Lecture 6

Normal Modes and Waves

Hai-Qing Lin

Beijing Computational Science Research Center

This PowerPoint Notes Is Based on the Textbook '***An Introduction to Computer Simulation Methods : Applications to Physical Systems***', 2nd Edition, Harvey Gould and Jan Tobochnik, Addison-Wesley(1996); "A First Course in Computational Physics"; "Numerical Recipes"; "Elementary Numerical Analysis"; "Computational Methods in Physics and Engineering".

Questions and Objectives

- Wave is a macroscopic phenomenon, while the oscillatory motion of each particle is a microscopic phenomenon, what is the relation between them and how to simulate them?
- By study coupled oscillators we introduce the concept of normal modes.
- We will also study an important mathematic tool in analyzing physical systems: the Fourier transform
- Simulation of waves.

VOLUME 89, NUMBER 15 PHYSICAL REVIEW LETTERS 7 OCTOBER 2002

Time-Delayed Spatial Patterns in a Two-Dimensional Array of Coupled Oscillators

Seong-Ok Jeong, Tae-Wook Ko,* and Hie-Tae Moon

Department of Physics, Korea Advanced Institute of Science and Technology, Taejon 305-701, Korea

(Received 20 March 2002; published 20 September 2002)

We investigated the effect of time delays on phase configurations in a set of two-dimensional coupled phase oscillators. Each oscillator is allowed to interact with its neighbors located within a finite radius, which serves as a control parameter in this study. It is found that distance-dependent time delays induce various patterns including traveling rolls, squarelike and rhombuslike patterns, spirals, and targets. We analyzed the stability boundaries of the emerging patterns and briefly pointed out the possible empirical implications of such time-delayed patterns.

We start with the following coupled oscillator model equations for phase variables $\theta_{ij}(t)$ (i and j are integers):

$$\dot{\theta}_{ij}(t) = w + \frac{K}{N(r_0)} \sum_{k,l}^{0<r_{kl,ij}\le r_0} W(r_{kl,ij}) \times \sin\left[\theta_{kl}\left(t - \frac{r_{kl,ij}}{v}\right) - \theta_{ij}(t)\right], \quad (1)$$

with $N(r_0) = \sum_{k,l}^{0<r_{kl,ij}\le r_0} W(r_{kl,ij})$. Here, $r_{kl,ij}$ denotes the distance between element (i, j) and element (k, l). The metric properties of the ensemble are fixed by setting $r_{kl,ij} = \sqrt{(i-k)^2 + (j-l)^2}$.

VOLUME 89, NUMBER 15 PHYSICAL REVIEW LETTERS 7 OCTOBER 2002

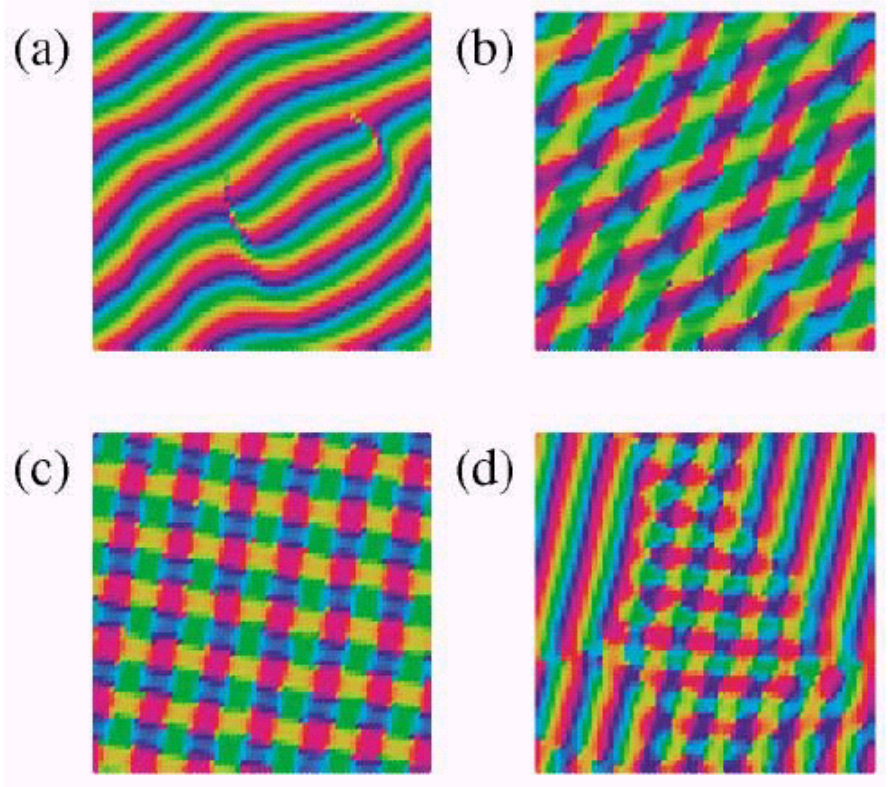


FIG. 3 (color). Typical patterns generated in the model: (a) a traveling roll, (b) a rhombuslike pattern, (c) a squarelike pattern, (d) coexistence of a rhombuslike pattern and a roll. $K = 1.25$, $r_0 = 25$, and $1/v = 1$, for (a), (b), and (c). Note that planar solutions are also found for this set of parameter values; $K = 0.6$, $r_0 = 10$, and $1/v = 1.1$, for (d). See Fig. 4(d) for the color coding.

