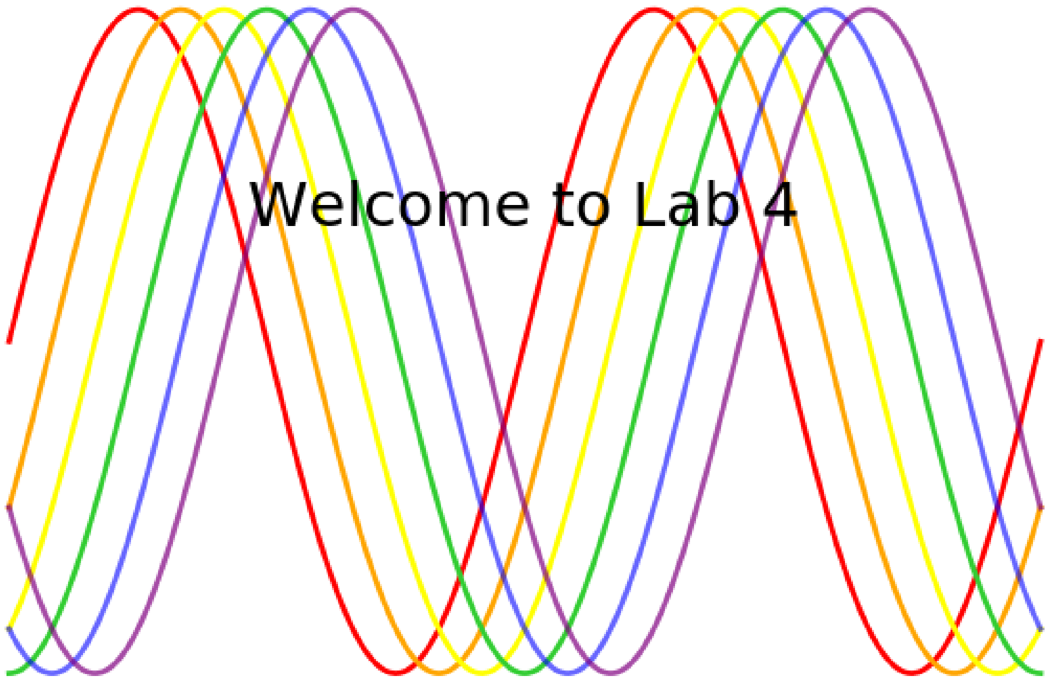
# Fall25-1110-Lab-4

December 30, 2025

[42]:
```
# remember to include these two lines of code at the start of your document!
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

## 1 MATH1110 Lab 4: Errors

[14]:
```
x = var('x')
g1 = plot( sin(x), -2*pi, 2*pi, color = 'red', thickness = 2)
g2 = plot( sin(x-pi/6), -2*pi, 2*pi, color = 'orange', thickness = 2)
g3 = plot( sin(x-pi/3), -2*pi, 2*pi, color = 'yellow', thickness = 2)
g4 = plot( sin(x-pi/2), -2*pi, 2*pi, color = 'limegreen', thickness = 2)
g5 = plot( sin(x-2*pi/3), -2*pi, 2*pi, color = 'blue', thickness = 2, alpha = 0.
  ↪6)
g6 = plot( sin(x-5*pi/6), -2*pi, 2*pi, color = 'purple', thickness = 2, alpha =␣
  ↪0.7)
show((g1+g2+g3+g4+g5+g6)+text('Welcome to Lab 4', (0, 0.4), fontsize = 23,␣
  ↪color = 'black'), \
     xmin = -2*pi, xmax = 2*pi, axes = False)
```



**Objectives for today:** This one's a little bit different...

1. Review of solving
2. Common error codes
3. Find the errors!

## 1.1 1. Review of solving equations

For Assignment 2 (which is due on friday), you'll have to know how to find the intersection of two curves by solving a system of equations. Here is another example of that process:

Plot the two implicitly defined curves $f_1$ and $f_2$ in different colours for $-5 \leq x \leq 15$, $-4 \leq y \leq 4$. Find their intersection using *solve()*.

[1]:
```
x = var('x')
y = var('y')
f1 = x==y^2+1
f2 = tan(ln(y^3-x*y))==0
```

Clear as mud? Any questions here?

## 1.2 2. Common error codes

Below are some examples of errors you'll commonly see while coding. Usually ***the very last line of the error message*** is most helpful.

[16]:
```
clear_vars()
y = t-1
# undefined variable ... "name 't' is not defined"
```

[87]:
```
clear_vars()
x = var('x')
h(x) = 4x+9/2
# SyntaxError
# missing a multiplication sign
```

Sage will sometimes put an arrow where the error was detected, or reference which line the problem occurred. Very handy!

A NameError can also mean that the command being used is not recognized:

[32]:
```
Show(h)
# anyone see the issue here?
```

[94]:
```
3*(cos( (2^4 - 1)*pi/6)
# parentheses don't match
# usually it says "unexpected EOF while parsing" if there's a close parenthesis␣
 ↪missing from the end
```

Notice that when you place your curser next to a set of parentheses, the complimenting one lights up in green. If there is not matching parenthesis, then it will be red.

The error code looks different if there's a missing opening parenthesis:

[36]:
```
3*cos( (2^4 - 1)*pi/6))
```

[96]:
```
x = var('x')
limit(sin(x))
# a ValueError will occur when you're missing a mandatory entry within a command
# it also gives you an example of what a correctly used limit command should␣
 ↪look like!
```

[38]:
```
plot(-x^2, colour = 'orange')
# Sage is trying real hard but can't figure it out
# A runtime error is produced when an option within a command isn't recognized␣
 ↪(colour has the wrong spelling)
```

[39]:
```
plot(-x^2, color = orange)
# and another NameError if you forget to put a bit of text in quotes/
 ↪apostrophes that should be
```

The trickiest errors to find are often those with parentheses, as sometimes the error code will point to the line *after* where the parenthsis was forgotten. Watch out for these instances!

When in doubt, read the last line of the error code and see if Sage is giving you any arrows to hint at where the error is occuring.

## 1.3 3. Find the errors!

Now it's your turn to work on those coding hawk eyes. Correct the following code and breifly describe in a #comment what the issue was. There may be more than one error to fix.

If you don't get to them all, don't worry about it! It's just for practice.

