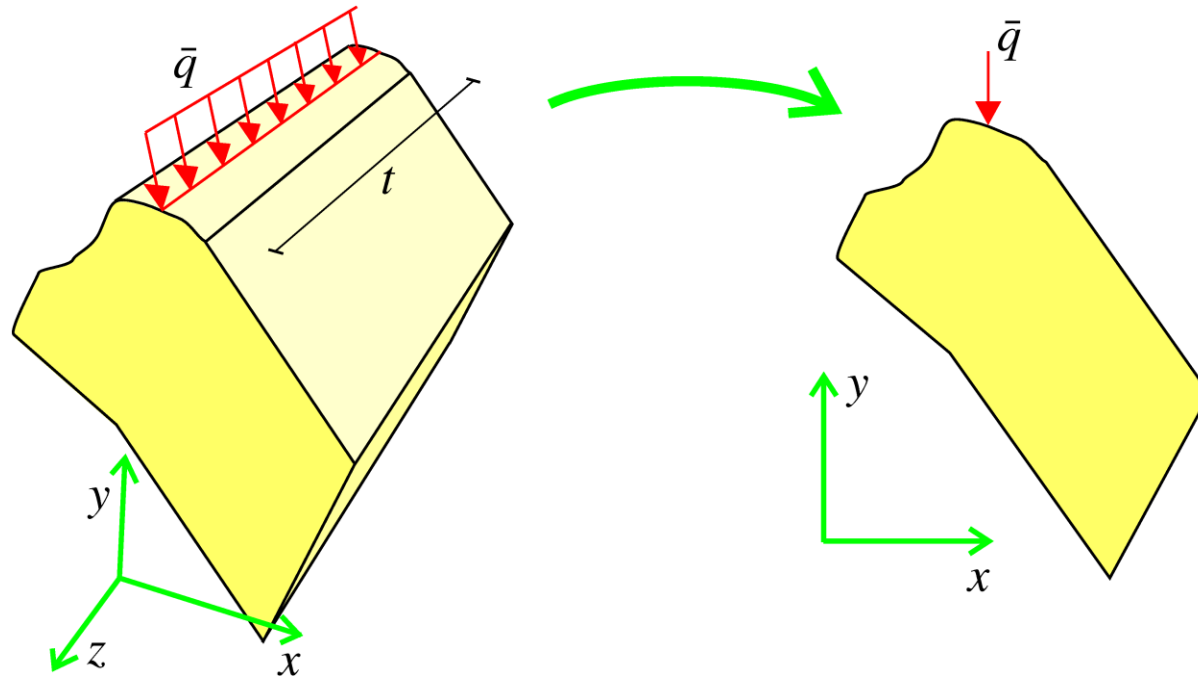


### Heat conduction in two dimensions

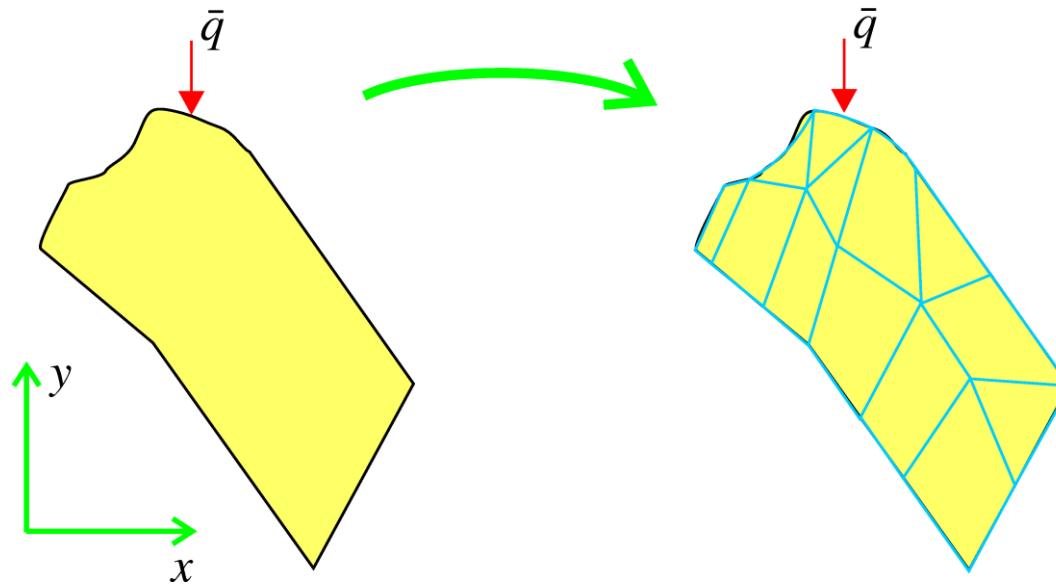


- ☐ All real bodies are three-dimensional (3D)
- ☐ If the heat supplies, prescribed temperatures and material characteristics are independent of the  $z$ -coordinate, the domain can be approximated with a 2D domain with the thickness  $t(x,y)$

# Finite Element Method

## 2D heat conduction

### Discretization into two-dimensional finite elements



- ☐ The body is discretized with a number of finite elements
- ☐ They can be triangular or quadrilateral
- ☐ Straight or curved element sides

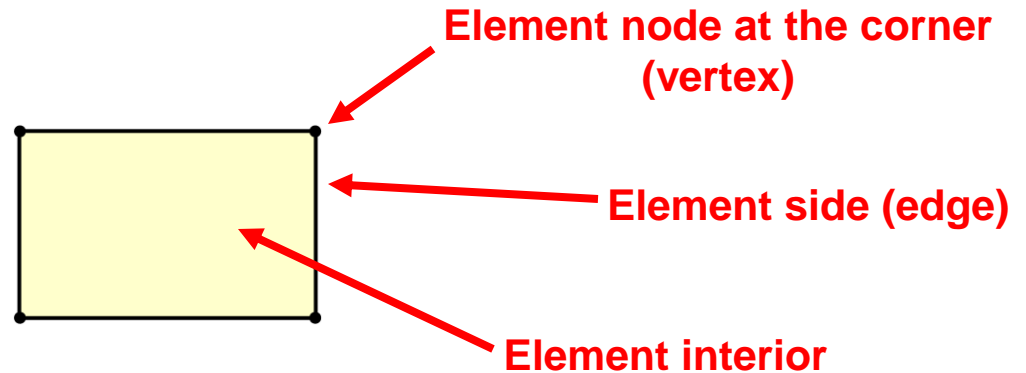
# Finite Element Method

## 2D heat conduction

### Today's elements

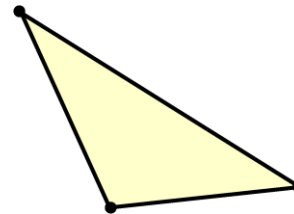
#### ☐ Melosh element

- ◆ Four nodes
- ◆ Only right angles
- ◆ Must be aligned with the coordinate axes



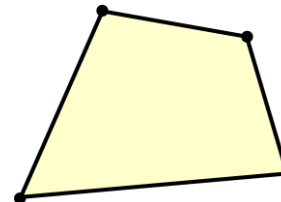
#### ☐ Constant strain triangle (CST)

- ◆ Three nodes
- ◆ Arbitrary angles
- ◆ Arbitrary orientation



#### ☐ Isoparametric four-node element

- ◆ Four nodes
- ◆ Arbitrary angles
- ◆ Arbitrary orientation

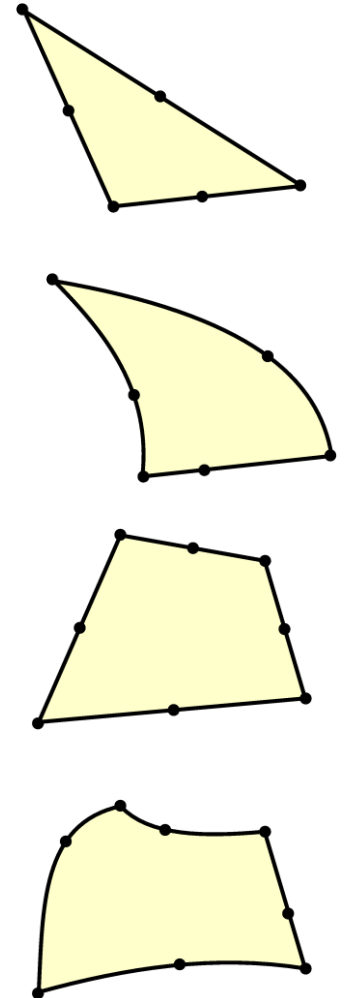


# Finite Element Method

## 2D heat conduction

### Other useful elements

- ☐ Triangular six-node element with straight edges and midside nodes (Lecture 8 + 9)
- ☐ Triangular isoparametric six-node element with curved edges and off-midside nodes
- ☐ Isoparametric eight-noded elements with straight edges and midside nodes
- ☐ Isoparametric eight-noded elements with curved edges and off-midside nodes (Lecture 8 + 9)

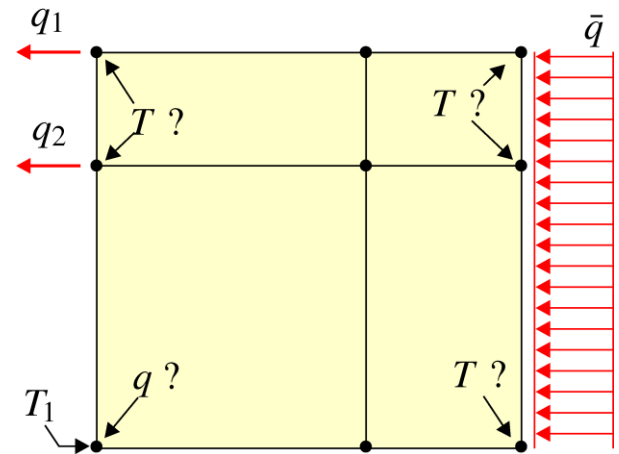


# Finite Element Method

## 2D heat conduction

### The patch test

- ❑ The main exercise of today is the patch test. You will program its main parts and use it to verify that the elements work properly.
- ❑ A patch test should include at least one internal node
- ❑ The patch test should be as unconstrained as possible, i.e. the temperature should be specified at only one node.
- ❑ Before we go into the equations of 2D heat conduction, you must now start MatLab and do a small exercise.

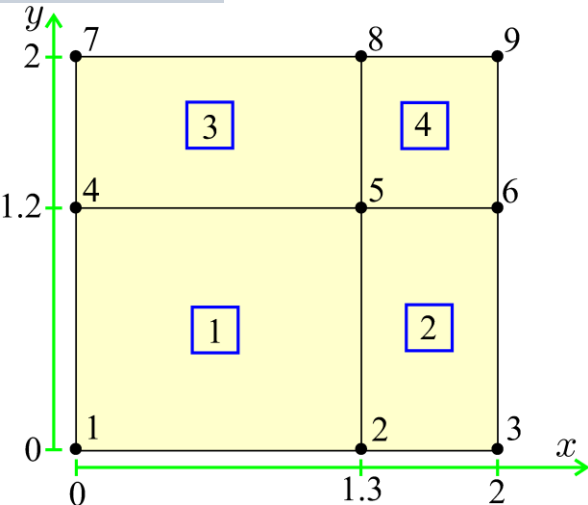


# Finite Element Method

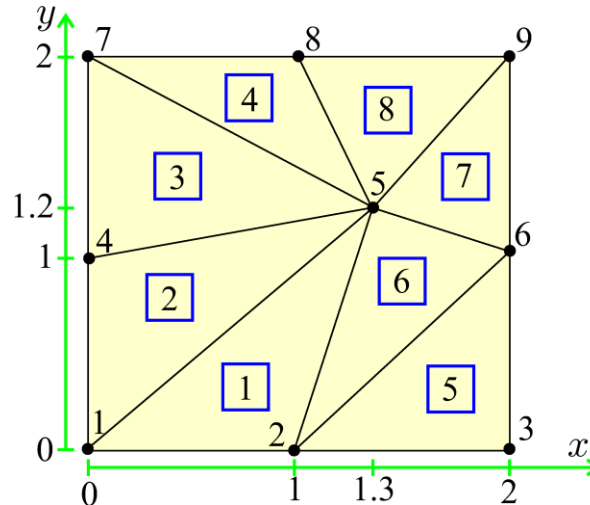
## 2D heat conduction

**Exercise: Program and plot the geometry and topology of a 2D patch test for the Melosh, CST and ISO4 elements**

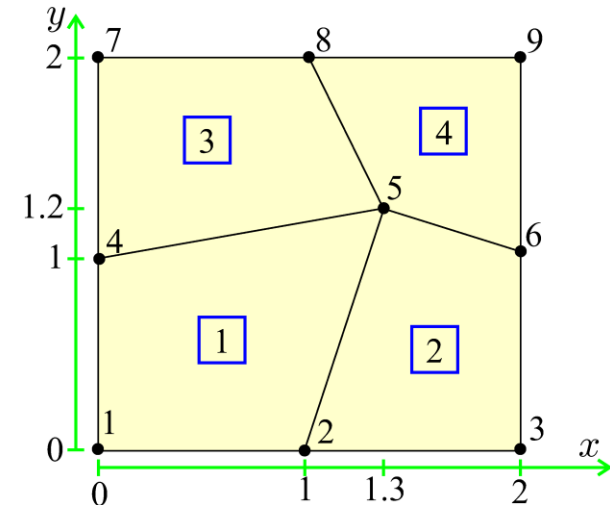
Melosh **etype = 1**



CST **etype = 3**



Iso4 **etype = 7**



**Coord = [ x1 y1 z1 ; x2 y2 y3 ; .... ] % Nodal coordinates in file “Coordinates.m”**

**Top = [ etype section etc ] % for each element in the file “Topology.m”**

**Plot by executing the line “visualize2D”**

### Basic steps of the finite-element method (FEM)

1. Establish strong formulation
  - ◆ Partial differential equation
2. Establish weak formulation
  - ◆ Multiply with arbitrary field and integrate over element
3. Discretize over space
  - ◆ Mesh generation
4. Select shape and weight functions
  - ◆ Galerkin method
5. Compute element stiffness matrix
  - ◆ Local and global system
6. Assemble global system stiffness matrix
7. Apply nodal boundary conditions
  - ◆ temperature/flux/forces/forced displacements
8. Solve global system of equations
  - ◆ Solve for nodal values of the primary variables (displacements/temperature)
9. Compute temperature/stresses/strains etc. within the element
  - ◆ Using nodal values and shape functions

# Finite Element Method

## 2D heat conduction

### Step I: Establish strong formulation for 2D heat conduction (OP pp. 76-84)

Heat flow (flux) vector  $\mathbf{q} = \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} \quad \left[ \frac{\text{J}}{\text{m}^2\text{s}} \right]$

