Uma Ferramenta Computacional para Simulação de Escoamento Pulsátil em Modelos de Árvores Arteriais 1D

Igor Pires dos Santos
igor.pires@ice.ufjf.br
Orientadores: Rafael Alves Bonfim de Queiroz
& Ruy Freitas Reis

Programa de Pós-Graduação em Modelagem Computacional
Universidade Federal de Juiz de Fora

20 de setembro de 2021

Sumário

Introdução

- 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.

- A propagação de ondas em um tubo é governada pela equação de onda para a pressão $p(x,t)$ e volume de fluxo $q(x,t)$, ambas sendo funções no tempo t e coordenada axial x ao longo do tubo.

$$\frac{\partial q}{\partial t} = -cY\frac{\partial p}{\partial x} \tag{1}$$

$$\frac{\partial p}{\partial t} = -\frac{c}{Y}\frac{\partial p}{\partial x} \tag{2}$$

- Para uma onda harmônica simples Eq.1 e Eq.2 ficam na forma:

$$p = \bar{p}_0 \exp\{i\omega(t - x/c)\} + R\bar{p}_0 \exp\{i\omega(t - 2L/c + x/c)\} \tag{3}$$

$$q = Y(\bar{p}_0 \exp\{i\omega(t - x/c)\} - R\bar{p}_0 \exp\{i\omega(t - 2L/c + x/c)\}) \tag{4}$$

Introdução (cont.)

- Para definir os valores do coeficiente de reflexão R e pressão média $\bar{p}_0$ de cada segmento foi utilizado o modelo matemático descrito por Duan & Zamir (1995).
- Este modelo matemático possui uma complexidade pequena e é capaz de ser iterado muito rapidamente para modelos geométricos numerosos.

- Inicialmente se definem as propriedades características de cada segmento :
- Espessura da parede (h):
-

$$h = 0.1 \times r \tag{5}$$

- Velocidade de Onda (c):
-

$$c = \sqrt{\frac{Eh}{\rho 2r}} \tag{6}$$

Modelo Matemático (cont.)

- Velocidade angular (ω):
- $$\omega = 2\pi f \tag{7}$$
- Beta (β):
- $$\beta = \omega \frac{L}{c} \tag{8}$$
- Admitância Característica (Y):
- $$Y = \frac{\pi r^2}{\rho c} \tag{9}$$

- **Caso 2:** Ângulo de fase
- Módulo de Young (E_c):
- $$E_c = \|E_c\| \exp\{i\phi\} \tag{10}$$
- Ângulo de Fase (ϕ):
- $$\phi = \phi_0(1 - \exp\{-w\}) \tag{11}$$

- **Caso 3:** Viscoso
- Velocidade de Onda Viscosa (c_v):
-

$$c_v = c\sqrt{\epsilon} \tag{12}$$

-
- Admitância (Y_v):
-

$$Y_v = Y\sqrt{\epsilon} \tag{13}$$

Modelo Matemático (cont.)

- Alpha (α):
- $$\alpha = R\sqrt{\frac{\omega\rho}{\mu}} \tag{14}$$
- Fator Viscoso (ϵ):
- $$\epsilon = 1 - F_{10}(\alpha) \tag{15}$$
- Função auxiliar do Fator Viscoso (F_{10}):
- $$F_{10}(\alpha) = \frac{2.0}{\alpha\sqrt{i}}(1.0 + \frac{1.0}{2.0\alpha}), \tag{16}$$

Modelo Matemático (cont.)

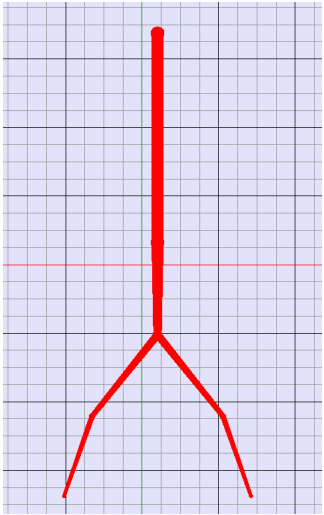
- **Reflection Coefficient «complex» e Admittance «complex»**
- Se folha $R = 0$, senão $R = \frac{Y-(Ye_r+Ye_l)}{Y+(Ye_r Ye_l)}$.
- Se folha $Ye = Y$, senão $Ye = Y * \frac{(1-R\exp\{-2i\beta\})}{(1+R\exp\{-2i\beta\})}$.
- **Medium Pressure «complex»**
- Se raiz $\bar{P} = \bar{P}_0$, senão $\bar{P} = \bar{P}_f * \frac{((1+R_f)\exp\{-i\beta_f\})}{(1+R\exp\{-2i\beta\})}$.

Modelo Matemático (cont.)

