4. Lectures 7,8

Plan:

- ◇ Least-squares approximation
- ◇ Moore-Penrose pseudo-inverse
- ◇ Examples of using SVD to approximate data
- ◇ On Principal Component Analysis (PCA), Proper Orthogonal Decomposition (POD), and model reduction
- ◇ A note on statistics and data correlations
- ◇ More general least squares, nonlinear problem
- ◇ Regularized least squares
- ◇ Weighted least squares
- ◇ A note on recursive least squares
- ◇ A note on the numerical solution of Shrödingers' equation

4.1. **Least-squares approximation and pseudo-inverse.** ..... Recall the example from the first lecture. We have data of the form (x_i, y_i) that we expect to depend on each other. The goal is to find a simple representation for this dependence that is good in some measure of "goodness". There are different options for this measure, but the main is based on a squared error between the data and their approximation – the least-squares method.

Let us begin with the simple choice of the approximation – a straight line, $y = ax + d$. The least-square error between the data and this function is

$$E = \sum_{i=1}^{m} (y_i - ax_i - d)^2 .$$

We choose a and d such that E is minimal. By calculus, this is solved by letting

$$\frac{\partial E}{\partial a} = -2 \sum_{i=1}^{m} x_i (y_i - ax_i - d) = 0$$
$$\frac{\partial E}{\partial d} = -2 \sum_{i=1}^{m} (y_i - ax_i - d) = 0,$$

which can be rearranged as

$$\left(\sum x_i^2\right) a + \left(\sum x_i\right) d = \sum x_i y_i$$
$$\left(\sum x_i\right) a + md = \sum y_i$$

or

$$\begin{bmatrix} \left(\sum x_i^2\right) & \left(\sum x_i\right) \\ \left(\sum x_i\right) & m \end{bmatrix} \begin{bmatrix} a \\ d \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}.$$

This system can be solved and will give a unique solution provided that

$$m \left(\sum x_i^2\right) - \left(\sum x_i\right)^2 \neq 0,$$

which is true unless all x_i are the same.

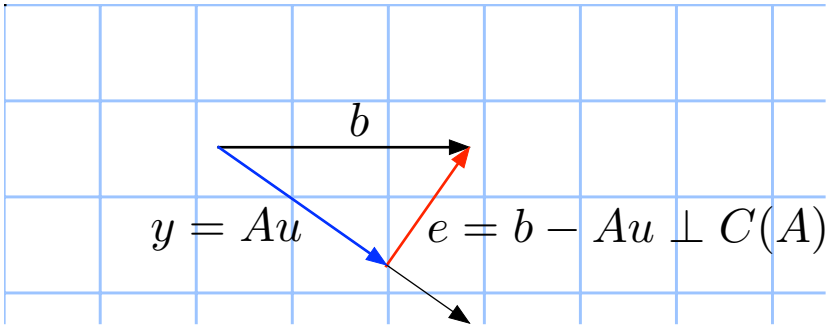

FIGURE 4.1. For any b, we want to project it onto the range of A. If b is not in the range of A, then there will be some error $e = b - Ax$. The error is minimal if the projection is orthogonal.

Instead of continuing in this fashion, we will turn to the geometric solution of this problem. Define matrix A and a vector b:

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_m & 1 \end{bmatrix}, \quad b = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix}.$$

The equation

$$Au = b$$

where $u = [a\ d]^T$ clearly has no solution unless b happens to be exactly in the range of A, an exceptional case. Now, we ask the question: What is the vector Au in the range of A that is closest to b? Meaning, closest in the 2-norm. From what we learned about projections, this vector is clearly the projection of b onto the column space of A. That is, $Au = Pb$ where $P = A\left(A^TA\right)^{-1}A^T$ is the projector. Therefore, $u = \left(A^TA\right)^{-1}A^Tb$ is the desired solution.

Another way to see the same (and basically repeating the discussion above) is to note that if Au is the projection of b, then the error $e = b - Au$ is its orthogonal complement, i.e. $e = b - Au$ is in the nullspace of A^T. Since the column space of A is orthogonal to the nullspace of A^T,we obtain

$$\begin{aligned} A^T(b - Au) &= 0 \\ A^TAu &= A^Tb \qquad \Leftarrow \text{ normal equation.} \end{aligned}$$

We obtain the same solution as before

$$u = \left(A^TA\right)^{-1}A^Tb.$$

That the A^TA has an inverse follows from linear independence of columns of A.

This matrix $\left(A^TA\right)^{-1}A^T$ is called *Moore-Penrose pseudo-inverse*, denoted by A^+. Thus, a general system of equations

$$Ax = b$$

where A has independent columns is solved by

$$x = A^+b,$$

and A^+ is just the regular inverse when A is square and invertible, $A^+ = A^{-1}A^{-T}A^T = A^{-1}$. When A is tall with independent columns then $x = A^+b$ gives the least squares solution to $Ax = b$.

In the latter case, using full SVD, we can write that

$$A^+ = \left(A^TA\right)^{-1}A^T = \left(V\Sigma^2V^T\right)^{-1}V\Sigma U^T = V\Sigma^+U^T$$

where Σ^+ contains $1/\sigma_i$ on its diagonal if $\sigma_i \neq 0$ and 0 in places where the diagonal element of Σ is zero. This is because the SVD can be written in economy form, wherein Σ contains no zeros on the diagonal. Hence the inversion of $V\Sigma^2V^T$ involves only inverses of nonzero σ_i.

Thus, the least-squares problem is solved by

$$x = V\Sigma^+U^Tb.$$

Example 15. Given the measurements

x_i	0	1	2	3	4
y_i	1	1	2	2	3

, find the best linear fit to the data. Determine the Moore-Penrose pseudo-inverse and the error of the approximation.

Solution. We put the data into matrix A and vector b:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 3 \end{bmatrix}$$

and form a normal system $A^TAu = A^Tb$:

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix} = 5 \begin{bmatrix} 6 & 2 \\ 2 & 1 \end{bmatrix} u = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 23 \\ 9 \end{bmatrix}$$

$$u = \frac{1}{10} \begin{bmatrix} 1 & -2 \\ -2 & 6 \end{bmatrix} \begin{bmatrix} 23 \\ 9 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 4/5 \end{bmatrix}.$$

So the best line is $y = \frac{1}{2}x + \frac{4}{5}$.

