

About.com. Artigos de medicina – anatomia da pele humana. Acesso: <http://dermatology.about.com/cs/skinanatomy/a/anatomy.htm> em 5 dez. 2005.

Alvim, C., **Análise numérica de membranas e biomembranas viscoelásticas submetidas à expansão axissimétrica**, Dissertação de Mestrado, PUC - Rio, 1999.

Austad E. D., e Gregory L. Rose, **A Self – Inflating Tissue Expander**, Plastic and Reconstructive Surgery, 1982.

Austad E. D., Pasyk K. A., McClatchey K. D., Cherry G. W., **Histomorphologic Evaluation of Guinea Pig Skin and Soft Tissue after Controlled Tissue Expansion**, Plastic and Reconstructive Surgery , 1982.

Calladine, C. R., **Theory of Shell Structures**. Cambridge University Press, Cambridge, 1983.

Carvalho, C., **Análise Numérica e Experimental do Comportamento da Pele Humana sob Expansão**, Tese de Doutorado, PUC - Rio, 2004.

Ernest, K., Manders, E.K., Michael, J., Schenden, **Soft – Tissue Expansion: Concepts and Complications**, Plastic and Reconstructive Surgery, 1984.

Fragoso, E.G., **Análise Estática Teórica e Experimental do Pericárdio Bovino**, Dissertação de Mestrado, PUC - Rio, 1995.

Fung, Y., **Biomechanics: Mechanical Properties of Living Tissues**, Springer-Verlag, New York, 1981

Green A. E., Adkins, J. E., **Large Elastic Deformations and Non-Linear Continuum Mechanics**, Oxford University Press, 1960.

Green A. E., Zerna, W., **Theoretical Elasticity**, Oxford University Press, 1968.

Hetenyi, M., **Beams on elastic foundation : theory with applications in the fields of civil and mechanical engineering** . Ann Arbor : University of Michigan Press, 1974.

Krayterman, B.L., **Nonlinear Analyses of Axisymmetric Membranes with Shooting Method**, J. Struct. Eng., ASCE 116, 1857-1876, 1990.

Lockett, F. J., **Creep and Stress- Relaxation Experiments for Non-Linear Materials**, Pergamon Press, 1965.

Lopes, S., **Comportamento Não-Linear e Instabilidade de Membranas e Cascas Hiperelásticas**, Tese de Doutorado, PUC - Rio 2003.

Maple. <http://www.maplesoft.com>.

Ogden, R. W., **Large Deformation Isotropic Elasticity – On the correlation of theory and experiment for incompressível rubberlike solids**, Proc. Royal Society London, A326, 565-584, 1972.

Pamplona, D.; Bevilacqua, L., **Large Deformations under Axial Force and Moment Load of Initially Flat Membranes**, Int. J. Non-Linear Mechanics 27, 639-650, 1992.

Reissner, E., **A Note on Deflections of Plates on a Viscolelastic Foundation**. Journal of Applied Mechanics, Vol. 25, TRANS. ASME, Vol. 80, 1958.

Rivlin, R. S., Thomas, A. G., **Large Elastic Deformations of Isotropic Materials: Strain Distribution around a Hole in a Sheet**, Int. J. Non-Linear Mechanics, 2, 289-298, 1950.

Samani, A., Bishop, J., Yaffe, M. J., Plewes D. B., **Biomechanical 3D Finite Element Modeling of the Human Breast Using MRI Data**. *IEEE Transactions on Medical Imaging*, 2001.

Serebrenick, G., **Análise da estabilidade de colunas esbeltas parcialmente enterradas em uma fundação elástica não-linear**, Dissertação de Mestrado, PUC – Rio, 2004.

Timoshenko, S. P., Woinowsky-Krieger, S., **Theory of Plates and Shells**, Int. Student Edition, McGraw Hill, 1959.

Treolar, L.R.G., **The Physics of Rubber Elasticity**, Oxford University Press, UK, 1975.

Vanderplaats, G. N., **Numerical Optimization Techniques for engineering desing – with applications**, McGraw-Hill, Inc, 1984.

Apêndice 1

Cálculo das coordenadas tridimensionais de uma expansão

A.1.1.

Aparato utilizado

Os espelhos são ligados a barra principal, através de duas barras de alumínio de mesmas dimensões. Esta ligação nos permite manter o mesmo ângulo de abertura para os dois espelhos, com relação ao posicionamento do expansor.

Para uma melhor visualização e precisão das medidas feitas nas réguas, utilizamos réguas de alumínio.



Figura A.1.1 – Ilustração do aparato



Figura A.1.2 – Foto do aparato

O ensaio fotográfico a seguir foi feito com uma *semi-esfera de isopor* de raio R aproximadamente igual a cinquenta milímetros a fim de comprovar a metodologia a ser utilizada para o cálculo do z dos pontos. Foram marcados pontos sobre a semi-esfera de isopor aproximadamente no centro e nas direções vertical e horizontal distados de dez milímetros.

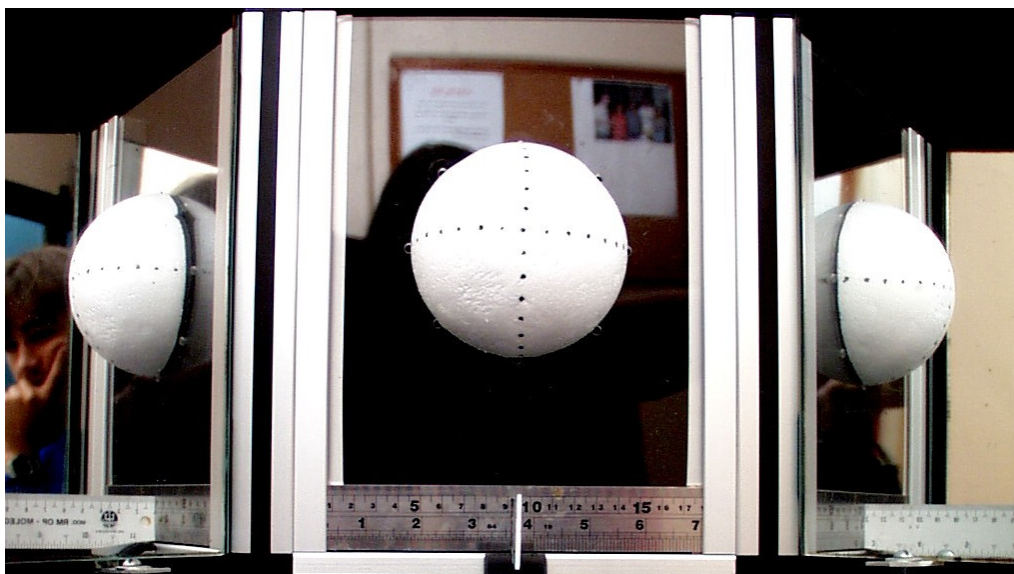


Figura A.1.3 - Aparato com semi-esfera de isopor

A.1.2. Realização das medições

Para calcular a altura z dos pontos marcados na semi-esfera de isopor, utilizamos os conceitos abaixo tendo em vista a coleta dos dados em *pixel* no programa *Microsoft Photo Editor*.

