Lecture 12: Integrating iterated integration

Aim of today

So far, this course has really focussed on developing coding literacy through basic mathematical concepts.

However, that's not all this course is! This is a *mathematical* computing course, so you need to learn both mathematical and computing techniques.

I want to acknowledge this because, this is a very broad church. As well as both advanced and normal streams, different students will be more or less familiar with the mathematical content, or the programming content (or neither!).

This lecture is going to bring back the emphasis to mathematical techniques. To give it all away, today we will:

- Define the error of an integration scheme
- Try investigating it with **integration by parts**
- Explore where that approach takes us (it's mathy and great, and illustrates how people actually find things out)
- Using our `Rational` and `Polynomial` classes to help us

Then, tomorrow we'll continue on this path and really answer why different integration schemes have different accuracy in different situations.

Recap on L10 and L11

Approximate functions

We are interested in computing the following integral:

$$I = \int_a^b f(x)\,\mathrm{d}x$$

For either the trapezoidal rule, or Simpsons rule, we approximate $f(x)$ by either a linear function (more properly called an '*affine*' function), $L(x)$, or a quadratic function, $Q(x)$. For the linear approximation L, we require

$$L(a) = f(a), \quad L(b) = f(b)$$

and for the quadratic approximation Q we require

$$Q(a) = f(a), \quad Q(b) = f(b), \quad Q(m) = f(m)$$

Either way, they are the same at the end points, a, b.

Approximate integrals

The hope with these approximations is that they are close to the actual function, so that, *in some sense*

$$f(x) \approx L(x), \quad \text{or} \quad f(x) \approx Q(x),$$

If that works, then it makes sense to expect

for the trapezoidal rule:

$$\int_a^b f(x)\,\mathrm{d}x \;\approx\; \int_a^b L(x)\,\mathrm{d}x \;=\; h\,\frac{f(a)+f(b)}{2}$$

or

for Simpsons rule:

$$\int_a^b f(x)\,\mathrm{d}x \;\approx\; \int_a^b Q(x)\,\mathrm{d}x \;=\; \frac{h}{6}\,(\,f(a)\; + 4f(m)\; +\; f(b)\,)$$

where

$$h \;\equiv\; b-a, \quad m \;\equiv\; \frac{b+a}{2}.$$

Important aside: Generalising

Note that we could use a cubic, of quartic, or any other degree we like. But, as you can see, every added degree requires an additional condition to calculate all the coefficients.

How do we choose these conditions?

What we are doing above is **interpolation**: forcing the approximation to agree with the function at certain locations. Hopefully, if the approximation agrees exactly at some points, then it won't get too bad between these points.

But there are **other** very sensible choices for approximating a function:

- maybe we want to minimise the *integral* of the difference.
- maybe the integral of the *square* of the difference...
- Heck, we might not even want to approximate the function with polynomials. Maybe there are different functions we like.

There are lots of ways to approximate a function...

Let's look at defining the error in a concrete sense. That way we can work with it.

Measuring the error

Error function e

(Not *that* error function).

In either case, we want to estimate how big the difference is between the true integral, and the approximation.

A sensible definition for the "error" function e is:

$$e(x) \;=\; f(x) - L(x), \quad \text{or} \quad e(x) \;=\; f(x) - Q(x)$$

Again, by definition $e(a) = e(b) = 0$.

How can we use e to quantify the error of of the integral of the function?

Error integral $\mathcal{E}$

We can compute the error between the integrals $\mathcal{E}$ by computing the integral of the error e between the functions (because integration is linear -- it splits over addition). Therefor we consider

$$\mathcal{E} \;\equiv\; \int_0^h e(x)\,\mathrm{d}x$$

From now on we will assume $a = 0$, and $b = h$; this is without loss of generality. Now what do we do?

Integration by parts

To analyse this, we want to use **integration by parts.** One of the top-three most useful techniques in the mathematical arsenal.

