Advances in the use of simplicial meshes for flow problems

R. Rossi, P. Dadvand, A. Larese de Tetto
N. Maireni, M. Davari - CIMNE

R. Wüchner - TUM

Barcelona, WCCM XI

International Center for Numerical Methods in Engineering CIMNE

Today's topics:

- **Mixed u-ε method**: towards applications in fluid mechanics
- Exploring **two different approaches to local enrichment** targeted to embedding objects into a "fluid domain"– laplacian model problem

Mixed u-ε method: towards applications in fluid mechanics

A Mixed u-ε method

Standard irreducible form of equilibrium:

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b} = \rho \ddot{\boldsymbol{u}} \text{ in } \Omega$$

Mixed u-ε form (see works of Cervera et al.):

$$\nabla \cdot \left(C(\varepsilon, \boldsymbol{\sigma}) : \varepsilon \right) + \boldsymbol{b} = \rho \ddot{\boldsymbol{u}} \text{ in } \Omega$$
$$-\varepsilon + \varepsilon(u) = 0 \quad \text{in } \Omega$$

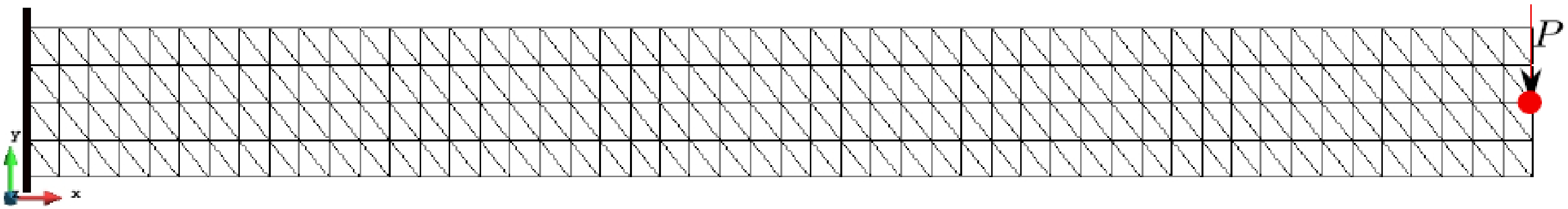
Explicit form

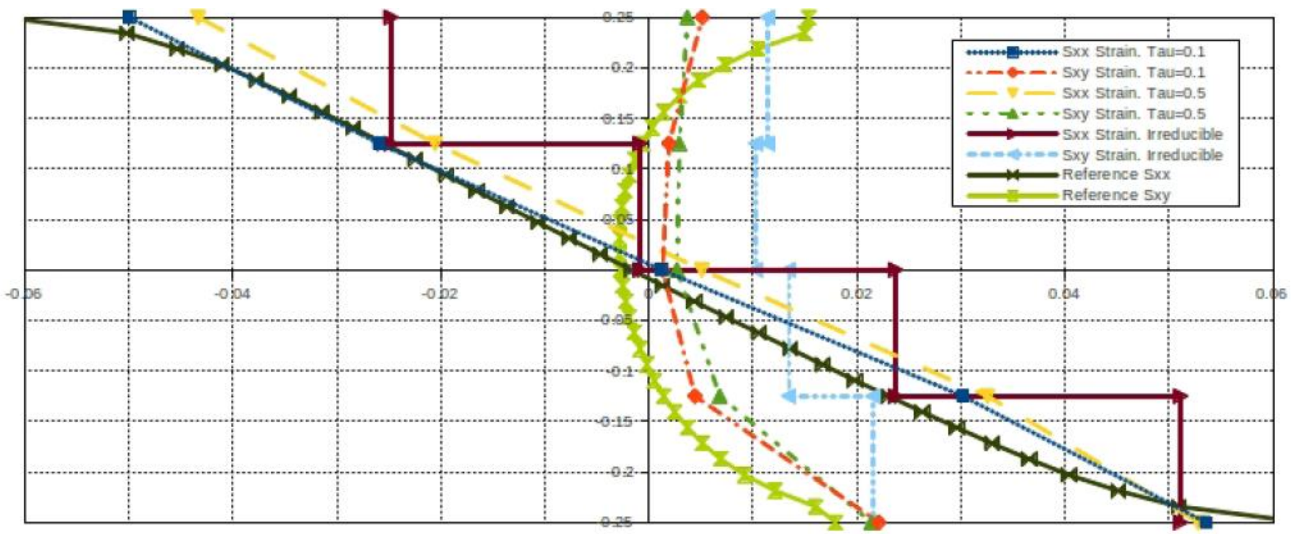
- compute the acceleration $\ddot{\boldsymbol{u}}_n = \frac{\dot{\boldsymbol{u}}^{n+\frac{1}{2}} - \dot{\boldsymbol{u}}^{n-\frac{1}{2}}}{\Delta t}$.
- evaluate on every element the discontinuous strain $\boldsymbol{\varepsilon}(\boldsymbol{u})$.
- evaluate the strain $\boldsymbol{\varepsilon}_h = \mathcal{P}(\boldsymbol{\varepsilon}(\boldsymbol{u}))$,that is, $\boldsymbol{\varepsilon}_{\boldsymbol{h}} = \breve{\boldsymbol{M}}_{\tau}^{-1} \breve{\boldsymbol{G}} \boldsymbol{u}$.
- compute internal forces taking into account the stabilized strain.
- compute the mid-step velocity by solving
  $\dot{\boldsymbol{u}}_{n+\frac{1}{2}} = [2\boldsymbol{M} + \Delta t \boldsymbol{D}]^{-1}[(2\boldsymbol{M} - \Delta t \boldsymbol{D})\dot{\boldsymbol{u}}_{n-\frac{1}{2}} + 2\Delta t(\boldsymbol{f}_n^{ext} - \boldsymbol{f}_n^{int}(\boldsymbol{u}_n, \boldsymbol{\varepsilon}_n^{stab}))]$
- compute end-of-step displacements as $\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \Delta t \dot{\boldsymbol{u}}_{n+\frac{1}{2}}$

Advantages of u-ε

- More accurate than irreducible for a given mesh (at the price of having more unknowns)
- Results are more “mesh independent”
- Ensures convergence of nodal strains
- **LARGER STABLE TIME STEP FOR EXPLICIT PROBLEMS** on a given mesh.
- **Suitable for large-scale parallelization**

One example of application

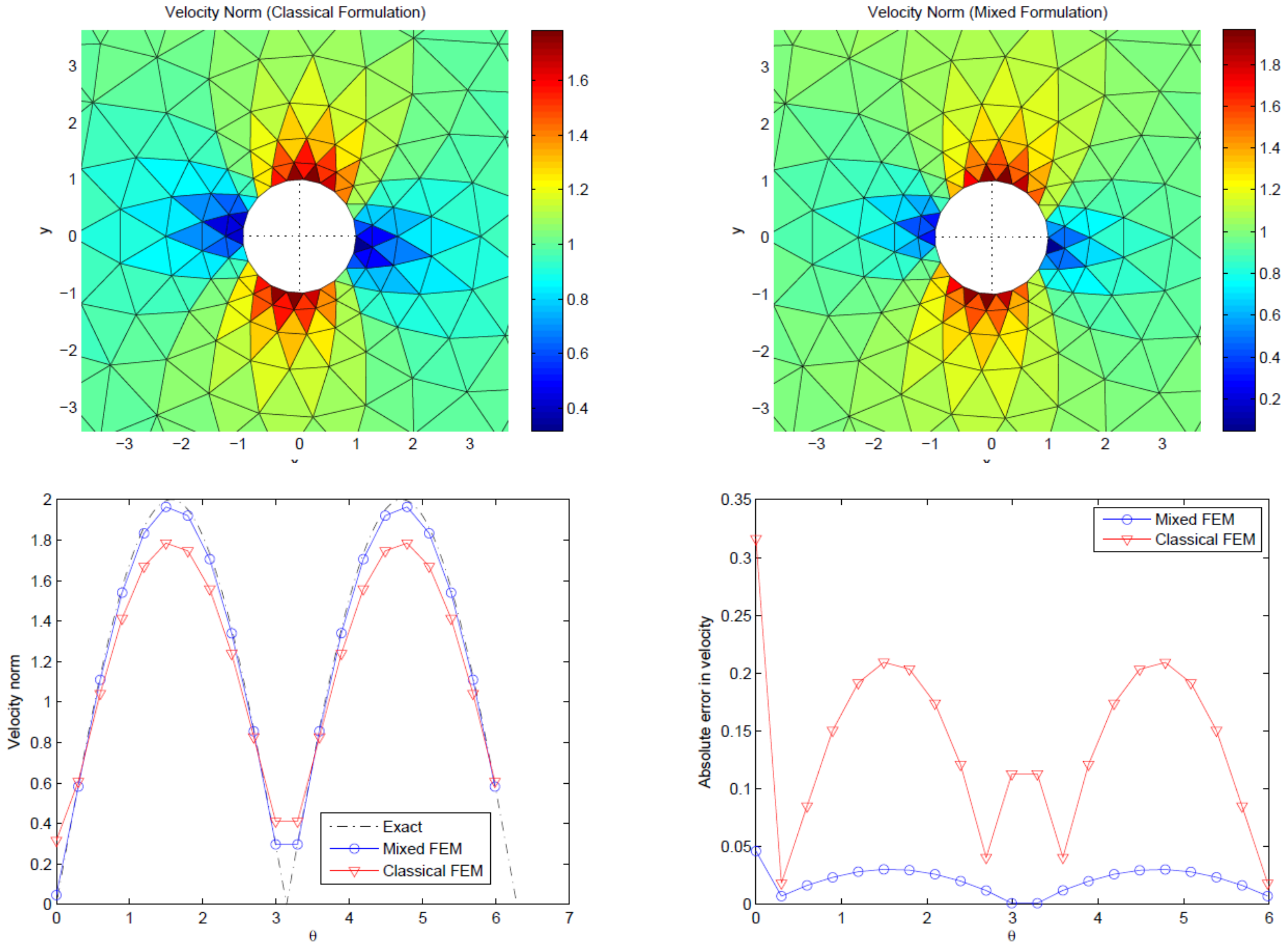




Stress distribution at left side:

