MATH46052|66052 Computational Exercises 2

1. To visualise bilinear and bicubic spline approximation on a two-dimensional structured grid of data points, create an M-file `grid2D.m` containing the following command sequence. Then run the M-file (by typing `grid2D` in the command window).

   ```
   [x,y]=meshgrid(0:0.2:1,0:0.2:1); % set up 36 sample points
   Z=sin(2*pi*x).*cos(2*pi*y); % evaluate 2D function at points
   figure(1) % data plot
   plot3(x,y,Z,’ob’), axis square
   title(’raw function data’), shg, pause(3)
   xi=[0:0.05:1]; yi=[0:0.05:1]; % refined subdivision in x & y
   [xf,yf]=meshgrid(xi,yi); % 21x21 interpolation point values
   s1=interp2(x,y,Z,xf,yf); % linear interpolant in x & y
   figure(2)
   mesh(xf,yf,s1), axis square; hold on;
   plot3(x,y,Z,’ob’);
   title(’... and the bilinear spline approximation’), pause(3)
   figure(3)
   s3=interp2(x,y,Z,xf,yf,’cubic’); % cubic interpolant in x & y
   mesh(xf,yf,s3), axis square; hold on;
   plot3(x,y,Z,’ob’);
   title(’... and the cubic spline approximation’), shg
   ```

2. To visualise a linear spline approximation on a scattered two-dimensional dataset create an M-file `scatter2D.m` containing the following commands. Then run the M-file (by typing `scatter2D` in the command window).

   ```
   x=rand(100,1)*4-2; y=rand(100,1)*4-2; % 100 random points
   z = x.*exp(-x.^2-y.^2); % compute function values
   figure(1) % data plot
   plot3(x,y,z,’ob’), axis square
   title(’raw function data’), shg, pause(3)
   figure(2) % triangulation picture
   dt = DelaunayTri(x,y); % triangulate interpolation points
   triplot(dt); hold on, plot(x,y,’ro’), axis square,
   title(’triangulation’), shg, pause(3)
   figure(3)
   xi=[-2:0.2:2]; yi=[-2:0.2:2]; % refined subdivision in x & y
   [xf,yf]=meshgrid(xi,yi); % 21x21 interpolation point values
   F = TriScatteredInterp(x,y,z,’linear’);
   p1 = F(xf,yf); %  piecewise linear interpolant
   mesh(xf,yf,p1), axis square, hold on, plot3(x,y,z,’ob’),
   title(’... and the linear spline approximation’), shg
   ```

   Note that simply typing `close all, scatter2D` will repeat the experiment with a different set of random points!