For any two functions $u(x)$, and $v(x)$,

$$\int_a^b u(x)\,v'(x)\,\mathrm{d}x \;=\; u(b)\,v(b) - u(a)\,v(a) - \int_a^b u'(x)\,v(x)\,\mathrm{d}x$$

The issue is that the integral $\mathcal{E}$ only has one function, not two, and no functions with a derivative on acting on them.

Forcing a fit

But we can fix this. We do this by making a mathematical Procrustean bed (https://en.wikipedia.org/wiki/Procrustes).

Ten years from now, if you only remember one thing about this lecture, it should be this. The idea is we don't have something that will fit with integration by parts. So we *make* something that will fit with integration by parts.

Adding a second function

Make the alteration:

$$\mathcal{E} \;\equiv\; \int_0^h e(x)\, B_0(x/h)\, \mathrm{d}x$$

Where

$$B_0(x) \;=\; 1$$

The reason for using the letter B will become apparent later. For now, we just inserted a function into the integral. It's starting to look more ready for integration by parts. But we still don't have a derivative. Let's make one.

Defining a derivative

Define

$$\frac{d}{dx} B_1(x) \;=\; B_0(x)$$

We know that $B_1(x) = x + b_1$, for some constant B_1. From now on, we'll have *functions* $B_n(x)$, and *numbers*

$$b_n \;\equiv\; B_n(0)$$

Now things are looking a lot better for integration by parts

$$\mathcal{E} \;\equiv\; h \int_0^h e(x)\, \frac{d}{dx} B_1(x/h)\, \mathrm{d}x$$

Integrate by parts

Let's integrate by parts. Using the formula,

$$\mathcal{E} \;\equiv\; h\, \left[e(h)\,(1 + b_1) - e(0)\, b_1 \right] - h \int_0^h e'(x)\, B_1(x/h)\, \mathrm{d}x$$

But recall that $e(h) = e(0) = 0$ *by design*. Therefore

$$\mathcal{E} \;=\; -h \int_0^h e'(x)\, B_1(x/h)\, \mathrm{d}x.$$

Choosing b_1

This is for $B_1(x) = x + b_1$ with b_1 not known. How do we pick b_1? There are a lot of ways we could choose to pick b_1. No matter what we do, we want to make it so that $B_1(x)$ is a small as possible. Otherwise, we would be making the estimate of $\mathcal{E}$ larger than it needs to be.

Three equivalent options are

$$|B_1(1)| = |B_0(0)|, \qquad \int_0^1 B_1(x)\,\mathrm{d}x \;=\; 0, \qquad \text{or} \qquad \int_0^1 B_1(x)^2\,\mathrm{d}x \;\to\; \text{minimum}$$

Therefore we get

$$b_1 \;\equiv\; -\frac{1}{2}.$$

Let's keep going...

Integrating again

Like before,

$$B_1(x) \;=\; \frac{1}{2}\frac{d}{dx}\left(x^2 - x + b_2\right) \;=\; \frac{1}{2}\frac{d}{dx}B_2(x)$$

In the same way as b_1 we solve and find

$$b_2 \;=\; \frac{1}{6}, \quad \text{and} \quad B_2 \;=\; x^2 - x + \frac{1}{6}$$

Therefore

$$\mathcal{E} \;=\; -\frac{h^2}{2}\int_0^h e'(x)\,\frac{d}{dx}B_2(x/h)\,\mathrm{d}x.$$

Using integration by parts and going to another,

$$\mathcal{E} \;=\; -\frac{h^2 b_2}{2}\left(e'(h) - e'(0)\right) \;+\; \frac{h^3}{3!}\int_0^h e''(x)\,\frac{d}{dx}B_3(x/h)\,\mathrm{d}x$$

Where

$$b_3 \;=\; 0, \quad \text{and} \quad B_3(x) \;=\; x^3 - \frac{3x^2}{2} + \frac{x}{2}$$

Because $b_3 \;=\; 0$, we have $B_3(0) \;=\; B_3(h/h) \;=\; 0$, so we can automatically go to another order in the integration by parts without getting a boundary term.

