

4

Sistema Computacional

Este trabalho está inserido na linha de Pesquisa “Computação Gráfica Aplicada” do Departamento de Engenharia Civil da PUC-Rio e do laboratório Tecgraf (Grupo de Tecnologia em Computação Gráfica) também localizado na PUC-Rio. Nesta linha são desenvolvidos simuladores computacionais de diversas áreas, como sistemas para engenharia civil, engenharia mecânica e geologia. Nesta última área destacam-se dois simuladores. O primeiro é o simulador chamado RECON, que é uma ferramenta computacional utilizada para reconstruir seções geológicas. Maiores detalhes podem ser vistos no trabalho de Santi (2002). O outro é o simulador chamado STENO que foi desenvolvido neste trabalho.

Este capítulo descreve o protótipo desenvolvido, mostrando a estratégia de programação utilizada para desenvolvê-lo, suas principais funcionalidades e o algoritmo implementado para visualizar superfícies que correspondem a horizontes geológicos de uma idade definida pelo usuário.

4.1

Sistema STENO

Para obter uma estrutura de dados consistente, de fácil modificação e manutenção, optou-se em utilizar, no desenvolvimento do STENO, uma linguagem orientada a objetos (no caso, a linguagem C++). 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). Logo, o objetivo principal no uso dessa filosofia de programação é ter correspondência, uma a uma, entre entidades no problema físico e objetos no programa.

A linguagem orientada a objetos possui um tipo de dado, chamado de *classe*, a partir das quais são criados objetos. O uso de *classes* 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 é simplesmente evitar duplicidade de código. Isso, as “funções” e “procedimentos” já faziam (Sebesta, 2000). Reutilização do 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 de orientação a objetos pode ser aplicado tanto à análise quanto à programação e essa é uma das grandes vantagens desta tecnologia.

No desenvolvimento de um sistema computacional existem duas fases bem claras: o projeto e o desenvolvimento. A fase de projeto de um sistema é fundamental, pois é nela que se estrutura toda a sua organização, isto é, a arquitetura da informação é construída. O desenvolvimento envolve a especificação e a implementação da representação de dados do programa. Em programação orientada a objetos, são três etapas a serem vencidas (Fenves, 1990):

- Seleção de classes: Determinam-se as classes de objetos necessários para representar o problema e sua solução; criam-se sub-classes para aumentar o grau de especificidade de representações do problema.
- Especificação das classes: Definem-se as operações nos objetos de uma classe ao se especificar o que faz cada mensagem, ou seu método associado; a especificação contém uma descrição precisa da operação invocada por cada mensagem.
- Implementação das classes: Seleccionam-se variáveis locais para objetos; programam-se um método para executar a operação especificada.

No STENO, toda a estrutura de dados responsável pelo armazenamento da região a ser modelada, ou seja, o *grid* e dos dados criados durante a simulação, foi implementada seguindo a filosofia descrita acima. A estrutura global é definida por quatro *classes* como mostra a figura 4.1.

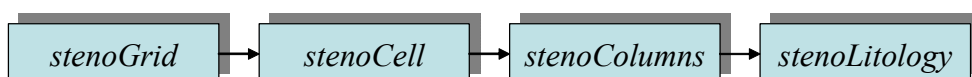


Figura 4.1 – Estrutura global das classes implementadas no STENO.

Inicialmente foi criada uma *classe*, chamada *stenoGrid*, para gerenciar todos os dados referentes ao *grid* (dimensões, números de células, resolução, etc.). Na figura 4.2 é mostradas a estrutura interna da *classe* *grid* e as ligações entre as principais *classes*.

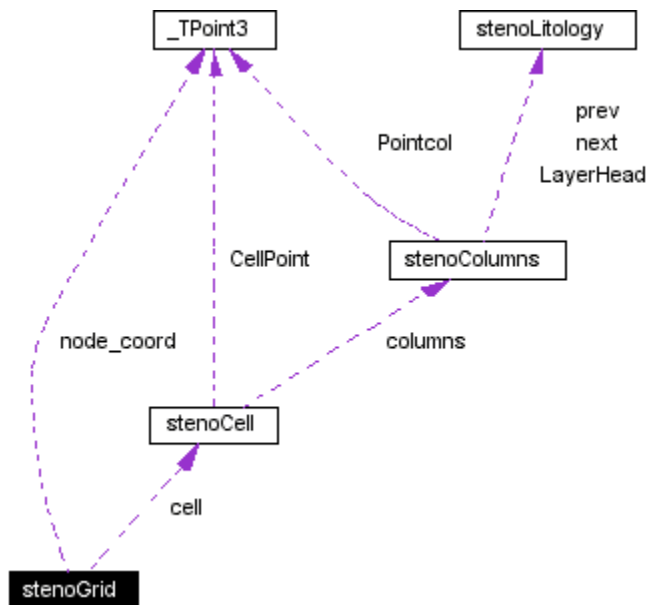


Figura 4.2 – Estrutura interna da classe *stenoGrid*.

