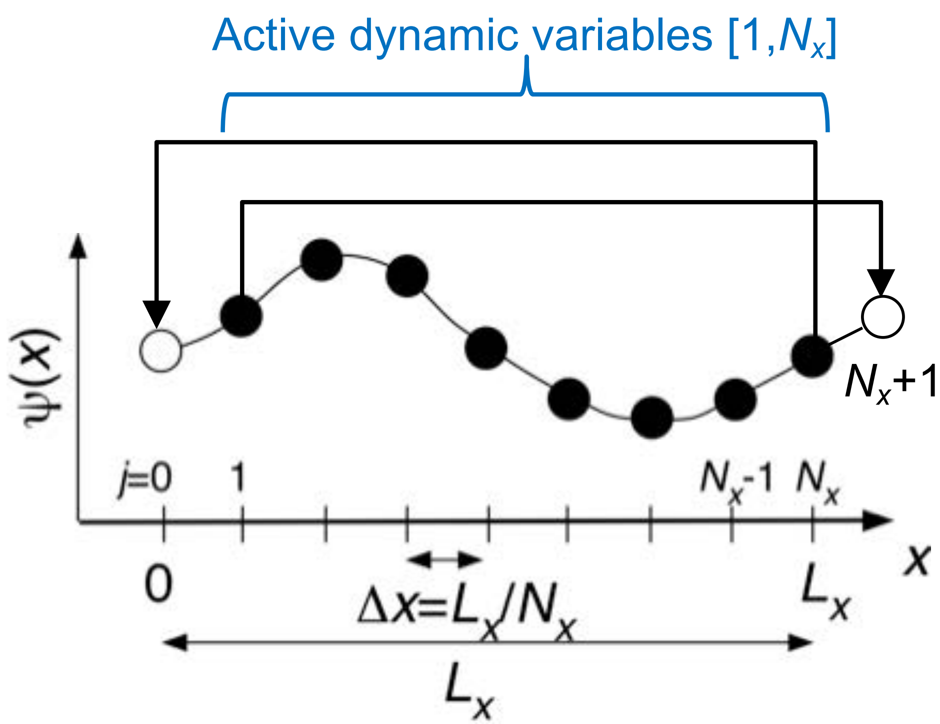
$$\exp\left(-\frac{i\Delta}{(2)}\begin{bmatrix} a & b \\ b & a \end{bmatrix}\right)$$

Use eigen-decomposition & telescoping

Data Structures in Program qd1.c

- **Wave function: psi[NX+2][2]** $psi[j][0|1] = (\mathrm{Re}|\mathrm{Im})\psi_{j\Delta x}$
- **Periodic boundary condition by auxiliary elements**

```
for (s=0; s<=1; s++) {
  psi[0][s] = psi[NX][s];
  psi[NX+1][s] = psi[1][s];
}
```



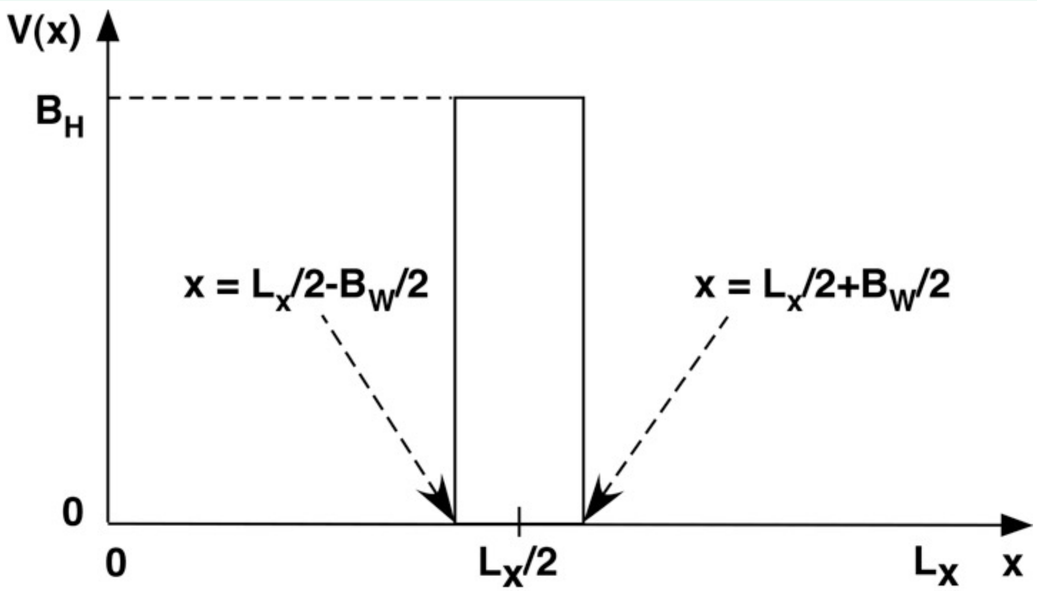
Potential Propagator in qd1.c

- **Potential barrier: v[NX+2]**
- **Potential propagator: $\exp(-iV\Delta t/2)$, u[NX+2][2]**
- **Potential propagation: $\psi \leftarrow \exp(-iV\Delta t/2)\,\psi$**

```
for (sx=1; sx<=NX; sx++) {
  wr=u[sx][0]*psi[sx][0]-u[sx][1]*psi[sx][1];
  wi=u[sx][0]*psi[sx][1]+u[sx][1]*psi[sx][0];
  psi[sx][0]=wr;
  psi[sx][1]=wi;
}
```

$$\begin{cases} u[j][0] = \cos\left(-\dfrac{\Delta}{2}V_j\right) \\ u[j][1] = \sin\left(-\dfrac{\Delta}{2}V_j\right) \end{cases}$$

$$\exp\left(-\frac{iV_j\Delta}{2}\right)\psi_j \equiv u\psi = (u_0 + iu_1)(\psi_0 + i\psi_1) = \overbrace{(u_0\psi_0 - u_1\psi_1)}^{\text{new } \psi_0} + i\,\overbrace{(u_0\psi_1 + u_1\psi_0)}^{\text{new } \psi_1}$$



Kinetic Propagator in qd1.c

$$\left(U_x^{(\mathrm{half})}\psi\right)_i = \overset{\beta_{\mathrm{half}}^{(\mathrm{low})}}{\varepsilon_2^- \delta_{\mathrm{mod}(i,2),0}}\psi_{i-1} + \overset{\alpha_{\mathrm{half}}}{\varepsilon_2^+}\psi_i + \overset{\beta_{\mathrm{half}}^{(\mathrm{up})}}{\varepsilon_2^- \delta_{\mathrm{mod}(i,2),1}}\psi_{i+1}$$

$$\left(U_x^{(\mathrm{full})}\psi\right)_i = \overset{\beta_{\mathrm{full}}^{(\mathrm{low})}}{\varepsilon_1^- \delta_{\mathrm{mod}(i,2),1}}\psi_{i-1} + \overset{\alpha_{\mathrm{full}}}{\varepsilon_1^+}\psi_i + \overset{\beta_{\mathrm{full}}^{(\mathrm{up})}}{\varepsilon_1^- \delta_{\mathrm{mod}(i,2),0}}\psi_{i+1}$$

```
for (sx=1; sx<=NX; sx++) { // wrk[][]|psi[][] holds new|old wave function
  wr=al[t][0]*psi[sx][0]-al[t][1]*psi[sx][1]; // al[0|1][]: α_half|full
  wi=al[t][0]*psi[sx][1]+al[t][1]*psi[sx][0];
  wr+=(bl[t][sx][0]*psi[sx-1][0]-bl[t][sx][1]*psi[sx-1][1]); // bl[0|1][][]: β^(low)_half|full
  wi+=(bl[t][sx][0]*psi[sx-1][1]+bl[t][sx][1]*psi[sx-1][0]);
  wr+=(bu[t][sx][0]*psi[sx+1][0]-bu[t][sx][1]*psi[sx+1][1]); // bu[0|1][][]: β^(high)_half|full
  wi+=(bu[t][sx][0]*psi[sx+1][1]+bu[t][sx][1]*psi[sx+1][0]);}
  wrk[sx][0]=wr;
  wrk[sx][1]=wi;}
for (sx=1; sx<=NX; sx++) // Copy new wave function back to psi
  for (s=0; s<=1; s++)
    psi[sx][s]=wrk[sx][s];
```

$$\psi_j \leftarrow \beta_l\,\psi_{j-1} + \alpha\psi_j + \beta_u\,\psi_{j+1}$$

$$\begin{cases} \varepsilon_n^+ = \dfrac{1}{2}\left[\exp\left(-\dfrac{i\Delta t}{n}(a+b)\right) + \exp\left(-\dfrac{i\Delta t}{n}(a-b)\right)\right] \\ \varepsilon_n^- = \dfrac{1}{2}\left[\exp\left(-\dfrac{i\Delta t}{n}(a+b)\right) - \exp\left(-\dfrac{i\Delta t}{n}(a-b)\right)\right] \end{cases}$$

$$\exp(-i\Delta t T_x) = U_x^{(\mathrm{half})}U_x^{(\mathrm{full})}U_x^{(\mathrm{half})} + O([\Delta t]^3)$$

