



SuLi - Simulador de Escoamentos com Superfície Livre

Manual do Usuário

Leonardo Romero Monteiro

Pedro Miguel de Andrade Filho

Vinícios Morigi

Versão 1.0
Janeiro de 2021



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; Andrade Filho, Pedro Miguel; Morigi, Vinícios

Manual do Usuário: SuLi - Simulador de Escoamentos com Superfície Livre - Joinville/SC:
UDESC/CCT/DEC/LaCiA, 2021.

16 Páginas

Palavras-chave:

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

Versão 1.0

Edição brasileira
Janeiro de 2021

Conteúdo

1	Introdução	2
1.1	Objetivo	2
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	Usuários Linux e Windows	3
2.3	Estrutura	4
2.4	Compilando	4
2.5	Rodando	5
3	Personalizando o código	6
3.1	Definindo o domínio	6
3.2	Condições e Configurações Iniciais	7
3.2.1	Métodos De Evolução Temporal ²	7
3.2.2	Termo Advectivo	7
3.2.3	Derivada Espacial	8
3.2.4	Obstáculos	8
3.2.5	Modelo de Turbulência	8
3.2.6	Ondas	9
3.2.7	Plotagem ²	9
3.2.8	Método da Solução Manufaturada	9
3.2.9	Coefficientes Específicos	10
3.2.10	Camada Esponja	10
3.2.11	Ativação do coeficiente de Chezy ¹	10
3.3	Condições de Contorno	10
3.4	Funções auxiliares	11
3.4.1	Obstáculos	12
3.4.2	Level-Set ²	12
3.4.3	Restart ²	12
4	Visualização Dos Resultados	14
A	Apêndice: ambiente Ubuntu no Windows	15
A.1	Preparando o Windows	15
A.2	Instalando o Ubuntu	15
A.3	Acessando os dados do Windows no subsistema Ubuntu	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). Hoje ele 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.

Este é um código 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 2.2.1).

Para o pós-processamento e visualização 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.

Obs.: 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 Usuários Linux e Windows

Para usuários Linux (ou Windows que já cumpriram o passo anterior), é necessário garantir que o sistema operacional esteja atualizado a fim de evitar erros de compatibilidade. Para tal, **no terminal de comando do Linux**, digite os comandos enumerados a seguir:

1) `sudo apt-get update`

(Compreende os programas e arquivos desatualizados)

2) `sudo apt-get upgrade`

(Efetivamente atualiza 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:

3) `sudo apt-get install gfortran`

(Para instalar o gfortran)

4) `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 no Capítulo 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 que serão aprofundadas ao longo do presente Manual:

- Compilação e inicialização do programa (subseções 2.4 e 2.5):
 - Arquivo gerador/compilador inicial “Makefile”.
- Cálculo da simulação numérica (seção 3):
 - 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.

2.4 Compilando

A partir do terminal de comando do Linux, navegue até o local onde está localizada a pasta do **SuLi**. Ciente dos arquivos presentes na pasta, o usuário pode digitar alguns comandos no terminal de forma 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).

Toda vez que o usuário modificar algo na escrita das rotinas no código (nos arquivos .f90), deve-se compilá-lo novamente com o comando **make**. Entretanto, se a mudança for em um módulo então deve-se primeiro remover os arquivos auxiliares (**make clean**) e depois sim compilar (**make**). *ACHO QUE PODERIAMOS EXPLICAR MELHOR AQUI O PASSO A PASSO PARA FICAR MAIS DIDÁTICO P QUEM NUNCA MEXEU*

* O comando **make** fornece liberdade ao usuário para definir itens específicos relevantes à compilação dentro do arquivo Makefile. Nesse sentido, ao ativar itens na linha “OPTFC =” haverá mudanças importantes nas 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).

Contudo, usuários que não tenham conhecimento do uso do Makefile devem utilizar a opção padrão definida (*EXEMPLIFICAR A OPÇÃO PADRÃO DEFINIDA*).

2.5 Rodando

Depois de compilado, para rodar o **SuLi** basta digitar no terminal o comando `./rodar_3d`

Note que, tanto a pasta na qual o programa foi instalado, quanto as pastas “dados” e “arquivos” vão receber novos arquivos auxiliares durante o processo da simulação.

Para interromper a simulação, feche o terminal de comando ou aperte o comando `Ctrl+c`.

