Convergence speed of conjugate gradient method applied to a matrix with clustered eigenvalues

Mele Giampaolo

October 5, 2012

Abstract

This document is the dissertation for the seminars of the professors Bertaccini and Filippone during the Rome Moscow school. The aim is to understand how the conjugate gradient method works with a matrix that have clustered eigenvalues (in some sense that we will try to define). The target is not to prove teorical result but just starting from numerical experiments try to understand and guess what happend.

How to construct random symmetric definite positive matrices with fixed eigenvalues

How we explain in the abstract we want to do some numericals experiments, so we need some test matrices to do that, we know that the conjugate gradient works with symmetric definite positive matrices, so we need a tool to make them. We also want to fix the eigenvalues of those matrices, so let $\lambda_1, \ldots, \lambda_n$ the numbers that we want as eigenvalues, the idea to make A is easy:

- Choose randomly the vectors $v_1, \ldots, v_n \in \mathbb{R}^n$
- Ortonormalize those vectors
- Let define

$$A = \sum_{k=1}^{n} \lambda_i v_i v_i^T$$

It is now clear that A has eigenvalues $\lambda_1, \ldots, \lambda_n$ and is symmetric.

Listing 1: Make a symmetric matrix with fixed eigenvalues

```
function A = mkmatrix(eigval)

n=size(eigval,2);

for i=1:1:n
    v(:,i)=rand(n,1);
end

% orth is a matlab function that make orthornormal
% the columns of v
v=orth(v);

A=zeros(n,n);

for i=1:1:n
    A=A+eigval(i)*v(:,i)*v(:,i)';
end

end
```

Now we have the tool to make our experiments.

Conjugate gradient

We recall fastly the algorithm of conjugate gradient. The aim is to solve

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \quad \text{with } \boldsymbol{A} \in \mathbb{R}^{n\times n} \text{ symmetric and definite positive}$$

The algorithm is the following

$$\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$$

$$\boldsymbol{p}_0 = \boldsymbol{r}_0$$

$$k = 0, \ldots, n$$

$$\alpha_k = \frac{\boldsymbol{r}_k^T \boldsymbol{r}_k}{\boldsymbol{p}_k^T \boldsymbol{A} \boldsymbol{p}_k}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k$$

$$\text{if } \boldsymbol{r}_{k+1} \text{ is small then } k = n$$

$$\beta_k = \frac{\boldsymbol{r}_{k+1}^T \boldsymbol{r}_{k+1}}{\boldsymbol{r}_k^T \boldsymbol{r}_k}$$

$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k$$

Where x_0 it's our initial guess of the solution.

We will use the following matlab code for our experiments

Listing 2: Make a symmetric matrix with fixed eigenvalues

```
function [x] = GC(A,b,x)
    r=b-A*x;
    p=r;
    rsold=r'*r;

    for i=1:1:size(A)
```

```
        Ap=A*p;
        alpha=rsold/(p'*Ap);
        x=x+alpha*p;
        r=r-alpha*Ap;
        rsnew=r'*r;
        if sqrt(rsnew)<1e-10
            i
            break;
        end
        p=r+rsnew/rsold*p;
        rsold=rsnew;
    end
end
```

Let's do some experiments!

Experiment 1

Let we make $A \in \mathbb{R}^{10\times 10}$ using the previous code and let we choose $\lambda_i = 1$ for $i = 1, \ldots, 9$ and $\lambda_{10} = 100$.

