

1.5.1 Analiza zaokrožitvenih napak za produkt $n + 1$ števil

Računamo produkt $p = x_0 x_1 \cdots x_n$ predstavljivih števil $x_0, x_1, \dots, x_n$.

Eksaktni algoritem:

$$p_0 = x_0$$

$$i = 1, \dots, n$$

$$p_i = p_{i-1} x_i$$

$$p = p_n$$

Dejanski algoritem:

$$\hat{p}_0 = x_0$$

$$i = 1, \dots, n$$

$$\hat{p}_i = \hat{p}_{i-1} x_i (1 + \delta_i), \quad |\delta_i| \leq u$$

$$\hat{p} = \hat{p}_n$$

Dobimo

$$\hat{p} = p(1 + \gamma) = p(1 + \delta_1) \cdots (1 + \delta_n).$$

Velja

$$(1 - u)^n \leq 1 + \gamma \leq (1 + u)^n.$$

Ocenimo

$$(1 + u)^n = 1 + \binom{n}{1}u + \binom{n}{2}u^2 + \cdots = 1 + nu + \mathcal{O}(u^2),$$

$$(1 - u)^n \geq 1 - nu. \quad (\text{indukcija})$$

Če je $nu \ll 1$ ocenimo $|\gamma| \leq nu$. Računanje produkta $n + 1$ števil je direktno in obratno stabilno.

1.5.2 Analiza zaokrožitvenih napak za skalarni produkt

Imamo dva vektorja predstavljenih števil $x = (x_1 \cdots x_n)^T$ in $y = (y_1 \cdots y_n)^T$, računamo pa $s = y^T x = \sum_{i=1}^n x_i y_i$.

Eksaktni algoritem:

$$s_0 = 0$$

$$i = 1, \dots, n$$

$$p_i = x_i y_i$$

$$s_i = s_{i-1} + p_i$$

$$s = s_n$$

Dejanski algoritem:

$$\hat{s}_0 = 0$$

$$i = 1, \dots, n$$

$$\hat{p}_i = x_i y_i (1 + \alpha_i), \quad |\alpha_i| \leq u$$

$$\hat{s}_i = (\hat{s}_{i-1} + \hat{p}_i)(1 + \beta_i), \quad |\beta_i| \leq u$$

$$\hat{s} = \hat{s}_n$$

Obratna analiza vrne $\hat{s} = \sum_{i=1}^n x_i y_i (1 + \gamma_i)$, kjer je

$$1 + \gamma_1 = (1 + \alpha_1)(1 + \beta_2) \cdots (1 + \beta_n)$$

in

$$1 + \gamma_i = (1 + \alpha_i)(1 + \beta_i) \cdots (1 + \beta_n), \quad i = 2, \dots, n.$$

Tako dobimo ocene $|\gamma_1| \leq nu$ in $|\gamma_i| \leq (n - i + 2)u$ za $i = 2, \dots, n$. To pomeni, da je $\hat{s}$ točni skalarni produkt relativno malo zmotenih vektorjev x in y . Računanje skalarne produkta je obratno stabilno.

Direktna analiza za izračun skalarnega produkta

Pri direktni analizi najprej izračunamo absolutno napako:

$$\hat{s} - s = \sum_{i=1}^n x_i y_i \gamma_i,$$

torej

$$|\hat{s} - s| \leq \sum_{i=1}^n |x_i| \cdot |y_i| \cdot |\gamma_i| \leq nu \sum_{i=1}^n |x_i| \cdot |y_i| = nu |y|^T |x|.$$

Dobimo

$$\left| \frac{\hat{s} - s}{s} \right| \leq \frac{|y|^T |x|}{|y^T x|} nu.$$

Če so vsi $x_i y_i$ enakega predznaka, dobimo $\left| \frac{\hat{s} - s}{s} \right| \leq nu$ in računanje je direktno stabilno, sicer pa imamo v primeru, ko sta vektorja skoraj pravokotna, lahko veliko relativno napako.

V splošnem torej računanje skalarnega produkta **ni direktno stabilno**.

