# Quantum Dynamics Basics

In this chapter, we will simulate the dynamics of a particle, such as an electron, which follows the law of quantum mechanics [1]. Basics of the quantum-dynamics (QD) method [2-5] are described, along with corresponding data structures in program, qd.c.

## §1. Schrödinger Wave Equation

### WAVE FUNCTION

The state of an electron at time $t$ is specified by a complex-valued wave function, $\psi(\vec{r},t) = \mathrm{Re}\,\psi(\vec{r},t) + i\,\mathrm{Im}\,\psi(\vec{r},t) \in \mathbf{C}$ (where $i = \sqrt{-1}$), which is spread in the 3-dimensional space, $\vec{r} = (x,y,z) \in \mathbf{R}^3$. Given the wave function, we can calculate various physical properties such as:

- $P(\vec{r},t) = \psi^*(\vec{r},t)\psi(\vec{r},t) = |\psi(\vec{r},t)|^2 = |\mathrm{Re}\,\psi(\vec{r},t)|^2 + |\mathrm{Im}\,\psi(\vec{r},t)|^2$: The probability to find the electron at position $\vec{r}$ at time $t$.
- $\langle \vec{r}(t) \rangle = \int dx \int dy \int dz |\psi(\vec{r},t)|^2 \vec{r}$ : The expected position of the electron at time $t$.

Here, $\psi^*(\vec{r},t) = \mathrm{Re}\,\psi(\vec{r},t) - i\,\mathrm{Im}\,\psi(\vec{r},t)$ is the complex conjugate of the wave function.

***Normalization***: The electron wave function must be normalized such that the electron must be found somewhere in the entire space with probability 1, i.e.,

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

### WAVE EQUATION

The time evolution of the electron state, subjected to a time-independent, real-valued potential, $V(\vec{r})$, is described by the following partial differential equation:

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

where $\hbar$ = 1.05457×10$^{-27}$ g•cm$^2$/s is the Planck constant, $m$ = 9.10938×10$^{-28}$ g is the electron mass, and $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is the Laplacian operator.

***Dimensionless equation***: In the following, we measure length ($x$, $y$, $z$) in unit of $\hbar^2/me^2$ = 0.529177×10$^{-8}$ cm, time ($t$) in unit of $\hbar^3/me^4$ = 2.41889×10$^{-17}$ s, and energy ($V$) in unit of $me^4/\hbar^2$ = 4.35974×10$^{-11}$ g•cm$^2$/s$^2$, where $e$ = 4.80321×10$^{-10}$ esu is the electron charge in the CGS unit. Substituting

$$\begin{cases} x = \dfrac{\hbar^2}{me^2}x', y = \dfrac{\hbar^2}{me^2}y', z = \dfrac{\hbar^2}{me^2}z' \\ t = \dfrac{\hbar^3}{me^4}t' \\ V = \dfrac{me^4}{\hbar^2}V' \end{cases} \tag{3}$$

in Eq. (2), we obtain

$$i\hbar\bullet\frac{me^4}{\hbar^3}\frac{\partial}{\partial t'}\psi(\vec{r}',t')=\left[-\frac{\hbar^2}{2m}\bullet\left(\frac{me^2}{\hbar^2}\right)^2\nabla'^2+\frac{me^4}{\hbar^2}V(\vec{r}')\right]\psi(\vec{r}',t'),$$

or the dimensionless equation,

$$i\frac{\partial}{\partial t'}\psi(\vec{r}',t')=\left[-\frac{\nabla'^2}{2}+V(\vec{r}')\right]\psi(\vec{r}',t'). \tag{4}$$

In the following, we will use the dimensionless variables discussed above but omit the ´ symbol for brevity.

## TWO-DIMENSIONAL ELECTRON

As a specific example, the program qd.c simulates the time evolution of an electron confined in the 2-dimensional plane ($z$ = 0). Such electrons are common at the interface between two heterogeneous materials in semiconductor devices. The electron state is now specified by the 2-dimensional wave function, $\psi(x,y,t)$, where $0 \le x \le L_x$ and $0 \le y \le L_y$ ($L_x$ and $L_y$ are the system sizes in the $x$ and $y$ directions, respectively), and its time evolution is governed by the 2-dimensional Schrödinger equation,

$$i\frac{\partial}{\partial t}\psi(x,y,t)=H\psi(x,y,t). \tag{5}$$

In Eq. (5), the Hamiltonian operator, $H$, is defined as

$$\begin{aligned}H&=-\frac{1}{2}\frac{\partial^2}{\partial x^2}-\frac{1}{2}\frac{\partial^2}{\partial y^2}+V(x,y)\\&=T_x+T_y+V\end{aligned}. \tag{6}$$

***Boundary condition***: We impose the periodic boundary condition on the wave function such that

$$\begin{cases}\psi(x+L_x,y)=\psi(x,y)\\\psi(x,y+L_y)=\psi(x,y)\end{cases}. \tag{7}$$

## DISCRETIZATION

The wave function is discretized on a regular mesh of size $\Delta x$ and $\Delta y$ in the x and y directions, respectively. Here $\Delta x = L_x/N_x$ and $\Delta y = L_y/N_y$, where $N_x$ and $N_y$ are the numbers of mesh points in the x and y directions, respectively. We denote the discretized wave function as $\psi_{j,k} = \psi(j\Delta x, k\Delta y)$, see the figure below.

