THỰC HÀNH TOÁN CAO CẤP

TÀI LIỆU PHỤC VỤ SINH VIÊN NGÀNH KHOA HỌC DỮ LIỆU

Nhóm biên soạn: TS. Hoàng Lê Minh – Khưu Minh Cảnh – Hoàng Thị Kiều Anh – Lê Thị Ngọc Huyên – …

TP.HCM – Năm 2019

MỤC LỤC

CHƯƠNG 5: BỔ SUNG KHÁI NIỆM CƠ BẢN, MỘT SỐ ỨNG DỤNG CỦA GIẢI TÍCH

Mục tiêu:

- *Các khái niệm cơ bản, giới thiệu hàm nhiều biến*
- *Các ứng dụng của giải tích trong cuộc sống.*

Nội dung chính:

1. Các khái niệm cơ bản trong giải tích

1.1. Phép lặp để giải phương trình

Trong tính toán, phép lặp là một phương pháp kỹ thuật để giải phương trình. Ví dụ sau liên quan đến số gọi là Tỉ số Vàng (Golden Ratio) bằng phép lặp for trong Python. Vấn đề, chúng ta cần giải phương trình sau:

$$x = \sqrt{1 + x}$$

Để giải phương trình trên, bước đầu tiên chúng ta chọn 1 nghiệm, nghiệm đó được gọi là nghiệm ban đầu. Và tiếp tục quá trình lặp để tìm các nghiệm chính xác hơn.

Thực hành 1: Lặp để tìm nghiệm

```
>>> x = 3

>>> print  (x)

…… ← sinh viên điền giá trị vào

>>> x = math.sqrt(1+x)

>>> print (x)

…… ← sinh viên điền giá trị vào

>>> x = math.sqrt(1+x)

>>> print (x)

…… ← sinh viên điền giá trị vào
```

>>> x = math.sqrt(1+x)

>>> print (x)

……. ← sinh viên điền giá trị vào

Sinh viên thực hiện các lệnh trên đến khi x không thay đổi và cho biết cần bao nhiêu lần thực hiện phép gán: x = sqrt(x)……?

Thực hành 2: Lặp bằng while để tìm nghiệm

Chúng ta có thể thử viết lệnh lặp để giải như sau:

```
>>> import math
>>> x = 3
>>> lap = 1
>>> while (x != math.sqrt(x+1)):
        x = math.sqrt(x+1)
        lap = lap  +1                    # lưu ý: enter 2 lần để thoát vòng lặp while
```

>>> x

……. ← sinh viên điền giá trị vào

>>> lap

……. ← sinh viên điền giá trị vào

Từ đó, chúng ta thấy qua các bước lặp, x chính là các giá trị như sau: 3, $\sqrt{1+3}$, $\sqrt{1+\sqrt{1+3}}$, $\sqrt{1+\sqrt{1+\sqrt{1+3}}}$, ... và x sẽ hội tụ tại một số bước lặp (mặt khác cũng do sai số của ngôn ngữ Python). Ở đây, chúng ta gọi điểm hội tụ là những điểm cố định (fixed point).

Thực hành 3: Giải phương trình bằng hàm solve trong sympy

Lưu ý: với sympy, chúng ta có thể giải phương trình $x = \sqrt{1+x}$

```
>>> import sympy as sp
>>> from sympy import Symbol
>>> x = Symbol('x')
>>> sp.solve(x-sp.sqrt(1+x),x)
```

……. ← sinh viên điền giá trị vào

1.2. Vector

Trong giải tích, hình học hoặc đại số, khái niệm vector là khái niệm cơ bản nhất. Một vector là một bộ số để chỉ vị trí, hướng và cung cấp thông tin về độ lớn của một sự vật hiện tượng theo hướng.

Với không gian 1 chiều, vector là bộ số chỉ gồm 1 số. Với không gian vector mặt phẳng Oxy 2 chiều, vector là bộ số gồm 2 số, thông thường, số đầu tiên chỉ giá trị x và số sau chỉ giá trị y.

Gói **numpy** trong Python hỗ trợ xử lý vector với kiểu dữ liệu **numpy.array**.

Thực hành 4: Các phép toán trên vector

>>> import **numpy** as np

>>> v1 = np.array([1., 2., 3.]) # tạo vector 3 chiều

>>> v2 = np.array([2., 1., 0.])

>>> v3 = v1 + v2   # cộng vector

>>> v3

……. ← sinh viên điền kết quả vào

Thực hiện phép toán trên vector:

>>> 3*v1 + 2*v2

……. ← sinh viên điền kết quả vào

* Lưu ý: kiểu **numpy.array** sẽ khác với kiểu dữ liệu **list** trong Python.

Thử nghiệm ví dụ sau (trên đối tượng list)

>>> [1, 2, 3] + [2, 1, 0]

……. ← sinh viên điền kết quả vào

>>> 3*[1, 2, 3] + 2*[2, 1, 0]

……. ← sinh viên điền kết quả vào

Dễ dàng thấy, phép cộng trên list không phải là phép cộng vector.