- **Pressure «complex»**
- $P = \bar{P} * (\exp\{-i\beta X\} + R\exp\{-i2\beta\}\exp\{i\beta X\})$.
- **Flow «complex»**
- $Q = M\bar{P} * (\exp\{-i\beta X\} - R\exp\{-i2\beta\}\exp\{i\beta X\})$.
- $M = \frac{Y}{Y_r}$

Exemplo

- Considera-se o exemplo de árvore arterial extraído extaído de Duan & Zamir (1995).



Exemplo (cont.)

Duan & Zamir
A árvore arterial conta com:

- 2 Artérias terminais.
- 6 segmentos totais (2 pares idênticos).
- Considerando o caso não-viscoso com ângulo de fase $\phi = 0$
- Observa-se o acoplamento feito para o caso viscoso e a variação do ângulo de fase.

O exemplo considera ainda o caso viscoso, o não viscoso e $\phi \in 0, 4, 8, 12$. O algoritmo proposto se dividiu em duas partes, a primeira com 7 variáveis a se definir e a segunda com 5. Para demonstrar o modelo matemático o caso não viscoso com $\phi = 0$ foi escolhido.

Exemplo (cont.)

- **Parâmetros de Entrada (r(cm),L(cm),ρ(g/cm^3),E(g/$cm * s^2$),f(Hz),ϵ,μ_0,ϕ_0)**
- $f = 3.65$Hz,$\epsilon = 0$,$\mu_0 = 0$,$\phi_0 = 0$.

0 $= (r = 0.65)$, $(L = 25)$,$(\rho = 0.96)$ e $(E = 4.8 * 10^6)$.

1 $= (r = 0.45)$, $(L = 11)$,$(\rho = 1.134)$ e $(E = 10^7)$.

2 $= (r = 0.3)$, $(L = 12)$,$(\rho = 1.172)$ e $(E = 10^7)$.

3 $= (r = 0.2)$, $(L = 10)$,$(\rho = 1.235)$ e $(E = 10^7)$.

Exemplo (cont.)

- **Wall Thickness (h)(cm)**
- $h = 0.1 * r$.

0 $= 0.065$.

1 $= 0.045$.

2 $= 0.03$.

3 $= 0.02$.

Exemplo (cont.)

- **Wavespeed (cm/s)**
- $C = \sqrt{\frac{Eh}{\rho 2r}}$.

0 = 500.

1 = 664.0158940747.

2 = 653.1624303415.

3 = 636.2847629758.

Exemplo (cont.)

- **Angular Frequency (ang/s)**
- $\omega = 2\pi f$.

0 = 22.9336263712.

1 = 22.9336263712.

2 = 22.9336263712.

3 = 22.9336263712.

Exemplo (cont.)

- **Beta «complex»**
- $\beta = \omega \frac{L}{c}$.

0 = (1.1466813186 , 0).

1 = (0.3799154393 , 0).

2 = (0.4213400790 , 0).

3 = (0.3604302299 , 0).

Exemplo (cont.)

- **Admittance «complex»**
- $Y = \frac{\pi r^2}{\rho c}$.

$0 = (0.0027652560\ ,\ 0)$.

$1 = (0.00084485573\ ,\ 0)$.

$2 = (0.0003693547\ ,\ 0)$.

$3 = (0.0001599158\ ,\ 0)$.

Exemplo (cont.)

- **Reflection Coefficient «complex»**
- Se folha $R = 0$, senão $R = \frac{Y-(Ye_r+Ye_l)}{Y+(Ye_r Ye_l)}$.

0 = (0.6262367793 , -0.2822851808).

1 = (0.3301552903 , -0.2838246367).

2 = (0.3957123386 , 0).

3 = (0 , 0).

Exemplo (cont.)

- **Effective Admittance «complex»**
- Se folha $Ye = Y$, senão $Ye = Y * \frac{(1-R\exp\{-2i\beta\})}{(1+R\exp\{-2i\beta\})}$.

0 = (0.0066356520 , 0.0071130215).

1 = (0.0005360764 , 0.0005730514).

2 = (0.0001850690 , 0.0001296261).

3 = (0.0001599158 , 0).

Exemplo (cont.)

- **Medium Pressure «complex»**
- Se raiz $\bar{P} = \bar{P}_0$, senão $\bar{P} = \bar{P}_f * \frac{((1+R_f)\exp\{-i\beta_f\})}{(1+R\exp\{-2i\beta\})}$.

0 = (1.0 , 0.0).

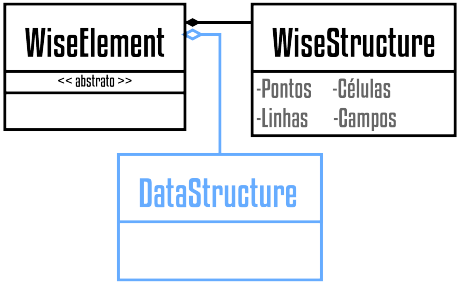
1 = (0.0005360764 , 0.0005730514).

2 = (0.0001850690 , 0.0001296261).

3 = (0.0001599158 , 0).