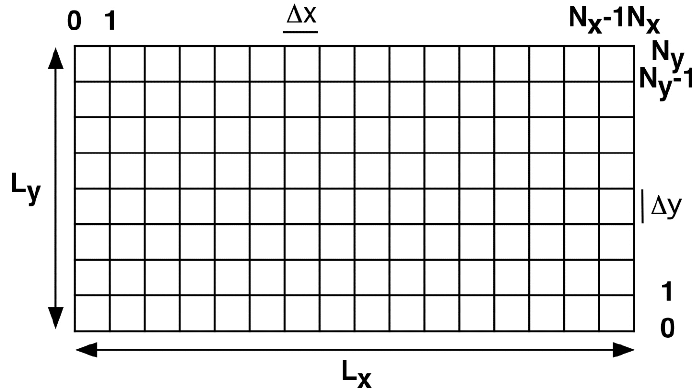


**Figure**: 2-dimensional mesh, on which the wave function is discretized.

In the discretized form, the Hamiltonian operator in Eq. (6) acts as

$$\left(H\psi\right)_{j,k} = \left(T_x\psi\right)_{j,k} + \left(T_y\psi\right)_{j,k} + \left(V\psi\right)_{j,k}, \tag{8}$$

where

$$\begin{cases} \left(T_x\psi\right)_{j,k} = -\dfrac{1}{2}\dfrac{\psi_{j-1,k} - 2\psi_{j,k} + \psi_{j+1,k}}{(\Delta x)^2} \\ \left(T_y\psi\right)_{j,k} = -\dfrac{1}{2}\dfrac{\psi_{j,k-1} - 2\psi_{j,k} + \psi_{j,k+1}}{(\Delta y)^2} \\ \left(V\psi\right)_{j,k} = V_{j,k}\psi_{j,k} \end{cases} \tag{9}$$

and the potential-energy function is discretized as $V_{j,k} = V(j\Delta x, k\Delta y)$.

Note that the discretized $H$ is a mapping from an $N_x \times N_y$ array $\psi$ ($\psi_{j,k}$ is its element at the $j$-th column and $k$-th row) to another $N_x \times N_y$ array $H\psi$. The ($j$, $k$)-th element, $(H\psi)_{j,k}$, of the output array, $H\psi$, is a linear combination of the input array with different indices, as specified in Eqs. (8) and (9).

## §2. Numerical Integration of Schrödinger Equation

The time evolution of the wave function is formally written down as

$$\psi(t + \Delta t) = \exp\left(-iH\Delta t\right)\psi(t), \tag{10}$$

where we omit the indices for simplicity. Here the exponential function of an operator is defined as a series expansion,

$$\exp\left(-iH\Delta t\right) = \sum_{n=0}^{\infty} \frac{1}{n!}\left(-iH\Delta t\right)^n. \tag{11}$$

In the split-operator method [2-5], the wave function is propagated for a small time interval, $\Delta t$, as

$$\begin{aligned} \psi(t + \Delta t) = {} & \exp\left(-iV\Delta t/2\right)\exp\left(-iT_x\Delta t\right)\exp\left(-iT_y\Delta t\right)\exp\left(-iV\Delta t/2\right)\psi(t) \\ & + O\left(\left[\Delta t\right]^3\right) \end{aligned}. \tag{12}$$

In Eq. (12), the application of the potential propagator, exp($-iV\Delta t/2$), is straightforward.

$$\begin{aligned} \left(\exp\left(-iV\Delta t/2\right)\psi\right)_{j,k} &= \psi_{j,k} - \frac{i\Delta t}{2}\left(V\psi\right)_{j,k} + \frac{\left(-i\dfrac{\Delta t}{2}\right)^2}{2!}\left(V^2\psi\right)_{j,k} + \cdots \\ &= \psi_{j,k} - \frac{i\Delta t}{2}V_{j,k}\psi_{j,k} + \frac{\left(-i\dfrac{\Delta t}{2}\right)^2}{2!}\left(V_{j,k}\right)^2\psi_{j,k} + \cdots. \\ &= \exp\left(-iV_{j,k}\Delta t/2\right)\psi_{j,k} \end{aligned} \tag{13}$$

Note that, for real number $a$,

$$\begin{aligned} \exp\left(ia\right) &= 1 + ia + \frac{1}{2!}(-a^2) + \frac{1}{3!}(-ia^3) + \frac{1}{4!}(a^4) + \frac{1}{4!}(ia^5) + \cdots \\ &= \left(1 - \frac{1}{2!}a^2 + \frac{1}{4!}a^4 + \cdots\right) + i\left(a - \frac{1}{3!}a^3 + \frac{1}{5!}a^5 + \cdots\right) \\ &= \cos(a) + i\sin(a) \end{aligned}. \tag{14}$$

Using Eq. (14) in Eq. (13),

$$\begin{aligned}\left(\exp\left(-iV\Delta t/2\right)\psi\right)_{j,k} &= \left[\cos\left(V_{j,k}\Delta t/2\right) - i\sin\left(V_{j,k}\Delta t/2\right)\right]\left[\mathrm{Re}\,\psi_{j,k} + i\,\mathrm{Im}\,\psi_{j,k}\right] \\ &= \left[\cos\left(V_{j,k}\Delta t/2\right)\mathrm{Re}\,\psi_{j,k} + \sin\left(V_{j,k}\Delta t/2\right)\mathrm{Im}\,\psi_{j,k}\right] \\ &\quad + i\left[\cos\left(V_{j,k}\Delta t/2\right)\mathrm{Im}\,\psi_{j,k} - \sin\left(V_{j,k}\Delta t/2\right)\mathrm{Re}\,\psi_{j,k}\right]\end{aligned} \quad . \tag{15}$$

Many algorithms have been proposed to apply the kinetic propagators such as exp($-iT_x\Delta t$). Among these algorithms, the space-splitting method (SSM) [4,5] is highly scalable on massively parallel computers. To understand the SSM, we first note that the operation of $T_x$ on $\psi_{j,k}$ is expressed as

$$T_x\psi_{j,k} = b\psi_{j-1,k} + 2a\psi_{j,k} + b\psi_{j+1,k} \tag{16}$$

where

$$\begin{cases} a = 1/2\left(\Delta x\right)^2 \\ b = -1/2\left(\Delta x\right)^2 \end{cases} \tag{17}$$

For each index $k$, the operation of $T_x$ on $\psi_{j,k}$ thus amounts to the multiplication of a tridiagonal matrix (note the periodic boundary condition),

$$T_x = \begin{bmatrix} 2a & b & & & & & b \\ b & 2a & b & & & & \\ & b & 2a & b & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & b & 2a & b & \\ & & & & b & 2a & b \\ b & & & & & b & 2a \end{bmatrix}. \tag{18}$$

