

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM**  
**COMPUTACIONAL**

**Igor Pires dos Santos**

**Simulação de Escoamento Pulsátil em Modelos de Árvores Arteriais**

Juiz de Fora

2021



**Igor Pires dos Santos**

**Simulação de Escoamento Pulsátil em Modelos de Árvores Arteriais**

Dissertação apresentada ao da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Modelagem Computacional.

Orientador: Titulação Rafael Alves Bonfim de Queiroz

Coorientador: Titulação Ruy Freitas Reis

Juiz de Fora

2021

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Pires dos Santos, Igor.

Simulação de Escoamento Pulsátil em Modelos de Árvores Arteriais /  
Igor Pires dos Santos. – 2021.

57 f. : il.

Orientador: Rafael Alves Bonfim de Queiroz

Coorientador: Ruy Freitas Reis

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Programa  
de Pós-Graduação em Modelagem Computacional. , 2021.

1. Árvores arteriais. 2. Escoamento pulsátil. 3. Hemodinâmica  
Computacional. I. Alves Bonfim de Queiroz, Rafael, orient. II. Dr..

Igor Pires dos Santos

**Simulação de Escoamento Pulsátil em Modelos de Árvores Arteriais**

Dissertação apresentada ao da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Modelagem Computacional.

Aprovada em (dia) de (mês) de (ano)

**BANCA EXAMINADORA**

---

Titulação Rafael Alves Bonfim de Queiroz - Orientador  
Universidade Federal de Juiz de Fora

---

Titulação Nome e sobrenome  
Universidade ???

---

Titulação Nome e sobrenome  
Universidade ??



Dedico este trabalho ...





## AGRADECIMENTOS

Agradeço aos ...



A construção de modelos de árvores arteriais é importante para a realização de estudos hemodinâmicos. Neste trabalho, apresentam-se: (i) um esquema analítico para o cálculo das características locais das ondas de fluxo e pressão em modelos de árvores arteriais 1D (ii) um ambiente computacional desenvolvido para a simulação e visualização dos resultados no tocante à construção de modelos e estudos hemodinâmicos. Os resultados obtidos neste trabalho estão condizentes com dados numéricos relatados na literatura.



## RESUMO

A construção de modelos de árvores arteriais é importante para a realização de estudos hemodinâmicos. Neste trabalho, apresentam-se: (i) um esquema analítico para o cálculo das características locais das ondas de fluxo e pressão em modelos de árvores arteriais 1D, (ii) um ambiente computacional desenvolvido para a simulação e visualização dos resultados no tocante à construção de modelos e estudos hemodinâmicos. Os resultados obtidos neste trabalho estão condizentes com dados numéricos relatados na literatura.

Palavras-chave: Árvores arteriais. Escoamento pulsátil. Hemodinâmica Computacional.



## ABSTRACT

The construction of arterial tree's models is crucial in hemodynamics studies. In this work, the following are presented: (i) an analytical scheme based on physics and mathematic laws to calculate the local characteristics of the pressure and flux wave in 1D arterial tree's models, (ii) a computational environment developed to simulate and visualize the results of the model's construction and hemodynamic studies. The results produced in this work are consistent to real morphometric data and numeric data related in the literature.

Keywords: Árvores arteriais. Pulsatile flow. Computational hemodynamics.





## LISTA DE ILUSTRAÇÕES

Notação usada para identificar cada segmento de vaso $(k, j)$ , onde $k$ é a geração/nível do vaso e $j$ é um número sequencial dentro daquela geração. . . .	30
Interface gráfica da ferramenta desenvolvida. . . . .	39
Representação de classes de um elemento inteligente. WiseElement a classe abstrata base e seus componentes: WiseStructure representa a estrutura contida em um arquivo VTK e DataStructure representa a estrutura de ponteiros e variáveis utilizadas na iteração. Tingido de azul as estruturas que nem sempre estão presentes. . . . .	41
Tipos de elementos inteligentes. <i>WiseGraphic</i> , um gráfico bidimensional. <i>WiseMesh</i> , uma malha bidimensional. <i>WisePoly</i> , uma malha tridimensional. <i>WiseArteryTree</i> , uma árvore arterial . . . . .	41
Pontos utilizados na especificação do modelo geométrico. Linhas utilizadas na especificação do modelo geométrico, através dos pontos previamente definidos. Células utilizadas na especificação do modelo geométrico, através dos pontos previamente definidos. . . . .	42
Máquina de Status que controla o funcionamento de um elemento inteligente.	43
Elemento inteligente enquanto no estado <i>Warming</i> . . . . .	43
Elemento inteligente enquanto no estado <i>Cold</i> . . . . .	44
Elemento inteligente enquanto no estado <i>Hot</i> . . . . .	44
Arquitetura de classes fábrica e fluxo de trabalho do elemento inteligente WiseElement. A fábrica WiseElementFactory é responsável por criar o elementos inteligentes, a fábrica WiseIterationFactory é responsável pela iteração do elemento inteligente e a fábrica GraphicFactory é responsável por criar as estruturas de visualização. . . . .	45
Objeto inteligente <i>WiseObject</i> e todos seu componentes: <i>WiseObjectFactory</i> , fábrica responsável pela criação de objetos inteligentes de um determinado tipo; <i>WiseIterationFactory</i> , fábrica de iteração; <i>WiseGraphicFactory</i> , fábrica gráfica; <i>WiseCollection</i> , coleção de elementos inteligentes; <i>GraphicModel</i> , coleção de objetos gráficos. . . . .	47
Modelo gráfico <i>GraphicModel</i> , contém uma coleção de objetos gráficos. . . .	47
Modelo gráfico <i>GraphicModel</i> , contém uma coleção de objetos gráficos. . . .	49



## LISTA DE TABELAS

Tabela 1 – Propriedades de cada segmento de vaso sanguíneo. . . . .	35
---	----



## **LISTA DE ABREVIATURAS E SIGLAS**

ABNT	Associação Brasileira de Normas Técnicas
------	--



## LISTA DE SÍMBOLOS

$\eta$	Viscosidade Sanguínea
$s$	Comprimento do segmento $s$
$\Delta p_s$	Queda de pressão ao longo do segmento $s$
$Q_s$	Fluxo sanguíneo através do segmento $s$
$p_{perf}$	Pressão de perfusão na posição proximal
$p_{term}$	Pressão terminal na posição distal
$\mathbf{x}_{prox}$	Posição proximal do segmento raiz
$r_p$	Raio do segmento pai
$r_{esq}$	Raio do segmento filho à esquerda
$r_{dir}$	Raio do segmento filho à direita
$ktot$	Número de segmentos existentes na árvore
$T$	Função custo para o CCO
$\mu_p$	Dimensionalidade de $T$
$\lambda$	Dimensionalidade de $T$
$p$	Pressão
$q$	Fluxo
$t$	Tempo
$x$	Coordenada axial ao longo do tubo arterial
$c$	Velocidade de onda
$Y$	Admitância do segmento arterial
$A$	Área da seção transversal do segmento arterial
$\rho$	Densidade do fluido
$w$	Frequência angular
$f$	Frequência da onda
$L$	Comprimento do tubo arterial
$k$	Geração do segmento na árvore
$j$	Posição do segmento arterial na geração da árvore
$E$	Módulo de Young
$h$	espessura da parede do segmento de tubo arterial
$p_f$	Pressão na posição proximal do segmento arterial
$p_b$	Pressão na posição distal do segmento arterial
$R$	Coefficiente de Reflexão
$Y_e$	Admitância efetiva do segmento arterial
$\alpha$	Número de Womersley
$\epsilon$	Fator viscoso do segmento arterial
$\mu$	Viscosidade no segmento arterial
$J_p$	Função de Bessel de índice $p$





## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>25</b>
<b>2</b>	<b>Escoamento sanguíneo pulsátil em árvores arteriais . . . . .</b>	<b>29</b>
2.1	Cálculo da pressão e do fluxo sanguíneo . . . . .	31
2.2	Cálculo dos coeficientes de reflexão e da admitância . . . . .	32
2.3	Cenários investigados com o modelo matemático do escoamento sanguíneo	33
2.4	Algoritmo para o escoamento pulsátil . . . . .	34
<b>3</b>	<b>Ferramenta computacional . . . . .</b>	<b>39</b>
3.1	Estrutura de Dados . . . . .	40
3.1.1	Elemento Inteligente . . . . .	40
3.1.2	Fábricas . . . . .	45
3.1.3	Objeto Inteligente . . . . .	46
3.1.4	Objeto Gráfico . . . . .	48
<b>4</b>	<b>CONCLUSÃO . . . . .</b>	<b>51</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>53</b>
	<b>APÊNDICE A – Título . . . . .</b>	<b>55</b>
	<b>ANEXO A – Título . . . . .</b>	<b>57</b>



## 1 INTRODUÇÃO

Os benefícios que a Matemática Aplicada e Computacional pode proporcionar à medicina vascular estão condicionados à superação de algumas barreiras. A primeira barreira está associada à construção do modelo geométrico da rede vascular, principalmente, a caracterização da estrutura geométrica dos vasos ao nível da circulação periférica (arteríolas e capilares). A segunda está atrelada à descrição matemática e simulação do escoamento sanguíneo pulsátil em modelos geométricos de rede vascular.

Sobre a primeira barreira, a construção de modelos geométricos de árvores arteriais *in silico* pode ser obtida empregando algoritmos baseados em leis fractais (DAWANT M. LEVIN, 2017; BEEK S. A. ROGER, 1989) e princípios de otimização (KARCH F. NEUMANN, 1999; QUEIROZ, 2013; SCHREINER, 1993). Modelos fractais assumem que as leis de ramificação são derivadas a partir de medições e repetitivamente são aplicadas em direção aos segmentos de vasos de menor calibre. Esta classe de modelos é relativamente fácil de gerar e reproduz as distribuições estatísticas dos segmentos (raios, comprimentos e ângulos de bifurcação) conhecidas a partir de medições. Porém, tal classe tem dificuldade, ou mesmo impossibilidade, de produzir um arranjo espacial anatomicamente apropriado dos segmentos. Pois, estes modelos são baseados em relações matemáticas que não controlam a estrutura geométrica dos segmentos durante o crescimento das redes vasculares.