Irreducible vs mixed vs reference

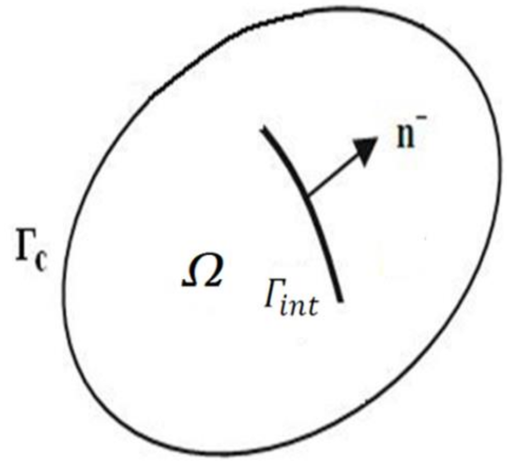
Application to diffusion problem



approaches to local enrichment
(taking diffusion as model problem)

•Transient heat transfer by conduction

$$\rho c_p \frac{\partial \boldsymbol{T}}{\partial t} - \nabla.(k\nabla \boldsymbol{T}) = 0 \ \ in \ \ \Omega \times (0.t_f)$$



•Boundary conditions

$$\boldsymbol{T} = \boldsymbol{T_c} \quad on\ \Gamma_c$$

$$\boldsymbol{T} = \boldsymbol{T_{int}} \quad on\ \Gamma_{int}$$

$$k\nabla \boldsymbol{T}.n = \boldsymbol{q_{int}} \quad on\ \Gamma_{int}$$

•Bilinear form acting on a test function w

$$a\left(\frac{\partial \boldsymbol{T}}{\partial t}, w\right) + b(\boldsymbol{T}, w) = l(w)$$

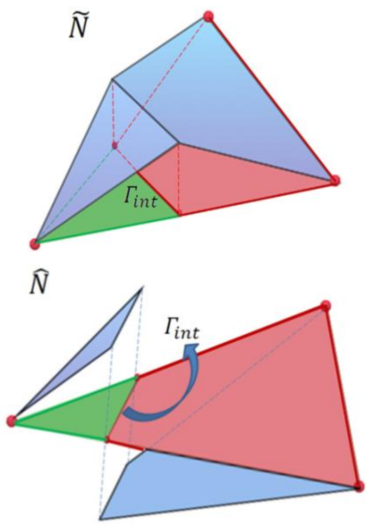
$$\left(\frac{\rho C_P \partial \boldsymbol{T}}{\partial t}, w\right) \qquad (k\nabla \boldsymbol{T}, \nabla w) \qquad -(\boldsymbol{q_w}, w)$$

- In finite element the temperature is approached discretized form

$$\boldsymbol{T}^h(x,t) = \sum_{i=1}^{3} N_i(x)\boldsymbol{T}_i(t)$$

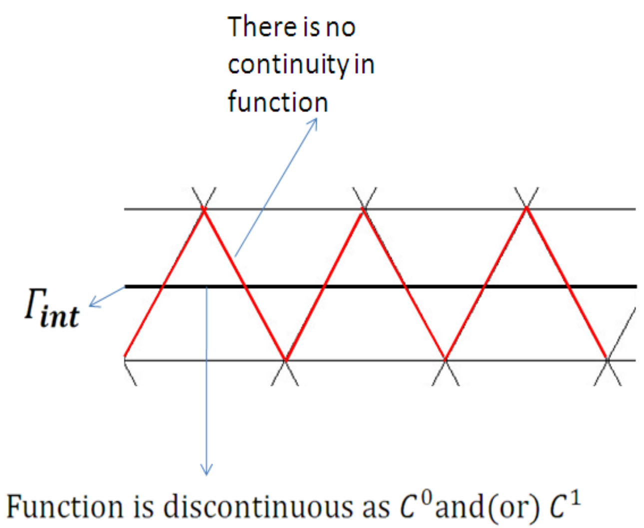
•METHOD I (ENRICHMENT)

$$\boldsymbol{T}^h(x) = \sum_{i \in I} N_i(x)\boldsymbol{T}_i + \underbrace{\underbrace{\tilde{N}(x)\tilde{\boldsymbol{T}}}_{kink} + \underbrace{\hat{N}(x)\hat{\boldsymbol{T}}}_{jump}}$$



•***Local*** *special functions to capture discontinuities in the solution*

- C^0 *continuity is violated across each of the edges intersected by the interface , in the work we show heuristically how the method appears to work satisfactorily in real cases despite this defect*



Inte

•*To define weak equation we decompose the discrete problem by using test functions from linear, kink and jump discontinuous respectively*

$$\boldsymbol{W} = w^i + \widetilde{w} + \widehat{w} \quad i = 1{,}3$$
$$\boldsymbol{T} = T^j + \widetilde{T} + \widehat{T} \quad j = 1{,}3$$

$$a_{ss}^{ij}\left(\frac{\partial \boldsymbol{T}^j}{\partial t}, w^i\right) + a_{sk}^{i}\left(\frac{\partial \widetilde{\boldsymbol{T}}}{\partial t}, w^i\right) + a_{sm}^{i}\left(\frac{\partial \widehat{\boldsymbol{T}}}{\partial t}, w^i\right) + b_{ss}^{ij}(\boldsymbol{T}^j, w^i) + b_{sk}^{i}(\widetilde{\boldsymbol{T}}, w^i) + b_{sm}^{i}(\widehat{\boldsymbol{T}}, w^i) = l^i(w^i) \quad i,j = 1{,}3$$

$$a_{ks}^{j}\left(\frac{\partial \boldsymbol{T}^j}{\partial t}, \widetilde{w}\right) + a_{kk}\left(\frac{\partial \widetilde{\boldsymbol{T}}}{\partial t}, \widetilde{w}\right) + a_{km}\left(\frac{\partial \widehat{\boldsymbol{T}}}{\partial t}, \widetilde{w}\right) + b_{ks}^{j}(\boldsymbol{T}^j, \widetilde{w}) + b_{kk}(\widetilde{\boldsymbol{T}}, \widetilde{w}) + b_{km}(\widehat{\boldsymbol{T}}, \widetilde{w}) = l(\widetilde{w}) \quad j = 1{,}3$$

$$a_{ms}^{j}\left(\frac{\partial \boldsymbol{T}^j}{\partial t}, \widehat{w}\right) + a_{mk}\left(\frac{\partial \widetilde{\boldsymbol{T}}}{\partial t}, \widehat{w}\right) + a_{mm}\left(\frac{\partial \widehat{\boldsymbol{T}}}{\partial t}, \widehat{w}\right) + b_{ms}^{j}(\boldsymbol{T}^j, \widehat{w}) + b_{mk}(\widetilde{\boldsymbol{T}}, \widehat{w}) + b_{\mathrm{mm}}(\widehat{\boldsymbol{T}}, \widehat{w}) = l(\widehat{w}) \quad j = 1{,}3$$

•*Where the sub-index s refers to standard temperature degrees of freedom and sub-indexes k and m refers to the additional degrees of freedom associate with kink and jump discontinuities respectively*

Once discretized we get

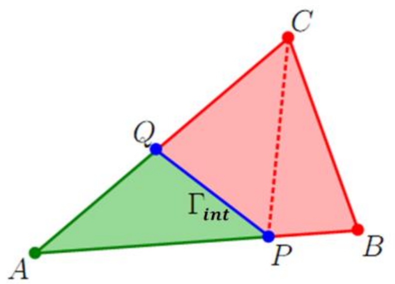
$$\begin{bmatrix} \begin{bmatrix} \mathrm{A}_{ss}^{ij} \end{bmatrix}^{(3\times3)} & \begin{bmatrix} \mathrm{A}_{sk}^{i} & \mathrm{A}_{sm}^{i} \end{bmatrix}^{(3\times2)} \\ \begin{bmatrix} \mathrm{A}_{ks}^{j} \\ \mathrm{A}_{ms}^{j} \end{bmatrix}^{(2\times3)} & \begin{bmatrix} \mathrm{A}_{kk} & \mathrm{A}_{km} \\ \mathrm{A}_{mk} & \mathrm{A}_{mm} \end{bmatrix}^{(2\times2)} \end{bmatrix} \times \begin{bmatrix} \mathbf{T}^{j^n} \\ \tilde{\mathbf{T}}^n \\ \hat{\mathbf{T}}^n \end{bmatrix}^{(5\times1)} = \begin{bmatrix} \mathbf{F}^j \\ \tilde{\mathbf{F}} \\ \hat{\mathbf{F}} \end{bmatrix}^{(5\times1)} \qquad i,j = 1,3$$

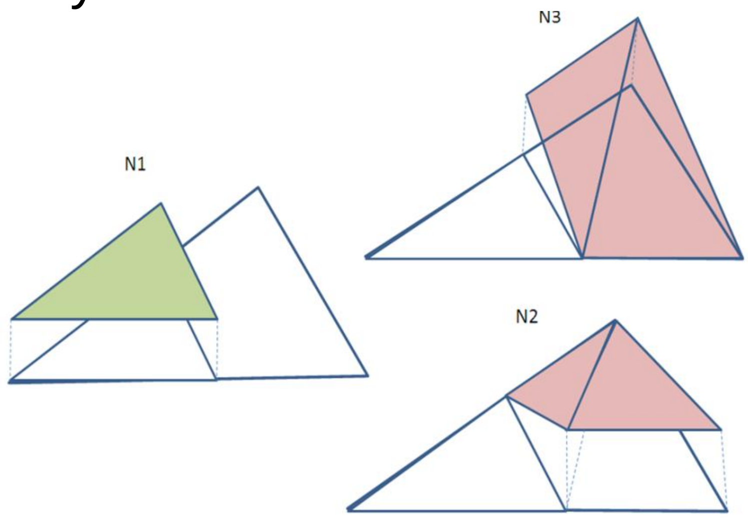
Using the fact that the enrichment functions are local to each element, we eliminate $\tilde{T}^n$ and $\hat{T}^n$ *at the elementary level before final assembly as follows*

$$\left([\mathrm{A_{ss}}] - [\mathrm{A_{sk}} \quad \mathrm{A_{sm}}] \begin{bmatrix} \mathrm{A_{kk}} & \mathrm{A_{km}} \\ \mathrm{A_{mk}} & \mathrm{A_{mm}} \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{A_{ks}} \\ \mathrm{A_{ms}} \end{bmatrix} \right) \mathbf{T}^{\mathrm{n}} = \mathbf{F} + [\mathrm{A_{sk}} \quad \mathrm{A_{sm}}] \begin{bmatrix} \mathrm{A_{kk}} & \mathrm{A_{km}} \\ \mathrm{A_{mk}} & \mathrm{A_{mm}} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{\mathbf{F}} \\ \hat{\mathbf{F}} \end{bmatrix}$$

METHOD II (Change FE space)

•The method with the ability to capture discontinuous within the element not by
•enrichment functions but by local modification in nodal shape function of the elements





