Stærð- og Reiknifræði

Skilaverkefni02 | Donn Eunice | deb5@hi.is | 250794-3479

Verkefni 2B Simpson Regla

Skrifa skal forrit til að nálga heildi með svonefndri Simpsons-regla. Í trapisureglu er heildisbilinu skipt í n hlutbil, fallið sem heilda skal nálgað með beinum línustrikum og heildi þess nálgað með flatarmálinu undir þessum línustrikum. Í Simpsonsreglu er fallið hinsvegar nálgað (eða brúað eins og það er kallað) með parabólum og heildið nálgað með flatarmálinu undir þeim. Skoðið endilega Wikipedíu-grein um aðferðina.

Simpsons-formúlan er eftirfarandi:

$\int ba f(x)dx \approx \Delta x 3(f(x0)+4f(x1)+2f(x2)+4f(x3)+2f(x4)+\cdots+4f(xn-1)+f(xn))$ þar sem Δx og xi eru eins og í A-lið og n er slétt tala.

Skrifið fall simpson(f,a,b,n) sem nálgar heildið af f frá a til b með samsettri Simpsons-reglu með n hlutbilum.

Prófið með heildunum (*) og (**) úr A-lið með 4 hlutbilum. Ef rétt er forritað ætti að fást $(*)$ 0.65933 og $(**)$ 1.71832. Kannið líka hve stórt n þarf að vera til að fá alla 7 aukastafina sem gefnir eru í töflunni í A-lið rétta. Heildið a' lokum eitthvert sjálfvalið fall þar sem afmælisdagur ykkar kemur við sögu.

In [8]:
```
def simpson(f, a, b, n):
    h = (b - a) / n
    x = a + h
    result = f(a) + f(b)
    for i in range(1, n):
        result += 4 * f(x) if i % 2 == 1 else 2 * f(x)
        x += h
    return (h / 3) * result

# test á aðgerð
def f(x):
    return 3 * x**2 + 2 * x + 1

print(simpson(f, 0, 2, 4)) # ætti að skila inn 0.65933

# Test á aðgerð 2
def g(x):
    return x**3 + 2 * x**2 + 3 * x + 4

print(simpson(g, 0, 2, 4)) # should return 1.71832

#Til þess að fá 7 aukastafi rétt þarf að auka n.

# afmælið mitt er 25.juní.1994
import datetime

def afmaeli(x):
    birthdate = datetime.datetime(1994, 6, 25)
    current_date = datetime.datetime.now()
    age = (current_date - birthdate).days / 365
```

```
    return x**2 + age

print(simpson(afmaeli, 0, 2, 4))
```

```
14.0
23.333333333333332
59.86118721461186
```

1. Bóluröðun

Hér er reiknirit sem raðar n-staka lista x með bóluröðun bubble sort, sem snýst um að rúlla í gegn um stökin og ef tvö samliggjandi stök eru í öfugri röð þá er víxlað á þeim. Þetta er endurtekið þar til öll stökin eru í röð. Minnstu stökin bobbla smám saman eins og loftbólur fremst í listann.

víxlað = satt meðan víxlað víxlað = ósatt fyrir i=1,...,n-1: ef x[i-1] > x[i] þá víxla á x[i-1] og x[i] víxlað = satt

Þýðið þetta reiknirit yfir í Python-fall bóluröðun(x). Athugið að til að víxla á breytum x og y má nota (x,y) = (y,x). Prófið með því að raða listanum [3,8,1,2,5,4].

In [9]:
```
#Bubble sort programm til að sortera tölurnar (3,8,1,2,5,4) frá minnsta í st
def bubbleSort(numer):
        n = len(numer)
        # Ef numer er nú þegar á minna en næsta þá þarf ekki að víxla
        swapped = False
        for i in range(n-1):
                for j in range(0, n-i-1):

    # víxla ef númer á undan er stærra en númerið á eftir
                        if numer[j] > numer[j + 1]:
                                swapped = True
                                numer[j], numer[j + 1] = numer[j + 1], numer

                if not swapped:
    # ef ekkert er víxlað þá fara úr loop
                        return

# upphaflega röðun á númer.
numer = [3,8,1,2,5,4]

bubbleSort(numer)

print("Raðað númer:")
for i in range(len(numer)):
        print(numer[i], end=" ")
```

```
Raðað númer:
1 2 3 4 5 8
```

38B) Vaxtareikningur

1) Skrifið fall með stika u, p, k, m sem reiknar heildarvexti, v, af upphæð u sem er á p % vöxtum í k ár og m mánuði skv. formúlunni

$v=u(1+a)k(1+am12)-u$, þar sem $a=p100$

Skerið af aura (með fallinu int), hafið viðeigandi skjölunarstreng í fallinu, og prófið það með því að reikna 2% vexti af 10000 kr. í 3 ár og 4 mánuði (ætti að gefa 682 kr.).

2) Reiknið heildarvexti til dagsins í dag ef 25000 kr. hefðu verið lagðar inn á 3% vexti á fæðingardegi ykkar (nálgið aldur ykkar í heilan mánuð), og reiknið jafnframt út hlutfall vaxtanna af upphaflegri upphæð. Notið f-strengi til að skrifa niðurstöðurnar með hæfilegum skýringartexta.

3) Skrifið fall sem ákvarðar hve mörg ár og mánuði það tekur upphæð á n % vöxtum að tvöfaldast (notið t.d. tvöfalda lykkju, og return á viðeigandi stað). Prófið með n = 13 (ætti að gefa 5 ár og 8 mánuði) og með n gefnu með fæðingarmánuði ykkar (t.d. 8 fyrir ágúst)

In [10]:
```
# 1)
def compound_interest(u, p, k, m):
    a = p / 100
    v = u * (1 + a) ** (k + (m / 12)) - u
    v = int(v)
    print(f"Heildarvirði {u} með {p}% vöxtum fyrir {k} ár og {m} mánuðir eru

compound_interest(10000, 2, 3, 4)
```

```
Heildarvirði 10000 með 2% vöxtum fyrir 3 ár og 4 mánuðir eru 682 kr.
```

In [23]:
```
# 2

import datetime

faedingardagur = datetime.datetime(1994, 6, 25)
núverandi_dagur = datetime.datetime.now()
dagar_síðan_faedingardags = (núverandi_dagur - faedingardagur).days

upphæð = 25000
voxtur = 0.03

heildarvextir = upphæð * voxtur * dagar_síðan_faedingardags / 365
heildarupphæð = upphæð + heildarvextir
hlutfall_vexta = heildarvextir / upphæð * 100

print(f"Ef 25000 kr. var lagt á 3% vexti á fæðingardegi {faedingardagur.date
print(f"Heildarvextirnir eru {heildarvextir:.2f}kr og heildarupphæðin er {he
print(f"sem er {hlutfall_vexta:.2f}% aukning á upphæðinni")
```

```
Ef 25000 kr. var lagt á 3% vexti á fæðingardegi 1994-06-25 til dagsetningar
2023-01-22
Heildarvextirnir eru 21447.95kr og heildarupphæðin er 46447.95kr
sem er 85.79% aukning á upphæðinni
```