1.5.3 Računanje vrednosti polinoma po Hornerju

Računamo vrednost polinoma $p(t) = a_0t^n + a_1t^{n-1} + \dots + a_n$ v točki x .

Eksaktni algoritem je:

$$p_0 = a_0$$

$$i = 1, \dots, n$$

$$p_i = p_{i-1}x + a_i$$

$$p = p_n$$

Dejanski algoritem pa:

$$\hat{p}_0 = a_0$$

$$i = 1, \dots, n$$

$$\hat{p}_i = (\hat{p}_{i-1}x(1 + \alpha_i) + a_i)(1 + \beta_i)$$

$$\hat{p} = \hat{p}_n$$

Dobimo

$$\hat{p} = a_0x^n(1 + \gamma_0) + a_1x^{n-1}(1 + \gamma_1) + \dots + a_n(1 + \gamma_n),$$

kjer lahko ocenimo $|\gamma_i| \leq 2nu$ za $i = 0, \dots, n$. Računanje vrednosti polinoma po Hornerju je obratno stabilno, saj dobimo vrednost bližnjega polinoma s koeficienti $a_i(1 + \gamma_i)$ namesto a_i v točki x .

Iz absolutne napake $\hat{p} - p = a_0x^n\gamma_0 + a_1x^{n-1}\gamma_1 + \dots + a_n\gamma_n$ sledi

$$\frac{|\hat{p} - p|}{|p|} \leq \frac{2nu(|a_0||x^n| + |a_1||x^{n-1}| + \dots + |a_n|)}{|a_0x^n + \dots + a_n|}.$$

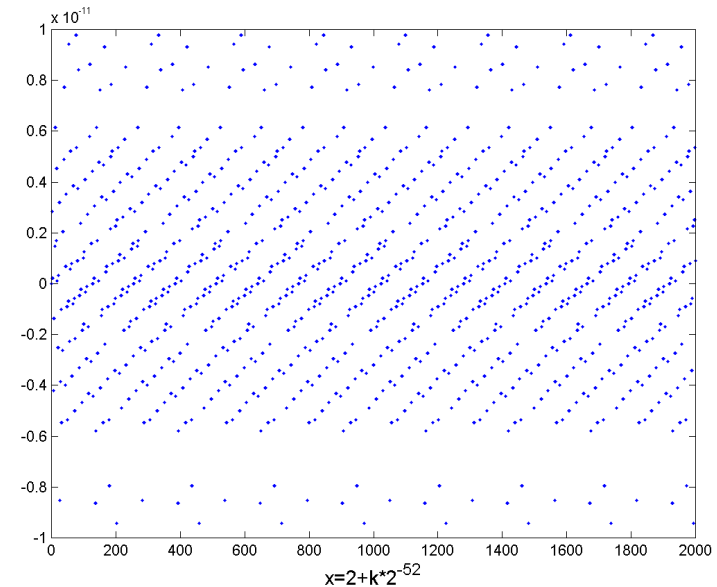
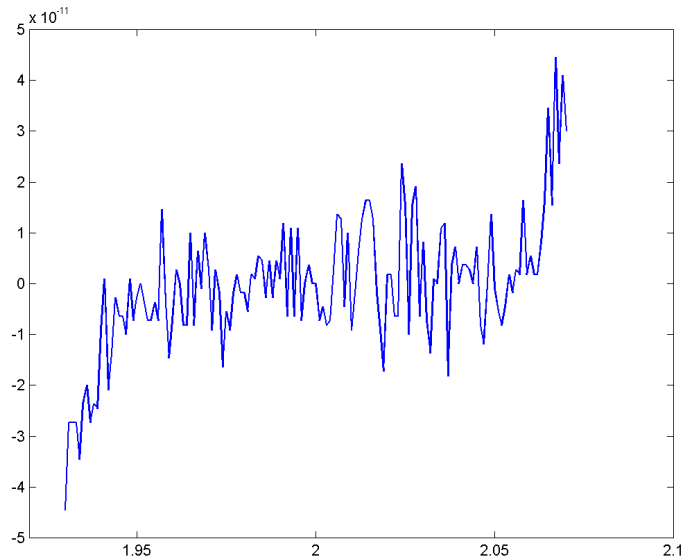
Računanje vrednosti polinoma po Hornerju ni direktno stabilno.

