Università della Svizzera italiana | Institute of Computational Science ICS

**Numerical Computing** **2020**

Student: Claudio Maggioni Discussed with: –

# Solution for Project 5

Due date: Wednesday, December 02, 2020, 11:59 PM

---

**Numerical Computing 2020 — 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, Matlab). 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 assignment is to gain insight on the theoretical and numerical properties of the Conjugate Gradient method. Here we use this method in an image processing application with the goal of deblur an image given the exact (noise-free) blurred image and the original transformation matrix. Note that the "noise-free" simplification is essential for us to solve this problem in the scope of this assignment.

## 1. General Questions [10 points]

### 1.1. What is the size of the matrix $A$?

$A$ is an $n^2$ by $n^2$ matrix, where $n$ is the width and height in pixels of the image to transform.

### 1.2. How many diagonals bands does $A$ have?

$A$ as $d^2$ diagonal bands, where $d$ is strictly an order of magnitude below $n$.

### 1.3. What is the length of the vectorized blur image $b$?

$b$ is the row-vectorized form of the image pixel matrix, and thus has dimensions 1 by $n^2$.

## 2. Properties of A [10 points]

### 2.1. If $A$ is not symmetric, how would this affect $\tilde{A}$?

If $A$ were not symmetric, then $\tilde{A}$ would not be positive definite since by definition $\tilde{A} = AA^T$, thus not satifying the assumptions taken when solving the system.

### 2.2. Explain why solving $Ax = b$ for $x$ is equivalent to minimizing $x^T Ax - b^T x$ over $x$.

First, we can say that:

$$f(x) = \frac{1}{2} x^T Ax - b^T x = \frac{1}{2} \langle Ax, x \rangle - \langle b, x \rangle$$

Then by taking the derivative of $f(x)$ w.r.t. $x$ we have (assuming $A$ is *spd*, which it is):

$$f'(x) = \frac{1}{2} A^T x + \frac{1}{2} Ax - b = Ax - b$$

which for $f'(x) = 0$ will be equivalent to solving $Ax = b$. By taking the second derivative we have:

$$f''(x) = A > 0$$

since $A$ is positive definite. Therefore, we can say that the absolute minima of $f(x)$ is the solution for $Ax = b$.

## 3. Conjugate Gradient [40 points]

### 3.1. Write a function for the conjugate gradient solver `[x,rvec]=myCG(A,b,x0,max_itr,tol)`, where `x` and `rvec` are, respectively, the solution value and a vector containing the residual at every iteration.

The implementation can be found in file `myCG.m` in the source directory. The test code for the function `myCG` can be found in the `test.m` file.

### 3.2. In order to validate your implementation, solve the system defined by `A_test.mat` and `b_test.mat`. Plot the convergence (residual vs iteration).

The plot of the squared residual 2-norms over all iterations can be found in Figure **??**.

### 3.3. Plot the eigenvalues of `A_test.mat` and comment on the condition number and convergence rate.

The eigenvalues of A can be found in figure **??**. The condition number for matrix $A$ according to `rcond(...)` is $\approx 3.2720 \cdot 10^7$, which is very low without sitting in the denormalized range (i.e. $<$ eps) and thus very good for the Conjugate Gradient algorithm. This well conditioning is also reflected in the eigenvalue plot, which shows a not so drastic increase of the first eigenvalues ordered in increasing order.

