

[More Stuff about interpolation, etc.

[want to load the CurveFitting routines

> with(CurveFitting)

[ArrayInterpolation, BSpline, BSplineCurve, Interactive, LeastSquares, Lowess,
PolynomialInterpolation, RationalInterpolation, Spline, ThieleInterpolation]

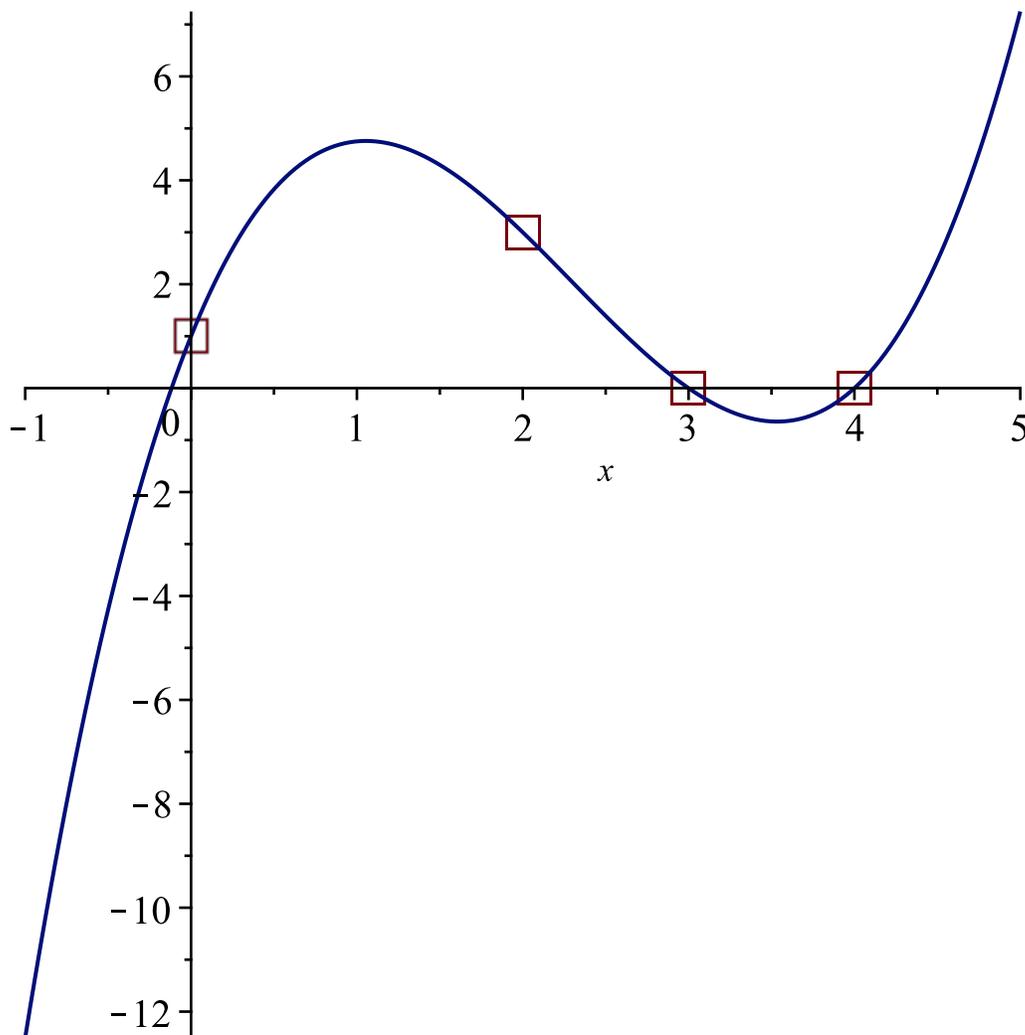
(1)

> p := PolynomialInterpolation([[0, 1], [2, 3], [3, 0], [4, 0]], x)

$$p := \frac{17}{24} x^3 - \frac{39}{8} x^2 + \frac{95}{12} x + 1$$

(2)

> plot([[[0, 1], [2, 3], [3, 0], [4, 0]], p], x=-1 .. 5, style=[point, line], symbolsize=24,
symbol=box)



