

4 Modelo Computacional

Modelagem computacional é uma área de conhecimento multidisciplinar que trata da aplicação de modelos matemáticos à análise, compreensão e estudo de problemas complexos em diversas áreas. É a área que trata da simulação de soluções para problemas científicos, analisando os fenômenos, desenvolvendo modelos matemáticos para sua descrição, e elaborando códigos computacionais para obtenção daquelas soluções.

A Dinâmica dos Fluidos Computacional, DFC, é a área científica que, com a ajuda de computadores digitais, analisa os movimentos dos fluidos de acordo com suas leis de conservação.

As etapas envolvidas em uma análise de DFC basicamente são pré-processamento, onde são informados todos os dados referentes à geometria e condições iniciais do problema, algoritmos de solução, que constitui um conjunto de técnicas numéricas utilizadas na discretização das equações governantes do fenômeno, e finalmente o pós-processamento, responsável pela visualização gráfica dos resultados obtidos.

4.1. Técnicas Numéricas

As equações utilizadas para modelagem do problema em estudo nesse trabalho são equações diferenciais parciais (EDP) não-lineares e, segundo Fortuna (2000), a teoria matemática dessa classe de equações ainda não está suficientemente desenvolvida para permitir a obtenção das soluções analíticas em regiões arbitrárias e em condições de contorno gerais. Diversos métodos numéricos são utilizados para resolução dessas EDPs, entre eles volumes finitos, elementos finitos, diferenças finitas, entre outros. As aproximações por diferenças finitas são bastante empregadas na dinâmica dos fluidos computacionais, assim como, nesta pesquisa.

A tarefa do método numérico é então resolver as equações diferenciais parciais, substituindo as derivadas existentes por expressões algébricas que envolvem a função incógnita. As soluções analíticas dessas equações fornecem,

a princípio, a solução em uma forma fechada, e seria possível, obter valores das variáveis dependentes em qualquer ponto em nível infinitesimal.

Assim, para que seja possível tratar numericamente as EDPs, deve-se representá-las por meio de expressões algébricas, ou seja, discretizar as EDPs. Dessa forma, essas expressões podem ser manipuladas por computadores, relacionando entre si os valores das grandezas nos pontos discretos de uma dada região R . Logo, antes de resolver EDPs de forma numérica deve-se encontrar, para os termos que nela aparecem, as expressões escritas em termos dos pontos da malha. Essas expressões são denominadas aproximações por diferenças finitas, e o resultado final desse processo é uma equação algébrica, denominada equação de diferenças finitas, EDF. A EDF é escrita em cada ponto da região discretizada em que se deseja calcular a solução do problema, e resolvendo-se as EDFs encontra-se a solução aproximada do problema.

A expressão, em diferenças finitas, que representa uma aproximação de primeira ordem para a primeira derivada da função f , utilizando diferença progressiva é dada por:

$$\left. \frac{\partial f}{\partial x} \right|_i = \frac{f_{i+1} - f_i}{\Delta x} \quad (50)$$

Nota-se que na eq. (50) o cálculo da derivada no ponto x_i é feito utilizando-se um ponto adiante. Porém, utilizando-se um ponto anterior a x_i , tem-se uma aproximação de primeira ordem para a primeira derivada da função f , utilizando diferença atrasadas.

$$\left. \frac{\partial f}{\partial x} \right|_i = \frac{f_i - f_{i-1}}{\Delta x} \quad (51)$$

Ou ainda, podem-se combinar as eq. (50) e (51) e obter-se uma aproximação que utiliza os pontos x_{i-1} e x_{i+1} para o cálculo da derivada no ponto x_i , neste caso, a aproximação de segunda ordem da primeira derivada da função f , é denominada diferença central.

$$\left. \frac{\partial f}{\partial x} \right|_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x} \quad (52)$$

As EDPs que descrevem os fenômenos da dinâmica dos fluidos computacional podem ser classificadas em três categorias: elípticas, parabólicas e hiperbólicas. Na formulação matemática das correntes de turbidez, as equações que regem o fenômeno é dada pela eq. (24), que é uma equação diferencial parcial hiperbólica.

Situações descritas por esse tipo de equação, necessitam de valores iniciais da grandeza física na região que estamos buscando a solução e

condições de fronteira. A solução numérica de problemas hiperbólicos é mais complicada que de problemas elípticos e parabólicos devido à velocidade finita de propagação das informações através do domínio computacional, e à falta de mecanismos dissipativos. O método numérico adotado para solução das equações governantes deve refletir, então, essas duas propriedades. Além disso, ele deve ser consistente com as velocidades e direções que as informações estão sendo propagadas através do campo de escoamento, isto nada mais é do que obedecer às leis da física.

Diferenças finitas centrais nem sempre carregam adequadamente as informações através do campo de escoamento e, em muitos casos elas extraem informações externas ao domínio do problema, e isto pode comprometer a precisão da solução numérica. Um método que não apresenta essa dificuldade é a chamada discretização à montante, *upwind scheme*, em que a aproximação da derivada espacial depende da direção de propagação das características, isto é, do sinal da velocidade.

Problemas que envolvem a variação temporal de grandezas físicas são ditos problemas transientes ou de marcha. Esses problemas são tais que a partir de valores iniciais de determinada grandeza no intervalo de tempo t_0 calcula-se, através da solução numérica das EDPs, os valores dessas grandezas em sucessivos intervalos de tempo Δt até que se alcance o instante de tempo final t_f (Fortuna, 2000).

$$t_0 + \Delta t, t_0 + 2\Delta t, t_0 + 3\Delta t, \dots, t_f - \Delta t, t_f \quad (53)$$

Em outras palavras, a discretização temporal visa a fornecer relações entre os valores de determinada grandeza, T , em instantes sucessivos de tempo. Assim, de posse dos valores de T no instante de tempo n , T^n , calcula-se os valores de T no instante de tempo $n + 1$, T^{n+1} .

