

Roberto Mentzingen Rolo

**Modelagem geológica implícita com funções  
distância assinaladas**

Porto Alegre

2017



Roberto Mentzingen Rolo

## **Modelagem geológica implícita com funções distância assinaladas**

Esta dissertação foi analisada e julgada adequada para a obtenção do título de Mestre em Engenharia, área de concentração de Tecnologia Mineral, e aprovada em sua forma final pelo Orientador e pela Banca Examinadora designada pelo Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais da Universidade Federal do Rio Grande do Sul.

Universidade Federal do Rio Grande do Sul - UFRGS

Escola de Engenharia

Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais

Orientador: Prof. Dr. João Felipe Coimbra Leite Costa

Porto Alegre

2017

---

Roberto Mentzingen Rolo

Modelagem geológica implícita com funções distância assinaladas/ Roberto Mentzingen Rolo. – Porto Alegre, 2017-  
104 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. João Felipe Coimbra Leite Costa

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Sul - UFRGS  
Escola de Engenharia  
Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais,  
2017.

1. modelagem geológica. 2. modelagem geológica implícita. 3. funções distância assinaladas I. Prof. Dr. João Felipe Coimbra Leite Costa. II. Universidade Federal do Rio Grande do Sul. III. Faculdade de Engenharia. IV. Modelagem geológica implícita com funções distância assinaladas

CDU 02:141:005.7

---

Roberto Mentzingen Rolo

## **Modelagem geológica implícita com funções distância assinaladas**

Esta dissertação foi analisada e julgada adequada para a obtenção do título de Mestre em Engenharia, área de concentração de Tecnologia Mineral, e aprovada em sua forma final pelo Orientador e pela Banca Examinadora designada pelo Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais da Universidade Federal do Rio Grande do Sul.

Trabalho aprovado. Porto Alegre, 24 de fevereiro de 2017:

---

**Prof. Dr. André Cezar Zingano**  
PPGE3M - UFRGS

---

**Prof. Dr. Diego Machado Marques**  
Convidado externo ao PPGE3M

---

**Dr. Marcelo Cheviche Godoy**  
Newmont Mining Corporation

Porto Alegre  
2017



# Agradecimentos

Ao Professor João Felipe por compartilhar um pouco do imenso conhecimento e pela recepção de braços abertos no laboratório. Aos colegas do LPM que contribuíram para o desenvolvimento desse trabalho, Ricardo Radkte, David Drummond, Áttila Leães e Péricles Machado. Aos demais colegas de laboratório, em especial, Ryu Okada, Marcel Bassani, Cristina Araújo, Augusto Torres, Anuar Bergamaschi, Lucas Roncarati, Marcelo Batelocchi, Ricardo Hundelshausen, Ricardo Rodrigues, José Guilherme e George Gasper. Aos ICs, Rafael e Bento.

À minha família, em especial, Custódia e Marcelo pelo suporte. À Camila pelo companheirismo e incondicional apoio.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa concedida. À Newmont pelo apoio e informação fornecida à essa dissertação.





*“Life is made of choices  
Full of opportunities  
All we have to do is  
Recognize the pieces of the game  
Make out what is substancial  
And be proud of our undying past”  
(Reffer - Hidden Scars)*



# Resumo

Previamente à cada estimativa ou simulação geoestatística os domínios geológicos do depósito devem ser modelados, o que tradicionalmente é feito de forma manual por um geomodelador, em um processo laborioso, demorado e subjetivo. Por essa razão novas técnicas conhecidas como métodos implícitos veem surgindo. Essas técnicas fornecem algoritmos que substituem o processo de digitalização manual dos métodos explícitos por alguma forma de procedimento automático. Essa dissertação visita alguns métodos implícitos bem estabelecidos com atenção especial à modelagem geológica implícita com funções distância assinalada. Um estudo de caso em um banco de dados real é apresentado e a aplicabilidade do método discutida. Embora não substitua por completo um geomodelador experiente, o método provou ser capaz de gerar modelos geológicos semi-automáticos realistas a partir dos dados amostrais, e se mostra útil principalmente nas fases iniciais da pesquisa mineral.

**Palavras-chaves:** modelagem geológica, modelagem geológica implícita, funções distância assinaladas.



# Abstract

Prior to every geostatistical estimation or simulation study there is a need for delimiting the geologic domains of the deposit, which is traditionally done manually by a geomodeler in a laborious, time consuming and subjective process. For this reason, novel techniques referred to as implicit modelling have appeared. These techniques provide algorithms that replace the manual digitization process of the traditional methods by some form of automatic procedure. This dissertation covers a few well established implicit methods currently available with special attention to the signed distance function methodology. A case study based on a real dataset was performed and its applicability discussed. Although it did not replace an experienced geomodeler, the method proved to be capable in creating semi-automatic geological models from the sampling data, especially in the early stages of exploration.

**Key-words:** geologic modeling, implicit geologic modeling, signed distance function, domaining.



# Lista de ilustrações

Figura 1 – Fases da mineração. . . . .	19
Figura 2 – Fluxograma ilustrando a metodologia passo a passo. . . . .	23
Figura 3 – Veio de quartzo-ouro em rocha máfica. . . . .	29
Figura 4 – Linha dos reais dividida em dois subdomínios pelos pontos $x = -1$ e $x = 1$ . . . . .	31
Figura 5 – Função implícita definindo as regiões $\Omega^-$ e $\Omega^+$ bem como a fronteira $\partial\Omega$ . . . . .	31
Figura 6 – Função distância assinalada definindo as regiões $\Omega^-$ e $\Omega^+$ bem como a fronteira $\partial\Omega$ . . . . .	32
Figura 7 – Furo de sonda interceptando uma camada de carvão, à esquerda, e o valor da função distancia assinalada à direita. . . . .	33
Figura 8 – Três amostras $s_1$ , $s_2$ e $s_3$ e o um ponto $q$ a ser estimado. . . . .	34
Figura 9 – Demonstração simplificada do funcionamento das funções de bases radiais em uma dimensão. . . . .	35
Figura 10 – Interpolantes do <i>Leapfrog</i> <sup>®</sup> . . . . .	36
Figura 11 – Interpolante isotrópico. . . . .	37
Figura 12 – Tendência global aplicada ao interpolante. . . . .	37
Figura 13 – Tendência estrutural aplicada ao interpolante. . . . .	37
Figura 14 – Princípio do método dos campos potenciais em duas dimensões. . . . .	40
Figura 15 – Modelo geológico criado a partir do método dos campos potenciais, em planta (acima). Seção N6 (abaixo). . . . .	43
Figura 16 – Transformação do banco de dados categóricos em indicadores binários. . . . .	46
Figura 17 – Função distância assinalada calculada para cada ponto amostral. . . . .	46
Figura 18 – Função distância interpolada para todos os locais de interesse. . . . .	48
Figura 19 – Modelo geológico criado a partir do sinal da função distância interpolada. . . . .	49
Figura 20 – Função distância interpolada para todos os locais de interesse. . . . .	50
Figura 21 – Mapas de probabilidades gerados com dois valores de gamma diferentes. À esquerda $\gamma = 80$ , à direita $\gamma = 175$ . . . . .	51
Figura 22 – Distâncias estimadas e transformadas em probabilidades para um mesmo bloco. . . . .	52
Figura 23 – Modelos geológicos gerados, sem correção das proporções à direita e com correção, usando um fator $\lambda = 0,99$ à esquerda. . . . .	53
Figura 24 – Visão do painel de algoritmos do <i>SGeMS</i> mostrando a opção <i>DFMOD</i> . . . . .	54
Figura 25 – Visão do <i>plugin signed distances</i> . . . . .	55
Figura 26 – Visão da aba <i>general</i> do <i>plugin interpolator</i> . . . . .	56
Figura 27 – Visão da aba <i>variogram</i> do <i>plugin interpolator</i> . . . . .	57
Figura 28 – Visão da aba <i>options</i> do <i>plugin interpolator</i> . . . . .	57

Figura 29 – <i>Scatterplot</i> entre as distâncias calculadas pelo <i>plugin</i> desenvolvido contra distâncias calculadas a partir da rotina <i>DFMod</i> do <i>GSLib</i> . . . . .	59
Figura 30 – <i>Scatterplot</i> entre os modelos criados pelo <i>plugin</i> desenvolvido e rotina <i>DFMod</i> do <i>GSLib</i> . . . . .	60
Figura 31 – Histogramas do modelo baseado somente nas distâncias estimadas à esquerda, dos dados ao centro, e do modelo após execução do <i>servo system</i> à direita. . . . .	61
Figura 32 – Dados em planta à esquerda e em perfil à direita. . . . .	63
Figura 33 – Histograma dos dados. . . . .	64
Figura 34 – Distâncias assinaladas calculadas para cada uma das litologias. . . . .	64
Figura 35 – Litologia 1, e respectiva proporção, modelada variando o range do variograma: 150m à esquerda, 1500m ao centro e 15000m à direita. . . . .	65
Figura 36 – Modelos calculados a partir de variogramas com diferentes proporções de efeito pepita. Da direita para a esquerda: 0%, 10%, 50% e 90%. . . . .	66
Figura 37 – Variograma das distâncias assinaladas calculadas para cada uma das litologias. . . . .	67
Figura 38 – Scatterplots entre modelos gerados com diferentes estratégias de krigagem. . . . .	69
Figura 39 – Comparação entre os modelos criados: modelo explícito à esquerda e modelo implícito à direita. . . . .	70
Figura 40 – <i>Scatterplot</i> entre os modelos explícitos e implícitos. . . . .	71
Figura 41 – Modelo criado utilizando a técnica do vizinho mais próximo. . . . .	72
Figura 42 – Seção vertical 10/70 em x. . . . .	73
Figura 43 – Seção vertical 37/70 em x. . . . .	74
Figura 44 – Seção vertical 56/70 em x. . . . .	74
Figura 45 – Seção vertical 67/70 em x. . . . .	75
Figura 46 – Seção vertical 16/60 em y. . . . .	75
Figura 47 – Seção vertical 27/60 em y. . . . .	76
Figura 48 – Seção vertical 43/60 em y. . . . .	76
Figura 49 – Seção vertical 54/60 em y. . . . .	77



# Lista de tabelas

Tabela 1 – Ângulos de rotação e proporções entre as direções principais usadas no cálculo da função distância anisotrópica. . . . .	59
Tabela 2 – Parâmetros para o cálculo da variograma experimental. . . . .	66
Tabela 3 – Propriedades do <i>grid</i> . . . . .	68
Tabela 4 – Parâmetros de krigagem ordinária. . . . .	68



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
<b>1.1</b>	<b>Meta</b>	<b>22</b>
<b>1.2</b>	<b>Objetivos específicos</b>	<b>22</b>
<b>1.3</b>	<b>Metodologia</b>	<b>22</b>
<b>1.4</b>	<b>Estrutura da dissertação</b>	<b>23</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>25</b>
<b>2.1</b>	<b>Estacionariedade</b>	<b>25</b>
2.1.1	Funções aleatórias	25
2.1.2	Momentos	26
2.1.2.1	Momento de primeira ordem	26
2.1.2.2	Momentos de segunda ordem	26
2.1.3	Hipótese de estacionariedade	27
2.1.3.1	Estacionariedade estrita	27
2.1.3.2	Estacionariedade segunda ordem	28
2.1.3.3	Hipótese intrínseca	28
2.1.3.4	Quasi-estacionariedade	28
2.1.4	Decisão de estacionariedade	29
<b>2.2</b>	<b>Funções implícitas</b>	<b>30</b>
2.2.1	Funções distâncias assinaladas	31
<b>2.3</b>	<b>Método implementado no software Leapfrog<sup>®</sup></b>	<b>33</b>
2.3.1	Metodologia	33
2.3.1.1	Interpolação	34
2.3.1.2	Anisotropia	36
2.3.1.3	Visão geral	38
<b>2.4</b>	<b>Método dos campos potenciais</b>	<b>38</b>
2.4.1	Metodologia	39
2.4.2	Tipos de dados	39
2.4.2.1	Codificação dos dados	40
2.4.2.2	Interpolação do campo potencial	41
2.4.2.3	Implementação do algoritmo de cokrigagem	41
2.4.2.4	Variância do campo potencial	42
2.4.2.5	Incerteza nos modelos	42
2.4.2.6	Visão geral	42

<b>3</b>	<b>MODELAGEM GEOLÓGICA IMPLÍCITA COM FUNÇÕES DISTÂNCIA ASSINALADAS</b> . . . . .	<b>45</b>
<b>3.1</b>	<b>Metodologia</b> . . . . .	<b>45</b>
3.1.1	Caso binário . . . . .	45
3.1.2	Aplicação da metodologia para múltiplos domínios geológicos . . . . .	48
<b>3.2</b>	<b>Uma medida heurística de incerteza (<i>softmax transformation</i>)</b> . . . . .	<b>50</b>
3.2.0.1	Transformação das distâncias . . . . .	51
<b>3.3</b>	<b>Correção das proporções globais dos domínios (<i>servo system</i>)</b> . . . . .	<b>52</b>
3.3.1	<i>Servo system</i> . . . . .	52
<b>3.4</b>	<b>O <i>Plug-in</i></b> . . . . .	<b>54</b>
3.4.1	<i>Signed distances</i> . . . . .	54
3.4.2	<i>Interpolator</i> . . . . .	55
<b>3.5</b>	<b>Validação</b> . . . . .	<b>58</b>
<b>4</b>	<b>ESTUDO DE CASO</b> . . . . .	<b>63</b>
<b>4.1</b>	<b>Variografia</b> . . . . .	<b>65</b>
<b>4.2</b>	<b>Interpolação</b> . . . . .	<b>68</b>
<b>4.3</b>	<b>Resultados</b> . . . . .	<b>69</b>
4.3.1	Comparação . . . . .	72
<b>4.4</b>	<b>Discussão</b> . . . . .	<b>77</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>81</b>
<b>5.1</b>	<b>Recomendações para trabalhos futuros</b> . . . . .	<b>81</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>83</b>
	<b>APÊNDICES</b> . . . . .	<b>87</b>
	<b>APÊNDICE A – ALGORITMO EM PYTHON QUE CALCULA AS DISTÂNCIAS ANISOTRÓPICAS ASSINALADAS (<i>SIGNED DISTANCES</i>)</b> . . . . .	<b>89</b>
	<b>APÊNDICE B – ALGORITMO EM PYTHON QUE INTERPOLA AS DISTÂNCIAS ASSINALADAS E CRIA O MODELO GEOLÓGICO (<i>INTERPOLATOR</i>)</b> . . . . .	<b>93</b>
	<b>ANEXOS</b> . . . . .	<b>101</b>
	<b>ANEXO A – ARQUIVO DE PARÂMETRO USADO NA VALIDAÇÃO</b> . . . . .	<b>103</b>

# 1 Introdução

Hustrulid, Kuchta e Martin (2013) fragmentam a evolução de uma mina em três fases distintas: (1) Planejamento, (2) Implementação e (3) Produção. A Figura 1 é uma linha do tempo mostrando a relação entre as diferentes fases e seus estágios. Na fase de planejamento os depósitos minerais são investigados no que diz respeito à sua atratividade econômica e exequibilidade técnica. A fase de planejamento culmina em um relatório de viabilidade, no qual é baseada a decisão de investir e dar continuidade ao empreendimento mineiro, iniciando a fase de implementação, período de maior fluxo de caixa, quando mina e usina de processamento são desenvolvidas. Finalmente a mina entra em operação, o minério é explorado e processado, e o produto vendido no mercado.

Figura 1 – Fases da mineração.



Modificado de Hustrulid, Kuchta e Martin (2013)

A avaliação das reservas é parte fundamental e alicerce do estudo de viabilidade, os teores e tonelagens são quantificados e modelos numéricos que caracterizam a geologia em subsuperfície criados com base nos dados de sondagem. A partir desses modelos, engenheiros e geólogos planejam e tomam as decisões econômicas e técnicas: Escolha do método de lavra, alternativas de processamento mineral, operações auxiliares, estimativas de custos operacionais, capital de investimento e projeção do lucro. Como resultado, uma estratégia que determina quanto material será removido ano a ano é traçada. O projeto determina os limites econômicos da mina e a sequência ótima de lavra bloco a bloco, baseado no modelo geoestatístico de blocos.

Na fase de planejamento é possível minimizar o capital e custo operacional do projeto final e ao mesmo tempo maximizar a operabilidade e lucro. A influência relativa nos custos de cada fase é mostrado na linha vermelha da Figura 1 (HUSTRULID; KUCHTA;

MARTIN, 2013). A principal causa de fracasso em empreendimentos mineiros é falta de conhecimento a respeito do corpo mineralizado. Isto posto, estimativas precisas e acuradas são importantes. Até mesmo um pequeno desvio entre produção planejada e real pode causar sérios prejuízos financeiros.

A avaliação dos recursos de uma mina é composta de duas etapas (CHILÈS et al., 2004): (1) delimitação das várias unidades geológicas, correspondentes às diferentes formações geológicas ou diferentes litologias; (2) estimativas e/ou simulação de teores em cada unidade modelada.

Desse modo, previamente à toda estimativa e/ou simulação de teores, há a necessidade de delimitar as unidades geológicas, implicando em decisão de estacionariedade e homogeneidade mineralógica em cada domínio modelado. Isto é, Os teores de cada domínio geológico pertencem a diferentes populações estatísticas e são caracterizados através de modelos de distribuição e semivariogramas específicos, resultando em estimativas e/ou simulações distintas para cada população (MCLENNAN, 2007).

Modelos geológicos descrevem a extensão, forma e volume das diferentes unidades geológicas no espaço. Gerar modelos geológicos corretos é necessário para que estimativas mais precisas de volume/massa e teores sejam obtidas e problemas de diluição ocasionados pela falta de aderência do modelo de blocos às estruturas geológicas do depósito sejam reduzidos (RASERA, 2014).

Tradicionalmente os modelos geológicos são criados explicitamente, através de um processo de digitalização dos contatos geológicos. Polígonos são construídos manualmente em seções, por um geomodelador, honrando os dados amostrais, esses polígonos são conectados por linhas-guia e então interpolados por triangulação, gerando sólidos que representam as unidades geológicas. Esse processo apresenta alguns pontos críticos segundo Cowan et al. (2003):

- É demorado e requer um geomodelador experiente;
- É subjetivo, já que cada geomodelador interpretará, e produzirá um modelo diferente a partir do mesmo banco de dados, tornando replicação e auditoria externa do modelo tarefas árduas;
- É inflexível, pois atualizar o modelo à medida que novos dados são adquiridos é demorado e laborioso.

Para muitas minas, apenas um único modelo geológico é mantido, dada a limitação de tempo imposta pelos métodos explícitos. Raramente há oportunidade de modelar interpretações alternativas e comparar estimativas de recursos baseadas em diferentes modelos. Não havendo assim, oportunidade de avaliar as incertezas inerentes ao modelamento geológico (COWAN et al., 2003).

Ainda que *softwares* modernos de mineração forneçam ferramentas computacionais para visualizar os dados de sondagem e agilizar o processo de digitalização manual, os métodos explícitos ainda sofrem com as desvantagens apresentadas. Por esse motivo, novas técnicas, conhecidas como modelagem implícita, vêm surgindo. São algoritmos que reduzem o nível de subjetividade, substituindo o processo de digitalização manual por alguma forma de procedimento automático (SILVA, 2015).

As técnicas de modelagem geológica implícita se dividem em dois grandes grupos:

O grupo das técnicas determinísticas. Métodos diretos e computacionalmente rápidos. Alguns métodos estabelecidos nessa categoria são a interpolação suavizada discreta (MALLET, 2002), o método dos campos potenciais (CHILÈS et al., 2004; CALCAGNO et al., 2008; RENARD et al., 2013), operacionalizado no *software Isatis*<sup>®</sup>, e a metodologia implementada no *software* de modelagem geológica *Leapfrog*<sup>®</sup> (COWAN et al., 2002; COWAN et al., 2003), amplamente utilizado pela indústria.

O outro grupo compreende as técnicas estocásticas de modelagem geológica. Métodos complexos que demandam grande esforço computacional. Entretanto, no lugar de um único modelo geológico, esses métodos geram diversas realizações equiprováveis da distribuição espacial das diferentes litologias, reproduzindo os atributos estatísticos das amostras. Assim, a partir da análise conjunta das realizações é possível avaliar a incerteza associada ao modelo geológico.

São métodos consolidados: A simulação sequencial dos indicadores (JOURNEL, 1983), simulação sequencial gaussiana truncada (JOURNEL; ISAAKS, 1984), simulação plurigaussiana (GALLI et al., 1994), métodos baseados em simulação objetos (*object-based*) (BRIDGE; LEEDER, 1979), recentemente surgiram os métodos de modelagem de superfície (*surface-based*) (PYRCZ; CATUNEANU; DEUTSCH, 2005). Contudo, os algoritmos tradicionais de simulação, baseados em estatísticas de dois pontos, não são capazes de reproduzir estruturas geológicas que apresentam complexas interdependências, assim surgem os métodos baseados em geoestatística multiponto (MPS) (GUARDIANO; SRIVASTAVA, 1993).

Silva e Deutsch (2012) apresentaram a modelagem geológica implícita com funções distância assinaladas para modelar múltiplos domínios geológicos simultaneamente. Deutsch e Wilde (2013) o utilizou para modelar múltiplas camadas de carvão, e mostrou que o método é uma poderosa ferramenta de modelagem. Wilde e Deutsch (2011) e Silva e Deutsch (2013) introduziram duas medidas heurísticas diferentes de avaliação de incertezas e Silva e Deutsch (2013) uma ferramenta para corrigir as proporções globais dos diferentes domínios. E mostraram, novamente, a competência do método. Silva (2015) revisitou o método com a finalidade de construir, a partir dos dados amostrais, imagens de treinamento (*data-driven training images*) que forneçam as estatísticas de múltiplos pontos (MPS) para a simulação de litologias.

A modelagem geológica implícita com funções distância assinaladas é uma técnica determinística, baseada na interpolação de uma função distância em um *grid*. Funções de distância assinaladas medem o grau de separação entre as diferentes litologias, e dependem da orientação, forma e extensão dos corpos geológicos. Distâncias positivas indicam o exterior do domínio enquanto distâncias negativas indicam o interior do domínio. As distâncias assinaladas são calculadas para cada ponto amostral e para cada litologia, e interpoladas para os locais não amostrados. Um modelo geológico é criado a partir dos valores estimados para as distâncias assinaladas. O método é computacionalmente eficiente, direto e não depende de funções matemáticas complicadas. Ainda assim, consegue reproduzir de maneira satisfatória estruturas geológicas de grande escala nos modelos numéricos (SILVA, 2015).

Essa dissertação é uma extensão do trabalho de Deutsch e Wilde (2013) e Silva (2015). Um *plug-in* implementando o método no *software SGeMS* foi desenvolvido e apresentado. A aplicabilidade do método mais uma vez foi posta à prova em um estudo de caso conduzido em um banco de dados real. E suas principais características, vantagens e limitações discutidas.

## 1.1 Meta

Essa dissertação de mestrado tem como meta investigar a aplicabilidade da modelagem geológica implícita com funções distância assinaladas como substituto ou método auxiliar à metodologia explícita de modelagem geológica, amplamente adotada pela indústria. Conhecidas as limitações e desvantagens da última e simplicidade e rapidez da primeira.

## 1.2 Objetivos específicos

A fim de atingir a meta proposta os seguintes objetivos foram delineados:

1. Operacionalizar o método no *software* geoestatístico de código aberto *SGeMS*, desenvolvendo um *plug-in* funcional em *python*;
2. Conduzir um estudo de caso em um banco de dados real e verificar a qualidade do modelo gerado pela metodologia proposta, comparando-o a um modelo de referência.

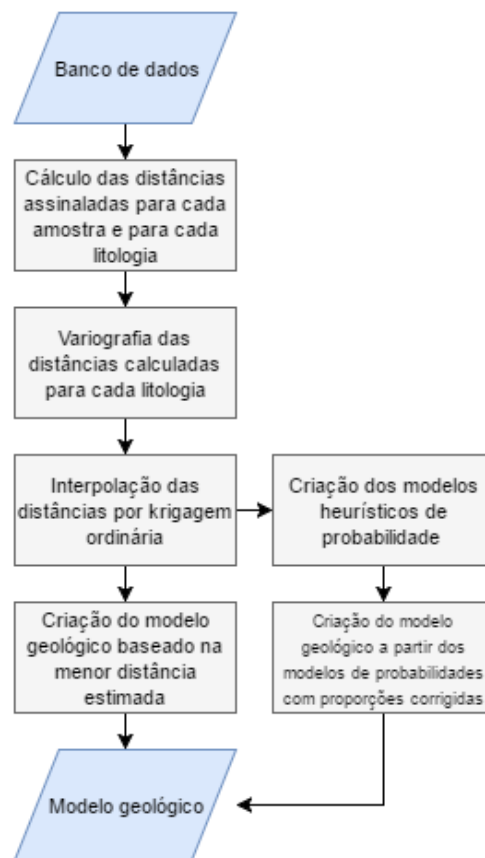
## 1.3 Metodologia

A partir de um banco de dados categóricos o valor da função distância assinalada é calculado para cada litologia em cada ponto amostral. As distâncias calculadas são então



variografadas e interpoladas, por krigagem ordinária, para todo o *grid*. Assim, o valor da função distância assinalada é conhecido em todos os nós e para cada litologia. Um modelo geológico pode ser criado diretamente a partir do valor estimado para as distâncias. Ou então, modelos de probabilidade podem ser criados a partir das distâncias. E então, um modelo geológico, que tem as proporções corrigidas para corresponder a uma proporção alvo, criado a partir dos modelos de probabilidade. A Figura 2 ilustra o fluxograma do processo.

Figura 2 – Fluxograma ilustrando a metodologia passo a passo.



O modelo criado implicitamente com a metodologia proposta foi comparado com um modelo criado explicitamente, a partir do mesmo banco de dados. Assim, foi possível checar se o algoritmo reproduziu satisfatoriamente as estruturas geológicas interpretadas pelo geomodelador.

## 1.4 Estrutura da dissertação

O capítulo 2 Revisa os conceitos de funções implícitas, estacionariedade e os principais métodos determinísticos de modelagem implícita baseados em funções implícitas: método dos campos potenciais, implementado no *software Isatis*<sup>®</sup>, e o método implementado no *software Leapfrog*<sup>®</sup>.

O capítulo 3 apresenta o arcabouço teórico da modelagem geológica implícita com funções distância assinaladas, o *plug-in* desenvolvido para o *software SGeMS* e os resultados da validação do algoritmo proposto tendo como referência resultados análogos da rotina *DFMod* da biblioteca de algoritmos geoestatísticos *GSLib*.

O capítulo 4 discute os resultados do estudo de caso conduzido em um banco de dados real, proveniente de um depósito de ouro. Avalia a aplicabilidade do método como substituto ou método auxiliar à metodologia tradicional (explícita) de modelagem geológica, comparando o modelo gerado a partir da metodologia proposta a um modelo de referência, criado explicitamente por um geomodelador.

O capítulo 5 encerra a dissertação apresentando as conclusões do trabalho e sugere trabalhos futuros relacionados ao tópico.

