Numerieke Modellering en Benadering
Taak 1: Numerieke Lineaire Algebra

Thibault Lahaye - r0713047, Ward Kerkhofs - r0750166

2021-2022

1 Gram-Schmidt

1. Implementeer deze algoritmes, met als signatuur See MATLAB file.

2. Leg uit waarom deze drie algoritmes wiskundig equivalent zijn. In classical Gram-Schmidt the orthogonalization step consists of only one projection whereas Modified Gram-Schmidt uses a projection for every previous q vector. These projections are equivalent according to formula (1).

$$P_j = P_{\perp q_{j-1}} ... P_{\perp q_2} P_{\perp q_1} \tag{1}$$

In Repeated Gram-Schmidt the same projection as in Regular Gram-Schmidt is applied twice. According to the idempotent property (2) of projectors, this is equivalent.

$$P_j P_j = P_j \tag{2}$$

3.Geef een uitdrukking voor de kolom a_j van de matrix A in functie van de kolommen van Q

$$a_j = 2^0 * q_j + 2^{1-j} * q_j \tag{3}$$

4. Is deze stijging logaritmisch, polynomiaal of exponentieel? Waarom? There is an exponential decrease of the diagonal elements of D. The vectors of A get exponentially closer to each other. This produces an exponential increase in absolute error, due to subtracting increasingly equal vectors.

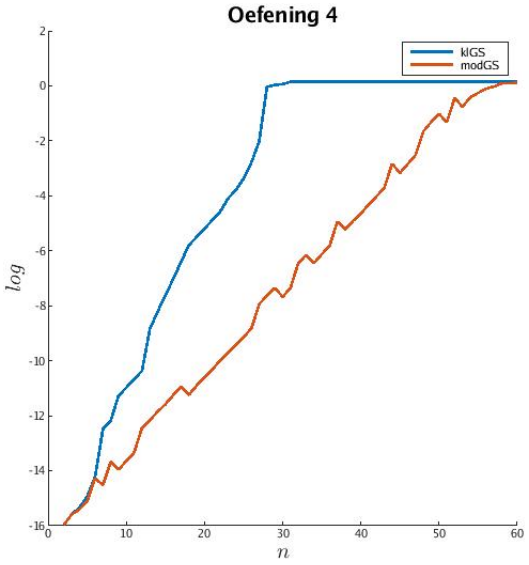

Figure 1: Exercise 4: graph of the experimental error

5.1. Voer de foutenanalyse uit voor de klassieke GS, en n = 3. Schrijf je bewijs netjes neer. In the first step we normalize vector a_1, since no dot products are used, no error is introduced.

$$\tilde{q_1} = q_1 \tag{4}$$

In the following steps we assume we calculate the exact value of the dot product, apart from a relative error ϵ. When normalizing, we assume the first factor to be much greater than the following factors, which allows us to simplify the normalization. This assumption holds true when n is not too large.

$$\tilde{a_2} \approx a_2 - fl(\langle a_2, \tilde{q_1}\rangle)\tilde{q_1} \approx q_1 + \frac{q_2}{2} - (1+\varepsilon_1)q_1 \approx \frac{q_2}{2} - \varepsilon_1 q_1 \tag{5}$$

$$\tilde{q_2} \approx q_2 - 2\varepsilon_1 q_1 \tag{6}$$

In the following step, there will be factors that are of $\mathcal{O}(\epsilon^2)$, since these are negligible, we will not write them down.