DESENVOLVIMENTO DA FÓRMULA PARA ACHAR O Z DE UM PONTO

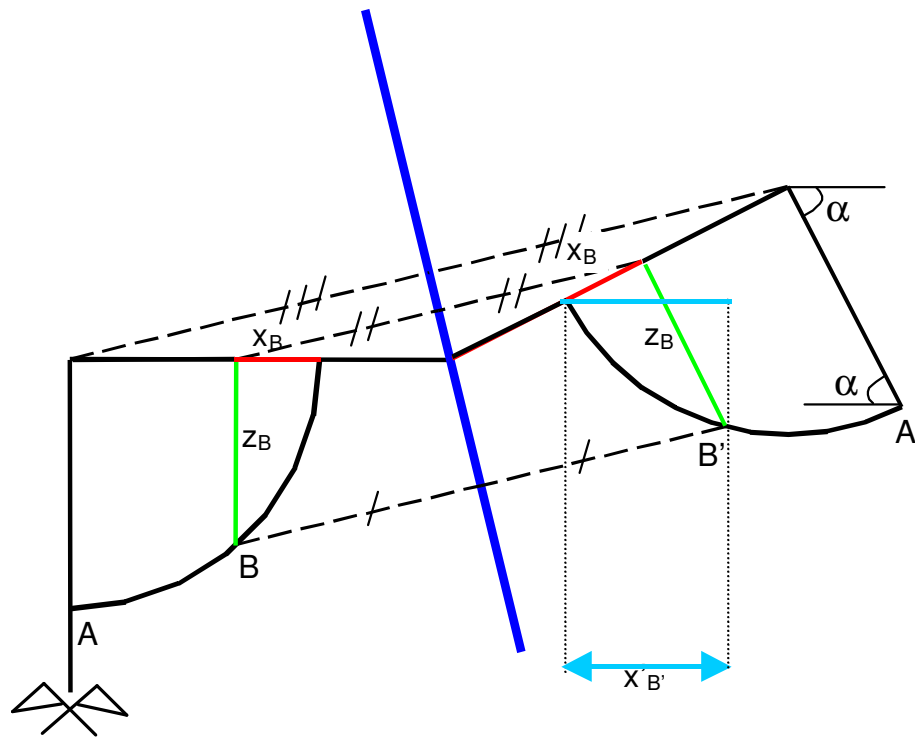


Figura A.1.4 – Esquema da reflexão no espelho

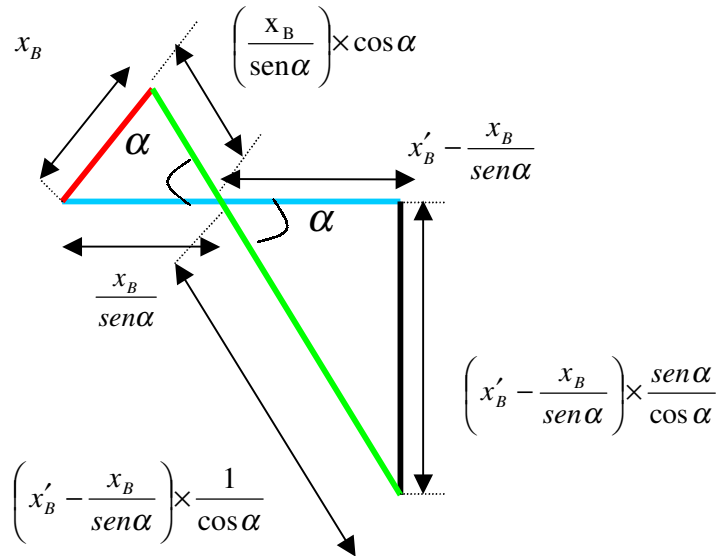
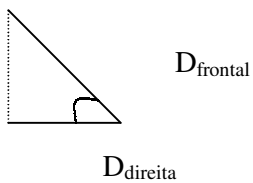


Figura A.1.5 – Geometria das coordenadas

A.1.2.1 Cálculo do ângulo α dos espelhos

Espelho da direita

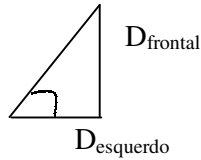


$$\cos \alpha_d = \frac{D_{direita}}{D_{frontal}}$$

$$\alpha_d = \cos^{-1} \left(\frac{18,125}{22,000} \right)$$

$$\alpha_d = 34,5^\circ$$

Espelho da esquerda



$$\cos \alpha_e = \frac{D_{esquerdo}}{D_{frontal}}$$

$$\alpha_e = \cos^{-1} \left(\frac{18,375}{22,000} \right)$$

$$\alpha_e = 33,4^\circ$$

A.1.2.2

Desenvolvimento dos cálculos

Triângulo menor

$$\operatorname{sen} \alpha = \frac{x_B}{\text{hipotenusa}} \quad \Rightarrow \quad \text{hipotenusa} = \frac{x_B}{\operatorname{sen} \alpha}$$

$$\cos \alpha = \frac{\text{cat. adjacente}}{\left(\frac{x_B}{\operatorname{sen} \alpha} \right)} \quad \Rightarrow \quad \text{cat. adjacente} = \left(\frac{x_B}{\operatorname{sen} \alpha} \right) \times \cos \alpha$$