Ghép nối (Concatenating) 2 hoặc nhiều vector:

>>> v4 = np.hstack([v1, v2])

……. ← sinh viên điền kết quả vào

* Phép nhân vô hướng 2 vector:

>>> np.dot(v1, v2)

……. ← sinh viên điền kết quả vào

Tính toán giá trị sin của vector:

>>> angles = np.linspace(0, np.pi/2, 5)

>>> angles

……. ← sinh viên điền kết quả vào

>>> np.sin(angles)

……. ← sinh viên điền kết quả vào

Tuy nhiên, hàm xử lý sin trong gói sympy sẽ không hỗ trợ việc tính toán trên toàn bộ vector chứa dữ liệu. Thử nghiệm:

>>> import sympy as sy

>>> sy.sin(angles)

……. ← sinh viên điền kết quả (tên lỗi)

Do đó, để sử dụng hàm sin của sympy, chúng ta có thể viết một đoạn chương trình lặp với vector mới được xây dựng sẵn tạm thời:

>>> from sympy import sin as sysin

>>> angles = np.linspace(0, np.pi/2, 5)

>>> sinangle = **np.zeros**(5) # tương đương >>> sinangle = np.array([0.0, 0.0, 0.0, 0.0, 0.0])

>>> len(angles) # kiểm tra kích thước/số chiều của vector angles.

>>> for i in range(len(angles)):

    sinangle[i] = sysin(angles[i]) # lưu ý: ở đây phải enter 2 lần để thoát khỏi vòng for.

>>> sinangle

…… ← sinh viên điền kết quả vào

Lưu ý một số lệnh tạo vector và ma trận:

Yêu cầu xử lý	Lệnh
	Tạo vector[*]
Tạo vector 0 có n phần tử	**np.zeros(n)**
Tạo vector 1 có n phần tử	**np.ones(n)**
Tạo vector có n phần tử ngẫn nhiên từ 0 đến 1	**np.rand(n)**
Tạo vector rỗng có n phần tử	**np.empty(n)**
	Tạo ma trận
Tạo ma trận đơn vị	**np.eye(n)**
Tạo ma trận toàn 0	**np.zeros([n,m])**
Tạo ma trận ngẫu nhiên	**np.rand(n,m)**
Tạo ma trận trống	**np.empty([n, m])**
Tích 2 vector hoặc 2 ma trận	**np.dot(ma_tran1, ma_tran2)**

*([*] Giả định thực thi đã lệnh >>> import numpy as **np** từ trước)*

Thực hành 5: Tính tổng các tích giữa ma trận và vector. Giả định gói thư viện numpy được đưa vào hệ thống có tên là np. Dạng:

$$\sum_j a_{ij}x_j \leftrightarrow \boldsymbol{np.dot}(A, x)$$

Minh họa thực hiện trên Python:

```
>>> import numpy as np
>>> goc = np.pi/3
>>> A = np.array([ [np.cos(goc), -np.sin(goc)],
                   [np.sin(goc),  np.cos(goc)] ])
>>> V = np.array([1. , 0. ])
>>> Y = np.dot(A, V)
>>> Y
```

…… ← sinh viên điền kết quả vào

* Lưu ý 1: Tích ma trận với vector hoặc ma trận cần lưu ý đến thứ tự.

* Lưu ý 2: Không nên sử dụng dấu * thay cho phép toán **dot**. Vì dấu * sẽ tính toán tích tại từng phần tử (elementwise) của mảng/ma trận thay cho phép nhân ma trận được định nghĩa trong toán học.

* Lưu ý 3: Một số dạng toán học và lệnh tương ứng trong Python

- Tích vector - vector: $s = \sum_i x_i y_i \leftrightarrow s = np.dot(x, y)$.

- Tích ma trận – ma trận: $C_{ij} = \sum_k A_{ik} B_{kj} \leftrightarrow s = np.dot(A, B)$.

- Tích vector - ma trận: $y_j = \sum_i x_i A_{ij} \leftrightarrow y_j = np.dot(x, A)$.

2. Một số khái niệm trong giải tích cần biết

Dưới đây là một số khái niệm trong giải tích sinh viên cần biết

2.1. Không gian hai chiều và nhiều chiều

Như đề cập trong chương 1, khác với trục số một chiều, không gian hai chiều gồm 2 thành phần, thường đặt là x và y (hoặc là u và v). Ví dụ: hiện tại, chúng ta đang ở trên thế giới có không gian 3 chiều là x, y, z và nếu tính theo chiều thời gian thì chúng ta có 4 chiều!

Theo đó, hàm số nhiều biến được hiểu là hàm số có trên không gian nhiều chiều. Ví dụ: vị trí của một chiếc tàu lửa/xe lửa sẽ có 3 biến là (x,y, t) là với x, y là 2 biến vị trí và t là thời gian cụ thể (có thể thêm z nếu chúng ta quan tâm đến độ cao của tàu lửa).