Durante a simulação cada célula do *grid* tem um comportamento isolado, isto é, a cada passo as células terão alturas diferentes. Essas alturas corresponderão às camadas de sedimentos já depositados. Desse modo, foi implementada a classe *stenoCell* (figura 4.3), que é responsável por armazenar as características de cada célula do *grid* durante a simulação.

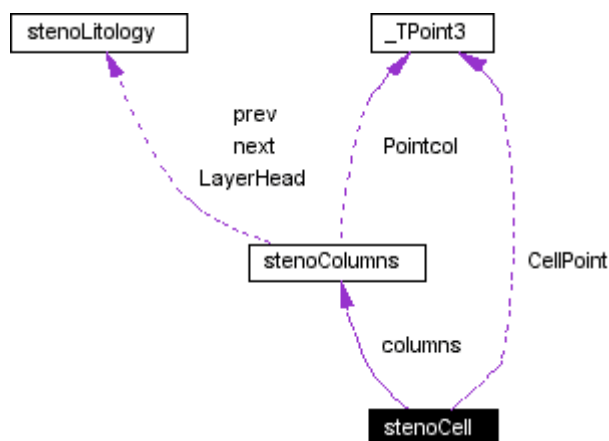


Figura 4.3 – Estrutura interna da classe *StenoCell*.

Como durante o processo de sedimentação as células podem ter diversos tipos de litologias, foi implementada a *classe* *stenoColumns* (figura 4.4). Essa

classe contém basicamente uma lista duplamente encadeada onde cada item da lista possui informações de cada litologia (idade, porosidade, decaimento, etc). Essas informações estão armazenadas dentro de uma *classe* chamada tipo *stenoLitology*.

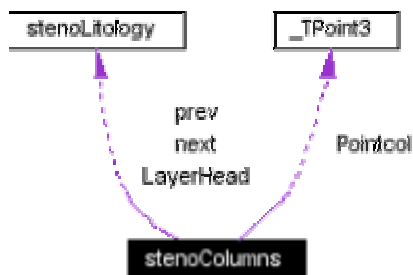


Figura 4.4 – Estrutura interna da classe *StenoColumns*.

O STENO foi desenvolvido utilizando a linguagem C++, o sistema de interface IUP (Levy, 1994) e a biblioteca gráfica *OpenGL* (Woo & Nieder, 1997). A aplicação é multiplataforma e a interface gráfica desenvolvida busca ser amigável para o usuário no que diz respeito à entrada de dados e visualização dos resultados. A figura 4.5 mostra o diálogo principal do STENO com uma superfície inicial e o nível do mar.

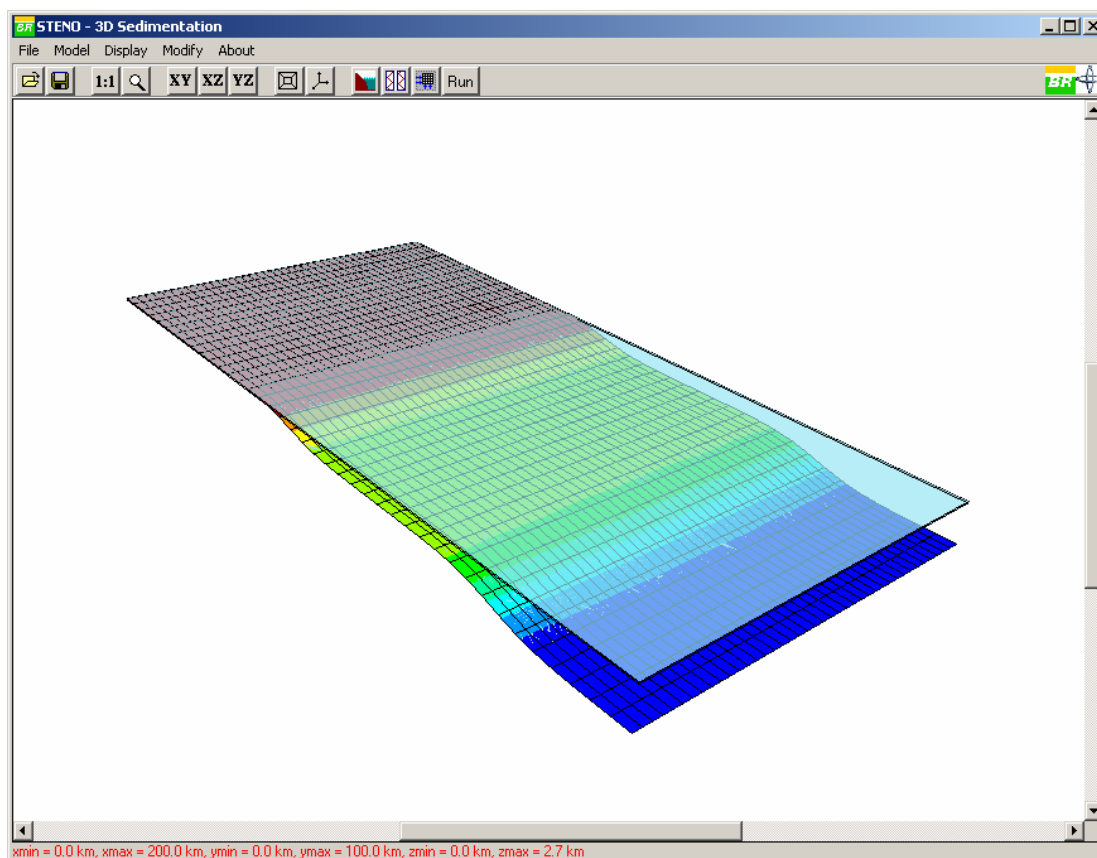


Figura 4.5: Diálogo Principal do STENO.

