A comparison between two adaptive numerical methods for edge plasma fluid modeling

F.Subba[a], O.V.Batishchev[b], R.Zanino[a]

December 1998

Plasma Science and Fusion Center
Massachusetts Institute of Technology
Cambridge MA 02139 USA

[a] Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
and Istituto Nazionale di Fisica della Materia, Italy
[b] Also at: Lodestar Research Corporation
Boulder CO 80301 USA, and Keldysh Institute for
Applied Mathematics, Moscow 125047 Russia

Abstract

Fluid simulation of edge plasmas is a challenging task due to several reasons. Firstly, regions characterized by the presence of strong gradients form often near the divertor plates, due to the interaction of the plasma with the neutral atoms and ions impurities coming from the solid walls. These fronts move with time, and must be well captured by the computational method in order to obtain accurate descriptions of the plasma edge physics. Moreover, any attempt to realistically simulate the tokamak edge environment must face the complex Scrape-off-Layer geometry. For the computations to be efficient, it is important to use grids well adapted to the shape of the domain and the features of the represented functions, which are not known before the calculations. For this reason, many efforts have been devoted in the past to provide simulation codes with adaptive meshes. In this work we compare two adaptive numerical methods developed independently by the authors. This is done by studying a set of test problems. For the more simple cases studied, the analytical solution is available, providing an accurate benchmark for the code performances.

1   Introduction

Obtaining accurate fluid simulations of the edge plasma region is very difficult. One of the main problems is the presence in the plasma properties of strong and moving gradients near the solid walls, particularly in the vicinity of the divertor plates. Moreover, the shape of the tokamak devices is complex, and requires considerable efforts to be adequately fitted. Many important phenomena take place in the regions of strong gradients (*e.g.* radiation-recombination), so that it is critical to represent them in detail. Automatically adaptive meshes are an important tool to face this problem, because they allow to obtain a good resolution of the most interesting regions while keeping as low as possible the total number of nodes. Unfortunately, a completely satisfactory method for their generation is presently lacking, despite the many efforts devoted to it [Batishchev, 1999][Zanino, 1998]. An additional difficulty is represented by the strong anisotropies in the plasma transport properties. As a consequence, meshes well aligned with the magnetic field are required in order to estimate the gradients of the plasma parameters.

In the following, we compare two adaptive numerical methods developed independently by the authors to perform plasma fluid simulations. The layout of the paper is as follows. In section three we describe briefly the two codes used: FE/BL2D [Subba, 1998] and RRC [Batishchev, 1998]. Section four is devoted to the description of the mesh generation procedures. In section five we present the results obtained by solving independently some very simple problems in order to test the code accuracy. We also present some quantitative statements on the alignment required between the mesh and the magnetic field. In section six we describe some results obtained in a more realistic geometry. Finally, in section seven, we summarize the main points of the comparison.

2   Codes description

In this section we give some details about the numerical methods implemented in the two codes FE/BL2D and RRC. This will be useful to better understand the following comparison.

2.1   FE/BL2D

FE/BL2D is the coupling of the fluid code FE with the automatic adaptive generator of triangular meshes BL2D. It solves the higlhy nonlinear, time dependent, diffusion-convection equation:

$$u_t - (k_\perp u^\alpha u_\perp)_\perp - (k_\parallel u^\beta u_\parallel)_\parallel + div(\overline{a}u) = S(x, y, u) \tag{1}$$

where u is the dependent variable, $K_\perp = k_\perp u^\alpha$ and $K_\parallel = k_\parallel u^\beta$ are the components of the diffusion tensor, $\overline{a}$ is the convective velocity and S is a source term. If the dependent variable u is thought of as the temperature, then equation (1) can be a model of the energy transport in a plasma. Note that equation (1) has been written in a coordinate system whose axis are locally aligned with (and perpendicular to) the magnetic field.

FE/BL2D solves equation (1) with the finite elements approach. It advances in time using the theta method, so that it is possible to implement a number of different schemes, ranging from the explicit up to full implicit. In all the calculations presented in this paper,

the full implicit scheme was chosen. The discretization of equation (1) leads, at each time step, to a nonlinear algebraic system to be solved. This is done in two nested loops. In the outer one, the equations are linearized with the Newton's algorithm. As Newton's method converges fastly but needs a good first approximation of the solution, we increase its radius of convergence with a backtracking procedure. In the inner loop the linear system generated at each Newton iteration is solved with a preconditioned Generalized Minimum Residual method (GMRES). At the beginnin of the computations, a mesh over the domain and an initial condition are assigned. The initial condition is stored, and at each time step the relative variation of the last obtained solution from it is checked. When the change exceeds a user specified threshold, FE suspends temporarily the calculations and gives the control of the procedure to the mesh generator BL2D. This produces a new mesh adapted to the last solution, which is then interpolated on the new grid. After that, FE starts again. The solution interpolated on the new mesh from BL2D is taken as the new reference value to choose when a next updating will be necessary.

2.2   RRC

The approach used bu RRC (Recursive Refinement Coarsening) exhibits a composite nature. Overall it may be called adaptive grid finite-volumes. With this method, the unknowns are defined in the "centers" of the cells. The total number of cells may vary from the previous iteration to the next iteration of the mesh adaptation. The number of neighbors for any cell is not fixed. A cell is refined when the variation of any important function of the solution (e.g. $S(u)$) differs more than E from ANY of it's neighbors. A cell is coarsened when this difference is smaller than ϵ (provided $\epsilon << E$) for ALL the neighboring elements.

The scheme is conservative, because all fluxes through adjacent cell surfaces are set to be the same. This is done by using a continuous 6-point (2nd order) or 2-point (1st order) interpolation which includes two adjacent cells. Moreover, we keep the mesh aligned with the magnetic field. Thus, the parallel flux does not "contaminate" the perpendicular flux. The set of implicit algebraic equations is solved by in iterative over-relaxation method, which is used in explicit-implicit modification. The convergence of iterations (mesh and/or coefficients adjusted) for the non-linear analytical problem of section 4.2 is fine (shown in Fig 1)

3   Mesh generation

3.1   BL2D

BL2D is an automatic adaptive generator of triangular meshes. It was developed aiming at the needs of the aeronautical community [Bourbaki, 1997], but shown good performances also for plasma simulations. It uses the Delaunay's method to generate a triangular grid. BL2D measures the distances according to a local, user specified, metric. The grid will then be almost equilateral (with size length $\simeq 1$) if measured on the local (non euclidean) metric. This will appear as an anisotropic grid in the euclidean plane. The user influences the grid

generation process by specifying a suitable metric map on the computational domain. This is done in FE/BL2D according to the following steps:

- First, for each grid point a map of the hessian H of the last obtained solution is determined. Il l is the local characteristic grid size, $\bar{l}H\bar{l}$ will provide an estimate for the local interpolation error.

- To specify the metric it is sufficient to choose its eigenvalues and eigenvectors. A possible choice could simply consist in taking the absolute values of the eigenvalues of H and the same eigenvectors. This would result in a uniform distribution of the interpolation error. Let's call M_1 the matrix map defined in this way.

- If λ_i are the eigenvalues of M_1 , define the norm of M_1 as $\|M_1\| = ((\lambda_1)^2 + (\lambda_2)^2)^{1/2}$. This will set a local characteristic size for the grid spacing. Let μ_1 and μ_2 be the maximum and minimum of $\|M_1\|$. It is desirable to control the range of allowable grid spacings. This can be done by specifying the corresponding range of acceptable metric norms, let it be (m_1, m_2). Then, rescale locally (with scale factors depending on the local value of $\|M_1\|$) the matrix M_1 so as the interval (μ_1, μ_2) will be superposed to (m_1, m_2).

- Finally, produce the metric M by stretching locally the eigenvalues of M_1 (at constant norm) so as the unit (according to the metric M) lengths in the directions perpendicular and parallel to the magnetic field will appear to have a user determined ration in the physical plane.

3.2   RRC

The initial mesh used by RRC is created by importing the A- and G- files from EFIT. Then, the grid generator finds the magnetic axis, the X point and the strike points. After that, it analyzes the magnetic flux (for the case considered here it was given by 33x33 arrray), and distributes the nodes along the magnetic surfaces. At this point, it creates a connectivity list by keeping two sides of each quadrilateral parallel to the magnetic field. Finally, it converts the initial structured data into the format adopted by RRC code. Fig 2 and Fig 3 illustrate the logic used by the grid generator.

The initial mesh so created is structured (regular), as shown in Fig 4. Thus, this approach can be easily applied to any existing structured meshes. During the calculations, each cell can be subdivided into four finer cells if the local gradient becomes high. This process is continued until saturation, as can be seen from Fig 5 and 13a. If the gradient becomes small, then the coarsening of the mesh may happen. Thus, the algorithm is flexible and reversible.

4   Test problems

In this section we describe some simple test problems that we solved independently with both the codes in order to check and compare the accuracy and reliability of the two methods. We also present some quantitative statements about the degree of alignment required between the mesh and the magnetic field in order to obtain good evaluations of the fluid fluxes.

4.1 Bifurcated solutions

We solved the equation:

$$(K_{\parallel}(u)u_y)_y = C\frac{u}{u_0}\exp(-(\frac{u-u_0}{\sigma})^2) \tag{2}$$

on the square $(x,y)\epsilon(0,1)\times(0,1)$ and with $K_{\parallel}(u) = k_{\parallel}u^{2.5}$. The boundary condition at the sides $x = 0$ and $x = 1$ were $u_x = 0$. At the other two boundaries, we imposed

$$-K_{\parallel}(u)u_y|_{y=0} = \gamma u, \qquad -K_{\parallel}(u)u_y|_{y=1} = -1 \tag{3}$$

Note that, even if it was solved on a bidimensional domain, this problem actually reduced to be onedimensional. It can be shown that the problem (2)-(3) can admit two coexisting solutions, depending on the initial condition [Batishchev, 1999]. If the initial temperature profile is sufficiently high, the source is almost inactive. Then the energy entering from the side $y = 1$ diffuses across all the domain and exits from the side $y = 0$. In this case, the steady temperature profile is almost flat along all the domain; let's call this the *hot* solution. Conversely, if the initial profile is low, the energy cannot be transmitted across the boundary $y = 0$, because the diffusion coefficient is strongly reduced. Then the temperature incrases until the source term activates and radiates almost all the incoming flux. We will speak in this case of the *cold* solution. We solved this problem independently with the two codes using the parameter values: $k_{\parallel} = 10^{-4}$, $C = 50$, $u_0 = 15$, $\sigma = 4$ and $\gamma = 0.03$. In Fig 6 we compare the hot and cold solutions obtained with the two codes FE/BL2D and RRC.

