report on project 2

Ekaterina Ilin and Isabelle Gauger
GitHub: https://github.com/CPekaterina/project-no2

September 22, 2014

Contents

1 questions a)

We look at the equation

$$|t| = \left|-\tau \pm \sqrt{1+\tau^2}\right| \tag{1}$$

and consider the three cases $\tau = 0$, $\tau > 0$ and $\tau < 0$. For $\tau = 0$ we get

$$|t| = |\pm 1| = 1 \tag{2}$$

For $\tau > 0$ the absolute value of t is smaller for the solution with plus.

$$|t| = \left|-\tau + \sqrt{1+\tau^2}\right| \tag{3}$$

Now we look at $\tau \to \infty$:

$$\lim_{\tau\to\infty} |t| = \lim_{\tau\to\infty} \left|-\tau + \sqrt{1+\tau^2}\right| = 0 \tag{4}$$

For $\tau < 0$ the absolute value of t is smaller for the solution with minus.

$$|t| = \left|-\tau - \sqrt{1+\tau^2}\right| \tag{5}$$

So now we look at $\tau \to -\infty$:

$$\lim_{\tau\to-\infty} |t| = \lim_{\tau\to-\infty} \left|-\tau - \sqrt{1+\tau^2}\right| = 0 \tag{6}$$

We have seen that for $\tau = 0$ the absolute value of t is one and if we increase or decrease τ it approachs zero.

$$|\tan\theta| \leq 1 \text{for} \, |\theta| \leq \frac{\pi}{4} \tag{7}$$

So if we choose t to be the smaller of the roots $|\theta| \leq \frac{\pi}{4}$ what is minimizing the difference between the matrices A and B. This can be seen if we look at the given equation

$$||\mathbf{B} - \mathbf{A}||_F^2 = 4(1-c) \sum_{i=1, i\neq k,l}^{n} (a_{ik}^2 + a_{il}^2) + \frac{2a_{kl}^2}{c^2} \tag{8}$$

$(1-c)$ at the beginning of the equation becomes zero when $\cos\theta = 1$ and than the total first part of the equation is zero. Also the second part of the equation reaches its minimum value for $\cos\theta = 1$. For $|\theta| \leq \frac{\pi}{4}$ the value of $\cos\theta$ is between 1 and ≈ 0.7 so the difference between the matrices A and B we get is near the minimum. This means that the non-diagonal matrix elements of A are nearly zero, what is what we want to achieve.

2 Estimation of the execution time

To estimate the number N of similarity transformations performed we assume that the algorithm needs roughly $O(n^2)$ of the latter simply because each transformation sets a non-diagonal element to zero. The algorithm converges as shown above but in general we still may obtain up to $n-2$ new non-zero, non-diagonal matrix elements after every transformation. Therefore this assumption has to be verified numerically. Up to this point we have not yet considered the dependence of the convergence rate on the precision ϵ of the search for the

Table 1: Number N of similarity transformations performed in the Jacobi algorithm with respect to the dimensionality n of the matrix. The ratio N/n^2 suggests a behaviour that can be written as $N(n) \approx a \cdot n^2$ with a ranging in the order of 10^0. More about the behaviour of a in figure **??**.

n	N	$a = N/n^2$
10	94	0.94
25	811	1.29
50	3542	1.42
100	14900	1.49
500	386898	1.55
1000	$> 10^6$	-

maximal non-diagonal element.
In the following discussion we set $\epsilon = 10^{-12}$ and $\rho_{max} = 5$ and vary the dimensionality of the matrix in order to estimate the program's behaviour with respect to n.
In table **??** the number of similarity transformations is listed against the dimensionality n of the matrix for some significant numbers. The range of n is limited upwards due to the growth of the execution time, which goes with n^3. For $n = 1000$ more than a million transformations are required and because each transformation takes $O(n)$ time the running time of the program falls outside of tolerance for greater n.
From this table we conclude that the assumption made in the beginning is correct if we add a parameter a to the function

$$N(n) = a \cdot n^2 \tag{9}$$

Let us now have a closer look at this parameter and assume a dependence on n, thus leading to

$$N(n) = a(n) \cdot n^2 \tag{10}$$

If we now plot a against n we can approximate its behaviour to a exponential decay function as shown in figure **??**. We note that in this model a approaches the saturation limit for n of the order of 10^2 so that we can lean back and perform the algorithm for greater n knowing that a should not change significantly. But, however, more computational power is needed to verify this theory.

Figure 1: The behaviour of the parameter a in the $N(n)$ function that describes the number of similarity operations on the matrix can be approximated with an exponential function as shown in this figure.