Homework 4, Computational Physics

Seann Smallwood

November 17, 2020

Abstract

This goal of this work was to analyze systems through matrices. The first was a quantum system for which we evaluated eigenvectors and values, performed matrix operations and compared the resulting matrices. The second, we used the companion matrix method to calculate the zeros of a given polynomial.

1 Introduction

1.1 Quantum Matrices

We used a quantum system with the total angular momentum quantum number $j = 5/2$. The ladder operators were defined as

$$\langle m^{'}|J_{\pm}|m\rangle = \sqrt{j(j+1) - m(m \pm 1)}\delta_{m',m\pm1}$$

Where m is the secodary quantum number ranging from $-j$ to j by ±1

We had three other matrices J_z, J_x, and J_y.

$$\langle m^{'}|J_z|m\rangle = m\delta_{m',m}$$

$$J_x = \frac{1}{2}(J_+ + J_-)$$

$$J_y = \frac{1}{2i}(J_+ - J_-)$$

The objective for this problem was to calculate the eigenvalues and eigenvectors of J_x, J_y, and $J^2 = J_x^2 + J_y^2 + J_z^2$.

We then calculated $e^{iJ_y\delta} * e^{iJ_x\delta} * e^{-iJ_y\delta} * e^{-iJ_x\delta}$ and compared to $e^{iJ_z\delta^2}$ for $\delta = \pi/10, \pi/20$.

1.2 Companion Matrix

Given the polynomial

$$P_n(t) = (n + 1) + (n)t + (n - 1)t^2 + ... + 2t^{n-1} + t^n$$

our objective was to find the zeros of $P_n(t)$ using the companion matrix method for $n = 20, 25, 30$.

1.3 Setup and General Methods

Fortran was used to analyze each system. To implement common mathematical terms and define the precision to which values were calculated a file was created. This file, named numtype, is shown below.

```
module numtype

integer,parameter :: dp = selected_real_kind(15,307)
integer,parameter :: qp = selected_real_kind(33,4931)
real(dp), parameter :: pi = 4*atan(1._dp)
complex(dp), parameter :: iic = (0._dp,1._dp)

end module numtype
```

A Makefile was used to compile the fortran code and create and executable file.

```
OBJS1 = numtype.o prob2.o

PROG1 = run

F90 = gfortran

F90FLAGS = -O3 -funroll-loops  -fexternal-blas

LIBS = -framework Accelerate

LDFLAGS = $(LIBS)

all: $(PROG1)

$(PROG1): $(OBJS1)
$(F90) $(LDFLAGS) -o $@ $(OBJS1)
```

```
clean:
rm -f $(PROG1) *.{o,mod} fort.*

.SUFFIXES: $(SUFFIXES) .f90

.f90.o:
$(F90) $(F90FLAGS) -c $<
```

2 Solutions

2.1 Quantum Matrices

The ladder operators $J_\pm$ and matrices J_x, J_y, and $J^2 = J_x^2 + J_y^2 + J_z^2$ were generated as shown in the code below. The internal LAPACK routine zheev was then used to calculate the eigenvalues and eigenvectors of the matrices.

To compare the exponential matrices $e^{iJ_y\delta} * e^{iJ_x\delta} * e^{-iJ_y\delta} * e^{-iJ_x\delta}$ and $e^{iJ_z\delta^2}$ the following power series expansion was used

