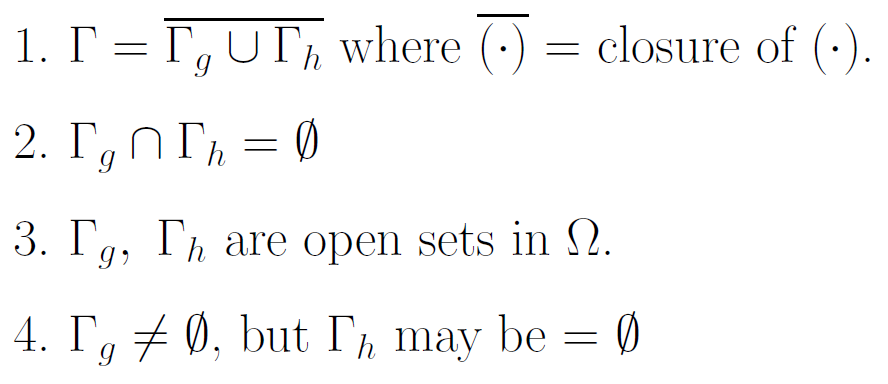
For steady-state (time-independent) heat diffusion the equations are

*qi,i = f* in Ω

*u = g* on Γ*g*

*-qini = h* on Γ*h*

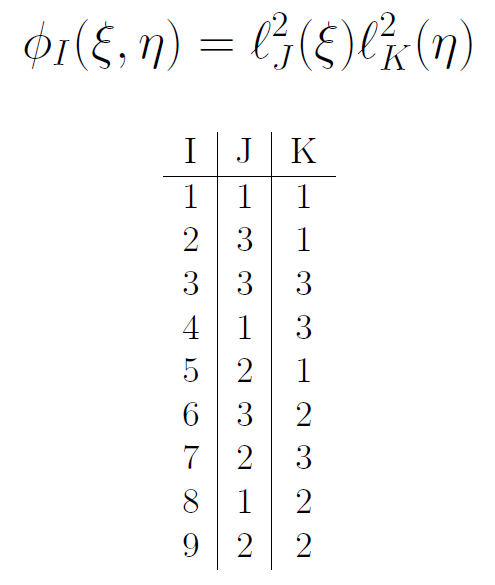
where *qi = -κiju,j*. First start by analyzing *elemstiff.m* where D should be replaced by a diagonal matrix assuming that our material is isotropic. Then in *applybcs.m* we have to make sure that with the total boundary being Γ, the boundary conditions Γ*g* and Γ*h* satisfy



are satisfied. Finally in *heat2d.m*, since we cannot assume that boundary conditions are homogeneous, if row *m* of the forcing vector lies on the Dirichlet boundary, we do not set row *m* of **bigk** to 0 but we still set column *m* of **bigk** to 0 and **bigk**(*m*,*m*) to 1.

The 9-node biquadratic elements consists of 9 shape functions that satisfies the Kronecker delta property for the local shape function number and the local node number. This is formed by taking the product of the 1D 3-node quadratic element in ξ and η. This still satisfies C1 element continuity, C0 global continuity, and linear precision so we can expect convergence of the resulting approximation. The reason why we may want to use biquadratic elements is that we can generally achieve higher accuracy than just the bilinear elements.

For the 9-node biquadratic elements start by analyzing at the original *elemstiff.m* file. Based on Lecture 13, there would be 9 nodes instead of 4 that would follow the ordering

