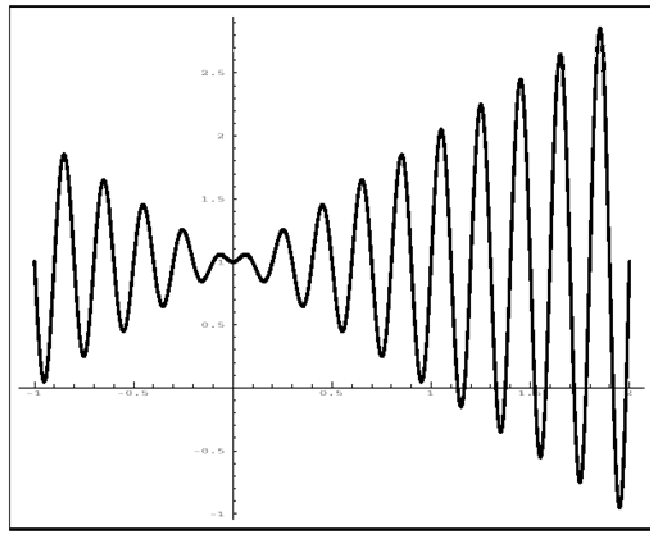


Exercici 8: Algorismes de cerca: cerca lineal.

Lliurament:

UN ÚNIC FITXER (exercici8.py) QUE CONTINGUI EL CONJUNT DE FUNCIONS QUE S'HAN IMPLEMENTAT.

- En aquest primer exercici de cerca ens concentrarem en el problema de cercar el màxim d'una funció en un determinat interval. Per exemple, considereu la funció $y = x \sin(10\pi x) + 1.0$ a l'interval $[-1.0, 2.0]$:



El problema que ens plantegem és trobar què val el màxim d'aquesta funció dins d'aquest interval i per quin valor de x el pren.

La primera aproximació que prendrem és la cerca per força bruta: generar tots els valors i determinar quin és el màxim. De totes maneres, per fer això podem usar certes funcionalitats de Python que ens faran el programa més elegant.

Una d'aquestes funcionalitats és la comanda `yield`. Aquesta comanda permet convertir una funció en un iterador. Per entendre el seu funcionament, executeu aquest programa i observeu el seu comportament quan hi entreu un *string* qualsevol. Com genera les permutacions?

```
def all_perms(str):
    if len(str) <= 1:
        yield str
    else:
        for perm in all_perms(str[1:]):
            for i in range(len(perm)+1):
                yield perm[:i] + str[0:1] + perm[i:]
for i in all_perms("abcd"): print i
```

Aquesta comanda (que està en lloc del que podrien ser els `return` de la funció) ens ha creat un iterador, que genera elements només quan els necessitem.

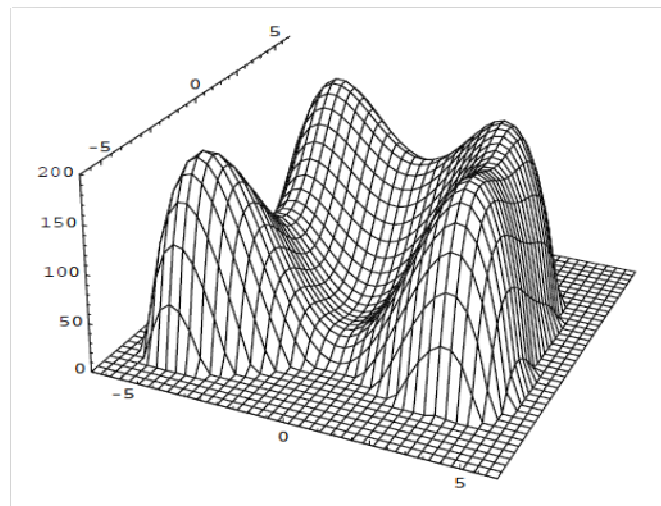
Per desenvolupar el programa aneu seguint els següents passos:

