# Introduction to CUDA Parallel Programming
# CUDA平行計算導論

https://ceiba.ntu.edu.tw/1092Phys8061_CUDA

Professor Ting-Wai Chiu (趙挺偉)
Email: twchiu@phys.ntu.edu.tw
Physics Department
National Taiwan University

# This lecture will cover:

- LAPACK –Linear Algebra Package http://www.netlib.org/lapack/
- MAGMA – GPU accelerated LAPACK
  https://developer.nvidia.com/magma
- Examples
  dsyevd – eigenproblem of a real and symmetric matrix
  zheevd – eigenproblem of a Hermitian matrix
- Physical Applications
  Laplacian operator on the 1-dimensional lattice
  -- Schrödinger equation, Particle in a box
  -- A chain of mass-spring system with fixed ends
  Momentum operator on the 1-dimensional lattice

# LAPACK –Linear Algebra Package

## http://www.netlib.org/lapack/

LAPACK is written in Fortran 90 and provides routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. The associated matrix factorizations (LU, Cholesky, QR, SVD, Schur, generalized Schur) are also provided, as are related computations such as reordering of the Schur factorizations and estimating condition numbers. Dense and banded matrices are handled, but not general sparse matrices. In all areas, similar functionality is provided for real and complex matrices, in both single and double precision.

LAPACK routines are written so that as much as possible of the computation is performed by calls to the Basic Linear Algebra Subprograms (BLAS). LAPACK is designed at the outset to exploit the Level 3 BLAS — a set of specifications for Fortran subprograms that do various types of matrix multiplication and the solution of triangular systems with multiple right-hand sides. Because of the coarse granularity of the Level 3 BLAS operations, their use promotes high efficiency on many high-performance computers, particularly if specially coded implementations are provided by the manufacturer.

# MAGMA – GPU accelerated LAPACK

https://developer.nvidia.com/magma

MAGMA (Matrix Algebra on GPU and Multicore Architectures) is a collection of next generation linear algebra libraries for heterogeneous architectures. MAGMA is designed and implemented by the team that developed LAPACK and ScaLAPACK, incorporating the latest developments in hybrid synchronization- and communication-avoiding algorithms, as well as dynamic runtime systems. Interfaces for the current LAPACK and BLAS standards are supported to allow computational scientists to seamlessly port any linear algebra reliant software components to heterogeneous architectures. MAGMA allows applications to fully exploit the power of current heterogeneous systems of multi/many-core CPUs and multi-GPUs to deliver the fastest possible time to accurate solution within given energy constraints

Here we focus on the routines: **magma_dsyevd** and **magma_zheevd**

# dsyevd – eigenproblem of a real and symmetric matrix

```
#include “magma_v2.h”
#include “magma_lapack.h”
...
int main(int argc, char **argv)
{
  magma_init();                          // initialize Magma
  ...
  magma_dmalloc_cpu(&w1, n);             // eigenvalues
  magma_dmalloc_cpu(&a, n2);             // eigenvectors, n2=n*n
  ...
  // Query for workspace sizes
  double aux_work[1];
  magma_int_t aux_iwork[1];
  magma_dsyevd(MagmaVec,MagmaLower,n,r,n,w1,aux_work,-1,aux_iwork,-1, &info);
  lwork  = (magma_int_t )aux_work[0];
  liwork =  aux_iwork[0];
  iwork  = (magma_int_t *)malloc(liwork * sizeof(magma_int_t));
  magma_dmalloc_cpu(&h_work, lwork);    // memory for workspace
  ...
// compute the eigenvalues and eigenvectors of a real and symmetric nxn matrix
  magma_dsyevd(MagmaVec,MagmaLower,n,r,n,w1,h_work,lwork,iwork,liwork, &info);
  ...
  magma_finalize();                       // finalize Magma
}
// complete code at twqcd80:/home/cuda_lecture_2021/maga/dsyevd/magma_dsyevd.c
```