Boundary normal  $\mathbf{n} = \begin{Bmatrix} n_x \\ n_y \end{Bmatrix} \quad |\mathbf{n}| = 1$

Heat flow out of the boundary (the flux)  
 $q_n = \mathbf{q} \cdot \mathbf{n} = \mathbf{q}^T \mathbf{n}$

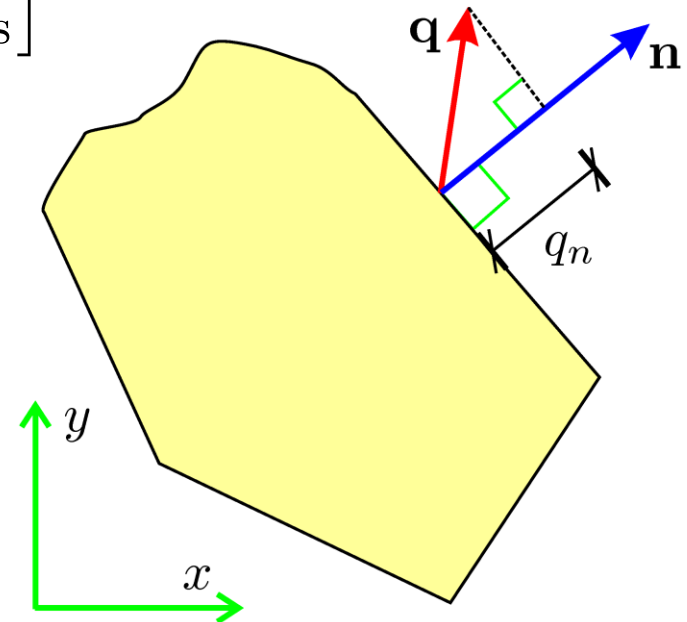
1D:  $q = -k \frac{dT}{dx}$

2D:  $q_x = -k_{xx} \frac{\partial T}{\partial x}$

$q_y = -k_{yy} \frac{\partial T}{\partial y}$

$k_{xx}, k_{yy}$  thermal conductivities [ $\text{J}/^\circ\text{Cms}$ ]

$T$  Temperature [ $^\circ\text{C}$ ]





# Finite Element Method

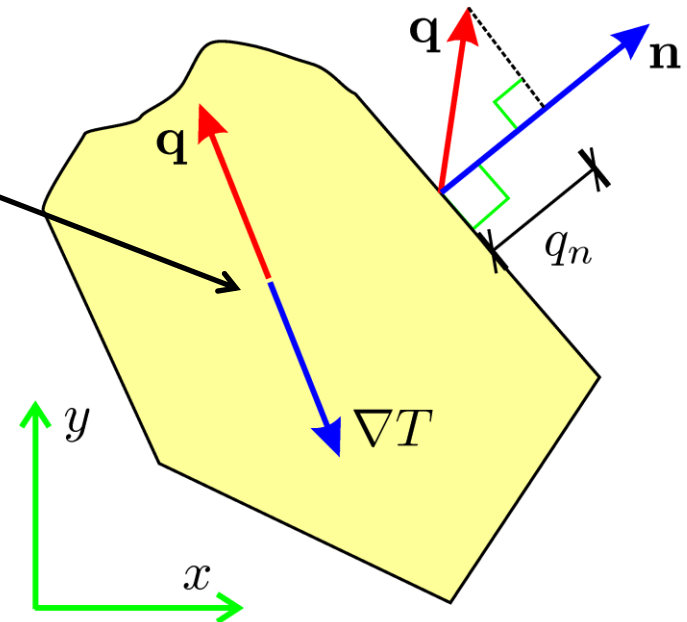
## 2D heat conduction

The constitutive relation in matrix notation:

$$\mathbf{q} = \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} = - \begin{bmatrix} k_{xx} & 0 \\ 0 & k_{yy} \end{bmatrix} \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{Bmatrix} = -\mathbf{D} \nabla T$$

$k_{xx} = k_{yy}$  , material isotropy leads to

$$\mathbf{D} = \begin{bmatrix} k_{xx} & 0 \\ 0 & k_{yy} \end{bmatrix}$$



$k_{xx}, k_{yy}$  thermal conductivities [J/°Cms]

$T$  Temperature [°C]

# Finite Element Method

## 2D heat conduction

Energy conservation and a time independent problem leads to: The amount of heat (energy) supplied to the body per unit of time must equal the amount of heat leaving the body per unit time:

$$\int_A Q t dA = \oint_{\mathcal{L}} q_n t d\mathcal{L}$$

$Q$  is internal heat supply [J/m<sup>3</sup>s]

$q_n = \mathbf{q}^T \mathbf{n}$

$$\oint_{\mathcal{L}} q_n t d\mathcal{L} = \oint_{\mathcal{L}} t \mathbf{q}^T \mathbf{n} d\mathcal{L} = \oint_{\mathcal{L}} (t \mathbf{q})^T \mathbf{n} d\mathcal{L} = \int_A \text{div}(t \mathbf{q}) dA$$

Gauss' divergence theorem (OP p. 73)

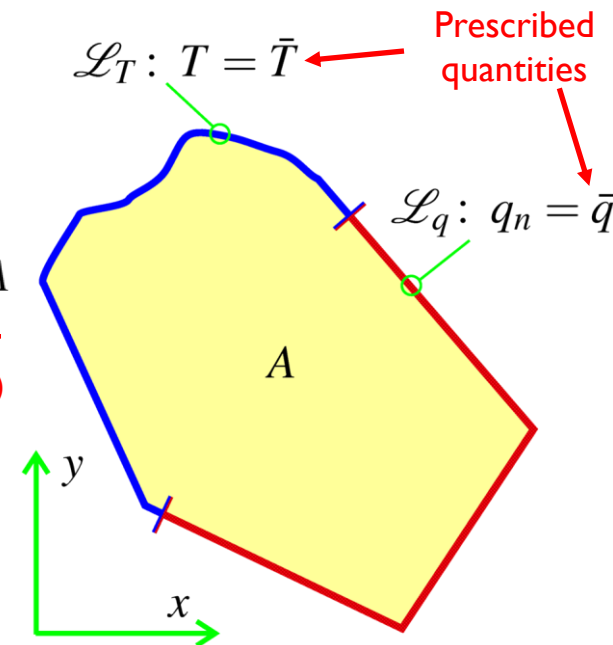
Divergence:

$$\text{div}(t \mathbf{q}) = t \frac{\partial q_x}{\partial x} + t \frac{\partial q_y}{\partial y}$$

$t$  is the thickness [m]

Rearranging leads to

$$\int_A (tQ - \text{div}(t \mathbf{q})) dA = 0 \quad \Leftrightarrow \quad tQ - \text{div}(t \mathbf{q}) = 0$$



# Finite Element Method

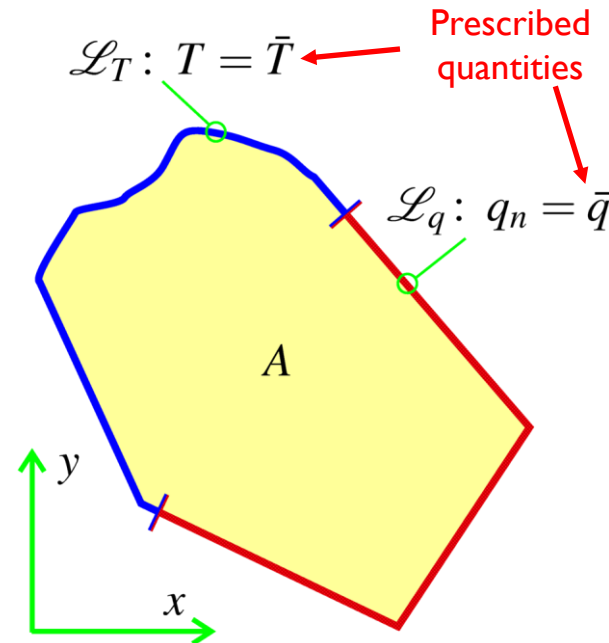
## 2D heat conduction

$$\mathbf{q} = -\mathbf{D}\nabla T$$

Insertion of the constitutive relation leads to the strong formulation:

$$\begin{aligned} \operatorname{div}(t\mathbf{D}\nabla T) + tQ &= 0 && \text{in region } A \\ q_n = \mathbf{q}^T \mathbf{n} &= \bar{q} && \text{on } \mathcal{L}_q \\ T &= \bar{T} && \text{on } \mathcal{L}_T \end{aligned}$$

Boundary conditions



Written out, the strong form for stationary 2D heat conduction is

$$\operatorname{div}(t\mathbf{D}\nabla T) + tQ = \frac{\partial}{\partial x} \left( tk_{xx} \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( tk_{yy} \frac{\partial T}{\partial y} \right) + tQ = 0$$

### Step 2: Establish weak formulation for 2D heat conduction (Cook pp. 136-137, 151), (OP Chapter 5, pp. 84-86)

Multiply the strong formulation with a weight function  $v(x,y)$  and integrate over the domain

$$\int_A v \operatorname{div}(t \mathbf{D} \nabla T) dA + \int_A v t Q dA = 0$$

$$\int_A v \operatorname{div}(t \mathbf{D} \nabla T) dA = \oint_{\mathcal{L}} v (t \mathbf{D} \nabla T)^T \mathbf{n} d\mathcal{L} - \int_A (\nabla v)^T t \mathbf{D} \nabla T dA$$

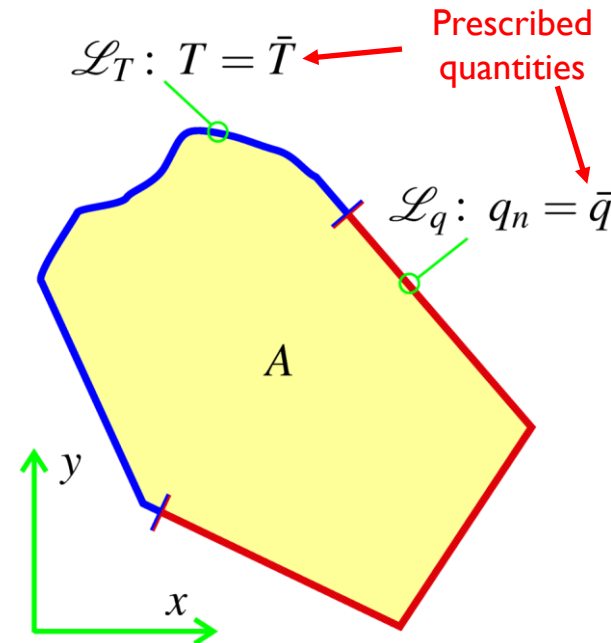
The Green-Gauss theorem (OP p. 74)

$$\mathbf{q} = -\mathbf{D} \nabla T$$

$$q_n = \mathbf{q}^T \mathbf{n}$$

Inserting into the first equation while replacing  $(t \mathbf{D} \nabla T)^T \mathbf{n} = t \mathbf{q}^T \mathbf{n} = t q_n$  yields

$$\oint_{\mathcal{L}} v t q_n d\mathcal{L} - \int_A (\nabla v)^T t \mathbf{D} \nabla T dA + \int_A v t Q dA = 0$$



# Finite Element Method

## 2D heat conduction

$$\oint_{\mathcal{L}} vtq_n d\mathcal{L} - \int_A (\nabla v)^T t \mathbf{D} \nabla T dA + \int_A vtQ dA = 0$$

The boundary integral is split into two terms to reflect the two different types of boundary conditions

$$\oint_{\mathcal{L}} vtq_n d\mathcal{L} = \oint_{\mathcal{L}_q} vt\bar{q} d\mathcal{L} + \oint_{\mathcal{L}_T} vtq_n d\mathcal{L}$$

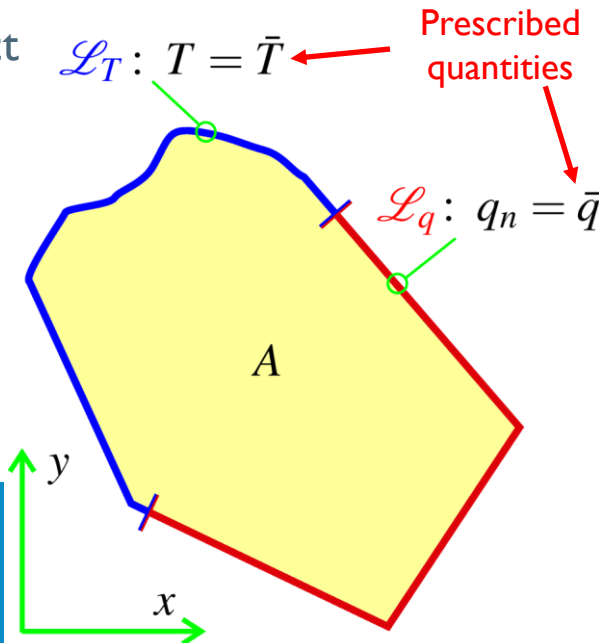
Insertion into the top equation followed by rearranging leads to the weak form of 2D heat flow

$$\int_A (\nabla v)^T t \mathbf{D} \nabla T dA = - \oint_{\mathcal{L}_q} vt\bar{q} d\mathcal{L} - \oint_{\mathcal{L}_T} vtq_n d\mathcal{L} + \int_A vtQ dA$$

$T = \bar{T}$  on  $\mathcal{L}_T$   
 $q_n = \bar{q}$  on  $\mathcal{L}_q$

Boundary conditions

Internal heat supply (heat load)



### Basic steps of the finite-element method (FEM)

1. Establish strong formulation
  - ◆ Partial differential equation
2. Establish weak formulation
  - ◆ Multiply with arbitrary field and integrate over element
3. Discretize over space
  - ◆ Mesh generation
4. Select shape and weight functions
  - ◆ Galerkin method
5. Compute element stiffness matrix
  - ◆ Local and global system
6. Assemble global system stiffness matrix
7. Apply nodal boundary conditions
  - ◆ temperature/flux/forces/forced displacements
8. Solve global system of equations
  - ◆ Solve for nodal values of the primary variables (displacements/temperature)
9. Compute temperature/stresses/strains etc. within the element
  - ◆ Using nodal values and shape functions

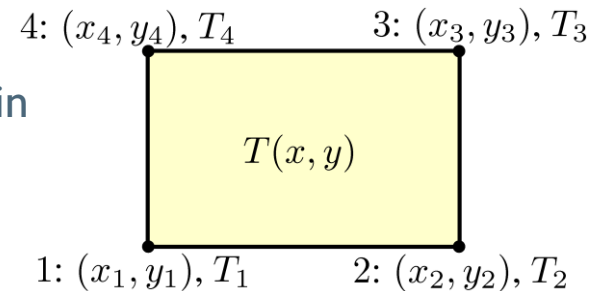
# Finite Element Method

## 2D heat conduction

### Step 3: Discretize over space

Now the domain is discretized into a number of finite elements. This determines the mesh coordinates and the element topology, i.e. the matrices `coord` and `Top` in our MatLab program.

