Homework: Practical Lanczos

October 23, 2023

This is an assignment for the course 'Numerical Linear Algebra'. The goal of this assignment is to explore Lanczos as a method of eigenvalue approximation.

Please note that the questions are deliberately open-ended! Unlike in the exercise sessions, you are encouraged to make your own analysis, provide your own results and construct your own report in your own way. The assignment is individual. Don't be discouraged if not everything works in the end, but show your work and report what you tried and whether you know why something did not work.

Your report should be a standalone text, should read pleasantly and should not refer to this assignment. Take special care of structure and visual presentation. You are allowed to copy text from this assignment document.

The goals of this assignment are:

- **Understanding why theoretical Lanczos does not work, and why Lanczos as you have seen it in the exercise sessions is not practical**
- **Implementing a practical Lanczos Algorithm**
- **Understanding the benefits of tridiagonalization**
- **Implementing a tridiagonal eigenvalue algorithm**
- **Applying this to a given realistic application**

1 Semi-orthogonality and strategic reorthogonalization

Recall from the exercise sessions the Lanczos algorithm (here given slightly rewritten):

Algorithm 1: Lanczos in exact arithmetic

```
input  : Linear, symmetric real operator $A$ on $\mathbb{R}$
output: Approximately orthogonally similar tridiagonal $T \sim A$,
        given by diagonals $\alpha, \beta$
1 init choose starting vector $\mathbf{r}_0$,$\beta_0 := \|\mathbf{r}_0\|$, $\mathbf{q}_0 = 0$
2 for $j = 1, 2, ...$ do
3     $\mathbf{q}_j := \mathbf{r}_{j-1}/\beta_{j-1}$
4     $\mathbf{v}_j = A\mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1}$
5     $\alpha_j = \mathbf{v}_j^*\mathbf{q}_j$
6     $\mathbf{r}_j = \mathbf{v}_j - \alpha_j\mathbf{q}_j$
7     $\beta_j = \|\mathbf{r}_j\|$
8 end
```

The matrix T then has the form

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \beta_{j-2} & \alpha_{j-1} & \beta_{j-1} \\ 0 & \cdots & 0 & \beta_{j-1} & \alpha_j \end{pmatrix}$$

One step in the main loop of algorithm 1 is called a Lanczos step, and is usually written as

$$\beta_j \mathbf{q}_{j+1} = A\mathbf{q}_j - \alpha_j \mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1}. \tag{1}$$

As you know, with $Q_j := [\mathbf{q}_1, \ldots, \mathbf{q}_j]$, we have the relation

$$AQ_j = Q_j T_j + \beta_j \mathbf{q}_{j+1}\mathbf{e}^*. \tag{2}$$

In exact arithmetic the above is a great procedure for tridiagonalization. Numerically this is not the case unfortunately.

1.1 Loss of orthogonality

All ‘practical’ routines you have seen so far fix loss of orthogonlity by orthogonalising against all Lanczos vectors, and in some cases even reorthogonalising agains a portion of them (**Why is ‘practical’ in quotes?**). A very natural question to ask is whether orthogonalization is w.r.t. all previous Lanczos vectors is necessary. The answer to this is no! This section will show precisely why and where orthogonality is lost. Key to this is to analyse the elements w_{ik} of the matrix $W_j := Q_j^* Q_j$, which measures the orthogonality of the Lanczos vectors. Note that for real valued Lanczos, which is what we consider here, W is symmetric. **Run simple Lanczos on the system in `Test.mtx` and report on the matrix W_j for large j in a clear and insightful way. What do you observe? Where is orthogonality lost? Where does convergence of Ritz values start to faulter? In what way?** This leads us to a very simple modified Lanczos scheme: simply keep track of