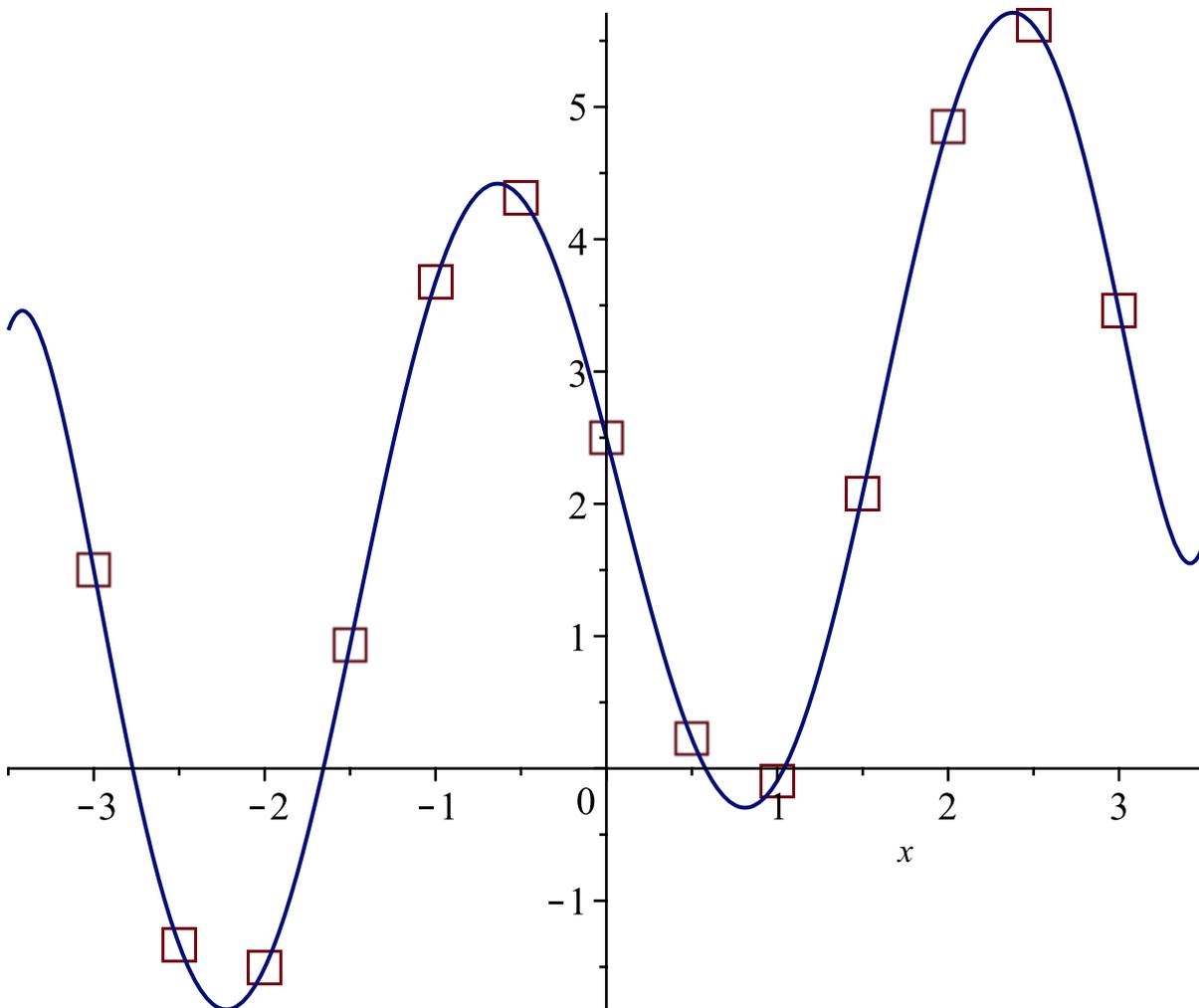
> Dat := [
[-3.0, 1.500718640],
[-2.5, -1.333413875],
[-2.0, -1.506526723],
[-1.5, .930868789],
[-1.0, 3.678347877],
[-.5, 4.315138568],
[0.0, 2.500000000],
[.5, .225163738],

```
[ 1.0, -.094494713],
[ 1.5, 2.079138715],
[ 2.0, 4.852883103],
[ 2.5, 5.617076061],
[ 3.0, 3.459451646]
];
```

```
Dat := [[-3.0, 1.500718640], [-2.5, -1.333413875], [-2.0, -1.506526723], [-1.5,
0.930868789], [-1.0, 3.678347877], [-0.5, 4.315138568], [0., 2.500000000], [0.5,
0.225163738], [1.0, -0.094494713], [1.5, 2.079138715], [2.0, 4.852883103], [2.5,
5.617076061], [3.0, 3.459451646]]
```

