Gibb's Overshoot: A Physics-style Analysis

First clear all symbols to avoid possible problems from definitions used earlier.

In[1]:=
```
Clear["Global`*"]
```

Square wave Fourier series

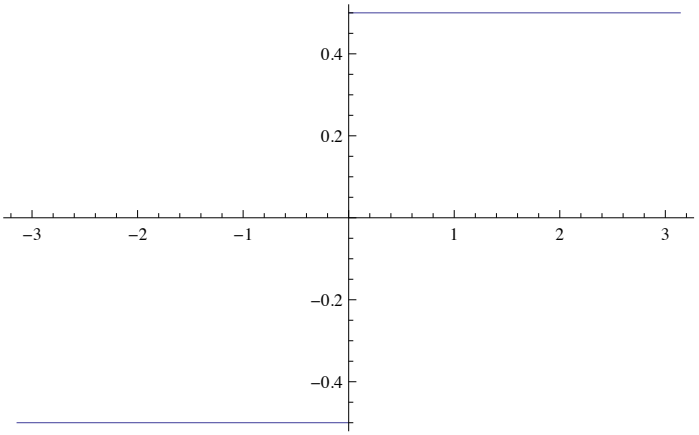
Let's make a theta function from -π to +π with the step at x=0.

In[2]:=
```
fstep[x_, h_] := Piecewise[{{-h / 2, -Pi < x ≤ 0}, {+h / 2, 0 < x < Pi}}]
```

Always plot to check it works with h=1:

In[3]:=
```
Plot[fstep[x, 1], {x, -Pi, Pi}]
```

Out[3]=



Find the coefficient using the period = 2π and we will only have sines because the function is odd about x=0:

In[4]:=
```
an[n_, h_] =
 2 / (2 Pi) Integrate[fstep[x, h] * Sin[(2 Pi) n x / (2 Pi)],
   {x, -Pi, Pi}, Assumptions → {h > 0, Element[{n}, Integers]}]
```

Out[4]=
$$\frac{2\,h\,\text{Sin}\left[\frac{n\,\pi}{2}\right]^2}{n\,\pi}$$

We have included two assumptions in the integral. Neither actually helps here, but you should include as a general practice anything you know about the parameters. Note that x is a dummy integration variable.

If we use FullSimplify and tell *Mathematica* n is an integer, then we get:

In[5]:=
```
an[n_, h_] = FullSimplify[
  2 / (2 Pi) Integrate[
    fstep[x, h] * Sin[(2 Pi) n x / (2 Pi)], {x, -Pi, Pi}, Assumptions → {h > 0}],
  Assumptions → {Element[{n}, Integers]}]
```

Out[5]=
$$-\frac{(-1+(-1)^n)\,h}{n\,\pi}$$

We see that h is an overall factor, so we can set h=1 without losing any generality of our analysis. It is like choosing units for convenience so that our basic length is 1.

We used = rather than := in defining the function an because we want the integration to be done at this point in time, rather than being delayed until the function an was evaluated later.

In[6]:=
```
Table[an[n, 1], {n, 1, 5}]
```

Out[6]=
$$\left\{\frac{2}{\pi}, 0, \frac{2}{3\,\pi}, 0, \frac{2}{5\,\pi}\right\}$$

Define a function to sum nmax terms in the Fourier series:

In[7]:=
```
fsumN[x_, nmax_] := Sum[an[n, 1] * Sin[n x], {n, 1, nmax}]
```

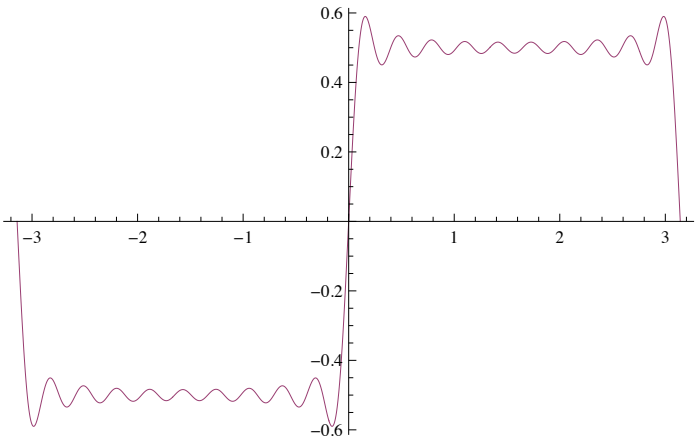
Always check that it does what you want!