Já em relação à simulação do escoamento sanguíneo pulsátil, destacam-se que estudos de simulação hemodinâmica sistêmica têm sido frequentemente baseados em modelos de árvores arteriais para obter uma melhor compreensão de todos os aspectos relacionados ao escoamento sanguíneo, desde a propagação de ondas e análise do pulso de pressão, passando pelo diagnóstico e inclusive com aplicações no planejamento cirúrgico. Como a representação do sistema cardiovascular através de um modelo puramente 3D que leve em conta a estrutura geométrica exata de todos os vasos não é, no momento, viável computacionalmente, vêm sendo empregados modelos dimensionalmente heterogêneos conhecidos como 0D (zero-dimensional)–1D (unidimensional)–3D (tridimensional) (BLANCO, 2008; FORMAGGIA J.F. GERBEAU, 2001; URQUIZA P.J. BLANCO, 2006). Modelos 3D (PESKIN, 1972; TAYLOR T.J.R. HUGHES, 1998) são utilizados para estudar em detalhe a hemodinâmica local de distritos arteriais de interesse, e a geometria destes modelos são provenientes de dados anatômicos obtidos normalmente via reconstrução de imagens médicas de pacientes específicos. Modelos 1D (AVOLIO, 1980; FORMAGGIA D. LAMPONI, 2003; STERGIOPOULOS D.F. YOUNG, 1992) são adotados para representar as artérias de maior calibre e a estrutura geométrica destes modelos pode ser construída a partir de dados anatômicos. Tais modelos são capazes de capturar os efeitos de propagação de ondas (ANLIKER R.L. ROCKWELL, 1971; DUAN, 1986), a interação das reflexões destas ondas e dar como resultado um pulso de pressão e vazão com significado fisiológico tanto em artérias centrais como periféricas. No entanto, um

modelo 1D de toda a árvore arterial sistêmica não é possível devido à falta de dados anatômicos precisos das regiões periféricas. Portanto, a árvore tem que ser truncada em algum nível. Normalmente, este truncamento é feito empregando modelos 0D (MATES F.J. KLOCKE, 1988; STERGIOPULOS D.F. YOUNG, 1992) conhecidos por terminais Windkessel à jusante da posição distal do modelo 1D para representar o comportamento de distritos arteriais relacionados com o nível de arteríolas e capilares.

No tocante à descrição do escoamento sanguíneo pulsátil em árvores arteriais, adotou-se neste trabalho o modelo 1D proposto por Duan e Zamir (DUAN, 1986). Eles propuseram um modelo relativamente simples para representação da pressão sanguínea e do fluxo em um modelo de árvore arterial. Dentro de cada segmento de vaso, o escoamento sanguíneo foi calculado baseado em uma aproximação de Womersley, incluindo a elasticidade da parede, bem como a densidade do sangue e a viscosidade.

Este trabalho está organizado como segue. ORGANIZAÇÃO.

A principal motivação para a construção automática de modelos de árvores arteriais é a inviabilidade de ter dados anatômicos suficientes que permitam caracterizar em detalhe a estrutura geométrica de redes vasculares periféricas (QUEIROZ, 2013). A representação adequada destas redes é necessária para modelar adequadamente o efeito dos leitos periféricos na hemodinâmica do sistema arterial humano, assim como também para permitir explorar as condições hemodinâmicas locais que se encontram na circulação periférica. Pois, os terminais Windkessel, os quais são modelos a parâmetros condensados (STERGIOPULOS D.F. YOUNG, 1992), representam de forma simplificada toda esta rede periférica rica em detalhe e que tem influência na resposta dos modelos 1D e 3D. De fato, os terminais Windkessel negligenciam a estrutura e as propriedades espaciais da microcirculação.

A incidência maior de picos na onda de pressão ao percorrer a aorta já foi documentada como evidência para os efeitos da reflexão em árvores vasculares (KOUCHOUKOS L.C. SHEPPARD, 1970; LIGHTHILL, 1975; MCDONALD, 1974). Enquanto as áreas de reflexão não podem ser completamente conhecidas ou localizadas, é geralmente aceito que a forma da onda de pressão é modificada significativamente enquanto progride pela aorta, de uma forma que só pode ser explicada por reflexões de onda. Um entendimento mais claro da relação entre modificações e fatores de modificação motiva a busca e desenvolvimento de modelos matemáticos que determinam a forma da onda que o pulso de pressão toma em cada ponto ao percorrer uma árvore arterial. Dentro deste contexto, estudou-se o modelo de Duan e Zamir (DUAN, 1986).

A construção de modelos geométricos para representação adequada dos leitos periféricos é mandatória para adquirir entendimento da influência destes nas respostas hemodinâmicas de modelos 1D e 3D do sistema cardiovascular. No entanto, a possibilidade de construir um modelo geométrico detalhado do ponto de vista anatômico de uma rede

vascular real a partir da reconstrução de imagens médicas de ressonância magnética é ainda limitada (HAREL P.J. BOLAN, 2010).

O cálculo correto das características locais das ondas de pressão e fluxo a medida que elas progridem ao longo de uma estrutura de árvore e se tornam modificadas por reflexões de onda possibilita capturar o pico de pressão existente no escoamento sanguíneo. Por isto, torna-se interessante e importante o estudo de modelos matemáticos como aquele de Duan e Zamir (DUAN, 1986) para compreender melhor a hemodinâmica do sistema cardiovascular.

Os objetivos pretendidos e alcançados neste trabalho foram:

- estudar e implementar o modelo matemático de Duan e Zamir que descreve o escoamento sanguíneo 1D em árvores arteriais;
- simular o escoamento sanguíneo em modelos de árvores arteriais sob diferentes cenários visando:
  - analisar o impacto da viscosidade sanguínea;
  - analisar o impacto da viscoelasticidade da parede do vaso;
  - investigar o efeito da combinação da viscosidade sanguínea e viscoelasticidade da parede do vaso;
- desenvolver um sistema computacional para a análise de escoamento pulsátil em árvores arteriais, com os seguintes recursos:
  - funcionamento multiplataforma;
  - armazenamento consistente de todo modelo geométrico e suas propriedades;
  - possibilitar análise de múltiplas árvores arteriais e modelos subsequentes;
  - visualizar em tempo real as modificações feitas no modelo;

Este trabalho tem caráter interdisciplinar envolvendo áreas de Matemática Aplicada, Computação, Biologia, Física e Medicina, o que torna fundamental o conhecimento básico de cada área.

Primeiramente, estudou-se o modelo matemático de Duan e Zamir (DUAN, 1986). Este estudo motivou o desenvolvimento de uma nova ferramenta computacional chamada **IGU (Iterador Gráfico Universal)** para descrever o escoamento sanguíneo pulsátil em árvores arteriais. Inicialmente para verificar a robustez da ferramenta, buscou-se realizar algumas simulações semelhantes aquelas de Duan e Zamir a título de comparação. Destaca-se que a ferramenta **IGU** foi desenvolvida em C++ e contou com a utilização das bibliotecas Qt/OpenGL para ajudar na elaboração da interface gráfica.

AUMENTAR.



## 2 Escoamento sanguíneo pulsátil em árvores arteriais

Nesta seção apresenta-se em detalhe a derivação do modelo matemático de Duan e Zamir (DUAN, 1986), para o escoamento sanguíneo pulsátil em árvores arteriais.

A propagação de ondas em um tubo é governada pela equações da onda para a pressão  $p(x, t)$  e fluxo  $q(x, t)$  como seguem:

$$\frac{\partial q}{\partial t} = -cY \frac{\partial p}{\partial x}, \quad (2.1)$$

$$\frac{\partial p}{\partial t} = -\frac{c}{Y} \frac{\partial q}{\partial x}, \quad (2.2)$$

nos quais  $t$  é o tempo,  $x$  é a coordenada axial ao longo do tubo,  $c$  é a velocidade de onda,  $Y = \frac{A}{\rho c}$  é a admitância e  $A$  é a área da seção transversal do tubo, e  $\rho$  é a densidade do fluido. Estas equações são baseadas na linearização das equações de movimento do fluido (DUAN, 1984; LIGHTHILL, 1975). Para uma onda harmônica simples, as equações (2.1) e (2.2) resultam em:

$$p = \bar{p}_0 \exp \left[ i\omega \left( t - \frac{x}{c} \right) \right] + R\bar{p}_0 \exp \left[ i\omega \left( t - \frac{2L}{c} + \frac{x}{c} \right) \right], \quad (2.3)$$

$$q = Y \left\{ \bar{p}_0 \exp \left[ i\omega \left( t - \frac{x}{c} \right) \right] - R\bar{p}_0 \exp \left[ i\omega \left( t - \frac{2L}{c} + \frac{x}{c} \right) \right] \right\}, \quad (2.4)$$

Onde  $\omega = 2\pi f$  é a frequência angular,  $f$  é frequência em Hertz,  $L$  é o comprimento do tubo,  $\bar{p}_0$  a amplitude da onda incidente,  $R$  é o coeficiente de reflexão definido pela razão entre as ondas refletidas pelas ondas que chegam no local de reflexão (DUAN, 1984; KARREMAN, 1952) e  $i$  é a unidade imaginária ( $i^2 = -1$ ).