$$e^X = \sum_{n=0}^{\infty} X^n/n!$$

Where X was our $6x6$ matrices, and the sum was ran to 20 terms.

```
program prob1

use numtype
implicit none

integer, parameter :: ndim = 6, lwork = 4*ndim          !2*j + 1 (spin states)
complex(dp), dimension(ndim,ndim) :: j_plus, j_minus, j_x, j_y, j_z, j, j_2, sum_j_x,
complex(dp) ::  i, work(lwork)
real(dp), dimension(ndim) :: w_x, w_y, w_2
real(dp) :: jj, mm, kplus, kminus, rwork(lwork), alpha, in
integer :: n, info, ii

i = (0._dp,1._dp)
jj = 5._dp/2._dp

mm = -jj
```

```
do n = 1, ndim                          !loop to generate J_z
    J_z(n,n) = cmplx(mm,0._dp)
    mm = mm + 1._dp

end do

j_plus = 0._dp                              !initialize matrices
j_minus = 0._dp
j_x = 0._dp
j_y = 0._dp
j_2 = 0._dp
sum_j_x = 0._dp
sum_j_y = 0._dp
sum_j_z = 0._dp
sum_j_xn = 0._dp
sum_j_yn = 0._dp

mm = -jj

if ((jj*(jj+1._dp))-(mm*(mm+1._dp)) >= 0._dp) then  !first j_plus

    kplus = sqrt((jj*(jj+1._dp))-(mm*(mm+1._dp)))
    j_plus(2,1) = cmplx(kplus, 0._dp)

else

    kplus = sqrt(-1._dp*((jj*(jj+1._dp))-(mm*(mm+1._dp))))
    j_plus(2,1) = cmplx(0._dp, kplus)
end if


mm = mm + 1._dp

Do n = 2, ndim - 1                          !loop to generate middle jplus/minus

    if ((jj*(jj+1._dp))-(mm*(mm+1._dp)) >= 0._dp) then

        kplus = sqrt((jj*(jj+1._dp))-(mm*(mm+1._dp)))
        j_plus(n+1,n) = cmplx(kplus, 0._dp)
```

```
        else

            kplus = sqrt(-((jj*(jj+1._dp))-(mm*(mm+1._dp))))
            j_plus(n+1,n) = cmplx(0._dp, kplus)
        end if

        if ((jj*(jj+1._dp))-(mm*(mm-1._dp)) >= 0._dp) then

            kminus = sqrt((jj*(jj+1._dp))-(mm*(mm-1._dp)))
            j_minus(n-1,n) = cmplx(kminus, 0._dp)

        else

            kminus = sqrt(-((jj*(jj+1._dp))-(mm*(mm-1._dp))))
            j_minus(n-1,n) = cmplx(0._dp, kminus)
        end if

        mm = mm + 1._dp
    end do

    if ((jj*(jj+1._dp))-(mm*(mm-1._dp)) >= 0._dp) then !last j_min

        kminus = sqrt((jj*(jj+1._dp))-(mm*(mm-1._dp)))
        j_minus(ndim-1,ndim) = cmplx(kminus, 0._dp)

    else

        kminus = sqrt(-1._dp*((jj*(jj+1._dp))-(mm*(mm-1._dp))))
        j_minus(ndim-1,ndim) = cmplx(0._dp, kminus)
    end if


    j_x = (j_plus + j_minus)/2.0_dp
    j_y = (j_plus - j_minus)/(2.0_dp*i)

    j_2 = (j_x ** 2) + (j_y ** 2) + (j_z ** 2)

!---------------------------------------------------------

    alpha = pi/10
```

```
in = 1
do n = 1, 20

sum_j_x = sum_j_x + ((i*j_x*alpha)**in)/product((/(ii,ii=1,n)/))
sum_j_y = sum_j_y + ((i*j_y*alpha)**in)/product((/(ii,ii=1,n)/))

sum_j_xn = sum_j_xn + ((-i*j_x*alpha)**in)/product((/(ii,ii=1,n)/))
sum_j_yn = sum_j_yn + ((-i*j_y*alpha)**in)/product((/(ii,ii=1,n)/))

sum_j_z = sum_j_z + ((i*j_y*(alpha**2))**in)/product((/(ii,ii=1,n)/))

in = in + 1
end do

lhs = sum_j_y*sum_j_x*sum_j_yn*sum_j_xn

do n = 1, ndim
    !print '(36f7.2)',j_plus(n,1:ndim)
    !print '(36f7.2)',j_minus(n,1:ndim)
    !print '(36f7.2)', j_x(1:ndim,n)
    !print '(36f7.2)', j_y(1:ndim,n)
    !print '(36f7.2)', j_z(1:ndim,n)
    print '(12f7.2)', j_2(1:ndim,n)
    !print '(12f7.2)', lhs(1:ndim,n)
end do

print *, '----------------------'

info = 0
call zheev ( 'v', 'u', ndim , j_x, ndim, w_x, work, lwork, rwork, info )
if( info /= 0 ) stop ' info /= 0 '
!print *,w_x(1:ndim)

info = 0
call zheev ( 'v', 'u', ndim , j_y, ndim, w_y, work, lwork, rwork, info )
if( info /= 0 ) stop ' info /= 0 '
!print *,w_y(1:ndim)

info = 0
call zheev ( 'v', 'u', ndim , j_2, ndim, w_2, work, lwork, rwork, info )
```

```
    if( info /= 0 ) stop ' info /= 0 '
    !print *,w_2(1:ndim)

    do n = 1, ndim
        !print '(10f7.2)', w_x(n)      !eigenvalues/vectors for j_x
        !print '(36f7.2)', j_x(1:ndim,n)

        !print '(10f7.2)', w_y(n)      !eigenvalues/vectors for j_y
        !print '(36f7.2)', j_y(1:ndim,n)

        print '(10f7.2)', w_2(n)      !eigenvalues/vectors for j_y
        print '(12f7.2)', j_2(1:ndim,n)
    end do

    do n = 1, ndim
        !print '(36f7.2)',j_plus(n,1:ndim)
        !print '(36f7.2)',j_minus(n,1:ndim)
        !print '(36f7.2)', j_x(1:ndim,n)
        !print '(36f7.2)', j_y(1:ndim,n)
        !print '(36f7.2)', j_z(1:ndim,n)
        !print '(36f7.2)', j_2(1:ndim,n)
        !print '(12f7.2)', sum_j_z(1:ndim,n)
    end do

end program prob1
```

