

# Numeričko rješavanje problema iz fizike

« Numeričke metode »

Ivo Batistić

Fizički odsjek, PMF  
Sveučilište u Zagrebu

predavanja 2005/2006

# Pregled predavanja

Određivanje perioda titranja

Gibanje u periodičnom potencijalu

Strujni krug

Put do Mjeseca i dalje

Kako napraviti mpeg film

Fazne trajektorije

Titranje molekula

## Određivanje perioda titranja

- ▶ Period titranja harmoničkog oscilatora (opruge, klatna) je jednostavno odrediti iz konstante elastičnosti  $k$ . Ako je energija:

$$E = \frac{p^2}{2m} + \frac{k x^2}{2},$$

period titranja je:

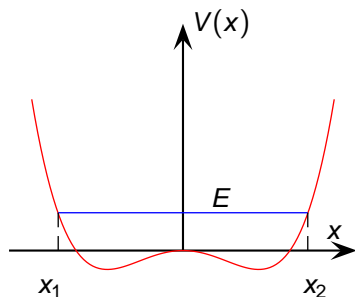
$$T = 2\pi\sqrt{\frac{m}{k}}.$$

- ▶ Što raditi ako nemamo harmonički oscilator, tj ako se čestica giba u nekom neobičnom potencijalu zadanim nekom funkcijom  $V(x)$ ?  
Npr.

$$V(x) = a x^2 + b x^4,$$

gdje su  $a$  i  $b$  neki parametri.

## Kvartični oscilator



Ako sa plavom linijom označimo iznos energije čestice, tada presjecišta te linije i potencijalne energije označavaju točke između kojih se čestica giba. U tačkama  $x_1$  i  $x_2$  čestica ima kinetičku energiju jednaku nuli, stoga, i brzinu jednaku nuli. U tim tačkama brzina mijenja svoj smjer, i čestica se počinje vraćati nazad.

Kako izračunati period ?

Iz definicije brzine:

$$\frac{dx}{dt} = v(x) \quad \Rightarrow \quad dt = \frac{dx}{v(x)} \quad \Rightarrow \quad t_2 - t_1 = \int_{x_1}^{x_2} \frac{dx}{v(x)}$$

## Kvartični oscilator

Period titranja je dvostruko vrijeme potrebno čestici da pređe put između točaka  $x_1$  i  $x_2$ . Sve što je potrebno je izračunati integral:

$$T = 2.0 \int_{x_1}^{x_2} \frac{dx}{v(x)},$$

gdje je ovisnost brzine o položaju zadana preko potencijalne energije  $V(x)$ :

$$v(x) = \sqrt{\frac{2}{m}} \sqrt{E - V(x)} = \sqrt{\frac{2}{m}} \sqrt{E - a x^2 - b x^4}$$

Račun možemo provesti numerički koristeći programe za integraciju iz CERNove biblioteke (DGAUSS). A za određivanje točaka u kojima je brzina jednaka nuli, i gdje čestica mijenja smijer gibanja ( $x_1$ ,  $x_2$ ), koristit ćemo CERNov programe za određivanje nula funkcije DZEROX.

# Kvartični oscilator

```
MODULE parametri_01
  REAL(KIND=8) :: a=-0.5_8, b=0.25_8 ! parametri potencijala
  REAL(KIND=8) :: e=1.0_8           ! energija
  REAL(KIND=8) :: eps = 1.0D-5      ! greska
  INTEGER       :: mxit = 100        ! broj iteracija
  INTEGER       :: nacin = 1         ! nacin trazenja nula
END MODULE

PROGRAM period_gibanja
  USE parametri_01
  REAL(KIND=8)          :: t1,t2
  REAL(KIND=8)          :: period,nula
  REAL(KIND=8), EXTERNAL :: brzina,kvadrat_brzine,DGAUSS,DZEROX

  ! »» Trazenje x_2 «««
  t1 = 0.0_8; t2= 3.0_8      ! trazimo nulu u ovom podrucju
  nula = DZEROX(t1,t2,eps,mxit,kvadrat_brzine,nacin)
```

