Numerical Analysis
Project 6
Ed MacDonald

For project 6 I was able to find a piece of software called "FontForge" (http://fontforge.sourceforge.net/), which is a tool for editing True Type fonts (.ttf files). Using this tool, I opened the "Times New Roman" font located on my Mac (/Library/Fonts/Times New Roman.ttf). Once I had the font loaded in the tool, I was presented with a wealth of information.

However, the steps required to get the font into a format where I could easily use it in my Matlab project were not trivial. The tool provides a "properties" window that will let you walk through the Bezier points one by one, but the sheer number of points made this too painful to attempt. After some experimentation, I found that the tool would export individual letters from the font to PostScript files. Using the information in the "Reality Check" on page 188 of the text, I was able to write a crude translator in Perl that would translate the PostScript files to Matlab function files (one for each letter: "g", "m", and "u").

Because I was able to use FontForge, I took on the added assignment of exploring a transform which would slant the letters. I hypothesized that for some constant, c, the transform:

x |--> x + y*c
y |--> y

where:
x, y are the vectors containing the x and y coordinates of the Bezier control and end points.

would slant the letters something similar to what you would expect in an italic font. I was pleased with the results (they looked as I had expected). While it may seem obvious that transforming the Bezier endpoints in such a manner would yield the results just described, I don't think it's as obvious that transforming the control points will yield these results. My experiment seems to indicate that the transform works for control and end points.

Following are descriptions for each of the attachments to this document.

- **convps**: This is the Perl script that translates PostScript files into Matlab m files.
- **gen**: This is a shell script that calls convps for each letter in "gmu". I included as an example of how to use convps.
- **g.m, m.m, u.m**: These are the m files generated by convps. They include the Bezier points for the letters.
- **bezierPlot.m**: This is the m file containing the code we were given to start the project. I modified it to calculate two widths for each character; the width with no scaling and the width with scaling. Depending on how you wish for spacing between the characters behave when scaling occurs, you would choose one of these over the other. In my plots, I used the width before scaling.

- **plotGMU.m**: This file calls bezierPlot for each of the letters and spaces them apart from each other. It is designed to be called in one of two ways. It can be called from another m file (command line style), or it can be called from a Matlab GUI callback function. In the latter case, it is heavily tied to the GUI I designed. This is because names of the widgets whose values it is looking for are hardcoded into the plotGMU.m file.
- **project6.m**: This file is supplied to run the project from the command line.

  Also attached to this document are four images. They are (in order):

- **"gmu"**: The letters "gmu" plotted with no Bezier points and no slanting.
- **"gmu" With Points**
- **"gmu" With Points, Slanted Left**
- **"gmu" With Points, Slanted Right**

Missing from this document are two files, project6gui.fig and project6gui.m. The former of these two files is the GUI definition file generated by GUIDE, Matlab's GUI creation tool. The latter is the m file generated by GUIDE that contains all of the initialization and callback functions for the GUI. Neither of these files is very interesting: the former is a binary file, the latter is very verbose and most of it is generated. It wasn't worth printing it out for the few lines in it that I changed. However, if you have a soft copy distribution of this project, they will both be included because they are required to start up the GUI!

