

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
PÓS-GRADUAÇÃO – FEC  
DEPARTAMENTO DE ESTRUTURAS**

## **RELATÓRIO DO CURSO**

**PROGRAMAÇÃO ORIENTADA PARA OBJETOS APLICADA A SIMULAÇÃO DE  
ELEMENTOS FINITOS – IC909 A**

**PROF. DR. PHILIPPE REMY BERNARD DEVLOO**

**ALUNO: LEANDRO SOARES VALDEZ**

**RA: 191307**

**CAMPINAS – SP  
Julho/2018**

## SUMÁRIO

<b>1</b>	<b>OBJETIVO.....</b>	<b>2</b>
<b>2</b>	<b>MÉTODO.....</b>	<b>2</b>
<b>3</b>	<b>ELEMENTOS GEOMÉTRICOS.....</b>	<b>3</b>
<b>4</b>	<b>REGRAS DE INTEGRAÇÃO NUMÉRICA .....</b>	<b>3</b>
4.1	INTEGRAÇÃO DE ELEMENTOS DE LINHA .....	4
4.2	INTEGRAÇÃO DE ELEMENTOS QUADRANGULARES.....	4
4.3	INTEGRAÇÃO DE ELEMENTOS TRIANGULARES.....	4
4.4	INTEGRAÇÃO DO ELEMENTO TETRAEDRO .....	4
<b>5</b>	<b>FUNÇÕES DE FORMA .....</b>	<b>5</b>
5.1	FUNÇÕES PARA ELEMENTOS DE LINHA .....	5
5.2	FUNÇÕES PARA ELEMENTOS TRIANGULARES .....	6
5.3	FUNÇÕES PARA ELEMENTOS QUADRANGULARES.....	6
5.4	FUNÇÕES PARA O ELEMENTO TETRAEDRO.....	7
<b>6</b>	<b>EXEMPLO DE APLICAÇÃO .....</b>	<b>8</b>
6.1	SOLUÇÃO ANALÍTICA .....	8
6.2	SOLUÇÃO PELO MÉTODO DOS ELEMENTOS FINITOS.....	10
6.3	ERRO.....	19
6.4	CONVERGÊNCIA .....	21
<b>7</b>	<b>CONCLUSÃO.....</b>	<b>26</b>
	<b>BIBLIOGRAFIA .....</b>	<b>27</b>

## 1 OBJETIVO

O objetivo desse relatório é apresentar os principais tópicos estudados no curso e demonstrar a aplicação do método de elementos finitos através da programação orientada a objetos (POO).

## 2 MÉTODO

Os códigos-fonte foram produzidos em linguagem C++ e estão disponibilizados no link: <https://github.com/lesoval/FEMCourse>. O repositório é compatível com o software “CMake” para criação de arquivos de desenvolvimento multiplataforma.

Para o desenvolvimento dos códigos foram disponibilizados cabeçalhos contendo os métodos e variáveis a serem implementados pelos alunos no decorrer do curso. O ambiente escolhido para o desenvolvimento foi o do Microsoft Visual Studio 2017.

A implementação dos métodos foi feita de forma livre e individual, porém sempre antecedida de orientações em sala de aula e amplamente baseada na biblioteca de elementos finitos NeoPZ. A atualização dos repositórios foi feita de maneira semanal seguindo o cronograma do curso.

### 3 ELEMENTOS GEOMÉTRICOS

O código contém quatro tipos de elementos geométricos disponíveis para utilização. Um elemento do tipo unidimensional (linha), dois elementos do tipo bidimensionais (triangular e quadrangular), e um elemento do tipo tridimensional (tetraedro).

### 4 REGRAS DE INTEGRAÇÃO NUMÉRICA

A regra de integração numérica escolhida como padrão foi a quadratura de Gauss-Legendre, que consiste na obtenção dos pontos notáveis de Gauss através das raízes dos polinômios de Legendre. Os pesos básicos são adquiridos pela integração dos polinômios de Legendre em um espaço paramétrico.

$$P_n(\varepsilon) = \frac{1}{2^n \cdot n!} \cdot \frac{d}{d\varepsilon^n} [(\varepsilon^2 - 1)^n]$$

$$w = \int_{-1}^1 P_n(\varepsilon) \cdot d\varepsilon$$

$P_n(\varepsilon)$  – Polinômio de Legendre no espaço paramétrico;

$\varepsilon$  – Variável paramétrica;

$n$  – Ordem do polinômio;

$w$  – Peso básico de Gauss.

O número de pontos de integração e pesos básicos varia conforme a ordem da função a ser integrada. Portanto, para as diferentes geometrias e funções de forma existirão diferentes pontos notáveis e pesos básicos para sua integração.

Assim, são usados diferentes algoritmos para obtenção desses valores de acordo com a geometria e ordem das funções de forma.

#### 4.1 INTEGRAÇÃO DE ELEMENTOS DE LINHA

Para a obtenção dos pontos e pesos básicos de integração de elementos de linha (1D), é utilizado o algoritmo “*gauleg*” encontrado em Press (2007, p. 184).

Esse algoritmo faz a obtenção dos pontos e pesos básicos para um polinômio de ordem qualquer no espaço paramétrico unidimensional de -1 a 1.

#### 4.2 INTEGRAÇÃO DE ELEMENTOS QUADRANGULARES

Para integração de elementos quadrangulares é utilizado o mesmo algoritmo da integração unidimensional, porém, por se tratar de um elemento bidimensional são encontradas coordenadas em duas dimensões e permutadas para obter os pontos em coordenadas paramétricas bidimensionais. Os pesos de cada ponto são o produto dos pesos nas coordenadas unidimensionais.

Assim como na integração linear a integração dos elementos quadrangulares faz a obtenção dos pontos e pesos básicos para um polinômio de ordem qualquer no espaço paramétrico bidimensional com coordenadas de -1 a 1.

#### 4.3 INTEGRAÇÃO DE ELEMENTOS TRIANGULARES

O algoritmo utilizado para integração dos elementos triangulares foi retirado do código NeoPZ e possui limitação de 91 pontos de integração e polinômios de ordem máxima igual a 21. O algoritmo calcula os pontos e pesos básicos para um espaço paramétrico com coordenadas de 0 a 1. Esse pode ser encontrado no arquivo “*tpzintrule.cpp*”.

#### 4.4 INTEGRAÇÃO DO ELEMENTO TETRAEDRO

Assim como na integração de elementos triangulares o algoritmo que obtém os pontos e pesos básicos de integração foi retirado do código NeoPZ. Sendo que esse possui limitação de 236 pontos de integração e ordem polinomial máxima igual a 14. O algoritmo calcula os pontos e pesos básicos para um espaço paramétrico com coordenadas de 0 a 1. Esse pode ser encontrado em “*tpzintrule3d.cpp*”.

## 5 FUNÇÕES DE FORMA

O número de funções de forma de cada elemento é definido conforme o número de pontos definidos dentro da geometria do mesmo. Os elementos utilizados nesse código são do tipo lagrangiano, ou seja, suas funções de forma são baseadas em polinômios interpoladores de Lagrange.

A ordem dos polinômios de Lagrange é definida de acordo com o número de pontos a serem interpolados dentro do elemento.

### 5.1 FUNÇÕES PARA ELEMENTOS DE LINHA

Abaixo segue o algoritmo para obtenção do polinômio de Lagrange em coordenadas paramétricas unidimensionais.

$$N_i(\varepsilon) = \prod_{j \neq i}^n \frac{\varepsilon - \varepsilon_j}{\varepsilon_i - \varepsilon_j}$$

Esse algoritmo permite a obtenção do polinômio de Lagrange de qualquer ordem em “n” pontos de interpolação.

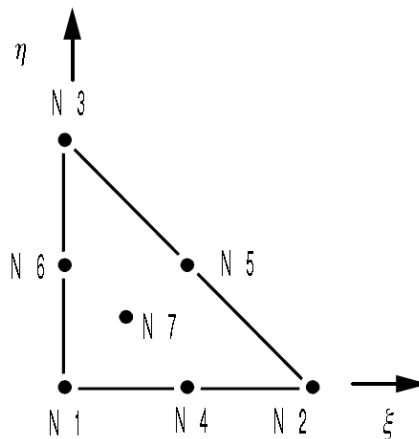
E sua derivada é obtida pelo algoritmo abaixo.

$$N_i'(\varepsilon) = \sum_{j \neq i}^n \frac{1}{\varepsilon_i - \varepsilon_j} \cdot \prod_{k \neq (i,j)}^n \frac{\varepsilon - \varepsilon_k}{\varepsilon_i - \varepsilon_k}$$

## 5.2 FUNÇÕES PARA ELEMENTOS TRIANGULARES

Para os elementos triangulares foram implementadas funções de forma bilineares com três pontos e funções quadráticas com seis e sete pontos. Todas as funções de forma e suas derivadas foram retiradas de Delmas (2017, p.6).

A figura 1 demonstra a localização e numeração dos pontos de interpolação.

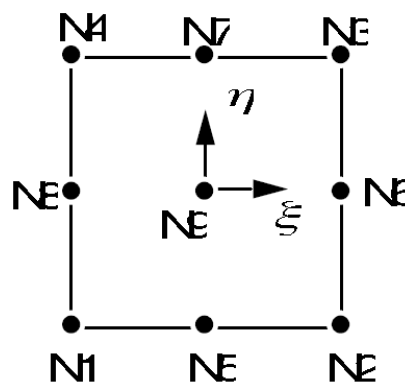


**Figura 1 – Pontos de interpolação para o elemento triangular (DELMAS, 2017)**

## 5.3 FUNÇÕES PARA ELEMENTOS QUADRANGULARES

Nos elementos quadrangulares foram implementadas funções de forma bilineares com quatro pontos e funções quadráticas com oito e nove pontos. Assim como para os elementos triangulares todas as funções de forma e suas derivadas foram retiradas de Delmas (2017, p.10).

A figura 2 demonstra a localização e numeração dos pontos de interpolação.



**Figura 2 - Pontos de interpolação para o elemento quadrangular (DELMAS, 2017)**

#### 5.4 FUNÇÕES PARA O ELEMENTO TETRAEDRO

Para o tetraedro foram implementadas funções de forma com quatro e dez pontos de interpolação. Todas as funções de forma foram retiradas de Enes (2009) e suas derivadas foram obtidas de forma manual.

A figura 3 demonstra a localização e numeração dos pontos de interpolação.

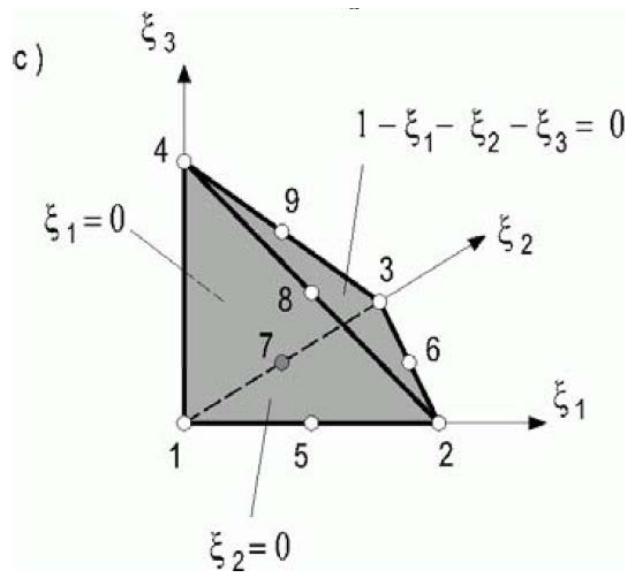


Figura 3 - Pontos de interpolação para o elemento tetraedro (ENES, 2009)



## 6 EXEMPLO DE APLICAÇÃO

Como exemplo de aplicação foi escolhido um problema de elasticidade em estado plano de tensões. O problema trata de um cilindro com aplicação de tensões normais uniformemente distribuídas na linha de perímetro interno e externo, ou seja, um problema de condições de contorno de Neumann.

