

Projects Welcome to CoCalc SageLab

**⚠ Free Trial (Day 0) - buy a license (starting at about \$3/month) and apply it to this project.**  
Otherwise, expect VERY bad performance (e.g., 10 times slower!) and you can't install packages, clone from GitHub, or download datasets. - [more info...](#)

Files New Log Find Info Settings SymPyLab\_with\_Sage\_Worksheet.x

File Run Stop in out A A Jupyter Save TimeTravel

Help Modes # Data Control Program Plots Calculus Linear Graphs Numbers Rings Snippets %sage

1 %md  
2 #  
3 # Laboratorio 5 SymPy  
4 Grupo 31  
5 \* Javier Alejandro Ortiz Silva.  
6 \* Gustavo Alberto Puentes Romero  
7 \* Cristian Daniel Ramirez Marin  
8  
9  
10  
11 #  
12

## Laboratorio 5 SymPy

Grupo 31

- Javier Alejandro Ortiz Silva.
- Gustavo Alberto Puentes Romero
- Cristian Daniel Ramirez Marin

---

13 %md  
14 #  
15 # Solucion Laboratorio 5 punto 2  
16 2. Create a Sage Math notebook in cocalc to replicate 1.  
17  
18

## Solucion Laboratorio 5 punto 2

2.Create a Sage Math notebook in CoCalc to replicate 1.

20 %md  
21 # Parte 1: Polinomios en SymPy  
22 Ingresamos el polinomio  $(x-1)(x-2)(x-3)(x-4)(x-5)(x-6)(x-7)(x-8)(x-9)(x-10)$  en WolframAlpha  
23   
24  
25

## Parte 1: Polinomios en SymPy

Ingresamos el polinomio  $(x-1)(x-2)(x-3)(x-4)(x-5)(x-6)(x-7)(x-8)(x-9)(x-10)$  en WolframAlpha

(x-1)(x-2)(x-3)(x-4)(x-5)(x-6)(x-7)(x-8)(x-9)(x-10)

Input:  $(x - 1)(x - 2)(x - 3)(x - 4)(x - 5)(x - 6)(x - 7)(x - 8)(x - 9)(x - 10)$

Plots:

Alternate form:  

$$(x^2 - 11x)^5 + 110(x^2 - 11x)^4 + 4708(x^2 - 11x)^3 + 97416(x^2 - 11x)^2 + 966240(x^2 - 11x) + 3628800$$