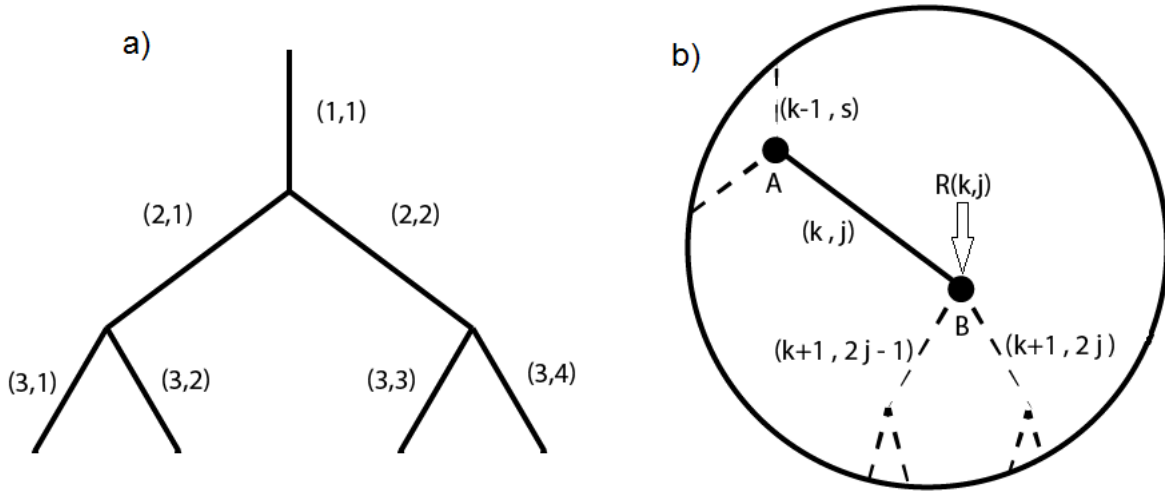
As equações (2.3) e (2.4) para pressão e fluxo são aplicadas em cada segmento de vaso do modelo de árvore arterial, tomando  $x = 0$  para o nó proximal e  $x = L$  para o nó distal do segmento. Um segmento de vaso é definido pelo intervalo vascular entre dois locais de ramificação (ZAMIR, 1988). No sistema arterial, as bifurcações são os locais de ramificação mais comuns (ZAMIR, 1976).

Em (DUAN, 1986), um segmento de vaso é identificado por  $(k, j)$ , onde o primeiro  $k$  representa o nível da geração e  $j$  representa a ordem do segmento naquela geração, como mostrado na Figura 1. Desta forma, a pressão e o fluxo ao longo de um segmento  $(k, j)$  do modelo de árvore arterial são dados por:

$$p(k, j) = \bar{p}(k, j) \exp \left[ i\omega \left( t - \frac{x(k, j)}{c(k, j)} \right) \right] + R(k, j)\bar{p}(k, j) \exp \left[ i\omega \left( t - \frac{2L(k, j)}{c(k, j)} + \frac{x(k, j)}{c(k, j)} \right) \right], \quad (2.5)$$

$$q(k, j) = Y(k, j) \left\{ \bar{p}(k, j) \exp \left[ i\omega \left( t - \frac{x(k, j)}{c(k, j)} \right) \right] - R(k, j)\bar{p}(k, j) \exp \left[ i\omega \left( t - \frac{2L(k, j)}{c(k, j)} + \frac{x(k, j)}{c(k, j)} \right) \right] \right\}, \quad (2.6)$$

nos quais  $\bar{p}(k, j)$  é a amplitude combinada do grupo de ondas progressivas no segmento  $(k, j)$  e  $R(k, j)$  é o coeficiente de reflexão no final daquele segmento, como é chamado a razão das ondas progressivas pelas atrasadas avaliadas no nó distal  $x(k, j) = L(k, j)$ . O grupo de ondas progressivas viaja no sentido positivo de  $x(k, j)$ , estas são compostas de ondas progressivas vindo de vasos acima deste, bem como, ondas refletidas na junção à montante  $x(k, j) = 0$ . O grupo de ondas atrasadas viaja no sentido oposto e é composto por ondas vindas de vasos à jusante como ondas refletidas na junção à jusante  $x(k, j) = L(k, j)$ . Assim, as equações (2.5) e (2.6) descrevem, respectivamente, as ondas de pressão e de fluxo localmente em um segmento  $(k, j)$  do modelo de árvore, e localmente na posição  $x(k, j)$  dentro deste segmento de vaso. As duas variáveis desconhecidas são a amplitude da pressão  $\bar{p}(k, j)$  e o coeficiente de reflexão  $R(k, j)$ , que são calculados na Seção 2.1 conforme proposto por Duan e Zamir (DUAN, 1986).



– Notação usada para identificar cada segmento de vaso  $(k, j)$ , onde  $k$  é a geração/nível do vaso e  $j$  é um número sequencial dentro daquela geração.

Os nós proximal e distal do segmento  $(k, j)$  são denotados por  $A$  e  $B$ , respectivamente. O coeficiente de reflexão  $R(k, j)$  do segmento  $(k, j)$  está associado ao nó distal  $B$  (adaptada de (DUAN, 1986)). Na equação (2.6), tem-se a admitância característica para cada segmento dada por:

$$Y(k, j) = \frac{A(k, j)}{\rho(k, j)c(k, j)}, \quad (2.7)$$

nos quais  $A(k, j)$  é a área da seção transversal do segmento  $(k, j)$ ,  $\rho(k, j)$  é a densidade do fluido dentro do vaso e  $c(k, j)$  é a velocidade da onda correspondente. A admitância de um segmento é o inverso de sua impedância  $Y = \frac{1}{Z}$ , ou seja, é uma medida do quanto o segmento permite o fluxo.

Assumindo um segmento elástico de parede fina, a velocidade da onda  $c(k, j)$  é



calculada por (DUAN, 1984):

$$c(k, j) = \sqrt{\frac{E(k, j)h(k, j)}{\rho(k, j)d(k, j)}}, \quad (2.8)$$

onde  $E(k, j)$  é o módulo de Young,  $d(k, j)$  é o diâmetro do segmento  $(k, j)$  e  $h(k, j)$  é a espessura da parede do segmento, a qual neste estudo é dada por (DUAN, 1986):  $h(k, j) = 0,05d(k, j)$ .

## 2.1 Cálculo da pressão e do fluxo sanguíneo

Para determinar a pressão  $\bar{p}(k, j)$  em um certo segmento  $(k, j)$ , aplica-se a condição de continuidade de pressão no nó proximal  $A$  (ver Figura 1). Escrevendo as componentes progressiva e atrasada da onda como  $p_f(k, j)$  e  $p_b(k, j)$  respectivamente, a pressão na posição proximal do segmento  $x(k, j) = 0$  é dada por:

$$[p(k, j)]_A = [p_f(k, j)]_A + [p_b(k, j)]_A, \quad (2.9)$$

nos quais as pressões  $[p_f(k, j)]_A$  e  $[p_b(k, j)]_A$  são expressas por:

$$[p_f(k, j)]_A = \bar{p}(k, j) \exp[i\omega t], \quad (2.10)$$

$$[p_b(k, j)]_A = R(k, j)\bar{p}(k, j) \exp\left[i\omega\left(t - \frac{2L(k, j)}{c(k, j)}\right)\right]. \quad (2.11)$$

Similarmente, a pressão no segmento pai  $(k - 1, s)$  pode ser escrita como:

$$p(k - 1, s) = p_f(k - 1, s) + p_b(k - 1, s), \quad (2.12)$$

nos quais  $s$  é um número sequencial do segmento pai e as pressões  $p_f(k - 1, s)$  e  $p_b(k - 1, s)$  são dadas por:

$$p_f(k - 1, s) = \bar{p}(k - 1, s) \exp\left[i\omega\left(t - \frac{x(k - 1, s)}{c(k - 1, s)}\right)\right], \quad (2.13)$$

$$p_b(k - 1, s) = R(k - 1, s)\bar{p}(k - 1, s) \exp\left[i\omega\left(t - \frac{2L(k - 1, s)}{c(k - 1, s)} + \frac{x(k - 1, s)}{c(k - 1, s)}\right)\right].$$

No nó distal do vaso superior,  $x(k - 1, s) = L(k - 1, s)$ , a pressão é dada por:

$$[p(k - 1, s)]_A = [p_f(k - 1, s)]_A + [p_b(k - 1, s)]_A, \quad (2.14)$$

nos quais

$$[p_f(k - 1, s)]_A = \bar{p}(k - 1, s) \exp\left[i\omega\left(t - \frac{L(k - 1, s)}{c(k - 1, s)}\right)\right], \quad (2.15)$$

$$[p_b(k - 1, s)]_A = R(k - 1, s)\bar{p}(k - 1, s) \exp\left[i\omega\left(t - \frac{L(k - 1, s)}{c(k - 1, s)}\right)\right]. \quad (2.16)$$

A condição de continuidade da pressão exige que na junção ela assuma um único valor, portanto

$$[p_f(k-1, s)]_A + [p_b(k-1, s)]_A = [p_f(k, j)]_A + [p_b(k, j)]_A. \quad (2.17)$$

Substituindo as equações (2.10), (2.11), (2.15) e (2.16) na equação (2.17) e resolvendo para  $\bar{p}(k, j)$ , resulta em:

$$\bar{p}(k-1) = \frac{\bar{p}(k-1, s) [1 + R(k-1, s)] \exp \left[ -\frac{i\omega L(k-1, s)}{c(k-1, s)} \right]}{1 + R(k, j) \exp \left[ -2i\omega \frac{L(k, j)}{c(k, j)} \right]}. \quad (2.18)$$

Conforme Duan e Zamir (DUAN, 1986), para efeitos de cálculo da pressão e fluxo, adimensionalizam-se as pressões em (2.18) em termos da pressão de entrada  $p_0 = \bar{p}_0 \exp[i\omega t]$ . Considerando  $P(k, j) = \frac{p(k, j)}{p_0}$  e  $\bar{P}(k, j) = \frac{\bar{p}(k, j)}{\bar{p}_0}$ , a equação (2.5) para o cálculo da pressão pode ser expressa de forma adimensionalizada por:

$$P(k, j) = \bar{P}(k, j) \{ \exp[-i\beta(k, j)X(k, j)] + R(k, j) \exp[-i2\beta(k, j)] \exp[i\beta(k, j)X(k, j)] \}, \quad (2.19)$$

nos quais  $\beta(k, j) = \frac{\omega(k, j)L(k, j)}{c(k, j)}$  e  $X = \frac{x(k, j)}{L(k, j)}$ . Similarmente, a equação (2.6) para o fluxo  $q(k, j)$  pode ser obtida de forma adimensionalizada por:

$$Q(k, j) = M(k, j)\bar{P}(k, j) \{ \exp[-i\beta(k, j)X(k, j)] - R(k, j) \exp[-i2\beta(k, j)] \exp[i\beta(k, j)X(k, j)] \}, \quad (2.20)$$

nos quais  $Q(k, j) = \frac{q(k, j)}{q_0}$ ,  $M = \frac{Y(k, j)}{Y(1, 1)}$  e  $q_0 = Y(1, 1)p_0$ . O cálculo da admitância  $Y(1, 1)$  na posição proximal do segmento raiz, ou seja, da artéria de alimentação é apresentado na próxima seção.