$$a_{\tilde{3},1} \approx a_3 - fl(\langle a_3, \tilde{q_1}\rangle)\tilde{q_1} \approx q_1 + \frac{q_3}{4} - (1+\varepsilon_2)q_1 \approx \frac{q_3}{4} - \varepsilon_2 q_1 \tag{7}$$

$$a_{\tilde{3},2} \approx a_{\tilde{3},1} - fl(\langle a_3, \tilde{q_2}\rangle)\tilde{q_2} \approx a_{\tilde{3},1} - fl\left\langle q_1 + \frac{q_3}{4}, q_2 - 2\varepsilon_2 q_1 \right\rangle \tilde{q_2} \approx \frac{q_3}{4} - \varepsilon_2 q_1 + 2\varepsilon_2 q_2 \tag{8}$$

$$\tilde{q_3} \approx q_3 - 4\varepsilon_2 q_1 + 8\varepsilon_2 q_2 \tag{9}$$

5.2. Veronderstel n = 4. Geef, opnieuw voor klassieke GS, een uitdrukking voor $\tilde{q_4}$ (zonder bewijs); wat is de fout op q_4? Voor n algemeen, maak een schatting voor de fout op q_n.

$$\tilde{q_4} \approx q_4 - 8\varepsilon_3 q_1 + 16\varepsilon_3 q_2 + 32\varepsilon_3 q_3 \tag{10}$$

$$\tilde{q_n} \approx q_n - 2^{n-1}\varepsilon_n q_1 + 2^{n-1}\varepsilon_n \sum_{i=2}^{n-1}(2^{i-1}q_i) \tag{11}$$

$$e_n \approx -2^{n-1}\varepsilon_n q_1 + 2^{n-1}\varepsilon_n \sum_{i=2}^{n-1}(2^{i-1}q_i) \tag{12}$$

5.3. Voer de foutenanalyse uit voor de aangepaste GS, en n algemeen, maar voldoende klein.

$$\tilde{a_{n,1}} \approx a_n - fl(\langle a_n, \tilde{q_1}\rangle)\tilde{q_1} \approx q_1 + \frac{q_n}{2^{n-1}} - (1+\varepsilon_n)q_1 \approx \frac{q_n}{2^{n-1}} - \varepsilon_n q_1 \tag{13}$$

$$\tilde{a_{n,2}} \approx \tilde{a_{n,1}} - fl(\langle \tilde{a_{n,1}}, \tilde{q_2}\rangle)\tilde{q_2} \approx \frac{q_n}{2^{n-1}} - \varepsilon_n q_1 + \mathcal{O}(\varepsilon_n^2) \approx \frac{q_n}{2^{n-1}} - \varepsilon_n q_1 \tag{14}$$

$$... \tag{15}$$

$$\tilde{a_{n,n-1}} \approx \tilde{a_{n,n-2}} - fl(\langle \tilde{a_{n,n-2}}, \tilde{q_{n-1}}\rangle)\tilde{q_{n-1}} \approx \frac{q_n}{2^{n-1}} - \varepsilon_n q_1 + \mathcal{O}(\varepsilon_n^2) \approx \frac{q_n}{2^{n-1}} - \varepsilon_n q_1 \tag{16}$$

So we find the general expressions for $\tilde{a_n}$, $\tilde{q_n}$ and e_n :

$$\tilde{a_n} \approx \frac{q_n}{2^{n-1}} - \varepsilon_n q_1 \tag{17}$$

$$\tilde{q_n} \approx q_n - 2^{n-1}\varepsilon_n q_1 \tag{18}$$

$$e_n \approx -2^{n-1}\varepsilon_n q_1 \tag{19}$$

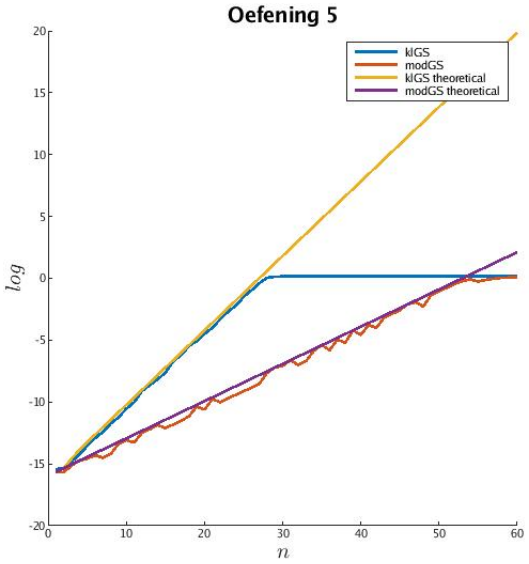

Figure 2: Exercise 5: graph of the experimental and theoretical error

6. Uit de experimentele data halen we dat de methode plots afvlakt en de fout constant blijft. Waarom wordt de fout niet arbitrair groot? At around n=28, the columns qn resemble q1 until machine precision. This causes the error to be zero and thus it wont increase anymore. After this point all the columns of A are exactly the same.

