



SuLi - Simulador de Escoamentos com Superfície Livre

Manual do Usuário

Leonardo Romero Monteiro

Versão 2.0
Setembro de 2024



Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas
Departamento de Engenharia Civil
Laboratório de Ciências das Águas



&



Universidade Federal do Rio Grande do Sul
Instituto de Pesquisas Hidráulicas
Núcleo de Estudos em Transição e Turbulência



Monteiro, Leonardo Romero

Manual do Usuário: SuLi - Simulador de Escoamentos com Superfície Livre - Joinville/SC:
UDESC/CCT/DEC/LaCiA, 2024.

17 Páginas

Palavras-chave:

Equações de Navier-Stokes, Escoamento monofásico, Escoamento bifásico, Simulador Numérico.

Colaboradores:

Bruna Fernanda Soares (2022-2024)

Karol Rocha Araújo (2023-2024)

Pedro Miguel de Andrade Filho (2022-2023)

Vinícios Morigi (2021-2022)

Versão 2.0

Edição brasileira

Setembro de 2024

Conteúdo

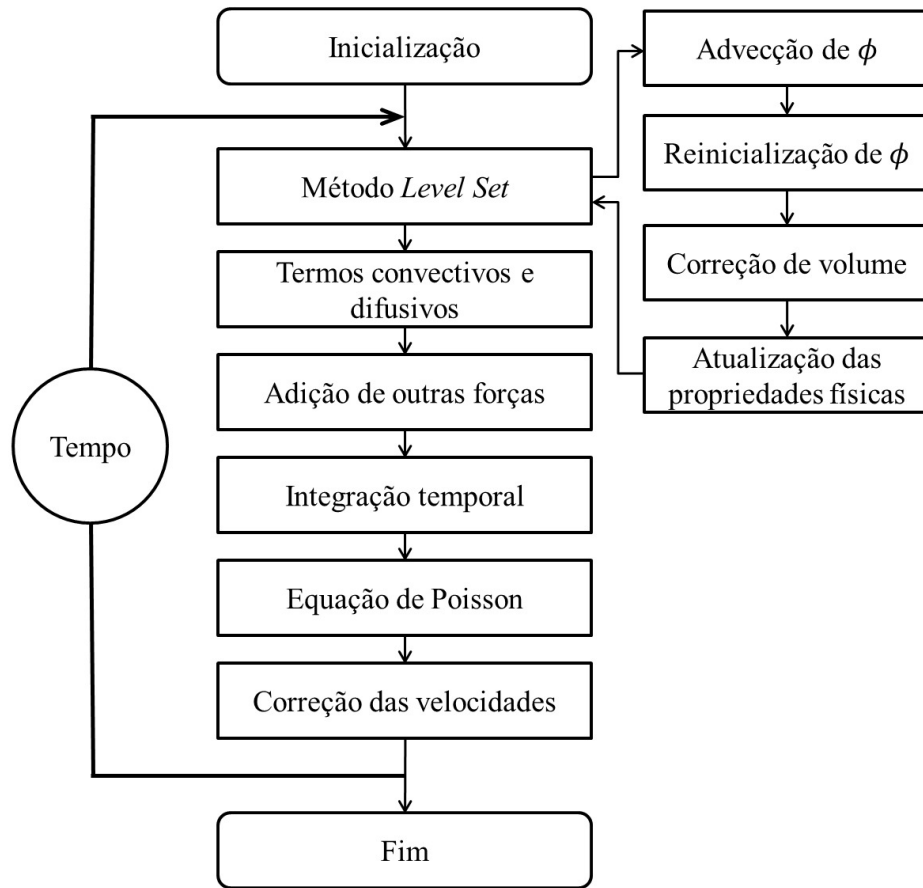
1	Introdução	1
1.1	Objetivo	1
1.2	Considerações adicionais	2
2	O SuLi	3
2.1	Pré-requisitos	3
2.2	Instalação	3
2.2.1	Usuários de Windows	3
2.2.2	Para usuários Linux e Windows	3
2.3	Estrutura	4
2.4	Compilando	4
2.5	Rodando	4
3	Personalizando o código	5
3.1	Definindo o domínio	5
3.2	Condições e configurações iniciais	6
3.2.1	Evolução temporal ²	6
3.2.2	Termo advectivo	6
3.2.3	Termo convectivo	7
3.2.4	Obstáculos	7
3.2.5	Modelo de turbulência	8
3.2.6	Ondas de entrada	9
3.2.7	Plotagem ^{IC}	9
3.2.8	Método da Solução Manufaturada	10
3.2.9	Coeficientes Específicos	10
3.2.10	Camada Esponja	10
3.2.11	Pressão Hidrostática	10
3.2.12	Level-Set ^{IC}	11
3.2.13	Restart ^{IC}	12
3.3	Condições de contorno	12
4	Visualização dos resultados	14
A	Apêndice: Ambiente Ubuntu no Windows	15
	Bibliografia	16

1 Introdução

O programa **SuLi (Superfície Livre)** foi criado em 2014 pelo Núcleo de Estudos em Transição e Turbulência (NETT) pertencente ao Instituto de Pesquisas Hidráulicas da Universidade Federal do Rio Grande do Sul (IPH/UFRGS). Atualmente o programa é também desenvolvido pelo Laboratório de Ciência das Águas (LaCiA) no Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina (CCT/UDESC).