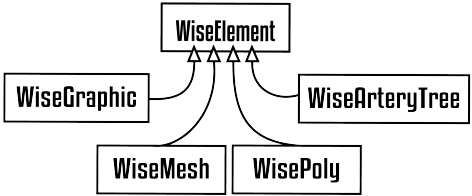
Modelo Computacional

- Um elemento inteligente (*WiseElement*) possui duas estruturas básicas. A *WiseStructure* que representa os dados dispostos no padrão *VTK*(*Visualization Toolkit*) e *DataStructure* que representa os dados abstratos disponíveis na estrutura.



Modelo Computacional (cont.)

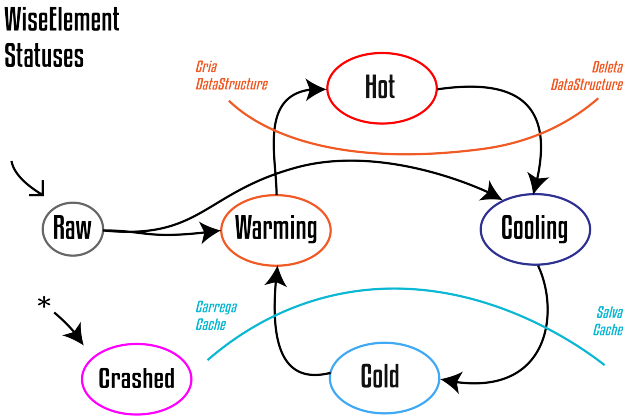
- Todos os elementos inteligentes recebem a estrutura básica *WiseStructure* por herança e requer por meio de funções virtuais a definição de métodos que possibilitem a manipulação dos dados abstratos.



Modelo Computacional (cont.)

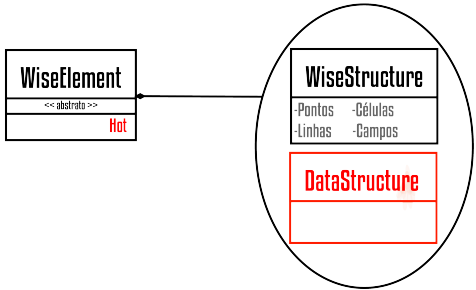
- Os elementos inteligentes são regidos por uma máquina de estados, aonde os estados representam condições esperadas do elemento inteligente e as transições indicam ações tomadas com o elemento inteligente.

Modelo Computacional (cont.)



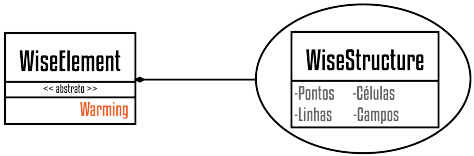
Modelo Computacional (cont.)

- Neste estado espera-se que um elemento possua consigo ambas as estruturas presentes no elemento inteligente, seus dados abstratos e a estrutura inteligente *WiseStructure*.



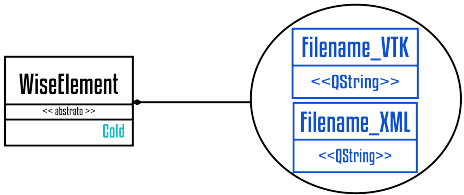
Modelo Computacional (cont.)

- O estado *Warming* é equivalente ao estado *Cooling* e *Raw*.
- Estes estados indicam que somente a estrutura inteligente deste elemento está presente. No caso de um elemento no estado *Raw*, não é esperado que a estrutura completa esteja presente nesta estrutura.
- Os outros estados indicam que o elemento está completamente carregado na estrutura inteligente e aguarda esfriamento ou aquecimento, processo de armazenar e recuperar arquivos em HD.



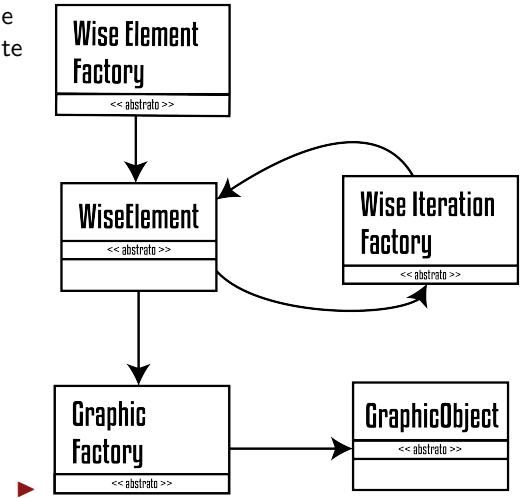
Modelo Computacional (cont.)

- Espera-se que os elementos neste estado estejam salvos em HD.



Modelo Computacional (cont.)

