Verslag bij practicum Numerieke Wiskunde

2024-04-24

Mara Levrau, Simeon Duwel

Opdracht 1

Stelling. Zij $n \in \mathbb{N}_0$ en zijn $A, L, U \in \mathbb{R}^{n \times n}$ matrices. Veronderstel dat L onderdriehoekig is, dat U bovendriehoekig is en dat $\forall i \in \{1, ..., n\} : L_{ii} = 1$. Veronderstel bovendien dat $A = LU$. Dan geldt

$$\forall i \in \{1, ..., n\} : \begin{cases} U_{ik} = A_{ik} - \sum_{j=1}^{i-1} L_{ij} U_{jk} & \forall k \in \{i, ..., n\} \\ L_{ki} = \frac{A_{ki} - \sum_{j=1}^{i-1} L_{kj} U_{ji}}{U_{ii}} & \forall k \in \{i+1, ..., n\} \end{cases} \tag{1}$$

Bewijs. We gebruiken een bewijs via inductie, over de dimensie van A, L en U.

Basisstap: $n = 1$
De kwantor $\forall i \in \{1, ..., n\}$ kan vereenvoudigd worden tot $i = 1$. Zo kunnen we ook $\forall k \in \{i, ..., n\}$ vereenvoudigen tot $k = 1$. Ingevuld leidt dit tot de volgende bewijslast:

$$U_{11} = A_{11} - \sum_{j=1}^{0} L_{1j} U_{j1} \wedge L_{11} = \frac{A_{11} - \sum_{j=1}^{0} L_{1j} U_{j1}}{U_{11}}$$

We gebruiken hier de conventie dat een som die itereert van een hogere naar een lagere index gelijk is aan nul – immers zijn er geen termen in de som, aangezien geen enkel natuurlijk getal j kan voldoen aan $j \geq 1 \wedge j \leq 0$. Dus rest ons te bewijzen dat

$$U_{11} = A_{11} \wedge L_{11} = \frac{A_{11}}{U_{11}}.$$

Gegeven dat alle diagonaalelementen van L gelijk zijn aan 1, weten we dat in het bijzonder $L_{11} = 1$. Aangezien $A = LU$ kunnen we over het element A_{11} zeggen dat het gelijk moet zijn aan $\sum_{j=1}^{1} L_{1j} U_{j1} = L_{11} \cdot U_{11}$. Omdat $L_{11} = 1$, geldt $U_{11} = A_{11}$. Dus is ook $L_{11} = \frac{A_{11}}{U_{11}}$. Hiermee is de basisstap aangetoond.

Inductiestap: als vergelijking 1 geldt voor n, dan geldt ze ook voor $n + 1$.
Zij $A \in \mathbb{R}^{(n+1)\times(n+1)}$ de matrix waarvoor we de bewijslast willen aantonen, en zij $A' \in \mathbb{R}^{n \times n} := A[1, 2, ..., n; 1, 2, ..., n]$ de submatrix van A waarvoor vergelijking 1 al geldt. Zijn tenslotte L en U twee matrices met dezelfde afmetingen als A, met L eenheidsonderdriehoekig en U bovendriehoekig en $LU = A$.

Merk op dat we de propositie alleen nog moeten aantonen voor $i = n + 1$. Immers, wegens de inductiehypothese geldt de uitspraak al $\forall i \in \{1, ..., n\}$; slechts over de 'buitenste schil' bestaat nog onzekerheid. Bovendien moet enkel van de meest rechtse kolom van U en onderste rij van L nog getoond worden dat hun elementen aan de gevraagde eigenschap voldoen. Van de elementen $U[n + 1; 1, ..., n]$ weten we immers dat ze nul zijn wegens het gegeven, en net zo voor $L[1, ..., n; n + 1]$.

