

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA
FONSECA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

BRUNO MELLO SILVEIRA

**DESENVOLVIMENTO DE CÓDIGO NUMÉRICO PARA ANÁLISE DE
VISCOELASTICIDADE EM TECIDOS MOLES**

TRABALHO DE CONCLUSÃO DE CURSO

**RIO DE JANEIRO
2021**

BRUNO MELLO SILVEIRA

**DESENVOLVIMENTO DE CÓDIGO NUMÉRICO PARA ANÁLISE DE
VISCOELASTICIDADE EM TECIDOS MOLES**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Engenharia Mecânica, do Departamento de Engenharia Mecânica, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca.

Orientador: Prof. Dr. Paulo Pedro Kenedi

RIO DE JANEIRO

2021

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

S587 Silveira, Bruno Mello
Desenvolvimento de código numérico para análise de viscoelasticidade em tecidos moles / Bruno Mello Silveira – 2021.
74f.: il. color. + anexo, enc.

Projeto Final (Graduação) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2021.

Bibliografia: f. 69-74

Orientador: Paulo Pedro Kenedi

1. Engenharia mecânica. 2. Viscoelasticidade – Cálculos numéricos. 3. Viscoelasticidade – Métodos de simulação.
I. Kenedi, Paulo Pedro (Orient.). II. Título.

CDD 620.1

AGRADECIMENTOS

Certamente neste parágrafo não será possível citar todas as pessoas que me auxiliaram em minha trajetória. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas palavras, mas elas podem ter certeza de que fazem parte do meu pensamento e de minha gratidão.

Por ser uma pessoa religiosa, agradeço primeiramente a Deus por onde consegui chegar.

Agraço à minha família que esteve comigo e me incentivou até em momentos de dificuldades, em especial ao meu pai por ter me dado condições de me dedicar a este trabalho, à minha mãe que me auxiliou a escrever este trabalho com mais clareza e objetividade e que junto de minha namorada Larissa e minha irmã Cíntia, me ajudaram grandemente em momentos de preocupação e estresse.

Agradeço ao meu professor e orientador Prof. Dr. Paulo Pedro Kenedi, pela sabedoria e paciência com que me guiou com exímio em minha trajetória acadêmica.

Agradeço a meus amigos que sempre me motivaram e compreenderam os momentos em que não pude estar junto com eles, deixam um agradecimento em especial a meus amigos Guilherme Monteiro, Agatha Dantas, Lucas Araújo e Tiago Frias que sempre demonstraram grande felicidade por minhas conquistas e apoio para alcançar mais.

Aos meus amigos que são membros e ex-membros da equipe Alpha de Fórmula SAE e Mud Runner Baja SAE, por terem me proporcionado um excelente ambiente em que pude me desenvolver.

À Stone, em especial os meus colegas de trabalho do setor de Clearing, por terem me propiciado um ótimo ambiente de trabalho em que pude evoluir e por me auxiliarem no desenvolvimento dos códigos utilizados neste trabalho.

E, por fim, ao professor Alexandre Silva Lima (in memorian), por ter me auxiliado em minha trajetória na equipe Alpha de Fórmula SAE, sendo um excelente orientador, coordenador e professor.

RESUMO

SILVEIRA, Bruno M. **Desenvolvimento de código numérico para análise de viscoelasticidade em tecidos moles.** 2021. 81f. Trabalho de Conclusão de Curso – Centro Federal de Educação Tecnológica Celso Suckow da Fonseca. Rio de Janeiro. Rio de Janeiro, 2021.

Para se calcular a evolução das tensões em tecidos moles, faz-se necessário utilizar equações que não possuem soluções analíticas fechadas, sendo, portanto, fundamental o uso de métodos numéricos para conseguir resolvê-las. Este trabalho busca desenvolver um código numérico nos moldes de uma API, sendo capaz de realizar cálculos baseando-se nos modelos de viscoelasticidade linear de Maxwell, quase-linear de Fung, e não-linear de Schapery gerando ao fim, um arquivo em formato CSV com os resultados para comparar com os dados experimentais. Esses dados são obtidos a partir de ligamentos de joelho de porco como: ligamento colateral lateral, ligamento cruzado anterior, ligamento cruzado posterior e ligamento colateral medial.

Palavras-chave: viscoelasticidade, ligamento de joelho, código numérico, API.

ABSTRACT

SILVEIRA, Bruno M. **Numerical code development for viscoelasticity analysis in soft tissue**. 2021. 81f. Trabalho de Conclusão de Curso - Federal Center of Technological Education – Rio de Janeiro. Rio de Janeiro, 2021.

To calculate the stress evolution in soft tissues, it is necessary to use equations that do not have closed analytical solutions, therefore, it is essential to use numerical methods to solve them. This work seeks to develop a numerical code based on API model, being able to perform calculations based on Maxwell's linear, Fung's quasi-linear and Schapery's non-linear viscoelastic models, creating, at the end, a file in CSV format with the results to compare with experimental data. These data are captured from porcine knee ligaments such as: lateral collateral ligament, anterior cruciate ligament, posterior cruciate ligament and medial collateral ligament.

Keywords: viscoelasticity, knee Ligament, numerical code, API.

LISTA DE ILUSTRAÇÕES

Figura 1 - Representação de um joelho humano	7
Figura 2 - Gráficos de tensão versus tempo e de deformação versus tempo em uma situação de: (a) fluência e (b) relaxação (COMPOSITE CONSTRUCTION, 2020) ...	11
Figura 3 - Exemplo de texto na formatação JSON	15
Figura 4 - Formas de passar o conjunto de dados da requisição no <i>Swagger</i>	16
Figura 5 - <i>Swagger</i> da API desenvolvida para este estudo (Stacker Overflow, 2018)	16
Figura 6 - Exemplo de código presente no <i>controller</i>	18
Figura 7 - Fluxograma básico das operações	19
Figura 8 - Representação de mola e amortecedor em série (Fung, 1993).....	24
Figura 9 - Tensão e deformação ao longo do tempo para (a) relaxação e para (b) fluência.....	27
Figura 10 - Modelo generalizado de Maxwell (BABAEI, <i>et al</i> , 2015) com nomenclatura adaptada.....	28
Figura 11 - Gráfico de deformação versus tempo utilizada nos ensaios	31
Figura 12 – Ajuste de curva durante relaxação	37
Figura 13 – Ajuste de curva durante tempo de rampa	38
Figura 14 – h_e e h_2 em função da deformação (BLANDFORD, 2017).	41
Figura 15 – Ajuste de curva para constantes C e n segundo (PROVENZANO <i>et al.</i> , 2002).	43
Figura 16 – Ajuste de curva para constantes C e n segundo (BLANDOFORD, 2017)	44
Figura 17 – (a) h_e e (b) h_2 em função da deformação segundo (BLANDOFORD, 2017).	45
Figura 18 – Ajuste de curva para determinar a constante $A(\varepsilon)$ usando (a) função linear e (b) função exponencial.	48
Figura 19 – Ajuste de curva para determinar a constante $B()$	49
Figura 20 – Fluxograma indicando a ordem de herança entre as classes modelo....	50
Figura 21 – Fluxograma para a sub-rotina “Calcula Resultados”	51
Figura 22 – Fluxograma das operações de (a) calcular os resultados do modelo e (b) fazer análise de sensibilidade das variáveis.....	52
Figura 23 – Gráfico de tensão versus tempo obtido experimentalmente para o LCA56	
Figura 24 – Gráfico de tensão versus tempo obtido numericamente com modelo de Fung desconsiderando o tempo de rampa comparando com curva experimental	58
Figura 25 – Gráfico de tensão versus tempo obtido numericamente com modelo de Fung considerando o tempo de rampa comparando com curva experimental	58
Figura 26 – Gráfico de tensão versus tempo para 3% de deformação usando constantes obtidos para 5% de deformação	59
Figura 27 – Gráfico de tensão versus tempo para Schapery desconsiderando o tempo de rampa e usando série de Prony (equação 50) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.	61

Figura 28 – Gráfico de tensão versus tempo para Schapery desconsiderando o tempo de rampa e usando lei da potência (equação 51) obtendo as constantes segundo (PROVENZANO, 2002) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.	62
Figura 29 – Gráfico de tensão versus tempo para Schapery desconsiderando o tempo de rampa e usando lei da potência (equação 51) obtendo as constantes segundo (BLANDFORD, 2017) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.	63
Figura 30 – Gráfico de tensão versus tempo para Schapery considerando o tempo de rampa e usando série de Prony (equação 50) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.	64
Figura 31 – Gráfico de tensão versus tempo para Schapery considerando o tempo de rampa e usando lei da potência (equação 51) obtendo as constantes segundo (PROVENZANO, 2002) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.	65
Figura 32 – Gráfico de tensão versus tempo para Schapery considerando o tempo de rampa e usando lei da potência (equação 51) obtendo as constantes segundo (BLANDFORD, 2017) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.	66

LISTA DE TABELAS

Tabela 1 - Status HTTP utilizados no código	17
Tabela 2 – Constantes de Fung para o ligamento colateral anterior (LCA).....	57
Tabela 3 – Constantes para Função Relaxação Transiente usando equação 51	60
Tabela 4 – Constantes para Função Relaxação Transiente usando equação 50	60
Tabela 5 – Valores de h_e e h_2 para cada nível de deformação	60

SUMÁRIO

1 INTRODUÇÃO	6
1.1 REVISÃO BIBLIOGRÁFICA.....	6
1.2 CONCEITOS FUNDAMENTAIS DE VISCOELASTICIDADE.....	10
1.3 CONCEITOS FUNDAMENTAIS DE PROGRAMAÇÃO	13
2 RECURSOS IMPLEMENTADOS	21
2.1 MÉTODOS NUMÉRICOS	21
2.2 DADOS EXPERIMENTAIS	23
2.3 OPERAÇÕES AUXILIARES	23
3 MODELO LINEAR DE VISCOLEASTICIDADE	24
3.1 MAXWELL	24
3.2 MAXWELL GENERALIZADO	27
4 MODELO QUASE-LINEAR DE VISCOELASTICIDADE	29
4.1 EQUACIONAMENTO.....	29
4.1.1 Deformação	30
4.1.2 Tensão.....	32
4.1.3 Resposta elástica (parcela elástica)	33
4.1.4 Função relaxação reduzida (parcela viscosa).....	34
4.2 OBTENÇÃO DAS CONSTANTES	37
5 MODELO NÃO-LINEAR DE VISCOELASTICIDADE	39
5.1 SCHAPERLY	39
5.1.1 Equacionamento	40
5.1.1.1 Deformação	40
5.1.1.2 Tensão.....	40
5.1.1.3 Função relaxação transiente.....	42
5.1.2 Obtenção das constantes	43
5.2 MÉTODO DE SUPERPOSIÇÃO NÃO-LINEAR	46
5.2.1 Equacionamento	46
5.2.1.1 Deformação	46
5.2.1.2 Tensão.....	46
5.2.1.3 Função Relaxação	47
5.2.2 Obtenção das variáveis	47
6 IMPLEMENTAÇÃO NUMÉRICA.....	50
6.1 MODELO LINEAR.....	54
6.2 MODELO QUASE-LINEAR.....	54
6.3 MODELO NÃO-LINEAR.....	55
6.3.1 Schapery.....	55
6.3.2 Método de superposição não-linear.....	55
7 ENSAIOS EXPERIMENTAIS	56

8 RESULTADOS E DISCUSSÕES	57
8.1 MODELO DE FUNG	57
8.2 MODELO DE SCHAPERY	59
9 TRABALHOS FUTUROS	68
REFERÊNCIAS.....	69
ANEXO A - Descrição das equações usadas para modelo de viscoelasticidade quase-linear	75

1 INTRODUÇÃO

O presente trabalho busca desenvolver um código numérico utilizando linguagem C# nos moldes de uma API capaz de realizar cálculos de viscoelasticidade aplicada a tecidos moles, baseando-se nos modelos linear de Maxwell, quase-linear de Fung e não-linear de Schapery, além de gerar um arquivo externo em formato CSV, para viabilizar a comparação dos resultados numéricos com os resultados experimentais obtidos de ensaios uniaxiais de relaxação. Busca-se com este projeto auxiliar a pesquisa sobre viscoelasticidade em tecidos moles desenvolvida pelo professor Paulo Pedro Kenedi, sendo apresentada os cálculos, as rotinas e as operações secundárias necessárias.

Neste trabalho as equações de viscoelasticidade são abordadas utilizando equações escalares, visto que, segundo (Fung, 1993) e (Schapery, 1969), para ensaios em que o carregamento é uniaxial, essa abordagem representa bem a realidade. Já em casos de carregamentos tridimensionais, faz-se necessário a utilização de tensores para descrever a deformação e, conseqüentemente, os demais parâmetros relacionados. Além disso, para os modelos quase-linear e não-linear, para o tempo de rampa, que representa o tempo em que a deformação está aumentando, são feitas duas ponderações: considerá-lo e desconsiderá-lo. Nos capítulos referentes a cada modelo, isso será mais bem discutido, já que as equações dos modelos são afetadas por estas escolhas.

1.1 REVISÃO BIBLIOGRÁFICA

No ramo da indústria civil, pode-se ver a aplicação da viscoelasticidade em modelos de fluência de estruturas mistas de aço, conforme apresentado por (TARECO, 2014) que usa os modelos lineares de Maxwell e de Kelvin para modelar uma estrutura de aço-concreto, analisando a relaxação e a fluência somente do concreto na resposta da estrutura mista e dos materiais que a compõem. Além disso, conforme apresentado (QUEIROZ, 2008) materiais viscoelásticos também são utilizados para atenuar vibrações e ruídos em estruturas, tendo aplicações tanto no setor automotivo quanto no setor aeroespacial. Com o objetivo de melhor compreender os modelos, (R. F. Navarro, 2017) aborda as combinações possíveis

entre as analogias mecânicas de mola e amortecedor que caracterizam o comportamento viscoelásticos de materiais reais de uso no ramo industrial.

A viscoelasticidade também é aplicada a outros materiais, como visto na obra de (WEINEROWSKA-BORDS, 2015), que analisou os aspectos relativos ao comportamento viscoelástico de paredes de tubos poliméricos durante um fluxo instável, também apresentou, para o modelo linear, a expressão que utiliza convolução do termo viscoelástico e comparou esta com a equação correspondente ao atrito instável.

A importância do estudo da viscoelasticidade não se restringe apenas à metais e polímeros, tendo também importâncias no estudo de tendões e ligamentos, conhecidos como tecidos moles, como mostrado por (ROSSETTO, 2009). Este reforça que esse conhecimento é de suma importância para melhor compreensão do comportamento dos tendões, permitindo que melhores análises possam ser feitas para determinar o treinamento físico, bem como nos casos de terapêuticos e das tendinopatias. (BERNARDES, *et. al*, 2005), buscou determinar os parâmetros biomecânicos para o modelamento da articulação do joelho humano, conforme mostrado esquematicamente na Fig.1, através de exercícios extensivos, junto de imagens obtidas por videofluoroscópio, onde a viscoelasticidade desempenha um importante papel. (ZHENG, *et al.*, 1998) buscou modelar as forças internas em um joelho humano, analiticamente, durante um exercício físico. Além disso, foi utilizado um software para otimizar os resultados e minimizar os erros.

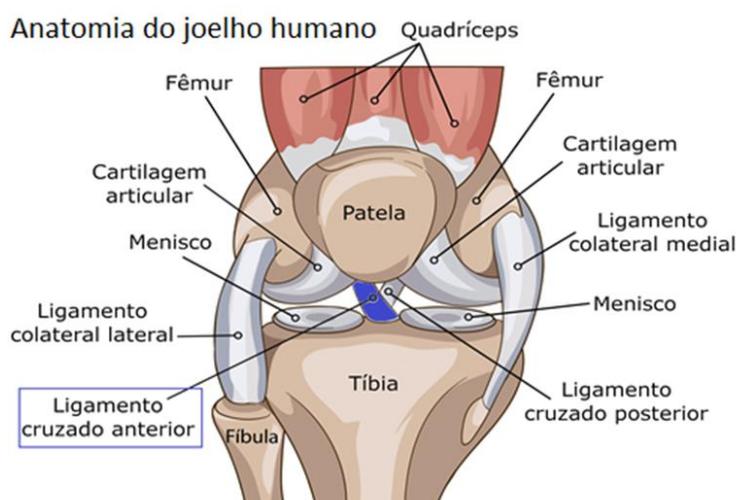


Figura 1 - Representação de um joelho humano

Devido a sua simplicidade, algumas pesquisas que foram feitas com o objetivo de caracterizar as propriedades mecânicas de tecidos moles, utilizaram modelos lineares de viscoelasticidade. Isto foi feito em (JAMISON, *et al.*, 1968) em que e foi apresentado um estudo caracterizando os tecidos moles via viscoelasticidade linear. Verificou-se que o modelo teórico não era compatível com a investigação experimental dos autores com pele de porco. Na obra de (MILLER, 1998), também é usado o modelo linear com o objetivo de aplicar em um software de elementos finitos para caracterizar as propriedades mecânicas de tecidos de cérebro.

Como os modelos lineares são mais simples, mas não conseguem reproduzir o comportamento real de muitos materiais viscoelásticos, alguns autores preferiram utilizar o modelo de viscoelasticidade quase-linear apresentado por (Fung, 1993) ou mesmo modelos não-lineares mais complexos, serem mais próximos da realidade. Em (PIAZZA, *et al.*, 2001) desenvolve-se um modelo dinâmico tridimensional da articulação tibio-femoral e patelo-femoral para prever o movimento de implantes no joelho durante atividades intensas baseando-se no modelo de Fung. Neste modelo, utilizou-se as equações dinâmicas de movimento sujeitas às forças geradas pelos músculos, ligamentos e contato nas articulações tibio-femoral e patelo-femoral. (DEBSKI, *et al.*, 2004) utilizou-se do modelo quase-linear para obter uma função de relaxamento que melhor se adequasse aos tecidos moles.

(XU e ENGQUIST, 2018) propuseram um modelo matemático para o módulo de relaxação e sua solução numérica, baseando-se em modelos não-lineares de viscoelasticidade. Este modelo apresentou-se capaz de simular fluência, deformação sinusoidal e dissipação de energia. Apresentou também melhor precisão no ajuste de dados experimentais e estabilidade de predição fora da faixa experimental quando comparado a outros modelos amplamente utilizados, como o modelo de Maxwell generalizado.

Com o objetivo de analisar o comportamento viscoelástico de tendões ao longo do tempo, diversos autores fizeram variados experimentos de modo a observar o comportamento tanto da tensão e quanto da deformação. (BONIFASI-LISTA, *et al.*, 2004) quantificaram a deformação do ligamento colateral medial (LCM) de um ser humano aplicando esforços nas direções longitudinais e transversais, e cisalhamento ao longo da direção das fibras. Também verificaram se o comportamento viscoelástico depende da direção de aplicação. (DUENWALD, *et al.*, 2009) utilizaram o LCM

impondo deformações de 1, 2, 3, 4, 5 e 6% com o objetivo de explorar a dependência com o tempo da recuperação deste ligamento. Foi observado que o relaxamento encontrado contrasta com o modelo de viscoelasticidade quase-linear (QLV), que prevê dependência com o tempo igual para várias deformações iniciais impostas, e que os ligamentos não se recuperaram até os níveis previstos pelos modelos não linear e quase-linear, embora tenham tido recuperação parcial.

A importância da utilização de recursos computacionais para a implementação de modelos de viscoelasticidade pode ser vista em diversos artigos, já que faz-se necessário executar cálculos complexos que, em alguns casos, que não possuem soluções analíticas. Em (DE BARROS, 2020), é abordado os conceitos primordiais, com revisão bibliográfica detalhada, para implementar matematicamente a abordagem viscoelástica em tecidos moles. (WEISS *et al*, 2001), fizeram uma revisão sobre técnicas antigas e atuais para modelar computacionalmente ligamentos e tendões, mostrando conceitos relevantes sobre mecânica do contínuo e elementos finitos, além de dar ênfase na influência microestrutural dos tecidos moles. (ABRAMOWITCH *et al.*, 2004) buscaram obter as cinco constantes para o modelo teórico quase-linear de viscoelasticidade: os que são usados para descrever a resposta elástica (constantes A e B) e os referentes a função relaxação reduzida (constantes C , τ_1 e τ_2). Submeteram seis ligamentos ligamento colateral lateral (LCL) de cabra a testes de tensão uniaxial com tempo de rampa na faixa de 18 segundos seguido de uma hora de relaxação, por fim, usaram as equações constitutivas do modelo QLV com convolução para os ajustes de curva. Nos testes realizados, a convergência falhou para três dos seis ligamentos, com os maiores erros nas constantes A , B e τ_1 . (DUENWALD, JR e LAKES, 2009) utilizaram métodos numéricos para expandir a janela experimental de tempo para o relaxamento e fluência. Mostraram as vantagens associadas a isso, também apresentaram um estudo sobre o papel do tempo de rampa, relaxação e recuperação no contexto das formulações para os modelos de viscoelasticidade quase-linear e não-linear. (BABAEI *et al.*, 2015) desenvolveram um algoritmo eficiente que permite a identificação de espectros de relaxação viscoelástica, sendo testado com a utilização de dados experimentais para caracterizar a sua robustez, identificar suas limitações e pontos fortes. (DE PASCALIS *et al.*, 2014) reavaliaram o modelo de Fung para viscoelasticidade quase-linear, expondo uma abordagem diferente que produz um

melhor comportamento e oferece um esquema simples para resolver uma ampla gama de modelos por meio de soluções numéricas para a equação de Volterra. Alertaram, ainda, que uma série de características negativas encontradas em outras obras, que são atribuídas ao modelo de Fung, são na verdade uma consequência da forma como o modelo foi aplicado.

Conforme será apresentado nos próximos capítulos, o modelo de Fung tem algumas limitações, fazendo com que diversos autores utilizem modelos de viscoelasticidade mais complexos e com uma ótica não linear, como o modelo de Schapery (1969, 1997, 2000). (HAJ-ALI e MULIANA, 2002) utilizaram modelos não-lineares de viscoelasticidade juntamente a uma modelagem micromecânica tridimensional para descrever o comportamento de alguns polímeros. (PROVENZANO *et al.*, 2002) analisaram se o modelo não-linear de Schapery, ou o método de superposição modificado, poderiam modelar adequadamente o comportamento de tensão dos ligamentos submetidos a relaxação, concluindo que ambos os modelos atingem este objetivo. (BLANDFORD, 2017) utilizou os modelos de Schapery e de superposição não linear para modelar o comportamento de ligamentos LCM de seres humanos submetidos a ensaio de relaxação. Obteve-se resultados satisfatórios para ambos os modelos. (RAMO *et al.* 2018) desenvolveram uma abordagem de integração numérica dependente da deformação, que permite o cálculo da tensão atual, utilizando o valor de tensão da etapa de tempo anterior, o que melhorou a eficiência de ajuste e a tratabilidade computacional. Validou-se esta metodologia comparando com ensaios de relaxação com seis níveis de deformação e ensaios dinâmicos com três frequências. Para essa abordagem, obtiveram resultados satisfatórios com erros menores do que 0,3%.