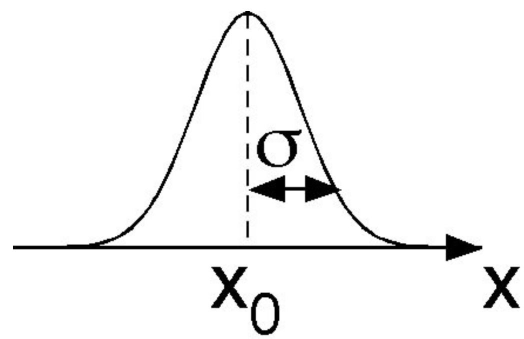
$$= \begin{bmatrix} \varepsilon_2^+ & \varepsilon_2^- & & & & & \\ \varepsilon_2^- & \varepsilon_2^+ & & & & & \\ & & \varepsilon_2^+ & \varepsilon_2^- & & & \\ & & \varepsilon_2^- & \varepsilon_2^+ & & & \\ & & & & \ddots & & \\ & & & & & \varepsilon_2^+ & \varepsilon_2^- \\ & & & & & \varepsilon_2^- & \varepsilon_2^+ \end{bmatrix} \begin{bmatrix} \varepsilon_1^+ & & & & & & \varepsilon_1^- \\ & \varepsilon_1^+ & \varepsilon_1^- & & & & \\ & \varepsilon_1^- & \varepsilon_1^+ & & & & \\ & & & \ddots & & & \\ & & & & \varepsilon_1^+ & \varepsilon_1^- & \\ & & & & \varepsilon_1^- & \varepsilon_1^+ & \\ \varepsilon_1^- & & & & & & \varepsilon_1^+ \end{bmatrix} \begin{bmatrix} \varepsilon_2^+ & \varepsilon_2^- & & & & & \\ \varepsilon_2^- & \varepsilon_2^+ & & & & & \\ & & \varepsilon_2^+ & \varepsilon_2^- & & & \\ & & \varepsilon_2^- & \varepsilon_2^+ & & & \\ & & & & \ddots & & \\ & & & & & \varepsilon_2^+ & \varepsilon_2^- \\ & & & & & \varepsilon_2^- & \varepsilon_2^+ \end{bmatrix}$$

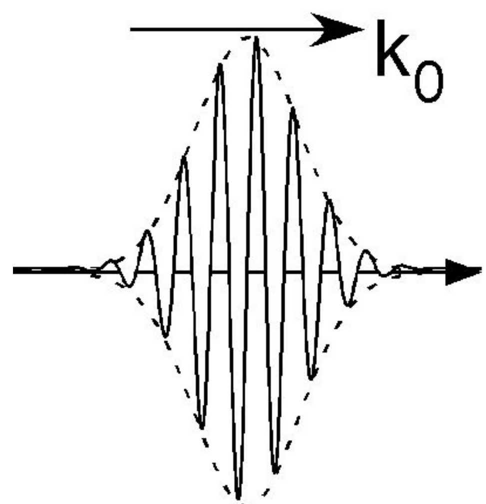
Initial Wave Function

- **Gaussian wave packet**

$$\psi(x, t = 0) = C\exp\left(-\frac{(x - x_0)^2}{4\sigma^2}\right)\exp(ik_0x)$$

Symbol	**Variable in qd1.c**
x_0 (packet center)	X0
σ (packet spread)	S0
$k_0^2/2$ (energy)	E0





Quantum Dynamics—III
Spectral Method

Aiichiro Nakano

Collaboratory for Advanced Computing & Simulations
Department of Computer Science
Department of Physics & Astronomy
Department of Chemical Engineering & Materials Science
Department of Quantitative & Computational Biology
University of Southern California

Email: anakano@usc.edu

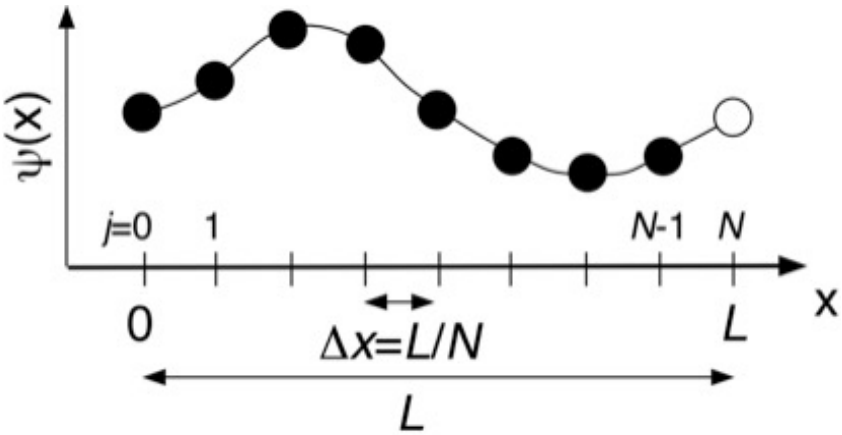
Goal: Understand Fourier transform in the context of the orthonormal plane-wave basis set in a vector space

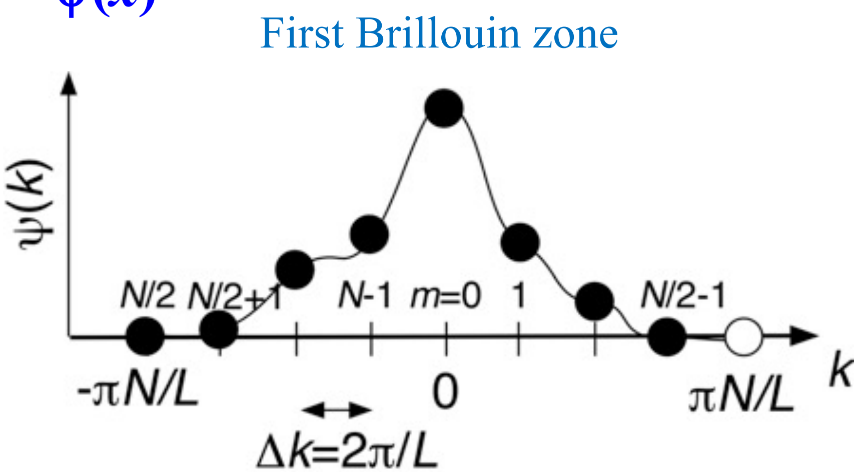
Resolution of identity:

$$1 = \sum_n |n\rangle\langle n|$$

Discrete Fourier Transform

- **Discretize $\psi(x) \in C$ ($x \in [0, L]$) on N mesh points, $x_j = j\Delta x$ ($j = 0, ..., N-1$), with equal mesh spacing, $\Delta x = L/N$**
- **Periodic boundary condition: $\psi(x + L) = \psi(x)$**



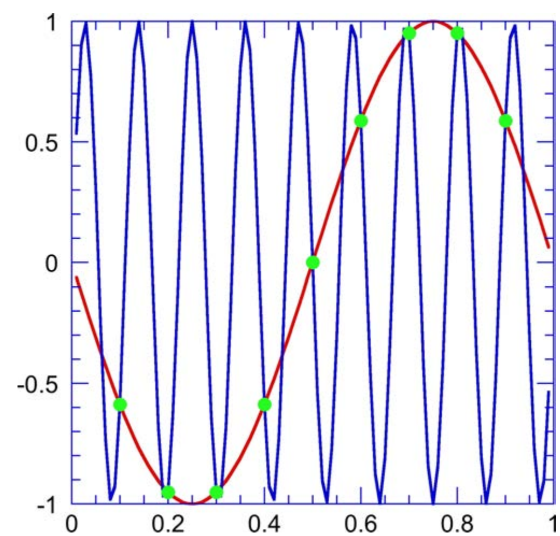


- **Discrete Fourier transform: Represents $\psi(x)$ as a linear combination of $\exp(ikx) = \cos(kx) + i \sin(kx)$, with different wave numbers, k**

$$\psi_j = \sum_{m=0}^{N-1} \tilde{\psi}_m \exp(i k_m x_j)$$

$$k_m = \begin{cases} 2\pi m/L & (m = 0,1,\ldots,N/2-1) \\ 2\pi(m-N)/L & (m = N/2, N/2+1, \ldots, N-1) \end{cases}$$

$$\tilde{\psi}_m = \frac{1}{N}\sum_{j=0}^{N-1} \psi_j \exp(-i k_m x_j)$$



Wave Numbers

- **Periodic boundary condition, $\psi(x+L) = \psi(x)$, is guaranteed by choosing $k_m = 2\pi m/L$**

$$e^{i\left(\frac{2\pi m}{L}(x+L)\right)} = e^{i\left(\frac{2\pi m}{L}x+2\pi m\right)} = e^{i\frac{2\pi m}{L}x}\overbrace{e^{i2\pi m}}^{(e^{i2\pi})^m=1^m=1} = e^{i\frac{2\pi m}{L}x}$$

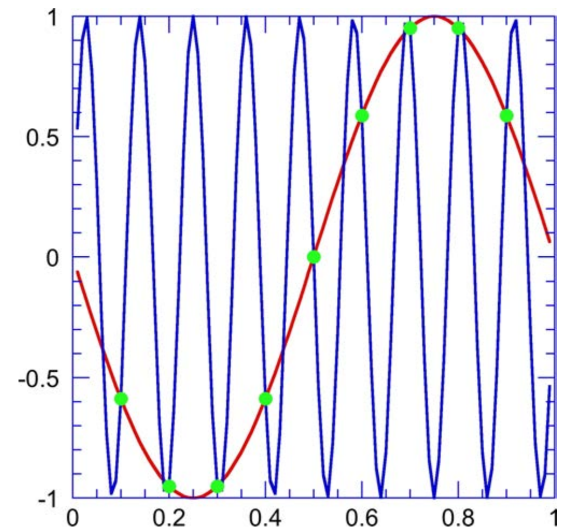
- **Folding back the latter half of wave numbers by** $\frac{2\pi N}{L} = 2\pi/\Delta x$ **(*cf.* first Brillouin zone):**

$\Delta x = \frac{L}{N}$

From $\left[0, \frac{2\pi N}{L}\right] = [0, \frac{2\pi}{\Delta x}]$ **to** $\left[-\frac{2\pi}{L}\cdot\frac{N}{2}, \frac{2\pi}{L}\cdot\frac{N}{2}\right] = [-\frac{\pi}{\Delta x}, \frac{\pi}{\Delta x}]$

The shift won't change wave-function value on any grid point

$$e^{i\left(k_m-\frac{2\pi N}{L}\right)x_j}$$
$$= e^{ik_m x_j}e^{-i\frac{2\pi N}{L}\bullet\frac{L}{N}j}$$
$$= e^{ik_m x_j}e^{\overbrace{-i2\pi j}^{1}}$$
$$= e^{ik_m x_j}$$