$$\begin{pmatrix}
2.7865 & 0.6234 & 5.9095 & -5.7171 & -2.9765 & -3.4235 & -3.9482 & 0.1147 & 7.8497 & -2.8068 \\
0.6234 & 1.2175 & 2.0622 & -1.9950 & -1.0387 & -1.1947 & -1.3777 & 0.0400 & 2.7392 & -0.9794 \\
5.9095 & 2.0622 & 20.5479 & -18.9114 & -9.8457 & -11.3244 & -13.0600 & 0.3793 & 25.9657 & -9.2844 \\
-5.7171 & -1.9950 & -18.9114 & 19.2956 & 9.5251 & 10.9557 & 12.6348 & -0.3669 & -25.1202 & 8.9821 \\
-2.9765 & -1.0387 & -9.8457 & 9.5251 & 5.9590 & 5.7038 & 6.5780 & -0.1910 & -13.0782 & 4.6763 \\
-3.4235 & -1.1947 & -11.3244 & 10.9557 & 5.7038 & 7.5604 & 7.5659 & -0.2197 & -15.0424 & 5.3786 \\
-3.9482 & -1.3777 & -13.0600 & 12.6348 & 6.5780 & 7.5659 & 9.7254 & -0.2534 & -17.3477 & 6.2030 \\
0.1147 & 0.0400 & 0.3793 & -0.3669 & -0.1910 & -0.2197 & -0.2534 & 1.0074 & 0.5038 & -0.1801 \\
7.8497 & 2.7392 & 25.9657 & -25.1202 & -13.0782 & -15.0424 & -17.3477 & 0.5038 & 35.4905 & -12.3326 \\
-2.8068 & -0.9794 & -9.2844 & 8.9821 & 4.6763 & 5.3786 & 6.2030 & -0.1801 & -12.3326 & 5.4097
\end{pmatrix}$$

Then the algorithm converge in 2 steps.

Experiment 2

Let now consider a different spectrum, (until now we have not defined what we mean with clustered eigenvalues). Let we consider $\boldsymbol{\varepsilon}$ such that $0 < \varepsilon_i \leq 10^{-1}$ and than let we take $\lambda_i = 1 + \varepsilon_i$ and $\lambda_{10} = 100 + \varepsilon_{10}$, then

$$\begin{pmatrix}
3.2351 & 6.3638 & -3.2231 & -0.2583 & 4.3169 & -2.5127 & -5.7615 & 2.0180 & 5.1293 & -8.7633 \\
6.3638 & 19.1193 & -9.1769 & -0.7355 & 12.2913 & -7.1543 & -16.4046 & 5.7456 & 14.6045 & -24.9514 \\
-3.2231 & -9.1769 & 5.6479 & 0.3725 & -6.2252 & 3.6235 & 8.3085 & -2.9100 & -7.3968 & 12.6372 \\
-0.2583 & -0.7355 & 0.3725 & 1.0299 & -0.4989 & 0.2904 & 0.6659 & -0.2332 & -0.5928 & 1.0128 \\
4.3169 & 12.2913 & -6.2252 & -0.4989 & 9.3379 & -4.8532 & -11.1281 & 3.8976 & 9.9071 & -16.9259 \\
-2.5127 & -7.1543 & 3.6235 & 0.2904 & -4.8532 & 3.8248 & 6.4773 & -2.2686 & -5.7665 & 9.8519 \\
-5.7615 & -16.4046 & 8.3085 & 0.6659 & -11.1281 & 6.4773 & 15.8521 & -5.2019 & -13.2224 & 22.5901 \\
2.0180 & 5.7456 & -2.9100 & -0.2332 & 3.8976 & -2.2686 & -5.2019 & 2.8219 & 4.6311 & -7.9121 \\
5.1293 & 14.6045 & -7.3968 & -0.5928 & 9.9071 & -5.7665 & -13.2224 & 4.6311 & 12.7715 & -20.1113 \\
-8.7633 & -24.9514 & 12.6372 & 1.0128 & -16.9259 & 9.8519 & 22.5901 & -7.9121 & -20.1113 & 35.3596
\end{pmatrix}$$

Then the algorithm converge in 2 steps.

Experiment 3 (Let's do it seriously!)