Triângulo maior

$$\cos \alpha = \frac{\left(x'_B - \frac{x_B}{\operatorname{sen} \alpha} \right)}{\text{hipotenusa}} \quad \Rightarrow \quad \text{hipotenusa} = \left(x'_B - \frac{x_B}{\operatorname{sen} \alpha} \right) \times \frac{1}{\cos \alpha}$$

$$\operatorname{sen} \alpha = \frac{\text{cat. oposto}}{\left(x'_B - \frac{x_B}{\operatorname{sen} \alpha} \right) \times \frac{1}{\cos \alpha}} \quad \Rightarrow \quad \text{cat. oposto} = \left(x'_B - \frac{x_B}{\operatorname{sen} \alpha} \right) \times \frac{\operatorname{sen} \alpha}{\cos \alpha}$$

Desenvolvimento das fórmulas

$$z_B = \left(\frac{x_B}{\operatorname{sen} \alpha} \right) \times \cos \alpha + \left(x'_B - \frac{x_B}{\operatorname{sen} \alpha} \right) \times \frac{1}{\cos \alpha}$$

$$z_B = x_B \times \left(\frac{\cos \alpha}{\operatorname{sen} \alpha} \right) + x'_B \times \left(\frac{1}{\cos \alpha} \right) - x_B \times \left(\frac{1}{\operatorname{sen} \alpha \cos \alpha} \right)$$

$$z_B = x_B \times \left(\frac{\cos \alpha}{\operatorname{sen} \alpha} - \frac{1}{\operatorname{sen} \alpha \cos \alpha} \right) + x'_B \times \left(\frac{1}{\cos \alpha} \right)$$

$$z_B = x_B \times \left(\frac{\cos^2(\alpha) - 1}{\operatorname{sen} \alpha \cos \alpha} \right) + x'_B \times \left(\frac{1}{\cos \alpha} \right)$$

$$z_B = x_B \times \left(\frac{-\operatorname{sen}^2(\alpha)}{\operatorname{sen} \alpha \cos \alpha} \right) + x'_B \times \left(\frac{1}{\cos \alpha} \right)$$

$$z_B = -x_B \times \left(\frac{\operatorname{sen} \alpha}{\cos \alpha} \right) + x'_B \times \left(\frac{1}{\cos \alpha} \right)$$

$$z_B = \frac{x'_B - x_B \times \operatorname{sen} \alpha}{\cos \alpha}$$

A.1.3. Validação da Metodologia

Medição de 1cm sobre as "réguas de referência"								
Régua da esquerda			Régua frontal			Régua da direita		
2,484	21,485	19,000	328,479	349,476	21,000	864,497	880,495	16,000
21,485	41,485	20,000	349,476	371,477	22,000	880,495	897,494	17,000
41,485	60,486	19,000	371,477	394,477	23,000	897,494	915,496	18,000
60,486	77,487	17,000	394,477	415,477	21,000	915,496	933,496	18,000
77,487	97,487	20,000	415,477	437,477	22,000	933,496	952,497	19,000
97,487	114,488	17,000	437,477	460,477	23,000	952,497	971,498	19,000
114,488	132,488	18,000	460,477	482,477	22,000	971,498	989,498	18,000
132,488	149,489	17,000	482,477	504,477	22,000	989,498	1009,497	20,000
		147,000	504,477	526,477	22,000			145,000
		18,375	526,477	547,477	21,000			18,125
			547,477	569,477	22,000			
			569,477	592,477	23,000			
			592,477	614,477	22,000			
			614,477	636,477	22,000			
			636,477	658,478	22,000			
			658,478	680,478	22,000			
					352,000			
					22,000			

Figura A.1.6 - Medição de 1cm sobre as "réguas de referência"
(pixels)

Origens de referência	
Espelho da esquerda	Espelho da direita
Oesquerda = 181,261	Odireita = 832,268
Ofrontal = 406,234	Ofrontal = 624,238

Figura A.1.7 - Origens de referência (pixels)

Medição dos pontos da semi-esfera de isopor											
Espelho da esquerda				Espelho frontal				Espelho da direita			
Horizontal		Vertical		Horizontal		Vertical		Horizontal		Vertical	
x	y	x	y	x	y	x	y	x	y	x	y
67	269	138	368	412	231	520	140	843	270	872	376
83	262	123	362	422	227	521	148	862	271	887	371
97	259	107	353	436	223	522	162	881	272	905	361
113	259	93	339	453	220	522	180	898	274	920	349
132	258	81	324	476	218	521	200	915	274	931	334
151	259	72	306	497	220	518	221	930	275	940	317
171	261	68	288	518	221	520	244	939	277	946	299
		67	269	542	223	518	267	947	280	947	280
		69	252	563	224	517	289			945	262
				584	227	516	309				
				600	231	516	327				
				612	234	516	338				
				618	237	516	346				
Distâncias relativas à semi-esfera de isopor											
Espelho da esquerda				Espelho frontal				Espelho da direita			
Horizontal		Vertical		Horizontal		Vertical		Horizontal		Vertical	
x	y	x	y	x	y	x	y	x	y	x	y
16	7	15	6	10	4	1	8	19	1	15	5
14	3	16	9	14	4	1	14	19	1	18	10
16	0	14	14	17	3	0	18	17	2	15	12
19	1	12	15	23	2	1	20	17	0	11	15
19	1	9	18	21	2	3	21	15	1	9	17
20	2	4	18	21	1	2	23	9	2	6	18
104	14	1	19	24	2	2	23	8	3	1	19
17,33333	2,33333	2	17	21	1	1	22	104	10	2	18
		73	116	21	3	1	20	14,85714	1,428571	77	114
		9,125	14,5	16	4	0	18			9,625	14,25
				12	3	0	11				
				6	3	0	8				
				206	32	12	206				
				17	3	1	17,16667				

Figura A.1.8 - Medição dos pontos da semi-esfera de isopor (pixels)

Medição das distâncias XB e X'B							
Espelho da esquerda				Espelho da direita			
XB		X'B		XB		X'B	
(pixel)	(cm)	(pixel)	(cm)	(pixel)	(cm)	(pixel)	(cm)
6,000	0,273	10,000	0,544	6,000	0,273	30,000	1,655
16,000	0,727	30,000	1,633	12,000	0,545	49,000	2,703
30,000	1,364	49,000	2,667	24,000	1,091	66,000	3,641
74,000	2,136	68,000	3,701	40,000	1,818	83,000	4,579
70,000	3,182	84,000	4,571	61,000	2,773	98,000	5,407
91,000	4,136	98,000	5,333	82,000	3,727	107,000	5,903
112,000	5,091	114,000	6,204	106,000	4,818	115,000	6,345

