

# Evaluation of Eigenvalues of Infinite Matrices

## Computational Physics - Phys 562

Benjamin Deutsch  
Department of Physics  
California State University Long Beach

April 18, 2017

### Abstract

No abstract

## 1 Introduction

Much meaning about a system can be gain if the eigenvalues can be calculated, while not normally a difficult and a skill taught in undergraduate math or physics classes. The issue arises that in quantum mechanical case it is not uncommon to encounter hamiltonians which rely on extremely large and in some cases, infinite matrices. To facilitate this problem, we required a computer to do these tedious calculations. It should be noted that while a computer does not function on the order to infinities, we can truncate the matrices, in relatively useful size dimensions according to the problem at hand. Once this has been done the programmer can simply use a language in order to carry out the operations, needed to manipulate the large matrices. Here we are asked to do just this against resultant hamiltonian of a harmonic oscillator;

$$\hat{H} = \frac{1}{2m}\hat{P}^2 + \frac{1}{2}m\omega^2\hat{X}^2, \quad (1)$$

The eigenvalues of the hamiltonian describe different discrete energy states of the system and are usually presented as,

$$\hat{H}\Psi = E_n\Psi; \quad E_n = (n + \frac{1}{2})\hbar\omega \quad (2)$$

Normalized units will be given for  $\hbar$  and  $m$  for ease, and will vary  $\omega$  at .5, 1, 2, and 3. Truncations of the infinite matrices  $\hat{X}^2$  and  $\hat{P}^2$  are set at  $N \times N = 10, 20, 50, 100$ , and 1000.

## 2 The Math and Theory

The hamiltonian of a harmonic oscillator shown here,

$$\hat{H} = \frac{1}{2m}\hat{P}^2 + \frac{1}{2}m\omega^2\hat{X}^2, \quad (3)$$

Contains operators  $\hat{X}^2$  and  $\hat{P}^2$  which are understood as infinite matrices

$$\hat{X} = \sqrt{\frac{1}{2}} \begin{bmatrix} 0 & \sqrt{1} & 0 & 0 & \dots \\ \sqrt{1} & 0 & \sqrt{2} & 0 & \dots \\ 0 & \sqrt{2} & 0 & \sqrt{3} & \dots \\ 0 & 0 & \sqrt{3} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \hat{P} = i\sqrt{\frac{1}{2}} \begin{bmatrix} 0 & -\sqrt{1} & 0 & 0 & \dots \\ \sqrt{1} & 0 & -\sqrt{2} & 0 & \dots \\ 0 & \sqrt{2} & 0 & -\sqrt{3} & \dots \\ 0 & 0 & \sqrt{3} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (4)$$

In order for the computer to use this infinite matrix, the programmer must set an upper limit of dimension, here we have chosen a limit of 1000 ( $N \times N$ ). With simplification steps taken, and constants set to one, we arrive at;

$$\hat{H} = \left(\frac{1}{4}\right) - \hat{P}^2 + \omega^2\hat{X}^2 \quad (5)$$

Aside notice the complex part of the momentum matrix,  $\hat{X}^2$  will disappeared since taking the square. The hamiltonian can now successfully be computed. The eigenvalues of the  $\hat{H}$  can be found explicitly here making use of;

$$\hat{H}\Psi = En\Psi; \quad En = \left(n + \frac{1}{2}\right)\hbar\omega \quad (6)$$

With substitution of a individual  $\omega$ , and  $n$  taken between 0...10, we now will have the first 10 real eigenvalues of the hamiltonian.

### 3 The code

For all Fortran code we must begin with the declaration of variables, all terms used here in this program are define in the heading, as real, or integer. The only exception being the real dimensional term `ndim`, which will be used to create an arrays of `i` and `j` for entries of the matrices. This is important in that we then applied conditions of `if then` statements to produce the given  $\hat{X}^2$  and  $\hat{P}^2$  matrices with the continual off diagonal entries. Printing in order to check the correct array was done here.

Next, in order to square the above operators we used the `matmal` function an intrinsic operation to fortran. Multiplying the operators by themselves produced a squared operator.

We then used the simplified equation (5) to define the hamiltonian, here we also identify one of the given  $\omega^2$  values. Note that this must be edited manually for each of the omega values. Squaring the  $\hat{H}$  is done again with the `matmal` function. It should be noted here that we have placed a `tt ndim-1` restriction on the arrays, this is done to remove the well known truncation error of eigenvalues.

Finally, we can compute the eigenvalues, in order to do this we have employed a neat trick, utilizing the LAPACK inclusion of fortran we can call upon `tt dsyev` a subroutine in fortran used to compute eigenvalues. All that needs to be done is an understanding and simple edit of in and out commands following the `tt call dsyev` function. (Fortunately like a washing machine). We printed values to check for correction.

Furthermore to investigate the behavior of these eigenvalues at high iteration of `tt ndim` (matix size), we cross checked this against the explicit calculation of eigenvalues for each  $\omega^2$  (6). What was found was written to external files for graphing.