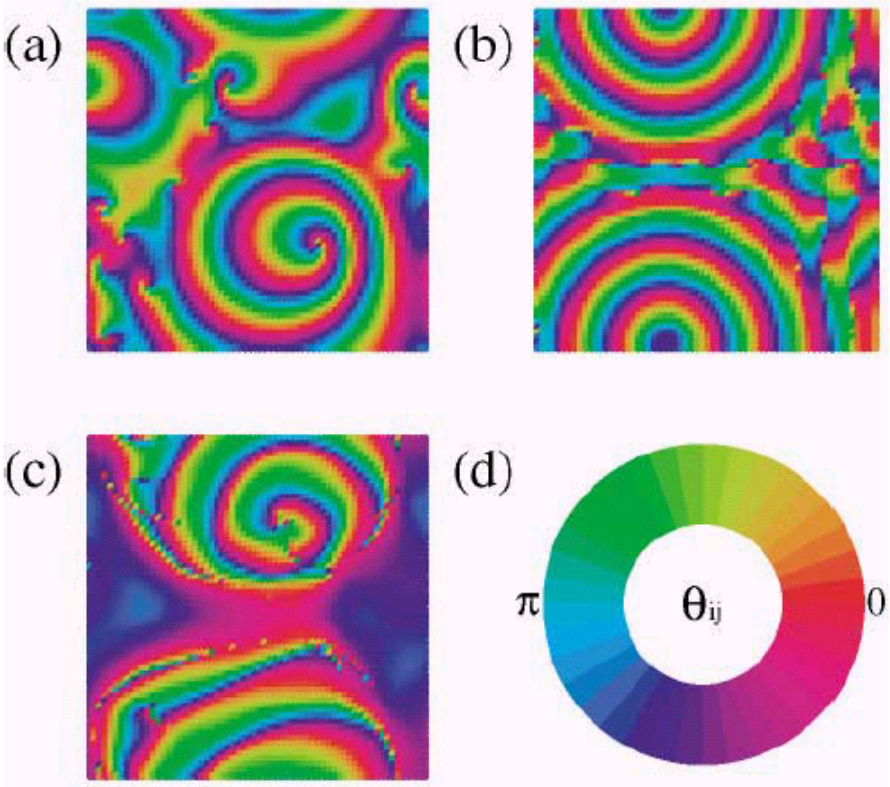
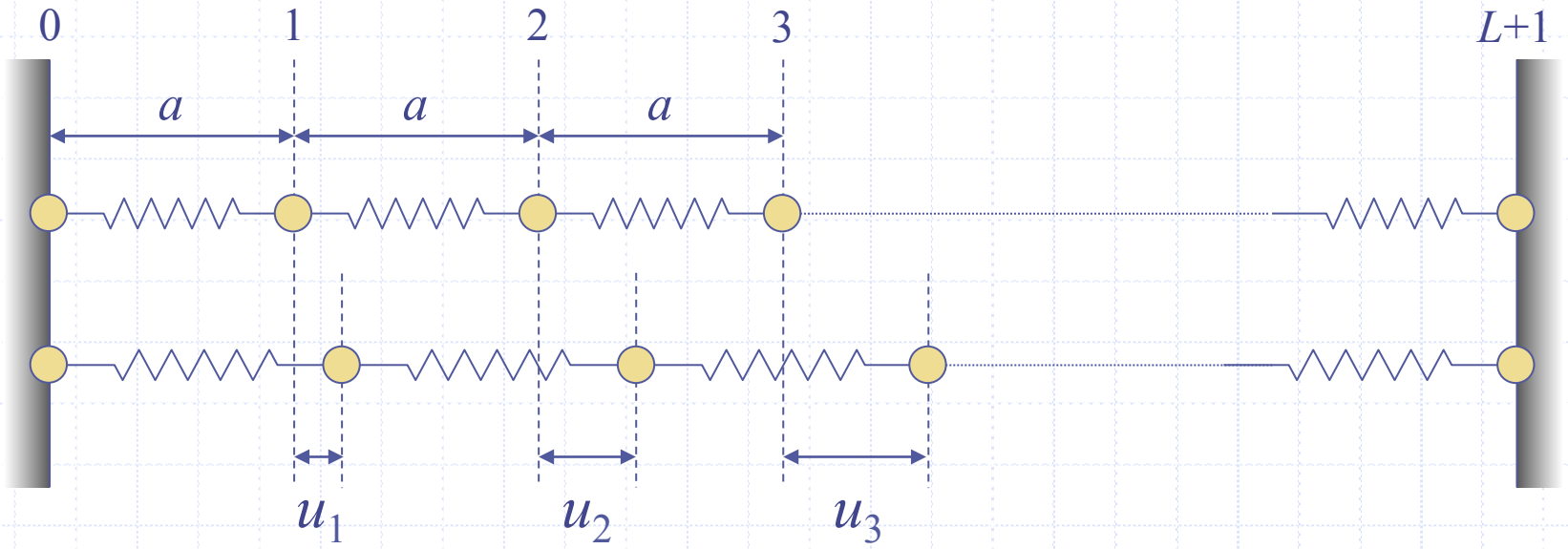


FIG. 4 (color). Patterns obtained in simulations of the model: (a) spirals ($K = 1.0$, $r_0 = 4$, and $1/v = 1$), (b) a target pattern ($K = 0.4$, $r_0 = 7$, and $1/v = 1$), (c) a spiral embedded in a planar solution ($K = 0.8$, $r_0 = 8$, and $1/v = 1$). In (c), the planar solution oscillates much more slowly than the spiral. Figure (d) shows the color code used in this paper.