1.2 CONCEITOS FUNDAMENTAIS DE VISCOELASTICIDADE

Entende-se como viscoelasticidade a propriedade dos materiais que apresentam comportamento viscoso e elástico ao mesmo tempo, sendo um conceito amplamente utilizado em diversos setores da indústria. Elasticidade, por exemplo, é a propriedade de um corpo de sofrer deformação quando submetido a carga e retornar ao seu estado de equilíbrio inicial ao cessar a carga. Por outro lado, viscosidade é a resistência de um fluido quanto ao escoamento, causando uma velocidade de fluxo

maior quanto mais afastado da superfície de contato. Atualmente há diversos modelos que descrevem o comportamento viscoelástico de diferentes materiais, podendo serem divididos, de forma didática, em linear, quase-linear e não-linear.

Alguns conceitos de viscoelasticidade são abordados, como a fluência e a relaxação. A fluência (*creep*, em inglês) ocorre quando aplicada uma tensão constante ao corpo (por exemplo através da aplicação de um peso morto) e observa-se uma contínua deformação ao longo do tempo, conforme pode ser visto na Fig. 2.a. Por outro lado, a relaxação (*relaxation*, em inglês) ocorre quando aplicada uma deformação constante ao corpo e observa-se a tensão diminuir ao longo do tempo até um determinado patamar e se mantendo constante depois, conforme mostrado na Fig 2.b.

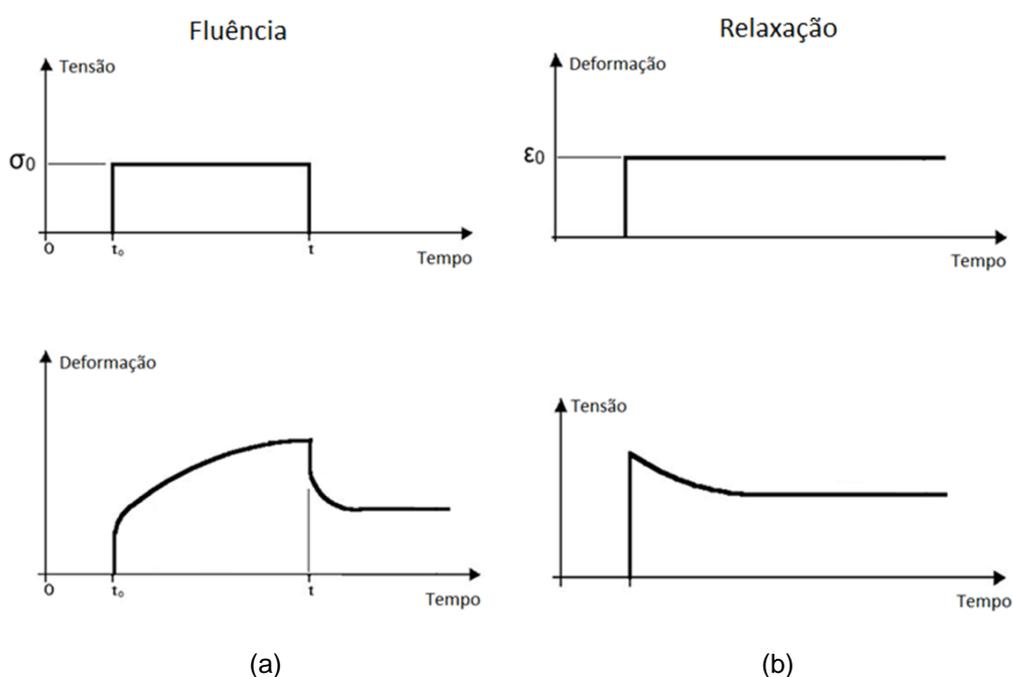


Figura 2 - Gráficos de tensão versus tempo e de deformação versus tempo em uma situação de: (a) fluência e (b) relaxação (COMPOSITE CONSTRUCTION, 2020)

O comportamento linear da deformação no caso de fluência pode ser expresso de acordo com a equação 1, que indica que a tensão é constante e a função fluência, expressa por $J(t)$, que depende somente do tempo.

$$\varepsilon(t) = \sigma_0 \cdot J(t) \quad (1)$$

Já para a tensão, seu comportamento linear no caso de relaxação pode ser expresso de acordo com a equação 2, mostrando que a deformação atua de maneira constante e a função relaxação, expressa por $G(t)$, que depende somente do tempo.

$$\sigma(t) = \varepsilon_0 \cdot G(t) \quad (2)$$

Outro conceito importante é o princípio da superposição de Boltzmann que é utilizado como base para os modelos de viscoelasticidade quase-linear e não-lineares. Segundo este princípio, para o caso unidimensional é considerado uma força $F(t)$ e um alongamento $u(t)$, que é causada por todo o histórico do carregamento até o tempo t . Se a função $F(t)$ for contínua e diferenciável, então, em um pequeno intervalo de tempo $d\tau$ no tempo t , o incremento do carregamento pode ser expresso por $(dF/d\tau)d\tau$. Este incremento continua atuando no material e contribui com $du(t)$ para o alongamento no tempo t , com uma constante de proporcionalidade c , também conhecida como função fluência, dependente do intervalo de tempo $t - \tau$. Isto está descrito na equação 3.

$$du(t) = c(t - \tau) \frac{dF(\tau)}{d\tau} d\tau \quad (3)$$

Algo similar pode ser escrito relacionando o comportamento do carregamento a partir da ação de um alongamento, conforme apresentado na equação 4, contudo, para esta, será usado uma constante de proporcionalidade k , também conhecida como função relaxação.

$$dF(t) = k(t - \tau) \frac{du(\tau)}{d\tau} d\tau \quad (4)$$

Essas formulações também podem ser aplicadas para se obter uma relação entre a deformação e a tensão, já que isto possui uma importância maior para os estudos de viscoelasticidade em tecidos moles, sendo estas descritas a seguir.

Contudo, vale lembrar que as constantes de proporcionalidades são diferentes, sendo estas J e G que também representam as equações para, respectivamente, a fluência e a relaxação, porém neste caso relacionando deformação e tensão.

$$d\varepsilon(t) = J(t - \tau) \frac{d\sigma(\tau)}{d\tau} d\tau \quad (5)$$

$$d\sigma(t) = G(t - \tau) \frac{d\varepsilon(\tau)}{d\tau} d\tau \quad (6)$$

De acordo com o princípio de superposição de Boltzmann, para o cálculo do efeito total do carregamento na estrutura analisada pode-se somar cada diferencial, aplicando uma integral do tempo inicial igual a 0 até o tempo t , conforme descrito em seguida.

$$\varepsilon(t) = \int_0^t J(t - \tau) \frac{d\sigma(\tau)}{d\tau} d\tau \quad (7)$$

$$\sigma(t) = \int_0^t G(t - \tau) \frac{d\varepsilon(\tau)}{d\tau} d\tau \quad (8)$$

Essas equações são lineares, já que estas dependem somente do tempo, entretanto, também são vistas utilizadas de maneira adaptada em modelos quase-lineares e não-lineares. Isso é também apresentado em (FUNG, 1993).

1.3 CONCEITOS FUNDAMENTAIS DE PROGRAMAÇÃO

Neste item, diversos conceitos da área de ciência da computação serão esclarecidos. A linguagem de programação escolhida, a linguagem C# que, segundo (Microsoft, 2021), “é uma linguagem de programação moderna, orientada a objeto e de tipo seguro”. Esta linguagem de programação é comumente utilizada no desenvolvimento de sistemas por empresas por seu poder computacional, além de possuir uma vasta documentação ofertada gratuitamente pela *Microsoft*.

O próximo conceito é a API, sigla para *Application Programming Interface*, que reúne diversos conceitos. Segundo a Microsoft, a API (Microsoft, 2019) “especifica como os componentes e sistemas de software devem interagir uns com os outros”. Segundo (TechTudo, 2020), corresponde a um conjunto de normas que possibilitam a comunicação entre sistemas através de uma série de padrões e protocolos. O padrão SOLID, expõe princípios de boas práticas de programação orientada a objeto (POO) de modo a tornar o código limpo, de fácil manutenção e escalável. O protocolo HTTP, determina a estrutura básica de comunicação a ser usada por aplicações WEB. Vale salientar que esse padrão e esse protocolo foram utilizados no desenvolvimento do código para este trabalho, com o objetivo de criar um código com grande poder computacional, de fácil compreensão e manutenibilidade.

Por utilizar os protocolos HTTP, pode-se chamar a aplicação desenvolvida para este estudo de API RESTful. De acordo com (HOSTGATOR, 2019 e REDHAT, 2021) A sigla REST significa *Representational State Transfer*, que traduzido para o português significa Transferência Representacional de Estado, e, por sua utilização por completo na API, recebe o sufixo “ful”. Devido a isso, foram utilizados alguns recursos essenciais para o correto funcionamento da API RESTful, como entrada e saída de dados usando formatação JSON, chamadas HTTP através de *endpoint*, que possuem verbos HTTP específicos para cada tipo de tarefa, status HTTP na resposta da operação e estrutura de criação de URLs estáticas, sendo escolhido o *framework* ASP.NET MVC. Como diversos conceitos são utilizados neste contexto, é necessária uma melhor explicação sobre cada um deles.

JSON (INTRODUCING, 2021), sigla para *JavaScript Object Notation* ou, em português, Notação de Objetos JavaScript, é uma formatação leve de troca de dados, simples de ler e escrever para o ser humano e de fácil interpretação e geração para máquinas, é independente de linguagem, por usar convenções que são familiares a diversas linguagens de programação, conforme pode ser visto na Fig. 3. No presente trabalho, será amplamente utilizada na serialização de objetos, isto é, converter uma cadeia de texto em um objeto utilizado internamente pelo código, e na desserialização de objetos, que é o caminho inverso da serialização. Ambos são usados, principalmente, pelo *Swagger* na entrada e saída de dados, respectivamente, conceito que será melhor apresentado adiante.

```
{
  "timeStep": 0.1,
  "finalTime": 600,
  "data": [
    {
      "softTissueType": "Numerial estimation",
      "viscoelasticConsideration": 1,
      "numOfRelaxations": 2,
      "strainRate": 0.001,
      "strainDecreaseRate": 0.001,
      "maximumStrain": 0.02,
      "minimumStrain": 0.01,
      "timeWithConstantMaximumStrain": 100,
      "timeWithConstantMinimumStrain": 100,
      "elasticStressConstant": 1000,
      "elasticPowerConstant": 0.1,
      "reducedRelaxationFunctionData": {
        "relaxationIndex": 0.05,
        "fastRelaxationTime": 0.005,
        "slowRelaxationTime": 50
      }
    }
  ]
}
```

Figura 3 - Exemplo de texto na formatação JSON

O próximo conceito a ser descrito é o *endpoint* (MILVUS, 2020) e (HOSTGATOR, 2019). Esse é um termo em inglês que pode ser traduzido para pontos de extremidade, que, no caso abordado neste documento, possui de fato esta característica, sendo o ponto de extremidade da aplicação. Nesse contexto, o *endpoint* corresponderá a uma URL por onde são feitas as requisições possibilitando que as operações em uma API sejam acessadas. Para efetuar essa requisição, é necessário definir dois parâmetros: o verbo HTTP definido pelo programador, que pode ser GET, para uma busca de dados, POST, para criar um dado, PUT, para atualizar algum dado, e DELETE, para excluir um dado; e o conjunto de dados da requisição passado pelo usuário. Este último pode ser preenchido de três maneiras diferentes, como pode ser visto na Fig. 4: na URL, escrita juntamente à rota do endpoint; por *query*, em que no *Swagger* permite escrever cada parâmetro separado; e no corpo da requisição, usando formatação JSON. Vale salientar que um *endpoint* está atrelado a uma operação que é uma classe que possui as lógicas para a execução de uma determinada rotina e/ou tarefa.



Figura 4 - Formas de passar o conjunto de dados da requisição no *Swagger*

O próximo conceito é o *Swagger* em que segundo (SMARTBEAR, 2021) é uma ferramenta *open source* para desenvolvimento de APIs, facilitando no design, construção, documentação, teste e padronização, além disso permite ao usuário acessar os *endpoints* disponíveis através de uma interface simples e de fácil compreensão com documentações, exemplos e explicações sobre os recursos, conforme pode ser visto na Fig. 5.

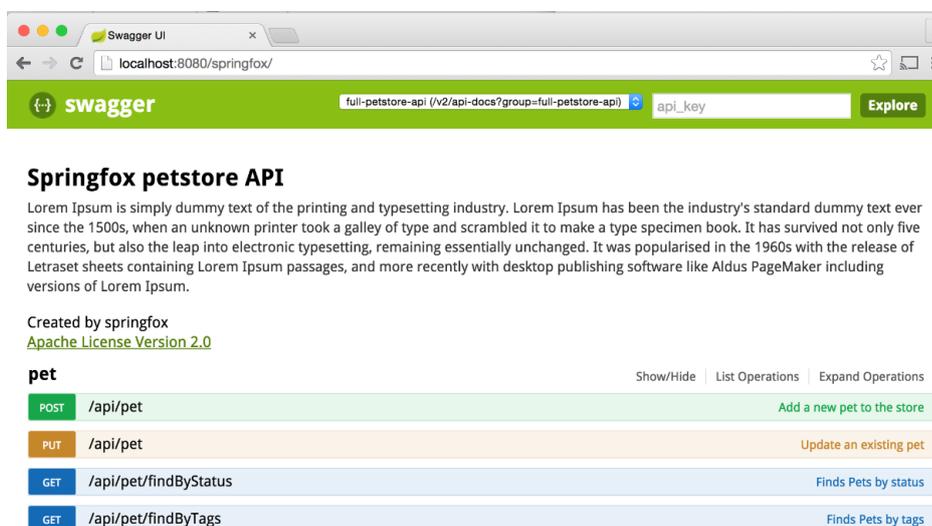


Figura 5 - *Swagger* da API desenvolvida para este estudo (Stacker Overflow, 2018)

Além disso, este recurso permite expor ao usuário a resposta detalhada da operação, que nada mais é do que uma cópia da estrutura da classe cuja responsabilidade única é conter essas informações. No caso deste estudo, todas as classes de resposta possuem a mesma estrutura com as seguintes propriedades: *Success*, que é do tipo booleano, ou seja, pode ser verdadeiro, quando a operação executar corretamente, e falso, caso falhe; *HttpStatusCode*, que contém o status HTTP da operação; *Errors*, que é uma lista onde são escritos os erros que ocorrem durante a aplicação, juntamente a um código pré-definido para facilitar a compreensão do motivo do erro; e *Data*, que contém os dados principais da resposta, variando através da necessidade de cada operação, já que em cada caso há informações diferentes que são importantes de retornar para o usuário.

Devido a sua grande gama de informações, vale informar que somente alguns status HTTP foram utilizados neste estudo, já que este pode informar diversos acontecimentos com a API, conforme mostrado pela empresa Runscope em seu site dedicado a isso. Os valores utilizados são descritos na Tabela 1 com sua nomenclatura, significado e dados de resposta obrigatórios.

Tabela 1 - Status HTTP utilizados no código

Status HTTP	Nomenclatura	Significado	Dados de resposta obrigatórios
200	OK	Operação executada com sucesso	Não tem
201	Created	Arquivo gerado com sucesso	Caminho do arquivo
202	Accepted	Dados de requisição válidos e aceitos para executar a operação	Não tem
400	Bad Request	Dados de requisição inválidos	Motivo do erro descrevendo quais dados estão inválidos
401	Unauthorized	Usuário não tem autorização para acessar o <i>endpoint</i>	Não tem
404	Not found	Nenhum dado foi encontrado de acordo com o filtro passado	Não tem
500	Internal Server Error	Erro interno	Descrição detalhada do motivo do erro
501	Not Implemented	Recurso não implementado	Descrição do recurso que não foi implementado

Fonte: Runscope, 2021

Para que a aplicação possa expor na WEB seus recursos, faz-se necessário a utilização de alguma ferramenta que possibilita isso, sendo escolhida para este trabalho o *framework* ASP.NET MVC, que possibilita criar sites estáticos, sendo interpretado com facilidade pelo *Swagger*. Seguindo (Rick Anderson, 2019 e GASPAROTTO, 2014) ASP.NET é uma plataforma da Microsoft criada para desenvolvimento WEB e a sigla MVC vem de *Model View Controller*, sendo este conjunto responsável pela apresentação da aplicação, visando criar um código que não possua conexão forte entre as partes, pois facilita a manutenção e adição de funcionalidades. Dentro deste padrão, há três componentes principais: *model*, responsável por representar a entidade da lógica de negócio da aplicação, ou seja, a operação; *view*, responsável por apresentar uma interface para o usuário, que neste caso estará em conjunto com o *Swagger*; e *controller*, que é responsável pelo controle dos elementos, propiciando uma ligação entre eles. Um exemplo de código escrito no *controller* pode ser visto na Fig. 6, destacando-se em vermelho a parte que representa a operação e em azul a parte que representa os dados de entrada passados pelo usuário através do *Swagger*.

```

/// <summary>
/// It is responsible to analyze the experimental results.
/// </summary>
/// <param name="analyzeResults"></param>
/// <param name="request"></param>
/// <returns></returns>
/// <response code="201">Returns the newly created files.</response>
/// <response code="400">If some validation do not passed.</response>
/// <response code="500">If occurred some error in process.</response>
/// <response code="501">If some resource is not implemented.</response>
[ProducesResponseType(StatusCodes.Status201Created)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
[ProducesResponseType(StatusCodes.Status501NotImplemented)]
[HttpPost("analyze")]
0 references
public async Task<ActionResult<AnalyzeResultsResponse>> AnalyzeResults(
    [FromServices] IAnalyzeResults analyzeResults,
    [FromQuery] AnalyzeResultsRequest request)
{
    AnalyzeResultsResponse response = await analyzeResults.ProcessAsync(request).ConfigureAwait(false);
    return response.BuildHttpResponse();
}

```

Figura 6 - Exemplo de código presente no *controller*

Com o objetivo de simplificar e padronizar o código, foi desenvolvido um conjunto de classes que servem de base para todas as operações, de modo que cada operação da API só precisa implementar seus parâmetros e lógicas específicas, evitando repetições de código. Dessa forma, todas as operações seguem o mesmo

fluxograma apresentado na Fig. 7, em que, após a entrada dos dados a partir do *Swagger*, esses são validados no método *ValidateOperationAsync*, caso sejam válidos, segue para o método *ProcessOperationAsync*, onde a operação será de fato executada e, caso não sejam, encerra o processo retornando mensagens de erro e status HTTP 400. Além disso, foram criados contratos específicos que servem como base para os demais. Entende-se como contratos as classes que contêm os dados da requisição e a resposta e seus dados principais de uma determinada operação, sendo, para o presente trabalho, as classes com os sufixos *Request*, *Response* e *ResponseData*.

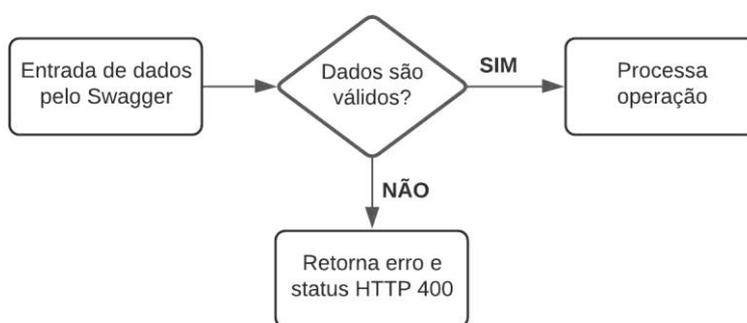


Figura 7 - Fluxograma básico das operações

Para tornar a aplicação otimizada e rápida, foi utilizado o conceito de paralelismo e assincronismo, otimizando o tempo gasto. O paralelismo diz respeito a executar tarefas em paralelo, ou seja, mais de uma tarefa será executada ao mesmo tempo. O assincronismo representa executar *threads* separadas, isto é, rodar tarefas em ordens de processamento diferentes. Esses conceitos foram aplicados usando a classe *Task* (Microsoft, 2021), nativa do C#, que possui todos os recursos necessários para esse gerenciamento. Isso foi utilizado em duas partes do código, sendo estas: na iteração dos dados de requisição e no cálculo de cada parâmetro dos modelos. Estas duas últimas serão detalhadas no decorrer deste trabalho. O assincronismo e o paralelismo devem ser utilizados com cautela, visto que há um maior consumo de recursos computacionais, logo, deve ser feita uma análise de custo-benefício levando em consideração, principalmente, a quantidade de tarefas em paralelo ou assíncronas que devem ser executadas.

Por último, o conceito de herança precisa ser detalhado, já é utilizado por todo o código, além de ser um tópico intrínseco à programação orientada a objetos. Segundo (DE CAMARGO, 2010), a herança permite que as classes partilhem seus atributos, métodos e outros membros da classe entre si, isto é, ao definir que uma classe herda da outra, a classe filho ou classe derivada recebe as características da classe pai ou classe base, tendo acesso a estas e podendo manipulá-las. Isso é bastante utilizado com o intuito de evitar replicações de código, além de impor que, em determinados contextos, alguns comportamentos sejam implementados, como será melhor apresentado mais à frente no capítulo sobre a implementação numérica dos modelos de viscoelasticidade.

2 RECURSOS IMPLEMENTADOS

Para que fosse possível atingir o objetivo deste trabalho, diversos recursos tiveram que ser implementados, sendo estes separados por tópicos, conforme mostrados a seguir.

Para cada modelo de viscoelasticidade, foi necessário implementar duas operações e as equações constitutivas, sendo estas últimas implementadas em classes separadas com o nome de cada modelo. A primeira operação é responsável pelo cálculo das propriedades mecânicas do tecido mole, em que o código recebe e retorna parâmetros específicos de cada modelo, funcionando somente como um orquestrador, em que primeiro verifica se os dados passados são válidos e, depois, chama o método responsável por efetuar os cálculos da classe que contém as equações do modelo, além de escrever esses resultados em um arquivo. A segunda operação é responsável por fazer uma análise de sensibilidade para cada parâmetro do modelo, baseando-se na funcionalidade citada anteriormente, diferindo somente nos dados de entrada, em que é possível passar um intervalo de valores para serem testados e comparados os resultados.

