

Ferramentas para geração de modelos 3D do Sistema Cardiovascular Humano

Rodrigo L. S. Silva, Jan Palach, Ignacio Larrabide, Raul A. Feijóo
LNCC – Laboratório Nacional de Computação Científica
{rodrigo, palach, nacho, feij}@lncc.br

Abstract. *The increase of cardiovascular diseases in the last years has motivated the development of specific techniques for their treatment, as the implant of stents, clips and bypass. The complexity of these surgical interventions is notably growing in the last years due the fast advances in modern medicine. With the intention of joining tools to aid the treatment of these diseases, the HeMoLab was created. HeMoLab is a computer system that allows the creation of personalized models of the cardiovascular human system.*

Resumo. *O aumento na incidência das doenças cardiovasculares nos últimos anos tem motivado o desenvolvimento de técnicas específicas para o seu tratamento, sendo implantação de stents, clips e bypass alguns exemplos. A complexidade destas intervenções cirúrgicas tem crescido consideravelmente nos últimos anos devido, principalmente, ao rápido avanço da medicina moderna. Com o intuito de reunir ferramentas que auxiliem no tratamento desses problemas surgiu o HeMoLab – Laboratório de Modelagem em Hemodinâmica. O HeMoLab é um sistema computacional que permite a criação de modelos personalizados do sistema cardiovascular humano.*

1. Introdução

O crescimento da importância das doenças cardiovasculares nos últimos anos tem motivado o desenvolvimento de novas técnicas cirúrgicas para melhorar a qualidade de vida dos pacientes. A complexidade destas novas técnicas requer o uso de métodos que permitam prever com aceitável grau de precisão o resultado destes tipos de intervenções ao serem aplicadas em diferentes pacientes. A modelagem 3D do Sistema Cardiovascular Humano (SCVH) tem por objetivo estudar com alto grau de detalhe as características da dinâmica do fluxo sanguíneo e como estas características são modificadas ao se aplicar estas intervenções em pacientes reais em determinadas regiões do sistema arterial. A princípio, qualquer região do sistema arterial pode ser representada em detalhes aproveitando informações específicas do paciente como imagens médicas da área de interesse. Casos particulares como a existência de singularidades geométricas (aneurismas, estenoses, bifurcações, etc.) podem ser considerados a fim de se estudar alterações locais na estrutura do fluxo sanguíneo e de dar informações mais completas de forma quantitativa. Neste caso, é fundamental que as geometrias de interesse consigam representar fielmente a anatomia do paciente. Assim, os dados de entrada dos modelos serão a geometria de interesse, as condições de contorno naquela região e as propriedades mecânicas do sangue e dos vasos.

Depois de ter sido realizada a reconstrução da geometria, estabelecidas às condições de contorno e definidas as propriedades mecânicas da parede arterial e do sangue na área de interesse, o próximo passo será obter as soluções numéricas das equações transientes não lineares que governam o problema. Para obter resultados de qualidade nas simulações numéricas, as malhas de elementos finitos devem ser refinadas e processadas para que a sua resolução espacial permita capturar com suficiente nível de detalhe as características de interesse do fluxo sanguíneo.

O objetivo da ferramenta apresentada neste artigo é propiciar um ambiente capaz de fornecer, de forma interativa, funcionalidades que permitam gerar e refinar malhas de elementos finitos 3D. Este ambiente foi desenvolvido integrado ao *Paraview* [1].

Na seção seguinte, serão apresentados os trabalhos relacionados ao presente artigo. Em seguida, será explicada a arquitetura do *Paraview* e a forma como esta arquitetura foi utilizada neste trabalho. Na seção 4 será apresentado o Módulo 3D do ambiente *HeMoLab* [13] seguido de suas principais funcionalidades. Finalmente na Seção 6 serão apresentadas as conclusões deste artigo e as propostas de trabalhos futuros.