A One-dimensional Model



Parameters

L : number of particles.
M : mass of a particles.
K : force constant of massless springs.
a : equilibrium separation.
$u_j(t)$: displacement of particle j.

Equation of Motion

$$M\frac{d^2u_j(t)}{dt^2} = -K\left[u_j(t)-u_{j+1}(t)\right]-K\left[u_j(t)-u_{j-1}(t)\right]$$

$$= -K\left[2u_j(t)-u_{j+1}(t)-u_{j-1}(t)\right]$$

- **Coupled linear equations** describing ***longitudinal*** oscillations.
- Similar equations hold for ***transverse*** oscillations.

Boundary Conditions

- **Fixed boundary**:
  $u_0(t) = u_{L+1}(t) = 0.$
- **Periodic boundary**:
  $u_0(t) = u_L(t);\ \ u_1(t) = u_{L+1}(t).$
- **Free boundary**:
  $u_0(t) = u_1(t);\ \ u_L(t) = u_{L+1}(t).$

Normal Modes

$$M\frac{d^2u_j(t)}{dt^2} = -K\left[u_j(t)-u_{j+1}(t)\right]-K\left[u_j(t)-u_{j-1}(t)\right]$$

$$= -K\left[2u_j(t)-u_{j+1}(t)-u_{j-1}(t)\right]$$

- Assuming $u_j(t) = u_j \cos \omega t$, we obtain

  $-\omega^2 u_j = -(K/M)\,[2u_j - u_{j+1} - u_{j-1}]$.

- This is an ***eigenvalue equation*** which can be solved by assuming $u_j = C \sin qja$, then

  $-\omega^2 \sin qja = -(K/M)\,[2\sin qja - \sin q(j+1)a - \sin q(j-1)a]$.

Normal Modes

- This is an ***eigenvalue equation*** which can be solved by assuming $u_j = C \sin qja$, then

$$-\omega^2 \sin qja = -(K/M)\,[2\sin qja - \sin q(j+1)a - \sin q(j-1)a].$$

- Using $\sin q(j\pm1)a = \sin qja \cos qa \pm \cos qja \sin qa$, we obtain $\omega^2 = 2(K/M)\,(1-\cos qa)$.

- To satisfy boundary conditions $u_0(t) = u_{L+1}(t) = 0$, we must have

$$q = q_n = \frac{\pi\ n}{a(L+1)} \qquad n = 1, 2, \mathrm{L}\ , L.$$

Normal Modes

- Thus we obtain the ***dispersion relation***:

$$\omega_n^2 = 2\frac{K}{M}\left(1-\cos q_n a\right) = 4\frac{K}{M}\sin^2\frac{q_n a}{2}$$

$$\omega_n = 2\sqrt{\frac{K}{M}}\sin\frac{q_n a}{2}$$

- The ***n*th normal mode** (corresponding to a particular value of integer ***n***) solution is

$$u_{j,n} = C\sin q_n j a.$$

Orthogonality

- The normal mode solution, $u_{j,n}$, are **orthogonal**, and if we choose $C^2 = 1 / [(L + 1)/2]$, then they are ***orthonormal***, i.e.,

$$\sum_{j=1}^{L} u_{j,n} u_{j,m} = \delta_{n,m}$$

→ p.21

- Since our equation of motion is ***linear***, the time dependence of the displacement of the ***j*th** particle can be written as a ***superposition*** of normal modes:

$$u_j(t) = C \sum_{n=1}^{L} \left(A_n \cos \omega_n t + B_n \sin \omega_n t \right) \sin q_n j a$$

Orthogonality

- where coefficients A_n and B_n are determined by the ***initial conditions***:

$$u_j(t=0) = \sum_{n=1}^{L} A_n u_{j,n} = C \sum_{n=1}^{L} A_n \sin q_n ja$$

$$v_j(t=0) = \sum_{n=1}^{L} \omega_n B_n u_{j,n} = C \sum_{n=1}^{L} \omega_n B_n \sin q_n ja.$$

- with the use of orthonormal condition, we obtain

$$A_n = C \sum_{j=1}^{L} u_j(t=0) \sin q_n ja$$

$$B_n = C \sum_{j=1}^{L} \left(v_j(t=0) / \omega_n \right) \sin q_n ja$$

Example

$$u_j(t=0)=0 \qquad v_1(t=0)=1$$

$$A_n = 0 \qquad B_n = \frac{C}{\omega_n}\sin q_n a$$

$$u_j(t) = \frac{2}{L+1}\sum_{n=1}^{L}\frac{1}{\omega_n}\cos\omega_n t \;\; \sin q_n a \;\; \sin q_n j a$$

More Complicated Cases

- Such as different boundary conditions, masses, spring constants, etc.
- The equation of motion is solvable as long as it is ***linear***. The general approach is to look for the ***eigenvalues*** ω_n and corresponding ***eigenvectors*** $u_{j,n}$ of the matrix equation: $\mathbf{Tu} = \omega^2\mathbf{u}$ where the matrix elements $T_{i,j}$ are zero except for

$$T_{i,i} = \frac{1}{M_i}\left[K_{i,i+1} + K_{i,i-1}\right], \quad T_{i,i+1} = -\frac{K_{i,i+1}}{M_i}, \quad T_{i,i-1} = -\frac{K_{i,i-1}}{M_i}$$

$K_{i,j}$ could be long-ranged, then there are more non-zero matrix elements.