Računanje vrednosti polinoma po Hornerju - 2.del

Za predstavo o dobljenih ocenah si poglejmo računanje polinoma $(x - 2)^9$ oz.

$$x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512$$

v okolici točke 2. Z Matlabom dobimo



Napake niso povsem naključne, saj v bližini 2 odkrijemo vzorec.

Računanje vrednosti polinoma po Hornerju - 3.del

V algoritem lahko vstavimo sprotno računanje ocene:

$$p_0 = a_0$$

$$e_0 = |a_0|$$

$$i = 1, \dots, n$$

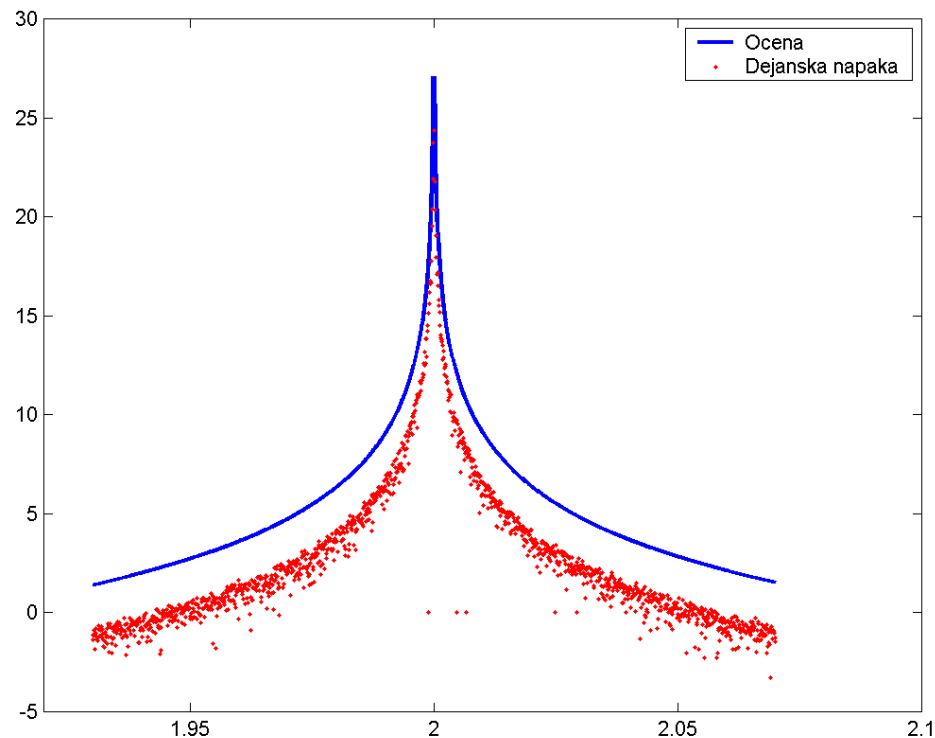
$$p_i = p_{i-1}x + a_i$$

$$e_i = e_{i-1}|x| + |a_i|$$

$$p = p_n$$

$$e = 2nu \frac{e_n}{|p|}$$

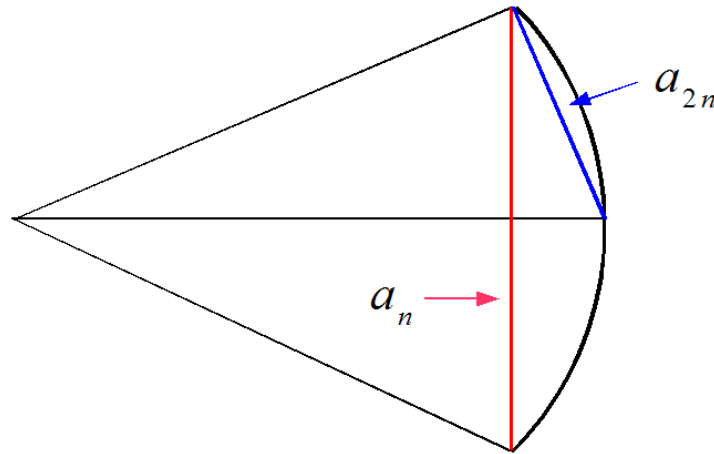
e je ocena za relativno napako.