## 2.2 Cálculo dos coeficientes de reflexão e da admitância

Para determinar os coeficientes de reflexão nas junções, consideram-se as duas junções  $A$  e  $B$  das extremidades de um segmento genérico  $(k, j)$  de um modelo de árvore arterial conforme Duan e Zamir (DUAN, 1986). Na posição distal  $B$ , o coeficiente de reflexão é definido por (DUAN, 1984; LIGHTHILL, 1975):

$$R(k, j) = \frac{Y(k, j) - [Y_e(k+1, 2j) + Y_e(k+1, 2j-1)]}{Y(k, j) + [Y_e(k+1, 2j) + Y_e(k+1, 2j-1)]}, \quad (2.21)$$

nos quais  $Y_e(k+1, 2j-1)$  e  $Y_e(k+1, 2j)$  são admitâncias efetivas nos segmentos à jusante de  $B$ . Estas admitâncias são determinadas pela razão entre o fluxo e pressão naquela posição, que é dada por:

$$Y_e(k+1, s) = \frac{Y(k+1, s) \{1 - R(k+1, s) \exp[-i2\beta(k+1, s)]\}}{1 + R(k+1, s) \exp[-i2\beta(k+1, s)]}, \quad (2.22)$$

nos quais  $s = 2j - 1$  e  $2j$  são os números sequenciais dos dois segmentos filhos e  $R(k + 1, s)$  é o coeficiente de reflexão na posição distal de cada segmento. Similarmente,  $Y_e(k, j)$ , a admitância na posição proximal  $A$  do segmento  $(k, j)$  pode ser dada por:

$$Y_e(k, j) = \frac{Y(k, j)\{1 - R(k, j) \exp[-i2\beta(k, j)]\}}{1 + R(k, j) \exp[-i2\beta(k, j)]}. \quad (2.23)$$

Substituindo  $R(k, j)$  da equação (2.21) em (2.23), obtém-se uma equação para cálculo das admitâncias efetivas ao longo do modelo de árvore arterial:

$$Y_e(k, j) = \frac{Y(k, j)[Y_e(k + 1, 2j) + Y_e(k + 1, 2j - 1) + iY(k, j) \tan \beta(k, j)]}{Y(k, j) + i[Y_e(k + 1, 2j) + Y_e(k + 1, 2j - 1)] \tan \beta(k, j)}. \quad (2.24)$$

Em segmentos terminais, pode ser assumido que não ocorrem mais reflexões à jusante das posições distais destes segmentos, portanto a admitância efetiva destes segmentos é igual às suas admitâncias características. Adotando a equação (2.23), todas as admitâncias efetivas podem ser determinadas percorrendo a árvore a partir dos segmentos terminais até o segmento raiz.

### 2.3 Cenários investigados com o modelo matemático do escoamento sanguíneo

A partir do modelo matemático aqui apresentado, os seguintes cenários são investigados nas simulações hemodinâmicas:

- **cenário 1:** análise do impacto da viscosidade sanguínea ( $\mu(k, j)$ ).

Os efeitos da viscosidade sanguínea podem ser investigados por substituir a velocidade da onda  $c(k, j)$  por uma velocidade da onda complexa (DUAN, 1995):

$$c_v(k, j) = c(k, j)\sqrt{\epsilon}, \quad (2.25)$$

onde  $\epsilon$  é um fator viscoso que corresponde a um tubo elástico com restrições (DUAN, 1995). Seja  $\alpha$  o número de Womersley adimensional

$$\alpha = r(k, j) \sqrt{\frac{\omega \rho(k, j)}{\mu(k, j)}}, \quad (2.26)$$

o fator viscoso  $\epsilon$  é calculado por:

$$\epsilon = 1 - F_{10}(\alpha), \quad (2.27)$$

onde a função  $F_{10}$  é avaliada deste modo:

$$F_{10}(\alpha) = \frac{2J_1(i^{1,5}\alpha)}{\alpha i^{1,5}J_0(i^{1,5}\alpha)}, \quad (2.28)$$

onde  $J_p$  denota a função de Bessel de índice  $p$ .

- **cenário 2:** análise do impacto da viscoelasticidade da parede do vaso ( $\phi_0$ ).

A viscoelasticidade da parede do segmento é incorporado substituindo o módulo de Young estático  $E(k, j)$  por um módulo elástico complexo  $E_c(k, j)$  no cálculo da velocidade  $c(k, j)$  na equação (2.8) da seguinte forma (DUAN, 1986):

$$E_c(k, j) = |E_c(k, j)| \exp\{i\phi\}, \quad (2.29)$$

onde  $\phi$  é o ângulo de fase entre a pressão e o deslocamento da parede do segmento (TAYLOR, 1966) expresso por  $\phi = \phi_0[1 - \exp(-\omega)]$  e  $|E_c(k, j)|$  corresponde ao módulo de Young fornecido para a simulação.

- **cenário 3:** efeitos da viscosidade sanguínea ( $\mu(k, j)$ ) e da viscoelasticidade da parede do segmento ( $\phi_0$ ) de forma combinada.

Neste último cenário, utiliza-se a equação (2.29) para determinar a velocidade da onda  $c(k, j)$  (2.8) no modelo. Com este resultado, calcula-se a equação (2.25) para determinar a velocidade complexa  $c_v(k, j)$  a ser considerada no modelo.

## 2.4 Algoritmo para o escoamento pulsátil

Para o cálculo correto da onda de pressão e fluxo presente nos modelos geométricos o algoritmo presente nesta seção é utilizado. Este algoritmo irá definir as variáveis na seguinte ordem:

- Cálculo da admitância ( $Y$ )
- Cálculo da admitância efetiva ( $Y_e$ )
- Cálculo do coeficiente de reflexão ( $R$ )
- Cálculo da pressão média ( $\bar{p}$ )
- Cálculo das ondas de pressão e fluxo ( $P(x)eQ(x)$ )

As equações (2.21) e (2.22) expõem a necessidade de um valor  $Y_e(k + 1, 2j)$  e  $Y_e(k + 1, 2j - 1)$  que fazem referência à valores de artérias adjacentes ao final deste segmento. Entretanto, o caminho reverso é necessário para se resolver o valor da pressão média de cada segmento como visto na equação (2.18) que necessita da pressão efetiva do segmento superior  $\bar{p}(k - 1, s)$ . Com estes desafios em mente, a estrutura de dados desenvolvida utiliza de ponteiros para acessar os valores de artérias adjacentes. O algoritmo desenvolvido se dividiu em duas fases, em um primeiro momento realiza o cálculo dos

segmentos finais ao segmento raiz (*bottom-up*) e ,finalmente, da raiz aos segmentos finais (*top-bottom*).

---

**Algoritmo 1:** Começo da recursão.

---

**Entrada:** *Modelo Arterial*,  $f(\text{Hz}), \epsilon, \mu_0, \phi_0$

**início**

Segmento  $a = \text{raiz}$  (*começa a recursão pela raiz*)

**se** *existe*( $a$ ) **então**

| Envia a recursão *fase um*( $a, f, \mu_0, \phi_0$ ) e armazena  $Y(k - 1, s)$

**fim**

**se** *existe*( $Y(k - 1, s)$ ) **então**

| Envia a recursão *fase dois*( $a, p_0, Y(k - 1, s)$ )

**fim**

**fim**

---

O valor complexo ( $Y(k - 1, s)$ ) armazenado é a admitância característica do segmento raiz, que é necessário para o cálculo da onda de fluxo como visto na equação (2.20).

A linguagem escolhida permite que objetos sejam passados por ponteiros, portanto é possível passar toda a estrutura geométrica da árvore arterial através de um ponteiro para o segmento raiz. Cada segmento arterial que contém as seguintes propriedades:

Variável	Unidade	Variável	Unidade
Comprimento	$L(\text{cm})$	Raio	$r(\text{cm})$
Densidade	$\rho(\text{g/cm}^3)$	Viscosidade	$\mu(\text{cm}^2/\text{s})$
Espessura da Parede	$h(\text{cm})$	Velocidade Angular	$\omega$
Velocidade de Onda	$c(\text{cm/s})$	Alpha	$\alpha$
Beta	$\beta$	Módulo de Young	$E(\text{g/cms}^2)$
Admitância Característica	$Y$	Admitância Efetiva	$Y_e$
Fator Viscoso	$\epsilon$	Coeficiente de Reflexão	$R$
Pressão Média	$\bar{p}$	Pressão	$P(x)$
Fluxo	$Q(x)$		

Tabela 1 – Propriedades de cada segmento de vaso sanguíneo.