4.2 Functions with variable sharpness

We solved the equation

$$-(k_{\perp}u^{\alpha}u_x)_x - (k_{\parallel}u^{\beta}u_y)_y = S \tag{4}$$

on the square $(x,y)\epsilon(0,1)\times(0,1)$ and Dirichlet boundary conditions at the four sides. The source term was chosen to be:

$$S(u,x,y) = \begin{cases} \alpha u^{\alpha-1}(u_0)(c^2\delta^2(u(\alpha+1)-\alpha u_0)-ua)+ \\ \beta T^{\beta-1}(u_0)(c^2(u(\beta+1)-\beta\epsilon)-ub) & , \ y < y_f(x) \\ u^{\alpha-1}(u-1-u_0)(d^2\delta^2(\alpha(u-1-u_0)+u)-eu)+ \\ u^{\beta-1}(u-1-u_0)(d^2(\beta(u-1-u_0)+u)-gu) & , \ y \geq y_f(x). \end{cases}$$

with

$$y_f(x) = y_v + r|x - x_v|^s \tag{5}$$

The coefficients a, b, c, d, e, g are defined as

$$a = \sigma p(y_f - y)^{p-2}f,$$
$$b = \sigma p(p-1)(y_f-y)^{p-2},$$
$$c = \sigma p(y_f-y)^{p-1},$$
$$d = \sigma p(y-y_f)^{p-1},$$
$$e = \sigma p(y-y_f)^{p-2}f,$$
$$f = (p-1)\delta^2 + (y_f-y)\Delta,$$
$$g = \sigma p(p-1)(y-y_f)^{p-2}$$

4

where $\delta = rs(x - x_v)^{s-1}$ and $\Delta = rs(s-1)(x - x_v)^{s-2}$

Despite its apparent complexity, this source term is extremely useful, because the corresponding solution is:

$$u(x,y) = \begin{cases} 0.5\exp(-\sigma(y_f(x) - y)^p) + u_0 & \text{if } y \leq y_f(x) \\ 1 - 0.5\exp(-\sigma(y - y_f(x))^p) + u_0 & \text{if } y \geq y_f(x) \end{cases} \quad (6)$$

Formula (6) describes a function depending on the seven parameters σ, p, u_0, y_v, r, x_v and s. The contours of this function show a characteristic **V** shape, with gradient strentgh varying according to the chosen parameter values. Consequently, it provides a very useful test to determine the numerical performances of any code, because its difficulty can be varied by the user. With the parameter values $\alpha = \beta = 0$, $\sigma = 1$, $p = 2$, $r = 2$, $x_v = 0.45$, $y_v = 0.35$, $k_{\parallel} = k_{\perp}$ both codes could reproduce the analytical solution with an average error of $O(10^{-3})$, but RRC shown a greater efficiency. With more sharp gradients, obtained by setting $\sigma = 10$, we could not manage to keep the error at an acceptably low level with FE/BL2D. In Fig 7 we compare the analitycal solution with those obtained by FE/BL2D and RRC.

4.3   Mesh alignment requirement

As we have previously mentioned, plasma modeling requires grids well aligned with the magnetic field, due to the huge anisotropies in the transport coefficients. For example, if some error are made in evaluating the gradients parallel to the magnetic field, this can result in an intolerable error in the parallel fluxes, because the parallel conductivities are big. Consequently, a good alignment level (defined as the minimum angle between the element sides and the magnetic field) is a key demanding in the mesh generation procedure. This is particularly true for FE/BL2D, because the Delaunay's type method implemented does not guarantee a perfect alignment. In order to assess quantitatively the sensitivity of the numerical scheme to the average misalignment, we mapped the function (6) on the rectangle $(x,y)\epsilon(0,1)\times(0,100)$ (by a linear transformation of the segment $y\epsilon(0,1)$ into $y\epsilon(0,100)$. Then we interpolated it on several uniform triangular meshes with comparable number of nodes and different misalignment levels produced by FE/BL2D. From the interpolated functions, we tried to reconstruct the fluxes $-K_{\parallel}(u)u_y$ along the boundaries y=0 and y=100. Table (1) shows clearly that, in order to keep the error in the calculated fluid fluxes to a level of no more than some percent, it is necessary to keep the mesh average misalignment at less than 1 degree.

Misaligment (deg)	Nodes	Rel. error at y=100	Rel. error at y=0
1.9026	7003	4.46	0.47
0.7432	7660	0.05	0.46
0.1907	7173	0.03	0.73

Table 1: Relative errors on the fluxes obtained by interpolating the function (6) on differently aligned grids. It can be seen that the precision at $y = 100$ increases strongly with the alignment, while this is not true at $y = 0$. This is probably due to the strong difference in the fluxes at the two sides: $q_{y=0} = 0.5751 \times 10^{-3}$, and $q_{y=100} = -36.9$. Getting at $y = 0$ the same error level as at $y = 100$ would require a much finer grid.

5 Tests in realistic geometry

In this section we describe some preliminary results obtained with the two codes on realistic geometries. Up to now, no problem in complex geometry has been attacked with both the codes contemporarily. Consequently, this section is not intended to provide a comparison between the different methods, but only to show some of the possibilities of FE/BL2D and RRC and to provide some sugestions for fututre developments

5.1 Boundary layer widening

One of the problems related to the misalignment between the mesh and the magnetic field is the introduction of numerical diffusivity perpendicular to $\overline{B}$. To study the capacity of BL2D to generate good quality meshes, we solved the equation

$$u_t + (k_\| u^{2.5} u_\|)_\| = 0 \qquad (7)$$

on the domain shown in Fig 8. It represents the portion of the scrape-off-layer region of a tokamak from the divertor leg up to the midplane. The initial condition was a stepwise function assuming two values, with the discontinuity perfectly aligned with the magnetic field. As the diffusion was only parallel to the magnetic field. it is clear that the initial condition is equal to the analytical steady state solution. If this problem is treated computationally, the discontinuity will relax in a boundary layer due to the abovementioned numerical diffusivity. The final thickness of this layer is extremely sensitive to the mesh quality.

We started the calculations with an initial uniform mesh of 6927 nodes, shown in Fig 8. During the solution process, BL2D reduced the number of nodes up to the final amount of 883, keeping them mainly in the region of the discontinuity. Moreover, due to the good alignment with the magnetic field, the final boundary layer still had a transverse dimension comparable with the initial one (which, beiing the interpolation of a discontinuity, was equal to the transverse spacing of the initial grid).

5.2 Pure convection test

Problems of pure convection provide severe tests for any adaptive computational scheme based on a remeshing strategy. In fact each mesh adaptation involves an interpolation

process which smoothes the solution. We studied the capacity of FE/BL2D to keep this effect at a low level by solving the following equation:

$$u_t + div(\overline{a}u) = 0 \tag{8}$$

where the speed $\overline{a} = a\overline{b}$ had constant modulus. The domain was the same as that of the previous section, shown in Fig 8. The initial condition (see Fig 9) was the superposition of a bell shaped *hole* on a flat function located near the divertor plate ($u_{max} = 2, u_{min} = 1$). This may be thoutgh of as a very rough representation of a Marfe instability. Fig 9 shows how the perturbation propagated up to the vicinity of the X point, where we interrupted the simulation. The initial mesh was uniform with 9936 nodes. Again, the adaptivity process allowed a strong reduction in the number of nodes, down to the final value of 829. It can be noticed that the initiallly flat plateau was modified during the simulation. This is due to the presence of the toroidal component in the magnetic field.

5.3   Mesh evolution for a full poloidal cross section

We illustrate the capacity of RRC to fit complex geometries by showing the evolution of an initial mesh to fit the full poloidal section of Alcator C-mod. The mesh can be non-orthogonal or quasi-ortogonal, but we find that an hybrid mesh is optimal (see Fig 10). The evolution of the source S(u) with the increase of the level of grid refinement is shown in Figs 11 through 13.

6   Conclusions

In this paper we compared two different adaptive methods developed to perform plasma fluid simulations on the particularly complex tokamak edge geometry. FE/BL2D is the coupling of a finite elements code with an automatic adaptive generator of triangular grids, while RRC uses the finite volumes method and, up to now, has performed calculations on quadrilateral elements. We made a preliminary comparison on some test problems solved on rectangular domains. In this case, RRC has shown to be more robust and faster than FE/BL2D, due probably to a combination of both the conservativeness of the finite volumes scheme and the facility to fit rectangular domains with quadrilateral elements. A similar comparison on more complex geometries is still lacking, even if the two codes have both been used to study some problem on realistic domains. Such a comparison should be the object of a future work. In particular, it would be interesting to understand how the robustness properties of the finite volumes method are retained when the computational domain assumes irregular shapes and whether quadrilateral meshes are still satisfactory in this case, or it could be desirable to couple the two approaches, using perhaps a finite volumes scheme on triangular grids.

7   Acknowledgements

8 Bibliography

[1] O. V. Batischev et al, *Contrib. Plasm. Phys.* **38** , 361 (1998); *MIT Report, PSFC/JA-98-21* (1998); J. Plasma Physics (1999).

[2] O. Batishchev, *Bulletin APS* , Vol. 43, No. 8, G4Q41 (1998).

[3] F. Subba, O. Batishchev and R. Zanino, *Bulletin APS* , Vol. 43, No. 8, G4Q49 (1998).

[4] R. Zanino, *J. Comput. Phys.* **138** , 881, (1997)

[5] R. Zanino, G. Belgiorno and F. Subba, *Numerical simulation of a 2-D conduction convection radiation model problem in divertor geometry using adaptive finite elements,* to be published in *Czechoslovak journal of physics* (1998).

[6] R. Zanino and F. Subba, *Contrib. Plasma Phys.* **38** (1998) 355.

[7] H. Borouchaki, et al., *Finite Elements in Analysis and Design* **25** , (1997), 61.

9 Figure captions

Fig 1. Example of convergence history for RRC code for the problem of section 4.2.

Fig 2. $\Psi(R, Z)$ map used to produce the initial mesh for RRC.

Fig 3. Insertion of points on the $\Psi(R, Z)$ map to create the actual grid.

Fig 4. Superposition of different refinement levels on the initial structured mesh of RRC code.

Fig 5. Enlarged view of a sector of the refinement zone of the mesh shown in FIg 4.

Fig 6. a)Profile and contour plot of the *hot* solution with bifurcations of section 4.1 as obtained by RRC code. IT is also shown the source intensity contour plot. b) Same as a) for the *cold* solution. c)*Hot* and *cold* solutions obtainhed by FE/BL2D. The parameter values used for this run are: $C = 50$, $u_0 = 15$, $\sigma = 4$, $q_{y=1} = 1$, $\gamma = 0.03$.

Fig 7. a) Solution of the analytical problem (section 4.2) as found by FE/BL2D. b) Top: same as a) found by RRC. Bottom: Analytical solution.

Fig 8. Top: final mesh for the problem of the layer widening described in section 5.1. Bottom: Steady state solution as found by FE/BL2D. The numbers near the different boundary tract mark the divertor plate (1), the first wall (2), the midplane (3), the plasma core (4) and the private region (5).

Fig 9. Top left: Contour plot of the initial condition for the pure convection test (section 5.2). Top right: final state. The bell shaped *hole* has moved near the divertor plate. Bottom left: the initial mesh (9936 nodes). Bottom right: the final mesh (829 nodes).

Fig 10. Exemple of *hybrid* initial mesh produced for RRC.

Fig 11. Evolution of the radiation function during RRC calculations. First stage, mesh with 0.8K elements.