O código possui dois módulos para representação da superfície livre numericamente. Um denominado **SuLi-IT** que utiliza métodos de rastreamento de interface (*interface-tracking method*) o qual simula o fluido de forma monofásica (Monteiro and Schettini (2014)) e outro **SuLi-IC** que utiliza o método de captura de interface (*interface-capturing method*), baseado na metodologia Level-Set (Monteiro, Lucchese, and Schettini (2019)) o qual considera uma composição bifásica do escoamento. Também é possível considerar um contorno sólido variável aplicando a estratégia do Método de Fronteiras Imersas (*Immersed Boundary Method - IBM*) por forçagem direta (*discrete forcing*). As etapas de cálculo de desenvolvimento do modelo matemático são apresentadas através do fluxograma na Figura 1.

Figura 1: Fluxograma de desenvolvimento do modelo matemático do código SuLi.



O código é voltado para o desenvolvimento acadêmico e quando aplicado em casos reais de engenharia, o usuário deve conhecer especificamente quais métodos são os mais adequados para cada uso e aplicação. Os desenvolvedores do código não se responsabilizam perante os resultados gerados.

1.1 Objetivo

O **SuLi** tem como objetivo ser um código aberto e livre que simula numericamente corpos de água e escoamentos com superfície livre utilizando as equações de Navier-Stokes, fornecendo

resultados capazes de suprir análises importantes destes escoamentos. O usuário tem a liberdade de definir as condições iniciais e de contorno desejadas, além de modificar o código como assim desejar para atingir os seus objetivos.

1.2 Considerações adicionais

O código é escrito na linguagem de programação Fortran 90 desenvolvido na plataforma Linux, utilizando o programa Gfortran (ao usuário do Windows é possível utilizar o código após algumas configurações também abordadas neste manual, vide seção 2.2.1).

Para o pós-processamento dos resultados é recomendado o programa livre **Paraview** (abordado na seção 4), entretanto o usuário tem a liberdade de utilizar qualquer outro programa de seu interesse, bastando adaptar o formato dos dados de saída do **SuLi** de acordo com o programa escolhido.

Ao utilizar o módulo **SuLi-IT** favor referenciar a publicação [Monteiro and Schettini \(2014\)](#) e caso utilize **SuLi-IC** favor referenciar a publicação [Monteiro et al. \(2019\)](#). O **SuLi** é um código livre e aberto e a única forma de fortalecer o mesmo é ampliando a sua utilização.

2 O SuLi

Para poder utilizar o **SuLi** da melhor forma, é necessário que alguns pré-requisitos de funcionamento sejam instalados e configurados em sua máquina, em contra partida não é essencial o conhecimento avançado do código para o seu uso mais básico, basta um entendimento geral conforme aborda a seção 3.

2.1 Pré-requisitos

Os requisitos necessários para a melhor utilização do **SuLi** solicitam um computador com sistema operacional 64-bit. A configuração do processador e da memória RAM dependerão dos casos que se deseja simular, mas, de qualquer forma, são indicados computadores com no mínimo: 4 GB de RAM, e; processador i5 ou equivalente. Aconselha-se também se ter ao menos 5 GB livres no HD antes de iniciar qualquer simulação. Quanto melhor for o computador mais rápidas serão as simulações e mais pontos podem ser utilizados no domínio computacional.

Observação: Os pré-requisitos são apenas necessários para usuários iniciantes. Usuários avançados podem adaptar as configurações de suas máquinas de acordo com suas necessidades.

2.2 Instalação

2.2.1 Usuários de Windows

Para usuários do Windows, há uma etapa de instalação inicial extra, que envolve a configuração do ambiente Ubuntu. É necessário que siga os passos contidos no Apêndice A e depois retorne a este tópico, prosseguindo a instalação naturalmente idealizada para usuários Linux.

2.2.2 Para usuários Linux e Windows

Para usuários Linux (ou Windows que já cumpriram o passo anterior), é necessário garantir que o sistema operacional está atualizado a fim de evitar erros de compatibilidade. Para tal, siga os passos seguintes no terminal de comando do Linux:

```
sudo apt-get update (Para compreender os programas e arquivos desatualizados)
```

```
sudo apt-get upgrade (Para efetivamente atualizar o sistema)
```

Proseguindo, é necessário instalar os programas **gfortran** e **MakeFile** em sua máquina, para que o código do **SuLi** possa ser corretamente interpretado e compilado. Portanto, ainda no terminal, siga estes comandos para realizar a instalação:

```
sudo apt-get install gfortran (Para instalar o gfortran)
```

```
sudo apt-get install make (Para instalar o MakeFile)
```

Agora está tudo instalado e configurado para a utilização do **SuLi**. A instalação do visualizador de dados Paraview é apresentada na seção 4. Caso a instalação de algum desses pacotes não seja efetuada com sucesso, tente refazer as etapas de configuração, instalação e atualização do sistema. Caso o problema persista entre em contato pelo e-mail: leonardo.monteiro@udesc.br

2.3 Estrutura

Para a utilização do **SuLi** existem 3 etapas:

- Compilação e inicialização do programa:
 - Arquivo gerador/compilador inicial “Makefile”.
- Cálculo da simulação numérica:
 - Algoritmos em Fortran 90 programados para efetivamente realizar a simulação. As condições iniciais e de contorno estão alocadas nesta etapa.
- Armazenamento dos dados obtidos:
 - Pastas “dados” e “arquivos” que armazenam os dados para etapas de pós-processamento e análises futuras.