$$\mathcal{E} \;=\; -\frac{h^2 b_2}{2}\left(e'(h) - e'(0)\right) \;-\; \frac{h^4}{4!}\int_0^h e'''(x)\,\frac{d}{dx}B_4(x/h)\,\mathrm{d}x,$$

where

$$b_4 \;=\; -\frac{1}{30}, \quad \text{and} \quad B_4(x) \;=\; x^4 - 2x^3 + x^2 - \frac{1}{30}$$

Doing the iteration one more time, and using the actual values for b_2 and b_4,

The is the first few terms of the

Euler-Maclaurin formula:

$$\int_0^h f(x)\,\mathrm{d}x \;=\; \frac{h}{2}(f(h) \;+\; f(0)\,) - \frac{h^2}{12}\left(f'(h) - f'(0)\,\right) \;+\; \frac{h^4}{720}\left(f'''(h) - f'''(0)\,\right)$$
$$-\frac{h^6}{6!}\int_0^h f^{\,(5)}(x)\,\frac{d}{dx}B_6(x/h)\,\mathrm{d}x,$$

Integrating by parts to get here, we naturally introduced a set of polynomials. These polynomials have names and are very important for a lot of reasons.

Bernoulli polynomials

https://en.wikipedia.org/wiki/Bernoulli_polynomials (https://en.wikipedia.org/wiki/Bernoulli_polynomials)

Bernoulli numbers

https://en.wikipedia.org/wiki/Bernoulli_number (https://en.wikipedia.org/wiki/Bernoulli_number)

<img src="https://upload.wikimedia.org/wikipedia/commons/1/19/Jakob_Bernoulli.jpg (https://upload.wikimedia.org/wikipedia/commons/1/19/Jakob_Bernoulli.jpg)" >

They're kind of a big deal

Recall how

$$\frac{d}{dx}\,x^n \;=\; n\,x^{n-1}$$

This is likely one of the first things you learned in Calculus. The ***Bernoulli polynomials*** are defined almost the same way

$$B_0(x) = 1, \quad \text{and} \quad \frac{d}{dx}\,B_n(x) \;=\; n\,B_{n-1}(x), \quad \int_0^1 B_n(x)\,\mathrm{d}x \;=\; 0, \quad \text{for} \quad n \geq 1.$$

We can use these definitions to find a relation for the Bernoulli numbers,

$$b_n \;=\; n\int_0^1 x\,B_{n-1}(x)\,\mathrm{d}x \;=\; B_n(0).$$

These polynomials and numbers have a huge number of remarkable properties. The polynomials and numbers show up in every branch of advanced mathematics; topology, algebraic geometry, combinatorics, probability, and more. They also show up in import areas of modern physics. They arise when trying to compute Feynman diagrams in quantum field theory and string theory.

Because of their similarity with monomials, they share a lot of similar properties. For example, they both have simple ***generating functions***:

$$e^{t\,x} \;=\; \sum_{n=0}^{\infty} x^n \, \frac{t^n}{n!}, \qquad \frac{t\,e^{t\,x}}{e^t - 1} \;=\; \sum_{n=0}^{\infty} B_n(x) \, \frac{t^n}{n!}$$

Also, they both satisfy a kind of ***Binomial theorem***,

$$(x+y)^n \;=\; \sum_{k=0}^{n} \frac{n!}{k!\,(n-k)!} \, x^k \, y^{n-k}, \qquad B_n(x+y) \;=\; \sum_{k=0}^{n} \frac{n!}{k!\,(n-k)!} \, B_k(x) \, B_{n-k}(y)$$

One of my favorites is that they relate to the computation of the sums of powers of integers,

$$\sum_{k=1}^{n} k^p \;=\; \int_1^{n+1} B_p(x)\,\mathrm{d}x \;=\; \frac{B_{p+1}(n) - b_{p+1}}{p+1} + n^p$$

The sums of integer powers relates to geometry very naturally:
https://opinionator.blogs.nytimes.com/2010/02/07/rock-groups/
(https://opinionator.blogs.nytimes.com/2010/02/07/rock-groups/)

Break time

You've got another two questions to attempt.