# Kvartični oscilator

```
t1 = 0.0_8
t2 = nula
period = 4.0_8*DGAUSS(brzina,t1,t2,eps)
PRINT *,'Period gibanja je ',period
END PROGRAM
```

```
FUNCTION kvadrat_brzine(x) RESULT (y)
  USE parametri_01
  IMPLICIT NONE
  REAL(KIND=8) :: x,y
  y = e - a*x*x - b*x*x*x*x
END FUNCTION
```

```
FUNCTION brzina(x) RESULT (y)
  USE parametri_01
  IMPLICIT NONE
  REAL(KIND=8) :: x,y
  y = 1.0_8/SQRT(2.0_8*(e - a*x*x - b*x*x*x*x))
END FUNCTION
```

## Kvartični oscilator

- ▶ Program izračunava period gibanja za česticu jedinične mase koja ima energiju  $E = 1.0$  (u jedinicam u kojima je potencijal zadan) te za parametre  $a = -0.5$  i  $b = 0.25$ .

```
prompt> ./01_period.x
```

```
Period gibanja je      5.63387171025670
```

- ▶ Da smo pretpostavili da je  $a = +0.5$  i  $b = 0.0$  dobili bi da je period jednak  $2\pi = 6.28318\dots$ , jer to odgovara gibanju harmoničkog oscilatora konstante elastičnosti  $k = 1$ .
- ▶ U slučaju harmoničkog oscilatora period ne ovisi o energiji čestice. Period je isti za sve je energije. To je posebno svojstvo harmoničkog oscilatora. Period titranja za ostale (anharmoničke) potencijale ovisi o energiji čestice.

## Gibanje u periodičnom potencijalu

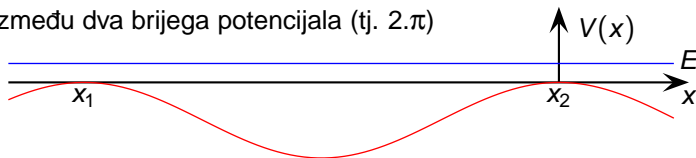
- Ista formula koju smo izveli za proračun period može nam poslužiti za proračun vremena koje je čestici potrebno da pređe put između dviju točaka (kod 1d-gibanja).

$$t(x_1 \rightarrow x_2) = \int_{x_1}^{x_2} \frac{dx}{v(x)}$$

- Pretpostvit ćemo da se čestica giba u periodičnom potencijalu:

$$V(x) = -1 + \cos(x).$$

Želimo odrediti koliko je vremena čestici potrebno da pređe put između dva brijega potencijala (tj.  $2\pi$ )



# Gibanje u periodičnom potencijalu

```
MODULE parametri_02
  REAL(KIND=8) :: pi = 3.141592653580_8
  REAL(KIND=8) :: e = 0.05_8           ! energija
  REAL(KIND=8) :: eps = 1.0D-5        ! greska
END MODULE

PROGRAM period_gibanja
  USE parametri_02
  REAL(KIND=8) :: t1,t2
  REAL(KIND=8) :: period
  REAL(KIND=8), EXTERNAL :: brzina,DGAUSS
  t1 = 0.0_8
  t2 = 2.0_8*pi
  period = DGAUSS(brzina,t1,t2,eps)
  PRINT *, 'Vrijeme potrebno za preciznost 2Pi je ', period
END PROGRAM

FUNCTION brzina(x) RESULT (y)
  USE parametri_02
  REAL(KIND=8) :: x,y
  y = 1.0_8/SQRT(2.0_8*( e + 1.0_8 - COS(x) ))
END FUNCTION
```

## Gibanje u periodičnom potencijalu

- ▶ Za energiju  $e = 0.05$  dobivamo:

```
prompt> ./02_vrijeme.x
```

```
Vrijeme potrebno za preciznost  $2\pi$  je      6.43394084079885
```

