32. We gaan ervan uit dat $p^2 \not\approx 4q$. We hebben in punt 26 gezien dat het probleem dan slecht gecondioneerd is.

Opmerking. *In het geval dat $p^2 \approx 4q$ kan het probleem geformuleerd worden als: Bepaal de oplossingen van de vergelijking*

$$y^2 - \frac{4q + t^2}{2t} y + q = 0, \qquad \textit{met}\, t = p + \sqrt{p^2 - 4q}.$$

Het is gemakkelijk in te zien de oplossingen gegeven worden door

$$y_1 = \frac{2q}{t}, \qquad y_2 = \frac{t}{2}.$$

De conditiematrix wordt nu gegeven door:

$$\mathcal{K} = \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}.$$

Onder de aanname dat $p^2 \not\approx 4q$ is het eerste deel van het algoritme

$$u = p \otimes p, \; v = u \ominus (4 \otimes q), \; w = \sqrt{v}(1 + \epsilon_v), \qquad \text{met } |\epsilon_v| \leq \texttt{eps},$$

stabiel. Zoals in punt 28 is aangegeven, nemen we de fouten als gevolg van de computervoorstelling van p en q niet mee in de analyse. Volgens de definitie gegeven in punt 28 is

$$w = \sqrt{((p^2(1+\epsilon_1) - 4q(1+\epsilon_2))(1+\epsilon_3)}(1+\epsilon_v), \qquad \text{met } |\epsilon_{1,2,3}| \leq \texttt{eps}.$$

De relatieve fout wordt nu gegeven door

$$\frac{w - \sqrt{p^2 - 4q}}{\sqrt{p^2 - 4q}}.$$

Om een bovengrens voor de relatieve fout te geven, gebruiken we dat

$$\begin{aligned} w &= \sqrt{p^2 - 4q + p^2\epsilon_1 - 4q\epsilon_2 + (p^2 - 4q)\epsilon_3 + \mathcal{O}(\texttt{eps}^2)}(1+\epsilon_v) \\ &= \left(\sqrt{p^2 - 4q} + \frac{p^2\epsilon_1 - 4q\epsilon_2 + (p^2 - 4q)\epsilon_3}{2\sqrt{p^2 - 4q}} \right)(1+\epsilon_v) + \mathcal{O}(\texttt{eps}^2) \\ &= \sqrt{p^2 - 4q}(1+\epsilon_v) + \frac{p^2\epsilon_1 - 4q\epsilon_2 + (p^2 - 4q)\epsilon_3}{2\sqrt{p^2 - 4q}} + \mathcal{O}(\texttt{eps}^2). \end{aligned}$$

In de tweede vergelijking hebben we gebruik gemaakt van

$$\sqrt{X+Y} = \sqrt{X} + \frac{1}{2\sqrt{X}} Y + \mathcal{O}(Y^2),$$

met $X = p^2 - 4q$ en $Y = p^2\epsilon_1 - 4q\epsilon_2 + (p^2-4q)\epsilon_3 + \mathcal{O}(\texttt{eps}^2)$. Hiermee verkrijgen we dat

$$\begin{aligned}\frac{w - \sqrt{p^2-4q}}{\sqrt{p^2-4q}} &\approx \frac{\sqrt{p^2-4q}\,\epsilon_v + \frac{p^2\epsilon_1 - 4q\epsilon_2 + (p^2-4q)\epsilon_3}{2\sqrt{p^2-4q}}}{\sqrt{p^2-4q}} \\ &= \epsilon_v + \frac{p^2\epsilon_1 - 4q\epsilon_2}{2(p^2-4q)} + \frac{1}{2}\epsilon_3,\end{aligned}$$

zodat

$$\left| \frac{w - \sqrt{p^2-4q}}{\sqrt{p^2-4q}} \right| \leq \left(\frac{3}{2} + \frac{p^2+4q}{2(p^2-4q)} \right) \texttt{eps}.$$

We vergelijken deze uitdrukking met de conditie van het probleem: Bereken $\sqrt{p^2-4q}$. De conditiematrix wordt gegeven door

$$\mathcal{K} = \begin{pmatrix} \dfrac{p^2}{p^2-4q} & \dfrac{-2q}{p^2-4q} \end{pmatrix}.$$

Het algoritme is dus stabiel aangezien de numerieke fouten niet de probleem-fouten overschrijden, zie blz. 6 van de studieleidraad.

Om nu de nulpunten te bepalen, kunnen we de volgende bewerkingen doen

$$y_1 = \frac{p+w}{2} \qquad \text{en} \qquad y_2 = \frac{p-w}{2}.$$

De berekening van y_1 zal zonder problemen verlopen aangezien p en w beide positief zijn. In punt 25 hebben we gezien dat het aftrekken van twee getallen die ongeveer even groot zijn slecht geconditioneerd is. Deze situatie komt voor wanneer $p^2 \gg 4q$, zodat $\sqrt{p^2-4q} \approx p$. Om dit te voorkomen merken we op dat

$$y_1 y_2 = q.$$

Dit geeft een andere manier om het tweede nulpunt te berekenen, namelijk $y_2 = \frac{q}{y_1}$. Nu is het probleem goed geconditioneerd (zie punt 25) en het algoritme stabiel.