Solution of Matrix Equation

- The solution of matrix equations is a well studied problem in linear programming.
- There exist many commercial subroutine packages such as **IMSL** or symbolic programming language such as **Mathematica**.
- However, for ***nonlinear*** force the matrix approach is, in general, inapplicable and one has to find the numerical solution of the equations of motion directly, say, by **iteration**.

Stationary Solution

$$0 = M_j \frac{d^2 u_j(t)}{dt^2} = -K_{j,j+1}[u_j(t) - u_{j+1}(t)] - K_{j-1,j}[u_j(t) - u_{j-1}(t)]$$

$$u_j^{(\text{new})}(t) = \frac{K_{j,j+1}u_{j+1}(t) + K_{j-1,j}u_{j-1}(t)}{K_{j,j+1} + K_{j-1,j}} + \gamma u_j^2(t)$$

- This equation could be used for iteration.
- It works for other potentials, e.g., nonlinear term.

In program ***oscillators*** we use the ***Euler-Richardson*** algorithm to simulate the dynamics of L linear coupled oscillators.

```
PROGRAM OSCILLATORS
! simulate coupled linear oscillators in one dimension
IMPLICIT NONE
Real, dimension(0:21) :: u, v, usave
Real     :: t, dt, ke
Integer :: L, i

  Call initgraph                                  ! Graphic setup
  Call initial(L, u, v, t, dt, usave)

  Do i = 1, 4000
    Call update(L, u, v, ke, t, dt, usave)
    Call animate(L, u, ke, t, usave)
  Enddo

  Call getch
  Call closegraph

END PROGRAM OSCILLATORS
```

```
Subroutine initial(L, u, v, t, dt, usave)
IMPLICIT NONE
Real, intent(out) :: t, dt
Real, dimension(0:21), intent(out) :: u, v, usave
Integer, intent(out) :: L
Integer :: j, y1, y2
  write(6,*) 'Enter L, t_0, d_t'
  read(5,*) L,t,dt                ! e.g., t = 0.; dt = 0.025; L = 10
  Call background(0,0,0)
  Call setcolor(5,0,0)
  Call line( -7*30+225,225,7*30+225,225 )
   u = 0.                         ! initial displacements; May read in
   v = 0.                         ! initial velocities
  u(2) = 0.5
  Do j = 1, L
   Call setcolor( 5,0,0 )
   Call disc( (j-5)*30 + int(u(j)*50) + 220,220,5 )
  Enddo
  u(0) = 0.                       ! fixed wall boundary conditions
  u(L+1) = 0.
  usave = u                       ! matrix assignment instruction
End Subroutine initial
```

```
Subroutine update(L, u, v, ke, t, dt, usave) ! Euler-Richardson algorithm
IMPLICIT NONE                                ! K/M equal to unity
Real, intent(inout) :: t, dt, ke
Real, dimension(0:21), intent(inout) :: u, v, usave
Integer, intent(in) :: L
Real, dimension(0:21) :: a, amid, vmid, umid
Integer :: j

  Do j = 1, L
    usave(j) = u(j)
    a(j) = -2.*u(j) + u(j+1) + u(j-1)
    umid(j) = u(j) + 0.5*v(j)*dt
    vmid(j) = v(j) + 0.5*a(j)*dt
  EndDO
  umid(0) = 0.
  umid(L+1) = 0.

  ke=0.
  Do j = 1, L
     amid(j)=-2.*umid(j)
              +umid(j+1) + umid(j-1)
     u(j) = u(j) + vmid(j)*dt
     v(j) = v(j) + amid(j)*dt
     ke = ke + v(j)*v(j)
  EndDO
  t = t + dt

End Subroutine update
```

```
Subroutine animate(L, u, ke, t, usave)
IMPLICIT NONE
Real, intent(inout) :: t, ke
Real, dimension(0:21), intent(inout) :: u, usave
Integer, intent(in) :: L
Character(20) :: aa
Integer :: j, y1, y2
Real :: pe, e
  pe = (u(1) - u(0))**2.                ! interaction with left spring
  Do j = 1, L                           ! compute potential energy
    pe = pe + (u(j+1) - u(j))**2.       ! transverse oscillation
    Call setcolor(0,0,0)
    Call disc((j-5)*30 + int(usave(j)*50) + 220,220,5)
    Call setcolor(5,0,0)
    Call disc((j-5)*30 + int(u(j)*50) + 220,220,5)
    Call line(-7*30+225,225,7*30+225,225)
  Enddo
  Call setcolor(5,5,5)
  Write(aa,10), "t = ",t
10 Format(A4,F6.2)
  Call outtext(350,350,aa)
  e = 0.5*(ke+pe)
  Write(aa, 11), 'e = ',e
11 Format(A4,F6.4)
  Call outtext(350,410,aa)
End Subroutine animate
```

Linear Problem

- For a physical system, we say it is linear system if we have the following property:

  $F(ax + by) = a\,F(x) + b\,F(y)$.