Na seção 3 a apresentação do código é feita de forma detalhada, fornecendo ferramentas para que o usuário compreenda a liberdade e tenha autonomia de personalizar o programa de acordo com a sua necessidade.

2.4 Compilando

A partir do terminal de comando do Linux, navegue até o local onde está a pasta do **SuLi**. Ciente dos arquivos presentes na pasta, o usuário pode realizar algumas operações no terminal de fomar individual com o **Makefile**, que são:

```
make          (para compilar o código)*;
make clean    (para remover os arquivos auxiliares criados na compilação);
make allclean (para remover todos os arquivos não básicos para o código).
```

* O comando **make** fornece liberdade ao usuário para definir itens específicos relevantes à compilação dentro do arquivo Makefile. Ativando itens na linha “OPTFC =” mudará importantes configurações de compilação, por exemplo:

- **-mcmmodel=medium** → possibilitará o uso de mais pontos para a simulação. Utilizar se estiver ocorrendo erro por causa do excesso de pontos.
- **-fcheck=all** → faz uma verificação do código para identificar se está escrito corretamente. Utilizar quando se modificar bruscamente o código para identificar possíveis erros de escrita.

Usuários que não tenham conhecimento do uso do Makefile devem utilizar a opção padrão definida.

Toda vez que o usuário modificar algo na escrita das rotinas no código (nos arquivos .f90) deve ser feita novamente a compilação com o comando **make**. Contudo, se a mudança for em um módulo, deve-se primeiro remover os arquivos auxiliares (**make clean**) e depois realizar a compilação (**make**).

2.5 Rodando

Depois de compilado, para rodar o **SuLi** basta digitar o comando **./rodar_3d** no terminal (note que tanto a pasta na qual o programa foi instalado quanto as pastas “dados” e “arquivos” vão receber novos arquivos, auxiliares ao processo da simulação). Para interromper a simulação, feche o terminal de comando ou aperte o comando **Ctrl+c**.

3 Personalizando o código

Nesta seção são definidos: domínio de cálculo, funções a serem utilizadas para o cálculo das equações, condições iniciais e condições de contorno. Incluem-se as principais modificações e personalizações que o usuário pode fazer de acordo com a sua necessidade.

Mesmo que o **SuLi-IT** e o **SuLi-IC** sejam semelhantes, existem funções/variáveis particulares de cada código. Quando o item for específico do **SuLi-IT** a função/variável terá sobrescrito ^{IT}, caso for específico do **SuLi-IC** terá sobrescrito ^{IC}.

O **SuLi** é composto de códigos reunidos em documentos de subrotina, dos quais se complementam e se interdependem. Abaixo são as pastas que compõem o **SuLi-IC**.

0_modules.f90	1_principal.f90	2_iniciais.f90	3_contorno.f90
4_plot.f90	5_viscosidade.f90	6_convdiff.f90	7_advectivo.f90
8_derivadas.f90	9_grad.f90	10_pos.f90	11_diversos.f90
12_paraview_3D.f90	13_mms.f90	14_level_set.f90 ²	15_ibm.f90

Atenção! A atribuição de parâmetros para uma simulação deve ser feita no documento 00_suli3d.prm. Feito isso, o parâmetro será automaticamente aplicado em sua respectiva pasta de subrotina.

Observação: Sempre que uma nova simulação for realizada é importante limpar todos os dados existentes referentes a simulações anteriores para que estes não influenciem no andamento e resultado da nova simulação. Para tanto, utilize os comandos apresentados em 2.4 de acordo com a necessidade.

3.1 Definindo o domínio

Na pasta 00_suli3d.prm são encontradas as variáveis para especificar as dimensões espaciais pelo o número de células do domínio e discretização do intervalo espacial, (nx, ny, nz) e (dx, dy, dz), respectivamente. As dimensões do domínio são referenciadas por:

Eixo x	→	$(nx + 1) \times dx$	→	Comprimento (m)
Eixo y	→	$(ny + 1) \times dy$	→	Largura (m)
Eixo z	→	$(nz + 1) \times dz$	→	Profundidade (m)

A malha temporal da simulação é definida pelas seguintes variáveis:

dt0	→	Discretização temporal Δt (s)
dt_frame	→	Número de tempo por arquivo plotado (s)
t_s	→	Tempo a ser simulado (s)

Observação: O domínio espacial do **SuLi** foi feito para realizar simulações em três dimensões (3D) e não em apenas duas (2D). Para tanto, o usuário que quiser minimizar os efeitos de uma terceira dimensão em sua simulação deverá possuir um bom entendimento das equações e leis que regem tais processos, para que os resultados obtidos estejam condizentes com a realidade física e matemática proposta à simulação.

