In 2-D

Assume we have a solution $u \in V$ in the following form:

$$u(x,y) = \sum_{h=1}^{n^2} \psi_h(x,y)u_h,\ u_h \in \mathbb{R}\ \forall h = 1, \dots, n^2 =: N \tag{1}$$

where each ψ_h is a shape function. These shape functions are given by a tensor product of one dimensional polynomials. The transformation given by

$$h = j(n-1) + i,\ i < n \leftrightarrow (i,j) \tag{2}$$

gives us an alternative representation of u:

$$u(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} \varphi_i(x)\varphi_j(y)u_{ij},\ i,j = 1, \dots n \tag{3}$$

Let $q = (q_1, \dots, q_n)^T$ be quadrature points of a quadrature rule in 1-D with corresponding weights $w = (w_1, \dots, w_n)^T$. We can get a two-dimensional rule by using the tensor product and obtain points $(\boldsymbol{q}_1, \dots, \boldsymbol{q}_N)$ and weights $(\boldsymbol{w}_1, \dots, \boldsymbol{w}_N)$ with $\boldsymbol{q}_h = (q_i, q_j)$ and $\boldsymbol{w}_h = w_i w_j$.

First, we want to evaluate u at every quadrature point. This can be done by a matrix vector multiplication:

$$\begin{pmatrix} \psi_1(\boldsymbol{q}_1) & \dots & \psi_N(\boldsymbol{q}_1) \\ \vdots & \ddots & \vdots \\ \psi_1(\boldsymbol{q}_N) & \dots & \psi_N(\boldsymbol{q}_N) \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} u(\boldsymbol{q}_1) \\ \vdots \\ u(\boldsymbol{q}_N) \end{pmatrix} \tag{4}$$

The matrix can be written as a tensor product of two (in this case even identical) matrices:

$$\mathcal{N} \otimes \mathcal{N} = \begin{pmatrix} \varphi_1(q_1) & \dots & \varphi_n(q_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(q_n) & \dots & \varphi_n(q_n) \end{pmatrix} \otimes \begin{pmatrix} \varphi_1(q_1) & \dots & \varphi_n(q_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(q_n) & \dots & \varphi_n(q_n) \end{pmatrix} \tag{5}$$

Let $(\mathcal{U}_{ij}) \in \mathbb{R}^{n \times n}$ be the matrix filled with the coefficients u_{ij}. Using sum factorization we can multiply these matrices instead of using the formula in (4) which gives us

$$\mathcal{N}\mathcal{U}\mathcal{N}^T = \bar{\mathcal{U}}, \quad \bar{\mathcal{U}}_{ij} = u(q_i, q_j) \tag{6}$$

The quadrature weights and the summation of the entries can also be done in a efficient way when the tensor product is exploited. Multiplying $\bar{\mathcal{U}}$ with quadrature weights w from the right will sum each weighted column, multyplying w^T from the left will do the rest. The full operation therefore reads

$$\int\int u(x,y)dx\ dy \approx w^T \mathcal{N}\mathcal{U}\mathcal{N}^T w \tag{7}$$

So far, we have only integrated u, but we can also test it with an ansatz function without having to change much. Consider some function $f(x,y) = g(x)h(y)$. Since f is seperable, we only have to change the last step. Instead of only multiplying weigths, we will also multiply with the ansatz function evaluated at the respective quadrature point:

$$\bar{w}^x = (w_1 g(q_1), \dots, w_n g(q_n))^T \qquad \bar{w}^y = (w_1 h(q_1), \dots, w_n h(q_n))^T$$

$$\int\int u(x,y)f(x,y)dx\ dy \approx (\bar{w}^y)^T \mathcal{N}\mathcal{U}\mathcal{N}^T \bar{w}^x \tag{8}$$