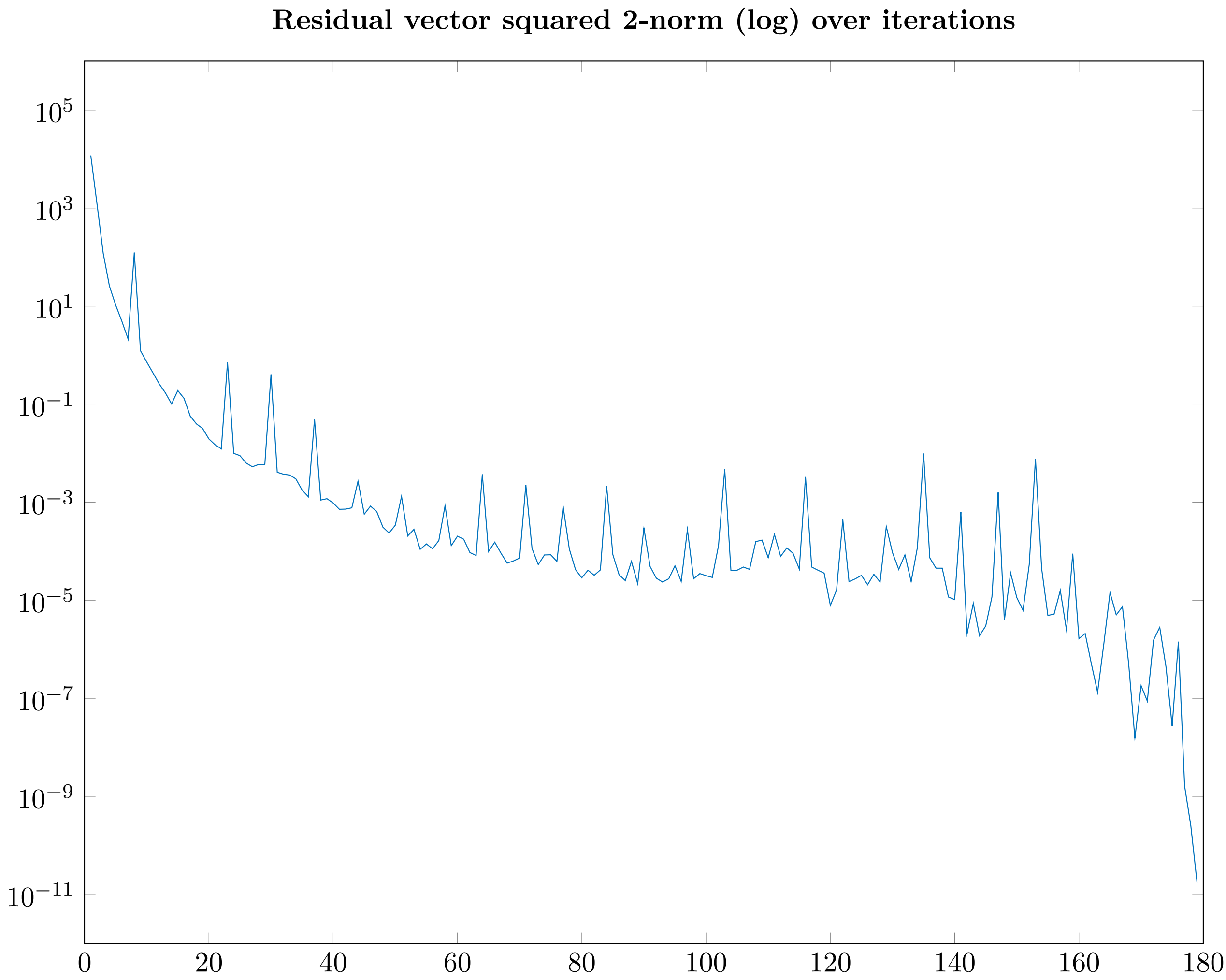


Figure 1: Semilog plot of the plot of the squared residual 2-norms over all iterations

## 4. Debluring problem [40 points]

**4.1. Solve the debluring problem for the blurred image matrix `B.mat` and transformation matrix `A.mat` using your routine `myCG` and Matlab's preconditioned conjugate gradient `pcg`. As a preconditioner, use `ichol` to get the incomplete Cholesky factors and set routine type to `nofill` with $\alpha = 0.01$ for the diagonal shift (see Matlab documentation). Solve the system with both solvers using $max\_iter = 200$ $tol = 10^{-6}$. Plot the convergence (residual vs iteration) of each solver and display the original and final deblurred image.**

**4.2. When would `pcg` be worth the added computational cost? What about if you are debluring lots of images with the same blur operator?**