2.2 Companion Matrix

The zeros of a polynomial can be found using the eigenvalues of the corresponding companion matrix. Given a general monic polynomial

$$a(x) = a_0 + a_1 x + a_2 x^2 ... a_{n-1} x^{n-1} + x^n$$

The corresponding n by n companion matrix is formed by a sub-diagonal of ones and the negative of the coefficients forming the last column as follows

$$\begin{bmatrix} 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 & -a_{n-1} \end{bmatrix}$$

For the given polynomial

$$P_n(t) = (n+1) + (n)t + (n-1)t^2 + ... + 2t^{n-1} + t^n$$

the coefficients began at $n+1$ and decreased by one every term.

The Companion matrix was formed as follows

```
program prob2

    use numtype
    implicit none

    integer, parameter :: ndim = 20, lwork = 4*ndim, ldvr = 1, ldvl = 1      !dimensions +
    real(dp), dimension(ndim,ndim) :: comp_p
    real(dp), dimension(ndim) :: wr, wi
    real(dp) :: nn, nmax, work(lwork), vl(ldvl,ndim), vr(ldvr,ndim)
    integer :: n, info

    nmax = 21
    nn = 0

    comp_p = 0._dp

    do n = 1, ndim - 1

        comp_p(n+1,n) = 1._dp

    end do

    do n = 1, ndim

        comp_p(n,ndim) = -(nmax - nn)
```

```
        nn = nn + 1
        !print '(21f7.2)',comp_p(n,1:ndim)

    end do

    info = 0
    call dgeev ('n','n',ndim,comp_p,ndim,wr, wi, vl, ldvl, vr, ldvr, work,lwork,info)

    print '(21f7.2)', wr(1:ndim)
    print '(21f7.2)', wi(1:ndim)
end program prob2
```

The resulting eigenvalues of the companion matrix were then calculated using the dgeev routine within the LAPACK library.