Fig 12. Evolution of the radiation function during RRC calculations. Second stage, mesh with 1.6K elements.

Fig 13. Evolution of the radiation function during RRC calculations. Fifth stage: a) mesh with 12.3K elements, b) contours.

Convergence of numerical solution vs iterations for a test problem

Δ

N

computer accuracy
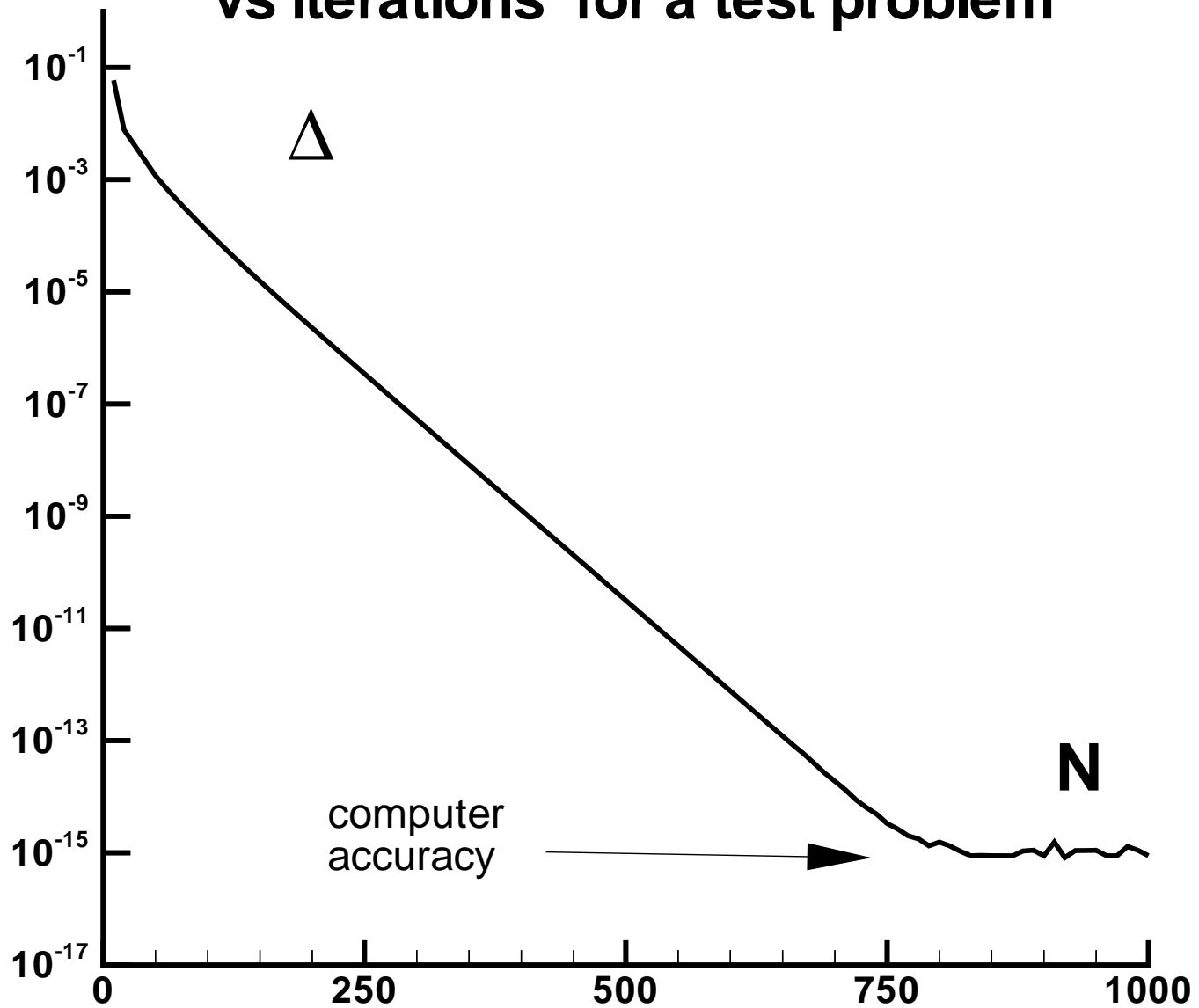
Fig 1

Mesh generator uses standard EFIT equilibria for C-Mod
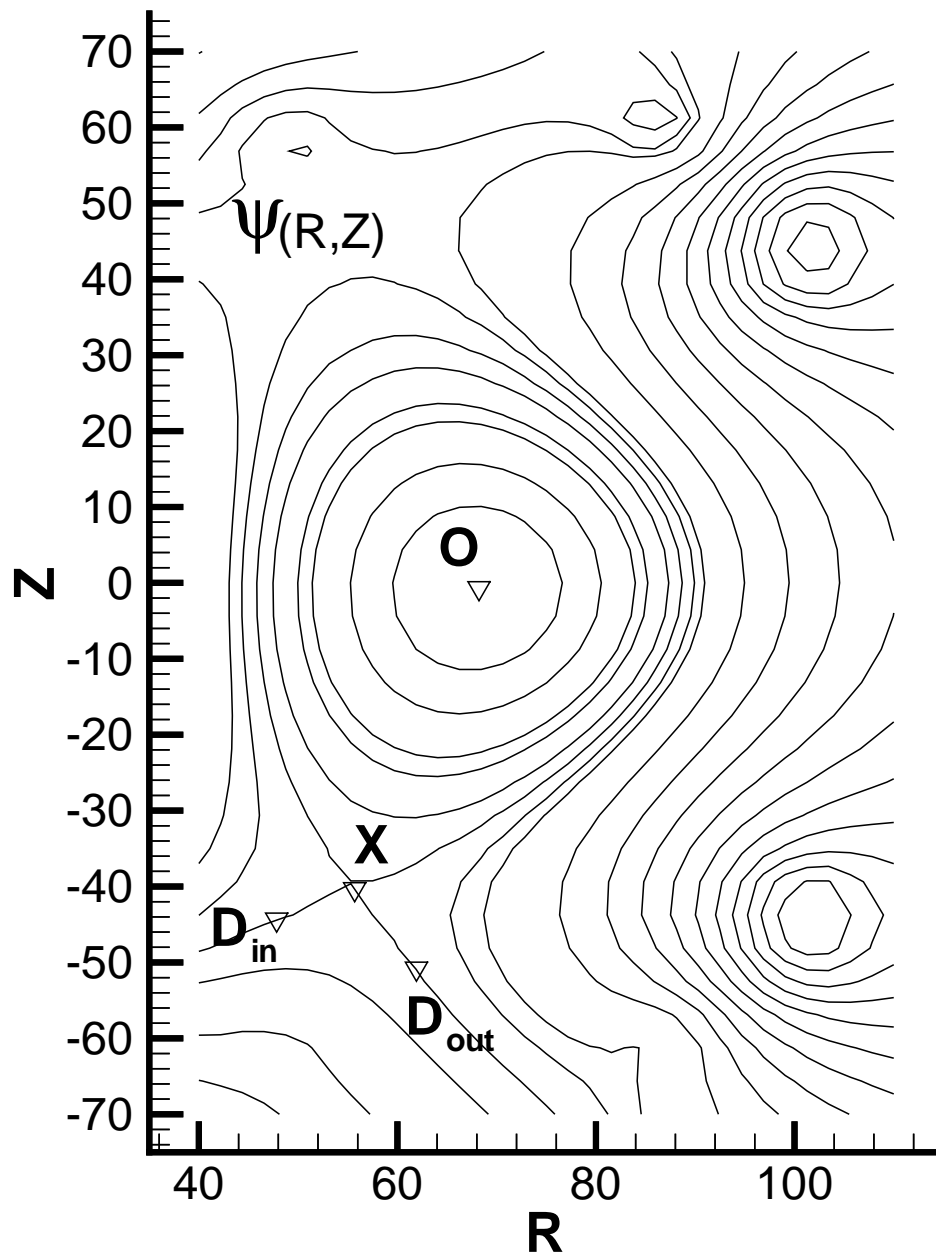
Fig 2