É a partir deste diálogo que o usuário inicia a criação do modelo, definindo os parâmetros de entrada para fazer a simulação. Primeiramente o usuário deve definir uma superfície inicial. Esta superfície deve estar armazenada em um arquivo texto no formato (x,y,z) ou deve ser gerada utilizando o método descrito na seção 3.2 do capítulo 3 (figura 4.6). Depois, o usuário deve especificar os seguintes parâmetros:

- (a) Tempo da simulação: tempo inicial, tempo final e passo da simulação.
Todos os valores de tempo são especificados em Milhões de ano (Ma);
- (b) Valores de referência do nível do mar e do nível de base das ondas. No STENO esses valores são especificados em km;
- (c) Coordenadas dos pontos onde se conhece a variação da subsidência;
- (d) Curva eustática que será utilizada na simulação.
- (e) Volume de aporte de sedimentos, por fração litológica ao longo da linha de costa; e
- (f) Variação da velocidade de corrente com a batimetria.

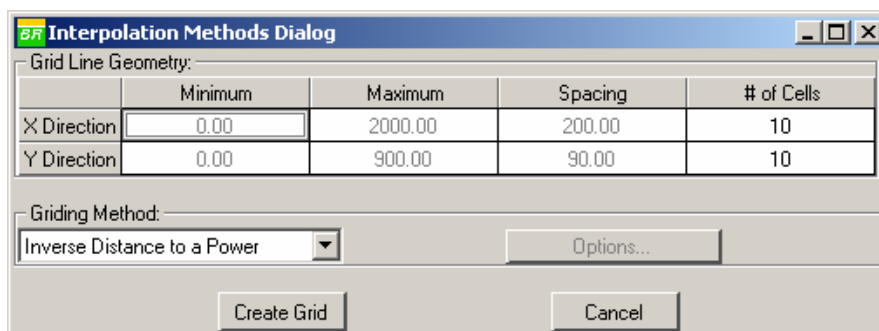


Figura 4.6: Diálogo utilizado para definir a superfície inicial (embasamento).

Os parâmetros dos itens (a) e (b) são especificados utilizando a interface mostrada abaixo (figura 4.7).

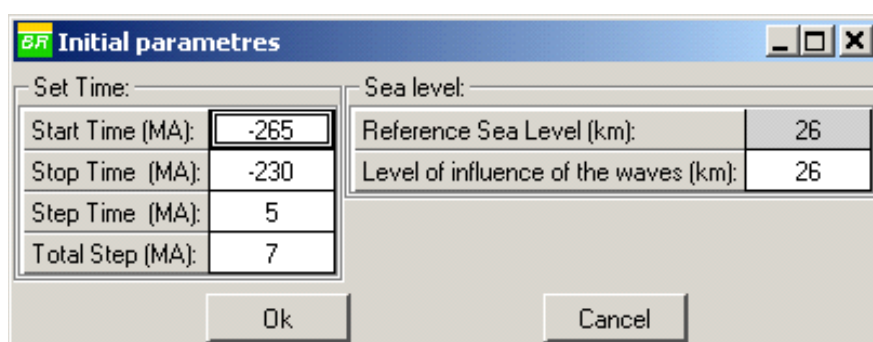


Figura 4.7: Diálogo onde o usuário especifica os parâmetros iniciais.

Os pontos onde se conhece a variação da subsidência, item (c), são definidos utilizando o diálogo da figura 4.8.

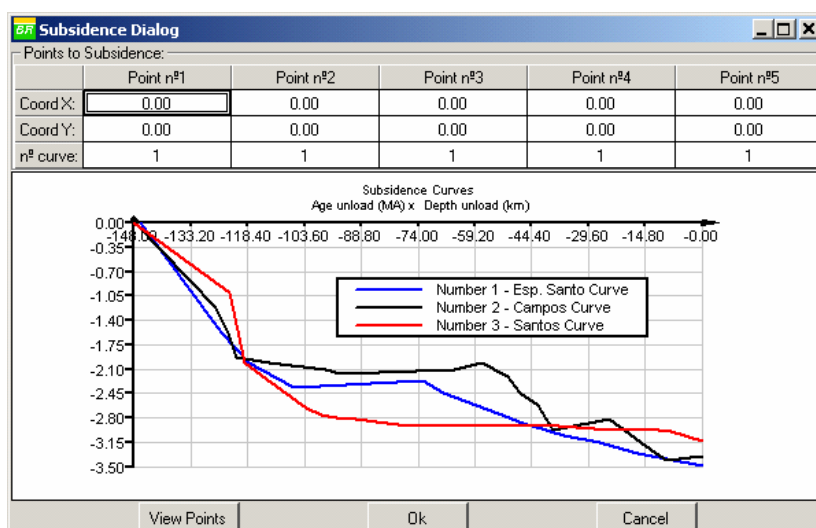


Figura 4.8: Diálogo utilizado para definir a variação da subsidência.

