

5 Modelo computacional

Por estar inserido na linha de pesquisa de *Computação Gráfica Aplicada* do Departamento de Engenharia Civil da PUC-Rio, parte extensa desse trabalho foi a elaboração de um modelo computacional completo para simulação de sedimentação carbonática. Um programa, totalmente desenvolvido no ambiente MATLAB (The Mathworks Inc., 2008), junta uma interface ao usuário intuitiva a técnicas de computação gráfica interativa. O CarbSM (*CARBonate Sedimentation Model*) efetua o pré e pós processamento junto aos algoritmos de deposição carbonática.

5.1. Sistema CarbSM

O MATLAB é um ambiente de programação muito utilizado no meio científico por ser prático de programar, apresentar uma estrutura interna preparada pra resolver sistemas de equações e por oferecer muitas opções de saídas gráficas. Sua estrutura de dados básica é uma matriz flexível a vários tipos de dados e que não necessita ser inicializada. Todas as operações trabalham em cima dessas matrizes e por isso o sistema todo é muito otimizado para lidar com matrizes. Portanto, o MATLAB apresenta um forte potencial para atuar na área científica que muitas vezes precisa de um sistema computacional para resolver grandes sistemas de equações que são armazenadas em forma matricial. CarbSM foi todo escrito seguindo conceitos de POO (Programação Orientada a Objetos) para conseguir uma estrutura de dados consistente, de fácil manutenção e melhor organização. Os recursos de POO para MATLAB não oferecem tanta flexibilidade como em outras linguagens orientadas a objetos, como C++ ou Java, mas são suficientes para criar um programa bem estruturado e consistente que sirva para o que esse trabalho se propõe a fazer.

Classes são criadas para juntar propriedades e métodos (funções) que podem ser aplicados em diversos objetos. Cada objeto tem suas propriedades que podem

ser modificadas e acessadas através de seus métodos. Em CarbSM foram criadas as seguintes classes:

- CarbonateSurface: guarda as matrizes com as coordenadas x , y , z e utiliza métodos para calcular a evolução temporal da superfície carbonática
- Restrictions: tem como propriedades os parâmetros necessários para o cálculo das funções de restrições e as matrizes contendo os valores das restrições para toda a superfície
- Carbonates: guarda os valores das taxas máximas de cada ambiente de deposição carbonática e as matrizes com as taxas resultantes para toda a superfície após aplicação das restrições
- CarbonateFile: classe que gerencia a escrita e leitura dos modelos gerados no CarbSM
- CarbonateModel: classe geral que inicializa os objetos das outras classes e gerencia a inicialização dos parâmetros e todo o processo da modelagem em si
- CarbonateVis: classe que faz a conexão entre a estrutura de dados do modelo e os elementos da interface gráfica.

5.2. Rotinas computacionais

Dentro dessas classes estão todas as rotinas que efetuam a modelagem computacional dos depósitos carbonáticos, referentes aos processos geológicos descritos no capítulo anterior. Essa seção dará uma explicação sobre as principais rotinas do modelo geológico e seus algoritmos empregados.

5.2.1. Superfície Inicial

O primeiro passo da simulação consiste em definir a superfície inicial em que os algoritmos de deposição serão aplicados. Para isso, é necessário um conjunto de dados anteriormente adquiridos chamados *pontos de controle*. Pontos de controle são conjuntos de pontos no formato (x, y, z) a partir dos quais se define a área que pretende ser modelada (Figura 5.1.). Não necessariamente esses

conjuntos serão os limites da superfície inicial. Os pontos de controle são somente os pontos que são utilizados para determinar uma representação matemática da superfície inicial. Essa representação matemática é obtida aplicando-se um método de interpolação nos pontos de controle. Estes pontos são geralmente obtidos no campo para casos de modelagens reais e quanto mais pontos de controle, mais próximo da realidade será a superfície resultante, necessitando de no mínimo um ponto para poder gerar a superfície. Porém este mínimo não apresenta um resultado muito convincente.

A superfície inicial obtida é um *grid* discreto bidimensional com espaçamento regular nas direções x e y . Em cada nó do *grid* será calculado o valor de z adequado a partir da interpolação dos pontos de controle. Esse trabalho utiliza o Método do Inverso da Potência das Distâncias (IPD) (Landim, 2000), um método de interpolação local, determinístico e exato, bastante utilizado em Sistemas de Informação Geográfica. O método consiste em fazer uma média ponderada dos valores de z dos pontos de controle para os nós do *grid*, sendo que os pontos mais distantes têm peso menor. Esse peso é determinado utilizando o inverso da potência da distância, no plano XY , do ponto de controle para o nó do *grid*. Os valores de z para pontos do *grid* que coincidem com algum ponto de controle não será interpolado e o valor exato de z do ponto de controle coincidente será utilizado no *grid*.

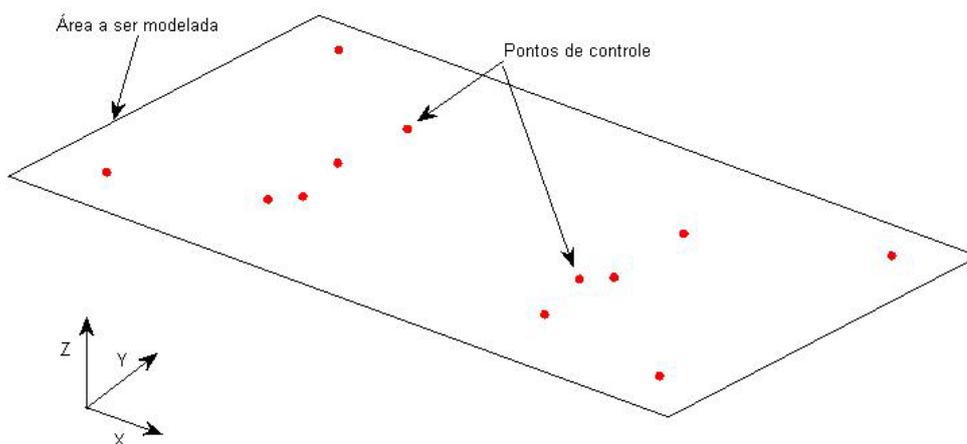


Figura 5.1. Pontos de controle predeterminados e área a ser modelada.

