



Ministério da  
**Ciência, Tecnologia  
e Inovação**



sid.inpe.br/mtc-m19/2012/11.19.16.34-TDI

## **REDES NEURAIS PARA ASSIMILAÇÃO DE DADOS EM UM MODELO DE CIRCULAÇÃO OCEÂNICA**

Helaine Cristina Morais Furtado

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Elbert Einstein Nehrer Macau, Haroldo Fraga de Campos Velho, e Rosângela Saher Corrêa Cintra, aprovada em 07 de dezembro de 2012.

URL do documento original:

<http://urlib.net/8JMKD3MGP7W/3D3J5QS>

INPE  
São José dos Campos  
2012

**PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

**CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):****Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

**Membros:**

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr<sup>a</sup> Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr<sup>a</sup> Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

**BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

**REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

**EDITORAÇÃO ELETRÔNICA:**

Maria Tereza Smith de Brito - Serviço de Informação e Documentação (SID)



Ministério da  
**Ciência, Tecnologia  
e Inovação**



sid.inpe.br/mtc-m19/2012/11.19.16.34-TDI

## **REDES NEURAIS PARA ASSIMILAÇÃO DE DADOS EM UM MODELO DE CIRCULAÇÃO OCEÂNICA**

Helaine Cristina Morais Furtado

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Elbert Einstein Nehrer Macau, Haroldo Fraga de Campos Velho, e Rosângela Saher Corrêa Cintra, aprovada em 07 de dezembro de 2012.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3D3J5QS>>

INPE  
São José dos Campos  
2012

Dados Internacionais de Catalogação na Publicação (CIP)

---

Furtado, Helaine Cristina Morais.

F984r      Redes neurais para assimilação de dados em um modelo de circulação oceânica / Helaine Cristina Morais Furtado. – São José dos Campos : INPE, 2012.

xxiv + 149 p. ; (sid.inpe.br/mtc-m19/2012/11.19.16.34-TDI)

Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2012.

Orientadores : Drs. Elbert Einstein Nehrer Macau, Haroldo Fraga de Campos Velho e Rosângela Saher Corrêa Cintra.

1. assimilação de dados. 2. redes neurais artificiais. 3. água rasa 2D. 4. método representante. 5. filtro de kalman. I.Título.

CDU 004.032.26

---

Copyright © 2012 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente com o propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2012 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming, or otherwise, without written permission from INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

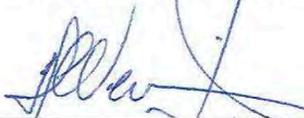
Aprovado (a) pela Banca Examinadora  
em cumprimento ao requisito exigido para  
obtenção do Título de **Doutor(a)** em  
**Computação Aplicada**

Dr. Ezzat Selim Chalhoub



Presidente / INPE / SJC Campos - SP

Dr. Haroldo Fraga de Campos Velho



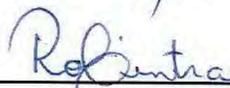
Orientador(a) / INPE / São José dos Campos - SP

Dr. Elbert Einstein Nehrer Macau



Orientador(a) / INPE / São José dos Campos - SP

Dra. Rosângela Saher Corrêa Cintra



Orientador(a) / INPE / São José dos Campos - SP

Dr. Joaquim José Barroso de Castro



Membro da Banca / INPE / SJC Campos - SP

Dr. Clemente Augusto Souza Tanajura



Convidado(a) / UFBA / Salvador - BA

Dra. Renata Hax Sander Reiser



Convidado(a) / UFPEL / Cidade de Pelotas - RS

Este trabalho foi aprovado por:

maioria simples

unanimidade

Aluno (a): **Helaine Cristina Moraes Furtado**

São José dos Campos, 07 de Dezembro de 2012



*A meus pais João e Lucilene, a meus irmãos Frank, Heloisa,  
Neto, Helen e Lilian*



## AGRADECIMENTOS

O processo realizado para alcançar o objetivo foi tão importante quanto o resultado adquirido. A trajetória percorrida me proporcionou conhecimento em vários aspectos. Além do conhecimento científico, o aprendizado do cotidiano, que muitas vezes exige superação e persistência.

Agradeço a Deus por mais uma conquista concedida. Obrigado Senhor por iluminar os meus caminhos e me proporcionar inteligência e discernimento para prosseguir.

Ao professor Dr. Haroldo Fraga de Campos Velho, muito obrigada por seu otimismo, entusiasmo e paciência. Ao professor Dr. Elbert E. N. Macau, obrigada por ter me mostrado a possibilidade de fazer a pós-graduação no INPE. A Dra. Rosângela Corrêa Saher Cintra meus agradecimentos, sua contribuição foi muito importante para a realização deste trabalho, você é para mim um exemplo de persistência e dedicação.

Aos membros da banca examinadora pela disponibilidade em analisar este trabalho, pelos comentários e sugestões feitos com o intuito de contribuir para aprimoramento deste trabalho.

A meus pais, João Medeiros Furtado e Maria Lucilene Moraes da Costa. Eles são exemplos de simplicidade, dedicação ao trabalho e competência. Agradeço a vocês que mesmo com a distância física se fizeram presente com incentivos, apoio e em algumas vezes acreditando mais em mim que eu mesma.

A meus irmãos, Francisco de Assis Moraes Furtado, Heloísa Moraes Furtado, Alírio Tenório Furtado Neto, Helen Cristina Moraes Furtado e Lilian Cristina Moraes Furtado pela compreensão de minha ausência em momentos importantes da vida de vocês.

A minha avó Claudia e a minha avó Maria por sempre rezarem por mim.

Aos meus tios, principalmente ao tio Ronaldo, que foi muito importante para que eu conseguisse o meu primeiro emprego. Trabalhar durante toda a graduação não foi fácil, mas era necessário para manter-me na faculdade, o primeiro passo do caminho a ser trilhado para chegar ao doutorado. A você tio meu muito obrigado.

Agradeço aos meus amigos, que nos momentos em que precisei estavam disponíveis

a me ajudar. Sou muito feliz por saber que sempre temos com quem contar. Na vida nem sempre, é possível estarmos com todos que amamos, mas o importante é não estarmos sozinhos.

Agradeço também a todas as pessoas que passaram por minha vida, pois como disse Charles Chaplin: “Cada pessoa que passa em nossa vida passa sozinha e não nos deixa só porque deixa um pouco de si e leva um pouquinho de nós.”

Agradeço ao Instituto Nacional de Pesquisas Espaciais pela oportunidade em aprimorar a minha qualificação profissional, além dos recursos físicos necessários para o desenvolvimento deste trabalho. Ao CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelo apoio financeiro.

Enfim, agradeço a todos aqueles que, de alguma maneira, colaboraram e me ajudaram a vencer mais esta etapa da vida.

O meu...

Muito Obrigada!!!!

## RESUMO

A descrição da maioria dos fenômenos físicos por meio de equações diferenciais envolve erros, uma vez que um modelo físico-matemático é sempre uma aproximação da realidade. Para sistemas operacionais de previsão, uma estratégia para lidar com incertezas do erro de modelagem e das observações é adicionar alguma informação real do sistema físico ao modelo matemático. Esta informação adicional consiste de observações (valores medidos) sobre o fenômeno que se deseja modelar. No entanto, os dados observados devem ser inseridos com cuidado para evitar uma degradação no desempenho da previsão. Técnicas de assimilação de dados são ferramentas que combinam de modo eficaz observações com dados de modelos físico-matemáticos para a determinação do dado de análise (condição inicial), que é usado para executar o modelo de previsão. Esse processo é fundamental na prática operacional da previsão numérica do tempo e em circulação oceânica. Com o desenvolvimento do sistema de observação e a evolução dos modelos numéricos, a ordem de dados observados disponíveis para serem assimilados é de  $10^5$  a  $10^7$ , enquanto o número de pontos de grade do modelo é da ordem de  $10^6$  a  $10^8$ , isso representa um grande desafio no processo de assimilação de dados, isto é, *o estudo de técnicas que utilizam todos os dados disponíveis produzindo uma análise dentro do período disponível da prática operacional*. Deste modo, este trabalho apresenta uma técnica de assimilação de dados que utiliza redes neurais artificiais aplicada ao modelo de água rasa em duas dimensões e na equação da onda em uma dimensão. As observações utilizadas foram sintéticas. As redes neurais aplicadas foram a Perceptron de Múltiplas Camadas, utilizada para emular as técnicas de Filtro de Kalman (FK) e o método *Representante*. O método empregado apresentou grande eficiência computacional com a mesma qualidade das análises obtidas por meio do FK e o método *Representante*. No cenário atual de dados de observação em crescente expansão, bem como o aumento da resolução dos modelos de circulação atmosférica e oceânica, esta técnica pode ser uma alternativa viável para o problema de assimilação de dados.



# ARTIFICIAL NEURAL NETWORKING TO DATA ASSIMILATION AT OCEAN CIRCULATION MODEL

## ABSTRACT

The description of physical phenomena through differential equations carries errors, since the mathematical model is always an approximation of reality. For an operational prediction system, one strategy to deal with uncertainties from the modeling and observation errors is to add some information from the real dynamics into the mathematical model. This additional information consists of observations on the phenomenon. However, the observational data insertion should be done carefully, in order to avoid loss of performance of the prediction. Technical data assimilation are tools to combine data from physical-mathematics model with observational data to obtain a better forecast. This process is fundamental to the practice of operational numerical weather prediction and ocean circulation. With the development of the observing system and the development of numerical models, the order of observed data available to be assimilated is  $10^5$  to  $10^7$ , while the number of grid points of the model is around  $10^6$  to  $10^8$ , this represents a hard challenge in the process of data assimilation, that is, *the development of techniques that uses all available data to produce an analysis within the available period of operational practice*. Thus, this work presents a data assimilation technique that uses artificial neural network models applied in shallow water in two dimensions and the wave equation in one dimension. Synthetic Observations were used. The neural network was applied to Multi-Layer Perceptron, which is used to emulate the techniques of Kalman Filter (KF) and representer method. The method showed high computational efficiency with the same quality of the analyzes obtained by the FK and representer method. In the current scenario observation data are becoming increasingly widespread, requiring higher resolution models of atmospheric and oceanic circulation, this technique may be a viable alternative to the problem of data assimilation.



## LISTA DE FIGURAS

	<u>Pág.</u>
1.1 Ilustração da sensibilidade das condições iniciais do sistema de Lorenz. Para condições iniciais ligeiramente diferentes o modelo produz trajetórias distintas. . . . .	4
1.2 Série temporal da componente $x$ do sistema de Lorenz. Curva azul: referência de verdade; quadrados verdes: observações; curva vermelha: estimativa. . . . .	5
1.3 Gráfico com os erros de assimilação para o sistema de Lorenz. Coluna à direita Filtro de Kalman (FK), Filtro de Partículas (FP), Método Variacional (MV). Coluna à esquerda estimativa com a rede PMC. As observações foram assimiladas a cada 12 passos de tempo. . . . .	8
2.1 Evolução temporal da Equação 2.1 com suas projeções nos eixos: $x$ distância; $y$ : tempo; $z$ : amplitude do <i>sóliton</i> . . . . .	13
2.2 Evolução temporal da Equação 2.1. . . . .	14
2.3 Esquema dos pontos de grade do modelo (representado pelo círculos pretos), as observações representadas pelo quadrado verde, que são distribuídas no ponto de grade. . . . .	16
2.4 Teoria de água rasa. . . . .	18
2.5 Canal periódico com paredes rígidas no plano $xy$ . . . . .	19
2.6 Esquema do método de Grade $C$ de Arakawa para diferenças espaciais. . . . .	21
2.7 Esquema do método de grade $C$ de Arakawa para diferenças temporais. . . . .	21
2.8 Evolução temporal da variável $q$ nas projeções $(x, y)$ . . . . .	22
3.1 Ilustração dos três tipos básicos de estimação: (a) filtragem, (b) <i>smoothing</i> e (c) previsão. . . . .	25
3.2 Ilustração da Lei de Hook. . . . .	27
3.3 Diagrama esquemático do Filtro de Kalman Linear. . . . .	42
4.1 Representação gráfica de um funcional de penalidade quadrático não negativo. . . . .	47
4.2 Esboço para a implementação do algoritmo representante com o cálculo direto do coeficiente $\hat{\beta}$ . . . . .	63
4.3 Esboço para a implementação do algoritmo do representante com o cálculo indireto do coeficiente $\hat{\beta}$ . . . . .	64

4.4	Interface gráfica do software IOM. . . . .	66
5.1	Representação de um neurônio biológico. . . . .	72
5.2	Representação de um neurônio artificial. . . . .	74
5.3	Exemplos de funções de ativação mais utilizadas: (a) tangente hiperbólica; (b) exponencial; (c) função degrau. . . . .	75
5.4	Ilustração da Rede Perceptron de Múltiplas Camadas . . . . .	80
6.1	Ilustração da matriz de dinâmica usada no Filtro de Kalman para o modelo 1D. . . . .	85
6.2	Assimilação de dados realizada a cada 10 passos de tempo com o Filtro de Kalman. Amplitude da onda no tempo $t = 200$ . Experimento 1: condição inicial usada um sólito KdV. . . . .	86
6.3	Assimilação de dados a cada 10 passos de tempo com o Filtro de Kalman. Amplitude da onda em todos os tempos. Experimento 1: condição inicial usada um sólito KdV. . . . .	86
6.4	Assimilação de dados realizada a cada 20 passos de tempo com o Filtro de Kalman. Amplitude da onda no tempo $t = 500$ . Experimento 2: função seno como condição inicial. . . . .	87
6.5	Assimilação de dados realizada a cada 20 passos de tempo com o Filtro de Kalman. Amplitude da onda em todos os tempos nas coordenadas $(x, y, z)$ . Experimento 2: função seno como condição inicial. . . . .	88
6.6	Curvas do erro de treinamento para os dados de treinamento para o modelo 1D. . . . .	88
6.7	Arquitetura da RNA empregada para o modelo 1D. . . . .	89
6.8	Conjunto de dados de treinamento para o experimento 1 do modelo em uma dimensão. . . . .	90
6.9	Conjunto de dados de treinamento para o experimento 2 do modelo em uma dimensão. . . . .	90
6.10	Assimilação de dados realizada a cada 10 passos de tempo com RNA. Amplitude da onda no tempo $t = 200$ . Experimento 1: condição inicial usada um sólito KdV. . . . .	91
6.11	Assimilação de dados realizada a cada 10 passos de tempo com RNA. Amplitude da onda em todos os tempos. Experimento 1: condição inicial usada um sólito KdV. . . . .	92
6.12	Assimilação de dados realizada a cada 20 passos de tempo com RNA. Amplitude da onda no tempo $t = 200$ . . . . .	92

