

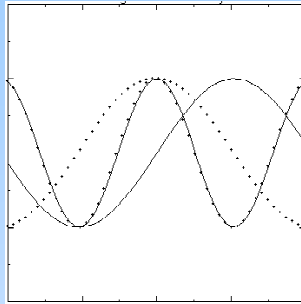
Math 157

Time series plots and phase plane plots
Graphics

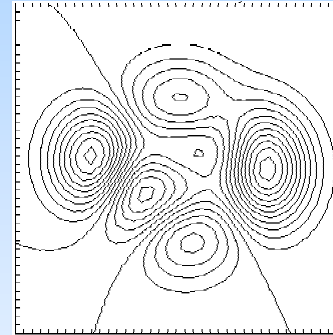
Feb. 4, 2009

Graphics for Scientific/Technical Computation

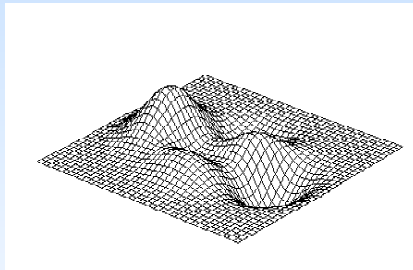
Line Plots



Contour Plots



Surface Plots



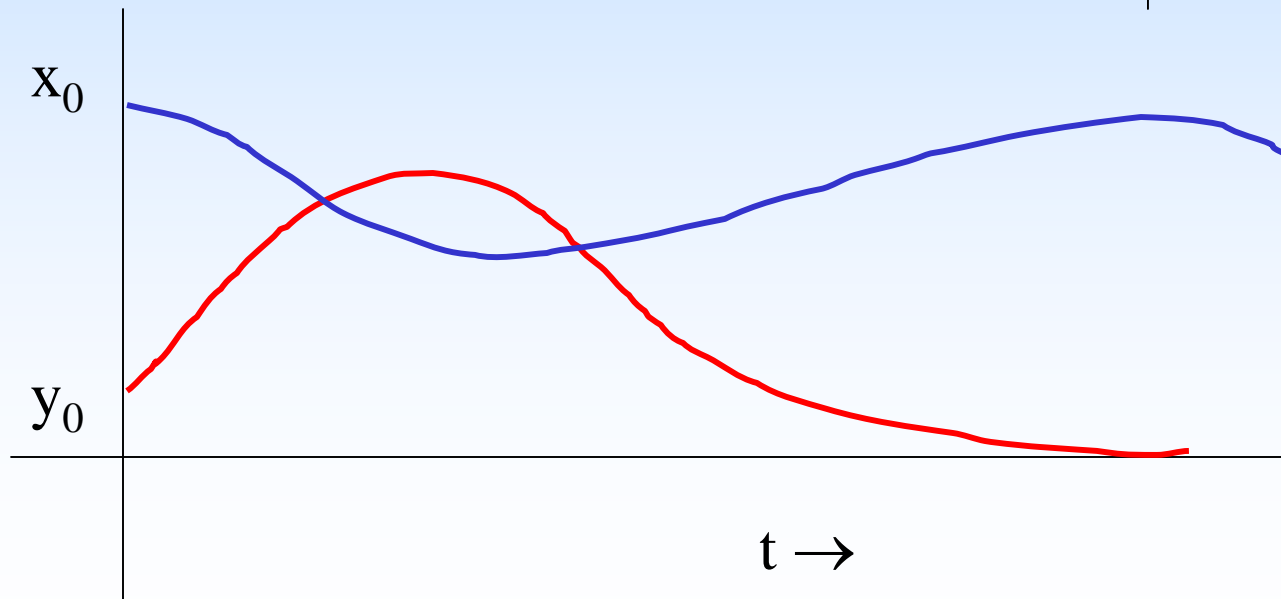
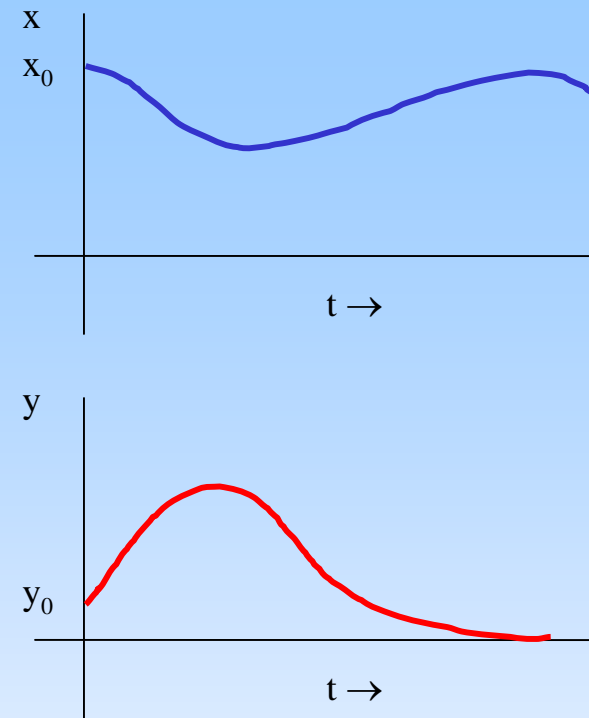
What type of plots do we want?...

- Time series plots
- Phase plane plots

Time series plots

Plots of $(t, x(t))$ and $(t, y(t))$.