Para cada nó do *grid* a ser interpolado, o valor de z é calculado com as seguintes equações:

$$z(x, y) = \frac{\sum_{i=1}^n \frac{z_i}{h_{ij}^\beta}}{\sum_{i=1}^n \frac{1}{h_{ij}^\beta}} \quad (5.1)$$

$$h_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5.2.)$$

Sendo:

$z(x, y)$ - o valor de z interpolado no nó do *grid* (x, y) ;

z_i - o valor de z conhecido de um ponto de controle i ;

n - o número de pontos de controle disponíveis;

h_{ij} - distância entre o nó do *grid* j e o ponto de controle i , calculado pela Eq.

5.1;

β - expoente da potência da distância.

Pode acontecer de alguns pontos de controle terem a mesma posição em xy que alguns nós do *grid*, para isto pode-se usar a seguinte condição para garantir o mesmo valor de z para esses casos:

$$z(x, y) = z_i \quad (5.3)$$

O valor do expoente da potência da distância é bastante importante e influencia no resultado final da interpolação. Quanto mais alto o valor do expoente, mais peso se dá aos pontos de controle mais próximos e menos peso aos pontos mais distantes. Um expoente mais baixo gera uma superfície mais suave e um expoente mais alto gere uma superfície com maiores variações de z . Para demonstrar essa diferença, a área referente à Figura 5.1. foi subdividido para obter um *grid* com 40 nós nas direção x e y . O Método do Inverso da Potência das Distâncias foi aplicado inicialmente com $\beta = 1$ (Figura 5.2.) e depois com $\beta = 4$ (Figura 5.3.). No segundo caso os extremos tem maior magnitude e os pontos de controle mais afastados têm menor influência nos resultados, obtendo-se uma feição mais ondulada.

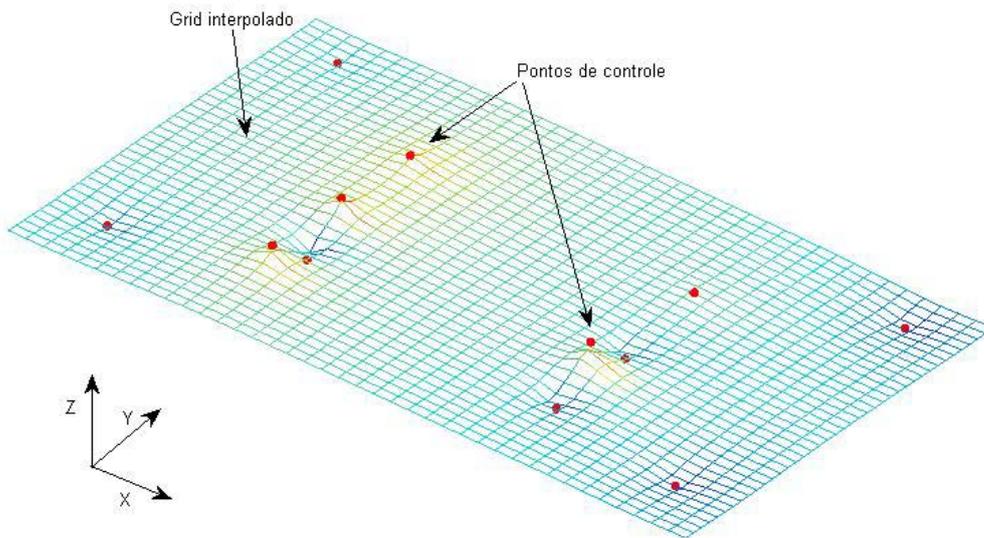


Figura 5.2. Grid interpolado com $\beta = 1$.

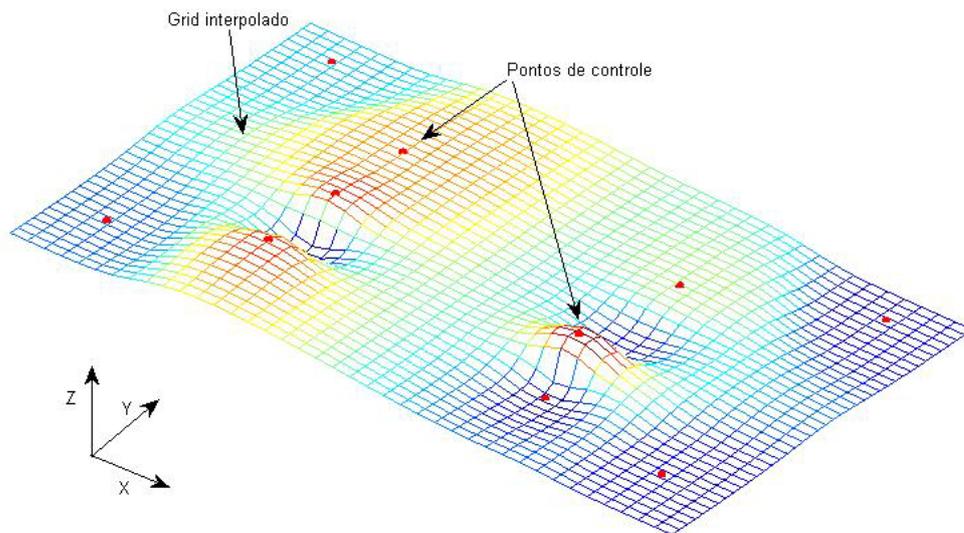


Figura 5.3. Mesmo Grid interpolado com $\beta = 4$

5.2.2. Convolução

A convolução é uma operação matemática na área de análise funcional que junta duas funções para criar uma terceira. Ela é definida como a integral do produto de uma das funções com uma cópia invertida, com relação a um

determinado plano, da outra. A função resultante depende do valor desse deslocamento:

$$(f \otimes g)(t) = \int_0^t f(\tau) * g(t - \tau) dt \quad (5.4)$$

Para funções discretas, pode-se aproximar a integral com um somatório:

$$(f \otimes g)[m] = \sum_x f[x] * g[m - x] \quad (5.5)$$

Operações de convolução podem ser utilizadas para resolver sistemas de equações lineares, mas são principalmente usadas em algoritmos de processamento de sinais e imagens. A Eq. 5.5 é expandida para casos bidimensionais da seguinte forma:

$$(f(x, y) \otimes g(x, y))[m, n] = \sum_x \sum_y f[x, y] * g[m - x, n - y] \quad (5.6)$$