Orthonormal Basis Set

- ***N*-dimensional vector space:** $|\psi\rangle = (\psi_0, \psi_1, \ldots, \psi_{N-1})$
- **Plane-wave basis set:** $\left\{ |m\rangle = b_m(x_j) = \frac{1}{\sqrt{N}} \exp(ik_m x_j) \middle| m = 0,1,\ldots,N-1 \right\}$
- **Orthonormality:** $\langle m|n\rangle = \sum_{j=0}^{N-1} b_m^*(x_j) b_n(x_j) = \delta_{m,n} = \begin{cases} 1 & (m = n) \\ 0 & (m \neq n) \end{cases}$

$$\therefore \langle m|n\rangle = \frac{1}{N}\sum_{j=0}^{N-1} \exp\left(i(k_n - k_m)x_j\right) = \frac{1}{N}\sum_{j=0}^{N-1} \exp\left(i\frac{2\pi}{N}(n-m)j\right)$$

$$(k_n - k_m)x_j = \frac{2\pi(n-m)}{L} \cdot \frac{L}{N} j$$

$$= \begin{cases} \frac{1}{N} \frac{\exp(i2\pi(n-m)) - 1}{\exp\left(i\frac{2\pi}{N}(n-m)\right) - 1} = 0 & (m \neq n) \\ \frac{1}{N} \cdot N = 1 & (m = n) \end{cases}$$

Geometric series

$$S = \sum_{j=0}^{N-1} \left(e^{i\frac{2\pi}{N}(n-m)}\right)^j = 1 + \cdots + r^{N-1}$$

$$rS = r + \cdots + r^N$$

$$\therefore (r-1)S = (r^N - 1)$$

$$|\psi\rangle = \sum_m c_m |m\rangle$$

$$\langle n| \times \Downarrow$$

$$\langle n|\psi\rangle = \sum_m c_m \overbrace{\langle n|m\rangle}^{\delta_{nm}} = c_n$$

- **Completeness:** $|\psi\rangle = \sum_{m=0}^{N-1} |m\rangle\langle m|\psi\rangle$ **or** $1 = \sum_{m=0}^{N-1} |m\rangle\langle m|$

$$||$$

- **Fourier transform:** $\psi_j = \sum_{m=0}^{N-1} \exp(ik_m x_j) \boxed{\frac{1}{N}\sum_{l=0}^{N-1} \exp(-ik_m x_l)\psi_l}$ $\tilde{\psi}_m$

Spectral Method

- **Kinetic-energy operator is diagonal in the momentum space:** $\tilde{\psi}_m \xrightarrow{T} \frac{k_m^2}{2}\tilde{\psi}_m$

$$-\frac{1}{2}\frac{\partial^2}{\partial x^2}\sum_{m=0}^{N-1}\tilde{\psi}_m \exp(ik_m x) = \sum_{m=0}^{N-1}\frac{k_m^2}{2}\tilde{\psi}_m \exp(ik_m x)$$

- **Potential-energy operator is diagonal in the real space:** $\psi_j \xrightarrow{V} V_j\,\psi_j$
- **Split-operator technique & spectral method**

$$\psi(t+\Delta t) = \exp\left(-\frac{iV\Delta t}{2}\right) {}^{F}\exp(-iT\Delta t) {}^{F^{-1}}\exp\left(-\frac{iV\Delta t}{2}\right)\psi(t) + O\left((\Delta t)^3\right)$$

1. $\psi_j \xrightarrow[\exp(-iV\Delta t/2)]{} \exp\left(-iV_j\Delta t/2\right)\psi_j$
2. $\psi_j \xrightarrow[F^{-1}]{} F^{-1}\psi_j = \tilde{\psi}_m = \frac{1}{N}\sum_{j=1}^{N}\psi_j \exp\left(-ik_m x_j\right)$
3. $\tilde{\psi}_m \xrightarrow[\exp(-iT\Delta t)]{} \exp\left(-ik_m^2\Delta t/2\right)\tilde{\psi}_m$
4. $\tilde{\psi}_m \xrightarrow[F]{} F\tilde{\psi}_m = \psi_j = \sum_{m=1}^{N}\tilde{\psi}_m \exp\left(ik_m x_j\right)$
5. $\psi_j \xrightarrow[\exp(-iV\Delta t/2)]{} \exp\left(-iV_j\Delta t/2\right)\psi_j$

$$\exp\left(\frac{i\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\sum_m \tilde{\psi}_m e^{-ik_m x} = \sum_m \exp\left(-\frac{i\Delta t k_m^2}{2}\right)\tilde{\psi}_m e^{-ik_m x}$$

Exact exponentiation!

Numerical Recipes FFT: four1()

- **Spectral method requires**

$$\psi_j \xrightarrow[F^{-1}]{} F^{-1}\psi_j = \tilde{\psi}_m = \frac{1}{N}\sum_{j=1}^{N}\psi_j \exp(-ik_m x_j)$$

$$\tilde{\psi}_m \xrightarrow[F]{} F\tilde{\psi}_m = \psi_j = \sum_{m=1}^{N}\tilde{\psi}_m \exp(ik_m x_j)$$

- `four1(double data[], unsigned long nn, int isign)`
  On input, the data[] array contains 2*nn elements that represent nn complex function values, such that data[2*j-1] & data[2*j] (j = 1, ..., nn) are the real & imaginary parts of the function value on the j-th grid point
- **On output data[] is replaced by:**

  —**isign = 1**

$$data_j \leftarrow \sum_{m=1}^{N} data_m \exp(i2\pi mj/N)$$

  —**isign = –1**

$$data_m \leftarrow \sum_{j=1}^{N} data_j \exp(-i2\pi mj/N)$$

$$k_m x_j = \frac{2\pi m}{L} \times \frac{L}{N} j = \frac{2\pi mj}{N}$$

- **Note that four1() does not perform the division by *N* in F^{-1}**

See `four1.c` **in the class home page**

Using four1()

- **Define double psi[2*N], where psi[2*j] & psi[2*j+1] (j = 0, ..., N−1) are the real & imaginary parts of ψ_j**

```
/* Fourier transform */
four1(psi-1, (unsigned long) N, 1);

/* Inverse Fourier transform */
four1(psi-1, (unsigned long) N, -1);
for (j=0; j<2*N; j++)
  psi[j] /= N;
```

- **Note that four1() assumes 1 offset for the first argument but psi[] is 0 offset**

$$\psi_j \xrightarrow[F^{-1}]{} F^{-1}\psi_j = \tilde{\psi}_m = \frac{1}{N}\sum_{j=1}^{N}\psi_j \exp\left(-ik_m x_j\right)$$

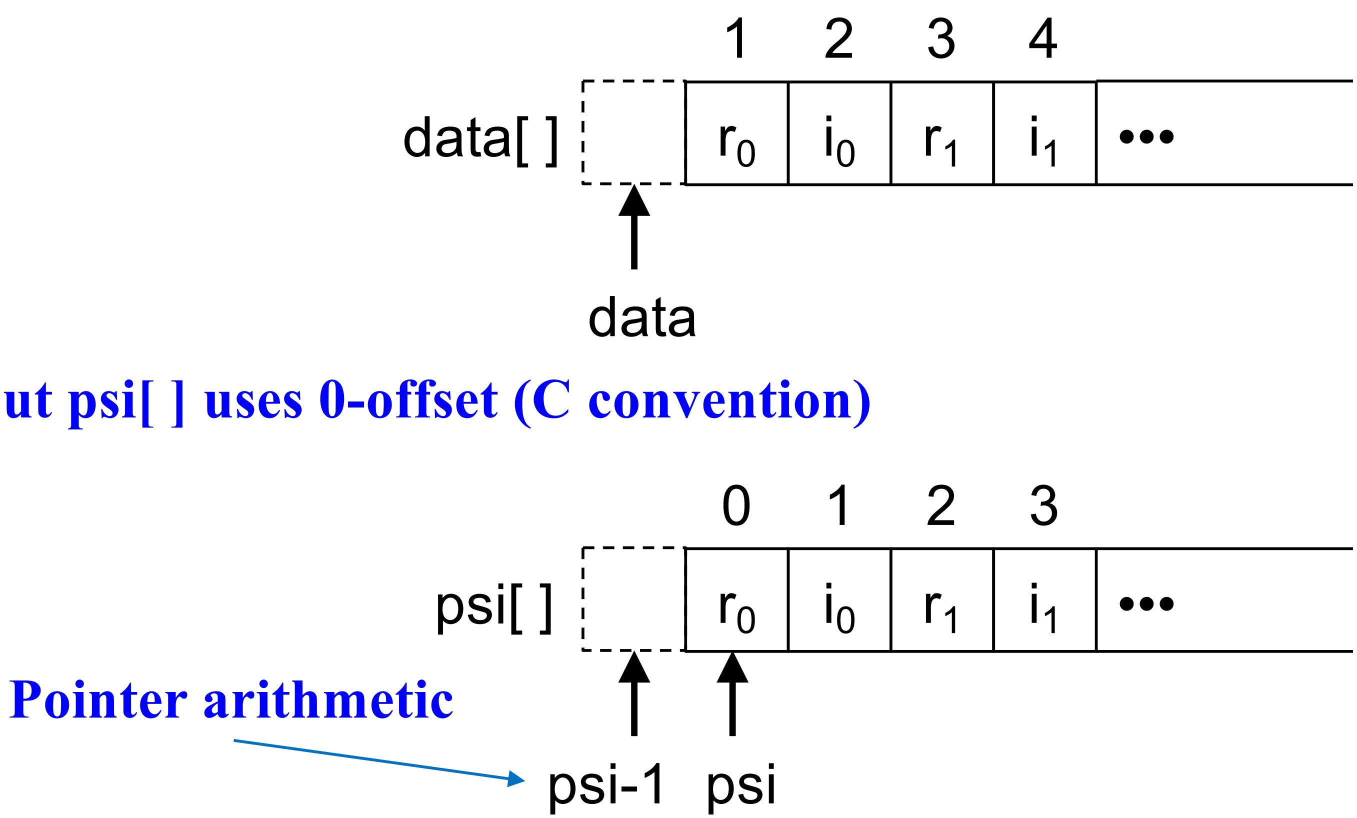
$$\tilde{\psi}_m \xrightarrow[F]{} F\tilde{\psi}_m = \psi_j = \sum_{m=1}^{N}\tilde{\psi}_m \exp\left(ik_m x_j\right)$$

