AMCS 394E: Contemp. Topics in Computational Science.
Computing with the finite element method

David I. Ketcheson and Manuel Quezada de Luna

King Abdullah University of Science and Technology

What have we seen so far?

Approximation theory:

What have we seen so far?

Approximation theory:

* How do the solutions of PDEs look?

What have we seen so far?

Approximation theory:

* How do the solutions of PDEs look?
* How can we approximate these solutions?

What have we seen so far?

Approximation theory:

* How do the solutions of PDEs look?
* How can we approximate these solutions?
* What is the error we make in this approximation?

What have we seen so far?

Approximation theory:

* How do the solutions of PDEs look?
* How can we approximate these solutions?
* What is the error we make in this approximation?

Approximation of PDEs:

What have we seen so far?

Approximation theory:

* How do the solutions of PDEs look?
* How can we approximate these solutions?
* What is the error we make in this approximation?

Approximation of PDEs:

* Why do we go from a strong to a weak formulation?

What have we seen so far?

Approximation theory:

* How do the solutions of PDEs look?
* How can we approximate these solutions?
* What is the error we make in this approximation?

Approximation of PDEs:

* Why do we go from a strong to a weak formulation?
* How do we get a finite dimensional approx. of the weak form?

What have we seen so far?

Approximation theory:

* How do the solutions of PDEs look?
* How can we approximate these solutions?
* What is the error we make in this approximation?

Approximation of PDEs:

* Why do we go from a strong to a weak formulation?
* How do we get a finite dimensional approx. of the weak form?
* How do we go from a finite dimensional approx. of the weak form to a linear algebra problem?

What have we seen so far?

Example:

Consider the following equation:

$$
\begin{aligned}
u - \Delta u &= 0, && \forall x \in \Omega, \\
u &= u_D, && \forall x \in \Gamma_D, \\
\partial_{\mathbf{n}} u &= b(x), && \forall x \in \Gamma_N.
\end{aligned}
$$

What have we seen so far?

Example:

Consider the following equation:

$$\begin{aligned} u - \Delta u &= 0, & \forall x &\in \Omega, \\ u &= u_D, & \forall x &\in \Gamma_D, \\ \partial_{\mathbf{n}} u &= b(x), & \forall x &\in \Gamma_N. \end{aligned}$$

What is the weak formulation?

$$\int_\Omega u\varphi dx + \int_\Omega \nabla u \cdot \nabla \varphi dx - \int_{\partial\Omega} (\nabla u \cdot \mathbf{n})\varphi ds = 0, \quad \forall \varphi \in V$$

What have we seen so far?

Example:

Consider the following equation:

$$
\begin{aligned}
u - \Delta u &= 0, & \forall x &\in \Omega, \\
u &= u_D, & \forall x &\in \Gamma_D, \\
\partial_{\mathbf{n}} u &= b(x), & \forall x &\in \Gamma_N.
\end{aligned}
$$

What is the weak formulation?

$$
\int_\Omega u\varphi dx + \int_\Omega \nabla u \cdot \nabla \varphi dx - \int_{\partial\Omega} b\varphi ds = 0, \quad \forall \varphi \in V
$$

What have we seen so far?

Example:

Consider the following equation:

$$
\begin{aligned}
u - \Delta u &= 0, && \forall x \in \Omega, \\
u &= u_D, && \forall x \in \Gamma_D, \\
\partial_{\mathbf{n}} u &= b(x), && \forall x \in \Gamma_N.
\end{aligned}
$$

What is the weak formulation?

$$\int_\Omega u\varphi dx + \int_\Omega \nabla u \cdot \nabla \varphi dx - \int_{\partial\Omega} b\varphi ds = 0, \quad \forall \varphi \in V$$

where

$$V = \{v \in H^1(\Omega) \mid v = 0 \text{ if } x \in \Gamma_D\}.$$

What have we seen so far?