- For linear problem, one can apply superposition principle, which simplifies the problem greatly.
- Examples: Newtonian mechanics, E&M, Quantum mechanics (Schodinger equation), etc.

Fourier Transforms

- Any arbitrary function $f(t)$ of period T can be expressed as a Fourier series of **sines** and **cosines**.

$$f(t) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty}\left(a_k \cos\omega_k t + b_k \sin\omega_k t\right)$$

→ p.11

where $\omega_k = k\omega_0$, $\omega_0 = 2\pi/T$.

- ω_0 is the fundamental frequency. Terms in Fourier Series for $k = 2,3, \ldots$ represent 2nd, 3rd, …, and higher order harmonics. The a_k and b_k are called the **Fourier Coefficients**, given by

$$a_k = \frac{2}{T}\int_0^T f(t)\cos\omega_k t dt, \quad b_k = \frac{2}{T}\int_0^T f(t)\sin\omega_k t dt$$

Functions $\sin\omega_k t$ and $\cos\omega_k t$ form a complete set and they satisfy the following orthogonality conditions:

$$\frac{2}{T}\int_0^T \cos\omega_k t \cos\omega_m t \quad dt = \delta_{k,m}$$

$$\frac{2}{T}\int_0^T \sin\omega_k t \sin\omega_m t \quad dt = \delta_{k,m}$$

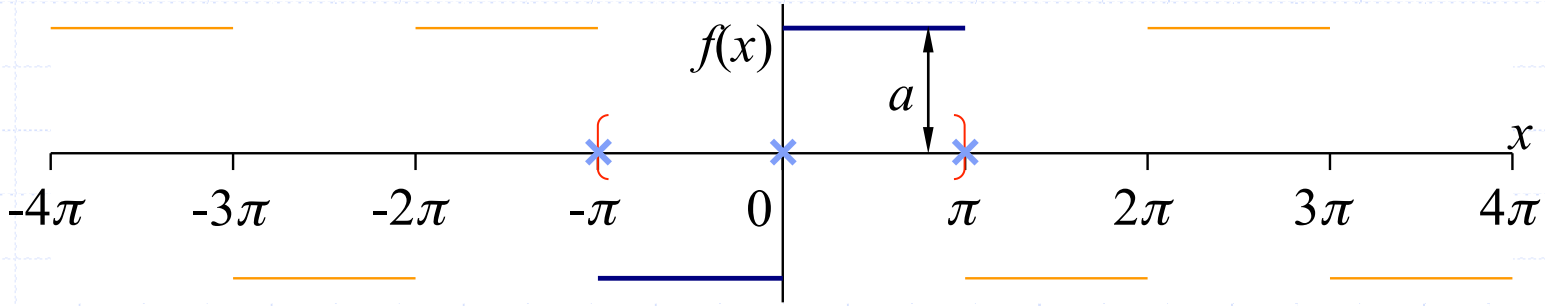
$$\frac{2}{T}\int_0^T \cos\omega_k t \sin\omega_m t \quad dt = 0$$

In general, an infinite number of terms is needed to represent an arbitrary periodic function exactly. But in practice, we usually only use a few terms to make approximation. A Fourier series can approximate a function **at all points** (unlike a power series).

Unit Step Function:

$$f(x) = a\begin{cases} +1 & 0 < x < \pi \\ -1 & -\pi < x < 0 \end{cases}$$

Period = 2π



$$a_n = 0; \quad b_n = \frac{2}{n\pi}\left(1-\cos n\pi\right) = \begin{cases} \frac{4}{n\pi} & n \text{ is odd} \\ 0 & n \text{ is even} \end{cases}$$