A curva eustática é definida pelo usuário utilizando o diálogo mostrado na figura 4.9. Até o momento, o sistema permite apenas fazer simulações utilizando a curva de baixas frequências de Haq que pode ser vista na figura 4.9. Porém o

STENO foi estruturado de forma a permitir a inclusão de novas curvas de baixas frequências e compor curvas de altas frequências.

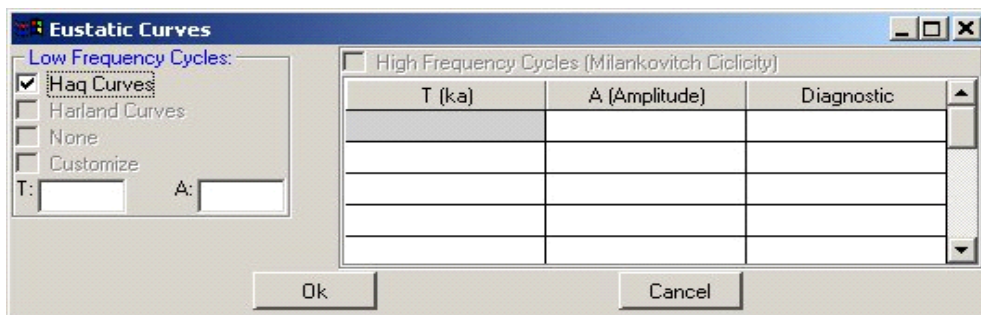


Figura 4.9: Diálogo onde o usuário define a curva eustática.

Na figura 4.10 também podem ser visualizados o tempo inicial e final e qual passo da simulação está sendo analisado.

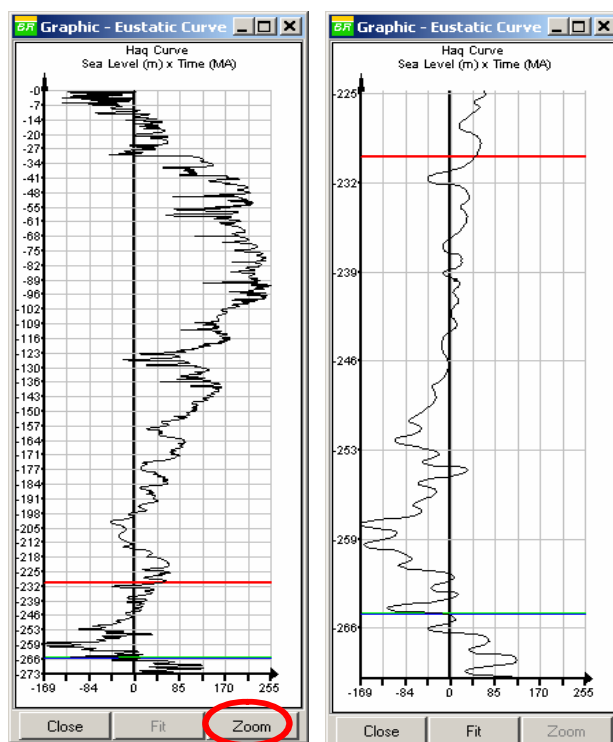


Figura 4.10 Curva de Haq, mostrando o tempo inicial (linha verde) o tempo final (linha vermelha) e o tempo atual da simulação (linha azul). A figura da esquerda mostra de forma detalhada o trecho da curva Haq que será utilizado na simulação.

O aporte de sedimentos, item (e), é definido através de dois diálogos. O primeiro diálogo (figura 4.11) é utilizado para definir em que pontos são conhecidos o volume de sedimentos por frações litológicas. Esses pontos são utilizados para interpolar o aporte de sedimentos ao longo da linha de costa. É nele também que o usuário define os volumes de sedimentos. Mas para isso,

existe um banco de dados com informações sobre as descargas volumétricas, máximas e mínimas, por frações litológicas de alguns rios significativos de descarga média e grande (figura 4.12). Essa tabela é acessada através de um botão como mostra a figura 4.11. O usuário deve escolher os dados utilizando o banco de dados ou digitar as informações diretamente na tabela do diálogo da figura 4.11.

