

MATEMATICKÁ OPTIMALIZACE V INŽENÝRSKÉ PRAXI

Studijní opory

*Ing. Martin Šotola Ph. D.
doc. Ing. Martin Fusek, Ph. D.
doc. Ing. Pavel Maršálek, Ph. D.
Ing. Michal Kořínek, Ph. D.
doc. Ing. Zdeněk Poruba, Ph. D.
Ing. Jakub Cienciala*

Ostrava 2026
Toto dílo je licencováno pod [CC BY 4.0](#)



Financováno
Evropskou unií
NextGenerationEU



NÁRODNÍ
PLÁN OBNOVY

MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY

Obsah

Předmluva	3
1 Úvod do procesu optimalizace	4
1.1 Optimalizace	4
1.2 Matematická formulace optimalizace	5
1.3 Klasifikace optimalizačních úloh	6
1.4 Rozdělení metod řešení optimalizačních problémů	7
1.5 Ukázkové příklady	8
1.6 Cvičení	18
2 Řešení optimalizačních úloh s jednou proměnnou	20
2.1 Vlastnosti funkcí s jednou proměnnou	20
2.2 Formulace optimalizačního problému a počáteční odhad	22
2.3 Metody eliminace	23
2.3.1 Metoda půlení intervalu	23
2.3.2 Metoda zlatého řezu	24
2.4 Metody approximace	25
2.4.1 Kvadratická approximace	26
2.4.2 Powellova metoda – postupná kvadratická approximace	26
2.5 Metody využívající derivace	27
2.5.1 Numerická derivace jedné proměnné	27
2.5.2 Metoda bisekce	28
2.5.3 Metoda sečny	28
2.5.4 Newton-Raphson metoda	29
2.6 Ukázkové příklady	29
2.7 Cvičení	38
3 Řešení optimalizace úlohy s více proměnnými	40
3.1 Vlastnosti funkcí s více proměnnými	40
3.2 Evoluční metoda	42
3.3 Simplexová metoda	43

3.4	Metoda největšího spádu	43
3.5	Metoda sdružených gradientů	44
3.6	Newtonova metoda	45
3.7	Levenberg–Marquardt metoda	45
3.8	Ukázkový příklad	46
3.9	Cvičení	47
4	Vázané optimalizační problémy	49
4.1	Metoda dosazení	49
4.2	Lagrangeova metoda	49
4.3	Karush-Kuhn-Tuckerovy podmínky	50
4.4	Metoda pokutové funkce	51
4.5	Ukázkový příklad	52
4.6	Cvičení	61
	Seznam použitých zdrojů	62

Předmluva

Tento studijní materiál je zaměřen na problematiku matematické optimalizace v inženýrské praxi se zvláštním důrazem na praktickou aplikaci numerických metod při řešení konstrukčního návrhu. Obsah skript je koncipován tak, aby čtenáři poskytl nejen teoretický základ optimalizačních postupů, ale také praktický návod k jejich implementaci a využití při návrhu.

Výklad je systematicky strukturován od základních pojmu optimalizace až po pokročilé metody řešení vázaných i nevázaných optimalizačních problémů. Nedílnou součástí materiálu jsou ukázkové příklady a cvičení, které podporují pochopení probírané problematiky a umožňují její bezprostřední aplikaci v praxi.

Významnou roli v tomto materiálu hraje programovací jazyk Python, který je využíván pro numerické výpočty, vizualizaci výsledků a demonstraci principů optimalizačních algoritmů. Konkrétně jsou využity knihovny jako NumPy, SciPy a Matplotlib, které umožňují efektivní řešení a grafickou interpretaci optimalizačních úloh.

Tento učební a podpůrný učební materiál¹ byl vytvořen v rámci projektu *Podpora zelených dovedností a udržitelnosti na VŠB-TUO* (NPO_VSB-TUO_MSMT-2144/2024-4). Překládaný materiál je určen primárně pro studenty navazujícího magisterského studia studijního oboru *Konstrukce a simulace strojů*, uskutečňovaném na *Fakultě strojní Vysoké školy báňské - Technické univerzity Ostrava*.

Theoretická část a zadání některých příkladů vychází či jsou inspirovány z následujících literárních zdrojů [1, 2, 3]. Poslední verze a řešení některých úloh jsou na GitHub².

Ing. Martin Šotola Ph. D.
doc. Ing. Martin Fusek, Ph. D.
doc. Ing. Pavel Maršíálek, Ph. D.
Ing. Michal Kořínek, Ph. D.
doc. Ing. Zdeněk Poruba, Ph. D.
Ing. Jakub Cienciala

VŠB–Technická univerzita Ostrava, Fakulta strojní, Katedra aplikované mechaniky
1. vydání, Ostrava 2026

¹Je vhodné zmínit, že text tohoto materiálu byl stylizován a gramaticky opraven pomocí umělé inteligence ChatGPT 5.1 (Listopad 2025). Obrázky a Python kódy jsou původní.

²<https://github.com/sot0036/OptimalizaceSkripta>

Kapitola 1

Úvod do procesu optimalizace

Proces konstrukčního návrhu začíná stanovením cílů a požadavků, které je nutné splnit pro zajištění funkčnosti. Návrh počítá s novým řešením, případně má v plánu zlepšit již existující řešení. Proces lze rozdělit do několika kroků: funkční syntéza, parametrická syntéza a kontrola návrhu.

Funkční syntéza připravuje návrh z hlediska požadavků na funkčnost. V případě mechanismu může jít například o požadavek na přenos síly a momentů.

Parametrická syntéza hledá hodnoty důležitých parametrů. Pro jejich nalezení je nutné stanovit pravidla k určení návrhových parametrů. Tato pravidla mohou být formulována jako kritérium kvality rozhodující mezi kandidáty (cílem procesu je obvykle maximalizace či minimalizace) či jako omezení. Příkladem u nádrží může být maximální objem jako cíl, omezením jsou mezní stavy (tj. pevnostní podmínky). Při návrhu je vhodné se rovněž zaměřit na vyrobiteľnost či smontovatelnost.

Tento návrh je tvořen pomocí sady parametrů (např. délky, materiálové vlastnosti, počty dílů, zuby na ozubení atd.). Některé z těchto parametrů jsou pevně dány a nelze je měnit. Po získání parametrů je provedena kontrola z hlediska funkčnosti, ale také posouzení, zda jsou stanovená pravidla dodržena.

Pro získání sady parametrů lze využít jeden ze tří přístupů:

- intuitivní přístup,
- variantní přístup,
- optimalizační přístup.

Intuitivní přístup je založen na intuici a zkušenostech návrháře. Vyžaduje dostatečné zkušenosti a databázi znalostí o řešení daného problému, například navázání na předchozí úspěšný návrh.

Variantní přístup je založen na přípravě několika variant, které mohou být získány jako výsledek intuitivního řešení. Z nich je poté vybrán kandidát (sada parametrů), který splňuje požadavky na kvalitu a omezení. K tomu lze využít výše zmíněná pravidla.

Optimalizační přístup je založen na záměrném hledání sady parametrů tak, aby co nejlépe vyhovovala stanoveným pravidlům. K jejich získání jsou využívány optimalizační metody.

1.1 Optimalizace

Obecně lze optimalizaci označit jako množinu matematických postupů a numerických metod pro hledání a určení nejlepší sady parametrů (kandidáta) ze všech možných variant

bez nutnosti vyčíslení všech alternativ. Tato sada splňuje předem vymezená pravidla. Optimalizace je proto charakterizována pomocí

- cíle, čeho má být dosaženo,
- omezení, která je nutné splnit, aby co nejlépe vyhovovala požadavkům,
- parametru nebo sady parametrů, které jsou hledány,
- omezení pro parametry.

Optimalizační úlohy ve strojírenství lze rozdělit do následujících kategorií:

- konstrukční úlohy – již zmíněný konstrukční návrh, jedná se například o optimalizaci tvaru či topologie,
- dopravní úlohy – přeprava zboží a lidí například z hlediska rychlosti; do této skupiny lze zařadit i distribuční úlohy,
- alokační úlohy – nalezení vhodného rozmístění, například rozložení výrobní linky.

Síla optimalizačních metod při hledání nejlepšího kandidáta ze všech možných případů spočívá ve využití i relativně skromné znalosti matematiky a numerického výpočtu založeného na jasně definovaných algoritmech a procedurách implementovaných pomocí výpočetní techniky. Proto je pro optimalizační algoritmy vhodné převést úlohu do matematické formulace.

1.2 Matematická formulace optimalizace

Na úvod je nutné stanovit základní pojmy. Sada parametrů, které jsou hledány, se nazývá **návrhové proměnné**. Kritérium kvality určující, zda jsou získané návrhové proměnné lepší či horší, se nazývá **cílová funkce**, případně také cenová či účelová funkce, anglicky *objective (price) function*. Tato kvantifikace možných návrhových proměnných je definována tak, aby v extrému poskytovala řešení optimalizační úlohy. Cílová funkce proto vede na úlohu **maximalizace** či **minimalizace**.

Následně omezující podmínky, označované také jako omezující funkce, jsou definovány ve formě **rovností** či **nerovností**. Tato omezení mohou být vztažena přímo k návrhovým proměnným (například minimální tloušťka stěny nádrže) a bývají označovány jako technologická omezení či vedlejší omezení.

Matematická formulace minimalizační úlohy je tedy následující

$$\begin{aligned}
 \text{Návrhové proměnné} & & x_i & \quad \text{pro } i = 1, 2, \dots, N \\
 \text{Cílová funkce} & & \min F(\mathbf{x}) & \\
 \text{Omezujející podmínky} & & g_j \geq 0 \text{ či } g_j > 0 & \quad \text{pro } j = 1, 2, \dots, J \\
 & & h_k = 0 & \quad \text{pro } k = 1, 2, \dots, K
 \end{aligned} \tag{1.1}$$

Omezení návrhových proměnných $x_{i,min} \leq x_i \leq x_{i,max}$,

kde $\mathbf{x} = \{x_1, \dots, x_N\}^T$ se nazývá vektor návrhových proměnných, N je počet návrhových proměnných, J a K určují počet omezujících podmínek ve formě nerovnosti, resp. rovnosti. Z matematického hlediska počet nerovností J nevede na omezení, avšak počet rovností K vede na tři možné stavy:

- $K < N$ – standardní úloha optimalizace,

- $K = N$ – nestandardní úloha optimalizace, řeší se jako soustava K rovnic s N neznámými (vede na jedno řešení, které však nemusí být optimální),
- $K > N$ – přeuročená úloha, nemusí mít řešení.

Cílová funkce může představovat fyzikální veličinu (např. hmotnost, dráhu objektu, polohu, počet šroubů apod.) nebo abstraktní veličinu (např. míru bezpečnosti, vyrobiteľnosti apod.). Funkci lze definovat explicitně či implicitně. Explicitní funkce je vyjádřena pomocí analytických vztahů a implicitní je vyjádřena nepřímo, přičemž řešení je získáno z jiného problému (např. řešení soustavy rovnic, výpočet vlastních čísel, stanovení cyklů apod.).

Tip 1.1. Při využití optimalizačních softwarů je nutné zkontořovat, jak jsou nerovnosti zadány, tj. zda je to „větší než“, „menší než“ či jejich kombinace. Pokud omezující podmínky mají jiný tvar, je nutné je transformovat, například vynásobením -1 :

$$g_j \leq 0 \rightarrow -g_j \geq 0. \quad (1.2)$$

Rovněž optimalizační procedury mohou vypustit omezující podmínu ve tvaru rovnosti $h_k = 0$, protože ji lze nahradit dvojicí nerovností

$$h_k \leq 0 \text{ a } h_k \geq 0. \quad (1.3)$$

Pro úplnost lze problém nerovnosti převést na rovnost zavedením dalšího parametru:

$$\begin{aligned} g_j - y^2 &= 0, \\ g_j - e^y &= 0. \end{aligned} \quad (1.4)$$

U cílové funkce lze rovněž převádět mezi maximalizací a minimalizací pomocí vynásobení -1 , tj. $\max F(x) = -\min[-F(x)]$.

1.3 Klasifikace optimalizačních úloh

Optimalizační úlohy lze dělit podle různých kritérií. V následující části budou představeny jednotlivé klasifikace.

1. Počet návrhových proměnných
 - S jednou návrhovou proměnnou
 - S více návrhovými proměnnými
2. Podle počtu omezujících podmínek
 - Nevázaný problém – pouze cílová funkce a případně omezení návrhových proměnných (nejsou však zahrnuta přímo do výpočtu)
 - Vázaný problém – má alespoň jednu omezující podmínu
3. Podle matematického charakteru cílové funkce a omezujících podmínek
 - Lineární – všechny funkce jsou lineární (pro řešení je nutné mít omezení)
 - Nelineární – alespoň jedna z funkcí je nelineární
4. Podle spojitosti
 - Spojitá optimalizace – návrhová proměnná a funkce jsou spojité, například rozdílová optimalizace nosníku
 - Diskrétní optimalizace – obvykle je kladen důraz na celočíselné výsledky, například počet zubů

5. Podle určitosti

- Deterministická – všechny návrhové proměnné nejsou náhodné, jsou určité
- Stochastická – alespoň jedna návrhová proměnná je náhodná

6. Podle počtu cílových funkcí

- Jednokriteriální optimalizace – jeden požadavek, jedna cílová funkce
- Vícekriteriální optimalizace – několik požadavků, zpravidla vede na sestavení celkové cílové funkce, například jako vážený součet (je nutné pravidla škálovat)

Cílem těchto skript je seznámit čtenáře s nevázanou a vázanou optimalizací s jednou cílovou funkcí, která se minimalizuje. Vícekriteriální optimalizaci však lze převést na vázaný problém s jednou cílovou funkcí, přičemž zbývající kritéria jsou zavedena jako omezující podmínky.

Tip 1.2. V anglické literatuře je optimalizace označována jako programování matematického charakteru. Například lineární optimalizace je označována jako „Linear Programming – LP“ a nelineární jako „Nonlinear Programming – NLP“.

1.4 Rozdělení metod řešení optimalizačních problémů

Výše uvedené úlohy lze řešit zpravidla pomocí tří metod:

1. grafické metody,
2. analytické metody,
3. iterační metody.

Grafické metody spočívají ve vykreslení cílové funkce a doplnění omezujících podmínek. Jsou použitelné pro jednu či dvě návrhové proměnné. Podmínkou je existence funkce v explicitním tvaru; u implicitních funkcí může nastat problém.

Analytické metody mají za cíl formulovat optimalizační úlohu jako soustavu rovnic a následně provést analytické řešení. Lze je použít s omezeními i bez nich.

Iterační (numerické) metody hledají výsledek postupnými kroky. Lze využít čistě numerické řešení či kombinaci numerických postupů a analytických vztahů. Tyto metody lze rozdělit na

- směrové – hledají směr největšího poklesu cílové funkce,
 - approximační – hledají důvěryhodnou approximaci cílové funkce, na jejímž základě je následně hledáno minimum,
 - eliminační – postupně vylučují části zkoumané oblasti, ve kterých neleží minimum.
- Dále lze metody rozdělit dle toho, zda vyžadují derivaci. Tyto metody se dělí na
- metody nultého řádu – využívají pouze hodnoty cílové funkce,
 - metody prvního řádu – využívají funkční hodnoty a první derivaci cílové funkce,
 - metody druhého řádu – využívají funkční hodnoty, první i druhou derivaci cílové funkce.

Metody nultého řádu se využívají v případě, že výpočet derivací je náročný nebo že je numerická derivace nepřesná. To se může stát například při implicitním zadání cílové funkce. Tyto metody lze využít i v případě, že derivace neexistují (například u nespojitých, nehladkých či diskrétních funkcí).

Metody prvního a druhého řádu využívají derivace k určení směru vedoucího k minimu¹, což vede ke zrychlení konvergence. Použití metod druhého řádu je možné tehdy, existuje-

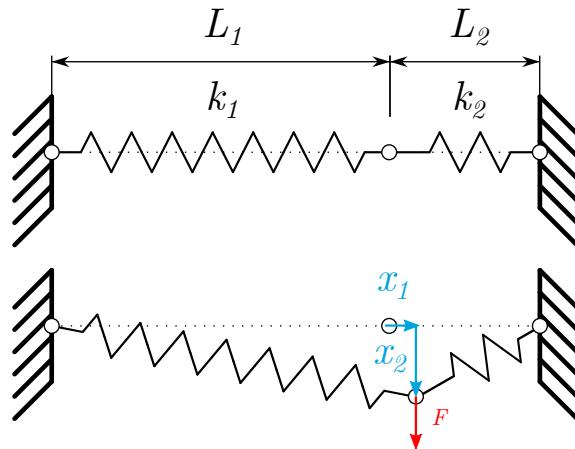
¹Již výše je uvedené, že mezi minimalizací a maximalizací lze převádět pomocí $\max f = -\min f$.

li druhá derivace (alespoň ve zkoumané oblasti). Je nutné podotknout, že efektivnost těchto metod závisí na časové náročnosti získání derivací (analytický výpočet derivace vs. numerická derivace).

1.5 Ukázkové příklady

V této části jsou představeny dvě optimalizační úlohy řešené grafickou metodou. Grafické řešení je připraveno pomocí Python knihovny Matplotlib.

Příklad 1.1. Pomocí grafické metody stanovte minimum celkové potenciální energie následujícího systému, viz Obr. 1.1. Parametry úlohy jsou délka první pružiny $L_1 = 120$ mm, délka druhé pružiny $L_2 = 80$ mm, tuhost první pružiny $k_1 = 10 \frac{\text{N}}{\text{mm}}$, tuhost druhé pružiny $k_2 = 100 \frac{\text{N}}{\text{mm}}$ a síla $F = 10$ N.



Obr. 1.1: Schéma mechanického systému

Řešení. Pro řešení úlohy je nejprve nutné sestavit potenciální energii jednotlivých částí systému a práci vnější síly:

$$\Pi = \frac{1}{2}k_1(\Delta L_1)^2 + \frac{1}{2}k_2(\Delta L_2)^2 - F \cdot x_2, \quad (1.5)$$

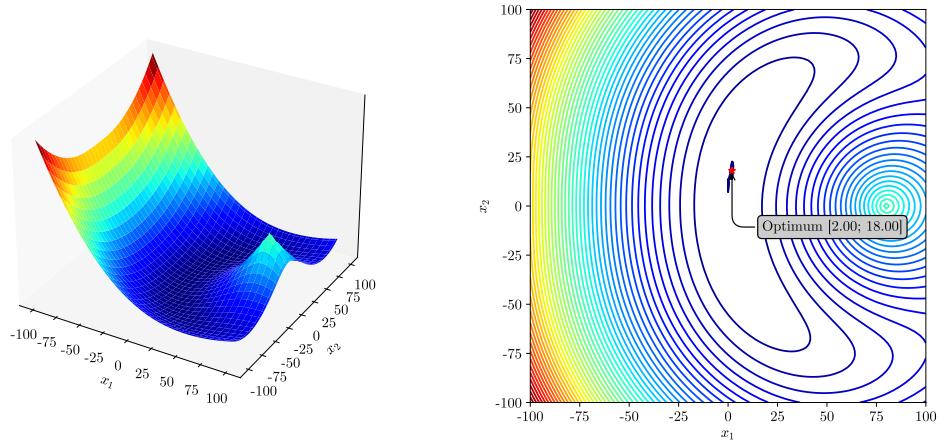
kde ΔL_1 a ΔL_2 představují prodloužení pružin, která jsou definována vztahem

$$\begin{aligned} \Delta L_1 &= \sqrt{(L_1 + x_1)^2 + x_2^2} - L_1, \\ \Delta L_2 &= \sqrt{(L_2 - x_1)^2 + x_2^2} - L_2. \end{aligned} \quad (1.6)$$

Návrhové proměnné x_1 a x_2 představují posuvy prostředního uzlu vlivem působící síly F . Cílová funkce je tedy definována jako

$$\min_{\{x_1, x_2\} \in \mathbb{R}^2} \Pi. \quad (1.7)$$

Protože nejsou předem známy rozsahy, ve kterých se optimum nachází, bylo provedeno několik vykreslení pro různé rozsahy návrhových proměnných. První zvolený rozsah je $x_1 \in \langle -100, 100 \rangle$ a $x_2 \in \langle -100, 100 \rangle$. Tento rozsah byl rozdělen na $N_{x_1} = 101$ a $N_{x_2} = 101$



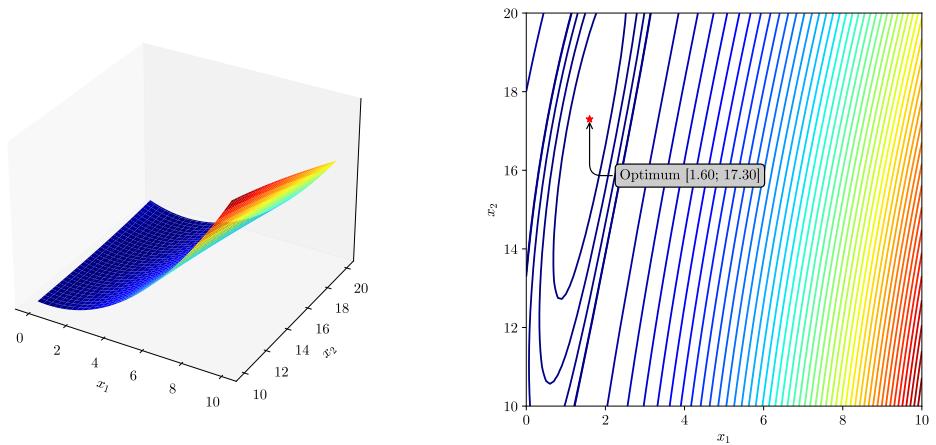
Obr. 1.2: Barevné mapy cílové funkce pro rozsah $x_1 \in \langle -100, 100 \rangle$ a $x_2 \in \langle -100, 100 \rangle$

dílků. Grafické znázornění je uvedeno na Obr. 1.2. Jeden z grafů je zobrazen ve 3D prostoru pro lepší představu o průběhu cílové funkce.