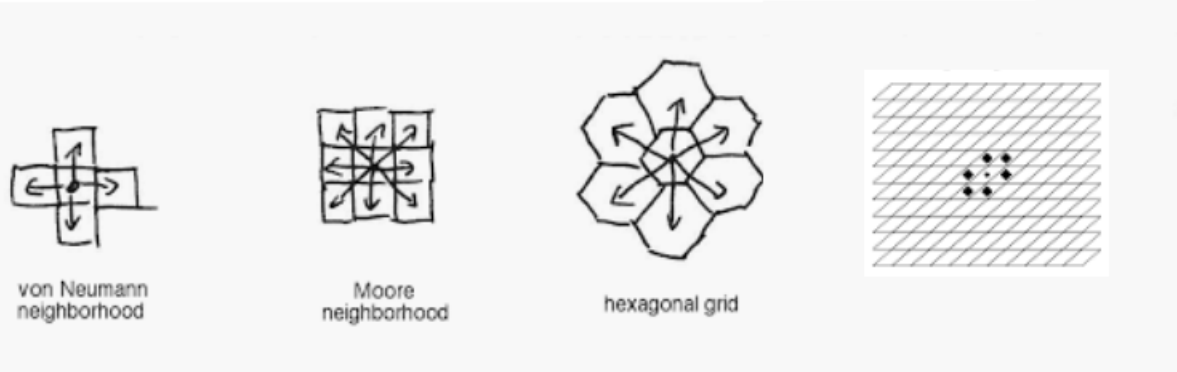
Đối với các hệ thống cơ học, một số mô hình lựa chọn mỗi thiết bị là một chiều với giá trị là các trạng thái của thiết bị đó. Khi đó, chúng ta có thể sử dụng những vector nhiều chiều để biểu diễn.

Để thể hiện hàm nhiều biến, trong sympy có biến nào trong hàm thì chúng ta phải khai báo nó như một đối tượng Symbol.

2.2. Các lân cận 4, 8

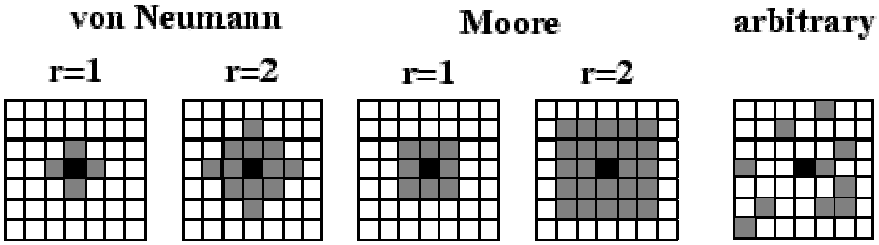
Trong tính toán, một số không gian cần rời rạc hóa thành các vị trí trên không gian. Thông thường, người ta sẽ lưu vào thành bảng hoặc ma trận và nhiều trường hợp là những lưới đều.

Với một lưới đều, trong giải tích, một vị trí sẽ có ít nhất hai loại lân cận (neighbourhood). Đó là lân cận 8 (còn gọi là Moore neighborhood) và lân cận 4 (còn gọi là lân cận Von Neumann neighborhood). Ngoài ra, một số dạng lân cận khác sẽ được sử dụng tùy theo các ứng dụng cụ thể. Ví dụ: lân cận hình tổ ong thường ứng dụng trong lĩnh vực truyền thông/phát tín hoặc sóng điện thoại. Hình bên dưới minh họa các lân cận:



Mỗi loại lân cận sẽ hỗ trợ giải quyết tính toán những loại bài toán khác nhau về đặc trưng tính toán khoảng cách, đặc biệt có liên quan đến vị trí giữa các điểm mà trên quan điểm triết học biện chứng là nơi đó có các đối tượng sẽ có sự tương tác với nhau!

Hình bên dưới sẽ cho thấy với bán kính r=1 và r=2 của các lân cận Von Neumann và Moore sẽ khác nhau về số lượng điểm lân cận:



Lân cận trong giải tích sẽ là những nền tảng trong các ứng dụng về y khoa (như việc loại bỏ tế bào ung thư…), các ứng dụng trong ngập lụt (vị trí nước có thể chảy đến),… hoặc các ứng dụng về tìm kiếm sự ảnh hưởng của một sự vật/hiện tượng có tính tự phát triển (là các ứng dụng CA – Cellular Automata) như: lan truyền nhiệt, lửa cháy, vi khuẩn phát sinh, lan truyền không khí/nước ô nhiễm, sự sinh sôi nẩy nở của vi khuẩn hoặc một hiện tượng xã hội gì đó…

Với vector $\vec{y} = (v_1, v_2, \dots v_n)$, chúng ta có phép tính vi phân như sau:

$$dy = (\frac{v_2 - v_1}{dx}, \frac{v_3 - v_2}{dx}, \dots, \frac{v_n - v_{n-1}}{dx})$$

Nghĩa là vector tạo thành sẽ giảm đi 1 chiều.

