Physics 562 - Computational Physics

Assignment 4: Eigenvalues of Simple Harmonic Oscillator

Josh Fernandes
Department of Physics & Astronomy
California State University Long Beach

April 8, 2014

Abstract

This paper examine

1 Hamiltonian

one dimensional harmonic oscillator is given by

$$m\frac{\partial^2 x}{\partial t^2} = -kx. \tag{1}$$

which has the solution

$$x = x_0 sin(wt + \theta), \tag{2}$$

where

$$w = \sqrt{\frac{k}{m}} \tag{3}$$

The total energy of the system is the same as the hamiltonian which is given by

$$E = \mathcal{H} = \frac{p^2}{2m} + \frac{mw^2x^2}{2} \tag{4}$$

let's say,

$$\xi = x\sqrt{\frac{mw}{\hbar}} \tag{5}$$

$$\pi = \frac{p}{\sqrt{\hbar mw}}. \tag{6}$$

This gives us the equation

$$\mathcal{H} = \frac{\hbar w}{2}(\pi^2 + \xi^2) \tag{7}$$

if you factorize the expression, it becomes

$$\mathcal{H} = \frac{\hbar w}{2}[(\xi + i\pi)(\xi - i\pi) + (\xi - i\pi)(\xi + i\pi)] \tag{8}$$

we can now define the operators

$$a = \frac{\xi + i\pi}{\sqrt{2}} = \frac{1}{\sqrt{2\hbar mw}}(mwx + ip) \tag{9}$$

$$a^\dagger = \frac{\xi - i\pi}{\sqrt{2}} = \frac{1}{\sqrt{2\hbar mw}}(mwx - ip) \tag{10}$$

from the commutator relation

$$[i\pi, \xi] = 1 \tag{11}$$

it follows that

$$[a, a^\dagger] = 1 \tag{12}$$

Finally, we can write the hamiltonian

$$\mathcal{H} = \hbar w(a^\dagger a + \frac{1}{2}) = \hbar w(N + \frac{1}{2}) \tag{13}$$

where

$$N = a^\dagger a \tag{14}$$

$$a^\dagger = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots \\ \sqrt{1} & 0 & 0 & 0 & \dots \\ 0 & \sqrt{2} & 0 & 0 & \dots \\ 0 & 0 & \sqrt{3} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

and

$$a = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \dots \\ 0 & 0 & \sqrt{2} & 0 & \dots \\ 0 & 0 & 0 & \sqrt{3} & \dots \\ 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

From this we get

$$x = \sqrt{\frac{\hbar}{2mw}}(a^\dagger + a), \tag{15}$$

$$p = i\sqrt{\frac{mw\hbar}{2}}(a^\dagger - a) \tag{16}$$

Plugging in for $a^\dagger$ and a we get

$$x = \sqrt{\frac{\hbar}{2mw}} \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \dots \\ \sqrt{1} & 0 & \sqrt{2} & 0 & \dots \\ 0 & \sqrt{2} & 0 & \sqrt{3} & \dots \\ 0 & 0 & \sqrt{3} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

and

$$p = i\sqrt{\frac{mw\hbar}{2}} \begin{pmatrix} 0 & -\sqrt{1} & 0 & 0 & \dots \\ \sqrt{1} & 0 & -\sqrt{2} & 0 & \dots \\ 0 & \sqrt{2} & 0 & -\sqrt{3} & \dots \\ 0 & 0 & \sqrt{3} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

So, we want to find the eigenvalue of a harmonic oscillator that is using the basis of a different basis. The first harmonic oscillator has the parameters of $\hbar = 1$, $m = 1$, and $w = .5$. The second harmonic oscillator has the parameters of $\hbar = 1$, $m = 1$, and $w = 1$. If you calculate the hamiltonians using equation **??** you get

$$\mathcal{H}_1 = (N + \frac{1}{2}) \tag{17}$$

$$\mathcal{H}_2 = .5 \cdot (N + \frac{1}{2}) \tag{18}$$

writing as a 10×10 matrix we get

$$\mathcal{H}_1 = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{5}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{7}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{9}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{11}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{13}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{15}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{17}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{19}{2} \end{pmatrix}$$

and

$$\mathcal{H}_2 = \begin{pmatrix} \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{3}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{5}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{7}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{9}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{11}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{13}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{15}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{17}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{19}{4} \end{pmatrix}$$

finally, we know that the hamiltonian we want is

$$\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2 + \frac{1}{2}m \cdot (\omega_2 - \omega_1)^2 \cdot (x_2 - x_1)^2 \tag{19}$$

where w_2 and x_2 are the matrices associated with the second harmonic oscillator, and w_1 and x_1 are the matrices associated with the first harmonic oscillator

2 The Fortran95 code

The code solves the equation of motion using the Runga-Kutta method. First a module called `NumType` is created to store all my global parameters.