What is the discrete weak formulation?

$$\int_{\Omega} u_h \varphi dx + \int_{\Omega} \nabla u_h \cdot \nabla \varphi dx = \int_{\partial\Omega} b \varphi ds, \quad \forall \varphi \in V_h$$

What have we seen so far?

What is the discrete weak formulation?

$$\int_\Omega u_h \varphi dx + \int_\Omega \nabla u_h \cdot \nabla \varphi dx = \int_{\partial\Omega} b \varphi ds, \quad \forall \varphi \in V_h$$

where

$$V_h = \{v \in C^0(\Omega) \mid v_K \in \mathbb{P}^p\}.$$

What have we seen so far?

What is the discrete weak formulation?

$$\int_\Omega u_h \varphi dx + \int_\Omega \nabla u_h \cdot \nabla \varphi dx = \int_{\partial\Omega} b \varphi ds, \quad \forall \varphi \in V_h$$

where

$$V_h = \{v \in C^0(\Omega) \mid v_K \in \mathbb{P}^p\}.$$

How do we get a linear algebra problem?

Consider and plug $\phi = \phi_i$ and $u_h = \sum_j U_j \varphi_j$ into the equation.

$$\sum_j U_j \int_\Omega \varphi_i \varphi_j dx + \sum_j U_j \int_\Omega \nabla \varphi_i \cdot \nabla \varphi_j dx = \int_{\partial\Omega} b \varphi_i ds.$$

What have we seen so far?

What is the discrete weak formulation?

$$\int_\Omega u_h \varphi dx + \int_\Omega \nabla u_h \cdot \nabla \varphi dx = \int_{\partial\Omega} b\varphi ds, \quad \forall \varphi \in V_h$$

where

$$V_h = \{v \in C^0(\Omega) \mid v_K \in \mathbb{P}^p\}.$$

How do we get a linear algebra problem?

Consider and plug $\phi = \phi_i$ and $u_h = \sum_j U_j \varphi_j$ into the equation.

$$\sum_j U_j \underbrace{\int_\Omega \varphi_i \varphi_j dx}_{=M_{ij}} + \sum_j U_j \underbrace{\int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j dx}_{=S_{ij}} = \underbrace{\int_{\partial\Omega} b\varphi_i ds}_{=F(\phi_i)}.$$

What have we seen so far?

What is the discrete weak formulation?

$$\int_\Omega u_h \varphi dx + \int_\Omega \nabla u_h \cdot \nabla \varphi dx = \int_{\partial\Omega} b\varphi ds, \quad \forall \varphi \in V_h$$

where

$$V_h = \{v \in C^0(\Omega) \mid v_K \in \mathbb{P}^p\}.$$

How do we get a linear algebra problem?

Consider and plug $\phi = \phi_i$ and $u_h = \sum_j U_j \varphi_j$ into the equation.

$$\sum_j U_j \Bigg[\underbrace{\int_\Omega \varphi_i \varphi_j}_{=M_{ij}} dx + \underbrace{\int_\Omega \nabla \varphi_i \cdot \nabla \varphi_j dx}_{=S_{ij}} \Bigg] = \underbrace{\int_{\partial\Omega} b\varphi_i ds}_{=F(\phi_i)}.$$

Linear algebra problem from a FE discretization

We want to solve

$$\sum_j U_j \underbrace{\int_\Omega \varphi_i\varphi_j + \nabla\varphi_i \cdot \nabla\varphi_j dx}_{=A_{ij}} = \underbrace{\int_{\partial\Omega} b\varphi_i ds}_{=F(\phi_i)}, \quad i = 1, \ldots, \dim(V_h).$$

Linear algebra problem from a FE discretization

We want to solve

$$\sum_j U_j \underbrace{\int_\Omega \varphi_i \varphi_j + \nabla\varphi_i \cdot \nabla\varphi_j dx}_{=A_{ij}} = \underbrace{\int_{\partial\Omega} b\varphi_i ds}_{=F(\phi_i)}, \quad i = 1, \ldots, \dim(V_h).$$