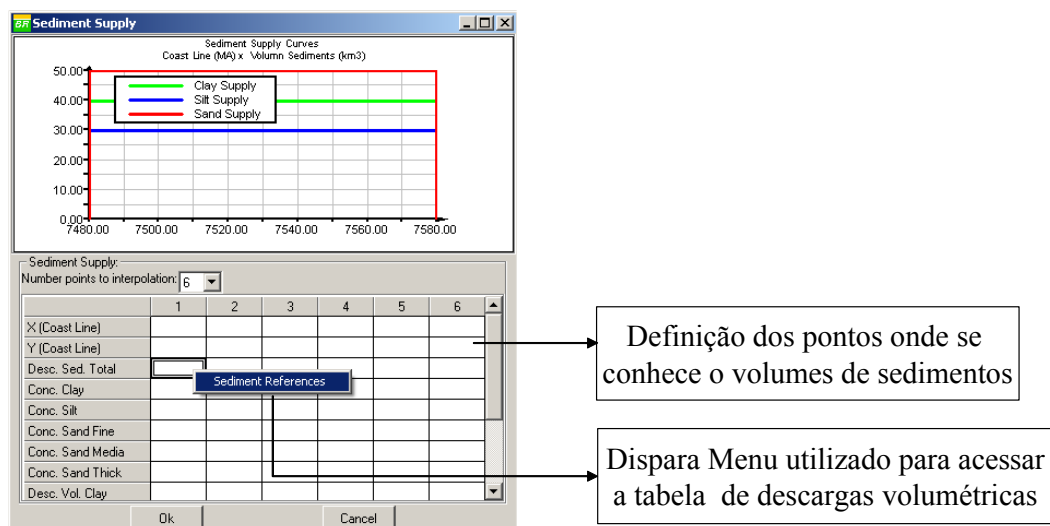


Figura 4.11: Tabela onde o usuário define o aporte de sedimentos.

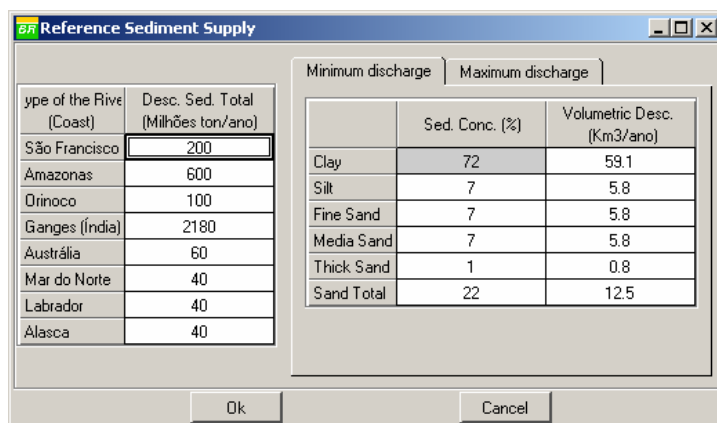


Figura 4.12: Tabela que contém as informações sobre as descargas volumétricas, máximas e mínimas, de alguns rios.

Finalmente, o usuário deve indicar como a velocidade de corrente varia com a batimetria, item (f). O usuário deverá escolher a fisiografia da borda continental:

- Tipo 1 - Plataforma / Talude / Bacia (figura 4.13); e
- Tipo 2 – Rampa (figura 4.14).

Com a fisiografia da borda continental definida, o usuário deverá escolher a velocidade de corrente para cada sub-ambiente da borda continental. Para isso, o usuário dispõe de uma tabela com informações sobre as velocidades de corrente marítimas (figura 4.15). Os valores dessa tabela foram coletados equipe do setor de Geologia do Cenpes (Centro de Pesquisa da PETROBRAS). Com isso, cada um dos sub-ambientes estará correlacionado com a batimetria, que por sua vez estará evoluindo com a variação do nível do mar ao longo da simulação.

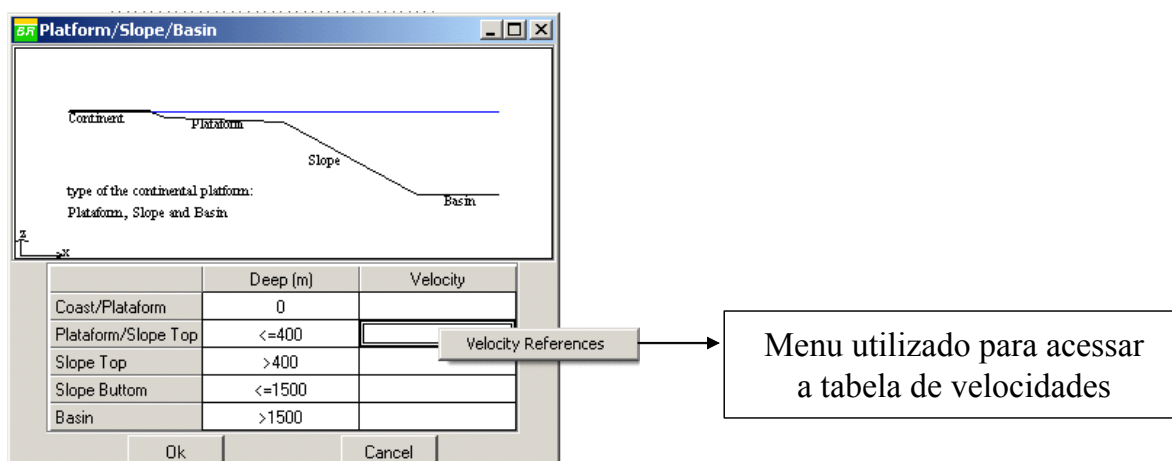


Figura 4.13: Borda continental do tipo 1.