Druhý obrázek je vykreslen pomocí barevných map s viditelnými hranicemi hodnot, které lze interpretovat jako vrstevnice cílové funkce. Minimum se nachází v oblasti kolem bodu $x_1 = 2.0 \pm 2.0$ a $x_2 = 18.0 \pm 2.0$ ².

Vzhledem k tomu, že původní rozsah je velmi široký, je vhodné jej zúžit do oblasti blízké těmto hodnotám. Alternativně lze zvýšit počet dílků v původním rozsahu, avšak tím by byly zbytečně vyhodnocovány také oblasti, ve kterých se optimum s vysokou pravděpodobností nenachází.

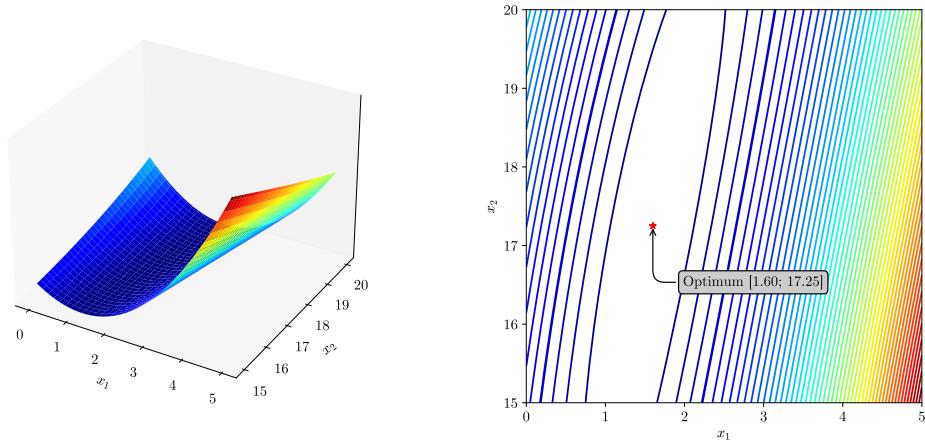
Nově zvolený rozsah je tedy $x_1 \in \langle 0, 10 \rangle$ a $x_2 \in \langle 10, 20 \rangle$, přičemž počet dílků zůstává nezměněn. Výsledek prvního zpřesnění je uveden na Obr. 1.3.



Obr. 1.3: Barevné mapy cílové funkce pro rozsah $x_1 \in \langle 0, 10 \rangle$ a $x_2 \in \langle 10, 20 \rangle$

²Hodnota ± 2.0 vychází z rozdílu rozsahu podčleněho počtem dílků ($N_{x_i} - 1$), tj. $\delta = \frac{x_{i_{\max}} - x_{i_{\min}}}{N_{x_i} - 1}$.

Z obrázku nového rozsahu je patrné, že minimum se nachází v oblasti kolem bodu $\{x_1 = 1.60 \pm 0.10; x_2 = 17.30 \pm 0.10\}$. Poslední zpřesnění bylo provedeno s rozsahem $x_1 \in \langle 0, 5 \rangle$ a $x_2 \in \langle 15, 20 \rangle$, přičemž počet dílků zůstal stejný. Výsledky finálního rozsahu jsou uvedeny na Obr. 1.4.



Obr. 1.4: Barevné mapy cílové funkce pro rozsah $x_1 \in \langle 0, 5 \rangle$ a $x_2 \in \langle 15, 20 \rangle$

Optimum se tedy nachází v $\{x_1 = 1.60 \pm 0.05; x_2 = 17.25 \pm 0.05\}$. Z fyzikálního hlediska se jedná o posuvy prostředního uzlu mechanické soustavy: horizontálně doprava o 1.60 ± 0.05 mm a svisle dolů o 17.25 ± 0.05 mm.

Je však nutné podotknout, že pro získání tohoto výsledku bylo zapotřebí provést přibližně $N_{x_1} \cdot N_{x_2} \approx 10^4$ výpočtů pro každý rozsah. Z toho je patrné, že výpočet zahrnující všechny možné kombinace není efektivní. Tato procedura je implementována v Pythonu, viz kód 1.1, pro rozsah $x_1 \in \langle 0, 10 \rangle$ a $x_2 \in \langle 0, 20 \rangle$.

Kód 1.1: Mechanická soustava, for-cyklus

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
#####
# Reseni pomoci For-cyklu
#####
# Zadani
L1 = 120 #mm
L2 = 80 #mm
k1 = 10 #N/mm
k2 = 100 #N/mm
F = 10 #N

Nx1=100+1 #Pocet dalku pro vykresleni x1
Nx2=100+1 #Pocet dalku pro vykresleni x2
x1 = np.linspace(-100,100,Nx1) #Rovnomerne rozdeleni x1 pomocí Nx1 dalku
```

```

x2 = np.linspace(-100,100,Nx2) #Rovnomerne rozdeleni x2 pomocí
                                Nx2 dílku

x1v,x2v = np.meshgrid(x1,x2) #Priprava mrizky pro výpočet

# Celkova potencialni energie systemu
Energie = np.zeros([Nx2,Nx1])

for i in np.arange(0,Nx1):
    for j in np.arange(0,Nx2):
        Energie[j,i]=1/2*k1*(np.sqrt((L1+x1[i])**2+(x2[j])**2)-
                               L1)**2+\n        1/2*k2*(np.sqrt((L2-x1[i])**2+(x2[j])**2)-L2)**2-\n        F*x2[j]

minimum = np.min(Energie) # maximum na rozsahu mrizky
maximum = np.max(Energie) # minimum na rozsahu mrizky

# Hledani indexu minima potencialni energie
ind = np.unravel_index(np.argmin(Energie, axis=None), Energie.
shape)

# Nastaveni urovne barevne mapy
levels1= np.linspace(minimum,maximum+60-1,5)
levels2= np.linspace(minimum+60,maximum,50)
levels=np.concatenate((levels1,levels2))
# Vypsani vysledku x1 a x2
print(f"Navrhova promena x1={x1[ind[1]]:0.2f}")
print(f"Navrhova promena x2={x2[ind[0]]:0.2f}")

# Vykresleni vysledku jako 3D plocha
fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
surface = ax.plot_surface(x1v,x2v,Energie,cmap=cm.jet)
plt.savefig("Priklad1_1_3D_0.pdf")
# Vykresleni vysledku jako 2D konturova plocha
plt.figure(2)
plt.contour(x1v,x2v,Energie,levels, cmap=cm.jet)

# Vykresleni optima x*
plt.plot(x1[ind[1]],x2[ind[0]],'r*')

plt.savefig("Priklad1_1_2D_0.pdf")

```

Alternativně lze využít maticový výpočet. Připravený kód v prostředí Python je vypsán v kódu 1.2.

Kód 1.2: Mechanická soustava, maticově

```

import numpy as np
import matplotlib.pyplot as plt

```

```

from matplotlib import cm
#####
# Reseni pomoci Maticoveho zapisu
#####
# Zadani
L1 = 120 #mm
L2 = 80 #mm
k1 = 10 #N/mm
k2 = 100 #N/mm
F = 10 #N

Nx1=100+1 #Pocet dalku pro vykresleni x1
Nx2=100+1 #Pocet dalku pro vykresleni x2
x1 = np.linspace(0,10,Nx1) #Rovnomerne rozdeleni x1 pomocí Nx1
dalku
x2 = np.linspace(0,20,Nx2) #Rovnomerne rozdeleni x2 pomocí Nx2
dalku

x1v,x2v = np.meshgrid(x1,x2) #Priprava mrizky pro vypocet

# Celkova potencialni energie systemu
Energie = 1/2*k1*(np.sqrt((L1+x1v)**2+(x2v)**2)-L1)**2
+1/2*k2*(np.sqrt((L2-x1v)**2
+(x2v)**2)-L2)**2-F*x2v

minimum = np.min(Energie) # maximum na rozsahu mrizky
maximum = np.max(Energie) # minimum na rozsahu mrizky

# Hledani indexu minima potencialni energie
ind = np.unravel_index(np.argmin(Energie, axis=None), Energie.
shape)

# Nastaveni urovne barevne mapy
levels= np.arange(minimum, maximum-10, 100)

# Vypsani vysledku x1 a x2
print(f"Navrhova_promena_x1={x1[ind[1]]:0.2f}")
print(f"Navrhova_promena_x2={x2[ind[0]]:0.2f}")

# Vykresleni vysledku jako 3D plocha
fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
surface = ax.plot_surface(x1v,x2v,Energie,cmap=cm.jet)
plt.savefig("Priklad1_1_3D.pdf")
# Vykresleni vysledku jako 2D konturova plocha
plt.figure(2)
plt.contour(x1v,x2v,Energie,levels, cmap=cm.jet)

# Vykresleni optima x*

```

```

plt.plot(x1[ind[1]], x2[ind[0]], 'r*')
plt.savefig("Priklad1_1_2D.pdf")

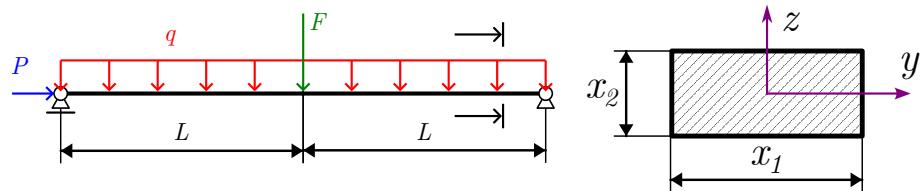
```

Příklad 1.2. Hliníkový nosník o délce $L = 1\ 500$ mm je zatížen spojitym zatížením způsobeným vlastní tíhou q a osamělou silou $F = 5\ 000$ N působící uprostřed nosníku. Kromě toho může být nosník zatížen axiální tlakovou silou $P = 20$ kN.

Cílem úlohy je nalézt rozměry obdélníkového průřezu tak, aby byla hmotnost nosníku minimální, přičemž je nutné dodržet následující omezení:

- napětí nesmí překročit dovolené napětí $\sigma_D = 100$ MPa,
- kritická vzpěrná síla F_{krit} podle Eulera má být vyšší než axiální síla $P = 20$ kN,
- průhyb pod osamělou silou nesmí překročit $w_{fD} = 8$ mm.

Pro řešení uvažujte předpoklad lineární pružnosti. Materiálová data jsou: hustota $\rho_{Al} = 2700 \text{ kg} \cdot \text{m}^{-3}$ a modul pružnosti $E_{Al} = 70\ 000$ MPa. Minimální příčné rozměry nosníku jsou 10 mm. Schéma úlohy je uvedeno na Obr. 1.5.



Obr. 1.5: Schéma úlohy

Řešení. Tento problém je již vázaný, obsahuje tři omezující podmínky (σ , w_F , F_{krit}) a dvě omezení návrhového prostoru (každý rozměr obdélníkového průřezu má být větší než 10 mm). Cílová funkce v tomto případě má tvar³

$$\begin{aligned} \min_{x_1, x_2} m &= \rho \cdot L \cdot x_1 \cdot x_2, \\ x_1 &\geq 10 \text{ mm}, \quad x_2 \geq 10 \text{ mm}, \end{aligned} \tag{1.8}$$

kde x_1 a x_2 jsou návrhové proměnné a reprezentují rozměry příčného průřezu. První omezující podmínka vychází z pevnostní podmínky

$$\sigma_o \leq \sigma_D. \tag{1.9}$$

Ohybové napětí se vypočítá pomocí vztahu

$$\sigma_o = \frac{M_{o_{max}}}{J_z} \cdot y_{max}, \tag{1.10}$$

kde $M_{o_{max}}$ je maximální ohybový moment, J_z je osový moment setrvačnosti kolem osy z a y_{max} definuje nejvzdálenější vlákno od těžiště průřezu ve směru y . Výsledné napětí pro zadaný nosník je

³Pro výpočet je vhodné převést jednotky na jednotný systém, například SI (m, N, kg, Pa, s), nebo na inženýrské jednotky (mm, N, t, MPa, s).

$$\frac{\left(\frac{F}{2} - \frac{q \cdot L}{2}\right) \cdot \frac{L}{2}}{\frac{1}{12} \cdot x_1 \cdot x_2^3} \cdot \frac{x_2}{2} \leq \sigma_D, \quad (1.11)$$

kde q reprezentuje spojité zatížení způsobené vlastní tíhou nosníku

$$q = \rho \cdot g \cdot x_1 \cdot x_2. \quad (1.12)$$

Další omezující podmínkou je průhyb uprostřed nosníku w_F , který má být menší než jeho dovolená hodnota w_D . Pro výpočet lze využít připravené vzorce ze strojnických tabulek a principu superpozice

$$w_F = \frac{F \cdot L^3}{48 \cdot E \cdot J_z} + \frac{5 \cdot q \cdot L^4}{384 \cdot E \cdot J_z} \leq w_D. \quad (1.13)$$

Další Eulerova kritická síla při vzpěru je dána jako

$$F_{krit} = \frac{\pi^2 \cdot E \cdot J_{min}}{L^2}, \quad (1.14)$$

kde J_{min} je minimální hlavní moment setrvačnosti. Zde je nutné rozhodnout, který ze dvou směrů má menší moment setrvačnosti⁴

$$J_{min} = \min \left(\frac{x_1^3 \cdot x_2}{12}, \frac{x_1 \cdot x_2^3}{12} \right). \quad (1.15)$$

Podmínu lze využít pro grafickou metodu bez změny, nicméně pro výpočet jinými optimizačními metodami je nutné ji rozdělit na dvě

$$\begin{aligned} \frac{\pi^2 \cdot E \cdot \frac{x_1^3 \cdot x_2}{12}}{L^2} &\geq P, \\ \frac{\pi^2 \cdot E \cdot \frac{x_1 \cdot x_2^3}{12}}{L^2} &\geq P. \end{aligned} \quad (1.16)$$

Řešení grafické metody je na obrázku 1.6. Hladiny kontur určují hmotnost pro dané rozměry. Dále jsou vykresleny hranice omezujících podmínek.

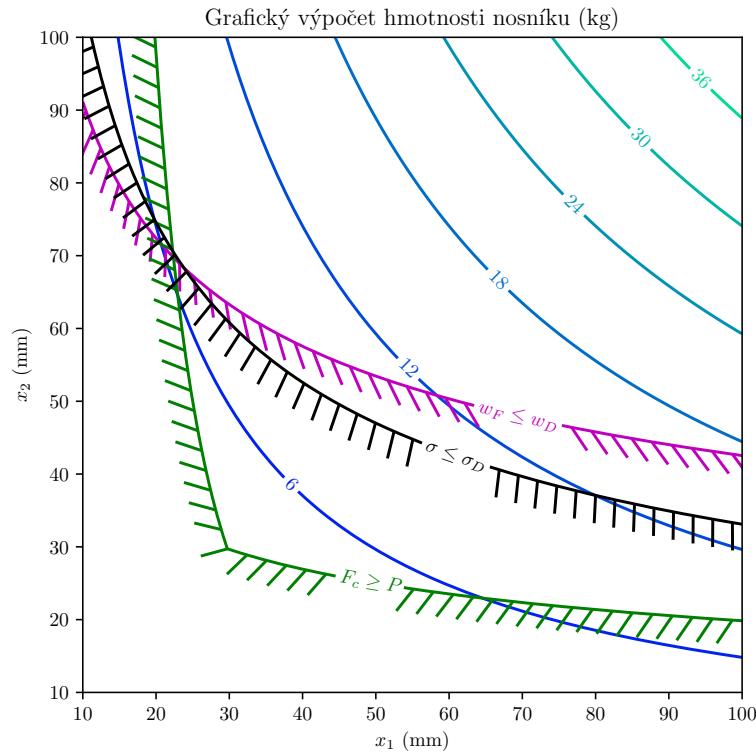
Optimum se tedy nachází v oblasti kolem $x_1 = 25$ mm a $x_2 = 70$ mm, jak je zobrazeno na Obr. 1.7a. Pro zpřesnění lze využít také přímku na Obr. 1.7b, která má směrnici $x_2/x_1 = 3$. Tato přímka byla odvozena postupným vykreslením přímek pro různé kandidáty.

Pro řešení je vybrán⁵ kandidát s parametry $x_1 = 23.3$ mm a $x_2 = 70$ mm. Pro tohoto kandidáta jsou hodnoty cílové funkce a omezujících funkcí následující:

$$\begin{aligned} m &= 6.605 \text{ kg}, \\ w_F &= 7.6 \text{ mm}, \\ F_{krit} &= 22\,657 \text{ N}, \\ \sigma_o &= 97.2 \text{ MPa}. \end{aligned} \quad (1.17)$$

⁴Průřez je symetrický a deviační moment setrvačnosti vychází 0 mm⁴, proto je snadné určit hlavní momenty setrvačnosti.

⁵Vybraný bod byl zvolen na základě zkušeností, nikoliv automaticky z procesu optimalizace.



Obr. 1.6: Grafické řešení vázaného problému

Níže je pro řešení a vykreslení vypsán kód 1.3.

Kód 1.3: Řešení vázaného problemu pomocí grafické metody

```

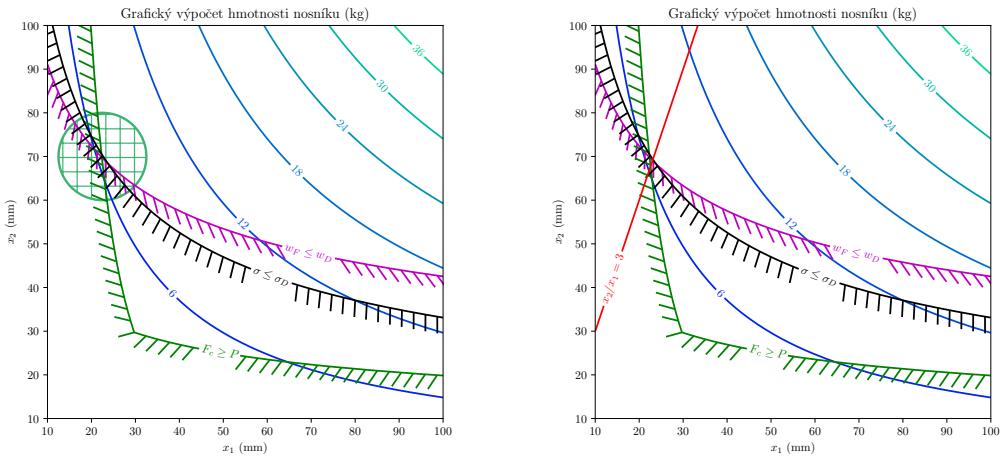
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib import patheffects

#####
#Zadani
#####

L=1500 # mm
E= 70000 # MPa
rho = 2.7*10**-9 # t/m^3
F = 5000 # N
g = 9810 # mm/s^2

#####
#Omezeni
#####
x1min = 10 # mm

```



(a) Oblast pomocí kruhu

(b) Oblast pomocí přímky

Obr. 1.7: Oblast možného řešení vázaného problému

```

x2min = 10 # mm

w_D=8 #mm
sigmaD= 100 # MPa
P = 20000 # N
#####
Nx1 = 101 # Pocet dílk u x1
Nx2 = 101 # Pocet dílk u x2

x1 = np.linspace(x1min,100,Nx1) #Rovnoměrné rozdelení x1 pomocí
#Nx1 dílku
x2 = np.linspace(x2min,100,Nx2) #Rovnoměrné rozdelení x1 pomocí
#Nx1 dílku

#####
# Priprava matic pro hodnoty cílové funkce a omezující funkce
#####
m = np.zeros([Nx2,Nx1])
w_F = np.zeros([Nx2,Nx1])
F_c = np.zeros([Nx2,Nx1])
sigma = np.zeros([Nx2,Nx1])

#####
# Vypocet jednotlivých kandidátů
#####

for i in np.arange(0,Nx1):
    for j in np.arange(0,Nx2):
        S = x1[i]*x2[j]
        m[j,i] = rho*L*S

```

```

q = rho*g*S
Mo = (F/2+q*L/2)*L/2-q*L**2/2
Jz = 1/12*x1[i]*x2[j]**3
Jy = 1/12*x1[i]**3*x2[j]
w_F[j,i] = (F*L**3)/(48*E*Jz) + (5*q*L**4)/(384*E*Jz)
F_c[j,i] = np.pi**2*E*min([Jz,Jy])/L**2
sigma[j,i] = Mo/Jz * x2[j]/2

#####
# Vykresleni vysledku
#####
x1v,x2v = np.meshgrid(x1,x2) # priprava mrizky navrhovych
# promennych
fig, ax = plt.subplots(figsize=(6, 6)) # priprava ramecku
# obrazku

# Vykresleni cilove funkce a vypsani hladin
objective = ax.contour(x1v,x2v,m*1000, cmap=cm.winter)
ax.clabel(objective, objective.levels, manual=[(30,30),(50,50),
       (70,70),(80,80),(90,90),(100,100)])