(3)

```
> q := PolynomialInterpolation(Dat, x) :
> plot( [ Dat, q ], x=-3.5 .. 3.5, style=[point, line], symbolsize=24, symbol=box)
```



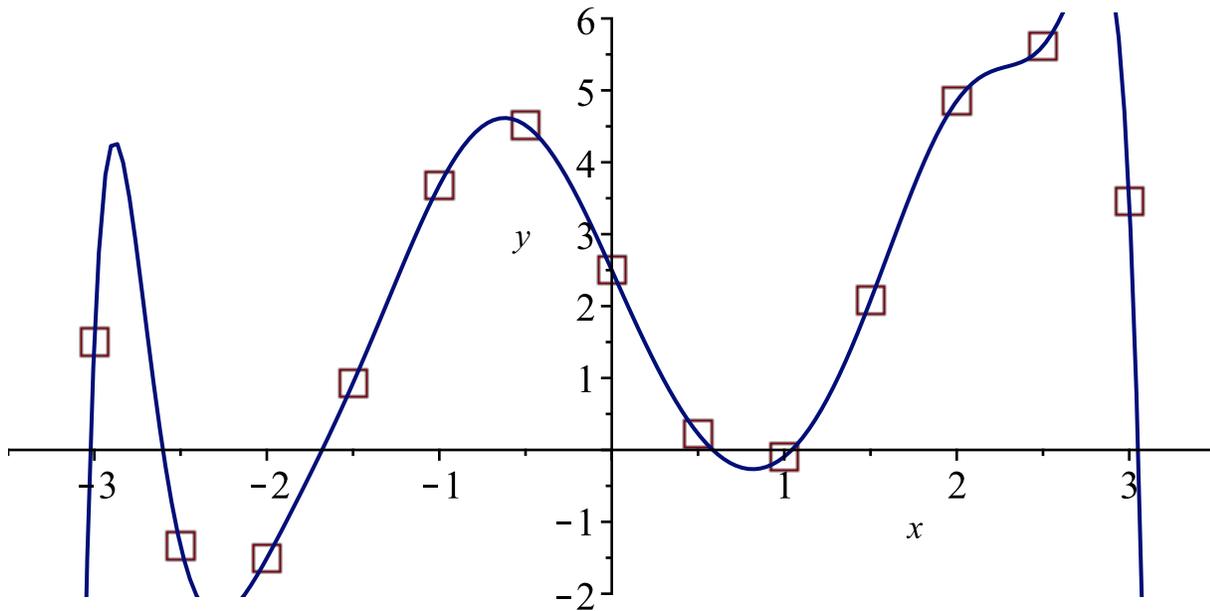
I want **Dat2** to be **Dat** but with the 6th point moved up by .2

```
> Dat2 := Dat :
> Dat2[6, 2] := Dat[6, 2] + .2 :
> Dat2[6], Dat[6]
```

```
[-0.5, 4.515138568], [-0.5, 4.315138568]
```

(4)

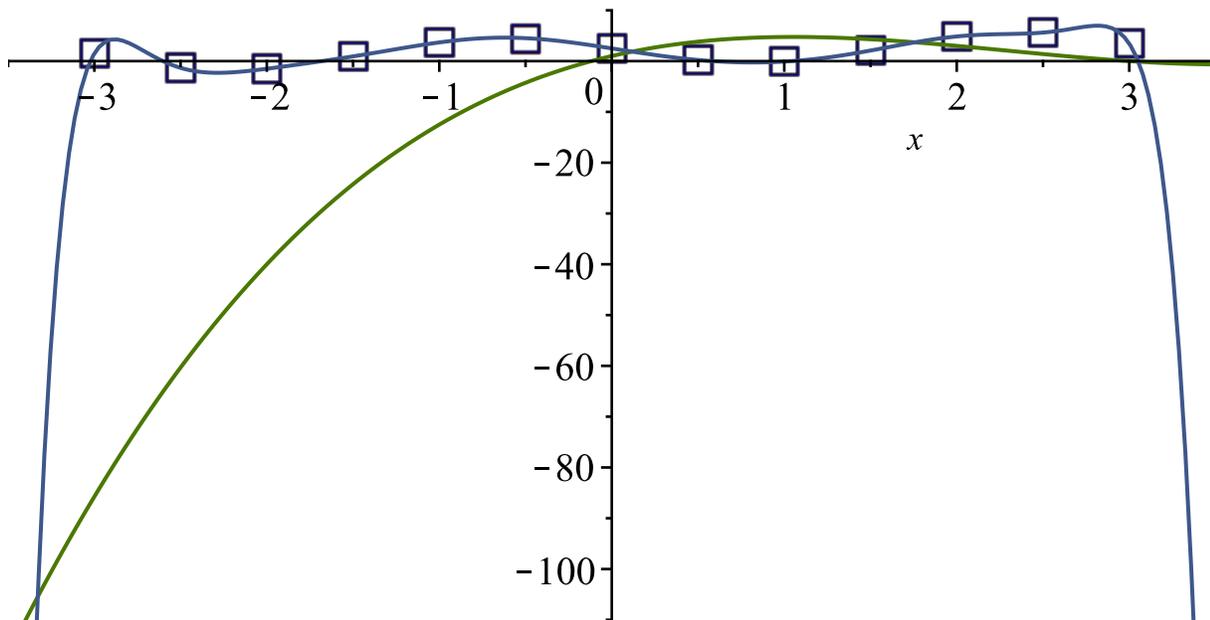
```
> plot( [ Dat2, PolynomialInterpolation(Dat2, x) ], x=-3.5 .. 3.5, y=-2 ..6, style = [ point, line],
        symbolsize = 24, symbol = box)
```



