

**Úloha 4:** Sestavme dva programy, které spočtou normu  $n$ -rozměrného vektoru dle vztahu

$$\|x\|_2 = \left( \sum_{i=1}^n x_i^2 \right)^{1/2}.$$

První program (Program 1) bude počítat normu tak, jak ji vidíme výše. Druhý program (Program 2) bude lepší v tom smyslu, že bude robustní, účinnější a přesnější. Rozdílné výsledky obou implementací budou uvedeny na závěr.

---

#### Program 1 norma\_primo.m

---

```
function norma_primo(x)
format long

suma = 0;
for i = 1:length(x)
    suma = suma + (x(i))^2;
end
norma_primo = sqrt(suma)
```

---

K sestavení druhého programu vyjdeme z následující úpravy normy:

$$\|x\|_2 = 1 \cdot \left( \sum_{i=1}^n x_i^2 \right)^{1/2} = M \cdot \left( \sum_{i=1}^n \frac{x_i^2}{M^2} \right)^{1/2} = M \cdot \left( \sum_{i=1}^n \left( \frac{x_i}{M} \right)^2 \right)^{1/2}, \quad \text{kde } M = \max_i |x_i|.$$

Nyní uveďme několik poznámek k Programu 2.

- Budeme-li vědět, že prvky vektoru jsou nezáporná čísla, pak je moudré v programu psát `ABS = x`. Neboť, zkusíme-li zadat vektor např. délky  $n = 10^7$  (`ABS` ponecháme beze změny) a pustíme-li si v Matlabu Profiler zjistíme, že nejvíce strojového času zabare tato operace (a výpočet), tj. `abs(x)` + operace na určení maxima a minima.
- O funkcích `abs(...)`, `max(...)` a `min(...)` se v nápovědě Matlabu dozvíme akorát to, jak je používat. Bohužel není zde uvedeno, jakým způsobem jsou implementovány<sup>1)</sup>. Jediné co nás v této chvíli napadne je, že se vektor prochází a porovnávají se jednotlivé prvky resp. jsou přeznačovány absolutní hodnotou. Výstupem je pak hodnota maxima, minima či vektor s nezápornými prvky. Stanovení `abs`, `max`, a `min` bychom tedy nahradili jedním průchodem, neboť v našem programu celkem třikrát procházíme vstupní vektor. Tato úprava by vedla ke zrychlení, protože vektor by se nemusel tolikrát procházet. Rovnou bychom ještě prvky vektoru řadili abychom výsledek zpřesnili (sčítali bychom od nejmenších prvků).
- Program je robustní, neboť je schopen pracovat s jakýmkoliv strojově zobrazitelným vektorem na vstupu. To zajišťuje podmínka `if M == Inf` (viz `x1`). Bohužel, v nápovědě Matlabu se toho moc o aktivaci Extended Double Precision nedovíme. (Jde to vůbec?) Touto aktivací bychom se na moment zbavili překročení číselného rozsahu nad hodnotu OFL.

---

<sup>1)</sup>Určitě bychom je našli v nějaké knihovně či v souboru `*.c`, `*.h` apod. jenž Matlab využívá.

- Pokud je vypočtená hodnota normy zobrazitelná jako standardní strojové číslo, pak je „shodná“ s přesnou (teoretickou) hodnotou normy a nedojde k překročení číselného rozsahu nad hodnotu OFL. To je dáno tím, že každý prvek dělíme prvkem maximálním. Tedy každému prvku (číslu) vektoru snížíme velikost exponentu o hodnotu exponentu prvku maximálního. Tím se některé prvky mohou dostat do subnormálních čísel či budou nahrazeny nulou (též ještě umocňujeme na druhou, viz x4). Odtud plyne, že výpočet  $\text{suma} = \text{suma} + (x(i)/M)^2$ ; nemůže přesáhnout hodnotu  $\text{OFL}^2$ . Může se ale stát (viz x2), že maximum bude blízké hodnotě OFL a  $\sqrt{\text{suma}}$  nějaká hodnota taková, že součin  $M \cdot \sqrt{\text{suma}}$  dá hodnotu Inf. Ovšem, to už není strojově zobrazitelné číslo. Jestliže tedy norma není menší nebo rovna hodnotě OFL, tj. bude platit `if norma == Inf`, pak se zobrazí `norma = Inf` a informace o tom, že prvky vektoru jsou příliš velké (viz x2).
- Z podobných (ne-li stejných) úvah jako v předchozím případě plyne, že nemůže nastat případ, kdy norma vektoru s nenulovými složkami bude nulová. Pokud bude splněna podmínka  $m < \text{UFL}$ , pak se zobrazí v command window informace o tom, že vstupní vektor obsahuje alespoň jedno subnormální číslo (viz x4). Může též nastat, že norma bude menší jak hodnota UFL. Pak se zobrazí předchozí hlášení a informace o tom, že hodnota normy je subnormální číslo (viz x5).