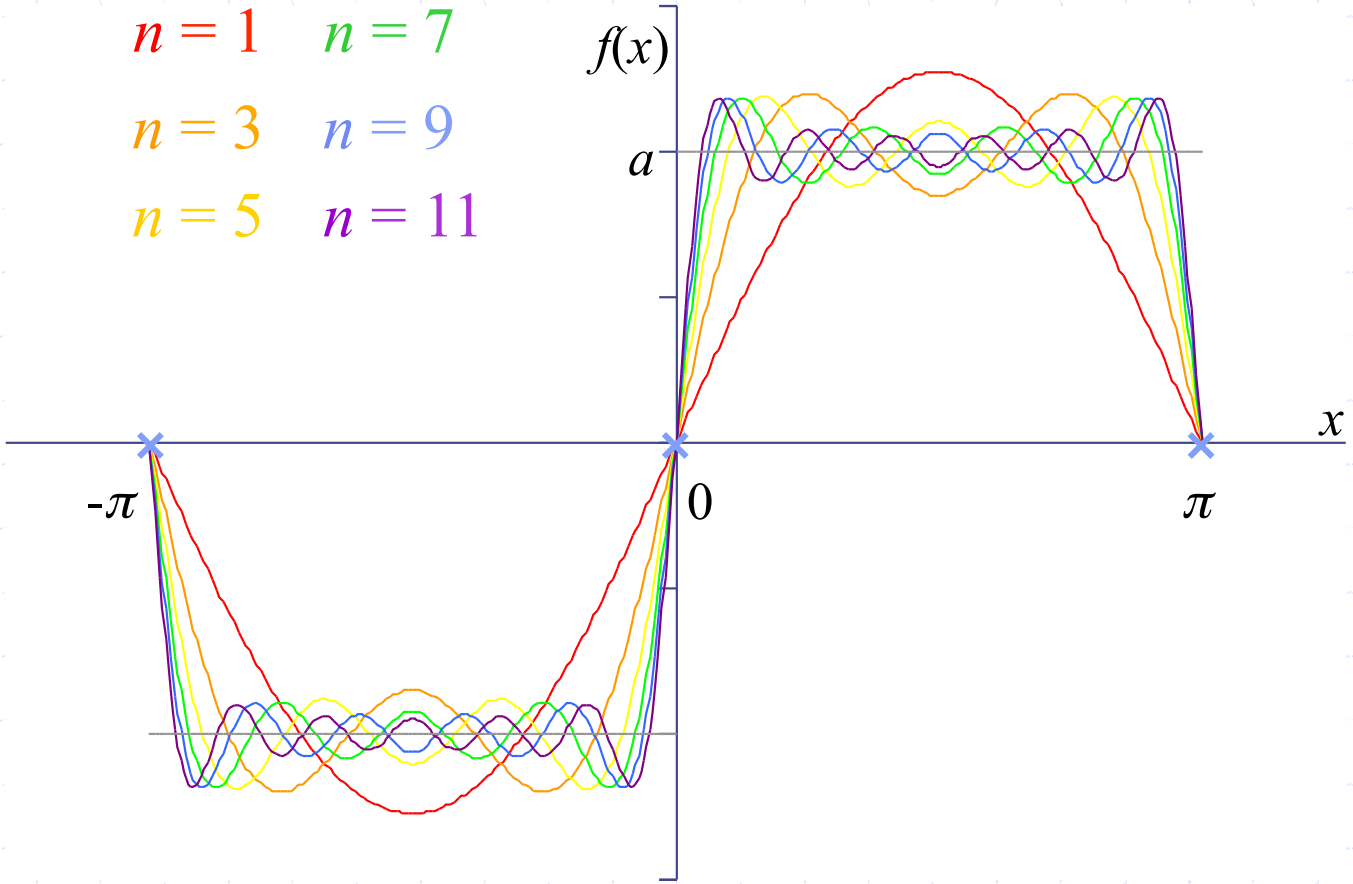
$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n \cos nx + b_n \sin nx\right) = \sum_{n=1}^{\infty}\frac{4}{(2n-1)\pi}\sin(2n-1)x$$

$$f(x=\pi/2) = 1 = \sum_{n=1}^{\infty}\frac{4(-1)^{n-1}}{(2n-1)\pi} \quad \Rightarrow \frac{\pi}{4} = \sum_{n=1}^{\infty}\frac{(-1)^{n-1}}{2n-1}$$

Unit Step Function:

$$f(x) = a\begin{cases} +1 & 0 < x < \pi \\ -1 & -\pi < x < 0 \end{cases}$$

Period = 2π



Use Fourier.pdf for more …

Program Synthesize (text book)

Program ***synthesize*** plots out various Fourier modes.

In the example attached, program synthesize was used to plot the following series

$$f(t) = \frac{2}{\pi}\left(\sin t + \frac{1}{3}\sin 3t + \frac{1}{5}\sin 5t + \cdots\right),$$

with $\omega_0 = 1$ ($T = 2\pi$).

Time Series Analysis

- Suppose we do not know $f(t)$ but we have a time series consisting of N data points, $f(t_i)$, where $t_i = 0, \Delta, 2\Delta, \ldots, (N-1)\Delta$, can we determine the Fourier coefficients by assuming the data **repeats itself** with a period $T = N\Delta$?
- Since we know only a finite number of terms in a time series, we can only find a finite set of Fourier coefficients. The maximum frequency we can analyse is

$$\omega_c = \pi/\Delta$$ **Nyquist critical frequency.**

Time Series Analysis

- We have $\omega_c/\omega_0 + 1$ independent coefficients for a_k, ω_c/ω_0 independent coefficients for b_k, so a total of $\boldsymbol{N+1}$ independent coefficients.

- Note, $\omega_0 = 2\pi/T$, $\omega_c = \pi/\Delta$, $T = N\Delta$.

- Since $b_{N/2} = 0$, there are total of $\boldsymbol{N}$ Fourier coefficients to be determined, just the same as the number of data points.