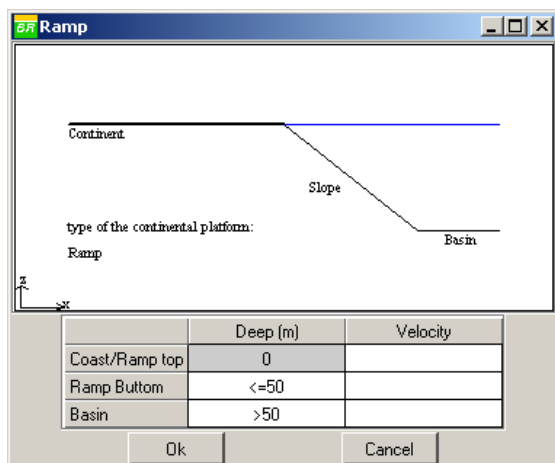


Figura 4.14: Borda continental do tipo 2.



Mean Longitude	Mean Latitude	Mean u Velocity	Mean v Velocity	Mean Velocity	Nº of 5 days Aver.
22.925	-56.483	10.337	3.572	5.052	13.000
28.301	-52.142	13.760	2.431	3.438	47.000
37.299	-46.813	15.862	1.990	2.815	48.000
35.894	-43.515	10.663	0.477	0.675	28.000
21.907	-36.898	11.333	5.157	7.292	43.000
21.246	-31.867	1.178	3.891	5.503	88.000
14.309	-28.485	-2.163	1.070	1.513	53.000
3.539	-15.907	-17.102	0.347	0.490	5.000
4.591	-11.094	-0.924	-7.358	10.406	6.000
50.437	64.604	7.010	2.246	3.176	12.000
2.660	65.359	0.533	3.826	5.411	8.000
6.288	-56.048	16.089	9.671	13.677	9.000
6.880	-51.647	12.436	0.571	0.808	38.000
7.085	-48.169	15.468	-1.908	2.698	40.000
8.086	-43.159	7.692	4.289	6.066	70.000
7.215	-36.723	5.541	3.536	5.001	47.000

Figura 4.15: Tabela com as velocidades de correntes marítimas.

Com os dados descritos da forma definida acima, a simulação pode ser iniciada. Durante cada passo, o usuário poderá, dentre outras opções, consultar informações sobre a malha (batimetria, resolução, etc.), visualizar o campo de velocidades e as linhas de correntes, visualizar o modelo de diversos ângulos e posições, alterar o aporte sedimentar e velocidades de corrente e visualizar um horizonte geológico, isto é, uma superfície do modelo com uma idade definida. A seguir será mostrado o algoritmo utilizado para visualizar essas superfícies.

4.2

Extração das superfícies

Em muitas ocasiões o geólogo precisará visualizar a estrutura tridimensional do topo das colunas que foram sedimentadas (figura 4.16). Isso é feito especificando a idade (Ma) de um dos passos da simulação. Essa visualização é importante, para que o geólogo possa avaliar melhor o resultados obtidos na simulação.

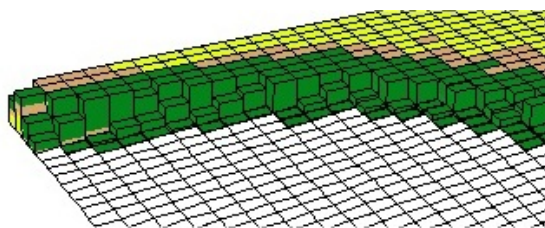


Figura 4.16: Sedimentos depositados no *grid* armazenados em colunas.

As duas abordagens básicas para a solução do problema de visualização volumétrica são a extração de superfícies (*surface fitting*) e o *rendering* direto de volumes (*volume rendering*). Elas diferem basicamente pela utilização ou não de representações intermediárias dos dados volumétricos para a geração da visualização adequada à aplicação (Costa, 1994). Enquanto no *rendering* direto de volumes a projeção é realizada diretamente a partir dos dados volumétricos, na extração de superfícies os dados volumétricos são convertidos para uma representação geométrica (polígonos). Esta técnica foi utilizada para visualizar a superfície de uma idade definida pelo usuário.

O algoritmo utilizado foi desenvolvido por Costa (1994). Neste algoritmo existem duas fases bem definidas. Primeiro o usuário define uma idade e todas as células que contenha pontos nesta idade são encontradas. Depois as células encontradas são examinadas e um conjunto de polígonos conectados é calculado e armazenado para ser visualizado posteriormente. Para construção das superfícies no STENO é necessário primeiramente, procurar em cada coluna o topo da camada cuja idade foi especificada pelo usuário. As colunas são montadas de tal forma que é possível, através de uma busca, obter a altura da coluna com idade especificada pelo usuário. A figura 4.17 mostra um exemplo esquemático de como as colunas estão armazenadas. Elas foram armazenadas utilizando uma estrutura de dados (seção 4.1) que facilitasse a busca do topo de uma idade definida pelo usuário.