6.13	Assimilação de dados realizada a cada 20 passos de tempo com RNA. Amplitude da onda em todos os tempos. . . . .	93
6.14	(lado esquerdo) experimento <b>A</b> ; (lado direito) experimento <b>B</b> . Os quadrados amarelos representam as observações no ponto de grade para o modelo 2D. . . . .	95
6.15	Conjunto de dados para o treinamento da rede para o experimento <b>A</b> do modelo 2D. . . . .	97
6.16	Conjunto de dados para o treinamento da rede para o experimento <b>B</b> do modelo 2D. . . . .	98
6.17	Arquitetura das redes para as variáveis $u$ , $v$ e $q$ . Os sobrescritos $m$ , $o$ e $a$ significam modelo, observação e análise, respectivamente. . . . .	98
6.18	Evolução temporal dos pontos $u(7,7)$ , $v(7,7)$ e $q(7,7)$ experimento <b>A</b> . . .	101
6.19	Evolução temporal dos pontos $u(7,7)$ , $v(7,7)$ e $q(7,7)$ experimento <b>B</b> . . .	102
6.20	Projeção da variável $q$ em três dimensões no passo de tempo 60. . . . .	103
6.21	Projeção da variável $q$ em três dimensões no passo de tempo 100. . . . .	103
6.22	(lado esquerdo) Assimilação com o Representante; (lado direito) Assimilação com a RNA. Curva azul: <i>verdade</i> ; curva vermelha: estimado com o representante; curva verde: estimado com a RNA. . . . .	105
6.23	Comparação entre o método Representer e a RNA. Curva azul: <i>verdade</i> ; curva vermelha: estimado com o representer; curva verde: estimado com a RNA. . . . .	106
6.24	Observações interpoladas no ponto de grade do modelo para $u$ , $v$ e $q$ ; ponto verde: observação no tempo 10; círculo vermelho: observação no tempo 11. . . . .	109
6.25	Janelas de integração do modelo de água rasa 2D. Os pontos verdes indicam o ponto da observação para a variável $u$ e $v$ . . . . .	110
6.26	Janelas de integração do modelo de água rasa 2D. Os pontos verdes indicam o ponto da observação para a variável $q$ . . . . .	111
6.27	Generalização da RNA; <i>janela 21</i> . . . . .	112
6.28	Variável $u$ (vento zonal em $m/s$ ); estimativa obtida com a RNA (curva verde); <i>verdade</i> (curva azul). . . . .	113
6.29	Variável $u$ (vento zonal em $m/s$ ); estimativa obtida com o método do Representante (curva vermelha); <i>verdade</i> (curva azul). . . . .	114
6.30	Gráficos das diferenças para a variável $u$ e $t = 10$ . lado esquerdo: rede x modelo ( <i>verdade</i> ); lado direito: representante x modelo ( <i>verdade</i> ). . . . .	114
6.31	Variável $v$ ( $m/s$ ) no tempo $t = 10$ ; lado direito: RNA x <i>verdade</i> ; lado esquerdo: Representante x <i>verdade</i> . . . . .	115

6.32	Gráficos das diferenças para a variável $v$ e $t = 10$ . lado esquerdo: rede x modelo ( <i>verdade</i> ); lado direito: representante x modelo ( <i>verdade</i> ). . . . .	115
6.33	Variável $q$ ( $m$ ) no tempo $t = 10$ ; estimativa obtida com a RNA; curva azul: modelo ( <i>verdade</i> ); curva verde: estimado. . . . .	116
6.34	Variável $q$ ( $m$ ) no tempo $t = 10$ ; estimativa obtida com o representante; curva azul: modelo ( <i>verdade</i> ); curva vermelha: estimado . . . . .	117
6.35	Gráficos das diferenças para a variável $q$ e $t = 10$ . lado esquerdo: rede x modelo ( <i>verdade</i> ); lado direito: representante x modelo ( <i>verdade</i> ). . . . .	117
6.36	Variável $u$ (vento zonal em $m/s$ ) no tempo $t = 20$ ; lado direito: RNA x verdade; lado esquerdo: Representante x verdade. . . . .	118
6.37	Gráficos das diferenças para a variável $u$ em $t = 20$ ; lado esquerdo: rede x modelo ( <i>verdade</i> ); lado direito: representante x modelo ( <i>verdade</i> ). . . . .	118
6.38	Variável $v$ no tempo $t = 20$ ; estimativa obtida com a RNA; curva azul: modelo ( <i>verdade</i> ); curva verde: estimado. . . . .	119
6.39	Variável $v$ no tempo $t = 20$ ; estimativa obtida com o método do representante; curva azul: modelo ( <i>verdade</i> ); curva vermelha: estimado. . . . .	120
6.40	Gráficos das diferenças para a variável $v$ em $t = 20$ ; estimativa obtida com a RNA; curva azul: verdade; curva verde: estimado. . . . .	120
6.41	Variável $q$ no tempo $t = 20$ ; estimativa obtida com a RNA; curva azul: modelo ( <i>verdade</i> ); curva verde: estimado. . . . .	121
6.42	Variável $q$ no tempo $t = 20$ ; estimativa obtida com o método do representante; curva azul: modelo ( <i>verdade</i> ); curva vermelha: estimado. . . . .	122
6.43	Gráficos das diferenças para a variável $q$ em $t = 20$ ; lado esquerdo: rede x modelo ( <i>verdade</i> ); lado direito: representante x modelo ( <i>verdade</i> ). . . . .	122

## LISTA DE TABELAS

	<u>Pág.</u>
2.1	Parâmetros usados na integração da equação da onda. . . . . 12
2.2	parâmetros de discretização do modelo 2D. . . . . 21
4.1	Algoritmo <i>representante</i> para o modelo de onda linear. . . . . 57
4.2	Algoritmo <i>representante</i> para o modelo de água rasa linear. . . . . 61
6.1	Resultados do erro de assimilação para FK e o PMC para o modelo 1D. . . 91
6.2	Custo computacional para FK e o PMC para o modelo 1D. . . . . 93
6.3	Parâmetros da rede PMC para o experimento <b>A</b> do modelo 2D. . . . . 99
6.4	Parâmetros da rede PMC para o experimento <b>B</b> do modelo 2D. . . . . 99
6.5	Resultados do erro de assimilação para FK e o PMC para o experimento <b>A</b> do modelo 2D. . . . . 100
6.6	Resultados do erro de assimilação para FK e o PMC para o experimento <b>B</b> do modelo 2D. . . . . 100
6.7	Custo computacional para FK e o PMC. . . . . 100
6.8	Valores das coordenadas para a observação 3. . . . . 108
6.9	Número de dados que formam os conjuntos de treinamento. . . . . 110
6.10	Conjuntos de treinamento. . . . . 111
6.11	Parâmetros da rede PMC para a estimação da condição inicial. . . . . 112
6.12	Parâmetros da rede PMC para a estimação da forçante. . . . . 112
6.13	Parâmetros da rede PMC para a estimação da condição de contorno. . . 112
7.1	Tempo de CPU para diferentes técnicas de assimilação com diferentes modelos de previsão, para vários experimentos numéricos. . . . . 124



## LISTA DE ABREVIATURAS E SIGLAS

c.i	–	condição inicial
c.c	–	condição de contorno
IOM	–	Inverse Ocean Modeling
I/O	–	Input/Output
EDP	–	Equações Diferenciais Parciais
EDO	–	Equações Diferenciais Ordinárias
FPGA	–	Field programmable gate array
EMQ	–	Erro Médio Quadrático
RMS	–	Root Mean Square
MQR	–	Mínimos Quadrados Recursivos
MQP	–	Mínimos Quadrados Ponderados
SVD	–	Singular Value Decomposition
BLUE	–	Best linear unbiased estimation
FK	–	Filtro de Kalman
EL	–	Euler-Lagrange
PMC	–	Perceptron de Múltiplas Camadas
RNA	–	Redes Neurais Artificiais
FKE	–	Filtro de Kalman Estendido
fdp	–	Função densidade de probabilidade
MPCA	–	Multi-Particle Collision Algortihm
FTCS	–	Forward-Time Central-Space



## LISTA DE SÍMBOLOS

$a$	–	Constante responsável pela inclinação da tangente hiperbólica
$E(.)$	–	Valor médio esperado
$\mathbf{I}$	–	Matriz identidade
$J(.)$	–	Função custo (funcional de penalidade)
$\beta$	–	Coefficiente do <i>representante</i>
$\delta$	–	Função delta de Dirac; operador variacional; função delta de Kronecker; gradiente local
$\partial$	–	Operador diferencial parcial
$\lambda$	–	Multiplicador de Lagrange
$w$	–	Matriz de covariância do erro de observação no método representer 4
$W_f$	–	Matriz de covariância do erro de modelagem no método representer 4
$W_i$	–	Matriz de covariância do erro da condição inicial
$C_d$	–	Coefficiente de arrasto
$\mathbf{F}$	–	Matriz que representa o modelo matemático (Sistema dinâmico)
$f$	–	Parâmetro de coriolis
$g$	–	Constante gravitacional
$l$	–	Índice referente a camada oculta das RNA
$L$	–	Índice referente a camada de saída das RNA
$o$	–	Vetor obtido na saída da RNA
$y_k$	–	Sinal de saída do neurônio
$\rho_a$	–	Densidade do ar
$\rho_w$	–	Densidade da água
$u_a$	–	Vento zonal
$\theta$	–	Vetor de pesos da RNA
$\mathbf{H}$	–	Matriz que representa o sistema de observação
$\mathbf{P}^a$	–	Matriz de matriz de covariância do erro de análise
$\mathbf{P}^p$	–	Matriz de matriz de covariância do erro de previsão
$\mathbf{Q}$	–	Matriz de covariância do erro de modelagem
$\mathbf{R}$	–	Matriz de covariância do erro de observação
$\mathbf{W}$	–	Matriz de ponderação
$\nabla$	–	gradiente



## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO</b> . . . . .	<b>1</b>
<b>2 MODELOS PARA TESTES DE MÉTODOS DE ASSIMILAÇÃO DE DADOS</b> . . . . .	<b>11</b>
2.1 Equação da onda linear . . . . .	11
2.1.1 Dados . . . . .	15
2.1.2 Descrição dos resíduos para a onda linear . . . . .	16
2.2 Modelo de água rasa linear 2D . . . . .	17
2.2.1 Discretização para o modelo de água rasa 2D . . . . .	19
<b>3 ESTIMAÇÃO POR MÍNIMOS QUADRADOS E FILTRO DE KALMAN</b> . . . . .	<b>23</b>
3.1 Mínimos quadrados . . . . .	28
3.2 Mínimos quadrados ponderados . . . . .	32
3.2.1 A escolha dos pesos . . . . .	33
3.3 Mínimos quadrados recursivos . . . . .	36
3.4 Filtro de Kalman . . . . .	37
<b>4 MÉTODO VARIACIONAL: TÉCNICA DO REPRESENTANTE</b> <b>45</b>	
4.1 Método variacional: equação da onda 1D . . . . .	45
4.1.1 Funcional de Penalidade . . . . .	45
4.1.2 Equação de Euler-Lagrange . . . . .	48
4.1.3 Método <i>Representante</i> . . . . .	52
4.1.4 Determinando o coeficiente do <i>representante</i> . . . . .	53
4.1.5 Determinando uma solução explícita para a solução ótima . . . . .	55
4.2 Método variacional: equação de água rasa 2D . . . . .	57
4.2.1 As equações de Euler-Lagrange . . . . .	58
4.2.2 Equações adjunta e <i>representante</i> . . . . .	59
4.3 Implementação do método representante . . . . .	61
4.3.1 Pacote de software IOM . . . . .	64
<b>5 REDES NEURAS ARTIFICIAIS</b> . . . . .	<b>71</b>

5.1	Neurônio Biológico . . . . .	71
5.2	Neurônio Artificial . . . . .	72
5.2.1	Funções de ativação . . . . .	74
5.3	Funcionamento das Redes Neurais . . . . .	75
5.3.1	Regras de Aprendizagem . . . . .	77
5.4	Algoritmo “ <i>Backpropagation</i> ” ou de Retropagação . . . . .	78
5.5	Perceptron de Múltiplas Camadas . . . . .	79
5.6	Rede neural artificial auto-configurável . . . . .	80
<b>6</b>	<b>ASSIMILAÇÃO DE DADOS COM REDES NEURAIIS . . . . .</b>	<b>83</b>
6.1	Redes neurais emulando o Filtro de Kalman . . . . .	83
6.1.1	Equação da onda 1D . . . . .	84
6.1.2	Modelo de água rasa linear 2D . . . . .	93
6.2	Redes neurais emulando o Método do Representante . . . . .	104
6.2.1	Equação da onda 1D . . . . .	104
6.2.2	Modelo de água rasa 2D . . . . .	106
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>123</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS . . . . .</b>	<b>127</b>
	<b>APÊNDICE A - DERIVANDO A EQUAÇÃO DE EULER LA-GRANGE . . . . .</b>	<b>.139</b>
	<b>APÊNDICE B - Estrutura de chamada dos programas do <i>software</i> IOM . . . . .</b>	<b>.143</b>
	<b>ANEXO A - Publicações . . . . .</b>	<b>.145</b>

## 1 INTRODUÇÃO

Muitos problemas em geociências requerem a estimação do estado de sistemas ao longo do tempo utilizando medidas ruidosas realizadas sobre o sistema em estudo. Esses sistemas podem ser descritos por meio de modelos matemáticos, isto é, por um conjunto de equações diferenciais (ISMAIL-ZADEH; TACKLEY, 2010). No entanto, descrever fenômenos físicos a partir de equações diferenciais, seja parciais ou ordinárias, isto é, por meio da modelagem matemática, é uma atividade passível de erro, uma vez que o modelo físico e matemático é sempre uma aproximação da realidade.

