MDS6106 - Optimization and Modeling
Exercise 3

Zijin CAI

2024-11-01

Name: *Zijin CAI*
Student ID: *224040002*
Date: *2024-11-01*

Assignment A3.1 Bisection and Golden Section Method

assume we can find x_l and x_r such that $g(x_l) < 0$ and $g(x_r) > 0$, by the intermediate value theorem, if g is continuous, there must exist a root x^* of g in $[x_l, x_r]$.

Bisection Method:

- define $x_m = \frac{x_l + x_r}{2}$,
- if $g(x_m) = 0$, then output x_m,
- otherwise:
  - if $g(x_m) < 0$, then $x_l = x_m$
  - if $g(x_m) > 0$, then $x_r = x_m$
- if $|x_r - x_l| < \epsilon$, stop and output $\frac{x_l + x_r}{2}$, otherwise go to step 1.

Golden Section Method:

assume we start with $[x_l, x_r]$, assume $0 < \phi < 0.5$

- set $x_l' = \phi x_r + (1 - \phi) x_l$ and $x_r' = (1 - \phi) x_r + \phi x_l$
- if $f(x_l') < f(x_r')$, then the minimizer must be in $[x_l, x_r']$, so set $x_r = x_r'$
- otherwise, the minimizer must be in $[x_l', x_r]$, so set $x_l = x_l'$
- if $|x_r - x_l| < \epsilon$, stop and output $\frac{x_l + x_r}{2}$, otherwise go to step 1.

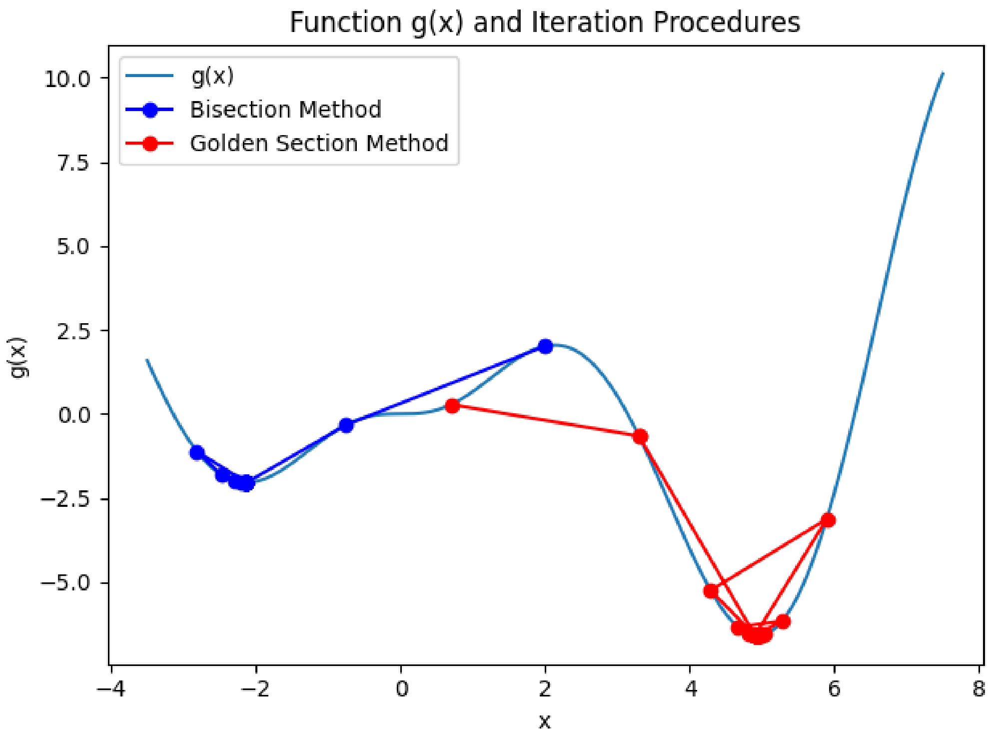
Python code implementation:

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize_scalar

# Define the function g(x) and its derivative g_prime(x)
def g(variable):
    return variable * np.sin(variable) * np.arctan(variable)

def g_prime(f):
    return np.sin(f) * np.arctan(f) + f * np.cos(f) * np.arctan(f) + f * np.sin(f) / (1 + f **
    2)

# Bisection method
def bisection_method(a, b, tol, g_derivative):
    iterations = 0
    x_values = [a, b]
    while (b - a) / 2.0 > tol:
        c = (a + b) / 2
        x_values.append(c)
        if g_derivative(c) == 0 or (b - a) / 2.0 < tol:
            return (a + b) / 2, iterations, x_values
```

```python
        elif g_derivative(a) * g_derivative(c) < 0:
            b = c
        else:
            a = c
        iterations += 1
    return (a + b) / 2, iterations, x_values

# Golden section method
def golden_section_method(a, b, tol, function):
    GR = (np.sqrt(5) - 1) / 2  # Golden ratio
    # GR_conjugate = 1 - GR
    iterations = 0
    x_values = [a, b]
    c = (1 - GR)*b + GR*a
    d = GR*b + (1 - GR)*a
    while (b - a) > tol:
        x_values.append(c)
        x_values.append(d)
        if function(c) == 0 or function(d) == 0 or (b - a)/2 < tol:
            return (a + b) / 2, iterations, x_values
        elif function(c) < function(d):
            b = d
            d = c
            c = (1 - GR) * b + GR * a
        else:
            a = c
            c = d
            d = GR * b + (1 - GR) * a
        iterations += 1
    return (a + b) / 2, iterations, x_values

# Define the interval and tolerance
interval = (-3.5, 7.5)
tolerance = 1e-5

# Apply the bisection method
bisection_result, bisection_iterations, bisection_x_values = bisection_method(interval[0], interval[1], tolerance, g_prime)
print(f"Bisection Method Result: x = {bisection_result}, Iterations = {bisection_iterations}")

# Apply the golden section method
golden_section_result, golden_section_iterations, golden_section_x_values = golden_section_method(interval[0], interval[1], tolerance, g)
print(f"Golden Section Method Result: x = {golden_section_result}, Iterations = {golden_section_iterations}")

# Vectorize g_prime to apply it element-wise
g_prime_vectorized = np.vectorize(g_prime)

# Plot g(x) and the iteration procedures
# (both from the third iteration for better visualization)
x = np.linspace(interval[0], interval[1], 400)
plt.plot(x, g(x), label='g(x)')
plt.plot(bisection_x_values[2:], g(bisection_x_values)[2:], 'bo-', label='Bisection Method')
plt.plot(golden_section_x_values[2:], g(golden_section_x_values)[2:], 'ro-', label='Golden Section Method')
plt.title('Function g(x) and Iteration Procedures')
plt.xlabel('x')
plt.ylabel('g(x)')
plt.legend()