In the SSM, this tridiagonal matrix is expressed as a direct sum of 2×2 submatrices,

$$\begin{aligned} T_x &= \begin{bmatrix} 2a & b & & & & & b \\ b & 2a & b & & & & \\ & b & 2a & b & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & b & 2a & b & \\ & & & & b & 2a & b \\ b & & & & & b & 2a \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} a & b & & & & & \\ b & a & & & & & \\ & & a & b & & & \\ & & b & a & & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix} + \begin{bmatrix} a & & & & & & b \\ & a & b & & & & \\ & b & a & & & & \\ & & & \ddots & & & \\ & & & & a & b & \\ & & & & b & a & \\ b & & & & & & a \end{bmatrix} + \frac{1}{2}\begin{bmatrix} a & b & & & & & \\ b & a & & & & & \\ & & a & b & & & \\ & & b & a & & & \\ & & & & \ddots & & \\ & & & & & a & b \\ & & & & & b & a \end{bmatrix} \end{aligned} \tag{19}$$

where we have omitted the index $k$. The exponential of a 2×2 matrix on the right-hand side of Eq. (19) is calculated analytically as follows:

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

$$\begin{bmatrix} \varepsilon_2^+ & \varepsilon_2^- & & & & & \\ \varepsilon_2^- & \varepsilon_2^+ & & & & & \\ & & \varepsilon_2^+ & \varepsilon_2^- & & & \\ & & \varepsilon_2^- & \varepsilon_2^+ & & & \\ & & & & \ddots & & \\ & & & & & \varepsilon_2^+ & \varepsilon_2^- \\ & & & & & \varepsilon_2^- & \varepsilon_2^+ \end{bmatrix} \begin{bmatrix} \varepsilon_1^+ & & & & & & \varepsilon_1^- \\ & \varepsilon_1^+ & \varepsilon_1^- & & & & \\ & \varepsilon_1^- & \varepsilon_1^+ & & & & \\ & & & \ddots & & & \\ & & & & \varepsilon_1^+ & \varepsilon_1^- & \\ & & & & \varepsilon_1^- & \varepsilon_1^+ & \\ \varepsilon_1^- & & & & & & \varepsilon_1^+ \end{bmatrix} \begin{bmatrix} \varepsilon_2^+ & \varepsilon_2^- & & & & & \\ \varepsilon_2^- & \varepsilon_2^+ & & & & & \\ & & \varepsilon_2^+ & \varepsilon_2^- & & & \\ & & \varepsilon_2^- & \varepsilon_2^+ & & & \\ & & & & \ddots & & \\ & & & & & \varepsilon_2^+ & \varepsilon_2^- \\ & & & & & \varepsilon_2^- & \varepsilon_2^+ \end{bmatrix} \tag{20}$$

where

$$\begin{cases} \varepsilon_n^+ = \dfrac{1}{2}\left[\exp\left(-\dfrac{i\Delta t}{n}(a+b)\right) + \exp\left(-\dfrac{i\Delta t}{n}(a-b)\right)\right] \\ \varepsilon_n^- = \dfrac{1}{2}\left[\exp\left(-\dfrac{i\Delta t}{n}(a+b)\right) - \exp\left(-\dfrac{i\Delta t}{n}(a-b)\right)\right] \end{cases} \tag{21}$$

The operation of exp($-iT_y\Delta t$) is executed in a similar manner.

## §3. Data Structures of qd.c

NX, NY: Number of mesh points in the x and y directions.
psi[NX+2][NY+2][2]: psi[i][j][0|1] is the real|imaginary part of the wave function on mesh point (i, j) in the xy plane.

The wave function to be simulated is in the range, 1 ≤ i ≤ NX and 1 ≤ j ≤ NY. To simplify the operation of the finite-difference operators considering the periodic boundary condition, the wave function values at the edge are duplicated as follows:

```
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];
   }
```

v[NX+2][NY+2]: v[i][j] is the potential energy at mesh point (i, j).
u[NX+2][NY+2][2]: u[i][j][0|1] is the real|imaginary part of the potential propagator at mesh point (i, j).

The potential propagator, exp($-iV\Delta t/2$), is operated in qd.c as follows, see Eq. (15).

```
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;
   }
```

The program qd.c simulates an electron incident on a potential barrier of height $B_H$ and width $B_W$, see the figure below. In addition, an edge potential of height $E_H$ is applied at i = 1 or NX or j = 1 or NY in v[i][j]. The potential v[i][j] = 0 at all the other mesh points.

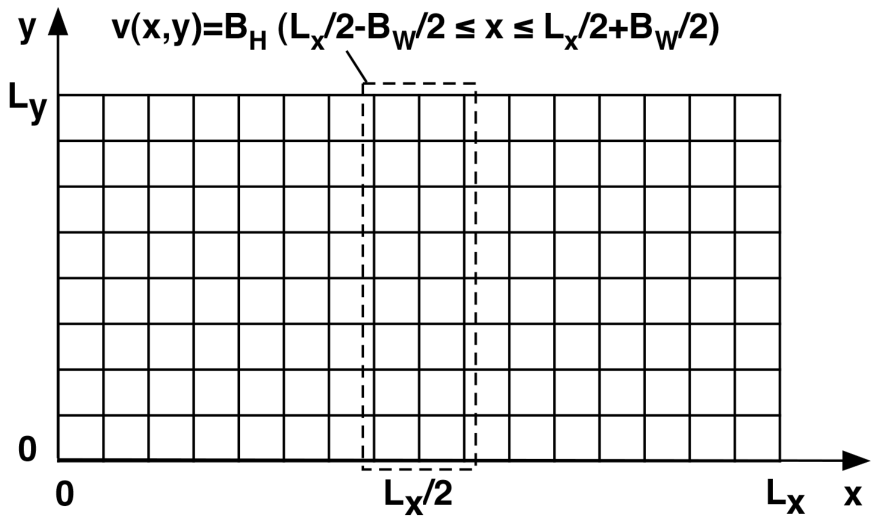


**Figure**: The potential energy function.

In classical mechanics, a particle coming from one side of the potential barrier with a higher kinetic energy than $B_H$ will pass through the barrier to the other side; otherwise, the particle will bounce back at the barrier. In quantum mechanics, a part of the electron wave function is transmitted through the barrier and the other part is reflected at the barrier.