Shape functions introduced by Buscaglia and others

IMPOSING BOUNDARY CONDITIONS ON DISCONTINUOUS

Neumann type boundary condition:

straightforward since it only needs to integrate the imposed flux over the cut surface. In particular for the application of an adiabatic boundary condition (zero heat flux) the terms including the flux need to be zero:

Dirichlet type boundary condition

- ***Method I → (***Local) Lagrange Multiplier
- ***Method II → (***Local) Penalty method

IMPOSING DIRICHLET BOUNDARY CONDITIONS (METHOD I)

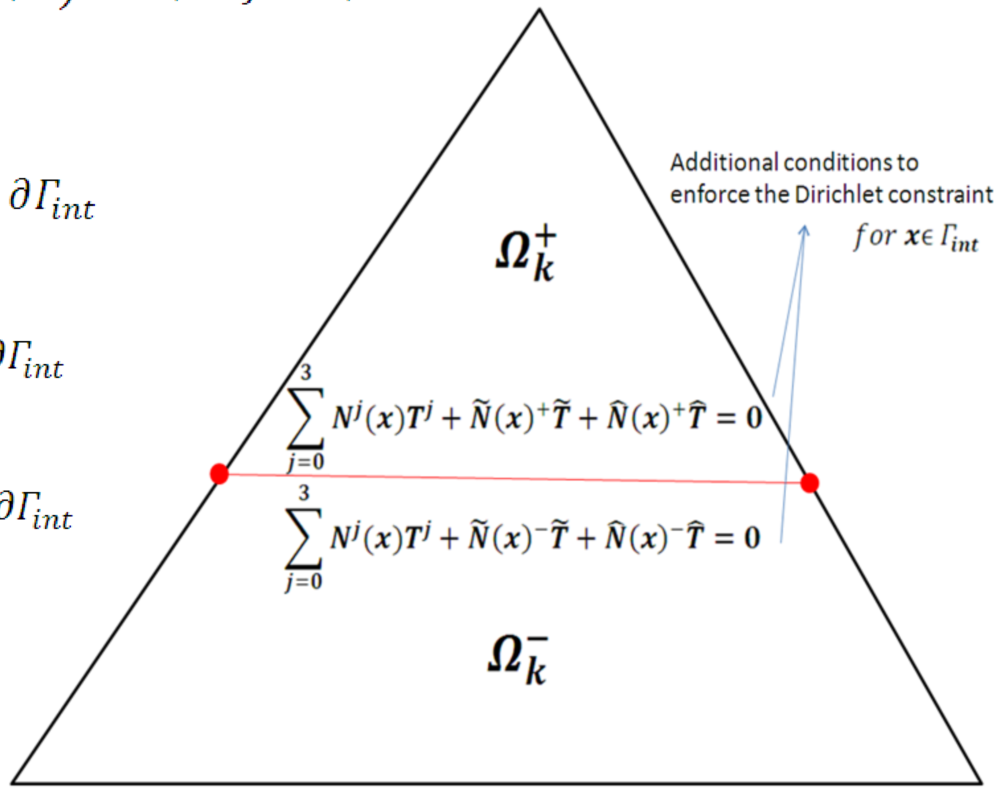
- To impose for instance a value say zero for the temperature on interface, Lagrange Multiplier method has been considered. we propose to add two following conditions to the weak form

$$c_s^j(\mathcal{V}, \boldsymbol{T}^j) + c_k{}^{+}(\tilde{\mathcal{V}}, \tilde{\boldsymbol{T}}) + c_m{}^{+}(\hat{\mathcal{V}}, \hat{\boldsymbol{T}}) = 0, \quad j = 1,3$$

$$c_s^j(\mathcal{V}, \boldsymbol{T}^j) + c_k{}^{-}(\tilde{\mathcal{V}}, \tilde{\boldsymbol{T}}) + c_m{}^{-}(\hat{\mathcal{V}}, \hat{\boldsymbol{T}}) = 0, \quad j = 1,3$$

Where:

$$\begin{cases} c_s^j(\mathcal{V}, \boldsymbol{T}^j) := \sum_{j=0}^{3} \int_{\Gamma_{int}} \boldsymbol{T}^j \mathcal{V} \, \partial \Gamma_{int} \\ c_k{}^{+(-)}(\tilde{\mathcal{V}}, \tilde{\boldsymbol{T}}) := \int_{\Gamma_{int}} \tilde{\boldsymbol{T}}^{+(-)} \tilde{\mathcal{V}} \, \partial \Gamma_{int} \\ c_m{}^{+(-)}(\hat{\mathcal{V}}, \hat{\boldsymbol{T}}) := \int_{\Gamma_{int}} \hat{\boldsymbol{T}}^{+(-)} \hat{\mathcal{V}} \, \partial \Gamma_{int} \end{cases}$$



•Hence the system formed by the enrichment variables and by the lagrange multipler can be writing as below

$$\begin{bmatrix} \begin{bmatrix} A_{ss}^{ij} \end{bmatrix}^{(3\times3)} & \begin{bmatrix} A_{sk}^{i} & A_{sm}^{i} & C_{s}^{i} & C_{s}^{i} \end{bmatrix}^{(3\times4)} \\ \begin{bmatrix} A_{ks}^{j} \\ A_{ms}^{j} \\ C_{s}^{j} \\ C_{s}^{j} \end{bmatrix}^{(4\times3)} & \begin{bmatrix} A_{kk} & A_{km} & C_{k}^{+} & C_{k}^{-} \\ A_{mk} & A_{mm} & C_{m}^{+} & C_{m}^{-} \\ C_{k}^{+} & C_{m}^{+} & 0 & 0 \\ C_{k}^{-} & C_{m}^{-} & 0 & 0 \end{bmatrix}^{(4\times3)} \end{bmatrix} \times \begin{bmatrix} \mathbf{T}^{j^{n}} \\ \tilde{\mathbf{T}}^{n} \\ \hat{\mathbf{T}}^{n} \\ \lambda^{+} \\ \lambda^{-} \end{bmatrix}^{(7\times1)} = \begin{bmatrix} \mathbf{F}^{j} \\ \tilde{\mathbf{F}} \\ \hat{\mathbf{F}} \\ 0 \\ 0 \end{bmatrix}^{(7\times1)} \qquad i,j = 1,3$$

•One of the features of this method is that the system formed by the enrichment variables and by the lagrange multipler can be statically condensed prior to assembly.

•Note that it is not possible in general to condense out the lagrange multipliers. It can be done in our case since matrix block that corresponds to the enrichment is invertible

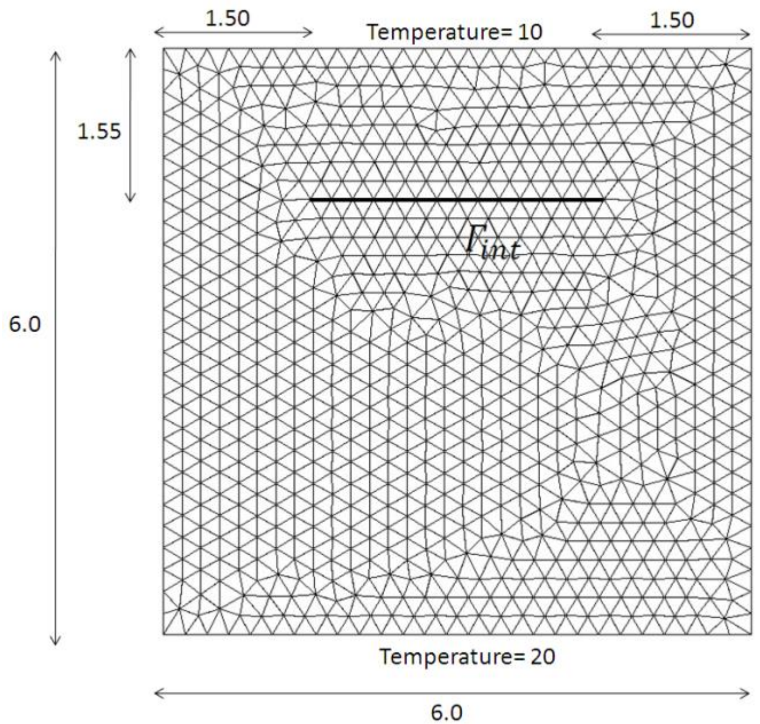
•IMPOSING DIRICHLET BOUNDARY CONDITIONS (METHOD II)

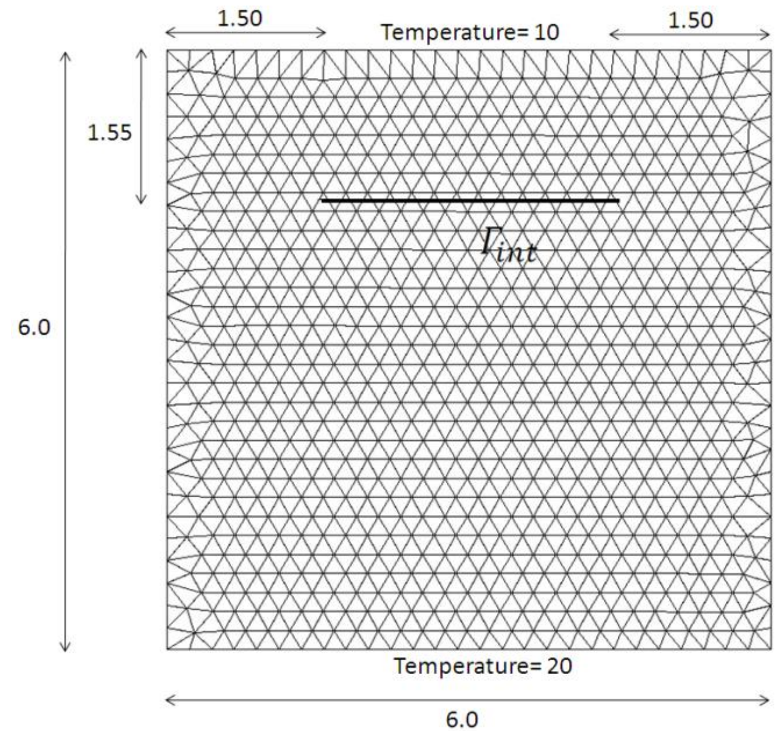
No local unknowns → Penalty method is employed (making the approach not attractive for real problems)