If you do have a soft copy of this project, go into the project directory and type project6gui.m (or open the file of the same name in Matlab and click the "run" button). If you're running a later version of Matlab, you should see the GUI pop up. Enjoy!

```perl
#!/opt/local/bin/perl

$infile = shift or die "No input file specified.";
$matlabFunctionName = shift or die "No Matlab function name specified.";

open INFILE ,"<$infile" or die "$!";

my @fields = ();
my $process = 0;
my $command = "";
my $x = [];
my $y = [];
my $row = 0;

while(my $cl = <INFILE>){
   $cl =~ s/(^\s+)|(\s+$)//g;
   @fields = split /\s+/, $cl;

   $command = $fields[-1];

   if($command eq 'newpath'){
      $process = 1;
      next;
   }
   if($command eq 'grestore'){
      last;
   }

   if($process){
      if($command eq 'moveto'){
         $x->[$row]->[0] = $fields[0];
         $y->[$row]->[0] = $fields[1];
      }
      elsif($command eq 'lineto'){
         $x->[$row]->[1] = $x->[$row]->[0];
         $y->[$row]->[1] = $y->[$row]->[0];

         $x->[$row]->[2] = $fields[0];
         $x->[$row]->[3] = $fields[0];

         $y->[$row]->[2] = $fields[1];
         $y->[$row]->[3] = $fields[1];

         $row++;

         #initialize the next row
         $x->[$row]->[0] = $x->[$row - 1]->[3];
         $y->[$row]->[0] = $y->[$row - 1]->[3];
      }
      elsif($command eq 'curveto'){
         $x->[$row]->[1] = $fields[0];
         $y->[$row]->[1] = $fields[1];

         $x->[$row]->[2] = $fields[2];
         $y->[$row]->[2] = $fields[3];

         $x->[$row]->[3] = $fields[4];
         $y->[$row]->[3] = $fields[5];

         $row++;

         #initialize the next row
         $x->[$row]->[0] = $x->[$row - 1]->[3];
         $y->[$row]->[0] = $y->[$row - 1]->[3];
      }
   }
}

#Both arrays will have one extra row due to steps wich initialize
#the "next" row. Note, this assumes that processing doesn't end on a
#moveto.
pop @$x;
pop @$y;
```

```
print "function [x, y] = $matlabFunctionName ()\n";
print "x = [\n";
for(my $i=0; $i < scalar(@$x); $i++){
   print "           ";
   for(my $j=0; $j < 4; $j++){
      print $x->[$i]->[$j], " ";
   }
   print ";\n";
}
print "];\n";

print "y = [\n";
for(my $i=0; $i < scalar(@$x); $i++){
   print "           ";
   for(my $j=0; $j < 4; $j++){
      print $y->[$i]->[$j], " ";
   }
   print ";\n";
}
print "];\n";
print "end\n";
```

```
convps g/g_TimesNewRomanPSMT.eps g >> g.m
convps m/m_TimesNewRomanPSMT.eps m >> m.m
convps u/u_TimesNewRomanPSMT.eps u >> u.m
```

```
function [x, y] = g ()
x = [
          309 253 210 180 ;
          180 150 135 135 ;
          135 135 169.167 237.5 ;
          237.5 305.833 393.333 500 ;
          500 587.333 663 727 ;
          727 727 921 921 ;
          921 949.667 966.333 971 ;
          971 975.667 979 981 ;
          981 985 987 987 ;
          987 987 985.333 982 ;
          982 980 976.5 971.5 ;
          971.5 966.5 949.667 921 ;
          921 921 802 802 ;
          802 839.333 858 858 ;
          858 858 825.333 760 ;
          760 694.667 607 497 ;
          497 451.667 405.333 358 ;
          358 328.667 308.833 298.5 ;
          298.5 288.167 283 283 ;
          283 283 288.5 299.5 ;
          299.5 310.5 332 364 ;
          364 382.667 429.333 504 ;
          504 641.333 730.333 771 ;
          771 833 882.5 919.5 ;
          919.5 956.5 975 975 ;
          975 975 939.667 869 ;
          869 765 629.333 462 ;
          462 333.333 224.667 136 ;
          136 86 61 61 ;
          61 61 64.6667 72 ;
          72 83.3333 106.667 142 ;
          142 146.667 180.667 244 ;
          244 209.333 184.833 170.5 ;
          170.5 156.167 149 149 ;
          149 149 158.5 177.5 ;
          177.5 196.5 240.333 309 ;
          483 433.667 392.333 359 ;
          359 325.667 309 309 ;
          309 309 331.667 377 ;
          377 411.667 455.667 509 ;
          509 559.667 601.333 634 ;
          634 666.667 683 683 ;
          683 683 660 614 ;
          614 580 536.333 483 ;
          299 267.667 244 228 ;
          228 212 204 204 ;
          204 204 223.333 262 ;
          262 328.667 425 551 ;
          551 671 759.5 816.5 ;
          816.5 873.5 902 902 ;
          902 902 885 851 ;
          851 816.333 747.667 645 ;
          645 495 379.667 299 ;
];
y = [
          334 361.333 399.5 448.5 ;
          448.5 497.5 551.667 611 ;
          611 701.667 779.667 845 ;
          845 910.333 943 943 ;
          943 943 921.667 879 ;
          879 879 879 879 ;
          879 879 878.167 876.5 ;
          876.5 874.833 872 868 ;
          868 862 851.333 836 ;
          836 818.667 806.667 800 ;
          800 796.667 794 792 ;
          792 790 789 789 ;
          789 789 789 789 ;
          789 741 679.667 605 ;
          605 519.667 446.667 386 ;
          386 325.333 295 295 ;
          295 295 301.667 315 ;
          315 289.667 267.5 248.5 ;
          248.5 229.5 213.333 200 ;
          200 188.667 177.667 167 ;
          167 156.333 148.667 144 ;
          144 141.333 139 137 ;
          137 133.667 129 123 ;
          123 114.333 91.3333 54 ;
          54 16.6667 -29.3333 -84 ;
          -84 -159.333 -230 -296 ;
          -296 -393.333 -442 -442 ;
          -442 -442 -413 -355 ;
          -355 -321.667 -287 -251 ;
          -251 -235 -219 -203 ;
          -203 -178.333 -144 -100 ;
          -100 -94 -58 8 ;
          8 28.6667 47.1667 63.5 ;
          63.5 79.8333 98.3333 119 ;
          119 142.333 169.667 201 ;
          201 232.333 276.667 334 ;
          895 895 875.333 836 ;
          836 796.667 736.333 655 ;
          655 549.667 468 410 ;
          410 366 344 344 ;
          344 344 363 401 ;
          401 439 498.667 580 ;
          580 686 769 829 ;
          829 873 895 895 ;
          0 -34 -65.6667 -95 ;
          -95 -124.333 -151.333 -176 ;
          -176 -208 -236 -260 ;
          -260 -301.333 -322 -322 ;
          -322 -322 -300.833 -258.5 ;
          -258.5 -216.167 -171 -123 ;
          -123 -88.3333 -63.6667 -49 ;
          -49 -34.3333 -25.6667 -23 ;
          -23 -19 -11.3333 0 ;
];
end
```