3.2 Condições e configurações iniciais

Ainda no `00_suli3d.prm`, a primeira condição inicial importante em relação ao escoamento é o parâmetro `uinicial`; que significa a velocidade inicial que o escoamento terá a partir do seu eixo x , isto é, ao longo do comprimento do seu escoamento (como denotado em 3.1). Também existem algumas outras configurações iniciais importantes para o desenvolvimento da simulação, neste e em outros arquivos/módulos/subrotinas, tratados a seguir:

3.2.1 Evolução temporal²

Diferentes métodos para calcular a evolução temporal da solução estão disponíveis no **SuLi-IC**. Os métodos podem resultar em um grau menor de precisão e com maior rapidez na resolução, ou serem mais precisos, porém consequentemente mais lentos. Esta função está disponível pelo parâmetro representado por `t_tempo`, acessível em `00_suli3d.prm` e aplicado em `11_diversos.f90`.

```
t_tempo
0 → Euler Explícito
1 → Runge-Kutta de 2ª Ordem
2 → Runge-Kutta de 3ª Ordem
3 → Adams-Bashforth de 2ª Ordem
```

Para casos com tipo de tempo variável, pode ser definido em `t_tempo_var`.

```
t_tempo_var
0 → Δt constante
1 → Δt adaptativo
```

Euler Explícito de 1ª Ordem (EE): Mais simples, mais rápido, menos preciso.

Runge-Kutta de 2ª Ordem (RK2): Mais complexo, mais lento, mais preciso.

Runge-Kutta de 3ª Ordem (RK3): Ainda mais complexo, ainda mais lento, ainda mais preciso.

Adams-Bashforth de 2ª Ordem (AB2): Menor custo computacional se comparado ao método de RK2, em contrapartida possui restrições quanto a estabilidade do código.

3.2.2 Termo advectivo

É possível obter formas matemáticas distintas, porém equivalentes, para o cálculo dos termos advectivos da equação de Navier-Stokes. Mesmo sendo equivalentes matematicamente, podem apresentar diferenças numéricas relevantes. Para modificar a forma de cálculo do termo advectivo utilizamos o parâmetro representado por `adv_type` aplicado no módulo `7_advectivo.f90`. Para a modificação deste parâmetro, também deve-se acessar a pasta `00_suli3d.prm`.

```
adv_type
1 → Advectivo Clássico
2 → Rotacional
3 → Antissimétrico
```

Formulação Clássica: $H_i = u_j \frac{\partial u_i}{\partial x_j}$

Formulação Rotacional: $H_i = u_j \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right)$

Formulação Antissimétrica: $H_i = \frac{1}{2} \left(\frac{\partial u_i u_j}{\partial x_j} + u_j \frac{\partial u_i}{\partial x_j} \right)$

3.2.3 Termo convectivo

As derivadas espaciais do termo convectivo são resolvidas por aproximações numéricas das mesmas. Os tipo de derivada que o **SuLi** utiliza para as resoluções dos métodos através de determinados esquemas de interpolação são escolhidos pelo parâmetro **der**, antes determinadas em `00_suli3d.prm`, e aplicadas no arquivo `0_module.f90`.

der	
1	→ Upwind
2	→ Centrado*
3	→ Upwind de 2ª Ordem ^{IC}

Upwind: Proposta para o cálculo de soluções aproximadas de equações de conservação. Possui grande erro de difusão numérica. Também chamado de *Upwind* de 1ª Ordem.

Centrado*: Simples, preciso e rápido mas pode facilmente apresentar problemas de estabilidade numérica.

Upwind de 2ª Ordem^{*,IC}: Mais preciso do que o *Upwind* de 1ª Ordem, entretanto mais custoso. Ainda possui erro expressivo de difusão numérica.

*Só é possível utilizar as opções quando **adv_type** for advectivo clássico.

3.2.4 Obstáculos

É possível adicionar obstáculos ao domínio da simulação do escoamento a partir da metodologia IBM (*Immersed Boundary Method*), dado por **ibm_t**. O IBM é utilizado para resolver situações onde há interação entre fluidos e objetos sólidos, como por exemplo, a simulação de escoamento ao redor de uma estrutura ou a movimentação de partículas dentro de um fluido.

ibm_t	
0	→ Sem IBM e sem obstáculo
1	→ IBM forçado
2	→ IBM induzido — Auguste et al. (2019)

Os obstáculos são representados no código através do parâmetro **obst_t** (definido na pasta `00_suli3d.prm`) e modificados no módulo `3_contorno.f90`. No **SuLi-IT** estão disponíveis 8 obstáculos pré-definidos, enquanto no **SuLi-IC** constam 2, conforme segue:

obst_t ^{IT}	
0	→ Sem obstáculo
1	→ Dunas
2	→ Yue, Lin, and Patel (2003)
3	→ Nasr-Azadani and Meiburg (2014)
4	→ Beji and Battjes (1994)
5	→ Canal tipo 0 — Rijn (1980)
6	→ Canal tipo 1 e 2 — Rijn (1980)
7	→ Monteiro, Lucchese, and Schettini (2017)
8	→ Rigotti et al. (2015)
obst_t ^{IC}	
0	→ Sem obstáculo
1	→ Rugosidade de fundo uniforme — Zampiron et al. (2022)
2	→ Barragem — Koshizuka (1995) e Kleefsman (2005)

Assim como o usuário pode optar por não adicionar obstáculo ao seu domínio (`obst_t = 0`), é possível que em versões futuras do **SuLi** e deste manual sejam disponibilizadas ferramentas e opções para modelagem de obstáculos diretamente pelo próprio usuário.