- ▶ Zanimljivo je ustanoviti kako to vrijeme ovisi o energiji. Program možemo modificirati i izračunati vrijeme za različite energije:

```
prompt> ./03_vrijeme.x
```

```
Energija      2.00000E+00 period      2.62206E+00
```

```
Energija      1.00000E+00 period      3.31328E+00
```

```
Energija      5.00000E-01 period      4.03781E+00
```

```
Energija      2.50000E-01 period      4.76802E+00
```

```
.....
```

```
Energija      3.05176E-05 period      1.38629E+01
```

```
Energija      1.52588E-05 period      1.45561E+01
```

```
Energija      7.62939E-06 period      1.52492E+01
```

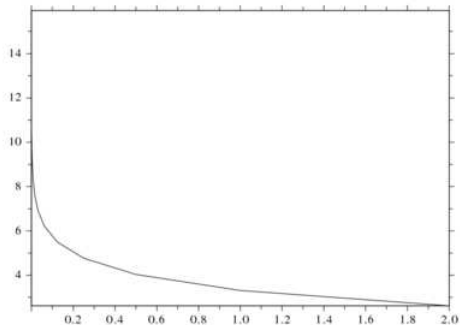
```
Energija      3.81470E-06 period      1.59424E+01
```

## Gibanje u periodičnom potencijalu

Očito period divergira kako je energija sve manja i manja. Možemo program modificirati tako da nam rezultat prikaže grafički. U program ubacujemo linije:

```
CALL METAFL('xwin')  
CALL QPLOT(REAL(energija,KIND=4),REAL(period,KIND=4), &  
          SIZE(energija))
```

za crtanje niza, a kao rezultat dobivamo:



Jel moguće ustanoviti kako period divergira ?

Napravimo plot perioda kao funkciju logaritma energije.



## Gibanje u periodičnom potencijalu

Da bi ustanovili točnu funkcionalnu ovisnost, iskoristit ćemo DLSQP1 podprogram iz CERNove biblioteke, kako bi za dani skup vrijednosti funkcije,  $y_i$  u točkama  $x_i$  pronašli pravac, zadan s koefijentima  $a$  i  $b$  za koji je funkcija:

$$F(a, b) = \sum_i (y_i - a \cdot x_i - b)^2$$

minimalna. Dakle u program ubacujemo liniju:

```
CALL DLSQP1(SIZE(energija), LOG(energija), period, b, a, sd, ifail)
```

koja će izračunati koeficijente pravca  $a$  i  $b$ . I kao rezultat dobivamo:

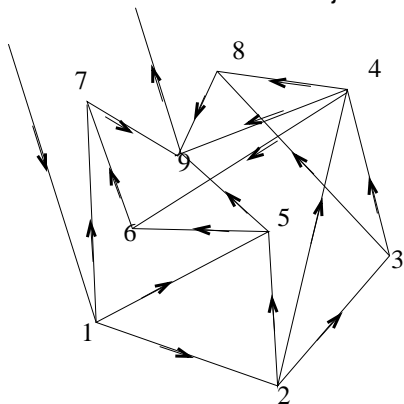
$$a = -1.01024 \quad b = 3.37350$$

Dakle:

$$\text{vrijeme} = 1.01024 \cdot \ln\left(\frac{1.0}{\text{energija}}\right) + 3.37350.$$

# Strujni krug - klupko otpornika

Pretpostavimo da imamo  $n$  čvorišta međusobno povezanih otpornicima. Čvorišta ćemo označiti indeksima od 1 do  $n$ . Neka je prvo čvorište ono u koje ulazi struja, a zadnje čvorište ono iz kojeg izlazi. Poredak u označavanju ostalih čvorista nije važan.