	0	1	2	3		2*N*-2	2*N*-1
psi[ ]	$\mathrm{Re}\psi_0$	$\mathrm{Im}\psi_0$	$\mathrm{Re}\psi_1$	$\mathrm{Im}\psi_1$	•••	$\mathrm{Re}\psi_{N-1}$	$\mathrm{Im}\psi_{N-1}$

Array Offset

```
four1(psi-1, (unsigned long) N, 1);
```

- **four1() assumes 1-offset (because of its Fortran origin)**
- **But psi[ ] uses 0-offset (C convention)**



Energy

$$\langle H \rangle = \langle T \rangle + \langle V \rangle$$

$$= \int dx \psi^*(x) \left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right) \psi(x) + \int dx \psi^*(x) V(x)\, \psi(x)$$

discretize $\cong dx \sum_{j=0}^{N-1} \psi_j^* \left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right) \psi_j + dx \sum_{j=0}^{N-1} \psi_j^* V_j \psi_j$

$= dxN \sum_{m=0}^{N-1} \frac{k_m^2}{2} \left|\tilde{\psi}_m\right|^2 + dx \sum_{j=0}^{N-1} V_j \left|\psi_j\right|^2$ weighted sums

In calc_energy():

1. $\tilde{\psi}_m \leftarrow F^{-1}\left[\psi_j\right]$
2. $E_{\text{kin}} \leftarrow dxN \sum_{m=0}^{N-1} \frac{k_m^2}{2} \left|\tilde{\psi}_m\right|^2$
3. $\psi_j \leftarrow F\left[\tilde{\psi}_m\right]$ **// don't forget**
4. $E_{\text{pot}} \leftarrow dx \sum_{j=0}^{N-1} V_j \left|\psi_j\right|^2$

Kinetic Energy

$$\langle T\rangle = dx \sum_{j=0}^{N-1} \overbrace{\sum_{m=0}^{N-1} \tilde{\psi}_m^* \exp(-ik_m x_j)}^{\psi^*(x_j)} \left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right) \overbrace{\sum_{n=0}^{N-1} \tilde{\psi}_n \exp(ik_n x_j)}^{\psi(x_j)}$$

$$= dx \sum_{j=0}^{N-1} \sum_{m=0}^{N-1} \tilde{\psi}_m^* \exp(-ik_m x_j) \sum_{n=0}^{N-1} \frac{k_n^2}{2} \tilde{\psi}_n \exp(ik_n x_j)$$

$$= dx \sum_{m=0}^{N-1} \tilde{\psi}_m^* \sum_{n=0}^{N-1} \frac{k_n^2}{2} \tilde{\psi}_n \underbrace{\sum_{j=0}^{N-1} \exp(i(k_n - k_m)x_j)}_{N\langle m|n\rangle}$$

$$= dx \sum_{m=0}^{N-1} \tilde{\psi}_m^* \sum_{n=0}^{N-1} \frac{k_n^2}{2} \tilde{\psi}_n \, N\delta_{m,n}$$

$$= dxN \sum_{m=0}^{N-1} \frac{k_m^2}{2} \left|\tilde{\psi}_m\right|^2$$

$$\frac{1-\cos(k_n\Delta x)}{\Delta x^2}$$

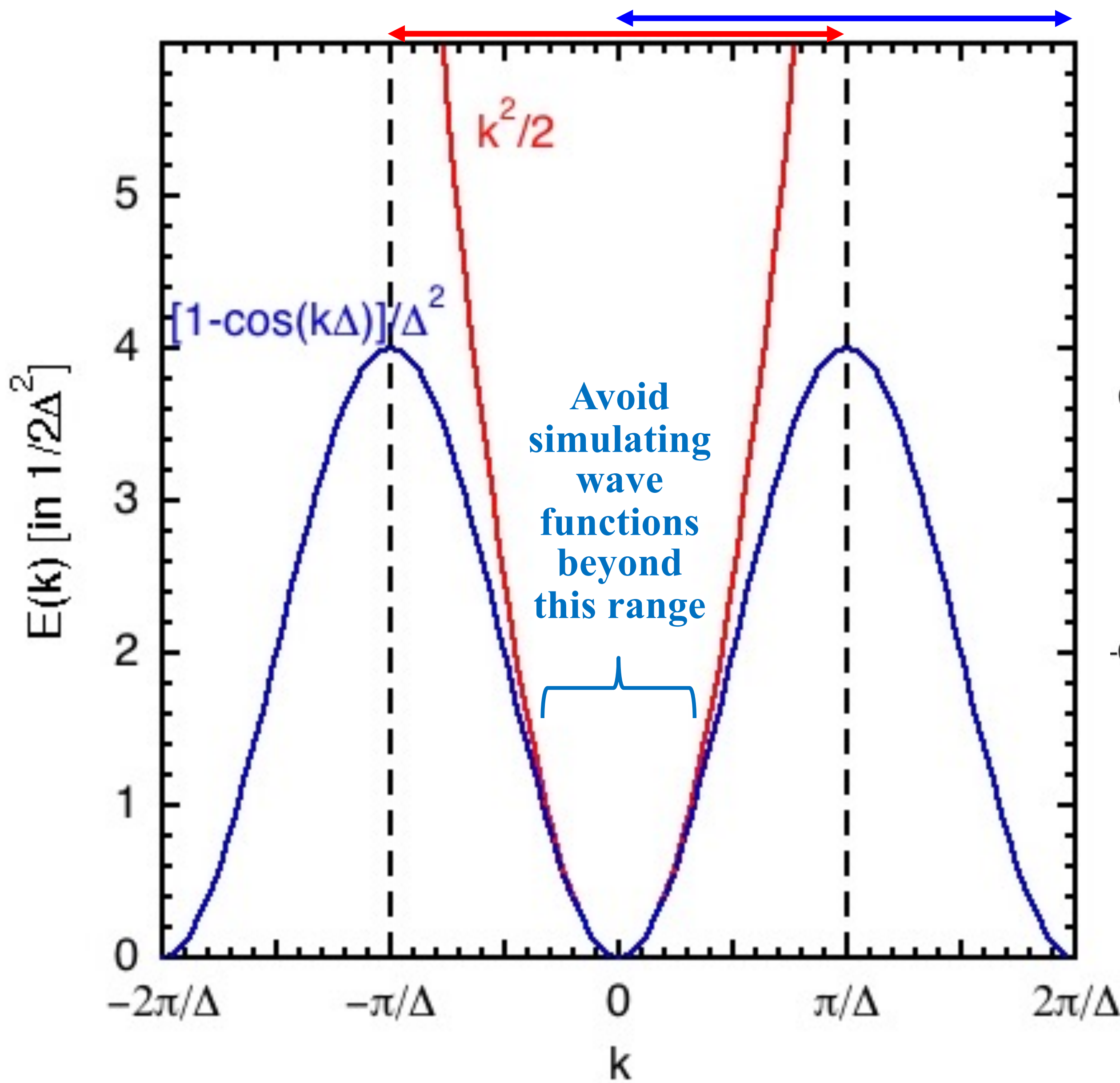
addition theorem

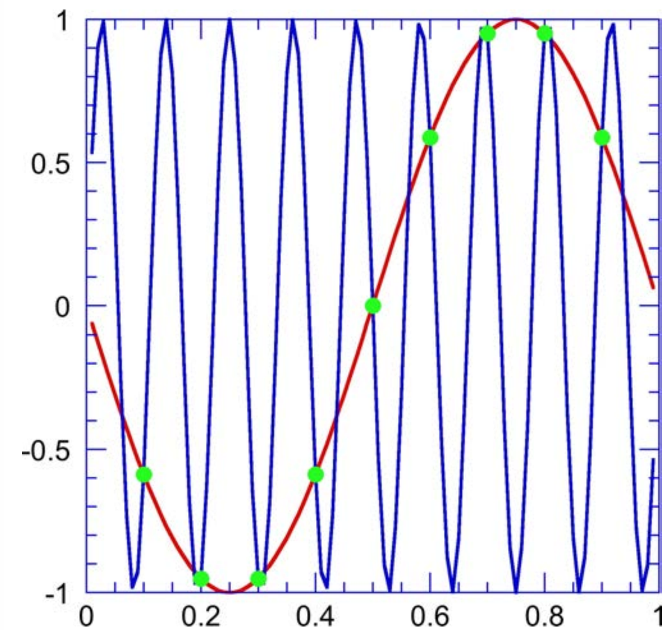
$$= \frac{2\sin^2(\frac{k_n\Delta x}{2})}{\Delta x^2} \xrightarrow[\Delta x\to 0]{} \frac{k_n^2}{2}$$

$$\frac{d^2}{dx^2} e^{ik_n x_j} \cong \frac{e^{-ik_n\Delta x} - 2 + e^{ik_n\Delta x}}{\Delta x^2} e^{ik_n x_j}$$

$$= \frac{2[\cos(k_n\Delta x) - 1]}{\Delta x^2} e^{ik_n x_j}$$

Continuum *vs.* Discrete Kinetic Energy





Total Energy Conservation

- **Energy eigenvalues & eigenvectors:** $H|n\rangle = \varepsilon_n|n\rangle \quad (n = 0, \ldots, N-1)$
- **Time evolution of a wave function**

$$|\psi(t)\rangle = \exp(-iHt)\underbrace{\boxed{\sum_{n=0}^{N-1}|n\rangle\langle n|}}_{=1\text{: completeness (resolution of identity)}}\psi(0)\rangle = \sum_{n=0}^{N-1}\exp(-i\varepsilon_n t)|n\rangle\langle n|\psi(0)\rangle$$