# zheevd – eigenproblem of a Hermitian matrix

```
#include “magma_v2.h”
#include “magma_lapack.h”
...
int main(int argc, char **argv)
{
  magma_init();                          // initialize Magma
  ...
  magma_dmalloc_cpu(&w1, n);             // eigenvalues
  magma_dmalloc_cpu(&a, n2*2);           // eigenvectors, n2=n*n
  ...
  double aux_work[2];  // Query for workspace sizes
  double aux_rwork[1];
  magma_int_t aux_iwork[1];
  magma_zheevd(MagmaVec, MagmaLower, n, (magmaDoubleComplex *)r, n,
               w1, (magmaDoubleComplex *)aux_work, -1, aux_rwork, -1,
               aux_iwork, -1, &info);
  lwork  = (magma_int_t)aux_work[0];
  lrwork = (magma_int_t)aux_rwork[0];
  liwork =  aux_iwork[0];
  iwork  = (magma_int_t *)malloc(liwork * sizeof(magma_int_t));
  magma_dmalloc_cpu(&h_work, lwork*2);      // memory for workspace
  magma_dmalloc_cpu(&r_work, lrwork);       // memory for workspace
  ...
  magma_zheevd(MagmaVec, MagmaLower, n, (magmaDoubleComplex *)r, n,
               w1, (magmaDoubleComplex *)h_work, lwork, r_work, lrwork,
               iwork, liwork, &info);
  ...
  magma_finalize();                          // finalize Magma
}
// complete code at twqcd80:/home/cuda_lecture_2021/maga/zheevd/magma_zheevd.c
```

# Laplacian operator on the 1-dim lattice

```
void laplacian_matrix(int n, double *a)   // laplacian op. on a circular lattice
{
   int  i, x, ix;
   memset(a, 0, n*n*sizeof(double));
   for (i=0; i < n; i++) {
       a[i+i*n] = -2.0;
       x = (i+1) % n;
       a[i+x*n] = 1.0;
       x = (i-1+n) % n;
       a[i+x*n] = 1.0;
   }
}
// twqcd80:/work/cuda_lecture_2021/maga/dsyevd_laplacian/magma_dsyevd_laplacian.c
```

$\phi(x)$: Field variable at the site $x = 0,\ 1,\ \cdots, N-1$

$$\nabla^2\phi(x) \to \frac{1}{a^2}\left[\phi(x+a) + \phi(x-a) - 2\phi(x)\right] \equiv \sum_{x'} M(x,x')\phi(x')$$

1-dimensional lattice with periodic boundary condition is equivalent to the lattice on a circle, i.e., $x_{-1} = x_{N-1},\ x_N = x_0$

Obviously, we have $\phi(x+L) = \phi(x),\ \ L = Na$

# Laplacian operator on the 1-dim lattice

```
void laplacian_matrix(int n, double *a)   // laplacian op. on a circular lattice
{
   int  i, x, ix;
   memset(a, 0, n*n*sizeof(double));
   for (i=0; i < n; i++) {
       a[i+i*n] = -2.0;
       x = (i+1) % n;
       a[i+x*n] = 1.0;
       x = (i-1+n) % n;
       a[i+x*n] = 1.0;
   }
}
// twqcd80:/home/cuda_lecture_2021/maga/dsyevd_laplacian/magma_dsyevd_laplacian.c
```

For $N = 6$

$$M = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 1 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 1 & 0 & 0 & 0 & 1 & -2 \end{pmatrix}$$

$\phi(x)$: Field variable at the site $x = 0,\ 1,\ \cdots, N-1$

$$\nabla^2\phi(x) \to \frac{1}{a^2}\left[\phi(x+a) + \phi(x-a) - 2\phi(x)\right] \equiv \sum_{x'} M(x,x')\phi(x')$$

1-dimensional lattice with periodic boundary condition is equivalent to the lattice on a circle, i.e., $x_{-1} = x_{N-1},\ x_N = x_0$

Obviously, we have $\phi(x+L) = \phi(x),\ \ L = Na$

# Schrödinger equation on the 1-Dim Lattice

$$H\psi = \frac{-\hbar^2}{2m}\nabla^2\psi(x) + V(x)\psi(x) = E\psi(x)$$

$\psi(x)$: wave function at the site $x$, $x = 0,\ 1,\ \cdots, N-1$

# Schrödinger equation on the 1-Dim Lattice

$$H\psi = \frac{-\hbar^2}{2m}\nabla^2\psi(x) + V(x)\psi(x) = E\psi(x)$$

$\psi(x)$: wave function at the site $x$, $x = 0,\ 1,\ \cdots, N-1$

For a free particle, $V(x) = 0$, it gives

$$-\psi(x+1) - \psi(x-1) + 2\psi(x) = \lambda\psi(x), \quad x = 0, \cdots, N-1,$$

Let $\psi(x) = \exp(i\alpha x)$, it gives $\lambda = 2 - 2\cos(\alpha)$.