# Vykresleni hladiny w_D
pw_f = ax.contour(x1v,x2v,w_F,[w_D], colors="m")
pw_f.set(path_effects=[patheffects.withTickedStroke(angle=135)])
ax.clabel(pw_f,pw_f.levels,fmt="$w_F \leq w_D$")

# Vykresleni hladiny P
pF_c= ax.contour(x1v,x2v,F_c,[P], colors='g')
pF_c.set(path_effects=[patheffects.withTickedStroke(angle=240)])
ax.clabel(pF_c,pF_c.levels,fmt="$F_c \geq P $",manual=[(50,30)])

# Vykresleni hladiny |sigma
pSigma = ax.contour(x1v,x2v,sigma,[sigmaD], colors='k')
pSigma.set(path_effects=[patheffects.withTickedStroke(angle=100)])
ax.clabel(pSigma,pSigma.levels,fmt="$|\sigma| \leq \sigma_D$",
       manual=[(60,40)])

# Priprava popisu obrazku
plt.xlabel("x_1(mm)")
plt.ylabel("x_2(mm)")

#Ulozeni obrazku
plt.savefig("Pr12.pdf")

```

1.6 Cvičení

Cvičení 1.1. Pro zadanou funkci je nutné nalézt extrémy funkce jedné proměnné na rozsahu $x \in \langle -2, 10 \rangle$ pomocí grafické metody:

$$f(x) = \frac{x^4}{(x-1) \cdot (x-3)^3}.$$

Cvičení 1.2. Pro zadanou funkci je nutné nalézt extrémy funkce jedné proměnné na rozsahu $x \in \langle -4, 4 \rangle$ pomocí grafické metody:

$$f(x) = x^3 - 12 \cdot x + 3.$$

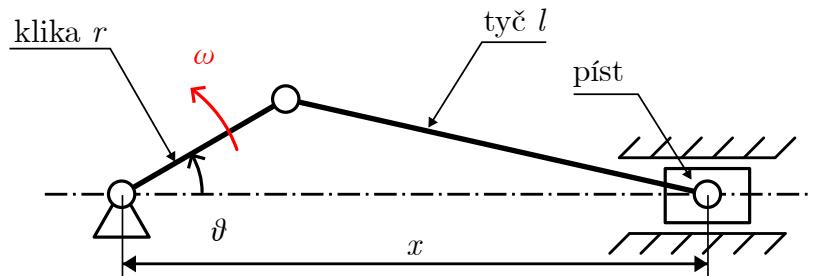
Cvičení 1.3. Pro zadanou funkci je nutné nalézt extrémy funkce jedné proměnné na rozsahu $x \in \langle -4, 4 \rangle$ pomocí grafické metody:

$$f(x) = 3 \cdot x^2 + \frac{12}{x^3 + 1} - 5.$$

Cvičení 1.4. Pro zadanou funkci je nutné nalézt extrémy funkce dvou proměnných na rozsahu $x_i \in \langle -10, 10 \rangle$ pomocí grafické metody:

$$f(x) = 2 \cdot x_1^2 - \frac{x_2^2}{5} + x_1 \cdot x_2 + 5 \cdot x_1 - 8 \cdot x_2.$$

Cvičení 1.5. Klikový mechanismus (viz Obr. 1.8) má konstantní úhlovou rychlosť kliky $\omega = 100$ rad/s. Cílem je nalézt délky kliky r a spojovací tyče l pomocí grafické metody, tak aby při úhlu $\vartheta = 30^\circ$ vykazoval píst maximální rychlosť v_p . Pro zajištění možnosti provést plnou otáčku platí $l \geq 2.5 \cdot r$. Další omezení jsou: délka kliky $0.5 \leq r \leq 10$ m, délka spojovací tyče $2.5 \leq l \leq 25$ m a délka mezi kloubem a pístem při úhlu $\theta = 30^\circ$ $10 \leq x \leq 20$ m.



Obr. 1.8: Schéma klikového mechanismu

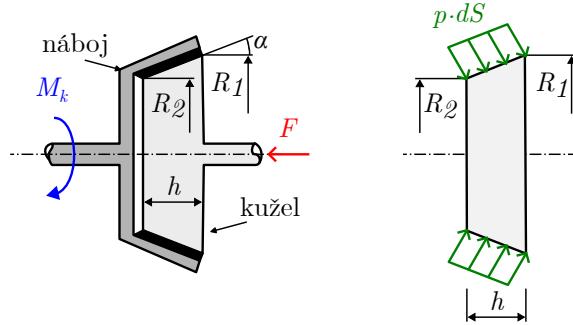
Cvičení 1.6. Klikový mechanismus z úlohy 1.5 má nový cíl: nalézt pomocí grafické metody délky kliky r a spojovací tyče l , tak aby při úhlu $\theta = 30^\circ$ měl píst maximální zrychlení a_p .

Cvičení 1.7. Krouticí moment přenášený kuželovou spojkou (viz Obr. 1.9) při rovnoměrném tlaku je dán vztahem:

$$M_k = \frac{2\pi f p}{3 \sin \alpha} (R_1^3 - R_2^3),$$

kde p je tlak mezi kuželem a nábojem, f je součinitel tření, α je úhel kuželet, R_1 je vnější poloměr a R_2 vnitřní poloměr. Cílem je nalézt poloměry R_1 a R_2 tak, aby objem kuželetu byl minimální.

Parametry kuželetu jsou: úhel $\alpha = 30^\circ$, přítlačná síla $F = 5000$ N, součinitel tření $f = 0.5$. Omezení jsou: kroutící moment $M_k \geq 250$ Nm a technologická omezení $R_1 \geq 2 \cdot R_2$, $0 \leq R_1 \leq 100$ mm.



Obr. 1.9: Schéma kuželové spojky

Kapitola 2

Řešení optimalizačních úlohy s jednou proměnnou

Optimalizační úlohy, kdy cílová funkce závisí na jedné proměnné, patří mezi základní typy optimalizačních úloh. Přesto mají významný teoretický základ nejen kvůli jejich častému výskytu v inženýrských aplikacích, ale také proto, že problémy s jednou proměnnou se typicky využívají jako subproblémy v rámci iterační optimalizace s více proměnnými.

2.1 Vlastnosti funkcí s jednou proměnnou

Funkce $f(x)$ je obecně pravidlo, které každému x přiřazuje hodnotu $y = f(x)$. Zde x označuje nezávislou proměnnou a y závislou proměnnou. Pokud nezávislá proměnná patří do množiny reálných čísel $x \in \mathbb{R}$, lze funkci $f(x)$ nazvat *skalárni funkci* a také neomezenou funkcí. Pokud však $x \in (\mathbb{R} \cap \mathbb{S})$, kde \mathbb{S} je podoblast reálných čísel, jedná se o funkci s omezením. Například:

$$f(x) = x^3 + 2 \cdot x^2 - x + 3, \quad x \in \mathbb{R} \quad (2.1)$$

je neomezená funkce, zatímco

$$f(x) = x^3 + 2 \cdot x^2 - x + 3, \quad x \in \mathbb{S} = \{x \mid -5 \leq x \leq 5\} \quad (2.2)$$

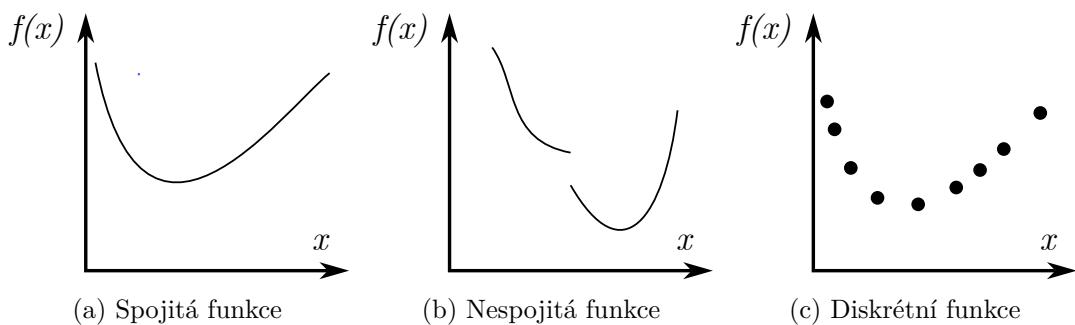
je funkce omezená. V rámci optimalizace se funkce $f(x)$ obvykle označuje jako *cílová funkce*, x je *návrhová proměnná* a \mathbb{S} představuje *oblast zájmu* (či definiční obor).

Funkce může mít různý charakter, jak je znázorněno na Obr. 2.1

- spojitá – tato funkce je spojitá v oblasti zájmu, viz Obr. 2.1a,
- nespojitá – tato funkce není spojitá někde v oblasti zájmu, viz Obr. 2.1b,
- diskrétní – není spojitá nikde v oblasti zájmu, viz Obr. 2.1c.

Poznámka 2.1.1. Je důležité zopakovat vlastnosti spojitých funkcí. Sčítání a násobení spojitých funkcí zachovává spojitost. V případě dělení dvou spojitých funkcí je výsledná funkce spojitá až na místa, kde jmenovatel není definován (tj., kde dochází k dělení nulou).

Je patrné, že volba řešení závisí nejen na charakteru funkce, ale také na charakteru návrhové proměnné x . Je proto nutné zvolit vhodnou metodu optimalizace pro hledání optima. Některé optimalizační procedury jsou připraveny pro diskrétní funkce, jiné pouze pro spojité.



Obr. 2.1: Charaktery funkcí jedné proměnné

Další důležitou vlastností je *monotónnost*. Funkce $f(x)$ je monotónní (rostoucí nebo klesající), pokud pro dva různé body x_1 a x_2 z návrhového prostoru s podmínkou $x_1 < x_2$ platí:

$$\begin{aligned} f(x_1) &< f(x_2) \quad (\text{monotónně rostoucí}), \\ f(x_1) &> f(x_2) \quad (\text{monotónně klesající}). \end{aligned} \tag{2.3}$$

Je třeba poznamenat, že funkce nemusí být spojitá, aby byla monotónní.

Dalším důležitým pojmem je *unimodální funkce*. Funkce $f(x)$ je unimodální na intervalu $a \leq x \leq b$, pokud je monotónní z obou stran kolem jediného bodu x^* , který představuje extrém. Například pokud x^* reprezentuje minimum funkce $f(x)$, pak pro dva libovolné body x_1 a x_2 z intervalu $a \leq x \leq b$ platí

$$x^* \leq x_1 \leq x_2 \text{ znamená } f(x^*) \leq f(x_1) \leq f(x_2) \quad (2.4)$$

a

$$x^* \geq x_1 \geq x_2 \text{ znamená } f(x^*) \leq f(x_1) \leq f(x_2). \quad (2.5)$$

Pro funkce uvedené na Obr. 2.1 platí, že se jedná o unimodální funkce, protože vyhovují výše uvedeným pravidlům.

Další důležité pojmy jsou *globální* a *lokální extrémy*. Funkce $f(x)$, $x \in \mathbb{S}$ má globální minimum v bodě $x^{**} \in \mathbb{S}$ právě tehdy, když platí

$$f(x^{**}) \leq f(x) \quad \forall x \in \mathbb{S}. \quad (2.6)$$

Funkce $f(x)$, $x \in \mathbb{S}$ má lokální minimum v bodě $x^* \in \mathbb{S}$ právě tehdy, když platí

$$f(x^*) \leq f(x) \quad \text{for } x \in [x^* - h, x^* + h], \quad (2.7)$$

kde existuje „šířka“ $h > 0$. Z toho vyplývají následující poznámky:

1. Otočením nerovností vznikají analogické vztahy pro globální a lokální maximum.
 2. V případě unimodální funkce je lokální minimum automaticky globálním minimem.
 3. Pokud funkce není unimodální, existuje několik lokálních optim a globální minimum se nalezne výběrem minima ze všech lokálních optim (tj. u minima je globální minimum to lokální minimum s nejmenší hodnotou).

Následuje určení optima funkce s jednou proměnnou. Funkce $f(x)$ je definována na otevřeném intervalu (a, b) a je diferencovatelná až do n -tého řádu. Pokud x^* je bod z tohoto intervalu, platí pro Taylorův rozvoj funkce v bodě $x^* + h$:

$$f(x^* + h) - f(x^*) \approx h \cdot \frac{df}{dx} \Big|_{x=x^*} + \frac{h^2}{2!} \cdot \frac{d^2 f}{dx^2} \Big|_{x=x^*} + \cdots + \frac{h^n}{n!} \cdot \frac{d^n f}{dx^n} \Big|_{x=x^*}. \quad (2.8)$$

Pokud x^* je lokální minimum na intervalu (a, b) , pak z definice okolí h optima x^* platí pro libovolné $x \in [x^* - h, x^* + h]$:

$$f(x) \geq f(x^*). \quad (2.9)$$

Z výše uvedené nerovnosti vyplývá:

$$h \cdot \frac{df}{dx} \Big|_{x=x^*} + \frac{h^2}{2!} \cdot \frac{d^2 f}{dx^2} \Big|_{x=x^*} + \cdots + \frac{h^n}{n!} \cdot \frac{d^n f}{dx^n} \Big|_{x=x^*} \geq 0. \quad (2.10)$$

Pro dostatečně malé h má první člen dominantní podíl a h může být kladné i záporné, takže nerovnost platí tehdy, když

$$\frac{df}{dx} \Big|_{x=x^*} = 0. \quad (2.11)$$

Tato rovnice se vztahuje k hledání stacionárního bodu. Pro další člen rozvoje platí opět nerovnost

$$\frac{d^2 f}{dx^2} \Big|_{x=x^*} \geq 0. \quad (2.12)$$

Pro lokální maximum platí vztahy obdobně, pouze je nerovnost (2.12) opačná. Vztahy (2.11) a (2.12) se nazývají nutná podmínka pro lokální minimum (resp. maximum), pokud je funkce dvakrát diferencovatelná. Pokud jedna z podmínek není splněna, nejedná se o optimum; pokud jsou obě splněny, neznamená to automaticky, že jde o optimum.

Tip 2.1. Podmínka stacionarity sama neurčuje, zda se jedná o optimum. Pro určení je zapotřebí znát hodnotu druhé derivace (resp. derivace sudého rádu). Pokud je tato derivace kladná, jedná se o minimum; pokud záporná, jde o maximum.

Poznámka 2.1.2. Výše uvedené podmínky platí pro spojitou funkci. V případě nespojitých funkcí však nemusí být platné a je vhodné připravit odpovídající nové podmínky.

2.2 Formulace optimalizačního problému a počáteční odhad

Po stanovení důležitých pojmu a vlastností je vhodné připravit formulaci optimalizace. Cílem následujícího problému je minimalizace cílové funkce $f(x)$ s případnými omezujícími podmínkami. Problém je stanoven jako

$$\min_x f(x), \quad (2.13)$$

$$g_j \geq 0, \quad (2.14)$$

$$x_{\min} \leq x \leq x_{\max}. \quad (2.15)$$

Omezující podmínky ve formě rovnosti se u optimalizace jedné proměnné obvykle neuplatňují. Omezující podmínky ve formě nerovnosti lze převést tak, aby omezovaly návrhový prostor. Proto je ve skutečnosti optimalizační problém definován jako

$$\begin{aligned} & \min_x f(x), \\ & x_{\min} \leq x \leq x_{\max}. \end{aligned} \quad (2.16)$$

Pro následující metody je vhodné určit rozsah (a, b) z intervalu $x_{\min} \leq x \leq x_{\max}$ tak, aby funkce byla unimodální a aby se v rozsahu nacházelo optimum. Rozsah lze obvykle stanovit heuristicky, případně lze využít extrapolacní metody. Pomocí Swannova metoda (anglicky *Swann's bracketing method*) se vypočítá $(k + 1)$ -tý bod jako

$$x_{k+1} = x_k + 2^k \Delta, \text{ pro } k = 0, 1, 2, \dots, \quad (2.17)$$

kde x_0 je počáteční bod a Δ je parametr kroku vhodné velikosti. Znaménko parametru Δ se určí podle hodnot cílové funkce $f(x_0)$, $f(x_0 + |\Delta|)$ a $f(x_0 - |\Delta|)$. Pro

$$f(x_0 - |\Delta|) \geq f(x_0) \geq f(x_0 + |\Delta|) \quad (2.18)$$

je parametr $\Delta > 0$. Předpokládá se, že funkce je unimodální, a proto se optimum nachází v pravém regionu od x_0 . Pro

$$f(x_0 - |\Delta|) \leq f(x_0) \leq f(x_0 + |\Delta|) \quad (2.19)$$

je parametr $\Delta < 0$. Stejný předpoklad platí, takže optimum je doleva od x_0 . Pokud však nastane

$$f(x_0 - |\Delta|) \geq f(x_0) \leq f(x_0 + |\Delta|), \quad (2.20)$$

znamená to, že minimum leží mezi hodnotami $x_0 - |\Delta|$ a $x_0 + |\Delta|$ a hledání intervalu se zastavuje. Zbývá případ

$$f(x_0 - |\Delta|) \leq f(x_0) \geq f(x_0 + |\Delta|), \quad (2.21)$$

který je vyloučen z důvodu unimodální funkce. Pokud by však tento případ nastal při minimalizaci, znamená to, že funkce není unimodální v okolí x_0 .

Poté se vypočítají jednotlivé hodnoty $f(x_k)$, dokud nedojde k nárustu hodnoty funkce. Pokud k nárustu dojde, znamená to, že mezi zkoumanými body leží minimum a volí se interval $(a, b) = (x_{k-1}, x_{k+1})$ ¹.

2.3 Metody eliminace

Po nalezení intervalu (a, b) je vhodné tento interval postupně zredukovat tak, aby optimum leželo v rozsahu $x^* \pm \delta$, kde parametr δ stanovuje interval nejistoty. Tento parametr se volí jako dostatečně malé číslo.

2.3.1 Metoda půlení intervalu

Tato metoda redukuje interval přibližně na polovinu. V anglické literatuře se také označuje jako *three-point equal-interval*, protože optimum se hledá pomocí tří rovnoměrně rozložených bodů. Postup je následující:

Krok 1. Vypočítá se $x_M = \frac{1}{2} \cdot (a + b)$ a $L = b - a$. Dále se stanoví $f(x_M)$.

Krok 2. Získají se $x_L = a + \frac{1}{4} \cdot L$ a $x_P = b - \frac{1}{4} \cdot L$. Stanoví se $f(x_L)$ a $f(x_P)$.

Krok 3. Porovnají se hodnoty $f(x_L)$ a $f(x_M)$.

(a) Pokud $f(x_L) < f(x_M)$, je eliminován interval (x_M, b) nastavením $b = x_M$ a pokračuje se na Krok 5.

¹Lze využít i počáteční odhad x_0 a x_{k+1} , ale toto zvětšuje oblast zájmu.

(b) Pokud $f(x_L) \geq f(x_M)$, pokračuje se na Krok 4.

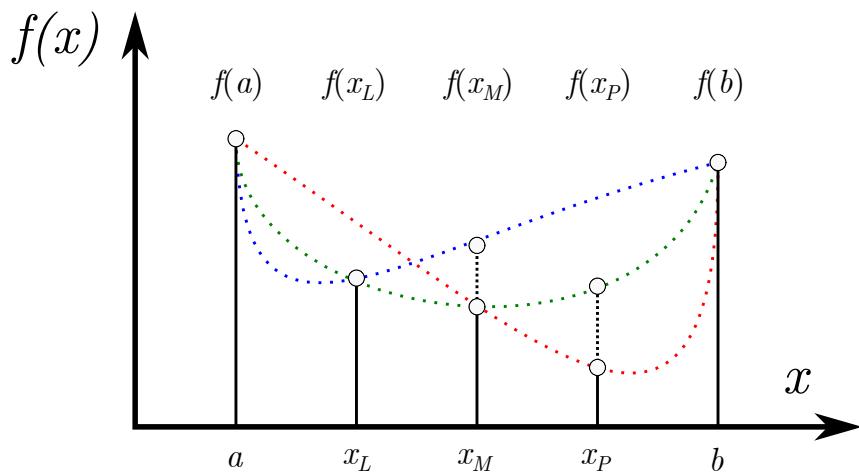
Krok 4. Porovnají se hodnoty $f(x_P)$ a $f(x_M)$.

(a) Pokud $f(x_P) < f(x_M)$, je eliminován interval (a, x_M) nastavením $a = x_M$ a pokračuje se na Krok 5.

(b) Pokud $f(x_P) \geq f(x_M)$, jsou eliminovány intervaly (a, x_L) a (x_P, b) a nové limity jsou $a = x_L$ a $b = x_P$. Pokračuje se na Krok 5.

Krok 5. Spočítá se nový interval $L = b - a$. Pokud $|L| < 2\delta$, vypočítá se $x_M = \frac{1}{2}(a + b)$. Pokud není podmínka splněna, návrat na Krok 1.

Výše uvedenou proceduru ilustruje obrázek 2.2. Tečkované čáry znázorňují možnou podobu funkce na daném úseku. Modrá křivka předpokládá minimum v intervalu $x^* \in (a, x_M)$, červená v intervalu $x^* \in (x_M, b)$ a zelená v intervalu $x^* \in (x_L, x_P)$. Metoda předpokládá existenci minima, avšak vzhledem k tomu, že výpočet probíhá pouze v omezeném počtu bodů, zpřesnění se dosáhne opakováním redukce.



Obr. 2.2: Znázornění procedury metody půlení intervalu

Poznámka 2.3.1. Procedura, jak je stanovena, vždy zredukuje interval přibližně na polovinu. Je vhodné ukládat hodnoty proměnných, protože střední bod x_M při nové iteraci vychází často stejně jako předchozí body x_L či x_P . Proto je potřeba počítat hodnoty cílové funkce pouze pro dvě nové hodnoty.

2.3.2 Metoda zlatého řezu

Výše uvedená metoda počítá s výpočtem alespoň tří hodnot funkce pro tři body. Existuje však přístup, kdy je zapotřebí spočítat pouze dvě hodnoty. Tyto body nejsou rozloženy rovnoměrně, ale nacházejí se od každého konce intervalu (a, b) ve vzdálenosti φ . Symetrické rozdělení intervalu vede k bodům

$$x_L = b - (b - a) \cdot \varphi, \quad x_P = a + (b - a) \cdot \varphi. \quad (2.22)$$

Po nalezení vhodného bodu se proces opakuje a interval se redukuje o hodnotu $1 - \varphi$. Parametr φ je určen z rovnice

$$1 - \varphi = \varphi^2, \quad (2.23)$$

její řešení je

$$\varphi = \frac{-1 \pm \sqrt{5}}{2}, \quad (2.24)$$

kde pozitivní komponenta je $\varphi \approx 0.618$. Toto číslo představuje tzv. *zlatý řez* (v umění také *božský poměr*). Proto se metoda nazývá *metoda zlatého řezu*.

Po n iteracích je rozsah zmenšen na $(1 - \varphi)^{n-1}$ a výsledek se nachází v intervalu

$$L_n = L_0 \cdot \varphi^{n-1}, \quad (2.25)$$

kde L_0 je výchozí délka intervalu. Postup je následující:

Krok 1. Vypočítají se body $x_L = b - (b - a) \cdot \varphi$ a $x_P = a + (b - a) \cdot \varphi$. Stanoví se hodnoty $f(x_L)$ a $f(x_P)$.

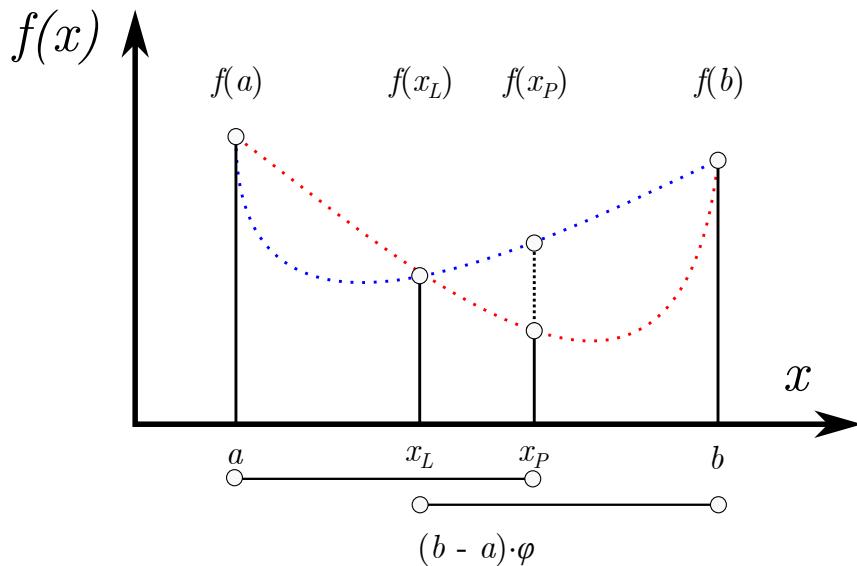
Krok 2. Porovnají se hodnoty $f(x_L)$ a $f(x_P)$.

- (a) Pokud $f(x_L) < f(x_P)$, je eliminován interval (x_P, b) nastavením $b = x_P$ a pokračuje se na Krok 3.
- (b) Pokud $f(x_L) > f(x_P)$, je eliminován interval (a, x_L) nastavením $a = x_L$ a pokračuje se na Krok 3.

Krok 3. Stanoví se nový interval $L = b - a$. Pokud $|L| < 2\delta$, vypočítá se $x_M = \frac{1}{2}(a + b)$.

Pokud není podmínka splněna, návrat na Krok 1.

Výše uvedenou proceduru ilustruje obrázek 2.3. Tečkované čáry znázorňují možnou podobu funkce na daném úseku. Modrá křivka předpokládá minimum v intervalu $x^* \in (a, x_P)$ a červená v intervalu $x^* \in (x_L, b)$. Metoda předpokládá existenci minima, avšak vzhledem k omezenému počtu výpočtů se zpřesnění dosáhne opakováním redukce.



Obr. 2.3: Znázornění procedury metody zlatého řezu

2.4 Metody approximace

Metody z předešlé kapitoly mají pouze jednu podmítku – funkce má být unimodální. Mohou být aplikovány na spojité, nespojité či diskrétní návrhové proměnné. Celé hledání optima je založeno na prostém srovnání dvou bodů. Avšak velikost rozdílu hodnot cílové funkce mezi těmito body není využita, což omezuje efektivitu metody.

Metody approximace tuto informaci využívají a obecně dosahují vyšší efektivity než metody eliminace. Je však nutné, aby funkce byla dostatečně hladká. Základní myšlenka je,

že hladkou funkci lze nahradit polynomem, který lze použít pro nalezení extrému. Funkce má být tedy unimodální a spojitá. Vybráním vhodného polynomu dostatečného rádu lze na intervalu (a, b) reprezentovat cílovou funkci a optimum polynomu pak approximuje optimum skutečné funkce. Vyšší řády polynomů však zvyšují složitost řešení, proto je vhodné volit kvadratickou approximaci.

2.4.1 Kvadratická approximace

Nejjednodušší approximace je kvadratická. Pokud se minimum nachází uvnitř intervalu, nahradní funkce má být alespoň kvadratická – lineární funkce by totiž umístila optimum na krajích intervalu. Kvadratická approximace tedy umožňuje hledat minimum, přičemž s každou iterací se approximace zlepšuje a blíží ke skutečnému minimu x^* .

Pro tři po sobě jdoucí body x_1, x_2, x_3 a korespondující hodnoty cílové funkce f_1, f_2, f_3 se stanoví konstanty a_0, a_1 a a_2 tak, aby byla připravena kvadratická funkce

$$q(x) = a_0 + a_1 \cdot (x - x_1) + a_2 \cdot (x - x_1)(x - x_2), \quad (2.26)$$

která prochází body $[x_i, f_i]$. Parametry se určují následujícím způsobem:

$$f_1 = f(x_1) = q(x_1) = a_0 \Rightarrow a_0 = f_1, \quad (2.27)$$

$$f_2 = f(x_2) = q(x_2) = f_1 + a_1(x_2 - x_1) \Rightarrow a_1 = \frac{f_2 - f_1}{x_2 - x_1}, \quad (2.28)$$

$$f_3 = f(x_3) = q(x_3) = f_1 + a_1(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2) \Rightarrow \quad (2.29)$$

$$\Rightarrow a_2 = \frac{1}{x_3 - x_2} \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right).$$

Stacionární bod kvadratické approximace se získá položením první derivace rovné nule:

$$\frac{dq}{dx} = a_1 + a_2(x - x_1) + a_2(x - x_2) = 0, \quad (2.30)$$

z čehož plyne odhad optima:

$$\bar{x} = \frac{x_1 + x_2}{2} - \frac{a_1}{2a_2}. \quad (2.31)$$

Pokud je cílová funkce $f(x)$ unimodální a kvadratická approximace $q(x)$ také unimodální v intervalu (x_1, x_3) , pak \bar{x} představuje vhodný odhad hledaného optima x^* . Metoda kvadratické approximace je znázorněna na Obr. 2.4.

2.4.2 Powellova metoda – postupná kvadratická approximace

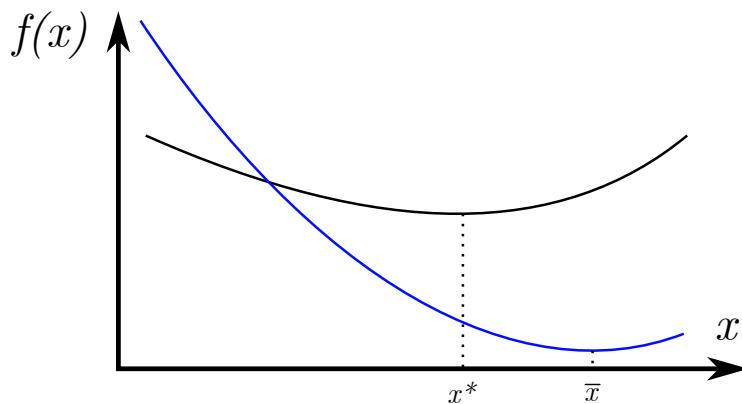
Tato metoda byla připravena Michaelm J. D. Powellem a využívá úspěšné zpřesnění odhadu. Procedura začíná volbou počátečního bodu x_1 a zvoleného kroku Δx . Postup je následující:

Krok 1. Výpočet $x_2 = x_1 + \Delta x$.

Krok 2. Vyhodnocení $f(x_1)$ a $f(x_2)$.

Krok 3. Určení x_3 podle porovnání hodnot.

- Pokud $f(x_1) > f(x_2)$, pak $x_3 = x_1 + 2 \cdot \Delta x$.
- Pokud $f(x_1) \leq f(x_2)$, pak $x_3 = x_1 - \Delta x$.



Obr. 2.4: Znázornění procedury kvadratické approximace

Krok 4. Vyhodnocení $f(x_3)$ a určení minima ze tří bodů:

$$\begin{aligned} F_{\min} &= \min\{f_1, f_2, f_3\}, \\ X_{\min} &= \text{bod } x_i \text{ odpovídající } F_{\min}. \end{aligned}$$

Krok 5. Určení odhadu \bar{x} pomocí kvadratické approximace získané z bodů x_1, x_2, x_3 .

Krok 6. Kontrola kritérií ukončení.

(a) Je rozdíl $F_{\min} - f(\bar{x})$ dostatečně malý?

(b) Je rozdíl $X_{\min} - \bar{x}$ dostatečně malý?

Pokud jsou obě podmínky splněny, výpočet končí. Pokud ne, pokračuje se dalším krokem.

Krok 7. Uložení nejlepšího bodu (X_{\min} nebo \bar{x}) a dvou hraničních bodů nebo bodů v jejich okolí, přeznačení a návrat k výpočtu od Kroku 4.

Poznámka 2.4.1. První průchod přes Krok 5 nemusí zaručit skutečné optimum, proto je nutné výpočet opakovat. Odhad \bar{x} může být extrapolován mimo x_3 ; v takovém případě je vhodné upravit bod \bar{x} pomocí kroku Δx .

2.5 Metody využívající derivace

Dosavadní metody měly pouze předpoklad, že funkce je unimodální, v některých případech je předpokládána i spojitost. Pokud je však funkce diferencovatelná, lze výpočet zefektivnit. Vychází se z hledání stacionárního bodu podle podmínky $f'(x) = df/dx|_{x=x^*} = 0$. Pokud je funkce třetího či vyššího řádu, řešení derivace může být náročné. Proto jsou nutné metody hledající approximaci stacionárního bodu $f(x)$. Derivaci lze získat třemi způsoby: analytickou (symbolickou) derivací, numerickou derivací či automatickou derivací. Dále je vysvětleno, jak lze připravit numerickou derivaci jedné proměnné.

2.5.1 Numerická derivace jedné proměnné

Numerická derivace je charakterizována konečnou velikostí kroku h (u analytické metody se h limitně blíží k 0). Existují tři způsoby. První je *dopředná derivace*:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}. \quad (2.32)$$

Druhý je *zpětná derivace*:

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}. \quad (2.33)$$

Třetí je *centrální derivace*:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2 \cdot h}. \quad (2.34)$$

Lze připravit i derivace druhého řádu

$$f''(x) \approx \frac{f(x+2 \cdot h) - 2 \cdot f(x) + f(x-2 \cdot h)}{4 \cdot h^2}. \quad (2.35)$$

Chyba tohoto výpočtu závisí na volbě h a ideálně by měla být co nejmenší. Při příliš malé hodnotě však převládá zaokrouhlovací chyba, proto je žádoucí tento parametr vhodně stanovit.

2.5.2 Metoda bisekce

Pokud je funkce unimodální a diferencovatelná, optimum se nachází v místě $f'(x^*) = 0$. Pro eliminaci regionu stačí vypočítat jeden bod k určení polohy $f'(x) = 0$. Pokud bod $x_M \in (a, b)$ vyjde $f'(x_M) < 0$, optimum leží vpravo od tohoto bodu, tedy v intervalu (x_M, b) . Naopak pokud $f'(x_M) > 0$, optimum je v intervalu (a, x_M) . Pro určení bodu se využívá vztah

$$x_M = \frac{a+b}{2}. \quad (2.36)$$

Protože rozdelení probíhá uprostřed intervalu, tato metoda se nazývá metoda bisekce či metoda půlení². Procedura je následující:

Krok 1. Připraví se $x_M = (a+b)/2$, pokud $f'(a) < 0$ a $f'(b) > 0$.

Krok 2. Vypočítá se $f'(x_M)$.

Krok 3. Pokud $|f'(x_M)| \leq \varepsilon$, výpočet je ukončen.

Krok 4. Pokud $f'(x_M) < 0$, nastaví se $a = x_M$. Pokud $f'(x_M) > 0$, nastaví se $b = x_M$.

Výpočet se opakuje od kroku 1.

Tato metoda je čistě založena na znaménku derivace, nikoli na její velikosti.

2.5.3 Metoda sečny

Oproti metodě bisekce metoda sečny využívá také velikost derivace. Funkce je také unimodální (derivace v krajních bodech intervalu (a, b) mají opačné znaménko). Spojením hodnot derivace $f'(a)$ a $f'(b)$ vzniká sečna, která obecně rozdělí interval (a, b) nerovnoměrně podle hodnoty x_s . Tato hodnota se určí pomocí podobnosti trojúhelníků a výsledný vztah je

$$x_s = b - \frac{f'(b) \cdot (b-a)}{f'(b) - f'(a)}. \quad (2.37)$$

Procedura je také obdobná metodě bisekce, avšak místo x_M se vypočítá x_s a interval zkoumání závisí na změně znaménka derivace:

- pokud $f'(a) \cdot f'(x_s) > 0$, interval se zúží na (x_s, b) ,
- pokud $f'(x_s) \cdot f'(b) > 0$, interval se zúží na (a, x_s) .

²Metoda půlení je obvykle označována při řešení nelineárních rovnic $f(x) = 0$, avšak lze ji využít i v optimalizaci. Je také nutné nezaměňovat s metodou půlení intervalu.

2.5.4 Newton-Raphson metoda

Newton-Raphson metoda předpokládá, že funkce je dvakrát diferencovatelná. Jejím cílem je approximovat stacionární bod pomocí odhadu x_k . Lineární approximace je sestavena jako³

$$\bar{f}'(x; x_k) = f'(x_k) + f''(x_k) \cdot (x - x_k). \quad (2.38)$$

Pokud výše uvedená derivace je rovna nule, lze získat approximaci dalšího bodu

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}. \quad (2.39)$$

Tato metoda může divergovat a je tedy závislá na počátečním odhadu. Také hodnoty approximace závisí na přesnosti derivací, a jsou tedy ovlivněny chybou numerické derivace.

Pro ukončení výpočtu se využívá buď kritérium $|f'(x_k)| < \varepsilon$, nebo $|x_{k+1} - x_k| < \varepsilon$.

2.6 Ukázkové příklady

Příklad 2.1. Cílem je spočítat minimum funkce pro $x \geq 0$

$$0.01 \cdot x^4 - 0.5 \cdot x^2 - x + 10. \quad (2.40)$$

Pro určení využijte výše uvedené metody:

1. metoda půlení intervalu,
2. metodu zlatého řezu,
3. metoda kvadratické approximace,
4. Powellova metoda,
5. metoda bisekce,
6. metoda sečny,
7. Newton-Raphson metoda.

Řešení. Prvním krokem je určení vhodného rozsahu, protože máme omezení pouze z jedné strany. Určení rozsahu je zvolena Swannova metoda. Jako počáteční bod je určen $x_0 = 0$. Pomocí této metody je zjištěno, že délka kroku $\Delta = 0.1$ nabývá kladné hodnoty. Poté jsou vypočítány hodnoty x_{k+1} , dokud nedojde k nárůstu funkce:

$$x_{k+1} = x_k + 0.1 \cdot 2^k \quad (2.41)$$

Postup je vidět v tabulce 2.1. Minimum se nachází v intervalu $0 \leq x \leq 12.7$, nicméně lze rozsah ještě zúžit na oblast, kde se pravděpodobně nachází, a to na $3.1 \leq x \leq 12.7$. Tedy $a = 3.1$ a $b = 12.7$.

Tab. 2.1: Hledání oblasti zájmu

k	0	1	2	3	4	5	6	7
x_k	0	0.1	0.3	0.7	1.5	3.1	6.3	12.7
$f(x_k)$	10	9.89	9.65	9.05	7.42	3.01	-0.39	176.8

³Newton-Raphson metoda se využívá také k řešení nelineárních rovnic ve tvaru $f(x) = 0$. Z principu se jedná o stejnou metodu, pouze se u optimalizace vyskytuje rovnice o řád vyšší.

Pro řešení je využit kód 2.1⁴, kde *fun* reprezentuje naši funkci.

Kód 2.1: Funkce pro Swannovu metodu hledání rozsahu