Te bewijzen is dus:

$$\forall i \in \{1, ..., n + 1\} : U_{i,n+1} = A_{i,n+1} - \sum_{j=1}^{i-1} L_{i,j} U_{j,n+1}$$

en tevens:

$$\forall i \in \{1, ..., n + 1\} : L_{n+1,i} = \frac{A_{n+1,i} - \sum_{j=1}^{i-1} L_{n+1,j} U_{j,i}}{U_{ii}}$$

We tonen de eerste deelbewijslast aan door een willekeurige $i \in \{1, ..., n+1\}$ te kiezen. Bemerk dat we $A_{i,n+1}$ kunnen schrijven als $(LU)_{i,n+1} = \sum_{j=1}^{n+1} L_{i,j}U_{j,n+1}$, en bemerk bovendien dat de term van de som 0 wordt wanneer $i < j$; dan is $L_{i,j}$ immers gelijk aan nul. Dus kunnen we de som herschrijven als $\sum_{j=1}^{i} L_{i,j}U_{j,n+1}$. Wanneer $i = j$ geldt bovendien dat $L_{i,j} = L_{i,i} = 1$, waardoor we die index af kunnen scheiden om te bekomen dat $\sum_{j=1}^{i} L_{i,j}U_{j,n+1} = U_{i,n+1} + \sum_{j=1}^{i-1} L_{i,j}U_{j,n+1}$.

Dus weten we dat $A_{i,n+1} = U_{i,n+1} + \sum_{j=1}^{i-1} L_{i,j}U_{j,n+1}$. Dit vormen we om naar U om $U_{i,n+1} = A_{i,n+1} - \sum_{j=1}^{i-1} L_{i,j}U_{j,n+1}$ te bekomen. Aangezien i willekeurig gekozen was, is de gelijkheid aangetoond.

We tonen de tweede deelbewijslast aan door opnieuw een willekeurige $i \in \{1, ..., n+1\}$ te kiezen. Bemerk dat $A_{n+1,i}$ gelijk is aan $(LU)_{n+1,i} = \sum_{j=1}^{n+1} L_{n+1,j}U_{j,i}$. Hier kunnen we alle indices $j > i$ negeren, aangezien $U_{j,i}$ dan nul is. We splitsen dan weer één element uit de som af om te bekomen dat $A_{n+1,i} = L_{n+1,i} \cdot U_{i,i} + \sum_{j=1}^{i-1} L_{n+1,j}U_{j,i}$. Dit kunnen we weer omvormen tot $L_{n+1,i} \cdot U_{i,i} = A_{n+1,i} - \sum_{j=1}^{i-1} L_{n+1,j}U_{j,i}$, en na deling door $U_{i,i}$ in beide leden is de gelijkheid aangetoond.

Dus geldt de eigenschap ook voor $n + 1$.

Via het inductieve principe weten we nu dat de eigenschap geldt voor alle $n \in \mathbb{N}_0$. Hiermee is de volledige stelling aangetoond. ■

Opdracht 3

NB. Per de MATLAB-documentatie gebruiken we de standaardtolerantiewaarde van 10^{-12} bij het testen of de verwachte decompositie en berekende decompositie gelijk zijn. We gebruiken hiervoor de functie `ismembertol`. In de praktijk is de echte afwijking meestal een paar grootteordes kleiner.

Opdracht 4

Waar algoritme 5.1 een indexvariabele `k` gebruikt die van `n` tot `1` loopt, gebruiken we in de voorwaartse substitutie een variabele die van `1` tot `n` loopt, aangezien we bij de eerste rij beginnen. Verder is het idee achter de oplossingsmethode identiek: omdat de matrices driehoekig kunnen we de rijen op zo'n manier doorlopen dat elke volgende rij extra informatie geeft over één variabele van het stelsel. We nemen dan een lineaire combinatie van de al gekende waarden om die nieuwe variabele te 'isoleren'. Na een deling door het resterende element bekomen we de correcte waarde voor de variabele.

Opdracht 5

De correctheid wordt hier niet formeel bewezen, maar aan de hand van een zgh. *nothing up my sleeve*-test case[1] gemotiveerd.

De kern van `solve_Lb` bestaat uit de volgende drie regels:

```
1  for k = 1:n
2      y(k) = (b(k) - L(k, 1:k) * y(1:k)) / L(k, k);
3  end
```

welke we splitsen in

```
1  for k = 1:n
2      y(k) = b(k);
3      y(k) = y(k) - L(k, 1:k) * y(1:k); % 1 aftrekking, k verm., k - 1 opt.
4      y(k) = y(k) / L(k, k);            % 1 deling
5  end
```

In totaal levert ons dit $\sum_{k=1}^{n}(2k + 1) = n^2$ bewerkingen voor $L \in \mathbb{R}^{n \times n}$.