```
[ ]: #(1) Show the function
     r = var('r')
     b = log(r+1)
     show(B)
```

```
[ ]: #(2) Plot the function
     plot(b, r, -10, 10, ymin -10, ymax = 10)
```

```
[ ]: #(3) Show the function
     h = sin*(3*r)/cos*(r+1)
     show(h)
```

```
[ ]: #(4) Expand the function g
     g = (r-1)(r+8)(2*r+1)(r-3)
     expand(g)
```

```
[ ]: #(5) Plot the function
     a5 = log(4*r)-r^3
     plot(a5, 10, -10, ymin = -10, ymax = 10)
```

```
[41]: #(6) Plot the function
      plot=(a5, ymin = -10, ymax = 10, colour = purple)
```

```
[ ]: #(7) Plot the two functions on the same graph
     clear_vars()
     x = var('x')
     f1 = (3*(cos(x)) - sin(1/x)
     f2 = cot( e^x ) # Note that e is a number
     plot( f1, f2, x, -pi, pi, ymin = -10, ymax = 10, detect_poles = 'show')
```

```
[ ]: #(8) Clear variables
     Clear_vars()
```

```
[ ]: #(9) Find the derivative of y using the limit definition
     t = var('t') ; h = var('h')
     y = (t^2 +1) / ((t+3)*(t^2 - 2))
     d9 = ( y(t+h) - y(t) ) / h
     limit(t, h = 0)
```

```
[ ]: #(10) Find the one-sided limits of y as t approaches -3
     limit(y, t = -3, dir = -)
     limit(y, t = -3, dir = +)
```

[ ]:
```
#(11) Add (3/4)*s^3 to the function g
clear_vars()
s = vars('s')
g = s^2 + (s^3)/4
show( g - (3/4)*s^3 )
```

[ ]:
```
#(12) Plot the function
Plot( s*csc(s)) , s, -10, 10)
```

[42]:
```
#(13) Plot the lines together on a graph
line1 = line( [(-1, -1), (1, 4)], color = 'pink')
line2 = line( [(-3, -1), (3, 1)], color = 'orange')
line3 = line( [(-3, 2), (4, 0)], color = 'navy')

# no error code, but where is the graph?
```

[5]:
```
#(14) Plot the four functions and points together on the graph
clear_vars()
ps = points( [(2.3, 2.8), (3.7, 2.8)], size = 70)
x = var('x')
curve1 = plot( (x-3)^2/2 + 1, x, 2.3, 3.7, thickness = 3, xmin = 0, xmax = 5)
curve2 = plot( 4*(x-3)^2 + 2, x, 2.9, 3.1, \
              title = "I'm so sorry this was supposed to look nice but it's a␣
 ↪bit scary .. Happy early Halloween!")
curve3 = plot( cos(x - 2.3) + 2.1, 2.2, 2.4)
curve4 = plot( cos(x - 3.7) + 2.1, 3.6, 3.8)
show(ps, curve1, curve2, curve3, curve4, xmin = 1, xmax = 5, ymin = 0, ymax = 4)
```

#(15) Show the function $f = \frac{\cos(x-x^2)\sin\frac{x}{2}}{3+x}$

[44]:
```
#(15) Show the function
clear_vars()
x = var('x')
f = (cos(x-x^2))sin(x/2) / 3+x
show(f)
```

### 1.3.1 Reminders

Assignment 2 is due this friday. Submit a pdf of your code to Blackboard.

Be sure to save your work!

[ ]:
```

```