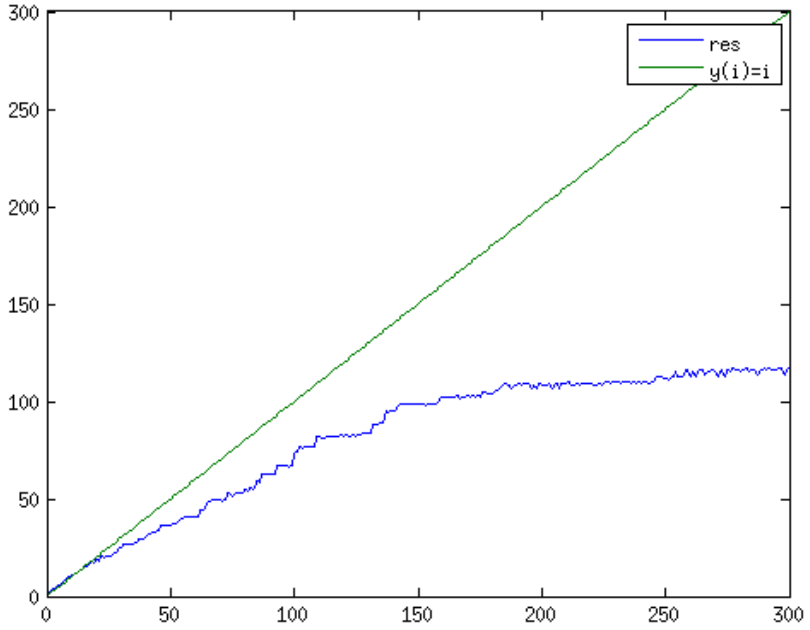
Now let we take $A \in \mathbb{R}^{1000\times 1000}$, let choose the eigenvalues all equals to 1 but 5 of them bigger. Of course we can put there the matrix but the algorithm converge in 5 steps.

Experiment 4

Now again let's take $A \in \mathbb{R}^{1000\times 1000}$,if λ_i are the eigenvalues of the previous example, let $\boldsymbol{\varepsilon}$ such that $0 < \varepsilon_i < 10^{-6}$, we want that the new eigenvalues are $\lambda_i + \varepsilon_i$, so all the new eigenvalues are $1 + \varepsilon_i$ except 5 or them that are bigger. Then the algorithm converge in 6 steps.

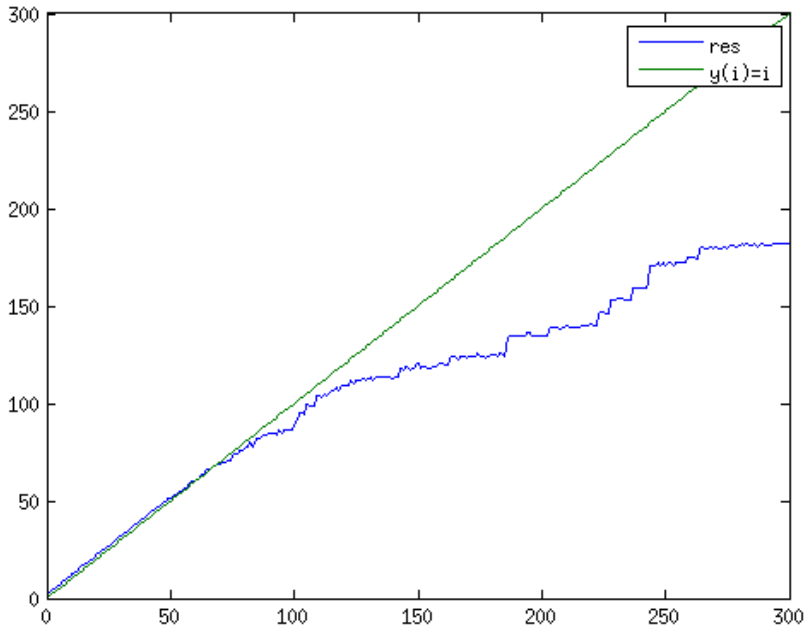
Experiment 5 (How the steps grow with the lost of clustering with trivial cluster)

Let $A_i \in \mathbb{R}^{300\times 300}$ with $300-i$ eigenvalues equal to 500 and i eigenvalues random numbers between 1 and 1000. So we want understand how many steps we need to solve $\boldsymbol{A}_i\boldsymbol{x} = \boldsymbol{b}$, we will call the numbers of steps $s(i)$. The result of the experiment show that $s(i) \leq i$, the following is the plot of $s(i)$.