2. Trabalhos relacionados

Existem na atualidade algumas ferramentas de simulação do sistema cardiovascular humano. Um exemplo deste tipo de sistema é o *GEMSS* [6]. Nesta ferramenta, o sistema arterial foi representado por 121 seções separadas. As propriedades do modelo podem ser modificadas para simular vários estados de doença. Após ser especificado um fluxo cardíaco (o qual varia no tempo) como entrada do modelo, pode-se obter resultados da simulação. Os resultados são visualizados fazendo-se uso de uma interface gráfica que permite ao usuário medir fluxo e pressão em qualquer parte do sistema arterial simulado. O *GEMSS* também possui um módulo 3D, apesar de não possuir uma ferramenta que possibilite editar estas estruturas. Neste sistema, a simulação é feita utilizando ferramentas comerciais externas.

O sistema *SimTK* [7] é um projeto *open-source* que fornece um conjunto de ferramentas para simulações cardiovasculares. Possui recursos de abertura de imagens 3D, segmentação de estruturas, geração de modelos, malhas e modelagem do fluxo sanguíneo e vasos deformados. Neste caso, a simulação é realizada utilizando ferramentas comerciais.

A complexidade e tamanho do sistema cardiovascular faz com que a sua simulação computacional seja um problema de alta complexidade. Atualmente, o uso de modelos tridimensionais completos do sistema cardiovascular é proibitivo devido ao altíssimo custo computacional. Por este motivo, são comumente utilizados modelos multidimensionais [13] que representam a maior parte do sistema com modelos simplificados unidimensionais do fluxo sanguíneo e as regiões de interesse com modelos completos de fluídos (equações de Navier-Stokes tridimensionais [10]).

A ferramenta proposta nesse trabalho tem o diferencial de prover funcionalidades que permitem integrar modelos unidimensionais e tridimensionais em um mesmo ambiente de trabalho, denominado *HeMoLab*. Neste ambiente são integradas ferramentas que permitem a edição e criação de modelos unidimensionais, processamento de imagens médicas para extração de geometrias de artérias, geração de

modelos de elementos finitos tridimensionais, e acoplamento entre modelos de diferentes dimensões, as quais utilizam o *pipeline* de processamento fornecido pelo *ParaView*. Neste trabalho, será abordado o módulo deste sistema computacional responsável por manipular malhas de elementos finitos tridimensionais, utilizada para gerar modelos tridimensionais de diferentes trechos do sistema arterial.

3. Arquitetura

O *ParaView*, é um ambiente para visualização de dados científicos, que dispõe de uma GUI (*Graphical User Interface*) intuitiva e que pode receber parâmetros fornecidos pelos usuários. Sua estrutura é baseada na biblioteca VTK [8], que permite a leitura, conversão e escrita de arquivos em diferentes formatos, permite também a criação de filtros para o processamento de dados e fornece mecanismos para renderização.

O VTK é uma das mais utilizadas bibliotecas destinadas à visualização científica e possui funcionalidades como geração de modelos, processamento de dados e visualização de resultados. Um fluxo de execução VTK é constituído das três etapas a seguir:

- Geração e leitura das estruturas de dados, geradas por simulação.
- Aplicação de filtros, utilizando o *pipeline* do VTK.
- Geração de visualizações para estruturas 2D e 3D. Após o processamento de qualquer dos filtros existentes no modelo 3D, é então gerada uma representação visual dos resultados.

No módulo 3D a ser descrito neste artigo, arquivos e estruturas de dados contendo informações sobre as superfícies e malhas de volume são gerados e processados. Os filtros deste módulo fazem uso da estrutura de *pipeline* do VTK, que permite mudar as entradas ou parâmetros dos filtros propagando no *pipeline* as modificações feitas.

O *ParaView* possui várias características que o fizeram a melhor escolha como ambiente de trabalho:

- *Extensibilidade*: permite de forma simples estender suas funcionalidades e ou adicionar novas rotinas.
- *Paralelismo*: o ambiente permite a execução de seus processos em máquinas diferentes, permitindo também a troca de informação entre os mesmos. O módulo 3D faz uso destas funcionalidades, podendo dividir a carga de processamento entre mais de uma máquina ou processo.
- *Versatilidade*: fornece suporte para uma grande variedade de dados, permitindo ao mesmo tempo a inclusão de novos tipos.