Voor `solve_Ub` gaan we analoog te werk:

[1] hier is U de getallen van één tot zes, b_1 de eerste drie kwadraten, L de rij van Fibonacci en b_2 de eerste drie cijfers van π

```
for k = n:-1:1
    y(k) = (b(k) - U(k, k+1:n) * y(k+1:n)) / U(k, k);
end
```

wordt

```
for k = n:-1:1
    y(k) = b(k);
    y(k) = y(k) - U(k, k+1:n) * y(k+1:n); % 1 aftrekking, n - k verm., n - k + 1 opt.
    y(k) = y(k) / U(k, k);                % 1 deling
end
```

Zo bekomen we $\sum_{k=1}^{n}(1+2(n-k)) = n^2 + n$ bewerkingen voor $U \in \mathbb{R}^{n\times n}$.

Opdracht 6

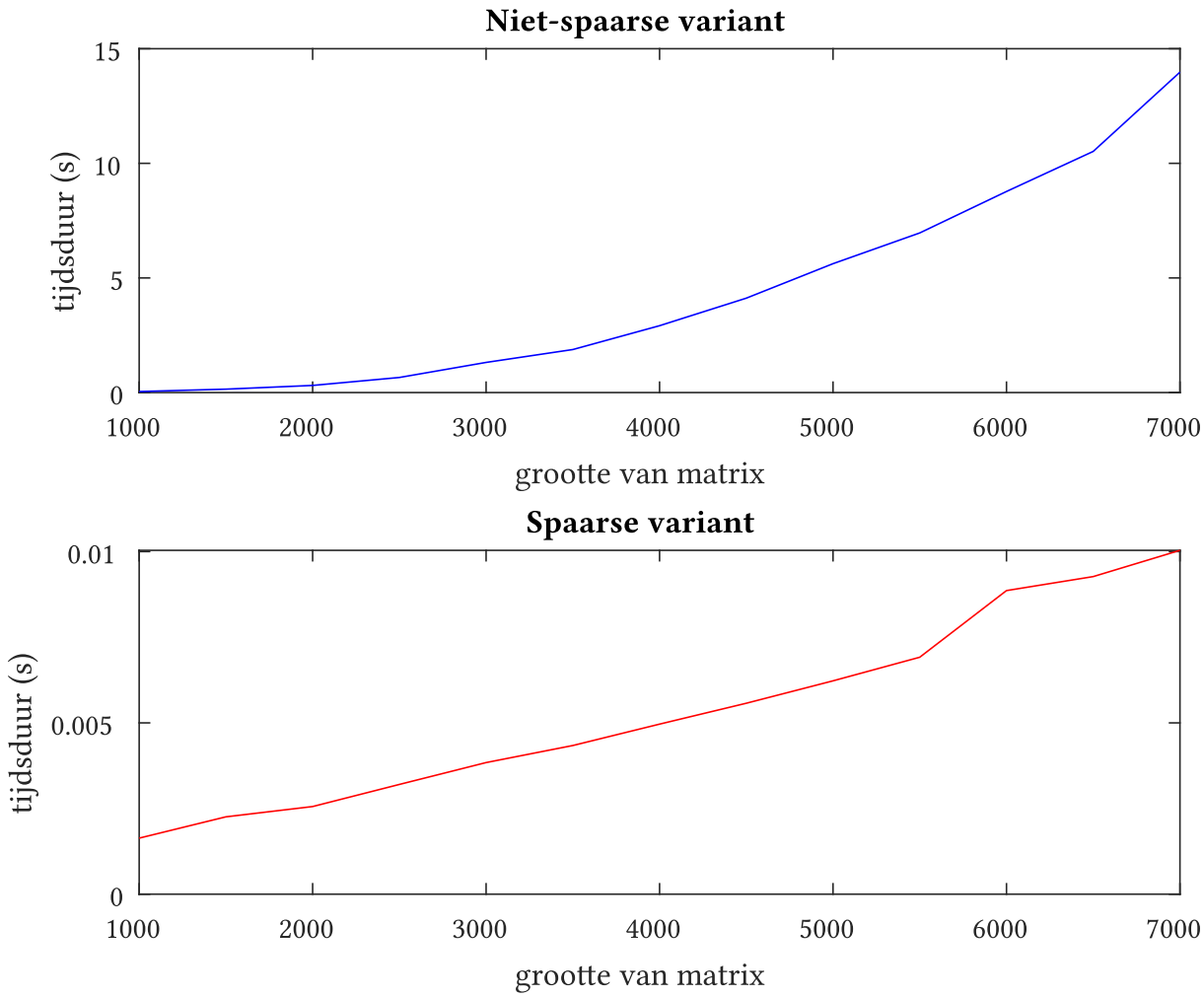
```
L_1 =

  1.000000000000000                   0                   0                   0                   0
  0.009090909090909   1.000000000000000                   0                   0                   0
  0.009090909090909  -0.000082651458798   1.000000000000000                   0                   0
  0.009090909090909  -0.000082651458798  -0.000082658290627   1.000000000000000                   0
  0.009090909090909  -0.000082651458798  -0.000082658290627  -0.000082665123584   1.000000000000000



U_1 =

  1.100000000000000   0.010000000000000   0.010000000000000   0.010000000000000   0.010000000000000
                   0   1.099090909090911  -0.000090909090909  -0.000090909090909  -0.000090909090909
                   0                   0   1.099090983395322  -0.000090916604678  -0.000090916604678
                   0                   0                   0   1.099090975880311  -0.000090924119689
                   0                   0                   0                   0   1.099090968364057



L_2 =

  1.000000000000000                   0                   0                   0                   0
                   0   1.000000000000000                   0                   0                   0
                   0                   0   1.000000000000000                   0                   0
                   0                   0                   0   1.000000000000000                   0
  0.009090909090909   0.009090909090909   0.009090909090909   0.009090909090909   1.000000000000000



U_2 =

  1.100000000000000                   0                   0                   0   0.010000000000000
                   0   1.100000000000000                   0                   0   0.010000000000000
                   0                   0   1.100000000000000                   0   0.010000000000000
                   0                   0                   0   1.100000000000000   0.010000000000000
                   0                   0                   0                   0   1.099636363636364
```