- ▶ Otpor između čvorišta  $i$  i  $j$  označimo sa  $R_{ij}$ .
- ▶ Ako nema otpornika koji povezuje  $i$  i  $j$ , to znači da je otpor beskonačan.
- ▶ Neka je smjer struje uvijek od čvorišta manjeg indeksa prema čvorištu većeg indeksa.
- ▶ Neka je napon na  $i$ -tom čvorištu jednak  $U_i$ .

# Strujni krug

Kao izračunati otpor klupka ?

- ▶ Ukupna struja koja ulazi u svaki čvor jednaka je ukupnoj struji koja iz njega izlazi:

$$\sum_{i < j} I_{ij} = \sum_{i > j} I_{ji},$$

osim za prvi čvor u koji ulazi struja iz vanjske mreže, te za zadnji čvor iz kojeg struja izlazi u vanjsku mrežu:

$$I = \sum_{i > 1} I_{1i}$$
$$\sum_{i < n} I_{in} = I$$

- ▶ Iznos struje između svaka dva čvorišta je:

$$I_{ij} = \frac{U_j - U_i}{R_{ij}}$$

## Strujni krug

Naponi u čvorištima zadovoljavaju skup linearnih jednažbi:

$$a_{11}U_1 + a_{12}U_2 + \dots + a_{1n}U_n = I$$

$$a_{21}U_1 + a_{22}U_2 + \dots + a_{2n}U_n = 0$$

$$a_{31}U_1 + a_{32}U_2 + \dots + a_{3n}U_n = 0$$

...

$$a_{n1}U_1 + a_{n2}U_2 + \dots + a_{nn}U_n = -I$$

gdje su:

$$a_{ij} = \frac{1}{R_{ij}} \quad \text{za } i \neq j$$

$$a_{ii} = \sum_{j < i} \frac{1}{R_{ij}} - \sum_{j > i} \frac{1}{R_{ij}}$$

Rješavanjem sustava jednažbi za napone na čvorištima možemo odrediti otpor strujnog kruga.

# Strujni krug

```
PROGRAM strujni_krug
  CHARACTER (LEN=80) :: datoteka = '05_strujni_krug.dat'
  REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: vv,bb
  REAL(KIND=8), ALLOCATABLE, DIMENSION(:, :) :: aa,rr
  INTEGER, ALLOCATABLE, DIMENSION(:) :: ir
  REAL(KIND=8) :: xx
  INTEGER :: mm
  INTEGER :: i,j,k,ifail,ierr

  OPEN (UNIT=10,FILE=TRIM(datoteka),STATUS='old', &
        ACTION='read',ERR=20,IOSTAT=ierr)
  READ (10,*) mm ! broj cvorista

  ALLOCATE (vv(mm))
  ALLOCATE (bb(mm))
  ALLOCATE (aa(mm,mm))
  ALLOCATE (rr(mm,mm))
  ALLOCATE (ir(mm))
```

# Strujni krug

```
rr = 0.0_8
DO
  READ (10,FMT='(2i3,g)',END=30,EOR=30,ADVANCE='yes') i,j,xx
  rr(i,j) = xx
END DO

20 IF (ierr /= 0) THEN
  PRINT *, 'Greska kod otvaranja = ',ierr
  STOP
END IF

30 CLOSE(10)

DO i=1,mm
DO j=i+1,mm
  rr(j,i) = rr(i,j)
END DO
END DO
```

# Strujni krug

```
DO i=1,mm
  DO j=1,i-1
    IF (rr(i,j) /= 0.0_8) aa(i,j) = 1.0/rr(i,j)
  END DO
  DO j=i+1,mm
    IF (rr(i,j) /= 0.0_8) aa(i,j) = 1.0/rr(i,j)
  END DO
  aa(i,i) = - SUM(aa(i,1:(i-1))) - SUM(aa(i,(i+1):mm))
END DO
aa(1,1) = aa(1,1) + 1.0
bb = 0.0_8 ; bb(1) = +1.0_8 ; bb(8) = -1.0_8
k = 1
vv = bb
CALL DEQN (mm,aa,mm,ir,ifail,k,vv)
IF (ifail == -1) THEN
  PRINT *, 'A je singularna'
ELSE
  PRINT '(lg)', vv
END IF
END PROGRAM
```