Recall the space V_h and that $\phi_i \in V_h$

$$V_h = \{\varphi \in C^0(\Omega) \mid \varphi|_K \in \mathbb{P}^p\}$$

Linear algebra problem from a FE discretization

We want to solve

$$\sum_j U_j \underbrace{\int_\Omega \varphi_i \varphi_j + \nabla \varphi_i \cdot \nabla \varphi_j dx}_{=A_{ij}} = \underbrace{\int_{\partial\Omega} b\varphi_i ds}_{=F(\phi_i)}, \quad i = 1, \ldots, \dim(V_h).$$

Recall the space V_h and that $\phi_i \in V_h$

$$V_h = \{\varphi \in C^0(\Omega) \mid \varphi|_K \in \mathbb{P}^p\}$$

Assume $\Omega = (0, 1)$ and consider $p = 1$

Linear algebra problem from a FE discretization

We want to solve

$$\sum_j U_j \underbrace{\int_\Omega \varphi_i \varphi_j + \nabla \varphi_i \cdot \nabla \varphi_j dx}_{=A_{ij}} = \underbrace{\int_{\partial\Omega} b \varphi_i ds}_{=F(\phi_i)}, \quad i = 1, \ldots, \dim(V_h).$$

Recall the space V_h and that $\phi_i \in V_h$

$$V_h = \{\varphi \in C^0(\Omega) \mid \varphi|_K \in \mathbb{P}^p\}$$

Assume $\Omega = (0, 1)$ and consider $p = 1$

* How do the shape (or basis) functions look like?

Linear algebra problem from a FE discretization

We want to solve

$$\sum_j U_j \underbrace{\int_\Omega \varphi_i \varphi_j + \nabla \varphi_i \cdot \nabla \varphi_j dx}_{=A_{ij}} = \underbrace{\int_{\partial\Omega} b \varphi_i ds}_{=F(\phi_i)}, \quad i = 1, \ldots, \dim(V_h).$$

Recall the space V_h and that $\phi_i \in V_h$

$$V_h = \{\varphi \in C^0(\Omega) \mid \varphi|_K \in \mathbb{P}^p\}$$

Assume $\Omega = (0, 1)$ and consider $p = 1$

* How do the shape (or basis) functions look like?
* Let's consider a discretization of Ω.

Linear algebra problem from a FE discretization

We want to solve

$$\sum_j U_j \underbrace{\int_\Omega \varphi_i\varphi_j + \nabla\varphi_i \cdot \nabla\varphi_j dx}_{=A_{ij}} = \underbrace{\int_{\partial\Omega} b\varphi_i ds}_{=F(\phi_i)}, \quad i = 1, \ldots, \dim(V_h).$$

Recall the space V_h and that $\phi_i \in V_h$

$$V_h = \{\varphi \in C^0(\Omega) \mid \varphi|_K \in \mathbb{P}^p\}$$

Assume $\Omega = (0,1)$ and consider $p = 1$

* How do the shape (or basis) functions look like?
* Let's consider a discretization of Ω.
* Let's compute the matrices and RHS.

Linear algebra problem from a FE discretization

Second example:

Let's consider the 1D equation (from the first week):

$$\partial_t u + a\partial_x u - \mu\partial_{xx} u = r(x), \quad \forall x \in \Omega = (0, 1),$$

with $\mu > 0$.

Linear algebra problem from a FE discretization

Second example:

Let's consider the 1D equation (from the first week):

$$\partial_t u + a\partial_x u - \mu\partial_{xx} u = r(x), \quad \forall x \in \Omega = (0,1),$$

with $\mu > 0$.

Get the finite element spatial semi-discretization:

Linear algebra problem from a FE discretization

Second example:

Let's consider the 1D equation (from the first week):

$$\partial_t u + a\partial_x u - \mu\partial_{xx} u = r(x), \quad \forall x \in \Omega = (0, 1),$$

with $\mu > 0$.