Let's prepare to compare

```
> p2 := PolynomialInterpolation(Dat2, x) :
```

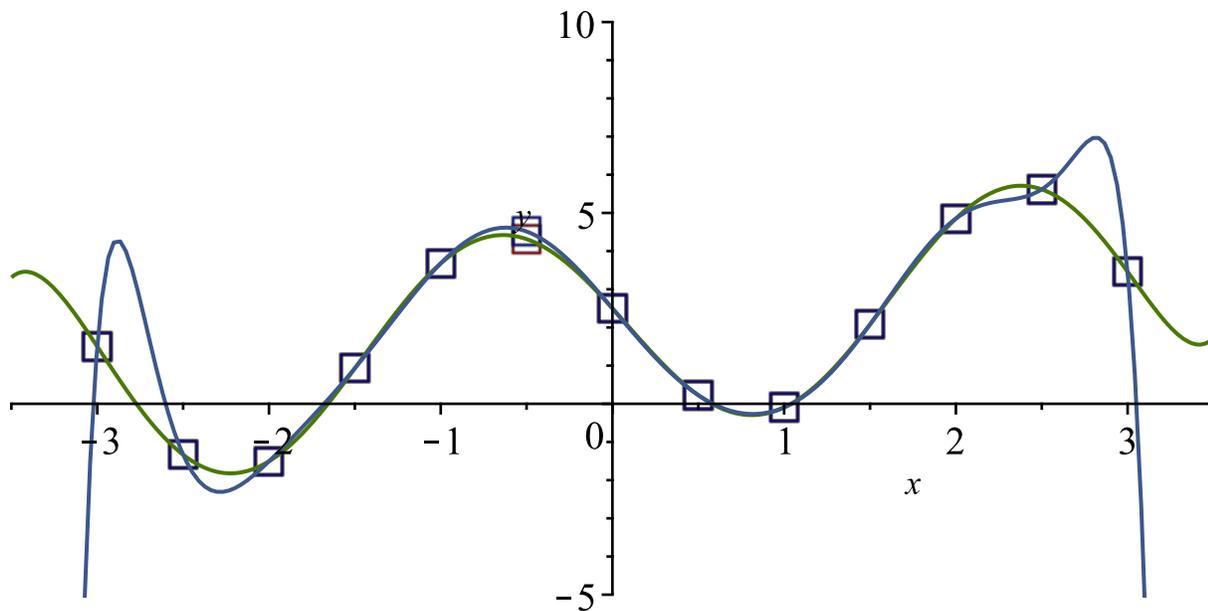
```
> plot( [ Dat, Dat2, p, p2], x=-3.5 .. 3.5, style = [ point, point, line, line],
        symbolsize = 24, symbol = box)
```



This is still screwy, what's wrong? I dunno, let's look. I wanted q, not p. I'm dumb.

```
> p2
```

```
> plot( [ Dat, Dat2, q, p2], x=-3.5 .. 3.5, y=-5 ..10, style = [ point, point, line, line],
        symbolsize = 24, symbol = box)
```