### 4 Results and Conclusion

To show all eigenvalues for each  $\omega^2$  term in the 1000x1000 matrix, would be very meaningless, however we can offer the first 11 eigenvalues (`N, 0..10`) for each as shown below;

$\omega^2 = .5$	$\omega^2 = 1$	$\omega^2 = 2$	$\omega^2 = 3$
0.35	.5	0.71	0.87
1.06	1.5	2.12	2.60
1.77	2.5	3.54	4.33
2.48	3.5	4.95	6.07
3.18	4.5	6.36	7.82
3.90	5.5	7.80	9.77
4.62	6.5	9.23	11.65
5.47	7.5	10.95	14.57
6.24	8.5	12.48	16.76
7.72	9.5	15.43	22.00
8.59	10.5	17.18	24.57

While this restrained graph does not commute much to us , we find an interesting pattern emerges when we plot the all `ndim=1000` explicitly calculated eigenvalues and the intrinsic `tt dsyev` function on the same graph.

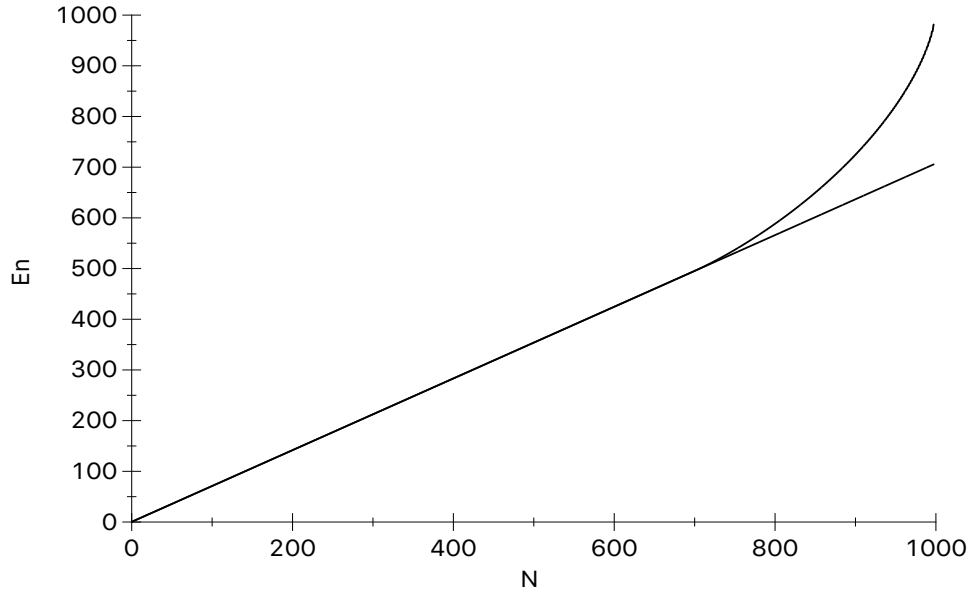


Figure 1:  $\omega^2 = 0.5$ . Explicit calculation is a straight line; Matrix interpretation curves upward

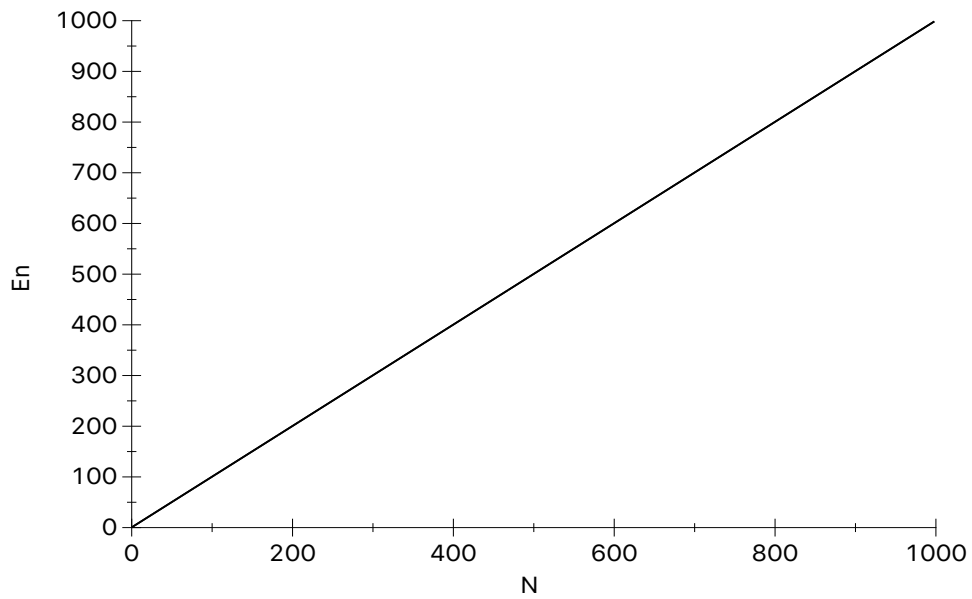


Figure 2: For  $\omega^2 = 1$ . Two lines lie on top of each other.