Em meteorologia e oceanografia, assimilação de dados é um meio de estimação do estado da atmosfera ou do oceano pela combinação de dados observacionais com uma estimativa *a priori* proveniente do modelo dinâmico (GRIFFITH, 1997). Esta estimativa do estado da atmosfera ou do oceano é frequentemente denominada de análise. No caso da meteorologia, a principal aplicação da assimilação de dados é na Previsão Numérica do Tempo, na qual ela é usada para obter uma boa estimativa do estado atual da atmosfera para iniciar um modelo de previsão.

Para a oceanografia, modelar o oceano consiste em resolver as equações do modelo tão exatas quanto possível comparando os resultados com as observações. Contudo, como o oceano é um fluido turbulento, pequenas mudanças nas condições iniciais podem originar impactos significativos sobre a evolução subsequente do fluido. Por outro lado, mesmo que fosse possível resolver completamente a equação diferencial parcial do movimento do fluido, a previsão do estado do oceano será limitada pela acurácia dos dados das condições iniciais e condições de contorno.

O modelo numérico, seja oceânico ou atmosférico, requer discretização das equações diferenciais, com conseqüente redução dos graus de liberdade das equações originais. Ocorre também a parametrização de alguns processos físicos, como turbulência, radiação, entre outros, gerando fonte significativa de erro. Estes fatores impossibilitam simular o fluxo de fluido com exatidão (ZARON, 2009; ROBINSON; LERMUSIAUX, 2000).

De acordo com Robinson e Lermusiaux (2000), o problema fundamental em oceanografia pode ser definido como: dado o estado passado do oceano, o estado do oceano no tempo posterior é estimado pela dinâmica do modelo. A lei básica e os princípios da física, química e biologia do oceano envolvem a variável de estado

do modelo avançando no tempo. Assim, do ponto de vista prático, estimar o estado presente e futuro das variáveis de estado do oceano para aplicações está intimamente acoplado ao fundamento da ciência do oceano.

Usualmente, as variáveis de estado são: componentes de velocidade do vento, pressão, densidade, temperatura, salinidade e elevação da superfície. Exemplos de variáveis de estados biológicas e químicas são: campos de concentração de nutrientes, plâncton e etc. Devido a complexidade dos sistemas biogeoquímicos marinhos, o número de variáveis possíveis é extremamente grande e a limitação para um subconjunto finito de variáveis de estado “*críticas*” é um problema importante de pesquisa contemporânea (ROBINSON; LERMUSIAUX, 2000).

A oceanografia operacional é uma área de crescente relevância para uma melhor caracterização do oceano. Os oceanos exercem uma grande influência no ambiente global, tornando a sua investigação fundamental para compreender os fenômenos que neles ocorrem. O seu conhecimento exige, entre outros, estudos das correntes, temperaturas, salinidades e elevação da superfície do mar, constituindo objetivos da oceanografia operacional, com aplicações no âmbito científico, ambiental e econômico.

Um fator importante para a compreensão das ferramentas de assimilação de dados é o conceito de erros, isto é, a estimação do erro associado aos dados observados e ao erro de modelagem. As observações contêm erros oriundos de diversas fontes, por exemplo, erro de instrumentação, amostragem e representatividade. O modelo dinâmico, por sua vez, é imperfeito, com erros provenientes da aproximação física (química ou biológica) que governam a evolução explícita das variáveis de estado, a representação de processos físicos que parametrizam a interação entre as variáveis e a discretização do modelo. Portanto, é necessário lidar com incertezas e limitações inerentes ao modelo físico-matemático e com as incertezas associadas aos dados observados. Estes fatores estimulam a utilização das ferramentas de assimilação de dados.

Os componentes básicos para sistemas operacionais de previsão são: a rede de dados de observação, o modelo numérico e o método de assimilação de dados. As técnicas de assimilação de dados são uma alternativa para combinar de modo eficaz os dados observados com os dados do modelo físico matemático. Essas técnicas podem ser baseadas em teoria da estimação (como: Filtro de Kalman (FK) (KALMAN, 1960);

Filtro de Partículas (FP) (GORDON et al., 1993; CHORIN; KRAUSE, 2004)); no cálculo variacional (TALAGRAND; COURTIER, 1987)(como: método do *Representante* (BENNETT, 2002)), ou ainda, baseadas em Inteligência Artificial (IA), com o uso de Redes Neurais Artificiais (RNA)(NOWOSAD, 2001; HARTER, 2004; CINTRA, 2010).

As técnicas de assimilação de dados tornaram-se uma ferramenta de extrema importância para modelos numéricos, por exemplo, na previsão numérica do tempo (KALNAY, 2003; DALEY, 1993), em circulação oceânica (BENNETT, 2002), e mais recentemente têm sido utilizada para a previsão de clima espacial (HÄRTER et al., 2008; SCHERLISS et al., 2004; PETRY et al., 2012).

Na previsão numérica do tempo, é importante ressaltar que, mesmo considerando, por hipótese, modelos determinísticos e observações perfeitas, a natureza caótica da atmosfera limitaria o tempo de previsibilidade (KALNAY, 2003). Esse fenômeno foi primeiramente observado no trabalho de Lorenz (1963), em que a solução de equações semelhantes às que governam os movimentos atmosféricos apresentaram dependência sensível com relação a variação da condição inicial, conforme ilustrado na Figura 1.1.

O processo de assimilação de dados pode ser visto como um problema de determinação de condição inicial (c. i.) e definido como: “*a ciência de se ter uma adequada combinação de dados de um modelo físico-matemático com dados de observação para se determinar o dado de análise (ou c. i.)*” (CAMPOS VELHO et al., 2007). Portanto, quanto mais próximo da realidade for a estimativa da condição inicial, melhor será a qualidade da previsão.

A Figura 1.2 ilustra a aplicação da técnica de assimilação de dados. Observa-se que, à medida que são inseridas as observações no sistema, a dinâmica da estimativa (curva vermelha) aproxima-se da referência de verdade (curva azul), uma vez que é interrompida a inserção das observações, ocorre o desacoplamento das dinâmicas. Assim, as observações, quando combinadas de modo eficaz com dados do modelo físico-matemático, servem para aproximar com a precisão desejada a estimativa da condição inicial real do sistema. Neste estudo de caso, as observações assimiladas são observações sintéticas, determinadas por meio da integração do sistema de Lorenz com um ruído adicionado.

A evolução histórica dos métodos de assimilação de dados passa pelo Ajuste de

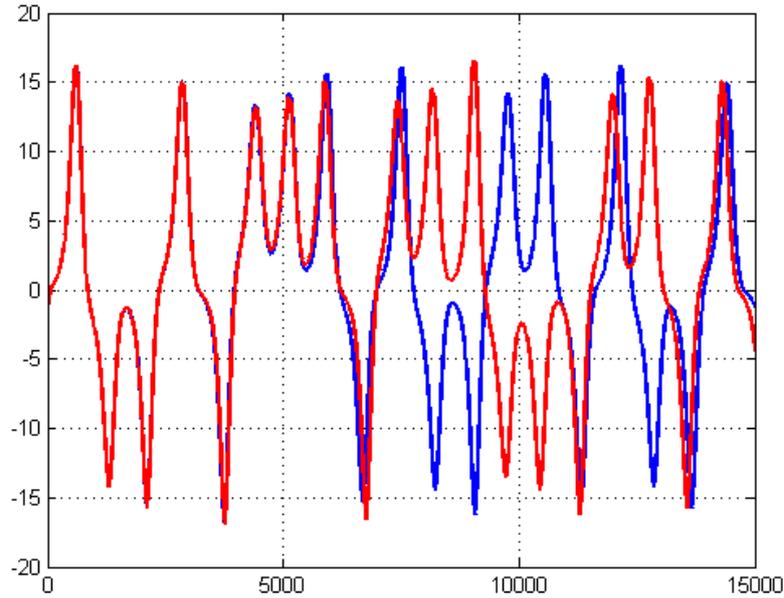


Figura 1.1 - Ilustração da sensibilidade das condições iniciais do sistema de Lorenz. Para condições iniciais ligeiramente diferentes o modelo produz trajetórias distintas.

Fonte: Adaptado de Furtado (2008)

Funções, Correções Sucessivas, Análise de Correções, Interpolação Ótima, Métodos Variacionais, Filtro de Kalman e Técnicas de Monte Carlo, (DALEY, 1993; KALNAY, 2003; GRIFFITH, 1997). Todos esses métodos implicam em diferentes técnicas de se combinar uma previsão de “background” (estimativa *a priori* do modelo de previsão) com observações, resultando no dado de análise ou condição inicial (c.i.), que será usado para iniciar os modelos numéricos para o ciclo seguinte de previsão.

Matematicamente, assimilação de dados é um processo que compreende dois passos:

(1) Passo de Previsão:

$$\mathbf{x}_k^p = \mathbf{F}(\mathbf{x}_{k-1}^a) \quad (1.1)$$

(2) Passo de Análise, que pode ser feito de várias formas, entre as quais:

(2.1) teoria da estimação

$$\mathbf{x}_k^a = \mathbf{x}_k^p + \mathbf{K}(\mathbf{y}_k - \mathbf{H}[\mathbf{x}_k^p]) \quad (1.2)$$

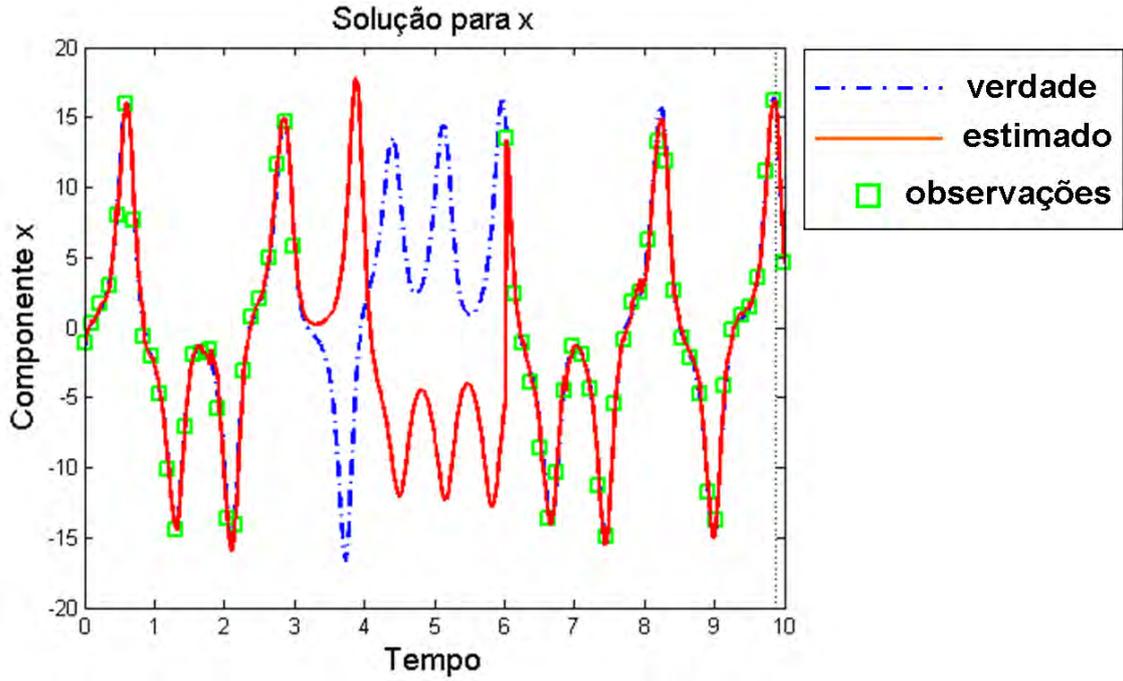


Figura 1.2 - Série temporal da componente  $x$  do sistema de Lorenz. Curva azul: referência de verdade; quadrados verdes: observações; curva vermelha: estimativa.  
 Fonte: Adaptado de Furtado (2008)

(2.2) cálculo variacional

$$\mathbf{x}_k^a = \text{MinJ}(\mathbf{x}^p) \quad (1.3)$$

$$\mathbf{x}_k^a = \mathbf{x}_k^p + \sum_{m=1}^M \beta_m r_m(x, k) \quad (1.4)$$

(2.3) redes neurais artificiais

$$\mathbf{x}_k^a = \mathbf{F}_{RNA_w^*}(\mathbf{y}_k, \mathbf{x}_k^p) \quad (1.5)$$

em que  $\mathbf{x}_k^p$  é o vetor de variáveis de estado prevista do modelo, o subscrito  $k$  indica o tempo discreto,  $\mathbf{x}_k^a$  representa o dado de análise (c.i.),  $\mathbf{y}_k$  o vetor de observações. Na Equação 1.1,  $\mathbf{F}$  representa o modelo numérico de previsão. Na Equação 1.2, a matriz  $\mathbf{H}$  é o operador linear que representa o sistema de observação,  $\mathbf{K}$  é a matriz de ganho de Kalman. Na Equação 1.4,  $x$  é a coordenada espacial,  $\beta$  é o coeficiente do representante e  $r$  é o valor do representante. No Capítulo 4, descreve-se como se determinam esses valores. Na Equação 1.5,  $\mathbf{F}_{RNA_w^*}$  é uma função que mapeia os

dados de entrada  $(\mathbf{y}_k, \mathbf{x}_k^p)$  com os dados da saída desejada  $\mathbf{x}_k^a$  para a determinação do  $w^*$  ótimo. No [Capítulo 5](#), descreve-se como esse processo é realizado.