2.1 MÉTODOS NUMÉRICOS

Conforme será mostrado a seguir, as equações para os modelos quase-linear e não linear possuem integrais e derivadas que precisam ser resolvidas numericamente. Para viabilizar isto, foi implementado uma classe que contém os métodos para calcular uma determinada integral e outra classe que contém as lógicas para efetuar a derivada.

O método de diferenças centrais foi utilizado para a implementação da derivada numérica por possuir uma boa precisão numérica e ser de fácil implementação. A formulação está mostrada na equação 9, em que o h é representa um delta da variável x .

$$\frac{df(x)}{dx} = \frac{f(x+h) - f(x-h)}{2h} \quad (9)$$

Para a integração numérica foi utilizado o método de Simpson composto (REGRA DE SIMPSON, 2021 e REGRAS COMPOSTAS, 2021) por apresentar uma grande precisão se o intervalo de integração for dividido em espaços muito pequenos. Esse método é aplicado considerando o intervalo $[a, b]$, sendo a e b quaisquer valores reais e que a função $f(x)$ a ser integrada seja contínua dentro deste intervalo. E são adotadas N divisões, com $h = \frac{b-a}{N}$ e $N/2$ subintervalos do tipo $[x_{2k-2}, x_{2k}]$, com $k = 0, 1, \dots, N/2$. Assim, considerando os subintervalos, temos:

$$\int_a^b f(x)dx = \sum_{k=1}^{N/2} \int_{x_{2k-2}}^{x_{2k}} f(x)dx \quad (10)$$

Aplicando a regra de Simpson simples na integral da direita do subintervalo, pode-se reescrever a equação 10:

$$\int_a^b f(x)dx = \sum_{k=1}^{N/2} \frac{h}{3} [f(x_{2k-2}) + 4 \cdot f(x_{2k-1}) + f(x_{2k})] \quad (11)$$

Simplificando a equação 11, é possível obter a equação final para a Regra de Simpson Composta.

$$\int_a^b f(x)dx = \frac{h}{3} \left[f(x_0) + 4 \cdot \sum_{k=1}^{N/2} f(x_{2k-1}) + 2 \cdot \sum_{k=1}^{(N/2)-1} f(x_{2k}) + f(x_N) \right] \quad (12)$$

Conforme pode ser observado, para calcular a integral e derivada de uma função no código seria necessário calcular previamente todos os pontos que o método utilizaria. Para evitar que isto fosse feito, foi utilizado a classe Func, nativa do C#, que permite encapsular um método do código em uma variável, sem perder suas características.

2.2 DADOS EXPERIMENTAIS

As operações implementadas para os dados experimentais, obtidos em ensaios com ligamentos de joelho de porcos, foram duas, baseando-se em duas necessidades: remover pontos indesejáveis e extrapolar os dados experimentais.

A primeira é responsável por somente analisar os dados experimentais removendo os pontos considerados inválidos. Isso foi feito se baseando no comportamento esperado das curvas, em que, para o tempo de rampa, é esperado que a tensão aumente, logo, primeira derivada positiva e a concavidade da curva seja para cima, por tanto, segunda derivada positiva, e para a relaxação, a tensão deve diminuir, primeira derivada negativa, e, assim como na anterior, concavidade da curva para cima.

A principal função da segunda operação é extrapolar os dados através do menor valor encontrado para a segunda derivada, que, com isto, prevê a primeira derivada e, em seguida, prevê o valor da tensão para aquele ponto usando, em ambos os casos, método de integração trapezoidal. Contudo, para que isso seja possível, antes da etapa principal, os pontos são analisados utilizando a primeira operação supracitada.

2.3 OPERAÇÕES AUXILIARES

Além das operações supracitadas, algumas outras foram necessárias para automatizar algumas etapas feitas manualmente. A primeira é responsável por diminuir a quantidade de pontos em um arquivo. A segunda tem como objetivo analisar os dados experimentais em que um filtro é passado de modo a remover os pontos considerados incorretos. Esse filtro consiste em calcular a primeira e segunda derivadas no ponto e em remover os pontos em que estes valores são considerados inconsistentes. Por exemplo, durante a relaxação a tensão diminui, primeira derivada negativa, e a curvatura do gráfico é para cima, segunda derivada positiva, sendo assim, todos os pontos devem obedecer a isso. Esta última é a mais importante, já que foi utilizada em todos os dados experimentais para que os ajustes de curva sejam feitos com maior precisão.

3 MODELO LINEAR DE VISCOELASTICIDADE

Uma função linear é caracterizada por depender somente de uma variável. Posto isto, o modelo de viscoelasticidade linear precisa ter uma função relaxação e uma função de fluência lineares, em que ambas dependam somente do tempo. Modelos lineares são mais comumente utilizados para metais, entretanto, devido a sua simplicidade, será utilizado neste trabalho para iniciar os estudos sobre implementação numérica de modelos de viscoelasticidade. Para este modelo, as relações entre deformação e tensão para fluência e relaxação são expressas, respectivamente, pelas equações 1 e 2.

3.1 MAXWELL

O modelo de viscoelasticidade linear de Maxwell considera que o tecido mole seria representado por um amortecedor e uma mola em série, conforme mostrado na Fig. 8.

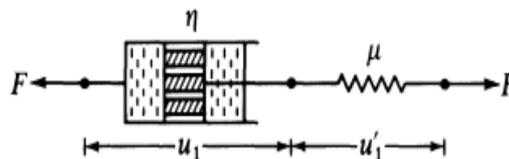


Figura 8 - Representação de mola e amortecedor em série (Fung, 1993)

Observando-se a Fig. 8, tem-se que F é a força, que dividida por uma área considerada constante, gera a tensão, μ é o módulo de elasticidade e η é a viscosidade. O deslocamento do amortecedor e da mola são representados, respectivamente, por u_1 e u'_1 . Usando as expressões de elasticidade, viscosidade e equilíbrio de forças e assumindo que as áreas são constantes por todo o comprimento, tem-se:

$$\sigma_1 = \mu \cdot \varepsilon'_1 \quad (13)$$

$$\sigma_2 = \eta \cdot \dot{\varepsilon}_1 \quad (14)$$

$$\sigma_1 = \sigma_2 = \sigma \text{ (as áreas da mola e do amortecedor são consideradas idênticas)} \quad (15)$$

$$\varepsilon_1 + \varepsilon_1' = \varepsilon \quad (16)$$

Derivando as equações 13 e 16, onde o ponto representa a derivada no tempo, tem-se:

$$\dot{\sigma}_1 = \mu \cdot \dot{\varepsilon}_1' \quad (17)$$

$$\dot{\varepsilon}_1 + \dot{\varepsilon}_1' = \dot{\varepsilon} \quad (18)$$

Substituindo as equações 14, 15 e 17 em 18, tem-se:

$$\dot{\varepsilon} = \frac{\sigma}{\eta} + \frac{\dot{\sigma}}{\mu} \quad (19)$$

Este modelo, considera-se como deformação inicial o valor da tensão inicial dividida pelo módulo de elasticidade, conforme visto na equação 20.

$$\varepsilon_0 = \frac{\sigma_0}{\mu} \quad (20)$$

Integrando a equação 14 em relação ao tempo, considerando a tensão constante e isolando ε_1 , tem-se:

$$\varepsilon_1 = \sigma_2 \cdot \frac{t}{\eta} \quad (21)$$

Isolando ε_1' na equação 13 e aplicando, em conjunto com a equação 21 na equação 16, tem-se:

$$\varepsilon = \frac{\sigma_1}{\mu} + \sigma_2 \cdot \frac{t}{\eta}$$

Substituindo σ_1 e σ_2 por σ , conforme a equação 15, ou ainda σ_0 , tem-se:

$$\varepsilon = \sigma_0 \left(\frac{1}{\mu} + \frac{t}{\eta} \right) \quad (22)$$

Por fim, se baseando na equação 1, a função fluência pode ser expressa conforme mostrado na equação 23.:

$$J(t) = \frac{1}{\mu} + \frac{t}{\eta} \quad (23)$$

Para a relaxação, o modelo de Maxwell assume a taxa de deformação igual a zero na equação 19, possuindo uma equação diferencial de primeira ordem que possui solução, considerando $\sigma(0) = \sigma_0$ e $\tau = \frac{\eta}{\mu}$. Sua resolução está descrita em sequência.

$$\frac{\sigma}{\eta} + \frac{\dot{\sigma}}{\mu} = 0$$

$$\dot{\sigma} + \frac{\mu}{\eta} \sigma = 0$$

$$\dot{\sigma} + \frac{\sigma}{\tau} = 0, \text{ onde } \tau = \frac{\eta}{\mu}$$

Resolvendo-se a equação diferencial de variáveis separáveis, tem-se:

$$\sigma(t) = \sigma_0 \cdot e^{-\frac{t}{\tau}} \quad (24)$$

Isolando a tensão inicial da equação 20 e a substituindo na equação 24, tem-se:

$$\sigma(t) = \varepsilon_0 \cdot \mu \cdot e^{-\frac{t}{\tau}} \quad (25)$$

Comparando-se a equação 25 com a equação 2, podemos definir a função relaxação como:

$$G(t) = \mu \cdot e^{-\frac{t}{\tau}} \quad (26)$$

As curvas de fluência e para o modelo de Maxwell podem ser mais bem observados nas Fig. 9.a e 9.b, onde pode-se notar o comportamento da tensão e deformação ao longo do tempo.

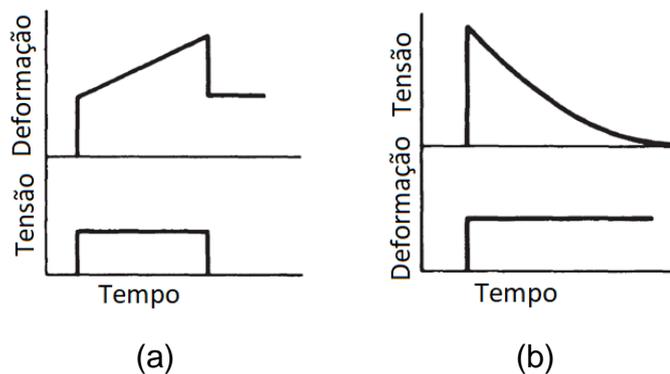


Figura 9 - Tensão e deformação ao longo do tempo para (a) fluência e para (b) relaxação

3.2 MAXWELL GENERALIZADO

O modelo generalizado de Maxwell propõe que o tecido mole pode ser representado como um conjunto de várias molas e amortecedores em série e todos estes conjuntos em paralelo com uma mola, conforme a Fig. 10.

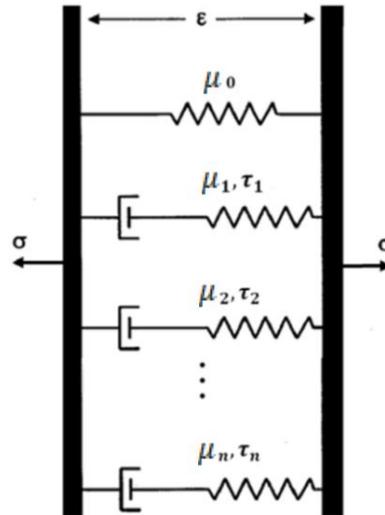


Figura 10 - Modelo generalizado de Maxwell (BABAEI, *et al*, 2015) com nomenclatura adaptada

Este modelo generalizado é capaz de expressar o comportamento de um tecido mole com uma grande quantidade de parâmetros, já que possui mais elementos do que o modelo simples de Maxwell. A função relaxação para este modelo pode ser expressa conforme a equação 27.

$$G(t) = \mu_e + \sum_{i=1}^n \mu_i \cdot e^{\left(-\frac{t}{\tau_i}\right)} \quad (27)$$

Neste caso, a função relaxação também é chamada de série de Prony para relaxação, onde μ_e é o módulo de relaxação ou módulo de Young para a situação de equilíbrio, μ_i são os múltiplos módulos de Young para cada elemento de mola e τ_i são os tempos de relaxação de cada elemento. Conforme mostrado por (BABAEI, *et al.*, 2015), na prática, essa equação apresenta boa aproximação utilizando somente 3 elementos no somatório.

4 MODELO QUASE-LINEAR DE VISCOELASTICIDADE

O modelo de Viscoelasticidade Quase-Linear, proposto inicialmente por (Fung, 1993), aplica a não-linearidade da relação entre tensão e deformação expressando a tensão em duas partes: a função relaxação reduzida, que depende somente do tempo, e a resposta elástica, que depende da deformação. As equações constitutivas para a relaxação e fluência foram desenvolvidas se baseando no método de superposição de Boltzmann para viscoelasticidade com alguns ajustes para aplicar a função relaxação reduzida e a resposta elástica, parâmetros citados anteriormente.

Este é um modelo comumente utilizado para tecidos moles, por conseguir representar a realidade com boa aproximação conforme será mostrado mais a diante nas comparações entre os resultados experimentais e numéricos. As constantes necessárias para as equações não apresentam grandes dificuldades de serem obtidas, já que não são necessários muitos experimentos para tal. Possuindo os dados experimentais, basta aplicar ajustes de curva através de softwares como o OriginPro ou MATLAB para calculá-las. Entretanto, assim como qualquer modelo, este apresenta limitações, uma vez que para diferentes relaxações e patamares de deformação são encontrados valores diferentes para estas constantes.

4.1 EQUACIONAMENTO

(Fung, 1993) propõe equações para serem usadas para o cálculo da função relaxação reduzida e tensão e não propõe uma formulação para a resposta elástica. Isso porque o autor considera que a deformação aumenta de forma tão rápida que o tempo de rampa pode ser desconsiderado e a resposta elástica é constante ao longo de todo o experimento com valor igual ao obtido no momento do pico da deformação. Contudo, diversos autores consideraram o tempo de rampa e propõem equações para descrever a resposta elástica, sendo a equação exponencial a mais utilizada, conforme será mostrado a seguir.

4.1.1 Deformação

Para a descrição da deformação, é utilizado a equação 28 em que se pretende representar como foi feito nos ensaios experimentais, e a equação 29 quando o tempo de rampa não é considerado. Os parâmetros λ e t_0 representam, respectivamente, a taxa de deformação aplicada durante o experimento e o tempo de rampa. λ_1 para o caso de deformações crescentes e λ_2 para o caso de deformações decrescentes. t_1, \dots, t_{14} representam os tempos limites em que o comportamento da deformação é alterado.

$$\varepsilon(t) = \begin{cases} \lambda_1 \cdot t, & 0 \leq t < t_0 \\ \varepsilon_0, & t_0 \leq t < t_1 \\ \varepsilon_0 - \lambda_2 \cdot (t - t_1), & t_1 \leq t < t_2 \\ \varepsilon_{\min}, & t_2 \leq t < t_3 \\ \varepsilon_{\min} + \lambda_1 \cdot (t - t_3), & t_3 \leq t < t_4 \\ \varepsilon_1, & t_4 \leq t < t_5 \\ \varepsilon_1 - \lambda_2 \cdot (t - t_5), & t_5 \leq t < t_6 \\ \varepsilon_{\min}, & t_6 \leq t < t_7 \\ \varepsilon_{\min} + \lambda_1 \cdot (t - t_7), & t_7 \leq t < t_8 \\ \varepsilon_2, & t_8 \leq t < t_9 \\ \varepsilon_2 - \lambda_2 \cdot (t - t_9), & t_9 \leq t < t_{10} \\ \varepsilon_{\min}, & t_{10} \leq t < t_{11} \\ \varepsilon_{\min} + \lambda_1 \cdot (t - t_{11}), & t_{11} \leq t < t_{12} \\ \varepsilon_3, & t_{12} \leq t < t_{13} \\ \varepsilon_3 - \lambda_2 \cdot (t - t_{11}), & t_{13} \leq t < t_{14} \\ \varepsilon_{\min}, & t \geq t_{14} \end{cases} \quad (28)$$

$$\varepsilon(t) = \varepsilon_i \quad (29)$$

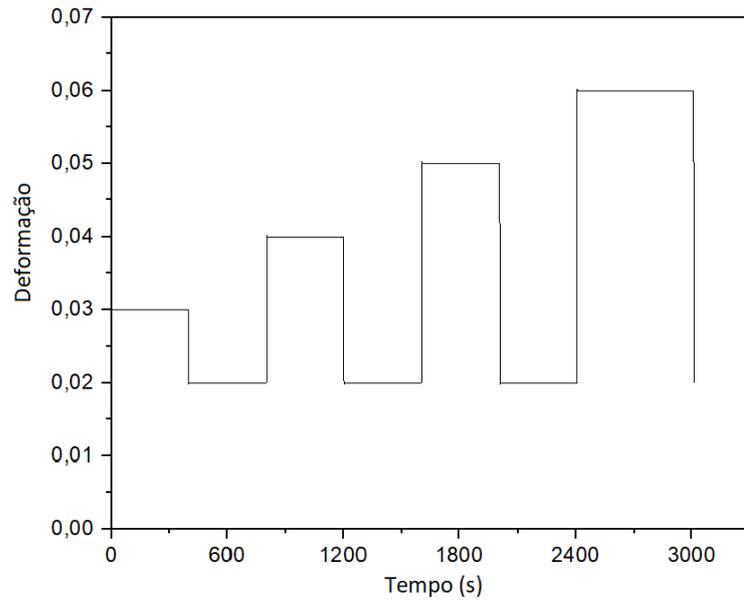


Figura 11 - Gráfico de deformação versus tempo utilizada nos ensaios

Desse modo, pode-se calcular a derivada da deformação, que mais a frente será usada para calcular a tensão. Derivando a equação 29, tem-se:

$$\frac{d\varepsilon(t)}{dt} = \begin{cases} \lambda_1, & 0 \leq t < t_0 \\ 0, & t_0 \leq t < t_1 \\ -\lambda_2, & t_1 \leq t < t_2 \\ 0, & t_2 \leq t < t_3 \\ \lambda_1, & t_3 \leq t < t_4 \\ 0, & t_4 \leq t < t_5 \\ -\lambda_2, & t_5 \leq t < t_6 \\ 0, & t_6 \leq t < t_7 \\ \lambda_1, & t_7 \leq t < t_8 \\ 0, & t_8 \leq t < t_9 \\ -\lambda_2, & t_9 \leq t < t_{10} \\ 0, & t_{10} \leq t < t_{11} \\ \lambda_1, & t_{11} \leq t < t_{12} \\ 0, & t_{12} \leq t < t_{13} \\ -\lambda_2, & t_{13} \leq t < t_{14} \\ 0, & t \geq t_{14} \end{cases} \quad (30)$$

Portanto, a derivada somente será diferente de zero quando a deformação não for constante.

4.1.2 Tensão

Para se calcular a tensão existem três diferentes equações de acordo com (Fung, 1993), sendo estas expressas a seguir.

$$\sigma(t) = \sigma^e(0) \cdot G(t) + \int_0^t G(t - \tau) \frac{\partial \sigma^e(\tau)}{\partial \tau} d\tau \quad (31)$$

$$\sigma(t) = G(0) \cdot \sigma^e(t) + \int_0^t \sigma^e(t - \tau) \frac{\partial G(\tau)}{\partial \tau} d\tau \quad (32)$$

$$\sigma(t) = \frac{\partial}{\partial t} \int_0^t \sigma^e(t - \tau) G(\tau) d\tau \quad (33)$$

Conforme será mostrado mais adiante, a resposta elástica e a função relaxação reduzida dependem somente do tempo, o que permite reescrever as derivadas parciais como derivadas totais. Ademais, para o modelo de Fung (1993), tem-se que $G(0) = 1$ e $\sigma^e(0) = 0$. Sendo assim, pela equação 34, a tensão é calculada a partir da integral do produto entre a função relaxação reduzida e o incremento da resposta elástica, enquanto as demais são equações matematicamente equivalentes.

$$\sigma(t) = \int_0^t G(t - \tau) \frac{d\sigma^e(\tau)}{d\tau} d\tau \quad (34)$$

$$\sigma(t) = \sigma^e(t) + \int_0^t \sigma^e(t - \tau) \frac{dG(\tau)}{d\tau} d\tau \quad (35)$$

$$\sigma(t) = \frac{d}{dt} \int_0^t \sigma^e(t - \tau) G(\tau) d\tau \quad (36)$$

Segundo (Fung, 1993), essas equações são matematicamente equivalentes, logo, em teoria, devem retornar os mesmos valores ou resultados muito próximos. Entretanto, vale ressaltar que a equação 34 é amplamente utilizada em diversos trabalhos, sendo considerada a principal equação para este estudo e utilizada como base para comparação dos resultados obtidos nas demais equações. Essas equações são mais bem aplicadas quando se considera o tempo de rampa, quando isto não é

considerado, as equações 35 e 36 podem ser reescritas conforme mostrado no Anexo A deste documento.

4.1.3 Resposta elástica (parcela elástica)

A resposta elástica pode ser calculada através da equação 37, proposta por (ABRAMOWITCH *et al.*, 2004), sendo em função da deformação, porém, como a deformação pode ser expressa em função do tempo, a resposta elástica também pode ser reescrita em função somente do tempo. Os parâmetros A e B da equação representam, respectivamente, *elastic stress constant* e *elastic stress power*. Mesmo que essa propriedade dependa da deformação, que para diferentes intervalos de tempo sejam usadas diferentes equações, pode-se usar somente a equação 37, além disso, observando a equação 28, pode-se perceber que em alguns intervalos de tempo a resposta elástica será constante e não dependerá do tempo.

$$\sigma^e(t) = A \cdot (e^{B \cdot \varepsilon(t)} - 1) \quad (37)$$

Para o caso em que o tempo de rampa é desconsiderado, é usado o valor de tensão inicial aplicado ao experimento, σ_0 , por todo o domínio de tempo.

$$\sigma^e(t) = \sigma_0 \quad (38)$$

Assim como feito para a deformação, também será calculado a derivada da resposta elástica, devido ao mesmo motivo, sua importância no cálculo da tensão, sendo assim, derivando-se a equação 37.

$$\frac{d\sigma^e(t)}{dt} = A \cdot B \cdot e^{B\varepsilon(t)} \cdot \frac{d\varepsilon(t)}{dt} \quad (39)$$

Para esta equação, vale a mesma consideração feita para a equação 37, onde é válido utilizar, para fins computacionais, a equação 39. Porém, vale lembrar que

$\frac{d\varepsilon(t)}{dt}$ só é diferente de zero para os casos em que a deformação não é constante, sendo assim, é somente nesses momentos em que a derivada da resposta elástica é diferente de zero.