plt.tight_layout()
plt.show()
```

```
Bisection Method Result: x = -2.131824016571045, Iterations = 20
Golden Section Method Result: x = 4.93956281265806, Iterations = 28
```

the following figure shows the function $g(x)$ and the iteration procedures of the bisection method and the golden section method: the bisection method converges to $x = -2.131824016571045$ after 20 iterations, while the golden section method converges to $x = 4.93956281265806$ after 28 iterations



the result indicates that both methods can be effective in finding local minima, but they may **not necessarily converge to the same point** (the required iterations for different methods can also vary significantly), and this is typical when applying to multimodal functions without further information to guide the search (in this case, the function $g(x) = x\sin(x)\arctan(x)$ has several convex regions in the interval $[-3.5, 7.5]$, in which the bisection method and the golden section method may iterate towards different local minima).

Assignment A3.2 Descent Directions

let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function, and consider $\mathbf{x} \in \mathbb{R}^n$ with $\nabla f(\mathbf{x}) \neq \mathbf{0}$

a): a vector $\mathbf{d} \in \mathbb{R}^n$ is a descent direction of f at $\mathbf{x}$ if $\nabla f(\mathbf{x})^T \mathbf{d} < 0$

given $\mathbf{d} = -(\nabla f(\mathbf{x})_j)\mathbf{e}_j$, where $\mathbf{e}_j \in \mathbb{R}^n$ is the j-th unit vector, this means $\mathbf{d}$ is a vector with all elements equal to zero except for the j-th element, which is $-\frac{\partial f}{\partial x_j}(\mathbf{x})$.

the dot product $\nabla f(\mathbf{x})^T \mathbf{d}$ can be written as

$$\begin{aligned} \nabla f(\mathbf{x})^T \mathbf{d} &= \nabla f(\mathbf{x})^T \times \left(-\frac{\partial f}{\partial x_j}(\mathbf{x}) e_j\right) \\ &= -\frac{\partial f}{\partial x_j}(\mathbf{x}) \nabla f(\mathbf{x})^T \\ &= -\left(\frac{\partial f}{\partial x_j}(\mathbf{x})\right)^2 \leq 0 \end{aligned} \tag{1}$$

because $\nabla f(\mathbf{x}) \neq \mathbf{0}$, $\frac{\partial f}{\partial x_j}(\mathbf{x}) \neq 0$:

$$\nabla f(\mathbf{x})^T \mathbf{d} = -\left(\frac{\partial f}{\partial x_j}(\mathbf{x})\right)^2 < 0 \tag{2}$$

therefore, it verifies that $\mathbf{d}$ is a descent direction of f at $\mathbf{x}$

b): given that $\mathbf{D} = \text{diag}(\delta_1, \dots, \delta_n) \in \mathbb{R}^{n\times n}$ is diagonal with $\delta_i > 0$ for all i, the inverse of $\mathbf{D}$, denoted as $\mathbf{D}^{-1}$, is also a diagonal matrix with diagonal entries $\delta_i^{-1} > 0$ for all i

and as $\mathbf{d} = -\mathbf{D}^{-1}\nabla f(\mathbf{x})$, the dot product $\nabla f(\mathbf{x})^T \mathbf{d}$ can be written as:

$$\nabla f(\mathbf{x})^T \mathbf{d} = \nabla f(\mathbf{x})^T \times \left(-\mathbf{D}^{-1} \nabla f(\mathbf{x})\right) = -\nabla f(\mathbf{x})^T \left(\mathbf{D}^{-1} \nabla f(\mathbf{x})\right) \tag{3}$$

the product $\mathbf{D}^{-1}\nabla f(\mathbf{x})$ is a vector that each component is scaled by corresponding δ_i^{-1}:

$$\text{let } \nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{x}) \end{pmatrix}, \quad \text{then } \mathbf{D}^{-1}\nabla f(\mathbf{x}) = \begin{pmatrix} \delta_1^{-1} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \delta_n^{-1} \frac{\partial f}{\partial x_n}(\mathbf{x}) \end{pmatrix} \tag{4}$$

then rewrite the dot product in (3):

$$\nabla f(\mathbf{x})^T \mathbf{d} = -\sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(\mathbf{x}) \times \delta_i^{-1} \frac{\partial f}{\partial x_i}(\mathbf{x}) = -\sum_{i=1}^{n} \delta_i^{-1} \left(\frac{\partial f}{\partial x_i}(\mathbf{x})\right)^2 \leq 0 \tag{5}$$

because $\delta_i, \delta_i^{-1} > 0$ for all i, each term of $\delta_i^{-1}\left(\frac{\partial f}{\partial x_i}(\mathbf{x})\right)^2$ is non-negative, therefore the minus sign of the sum $-\sum_{i=1}^{n} \delta_i^{-1}\left(\frac{\partial f}{\partial x_i}(\mathbf{x})\right)^2$ is non-positive. and given that $\nabla f(\mathbf{x}) \neq \mathbf{0}$, this implies that at least one of the partial derivative is non-zero, (i.e., $\frac{\partial f}{\partial x_i}(\mathbf{x}) \neq 0$ for some i), thus $\nabla f(\mathbf{x})^T \mathbf{d} = -\nabla f(\mathbf{x})^T \left(\mathbf{D}^{-1}\nabla f(\mathbf{x})\right) < 0$

and this verifies that $\mathbf{d}$ is a descent direction of f at $\mathbf{x}$

Assignment A3.3 Lipschitz Continuity

the function f is Lipschitz smooth / ∇f is Lipschitz continuous, if there exists a constant $L \geq 0$ (Lipschitz constant) such that:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \tag{6}$$

on $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$, all norms are equivalent to each other: if $||\cdot||_a$, $||\cdot||_b$ are two different norms on $\mathbb{R}^n$ ($\mathbb{R}^{m\times n}$), then there exists constants $c, C > 0$ such that:

$$c||\mathbf{x}||_a \leq ||\mathbf{x}||_b \leq C||\mathbf{x}||_a, \quad \forall \mathbf{x} \tag{7}$$

this implies if $\nabla^2 f(\mathbf{x})$ is unbounded in one (simple) norm, then it is also unbounded in any other norm ($\nabla^2 f(\mathbf{x})$ will be unbounded in $||\cdot||_2$ as well, no need to compute eigenvalues)

a): $f_1 : \mathbb{R}^3 \to \mathbb{R}$, $f_1(\mathbf{x}) := \frac{3}{2}x_1^2 + 2x_1x_2 - \frac{1}{3}x_3^3$

the gradient and the Hessian matrix of f_1:

$$\nabla f_1(\mathbf{x}) = \begin{pmatrix} 3x_1 + 2x_2 \\ 2x_1 \\ -x_3^2 \end{pmatrix} \quad \text{and} \quad \nabla^2 f_1(\mathbf{x}) = \begin{pmatrix} 3 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & -2x_3 \end{pmatrix} \tag{8}$$

we can simply compute that:

$$\left\|\nabla^2 f_1(\mathbf{x})\right\|_1 = \max\{5, 2, |-2x_3|\} \to \infty \quad \text{as } x_3 \to \infty \tag{9}$$

because $\nabla^2 f(\mathbf{x})$ is unbounded in $||\cdot||_1$ norm, and it is also unbounded in any other norm, therefore, $\left\|\nabla^2 f_1(\mathbf{x})\right\|$ is unbounded as well, and $\nabla \mathbf{f}_1$ **is not Lipschitz continuous**

b): $f_2 : \mathbb{R}^2 \to \mathbb{R}$, $f_2(\mathbf{x}) := \sqrt{1 + x_1^2 + x_2^2}$

denote that: $a(\mathbf{x}) = \left(1 + x_1^2 + x_2^2\right)^{-1/2}$

the gradient and the Hessian matrix of f_2:

$$\nabla f_2(\mathbf{x}) = \begin{pmatrix} x_1 a(\mathbf{x}) \\ x_2 a(\mathbf{x}) \end{pmatrix} \quad \text{and} \quad \nabla^2 f_2(\mathbf{x}) = \begin{pmatrix} a(\mathbf{x}) - x_1^2 a^3(\mathbf{x}) & -x_1x_2a^3(\mathbf{x}) \\ -x_1x_2a^3(\mathbf{x}) & a(\mathbf{x}) - x_2^2 a_3(\mathbf{x}) \end{pmatrix} \tag{10}$$

as the Hessian matrix $\nabla^2 f_2(\mathbf{x})$ is symmetric, the Frobenius norm of $\nabla^2 f_2(\mathbf{x})$ can be computed as:

$$\begin{aligned} \left\|\nabla^2 f_2(\mathbf{x})\right\|_F &= \sqrt{\left(a(\mathbf{x}) - x_1^2 a^3(\mathbf{x})\right)^2 + 2x_1^2x_2^2a^6(\mathbf{x}) + \left(a(\mathbf{x}) - x_2^2a^3(\mathbf{x})\right)^2} \\ &= \sqrt{2a^2(\mathbf{x}) - 2a^4(\mathbf{x})\left(x_1^2 + x_2^2\right) + \left(x_1^2 + x_2^2\right)^2 a^6(\mathbf{x})} \\ &\leq \sqrt{2 - \left(x_1^2 + x_2^2\right) + \left(x_1^2 + x_2^2\right)^2} \leq \sqrt{2 - 1 + 1} = \sqrt{2} \quad \text{when } x_1 = x_2 = 0 \end{aligned} \tag{11}$$

therefore, the Hessian matrix $\nabla^2 f_2(\mathbf{x})$ is bounded in the Frobenius norm, and thus $\nabla \mathbf{f}_2$ **is Lipschitz continuous**

for the Lipschitz constant L of ∇f_2, let $\left|\nabla^2 f_2 - \lambda \mathbf{I}\right| = 0$:

$$\begin{gathered} \left(a(\mathbf{x}) - x_1^2a^3(\mathbf{x}) - \lambda\right)\left(a(\mathbf{x}) - x_2^2a^3(\mathbf{x}) - \lambda\right) = x_1^2x_2^2a^6(\mathbf{x}) \\ \text{solve for } \lambda_1 = a(\mathbf{x}) \text{ and } \lambda_2 = a^3(\mathbf{x}) \end{gathered} \tag{12}$$

therefore, $\max|\lambda_i| = \lambda_1 = \left(1 + x_1^2 + x_2^2\right)^{-1/2} \leq 1$

to conclude: $\nabla \mathbf{f}_2$ **is Lipschitz continuous with Lipschitz constant** $\mathbf{L = 1}$

c): $f_3 : \mathbb{R}^2 \to \mathbb{R}$, $f_3(\mathbf{x}) := \ln(1 + x_1^2) + \ln(1 + x_2^2)$

the gradient and the Hessian matrix of f_3:

$$\nabla f_3(\mathbf{x}) = \begin{pmatrix} \frac{2x_1}{1+x_1^2} \\ \frac{2x_2}{1+x_2^2} \end{pmatrix} \quad \text{and} \quad \nabla^2 f_3(\mathbf{x}) = \begin{pmatrix} \frac{2-2x_1^2}{(1+x_1^2)^2} & 0 \\ 0 & \frac{2-2x_2^2}{(1+x_2^2)^2} \end{pmatrix} \tag{13}$$

for $\left\|\nabla^2 f_3(\mathbf{x})\right\|_2 = \max\left\{\left|\frac{2-2x_1^2}{(1+x_1^2)^2}\right|, \left|\frac{2-2x_2^2}{(1+x_2^2)^2}\right|\right\}$, which is equivalent to maximize one of the x in $\left|\frac{2-2x^2}{(1+x^2)^2}\right|$:

$$\begin{aligned} \left\|\nabla^2 f_3(\mathbf{x})\right\|_2 &= \max\left|\frac{2-2x^2}{(1+x^2)^2}\right| \\ &= \max\left|-\left(\frac{2}{(1+x^2)} + \frac{4}{(1+x^2)^2}\right)\right| \\ &= \max\left|-\left(a + a^2\right)\right|, \text{ where: } a = \frac{2}{1+x^2} \leq 2 \end{aligned} \tag{14}$$

this implies that $\left\|\nabla^2 f_3(\mathbf{x})\right\|_2$ is bounded by 2, thus $\left\|\nabla^2 f_3(\mathbf{x})\right\|_2$ as well, and $\nabla \mathbf{f}_3$ **is Lipschitz continuous**

for the Lipschitz constant L of ∇f_3, let $\left|\nabla^2 f_3 - \lambda \mathbf{I}\right| = 0$

$$\begin{gathered} \left|\nabla^2 f_3(\mathbf{x}) - \lambda \mathbf{I}\right| = \begin{pmatrix} \frac{2-2x^2}{(1+x^2)^2} - \lambda & 0 \\ 0 & \frac{2-2x^2}{(1+x^2)^2} - \lambda \end{pmatrix} \\ \text{solve for } \lambda_i = \frac{2-2x^2}{(1+x^2)^2} \leq 2 \end{gathered} \tag{15}$$

to conclude: $\nabla \mathbf{f}_3$ **is Lipschitz continuous with Lipschitz constant** $\mathbf{L = 2}$