Tanto a solução analítica quanto a solução por MEF foram resolvidas para uma espessura unitária.

### 6.1 SOLUÇÃO ANALÍTICA

A solução analítica pode ser encontrada em Obert e Duval (1967). Originalmente a solução desse problema é dada em coordenadas polares, no entanto, nesse relatório será apresentada em coordenadas cartesianas.

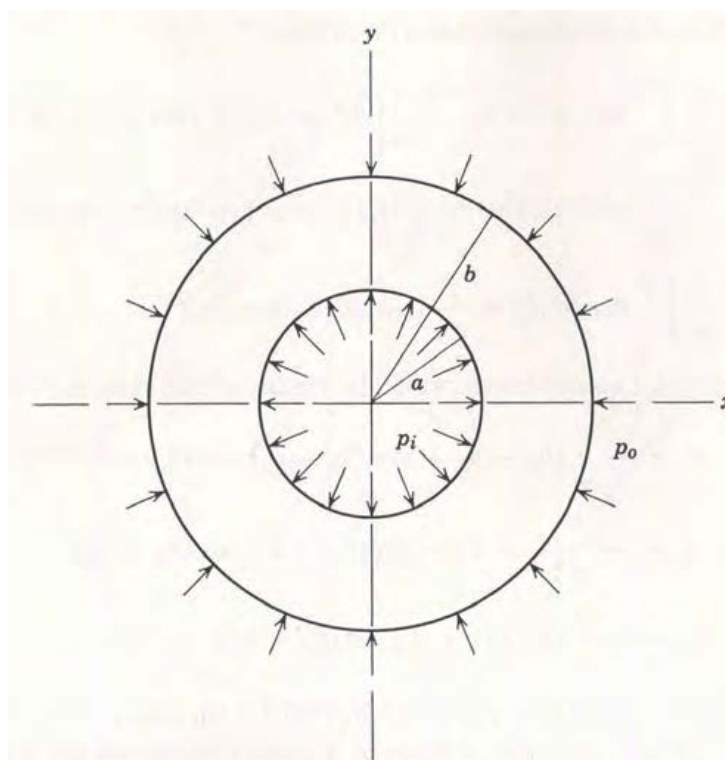


Figura 4 - Esquema de aplicação das tensões no cilindro (OBERT e DUVALL, 1967)

Seguem abaixo as soluções analíticas para os deslocamentos e deformações em coordenadas cartesianas.

$$u_x = \frac{x}{E \cdot (b^2 - a^2)} \cdot \left[ (a^2 \cdot p_i - b^2 \cdot p_0) \cdot (1 - \nu) - \frac{a^2 \cdot b^2 \cdot (p_0 - p_i)}{r^2} \cdot (1 + \nu) \right]$$

$$u_y = \frac{y}{E \cdot (b^2 - a^2)} \cdot \left[ (a^2 \cdot p_i - b^2 \cdot p_0) \cdot (1 - \nu) - \frac{a^2 \cdot b^2 \cdot (p_0 - p_i)}{r^2} \cdot (1 + \nu) \right]$$

$$e_x = \frac{1}{E \cdot (b^2 - a^2)} \cdot \left[ (a^2 \cdot p_i - b^2 \cdot p_0) \cdot (1 - \nu) + \frac{a^2 \cdot b^2 \cdot (p_0 - p_i) \cdot (1 + \nu)}{r^4} \cdot (x^2 - y^2) \right]$$

$$e_y = \frac{1}{E \cdot (b^2 - a^2)} \cdot \left[ (a^2 \cdot p_i - b^2 \cdot p_0) \cdot (1 - \nu) + \frac{a^2 \cdot b^2 \cdot (p_0 - p_i) \cdot (1 + \nu)}{r^4} \cdot (y^2 - x^2) \right]$$

$$e_{xy} = \frac{4 \cdot x \cdot y}{E \cdot (b^2 - a^2)} \cdot \left[ \frac{a^2 \cdot b^2 \cdot (p_0 - p_i) \cdot (1 + \nu)}{r^4} \right]$$

$E$  – Módulo de Young;

$\nu$  – Coeficiente de Poisson;

$a$  – Raio interno;

$b$  – Raio externo;

$r = \sqrt{x^2 + y^2}$ ;

$p_i$  – Tensão no perímetro interno;

$p_0$  – Tensão no perímetro externo.

## 6.2 SOLUÇÃO PELO MÉTODO DOS ELEMENTOS FINITOS

Para solução do problema foram utilizados os seguintes dados:

$$E = 210.000 \text{ mPa};$$

$$\nu = 0,30;$$

$$a = 30 \text{ mm};$$

$$b = 40 \text{ mm};$$

$$p_i = 100 \text{ mPa};$$

$$p_0 = 75 \text{ mPa}.$$

Na solução por MEF foram utilizadas seis composições de malhas distintas, sendo três contendo elementos triangulares e três contendo elementos quadrangulares.

Como o código utiliza a matriz de rigidez global em sua forma completa a resolução do sistema de equações se torna prejudicada para malhas muito refinadas devido ao alto consumo de hardware. Por essa limitação foram testadas apenas quatro das seis malhas contendo elementos quadráticos.

O quadro a seguir contém os tipos e números de elementos contidos em cada malha.

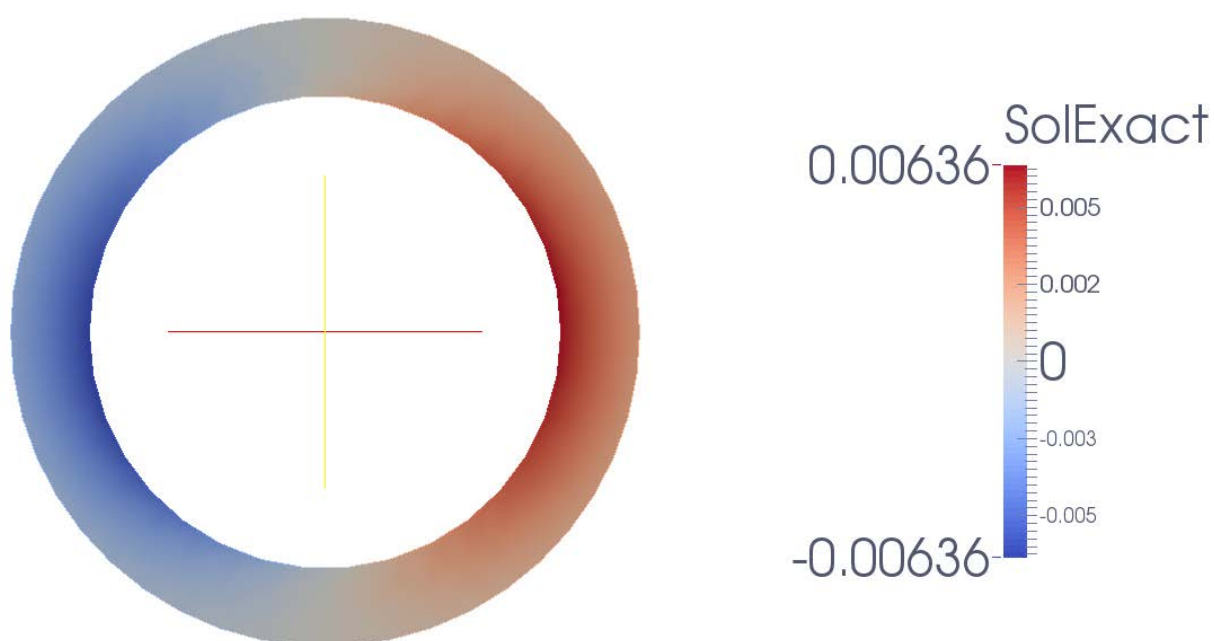
**Quadro 1 - Tipo e número de elementos de cada malha**

Tipo	Nº de elementos (triangulares)			Nº de elementos (quadrangulares)		
	46	176	720	22	88	360
Bi lineares	X	X	X	X	X	X
Quadráticos	X	X		X	X	

As malhas foram geradas no software “*Gmsh*” com a utilização do algoritmo “*Delunay*” e algoritmo de recombinação “*Blossom*”. Essas foram geradas de maneira que os elementos mantivessem uma homogeneidade quanto às suas dimensões, tornando mais assimilável o cálculo da convergência da solução.

Com intuito de exemplificação serão mostrados apenas os resultados das malhas que obtiveram os melhores resultados de aproximação, são elas: malha com 176 elementos triangulares quadráticos e malha com 360 elementos quadrangulares bi lineares.

Abaixo seguem os resultados das soluções analítica e numérica (MEF) para os deslocamentos em coordenadas cartesianas.



**Figura 5 - Solução exata para  $u_x$  [mm] (Paraview)**

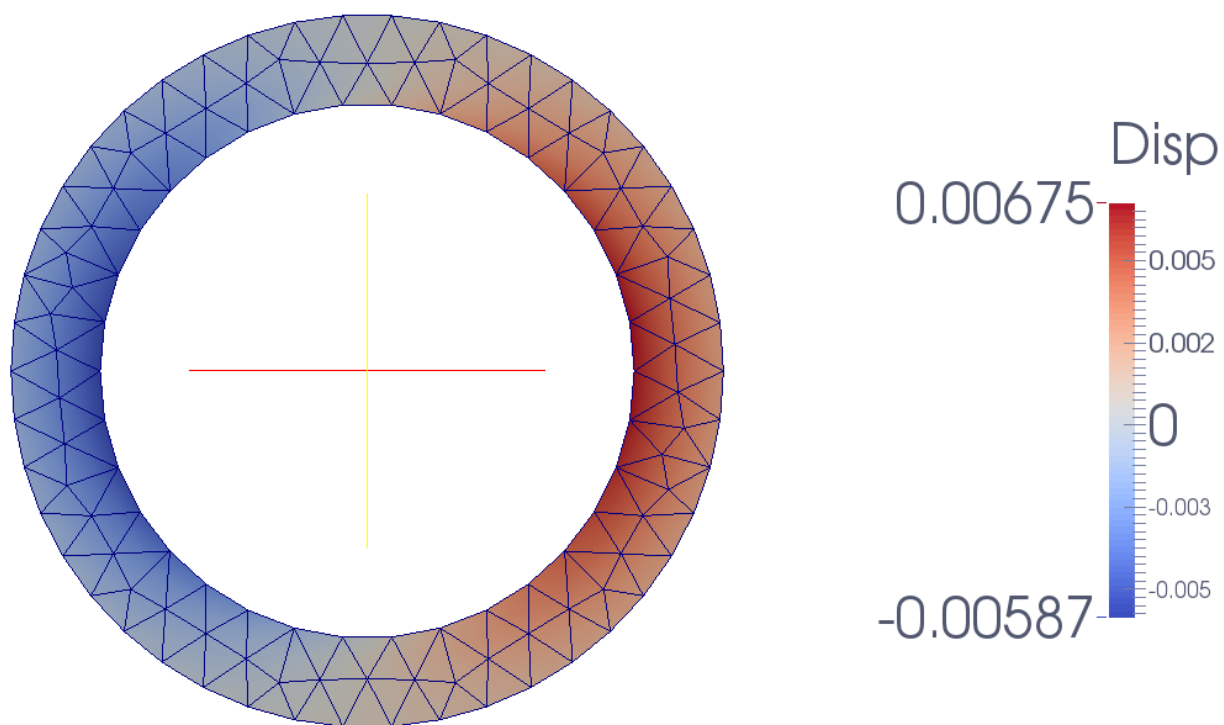


Figura 6 - Solução de  $u_x$  [mm] para a malha de 176 elementos triangulares (Paraview)