No CarbSM, a convolução bidimensional é utilizada cada vez que é preciso determinar a função de redução por condições marinhas. A superfície deposicional é suavizada seguindo a Eq. 5.6 de forma que $f(x, y)$ seja a superfície a ser suavizada e $g(x, y)$ seja o *kernel* do filtro Gaussiano, como mostrado na Eq. 4.9. O alcance do filtro é definido pelo termo σ_G , o desvio padrão da Gaussiana; quanto maior, maior será o alcance. Para aplicar o filtro, é necessário determinar inicialmente esse alcance em número de nós para poder montar a matriz *kernel* do filtro.

$$nfx = \sigma_G / \Delta x \quad (5.7)$$

$$nfy = \sigma_G / \Delta y \quad (5.8)$$

Sendo:

nfx, nfy – número de nós na direção x e y , respectivamente, do *kernel*

$\Delta x, \Delta y$ – incremento em x e y , respectivamente, do *grid* principal

Tendo as dimensões prontas, a matriz do filtro Gaussiano pode ser montada e aplicada na matriz do *grid* principal. Para cada nó do *grid*, o filtro é aplicado de modo a sempre ter seu elemento central coincidindo com o nó em questão. Ou seja, considerando o nó i, j , o filtro é aplicado do nó $i - ((nfx - 1) / 2)$ até $i + ((nfx - 1) / 2)$ na direção x e do nó $j - ((nfy - 1) / 2)$ até $j + ((nfy - 1) / 2)$ na direção y como pode ser visto na Figura 5.4.

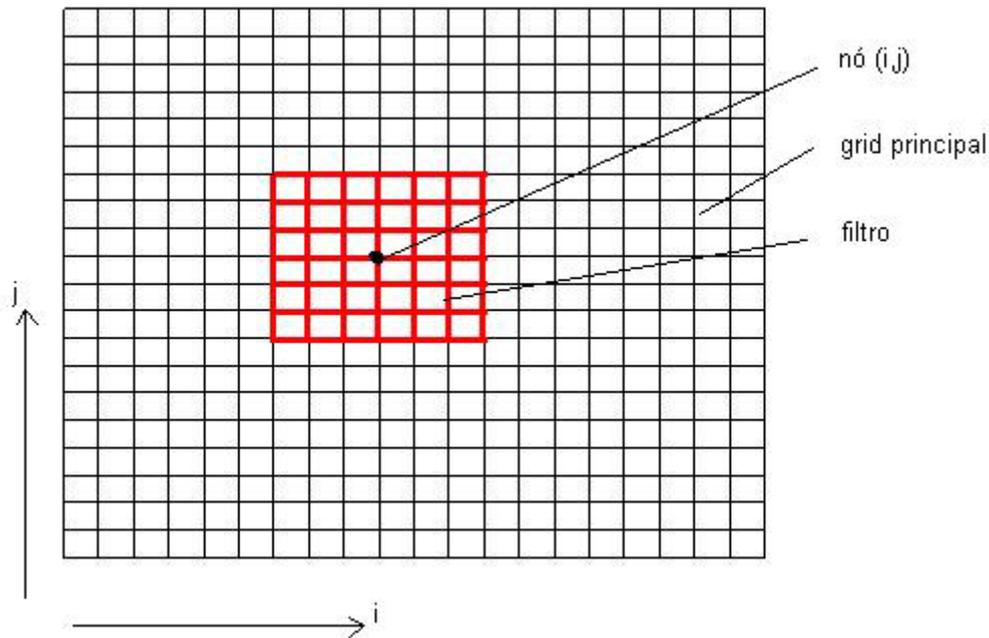


Figura 5.4. O *grid* principal da superfície deposicional com um filtro Gaussiano aplicado em um de seus nós.

O algoritmo de filtro percorre cada nó do *grid* e aplica o *kernel* de forma centralizada no nó, como mostra a Figura 5.4. A função Gaussiana é então aplicada em cima das coordenadas dos nós dentro do alcance do filtro da seguinte maneira, segundo a Eq. 5.6:

$$z_{s,ij} = \sum_{if=i-((nfx-1)/2)}^{i+((nfx-1)/2)} \sum_{jf=j-((nfy-1)/2)}^{j+((nfy-1)/2)} z_{ij} * G(x_{if,jf}, y_{if,jf}) \quad (5.9)$$

Sendo:

$z_{s,ij}$, o valor da coordenada z da superfície deposicional suavizada no nó i, j

z_{ij} , o valor da coordenada z da superfície deposicional no nó i, j

G , a função Gaussiana mostrada na Eq.4.10

x_{iff} e y_{iff} , as coordenadas x e y , respectivamente, da superfície deposicional

Em vários pontos do *grid* perto das bordas do modelo, o filtro Gaussiano vai tentar acessar nós inexistentes por estarem fora da região do modelo (Figura 5.5.). Para cuidar dessas condições, os valores das coordenadas z das bordas são replicadas para os nós virtuais fora do modelo. Para tornar o algoritmo mais eficiente, um *grid* auxiliar é criado com as bordas expandidas já contendo os valores replicados para os nós virtuais. Outra solução para tratar das condições de borda seria de simplesmente dispensar os pontos do filtro Gaussiano que ficariam fora do *grid* principal. Desta maneira, para cada ponto a ser suavizado, teria que ser feito uma verificação para saber quais pontos do filtro Gaussiano estariam dentro do *grid* principal e descartar os outros pontos. Este método, porém, se mostra bem menos eficiente, pois o MATLAB não trabalha bem com verificações e trabalha de maneira muito mais eficiente com aplicações matriciais diretas, por isso esta hipótese foi descartada.