3 Results

3.1 Quantum Matrices

The eigenvalues and eigen vectors for J_x

$$J_x \to -2.50 \begin{bmatrix} -0.18 \\ 0.40 \\ -0.56 \\ 0.56 \\ -0.40 \\ 0.18 \end{bmatrix} -1.50 \begin{bmatrix} 0.40 \\ -.53 \\ 0.25 \\ 0.25 \\ -0.53 \\ 0.40 \end{bmatrix} -0.50 \begin{bmatrix} -0.56 \\ 0.25 \\ 0.35 \\ -0.35 \\ -0.25 \\ 0.56 \end{bmatrix} 0.50 \begin{bmatrix} -0.56 \\ -0.25 \\ 0.35 \\ 0.35 \\ -0.25 \\ -0.56 \end{bmatrix} 1.5 \begin{bmatrix} -0.40 \\ -0.53 \\ -0.25 \\ 0.25 \\ 0.53 \\ 0.40 \end{bmatrix} 2.5 \begin{bmatrix} 0.18 \\ 0.40 \\ 0.56 \\ 0.56 \\ 0.40 \\ 0.18 \end{bmatrix}$$

$$J_y \to -2.50 \begin{bmatrix} -0.18i \\ 0.40 \\ 0.56i \\ -0.56 \\ -0.40i \\ 0.18 \end{bmatrix} -1.50 \begin{bmatrix} -0.40i \\ .53 \\ 0.25i \\ 0.25 \\ 0.53i \\ -0.40 \end{bmatrix} -0.50 \begin{bmatrix} -0.56i \\ 0.25 \\ -0.35i \\ 0.35 \\ -0.25i \\ 0.56 \end{bmatrix} 0.50 \begin{bmatrix} -0.56i \\ -0.25 \\ -0.35i \\ -0.35 \\ -0.25i \\ -0.56 \end{bmatrix} 1.5 \begin{bmatrix} -0.40i \\ -0.53 \\ 0.25i \\ -0.25 \\ 0.53i \\ 0.40 \end{bmatrix} 2.5 \begin{bmatrix} 0.18i \\ 0.40 \\ -0.56i \\ -0.56 \\ 0.40i \\ 0.18 \end{bmatrix}$$

$$J^2 \to 0.25 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} 0.25 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} 2.25 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} 2.25 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} 6.25 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} 6.25 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

For the $\delta = pi/10$ case no substantial correlation was found when comparing $e^{iJ_y\delta} * e^{iJ_x\delta} * e^{-iJ_y\delta} * e^{-iJ_x\delta}$ and $e^{iJ_z\delta^2}$. However both matrices went to zero for $\delta = pi/20$.

3.2 Companion Matrix

The eigenvalues of the companion matrix are the zeros of the polynomial. Each zero listed with a $\pm$ is a complex conjugate pair.

$$n = 20 \begin{bmatrix} 1.04 \pm 0.37i \\ 0.90 \pm 0.68i \\ 0.68 \pm 0.93i \\ 0.38 \pm 1.10i \\ 0.05 \pm 1.17i \\ -0.29 \pm 1.14i \\ -1.18 \pm 0.17i \\ -1.08 \pm 0.51i \\ -0.88 \pm 0.79i \\ -0.61 \pm 1.01i \end{bmatrix} n = 25 \begin{bmatrix} 1.04 \pm 0.30 \\ 0.96 \pm 0.55i \\ 0.81 \pm 0.77i \\ 0.61 \pm 0.95i \\ 0.37 \pm 1.08i \\ 0.11 \pm 1.14i \\ -0.17 \pm 1.14i \\ -0.43 \pm 1.07i \\ -1.16 \\ -1.13 \pm 0.27i \\ -1.03 \pm 0.53i \\ -0.88 \pm 0.76i \\ -0.67 \pm 0.94i \end{bmatrix} n = 30 \begin{bmatrix} 1.04 \pm 0.25 \\ 0.98 \pm 0.46i \\ 0.88 \pm 0.66i \\ 0.74 \pm 0.83i \\ 0.56 \pm 0.96i \\ 0.36 \pm 1.06i \\ 0.14 \pm 1.12i \\ -0.08 \pm 1.13i \\ -0.30 \pm 1.09i \\ -0.52 \pm 1.01i \\ -1.13 \pm 0.11i \\ -1.09 \pm 0.34i \\ -1.00 \pm 0.55i \\ -0.87 \pm 0.73i \\ -0.71 \pm 0.89i \end{bmatrix}$$