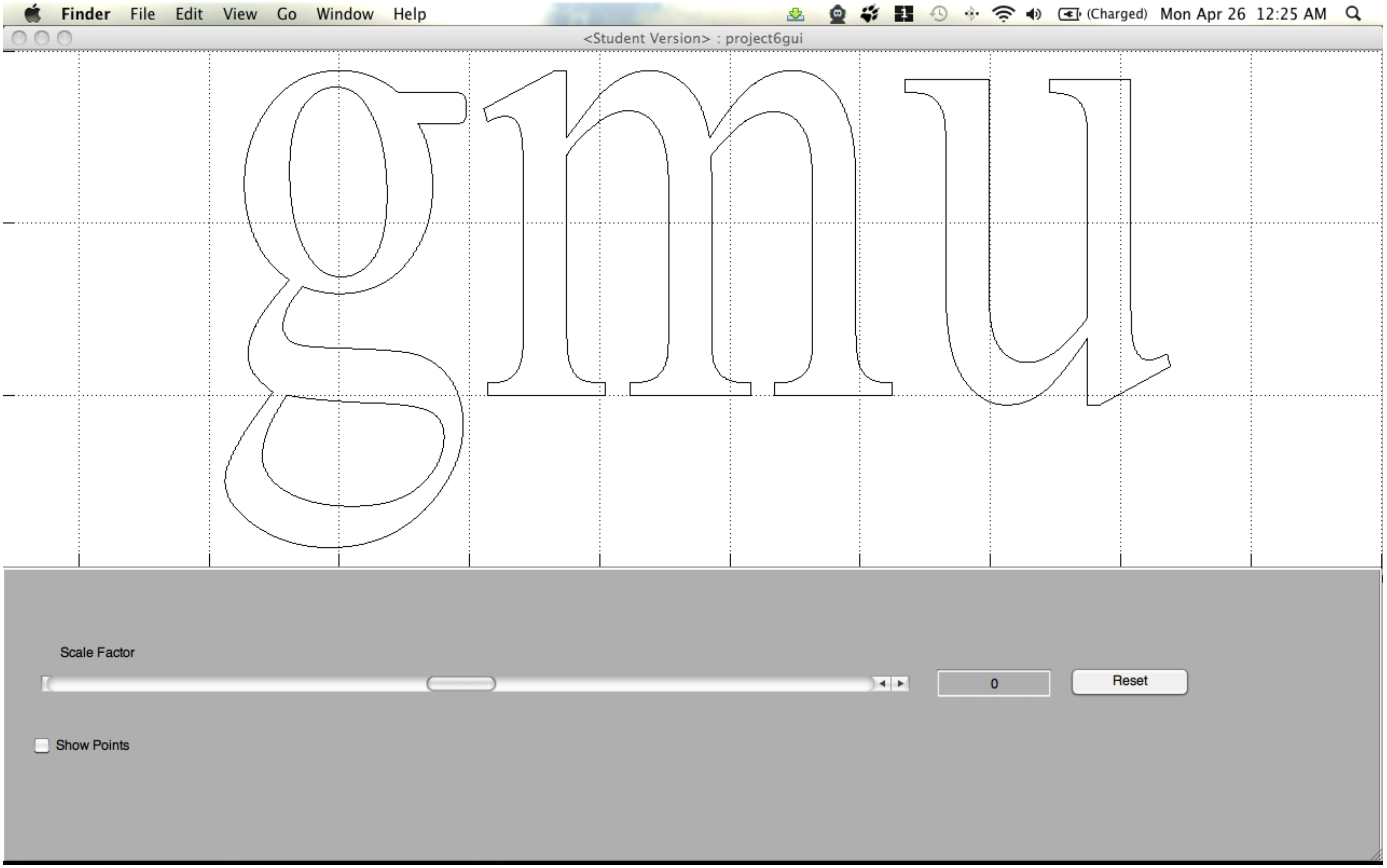
```
function [x, y] = m ()
x = [
          336 402.667 442 454 ;
          454 484 516.333 551 ;
          551 585.667 620 654 ;
          654 711.333 760.667 802 ;
          802 843.333 871 885 ;
          885 953.667 1011.67 1059 ;
          1059 1106.33 1155 1205 ;
          1205 1253.67 1296.83 1334.5 ;
          1334.5 1372.17 1402 1424 ;
          1424 1438.67 1446 1446 ;
          1446 1446 1446 1446 ;
          1446 1446 1450.33 1459 ;
          1459 1465.67 1478 1496 ;
          1496 1514 1543.33 1584 ;
          1584 1584 1584 1584 ;
          1584 1584 1132 1132 ;
          1132 1132 1132 1132 ;
          1132 1132 1151 1151 ;
          1151 1190.33 1221 1243 ;
          1243 1258.33 1269.33 1276 ;
          1276 1278.67 1280 1280 ;
          1280 1280 1280 1280 ;
          1280 1280 1271 1253 ;
          1253 1227 1185.33 1128 ;
          1128 1092.67 1057.17 1021.5 ;
          1021.5 985.833 942.667 892 ;
          892 892 890 890 ;
          890 890 892 892 ;
          892 892 892 892 ;
          892 892 895.5 902.5 ;
          902.5 909.5 922.667 942 ;
          942 961.333 994.333 1041 ;
          1041 1041 1041 1041 ;
          1041 1041 578 578 ;
          578 578 578 578 ;
          578 628.667 663.5 682.5 ;
          682.5 701.5 714.667 722 ;
          722 725.333 727 727 ;
          727 727 727 727 ;
          727 727 716 694 ;
          694 664.667 623.667 571 ;
          571 535 499.333 464 ;
          464 408.667 366 336 ;
          336 336 336 336 ;
          336 336 340.167 348.5 ;
          348.5 356.833 369.167 385.5 ;
          385.5 401.833 435 485 ;
          485 485 485 485 ;
          485 485 32 32 ;
          32 32 32 32 ;
          32 74 103.333 120 ;
          120 136.667 149.333 158 ;
          158 166.667 171 171 ;
          171 171 171 171 ;
          171 171 168 162 ;
          162 157.333 150 140 ;
          140 130 116.333 99 ;
          99 80.3333 58 32 ;