4.1.4 Função relaxação reduzida (parcela viscosa)

A função relaxação reduzida corresponde a parcela viscosa das equações de viscoelasticidade e, conceitualmente, corresponde a função relaxação, que representa a rigidez, dividida pelo valor de tensão no início da relaxação, dessa forma, no modelo de Fung, esse parâmetro é adimensional. Ela pode ser escrita de duas maneiras segundo (Fung, 1993), a primeira, a equação 40, é chamada de simplificada e apresenta maior facilidade de ser obtida em experimentos por ser representada conforme a série de Prony, enquanto a segunda, a equação 41, foi obtida a partir de diversos cálculos matemáticos mais complexos baseados no modelo de Kelvin (*standard linear solid*) (Fung, 1993). Vale ressaltar que para o presente trabalho foi considerado somente três somas de exponenciais da equação 40, conforme citado anteriormente para o modelo de Maxwell generalizado. Ademais, para o modelo de Fung (1993), para o tempo igual a zero, a função relaxação reduzida é igual a 1, $G(0) = 1$.

Os parâmetros C e τ correspondem, respectivamente, a amplitude de relaxação e o tempo de relaxação, para a equação 40, os índices são somente para diferenciar cada constante, enquanto para a equação 41, τ_1 representa o tempo de relaxação rápido e τ_2 , o tempo de relaxação lento, sendo τ_2 sempre maior do que τ_1 .

$$G(t) = C_0 + C_1 \cdot e^{-\frac{t}{\tau_1}} + C_2 \cdot e^{-\frac{t}{\tau_2}} + C_3 \cdot e^{-\frac{t}{\tau_3}} \quad (40)$$

$$G(t) = \frac{1 + C \cdot \left[E_1\left(\frac{t}{\tau_2}\right) - E_1\left(\frac{t}{\tau_1}\right) \right]}{\left[1 + C \cdot \ln\left(\frac{\tau_2}{\tau_1}\right) \right]}, \text{ com } E_1(z) = \int_z^{\infty} \frac{e^{-x}}{x} dx \quad (41)$$

Como $\tau_2 > \tau_1$, pode-se afirmar que $\frac{t}{\tau_2} < \frac{t}{\tau_1}$, sendo assim, $E_1\left(\frac{t}{\tau_2}\right)$ pode ser reescrito como:

$$E_1\left(\frac{t}{\tau_2}\right) = \int_{\frac{t}{\tau_2}}^{\infty} \frac{e^{-x}}{x} dx = \int_{\frac{t}{\tau_2}}^{\frac{t}{\tau_1}} \frac{e^{-x}}{x} dx + \int_{\frac{t}{\tau_1}}^{\infty} \frac{e^{-x}}{x} dx$$

Mas, $E_1\left(\frac{t}{\tau_1}\right) = \int_{\frac{t}{\tau_1}}^{\infty} \frac{e^{-x}}{x} dx$.

Logo:

$$E_1\left(\frac{t}{\tau_2}\right) - E_1\left(\frac{t}{\tau_1}\right) = \int_{\frac{t}{\tau_2}}^{\frac{t}{\tau_1}} \frac{e^{-x}}{x} dx + \int_{\frac{t}{\tau_1}}^{\infty} \frac{e^{-x}}{x} dx - \int_{\frac{t}{\tau_1}}^{\infty} \frac{e^{-x}}{x} dx = \int_{\frac{t}{\tau_2}}^{\frac{t}{\tau_1}} \frac{e^{-x}}{x} dx$$

Por fim, a equação 41 pode ser reescrita como:

$$G(t) = \frac{1 + C \cdot \int_{\frac{t}{\tau_2}}^{\frac{t}{\tau_1}} \frac{e^{-x}}{x} dx}{1 + C \cdot \ln\left(\frac{\tau_2}{\tau_1}\right)} \quad (42)$$

Simplificando, tem-se:

$$G(t) = \frac{1 + C \cdot I(t)}{1 + C \cdot \ln\left(\frac{\tau_2}{\tau_1}\right)}, \quad \text{onde } I(t) = \int_{\frac{t}{\tau_2}}^{\frac{t}{\tau_1}} \frac{e^{-x}}{x} dx \quad (43)$$

Como feito nas propriedades anteriores, calcula-se a derivada da função relaxação reduzida para as equações 40 e 43. Derivando-se a equação 40:

$$\frac{dG(t)}{dt} = -\frac{C_1}{\tau_1} \cdot e^{-\frac{t}{\tau_1}} - \frac{C_2}{\tau_2} \cdot e^{-\frac{t}{\tau_2}} - \frac{C_3}{\tau_3} \cdot e^{-\frac{t}{\tau_3}} \quad (44)$$

Derivando-se a equação 43:

$$\frac{dG(t)}{dt} = \frac{d}{dt} \left[\frac{1 + C \cdot I(t)}{1 + C \cdot \ln\left(\frac{\tau_2}{\tau_1}\right)} \right] = \frac{C}{1 + C \cdot \ln\left(\frac{\tau_2}{\tau_1}\right)} \cdot \frac{d}{dt} [I(t)]$$

$$\frac{d}{dt} [I(t)] = \frac{d}{dt} \left[\int_{\frac{t}{\tau_2}}^{\frac{t}{\tau_1}} \frac{e^{-x}}{x} dx \right]$$

Aplicando a definição de cálculo para derivada de uma integral determinada:

$$\frac{d}{dt} \int_b^a f(x) dx = \frac{d}{dt} [F(a) - F(b)] = f(a) \cdot \frac{da}{dt} - f(b) \cdot \frac{db}{dt}$$

Em que:

$$f(x) = \frac{e^{-x}}{x}, \quad a = \frac{t}{\tau_1}, \quad b = \frac{t}{\tau_2}$$

Tem-se que:

$$\frac{d}{dt} \left[\int_{\frac{t}{\tau_2}}^{\frac{t}{\tau_1}} \frac{e^{-x}}{x} dx \right] = \frac{e^{-\frac{t}{\tau_1}}}{\frac{t}{\tau_1}} \cdot \frac{d}{dt} \left(\frac{t}{\tau_1} \right) - \frac{e^{-\frac{t}{\tau_2}}}{\frac{t}{\tau_2}} \cdot \frac{d}{dt} \left(\frac{t}{\tau_2} \right) = \frac{e^{-\frac{t}{\tau_1}}}{\frac{t}{\tau_1}} \cdot \frac{1}{\tau_1} - \frac{e^{-\frac{t}{\tau_2}}}{\frac{t}{\tau_2}} \cdot \frac{1}{\tau_2} = \frac{e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}}}{t}$$

Logo:

$$\frac{d}{dt} [I(t)] = \frac{e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}}}{t}$$

Portanto:

$$\frac{dG(t)}{dt} = \frac{C \cdot \left[e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}} \right]}{\left[1 + C \cdot \ln\left(\frac{\tau_2}{\tau_1}\right) \right] \cdot t} \quad (45)$$

4.2 OBTENÇÃO DAS CONSTANTES

Neste trabalho, será utilizado somente a forma simplificada da função relaxação reduzida, expressa pela equação 40. Como citado anteriormente, as constantes para o modelo de Fung são obtidas a partir dos dados experimentais ao fazer ajustes de curva nestes.

Para a obtenção das constantes para a função relaxação reduzida, precisa-se utilizar a parte em que ocorre a relaxação no gráfico de tensão versus tempo, em que nesta parte é aplicado um ajuste de curva utilizando a equação 40, conforme mostrado na Fig. 12.

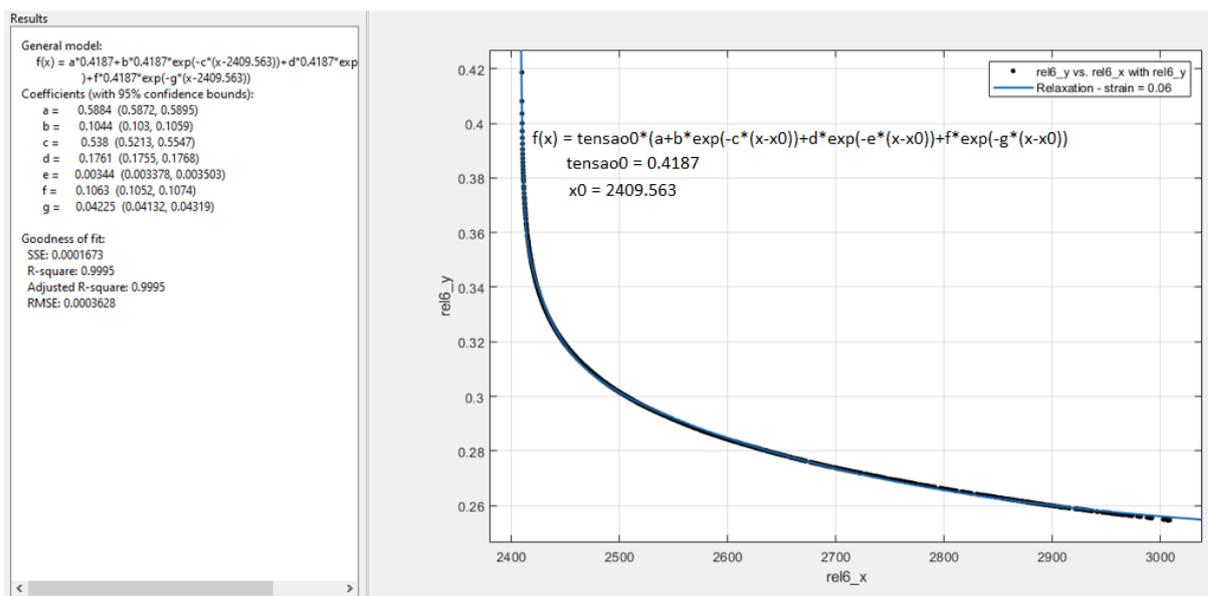


Figura 12 – Ajuste de curva durante relaxação

Para obter as constantes A e B para a resposta elástica é necessário utilizar o gráfico tensão versus tempo durante o tempo de rampa ($0 \leq t \leq t_0$) e basta aplicar um ajuste de curva nesta parte utilizando a equação 37 e considerando a deformação igual a $\lambda_1 \cdot t$ (equação utilizada para descrever a deformação durante o tempo de rampa). Isso é mostrado na Fig. 13.

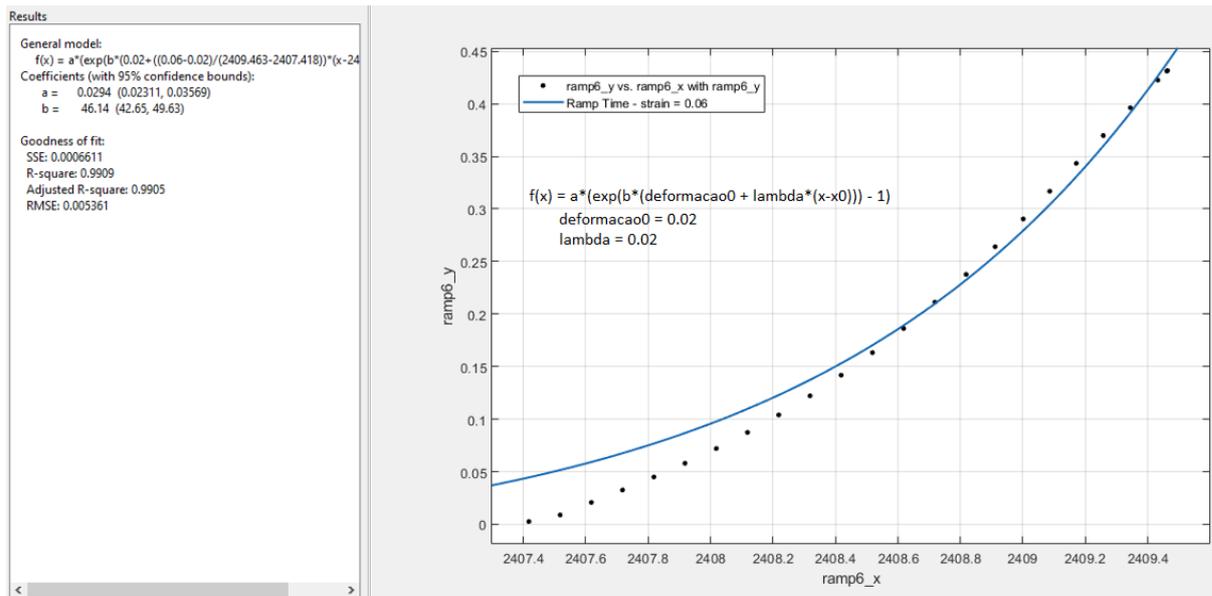


Figura 13 – Ajuste de curva durante tempo de rampa

5 MODELO NÃO-LINEAR DE VISCOELASTICIDADE

Assim como o modelo quase-linear de Fung, os modelos não-lineares se baseiam no método de superposição de Boltzmann para viscoelasticidade, em que algumas adaptações são feitas para que uma relação não-linear seja estabelecida entre tensão e deformação, conforme será apresentado com o desenvolvimento do trabalho. Além disto, para os modelos não-lineares, diferentemente do modelo de Fung, a função relaxação representa a rigidez do material e possui unidade em MPa. Para estes modelos, a ausência do tempo de rampa não impacta na obtenção de suas constantes.

Os modelos de Schapery e de Superposição Não-Linear, que serão melhor apresentados a seguir, apresentam maior precisão quando comparados com o modelo de Fung, uma vez que suas constantes levam em consideração diferentes níveis de deformação. Devido a isso, estes modelos apresentam grandes aplicações em estudos de viscoelasticidade em diversos materiais, além de tecidos moles, já que representam com maior precisão a realidade, mesmo em diferentes patamares de deformação. As constantes para esses modelos também são obtidas após ajustes de curva dos dados experimentais por meio de softwares, como OriginPro e MATLAB.

5.1 SCHAPERY

O modelo não-linear de Schapery (1969, 1997, 2000) aplica ao método de superposição de Boltzmann conceitos de termodinâmica para estabelecer uma relação não-linear entre deformação e tensão. Dessa forma, é possível considerar a variação da temperatura nas equações. Entretanto, isto não será considerado, visto que o foco deste trabalho é em ensaios de relaxação à temperatura constante.

A principal limitação de Schapery é que este modelo depende do valor da função relaxação na condição de equilíbrio, ou seja, para um tempo que tende ao infinito, conforme será apresentado mais a diante. Isto é considerado uma limitação, visto que a precisão do modelo numérico depende de encontrar o estado de equilíbrio nos ensaios experimentais, o que pode demandar tempos na faixa de horas, como observado em diversos estudos.

5.1.1 Equacionamento

Conforme citado anteriormente, Schapery (1969) parte do modelo de superposição de Boltzmann com modificações para que sejam considerados fenômenos termodinâmicos. Dessa forma, foi estabelecido uma relação não-linear entre a deformação e tensão para o caso de fluência e, a partir disso, Schapery (1969) afirma que para o caso de relaxação, uma equação análoga pode ser escrita, conforme será apresentado em sequência.

5.1.1.1 Deformação

Para a deformação e sua derivada, foi utilizado a equação 28, a mesma utilizada para Fung ao considerar o tempo de rampa.

5.1.1.2 Tensão

Inicialmente, Schapery (1969) expressa a tensão sendo calculada a partir da condição de equilíbrio e de uma parcela transiente. Dessa forma, sob uma ótica linear, pode se determinar a seguinte equação: $\sigma(t) = G_e \cdot \varepsilon + \Delta G(t) \cdot \varepsilon$, em que $G_e \cdot \varepsilon$ corresponde à condição de equilíbrio e $\Delta G(t) \cdot \varepsilon$, com $\Delta G(t) = G(t) - G_e$, corresponde à parcela transiente.

Ao utilizar o método de superposição de Boltzmann, essa relação é expressa da seguinte forma:

$$\sigma(t) = G_e \cdot \varepsilon(t) + \int_0^t \Delta G(t - \tau) \frac{d\varepsilon(\tau)}{d\tau} d\tau \quad (46)$$

Contudo, esta relação é linear, sendo assim, Schapery (1969) propôs algumas mudanças na equação 46 de modo que haja uma relação não-linear entre tensão e deformação. Essas mudanças consistem em considerar 4 constantes dependentes da deformação, sendo estas h_e , h_1 , h_2 e a_ε , além disso, as três primeiras também dependem da energia livre de Helmholtz (ANDRADE *et al.* 2020) e a última depende

da entropia e energia livre do corpo. Sendo assim, a equação 46 foi reescrita da seguinte forma:

$$\sigma(\varepsilon, t) = h_e(\varepsilon) \cdot G_e \cdot \varepsilon(t) + h_1(\varepsilon) \cdot \int_0^t \Delta G(\rho(t) - p(\tau)) \cdot \frac{d[h_2(\varepsilon) \cdot \varepsilon(\tau)]}{d\tau} d\tau \quad (47)$$

$$\text{onde, } \rho(z) = \int_0^z \frac{dx}{a_\varepsilon[\varepsilon(x)]}$$

Para valores baixos de tensão, a equação 47 pode ser simplificada definindo as constantes h_1 e a_ε iguais a 1, apresentada na equação 48, conforme apresentado por (SCHAPERLY, 1969), (PROVENZANO *et al.*, 2002) e (DUENWALD *et al.*, 2009). Esta equação será adotada uma vez que isto se enquadra em análises de tecidos moles.

$$\sigma(\varepsilon, t) = h_e(\varepsilon) \cdot G_e \cdot \varepsilon(t) + \int_0^t \Delta G(t - \tau) \cdot \frac{d[h_2(\varepsilon) \cdot \varepsilon(\tau)]}{d\tau} d\tau \quad (48)$$

Schaperly (1969) propõe que para as constantes h_e e h_2 sejam utilizados valores absolutos, uma vez que em seus ensaios, a deformação aumenta de maneira instantânea.

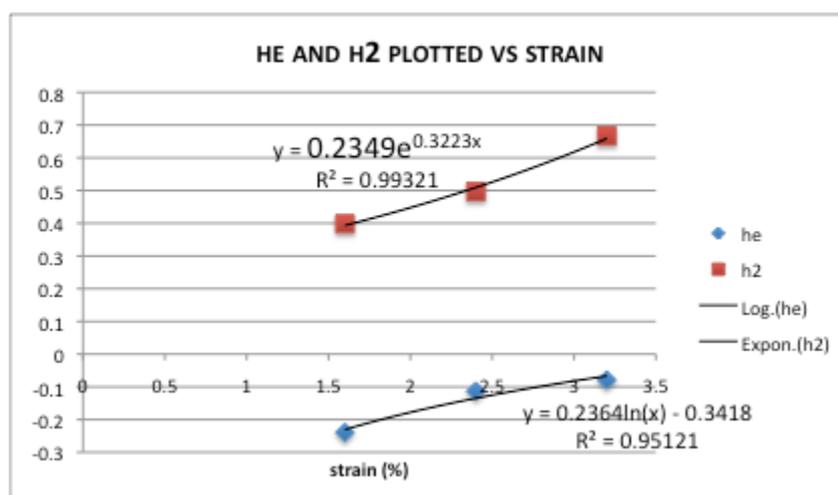


Figura 14 – h_e e h_2 em função da deformação (BLANDFORD, 2017).

Por outro lado, (BLANDFORD, 2017) propõe uma relação exponencial para h_2 , em que $h_2 = a \cdot e^{b\varepsilon}$, e logarítmica para h_e , em que $h_e = a + b \cdot \ln(\varepsilon)$, sendo a e b constantes reais para ambos os casos, presente na Fig. 14. Dessa forma considera-se que a deformação evolui de modo gradual conferindo maior precisão ao modelo numérico. Essas relações são utilizadas, pois, segundo a autora, foram as equações que apresentaram menor erro.

Ao desconsiderar o tempo de rampa e assumindo uma deformação ε_i qualquer, segundo (PROVENZANO, 2002) e (BLANDFORD, 2017), a equação 48 pode ser reescrita conforme apresentado na equação 49.

$$\sigma(\varepsilon_i, t) = h_e(\varepsilon_i) \cdot G_e \cdot \varepsilon_i + h_2(\varepsilon_i) \cdot \varepsilon_i \cdot \Delta G(t) \quad (49)$$

5.1.1.3 Função relaxação transiente

Para esta função, (Schapery, 1969) propõe a utilização da série de Prony, similar à utilizada para o modelo simplificado de Fung, equação 50, em que o parâmetro τ corresponde ao tempo de relaxação em segundo e G representa a rigidez do material com unidade em MPa.

$$\Delta G(t) = G_1 \cdot e^{-\frac{t}{\tau_1}} + G_2 \cdot e^{-\frac{t}{\tau_2}} + G_3 \cdot e^{-\frac{t}{\tau_3}} \quad (50)$$

Contudo, (DUENWALD *et al.*, 2009), (PROVENZANO *et al.*, 2002) e (BLANDFORD, 2017) propõem a utilização do modelo de Ostwald-de-Waele, equação 51, (POWER LAW, 2007), comumente aplicado em estudos de viscosidade para fluidos não-newtonianos, em que, para tecidos moles, o tempo e a tensão durante a relaxação são correlacionados de maneira logarítmica. Para esta equação, os parâmetros C representa a rigidez do material à relaxação no tempo inicial e n representa a taxa de relaxação, que é a inclinação em um gráfico log-log de tensão versus tempo.

$$\Delta G(t) = C \cdot t^{-n} \quad (51)$$

Aplicando a equação 51 na equação 49, temos a equação 52 que é muito utilizada em diversos estudos como base para obtenção das constantes de Schapery.

$$\sigma(\varepsilon_i, t) = h_e(\varepsilon_i) \cdot G_e \cdot \varepsilon_i + h_2(\varepsilon_i) \cdot C \cdot \varepsilon_i \cdot t^{-n} \quad (52)$$

5.1.2 Obtenção das constantes

A obtenção das constantes, para a função relaxação transiente para o caso em que é utilizado a série de Prony, equação 50, é feita de maneira parecida como foi feito para o modelo de Fung. Contudo, para Schapery, as constantes obtidas para um patamar de deformação são utilizadas para todos os demais níveis de deformação, em que, é comumente escolhido o menor patamar de deformação.

Já para o caso em que é utilizado o modelo de Ostwald-de-Waele, equação 51, também é utilizado o período de relaxação para a obtenção das constantes C e n presentes na equação 51. (PROVENZANO *et al.*, 2002) propõem que, para determinar C e n , é necessário escolher a curva de uma determinada deformação, sendo mais comum ser usado a de menor deformação, em escala log-log e definir h_e e h_2 iguais a um e fazer o ajuste de curva se baseando na equação 52. Isso é mostrado na Figura 15.