Figura A.1.9 - Medição das distâncias XB e X'B (pixels e cm)

Cálculo do z			
Medidas		Verdadeira	
Espelho da esquerda	Espelho da direita	Esquerda	Direita
0,472	1,821	0,5	1,5
1,477	2,905	1,5	2,5
2,295	3,668	2,5	3,5
3,025	4,307	3,0	4,2
3,377	4,655	3,5	4,3
3,661	4,601	4,0	4,4
4,074	4,388	4,5	4,5

Figura A.1.10 – Cálculo do z

Comprovamos, dessa forma, a metodologia a ser utilizada no cálculo do z dos pontos. Visto que a aproximação das medidas calculadas e das dimensões verdadeiras é mínima.

A.1.4.

Preparação da membrana de borracha

Fixamos a membrana sobre papel quadriculado com a ajuda da fita crepe, garantindo que a superfície da membrana estivesse lisa, isto é, sem ondulações visíveis, porém sem estar tracionada. Neste papel quadriculado (escala 5 mm) desenhemos um círculo de raio de 66 mm e fizemos marcações de pontos distados de 10 milímetros nas direções: vertical, horizontal e diagonais (inclinadas de 45° das direções principais). Com o auxílio de um pilot fizemos essas mesmas marcações na membrana de borracha.

A.1.5.

Análise da expansão de um lençol de borracha

Inicialmente foram feitos ensaios com oito parafusos de fixação, onde se verificou a dificuldade de realizar algumas medições. Foram eliminados quatro parafusos de fixação e se verificou a interferência dessa eliminação nos resultados. Este ensaio foi realizado inserindo ar no expansor, com uma membrana de espessura de 0,168mm.

Pressão 8 parafusos	Cálculo do Z	Pressão 4 parafusos	Cálculo do Z
0	0,1705	0	0,9138
100	1,6581	100	1,0179
199	2,2539	200	1,8020
300	2,7539	299	2,2112
400	3,3454	400	2,7658
501	3,4558	500	3,3758
600	3,6224	601	3,6087
700	3,9558	699	3,8758
802	4,5266	800	4,3854
901	4,6766	902	5,1845

Figura A.1.11 - Tabelas de comparação do cálculo do z do centro

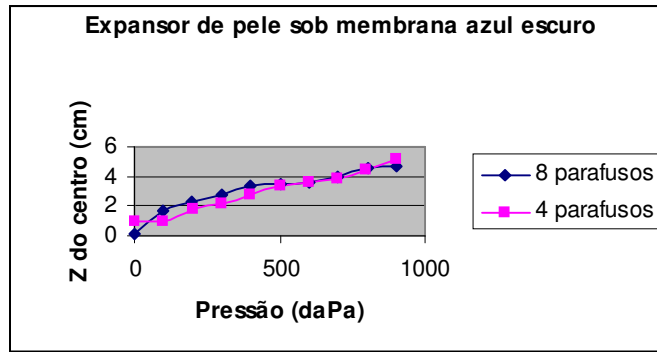


Figura A.1.12 - Gráfico z do centro (cm) versus Pressão (da Pa)

Apêndice 2

Testes de bases elásticas para a experiência

Para conseguirmos uma base elástica que simulasse bem a gordura, fizemos diversos ensaios com diferentes materiais, até chegarmos a um resultado satisfatório que foi a bexiga com um material viscoso no seu interior (Geleca). Utilizamos espuma tradicional, espuma especial (que preservava a forma), gelatina e uma bexiga com margarina no seu interior que foram testadas em ensaios de compressão, mas não tiveram bons resultados.

A.2.1

Espuma de Tradicional de Densidade D23

Os ensaios a seguir foram de compressão utilizando pesos como carga. A amostra de espuma possui dimensões 6 cm x 6 cm x 2 cm.

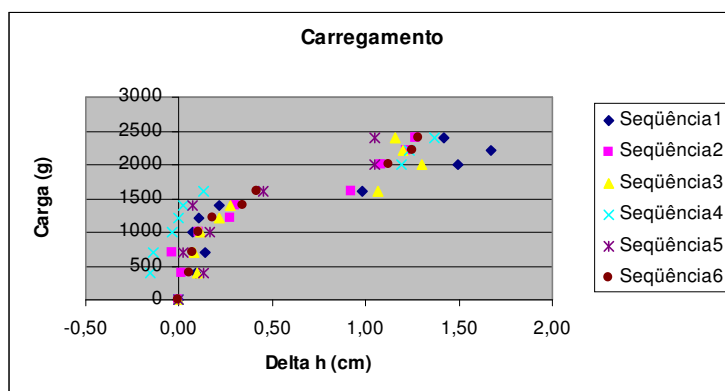


Figura A.2.1 – Carregamento do ensaio de compressão da espuma tradicional

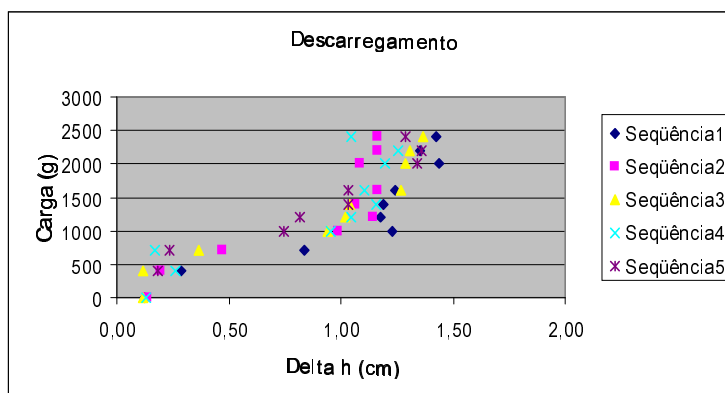


Figura A.2.2 - Descarregamento do ensaio de compressão da espuma tradicional

Notamos pelas figuras A.2.1 e A.2.2 que o ensaio de compressão é dividido em duas etapas, com uma descontinuidade durante o ensaio tanto no carregamento quanto no descarregamento.