L_1 en U_1 bevatten ieder tien nullen ($= 2n$) en is dus niet per se spaars te noemen.
L_2 en U_2 bevatten ieder zestien nullen ($= (n-1)(n-2) + (n-1) = (n-1)^2$) en is dus volgens de conventie spaars te noemen.

Opdracht 8

`solve_Ub_special` vereist $\sum_{k=1}^{n-1} 3 = 3(n-1)$ bewerkingen (één vermenigvuldiging, één aftrekking en één deling). In vergelijking met de implementatie uit opdracht 5 is dit verband lineair in plaats van kwadratisch.

Opdracht 9



Figuur 1: Onze bevindingen: spaarse matrices hebben hun nut

Opdracht 11

Aangezien de bovengrens op relatieve fout recht evenredig is met het conditiegetal van de matrix; cfr. paragraaf 5.5.3 uit de cursus.

Opdracht 12

Als we beide leden met M_1 linksvermenigvuldigen en vervolgens de definitie van y invullen in het linkerlid, bekomen we (gegeven dat $M_2 x = y$):

$$M_1 M_1^{-1} A M_2^{-1}(M_2 x) = M_1 M_1^{-1} b$$
$$\Leftrightarrow Ax = b$$

Opdracht 13

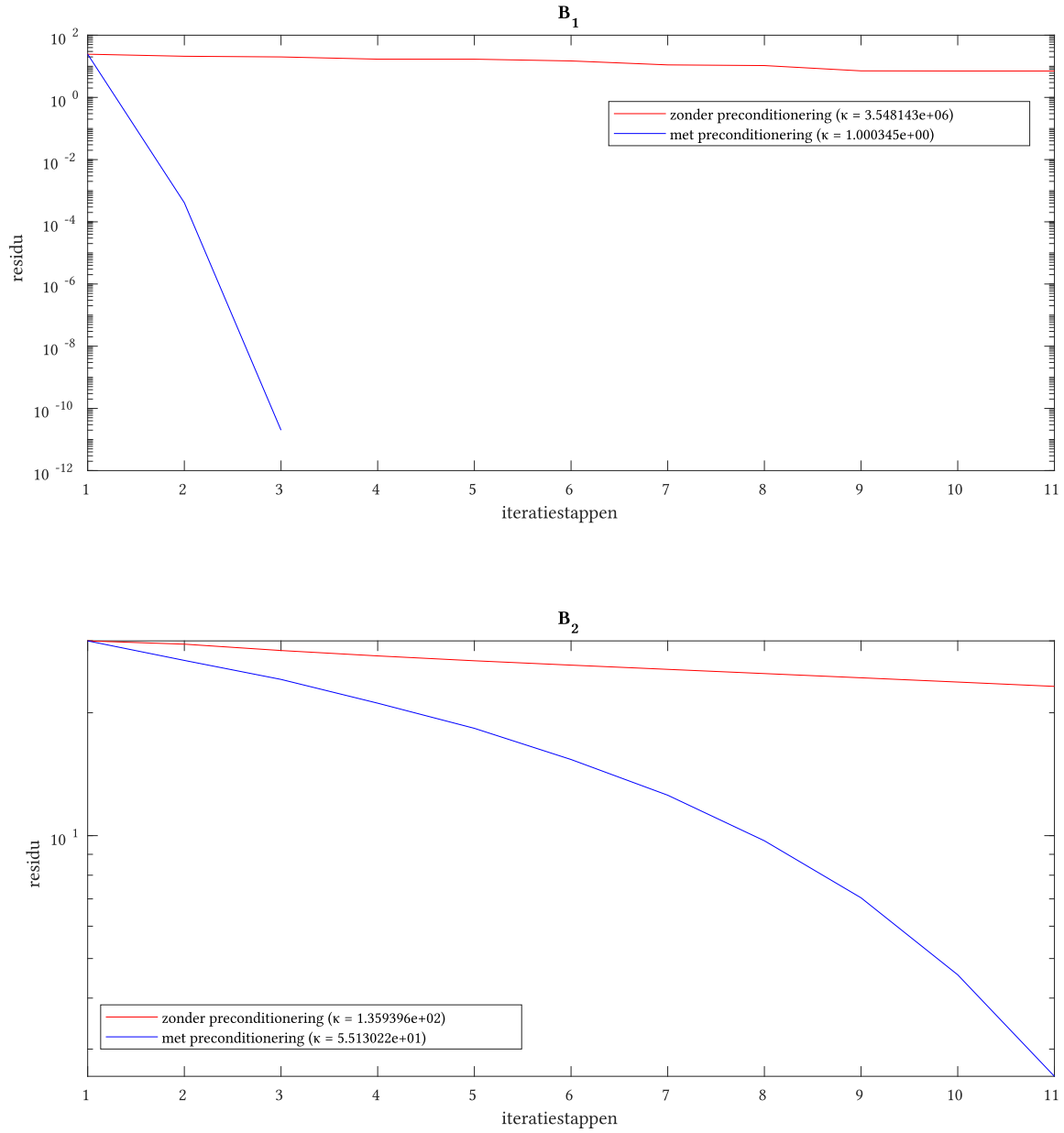
De preconditionering zal een impact hebben op het convergentiegedrag van B_1 in `gmres`, en niet op dat van B_2. Uit de tips over `gmres` halen we de informatie dat het conditiegetal van de matrix sterk samenhangt met het convergentiegedrag.

Aangezien B_1 en B_2 in MATLAB als spaarse matrices opgeslagen zijn, gebruiken we `condest` om het conditiegetal te berekenen. Bovendien gebruiken we de 2-norm. Voor B_1 geeft dit een waarde van $3.5481 \cdot 10^6$, voor B_2 verkrijgen we 135.9396. Na het toepassen van de preconditionering (lees: het rechtsvermenigvuldigen met $M^{-}1$, waarbij $M = LU$) bemerken we dat $\kappa(B_1) = 1.0003$, terwijl $\kappa(B_2) = 55.1302$. Deze enorme vermindering voor B_1 maar relatief kleine daling voor B_2 is een verklaring voor de verbetering van het convergentiegedrag.