The Moore-Penrose pseudo-inverse is

$$A^+ = \left(A^TA\right)^{-1}A^T = \begin{bmatrix} 1 & -2 \\ -2 & 6 \end{bmatrix}.$$

The approximation error is found as

$$e = b - Au = \frac{1}{10} \begin{bmatrix} 2 \\ -3 \\ 2 \\ -3 \\ 2 \end{bmatrix}, \quad \|e\|_2 = 0.5477.$$

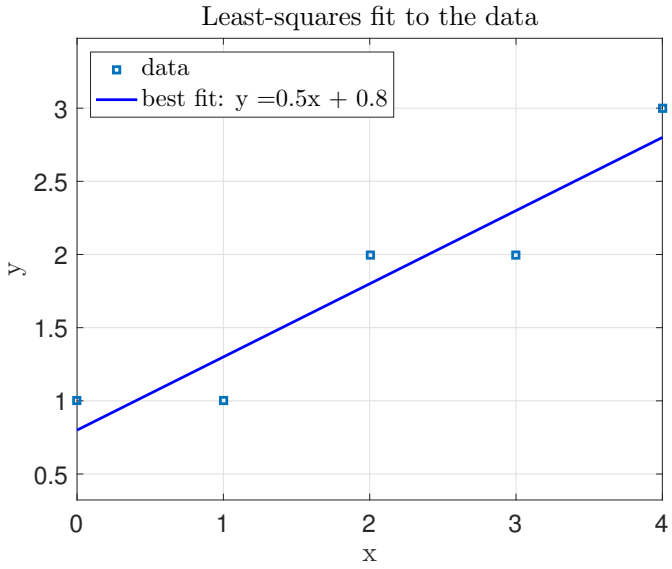

FIGURE 4.2. The data and the best fit for the example 15.

```
%% Example of least square calculation
clear all   % remove all variables from the workspace
clc         % clear command window

x = [0 1 2 3 4]' %data
y = [1 1 2 2 3]'
A = [x ones(size(x))]  % put data in a matrix
B = A'*A               %normal matrix
Aty = A'*y             %right-hand side
u = B\Aty              %solve the system to find the fit coefficients
e = y - A*u            %error
e2norm = norm(e,2)

% plot the fit on a finer grid
xfit = linspace(x(1),x(end),100);
yfit = u(1)*xfit + u(2);
plot(x,y,'s',xfit,yfit,'b-','LineWidth',2)
grid on; axis equal

% cell array of text options
txtoptions = {'Interpreter','latex','FontSize',18};
xlabel ('x',txtoptions{:}); ylabel ('y',txtoptions{:})

% add legendsand title
fiteqn = ['best fit: y =',num2str(u(1)),'x + ',num2str(u(2))];
legends = {'data', fiteqn};
legend(legends,txtoptions{:},'Location','NorthWest');
```

```
title('Least-squares fit to the data',txtoptions{:})

ax = gca;   %get current axis
ax.FontSize = 18; % change the font size of the axes labels
```

4.2. **Least squares with penalty. ..** If A does not have independent columns, then $Ax = 0$ has a nullspace and $A^T Ax = A^T b$ does not have a solution. We can still use the pseudo-inverse A^+ to find the minimal-norm solution, $x = A^+ b$. This is done using regularization. Instead of minimizing $\|Ax - b\|^2$, we minimize $\|Ax - b\|^2 + \delta^2 \|x\|^2$ with the penalty term δ. Then the normal equation is $\left(A^T A + \delta^2 I\right) x = A^T b$, which has a solution. This is called "ridge regression".

The pseudo-inverse A^+ is the limit of $\left(A^T A + \delta^2 I\right)^{-1} A^T$ as $\delta \to 0$.

Indeed, if $A = \sigma$ is just a number, we get

$$A^+ = \lim_{\delta \to 0} \frac{\sigma}{\sigma^2 + \delta^2} = \begin{cases} 0, & \sigma = 0 \\ \frac{1}{\sigma}, & \sigma \neq 0 \end{cases}.$$

So, if you run into division by 0, just keep 0.

For diagonal $A = \Sigma$,

$$\left(\Sigma^T \Sigma + \delta^2 I\right)^{-1} \Sigma^T = \text{diag}\{\frac{\sigma_i}{\sigma_i^2 + \delta^2}, \quad \text{if } \sigma_i \neq 0, \text{ otherwise } 0.\}$$

tends to Σ^+. Again, the inversion affects only non-zero terms on the diagonal of Σ.

This remains true in general. Let $A = U \Sigma V^T$. Then

$$A^T A + \delta^2 I = V \left(\Sigma^T \Sigma + \delta^2 I\right) V^T$$

and

$$\left(A^T A + \delta^2 I\right)^{-1} A^T = V \left(\Sigma^T \Sigma + \delta^2 I\right)^{-1} V^T V \Sigma^T U^T = V \left[\left(\Sigma^T \Sigma + \delta^2 I\right)^{-1} \Sigma^T\right] U^T.$$

As $\delta \to 0$, the term in [] tends to Σ^+, as above, and hence in the limit, we obtain the pseudo-inverse

$$A^+ = V \Sigma^+ U^T.$$

For example, if $A = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$, then $A^+ = \begin{bmatrix} 1/2 & 0 \\ 0 & 0 \end{bmatrix}$.

Thus, $x = A^+ b$ gives the least squares solution for any A, only when A has dependent columns this solution is of minimal norm out of all possible. Indeed, even though the error vector $e = b - A\left(x^+ + x_0\right)$ is unaffected by the nullspace, $Ax_0 = 0$, the length of the solution $\|x^+ + x_0\|^2$ is affected – it becomes $\|x^+\|^2 + \|x_0\|^2$, and the minimal length is given by $x_0 = 0$, i.e. if we take nothing from the nullspace of A. Note $x^+ \perp x_0$ as x^+ is in $R(A)$.

4.3. **Examples of SVD analysis of data. Basic idea of PCA. .....** Generally, given a function f, one can approximate it via an expansion in terms of various basis functions, such as x^i (Taylor series), e^{ix} (Fourier series), $\phi_i(x)$ (eigenfunction expansion), etc. The expansion is of the form

$$f = \sum_{i=1}^{N} a_i \phi_i$$

and with ϕ_i orthonormal, i.e. when

$$\int \phi_i(x)\,\phi_j(x)\,dx = \delta_{ij},$$

then the coefficients are found via

$$a_i = \int f(x)\,\phi_i(x)\,dx.$$

The accuracy of such an approximation will require a different number of terms in each case. Some basis functions are better than others in this respect. If we want the fewest number of terms in the expansion, then we are dealing with Proper Orthogonal Modes - POD, which are the best, or most natural, for the given function.