Mesh generator first finds nodes on selected flux surfaces
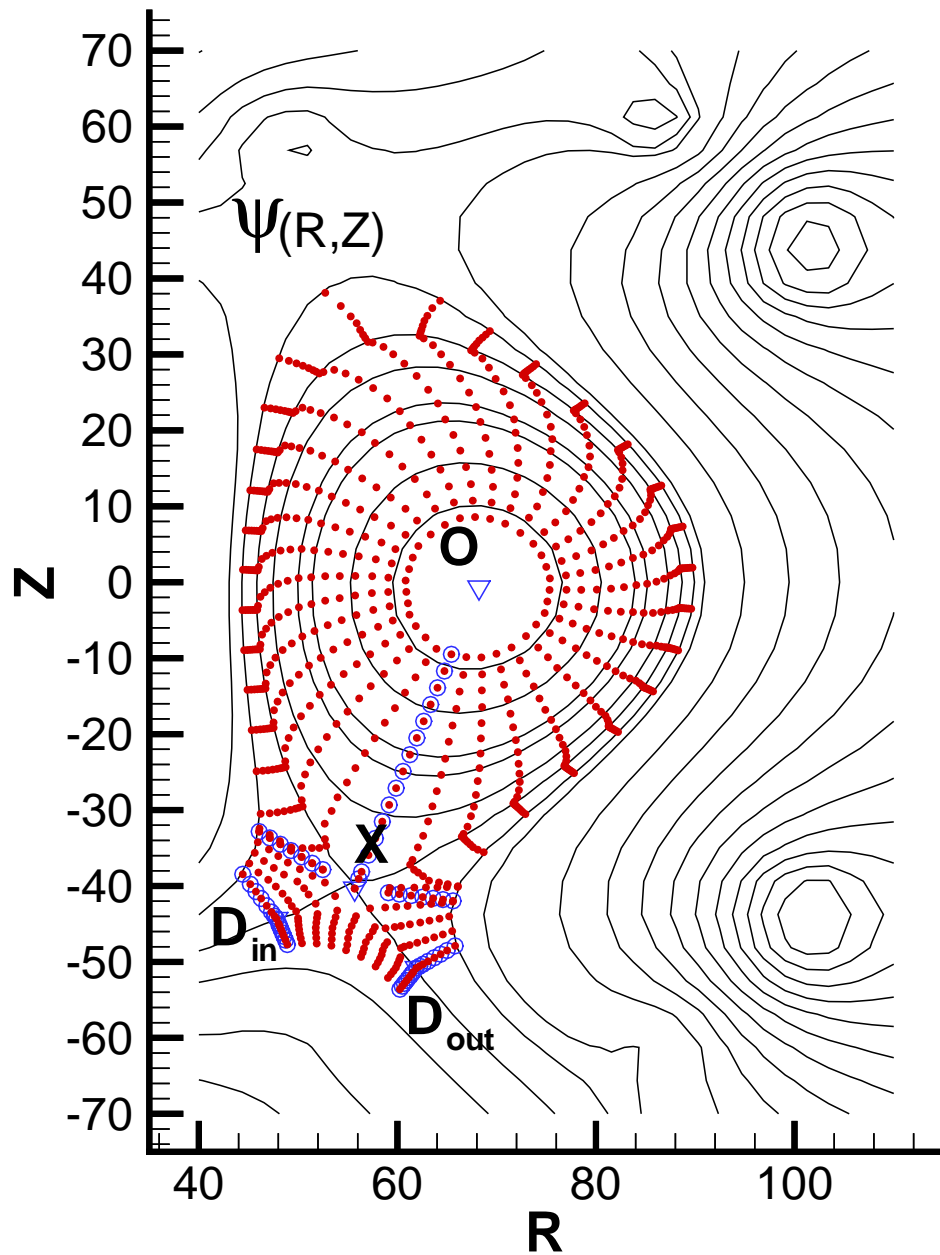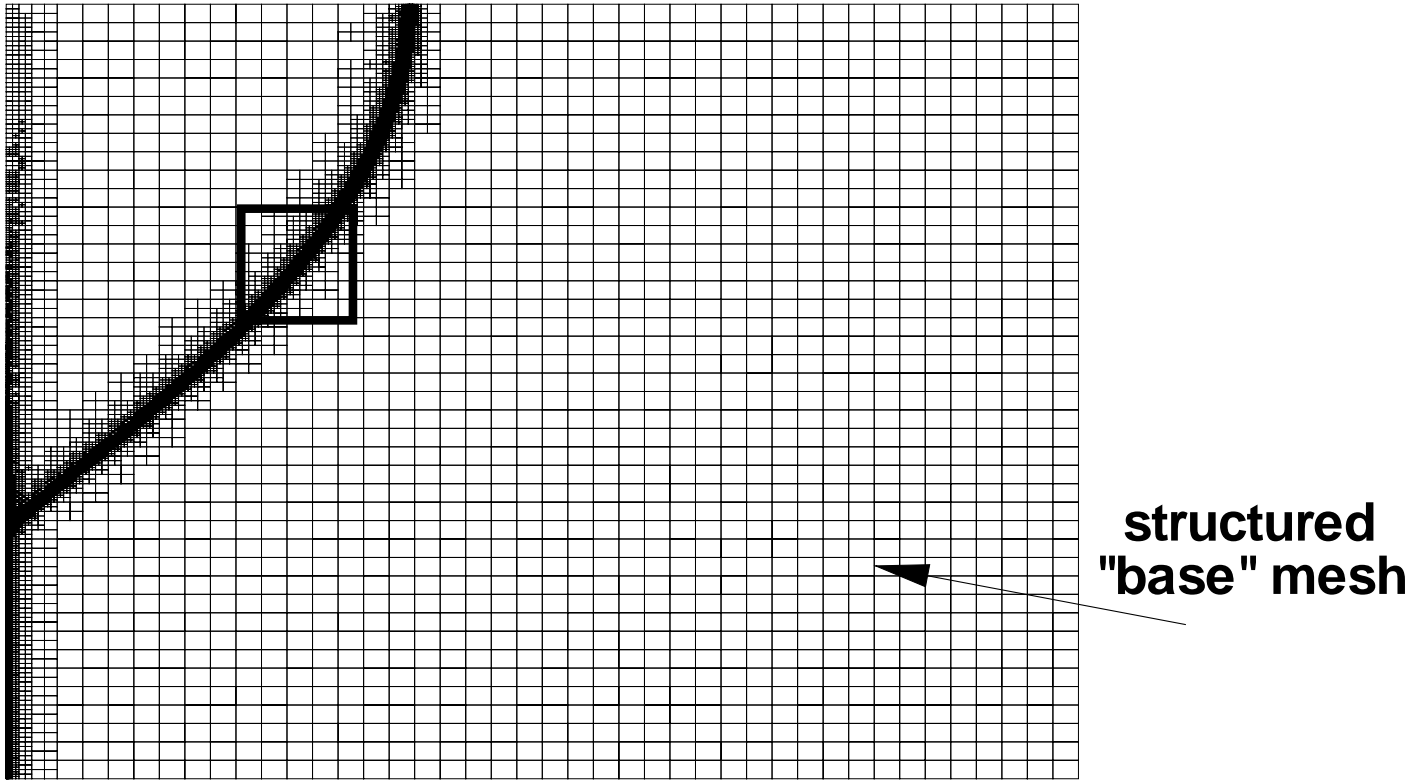
$\Psi_{(R,Z)}$