Get the finite element spatial semi-discretization:

* Consider piecewise linear continuous finite elements.

Linear algebra problem from a FE discretization

Second example:

Let's consider the 1D equation (from the first week):

$$\partial_t u + a\partial_x u - \mu \partial_{xx} u = r(x), \quad \forall x \in \Omega = (0, 1),$$

with $\mu > 0$.

Get the finite element spatial semi-discretization:

* Consider piecewise linear continuous finite elements.
* What is the discrete weak formulation?

Linear algebra problem from a FE discretization

Second example:

Let's consider the 1D equation (from the first week):

$$\partial_t u + a\partial_x u - \mu\partial_{xx} u = r(x), \quad \forall x \in \Omega = (0, 1),$$

with $\mu > 0$.

Get the finite element spatial semi-discretization:

* Consider piecewise linear continuous finite elements.
* What is the discrete weak formulation?
* Compute the mass, transport and stiffness matrices.

Linear algebra problem from a FE discretization

Second example:

Let's consider the 1D equation (from the first week):

$$\partial_t u + a\partial_x u - \mu\partial_{xx} u = r(x), \quad \forall x \in \Omega = (0, 1),$$

with $\mu > 0$.

Get the finite element spatial semi-discretization:

* Consider piecewise linear continuous finite elements.
* What is the discrete weak formulation?
* Compute the mass, transport and stiffness matrices.
* Express the equation for the *i*-th DoF.

Linear algebra problem from a FE discretization

Second example:

Let's consider the 1D equation (from the first week):

$$\partial_t u + a\partial_x u - \mu\partial_{xx} u = r(x), \quad \forall x \in \Omega = (0, 1),$$

with $\mu > 0$.

Get the finite element spatial semi-discretization:

* Consider piecewise linear continuous finite elements.
* What is the discrete weak formulation?
* Compute the mass, transport and stiffness matrices.
* Express the equation for the i-th DoF.
* Compare this versus the finite difference formulation.

Summary of what we did

Given the discrete weak formulation we proceeded as follows:

Summary of what we did

Given the discrete weak formulation we proceeded as follows:

* Identify each matrix and vectors we needed to compute.

Summary of what we did

Given the discrete weak formulation we proceeded as follows:

* Identify each matrix and vectors we needed to compute.
* To compute the matrices and vectors we needed to perform integrals on the domain.

Summary of what we did

Given the discrete weak formulation we proceeded as follows:

* Identify each matrix and vectors we needed to compute.
* To compute the matrices and vectors we needed to perform integrals on the domain.
* We broke the integration over the domain in elements.

Summary of what we did

Given the discrete weak formulation we proceeded as follows:

* Identify each matrix and vectors we needed to compute.
* To compute the matrices and vectors we needed to perform integrals on the domain.
* We broke the integration over the domain in elements.
* We considered any node i and its shape function φ_i.

Summary of what we did

Given the discrete weak formulation we proceeded as follows:

* Identify each matrix and vectors we needed to compute.
* To compute the matrices and vectors we needed to perform integrals on the domain.
* We broke the integration over the domain in elements.
* We considered any node i and its shape function φ_i.

  Note: if the mesh is not uniform, we need to consider all nodes independently.

Summary of what we did

Given the discrete weak formulation we proceeded as follows:

* Identify each matrix and vectors we needed to compute.
* To compute the matrices and vectors we needed to perform integrals on the domain.
* We broke the integration over the domain in elements.
* We considered any node i and its shape function φ_i.

  Note: if the mesh is not uniform, we need to consider all nodes independently.
* To compute the matrices, for a given node i, we considered all j-shape functions.

Summary of what we did

Given the discrete weak formulation we proceeded as follows:

* Identify each matrix and vectors we needed to compute.
* To compute the matrices and vectors we needed to perform integrals on the domain.
* We broke the integration over the domain in elements.
* We considered any node i and its shape function φ_i.

