Quantum Information and Computing

2021 - 2022

Saverio Monaco

28/11/2021

Solution of the Time Independent Schrödinger Equation for an harmonic oscillator

Theory

$$\begin{cases} \hat{H} = \hat{p}^2 + \omega^2 \hat{x}^2 \quad \text{where } \hbar \equiv m \equiv 1 \\ \hat{H}\psi = E\psi \\ \hat{p} \to -i\hbar\partial/\partial x, \quad \hat{x} \to x \end{cases} \rightarrow \begin{cases} \left(-\frac{1}{2}\frac{\partial^2}{\partial x^2} + \frac{1}{2}\omega^2 x^2\right)\psi(x) = E_n \psi(x) \\ \psi_k'' = \frac{\psi_{k+1} - 2\psi_k + \psi_{k-1}}{dx^2} \end{cases}$$

$$-\frac{1}{2}\left[\frac{\psi_{k+1} - 2\psi_k + \psi_{k-1}}{dx^2}\right] + \frac{1}{2}\omega^2 x_k \psi_k = E\psi_k$$

$$H_{ij} = <\psi_i|H|\psi_j>$$

$$H = \frac{1}{2}\begin{pmatrix} \frac{2}{dx^2} + \omega^2 x_1^2 & -\frac{1}{dx^2} & 0 & \cdots & 0 \\ -\frac{1}{dx^2} & \frac{2}{dx^2} + \omega^2 x_2^2 & -\frac{1}{dx^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{2}{dx^2} + \omega^2 x_N^2 \end{pmatrix} \longleftrightarrow H\psi = E\psi$$

$$dx \equiv \frac{L}{N}$$

Code Development

To initialize the complex matrix to represent the hamiltonian of the system given L and N

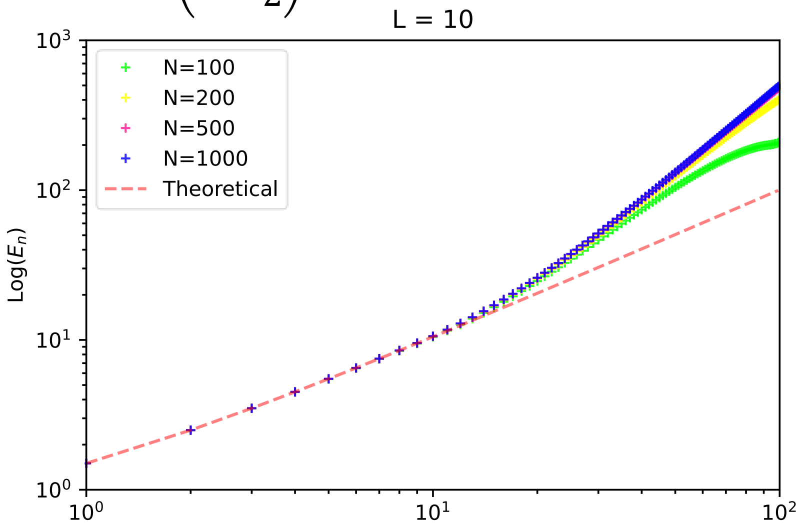
```fortran
function qho_H_init(L,N,omega) result(H)
    real                                      :: L,omega,dx
    integer                                   :: N, ii
    real*16, dimension(:,:), allocatable      :: elem_real
    type(cmatrix)                             :: H

    dx = L/N

    allocate(elem_real(N+1,N+1))
    elem_real = 0 * elem_real ! Initialize everything to 0

    ! diagonal elements
    do ii=1, N+1, 1
      elem_real(ii,ii) = ( 2 /(dx**2) ) + (omega**2)*((ii-1)*dx - L/2)**2
    end do

    ! tridiagonal elements
    do ii=2, N+1, 1
      elem_real(ii,ii-1) = - 1/(dx**2)
      elem_real(ii-1,ii) = - 1/(dx**2)
    end do

    elem_real = 0.5* elem_real ! Everything must be divided by 2

    H = cmatrix_init(cmplx(X=elem_real,KIND=8))
  end function qho_H_init
```