Graf prikazuje resnično napako in oceno za napako v logaritemski skali. Ocena je res zgornja meja in dejanska napaka je manjša ali enaka, vendar zgled kaže, da so v bližini točke 2 napake res večje in tudi obnašajo se tako, kot napoveduje ocena.

1.6.1 Poučni primeri - računanje števila π

π je limita obsega S_n pravilnega mnogokotnika, včrtanega v krog s polmerom $r = \frac{1}{2}$. Naj bo a_n stranica pravilnega n -kotnika. Poiščimo zvezo med a_n in a_{2n} :



Velja

$$a_{2n} = \sqrt{\left(\frac{a_n}{2}\right)^2 + \left(\frac{1}{2} - \sqrt{\frac{1}{4} - \left(\frac{a_n}{2}\right)^2}\right)^2} = \sqrt{\frac{1 - \sqrt{1 - a_n^2}}{2}},$$

od tod pa iz $S_n = na_n$ sledi

$$S_{2n} = 2na_{2n} = 2n\sqrt{\frac{1 - \sqrt{1 - \left(\frac{S_n}{n}\right)^2}}{2}}.$$

Računanje števila π , 2.del

Začnemo pri $S_6 = 3$ in uporabljamo formulo $S_{2n} = 2n \sqrt{\frac{1 - \sqrt{1 - \left(\frac{S_n}{n}\right)^2}}{2}}$.

n	S_n	n	S_n
6	3.0000000	768	3.1430728
12	3.1058285	1536	3.1486604
24	3.1326292	3072	3.1374750
48	3.1393456	6144	3.1819806
96	3.1410186	12288	3.0000000
192	3.1414995	24576	4.2426405
384	3.1416743	49152	0.0000000

Formula odpove, saj pride do odštevanja skoraj enako velikih števil, napaka pa se množi z $2n$. V zgornji tabeli (enojna natančnost) so napačne decimalke rdeče.

Kadar imamo nestabilen postopek, nam ne pomaga niti računanje z večjo natančnostjo. Prava rešitev je preurediti postopek tako, da se med računanjem ne izgublja natančnost.

Računanje števila π , 3.del

Za stabilno računanje je potrebno formulo preurediti. Stabilna oblika je

$$S_{2n} = 2n \sqrt{\frac{\left(1 - \sqrt{1 - \left(\frac{S_n}{n}\right)^2}\right) \left(1 + \sqrt{1 - \left(\frac{S_n}{n}\right)^2}\right)}{2 \left(1 + \sqrt{1 - \left(\frac{S_n}{n}\right)^2}\right)}}} = S_n \sqrt{\frac{2}{1 + \sqrt{1 - \left(\frac{S_n}{n}\right)^2}}}.$$

Sedaj dobimo pravilne rezultate:

n	S_n	n	S_n
6	3.0000000	768	3.1415839
12	3.1058285	1536	3.1375904
24	3.1326287	3072	3.1415920
48	3.1393502	6144	3.1415925
96	3.1410320	12288	3.1415925
192	3.1414526	24576	3.1415925
384	3.1415577	49152	3.1415925