A.2.2 Espuma especial

Os ensaios a seguir foram de compressão utilizando pesos como carga. A figura A.2.3 ilustra a amostra de espuma utilizada. Esta é de material visco elástico, fabricada sem CFC e com expansão à base de água.



Figura A.2.3 – Espuma especial

As figuras A.2.4 e A.2.5 mostram o carregamento e o descarregamento do ensaio de compressão.

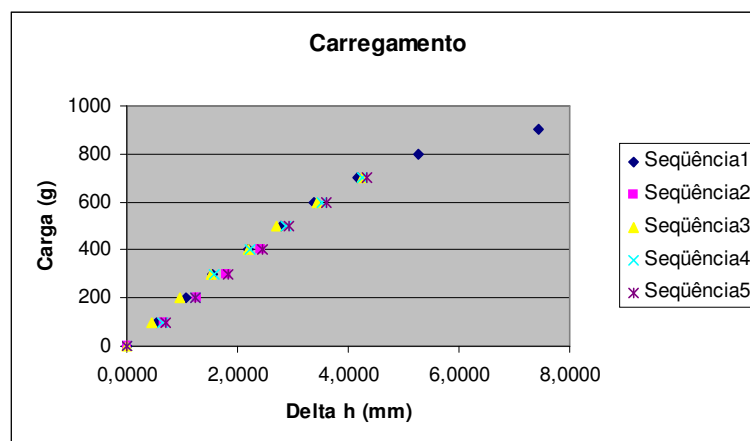


Figura A.2.4 - Carregamento do ensaio de compressão da espuma especial

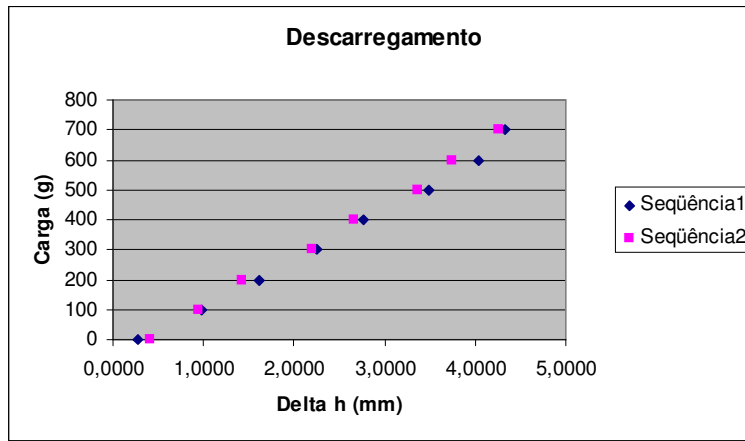


Figura A.2.5 - Descarregamento do ensaio de compressão da espuma especial

Não foi possível utilizar esse material devido à dificuldade de moldá-lo como uma base elástica plana.

A.2.3 Gelatina

Testamos a gelatina como um possível material para a base elástica, porém encontramos dificuldades no seu uso, já que a gelatina precisa ser conservada a uma temperatura constante para preservar as suas propriedades. A figura A.2.6 mostra a gelatina feita para o estudo.

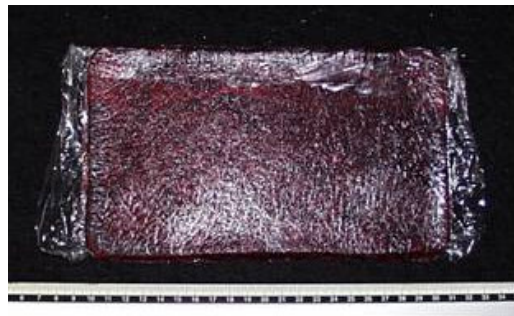


Figura A.2.6 – Gelatina

A.2.4 Margarina dentro de bexiga

Os ensaios a seguir foram de compressão utilizando pesos como carga. Colamos 400g de margarina dentro de uma bexiga de 15 cm de diâmetro.

Fechamos a bexiga com cola para látex e grampos. Foram feitos inicialmente furos com ilhóes na bexiga a fim de facilitar a fixação no aparato para futuros ensaios com o expansor de tecido. Estes furos foram reforçados com plástico grosso para reduzir o risco de romper o tecido da bexiga. Podemos ver na figura A.2.7 a margarina dentro da bexiga.



Figura A.2.7 – Margarina dentro da bexiga

Conservamos a margarina em geladeira e notamos que à medida que o tempo passava, a temperatura da margarina aumentava, fazendo com que ela perdesse a sua consistência, com isso diminuindo a sua resistência à compressão. Seria preciso manter a temperatura da margarina constante e isso iria requerer maiores preparativos.

Apêndice 3

Relaxamento

O ar foi inserido no expansor até atingir uma pressão de 900daPa. Deixamos, então, o aparato montado com expansor e membrana descansando a fim de observar seu comportamento no processo de Relaxamento.

t (min)	p (daPa)
0	900
1	897
2	894
3	893
4	892
5	891
10	887
15	883
20	880
25	877
30	875
40	870
50	865
60	861
75	854
90	847
105	841
120	835
140	825
160	817
180	808
210	794
240	782

Tabela A.3.1 - Tabela referente ao ensaio de relaxamento

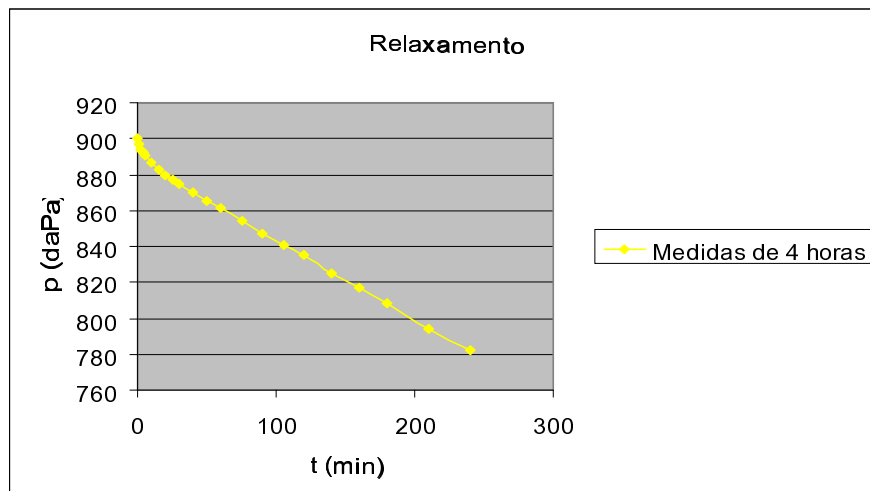


Figura A.3.1 - Gráfico pressão (da Pa) x tempo (minutos)