Periodic boundary condition $\psi(-1) = \psi(N-1), \quad \psi(N) = \psi(0)$

gives $\alpha N = 2j\pi \ \Rightarrow\ \alpha_j = \frac{2j\pi}{N},\ j = 0, \cdots, N-1$

Eigenvalue: $\lambda_j = 2 - 2\cos(\alpha_j)$

Eigenvector: $\psi_j(x) = \cos(\alpha_j x)$

# Schrödinger equation, Particle in a box

$$H\psi = \frac{-\hbar^2}{2m}\nabla^2\psi(x) + V(x)\psi(x) = E\psi(x)$$

$\psi(x)$: wave function at the site $x$, $x = 0,\ 1,\ \cdots, N-1$

# Schrödinger equation, Particle in a box

$$H\psi = \frac{-\hbar^2}{2m}\nabla^2\psi(x) + V(x)\psi(x) = E\psi(x)$$

$\psi(x)$: wave function at the site $x$, $x = 0,\ 1,\ \cdots, N-1$

For a particle in a box, $V(x) = 0,\ \psi(x_{-1}) = 0,\ \psi(x_N) = 0.$

The Hamiltonian can be generated as follows.

# Schrödinger equation, Particle in a box

$$H\psi = \frac{-\hbar^2}{2m}\nabla^2\psi(x) + V(x)\psi(x) = E\psi(x)$$

$\psi(x)$: wave function at the site $x = 0,\ 1,\ \cdots, N-1$

For a particle in a box, $V(x) = 0,\ \psi(x_{-1}) = 0,\ \psi(x_N) = 0.$

The Hamiltonian can be generated as follows.

```
void laplacian_matrix(int n, double *a)  // Hamiltonian of a particle in a box
{
   int  i, x, ix;
   memset(a, 0, n*n*sizeof(double));
   for (i=0; i < n; i++) {
       a[i+i*n] = -2.0;
       x = (i+1) % n;
       a[i+x*n] = 1.0;
       x = (i-1+n) % n;
       a[i+x*n] = 1.0;
   }
   a[n-1]=0.0;            // boundary conditions
   a[(n-2)*(n-1)]=0.0;
} //see twqcd80:/home/cuda_lecture_2021/maga/dsyevd_pinbox/magma_dsyevd_pinbox.c
```

# Schrödinger equation, Particle in a box

$$H\psi = \frac{-\hbar^2}{2m}\nabla^2\psi(x) + V(x)\psi(x) = E\psi(x)$$

$\psi(x)$: wave function at the site $x = 0,\ 1,\ \cdots, N-1$

For a particle in a box, $V(x) = 0,\ \psi(x_{-1}) = 0,\ \psi(x_N) = 0.$

The Hamiltonian can be generated as follows.

```
void laplacian_matrix(int n, double *a)  // Hamiltonian of a particle in a box
{
   int  i, x, ix;
   memset(a, 0, n*n*sizeof(double));
   for (i=0; i < n; i++) {
       a[i+i*n] = -2.0;
       x = (i+1) % n;
       a[i+x*n] = 1.0;
       x = (i-1+n) % n;
       a[i+x*n] = 1.0;
   }
   a[n-1]=0.0;  // boundary conditions
   a[n*(n-1)]=0.0;
} //see twqcd80:/home/cuda_lecture_2021/maga/dsyevd_pinbox/magma_dsyevd_pinbox.c
```

For $N = 6$

$$M = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{pmatrix}$$

# Schrödinger equation, Particle in a box

$$\frac{-\hbar^2}{2m}\nabla^2\psi(x) = E\psi(x)$$

$\psi(x)$: wave function at the site $x = 0,\ 1,\ \cdots, N-1$

Boundary condition: $\psi(x_{-1}) = 0,\ \psi(x_N) = 0.$

# Schrödinger equation, Particle in a box

$$\frac{-\hbar^2}{2m}\nabla^2\psi(x) = E\psi(x)$$

$\psi(x)$: wave function at the site $x = 0,\ 1,\ \cdots, N-1$

Boundary condition: $\psi(x_{-1}) = 0,\ \psi(x_N) = 0.$

$$-\psi(x+1) - \psi(x-1) + 2\psi(x) = \lambda\psi(x), \quad x = 0, \cdots, N-1,$$