Fourier Series with Complex Coefficient

$$f(t)=\sum_{-\infty}^{\infty}c_k e^{i\omega_k t},\quad \text{with } c_k=\frac{1}{T}\int_0^T dt\quad f\left(t\right)e^{-i\omega_k t}.$$

We have these relations

$$c_0=\frac{1}{2}a_0;\quad c_k=\frac{1}{2}\left(a_k-ib_k\right);\quad c_{-k}=\left(c_k\right)^*.$$

For finite number of values of $f(j\Delta)$,we have

$$c_k\approx\frac{\Delta}{T}\sum_{j=0}^{N-1}f(j\Delta)e^{-i\omega_k j\Delta}\equiv\frac{\Delta}{T}g\left(\omega_k\right).$$

$$g\left(\omega_k\right)=\sum_{j=0}^{N-1}f(j\Delta)e^{-i\omega_k j\Delta}=\sum_{j=0}^{N-1}f(j\Delta)e^{-i2\pi\, k\, j/N}.$$

The Inverse Fourier Transform

$$f(j\Delta) = \frac{1}{N}\sum_{k=0}^{N-1} g(\omega_k) e^{-i2\pi kj/N} = \frac{1}{N}\sum_{k=0}^{N-1} g(\omega_k) e^{i\omega_k t_j}$$

- If $f(t)$ is real, we have $g(\omega_k) = g(\omega_k - \omega_N)$ and $g(-\omega_k) = g(\omega_k)$.
- To see the contribution of a particular frequency component within a signal, we measure the power $P(\omega)$ associated with that frequency. Start from:

$$\sum_{j=0}^{N-1} \left| f(t_j) \right|^2 = \frac{1}{N}\sum_{k=0}^{N-1} \left| g(\omega_k) \right|^2$$

- We define the ***power spectrum*** (the power in the frequency component ω_k is proportional to)

$$P(\omega_k) = \frac{1}{N^2}\left[|g(\omega_k)|^2 + |g(-\omega_k)|^2\right]$$

$$= \frac{2}{N^2}|g(\omega_k)|^2. \quad (0 < \omega_k < \omega_c)$$

$$P(\omega_c) = \frac{1}{N^2}|g(\omega_c)|^2; \quad P(0) = \frac{1}{N^2}|g(0)|^2.$$

- Note that there will be spurious results if $P(\omega)$ does not vanish above the Nyquist frequency (*aliasing*).
- The naive procedure to compute $P(\omega)$ is order of N^2.

Fast Fourier Transform (FFT) (Appendix 9A)

FFT is a very fast way of computing the Fourier transform $g(\omega_k)$ given the data set $f(j\Delta)$. If N is even, we have

$$g(\omega_k) = \sum_{j=0}^{N-1} f(j\Delta) e^{-i2\pi kj/N} = \left\{ \sum_{\substack{j=0 \\ j\in \text{even}}}^{N-1} + \sum_{\substack{j=0 \\ j\in odd}}^{N-1} \right\} f(j\Delta) e^{-i2\pi kj/N}$$

$$= \sum_{j=0}^{N/2-1} e^{-i2\pi kj/(N/2)} \left\{ f(2j\Delta) + W^k f\left[(2j+1)\Delta\right] \right\}$$

$$= g_{\text{even}}(\omega_k) + W^k g_{\text{odd}}(\omega_k) = g_k^{\text{even}} + W^k g_k^{\text{odd}}$$

where $W = e^{-\text{i}2\pi/N}$. g_k^{even} and g_k^{odd} are themselves DFTs, with half as many points, over the original even- and odd- numbered points.

Fast Fourier Transform (FFT)

- The number of operations is reduced from the order of N^2 to $2 \times (N/2)^2$ by this decomposition.
- Of course, we can continue doing this as long as they each contains an even number of points.
- Let's say $N = 2^k$, after k-steps, there will be N transforms to be evaluated, each containing only ***one*** point! The total operations is thus the order of $N\log_2 N$, not N^2.
- To perform FFT, we first reorder the components so that a bit reversal procedure on the original array representing the data could be done. In the next step this array is replaced by its Fourier transform.

Example, $N = 8$

- Decompose j and k into binary form

$$j = 4j_2 + 2j_1 + j_0 \qquad (j_2, j_1, j_0 = 0, 1)$$
$$k = 4k_2 + 2k_1 + k_0 \qquad (k_2, k_1, k_0 = 0, 1)$$

- Then

$$g(k_2,k_1,k_0) = \sum_{j_0=0}^{1}\sum_{j_1=0}^{1}\sum_{j_2=0}^{1} f(j_2,j_1,j_0) W^{(4k_2+2k_1+k_0)(4j_2+2j_1+j_0)}$$

- Since $W^8 = W^{16} = \cdots = 1$, we have

$$g(k_2,k_1,k_0) = \sum_{j_0=0}^{1} W^{(4k_2+2k_1+k_0)j_0} \sum_{j_1=0}^{1} W^{(2k_1+k_0)2j_1} \sum_{j_2=0}^{1} f(j_2,j_1,j_0) W^{4k_0j_2}$$

- We calculate the nested sums in layers. The inner sum over j_2 is evaluated in the first layer as

  $$f_1(k_0, j_1, j_0) = f(0, j_1, j_0) + f(1, j_1, j_0)\, W^{4k_0}$$

  for k_0, j_1 and $j_0 = 0, 1$. (8 complex '×' and '+')

- $$f_2(k_0, k_1, j_0) = f_1(k_0, 0, j_0) + f_1(k_0, 1, j_0)\, W^{(4k_1+2k_0)}$$

  for k_0, k_1 and $j_0 = 0, 1$ and

- $$f_3(k_0, k_1, k_2) = f_1(k_0, k_1, 0) + f_1(k_0, k_1, 1)\, W^{(4k_2+2k_1+k_0)}$$

  for k_0, k_1 and $k_2 = 0, 1$.