Ook kunnen we de voorwaartse fout naar boven afschatten aan de hand van dezelfde formule die wij reeds uit paragraaf 5.5.3 haalden, namelijk dat

$$\frac{\|\Delta x\|}{\|x\|} \leq \kappa(A)\frac{\|r\|}{\|b\|}.$$

Zo bekomen we voor B_1 zonder preconditionering een bovengrens van $3.5481 \cdot 10^6 \cdot \frac{51.2195}{\sqrt{600}} \approx 7.4193 \cdot 10^6$ en na preconditionering een bovengrens van 1.000345. Voor B_2 bedragen deze waarden $3.979178 \cdot 10^2$ respectievelijk $1.095138 \cdot 10^2$.



Figuur 2: De convergentiesnelheden van B_1 en B_2 met en zonder preconditionering.

Opdracht 14

Zij $n \in \mathbb{N}$ een willekeurig getal, en zij A_1 en A_2 dan de resulterende matrices zoals we ze in opdracht 6 definieerden. Zij $P \in \mathbb{R}^{n \times n}$ de matrix met enen op de antidiagonaal en nullen op alle andere posities. We zoeken dan de matrix Q opdat $PA_1Q = A_2$.

Bemerk dat linksvermenigvuldigen met P het effect heeft van de matrix "over de horizontale as" te spiegelen: de eerste rij van PA_1 is gelijk aan de laatste rij van A_1; de tweede rij van PA_1 is gelijk aan de voorlaatste rij van A_1, enzovoort. We bemerken dat om van PA_1 naar A_2 te gaan, dus enkel nog "over de verticale as" gespiegeld moet worden: de eerste kolom van PA_1 moet gelijk worden aan de laatste kolom van A_2, de tweede kolom van PA_1 aan de voorlaatste van A_2, etcetera. Beschouw de matrix PA_1 als een eigen entiteit; noem deze bijvoorbeeld B.

We zoeken dus Q zodat $BQ = A_2$, waarbij Q een "kolomspiegeling" bewerkstelligt. Neem nu het getransponeerde van beide leden. We bekomen dan $Q^T B^T = A_2^T$; waarbij Q^T een "rijspiegeling" bewerkstelligt. Maar we hebben al een matrix die bij linksvermenigvuldiging rijen spiegelt, namelijk P. Dus geldt dat $Q^T = P$. Echter is P per definitie symmetrisch, dus is $Q = P$. Hierbij is de algemene vorm van Q gevonden.