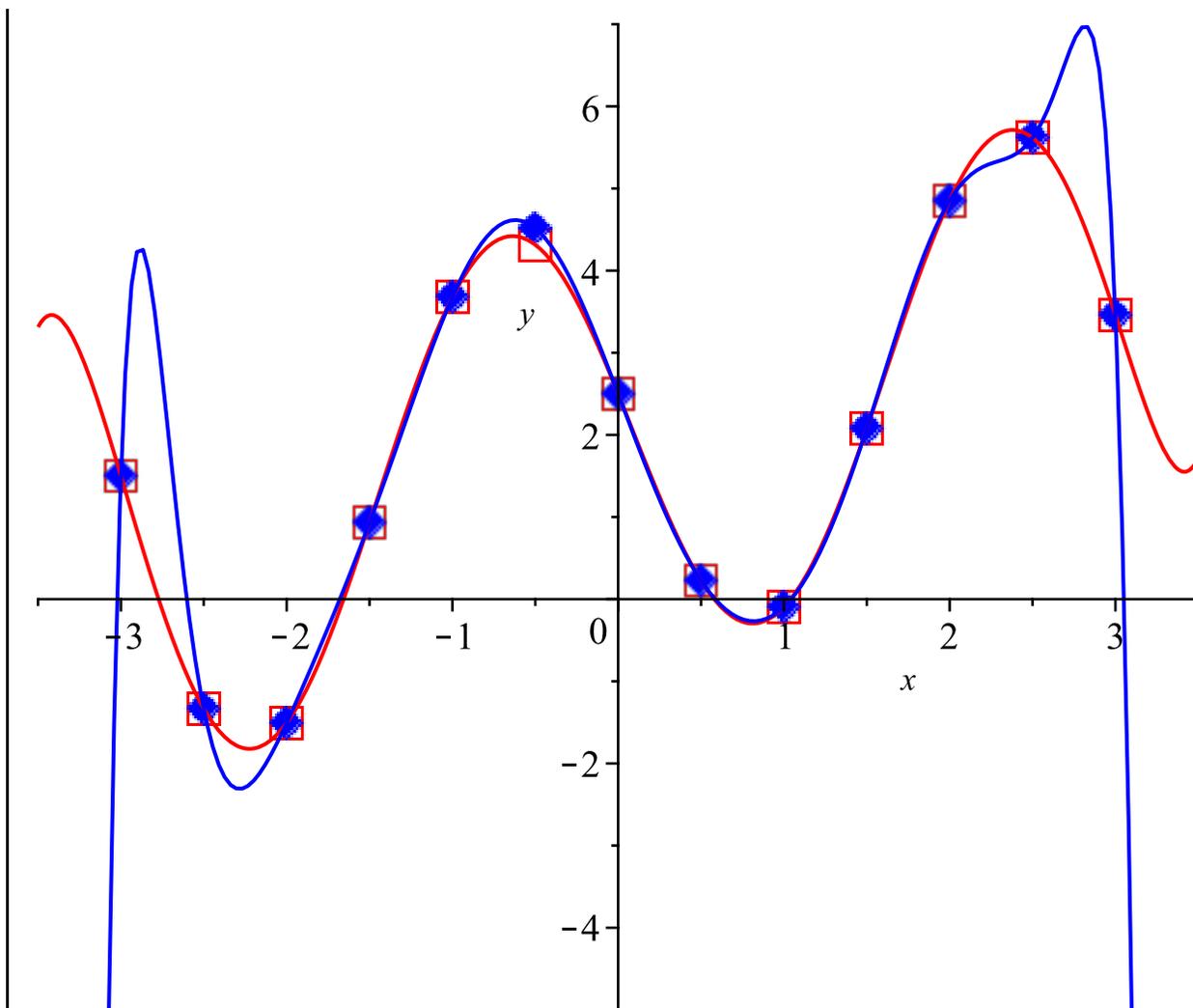
To make it clearer what's happening here, let's draw our points with two types of symbols, and maybe even make colors match.

(5)

```
> plot( [ Dat, Dat2, q, p2], x=-3.5 .. 3.5, y=-5 ..10, style= [ point, point, line, line],
        symbolsize=24, symbol= [ box, diamondwrong])
```

Error, (in plot) expecting option symbol to be of type identical ("asterisk", "box", "cross", "circle", "diagonalcross", "diamond", "point", "solidcircle", "solidbox", "soliddiamond") but received diamondwrong

```
> plot( [ Dat, Dat2, q, p2], x=-3.5 .. 3.5, y=-5 ..7, style= [ point, point, line, line],
        symbolsize=24, symbol= [ box, soliddiamond], color= [ red, blue, red, blue])
```



Before we deal with polynomials being sensitive, let's look at loading data some more.