Thực hành 6: Tính toán đạo hàm trên 1 vector dữ liệu

+ Trường hợp dx cố định:

>>> **from** numpy **import** diff

>>> dx = 0.1

>>> y = [1, 2, 3, 4, 4, 5, 6]

>>> dy = diff(y)/dx

>>> dy

…… ← sinh viên điền kết quả vào

>>> z = np.array([1, 2, 3, 4, 4, 5, 6])

>>> dz = diff(z)/dx

>>> dz

…… ← sinh viên điền kết quả vào

Sinh viên hãy so sánh 2 kết quả tính và nhận xét: ……

+ Trường hợp dx là một dãy số:

>>> from numpy import diff

>>> x = [.1, .2, .5, .6, .7, .8, .9]

>>> y = [1, 2, 3, 4, 4, 5, 6]

>>> dydx = diff(y)/diff(x)

>>> print (dydx)

…… ← sinh viên điền kết quả vào

2.3. Các tiêu chuẩn đo khoảng cách (distance)

Từ việc xác định các lân cận và trên tiêu chuẩn không gian liên tục, nếu có 2 điểm $A(x_A, y_A)$ và $B(x_B, y_B)$ thì khoảng cách $d(A, B)$ của 2 điểm A và B được xác định theo cách tiêu chuẩn sau:

- Với không gian Euclide:

$$d(A,B) = \sqrt[2]{(x_A - x_B)^2 + (y_A - y_B)^2}$$

- Với lân cận 4, đây thực sự giống như là khoảng cách các khối nhà hình bàn cờ (hay còn gọi là khoảng cách Mahattan):

$$d(A,B) = |x_A - x_B| + |y_A - y_B|$$

- Với lân cận 8:

$$d(A,B) = \max(|x_A - x_B|, |y_A - y_B|)$$

Thực hành 7: Một số hàm xử lý phẳng (không gian Euclide)

>>> from **sympy.geometry** import *

 # Tạo các điểm P1, P2, P3 và P4

>>> P1 = Point(0, 0)

>>> P2 = Point(3, 4)

>>> P3 = Point(2, -1)

>>> P4 = Point(-1, 5)

 # Tạo 2 đoạn đường S1 và S2:

>>> S1 = Segment(P1, P2)

>>> S2 = Segment(P3, P4)

 # Kiểm 3 điểm thẳng hàng:

>>> Point.**is_collinear**(P1, P2, P3)

…… ← sinh viên điền kết quả vào

Độ dài của đoạn đường S1

>>> S1.**length**

…… ← sinh viên điền kết quả vào

 # Lấy trung điểm của đoạn 2:

>>> S2.**midpoint**

…… ← sinh viên điền kết quả vào

 # Tính độ dốc của đường S1

>>> S1.slope

…… ← sinh viên điền kết quả vào

 # Tìm vị trí giao nhau giữa hai đoạn đường

>>> S1.intersection(S2)

…… ← sinh viên điền kết quả vào

 # Góc giữa hai đoạn đường

>>> Segment.angle_between(S1, S2)

…… ← sinh viên điền kết quả vào

 # Kiểm đoạn đường S1 có chứa điểm P3 hay không?

>>> S1.contains(P3)

…… ← sinh viên điền kết quả vào

 # Lập và xem phương trình đường thẳng L1 đi qua 2 điểm P1 và P2:

>>> L1 = Line(P1, P2)

>>> L1.equation()

…… ← sinh viên điền kết quả vào

Kiểm tính song song:

>>> L1.is_parallel(S1)

…… ← sinh viên điền kết quả vào

>>> L1.is_parallel(S2)

…… ← sinh viên điền kết quả vào

3. Ôn luyện giới hạn, đạo hàm và tích phân

3.1. Giới hạn

Thực hành 8: Bằng phương pháp đồ thị, hãy cho biết giới hạn của hàm số sau

$$\lim_{x \to 0} x sin(\frac{1}{x})$$

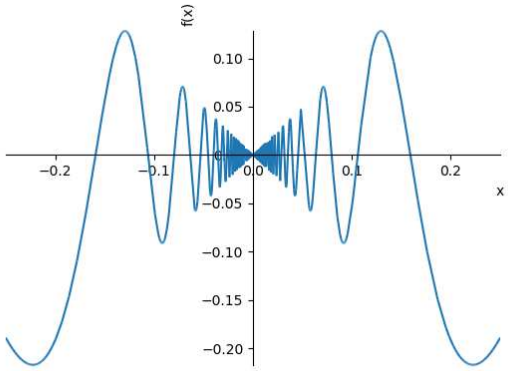
Gợi ý:

Hãy vẽ đồ thị của các hàm: $g(x) = -x$, $f(x) = xsin(\frac{1}{x})$, $h(x) = x$. Nhận định: giới hạn của cả hai hàm $g(x)$ và $h(x)$ đều bằng 0. Sau đó sử dụng định lý "kẹp" để suy ra giới hạn của hàm $f(x)$ đã cho.

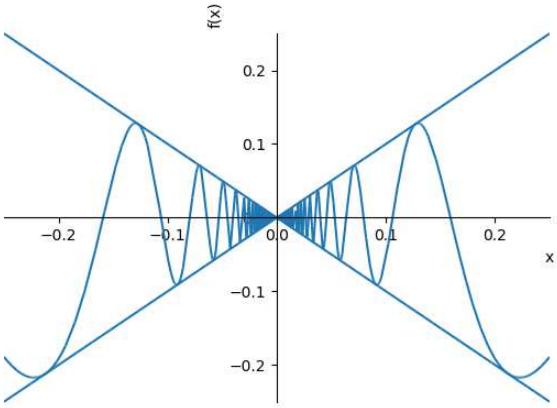
Giải bằng Python:

import sympy

```
from sympy import *

x = Symbol('x')

f = x * sin(1/x)

c = Symbol('c')

delta = Symbol('delta')

c = 0

delta = 1/4

sympy.plot(f,(x, c - delta, c + delta))
```




```
sympy.plot(f, abs(x), -abs(x),(x, c - delta, c + delta))
```



Dựa vào đồ thị trên, ta thấy giới hạn của hàm đã cho là 0.