- Utilizando o padrão de fábricas para criar e manipular elementos inteligentes, o seguinte fluxo de trabalho foi idealizado:

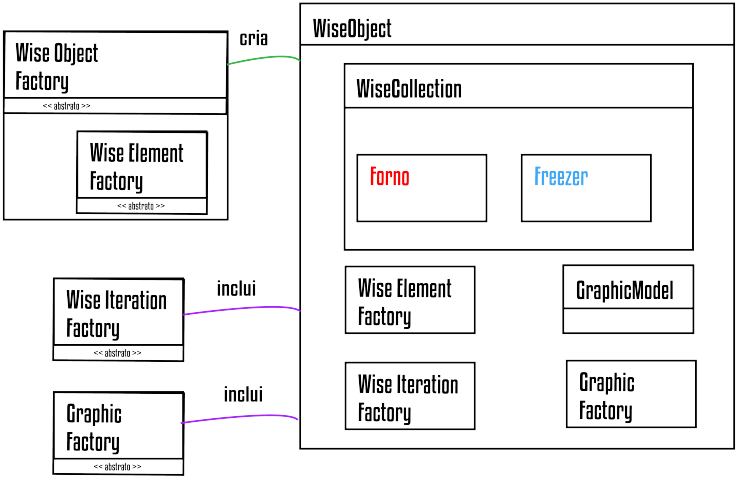


Modelo Computacional (cont.)

- A primeira fábrica *WiseElementFactory* é responsável por criar corretamente cada tipo de elemento inteligente.
- A fábrica *WiseIterationFactory* tem a função de utilizar os dados abstratos de um elemento inteligente com a finalidade de executar algum algoritmo.
- Finalmente, a fábrica *GraphicFactory* gera os objetos capazes de se desenhar com diretivas OpenGL *GraphicObject*.

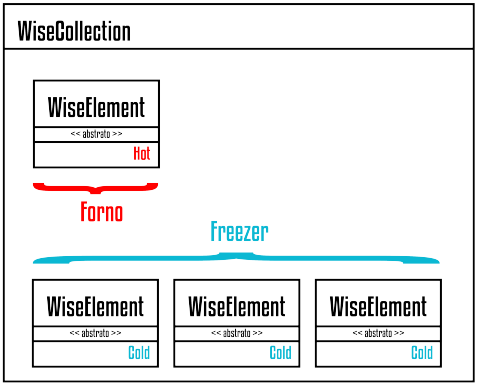
Modelo Computacional (cont.)

- O objeto inteligente *WiseObject* é o objeto capaz de executar todo esse fluxo de trabalho.



Modelo Computacional (cont.)

- Cada objeto inteligente contém uma coleção de elementos inteligentes:

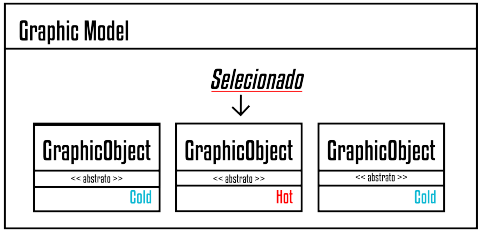


Modelo Computacional (cont.)

- Pode-se criar um objeto inteligente com um elemento inteligente, portanto é possível construir um objeto inteligente à partir de qualquer instância de elemento inteligente e sua fábrica.
- A estrutura *Forno* é um ponteiro para um elemento que estará sempre no estado *Hot*, ele representa o último elemento gerado pela iteração e é utilizado a cada nova iteração.
- Finalmente, a estrutura *Freezer* é responsável por armazenar os elementos em cache e manter o seu registro.

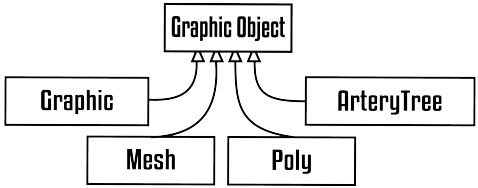
Modelo Computacional (cont.)

- É possível também que seja incluída a estrutura de visualização do objeto inteligente:



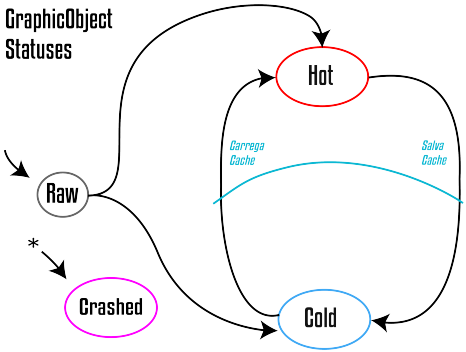
Modelo Computacional (cont.)

- Os objetos gráficos são objetos que possuem uma lista de elementos gráficos com valores e é capas de desenhá-los em um quadro OpenGL.
- O modelo gráfico *GraphicModel* irá manter a coleção de objetos gráficos em memória conforme a necessidade para visualização, mantendo uma quantidade mínima de objetos em memória a todo momento.

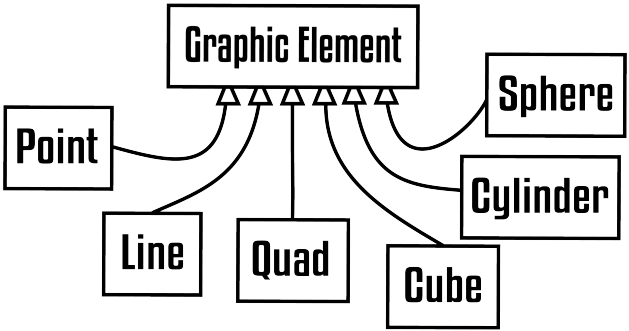


Modelo Computacional (cont.)