The following two examples illustrate the concept of PCA, the Principal Component Analysis, whereby a signal is represented by its dominant modes, which will be identified by the analysis of the signal via SVD. For simplicity, the data are generated by an explicit function. However, it is clear that they could have come from anywhere – experiment or numerical simulation of a complex system, such as a fluid flow or wave in a quantum system. In either case, we have the data in a matrix and the goal is to understand if the data have some simplicity in them, meaning if the entire set of data can be represented by a few simple basis functions (POD modes). This is the idea of low-order approximation, or model reduction which is aimed at identifying what is redundant and throwing that out.

Example 16. Approximate the surface given by the function

$$f(x,y) = e^{-|(x-0.5)(y-1)|} + \sin(xy), \quad x \in [0,1], y \in [0,2].$$

This example is illustrated in the Matlab code *PODexample1.m.*

Example 17. Approximate the surface given by the function

$$f(x,t) = (1 - 0.5\cos(2t))\,\mathrm{sech}(x) + (1 - 0.5\sin(2t))\,\mathrm{sech}(x)\tanh(x)$$

on $x \in [-10,10], t \in [0,10]$.

This example is illustrated in the Matlab code *PODexample2.m.*

In these examples, we follow the procedure of identifying principal components of f via SVD. The domain in (x,t) space is discretized by a finite number of points in each direction, and matrix $f_{ij} = f(x_i,t_j)$ is SVD-ed as

$$F = U\Sigma V^T = \sum \sigma_i u_i v_i^T.$$

Importantly, the columns of U are identified as spatial modes while components of V contain the time evolution of the modes. The singular values σ_i measure how much of the mode i contributes to the "energy" of the signal given by f. A good measure of the "energy" is the Frobenius norm of F: $\|F\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + ... + \sigma_n^2}$. Or one can take $\sum \sigma_i$. Then the relative energy of the k-term expansion is

$$E_k = \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{n} \sigma_i}$$

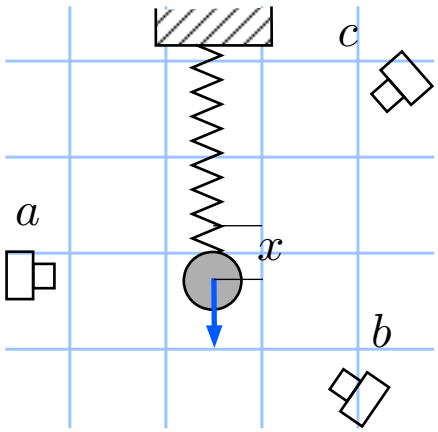

FIGURE 4.3. A spring oscillates vertically. Three cameras record its position not knowing that oscillations are actually vertical and 1D.

4.4. **A note on statistics and data correlations ...** Consider the example of a mass on a spring that is oscillating vertically in gravity.

We take three cameras (a,b,c) at different locations and measure the position of the mass at different times. The camera i records the coordinates (x_i, y_i), where x_i and y_i are vectors of length equal to the number of time measurements, say n. We put these data into a matrix

$$X = \begin{bmatrix} x_a \\ y_a \\ x_b \\ y_b \\ x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x_{a1} & x_{a2} & \dots & x_{an} \\ y_{a1} & y_{a2} & \dots & y_{an} \\ x_{b1} & x_{b2} & \dots & x_{bn} \\ y_{b1} & y_{b2} & \dots & y_{bn} \\ x_{c1} & x_{c2} & \dots & x_{cn} \\ y_{c1} & y_{c2} & \dots & y_{cn} \end{bmatrix}$$

which will have dimensions $6 \times n$, with n typically very large.

Next, there are two issues to address:

- ◇ Noise. Any measurements will have some noise associated with it due to various factors.
- ◇ Redundancy. Not all the data are necessary due to possible correlations between different measurements. Clearly, in the given example, the motion is 1D, so different cameras record essentially 1D motion, but under different angles. In reality, it would be enough to use a single camera. The problem is, in general, we do not know how many are needed and where to place them. We have to figure this out after the measurements.

How to identify redundancy? The basic idea is to look at correlations between different variables that have been measured. If some are correlated, then there is redundancy.

For example, suppose $a = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}^T$ and $b = \begin{bmatrix} b_1 & b_2 & \dots & b_n \end{bmatrix}^T$ are two vectors of measured data. Assume they have mean zero (otherwise, subtract the mean). Then $a^T a$ measures variance of a, i.e. the extent to which it varies from the mean 0. Similarly for b. This is usually normalized with $n-1$ and denoted as

$$\sigma_a^2 = \frac{1}{n-1} a^T a, \quad \sigma_b^2 = \frac{1}{n-1} b^T b.$$

The covariance between these data is defined as

$$\sigma_{ab} = \frac{1}{n-1} a^T b,$$

and it measures how much the data are correlated, or aligned with each other. If $a^T b = 0$, then the data are completely uncorrelated, if $b = a$ – completely correlated.

With many vectors of data, we form a covariance matrix

$$C_X = \frac{1}{n-1} X X^T$$

that will contain both variances and covariances between different data vectors. It is in our case

$$C_X = \frac{1}{n-1} \begin{bmatrix} \sigma^2_{x_a x_a} & \sigma_{x_a y_a} & \sigma_{x_a x_b} \dots \\ \sigma_{y_a x_a} & \sigma^2_{y_a y_a} & \sigma_{y_a x_b} \dots \\ \sigma_{x_b x_a} & \sigma_{x_b y_a} & \sigma^2_{x_b x_b} \dots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

and note that this is a symmetric 6×6 matrix. Its off-diagonal terms are covariances between different measurements. If they are large, then there is much redundancy. The diagonal terms tell how strong the variance is in a particular measurement. If it is large, then the dynamics in that direction is strong.

The next step is to identify the basis in which the matrix C_Y is diagonal. It is the basis of e-vectors of C_X, and they are also singular vectors of X.

Letting $X = U\Sigma V^T$, we get

$$C_X = \frac{1}{n-1} U\Sigma V^T V\Sigma U^T = \frac{1}{n-1} U\Sigma^2 U^T.$$

Therefore

$$\Sigma^2 = (n-1)\, U^T C_X U = U^T X X^T U = \left(U^T X\right)\left(U^T X\right)^T = Y Y^T = C_Y$$

is the new covariance matrix which is diagonal in the frame defined by $Y = U^T X$. That is, the measurements X were done in, say, a standard basis, and there is a lot of redundancy in that basis. If we rotate the frame to the basis made of rows of U, then in that frame the measurements become Y, and then the redundancy is gone. In this new frame, there is no covariance, but only variance. And the largest variance is given by the largest singular value of C_X, σ_1^2, that identifies the dominant dynamics that takes place along the first singular vector u_1. For the particular case of a mass on a spring oscillating up and down, there will be only one nonzero singular value, since it is a single degree of freedom system.