ND: The number of spatial dimensions = 2.
al[ND][2][2]: al[0|1][0|1][0|1] is the x|y-direction, half ($\Delta t/2$)|full ($\Delta t$)-step, real|imaginary-part of the diagonal element of the kinetic propagator, see Eqs. (20) and (21).
bux|y[2][NX+2|Y+2][2]: bux|y[0|1][i][0|1] is the x|y-direction, half ($\Delta t/2$)|full ($\Delta t$)-step, real|imaginary-part of the upper off-diagonal kinetic propagator on mesh i, see Eqs. (20) and (21).
blx|y[2][NX+2|Y+2][2]: blx|y[0|1][i][0|1] is the x|y-direction, half ($\Delta t/2$)|full ($\Delta t$)-step, real|imaginary-part of the lower off-diagonal kinetic propagator on mesh i, see Eqs. (20) and (21).

The 2×2 block-diagonal form of the kinetic propagator, Eq. (20), can be handled

$$\left(U_x^{(\mathrm{half})}\psi\right)_{i,j} = \varepsilon_2^- \delta_{\mathrm{mod}(i,2),0}\psi_{i-1,j} + \varepsilon_2^+ \psi_{i,j} + \varepsilon_2^- \delta_{\mathrm{mod}(i,2),1}\psi_{i+1,j} \tag{22}$$

$$\left(U_x^{(\mathrm{full})}\psi\right)_{i,j} = \varepsilon_1^- \delta_{\mathrm{mod}(i,2),1}\psi_{i-1,j} + \varepsilon_1^+ \psi_{i,j} + \varepsilon_1^- \delta_{\mathrm{mod}(i,2),0}\psi_{i+1,j} \tag{23}$$