          32 32 17 17 ;
          17 17 293 293 ;
          293 293 336 336 ;
          336 336 336 336 ;
];
y = [
          748 814.667 853 863 ;
          863 888.333 908 922 ;
          922 936 943 943 ;
          943 943 926.333 893 ;
          893 859.667 811.333 748 ;
          748 828 880.5 905.5 ;
          905.5 930.5 943 943 ;
          943 943 930.5 905.5 ;
          905.5 880.5 839.667 783 ;
          783 744.333 683.667 601 ;
          601 601 207 207 ;
          207 149.667 110.333 89 ;
          89 74.3333 61.8333 51.5 ;
          51.5 41.1667 36 36 ;
          36 36 0 0 ;
          0 0 0 0 ;
          0 0 36 36 ;
          36 36 36 36 ;
          36 36 43.6667 59 ;
          59 69.6667 86.6667 110 ;
          110 121.333 153.667 207 ;
          207 207 601 601 ;
          601 675.667 728.333 759 ;
          759 801.667 823 823 ;
          823 823 814.167 796.5 ;
          796.5 778.833 746 698 ;
          698 698 687 687 ;
          687 687 644 644 ;
          644 644 207 207 ;
          207 144.333 105.333 90 ;
          90 74.6667 61.8333 51.5 ;
          51.5 41.1667 36 36 ;
          36 36 0 0 ;
          0 0 0 0 ;
          0 0 36 36 ;
          36 36 42 54 ;
          54 66 84 108 ;
          108 119.333 152.333 207 ;
          207 207 601 601 ;
          601 675.667 729.333 762 ;
          762 804.667 826 826 ;
          826 826 816.333 797 ;
          797 767.667 734.667 698 ;
          698 698 207 207 ;
          207 147 108 90 ;
          90 72 58.5 49.5 ;
          49.5 40.5 36 36 ;
          36 36 0 0 ;
          0 0 0 0 ;
          0 0 36 36 ;
          36 36 40.5 49.5 ;
          49.5 58.5 72.8333 92.5 ;
          92.5 112.167 150.333 207 ;
          207 207 557 557 ;
          557 657.667 722.667 752 ;
          752 774 789.167 797.5 ;
          797.5 805.833 810 810 ;
          810 810 805 795 ;
          795 795 831 831 ;
          831 831 943 943 ;
          943 943 943 943 ;
          943 943 748 748 ;
];
end
```