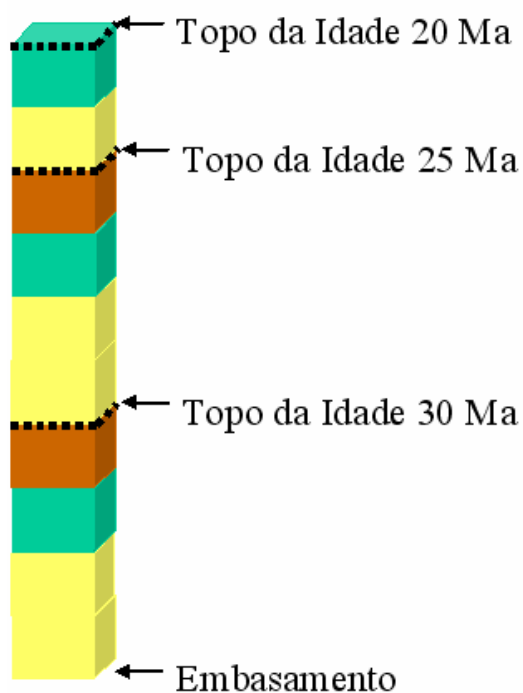
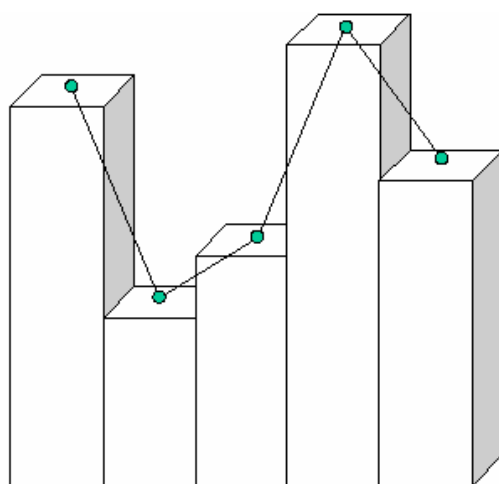


Figura 4.17: Exemplo esquemático mostrando como as colunas são montadas.



● Topo da coluna com idade definida pelo usuário

Figura 4.18: Topo das colunas após a escolha da idade pelo usuário.

Desse modo, como mostra a figura 4.18, cada coluna da malha terá altura diferente para a mesma idade. Em planta pode-se com os centros de cada célula montar uma malha interna à malha da simulação. Esta malha se chama malha de elevação da superfície e pode ser vista em planta de forma esquemática na figura 4.19.

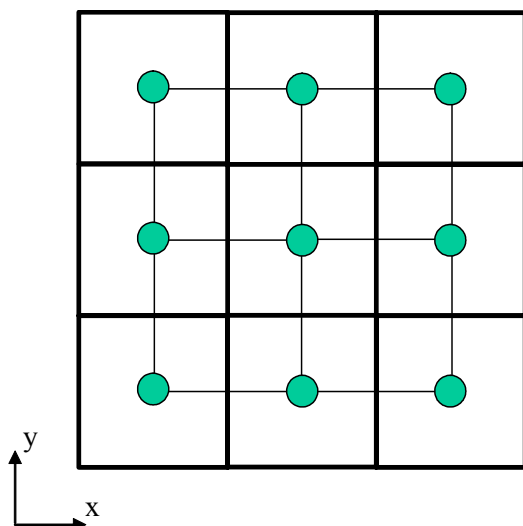


Figura 4.19: Malha de elevação da superfície de idade definida pelo usuário.

O próximo passo é estender a malha de elevação das superfícies até as bordas da malha da simulação. Isto é feito mantendo as mesmas cotas dos pontos da malha de elevação (figura 4.20).

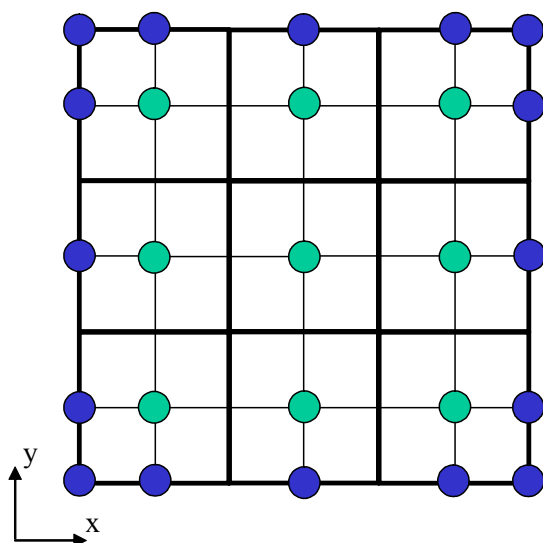


Figura 4.20: Extensão da malha de elevação da superfície até as bordas da malha de simulação.

Depois, deve-se calcular o valor da altura nas interseções das células da simulação. Com os pontos $p1$ e $p2$, deve-se calcular a coordenada z do ponto p (figura 4.21 e 4.22).