- Finally, 'bit reversal' to give

  $$g(k_2, k_1, k_0) = f_3(k_0, k_1, k_2).$$

  See program **FFT** as illustration.

Multi-Dimensional Fourier Series

- We can extend the ideas of Fourier analysis to higher dimensions, e.g., in 2D

$$f(x,y) = \sum_{n=-N/2}^{N/2} \sum_{m=-M/2}^{M/2} c_{n,m} e^{iq_n x} e^{iq_m y},$$

$$c_{n,m} = \int_{-X/2}^{X/2} \int_{-Y/2}^{Y/2} dxdy\, f(x,y)\, e^{i(q_n x + q_m y)},$$

$$q_n = \frac{2\pi n}{X} \quad \text{and} \quad q_m = \frac{2\pi m}{Y} \; ; X/Y \text{ are periods.}$$

- It is best to implement multi-dimensional Fourier transform by using the FFT algorithm.

Anderson Localization (optional)

- Only some of the particles move significantly (***extended***).
- The other remain essentially at rest (***localized***).
- Then there exists a threshold frequency, which is well defined for large systems. For 1D infinite chain with ***finite disorder*** all states are localised.

FPU (Fermi, Pasta, Ulam) Problem

- A set of linearly coupled oscillators is not **ergodic**.
- Because the system may stay in normal mode forever.
- Will the system be ergodic if the interaction between the particles is slightly nonlinear?

See another note

Wave Motion

- For coupled oscillators, the equation of motion is

$$M\frac{d^2u_j(t)}{dt^2} = -K\left[u_j(t)-u_{j+1}(t)\right]-K\left[u_j(t)-u_{j-1}(t)\right]$$

- In the limit $L \to \infty$, $a \to 0$, with the length La fixed
- Replacing $u_j(t)$ where j is a discrete variable, by the function $u(x,t)$ with continuous variables x and using the Taylor series expansion

$$u(x\pm a) = u(x) \pm a\frac{\partial u}{\partial x} + \frac{a^2}{2}\frac{\partial^2 u}{\partial x^2} + \cdots$$

$$\frac{1}{a^2}\left[u(x+a,t)-2u(x,t)+u(x-a,t)\right] \xrightarrow{a\to 0} \frac{\partial^2 u(x,t)}{\partial x^2}$$

The Continuous Wave Equation

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2} \quad \text{where } c^2 = Ka^2 / M.$$

We can introduce $\mu = M/a$ and $T = Ka$ so that $c^2 = T/\mu$.

- It is easy to see that any function of the form $f(x \pm ct)$ is a solution of the wave equation,
  e.g., $u(x,t) = A \cos 2\pi(x \pm ct)/\lambda$, $A \sin 2\pi(x \pm ct)/\lambda$.

- The wave equation is linear so it also satisfies the superposition principle and the behaviour of a wave of arbitrary shape can be represented by a sum of sinusoidal waves, similar to the Fourier series.

Numerical Solution of the Wave Equation

- Convert the continuum equation to a physically motivated discrete form frequently leads to useful numerical algorithms.
- Replace a by the spatial interval Δx and taking Δt to be the time step, we can rewrite the equation of motion

$$M\frac{d^2u_j(t)}{dt^2} = -K\left[u_j(t)-u_{j+1}(t)\right]-K\left[u_j(t)-u_{j-1}(t)\right]$$

as $$\frac{1}{(\Delta t)^2}\left[u(x,t+\Delta t)-2u(x,t)+u(x,t-\Delta t)\right]=$$

$$\frac{c^2}{(\Delta x)^2}\left[u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t)\right]$$

Numerical Solution of the Wave Equation

$$u(x, t+\Delta t) = -u(x, t-\Delta t) + 2(1 - b)u(x, t) + b[u(x+\Delta x, t) + u(x-\Delta x, t)]$$

- where $b \equiv (c\Delta t / \Delta x)^2$.
- **This formula serves as an updating algorithm.**
- With the initial conditions at $u(x,t_0)$ and the boundary conditions we can use this formula to find $u(x,t)$.
- Superposition of waves, reflection of waves, and standing waves can all be obtained from the wave equation.

Nonlinear Wave Equation (optional)

- Consider a system of L coupled oscillators with the Morse potential

$$V = \frac{1}{2}\sum_{j=1}^{L}\left(e^{-(u_j - u_{j-1})} - 1\right)^2$$

- The force on the j th particle is

$$F_j = -\frac{\partial V}{\partial u_j} = Q_j\left(1 - Q_j\right) - Q_{j+1}\left(1 - Q_{j+1}\right)$$

where $Q_j \equiv \exp[-(u_j - u_{j-1})]$

- There exist solutions that maintain their shape indefinitely, called ***solitons***.

Lecture 6 Review and Required

- **One-dimensional chain of coupled oscillators.**
- **Normal mode, Orthogonality, Eigenvalue equation**.
- **Fourier transform**, DFT, FFT, Power spectrum.
- Localization, Nonlinearity.
- Wave (*Interference*/*Diffraction*).
- Program **oscillator**, synthesize, *analyze*, *FFT*. (check "Simulation" book)