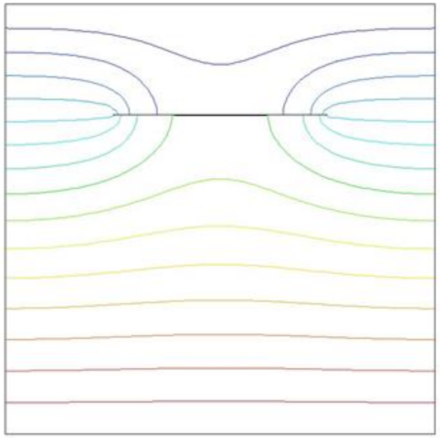
Here ***local*** Lagrange Multiplier can NOT be used since there are no local enrichments. Use of lagrange multipliers would hence imply modifying the global system

EXAMPLES

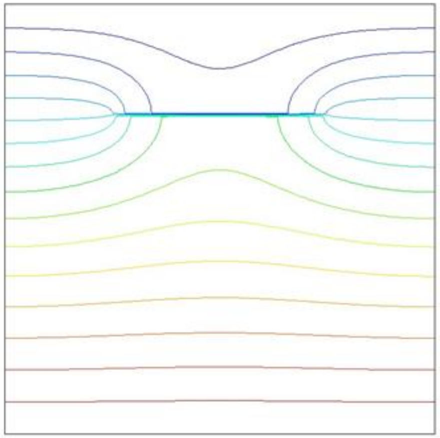
- Thermal conductivity K = 1.0 (for the entire domain)
- Both the temperature and its gradient are enforced to be zero separately
- Results obtain from our proposed methods are compared with results of classic finite element method where the internal interface is matched by the mesh



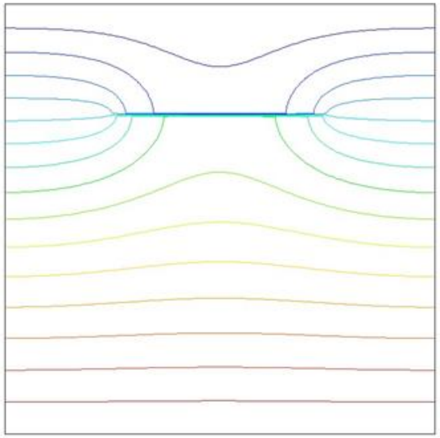




•Classic Finite Element Method



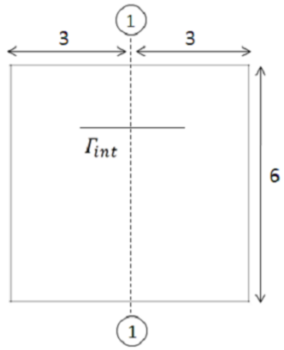
•**Method I (enrichment)**



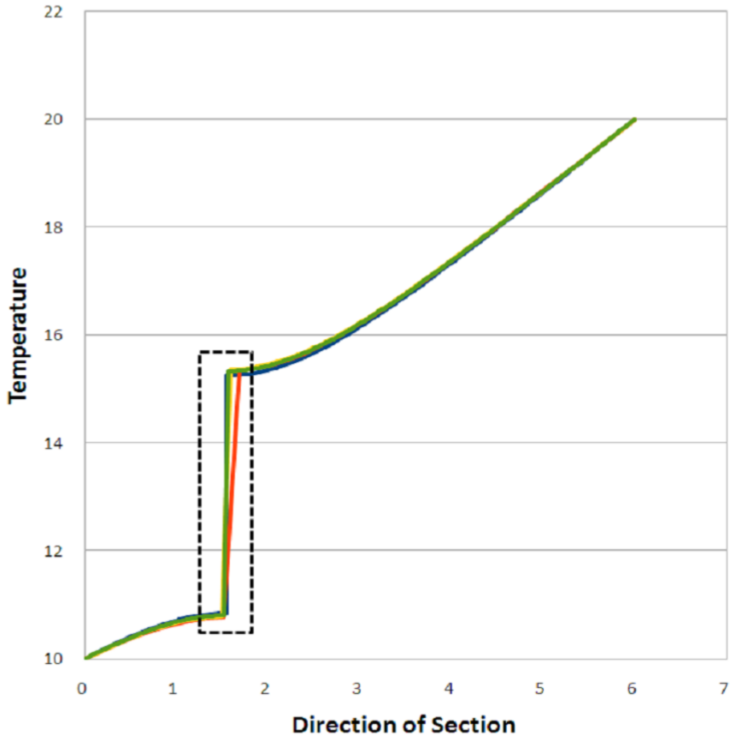
•**Method II**

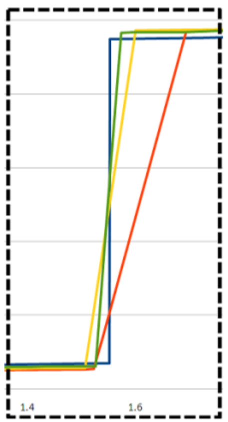
•showing contour lines of the temperature where the **Neumann** boundary condition is imposed

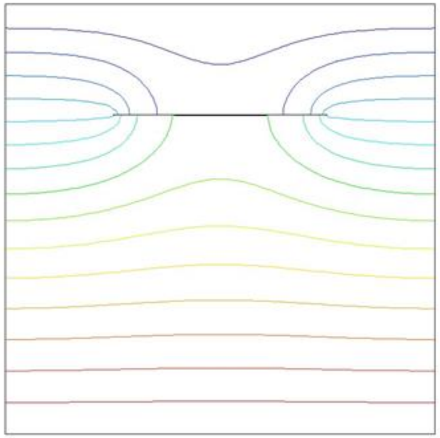
$$(\nabla T . n = 0)$$



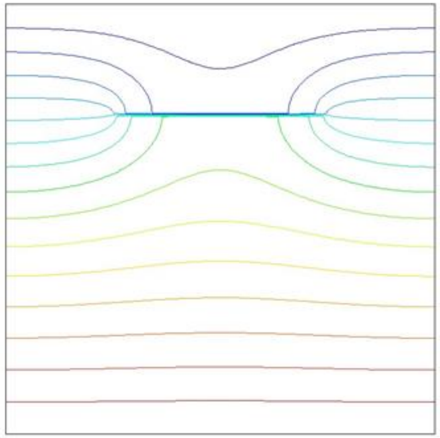
METHOD I



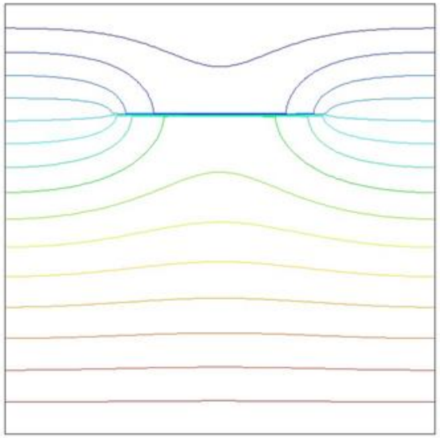




•Classic Finite Element Method



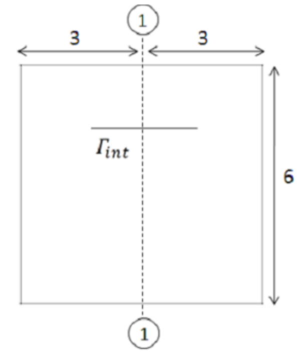
•Method I



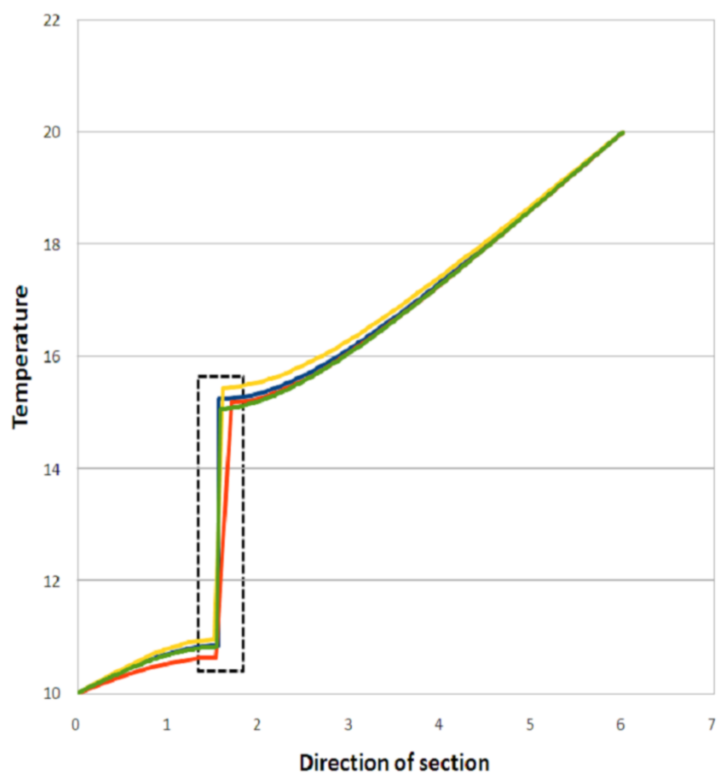
•Method II

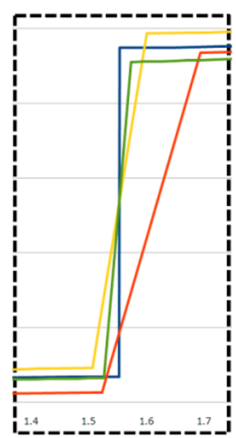
•showing contour lines of the temperature where the **Neumann** boundary condition is imposed