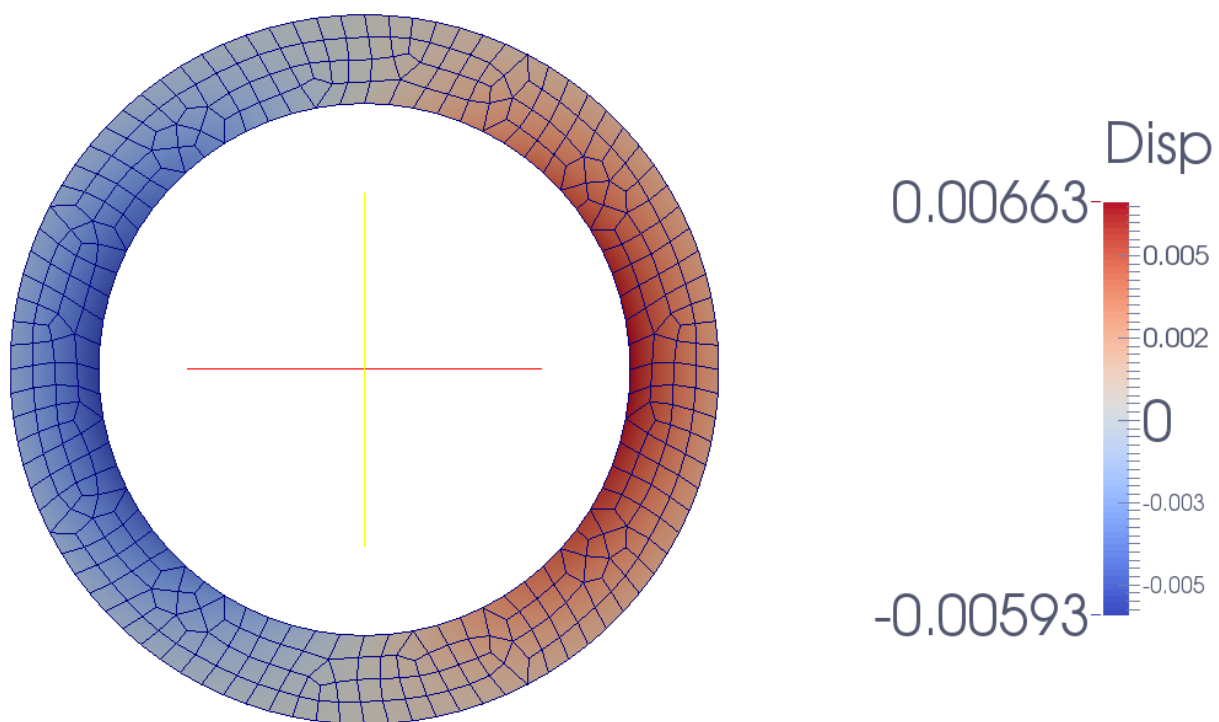
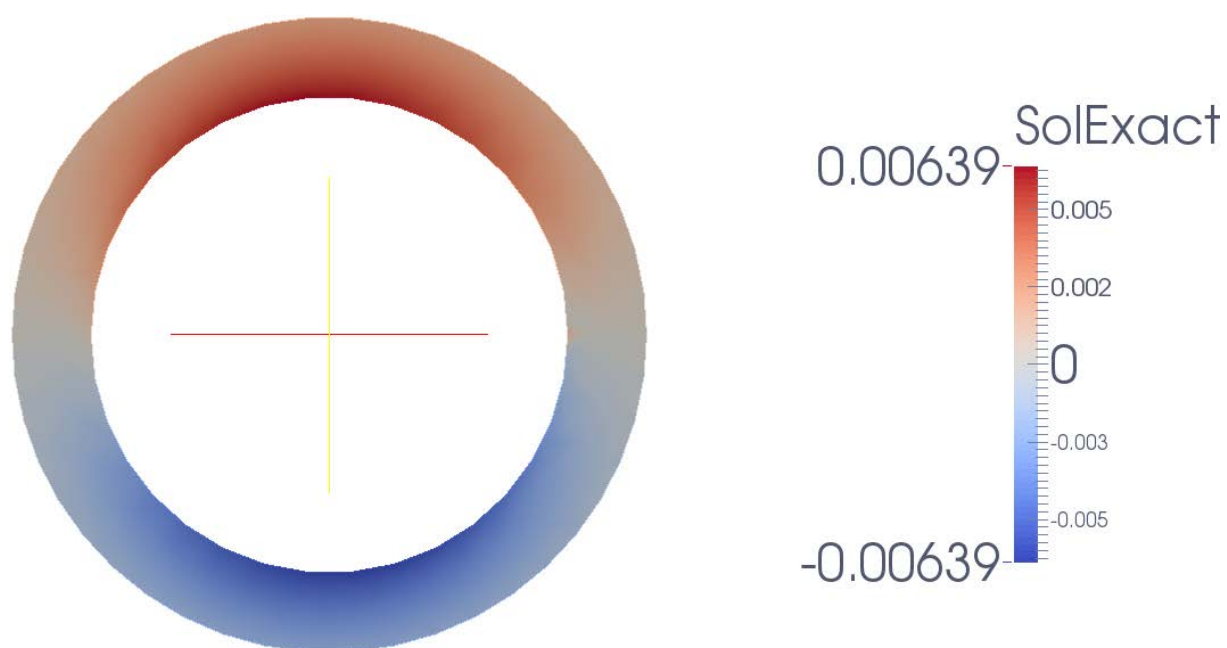
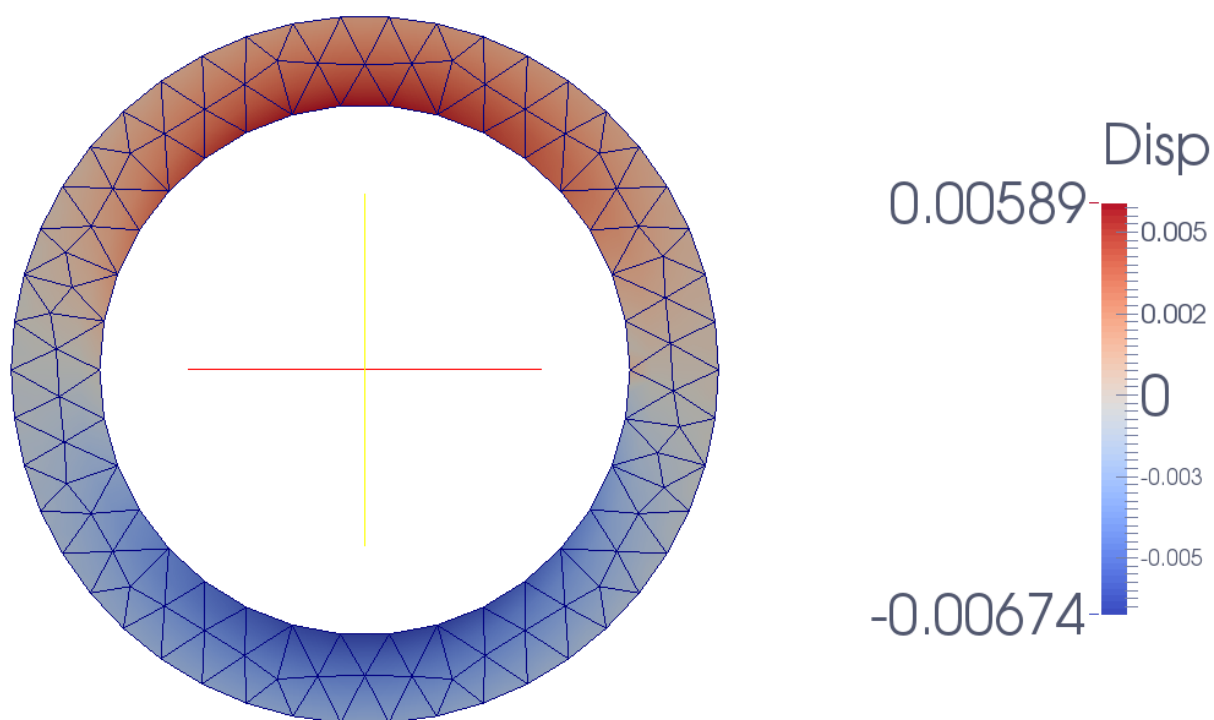


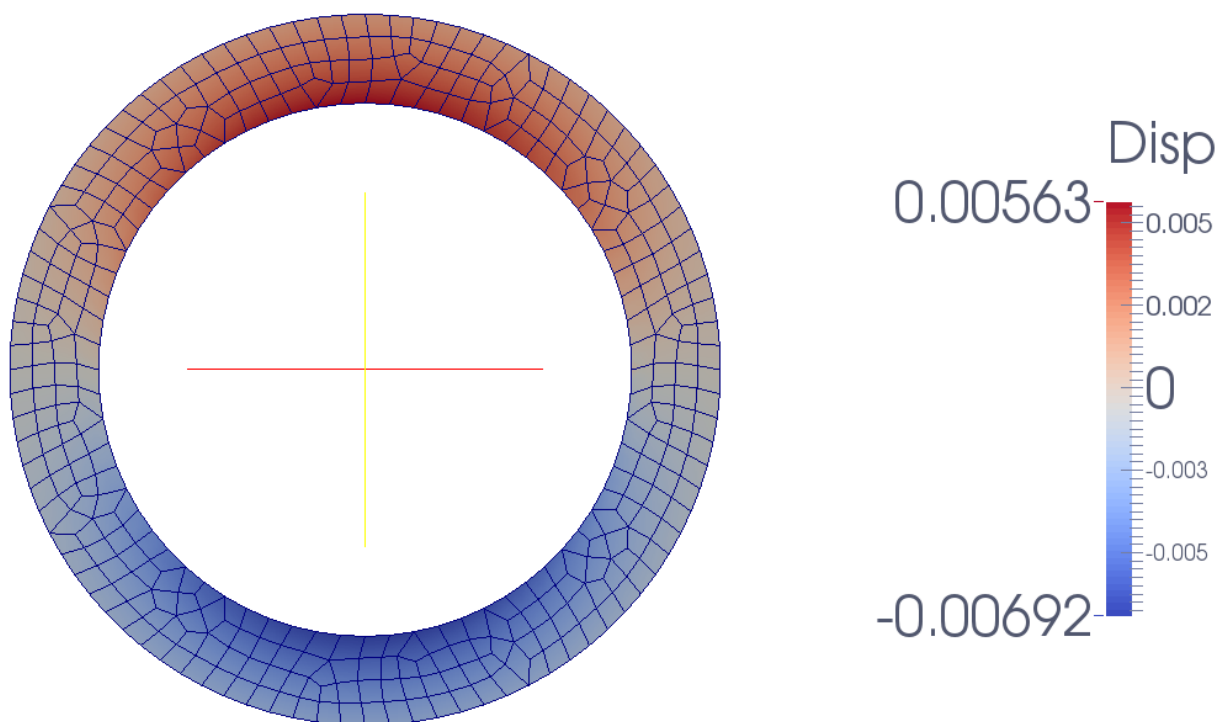
Figura 7 - Solução de  $u_x$  [mm] para a malha de 360 elementos quadrangulares (Paraview)