3.2. Đạo hàm

Nhắc lại về các bước giải bài toán tối ưu hóa

1. Xác định các biến và các hằng số, vẽ đồ thị phù hợp, hiểu rõ cần tìm cực đại, cực tiểu của hàm nào.
2. Viết công thức cho hàm muốn tìm cực đại, cực tiểu.
3. Viết hàm phụ thuộc vào một biến duy nhất: $f(x)$
4. Tìm $f'(x)$ và giải $f'(x) = 0$. Kiểm tra tất cả các giá trị tới hạn và điểm đầu mút để tìm cực trị.

Thực hành 9: Tìm giá trị cực đại của hàm

$$f(x) = -x^2 + 4x - 3$$

trên khoảng $[0,4]$ và vẽ đồ thị hàm số $f(x)$.

Gợi ý:

Trước hết, lưu ý rằng $f'(x) = -2x + 4 = 0$ khi $x = 2$ và $f(2) = 1$.

Tiếp theo, $f(x)$ xác định với mọi x, do đó sẽ không có cực trị nào khác.

Ngoài ra, $f(0) = -3$ và $f(4) = -3$.

Vậy, giá trị lớn nhất của hàm số $f(x)$ trên đoạn $[0,4]$ là $f(2) = 1$

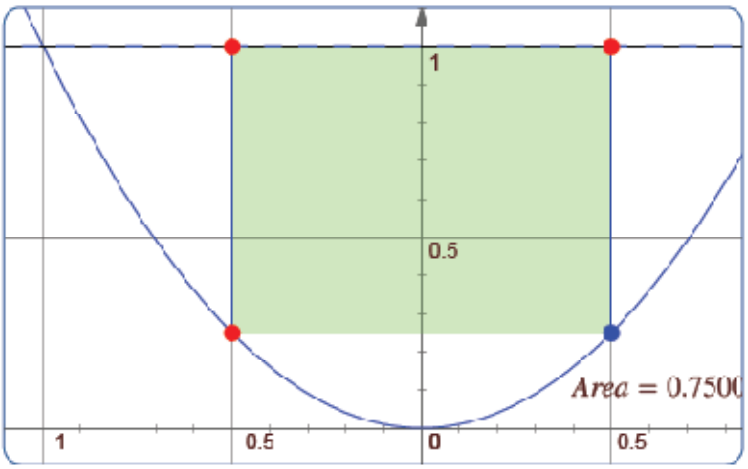
Giải:

```
>>> from sympy import Symbol, solve, Derivative
>>> x = Symbol('x')
>>> f = -x**2+4*x-3
>>> d1=Derivative(f, x).doit()
>>> cuctri = solve(d1)
>>> cuctri
[2]
>>> A = cuctri[0]
>>> d2 = Derivative(d1, x).doit()
```

```
>>> d2.subs({x:A}).evalf()
…… ← sinh viên điền vào
>>> x_min=0
>>> x_max=4
>>> f.subs({x:A}).evalf()
…… ← sinh viên điền vào
>>> f.subs({x:x_min}).evalf()
…… ← sinh viên điền vào
>>> f.subs({x:x_max}).evalf()
…… ← sinh viên điền vào
>>> # so sánh các giá trị, ta thấy GTLN là f(2)=1
```

Thực hành 10: Tìm diện tích cực đại để lắp các tấm pin mặt trời trên một sân phẳng có hình parabol

Tìm hình chữ nhật có diện tích lớn nhất nằm trong đồ thị của parabol $y = x^2$ và đường thẳng $y = a$ với a là một giá trị hằng số (như hình bên trên).



Gợi ý:

Điều chúng ta cần là tìm giá trị lớn nhất của một hàm $A(x)$ thể hiện diện tích của hình chữ nhật. Phần khó nhất trong bài này chính là tìm được vị trí x.

Giả định $x > 0$, vị trí của 4 điểm của hình chữ nhật 1, 2, 3 và 4 (với 1 là vị trí thấp nhất bên phải của hình chữ nhật) lần lượt theo x là $1(x, x^2), 2\ (x, a), 3\ (-x, a), 4\ (-x, x^2)$.

Từ đó, chúng ta dễ dàng xác định được diện tích của hình chữ nhật là hàm số:

$$A(x) = (2x)(a - x^2) = -2x^3 + 2ax, x \in [0, \sqrt{a}]$$

Sử dụng đạo hàm để tính toán giá trị lớn nhất của $A(x)$ như sau:

$$0 = A'(x) = -6x^2 + 2a$$

$$\leftrightarrow x = \sqrt{\frac{a}{3}}$$

Đó là giá trị tới hạn duy nhất của hàm $A(x)$. Chúng ta có thể thử lại các giá trị biên trong miền xác định của x.

Từ đó, sinh viên hãy cho biết giá trị cực đại của diện tích hình chữ nhật.