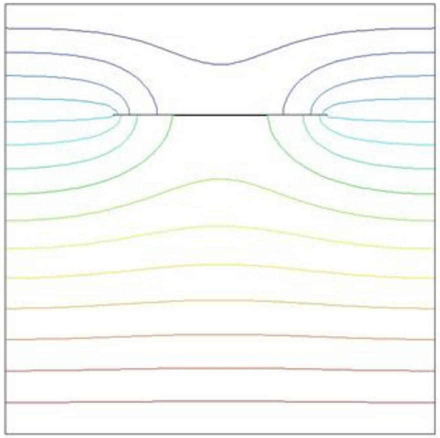
$$(\nabla T . n = 0)$$



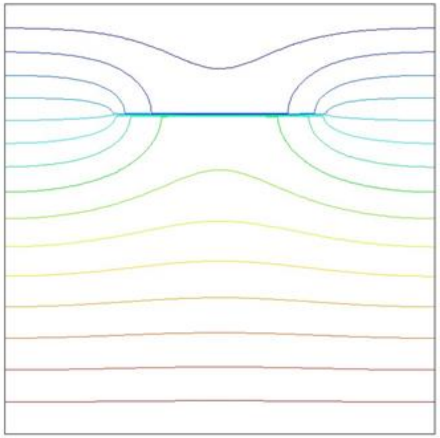
METHOD II



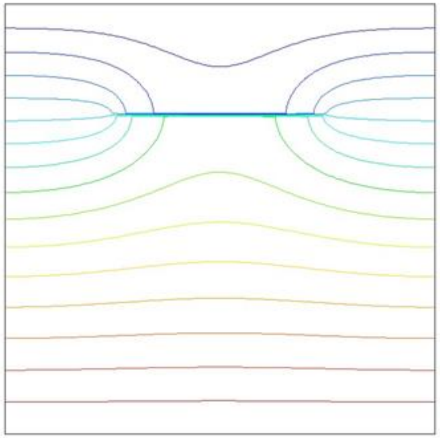




•Classic Finite Element Method



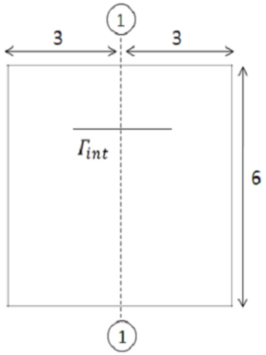
•Method I



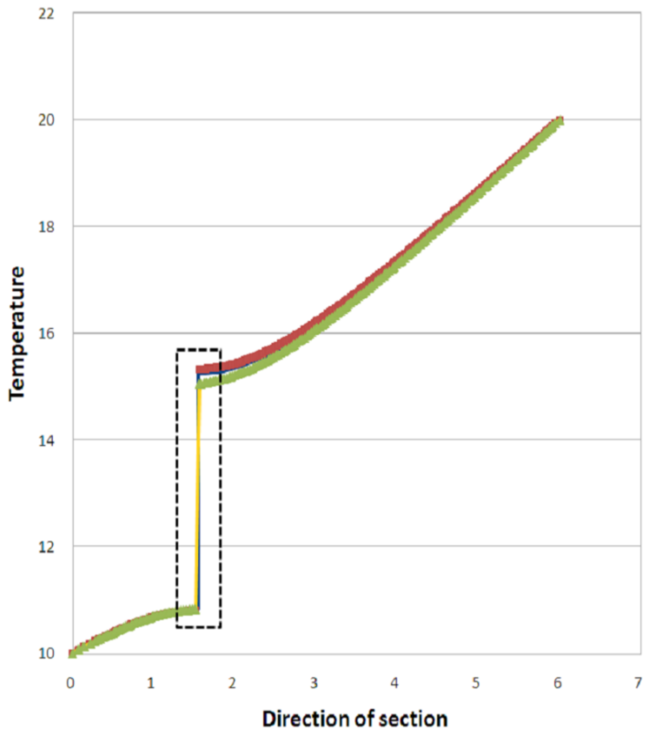
•Method II

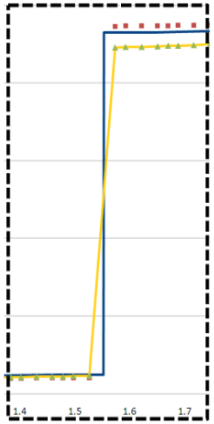
•showing contour lines of the temperature where the Neumann boundary condition is imposed

$$(\nabla T . n = 0)$$



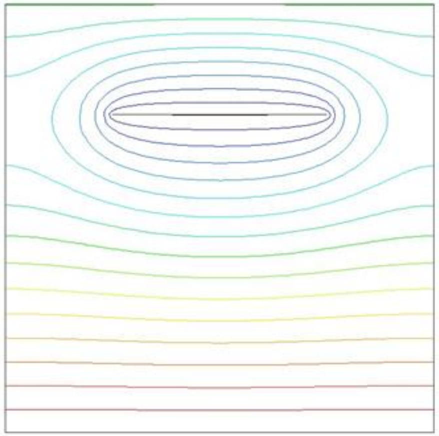
COMPARISON



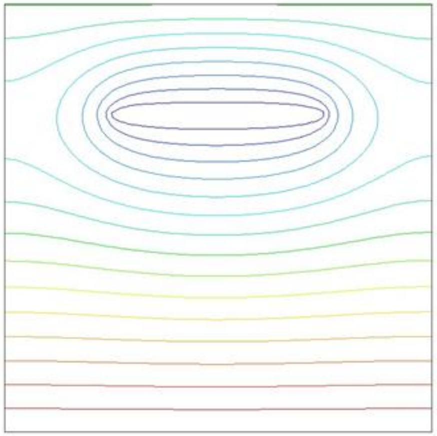


•showing contour lines of the temperature where the value of the temperature is imposed to zero

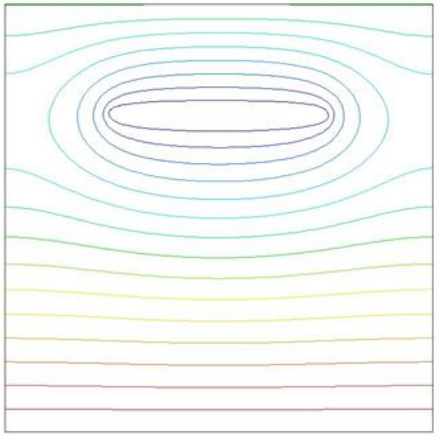
•*(T = 0)*



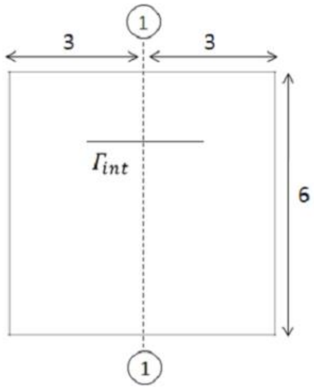
•Classic Finite Element Method



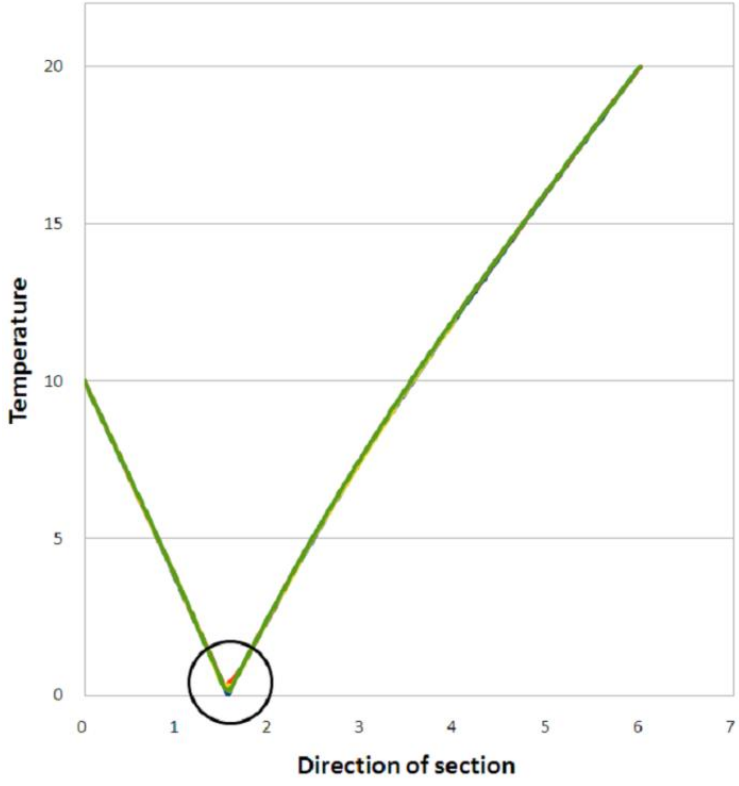
•Method I

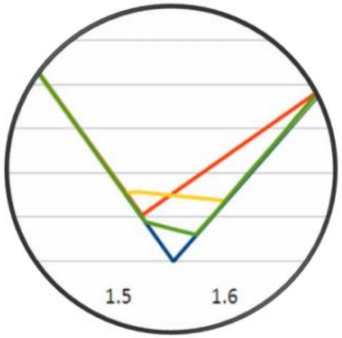


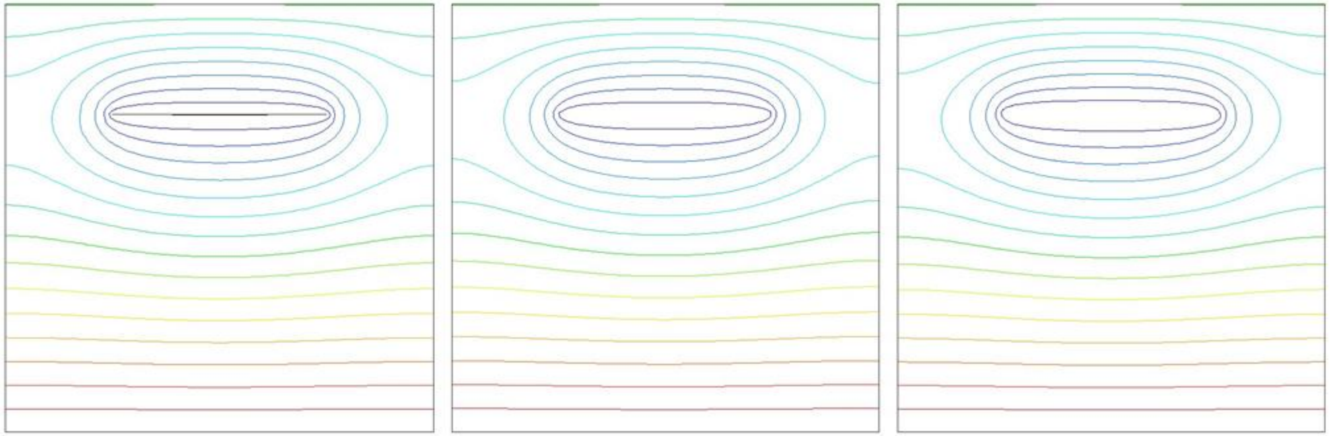
•Method II



METHOD I







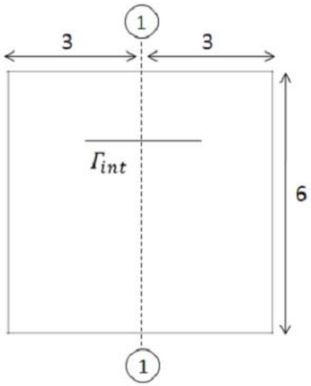
•showing contour lines of the temperature where the value of the temperature is imposed to zero