## 2 Revisão bibliográfica

### 2.1 Estacionariedade

Nesta seção são apresentadas as definições de funções aleatórias e estacionariedade. Fundamentais ao pleno entendimento da necessidade da modelagem dos domínios geológicos de um depósito mineral sob o ponto de vista matemático. (MATHERON, 1965; JOURNEL; HUIJBREGTS, 1978; ARMSTRONG, 1998).

#### 2.1.1 Funções aleatórias

Um fenômeno natural pode muitas vezes ser caracterizado pela distribuição espacial de uma ou mais variáveis, chamadas variáveis regionalizadas (ReV).

Uma variável aleatória (RV), é uma variável que recebe um certo número de valores numéricos de acordo com uma determinada distribuição de probabilidade. O resultado do lançamento de um dado, por exemplo, pode ser considerado uma RV que pode receber um entre seis valores equiprováveis.

Um valor observado em um ponto amostral  $x_i$  é a realização,  $z(x_i)$  de uma variável regionalizada,  $Z(x_i)$ . A família de todas as variáveis aleatórias auto-correlacionadas  $Z(x)$  quando  $x$  varia por todo o domínio ( $D$ ) de um depósito é uma função aleatória (RF)  $\{Z(x), x \in D\}$ . Essa definição de função aleatória expressa o aspecto aleatório e estruturado das variáveis regionalizadas.

Localmente, no ponto  $x_i$ ,  $Z(x_i)$  é uma variável aleatória.  $Z(x)$  é uma RF, já que para cada par de pontos  $x_i$  e  $x_i + h$ , as RV correspondentes,  $Z(x_i)$  e  $Z(x_i + h)$  não são independentes, são relacionadas por uma correlação que expressa a estrutura espacial da variável regionalizada  $z(x)$ .

De acordo com Journel e Huijbregts (1978), o problema então, passa a ser representar a variabilidade da função aleatória no espaço (quando  $x$  varia), essa representação será usada para resolver problemas de estimativa do valor  $z(x_0)$  em um ponto  $x_0$  que não foi amostrado.

A interpretação probabilística da Variável regionalizada  $z(x)$  como sendo uma determinada realização de uma certa RF  $Z(x)$  faz sentido apenas quando é possível inferir, mesmo que parcialmente, a lei de probabilidade que define a RF.

É virtualmente impossível inferir a lei de probabilidade de uma RF  $Z(x)$  a partir de uma única realização  $z(x)$ , que por sua vez é limitada a um número finito de pontos amostrais  $x_i$ . Da mesma forma, é impossível determinar a lei de probabilidade que rege o

lançamento de um dado a partir do resultado de um único lançamento, vários lançamentos são necessários para sua determinação. Várias realizações  $z_1(x), z_2(x), \dots, z_k(x)$  da RF  $Z(x)$  seriam necessárias para inferir a lei de probabilidade de  $Z(x)$ , na prática estamos limitados a uma única realização  $\{z(x_i)\}$  da RF nas posições  $x_i$ , então, certas suposições envolvendo graus de homogeneidade espacial devem ser estabelecidas, a hipótese de estacionariedade.

Na prática, mesmo que apenas certa região do fenômeno possa ser considerada homogênea, a ReV se repete no espaço. Essa homogeneidade ou repetição é equivalente a várias realizações da mesma RF  $Z(x)$  e permite certo grau de inferência estatística. Dois pontos experimentais  $z(x_0)$  e  $z(x_0 + h)$  em dois pontos diferentes,  $x_0$  e  $x_0 + h$ , podem então, ser tratados como duas realizações da mesma RV  $Z(x_0)$ . Essa abordagem é usada para inferir a lei de distribuição da RF  $Z(x)$  a partir do histograma dos dados  $\{z(x_i)\}$ , assim como o valor esperado  $E\{Z(x)\}$  a partir da média aritmética dos dados.

## 2.1.2 Momentos

Considere a RF  $Z(x)$ . Para cada conjunto de  $k$  pontos no  $R^n$  (espaço n-dimensional)  $x_1, x_2, \dots, x_k$ , corresponde um componente vetorial k-dimensional de variáveis aleatórias  $\{Z(x_1), Z(x_2), \dots, Z(x_k)\}$ .

Essa RV vetorial é caracterizada pela função de distribuição (ccdf), condicionada aos  $k$  pontos amostrais  $F_{x_1, x_2, \dots, x_k}(z_1, z_2, \dots, z_k) = Prob\{Z(x_1) < z_1, \dots, Z(x_k) < z_k\}$ .

O conjunto de todas as funções de distribuição, para todos os  $k$  inteiros e positivos e para todas as escolhas possíveis de pontos no  $R^n$ , constitui a lei espacial da RF  $Z(x)$ .

Em aplicações na mineração, não é necessário caracterizar toda a lei espacial, apenas os dois primeiros momentos da lei são necessários para uma solução aproximada dos problemas encontrados. Além disso, a quantidade de dados disponível é insuficiente pra inferir a lei espacial em sua totalidade (JOURNAL; HUIJBREGTS, 1978).

Seguem as definições dos momentos segundo Journal e Huijbregts (1978)

### 2.1.2.1 Momento de primeira ordem

Considere a RV  $Z(x)$  no ponto  $x$ . Se a função de distribuição de  $Z(x)$  tem uma esperança matemática que depende do ponto  $x$ , essa é:

$$E\{Z(x)\} = m(x) \quad (2.1)$$

### 2.1.2.2 Momentos de segunda ordem

Na geoestatística, os momentos de segunda ordem são:

A variância, ou mais precisamente, variância *a priori* de  $Z(x)$ , é definida como o momento de segunda ordem em relação a esperança  $m(x)$  da RV  $Z(x)$ ,

$$\text{Var}\{Z(x)\} = E\{[Z(x) - m(x)]^2\} \quad (2.2)$$

A covariância. Pode ser demonstrado que se duas RV  $Z(x_1)$  e  $Z(x_2)$  têm variâncias nos pontos  $x_1$  e  $x_2$  elas também têm uma covariância, que é função das duas localidades  $x_1$  e  $x_2$ , e pode ser escrita como:

$$C(x_1, x_2) = E\{[Z(x_1) - m(x_1)][Z(x_2) - m(x_2)]\} \quad (2.3)$$

O variograma, que é definido como a variância do incremento  $[Z(x_1) - Z(x_2)]$ , pode ser escrito como:

$$2\gamma(x_1, x_2) = \text{Var}\{Z(x_1) - Z(x_2)\} \quad (2.4)$$

A função  $\gamma(x_1, x_2)$  é o semi-variograma.

### 2.1.3 Hipótese de estacionariedade

Por definição, funções variograma e covariância dependem simultaneamente de dois pontos  $x_1$  e  $x_2$ , então várias realizações do par de RV  $\{Z(x_1), Z(x_2)\}$  são necessários para que qualquer inferência estatística seja possível.

Por outro lado, se essas funções dependerem somente da distância entre dois pontos, isto é, do vetor  $h = x_1 - x_2$  que separa  $x_1$  e  $x_2$ , a inferência estatística se torna possível. Cada par de dados  $\{z(x_k), z(x_{k'})\}$  separados pela distância  $(x_k - x_{k'})$ , igual ao vetor  $h$ , pode ser considerado como uma realização diferente do par de RV  $\{Z(x_1), Z(x_2)\}$ .

É intuitivo perceber que em zonas de mineralização homogêneas (domínios geológicos), a correlação que existe entre dois pontos  $z(x_k)$  e  $z(x_{k'})$  não depende das posições dentro dessas zonas homogêneas, mas da distância que os separa.

[Journel e Huijbregts \(1978\)](#) apresentam diferentes tipos de estacionariedade.

#### 2.1.3.1 Estacionariedade estrita

Uma RF é dita estacionária estrita quando sua lei espacial é invariante à translação. As duas componentes vetoriais  $\{Z(x_1), Z(x_k)\}$  e  $\{Z(x_1 + h), Z(x_k + h)\}$  têm a mesma lei de distribuição, independente do vetor de translação  $h$ .

Na geoestatística linear, estamos interessados apenas nos dois primeiros momentos, então será necessário assumir a estacionariedade apenas para eles.

### 2.1.3.2 Estacionariedade segunda ordem

Uma RF é dita estacionária de segunda ordem quando sua esperança matemática existe e não depende do ponto  $x$ .

$$E\{Z(x)\} = m, \quad \forall x \quad (2.5)$$

A média de  $Z(x)$  é a média do domínio geológico estacionário.

Para cada par de RV  $\{Z(x), Z(x+h)\}$ , a covariância existe e depende da distância de separação  $h$ .

$$C(h) = E\{Z(x_h)Z(x)\} - m^2, \quad \forall x \quad (2.6)$$

$h$  representa um vetor de coordenadas no espaço tridimensional.

Sob a hipótese de estacionariedade de segunda ordem  $\gamma(h) = C(0) - C(h)$ .

### 2.1.3.3 Hipótese intrínseca

Uma RF  $Z(x)$  é dita intrínseca quando sua esperança matemática existe e não depende do ponto  $x$ .

$$E\{Z(x)\} = m, \quad \forall x \quad (2.7)$$

A média de  $Z(x)$  é a média do domínio geológico estacionário.

Para todos os vetores  $h$  o incremento  $[Z(x+h) - Z(x)]$  tem variância finita que não depende  $x$

$$Var\{Z(x+h) - Z(x)\} = E\{[Z(x+h) - Z(x)]^2\} = 2\gamma(h), \quad \forall x \quad (2.8)$$

Então, a estacionariedade de segunda ordem implica na hipótese intrínseca, mas o contrário não é verdadeiro. A hipótese intrínseca pode ser vista como uma limitação da estacionariedade de segunda ordem aos incrementos da RF  $Z(x)$

### 2.1.3.4 Quasi-estacionariedade

Na prática, a covariância ou variograma são usados apenas para distâncias limitadas  $|h| \leq b$ , como por exemplo,  $b$  poderia ser o diâmetro da vizinhança de busca, em outros casos  $b$  pode ser uma zona de mineralização homogênea (domínio geológico) e duas variáveis  $Z(x_k)$  e  $Z(x_k+h)$  não podem ser consideradas provenientes do mesmo domínio se  $|h| > b$ .

Nesses casos, devemos considerar a função  $C(x, x + h)$  ou  $\gamma(x, x + h)$  não mais do que localmente estacionárias, para distâncias  $|h|$  menores que  $b$ . Essa limitação da hipótese de segunda ordem é a hipótese de quasi-estacionariedade.

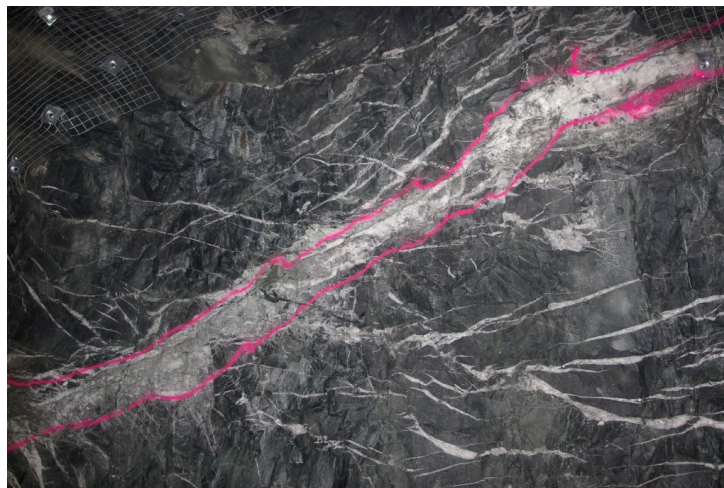
#### 2.1.4 Decisão de estacionariedade

A decisão de estacionariedade é uma suposição, e requisito a aplicação das técnicas geostatísticas de estimativa e simulação. Estacionariedade não é uma propriedade geológica, é uma propriedade matemática.

Uma função aleatória estacionária (SRF) é a representação probabilística de uma propriedade petrofísica com esperança matemática e covariância constantes independente da localização. Raramente, é apropriado considerar a totalidade do depósito mineral como um único domínio estacionário, que será modelado a partir de uma SRF. Geralmente, é necessário identificar diferentes domínios no depósito, cada um com seu modelo de SRF, consistentes com as suposições matemáticas de uma SRF (MCLENNAN, 2007).

A Figura 3 mostra um veio mineralizado de quartzo-ouro em rocha encaixante máfica, os teores de ouro se comportam de maneira diferente no veio e rocha encaixante, os teores são mais altos e mais variáveis no veio. Assumir estacionariedade, e modelar com uma única SRF toda a região não seria adequado. O modelo numérico será mais consistente com a realidade se duas SRF forem usadas, uma para cada unidade geológica.

Figura 3 – Veio de quartzo-ouro em rocha máfica.



Fonte: <<http://rses.anu.edu.au/people/steve-cox>>, acesso em julho de 2016

Segundo McLennan (2007), para que a decisão de estacionariedade seja razoável um total de cinco fases devem ser consideradas:

1. Escolher o número e tipo dos domínios que terão suas propriedades petrofísicas modeladas;
2. Modelar as fronteiras desses domínios;
3. Caracterizar a natureza das transições entre os domínios e modelá-las;
4. Modelar as tendências (*trends*) dentro de cada domínio;
5. Estimar com um modelo de tendência.

A escolha dos domínios deve ser avaliada com base em um equilíbrio entre um modelo geológica e fisicamente consistentes e o número de amostras disponível para inferir os parâmetros das SRF (MCCLENNAN, 2007).

Além da imposição matemática, os domínios de um depósito mineral devem ser modelados para que estimativas mais precisas de volume/massa e teores sejam obtidas e problemas de diluição ocasionados pela falta de aderência do modelo de blocos às estruturas geológicas do depósito sejam reduzidos (RASERA, 2014).

## 2.2 Funções implícitas

Alguns dos métodos determinísticos de modelagem geológica disponíveis, e revisados nessa dissertação, são baseados em funções implícitas. O uso da modelagem implícita foi introduzido no campo da computação gráfica para criação de objetos de diversas geometrias e complexidades por Bloomenthal e Bajaj (1997). A ideia central da técnica é usar uma função implícita para demarcar regiões de diferentes formas e extensões no espaço (SILVA, 2015).

Na matemática, uma equação implícita é uma relação na forma  $R(x_1, \dots, x_n) = 0$ . Por exemplo, a equação implícita de um círculo unitário é  $x^2 + y^2 - 1 = 0$ .

Uma função implícita é uma função definida implicitamente por uma equação implícita, associando uma das variáveis (o valor) com as outras (os argumentos). A função implícita que define o círculo unitário pode ser escrita como  $x^2 + f(x)^2 - 1 = 0$ . Essa equação implícita define  $f$  como uma função de  $x$  apenas se  $-1 \leq x \leq 1$ .

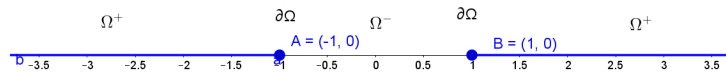
Os exemplos a seguir, que ilustram a metodologia, foram propostos por Osher e Fedkiw (2003).

Na Figura 4, em uma dimensão, a linha dos números reais foi dividida em três partes distintas usando os pontos  $x = -1$  e  $x = 1$ . Isto é, definimos  $(-\infty, -1)$ ,  $(-1, 1)$  e  $(1, \infty)$ . A primeira e terceira partes são peças desconexas da mesma região. Assim, definimos  $\Omega^- = (-1, 1)$  como parte interna do domínio e  $\Omega^+ = (-\infty, -1) \cup (1, \infty)$  como parte externa do domínio. A fronteira entre a parte interna e externa consiste dos dois



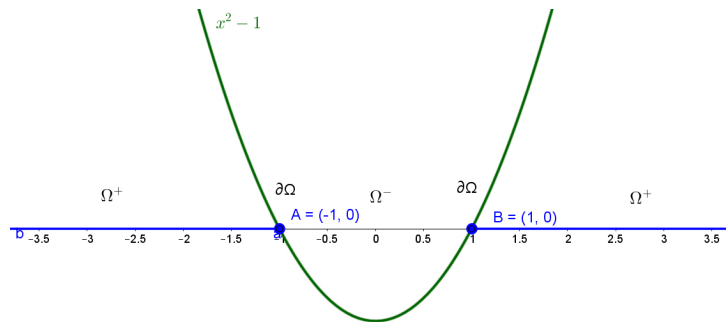
pontos  $\partial\Omega = \{-1, 1\}$ , e é chamada de interface. No  $\mathbb{R}^n$ , os subdomínios são  $n$ -dimensionais, enquanto a interface apresenta  $n - 1$  dimensões.

Figura 4 – Linha dos reais dividida em dois subdomínios pelos pontos  $x = -1$  e  $x = 1$ .



Numa representação explícita da interface, escrevemos explicitamente os pontos que a definem como feito anteriormente ao definir  $\partial\Omega = \{-1, 1\}$ . Também é possível definir implicitamente a interface como o isocontorno de alguma função. Por exemplo, o isocontorno zero de  $\phi(x) = x^2 - 1$  é o conjunto de pontos para os quais  $\phi(x) = 0$ , isto é, exatamente  $\partial\Omega = \{-1, 1\}$ , como pode ser observado na Figura 5.

Figura 5 – Função implícita definindo as regiões  $\Omega^-$  e  $\Omega^+$  bem como a fronteira  $\partial\Omega$ .



Modificado de Osher e Fedkiw (2003)

O isocontorno zero foi escolhido. Porém não há nada de especial nessa escolha, o isocontorno  $\phi(x) = 1$  da função  $\phi(x) = x^2$  define a mesma interface.

### 2.2.1 Funções distâncias assinaladas

Na seção 2.2, foram definidas as funções implícitas, seu valor é  $\phi(x) = 0$  na interface,  $\phi(x) < 0$  na região interna e  $\phi(x) > 0$  na região externa. As funções distância assinaladas são um subconjunto das funções implícitas.

Uma função distância  $d(\vec{x})$  é definida como:

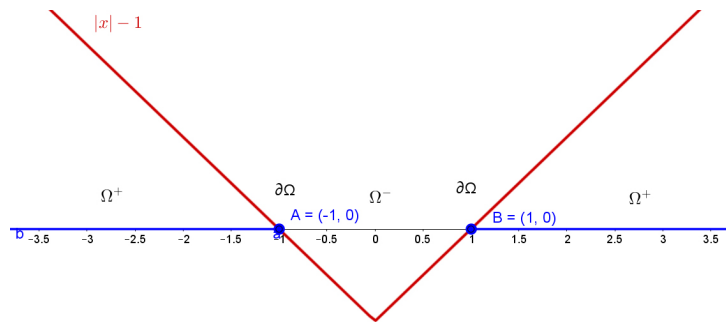
$$d(\vec{x}) = \min(|\vec{x} - \vec{x}_I|) \quad \forall \quad \vec{x}_I \in \partial\Omega \quad (2.9)$$

Onde  $\vec{x}_I$  é um ponto pertencente à interface. Ou seja, para um dado ponto  $\vec{x}$ ,  $d(\vec{x})$  é a menor distância entre  $\vec{x}$  e um ponto  $\vec{x}_C$  pertencente à interface  $\partial\Omega$ .

Uma função distância assinalada é uma função implícita  $\phi$ , com  $|\phi(\vec{x})| = d(\vec{x})$  para todo  $\vec{x}$ . Então,  $\phi(\vec{x}) = d(\vec{x}) = 0$  para todo  $\vec{x} \in \partial\Omega$ ,  $\phi(\vec{x}) = -d(\vec{x})$  para todo  $\vec{x} \in \Omega^-$  e  $\phi(\vec{x}) = d(\vec{x})$  para todo  $\vec{x} \in \Omega^+$ .

Na [seção 2.2](#), a função  $\phi(x) = x^2 - 1$  foi usada para representar implicitamente a interface  $\partial\Omega = \{-1, 1\}$ . Uma representação baseada em funções distância assinaladas da mesma interface é  $\phi(x) = |x| - 1$ , como visto na [Figura 6](#)

Figura 6 – Função distância assinalada definindo as regiões  $\Omega^-$  e  $\Omega^+$  bem como a fronteira  $\partial\Omega$ .

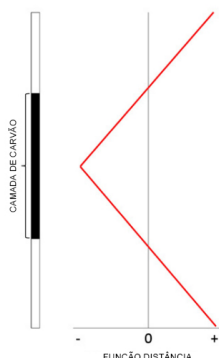


Modificado de [Osher e Fedkiw \(2003\)](#)

Contextualizando, considere o furo de sondagem vertical interceptando uma camada de carvão, à direita na [Figura 7](#), e o valor da função distância assinalada para essa litologia à esquerda. O valor da função decresce à medida que um ponto ao longo do furo se aproxima da camada, e seu valor é a menor distância desse ponto até um ponto pertencente à interface. Vale zero na interface, e passa a ser negativo para pontos no interior da camada, atingindo o menor valor no centro, parte mais interna da camada de carvão.

Os exemplos apresentados são casos simplórios baseados em funções paramétricas e não são aplicáveis a características geológicas, embora a ideia central seja a mesma, algumas adaptações devem ser feitas para o emprego da técnica sob uma perspectiva geológica. A [seção 3.1](#) descreve como as funções implícitas devem ser construídas e aplicadas.

Figura 7 – Furo de sonda interceptando uma camada de carvão, à esquerda, e o valor da função distancia assinalada à direita.



Modificado de [Deutsch e Wilde \(2013\)](#)

## 2.3 Método implementado no software Leapfrog<sup>®</sup>

O assunto foi introduzido no campo da geologia por [Cowan et al. \(2002\)](#), [Cowan et al. \(2003\)](#), baseado no trabalho de [Savchenko et al. \(1995\)](#) para modelar objetos através da interpolação de uma função implícita.

O Leapfrog<sup>®</sup> permite a construção de modelos, geológicos e/ou de iso teores (*grade shells*), tridimensionais a partir dos dados brutos de sondagem em questão de minutos ou horas, contra dias ou até semanas gastos com o método tradicional. Desse modo, diferentes cenários geológicos podem ser testados. Os modelos são gerados de forma semi automática. Ademais, informação geológica qualitativa pode ser incorporada ao modelo de forma prática e rápida ([COWAN et al., 2002](#)).

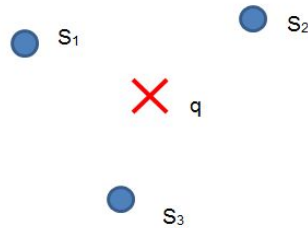
### 2.3.1 Metodologia

Corpos geológicos são definidos implicitamente a partir dos dados de litologia provenientes da sondagem. O valor de uma função implícita (função distância assinalada ou função volume) é calculado para os pontos amostrais e interpolado para todo o espaço. Então, o volume que representa o corpo geológico é definido por uma iso superfície (geralmente a iso superfície zero) da função implícita interpolada ([COWAN et al., 2003](#)). A metodologia para construção da função distância assinalada é elucidada na [seção 3.1](#). Os corpos geológicos são modelados individualmente, em ordem geocronológica, do mais recente para o mais antigo, evitando a sobreposição de domínios. O Leapfrog<sup>®</sup> trabalha convertendo uma função implícita em uma superfície triangulada (*mesh*) que representa a unidade geológica.

### 2.3.1.1 Interpolação

O exemplo a seguir, que ilustra a metodologia de interpolação implementada no *software*, foi apresentado por Lane (2013).

Figura 8 – Três amostras  $s_1$ ,  $s_2$  e  $s_3$  e o um ponto  $q$  a ser estimado.



Fonte: Lane (2013)

A Figura 8 ilustra o problema fundamental de estimar o valor  $q$  a partir de três amostras  $s_1$ ,  $s_2$  e  $s_3$ .

A krigagem, em sua forma mais simples, estima o valor em um ponto como uma combinação linear das amostras conhecidas, os pesos são determinados matematicamente pela distribuição das amostras em relação ao ponto a ser estimado. Considerando o exemplo anterior:

$$q = w_1 s_1 + w_2 s_2 + w_3 s_3 \quad (2.10)$$

Onde  $w_1$ ,  $w_2$  e  $w_3$  são os pesos atribuídos às amostras.

Fica evidente que a krigagem pode ser usada como forma de interpolação, já que é possível estimar um valor em qualquer posição arbitrária onde não há dados. Todavia, para cada nova estimativa, é necessário recalcular os pesos, um processo que toma esforço computacional e tempo.

É possível rearranjar a equação de krigagem da seguinte forma:

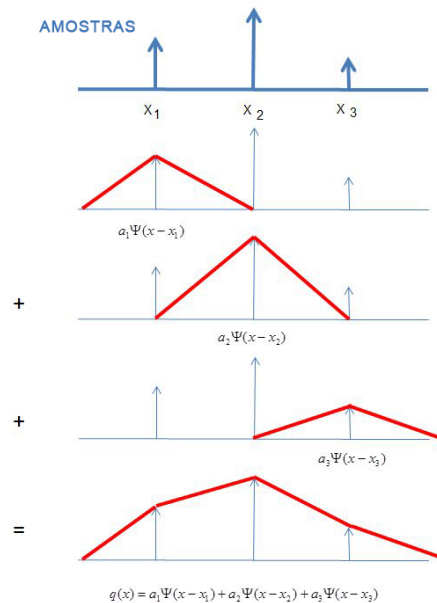
$$q(x) = a_1 \psi(x - x_1) + a_2 \psi(x - x_2) + a_3 \psi(x - x_3) \quad (2.11)$$

Esse método é conhecido como *dual kriging* (GALLI; MURILLO; THOMANN, 1984).  $a_1 \psi(x - x_1)$  descreve como a amostra localizada na posição  $x_1$  influencia a estimativa na sua vizinhança,  $\psi(x)$  é equivalente ao variograma (ou interpolante no casos das RBF). Nessa abordagem, as influências ponderadas de cada amostra são somadas para produzir uma equação que possa ser usada para estimar um valor em qualquer local  $x$ . Uma

estimativa feita dessa maneira produz o mesmo resultado do que o método tradicional de krigagem.

A [Figura 9](#) ilustra um exemplo unidimensional do método

Figura 9 – Demonstração simplificada do funcionamento das funções de bases radiais em uma dimensão.



Modificada de [Lane \(2013\)](#)

Utilizando a metodologia tradicional de krigagem, para cada nova estimativa em um local diferente é preciso recalculer os pesos, já com a *dual kriging*, onde a influência das amostras é somada, os pesos só precisam ser calculados uma única vez. Acelerando o processo.

O *Leapfrog*® usa funções de bases radiais (RBF) para interpolar a função distância assinalada retendo todas as amostras disponíveis (estimador global) em todas as estimativas. Mais especificamente *FastRBF*™, cuja diferença principal em relação à RBF tradicional, é a capacidade de trabalhar com bancos de dados volumosos (mais de 1.000.000 de amostras) em um hardware convencional de forma relativamente rápida ([LEAPFROG3D, 2015](#)).