Here it is illustrated with a single Mesh element.



# Finite Element Method

## 2D heat conduction

### Step 4: Weight and shape functions

The Galerkin method is chosen. This means that the main variable (the temperature) and the weight function  $v(x,y)$  are interpolated using the same interpolation functions

$$T(x,y) = N_1(x,y)T_1 + N_2(x,y)T_2 + N_3(x,y)T_3 + N_4(x,y)T_4$$

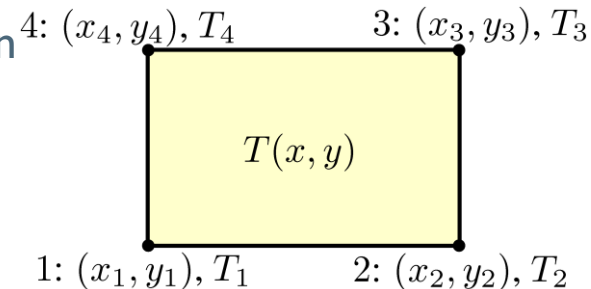
$$= \underbrace{[N_1(x,y) \ N_2(x,y) \ N_3(x,y) \ N_4(x,y)]}_{\text{Known shape functions (depends only on x and y)}} \underbrace{\begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix}}_{\text{Unknown nodal values}} = \mathbf{N}(x,y)\mathbf{a}$$

Known shape functions  
(depends only on x and y)

Unknown nodal values

$$v(x,y) = N_1(x,y)V_1 + N_2(x,y)V_2 + N_3(x,y)V_3 + N_4(x,y)V_4 = \mathbf{N}(x,y)\mathbf{v}$$

Examples:  $N_1(x_1,y_1) = 1, \quad N_1(x_2,y_2) = 0, \quad N_1(x_3,y_3) = 0, \quad N_1(x_4,y_4) = 0$   
 $N_2(x_1,y_1) = 0, \quad N_2(x_2,y_2) = 1, \quad N_2(x_3,y_3) = 0, \quad N_2(x_4,y_4) = 0$





The weak form can now be written as

$$\begin{aligned} \int_A (\nabla \mathbf{v})^T t \mathbf{D} \nabla T dA &= - \oint_{\mathcal{L}_q} v t \bar{q} d\mathcal{L} - \oint_{\mathcal{L}_T} v t q_n d\mathcal{L} + \int_A v t Q dA \quad \Leftrightarrow \\ \int_A (\nabla \mathbf{N} \mathbf{v})^T t \mathbf{D} \nabla \mathbf{N} \mathbf{a} dA &= - \oint_{\mathcal{L}_q} \mathbf{N} v t \bar{q} d\mathcal{L} - \oint_{\mathcal{L}_T} \mathbf{N} v t q_n d\mathcal{L} + \int_A \mathbf{N} v t Q dA \end{aligned}$$

Noticing that  $\mathbf{v}$  and  $\mathbf{a}$  are constants, rearranging yields

$$\mathbf{v}^T \left( \int_A \nabla \mathbf{N}^T t \mathbf{D} \nabla \mathbf{N} dA \mathbf{a} + \oint_{\mathcal{L}_q} \mathbf{N}^T t \bar{q} d\mathcal{L} + \oint_{\mathcal{L}_T} \mathbf{N}^T t q_n d\mathcal{L} - \int_A \mathbf{N} t Q dA \right) = 0$$

Because  $\mathbf{v}$  is arbitrary the parenthesis term must vanish (equal zero)

$$\begin{aligned} \int_A \nabla \mathbf{N}^T t \mathbf{D} \nabla \mathbf{N} dA \mathbf{a} + \oint_{\mathcal{L}_q} \mathbf{N}^T t \bar{q} d\mathcal{L} + \oint_{\mathcal{L}_T} \mathbf{N}^T t q_n d\mathcal{L} - \int_A \mathbf{N} t Q dA &= \mathbf{0} \quad \Leftrightarrow \\ \int_A \mathbf{B}^T t \mathbf{D} \mathbf{B} dA \mathbf{a} &= - \oint_{\mathcal{L}_q} \mathbf{N}^T t \bar{q} d\mathcal{L} - \oint_{\mathcal{L}_T} \mathbf{N}^T t q_n d\mathcal{L} + \int_A \mathbf{N} t Q dA \end{aligned}$$

where  $\mathbf{B} = \nabla \mathbf{N}$

# Finite Element Method

## 2D heat conduction

In order to be able to write the equations in a compact form the following terms are introduced

$$\mathbf{B} = \nabla \mathbf{N}$$

$$\mathbf{D} = \begin{bmatrix} k_{xx} & 0 \\ 0 & k_{yy} \end{bmatrix}$$

$$\mathbf{K} = \int_A \mathbf{B}^T t \mathbf{D} \mathbf{B} dA$$

The stiffness matrix

$$\mathbf{f}_b = - \oint_{\mathcal{L}_q} \mathbf{N}^T t \bar{q} d\mathcal{L} - \oint_{\mathcal{L}_T} \mathbf{N}^T t q_n d\mathcal{L}$$

Boundary terms (boundary load vector)

$$\mathbf{f}_l = \int_A \mathbf{N}^T t Q dA$$

Internal load vector

The equations can now be written as

$$\mathbf{K} \mathbf{a} = \mathbf{f}_b + \mathbf{f}_l \quad \text{or} \quad \mathbf{K} \mathbf{a} = \mathbf{f}, \quad \mathbf{f} = \mathbf{f}_b + \mathbf{f}_l$$

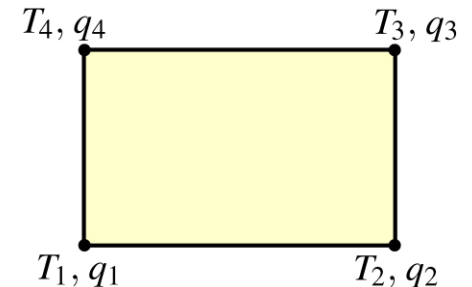
Example: The Melosh heat conduction element:

Degrees of freedom, dof

$$\mathbf{K}_{4 \times 4}, \quad \mathbf{a} = \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix}, \quad \mathbf{f} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix}$$

$$\mathbf{N} = [N_1 \ N_2 \ N_3 \ N_4]$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y} \end{bmatrix}$$



# Finite Element Method

## 2D heat conduction

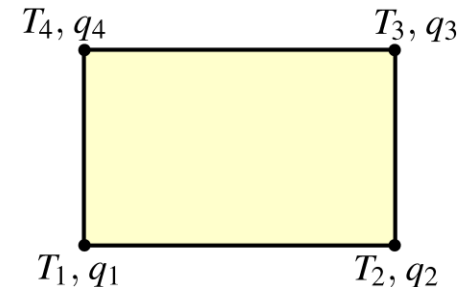
The definition of the heat flux was

$$\mathbf{q}(x,y) = \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} = - \begin{bmatrix} k_{xx} & 0 \\ 0 & k_{yy} \end{bmatrix} \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{Bmatrix} = -\mathbf{D}\nabla T(x,y)$$

With the finite element formulation we have  $T(x,y) = \mathbf{N}(x,y)\mathbf{a}$

This gives us the flux within the element as

$$\begin{aligned} \mathbf{q}(x,y) &= -\mathbf{D}\nabla T(x,y) = -\mathbf{D}\nabla(\mathbf{N}\mathbf{a}) \\ &= -\mathbf{D}\nabla\mathbf{N}\mathbf{a} = -\mathbf{D}\mathbf{B}\mathbf{a} \end{aligned}$$



$$\mathbf{N} = [N_1 \ N_2 \ N_3 \ N_4]$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y} \end{bmatrix}$$

# Finite Element Method

## 2D heat conduction

### Shape and weight functions

Example: The melosh element

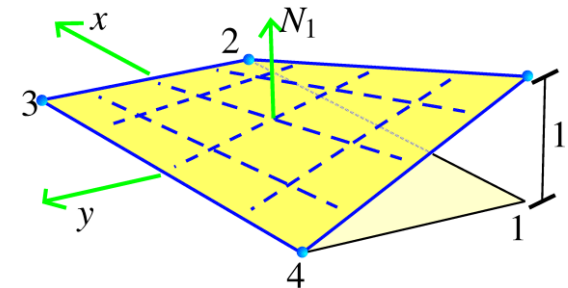
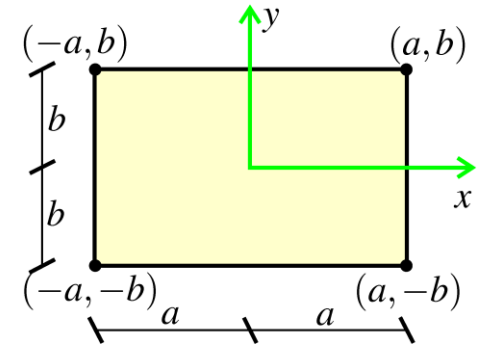
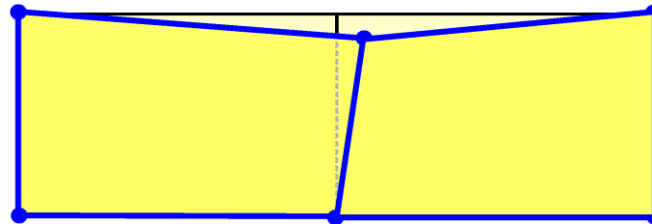
$$N_1 = \frac{(a-x)(b-y)}{4ab} \quad N_2 = \frac{(a+x)(b-y)}{4ab}$$

$$N_3 = \frac{(a+x)(b+y)}{4ab} \quad N_4 = \frac{(a-x)(b+y)}{4ab}$$

Notice that  $N_i$  vary linearly along element edges.

Example:  $N_1(-a, y) = \frac{2a(b-y)}{4ab}$

This ensures inter-element compatibility, i.e. no gaps and no overlapping:



# Finite Element Method

## 2D heat conduction

### Exercise: compute and program the Melosh stiffness matrix

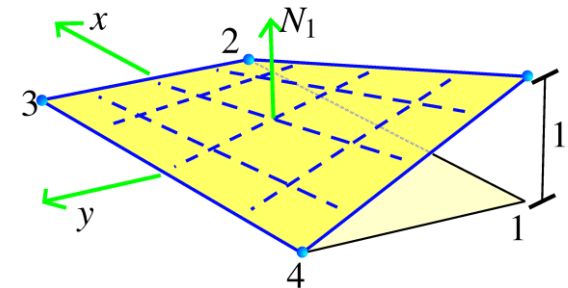
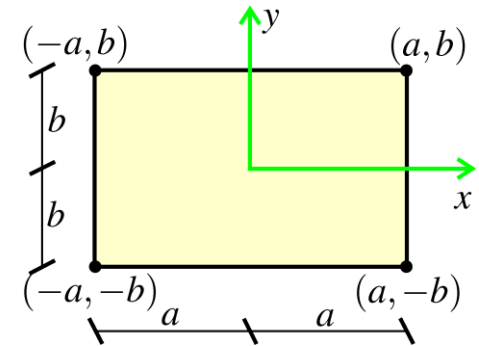
Assumptions:  $\mathbf{D}$  and  $t$  are constant.

Element width:  $2a$ , height  $2b$

Steps:

1. Compute  $\mathbf{B}$ -matrix
2. Carry out matrix multiplication
3. Integrate each element of the matrix product to obtain  $\mathbf{K}$
4. Program  $\mathbf{K}$  into the file KmeloshHeat.m
5. A test value with the parameters,  $2a = 2$ ,  $2b = 1$ ,  $t = 1.3$ ,  $k_{xx} = 4.56$ ,  $k_{yy} = 3.8$  is :

$$\mathbf{K} = \begin{bmatrix} 4.2813 & 0.6587 & -2.1407 & -2.7993 \\ 0.6587 & 4.2813 & -2.7993 & -2.1407 \\ -2.1407 & -2.7993 & 4.2813 & 0.6587 \\ -2.7993 & -2.1407 & 0.6587 & 4.2813 \end{bmatrix}$$



### Boundary load vector

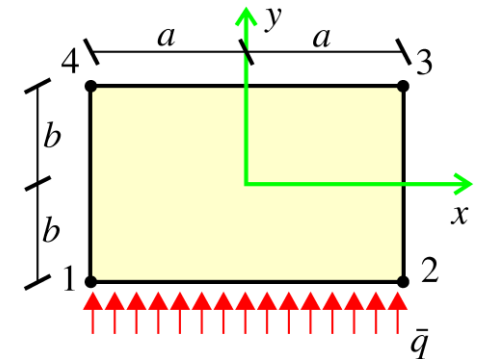
Can only contain values at nodes. So uniform heat supplies must be converted into nodal loads

$$\mathbf{f}_b = - \underbrace{\oint_{\mathcal{L}_q} \mathbf{N}^T t \bar{q} d\mathcal{L}}_{\text{Prescribed term}} - \underbrace{\oint_{\mathcal{L}_T} \mathbf{N}^T t q_n d\mathcal{L}}_{\text{Reactions}}$$

Can only contain nodal values

Unit: J/s

Example: constant heat supply between nodes 1 and 2



$$\begin{aligned} \mathbf{f}_b &= - \int_{-a}^a \mathbf{N}(x, -b)^T t \bar{q} d\mathcal{L} = - \begin{Bmatrix} \int_{-a}^a N_1(x, -b) dx \\ \int_{-a}^a N_2(x, -b) dx \\ \int_{-a}^a N_3(x, -b) dx \\ \int_{-a}^a N_4(x, -b) dx \end{Bmatrix} t \bar{q} = - \begin{Bmatrix} \int_{-a}^a (a-x) 2b dx \\ \int_{-a}^a (a+x) 2b dx \\ 0 \\ 0 \end{Bmatrix} \frac{t \bar{q}}{4ab} \\ &= - \begin{Bmatrix} [(ax - \frac{1}{2}x^2) 2b]_{-a}^a \\ [(ax + \frac{1}{2}x^2) 2b]_{-a}^a \\ 0 \\ 0 \end{Bmatrix} \frac{t \bar{q}}{4ab} = - \begin{Bmatrix} 2a^2 2b \\ 2a^2 2b \\ 0 \\ 0 \end{Bmatrix} \frac{t \bar{q}}{4ab} = - \begin{Bmatrix} a \\ a \\ 0 \\ 0 \end{Bmatrix} \frac{t \bar{q}}{2} \end{aligned}$$