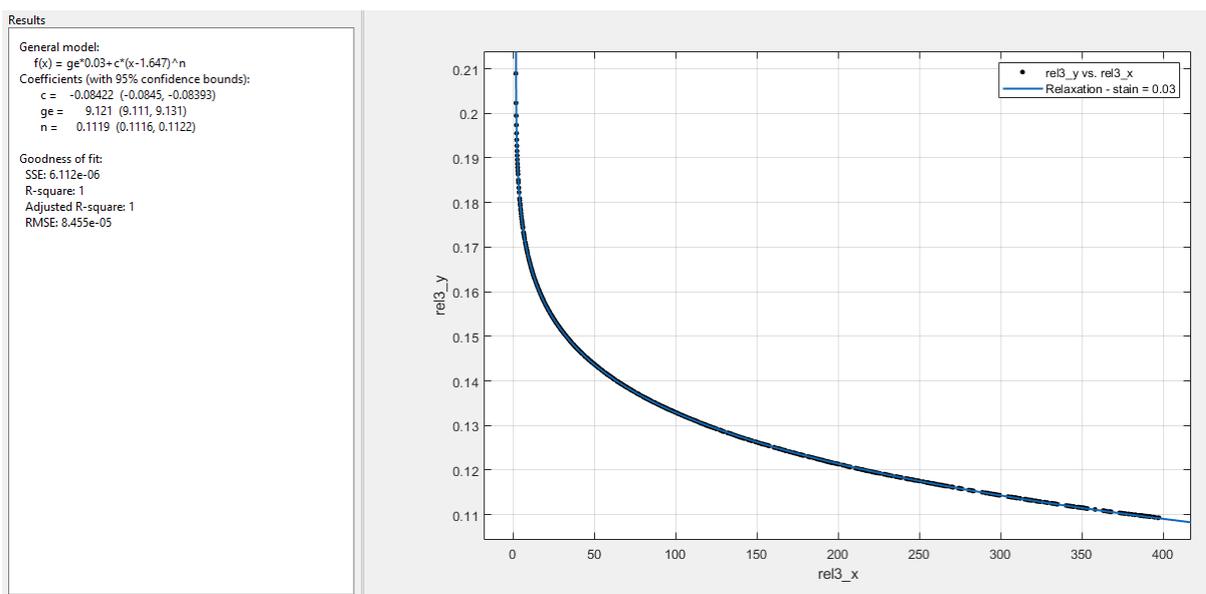


Figura 15 – Ajuste de curva para constantes C e n segundo (PROVENZANO *et al.*, 2002).

A escala em log-log é importante pois, a partir da inclinação da curva, é definido o valor de n e, a partir do valor inicial, é extraído o valor de C .

Entretanto, (BLANDOFORD, 2017) alega que isso não apresenta grande precisão mesmo fazendo sentido matematicamente. Ela propôs que fosse calculado o módulo da tangente em cada tempo para a curva de relaxação e, com esses valores, fosse feito um ajuste de curva aplicando a derivada da equação 52. Isso está representado na Fig. 16.

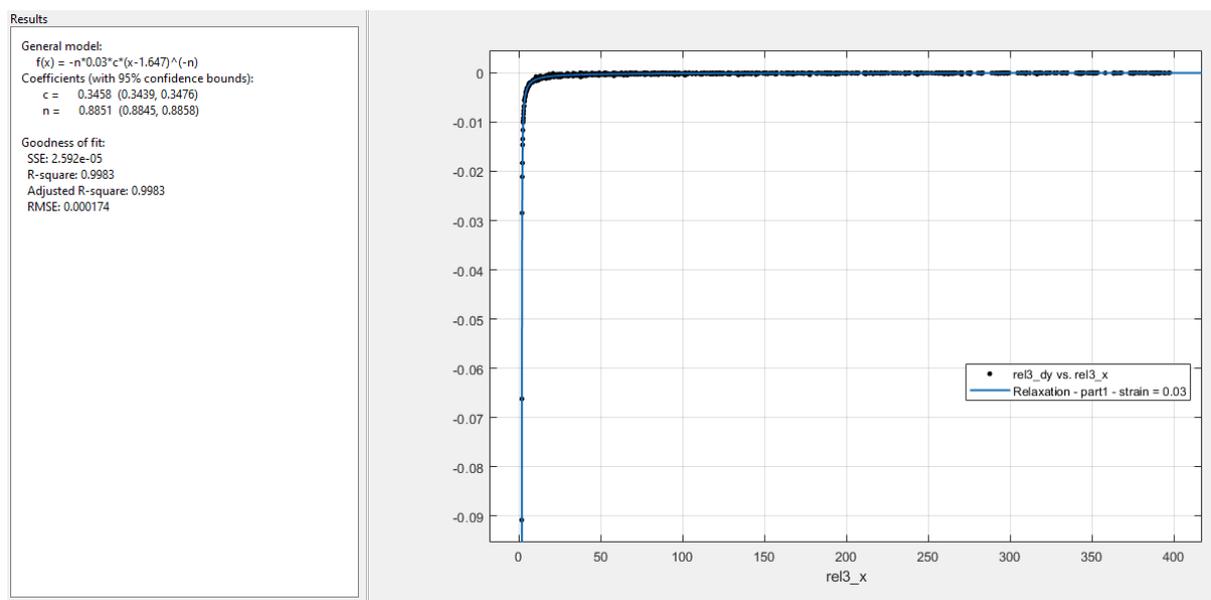
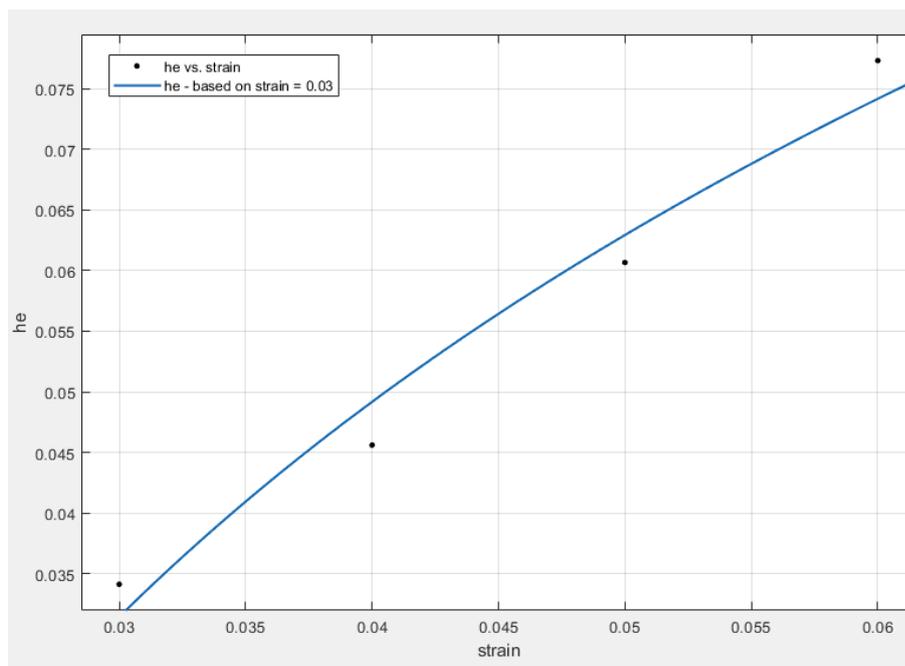


Figura 16 – Ajuste de curva para constantes C e n segundo (BLANDOFORD, 2017)

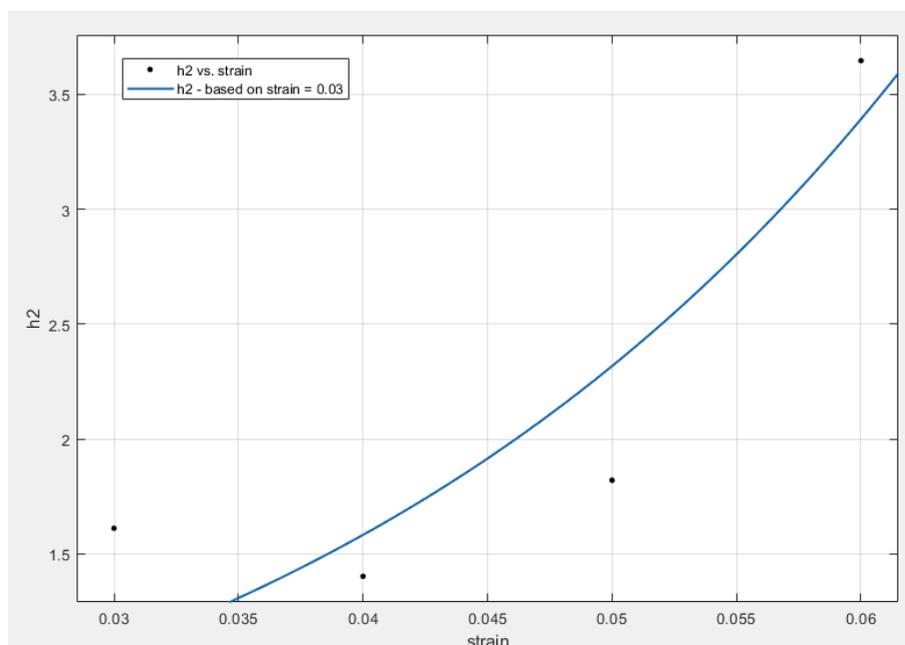
Para a constante G_e , (PROVENZANO *et al.*, 2002) propõe extrair dos dados experimentais, sendo necessários longos tempos de relaxação, já que eles só pararam seus ensaios ao perceberem que a tensão estabilizou em um valor durante o relaxamento. Porém, como citado anteriormente, para chegar neste valor, é necessários tempos muito longos de ensaio, em alguns casos podendo durar horas, (BLANDOFORD, 2017) propôs em calcular essa constante a partir do módulo tangente no gráfico tensão versus tempo para o maior tempo do ensaio. Todavia, para este estudo, a constante G_e foi obtida a partir da série de Prony.

Para a obtenção das constante h_e e h_2 , ambos os autores citados acima fazem da mesma forma. Após definirem as constantes para a função relaxação transiente, utilizam a equação 52 para fazer um ajuste de curva nas demais curvas de relaxação para cada deformação, obtendo assim, os valores de h_e e h_2 para cada deformação. (BLANDOFORD, 2017), além disso, conforme citado anteriormente, estabelece

relações exponencial e logarítmica entre os valores obtidos para h_e e h_2 , conferindo maior precisão ao modelo. Estas relações não foram utilizadas já que apresentaram grandes divergências conforme apresentado na Fig. 17.



(a)



(b)

Figura 17 – (a) h_e e (b) h_2 em função da deformação segundo (BLANDOFORD, 2017).

5.2 MÉTODO DE SUPERPOSIÇÃO NÃO-LINEAR

O modelo de superposição não-linear, mais conhecido como método de superposição modificado, inicialmente proposto por (FINDLEY *et al.*, 1976) e mais bem detalhado por (LAKES, 1998), se baseia no modelo de superposição de Boltzmann modificando apenas a função relaxação, para esta ser dependente do tempo e da deformação. Dessa forma, uma equação não-linear pode ser escrita para a tensão, fazendo com que esta dependa tanto da deformação quanto do tempo. A principal limitação deste modelo é que, para níveis de deformação fora do intervalo conhecido, sua precisão não será a mesma, conforme será explicado mais a diante.

5.2.1 Equacionamento

Para este modelo, diversos estudos propõem abordagens análogas conforme será apresentado a seguir, em que a única preocupação é em determinar o comportamento da função relaxação, que neste caso depende tanto da deformação quanto do tempo.

5.2.1.1 Deformação

Para a deformação e sua derivada, foi utilizado a equação 28, a mesma utilizada para Fung ao considerar o tempo de rampa.

5.2.1.2 Tensão

A equação utilizada que relaciona tensão e deformação para ensaios de relaxação e ao considerar o tempo de rampa é expressa pela equação 53, conforme apresentado por (FINDLEY *et al.*, 1976) e (LAKES, 1998).

$$\sigma(\varepsilon, t) = \int_0^t G(t - \tau, \varepsilon(\tau)) \cdot \frac{d\varepsilon(\tau)}{d\tau} d\tau \quad (53)$$

Ao desconsiderar o tempo de rampa, (PROVENZANO, 2002) e (BLANDFORD, 2017) apresentam a seguinte formulação:

$$\sigma(\varepsilon_i, t) = G(t, \varepsilon_i) \cdot \varepsilon_i \quad (54)$$

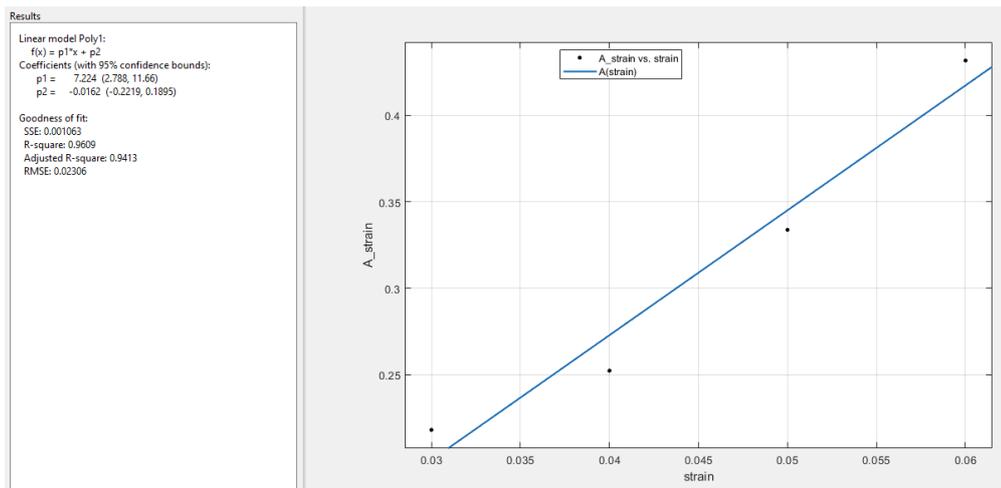
5.2.1.3 Função Relaxação

Para este modelo, a função relaxação é expressa pela equação 55, escrita se baseando no modelo de Ostwald-de-Waele (POWER LAW, 2007), em que os parâmetros $A(\varepsilon)$ e $B(\varepsilon)$ representa, respectivamente, o módulo de elasticidade inicial e a taxa de relaxação, sendo ambas dependentes da deformação. A primeira pode ser expressa como uma função linear, mostrado por (BLANDFORD, 2017), ou como uma função exponencial, apresentado por (PROVENZANO, 2002). Ambas as formas são aceitáveis, uma vez que o intuito é de escolher uma equação que apresente menor erro no ajuste de curva. A segunda é expressa como uma função polinomial que é obtida a partir das inclinações das curvas de relaxação conforme será mais bem explicado no tópico “Obtenção das variáveis”.

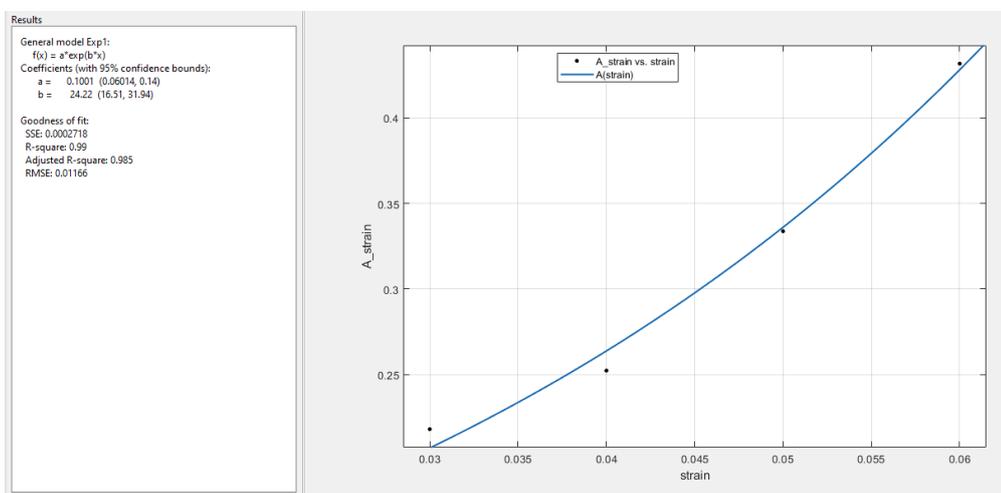
$$\begin{aligned} G(t, \varepsilon) &= A(\varepsilon) \cdot t^{B(\varepsilon)} \\ A(\varepsilon) &= a \cdot \varepsilon^b \text{ ou } A(\varepsilon) = a \cdot \varepsilon + b \\ B(\varepsilon) &= \sum_i^n a_i \cdot \varepsilon^i, \text{ com } i = 0, 1, \dots, n \end{aligned} \quad (55)$$

5.2.2 Obtenção das variáveis

Para este modelo, o tempo de rampa não possui tanta importância como para os modelos anteriormente apresentados, visto que, somente será utilizado o valor da tensão após o tempo de rampa ($t = t_0$). Dessa forma, com esses valores, é possível calcular $A(\varepsilon)$ para cada deformação e, com estes valores, é necessário fazer um ajuste de curva, conforme apresentado anteriormente. Neste trabalho, foram feitas para as funções linear e exponencial, comparando-se o erro entre cada uma, conforme presente na Fig. 18.



(a)



(b)

Figura 18 – Ajuste de curva para determinar a constante $A(\varepsilon)$ usando (a) função linear e (b) função exponencial.

Para determinar $B(\varepsilon)$, é necessário calcular a inclinação das curvas de relaxação para os gráficos tensão versus tempo para cada deformação utilizando a escala log-log. Com essas inclinações, uma equação polinomial deve ser definida, conforme mostrado na Fig. 19. Vale salientar que, quanto mais pontos, maior será o grau deste polinômio e maior precisão o modelo possuirá. Por se tratar de um polinômio, o parâmetro $B(\varepsilon)$ possuirá boa precisão somente dentro do intervalo de deformações conhecidas, já para níveis de deformação fora do intervalo conhecido, uma função polinomial não confere a mesma precisão.

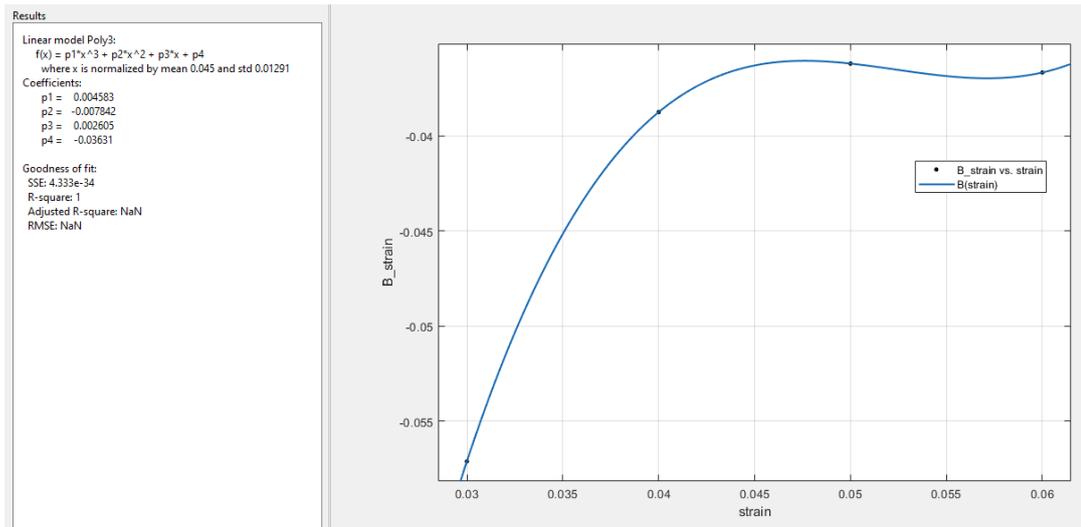


Figura 19 – Ajuste de curva para determinar a constante $B(\varepsilon)$.

6 IMPLEMENTAÇÃO NUMÉRICA

Para implementar numericamente todos os modelos apresentados neste trabalho, foram criadas três classes principais: a primeira, responsável por conter as equações constitutivas do modelo; a segunda, responsável por calcular os resultados do modelo; e a terceira, responsável por fazer análise de sensibilidade para um conjunto de constantes do modelo. Isso foi feito com o objetivo de evitar repetições de código e torná-lo mais legível e de fácil manutenção, de modo a atender os padrões de desenvolvimento de software citados na introdução deste documento, sendo o principal conceito o de responsabilidade única, um dos princípios SOLID para programação orientada a objetos, em que cada classe deve ter somente uma responsabilidade. Neste tópico, para facilitar suas citações, a primeira classe será chamada de classe modelo e as demais de operação.

Para todos os modelos de viscoelasticidade, é necessário calcular a deformação, a tensão e, para relaxação, a função relaxação e a derivada da deformação ou, para fluência, a função fluência e a derivada da tensão. Sendo assim, foi estabelecido grau de herança entre os modelos, conforme apresentada no fluxograma da Fig. 20, para se evitar repetições de código, sendo necessário implementar somente as lógicas específicas de cada modelo. Por exemplo, a classe modelo de Schapery herda da classe modelo não-linear que herda da classe modelo genérico, isto também é feito para as operações.

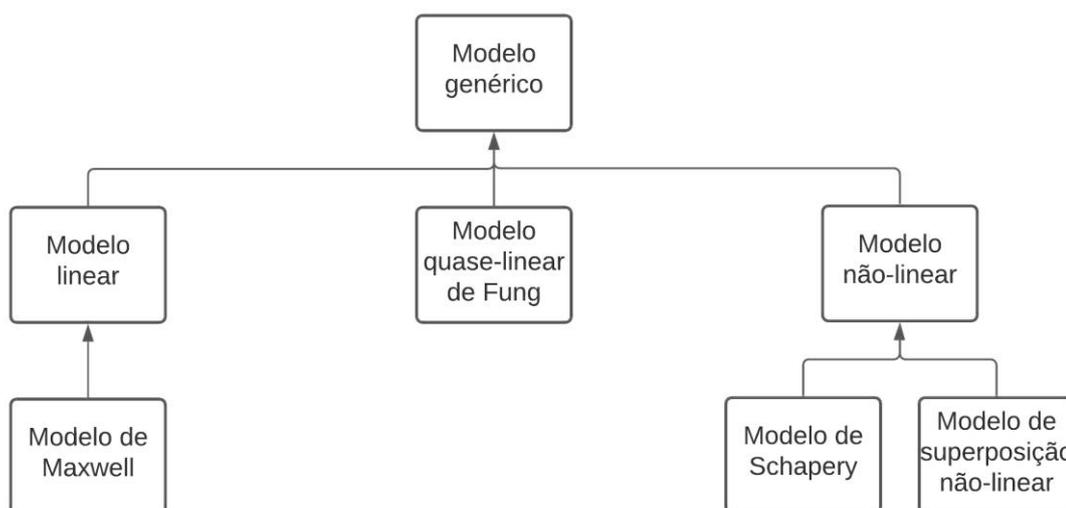


Figura 20 – Fluxograma indicando a ordem de herança entre as classes modelo.