Além destas informações o segmento possui ainda três ponteiros para segmentos adjacentes. Um destes para o segmento adjacente ao nó proximal, ou segmento de vaso superior ( $k - 1, s$ ); Os restantes para os segmentos adjacentes ao nó distal, arbitrariamente segmento à esquerda e à direita, ou segmentos de vaso inferiores ( $k + 1, 2j - 1$ ) e ( $k + 1, 2j$ ). Esta estrutura é equivalente à uma árvore duplamente encadeada, onde através de um único ponteiro para um segmento, é possível trafegar a árvore nos dois sentidos. Isto se torna particularmente útil quando é necessário acessar dados de outro segmento para o cálculo.

Os ponteiros permitem também determinar se a artéria é a raiz ou uma folha, caso o ponteiro para o segmento superior ( $k - 1, s$ ) seja igual ao valor de *Flag* nulo se trata de

um segmento raiz. Caso os dois segmentos  $(k + 1, 2j - 1)$  e  $(k + 1, 2j)$  sejam iguais ao valor de *Flag* nulo se trata de uma folha.

A primeira fase do algoritmo foi desenvolvida para calcular os valores de admitância efetiva ( $Y_e$ ) e coeficiente de reflexão ( $R$ ) de cada segmento. Desta forma é necessário que a admitância efetiva dos segmentos inferiores ( $Y_e(k + 1, 2j - 1)$  e  $Y_e(k + 1, 2j)$ ) já estejam definidos. Em vista disso, durante a primeira fase de cálculo é verificada a existência de segmentos inferiores e caso existam a recursão é enviada até eles. Após o envio da recursão é feito o cálculo do segmento atual, desta forma o segmento inferior será definido antes do segmento atual e é garantida a existência de  $Y_e(k + 1, 2j - 1)$  e  $Y_e(k + 1, 2j)$ .

---

**Algoritmo 2:** Primeira fase do cálculo, recursão (*bottom-up*).

---

**fase um**( $a, f, \epsilon, \mu_0, \phi_0$ )

**Entrada:** Segmento  $a, f(\text{Hz}), \epsilon, \mu_0, \phi_0$

**início**

**se** *existe*( $a \rightarrow esquerda$ ) **então**

        Envia a recursão *fase um*( $a \rightarrow esquerda, f, \mu_0, \phi_0$ )

**fim**

**se** *existe*( $a \rightarrow direita$ ) **então**

        Envia a recursão *fase um*( $a \rightarrow direita, f, \mu_0, \phi_0$ )

**fim**

    Calcula  $c, \omega, \beta$

**se** *não viscoso* **então**

        Calcula  $Y$

**fim**

**senão**

        Calcula  $\alpha, \epsilon, E_v, c_v$  e  $Y_v$

        Adota  $c = c_v$  e  $Y = Y_v$

**fim**

**se**  $a \in folha$  **então**

$Y_e = Y$  e  $R = 0$

**fim**

**senão**

        Calcula  $Y_e$  e  $R$

**fim**

**fim**

---

No caso de seguimentos que possuam apenas um vaso inferior, a recursão é enviada para o segmento existente e o para o segmento ausente se assume que  $Y_e = 0$ .

A segunda fase do algoritmo foi desenvolvida para calcular o valor da pressão média e ondas de pressão e fluxo em cada segmento. Como visto na equação (2.18) o valor da pressão média requer o valor da pressão média do segmento superior  $(k - 1, s)$ . Com isso, verifica-se se o segmento atual é a raiz, neste caso  $\bar{p} = \bar{p}_0$ , caso contrário se faz necessário

o cálculo de  $\bar{p}$  com a equação (2.18). Em seguida o valor das ondas de pressão e fluxo são adquiridos pelo domínio do segmento. Por último a recursão é enviada aos segmentos inferiores, desta forma se garante a existência de um valor  $\bar{p}(k - 1, s)$ .

---

**Algoritmo 3:** Segunda fase do cálculo, recursão (*bottom-up*).

---

```

fase dois(a,f,ε,ϕ0)
  início
    se  $a \in \text{raiz}$  então
      |  $\bar{p} = \bar{p}_0$ 
    fim
    senão
      | Calcula  $\bar{p}$ 
    fim
    Calcula  $P(X)eQ(X)\forall X \in [0, 1]$ 
    se  $\text{existe}(a \rightarrow \text{esquerda})$  então
      | Envia a recursão  $\text{fase dois}(a \rightarrow \text{esquerda}, f, \epsilon, \phi_0)$ 
    fim
    se  $\text{existe}(a \rightarrow \text{direita})$  então
      | Envia a recursão  $\text{fase dois}(a \rightarrow \text{direita}, f, \epsilon, \phi_0)$ 
    fim
  fim

```

---

Ao término da execução os valores contidos na artéria são analisados separadamente. A onda de pressão e fluxo representa um conjunto de valores por segmento que precisam ser calculados e armazenados, para isto é preciso determinar a quantidade de pontos  $N$  que serão analisados dentro do espaço  $X \in [0, 1]$ , foi adotado  $N = 101$ . No caso da onda de pressão os valores encontrados para ramo de árvore deve resultar em um gráfico contínuo, isto é  $P(k, j)(X = 0) = P(k + 1, 2j)(X = 1)$  e  $P(k, j)(X = 0) = P(k + 1, 2j - 1)(X = 1)$ .



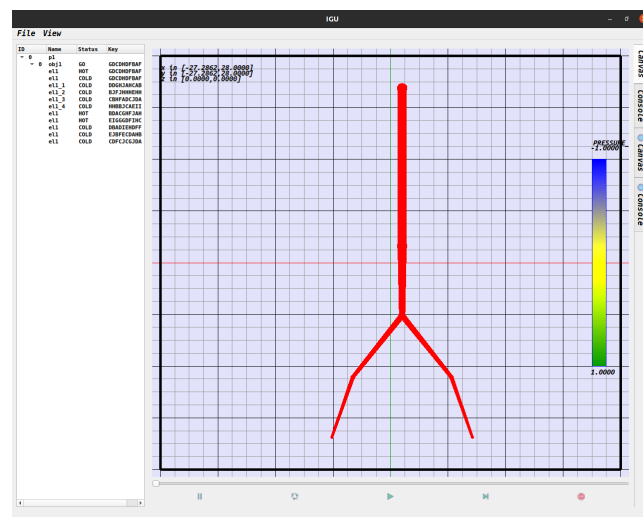


### 3 Ferramenta computacional

Nesta seção apresentam-se detalhes da ferramenta computacional desenvolvida para simulação do escoamento sanguíneo em modelos de árvores arteriais. Com esta ferramenta, o usuário poderá visualizar a estrutura da árvore arterial e depois da simulação hemodinâmica, visualizar curvas de distribuição de fluxo sanguíneo e pressão.

Esta ferramenta foi desenvolvida em C++ utilizando as bibliotecas comuns do Qt 5.15.0 (QT, 2019) e do OpenGL (GROUP, 2019), que ajudam na construção da interface gráfica e na exibição de objetos, como árvores arteriais e gráficos. Ela foi nomeada de *Iterador Gráfico Universal* (IGU), pois em seu modelo de classes qualquer objeto que implemente a classe *WiseObject*, elucidada na Seção 3.1, está apto para realizar iterações e desenhar-se através de diretivas OpenGL em um objeto de interface gráfica. Buscou-se no desenvolvimento desta ferramenta alto grau de generalização para que possa ser utilizada por diferentes objetos dentro do ambiente. Além de árvores arteriais menciona-se que a ferramenta itera diferentes estruturas de dados, entre eles malhas estruturadas e não estruturadas, por isto uma nomenclatura genérica.

A Figura 2 ilustra a ferramenta desenvolvida. À seguir, apresentam-se em detalhes a implementação computacional realizada.



– Interface gráfica da ferramenta desenvolvida.

Através da modelagem de classes no paradigma do C++ (PARKER, 1959), foi possível realizar diversas generalizações para ampliar a quantidade de objetos que podem ser inseridos neste modelo de classes. Na seção seguinte apresentamos um esquema de classes virtuais, ou abstratas, que foram criadas para facilitar o manuseio dos objetos dentro do ambiente computacional e prover funções básicas, a classe *WiseElement*. Objetos que recebem essa estrutura básica por herança são nomeados de elementos inteligentes.

### 3.1 Estrutura de Dados

Nesta seção apresentam-se detalhes da estrutura de dados adotada no funcionamento da ferramenta computacional. Como mencionado na Seção 2.4 é utilizada uma estrutura de ponteiros que é capaz de armazenar todas as informações do modelo geométrico da árvore arterial. A ferramenta computacional foi desenvolvida para ser capaz de armazenar, carregar e iterar este modelo bem como gerar novas estruturas para visualização.

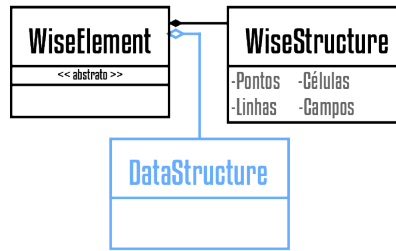
As seções à seguir descrevem a estrutura de dados genérica, sendo o modelo principal de estudo é o estudo do fluxo pulsátil atravessando a estrutura de uma árvore arterial e a visualização de seus resultados. Em versões anteriores deste ambiente computacional os objetos eram definidos em apenas uma classe abstrata, com diversos métodos virtuais que precisavam ser implementados, isto acarretava em classes herdeiras extremamente complexas. Isto porque o mesmo objeto ficava responsável por mais de uma função, era capaz de se instanciar, de se iterar e se desenhar na tela através de diretivas OpenGL. Além disso, após iterar o objeto repetidas vezes não era possível analisar objetos de iterações passadas, isto porque os dados eram sempre sobrescritos.