A inovação tecnológica permitiu o surgimento de supercomputadores, a evolução dos modelos numéricos de previsão e da rede de observação. Esses fatores vem proporcionando maior confiabilidade na previsão numérica operacional. O aumento na quantidade de satélites lançados disponibiliza uma grande quantidade de informações meteorológicas e oceanográficas. O aperfeiçoamento da resolução de grade dos modelos numéricos de previsão também é fato. Isso significa que a ordem de grandeza de dados observacionais disponíveis é da ordem de  $10^5$  a  $10^7$ , enquanto o número de variáveis do modelo é da ordem de  $10^6$  a  $10^8$  ([KALNAY, 2003](#)).

Pode-se dizer que o desafio atual da assimilação de dados consiste em: conceber formas para utilizar os dados observados disponíveis em modelos com crescente número de pontos de grade e produzir uma análise dentro do tempo disponível na prática operacional. De acordo com os trabalhos de [Nowosad et al. \(2000\)](#), [Harter \(2004\)](#) e [Cintra \(2010\)](#), a abordagem para assimilação de dados por meio de Redes Neurais Artificiais pode ser uma solução.

O uso de redes neurais para assimilação de dados é recente. Esta metodologia foi sugerida como uma possível técnica por [Hsieh e Tang \(1998\)](#). No entanto, a primeira implementação foi feita por [Nowosad et al. \(2000\)](#), apresentando o uso de um Filtro de Kalman Estendido Adaptativo e Redes Neurais Perceptrons de Múltiplas Camadas. Essas técnicas foram aplicadas para três modelos de sistemas dinâmicos não lineares: sistema de tempo discreto caótico de Hénon, sistema de Lorenz em regime caótico e modelo computacional DYNAMO<sup>1</sup>. A RNA utilizada para fazer assimilação de dados foi treinada para “emular” o filtro de Kalman, com o objetivo de reduzir o custo computacional. [Nowosad et al. \(2000\)](#) concluiu que para sistemas de alta dimensão as redes treinadas podem ser computacionalmente mais rápidas que os filtro de Kalman. O algoritmo foi paralelizado em [Vijaykumar et al. \(2002\)](#), havendo uma redução significativa do tempo de processamento durante o treinamento do Perceptron de Multiplas Camadas (PMC).

Mais tarde, o trabalho de [Harter \(2004\)](#) apresentou o desempenho das RNA Funções de Base Radial (FBR), Elman (RN-E), Jordan (RN-J) e PMC, avaliando a eficiência destas em “emular” o Filtro de Kalman. As aplicações foram feitas nos sistemas não

---

<sup>1</sup>Simulador meteorológico baseado na equação de Água Rasa 1D ([LYNCH, 1989](#)).

lineares caóticos de Lorenz e no modelo DYNAMO, sendo que o PMC também foi aplicado ao modelo de interação não linear entre as ondas de Langmuir, Whistler e Alfvén (Modelo de 3 ondas)<sup>2</sup>. Esse foi um dos resultados inéditos do trabalho de Harter (2004): aplicação de assimilação de dados com RNA em clima espacial. Ressalta-se que as RNAs desenvolvidas por Harter (2004) foram implementadas segundo uma técnica conhecida como validação cruzada (*cross-validation*), o que permitiu que se conhecesse a superfície de erros de treinamento e validação, proporcionando a obtenção do melhor conjunto de pesos para o problema investigado.

Salienta-se também que houve uma evolução significativa no desenvolvimento do algoritmo de assimilação de dados com redes neurais artificiais: ao invés de se usar uma *rede global*, que se baseia em uma análise sobre todo o domínio (NOWOSAD, 2001), foi desenvolvida uma estratégia de *rede local*, em que a análise é gerada em cada ponto de grade (HARTER; CAMPOS VELHO, 2008). O ganho com a diminuição do espaço de busca foi um resultado importante da pesquisa de Harter (2004), pois em problemas de minimização de gradiente em aplicações de grande dimensão, diminuir o espaço de busca pode ser o fator a tornar a aplicação operacionalmente viável.

Dando continuidade a essas investigações, a pesquisa de mestrado desenvolvida por Furtado (2008) avaliou o desempenho de uma rede neural PMC treinada para emular: o Filtro de Kalman (FK), o Filtro de Partículas (FP) e o Método Variacional (MV), técnicas essas aplicadas ao sistema de Lorenz em regime caótico. Na Figura 1.3, apresenta-se o gráfico de erros para essas metodologias, em que o erro é a diferença absoluta entre a estimativa (trajetória do modelo obtida pela técnica de assimilação) e a referência de verdade (integração do modelo no tempo, ou seja, considerou-se que o modelo é perfeito). Na coluna à esquerda, tem-se a estimativa com o FK, FP e o MV, e à direita o erro para a rede PMC “*emulando*” essas técnicas, respectivamente.

De acordo com o gráfico, nota-se que a técnica que obteve a melhor estimativa foi o MV. Na coluna à direita, pode-se ver que a rede PMC foi compatível com as demais metodologias abordadas. É importante ressaltar que o FP é um método que viabiliza fazer assimilação dos dados observados a cada 500 passos de tempo, e a rede neural também é capaz de assimilar os dados com essa mesma frequência de inserção de observações (FURTADO et al., 2008), sendo que o FK e o MV falham. No trabalho de Furtado et al. (2011), compara-se o método de RNA com a formulação variacional. No entanto, a abordagem para assimilação de dados com redes neurais

---

<sup>2</sup>Aplicação em assimilação de Dados em Clima Espacial, veja (HARTER, 2004) pag. 93

artificiais possui vantagens no seguinte aspecto: o algoritmo é altamente paralelizável e permite implementações em hardware.

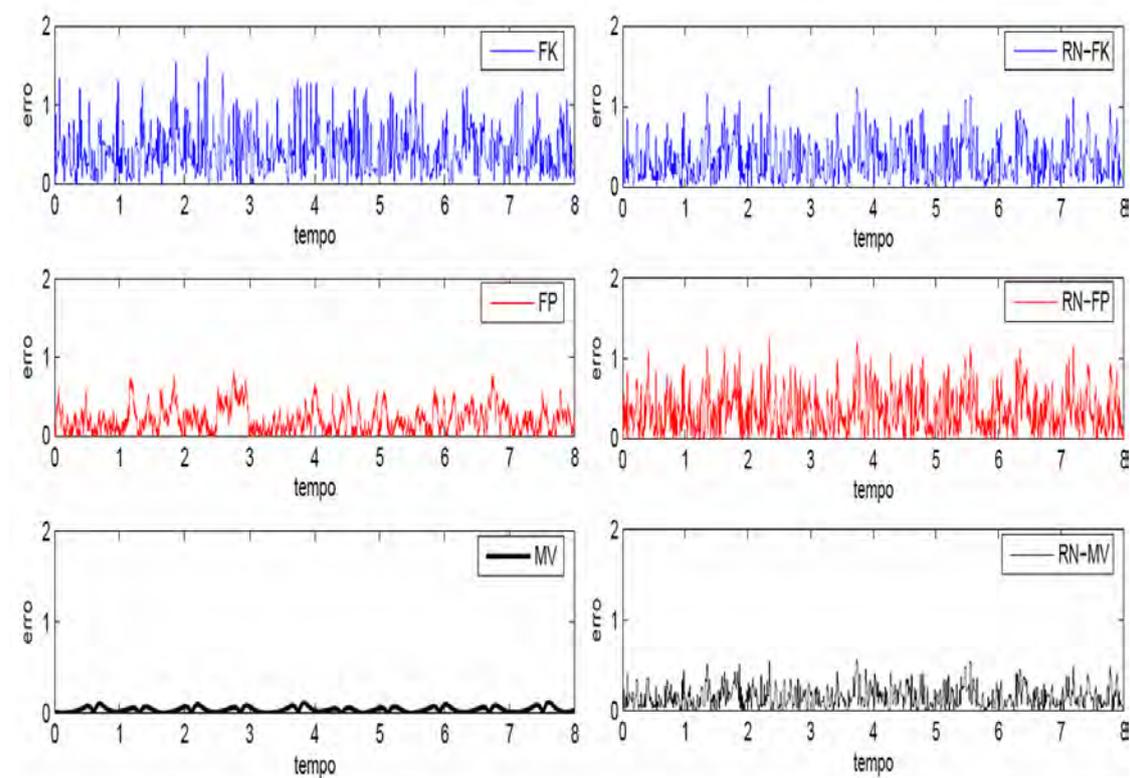


Figura 1.3 - Gráfico com os erros de assimilação para o sistema de Lorenz. Coluna à direita Filtro de Kalman (FK), Filtro de Partículas (FP), Método Variacional (MV). Coluna à esquerda estimativa com a rede PMC. As observações foram assimiladas a cada 12 passos de tempo.  
Fonte: Adaptado de Furtado (2008)

O último trabalho desenvolvido por esse grupo de pesquisa intitula-se “Assimilação de dados com redes neurais artificiais em um modelo de circulação geral da atmosfera.” (CINTRA, 2010), que fez uso do modelo SPEEDY (*Simplified Parameterizations, primitivE-Equation DYnamics*). Este é um modelo de circulação geral da atmosfera com grade em coordenadas tridimensionais (3D), que possui características semelhantes aos modelos operacionais de previsão do tempo (MOLTENI, 2003). A técnica de redes neurais artificiais foi utilizada para *emular* o método do filtro de Kalman por conjunto transformado e Localizado (*Local Ensemble Transform Kalman Filter* - LETKF). Esse método é uma aproximação do Filtro de Kalman

onde um conjunto de previsões de curto prazo são usadas para estimar as covariâncias do erro do modelo de previsão. Maiores informações encontram-se em (CINTRA, 2010).

Assim, o objetivo deste trabalho é continuar a investigação da metodologia de redes neurais artificiais no contexto de assimilação de dados com ênfase na:

- (a) aplicação no contexto de circulação oceânica;
- (b) utilização das redes neurais para emular o Filtro de Kalman e o método do Representante (uma variante da técnica Variacional).

O método do representante é uma técnica de assimilação de dados desenvolvida por Bennett e colaboradores para modelos de circulação oceânica. Uma compilação de desenvolvimento e aplicações da técnica podem ser consultadas nos livros Bennett (1992), Bennett (2002). As aplicações das técnicas de assimilação são testadas nos mesmos modelos usados em teste preliminares pelo método do representante:

- (a) equação da onda de primeira ordem 1D (BENNETT; BAUGH, 1992; BENNETT, 2002),
- (b) modelo de água rasa 2D (BENNETT, 2002).

Esta tese tem como principal contribuição apresentar a metodologia de Redes Neurais Artificiais (RNA) como uma técnica de assimilação de dados para modelos de circulação oceânica. A RNA emula o método do representante. Essa ferramenta obtém ganho computacional com relação as demais metodologias de assimilação de dados aqui apresentadas.

Assim, o texto desta tese está estruturado da seguinte forma: no **Capítulo 2** apresentam-se os modelos físico-matemáticos, que serão usados para validar as metodologias aqui abordadas. No **Capítulo 3**, deriva-se o método de Filtro de Kalman a partir do métodos de mínimos quadrados, que se enquadram na categoria de filtragem. No **Capítulo 4**, descreve-se o método do Representante, uma abordagem baseada no método variacional. No **Capítulo 5**, apresenta-se a técnica de assimilação de dados, que é baseada em inteligência artificial, o principal foco deste trabalho. Utiliza-se redes neurais artificiais por meio de um Perceptron de Múltiplas Camadas.

O [Capítulo 6](#) é reservado para os resultados. Por fim, no [Capítulo 7](#), apresentam-se as conclusões e sugestões para trabalhos futuros.

## 2 MODELOS PARA TESTES DE MÉTODOS DE ASSIMILAÇÃO DE DADOS

As equações matemáticas que descrevem o movimento do oceano são baseadas na Lei de Newton e nas Leis da Termodinâmica. Uma representação realista do oceano origina equações que são muito complicadas para serem resolvidas por métodos matemáticos usuais. Como uma aproximação prática para o problema, criam-se aproximações para estas equações que possam ser resolvidas numericamente por meio de métodos computacionais (ISMAIL-ZADEH; TACKLEY, 2010).

Os modelos oceânicos são convencionalmente formulados como equações do movimento com forçantes distribuídas, condições iniciais e condições de contorno. As equações são resolvidas por uma aproximação numérica, produzindo campos de circulação oceânica por todo o domínio do modelo em algum intervalo de tempo. Se as observações reais do oceano estão disponíveis para o mesmo intervalo de tempo, então é possível fazer a comparação. No entanto, há usualmente uma discordância entre a saída do modelo e as observações do sistema real. É plausível que as entradas do modelo, tais como, condições iniciais, de contorno e forçantes, sejam ajustadas até que haja um acordo satisfatório com as observações (CHUA; BENNETT, 2001).

O melhor ajuste ou aproximação para a circulação estimada, requer uma extensão para a definição de um modelo oceânico. Além disso, para a equação de movimento, forçantes distribuídas, condições iniciais e de contorno adota-se a hipótese de que os respectivos erros seguem uma distribuição gaussiana. No caso mais simples, a média e covariâncias do erro serão prescritas com a hipótese de Gaussianidade. Em resumo, um modelo oceânico, segundo Chua e Bennett (2001) é uma hipótese nula para a distribuição de probabilidade dos erros nas componentes do modelo oceânico *forward*.