$$w_{j,\infty} := \|W_j - I_j\|_{max}$$

and as soon some threshold is reached for a **proposed** Lanczos vector $\mathbf{q}'_j$, reorthogonalize $\mathbf{q}'_j$ before computing β_j and follow up by reorthogonalizing $\mathbf{q}'_{j+1}$ as well. **Why also $\mathbf{q}_{j+1}$? Argue that this scheme is still not very practical. You have been given an incomplete `MatLab`function ‘`Lanczos_HW1.m`’. You will be asked to complete this code in three increasing steps of complexity, `Lanczos1.m`, `Lanczos2.m` and (optionally) `Lanczos3.m`. Each step either implements a way of detecting loss of orthogonality or a way of orthogonalizing against a subset of vectors. In `Lanczos1.m`, implement the above, i.e. compute the loss of orthogonality by computing $w_{j,\infty}$. Test your implementation on `Test.mtx` (matrix attached to assignment).** When it comes to the threshold to use, we can introduce the concept of *semi-orthogonality*: we say Q_j is semiorthogonal if $w_{j,\infty} = \mathcal{O}(\sqrt{\epsilon})$. Why this is sufficient is captured in the following theorem. **You do not need to prove this. In the given code this threshold is given as `delta`**.

Theorem 1 *With notation from before, let P_j denote the orthogonal basis for span(Q_j). If $w_{j,\infty} < \sqrt{\frac{\epsilon}{j}}$ then*

$$\|P_j^* A P_j - T_j\|_{max} = \mathcal{O}(\epsilon \|A\|).$$

We can improve upon `Lanczos1.m`, by gaining more insight into W.
To study W, we begin by re-writing equations 1 and 2 to

$$\beta_j \mathbf{q}_{j+1} = A\mathbf{q}_j - \alpha_j \mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1} - \mathbf{f}_j \tag{3}$$

and

$$AQ_j = Q_j T_j + \beta_j \mathbf{q}_{j+1}\mathbf{e}_j^* + F_j \tag{4}$$

where $\mathbf{f}_j$ is an error vector and F_j a corresponding error matrix. We assume $\|F_j\| \leq \epsilon\|A\|$, with ϵ machine precision. For simplicity we make the following two assumptions for all $k \in \{1, \ldots, j\}$:

- $\|\mathbf{q}_k\|_2 = 1$
- $\beta_k \mathbf{q}_{k+1}^* \mathbf{q}_k = C\epsilon\|A\|$, with C a modest constant.

Strictly speaking they are not necessary, but they greatly simplify the derivations. **Interpret these assumptions. Are they likely/defensible? Using the above considerations, show the following recursion:**

Theorem 2 *Let $W_j := Q_j^* Q_j$, with Q_j the matrix of Lanczos vectors found by algorithm 1. Then, with $w_{j0} := 0$:*

$$\begin{aligned}
w_{kk} &= 1 && \textit{for } k = 1, \ldots, j \\
w_{kk-1} &= \underbrace{\mathbf{q}_k^* \mathbf{q}_{k-1}}_{:=\psi_k} && \textit{for } k = 2, \ldots, j \\
w_{jk+1} &= w_{k+1j} && \textit{for } k = 1, \ldots, j-1 \\
\beta_j w_{j+1k} &= \beta_k w_{jk+1} + (\alpha_k - \alpha_j) w_{jk} + \beta_{k-1} w_{jk-1} - \beta_{j-1} w_{j-1k} + \underbrace{\mathbf{q}_j^* \mathbf{f}_k - \mathbf{q}_k^* \mathbf{f}_j}_{:=\theta_{j,k}}
\end{aligned}$$

where the last equation holds for $k = 1, \ldots, j-1$.

What does this imply? Incorporate your findings into a new algorithm, `Lanczos2.m`, that computes loss of orthogonality using this. Don't forget to update w afterwards! Is this scheme practical?
You can take $\theta_{jk} = \epsilon(\beta_j + \beta_k)z_1$ with $z_1 \sim \mathcal{N}(0, .3)$ and $\psi_k = \sqrt{n}\epsilon(\beta_1/\beta_j)z_2$ with $z_2 \sim \mathcal{N}(0, .6)$. You can update the selected elements in w to ϵz_3, with $z_3 \sim \mathcal{N}(0, 3/2)$. You do not need to worry where these numbers come from. Finally we will make a little extra effort to carefully select the correct Lanczos vectors to orthogonalize against. **Halt `Lanczos2.m` at a couple of**

detections of loss of orthogonality. Inspect $Q_j^*\mathbf{q}'_{j+1}$. What do you observe? Plot the clearest case. Knowing this, does it make sense to orthogonalize only against those $\mathbf{q}_i$ such that $|w_{ij+1}| > \sqrt{\epsilon}$? Use this to motivate the following strategy for orthogonalization:

(1) Detect loss of orthogonality using theorem 2

(2) Construct $\mathcal{L}(j) := \{\mathbf{q}_i : |\mathbf{q}_{j+1}^*\mathbf{q}_i| > \sqrt{\epsilon/j_{\max}}\}$ (approximately).