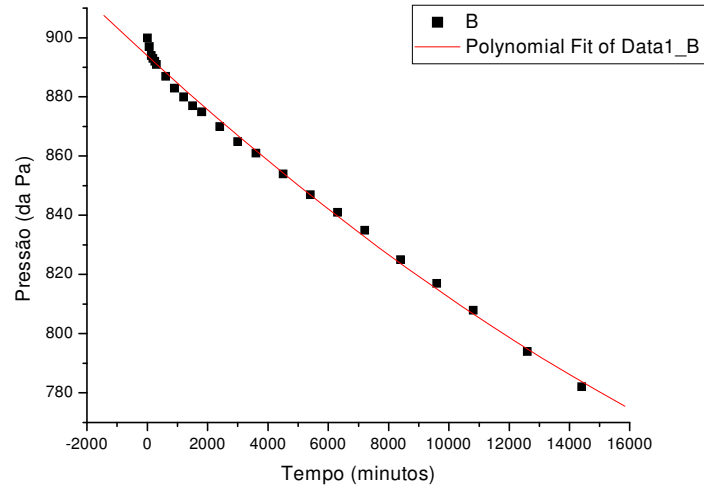


Figura A.3.2 - Gráfico da pressão (daPa) x tempo (minutos)

Em Lockett podemos ver mais experimentos de relaxamento para outros materiais (1965).

Apêndice 4

Arquivos do programa Maple 9

Neo-Hookeano

```
> restart; with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

Warning, the name changecoords has been redefined

```
> a:=1.5: #chute de U(0.000001)#
```

```
> tol:=0.001:
```

```
> dU:=((r'*F)+(z'*G))/lambda[1]^2:
```

```
> dV:=((z'*F)-(r'*G))/lambda[1]^2:
```

```
> dVol:=Pi*((r(rho))^2)*V(rho):
```

```
> F:=(lambda[1]/W[11])*((d(lambda[2])*((W[1]/lambda[2])-W[12])-  
((r'/(r(rho)*lambda[1]))*(lambda[1]*W[1]-lambda[2]*W[2]))))):
```

```
> G:=(lambda[1]^2/W[1])*(((lambda[1]*lambda[2]*P)/H  
+((z'/(r(rho)*lambda[1]))*lambda[2]*W[2]))):
```

```
> W:=C*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-3):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> W[1]:=diff(W,lambda[1]):
```

```
> W[11]:=diff(W[1],lambda[1]):
```

```
> W[12]:=diff(W[1],lambda[2]):
```

```
> Wi:=C*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-3):
```

```
> W[2]:=diff(Wi,lambda[2]):
```

```
> W[22]:=diff(W[2],lambda[2]):
```

```
> W[21]:=diff(W[2],lambda[1]):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> lambda[1]:=sqrt((diff(r(rho),rho)^2)+(diff(z(rho),rho)^2)):
```

```
> lambda[2]:=r(rho)/rho:
```

Derivadas de lambda[2] , r , z em relação a rho

```

> d(lambda[2]):=diff(lambda[2],rho):
> r':=diff(r(rho),rho):
> z':=diff(z(rho),rho):
> H:=1:C:=1:P:=1:
> for i from 1 to 5 do
>
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a,z
(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0}:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolr:=subs(dsol,r(rho)):
dsolz:=subs(dsol,z(rho)):
> if (abs(dsolr(1)-1)>tol) then
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a+
0.01,z(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0} end if:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolR:=subs(dsol,r(rho)):
dsolZ:=subs(dsol,z(rho)):
r(1)=dsolR(1);
> if (abs(dsolR(1)-1)>tol) then a:=a-((a+0.01)-a)*(dsolr(1)-1)/(dsolR(1)-dsolr(1))
end if;
> if (abs(dsolR(1)-1)<=tol) then i:=6 end if;
end do;
>dsol(1);

```

Neo-Hookeano – base elástica

```
> restart; with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

Warning, the name changecoords has been redefined

```
> a:=1.5: #chute de U(0.000001)#
```

```
> tol:=0.001:
```

```
> dU:=((r'*F)+(z'*G))/lambda[1]^2:
```

```
> dV:=((z'*F)-(r'*G))/lambda[1]^2:
```

```
> dVol:=Pi*((r(rho))^2)*V(rho):
```

```
> F:=(lambda[1]/W[11])*((d(lambda[2])*((W[1]/lambda[2])-W[12])-
((r'/(r(rho)*lambda[1]))*(lambda[1]*W[1]-lambda[2]*W[2]))):
```

```
> G:=(lambda[1]^2/W[1])*(((lambda[1]*lambda[2]*(P-k*a*z(rho)*
(r'/(lambda[1])))/H)+((z'/(r(rho)*lambda[1]))*lambda[2]*W[2])):
```

```
> W:=C*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-3):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> W[1]:=diff(W,lambda[1]):
```

```
> W[11]:=diff(W[1],lambda[1]):
```

```
> W[12]:=diff(W[1],lambda[2]):
```

```
> Wi:=C*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-3):
```

```
> W[2]:=diff(Wi,lambda[2]):
```

```
> W[22]:=diff(W[2],lambda[2]):
```

```
> W[21]:=diff(W[2],lambda[1]):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> lambda[1]:=sqrt((diff(r(rho),rho)^2)+(diff(z(rho),rho)^2)):
```

```
> lambda[2]:=r(rho)/rho:
```

