Presentation of Assignment 4

Quantum Information and Computing, A.Y. 2022/2023

Paolo Zinesi
29/11/2022

Theoretical introduction

The numerical simulation is implemented considering the dimensionless time-independent Hamiltonian $\hat{H}$, obtained from the 1D harmonic oscillator Hamiltonian $\hat{\hat{H}}$

$$\hat{\hat{H}} = \frac{\hat{\hat{p}}^2}{2m} + \frac{m}{2}\tilde{\omega}^2\hat{\hat{q}}^2 \xrightarrow[\tilde{\omega}=2\omega]{\hbar=1,\, 2m=1} \hat{H} = \hat{p}^2 + \omega^2\hat{q}^2$$

Eigenvalues $$E_k = 2\omega\left(k + \frac{1}{2}\right)$$

Eigenfunctions (q-representation) $$\Phi_k(q) = \frac{1}{\sqrt{2^k k!}}\left(\frac{\omega}{\pi}\right)^{1/4}\exp\left(-\frac{\omega}{2}q^2\right)\mathcal{H}_k(\sqrt{\omega}\,q)$$

where $\mathcal{H}_k(\cdot)$ denotes the Hermite polynomial of order k

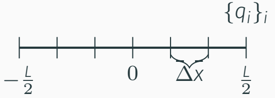
Eigensystem $$\hat{H}\,\Phi_k(q) = \frac{\partial^2}{\partial q^2}\Phi_k(q) + \omega^2 q^2\Phi_k(q) = E_k\Phi_k(q)$$

Discretized eigensystem $$M\Psi_k = E_k\Psi_k$$

with $E_k \in \mathbb{R}$, $\Psi_k \in \mathbb{R}^N$, $M \in \mathbb{R}^{N\times N}$

Parameters:

- N: evaluation points (excluded $\pm\frac{L}{2}$)
- K_{max}: number of first eigenvalues to compute
- ω: frequency of the oscillator



$$\Delta x = \frac{L}{N+1}, \quad \frac{L}{2} = 2\sqrt{\frac{K_{max}}{\omega}}$$

$$\underbrace{\begin{pmatrix} d_1 & a & & & \\ a & d_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & d_{N-1} & a \\ & & & a & d_N \end{pmatrix}}_{M} \underbrace{\begin{pmatrix} \psi_1 \\ \vdots \\ \vdots \\ \vdots \\ \psi_N \end{pmatrix}}_{\Psi}$$

$$d_i = \frac{2}{\Delta x^2} + \omega^2 q_i^2$$
$$a = -\frac{1}{\Delta x^2}$$

Important assumption: $\psi_0 = \psi_{N+1} = 0 \implies \Psi(\pm L/2) = 0$

Code development

LAPACK subroutine **DSTEVX** is specialized to compute the first K_{max} eigenvalues and eigenvectors of the **symmetric tridiagonal matrix** $\mathbf{M}$.

Preconditions on $N, K_{max}, \omega > 0$ and $K_{max} \leq N$ are checked beforehand.

HarmOsc_1D.f90

```
! fill arrays that define tridiagonal matrix
xgrid(:) = (/ (-0.5D0*Ltot + ii*deltax, ii=1,Ntot) /)
diag(:) = (/ (2.D0/(deltax**2) + (omega*(-0.5D0*Ltot + ii*deltax))**2, ii=1,Ntot) /)
upper_diag(:) = -1.D0/(deltax**2)

! ...

CALL DSTEVX('V', 'I', Ntot, diag, upper_diag, 0.D0, 1.D0, 1, k_max, &
            2*DLAMCH('S'), eig_num, eigenvals, eigenvects, Ntot, &
            work, iwork, ifail, info)

! function normalization with Simpson's rule
DO ii = 1, k_max
    norm2 = (SUM(eigenvects(1:Ntot:2, ii)**2)*4.D0 + &
             SUM(eigenvects(2:Ntot:2, ii)**2)*2.D0) * deltax/3.D0
    eigenvects(:,ii) = eigenvects(:,ii) / SQRT(norm2)
END DO

! write results on file ...
```

Results

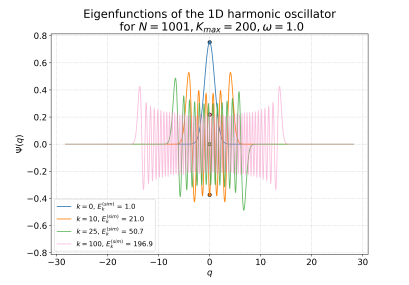
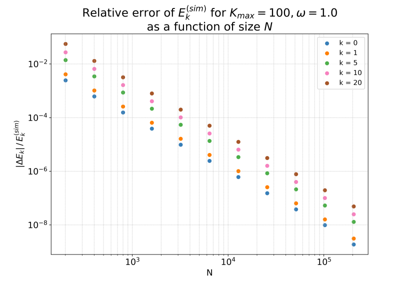


Figure 1: Eigenfunctions

1. **Correctness**: Figure 1 presents some of the computed eigenfunctions. Functions are symmetric or anti-symmetric according to the sign of $(-1)^k$ as expected.
2. **Stability**: When fixing K_{max} and ω, the relative errors of eigenvalues decrease for increasing N, according to Figure 2.



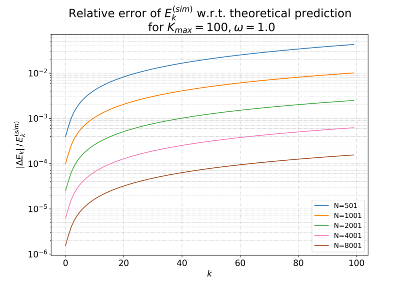


Figure 2: Eigenvalue errors scalings with N and k

Results

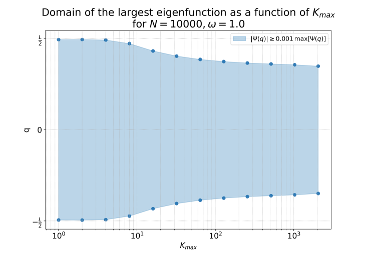


Figure 3: Discretization

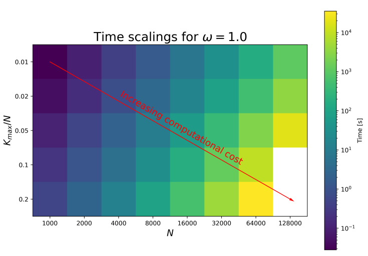


Figure 4: Efficiency

3. **Accurate discretization**: The choice of $\frac{L}{2} = 2\sqrt{\frac{K_{max}}{\omega}}$ is fundamental to obtain a good numerical estimate of the eigenfunction. Figure 3 demonstrates the validity of the assumption $\Psi(\pm L/2) = 0$.
4. **Flexibility**: The results are saved in an output file that can be easily imported into another code.
5. **Efficiency**: The subroutine **DSTEVX** uses minimal information to solve the eigenproblem. In particular, only two N-dimensional vectors are used to fully specify the symmetric tridiagonal matrix and it is possible to go beyond $N > 10^5$ if K_{max} is not too large. The time scaling of the program is shown in Figure 4.