In[8]:=
```
fsumN[x, 10]
```

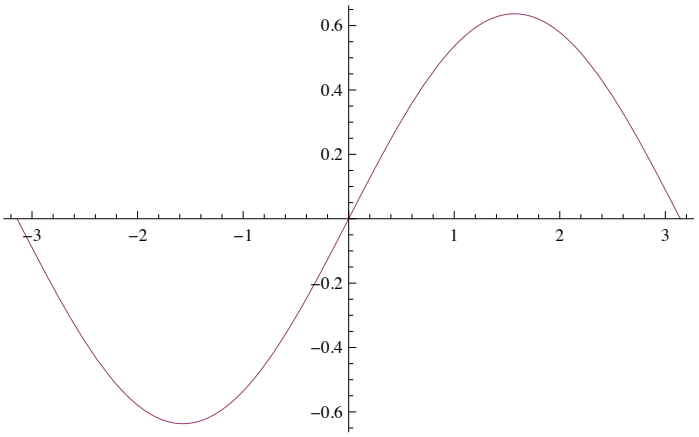
Out[8]=
$$\frac{2\,\text{Sin}[x]}{\pi} + \frac{2\,\text{Sin}[3\,x]}{3\,\pi} + \frac{2\,\text{Sin}[5\,x]}{5\,\pi} + \frac{2\,\text{Sin}[7\,x]}{7\,\pi} + \frac{2\,\text{Sin}[9\,x]}{9\,\pi}$$

Let's first look at a plot of a lot of terms to check that it works:

In[9]:=
```
Plot[{fstep[x], fsumN[x, 20]}, {x, -Pi, Pi}]
```

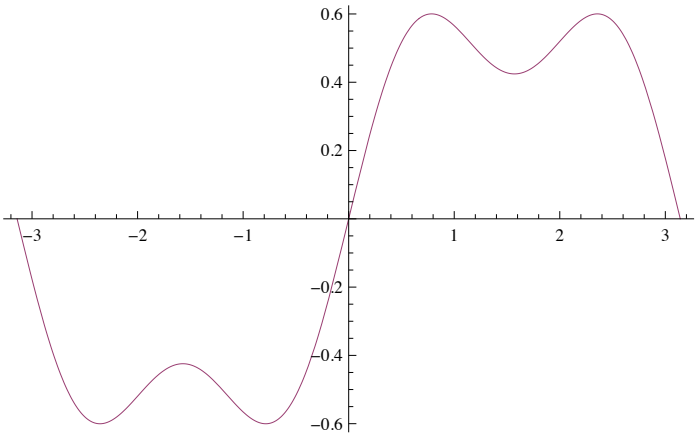
Out[9]=


In[10]:=
```
Plot[{fstep[x], fsumN[x, 1]}, {x, -Pi, Pi}]
```

Out[10]=


In[11]:=
```
Plot[{fstep[x], fsumN[x, 3]}, {x, -Pi, Pi}]
```

Out[11]=



In[12]:=
```
? FindMaximum
```

FindMaximum[f, x] searches for a local maximum in f, starting from an automatically selected point.
FindMaximum[f, {x, x_0}] searches for a local maximum in f, starting from the point $x = x_0$.
FindMaximum[f, {{x, x_0}, {y, y_0}, …}] searches for a local maximum in a function of several variables.
FindMaximum[{f, *cons*}, {{x, x_0}, {y, y_0}, …}] searches for a local maximum subject to the constraints *cons*.
FindMaximum[{f, *cons*}, {x, y, …}] starts from a point within the region defined by the constraints. ≫