where $\delta_{\mathrm{mod}(i,2),0} = 1$ (if mod($i$, 2) = 0) and 0 (else), *etc*. The above kinetic propagator is used in qd.c to update the wave function as follows, where d (= 0 for x; 1 for y) is the direction and t (= 0 for $\Delta t/2$—half; 1 for $\Delta t$—full) is the time step:

```
/* 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

$$\psi(x,y,t=0) = C\exp\left(-\frac{(x-x_0)^2}{4\sigma^2}\right)\exp\left(ik_0x\right)\sin\left(\frac{\pi y}{L_y}\right) \tag{24}$$

To understand the meaning of this wave function, consider

$$\psi(x,y) = C\exp\left(ik_0x\right), \tag{25}$$

where $C = \sqrt{L_xL_y}$ is the normalization constant such that

$$\int_0^{L_x} dx \int_0^{L_y} dy\,|\,\psi(x,y)\,|^2 = \int_0^{L_x} dx \int_0^{L_y} dy C^2\left(\cos^2(k_0x)+\sin^2(k_0x)\right) = 1\,. \tag{26}$$

Then

$$\begin{aligned} H\psi(x,y) &= -\frac{1}{2}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)C\exp\left(ik_0x\right) \\ &= -\frac{C}{2}\frac{d^2}{dx^2}\exp\left(ik_0x\right) \\ &= -\frac{ik_0C}{2}\frac{d}{dx}\exp\left(ik_0x\right) \\ &= \frac{k_0^2C}{2}\exp\left(ik_0x\right) = \frac{k_0^2}{2}\psi(x,y) \end{aligned}, \tag{27}$$

and thus the expectation value of its energy is

$$\begin{aligned} E_0 &= \int_0^{L_x} dx \int_0^{L_y} dy \psi^*(x,y) H\psi(x,y) \\ &= \int_0^{L_x} dx \int_0^{L_y} dy \psi^*(x,y) \frac{k_0^2}{2}\psi(x,y)\,. \\ &= \frac{k_0^2}{2} \end{aligned} \tag{28}$$

or $k_0 = \sqrt{2E_0}$ .

We can show that the following ‘traveling’ wave function is a solution to the time-dependent Schrödinger equation, if the potential energy function is 0:

$$\psi(x,y,t) = C\exp\left(ik_0\left(x - v_0t\right)\right), \tag{29}$$

where $v_0 = E_0 / k_0$ is the wave speed.

The last factor in Eq. (24) also satisfies the Schrödinger equation and follows the boundary condition, $\psi(x,0) = \psi(x,L_y) = 0$, which is required if there is a very high potential barrier at $y = 0$ and $L_y$ (the electron then cannot exist). Finally, the first Gaussian factor in Eq. (24) acts to localize the wave function around $x = x_0$ and spread $\sigma$.

## References

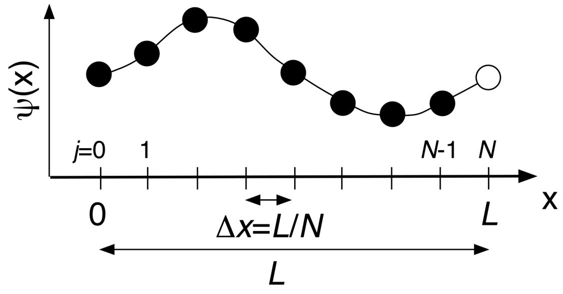
1. L. I. Schiff, *Quantum Mechanics, 3rd Ed.* (McGraw-Hill, 1968).
2. S. E. Koonin, *Computational Physics* (Benjamin-Cummings, 1986).
3. H. De Raedt, "Product formula algorithms for solving the time dependent Schrödinger equation," *Computer Physics Reports* **7**, 1 (1987).
4. J. L. Richardson, "Visualizing quantum scattering on the CM-2 supercomputer," *Computer Physics Communications* **63**, 84 (1991).
5. A. Nakano, P. Vashishta, and R. K. Kalia, "Massively parallel algorithms for computational nanoelectronics based on quantum molecular dynamics," *Computer Physics Communications* **83**, 181 (1994).

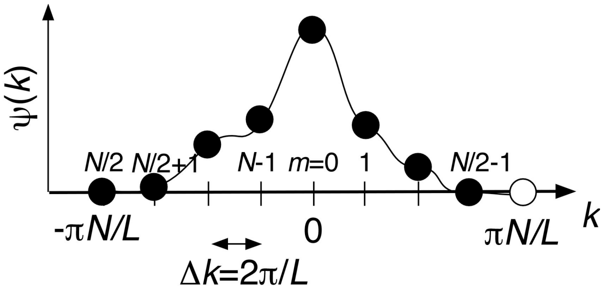
# Quantum Dynamics Basics II—Spectral Method

In this chapter, we will solve the time-dependent Schrödinger equation using another numerical technique, i.e., the spectral method, which is based on Fourier transformation.

## §1. Discrete Fourier Transform

Consider a complex-valued function, $\psi(x) \in \mathbf{C}$, in the range, $x \in [0, L]$. We assume the periodic boundary condition: $\psi(x + L) = \psi(x)$. Let us discretize $\psi(x)$ on $N$ mesh points, $x_j = j\Delta x$ ($j = 0, ..., N$-1), with equal mesh spacing, $\Delta x = L/N$ (see the left figure below). We denote the discrete function values as $\psi_j = \psi(x_j)$.





Discrete Fourier transformation represents $\psi(x)$ as a linear combination of trigonometric functions, $\exp(ikx) = \cos(kx) + i\sin(kx)$, with different wave numbers, $k$:

$$\psi_j = \sum_{m=0}^{N-1} \tilde{\psi}_m \exp\left(ik_m x_j\right), \tag{1}$$

where the discrete wave numbers, $k_m$, are defined as

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

and the expansion coefficients are given by

$$\tilde{\psi}_m = \frac{1}{N}\sum_{j=0}^{N-1} \psi_j \exp\left(-ik_m x_j\right). \tag{3}$$

Note that the choice of wave numbers in Eq. (2) guarantees that $\psi(x)$ has the periodicity of $L$. Also, because of the discrete sampling in the real space, wave numbers separated by $2\pi nN/L = 2\pi n/\Delta x$ ($n = \pm1, \pm2, ...$) are all equivalent. (Higher wave numbers oscillate more, but come back to the same value as their lower wave number counterparts at $x_j$.) Among these equivalent wave numbers, we use the smallest-magnitude wave number, since physically it represents the lowest-energy state and, mathematically, the discrete mesh points in the real space cannot represent higher wave numbers. Accordingly, in Eq. (2), the wave numbers for the higher indices, $m = N/2, N/2+1, ..., N$-1, are folded back by $2\pi N/L$, so that all the wave numbers are in the range, $[-\pi/\Delta x, \pi/\Delta x]$, see the right figure above. (For simplicity, we assume that $N$ is an even number.)

To prove the correctness of the above Fourier expansion, it is convenient to think the discrete function, $\psi_j$, as a vector in the $N$-dimensional vector space: $\left|\psi\right\rangle = \left(\psi_0, \psi_1, \ldots, \psi_{N-1}\right)$. In this vector space, we define the plane-wave basis set,