4.5. **More general least-squares. .....** Suppose we want to fit with a quadratic function $y = ax^2 + cx + d$. Then the wish equations $ax_i^2 + cx_i + d = y_i$ can be written in a matrix form $Au = b$ with b again containing y_i, $u = [a, c, d]^T$ and matrix A becoming bigger

$$A = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ & \vdots & \\ \\ x_n^2 & x_n & 1 \end{bmatrix}.$$

Still, this is a linear system that is solved via the normal equation $A^T A u = A^T b$ to get the least squares solution. One could replace x and x^2 with any functions of x, e.g. e^x, $\sin x$, etc. This is still a linear problem, because a, c, d enter linearly.

On the other hand, if one wishes to fit the data with a function like $y = ae^{cx}$, then the problem is no longer linear. We need to solve the system

$$y_i = ae^{cx_i}, \quad i = 1, 2, 3, ..., n$$

in some least-squares sense. One can proceed by minimizing the error $E = \sum (y_i - ae^{cx_i})^2$ via derivatives to get

$$\frac{\partial E}{\partial a} = -2 \sum (y_i - ae^{cx_i}) e^{cx_i} = 0$$
$$\frac{\partial E}{\partial c} = -2 \sum (y_i - ae^{cx_i}) a x_i e^{cx_i} = 0$$

from which we get

$$a \sum e^{2cx_i} - \sum y_i e^{cx_i} = f_1(a, c) = 0$$
$$a \sum x_i e^{2cx_i} - \sum x_i y_i e^{cx_i} = f_2(a, c) = 0,$$

a system of two nonlinear equations for a and c. Generally, this may be a large system of nonlinear equations

$$\mathbf{F}(\mathbf{a}) = 0$$

needs to be solved by some nonlinear solver, such as the Newton's method or some others that we will discuss later in the course. Note, that in the particular case at hand, the problem can be reduced to linear by changing the variables as $\ln y_i = \ln a + c \ln x_i$, which is now a linear relationship between $\ln y_i$ and $\ln x_i$ for the unknowns $\ln a$ and c.

4.6. **More on regularized least squares** ..... The standard or weighted least-squares minimization problems (here, in the second problem different equations may have different weights represented by the matrix C)

$$\text{minimize } \|Ax - b\|^2 \text{ by solving } A^T A\hat{x} = A^T b$$
$$\text{minimize } (b - Ax)^T C (b - Ax) \text{ by solving } A^T CA\hat{x} = A^T Cb$$

do not always have solutions or the solutions may be non-unique.

For example, if A is wide, having more columns than rows, then the problem is underdetermined, meaning that there will be infinitely many solutions: $Ax = b$ with wide A has $x = x_p + x_0$ with $Ax_0 = 0$, $x_0 \neq 0$. The questions is: How do we select the "best" solution? The answer depends on our definition of the "best".

Another situation may be that we wish to minimize $\|Ax - b\|^2$, but also make sure the solution x does not get out of hand by becoming, say, too large or too oscillatory. This happens, for example, when fitting data with a polynomial of high degree. When the degree of the polynomial increases, there is a Runge phenomenon – the interpolating polynomial becomes too oscillatory in between the data.

To address the problem, we add a penalty term to the objective function

$$\text{minimize } \|Ax - b\|^2 + \alpha \|Bx - d\|^2 \tag{4.1}$$
$$\text{by solving } \left(A^T A + \alpha B^T B\right) \hat{x} = A^T b + \alpha B^T d. \tag{4.2}$$

Here we have two objectives. We want to minimize $\|Ax-b\|^2$ making sure that $\|Bx-d\|^2$ is also small. This last quantity could be just $\|x\|$ as before, in which case we are looking to find the smallest minimizer of $\|Ax-b\|^2$.

That solving (4.1) reduces to solving (4.2) is explained as follows. Consider the wish system for which we want to find the least-squares solution:

$$\begin{aligned} Ax =& b, \\ Bx =& d. \end{aligned}$$

However, we want to put a different weight on the second equation, say scale it by a factor of α:

$$\begin{aligned} Ax =& b, \\ \alpha Bx =& \alpha d. \end{aligned}$$

This system can now be written as

$$\begin{bmatrix} A \\ \alpha B \end{bmatrix} x = \begin{bmatrix} b \\ \alpha d \end{bmatrix}$$

or

$$\begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} x = \begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} b \\ d \end{bmatrix}.$$

Thus, defining

$$C = \begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A \\ B \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} b \\ d \end{bmatrix},$$

we have a weighted least-squares problem

$$C\bar{A}x = C\bar{b}.$$

The normal equation for this is:

$$\bar{A}^T C \bar{A} \hat{x} = \bar{A}^T C \bar{b}$$

which becomes

$$\begin{bmatrix} A^T & B^T \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \hat{x} = \begin{bmatrix} A^T & B^T \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\left(A^T A + \alpha B^T B\right) \hat{x} = A^T b + \alpha B^T d.$$

Example 18. The goal is to minimize $f = x_1^2 + x_2^2$ subject to $x_1 - x_2 = 8$. To put it in the above framework, let $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $B = \begin{bmatrix} 1 & -1 \end{bmatrix}$, and $d = 8$. Then we minimize $\|Ax\|^2$ subject to $Bx = d$.

Of course, this problem is so simple that one could simply eliminate $x_2 = x_1 - 8$, substitute it into $f = x_1^2 + (x_1 - 8)^2$ and then set the derivative to zero to find $x_1 = 4$ and then $x_2 = -4$. This approach will be generalized below as the nullspace method.

So, next we do it in three different ways: 1) using the nullspace method, 2) with a penalty method; 3) via the Lagrange multipliers.

(1) *Nullspace method.* The constraint is written as $Bx = d$: $\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 8$. Here B is rank 1, therefore B has a nullspace, $Bx_0 = 0$. It is easy to find: $x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. A particular solution of $Bx = d$ is $x_p = \begin{bmatrix} 8 \\ 0 \end{bmatrix}$. Then the general solution is

$$x = x_p + zx_0 = \begin{bmatrix} 8 \\ 0 \end{bmatrix} + z \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Now, the idea is to substitute this solution into $f = \|Ax\|^2$ and minimize the result over z:

$$f = \|Ax - b\|^2 = \| \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_p + z \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_0\|^2 = \| \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 0 \end{bmatrix} + z \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \|^2$$
$$= \| \begin{bmatrix} 8+z \\ z \end{bmatrix} \|^2 = (8+z)^2 + z^2.$$

Minimizing this, we find $z = -4$ and therefore $x = \begin{bmatrix} 8 \\ 0 \end{bmatrix} + z \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ -4 \end{bmatrix}$, as before.