Code Python:

<sinh viên tự nghiên cứu viết>

Thực hành 11: Bài toán Black Friday - Tìm giá tốt cho đợt giảm giá điện thoại iPhone 11 cuối năm

Bạn muốn bán n điện thoại Iphone 11 sao cho lợi nhuận là cao nhất. Bộ phận nghiên cứu thị trường của công ty cho thấy nếu bán với giá \$1500 thì có thể bán được 5000 chiếc và nếu cứ giảm \$100 cho mỗi điện thoại thì sẽ bán thêm được 1000 chiếc. Giả sử chi phí vốn là cố định (chi phí khởi nghiệp) bằng \$2.000.000 và tổng chi phí mua về (chi phí biên) cho mỗi điện thoại là \$500. Tìm giá bán cho mỗi điện thoại (x) và tổng số điện thoại bán được (n) để lợi nhuận là tối đa. Tìm lợi nhuận tối đa đó.

Gợi ý:

Trước hết, ta chuyển sang bài toán tìm giá trị lớn nhất của hàm lợi nhuận $P(x)$, là hàm phụ thuộc vào giá bán x. Khi đó:

$$P(x) = nx - 2000000 - 500n$$

Theo các mô tả trên, số điện thoại bán được là:

$$n = 5000 + 1000\frac{(1500 - x)}{100}$$

Lưu ý: Do cứ giảm \$100 thì bán thêm được 1000 chiếc, nên nếu giảm $1500 - x$ thì sẽ bán thêm được $1000\frac{(1500-x)}{100}$ chiếc

Thay n vào hàm lợi nhuận:

$$P(x) = -10x^2 + 25000x - 12000000$$

Ta cần tìm cực đại với x từ 0 tới 1500 nên sẽ lấy đạo hàm:

$$P'(x) = -20x + 25000$$

Do đó hàm đạt cực trị tại $x = 1250$.

Vì $P''(x) = -20 < 0$ nên tại đây, hàm đạt cực đại địa phương.

Ta có: $P(0) = -12000000$, $P(1250) = 3625000$ và $P(1500) = 3000000$.

Do đó, cực đại địa phương chính là cực đại toàn cục.

Và lợi nhuận tối đa là $P = \$3625000$, đạt được khi giá bán là $x = \$1250$.

Code Python:

<sinh viên tự nghiên cứu viết>

Gợi nhớ các hàm cần sử dụng:

- Hàm Derivative để tính đạo hàm, cụ thể ra số là doit()
- Hàm solve để giải phương trình tìm các cực trị
- Hàm subs để thay thế
- Hàm evalf để tính giá trị

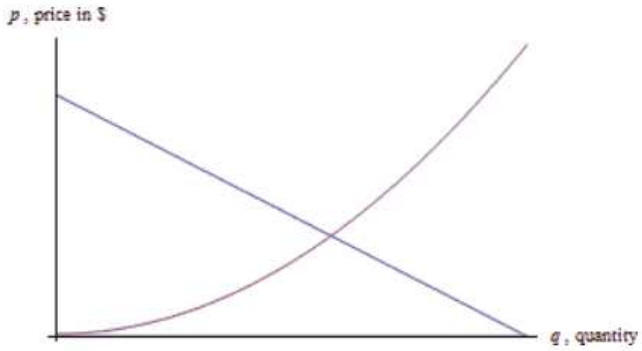
Đáp án: Giá trị \$ lớn nhất thu về là 3.625.000 (!?)

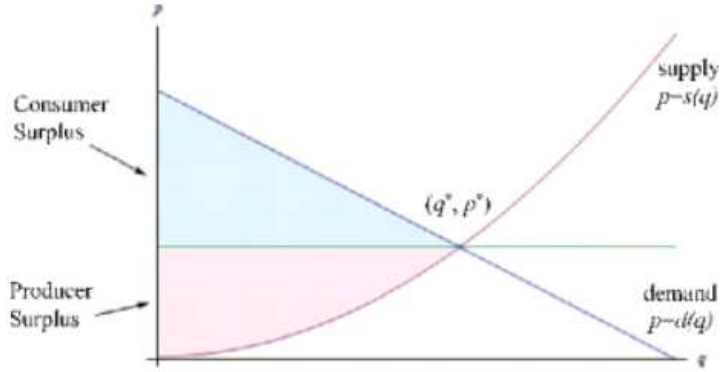
3.3. Tích phân

Thực hành 12: Bài toán Cung - cầu

Trên thực tế, nhu cầu sẽ giảm khi giá của một sản phẩm gia tăng. Ngược lại, khi giá càng tăng, sản phẩm sẽ có nhiều trên thị trường. Với đồ thị thể hiện giữa nhu cầu và cung cấp theo hai đại lượng giá (p) và số lượng sản phẩm được bán (q) được thể hiện và chúng ta dễ dàng thấy sự nghịch biến. Trong đồ thị đó, điểm cân bằng (equilibrium) là điểm (q^*, p^*) giao điểm giữa hai xu hướng chính là giá trị cân bằng của thị trường.