Este capítulo se propõe a apresentar os modelos utilizados, com a finalidade de testar as técnicas de assimilação de dados aqui empregadas. Utilizou-se dois modelos, um modelo linear de uma dimensão e o modelo de água rasa em duas dimensões.

### 2.1 Equação da onda linear

Este modelo é considerado como um exemplo que envolve uma “circulação oceânica” não conhecida  $\eta = \eta(x, t)$ , em que  $x$  e  $t$  são variáveis reais. A “bacia do oceano” é o intervalo  $0 \leq x \leq L$ , enquanto o tempo de interesse é  $0 \leq t \leq T$ . A “dinâmica

do oceano” é expressa pela equação diferencial parcial de primeira ordem, dada a seguir:

$$\frac{\partial \eta_F}{\partial t} + c \frac{\partial \eta_F}{\partial x} = F(x, t) \quad (2.1)$$

em que  $\eta$  é o deslocamento,  $c$  é uma constante conhecida, positiva, denominada de velocidade de fase,  $F = F(x, t)$  é um campo forçante especificado não homogêneo,  $t$  é o tempo e  $x$  é o espaço. O subíndice  $F$  indica a solução avançada, isto é, a solução *a priori*. A Equação 2.1 pode ser interpretada também como a equação da convecção, em que  $\eta$  é a concentração e  $c$  é o coeficiente de convecção.

Uma condição inicial é  $\eta(x, 0) = I(x)$ , para  $0 \leq x \leq L$ , em que  $I$  é especificado. Uma condição de contorno periódica é  $\eta(0, t) = C(t)$  para  $0 \leq t \leq T$ , em que  $C$  é especificado.

Para a integração numérica da Equação 2.1, a condição inicial periódica, utilizada na integração da equação da onda é a solução analítica da equação KdV (Korteweg-de Vries) avaliada em  $t$  é dada por:

$$\eta(x, t) = \eta_0 \frac{1}{\cosh^2[(x - vt)/\Delta]} \quad (2.2)$$

em que  $\eta_0$  é a amplitude do *sóliton*<sup>1</sup>,  $v$  é a velocidade de fase do *sóliton* e  $\Delta$  o tamanho da escala do *sóliton*:  $\Delta = \sqrt{\frac{12\beta}{\alpha\eta_0}}$  e  $v = c + \frac{\alpha\eta_0}{3}$ . Na Tabela 2.1, apresentam-se os parâmetros usados na integração do modelo. Na Figura 2.1, ilustra-se a evolução temporal em 2000 passos de tempo com 128 pontos na coordenada  $x$  para a Equação 2.1, projeções nas coordenadas  $(x, y, z)$ . Na Figura 2.2, tem-se a evolução temporal para a equação da onda nos instantes inicial (2.2(a)) e em  $t = 150$  (2.2(b)).

Tabela 2.1 - Parâmetros usados na integração da equação da onda.

Parâmetro	Valor
$\eta_0$	-60 m
$c$	2,42 $ms^{-1}$
$\alpha$	$-1,62 \times 10^{-2} s^{-1}$
$\beta$	$1,46 \times 10^5 m^3 s^{-1}$
$\Delta$	1340 m
$v$	2,75 $ms^{-1}$

<sup>1</sup>é uma onda solitária que mantém sua forma enquanto se propaga em velocidade constante.

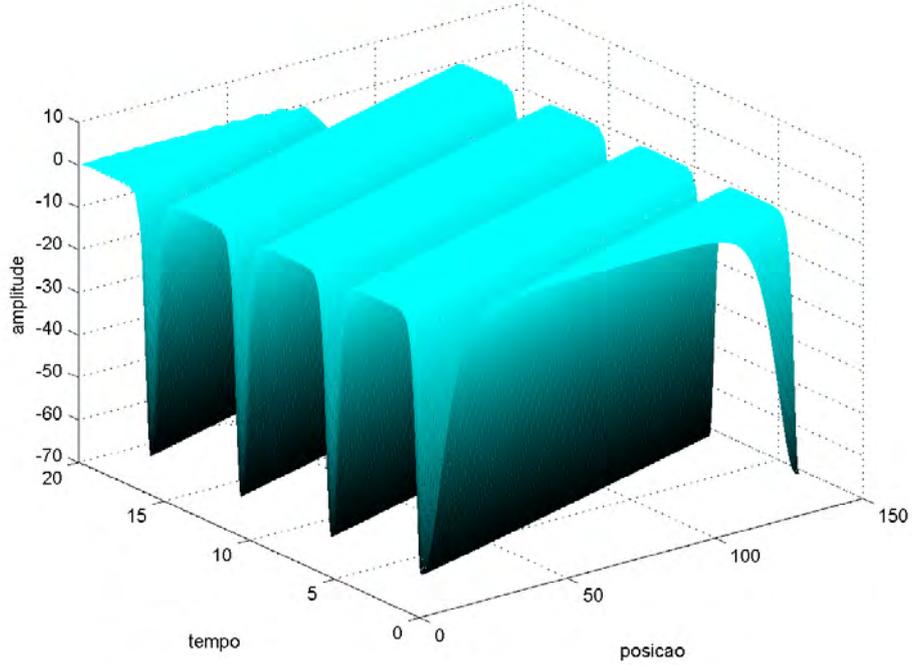


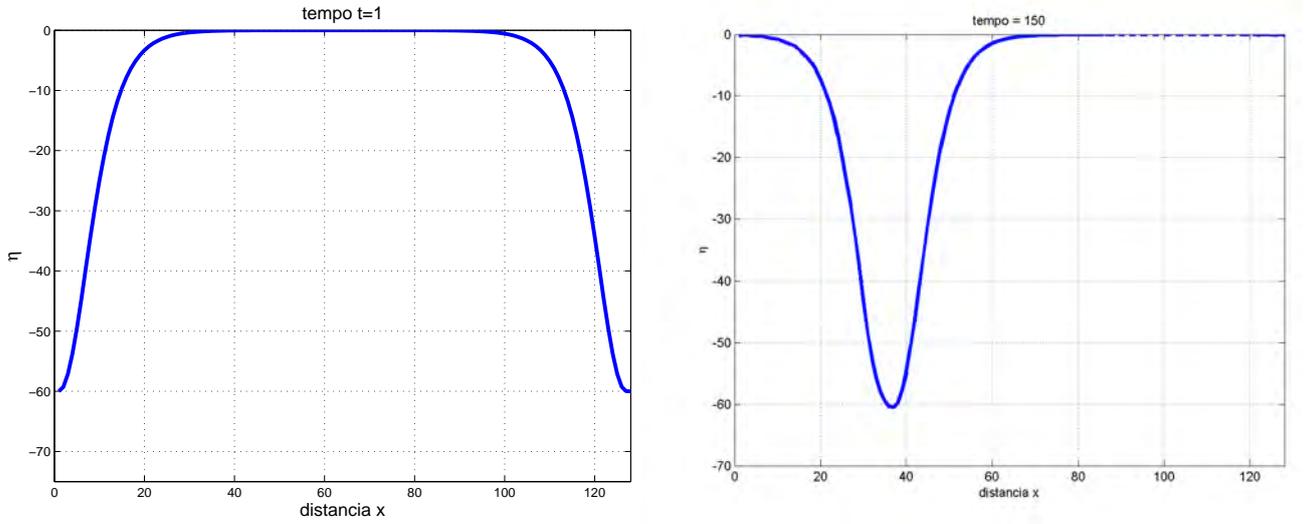
Figura 2.1 - Evolução temporal da Equação 2.1 com suas projeções nos eixos:  $x$  distância;  $y$ : tempo;  $z$ : amplitude do *sóliton*.

O modelo foi integrado com o método de Crank Nicholson (LYNCH, 2004; ISMAIL-ZADEH; TACKLEY, 2010) com derivada espacial de quarta ordem, dado pela equação a seguir:

$$\eta_p^{k+1} = \eta_p^k - c_1(\eta_{p-2}^{k+1} + 8\eta_{p-1}^{k+1} - 8\eta_{p+1}^{k+1} + \eta_{p+2}^{k+1}) - c_1(\eta_{p-2}^k + 8\eta_{p-1}^k - 8\eta_{p+1}^k) + \frac{1}{2}(F_p^{k+1} + F_p^k) \quad (2.3)$$

em que  $c_1 = \frac{c}{24} \frac{\Delta t}{\Delta x}$ ,  $F$  é a forçante do modelo,  $p = 1, nx$  e  $k = 1, nt$ . Rearranjando a Equação 2.3 tem-se:

$$c_1\eta_{p-2}^{k+1} - c_18\eta_{p-1}^{k+1} + \eta_p^{k+1} + c_18\eta_{p+1}^{k+1} - c_1\eta_{p+2}^{k+1} = -c_1\eta_{p-2}^k + c_18\eta_{p-1}^k + \eta_p^k - c_18\eta_{p+1}^k + c_1\eta_{p+2}^k + \frac{1}{2}(F_p^{k+1} + F_p^k) \quad (2.4)$$



(a) Integração numérica da Equação 2.1 no instante inicial. (b) Integração numérica da Equação 2.1 no instante  $t = 150$ .

Figura 2.2 - Evolução temporal da Equação 2.1.

Para a solução da Equação 2.4, é necessário resolver o sistema linear na forma:

$$A\eta^{(k+1)} = B\eta^k \quad (2.5)$$

Para efeito ilustrativo ao considerarmos  $nx = 8$ , as matrizes  $A$  e  $B$  obtidas são:

$$A = \begin{pmatrix} 1 & 8c_1 & -c_1 & 0 & 0 & 0 & c_1 & -8c_1 \\ -8c_1 & 1 & 8c_1 & -c_1 & 0 & 0 & 0 & c_1 \\ c_1 & -8c_1 & 1 & 8c_1 & -c_1 & 0 & 0 & 0 \\ 0 & c_1 & -8c_1 & 1 & 8c_1 & -c_1 & 0 & 0 \\ 0 & 0 & c_1 & -8c_1 & 1 & 8c_1 & -c_1 & 0 \\ 0 & 0 & 0 & c_1 & -8c_1 & 1 & 8c_1 & -c_1 \\ -c_1 & 0 & 0 & 0 & c_1 & -8c_1 & 1 & 8c_1 \\ 8c_1 & -c_1 & 0 & 0 & 0 & c_1 & -8c_1 & 1 \end{pmatrix} \quad (2.6)$$

$$B = \begin{pmatrix} -1 & -8c_1 & c_1 & 0 & 0 & 0 & -c_1 & 8c_1 \\ 8c_1 & -1 & -8c_1 & c_1 & 0 & 0 & 0 & -c_1 \\ -c_1 & 8c_1 & -1 & -8c_1 & c_1 & 0 & 0 & 0 \\ 0 & -c_1 & 8c_1 & -1 & -8c_1 & c_1 & 0 & 0 \\ 0 & 0 & -c_1 & 8c_1 & -1 & -8c_1 & c_1 & 0 \\ 0 & 0 & 0 & -c_1 & 8c_1 & -1 & -8c_1 & c_1 \\ c_1 & 0 & 0 & 0 & -c_1 & 8c_1 & -1 & -8c_1 \\ -8c_1 & c_1 & 0 & 0 & 0 & -c_1 & 8c_1 & -1 \end{pmatrix} \quad (2.7)$$

Também como exemplo, podemos fixar a seguinte condição de contorno periódica, isto é:

$$\eta_1^k = \eta_{nx}^k, \quad 1 \leq k \leq nt \quad (2.8)$$

### 2.1.1 Dados

Para o processo de assimilação de dados assume-se um número finito de observações, que são coletadas dentro de um domínio espacial ( $0 \leq x \leq L$ ) e temporal ( $0 \leq t \leq T$ ). As observações ( $d_m$ ) são medidas pontuais e imperfeitas das variáveis independentes  $\eta(x, t)$  coletadas em  $M$  pontos no espaço e no tempo ( $x_m, t_m$ ), isto é:

$$d_m = \eta(x_m, t_m) + \varepsilon_m, \quad 1 \leq m \leq M \quad (2.9)$$

em que  $\eta(x, t)$  é o campo de deslocamento real, que não é conhecido. O valor  $\varepsilon_m$  representa o erro de medida. Dado que a forçante, a condição inicial e os dados contêm erros, não se pode esperar que o modelo seja perfeitamente consistente com os dados. Então, espera-se que:

$$\eta_F(x, t) \neq d_m, \quad 1 \leq m \leq M \quad (2.10)$$

De modo análogo para o caso discreto, tem-se:

$$d_m = \eta_{p_m}^{k_m} + \varepsilon_m, \quad 1 \leq m \leq M \quad (2.11)$$

A [Figura 2.3](#) ilustra o domínio de integração do modelo, com os pontos de grade do modelo e as observações, em que  $p$  é o contador espacial e  $k$  o contador temporal.

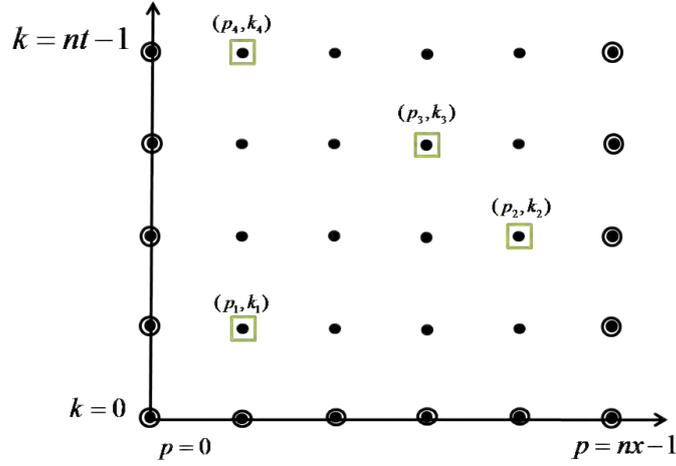


Figura 2.3 - Esquema dos pontos de grade do modelo (representado pelo círculos pretos), as observações representadas pelo quadrado verde, que são distribuídas no ponto de grade.