Rather than copy/paste, I can read from a file.

```
> currentdir( ) # this is the directory maple is running in.
```

```
"C:\Users\ssutherland\Downloads"
```

(6)

```
> Dat := "nope nope";
```

```
Dat := "nope nope"
```

(7)

```
> Dat
```

```
"nope nope"
```

(8)

I've saved my file to my local disk, now run it in maple.

```
> read("09-10-wiggly.txt")
```

```
Dat := [[-3.0, 1.500718640], [-2.5, -1.333413875], [-2.0, -1.506526723], [-1.5, 0.930868789], [-1.0, 3.678347877], [-0.5, 4.315138568], [0., 2.500000000], [0.5, 0.225163738], [1.0, -0.094494713], [1.5, 2.079138715], [2.0, 4.852883103], [2.5, 5.617076061], [3.0, 3.459451646]]
```

(9)

```
> save(Dat2, "09-10-piggly.txt")
```

What If I can't write here, or I want somewhere else?

```
> olddir := currentdir( )
                                olddir := "C:\Users\ssutherland\Downloads" (10)
```

```
> currentdir("C:/Users/ssutherland")
                                "C:\Users\ssutherland" (11)
```

```
> currentdir(olddir)
                                "C:\Users\ssutherland" (12)
```

```
> save(q, p2, Dat, Dat2, "C:/Users/ssutherland/Downloads/higglety.txt")
```

Or. I can read directly from a file on the web.

```
> with(HTTP)
                                [Code, Form, Get, Post, URLDecode, URLEncode, URLParse] (13)
```

```
> status, page, headers :=
    Get("http://www.math.stonybrook.edu/~scott/mat331.fall19/daily/extras/") :
```

```
> page[1..100]
    "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
                                <html>
                                <head>
                                <title>Index of /~scott/mat" (14)
```

```
> status
                                200 (15)
```

```
> Code(status)
                                "OK" (16)
```

```
> status, wigglystuff, headers :=
    Get("http://www.math.stonybrook.edu/~scott/mat331.fall19/daily/extras/09-10-wiggly.
    txt") :
```

```
> wigglystuff
                                "Dat := [ (17)
                                [-3.0, 1.500718640],
                                [-2.5, -1.333413875],
                                [-2.0, -1.506526723],
                                [-1.5, .930868789],
                                [-1.0, 3.678347877],
                                [-.5, 4.315138568],
                                [ 0.0, 2.500000000],
                                [ .5, .225163738],
                                [ 1.0, -.094494713],
                                [ 1.5, 2.079138715],
                                [ 2.0, 4.852883103],
                                [ 2.5, 5.617076061],
                                [ 3.0, 3.459451646]
                                ];
                                "
```

parse says Do the maple commands in this string.

```
> parse(wigglystuff)
```

```
Dat := [[-3.0, 1.500718640], [-2.5, -1.333413875], [-2.0, -1.506526723], [-1.5, 0.930868789], [-1.0, 3.678347877], [-0.5, 4.315138568], [0., 2.500000000], [0.5, 0.225163738], [1.0, -0.094494713], [1.5, 2.079138715], [2.0, 4.852883103], [2.5, 5.617076061], [3.0, 3.459451646]]
```

(18)

I grabbed this from the class page and stuck it here.

```
> #
# Maple procedure to execute maple code stored on theweb
#
ExecuteFromWeb:=proc(URL::string, {printfile::truefalse:=false})
  local n,m, status, webfile, headers;
  # try to get the URL
  status,webfile,headers:=HTTP[Get](URL):
  if ( HTTP[Code](status) <> "OK") then
    error(HTTP[Code](status),URL);
  fi;
  # now interpret the maple on the web page
  n:=0:
  while (n < length(webfile)) do
    m:=n;
    parse(webfile,statement,lastread='n', offset=n);
    if (printfile) then printf("%s",webfile[m+1..n]); fi;
  od:
end:
```

```
> Dat2 := "nuh"
```

```
Dat2 := "nuh"
```

(19)

```
> ClassURL := "http://www.math.stonybrook.edu/~scott/mat331.fall19/daily/extras/";
ClassURL := "http://www.math.stonybrook.edu/~scott/mat331.fall19/daily/extras/"
```