4.1.1. Solução das Equações do Problema

As equações matemáticas do modelo numérico para simulação das correntes de turbidez, proposto no contexto deste estudo, a serem resolvidas são:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(h\bar{u}) + \frac{\partial}{\partial y}(h\bar{v}) = -\frac{S_k}{c_k} \quad (54)$$

$$\frac{\partial \bar{u}}{\partial t} + \frac{\partial}{\partial x}(\bar{u}^2) + \frac{\partial}{\partial y}(\bar{u}\bar{v}) = -\frac{1}{\rho} \left[\Delta \rho g \frac{\partial H}{\partial x} + \rho \bar{u} \sqrt{\bar{u}^2 + \bar{v}^2} \left(\frac{k}{b \ln \left(\frac{hf}{z_0} \right)} \right)^2 \right] \quad (55)$$

$$\frac{\partial \bar{v}}{\partial t} + \frac{\partial}{\partial y}(\bar{v}^2) + \frac{\partial}{\partial x}(\bar{u}\bar{v}) = -\frac{1}{\rho} \left[\Delta \rho g \frac{\partial H}{\partial y} + \rho \bar{v} \sqrt{\bar{u}^2 + \bar{v}^2} \left(\frac{k}{b \ln \left(\frac{hf}{z_0} \right)} \right)^2 \right] \quad (56)$$

A solução dessas equações exige alguns cuidados especiais com relação ao armazenamento das incógnitas h , \bar{u} e \bar{v} na malha. Uma alternativa seria considerar a malha deslocada (Fortuna, 2000). Esse tipo de malha consiste em armazenar as alturas h do fluxo e d do depósito no centro da célula, a componente \bar{u} da velocidade na direção x nas faces laterais, que distam $\pm \Delta x/2$ do centro, e a componente \bar{v} da velocidade na direção y nas faces superiores e inferiores, que distam $\pm \Delta y/2$ do centro.

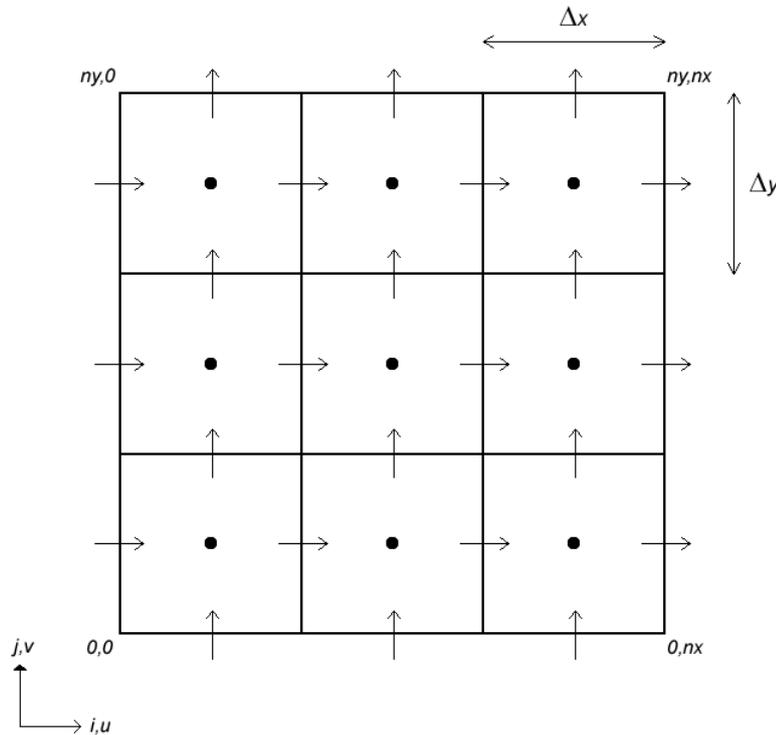


Figura 18 - Célula da malha desloçada. A altura é armazenada no centro enquanto que as velocidades são armazenadas no centro

A Figura 18 mostra uma parte do domínio computacional discretizado com a malha desloçada. A equação do momento na direção x , eq. (55), é então discretizada sobre a posição $i + 1/2, j$. Analogamente, a equação do momento na direção y , eq. (56), é discretizada sobre a posição $i, j + 1/2$. Por fim, a

equação da continuidade, eq. (54), é discretizada sobre o centro da célula, no ponto i, j .

Considere-se a discretização da equação de momento na direção x na célula de diferenças finitas, mostrada na Figura 19.

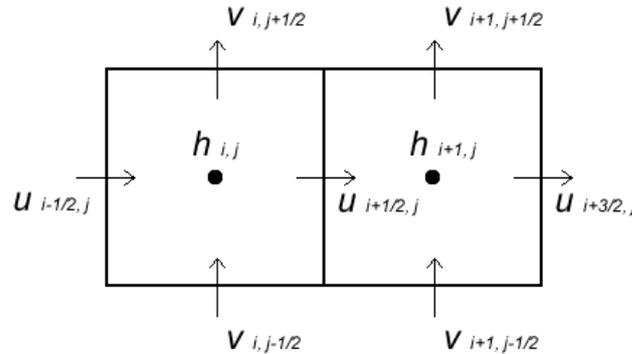


Figura 19 – Discretização das derivadas presentes na equação de momento na direção x