Với $p = d(q)$ là hàm nhu cầu (hàm giảm), $p = s(q)$ là hàm cung cấp (hàm tăng)





Từ đồ thị, chúng ta dễ thấy rằng để đạt đến trạng thái cân bằng, chúng ta có thể phân tích với lượng tiền "dư" (surplus) từ bên "cầu" (consumer) và bên "cung cấp" (producer) như sau:

$$Tiền\ dư\ bên\ cung = \int_0^{q^*} (d(q) - p^*)dq = \int_0^{q^*} d(q)dq - \int_0^{q^*} p^*dq = -p^*q^* + \int_0^{q^*} d(q)dq$$

Tương tự, công thức tính $Tiền\ dư\ bên\ cầu$ sẽ là:

$$Tiền\ dư\ bên\ cầu = \int_0^{q^*} \big(p^* - s(q)\big)dq = \int_0^{q^*} p^*dq - \int_0^{q^*} s(q)dq -= p^*q^* - \int_0^{q^*} s(q)dq$$

Lưu ý: đơn vị của hai phép toán trên là tiền.

Bài toán thực tế:

Honda dự kiến bán xe SH 2019. Sau vài tháng thăm dò, Honda nhận ra các quy luật như sau:

+ Quy luật cầu: $p = d(q) = -0.8q + 150$, nghĩa là giá (150-0.8) triệu sẽ bán được 1 xe/tháng ($q = 1$), (150-1.6) triệu mỗi tháng sẽ bán được 2 xe/tháng (tương ứng với $q = 2$),…

+ Quy luật cung: $p = s(q) = 5.2q$, nghĩa là mỗi xe SH bán ra Honda sẽ được lợi nhuận là 5.2 triệu đồng.

Các bạn sinh viên hãy giúp:

1. Xác định giá nên bán trên thị trường (trạng thái cân bằng). Vẽ đồ thị.
2. Tại thời điểm cân bằng, tổng số tiền người mua phải trả dư/cao hơn (consumer surplus).
3. Tại thời điểm cân bằng, tổng số tiền mà sản phẩm thu dư được (producer surplus).

Giải:

1. Giải phương trình $-0.8q + 150 = 5.2q$ sẽ tìm được nghiệm $q = 25$, nghĩa là cân bằng khi mỗi tháng bán được 25 xe với giá $q = -0.8(25) + 150 = 130$ triệu đồng.
2. Tổng số tiền người mua phải trả dư/cao hơn (consumer surplus):
$$\int_0^{25} (-0.8q + 150)dq - 130 \times 25 = 250\ triệu$$
3. Tổng số tiền mà sản phẩm thu dư được (producer surplus):
$$130 \times 25 - \int_0^{25} 5.2qdq = 1625\ triệu$$

Sinh viên tự thể hiện các lệnh tính toán trên bằng Python!

Thực hành 13: Bài toán lợi nhuận đầu tư

Một nhóm khởi nghiệp trình bày ý tưởng đầu tư một máy sản xuất trà sữa như sau: giá thành của máy là 650 triệu. Mỗi năm máy sẽ giúp thu lợi 70 triệu và từ năm thứ 2, mỗi năm kế tiếp sẽ tăng thêm 8 triệu. Biết rằng tiền sẽ tăng 1.7% mỗi năm với lãi suất ngân hàng. Với tư cách là nhà đầu tư (shark), bạn có thấy ý tưởng khởi nghiệp này có lợi nhuận hay không?

Giải:

Gọi t là thời gian tính theo năm. Trong bài toán trên, t = 0 là thời điểm ban đầu, t=1 là năm đầu tiên. Giá trị thu lại ở năm đầu là 70 triệu. Thu nhập các năm tiếp theo sẽ tăng thêm 8 đồng. Nghĩa là tuân theo hàm F(t)=70+8(t-1)=62+8t. Để tính giá trị đến năm thứ 8, chúng ta có hai tổng tích phân như sau: tổng lợi nhuận của năm đầu và tổng lợi nhuận từ năm 1 đến năm 8.

$$Lợi\ nhuận = \int_0^1 70e^{-0.017t}dt + \int_1^8 (62 + 8t)e^{-0.017t}dt \approx 701{,}66\ triệu$$

Giá trị thu lại cao hơn so với giá của máy nên xét riêng về lợi nhuận thì bạn có thể đồng ý đầu tư.

Sinh viên tự thể hiện các lệnh tính toán trên bằng Python!

BÀI TẬP CHƯƠNG 5

Bài tập 1:

Trong tất cả các hình chữ nhật có diện tích 100m2, hình nào có chu vi nhỏ nhất?

Gợi ý:

Trước hết, ta gọi $x(x > 0)$ là chiều rộng của hình chữ nhật, $y(y > 0)$ là chiều dài. Lúc đó, diện tích của hình là $x.y = 100$. Do đó, $y = \frac{100}{x}$.

Chu vi của hình chữ nhật là $f(x) = 2(x + y) = 2\left(x + \frac{100}{x}\right) = 2x + 2.\frac{100}{x}$.

Tiếp theo, ta có $f'(x) = 2 - \frac{200}{x^2} = 0$ khi $x = 10$ hoặc $x = -10$. Tuy nhiên, x chỉ nhận giá trị dương nên ta chọn $x = 10$.

Vì $f(x)$ xác định trên khoảng $(0, \infty)$ nên không có giá trị tới hạn nào khác và cũng không có các điểm đầu mút. Liệu có cực tiểu, cực đại địa phương hay không tại $x = 10$.