(3) For each $\mathbf{q}_i \in \mathcal{L}$, add the vectors $\mathbf{q}_{i-s_1}, \mathbf{q}_{i-s_1+1}, \ldots, \mathbf{q}_i, \ldots, \mathbf{q}_{i+s_2-1}, \mathbf{q}_{i+s_2}$ s.t. $\mathbf{q}_{i-s_1}$ and $\mathbf{q}_{i+s_2}$ are the first vectors to the left and right of $\mathbf{q}_i$ for which $|w_{i-s_1,j+1}|, |w_{i-s_2,j+1}| \leq \eta$. You can take $\eta = \epsilon^{3/4}/\sqrt{j_{\max}}$.

1.2 Optional: Lanczos 3

Optionally, you can implement this last version. Call this procedure `Lanczos3.m` and test your routine on `Test.mtx`. Provide numerical results and motivate your choices. This part of the Homework can only count positively towards your grade.

1.3 The unavoidability of orthogonality loss

Now suppose that $W_j := I_j + U_j + U_j^*$ with U_j strictly upper triangular. Suppose the columns of U are given by $U = [\mathbf{u}_1, \ldots, \mathbf{u}_j]$. Show then that we can write the above as

$$\beta_j Q_j^*\mathbf{q}_{j+1} = T_j\mathbf{u}_j - \alpha_j\mathbf{u}_j - \beta_{j-1}\mathbf{u}_{j-1} + \underbrace{(F_j^*\mathbf{q_j} - Q_j^*\mathbf{f}_j)}_{:=\mathbf{g}_j} + \mathbf{e}_j(\beta_j\mathbf{q}_{j+1}^*\mathbf{q}_j - \beta_{j-1}\mathbf{q}_j^*\mathbf{q}_{j-1})$$

Group these expressions, indexed over j, together into an expression

$$\beta_j Q_j^*\mathbf{q}_{j+1}\mathbf{e}_j^* = T_jU_j - U_jT_j + G_j$$

Show this. What is G_j? We are now ready to prove Paige's lemma:

Lemma 1 (Paige, 1971) *Let T_j denote the Lanczos matrix for some given A at iteration j. Suppose T_j has an eigendecomposition $T_jS_j = S_j\Theta_j$, with* *associated Ritz vectors* $Y = Q_j S_j = Q_j[\mathbf{s}_1, \dots, \mathbf{s}_j]$. *Say* $s_{ji} = \mathbf{e}_j^* \mathbf{s}_i$. *Let* (θ_i, y_i) *be a Ritz pair at this iteration for* T_j. *Define*

$$\beta_{ji} := \|Ay_i - \theta_i y_i\|_2 = \beta_j |s_{ji}|$$

Then

$$|y_i^* q_{j+1}| = \frac{|\gamma_{ii}|}{\beta_{ji}}$$

with $\gamma_{ii} = \mathbf{s}_i^* G_j \mathbf{s} \approx \epsilon \|A\|$.

Prove this lemma, including $\|Ay_i - \theta_i y_i\|_2 = \beta_j |s_j i|$. Using the above lemma, justify the following claim: "Orthogonality is lost only, and always, in the direction of converging Ritz vectors". Additionaly, find an explicit expression for the β_{ji} 's in terms of the eigenvectors of T_j. It is possible to leverage the above observation into a functioning scheme, called *selective orthogonalization*. However, this implementation is quite complicated and not very performant. Therefore, we will not elaborate on this.