Com estas barreiras em mente, a classe representante do modelo foi dividida em suas funções e fábricas ficam responsáveis por suas novas instâncias. A classe de elemento inteligente, *WiseElement*, que é responsável por manter uma estrutura genérica do modelo e gerenciar a localização desta estrutura. Outra atribuição dada as fábricas foi a iteração dos modelos contidos em um elemento inteligente, isto porque o processo de iteração envolve a criação de um novo objeto para que não haja sobrescrita. Os objetos gráficos, *GraphicObject*, são objetos criados à partir da estrutura contida em um elemento inteligente que são capazes de se desenhar.

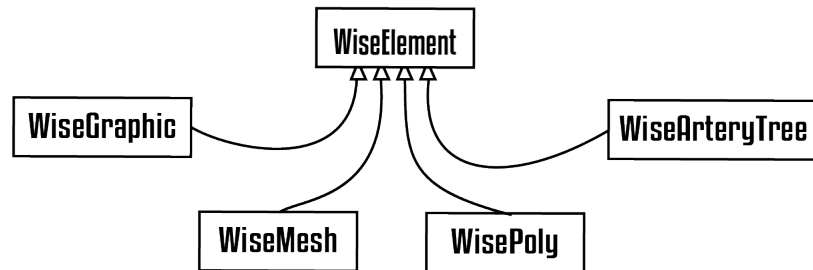
#### 3.1.1 Elemento Inteligente

Através das estruturas estruturadas e desestruturadas presentes na biblioteca *VTK* (*Visualization ToolKit*) é possível descrever os mais diversos tipos de dados com diretivas simples. Baseando-se nessa estrutura básica a arquitetura presente na Figura 3 foi criada.

A Figura 3 mostra que um elemento inteligente é composto por duas outras estruturas: A primeira *WiseStructure*, utiliza pontos, linhas, células e campos para determinar estruturas geométricas; A segunda *DataStructure*, representa os dados abstratos específicos de cada elemento. Estas estruturas são equivalentes entre si, isto é feito para que a estrutura siga um formato padrão de pontos, linhas, células e campos seja mantida enquanto dados abstratos equivalentes podem ser utilizados. Isto significa que os dados de uma árvore arterial estão disponíveis na estrutura padrão *WiseStructure* e também disponibiliza ponteiros através da *DataStructure* de uma *WiseArteryTree*.



- Representação de classes de um elemento inteligente. *WiseElement* a classe abstrata base e seus componentes: *WiseStructure* representa a estrutura contida em um arquivo VTK e *DataStructure* representa a estrutura de ponteiros e variáveis utilizadas na iteração. Tingido de azul as estruturas que nem sempre estão presentes.



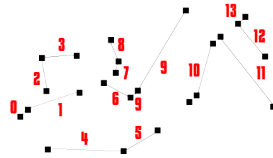
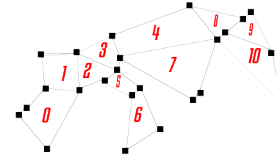
- Tipos de elementos inteligentes. *WiseGraphic*, um gráfico bidimensional. *WiseMesh*, uma malha bidimensional. *WisePoly*, uma malha tridimensional. *WiseArteryTree*, uma árvore arterial

Como demonstrado na Figura 4 um elemento inteligente é aquele que implementa a classe abstrata *WiseElement*, os dados abstratos de cada classe podem ser salvos na estrutura disponível (*WiseStructure*) e utilizados quando necessário.

Os elementos inteligentes servem como estruturas de armazenamento padrão para que possam ser utilizados por outros objetos. O ciclo de manipulação desses elementos se divide em três partes: A criação, aonde os objetos podem ser criados à partir de exemplos pré-definidos ou através de um arquivo de entrada *VTK* ou *XML* (*eXtensible Markup Language*); A iteração, processo em que o elemento inteligente com todas as estruturas definidas e consistentes é utilizado para o cálculo de alguma lógica pré-definida; A exibição, como os elementos inteligentes possuem uma variedade de parâmetros, foi criada uma estrutura com um único parâmetro que é especializada em desenhar-se na tela.

O elemento inteligente então poderá ser representado em qualquer uma das estruturas que o compõe, sendo *DataStructure* a estrutura de dados abstratos utilizados em cada

iteração e a *WiseStructure* utilizada principalmente para leitura e escrita. Em seguida, poderá ser também visualizado ao criar-se uma estrutura equivalente *GraphicObject*, que possuirá os elementos gráficos e as diretivas *OpenGL* necessárias para visualizar a forma.

**Pontos****Linhas****Células**

- Pontos utilizados na especificação do modelo geométrico. Linhas utilizadas na especificação do modelo geométrico, através dos pontos previamente definidos. Células utilizadas na especificação do modelo geométrico, através dos pontos previamente definidos.

Utilizando pontos e linhas é possível representar o mesmo modelo geométrico de uma árvore arterial, basta considerar cada ponto uma bifurcação de vasos e cada segmento de vaso uma linha. Através dessa estrutura de dados é possível armazenar e acessar dados sobre cada ponto e linha, desta forma é possível armazenar as informações de cada segmento pertencente à uma árvore arterial através da estrutura *WiseStructure*. Para dados gerais do modelo, como a frequência  $f$ , os campos da estrutura são utilizados.

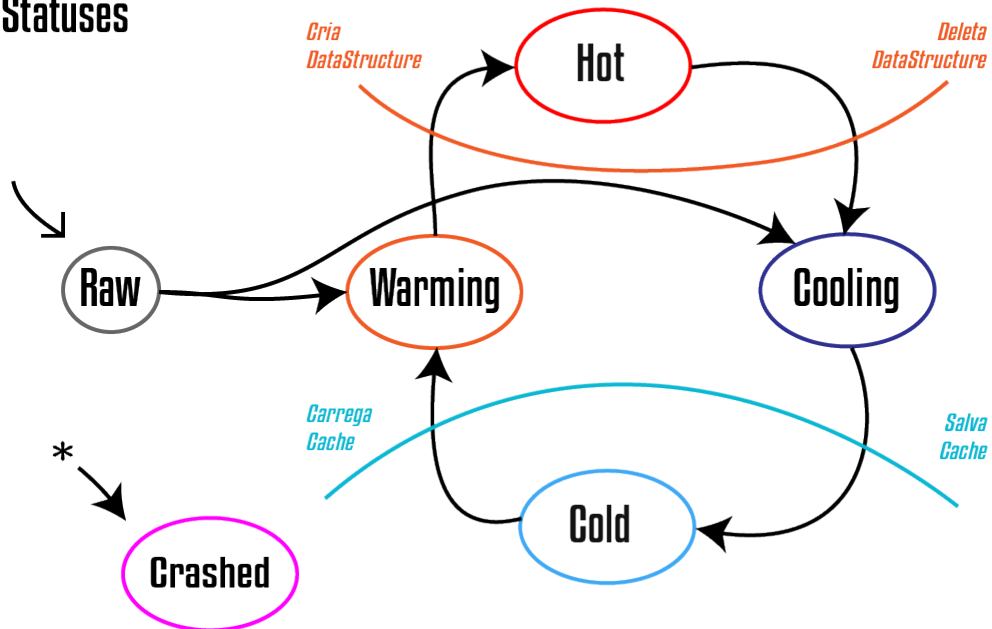
Isto foi feito para facilitar a leitura e escrita de objetos, pois a classe *WiseElement* foi desenvolvida para construir a estrutura abstrata (*DataStructure*) à partir das informações contidas na estrutura inteligente (*WiseStructure*). A estrutura abstrata não está sempre presente, pois como veremos é custoso manter em memória esta estrutura para todos os elementos. Portanto, todos os elementos inteligente obedecem à máquina de status contida na Figura 6.

Todo elemento inteligente é criado no estado *Raw*, ou cru, que representa um elemento ainda sem estruturas carregadas. Uma vez que a estrutura inteligente é inserida e verificada o elemento muda para o status *Warming* ou *Cooling*, respectivamente esquentando ou esfriando, que estão representados na Figura 7.

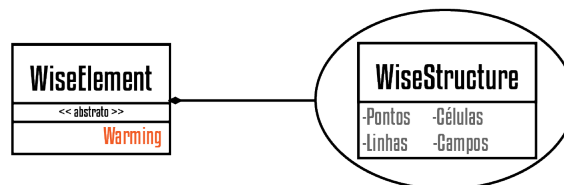
A estrutura contida em um elemento no estado *Warming* é igual à dos estados *Raw* e *Cooling*. Os estados diferem em finalidade, enquanto o estado *Warming* indica que o elemento está esperando a construção de sua estrutura abstrata (*DataStructure*). O estado *Raw* indica que não é esperado que os dados sejam consistentes e o estado *Cooling* indica que o elemento aguarda que seus dados sejam salvos em cache.

Como demonstrado na figura 6, o estado frio (*Cold*) está associado com o uso de um cache para elementos estruturais, onde somente a estrutura *WiseStructure* é salva. A estrutura representa um arquivo VTK mas são efetivamente salvos em um arquivo *XML*. Caso sejam novamente carregados por uma mudança de estado ou deletados o arquivo em

## WiseElement Statuses

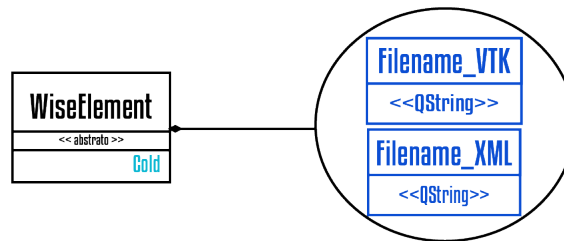


- Máquina de Status que controla o funcionamento de um elemento inteligente.



- Elemento inteligente enquanto no estado *Warming*.