Let $\psi(x) = \exp(i\alpha x)$, it gives $\lambda = 2 - 2\cos(\alpha).$

$$\psi(-1) = 0 \ \Rightarrow\ \psi(x) = \sin(\alpha(x+1)).$$

Then $\psi(N) = 0 \ \Rightarrow\ \sin(\alpha(N+1)) = 0 \ \Rightarrow\ \alpha_j = \frac{j\pi}{N+1},\ j = 1, \cdots, N$

Eigenvalue: $\lambda_j = 2 - 2\cos(\alpha_j)$, Eigenvector: $\psi_j(x) = \sin(\alpha_j(x+1))$

# Schrödinger equation, Particle in a box

$$\frac{-\hbar^2}{2m}\nabla^2\psi(x) = E\psi(x)$$

$\psi(x)$: wave function at the site $x = 0,\ 1,\ \cdots, N-1$

Boundary condition: $\psi(x_{-1}) = 0,\ \psi(x_N) = 0.$

$$-\psi(x+1) - \psi(x-1) + 2\psi(x) = \lambda a^2 \psi(x), \quad x = 0, \cdots, N-1,$$

Let $\psi(x) = \exp(i\alpha x)$, it gives $\lambda = 2 - 2\cos(\alpha).$

$\psi(-1) = 0 \ \Rightarrow\ \psi(x) = \sin(\alpha(x+1)).$

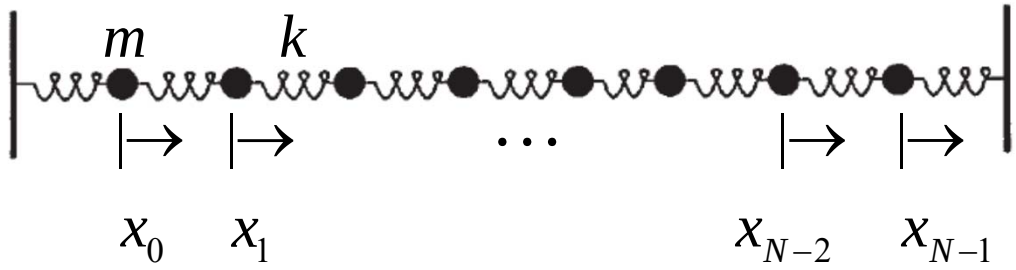
Then $\psi(N) = 0 \ \Rightarrow\ \sin(\alpha(N+1)) = 0 \ \Rightarrow\ \alpha_j = \dfrac{j\pi}{N+1},\ j = 1, \cdots, N$

Eigenvalue: $\lambda_j = 2 - 2\cos(\alpha_j)$, Eigenvector: $\psi_j(x) = \sin(\alpha_j(x+1))$

In the continuum limit, $a \to 0,\ N \to \infty,\ (N+1)a = L = \text{fixed},$

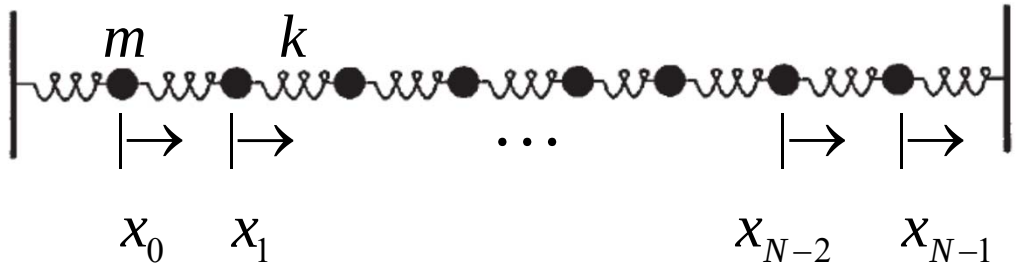
$$\lambda_j = \alpha_j{}^2 = \frac{\pi^2 j^2}{L^2},\ j = 1, 2, \cdots \quad E_j = \frac{\hbar^2}{2m}\frac{\pi^2 j^2}{L^2},\ \psi_j(x) = \sin(\alpha_j x)$$