33. Volgens Regel 1 is het aanbevolen om eerst de bewerkingen uit te voeren die slecht geconditioneerd zijn en daarna de beter geconditioneerde bewerkingen. Wanneer $x \approx -\frac{a_1}{a_2}$ is het dus aanbevolen om eerst $u = a_2 \otimes x$, dan $v = a_1 \oplus a_2 x$ te berekenen en vervolgens $w = x \otimes u$. Een gelijksoortige opmerking geeft Regel 2.

Voorbeeld 1. *Beschouw het polynoom $p(x) = (x-1)^8$. Het Binomium Van Netwon geeft*

$$p(x) = x^8 - 8x^7 + 28x^6 - 56x^5 + 70x^4 - 56x^3 + 28x^2 - 8x + 1.$$

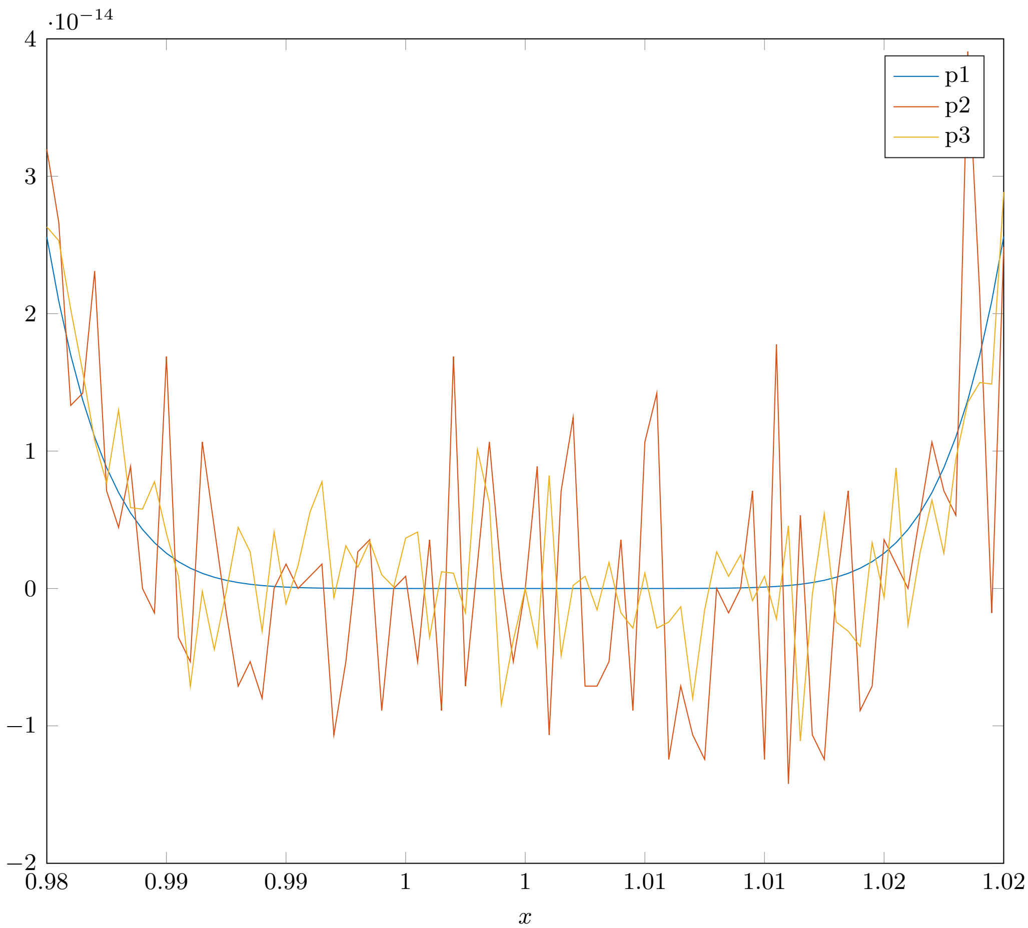
Toepassen van het algoritme van Horner geeft vervolgens dat

$$p(x) = x\,(x\,(x\,(x\,(x\,(x\,(x\,(x-8)+28)-56)+70)-56)+28)-8)+1.$$

In Algoritme 1 is de MATLAB code te zien die een plot genereert rond het nulpunt $x = 1$ waarin de verschillende manieren om $p(x)$ te berekenen worden vergeleken. In fig. 1 zien we duidelijk dat het algoritme van Horner gemiddeld betere functiewaarden oplevert dan het berekenen van de functiewaarden in de uitgeschreven vorm.

Algoritme 1: MATLAB code

```
x=linspace(0.98,1.02,81);
p1=(1-x).^8;
p2=x.^8 - 8*x.^7 + 28*x.^6 - 56*x.^5 + 70*x.^4 -
    56*x.^3 + 28*x.^2 - 8*x + 1;
p3=x.*(x.*(x.*(x.*(x.*(x.*(x.*(x - 8) + 28) - 56)
    + 70) - 56) + 28) - 8) + 1;
plot(x,p1,x,p2,x,p3)
legend
xlabel('x');
```



Figuur 1: Vergelijking tussen de functiewaarden van het polynoom $p(x)$ berekend op drie verschillende manieren. Zie de tekst voor verdere uitleg.