CSUMS: A Sage Potpourii

last edited on April 17, 2012 02:23 AM by admin

Save Save & quit Discard & quit

File... Action. Data... sage Typeset Print Worksheet Edit Text Undo Share Publish

CSUMS: A Sage Potpourii

April 17, 2012

Prof. Adam O. Hausknecht

1. Basic Examples:

A. Algebra Examples

```
factorial(100)
```

```
digits = floor(float(log(factorial(100)))/float(log(10)))+1
digits
```

```
var('x, y') ## Declare the variables to be symbolic
ex1 = expand((x-y)^10)
ex1
```

```
factor(ex1)
```

```
simplify((x^2-100)/(x-10))
```

```
solve(x^3-1 == 0,x)
```

```
solve(y==(2*x-1)/(3*x+1),x)
```

```
f = (x^3-8)/(x^3-2*x^2-8*x)
```

```
f.partial_fraction(x)
```

B. Calculus Examples

```
intF = f.integrate(x)
intF
```

```
intF.diff(x)
```

```
f.integrate(x, 5, 10)
```

```
f.nintegrate(x, 5, 10)
```

B. 2-D Plotting

Exampe 1: A simple $y = f(x)$ plot

```
plot(x*sin(1/x), (x, -1,1))
```

Example 2: A fancy $y = f(x)$ plot.

```
var('x')
f=x*sin(1/x)
p = plot(f, (x, -1,1), legend_label="y = x sin(1/x)")
p = p+plot(x, (x,-1,1), rgbcolor=(1,0,0), legend_label="y = x")
p = p+plot(-x, (x,-1,1), rgbcolor=(0,1,0), legend_label="y = -x")
p.show()
p.legend()
```

Example 3: A two-color 3D plot of $z = f(x, y)$.

```
var('x, y')
f = (x^3-y^3)*exp(-x^2-y^2)
## plot the function using "adaptive  = True" so that two colors can be used.
plot3d(f,(x,-3,3),(y,-3,3), adaptive = True, color=['gray','white'])
```

Example 4: A 3D plot with better shading

```
s3d =plot3d(f,(x, -3,3),(y, -3,3), adaptive=True, color=rainbow(60, 'rgbtuple'),
max_bend=.1, max_depth=15)
s3d.show()
```

Example 5: Rendering a surface with a 3D ray tracer.

```
s3dRx = s3d.rotateX(pi/2)
s3dRz = s3d.rotateZ(pi/3)
s3dRz.show(viewer='tachyon')
```

Example 6: A ray rracer Plot of spheres on a Helix.

```
t = Tachyon(xres=512,yres=512, camera_center=(3,0.3,0))
t.light((4,3,2), 0.2, (1,1,1))
t.texture('t0', ambient=0.1, diffuse=0.9, specular=0.5, opacity=1.0, color=(1.0,0,0))
t.texture('t1', ambient=0.1, diffuse=0.9, specular=0.3, opacity=1.0, color=(0,1.0,0))
t.texture('t2', ambient=0.2,diffuse=0.7, specular=0.5, opacity=0.7, color=(0,0,1.0))
k=0
for i in srange(-1,1,0.05):
     k += 1
     t.sphere((cos(2*pi*i),sin(2*pi*i),i), 0.1, 't%s'%(k%3))
t.show()
```

Example 7: A 2D countour plot.

```
contour_plot(f, (x, -3, 3), (y, -3, 3), fill=True, plot_points=150)
```

Example 8: An interactive 2 D plot.

```
@interact
def foo(q1= slider([-3..3,step=.5], default=-1, label = 'A'), q2=slider([-
3..3,step=.5], default=2,label='B')):
    x,y = var('x,y')
    f = (q1*x^2+q2*y^2)*exp(-x^2-y^2)
    C = contour_plot(f, (x, -3,3), (y, -3,3), plot_points=30, contours=15,
cmap='cool')
    show(C, figsize=3, aspect_ratio=1)
    show(plot3d(f, (x,-3,3), (y,-3,3)), figsize=4)
```

Example 9: A MATLAB like $y = f(x)$ saved in a .PNG image file.

```
from pylab import *
t = arange(0.0, 2.0, 0.01)
s = sin(2*pi*t)
P = plot(t, s, linewidth=1.0)
xl = xlabel('$t = time$')
yl = ylabel('$y = sin(t)$')
t = title('Goodbye MATLAB!')
grid(True)
savefig('/Users/ahausknecht/Desktop/sinePlot.png')
```

Exmple 10: A MATLAB like contour plot of $z = f(x, y)$.

```
from pylab import *
clf()
x = arange(-5, 5, .1)
y = arange(-5, 5, .1)
xx, yy = meshgrid(x, y)
zz = xx*cos(yy)-yy*cos(xx)
zMin = zz.min()
zMax = zz.max()
c = arange(zMin, zMax, (zMax-zMin)/10)
hold(True)
contourf(xx, yy, zz, levels=c)
contour(xx, yy, zz, linewidths = 3, levels = c, colors='black')
title("$z = x \cos(y)-y \cos(x)$")
xlabel('x')
ylabel('y')

savefig('/Users/ahausknecht/Desktop/contourPlot.png')
```