Sem IBM: O IBM está desativado e não há obstáculos no domínio.

IBM forçado: Utilizado para forçar a velocidade a zero nas células da malha que correspondem à região ocupada pelo corpo sólido dentro do fluido; significa que o movimento do fluido é impedido dentro dessas células, criando um "bloqueio" que simula o efeito de um obstáculo rígido.

IBM induzido: As forças de interação entre o fluido e o obstáculo são calculadas de maneira a induzir o comportamento desejado no fluido ao redor do obstáculo. Esta técnica é baseada no trabalho de [Auguste et al. \(2019\)](#).

3.2.5 Modelo de turbulência

Utilizado para calcular e antecipar os efeitos da turbulência em fluidos, através da resolução numérica das equações de Navier-Stokes. O usuário pode escolher entre a não aplicação de um modelo de turbulência ou entre os diferentes modelos disponíveis quando do uso de uma malha não tão refinada. O uso de cada opção depende do nível de refino da malha utilizada na simulação, sendo a viscosidade total obtida pelo somatório da viscosidade cinemática com a viscosidade cinemática turbulenta. Este parâmetro é representado por `m_turb`, e definido também no `00_suli3d.prm`.

<code>m_turb</code>	
0	→ Sem Modelo
1	→ LES Smagorinsky–Lilly Clássico – Lilly (1967) ; Smagorinsky (1963)
2	→ RANS com energia cinemática turbulenta – Heinz (2020)
3	→ DES – Heinz (2020)

Sem modelo: Quando a malha estiver refinada o suficiente, excluindo a necessidade do uso de um modelo de turbulência ou outros métodos estejam sendo utilizados para representar a turbulência.

LES Smagorinsky-Lilly Clássico: Modelo de simulação de grande escala (*Large Eddy Simulation* - LES) que calcula a viscosidade cinemática turbulenta a partir do coeficiente de Smagorinsky (`csmag`), do tamanho representativo da malha e do tensor de taxa de deformação.

RANS com energia cinemática turbulenta: Modelo de Navier-Stokes por médias de Reynolds (*Reynolds Average Navier Stokes* - RANS) que calcula a viscosidade cinemática turbulenta (ν_t) a partir do parâmetro do modelo (`cmu`), da energia cinética turbulenta e da escala de tempo de dissipação.

***Detached Eddy Simulation* (DES):** Modelo híbrido RANS-LES que calcula a viscosidade cinemática turbulenta a partir dos parâmetros do modelo (`cmu` e `cdes`), da energia cinética turbulenta e da escala de tempo de dissipação.

Tendo definido o modelo de turbulência, é necessário a aplicação de coeficiente a eles, definidos por `cka`, `csmag`, `cmu`, `cdes`.

```
cka    → Constante de Kármán (usualmente 0,41)
csmag  → Coeficiente de Smagorinsky
cmu    → Coeficiente de modelo de turbulência LES/RANS
cdes   → Coeficiente de modelo de turbulência DES (usualmente 0,65)
```

Observação 1: Recomenda-se a calibração dos coeficientes dos modelos de turbulência.

Observação 2: Considerando a possibilidade de aplicação de uma velocidade de entrada (`uinitial`), a energia cinética turbulenta inicial é considerada a partir da intensidade de turbulência inicial (`iturb`) definida no `00_suli3d.prm`.

Observação 3: O comprimento de escala do modelo deve ser corrigido no denominador do calculo da taxa de dissipação do código `5_viscosidade.f90`.

3.2.6 Ondas de entrada

O **SuLi** tem a capacidade de simular a entrada de ondas no domínio espacial do escoamento, através da aplicação numérica da Teoria de Ondas de Stokes. Esta codificação foi baseada no trabalho de [Higuera, Lara, and Losada \(2013\)](#). Os parâmetro a serem representados por `wave_t`, e também definidos na pasta `00_suli3d.prm`, são:

```
wave_t
0  → Sem onda
1  → Onda de Stokes I
2  → Onda de Stokes II
5  → Onda de Stokes V
```

Sem onda: ausência de ondas na simulação.

Onda de Stokes I: ondas de menor amplitude e maior comprimento.

Onda de Stokes II: ondas de amplitude e comprimento intermediários.

Onda de Stokes V: ondas de maior amplitude e menor comprimento.

3.2.7 Plotagem^{IC}

Refere-se ao que será apresentado ao final dos cálculos realizados pelo **SuLi** na simulação, sendo para a visualização/análise do usuário. Parâmetro aqui representado por `t_plot`.

```
t_plot
0  → Modo simples
1  → Modo completo
```

Na seção 4 é possível entender melhor o que e como serão visualizados os itens da simulação.