```
def Swann(fun ,x0 ,Delta=0.1):
    x = np.empty([20])
    x[0] = x0
    f0 = fun(x0)
    fL= fun(x0-Delta)
    fP=fun(x0+Delta)
    f = np.empty([20])
    f[0] = f0
    # Rozhodnutí zda je kladny parametr hledani Delta
    if fL > f0 > fP:
        for i in np.arange(20):
            x[i+1] = x[i] + 2**i * Delta
            f[i+1]= fun(x[i+1])
            if f[i+1]>f[i]: #Podminka ukonceni
                a=x[i-1]
                b=x[i+1]
                break

    elif fL < f0 < fP:
        for i in np.arange(20):
            x[i+1] = x[i] - 2**i * Delta
            f[i+1]= fun(x[i+1])
            if f[i+1]>f[i]: #Podminka ukonceni
                a=x[i+1]
                b=x[i-1]
                break

    elif fL > f0 < fP:
        a = x0-Delta
        b = x0+Delta

    return a,b
```

Následují výsledky jednotlivých metod, konkrétně jsou uvedeny první tři iterace, výsledek při zvolené toleranci a celkový počet iterací. Tolerance pro ukončení metod nultého řádu je $\delta = 0.01$ a pro metody druhého řádu $\varepsilon = 0.01$.

Metoda půlení intervalu

1. iterace:

$$x_M = \frac{a+b}{2} = 7.9, \quad L = b-a = 9.6,$$

$$x_L = a + \frac{L}{4} = 5.5, \quad x_P = b - \frac{L}{4} = 10.3.$$

⁴Definované funkce využívají knihovnu *NumPy*, proto je vhodné ji načíst na začátku příkazem *import numpy as np*.

Pro tyto body jsou určeny hodnoty cílové funkce

$$f(x_L) = -1.474, \quad f(x_M) = 9.845, \quad f(x_P) = 59.205.$$

Platí, že $f(x_L) < f(x_M)$, proto se předpokládá výsledek v intervalu $x \in < a, x_M >$ a tedy nové $b = x_M$.

2. iterace:

$$x_M = \frac{a + b}{2} = 5.5, \quad L = b - a = 4.8,$$

$$x_L = a + \frac{L}{4} = 4.3, \quad x_P = b - \frac{L}{4} = 6.7.$$

Pro tyto body jsou určeny hodnoty cílové funkce

$$f(x_L) = -0.126, \quad f(x_M) = -1.474, \quad f(x_P) = 1.006.$$

Platí, že $f(x_L) > f(x_M)$ a $f(x_M) < f(x_P)$, proto se předpokládá výsledek v intervalu $x \in < x_L, x_P >$ a tedy nové $a = x_L$ a $b = x_P$.

3. iterace:

$$x_M = \frac{a + b}{2} = 5.5, \quad L = b - a = 2.4,$$

$$x_L = a + \frac{L}{4} = 4.9, \quad x_P = b - \frac{L}{4} = 6.1.$$

Pro tyto body jsou určeny hodnoty cílové funkce

$$f(x_L) = -1.140, \quad f(x_M) = -1.474, \quad f(x_P) = -0.859.$$

Platí, že $f(x_L) > f(x_M)$ a $f(x_M) < f(x_P)$, proto se předpokládá výsledek v intervalu $x \in < x_L, x_P >$ a tedy nové $a = x_L$ a $b = x_P$.

Metoda půlení intervalu potřebuje pro vyřešení v dané toleranci 9 iterací a výsledek je $x_{PI}^* = 5.44375 \pm 0.00937$. Metoda půlení intervalu je implementována v kódu 2.2.

Kód 2.2: Funkce pro metodu půlení intervalu

```
def PuleniInt (fun ,a ,b ,tol=0.01):
    delta = 1                      # Zahajení vypočtu
    i = 0                           # Počítadlo
    while delta > tol:
        L = (b-a)                   # Délka intervalu
        # Vypocet kandidatu
        xL = a+1/4*L
        xM = (a+b)/2
        xP = b-1/4*L
        # Vypocet cilove funkce
        fL = fun (xL)
        fM = fun (xM)
        fP = fun (xP)
        # Eliminace regionu
```