O *ParaView* baseia-se em uma arquitetura *Cliente-Servidor* (Figura 1) que permite a separação do processamento de dados, a visualização e a GUI, de forma que estes processos podem estar interagindo mesmo estando em diferentes processos ou computadores conectados através de uma rede. A grande vantagem desta característica é um melhor escalonamento dos recursos computacionais disponíveis, podendo-se utilizar máquinas com maior capacidade de processamento para tarefas que consomem mais recursos, como por exemplo, o resultado de uma simulação envolvendo geometrias e grandes volumes de dados. Para visualização podem-se utilizar máquinas com menor poder de processamento, mas que possuam *hardware* que permita utilizar renderização

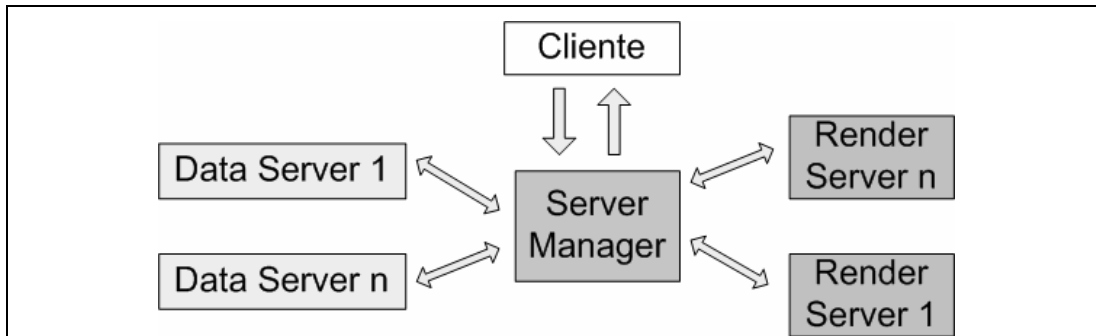


Figura 1. Arquitetura Cliente – Servidor do *ParaView*.

3D, enquanto que a GUI pode ser apresentada em um computador que possua uma arquitetura mais simples.

A arquitetura do *ParaView* é composta pelos seguintes componentes:

- *Client* – Componente que possui interface de usuário, permitindo assim sua interação com o ambiente.
- *Server Manager* – Componente que permite a comunicação entre a interface do usuário, e as classes que residem no lado servidor da aplicação.
- *Render Server* – Componente que tem por finalidade realizar o processo de renderização tridimensional dos dados.
- *Data Server* – Componentes que armazenam e processam volumes de dados através de filtros.

Os componentes citados anteriormente comunicam-se entre si através de uma linguagem Cliente-Servidor: Esta linguagem é utilizada para comunicar os diferentes processos. Permite executar métodos em objetos que se encontram em diferentes processos, em diferentes computadores ou até diferentes sistemas operacionais.

A relação entre os quatro componentes da arquitetura do *ParaView* pode ser vista na Figura 1. As subseções a seguir descrevem cada um dos componentes e como são utilizados.

3.1 *Client*

O componente *Client* está diretamente relacionado com a GUI, pois é responsável por enviar a informação fornecida pelo usuário ao *Server Manager*. O usuário fornece estes dados por meio de controles de interface, como por exemplo, caixas de texto, botões, etc.

Os controles de interface para leitura, escrita e filtragem são especificados utilizando arquivos XML. O *ParaView* utiliza dois tipos de arquivo XML:

- XML Cliente: especifica os tipos de controles (botões, caixas de texto, menus) que a GUI deve possuir para controlar os módulos de escrita, leitura e filtragem que serão utilizados pelo usuário para edição de parâmetros.
- XML Servidor: tem por objetivo relacionar classes do *Server Manager* (*proxies*) com os correspondentes leitores, escritores e filtros existentes no lado servidor (objetos VTK). Utilizando os *proxies* existentes no *Server Manager*, as classes de interface podem então estabelecer comunicação com as classes existentes no lado servidor.