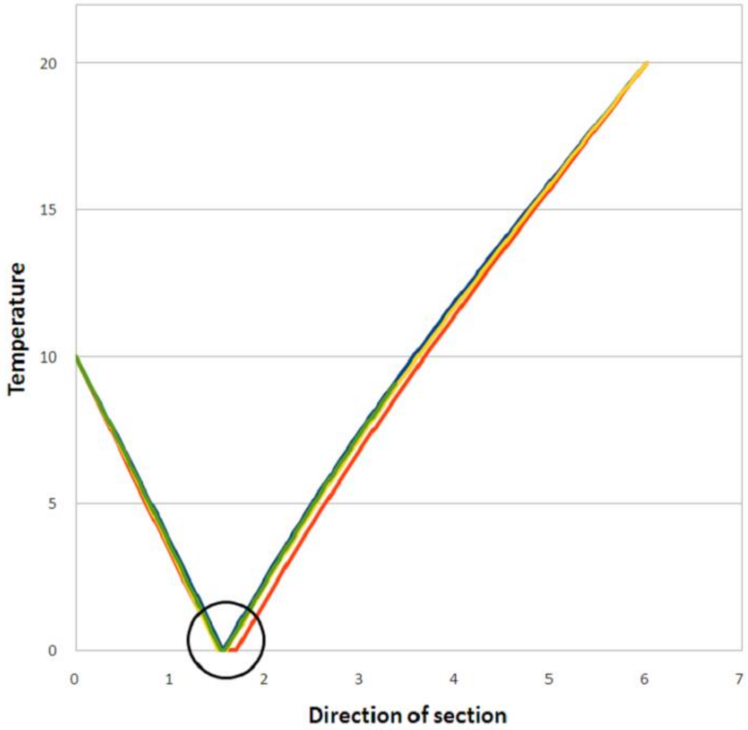
•*(T = 0)*

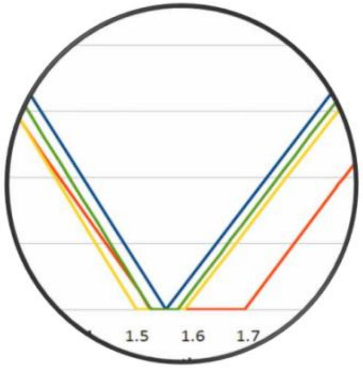
•Classic Finite Element Method

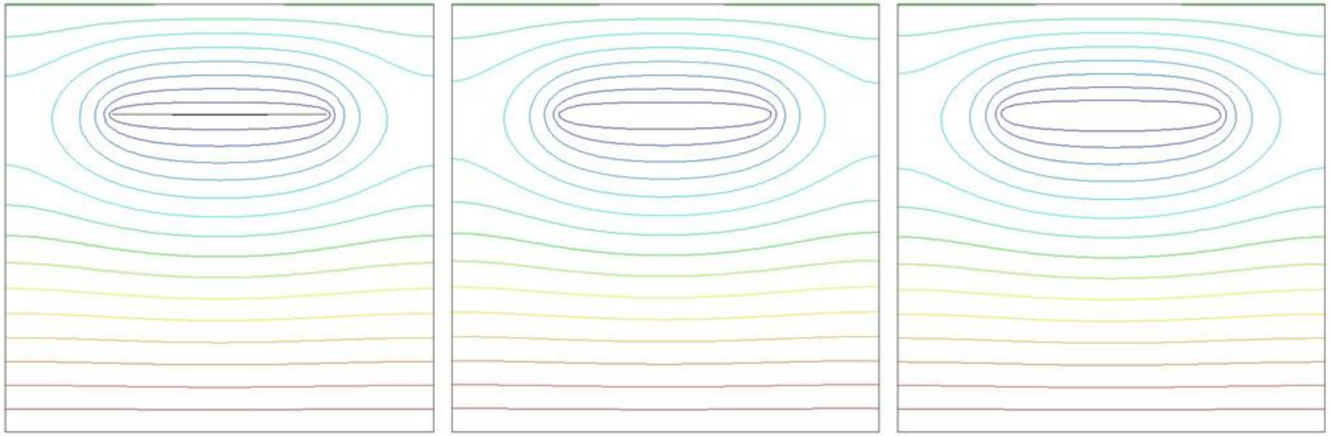
•Method I

•Method II









•showing contour lines of the temperature where the value of the temperature is imposed to zero