Investigating the `class` of polynomials

Using the `classes` we created earlier in the semester, we can build and manipulate the Bernoulli (or any) polynomials:

So we've made a rudimentary symbolic algebra package! Any time you plug something in to Mathematica and it seems like magic, remember that it's really not that different to what we've already done.

You can access the polynomial and rational classes from this lecture to check your answers from the last tutorial. See if you notice anything to fix too!

Let's see what happens

In [1]:

```
from polynomials import *
from rational import *

import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
one = rational(1)
I = Polynomial({0:one})
X = Polynomial({1:one})
B = {0:I}
T = {0:I,1:X}
B[1] = X - rational(1,2)
b = {0:B[0](0),1:B[1](0)}

for k in range(2,100):
    B[k] = k*B[k-1].integral()
    B[k] = B[k] - B[k].integrate(0,1)
    b[k] = B[k](0)
    T[k] = 2*X*T[k-1] - T[k-2]
```

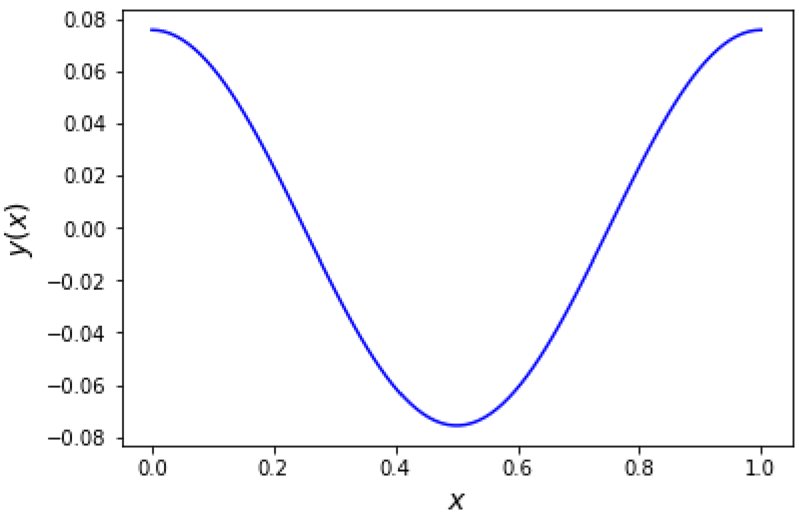
In [3]:

```
r = rational(17,19)

(X*X + r).show()

Y = X*X + r

print(Y(3))

B[10].show()
B[10].plot(0,1,)
```

$\frac{17}{19} + x^2$

188/19

$\frac{5}{66} - \frac{3}{2}x^2 + 5\,x^4 - 7\,x^6 + \frac{15}{2}x^8 - 5\,x^9 + x^{10}$



Here is a simple check that one of the formulae gives the right result:

In [4]:

```
n = 14
print(B[n](0))
print(b[14])
print((n*X*B[n-1]).integrate(0,1) )
```

```
7/6
7/6
7/6
```

Here's what a few of the first polynomials look like:

In [5]:

```
for n in range(1,7): B[n].show()
```

$-\frac{1}{2} + x$

$\frac{1}{6} - x + x^2$

$\frac{1}{2}x - \frac{3}{2}x^2 + x^3$

$-\frac{1}{30} + x^2 - 2x^3 + x^4$

$-\frac{1}{6}x + \frac{5}{3}x^3 - \frac{5}{2}x^4 + x^5$

$\frac{1}{42} - \frac{1}{2}x^2 + \frac{5}{2}x^4 - 3x^5 + x^6$

Here are the first 50 Bernouli number in floating-point and rational form:

In [6]:

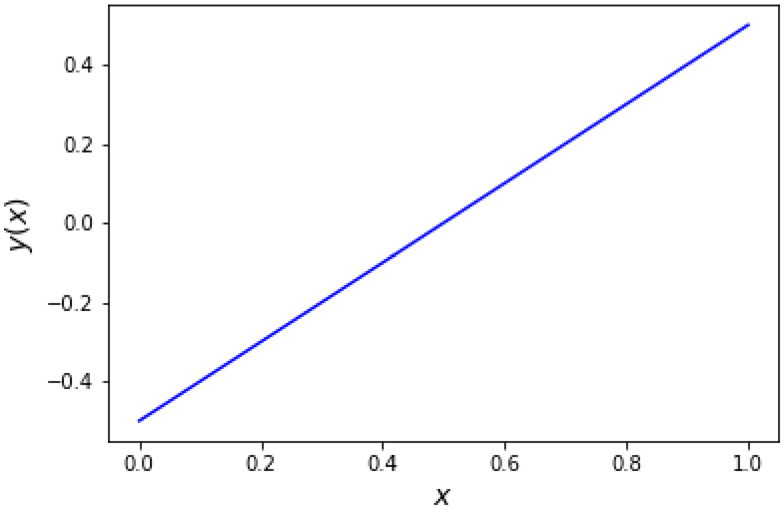
```
for n in range(0,50): print("{:>2d}: {:>9.2e} = {}".format(n, float(b[n]), str(b
[n])))
```

```
 0:  1.00e+00 = 1
 1: -5.00e-01 = -1/2
 2:  1.67e-01 = 1/6
 3:  0.00e+00 = 0
 4: -3.33e-02 = -1/30
 5:  0.00e+00 = 0
 6:  2.38e-02 = 1/42
 7:  0.00e+00 = 0
 8: -3.33e-02 = -1/30
 9:  0.00e+00 = 0
10:  7.58e-02 = 5/66
11:  0.00e+00 = 0
12: -2.53e-01 = -691/2730
13:  0.00e+00 = 0
14:  1.17e+00 = 7/6
15:  0.00e+00 = 0
16: -7.09e+00 = -3617/510
17:  0.00e+00 = 0
18:  5.50e+01 = 43867/798
19:  0.00e+00 = 0
20: -5.29e+02 = -174611/330
21:  0.00e+00 = 0
22:  6.19e+03 = 854513/138
23:  0.00e+00 = 0
24: -8.66e+04 = -236364091/2730
25:  0.00e+00 = 0
26:  1.43e+06 = 8553103/6
27:  0.00e+00 = 0
28: -2.73e+07 = -23749461029/870
29:  0.00e+00 = 0
30:  6.02e+08 = 8615841276005/14322
31:  0.00e+00 = 0
32: -1.51e+10 = -7709321041217/510
33:  0.00e+00 = 0
34:  4.30e+11 = 2577687858367/6
35:  0.00e+00 = 0
36: -1.37e+13 = -26315271553053477373/1919190
37:  0.00e+00 = 0
38:  4.88e+14 = 2929993913841559/6
39:  0.00e+00 = 0
40: -1.93e+16 = -261082718496449122051/13530
41:  0.00e+00 = 0
42:  8.42e+17 = 1520097643918070802691/1806
43:  0.00e+00 = 0
44: -4.03e+19 = -27833269579301024235023/690
45:  0.00e+00 = 0
46:  2.12e+21 = 596451111593912163277961/282
47:  0.00e+00 = 0
48: -1.21e+23 = -5609403368997817686249127547/46410
49:  0.00e+00 = 0
```