```

if fL < fM:
    b = xM
elif fP < fM:
    a = xM
else :
    a = xL
    b = xP
# Vypocet rozsahu hledani
delta = (b-a)/2
i+=1

xStar = (b+a)/2
return xStar , delta

```

Metoda zlatého řezu

1. iterace:

$$x_L = b - (b - a) \cdot \varphi = 6.767, \quad x_P = a + (b - a) \cdot \varphi = 9.033.$$

Pro tyto body jsou určeny hodnoty cílové funkce

$$f(x_L) = 1.305, \quad f(x_P) = 26.749.$$

Platí, že $f(x_L) < f(x_P)$, proto se předpokládá výsledek v intervalu $x \in [a, x_P]$ a tedy nové $b = x_P$.

2. iterace:

$$x_L = b - (b - a) \cdot \varphi = 5.366, \quad x_P = a + (b - a) \cdot \varphi = 6.769.$$

Pro tyto body jsou určeny hodnoty cílové funkce

$$f(x_L) = -1.472, \quad f(x_P) = 1.305.$$

Platí, že $f(x_L) < f(x_P)$, proto se předpokládá výsledek v intervalu $x \in [a, x_P]$ a tedy nové $b = x_P$.

3. iterace:

$$x_L = b - (b - a) \cdot \varphi = 4.500, \quad x_P = a + (b - a) \cdot \varphi = 5.366.$$

Pro tyto body jsou určeny hodnoty cílové funkce

$$f(x_L) = -0.525, \quad f(x_P) = -1.472.$$

Platí, že $f(x_L) < f(x_P)$, proto se předpokládá výsledek v intervalu $x \in [x_L, b]$ a tedy nové $a = x_L$.

Metoda zlatého řezu potřebuje pro vyřešení v dané toleranci 13 iterací a výsledek je $x_{ZR}^* = 5.44213 \pm 0.00921$. Metoda zlatého řezu je uvedena v kódu 2.3.

Kód 2.3: Funkce pro metodu zlatého řezu

```
def ZlatyRez ( fun ,a ,b , tol=0.01) :
    delta = 1
    phi = (-1+5**(1/2))/2
    i = 0
    while delta > tol:
        xL = b-(b-a)*phi
        xP = a+(b-a)*phi

        fL = fun (xL)
        fP = fun (xP)

        if fL < fP:
            b = xP
        else:
            a = xL
            delta = (b-a)/2
            i +=1
    xStar = (b+a)/2
    return xStar , delta
```

Metoda kvadratické approximace

Pro metodu kvadratické approximace jsou využity tři body, a to $x_1 = a$, $x_2 = (a + b)/2$ a $x_3 = b$. Jsou tedy určeny tři hodnoty cílové funkce

$$f(x_1) = 3.018, \quad f(x_2) = 9.845, \quad f(x_3) = 176.799.$$

Pomocí těchto bodů jsou získány koeficienty approximační kvadratické funkce

$$\begin{aligned} a_0 &= f(x_1) = 3.018, \\ a_1 &= \frac{f_2 - f_1}{x_2 - x_1} = 1.422, \\ a_2 &= \frac{1}{x_3 - x_2} \cdot \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right) = 3.475. \end{aligned}$$

Odhad řešení je spočítán jako

$$\bar{x} = \frac{x_1 + x_2}{2} - \frac{a_1}{2 \cdot a_2} = 5.295.$$

Tato metoda je připravena v kódu 2.4. Kromě odhadu jsou výstupem i koeficienty a_i pro kvadratickou approximaci.

Kód 2.4: Funkce pro metodu kvadratické approximace

```
def KvadApro( fun ,x1 ,x2 ,x3 ) :
    f1 = fun (x1)
    f2 = fun (x2)
    f3 = fun (x3)
```

```

a0 = f1
a1 = (f2-f1)/(x2-x1)
a2 = 1/(x3-x2)*((f3-f1)/(x3-x1) - (f2-f1)/(x2-x1))

a = np.array([a0,a1,a2])
xOver = (x1+x2)/2-a1/(2*a2)

return xOver, a

```

Powellova metoda

1. iterace: Pro první iteraci je zvoleno

$$x_1 = \frac{a+b}{2} = 7.9, \Delta = 0.01.$$

Další bod je spočítán

$$x_2 = x_1 + \Delta = 8.0.$$

Pro určení x_3 je nutné určit hodnoty cílové funkce předešlých bodů

$$f(x_1) = 9.845, f(x_2) = 10.960 \rightarrow x_3 = x_1 - \Delta = 7.8 \rightarrow f(x_3) = 8.795.$$

Ze získaných hodnot jsou určeny $F_{min} = 8.795$ a $X_{min} = 7.8$. Je proveden odhad minima pomocí kvadratické approximace $\bar{x} = 6.231$.

2. iterace: Pro další výpočet platí $x_1 = \bar{x}$.

Další bod je určen

$$x_2 = x_1 + \Delta = 6.331.$$

Pro určení x_3 je nutné spočítat hodnoty cílové funkce předešlých bodů

$$f(x_1) = -0.567, f(x_2) = -0.303 \rightarrow x_3 = x_1 - \Delta = 6.131 \rightarrow f(x_3) = -0.794.$$

Ze získaných hodnot jsou určeny $F_{min} = -0.794$ a $X_{min} = 6.131$. Je proveden odhad minima pomocí kvadratické approximace $\bar{x} = 5.562$.

3. iterace: Pro další výpočet platí $x_1 = \bar{x}$

Další bod je spočítán

$$x_2 = x_1 + \Delta = 5.662.$$

Pro určení x_3 je nutné určit hodnoty cílové funkce předešlých bodů

$$f(x_1) = -1.459, f(x_2) = -1.413 \rightarrow x_3 = x_1 - \Delta = 5.462 \rightarrow f(x_3) = -1.478.$$

Ze získaných hodnot jsou určeny $F_{min} = -1.478$ a $X_{min} = 5.462$. Je proveden odhad minima pomocí kvadratické approximace $\bar{x} = 5.443$.

Powellova metoda vyžadovala pouze tři iterace k získání dostatečného odhadu $x_{PM}^* = 5.443$. Hodnoty z první a druhé iterace byly značně extrapolovány, což je třeba sledovat u jiných aplikací. Powellova metoda je uvedena v kódu 2.5.

Kód 2.5: Funkce pro Powellovu metodu

```
def PowellMetoda(fun ,x1 ,Delta=0.1 ,tol=0.01):
    ErrF = 1
    ErrX = 1
    i=0
    while ErrF > tol and ErrX > tol:

        x2 = x1 + Delta
        f1 = fun(x1)
        f2 = fun(x2)

        if f1 > f2:
            x3 = x1+2*Delta
        else:
            x3=x1-Delta
        f3 = fun(x3)
        Fmin = np.min([f1 ,f2 ,f3 ])
        Xmin = x1 if Fmin==f1 else x2 if Fmin==f2 else x3
        xOver = KvadApro(fun ,x1 ,x2 ,x3 ,flag=0)[0]

        ErrF = np.abs(Fmin - fun(xOver))
        ErrX = np.abs(Xmin - xOver)
        x1=xOver
        i+=1
    return xOver
```

Metoda bisekce

Pro určení derivací je využita numerická derivace, specificky centrální derivace s $h = 10^{-5}$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2 \cdot h}.$$

Centrální derivace je připravena v kódu 2.6.

Kód 2.6: Funkce centrální derivace

```
def cdiff(fun ,x ,h=10**-5):
    return (fun(x+h)-fun(x-h))/(2*h)
```

1. iterace:

$$x_M = \frac{a+b}{2} = 7.9, \quad f'(x_M) = 10.82.$$

Derivace $f'(x_M) > 0$ a $|f'(x_M)| > \varepsilon$, proto je optimum v intervalu $x \in [a, x_M]$, a tedy $b = x_M$.

2. iterace:

$$x_M = \frac{a+b}{2} = 5.5, \quad f'(x_M) = 0.155.$$

Derivace $f'(x_M) > 0$ a $|f'(x_M)| > \varepsilon$, proto je optimum v intervalu $x \in < a, x_M >$, a tedy $b = x_M$.

3. iterace:

$$x_M = \frac{a+b}{2} = 4.3, \quad f'(x_M) = -2.119.$$

Derivace $f'(x_M) < 0$ a $|f'(x_M)| > \varepsilon$, proto je optimum v intervalu $x \in < x_M, b >$, a tedy $a = x_M$.

Výsledek optimalizace pomocí metody bisekce je $x_{MB}^* = 5.44375 \pm 0.01875$ a bylo zapotřebí 9 iterací. Metoda bisekce je uvedena v kódu 2.7.

Kód 2.7: Funkce pro metodu bisekce

```
def Bisekce (fun ,a ,b ,tol=0.01):
    dfM = 1
    i=0
    while np . abs (dfM) > tol :
        i +=1
        xM = (b+a)/2
        dfM = c diff (fun ,xM)
        if np . abs (dfM) < tol :
            break
        if dfM <0:
            a = xM
        else :
            b = xM
    delta = (b-a)/2
    return xM, delta
```

Metoda sečny

Obdobně jako v případě metody bisekce, i zde je využita numerická centrální derivace.

1. iterace:

Jsou spočítány derivace funkce v bodech $x = a$ a $x = b$

$$f'(a) = -2.908, \quad f'(b) = 68.235.$$

Nový bod je získán jako

$$x_s = b - \frac{f'(b) \cdot (b-a)}{f'(b) - f'(a)} = 3.492, \quad f'(x_s) = -2.788.$$

Protože $|f'(x_s)| > \varepsilon$ a $f'(a) \cdot f'(x_s) > 0$, tak je optimum v intervalu $x \in < x_s, b >$.

2. iterace:

Jsou spočítány derivace funkce v bodech $x = a$ a $x = b$

$$f'(a) = -2.785, \quad f'(b) = 68.235.$$

Nový bod je získán jako

$$x_s = b - \frac{f'(b) \cdot (b - a)}{f'(b) - f'(a)} = 3.854, \quad f'(x_s) = -2.564.$$

Protože $|f'(x_s)| > \varepsilon$ a $f'(a) \cdot f'(x_s) > 0$, tak je optimum v intervalu $x \in < x_s, b >$.

3. iterace:

Jsou spočítány derivace funkce v bodech $x = a$ a $x = b$

$$f'(a) = -2.564, \quad f'(b) = 68.235.$$

Nový bod je získán jako

$$x_s = b - \frac{f'(b) \cdot (b - a)}{f'(b) - f'(a)} = 4.174, \quad f'(x_s) = -2.264.$$

Protože $|f'(x_s)| > \varepsilon$ a $f'(a) \cdot f'(x_s) > 0$, tak je optimum v intervalu $x \in < x_s, b >$.

Výsledek pomocí metody sečny je $x_{MB}^* = 5.43712^5$ a metoda potřebovala pro dosažení dané tolerance 23 iterací. Metoda sečny je uvedena v kódu 2.8.

Kód 2.8: Funkce metody sečny

```
def Secant( fun ,a ,b ,tol=0.01):
    dfs = 1
    i=0
    while np.abs(dfs) > tol:
        i +=1
        dfa = cdiff(fun ,a)
        dfb = cdiff(fun ,b)
        xs = b - (dfb*(b-a))/(dfb-dfa)
        dfs = cdiff(fun ,xs)
        if np.abs(dfs) < tol:
            break
        if dfa*dfs >0:
            a = xs
        else:
            b = xs

    delta = (b-a)/2
return xs ,delta
```

Newton-Raphson metoda

K určení druhé derivace je využit vztah pro numerickou derivaci s $h = 10^{-5}$

$$f''(x) = \frac{f(x + 2 \cdot h) - f(x) + f(x - 2 \cdot h)}{4 \cdot h^2}.$$

Druhé derivace je uvedena v kódu 2.9.

Kód 2.9: Funkce druhé derivace

⁵Pokud by se rozsah počítal z intervalu $x \in < a, b >$, tak by ve 23. iteraci byla $\delta = (b - a)/2 = 3.630$, avšak způsob eliminace není vždy přesně 50 %, proto tato hodnota u výsledku není uvedena.

```
def cdiff2 ( fun ,x ,h=10**-5) :
    return ( fun (x+2*h)-2*fun (x)+fun (x-2*h) ) / (4*h**2)
```

1. iterace:

Jsou spočítány první derivace a druhá derivace funkce v bodu $x_0 = \frac{a+b}{2} = 7.9$

$$f'(x_0) = 10.821, f''(x_0) = 6.489.$$

Nový bod je získán jako

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = 6.232.$$

2. iterace:

Jsou spočítány první derivace a druhá derivace funkce v bodu x_1

$$f'(x_1) = 2.451, f''(x_1) = 3.661.$$

Nový bod je získán jako

$$x_2 = x_1 - \frac{f'(x_1)}{f''(x_1)} = 5.563.$$

3. iterace:

Jsou spočítány první derivace a druhá derivace funkce v bodu x_2

$$f'(x_2) = 0.323, f''(x_2) = 2.713.$$

Nový bod je získán jako

$$x_3 = x_2 - \frac{f'(x_2)}{f''(x_2)} = 5.443.$$

Newton-Raphson metoda potřebuje 4 iterace pro získání optima $x_{NR}^* = 5.440$. Newton-Raphson metoda je uvedena v kódu 2.10.

Kód 2.10: Funkce Newton-Raphson metody