3 Personalizando o código

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

Nesta seção são definidos: o domínio de cálculo, as funções a serem utilizadas para o cálculo das equações, as condições iniciais e as condições de contorno. Aqui entram 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 um. Quando o item for específico do **SuLi-IT** a função/variável terá a marcação ¹, caso for específico do **SuLi-IC** terá a marcação ².

O **SuLi** é composto de códigos reunidos em documentos de subrotina, dos quais se complementam e se interdependem, a seguir serão abordadas modificações que podem ser realizadas nos módulos:

0_modules.f90	1_principal.f90	2_iniciais.f90	3_contorno.f90
4_plot.f90	5_viscosidade.f90	6_convdiff.f90	7_adectivo.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	

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 `0_modules.f90` 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	$\rightarrow (nx + 1) \times dx \rightarrow$	Comprimento	(m)
Eixo y	$\rightarrow (ny + 1) \times dy \rightarrow$	Largura	(m)
Eixo z	$\rightarrow (nz + 1) \times dz \rightarrow$	Profundidade	(m)

A malha temporal da simulação é definida pelas seguintes variáveis:

dt \rightarrow	Intervalo de tempo (s)*, ²
ts \rightarrow	Tempo total (s)

***,²** Toda vez que o parâmetro `dt` for alterado, seu respectivo valor também deverá ser alterado no denominador da expressão do parâmetro `ts`!

Observação: É importante salientar que 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

Continuando em `0_modules.f90`, a primeira condição inicial importante em relação ao escoamento é o parâmetro `u_inicial`; 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 Métodos De 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 `0_modules.f90` e aplicado em `11_diversos.f90`.

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

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): possui 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. Entretanto, 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` no módulo `7_advectivo.f90`

```
adv_type
→ 1 para Advectivo Clássico
→ 2 para Rotacional
→ 3 para 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 Derivada Espacial

As derivadas espaciais são aproximadas 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**, no arquivo `0_module.f90`.

```
der
→ 1 para Upwind
→ 2 para Centrado*
→ 3 para Upwind de 2ª Ordem*,2
```

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*,2: Mais preciso do que o Upwind de primeira 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

No **SuLi** é possível adicionar obstáculos ao domínio da simulação do escoamento. Este parâmetro é representado no código por **obst_t**, do qual estão acessíveis 10 obstáculos pré-definidos, parâmetro **obst_t** $\in [0, 10]$, cujas características e origens estão referenciadas na seção 3.4.1.

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, aqui expostas através dos estudos de LES (Large Eddy Simulation - Simulação de Grandes Vórtices) de Joseph Smagorinsky. O uso de cada opção depende do nível de refino da malha utilizada na simulação. Este parâmetro é representado por **m_turb**.

```
m_turb
→ 0 para Sem Modelo
→ 1 para LES Smagorinsky-Lilly Clássico
→ 2 para LES Smagorinsky-Lilly Direcional
```

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: quando do uso de uma malha não tão refinada, tendo no modelo de turbulência a representação aproximada dos efeitos.

LES Smagorinsky-Lilly Direcional: quando do uso de uma malha não tão refinada, tendo no modelo de turbulência a representação aproximada dos efeitos. A denominação "Direcional" surgiu após manipulações da equipe. A aplicação deste modelo ainda não foi publicada e nem validada.

3.2.6 Ondas

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` são:

```
wave_t
→ 0 para Sem Onda
→ 1 para Onda De Stokes I
→ 2 para Onda De Stokes II
→ 5 para 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 Plotagem²

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 para Modo Simples
→ 1 para 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`. A presente solução manufaturada é descrita com detalhes por [Wang, Roache, Jia, Schmalz, and Smith \(2009\)](#).

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

Os parâmetros a e $h\theta$, encontradas no arquivo `0_module.f90` referem-se a altura da onda e a profundidade do escoamento, respectivamente, para a verificação do código.

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

3.2.9 Coeficientes Específicos

Existem ainda 3 parâmetros muito importantes para a simulação (aqui tratados como coeficientes), são eles: θ^1 , `chezy` e `decliv` encontrados em `0_module.f90`.

Ponderador θ^1 : Este é um coeficiente ponderador de implicitude do termo da pressão. Tem função apenas na versão monofásica. Aconselha-se deixar sempre no valor de 0,6. Este parâmetro é baseado no trabalho de [Patnaik, Guirguis, Boris, and Oran \(1987\)](#).