- Objetos gráficos também possuem uma máquina de estados que dita se a estrutura está presente em memória ou armazenada em cache:



- Os elementos gráficos são todas as instâncias que implementam a classe abstrata *GraphicElement*:

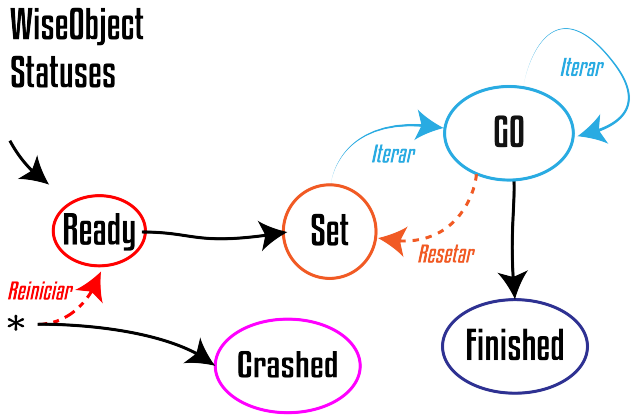


Modelo Computacional (cont.)

- Os elementos gráficos armazenam os pontos necessários para desenhá-lo e um valor associado. Este valor associado corresponde à algum valor armazenado em um ponto, uma linha, uma célula ou um campo da *WiseStructure*.

Modelo Computacional (cont.)

- Os objetos inteligentes *WiseObject* possuem também uma máquina de estados:

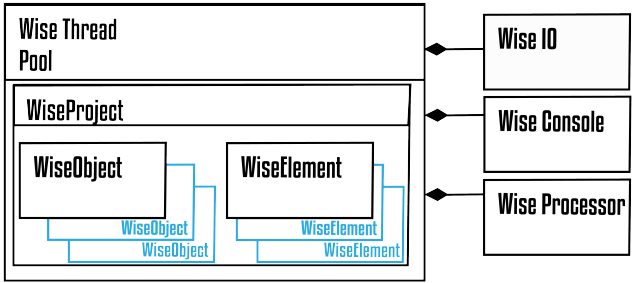


Modelo Computacional (cont.)

- Todas as transições de estados e as ações que podem ser executadas sobre um objeto inteligente são interações do usuário. Ao se criar o objeto somente o elemento inicial e sua cópia estarão presentes no *Forno* e no *Freezer*, respectivamente.
- No estado inicial *Ready* o objeto é capaz de incluir suas fábricas de iteração e gráficas.
- Após a inclusão de uma fábrica de iteração o objeto pode avançar para o estado *Set*. Neste estado os parâmetros de iteração são adicionados à estrutura *WiseStructure* e podem ser editados pelo usuário, parâmetros como frequência, viscosidade e ângulo de fase.
- Pode-se arbitrariamente definir o fim das iterações e enviar o objeto para o estado *Finished*, que impossibilitará o objeto de continuar iterando.
- Finalmente, todos os estados podem levar ao estado *Crashed* que indica o mau funcionamento do objeto.

Modelo Computacional (cont.)

- Com todas as estruturas básicas definidas, o pacote de classes que organiza or objetos e elementos inteligentes e suas necessidades intitula-se *WiseThreadPool*.

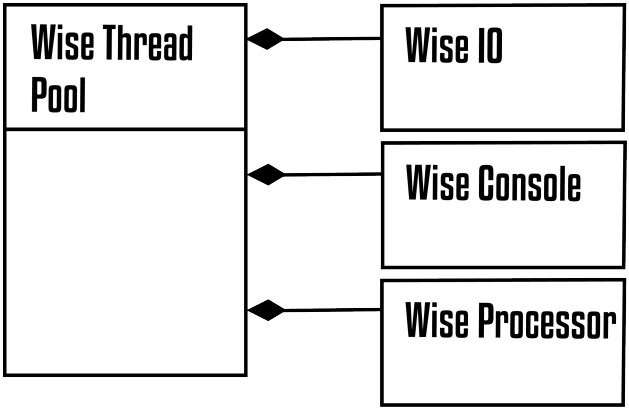


Modelo Computacional (cont.)

- Os objetos e elementos inteligentes organizam-se em projetos inteligentes, *WiseProject*.
- A classe *WiseThreadPool* é responsável por alocar estes projetos e receber suas demandas, resfriar ou aquecer objetos. É responsável também pelas demandas feitas pelo usuário através de linhas de comando.

Modelo Computacional (cont.)