Para cada controle de interface que se deseje agregar à GUI de um determinado módulo, deve existir uma *tag* (campo específico de um arquivo XML) correspondente no XML Cliente. Abaixo é apresentado um trecho do arquivo XML Cliente do modelo 3D.

```
<Module name = "Trisurf"
  menu_name   = "HM Trisurf Filter"
  root_name   = "Trisurf"
  module_type = "Filter"
  button_visibility = "1"
  long_help   = "Apply Several functions to triangular meshes.">
  <Filter class = "vtkHM3DTrisurfFilter">
    <Input name = "Input" class = "vtkMeshData"/>
  </Filter>
  <InputMenu trace_name = "Input" label = "Input" property = "Input"
    Help = "Set the input to this filter." input_name = "Input"/>

  <HM3DTrisurfFilter trace_name = "HM3DTrisurfFilter"/>

  <Documentation>
    Apply Several functions to triangular meshes
  </Documentation>
</Module>
```

3.2 Server Manager

O *Server Manager* é o componente responsável por gerenciar a comunicação entre cliente e os componentes que residem no lado servidor. Outra função do *Server Manager* é manter uma cópia do estado das propriedades dos objetos VTK localizados nos servidores. Com isso é garantida a consistência da informação que é trafegada na comunicação dos diferentes processos. No *Server Manager*, existem referências a *proxies* (objetos locais) que se comunicam com os objetos VTK (instanciados remotamente) nos servidores.

Por sua vez, o *Server Manager* possui os seguintes componentes:

- *Proxies*: *vtkSMProxy* e suas sub-classes, fornecem uma interface de comunicação com os objetos VTK que residem nos servidores. As principais funções de um *proxy* são criar objetos VTK nos servidores e manter referências a estes objetos usando para os chamados *Ids Cliente-Servidor (CSID)*.
- *Properties*: *vtkSMProperty* e suas sub-classes permitem ao cliente acessar a interface referente aos objetos que residem no lado servidor da aplicação.
- *Domínios*: *vtkSMDomain* e suas sub-classes, consistem em conjuntos de valores que uma propriedade pode assumir.

Existem diferentes *Proxies* que precisam ser gerenciados. O responsável por este papel é o *Proxy Manager (vtkSMProxyManager)*, que consiste em uma classe *Singleton* [9], i.e, onde existe uma única referência dela em memória. O *Proxy Manager* mantém referência aos *proxies* e suas respectivas *properties* de acordo com o especificado no XML do servidor. Cada *proxy* está relacionado a um objeto VTK no servidor por meio do *CSID*. O *CSID* é gerado pelo *Process Module (vtkProcessModule, classe Singleton)*,

que tem as funções de atribuir um *CSID* único para cada objeto VTK criado através dele e de obter a referência ao objeto dado um *CSID*.

A configuração do *Server Manager* é armazenada em arquivos XML, onde se especifica com que objeto VTK no servidor cada *proxy* irá se comunicar e quais são as propriedades associadas a esta comunicação. Enquanto os arquivos XML do cliente especificam que controles de interface devem aparecer para permitir que o usuário realize a edição de dados e parâmetros de filtros e módulos, o XML do servidor determina qual *proxy* será responsável por tratar/enviar cada um destes dados aos objetos no servidor e quais serão as propriedades associadas.

Nos arquivos XML do servidor existe uma *tag* principal chamada *ServerManagerConfiguration* que determina uma configuração para o *ServerManager*. Esta *tag*, possui uma “*sub-tag*” chamada *ProxyGroup*, que tem a função de agrupar *proxies* de acordo com sua funcionalidade (leitores, escritores, filtros, etc). Um breve exemplo de XML do servidor referente a um dos filtros que compõem o Módulo 3D chamado *HM3DTrisurfFilter* será apresentado a seguir.