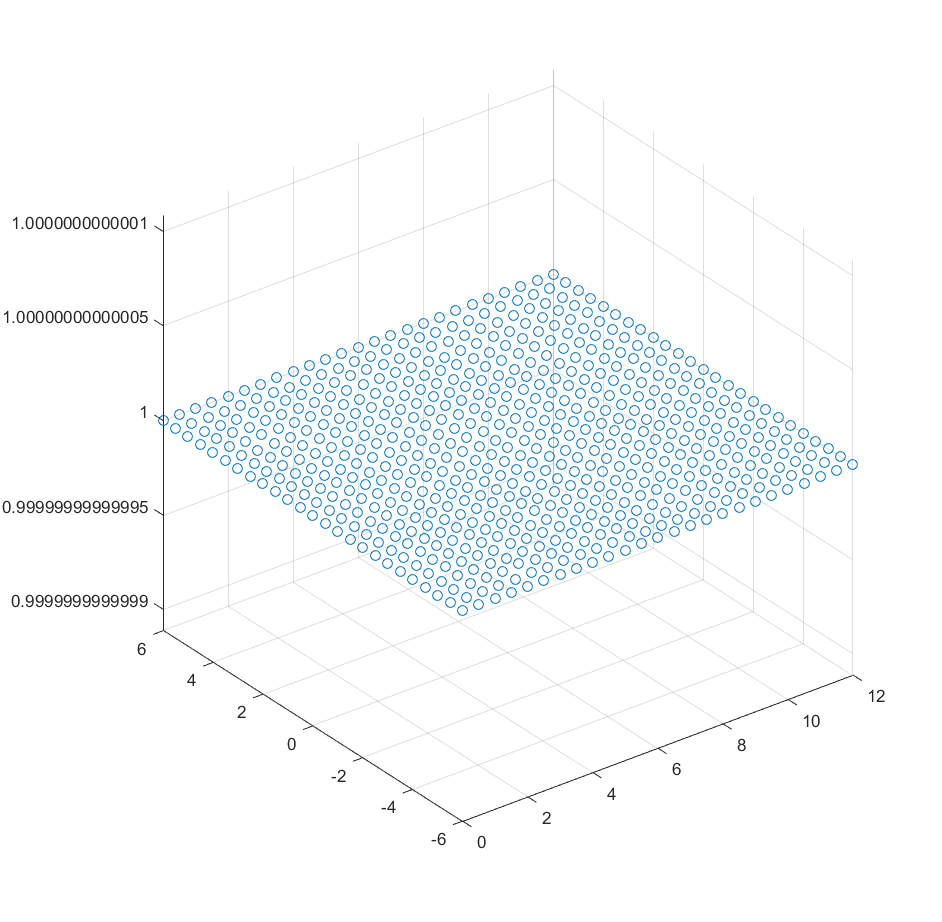


so begin by writing down the 9 shape functions for each node of the element in terms of ξ and η. Then replace **NJpsi** with the row vector of derivatives of the shape functions with respect to ξ in order and do the same for **NJeta** for derivatives with respect to η. Now these become 1x9 matrices so the next step is to match the dimensions of other matrix quantities. **ke** is initialized to be 18x18 and **BJ** to be 3x18 since there are 9 stacked sets of 3x2 **B** matrices. **BJ** would then have to load in the proper values from the 2x9 matrix **NJdxy**. We can keep using 2-point Gaussian quadrature since it gives us forth-order accuracy. Finally, the new coordinates of the element nodes would have to be loaded from **node** so that **xe** and **ye** have the correct values, so next we move on to *mesh2d.m*, where **node** is generated.

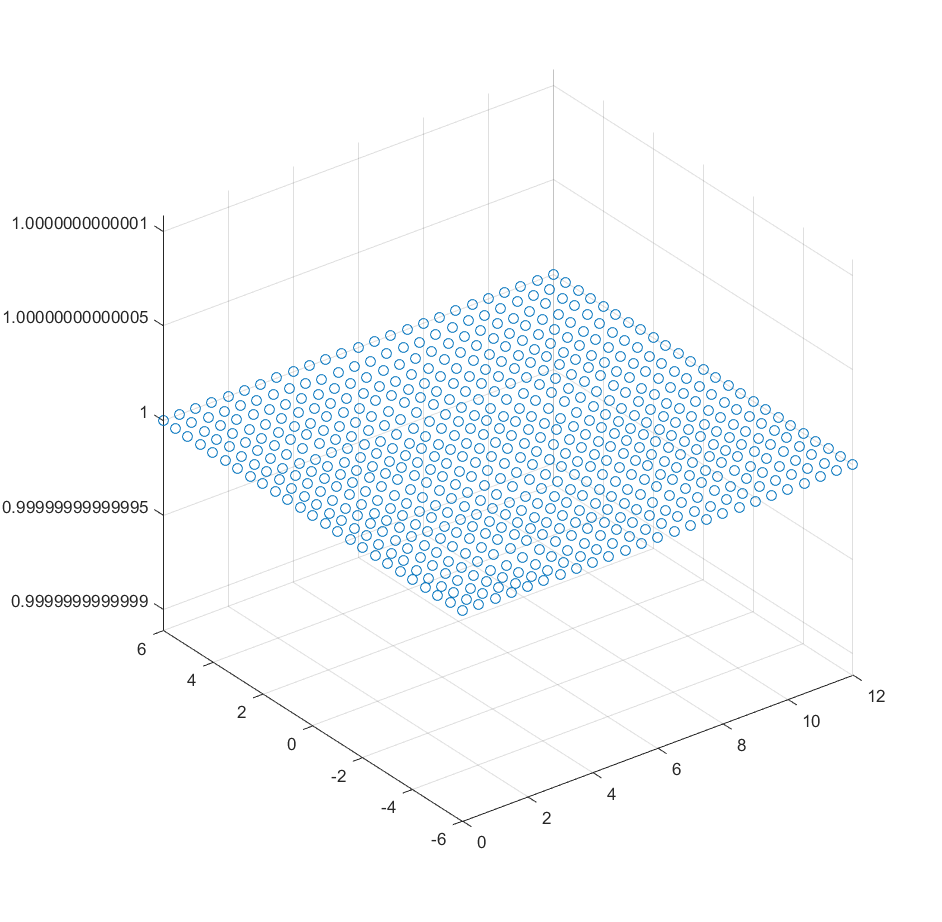
Since we are now using biquadratic elements, now *n* elements in a dimension corresponds to having *n+2* nodes so we update *numnod* and other occurrences of looping over nodes and interpolating nodal positions to reflect this. **nodet** will now have 9 rows with a numbering format similar to the original bilinear nodes. Lastly, *nlink* in the main *heat2d.m* is set to 9.