Modo Simples: Consideram-se os parâmetros **Velocidade** do escoamento; **Level-Set**, ferramenta utilizada pelo **SuLi** para diferenciar os fluidos do domínio, em geral água e ar; e **IBM** (*Immersed Boundary Method*), uma técnica utilizada para os cálculos em simulações de interação fluido-estrutura.

Modo Completo: Além dos anteriores, também são levados em consideração os parâmetros Pressão, Vorticidade e Viscosidade do escoamento.

3.2.8 Método da Solução Manufaturada

O Método da Solução Manufaturada (*Method of Manufactured Solutions* - MMS) é utilizado para a verificação das aplicações numéricas, atribuído apenas aos desenvolvedores de esquemas numéricos a fim de viabilizar o seu uso no código. Este parâmetro é representado por `mms_t` e se define em `00_suli3d.prm`. A presente solução manufaturada é descrita com detalhes por Wang, Roache, Jia, Schmalz, and Smith (2009).

```
mms_t
0 → Sem MMS
1 → MMS permanente
2 → MMS não permanente
```

Mais informações a respeito de formulações e aplicações podem ser encontradas no artigo Monteiro et al. (2019).

3.2.9 Coeficientes Específicos

Existem ainda 4 parâmetros muito importantes para a simulação (aqui tratados como coeficientes), são eles: `tetahIT`, `chezy`, `decliv`, e `z0IC` encontrados em `0_module.f90`, mas também determinados inicialmente em `00_suli3d.prm`.

```
tetahIT → Coeficiente ponderador de implicitude do termo da pressão; aconselha-
se manter o valor de 0,6 – Patnaik et al.
(1987)
chezy → Coeficiente de Chezy (C); pode ser aproximado ao coeficiente de
Manning (n) através da equação  $C = Rh^{1/6}/n$ 
decliv → Declividade (m/m) aplicada ao longo do comprimento do domínio
z0IC → Altura da rugosidade de fundo (m)
```

3.2.10 Camada Esponja

A camada esponja é um artifício utilizado para eliminar ondas de dentro do domínio computacional, muito utilizado como condição de contorno de saída para que as ondas não retornem quando passarem por esta condição de contorno. Na pasta `00_suli3d.prm`, define-se:

```
esp_type
0 → Sem camada esponja
1 → Considerar a profundidade
2 → Desconsiderar a profundidade
3 → Método da Tangente Hiperbólica
```

Mais características como o comprimento da camada espoja e sua posição podem ser alteradas da subtorina `sponge_layer`, dentro do arquivo `3_contorno.f90`.

3.2.11 Pressão Hidrostática

O coeficiente `t_press`, determinado no `00_suli3d.prm`, especifica o tipo de cálculo de pressão a ser utilizado na simulação.

```
t_press
0 → Aproximação Hidrostática de pressão
1 → Aproximação Não-Hidrostática
```

Aproximação Hidrostática: Considera que a pressão no fluido varia apenas devido ao peso do próprio fluido

Aproximação Não-Hidrostática: Considera todas as componentes da pressão, incluindo as variações dinâmicas causadas pelo movimento do fluido. É uma abordagem mais completa e precisa para simulações onde a pressão varia significativamente devido às forças inerciais e viscosas.

3.2.12 Level-Set^{IC}

O *Level-Set* é um método para caracterização de diferentes fluidos imiscíveis envolvidos na simulação (água e ar, por exemplo). No modulo `00_suli3d.prm` é possível personalizar estas configurações, que serão aplicadas ao `14_level_set.f90`, a partir dos seguintes parâmetros:

α_1 : Número de células que farão parte da espessura da interface entre fluidos, representado por `alpha1`.

ρ : Massa específica (kg/m^3) dos fluidos 1 e 2, representadas por `rho_f1` e `rho_f2`, respectivamente.

μ : Viscosidade dinâmica ($Pa.s$) dos fluidos 1 e 2, representadas por `mi_f1` e `mi_f2`, respectivamente.

σ : Tensão superficial (N/m), representada por `sigma`; pode ser ativada utilizando o comando `t_tens = 1` ou desativada com `t_tens = 0`.

Para isso, existem pré-definidas 7 possibilidades para a definição da interface entre o ar e a água onde:

tipo	
1	→ Onda
2	→ Onda 2 (Criação de onda no início do domínio baseado em Higuera et al. (2013))
3	→ Barragem
4	→ Gota
5	→ Fennema and Chaudhry (1990) ou Aureli, Maranzoni, Mignosa, and Ziveri (2008)
6	→ MMS – Wang et al. (2009)
7	→ Koshizuka, Oka, and Tamako (1995) .

Dessa forma, após definir o `tipo` de interface, é possível configurar dados complementares para a simulação, como:

Λ_x : Lâmbda X (m), define o comprimento de uma onda ou perturbação ao longo do eixo x.

Λ_y : Lâmbda X (m), define o comprimento de uma onda ou perturbação ao longo do eixo y.

Amplitude (m): Refere-se à medida da variação máxima da altura do fluido causada por uma onda (`tipo = 1`); Fator de adição do comprimento em todas as direções (`tipo = 3`); Raio ou metade de uma aresta de uma gota (`tipo = 4`).