Assim sendo, a Eq. 5.9 é executada como uma multiplicação de matrizes. Também por esse motivo, fica muito mais eficiente gerar uma matriz auxiliar estendida nas bordas e trabalhar com ela. Depois de ter passado por todos os nós da matriz original com o algoritmo de suavização, os valores obtidos são retirados da matriz auxiliar e colocados na matriz z_s . A matriz z , com as coordenadas do *grid* principal é montada inicialmente com nx e ny nós na direção x e y , respectivamente. A matriz auxiliar, z_{aux} , é então montada com $nx+(nfx-1)$ e $ny+(nfy-1)$ nós na direção x e y , respectivamente. Os valores referentes à matriz z são inseridos de maneira central na matriz auxiliar, de $1+((nfx-1)/2)$ até $nax-((nfx-1)/2)$ na direção x e de $1+((nfy-1)/2)$ até $nay-((nfy-1)/2)$ na direção y . Sendo nax e nay o número total de nós nas direções x e y da matriz auxiliar z_{aux} .

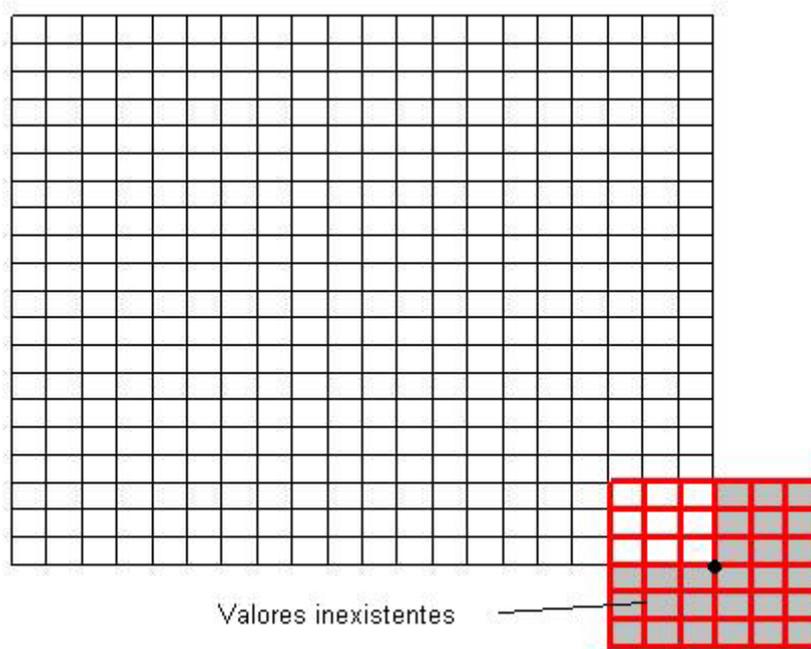


Figura 5.5. Exemplo de valores inexistentes na aplicação do filtro no nó $(nx, 1)$.

Os valores de borda de z são replicados da seguinte maneira para dentro da matriz z_{aux} , e as regiões detalhadas são indicadas na Figura 5.6:

1. De $1 + ((nfx-1)/2)$ até $nax - ((nfx-1)/2)$ na direção x e 1 até $(nfy-1)/2$ na direção $y = z(i, 1)$
2. De $1 + ((nfx-1)/2)$ até $nax - ((nfx-1)/2)$ na direção x e $nay - ((nfy-1)/2)$ até nay na direção $y = z(i, ny)$
3. De 1 até $(nfx-1)/2$ na direção x e $1 + ((nfy-1)/2)$ até $nay - ((nfy-1)/2)$ na direção $y = z(1, j)$
4. De $nax - ((nfx-1)/2)$ até nax na direção x e $1 + ((nfy-1)/2)$ até $nay - ((nfy-1)/2)$ na direção $y = z(nx, j)$
5. De 1 até $(nfx-1)/2$ na direção x e 1 até $(nfy-1)/2$ na direção $y = z(1, 1)$
6. De 1 até $(nfx-1)/2$ na direção x e $nay - ((nfy-1)/2)$ até nay na direção $y = z(1, ny)$
7. De $nax - ((nfx-1)/2)$ até nax na direção x e $nay - ((nfy-1)/2)$ até nay na direção $y = z(nx, ny)$
8. De $nax - ((nfx-1)/2)$ até nax na direção x e 1 até $(nfy-1)/2$ na direção $y = z(nx, 1)$