- **Total energy**

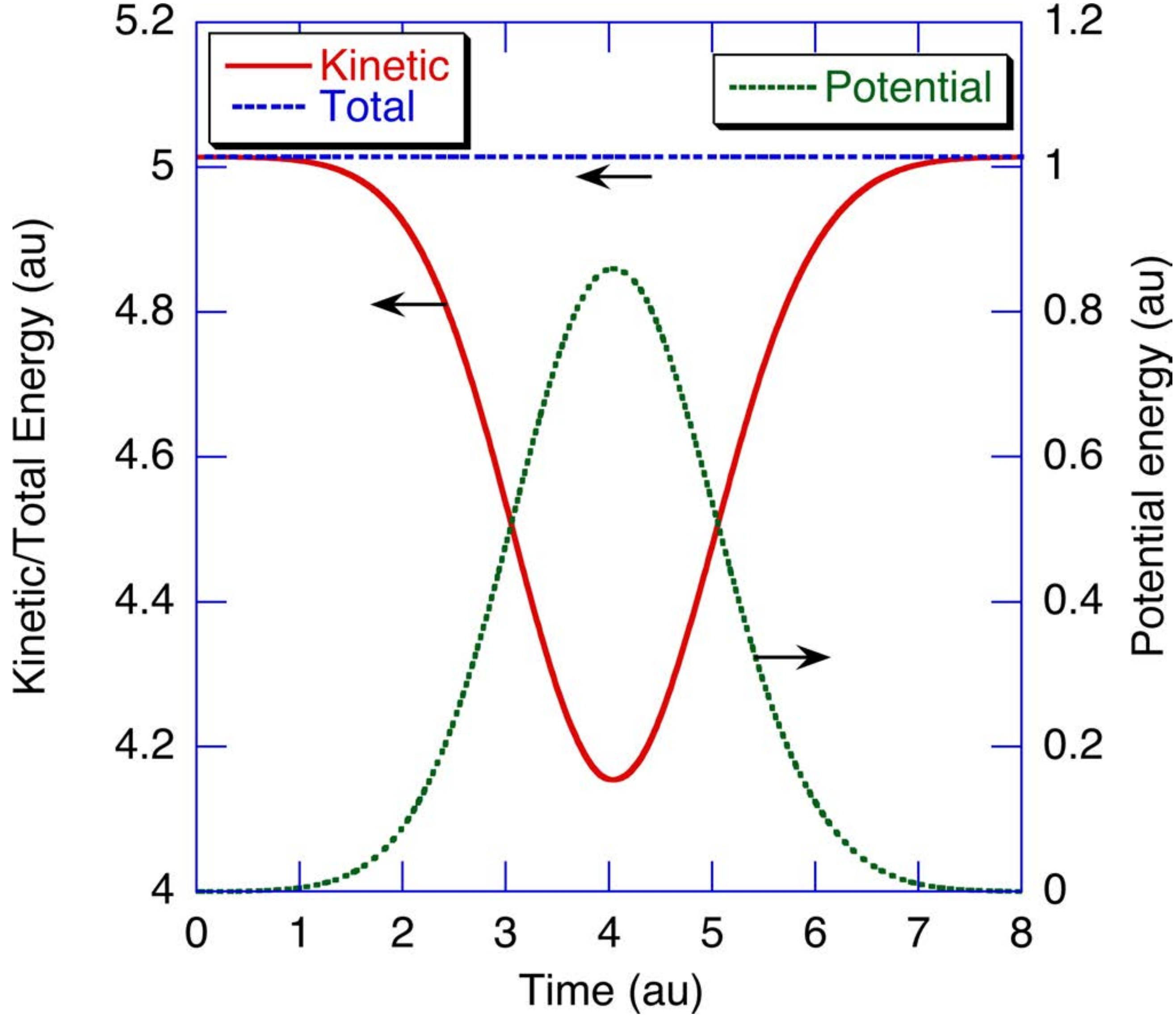
$$\langle\psi(t)|H|\psi(t)\rangle = \left(\sum_{m=0}^{N-1}\langle\psi(0)|m\rangle\exp(i\varepsilon_m t)\langle m|\right)H\left(\sum_{n=0}^{N-1}\exp(-i\varepsilon_n t)|n\rangle\langle n|\psi(0)\rangle\right)$$

$$= \left(\sum_{m=0}^{N-1}\langle\psi(0)|m\rangle\exp(i\varepsilon_m t)\langle m|\right)\left(\sum_{n=0}^{N-1}\exp(-i\varepsilon_n t)\varepsilon_n|n\rangle\langle n|\psi(0)\rangle\right)$$

$$= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\exp(i(\varepsilon_m - \varepsilon_n)t)\varepsilon_n\langle\psi(0)|m\rangle\langle n|\psi(0)\rangle\underbrace{\langle m|n\rangle}_{\delta_{m,n}}$$

$$= \sum_{n=0}^{N-1}\varepsilon_n|\langle n|\psi(0)\rangle|^2 = \text{constant}$$

Energy Conservation for 1D Square Barrier



Energy conservation: good program verification

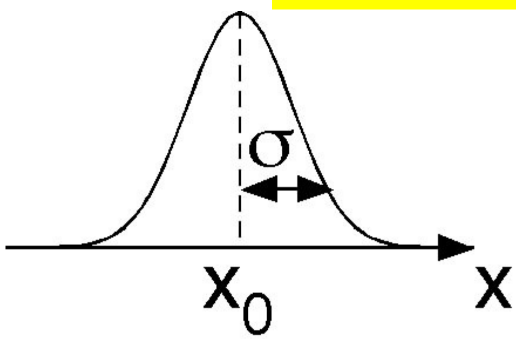
Initial Wave Function

- **Gaussian wave packet**

X0

$$\psi(x,t=0) = C\exp\left(-\frac{(x-x_0)^2}{4\sigma^2}\right)\exp(ik_0x)$$

S0 E0 = $k_0^2/2$

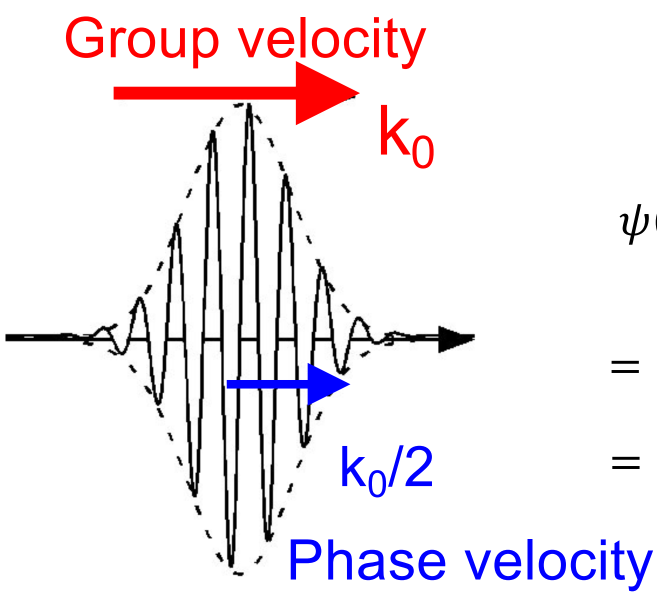


$$i\frac{\partial}{\partial t}\exp\left(ik_0x - i\frac{k_0^2}{2}t\right) = \frac{k_0^2}{2}\exp\left(ik_0x - i\frac{k_0^2}{2}t\right)$$

Free-space solution

||

$$\left[-\frac{1}{2}\frac{d^2}{dx^2}+0\right]\exp\left(ik_0x - i\frac{k_0^2}{2}t\right) = \frac{k_0^2}{2}\exp\left(ik_0x - i\frac{k_0^2}{2}t\right)$$



$$\psi(x,t) = \int dk\tilde{\psi}(k)\exp(ikx - i\omega(k)t) \quad \text{Here, } \omega(k) = k^2/2$$

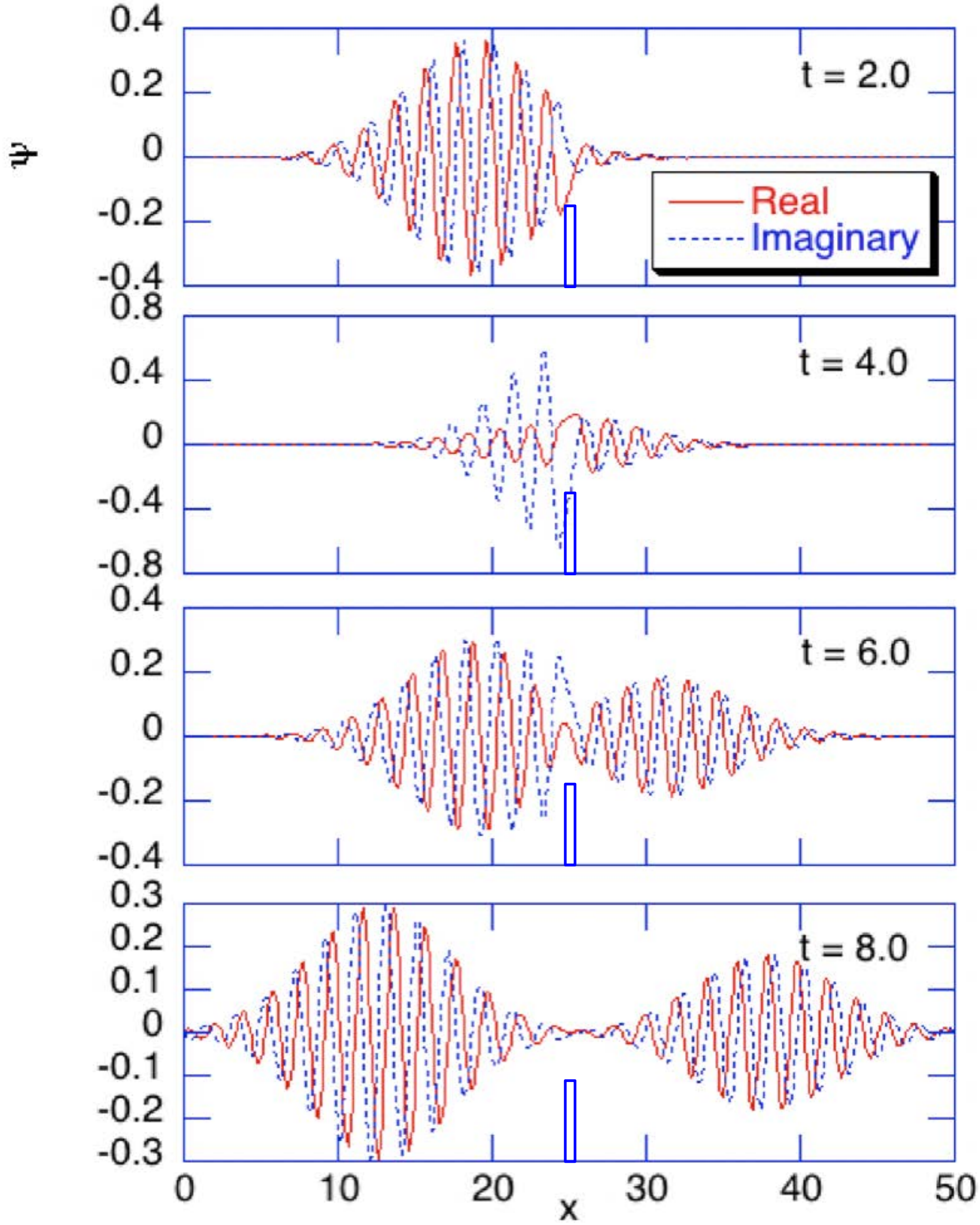
$$= \int dk\tilde{\psi}(k)\exp\left(i\overbrace{k}^{k_0+k-k_0}x - i\overbrace{\omega(k)}^{\cong\omega(k_0)+\frac{d\omega}{dk_0}(k-k_0)}t\right)$$