**Figura 8 - Solução exata para  $u_y$  [mm] (Paraview)**

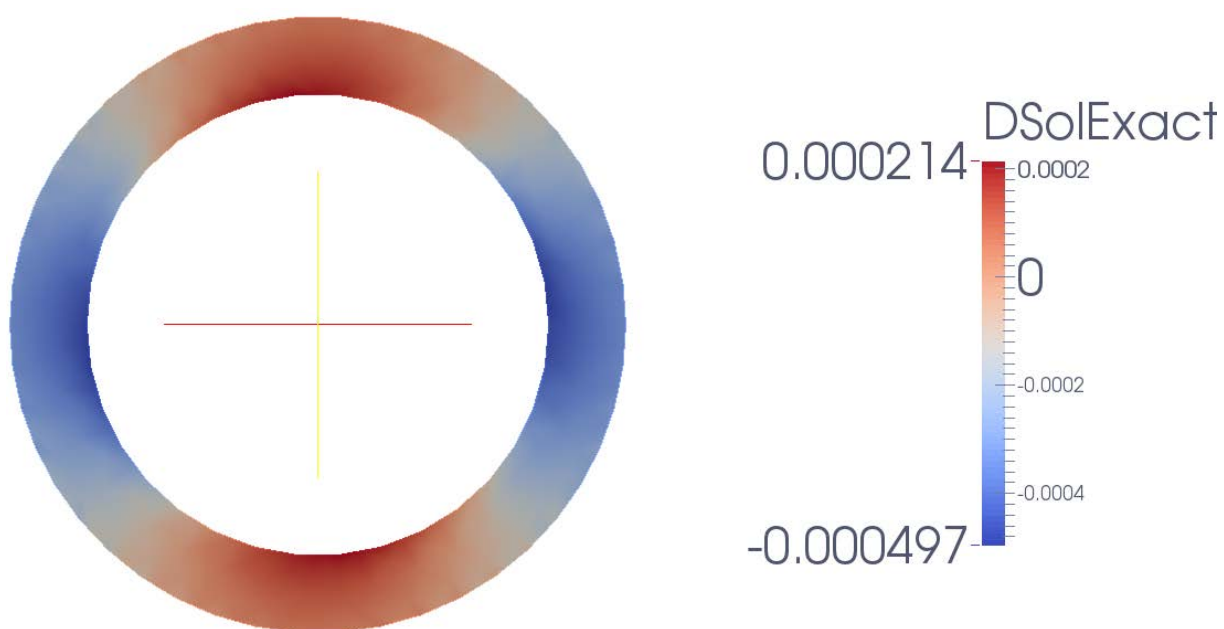


**Figura 9 - Solução de  $u_y$  [mm] para a malha de 176 elementos triangulares (Paraview)**



**Figura 10 - Solução de  $u_y$  [mm] para a malha de 360 elementos quadrangulares (Paraview)**

Abaixo seguem os resultados das soluções analítica e numérica (MEF) para deformações normais e tangenciais.



**Figura 11 - Solução exata para  $\epsilon_{xx}$  (Paraview)**

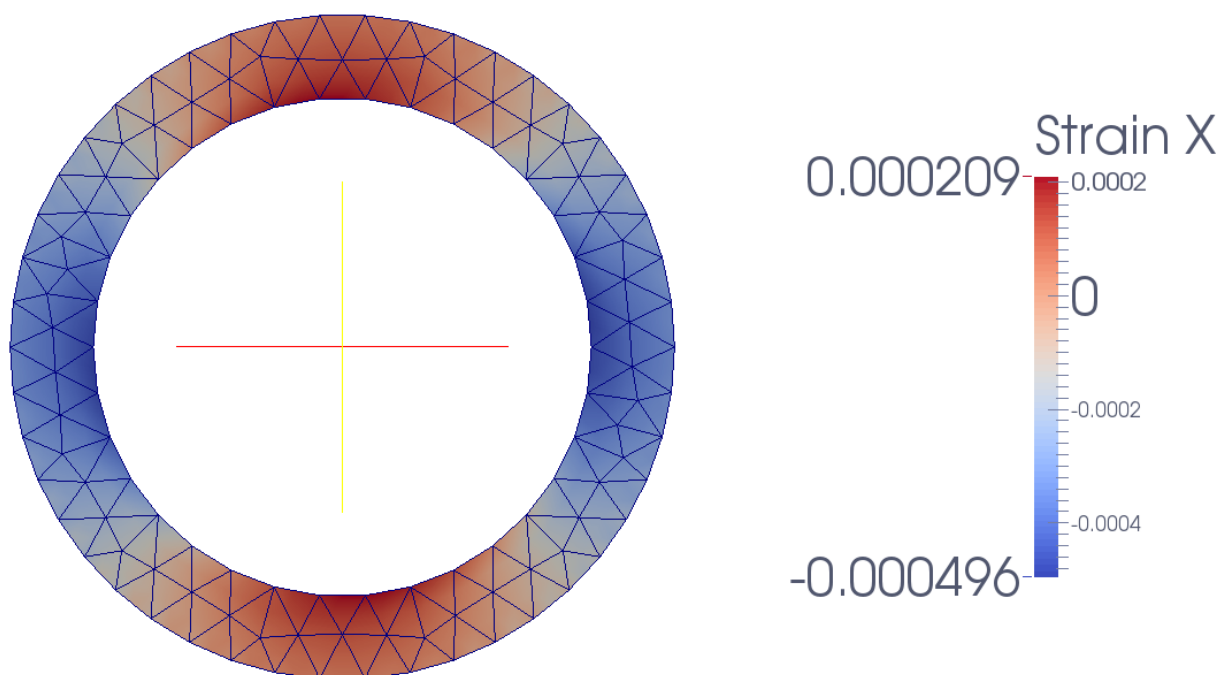


Figura 12 - Solução de  $\epsilon_{xx}$  para a malha de 176 elementos triangulares (Paraview)

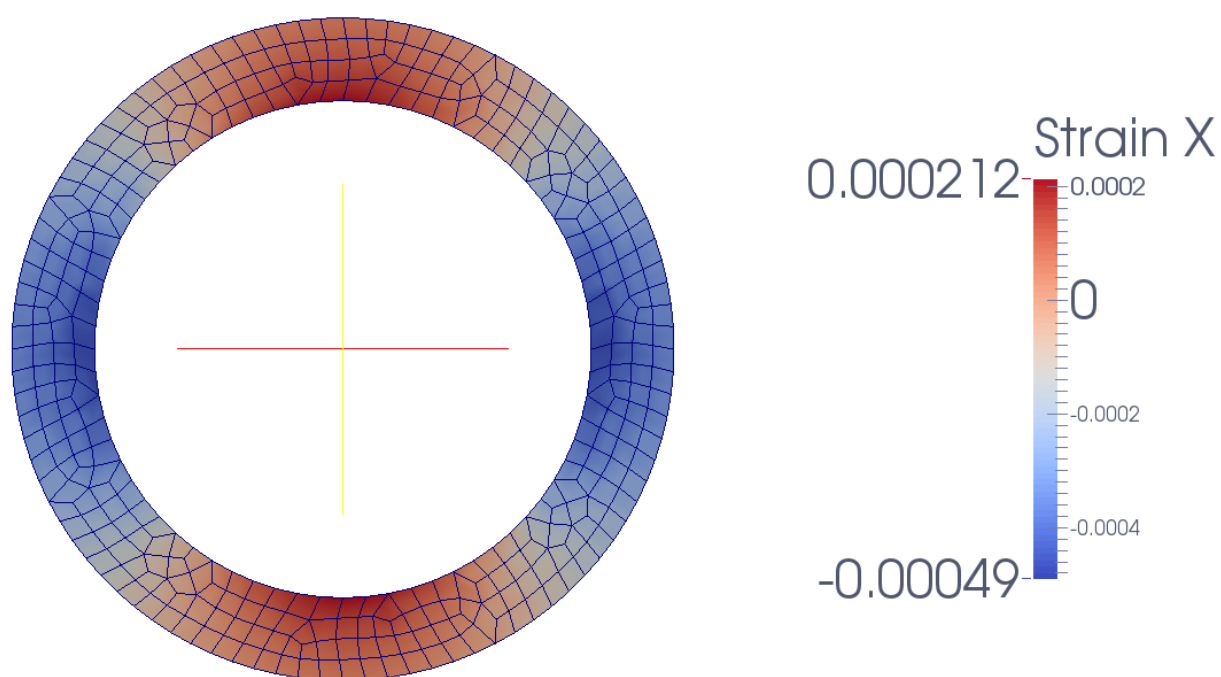


Figura 13 - Solução de  $\epsilon_{xx}$  para a malha de 360 elementos quadrangulares (Paraview)



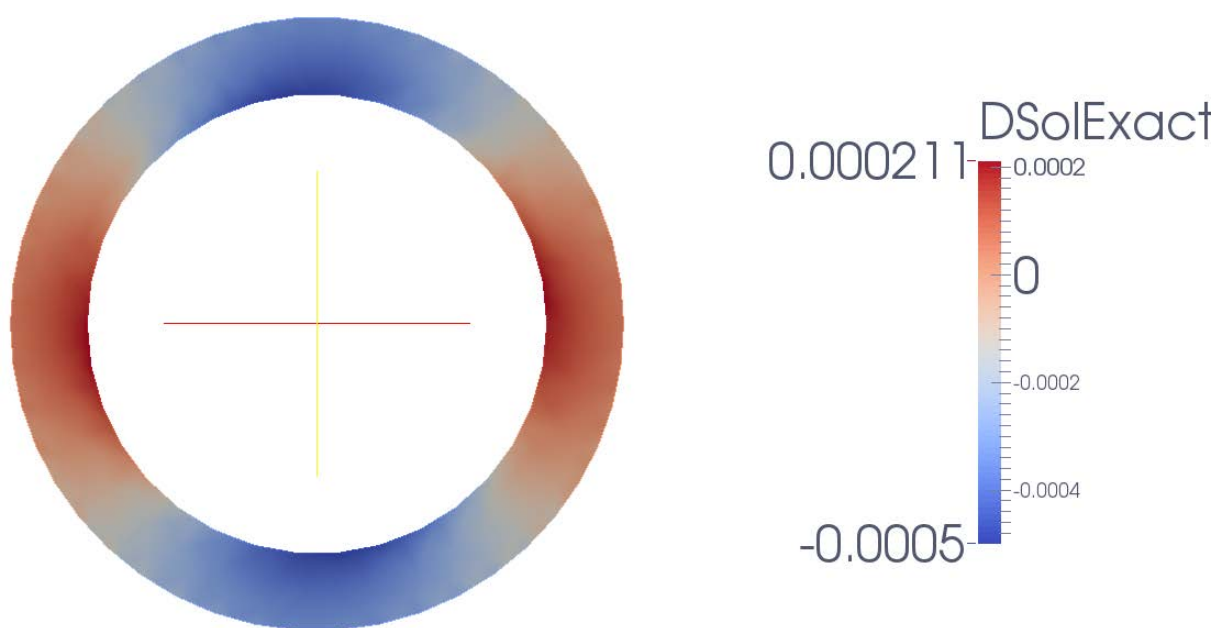


Figura 14 - Solução exata para  $e_{yy}$  (Paraview)