(20)

```
> myURL := cat(ClassURL, "09-10-wiggly2.txt")
```

```
myURL :=
```

(21)

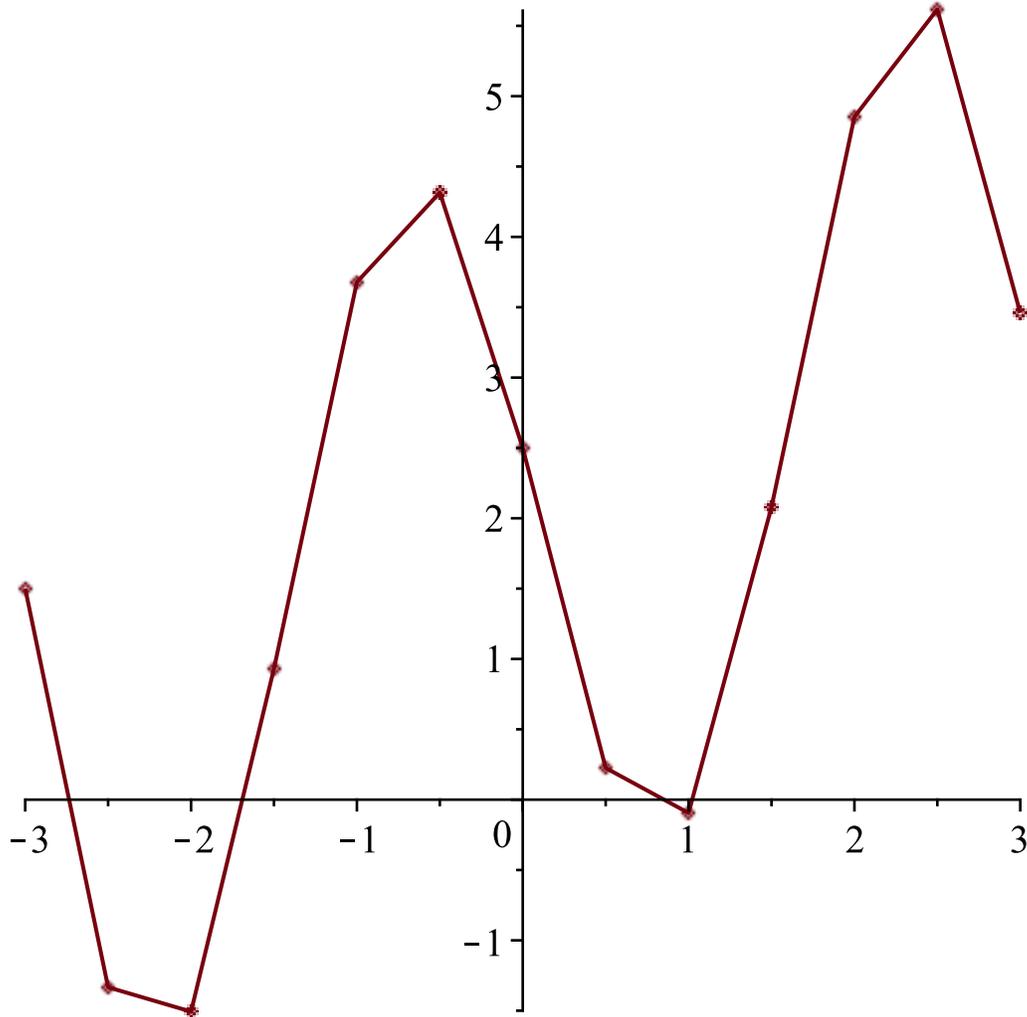
```
"http://www.math.stonybrook.edu/~scott/mat331.fall19/daily/extras/09-10-wiggly2.txt"
```

```
> ExecuteFromWeb(myURL, printfile = true)
```

```
Dat := [
[-3.0, 1.500718640],
[-2.5, -1.333413875],
[-2.0, -1.506526723],
[-1.5, .930868789],
[-1.0, 3.678347877],
[ -.5, 4.315138568],
[ 0.0, 2.500000000],
[ .5, .225163738],
[ 1.0, -.094494713],
[ 1.5, 2.079138715],
[ 2.0, 4.852883103],
[ 2.5, 5.617076061],
[ 3.0, 3.459451646]
];
```