O

X

D$_{in}$

D$_{out}$

Fig 3

RRC mesh

structured "base" mesh

Fig 4

zoomed

Nth

3rd

2nd
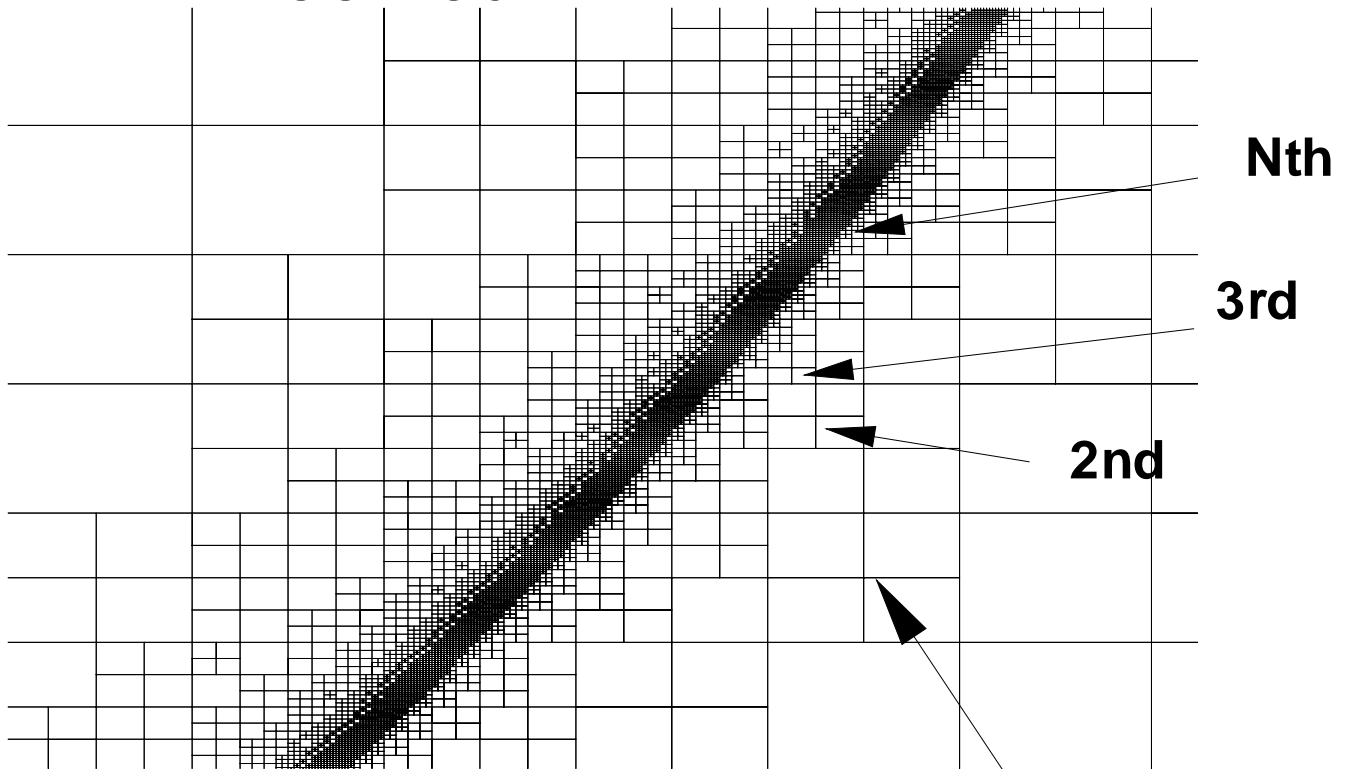
1st level of refinement

Fig 5
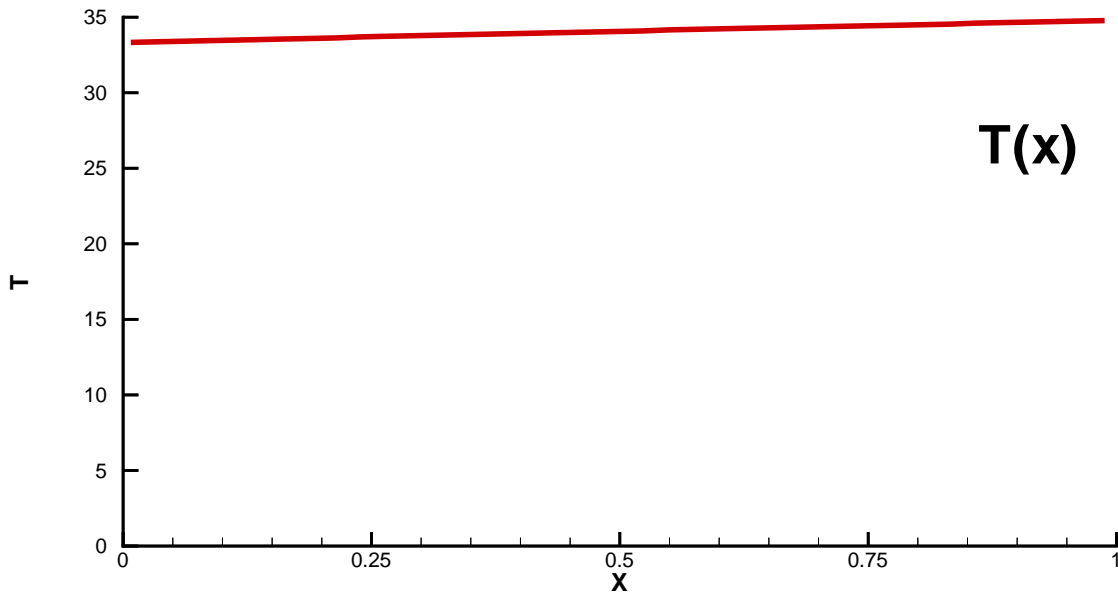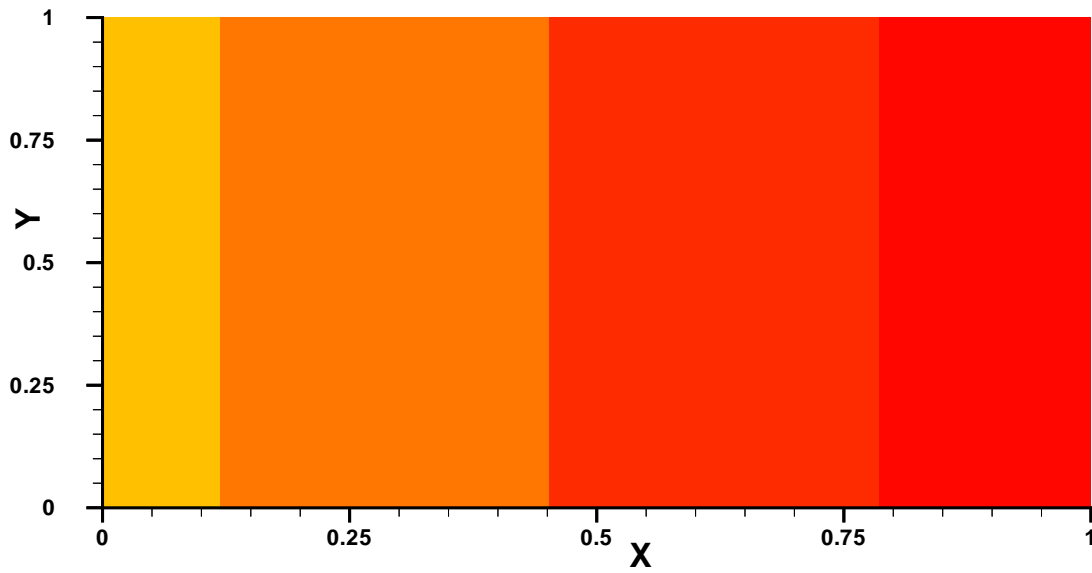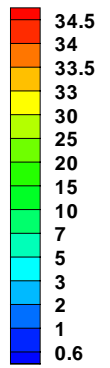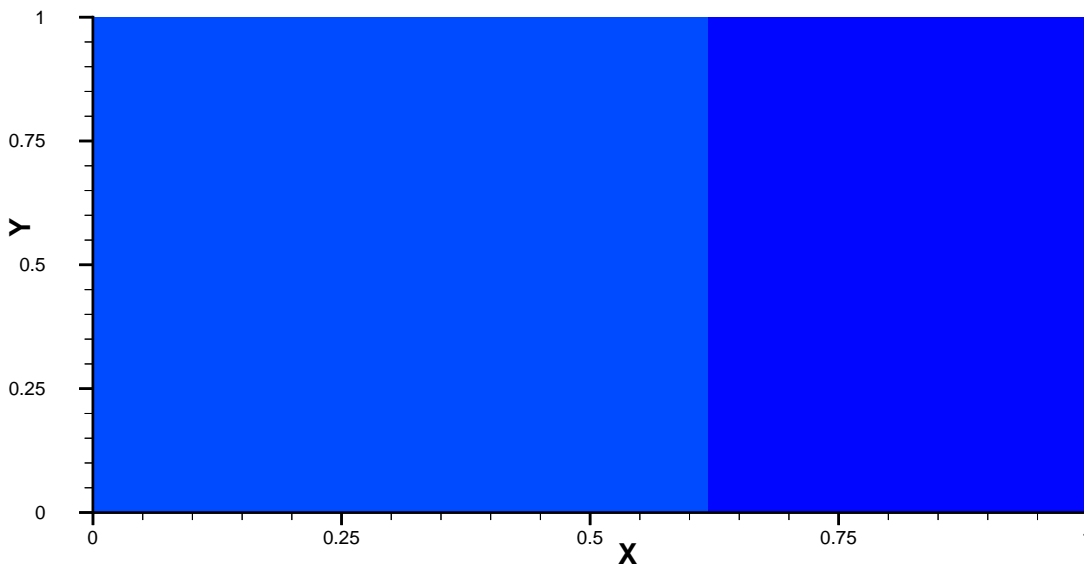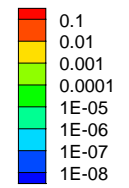
Fig 6a

COLD Numerical Solution

T(x)
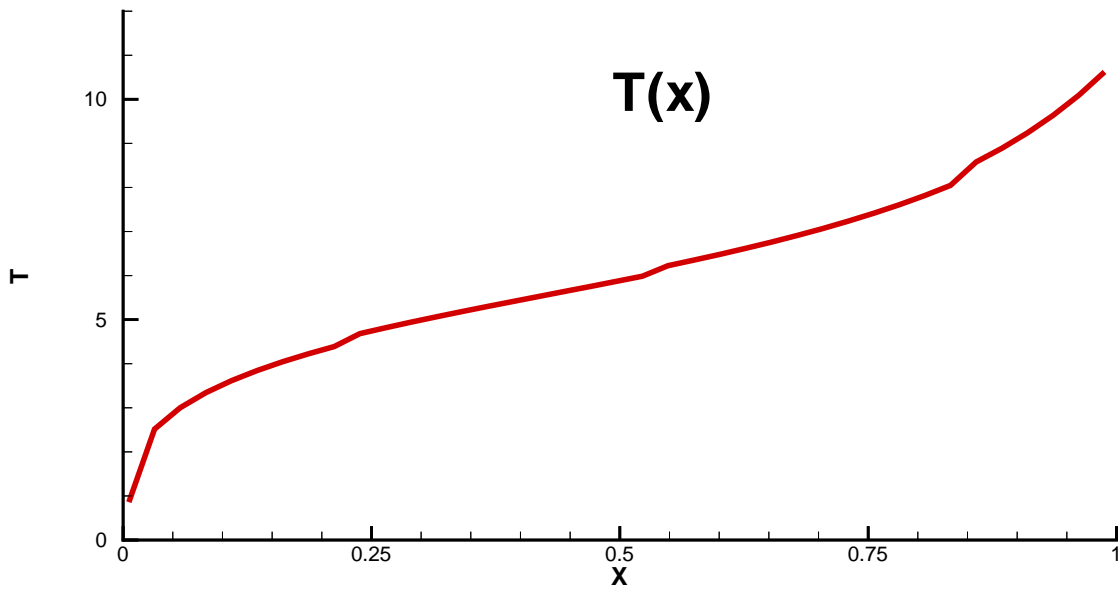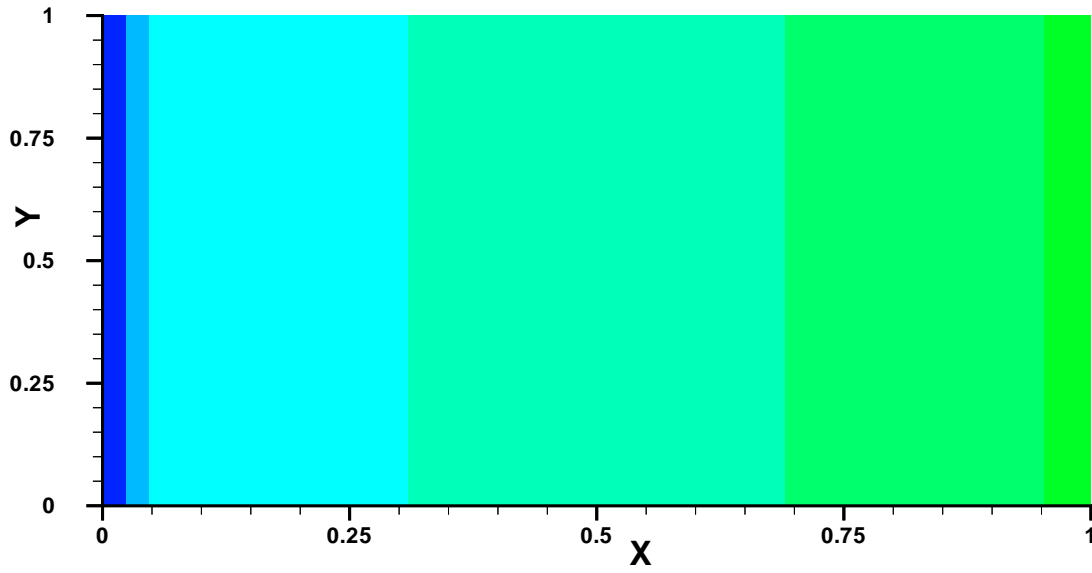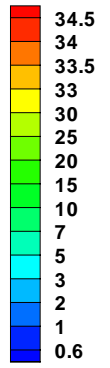
Radiation
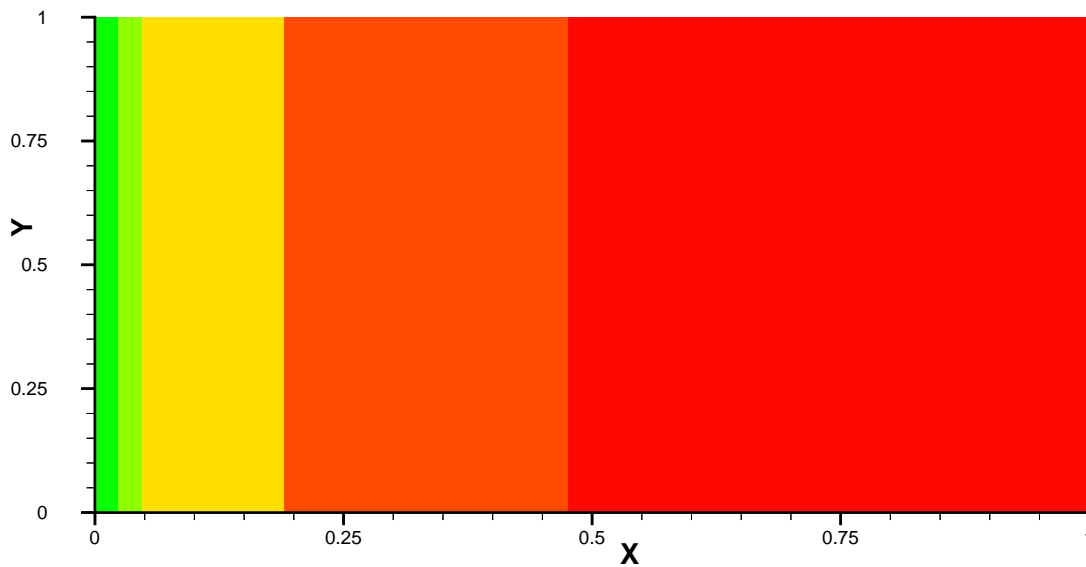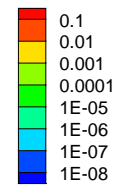
Fig 6b
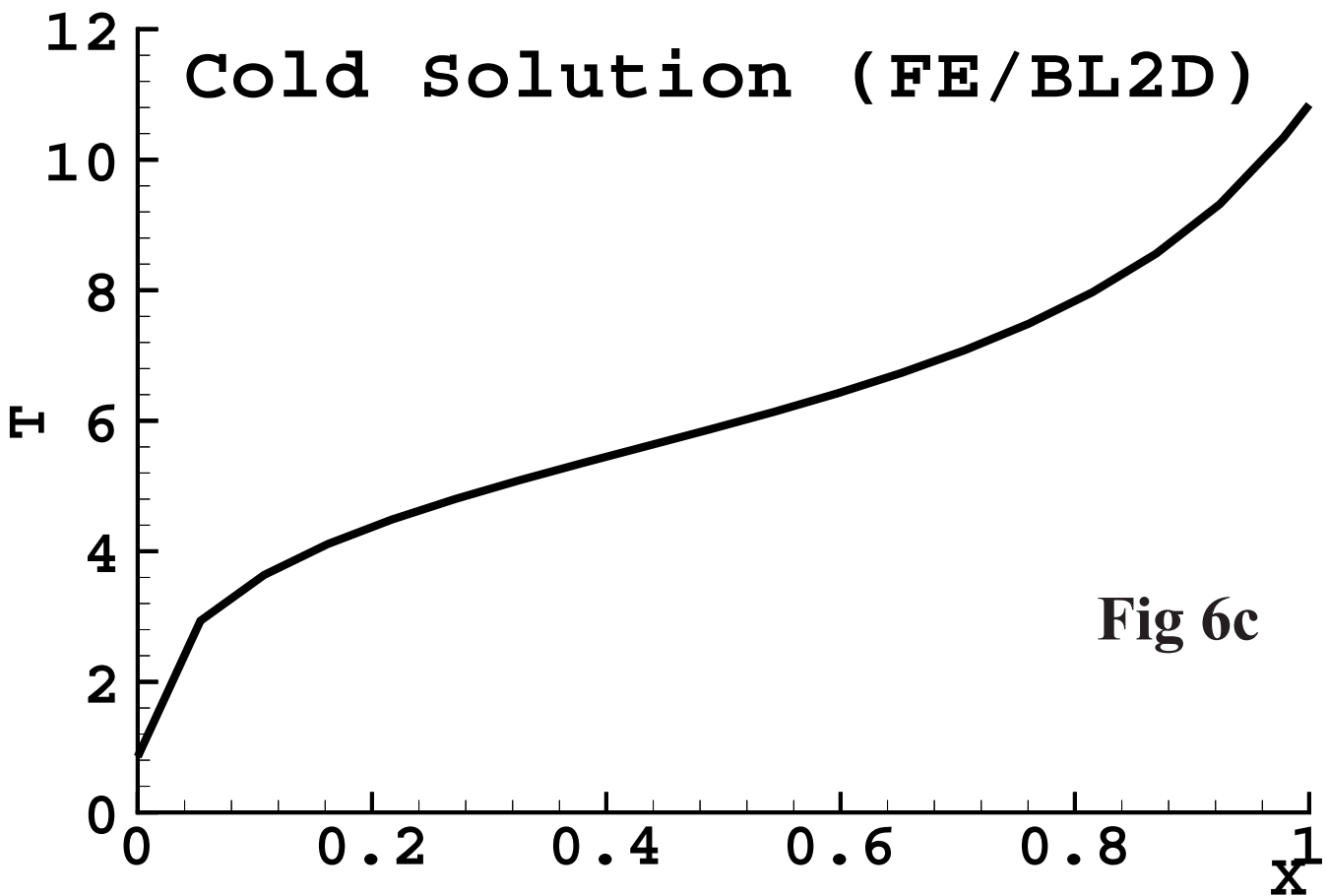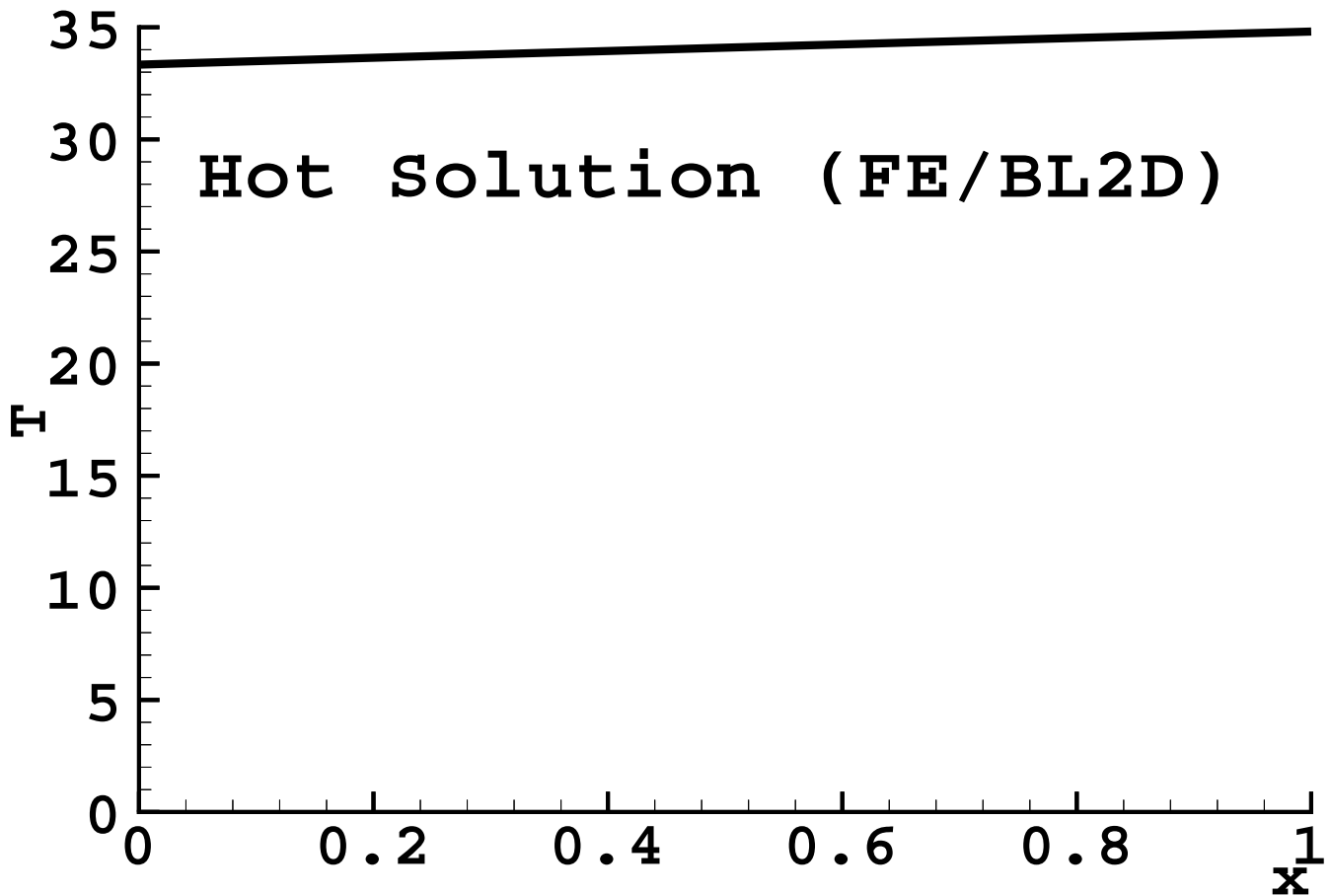
Fig 6c