2 The eigenvalues of a tridiagonal matrix

In this section you will learn why tridiagonalisation is desired. Suppose a tridiagonal symmetric (real) matrix T is given:

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & \cdots & 0 & \beta_{n-1} & \alpha_n \end{pmatrix}$$

Let $T_j := T(1:j, 1:j)$ as before. Now let p_j denote the characteristic polynomial of T_j (with $p_0 = 1$). **Show that**

$$p_j(x) = (\alpha_j - x)p_{j-1}(x) - \beta_{j-1}^2 p_{j-2}(x)$$

for $j \geq 2$ and for all $x \in \mathbb{R}$. **Use this to show that p_n can be evaluated in O(n) floating point operations.** An important property of tridiagonal symmetric matrices is the Sturm Sequence Property:

Theorem 3 (Sturm sequence property) *Suppose T is such that $\{\beta_i\}_{i=1}^{n-1}$ contains no zeros. Then for any $j \in \{1, \ldots, n\}$, the eigenvalues of T_j and T_{j-1} interlace as follows:*

$$\lambda_j(T_j) < \lambda_{j-1}(T_{j-1}) < \lambda_{j-1}(T_j) < \cdots < \lambda_2(T_j) < \lambda_1(T_{j-1}) < \lambda_1(T_j)$$

.

Why are zero β's in T not actually a problem? From the Sturm sequence property another important result can be derived. Suppose $s(x)$ denotes the number of sign changes in the sequence $p_0(x), \ldots, p_n(x)$. The convention is adopted that if $p_j(x) = 0$, then it has opposite sign to $p_{j-1}(x)$. The result is then that for any x, $s(x)$ equals the number of eigenvalues of T that are less than or equal to x. **Interpret this result carefully and prove it using the Sturm sequence property. Use it in combination with the Gershgorin circle theorem to construct an algorithm that finds the kth eigenvalue of a given tridiagonal matrix. Take inspiration from the bisection method for finding roots of polyomials, i.e. let your algorithm produce a sequence of intervals, decreasing in size, which all contain the kth eigenvalue. You can assume there are no repeated eigenvalues.**
Can you create a more sophisticated algorithm that recycles information to find the first k eigenvalues efficiently?
Comment on the stability of the above recursive procedure. Can you find a more stable version in the literature? Would you always need to use a more stable implementation?

3 Application: Atomic force microscopy

3.1 Background

Atomic force microscopy (AFM) is a microscopy technique that works on the same length scales a scanning tunneling microscopy (STM), but does not need its subject to be electrically conductive. The driving transmitted of information is not tunneling electons, but various atomic forces. This makes AFM better suited to study proteins, blood cells, etc. than STM.

The basic priciple is that a cantilever with a small tip at its free end is brought close to, or in contact with a material or material surface. A concentrated

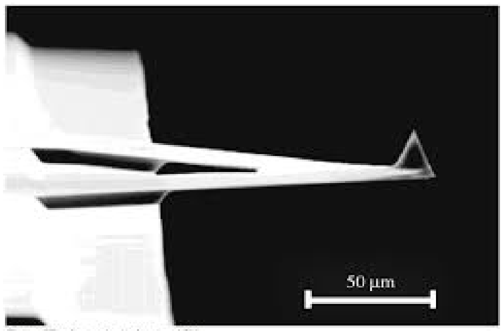
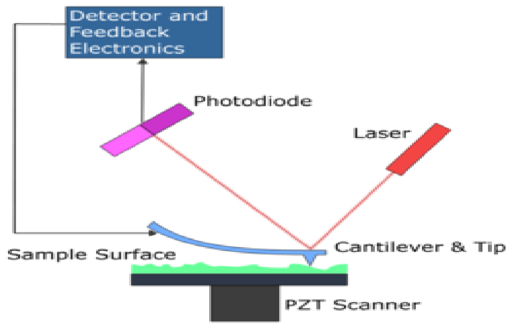


Figure 5 The fine probe tip from an AFM



light beam is bounced off of the cantilever and captured on a light sensitive receiver. Then the movement of the cantilever due to atomic forces can be recorded and information on the sample can be deduced. AFM comes in three modes: contact mode, tapping mode and non-contact mode.For this homework, we are only interested in the non-contact mode.

In non-contact atomic force microscopy the cantilever tip does not come in contact with the sample surface, but rather sits at some distance, where chosen non-contact forces are strongest. For instance the van der Waals forces are strongest from 1 nm to 10 nm above the surface. The cantilever is oscillated at one of its resonant frequencies (frequency modulation) or just above one (amplitude modulation). The long-range forces above the surface decrease the resonance frequency of the cantilever. By moving the tip closer and further this decrease can be counteracted. As such, a constant oscillation (in amplitude or frequency) is created, which can be used to measure the distance to the sample. Since this mode of AFM operation does not actually contact the sample, it is much more suited for the observation of delicate systems.