  Note: if the mesh is not uniform, we need to consider all nodes independently.
* To compute the matrices, for a given node i, we considered all j-shape functions.
* We performed the integration based on a reference element.

Quadratures

We approximate the integrals via quadratures. That is (for 1D),

$$\int_{-1}^{1} f(x)dx \approx \sum_{q=1}^{N_q} w_q f(x_q),$$

where N_q is the number of quadrature points.

Quadratures

We approximate the integrals via quadratures. That is (for 1D),

$$\int_{-1}^{1} f(x)dx \approx \sum_{q=1}^{N_q} w_q f(x_q),$$

where N_q is the number of quadrature points.

Gaussian quadratures

Quadratures

We approximate the integrals via quadratures. That is (for 1D),

$$\int_{-1}^{1} f(x)dx \approx \sum_{q=1}^{N_q} w_q f(x_q),$$

where N_q is the number of quadrature points.

Gaussian quadratures

* In finite elements, it is common tu use Gaussian quadratures.

Quadratures

We approximate the integrals via quadratures. That is (for 1D),

$$\int_{-1}^{1} f(x)dx \approx \sum_{q=1}^{N_q} w_q f(x_q),$$

where N_q is the number of quadrature points.

Gaussian quadratures

* In finite elements, it is common tu use Gaussian quadratures.
* The quad points $\{x_q\}$ are the roots of a polynomial.

Quadratures

We approximate the integrals via quadratures. That is (for 1D),

$$\int_{-1}^{1} f(x)dx \approx \sum_{q=1}^{N_q} w_q f(x_q),$$

where N_q is the number of quadrature points.

Gaussian quadratures

* In finite elements, it is common tu use Gaussian quadratures.
* The quad points $\{x_q\}$ are the roots of a polynomial.
* Gauss-Legendre: $\{x_q\}$ are the roots of Legendre polynomials.

Quadratures

We approximate the integrals via quadratures. That is (for 1D),

$$\int_{-1}^{1} f(x)dx \approx \sum_{q=1}^{N_q} w_q f(x_q),$$

where N_q is the number of quadrature points.

Gaussian quadratures

* In finite elements, it is common tu use Gaussian quadratures.
* The quad points $\{x_q\}$ are the roots of a polynomial.
* Gauss-Legendre: $\{x_q\}$ are the roots of Legendre polynomials.
* Gauss-Legendre quadrature is exact for polynomials of degree up to $2N_q - 1$.

Quadratures: Gauss-Legendre

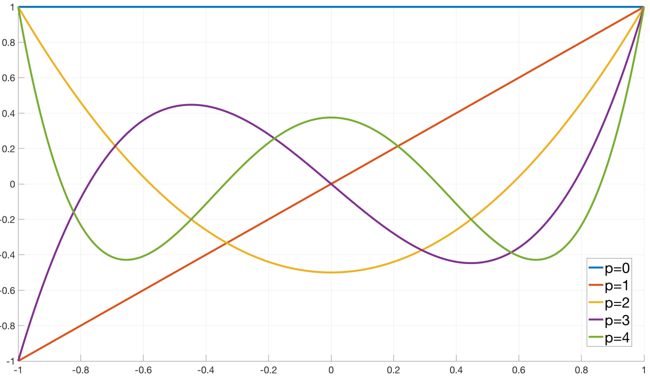
Quadrature points

The quadrature points $\{x_1, \ldots, x_{N_q}\}$ are the roots of the Legendre polynomials of degree N_q.

Quadratures: Gauss-Legendre

Quadrature points

The quadrature points $\{x_1, \ldots, x_{N_q}\}$ are the roots of the Legendre polynomials of degree N_q.



Quadratures: Gauss-Legendre

Quadrature weigths

The weights are defined so that polynomials of degree up to $2N_q - 1$ are integrated exactly.