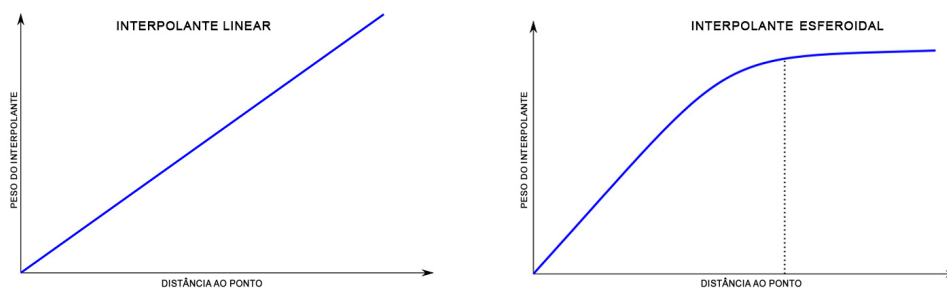
RBF são uma forma de implementar a *dual kriging* ([LANE, 2013](#)). Enquanto a krigagem usa uma função covariância obtida através dos dados para determinar os pesos dados às amostras, as RBF usam funções básicas (*interpolant functions*) pré-definidas para determiná-la.

Duas funções são usadas para derivar os pesos dados à cada amostra no *Leapfrog*®, o interpolante linear ([Figura 10](#) à esquerda) e o interpolante esferoidal ([Figura 10](#) à

direita). Equivalentes aos modelos de variograma potência e esférico, respectivamente Spragg (2013b).

O interpolante linear é multi escala, o que o torna bastante flexível e de uso geral. Particularmente recomendado para casos em que há um grande número de dados de litologia concentrados em bolsões localizados. Não é recomendado para teores, já que extrapola violentamente para além do limite dos dados. O interpolante esferoidal contorna o problema de extrapolação, muitas vezes não realista, do modelo linear (MCLENNAN, 2013). Os parâmetros do modelo como patamar, alcance e efeito pepita podem ser ajustados.

Figura 10 – Interpolantes do *Leapfrog*<sup>®</sup>.



Modificada de Spragg (2013b)

### 2.3.1.2 Anisotropia

O *Leapfrog*<sup>®</sup> conta com ferramentas que dão ao usuário controle sobre a continuidade dos corpos geológicos em determinadas direções com possibilidade de incorporar à interpolação uma tendência global e/ou uma tendência estrutural.

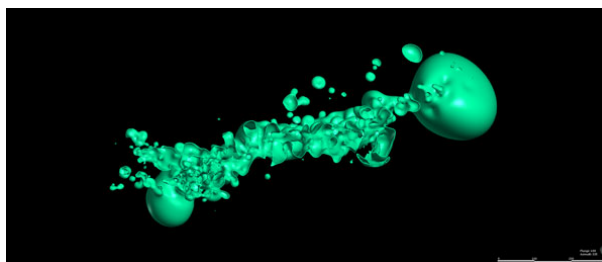
A tendência global controla o comportamento do modelo interpolante, e deve ser utilizado em situações onde a geologia sugere continuidade do corpo numa direção planar sobre uma grande área, se a geologia sugerir que a direção dos corpos varia no espaço, a tendência estrutural é mais indicada (SPRAGG, 2013a).

A tendência global é inserida ao modelo a partir de um elipsoide de anisotropia, de forma análoga aos variogramas anisotrópicos da geoestatística.

A tendência estrutural é uma generalização da tendência global que permite alterar a direção de continuidade ao longo de uma superfície definida pelo usuário, ao invés de ser baseada em um plano, como a tendência global, definida a partir de um elipsoide de anisotropia. A tendência estrutural é baseada em uma superfície que pode apresentar qualquer forma e orientação, e é condicionada, geralmente, por restrições geológicas, como falhas, dobras, foliações... Para cada ponto da superfície (*mesh*) definida, a tendência local é determinada, produzindo uma anisotropia que varia por todo o espaço definido (JOYNT, 2014).

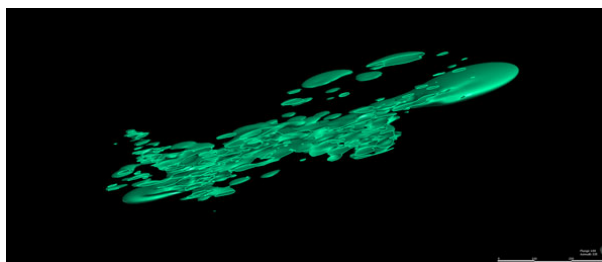
A tendência estrutural não determina a forma final da unidade geológica, ela é determinada pelo interpolante e pelas amostras retidas na interpolação. A Figura 11 mostra uma superfície interpolada usando um interpolante isotrópico, a Figura 12 um interpolante com tendência global e a Figura 13 um interpolante com tendência estrutural. Todos os sólidos derivam da mesma base de dados.

Figura 11 – Interpolante isotrópico.



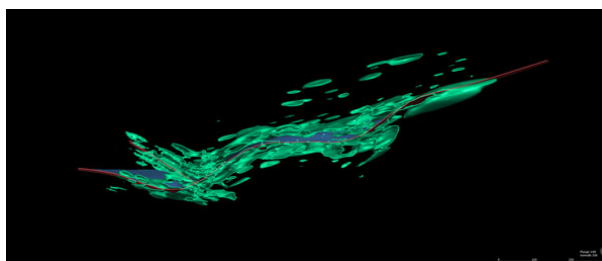
Fonte: Joynt (2014)

Figura 12 – Tendência global aplicada ao interpolante.



Fonte: Joynt (2014)

Figura 13 – Tendência estrutural aplicada ao interpolante.



Fonte: Joynt (2014)

O Leapfrog® disponibiliza três tipos de tendência estrutural (*non-decaying trends*, *blended trends* e *strongest along meshes*), que compartilham os mesmos parâmetros força

(*strength*) e alcance (*range*). A força é semelhante à razão do elipsoide e dita quanta alongação existirá em uma determinada direção. Enquanto o alcance dita até que distância da superfície definida (*mesh*) a tendência ainda tem efeito (JOYNT, 2014).

### 2.3.1.3 Visão geral

A metodologia do *Leapfrog*<sup>®</sup> compreende cinco passos (COWAN et al., 2002):

1. Validação dos dados e compositagem: Esse passo pode ser executado na maior parte dos software de mineração disponíveis;
2. Interpolação e *meshing*: Extrair iso superfícies de uma função distância assinalada interpolada;
3. Incorporar geomorfologia: estruturas geológicas interpretadas pelo geomodelador são inseridas no modelo como tendência estrutural;
4. Interpolação da informação morfológica: Uma função de interpolação é ajustada à informação digitalizada no passo anterior;
5. Interpolação condicionada à informação geomorfológica: A interpolação do segundo passo é repetida, dessa vez, condicionada pela função de interpolação da informação morfológica gerada no quarto passo. Essa etapa gera modelos que honram os dados amostrais, e são consistentes com a informação qualitativa interpretada pelo geomodelador.

A metodologia implementada no *leapfrog*<sup>®</sup> supera muitas das limitações dos métodos explícitos. Os domínios são construídos de forma rápida, o método é objetivo, replicável e nova informação geológica pode ser incorporada facilmente aos modelos. O *software* conta com funções interessantes como a possibilidade de inserir no modelo informação geológica qualitativa e ferramentas específicas para modelar estruturas geológicas complexas como diques. No entanto, o fato da RBF ser um interpolador global pode inviabilizar o uso do método em bancos de dados muito grandes, o método não trabalha com múltiplos domínios de forma direta e sua licença é bastante cara (MCLENNAN; DEUTSCH, 2006).

## 2.4 Método dos campos potenciais

O método dos campos potenciais, implementado no *software Isatis*<sup>®</sup>, foi criado para construir modelos geológicos tridimensionais a partir de dados de geologia e de exploração (CHILÈS et al., 2004). O princípio do método é derivar a geometria dos domínios a partir da interpolação tridimensional de um campo escalar, conhecido como campo potencial. O



campo potencial é construído através da cokrigagem da informação dos contatos geológicos provenientes de furos de sondagem e dados estruturais (RENARD et al., 2013).

A localização dos contatos estabelece a posição de referência de iso valores do campo potencial enquanto os dados de orientação são o gradiente da função escalar (campo potencial). A geometria dos corpos geológicos é obtida a partir da discretização de iso valores de referência (CALCAGNO et al., 2008), semelhante a como uma unidade geológica é definida a partir de uma iso superfície de uma função implícita nos métodos apresentados na seção 2.3 e seção 3.1.

A covariância da cokrigagem e o gradiente do campo potencial estimado podem ser transformados em medidas de incerteza da localização da fronteira dos domínios ou usadas para mapear a probabilidade de um local não amostrado pertencer a um determinado domínio geológico (RENARD et al., 2013).

### 2.4.1 Metodologia

O método consiste em modelar uma interface geológica ou uma série de interfaces sub paralelas  $l_k$ ,  $k = 1, 2, \dots$ . Seu princípio é sumarizar a geologia a partir de um campo potencial, isto é, uma função escalar  $T(p)$  de qualquer ponto  $p = (x, y, z)$  no espaço tridimensional, a interface  $l_k$  corresponde a uma superfície iso potencial, ou seja, o conjunto de pontos  $p$  que satisfazem  $T(p) = t_k$  para algum valor desconhecido  $t_k$  do campo potencial. De forma análoga, a formação geológica entre duas interfaces sucessivas  $l_k$  e  $l_{k'}$  é definida por todos os pontos  $p$  os quais o campo potencial pertence ao intervalo definido por  $t_k$  e  $t_{k'}$  (CHILÈS et al., 2004).

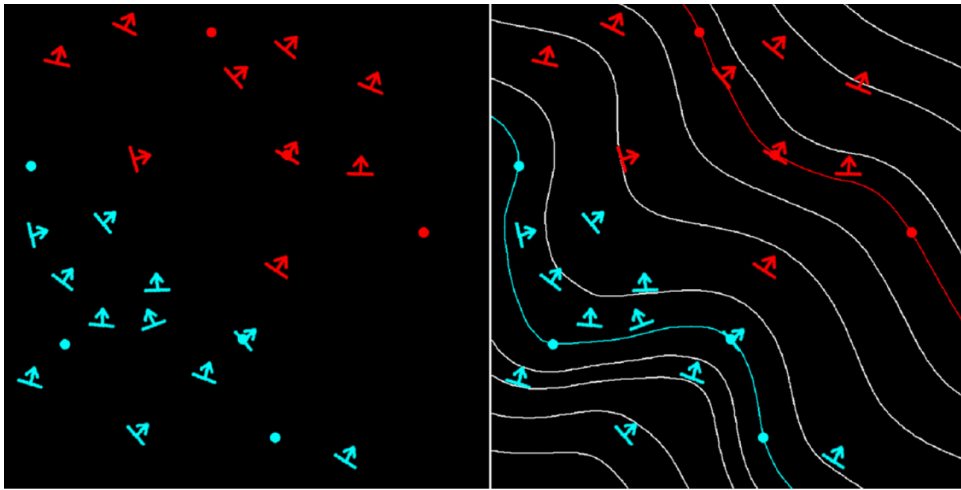
A Figura 14 ilustra o princípio em duas dimensões. À esquerda, uma formação geológica foi mapeada pela posição de suas interfaces com outras duas formações (pontos vermelhos e azuis) e medidas do mergulho (setas). À direita, a formação geológica foi modelada pelo método dos campos potenciais, curvas vermelhas e azuis representam iso valores de referência dos contatos modelados. Curvas brancas são iso valores selecionados do campo potencial, e representam tendências geológicas ou trajetórias de foliações. As interfaces honram tanto os pontos pertencentes aos contatos quanto os vetores de orientação (CALCAGNO et al., 2008).

### 2.4.2 Tipos de dados

$T(p)$  é modelada com dois tipos diferentes de dados (CHILÈS et al., 2004):

1. Coordenadas tridimensionais de pontos que pertencem às interfaces  $l_1, l_2, \dots$  (contatos geológicos);

Figura 14 – Princípio do método dos campos potenciais em duas dimensões.



Fonte: Calcagno et al. (2008)

2. Vetores unitários tridimensionais que pertencem ao campo de orientação das interfaces.

Na modelagem tridimensional de interfaces sedimentares, a informação estrutural consiste em vetores unitários ortogonais ao principal plano de anisotropia (acamamento, xistosidade, foliação) e apontando no sentido das formações mais jovens. Na modelagem de um domínio mineralizado, os vetores unitários apontam do interior para o exterior dos domínios (RENARD et al., 2013).

#### 2.4.2.1 Codificação dos dados

Para interpolar o campo potencial, os dados devem ser codificados da seguinte forma (CHILÈS et al., 2004; CALCAGNO et al., 2008; RENARD et al., 2013):

1. Pontos da interface: O valor potencial  $c$  associado com a interface não é conhecido, mas a diferença potencial entre dois pontos de uma interface é igual a zero (já que as superfícies são equipotenciais). Os  $m + 1$  pontos  $p_0, p_1, \dots, p_m$  da interface são transladados em  $m$  incrementos  $T(p_\alpha) - T(p'_\alpha) = 0$ ,  $\alpha = 1, \dots, m$ .  $p'_\alpha$  pode ser tanto o ponto  $p_0$  ou o ponto  $p_{\alpha-1}$  (a escolha não tem impacto no resultado);
2. Dados de orientação: Os vetores unitários normais a cada plano estrutural, são considerados como gradiente do campo potencial, equivalentes ao valor da derivada parcial  $\frac{\partial T(p)}{\partial u}$  no ponto medido  $p_\beta$  em qualquer direção  $u$ . O dado estrutural em  $p_\beta$  pode ser escrito como três derivadas parciais da forma  $\frac{\partial T(p_\beta)}{\partial u_\beta}$ ,  $u_\beta$ , representando os eixos de coordenadas  $x, y, z$  (um outro conjunto de derivadas parciais em três

direções ortogonais também pode ser considerado. Por exemplo, o vetor unitário pode ser expressado por uma derivada parcial igual a um na direção do vetor, e duas derivadas parciais iguais a zero nas direções que definem o plano estrutural).

#### 2.4.2.2 Interpolação do campo potencial

A função potencial  $T(\cdot)$  é definida até uma constante arbitrária que é o potencial em um ponto de referência arbitrário  $p_0$ . Em cada ponto  $p$  o incremento do campo potencial  $T(p) - T(p_0)$  é krigado. O estimador de cokrigagem é escrito como:

$$T^*(p) - T^*(p_0) = \sum_{\alpha=1}^M \mu_{\alpha} (T(p_{\alpha}) - T(p'_{\alpha})) + \sum_{\beta=1}^N \nu_{\beta} \frac{\partial T(p_{\beta})}{\partial u_{\beta}} \quad (2.12)$$

Onde os pesos  $\mu_{\alpha}$  e  $\nu_{\beta}$ , determinados pelo sistema de cokrigagem, são função de  $p$  (e  $p_0$ ). Embora a contribuição dos incrementos potenciais seja zero, os pesos  $\nu_{\beta}$  são diferentes dos pesos baseados apenas nos dados de gradiente.

A cokrigagem é estabelecida sob o modelo de função aleatória (subseção 2.1.1).  $T(\cdot)$  é assumido como uma função aleatória com deriva (*drift*) polinomial  $m(p)$ :

$$m(p) = \sum_{l=0}^L b_l f^l(p) \quad (2.13)$$

E uma variância estacionária  $K(h)$  (CALCAGNO et al., 2008; RENARD et al., 2013). Uma vez que as funções básicas  $f^l(p)$  da deriva e a covariância  $K(h)$  de  $T(\cdot)$  sejam conhecidas, a cokrigagem, na presença de dados do gradiente, pode ser calculada. Já que a derivada de  $\frac{\partial T(p)}{\partial u}$  vale  $\frac{\partial m(p)}{\partial u}$ , isto é, uma combinação linear das derivadas parciais  $\frac{\partial f^l(p)}{\partial u}$  com os mesmos coeficientes desconhecidos  $b_l$  de  $m(p)$ . As covariâncias das derivadas parciais são derivadas parciais de segunda ordem de  $K(\cdot)$ , e as covariâncias cruzadas entre campo potencial e as derivadas parciais são derivadas parciais de  $K(\cdot)$  (CHILÈS et al., 2004).

#### 2.4.2.3 Implementação do algoritmo de cokrigagem

O incremento dos dados potenciais não contribuem com as estimativas finais. O estimador pode ser visto como integração dos dados de gradiente (CHILÈS et al., 2004). Para garantir a continuidade espacial do estimador, é usada apenas uma vizinhança incluindo todos os dados. Quando não há interesse na variância, a cokrigagem é implementada em sua forma *dual* (*dual kriging*), assim, o sistema de krigagem é resolvido uma única vez.

$T^*(p) - T^*(p_0)$  pode ser obtido em qualquer ponto. Seu sinal indica onde  $p$  está localizado, negativo para formações mais antigas, positivo para formações mais recentes e zero para a interface. Essa propriedade permite a visualização em três dimensões do modelo

geológico a partir de um algoritmo de cubos marchantes (*marching cubes*) (LORENSEN; CLINE, 1987), que rastreia a iso superfície do campo potencial (CALCAGNO et al., 2008).

#### 2.4.2.4 Variância do campo potencial

Na geoestatística, a covariância ou variograma de uma variável são modeladas a partir dos dados. Porém, não há medidas do potencial  $T(x)$ , e os incrementos potenciais usados na interpolação não podem ser usados na inferência de  $K$ , já que todos valem zero.

Na sua primeira implementação, o algoritmo era baseado em um modelo de covariância arbitrariamente escolhido pelo usuário (CHILÈS et al., 2004).

Quando existem muitos dados estruturais disponíveis, o variograma das derivadas parciais pode ser calculado. Os variogramas das derivadas parciais são derivadas de segunda ordem do variograma do campo potencial. Renard et al. (2013) recomenda que uma soma de modelos anisotrópicos cúbicos de covariância sejam adotados para modelar a covariância.

#### 2.4.2.5 Incerteza nos modelos

Uma formação geológica definida pelo conjunto de pontos  $p$  tais que  $T(p) - T(p_0)$  está compreendido entre os dois valores  $t$  e  $t'$ . Sob a suposição de que o campo potencial é uma função aleatória gaussiana, a probabilidade de um dado ponto  $p$  pertencer à essa formação é:

$$P\{t \leq T(p) - T(p_0) < t'\} = G\left(\frac{t' - (T^*(p) - T^*(p_0))}{\sigma_{CK}(p)}\right) - G\left(\frac{t - (T^*(p) - T^*(p_0))}{\sigma_{CK}(p)}\right) \quad (2.14)$$

Onde  $G$  é a função de distribuição acumulada normal.

De forma análoga, para a interface que passa por  $p_0$

$$R(p) = T(p) - T(p_0) = \frac{T^*(p) - T^*(p_0)}{\sigma_{CK}} \quad (2.15)$$

Mede a probabilidade de  $p$  pertencer à interface.

#### 2.4.2.6 Visão geral

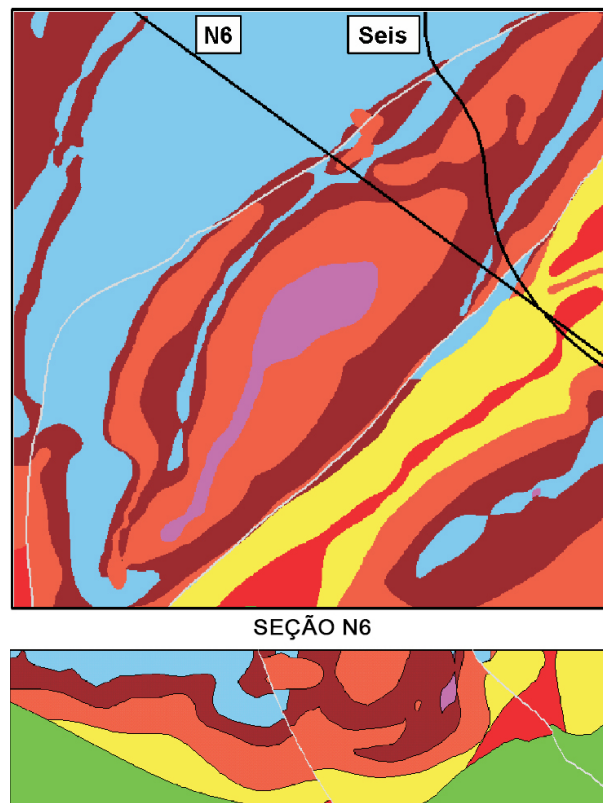
A metodologia compreende cinco passos (MCLENNAN; DEUTSCH, 2006):

1. Coletar dados de interseção e dados de orientação em superfície;
2. Determinar a forma da deriva (*drift*) localmente variável;
3. Inferir a função covariância do campo potencial;

4. Interpolar o campo potencial com cokrigagem;
5. Visualizar a incerteza na localização das fronteiras estimadas.

O método ainda permite a incorporação de falhas e informação geofísica. É um método flexível, que incorpora uma gama de dados diferentes na modelagem dos domínios geológicos. É rápido, em comparação com os métodos explícitos, é objetivo e replicável. Capaz de produzir modelos bastante complexos como o da [Figura 15](#). Porém, o procedimento não é simples. É difícil inferir a covariância dos campos potenciais já que não existem dados amostrais do campo. As medições de orientações nem sempre existem ou são de fácil obtenção. O método está implementado apenas em software pago.

Figura 15 – Modelo geológico criado a partir do método dos campos potenciais, em planta (acima). Seção N6 (abaixo).



Fonte: [Chilès et al. \(2004\)](#)



## 3 Modelagem geológica implícita com funções distância assinaladas

O método é baseado na interpolação de uma função implícita calculada para cada amostra e posterior definição de uma superfície de iso valores da função interpolada. A cada amostra é atribuída a distância euclideana anisotrópica entre ela mesma e a amostra mais próxima que pertence a um outro domínio geológico. O sinal da distância calculada revela se a amostra se encontra no interior (sinal negativo) ou exterior (sinal positivo) do domínio a ser modelado. A interpolação dessas medidas de distância permite a construção de um modelo binário de domínio geológico aplicando uma regra de corte (valor zero) na função implícita interpolada (SILVA; DEUTSCH, 2012).

Ao contrário dos métodos visitados nas seção 2.3 e seção 2.4, que permitiam a modelagem de um domínio por vez, esse método permite modelar múltiplos domínios simultaneamente. Sendo assim, a categoria correspondente a menor distância estimada deve ser retida.

### 3.1 Metodologia

#### 3.1.1 Caso binário

Primeiramente, o conjunto de dados  $z(u_\alpha)$ ,  $\alpha = 1, \dots, n$  é codificado em indicadores binários de acordo com a Equação 3.1, especificando se a amostra pertence ou não ao domínio.

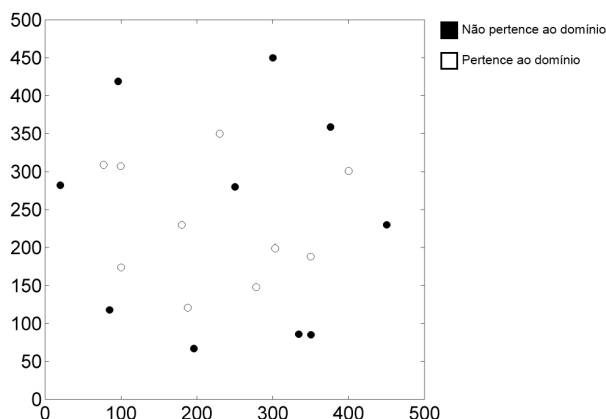
$$i(u_\alpha) = \begin{cases} 1, & \text{se } z(u_\alpha) \text{ se pertence ao domínio} \\ 0, & \text{se } z(u_\alpha) \text{ caso contrário} \end{cases} \quad (3.1)$$

Silva (2015) apresentou um exemplo simples em duas dimensões para ilustrar o método. Na Figura 16 os pontos amostrais foram codificados em indicadores.

Então, as distâncias assinaladas são calculadas para cada amostra de acordo com a Equação 3.2. Se a amostra pertence ao domínio modelado a distância é negativa, caso contrário positiva.

$$d(u_\alpha) = \begin{cases} - \| u_\alpha - u_\beta \|, & \text{se } i(u_\alpha) = 1 \\ + \| u_\alpha - u_\beta \|, & \text{se } i(u_\alpha) = 0 \end{cases} \quad (3.2)$$

Figura 16 – Transformação do banco de dados categóricos em indicadores binários.

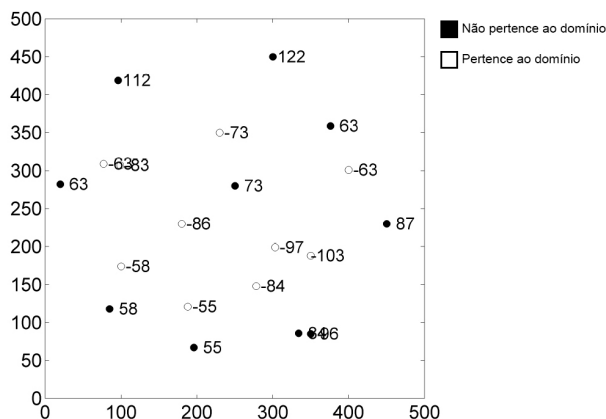


Fonte: Modificado de [Silva \(2015\)](#)

O local  $u_\beta$  corresponde à amostra mais próxima que pertença a um domínio diferente de  $u_\alpha$ . A regra euclideana é usada no cálculo.

A [Figura 17](#) mostra as distâncias assinaladas calculadas para cada ponto amostral.

Figura 17 – Função distância assinalada calculada para cada ponto amostral.



Fonte: Modificado de [Silva \(2015\)](#)

Nos casos em que há o conhecimento de que o corpo geológico se estende mais em uma direção do que nas outras (como os corpos tabulares, por exemplo) as distâncias calculadas podem ser anisotrópicas. Assim sendo, as coordenadas originais  $x$ ,  $y$  e  $z$  dos dados devem ser rotacionadas e/ou contraídas/dilatadas, a partir da transformação da [Equação 3.3](#). Então, as distâncias euclidianas ([Equação 3.2](#)) são calculadas normalmente



para as novas coordenadas  $x''$ ,  $y''$ ,  $z''$ .

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{max}} & 0 & 0 \\ 0 & \frac{1}{a_{min}} & 0 \\ 0 & 0 & \frac{1}{a_{vert}} \end{bmatrix} \begin{bmatrix} \cos \alpha \cos \phi - \sin \alpha \sin \beta \sin \phi & -\sin \alpha \cos \phi - \cos \alpha \sin \beta \sin \phi & \cos \beta \sin \phi \\ \sin \alpha \cos \beta & \cos \alpha \cos \beta & \sin \beta \\ -\cos \alpha \sin \phi - \sin \alpha \sin \beta \cos \phi & \sin \alpha \sin \phi - \cos \alpha \sin \beta \cos \phi & \cos \beta \cos \phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.3)$$

Onde  $a_{max}$ ,  $a_{min}$  e  $a_{vert}$  são as dimensões dos eixos máximo, médio e mínimo de anisotropia, e  $\alpha$ ,  $\beta$  e  $\phi$ , os ângulos de azimute, mergulho e *rake*, respectivamente.

A construção da função implícita em cada ponto amostral é a mesma para o método visitado na [seção 2.3](#).