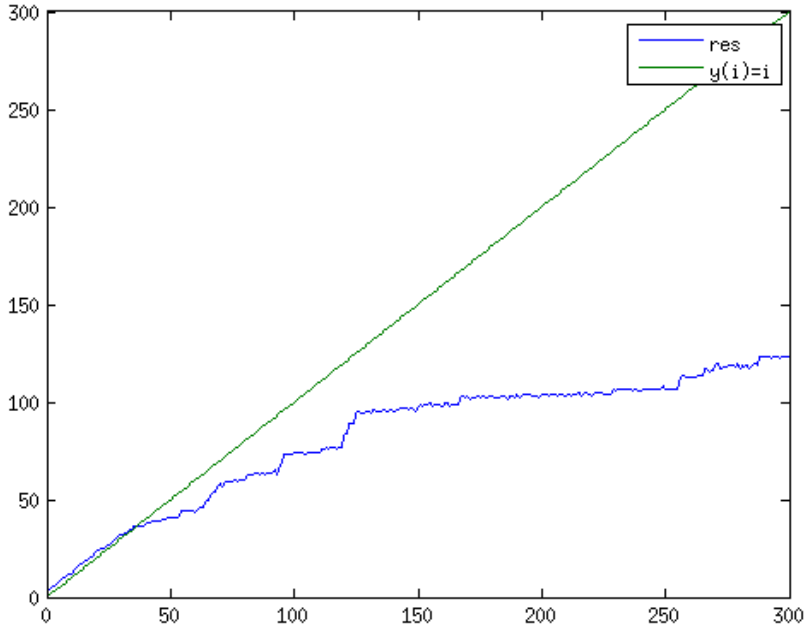
Experiment 6 (How the steps grow with the lost of clustering and with not trivial cluster)

Let's take the previous example with a perturbation of the eigenvalues to create a not trivial cluster, so if λ_i are the old eigenvalues now we consider $\lambda_i + \varepsilon_i$ with $0 < \varepsilon_i < 10^{-6}$ and we do the same computation, the result is very near to the previous but $s(i) \leq i$ is it true for i bigger than a fixed number.

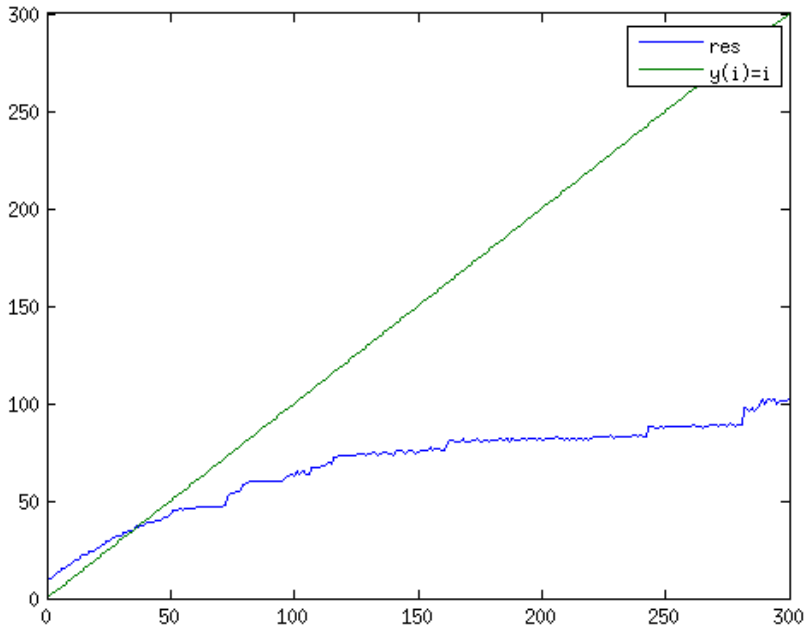


Experiment 6 (How the steps grow with the lost of clustering and with not trivial and larger cluster)