7.1. Welke stap uit de foutenanalyse is niet langer geldig voor grote n? The normalization step. For large n, the error becomes increasingly dominant with respect to the vector q. This means we must include it when normalizing q.

7.2. Wat is q dan wel voor grote n? Lees uit de figuur de uitendelijke waarde van de fout af. Komt deze overeen met wat je theoretisch uitkomt?

$$\tilde{q}_n = \frac{q_n * 2^{1-n} - \epsilon_{mach} * q_1}{\sqrt{2^{2(1-n)} + \epsilon_{mach}^2}} \tag{20}$$

From the graph we get 1.3165 for the error. Theoretically we get an error of 0.9998. This is a little smaller then the measured value, but our calculations do not include the errors from the other operations.

8. Leg uit waarom deze waarde een goede keuze is. Herneem nu de figuur uit opgave 4, en voeg voor beide methodes de schattingen toe. The matrix product can be seen as a combination of the dot products of the column vectors. These dot products give us the errors calculated in the previous exercises. The larger the error, the worse the orthogonality.

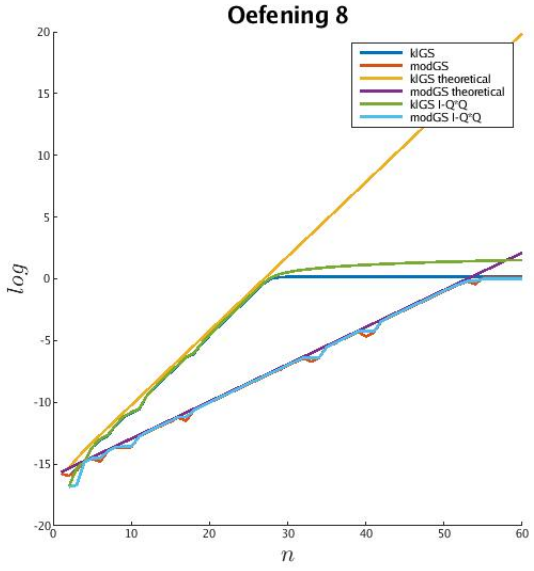

Figure 3: Exercise 8: graph of the orthogonality error

9. Wat is de conditie van deze matrix?

$$\|A\| = \|D\| = 2^k \qquad \left\|A^{-1}\right\| = \left\|D^{-1}\right\| = 2^0 \tag{21}$$

$$\kappa = \|A\| \left\|A^{-1}\right\| = 2^k \tag{22}$$

10. Maak een figuur waar je de fout op de orthogonaliteit plot voor de drie algoritmen, ten opzichte van de waarde k op de x-as. Wat is r? Increasing the value of k with 1, causes us to lose 0.30103 digits (7). Using the given equation we determine that r equals 1.732

$$\log_{10}(2^{k+1}) - \log_{10}(2^k) = \log_{10}(2) = 0.30103 \tag{23}$$

$$\frac{\epsilon_{mach} * 2^{kr}}{\epsilon_{mach} * 2^{(k+1)r}} = 2^{-r} = 0.30103 \tag{24}$$