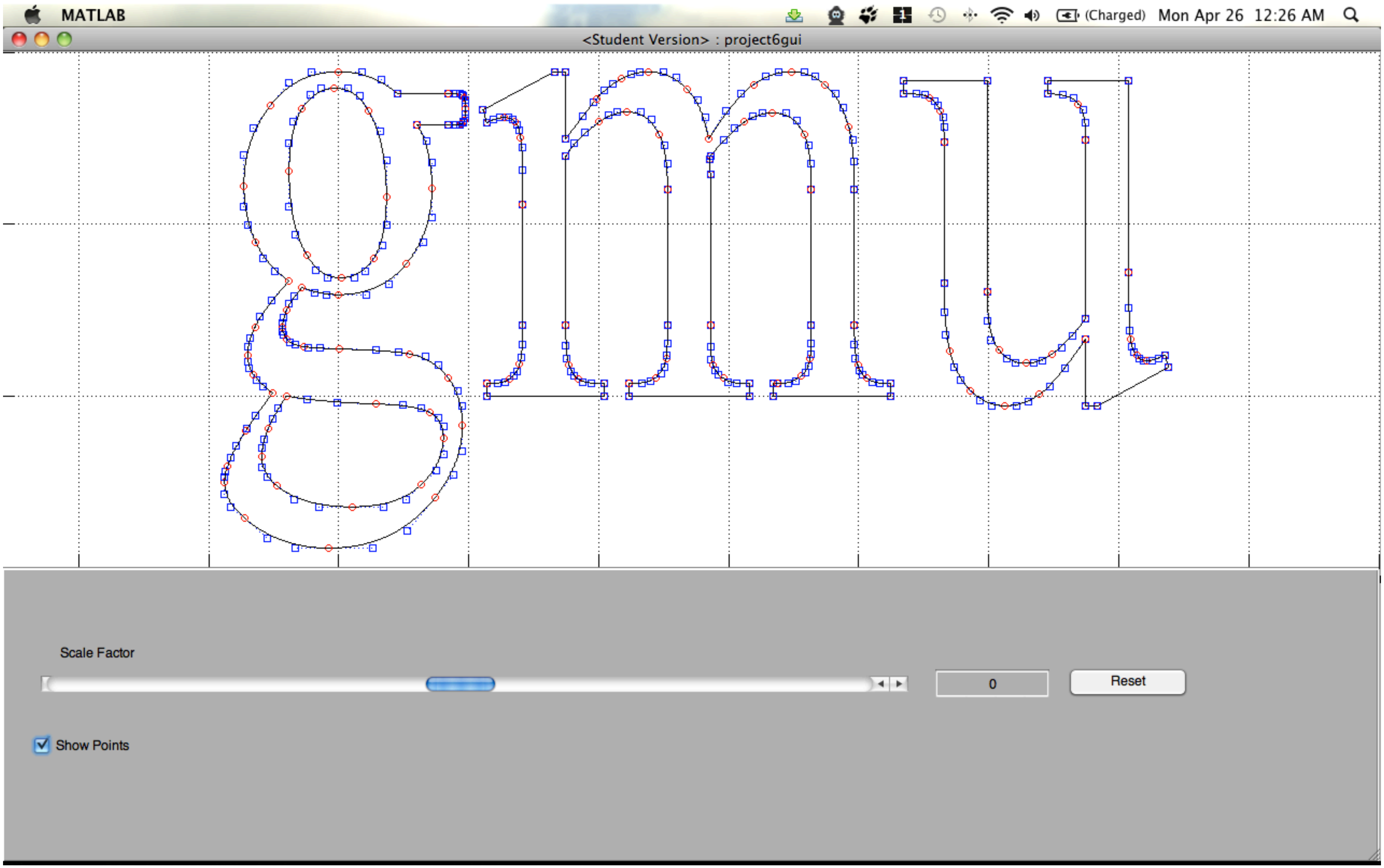
```
function [x, y] = u ()
x = [
          867 867 867 867 ;
          867 867 869.5 874.5 ;
          874.5 879.5 887.5 898.5 ;
          898.5 909.5 922.333 937 ;
          937 957.667 981 1007 ;
          1007 1007 1021 1021 ;
          1021 1021 747 747 ;
          747 747 702 702 ;
          702 702 702 702 ;
          702 623.333 563.333 522 ;
          522 480.667 437 391 ;
          391 339.667 295.167 257.5 ;
          257.5 219.833 193.667 179 ;
          179 164.333 157 157 ;
          157 157 157 157 ;
          157 157 152.333 143 ;
          143 133.667 119.833 101.5 ;
          101.5 83.1667 50 2 ;
          2 2 2 2 ;
          2 2 323 323 ;
          323 323 323 323 ;
          323 323 337.833 367.5 ;
          367.5 397.167 433 475 ;
          475 503.667 536.167 572.5 ;
          572.5 608.833 652 702 ;
          702 702 702 702 ;
          702 702 692.5 673.5 ;
          673.5 654.5 615 555 ;
          555 555 555 555 ;
          555 555 867 867 ;
];
y = [
          916 916 361 361 ;
          361 255 190.167 166.5 ;
          166.5 142.833 126.333 117 ;
          117 107.667 103 103 ;
          103 103 108.667 120 ;
          120 120 85 85 ;
          85 85 -28 -28 ;
          -28 -28 -28 -28 ;
          -28 -28 166 166 ;
          166 80.6667 27 5 ;
          5 -17 -28 -28 ;
          -28 -28 -13.1667 16.5 ;
          16.5 46.1667 84.3333 131 ;
          131 177.667 243.667 329 ;
          329 329 738 738 ;
          738 781.333 811.333 828 ;
          828 844.667 857.5 866.5 ;
          866.5 875.5 879.667 879 ;
          879 879 916 916 ;
          916 916 916 916 ;
          916 916 303 303 ;
          303 217.667 161.667 135 ;
          135 108.333 95 95 ;
          95 95 104 122 ;
          122 140 174.333 225 ;
          225 225 744 744 ;
          744 796 831.167 849.5 ;
          849.5 867.833 877.667 879 ;
          879 879 916 916 ;
          916 916 916 916 ;
];
end
```

```
function [ maxX, preScaleMaxX ] = bezierPlot( axesHandle, x, y, c, showPoints )
%bezierPlot
%
    numcurves=size(x,1);

    t=0:.002:1;

    maxX = 0;
    preScaleMaxX = 0;

    x = x + y*c;

    hold(axesHandle, 'on');