Đạo hàm cấp 2 là $f''(x) = \frac{400}{x^3}$ và $f''(10) > 0$. Do đó, có một giá trị cực tiểu địa phương. Do chỉ có một giá trị tới hạn duy nhất, nên đây cũng là cực tiểu toàn cục.

Như vậy, hình chữ nhật có diện tích 100m2 thì hình vuông 10x10 là hình có chu vi nhỏ nhất.

Sinh viên tự viết các câu lệnh Python!

Bài tập 2:

Giả sử bạn muốn đến một điểm A (nằm trên cát) từ một con đường gần đó (xem hình 6.1.5). Giả sử đường đi thẳng và b là khoảng cách từ A đến điểm C (C là điểm gần nhất tính từ A đến con đường). Đặt v là tốc độ của bạn trên đường và w, nhỏ hơn v, là tốc độ của bạn trên cát. Hiện tại bạn đang ở điểm D, cách C một khoảng a . Tại điểm B nào (B là điểm giữa D và C) bạn nên tách đường để băng qua cát sao cho thời gian đến được A là ít nhất?

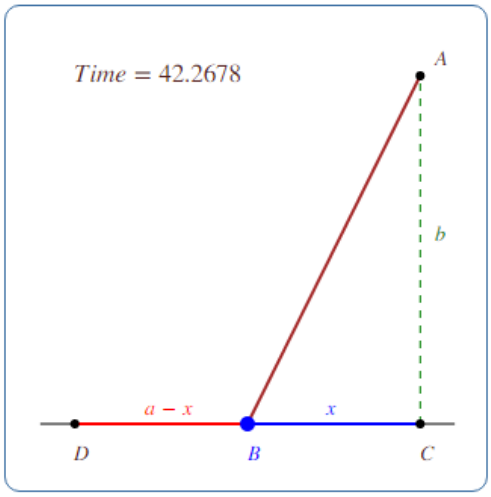


Figure 6.1.5. *Drag the blue point to minimize travel time.*

Gợi ý:

Gọi x là khoảng cách từ B đến C. Ta cần tối thiểu tổng thời gian di chuyển. Nhớ rằng khi di chuyển với tốc độ không đổi, thời gian bằng quãng đường chia vận tốc.

Ta di chuyển một khoảng DB với vận tốc v, sau đó là BA với vận tốc w. Vì $DB = a - x$, theo Pythagore, ta có $BA = \sqrt{x^2+b^2}$, tổng thời gian di chuyển là: $f(x) = \frac{a-x}{v} + \frac{\sqrt{x^2+b^2}}{w}$

Ta cần tìm giá trị nhỏ nhất của hàm f với x giữa 0 và a.

Ta có:

$$0 = f'(x) = -\frac{1}{v} + \frac{x}{w(\sqrt{x^2+b^2})}$$

$$w\sqrt{x^2+b^2} = vx$$

$$w^2(x^2+b^2) = v^2x^2$$

$$w^2b^2 = (v^2 - w^2)x^2$$

$$x = \frac{wb}{\sqrt{v^2-w^2}}$$

Lưu ý rằng mặc dù a không có trong biểu thức cuối cùng nhưng nó có liên quan, vì ta chỉ quan tâm đến các giá trị tới hạn trong $[0, a]$ và $\frac{wb}{\sqrt{v^2-w^2}}$ có thể nằm trong khoảng này hoặc không.

Nếu có nằm trong này, ta tìm đạo hàm cấp 2:

$$f''(x) = \frac{b^2}{w(x^2+b^2)^{3/2}}$$

Giá trị này luôn dương nên sẽ luôn có cực tiểu địa phương tại điểm tới hạn, đó cũng là cực tiểu toàn cục.

Nếu giá trị tới hạn không nằm trong khoảng này thì nó sẽ lớn hơn a. Trong trường hợp này giá trị nhỏ nhất đạt được tại một trong các điểm đầu mút.

Ta có thể tính:

$$f(0) = \frac{a}{v} + \frac{b}{w}$$

$$f(a) = \frac{\sqrt{a^2+b^2}}{w}$$

Rất khó để xác định giá trị nào nhỏ hơn bằng cách so sánh trực tiếp, do không biết được giá trị của v, w, a và b. Dù vậy, ta vẫn có thể tìm được giá trị nhỏ nhất.

Vì $f''(x) > 0$ nên hàm $f'(x)$ luôn tăng. Ngoài ta, tại $\frac{wb}{\sqrt{v^2-w^2}}$, đạo hàm bằng 0, do đó với những x nhỏ hơn giá trị tới hạn, đạo hàm luôn âm. Nghĩa là $f(0) > f(a)$. Như vậy, giá trị nhỏ nhất đạt tại $x = a$

Kết quả là, nếu ta đang trên đường, chưa tới vị trí cách C một khoảng $\frac{wb}{\sqrt{v^2-w^2}}$ thì ta nên đi tiếp cho đến khi đến vị trí đó rồi ta tách đường để đi vào cát, tiến đến A. Còn nếu ta đã đi qua vị trí ấy thì ngay lập tức ta cần tách đường để đi vào cát, tiến đến A.

Sinh viên tự viết các câu lệnh Python!