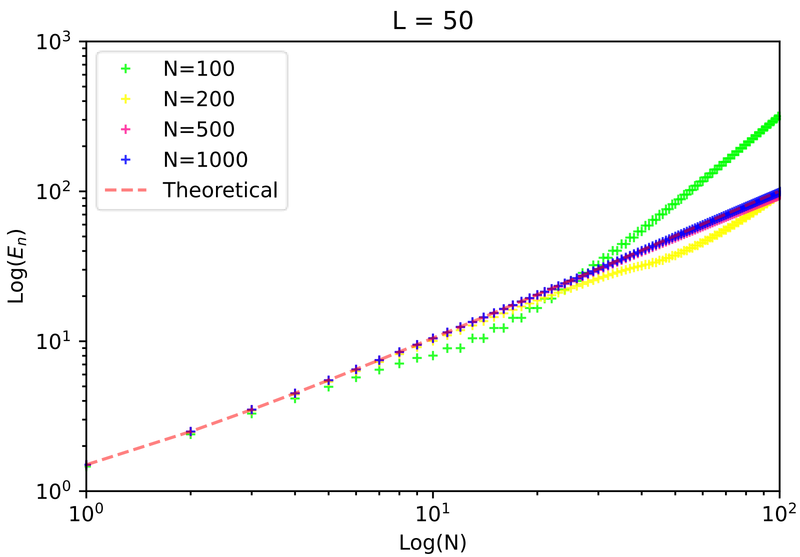
To compute eigenvalues and eigenvectors

```fortran
subroutine cmatrix_herm_eigens(cmat,eigenv,eigenh,success)
    type(cmatrix)                                        :: cmat
    real*8, dimension(:)                                 :: eigenv
    complex(kind=8), dimension(:,:)                      :: eigenh
    integer, optional                                    :: success

    ! LAPACK variables
    double precision, dimension(:), allocatable          :: RWORK
    integer                                              :: INFO, LWORK, N
    integer, parameter                                   :: LWMAX = 100000
    complex*16                                           :: WORK(LWMAX)
    complex(kind=8), dimension(:,:), allocatable         :: VR

    ! Check if matrix is squared
    if(cmat%dim(1) == cmat%dim(2)) then
      N = cmat%dim(1)

      allocate(RWORK(3*N-2))
      allocate(VR(N,N))

      ! Compute optimal size of workspace
      LWORK = -1
      eigenh = cmat%element

      call ZHEEV('Vectors', 'U', N, eigenh, N, eigenv, WORK,LWORK,RWORK,INFO
      LWORK = min(LWMAX, int(WORK(1)))

      ! Compute eigenvalues
      call ZHEEV('Vectors', 'U', N, eigenh, N, eigenv, WORK,LWORK,RWORK,INFO

      if(present(success)) then
        success = INFO
      end if
    end if
  end subroutine cmatrix_herm_eigens
```

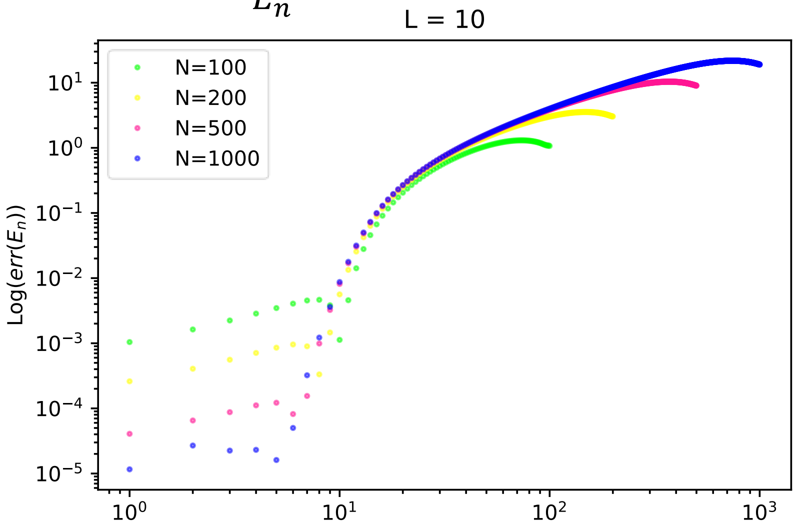
Results

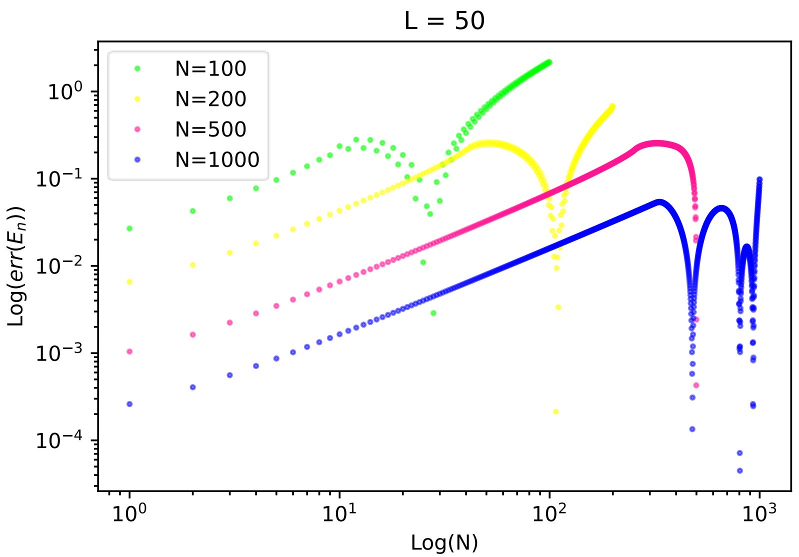
$$E_n^{true} = \omega\left(n + \frac{1}{2}\right)$$