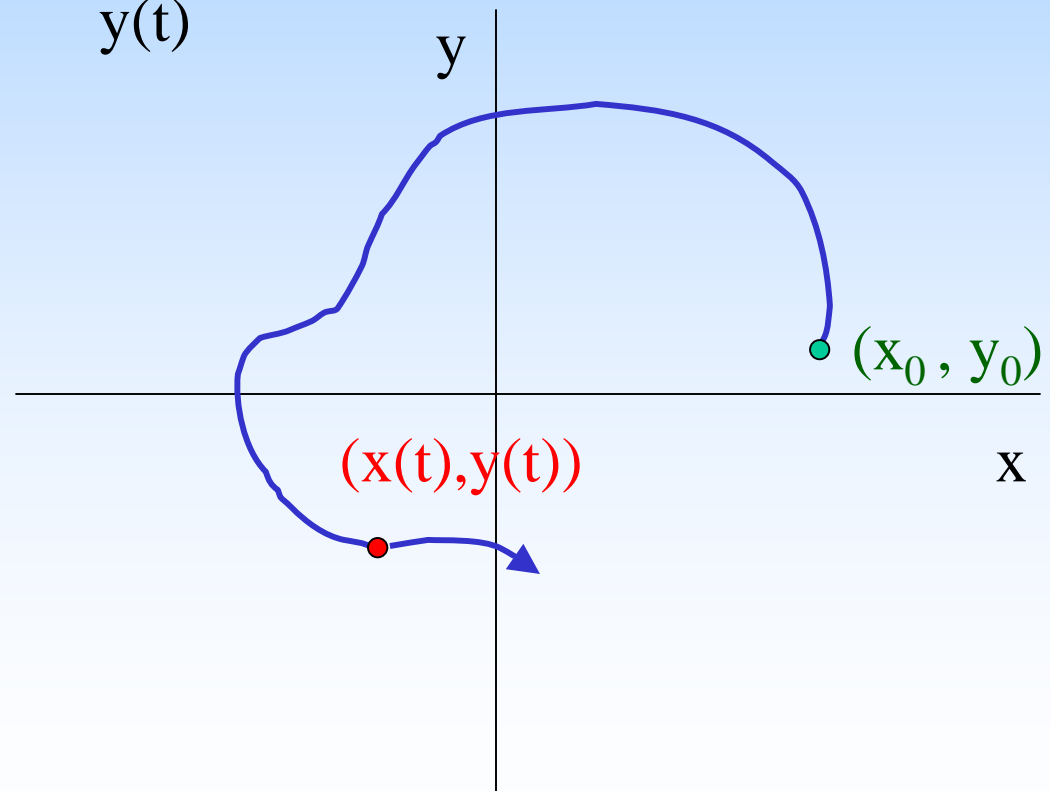
$$\left. \begin{aligned} \frac{dx}{dt} &= f_1(x,y) & x(0) &= x_0 \\ \frac{dy}{dt} &= f_2(x,y) & y(0) &= y_0 \end{aligned} \right\} \begin{array}{l} x(t) \\ y(t) \end{array}$$



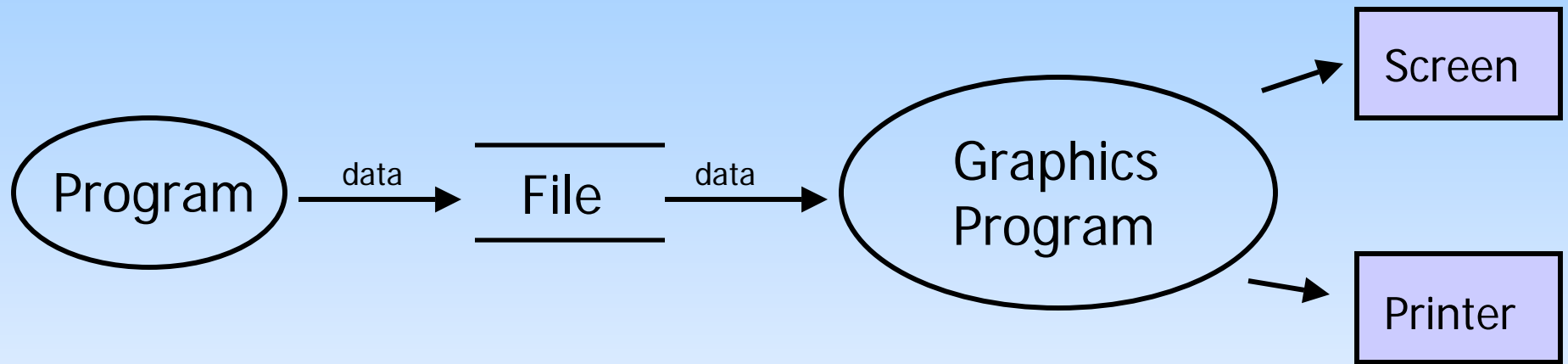
Phase plane plots

A parametric plot of $(x(t), y(t))$.

$$\left. \begin{array}{l} \frac{dx}{dt} = f_1(x, y) \quad x(0) = x_0 \\ \frac{dy}{dt} = f_2(x, y) \quad y(0) = y_0 \end{array} \right\} \begin{array}{l} x(t) \\ y(t) \end{array}$$

