

Note on Quantum Monte Carlo (QMC) Programming

QMC Recap: Solving Imaginary-Time Schrödinger Equation by Random Walk

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

$$\downarrow \tau \equiv it$$

$$i\hbar \frac{\partial}{\partial \tau} \psi = \left[-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right] \psi$$

$$\therefore \frac{\partial}{\partial \tau} \psi = \left[\frac{\hbar}{2m} \frac{\partial^2}{\partial x^2} - \frac{V(x)}{\hbar} \right] \psi$$

By introducing population control, V_{ref} ,

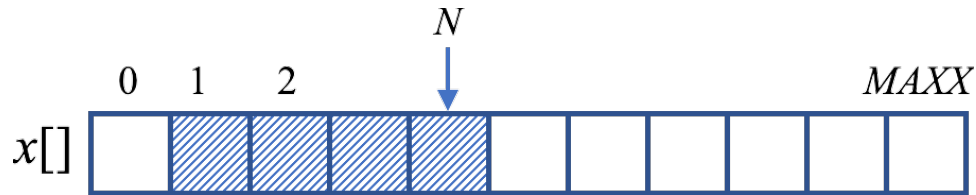
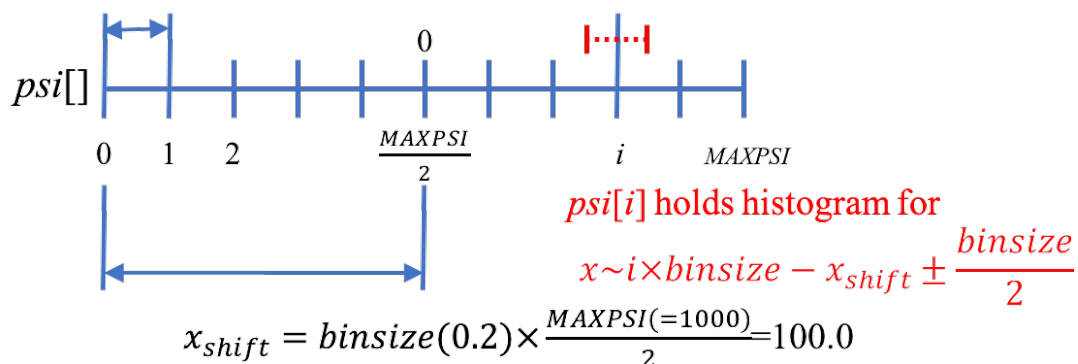
$$\underbrace{\frac{\partial}{\partial \tau} \psi}_{\text{population-change rate}} = \left[\underbrace{\frac{\hbar}{2m}}_{D = \frac{1}{2} [\text{a.u.}] = \frac{(ds)^2}{2dt} \text{ diffusion}} \frac{\partial^2}{\partial x^2} + \underbrace{\frac{V_{\text{ref}} - V(x)}{\hbar}}_{\text{birth/death rate}} \right] \psi$$

Data Structures

```

int N // Number of random walkers
double x[MAXX+1] // x[i] (i = 1,...,N) is the position of the i-th random walker (MAXX = 2000)
double psi[MAXPSI+1] // Histogram of random walkers (MAXPSI = 1000)

```


$$binsize = 2ds = 0.2$$


Algorithm

```

N ← N0 = 50 // Initialize the number of random walkers to the desired value
x[1:N] ← uniform random number in the range [-1,1] // 2rand()/(double)RAND_MAX-1
Vref ←  $\frac{1}{N} \sum_{i=1}^N V(x[i])$  // Initial reference energy, where  $V(x) = \frac{x^2}{2}$  is the harmonic potential
Reset the histogram, psi[0:MAXPSI] ← 0
for step = 1 to nequil = 0.4 × mcs = 0.4 × 500 = 200 // Equilibrate random walkers
    walk()
for step = 1 to mcs = 500 // Main MC loop for sampling
    walk()
    Add the N random walkers' positions to the histogram, psi[]

```

Function walk(): Random Walk with Birth/Death

```

Nin ← N // Number of walkers at the beginning of this MC step
Vsum ← 0.0 // Reset the accumulator to sum the potential energies of walkers
for (i=Nin; i>=1; --i) // In descending order to handle birth/death
    // Random walk by step ds = 0.1
    if (rand()%2 == 0)
        x[i] += ds
    else
        x[i] -= ds
    // Birth or death of walkers
    potential ← V(x[i]) = x2/2
    dV ← potential - Vref
    if (dV < 0) // Check whether to duplicate the walker
        if (rand()/(double)RAND_MAX < -dV × dt) // Duplicate the walker with probability -dV × dt
            ++N
            x[N] = x[i] // Clone a new walker at the same position
            Vsum += 2 × potential // Factor 2 since two walkers at the same position
        else
            Vsum += potential // Only one walker
    else // Check whether to remove the walker
        if (rand()/(double)RAND_MAX < dV × dt) // Remove the walker with probability dV × dt
            x[i] ← x[N] // Fill the gap created by death
            --N
        else
            Vsum += potential // The walker survived
Vaverage ← Vsum/N
Vref ← Vaverage -  $\frac{N-N_0}{N_0 \times dt}$  // New reference energy; note dt = ds2 = 0.01

```