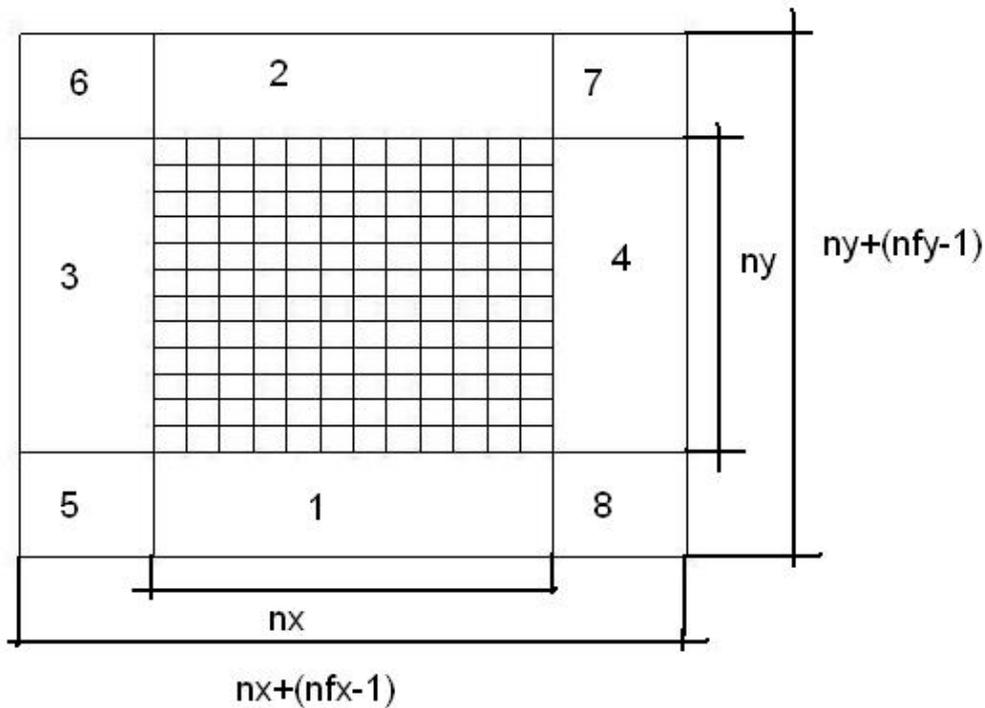


Figura 5.6. Matriz auxiliar z_{aux} com as regiões onde os valores da borda da matriz z são replicados.

Pode-se então calcular os valores para z_s a partir da matriz auxiliar e depois do cálculo essa matriz é descartada. Deve-se salientar também que os valores de z_s somente são utilizados para calcular os valores das funções de restrição S_{OM} e S_{RM} e os depósitos resultantes são somados na matriz das coordenadas z do *grid* principal, sem suavização.

5.2.3. Sedimentação carbonática

Após definir a superfície inicial e calcular todas as funções de restrição por todo o *grid* principal, falta aplicar a equação diferencial Eq. 4.19 para poder calcular a nova camada de acréscimo de sedimentos carbonáticos sobre as camadas já depositadas. O problema é resolvido de forma explícita, pegando os valores já calculados de um passo de tempo anterior para encontrar valores no passo de tempo atual. A derivada da Eq. 4.19 é aproximada utilizando o método numérico de Diferenças Finitas que é bastante simples de ser aplicado nesse caso:

$$(P_{RA}(t) + P_{RR}(t) + P_{RP}(t)) * \Delta t \quad (5.10)$$

Com isso, é obtida a camada de sedimentos a ser acrescentada para um determinado passo de tempo e a espessura em cada nó do *grid* é somada ao que se tinha antes (Figura 5.7).

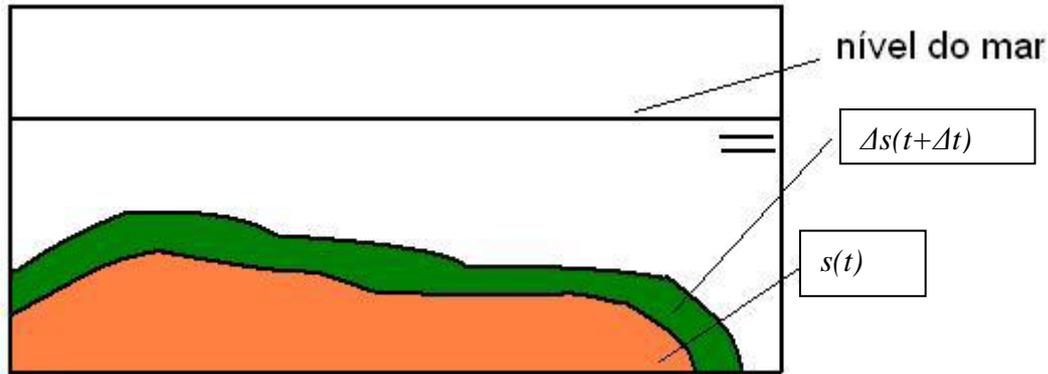


Figura 5.7. Camada de sedimentos adicionais, Δs , calculadas na rotina do tempo $(t+\Delta t)$ é sobreposta à camada de deposições antigas, $s(t)$

A matriz com os valores de $\Delta s(t+\Delta t)$ define um valor de espessura adicional para cada nó do *grid*, mas ainda precisa identificar qual tipo de sedimento vai ser adicionado. Para isso, é calculado o valor relativo de sedimento adicional para cada um dos ambientes deposicionais.

$$PR_{RA} = (P_{RA}(t) * dt / ds(t+dt)) * 100 \quad (5.11)$$

$$PR_{RR} = (P_{RR}(t) * dt / ds(t+dt)) * 100 \quad (5.12)$$

$$PR_{RP} = (P_{RP}(t) * dt / ds(t+dt)) * 100 \quad (5.13)$$

Esses valores percentuais são aplicados na Tabela 1 e é determinado qual tipo de sedimento a ser depositado. Os tipos de depósito em cada nó são armazenadas em uma matriz que é utilizada no momento em que CarbSM desenha as camadas sedimentares.

5.2.4. Rotina Principal

A rotina principal para gerar uma modelagem em CarbSM segue os seguintes passos:

1. Geração da superfície inicial, adicionando pontos de controle e determinando número de nós, distância entre nós e orientação do *grid*
2. Entrada de dados das taxas máximas de crescimento carbonático e dos parâmetros controladores das funções de restrição
3. Determinação do tempo total de simulação e do tamanho do passo de tempo
4. Geração da curva da variação de nível do mar
5. *Loop* através dos passos de tempo
 - 5.1. Suavização da superfície do passo de tempo anterior utilizando filtro Gaussiano para determinar o posicionamento relativo de cada ponto dentro da plataforma carbonática
 - 5.2. Cálculo dos valores das funções de restrição para todo o *grid*
 - 5.3. Aplicação das restrições em cima das taxas máxima de crescimento carbonático para obter P_{RA} , P_{RR} e P_{RP}
 - 5.4. Cálculo dos valores de acréscimo relativo para cada ambiente deposicional, PR_{RA} , PR_{RR} e PR_{RP}
 - 5.5. Atualização na superfície deposicional e armazenamento da matriz dos tipos de sedimentos; fim do *loop*
6. Preparo dos planos de corte e geração do modelo de visualização

5.3. Interface Gráfica

Nessa seção será mostrados os principais componentes da interface gráfica do programa CarbSM.