FE/BL2D Solution
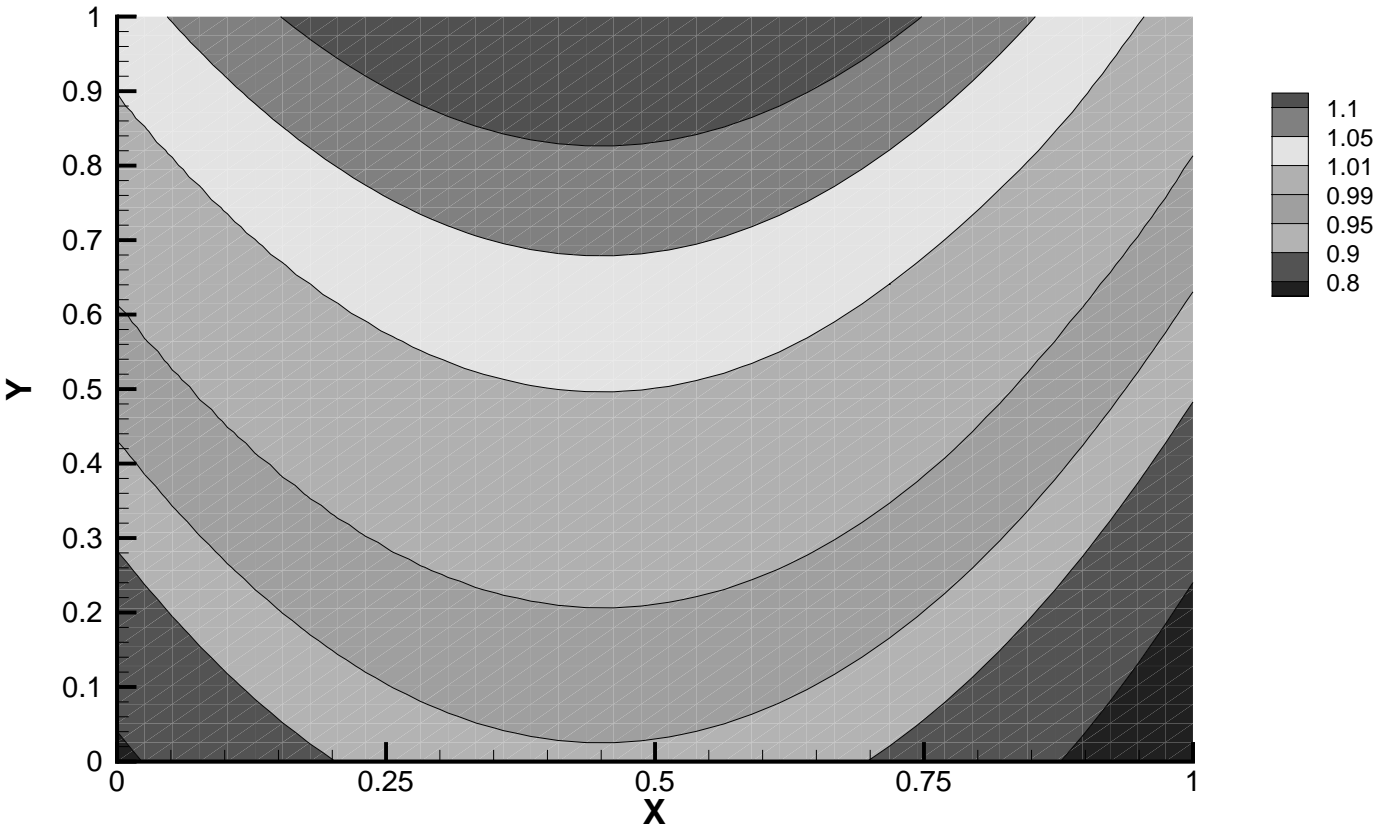
Fig 7a
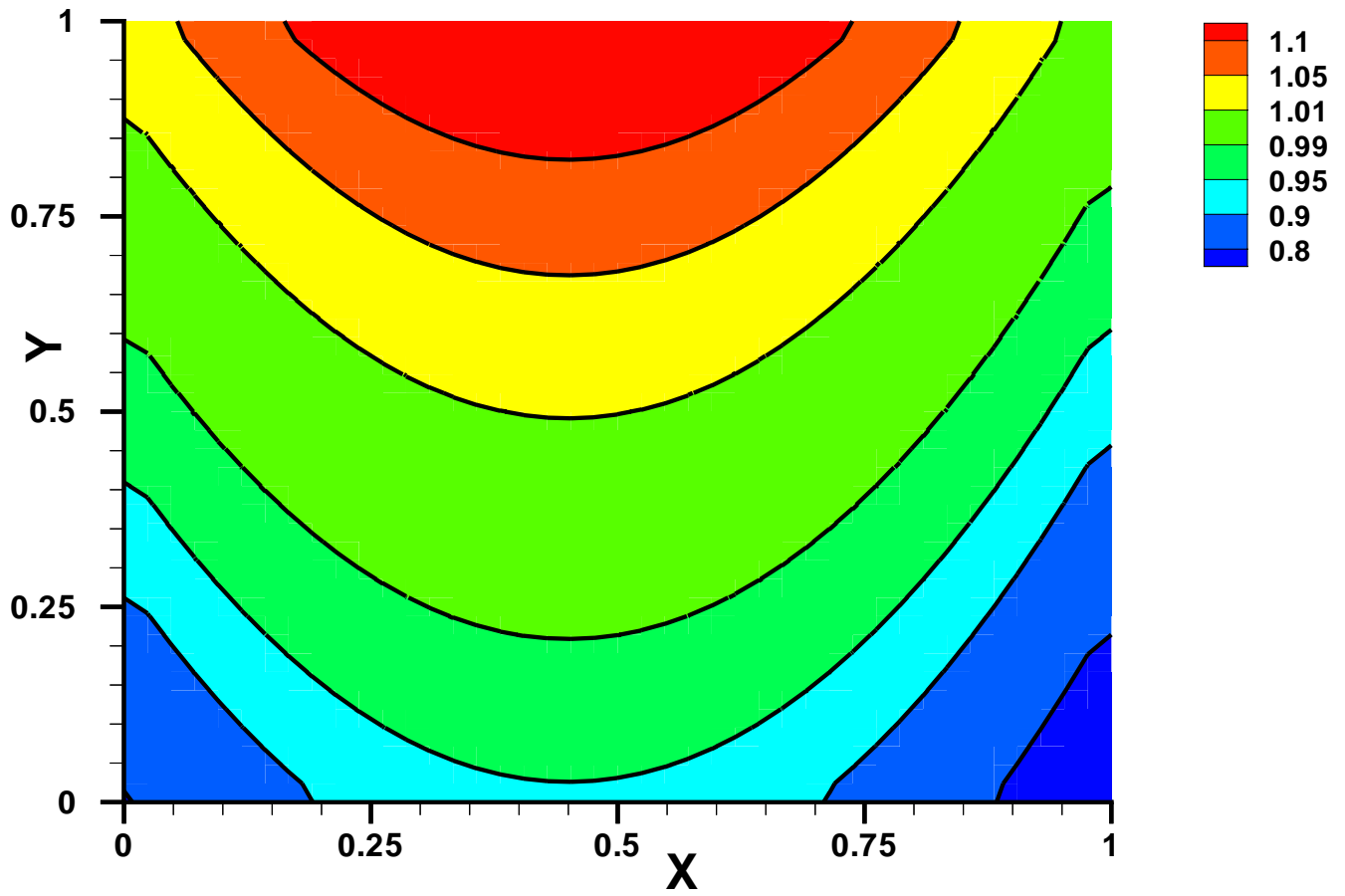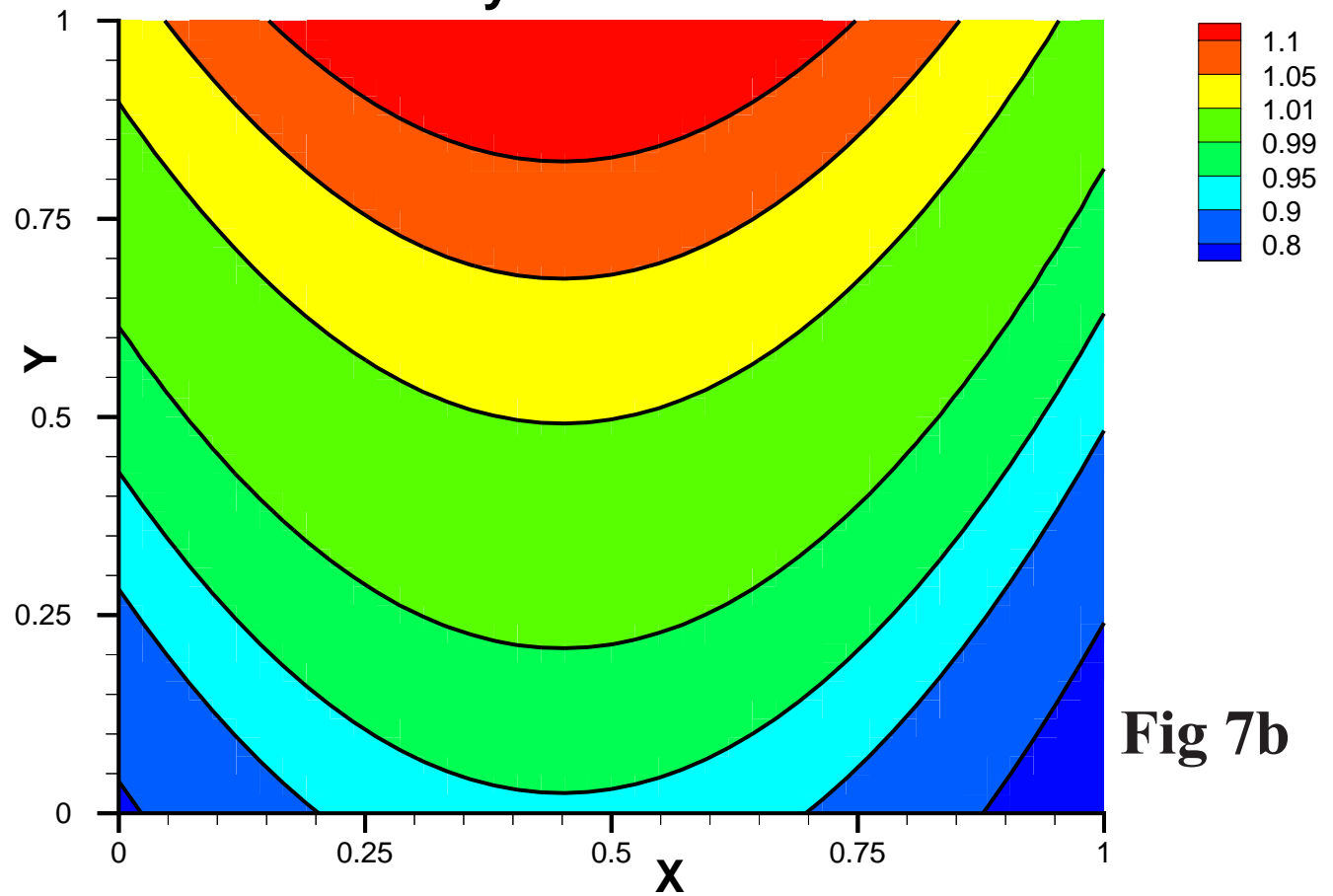
Fig 7b