(2) *Penalty method.* Here we solve

$$\left(A^T A + \alpha B^T B\right) \hat{x} = A^T b + \alpha B^T d,$$

which is equivalent to minimizing $\|Ax - b\|^2 + \alpha \|Bx - d\|^2$. Then the idea is to let $\alpha \to \infty$ to obtain the solution. Since $A^T A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $B^T B = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$, the normal equation becomes

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \alpha \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right) \hat{x} = \alpha \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot 8$$
$$\begin{bmatrix} 1+\alpha & -\alpha \\ -\alpha & 1+\alpha \end{bmatrix} \hat{x} = \begin{bmatrix} 8\alpha \\ -8\alpha \end{bmatrix}$$
$$\hat{x} = \frac{1}{1+2\alpha} \begin{bmatrix} 1+\alpha & \alpha \\ \alpha & 1+\alpha \end{bmatrix} \begin{bmatrix} 8\alpha \\ -8\alpha \end{bmatrix}$$
$$\hat{x} = \frac{8\alpha}{1+2\alpha} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{4}{\frac{1}{2\alpha}+1} \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

As $\alpha \to \infty$, the factor in front tends to 4 and therefore the solution tends to $x_1 = 4$ and $x_2 = -4$.

(3) *Lagrange multipliers.* Define the Lagrange function

$$\begin{aligned} L &= \frac{1}{2}\|Ax - b\|^2 + \lambda^T (Bx - d) \\ &= \frac{1}{2} (Ax - b)^T (Ax - b) + \lambda^T (Bx - d) \\ &= \frac{1}{2} x^T A^T A x - x^T A^T b + \lambda^T (Bx - d), \end{aligned}$$

and form the saddle point system:

$$\frac{\partial L}{\partial x} = 0 : \qquad A^T Ax - A^T b + B^T \lambda = 0,$$
$$\frac{\partial L}{\partial \lambda} = 0 : \qquad Bx = d.$$

These can be combined into

$$\begin{bmatrix} A^T A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} A^T b \\ d \end{bmatrix}.$$

Note now that this saddle-point matrix may be singular. In our particular example it is not singular:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 8 \end{bmatrix}.$$

This is solved by

$$\begin{bmatrix} x_1 \\ x_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} 4 \\ -4 \\ -4 \end{bmatrix}.$$

4.6.1. *Tychonov regularization.* The problem is to minimize $\|Ax - b\|^2$ when A has dependent columns. Then $A^T A$ is no longer invertible and $A^T Ax = 0$ has nonzero solutions. We have to regularize the problem. This is done by requiring that $\|Ax - b\|^2$ be minimized together with the size of the solution, $\|x\|^2$. That is, out of the many solutions that minimize $\|Ax - b\|^2$, we want the one that has smallest length, $\|x\|^2$. This will lead to the pseudo-inverse as shown earlier as well.

The regularized problem is stated as follows:

$$\text{minimize } \|Ax - b\|^2 + \delta^2 \|x\|^2 \text{ by solving } \left(A^T A + \delta^2 I\right) \hat{x} = A^T b.$$

This is called *ridge regression* as the regularizing term $\delta^2 I$ is added to the *ridge (diagonal)* of the singular matrix $A^T A$. It is also called *Tychonov regularization.* It is the special case of the previous problem with $B = I$, $d = 0$, and $\alpha = \delta^2$.

Two interesting facts:

(1) $A^T A + \delta^2 I$ is invertible for any $\delta \neq 0$, no matter how singular $A^T A$ is.

(2) $\lim\limits_{\delta \to 0} \hat{x} = x^+ = A^+ b$, where A^+ is the pseudo-inverse.

To establish fact 1, we write $A = U \Sigma V^T$, where A is $m \times n$, maybe with $n > m$, but in any case with dependent columns, U is $m \times m$, Σ is $m \times n$, and V is $n \times n$. Then $A^T A = V \Sigma^T \Sigma V^T$ and

$$A^T A + \delta^2 I = V \Sigma^T \Sigma V^T + \delta^2 V V^T = V \left(\Sigma^T \Sigma + \delta^2 I\right) V^T.$$

Suppose A is wide and has independent rows, so that rank is m. Then $\Sigma^T \Sigma$ is $n \times n$ with m singular values, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m > 0$ down the main diagonal and the rest of the diagonal zero. We now add δ^2 to the entire diagonal of $\Sigma^T \Sigma$, and that will make the matrix invertible for any $\delta \neq 0$.

Then

$$\hat{x} = \left(A^T A + \delta^2 I\right)^{-1} A^T b = V \left(\Sigma^T \Sigma + \delta^2 I\right)^{-1} V^T V \Sigma^T U^T = V \left[\left(\Sigma^T \Sigma + \delta^2 I\right)^{-1} \Sigma^T\right] U^T.$$

The questions is now: What is the limit of the inverse in the last expression when $\delta \to 0$?

Consider a simple special case to see what happens. If A is just a number, σ, then

$$\left(A^TA+\delta^2 I\right)^{-1}A^T=\frac{\sigma}{\sigma^2+\delta^2}\underset{\delta\to 0}{\to}\begin{cases}0, & \sigma=0\\ \frac{1}{\sigma}, & \sigma\neq 0.\end{cases}$$

That is, this limit is discontinuous. Same way

$$\left(\Sigma^T\Sigma+\delta^2 I\right)^{-1}\Sigma^T=$$

$$n\times n:\begin{bmatrix}\sigma_1^2+\delta^2 & & & & & & & \\ & \sigma_2^2+\delta^2 & & & & & & \\ & & \ddots & & & & & \\ & & & \sigma_m^2+\delta^2 & & & & \\ & & & & \delta^2 & & & \\ & & & & & \ddots & & \\ & & & & & & \delta^2 & \end{bmatrix}^{-1}\begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_m \\ & & & \end{bmatrix}_{n\times m}=$$

$$n\times m:\begin{bmatrix}\frac{\sigma_1}{\sigma_1^2+\delta^2} & & & \\ & \frac{\sigma_2}{\sigma_2^2+\delta^2} & & \\ & & \ddots & \\ & & & \frac{\sigma_m}{\sigma_m^2+\delta^2} \\ & & & \end{bmatrix}\to\begin{bmatrix}\frac{1}{\sigma_1} & & & \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_m} \\ & & & \end{bmatrix}_{n\times m}=\Sigma^+.$$

And this establishes the fact 2: the limit of $\hat{x}$ is $V\Sigma^+U^Tb$ which is exactly $x^+=A^+b$.