No modelo genérico, foi implementado um método para calcular cada um dos seguintes parâmetros: a tensão e sua derivada; a deformação e sua derivada; a função relaxação; e a função fluência. Além disso, também foi criado um método denominado *CalculateResults*, representado na Fig. 21 pela sub-rotina “Calcula Resultados”, que funciona como um orquestrador, em que sua única inteligência é executar em paralelo os métodos que calculam os principais parâmetros do modelo e retorná-los em uma classe. Sendo assim, este método não compreende como cada parâmetro deve ser calculado nem a sequência de cálculo. Por exemplo, para o caso de relaxação, os métodos para calcular a tensão e função relaxação são chamados pelo método *CalculateResults*, porém a lógica de como e quando a função relaxação deve ser calculada para se obter o valor da tensão, está somente no método que calcula a tensão. Desse modo, ao implementar um novo modelo, basta implementar suas equações específicas e, caso necessário, alterar o comportamento do método *CalculateResults*. Ademais, vale salientar, que a classe modelo genérica foi estruturada de maneira que se adeque tanto para análises de fluência quanto para relaxação, conforme o fluxograma apresentado na Fig. 21. Além disso, como os modelos quase-linear e não-linear podem ou não considerar o tempo de rampa, foi adicionado um *if* dentro do método que calcula a tensão para que equações diferentes fossem utilizadas conforme apresentado anteriormente.

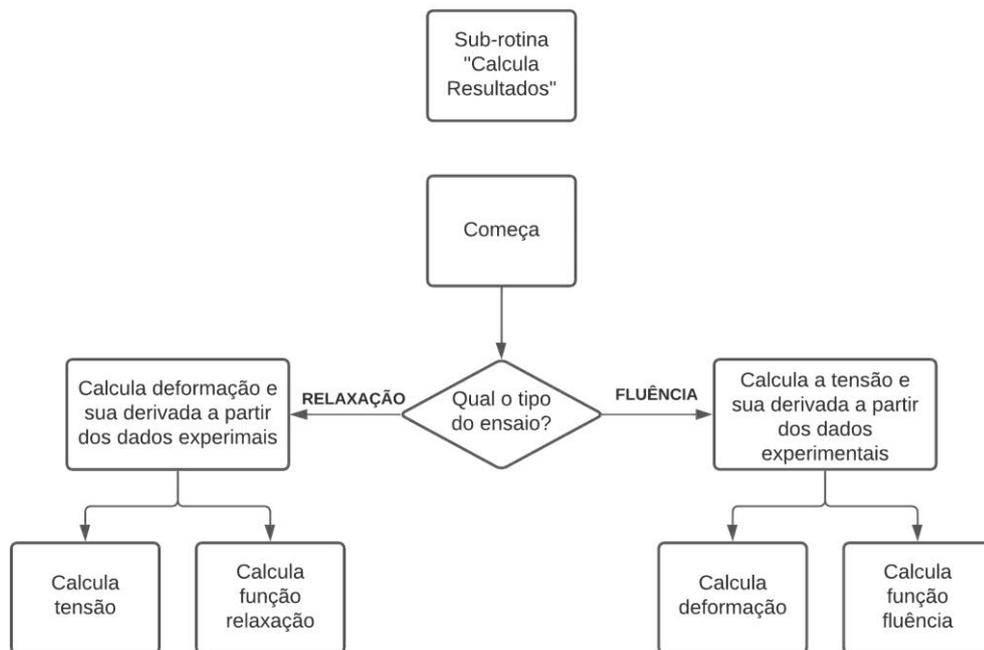


Figura 21 – Fluxograma para a sub-rotina “Calcula Resultados”

As operações responsáveis por calcular os resultados para o modelo e por fazer análise de sensibilidade das variáveis foram desenvolvidas de modo que possam ser utilizadas em todos os modelos. Sendo assim, foi implementado uma classe abstrata para cada operação, que possuem as lógicas apresentadas nos fluxogramas presentes nas Figuras 22a e 22b, e uma classe concreta para cada modelo que herda desta classe abstrata e implementa somente como os dados de entrada que devem ser construídos e especifique qual classe modelo deve ser utilizada. Vale salientar que, para facilitar a implementação, ambas as operações vão se comunicar com a classe modelo através da sub-rotina “Calcula resultados”.

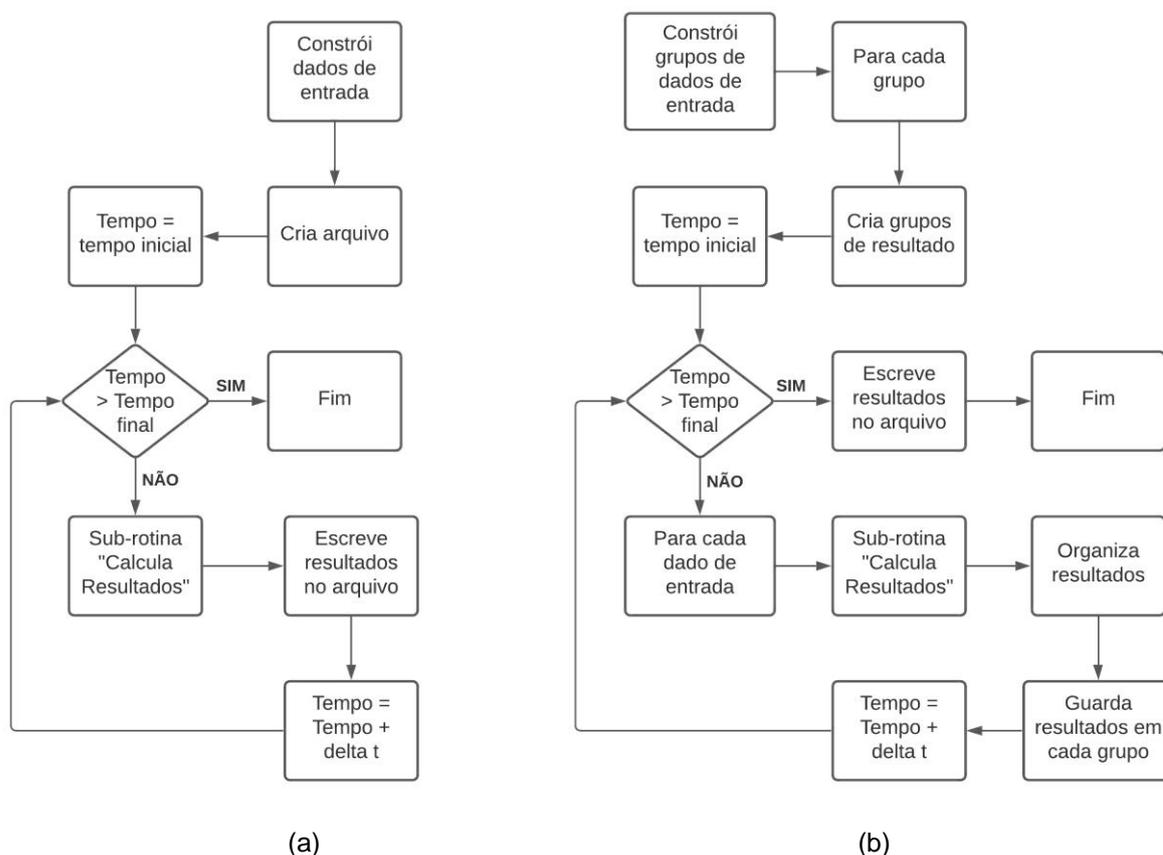


Figura 22 – Fluxograma das operações de (a) calcular os resultados do modelo e (b) fazer análise de sensibilidade das variáveis.

Conforme pode ser visto nos fluxogramas da Fig. 22, é necessário construir a os dados de entrada a serem utilizados na sub-rotina “Calcula resultados” a partir do *request*. Isso foi feito para seguir o princípio de responsabilidade única apresentado pelo padrão SOLID, já que os contratos das operações devem ser manipulados somente por estas e classes externas não devem ter conhecimento sobre estes

contratos. Posto isso, fez-se necessário criar classes que atuariam como contratos para as classes modelos, em que existe uma responsável por conter os dados de entrada e outra, os resultados dos cálculos. Desta forma, pode-se observar o motivo da etapa de construir os dados de entrada, em que o *request* da operação é mapeado para o contrato da classe modelo. Mesmo isso conferindo maior complexidade ao código, essa decisão facilitou as manutenções necessárias, além de ter permitido escalabilidade do código e adição de recursos com grande facilidade.

Na Fig. 22b, na primeira parte do fluxograma, é construído uma lista de *InputGroup*, essa classe é responsável por armazenar uma lista de dados de entrada para o modelo, em que somente uma variável é iterada para ser submetida à análise de sensibilidade, e um identificador para essa lista. Na etapa “Para cada grupo”, é utilizada a estrutura *foreach* em que é feita uma iteração em cada elemento dentro da lista de *InputGroup*. Na etapa “Cria grupos de resultados”, é criada uma lista de *ResultGroup*, esta classe é responsável por armazenar os resultados de um determinado parâmetro do modelo (tensão, deformação, função relaxação, entre outros) para serem escritos no arquivo. Na etapa “Para cada dado de entrada”, é utilizada a estrutura *foreach* para iterar em cada elemento dentro da lista de dados de entrada. Na etapa “Organiza resultados”, os resultados para cada classe de dados de entrada são separados de acordo com cada parâmetro calculado, para serem adicionados em um *ResultGroup* diferente na etapa “Guarda resultados em cada grupo”.

Conforme pode ser observado nos fluxogramas da Fig. 22, durante a etapa “Escreve resultados no arquivo”, os resultados são escritos em um arquivo no formato CSV, para que este possa ser utilizado em algum outro software de edição gráfica, como o OriginPro. Entretanto, em cada operação os dados de saída são escritos de maneiras diferentes no arquivo. Para a operação que gera os resultados para um determinado modelo de viscoelasticidade, os resultados são escritos em um único arquivo, já para a operação que faz análise de sensibilidade das variáveis, os resultados são escritos em um arquivo para cada parâmetro calculado, em que cada coluna corresponde um conjunto de dados de entrada diferente.

6.1 MODELO LINEAR

Para o modelo linear de viscoelasticidade, toda a implementação da classe modelo genérico foi reaproveitada, sendo necessário implementar somente as equações referente ao modelo e definir como os dados de entrada são construídos nas operações. Neste caso, os dados de entrada são a viscosidade e a rigidez do material, além da tensão inicial, para o caso de fluência, e da deformação inicial, para o caso de relaxação. Já os resultados de saída são os mesmos da classe modelo genérico.

6.2 MODELO QUASE-LINEAR

Os modelos quase-linear apresentam maior complexidade na estruturação dos dados de entrada, uma vez que, ao considerar ou não o tempo de rampa, são passados parâmetros diferentes, logo, a etapa de construir os dados de entrada, visto na Fig. 21, é alterada para cada consideração para que seja possível mapear esses valores corretamente.

Ao considerar o tempo de rampa, é necessário passar no *request* da operação as constantes A e B para que seja possível calcular a resposta elástica, entretanto, ao desconsiderar, é necessário informar somente a tensão inicial, em que será considerado que este é o valor da resposta elástica e esta é constante por todo o domínio de tempo. Como para calcular a função relaxação reduzida há duas equações (40 e 43) e cada uma necessita de constantes distintas, foi criado uma classe modelo para uma delas, a primeira chamada de *FungModel* e a outra, de *SimplifiedFungModel* com contratos distintos para cada uma.

Foi necessário implementar outros métodos específicos, sendo um para calcular cada um dos seguintes parâmetros: resposta elástica e sua derivada; e tensão, sendo um método para cada equação apresentada anteriormente (34, 35 e 36). Também foi necessário alterar o comportamento do método "*CalculateResults*" para retornar, também, a resposta elástica e a tensão calculada através das equações 35 e 36, quando é considerado o tempo de rampa.

6.3 MODELO NÃO-LINEAR

Para os modelos não-lineares de viscoelasticidade, toda a implementação da classe modelo genérico foi reaproveitada, sendo necessário implementar somente as equações referentes a cada modelo e definir como os dados de entrada são construídos nas operações.

6.3.1 Schapery

Assim como o modelo de Fung, para o modelo de Schapery há duas formas de se calcular a função relaxação, sendo criadas as classes modelos *SimplifiedSchaperyModel*, que usa a equação 50, e *SchaperyModel*, que usa a equação 51. Além disso, os demais dados de entrada são as constantes a e b utilizadas para se calcular h_e e h_2 , conforme apresentado anteriormente.

6.3.2 Método de superposição não-linear

Para o modelo de superposição não-linear, os dados de entrada são as constantes utilizadas para se calcular os parâmetros $A(\varepsilon)$ e $B(\varepsilon)$ utilizados na função relaxação.

7 ENSAIOS EXPERIMENTAIS

Conforme apresentado inicialmente, este estudo foi desenvolvido com o intuito de auxiliar a pesquisa desenvolvida pelo professor orientador deste trabalho. Sendo assim, foram utilizados os dados do ensaio experimental de relaxação para o ligamento cruzado anterior (LCA) de joelho de porco, que foram obtidos pelo professor orientador e os colaboradores da pesquisa. Inicialmente, para preparar o ligamento para o ensaio, foram feitos carregamentos cíclicos oscilando a deformação de 0 até 2%. Em seguida, variou-se a deformação conforme apresentado na Fig. 11 e equação 29, com os patamares de deformação iguais a 3, 4, 5 e 6%, valor mínimo igual a 2% e taxa de deformação igual a, aproximadamente, 1,5%/s. Além disso, o tempo que os ligamentos foram mantidos relaxando e se recuperando foram iguais a 400 segundos, com exceção da última relaxação que durou 600 segundos.

O resultado obtido para a curva de tensão versus tempo para o ligamento está apresentado na Fig. 23.

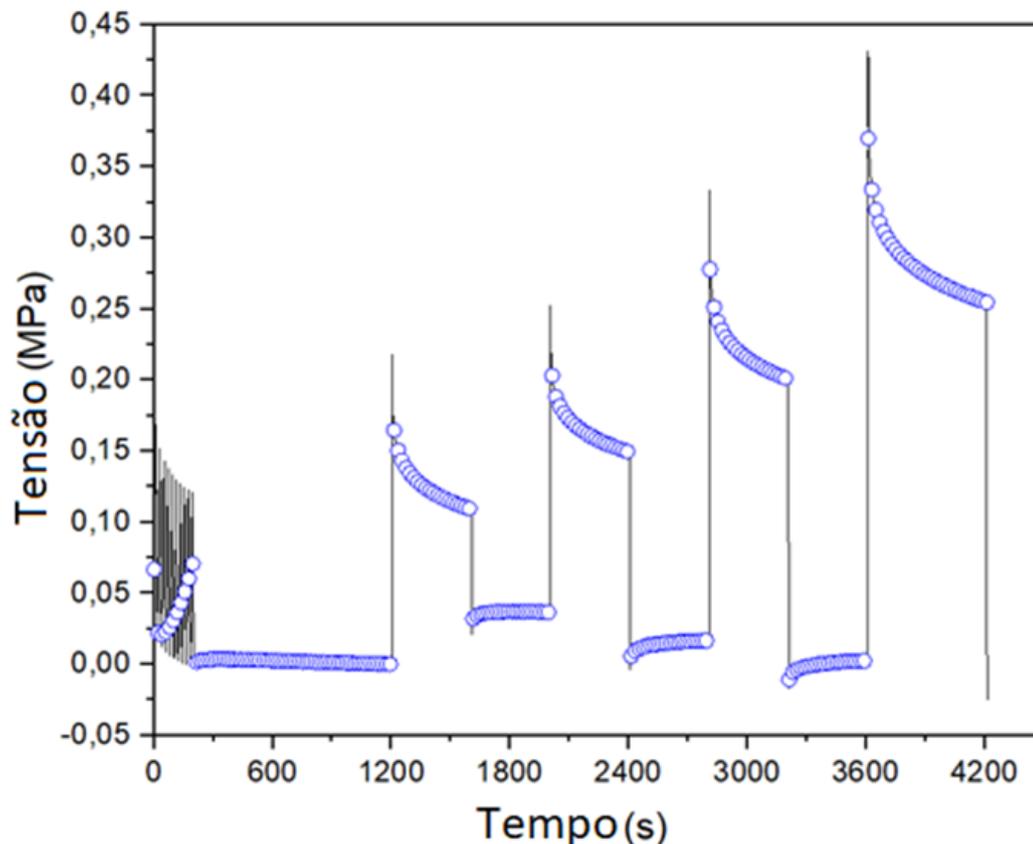


Figura 23 – Gráfico de tensão versus tempo obtido experimentalmente para o LCA

8 RESULTADOS E DISCUSSÕES

Neste tópico, serão abordados os resultados obtidos numericamente para o modelo de Fung e Schapery, além da comparação entre estes e com os dados experimentais.

8.1 MODELO DE FUNG

Para o modelo de Fung, conforme citado anteriormente, duas considerações foram feitas: usar o tempo de rampa ou não. Gerou-se resultados numéricos para cada consideração. Para Fung diferentes constantes são obtidas para diferentes graus de deformação conforme apresentado na Tabela 2. Vale salientar que as comparações entre os resultados numéricos e experimentais foram feitas para 5% de deformação. Foi feito somente para um nível de deformação, pois, para validar o modelo de Fung, isto é já é suficiente.

Tabela 2 – Constantes de Fung para o ligamento cruzado anterior (LCA)

Constantes	3 % de Deformação	4 % de Deformação	5 % de Deformação	6 % de Deformação
A (MPa)	0,1999	0,0128	0,01968	0,0287
B	24,81	76,49	58,04	46,58
C_0	0,4962	0,5875	0,6011	0,5885
C_1	0,1163	0,0986	0,1023	0,1046
C_2	0,1285	0,1009	0,1040	0,1063
C_3	0,2346	0,1910	0,1710	0,1761
τ_1	1,3780	1,2620	1,3845	1,8587
τ_2	17,03	16,42	17,27	23,66
τ_3	192,94	209,20	199,0	290,70

Fonte: Autor

Para o modelo de Fung, quando não é considerado o tempo de rampa, os resultados coincidiram com os dados experimentais, conforme apresentado na Fig. 24.

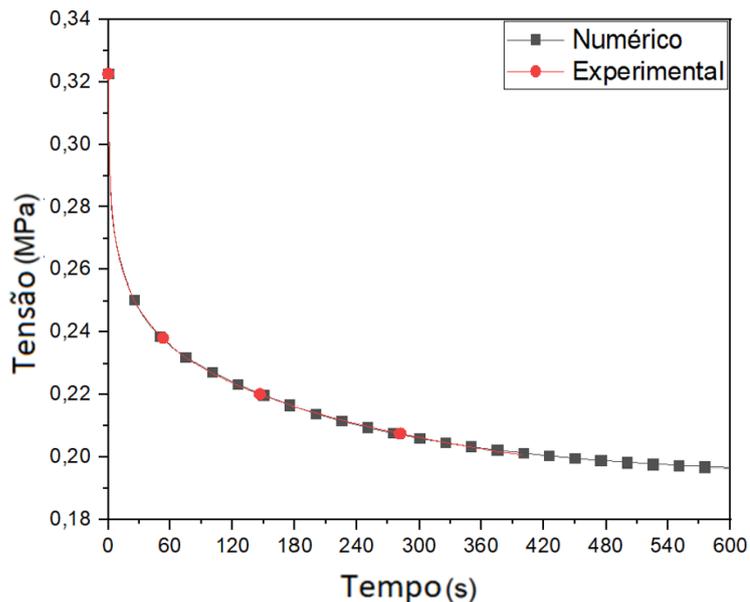


Figura 24 – Gráfico de tensão versus tempo obtido numericamente com modelo de Fung desconsiderando o tempo de rampa comparando com curva experimental

Ao considerar o tempo de rampa, os resultados numéricos apresentaram certa imprecisão quando comparados com os dados experimentais, conforme pode ser vista na Fig. 25. Entretanto, essa imprecisão é pequena para a relaxação e é mais evidente para a fase de recuperação. Com isso, pode-se mostrar que o modelo pode ser utilizado para ensaios de relaxação em que não levam em consideração a recuperação do material, evidenciando mais uma limitação do modelo.

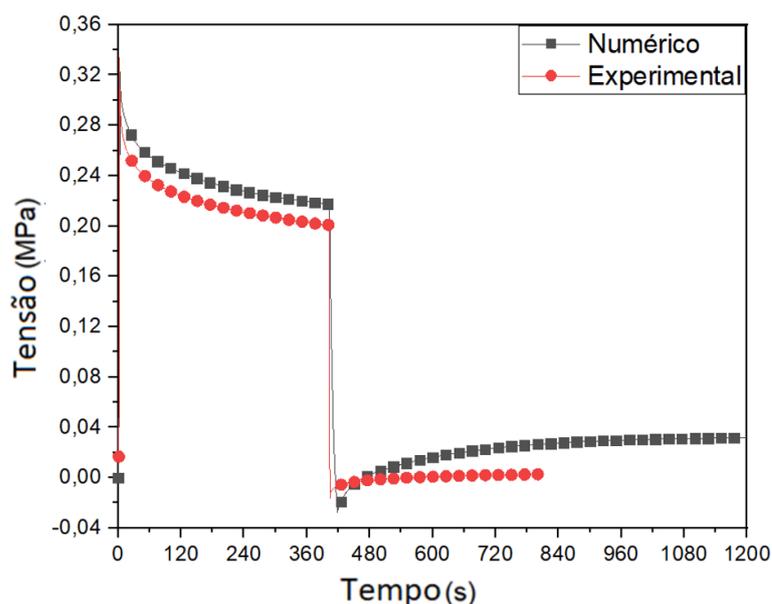


Figura 25 – Gráfico de tensão versus tempo obtido numericamente com modelo de Fung considerando o tempo de rampa comparando com curva experimental

Vale lembrar, que o modelo de Fung é aplicado para um patamar de deformação em específico e tendo sido desenvolvido para que os valores das constantes sejam aplicáveis somente para aquele valor de deformação. Logo, não deve-se utilizar estes valores para outros patamares de deformação, havendo grandes divergências conforme apresentado na Fig. 26.