Expanded form:  Step-by-step solution  

$$x^{10} - 55x^9 + 1320x^8 - 18150x^7 + 157773x^6 - 902055x^5 + 3416930x^4 - 8409500x^3 + 12753576x^2 - 10628640x + 3628800$$

```

34 %
35 %
36 1 %md
37 2 Usando la funcion factor ingresamos la forma extendida del polinomio para que nos genere la forma completamente factorizada
38 %
39 %
40 1 %python3
41 2 from sympy import *
42 3
43 4 x = Symbol('x')
44 5 factor(x**10 - 55*x**9 + 1320*x**8 - 18150*x**7 + 157773*x**6 - 902055*x**5 + 3416930*x**4 - 8409500*x**3 + 12753576*x**2 - 10628640*x + 3628800 )
45 %
46 %
47 1 %md
48 2 Ahora usaremos la division de polinomios mediante la funcion div. Dividiremos el polinomio en su forma expandida por cada uno de sus terminos, excepto el ultimo, de la forma factorizada. El resultado sera el termino faltante ( $x-10$ ). Tambien notese que con cada division el polinomio ve su grado reducido en 1.
49 %
50 %
51 1 %python3
52 2 from sympy import Symbol
53 3 from sympy import div
54 4
55 5 x = Symbol('x')
56 6 print("-----")
57 7 print("Division sintetica")
58 8 print("-----\n\n\n")
59 %
60 10
61 11
62 12 print("Polinomio en forma expandida grado 10:")
63 13 print("      x**10 - 55*x**9 + 1320*x**8 - 18150*x**7 + 157773*x**6 - 902055*x**5 + 3416930*x**4 - 8409500*x**3 + 12753576*x**2 - 10628640*x + 3628800\n\n\n")
64 14
65 15
66 16 print("division por (x - 1):")
67 17 p = x**10 - 55*x**9 + 1320*x**8 - 18150*x**7 + 157773*x**6 - 902055*x**5 + 3416930*x**4 - 8409500*x**3 + 12753576*x**2 - 10628640*x + 3628800_
68 18
69 19 p, r = div(p, x-1)
70 20
71 21 print(p)
72 22 print(r)
73 %
74 24
75 25 print("\n\n\ndivision por (x - 2):")
76 26 p, r = div(p, x-2)
77 27
78 28 print(p)
79 29 print(r)
80 %
81 31
82 32 print("\n\n\ndivision por (x - 3):")
83 33 p, r = div(p, x-3)
84 34
85 35 print(p)
86 36 print(r)
87 %
88 38
89 39 print("\n\n\ndivision por (x - 4):")
90 40 p, r = div(p, x-4)
91 41
92 42 print(p)
93 43 print(r)
94 %
95 45
96 46 print("\n\n\ndivision por (x - 5):")
97 47 p, r = div(p, x-5)
98 48
99 49 print(p)
100 50 print(r)
101 %
102 51
103 52 print("\n\n\ndivision por (x - 6):")
104 53 p, r = div(p, x-6)
105 54
106 56 print(p)
107 57 print(r)
108 %
109 59
110 60 print("\n\n\ndivision por (x - 7):")
111 61 p, r = div(p, x-7)
112 62
113 63 print(p)
114 64 print(r)
115 %
116 66
117 67
118 68 print("\n\n\ndivision por (x - 8):")
119 69 p, r = div(p, x-8)
120 70
121 71 print(p)
122 72 print(r)
123 %
124 74
125 75 print("\n\n\ndivision por (x - 9):")
126 76 p, r = div(p, x-9)
127 %
128 78 print(p)
129 79 print(r)
130 %
-----
```

### Division sintetica

Polinomio en forma expandida grado 10:

$$x^{10} - 55x^9 + 1320x^8 - 18150x^7 + 157773x^6 - 902055x^5 + 3416930x^4 - 8409500x^3 + 12753576x^2 - 10628640x + 3628800$$

division por (x - 1):

$$x^9 - 54x^8 + 1266x^7 - 16884x^6 + 140889x^5 - 761166x^4 + 2655764x^3 - 5753736x^2 + 6999840x - 3628800$$

division por (x - 2):

$$x^8 - 52x^7 + 1162x^6 - 14560x^5 + 111769x^4 - 537628x^3 + 1580508x^2 - 2592720x + 1814400$$

0

```
division por (x - 3):  
x**7 - 49*x**6 + 1015*x**5 - 11515*x**4 + 77224*x**3 - 305956*x**2 + 662640*x - 604800  
0
```

```
division por (x - 4):  
x**6 - 45*x**5 + 835*x**4 - 8175*x**3 + 44524*x**2 - 127860*x + 151200  
0
```

```
division por (x - 5):  
x**5 - 40*x**4 + 635*x**3 - 5000*x**2 + 19524*x - 30240  
0
```

```
division por (x - 6):  
x**4 - 34*x**3 + 431*x**2 - 2414*x + 5040  
0
```

```
division por (x - 8):  
x**2 - 19*x + 90  
0
```

```
division por (x - 9):  
x - 10  
0
```

```
131  
132 1  
133 2  
134 3  
135 4  
136 5  
#  
# Parte 2: Integrales en SymPy  
137 6  
## A. Integrales simples en una variable  
138 7  
139 8  
140 9  
141 10  
142
```

## Parte 2: Integrales en SymPy

### A. Integrales simples en una variable

```
143  
144 1  
145 2  
146 3  
## Ejemplo 1  
147 4  
148 5 Integral de una funcion producto de polinomio, seno y exponencial.  
149 6  
150 7 $$\int_{-5}^5 x^2 \sin(x) e^x dx$$
```

#### Ejemplo 1

Integral de una función producto de polinomio, seno y exponencial.