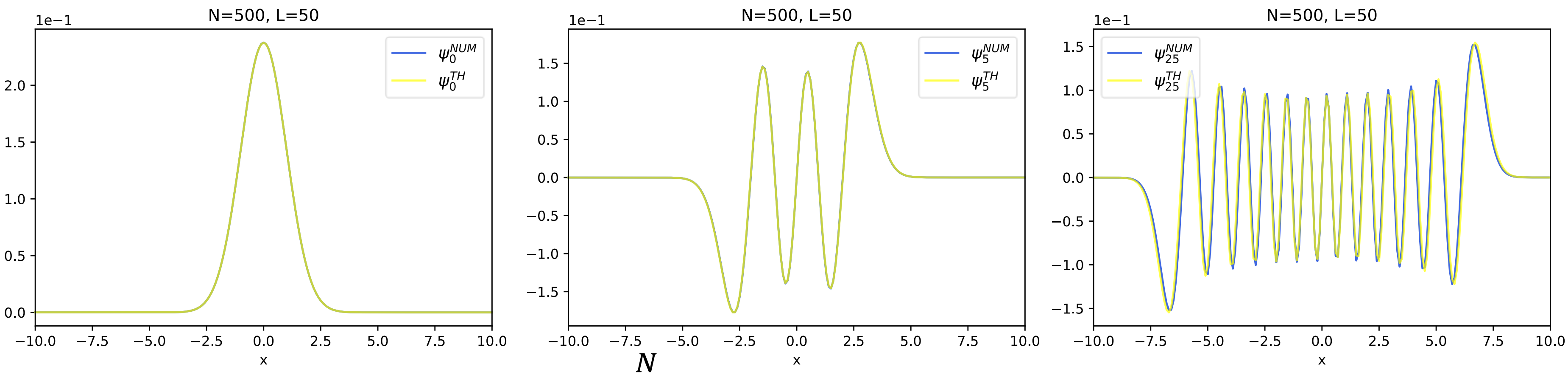
$$err(E_n) \equiv \frac{|E_n - E_n^{true}|}{E_n^{true}}$$



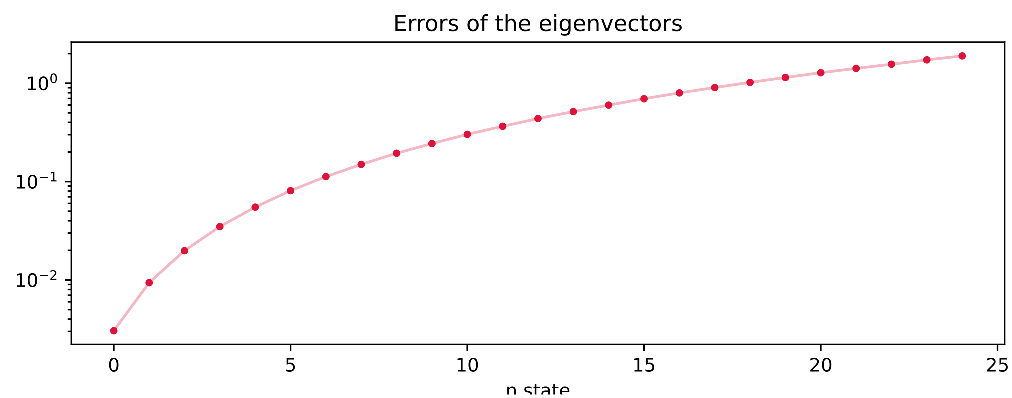


Results

$$\psi_n^{TH}(x) = \frac{1}{\sqrt{2^n n!}}\left(\frac{\omega}{\pi}\right)^{\frac{1}{4}} e^{-\frac{\omega x^2}{2}} H_n\left(\sqrt{\omega}\, x\right), \qquad n = 0, 1, 2, \ldots$$



$$err(\psi_n) = \sum_i^N \left|\psi_n^{TH}(x_i) - \psi_n^{NUM}(x_i)\right|$$



Self evaluation

Correctness: Hugely depends on the choice of the parameters. It can be still improved by using a better formula for the derivative

Stability: Various checks have been implemented to make the program as stable as possible

Accurate discretization: For the right parameters (for example L=50 and N=1000) the programs gives accurate results

Flexibility: Based on ZHEEV and not DSTEMR for generalizing to other problems

Efficiency: Can be improved by considering DSTEMR and real-only matrices

800 ANNI 1222 • 2022

UNIVERSITÀ DEGLI STUDI DI PADOVA

Thanks for the attention