



Solution for Project 1

Due date: Wednesday, 12 October 2022, 23:59 AM

Numerical Computing 2022 — Submission Instructions

(Please, notice that following instructions are mandatory:
submissions that don't comply with, won't be considered)

- Assignments must be submitted to iCorsi (i.e. in electronic format).
- Provide both executable package and sources (e.g. C/C++ files, Julia). If you are using libraries, please add them in the file. Sources must be organized in directories called:
Project_number_lastname_firstname
and the file must be called:
project_number_lastname_firstname.zip
project_number_lastname_firstname.pdf
- The TAs will grade your project by reviewing your project write-up, and looking at the implementation you attempted, and benchmarking your code's performance.
- You are allowed to discuss all questions with anyone you like; however: (i) your submission must list anyone you discussed problems with and (ii) you must write up your submission independently.

The purpose of this assignment¹ is to learn the importance of numerical linear algebra algorithms to solve fundamental linear algebra problems that occur in search engines.

PageRank Algorithm

1. Theory [20 points]

(a) What are an eigenvector, an eigenvalue and an eigenbasis?

An eigenvector is a nonzero vector that, when involved in a linear transformation gets stretched and scaled but stays on the same span.

An eigenvalue, denoted with symbol λ represents the factor of how much the eigenvector gets stretched along his direction.

The eigenvalue can be negative, as it means the resulting vectors inverts its direction after the linear transformation.

An eigenbasis is a diagonal matrix that has eigenvectors as its columns, and eigenvalues along its diagonal.

Given a matrix A and a linear transformation T , if there exists a vector v such that $Tv = \lambda v$, then v is an eigenvector of A and λ is an eigenvalue of A . If there exists a set of vectors V such that $T(V) = \lambda V$, then V is an eigenbasis of A .

¹This document is originally based on a SIAM book chapter from *Numerical Computing with Matlab* from Clever B. Moler.

(b) **What assumptions should be made to guarantee convergence of the power method?**

We need to assume that the eigenvalue to which the power method converges is the dominant eigenvalue, and we also need to assume that the randomly-chosen initial vector has a component in the same direction as the eigenvector.

Also, to guarantee a faster convergence, λ_1 and λ_2 have to be distant, as the asymptotic error constant is $|\frac{\lambda_1}{\lambda_2}|$ meaning convergence is extremely slow if the two eigenvalues are close to each other.

(c) **What is the shift and invert approach?**

As mentioned previously, if the eigenvalues are close to each other convergence can be extremely slow.

The inverse iteration uses the shift and invert technique to speed up the convergence.

If the eigenvalues of A are λ_j , the eigenvalues of $A - \alpha I$ are $\lambda_j - \alpha$, and the eigenvalues of $B = (A - \alpha I)^{-1}$ are $\mu_j = \frac{1}{\lambda_j - \alpha}$

If we apply the power method to B we get an improved rate of $|\frac{\mu_2}{\mu_1}| = |\frac{\frac{1}{\lambda_2 - \alpha}}{\frac{1}{\lambda_1 - \alpha}}| = |\frac{\lambda_1 - \alpha}{\lambda_2 - \alpha}|$.

It is therefore in our best interests to find an α as close as possible to any eigenvalue.

The shift and invert approach is to choose α such that the improved rate is as small as possible.

(d) **What is the difference in cost of a single iteration of the power method, compared to the inverse iteration?**

Using the power method, each iteration involves a *matrix-vector multiplication*, while the *inverse iteration* requires solving a linear system. Solving a linear system is way more computationally expensive than a matrix-vector multiplication, having a fast convergence for inverse iteration is necessary in order for it to be effective; this would be impossible when tackling larger issues such as Internet searching. Inverse iteration must be used for smaller problems or problems with special structure that enables fast direct methods.

(e) **What is a Rayleigh quotient and how can it be used for eigenvalue computations?**

The Rayleigh quotient is an improvement over the inverse iteration method for finding eigenvalues.

It guarantees very fast convergence, in most cases even cubically. To guarantee fast convergence, we change the value of α dynamically to be the Rayleigh quotient in each iteration.

Even though Rayleigh quotient iteration is more computationally expensive as it requires to refactor the matrix in each iteration, the cubic convergence makes it worthwhile.

2. Other webgraphs [10 points]

3. Connectivity matrix and subcliques [5 points]

4. Connectivity matrix and disjoint subgraphs [10 points]

5. PageRanks by solving a sparse linear system [40 points]

6. Quality of the Report [15 points]