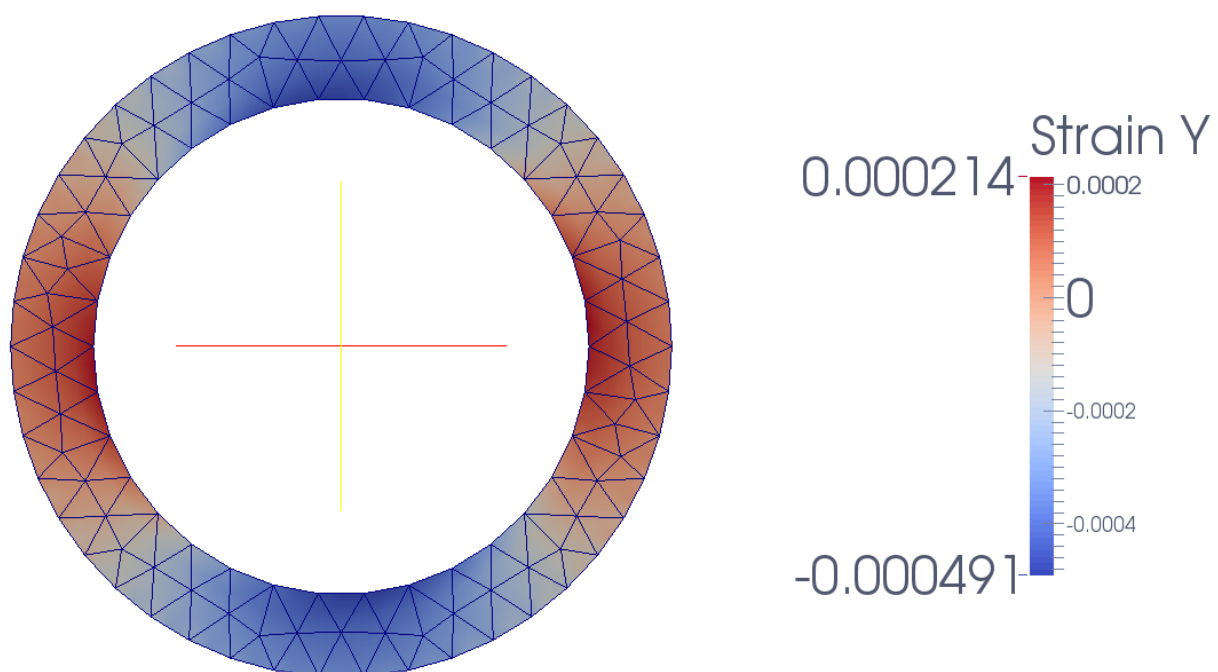


Figura 15 - Solução de  $e_{yy}$  para a malha de 176 elementos triangulares (Paraview)

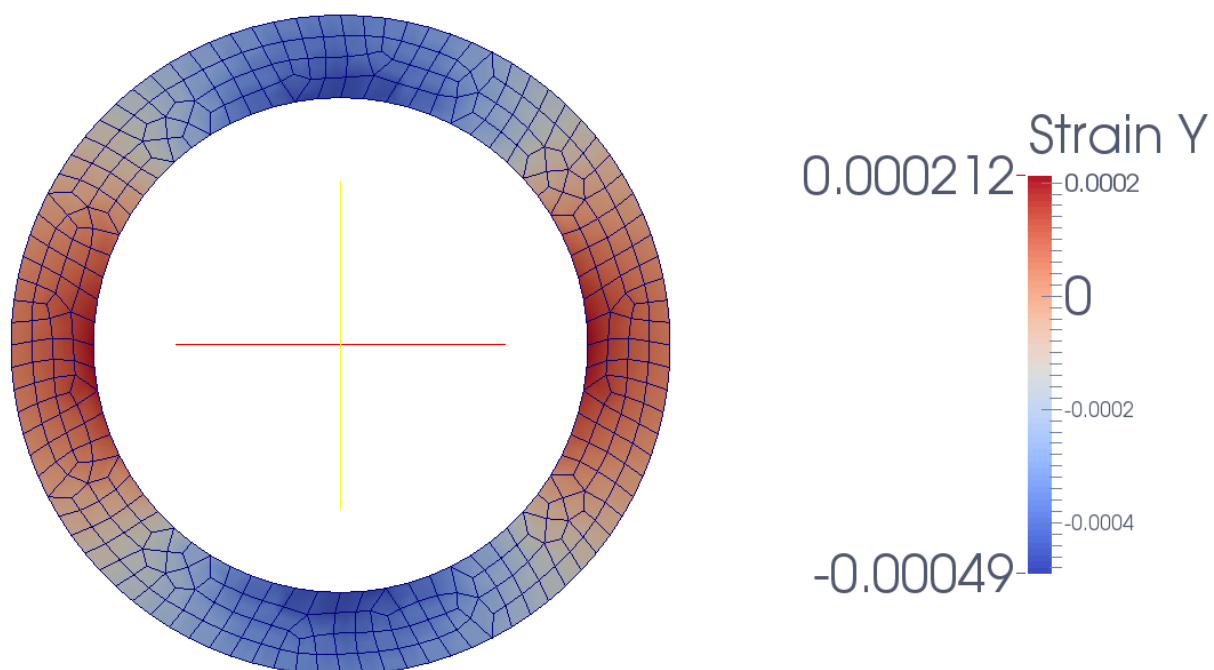


Figura 16 - Solução de  $\epsilon_{yy}$  para a malha de 360 elementos quadrangulares (Paraview)

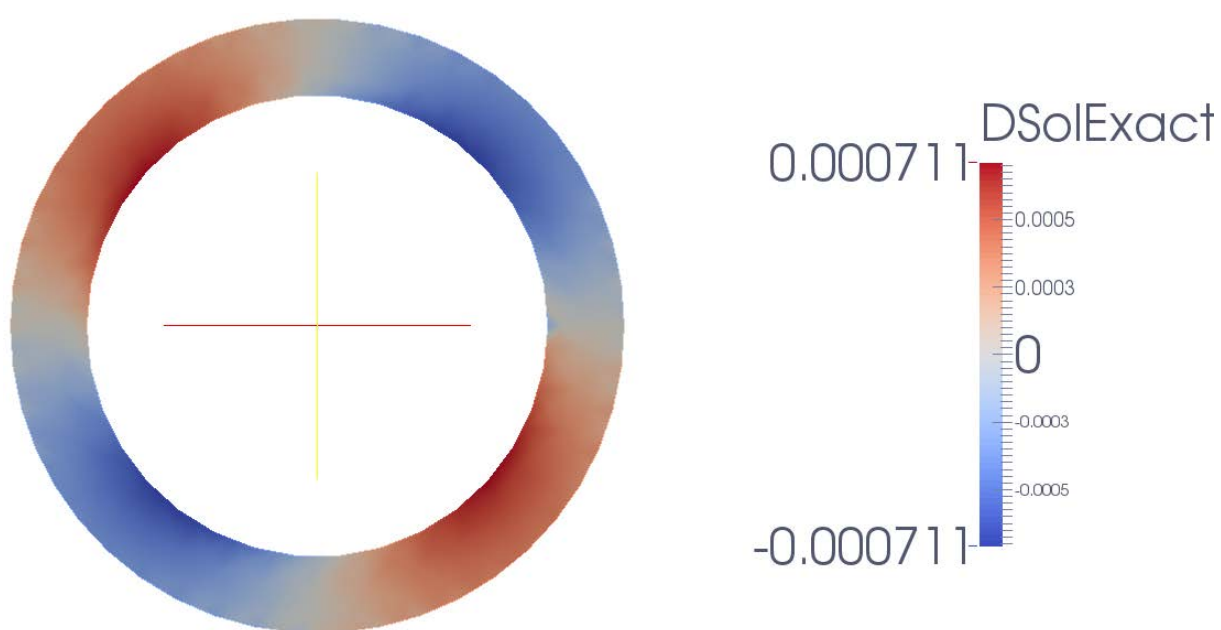


Figura 17 - Solução exata para  $\epsilon_{xy}$  (Paraview)

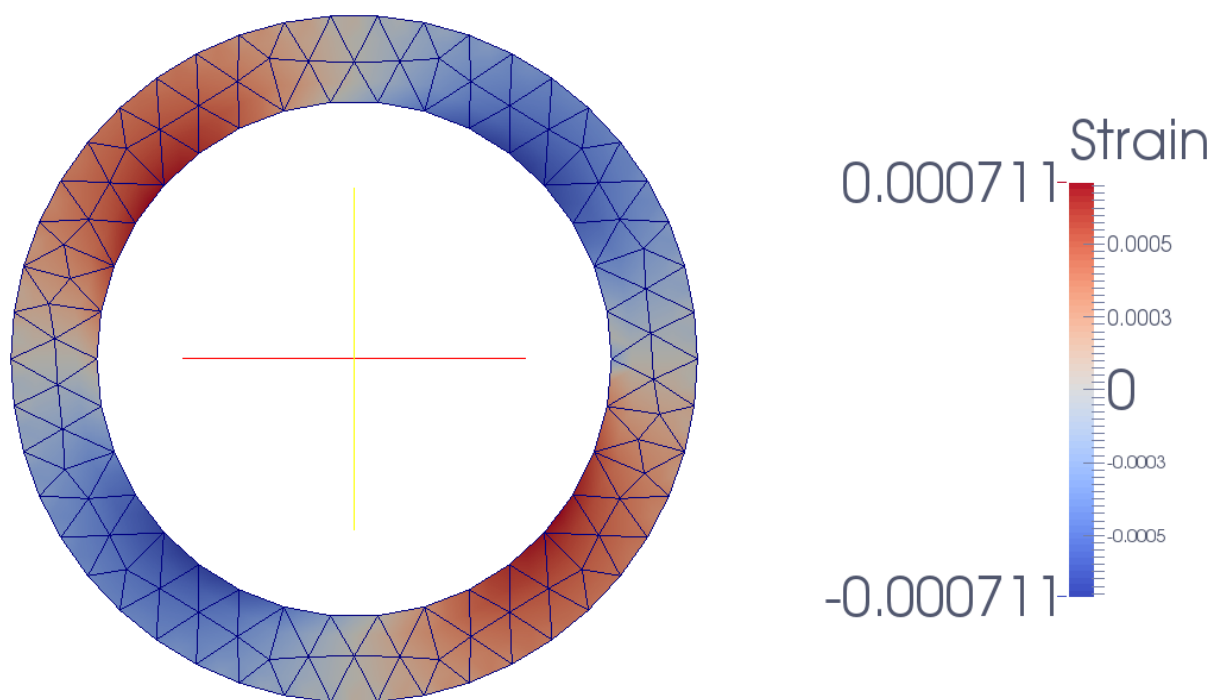


Figura 18 - Solução de  $\epsilon_{xy}$  para a malha de 176 elementos triangulares (Paraview)

