1. Výpočet Eulerova čísla

(Martin Mareš, 27. 11. 2006)

Úloha: Spočítejte Eulerovo číslo na n desetinných míst.

Řešení: Vyjdeme ze známého vzorečku

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}, \quad \text{a tedy } e = \sum_{k=0}^{\infty} \frac{1}{k!}.$$

(důkaz viz Taylorova věta v libovolných skriptech z matematické analýzy). Budeme počítat s dlouhými čísly s přesností na N desetinných míst (pro nějaké $N \geq n$; jaké přesně, to si zvolíme časem), jakmile $1/k!$ bude menší než tato přesnost, se sčítáním skončíme. Jelikož víme, že $2 < e < 3$ a první dva členy jsou jedničky, stačí počítat s čísly v intervalu $\langle 0, 1)$ a úvodní dvojku vypsat zvlášť.

Algoritmus tedy bude vypadat následovně (aritmetické operace na proměnných s a f pracují s přesností $\pm 10^{-N}$):

1. $s \leftarrow 0$
2. $f \leftarrow 1/2$
3. $i \leftarrow 2$
4. Dokud $f > 0$ (tedy dokud $f \geq 10^{-N}$, protože počítáme na N desetinných míst):
5. $\quad s \leftarrow s + f$
6. $\quad i \leftarrow i + 1$
7. $\quad f \leftarrow f/i$
8. Vydáme výsledek $2 + s$.

Analýza: Pochopit, jak program funguje a jak přesný výsledek v závislosti na N vydá, není úplně jednoduché, protože je zatížen různými zaokrouhlovacími chybami. Označme si tedy:

- $\varepsilon = 10^{-N}$ přesnost naší aritmetiky (víme, že výsledek každé operace se od správného liší o méně než ε)
- s_i a f_i hodnoty proměnných s a f v i-té iteraci
- m celkový počet iterací, po nichž se algoritmus zastaví (tedy nejmenší m takové, že $s_m < \varepsilon$)
- s_i^* a f_i^* hodnoty, které by proměnné měly, pokud bychom vůbec nezaokrouhlovali.
  Tedy $f_i^* = 1/i!$ a $s_i^* = \sum_{k=0}^{i} 1/i!$.
- $s_i^\Delta = s_i^* - s_i$ a $f_i^\Delta = f_i^* - f_i$ chybu našeho výpočtu. Jelikož zaokrouhlujeme dolů, je $s_i^\Delta \geq 0$ a $f_i^\Delta \geq 0$.

Časová složitost: Zkusme nejdříve spočítat, po kolika iteracích se algoritmus zastaví. Zjistit to přesně je těžké, ale najděme nějaký horní odhad, tedy nějaké (ne nutně minimální) M takové, že $f_M < \varepsilon$. Jelikož $f_i \leq f_i^*$, stačí, aby $f_M^* = 1/M!$ bylo menší než ε. Chceme tedy vyřešit nerovnici:

$$\frac{1}{M!} < 10^{-N}.$$

Faktoriál se invertuje těžko, ale pomůžeme si tím, že $k! \geq k^{k/2}$, takže stačí, aby hledané M splňovalo:

$$\frac{1}{M^{M/2}} < \frac{1}{10^N}, \quad \text{čili} \quad \frac{M}{2} \cdot \log M > N.$$

$M \log M$ se ale také těžko invertuje přesně, tak zvolíme hrubší odhad: $M = 4N/\log N$ a získáme:

$$\frac{M \log M}{2} = \frac{4N \log(4N/\log N)}{2 \log N} = \frac{2N(\log 4 + \log N - \log\log N)}{\log N} > N$$

(přičemž poslední nerovnost platí proto, že $\log\log N < \frac{1}{2} \log N$, a tedy výraz v závorce vyjde $> \frac{1}{2} \log N$).[1]

Jednu iteraci algoritmu zvládneme spočítat v čase $\mathcal{O}(N)$, iterací je celkem $\mathcal{O}(N/\log N)$, takže časová složitost vychází $\mathcal{O}(N^2/\log N)$.

Přesnost: Ještě jsme neodhalili, jak přesně musíme počítat (tedy jaké zvolit N), abychom dostali výsledek správně na n desetinných míst. V algoritmu jsou dva zdroje chyb: jednak místo nekonečné řady sčítáme konečnou, jednak jednotlivé operace provádíme jen s omezenou přesností. Pojďme tedy zjistit, jak velké tyto chyby jsou:

Chyba daná chybějícími členy: Z další chytré věty z analýzy (konkrétně věta o Lagrangeově tvaru Taylorova zbytku) plyne, že:

$$\left| e^x - \sum_{k=0}^{m} \frac{x^k}{k!} \right| = \frac{x^{k+1} \cdot e^\xi}{(k+1)!}$$

pro nějaké $\xi \in \langle 0, x \rangle$. My ovšem dosazujeme $x = 1$, takže si můžeme být jisti, že $|e - s_m^*| \leq e/(m+1)!$. Přitom $1/(m+1)! < \varepsilon$, pročež chyba je nejvýše $e \cdot \varepsilon < 3\varepsilon$. Stačí tedy spočítat o číslici víc :)

Chyba zaokrouhlovací: Přemýšlejme, kolik činí f_i^Δ a s_i^Δ. Na počátku výpočtu je $f_0^\Delta = s_0^\Delta$, v každé další iteraci pak $s_{i+1}^\Delta \leq s_i^\Delta + f_i^\Delta$ (samo sčítání je dokonale přesné, ale přičítáme f_i, které je zatíženo chybou) a $f_{i+1}^\Delta \leq f_i^\Delta/(i+1) + \varepsilon$

[1] Pro ty, kterým vrtá hlavou, jak přesně jsme m odhadli: Podobně můžeme pomocí nerovnosti $k! \leq (k/2)^k$ spočítat dolní odhad a ukázat, že $m \geq \alpha N/\log N$ pro nějakou konstantu α. To ale nebudeme potřebovat.

(hodnota f_{i+1} je jednak zatížena zaokrouhlovací chybou ve výši maximálně ε a jednak vychází z nepřesné hodnoty f_i, ovšem tu použije vydělenou číslem $i+1$, takže i chyba tolikrát klesne).

Indukcí snadno dokážeme, že pro všechna i je $f_i^\Delta < 2\varepsilon$, a tedy $s_i^\Delta < 2i\varepsilon$. Proto $s_m^\Delta < 2m\varepsilon < 4\varepsilon(N/\log N) = 4 \cdot 10^{-N} \cdot N/\log N$. Proto chceme zvolit N tak, aby platilo:

$$s_m^\Delta < 10^{-n}, \quad \text{a tedy} \quad \frac{4 \cdot 10^{-N} \cdot N}{\log N} < 10^{-n}.$$

Po zlogaritmování obou stran a přehození znamének:

$$-\log 4 + N - \log N + \log\log N > n.$$

Pokud nyní zvolíme $N > n + \log n + 2$, máme výsledek zaručen.

Závěrem: Posčítáním obou korekcí zjistíme, že máme počítat na $N = n + \log n + 4$ desetinných míst a to zvládneme v čase $O(n^2/\log n)$.