- Feu una funció **func1d(x)** que prengui el valor de x com a paràmetre i retorni el valor de la funció.
- Feu una funció **frange1d(start, end, inc)** que implementi la mateixa funcionalitat que té la instrucció `range` pels enters però pels reals. Feu que aquesta funció torni els valors amb un `yield` enlloc de retornar una llista.
- Feu una funció **search1d()** que retorni amb un `print` les informacions següents (en el mateix ordre):
 - Quin és el valor màxim de la funció a l'interval $[-1,2]$ quan mostregem els valors de x cada 0.01 unitats (`inc=0.01`), per quin valor de x es dona, quantes avaluacions de la funció heu fet i quan ha trigat.
 - Quin és el valor màxim de la funció a l'interval $[-1,2]$ quan mostregem els valors de x cada 0.0001 unitats (`inc=0.0001`), per quin valor de x es dona, quantes avaluacions de la funció heu fet i quan ha trigat.
 - Quin és el valor màxim de la funció a l'interval $[-1,2]$ quan mostregem els valors de x cada 0.000001 unitats (`inc=0.000001`), per quin valor de x es dona, quantes avaluacions de la funció heu fet i quan ha trigat.

Observació:

El valor màxim de la funció està al voltant del valor 2,85 i es dona per una x al voltant del valor 1,85.

- Reescriuiu les funcions anteriors (i creeu les funcions **func2d**, **frange2d**, **search2d**) per la funció $f(x,y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$ a l'interval $-6 \leq x,y \leq +6$.



Observacions:

Si mostregeu (x,y) amb passos molt petits això pot arribar a trigar molt. Comproveu fins on podeu mostrejar per poder donar un resultat en menys de 10 segons.

Aquesta funció té quatre màxims amb el mateix valor al voltant dels punts $(3.58, -1.86)$, $(3.0, 2.0)$, $(-2.815, 3.125)$ i $(-3.78, -3.28)$.

- Descarregueu-vos el mòdul **genetic.py** del campus virtual. Aquest programa calcula el màxim de la funció $y = x \sin(10\pi x) + 1.0$ a l'interval [-1.0, 2.0] (la solució amb 6 decimals de precisió és $f(x)=2.850273$, $x=1.850547$). Modifiqueu la funció **genetic** (sense canviar el nom) de manera que:
 - Calculi i imprimeixi el temps que triga a calcular la solució.
 - Calculi i imprimeixi el nombre d'avaluacions de la funció cost que ha fet.
- Creeu una funció (anomenada **genetic2**) que calculi una solució per la següent funció (ajuste els paràmetres de la funció **genetic** segons el vostre criteri): $f(x,y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$ a l'interval $-6 \leq x, y \leq +6$ amb sis decimals de precisió.

Notes:

- Exemples de “lists comprehensions”:

```
>>> freshfruit = [' banana', ' loganberry ', 'passion fruit ']
>>> [weapon.strip() for weapon in freshfruit]
['banana', 'loganberry', 'passion fruit']
>>> vec = [2, 4, 6]
>>> [3*x for x in vec]
[6, 12, 18]
>>> [3*x for x in vec if x > 3]
[12, 18]
>>> [3*x for x in vec if x < 2]
[]
>>> [[x,x**2] for x in vec]
[[2, 4], [4, 16], [6, 36]]
>>> vec1 = [2, 4, 6]
>>> vec2 = [4, 3, -9]
>>> [x*y for x in vec1 for y in vec2]
[8, 6, -18, 16, 12, -36, 24, 18, -54]
>>> [x+y for x in vec1 for y in vec2]
[6, 5, -7, 8, 7, -5, 10, 9, -3]
>>> [vec1[i]*vec2[i] for i in range(len(vec1))]
[8, 12, -54]
```

- Consulteu el pas de paràmetres a una funció en Python:
<http://docs.python.org/tutorial/controlflow.html#more-on-defining-functions>