The bottom line: When A has dependent columns, the solution of $Ax=b$ given by $\text{pinv}(A)\,b=A^+b$ is the smallest least squares solution minimizing $\|Ax-b\|^2$. Smallest meaning $\|x\|$ is as small as possible out of all minimizers of $\|Ax-b\|^2$.

Remark 19. It is an important problem to minimize $\|Ax-b\|^2$ with the penalty term of the form $\lambda\|x\|_1$, i.e. the l_1-norm. This time, the solution is *sparse,* i.e., it has many zero components in x. This is important because if one knows that the solution is sparse, than one should find the minimum with the l_1 norm instead of the usual l_2.

4.7. **Weighted least squares.** We return to the least-squares problem, $Ax=b$, and ask a question:

- ◇ If we have some error in b, with the covariance matrix V, what is the corresponding error in the least-squares solution u of $A^TAu=A^Tb$? Each component of b, b_i has an error e_i. The covariance matrix has variances of e_i on the main diagonal and covariances among e_i and e_j off the main diagonal.
- ◇ In other words, we want to estimate the *reliability* of such a solution. This reliability is measured by the covariance matrix W of the error in u. If the variances on the main diagonal of W are small, that means that the solution u is reliably predicted even though there are errors in b. Next, we need to derive an expression for W.

Since not all equations in $Ax=b$ are equivalent in terms of the error they introduce into the solution, we divide these equations by various coefficients that are supposed to give *different*

weights to different equations. Of course, at this point we have no clue which equations should be given higher weight, which lower.

The rescaling is provided by multiplication with a matrix C, to be determined:

$$CAx = Cb.$$

Now we form the normal equation:

$$A^T CAu = A^T Cb,$$

from which we get

$$u = \left(A^T CA\right)^{-1} A^T Cb = Lb.$$

Note that $LA = I$.

We want covariances for the output error: $x - u$.

Using $u = Lb$ and $x = Ix = LAx$, we get

$$x - u = L\left(Ax - b\right) = -Le,$$

so this relates the error in the rhs and the error in the solution.

Next,

$$W = E\left[(x-u)\left(x-u\right)^T\right] = E\left[Lee^T L^T\right] = LE\left[ee^T\right]L^T = LVL^T.$$

We took L outside of the expectation assuming it is constant.

The next step is to minimize this W by choosing appropriate C in L.

Theorem 20. *The best W is obtained with $C = V^{-1}$, in which case $W = \left(A^T V^{-1} A\right)^{-1}$.*

Proof. To verify, just plug in $C = V^{-1}$ into $L = \left(A^T CA\right)^{-1} A^T C$ to obtain L^* and then compute LVL^T.

If we take a different choice of C it will change L^* to $L = L^* + (L - L^*)$, but then W will be bigger, in the sense that a positive semi-definite matrix will be added. Indeed,

$$W = LVL^T = L^*VL^{*T} + (L-L^*)VL^{*T} + L^*V(L-L^*)^T + (L-L^*)V(L-L^*)^T.$$

The two middle terms can be shown to be zero. The last term is positive semi-definite, and therefore W is smallest when the term vanishes, i.e. when $L = L^*$. □

Matrix $W^{-1} = A^T V^{-1} A$ is called the *information matrix.* When V is small, i.e. variances are small, hence better accuracy, this matrix is large, contains more information. Furthermore, W^{-1} also increases, when more data are added that increase the matrix A with more rows.

Note that W does not depend on b, only on the error in b and matrix A.

Example 21. Suppose a doctor measures your heart rate x three times ($m = 3$, $n = 1$). Then $Ax = b$ with

$$A = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad b = \begin{bmatrix} 60 \\ 70 \\ 90 \end{bmatrix}, \quad \text{so } x = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 60 \\ 70 \\ 90 \end{bmatrix}.$$

And assume that $V = diag\begin{pmatrix} \sigma_1^2, & \sigma_2^2, & \sigma_3^2 \end{pmatrix} = diag\begin{pmatrix} \frac{1}{9}, & \frac{1}{4}, & 1 \end{pmatrix}$, which implies that with every measurement, the error has increased. Something went wrong between you and the doctor. We want the best estimate of the heart rate.

The weighted least-squares solution is obtained by solving $A^TV^{-1}Au = A^TV^{-1}b$, which is

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 9 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 9 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 60 \\ 70 \\ 90 \end{bmatrix}.$$

Solving this, we get

$$u = \frac{9b_1 + 4b_2 + b_3}{14} = 65$$

as the best weighted average. The variance of u is

$$W = \left(A^TV^{-1}A\right)^{-1} = \left(\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 9 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)^{-1} = \frac{1}{14},$$

which is smaller than $\frac{1}{9}$ in the first measurement b_1 due to accounting for the additional measurements.

4.8. **Recursive least squares and the Kalman filter.** How do we update the solution of the least-squares problem when new data and new equations come in using the solution already found before?

Let

$$A_0x_0 = b_0, \Rightarrow A_0^TV_0^{-1}A_0u_0 = A_0^TV_0^{-1}b_0, \quad \text{old problem}$$

and with the new data coming in A_1 and b_1 we need to solve a new system:

$$\begin{bmatrix} A_0 \\ A_1 \end{bmatrix} x_1 = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}, \quad \text{new problem}$$

$$\Rightarrow \begin{bmatrix} A_0^T & A_1^T \end{bmatrix} \begin{bmatrix} V_0^{-1} & \\ & V_1^{-1} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} u_1 = \begin{bmatrix} A_0^T & A_1^T \end{bmatrix} \begin{bmatrix} V_0^{-1} & \\ & V_1^{-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

Now u_1 is found as an update to u_0:

$$u_1 = u_0 + K_1\left(b_1 - A_1u_0\right)$$

with Kalman gain matrix K_1 and the mismatch between the previous state A_1u_0 and the new measurement b_1.