In[13]:=
```
FindMaximum[{fsumN[x, 1], 0 < x < Pi}, {x, 0.1}]
```

Out[13]=
```
{0.63662, {x → 1.5708}}
```

In[14]:=
```
FindMaximum[{fsumN[x, 3], 0 < x < Pi / 2}, {x, 0.1}]
```

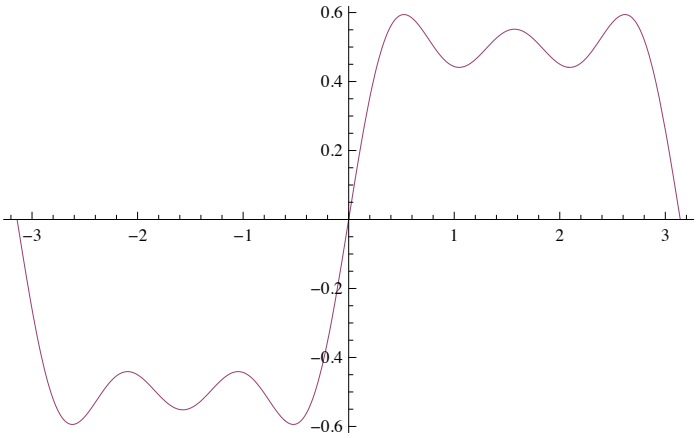
Out[14]=
```
{0.600211, {x → 0.785398}}
```

In[15]:=
```
Pi / 4.
```

Out[15]=
```
0.785398
```

In[16]:=
```
Plot[{fstep[x], fsumN[x, 5]}, {x, -Pi, Pi}]
```

Out[16]=



In[17]:=
```
FindMaximum[{fsumN[x, 5], 0 < x < Pi / 2}, {x, 0.1}]
```

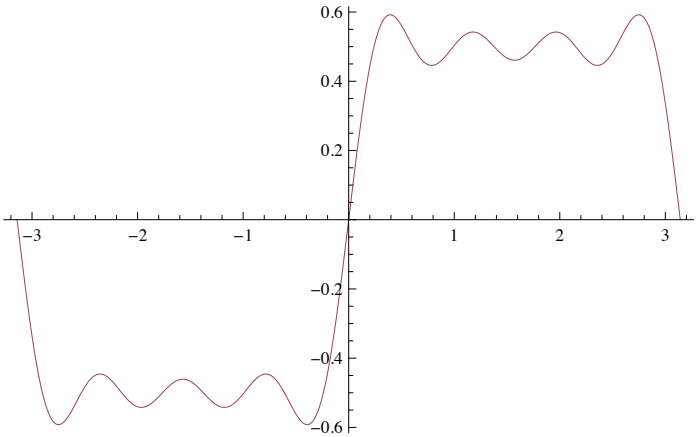
Out[17]=
```
{0.594178, {x → 0.523599}}
```

In[18]:=
```
Pi / 6.
```

Out[18]=
```
0.523599
```

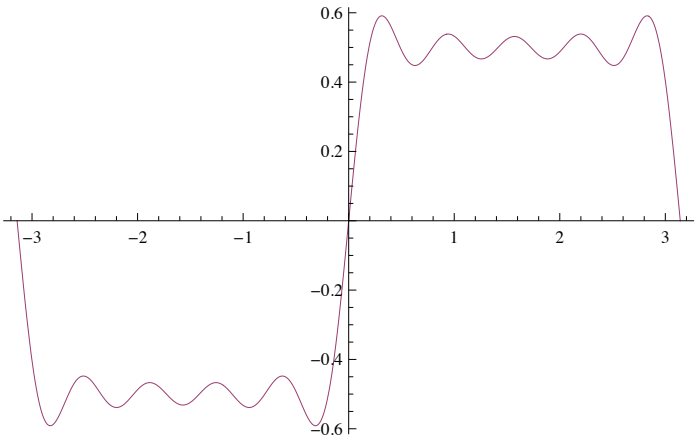
In[19]:=
```
Plot[{fstep[x], fsumN[x, 7]}, {x, -Pi, Pi}]
```