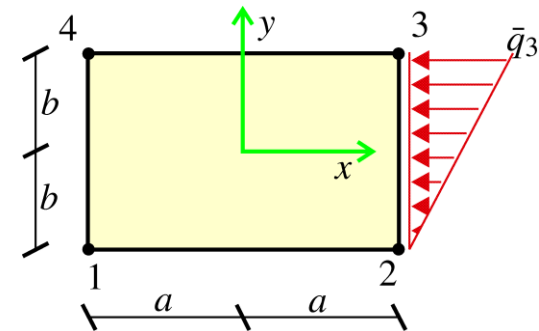
Remember: If heat is transferred *into* the domain,  $\bar{q}$  is negative

# Finite Element Method

## 2D heat conduction

### Exercise: Calculate $\mathbf{f}_b$ with triangular heat load

The heat load varies linearly from zero at node 2 to the value  $\bar{q}_3$  at node 3



$$\mathbf{f}_b = - \underbrace{\oint_{\mathcal{L}_q} \mathbf{N}^T t \bar{q} d\mathcal{L}}_{\text{Prescribed term}} - \oint_{\mathcal{L}_T} \mathbf{N}^T t q_n d\mathcal{L}$$

Can only contain nodal values

Steps:

1. Find the expression for the heat load as a function of the coordinates  $\bar{q}(x,y)$
2. Carry out the integration as shown on the previous slide.

$$N_1 = \frac{(a-x)(b-y)}{4ab}$$

$$N_2 = \frac{(a+x)(b-y)}{4ab}$$

$$N_3 = \frac{(a+x)(b+y)}{4ab}$$

$$N_4 = \frac{(a-x)(b+y)}{4ab}$$

Remember: If heat is transferred *into* the domain,  $\bar{q}$  is negative

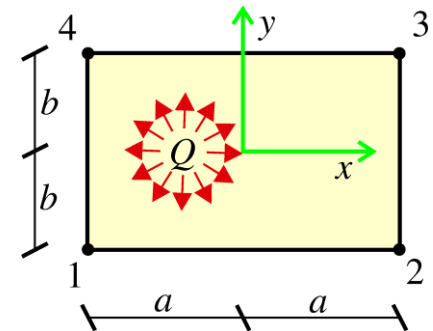
# Finite Element Method

## 2D heat conduction

### Internal load vector

Can only contain values at nodes. So internal heat supplies must be converted into nodal loads

$$\mathbf{f}_l = \int_A \mathbf{N}^T tQ dA$$



Example: constant internal heat supply in a Melosh element

$$\begin{aligned} \mathbf{f}_l &= \int_{-a}^a \int_{-b}^b \mathbf{N}(x,y)^T tQ dy dx = \int_{-a}^a \int_{-b}^b \begin{Bmatrix} (a-x)(b-y) \\ (a+x)(b-y) \\ (a+x)(b+y) \\ (a-x)(b+y) \end{Bmatrix} dy dx \frac{tQ}{4ab} = \int_{-a}^a \begin{Bmatrix} (a-x)2b^2 \\ (a+x)2b^2 \\ (a+x)2b^2 \\ (a-x)2b^2 \end{Bmatrix} dx \frac{tQ}{4ab} \\ &= \begin{Bmatrix} 2a^2 \\ 2a^2 \\ 2a^2 \\ 2a^2 \end{Bmatrix} 2b^2 \frac{tQ}{4ab} = \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix} 2a^2 2b^2 \frac{tQ}{4ab} = \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix} abtQ = \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix} \frac{1}{4} tAQ \end{aligned}$$

Element area  $A = 4ab$



### Step 6: Assembling

- ☐ The assembling procedure is exactly the same as in the last lecture:
  1. Determine the local stiffness matrix
  2. determine the global number of dof corresponding to the local dof for the element
  3. add the components of the local stiffness matrix to the rows and columns of the global stiffness matrix corresponding to the global dof numbers
  4. repeat 1-3 until all contributions from all elements have been added.
- ☐ In MatLab this is done in `assembling.m`

$$K(gDof,gDof) = K(gDof,gDof) + K_e;$$

# Finite Element Method

## 2D heat conduction

- ❑ In order to be able to create a global system matrix we need to give information about
  - ◆ Material for each element
  - ◆ Section dimensions for each element
  - ◆ Coordinates for each node (and a numbering)
  - ◆ Topology for each element (which nodes are in the element)
    - ❑ dof numbering is given from the node numbering and number of dofs per node (one in this case)
- ❑ See `calc_globdof.m` for numbering of global dof
  - ◆ `ElemDof = [neDof1 GDof1 GDof2 ...]`
    - ❑ dof numbering of each element
  - ◆ `GlobDof = [Gnode1 GDof1 GDof2 ...]`
    - ❑ dof numbering of each node (used for plotting)
  - ◆ `nDof` = total number of Dof

# Finite Element Method

## 2D heat conduction

### Example of assembling the load vector

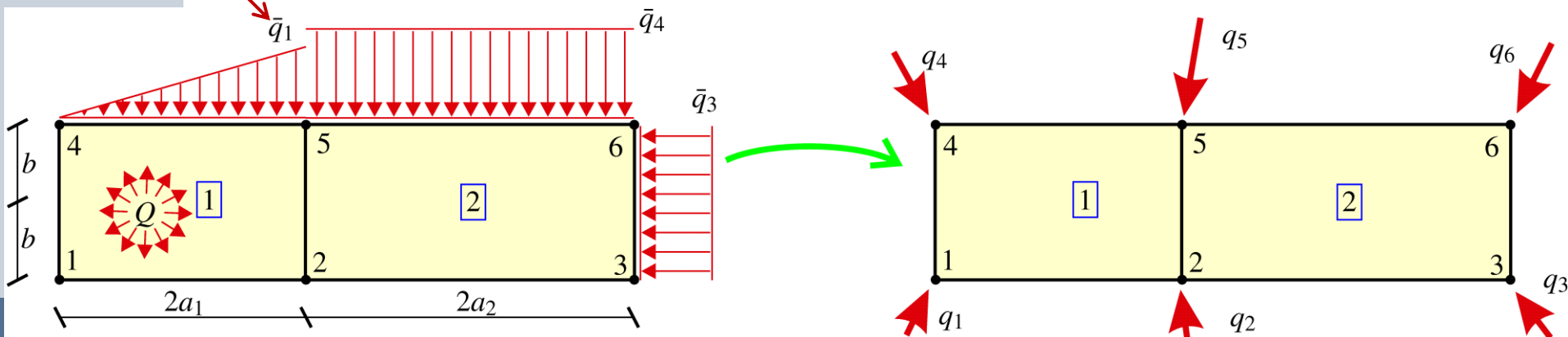
The global matrices are assembled in exactly the same way as with the bar elements of the previous lecture. Here an example of the assembly of the load vector of a two element system is shown

$$\mathbf{f} = \mathbf{f}_b + \mathbf{f}_l \quad \text{Total heat load vector}$$

$$\mathbf{f} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix} = - \begin{Bmatrix} 0 \\ 0 \\ \frac{1}{2}2b\bar{q}_3 \\ \frac{1}{3}2a_1\bar{q}_1 \\ \frac{2}{3}2a_1\bar{q}_1 + \frac{1}{2}2a_2\bar{q}_2 \\ \frac{1}{2}2a_2\bar{q}_2 + \frac{1}{2}2b\bar{q}_3 \end{Bmatrix} t + \begin{Bmatrix} Qa_1b \\ Qa_1b \\ 0 \\ Qa_1b \\ Qa_1b \\ 0 \end{Bmatrix} t$$

The thicknesses of the two elements are identical

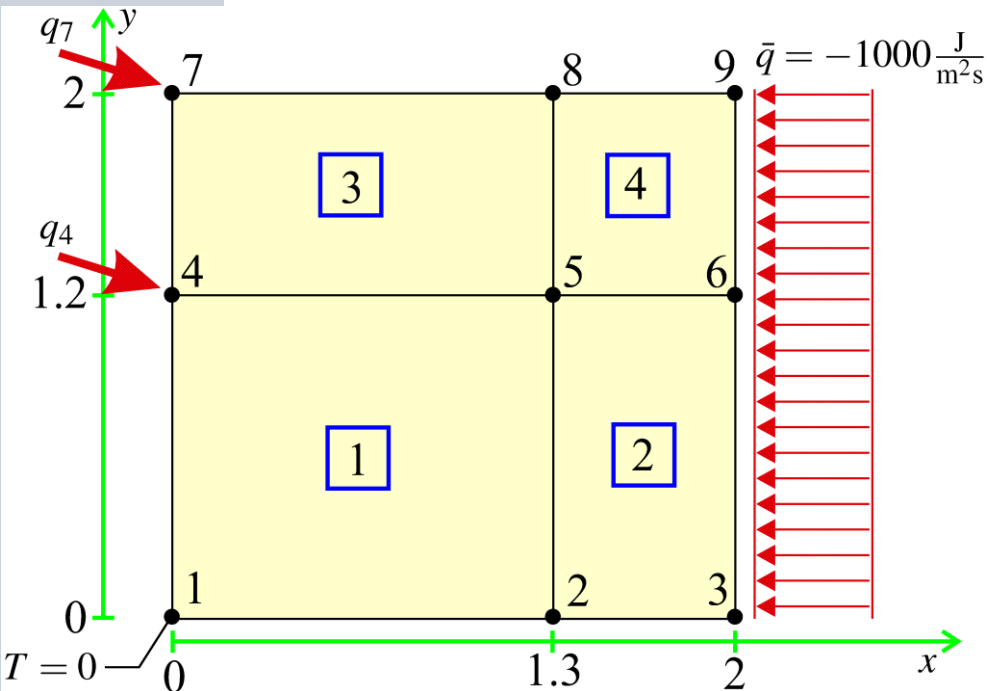
Remember: If heat is transferred into the domain,  $\bar{q}$  is negative



# Finite Element Method

## 2D heat conduction

### Exercise: Program and plot the results of a 2D patch test for the Melosh element



$t = 1.2 \text{ m}$ ,  $k_{xx} = k_{yy} = 3 \text{ J/Cms}$

#### Steps

1. Program the geometry (done earlier today)
2. Determine nodal heat supplies  $q_3$ ,  $q_6$ , and  $q_9$
3. Determine nodal heat supply  $q_4$  and  $q_7$  (We want  $T = 0$  on the left boundary,  $x = 0$ )
4. Make the appropriate programming in Topology.m and BoundaryConditions.m
5. Plot by executing the line “visualizeD2

Compare with the analytical solution:  $T(x) = -\frac{\bar{q}}{k_{xx}}x + T_{x=0}$

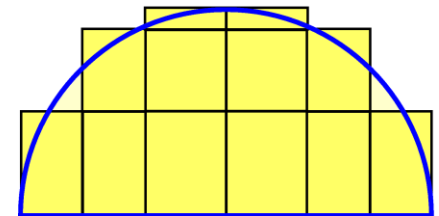
Hint: on the colorplot the temperatures can be found by activating the figure and typing “colorbar” at the command line

# Finite Element Method

## 2D heat conduction

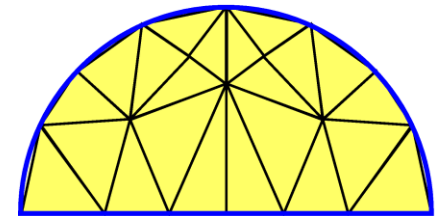
### Why use other elements than the Melosh element?

- The Melosh element must be rectangular and positioned along the coordinate axes
- It is therefore not very good at approximating boundaries not aligned with the coordinate axes
- Conclusion: The Melosh element is not very flexible



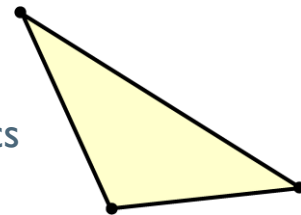
### Why use triangular elements?

- Triangular elements can be rotated arbitrarily
- They can therefore approximate boundaries not aligned with coordinate axes well.
- It is relatively simple to make a computer code that meshes an arbitrary area with triangles.



### Next: The constant strain triangle (CST)

- The name stems from its original development within structural mechanics
- In heat conduction analysis the name “constant flux triangle” would be appropriate



# Finite Element Method

## 2D heat conduction

### Preliminary: Natural coordinates (Area coordinates)

- For general triangular elements it has proven useful to derive the stiffness matrix using the so-called “natural coordinates” also referred to as “area coordinates

Point P defines three areas. These define the natural coordinate of the point

$$\xi_1 = \frac{A_1}{A}, \quad \xi_2 = \frac{A_2}{A}, \quad \xi_3 = \frac{A_3}{A}$$

The following constraint applies

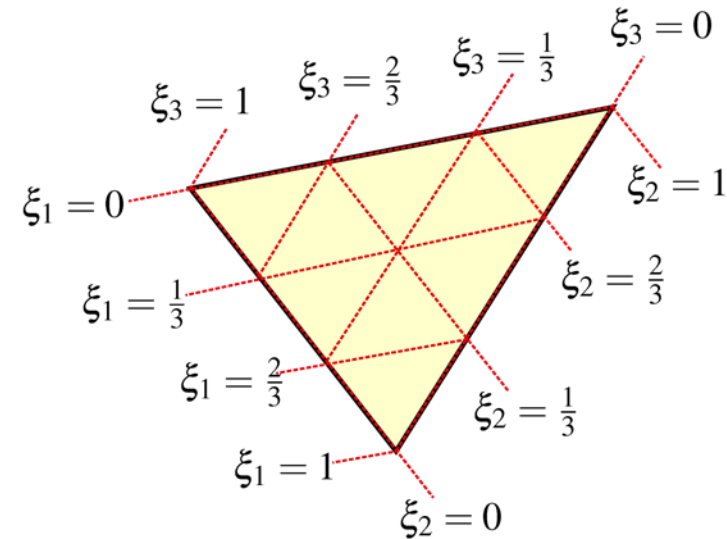
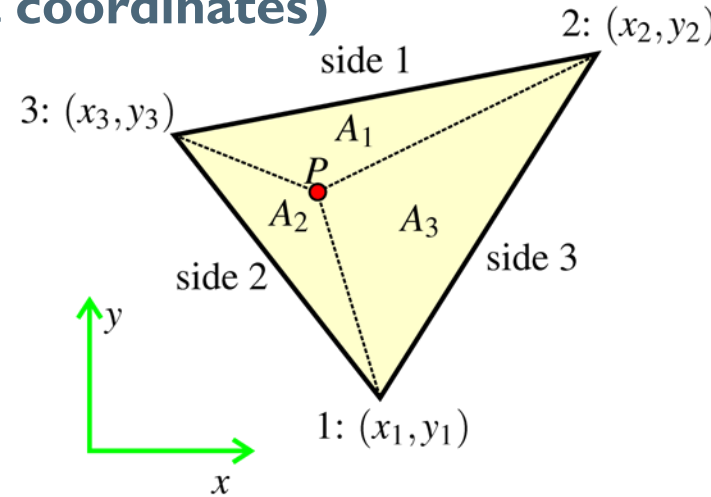
$$A = A_1 + A_2 + A_3 \Rightarrow \xi_1 + \xi_2 + \xi_3 = 1$$