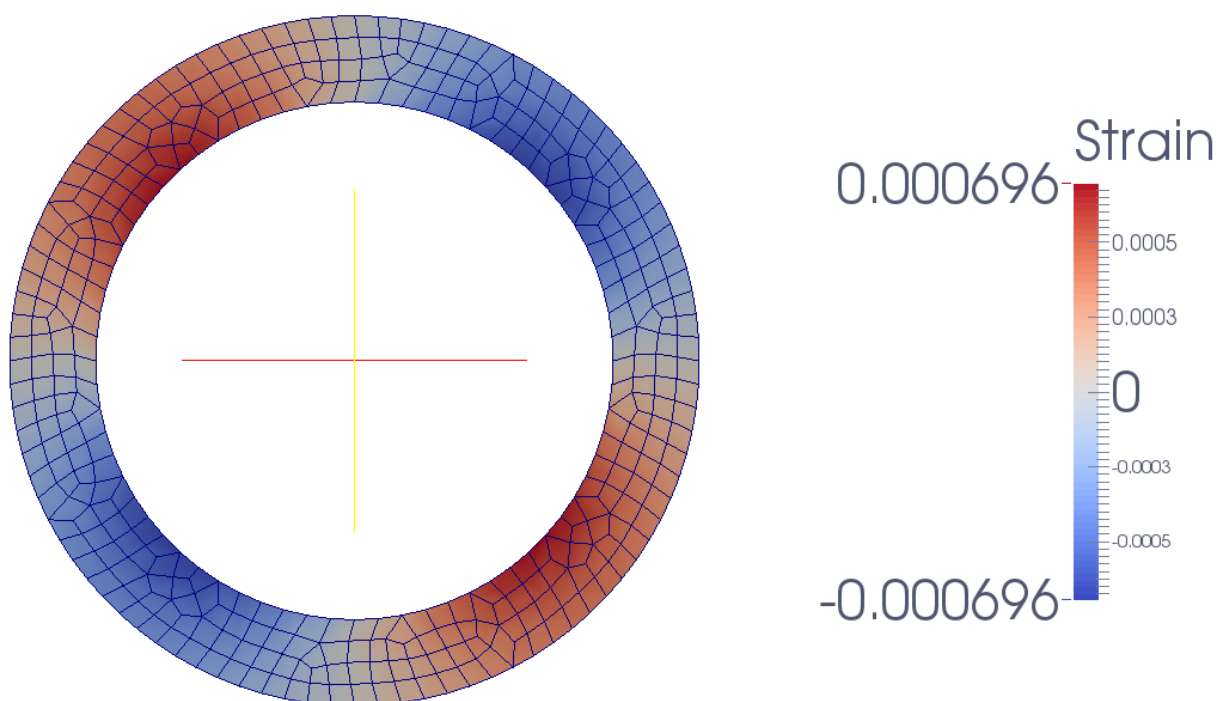
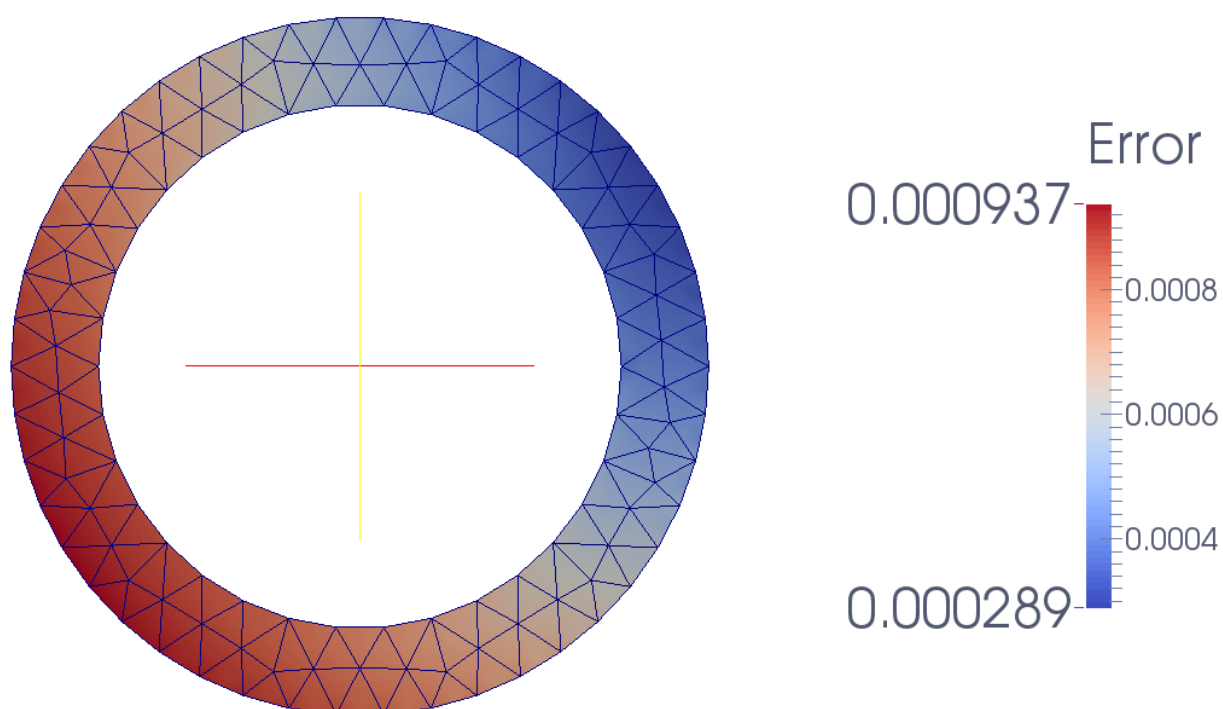


Figura 19 - Solução de  $\epsilon_{xy}$  para a malha de 360 elementos quadrangulares (Paraview)

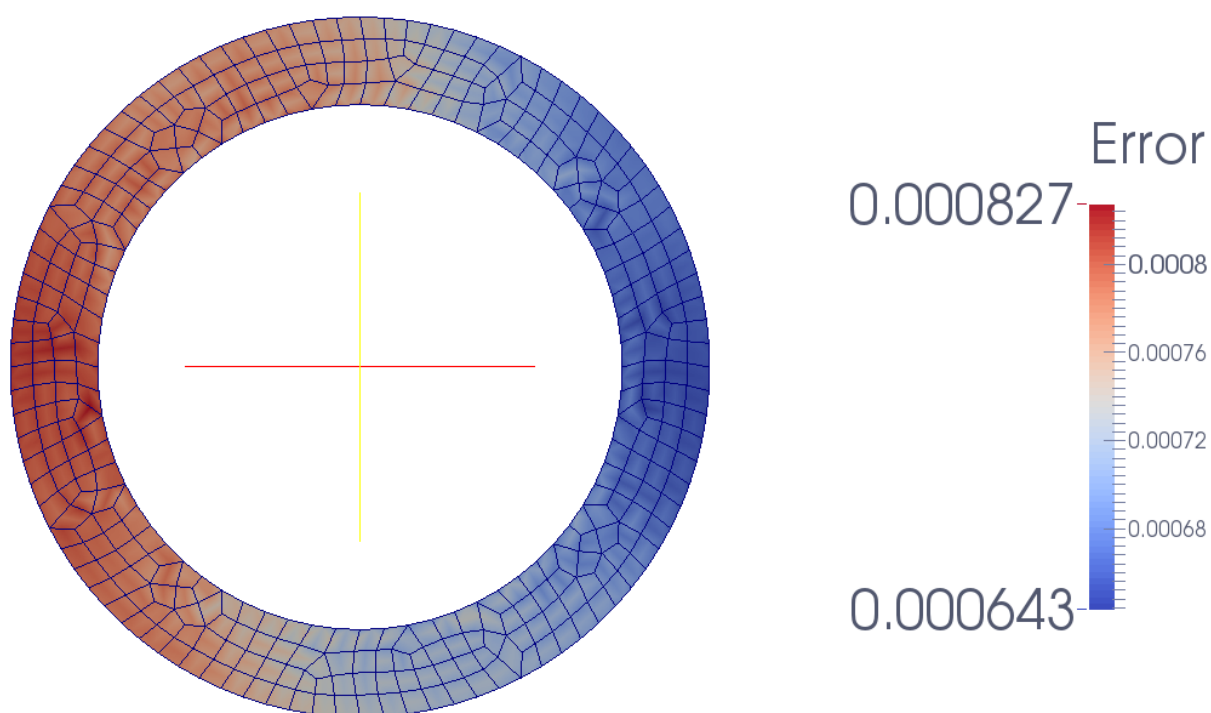
### 6.3 ERRO

Por se tratar de um método aproximado de resolução de equações diferenciais é de extrema importância a valoração da precisão da solução (ODEN, CAREY e BECKER). Para isso é calculado o erro, ou seja, a diferença entre a solução numérica (aproximada) e a solução analítica (exata).

As figuras a seguir demonstram graficamente a distribuição do módulo do erro do deslocamento para as malhas com 176 elementos triangulares quadráticos e malha com 360 elementos quadrangulares bi lineares respectivamente.