Onde a e b são as distâncias em x dos pontos $p1$ e $p2$ até p . No caso do STENO, a e b são sempre iguais ao espaçamento do *grid* (malha da simulação). O cálculo é feito da mesma forma para a direção y .

Por último, deve-se calcular o valor da altura nos vértices da malha da simulação, ou seja, vértices do *grid*. (figura 4.23 e 4.24).

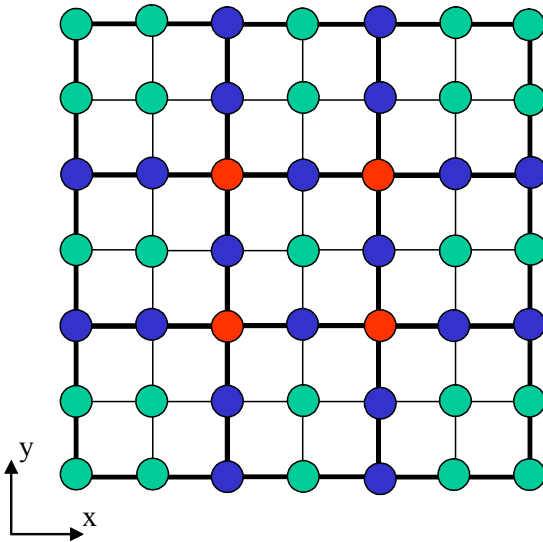


Figura 4.23: Pontos calculados após o cálculo dos valores da altura nos vértices da malha da simulação.

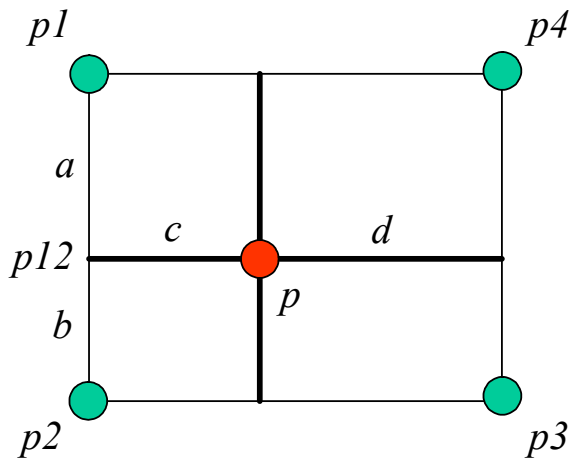


Figura 4.24: Pontos calculados após o cálculo dos valores da altura nos vértices da malha da simulação.

Na figura 4.24 a coordenada z do ponto p é aproximada por:

$$p_z = p1z \times \frac{d}{c+d} + p3z \times \frac{c}{c+d} \quad (4.2)$$

onde

$$p12_1 = p1_z \times \frac{b}{a+b} + p2_z \times \frac{a}{a+b} \quad (4.3)$$

$$p34_z = p4_z \times \frac{b}{a+b} + p3_z \times \frac{a}{a+b} \quad (4.4)$$

Com os pontos mostrados na figura 4.23, tem-se uma superfície suavizada da idade definida pelo usuário. As figuras 4.25 e 4.27 mostram a visualização tridimensional não suavizada de uma idade da simulação definida pelo usuário, ou seja, as células do *grid* em forma de colunas e as figuras 4.26 e 4.28 mostram respectivamente as superfícies 4.25 e 4.27 suavizadas pelo algoritmo mostrado acima.

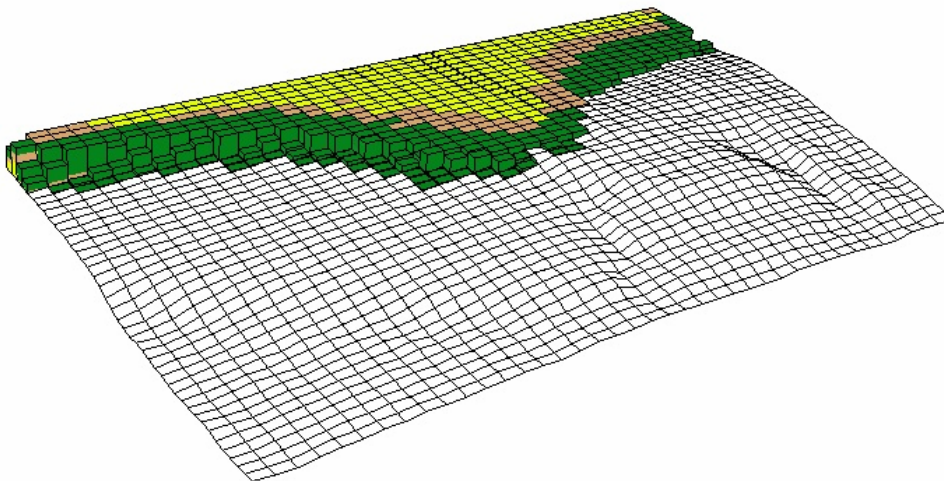


Figura 4.25: Visualização tridimensional, com as células do grid em forma de colunas, de uma idade definida pelo usuário.