A discretização no ponto $i + 1/2, j$, dos termos convectivos e do termo que relaciona a pressão, é feita utilizando a aproximação por diferenças centrais de segunda ordem.

$$\left. \frac{\partial H}{\partial x} \right|_{i+1/2,j} \approx \frac{H_{i+1,j} - H_{i,j}}{\Delta x} \quad (57)$$

$$\left. \frac{\partial}{\partial x} (\bar{u}^2) \right|_{i+1/2,j} \approx \frac{\bar{u}^2_{i+1,j} - \bar{u}^2_{i,j}}{\Delta x} \quad (58)$$

$$\left. \frac{\partial}{\partial y} (\bar{uv}) \right|_{i+1/2,j} \approx \frac{(\bar{uv})_{i+1/2,j+1/2} - (\bar{uv})_{i+1/2,j-1/2}}{\Delta y} \quad (59)$$

Os termos $\bar{u}^2_{i+1,j}$, $\bar{u}^2_{i,j}$, $(\bar{uv})_{i+1/2,j+1/2}$ e $(\bar{uv})_{i+1/2,j-1/2}$ nas eq. (58) e (59) não estão definidos na malha, e são obtidos a partir da interpolação linear dos valores de \bar{u} e \bar{v} das faces das células. O valor de H , na eq. (57), é dado pelo somatório da espessura do depósito, d , da coordenada z e da altura do fluxo, h , em cada ponto.

No último termo da eq. (55) o valor da velocidade \bar{v} e da altura h não estão definidos para o ponto $i + 1/2, j$ da malha. O valor da velocidade \bar{v} , nesse ponto, é obtido então pela média dos valores das velocidades \bar{v} nas faces das células adjacentes.

$$\bar{v}|_{i+1/2,j} \approx \frac{\bar{v}_{i,j-1/2} + \bar{v}_{i+1,j-1/2} + \bar{v}_{i,j+1/2} + \bar{v}_{i+1,j+1/2}}{4} \quad (60)$$

E o valor da altura h no ponto $i + 1/2, j$ é obtido pela média dos valores das alturas no centro das células adjacentes.

$$h|_{i+1/2,j} \approx \frac{h_{i+1,j} + h_{i,j}}{2} \quad (61)$$

Analogamente, a discretização no ponto $i, j + 1/2$ das derivadas presentes na equação de momento na direção y é obtida segundo os pontos mostrados na Figura 20.

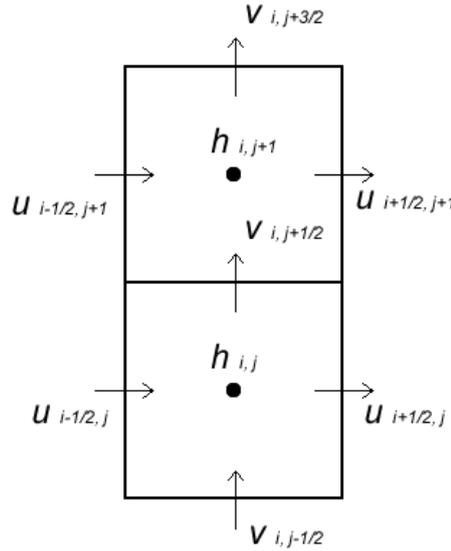


Figura 20 – Discretização das derivadas presentes na equação de momento na direção y

Assim como na equação de momentos na direção x , os temos, $\bar{v}^2_{i,j+1}$, $\bar{v}^2_{i,j}$, $(\bar{uv})_{i+1/2,j+1/2}$ e $(\bar{uv})_{i-1/2,j+1/2}$ também não estão definidos na malha, e são interpolados linearmente a partir dos valores de \bar{u} e \bar{v} das faces das células.

$$\frac{\partial H}{\partial y} \Big|_{i,j+1/2} \approx \frac{H_{i,j+1} - H_{i,j}}{\Delta y} \quad (62)$$

$$\frac{\partial}{\partial y} (\bar{v}^2) \Big|_{i,j+1/2} \approx \frac{\bar{v}^2_{i,j+1} - \bar{v}^2_{i,j}}{\Delta y} \quad (63)$$

$$\frac{\partial}{\partial x} (\bar{uv}) \Big|_{i,j+1/2} \approx \frac{(\bar{uv})_{i+1/2,j+1/2} - (\bar{uv})_{i-1/2,j+1/2}}{\Delta x} \quad (64)$$

O valor H , na eq. (62), é dado pelo somatório da espessura do depósito, d , da coordenada z e da altura do fluxo, h , em cada ponto. A média da velocidade \bar{u} , no último termo da eq. (56), é calculada da mesma forma que no caso da equação de momento na direção x , ou seja, pela média dos valores das velocidades \bar{u} nas faces das células adjacentes.

$$\bar{u} \Big|_{i,j+1/2} \approx \frac{\bar{u}_{i-1/2,j} + \bar{u}_{i+1,j} + \bar{u}_{i-1/2,j+1} + \bar{u}_{i+1/2,j+1}}{4} \quad (65)$$

E a altura h é dada pela média dos valores do centro das células adjacentes.

$$h|_{i,j+1/2} \approx \frac{h_{i,j+1} + h_{i,j}}{2} \quad (66)$$

A equação da continuidade, eq. (54), é calculada através do método da discretização à montante, *upwind*, em que a aproximação da derivada espacial depende do sinal da velocidade. Para o ponto i,j a discretização da equação da continuidade é dada de acordo com os pontos da Figura 21.