    for i=1:numcurves
      if showPoints
          plot(axesHandle, x(i,1),y(i,1),'ro',x(i,4),y(i,4),'ro')
          plot(axesHandle, [x(i,1),x(i,2)],[y(i,1),y(i,2)],'b:')
          plot(axesHandle, [x(i,3),x(i,4)],[y(i,3),y(i,4)],'b:')
          plot(axesHandle, x(i,2),y(i,2),'bs',x(i,3),y(i,3),'bs')
      end

      bx = 3*(x(i,2) - x(i,1));
      cx = 3*(x(i,3) - x(i,2)) - bx;
      dx = x(i,4) - x(i,1) - bx -cx;

      by = 3*(y(i,2) - y(i,1));
      cy = 3*(y(i,3) - y(i,2)) - by;
      dy = y(i,4) - y(i,1) - by -cy;

      xp=x(i,1)+bx*t+cx*t.*t+dx*t.^3;
      yp=y(i,1)+by*t+cy*t.*t+dy*t.^3;
      plot(axesHandle, xp,yp,'k')

      if max(xp) > maxX
          maxX = max(xp);
      end

      if( max(xp - yp*c) > preScaleMaxX)
         preScaleMaxX = max(xp - yp*c);
      end
    end
    %axis square
    grid on
    hold(axesHandle, 'off')
end
```

```
function plotGMU(varargin)
%plotGMU
%

    if nargin == 1
        %If nargin == 1, I'm assuming the only argument is the handles
        %object from project6gui. Yeah, this makes for a dependency on the
        %gui, but it will work without it too.
        handles    = varargin{1};
        hObject    = handles.axes1;
        spacing    = 50; %may want to create a gui object to vary this
        c          = get(handles.slider2, 'Value');
        if(get(handles.checkbox1, 'Value') == get(handles.checkbox1, 'Max'))
            showPoints = 1;
        else
            showPoints = 0;
        end
    else
        hObject       = varargin{1};
        spacing       = varargin{2};
        c             = varargin{3};
        showPoints    = varargin{4};
    end

    cla(hObject);

    axis([-800 4500 -500 1000]);

    hold on;

    [gx,gy] = g;
    [mx,my] = m;
    [ux,uy] = u;

    [maxGx, preScaleMaxGx] = ...
        bezierPlot(hObject, gx, gy, c, showPoints);

    [maxMx, preScaleMaxMx] = ...
        bezierPlot( hObject, ...
                    mx + preScaleMaxGx + spacing, ...
                    my, c, showPoints);

    [maxUx, preScaleMaxUx] = ...
        bezierPlot( hObject, ...
                    ux + preScaleMaxMx + spacing, ...
                    uy, c, showPoints);

    hold off;
end
```