$$\left\{ \left|m\right\rangle = b_m\left(x_j\right) = \frac{1}{\sqrt{N}}\exp(ik_m x_j) \middle| m = 0,1,\ldots,N-1 \right\}, \tag{4}$$

which is **orthonormal**, i.e., the inner products of the basis functions are

$$\left\langle m \middle| n \right\rangle \equiv \sum_{j=0}^{N-1} b_m^*\left(x_j\right) b_n\left(x_j\right) = \delta_{m,n} = \left\{ \begin{array}{ll} 1 & m = n \\ 0 & m \neq n \end{array} \right. . \tag{5}$$

---

∵ For $m \neq n$, carry out the sum of geometric series; otherwise ($m = n$), all $N$ summands are $1/N$.

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

$$= \left\{ \begin{array}{ll} \dfrac{1}{N}\dfrac{\exp\left(i2\pi\left(n-m\right)\right)-1}{\exp\left(i\dfrac{2\pi}{N}\left(n-m\right)\right)-1} = 0 & m \neq n \\ \dfrac{1}{N}\bullet N = 1 & m = n \end{array} \right. \quad .//$$

---

The above basis set is also **complete**, i.e., any discrete function, $\psi_j$, in this $N$-dimensional vector space can be represented as a linear combination of $N$ basis set functions, $b_m(x_j)$. Specifically,

$$\left|\psi\right\rangle = \sum_{m=0}^{N-1}\left|m\right\rangle\left\langle m \middle| \psi\right\rangle, \tag{6}$$

or

$$1 = \sum_{m=0}^{N-1}\left|m\right\rangle\left\langle m\right|. \tag{7}$$

---

∵ Suppose the function is expanded as

$$\left|\psi\right\rangle = \sum_{n=0}^{N-1} c_n\left|n\right\rangle.$$

Multiplying both sides by $\left\langle m\right|$ and using the orthonormality, Eq. (5), we get $\left\langle m \middle| \psi\right\rangle = c_m$ .//

---

The Fourier coefficients, $\tilde{\psi}_m$, in Eq. (3) are readily obtained from Eq. (6). Substituting the definitions of the basis functions and the inner product in Eq. (6), we obtain

$$\psi_j = \sum_{m=0}^{N-1}\exp\left(ik_m x_j\right)\frac{1}{N}\sum_{l=0}^{N-1}\exp\left(-ik_m x_l\right)\psi_l .$$

Comparing this equation with Eq. (1) identifies the expansion coefficients, $\tilde{\psi}_m$, in Eq. (1) as Eq. (3).

## §2. Spectral Method for Integrating Time-Dependent Schrödinger Equation

### HAMILTONIAN OPERATOR

Consider the time-dependent Schrödinger equation in one dimension in atomic unit,

$$i\frac{\partial}{\partial t}\psi(x,t) = H\psi(x,t), \tag{8}$$

where the Hamiltonian operator, $H$, is defined as

$$\begin{aligned} H &= -\frac{1}{2}\frac{\partial^2}{\partial x^2} + V(x) \\ &= T + V \end{aligned}, \tag{9}$$

with $T$ and $V$ being the kinetic- and potential-energy operators, respectively.

The kinetic energy operator is diagonal in the Fourier (or momentum) space. To see this, we operate $T$ on the wave function in its Fourier representation, Eq. (1):

$$-\frac{1}{2}\frac{\partial^2}{\partial x^2}\sum_{m=0}^{N-1}\tilde{\psi}_m \exp(ik_m x) = \sum_{m=0}^{N-1}\frac{k_m^2}{2}\tilde{\psi}_m \exp(ik_m x), \tag{10}$$

i.e., the kinetic energy operator multiplies the factor, $k_m^2/2$, to the Fourier coefficient of the wave function:

$$\tilde{\psi}_m \xrightarrow[T]{} \frac{k_m^2}{2}\tilde{\psi}_m .$$

Recall, on the other hand, the potential energy operator is diagonal in the real space, i.e., it multiplies the factor, $V_j = V(x_j)$ to the wave function:

$$\psi_j \xrightarrow[V]{} V_j \psi_j .$$

## SPLIT-OPERATOR TECHNIQUE AND SPECTRAL METHOD

The above observation, that the kinetic- and potential-energy operators are diagonal in the real- and momentum-spaces, respectively, suggests an efficient algorithm for the time evolution of the wave function. Recall the Trotter expansion (also called the split-operator technique):

$$\psi(x,t+\Delta t) = \exp(-iV\Delta t/2)\exp(-iT\Delta t)\exp(-iV\Delta t/2)\psi(x,t) + O\left([\Delta t]^3\right). \tag{11}$$

The time evolution operator, $\exp(-iV\Delta t/2)$, arising from the potential energy, is easily operated in the real space,

$$\exp(-iV\Delta t/2)\psi_j = \exp(-iV_j\Delta t/2)\psi_j , \tag{12}$$

or

$$\psi_j \xrightarrow[\exp(-iV\Delta t/2)]{} \exp(-iV_j\Delta t/2)\psi_j .$$

On the other hand, the time evolution operator, $\exp(-iT\Delta t)$, arising from the kinetic energy, is operated in the Fourier space as

$$\exp(-iT\Delta t)\tilde{\psi}_m = \exp(-ik_m^2\Delta t/2)\tilde{\psi}_m , \tag{13}$$

or

$$\tilde{\psi}_m \xrightarrow[\exp(-iT\Delta t)]{} \exp(-ik_m^2\Delta t/2)\tilde{\psi}_m .$$

The spectral method is formally represented in terms of the forward and inverse Fourier transformation operators, $F$ and $F^{-1}$,

$$\psi_j \xrightarrow[F^{-1}]{} F^{-1}\psi_j = \tilde{\psi}_m = \frac{1}{N}\sum_{j=1}^{N}\psi_j \exp(-ik_m x_j), \tag{14}$$

$$\tilde{\psi}_m \xrightarrow[F]{} F\tilde{\psi}_m = \psi_j = \sum_{m=1}^{N} \tilde{\psi}_m \exp\left(ik_m x_j\right), \tag{15}$$

as follows

$$\psi(t+\Delta t) = \exp\left(-iV\Delta t/2\right) F \exp\left(-iT\Delta t\right) F^{-1} \exp\left(-iV\Delta t/2\right)\psi(t)\,. \tag{16}$$

Equation (16) amounts to the following algorithm.

**Spectral Split-Operator Algorithm**