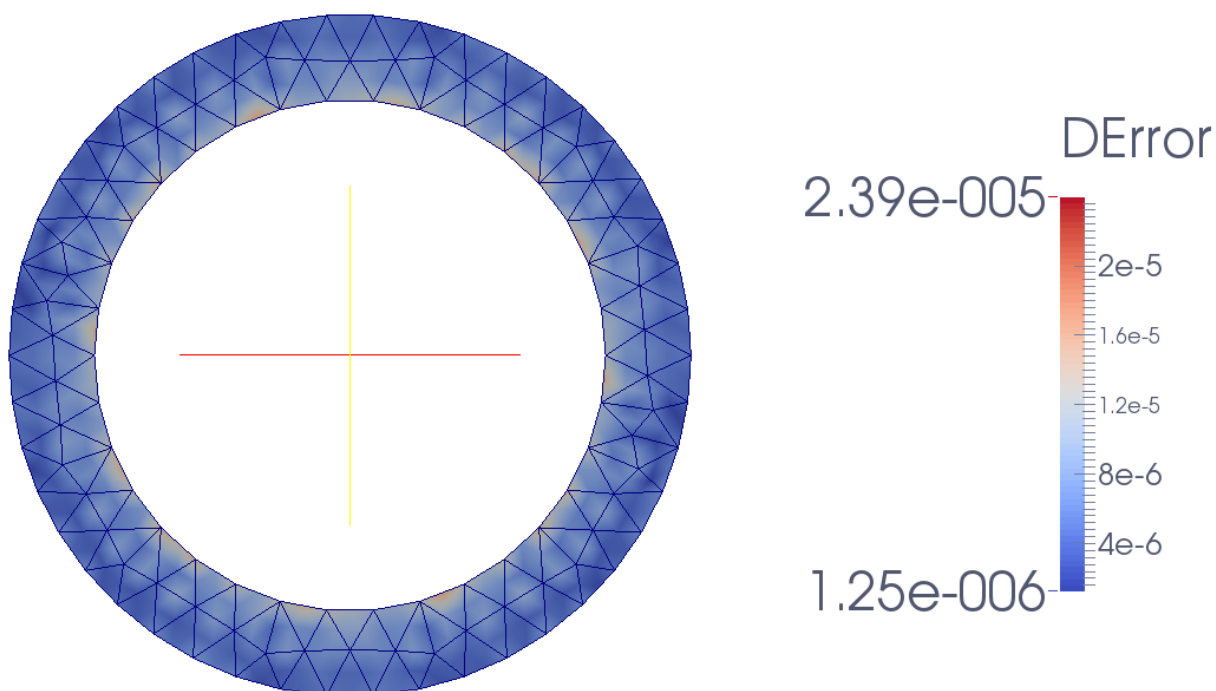


**Figura 20 - Erro do deslocamento para malha de 176 elementos triangulares (Paraview)**

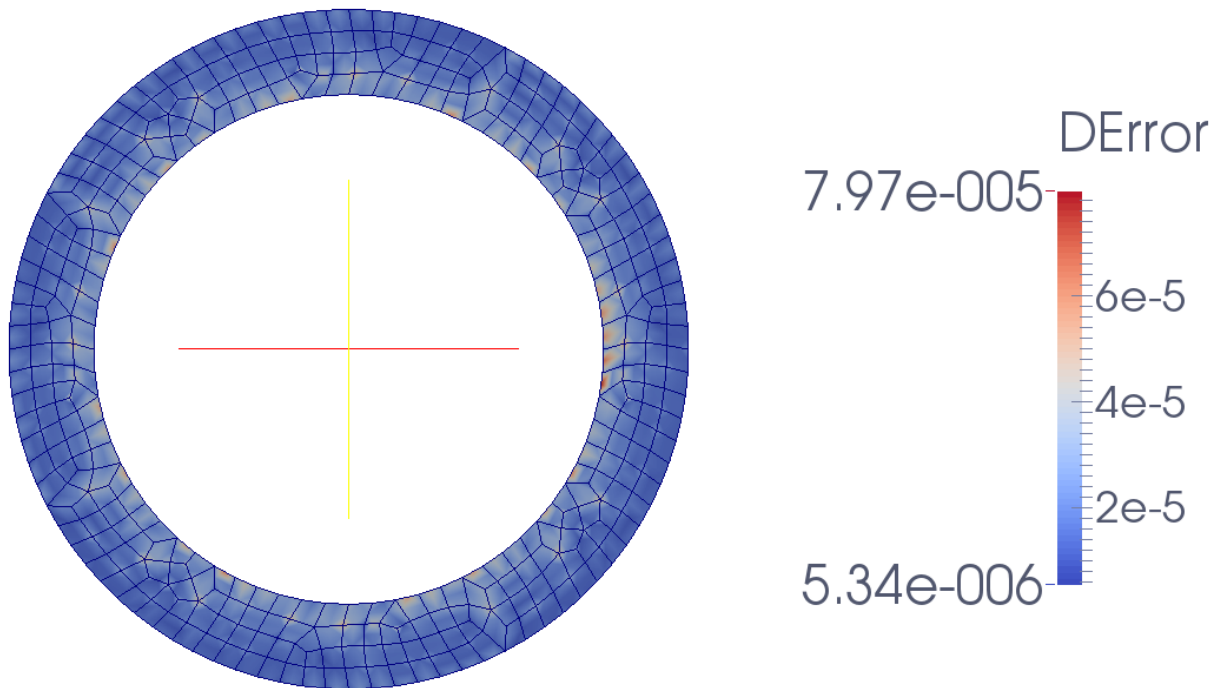


**Figura 21 - Erro dos deslocamentos para a malha de 360 elementos quadrangulares (Paraview)**

As figuras abaixo demonstram a distribuição do módulo do erro da deformação para as malhas com 176 elementos triangulares quadráticos e malha com 360 elementos quadrangulares bi lineares respectivamente.



**Figura 22 - Erro das deformações para a malha de 176 elementos triangulares (Paraview)**



**Figura 23 - Erro das deformações para a malha de 360 elementos quadrangulares (Paraview)**

#### 6.4 CONVERGÊNCIA

Para precisar a convergência do erro com o refino da malha foram utilizados dois parâmetros: a norma da energia e a norma do erro da sua derivada, ambas integradas em todo o domínio do problema.

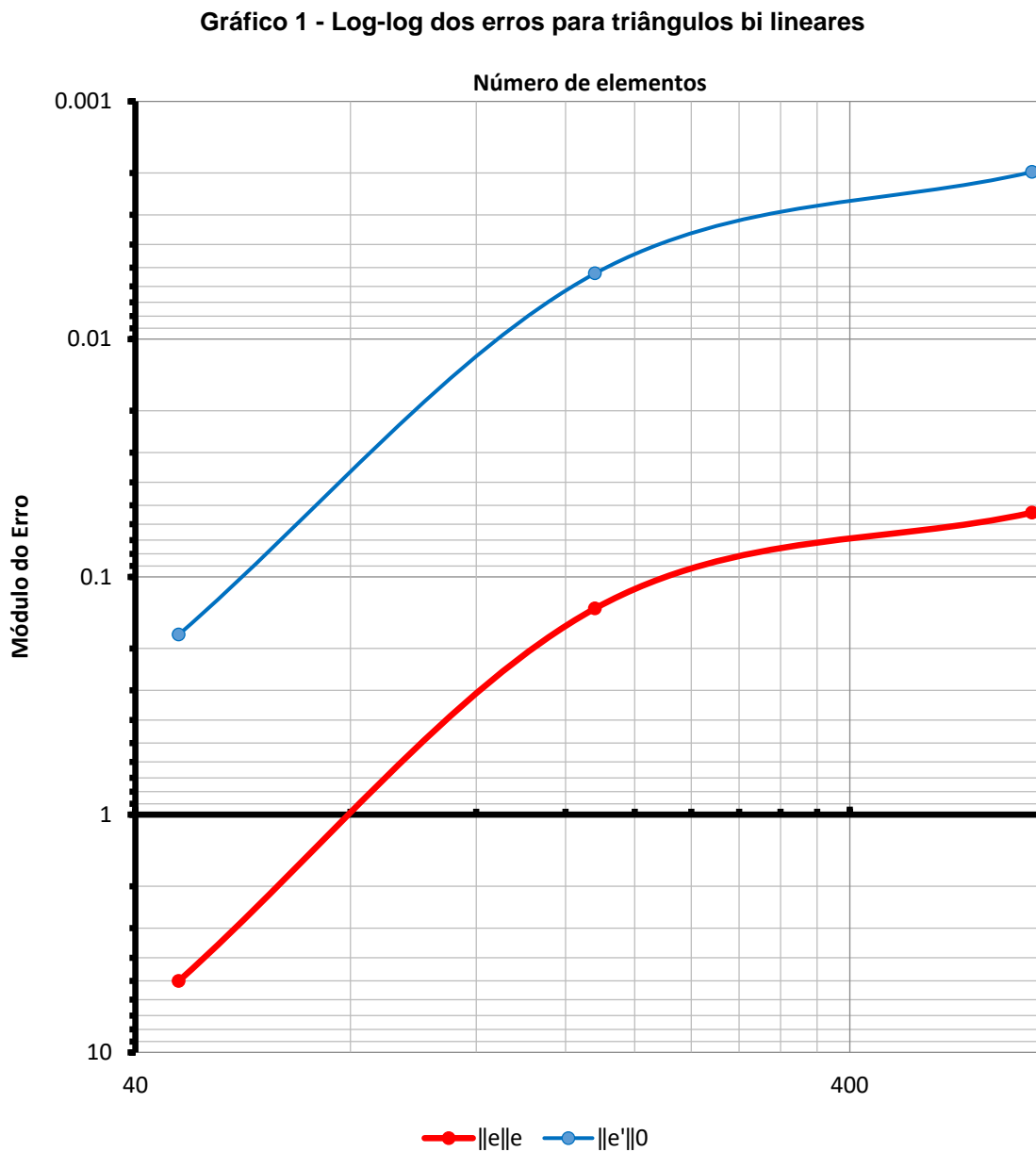
Abaixo seguem as equações da energia e da norma dos erros respectivamente.

$$\|e\|_E = \sqrt{\int [(e')^2 + e^2] \cdot dV}$$

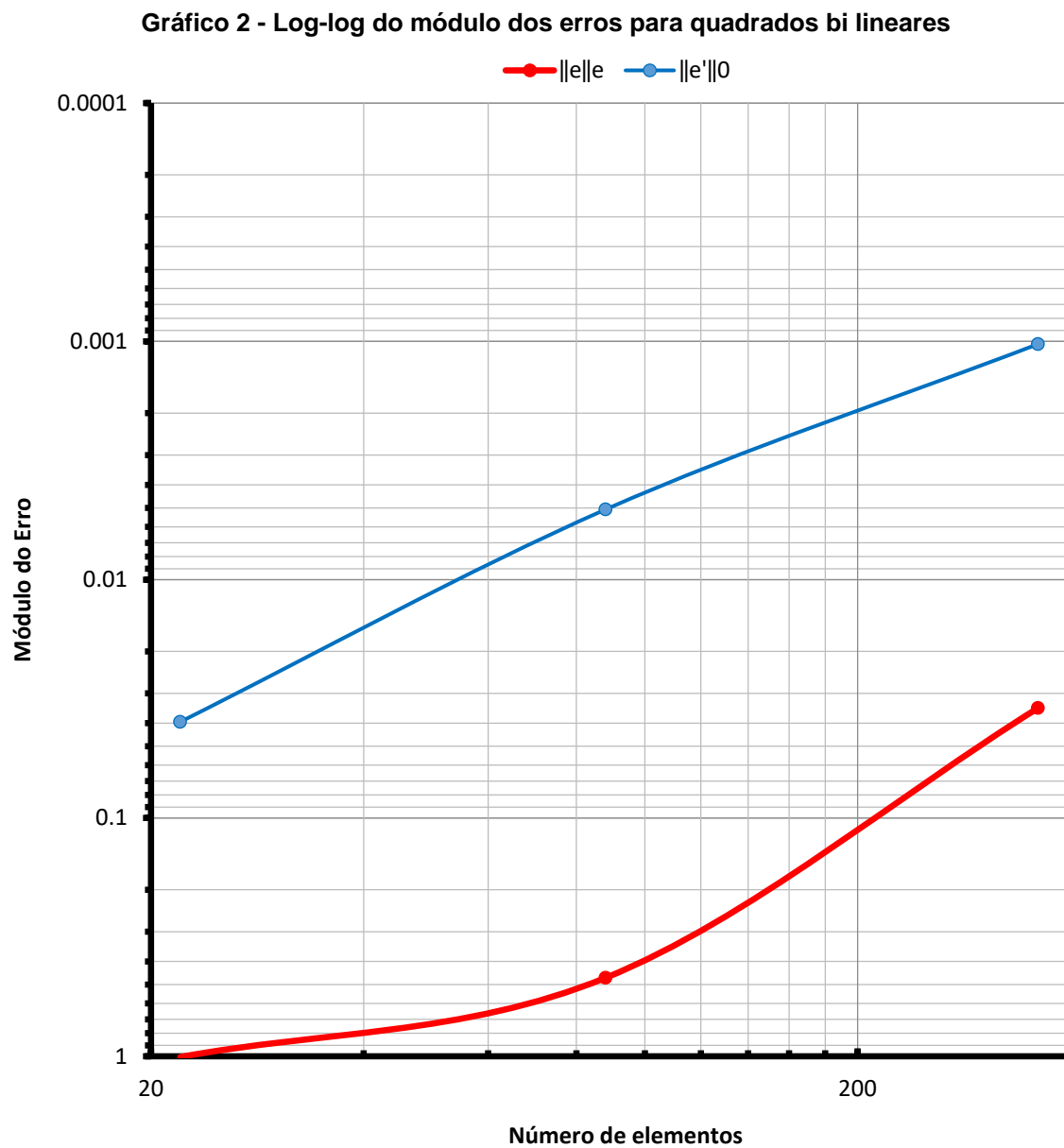
$$\|e'\|_0 = \sqrt{\int (e')^2 \cdot dV}$$

Para melhor interpretação da taxa de convergência os valores dos erros no domínio são apresentados em um gráfico logarítmico.