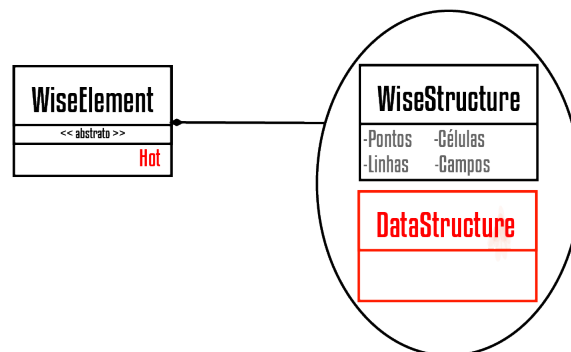
cache é deletado. Quando um elemento inteligente está neste estado ele contém apenas o endereço para o arquivo em que foi armazenado.



– Elemento inteligente enquanto no estado *Cold*.

O estado *Crashed* serve para identificar objetos que não tem mais o funcionamento esperado. Durante a troca de estados do elemento é sempre verificado se os elementos esperados estão presentes, caso não estejam o objeto fica com este estado.

Finalmente, o estado *Hot* representa os elementos que possuem todas as estruturas presentes. No caso de uma árvore arterial *WiseArteryTree* isto significa que a estrutura geométrica da *WiseStructure* está carregada e a estrutura abstrata equivalente *DataStructure* está presente e consistente.

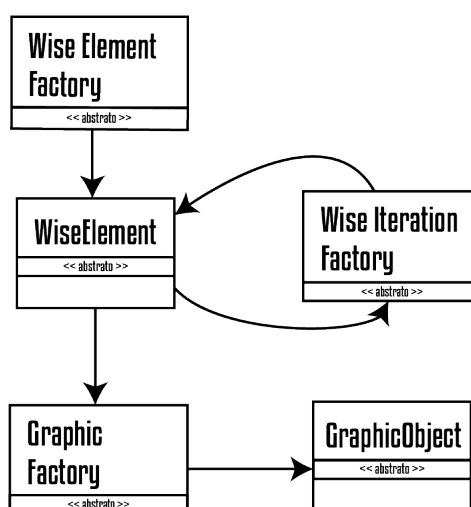


– Elemento inteligente enquanto no estado *Hot*.

Para que um elemento possa ser iterado por alguma fábrica do tipo *WiseIterationFactory* ele precisa estar no estado *Hot*, isto porque durante a iteração os dados abstratos são utilizados. A cada passo da iteração a estrutura *DataStructure* é atualizada, exigindo uma atualização da estrutura *WiseStructure*. Apenas a estrutura de um elemento inteligente não é capaz de armazenar todos os passos da iteração, isto é, a cada atualização os dados são sobrescritos.

### 3.1.2 Fábricas

Com estes passos em mente foi idealizada uma arquitetura de classes que permite a execução de cada passo através do paradigma de Fábricas Dinâmicas (??). Uma arquitetura com fábricas permite a criação de instâncias com definições concretas, armazenadas como metadados. Isso facilita a adição de novos objetos que podem ser interpretados sem modificar o código da fábrica em si.



– Arquitetura de classes fábrica e fluxo de trabalho do elemento inteligente *WiseElement*. A fábrica *WiseElementFactory* é responsável por criar o elementos inteligentes, a fábrica *WiseIterationFactory* é responsável pela iteração do elemento inteligente e a fábrica *GraphicFactory* é responsável por criar as estruturas de visualização.

Primeiramente, a fábrica *WiseElementFactory* é utilizada para a criação de elementos inteligentes, ao se criar um elemento à partir de um exemplo é preciso identificar o tipo de objeto à ser criado e qual exemplo será utilizado. Caso seja um arquivo *VTK* é necessário informar apenas o tipo e caminho do arquivo. Com um arquivo *XML* só o caminho do arquivo é necessário, pois a fábrica ao ler o XML identifica o tipo de elemento inteligente e identifica qual fábrica é a responsável por recriar aquele tipo de objeto. Devido à forma como os dados são carregados para cada elemento, é necessário que haja uma fábrica para cada tipo de elemento inteligente.

A fábrica *WiseIterationFactory* também só pode operar com um tipo específico de elemento inteligente, entretanto é possível que haja mais de uma fábrica de iteração disponível por tipo de elemento inteligente. Desta forma uma árvore arterial pode ser

iterada por diferentes algoritmos de iteração e o mesmo ocorre com os outros tipos de elementos. Estas fábricas são responsáveis por executar algum algoritmo que utilize o tipo de dados do elemento inteligente. No caso de uma árvore arterial é possível utilizar uma fábrica que irá executar o modelo matemático descrito na seção 2.4 utilizando os ponteiros para segmentos disponíveis em uma *WiseArteryTree*.

Por último, a fábrica *GraphicFactory* irá criar o objeto gráfico correspondente ao elemento inteligente. Assim como um elemento inteligente um objeto gráfico pode ser salvo em cache, contudo somente em um arquivo *XML*. Um elemento inteligente é composto por todas as linhas, pontos, células e seus valores associados, enquanto um objeto gráfico contém todos os elementos gráficos, como esferas, pontos e outros elementos gráficos, entretanto armazena apenas um valor associado à cada elemento gráfico. O que significa que só é possível selecionar um parâmetro por vez ao visualizar um objeto gráfico.

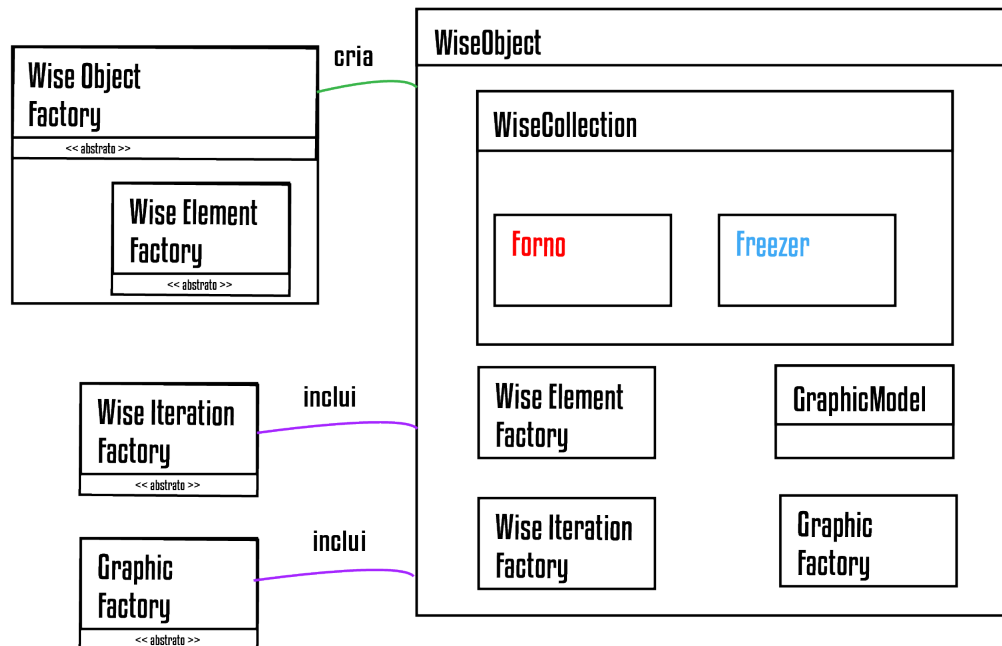
### 3.1.3 Objeto Inteligente

Para preservar todos os passos de iteração e poupar a quantidade de recursos mantida em memória, a classe *WiseObject*, ou objeto inteligente foi idealizada. Um objeto inteligente é composto por uma coleção de elementos inteligentes e objetos gráficos equivalentes entre si. Neste tipo de objeto é facultativa a presença de uma fábrica gráfica, possibilitando que objetos sejam iterados sem que alguma estrutura seja disponibilizada para visualização. Foi utilizada novamente uma arquitetura de fábricas para garantir que um objeto inteligente seja criado corretamente, pode-se criar um objeto inteligente através de um elemento inteligente previamente definido, este elemento será clonado e uma instância será aquecida na estrutura *Forno* enquanto a outra será congelada na estrutura *Freezer*.

Através do modelo de classes de um objeto inteligente presente na Figura 11 é possível identificar todos os componentes presentes em um objeto inteligente. Quando se gera um objeto inteligente é incluída a fábrica de criação de elementos inteligentes correspondente, *WiseElementFactory*, que é capaz de criar elementos vazios ou cloná-los. A coleção de elementos inteligentes, *WiseCollection*, contém duas outras estruturas, um forno e um freezer. Estas estruturas são um ponteiro para um elemento obrigatoriamente no estado quente e uma lista de objetos frios, respectivamente. O forno apresenta com o elemento que será utilizado durante as iterações, enquanto o freezer é responsável por gerir as estruturas que foram geradas previamente.

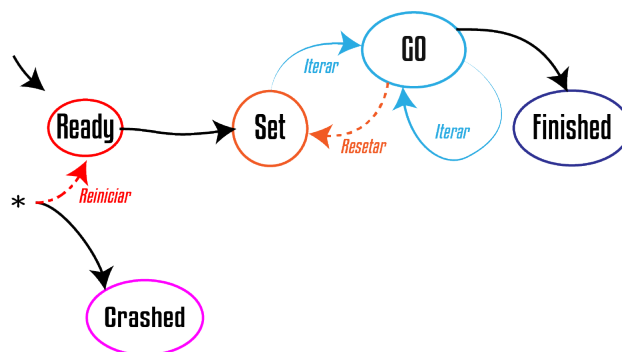
Objetos inteligentes também tem seu funcionamento descrito por uma máquina de estados. Diferentemente dos estados de um elemento inteligente os estados de um objeto inteligente não tem relação com os dados armazenados em cache ou com os dados abstratos. Os estados de um objeto inteligente regulam o processo iterativo e de visualização de um objeto inteligente.