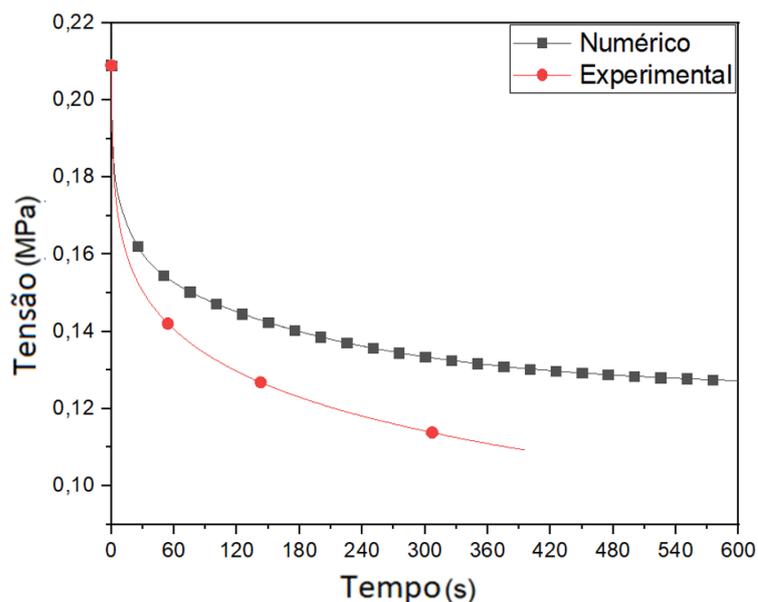


Figura 26 – Gráfico de tensão versus tempo para 3% de deformação usando constantes obtidos para 5% de deformação

8.2 MODELO DE SCHAPERY

Para o modelo de Schapery, anteriormente foram apresentadas duas formas de descrever a função relaxação, equações 51 e 50. Para a primeira equação, as constantes utilizadas estão presentes na Tabela 3 e, para a segunda, na Tabela 4, se baseando em, respectivamente, em (PROVENZANO, 2002) e (BLANDFORD, 2017). Vale ressaltar que, como os métodos de obtenção das constantes são diferentes em alguns pontos, era de se esperar que fossem encontrados valores diferentes para as constantes.

Tabela 3 – Constantes para Função Relaxação Transiente usando equação 51

Constantes	(PROVENZANO, 2002)	(BLANDFORD, 2017)
G_e	9,121	9,121
C	-2,807	0,3458
n	-0,1119	0,8851

Fonte: Autor

Tabela 4 – Constantes para Função Relaxação Transiente usando equação 50

Constantes	Valor	Constantes	Valor
G_e	3,457	τ_1	1,3780
G_1	0,02431	τ_2	17,04
G_2	0,02683	τ_3	192,98
G_3	0,04901		

Fonte: Autor

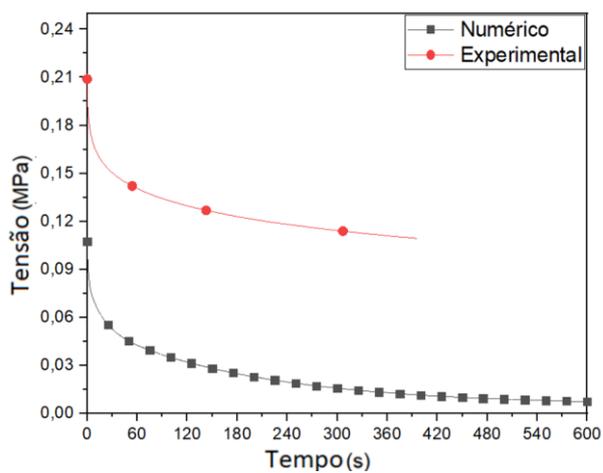
Conforme supracitado, (BLANDFORD, 2017) estabelece uma relação logarítmica e exponencial, respectivamente, entre os parâmetros h_e e h_2 , entretanto, isto não foi encontrado. Ao utilizar estas relações para fazer o ajuste de curva, foi encontrada muita divergência entre os dados, sendo assim, outras relações foram testadas. Para os parâmetros h_e e h_2 , foi encontrado uma relação exponencial com a deformação, sendo necessário dois termos para ambos. Na Tabela 5 é apresentado os valores de h_e e h_2 para cada deformação e quando é utilizada a equação 50 ou a 51, obtida tanto para em (PROVENZANO, 2002) quanto para (BLANDFORD, 2017).

Tabela 5 – Valores de h_e e h_2 para cada nível de deformação

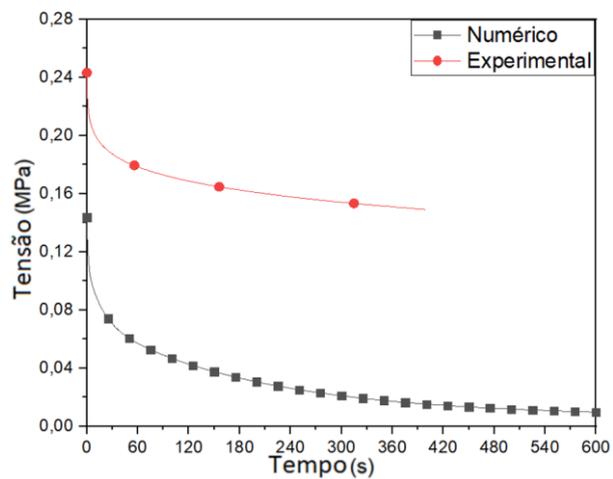
Deformação	Série de Prony		Lei de potência - (PROVENZANO, 2002)		Lei de potência - (BLANDFORD, 2017).	
	h_e	h_2	h_e	h_2	h_e	h_2
3%	1	33,24	1	1	0,4313	53,74
4%	1,396	30,36	0,8205	0,6824	0,5762	35,07
5%	1,872	38,1	0,8497	0,6822	0,7664	36,42
6%	2,474	49,05	0,9169	0,7177	0,9766	60,77

Fonte: Autor

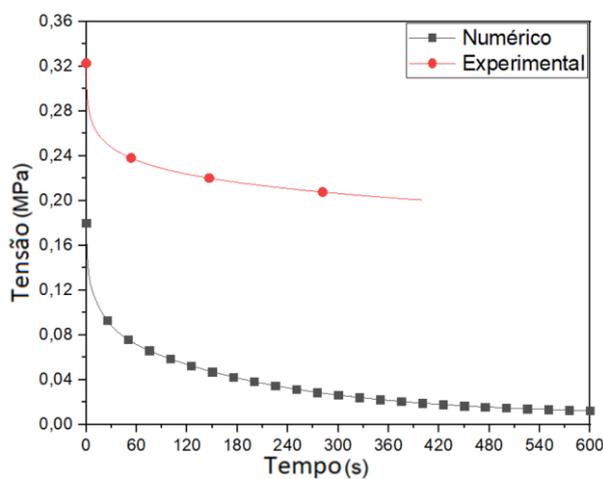
Com as constantes apresentadas anteriormente, foram gerados os gráficos presentes nas Figuras 27, 28 e 29 ao desconsiderar o tempo de rampa.



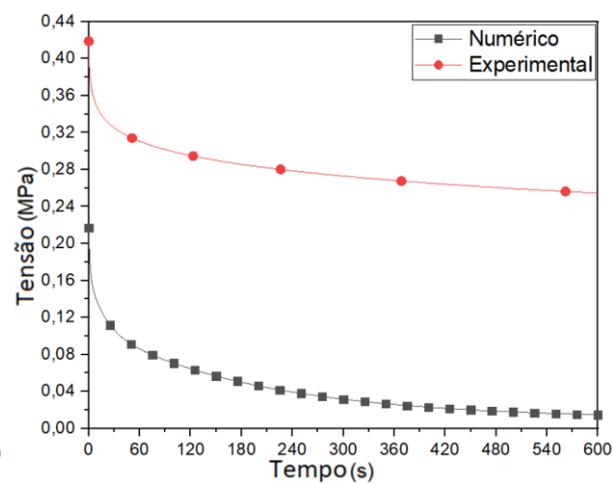
(a)



(b)



(c)



(d)

Figura 27 – Gráfico de tensão versus tempo para Schapery desconsiderando o tempo de rampa e usando série de Prony (equação 50) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.

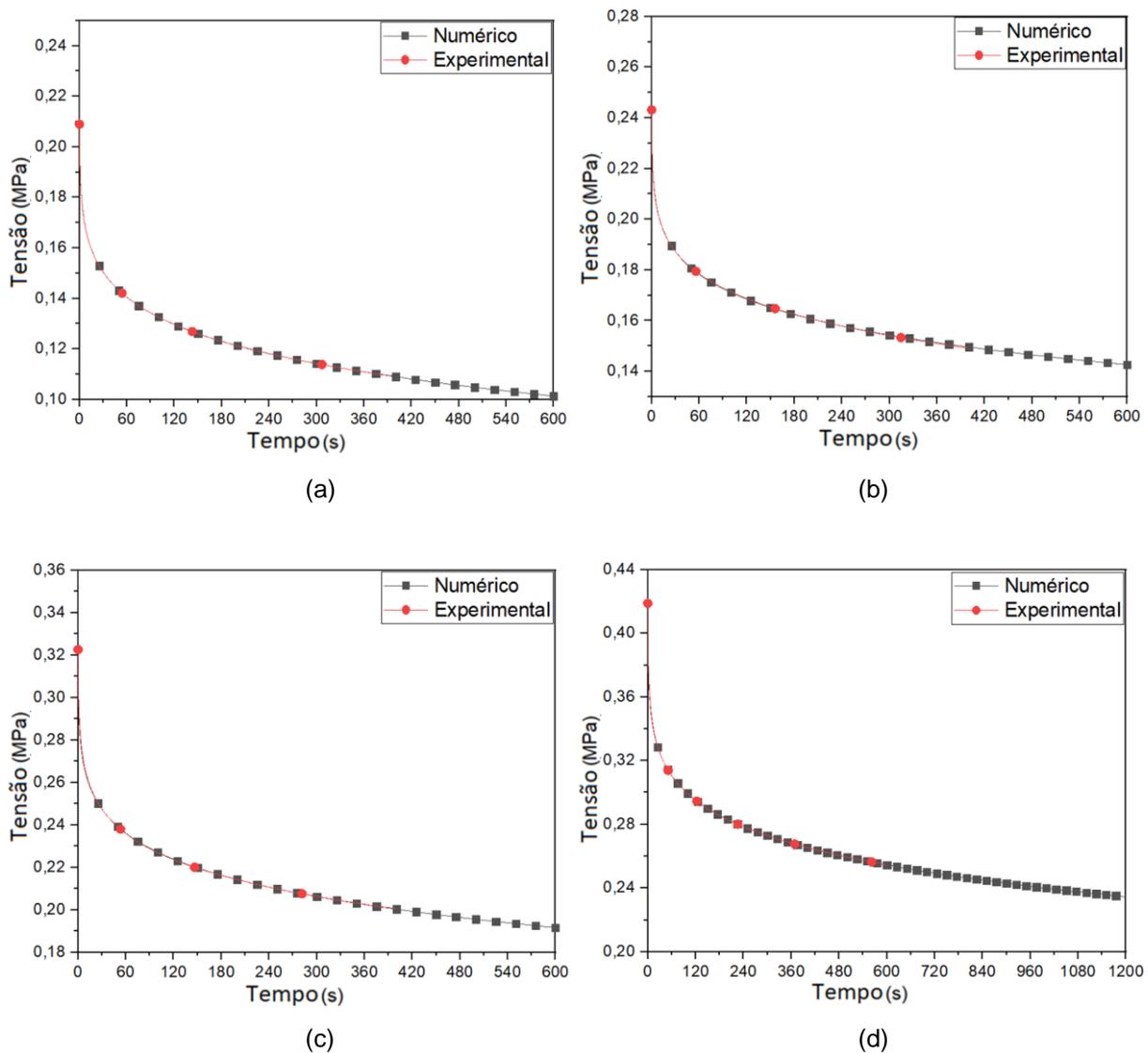


Figura 28 – Gráfico de tensão versus tempo para Schapery desconsiderando o tempo de rampa e usando lei da potência (equação 51) obtendo as constantes segundo (PROVENZANO, 2002) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.

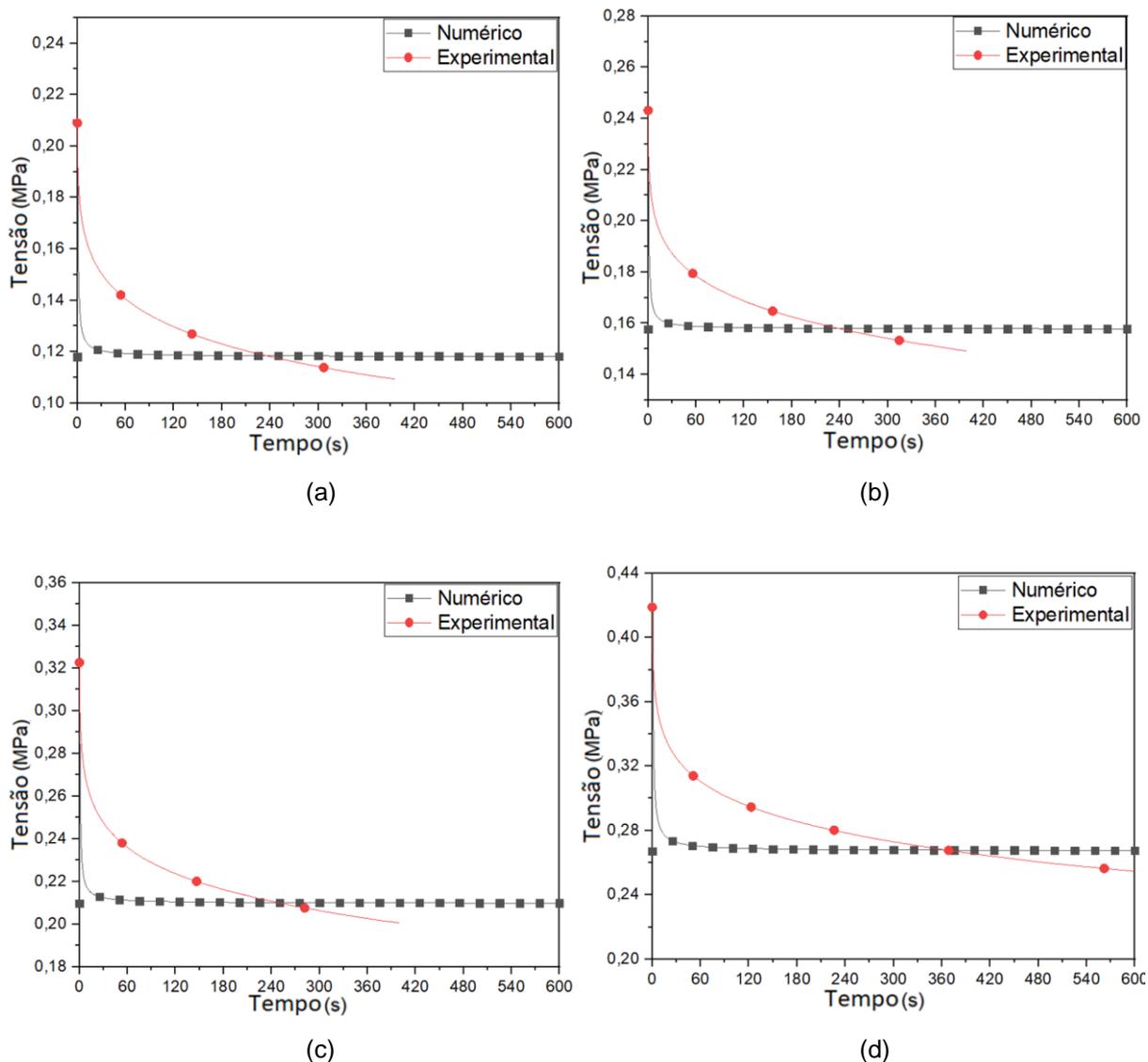
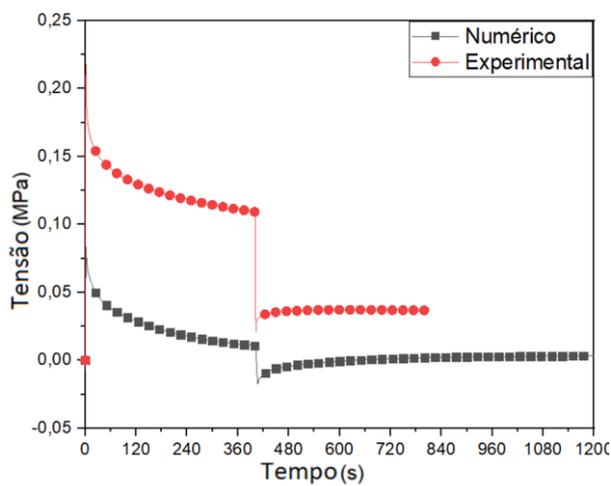


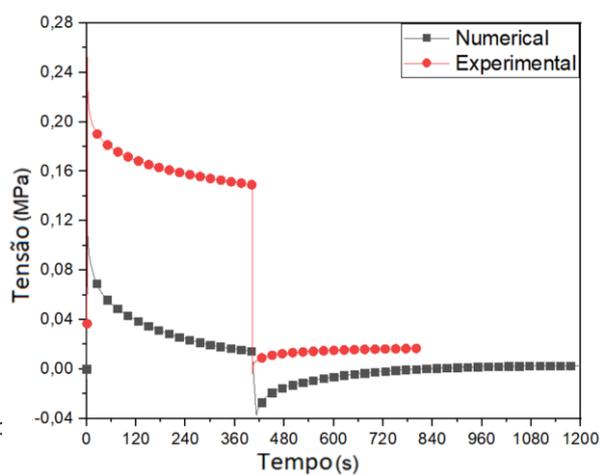
Figura 29 – Gráfico de tensão versus tempo para Schapery desconsiderando o tempo de rampa e usando lei da potência (equação 51) obtendo as constantes segundo (BLANDFORD, 2017) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.

Como pode ser visto na Fig. 28, os resultados coincidiram com os dados experimentais, sendo possível validar estas soluções para o modelo de Schapery. Contudo, para as Figuras 27 e 29, foi observado divergência entre os dados, devido a forma como as constantes foram obtidas, indicando que, se basear em (BLANDFORD, 2017) e na série de Prony para a obtenção das constantes de Schapery para o ligamento cruzado anterior de joelho de porco, que foi analisado neste projeto final, não é uma solução viável.

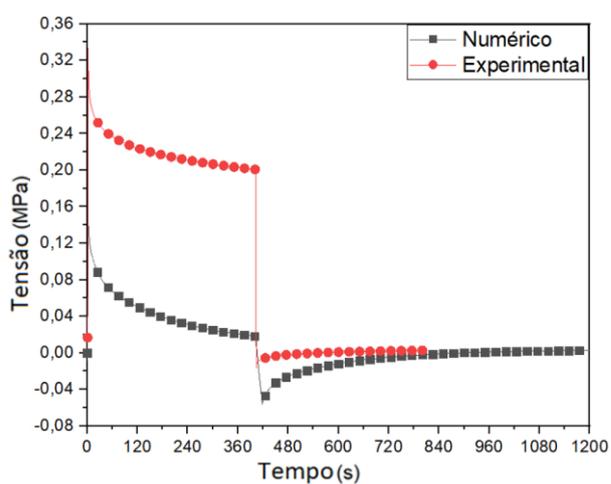
Ao considerar o tempo de rampa, foram obtidos os resultados presentes nas Figuras 30, 31 e 32.



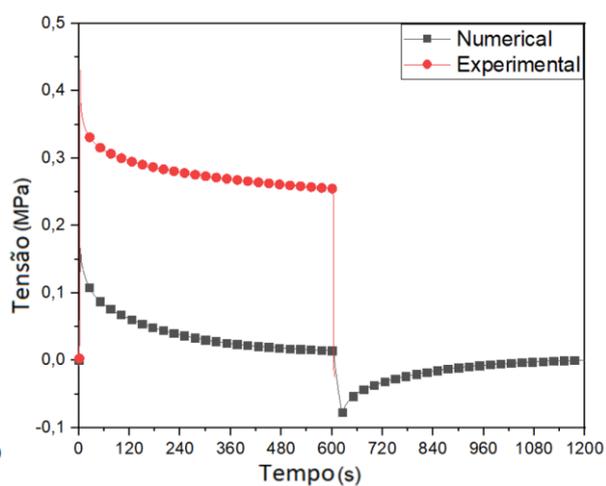
(a)



(b)

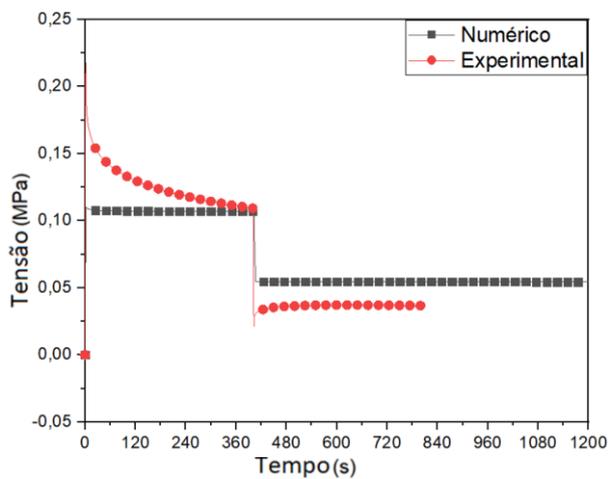


(c)

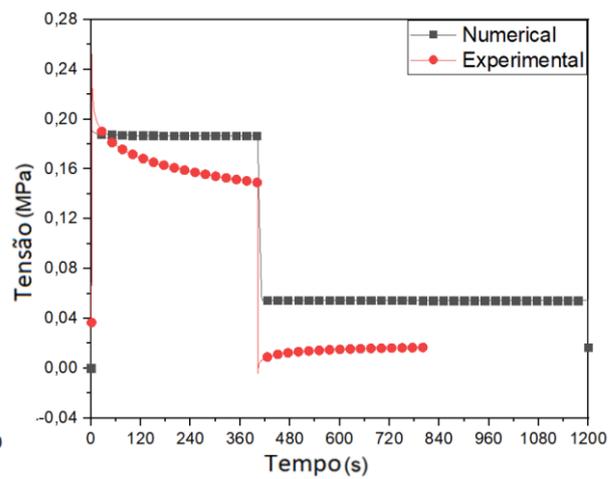


(d)

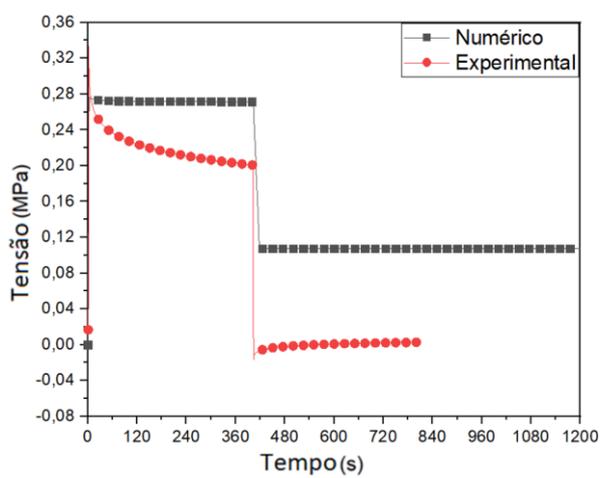
Figura 30 – Gráfico de tensão versus tempo para Schapery considerando o tempo de rampa e usando série de Prony (equação 50) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.