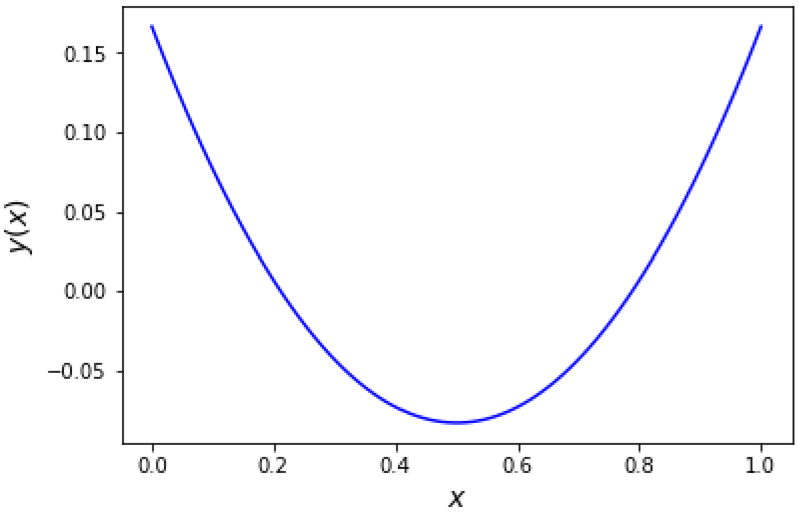
Here are some plots of the fist few polynomials:

In [7]:

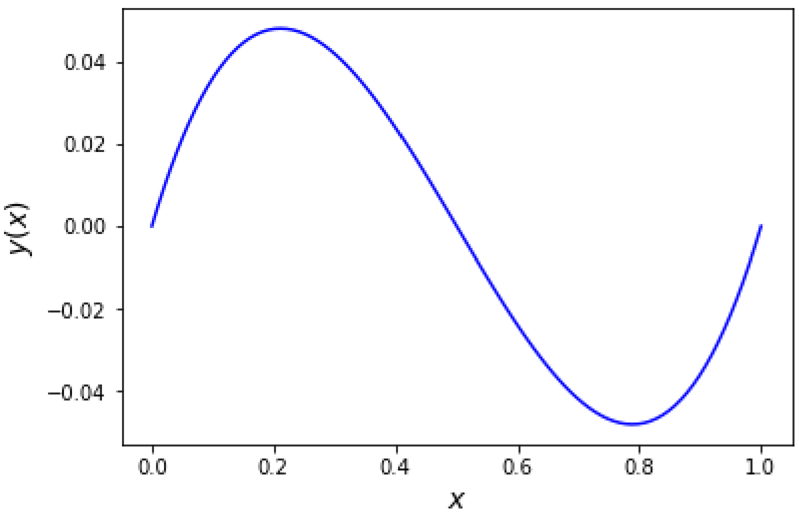
```
for i in range(1,7):  B[i].plot(0,1)
```



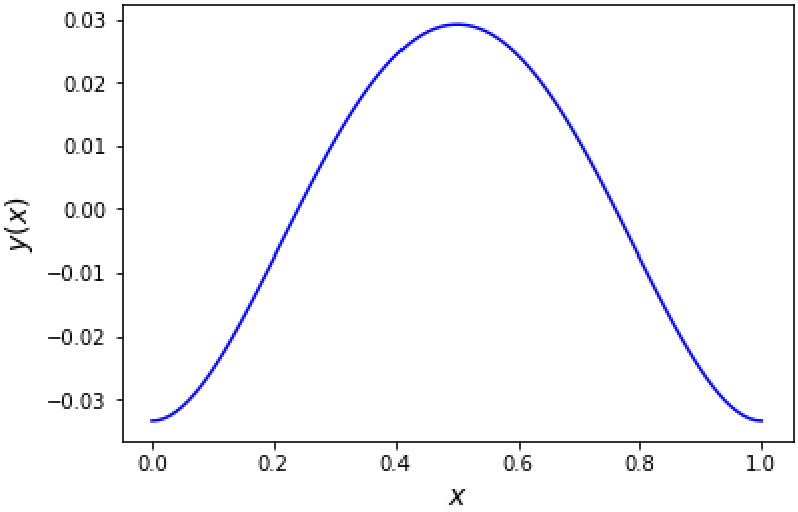
0.4
0.2
0.0
−0.2
−0.4
y(x)
0.0
0.2
0.4
0.6
0.8
1.0
x