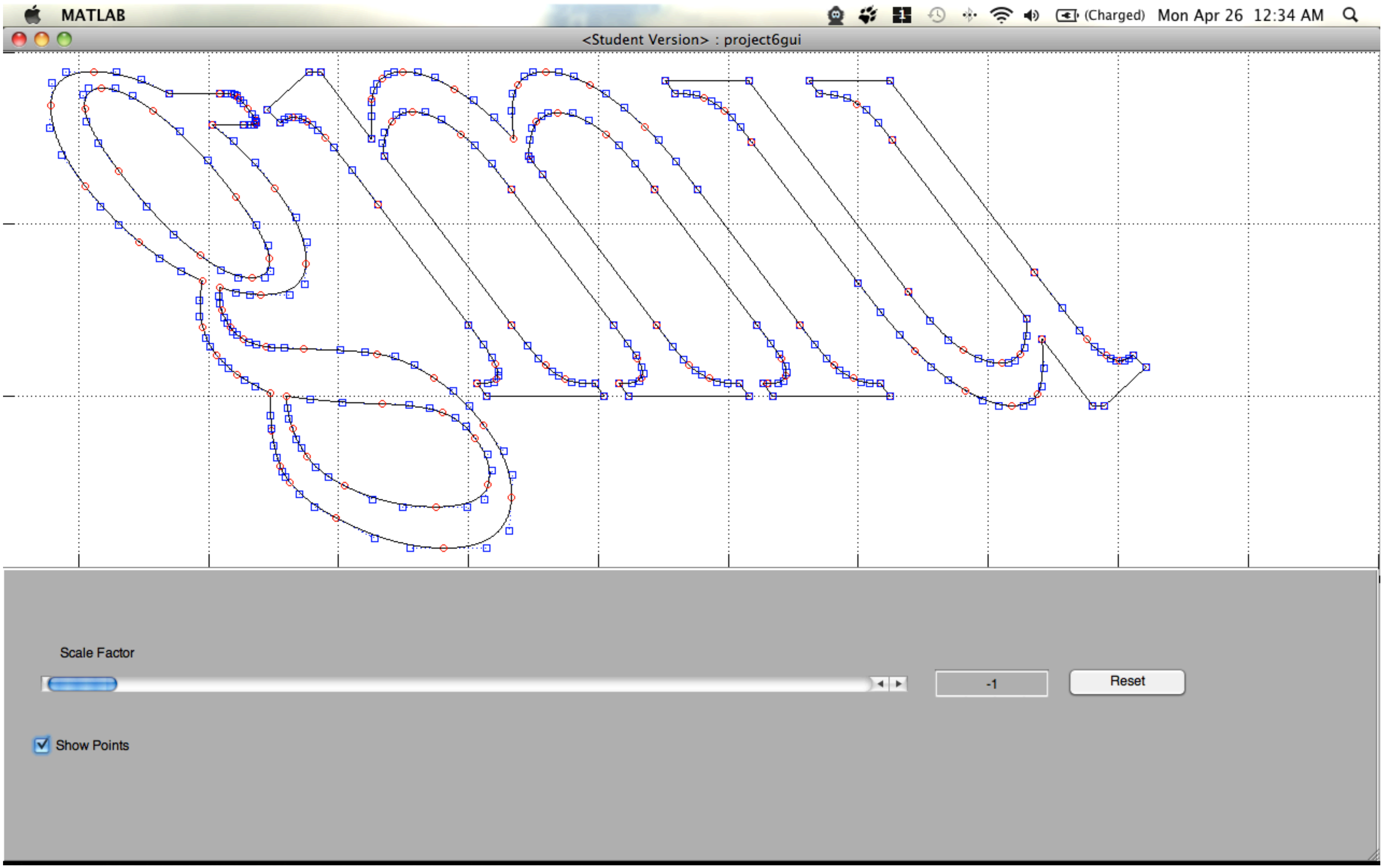
```
clc;
clear all;
clf;

hold on;

plotGMU(gca, 50, 0, 0);

hold off;
```