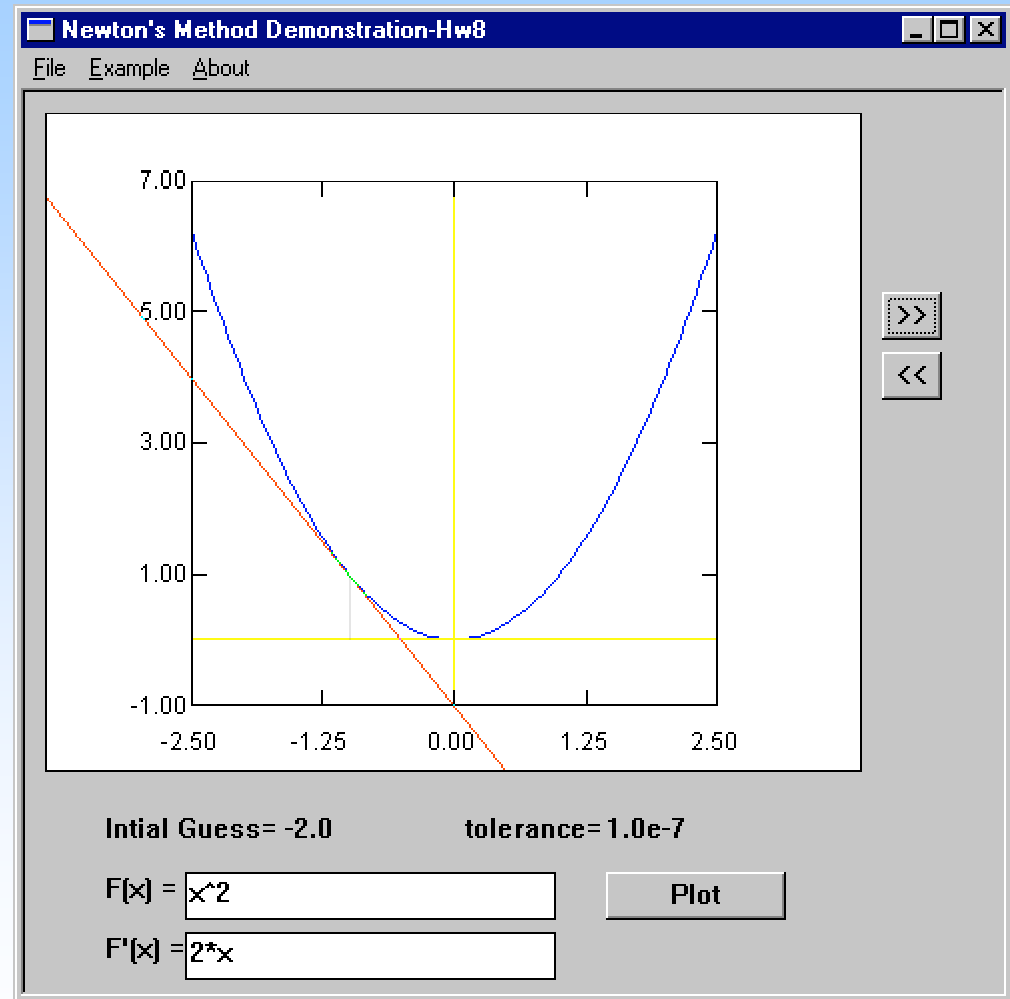


Use of external graphics packages -- GNUplot, Matlab, Excel, XMgrace, TekPlot, ...