O gráfico 1 apresenta a convergência do erro para as malhas compostas de elementos triangulares bi lineares.

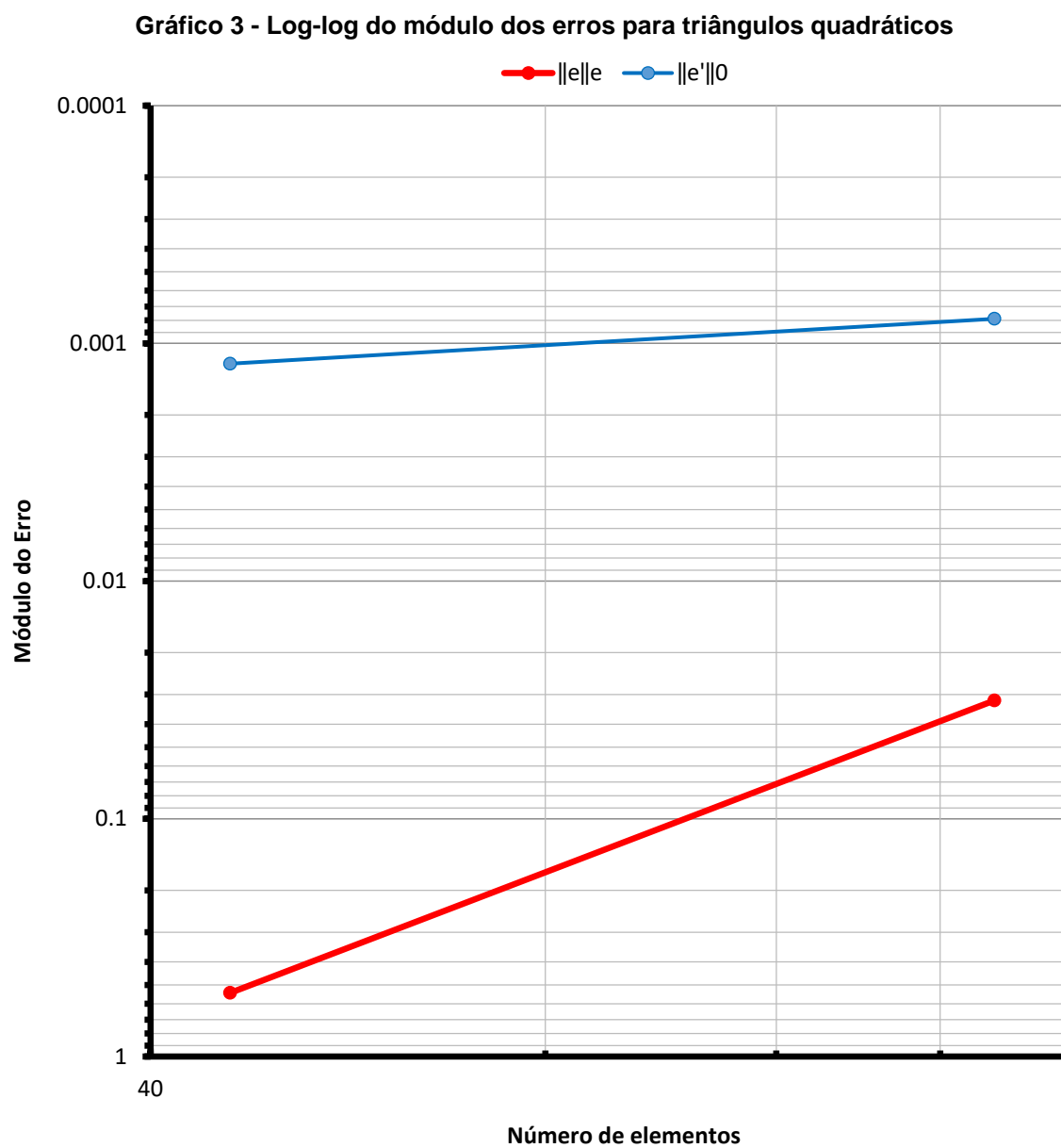


O gráfico 2 apresenta a convergência do erro para as malhas compostas de elementos quadrangulares bi lineares.



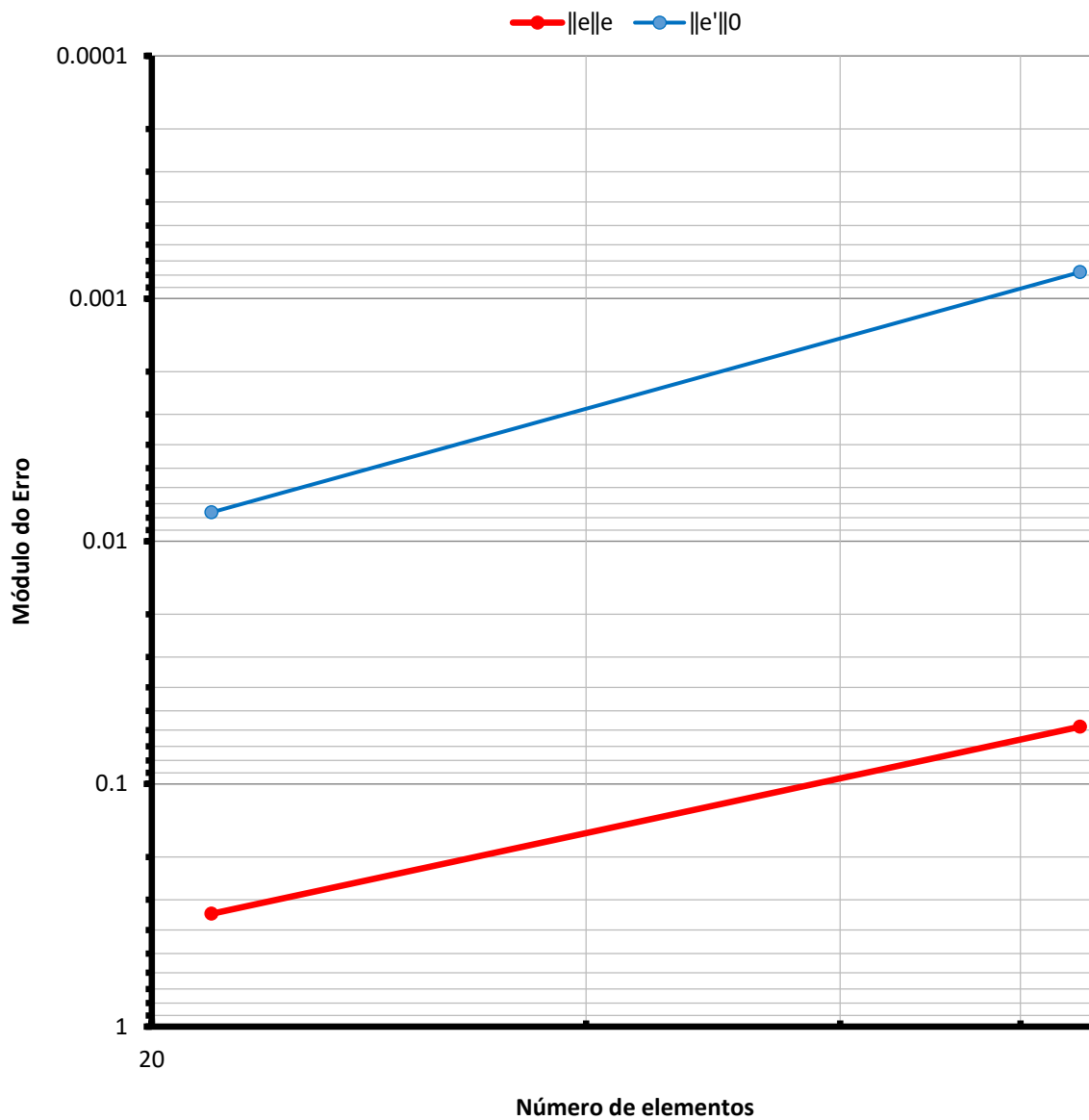


O gráfico 3 apresenta a convergência do erro para as malhas compostas de elementos triangulares quadráticos.



O gráfico 4 apresenta a convergência do erro para as malhas compostas de elementos quadrangulares quadráticos.

Gráfico 4 - Log-log do módulo dos erros para elementos quadrangulares quadráticos



## 7 CONCLUSÃO

O código obteve êxito em solucionar equações diferenciais de forma aproximada pelo método dos elementos finitos, e calcular os erros e taxas de convergência. Também está apto a ler arquivos de extensão “.msh” e gerar arquivos de solução gráfica de extensão “.vtk”.

Para obter maior velocidade de processamento e consequentemente maior viabilidade para o cálculo de soluções para malhas mais densas de elementos de altas ordens seria necessária a implementação de um “solver” de matrizes esparsas. Pois, as soluções com matrizes de rigidez completas exigem um alto consumo de memória e tempo de processamento, tornando improdutivo a resolução de problemas de alto grau de refinamento e tridimensionais.

Através do exemplo de aplicação do código MEF pôde-se notar que para o problema proposto os elementos quadráticos possuíram os menores erros e obtiveram uma taxa de convergência mais suave do que os elementos bi lineares, apesar de os dados terem sido obtidos apenas de duas malhas refinadas para elementos quadráticos.

O desenvolvimento do curso foi satisfatório do ponto de vista prático e teórico. Exigindo a constante evolução do código e aperfeiçoamento dos métodos durante o período letivo. A biblioteca NeoPZ foi fundamental para nortear a lógica de desenvolvimento facilitando o entendimento dos processos necessários para a criação do código.

Conclui-se que o aperfeiçoamento de uma biblioteca de MEF deve ser constante, devendo levar como princípio a robustez do código evitando sobreposição de dados e criando mecanismos de quebras para eventuais erros de inserção.

## BIBLIOGRAFIA

DELMAS, J. **Functions of form and points of integration finite elements**. [S.l.]. 2017.

ENES, S. **Shape Functions generation, requirements, etc**. Ruhr Universität Bochum. Bochum. 2009.

GMSH - A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Disponível em: <<http://gmsh.info/>>.

LABMEC - LABORATÓRIO DE MECÂNICA COMPUTACIONAL. NeoPZ. Disponível em: <<https://github.com/labmec/neopz>>.

OBERT, L.; DUVALL, W. I. **Rock Mechanics and the Desing of Structures in Rock**. 1ª. ed. [S.l.]: John Wiley & Sons Inc, 1967.

ODEN, T.; CAREY, G. F.; BECKER, E. B. **Finite Elements An Introduction**. Nova Jersey: Prentice-Hall, v. I.

PARAVIEW. Disponível em: <<https://www.paraview.org>>.

PRESS, W. H. et al. **Numerical Recipes - The Art of Scientific Computing**. 3ª. ed. Cambridge: Cambridge University Press, 2007.