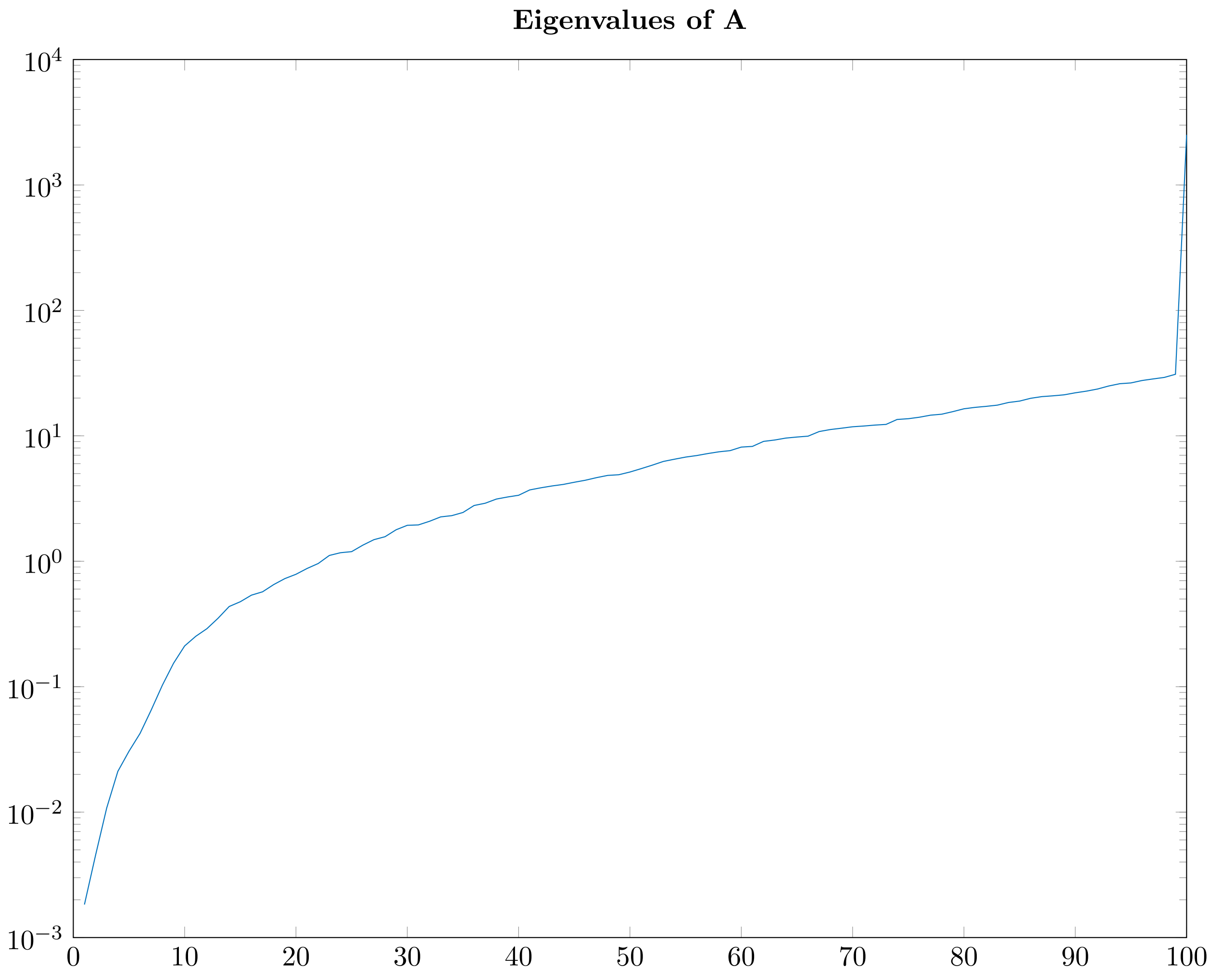


Figure 2: Semilog plot of the eigenvalues of A