Profundidade (m): Distância vertical do fundo do domínio até a superfície livre do fluido. Podendo ser: a profundidade do escoamento sem a onda em `tipo = 1` ou comprimento da onda na direção z na simulação `tipo = 3`.

Curvatura do Chanfro (*adimensional*): Curvatura do chanfro entre planos na barragem (`tipo = 3`) e na gota (`tipo = 4`).

3.2.13 Restart^{IC}

Esta função serve para auxiliar o usuário em situações em que exista a necessidade de interromper a simulação sem perder o progresso da mesma, fazendo um backup que possa ser utilizado para reiniciar a simulação. Os parâmetros representados por `interv_rest` e `irest` são acessíveis em `0_modules.f90`, módulo *restart* e aplicados nas subrotinas *restart_ini* do arquivo `11_diversos.f90`.

```
interv_rest = intervalo de tempo a ser salvo para o restart
irest
  0 → simulação não é um restart de outra
  1 → simulação é um restart de outra
```

3.3 Condições de contorno

Em `modules.f90`, devido ao seu domínio tridimensional, o **SuLi** possui 6 contornos espaciais a serem definidos com até 5 opções diferentes, explicados logo na sequência:

```
ccx0  0 → periódico
      1 → free-slip
      2 → no-slip
      3 → prescrita

ccxf  0 → periódico
      1 → free-slip
      2 → no-slip
      3 → prescrita
      4 → saída livre

ccy0  0 → periódico
      1 → free-slip
      2 → no-slip
      3 → prescrita

ccyf  0 → periódico
      1 → free-slip
      2 → no-slip
      3 → prescrita

ccz0  1 → free-slip
      2 → no-slip
      3 → prescrita

cczf  1 → free-slip
      3 → prescrita
```

Onde:

```
cc = condição de contorno
x,y,z = eixos do escoamento
0,f = início/fim do eixo de escoamento
```

Periódico: Tudo que entra, sai do domínio e vice-versa, seguindo um regime de fluxo periódico. É importante citar que adotando a condição periódica no início (`ccx0`, `ccy0` e/ou `ccz0`) é necessário adotar também para o final (`ccxf`, `ccyf` e/ou `cczf`)

Free-slip: Deslizamento livre, não há velocidade normal, apenas tangencial.

No-slip: Ausência de deslizamento, portanto, velocidade normal e tangencial nulas.

Prescrita: Velocidade imposta pelo usuário, prescrita conforme desejar.

Saída livre: Fornece liberdade de saída para o escoamento, mas necessita a adição de uma camada esponja para suavizá-lo ao final do domínio.

4 Visualização dos resultados

Como já mencionado, a opção dos autores para visualização dos resultados da simulação é o programa livre **Paraview**, disponível em www.paraview.org/, com versões disponíveis para usuários de Linux e/ou Windows. No Linux (Ambiente Ubuntu) a instalação do paraview pode ser facilmente realizada por meio do comando

```
sudo apt install paraview
```

No Linux, para visualizar os resultados dos campos das variáveis calculadas com o **SuLi** deve-se acessar a pasta "arquivos" via terminal e digitar `paraview visu.xdmf`. No Windows, o comando direto via terminal não é possível, sendo assim, a opção é iniciar o Paraview em janela independente e abrir a pasta `visu.xdmf` diretamente nele.

O paraview acessará então os dados calculados pelo **SuLi** e os apresentará de acordo com as ferramentas disponíveis na versão e interface do programa. O intervalo de criação dos campos (os frames da simulação) para esta posterior visualização é definido pelo parâmetro `dt_frame` determinado em `00_suli3d.prm`

Assim, o usuário, tendo um conhecimento básico do programa, pode usufruir dos resultados de sua simulação.

A Apêndice: Ambiente Ubuntu no Windows

Preparando o Windows: É necessário habilitar o sistema para aceitar qualquer distribuição Linux que venha a ser utilizada, para tal siga o caminho abaixo.

No "Menu Iniciar" do seu Windows vá para o "Painel De Controle".

Então clique em "Programas".

Depois clique em "Ativar ou desativar recursos do Windows".

Na janela que será aberta habilite a opção "Subsistema do Windows para Linux"

Após ter realizado esse procedimento, é preciso reiniciar seu computador para que as alterações sejam devidamente atualizadas, caso contrário o usuário enfrentará problemas na próxima etapa de instalação.

Instalando o Ubuntu: Após a reinicialização do sistema, é preciso instalar efetivamente a distribuição Linux desejada, neste caso, o ambiente Ubuntu. Para tanto, abra o aplicativo "Microsoft Store" e pesquise por "Ubuntu", faça o download e abra o aplicativo Ubuntu que foi instalado em seu computador.

A primeira vez que estiver acessando o ambiente Ubuntu, será pedido que um usuário e uma senha sejam cadastrados, para que o processo de instalação seja efetuado com sucesso. Guarde estas informações, pois elas podem ser importantes para a utilização do Ubuntu como subsistema do Windows.

Agora, com o Windows configurado e o ambiente Ubuntu devidamente instalado, basta retornar ao tópico 2.2.1 para a instalação do **SuLi**.

Acessando dados do Windows no subsistema Ubuntu: No terminal do subsistema Ubuntu é possível acessar os arquivos do Windows, dentro da pastar "mnt/" localizada na pasta raiz do sistema.