Quadratures: Gauss-Legendre

Quadrature weigths

The weights are defined so that polynomials of degree up to $2N_q - 1$ are integrated exactly.

Example:

Consider piecewise linear poly. to solve the advection-diffusion eqn.

Quadratures: Gauss-Legendre

Quadrature weigths

The weights are defined so that polynomials of degree up to $2N_q - 1$ are integrated exactly.

Example:

Consider piecewise linear poly. to solve the advection-diffusion eqn.

- What is the highest degree of polynomials that I need to integrate?

Quadratures: Gauss-Legendre

Quadrature weigths

The weights are defined so that polynomials of degree up to $2N_q - 1$ are integrated exactly.

Example:

Consider piecewise linear poly. to solve the advection-diffusion eqn.

* What is the highest degree of polynomials that I need to integrate?
* The highest degree is given by the mass matrix.

Quadratures: Gauss-Legendre

Quadrature weigths

The weights are defined so that polynomials of degree up to $2N_q - 1$ are integrated exactly.

Example:

Consider piecewise linear poly. to solve the advection-diffusion eqn.

- What is the highest degree of polynomials that I need to integrate?
- The highest degree is given by the mass matrix.
- The integrand of the mass matrix is quadratic.

Quadratures: Gauss-Legendre

Quadrature weigths

The weights are defined so that polynomials of degree up to $2N_q - 1$ are integrated exactly.

Example:

Consider piecewise linear poly. to solve the advection-diffusion eqn.

* What is the highest degree of polynomials that I need to integrate?
* The highest degree is given by the mass matrix.
* The integrand of the mass matrix is quadratic.
* Then I need $2N_q - 1 \geq 2$. Therefore, $N_q \geq 3/2 \implies N_q = 2$.

Quadratures: Gauss-Legendre

Quadrature weigths

The weights are defined so that polynomials of degree up to $2N_q - 1$ are integrated exactly.

Example:

Consider piecewise linear poly. to solve the advection-diffusion eqn.

* What is the highest degree of polynomials that I need to integrate?
* The highest degree is given by the mass matrix.
* The integrand of the mass matrix is quadratic.
* Then I need $2N_q - 1 \geq 2$. Therefore, $N_q \geq 3/2 \implies N_q = 2$.
* With $N_q = 2$, I can integrate exactly polynomials up to degree 3.

Quadratures: Gauss-Legendre

Quadrature weigths

The weights are defined so that polynomials of degree up to $2N_q - 1$ are integrated exactly.

Example:

Consider piecewise linear poly. to solve the advection-diffusion eqn.

* What is the highest degree of polynomials that I need to integrate?
* The highest degree is given by the mass matrix.
* The integrand of the mass matrix is quadratic.
* Then I need $2N_q - 1 \geq 2$. Therefore, $N_q \geq 3/2 \implies N_q = 2$.
* With $N_q = 2$, I can integrate exactly polynomials up to degree 3.

Verify that I can integrate third order polynomials with $N_q = 2$.

The finite element (FE) loop

The kernel of most FE codes is the FE loop

The finite element (FE) loop

The kernel of most FE codes is the FE loop

* Loop on elements. We do this since we broke the integral in element integrals.

The finite element (FE) loop

The kernel of most FE codes is the FE loop

* Loop on elements. We do this since we broke the integral in element integrals.
* Compute element based integrals (vectors and matrices).

The finite element (FE) loop

The kernel of most FE codes is the FE loop

* Loop on elements. We do this since we broke the integral in element integrals.
* Compute element based integrals (vectors and matrices).
* Combine (or assemble) the element integrals to global integrals.

The finite element (FE) loop

The kernel of most FE codes is the FE loop

* Loop on elements. We do this since we broke the integral in element integrals.
* Compute element based integrals (vectors and matrices).
* Combine (or assemble) the element integrals to global integrals.