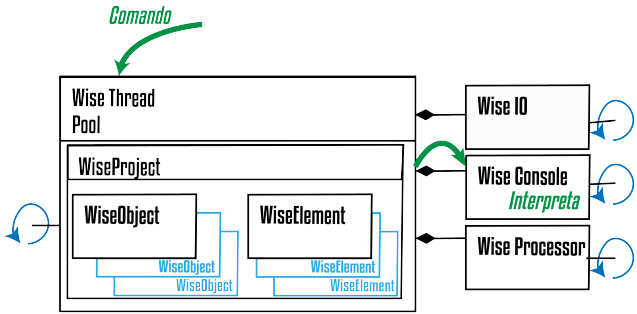
- Cada uma das classes à seguir está contida em uma *thread* própria e tem o próprio *loop* de iteração.



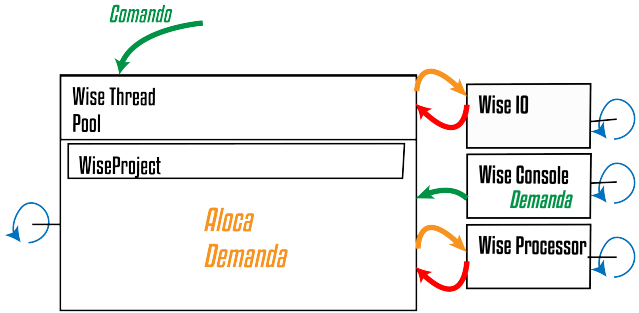
Modelo Computacional (cont.)

- Ao receber uma linha de comando o objeto *WiseThreadPool* enviar a mensagem e o projeto atual à uma instância de *WiseConsole*, que é responsável por interpretar o comando.
- Caso seja um processo que utilize o disco rígido, uma instância *WiseIO* será necessária para executar o comando.
- Caso seja um processo de iteração, uma instância *WiseProcessor* será utilizada.
- Para os restantes dos casos a própria classe *WiseConsole* será utilizada para finalizar o comando. A alteração de parâmetros ou sua escala cai neste último caso.

Modelo Computacional (cont.)

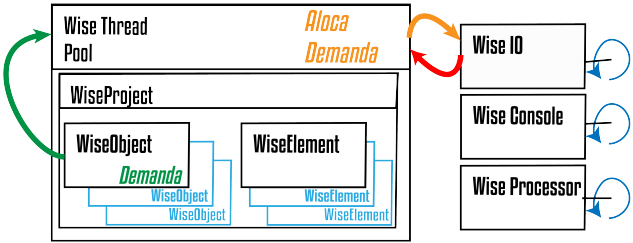


Modelo Computacional (cont.)



Modelo Computacional (cont.)

- O resfriamento ou aquecimento de elementos envolve sempre uma instância de *WiseIO*.



Modelo Computacional (cont.)

- Um *WiseThreadPool* irá conter todas as instâncias de *WiseIO*, *WiseConsole* e *WiseProcessor* e está preparado para utilizar mais de uma instância destes objetos por vez.
- Essencialmente isto significa que os objetos e elementos armazenados pelo projeto inteligente, *WiseProject*, podem ser acessados pelos diferentes processadores independentemente.
- Isto dá ao usuário uma imensa liberdade na hora de decidir executar algum algoritmo concorrentemente, entretanto requer cuidados especiais na hora de utilizar os objetos.
- Como os objetos podem ser acessados por mais de um processador, eles são bloqueados por uso.

Ferramenta Computacional

- Com a estrutura de dados definida a ferramenta foi dividida em dois ambientes:
- A interface gráfica ***IGU*** (**I**nterface **G**ráfica **U**niversal), que representa o ambiente gráfico da ferramente.
- E, o console ***InGU***(**I**nterface **n**ão-**G**ráfica **U**niversal), que é um ambiente executado sem interface gráfica e recebe comandos via texto.

Ferramenta Computacional (cont.)

- O ambiente **IngU** ao ser executado imprime no console o cabeçalho do programa e carrega os elementos previamente carregados na ferramenta.

```
.    /=====================================================================/
.   /====/       IGU (Iterador Gráfico Universal)       /==================/
.  /====/          (or, Universal Graphic Iterator)     /==================/
. /=====================================================================/
. /=1.1==============================================================/
. /=====================================================================/
. segunda-feira 09/08/2021 19:19:08 313
. /=====================================================================/
. [WISE CONSOLE] LOAD invoked
. <p1:WISE_PROJECT> 'WISE_ID [0] (NAME: p1) WISE PROJECT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [0] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_OBJECT> 'WISE_ID [0] (NAME: obj1) WISE OBJECT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [1] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [2] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [3] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [4] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [5] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [6] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [7] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [8] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [9] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [10] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [11] (NAME: el1) WISE ELEMENT CREATED'
. <el1:WISE_ELEMENT> 'WISE_ID [12] (NAME: el1) WISE ELEMENT CREATED'
```

Ferramenta Computacional (cont.)

- O comando de ajuda é o primeiro comando da interface e foi feito para listar todas as entradas possíveis do programa.
- Ao receber este comando a thread *WiseConsole* envia o texto pré-definido com todos os comandos.