Chezy: coeficiente de rugosidade do escoamento. O coeficiente de Chezy (Cz) pode ser aproximado ao coeficiente de Manning (n) através da equação

$$Cz = \frac{Rh^{1/6}}{n}.$$

Decliv: Declividade (m/m) aplicada ao longo do comprimento do domínio.

Algumas outras condições iniciais devem ser impostas no arquivo `2_iniciais.f90`. Variáveis como a velocidade (u, v, w), as pressões (`prd1`) e desnível¹ (`eta1`) podem ser inseridas de acordo com o usuário sem grandes dificuldades (sempre respeitadas as condições de estabilidade do código).

No módulo **SuLi-IC**, o desnível da água (a posição da interface entre a água e o ar) deve ser definida na subtirona `level_set_ini` dentro do arquivo `14_level_set.f90`, melhor detalhado no subitem 3.4.2.

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.

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

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

3.2.11 Ativação do coeficiente de Chezy¹

Está é uma simples função para ativar ou desativar o coeficiente de Chezy no contorno do domínio computacional.

```
chezy_t
→ 0 para Desconsiderar atrito no fundo
→ 1 para Considerar atrito no fundo
```

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	para	periódico
	→	1	para	free-slip
	→	2	para	no-slip
	→	3	para	prescrita
ccxf	→	0	para	periódico
	→	1	para	free-slip
	→	2	para	no-slip
	→	3	para	prescrita
	→	4	para	saída livre
ccy0	→	0	para	periódico
	→	1	para	free-slip
	→	2	para	no-slip
	→	3	para	prescrita
ccyf	→	0	para	periódico
	→	1	para	free-slip
	→	2	para	no-slip
	→	3	para	prescrita
ccz0	→	1	para	free-slip
	→	2	para	no-slip
	→	3	para	prescrita
cczf	→	1	para	free-slip
	→	3	para	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.

3.4 Funções auxiliares

Nesta seção estão abordados tópicos relacionados a funções/rotinas auxiliares á aplicação do **SuLi**, em áreas ainda mais aprofundadas do código. Portanto, adverte-se a necessidade de possuir um entendimento maior do propósito de cada aplicação antes de alterá-la em código.

3.4.1 Obstáculos

Assim como o usuário pode optar por não adicionar nenhum 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. A seguir estão as opções pré-definidas citadas na seção 3.2.4 e suas respectivas referências, aplicadas ao código no módulo `3_contorno.f90`:

```
obst_t
0 → Sem Obstáculo
1 → Dunas
2 → Yue, Lin, and Patel \(2003\)
3 → Nasr-Azadani and Meiburg \(2014\)
4 → Beji and Battjes \(1994\)
5 → canal 0 — Rijn \(1980\)
6 → canal 1 e 2 — Rijn \(1980\)
7 → Monteiro, Lucchese, and Schettini \(2017\)
8 → Rigotti et al. \(2015\)
```

3.4.2 Level-Set²

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

α_1 : Número de células que farão parte da espessura da interface entre fluidos.

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)

Existem pré-definidas 7 possibilidades para a definição da interface entre o ar e a água onde:

```
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 \(1995\).
```

3.4.3 Restart²

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 para indicar se a simulação não é um restart de outra  
→ 1 para indicar se a simulação é um restart de outra
```

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 a seguir:

```
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 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` no módulo `0_modules.f90`

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

A Apêndice: ambiente Ubuntu no Windows

A.1 Preparando o Windows

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

- 1) No "Menu Iniciar" do seu Windows, vá para o "Painel De Controle";
- 2) Clique em "Programas";
- 3) Clique em "Ativar ou desativar recursos do Windows";
- 4) 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.

A.2 Instalando o Ubuntu

Após a reinicialização do sistema, é preciso **instalar** efetivamente a distribuição Linux desejada, neste caso, o ambiente Ubuntu. Dessa forma, siga a enumeração seguinte:

- 1) Abra o aplicativo ‘‘Microsoft Store’’;
- 2) Pesquise por ‘‘Ubuntu’’;
- 3) Faça o download;
- 4) 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**.

A.3 Acessando os dados do Windows no subsistema Ubuntu

No terminal do subsistema Ubuntu é possível acessar os arquivos do Windows, dentro da pasta "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 também 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

- 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.
- 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).
- 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., Guirguis, R., Boris, J., & Oran, E. (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].
- 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