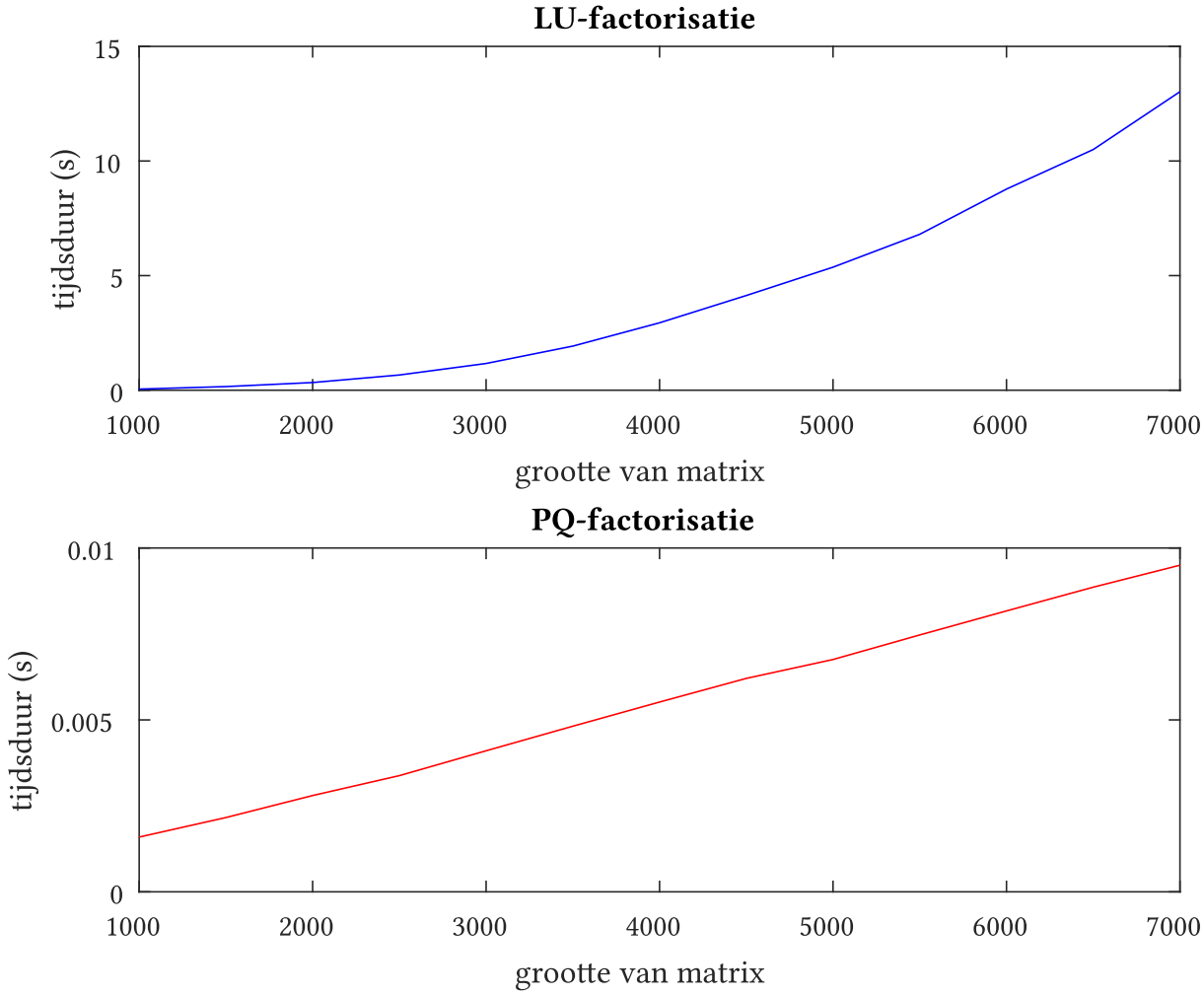
Opdracht 15

Gegeven dat $A_2 z = Pb$, met $A_1 x = b$, kunnen we deze gelijkheid invullen om $A_2 z = PA_1 x$ te bekomen. Verder weten we dat $PA_1 Q = A_2$, dus herschrijven we de vorige gelijkheid als $PA_1 Qz = PA_1 x$. Door links te vermenigvuldigen met P^{-1} en voorts met A_1^{-1} bekomen we

$$\begin{aligned} A_1^{-1} P^{-1} PA_1 Qz &= A_1^{-1} P^{-1} PA_1 x \\ \Leftrightarrow Qz &= x \end{aligned}$$

Opdracht 16

Weer merken we een kwadratisch-lineair onderscheid. Ook dit is te verwachten, aangezien we gebruik kunnen maken van spaarse matrices en hun lineaire uitvoeringstijden bij de PQ-methode, cfr. opdracht 9.



Opdracht 17

De boosdoener is de deling. Wegens het feit dat we zonder pivotering werken, komt het voor dat we door het zeer kleine element 10^{-20} delen. Zoals we in de cursus kunnen lezen:

> *"Kleine spilelementen veroorzaken grote afrondingsfouten"*
> pagina 94, hoofdstuk 5.5.1

Opdracht 18

De oplossing voor dit probleem is het gebruiken van rijpivotering. We gebruiken daarvoor het feit dat we de rijen vrijelijk mogen verwisselen in een stelsel zonder dat de oplossing verandert. Als we dit bijhouden in een permutatiematrix P, kunnen we zowel in het linker- als het rechterlid vermenigvuldigen met P om de gelijkheid te behouden. Dit kunnen we in ons voordeel laten werken door steeds een "tactisch" element op de spilpositie te plaatsen – dat houdt in: we willen een element dat in absolute waarde zo groot mogelijk is, zodat de afrondingsfouten tot een minimum beperkt blijven. De keuze bij uitstek is dus $\|a[i,n]\|_\infty$ waarbij a de i-de kolom van A is.