## Strujni krug

U problemu strujnog kruga postoji neodređenost. Iznosi napona,  $U_i$ , nisu apsolutno određeni, nego samo njihove međusobne razlike. To se u problemu pojavljuje tako da je jedna od jednačbi linearno ovisna o drugim. Da bi *popravili* tu situaciju, sustavu jednačbi dodali smo još jednu jednačbu (tj. pribrojili smo je prvoj):

$$1 \cdot U_1 = 0$$

Time se prisiljava da vrijednost potencijala na prvom čvorištu bude jednaka nuli.

```
prompt> ./05_strujni_krug.x
```

```
-0.5551115123125783E-15
```

```
0.4787666433076381
```

```
1.251164516262006
```

```
1.789455459829342
```

```
0.3693474586751307
```

```
1.034172884290367
```

```
2.698544869945176
```

```
6.803808895667588
```

Problem se lako može poopćiti na složene krugove koji sadrže kondenzatore i zavojnice, te kroz koje teče izmjenična struja.

## Put do Mjeseca i dalje

Želimo promotriti problem putovanja rakete (projektila) do Mjeseca. Gibanje je opisano diferencijalnim jednažbama koje je potrebno riješiti:

$$\frac{dx}{dt} = v_x$$

$$\frac{dv_x}{dt} = \frac{1}{m} F_x(x, y, t)$$

$$\frac{dy}{dt} = v_y$$

$$\frac{dv_y}{dt} = \frac{1}{m} F_y(x, y, t)$$

pretpostavit ćemo da imamo samo gibanje u ravnini.

Rješavanjem sustava jednažbi dobiva se kako položaj  $(x, y)$  i brzine  $(v_x, v_y)$  ovise o vremenu. Pretpostavit ćemo da je početni položaj rakete površina Zemlje, te da je ispaljena s brzinom od oko 11 km/s.

## Put do Mjeseca i dalje

Sile koje djeluju na raketu su gravitacija Zemlje i gravitacija Mjeseca:

$$F_x = -\frac{G m M_Z x}{(x^2 + y^2)^{(3/2)}} - \frac{G m M_M (x - x_M(t))}{((x - x_M(t))^2 + (y - y_M(t))^2)^{(3/2)}}$$
$$F_y = -\frac{G m M_Z y}{(x^2 + y^2)^{(3/2)}} - \frac{G m M_M (y - y_M(t))}{((x - x_M(t))^2 + (y - y_M(t))^2)^{(3/2)}}$$

U CERNovoj biblioteci postoje potprogrami (DDEQBS) koje možemo koristiti u računu.

## Put do Mjeseca i dalje

```
MODULE parametri_06
  REAL(kind=KIND(1.0D0)), PARAMETER :: pi = 3.14159265350_8
  REAL(kind=KIND(1.0D0)), PARAMETER :: zm = 384000_8          ! km
  REAL(kind=KIND(1.0D0)), PARAMETER :: tm = 27.32*24*3600    ! s
  REAL(kind=KIND(1.0D0)), PARAMETER :: omega = 2.0_8*pi/tm   ! s^-1
  REAL(kind=KIND(1.0D0)), PARAMETER :: phi = 0.77_8
  REAL(kind=KIND(1.0D0)), PARAMETER :: gz = 398432_8         ! km^3/s^2
  REAL(kind=KIND(1.0D0)), PARAMETER :: gm = 4905.32_8       ! km^3/s^2
END MODULE

PROGRAM raketa
  USE parametri_06
  INTEGER, PARAMETER :: n = 4
  REAL(kind=KIND(1.0D0)) :: t,t0,t1,dt,tmx
  REAL(kind=KIND(1.0D0)) :: y1,y2,y3,y4
  REAL(kind=KIND(1.0D0)), DIMENSION(n) :: y
  REAL(kind=KIND(1.0D0)) :: h0
  REAL(kind=KIND(1.0D0)) :: eps,del
  REAL(kind=KIND(1.0D0)), DIMENSION(40*n) :: work
  INTEGER :: i, imax = 10000000
  REAL(kind=KIND(1.0D0)) :: eng
  REAL(kind=KIND(1.0D0)), DIMENSION(2) :: rm ! polozaj mjeseca
  EXTERNAL :: postavi_silu
```