Out[19]=



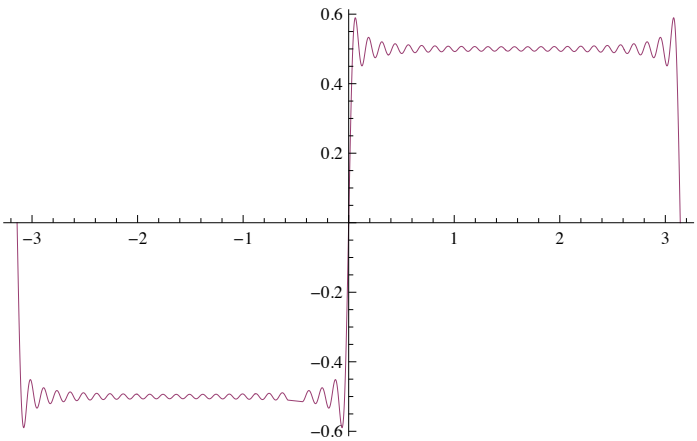
In[20]:=
```
Plot[{fstep[x], fsumN[x, 9]}, {x, -Pi, Pi}]
```

Out[20]=



In[21]:=
```
Plot[{fstep[x], fsumN[x, 50]}, {x, -Pi, Pi}]
```

Out[21]=



In[22]:=
```
FindMaximum[{fsumN[x, 50], 0 < x < Pi / 2}, {x, 0.1}]
```

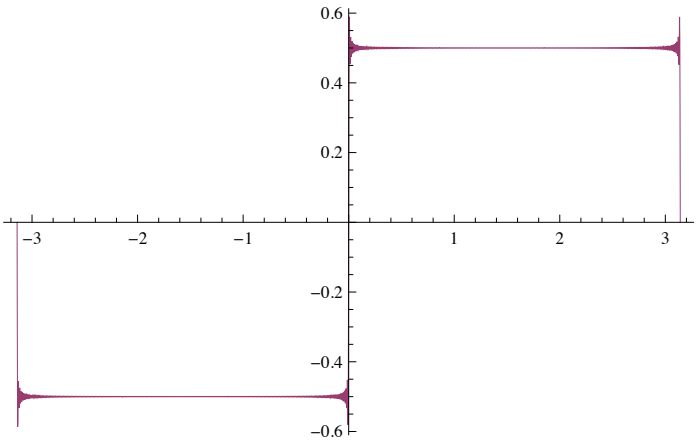
Out[22]=
```
{0.589557, {x → 0.0628319}}
```

In[23]:=
```
Pi / 50.
```

Out[23]=
```
0.0628319
```

In[24]:=
```
Plot[{fstep[x], fsumN[x, 500]}, {x, -Pi, Pi}]
```

Out[24]=



So the maximum is at x = π/nmax if we sum up to nmax with even values of nmax (so we sum over odd n = 1 to nmax-1). The overshoot area moves closer and closer to the origin with increasing nmax, so we expect convergence everywhere except at the point of discontinuity (or at least only nearby). Check the limit:

In[25]:=
```
gibbs[nmax_] := N[fsumN[Pi / nmax, nmax]]
```

In[26]:=
```
gibbs[10]
```

Out[26]=
```
0.591164
```

In[27]:=
```
gibbs[50]
```

Out[27]=
```
0.589557
```

In[28]:=
```
gibbs[1000]
```

Out[28]=
```
0.58949
```

Looks like the overshoot has a well defined limit of about 0.0895 above the square well height of 1/2.

In Arfken (7th edition, section 19.3), the overshoot is shown to be (with our choice of height):

In[29]:=
$$\int_0^{\pi} \frac{\sin(t)}{\pi t}\, dt - 1/2$$

Out[29]=
$$-\frac{1}{2} + \frac{\text{SinIntegral}[\pi]}{\pi}$$

We can easily show that our sum evaluated at x = π/nmax is an approximation to this integral, becoming exact as nmax -> ∞.

In[30]:=
```
Integrate[(1 / Pi) Sin[t] / t, {t, 0, Pi}] - 1 / 2  // N
```

Out[30]=
```
0.0894899
```

The general result is that the overshoot is about 9% of the discontinuity (which is from -1/2 to +1/2 here, so equal to 1).