A função distância calculada para cada amostra é então interpolada para todos os pontos de interesse, a partir da [Equação 3.4](#). Qualquer técnica de interpolação pode ser utilizada. Inverso da distância e krigagem produzem limites realísticos para os domínios. Porém, a krigagem permite levar em consideração a anisotropia e continuidade espacial das distâncias calculadas ([SILVA; DEUTSCH, 2012](#)).

[Silva e Deutsch \(2012\)](#), [Deutsch e Wilde \(2013\)](#), [Silva \(2015\)](#) recomendam que a krigagem global ([NEUFELD; WILDE, 2005](#)), um interpolador suave que retém todas as amostras nas estimativas (não há estratégia de busca), seja utilizada. Desse modo, evita-se o surgimento de artefatos nos mapas gerados. Todavia, reter todas as amostras na interpolação pode aumentar consideravelmente o tempo de execução do algoritmo. Testes realizados no [Capítulo 4](#) mostraram que o uso da krigagem ordinária com uma vizinhança de busca e número de amostras retidas suficientemente grandes, produzem um resultado muito semelhante à krigagem global.

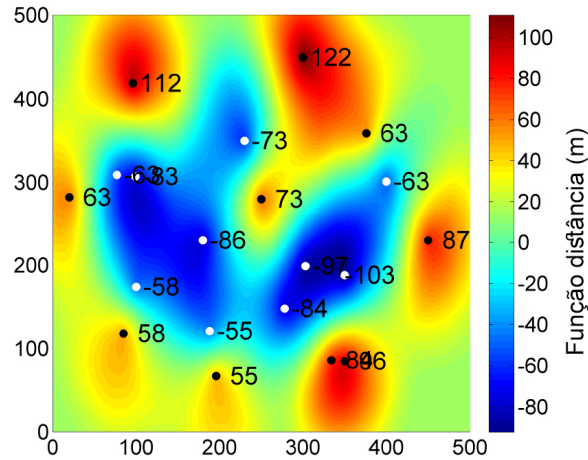
$$d^*(u) = \sum_{\alpha=1}^n \lambda_{\alpha}^{OK}(u) d(u_{\alpha}) \quad (3.4)$$

Onde  $d^*(u)$  é a distância estimada para cada local não amostrado  $u$ ,  $\lambda_{\alpha}^{OK}(u)$  os pesos de krigagem ordinária e  $d(u_{\alpha})$  o valor da função distância assinalada calculado para cada amostra.

A [Figura 18](#) mostra a função distância interpolada por krigagem ordinária em todos os pontos de interesse.

Por fim, cada local de interesse é classificado com base no sinal da função distância interpolada, de acordo com a [Equação 3.5](#). Blocos em que a função tem valor negativo, são

Figura 18 – Função distância interpolada para todos os locais de interesse.



Fonte: Modificado de [Silva \(2015\)](#)

classificados como pertencentes ao domínio. Blocos em que a função tem valor positivo, classificados como não pertencentes ao domínio.

Como as funções distância são negativas no interior do domínio e positivas no exterior, um bom palpite inicial para a interface que separa os domínios no espaço, seria a linha (em duas dimensões) ou superfície (em três dimensões) que corresponda ao valor zero da função distância assinalada ([WILDE; DEUTSCH, 2011](#)).

$$i^*(u) = \begin{cases} 1, & \text{se } d^*(u) \leq 0 \\ 0, & \text{caso contrário} \end{cases} \quad (3.5)$$

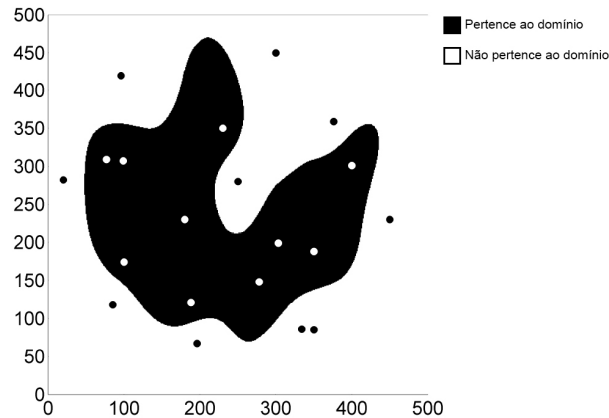
A [Figura 19](#) mostra o modelo geológico criado para o domínio a partir do sinal da função distância interpolada ([Figura 18](#)).

### 3.1.2 Aplicação da metodologia para múltiplos domínios geológicos

A metodologia apresentada na [subseção 3.1.1](#) é adequada apenas para casos binários, na presença de múltiplas categorias [Silva e Deutsch \(2012\)](#) propuseram, também, uma metodologia. O objetivo é modelar os diversos domínios geológicos simultaneamente de forma similar ao caso binário.

Se existirem  $K$  múltiplos domínios no depósito mineral, para cada ponto amostral  $z(u_\alpha)$ ,  $\alpha = 1, \dots, n$ , um vetor de indicadores de  $K$  elementos é codificado de acordo com a

Figura 19 – Modelo geológico criado a partir do sinal da função distância interpolada.



Fonte: Modificado de [Silva \(2015\)](#)

Equação 3.6.

$$i_k(u_\alpha) = \begin{cases} 1, & \text{se } z(u_\alpha) = k \\ 0, & \text{se } z(u_\alpha) \text{ caso contrário} \end{cases} \quad k = 1, \dots, K \quad (3.6)$$

As amostras são codificadas  $K$  vezes, uma vez para cada domínio.

Da mesma forma que na [Equação 3.2](#), a função distância é calculada, individualmente para cada elemento  $k$  do vetor, de acordo com a [Equação 3.7](#).

$$d_k(u_\alpha) = \begin{cases} - \| u_\alpha - u_\beta \|, & \text{se } i_k(u_\alpha) = 1 \\ + \| u_\alpha - u_\beta \|, & \text{se } i_k(u_\alpha) = 0 \end{cases} \quad k = 1, \dots, K \quad (3.7)$$

Diferentes anisotropias podem ser incorporadas para cada categoria, a correlação entre as distâncias calculadas não é considerada.

As distâncias calculadas são então interpoladas, individualmente, para cada categoria, de acordo com a [Equação 3.8](#).

$$d_k^*(u) = \sum_{\alpha=1}^n \lambda_\alpha^{OK}(u) d_k(u_\alpha) \quad k = 1, \dots, K \quad (3.8)$$

Quando os múltiplos variogramas são similares, um único modelo pode ser considerado.

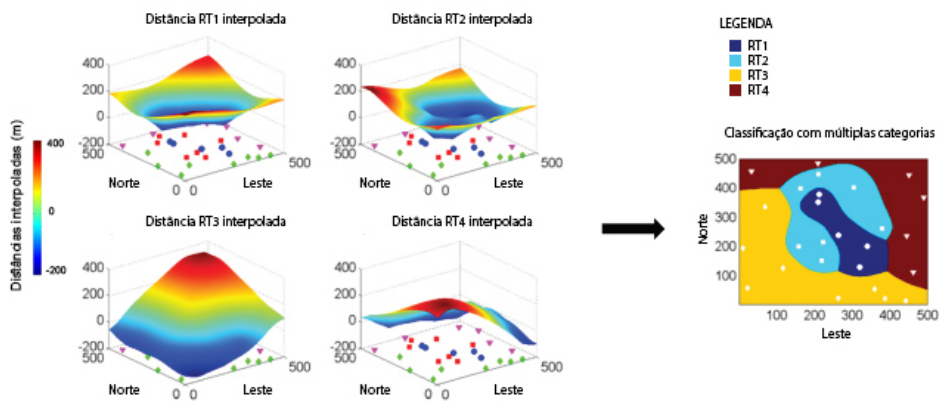
Por fim, cada bloco é classificado pela [Equação 3.9](#)

$$i^*(u) = k' \text{ de modo que } d_{k'}^* = \min\{d_k^*(u)\}_{k=1}^K \quad (3.9)$$

As distâncias estimadas fornecem uma medida de proximidade ao domínio oposto mais próximo. Sendo assim, a mínima distância assinalada estimada é tida como o domínio mais provável de ser encontrado numa região não amostrada. A [Equação 3.9](#) resume essa ideia ([SILVA, 2015](#)). A categoria associada com a menor distância estimada é carimbada em cada bloco.

A [Figura 20](#) mostra um exemplo de um modelo geológico criado a partir da mínima distância assinalada estimada. À esquerda, a figura mostra a projeção das distâncias assinaladas para cada uma das quatro categorias, e à direita a classificação final.

Figura 20 – Função distância interpolada para todos os locais de interesse.



Fonte: Modificado de [Silva \(2015\)](#)

### 3.2 Uma medida heurística de incerteza (*softmax transformation*)

A metodologia proposta por [Silva e Deutsch \(2012\)](#) não caracteriza a incerteza associada ao modelo geológico. [Silva \(2015\)](#) apresentou uma forma heurística para determinação da incerteza, essa metodologia não é baseada em múltiplas realizações de uma função aleatória, é um pós processamento derivado de uma transformação.

Várias formas de determinar a incerteza para métodos determinísticos de modelagem geológica implícita foram desenvolvidos. [McLennan e Deutsch \(2006\)](#) propuseram uma metodologia para quantificar a incerteza de volumes geológicos criados a partir de funções volume (muito similares às funções distância assinaladas) a partir da técnica de *bootstrap sampling*. [Munroe \(2012\)](#) propôs calibrar uma banda de incerteza ao longo dos contatos entre os domínios, uma ideia similar foi implementada por [Wilde e Deutsch \(2011\)](#), onde a incerteza é considerada diretamente na krigagem, adicionando uma constante às estimativas.

As distâncias estimadas em cada bloco para cada categoria carregam consigo

informação adicional. Como já discutido, as distâncias estimadas fornecem uma medida de proximidade à interface mais próxima. Então, as distâncias podem ser standardizadas entre zero e um, fornecendo uma medida de incerteza (SILVA, 2015).

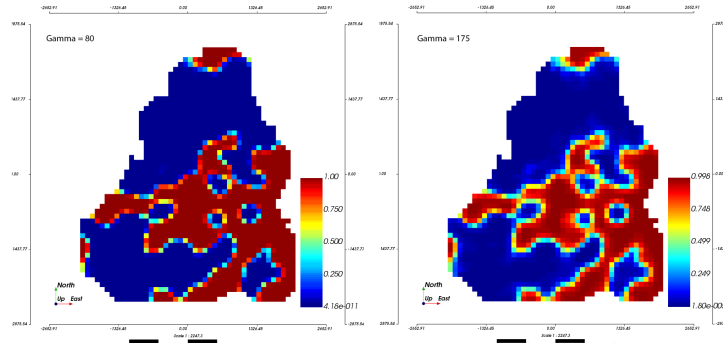
### 3.2.0.1 Transformação das distâncias

*Softmax transformation* é uma técnica amplamente usada em métodos de classificação para múltiplas classes (MCCULLAGH; NELDER, 1989). A ideia é transformar as distâncias estimadas em probabilidades a partir da Equação 3.10. Os valores transformados encontram-se entre zero e um e sua soma deve ser igual a um, para cada bloco estimado.

$$P(i(u) = k) = \frac{e^{-\frac{d_k^*(u)}{\gamma}}}{\sum_{k'=1}^K e^{-\frac{d_{k'}^*(u)}{\gamma}}} \quad (3.10)$$

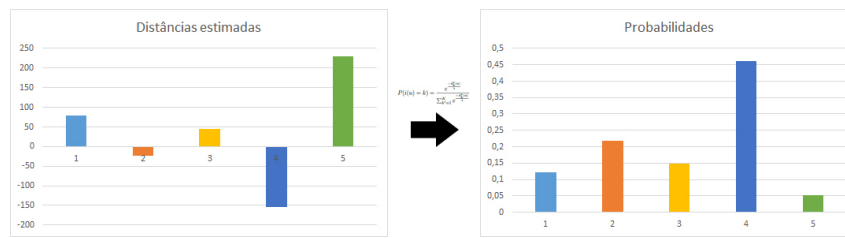
$P(i(u) = k)$  representa a probabilidade de um local  $u$  pertencer à categoria  $k$ ,  $d_k^*(u)$  é a distância estimada para a categoria  $k$  e  $\gamma$  é um parâmetro que regula a inter-relação entre as probabilidades das  $K$  diferentes categorias. Quanto maior  $\gamma$ , maior as diferenças entre as probabilidades (maior a banda de incerteza), como pode ser observado na Figura 21.

Figura 21 – Mapas de probabilidades gerados com dois valores de gamma diferentes. À esquerda  $\gamma = 80$ , à direita  $\gamma = 175$ .



Para exemplificar a transformação, a Figura 22, mostra à esquerda as distâncias estimadas para cada uma de cinco categorias em um mesmo bloco em particular. À direita as distâncias transformadas pela Equação 3.10 em probabilidades daquele bloco pertencer a cada uma das cinco categorias. Observe que distâncias menores (mais negativas) geram probabilidades maiores, e vice-versa.

Figura 22 – Distâncias estimadas e transformadas em probabilidades para um mesmo bloco.



### 3.3 Correção das proporções globais dos domínios (*servo system*)

Uma das preocupações durante a modelagem geológica implícita de domínios geológicos é quanto à inserção de viés nos modelos. O viés pode ser proveniente da interação entre os parâmetros e o algoritmo, da variabilidade geológica ou da configuração da malha de amostragem.

Tipicamente, as campanhas de sondagem têm como foco áreas de altos teores. Como se espera obter lucro ainda nos primeiros anos de operação, mais informação das áreas mais ricas do depósito é fundamental. Essa prática é conhecida como amostragem preferencial e pode levar ao cálculo enviesado das proporções de cada domínio, caso não receba o tratamento devido. Existem diversas técnicas de desagrupamento disponíveis, dentre as mais utilizadas, o método dos polígonos e método das células (SILVA, 2015).

Silva (2015) propôs, como uma extensão do algoritmo, gerar modelos que correspondem às estatísticas representativas (desagrupadas) dos domínios do depósito. Essas proporções são calculadas a partir das probabilidades obtidas através da *softmax transformation*.

#### 3.3.1 *Servo system*

A correção com *servo system* foi introduzida por Strébel e Journal (2000) e objetiva ajustar a probabilidade marginal ao longo do processo iterativo de simulação por MPS em cada nó.

A técnica possibilita reproduzir um conjunto de proporções pré-definido, as proporções observadas nos dados desagrupados, por exemplo. A ideia central do algoritmo é atualizar as probabilidades para cada domínio, em cada nó, baseado na diferença entre a proporção alvo e a proporção marginal corrente (proporção encontradas nos blocos visitados até então) (Equação 3.11), a intensidade da correção é proporcional à magnitude da diferença.

$$P(i(u) = k)^{\text{update}} = P(i(u) = k) + \mu(p(k) - P_k^c(u)) \quad (3.11)$$

$P(i(u) = k)^{\text{update}}$  é a probabilidade atualizada em cada bloco  $u$ ,  $P(i(u) = k)$  é a probabilidade para o bloco  $u$  calculada a partir da *softmax transformation*,  $\mu$  é o parâmetro que controla a intensidade da correção, e vale  $\mu = \frac{\lambda}{1-\lambda}$ , quanto mais próximo de um  $\lambda$  for, mais correção nas proporções.  $p(k)$  e  $P_k^c(u)$  são respectivamente, a proporção alvo e a proporção marginal corrente para a litologia  $k$ . O algoritmo deve visitar os blocos em um caminho aleatório para evitar o surgimento de artefatos nos modelos gerados.

Após as probabilidades de cada bloco pertencer a cada litologia serem atualizadas com a [Equação 3.11](#), um novo modelo geológico deve ser gerado a partir da [Equação 3.12](#).

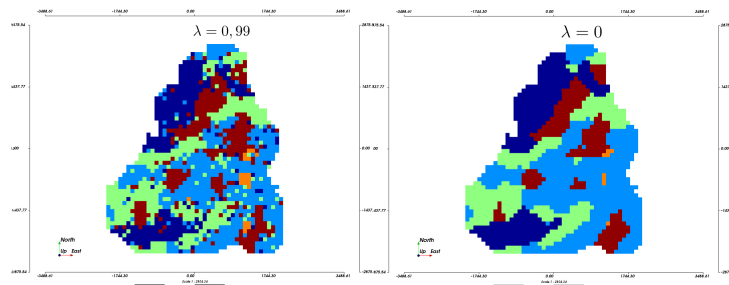
$$i^*(u) = \operatorname{argmax} P(i(u) = k)^{\text{updated}} \quad (3.12)$$

Cada bloco é reclassificado, não mais pela menor distância estimada, mas pela maior probabilidade atualizada.

Muitas vezes, a escolha do parâmetro  $\mu$  (muito alta), gera modelos com aparência granulada ou *salt and pepper* ([Figura 23](#) à esquerda), que não fazem sentido físico. Uma fase de pós processamento para suavizar o modelo está embutida no algoritmo do *servo system* (duas novas propriedades são criadas, uma não corrigida e uma corrigida). Cada nó do *grid* é visitado seguindo um caminho aleatório, novamente para evitar o surgimento de artefatos nos modelos. Um cubo de dimensões  $3 \times 3 \times 3$  blocos cúbicos é centrado no bloco visitado e a proporção de cada litologia é calculada, considerando apenas os blocos abrangidos pelo cubo. A litologia que apresenta a maior proporção é carimbada no bloco visitado, metodologia semelhante é utilizada em classificação de recursos minerais.

A [Figura 23](#) mostra um modelo gerado com um parâmetro  $\lambda = 0.99$  à esquerda e um modelo gerado com  $\lambda = 0$ , ou seja, um modelo sem correção baseado na menor distância estimada.

Figura 23 – Modelos geológicos gerados, sem correção das proporções à direita e com correção, usando um fator  $\lambda = 0,99$  à esquerda.



Alguns modelos podem apresentar efeito de borda, quando a influência de amostras localizadas nas bordas da malha de sondagem é extrapolada com exagero pelo algoritmo,

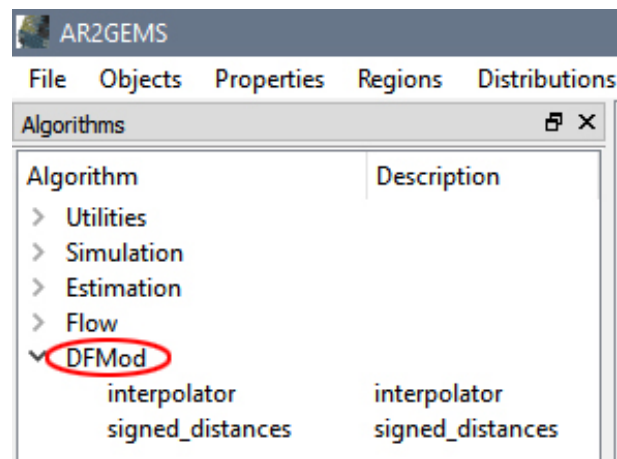
criando estruturas inverossímeis. A aplicação do *servo system* também controla a extensão desse tipo de estrutura, reduzindo o efeito de borda.

### 3.4 O Plug-in

O método foi operacionalizado no *software* geostatístico de código aberto *SGeMS*, com a criação de dois *plug-ins* em linguagem *python*. Os códigos podem ser encontrados no Apêndice A e Apêndice B, bem como no repositório *online* <<https://github.com/LPM-UFRGS/dfmod>>

No painel de algoritmos, há a opção *DFMod* (Figura 24), que traz dois *plugins*: *signed distances* e *interpolator*. O primeiro calcula o valor da função distância assinalada para cada amostra e para cada litologia. O segundo interpola a função distância, usando krigagem ordinária, para todo o *grid*, além de apresentar opções de calcular a incerteza com *softmax transformation* (seção 3.2) e corrigir as proporções com *servo system* (seção 3.3).

Figura 24 – Visão do painel de algoritmos do *SGeMS* mostrando a opção *DFMOD*.



#### 3.4.1 Signed distances

O *plugin signed distances* (Figura 25) calcula, para uma propriedade categórica, o valor da função distância anisotrópica (ou isotrópica), para cada amostra e cada litologia. O *SGeMS* aceita apenas litologias codificadas em caracteres numéricos.



Figura 25 – Visão do *plugin signed distances*.

Em (i), usuário deve imputar a propriedade que informa a litologia (*rock type*) de cada amostra. Em (ii), o usuário imputa os ângulos de rotação e razões de anisotropia, para o cálculo das distâncias anisotrópicas, o valor padrão calcula distâncias isotrópicas.

Como *output*, o plugin gera uma propriedade *signed distances RTX* para cada litologia, onde  $X$  é o algarismo que representa aquela litologia.

O quadro Algoritmo 1 abaixo, mostra o funcionamento do algoritmo.

#### Algoritmo 1: Signed distances

**Input:** Banco de dados categórico, ângulos de rotação e razões de anisotropia

**Output:** Propriedades *signed distances RTX*

- 1 Cria a matriz de rotação - dilatação/compressão;
- 2 Transforma os dados originais  $x, y, z$  em  $x'', y'', z''$  com a matriz criada;
- 3 Codifica as amostras, para cada domínio  $k$ , em  $i_k(u_\alpha)$ ;
- 4 **for** *litologias* **do**
- 5     **for** *amostras* **do**
- 6         Calcula a distância entre cada amostra e todas as demais que pertencem a um domínio oposto;
- 7         Retém a distância mínima para cada amostra em uma lista;
- 8         Assinala de acordo com o indicador
- 9     **end**
- 10     Cria uma propriedade para cada litologia, referente a lista de menores distâncias criada em cada amostra (*signed distances RTX*)
- 11 **end**

### 3.4.2 Interpolator

O *plugin interpolator* apresenta três abas: *general*, *variogram* e *options*. Na primeira aba (Figura 26), o usuário deve imputar em (i) a propriedade categórica, nesse caso, para que os blocos que coincidem com amostras sejam carimbados com a respectiva litologia. Em

(ii), o *grid* no qual as distâncias serão interpoladas. Em (iii), o usuário deve imputar todas as distância *signed distances RTX* calculadas com o *plugin signed distances* (subseção 3.4.1). Em (iv), a estratégia de busca da krigagem originária deve ser informada, número máximo e mínimo de dados condicionantes e elipsoide de busca.

Figura 26 – Visão da aba *general* do *plugin interpolator*.

General Variogram Options

Choose rock property (i)

<- None ->

Choose the grid to interpolate (ii)

<- None ->

Choose signed distances properties (iii)

<- None ->

Choose Properties...

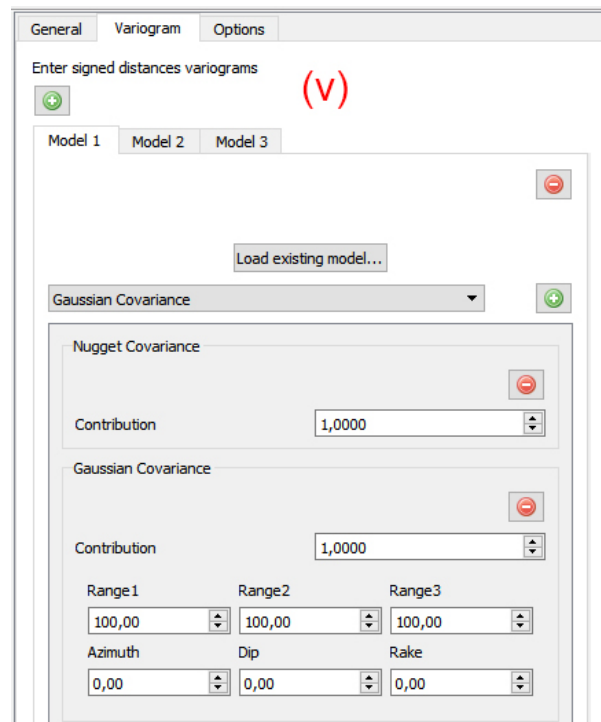
Conditioning data (iv)

Min 0 Max 0

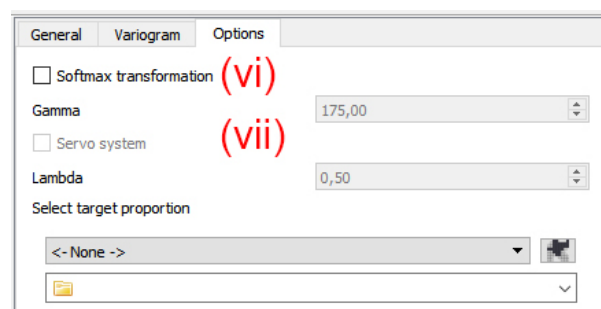
Search strategy

Range1	Range2	Range3
1,00	1,00	1,00
Azimuth	Dip	Strike
0,00	0,00	0,00

Na segunda aba (Figura 27), os modelos variográficos referentes a cada uma das propriedades *signed distances RTX*, devem ser informados, na mesma ordem em que foram escolhidos na aba anterior (Figura 26 - (iii)).

Figura 27 – Visão da aba *variogram* do *plugin interpolator*.

Na terceira e última aba (Figura 28), o usuário escolhe se o modelo será pós processado. Em (vi), a opção *softmax transformation* deve ser marcada, assim como o parâmetro  $\gamma$  informado, em (vii), a opção *servo system* deve ser marcada, o parâmetro  $\lambda$  informado e a propriedade para a qual se deseja reproduzir as proporções imputada. A seleção do *servo system* está condicionada à seleção da *softmax transformation*.

Figura 28 – Visão da aba *options* do *plugin interpolator*.

Como *output*, o *plugin* gera, caso nenhuma opção da aba *options* tenha sido selecionada, uma propriedade com o nome *geo\_model*. Esse é o modelo geológico baseado na menor distância estimada para cada bloco. Ao marcar a opção *softmax transformation*, uma propriedade *Probability RTX* é criada, para cada litologia, onde *X* é o algarismo que representa aquela litologia. Caso a opção *servo system* seja selecionada, as propriedades *Geologic\_Model\_Servo\_System* e *Geologic\_Model\_Corrected* são criadas. A primeira

representa o modelo criado a partir das probabilidades atualizadas e a segunda representa esse modelo suavizado (a correção é aplicada aqui uma única vez).

O quadro Algoritmo 2 abaixo mostra o funcionamento do algoritmo.