#### Examples:

Element centroid:  $(\xi_1, \xi_2, \xi_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

Node 1:  $(\xi_1, \xi_2, \xi_3) = (1, 0, 0)$

Node 3:  $(\xi_1, \xi_2, \xi_3) = (0, 0, 1)$



# Finite Element Method

## 2D heat conduction

The relation between natural and cartesian coordinates is

$$x = x_1 \xi_1 + x_2 \xi_2 + x_3 \xi_3$$

$$y = y_1 \xi_1 + y_2 \xi_2 + y_3 \xi_3$$

In matrix notation

$$\begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} = \mathbf{A} \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} \quad \text{and} \quad \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} = \mathbf{A}^{-1} \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix}$$

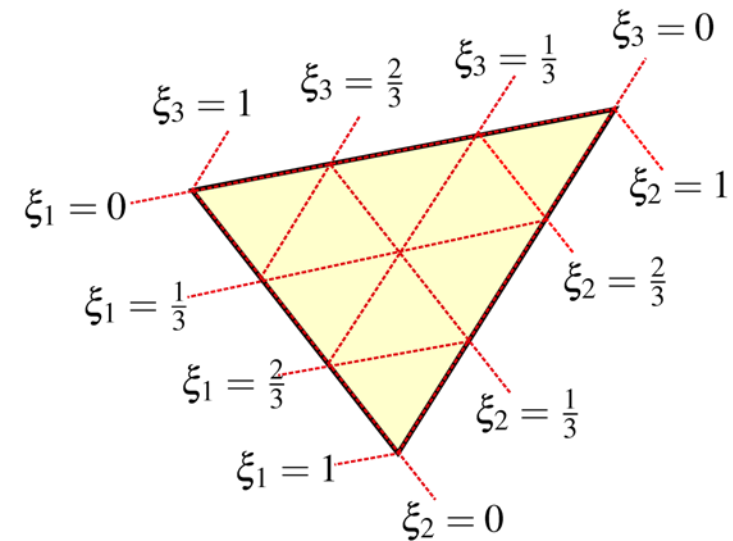
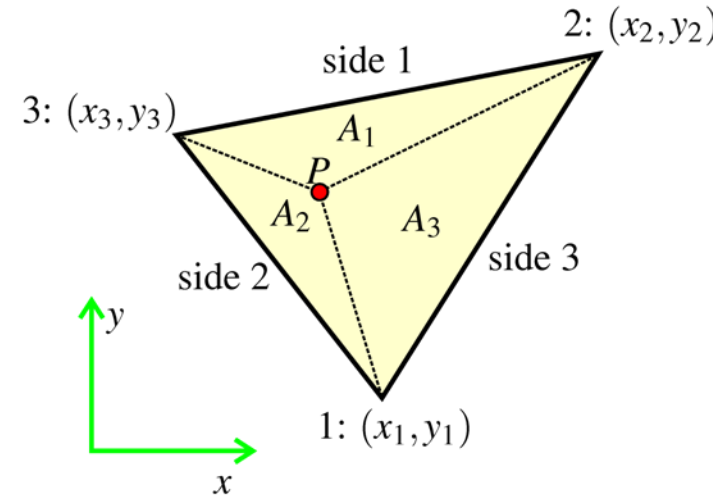
With the coordinate transformation matrix  $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \quad \text{and} \quad \mathbf{A}^{-1} = \frac{1}{2A} \begin{bmatrix} x_2 y_3 - x_3 y_2 & y_{23} & x_{32} \\ x_3 y_1 - x_1 y_3 & y_{31} & x_{13} \\ x_1 y_2 - x_2 y_1 & y_{12} & x_{21} \end{bmatrix}$$

here  $x_{jk} = x_j - x_k$  and  $y_{jk} = y_j - y_k$

The element area can be found by

$$2A = \det \mathbf{A} = x_{21} y_{31} - x_{31} y_{21}$$



# Finite Element Method

## 2D heat conduction

### Differentiation in natural coordinates

A function  $N$  (e.g. a shape function) is expressed in natural coordinates,  $N(\xi_1, \xi_2, \xi_3)$ . The chain rule then gives us

$$\frac{\partial N}{\partial x} = \frac{\partial N}{\partial \xi_1} \frac{\partial \xi_1}{\partial x} + \frac{\partial N}{\partial \xi_2} \frac{\partial \xi_2}{\partial x} + \frac{\partial N}{\partial \xi_3} \frac{\partial \xi_3}{\partial x}$$

$$\frac{\partial N}{\partial y} = \frac{\partial N}{\partial \xi_1} \frac{\partial \xi_1}{\partial y} + \frac{\partial N}{\partial \xi_2} \frac{\partial \xi_2}{\partial y} + \frac{\partial N}{\partial \xi_3} \frac{\partial \xi_3}{\partial y}$$

With the coordinate transformation

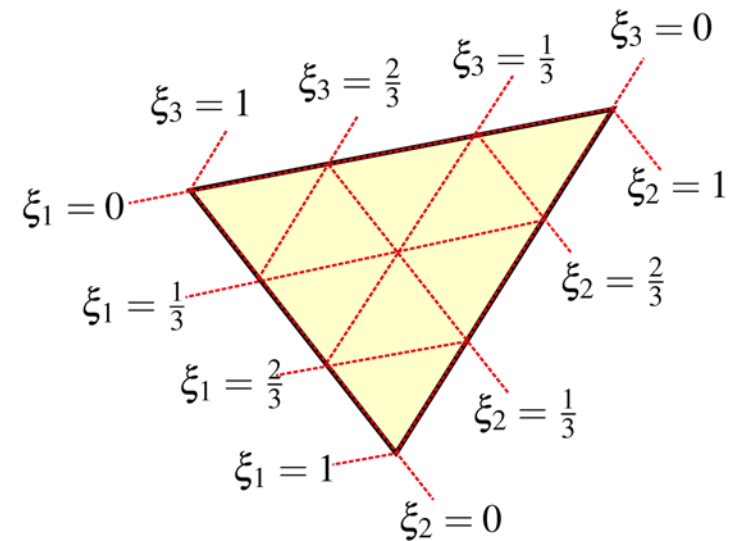
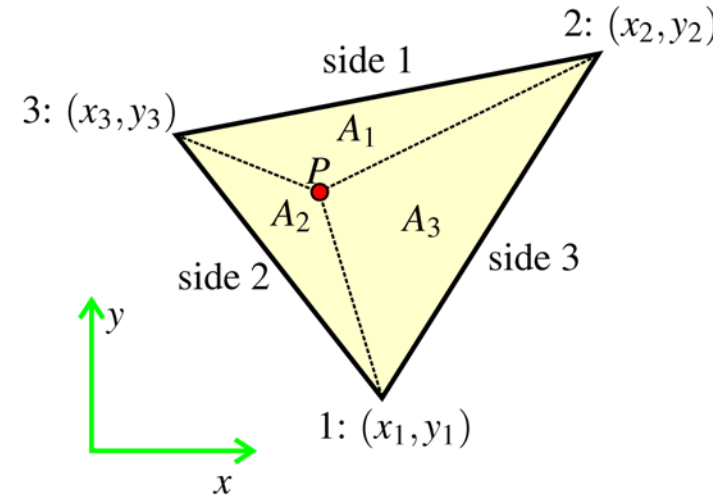
$$\begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} x_2y_3 - x_3y_2 & y_{23} & x_{32} \\ x_3y_1 - x_1y_3 & y_{31} & x_{13} \\ x_1y_2 - x_2y_1 & y_{12} & x_{21} \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix}$$

$\mathbf{A}^{-1}$

we get

$$\frac{\partial \xi_1}{\partial x} = \frac{y_{23}}{2A}, \quad \frac{\partial \xi_2}{\partial x} = \frac{y_{31}}{2A}, \quad \frac{\partial \xi_3}{\partial x} = \frac{y_{12}}{2A}$$

$$\frac{\partial \xi_1}{\partial y} = \frac{x_{32}}{2A}, \quad \frac{\partial \xi_2}{\partial y} = \frac{x_{13}}{2A}, \quad \frac{\partial \xi_3}{\partial y} = \frac{x_{21}}{2A}$$





# Finite Element Method

## 2D heat conduction

### Integration in natural coordinates

A polynomial function expressed in natural coordinates,  $f(\xi_1, \xi_2, \xi_3)$  can easily be integrated over the element area using the formula

$$\int_A \xi_1^k \xi_2^\ell \xi_3^m dA = 2A \frac{k! \ell! m!}{(2+k+\ell+m)!}$$

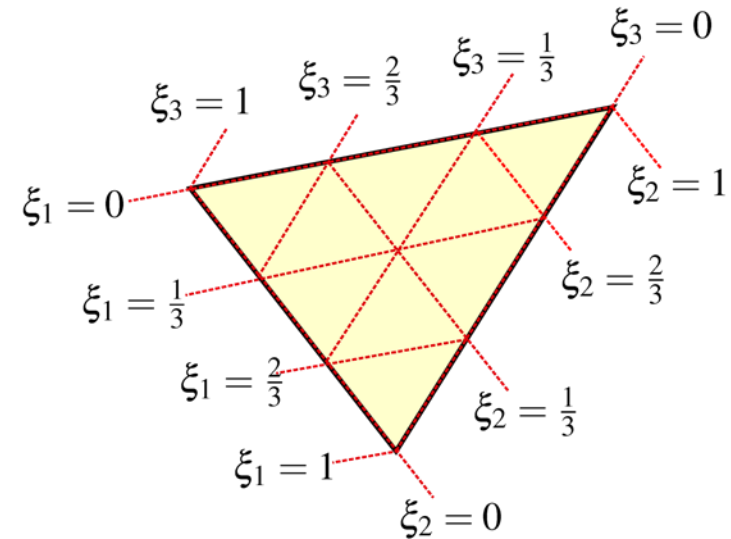
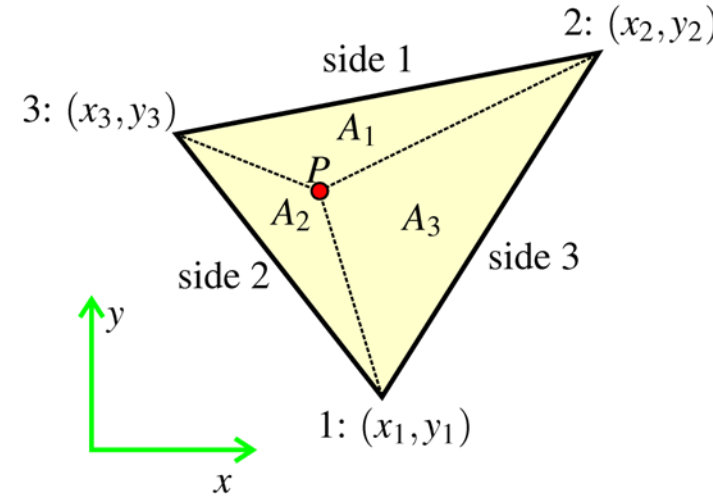
Example:  $f(\xi_1, \xi_2, \xi_3) = (2 + \xi_1)(1 - \xi_3^2)$

$$\int_A f dA = \int_A (2 - 2\xi_3^2 + \xi_1 - \xi_1 \xi_3^2) dA$$

$\begin{matrix} k=0 & k=0 & k=1 & k=1 \\ \ell=0 & \ell=0 & \ell=0 & \ell=0 \\ m=0 & m=2 & m=0 & m=2 \end{matrix}$

$$= 2A \left( \frac{1 \cdot 1 \cdot 1}{(2+0+0+0)!} - 2 \frac{1 \cdot 1 \cdot 2}{(2+0+0+2)!} + \frac{1 \cdot 1 \cdot 1}{(2+1+0+0)!} - \frac{1 \cdot 1 \cdot 2}{(2+1+0+2)!} \right)$$

$$= 2A \left( \frac{1}{2} - \frac{4}{24} + \frac{1}{6} - \frac{2}{120} \right) = \frac{29}{30} A$$



# Finite Element Method

## 2D heat conduction

### Shape functions for the CST-element

The shape functions for the CST-element in natural coordinates are very simple:

$$N_1 = \xi_1, \quad N_2 = \xi_2, \quad N_3 = \xi_3$$

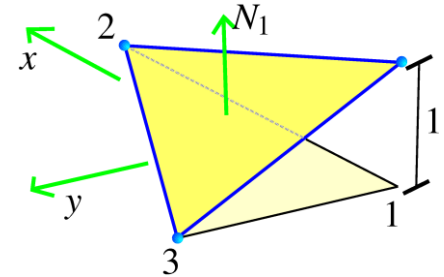
The temperature in the element interior is then given by

$$T(\xi_1, \xi_2, \xi_3) = [N_1 \ N_2 \ N_3] \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} = \mathbf{N}\mathbf{a}$$

The stiffness matrix is again given by  $\mathbf{K} = \int_A \mathbf{B}^T t \mathbf{D} \mathbf{B} dA$

with

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} \end{bmatrix} \quad \text{and} \quad \mathbf{D} = \begin{bmatrix} k_{xx} & 0 \\ 0 & k_{yy} \end{bmatrix}$$



# Finite Element Method

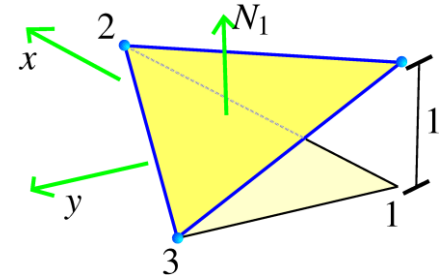
## 2D heat conduction

### Exercise: Calculate and program the stiffness matrix of the heat conduction CST-element

#### Steps

1. Determine the derivatives in the **B**-matrix
2. Setup the **B**-matrix. Think about and explain why the element name “constant flux element” is appropriate (hint on slide 18)
3. Carry out the matrix multiplication indicated on the previous slide.
4. Integrate over the element area to obtain the stiffness matrix **K**
5. Program the stiffness matrix into the file K\_CST\_Heat.m
6. A test value with the parameters  $(x_1, x_2, x_3) = (1, 3, -1)$ ,  $(y_1, y_2, y_3) = (0, 2, 1)$ ,  $t = 1.3$ ,  $k_{xx} = 4.7$  and  $k_{yy} = 5.1$  is