Derivadas de lambda[2] , r , z em relação a rho

```
> d(lambda[2]):=diff(lambda[2],rho):
```

```
> r':=diff(r(rho),rho):
```

```
> z':=diff(z(rho),rho):
```

```
> H:=0.01363:C:= 0.359628038:P:= 0.004925779:k:=0.002:
```

```
> a:= 5:
```

```

> for i from 1 to 5 do
>
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a,z
(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0}:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolr:=subs(dsol,r(rho)):
dsolz:=subs(dsol,z(rho)):
> if (abs(dsolr(1)-1)>tol) then
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a+
0.01,z(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0} end if:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolR:=subs(dsol,r(rho)):
dsolZ:=subs(dsol,z(rho)):
r(1)=dsolR(1);
> if (abs(dsolR(1)-1)>tol) then a:=a-((a+0.01)-a)*(dsolr(1)-1)/(dsolR(1)-dsolr(1))
end if;
> if (abs(dsolR(1)-1)<=tol) then i:=6 end if;
end do;
>dsol(1);

```

Mooney-Rivlin

```
> restart; with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

Warning, the name changecoords has been redefined

```
> a:=1.5: #chute de U(0.000001)#
```

```
> tol:=0.001:
```

```
> dU:=((r'*F)+(z'*G))/lambda[1]^2:
```

```
> dV:=((z'*F)-(r'*G))/lambda[1]^2:
```

```
> dVol:=Pi*((r(rho))^2)*V(rho):
```

```
> F:=(lambda[1]/W[11])*((d(lambda[2])*((W[1]/lambda[2])-W[12])-
((r'/(r(rho)*lambda[1]))*(lambda[1]*W[1]-lambda[2]*W[2]))):
```

```
> G:=(lambda[1]^2/W[1])*(((lambda[1]*lambda[2]*P)/H)
```

```
+((z'/(r(rho)*lambda[1]))*lambda[2]*W[2])):
```

```
> W:=C[1]*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-
```

```
3)+C[2]*(1/(lambda[1]^2)+1/(lambda[2]^2)+1/(lambda[3]^2)-3):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> W[1]:=diff(W,lambda[1]):
```

```
> W[11]:=diff(W[1],lambda[1]):
```

```
> W[12]:=diff(W[1],lambda[2]):
```

```
> Wi:=C[1]*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-
```

```
3)+C[2]*(1/(lambda[1]^2)+1/(lambda[2]^2)+1/(lambda[3]^2)-3):
```

```
> W[2]:=diff(Wi,lambda[2]):
```

```
> W[22]:=diff(W[2],lambda[2]):
```

```
> W[21]:=diff(W[2],lambda[1]):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> lambda[1]:=sqrt((diff(r(rho),rho)^2)+(diff(z(rho),rho)^2)):
```

```
> lambda[2]:=r(rho)/rho:
```

Derivadas de lambda[2] , r , z em relação a rho

```
> d(lambda[2]):=diff(lambda[2],rho):
```

```
> r':=diff(r(rho),rho):
```

```
> z':=diff(z(rho),rho):
```

```

> H:=0.003675:C[1]:=0.148964:C[2]:=0.013929:P:=0.002112:
> for i from 1 to 5 do
>
dsys:={diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a,z
(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0}:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolr:=subs(dsol,r(rho)):
dsolz:=subs(dsol,z(rho)):
> if (abs(dsolr(1)-1)>tol) then
dsys:={diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a+
0.01,z(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0} end if:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolR:=subs(dsol,r(rho)):
dsolZ:=subs(dsol,z(rho)):
r(1)=dsolR(1);
> if (abs(dsolR(1)-1)>tol) then a:=a-((a+0.01)-a)*(dsolr(1)-1)/(dsolR(1)-dsolr(1))
end if;
> if (abs(dsolR(1)-1)<=tol) then i:=6 end if;
end do;
>dsol(1);

```


Ogden

```
> restart; with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

Warning, the name changecoords has been redefined

```
> a:=1.5: #chute de U(0.000001)#
```

```
> tol:=0.001:
```

```
> dU:=((r'*F)+(z'*G))/lambda[1]^2:
```

```
> dV:=((z'*F)-(r'*G))/lambda[1]^2:
```

```
> dVol:=Pi*((r(rho))^2)*V(rho):
```

```
> F:=(lambda[1]/W[11])*((d(lambda[2])*((W[1]/lambda[2])-W[12])-
```

```
((r'/(r(rho)*lambda[1]))*(lambda[1]*W[1]-lambda[2]*W[2]))):
```

```
> G:=(lambda[1]^2/W[1])*(((lambda[1]*lambda[2]*P)/H)+((z'/(r(rho)
*lambda[1]))*lambda[2]*W[2])):
```

```
> W:=(2*mu[1]/alpha[1]^2)*((lambda[1]^alpha[1])+(lambda[2]^alpha[1])+
(lambda[3]^alpha[1])-3)+(2*mu[2]/alpha[2]^2)*
```

```
((lambda[1]^alpha[2])+(lambda[2]^alpha[2])+(lambda[3]^alpha[2])-3):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> W[1]:=diff(W,lambda[1]):
```

```
> W[11]:=diff(W[1],lambda[1]):
```

```
> W[12]:=diff(W[1],lambda[2]):
```

```
> Wi:=(2*mu[1]/alpha[1]^2)*((lambda[1]^alpha[1])+(lambda[2]^alpha[1])
```

```
+(lambda[3]^alpha[1])-3)+(2*mu[2]/alpha[2]^2)*((lambda[1]^alpha[2])+
(lambda[2]^alpha[2])+(lambda[3]^alpha[2])-3):
```

```
> W[2]:=diff(Wi,lambda[2]):
```

```
> W[22]:=diff(W[2],lambda[2]):
```

```
> W[21]:=diff(W[2],lambda[1]):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> lambda[1]:=sqrt((diff(r(rho),rho)^2)+(diff(z(rho),rho)^2)):
```

```
> lambda[2]:=r(rho)/rho:
```

Derivadas de lambda[2] , r , z em relação a rho

```
> d(lambda[2]):=diff(lambda[2],rho):
```

```

> r':=diff(r(rho),rho):
> z':=diff(z(rho),rho):
> H:=0.003675:P:=0.002521116:
> mu[1]:=-1.2:alpha[1]:=3:
> mu[2]:=1.5:alpha[2]:=5:
> for i from 1 to 5 do
>
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a,z
(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0}:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolr:=subs(dsol,r(rho)):
dsolz:=subs(dsol,z(rho)):
> if (abs(dsolr(1)-1)>tol) then
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a+
0.01,z(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0} end if:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolR:=subs(dsol,r(rho)):
dsolZ:=subs(dsol,z(rho)):
r(1)=dsolR(1);
> if (abs(dsolR(1)-1)>tol) then a:=a-((a+0.01)-a)*(dsolr(1)-1)/(dsolR(1)-dsolr(1))
end if;
> if (abs(dsolR(1)-1)<=tol) then i:=6 end if;
end do;
>dsol(1);