<b>Algoritmo 2:</b> Interpolator	
	<b>Input:</b> Banco de dados categórico, propriedades <i>signed distances RTX</i> , estratégia de krigagem, modelos variográficos, proporção alvo e parâmetros
	<b>Output:</b> Propriedades <i>geo_model</i> , <i>Probability RTX</i> , <i>Geologic_Model_Servo_System</i> e <i>Geologic_Model_Corrected</i>
1	<b>for</b> cada propriedade <i>signed distances RTX</i> <b>do</b>
2	Chama a krigagem ordinária no <i>SGeMS</i> com os parâmetros imputados pelo usuário e interpola as distâncias calculadas para cada nó do grid informado;
3	<b>end</b>
4	<b>for</b> cada nó do grid <b>do</b>
5	Carimba no nó a categoria referente à menor distância estimada
6	<b>end</b>
7	Cria a propriedade <i>geo_model</i> , baseada na menor distância interpolada;
8	<b>if</b> opção <i>softmax transformation</i> marcada <b>then</b>
9	<b>for</b> cada propriedade interpolada <b>do</b>
10	<b>for</b> cada nó do grid <b>do</b>
11	Transforma cada distância interpolada em probabilidade daquele nó pertencer a uma das litologias
12	<b>end</b>
13	Cria uma propriedade <i>Probability RTX</i> para cada litologia
14	<b>end</b>
15	<b>end</b>
16	<b>if</b> opção <i>servo system</i> marcada <b>then</b>
17	Gera caminho aleatório;
18	<b>for</b> nó do grid em caminho aleatório <b>do</b>
19	Atualiza a proporção corrente marginal de cada categoria;
20	<b>for</b> cada propriedade <i>Probability RTX</i> <b>do</b>
21	Atualiza as probabilidades de cada nó pertencer a cada uma das litologias de acordo com a diferença entre a proporção alvo e a proporção corrente marginal
22	<b>end</b>
23	<b>end</b>
24	Cria a propriedade <i>Geologic_Model_Servo_System</i> ;
25	Suaviza o modelo e cria a propriedade <i>Geologic_Model_Corrected</i>
26	<b>end</b>
27	Carimba nos blocos a categoria correspondente ao dado colocado

### 3.5 Validação

Os *plugins* desenvolvidos foram validados tendo como referência resultados análogos da rotina *DFMod* da biblioteca de algoritmos geoestatísticos *GSLib*. Foi utilizado um

banco de dados em três dimensões que possui 3276 amostras de 5 categorias diferentes para a validação do cálculo da função distância e modelo gerado a partir das distâncias estimadas. Ainda foi utilizado o *jura data set* (GOOVAERTS, 1997) para validação da *softmax transformation e servo system*.

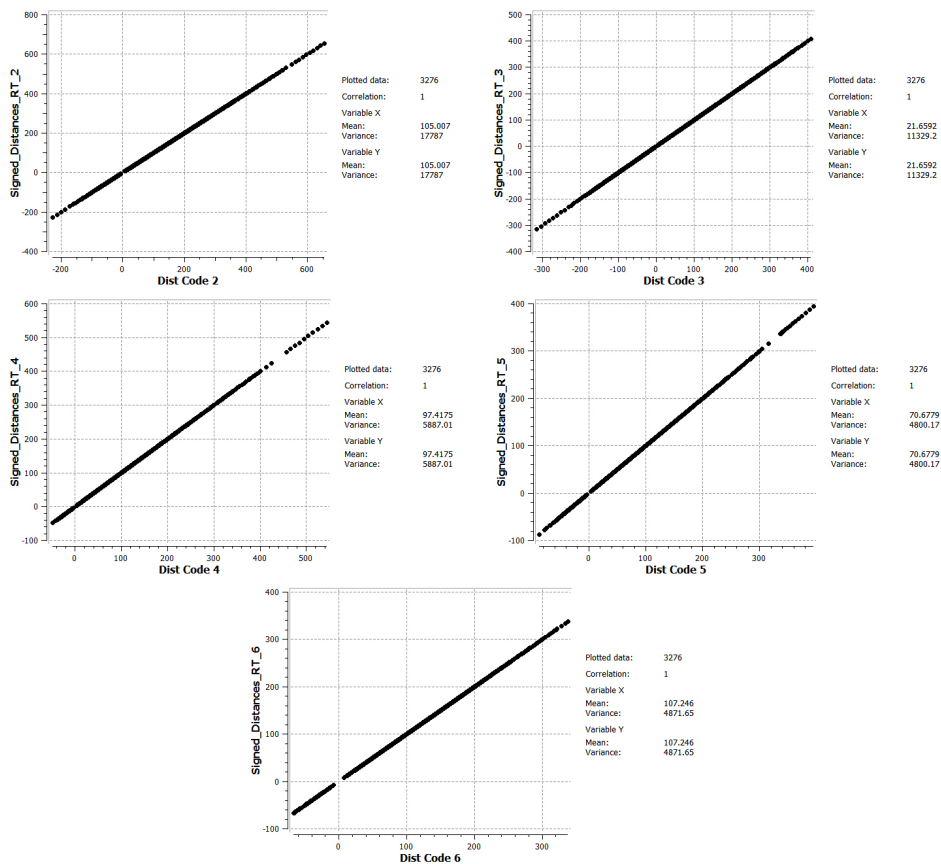
Distâncias anisotrópicas foram calculadas com os parâmetros da Tabela 1, escolhidos de forma arbitrária. E os resultados comparados.

Tabela 1 – Ângulos de rotação e proporções entre as direções principais usadas no cálculo da função distância anisotrópica.

Rotações	30°	45°	60°
Proporções	2	5	9

A Figura 29 mostra os *scatterplots* entre as distâncias calculadas com o *plugin* desenvolvido, no eixo *y*, contra as calculadas a partir da rotina *DFMod* do *GSLib* no eixo *x*, para todas as cinco categorias do banco de dados. A correlação é perfeita (100%) para todos os gráficos.

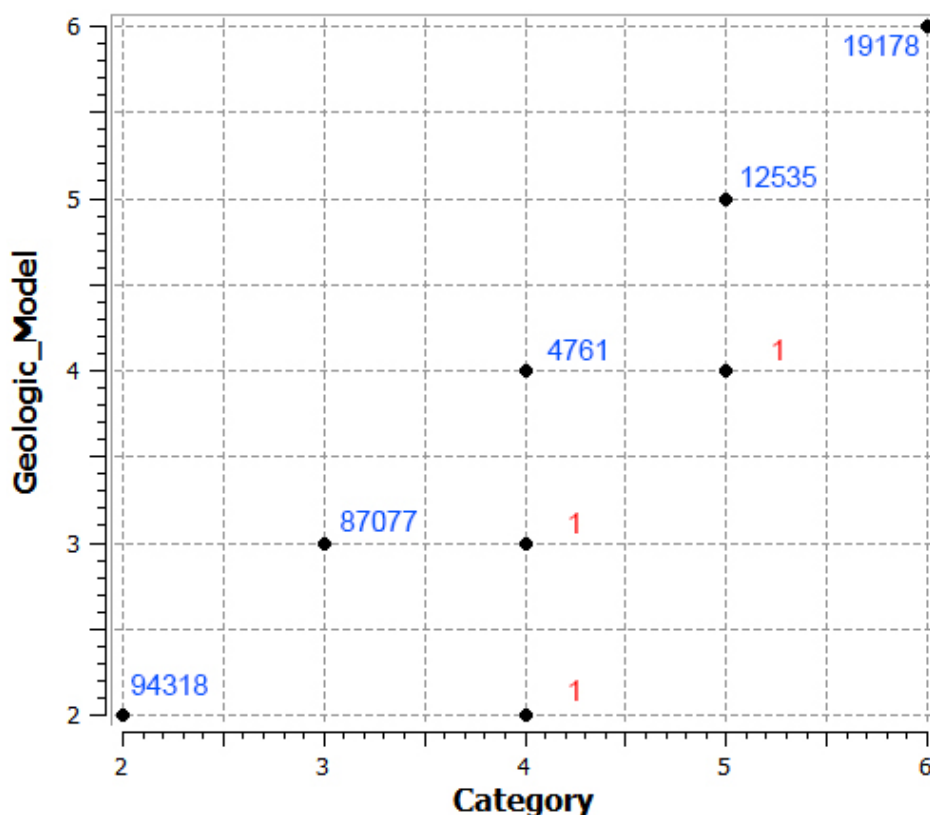
Figura 29 – *Scatterplot* entre as distâncias calculadas pelo *plugin* desenvolvido contra distâncias calculadas a partir da rotina *DFMod* do *GSLib*.



Dois modelos foram gerados a partir dos parâmetros do Apêndice A. E foram

comparados, novamente, em um *scatterplot* categórico (Figura 30). O modelo gerado pelo *plugin* desenvolvido aparece no eixo *y*, e no eixo *x*, o modelo criado a partir da rotina *DFMod* do *GSLib*. 99,998% dos blocos são concordantes em ambos os modelos, blocos concordantes caem sobre a linha  $x=y$  e blocos discordantes fora dela, existem apenas três blocos discordantes entre os modelos gerados. 100% de blocos concordantes não pôde ser atingido devido a diferença entre os códigos e linguagens de programação, as rotinas do *GSLib* são escritas em *fortran 90* enquanto o *SGeMS* e seus *plugins*, em *C++* e *python*, respectivamente.

Figura 30 – *Scatterplot* entre os modelos criados pelo *plugin* desenvolvido e rotina *DFMod* do *GSLib*.

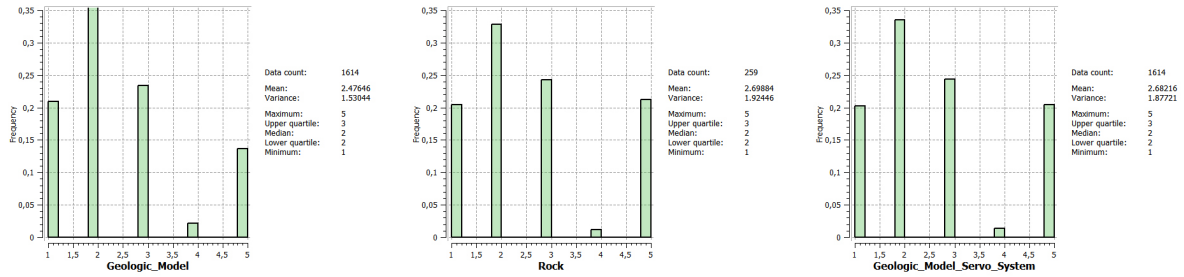


Em nenhuma das versões do *DFMod* encontradas nos repositórios do *GSLib* há versões funcionais, tanto da *softmax transformation* quanto do *servo system*. Então, como alternativa para validação, a correção *servo system* foi executada no banco de dados jura, com um parâmetro  $\lambda = 0,99$ . Dessa forma, é esperado que se, tanto o algoritmo da *softmax transformation* quanto o algoritmo do *servo system* (que depende do último), estejam corretos a proporção alvo seja reproduzida, ou pelo menos que um resultado bastante próximo seja obtido, já que não é possível usar um fator  $\lambda = 1$  (divisão por zero), que acarretaria na máxima correção possível, isto é, histogramas idênticos.

A Figura 31 mostra os histogramas do modelo baseado somente nas distâncias

estimadas à esquerda, dos dados ao centro (este foi utilizado como alvo para o *servo system*), e do modelo após execução do *servo system* à direita. As proporções alvo foram quase reproduzidas com perfeição, a diferença entre as médias dos histogramas é menor que 0,01.

Figura 31 – Histogramas do modelo baseado somente nas distâncias estimadas à esquerda, dos dados ao centro, e do modelo após execução do *servo system* à direita.







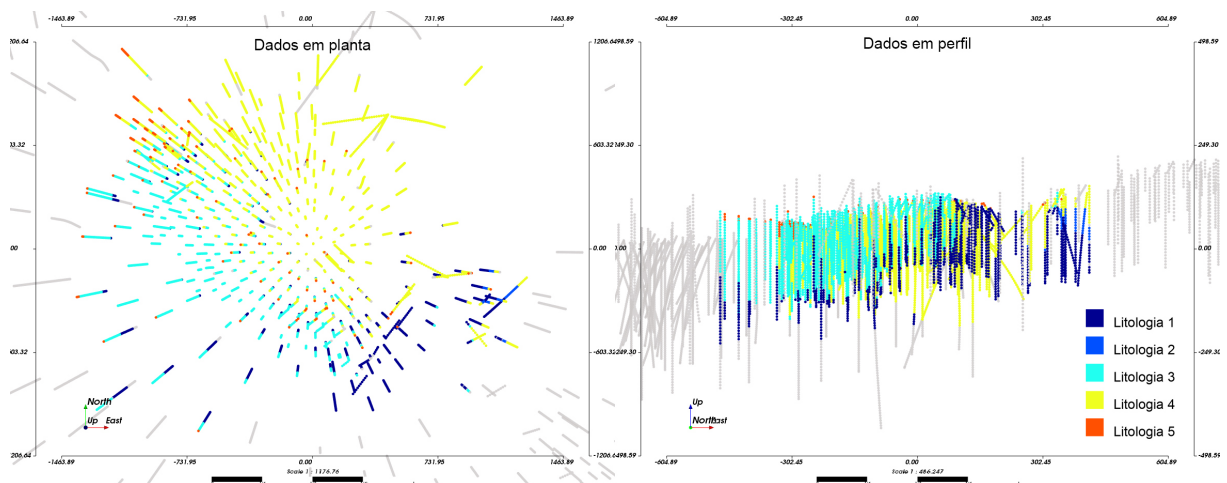
## 4 Estudo de caso

A metodologia desenvolvida no decorrer desse trabalho foi aplicada a um banco de dados real, proveniente de uma grande mineradora de ouro. Para verificar a competência do método, o modelo criado implicitamente pelo algoritmo proposto foi comparado com um modelo criado por um geomodelador, através da digitalização manual de polígonos em seções, e posterior construção dos sólidos que representam as unidades litológicas. Assim, foi possível avaliar se as estruturas geológicas interpretadas foram satisfatoriamente reproduzidas pelo algoritmo.

O banco de dados categóricos fornecido abrange 9140 amostras, distribuídas entre cinco diferentes litologias. O depósito cobre uma área de aproximadamente  $10\text{km}^2$  com  $1300\text{m}$  de profundidade. Maiores detalhes em relação à geologia e localização do depósito serão omitidos por questões de confidencialidade.

A [Figura 32](#) evidencia as amostras em planta, à esquerda, e em perfil, à direita. Enquanto a [Figura 33](#), mostra o histograma dos dados, que expressa as proporções de cada litologia nos dados amostrais.

Figura 32 – Dados em planta à esquerda e em perfil à direita.



A partir das amostras, o valor da função distância assinalada foi calculado para cada litologia, em cada ponto amostral. O depósito não apresenta morfologia que justifique o cálculo de distâncias anisotrópicas, então, o caso isotrópico foi escolhido.

A [Figura 34](#) mostra as distâncias assinaladas calculadas para cada uma das cinco litologias, a transição suave entre as cores denota o comportamento extremamente contínuo das distâncias assinaladas.

Figura 33 – Histograma dos dados.

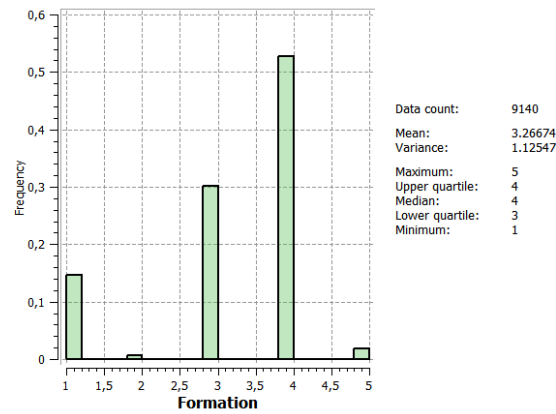
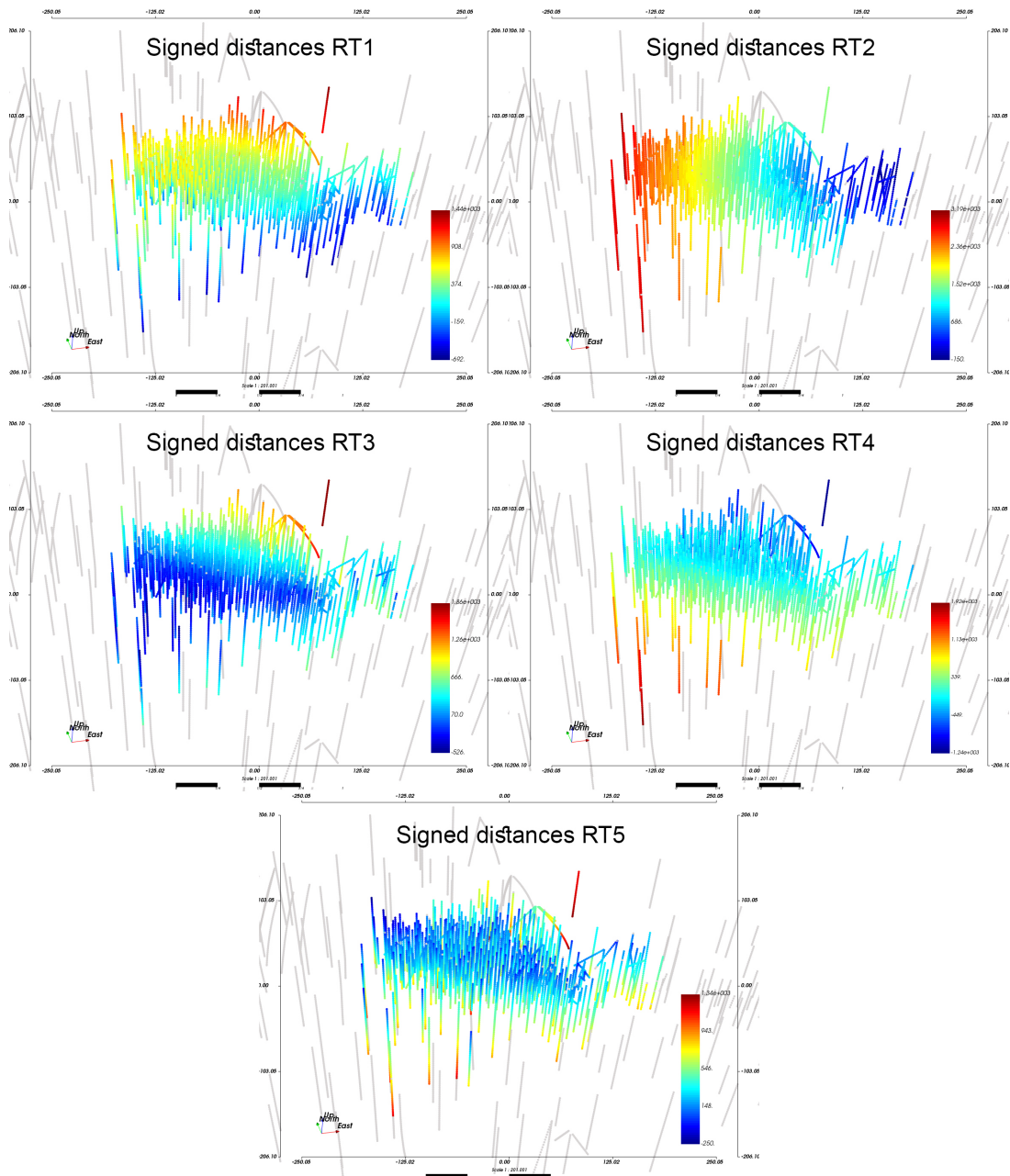


Figura 34 – Distâncias assinaladas calculadas para cada uma das litologias.



## 4.1 Variografia

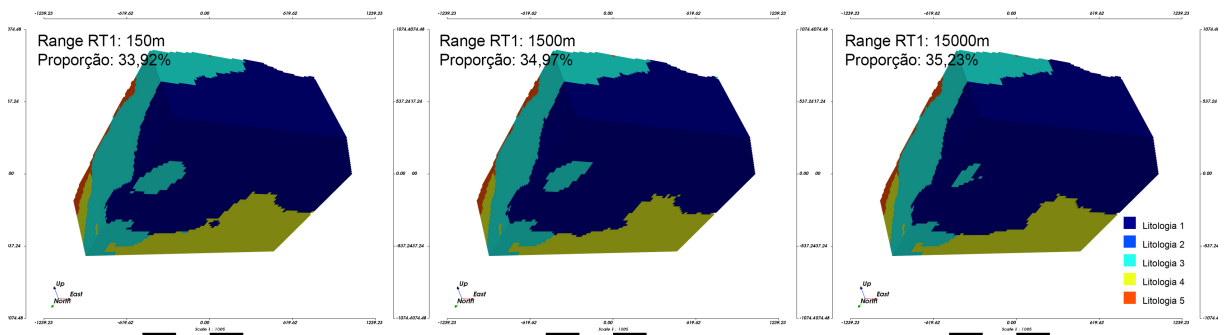
As distâncias assinaladas apresentam, além do comportamento extremamente contínuo observado na [Figura 34](#), um comportamento não estacionário. Isto é, seu variograma experimental não se estabiliza em um patamar. Essa característica torna a inferência do alcance arbitrária, como observado na [Figura 37](#).

O alcance do variograma, juntamente com a anisotropia, controlam a extensão e forma dos domínios modelados. Na [Figura 35](#), a litologia 1 foi modelada a partir de três valores diferentes de alcance de um variograma isotrópico, mantido o mesmo patamar: 150m à esquerda, 1500m no centro, e 15000m à direita. O variograma das demais litologias não foi alterado.

Ao aumentar o alcance do variograma de um dos domínios modelados, a estrutura associada à esse domínio ganha um corpo mais volumoso, crescendo na direção de maior continuidade do variograma, e conseqüentemente sua proporção no depósito aumenta, como observado na análise da litologia 1 da [Figura 35](#). A diferença entre as proporções não é muito pronunciada já que a densidade amostral do banco de dados é muito alta. Em análise semelhante, [Silva \(2015\)](#) obteve resultados análogos.

A inferência do alcance dos variogramas pode se tornar arbitrária devido ao comportamento não estacionário das distâncias assinaladas. Uma alternativa consiste em fixar a variância dos variogramas na variância *a priori* dos dados, e ajustar um modelo ao variograma experimental dessa forma. Uma outra alternativa é inferir o alcance a partir das dimensões das estruturas observadas nos dados amostrais.

Figura 35 – Litologia 1, e respectiva proporção, modelada variando o range do variograma: 150m à esquerda, 1500m ao centro e 15000m à direita.

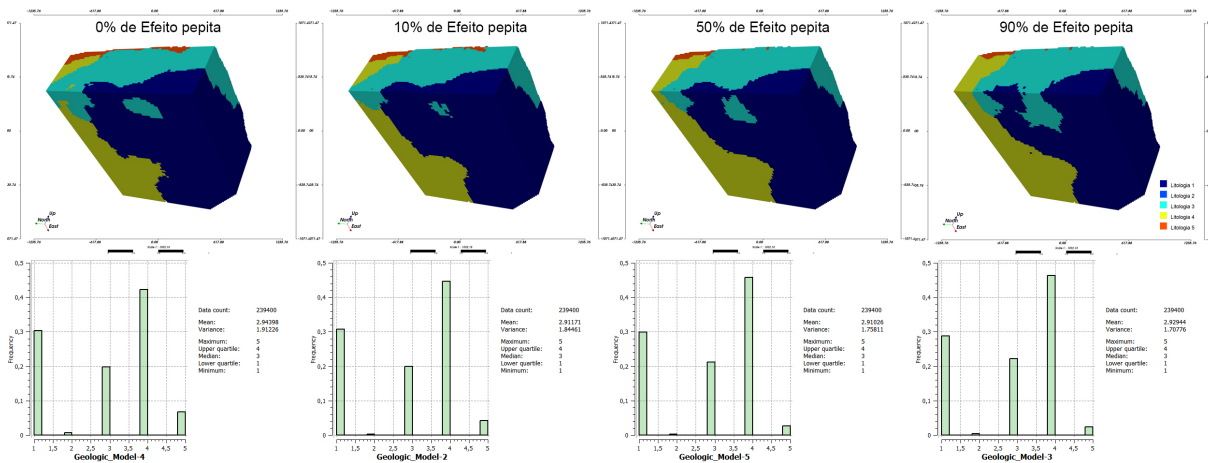


Os variogramas experimentais das distâncias assinaladas nunca exibem efeito pepita. Porém, este pode ser adicionado arbitrariamente pelo usuário para controlar a interconectividade dos domínios. A [Figura 36](#) mostra quatro modelos gerados a partir de variogramas isotrópicos com o mesmo alcance para todas as litologias e seus respectivos histogramas. Apenas a proporção entre a contribuição do efeito pepita e a contribuição

total do variograma foi alterada, da mesma maneira, para todas as litologias.

Na [Figura 36](#), da esquerda para a direita, não há efeito pepita algum na primeira imagem, 10% de efeito pepita na segunda imagem, 50% e 90% de efeito pepita nas imagens seguintes.

Figura 36 – Modelos calculados a partir de variogramas com diferentes proporções de efeito pepita. Da direita para a esquerda: 0%, 10%, 50% e 90%.



O efeito do aumento da contribuição do efeito pepita para a variância total é mais pronunciado na litologia 1, é possível observar que a medida que a contribuição aumenta o domínio representado pela litologia 1 se torna mais desconexo. Na última imagem (90% de efeito pepita), pode-se observar dois "braços" que são interligados quando o modelo é gerado com proporções menores de efeito pepita nos variogramas. Os histogramas pouco se alteram, novamente, devido à alta densidade amostral do banco de dados.

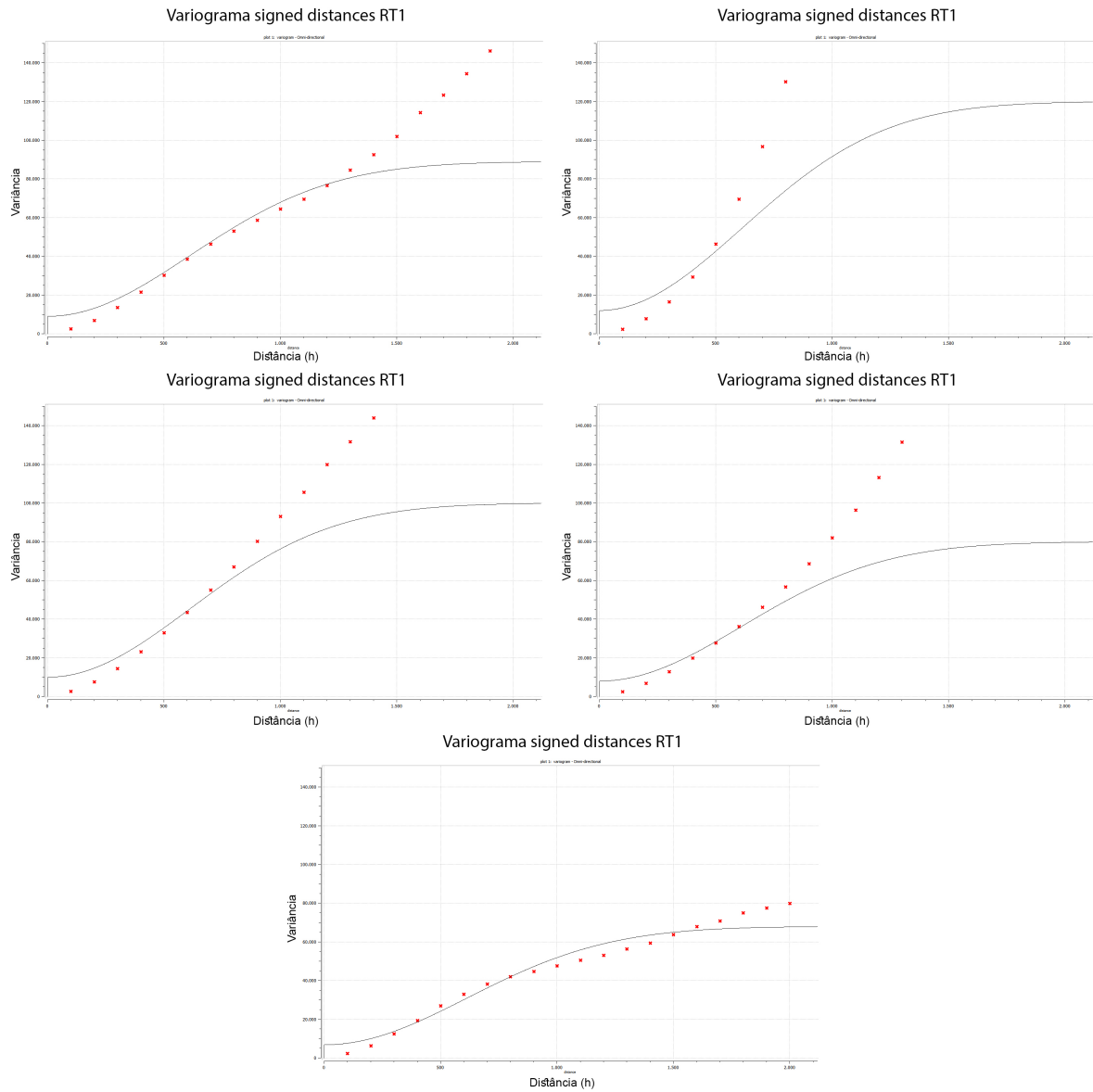
Posto isso, os variogramas experimentais das distâncias assinaladas referentes às cinco litologias do depósito foram calculados, a partir dos parâmetros da [Tabela 2](#), e modelados. Os variogramas não apresentaram nenhuma direção preferencial de continuidade. [A Figura 37](#) mostra os variogramas experimentais isotrópicos para todas as cinco distâncias assinaladas e seu respectivo modelo ajustado ([Equação 4.1](#) a [Equação 4.5](#)).