Um exemplo de chamada de um *proxy* via XML é o caso dos filtros (todos os tipos de filtros são considerados fontes de dados ou “*sources*”). Cada *source* é representado no arquivo XML pela *tag SourceProxy* que possui o atributo *name*, que é um identificador único deste *SourceProxy*. Dentro desta *tag* encontram-se as definições de *properties* que estão associadas a este *proxy*.

As *properties* são usadas para representar a chamada de um método (e qualquer argumento a ele associado) de um objeto VTK armazenado no *Client* ou em qualquer um dos servidores. No trecho de código abaixo é apresentado também um domínio, o qual tem a finalidade de especificar a faixa de valores que a propriedade pode assumir.

```
<ServerManagerConfiguration>
  <ProxyGroup name = "filters">
    <SourceProxy name = "Trisurf" class = "vtkHM3DTrisurfFilter">
      <InputProperty
        Name = "Input"
        Command = "SetInput">
        <ProxyGroupDomain name = "groups">
          <Group name = "sources"/>
          <Group name = "filters"/>
        </ProxyGroupDomain>
        <DataTypeDomain name = "input_type">
          <DataType value = "vtkMeshData"/>
        </DataTypeDomain>
      </InputProperty>
    </SourceProxy>
  </ProxyGroup>
</ServerManagerConfiguration>
```

3.3 Data Server

O *Data Server* é responsável por criar ou realizar a leitura de dados e aplicar filtros sobre os mesmos. Este componente é composto por leitores, *sources* e filtros VTK. É no *Data Server* que são instanciados os filtros do módulo 3D.

3.4 *Render Server*

Este componente é responsável pelo processamento necessário para se exibir e renderizar conjuntos de dados em 3D. Sempre que um conjunto de dados é lido ou gerado, neste componente é aplicado um filtro, que tem por finalidade gerar a sua representação poligonal. Posteriormente é criada uma série de *mappers*, propriedades, e tabelas de cores, que nos permite visualizar a representação poligonal destes dados. Após o processamento dos filtros do módulo 3D, a visualização final dos resultados é feita no *Render Server*.

3.5 *Linguagem Cliente-Servidor*

A linguagem Cliente-Servidor do VTK, é uma linguagem independente de plataforma que, através de troca de mensagens entre os componentes que integram o *ParaView*, permite a criação, manipulação e destruição de objetos VTK. Mensagens são enviadas pelo componente cliente ao servidor onde são interpretadas e executadas.

A seguir, será descrito o módulo de processamento de malhas de elementos tridimensionais desta ferramenta.

4. Módulo 3D

As ferramentas e funcionalidades descritas neste trabalho fazem parte de um sistema computacional que permite a geração e análise de modelos do sistema cardiovascular humano chamado *HeMoLab*. Este sistema computacional é composto de vários módulos: módulo 1D (geração de árvores arteriais 1D), módulo de processamento de imagens, módulo 3D e módulo de acoplamento. Apesar de todos os módulos trabalharem de forma integrada, o intuito deste trabalho é explorar apenas as funcionalidades oferecidas pelo módulo 3D.

O objetivo do módulo 3D é prover funcionalidades que permitam, a partir de uma geometria inicial de baixa qualidade da artéria ou região de interesse, gerar uma malha de elementos finitos de boa qualidade e adequadamente refinada. Uma vez que todas as informações foram introduzidas pelo usuário, estas são salvas em arquivos que serão posteriormente processadas pelo *SolverGP* [5]. O *SolverGP* é um software para resolução numérica das equações de Navier-Stokes aplicadas a modelos acoplados 3D - 1D deformáveis, com discretização espacial das equações via Método dos Elementos Finitos.

No módulo 3D foram reunidos diferentes filtros para processamento de malhas de superfície que permitem processar a triangulação de uma geometria para melhorar a sua qualidade [4]. Assim, uma vez que a região de interesse foi isolada (segmentada) em uma imagem médica, a superfície (triangulação) que caracteriza aquela região é reconstruída.