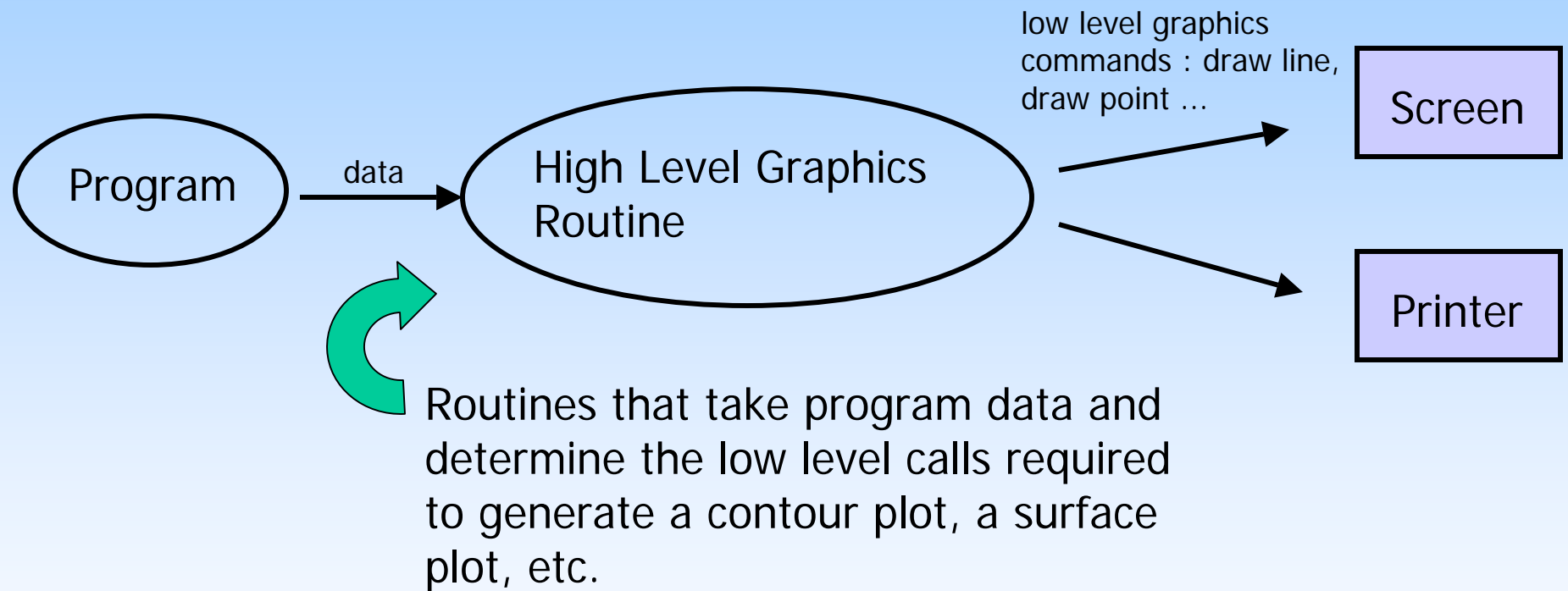


“Stand-alone” programs are not always sufficient

- Interactive applications
- Program development/debugging
- Custom graphics

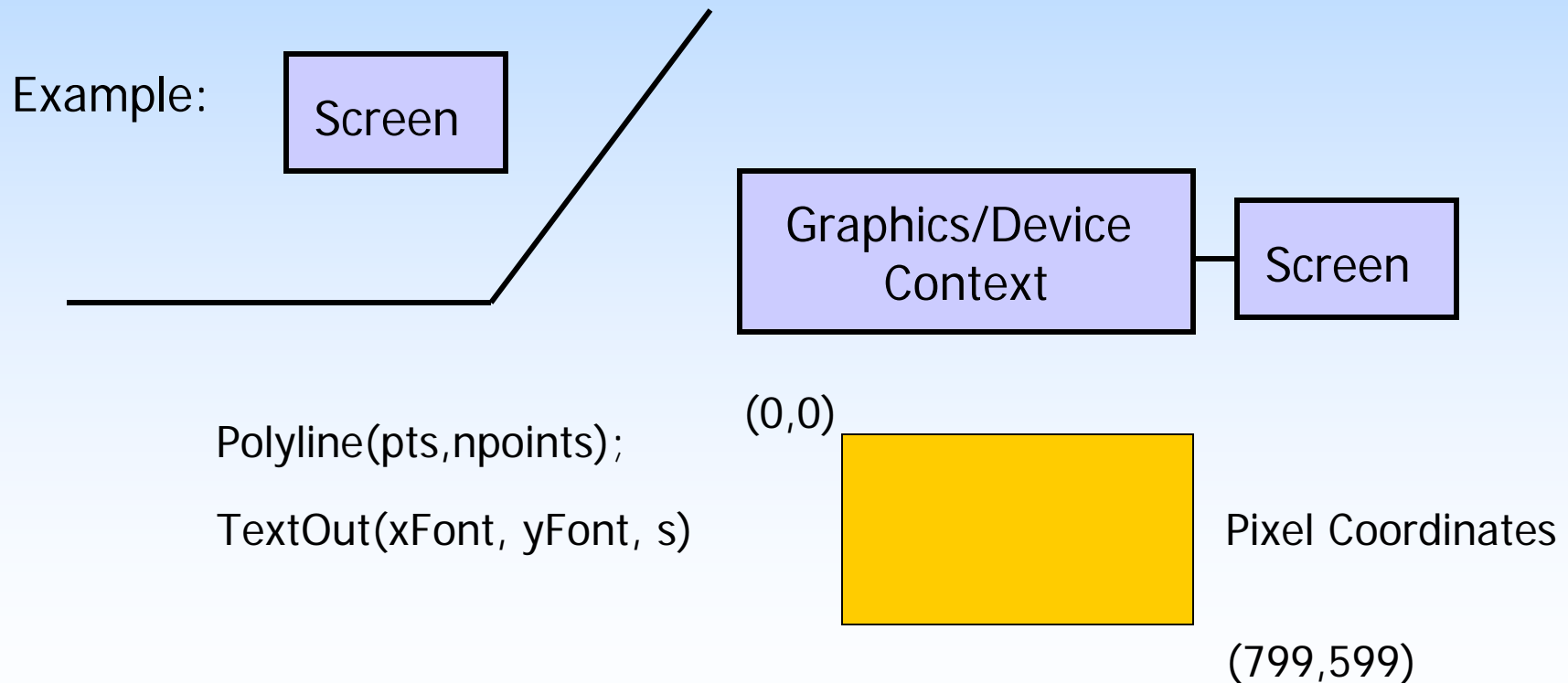


Incorporation of graphics within the program..



Fact about display devices:

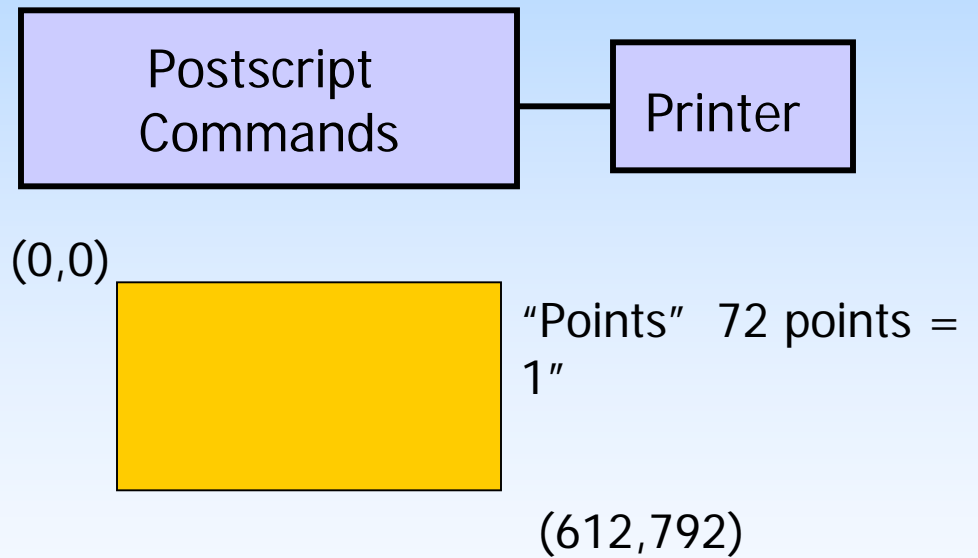
- Each display device has its own, typically unique, low level functions to output low level graphics
- Each display device has its own coordinate system.