```
    1. $\psi_j \leftarrow \exp\left(-iV_j\Delta t/2\right)\psi_j$
    2. $\tilde{\psi}_m \xleftarrow[F^{-1}]{} \psi_j$
    3. $\tilde{\psi}_m \leftarrow \exp\left(-ik_m^2\Delta t/2\right)\tilde{\psi}_m$
    4. $\psi_j \xleftarrow[F]{} \tilde{\psi}_m$
    5. $\psi_j \leftarrow \exp\left(-iV_j\Delta t/2\right)\psi_j$
```

## COMPUTATION OF THE ENERGY

The total energy is a conserved quantity for the time-dependent Schrödinger equation, Eq. (8), and is useful for estimating the discretization error. The total energy can be calculated as follows:

$$\begin{aligned}\langle H\rangle &= \langle T\rangle + \langle V\rangle \\ &= \int dx\psi^*(x)\left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right)\psi(x) + \int dx\psi^*(x)V(x)\psi(x)\,. \\ &\cong dx\sum_{j=0}^{N-1}\psi_j^*\left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right)\psi_j + dx\sum_{j=0}^{N-1}V_j\left|\psi_j\right|^2\end{aligned} \tag{17}$$

To calculate the first term (i.e., the kinetic energy) in Eq. (17), let us expand the wave function in terms of its Fourier components as in Eq. (1):

$$\begin{aligned}\langle T\rangle &= dx\sum_{j=0}^{N-1}\sum_{m=0}^{N-1}\tilde{\psi}_m^*\exp\left(-ik_m x_j\right)\left(-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right)\sum_{n=0}^{N-1}\tilde{\psi}_n\exp\left(ik_n x_j\right) \\ &= dx\sum_{j=0}^{N-1}\sum_{m=0}^{N-1}\tilde{\psi}_m^*\exp\left(-ik_m x_j\right)\sum_{n=0}^{N-1}\frac{k_n^2}{2}\tilde{\psi}_n\exp\left(ik_n x_j\right) \\ &= dx\sum_{m=0}^{N-1}\tilde{\psi}_m^*\sum_{n=0}^{N-1}\tilde{\psi}_n\sum_{j=0}^{N-1}\frac{k_n^2}{2}\exp\left(i\left(k_n-k_m\right)x_j\right) \qquad . \\ &= dx\sum_{m=0}^{N-1}\tilde{\psi}_m^*\sum_{n=0}^{N-1}\frac{k_n^2}{2}\tilde{\psi}_n N\delta_{m,n} \\ &= dxN\sum_{m=0}^{N-1}\frac{k_m^2}{2}\left|\tilde{\psi}_m\right|^2\end{aligned} \tag{18}$$

By substituting Eq. (18) in (17), we obtain

$$\begin{aligned}\langle H\rangle &= \langle T\rangle + \langle V\rangle \\ &= dxN\sum_{m=0}^{N-1}\frac{k_m^2}{2}\left|\tilde{\psi}_m\right|^2 + dx\sum_{j=0}^{N-1}V_j\left|\psi_j\right|^2\,.\end{aligned} \tag{19}$$

### CONSERVATION OF THE TOTAL ENERGY

Let's consider the temporal change of the total energy $\langle H\rangle$. It is convenient to introduce the eigenvalues, $\varepsilon_n$, and eigenvectors, $|n\rangle$, of the $N\times N$ Hamiltonian in the $N$-dimensional vector space:

$$H|n\rangle = \varepsilon_n|n\rangle \quad (n = 0,\ldots,N-1) \ .$$

The wave function is then expanded with the orthonormal basis set consisting of the energy eigenvectors as

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

and accordingly the expectation value of the total energy at time $t$ is

$$\begin{aligned}
\langle H(t)\rangle &= \langle\psi(t)|H|\psi(t)\rangle \\
&= \left(\sum_{m=0}^{N-1}\langle\psi(0)|m\rangle\exp\left(i\varepsilon_m t\right)\langle m|\right)H\left(\sum_{n=0}^{N-1}\exp\left(-i\varepsilon_n t\right)|n\rangle\langle n|\psi(0)\rangle\right) \\
&= \left(\sum_{m=0}^{N-1}\langle\psi(0)|m\rangle\exp\left(i\varepsilon_m t\right)\langle m|\right)\left(\sum_{n=0}^{N-1}\exp\left(-i\varepsilon_n t\right)\varepsilon_n|n\rangle\langle n|\psi(0)\rangle\right) \\
&= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\exp\left(i(\varepsilon_m-\varepsilon_n)t\right)\varepsilon_n\langle\psi(0)|m\rangle\langle n|\psi(0)\rangle\langle m|n\rangle \\
&= \sum_{n=0}^{N-1}\varepsilon_n\left|\langle n|\psi(0)\rangle\right|^2 = \text{constant}
\end{aligned}$$

i.e., the total energy is time invariant. Here we have used the orthonormality, $\langle m|n\rangle = \delta_{m,n}$.

## §3. Fast Fourier Transform

The bottleneck in implementing the above spectral method is the computational cost associated with the discrete Fourier transform. Since the computation of each of the $N$ Fourier coefficients, $\tilde{\psi}_m$, involves summation over $N$ terms, the computational time grows as $O(N^2)$. The fast Fourier transform (FFT) algorithm[1] reduces this complexity to $O(N\log N)$, and makes the quantum-dynamics simulation less compute-intensive. The discussion in this lecture note follows Chapter 12 in the *Numerical Recipes*.[2] First, download and read sections 12.1 "Fourier Transform of Discretely Sampled Data" and 12.2 "Fast Fourier Transform" at http://www.library.cornell.edu/nr/bookcpdf.html.

### DANIELSON-LANCZOS ALGORITHM

The summation in the Fourier transform can be split into two partial sums as follows:

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput. **19**, 297 (1965).

[2] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C, 2nd Ed.* (Cambridge U Press, 1993).

$$\begin{aligned}\psi_j &= \sum_{m=0}^{N-1} \tilde{\psi}_m \exp\left(ik_m x_j\right) = \sum_{m=0}^{N-1} \tilde{\psi}_m \exp\left(i2\pi mj/N\right) \\ &= \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m} \exp\left(i2\pi(2m)j/N\right) + \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m+1} \exp\left(i2\pi(2m+1)j/N\right) \\ &= \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m} \exp\left(i2\pi mj/(N/2)\right) + \exp\left(i2\pi j/N\right) \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m+1} \exp\left(i2\pi mj/(N/2)\right)\end{aligned}.$$

Therefore,

$$\psi_j = \psi_j^0 + W_N^j \psi_j^1, \tag{20}$$

where

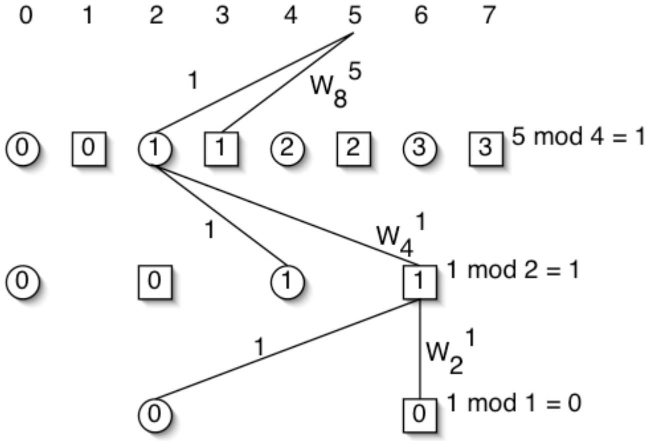
$$\begin{cases} \psi_j^0 = \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m} \exp\left(i2\pi mj/(N/2)\right) \\ \psi_j^1 = \sum_{m=0}^{N/2-1} \tilde{\psi}_{2m+1} \exp\left(i2\pi mj/(N/2)\right) \\ W_N = \exp\left(i2\pi/N\right) \end{cases}. \tag{21}$$