2. Fourier Series

A. Calculating Fourier Series Directly

Step 1: Four periodic functions.

```
var('t')

squareWave = (sgn(sin(pi*t))+1)/2
pSW = plot(squareWave, (t,-2,2), legend_label="Square Wave", thickness=2)

sawToothWave = (t-1)- floor(t-1)
pST = plot(sawToothWave, (t, -2,2), legend_label = "Saw Tooth", color='red',
thickness=2 )

treeWave = 2.0*abs((t -0.5)- floor(t-0.5)- 0.5)
pTW = plot(treeWave, (t,-2,2), legend_label = "Trees Wave", color='green',
thickness=2)

halfSineWave = 0.5*(abs(sin(pi*t-pi))-sin(pi*t-pi))
pRW = plot(halfSineWave, (t,-2,2), legend_label = "Half-Sine Wave", color='cyan',
thickness=2)


pSW
```

Example 2: Functions for computing the partial Fourier series sums for a function $f(t)$ defined on $[-L, L]$ and periodically extended to the real line.

$$a_0 = \frac{1}{L}\int_{-L}^{L} f(t)dt,$$

$$a_n = \frac{1}{L}\int_{-L}^{L} f(t)\cos\left(\frac{n\pi t}{L}\right)dt,$$

$$b_n = \frac{1}{L}\int_{-L}^{L} f(t)\sin\left(\frac{n\pi t}{L}\right)dt$$

$$f(t) \sim \frac{a_0}{2} + \sum_{n=1}^{n=N}\left(a_n\cos\left(\frac{n\pi t}{L}\right) + b_n\sin\left(\frac{n\pi t}{L}\right)\right)$$

```
import math

def findAn(f, lastN, L = math.pi):
    """ A function that calculates the Fourier series cosine
        coefficients for f(t) with fundamental domain [-L, L]."""
    r = f.nintegral(t, -L, L)
    aN = [r[0]/(2*L)]
    for n in range(1, lastN+1):
        fCosN = f*cos(n*math.pi/L*t)
        r = fCosN.nintegral(t, -L, L)
        aN.append(r[0]/L)
    return aN


def findBn(f, lastN, L = math.pi):
    """ A function that calculates the Fourier series sine
        coefficients for f(t) with fundamental domain [-L, L]."""
    bN = [0]
    for n in range(1, lastN+1):
        fSinN = f*sin(n*math.pi/L*t)
        r = fSinN.nintegral(t, -L, L)
        bN.append(r[0]/L)
    return bN


def findFourierSeriesSum(f, lastN, L = math.pi):
    """ A generates a partial sum of the Fourier series for f(t)
        with fundamental domain [-L, L]."""
    aN = findAn(f, lastN, L)
    bN = findBn(f, lastN, L)
    s = 0
    for n, coef in enumerate(aN):
        s += coef*cos(n*math.pi/L*t)
    for n, coef in enumerate(bN):
        s += coef*sin(n*math.pi/L*t)
    return s


def doFourierPlot(f, n = 8):
    """A function that plots the periodic function f(t) and the n-th
      partial sum of the Fourier series along with a plot of the errors."""
    s = findFourierSeriesSum(f, n, 1)
    p1 = plot(s, -2, 2, rgbcolor=(0,1,0), thickness = 2)
    p2 = plot(f, -2, 2, rgbcolor=(1,0,1), thickness = 3)
    p3 = p2+p1
    p3.show()
    show(plot(s-f, -2, 2, rgbcolor=(0,0,0)))
    return p3


pF = doFourierPlot(squareWave)
```

B. Calculating Fourier Series Using Sage's Piecewise Command

Step 1: Setup the function ring and create two constant functions.

```
R.<t> = QQ[] ## The ring of functions
f0 = lambda t:0 ## The constant function 0
f1 = lambda t:1 ## The constant function 1
```

Step 2: Create a list of constant functions together with their domains.

```
functionList = []
for i in range(-2, 1, 2):
    functionList.append([(i, i+1), f1])
    functionList.append([(i+1, i+2), f0])

for f in functionList:
    print f
```

Step 3: Create the piecewise function object

```
fp = Piecewise(functionList)
fp
```

Step 4: Create plots of the periodic function and a partial sum of the Fourier series.

```
p = fp.plot(rgbcolor = (0,1,1), thickness=3)
pF2 = p+fp.plot_fourier_series_partial_sum(16, 2, -2,2, color='red')
pF2.show()
```

```
p = pF+pF2
p.show()
```

3. Animation

Example 1: Creating an animated plot of the sine function.

```
from pylab import *
t = arange(0.0, 2.0, 0.01)
for k in range(50):
    clf()
    s = k/10*sin(2*pi*t)
    P = plot(t, s, linewidth=2.0)
    axis([0, 2, -5, 5])

    if k < 10:
       savefig('/Users/ahausknecht/Desktop/csumsImages/sinePlot0' +str(k)+'.png')
    else:
       savefig('/Users/ahausknecht/Desktop/csumsImages/sinePlot' +str(k)+'.png')
```