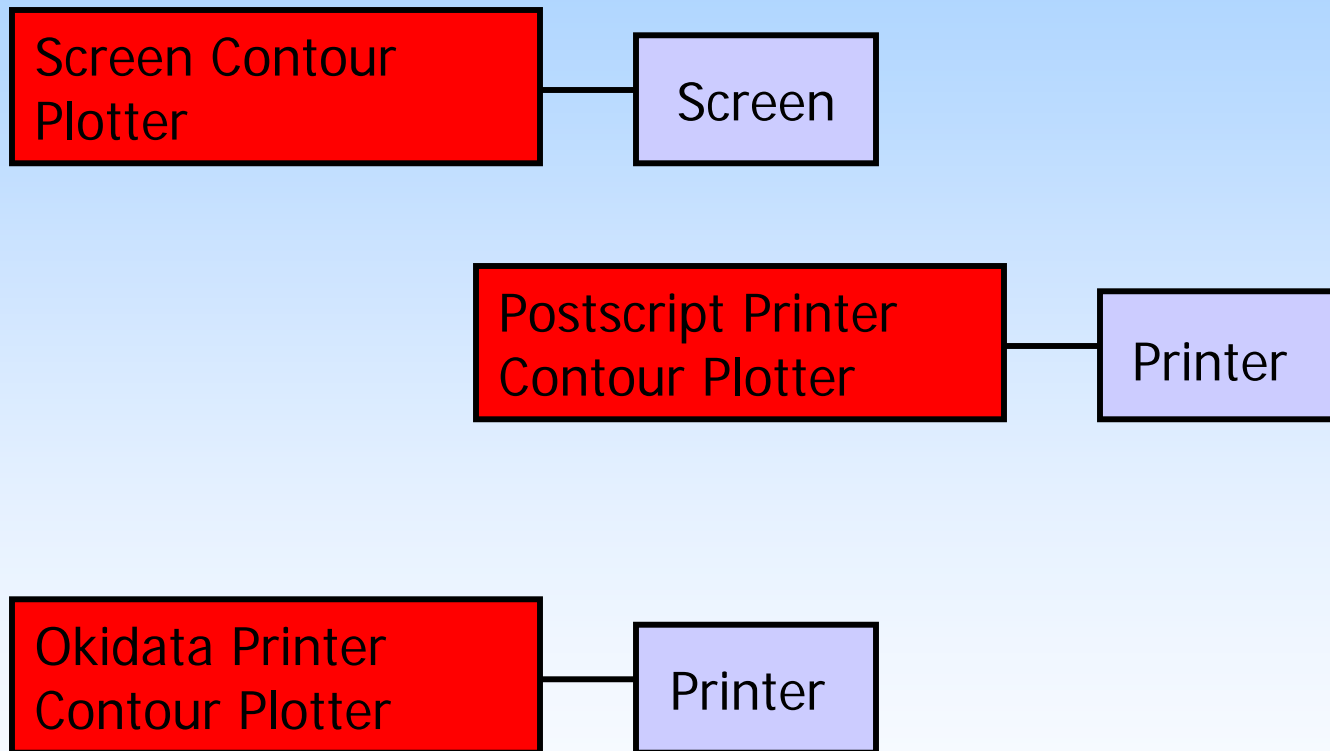
Display devices ...