1.6.2 Računanje I_{10}

Integrale $I_n = \int_0^1 x^n e^{x-1} dx$, $n = 0, 1, \dots$, lahko računamo rekurzivno preko formule

$$I_n = x^n e^{x-1} \Big|_0^1 - n \int_0^1 x^{n-1} e^{x-1} dx = 1 - nI_{n-1},$$

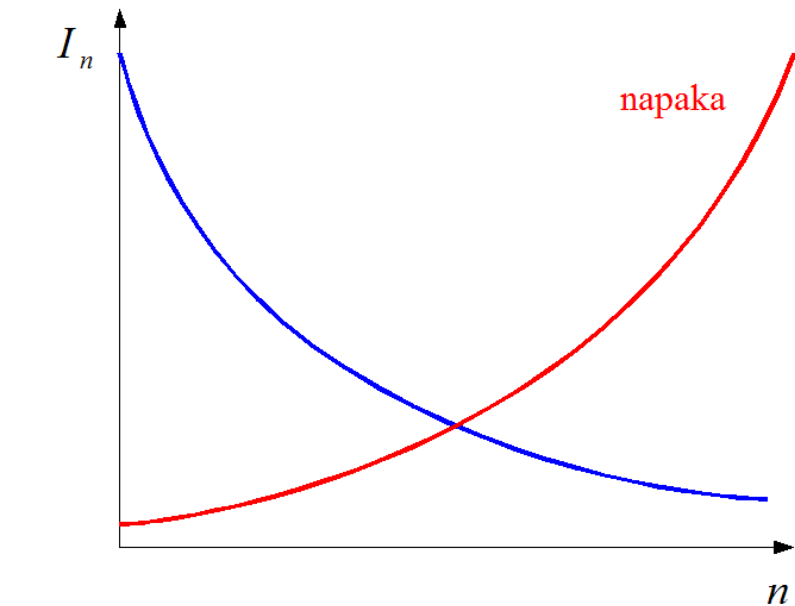
saj poznamo začetno vrednost $I_0 = 1 - e^{-1}$. V enojni natančnosti dobimo

n	I_n	n	I_n
0	0.6321205	7	0.1124296
1	0.3678795	8	0.1005630
2	0.2642411	9	0.0949326
3	0.2072767	10	0.0506744
4	0.1708932	11	0.4425812
5	0.1455340	12	-4.3109741
6	0.1267958	13	57.0426636

Razlog je v formuli $I_n = 1 - nI_{n-1}$. Napaka pri členu I_{n-1} se pomnoži z n in torej po absolutni vrednosti hitro narašča, točne vrednosti I_n pa padajo.

Računanje I_{10} , 2. del

Če računamo v obratni smeri: $I_{n-1} = \frac{1-I_n}{n}$, se napaka v vsakem koraku deli z n . Če začnemo pri nekem dovolj velikem členu, lahko z začetnim $I_n = 0$ izračunamo vse začetne člene dovolj natančno. Če začnemo z $I_{26} = 0$ tako dobimo (v enojni natančnosti) vse člene od I_{12} do I_0 na vse decimalke točno.



$$I_n = 1 - nI_{n-1}$$

n	I_n	n	I_n
0	0.6321205	8	0.1009320
1	0.3678795	9	0.0916123
2	0.2642411	10	0.0838771
3	0.2072766	11	0.0773522
4	0.1708934	12	0.0717733
5	0.1455329	13	0.0669477
6	0.1268024	$\vdots$	$\vdots$
7	0.1123835	26	0.0000000

1.6.3 Reševanje kvadratne enačbe

Rešitvi kvadratne enačbe $ax^2 + bx + c = 0$ sta podani s formulo

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

$a = 1.2345678$, $b = 76543210.5$, $c = 0.1122334455$, dvojna natančnost $\implies$

$$x_1 = -6.200000558900046 \cdot 10^7, \quad x_2 = -6.034970778375905 \cdot 10^{-9},$$

točni ničli pa sta

$$\tilde{x}_1 = -6.200000558900046 \cdot 10^7, \quad \tilde{x}_2 = -1.466275647008561 \cdot 10^{-9}.$$

Težava se je pojavila pri odštevanje približno enako velikih števil pri računanju x_2 . Rešitev je, da eno rešitev izračunamo po formuli (odvisno od predznaka b), drugo pa dobimo iz Vietove formule $x_1 x_2 = c/a$.

Tako iz $x_2 = c/(ax_1)$ dobimo pravilno $x_2 = -1.466275647008561 \cdot 10^{-9}$.