```
def NewtonRaphson ( fun ,xk ,tol=0.01) :
    err = 1
    i=0
    while err > tol :
        i+=1
        x0=xk
        xk=x0-cdiff ( fun , x0 ) / cdiff2 ( fun , x0 )
        err =np . abs ( cdiff ( fun , x0 ) )
    return xk
```

2.7 Cvičení

Cvičení 2.1. Pro zadanou funkci nalezněte maximum funkce jedné proměnné na rozsahu $x \in [1.01, 2.99]$ pomocí metody zlatého řezu

$$f(x) = \frac{x^4}{(x-1) \cdot (x-3)^3}.$$

Cvičení 2.2. Pro zadanou funkci nalezněte minimum funkce jedné proměnné stanovené podmínkou $x > 0$

$$f(x) = x^3 - 12 \cdot x + 3.$$

Cílem je určit pomocí Swannovy metody oblast zájmu a následně nalézt minimum pomocí metody sečny.

Cvičení 2.3. Pro zadanou funkci nalezněte extrémy funkce jedné proměnné na rozsahu $x \in <0.5, 15>$ pomocí Powellovy metody

$$f(x) = 3 \cdot x^2 + \frac{12}{x^3 + 1} - 5.$$

Cvičení 2.4.

Cvičení 2.5. Průhyb ocelového nosníku ($E = 200$ GPa) je definován pomocí složené funkce $w(x)$:

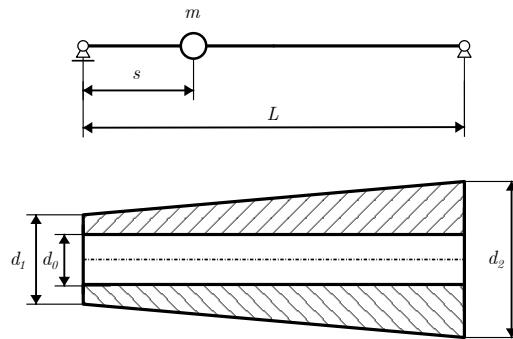
$$w(x) = -\frac{R_A \cdot \frac{x^3}{6} - q \cdot \frac{x^4}{24}}{E \cdot J_z} + C_1 \cdot x, \quad \text{pro } x \in <0, a>,$$

$$w(x) = -\frac{R_B \cdot \frac{(a+b-x)^3}{6}}{E \cdot J_z} + C_3 \cdot (a+b-x), \quad \text{pro } x \in <a, b>,$$

kde reakce jsou $R_A = 1343.75$ N a $R_B = 156.25$ N, spojití zatížení $q = 5$ N/mm, délky jsou $a = 500$ mm a $b = 300$ mm, moment setrvačnosti $J_z = 39760$ mm⁴ a integrační konstanty $C_1 = 0.005583$ a $C_3 = 0.003323$. Cílem je nalézt maximální průhyb nosníku.

Cvičení 2.6. Cílem je nalézt vzdálenost s , pro kterou bude uložení břemene na nehmotný nosník (viz Obr. 2.5) takové, aby vlastní frekvence nosníku byla minimální. Nosník má tvar dutého kuželu s průměry $d_0 = 0.045$ m, $d_1 = 0.05$ m a $d_2 = 0.1$ m a délku $L = 2$ m. Nosník je vyroben z konstrukční oceli s modulem pružnosti $E = 210$ GPa. Vlastní frekvence se vypočítá jako

$$\Omega_0 = \sqrt{\frac{k(s)}{m}}.$$



Obr. 2.5: Schéma nosníku

Poznámka: Tuhost $k(s)$ je ohybová tuhost, proto je nutné použít Castiglianovu větu s parametrem s .

Kapitola 3

Řešení optimalizace úlohy s více proměnnými

Tato kapitola se zaměřuje na metody řešení optimalizace s více proměnnými. Nejprve se však naváže na teorii stanovenou v kapitole 2. V tomto případě má optimalizační problém tvar

$$\min_{\mathbf{x}} F(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^N, \quad (3.1)$$

kde \mathbf{x} je vektor návrhových proměnných s dimenzí N . Výsledek cílové funkce $F(\mathbf{x})$ je skalární hodnota. Návrhové proměnné \mathbf{x} mohou být bez omezení nebo s omezením. Předpokládá se, že funkce a její derivace existují a jsou spojité všude, i přestože optimum může ležet v místě nespojitosti. Gradient cílové funkce se vypočítá jako vektor parciálních derivací jednotlivých návrhových proměnných:

$$\nabla F = \left[\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \frac{\partial F}{\partial x_3}, \dots, \frac{\partial F}{\partial x_N} \right]^T. \quad (3.2)$$

Mnohé funkce nejsou konvexní, ale naopak multimodální, a mohou mít více lokálních optim.

3.1 Vlastnosti funkcí s více proměnnými

Pro stanovení optima je vhodné připravit podmínky pro charakterizaci bodu. Taylorův rozvoj funkce s více proměnnými má tvar

$$F(\mathbf{x}) = F(\bar{\mathbf{x}}) + \nabla F(\bar{\mathbf{x}})\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \nabla^2 F(\bar{\mathbf{x}})\Delta\mathbf{x} + O_3(\Delta\mathbf{x}), \quad (3.3)$$

kde $\bar{\mathbf{x}}$ je bod rozvoje, $\Delta\mathbf{x}$ je změna návrhové proměnné, $\nabla F(\bar{\mathbf{x}})$ je N -rádkový vektor první derivace vyhodnocený v bodě rozvoje a $\nabla^2 F(\bar{\mathbf{x}}) = H_F(\bar{\mathbf{x}})$ je $N \times N$ symetrická matici druhých parciálních derivací cílové funkce, označovaná jako *Hessián*¹. Člen $O_3(\Delta\mathbf{x})$ zahrnuje členy vyššího řádu než dva a lze jej zanedbat. Změna cílové funkce je určena jako

$$\Delta F = F(\mathbf{x}) - F(\bar{\mathbf{x}}) = \nabla F(\bar{\mathbf{x}})^T \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \nabla^2 F(\bar{\mathbf{x}})\Delta\mathbf{x}. \quad (3.4)$$

¹Člen i -tého řádku a j -tého sloupce se vyjádří jako $\partial^2 F / \partial x_i \partial x_j$.

Pro připomenutí, minimum je bod, v jehož okolí všechny ostatní body mají vyšší hodnotu cílové funkce, tj.

$$\Delta F = F(\mathbf{x}) - F(\bar{\mathbf{x}}) \geq 0. \quad (3.5)$$

Návrhové proměnné $\bar{\mathbf{x}}$ představují globální minimum, pokud platí pro $\forall \mathbf{x} \in \mathbb{R}^N$. V této kapitole je globální minimum označeno jako \mathbf{x}^{**} . Pokud však toto platí pouze pro okolí definované $\|\mathbf{x} - \bar{\mathbf{x}}\| \leq \delta$ pro $\delta > 0$, pak se jedná o lokální minimum označené \mathbf{x}^* .

Pro maximum platí opačná nerovnost, tj.

$$\Delta F = F(\mathbf{x}) - F(\bar{\mathbf{x}}) \leq 0. \quad (3.6)$$

Pokud však hodnota ΔF závisí na výběru bodu z okolí a může být kladná, záporná či nulová, pak se takový bod označuje jako sedlový bod.

Stejně jako v případě funkce s jednou proměnnou se optimum nachází ve stacionárním bodě, tj. platí

$$\nabla F(\bar{\mathbf{x}}) = \mathbf{0}. \quad (3.7)$$

Rovnice změny funkce je poté definována pouze jako

$$\Delta F(\mathbf{x}) = \frac{1}{2} \Delta \mathbf{x}^T \nabla^2 F(\bar{\mathbf{x}}) \Delta \mathbf{x}. \quad (3.8)$$

Znaménko ΔF závisí na povaze tzv. kvadratické formy

$$Q(\mathbf{x}) = \Delta \mathbf{x}^T \nabla^2 F(\bar{\mathbf{x}}) \Delta \mathbf{x} \quad (3.9)$$

či

$$Q(\mathbf{z}) = \mathbf{z}^T \cdot A \cdot \mathbf{z} \quad (3.10)$$

Pro další hodnocení je nutné zopakovat pojem definitnosti matice:

1. A je pozitivně definitní, pokud $\forall z, Q(z) > 0$,
2. A je pozitivně semidefinitní, pokud $\forall z, Q(z) \geq 0$,
3. A je negativně definitní, pokud $\forall z, Q(z) < 0$,
4. A je negativně semidefinitní, pokud $\forall z, Q(z) \leq 0$,
5. A je indefinitní, pokud $\exists z, Q(z) > 0$ a $\exists z, Q(z) < 0$.

Z výše uvedených definic vyplývá:

1. Bod je lokální minimum, pokud $\nabla^2 F(\bar{\mathbf{x}})$ je pozitivně definitní.
2. Bod je lokální maximum, pokud $\nabla^2 F(\bar{\mathbf{x}})$ je negativně definitní.
3. Bod je sedlový, pokud $\nabla^2 F(\bar{\mathbf{x}})$ je indefinitní.

Pro další práci je vhodné zmínit nutnou a postačující podmínu. Nutná podmínka pro lokální minimum je

$$\nabla F(\mathbf{x}^*) = 0 \quad \text{a} \quad \nabla^2 F(\mathbf{x}^*) \text{ je pozitivně semidefinitní.} \quad (3.11)$$

Postačující podmínka je

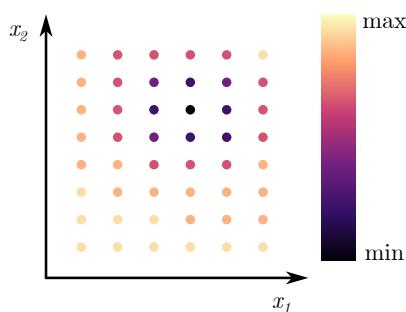
$$\nabla F(\mathbf{x}^*) = 0 \quad \text{a} \quad \nabla^2 F(\mathbf{x}^*) \text{ je pozitivně definitní,} \quad (3.12)$$

pak \mathbf{x}^* je izolované lokální minimum.

3.2 Evoluční metoda

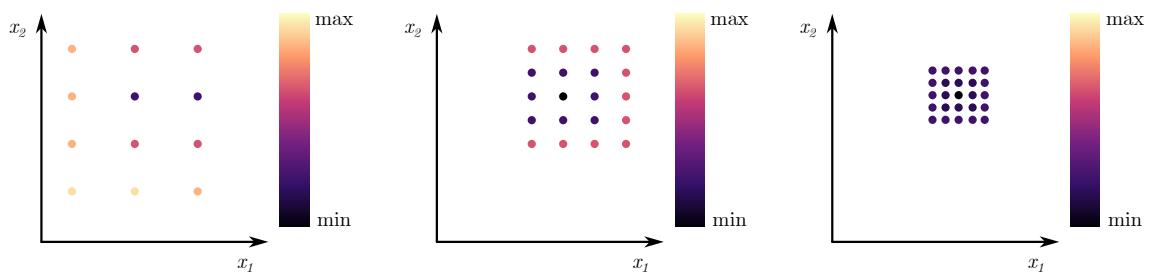
Evoluční metody cílí na přímé hledání optima. K tomu potřebují pouze hodnotu cílové funkce a nepotřebují gradient ani hessián. Existují různé evoluční metody, typicky inspirované přírodou (například *Optimalizace hejmem částic* z anglického *Particle Swarm Optimization*, genetické algoritmy/programování apod.). V rámci studijního textu jsou představeny základní metody, které lze dále rozšiřovat.

Prvním krokem evoluční metody je určení oblasti zájmu. Toho lze docílit rovnoměrným rozdelením intervalu $x_{i_{min}} \leq x_i \leq x_{i_{max}}$ pomocí kroku δ_{x_i} ², čímž se provede tzv. diskretizace prostoru (vytvoří se síť uzlů). Krok intervalu by měl odpovídat požadované přesnosti. Následně jsou na této síti spočítány hodnoty cílové funkce a hledá se uzel s minimem. Příklad výpočtu je na Obr. 3.1. Pro zvýšení efektivnosti lze využít parametrizace, paralelního výpočtu a případně vektorový výpočtu (pokud to dovoluje rozměr úlohy).



Obr. 3.1: Ukázka evoluční metody, pravidelná diskretizace

Nicméně výpočet cílové funkce může být náročný, proto lze přistoupit k určitým optimalizačním krokům. Prvním je snížit počet výpočtů větším krokem δ_{x_i} – začít na hrubé síti, nalézt minimum a poté v daném okolí připravit jemnější síť a výpočet zopakovat až do požadované přesnosti. Tento proces „evoluce“ dal metodě její název a lze ho vidět na Obr. 3.2.



Obr. 3.2: Ukázka evoluční metody, postupná diskretizace

Další snížení výpočetní náročnosti je možné opět snížením počtu výpočtů. Zvolí se bázový uzel \mathbf{x}_{base} , tj. odhad počátečního optima, a vhodný krok δ_{x_i} . Proces je popsán následovně:

Krok 1. Připraví se „rohové“ uzly vzdálené o hodnotu kroku, tj. $\mathbf{x}_{corner} = \mathbf{x}_{base} \pm \delta_x$ ³.

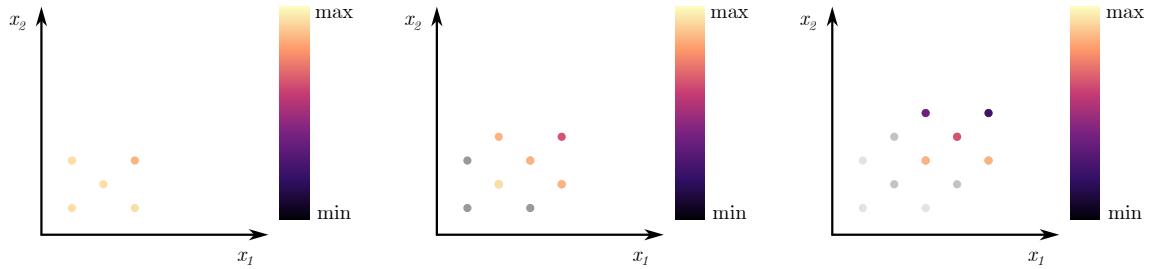
Krok 2. Vypočítají se hodnoty cílové funkce a hledá se minimum:

²Rozdelení má být rovnoměrné, avšak pro každou návrhovou proměnnou lze zvolit různý krok $\delta_{x_i} \neq \delta_x$.

³V případě dvojrozměrného prostoru vzniká pravidelný obdélník.

- (a) Pokud je minimum na rohovém uzlu, stává se tento uzel novým bázovým bodem a výpočet se opakuje od kroku 1.
- (b) Pokud se minimum nachází na bázovém uzlu, jsou kroky δ_x zmenšeny a výpočet se opakuje od kroku 1.

Proces výpočtu je ukončen po dosažení požadované přesnosti. Tento postup je znázorněn na obrázku 3.3.



Obr. 3.3: Ukázka evoluční metody, hledání

3.3 Simplexová metoda

Bylo zjištěno, že evoluční metodu lze ještě více zjednodušit, a to na tzv. Simplexovou metodu (v anglické literatuře *Nelder-Mead method*). Simplex je nejjednodušší polytop v dané dimenzi (0 dimenze – bod, 1 dimenze – úsečka, 2 dimenze – trojúhelník, 3 dimenze – čtyřstěn apod.). Pro výpočet se stanoví počáteční odhad x_0 a škálovací parametr α , určující velikost simplexu. Následně se vygenerují vrcholy simplexu a pro každý vrchol se vypočítá hodnota cílové funkce. Pro minimalizaci se tentokrát nehledá uzel s minimální hodnotou, ale maximální. Tento značí, že v daném směru není optimum, a je tedy nutné posunout se opačným směrem. Zbylé vrcholy slouží k nalezení reflexního bodu, přes který se uzel s maximální hodnotou zrcadlí. Pro nový bod se opět vypočítá hodnota cílové funkce a zjišťuje se maximum.

Pro nalezení optima je nutné dodržet několik základních pravidel:

1. Oscilace kolem minima – pokud se zrcadlení vrací do stejných bodů a výpočet tedy osciluje přes jednu hranu, je vhodné zvolit druhý největší vrchol pro zrcadlení.
2. Cyklování – při výpočtu zůstává jeden bod simplexu neměnný, což naznačuje dosažení optima; v takovém případě se zmenší škálovací parametr a vypočítá se nový simplex.

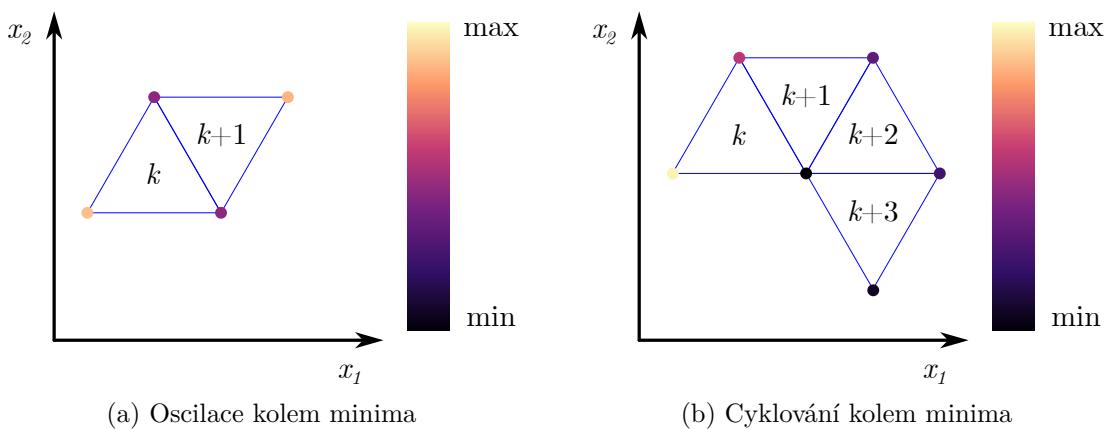
Nevýhodou této metody je pravidelnost simplexu. V případě, že je úloha špatně škálovatelná (jedna návrhová proměnná má jiný řád než ostatní), může konvergence být pomalá. Další nevýhodou je, že metoda nevyužívá historii předchozích iterací. Tyto problémy lze částečně vyřešit vhodným škálováním a modifikací metody.

3.4 Metoda největšího spádu

Další metody využívají první derivaci cílové funkce, která zpravidla určuje směr poklesu. Obecně se jedná o iterační metody s předpisem

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \cdot \mathbf{s}_k, \quad (3.13)$$

kde α_k je škálovací faktor určený pro maximální pokles a \mathbf{s}_k je vektor určující směr poklesu.



Obr. 3.4: Pravidla pro Simplexovou metodu

Myšlenka metody největšího spádu (také nazývané *Cauchyho metodou*) spočívá v tom, že směr největšího přírůstku funkce je ve směru gradientu, a tedy největší pokles nastává proti gradientu. Směr je dán vztahem

$$\mathbf{s}_k = -\nabla F(\mathbf{x}_k). \quad (3.14)$$

Následně stanoví se vhodně α_k tak, aby byl pokles co největší. Tento krok je však náročný, zejména pokud $\nabla F(\mathbf{x}_k) \rightarrow 0$, což je hlavní nevýhoda metody – dochází ke stagnaci v okolí optima. Iterační proces je tedy definován jako

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \cdot \nabla F(\mathbf{x}_k). \quad (3.15)$$

Pro nalezení α_k je využita optimalizace s jednou proměnnou tak, aby cílová funkce v další iteraci $F(\mathbf{x}_{k+1})$ byla minimalizována⁴.

3.5 Metoda sdružených gradientů

Přestože minulá metoda nese název „největšího spádu“, nejedná se o nejrychlejší metodu, zejména kvůli zmíněnému problému v okolí optima⁵. Metoda sdružených gradientů má vlastnost *kvadratické konvergenci* a využívá druhé derivace cílové funkce (tj. Hessián). Předpokládá se kvadratická formulace (případně approximace)

$$F(\mathbf{x}) = a + \mathbf{x}^T \cdot \mathbf{b} + \mathbf{x}^T \cdot \mathbf{C} \cdot \mathbf{x}. \quad (3.16)$$

Gradient je poté spočítán jako

$$\nabla F(\mathbf{x}_k) = \mathbf{C} \cdot \mathbf{x}_k + \mathbf{b} = \mathbf{g}_k. \quad (3.17)$$

Směr lze určit podle Fletcher a Reevese:

$$\mathbf{s}_k = -\mathbf{g}_k + \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_{k-1}\|^2} \cdot \mathbf{s}_{k-1}, \quad (3.18)$$

nebo podle Polaka a Ribiere:

⁴Hledání proměnné α_k se v literatuře označuje také jako *Line Search*.

⁵Metoda největšího spádu je zpočátku rychlá, avšak rychlosť konvergencie se postupne zpomaluje.

$$\mathbf{s}_k = -\mathbf{g}_k + \frac{(\mathbf{g}_k - \mathbf{g}_{k-1})^T \cdot \mathbf{g}_k}{\|\mathbf{g}_{k-1}\|^2} \cdot \mathbf{s}_{k-1}. \quad (3.19)$$

Pro počáteční odhad \mathbf{x}_0 je směr určen pouze z gradientu, tj. $\mathbf{s}_0 = \mathbf{g}_0$. Iterační proces je ukončen v případě, kdy je velikost gradientu menší než zvolená tolerance, $\|\mathbf{g}_k\| < \varepsilon$.

Pokud lze cílovou funkci vyjádřit přímo pomocí kvadratické formulace a Hessián je pozitivně definitní, pak se po N iteracích nalezne přesné řešení. Jedná se tedy o finitní metodu. Pokud však funkce není kvadratická, je vhodné po N krocích nebo při splnění podmínky $|\mathbf{g}_k^T \mathbf{g}_{k-1}| \geq 0.2 \|\mathbf{g}_k\|^2$ provést tzv. restart. Restart může mít podobu $\mathbf{s}_k = \mathbf{g}_k$.

3.6 Newtonova metoda

Další metoda využívá informace vyššího řádu z Taylorova rozvoje. Kvadratická approximace cílové funkce (bez členů vyššího řádu) má tvar

$$\tilde{F}(\mathbf{x}) = F(\mathbf{x}_k) + \nabla F(\mathbf{x}_k)^T \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \nabla^2 F(\mathbf{x}_k) \Delta \mathbf{x}. \quad (3.20)$$

Cílem je nalézt \mathbf{x}_{k+1} tak, aby approximace gradientu byla nulová. Tedy

$$\nabla \tilde{F}(\mathbf{x}, \mathbf{x}_k) = \nabla F(\mathbf{x}_k) + \nabla^2 F(\mathbf{x}_k) \Delta \mathbf{x} = \mathbf{0} \quad \rightarrow \quad \Delta \mathbf{x} = -\nabla^2 F(\mathbf{x}_k)^{-1} \nabla F(\mathbf{x}_k). \quad (3.21)$$

Takto postupná kvadratická approximace se nazývá Newtonova metoda optimalizace a má tvar

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \nabla^2 F(\mathbf{x}_k)^{-1} \nabla F(\mathbf{x}_k). \quad (3.22)$$

V blízkosti optima konverguje metoda rychle, zatímco ve větší vzdálenosti je konvergence pomalejší. Toto lze upravit pomocí škálovacího faktoru, modifikací připomínajících metodu největšího spádu. Také je důležité zajistit, aby Hessián byl pozitivně definitní, což zaručuje pokles funkce. Efektivnost metody závisí rovněž na přesnosti určení Hessiánu.

3.7 Levenberg–Marquardt metoda

Tato metoda kombinuje dvě výše uvedené metody – metodu největšího spádu a Newtonovu metodu – tak, aby se využily přednosti obou přístupů, a zároveň již využívá Hessián \mathbf{H}_k . Síla metody největšího spádu spočívá v hledání daleko od optima, zatímco Newtonova metoda je rychlá v blízkosti optima. Tato metoda byla popsána autory Levenberg a Marquardt. Směr hledání je určen jako

$$\mathbf{s}_k = -[\mathbf{H}_k + \lambda_k \cdot \mathbf{I}]^{-1} \nabla F(\mathbf{x}_k), \quad (3.23)$$

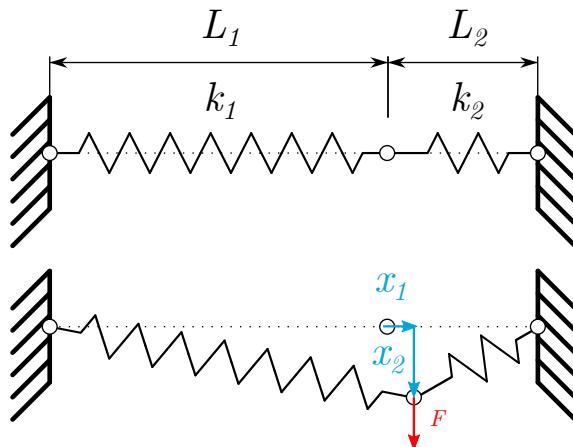
kde λ_k se stanoví jako velké číslo, například $\lambda_0 = 10^4$, a \mathbf{I} je jednotková matice. Velká hodnota λ vede k dominantní složce

$$[\mathbf{H}_0 + \lambda_0 \cdot \mathbf{I}]^{-1} \cong [\lambda_0 \cdot \mathbf{I}]^{-1} = \left(\frac{1}{\lambda_0} \right) \cdot \mathbf{I} \quad \rightarrow \quad \mathbf{s}_0 \cong \nabla F(\mathbf{x}_0). \quad (3.24)$$

S postupně se zmenšující hodnotou λ se metoda přepíná z gradientního směru na směr určený Newtonovou metodou. Pokud tedy dojde k úspěšné iteraci ($F(\mathbf{x}_{k+1}) < F(\mathbf{x}_k)$), nastaví se $\lambda_{k+1} < \lambda_k$ a iterace pokračuje. Naopak pokud nedojde k poklesu funkce, λ se zvětší a iterace pokračuje.

3.8 Ukázkový příklad

Příklad 3.1. Určete minimum celkové potenciální energie systému zobrazeného na obr. 3.5 pomocí optimalizace. Parametry úlohy jsou následující: délka první pružiny $L_1 = 120$ mm, délka druhé pružiny $L_2 = 80$ mm, tuhost první pružiny $k_1 = 10 \frac{\text{N}}{\text{mm}}$, tuhost druhé pružiny $k_2 = 100 \frac{\text{N}}{\text{mm}}$ a působící síla $F = 10$ N.



Obr. 3.5: Schéma mechanického systému

Řešení. Pro vyřešení úlohy připraví se potenciální energii jednotlivých pružin a práci působící sily:

$$\Pi = \frac{1}{2}k_1 \cdot (\Delta L_1)^2 + \frac{1}{2}k_2(\Delta L_2)^2 - F \cdot x_2, \quad (3.25)$$

kde ΔL_1 a ΔL_2 reprezentují prodloužení pružin a jsou definovány vzorce

$$\begin{aligned} \Delta L_1 &= \sqrt{(L_1 + x_1)^2 + x_2^2} - L_1, \\ \Delta L_2 &= \sqrt{(L_2 - x_1)^2 + x_2^2} - L_2. \end{aligned} \quad (3.26)$$

Návrhové proměnné x_1 a x_2 představují posuvy prostředního uzlu způsobené působením síly F . Cílová funkce je tedy formulována jako

$$\min_{\{x_1, x_2\} \in \mathbb{R}^2} \Pi. \quad (3.27)$$

Pro řešení jsou využity knihovny *scipy.optimize*, které implementují výše uvedené metody. Příklad je vyřešen dvěma různými metodami a výsledné řešení je $x = \{1.583, 17.215\}^T$. Řešení tohoto příkladu je uveden v kódu 3.1.

Kód 3.1: Řešící příkladu 3.1