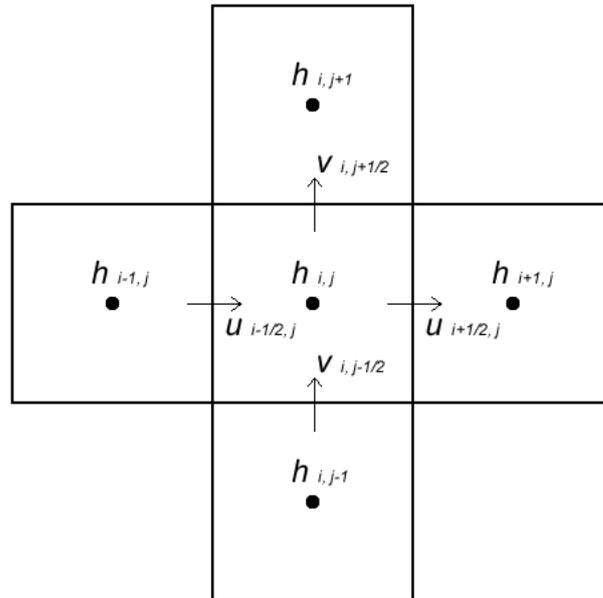


Figura 21 – Discretização das derivadas presentes na equação de continuidade

O primeiro termo do lado direito da eq. (54), que representa a derivada da função em relação a x , é dado por:

$$\frac{\partial}{\partial x} (h\bar{u}) \Big|_{i,j} \approx \frac{h_d \bar{u} - h_e \bar{u}}{\Delta x} \quad (67)$$

onde :

$$h_d = \begin{cases} h_{i,j}, & \bar{u}_{i-1/2,j} < 0 \\ h_{i-1,j}, & \bar{u}_{i-1/2,j} \geq 0 \end{cases} \quad (68)$$

$$h_e = \begin{cases} h_{i+1,j}, & \bar{u}_{i+1/2,j} < 0 \\ h_{i,j}, & \bar{u}_{i+1/2,j} \geq 0 \end{cases} \quad (69)$$

E o segundo termo, que representa a derivada da função em relação a y , é dado por:

$$\frac{\partial}{\partial y} (h\bar{v}) \Big|_{i,j} \approx \frac{h_t \bar{v} - h_b \bar{v}}{\Delta y} \quad (70)$$

onde :

$$h_t = \begin{cases} h_{i,j+1}, & \bar{v}_{i,j+1/2} < 0 \\ h_{i,j}, & \bar{v}_{i,j+1/2} \geq 0 \end{cases} \quad (71)$$

$$h_b = \begin{cases} h_{i,j}, & \bar{v}_{i,j-1/2} < 0 \\ h_{i,j-1}, & \bar{v}_{i,j-1/2} \geq 0 \end{cases} \quad (72)$$

A discretização temporal das equações de momento é baseada no método de Euler explícito, em que todos os termos que envolvem as velocidades são discretizados no nível de tempo n . A seguir o termo envolvendo a altura h do fluxo é discretizada no nível de tempo $n + 1$, de forma que, após os cálculos das velocidades \bar{u}^{n+1} e \bar{v}^{n+1} todas as variáveis do escoamento tenham sido avançadas no tempo.

Na discretização explícita tem-se a vantagem das equações serem facilmente calculadas no nível de tempo n , contudo métodos explícitos normalmente apresentam restrições de estabilidade, o que limita o valor de Δt efetivo que pode ser utilizado. Por outro lado, aproximações no nível de tempo $n + 1$ resultam discretizações implícitas que são, em geral, incondicionalmente estáveis para qualquer valor de Δt . Entretanto, elas fornecem um conjunto de equações acopladas, cuja solução exige um maior esforço computacional do que para as equações dos métodos explícitos (Fortuna, 2000).

Dessa forma, a discretização temporal da equação da continuidade, eq. (54), para a célula i, j da malha é dada por:

$$\frac{h_{i,j}^{n+1} - h_{i,j}^n}{\Delta t} \approx - \left(\frac{h_d \bar{u} - h_e \bar{u}}{\Delta x} + \frac{h_t \bar{v} - h_b \bar{v}}{\Delta y} \right) - \frac{s_k}{c_k} \quad (73)$$

E a discretização temporal das equações de momentos nas direções x e y , na face da célula i, j da malha, são dadas pelas equações (74) e (75) respectivamente.

$$\begin{aligned} & \frac{\bar{u}_{i+1/2,j}^{n+1} - \bar{u}_{i+1/2,j}^n}{\Delta t} \\ & \approx - \frac{\Delta \rho g}{\rho} \frac{H_{i+1,j} - H_{i,j}}{\Delta x} \\ & - \left(\frac{\bar{u}_{i+1,j}^2 - \bar{u}_{i,j}^2}{\Delta x} + \frac{(\bar{u}\bar{v})_{i+1/2,j+1/2} - (\bar{u}\bar{v})_{i+1/2,j-1/2}}{\Delta y} \right) \\ & - \bar{u}_{i+1/2,j}^n \sqrt{(\bar{u}_{i+1/2,j}^n)^2 + (\bar{v}_{i+1/2,j}^n)^2} \left(\frac{k}{b \ln \left(\frac{h_{i+1/2,j} f}{z_0} \right)} \right)^2 \end{aligned} \quad (74)$$

$$\begin{aligned}
 & \frac{\bar{v}_{i,j+1/2}^{n+1} - \bar{v}_{i,j+1/2}^n}{\Delta t} \\
 & \approx \frac{\Delta \rho g H_{i,j+1} - H_{i,j}}{\rho \Delta y} \\
 & - \left(\frac{\bar{v}_{i,j+1}^2 - \bar{v}_{i,j}^2}{\Delta y} + \frac{(\bar{u}\bar{v})_{i+1/2,j+1/2} - (\bar{u}\bar{v})_{i-1/2,j+1/2}}{\Delta x} \right) \quad (75) \\
 & - \bar{v}_{i,j+1/2}^n \sqrt{(\bar{u}_{i,j+1/2}^n)^2 + (\bar{v}_{i,j+1/2}^n)^2} \left(\frac{k}{b \ln \left(\frac{h_{i,j+1/2} f}{z_0} \right)} \right)^2
 \end{aligned}$$