$$= \exp\left(ik_0\left(x-\frac{\omega(k_0)}{k_0}t\right)\right)\int dk\tilde{\psi}(k)\exp\left(i(k-k_0)\left(x-\frac{d\omega}{dk_0}t\right)\right)$$

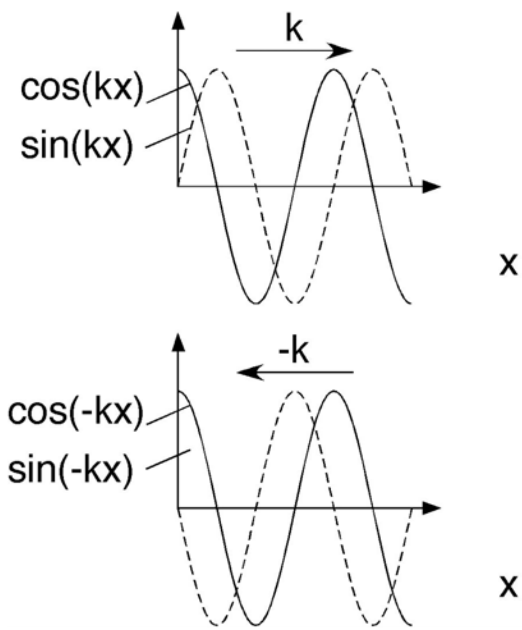
Phase velocity **Group velocity**

See Gaussian wave-packet movie at https://www.cond-mat.de/teaching/QM/JSim/wpack.html