## Put do Mjeseca i dalje

```
REAL :: dm = 800000.0
REAL :: dd = 100000.0

t = 0.0_8           ! pocetak
dt = 3600.0_8       ! vremenski korak
tmx = 800000        ! maksimalno vrijeme
h0 = 1.0_8
eps = 1.0D-7
del = 1.0D-3

y1 = 6370_8         ! inicijalni polozaji i brzine
y2 = 11.1_8
y3 = 0.0_8
y4 = 0.0_8

rm(1) = zm*COS(omega*t-phi) ! polozaj Mjeseca
rm(2) = zm*SIN(omega*t-phi)

!!«« pocetak grafickog prikaza »»!!
```

## Put do Mjeseca i dalje

```
i = 0
DO WHILE (t<tmx)
  t0 = t
  t1 = t+dt
  y(1) = y1; y(2) = y2; y(3) = y3; y(4) = y4
  CALL DDEQBS (n,t0,t1,y,h0,eps,postavi_silu,work)
  t = t1
  y1 = y(1); y2 = y(2); y3 = y(3); y4 = y(4)
  i = i+1
  rm(1) = zm*COS(omega*t-phi)
  rm(2) = zm*SIN(omega*t-phi)
  eng = 0.5*(y(2)**2+y(4)**2) - gz/SQRT(y(1)**2+y(3)**2) &
        - gm/SQRT((y(1)-rm(1))**2+(y(3)-rm(2))**2)
  !! ««« grafiski prikaz rakete i Mjeseca »»» !!
END DO
WRITE (*,'(1x,a10,g12.5,a7,g12.5)') 'vrijeme = ',t, &
                                     ' eng = ',eng
END PROGRAM
```

## Put do Mjeseca i dalje

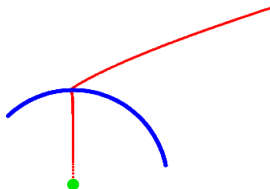
```
SUBROUTINE postavi_silu(t,y,sila)
  USE parametri_06
  REAL(kind=KIND(1.0D0)),INTENT(IN)                :: t
  REAL(kind=KIND(1.0D0)),INTENT(IN), DIMENSION(4) :: y
  REAL(kind=KIND(1.0D0)),INTENT(OUT),DIMENSION(4) :: sila

  REAL(kind=KIND(1.0D0)), DIMENSION(2)            :: rm

  rm(1) = zm*COS(omega*t-phi)
  rm(2) = zm*SIN(omega*t-phi)

  sila(1) = +y(2)
  sila(2) = -gz*y(1)/(y(1)**2+y(3)**2)**1.5 &
    -gm*(y(1)-rm(1))/((y(1)-rm(1))**2+(y(3)-rm(2))**2)**1.5
  sila(3) = +y(4)
  sila(4) = -gz*y(3)/(y(1)**2+y(3)**2)**1.5 &
    -gm*(y(3)-rm(2))/((y(1)-rm(1))**2+(y(3)-rm(2))**2)**1.5
END SUBROUTINE
```

## Put do Mjeseca i dalje



Kao rezultat dobivamo da raketa u blizini Mjeseca ulijeće u hiperboličnu putanju iz koje izlijeće s dodatnom energijom koja joj omogućuje da se oslobodi gravitacijskog polja i Zemlja i Mjeseca.