Programar essas equações no computador da forma como foram apresentadas não seria possível, pois as linguagens de computação não suportam índices de vetores fracionados, como os utilizados para representar a célula da malha deslocada, para endereçamento das variáveis. Assim, para a implementação no computador das equações apresentadas neste trabalho adota-se os índices inteiros para endereçamento das variáveis de uma célula i, j , Figura 22.

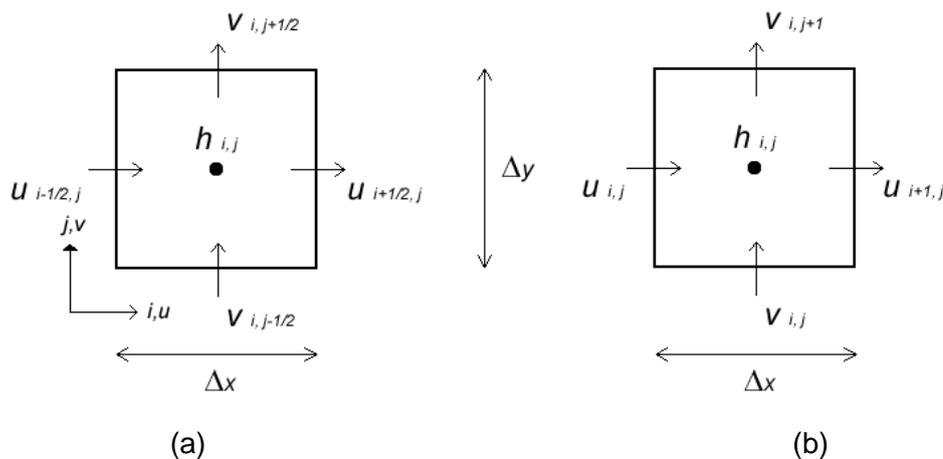


Figura 22 – Índices utilizados para endereçamento das variáveis em uma malha deslocada com coordenadas i, j . (a) fracionário (b) inteiro

Para se obter o valor de $\bar{u}_{i\pm 1/2,j}$ basta somar 1/2 ao índice i , e para se obter $\bar{v}_{i,j\pm 1/2}$ basta somar 1/2 ao índice j . Os índices para a altura h não necessitam de alteração. Isso significa dizer que no código fonte da implementação dessas equações, a velocidade \bar{u} no ponto $i + 1/2, j$ das equações apresentadas está armazenado no vetor de velocidades \bar{u} na posição i, j , como mostrado na Figura 22.

4.1.2. Estabilidade

A estabilidade do método numérico é garantida usando a condição de Courant-Friedrichs-Lewy, também conhecida como condição CFL. A condição CFL representa uma relação entre a malha, o passo de tempo e a velocidade de entrada do fluido e garante a estabilidade de solução de modelos explícitos. Essa condição declara que a onda numérica deve propagar-se pelo menos tão rapidamente quanto a onda física. Isso significa que a velocidade da onda numérica $\Delta x/\Delta t$ deve ser, pelo menos, tão rápida quanto a velocidade da onda física $|u|$, isto é, $\Delta x/\Delta t > |u|$, (Osher & Fedkiw, 2002). Assim, a condição CFL conduz a seguinte restrição no passo de tempo:

$$\Delta t < \frac{\Delta x}{\max\{|u|\}} \quad (76)$$

Onde o termo $\max\{|u|\}$ é dado pelo máximo valor da velocidade na malha. A eq. (76) geralmente é aplicada adotando-se um número para a condição CFL.

$$\Delta t \left(\frac{\max\{|u|\}}{\Delta x} \right) = \alpha \quad (77)$$

sendo $0 < \alpha < 1$. Segundo Osher & Fedkiw (2002) uma boa escolha é $\alpha = 0,9$ e uma escolha mais conservadora é $\alpha = 0,5$. A condição CFL multidimensional pode ser escrita da seguinte forma:

$$\Delta t \max \left\{ \frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} \right\} = \alpha \quad (78)$$

Ou ainda:

$$\Delta t \left(\frac{\max\{|\vec{V}|\}}{\min\{\Delta x, \Delta y, \Delta z\}} \right) = \alpha \quad (79)$$

4.1.3. Algoritmo de Solução do Problema

As etapas do processo de cálculo da solução do problema resume-se em:

1. Calcula-se dt de acordo com base nas condições de estabilidade, ou seja, $\Delta t = \alpha \min\{\Delta x, \Delta y, \Delta z\} / \max\{|\vec{V}|\}$.
2. Para cada face da célula da malha deslocada calculam-se os termos $\partial \bar{u} / \partial t$ e $\partial \bar{v} / \partial t$ de acordo com as eq. (57) a (66).
3. Atualiza-se os termos contendo a integração temporal nas equações de momento usando o método de Euler explícito de acordo como as eq. (74) e (75).