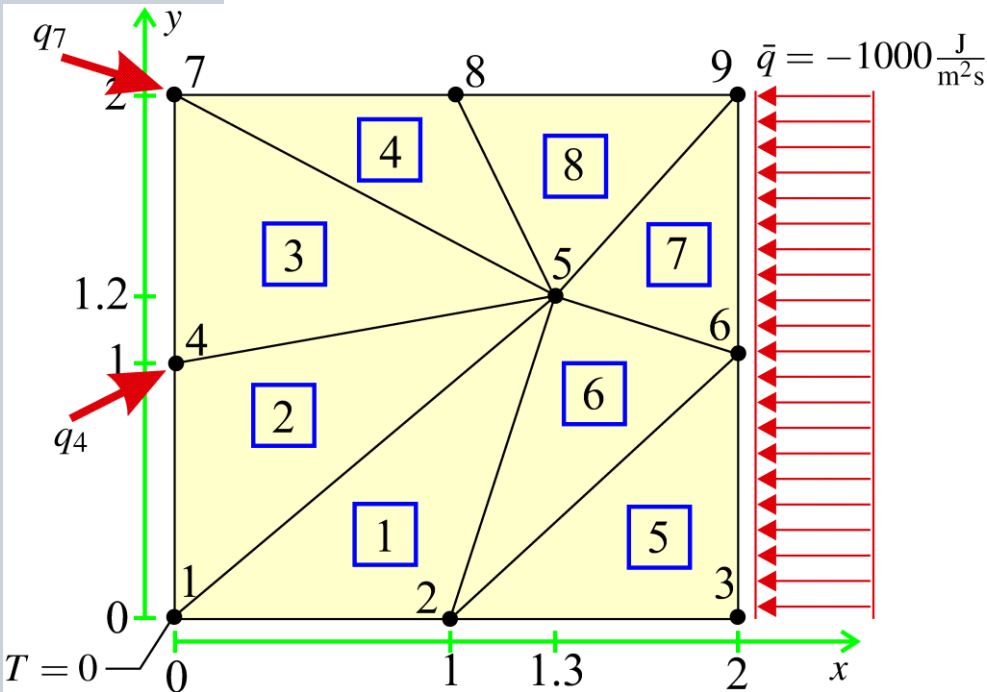
$$\mathbf{K} = \begin{bmatrix} 9.3492 & -3.9108 & -5.4383 \\ -3.9108 & 2.7192 & 1.1917 \\ -5.4383 & 1.1917 & 4.2467 \end{bmatrix}$$



# Finite Element Method

## 2D heat conduction

### Exercise: Program and plot the results of a 2D patch test for the CST element



$t = 1.2 \text{ m}, \quad k_{xx} = k_{yy} = 3 \text{ W/ Cm}$

#### Steps

1. Program the geometry (done earlier today)
2. Determine nodal heat supplies  $q_3$ ,  $q_6$ , and  $q_9$
3. Determine nodal heat supply  $q_4$  and  $q_7$  (We want  $T = 0$  on the left boundary,  $x = 0$ )
4. Make the appropriate programming in Topology.m and BoundaryConditions.m
5. Plot by executing the line “visualizeD2

Compare with the analytical solution:  $T(x) = -\frac{\bar{q}}{k_{xx}}x + T_{x=0}$

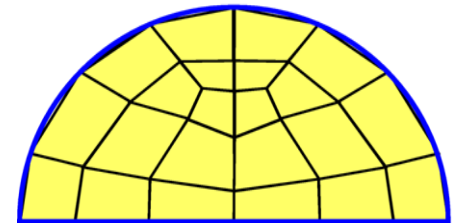
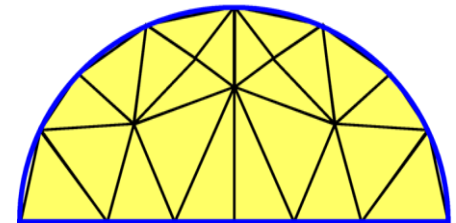
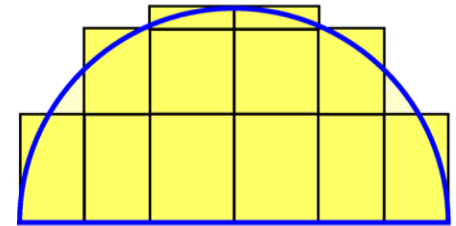
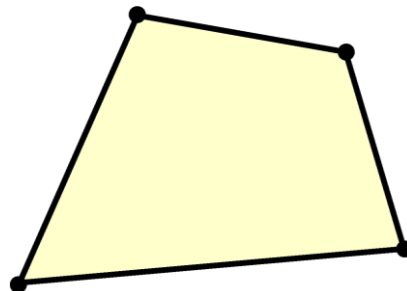
Hint: on the colorplot the temperatures can be found by activating the figure and typing “colorbar” at the command line

# Finite Element Method

## 2D heat conduction

### The isoparametric four node element (Iso4)

- A quadrilateral element which can be distorted from the rectangular shape and rotated arbitrarily in the plane
- Introduces several important concepts in finite element theory, such as
  - Isoparametric coordinates
  - Parent and global domain
  - The Jacobian matrix and the Jacobian
  - Numerical integration by Gauss quadrature
- The isoparametric formulation forms the basis for nearly all the elements in practical use



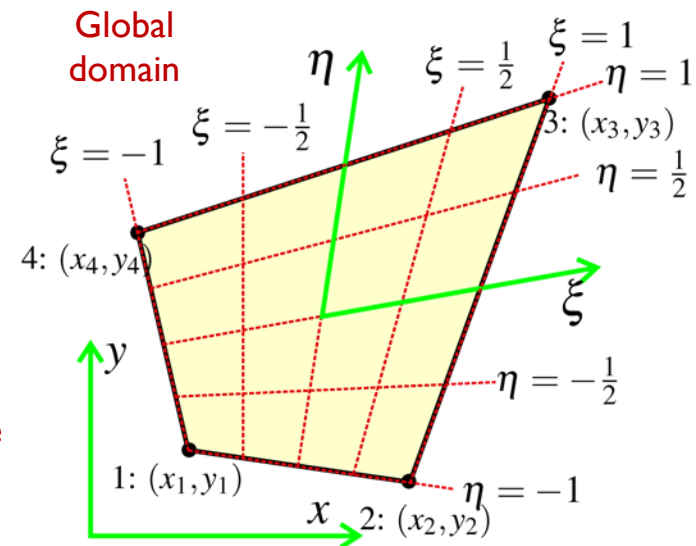
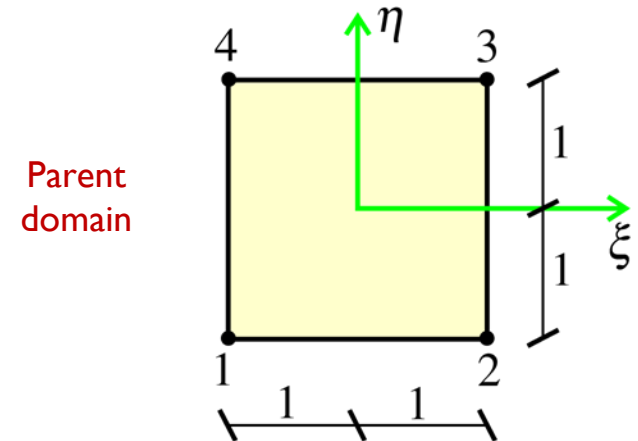
# Finite Element Method

## 2D heat conduction

### Parent and global domain

- For general quadrilateral elements it has proven useful to formulate the stiffness matrix using the isoparametric formulation.
- In the parent domain the element is always a square element with the side length 2.
- The isoparametric coordinates  $\xi$  and  $\eta$  have their origin at the element centroid. This means that the vertex (corner) nodes have the coordinates  $(\xi, \eta)$ : 1:  $(-1, -1)$ , 2:  $(1, -1)$ , 3:  $(1, 1)$  and 4:  $(-1, 1)$
- In the global domain the nodes of the four node element have the coordinates:  $(x, y)$ : 1:  $(x_1, y_1)$ , 2:  $(x_2, y_2)$ , 3:  $(x_3, y_3)$ , and 4:  $(x_4, y_4)$ .
- The question is “how do we connect these two sets of coordinates?” (much in a similar way as we did with the triangle.)

**Note:** with the natural coordinates of the triangle the coordinate limits were  $0 \leq \xi_1, \xi_2, \xi_3 \leq 1$ . For the isoparametric coordinates they are  $-1 \leq \xi, \eta \leq 1$



# Finite Element Method

## 2D heat conduction

### The isoparametric coordinate transformation

- The idea is that we use the shape functions to interpolate the coordinates between the nodes

$$x = \mathbf{N}(\xi, \eta) \mathbf{x} \quad \text{and} \quad y = \mathbf{N}(\xi, \eta) \mathbf{y}$$

$$\mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix}, \quad \mathbf{y} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix}, \quad \text{and} \quad \mathbf{N}(\xi, \eta) = [N_1 \ N_2 \ N_3 \ N_4]$$

e.g.

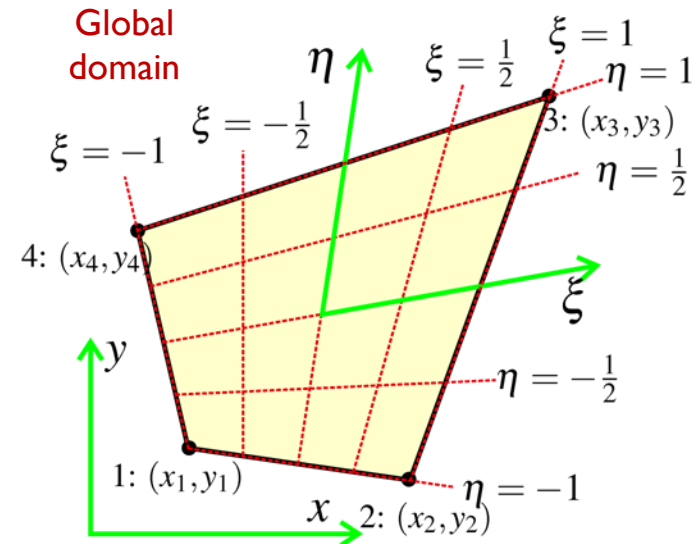
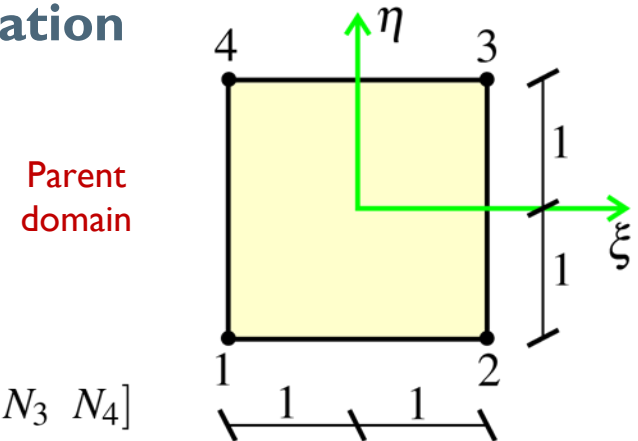
$$x = N_1 x_1 + N_2 x_2 + N_3 x_3 + N_4 x_4$$

With the shape functions:

$$N_1 = \frac{1}{4}(1 - \xi)(1 - \eta) \quad N_2 = \frac{1}{4}(1 + \xi)(1 - \eta)$$

$$N_3 = \frac{1}{4}(1 + \xi)(1 + \eta) \quad N_4 = \frac{1}{4}(1 - \xi)(1 + \eta)$$

Notice that these are the shape functions of the Melosh element with  $a = b = 1$



# Finite Element Method

## 2D heat conduction

### Temperature and flux interpolation

As usual the temperature within the element is interpolated from the nodal values

$$T(x,y) = \mathbf{N}\mathbf{a}$$

The heat flux is also still given by

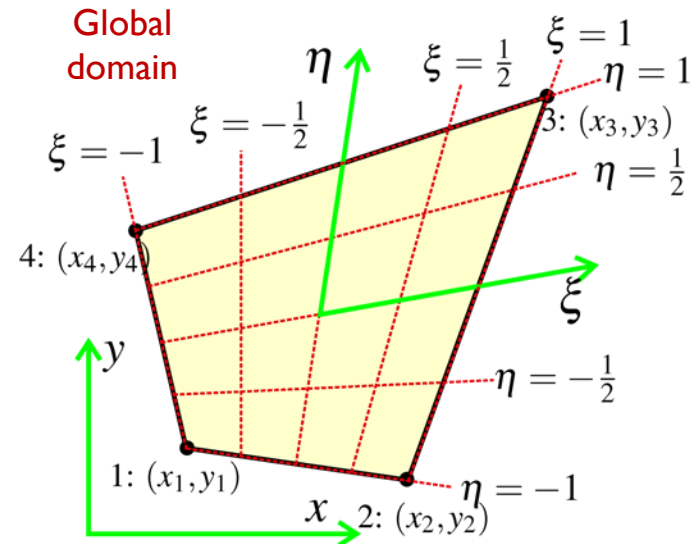
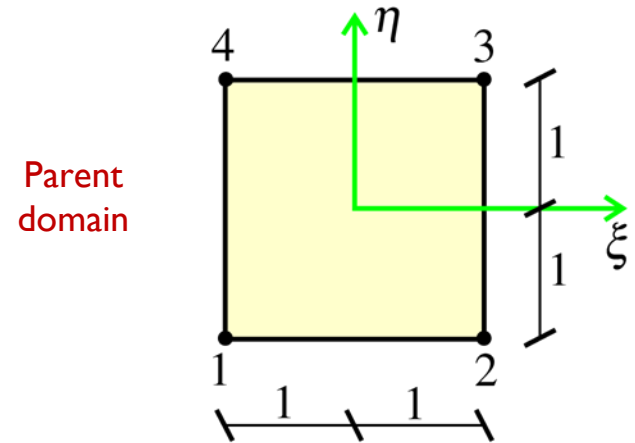
$$\mathbf{q}(x,y) = \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} = -\mathbf{D}\mathbf{B}(x,y)\mathbf{a}$$

with

$$\mathbf{B}(x,y) = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y} \end{bmatrix}$$

**Problem:** we have to find the derivatives of the shape functions with respect to the global  $(x,y)$  coordinates, but the shape functions are expressed in the isoparametric  $(\xi,\eta)$  coordinates. To overcome this problem we will define  $\mathbf{B}$  as a product of two matrices:

$$\mathbf{B}(x,y) = \mathbf{J}^{-1} \mathbf{D}_N$$





# Finite Element Method

## 2D heat conduction

$$\mathbf{B}(x,y) = \mathbf{J}^{-1} \mathbf{D}_N$$

The matrix  $\mathbf{D}_N$  contains the derivatives of the shape functions with respect to the isoparametric coordinates