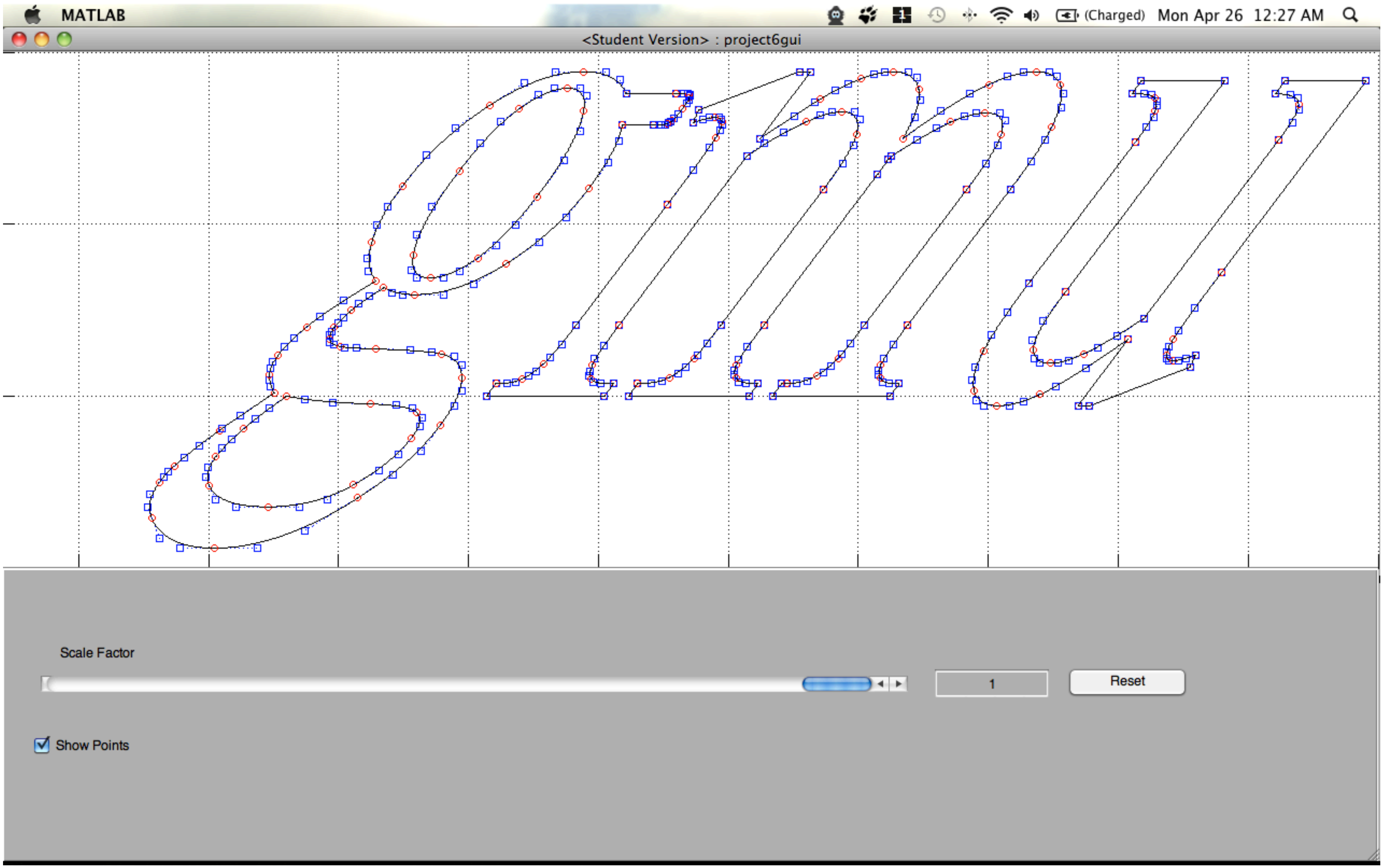

Finder File Edit View Go Window Help
(Charged) Mon Apr 26 12:25 AM
<Student Version> : project6gui
Scale Factor
0
Reset
Show Points


MATLAB
(Charged) Mon Apr 26 12:26 AM
<Student Version> : project6gui
Reset

Scale Factor 0

[✓] Show Points


MATLAB
(Charged) Mon Apr 26 12:34 AM
<Student Version> : project6gui
Scale Factor
-1
Reset
Show Points


MATLAB
Mon Apr 26 12:27 AM
(Charged)
<Student Version> : project6gui
Scale Factor
1
Reset
Show Points