A primeira triangulação da geometria gerada depois da segmentação de um volume de imagens costuma ser de má qualidade, apresentando triângulos muito pequenos ou em forma de “agulha”, o que aumenta consideravelmente o erro dos métodos de discretização como Elementos Finitos, requerendo-se um processamento adequado. Para gerar as malhas de elementos finitos, foram incluídos diferentes métodos clássicos de geração e otimização de malhas não estruturadas (triangulares e

tetraédricas). Estes algoritmos encontram-se implementados em um software chamado *Gemesis3D* [2]. As funcionalidades oferecidas por este software foram adicionadas e disponibilizadas no Módulo 3D, integradas às funcionalidades já existentes no *ParaView*. Depois que a superfície foi adequadamente processada, é gerada uma malha de volume (utilizando o algoritmo de *Delaunay* [3]). Posteriormente esta malha de volume (composta de tetraedros) é otimizada [4]. Obtidas as malhas de elementos finitos, pode ser de interesse do usuário realizar a simulação computacional somente do modelo 3D (impondo condições de contorno, é possível realizar a simulação do modelo 3D isolado). Isto pode ser útil em diferentes circunstâncias, e.g., geração das condições iniciais do modelo acoplado. Neste caso, devem ser fornecidas as condições no contorno nas entradas/saídas da geometria.

Um dos motivos da escolha do *ParaView* como base para o desenvolvimento deste ambiente foi a sua arquitetura de processamento de dados na forma de um *pipeline* de processamento. Assim, pode-se encadear diversos filtros e modificá-los de forma interativa. Quando uma alteração é introduzida na entrada ou nos parâmetros de um determinado filtro, este é re-executado nos dados de entrada, propagando estas modificações nos filtros que tem como entrada a saída do filtro re-executado, o que facilita o tratamento dos dados. Os filtros que compõem esse módulo foram originalmente implementados na ferramenta *Gemesis3D*. Contudo, nessa ferramenta, os filtros eram aplicados de forma incremental sempre operando no mesmo conjunto de dados, destruindo assim os dados de entrada e sem a possibilidade de mudar os parâmetros dos filtros previamente aplicados. Essa limitação foi eliminada neste módulo através da implementação de um adaptador que traduz o formato de dados utilizado pelo *ParaView* (o formato de dados padrão do VTK) para o formato utilizado pelo *Gemesis3D* e vice versa. Tanto a entrada como a saída de cada instância do filtro aplicado neste módulo foi traduzida de maneira a permitir o uso dos recursos providos pelo *ParaView*. Como resultado, todas as etapas e filtros que compõem o módulo 3D, podem ser utilizadas de maneira integrada com resultados dos filtros de outros módulos (e.g., módulo de processamento de imagens, módulo 1D, módulo de acoplamento) de forma interativa e intuitiva, facilitando assim o uso da ferramenta e provendo uma maior versatilidade.

5. Funcionalidades do Módulo 3D

O funcionamento da ferramenta será demonstrado através de um estudo de caso (Figura 2). A idéia é ilustrar as etapas que foram necessárias para processar a geometria de uma artéria obtida a partir de imagens médicas de um paciente real. Esta malha, inicialmente de má qualidade, foi processada para se obter uma malha de elementos finitos devidamente refinada de boa qualidade.

O *HeMolab* também fornece funcionalidades para criar modelos acoplados 1D-3D [10][11][12]. Os modelos 1D são utilizados para representar as maiores artérias do corpo humano. Estes modelos são uma simplificação das equações de Navier-Stokes 3D, sendo obtidos ao assumir-se uma série de hipóteses no fluxo: 1) os vasos são retos, 2) o perfil de velocidade em qualquer ponto é conhecido e 3) uma relação constitutiva entre a pressão e a área do vaso. Desta forma, os modelos acoplados têm como característica a possibilidade de associar partes do modelo 3D (denominadas **grupos de**

elementos) aos segmentos unidimensionais simplificados do modelo 1D, estando essa etapa contida no Módulo de Acoplamento. Para que essa associação seja realizada, os grupos da geometria 3D devem ser selecionados e posteriormente associados aos terminais do módulo 1D (Figura 3). O recurso de seleção disponível no módulo 3D, utilizado no Módulo de Acoplamento, permite ao usuário escolher um grupo e o mesmo fica realçado na janela de renderização.