Back to mathy stuff.

```
> plot(Dat, style = pointline)
```

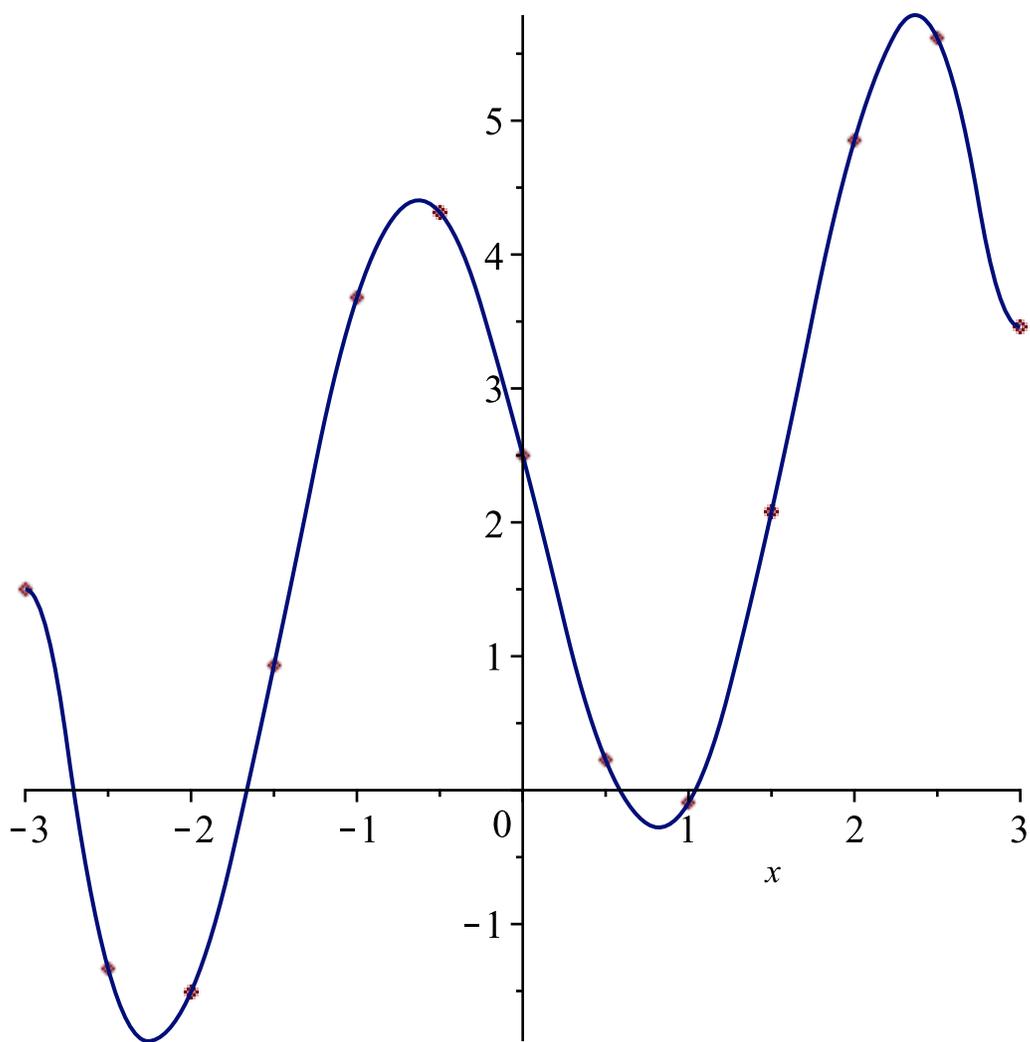


> *Spline(Dat, x, degree = 1)*

-15.50407645 - 5.668265030 x	x < -2.5
-2.198978116 - 0.3462256960 x	x < -2.0
8.243055324 + 4.874791024 x	x < -1.5
9.173306054 + 5.494958176 x	x < -1.0
4.951929260 + 1.273581382 x	x < -0.5
2.500000000 - 3.630277136 x	x < 0.
2.500000000 - 4.549672524 x	x < 0.5
0.5448221890 - 0.6393169020 x	x < 1.0
-4.441761568 + 4.347266856 x	x < 1.5
-6.242094448 + 5.547488776 x	x < 2.0
1.796111280 + 1.528385916 x	x < 2.5
16.40519813 - 4.315248830 x	otherwise

(22)

> *plot([Dat, Spline(Dat, x, degree = 2)], x = -3 .. 3, style = [point, line])*



`> plot([Dat, Spline(Dat, x)], x=-3..3, style=[point, line])`