Ispis iz programa:

```
vrijeme = 0.0000      eng = -0.95612    (negativna energija!)  
vrijeme = 0.80280E+06 eng = 0.81084    (pozitivna energija!)
```

Moguće je napraviti slikice položaja rakete u raznim vremenskim trenucima, te ih sve zajedno objediniti u film. (verzija programa 07\_raketa.f90 generira oko 223 slika). Ima raznih alata za povezivanje slika u film. Ovdje smo iskoristili mencoder (<http://www.mplayerhq.hu/>)

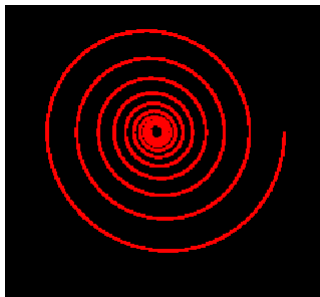
## Fazne trajektorije

- ▶ U fizici je uobičajeno prikazivati kretanje čestica u tzv. faznom prostoru (ili faznom dijagramu). Fazni prostor je prostor u kojem su koordinatne osi koordinate i impulsi (brzine). U slučaju jednodimenzionalnog gibanja to je dvodimenzionalni prostor  $(x, p_x)$ .
- ▶ Skup točaka  $(x(t), p_x(t))$  predstavlja krivulju koja se nigdje ne presjeca sama sa sobom (u faznom prostoru). Ta se krivulja zove **fazna trajektorija**. Ako vrijedi zakon sačuvanja energije, onda je fazna trajektorija krivulja konstantne energije.

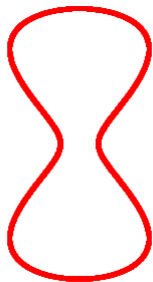
$$\frac{p_x^2}{2m} + V(x) = E$$

- ▶ Program koji smo razvili za gibanje rakete može se modificirati za praćenje gibanja čestice u faznom prostoru.

## Fazne trajektorije



Fazna trajektorija gušenog harmoničkog oscilatora. Energija nije sačuvana. Ako ne bi bilo gušenja, fazna trajektorija bi bila elipsa.



Fazna trajektorija kvartičnog oscilatora čija je potencijalna energija:

$$V(x) = -0.5x^2 + 0.25x^4.$$

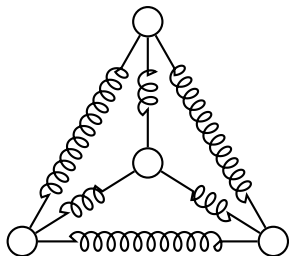
Trajektorija predstavlja krivulju konstantne energije. Kod periodičnog gibanja trajektorija je zatvorena krivulja.

# Titranje molekula

Promotrit ćemo jednostavan slučaj (jednodimenzionalni) grupe atoma vezanih elastičnim silama (oprugama). Gibanje (titranje) molekule može se prikazati kao superpozicija titranja nekoliko osnovnih frekvencija:

$$x_i(t) = \sum_{\lambda} a_{\lambda} \cos(\omega_{\lambda} \cdot t - \phi_{\lambda}).$$

Broj različitih frekvencija jednak je broju atoma koje čine molekulu. Želimo izračunati frekvencije titranja  $\omega_{\lambda}$ .



Potencijalna energija skupa čestica vezanih oprugama različite jačine:

$$V(x_1, x_2, \dots) = \sum_{i < j} \frac{K_{ij}}{2} \cdot (x_i - x_j)^2$$

# Titranje molekula

Jednadžbe gibanja:

$$m\ddot{x}_1 = -\frac{dV}{dx_1} = -\sum_{j \neq 1} K_{1,j}(x_1 - x_j)$$

$$m\ddot{x}_2 = -\frac{dV}{dx_2} = -\sum_{j \neq 2} K_{2,j}(x_2 - x_j)$$

....