Example : Printer

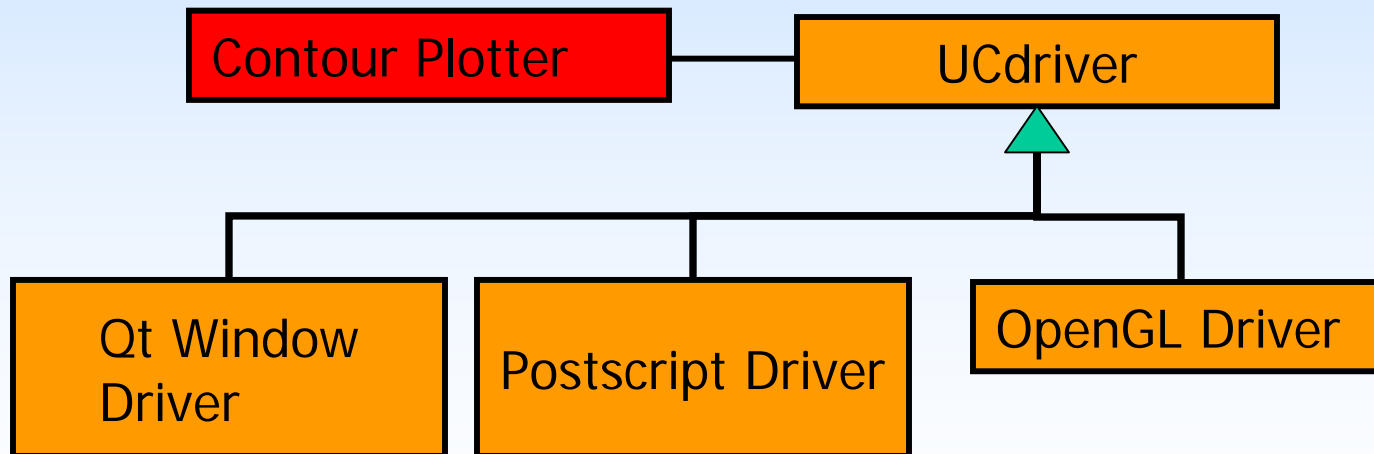
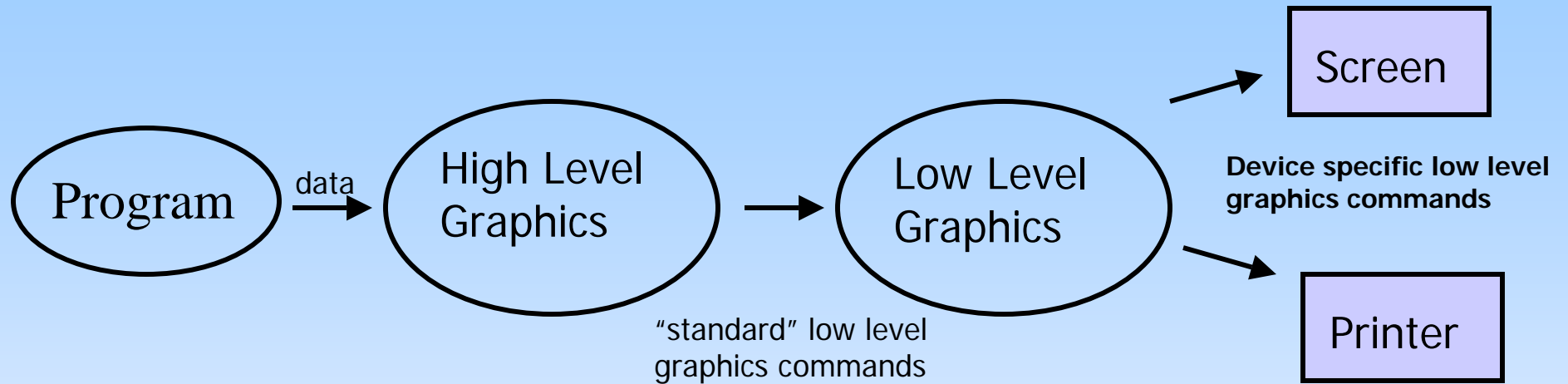
```
468 468 scale
newpath
0.3084 0.5187 moveto
0.307 0.5211 lineto
0.3073 0.5222 lineto
0.3084 0.5256 lineto
0.3131 0.5211 lineto
0.3096 0.5199 lineto
0.3084 0.5187 lineto
stroke
```



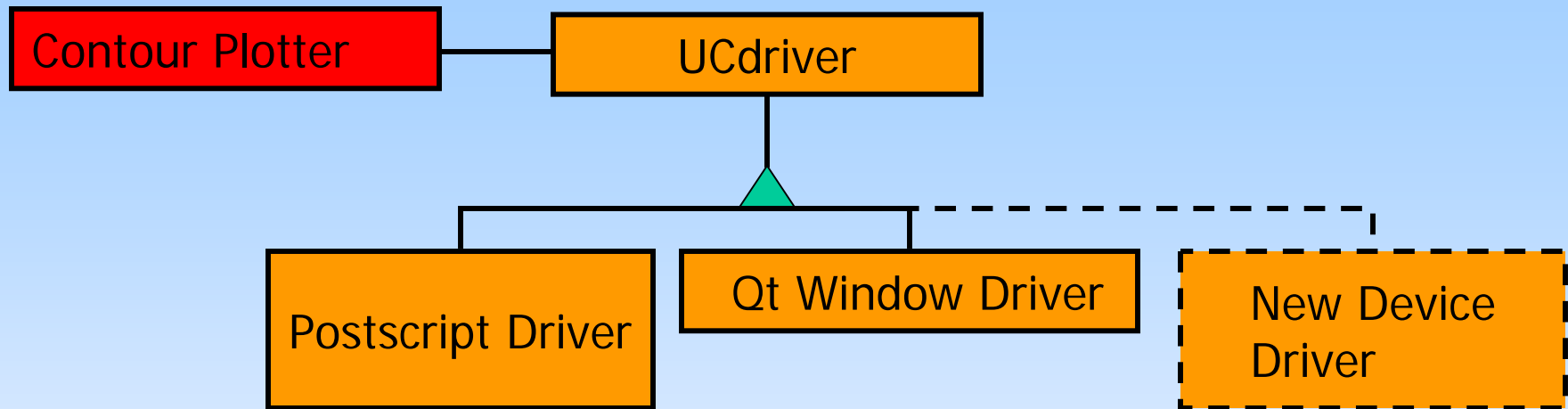
Problem : If you write high level routines that use specific device calls, then you need to have separate code for each device



Solution : Use an interface class.