Layer widening mesh
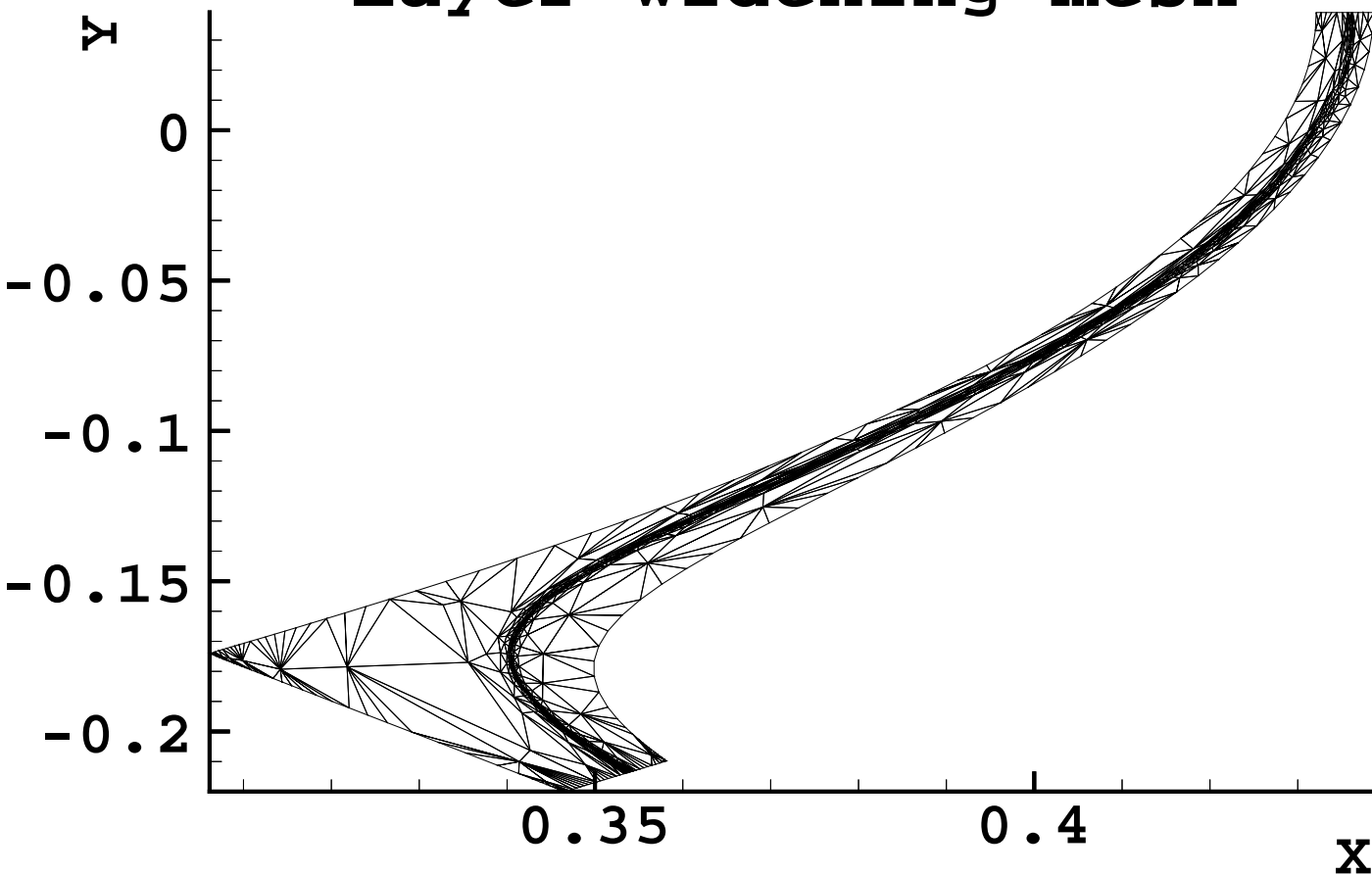
Layer widening solution