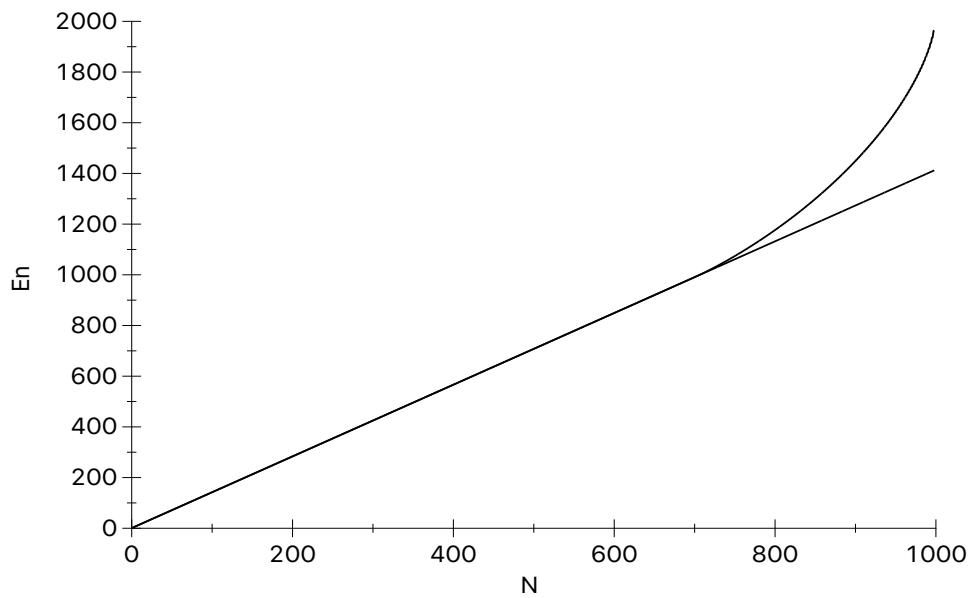


Figure 3: For  $\omega^2 = 2$ . Explicit calculation is a straight line; Matrix interpretation curves upward.

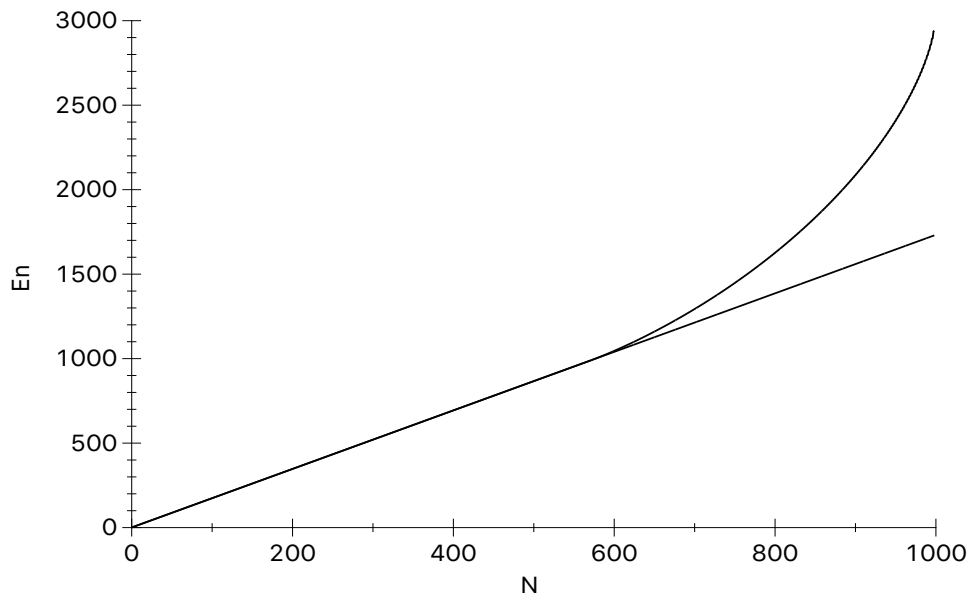


Figure 4: For  $\omega^2 = 3$ . Explicit calculation is a straight line; Matrix interpretation curves upward.

After analysis, the orientation of the lines is a telling sign of the form of matrix (eigenvalues) that each  $\omega^2$  iterates. We see that for  $\omega^2 = 1$  the real eigenvalues of both calculations are consist, however this is not the case for the other  $\omega^2$  values. In printing out a smaller matrix of these eigenvalues is clear as to why, while the  $\omega^2$  for both explicit and matrix methods align themselves to the diagonals and as a result match numerically. The entries for  $\omega^2 = .5, 2$ , and  $3$  do not, and as the dimension of the matrices increase in size the error of the matrix interpretation (intrinsic function) grows as well. The eigenvalues behavior interestingly occurs around the same place in all 3 cases, becoming divergent at around  $N=600...800$ .

## References

- [1] Z. Papp and A. Bill, *Computational Physics Lecture Notes*, California State University Long Beach.