1.6.4 Računanje $(e^x - 1)/x$

Računamo $f(x) = (e^x - 1)/x$ v bližini $x = 0$, kjer je $\lim_{x \rightarrow 0} f(x) = 1$.

Algoritem 1

```
if  $x = 0$ 
     $y = 1$ 
else
     $y = (e^x - 1)/x$ 
end
```

Algoritem 2

```
 $z = e^x$ 
if  $z = 1$ 
     $y = 1$ 
else
     $y = (z - 1)/\ln z$ 
end
```

x	Algoritem 1	Algoritem 2
10^{-7}	1.0000000049433680	1.0000000050000002
10^{-8}	$9.999999939225290 \cdot 10^{-1}$	1.0000000005000000
10^{-9}	1.0000000082740371	1.0000000000500000
10^{-10}	1.0000000082740371	1.0000000000050000
10^{-11}	1.0000000082740371	1.0000000000005000
10^{-12}	1.000088900582341	1.0000000000000500
10^{-13}	$9.992007221626408 \cdot 10^{-1}$	1.0000000000000050
10^{-14}	$9.992007221626408 \cdot 10^{-1}$	1.0000000000000005
10^{-15}	1.110223024625156	1.0000000000000000
10^{-16}	0.0000000000000000	1.0000000000000000

Primerjava algoritmov za računanje $(e^x - 1)/x$ pri $x = 7 \cdot 10^{-14}$

Pri prvem algoritmu dobimo

$$y = fl\left(\frac{e^x - 1}{x}\right) = fl\left(\frac{6.994405055138486 \cdot 10^{-14}}{7.0000000000000001 \cdot 10^{-14}}\right) = 0.99920072216264.$$

Pri drugem algoritmu pa

$$y = fl\left(\frac{e^x - 1}{\ln(e^x)}\right) = fl\left(\frac{6.994405055138486 \cdot 10^{-14}}{6.994405055138241 \cdot 10^{-14}}\right) = 1.0000000000000004.$$

To je sicer lepo, vendar! Točne vrednosti, ki bi morale nastopati v drugem algoritmu so

$$y = fl\left(\frac{e^x - 1}{\ln(e^x)}\right) = fl\left(\frac{7.00000000000000245 \cdot 10^{-14}}{7.0000000000000001 \cdot 10^{-14}}\right) = 1.0000000000000004.$$

Kako je možno, da iz dveh netočnih vrednosti dobimo točno? Pokažimo, da je drugi algoritem stabilen.

Analiza stabilnega algoritma za računanje $(e^x - 1)/x$

Najprej izračunamo $\hat{z} = e^x(1 + \delta)$, kjer je $|\delta| \leq u$.

Denimo, da je $\hat{z} = 1$. To pomeni $e^x(1 + \delta) = 1$, torej

$$x = -\ln(1 + \delta) = -\delta + \delta^2/2 - \delta^3/3 + \dots,$$

torej je pravilno zaokrožena vrednost $f(x)$ na 1, saj je $f(x) = 1 + x/2 + x^2/6 + \dots$.

Če je $\hat{z} \neq 1$, potem računamo $y = (z - 1)/\ln z$. Numerično velja

$$\hat{y} = \frac{(\hat{z} - 1)(1 + \epsilon_1)}{\ln \hat{z}(1 + \epsilon_2)}(1 + \epsilon_3), \quad |\epsilon_1|, |\epsilon_2|, |\epsilon_3| \leq u.$$

Če vpeljemo $w = \hat{z} - 1$, dobimo

$$g(\hat{z}) := \frac{\hat{z} - 1}{\ln \hat{z}} = \frac{w}{\ln(1 + w)} = 1 + \frac{w}{2} + \mathcal{O}(w^2).$$

To pomeni, da za majhne x , ko je $z \approx 1$, velja

$$g(\hat{z}) - g(z) \approx \frac{\hat{z} - z}{2} \approx \frac{e^x \delta}{2} \approx \frac{\delta}{2}.$$

Ker je $g(z) \approx 1$, to pomeni, da je $\hat{y} = y(1 + \alpha)$, kjer je $\alpha \leq 3.5u$.