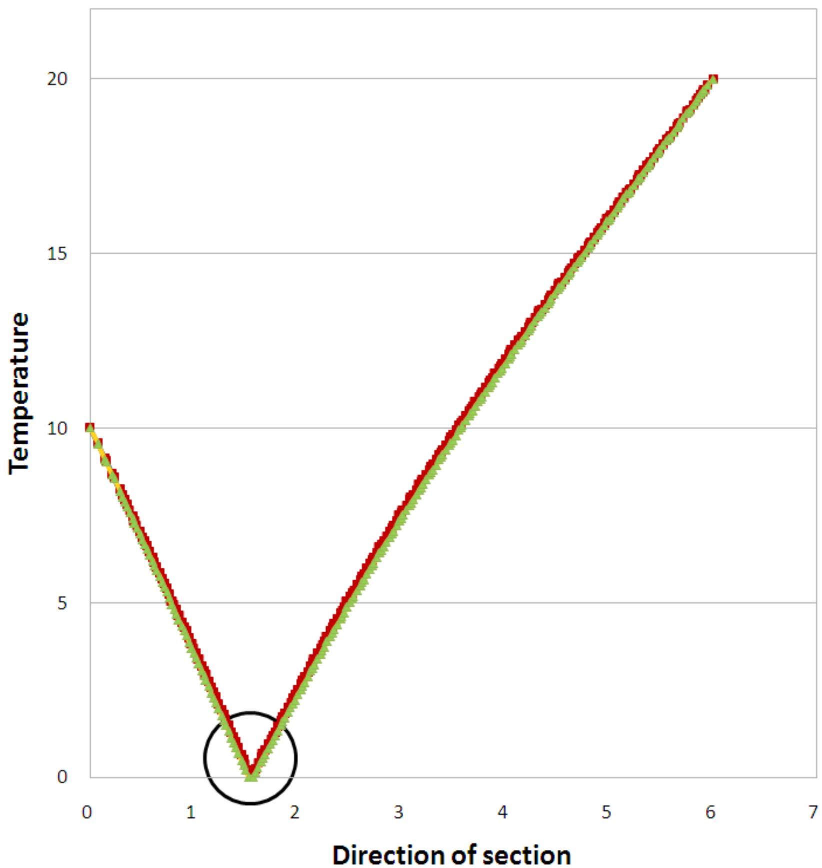
•*(T = 0)*

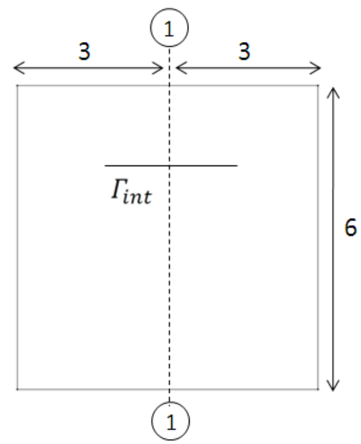
•Classic Finite Element Method

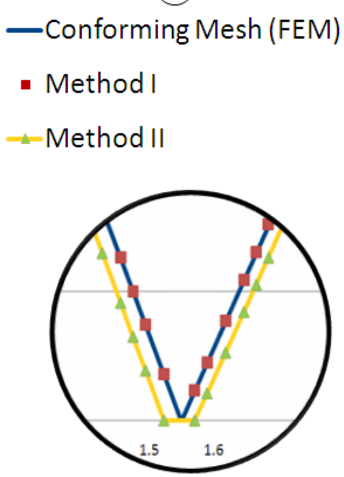
•Method I

•Method II

METHOD II

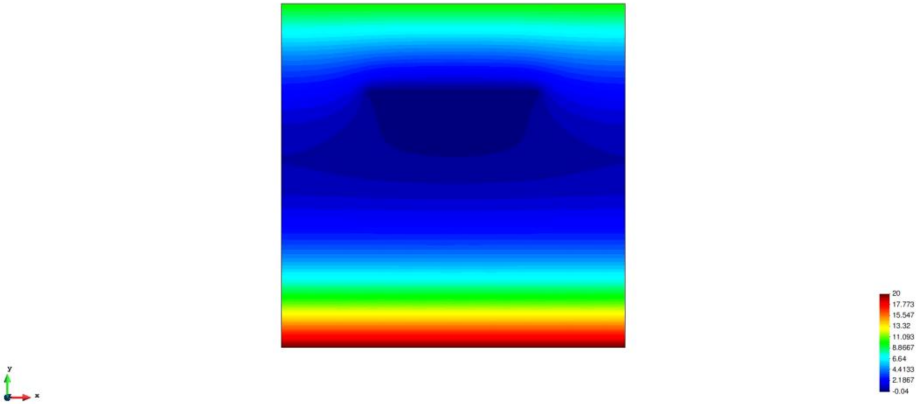




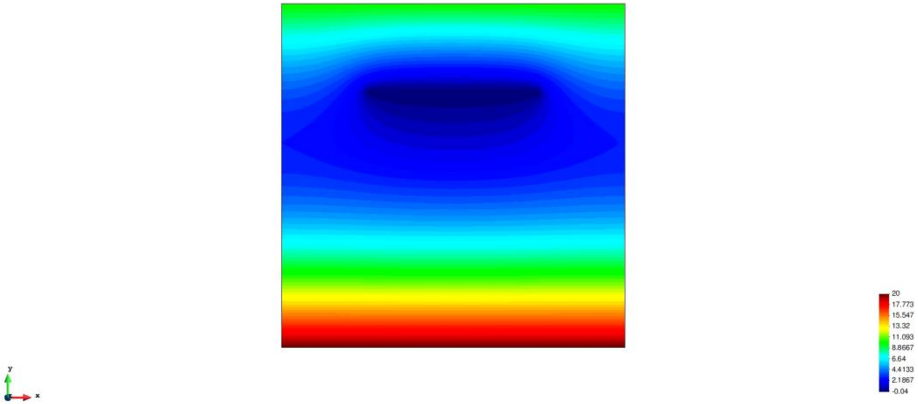


ANIMATIONS

Results in animate form where the **Dirichlet** boundary condition is imposed



- Method I

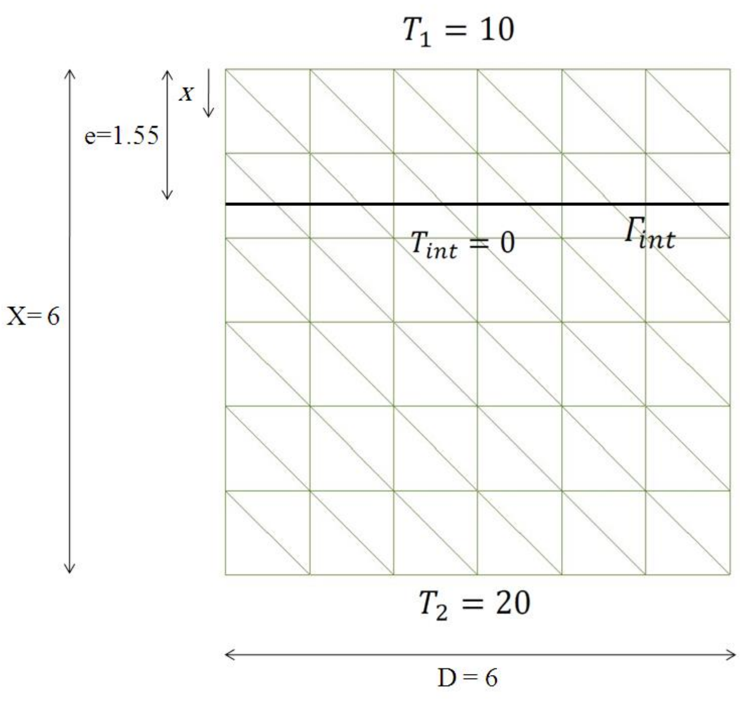


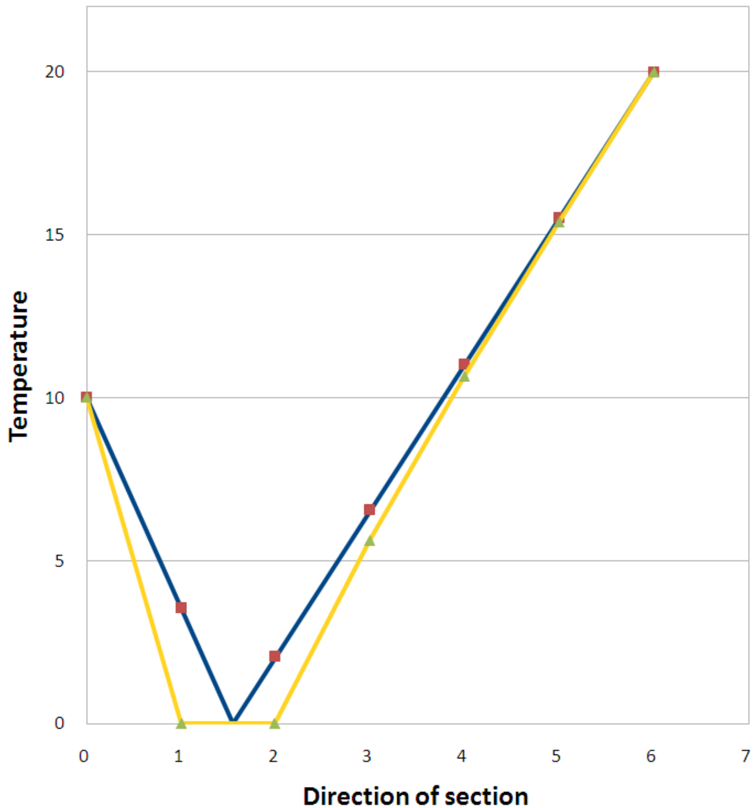
- Method II

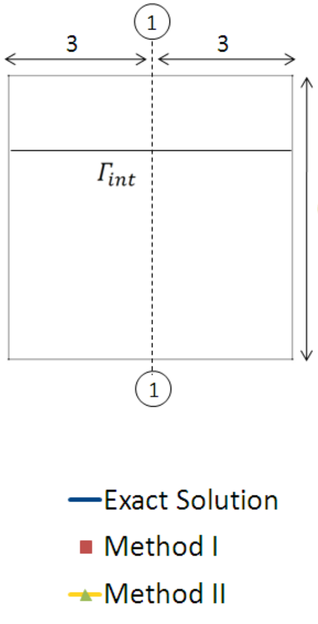
Test with exact solution

- Thermal conductivity $K = 1.0$ (for the entire domain)
- The temperature is enforced to be zero at the interface
- Results obtain from our proposed methods are compared with Exact Solution

•Exact Solution
$$\begin{cases} T_1 - \left(\frac{T_1}{e}\right) \times x & for\ x < e \\ 0 & for\ x = e \\ \frac{T_2}{(X-e)} \times (x-e) & for\ x > e \end{cases}$$

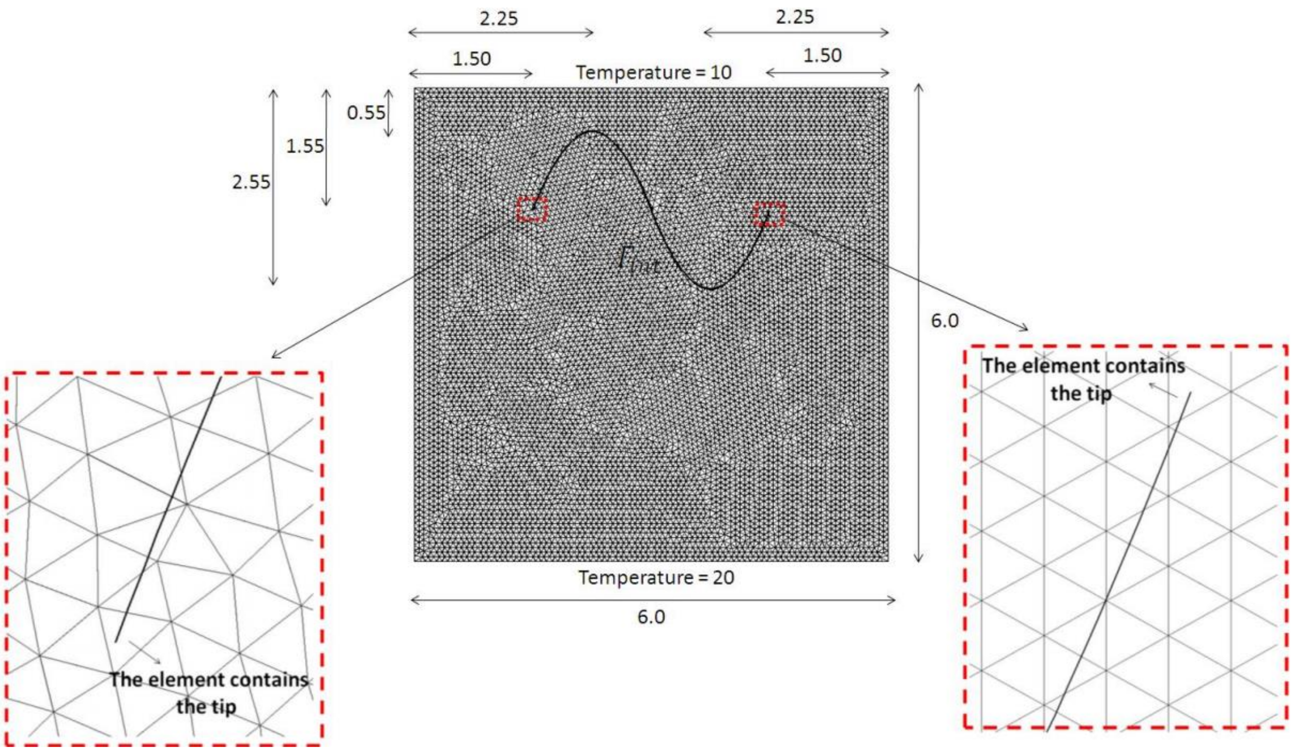




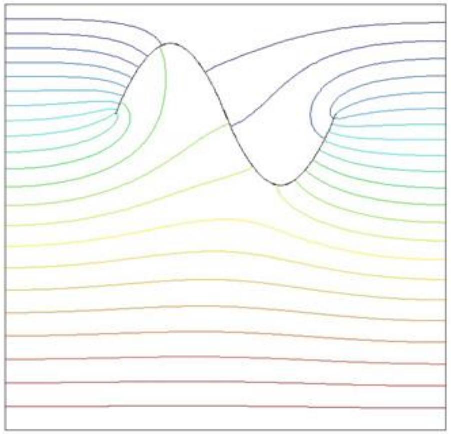


•*Arbitrary shape*

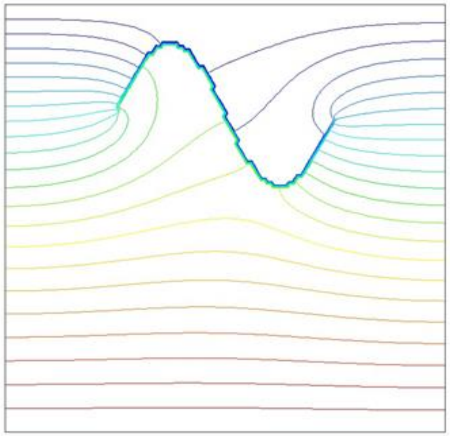
•Thermal conductivity K = 1.0 (for the entire domain)
•Both the temperature and its gradient are enforced to be zero separately
•Results obtain from our proposed methods are compared with results of classic finite element method where the internal interface is matched by the mesh