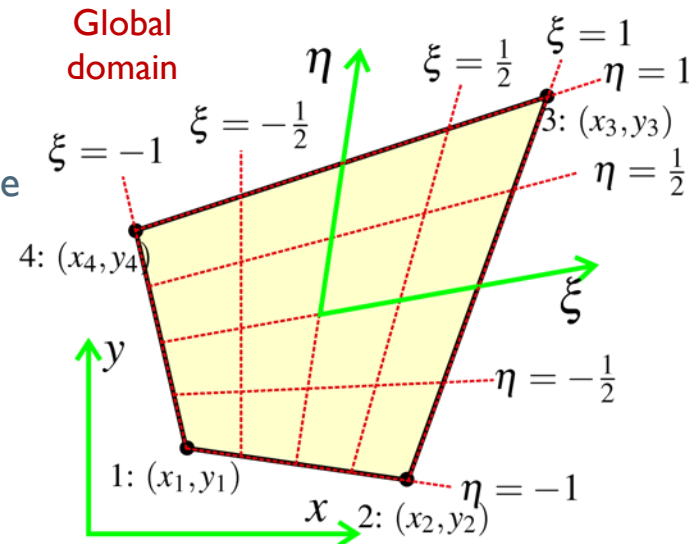
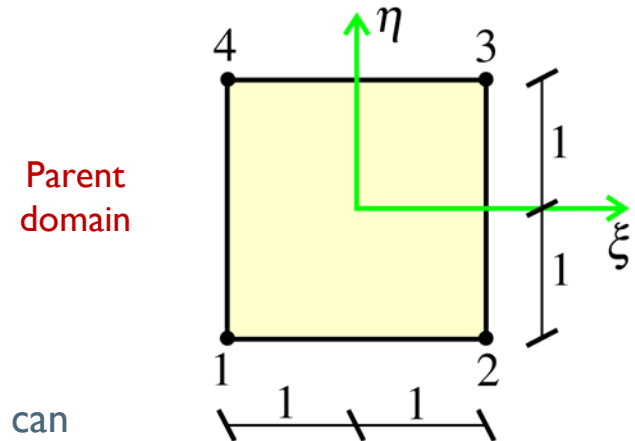
$$\mathbf{D}_N = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix}$$

With  $\mathbf{D}_N$  the temperature derivatives within the element can be found by

$$\left\{ \frac{\partial T}{\partial \xi} \right\} = \mathbf{D}_N \mathbf{a}$$

If we wanted the temperature derivatives the  $(\xi, \eta)$ -coordinates we were done now. But we need them in the  $(x,y)$  system in order to determine the heat flux. The transformation of derivatives from the  $(\xi, \eta)$  system into the  $(x,y)$  system is carried out by the inverse of the so-called "Jacobian matrix",  $\mathbf{J}^{-1}$

$$\mathbf{q}(x,y) = \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} = -\mathbf{D} \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{Bmatrix} = -\mathbf{D} \mathbf{B} \mathbf{a}$$



# Finite Element Method

## 2D heat conduction

### The Jacobian matrix

$$\mathbf{B}(x,y) = \mathbf{J}^{-1} \mathbf{D}_N$$

The matrix  $\mathbf{J}$  is called “the Jacobian matrix” and it relates the temperature derivatives in the two coordinate systems

$$\begin{Bmatrix} \frac{\partial T}{\partial \xi} \\ \frac{\partial T}{\partial \eta} \end{Bmatrix} = \mathbf{J} \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{Bmatrix}$$

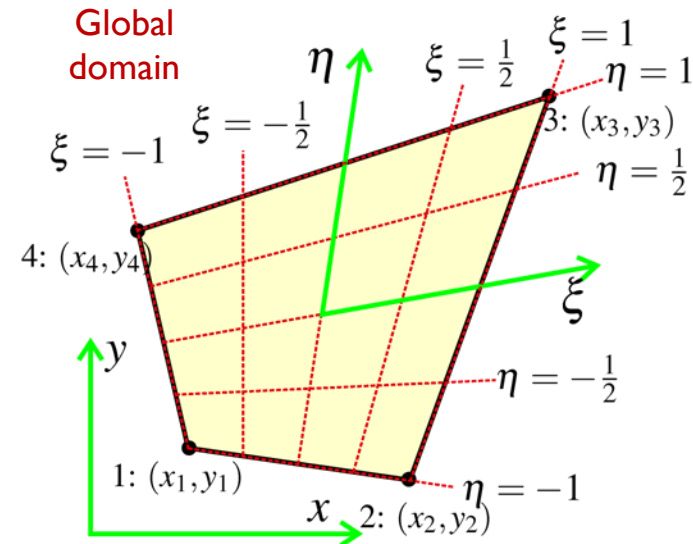
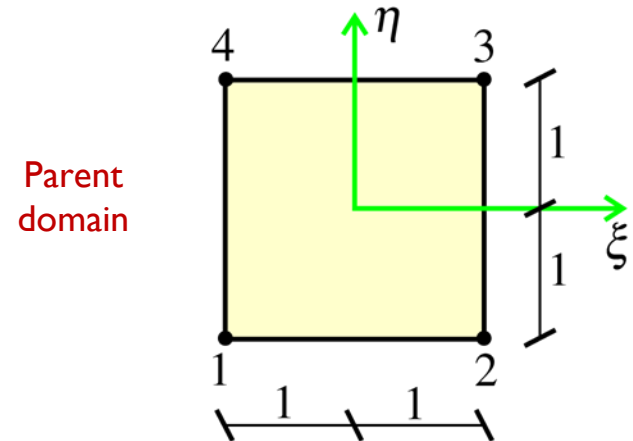
The elements of  $\mathbf{J}$  can be found by differentiating the temperature with respect to  $(\xi, \eta)$  by invoking the chain rule:

$$\frac{\partial T}{\partial \xi} = \frac{\partial T}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial T}{\partial y} \frac{\partial y}{\partial \xi}$$

$$\frac{\partial T}{\partial \eta} = \frac{\partial T}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial T}{\partial y} \frac{\partial y}{\partial \eta}$$

This gives us the elements of  $\mathbf{J}$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{N}}{\partial \xi} \mathbf{x} & \frac{\partial \mathbf{N}}{\partial \xi} \mathbf{y} \\ \frac{\partial \mathbf{N}}{\partial \eta} \mathbf{x} & \frac{\partial \mathbf{N}}{\partial \eta} \mathbf{y} \end{bmatrix} = \mathbf{D}_N [\mathbf{x} \ \mathbf{y}]$$



# Finite Element Method

## 2D heat conduction

### The Jacobian

The inverse of  $\mathbf{J}$ , which was needed in the expression for  $\mathbf{B}$ ,  $\mathbf{B}(x,y) = \mathbf{J}^{-1} \mathbf{D}_N$ , can now be found by

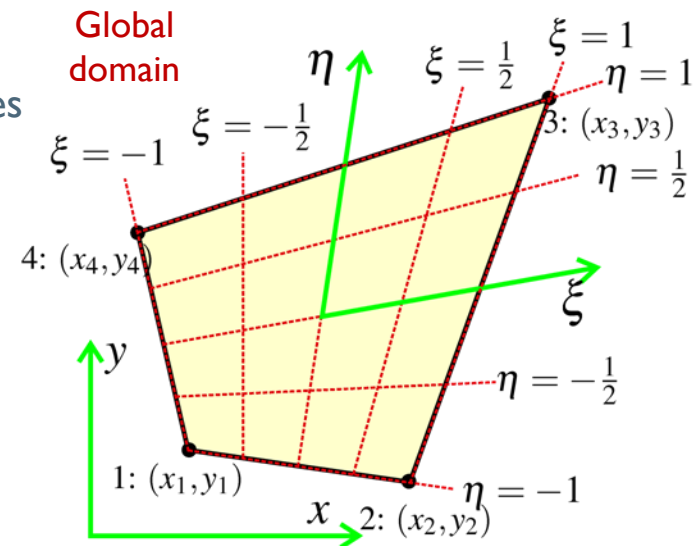
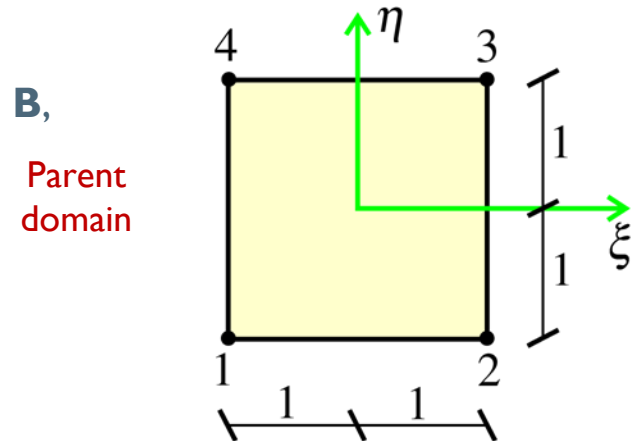
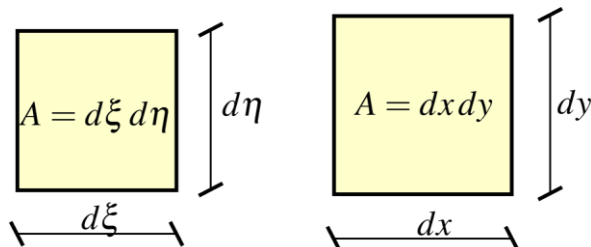
$$\mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix}$$

where the determinant of  $\mathbf{J}$ , which is often called “the Jacobian”, is given by

$$\det \mathbf{J} = J_{11}J_{22} - J_{21}J_{12}$$

The Jacobian can be regarded as a scale factor that relates infinitesimal areas in the parent domain and the global domain, see (Cook pp. 206 – 207), (OP pp. 376 – 380)

$$dx dy = d\xi d\eta \det \mathbf{J}$$



# Finite Element Method

## 2D heat conduction

### Stiffness matrix for the Iso4 element

The definition of the stiffness matrix was

$$\mathbf{K} = \int_A \mathbf{B}(x,y)^T t \mathbf{D} \mathbf{B}(x,y) dA = \int \int_A \mathbf{B}(x,y)^T t \mathbf{D} \mathbf{B}(x,y) dx dy$$

With the Jacobian matrix and its determinant we now have the tools to calculate the stiffness matrix of a isoparametric quadrilateral element.

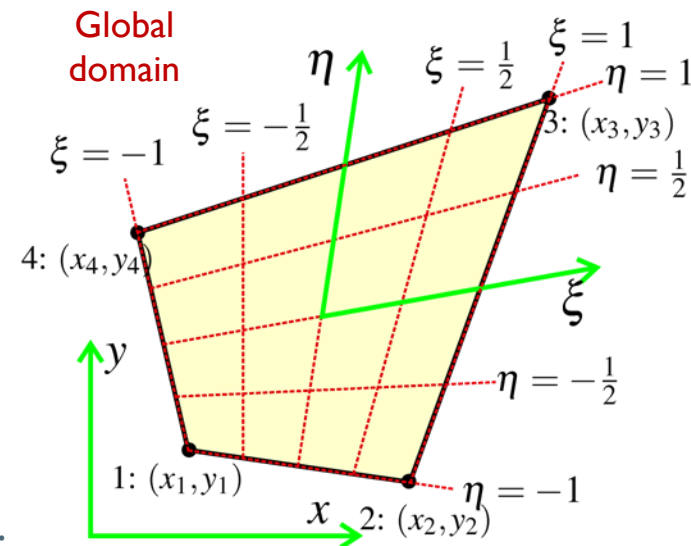
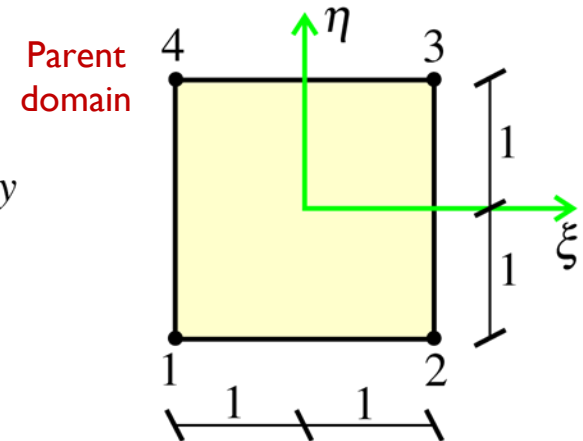
$$\mathbf{B}(x,y) = \mathbf{J}^{-1} \mathbf{D}_N$$

$$dx dy = d\xi d\eta \det \mathbf{J}$$

$$\mathbf{K} = \int_{-1}^1 \int_{-1}^1 (\mathbf{J}^{-1} \mathbf{D}_N)^T t \mathbf{D} \mathbf{J}^{-1} \mathbf{D}_N \det(\mathbf{J}) d\xi d\eta$$

The integral is non-linear and can, in general, not be solved analytically. Therefore we must use numerical integration.

As a remark it should be mentioned that the shape functions are still linear along the element sides. This means that boundary loads are distributed to the nodes in the same way as it was done with the Melosh element.



# Finite Element Method

## 2D heat conduction

### Numerical integration

Our concern is to find the following integral numerically

$$I = \int_{x_a}^{x_b} f(x) dx$$

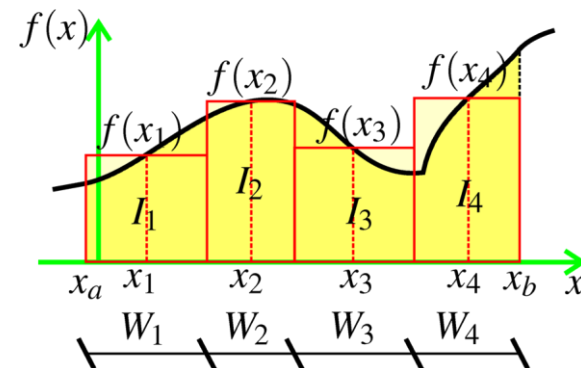
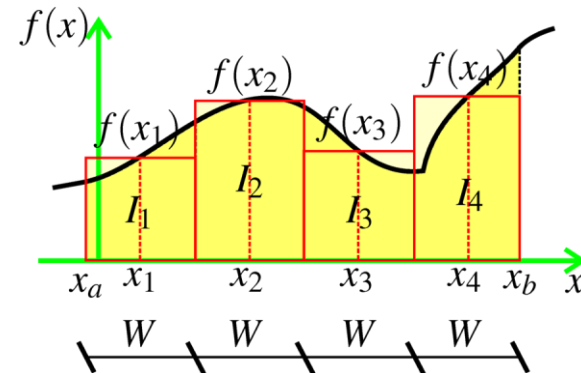
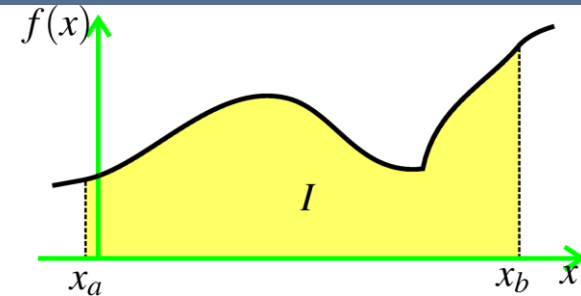
One way of doing this is to divide the domain in  $n$  equal intervals of equal width  $W$ , and sum up the contribution from  $n$  rectangular areas with the height given by the function value in the centre of the intervals

$$I \approx \sum_1^n I_i = \sum_1^n f(x_i) W$$

A second method is analogous to the above but with varying interval widths  $W_i$

$$I \approx \sum_1^n I_i = \sum_1^n f(x_i) W_i$$

Both methods will converge towards the exact integral when the number of intervals,  $n$ , increases.