Tabela 2 – Parâmetros para o cálculo da variograma experimental.

nº de lags	sep. lag (m)	tol. lag (m)	larg. de banda (m)
20	100	50	50

O comportamento contínuo das distâncias faz com que o modelo gaussiano seja o que melhor se ajusta aos pontos do variograma experimental, geralmente, apenas uma estrutura é suficiente para um bom ajuste. O efeito pepita foi adicionado arbitrariamente para controlar a interconexão entre as litologias.

Figura 37 – Variograma das distâncias assinaladas calculadas para cada uma das litologias.



$$\gamma_{SD1}(h) = 9000 + 80000gauss\left(\frac{h}{1500m}\right) \quad (4.1)$$

$$\gamma_{SD2}(h) = 12000 + 108000gauss\left(\frac{h}{1500m}\right) \quad (4.2)$$

$$\gamma_{SD3}(h) = 10000 + 90000gauss\left(\frac{h}{1500m}\right) \quad (4.3)$$

$$\gamma_{SD4}(h) = 8000 + 72000gauss\left(\frac{h}{1500m}\right) \quad (4.4)$$

$$\gamma_{SD5}(h) = 6800 + 61200gauss\left(\frac{h}{1500m}\right) \quad (4.5)$$

O valor absoluto da variância dos variogramas não tem influência alguma no resultado da krigagem ordinária, somente na variância de krigagem, que não é usada para criar os modelos gerados implicitamente pelo método das distâncias assinaladas. Apenas a proporção das contribuições das estruturas para a variância total modifica o resultado da krigagem. Como os modelos ajustados aos variogramas experimentais da [Figura 37](#) apresentam, todos eles, proporção de 10% efeito pepita e 90% de contribuição da estrutura gaussiana, além do mesmo range, foi possível o uso do mesmo modelo variográfico da [Equação 4.6](#) para todas as litologias.

$$\gamma_{SD}(h) = 0,1 + 0,9gauss\left(\frac{h}{1500m}\right) \quad (4.6)$$

## 4.2 Interpolação

As distâncias calculadas e posteriormente variografadas, foram interpoladas no mesmo *grid* do modelo de referência fornecido. As propriedades do *grid* são mostradas na [Tabela 3](#). Uma máscara (*region*) correspondente à topografia foi aplicada à esse *grid*.

Tabela 3 – Propriedades do *grid*.

Número de blocos			Dimensão dos blocos (m)		
Num. X	Num. Y	Num. Z	Dim. X	Dim. Y	Dim. Z
70	60	57	50	50	25

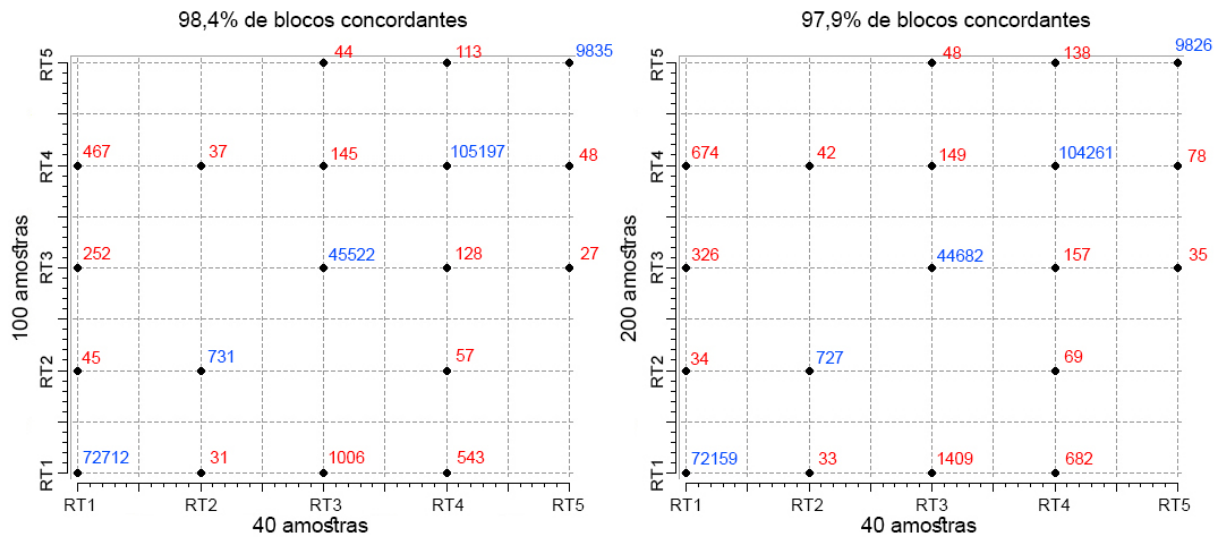
As distâncias foram interpoladas por krigagem ordinária em todos os nós do *grid*, com a estratégia de krigagem da [Tabela 4](#).

Tabela 4 – Parâmetros de krigagem ordinária.

	Vizinhança de busca (m)			Número de amostras	
	Raio (X)	Raio (Y)	Raio (Z)	Min. amostras	Max. amostras
Litologias (1-5)	3000	3000	3000	4	40

A [Figura 38](#) mostra, à esquerda, um scatterplot categórico entre um modelo criado utilizando no máximo 40 amostras por estimativa, no eixo x, e no eixo y, um modelo criado usando no máximo 100 amostras por estimativa. E à direita, um outro *scatterplot*, o mesmo modelo criado utilizando no máximo 40 amostras, no eixo x, e no eixo y, um modelo criado usando no máximo 200 amostras por estimativa. Todas as estratégias de krigagem utilizadas apresentam alcance de busca suficientemente grande para abranger o número máximo de amostras.

Figura 38 – Scatterplots entre modelos gerados com diferentes estratégias de krigagem.

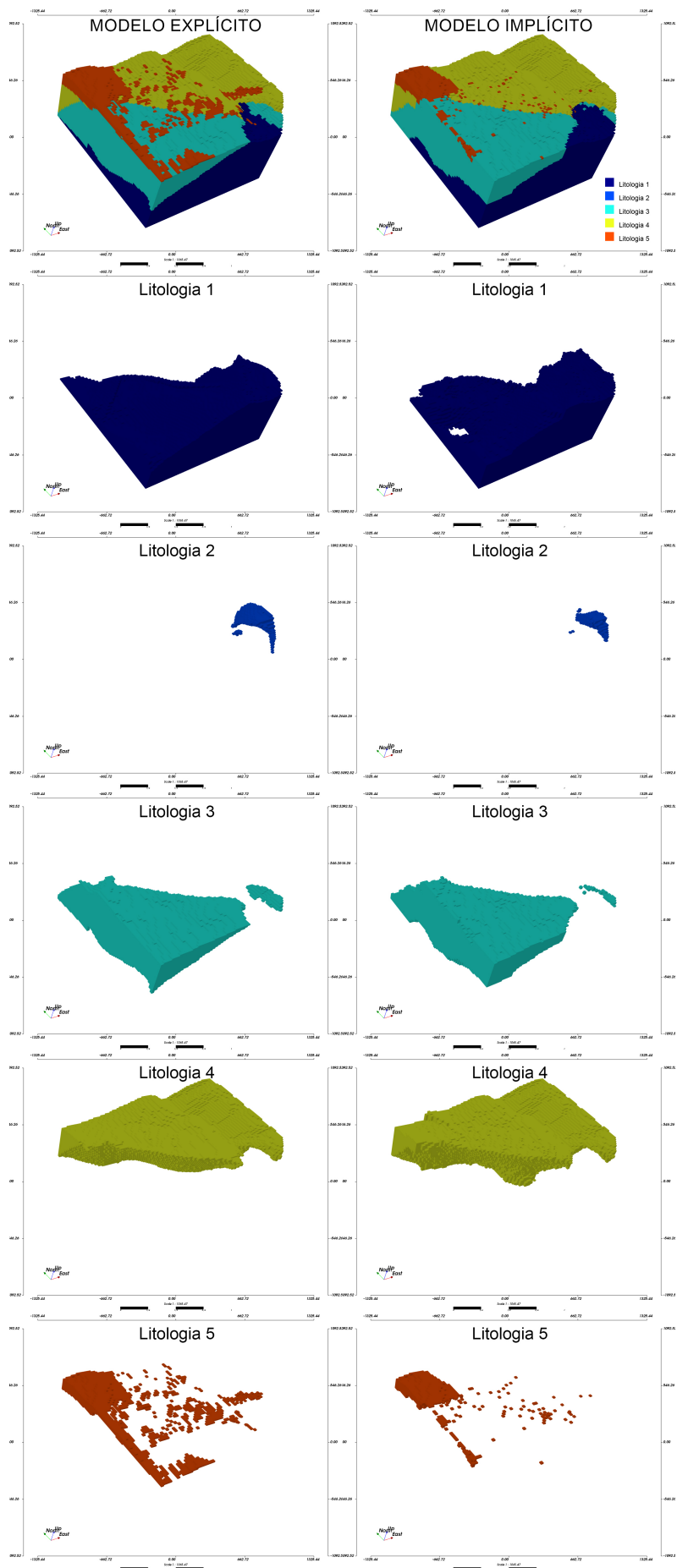


O modelo criado com 40 amostras e o modelo criado com 100 amostras apresentam 98,4% dos blocos concordantes e o modelo criado com 40 amostras e 200 amostras 97,9% dos blocos concordantes. No primeiro caso, a semelhança entre os modelos é suficientemente alta para justificar o uso de no máximo 40 amostras, tendo em vista que o tempo de execução do algoritmo passa de poucos minutos para algumas horas. No segundo caso, a diferença no número de blocos concordantes, para o primeiro caso, é extremamente baixa, e a diferença no tempo de execução significativa. O uso de mais do que 200 amostras por estimativa torna a aplicação do método, utilizando os *plug-ins* em *python* desenvolvidos, inviável. Isso se deve à uma limitação da própria linguagem de programação que trabalha apenas com um núcleo do processador por vez.

## 4.3 Resultados

Um modelo baseado na menor distância interpolada em cada bloco foi gerado pelo *plug-in*, e é mostrado, em perspectiva, lado a lado com o modelo criado por um geomodelador na Figura 39. O modelo criado implicitamente pelo algoritmo à direita e o modelo criado, explicitamente, pelo geomodelador à esquerda. Cada litologia modelada foi colocada lado a lado separadamente para fins de comparação.

Figura 39 – Comparação entre os modelos criados: modelo explícito à esquerda e modelo implícito à direita.

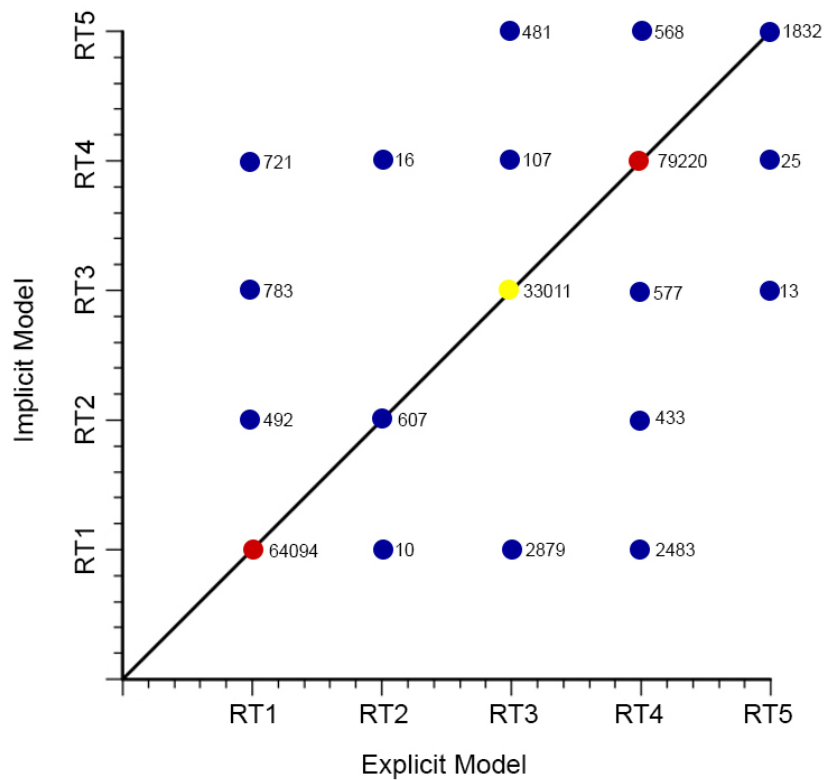




A Figura 40 mostra o *scatterplot* categórico entre os dois modelos criados: implícito no eixo y, e explícito no eixo x. Ao lado de cada ponto há o número de blocos comparados. Pontos que caem sobre a linha 45° são blocos concordantes em ambos os modelos, e pontos que caem fora dessa linha, blocos discordantes.

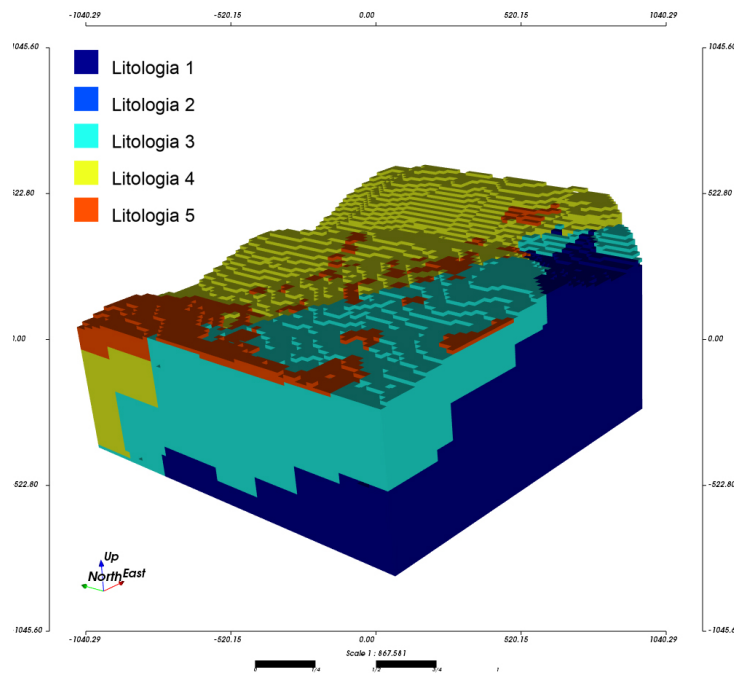
95% dos blocos são concordantes em ambos os modelos. Todavia, apenas uma alta proporção de blocos concordantes não é indicativo de um bom modelo. Métodos menos sofisticados atingem alta proporção, 93% no caso do vizinho mais próximo.

Figura 40 – *Scatterplot* entre os modelos explícitos e implícitos.



Porém, geram modelos que não fazem sentido físico do ponto de vista geológico, como observado na Figura 41, um modelo gerado pela técnica do vizinho mais próximo a partir do mesmo banco de dados, são notáveis estruturas em forma de escada com mudança abrupta de angulação, diferentemente das formas suaves criadas pelo algoritmo proposto vistos na Figura 39.

Figura 41 – Modelo criado utilizando a técnica do vizinho mais próximo.



A incerteza também foi calculada como a probabilidade de cada de cada bloco pertencer a cada uma das categorias pela *softmax transformation*. O parâmetro  $\gamma = 175$ , valor padrão do *plug-in*, foi escolhido. O resultado é mostrado nas seções verticais da [subseção 4.3.1](#).

### 4.3.1 Comparação

Uma comparação inicial pode ser feita observando os modelos, implícito e explícito, lado a lado na [Figura 39](#). A litologia 5 aparece como blocos isolados e dispersos no modelo implícito, enquanto que no modelo explícito apresenta forma mais estruturada e contínua. Examinando a [Figura 32](#), percebe-se algumas amostras de litologia 5 espalhadas, em meio à abundantes amostras de outras litologias. O método das distâncias assinaladas não atribuirá aos blocos ao redor de amostras dispersas, sua litologia, já que a menor distância estimada para esses blocos será referente à categoria das amostras abundantes e agrupadas.

Os blocos isolados pertencentes à litologia 5 que aparecem no modelo implícito são blocos carimbados, após a geração do modelo baseado nas distâncias, com a litologia da amostra colocada, como visto no quadro Algoritmo 2 da [subseção 3.4.2](#). O geomodelador aumentou a influência das amostras de litologia 5 no modelo explícito arbitrariamente ou munido de informação adicional, que não pode ser imputada no algoritmo proposto.

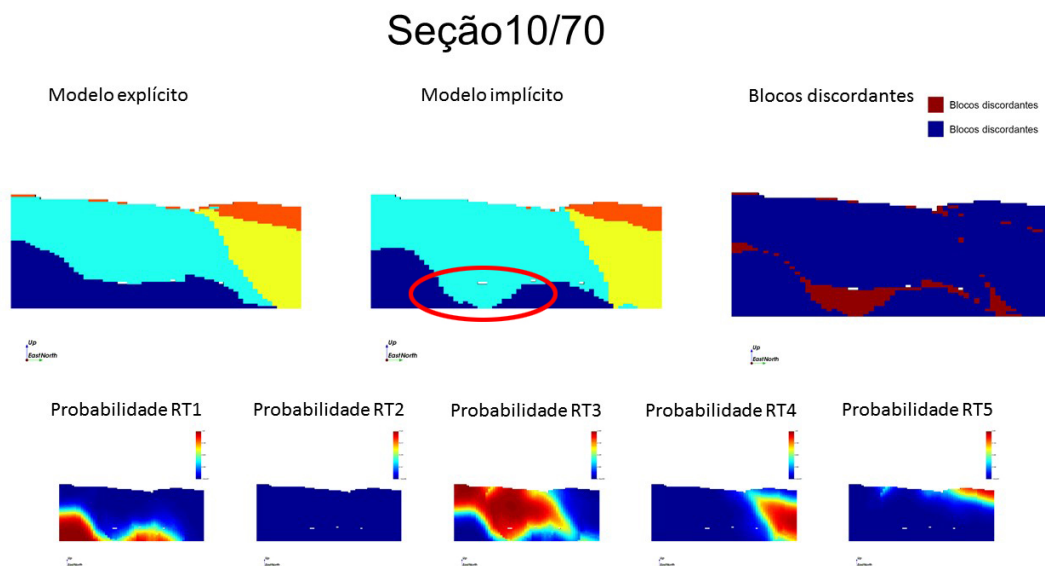
Para checar se o algoritmo reproduziu satisfatoriamente as estruturas interpretadas pelo geomodelador, seções verticais ao longo das direções x e y foram comparadas. Quatro seções espalhadas por toda extensão do modelo foram escolhidas em cada direção.

Em x, o modelo apresenta 70 seções, a [Figura 42](#) mostra a seção 10/70. À esquerda o modelo criado pelo geomodelador, ao centro o modelo criado pelo algoritmo proposto, à direita blocos discordantes e concordantes e abaixo as probabilidades de cada bloco pertencer a cada uma das cinco categorias. Todas as seções analisadas seguem a mesma diagramação.

Estruturas geológicas que foram reproduzidas satisfatoriamente pelo algoritmo foram destacadas por elipses verdes, enquanto estruturas que não foram bem reproduzidas, destacadas por elipses vermelhas.

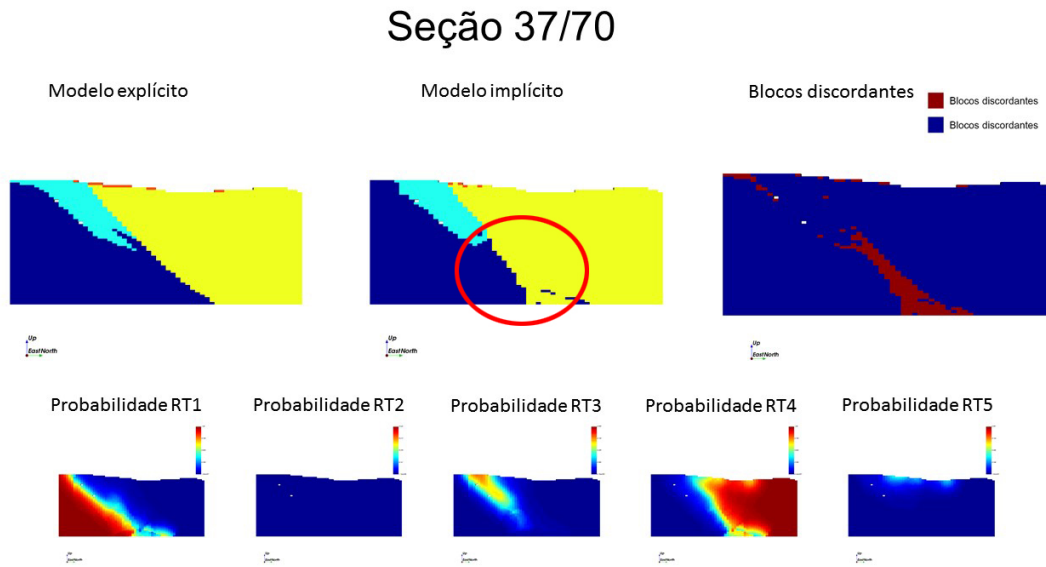
Na região destacada pela elipse vermelha da [Figura 42](#), não há amostras da litologia 1. Provavelmente, o geomodelador munido de sua experiência e/ou informação adicional atribuiu a litologia 1 aos blocos dessa região. Note que alguns blocos na superfície do modelo implícito não foram carimbados com a litologia 5 como no modelo explícito. Porém, ao analisar a probabilidade RT5, existe uma chance considerável desses blocos pertencerem à litologia 5, mesmo que não tenham sido atribuídos àqueles blocos.

Figura 42 – Seção vertical 10/70 em x.



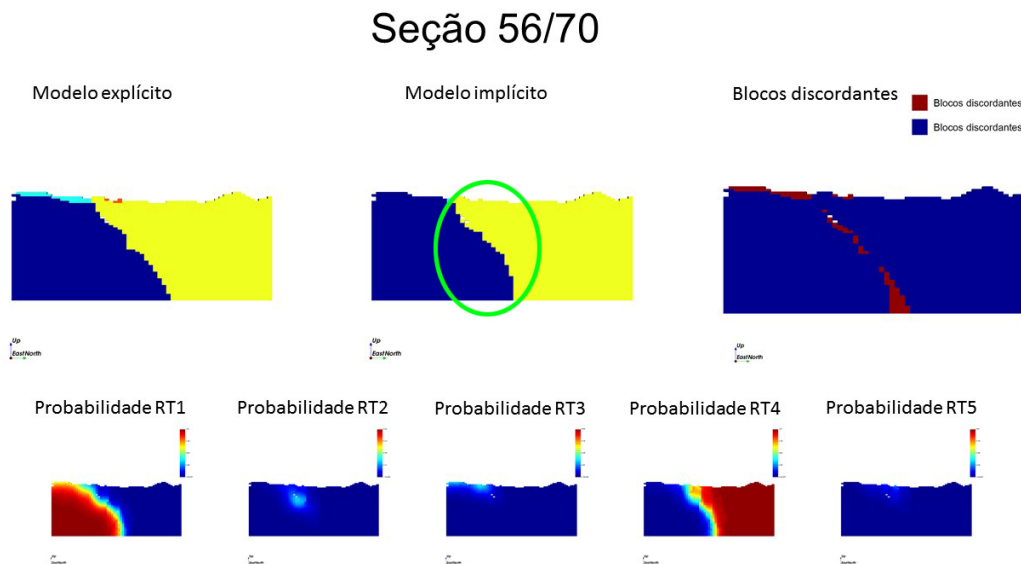
Na seção da [Figura 43](#), a região destacada pela elipse vermelha mostra uma estrutura desenhada pelo geomodelador no modelo explícito, com formas suaves e intrincadas, de difícil reprodução pelos métodos matemáticos, a não ser em casos de alta densidade amostral.

Figura 43 – Seção vertical 37/70 em x.



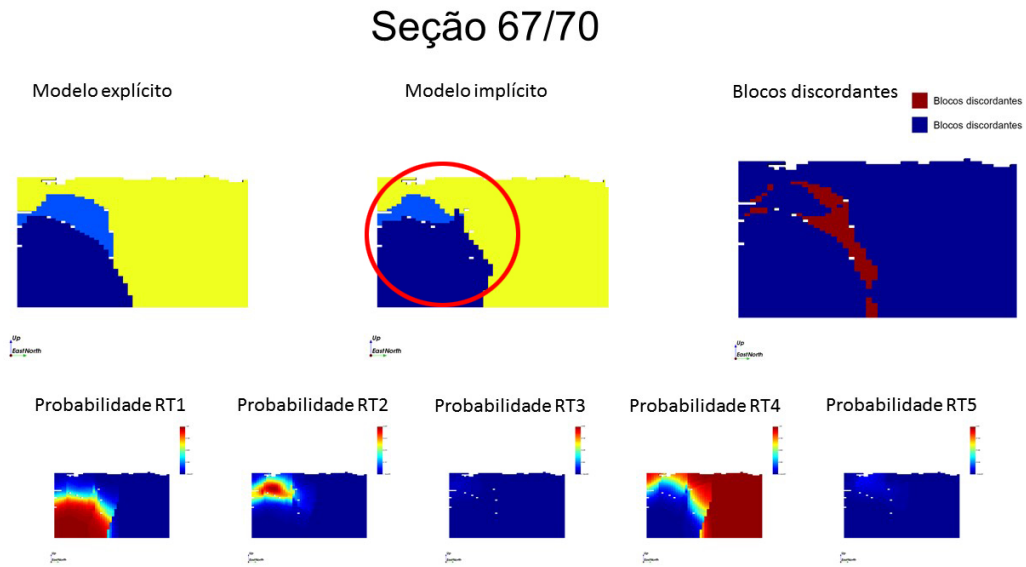
As estruturas presentes na seção 56 (Figura 44), são estruturas pouco complexas, e foram bem reproduzidas pelo algoritmo.