**Example problems**

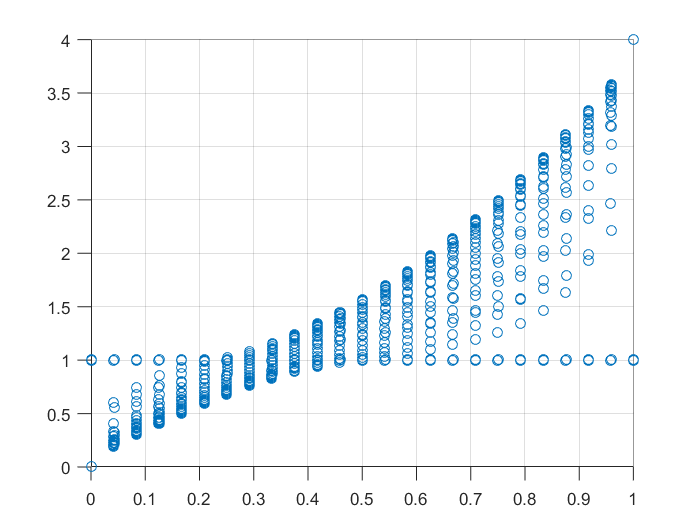
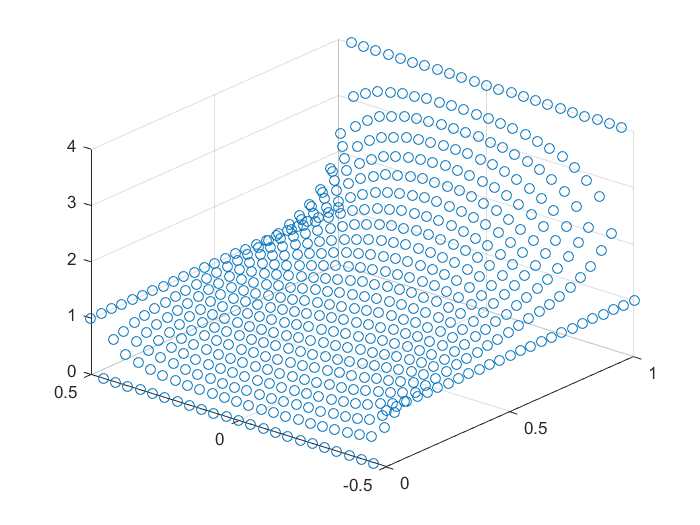
Length=12, Height=12, 25 nodes in x and y directions, u=1 on Γ*g* = All 4 edges, Γ*h* = ∅, Bilinear element:



Length=12, Height=12, 25 nodes in x and y directions, u=1 on Γ*g* = All 4 edges, Γ*h* = ∅, Biquadratic element:



Length=1, Height=1, 25 nodes in x and y directions, u=0 on x=0, u=1 on y=-height/2 and height/2, u=4 on x=length, Γ*h* = ∅, Biquadratic element:



Length=1, Height=1, u=0 on x=0, u=1 on y=-height/2 and height/2, u=4 on x=length, Γ*h* = ∅, MATLAB Parabolic PDE solver (based on <http://www.mathworks.com/help/pde/examples/inhomogeneous-heat-equation-on-a-square-domain.html>):