Caso o SuLi esteja na pasta c: no Windows, é possível acessar a pasta pelo comando: `cd ../../mnt/c/suli` . Observe que neste exemplo a pasta que possui o código do SuLi se chama "suli".

Observação: Vale frisar que sempre que for necessário realizar uma ação no terminal de comando do computador, o usuário deverá utilizá-lo através do ambiente Ubuntu.

A utilização do ambiente Ubuntu como subsistema fica a cargo do próprio usuário, outros subsistemas compatíveis podem oferecer o mesmo resultado (pesquise sobre outras distribuições para subsistemas!).

Bibliografia

- Auguste, F., Réa, G., Paoli, R., Lac, C., Masson, V., & Cariolle, D. (2019, 7). Implementation of an immersed boundary method in the meso-nh v5.2 model: applications to an idealized urban environment. *Geoscientific Model Development*, 12, 2607–2633. Retrieved from <https://gmd.copernicus.org/articles/12/2607/2019/> doi: 10.5194/gmd-12-2607-2019
- Aureli, F., Maranzoni, A., Mignosa, P., & Ziveri, C. (2008). Dam-break flows: acquisition of experimental data through an imaging technique and 2d numerical modeling. *Journal of Hydraulic Engineering*, 134(8), 1089–1101. doi: 10.1061/(ASCE)0733-9429(2008)134:8(1089)
- Beji, S., & Battjes, J. (1994, 05). Numerical simulation of nonlinear wave propagation over a bar. *Coastal Engineering*, 23, 1–16. doi: 10.1016/0378-3839(94)90012-4
- Fennema, R. J., & Chaudhry, M. H. (1990). Explicit methods for 2-d transient free surface flows. *Journal of Hydraulic Engineering*, 116(8), 1013–1034.
- Heinz, S. (2020, 4). A review of hybrid rans-les methods for turbulent flows: Concepts and applications. *Progress in Aerospace Sciences*, 114, 100597. Retrieved from <https://linkinghub.elsevier.com/retrieve/pii/S0376042119301861> doi: 10.1016/j.paerosci.2019.100597
- Higuera, P., Lara, J. L., & Losada, I. J. (2013). Realistic wave generation and active wave absorption for navier–stokes models: Application to openfoam®. *Coastal Engineering*, 71, 102–118.
- Koshizuka, S., Oka, Y., & Tamako, H. (1995). *A particle method for calculating splashing of incompressible viscous fluid* (Tech. Rep.). American Nuclear Society, Inc., La Grange Park, IL (United States).
- Lilly, D. K. (1967). The representation of small scale turbulence in numerical simulation experiments. *National Center for Atmospheric Research*. doi: 10.5065/D62R3PMM
- Monteiro, L. R., Lucchese, L. V., & Schettini, E. B. C. (2017). Distribuição de pressão em escoamentos com superfície livre devido à curvatura das linhas de corrente. In *Xxii simpósio brasileiro de recursos hídricos*.
- Monteiro, L. R., Lucchese, L. V., & Schettini, E. B. C. (2019). Comparison between hydrostatic and total pressure simulations of dam-break flows. *Journal of Hydraulic Research*, 58(5), 725–737. doi: 10.1080/00221686.2019.1671509
- Monteiro, L. R., & Schettini, E. B. C. (2014). Comparação entre a aproximação hidrostática e a não-hidrostática na simulação numérica de escoamentos com superfície livre. *Revista Brasileira de Recursos Hídricos*, 20(4). doi: 10.21168/rbrh.v20n4.p1051-1062
- Nasr-Azadani, M. M., & Meiburg, E. (2014). Turbidity currents interacting with three-dimensional seafloor topography. *Journal of Fluid Mechanics*, 745, 409–443. doi: 10.1017/jfm.2014.47
- Patnaik, G., et al. (1987). A barely implicit correction for flux-corrected transport. *Journal of Computational Physics*, 71(1), 1–20.
- Rigotti, J. A., et al. (2015). Funcionalidade ecológica em cursos d’água: implicações para a requalificação da paisagem fluvial. *Mestrado Acadêmico no Programa de Pós-graduação em Geografia da Universidade Federal de Santa Catarina*.
- Rijn, L. v. (1980). *Storm surge barrier oosterschelde - computation of siltation in dredged trenches: Semi-empirical model for the flow in dredged trenches* [investigation].
- Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3), 99–164. doi: 10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2
- Wang, S. S., Roache, P. J., Jia, Y., Schmalz, R. A., & Smith, P. E. (2009). Verification and validation of 3d free-surface flow models.. doi: 10.1061/9780784409572

- Yue, W., Lin, C.-L., & Patel, V. C. (2003). Numerical simulation of unsteady multidimensional free surface motions by level set method. *International Journal for Numerical Methods in Fluids*, 42(8), 853–884. doi: 10.1002/fld.555
- Zampiron, A., et al. (2022). Flow development in rough-bed open channels: mean velocities, turbulence statistics, velocity spectra, and secondary currents. *Journal of Hydraulic Research*. Retrieved from <https://doi.org/10.1080/00221686.2022.2132311>