5.3.1. Tela principal

A tela principal de CarbSM (Figura 5.8.) tem como parte central o *canvas* onde é desenhada a superfície resultante, tem todos os controles de visualização dos resultados e um menu para entrada de dados iniciais.

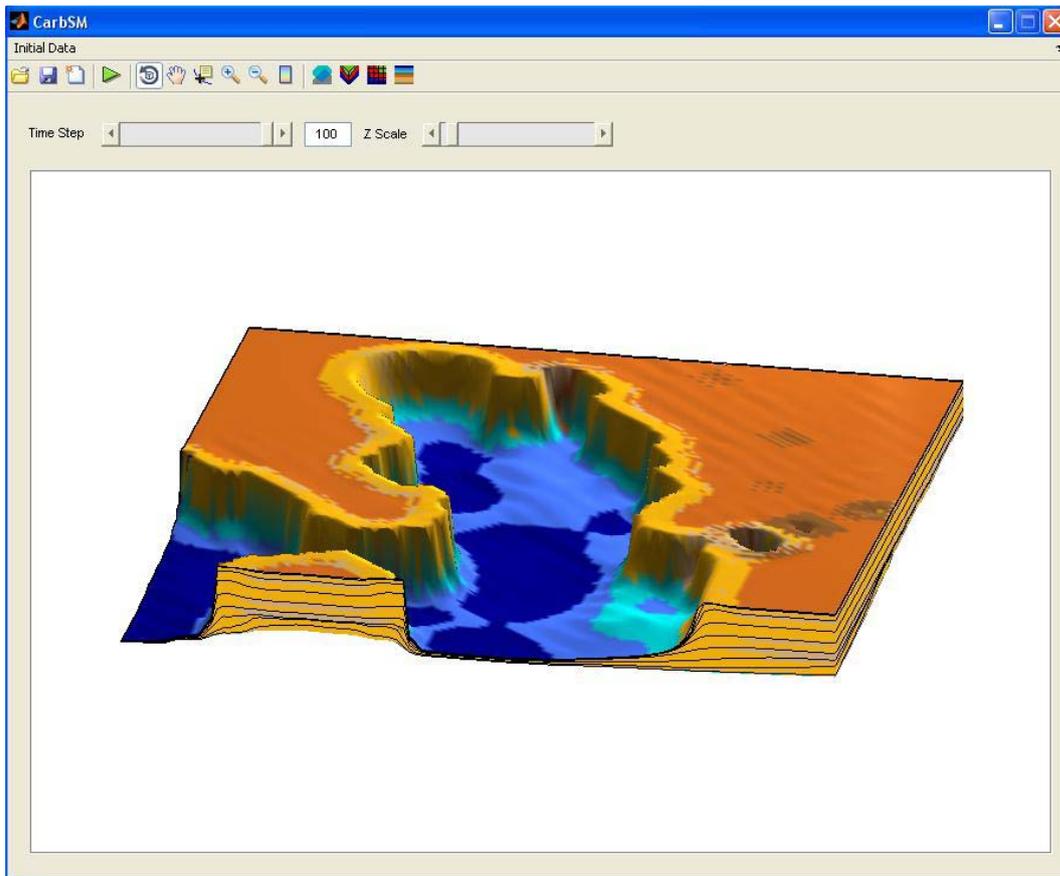


Figura 5.8. Tela principal do programa CarbSM com os controles de visualização, o *canvas* principal e o menu de entrada de dados

5.3.2. Entrada de dados

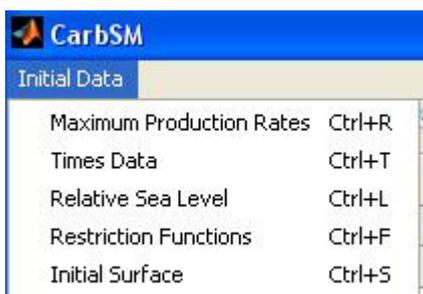


Figura 5.9. *Menu* para entrada dos dados necessários para rodar a simulação do CarbSM

A entrada de dados (Figura 5.9.) é dividida pelos seguintes itens:

- Taxas máximas de produção carbonática (Figura 5.10.):

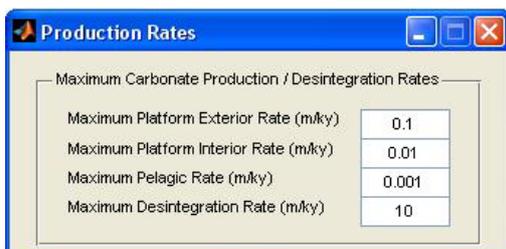


Figura 5.10. Entrada de dados para taxas máximas de produção carbonática para as diferentes fábricas

- Tempo de simulação

No tempo de simulação é definido o tempo total da simulação, o intervalo de tempo da simulação e o intervalo para as curvas de tempo na visualização do modelo resultante (Figura 5.11.):

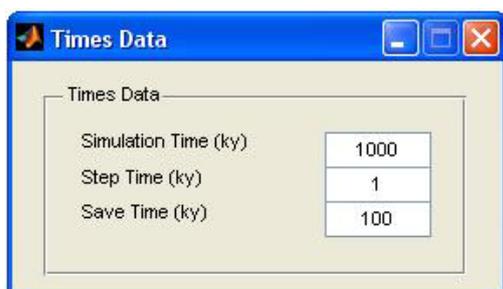


Figura 5.11. Entrada dos tempos da simulação

- Definição da curva do nível relativa do mar (Figura 5.12.)

Aqui é definida e ajustada a curva de nível relativo do mar, segundo a Eq. 4.2. Outras curvas que não podem ser definidas através da equação podem ser carregadas por arquivo de texto. Além de entrar com os parâmetros da Eq. 4.2 para calcular a curva, o usuário tem a opção de fazer um ajuste fino do nível inicial do mar e da subsidência linear com ajuda do mouse na curva.