```

Delfino – Couro Cabeludo

```
> restart;with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

Warning, the name changecoords has been redefined

```
> a:=1.5: #chute de U(0.000001)#
```

```
> tol:=0.001:
```

```
> dU:=((r'*F)+(z'*G))/lambda[1]^2:
```

```
> dV:=((z'*F)-(r'*G))/lambda[1]^2:
```

```
> dVol:=Pi*((r(rho))^2)*V(rho):
```

```
> F:=(lambda[1]/W[11])*((d(lambda[2])*((W[1]/lambda[2])-W[12])-
((r'/(r(rho)*lambda[1]))*(lambda[1]*W[1]-lambda[2]*W[2])))):
```

```
> G:=(lambda[1]^2/W[1])*(((lambda[1]*lambda[2]*P)/H)+
((z'/(r(rho)*lambda[1]))*lambda[2]*W[2])):
```

```
> W:= a/b*(exp(b/2*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-3))-1):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> W[1]:=diff(W,lambda[1]):
```

```
> W[11]:=diff(W[1],lambda[1]):
```

```
> W[12]:=diff(W[1],lambda[2]):
```

```
> Wi:= a/b*(exp(b/2*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-3))-1):
```

```
> W[2]:=diff(Wi,lambda[2]):
```

```
> W[22]:=diff(W[2],lambda[2]):
```

```
> W[21]:=diff(W[2],lambda[1]):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> lambda[1]:=sqrt((diff(r(rho),rho)^2)+(diff(z(rho),rho)^2)):
```

```
> lambda[2]:=r(rho)/rho:
```

Derivadas de lambda[2] , r , z em relação a rho

```
> d(lambda[2]):=diff(lambda[2],rho):
```

```
> r':=diff(r(rho),rho):
```

```
> z':=diff(z(rho),rho):
```

```
> H:=0.074:P:=0.2:a:=0.213:b:=31.5:
```

```
> for i from 1 to 5 do
```

```

>
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a,z
(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0}:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolr:=subs(dsol,r(rho)):
dsolz:=subs(dsol,z(rho)):
> if (abs(dsolr(1)-1)>tol) then
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a+
0.01,z(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0} end if:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolR:=subs(dsol,r(rho)):
dsolZ:=subs(dsol,z(rho)):
r(1)=dsolR(1);
> if (abs(dsolR(1)-1)>tol) then a:=a-((a+0.01)-a)*(dsolr(1)-1)/(dsolR(1)-dsolr(1))
end if;
> if (abs(dsolR(1)-1)<=tol) then i:=6 end if;
end do;
>dsol(1);

```

Delfino – Couro Cabeludo – base elástica

```
> restart; with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

Warning, the name changecoords has been redefined

```
> a:=1.5: #chute de U(0.000001)#
```

```
> tol:=0.001:
```

```
> dU:=((r'*F)+(z'*G))/lambda[1]^2:
```

```
> dV:=((z'*F)-(r'*G))/lambda[1]^2:
```

```
> dVol:=Pi*((r(rho))^2)*V(rho):
```

```
> F:=(lambda[1]/W[11])*((d(lambda[2])*((W[1]/lambda[2])-W[12])-
((r'/(r(rho)*lambda[1]))*(lambda[1]*W[1]-lambda[2]*W[2]))):
```

```
> G:=(lambda[1]^2/W[1])*(((lambda[1]*lambda[2]*(P-k*I*z(rho)*
(r')/lambda[1]))/H)+((z'/(r(rho)*lambda[1]))*lambda[2]*W[2])):
```

```
> W:= a/b*(exp(b/2*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-3))-1):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> W[1]:=diff(W,lambda[1]):
```

```
> W[11]:=diff(W[1],lambda[1]):
```

```
> W[12]:=diff(W[1],lambda[2]):
```

```
> Wi:= a/b*(exp(b/2*((lambda[1]^2)+(lambda[2]^2)+(lambda[3]^2)-3))-1):
```

```
> W[2]:=diff(Wi,lambda[2]):
```

```
> W[22]:=diff(W[2],lambda[2]):
```

```
> W[21]:=diff(W[2],lambda[1]):
```

```
> lambda[3]:=1/(lambda[1]*lambda[2]):
```

```
> lambda[1]:=sqrt((diff(r(rho),rho)^2)+(diff(z(rho),rho)^2)):
```

```
> lambda[2]:=r(rho)/rho:
```

Derivadas de lambda[2] , r , z em relação a rho

```
> d(lambda[2]):=diff(lambda[2],rho):
```

```
> r':=diff(r(rho),rho):
```

```
> z':=diff(z(rho),rho):
```

```
> H:=0.074:P:=0.2:a:=0.213:b:=31.5:
```

```
> l:= 5:
```

```

> for i from 1 to 5 do
>
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a,z
(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0}:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolr:=subs(dsol,r(rho)):
dsolz:=subs(dsol,z(rho)):
> if (abs(dsolr(1)-1)>tol) then
dsys:={ diff(r(rho),rho)=U(rho),diff(U(rho),rho)=dU,diff(z(rho),rho)=V(rho),diff(
V(rho),rho)=dV,diff(Vol(rho),rho)=dVol,r(0.000001)=0.000001,U(0.000001)=a+
0.01,z(0.000001)=0,V(0.000001)=0,Vol(0.000001)=0} end if:
> dsol:=dsolve(dsys,numeric,output=listprocedure,range=0..1):
> dsolR:=subs(dsol,r(rho)):
dsolZ:=subs(dsol,z(rho)):
r(1)=dsolR(1);
> if (abs(dsolR(1)-1)>tol) then a:=a-((a+0.01)-a)*(dsolr(1)-1)/(dsolR(1)-dsolr(1))
end if;
> if (abs(dsolR(1)-1)<=tol) then i:=6 end if;
end do;
>dsol(1);

```