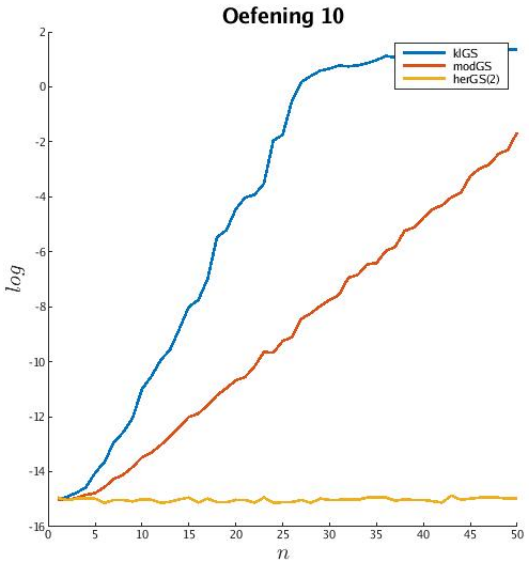

Figure 4: Exercise 10: graph of the orthogonality error

11.1. Hoe verschilt de adaptieve herhaalde GS van de herhaalde GS? Wat moet je aanpassen om de herhaalde GS te bekomen? The case checks should be removed. The second orthogonalization step should always be done. Case III needs to be removed completely. The set q_j should be removed as well.

11.2. Wat verifieert de voorwaarde van Case I? Wat betekent dit in praktijk? It validates how orthogonal $a_j^{(1)}$ is with respect to $q_1, q_2, ...q_{j-1}$. It does this by measuring the change in norm after orthogonalizing the initial vector. If the new vector has become to small, a lot of error was introduced.

11.3. Wat is de significatie van κ? Wat is in praktijk het verschil tussen een grote κ en een kleine κ? We compare with the condition number κ as a measure of orthogonality of the newly computed q_j. A large condition number means the problem is prone to error. The larger the condition number, the more often the second orthogonalization will take place.

11.4. Wat is de betekenis van de voorwaarde in Case II? This measures the orthogonality of the second orthogonalization compared to the first one. We expect this ratio to get larger than the ration in case I.

11.5. Wat is de betkenis van de voorwaarde in Vase III? This means that the first and second orthogonalization have not met our criteria.

11.6. Wat is de significantie van q=0 in Case III? By setting $q_j = 0$ prevent it form introducing errors to this and the following q_{j+1}.... This prevents further loss of orthogonality in the base at the cost of a reduction in rank.

11.7. Leg uit hoe dit algoritme werkt. To sum up the Adaptive Gram-Schmidt algorithm. It works on the same principle as the Repeated Gram-Schmidt when the first orthogonalization attempt isn't precise enough. If the second attempt still isn't orthogonal enough the whole vector is discarded to prevent propagation the error.

12.1. Leg kort uit wat deze stelling aantoont Part one of Stelling 1 tells us the absolute error has an upper bound proportional to the condition number and the norm of a_2. The second part tells us the dot product is inversely proportional to the condition number. This functions as a criteria for orthogonality.

12.2. Bewijs deze stelling voor Case I en Case II

Case I, Deel 1

$$\|e\| \leq \varepsilon \|a_2\| = \varepsilon \left\| p + \frac{a_1}{\|a_1\|^2} a_1^* a_2 \right\| \leq \varepsilon \|p + a_2\| \tag{25}$$

Here we used the Hölder inequality and the given equations. In the following step we neglect e since $\epsilon * e$ is very small.

$$\varepsilon \|p + a_2\| = \varepsilon \|x - e + a_2\| \leq \varepsilon \|x + a_2\| \leq \varepsilon \|\kappa a_2 + a_2\| = (\kappa + 1)\varepsilon \|a_2\| \tag{26}$$

Case I, Deel 2

$$\|a_1^* x\| = \left\|a_1^* x^{'}\right\| = \|a_1^* e + a_1^* p\| \tag{27}$$

Since p is orthogonal to a_1, this dot product becomes 0. In the following step we use the Hölder inequality to split up the dot product. The other steps make use of the given equations.

$$\|a_1^* e\| \leq \|a_1\| \|e\| \leq \|a_1\| \epsilon \|a_2\| \leq \|a_1\| \epsilon \frac{\left\|x^{'}\right\|}{\kappa} = \frac{1}{\kappa} \epsilon \|a_1\| \|x\| \tag{28}$$