# A chain of mass-spring system with fixed ends




Consider a chain of particles (each of mass $m$) connected by springs (each of spring constant $k$), and with both ends fixed at the walls, as shown in the figure. Using Newton's law, we have

# A chain of mass-spring system with fixed ends



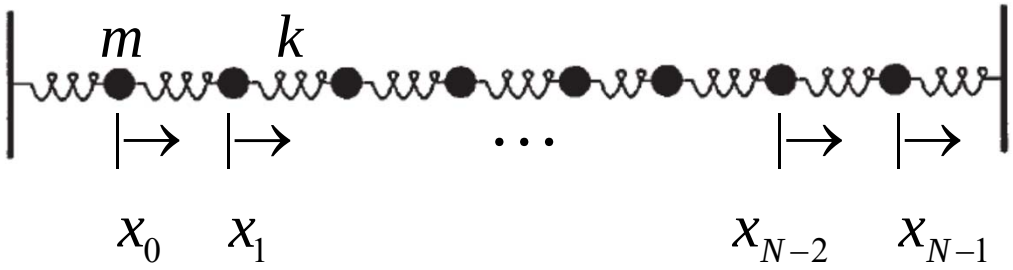

Consider a chain of particles (each of mass $m$) connected by springs (each of spring constant $k$), and with both ends fixed at the walls, as shown in the figure. Using Newton's law, we have

$$m\ddot{x}_0 = -k(x_0 - 0) - k(x_0 - x_1) = k(x_1 - 2x_0),$$

$$m\ddot{x}_i = -k(x_i - x_{i-1}) - k(x_i - x_{i+1}) = k(x_{i+1} + x_{i-1} - 2x_i), \quad i = 1, \cdots, N-2$$

$$m\ddot{x}_{N-1} = -k(x_{N-1} - x_{N-2}) - k(x_{N-1} - 0) = k(x_{N-2} - 2x_{N-1}),$$

# A chain of mass-spring system with fixed ends




Consider a chain of particles (each of mass $m$) connected by springs (each of spring constant $k$), and with both ends fixed at the walls, as shown in the figure. Using Newton's law, we have

$$m\ddot{x}_0 = -k(x_0 - 0) - k(x_0 - x_1) = k(x_1 - 2x_0),$$

$$m\ddot{x}_i = -k(x_i - x_{i-1}) - k(x_i - x_{i+1}) = k(x_{i+1} + x_{i-1} - 2x_i), \quad i = 1, \cdots, N-2$$

$$m\ddot{x}_{N-1} = -k(x_{N-1} - x_{N-2}) - k(x_{N-1} - 0) = k(x_{N-2} - 2x_{N-1}),$$

Let $x_i(t) = u_i \exp[-i\omega t]$, then these equations become

$$\left.\begin{aligned} -\lambda u_0 &= u_1 - 2u_0, \qquad \lambda \equiv m\omega^2 / k \\ -\lambda u_i &= u_{i+1} + u_{i-1} - 2u_i, \quad i = 1, \cdots, N-2 \\ -\lambda u_{N-1} &= u_{N-2} - 2u_{N-1}, \end{aligned}\right\} \Leftrightarrow -\lambda u = \nabla^2 u$$

# A chain of mass-spring system with fixed ends


Consider a chain of particles (each of mass $m$) connected by springs (each of spring constant $k$), and with both ends fixed at the walls, as shown in the figure. Using Newton's law, we have

$$m\ddot{x}_0 = -k(x_0 - 0) - k(x_0 - x_1) = k(x_1 - 2x_0),$$

$$m\ddot{x}_i = -k(x_i - x_{i-1}) - k(x_i - x_{i+1}) = k(x_{i+1} + x_{i-1} - 2x_i), \quad i = 1, \cdots, N-2$$

$$m\ddot{x}_{N-1} = -k(x_{N-1} - x_{N-2}) - k(x_{N-1} - 0) = k(x_{N-2} - 2x_{N-1}),$$