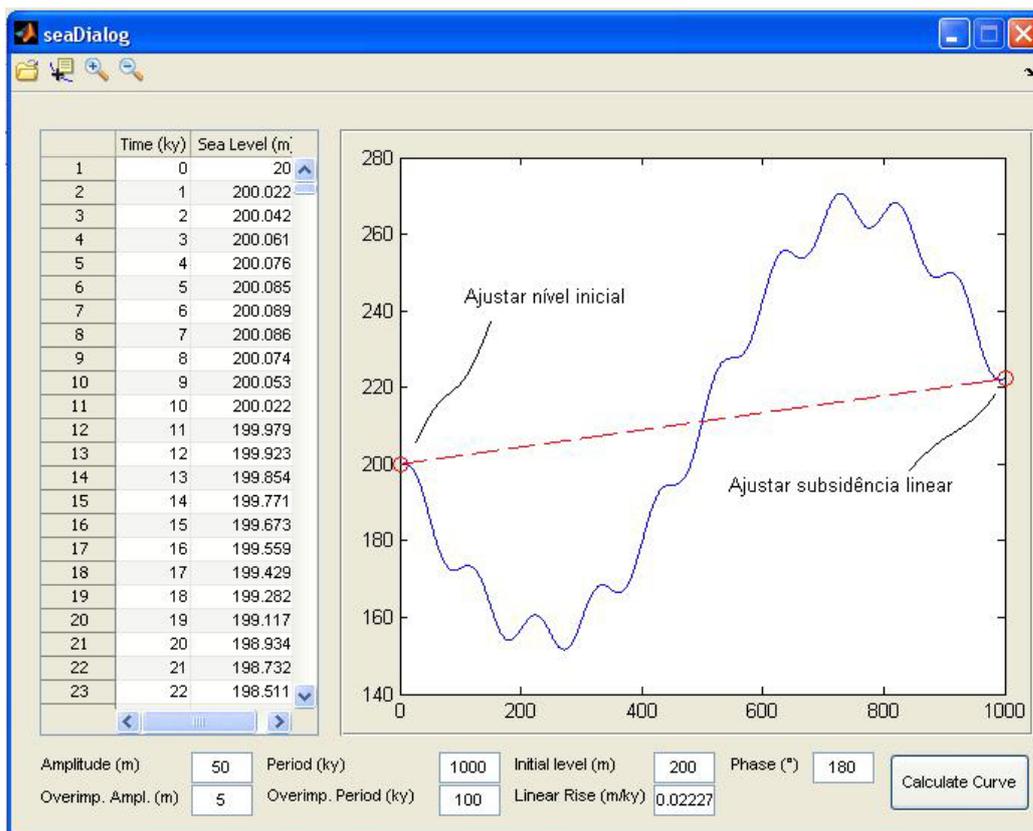


Figura 5.12. Definição da curva do nível relativo do mar.

- Funções de restrição (Figura 5.13.)

Para definição das funções de restrição utilizadas no modelo, basta o usuário entrar com todos os parâmetros apresentados na seção anterior, além de ter uma visualização das curvas geradas. Dependendo das condições da bacia a ser simulada, o usuário pode optar pelas funções de restrição a serem ativadas para a simulação:

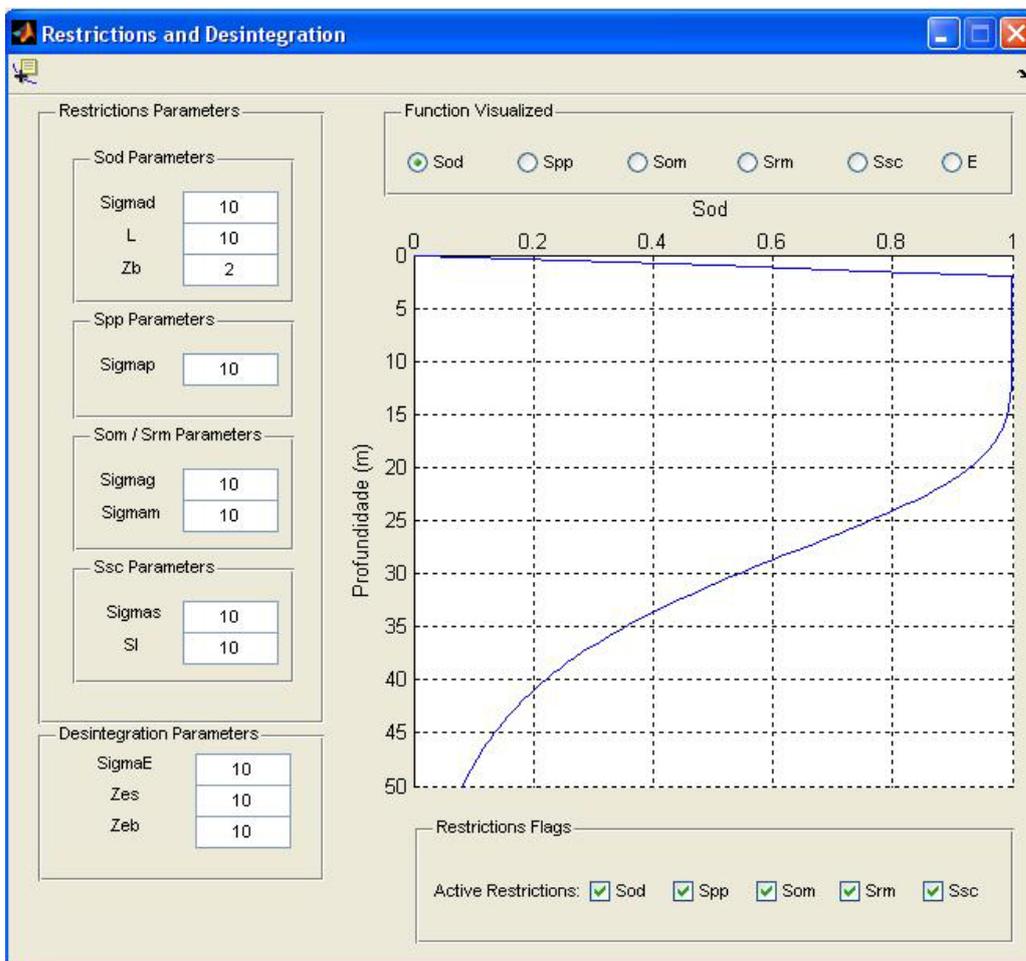


Figura 5.13. Definição dos parâmetros das funções de restrição e do uso das mesmas.

