Následující návod je určen studentům předmětu *M5170: Matematické programování* (PřF MU) pro „snadnou“ implementaci numerickým metod nepodmíněné minimalizace s pomocí programů MATLAB a MAPLE v textovém módu, které jsou dostupné na serveru Vader na ÚMS PřF MU (použití na vlastním počítači je samozřejmě také možné). Jedná se o zjednodušenou alternativu k Mapletům, které jsou dostupné na úložišti MapleCloud prostřednictvím odkazů uveřejněných na adrese https://goo.gl/H2fC7P.

Další šíření zdrojových kódů není povoleno (prosím o respektování). Připomínky, náměty, zjištěné nedostatky apod. jsou vítány na adrese zemanekp@math.muni.cz.

NÁVOD (v1.5 /6. září 2017/)

Složku „Metody“ zkopírujte do domovského adresáře na serveru Vader. Tato složka obsahuje zdrojové kódy pro metody jednorozměrné nepodmíněné minimalizace (soubory s příponou `.m`) použitelné v programu MATLAB a zdrojový kód pro metody vícerozměrné nepodmíněné minimalizace (soubor s příponou `.mpl`) použitelné v programu MAPLE.

Přihlaste se na tento server (typicky pomocí programu putty.exe) a přesuňte se do složky Metody (příkazem „`cd Metody`“).

Metody jednorozměrné minimalizace (MATLAB)

Příkazem

```
matlab -nojvm -nodisplay -nosplash
```

spustíte MATLAB v textovém módu (ukončíte jej příkazem `exit`). Zadáním vhodného příkazu můžete otestovat jednotlivé metody.

Metoda prosteho deleni (ekvidistantní)

příklad použití: `ProsteDeleniEKVDST(@(x)7*x^2-4*x+15, 0, 1, 10)`

obecná syntexe: `ProsteDeleniEKVDST(f,a,b,N)`

Metoda prosteho deleni (sudy počet dělení)

příklad použití: `ProsteDeleniSUDE(@(x)7*x^2-4*x+15, 0, 1, 10, 0.1)`

obecná syntaxe: `ProsteDeleniSUDE(f,a,b,N,delta)`

Metoda půlení intervalu

příklad použití: `PuleniIntervalu(@(x)7*x^2-4*x+15, 0, 1, 10, 0.1)` obecná syntaxe: `PuleniIntervalu(f,a,b,N,delta)`

Fibonacciho metoda

příklad použití: `Fibonacci(@(x)7*x^2-4*x+15, 0, 1, 6, 0.05)`

obecná syntaxe: `Fibonacci(f,a,b,N,delta)`

Metoda zlatého řezu

příklad použití: `ZlatyRez(@(x)7*x^2-4*x+15, 0, 1, 10)`

obecná syntaxe: `ZlatyRez(f,a,b,N)`

Význam jednotlivých symbolů v obecné syntaxi

`f` … funkce (musí být zadána v uvedeném tvaru s @, aby byla určena proměnná)
`a` … levý krajní bod výchozího intervalu
`b` … pravý krajní bod výchozího intervalu
`N` … počet vyčíslení (u Metody půlení intervalu je výsledný počet vyčíslení 2N)
`delta` … význam v souladu s odpřednášeným popisem metod (u jednotlivých metod se liší)

Metody vícerozměrné minimalizace (MAPLE)

Příkazem

```
maple2015
```

spustíte MAPLE v textovém módu (ukončíte jej příkazem `quit;`). Nejdříve zadejte příkaz

```
read "Metody/MetodyVRn.mpl";
```

(nebo jinou cestu k souboru MetodyVRn.mpl, nezapomínejte na středník na konci každého příkazu). Nyní můžete importovat jednotlivé metody příkazem

```
with(MetodyVRn);
```

Tímto se definují příkazy MNS, NM a MSG.