### 2.1.2 Descrição dos resíduos para a onda linear

Devido aos erros na estimativa *a priori* para  $F, I$  e  $C$  que são: forçante, condição inicial e condição de contorno, respectivamente. A circulação real deve satisfazer às Equações 2.12 e 2.13:

$$\frac{\partial \eta_F}{\partial t} + c \frac{\partial \eta_F}{\partial x} = F(x, t) + f(x, t) \quad 0 \leq x \leq L \quad (2.12)$$

$$\eta(x, 0) = I(x) + i(x) \quad 0 \leq x \leq L \quad (2.13)$$

$$\eta(0, t) = B(t) + b(t) \quad 0 \leq t \leq T \quad (2.14)$$

em que  $f(x, t)$  representa o erro na forçante,  $i(x)$  o erro na condição inicial,  $b(t)$  o erro na condição de contorno. A condição de contorno é periódica no tempo com  $\eta(0, t) = \eta(L, t)$  com  $0 \leq t \leq T$ .

Portanto, as Equações 2.9, 2.12, 2.13 e 2.14 correspondem aos *residuais* de observação, modelagem, condição inicial e condição de contorno, respectivamente. Os experimentos numéricos e discussões sobre os resultados para esse modelo encontram-se na Subseção 6.1.1 do Capítulo 6.

## 2.2 Modelo de água rasa linear 2D

As equações de água rasa descrevem a propagação de perturbações na água e a evolução de outros fluidos incompressíveis em resposta à aceleração gravitacional e rotacional. Os modelos de água rasa podem ser utilizados para prever a velocidade da água e seu nível em vários pontos em uma região do fluido em diferentes tempos (SAMPSON, 2008; RANDALL, 2006). O período de tempo de interesse para um modelo relativo a marés é de 12 a 24 horas, para tsunamis é de 15 a 30 minutos, enquanto o período de interesse para tempestades é de vários dias (SAMPSON, 2008). Algumas aplicações desse modelo podem ser vistas em Aravequia e Dias (2004), utilizado para a previsão de anomalias climáticas e em Cardoso et al. (2002).

As equações de água rasa são derivadas dos princípios de conservação de massa e de momento. Essas equações são um conjunto de equações diferenciais parciais hiperbólicas derivadas das equações de Navier-Stokes, no caso em que a escala de tamanho horizontal é muito maior que a escala de tamanho vertical. Sob essa condição, a conservação da massa implica que a velocidade vertical do fluido é pequena. Isto pode ser mostrado na equação de momento em que o gradiente de pressão vertical é aproximadamente hidrostático, e que os gradientes de pressão horizontal são devido ao deslocamento da superfície de pressão. Isto implica que o campo de velocidade seja aproximadamente constante através da profundidade do fluido. As equações de água rasa são derivadas, tomando a velocidade vertical e as variações por meio da profundidade do fluido iguais a zero nas equações de Navier-Stokes.

As variáveis independentes são a profundidade ou altura do fluido,  $H$ , e o campo de velocidade do fluido bi-dimensional,  $u$  e  $v$ . A força que age sobre o fluido é a gravidade, que é representada pela constante gravitacional,  $g$ .

O modelo abordado neste trabalho é o linear em duas dimensões dado a seguir:

$$\frac{\partial u}{\partial t} - fv + g\frac{\partial q}{\partial x} + r_u u = F_u \quad (2.15a)$$

$$\frac{\partial v}{\partial t} + fu + g\frac{\partial q}{\partial y} + r_v v = F_v \quad (2.15b)$$

$$\frac{\partial q}{\partial t} + H \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + r_q q = 0 \quad (2.15c)$$

sobre o domínio  $0 \leq x \leq X$  e  $0 \leq y \leq Y$ ,  $f$  representa o parâmetro de Coriolis,  $r_u$ ,

$r_v$ ,  $r_q$  são os coeficientes de amortecimento,  $u$  e  $v$  componentes de velocidade,  $q$  é a perturbação de superfície livre, ilustrado na Figura 2.4, se  $q \equiv q'$ , então o oceano está em balanço hidrostático ou em estado de equilíbrio,  $F_u$  e  $F_v$  são forçantes externas.

Com condições iniciais:

$$u(x, y, 0) = I^u(x, y) = 0$$

$$v(x, y, 0) = I^v(x, y) = 0$$

$$q(x, y, 0) = I^q(x, y) = 0$$

e condições de contorno, dadas a seguir:

$$u(x \pm X, y, t) = u(x, y, t) \quad (2.17a)$$

$$v(x \pm X, y, t) = v(x, y, t) \quad (2.17b)$$

$$q(x \pm X, y, t) = q(x, y, t) \quad (2.17c)$$

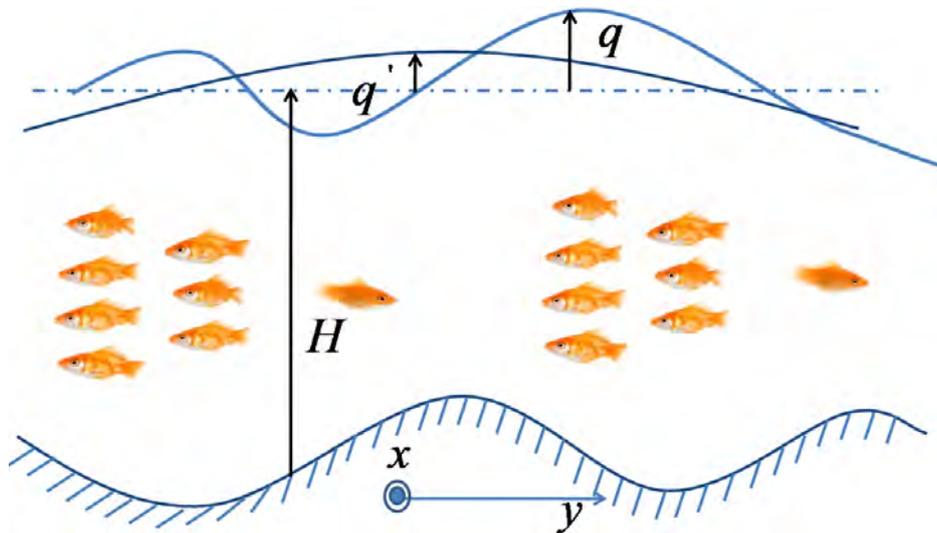


Figura 2.4 - Teoria de água rasa.

Fonte: Adaptado de Bennett (2002) página 127.

### 2.2.1 Discretização para o modelo de água rasa 2D

Nesta seção, descreve-se a discretização para integração do modelo de água rasa 2D descrito na seção 2.2. O modelo foi discretizado espacialmente com o método de grade-C de Arakawa, veja a Figura 2.6 e Figura 2.7 com o método *forward-backward* para a discretização temporal (MESINGER; ARAKAWA, 2002). Para as condições de contorno, tem-se as paredes norte e sul rígidas, isto é,  $u(x, 0, t) = v(x, Y, t) = 0$ , enquanto, todos os campos são periódicos na direção  $x$ , veja Figura 2.5.

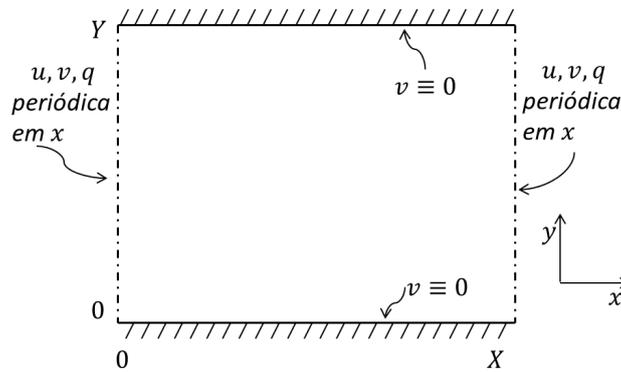


Figura 2.5 - Canal periódico com paredes rígidas no plano  $xy$ .  
Fonte: Adaptado de Bennett (2002) página 196.

As forçantes do modelo são:

$$\begin{aligned} F_u &= -\frac{C_d \rho_a u_a^2}{H \rho_w} \\ F_v &= 0 \end{aligned}$$

em que  $C_d$  é o coeficiente de arrasto,  $\rho_a$  é a densidade do ar,  $\rho_w$  densidade da água,  $u_a$  é o vento zonal. As condições de contorno rígidas são dadas por:

$$v_{i,1}^k = 0 \text{ e } v_{i,NJ}^k = 0 \quad (2.18)$$

As condições de contorno periódicas são:

$$\begin{aligned}
u_{0,j}^k &= u_{NI,j}^k & (2.19) \\
u_{NI+1,j}^k &= u_{1,j}^k \\
v_{0,j}^k &= v_{NI,j}^k \\
v_{NI+1,j}^k &= v_{1,j}^k \\
q_{0,j}^k &= q_{NI,j}^k \\
q_{NI+1,j}^k &= q_{1,j}^k
\end{aligned}$$

As equações discretizadas para o modelo representado pela [Equação 2.15](#) pelo método de grade-C de Arakawa são dadas a seguir:

$$\frac{q_{i,j}^{k+1} - q_{i,j}^k}{\Delta t} + H \left( \frac{u_{i+1,j}^k - u_{i,j}^k}{\Delta x} + \frac{v_{i,j+1}^k - v_{i,j}^k}{\Delta y} \right) + r_q q_{i,j}^k = 0 \quad (2.20)$$

$$\begin{aligned}
\frac{u_{i,j}^{k+1} - u_{i,j}^k}{\Delta t} - f \left( \frac{v_{i,j+1}^k + v_{i,j}^k + v_{i-1,j+1}^k + v_{i-1,j}^k}{4} \right) + \\
g \left( \frac{q_{i,j}^{k+1} - q_{i-1,j}^{k+1}}{\Delta y} \right) + r_u u_{i,j}^k = F_{u_{i,j}^k} \quad (2.21)
\end{aligned}$$

$$\begin{aligned}
\frac{v_{i,j}^{k+1} - v_{i,j}^k}{\Delta t} + f \left( \frac{u_{i+1,j}^k + u_{i,j}^k + u_{i+1,j-1}^k + u_{i,j-1}^k}{4} \right) + \\
g \left( \frac{q_{i,j}^{k+1} - q_{i,j-1}^{k+1}}{\Delta y} \right) + r_v v_{i,j}^k = F_{v_{i,j}^k} \quad (2.22)
\end{aligned}$$

A [Tabela 2.2](#) apresenta os parâmetros usados para a integração do modelo de água rasa 2D. Adotou-se esses parâmetros para reproduzir o experimento de [Bennett \(2002\)](#).

A [Figura 2.6](#) ilustra a discretização espacial pelo método de Grade C de Arakawa e [Figura 2.7](#) ilustra a discretização temporal para o modelo de água rasa 2D discretizado de acordo com as equações [2.20](#), [2.21](#) e [2.20](#). Na [Figura 2.8](#), tem-se a projeção da variável  $q$  do modelo nas coordenadas  $(x, y, z)$ .

Tabela 2.2 - parâmetros de discretização do modelo 2D.

parâmetro	valor
H	5000 m
T	$1,8 \times 10^4$ s
g	$9,806 m s^{-2}$
f	$1,0 \times 10^{-4} s^{-1}$
$r_u$	$(1,8 \times 10^4 s)^{-1}$
$r_v$	$(1,8 \times 10^4 s)^{-1}$
$r_q$	$(1,8 \times 10^4 s)^{-1}$
$C_d$	$1,6 \times 10^{-3}$
$\rho_a$	$1,275 kg m^{-3}$
$\rho_w$	$1,0 \times 10^3 kg m^{-3}$

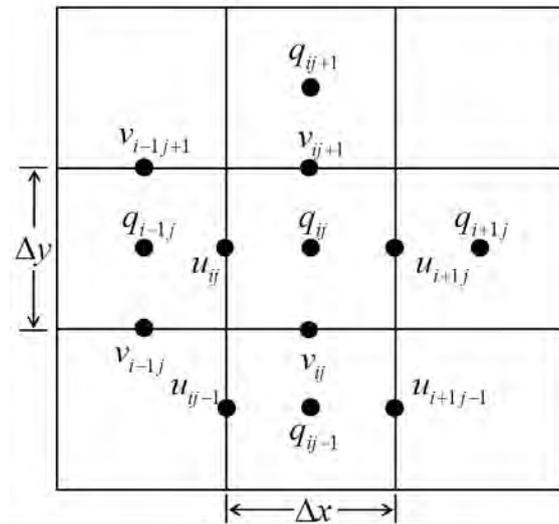


Figura 2.6 - Esquema do método de Grade C de Arakawa para diferenças espaciais.  
 Fonte: Adaptado de Bennett (2002) página 198.

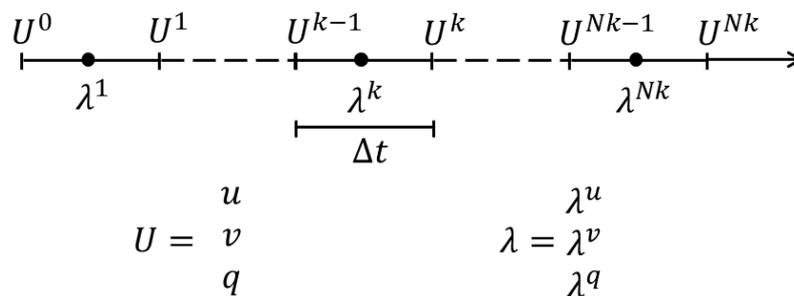


Figura 2.7 - Esquema do método de grade C de Arakawa para diferenças temporais.  
 Fonte: Adaptado de Bennett (2002) página 198.