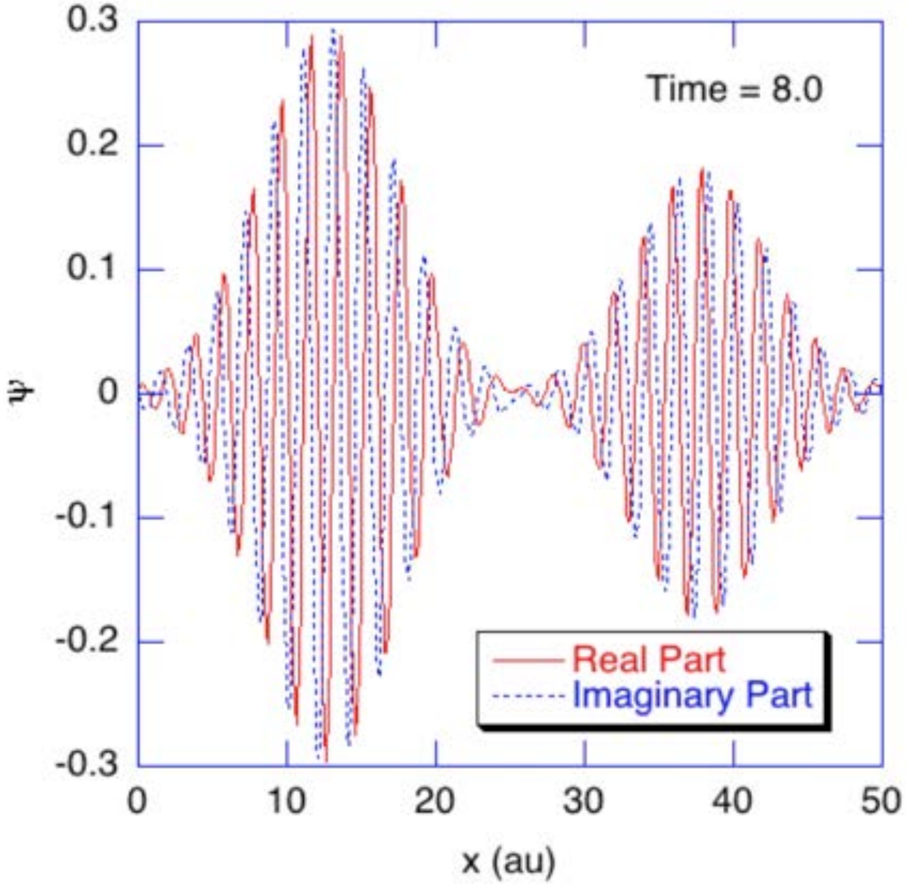
Numerical Example

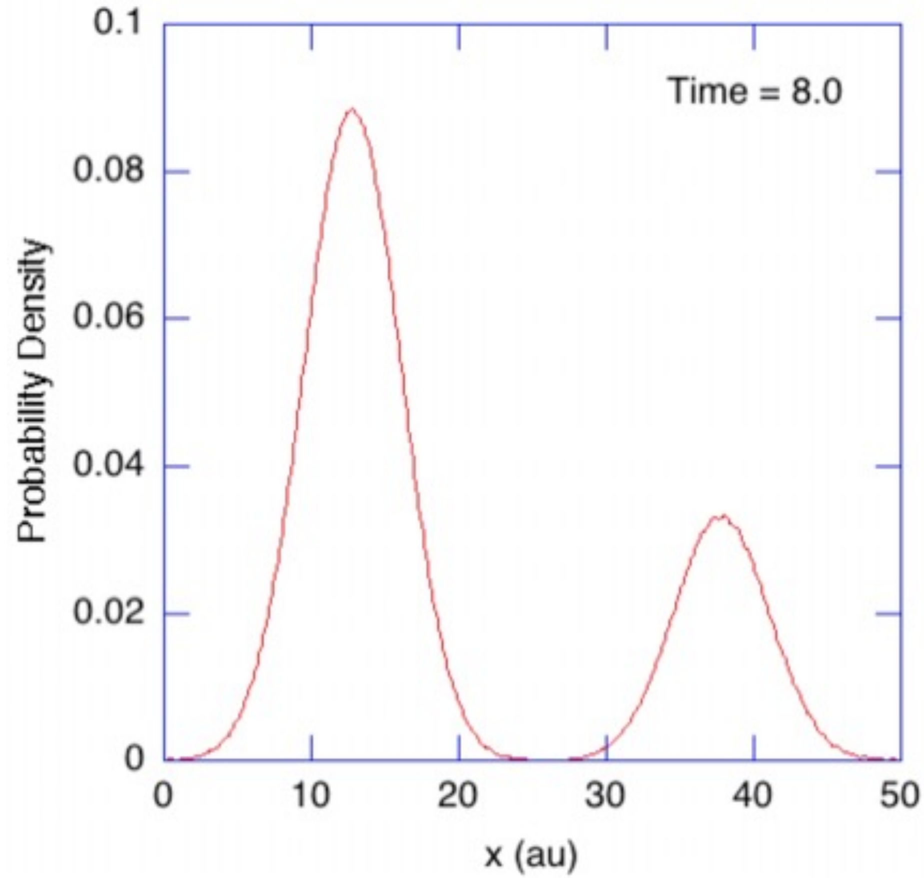


$\Delta t = 2\text{x}10^{-3}$ au



Wave Function & Probability





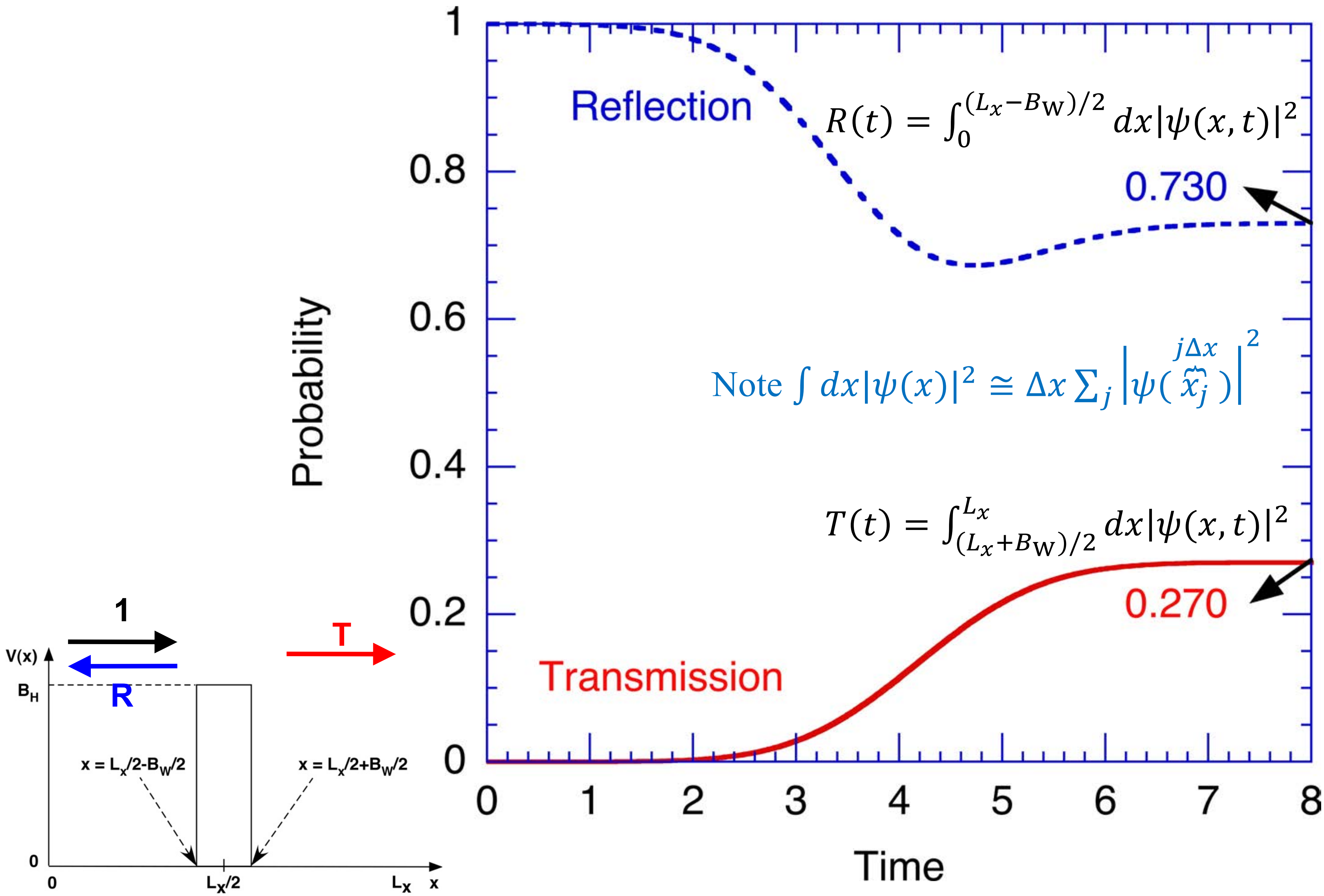
qd1.in

```
50.0            LX
2.0e-3          DT
4000            NSTEP
10              NECAL
12.5 3.0 5.0    X0 S0 E0
5.0 1.0         BH BW
50.0            EH
```

qd1.h

```
#define NX 512
```

Transmission & Reflection Coefficients



Top 10 Algorithms in History

In putting together this issue of *Computing in Science & Engineering*, we knew three things: it would be difficult to list just 10 algorithms; it would be fun to assemble the authors and read their papers; and, whatever we came up with in the end, it would be controversial. We tried to assemble the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century. Following is our list (here, the list is in chronological order; however, the articles appear in no particular order):

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

PHYS 516
CSCI 596
CSCI 653

IEEE CiSE, Jan/Feb (2000)

Fast Fourier Transform

- **Danielson-Lanczos algorithm:**

$$\psi_j = \sum_{m=0}^{N-1} \tilde{\psi}_m \exp(ik_m x_j) = \sum_{m=0}^{N-1} \tilde{\psi}_m \exp(i2\pi mj/N) \quad \boldsymbol{O(N^2)!}$$

$$= \underbrace{\sum_{m=0}^{N/2-1} \tilde{\psi}_{2m} \exp(i2\pi(2m)j/N)}_{\text{even terms}} + \underbrace{\sum_{m=0}^{N/2-1} \tilde{\psi}_{2m+1} \exp(i2\pi(2m+1)j/N)}_{\text{odd terms}}$$

$$= \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m} \exp(i2\pi mj/(N/2)) + \exp(i2\pi j/N) \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m+1} \exp(i2\pi mj/(N/2))$$

$$\psi_j = \psi_j^0 + W_N^j \psi_j^1 \qquad \begin{cases} \psi_j^0 = \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m} \exp(i2\pi mj/(N/2)) \\ \psi_j^1 = \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m+1} \exp(i2\pi mj/(N/2)) \\ W_N = \exp(i2\pi/N) \end{cases}$$