Obecná syntaxe jednotlivých metod je stejná. Pro **Metodu sdružených gradientů** je

```
MNS(f, P, N, VypsatIterace,VypsatIteracePodrobne)
```

pro **Newtonovu metodu**

```
NM(f, P, N, VypsatIterace,VypsatIteracePodrobne)
```

pro **Metodu sdružených gradientů ve variantě bez resetování hodnoty beta**

`MSG(f, P, N, VypsatIterace,VypsatIteracePodrobne)`

pro **Metodu sdružených gradientů ve variantě s resetováním hodnoty beta**

`MSGreset(f, P, N, VypsatIterace,VypsatIteracePodrobne)`

Význam jednotlivých symbolů v obecné syntaxi

`f` … funkce s proměnnými ve tvaru `x[1]`, `x[2]` atd.
`P` … počet proměnných
`N` … počet iterací
`VypsatIterace` … formát výpisu řešení (zadejte buď `true` nebo `false`*)*
`VypsatIteracePodrobne` … formát výpisu řešení (zadejte buď `true` nebo `false`*)*

Pro správné fungování metod je také potřeba zadat počáteční aproximaci. Máme-li např. Funkci 3 proměnných a počáteční aproximací je bod `[0,0,0]`, pak je potřeba zadat příkaz

```
VychoziBod[1]:=0; VychoziBod[2]:=0; VychoziBod[3]:=0;
```

Bude-li výpis řešení příliš komplikovaný (velké zlomky aj.), zadejte předchozí příkaz s tečkou za jednotlivými čísly, tj.

```
VychoziBod[1]:=0.; VychoziBod[2]:=0.; VychoziBod[3]:=0.;
```

Budou-li čísla zadána v předchozím tvaru je možné, že Metoda sdružených gradientů nenalezne přesné řešení pro kvadratické formy v konečném počtu kroků z důvodu zaokrouhlování.

Nyní již stačí zadefinovat funkci, např.

```
f:=(3/2)*x[1]^2+2*x[2]^2+(3/2)*x[3]^2+x[1]*x[3]+2*x[2]*x[3]-3*x[1]-x[3];
```

a spustit jednotlivé příkazy

```
MNS(f,3,5,true,false);

NM(f,3,5,true,false);

MSG(f,3,5,true,false);
```

Zkuste také např. Rosenbrockovu funkci

```
f:=100*(x[2]-x[1]^2)^2+(1-x[1])^2;
```

nebo např. (zde je potřeba změnit počáteční aproximaci)

```
f:=10*(x[2]-sin(x[1]))^2+x[1]^2/10;
```

Historie změn

- 20. prosince 2015
  - vytvořeny kódy pro numerické metody jednorozměrné minimalizace (Metoda prostého dělení s ekvidistantní velikostí dělení, Metoda prostého dělení se sudým počtem vyčíslení, Metoda půlení intervalu, Fibonacciho metoda, Metoda zlatého řezu)

- 23. prosince 2015
  - upraven formát čísel ve výpisu v MATLABu
  - opravena chyba ve výpočtu „přesného" řešení pro metody v MATLABu

- 27. prosince 2015
  - vytvořeny kódy pro numerické metody vícerozměrné minimalizace (Metoda největšího spádu, Newtonova metoda a Metoda sdružených gradientů)

- 4. ledna 2016
  - upraven drobný překlep ve výpisu iterací pro Newtonovu metodu
  - přidána Metoda sdružených gradientů s resetováním hodnoty beta (`MSGreset`)

- 6. ledna 2016
  - upraven formát zadávaných funkcí (proměnné ve tvaru `x[i]` místo `t[i]`)

- 12. listopadu 2016
  - formální změny v návodu

- 22. listopadu 2016
  - oprava chyby ve výpočtu apriorního odhadu chyby aproximace pomocí Fibonacciho metody

- 8. dubna 2017
  - vylepšen výpočet jednorozměrné minimalizace pro metody v R^n

- 6. září 2017
  - formální změny v návodu