```
import numpy as np
import scipy.optimize as sco

def myfun(x):
    k1=10
    k2 = 100
```

```

L1 = 120
L2 = 80
F = 10
delta1 = np.sqrt((L1+x[0])**2+x[1]**2)-L1
delta2 = np.sqrt((L2-x[0])**2+x[1]**2)-L2
Pi = 1/2*k1*delta1**2+1/2*k2*delta2**2-F*x[1]
return Pi

x0=np.array([0,0])
solution = sco.minimize(myfun, x0, method="Nelder-Mead")
xOptiSimplex = solution.x
PocetSimplex = solution.nit
print("Reseni pomocí Simplexovy metody je ",xOptiSimplex," . ")
print("Pro reseni je potreba ",PocetSimplex," iteraci.\n")
x0=np.array([0,0])
solution = sco.minimize(myfun, x0, method="CG")
xOptiCG = solution.x
PocetCG = solution.nit
print("Reseni pomocí sdruzenych gradientu je ",xOptiCG," . ")
print("Pro reseni je potreba ",PocetCG," iteraci.\n")

```

3.9 Cvičení

Cvičení 3.1. Nalezněte extrémy funkce dvou proměnných na rozsahu $x_i \in \langle 0, 200 \rangle$ pomocí optimalizačních metod:

$$f(x) = 2 \cdot x_1^2 + \frac{x_2^2}{5} + x_1 \cdot x_2 + 5 \cdot x_1 - 8 \cdot x_2.$$

Cvičení 3.2. Připravte pro cílové funkce Hessián a rozhodněte o jeho definitnosti v bodě $\mathbf{x} = \mathbf{0}$:

1. $f_1(\mathbf{x}) = 4 \cdot x_1^2 + 3 \cdot x_2^2 - 5 \cdot x_1 \cdot x_2 - 8 \cdot x_1 + 10$
2. $f_2(\mathbf{x}) = 2 \cdot x_1^2 + 10 \cdot x_2^2 - 3 \cdot x_1 \cdot x_2 - 80 \cdot x_1 + 40 \cdot x_2$
3. $f_3(\mathbf{x}) = -x_1^2 - 3 \cdot x_2^2 + 3 \cdot x_1 \cdot x_2 - 12 \cdot x_1 + 3 \cdot x_2$
4. $f_4(\mathbf{x}) = -3 \cdot x_1^2 - 3 \cdot x_2^2 + 2 \cdot x_1 \cdot x_2 - 2 \cdot x_1 - 2 \cdot x_2$

Cvičení 3.3. Minimalizujte funkci

$$f(x_1, x_2) = \frac{-10}{x_1^2 + x_2^2 + 2}$$

s využitím Newtonovy metody. Počáteční odhad je $\mathbf{x}_0 = \{4, 0\}^T$.

Cvičení 3.4. Z tahového diagramu materiálu jsou vybrány a zpracovány významné body vypsány v tabulce 3.1. V tabulce jsou vypsány hodnoty skutečného napětí σ_x a hodnoty skutečné plastické deformace ε_{px} . Tyto data jsou využity pro přípravu materiálového modelu pro numerickou simulaci.

Matematický model (Chabocheův model se dvěma kinematickými členy) je definován pomocí vzorce

$$\sigma_{CHAB} = \sigma_Y + \frac{C_1}{\gamma_1} \cdot (1 - e^{-\gamma_1 \cdot \varepsilon_{px}}) + C_2 \cdot \varepsilon_{px},$$

kde σ_Y , C_1 , C_2 a γ_1 jsou materiálové konstanty. V tomto příkladu se jedná o návrhové proměnné. Pro řešení je vhodné využít metodu nejmenších čtverců v podobě

$$\min_{\mathbf{x} \in \mathbb{R}^4} \sum (\sigma_i - \sigma_{CHAB}(\varepsilon_{p_i}))^2,$$

kde materiálový model je přepsána s návrhovými proměnnými jako

$$\sigma_{CHAB} = x_1 + \frac{x_2}{x_3} \cdot (1 - e^{-x_3 \cdot \varepsilon_{px}}) + x_4 \cdot \varepsilon_{px}.$$

Počáteční odhad $\mathbf{x}_0 = \{150, 200\ 000, 1\ 200, 4\ 000\}^T$.

Tab. 3.1: Zadané hodnoty

σ (MPa)	ε_p (-)
190	0.0003
300	0.002
360	0.01
400	0.02
410	2.5

Cvičení 3.5. Rayleighho kvocient mechanického systému se třemi stupni volnosti je spočítán jako

$$R(\mathbf{x}) = \frac{\mathbf{x}^T \cdot \mathbf{K} \cdot \mathbf{x}}{\mathbf{x}^T \cdot \mathbf{M} \cdot \mathbf{x}}$$

kde

$$\mathbf{K} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix},$$

kde \mathbf{K} je matice tuhosti, \mathbf{M} je matice hmotnosti a \mathbf{x} je vektor vlastních tvarů. Minimalizací $R(\mathbf{x})$ je možné získat minimální vlastní frekvenci. Připravte tedy cílovou funkci a nalezněte vlastní tvar pro minimum.

Kapitola 4

Vázané optimalizační problémy

Předešlé kapitoly se věnovaly nevázané optimalizaci, avšak úlohy inženýrské praxe bývají obvykle zatíženy různými omezeními. Tato omezení redukují zkoumaný prostor. Na první pohled se může zdát, že to usnadní optimalizaci, avšak omezující funkce bývají často nelineární a ztěžují výpočet. Například podmínka stacionárního bodu nemusí být splněna v optimu (gradient cílové funkce v optimu nemusí být nulový). Omezující podmínky se mohou vyskytovat ve formě rovností či nerovností. Začneme s rovnostmi, protože jsou jednodušší.

4.1 Metoda dosazení

Optimalizační problém s omezujícími podmínkami lze definovat jako

$$\begin{aligned} & \min_{\boldsymbol{x} \in \mathbb{R}^N} F(\boldsymbol{x}), \\ & h_k(\boldsymbol{x}) = 0, \quad k = 1, \dots, K. \end{aligned} \tag{4.1}$$

Nejjednodušší metodou řešení je využití omezující podmínky k přímému vyjádření některých návrhových proměnných \boldsymbol{x} . Tím se redukuje dimenze návrhového prostoru z N na $N - K$ (za předpokladu, že $N > K$), a následně lze použít metody popsané v předchozích kapitolách. Nevýhodou je, že je nutné explicitně vyjádřit omezující podmínky. Pokud mají omezení charakter implicitní, je nutné použít jiné metody.

4.2 Lagrangeova metoda

Tato metoda stanovuje sadu nutných podmínek pro identifikaci kandidáta na optimum optimalizace s omezením ve formě rovnosti. Metoda převádí vázaný problém na ekvivalentní nevázanou optimalizaci pomocí tzv. *Lagrangeových multiplikátorů*. Vázaný optimalizační problém je definován jako

$$\begin{aligned} & \min_{\boldsymbol{x} \in \mathbb{R}^N} F(\boldsymbol{x}), \\ & h_k(\boldsymbol{x}) = 0, \quad k = 1, \dots, K. \end{aligned} \tag{4.2}$$

Pomocí Lagrangeových multiplikátorů se problém převede na

$$\min L(\boldsymbol{x}, \boldsymbol{\lambda}) = F(\boldsymbol{x}) - \sum_{k=1}^K \lambda_k \cdot h_k(\boldsymbol{x}), \tag{4.3}$$

kde nevázaná funkce $L(\mathbf{x}, \boldsymbol{\lambda})$ se nazývá *Lagrangeova funkce* (také *Lagrangián*) a λ_k je Lagrangeův multiplikátor pro k -tou podmínu. Tyto multiplikátory představují další parametry optimalizace a řešení lze hledat obdobně jako u nevázaného problému, pouze návrhový prostor je rozšířen o multiplikátory. Hledá se tedy řešení pro $N + K$ neznámých pomocí stacionární podmínky

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = 0, \quad \nabla_{\boldsymbol{\lambda}} L(\mathbf{x}, \boldsymbol{\lambda}) = 0. \quad (4.4)$$

Tato podmínka určí stacionární bod ekvivalentního problému, avšak k rozhodnutí, zda se jedná o minimum, je zapotřebí vyhodnotit Hessián¹ pro dané kandidáty na optimum. Platí opět, že minimum má Hessián pozitivně definitní.

Pokud není omezení definováno jako rovnost, lze jej upravit pomocí transformace zavedením dalšího parametru y

$$g_j(\mathbf{x}) \geq 0 \rightarrow h_j(\mathbf{x}, y) = g_j(\mathbf{x}) - y^2. \quad (4.5)$$

4.3 Karush-Kuhn-Tuckerovy podmínky

Autoři William Karush, Harold W. Kuhn a Albert W. Tucker rozšířili Lagrangeovu metodu a zobecnili ji pro nelineární optimalizační problémy (nelineární programování) s omezeními ve formě rovností i nerovností. Problém optimalizace je definován jako

$$\begin{aligned} & \min_{\mathbf{x} \in \mathbb{R}^N} F(\mathbf{x}), \\ & g_j(\mathbf{x}) \geq 0, \quad j = 1, \dots, J, \\ & h_k(\mathbf{x}) = 0, \quad k = 1, \dots, K. \end{aligned} \quad (4.6)$$

Omezující podmínka ve formě nerovnosti může být ve dvou stavech: buď je aktivní $g_j(\mathbf{x}) = 0$, nebo neaktivní $g_j(\mathbf{x}) > 0$. Pokud je omezení neaktivní, není nutné jej zahrnovat do výpočtu a optimalizační problém se zjednoduší. Problém však nastává při identifikaci, zda je omezení aktivní, ještě před vyřešením problému.

Zmínění autoři stanovili nutné a postačující podmínky pro nelineární optimalizaci s předpokladem, že všechny funkce (cílová i omezující) jsou diferencovatelné. Tyto podmínky se nazývají *Karush-Kuhn-Tuckerovy podmínky* (také jen *Kuhn-Tuckerovy podmínky* nebo *KKT/KT podmínky*).

Cílem KKT podmínek je nalézt řešení tak, aby byly splněny podmínky odvozené z Lagrangeovy funkce

$$L(\mathbf{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) = F(\mathbf{x}) - \sum_{j=1}^J \nu_j \cdot g_j(\mathbf{x}) - \sum_{k=1}^K \lambda_k \cdot h_k(\mathbf{x}). \quad (4.7)$$

Podmínka stacionarity

$$\nabla F(\mathbf{x}) - \sum_{j=1}^J \nu_j \cdot \nabla g_j(\mathbf{x}) - \sum_{k=1}^K \lambda_k \cdot \nabla h_k(\mathbf{x}) = \mathbf{0} \quad (4.8)$$

¹Hessián je tvořen parciálními derivacemi pouze s ohledem na návrhové proměnné x_i a x_j , nikoliv Lagrangeovy multiplikátory λ_k , avšak hodnoty Hessiánu mohou být multiplikátory ovlivněny.

Podmínka komplementarity

$$\nu_j \cdot g_j(\mathbf{x}), \quad j = 1, \dots, J \quad (4.9)$$

Podmínka nezápornosti pro multiplikátory nerovností

$$\nu_j \geq 0, \quad j = 1, \dots, J \quad (4.10)$$

Podmínka přípustnosti

$$g_j(\mathbf{x}) \geq 0, \quad j = 1, \dots, J \quad (4.11)$$

$$h_k(\mathbf{x}) = 0, \quad k = 1, \dots, K \quad (4.12)$$

Problém určení, zda je podmínka aktivní, je vyřešen pomocí rovnic (4.9) a (4.10). Pokud je multiplikátor $\nu_j = 0$, znamená to, že podmínka není aktivní, protože $g_j(\mathbf{x}) > 0$. Naopak pokud $g_j(\mathbf{x}) = 0$, pak je multiplikátor $\nu_j > 0$.²

Řešení pomocí výše uvedených rovnic a nerovnic obvykle vede na více bodů, které se označují jako Karush-Kuhn-Tuckerovy body (KKT body), tj. vhodní kandidáti optima. Pro určení, zda daný kandidát je skutečné optimum, jsou připraveny další nutné a postačující podmínky vycházející z podmínky *regularity*.

Podmínka regularity

Podmínka regularity přípustného řešení vyžaduje, aby vektory $\nabla g_j(\mathbf{x})$ pro všechny j , pro které platí $g_j(\mathbf{x}) = 0$, a vektory $\nabla h_k(\mathbf{x})$ byly lineárně nezávislé.

Díky podmínce regularity lze o KKT bodech a optimech učinit následující závěry:

1. Pokud je daný bod z oblasti přípustných řešení optimem a splňuje podmínu regularity, pak je tento bod součástí KKT bodů.
2. Pokud je daný bod z oblasti přípustných řešení optimem, ale nesplňuje podmínu regularity, může být KKT bodem, ale nemusí.
3. Jsou-li podmínky regularity splněny pro všechny body přípustného řešení, pak optimum leží mezi KKT body.
4. Postačující podmínka, aby KKT bod byl lokálním minimem nelineárního optimalizačního problému, je, že Hessián Lagrangeovy funkce v tomto bodě je pozitivně definitní.³

V praxi je však obvykle náročné spočítat Hessián Lagrangeovy funkce, proto se často předpokládá, že KKT bod je minimum (minimálně na základě vyčíslení hodnot cílové funkce a kontroly omezujících podmínek).

4.4 Metoda pokutové funkce

Cílem je připravit ekvivalentní nevázaný problém pomocí tzv. pokutové funkce:

$$P(\mathbf{x}, \mathbf{r}) = F(\mathbf{x}) + \Omega[h_k(\mathbf{x}), g_j(\mathbf{x}), \mathbf{r}], \quad (4.13)$$

kde \mathbf{r} jsou penalizační parametry a Ω je penalizační člen závislý na parametru \mathbf{r} a na omezujících podmínkách. Pokutová funkce se následně řeší obvyklými metodami, hledá se stacionární bod $\nabla P(\mathbf{x}, \mathbf{r}) = \mathbf{0}$ a vhodná velikost penalizačního parametru. Parametr \mathbf{r} se obvykle aktualizuje v průběhu iteračního procesu. Proces pokutové funkce by měl mít následující charakteristiky:

²Způsob definice omezující podmínky a Lagrangeovy funkce ovlivňuje znaménko multiplikátoru. Například pro podmínu $g_j(\mathbf{x}) \leq 0$ se do Lagangiánu přidává kladný člen $L(\mathbf{x}, \boldsymbol{\nu}) = F(\mathbf{x}) + \sum \nu_j \cdot g_j(\mathbf{x})$.

³Přestože tato podmínka není splněna, bod stále může být optimum.

1. S počtem iterací se dosáhne optimálního řešení $\lim_{k \rightarrow T < \infty} \mathbf{x}_k = \mathbf{x}^*$.
2. Problém minimalizace $P(\mathbf{x}, \mathbf{r})$ má stejnou náročnost výpočtu jako původní cílová funkce $F(\mathbf{x})$.

3. Aktualizace \mathbf{r}_{k+1} by měla být jednoduchá.

Následují příklady penalizačních členů pro různé typy omezení. Tyto členy obsahují penalizační parametr R , kdy $R \rightarrow \infty$ (reálně se volí dostatečně velké číslo).

1. Kvadratická pokutová funkce (rovnosti):

$$\Omega = R \cdot \{h(\mathbf{x})\}^2. \quad (4.14)$$

Tato funkce penalizuje kladné i záporné hodnoty $h(\mathbf{x})$ stejnou měrou. S rostoucím R se omezující podmínka „aktivuje“, tj. $h(\mathbf{x}_T) = 0$. Funkce Ω i její derivace jsou spojité.

2. Nekonečné bariéry:

$$\Omega_j = R_j \cdot |\tilde{g}_j(\mathbf{x})|, \quad \tilde{g}_j(\mathbf{x}) = \begin{cases} g_j(\mathbf{x}), & g_j(\mathbf{x}) < 0, \\ 0, & g_j(\mathbf{x}) \geq 0. \end{cases} \quad (4.15)$$

Tato funkce je nespojitá na hranici zkoumané oblasti a gradient $\nabla P(\mathbf{x}, \mathbf{r})$ zde neexistuje.

3. Logaritmická bariéra:

$$\Omega_j = -\frac{1}{R_j} \ln[g_j(\mathbf{x})]. \quad (4.16)$$

Funkce penalizuje vnitřní body $0 < g_j(\mathbf{x}) < 1$ kladně a $g_j(\mathbf{x}) > 1$ negativně. Lze vynulovat penalizaci pro $g_j(\mathbf{x}) > 1$, avšak vzniká nespojitost. Funkce není definována pro $g_j(\mathbf{x}) \leq 0$.

4. Inverzní funkce:

$$\Omega_j = \frac{1}{R_j} \frac{1}{g_j(\mathbf{x})}. \quad (4.17)$$

Jedná se o další bariérovou funkci, která penalizuje nepřípustné body. Gradient na hranici však neexistuje.

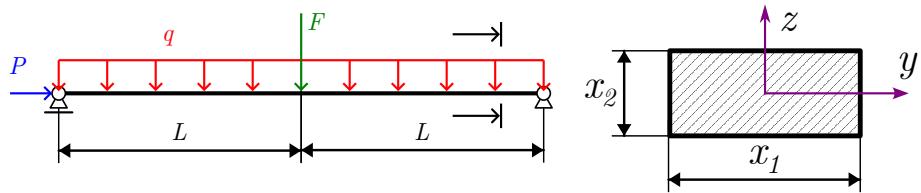
5. Modifikovaná kvadratická pokutová funkce (nerovnosti):

$$\Omega_j = R_j \cdot [\tilde{g}_j(\mathbf{x})]^2, \quad \tilde{g}_j(\mathbf{x}) = \begin{cases} g_j(\mathbf{x}), & g_j(\mathbf{x}) \leq 0, \\ 0, & g_j(\mathbf{x}) > 0. \end{cases} \quad (4.18)$$

Tato funkce penalizuje pouze vnější body, takže optimum může ležet mimo přípustný prostor. Přípustné i nepřípustné body jsou vyhodnoceny tímto členem konzistentně, bez penalizace přípustných bodů.

4.5 Ukázkový příklad

Příklad 4.1. Hliníkový nosník s délkou $L = 1\ 500$ mm je zatížen spojitým zatížením způsobeným vlastní tíhou q a osamělou silou $F = 5\ 000$ N působící uprostřed nosníku. Navíc může být nosník zatížen axiální tlakovou silou $P = 20$ kN. Cílem je nalézt rozměry obdélníkového průřezu tak, aby byla hmotnost minimální a současně byly splněny podmínky: dodržení dovoleného napětí $\sigma_D = 100$ MPa, kritické vzpěrné síly F_{krit} podle Eulera nižší než axiální síla $P = 20$ kN a průhyb pod silou nepřekročí hodnotu $w_{fD} = 8$ mm. Pro řešení uvažujte předpoklady lineární pružnosti. Materiálová data jsou $\rho_{Al} = 2700$ kg · m⁻³ a $E_{Al} = 70\ 000$ MPa. Minimální příčné rozměry nosníku jsou 10 mm. Schéma úlohy je na obrázku 4.1.



Obr. 4.1: Schéma úlohy

Řešení. Tento problém je již vázaný, má tři omezující podmínky (σ_o , w_F , F_{krit}) a dvě omezení návrhového prostoru (každý rozměr obdélníku má být větší než 10 mm). Cílová funkce má v tomto případě tvar⁴

$$\begin{aligned} \min_{x_1, x_2} m &= \rho \cdot L \cdot x_1 \cdot x_2, \\ x_1 &\geq 10 \text{ mm}, \quad x_2 \geq 10 \text{ mm}, \end{aligned} \tag{4.19}$$

kde x_1 a x_2 jsou návrhové proměnné a reprezentují rozměry příčného průřezu. První omezující podmínka vychází z pevnostní podmínky

$$\sigma_o \leq \sigma_D. \tag{4.20}$$

Ohybové napětí se vypočítá pomocí vztahu

$$\sigma_o = \frac{Mo_{max}}{J_z} \cdot y_{max}, \tag{4.21}$$

kde Mo_{max} je maximální ohybový moment, J_z je osový moment setrvačnosti kolem osy z a y_{max} definuje nejvzdálenější vlákno od těžiště průřezu ve směru y . Výsledné napětí pro zadaný nosník je

$$\frac{\left(\frac{F}{2} - \frac{q \cdot L}{2}\right) \cdot \frac{L}{2}}{\frac{1}{12} \cdot x_1 \cdot x_2^3} \cdot \frac{x_2}{2} \leq \sigma_D, \tag{4.22}$$

kde q reprezentuje spojité zatížení způsobené vlastní tíhou nosníku

$$q = \rho \cdot g \cdot x_1 \cdot x_2. \tag{4.23}$$

Další omezující podmínkou je průhyb uprostřed nosníku w_F , který má být menší než jeho dovolená hodnota w_D . Pro výpočet lze využít připravené vzorce ze strojnických tabulek a principu superpozice

$$w_F = \frac{F \cdot L^3}{48 \cdot E \cdot J_z} + \frac{5 \cdot q \cdot L^4}{384 \cdot E \cdot J_z} \leq w_D. \tag{4.24}$$

Eulerova kritická vzpěrná síla je spočítána jako

$$F_{krit} = \frac{\pi^2 \cdot E \cdot J_{min}}{L^2}, \tag{4.25}$$

⁴Pro výpočet je vhodné převést jednotky na stejné, například na SI (m, N, kg, Pa, s) nebo inženýrské (mm, N, t, MPa, s).

kde J_{min} je minimální hlavní moment setrvačnosti. Zde je nutné rozhodnout, který ze dvou směrů má menší moment setrvačnosti⁵.

Pro výpočet pomocí jiných optimalizačních metod je nutné problém rozdělit na dvě omezující funkce

$$\begin{aligned} \frac{\pi^2 \cdot E \cdot \frac{x_1^3 \cdot x_2}{12}}{L^2} &\geq P, \\ \frac{\pi^2 \cdot E \cdot \frac{x_1 \cdot x_2^3}{12}}{L^2} &\geq P. \end{aligned} \quad (4.26)$$

Optimalizační problém je tedy stanoven jako

$$\begin{aligned} \min_{x_1, x_2} m &= \rho \cdot L \cdot x_1 \cdot x_2, \\ g_1 &= x_1 - 10 \geq 0, \\ g_2 &= x_2 - 10 \geq 0, \\ g_3 &= \sigma_D - \frac{\left(\frac{F}{2} - \frac{\rho \cdot g \cdot x_1 \cdot x_2 \cdot L}{2} \right) \cdot \frac{L}{2}}{\frac{1}{6} \cdot x_1 \cdot x_2^2} \geq 0, \\ g_4 &= w_D - \left(\frac{F \cdot L^3}{48 \cdot E \cdot \frac{1}{12} \cdot x_1 \cdot x_2^3} + \frac{5 \cdot \rho \cdot g \cdot L^4}{384 \cdot E \cdot \frac{1}{12} \cdot x_2^2} \right) \geq 0, \\ g_5 &= \frac{\pi^2 \cdot E \cdot \frac{x_1^3 \cdot x_2}{12}}{L^2} - P \geq 0, \\ g_6 &= \frac{\pi^2 \cdot E \cdot \frac{x_1 \cdot x_2^3}{12}}{L^2} - P \geq 0. \end{aligned} \quad (4.27)$$

Lagrangeova funkce je tedy stanovena jako

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\nu}) &= m(x_1, x_2) - \nu_1 \cdot g_1(x_1, x_2) - \nu_2 \cdot g_2(x_1, x_2) \\ &\quad - \nu_3 \cdot g_3(x_1, x_2) - \nu_4 \cdot g_4(x_1, x_2) \\ &\quad - \nu_5 \cdot g_5(x_1, x_2) - \nu_6 \cdot g_6(x_1, x_2). \end{aligned} \quad (4.28)$$

Gradienty dílčích funkcí jsou

$$\frac{\partial m}{\partial x_1} = \rho \cdot L \cdot x_2, \quad (4.29)$$

$$\frac{\partial m}{\partial x_2} = \rho \cdot L \cdot x_1, \quad (4.30)$$

$$\frac{\partial g_1}{\partial x_1} = 1, \quad (4.31)$$

⁵Průřez je symetrický a deviační moment setrvačnosti vychází 0 mm⁴, proto je snadné určit hlavní momenty setrvačnosti.

$$\frac{\partial g_1}{\partial x_2} = 0, \quad (4.32)$$

$$\frac{\partial g_2}{\partial x_1} = 0, \quad (4.33)$$

$$\frac{\partial g_2}{\partial x_2} = 1, \quad (4.34)$$

$$\frac{\partial g_3}{\partial x_1} = -\frac{\left(-\frac{\rho \cdot g \cdot x_2 \cdot L}{2}\right) \cdot \frac{L}{2} \cdot \left(\frac{1}{6} \cdot x_1 \cdot x_2^2\right) - \left(\frac{F}{2} - \frac{\rho \cdot g \cdot x_1 \cdot x_2 \cdot L}{2}\right) \cdot \frac{L}{2} \cdot \left(\frac{1}{6} \cdot x_2^2\right)}{\left(\frac{1}{6} \cdot x_1 \cdot x_2^2\right)^2}, \quad (4.35)$$

$$\frac{\partial g_3}{\partial x_2} = -\frac{\left(-\frac{\rho \cdot g \cdot x_1 \cdot L}{2}\right) \cdot \frac{L}{2} \cdot \left(\frac{1}{6} \cdot x_1 \cdot x_2^2\right) - \left(\frac{F}{2} - \frac{\rho \cdot g \cdot x_1 \cdot x_2 \cdot L}{2}\right) \cdot \frac{L}{2} \cdot \left(\frac{1}{3} \cdot x_1 \cdot x_2\right)}{\left(\frac{1}{6} \cdot x_1 \cdot x_2^2\right)^2}, \quad (4.36)$$

$$\frac{\partial g_4}{\partial x_1} = \left(\frac{F \cdot L^3}{48 \cdot E \cdot \frac{1}{12} \cdot x_1^2 \cdot x_2^3} \right), \quad (4.37)$$

$$\frac{\partial g_4}{\partial x_2} = \left(\frac{3 \cdot F \cdot L^3}{48 \cdot E \cdot \frac{1}{12} \cdot x_1 \cdot x_2^4} + \frac{15 \cdot \rho \cdot g \cdot L^4}{384 \cdot E \cdot \frac{1}{12} \cdot x_2^3} \right), \quad (4.38)$$

$$\frac{\partial g_5}{\partial x_1} = \frac{\pi^2 \cdot E \cdot \frac{3 \cdot x_1^2 \cdot x_2}{12}}{L^2}, \quad (4.39)$$

$$\frac{\partial g_5}{\partial x_2} = \frac{\pi^2 \cdot E \cdot \frac{x_1^3}{12}}{L^2}, \quad (4.40)$$

$$\frac{\partial g_6}{\partial x_1} = \frac{\pi^2 \cdot E \cdot \frac{x_2^3}{12}}{L^2}, \quad (4.41)$$

$$\frac{\partial g_6}{\partial x_2} = \frac{\pi^2 \cdot E \cdot \frac{3 \cdot x_1 \cdot x_2^2}{12}}{L^2}. \quad (4.42)$$

KKT podmínky pro tento příklad jsou

$$\begin{aligned}
& \rho \cdot L \cdot x_2 - \nu_1 + \\
& + \nu_3 \cdot \frac{\left(\frac{-\rho \cdot g \cdot x_2 \cdot L}{2} \right) \cdot \frac{L}{2} \cdot \left(\frac{1}{6} \cdot x_1 \cdot x_2^2 \right) - \left(\frac{F}{2} - \frac{\rho \cdot g \cdot x_1 \cdot x_2 \cdot L}{2} \right) \cdot \frac{L}{2} \cdot \left(\frac{1}{6} \cdot x_2^2 \right)}{\left(\frac{1}{6} \cdot x_1 \cdot x_2^2 \right)^2} + \\
& - \nu_4 \cdot \left(\frac{F \cdot L^3}{48 \cdot E \cdot \frac{1}{12} \cdot x_1^2 \cdot x_2^3} \right) - \nu_5 \cdot \frac{\pi^2 \cdot E \cdot \frac{3 \cdot x_1^2 \cdot x_2}{12}}{L^2} - \nu_6 \cdot \frac{\pi^2 \cdot E \cdot \frac{x_2^3}{12}}{L^2} = 0, \quad (4.43)
\end{aligned}$$

$$\begin{aligned}
& \rho \cdot L \cdot x_2 - \nu_2 + \\
& + \nu_3 \cdot \frac{\left(\frac{-\rho \cdot g \cdot x_1 \cdot L}{2} \right) \cdot \frac{L}{2} \cdot \left(\frac{1}{6} \cdot x_1 \cdot x_2^2 \right) - \left(\frac{F}{2} - \frac{\rho \cdot g \cdot x_1 \cdot x_2 \cdot L}{2} \right) \cdot \frac{L}{2} \cdot \left(\frac{1}{3} \cdot x_1 \cdot x_2 \right)}{\left(\frac{1}{6} \cdot x_1 \cdot x_2^2 \right)^2} + \\
& - \nu_4 \left(\frac{3 \cdot F \cdot L^3}{48 \cdot E \cdot \frac{1}{12} \cdot x_1 \cdot x_2^4} + \frac{15 \cdot \rho \cdot g \cdot L^4}{384 \cdot E \cdot \frac{1}{12} \cdot x_2^3} \right) - \nu_5 \frac{\pi^2 \cdot E \cdot \frac{x_1^3}{12}}{L^2} - \nu_6 \cdot \frac{\pi^2 \cdot E \cdot \frac{3 \cdot x_1 \cdot x_2^2}{12}}{L^2} = 0, \quad (4.44)
\end{aligned}$$

$$\nu_1 \cdot g_1 = 0, \quad (4.45)$$

$$\nu_2 \cdot g_2 = 0, \quad (4.46)$$

$$\nu_3 \cdot g_3 = 0, \quad (4.47)$$

$$\nu_4 \cdot g_4 = 0, \quad (4.48)$$

$$\nu_5 \cdot g_5 = 0, \quad (4.49)$$

$$\nu_6 \cdot g_6 = 0, \quad (4.50)$$

$$\nu_1 \geq 0, \quad (4.51)$$

$$\nu_2 \geq 0, \quad (4.52)$$

$$\nu_3 \geq 0, \quad (4.53)$$

$$\nu_4 \geq 0, \quad (4.54)$$

$$\nu_5 \geq 0, \quad (4.55)$$

$$\nu_6 \geq 0, \quad (4.56)$$

Z fyzikálního hlediska malý průřez nedává smysl, že $g_1 > 0$ a $g_2 > 0$, proto se úloha zredukuje kvůli $\nu_1 = 0$ a $\nu_2 = 0$. Neznámými jsou tedy $x_1, x_2, \nu_3, \nu_4, \nu_5, \nu_6$. Pro analytické

Tab. 4.1: Možné řešení multiplikátorů dané úlohy

Varianta	$\nu_3 = 0$	$\nu_4 = 0$	$\nu_5 = 0$	$\nu_6 = 0$
1	Ne	Ne	Ne	Ne
2	Ne	Ne	Ne	Ano
3	Ne	Ne	Ano	Ne
4	Ne	Ne	Ano	Ano
5	Ne	Ano	Ne	Ne
6	Ne	Ano	Ne	Ano
7	Ne	Ano	Ano	Ne
8	Ne	Ano	Ano	Ano
9	Ano	Ne	Ne	Ne
10	Ano	Ne	Ne	Ano
11	Ano	Ne	Ano	Ne
12	Ano	Ne	Ano	Ano
13	Ano	Ano	Ne	Ne
14	Ano	Ano	Ne	Ano
15	Ano	Ano	Ano	Ne
16	Ano	Ano	Ano	Ano

řešení této úlohy existuje 16 řešení pomocí komplementarity, viz tabulka 4.1. Pro výpočet je využita knihovna *SciPy*.

Varianta 1

První varianta vede k řešení 6 rovnic a 6 neznámých ($x_1, x_2, \nu_3, \nu_4, \nu_5, \nu_6$), všechny omezující podmínky jsou aktivní. Příklad je však definován pouze dvěma návrhovými proměnnými, takže řešení této varianty neexistuje.

Varianta 2

Druhá varianta vede k řešení 5 rovnic a 5 neznámých ($x_1, x_2, \nu_3, \nu_4, \nu_5$), všechny omezující podmínky jsou aktivní kromě g_6 . Příklad je však definován pouze dvěma návrhovými proměnnými, takže řešení této varianty neexistuje.

Varianta 3

Třetí varianta vede k řešení 5 rovnic a 5 neznámých ($x_1, x_2, \nu_3, \nu_4, \nu_6$), všechny omezující podmínky jsou aktivní kromě g_5 . Příklad je však definován pouze dvěma návrhovými proměnnými, takže řešení této varianty neexistuje.

Varianta 4

Čtvrtá varianta vede k řešení 4 rovnic a 4 neznámých (x_1, x_2, ν_3, ν_4), všechny omezující podmínky jsou aktivní kromě g_5 a g_6 . Řešení lze získat z

$$\begin{aligned} g_3 &= 0, \\ g_4 &= 0. \end{aligned}$$

Řešení je $\mathbf{x}_{(4)} = \{23.74, 68.38\}^T$. Ostatní podmínky jsou splněny a hodnota cílové funkce je $m(\mathbf{x}_{(4)}) = 6.576$ kg.

Varianta 5

Pátá varianta vede k řešení 5 rovnic a 5 neznámých $(x_1, x_2, \nu_3, \nu_5, \nu_6)$, všechny omezující podmínky jsou aktivní kromě g_4 . Příklad je však definován pouze dvěma návrhovými proměnnými, takže řešení této varianty neexistuje.

Varianta 6

Šestá varianta vede k řešení 4 rovnic a 4 neznámých (x_1, x_2, ν_3, ν_5) , všechny omezující podmínky jsou aktivní kromě g_4 a g_6 . Řešení lze získat z

$$\begin{aligned} g_3 &= 0, \\ g_5 &= 0. \end{aligned}$$

Řešení je $\mathbf{x}_{(6)} = \{23.744, 68.386\}^T$. Ostatní podmínky jsou splněny a hodnota cílové funkce je $m(\mathbf{x}_{(6)}) = 6.372$ kg.

Varianta 7

Sedmá varianta vede k řešení 4 rovnic a 4 neznámých (x_1, x_2, ν_3, ν_6) , všechny omezující podmínky jsou aktivní kromě g_4 a g_5 . Řešení lze získat z

$$\begin{aligned} g_3 &= 0, \\ g_6 &= 0. \end{aligned}$$

Řešení je $\mathbf{x}_{(7)} = \{100, 19.84\}^T$ ⁶. Avšak $g_4(\mathbf{x}_{(7)}) < 0$, takže se nejedná o řešení problému.

Varianta 8

Osmá varianta vede k řešení 3 rovnic a 3 neznámých (x_1, x_2, ν_3) , všechny omezující podmínky jsou aktivní kromě g_4 , g_5 a g_6 . Řešení je $\mathbf{x}_{(8)} = \{35.66, 35.66\}^T$. Avšak $g_4(\mathbf{x}_{(8)}) < 0$, takže se nejedná o řešení problému.

Varianta 9

Devátá varianta vede k řešení 5 rovnic a 5 neznámých $(x_1, x_2, \nu_4, \nu_5, \nu_6)$, všechny omezující podmínky jsou aktivní kromě g_3 . Příklad je však definován pouze dvěma návrhovými proměnnými, takže řešení této varianty neexistuje.

Varianta 10

Desátá varianta vede k řešení 4 rovnic a 4 neznámých (x_1, x_2, ν_4, ν_5) , všechny omezující podmínky jsou aktivní kromě g_3 a g_6 . Řešení lze získat z rovnic

$$\begin{aligned} g_4 &= 0, \\ g_5 &= 0. \end{aligned}$$

⁶Numerický výpočet byl omezen hodnotou $x_{max_n} \leq 100$. Bez tohoto omezení by řešení mohlo splnit omezující podmínky, avšak nejednalo by se o optimum kvůli rozdílu mezi hodnotami g_4 a g_5 .

Řešení je $\mathbf{x}_{(10)} = \{22.378, 69.744\}^T$. Zbylé omezující podmínky nejsou porušeny a hodnota cílové funkce je $m(\mathbf{x}_{(10)}) = 6.321$ kg.

Varianta 11

Jedenáctá varianta vede k řešení 4 rovnic a 4 neznámých (x_1, x_2, ν_4, ν_6) , všechny omezující podmínky jsou aktivní kromě g_3 . Řešení lze získat z rovnic

$$\begin{aligned} g_4 &= 0, \\ g_6 &= 0. \end{aligned}$$

Řešení je $\mathbf{x}_{(11)} = \{22.06, 32.84\}^T$. Avšak $g_3(\mathbf{x}_{(11)}) < 0$, takže se nejedná o řešení problému.

Varianta 12

Dvanáctá varianta vede k řešení 3 rovnic a 3 neznámých (x_1, x_2, ν_4) , všechny omezující podmínky jsou aktivní kromě g_3, g_5 a g_6 . Řešení je mimo hledaný interval.

Varianta 13

Třináctá varianta vede k řešení 4 rovnic a 4 neznámých (x_1, x_2, ν_5, ν_6) , všechny omezující podmínky jsou aktivní kromě g_3 a g_4 . Řešení je mimo hledaný interval.

$$\begin{aligned} g_5 &= 0, \\ g_6 &= 0. \end{aligned}$$

Řešení je $\mathbf{x}_{(13)} = \{29.73, 29.73\}^T$. Avšak podmínka $g_3 < 0$, takže se nejedná o řešení problému.

Varianta 14

Čtrnáctá varianta vede k řešení 3 rovnic a 3 neznámých (x_1, x_2, ν_5) , všechny omezující podmínky jsou aktivní kromě g_3, g_4 a g_6 . Řešení je mimo hledaný interval.

Varianta 15

Patnáctá varianta vede k řešení 3 rovnic a 3 neznámých (x_1, x_2, ν_6) , všechny omezující podmínky jsou aktivní kromě g_3, g_4 a g_5 . Řešení je mimo hledaný interval.

Varianta 16

Poslední varianta vede k řešení $x_1 = 0$ a $x_2 = 0$, což je v rozporu s technologickými podmínkami $x_1 \geq 10$ a $x_2 \geq 10$, které jsou tedy eliminovány.

Řešení z výše uvedených variant se nachází mezi čtvrtou, šestou a desátou variantou. Nejmenší hodnota je u desáté varianty: $\mathbf{x}_{(10)} = \{25.07, 49.60\}^T$.

Vzhledem k tomu, že je v tomto případě šest omezujících podmínek, je výpočetně vhodnější využít pokutovou funkci s kvadratickým penalizačním členem. Pokutová funkce se všemi penalizačními členy je definována jako

$$P(\mathbf{x}, R) = m(\mathbf{x}) + R \cdot \sum_{j=1}^6 \tilde{g}_j. \quad (4.57)$$

Výsledek je následně získán metodami pro více proměnných. Parametr R se postupně zvětšuje, při $R = 10^5$ je dostatečně velký a výsledek je $\mathbf{x}_{(PF)} = \{22.287, 70.601\}^T$, přičemž hmotnost nosníku je $m(\mathbf{x}_{(PF)}) = 6.372$ kg. Metoda pokutové funkce je uvedena v kódu 4.1.

Kód 4.1: Realizace řešení pomocí pokutové funkce příkladu 4.1