4. Para cada célula da malha deslocada calculam-se os termos $\partial(h\bar{u})/\partial t$ e $\partial(h\bar{v})/\partial t$ de acordo com as eq. (67) e (70).
5. Calcula-se a taxa de sedimentação $\partial s/\partial t$, para cada célula da malha, de acordo com o critério dado pela eq. (80).

$$\begin{aligned} \partial s/\partial t &= c_k(v_k - u_*) & u_* \leq v_k \\ \partial s/\partial t &= 0 & u_* > v_k \end{aligned} \quad (80)$$

A velocidade de queda de cada tipo de grão é dada pela equação de Stokes, eq. (81).

$$v_k = \frac{\Delta\rho g D^2}{0.018} \quad (81)$$

E a velocidade cisalhante u_* é dada por:

$$u_* = \frac{kV^2}{b \ln\left(\frac{hf}{z_0}\right)} \quad (82)$$

6. Atualiza-se os termos contendo a integração temporal da equação de continuidade utilizando o método de Euler explícito conforme eq. (73).

A Figura 23 mostra o algoritmo com as etapas de todo o processo de cálculo.

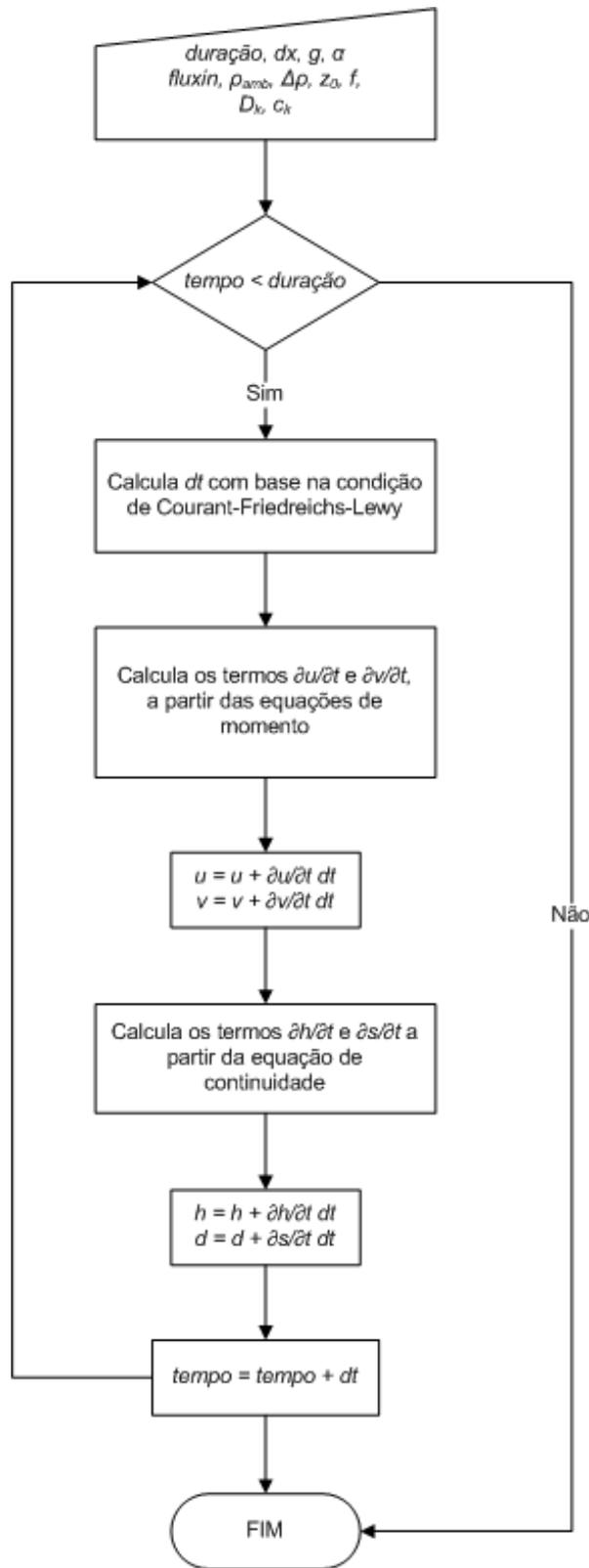


Figura 23 – Etapas do processo de solução do problema

4.2. Programa Para Simulação, Pré e Pós Processamento de Correntes de Turbidez.

Neste trabalho foi desenvolvido o programa *Turb3D*, Figura 25. Esse programa simula a evolução e o processo deposicional de correntes de turbidez de baixa densidade. Ele possui uma interface gráfica com usuário bastante amigável, que facilita todo o processo de análise, pré e pós-processamento dos resultados.

O aplicativo foi escrito usando os paradigmas da orientação a objetos com o uso da linguagem C++, sua interface gráfica foi implementada com a biblioteca de interfaces gráficas IUP (Levy, 1993) e com a biblioteca gráfica OpenGL (Woo *et al*, 2005).

Em programação orientada a objetos, um determinado problema é examinado por “entidades” independentes, chamadas objetos, que se relacionam com outras partes do problema. Tais entidades não são escolhidas por sua “computabilidade”, mas, sim, por terem uma relação física ou conceitual que as separa do resto do problema (Cox, 1991).

Na programação orientada a objetos, os dados e as funções que manipulam esses dados são agrupados em uma mesma entidade denominada *Classe*, isto simplifica bastante a implementação e manutenção do código. A *Classe* representa um conjunto de objetos com características afins. Uma classe define o comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos. Em outras palavras, o uso da classe permite agrupar os dados e funções de um determinado objeto de modo a obter uma implementação simplificada e uma maior reutilização do código. Reutilizar o código não significa evitar a duplicidade de código. Isso, as “funções” e “procedimentos” das linguagens procedurais já o faziam. Reutilização de código significa que o sistema poderá crescer com consistência e novos sistemas poderão se beneficiar de objetos criados anteriormente, aumentando ainda mais a eficiência e diminuindo custos. O enfoque da orientação a objetos pode ser aplicado tanto a análise quanto à programação e essa é uma das grandes vantagens dessa tecnologia (Alencar, 2002).