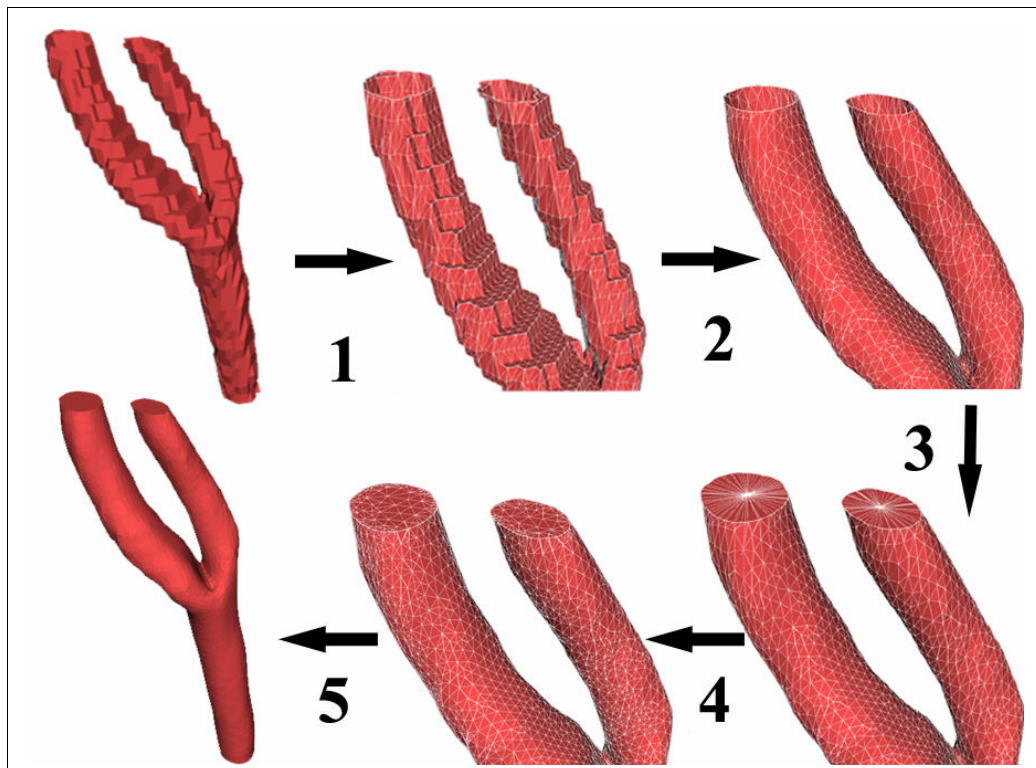


Figura 2. Exemplo de uso dos filtros de processamento de malha do Módulo 3D. (1) Visualização da malha da superfície a ser processada; (2) Aplicação do filtro de suavização (*smooth*); (3) Filtro para inserir novos grupos (tampas); (4) Filtro para inserir novos nós à malha (refinamento) e (5) visualização do resultado final.

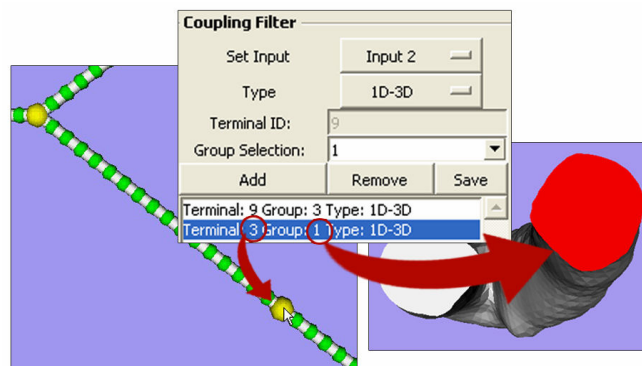


Figura 3. Exemplo de geometria 3D com um dos grupos selecionado e a posterior seleção de um terminal em um modelo unidimensional.