Let $x_i(t) = u_i \exp[-i\omega t]$, then these equations become

$$\left.\begin{aligned} -\lambda u_0 &= u_1 - 2u_0, \qquad \lambda \equiv m\omega^2 / k \\ -\lambda u_i &= u_{i+1} + u_{i-1} - 2u_i, \quad i = 1, \cdots, N-2 \\ -\lambda u_{N-1} &= u_{N-2} - 2u_{N-1}, \end{aligned}\right\} \Leftrightarrow -\lambda u = \nabla^2 u$$

For $N = 6$

$$M = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{pmatrix}$$

# Momentum operator on the 1-dim lattice

```
void momentum_matrix(int n, double *a)  // momentum op. on a circular lattice
{
   int  i, x, ix;

   memset(a, 0, 2*n*n*sizeof(double));
   for (i=0; i< n; i++) {
       x = (i+1) % n;
       a[(i+x*n)*2+1] = -0.5;    // -i/2
       x = (i-1+n) % n;
       a[(i+x*n)*2+1] =  0.5;    //  i/2
   }
}
// twqcd80:/home/cuda_lecture_2021/maga/zheevd_momentum/magma_zheevd_momentum.c
```

$\psi(x)$: wave function at the site $x = 0,\ 1,\ \cdots, N-1$

$$P \to -i\hbar\nabla,\ P\psi(x) \to \frac{-i\hbar}{2a}\left[\psi(x+a)-\psi(x-a)\right] \equiv \sum_{x'} P(x,x')\psi(x')$$

1-dimensional lattice with periodic boundary condition is equivalent to the lattice on a circle, i.e., $x_{-1} = x_{N-1},\ x_N = x_0$

Obviously, we have $\psi(x+L) = \psi(x),\ L = Na$

# Momentum operator on the 1-dim lattice

```
void momentum_matrix(int n, double *a)  // momentum op. on a circular lattice
{
   int  i, x, ix;

   memset(a, 0, 2*n*n*sizeof(double));
   for (i=0; i< n; i++) {
       x = (i+1) % n;
       a[(i+x*n)*2+1] = -0.5;    // -i/2
       x = (i-1+n) % n;
       a[(i+x*n)*2+1] =  0.5;    //  i/2
   }
}
// twqcd80:/home/cuda_lecture_2021/maga/zheevd_momentum/magma_zheevd_momentum.c
```

For $N = 6$

$$P = \begin{pmatrix} 0 & -i/2 & 0 & 0 & 0 & i/2 \\ i/2 & 0 & -i/2 & 0 & 0 & 0 \\ 0 & i/2 & 0 & -i/2 & 0 & 0 \\ 0 & 0 & i/2 & 0 & -i/2 & 0 \\ 0 & 0 & 0 & i/2 & 0 & -i/2 \\ -i/2 & 0 & 0 & 0 & i/2 & 0 \end{pmatrix}$$

$\psi(x)$: wave function at the site $x = 0,\ 1,\ \cdots, N-1$

$$P \to -i\hbar\nabla,\ P\psi(x) \to \frac{-i\hbar}{2a}\left[\psi(x+a) - \psi(x-a)\right] \equiv \sum_{x'} P(x,x')\psi(x')$$

1-dimensional lattice with periodic boundary condition is equivalent to the lattice on a circle, i.e., $x_{-1} = x_{N-1},\ x_N = x_0$

Obviously, we have $\psi(x+L) = \psi(x),\ L = Na$

# Momentum operator on the 1-dim lattice

$\psi(x)$: wave function at the site $x = 0,\ 1,\ \cdots, N-1$

$$P \to -i\hbar\nabla,\ P\psi(x) \to \frac{-i\hbar}{2a}\left[\psi(x+a)-\psi(x-a)\right] \equiv \sum_{x'} P(x,x')\psi(x')$$

$$-\frac{i}{2}[\psi(x+1)-\psi(x-1)] = \lambda\psi(x), \quad x = 0,\cdots,N-1,$$

Let $\psi(x) = \exp(i\alpha x)$, it gives $\lambda = \sin(\alpha)$.

Periodic boundary condition $\psi(-1) = \psi(N-1), \quad \psi(N) = \psi(0)$

gives $\alpha N = 2j\pi \ \Rightarrow\ \alpha_j = \frac{2j\pi}{N},\ j = 0,\cdots,N-1$

Momentum eigenvalue: $\lambda_j = \sin(\alpha_j)$

Momentum eigenvector: $\psi_j(x) = \exp(i\alpha_j x)$