O processo de desenvolvimento de um sistema computacional pode, numa visão genérica, ser estruturado em três fases distintas que correspondem ao seu ciclo de vida. Essas fases são chamadas de fase de definição ou concepção do produto, fase de desenvolvimento e fase de manutenção.

Na fase de definição identifica-se o problema, ou seja, que informação deve ser processada, que funções e desempenho são pretendidos, que interfaces são necessária, que restrições devem ser consideradas e que critérios devem ser utilizados na avaliação do projeto. Na fase de desenvolvimento identifica-se a solução, como é que as estruturas de dados, arquitetura do produto e funções serão realizadas e como tudo isso será implementado em uma linguagem de programação. A fase de manutenção foca nas alterações do produto devido a erros não detectados nas fases anteriores ou nas alterações propostas pelos usuários do sistema.

Essa metodologia de desenvolvimento de software descrita acima foi adotada na elaboração do sistema computacional desenvolvido neste trabalho. A Figura 24 apresenta o diagrama de classe da Interface de Programação de Aplicativos, API, do inglês *Application Programming Interface*, para a biblioteca que implementa a simulação de correntes de turbidez elaborada na fase de desenvolvimento do produto. A API é um conjunto de rotinas e padrões estabelecidos por um *software* para a utilização das suas funcionalidades por programas aplicativos que não querem envolver-se em detalhes da implementação do *software*, mas apenas usar seus serviços (Wikipédia).

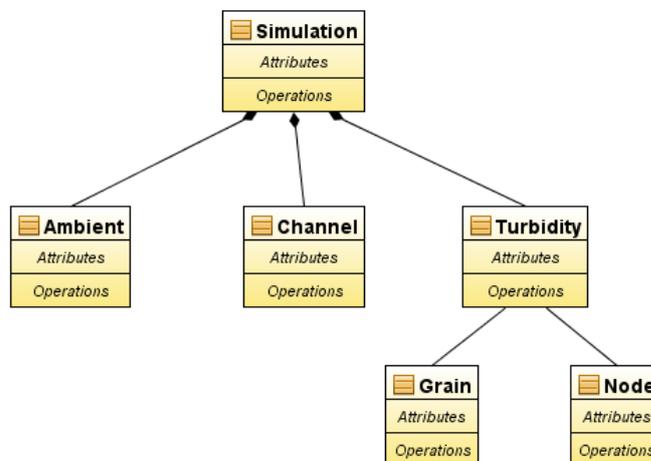


Figura 24 – Diagrama de classes da API do Turb3D

A classe *Node* encapsula todos os dados e métodos referentes às coordenadas x, y, z da malha e a classe *Grain* se encarrega de gerenciar as informações referentes ao diâmetro e a concentração de cada tipo de sedimento presente na corrente.

A classe *Turbidity*: contém os atributos e métodos necessários para manipulação de dados como as velocidades, altura da corrente, altura dos depósitos e demais parâmetros associados à corrente. As classes *Ambient* e

Channel são classes que guardam os dados e métodos referentes ao meio circundante e ao canal de entrada da corrente no meio circundante, respectivamente.

Por fim, tem-se a classe *Simulation* que possui as classes *Turbidity*, *Channel* e *Ambient* como atributos, ou seja, instâncias dessas classes são criadas e associadas a classe *Simulation* para que esta se encarregue de gerenciar todos os dados da simulação, bem como realizar todo o processo de cálculo da simulação.

Além da API, que fornece os métodos necessários para a simulação numérica, tem-se uma interface gráfica para pré-processamento e pós-processamento, Figura 25. Essa interface é responsável por coletar os dados de entrada, processar todas as informações por meio de chamadas a API de cálculo e pela posterior visualização dos resultados.