Comparing the solutions for u_0 and u_1, we find

$$K_1 = W_1A_1^TV_1^{-1},$$

where the covariance of errors in u_1 is

$$W_1^{-1} = W_0^{-1} + A_1^TV_1^{-1}A_1,$$

which is found using the earlier results and (maybe) the Sherman-Morrison-Woodbury formula for the inverse $\left(A - UV^T\right)^{-1}$,

$$\left(A - UV^T\right)^{-1} = A^{-1} + A^{-1}U\left(I - V^TA^{-1}U\right)^{-1}V^TA^{-1},$$

which becomes

$$M^{-1} = I + \frac{uv^T}{1 - v^Tu}$$

when rank-1 correction is considered. The old covariance matrix was

$$W_0 = \left(A_0^TV_0^{-1}A_0\right)^{-1}.$$

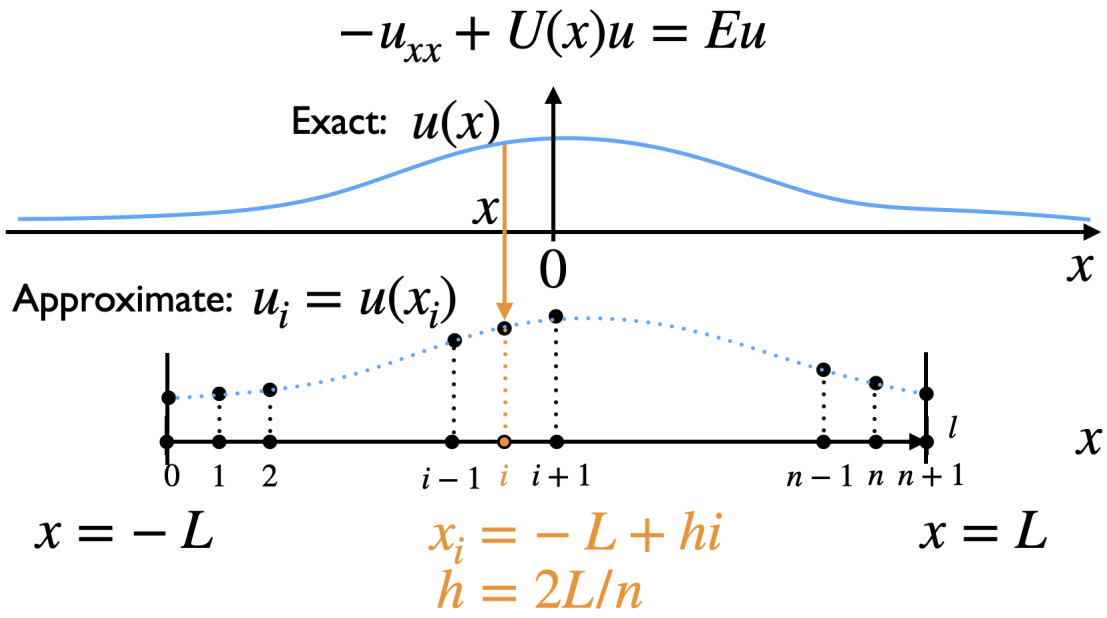

FIGURE 4.4. Solving Shrödinger's equation numerically.

Extra: A note on the numerical solution of the Shrödinger equation in pset 2.

(1) Consider the steady Shrödinger's equation for a harmonic oscillator which in dimensional variables is given by $-\frac{\hbar^2}{2m}u'' + Uu = Eu$, where $u(x)$ is the wave function, $U = kx^2/2$ is the potential energy of the particle, and E is the total energy. After some clean-up, the equation can be written as

$$-u'' + x^2 u = \lambda u, \tag{4.3}$$

where $\lambda = 2E/\hbar\omega$ and $\omega = \sqrt{k/m}$. The boundary conditions are $u(\pm\infty) = 0$.

(a) Find the eigenvalues λ of (4.3) numerically using a second derivative matrix D_n coming from $u'' \approx (u_{i-1} - 2u_i + u_{i+1})/h^2$, the grid points $x = x_i = -L + ih$, $i = 0, 1, 2, ..., n+1$, the grid size $h = 2L/(n+1)$, some large L (say 10), and for sufficiently large number of grid point n, say 50 or 100. It is known that the energy of the harmonic oscillator is quantized as $E_s = \hbar\omega\left(s + \frac{1}{2}\right)$, $s = 0,\ 1,\ 2, ...$, therefore the eigenvalues λ better be close to $1,\ 3,\ 5, \ldots$.

(b) Plot the eigenvectors that correspond to the lowest five eigenvalues and on a separate plot show the first ten numerically found eigenvalues λ together with their theoretical values.

The important thing to do here is to evaluate the equation not everywhere on the real line, but at some discrete collection of points (grid points, x_i) on some finite but large enough domain $[-L, L]$. Take a uniform set of grid points by, for example, $x_i = -L + hi$, $i = 1, 2, ..., n$, where $h = 2L/n$ and n is the number of points that you decide to choose. Then $x_1 = -L + h$, $x_2 = -L + 2h$, ..., $x_n = -L + 2L = L$.

The more points you take, the more accurately you can expect to represent the solution on this interval $[-L, L]$.

Then

$$\left(-u'' + x^2 u\right)_{x=x_i} = (\lambda u)_{x=x_i}$$

is the Shrödinger's equation sampled at the grid points x_i. All terms here can be evaluated as follows:

$$\left(\lambda u\left(x\right)\right)_{x=x_i} = \lambda u\left(x_i\right) = \lambda u_i$$
$$\left(x^2 u\left(x\right)\right)_{x=x_i} = x_i^2 u\left(x_i\right) = x_i^2 u_i,$$

where $u\left(x_i\right)$ is defined as u_i, which is now a number, not a function anymore. But it must be remembered that this number comes from a function, it is a sample of $u\left(x\right)$ at $x = x_i$.

Finally, the derivative is approximated as

$$u'' \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}.$$

Then the sampled equation becomes

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + x_i^2 u_i = \lambda u_i, \quad i = 1, 2, 3, ..., n$$

One then has to put these equations into a matrix equation for the vector $u = [u_1, u_2, ..., u_n]^T$:

$$\left(\underbrace{\frac{1}{h^2}\begin{bmatrix} 2 & -1 & 0 & \cdots & \\ -1 & 2 & -1 & 0 & \\ 0 & -1 & 2 & -1 & 0 \\ \vdots & & \ddots & & -1 \\ & & & -1 & 2 \end{bmatrix}}_{K_n} + \underbrace{\begin{bmatrix} x_1^2 & & & \cdots & \\ & x_2^2 & & & \\ & & & & \\ \vdots & & \ddots & & \\ & & & & x_n^2 \end{bmatrix}}_{D_n} \right) u = \lambda u.$$

Now we see that u is the e-vector of $A = K_n + D_n$ and λ is the e-value of A. Finding these is equivalent to finding the solution of the Shrödinger's equation evaluated at the set of points x_i, $i = 1, 2, 3, ..., n$.

The problem as stated in the homework is an eigenvalue problem as well, not for a matrix, but for a differential operator

$$\hat{L} = -\frac{d^2}{dx^2} + x^2, \text{ so that } \hat{L}u = \lambda u.$$

We have converted this problem to an eigenvalue problem for a matrix A: $Au = \lambda u$, where A approximates $\hat{L}$, u a vector approximating a function $u\left(x\right)$ and λ approximating the eigenvalue.