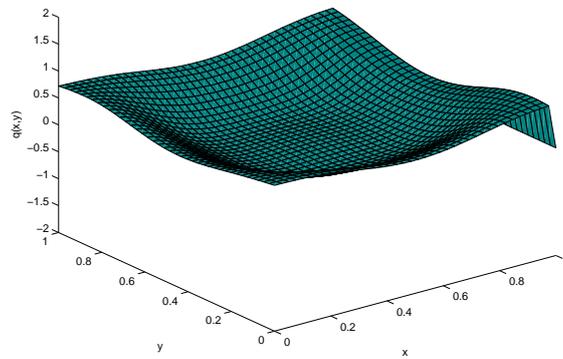
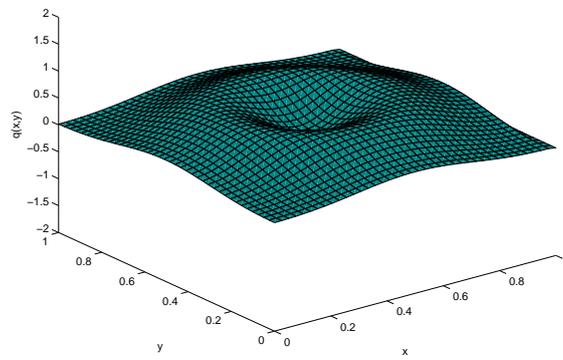
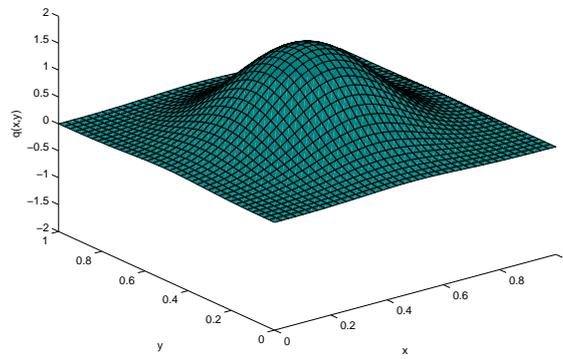
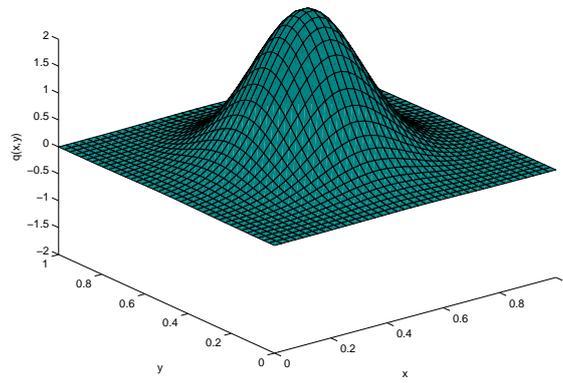


Figura 2.8 - Evolução temporal da variável  $q$  nas projeções  $(x, y)$ .

### 3 ESTIMAÇÃO POR MÍNIMOS QUADRADOS E FILTRO DE KALMAN

Neste capítulo, apresenta-se o método do Filtro de Kalman, que será deduzido a partir do método de mínimos quadrados. Primeiro, faz-se uma introdução ao método de mínimos quadrados, mínimos quadrados ponderados e recursivos. Por fim, apresenta-se o algoritmo do Filtro de Kalman. Este método combina a formulação matemática de um sistema com observações desse sistema. Aqui esta metodologia é usada no contexto de assimilação de dados.

De acordo com Polavarapu (2004) assimilação de dados é um problema que pode ser descrito como: dado um conjunto de observações e um modelo de algum fenômeno físico determinar o conhecimento que as observações podem fornecer a respeito das variáveis de estado do modelo.

Um modelo dinâmico para aproximar a natureza física de algum fenômeno consiste de um conjunto de equações, em geral não lineares acopladas para cada variável de estado de interesse. Propriedades fundamentais que caracterizam o sistema podem aparecer como parâmetros. Em princípio, esses parâmetros do sistema podem ser estimados diretamente das medidas, que são dados observados a partir de experimentos (ROBINSON; LERMUSIAUX, 2000).

O estímulo para o desenvolvimento da teoria da estimação veio dos estudos astronômicos em que o movimento dos planetas e cometas foram estudados a partir de dados de medição telescópica. O problema de estimação considerado consistiu em inferir os valores dos parâmetros que caracterizam o movimento dos corpos a partir de dados medidos. Para resolver o problema relativo às órbitas dos corpos celestes, o método de mínimos quadrados foi idealizado por Karl Friedrich Gauss. Há uma certa controvérsia a cerca do desenvolvimento dessa ferramenta. O conflito surge pelo fato da não publicação da descoberta de Gauss em 1795 e a publicação dos resultados produzidos, independentemente, por Legendre em 1806 (SORENSEN, 1970).

A teoria da estimação é um ramo da probabilidade e estatística que trata do problema de inferir informações a respeito de propriedades das variáveis aleatórias e processos estocásticos, dado um conjunto de amostras observadas. O termo *estimador* ou *filtro* é comumente usado para se referir ao sistema que é designado para

extrair informações sobre uma quantidade prescrita de interesse de dados ruidosos. A teoria da estimação tem um campo de aplicação muito amplo. Por exemplo, comunicações, radar, navegação, engenharia biomédica e financeira, entre outras aplicações (HAYKIN; KAILATH, 2002; JAZWINSKI, 1970; POLAVARAPU, 2004).

A aplicação de interesse desse trabalho está voltada para o problema de circulação oceânica. O estudo do oceano, tecnologia marinha e operações requer o conhecimento da distribuição e evolução no espaço e no tempo das características do mar. As funções no espaço e no tempo que caracterizam como o estado do sistema varia ao longo do tempo sob observação, são classicamente designadas como campo. A determinação da distribuição ou evolução das variáveis de estado define um problema de estimação em três ou em quatro dimensões (ROBINSON; LERMUSIAUX, 2000).

Há três tipos básicos de estimação, que são: filtragem, suavização no sentido da técnica *smoothing* e previsão, todos podendo ser executados por um estimador. As diferenças entre estas operações são ilustradas na Figura 3.1 e descritas a seguir de acordo com Haykin e Kailath (2002).

- *Filtragem* é uma operação que envolve a extração de informação sobre uma quantidade de interesse no tempo  $t$  usando os dados medidos até o tempo  $t$ , isto é, são utilizados dados medidos no passado e no presente.
- *Suavização* é uma forma de estimação *a posteriori*, em que os dados medidos depois do tempo de interesse  $t'$  são usados na estimação.
- *Previsão* é a estimativa de um estado futuro. O objetivo é inferir informação sobre a quantidade de interesse que será alterada em algum tempo  $t + \tau$  no futuro (para algum  $\tau > 0$ ) usando inclusive dados medidos no tempo  $t$ .

Estimação de parâmetros via assimilação de dados vem tendo um impacto cada vez mais significativo em ciências oceânicas. Isto engloba teorias usadas para estimar o estado de um sistema pela combinação, usualmente, de uma aproximação estatística, de todo o conhecimento disponível do sistema incluindo dados observados e modelos teóricos.

A hipótese *a priori* e o critério de estimação são cruciais no processo de estimação, uma vez que eles determinam a influência da dinâmica e das observações sobre a

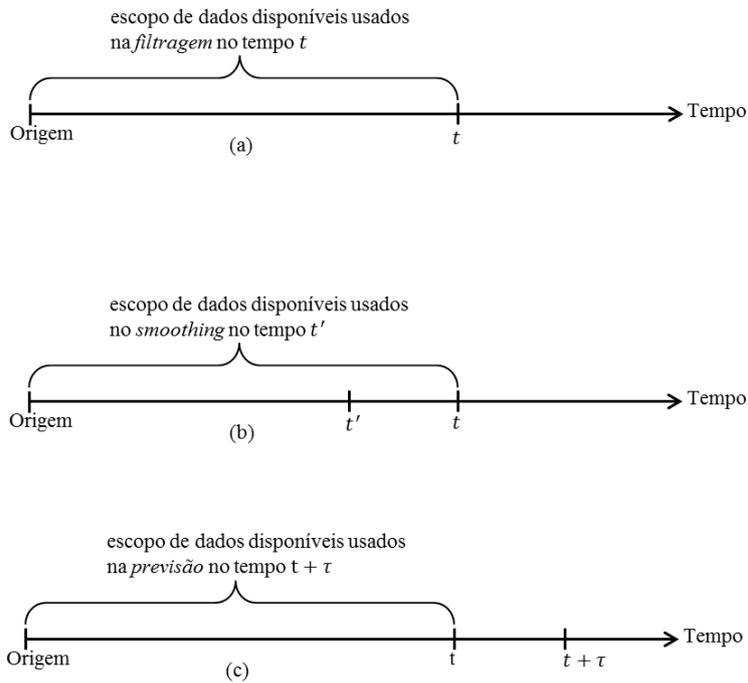


Figura 3.1 - Ilustração dos três tipos básicos de estimação: (a) filtragem, (b) *smoothing* e (c) previsão.

Fonte: Adaptado de Haykin e Kailath (2002) página 3.

variável estimada. Assim, a informação que se deseja conhecer é dada pela função densidade de probabilidade (*fdp*) dada por:

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) = \frac{p_{\mathbf{xy}}(\mathbf{x}, \mathbf{y})}{p_{\mathbf{y}}(\mathbf{y})} \quad (3.1)$$

em que  $p_{\mathbf{y}}(\mathbf{y}) \neq 0$ . No entanto, é muito difícil obter essa informação, na prática, principalmente ao se considerar modelos envolvendo várias escalas e complexos. O fato é que existem importantes atributos de  $p_{\mathbf{x}|\mathbf{y}}$  que auxiliam na estimativa de  $\mathbf{x}$ . Denota-se essa estimativa por  $\hat{\mathbf{x}}$  para diferenciar da variável aleatória  $\mathbf{x}$ . Esta estimativa dependerá dos dados observados. Assim, quando  $\hat{\mathbf{x}} = \hat{\mathbf{x}}(\mathbf{y})$  é visto como uma função de  $\mathbf{y}$ ,  $\hat{\mathbf{x}}$  é denominado de estimador. Um estimador informa a melhor estimativa de  $\hat{\mathbf{x}}$  para um dado  $\mathbf{y}$ , tendo por base algum princípio (POLAVARAPU, 2004).

Estimadores podem ter várias propriedades desejáveis, por exemplo, ser enviesado

(imparcial), isto é,

$$E(\mathbf{x}^t - \hat{\mathbf{x}}) = 0 \quad (3.2)$$

em que  $\mathbf{x}^t$  é o valor verdadeiro de  $\mathbf{x}$  projetado sobre a base do modelo. Se  $\mathbf{x}$  é tendencioso, o viés (*bias*) é definido por:

$$b(\hat{\mathbf{x}}) = E(\mathbf{x}^t - \hat{\mathbf{x}}) = E(\mathbf{x}^t) - \hat{\mathbf{x}} \quad (3.3)$$

Uma vez que a média é com relação a  $\mathbf{x}$  e o estimador é apenas uma função de  $\mathbf{y}$ , então a média do estimador é um estimador. Dados dois estimadores, escolhe-se aquele com menor variância. Portanto, se os dois estimadores são tendenciosos, então aquele com menor variância não é necessariamente o preferível, uma vez que eles podem ter grande *bias*. Neste caso, é preferível aquele com menor *erro médio quadrático*. O Erro Médio Quadrático (EMQ) é definido por:

$$\begin{aligned} EMQ &= E[(\mathbf{x}^t - \hat{\mathbf{x}})^2] = E[(\mathbf{x}^t - E(\mathbf{x}^t) + E(\mathbf{x}^t) - \hat{\mathbf{x}})^2] & (3.4) \\ &= E[(\mathbf{x}^t - E(\mathbf{x}^t) + b(\hat{\mathbf{x}}))^2] \\ &= E[(\mathbf{x}^t - E(\mathbf{x}^t))^2] + 2E[(\mathbf{x}^t - E(\mathbf{x}^t))b(\hat{\mathbf{x}})] + E[b(\hat{\mathbf{x}})^2] \\ &= E[(\mathbf{x}^t - E(\mathbf{x}^t))^2] + 2E[(\mathbf{x}^t - E(\mathbf{x}^t))]b(\hat{\mathbf{x}}) + b(\hat{\mathbf{x}})^2 \\ &= \sigma^2(\mathbf{x}^t) + b(\hat{\mathbf{x}})^2 & (3.5) \end{aligned}$$

Para estimadores não tendenciosos, o EMQ é igual a variância  $\sigma^2(\mathbf{x}^t)$ , mas para estimadores tendenciosos, o EQM é igual a variância mais o quadrado do *bias*. A Raiz do Erro Quadrático Médio é definida por:

$$RMEQ = \sqrt{EMQ} \quad (3.6)$$

Ao se trabalhar com duas variáveis, por exemplo, a correlação entre essas variáveis poderá ser alta. Essa correlação é representada pela equação a seguir:

$$\rho_{xy} = \frac{cov(x, y)}{\sigma_x \sigma_y} = \frac{E[(x - \sigma_x)(y - \sigma_y)]}{\sigma_x \sigma_y} \quad (3.7)$$

em que *cov* é a covariância,  $E[.]$  é o valor esperado ou média. A variância é definida por:

$$\sigma_x^2 = E[(x - E(x))^2] = E[(x - \mu_x)^2] \quad (3.8)$$

Pode-se ilustrar a aplicação desses conceitos em um sistema massa mola, através da

lei de Hooke, em que há uma relação entre força e deformação de certo material. Neste caso, o problema consiste em determinar qual é o melhor ajuste para a constante ( $k$ ) que satisfaça a relação  $F = -kx$ . Um exemplo de tal sistema é uma mola. Uma vez que exercemos uma força esticando ou comprimindo uma mola, esta se deforma, e como consequência surge uma força restauradora, também chamada de força elástica, que age no sentido de recuperar o comprimento original da mola. A [Figura 3.2](#) ilustra uma mola com seu comprimento natural (em repouso) e em seguida a mola sob a ação de uma força  $F$  que a deforma esticando-a em um comprimento  $x$ .