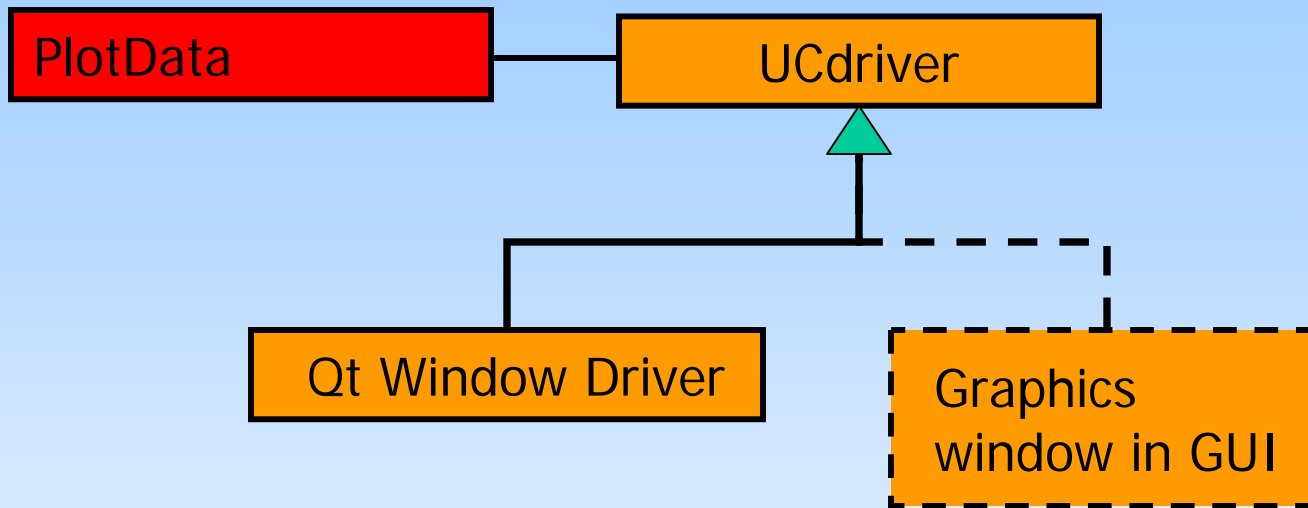


Interface class ...



- For any new device, just create a new class that translates UCdriver calls to the new device calls.
- High level graphics routines, e.g. the Contour Plotter class routines are unchanged.

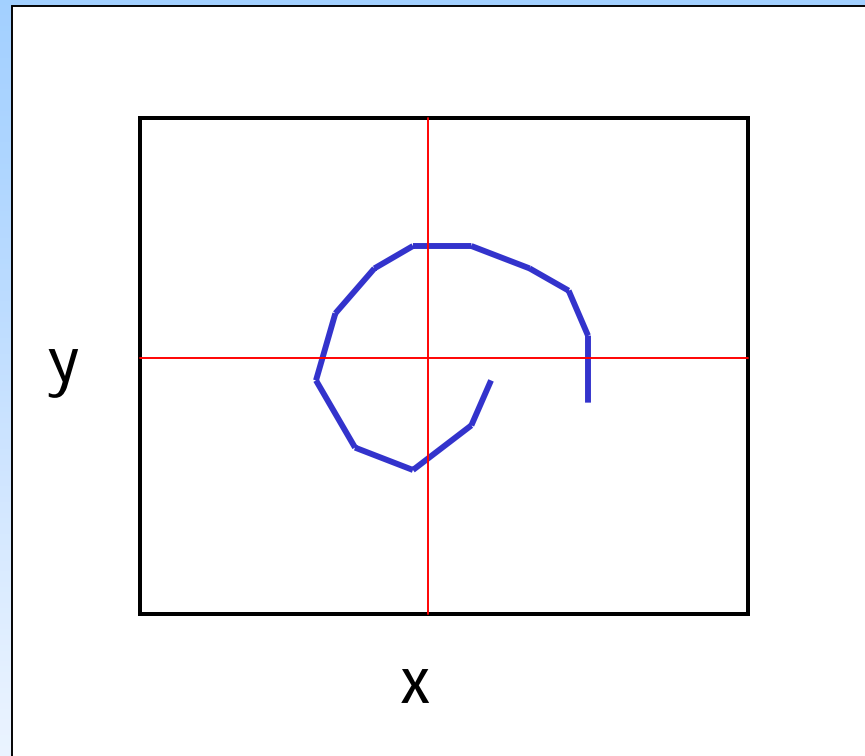
You will be creating a class providing plot routines that will display solutions of systems of ODE's



- Classes/code you create will work later within a GUI

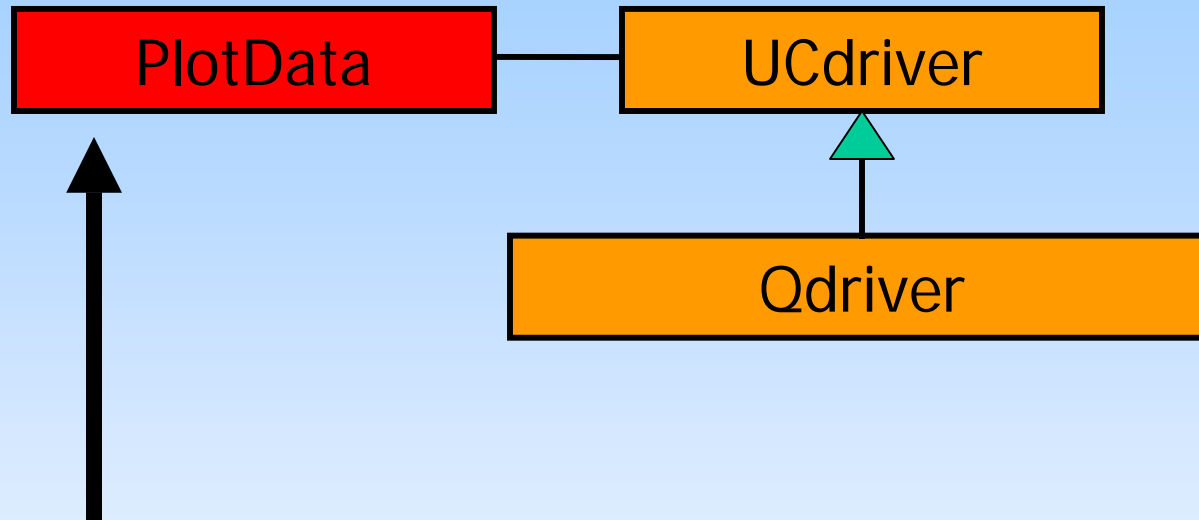
Assignment #6?