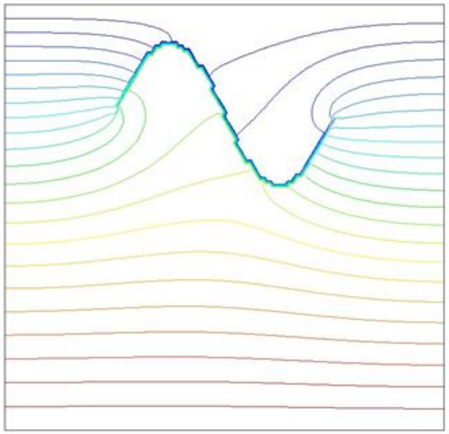
•Classic Finite Element Method

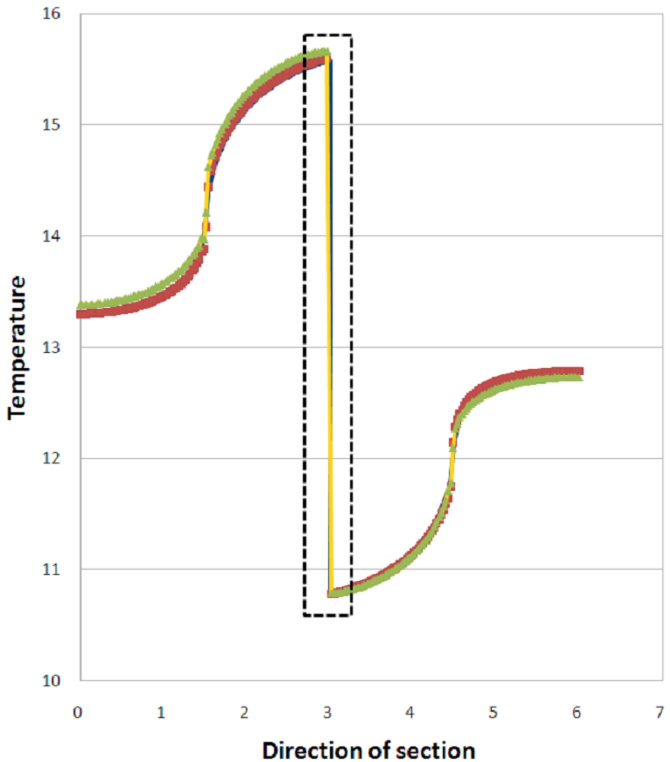


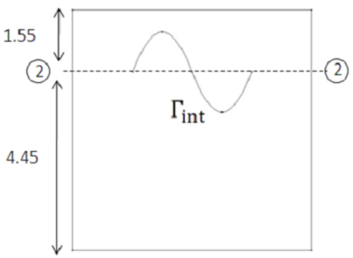
•Method I

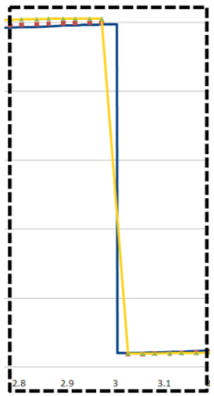


•Method II

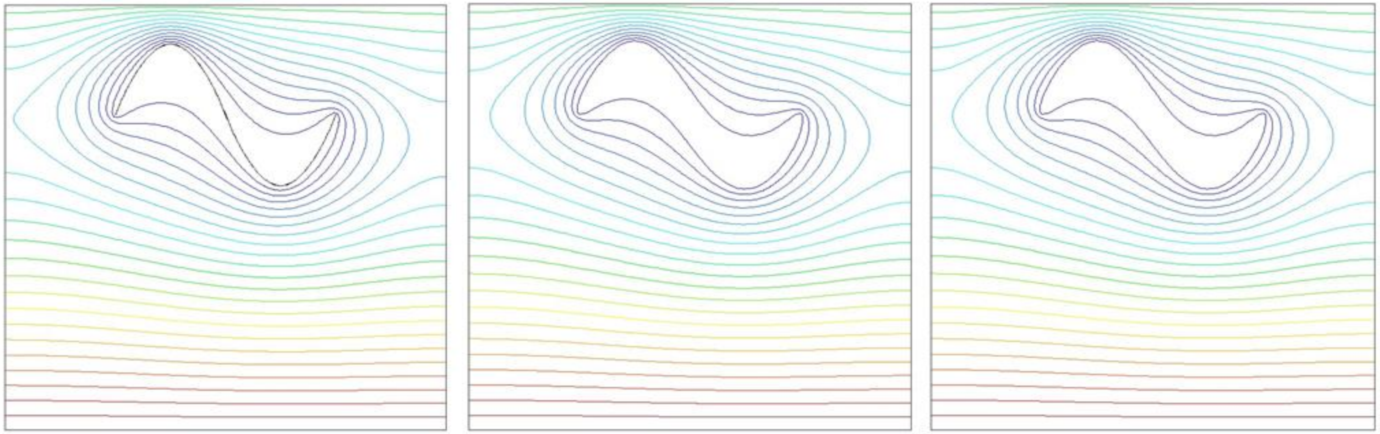


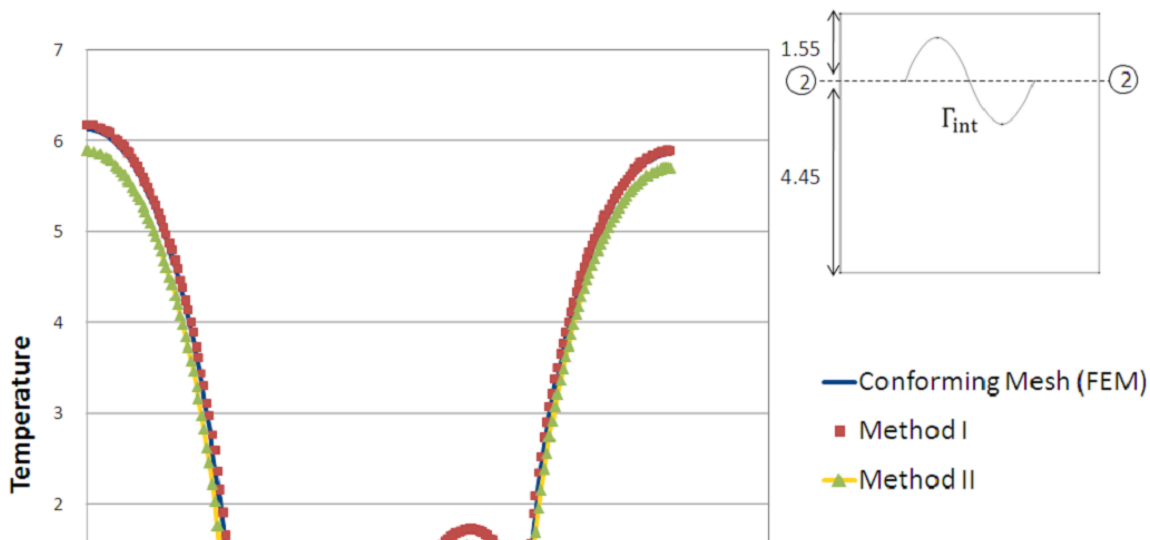


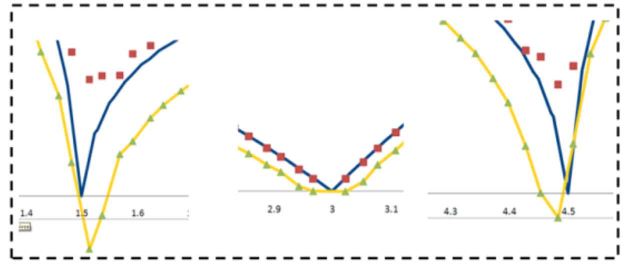




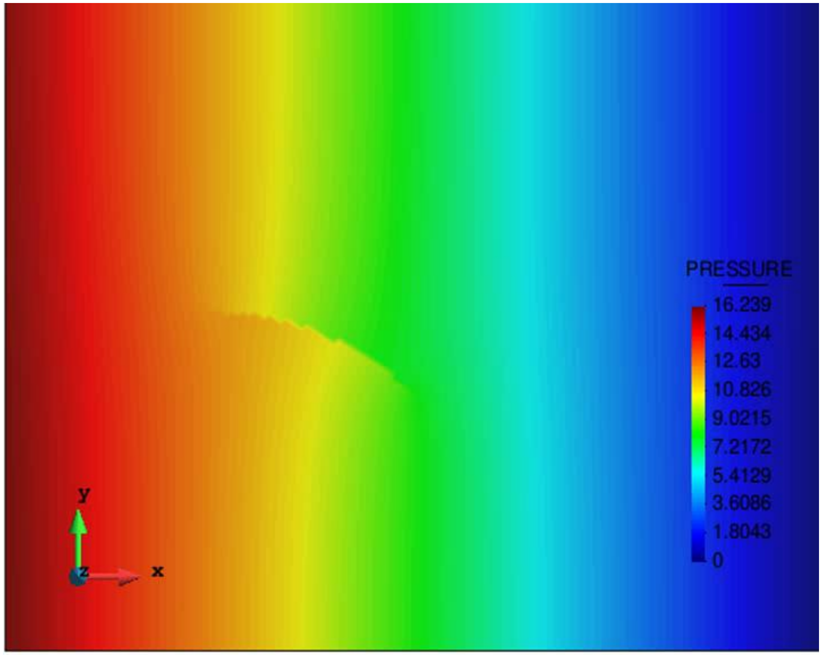
•Classic Finite Element Method •Method I •Method II

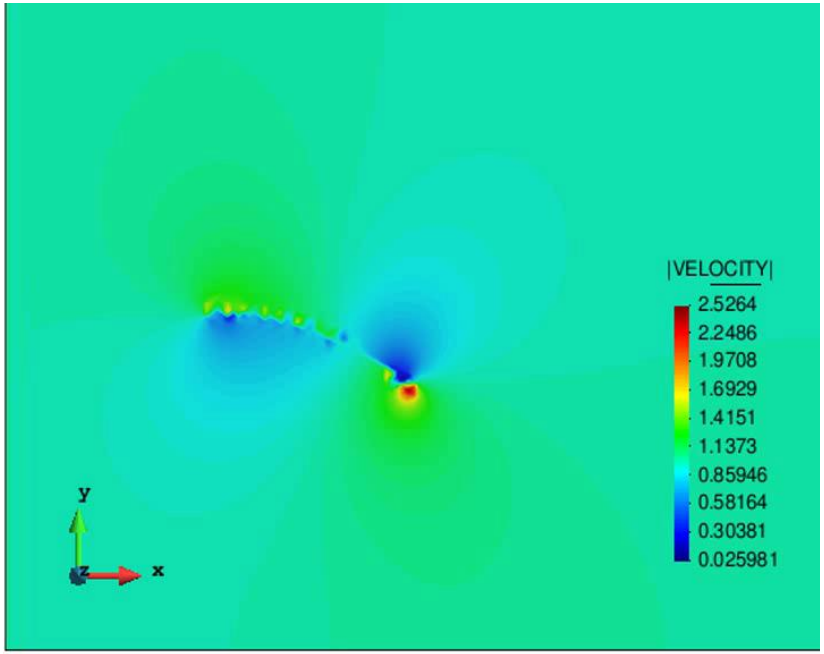






Same method can also be applied to CFD





"Industrial" examples at the presentation of Dr. Antonia Larese – MS042A

On Wednesday 14-16

Conclusions

A mixed u-e formulation is being investigated for CFD applications

Two possible approaches are investigated to include an object into the solution of a “cfd” problem

Our hope is to combine all in one…

Acknowledgments:

We wish to thank the support of the ERC through projects:

- *uLites FP7-314891*
- *Numexas FP7-611636*