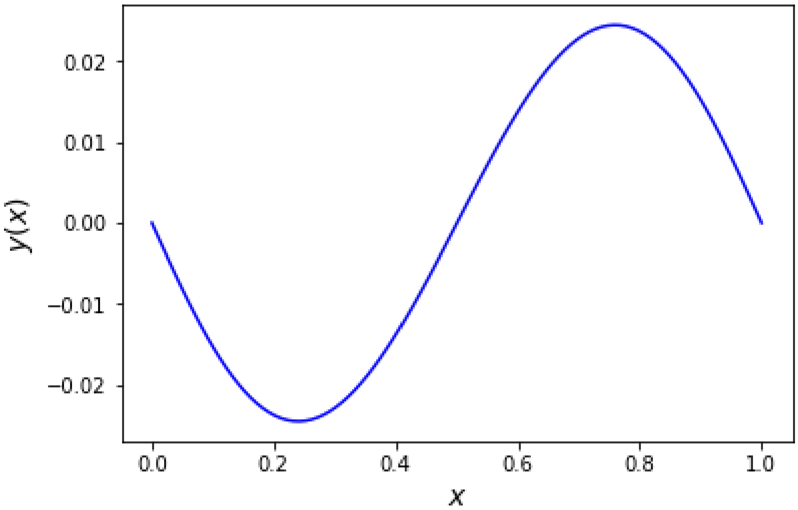
0.15
0.10
0.05
0.00
−0.05
y(x)
0.0
0.2
0.4
0.6
0.8
1.0
x



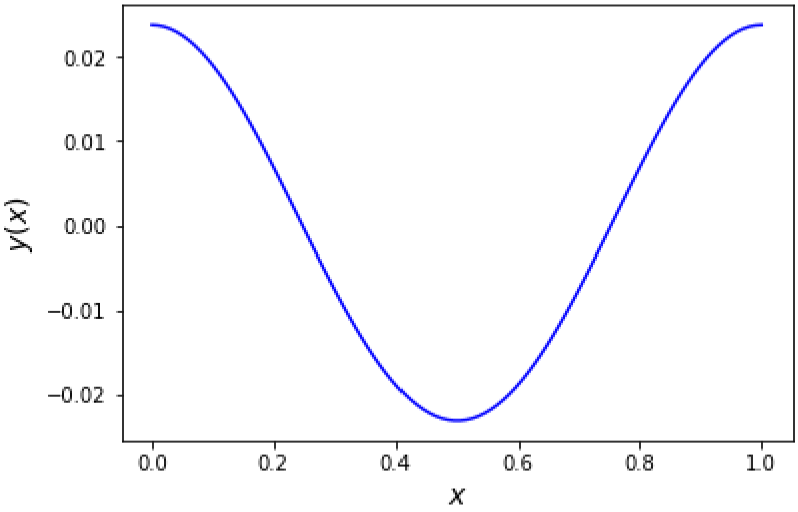
0.04
0.02
0.00
−0.02
−0.04
y(x)
0.0
0.2
0.4
0.6
0.8
1.0
x



0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
y(x)
0.0
0.2
0.4
0.6
0.8
1.0
x