$$\int_{-5}^5 x^2 \sin(x) e^x dx$$

```
152  
153 1  
154 2   
155
```



integrate\_{-5}^5 f(x)= x^2\*sin(x)\*e^x

Extended Keyboard    Upload

Examples    Random

Definite integral:

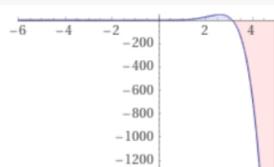
More digits    Step-by-step solution

$$\int_{-5}^5 x^2 \sin(x) e^x dx = 24 \sin(5) \cosh(5) + 2 \cos(5) (5 \cosh(5) - 13 \sinh(5)) \approx -2044.6$$

cosh(x) is the hyperbolic cosine function

sinh(x) is the hyperbolic sine function

Visual representation of the integral:



Indefinite integral:

 Step-by-step solution

$$\int x^2 \sin(x) e^x dx = \frac{1}{2} e^x ((x^2 - 1) \sin(x) - (x - 1)^2 \cos(x)) + \text{constant}$$

[Download Page](#)

POWERED BY THE WOLFRAM LANGUAGE

```

156 %
157 1 %md
158 2 ## Solucion analitica usando el teorema fundamental del calculo
159 3 Solucion analitica usando el teorema fundamental del calculo

160 %
161 1 %python3
162 2 import sympy
163 3 # we'll save results using different methods in this data structure, called a dictionary
164 4 result = {}
165 5 x = sympy.Symbol("x")
166 6 i = sympy.integrate(x**2*sympy.sin(x) * sympy.exp(x))
167 7 print("Antiderivada o integral indefinida\n",i,"\\n\\n")
168 8
169 9 print("Solucion analitica F(b) - F(a) donde F es la antiderivada, b limite superior y a limite inferior")
170 10
171 11 result["analytical"] = float(i.subs(x, 5) - i.subs(x, -5))
172 12
173 13 print(" ",result["analytical"])
174 Antiderivada o integral indefinida
      x**2*exp(x)*sin(x)/2 - x**2*exp(x)*cos(x)/2 + x*exp(x)*cos(x) - exp(x)*sin(x)/2 - exp(x)*cos(x)/2

      Solucion analitica F(b) - F(a) donde F es la antiderivada, b limite superior y a limite inferior
      -2044.6405105725285

```

```

175 %
176 1 %md
177 2 ## Estimacion por metodo de montecarlo
178 3 Estimacion por metodo de montecarlo

```

```

179 %
180 1 %python3
181 2 import numpy
182 3 N = 100_000
183 4 accum = 0
184 5 for i in range(N):
185 6     x = numpy.random.uniform(-5, 5)
186 7     accum += x**2*sympy.sin(x) * sympy.exp(x)
187 8 volume = 5 - (-5)
188 9 result["MC"] = volume * accum / float(N)
189 10 print("Estimacion por metodo estandar de MonteCarlo: {}".format(result["MC"]))
190 Estimacion por metodo estandar de MonteCarlo: -2023.70542253138

```

```

191 %
192 1 %md
193 2 ## Ejemplo 2
194 3 Integral de una funcion producto de polinomio y logaritmo.
195 4
196 5 $\$\\int_{[1]}^5-x^4\log_2([x])\\,dx\\$$_
197 6

```

## Ejemplo 2

Integral de una funcion producto de polinomio y logaritmo.

$$\int_1^5 -x^4 \log_2(x) dx$$

```

199 %
200 1 %md
201 2 
202 3
203

```









Definite integral:

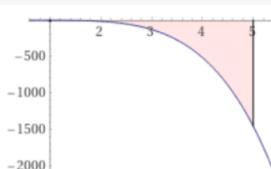
 Step-by-step solution

$$\int_1^5 -x^4 \log_2(x) dx = -\frac{625 \log(5) - \frac{3125}{25}}{\log(2)} \approx -1270.93$$

$\log_b(x)$  is the base-  $b$  logarithm

$\log(x)$  is the natural logarithm

Visual representation of the integral:



Indefinite integral:

 Step-by-step solution

$$\int -x^4 \log_2(x) dx = -\frac{\frac{1}{5} x^5 \log(x) - \frac{x^5}{25}}{\log(2)} + \text{constant}$$

(assuming a complex-valued logarithm)

[Download Page](#)

POWERED BY THE WOLFRAM LANGUAGE

```
204
205 1
206
207
208 1
209 2 %%md
## Solucion analitica usando el teorema fundamental del calculo
210 Solucion analitica usando el teorema fundamental del calculo
```

```
211
212 1
213 2 %%python3
214 3 import sympy
215 4 # we'll save results using different methods in this data structure, called a dictionary
216 5 result = {}
217 6 x = sympy.Symbol("x")
218 7 i = sympy.integrate(-(x**4)*sympy.log(x,2))
219 8
220 9 print("Antiderivada o integral indefinida\n      ",i,"\\n\\n\\n")
221 10
222 11 print("Solucion analitica F(b) - F(a) donde F es la antiderivada, b limite superior y a limite inferior")
223 12
224 13 result["analytical"] = float(i.subs(x, 5) - i.subs(x, 1))
225 14
226 15 print("      ",result["analytical"])
227 16
228 17
229 18
230 Antiderivada o integral indefinida
-x**5*log(x)/(5*log(2)) + x**5/(25*log(2))

Solucion analitica F(b) - F(a) donde F es la antiderivada, b limite superior y a limite inferior
-1270.9258869951166
```

```
231
232 1 %%md
## Estimacion por metodo de montecarlo
233 2
234 Estimacion por metodo de montecarlo
```

```
235
236 1 %%python3
237 2 import numpy
238 3 N = 100_000
239 4 accum = 0
240 5 for i in range(N):
241 6     x = numpy.random.uniform(1, 5)
242 7     accum += -(x**4)*float(sympy.log(x,2))
243 8 volume = 5 - 1
244 9 result["MC"] = volume * accum / float(N)
245 10 print("Estimacion por metodo estandar de Montecarlo: {}".format(result["MC"]))
246 Estimacion por metodo estandar de MonteCarlo: -1270.154609252245
```

```
247
248 1 %%md
## B. Integrales triples para funciones de 3 variables
249 2
250 3 B. Integrales triples para funciones de 3 variables
251
```

```
252
253 1 %%md
254 2
255 3 ## Ejemplo 1
256 4 Integral triple de una funcion producto de seno, polinomio, exponencial y logaritmo natural
257 5 $$\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \sin(x)y^3e^z\ln(y) dx dy dz$$
258
```

### Ejemplo 1

Integral triple de una funcion producto de seno, polinomio, exponencial y logaritmo natural

$$\int_0^2 \int_0^2 \int_0^2 \sin(x)y^3 e^z \ln(y) dx dy dz$$

```
259
260 1 %%md
261 2 
262
```



$\int_0^2 \int_0^2 \int_0^2 \sin(x)y^3 e^z \ln(y) dx dy dz$

$\int$  Extended Keyboard

$\uparrow$  Upload

$\oplus$  Exam

Definite integral:

$$\int_0^2 \int_0^2 \int_0^2 \sin(x) y^3 e^z \log(y) dx dy dz = 2(e^2 - 1)(\log(16) - 1) \sin^2(1) \approx 16.0381$$

log(x) is

[Download Page](#)

POWERED BY TH

```
263 1
264 2
265 3
266 4
267 5 %md
## Solucion analitica usando el teorema fundamental del calculo
268 6
269 7 Solucion analitica usando el teorema fundamental del calculo
270 8
271 9
272 10
273 11
274 12
275 13
276 14
277 15
278 16
279 17
280 18
281 19
282 20
283 21
284 22
285 23
286 24
287 25
288 26
289 27
290 28
291 29
292 30
293 31
294 32
295 33
296 34
297 35
298 36
299 37
300 38
301 39
302 40
303 41
304 42
305 43
306 44
307 45
308 46
309 47
310 48
311 49
312 50
313 51
314 52
315 53
316 54
317 55
318 56
319 57
320 58
321 59
322 60
323 61
```