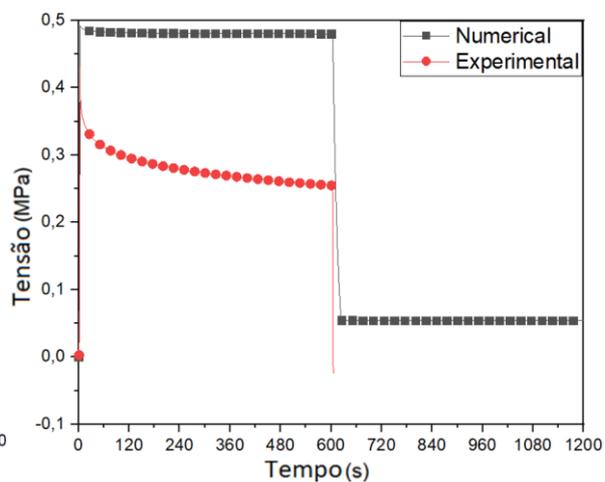
(a)



(b)



(c)



(d)

Figura 31 – Gráfico de tensão versus tempo para Schapery considerando o tempo de rampa e usando lei da potência (equação 51) obtendo as constantes segundo (PROVENZANO, 2002) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.

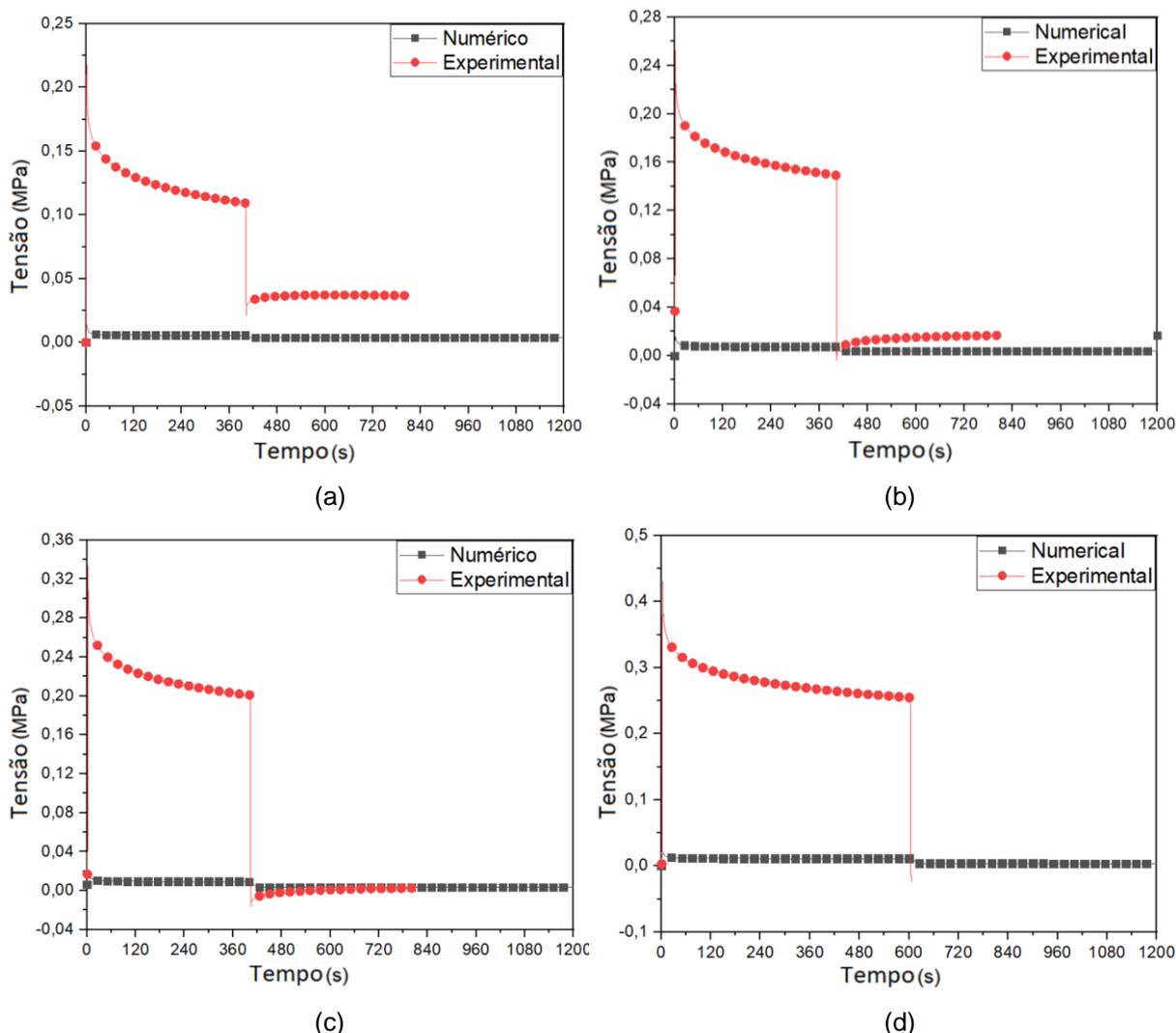


Figura 32 – Gráfico de tensão versus tempo para Schapery considerando o tempo de rampa e usando lei da potência (equação 51) obtendo as constantes segundo (BLANDFORD, 2017) para (a) 3%, (b) 4%, (c) 5% e (d) 6% de deformação.

Como pode ser observado nas Figuras 30, 31 e 32, os resultados obtidos usando o modelo de Schapery não foram satisfatórios ao considerar o tempo de rampa, assim como para o modelo de Fung. O principal fator para isto pode estar nas bibliografias em que este estudo foi baseado, cuja preocupação dos autores eram em descrever a evolução da tensão para o caso de relaxação para deformações constantes. Entretanto, vale salientar que uma possível solução para próximos trabalhos é tentar reescrever a equação principal do modelo atendendo cada caso, seja durante o tempo de subida, a relaxação, o tempo de descida e recuperação. Isso vai de encontro ao que foi feito por (PROVENZANO, 2002) e (BALDNFORD, 2017) que reescreveram as equações para o caso da relaxação para ambos os modelos não-lineares abordados neste trabalho.

Em suma, o presente trabalho alcançou seu objetivo principal que é o desenvolvimento de uma API REST para análises de viscoelasticidade em tecidos moles. Sendo possível validar os modelos de Fung e de Schapery e que as soluções propostas pelas bibliografias (ABRAMOWITCH *et al.*, 2004), (FUNG, 1993), (PROVENZANO, 2002) são viáveis somente durante a relaxação e os resultados obtidos numericamente realmente coincidem com os dados experimentais. Além disso, foram encontradas limitações em cada modelo, já que, para o modelo de Fung, as constantes obtidas em um determinado ensaio não podem ser utilizadas para diferentes patamares de deformação e, para o modelo de Schapery, a equação principal do modelo precisou ser reescrita para que os resultados numéricos aproximassem mais dos resultados experimentais durante a relaxação.

9 TRABALHOS FUTUROS

Como trabalhos futuros podem ser destacados os seguintes tópicos:

- Obter as constantes para a equação 37 da função relaxação reduzida para o modelo quase-linear de Fung;
- Fazer análise de sensibilidade para as constantes utilizadas nos modelos de Fung, Schapery e Método de Superposição Não-Linear;
- Analisar se é possível reescrever a equação de Schapery para a tensão para ser usada durante o tempo de rampa, o tempo de descida e a recuperação;
- Procurar bibliografias que tratem sobre o tempo de rampa, tempo de descida e recuperação para ensaios de relaxação que apliquem o modelo de Schapery e Método de Superposição Não-Linear;
- Investigar com mais detalhes o motivo de não serem encontrados resultados satisfatórios ao utilizar a série de Prony e ao se basear em (BLANDFORD, 2017);
- Utilizar forma tensorial das equações;
- Analisar a influência do número de relaxações nos resultados;
- Iniciar estudo sobre fluência;
- Implementar ao código as lógicas e equações para fluência.

REFERÊNCIAS

ABRAMOWITZ, Steven D. e WOO, Savio L.-Y. An Improved Method to Analyze the Stress Relaxation of Ligaments Following a Finite Ramp Time Based on the Quasi-Linear Viscoelastic Theory. *JOURNAL OF BIOMECHANICAL ENGINEERING*. 2004. **ASME**. Vol. 126. P. 92-97. DOI: 10.1115/1.1645528.

Andrade, G., Santana, I. L., Takiya, C. **O conceito da energia livre de helmholtz em livros didáticos e uma nova abordagem de sua definição**. 2020. DOI: 10.51162/brc.dev2020-00040.

BABAEI, Behzad, *et al.* **Efficient and optimized identification of generalized Maxwell viscoelastic relaxation spectra**. 2015 Disponível em: <http://dx.doi.org/10.1016/j.jmbbm.2015.10.008>. Acessado em: 6 maio 2021.

BERNARDES, C., *et al.* **Determinação de parâmetros biomecânicos para o modelamento da articulação do joelho**. 2005. Laboratório de Pesquisa do Exercício, Universidade Federal do Rio Grande do Sul. Porto Alegre, 2005.

BLANDFORD, C. **Nonlinear visco-elastic models applied to human medial collateral ligaments**. 2017. CMBEC40 Conference. Department of Mechanical and Materials Engineering, Faculty of Engineering the University of Western Ontario, London, ON.

BONIFASI-LISTA, Carlos; LAKE, Spencer P.; SMALL, Michael S.; WEISS, Jeffrey A. **Viscoelastic properties of the human medial collateral ligament under longitudinal, transverse and shear loading**. 2004. University of Utah. Salt Lake City, Utah, USA.

COMPOSITE CONSTRUCTION. 2020. Disponível em: <http://fgg-web.fgg.uni-lj.si/~pmoze/ESDEP/master/wg10/l0520.htm>. Acessado em: 6 maio 2021.

DE BARROS, Stephanie A. S. **Research on viscoelasticity models for human knee ligaments**. Relatório final de iniciação científica. Rio de Janeiro. 2020.

DE CAMARGO, Wellington Balbo. **Conceitos e Exemplos – Herança: Programação Orientada a Objetos – Parte 1**. DEVMEDIA. 2010. Disponível em: <https://www.devmedia.com.br/conceitos-e-exemplos-heranca-programacao-orientada-a-objetos-parte-1/18579>. Acessado em 23 maio 2021.

DE PASCALIS, R.; ABRAHAMS, I.D.; PARNELL, W.J.. **On nonlinear viscoelastic deformations: a reappraisal of Fung's quasi-linear viscoelastic model**. 2014. Proc. R. Soc. A 470: 20140058. Disponível em: <http://dx.doi.org/10.1098/rspa.2014.0058>. Acessado em 4 maio 2021.

DEBSKI, R. E; ABRAMOWITCH, S. D.; WOO, S. L.-Y.; CLINEFF, T. D. **An evaluation of the quasi-linear viscoelastic properties of the healing medial collateral ligament in a goat model**. 2004. Annals of Biomedical Engineering.

DUENWALD, *et al.* **Constitutive equations for ligament and Other soft tissue: evaluation by experiment**. 2009. University of Wisconsin. Madison, Wisconsin, USA. DOI: 10.1007/s00707-009-0161-8.

DUENWALD, S. E.; JR., R. V.; LAKES, R. S. **Viscoelastic Relaxation and Recovery of Tendon**. 2009. University of Wisconsin-Madison. Madison, USA.

Findley, W.N.; Lai, J.S.; Onaran, K. **Creep and relaxation of nonlinear viscoelastic materials**. 1976. Dover, New York.

Fung, Y. **Biomechanics: Mechanical Properties of Living Tissues**. 1993. Springer New York, University of Michigan.

GASPAROTTO, H. M. **Introdução ao ASP.NET MVC**. DEVMEDIA. 2014. Disponível em: devmedia.com.br/introducao-ao-asp-net-mvc/31878. Acessado em: 29 abr. 2021.

HAIJ-ALI, R. M.; MULIANA, A. H. **A micromechanical constitutive framework for the nonlinear viscoelastic behavior of pultruded composite materials**. 2002. DOI: 10.1016/s0020-7683(02)00663-7.

HOSTGATOR. **API Restful: conceito, princípios e como criar**. 2019. Disponível em: <https://www.hostgator.com.br/blog/api-restful/>. Acessado em 29 abr. 2021.

INTRODUCING JSON. Disponível em: <https://www.json.org/json-en.html>. Acessado em: 29 abr. 2021.

JAMISON, C. E.; MARANGONI, R. D.; GLASER, A. A. **Viscoelastic properties of soft tissue by discrete model characterization**. 1968. Journal of Engineering for Industry.

Lakes, R.S. **Viscoelastic solids**. 1998. CRC Press, New York.

MICROSOFT. **APIS +MICROSERVICES: The ultimate guide to creating an enterprise API platform**. 2019. Disponível em: https://azure.microsoft.com/mediahandler/files/resourcefiles/apis-microservices-ebook/Azure_API-Microservices_eBook.pdf. Acesso em: 27 abr. 2021.

MICROSOFT. **Task Classe**. 2021. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/api/system.threading.tasks.task?view=net-5.0>. Acessado em: 4 maio 2021.

MICROSOFT. **Um tour pela linguagem C#**. 2021. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>. Acesso em 27 abr. 2021.

MILLER, Karol. **Constitutive model of brain tissue suitable for finite element analysis of surgical procedures**. 1998. Department of Mechanical and Materials Engineering, The University of Western Australia, Nedlands/Perth WA 6907, Australia.

MILVUS. **O que é um endpoint: Como proteger este tipo de dispositivo**. 2020. Disponível em: <https://milvus.com.br/o-que-e-endpoint/>. Acessado em: 29 abr. 2021.

NAVARRO, R. F. **Modelos Viscoelásticos Aplicáveis a Materiais Reais: uma Revisão**. 2017. Universidade Federal de Campina Grande, Unidade Acadêmica de Engenharia de Materiais, Laboratório CERTBIO. Bodocongó, Campina Grande, PB.

PIAZZA, Stephen J.; DELP, Scott L. Three-Dimensional Dynamic Simulation of Total Knee Replacement Motion During a Step-Up Task. JOURNAL OF BIOMECHANICAL ENGINEERING. 2001. **ASME**. Vol. 123. P. 599-606. DOI: 10.1115/1.1406950.

POWER LAW (Ostwald-deWaele model). 2007. In: Gooch J.W. (eds) **Encyclopedic Dictionary of Polymers**. Springer, New York, NY. Disponível em: https://doi.org/10.1007/978-0-387-30160-0_9193. Acessado em 26 set. 2021.

PROVENZANO, P. P.; LAKES, R. S.; CORR, D. T., JR, R. V. **Application of nonlinear viscoelastic models to describe ligament behavior**. 2002. DOI: 10.1007/s10237-002-0004-1.

QUEIROZ, José Aparecido Silva de. **Análise de estruturas flexíveis com aplicação de materiais viscoelásticos**. 2008. 101 f. Dissertação (mestrado) - Universidade Estadual Paulista, Faculdade de Engenharia de Bauru, 2008. Disponível em: <https://repositorioslatinoamericanos.uchile.cl/handle/2250/2568006>. Acessado em 4 maio 2021.

RAMO, N. L.; PUTTLITZ, C. M.; TROYER, K. L. **The development and validation of a numerical integration method for non-linear viscoelastic modeling**. 2018. Disponível em: <https://doi.org/10.1371/journal.pone.0190137>. Acessado em 24 set. 2021.

REDHAT. **O que é API REST?** Disponível em: <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>. Acessado em: 29 abr. 2021.

REGRA DE SIMPSON. Instituto Superior Técnico de Lisboa. Disponível em: <https://www.math.tecnico.ulisboa.pt/~calves/courses/integra/capiii33.html#:~:text=Regra%20de%20Simpson%20aplicada%20a%20dois%20sub%2Dintervalos.&text=Assim%2C%20podemos%20considerar%20tr%C3%AAs%20n%C3%B3s,cada%20um%20destes%20sub%2Dintervalos>. Acessado em: 18 maio 2021.

REGRAS COMPOSTAS. Universidade Federal do Rio Grande do Sul. Disponível em: https://www.ufrgs.br/reatmat/CalculoNumerico/livro-oct/in-regras_compostas.html. Acessado em 18 maio 2021.

RICK ANDERSON. **Visão geral do ASP.NET**. Microsoft. 2019. Disponível em: <https://docs.microsoft.com/pt-br/aspnet/overview>. Acessado em: 29 abr. 2021.

ROSSETTO, N. P. **A viscosidade no alongamento de tendões**. 2009. Trabalho de conclusão de curso em Educação Física – Universidade Estadual de Campinas. Campinas, 2009.

RUNSCOPE. **HTTP Status Codes**. Disponível em: <https://httpstatuses.com/>. Acessado em: 30 abr. 2021.

SCHAPERY, R. A. **On the characterization of nonlinear viscoelastic materials.** 1969. Disponível em: <https://doi.org/10.1002/pen.760090410>. Acessado em 24 set. 2021.

SCHAPERY, R. A. **Nonlinear Viscoelastic and Viscoplastic Constitutive Equations Based on Thermodynamics.** 1997. *Mechanics of Time-Dependent Materials* 1, p. 209–240. Disponível em: <https://doi.org/10.1023/A:1009767812821>. Acessado em 24 set. 2021.

SCHAPERY, R. A. **Nonlinear viscoelastic solids.** *International Journal of Solids and Structures* Volume 37, Issues 1-2, p. 359-366. Disponível em: [https://doi.org/10.1016/S0020-7683\(99\)00099-2](https://doi.org/10.1016/S0020-7683(99)00099-2). Acessado em 24 set. 2021.

SMARTBEAR. **About Swagger.** 2021. Disponível em: <https://swagger.io/about/>. Acessado em: 29 abr. 2021.

STACK OVERFLOW. **How to export a Swagger JSON/YAML file from Swagger UI?** 2019. Disponível em: <https://stackoverflow.com/questions/48525546/how-to-export-a-swagger-json-yaml-file-from-swagger-ui>. Acesso em: 24 maio 2021.

SILVEIRA, Bruno M. **SoftTissue, Repositório do GitHub.** 2020. Disponível em: <https://github.com/M3110/SoftTissue>. Acessado em: 24 maio 2021.

TARECO, M. A. C. **Conceitos de viscoelasticidade na modelação da fluência em estruturas mistas aço-betão.** 2014 154f. Dissertação (Mestrado) – Engenharia Civil, Faculdade de Ciências e Tecnologia. Lisboa, 2014. Disponível em: https://run.unl.pt/bitstream/10362/12481/1/Tareco_2014.pdf. Acessado em 4 maio 2021.

TechTudo. **O que é API e para que serve? Cinco perguntas e respostas.** 2020. Disponível em: <https://www.techtudo.com.br/listas/2020/06/o-que-e-api-e-para-que-serve-cinco-perguntas-e-respostas.ghtml>. Acessado em: 27 abr. 2021.

WEINEROWSKA-BORDS, Katarzyna. **Alternative Approach to Convolution Term of Viscoelasticity in Equations of Unsteady Pipe Flow.** 2015. *Journal of biomechanical engineering*. ASME. Vol. 137. p. 054501-1 até 054501-8. DOI: 10.1115/1.4029573.

WEISS, Jeffrey A. e GARDINER, John C. **Computational Modeling of Ligament Mechanics**. 2001. University of Utah, Department of Bioengineering, Salt Lake City, Utah 84112.

WISMANS, J. S. H. M. **A three-dimensional mathematical model of the human knee joint Eindhoven**. 1980. Technische Hogeschool Eindhoven. DOI: 10.6100/IR155825.

XU, Qinwu e ENGQUIST, Björn. **A mathematical model for fitting and predicting relaxation modulus and simulating viscoelastic responses**. 2018. Disponível em: <https://doi.org/10.6084/m9.figshare.c.4088969>. Acessado em: 6 maio 2021.

ZHENG, N.; *et al.* **An analytical model of knee for estimation of internal forces during exercise**. 1998. American Sports Medicine Institute. Birmingham, Alabama, USA.

ANEXO A - Descrição das equações usadas para modelo de viscoelasticidade quase-linear

Ao desconsiderar o tempo de rampa, tem-se que a resposta elástica sempre será constante, logo, sua derivada é igual a zero.

$$\sigma(t) = \int_0^t G(t - \tau) \frac{d\sigma^e(\tau)}{d\tau} d\tau \quad (\text{repetida}) \quad (34)$$

Para esta equação, não será possível utilizá-la já que, como a resposta elástica é constante, sua derivada será igual a zero, logo, para todo o domínio de tempo, essa equação será igual a zero.

$$\sigma(t) = \sigma^e(t) + \int_0^t \sigma^e(t - \tau) \frac{dG(\tau)}{d\tau} d\tau \quad (\text{repetida}) \quad (35)$$

$$\sigma(t) = \sigma^e(t) + \int_0^t \sigma^e(t - \tau) \frac{dG(\tau)}{d\tau} d\tau = \sigma_0 + \int_0^t \sigma_0 \frac{dG(\tau)}{d\tau} d\tau = \sigma_0 + \sigma_0 \cdot \int_0^t \frac{dG(\tau)}{d\tau} d\tau$$

$$\sigma(t) = \sigma_0 \cdot \left[1 + \int_0^t \frac{dG(\tau)}{d\tau} d\tau \right] = \sigma_0 \cdot [1 + G(t) - G(0)] = \sigma_0 \cdot [1 + G(t) - 1]$$

$$\sigma(t) = \sigma_0 \cdot G(t) \quad (A1)$$

$$\sigma(t) = \frac{d}{dt} \int_0^t \sigma^e(t - \tau) G(\tau) d\tau \quad (\text{repetida}) \quad (36)$$

$$\sigma(t) = \frac{d}{dt} \int_0^t \sigma^e(t - \tau) G(\tau) d\tau = \frac{d}{dt} \int_0^t \sigma_0 \cdot G(\tau) d\tau = \sigma_0 \cdot \frac{d}{dt} \int_0^t G(\tau) d\tau$$

$$\sigma(t) = \sigma_0 \cdot \left[G(t) \cdot \frac{dt}{dt} - G(0) \cdot \frac{d0}{dt} \right]$$

$$\sigma(t) = \sigma_0 \cdot G(t) \quad (A1)$$