Case II, Deel 1

$$\left\|e^{''}\right\| \leq \epsilon_{mach} \left\|x^{'}\right\| \leq \epsilon_{mach} \kappa \|a_2\| \tag{29}$$

$$\left\|x^{''} - (x^{'} - \frac{a_1 a_1^* x^{'}}{\|a_1\|^2})\right\| \leq \epsilon_{mach} \kappa \|a_2\| \tag{30}$$

$$\left\|x^{''} - ((p+e) - \frac{a_1 a_1^* (p+e)^{'}}{\|a_1\|^2})\right\| \leq \epsilon_{mach} \kappa \|a_2\| \tag{31}$$

$$\left\|x^{''} - (p + e - \frac{a_1 a_1^* p^{'}}{\|a_1\|^2} - \frac{a_1 a_1^* e^{'}}{\|a_1\|^2})\right\| \leq \epsilon_{mach} \kappa \|a_2\| \tag{32}$$

The dot products equate to zero. In the next step we use the Hölder inequality.

$$\left\|x^{''} - (p+e)\right\| \leq \epsilon_{mach} \kappa \|a_2\| \tag{33}$$

$$\left\|x^{''} - p\right\| \leq \epsilon_{mach} \kappa \|a_2\| + \epsilon_{mach} \|a_2\| \tag{34}$$

Case II, Deel 2

$$\|a_1^* x\| = \left\|a_1^* x^{''}\right\| = \|a_1^* e + a_1^* p\| \tag{35}$$

Since p is orthogonal to a_1, this dot product becomes 0. In the following step we use the Hölder inequality to split up the dot product. The other steps make use of the given equations.

$$\|a_1^* e\| \leq \|a_1\| \|e\| \leq \|a_1\| \epsilon \left\|x^{'}\right\| \leq \|a_1\| \epsilon \frac{\left\|x^{''}\right\|}{\kappa} = \frac{1}{\kappa} \epsilon \|a_1\| \|x\| \tag{36}$$

2 Iterative Methods

1. Bespreek bondig het convergentiegedrag. The outlying value converges first. The other values are about the same distance from the center of the Arnoldi lemniscate. Which explains why they converge at about the same rate. If their had been more outlying eigenvalues, they would have converged inwards.

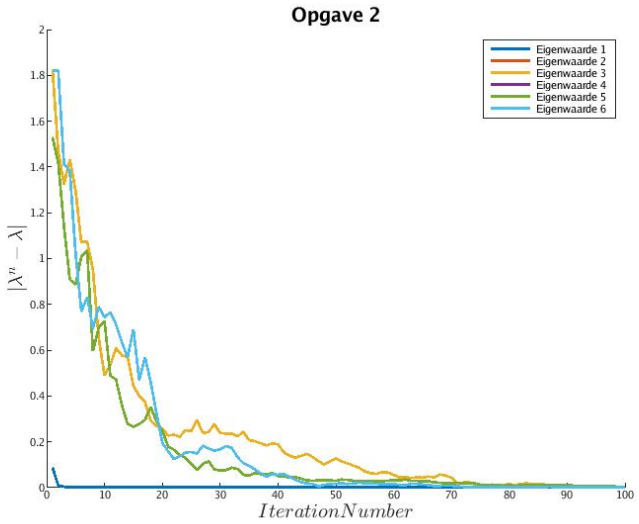

Figure 5: Convergence of Ritz values to the eigenvalues of A

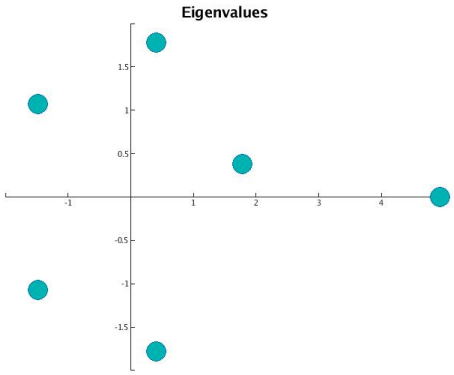

Figure 6: Eigenvalues of A