Linha de Comando	help
Escopo	nenhum
Thread Responsável	WiseConsole
Entrada	Nenhuma

Tabela 1: Descrição do comando ajuda.

Ferramenta Computacional (cont.)

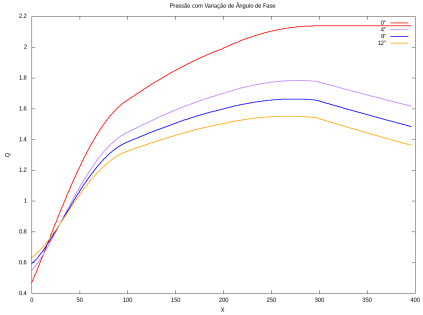
Assim como o comando de criação de elementos inteligentes, o comando de criação de objetos irá acessar a fábrica de elementos inteligentes *WiseElementFactory*. É possível criar objetos inteligentes de duas formas: A primeira, utilizando um elemento inteligente; A segunda utilizando os exemplos disponibilizados pela fábrica de elementos inteligentes.

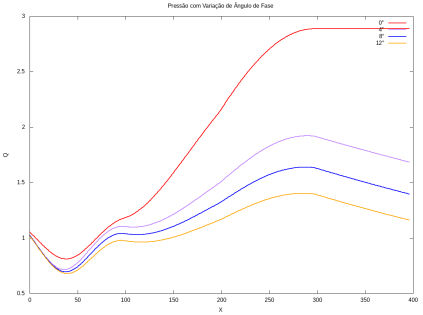
Assim como descrito na Seção **??**, ao criar um objeto inteligente, um elemento inteligente é adicionado à estrutura do *Forno*, enquanto um Clone é acoplado ao *Freezer*;

Linha de Comando	object create <object_name> <element_name> object create <type> <example> <name> <element_name> [ARGS]	
Escopo	OBJECT	
Thread Responsável	WiseConsole	
Entrada	<object_name>	Nome do objeto à ser criado.
	<element_name>	Nome do elemento à ser utilizado na criação ou do elemento à ser criado a partir do exemplo.
	<type>	Tipo de elemento inteligente à ser criado.
	<name>	Nome do exemplo de elemento inteligente à ser criado.
	[ARGS]	Individualmente, as fábricas podem receber parâmetros para a criação de elementos.

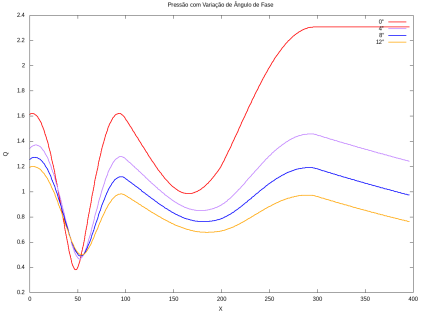
Tabela 2: Descrição do comando para criar objetos inteligentes.

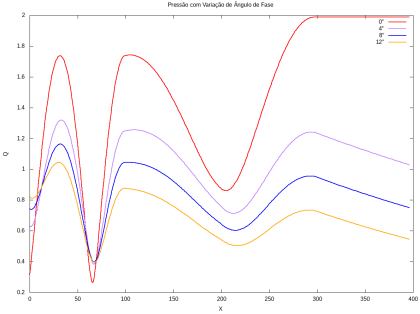
Resultados


Pressão com Variação do Ângulo de Fase
0°
4°
8°
12°
Q
X
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0
50
100
150
200
250
300
350
400

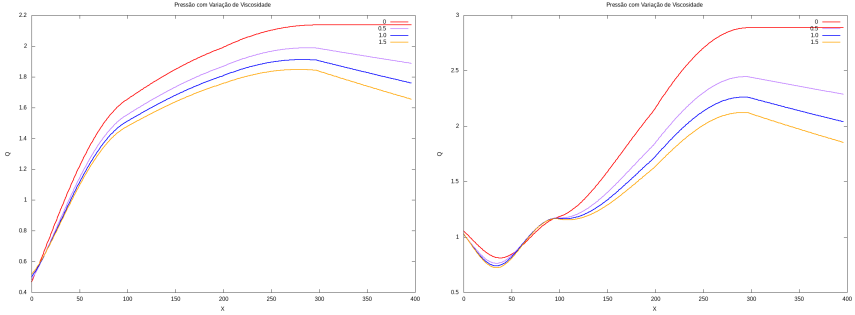

Pressão com Variação do Ângulo de Fase
0°
4°
8°
12°
Q
X
3
2.5
2
1.5
1
0.5
0
50
100
150
200
250
300
350
400

Resultados (cont.)