- Superfície inicial (Figura 5.14.)

Para iniciar a simulação, é necessário definir uma superfície inicial que é interpolada através dos pontos de controle lidos de um arquivo texto. Após a leitura dos pontos de controle é selecionada a área que se deseja usar na simulação. Então o usuário define todos os parâmetros do *grid* a ser interpolado, como número de nós nas direções x e y , a distância uniforme entre os nós e o expoente para o algoritmo de interpolação. Além disto, o diálogo também apresenta vários controles para melhorar a visualização do *grid* criado.

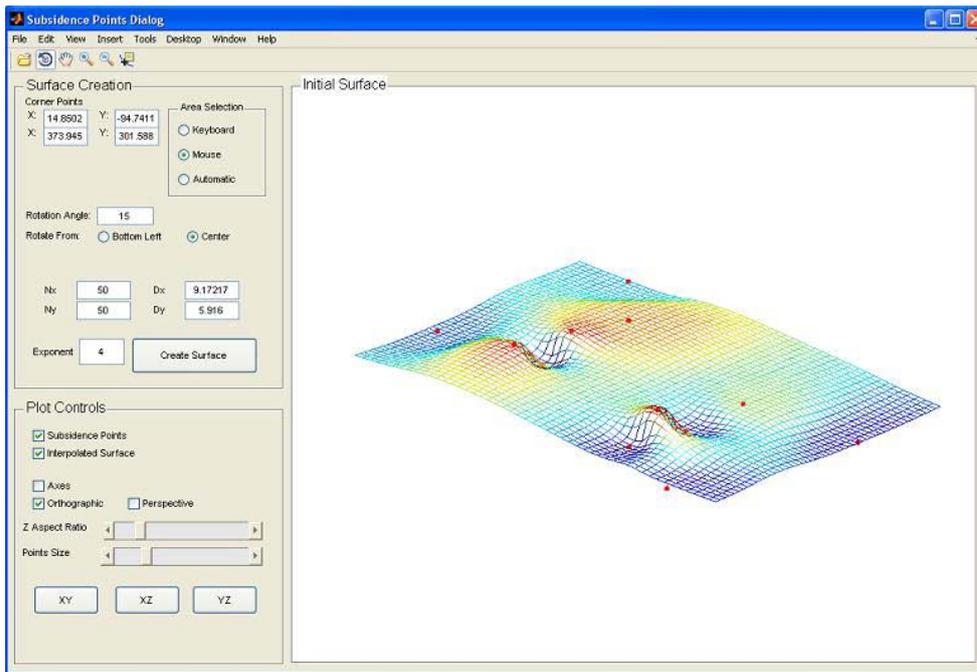


Figura 5.14. Diálogo para definição do *grid* que será utilizado na simulação

- Visualização da superfície resultante (Figura 5.15.)

Depois de feita a simulação, a tela principal de CarbSM apresenta o modelo resultante e algumas opções de visualização.

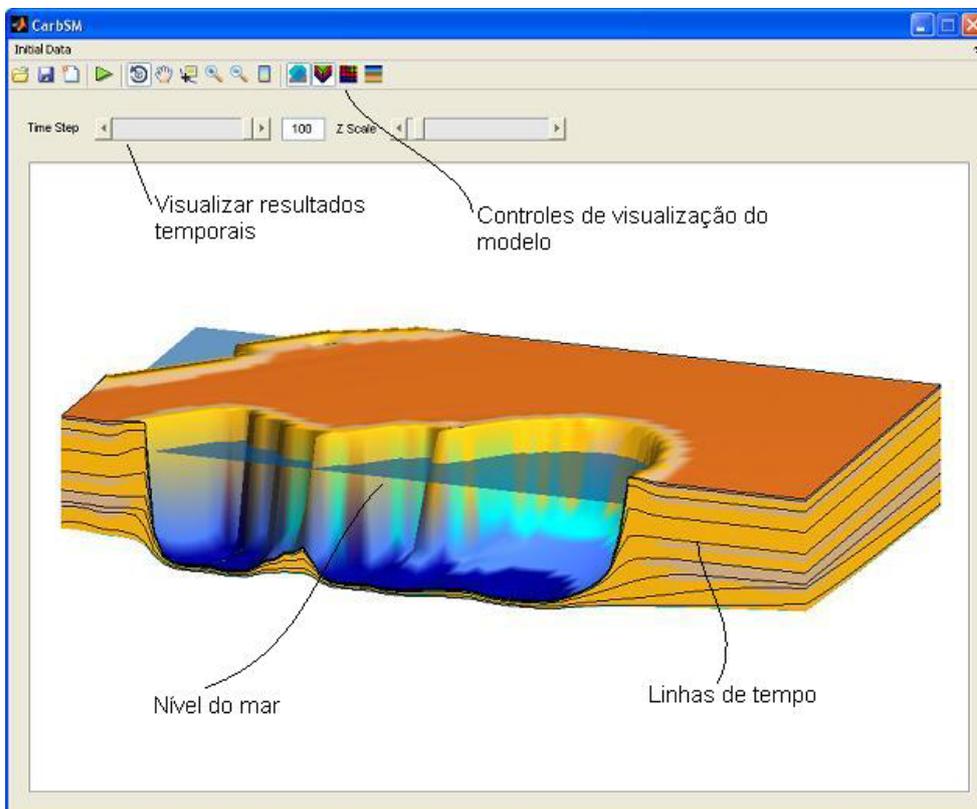


Figura 5.15. Visualização dos resultados