6. Conclusões

O uso de ferramentas de predição por profissionais de medicina e biologia tem grande importância na atualidade. A grande quantidade de problemas relacionados ao SCVH foi a motivação para a criação do *HeMoLab*, uma ferramenta que permite a criação de modelos personalizados do sistema cardiovascular. Esta ferramenta foi desenvolvida sobre o *ParaView*, um software multiplataforma de visualização científica que utiliza computação paralela para processar grandes volumes de informação.

O presente artigo descreve as características do Módulo 3D, que tem por objetivo prover funcionalidades que permitam processar geometrias tridimensionais para a geração de malhas de elementos finitos adequadas e de boa qualidade. Os dados provenientes deste módulo são posteriormente salvos em arquivos e processados pelo software de simulação numérica, que permite tratar problemas não-lineares transientes que compreendem modelos de diferentes dimensões. Acredita-se que a utilização desta ferramenta no ambiente médico poderá fornecer informações mais precisas e valiosas na realização de diagnósticos e no planejamento cirúrgico.

Atualmente, no âmbito do Módulo 3D, estão sendo desenvolvidas novas técnicas de processamento de malhas tridimensionais para identificar, tratar e visualizar triângulos de baixa qualidade. Em relação aos trabalhos futuros, pretende-se incorporar ao Módulo 3D do *HeMoLab* a possibilidade de adição de *stents* e *bypass* para auxiliar na predição de resultados de cirurgias que façam uso destes recursos.

7. Referências

- [1] Kitware Inc. ParaView. <http://www.paraview.org>.
- [2] Vénere, M.J. Técnicas adaptativas para el Método de Elementos Finitos en dos y tres dimensiones. PhD thesis, Instituto Balseiro, Universidad Nacional de Cuyo, Novembro, 1996.
- [3] Baker, T. Automatic mesh generation for complex three dimensional regions using a constrained delaunay triangulation. *Engineering Computations*, 5:161–175, 1989.
- [4] Vénere, M.J.. Optimización de la calidad de mallas de elementos finitos mediante cambios localizados de topologia. *Rev. Int. de Mét. Num. para Cálculo y Diseño en Ing.*, 13:3–13, 1997.
- [5] Urquiza, S.A. and Vénere, M.J. An application framework architecture for fem and other related solvers. In S.R.Idelsohn, V.E.Sonzogni, and A.Cardona, *Mecánica Computacional*, volume 22, pages 3099–3109, Argentina, Outubro, 2002.
- [6] Jones, D. et al: The GEMSS Grid: An Evolving HPC Environment for Medical Applications. In *Proceedings of HealthGrid 2004*.
- [7] SimTK – The Simulation Toolkit. <https://simtk.org/xml/cardiovascular.xml>
- [8] VTK – The Visualization Toolkit. <http://www.vtk.org>.
- [9] Gamma, E., Helm, R., Johnson, R.. *Design Patterns. Elements Of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. Addison-Wesley, Gam E 95:11.Ex, 1995
- [10] Formaggia, L. et al. Numerical treatment of defective boundary conditions for the navier-stokes equations. *SIAM J. Numer. Anal.*, 40(1):376-401, 2002.
- [11] Lamponi, D.N. One dimensional and multiscale models for blood flow circulation. Tese de Doutorado, École Polytechnique Fédérale de Lausanne, 2004.
- [12] Urquiza, S. A., Blanco, P. J., Vénere, M. J. e Feijóo, R. A. Multidimensional modelling for carotid artery blood flow. *Computer Methods in Applied Mechanics and Engineering*, 195(33-36):4002-4017, Julho, 2006.
- [13] Larrabide, I., Feijo, R. A, *HeMoLab: Laboratório de Modelagem em Hemodinâmica; Relatório Técnico 13/2006*, LNCC, Brasil, 2006.