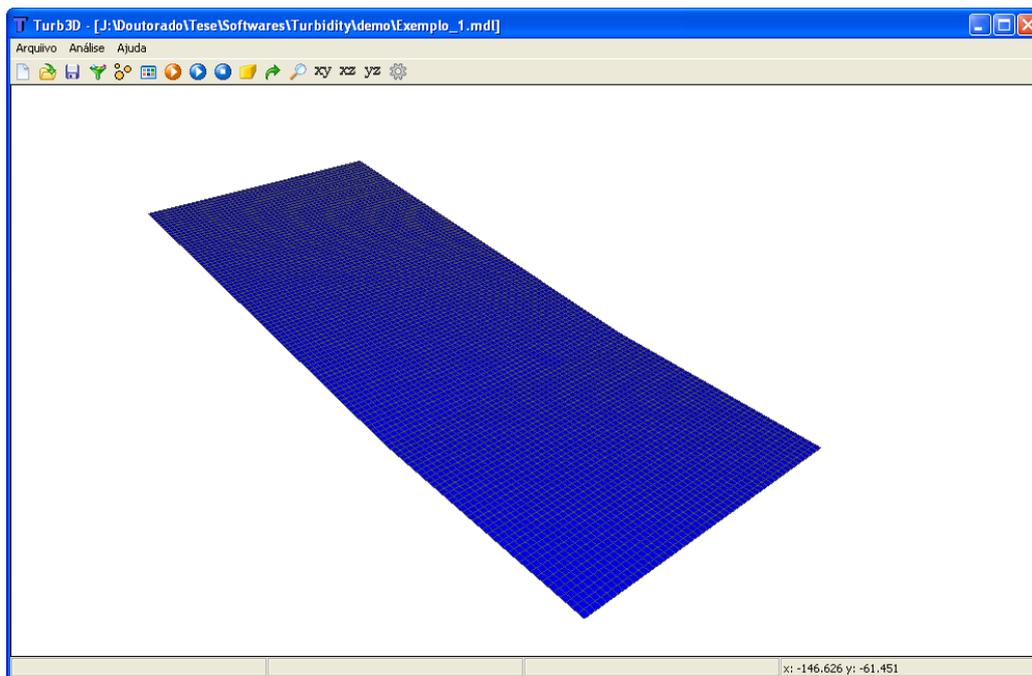


Figura 25 – Interface gráfica do programa *Turb3D*

A partir desse diálogo pode-se dar início a criação de um novo modelo de corrente de turbidez. Inicialmente deve-se fornecer os dados da superfície inicial por onde a corrente irá escoar. Essa superfície deve ser representada por uma malha uniforme e regular. A malha é então fornecida como dado de entrada ao programa através de um arquivo de texto no formato x, y, z , Figura 26. Os valores da abertura, comprimento e largura da malha são calculados automaticamente a partir dos dados contidos no arquivo texto.

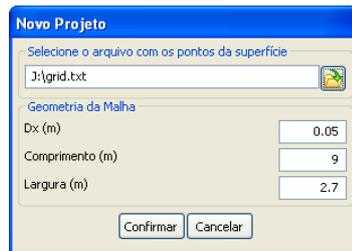


Figura 26 – Diálogo de entrada da malha da superfície

A seguir especificam-se os parâmetros referentes ao canal de entrada do fluxo no fluido ambiente, Figura 27. Nesse diálogo deve-se fornecer as coordenadas do vértice inferior esquerdo e do vértice superior direito do retângulo correspondente ao canal na malha, e a altura de fluxo na saída do canal.

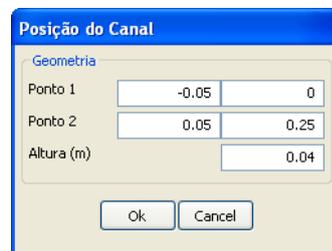


Figura 27 – Diálogo de dados do canal

A Figura 28 apresenta o diálogo responsável pela coleta dos dados referentes à simulação, ao fluxo e ao fluido ambiente. A seguir descreve-se em detalhe cada campo do diálogo.

<i>Tipo</i>	Tipo de iniciação do fluxo, que pode ser contínuo ou pulso.
<i>Duração</i>	Duração em que o canal de entrada do fluxo permanece alimentando a simulação, no caso de fluxo contínuo.
<i>Fator de suavização</i>	Fator usado para garantir a estabilidade numérica da análise.
<i>CFL constante</i>	Constante de Courant-Friedrichs-Lewy.
<i>Aceleração da gravidade</i>	Valor da aceleração devido à gravidade.
<i>Vazão</i>	Vazão do fluxo de entrada.
<i>Densidade do ambiente</i>	Densidade do fluido ambiente ou circundante.
<i>Diferença de densidade</i>	Diferença entre a densidade do sedimento e do fluido.
<i>Rugosidade</i>	Rugosidade da superfície.

Altura fracionária da camada limite Razão entre a camada limite turbulenta e a altura total do fluxo.

Figura 28 – Diálogo de dados da simulação

O diálogo da Figura 29 captura todos os dados de entrada referentes aos sedimentos. É através dele que se deve informar ao programa o diâmetro dos grãos e suas respectivas concentrações dentro da corrente. Pode-se ainda visualizar no gráfico, lado direito do diálogo, uma curva granulométrica formada pelos tipos de grãos inseridos. O programa foi escrito para simular correntes de turbidez com mais de um tamanho de grão. Contudo, como os ensaios físicos realizados para aferição do modelo proposto só contemplam um tipo de sedimento, não serão usados mais do que um grão nesse diálogo.

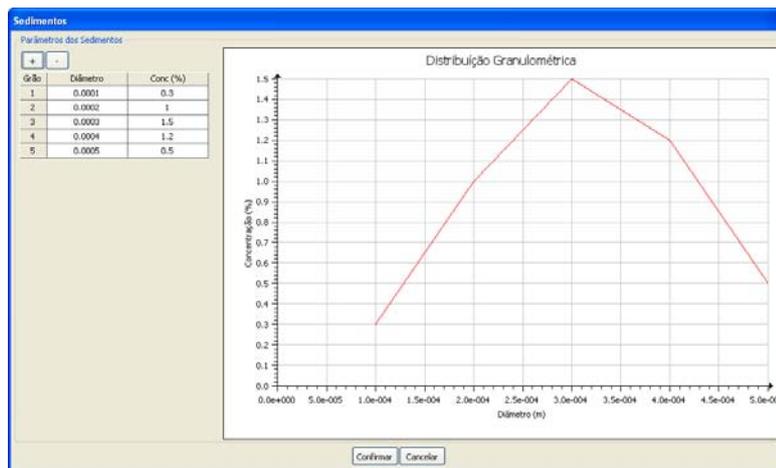


Figura 29 – Diálogo de sedimentos

Com todos os parâmetros definidos pode-se então proceder a análise do modelo. Para a realização da simulação numérica o programa inicialmente faz chamadas a métodos disponibilizados pela API de cálculo, recebe os resultados da API contendo a solução do algoritmo, e os exibe na tela em cada passo de

tempo até que o tempo de simulação seja alcançado. Com isso, é possível visualizar a evolução da corrente e o depósito resultante a cada passo de tempo.