Figura 44 – Seção vertical 56/70 em x.



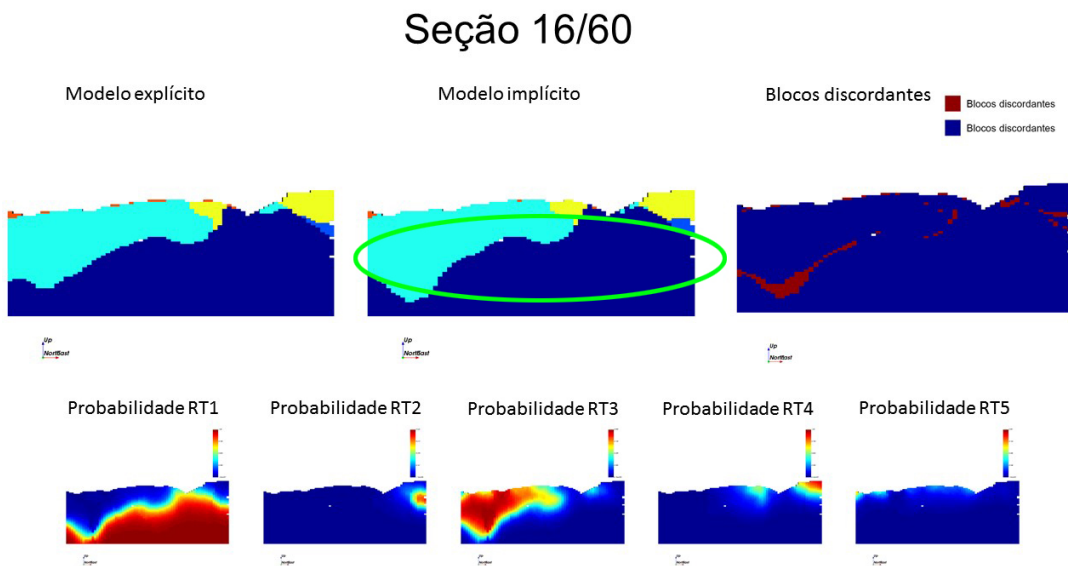
A seção da Figura 45, mostra destacado pela elipse vermelha, uma estrutura que não foi bem reproduzida, o algoritmo deu mais volume à litologia 1 e menos à litologia 2 em relação ao modelo criado explicitamente.

Figura 45 – Seção vertical 67/70 em x.



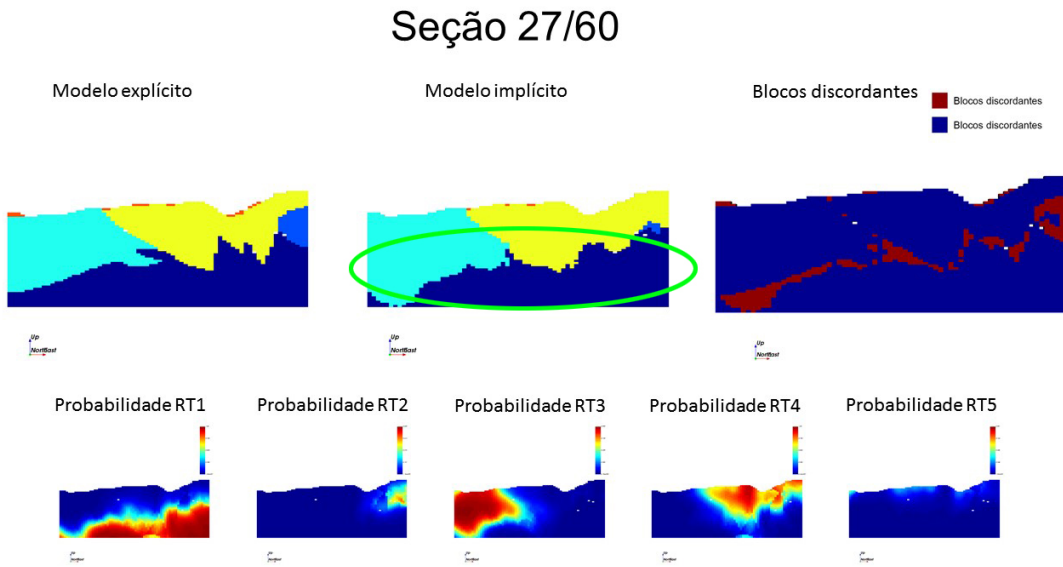
Na direção y, o modelo apresenta 60 seções. A seção 16 (Figura 46) exhibe poucos blocos discordantes, o algoritmo reproduziu satisfatoriamente as estruturas interpretadas pelo geomodelador.

Figura 46 – Seção vertical 16/60 em y.



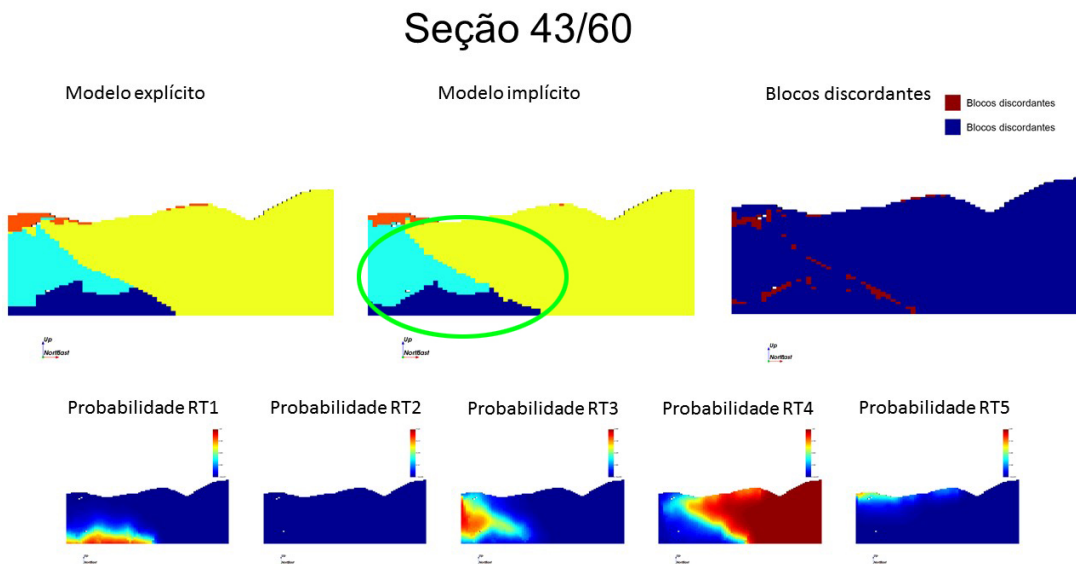
Embora as estruturas interpretadas exibidas na Figura 47 sejam bastante complexas, o algoritmo as reproduziu de forma razoável.

Figura 47 – Seção vertical 27/60 em y.



Novamente, o método implícito reproduziu de forma satisfatória as estruturas interpretadas na [Figura 48](#).

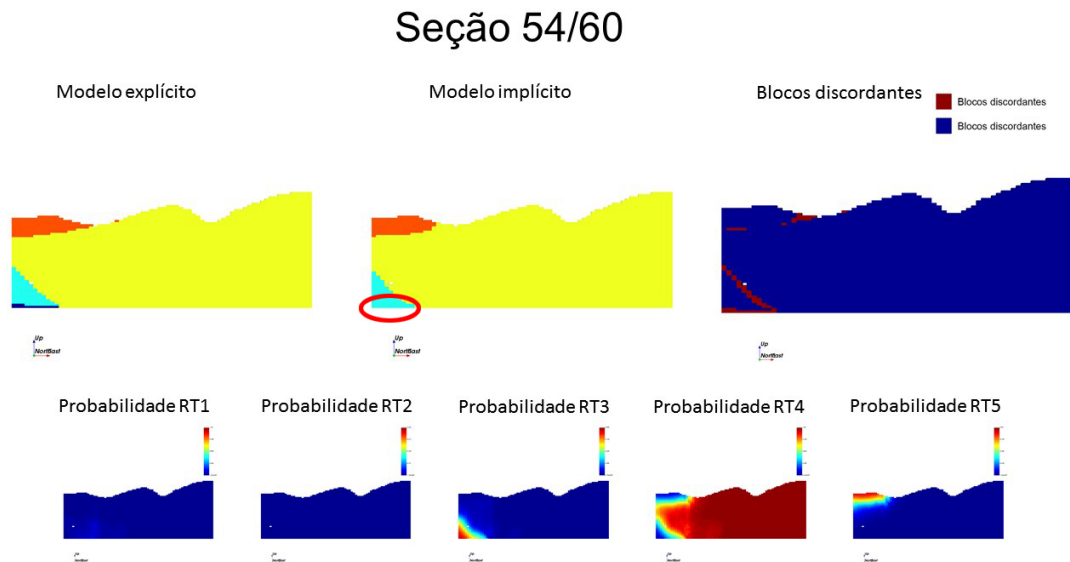
Figura 48 – Seção vertical 43/60 em y.



A seção da [Figura 49](#) mostra, na zona destacada em vermelho, blocos atribuídos com a litologia 1 pelo geomodelador enquanto no modelo implícito esses blocos são atribuídos com a litologia 3. Novamente, essa é uma região sem amostras de litologia 1, o

geomodelador, provavelmente, contou com sua experiência ou informação adicional para inferir a litologia desses blocos.

Figura 49 – Seção vertical 54/60 em y.



Nesse estudo de caso, a aplicação do *servo system* torna o modelo mais discordante em relação ao modelo de referência fornecido para comparação. Embora, as proporções dos dados ou dados desagrupados possam ser reproduzidas.

## 4.4 Discussão

A informação geológica é incorporada ao modelo através, não só do cálculo das distâncias assinaladas anisotrópicas, como também pelas direções de continuidade e efeito pepita dos variogramas. Limites suaves entre as litologias são garantidos pelo emprego da krigagem ordinária. Embora a complexidade geológica de pequena escala não seja bem reproduzida, devido ao efeito de suavização da krigagem, as grandes estruturas são bem representadas pelo algoritmo. A aplicação de um elipsoide de busca grande o suficiente e dados condicionantes em grandes quantidades aproximam os resultados da krigagem ordinária à krigagem global, evitando o surgimento de artefatos nos modelados gerados. O método implementado para múltiplos domínios geológicos evita sobreposição de litologias ou a necessidade de estabelecer qualquer tipo de ordem de prioridade para sua modelagem.

À primeira vista a modelagem dos variogramas para todas as categorias parece uma tarefa laboriosa e tediosa. Todavia, O comportamento contínuo das distâncias assinaladas os torna fáceis de modelar, e em muitos casos são semelhantes entre si, permitindo o uso de um mesmo modelo para todas as litologias.

Algoritmos de modelagem geológica implícita devem satisfazer seis critérios estabelecidos por McLennan (2007): simplicidade, velocidade, objetividade, integração de dados, avaliação de incertezas e realismo geológico. O método proposto foi avaliado à luz desses critérios, a partir de observações feitas ao longo desse trabalho e do trabalho de Silva (2015). Pontos positivos e negativos são aqui apresentados.

- Simplicidade: O algoritmo é de simples implementação e não envolve nenhuma equação matemática complicada. A ideia de uma distância assinalada para cada domínio é de fácil compreensão;
- Velocidade: O algoritmo é baseado em krigagem, que são métodos notadamente rápidos. A limitação é quanto ao número máximo de dados condicionantes usados na vizinhança de busca ao estimar cada bloco (acima de 200 tornam a execução bastante demorada);
- Objetividade: A abordagem implícita surgiu para eliminar a subjetividade dos métodos explícitos de modelagem. É esperado que modelos sejam reproduzidos com exatidão quando os parâmetros envolvidos nos cálculos sejam os mesmos. Os modelos não estão sujeitos à influência individual de cada geomodelador;
- Integração de dados: O algoritmo é flexível quanto à incorporação de novos dados do mesmo tipo. Entretanto, informação secundária (de outro tipo) não pode ser adicionada;
- Avaliação de incerteza: O algoritmo não avalia incerteza baseado em múltiplas realizações equiprováveis de modelos geológicos. Contudo, uma forma alternativa de medição heurística da incerteza, baseada na transformação das distâncias estimadas em probabilidades, é implementada no método.
- Realismo geológico: Modelos são considerados realísticos quando possuem concordância com a interpretação, e evidências geológicas coletadas pelo geomodelador (SILVA, 2015). A comparação da subseção 4.3.1, mostra que o algoritmo reproduz satisfatoriamente, dadas suas limitações, as estruturas interpretadas. A krigagem ordinária, base do algoritmo, possui efeito de suavização. Então, embora estruturas de larga escala sejam bem reproduzidas, a reprodução da complexidade geológica de pequena escala fica comprometida.

Um geomodelador experiente dificilmente será substituído por um código e um computador. Porém, o algoritmo proposto é de grande ajuda, como método complementar à modelagem explícita. Principalmente, em fases iniciais do projeto ou na criação de proto modelos que devem ser refinados, poupando tempo e trabalho. À medida que a densidade amostral aumenta, o modelo implícito se torna mais próximo do modelo explícito.



Por fim, as vantagens e desvantagens da metodologia apresentada são discutidas.

Vantagens:

- Os modelos podem ser atualizados com facilidade e rapidez conforme novos dados são obtidos. Não há necessidade de nova digitalização manual;
- Usando os mesmos parâmetros, os modelos são reproduzidos com exatidão, tornando a checagem e auditoria externa simples;
- Uma variedade de análises de sensibilidade pode ser aplicadas ao modelo, variando parâmetros envolvidos na interpolação e nos variogramas das distâncias assinaladas;
- O processo é rápido, poupando tempo e trabalho de geomodeladores na tediosa tarefa de construir e digitalizar manualmente seções verticais e horizontais.

Desvantagens:

- A interpolação das distâncias assinaladas pode impedir que um profissional treinado interprete estruturas e tendências da geologia que possam levar a melhores resultados;
- O algoritmo é baseado em krigagem, então, apenas relações lineares entre domínios são modeladas (caso não existam amostras suficientes). A modelagem de estruturas complexas e curvilíneas depende de outros métodos geoestatísticos. Além disso, a propriedade suavizadora da krigagem compromete a modelagem da geologia de pequena escala;
- O comportamento não estacionário das distâncias assinaladas torna a inferência do alcance dos variogramas arbitrária e questionável;
- Não avalia incerteza baseado em múltiplas realizações equiprováveis do modelo;
- O efeito de borda pode criar estruturas irreais nos limites do modelo.



## 5 Considerações finais

Essa dissertação propôs como meta investigar a aplicabilidade da modelagem geológica implícita com funções distância assinaladas como método substituto ou auxiliar aos métodos clássicos de modelagem. Para tanto, foram apresentados dois objetivos específicos:

1. Operacionalizar o método no software geoestatístico de código aberto *SGeMS*, desenvolvendo um *plug-in* funcional em *python*.
2. Conduzir um estudo de caso em um banco de dados real e verificar a qualidade do modelo gerado pela metodologia proposta, comparando-o a um modelo de referência.

Quanto ao primeiro objetivo, um *plug-in* foi desenvolvido e validado. É intuitivo e de fácil utilização. Uma alternativa mais amigável ao usuário em relação à rotina do *GSLib* desenvolvida por [Silva \(2015\)](#).

A partir do estudo de caso ([Capítulo 4](#)) foi possível concluir que a modelagem geológica implícita com funções distância assinaladas é um método simples, rápido e de grande auxílio ao geomodelador. Apesar de não substituir completamente os métodos explícitos, o método gerou um modelo geológico bastante semelhante ao modelo criado explicitamente, tomado como referência, em poucos minutos e que pode ser facilmente replicado, permitindo simples checagem e auditoria externa. Ainda é possível avaliar, mesmo que de uma forma heurística, a incerteza associada ao modelo geológico e reproduzir proporções, evitando a introdução de viés.

A modelagem geológica é uma etapa fundamental da avaliação depósitos minerais já que as estimativas e simulações dependem de domínios estacionários. Além disso, um bom modelo geológico pode trazer maior rentabilidade ao empreendimento, ou até mesmo ser crucial para a decisão de investir ou não baseada nos resultados do relatório de viabilidade, uma vez que a maior causa de fracasso nos empreendimentos mineiros é a falta de conhecimento a respeito do corpo mineralizado.

### 5.1 Recomendações para trabalhos futuros

Como recomendação para trabalhos futuros:

- Repetir o estudo de caso em banco de dados com menor densidade amostral e em depósitos que apresentem geologia mais complexa;

- Aprimorar o *plug-in* desenvolvido, se possível, reescrevendo-o em *C++*. Em especial o algoritmo do *servo system*, que pode se tornar muito demorado em banco de dados volumosos.
- Investigar outras formas, além da *softmax transformation*, para medir incertezas nos contatos geológicos. McLennan e Deutsch (2006) propuseram uma metodologia para quantificar incertezas através da técnica de *bootstrap*, enquanto Munroe (2012), Wilde e Deutsch (2011) propuseram calibrar uma banda de incertezas ao longo dos contatos entre os domínios;
- Investigar a viabilidade de simular as distâncias assinaladas, avaliando assim, a incerteza baseada em múltiplas realizações equiprováveis do modelo geológico;
- Estudar a possibilidade de agregar ao método informação secundária, usada pelo geomodelador para construir modelos explícitos;
- Implementar o *plug-in* independentemente de um *grid* de interpolação, permitindo a criação de modelos (*wireframes*) em qualquer resolução;
- Desenvolver técnicas híbridas de modelagem geológica, considerando as abordagens explícita, implícita e estocástica.

# Referências

- ARMSTRONG, M. *Basic linear geostatistics*. [S.l.]: Springer Science & Business Media, 1998. Citado na página 25.
- BLOOMENTHAL, J.; BAJAJ, C. *Introduction to implicit surfaces*. [S.l.]: Morgan Kaufmann, 1997. Citado na página 30.
- BRIDGE, J. S.; LEEDER, M. R. A simulation model of alluvial stratigraphy. *Sedimentology*, Wiley Online Library, v. 26, n. 5, p. 617–644, 1979. Citado na página 21.
- CALCAGNO, P. et al. Geological modelling from field data and geological knowledge: Part i. modelling method coupling 3d potential-field interpolation and geological rules. *Physics of the Earth and Planetary Interiors*, Elsevier, v. 171, n. 1, p. 147–157, 2008. Citado 5 vezes nas páginas 21, 39, 40, 41 e 42.
- CHILÈS, J.-P. et al. Modelling the geometry of geological units and its uncertainty in 3d from structural data: the potential-field method. In: *Proceedings of International Symposium on Orebody Modelling and Strategic Mine Planning, Perth, Australia*. [S.l.: s.n.], 2004. v. 22, p. 24. Citado 8 vezes nas páginas 20, 21, 38, 39, 40, 41, 42 e 43.
- COWAN, E. et al. Rapid geological modelling. *Applied Structural Geology for Mineral Exploration and Mining, International Symposium*, 2002. Citado 3 vezes nas páginas 21, 33 e 38.
- COWAN, E. et al. Practical implicit geological modelling. In: *Fifth International Mining Geology Conference*. [S.l.: s.n.], 2003. p. 17–19. Citado 3 vezes nas páginas 20, 21 e 33.
- DEUTSCH, C. V.; WILDE, B. J. Modeling multiple coal seams using signed distance functions and global kriging. *International Journal of Coal Geology*, n. 112, p. 87–93, 2013. Citado 4 vezes nas páginas 21, 22, 33 e 47.
- GALLI, A. et al. The pros and cons of the truncated gaussian method. In: *Geostatistical simulations*. [S.l.]: Springer, 1994. p. 217–233. Citado na página 21.
- GALLI, A.; MURILLO, E.; THOMANN, J. Dual kriging-its properties and its uses in direct contouring. *Geostatistics for Natural Resources Characterization, Part 2*, p. 621–634, 1984. Citado na página 34.
- GOOVAERTS, P. *Geostatistics for natural resources evaluation*. [S.l.]: Oxford University Press on Demand, 1997. Citado na página 59.
- GUARDIANO, F. B.; SRIVASTAVA, R. M. Multivariate geostatistics: beyond bivariate moments. In: *Geostatistics Troia'92*. [S.l.]: Springer, 1993. p. 133–144. Citado na página 21.
- HUSTRULID, W. A.; KUCHTA, M.; MARTIN, R. K. *Open Pit Mine Planning and Design, Two Volume Set & CD-ROM Pack*. [S.l.]: CRC Press, 2013. Citado 2 vezes nas páginas 19 e 20.

- JOURNAL, A.; ISAACS, E. Conditional indicator simulation: application to a saskatchewan uranium deposit. *Journal of the International Association for Mathematical Geology*, Springer, v. 16, n. 7, p. 685–718, 1984. Citado na página 21.
- JOURNAL, A. G. Nonparametric estimation of spatial distributions. *Journal of the International Association for Mathematical Geology*, Springer, v. 15, n. 3, p. 445–468, 1983. Citado na página 21.
- JOURNAL, A. G.; HUIJBREGTS, C. J. *Mining geostatistics*. [S.l.]: Academic press, 1978. Citado 3 vezes nas páginas 25, 26 e 27.
- JOYNT, P. *Leapfrog's structural trend*. 2014. Artigo publicado no blog leapfrog3d. Disponível em: <<http://blog.leapfrog3d.com/2014/06/04/leapfrogs-structural-trend/>>. Acesso em: 23-11-2016. Citado 3 vezes nas páginas 36, 37 e 38.
- LANE, R. *Predictions at locations where there are no measurements*. 2013. Artigo publicado no blog leapfrog3d. Disponível em: <<http://blog.leapfrog3d.com/2013/11/19/the-connection-between-radial-basis-functions-rbfs-and-kriging/>>. Acesso em: 23-11-2016. Citado 2 vezes nas páginas 34 e 35.
- LEAPFROG3D. *Leapfrog's fast RBF*. 2015. Artigo publicado no blog leapfrog3d. Disponível em: <<http://blog.leapfrog3d.com/2015/07/28/leapfrogs-fast-rbf/>>. Acesso em: 23-11-2016. Citado na página 35.
- LORENSEN, W. E.; CLINE, H. E. Marching cubes: A high resolution 3d surface construction algorithm. In: ACM. *ACM siggraph computer graphics*. [S.l.], 1987. v. 21, n. 4, p. 163–169. Citado na página 42.
- MALLET, J.-L. L. *Geomodeling*. [S.l.]: Oxford University Press, Inc., 2002. Citado na página 21.
- MATHERON, G. *Les variables régionalisées et leur estimation: une application de la théorie des fonctions aléatoires aux sciences de la nature*. [S.l.]: Masson et CIE, 1965. Citado na página 25.
- MCCULLAGH, P.; NELDER, J. A. *Generalized linear models*. [S.l.]: Chapman and Hall, 1989. Citado na página 51.
- MCLENNAN, J. A. *The decision of stationarity*. Tese (Doutorado) — University of Alberta, 2007. Citado 4 vezes nas páginas 20, 29, 30 e 78.
- MCLENNAN, J. A.; DEUTSCH, C. V. Implicit boundary modeling (boundsim). *CCG Annual Report*, 2006. Citado 4 vezes nas páginas 38, 42, 50 e 82.
- MCLENNAN, T. *Interpolant function in leapfrog geo*. 2013. Artigo publicado no blog leapfrog3d. Disponível em: <<http://blog.leapfrog3d.com/2013/07/26/interpolant-functions-in-leapfrog-geo/>>. Acesso em: 23-11-2016. Citado na página 36.
- MUNROE, M. J. A methodology for calculating tonnage uncertainty in vein-type deposits. In: *Masters Abstracts International*. [S.l.: s.n.], 2012. v. 51, n. 03. Citado 2 vezes nas páginas 50 e 82.
- NEUFELD, C.; WILDE, B. A global kriging program for artifact-free maps. *CCG Annual Report*, v. 7, p. 403–1, 2005. Citado na página 47.

- OSHER, S.; FEDKIW, R. *Level set methods and dynamic implicit surfaces*. [S.l.]: Springer-Verlag New York, Inc., 2003. Citado 3 vezes nas páginas 30, 31 e 32.
- PYRCZ, M. J.; CATUNEANU, O.; DEUTSCH, C. V. Stochastic surface-based modeling of turbidite lobes. *AAPG bulletin*, American Association of Petroleum Geologists, v. 89, n. 2, p. 177–191, 2005. Citado na página 21.
- RASERA, L. G. *Geoestatística de múltiplos pontos aplicada à simulação de modelos geológicos em grids estratigráficos*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul - PPGE3M, 2014. Citado 2 vezes nas páginas 20 e 30.
- RENARD, D. et al. Modeling the geometry of a mineral deposit domain with a potential field. In: *36th APCOM*. [S.l.: s.n.], 2013. p. Modeling—the. Citado 5 vezes nas páginas 21, 39, 40, 41 e 42.
- SAVCHENKO, V. V. et al. Function representation of solids reconstructed from scattered surface points and contours. In: WILEY ONLINE LIBRARY. *Computer Graphics Forum*. [S.l.], 1995. v. 14, n. 4, p. 181–188. Citado na página 33.
- SILVA, D. *Signed distance function modeling with multiple categories*. 2015. Artigo publicado no site geostatisticslessons.com. Disponível em: <<http://www.geostatisticslessons.com/lessons/signeddistancefunctions>>. Acesso em: 28-12-2016. Citado na página 50.
- SILVA, D.; DEUTSCH, C. Modeling multiple rock types with distance functions: methodology and software. In: *CCG Annual Report*. [S.l.: s.n.], 2012. Citado 5 vezes nas páginas 21, 45, 47, 48 e 50.
- SILVA, D.; DEUTSCH, C. Correcting distance function models to correct proportions. In: *CCG Annual Report*. [S.l.: s.n.], 2013. Citado na página 21.
- SILVA, D. M. A. *Enhanced geologic modeling with data-driven training images for improved resources and recoverable reserves*. Tese (Doutorado) — University of Alberta, 2015. Citado 14 vezes nas páginas 21, 22, 30, 45, 46, 47, 48, 49, 50, 51, 52, 65, 78 e 81.
- SPRAGG, k. *Interpolation and anisotropy*. 2013. Artigo publicado no blog leapfrog3d. Disponível em: <<http://blog.leapfrog3d.com/2013/10/18/interpolation-and-anisotropy/>>. Acesso em: 23-11-2016. Citado na página 36.
- SPRAGG, k. *Leapfrog interpolation basics*. 2013. Artigo publicado no blog leapfrog3d. Disponível em: <<http://blog.leapfrog3d.com/2013/05/08/leapfrog-interpolation-basics/>>. Acesso em: 23-11-2016. Citado na página 36.
- STRÉBELLE, S.; JOURNEL, A. Sequential simulation drawing structures from training images. Stanford University California, 2000. Citado na página 52.
- WILDE, B.; DEUTSCH, C. A new way to calibrate distance function uncertainty. In: *CCG Annual Report*. [S.l.: s.n.], 2011. Citado 4 vezes nas páginas 21, 48, 50 e 82.