Fig 8
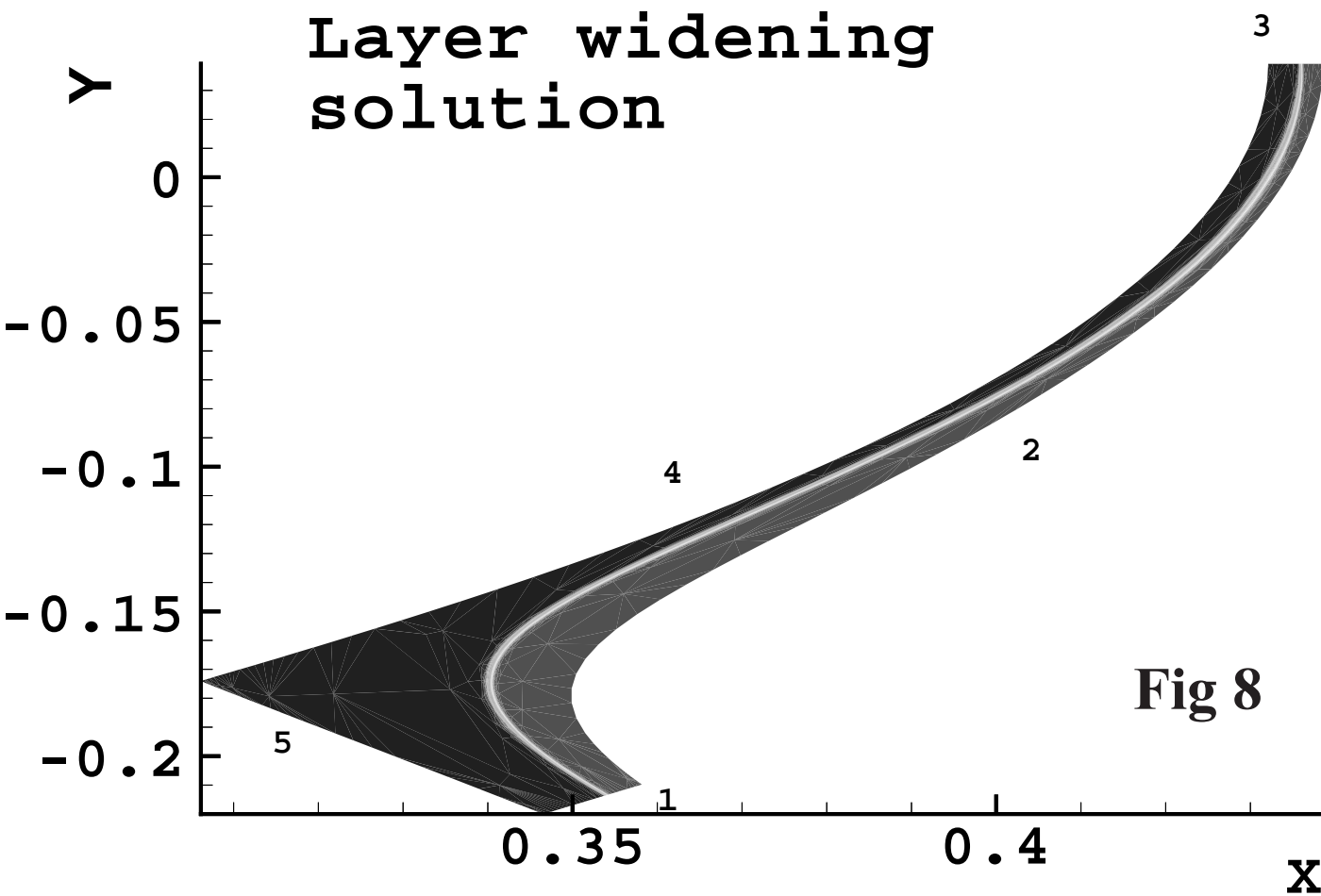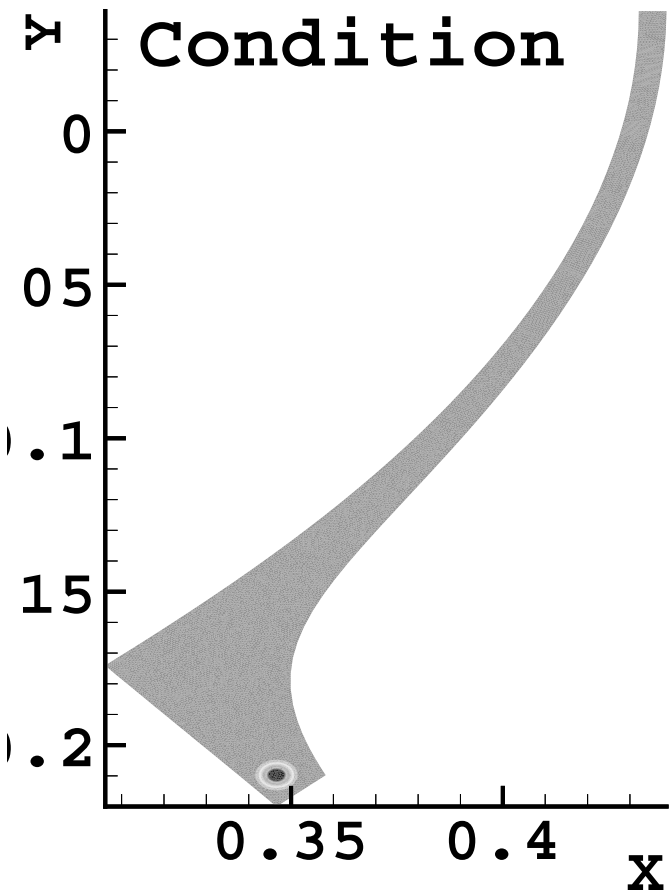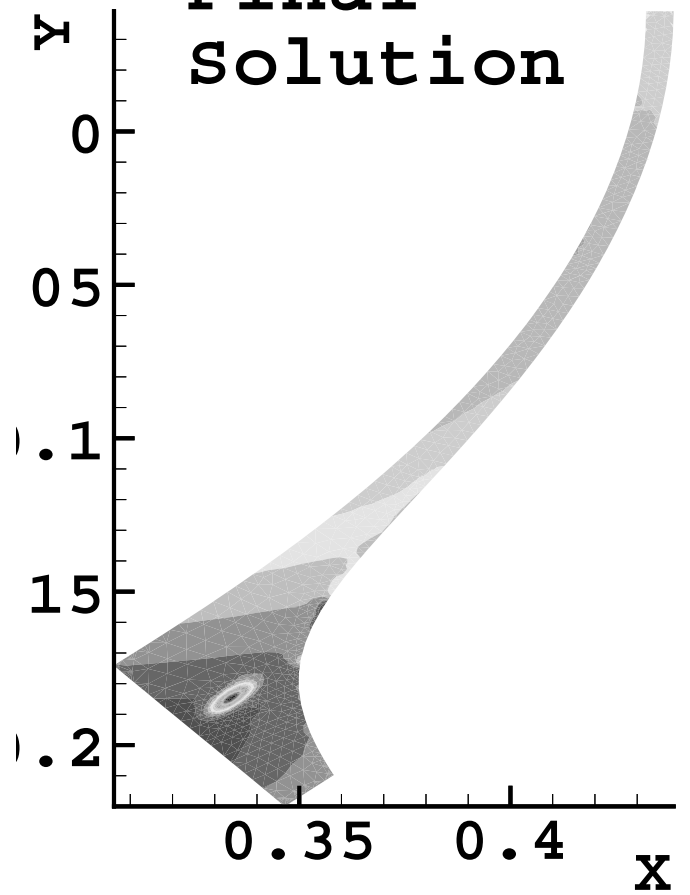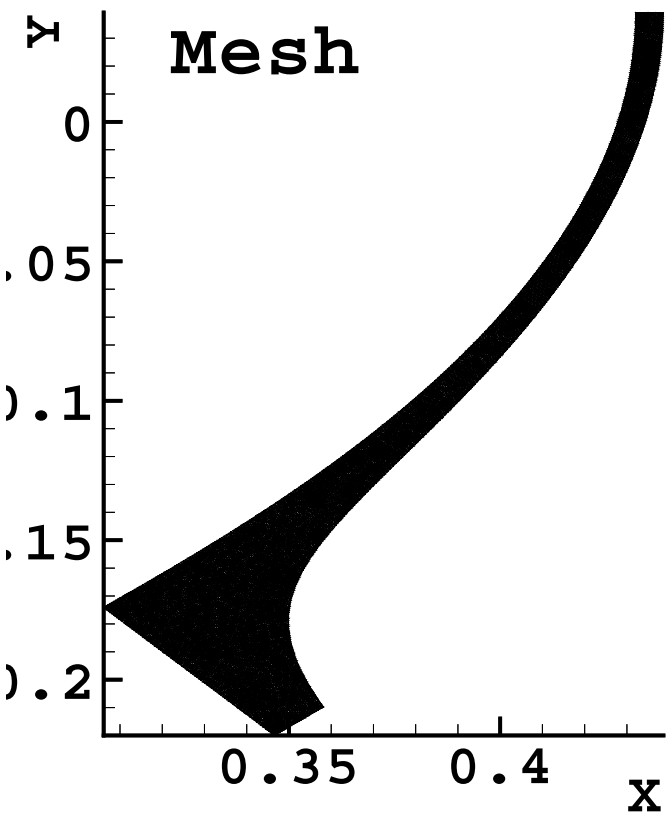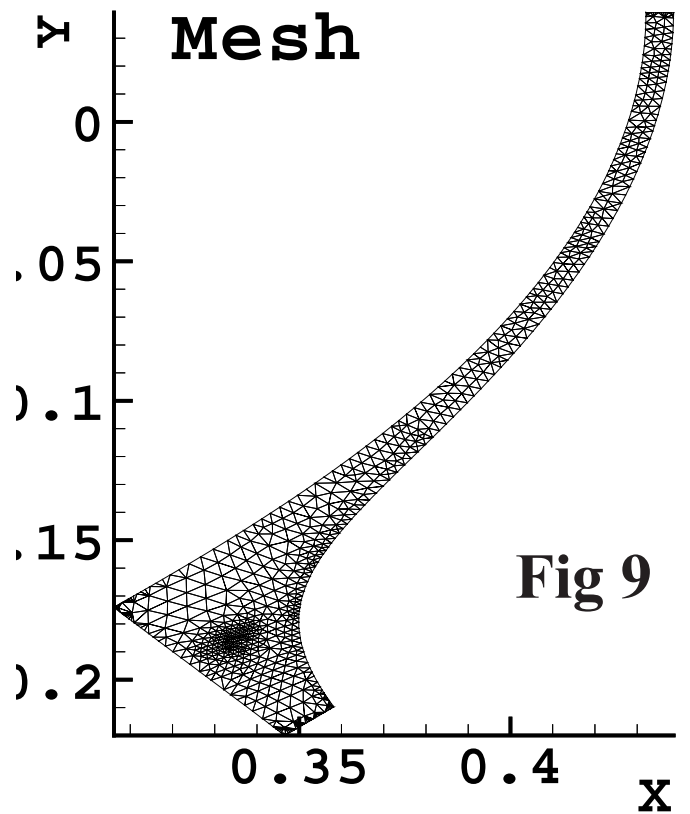
Fig 9

initial adaptive mesh for C-Mod (hybrid)

V=816
E=763

non-orth

orthogonal

O

D_{in}

D_{out}

Z

R

Fig 10

Fig 11

Fig 12

adaptive mesh, l=5

E=12.3K
level=5

O

D$_{in}$

D$_{out}$

Fig 13a

Fig 13 b