Note that $\psi_j^0$ and $\psi_j^1$ represent *N*/2-element Fourier transforms consisting of even and odd sub-arrays, respectively. Accordingly, in the sub-array Fourier components, $\psi_j^0$ and $\psi_j^1$, *j* should be read as *j* mod (*N*/2), i.e., an index in a *N*/2-long Fourier transform.

The top part of the figure below illustrates this decomposition due to Danielson and Lanczos for an 8-element Fourier transform, specifically for

$$\psi_5 = \psi_{5\,\mathrm{mod}\,4=1}^0 + W_8^5 \psi_{5\,\mathrm{mod}\,4=1}^1,$$

where circles and squares denote the even and odd sub-arrays, respectively.



In a similar manner, each *N*/2-element Fourier transform is further decomposed into two *N*/4-element Fourier transforms, e.g.,

$$\begin{aligned}\psi_j^0 &= \sum_{m=0}^{N/4-1} \tilde{\psi}_{2(2m)} \exp\left(i2\pi(2m)j/(N/2)\right) + \sum_{m=0}^{N/4-1} \tilde{\psi}_{2(2m+1)} \exp\left(i2\pi(2m+1)j/(N/2)\right) \\ &= \sum_{m=0}^{N/4-1} \tilde{\psi}_{2(2m)} \exp\left(i2\pi mj/(N/4)\right) + \exp\left(i2\pi j/(N/2)\right) \sum_{m=0}^{N/4-1} \tilde{\psi}_{2(2m+1)} \exp\left(i2\pi mj/(N/4)\right)\end{aligned}.$$

Therefore,

$$\psi_j^0 = \psi_j^{00} + W_{N/2}^j \psi_j^{01}, \tag{22}$$

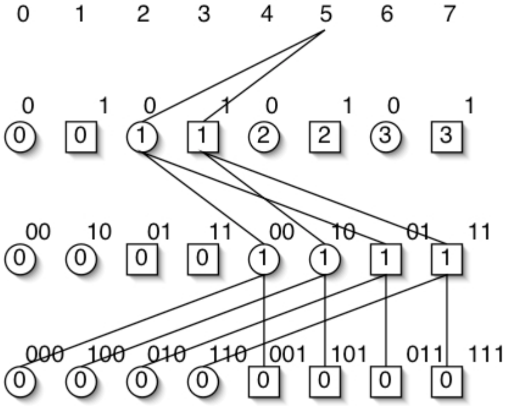
where

$$\begin{cases} \psi_j^{00} = \sum_{m=0}^{N/4-1} \tilde{\psi}_{2(2m)} \exp\left(i2\pi mj/(N/4)\right) \\ \psi_j^{01} = \sum_{m=0}^{N/4-1} \tilde{\psi}_{2(2m+1)} \exp\left(i2\pi mj/(N/4)\right) \\ W_{N/2} = \exp\left(i2\pi/(N/2)\right) \end{cases}. \tag{23}$$

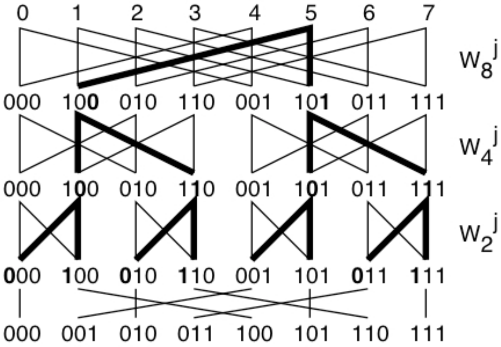
In $\psi_j^{00}$ and $\psi_j^{01}$ (see Eq. (23)), $j$ should be read as $j$ mod ($N$/4).

The decomposition of Fourier transform into sums of two sub-array Fourier transforms, e.g., Eqs. (20) and (22), continues recursively. The figure above illustrates how the even (circles) and odd (squares) sub-array Fourier components are combined. Eventually, the sub-array contains only one element, at which stage the recursion terminates, and the sub-array Fourier components is a function value at some grid point. For example, in the figure above, the bottom circle and square represent $\psi_j^{010}$ and $\psi_j^{011}$, which turn out to be $\tilde{\psi}_2$ and $\tilde{\psi}_6$, respectively. In general, we can obtain the wave function index from the bit sequence to specify the recursive sub-array by reversing the bit sequence and converting it to decimal. This works because successive subdivisions of the data into even and odd are tests of successive low-order (least significant) bits of the wave function index.

The right figure below shows all the wave functions that participate in the construction of $\psi_5$, and how they are combined to construct $\psi_5$.



In the FFT algorithm, the input wave function values are first re-ordered by applying the bit-reversal operation to each wave function index. The Danielson-Lanczos procedures, such as Eqs. (20) and (22), are then applied recursively, starting from the smaller sub-arrays up. The figure below shows all the combinations of sub-array Fourier coefficients to construct all the Fourier components in the bit-reversed scheme, in which the combinations to construct $\psi_5$ are represented by bold lines.



The figure above shows that, to compute all *N* Fourier transforms, the sub-array Fourier transforms can be re-used. Consequently, there are *N* complex multiplications and *N* complex additions at each recursive step. (Note that the Danielson-Lanczos procedure, e.g., in Eq. (20) and (22), involves one multiplication and one addition, and is represented by two lines in the figure above.) To compute all the Fourier components, every array element is connected to two (even and odd) sub-array elements at each recursive step. Since there are $\log_2 N$ recursive steps, the number of complex floating-point operations in the FFT algorithm is 2 $\log_2 N$.

The program, four1(double data[], unsigned long nn, int isign), in *Numerical Recipes in C* implements the above algorithm. On input, the data[] array contains 2*nn elements that represent nn complex function values, such that data[2*j-1] and data[2*j] (j = 1, ..., nn) are the real and imaginary parts of the function value on the j-th grid point. If isign = 1, four1() performs the Fourier transform,

$$data_j \leftarrow \sum_{m=0}^{N-1} data_m \exp\left(i2\pi mj / N\right),$$

and, on output, data[] contains the transformed function values. Else if isign = −1, four1() performs a part of inverse Fourier transform,

$$data_m \leftarrow \sum_{j=0}^{N-1} data_j \exp\left(-i2\pi mj / N\right),$$

without dividing the result by nn. To complete the inverse Fourier transform, the caller of the four1() function needs to divide the resulting data[] array by nn.

In your 1D quantum dynamics program, you may define

double psi[2*N],

where psi[2*j] and psi[2*j+1] (j = 0, ..., N−1) are the real and imaginary parts of the wave function on the j-th grid point. Since four1() expects the index to start from 1, instead of 0 in the above psi[] array, we need to call four1() with psi-1 as the first argument. (Note in C, the array name is a pointer to its first element.) The following shows typical calls to four1() in your quantum dynamics program:

```
/* 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;
```