# Apêndices



APÊNDICE A – Algoritmo em python que  
calcula as distâncias anisotrópicas assinaladas  
(*signed distances*)

```

1. #-----
2. #
3. # Signed Distance Function Calculator
4. # *****
5. #
6. # This SGeMS plugin calculates the anisotropic signed distances for each data
7. # point and each rock type
8. #
9. # AUTHOR: Roberto Mentzingen Rolo
10. #
11. #-----
12.
13. #!/bin/python
14. import sgems
15. import math
16. import numpy as np
17.
18. #Calculates the distances
19. def dist(x1, y1, z1, x2, y2, z2):
20.     return math.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2 + (z1 - z2) ** 2)
21.
22. #Defines the rotation and dilatation matrices
23. def rot(range1, range2, range3, azimuth, dip, rake, vetor):
24.
25.     if azimuth >= 0 and azimuth <=270:
26.         alpha = math.radians(90-azimuth)
27.     else:
28.         alpha = math.radians(450-azimuth)
29.         beta = -math.radians(dip)
30.         phi = math.radians(rake)
31.
32.     rot_matrix = np.zeros((3,3))
33.
34.     rot_matrix[0,0] = math.cos(beta)*math.cos(alpha)
35.     rot_matrix[0,1] = math.cos(beta)*math.sin(alpha)
36.     rot_matrix[0,2] = -math.sin(beta)
37.     rot_matrix[1,0] = (range1/range2)*(-
math.cos(phi)*math.sin(alpha)+math.sin(phi)*math.sin(beta)*math.cos(alpha))
38.     rot_matrix[1,1] = (range1/range2)*(math.cos(phi)*math.cos(alpha)+math.sin(phi)*math.si
n(beta)*math.sin(alpha))
39.     rot_matrix[1,2] = (range1/range2)*(math.sin(phi)*math.cos(beta))
40.     rot_matrix[2,0] = (range1/range3)*(math.sin(phi)*math.sin(alpha)+math.cos(phi)*math.si
n(beta)*math.cos(alpha))
41.     rot_matrix[2,1] = (range1/range3)*(-
math.sin(phi)*math.cos(alpha)+math.cos(phi)*math.sin(beta)*math.sin(alpha))
42.     rot_matrix[2,2] = (range1/range3)*(math.cos(phi)*math.cos(beta))
43.
44.     vetor = np.array(vetor)
45.
46.     return np.dot(rot_matrix, vetor)
47.
48. #Transform the data with the rotation/dilatation matrices
49. def anis_search(X, Y, Z, range1, range2, range3, azimuth, dip, rake):
50.
51.     X_linha = []
52.     Y_linha = []
53.     Z_linha = []
54.
55.     for i in range(len(X)):
56.         vet = [X[i],Y[i],Z[i]]
57.
58.         vet_rot = rot(range1, range2, range3, azimuth, dip, rake, vet)
59.
60.         X_linha.append(vet_rot[0])
61.         Y_linha.append(vet_rot[1])
62.         Z_linha.append(vet_rot[2])

```

```

63.
64.     return X_linha, Y_linha, Z_linha
65.
66. #Shows every parameter of the plugin in the command pannel
67. def read_params(a,j=''):
68.     for i in a:
69.         if (type(a[i])!=type({'a':1})):
70.             print j+"['"+str(i)+"']="+str(a[i])
71.         else:
72.             read_params(a[i],j+"['"+str(i)+"'"])
73.
74. class signed_distances:
75.     def __init__(self):
76.         pass
77.
78.     def initialize(self, params):
79.         self.params = params
80.         return True
81.
82.     def execute(self):
83.
84.         '''#Execute the funtion read_params
85.         read_params(self.params)
86.         print self.params'''
87.
88.         #Get the grid and rock type property
89.         grid = self.params['propertyselectornoregion']['grid']
90.         prop = self.params['propertyselectornoregion']['property']
91.
92.         #Error message
93.         if len(grid) == 0 or len(prop) == 0:
94.             print 'Select the rocktype property'
95.             return False
96.
97.         #Get the X, Y and Z coordinates and RT property
98.         X = sgems.get_property(grid, '_X_')
99.         Y = sgems.get_property(grid, '_Y_')
100.        Z = sgems.get_property(grid, '_Z_')
101.        RT = sgems.get_property(grid, prop)
102.
103.        elipsoide = self.params['ellipsoidinput']['value']
104.        elipsoide_split = elipsoide.split()
105.
106.        range1 = float(elipsoide_split[0])
107.        range2 = float(elipsoide_split[1])
108.        range3 = float(elipsoide_split[2])
109.
110.        azimuth = float(elipsoide_split[3])
111.        dip = float(elipsoide_split[4])
112.        rake = float(elipsoide_split[5])
113.
114.        X, Y, Z = anis_search(X, Y, Z, range1, range2, range3, azimuth, dip, rake)
115.
116.        #Creates a list of all rock types
117.        rt_list = []
118.        for i in RT:
119.            if i not in rt_list and not math.isnan(i):
120.                rt_list.append(i)
121.
122.        #Sort the rock type list in crescent order
123.        rt_list = [int(x) for x in rt_list]
124.        rt_list.sort()
125.
126.        #Create a empty distance matrix
127.        dist_matrix = np.zeros(shape = ((len(rt_list)), (len(RT))))
128.
129.        #Calculates the signed distances, and append it in the distance matrix
130.        for i in range(len(rt_list)):

```

```

131.         rock = rt_list[i]
132.
133.         for j in range(len(RT)):
134.
135.             if math.isnan(RT[j]):
136.                 dist_matrix[i][j] = float('nan')
137.
138.             elif RT[j] == rock:
139.                 dsmin = 1.0e21
140.
141.                 for k in range(len(RT)):
142.
143.                     if RT[j] != RT[k] and not math.isnan(RT[k]):
144.                         if (dist(X[j], Y[j], Z[j], X[k], Y[k], Z[k])) < dsmin:
145.
146.                             dsmin = (dist(X[j], Y[j], Z[j], X[k], Y[k], Z[k]))
147.
148.                             dist_matrix[i][j] = -dsmin
149.
150.             else:
151.                 dsmin = 1.0e21
152.
153.                 for k in range(len(RT)):
154.
155.                     if RT[k] == rock:
156.                         if (dist(X[j], Y[j], Z[j], X[k], Y[k], Z[k])) < dsmin:
157.
158.                             dsmin = (dist(X[j], Y[j], Z[j], X[k], Y[k], Z[k]))
159.
160.                             dist_matrix[i][j] = dsmin
161.
162.                 #Creates the signed distances properties
163.                 lst_props_grid=sgems.get_property_list(grid)
164.
165.                 for k in range(len(dist_matrix)):
166.                     prop_final_data_name = 'Signed_Distances_RT_' + str(rt_list[k])
167.
168.                     if (prop_final_data_name in lst_props_grid):
169.                         flag=0
170.                         i=1
171.                         while (flag==0):
172.                             test_name=prop_final_data_name+'-'+str(i)
173.                             if (test_name not in lst_props_grid):
174.                                 flag=1
175.                                 prop_final_data_name=test_name
176.                                 i=i+1
177.
178.                     list = dist_matrix[k].tolist()
179.                     sgems.set_property(grid, prop_final_data_name, list)
180.
181.                 return True
182.
183.     def finalize(self):
184.         return True
185.
186.     def name(self):
187.         return "signed_distances"
188.
189.     #####
190.     def get_plugins():
191.         return ["signed_distances"]

```

APÊNDICE B – Algoritmo em python que  
interpola as distancias assinaladas e cria o  
modelo geológico (*interpolator*)

```

1. #-----
2. #
3. # Signed Distances Function Interpolator
4. # *****
5. #
6. # This SGeMS plugin interpolates (OK) the signed distance function calculated
7. # for each data and rock type, and creates a geologic model based on the minimum
8. # estimated distance.
9. #
10. # AUTHOR: Roberto Mentzingen Rolo
11. #
12. #-----
13.
14. #!/bin/python
15. import sgems
16. import math
17. import numpy as np
18. import random
19. import copy
20.
21. #Creates a random path given the size of the grid
22. def random_path(prop):
23.     nodes_not_nan = []
24.     for i in range(len(prop)):
25.         if not math.isnan(prop[i]):
26.             nodes_not_nan.append(i)
27.     random.shuffle(nodes_not_nan)
28.     return nodes_not_nan
29.
30. #Calculates the proportions of variables on a list
31. def proportion(var, RT):
32.     rock_types = []
33.     target_prop = []
34.     for k in range(len(RT)):
35.         target_prop.append(0)
36.         rock_types.append(int(RT[k][-1]))
37.     rock_types.sort()
38.     var_not_nan = []
39.     for i in var:
40.         if not math.isnan(i):
41.             var_not_nan.append(i)
42.     for i in range(len(rock_types)):
43.         target_prop[i] = float(var.count(rock_types[i]))/len(var_not_nan)
44.     return target_prop
45.
46. #Transform i,j,k in n
47. def ijk_in_n(grid, i, j, k):
48.     dims = sgems.get_dims(grid)
49.     n = k*dims[0]*dims[1]+j*dims[0]+i
50.     return n
51.
52. #Crestes a list with indices of the neighbors valid blocks
53. def neighb(grid, indice):
54.     ijk = sgems.get_ijk(grid, indice)
55.     neighborhood = []
56.     for i in range(ijk[0]-1,ijk[0]+2):
57.         for j in range(ijk[1]-1,ijk[1]+2):
58.             for k in range(ijk[2]-1,ijk[2]+2):
59.                 ijk_blk = [i,j,k]
60.                 neighborhood.append(ijk_blk)
61.     dims = sgems.get_dims(grid)
62.     neighborhood_cp = copy.copy(neighborhood)
63.     for i in neighborhood_cp:
64.         if dims[2] == 1:
65.             if i[0] < 0 or i[1] < 0:
66.                 neighborhood.remove(i)

```



```

67.         elif i[0] > (dims[0] - 1) or i[1] > (dims[1] - 1):
68.             neighborhood.remove(i)
69.         elif i[2] != 0:
70.             neighborhood.remove(i)
71.         elif i == sgems.get_ijk(grid, indice):
72.             neighborhood.remove(i)
73.     else:
74.         if i[0] < 0 or i[1] < 0 or i[2] < 0:
75.             neighborhood.remove(i)
76.         elif i[0] > (dims[0] - 1) or i[1] > (dims[1] - 1) or i[2] > (dims[2] - 1):
77.             neighborhood.remove(i)
78.         elif i == sgems.get_ijk(grid, indice):
79.             neighborhood.remove(i)
80.     neighborhood_n = []
81.     for i in neighborhood:
82.         neighborhood_n.append(ijk_in_n(grid,i[0],i[1],i[2]))
83.     return neighborhood_n
84.
85.
86. # Shows every parameter of the plugin in the command pannel
87. def read_params(a, j=''):
88.     for i in a:
89.         if (type(a[i]) != type({'a': 1})):
90.             print j + "[" + str(i) + "]" = " + str(a[i])
91.         else:
92.             read_params(a[i], j + "[" + str(i) + "]")
93.
94. class interpolator:
95.     def __init__(self):
96.         pass
97.
98.     def initialize(self, params):
99.         self.params = params
100.        return True
101.
102.        def execute(self):
103.
104.            '''# Execute the funtion read_params
105.            read_params(self.params)
106.            print self.params'''
107.
108.            #Get the grid and rock type property
109.            grid = self.params['propertyselectornoregion']['grid']
110.            prop = self.params['propertyselectornoregion']['property']
111.
112.            #Get the X, Y and Z coordinates and RT property
113.            X = sgems.get_property(grid, '_X_')
114.            Y = sgems.get_property(grid, '_Y_')
115.            Z = sgems.get_property(grid, '_Z_')
116.            RT_data = sgems.get_property(grid, prop)
117.
118.            # Getting properties
119.            grid_krig = self.params['gridselectorbasic_2']['value']
120.            grid_var = self.params['gridselectorbasic']['value']
121.            props = (self.params['orderedpropertyselector']['value']).split(';')
122.            n_var = int(self.params['indicator_regionalization_input']['number_of_indic
ator_group'])
123.            n_prop = int(self.params['orderedpropertyselector']['count'])
124.            min_cond = self.params['spinBox_2']['value']
125.            max_cond = self.params['spinBox']['value']
126.
127.            # Error messages
128.            if len(grid_var) == 0 or len(grid_krig) == 0:
129.                print 'Select the variables'
130.                return False
131.
132.            if n_var != n_prop:

```

```

133.         print 'Number of variables and number of variograms models are diferent
134.     .
135.         return False
136.
137.         #Creating an empty list to store the interpolated distances
138.         SG_OK_list = []
139.
140.         # Loop in every variable
141.         for i in xrange(0, n_var):
142.
143.             # Getting variables
144.             prop_HD = props[i]
145.             prop_name = "Interpolated_" + str(prop_HD)
146.             prop_name_var = "Interpolated_" + str(prop_HD) + ' krig_var'
147.             var_str = ''
148.             indicator_group = "Indicator_group_" + str(i + 1)
149.             elipsoide = self.params['ellipsoidinput']['value']
150.             n_struct = int(self.params['indicator_regionalization_input'][indicator_group]['Covariance_input']['structures_count'])
151.
152.             # Error message
153.             if n_struct == 0:
154.                 print 'Variogram have no structures'
155.                 return False
156.
157.             # Loop in every variogram structure
158.             for j in xrange(0, n_struct):
159.                 # Getting variogram parameters
160.                 Structure = "Structure_" + str(j + 1)
161.
162.                 cov_type = self.params['indicator_regionalization_input'][indicator_group]['Covariance_input'][Structure]['Two_point_model']['type']
163.
164.                 cont = self.params['indicator_regionalization_input'][indicator_group]['Covariance_input'][Structure]['Two_point_model']['contribution']
165.
166.                 if cov_type == 'Nugget Covariance':
167.                     #Writing variogram parameters on a variable in nugget effect case
168.                     var_str = var_str + '<{} type="{}"> <Two_point_model contribution="{}" type="{}" > </Two_point_model> </Structure_1> '.format(Structure, 'Covariance', cont, cov_type, Structure)
169.
170.                 else:
171.                     range1 = self.params['indicator_regionalization_input'][indicator_group]['Covariance_input'][Structure]['Two_point_model']['ranges']['range1']
172.                     range2 = self.params['indicator_regionalization_input'][indicator_group]['Covariance_input'][Structure]['Two_point_model']['ranges']['range2']
173.                     range3 = self.params['indicator_regionalization_input'][indicator_group]['Covariance_input'][Structure]['Two_point_model']['ranges']['range3']
174.
175.                     rake = self.params['indicator_regionalization_input'][indicator_group]['Covariance_input'][Structure]['Two_point_model']['angles']['rake']
176.                     dip = self.params['indicator_regionalization_input'][indicator_group]['Covariance_input'][Structure]['Two_point_model']['angles']['dip']
177.                     azimuth = self.params['indicator_regionalization_input'][indicator_group]['Covariance_input'][Structure]['Two_point_model']['angles']['azimuth']
178.
179.                     # Writing variogram parameters on a variable in other cases
180.                     var_str = var_str + '<{} type="{}"> <Two_point_model contribution="{}" type="{}" > <ranges range1="{}" range2="{}" range3="{}" /> <angles azimuth="{}" dip="{}" rake="{}" /> </Two_point_model> </{}> '.format(Structure, 'Covariance', cont, cov_type, range1, range2, range3, azimuth, dip, rake, Structure)
181.
182.             # Calling ordinary kriging for each variable, using the variograms parameters above
183.             sgems.execute('RunGeostatAlgorithm kriging::GeostatParamUtils/XML::<p arameters> <algorithm name="kriging" /> <Variogram structures_count="{}" > {} </Variogram> <ouput_kriging_variance value="1" /> <output_n_samples_ value="0" />

```

```

        <output_average_distance value="0" />    <output_sum_weights value="0" />    <output_sum_positive_weights value="0" />    <output_lagrangian value="0" />    <Nb_processors value="2" />    <Grid_Name value="{}" region="" />    <Property_Name value="{}" />    <Hard_Data grid="{}" property="{}" region="" />    <Kriging_Type type="Ordinary Kriging (OK)" >    <parameters />    </Kriging_Type>    <do_block_kriging value="1" />    <npoint_s_x value="5" />    <npoints_y value="5" />    <npoints_z value="5" />    <Min_Conditioning_Data value="{}" />    <Max_Conditioning_Data value="{}" />    <Search_Ellipsoid value="{}" />    <AdvancedSearch use_advanced_search="0"></AdvancedSearch>    </parameters>'.format(n_struct, var_str, grid_krig, prop_name, grid_var, prop_HD, min_cond, max_cond, elipsoide))
183.
184.         SG_OK_list.append(sgems.get_property(grid_krig, prop_name))
185.
186.         #Deleting kriged distances
187.         sgems.execute('DeleteObjectProperties {}:{}'.format(grid_krig, prop_name))
188.         sgems.execute('DeleteObjectProperties {}:{}'.format(grid_krig, prop_name_var))
189.
190.         RT = (self.params['orderedpropertyselector']['value']).split(';')
191.
192.         #Determining geomodel based on minimum estimated signed distance function
193.         GeoModel = SG_OK_list[0][:]
194.
195.         t = 0
196.         for i in range(len(SG_OK_list[0])):
197.             sgmin = 10e21
198.             for j in range(len(SG_OK_list)):
199.                 if SG_OK_list[j][i] < sgmin:
200.                     sgmin = SG_OK_list[j][i]
201.                     t = j
202.             if math.isnan(SG_OK_list[j][i]):
203.                 GeoModel[i] = float('nan')
204.             else:
205.                 GeoModel[i] = (int(RT[t][-1]))
206.
207.         #Creating GeoModel property
208.         lst_props_grid=sgems.get_property_list(grid_krig)
209.         prop_final_data_name = 'Geologic_Model'
210.
211.         if (prop_final_data_name in lst_props_grid):
212.             flag=0
213.             i=1
214.             while (flag==0):
215.                 test_name=prop_final_data_name+'-'+str(i)
216.                 if (test_name not in lst_props_grid):
217.                     flag=1
218.                     prop_final_data_name=test_name
219.                     i=i+1
220.
221.         #Assign conditioning data to grid node
222.         for i in range(len(RT_data)):
223.             if not math.isnan(RT_data[i]):
224.                 closest_node = sgems.get_closest_nodeid(grid_krig, X[i],Y[i],Z[i])
225.
226.                 GeoModel[closest_node] = RT_data[i]
227.
228.         sgems.set_property(grid_krig, prop_final_data_name, GeoModel)
229.
230.         #Operating softmax transformation
231.         if self.params['softmax_check']['value']=='1':
232.             gamma =float( self.params['Gamma']['value'])
233.             Prob_list = SG_OK_list[:]
234.
235.             for i in range(len(SG_OK_list[0])):
236.                 soma = 0
237.                 for j in range(len(SG_OK_list)):

```

```

238.         soma = soma + math.exp(-SG_OK_list[j][i]/gamma)
239.     for j in range(len(SG_OK_list)):
240.         Prob_list[j][i] = math.exp(-SG_OK_list[j][i]/gamma)/soma
241.
242.     #Creating probabilities properties
243.     for k in range(len(Prob_list)):
244.         prop_final_data_name = 'Probability_RT'+str(RT[k][-1])
245.
246.         if (prop_final_data_name in lst_props_grid):
247.             flag=0
248.             i=1
249.             while (flag==0):
250.                 test_name=prop_final_data_name+'-'+str(i)
251.                 if (test_name not in lst_props_grid):
252.                     flag=1
253.                     prop_final_data_name=test_name
254.                     i=i+1
255.
256.         sgems.set_property(grid_krig, prop_final_data_name, Prob_list[k])
257.
258.     #Operating servo-system
259.     if self.params['servo_check']['value'] == '1':
260.         var_rt_grid = self.params['targe_prop']['grid']
261.         var_rt_st = self.params['targe_prop']['property']
262.         var_rt_region = self.params['targe_prop']['region']
263.         if len(var_rt_grid) == 0 or len(var_rt_st) == 0:
264.             print 'Select the target proportion property'
265.             return False
266.
267.     #Getting variables
268.     var_rt = sgems.get_property(var_rt_grid, var_rt_st)
269.
270.     #Getting parameters
271.     lambda1 = float(self.params['Lambda']['value'])
272.     mi = lambda1/(1-lambda1)
273.
274.     #Checking if a region exist
275.     if len(var_rt_region) == 0:
276.         #Variable without a region
277.         var_region = var_rt
278.
279.     else:
280.         region_rt = sgems.get_region(var_rt_grid, var_rt_region)
281.         #Getting the variable inside the region
282.         var_region = []
283.         for i in range(len(var_rt)):
284.             if region_rt[i] == 1:
285.                 var_region.append(var_rt[i])
286.
287.     #Getting the target proportion
288.     target_prop = proportion(var_region, RT)
289.
290.     #Getting the random path
291.     ran_path = random_path(Prob_list[0])
292.
293.     #Removing the blocks outside the region from randon path
294.     if len(var_rt_region) != 0:
295.         for i in range(len(region_rt)):
296.             if region_rt[i] == 0:
297.                 ran_path.remove(i)
298.
299.     #servo system
300.     p = 0
301.     GeoModel_corrected = GeoModel[:]
302.
303.     visited_rts = []
304.     for j in ran_path:
305.         visited_rts.append(GeoModel[j])
306.         instant_proportions = proportion(visited_rts,RT)

```

```

307.
308.             sgmax = 10e-21
309.             for i in range(len(Prob_list)):
310.                 Prob_list[i][j] = Prob_list[i][j] + (mi * (target_prop[i] -
instant_proportions[i]))
311.                 if Prob_list[i][j] > sgmax:
312.                     sgmax = Prob_list[i][j]
313.                     p = i
314.
315.                 GeoModel_corrected[j] = int(RT[p][-1])
316.                 visited_rts[-1] = int(RT[p][-1])
317.
318.             #Correcting servo servo-
system by the biggest proportion on a neighborhood
319.             GeoModel_corrected_servo_prop = GeoModel_corrected[:]
320.             ran_path_servo_correction = random_path(GeoModel_corrected_servo_pr
op)
321.             for i in ran_path_servo_correction:
322.                 vizinhanca = neighb(grid_krig,i)
323.
324.                 blk_geo_model_corrected_servo = []
325.                 for j in vizinhanca:
326.                     blk_geo_model_corrected_servo.append(GeoModel_corrected_ser
vo_prop[j])
327.
328.                 proportions_servo = proportion(blk_geo_model_corrected_servo, R
T)
329.                 indice_max_prop = proportions_servo.index(max(proportions_servo
))
330.
331.                 GeoModel_corrected_servo_prop[i] = int(RT[indice_max_prop][-
1])
332.
333.             #Creating Geologic_Model_Servo_System property
334.             prop_final_data_name = 'Geologic_Model_Servo_System'
335.
336.             if (prop_final_data_name in lst_props_grid):
337.                 flag=0
338.                 i=1
339.                 while (flag==0):
340.                     test_name=prop_final_data_name+'-'+str(i)
341.                     if (test_name not in lst_props_grid):
342.                         flag=1
343.                         prop_final_data_name=test_name
344.                         i=i+1
345.
346.             #Creating Geologic_Model_Corrected property
347.             prop_final_data_name1 = 'Geologic_Model_Corrected'
348.
349.             if (prop_final_data_name1 in lst_props_grid):
350.                 flag=0
351.                 i=1
352.                 while (flag==0):
353.                     test_name1=prop_final_data_name1+'-'+str(i)
354.                     if (test_name1 not in lst_props_grid):
355.                         flag=1
356.                         prop_final_data_name1=test_name1
357.                         i=i+1
358.
359.             #Assign conditioning data to grid node
360.             for i in range(len(RT_data)):
361.                 if not math.isnan(RT_data[i]):
362.                     closest_node = sgems.get_closest_nodeid(grid_krig, X[i],Y[i
],Z[i])
363.                     GeoModel_corrected[closest_node] = RT_data[i]
364.                     GeoModel_corrected_servo_prop[closest_node] = RT_data[i]
365.
366.             #Setting properties

```

```
367.             sgems.set_property(grid_krig, prop_final_data_name, GeoModel_correc
ted)
368.             sgems.set_property(grid_krig, prop_final_data_name1, GeoModel_corre
cted_servo_prop)
369.
370.             return True
371.
372.         def finalize(self):
373.
374.             return True
375.
376.         def name(self):
377.
378.             return "interpolator"
379.
380.         #####
381.         def get_plugins():
382.             return ["interpolator"]
```

# Anexos





# ANEXO A – Arquivo de parâmetro usado na validação

Parameters for DFMOD

\*\*\*\*\*

START OF PARAMETERS:

```

cleaned_samples.prn      - file with data
0  1  2  3  4           - columns for DH,X,Y,Z,var
-1.0    1.0e21         - trimming limits
5                               - number of categories
2 3 4 5 6             - codes
dfmod.out              - file for distance function output
rtmod_2.out           - file for rock type output
51    470170    25     - nx,xmn,xsiz
48    6936500   25     - ny,ymn,ysiz
89    3115      15     - nz,zmn,zsiz
5    5    5         - x,y and z block discretization
4    32           - min, max data for kriging
0                               - max per octant (0-> not used)
1000  1000  1000     - maximum search radii
0.0   0.0   0.0     - angles for search ellipsoid
1    2.302         - 0=SK,1=OK
2    100           -nst, nugget effect
3    3000  0.0  0.0  0.0 -it,cc,ang1,ang2,ang3
      280  280  820.0 -a_hmax, a_hmin, a_vert
3    40000 0.0  0.0  0.0 -it,cc,ang1,ang2,ang3
      9999999999 9999999999 820 -a_hmax, a_hmin, a_vert
2    100           -nst, nugget effect
3    7000  0.0  0.0  0.0 -it,cc,ang1,ang2,ang3
      400  400  820.0 -a_hmax, a_hmin, a_vert
3    20000 0.0  0.0  0.0 -it,cc,ang1,ang2,ang3
      9999999999 9999999999 820 -a_hmax, a_hmin, a_vert
2    100           -nst, nugget effect
3    4000  0.0  0.0  0.0 -it,cc,ang1,ang2,ang3
      300  300  400.0 -a_hmax, a_hmin, a_vert

```

```
3 15000 0.0 0.0 0.0 -it,cc,ang1,ang2,ang3
    900 900 9999999999 -a_hmax, a_hmin, a_vert
2 100 -nst, nugget effect
3 4000 0.0 0.0 0.0 -it,cc,ang1,ang2,ang3
    250 250 500.0 -a_hmax, a_hmin, a_vert
3 15000 0.0 0.0 0.0 -it,cc,ang1,ang2,ang3
    820 820 9999999999 -a_hmax, a_hmin, a_vert
2 100 -nst, nugget effect
3 4000 0.0 0.0 0.0 -it,cc,ang1,ang2,ang3
    320 320 500.0 -a_hmax, a_hmin, a_vert
3 15000 0.0 0.0 0.0 -it,cc,ang1,ang2,ang3
    820 820 9999999999 -a_hmax, a_hmin, a_vert
```