Pretpostavit ćemo da čestice titraju nekom nepoznatom frekvencijom titranja  $\omega$ :

$$x_i(t) = x_i \cdot e^{-i\omega t}$$

Uvršavanjem u sustav diferencijalnih jednadžbi, dobiva se:

$$m\omega^2 x_1 = \sum_j D_{1j} x_j$$

$$m\omega^2 x_2 = \sum_j D_{2j} x_j$$

...

$$m\omega^2 x_n = \sum_j D_{nj} x_j$$

ili

$$m\omega^2 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = (D_{ij}) \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

## Titranje molekula

To je u stvari problem određivanja vlastiti vrijednosti i vlastitih vektora matrice  $D_{ij}$ . Pri tome matricni elementi matrice  $D$  su:

$$D_{ij} = -K_{ij} \quad \text{za} \quad i \neq j$$
$$D_{ii} = \sum_{j \neq i} K_{ij}$$

U CERNovoj biblioteci ne postoje programi za određivanja vlastitih vrijednosti, nego se CERNova biblioteka oslanja na druge pakete, kao npr. LAPACK. Web adresa LAPACK paketa je

<http://www.netlib.org/lapack/>. To je paket koji ima veliki broj raznih programa za rad s matrica. Mnogi od programa su specijalizirani za rad s posebnim vrstama matrica (simetrična, nesimetrična, tridijagonalna, itd.).

Mi ćemo iskoristiti DSYEV program koji izračunava vlastite vrijednosti i vlastite vektore simetrične matrice.

# Titranje molekula

```
PROGRAM frekvencije
  CHARACTER (LEN=80) :: datoteka = '10_titranje.dat'
  REAL(KIND=8), ALLOCATABLE, DIMENSION(:, :) :: kk
  REAL(KIND=8), ALLOCATABLE, DIMENSION(:)    :: omega
  REAL(KIND=8), ALLOCATABLE, DIMENSION(:)    :: work
  REAL(KIND=8)  :: xx
  INTEGER       :: mm, lwork
  INTEGER       :: i, j, k, ifail, ierr

  OPEN (UNIT=10, FILE=TRIM(datoteka), STATUS='old', &
        ACTION='read', ERR=20, IOSTAT=ierr)

  READ (10, *) mm      ! broj cvorista
  lwork = 4*mm

  ALLOCATE (kk(mm, mm))
  ALLOCATE (omega(mm))
  ALLOCATE (work(lwork))
```

# Titranje molekula

```
kk = 0.0_8
DO
  READ (10,FMT='(2i3,g)', END=30, EOR=30, ADVANCE='yes') i,j,xx
  kk(i,j) = xx
END DO
20 IF (ierr /= 0) THEN
  PRINT *, 'Greska kod otvaranja = ', ierr
  STOP
END IF
30 CLOSE(10)
DO i=1,mm
DO j=i+1,mm
  kk(j,i) = kk(i,j)
END DO
END DO
DO i=1,mm
  kk(i,i) = - SUM(kk(i,1:(i-1))) - SUM(kk(i,(i+1):mm))
END DO
```

# Titranje molekula

```
CALL DSYEV ('V', 'U', mm, -kk, mm, omega, work, lwork, ifail)
IF (ifail == -1) THEN
PRINT *, 'matrica je singularna'
ELSE
PRINT '(1g)', SQRT(omega)
END IF

END PROGRAM
```

Izvođenjem programa:

```
prompt> ./10_titranje.x
```

```
8.590409574352428E-008
```

```
2.51195265918648
```

```
2.70985412231195
```

```
3.48008414309677
```

```
3.71836026036897
```

```
4.55627576721137
```

```
5.36689460335591
```

```
6.54571533558111
```

Jedna od frekvencija je jednaka nuli. To odgovara gibanju u kojem se svi atomi se pomiču za isti iznos, tj. radi se translatorsnom gibanju cijele molekule.