# Finite Element Method

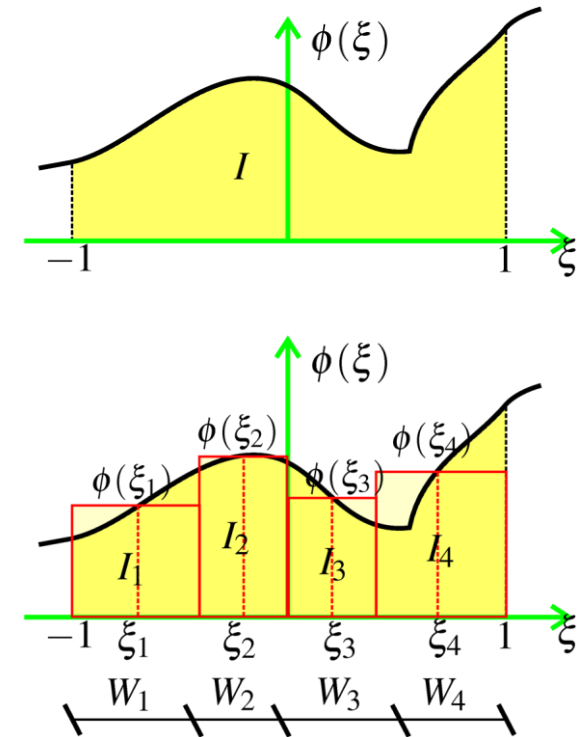
## 2D heat conduction

### Gauss quadrature

The position of the evaluation points  $x_i$  and the size of the widths (also called weight factors) can be optimized with respect to the function that is integrated. For integrals of polynomials in the interval  $[-1 ; 1]$  this is known as the so-called Gauss quadrature. Notice that this is the interval we are interested in with isoparametric elements as the isoparametric coordinates  $-1 \leq \xi, \eta \leq 1$ . With Gauss quadrature we can evaluate integrals as

$$I = \int_{-1}^1 \phi(\xi) d\xi \approx \sum_1^n I_i = \sum_1^n \phi(\xi_i) W_i$$

with the following positions and weights



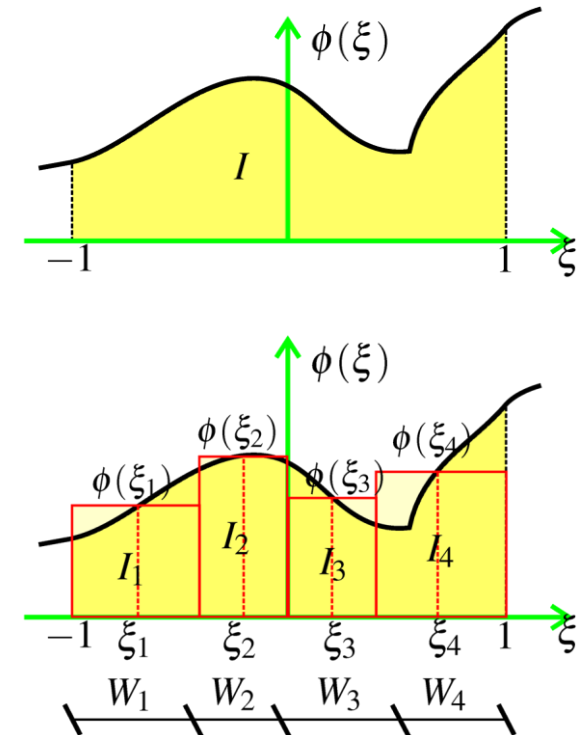
# Finite Element Method

## 2D heat conduction

### Gauss points and weights

Order $n$	Location of point $\xi_i$	Weight factor $W_i$
1	0	2
2	$\pm 1/\sqrt{3}$	1
3	$\pm\sqrt{0.6}$	$\frac{5}{9}$
	0	$\frac{8}{9}$
4	$\pm\sqrt{\frac{3+2\sqrt{1.2}}{7}}$	$\frac{1}{2} - \frac{1}{6\sqrt{1.2}}$
	$\pm\sqrt{\frac{3-2\sqrt{1.2}}{7}}$	$\frac{1}{2} + \frac{1}{6\sqrt{1.2}}$

A polynomial of degree  $2n - 1$  is integrated exact with a Gauss quadrature of degree  $n$ .



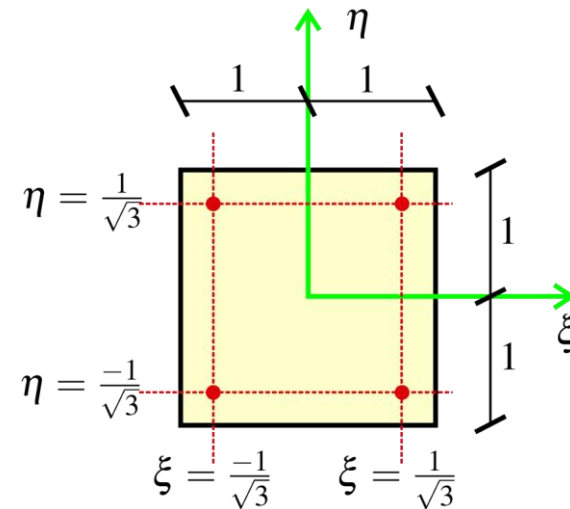
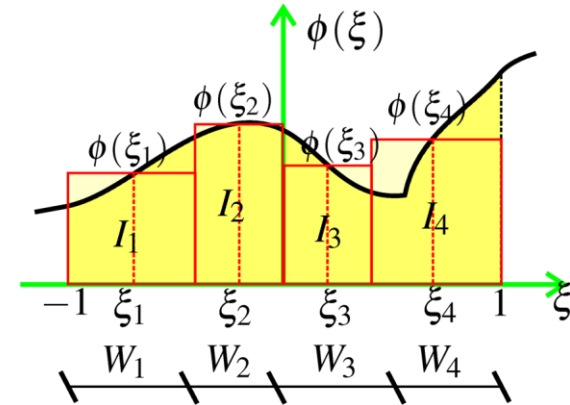
### Gauss quadrature in two dimensions

For the calculation of the stiffness matrix we need the Gauss quadrature in two dimensions. This is obtained as follows

$$\begin{aligned}
 I &= \int_{-1}^1 \int_{-1}^1 \phi(\xi, \eta) d\xi d\eta \approx \int_{-1}^1 \left( \sum_{i=1}^n \phi(\xi_i, \eta) W_i \right) d\eta \\
 &\approx \sum_{j=1}^n W_j \left( \sum_{i=1}^n \phi(\xi_i, \eta_j) W_i \right) = \sum_{i=1}^n \sum_{j=1}^n W_i W_j \phi(\xi_i, \eta_j)
 \end{aligned}$$

Example: Numerical integration of  $\phi(\xi, \eta)$  with Gauss order  $n = 2$ , which gives  $2 \cdot 2 = 4$  points:

$$\begin{aligned}
 I &\approx \phi(\xi_1, \eta_1) W_1 W_1 + \phi(\xi_2, \eta_1) W_2 W_1 \\
 &\quad + \phi(\xi_1, \eta_2) W_1 W_2 + \phi(\xi_2, \eta_2) W_2 W_2 \\
 &= \phi\left(\frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}\right) \cdot 1 \cdot 1 + \phi\left(\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}\right) \cdot 1 \cdot 1 \\
 &\quad + \phi\left(\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) \cdot 1 \cdot 1 + \phi\left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) \cdot 1 \cdot 1
 \end{aligned}$$





### Exercise: Numerical integration of two functions in 2D

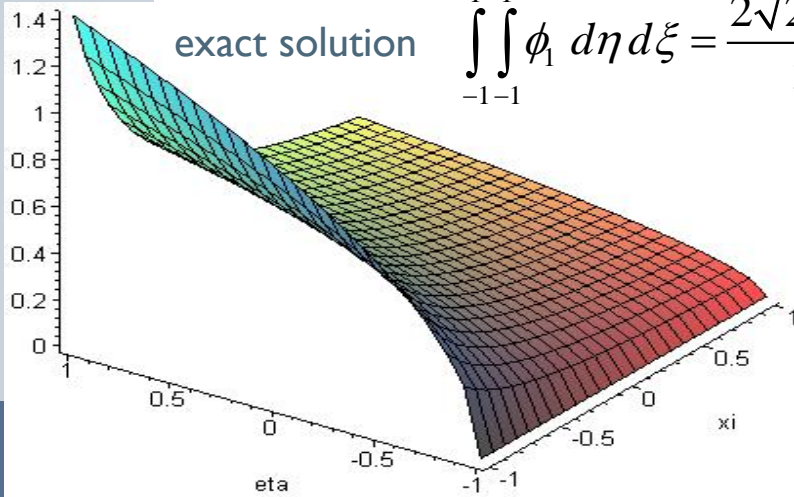
#### Steps

1. Open NumIntExc\_Lec34\_student.m
2. In the loop calculate the function value  $\phi$  at each Gauss point
3. Sum the integral in the variable "numint". Remember the weight functions (see the previous slide)
4. Save the exact solution in the variable "int\_exact".
5. Start with one Gausspoint, then two, and so on..
6. Run the program by pressing F5.

What is different about the two functions?

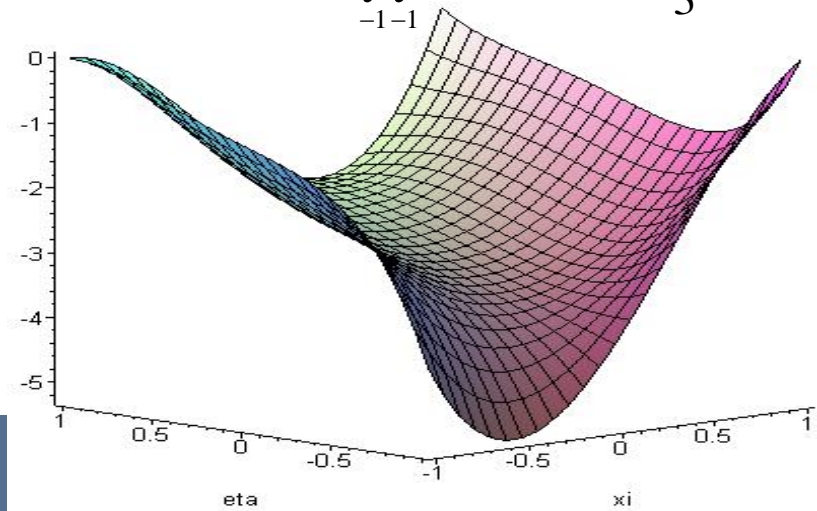
function 1:  $\phi_1 = \frac{\sqrt{\eta+1}}{2\xi+3}$

exact solution  $\int_{-1}^1 \int_{-1}^1 \phi_1 d\eta d\xi = \frac{2\sqrt{2}\ln 5}{3}$



function 2:  $\phi_2 = (\eta^3\xi + 1)(2\xi^2 - 3 + \eta)$

exact solution  $\int_{-1}^1 \int_{-1}^1 \phi_2 d\eta d\xi = -\frac{28}{3}$



# Finite Element Method

## 2D heat conduction

### Exercise: Calculate and program the stiffness matrix of the heat conduction Iso4-element

#### Steps

1. Open the file K\_QuadIso\_Heat.m
2. Setup a loop over the Gauss points
3. Initiate the stiffness matrix as zeros
4. Perform steps 5 to 9 inside a loop over the Gauss points
5. Calculate the and setup the  $\mathbf{D}_N$ -matrix
6. Calculate the Jacobian matrix
7. Calculate the determinant of the Jacobian matrix
8. Calculate the inverse Jacobian matrix
9. Calculate the integral at the current Gauss point and add it to the current value of the stiffness matrix,  $\mathbf{K}_e$
10. A test value with the parameters  $(x_1, x_2, x_3, x_4) = (1, 3, 3.5, -1)$ ,  $(y_1, y_2, y_3, y_4) = (0, -1, 1, 1)$ ,  $t = 1.3$ ,  $k_{xx} = 4.7$  and  $k_{yy} = 5.1$   $n = 2$  (Gauss order)

$$\mathbf{K} = \begin{bmatrix} 16.2632 & -1.5016 & -7.2511 & -7.5105 \\ -1.5016 & 4.8946 & -3.0044 & -0.3885 \\ -7.2511 & -3.0044 & 7.2511 & 3.0044 \\ -7.5105 & -0.3885 & 3.0044 & 4.8946 \end{bmatrix}$$

The diagram shows a rectangular domain of size 2x2, discretized into five elements (1, 2, 3, 4, 5). The nodes are numbered 1 through 9. The elements are numbered 1 through 5. The boundary conditions are:  $q_4 = 1$  (point load at node 4),  $q_7 = 2$  (point load at node 7), and  $\bar{q} = -1000 \frac{\text{J}}{\text{m}^2 \text{s}}$  (distributed load on the right boundary). The x and y axes are shown with green arrows and labels.

## Steps

1. Program the geometry (done earlier today)
2. Determine nodal heat supplies  $q_3$ ,  $q_6$ , and  $q_9$
3. Determine nodal heat supply  $q_4$  and  $q_7$  (We want  $T = 0$  on the left boundary,  $x = 0$ )
4. Make the appropriate programming in Topology.m and BoundaryConditions.m
5. Plot by executing the line “visualizeD2

1.3 Compare with the analytical solution:  $T(x) = -\frac{\bar{q}}{k_{xx}}x + T_{x=0}$

Hint: on the colorplot the temperatures can be found by activating the figure and typing “colorbar” at the command line

# Finite Element Method

## 2D heat conduction

### Exercise: Get familiar with the program and test the different elements

We will now try to carry out a calculation of the temperature in the tapered beam from the last lecture. This time we will use 2D elements.

- Set the variable `atype = 2` in `main.m`
- Try to run the program
- The number of elements can be controlled by changing the variable `nelem_y` in `Coordinates.m`
- Does the three elements of todays lecture give the same result?
- Does it make any difference what order of Gauss quadrature you use in the Iso4-element? (this is controlled in the file `K_QuadIso_Heat.m`)
- Try to take a look at how the coordinates and the topology is calculated. This might give you inspiration for your own program in the semester project.