vstupní vektor	norma(x)	norma_primo(x)
<code>x1 = [10^309 100 -500]</code>	(1)	Inf
<code>x2 = ones(1,4)*10^308;</code>	Inf (2)	Inf
<code>x3 = ones(1,3)*10^308;</code>	1.732050807568877e+308	Inf
<code>x4 = [10^-310 10^308 10^308];</code>	1.414213562373095e+308 (3)	Inf
<code>x5 = ones(1,4)*10^-308;</code>	2.000000000000000e-308 (3)(4)	0
<code>x6 = ones(1,4)*realmin;</code>	4.450147717014403e-308	0
<code>x7 = ones(1,100)*realmin*eps;</code>	4.940656458412465e-323 (3)(4)	0
<code>x8 = [5 ones(1,10000)*0.2];</code>	20.61552812808732	20.61552812809001
<code>x9 = [ones(1,10000)*0.2 5];</code>	20.61552812808739	20.61552812808969

V command window budete informováni o tom, že:

- (1) Normu nelze stanovit, neboť vektor obsahuje hodnotu INF.
- (2) Složky vektoru jsou příliš velké. Doslo k překročení horní hranice strojových čísel.
- (3) Vektor obsahuje alespoň jedno subnormální číslo.
- (4) Doslo k překročení dolní hranice strojových čísel. Hodnota normy je subnormální číslo.

Přesnější je program 2, protože vypočtené výsledky jsou „blíže“ k teoretickým výsledkům. Matlabem spočtená norma je

$$\|x8\|_2 = \|x9\|_2 = \sqrt{5^2 + 0.2^2 \cdot 10^4} = \sqrt{0.2^2 \cdot 10^4 + 5^2} = 20.61552812808830$$

Pro normu vektoru x8 a x9 je absolutní chyba přibližně rovna  $9 \cdot 10^{-13}$ , kdežto u programu 1 je přibližně  $2 \cdot 10^{-12}$ . Jak již bylo řečeno, sčítáme-li od nejmenších prvků je výsledek přesnější jak ukazuje příklad vektoru x9.

<sup>2)</sup>Vektor o velikosti  $n = 10^{308}$  a víc považuji v současné době s mými vědomostmi za nesmyslné. Proto to v programu nijak zvlášť neošetřuji.

Účinnější je program 2, neboť dává lepší výsledky než program 1. Lepší v tom smyslu, že lépe reaguje na vstupní vektor jak je možné vidět ve výše uvedené tabulce.

---

**Program 2** norma.m

---

```
function norma(x)
%% Pouziti:
%% >> x = [10^-250 0 23 10^-324 89 ... Inf];
%% >> norma(x)

format long

ABS = abs(x);
M = max(ABS);
m = min(ABS);
n = length(x);
UFL = realmin;

if M == Inf
    disp('Normu nelze stanovit, neboť vektor obsahuje hodnotu INF.')
    disp(' ')
elseif M == 0
    norma = M
elseif n == 1
    norma = ABS(1)
else
    if m < UFL
        disp('Vektor obsahuje alespon jedno subnormalni cislo.')
    end

    suma = 0;
    for i = 1:n
        suma = suma + (x(i)/M)^2;
    end
    norma = M*sqrt(suma)

    if norma == Inf
        disp('Slozky vektoru jsou prilis velke. Doslo k prekroceni
            horni hranice strojovych cisel.')
        disp(' ')
    elseif norma < UFL
        disp('Doslo k prekroceni dolni hranice strojovych cisel.
            Hodnota normy je subnormalni cislo.')
        disp(' ')
    end
end
end
```

---