## Solucion analitica usando el teorema fundamental del calculo

Solucion analitica: 16.038101950678463

```
%md
## Estimacion por metodo de montecarlo
```

## Estimacion por metodo de montecarlo

```
%python3
N = 1_000_000
accum = 0
for i in range(N):
    x = numpy.random.uniform(0,2)
    y = numpy.random.uniform(0,2)
    z = numpy.random.uniform(0,2)
    accum += numpy.sin(x)*y**3*numpy.exp(z)*numpy.log(y)
volume = 8
result = {}
result["MC"] = volume * accum / float(N)
print("Estimacion por metodo estandar de MonteCarlo: {}".format(result["MC"]))
```

Estimacion por metodo estandar de MonteCarlo: 16.101680519459443

```
%md
## Ejemplo 2
```

Integral triple de una funcion producto de polinomio, seno cuadrado, exponencial negativo y logaritmo base dos  
\$\int\_{-3}^2 \int\_{-1}^4 \int\_0^5 x^2 \sin^2(y) e^{-z} \log\_2(x) dx dy dz\$

## Ejemplo 2

Integral triple de una funcion producto de polinomio, seno cuadrado, exponencial negativo y logaritmo base dos

$$\int_{-3}^2 \int_{-1}^4 \int_0^5 x^2 \sin^2(y) e^{-z} \log_2(x) dx dy dz$$


$\int_{-3}^2 \int_{-1}^4 \int_0^5 x^2 \sin^2(y) e^{-z} \log_2(x) dx dy dz$

=

Extended Keyboard

Upload

Examples

Random

Definite integral:

More digits

$$\int_{-3}^2 \int_{-1}^4 \int_0^5 x^2 \sin^2(y) e^{-z} \log_2(x) dx dy dz =$$

$$-\frac{125(e^5 - 1)(\log(125) - 1)(-10 + \sin(2) + \sin(8))}{36 e^2 \log(2)} \approx 3099.52$$

$\log_b(x)$  is the base-  $b$  logarithm

$\log(x)$  is the natural logarithm

[Download Page](#)

POWERED BY THE WOLFRAM LANGUAGE

324

325 1

326 2

327 3

328 4

329 5

%md  
## Solucion analitica usando el teorema fundamental del calculo

## Solucion analitica usando el teorema fundamental del calculo

```
330 v
331 1 %python3
332 2 import sympy
333 3 sympy.init_printing()
334 4
335 5 x = sympy.Symbol("x")
336 6 y = sympy.Symbol("y")
337 7 z = sympy.Symbol("z")
338 8 expr = (x**2)*((sympy.sin(y)**2)*sympy.exp(-z)*sympy.log(x,2))
339 9 res = sympy.integrate(expr,
340 10     (x, 0,5),
341 11     (y, -1,4),
342 12     (z, -3,2))
343 13 # Note: we use float(res) to convert res from symbolic form to floating point form
344 14 result = []
345 15 result["analytical"] = float(res)
346 16 print("solucion analitica: {}".format(result["analytical"]))
347
348 v Solucion analitica: 3099.5165072370155
```

349

350 1

351 2

%md  
## Estimacion por metodo de montecarlo

## Estimacion por metodo de montecarlo

```
352 v
353 1 %python3
354 2
355 3 N = 10_000_000
356 4 accum = 0
357 5 for i in range(N):
358 6     x = numpy.random.uniform(0,5)
359 7     y = numpy.random.uniform(-1,4)
360 8     z = numpy.random.uniform(-3,2)
361 9     accum += (x**2)*((numpy.sin(y)**2)*numpy.exp(-z)*numpy.math.log2(x))
362 10 volume = 125
363 11 result = []
364 12 result["MC"] = volume * accum / float(N)
365 13 print("Estimacion por metodo estandar de MonteCarlo: {}".format(result["MC"]))
366
367 v Estimacion por metodo estandar de MonteCarlo: 3101.448008621367
```

368

369 1

370 2

371 3

372 4

373 5

374 6

375 7

376 8

377 9

378 10