0.02
0.01
0.00
−0.01
−0.02
y(x)
0.0
0.2
0.4
0.6
0.8
1.0
x



0.02
0.01
0.00
−0.01
−0.02
y(x)
0.0
0.2
0.4
0.6
0.8
1.0
x

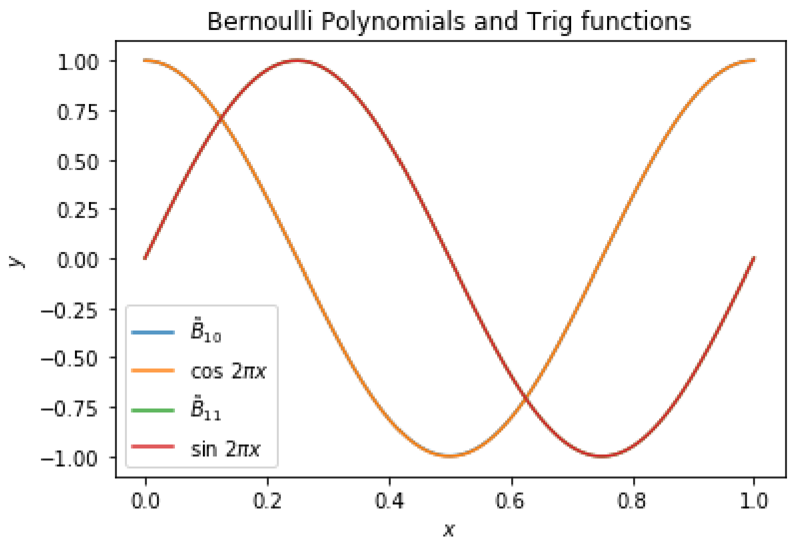
Perhaps you can see a pattern? Let's investigate:

In [8]:

```
x = np.linspace(0,1,200)
s, c = np.sin(2*np.pi*x), np.cos(2*np.pi*x)

y10 = B[10](x)
y10 /= max(y10)

y11 = B[11](x)
y11 /= max(y11)

fig, ax = plt.subplots()
ax.plot(x,y10,label=r'$\tilde{B}_{10}$')
ax.plot(x,c,label='$\cos{\ 2\pi x}$')

ax.plot(x,y11,label=r'$\tilde{B}_{11}$')
ax.plot(x,s,label='$\sin{\ 2\pi x}$')

ax.set(xlabel='$x$',ylabel='$y$',title='Bernoulli Polynomials and Trig function
s')
ax.legend()
```

Out[8]:

```
<matplotlib.legend.Legend at 0x7f23bbe427b8>
```



The polynomials converge to sine and/or cosine for large degree,

$$B_n(x) \quad \longrightarrow \quad \frac{2\,n!}{(2\pi)^n}\left[cos\left(2\,\pi\,x - \frac{(n+2)\,\pi}{2} \right) \;+\; \mathcal{O}\left(2^{-n} \right) \right], \quad \text{as} \quad n \;\rightarrow \infty.$$

It's easy to find interesting properties fo $B_n(x)$.

What about

$$B_n(x+1) - B_n(x) \quad ?$$

In [9]:

```
for n in range(10): (B[n](X+1) - B[n](X)).show()
```

0

1

$2x$

$3x^2$

$4x^3$

$5x^4$

$6x^5$

$7x^6$

$8x^7$

$9x^8$

What about

$$\frac{1}{r}\sum_{k=0}^{r-1} B_n(\,(\,x+k\,)\,/\,r\,) \qquad = \qquad ?$$

In [10]:

```
r = 6
q = rational(1,r)

for n in range(10):
    S = 0*X
    for k in range(r): S = S + q*B[n](q*(X+k))
    print('(S(%s,%s) == B[%s]/%s**%s) = %s' %(r,n,n,r,n,str(S == (q**n) * B[n
])))
```

```
(S(6,0) == B[0]/6**0) = True
(S(6,1) == B[1]/6**1) = True
(S(6,2) == B[2]/6**2) = True
(S(6,3) == B[3]/6**3) = True
(S(6,4) == B[4]/6**4) = True
(S(6,5) == B[5]/6**5) = True
(S(6,6) == B[6]/6**6) = True
(S(6,7) == B[7]/6**7) = True
(S(6,8) == B[8]/6**8) = True
(S(6,9) == B[9]/6**9) = True
```

$$\frac{1}{r}\sum_{k=0}^{r-1} B_n(\,(\,x+k\,)\,/\,r\,) \qquad = \qquad \frac{1}{r^n}B_n(x)$$

This is an ***eigenvalue*** relationship.

There are many more interesting things about Bernoulli polynomials.