Additionally, the cantilever oscillation frequencies and modes can provide insight into the types of forces acting on the AFM system and even on the indiviual atoms that the sample is made up of. It should be clear to you from these considerations that a rigorous modal analysis of the AFM cantilever is absolutely vital. This is what we will do in this final part of the assignment.

3.2 Modal analysis using the finite element method

In a modal analysis the typical assumption is that of undamped harmonic motion without driving forces. Whenever the given volume is discretized to a mesh and a choice of basis functions is made (taking boundary conditions into account) this leads to a system

$$(K - \omega^2 M)\delta\mathbf{u} = 0$$

with

$$\delta\mathbf{u} = \begin{pmatrix} \delta\mathbf{u}_x \\ \delta\mathbf{u}_y \\ \delta\mathbf{u}_z \end{pmatrix}$$

a vector of displacements. Such a problem is called a generalized eigenvalue problem. The matrix K is called the stiffness matrix, and the matrix M is called the mass matrix. Note that each node of the given grid produces three degrees of freedom (dof). Elements in the stiffness matrix correspond to the stiffness at dof-dof interactions (e.g. the xy-stiffness at a given node), and the elements of the mass matrix correspond to the masses at the corresponding dof-dof interactions.

Knowing this, analyze the structure of the matrices and give an explanation (you do not have to go into too much detail). Think of the application of AFM, can you formulate what the boundary conditions should be? You have been provided code for the visualization of a given mesh as well as code for the visualization of a reduction of the mesh superimposed with a vibrational mode, once this last one has been calculated. You have also been given the matrices K and M , with the fixed boundary nodes eliminated. Convert the generalized eigenvalue problem to a standard one. What assumptions do you have to make for this? Are they satisfied? Implement an efficient Lanczos tridiagonalization (with orthogonalization of your choosing) that reduces the system matrix of the standard eigenvalue problem associated to the generalized eigenvalue problem above to tridiagonal form. Compute its 5 dominant eigenvalues (i.e. the smallest in absolute value) using the algorithm you wrote in the previous section. What must you do to find the smallest eigenvalues, rather than the largest? Verify using `Matlab`'s `eigs`, calculate and visualise the associated eigenmodes. Report your findings.

3.3 Description of the objects provided

(1) A .mat file containing the sparse matrices K, M and the grid, represented by a double array `[nodes, elements, actualDofs, surf]`. Here actualDofs represents a subsets of the degrees of freedom that are free to move (i.e. the ones actually considered in the eigenvalue problem). What value does the deformation field have at the dofs that aren't free?. The array surf is a one-dimensional representation of the cantilever.

(2) Plotting routines `plot_afm_mesh`, `plot_afm_surf` and `plot_afm_frame`. The first plots the full mesh, whereas the second and third one take as input the nodes plus a deformation and respectively plot this as a frame and as a surface, but only for a one-dimensional representation of the cantilever, for clarity of presentation. These can then be overlayed to illustrate the deformation. Tip: amplify the deformations found by the eigenvalue solver by some factor to increase visibility.

Your code should look something like:

```
load('HW1.mat');

figure(1)

plot_afm_mesh(nodes,elements); %plot the AFM tip

...%code to calculate eigenvectors

dU = zeros(848*3,1);

dU(actualDofs) = V(:,k);% k the index of eigenvalue

dU = reshape(dU,[],3);

displaced_nodes = nodes+dU*200;

displaced_nodes = nodes+dU*200;

figure(2)

plot_afm_surf(nodes,elements,...)  %choose fig.  opts.

hold on
```

```
    plot_afm_frame(displaced_nodes,elements,...)  %choose fig.  opts.
```

Ofcourse you are free to choose a different visualisation, so long as it is equally clear.

4 Quotation and questions

Points are awarded based on the correctness of your results and the quality of your code (40%), the insight demonstrated in your report (40%) and the organization and presentation of your report (20%). Don't hesitate to email me at simon.dirckx@kuleuven.be if anything is unclear or if you suspect something is wrong. If you are stuck on something you can always ask for a hint. This will be taken into account in the evaluation but will not severely impact your grade.