```

import numpy as np
import scipy.optimize as spo

L = 1500
F = 5000
P = 20000
sigmaD = 100
wD = 8
rho = 2.7*10**-9
g = 9810
E = 70000

def m(x):
    return x[0]*x[1]*L*rho

def g3(x):
    return (sigmaD - (F/2 - L/2 * rho * g * x[0] * x[1]) * L / 2 / (1/6 * x[0] * x[1]**2)) / sigmaD

def g4(x):
    return (wD - (((F*L**3)/(48*E*1/12*x[0]*x[1]**3)) + ((5*rho*g*L**4)/((384*E*1/12*x[1]**2)))) / wD

def g5(x):
    return ((np.pi**2*E*x[0]**3*x[1]/12) / (L**2) - P) / P
def g6(x):
    return ((np.pi**2*E*x[0]*x[1]**3/12) / (L**2) - P) / P

def Pfun(x,R):
    gT1 = R*(np.min((x[0]-10,0)))**2
    gT2 = R*(np.min((x[1]-10,0)))**2
    gT3 = R*(np.min((g3(x),0)))**2
    gT4 = R*(np.min((g4(x),0)))**2
    gT5 = R*(np.min((g5(x),0)))**2
    gT6 = R*(np.min((g6(x),0)))**2
    return m(x)+gT1+gT2+gT3+gT4+gT5+gT6

err = 1
tol=10**-3
x0 = np.array([100,100])
R=10000

```

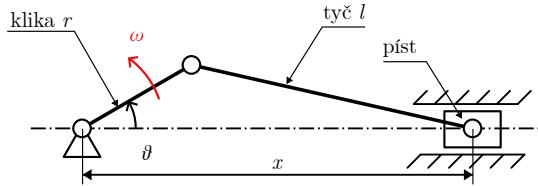
```

solution = spo.minimize(Pfun, x0, args=(R))
xN=solution.x
print(xN)
print(solution.success)
print(m(xN)*1000)

```

4.6 Cvičení

Cvičení 4.1. Klikový mechanismus (viz Obr. 4.2) má konstantní úhlovou rychlosť kliky $\omega = 100 \text{ rad/s}$. Cílem je nalézt délky kliky r a spojovací tyče l pomocí optimalizace, tak aby při úhlu $\vartheta = 30^\circ$ měl píst maximální rychlosť v_p . Pro zajištění možnosti provést plnou otáčku platí $l \geq 2.5 \cdot r$. Další omezení jsou: délka kliky $0.5 \leq r \leq 10 \text{ m}$, délka spojovací tyče $2.5 \leq l \leq 25 \text{ m}$ a délka mezi kloubem a pístem při úhlu $\theta = 30^\circ$ $10 \leq x \leq 20 \text{ m}$.



Obr. 4.2: Schéma klikového mechanismu

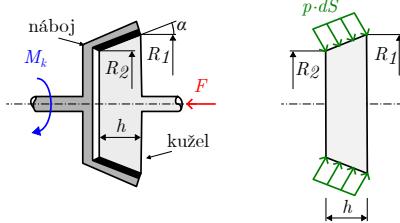
Cvičení 4.2. Klikový mechanismus z problému 4.1 má nový cíl: nalézt pomocí grafické metody délky kliky r a spojovací tyče l , tak aby při úhlu $\theta = 30^\circ$ měl píst maximální zrychlení a_p .

Cvičení 4.3. Kroutící moment přenášený kuželovou spojkou (viz Obr. 4.3) při rovnoměrném tlaku je dán vztahem:

$$M_k = \frac{2\pi f p}{3 \sin \alpha} (R_1^3 - R_2^3), \quad (4.58)$$

kde p je tlak mezi kuželem a nábojem, f je součinitel tření, α je úhel kužele, R_1 je vnější poloměr a R_2 vnitřní poloměr. Cílem je nalézt poloměry R_1 a R_2 tak, aby objem kužele byl minimální.

Parametry kužele jsou: úhel $\alpha = 30^\circ$, přítlačná síla $F = 5000 \text{ N}$, součinitel tření $f = 0.5$. Omezení jsou: kroutící moment $M_k \geq 250 \text{ Nm}$ a technologická omezení $R_1 \geq 2 \cdot R_2$, $0 \leq R_1 \leq 100 \text{ mm}$.



Obr. 4.3: Schéma kuželové spojky

Seznam použitých zdrojů

- [1] RAVINDRAN, A.; REKLAITIS, G. V.; RAGSDELL, K. M. *Engineering optimization: methods and applications.* 2nd ed. Hoboken, N.J.: John Wiley, 2006. ISBN 978-0-471-55814-9.
- [2] RAO, Singiresu S. *Engineering optimization: theory and practice.* 4th ed. New York: Wiley, c2009. ISBN 978-0-470-18352-6.
- [3] MARTINS, Joaquim R. R. A.; NING, Andrew. *Engineering design optimization.* Cambridge: Cambridge University Press, 2022. ISBN 978-110-8833-417.