If we take $0 < \varepsilon_i < 10^{-2}$ we obtain

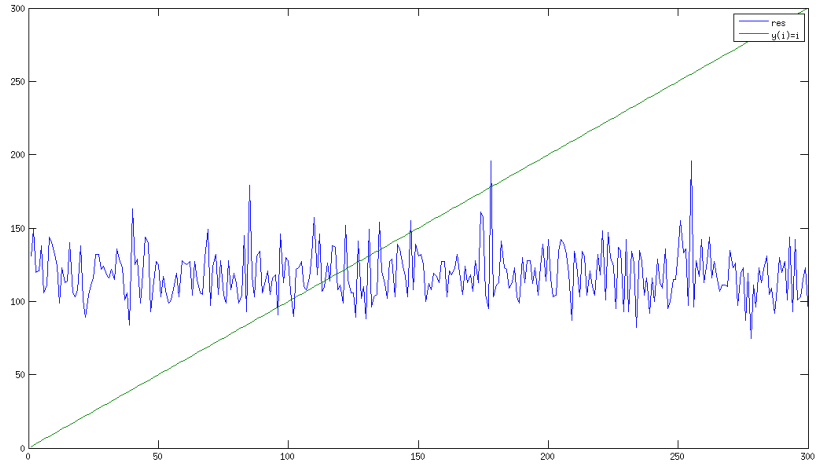


If we take $0 < \varepsilon_i < 50$ we obtain



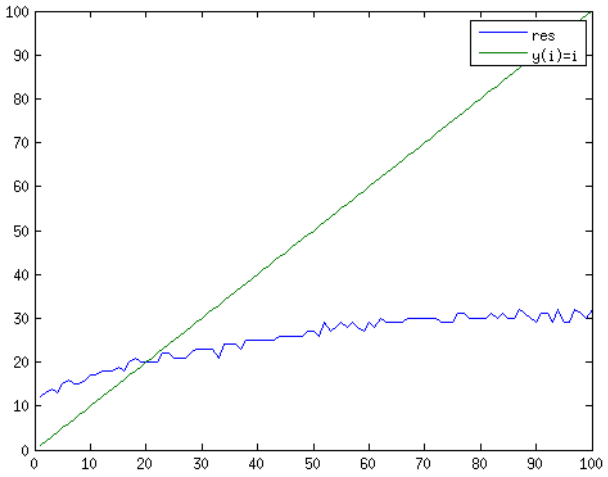
Experiment 7 (Just to understand why we are lucky with the cluster)

If we consider for istance $A_i \in \mathbb{R}^{300\times 300}$ where there are no cluster, so we choose the eigenvalues between 1 and 100, then we obtain



Experiment 8 (How the steps grow with increasing cluster size)

Let now consider $A \in \mathbb{R}^{100\times 100}$ with eigenvalues $\lambda_i = 10 + \varepsilon_i$ for $i = 1, \ldots, 95$ and λ_i numbers between 1 and 1000 for $i = 95, \ldots, 100$ and $\varepsilon_j = h_j(10 + j)/10$ (h_j is a random number between 0 and 1). The following is the plot of the numer of steps $s(i)$ (now the numer of eigenvalues in the cluster is fixed but the size of cluster is the new variable).



Notation

Fixed a matrix $A \in \mathbb{R}^{n \times n}$ and the λ_i eigenvalues, if we have $\lambda_i \in [a - \varepsilon, a + \varepsilon]$ for $i = 1, \ldots, m$, then the speed of convergence is a function $s(\varepsilon, m)$.

Experiment 9 (plot of s)

We choose $A \in \mathbb{R}^{100 \times 100}$, $a = 25$ and

$$\lambda_i = \begin{cases} 25 + 100i \cdot h_i & 1 \leq i \leq m \\ 25 + \varepsilon_i & \text{otherwise} \end{cases}$$

Where h_i is a random number between 0 and 1. We choose

$$\varepsilon_i = i \frac{\max\limits_{1 \leq i \leq m} \lambda_i - 25}{n}$$

The following is the plot of $s(\varepsilon_i, m)$ with $i = 1, \ldots, n$ and $m = 1, \ldots, n$