How to compute the element based integrals?

For each element, proceed as follows:

The finite element (FE) loop

The kernel of most FE codes is the FE loop

* Loop on elements. We do this since we broke the integral in element integrals.
* Compute element based integrals (vectors and matrices).
* Combine (or assemble) the element integrals to global integrals.

How to compute the element based integrals?

For each element, proceed as follows:

* Compute all quantities with respect to the reference element.

The finite element (FE) loop

The kernel of most FE codes is the FE loop

* Loop on elements. We do this since we broke the integral in element integrals.
* Compute element based integrals (vectors and matrices).
* Combine (or assemble) the element integrals to global integrals.

How to compute the element based integrals?

For each element, proceed as follows:

* Compute all quantities with respect to the reference element.
* Loop on quadrature points. We do this since we approximate the integrals via quadratures.

The finite element (FE) loop

The kernel of most FE codes is the FE loop

* Loop on elements. We do this since we broke the integral in element integrals.
* Compute element based integrals (vectors and matrices).
* Combine (or assemble) the element integrals to global integrals.

How to compute the element based integrals?

For each element, proceed as follows:

* Compute all quantities with respect to the reference element.
* Loop on quadrature points. We do this since we approximate the integrals via quadratures.
* Loop on i-DoFs.

The finite element (FE) loop

The kernel of most FE codes is the FE loop

* Loop on elements. We do this since we broke the integral in element integrals.
* Compute element based integrals (vectors and matrices).
* Combine (or assemble) the element integrals to global integrals.

How to compute the element based integrals?

For each element, proceed as follows:

* Compute all quantities with respect to the reference element.
* Loop on quadrature points. We do this since we approximate the integrals via quadratures.
* Loop on i-DoFs.
* Loop on j-DoFs and more indices if needed.

The finite element (FE) loop

The FE loop looks as follows:

The finite element (FE) loop

The FE loop looks as follows:

for $K = 1$ to N_{el} **do**

end for

The finite element (FE) loop

The FE loop looks as follows:

for $K = 1$ to N_{el} **do**
Compute quantities wrt reference element.

end for

The finite element (FE) loop

The FE loop looks as follows:

for $K = 1$ to N_{el} **do**
  Compute quantities wrt reference element.
  for $q = 1$ to N_q **do**

  end for
end for

The finite element (FE) loop

The FE loop looks as follows:

for $K = 1$ to N_{el} **do**
  Compute quantities wrt reference element.
  for $q = 1$ to N_q **do**
    for $i = 1$ to DoFs per element **do**

    end for

  end for
end for

The finite element (FE) loop

The FE loop looks as follows:

for $K = 1$ to N_{el} **do**
  Compute quantities wrt reference element.
  for $q = 1$ to N_q **do**
    for $i = 1$ to DoFs per element **do**
      Compute element based vectors.
    end for
  end for
end for

The finite element (FE) loop

The FE loop looks as follows:

```
for K = 1 to N_el do
   Compute quantities wrt reference element.
   for q = 1 to N_q do
      for i = 1 to DoFs per element do
         Compute element based vectors.
         for j = 1 to DoFs per element do

         end for
      end for

   end for
end for
```

The finite element (FE) loop

The FE loop looks as follows:

```
for K = 1 to N_el do
    Compute quantities wrt reference element.
    for q = 1 to N_q do
        for i = 1 to DoFs per element do
            Compute element based vectors.
            for j = 1 to DoFs per element do
                Compute element based matrices.
            end for
        end for

    end for
end for
```

The finite element (FE) loop

The FE loop looks as follows:

for $K = 1$ to N_{el} **do**
  Compute quantities wrt reference element.
  for $q = 1$ to N_q **do**
    for $i = 1$ to DoFs per element **do**
      Compute element based vectors.
      for $j = 1$ to DoFs per element **do**
        Compute element based matrices.
      end for
    end for
    Assemble from local to global operators.
  end for
end for