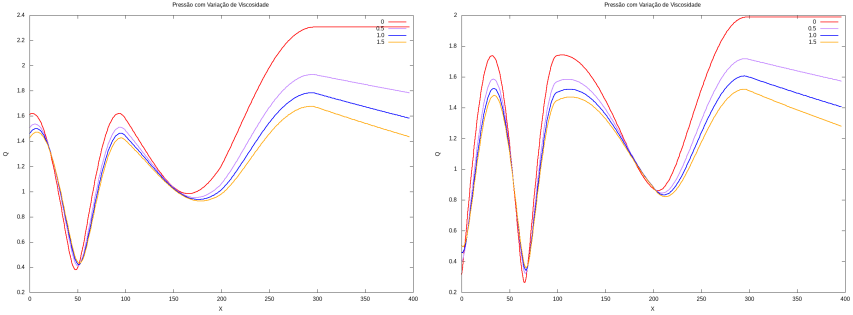




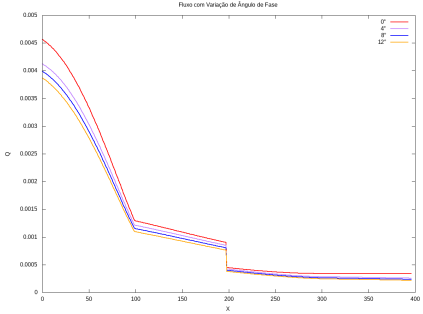
Resultados (cont.)

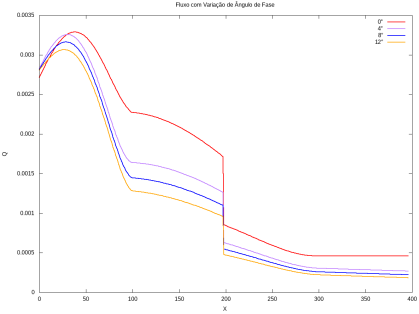


Resultados (cont.)

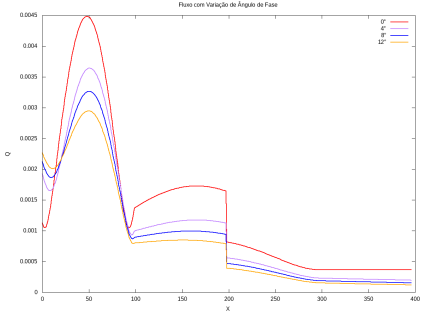


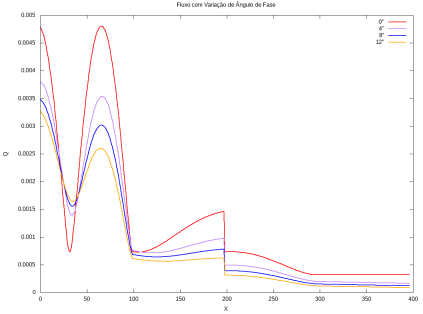
Resultados (cont.)



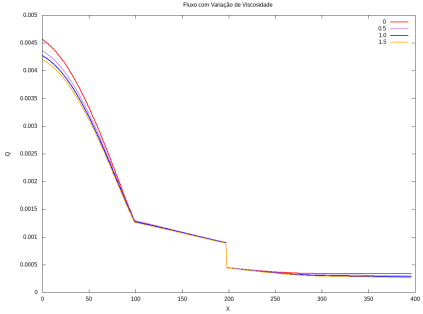


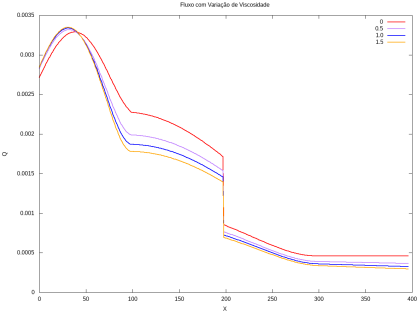
Resultados (cont.)



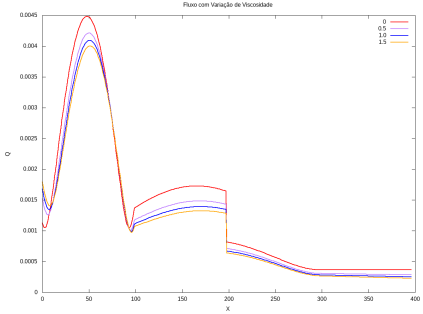


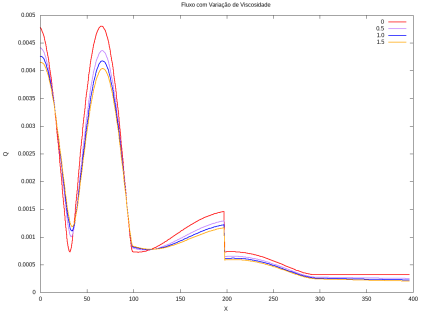
Resultados (cont.)





Resultados (cont.)





- Ferramenta Computacional
- Estrutura de dados
- **dividir em elemento inteligente, objeto inteligente e objeto gráfico?**
- (Sinais e slots, paralelização)
- Lista de comandos
- Interface gráfica
- Resultados
- Fluxo
- Pressão
- Conclusão

Cronograma

- Mar: Ajustes dissertação + Fim da dissertação
- Abr: Ajustes finais
- Mai: Ajustes finais