## **Computer Implementation 6.4** ***(Matlab)*** *2D numerical integration (p. 417)*

Consider evaluation of the following integral using Gauss quadrature.

$$I = \int_{-1}^{1}\int_{-1}^{1}(400\,s^5 + 675\,s^3 + 25\,s - 900\,s^2\,t^6 - 200\,t^2 + 0.2)\ \mathrm{ds}\ \mathrm{dt}$$

The built-in *Matlab* function dblquad can be used directly to integrate this function. The function uses an adaptive approach. It keeps increasing the order of integration until the integral has been evaluated to a desired precision.

```
>> dblquad(inline('0.2 + 25.*s - 200.*t.^2 + 675.*s.^3 - 900.*s.^2.*t.^6 + 400.*s.^5'), -1,1,-1,1)

ans =

 -437.2952
```

However, frequently in finite element computations, a more direct control is desired over the Gauss points used in the evaluation. In these situations we can take the product of locations and weights of the one dimensional Gauss quarature and then perform calculations as shown in the previous examples. For example to evaluate the given integral using $3\times4$ rule in *Matlab* these calculations can be conveniently organized as follows.

MatlabFiles\Chap6\TwoDGaussQuadratureEx.m

```
% Integration over a square using 4x4 Gaussian quadrature
gaussPoints=[-0.8611363115940526, -0.8611363115940526;
   -0.8611363115940526, -0.3399810435848563;
   -0.8611363115940526, 0.3399810435848563;
   -0.8611363115940526, 0.8611363115940526;
   -0.3399810435848563, -0.8611363115940526;
   -0.3399810435848563, -0.3399810435848563;
   -0.3399810435848563, 0.3399810435848563;
   -0.3399810435848563, 0.8611363115940526;
   0.3399810435848563, -0.8611363115940526;
   0.3399810435848563, -0.3399810435848563;
   0.3399810435848563, 0.3399810435848563;
   0.3399810435848563, 0.8611363115940526;
   0.8611363115940526, -0.8611363115940526;
   0.8611363115940526, -0.3399810435848563;
   0.8611363115940526, 0.3399810435848563;
   0.8611363115940526, 0.8611363115940526];

gaussWeights=[0.12100299328560216, 0.22685185185185194, ...
      0.22685185185185194, 0.12100299328560216, ...
```

```
        0.22685185185185194, 0.42529330301069407, ...
        0.42529330301069407, 0.22685185185185194, ...
        0.22685185185185194, 0.42529330301069407, ...
        0.42529330301069407, 0.22685185185185194, ...
        0.12100299328560216, 0.22685185185185194, ...
        0.22685185185185194, 0.12100299328560216];
int=0;
for i=1:length(gaussWeights)
    s = gaussPoints(i,1); t = gaussPoints(i,2);
    fst = 0.2 + 25*s + 675*s^3 + 400*s^5 - 200*t^2 - 900*s^2*t^6;
    int = int + gaussWeights(i)*fst;
end
int
```

```
>> TwoDGaussQuadratureEx

int =

 -437.2952
```

Comparing with the exact solution the $4 \times 4$ formula gives the exact integral. Thus any higher order integration formula will give the same answer.