subarray Fourier decompositions

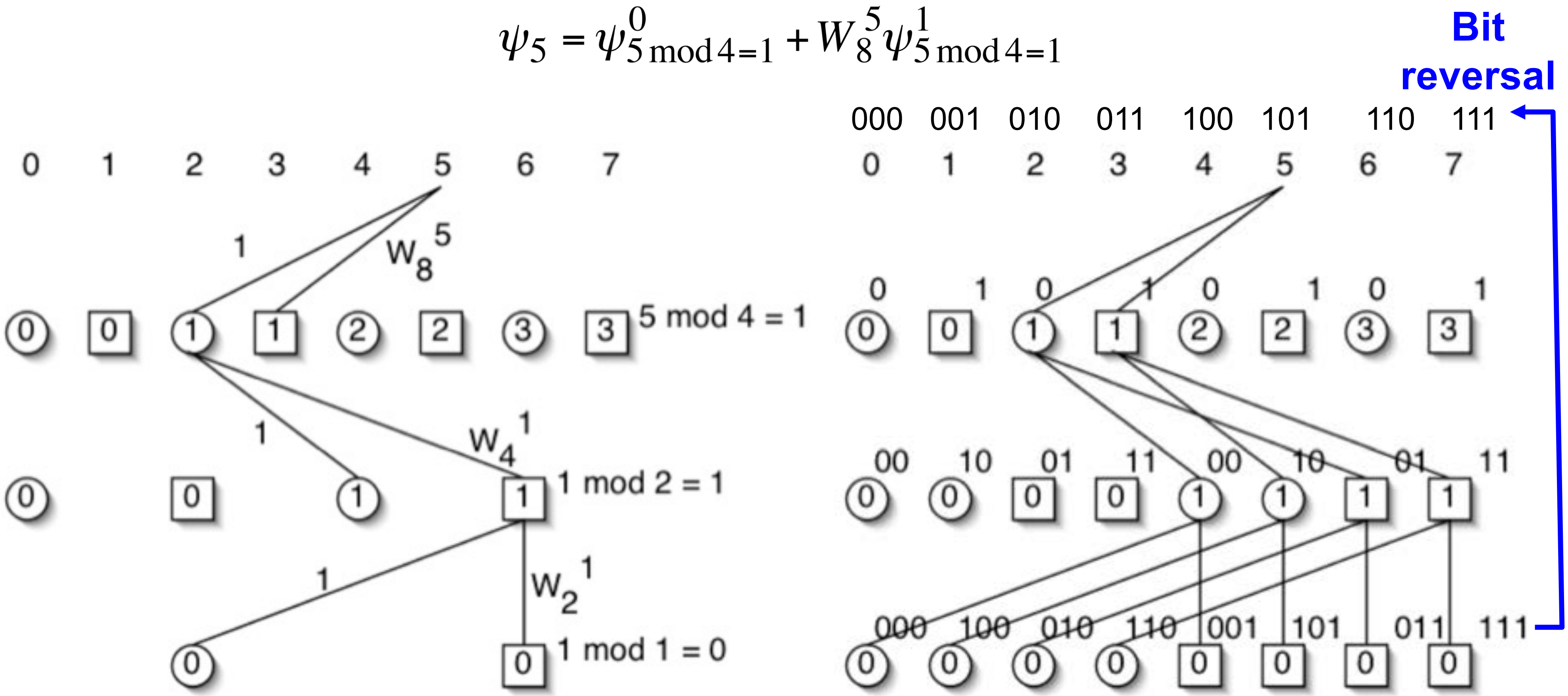
j **read as** ***j*** **mod** ***N*/2**

Divide-and-conquer

Fast Fourier Transform

- **Recursive sub-Fourier transforms:** $\psi_j = \psi_j^0 + W_N^j \psi_j^1$

$$\psi_5 = \psi^0_{5 \bmod 4 = 1} + W_8^5 \psi^1_{5 \bmod 4 = 1}$$



010→010→2 011→110→6

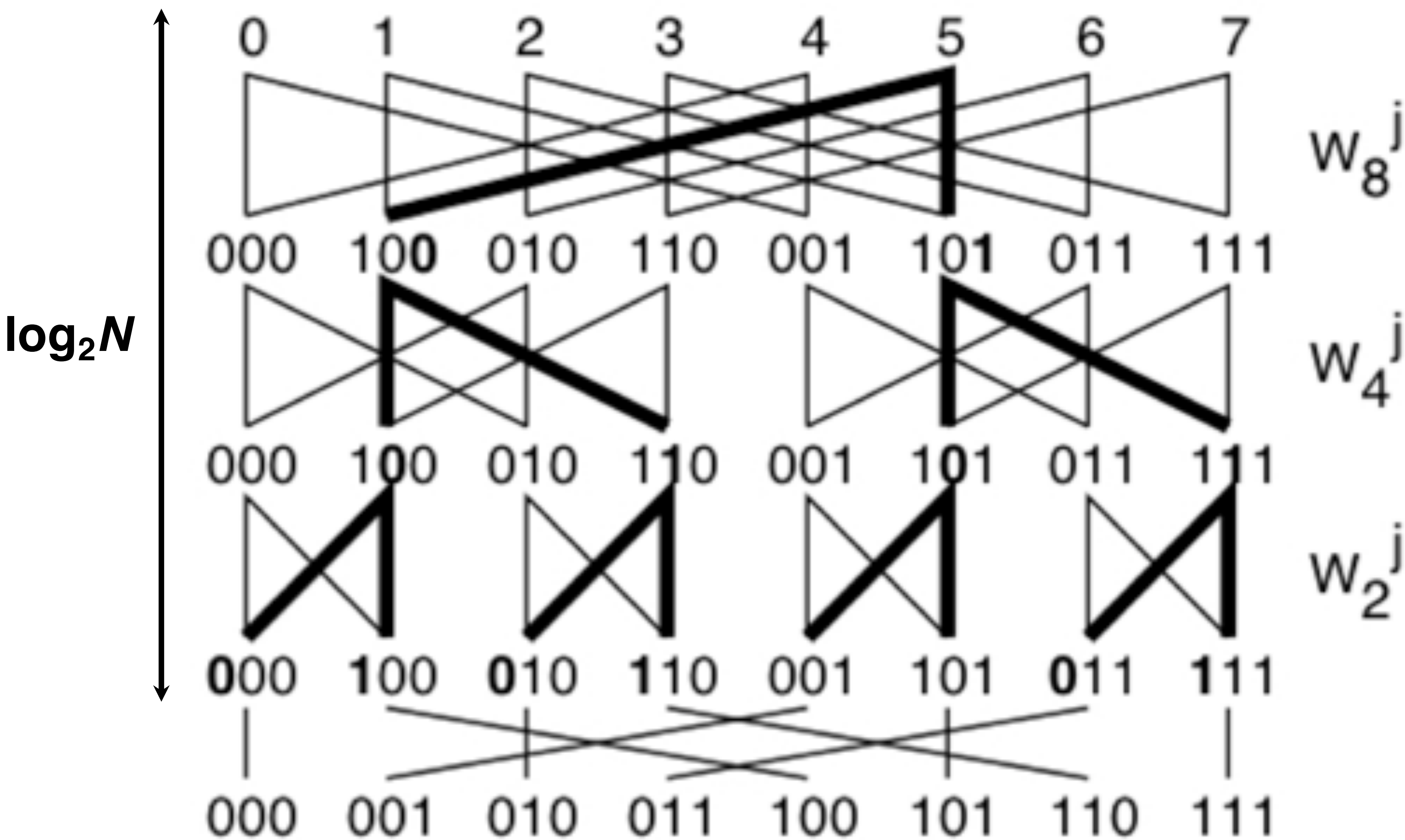
Path bit reversal = element

Recursion tree

***O*(*N*) operations per element**

Fast Fourier Transform Algorithm

- **Butterfly (hypercube) data exchange after bit-reversal:**



- **Many computations are shared among the recursion trees**
- **$2N\log_2 N$ arithmetic operations**

Parallelizing Quantum Dynamics

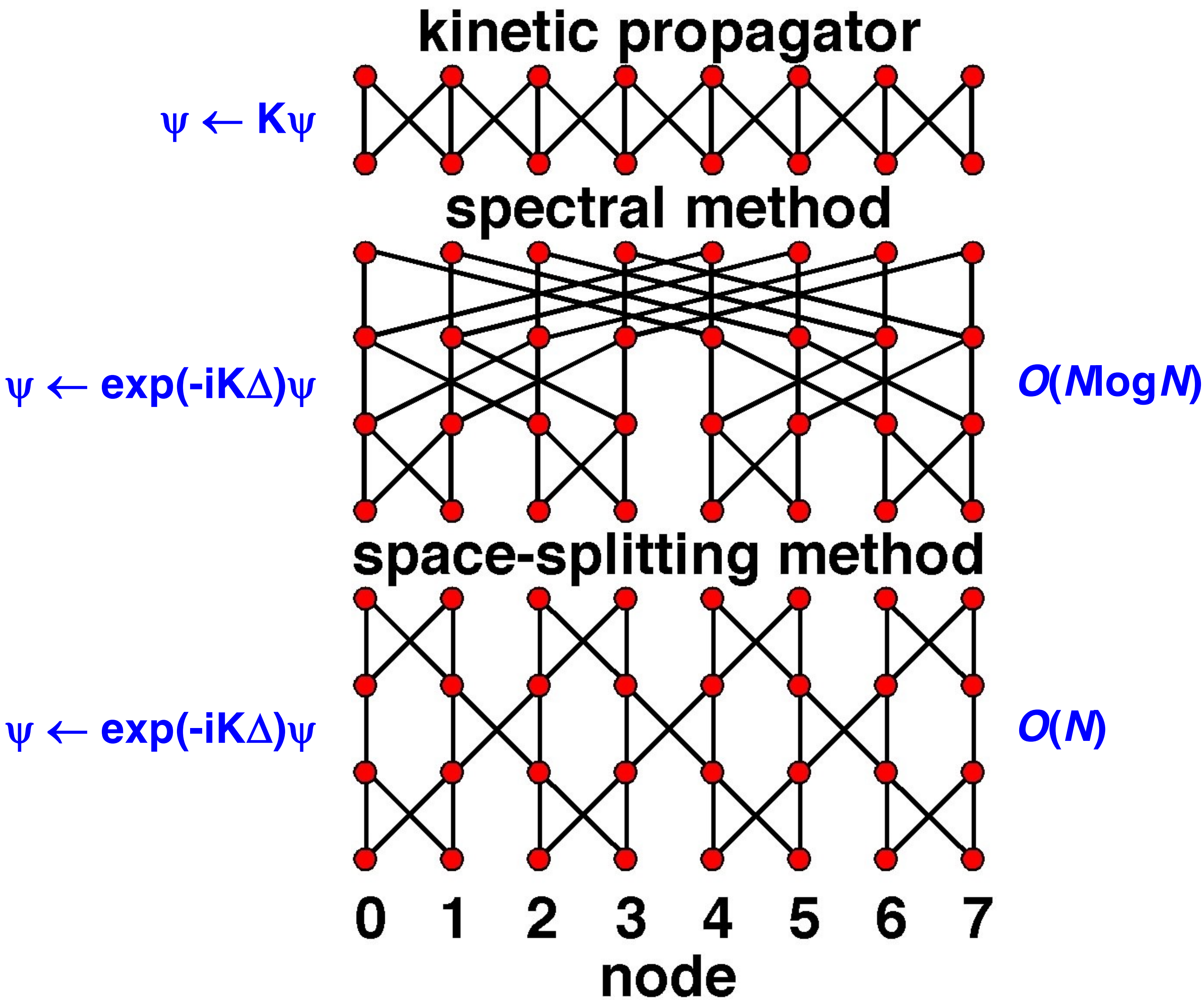
Aiichiro Nakano

Collaboratory for Advanced Computing & Simulations
Department of Computer Science
Department of Physics & Astronomy
Department of Chemical Engineering & Materials Science
Department of Quantitative & Computational Biology
University of Southern California

Email: anakano@usc.edu

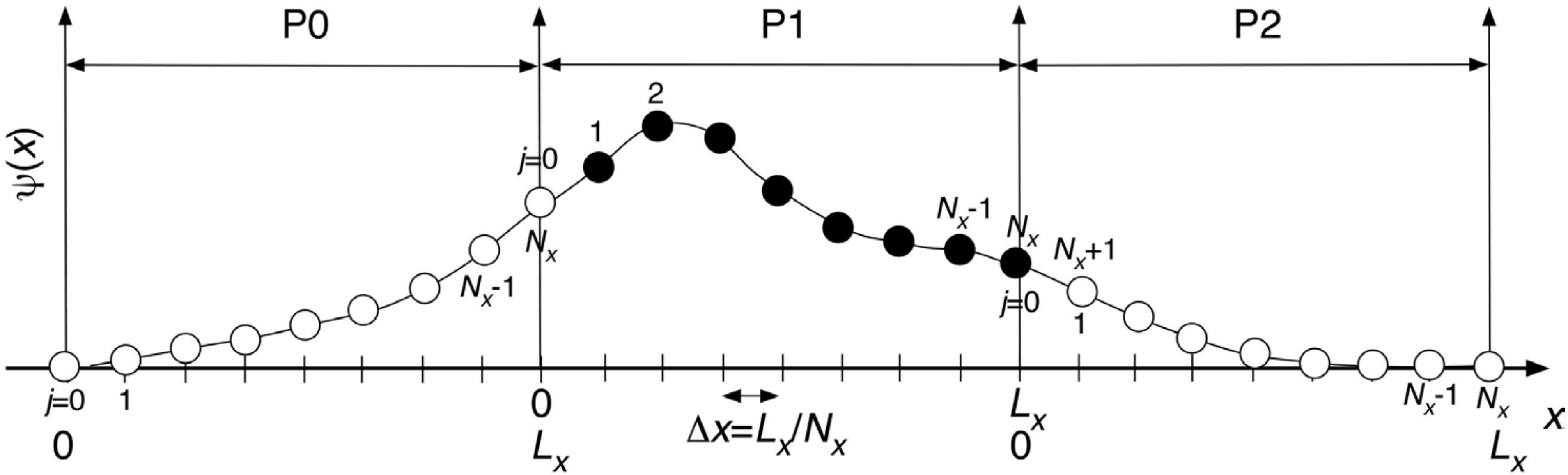
See https://aiichironakano.github.io/cs596.html

Parallel QD Communications



Parallelization of Space Splitting Method

- **Self-centric spatial decomposition**



- **Local & global coordinates**

$$\begin{cases} x_j = j\Delta x \\ x_j^{(\text{global})} = j\Delta x + pL_x \end{cases}$$

- **Global coordinates only in `init_prop()` & `init_wavefn()`**

Boundary Wave Function Caching

- **Parallelized `periodic_bc()`**

```
plw = (myid-1+nproc)%nproc; /* Lower partner process */
pup = (myid+1      )%nproc; /* Upper partner process */

/* Cache boundary wave function value at the lower end */
dbuf[0:1] ← psi[NX][0:1];
Send dbuf to pup;
Receive dbufr from plw;
psi[0][0:1] ← dbufr[0:1];

/* Cache boundary wave function value at the upper end */
dbuf[0:1] ← psi[1][0:1];
Send dbuf to plw;
Receive dbufr from pup;
psi[NX+1][0:1] ← dbufr[0:1];
```