To appreciate the power of this kind of numerical experimentation, take a look at the Matlab code that does the computation. The essential elements of that code are just these lines:

```
L = 10; %the x-domain is [-L, L] with large enough L to represent infinity
n = 200; %the number of grid points on the domain
h = 2*L/n; %step size in x to approximate derivatives: u" = (u(i+1)-2*u(i)+u(i-1))/h^2
x = -L + h*(1:n)'; %locations of the grid points
Kn= 1/h^2*toeplitz([2 -1 zeros(1,n-2)]); %the 2nd derivative matrix
Kn(end,1) = 1; Kn(1,end) = 1; %boundary conditions
A = Kn + diag(x.^2); %the full matrix to represent [-()"+x^2]
[V D] = eig(A); %e-values/vectors of A
```

Solving a different eigenvalue problem for a new potential energy function, say $U = x^4$, would entail just changing the **diag(x.^2)** to a new line, say **diag(x.^4)**. Everything else would

remain the same. This allows us to find the spectrum of a problem that could be extremely complicated or impossible to solve analytically with an ease that we almost do not deserve.

```
%% Eigenvalues of the Shroedinger equation for the harmonic oscillator
% Find eigenvalues of -u'' + x^2*u =lam*u subject to u(-inf)=u(inf)=0
clear all; clc;

L = 10;          %the x-domain is [-L, L] with large enough L to represent infinity
n = 200;         %the number of grid points on the domain
h = 2*L/n;       %step size in x to approximate derivatives:
                 % u'' = (u(i+1)-2*u(i)+u(i-1))/h^2
x = -L + h*(1:n)';                          %locations of the grid points
Kn= 1/h^2*toeplitz([2 -1 zeros(1,n-2)]);    %the 2nd derivative matrix
Kn(end,1) = 1; Kn(1,end) = 1;               %boundary conditions
A = Kn + diag(x.^2);                        %the full matrix to represent [-()''+
[V D] = eig(A);                             %e-values/vectors of A

nn = [1 2 3];                               %which e-vectors to plot
subplot(311), plot(x,V(:,nn),'.-','MarkerSize',15), grid on
legend(int2str(nn'));
title(['Wavefunctions of quantum harmonic oscillator at L=',int2str(L),', n=',int

eig_exact = 2*(0:size(D,1))' + 1;           %exact e-values from QM books
eig_numer = diag(D);                        %numerically found values
eig_end = min(20,n);                        %how many e-values to show
subplot(312), plot(eig_exact(1:eig_end),'gx-','MarkerSize',10), grid on, hold on
subplot(312), plot(eig_numer(1:eig_end),'r.','MarkerSize',25), hold off
legend('exact','numerical','Location','NW'); title('Eigenvalues of quantum harmo

err = (eig_exact(1:eig_end) - eig_numer(1:eig_end))./eig_exact(1:eig_end);
subplot(313), semilogy(abs(err),'b.-','MarkerSize',10), grid on
title('Relative error in eigenvalues, |\lambda_{exact}-\lambda_{numerical}|/|\lamb
```

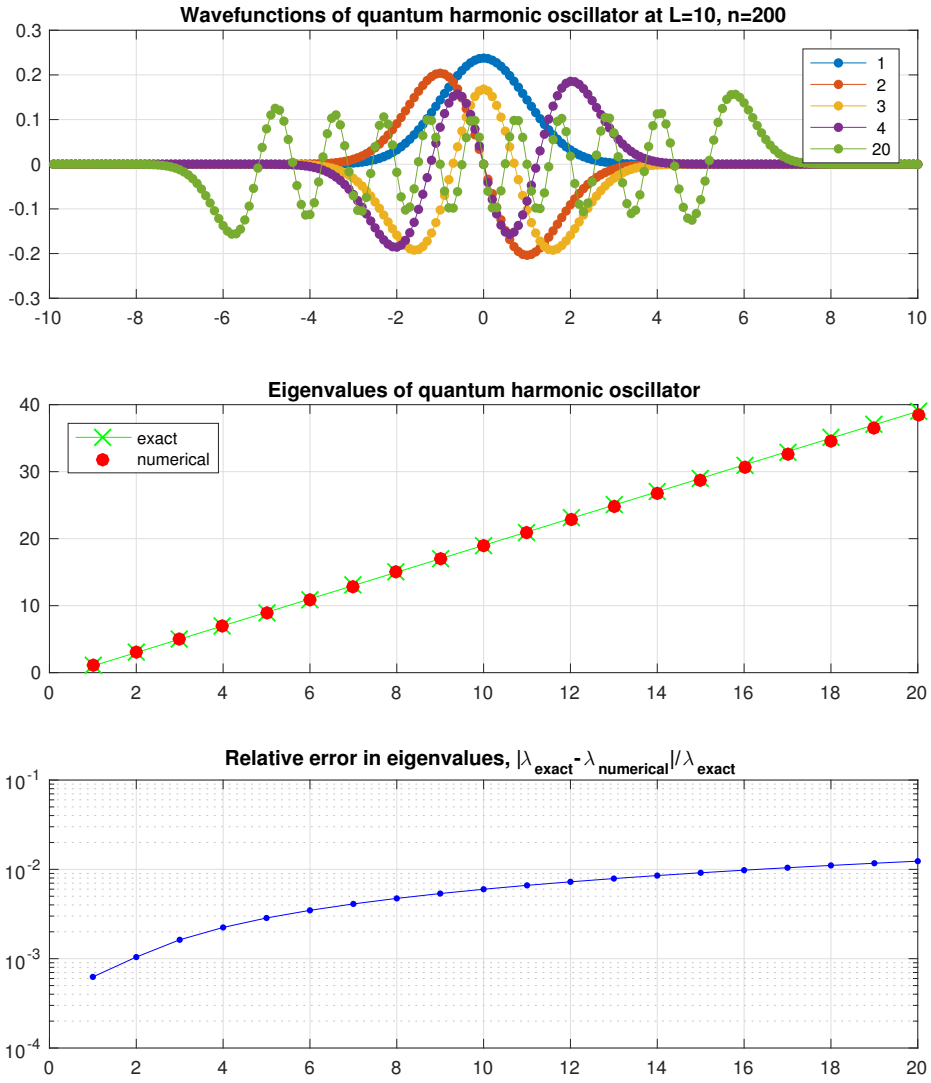

FIGURE 4.5. Spectrum of the quantum harmonic oscillator.