A phase plane plot of the numerical solution to the ODE.



- Need to be able to plot a collection of segments
- Data points are given in the ODE (e.g. problem) coordinate system.

Class Diagram



- translates the request for drawing segments in the problem coordinate system to requests for drawing a segment in the normalized coordinate system.
- annotates the plot (bounding box, axis etc)

Use ...

```
int main()
```

```
{
```

```
    *
```

```
    *
```

```
    Qdriver Qd;
```

```
    PlotData P;
```



Declare "low level" and "high level" classes

```
    P.setGraphicsDriver(&Qd);
```



Associate Qd with P

```
    *
```

```
    *
```

```
    P.plot(X,Y,xmin,xmax,ymin,ymax);
```

```
    P.drawBoundingBox();
```



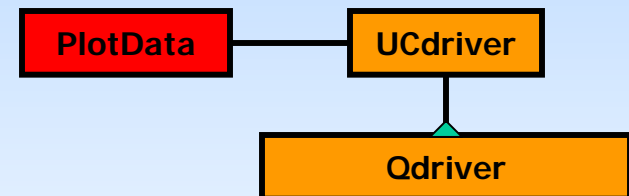
P's UCdriver calls will use Qd's implementation

```
    Qd.frame();
```



Need to tell the display that you are done plotting, so it can then display

```
}
```



What functionality must the PlotData class provide?

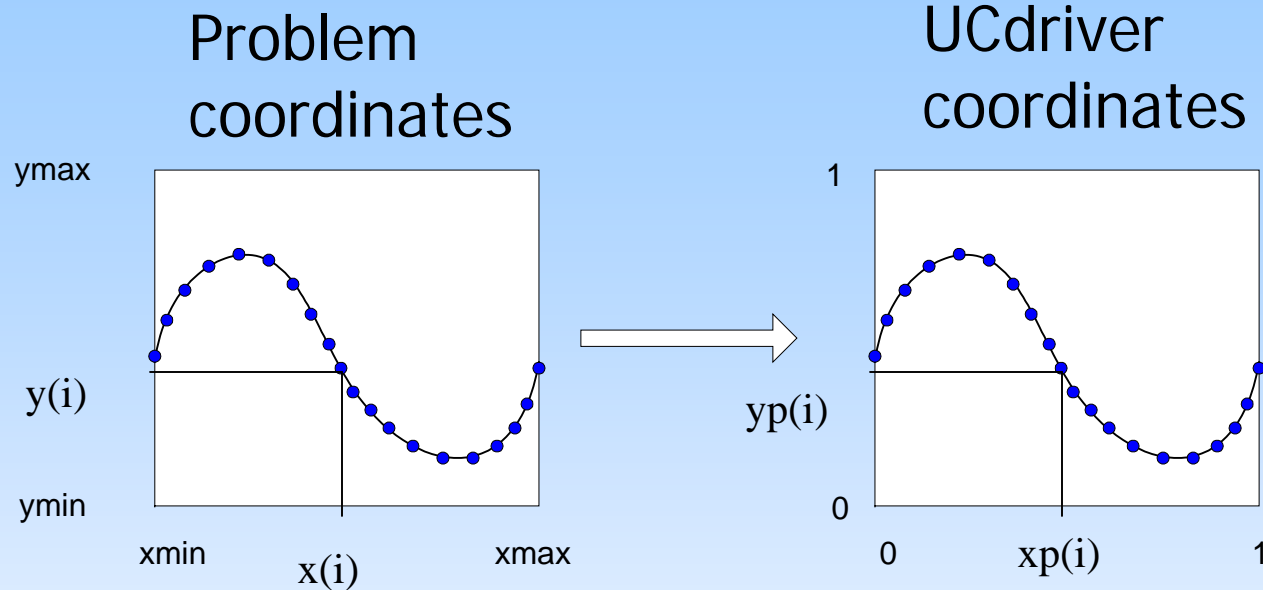
Public functionality

- Given a collection or list of pairs of data points, draw the segments between the data points
- Draw text or characters at specified locations (specified in either coordinate system)
- Draw plot annotations, at least a bounding box

Private functionality

- Perform scaling between the ODE coordinate system and the $[0,1] \times [0,1]$ coordinate system of the UCdriver

Primary PlotData task transform data



$$(x(i) - x_{min}) / (x_{max} - x_{min}) = (x_p(i) - 0.0) / (1.0 - 0.0)$$

$$(y(i) - y_{min}) / (y_{max} - y_{min}) = (y_p(i) - 0.0) / (1.0 - 0.0)$$