- Objeto inteligente *WiseObject* e todos seu componentes: *WiseObjectFactory*, fábrica responsável pela criação de objetos inteligentes de um determinado tipo; *WiseIterationFactory*, fábrica de iteração; *WiseGraphicFactory*, fábrica gráfica; *WiseCollection*, coleção de elementos inteligentes; *GraphicModel*, coleção de objetos gráficos.

### WiseObject Statuses



- Modelo gráfico *GraphicModel*, contém uma coleção de objetos gráficos.

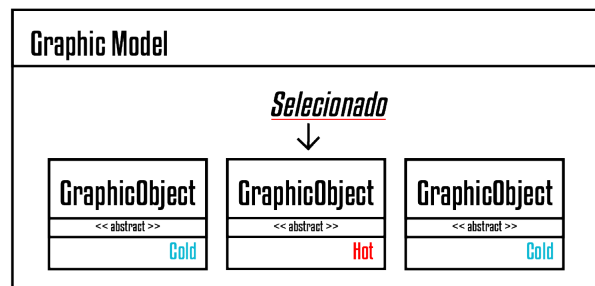
Diferentemente da troca de estados presente nos elementos inteligentes, aqui a troca se dá por comandos do usuário. Portanto cabe ao usuário indicar as fábricas à serem adicionadas bem como os valores de parâmetros desejados. Para o caso de escoamento pulsátil através de uma árvore arterial é necessário se adicionar a fábrica de iteração correspondente ao algoritmo da seção 2.4 e em seguida definir os parâmetros desejados, como a frequência, a viscosidade e o ângulo de fase. Quando as alterações são concluídas pelo usuário é ele que faz também a mudança de estados.

Inicialmente, um objeto é criado no estado *Ready* com somente dois elementos gráficos e uma fábrica do tipo *WiseElementFactory*. Neste estado é esperada a inclusão das fábricas de iteração e gráficas. Uma vez que elas estejam corretamente acopladas ao objeto inteligente, é possível fazer a troca do estado *Ready* para o estado *Set*. Com a mudança de estado, é adicionado à estrutura *WiseStructure* todos os parâmetros disponibilizados pela fábrica de iteração e, caso tenha sido incluída, fábrica gráfica. Um objeto no estado *Set* indica que o objeto foi corretamente criado, uma fábrica de iteração foi adicionada, possivelmente uma fábrica gráfica também foi adicionada e agora aguarda alterações nestes parâmetros.

Com os parâmetros definidos e as fábricas devidamente acopladas o objeto está pronto para a iteração. O processo iterativo de um objeto inteligente é representado na transição para o estado *Go*. Uma iteração de uma *WiseArteryTree* representa o cálculo dos valores de pressão e fluxo em toda a árvore arterial. Caso algum erro ocorra durante o processamento de dados o objeto se desloca para o estado *Crashed*, assim como elementos inteligentes. É possível também finalizar a execução de um objeto inteligente o enviando para o estado *Finished*, neste estado o objeto não poderá ser iterado novamente. O objeto inteligente também pode ser reiniciado à partir de qualquer estado, o que significa que todas as alterações feitas na estrutura do objeto inteligente serão apagadas. Para que parâmetros da iteração possam ser alterados sem que se perda os dados até então definidos é possível que um objeto no estado *Go* seja resetado e transite para o estado *Set*.

#### 3.1.4 Objeto Gráfico

Caso seja requisitado o objeto pode também incluir os componentes de visualização fábrica gráfica e coleção de objetos gráficos *GraphicModel*. Somente depois de adicionar uma fábrica gráfica à um objeto que a coleção de objetos gráficos é alocada e o primeiro objeto gráfico criado. Objetos gráficos assim como elementos inteligentes são colecionados por outras estruturas. Enquanto a estrutura responsável por elementos inteligentes, *WiseCollection*, é responsável por manter os últimos dados de iteração aquecidos, a estrutura *GraphicModel* mantém o objeto que está sendo exibido por alguma tela e elementos próximos.



- Modelo gráfico *GraphicModel*, contém uma coleção de objetos gráficos.



## 4 CONCLUSÃO

Este elemento é obrigatório e é a parte final do texto. Nele, são apresentadas as conclusões identificadas a partir do desenvolvimento da pesquisa.



## REFERÊNCIAS

- ANLIKER R.L. ROCKWELL, E. O. M. Nonlinear analysis of flow pulses and shock waves in arteries. *ZAMP*, n. 22, p. 217–246, 1971.
- AVOLIO, A. Multi-branched model of the human arterial system. *Med. Biol. Eng. Comput*, n. 18, p. 709–718, 1980.
- BEEK S. A. ROGER, J. B. B. J. H. G. M. V. Regional myocardial flow heterogeneity explained with fractal networks. *American Journal of Physiology*, n. 257, p. H1670–H1680, 1989.
- BLANCO, P. *Incompatibilidade cinemática, imersão de domínios e modelagem constitutiva de multiescala: nexo com a modelagem do sistema cardiovascular humano*. Tese (Doutorado) — Laboratório Nacional de Computação Científica, 2008.
- DAWANT M. LEVIN, A. P. B. Automatic construction of 3d models of arterial tree incorporating the fahraeus-lindqvist effect. *Revista Eletrônica Paulista de Matemática*, n. 10, p. 38–49, 2017.
- DUAN, M. Z. B. *Biodynamics: Circulation*. New York: Springer-Verlag, 1984.
- DUAN, M. Z. B. Effect of dispersion of vessel diameters and lengths in stochastic networks. i. modeling of microcirculatory flow. *Microvascular Research*, n. 31, p. 203–222, 1986.
- DUAN, M. Z. B. Viscous damping in one dimensional wave transmission. *J. Acoust. Soc. Am.*, n. 92, p. 3358–3363, 1995.
- FORMAGGIA D. LAMPONI, A. Q. L. One-dimensional models for blood flow in arteries. *Journal of Engineering Mathematics*, n. 47, p. 251–276, 2003.
- FORMAGGIA J.F. GERBEAU, F. N. A. Q. L. Computer methods in applied mechanics and engineering. *Revista Eletrônica Paulista de Matemática*, n. 191, p. 561–582, 2001.
- GROUP, K. *OpenGL*. 2019. Disponível em: <<https://www.opengl.org/>>.
- HAREL P.J. BOLAN, R. T. K. U. E. Y. N. Recent advances in high-resolution mr application and its implications for neurovascular coupling research. *Front Neuroenergetics*, p. 2, 130, 2010.
- KARCH F. NEUMANN, M. N. W. S. R. A three-dimensional model for arterial tree representation, generated by constrained constructive optimization. *Computers in biology and medicine*, n. 29, p. 19–38, 1999.
- KARREMAN, G. Some contributions to the mathematical biology of blood circulation. reflection of pressure waves in the arterial system. *Bull. Math. Biophys.*, n. 14, p. 327–350, 1952.
- KOUCHOUKOS L.C. SHEPPARD, D. A. M. N. Estimation of stroke volume in the dog by a pulse contour method. *Circ. Res.*, n. 26, p. 611–23, 1970.
- LIGHTHILL, M. *Mathematical biofluidmechanics*. Philadelphia: Society for Industrial & Applied Mathematics, 1975.

- MATES F.J. KLOCKE, J. C. R. Coronary capacitance. *Progress in Cardiovascular Diseases*, n. 31, p. 1–15, 1988.
- MCDONALD, D. A. Blood flow in arteries. *Baltimore: Williams & Wilkins*, 1974.
- PARKER, A. *Algorithms and Data Structures in C++*. [S.l.: s.n.], 1959.
- PESKIN, C. Flow patterns around heart valves: a numerical method. *J. Comput. Phys.*, n. 10, p. 252–271, 1972.
- QT. 2019. Disponível em: <<https://doc-snapshots.qt.io/qt5-5.9/classes.html>>.
- QUEIROZ, R. *Construção automática de modelos de árvores circulatórias e suas aplicações em hemodinâmica computacional*. Tese (Doutorado) — Laboratório Nacional de Computação Científica, 2013.
- SCHREINER, P. B. W. Computer-optimization of vascular trees. *IEEE Transactions on Biomedical Engineering, IEE*, n. 440, p. 482–491, 1993.
- STERGIOPULOS D.F. YOUNG, T. R. R. N. Computer simulation of arterial flow with applications to arterial and aortic stenoses. *Journal of Biomechanics*, n. 25, p. 1477–1488, 1992.
- TAYLOR, M. The input impedance of an assembly of randomly branching elastic tubes. *Biophys. J*, n. 6, p. 29–51, 1966.
- TAYLOR T.J.R. HUGHES, C. Z. C. Finite element modeling of three-dimensional pulsatile flow in abdominal aorta: relevance to atherosclerosis. *Annals of Biomedical Engineering*, n. 26, p. 975–987, 1998.
- URQUIZA P.J. BLANCO, M. V. R. F. S. Multidimensional modelling for carotid artery blood flow. *Computer Methods in Applied Mechanics and Engineering*, n. 195, p. 4002–4017, 2006.
- ZAMIR, M. ptimality principles in arterial branching. *J. Theor. Biol.*, n. 62, p. 227–251, 1976.
- ZAMIR, S. P. M. Network analysis of an arterial tree. *American Journal of Physiology*, n. 21, p. 25–34, 1988.



## APÊNDICE A – Título

Este elemento é opcional. Apresenta um texto ou documento elaborado pelo autor com o objetivo de complementar sua argumentação, sem prejuízo da unidade nuclear do trabalho.



## ANEXO A – Título

Este elemento é opcional. Apresenta um texto ou documento **não** elaborado pelo autor com o objetivo de complementar ou comprovar sua argumentação.