Listing 1: Module `NumType`

```
module NumType

    save
    integer, parameter          ::  dp = kind(1.d0)
    real(dp), parameter         ::  pi = 4*atan(1._dp)
    complex(dp), parameter      ::  iic = (0._dp,1._dp),&
                                    one = (1._dp,0._dp),&
                                    zero = (0._dp,0._dp)

end module NumType
```

Listing 2: `mtest.f95`

```
module setup

    use NumType
    implicit none
    integer,    parameter   ::  ndim=10,lwork=5*ndim
    real(dp),   parameter   ::  mass=1.0_dp, &
                                hbar=1.0_dp,&
                                omega1=0.5_dp,&
                                omega2=1._dp

end module setup

program matrix

    use setup
    implicit none

    complex(dp), dimension(ndim,ndim)   :: A,B,E,H1,H2,H3
    real(dp),    dimension(ndim)        :: w
    integer                             :: i, nn, info
```

```
    complex(dp)                                 :: work(lwork)
    real(dp)                                    :: rwork(lwork)

    A(1:10,1:10) = reshape((/                                       &
            zero, sqrt(1*one), zero, zero, zero,                    &
            zero, zero, zero, zero, zero,                           &
            sqrt(1*one), zero, sqrt(2*one), zero, zero,             &
            zero, zero, zero, zero, zero,                           &
            zero, sqrt(2*one), zero, sqrt(3*one), zero,             &
            zero, zero, zero, zero, zero,                           &
            zero, zero, sqrt(3*one), zero, sqrt(4*one),             &
            zero, zero, zero, zero, zero,                           &
            zero, zero, zero, sqrt(4*one), zero,                    &
            sqrt(5*one), zero, zero, zero, zero,                    &
            zero, zero, zero, zero, sqrt(5*one),                    &
            zero, sqrt(6*one), zero, zero, zero,                    &
            zero, zero, zero, zero, zero,                           &
            sqrt(6*one), zero, sqrt(7*one), zero, zero,             &
            zero, zero, zero, zero, zero,                           &
            zero, sqrt(7*one), zero, sqrt(8*one), zero,             &
            zero, zero, zero, zero, zero,                           &
            zero, zero, sqrt(8*one), zero, sqrt(9*one),             &
            zero, zero, zero, zero, zero,                           &
            zero, zero, zero, sqrt(9*one), zero                     &
    /),                                                             &
    (/10,10/))

    H1(1:10,1:10) = reshape((/                                      &
            1/2._dp*one, zero, zero, zero, zero,                    &
            zero, zero, zero, zero, zero,                           &
            zero, 3/2._dp*one, zero, zero, zero,                    &
            zero, zero, zero, zero, zero,                           &
            zero, zero, 5/2._dp*one, zero, zero,                    &
            zero, zero, zero, zero, zero,                           &
            zero, zero, zero, 7/2._dp*one, zero,                    &
            zero, zero, zero, zero, zero,                           &
            zero, zero, zero, zero, 9/2._dp*one,                    &
            zero, zero, zero, zero, zero,                           &
            zero, zero, zero, zero, zero,                           &
            11/2._dp*one, zero, zero, zero, zero,                   &
```

```
            zero, zero, zero, zero, zero,                  &
            zero, 13/2._dp*one, zero, zero, zero,          &
            zero, zero, zero, zero, zero,                  &
            zero, zero, 15/2._dp*one, zero, zero,          &
            zero, zero, zero, zero, zero,                  &
            zero, zero, zero, 17/2._dp*one, zero,          &
            zero, zero, zero, zero, zero,                  &
            zero, zero, zero, zero, 19/2._dp*one           &
    /),                                                    &
    (/10,10/))

    H2(1:10,1:10) = reshape((/                             &
            1/4._dp*one, zero, zero, zero, zero,           &
            zero, zero, zero, zero, zero,                  &
            zero, 3/4._dp*one, zero, zero, zero,           &
            zero, zero, zero, zero, zero,                  &
            zero, zero, 5/4._dp*one, zero, zero,           &
            zero, zero, zero, zero, zero,                  &
            zero, zero, zero, 7/4._dp*one, zero,           &
            zero, zero, zero, zero, zero,                  &
            zero, zero, zero, zero, 9/4._dp*one,           &
            zero, zero, zero, zero, zero,                  &
            zero, zero, zero, zero, zero,                  &
            11/4._dp*one, zero, zero, zero, zero,          &
            zero, zero, zero, zero, zero,                  &
            zero, 13/4._dp*one, zero, zero, zero,          &
            zero, zero, zero, zero, zero,                  &
            zero, zero, 15/4._dp*one, zero, zero,          &
            zero, zero, zero, zero, zero,                  &
            zero, zero, zero, 17/4._dp*one, zero,          &
            zero, zero, zero, zero, zero,                  &
            zero, zero, zero, zero, 19/4._dp*one           &
    /),                                                    &
    (/10,10/))

    A(1:10,1:10)= sqrt(hbar/(2*mass*omega1))*A(1:10,1:10)
    B(1:10,1:10)= sqrt(hbar/(2*mass*omega2))*A(1:10,1:10)
    H3(1:10,1:10)= H1 + H2 + 1/2._dp*mass*&
                   (omega2-omega1)**2*(B-A)**2

```

```
    nn = 10
    info = 0
    E(1:nn,1:nn) = H3(1:nn,1:nn)

    call zheev('v','u',nn,E,ndim,w,work,lwork,rwork,info)

    print *, 'info=', info

    do i = 1,10
        print '(a,f15.8,a,20f6.0)','eigenvalues',w(i),&
        'vector', dble(e(1:nn,i))
    end do

end program matrix
```

The code is run by typing `./mat`. The resulting eigenvalues and eigenvectors are printed out to the terminal.

3 Results

The eigenvalues are

4 Summary and conclusions

The eigenvalues tell us that Natalie is super cool.

References

[1] M. Metcalf, J. Reid and M. Cohen, *Fortran 95/2003 explained.* Oxford University Press, 2004.