Para calcular parâmetros similares ao problema da Lei de Hook, pode-se fazer uso da teoria da estimação. Neste contexto, faz-se necessário que o modelo matemático esteja disponível. O uso da dinâmica do modelo é de importância fundamental para uma estimação eficiente do parâmetro. Para a Lei de Hook, por exemplo, o modelo matemático é a própria Lei  $F = -kx$ .

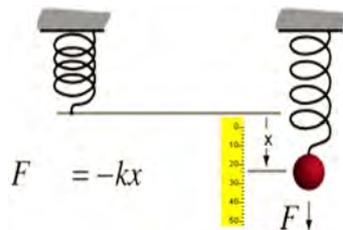


Figura 3.2 - Ilustração da Lei de Hook.

Existem várias estratégias para a estimação do parâmetro  $k$ , tais como:

- Mínimos quadrados
- Mínimos quadrados com restrições
- Métodos Baesianos
- Filtro de Wiener
- Filtro de Kalman
- Filtro de Partícula

- Mínima Variância
- Redes Neurais Artificiais

Um outro exemplo interessante para o problema de estimação é o “*falling body*”, que encontra-se em [Kleeman \(2012\)](#). A seguir descreve-se o método de mínimos quadrados para a construção do método de Filtro de Kalman.

### 3.1 Mínimos quadrados

A própria denominação do termo de mínimos quadrados nos remete a pensar no quadrado do erro. Esta metodologia é uma técnica de otimização que tem por objetivo determinar o melhor ajuste para um conjunto de dados. Consideremos o sistema  $\mathbf{H}\mathbf{x} = \mathbf{y}$ , quando esse sistema não possui solução, tem-se que, o melhor  $\mathbf{x}$  possível é aquele que minimiza a diferença quadrática  $\|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2$ . No entanto, alguns questionamentos são:

- em que sentido  $\mathbf{x}$  é a melhor solução;
- as medidas  $\mathbf{y}_i$  são igualmente realizáveis;
- é razoável associar maior peso as medidas mais frequentes;

Para a definição do problema de estimação, consideremos que  $m$  quantidades medidas estão disponíveis nos tempos discretos  $(t_1, t_2, \dots, t_n)$  e são denotadas em cada tempo  $t_i$  como  $y_i$  e que o parâmetro  $x$  é determinado a partir dos dados observados e da relação:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e} \quad (3.9)$$

em que  $\mathbf{y}$  é um vetor de observação  $m$ -dimensional,  $\mathbf{x}$  é o vetor de variável de estado *a priori*  $n$ -dimensional, o vetor  $\mathbf{e}$  representa o erro de observação  $m$ -dimensional que ocorre em cada passo de tempo.  $\mathbf{H}$  é uma matriz  $m \times n$  que representa o mapeamento das variáveis do modelo para o espaço das observações, conhecido como o operador de observação,  $\mathbf{e}$  é uma variável aleatória.

Se a primeira observação é mais confiável que a segunda, considerando um grande conjunto de medidas, deseja-se minimizar  $w_1^2(Hx - y)_1^2 + w_2^2(Hx - y)_2^2 + \dots$  com  $w_1 > w_2$ . Uma correlação entre as observações pode ser calculada por uma matriz não diagonal de pesos  $\mathbf{W}$ .

O primeiro passo é encontrar a melhor solução  $\mathbf{x}$  para um dado  $\mathbf{W}$ . O próximo passo é decidir sobre uma matriz de ponderação “correta”, ou o equivalente a  $\mathbf{C} = \mathbf{W}^T \mathbf{W}$ , se há alguma informação estatística a respeito dos erros. Espera-se que a medida  $y_i$  contenha um erro  $e_i$ , é fato que, não conhecemos esse erro. Contudo, é frequentemente razoável supor que se conhece a média do erro, ou o valor esperado, de  $e_i$ ,  $e_i^2$  e  $e_i e_j$ . Isto determinará um  $\mathbf{W}$  ótimo.

O terceiro passo ocorre com a chegada de novas medidas. Suponha que o ajuste de dados é feito por uma linha reta, e o experimento produz uma medida a cada segundo. Cada novo ponto de dado significa uma mudança na melhor reta. A cada chegada de observação no tempo, resolve-se uma nova equação normal  $\mathbf{H}^T \mathbf{H} \mathbf{x} = \mathbf{H}^T \mathbf{y}$ . No entanto, deseja-se apenas uma mudança em  $\mathbf{x}$ . O que é necessário para o cálculo em tempo real é o método de **mínimos quadrados recursivos**. Portanto, procura-se a mudança em  $(\mathbf{H}^T \mathbf{H})^{-1}$  ou  $(\mathbf{H}^T \mathbf{C} \mathbf{H})^{-1}$  produzida por uma nova linha de  $\mathbf{H}$ .

Finalmente, há a possibilidade que o próprio modelo seja não estacionário; o experimento não está estabelecido ainda. Neste caso, uma quantidade diferente  $\mathbf{x}_i$  é estimada depois de cada passo de tempo. Se o estado  $\mathbf{x}_{i+1}$  é não correlacionado a  $\mathbf{x}_i$ , os cálculos anteriores serão desnecessários. É muito mais comum ter uma relação linear  $\mathbf{x}_{i+1} = \mathbf{F}_i \mathbf{x}_i$  com  $\mathbf{F}_i$  conhecido e  $\mathbf{x}_i$  não conhecido (STRANG, 1986). Cada passo traz uma nova informação por meio de uma medida  $\mathbf{y}_i$  de  $\mathbf{H}_i \mathbf{x}_i$ , porém existem erros nas medidas e na relação linear:

$$\begin{aligned} \mathbf{y}_i &= \mathbf{H}_i \mathbf{x}_i + \mathbf{e}_i \\ \mathbf{x}_{i+1} &= \mathbf{F}_i \mathbf{x}_i + \varepsilon_i \end{aligned} \tag{3.10}$$

O problema consiste em separar o sinal do ruído. A solução para problemas não estacionários pode ser dada pelo Filtro de Kalman. O FK é um filtro digital (o processo é realizado em passos discretos) e recursivo, a estimativa de  $\mathbf{x}_{i+1}$  é calculada da estimativa anterior de  $\mathbf{x}_i$  e de uma nova medida de acordo com a [Equação 3.11](#). O valor mais provável é definido como o valor que minimiza a soma dos quadrados dos resíduos, que é a diferença entre o observado e os valores de medidas calculados (SORENSEN, 1970).

O método de mínimos quadrados consiste em determinar o valor mais provável de  $\mathbf{x}$ , denotado aqui por  $\hat{\mathbf{x}}$ , que minimiza a soma dos quadrados dos resíduos. Portanto,

escolhe-se o  $\mathbf{x}$  que minimiza o funcional a seguir:

$$J(\mathbf{x}) = \frac{1}{2} \sum_{i=0}^n (\mathbf{y}_i - \mathbf{H}_i \mathbf{x})^T \mathbf{W}_i (\mathbf{y}_i - \mathbf{H}_i \mathbf{x}) \quad (3.11)$$

Os elementos da matriz  $\mathbf{W}_i$  são selecionados para indicar o grau de confiança que se pode colocar em cada medida.

Para uma melhor compreensão de como usar o método de mínimos quadrados como um estimador, considera-se o seguinte modelo linear:

$$y(x) = \beta_0 + \beta_1 x \quad (3.12)$$

Deseja-se estimar  $\beta_0$  e  $\beta_1$  constantes, de acordo com algum critério, por exemplo, pela minimização dos resíduos, isto é, pela minimização do funcional quadrático dado por:

$$J(\beta_0, \beta_1) = \|e\|^2 = \|y^{obs} - y^{mod}\|^2 = \|y^{obs} - (\beta_0 + \beta_1 x)\|^2 \quad (3.13)$$

em que  $y^{obs}$  representa o dado observado e  $y^{mod}$  o dado do modelo. Considerando um número finito de medidas e usando a forma discreta para [Equação 3.13](#), tem-se

$$\begin{aligned} \delta J(\beta_0, \beta_1) &= \delta \|y^{obs} - y^{mod}\|^2 \\ &= \delta \|y^{obs} - (\beta_0 + \beta_1(x))\|^2 \\ &= \delta \left\| \sum_i [(y^{obs})_i - (\beta_0 + \beta_1 x_i)] \right\|^2 \\ &= \delta e \end{aligned} \quad (3.14)$$

de onde chega-se na condição:

$$\delta J[y] = \nabla J \delta x = 0 \Rightarrow \nabla J = 0 \quad (3.15)$$

em que o residual  $e$  é um vetor de tamanho  $m$ :

$$\|e\|^2 = e^T e = e^2 e = e_1^2 + e_2^2 + \dots + e_m^2 \quad (3.16)$$

$$\|e\|^2 = [y_1^{obs} - (\beta_0 + \beta_1 x_1)]^2 + [y_2^{obs} - (\beta_0 + \beta_1 x_2)]^2 + \dots + [y_m^{obs} - (\beta_0 + \beta_1 x_m)]^2 \quad (3.17)$$

Minimizando o funcional 3.13 por regressão linear tem-se:

$$\nabla J \equiv \frac{\partial J(\beta_0, \beta_1)}{\partial(\beta_0, \beta_1)} \equiv \begin{bmatrix} \frac{\partial J}{\partial \beta_0} \\ \frac{\partial J}{\partial \beta_1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.18)$$

as derivadas são dadas por:

$$\frac{\partial J}{\partial \beta_0} = -2 \sum_{i=1}^m [y_i^{obs} - (\beta_0 + \beta_1 x_i)] \quad (3.19)$$

$$\frac{\partial J}{\partial \beta_1} = -2 \sum_{i=1}^m x_i [y_i^{obs} - (\beta_0 + \beta_1 x_i)] \quad (3.20)$$

Em forma de sistema, tem-se:

$$\begin{cases} m\beta_0 + \beta_1 \sum_{i=1}^m x_i & = \sum_{i=1}^m y_i^{obs} \\ \beta_0 \sum_{i=1}^m x_i + \beta_1 \sum_{i=1}^m x_i^2 & = \sum_{i=1}^m x_i y_i^{obs} \end{cases} \quad (3.21)$$

Na forma matricial

$$\begin{bmatrix} m & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m y_i^{obs} \\ \sum_{i=1}^m x_i y_i^{obs} \end{bmatrix} \quad (3.22)$$

A solução para o sistema dado pela Equação 3.22 é:

$$\beta_1 = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i^{obs} - \bar{y})}{\sum_{i=1}^m (x_i - \bar{x})^2} \quad (3.23)$$

$$\beta_0 = \frac{1}{m} \left( \sum_{i=1}^m y_i - \beta_1 \sum_{i=1}^m x_i \right) = \langle y^{obs} \rangle - \beta_1 \langle x \rangle = \bar{y}^{obs} - \bar{x} \quad (3.24)$$

O sistema 3.22 pode ser escrito como  $\mathbf{H}\beta = \mathbf{y}$  e o funcional 3.15 por  $J(\beta) = (\mathbf{H}\beta - \mathbf{y})^T (\mathbf{H}\beta - \mathbf{y}) \equiv e^T e$ , cuja a solução é  $\mathbf{H}^T \mathbf{H}\beta = \mathbf{H}^T \mathbf{y}$ , o que implica que  $\beta = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}$  é a solução algébrica. A matriz  $(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$  é denominada de matriz de inversa generalizada de Moore-Penrose (LAWSON; HANSON, 1974). O Teorema de Gauss-Markov garante que para um modelo linear em que o erro tem média zero, não correlacionado e tem a mesma variância, o melhor estimador linear dos

coeficientes é o estimador de mínimos quadrados (SORENSEN, 1970; JAZWINSKI, 1970; POLAVARAPU, 2004).

### 3.2 Mínimos quadrados ponderados

O comum para problemas de mínimos quadrados é ter um sistema  $\mathbf{H}\mathbf{x} = \mathbf{y}$  sem solução, pois a matriz  $\mathbf{H}$  é não inversível. Neste caso, o uso da inversa de Moore-Penrose, por meio da decomposição em valores singulares (SVD) é uma solução apropriada (CAMPOS VELHO; CLAEYSSEN, 1992).  $\mathbf{H}$  tem  $m$  linhas e  $n$  colunas, com  $m > n$ ; há mais dados observados  $y_1, \dots, y_m$  e parâmetros  $x_1, \dots, x_n$ . A melhor escolha  $\hat{\mathbf{x}}$ , é aquela que minimiza o tamanho do vetor de erro  $\mathbf{e} = \mathbf{y} - \mathbf{H}\hat{\mathbf{x}}$ , assim como visto na seção 3.1. Se a medida do comprimento do vetor é feita de modo usual, assim que  $\|\mathbf{e}\|^2 = (\mathbf{y} - \mathbf{H}\hat{\mathbf{x}})^T(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}})$  é a soma dos quadrados dos erros. De acordo com Strang (1986) minimiza-se o erro quadrático conforme a equação a seguir:

$$\mathbf{H}^T\mathbf{H}\hat{\mathbf{x}} = \mathbf{H}^T\mathbf{y}, \text{ ou } \hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y} \quad (3.25)$$

Geometricamente,  $\mathbf{H}\hat{\mathbf{x}}$  é uma projeção de  $\mathbf{y}$ , isto é, o vetor mais próximo à  $\mathbf{y}$  entre todos os vetores  $\mathbf{H}\mathbf{x}$  possíveis. Estes candidatos preenchem o espaço coluna de  $\mathbf{H}$ , e o mínimo quadrado escolhido:

$$\mathbf{H}\hat{\mathbf{x}} = \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y} = \mathbf{P}\mathbf{y} \quad (3.26)$$

é uma projeção de  $\mathbf{y}$  sobre este espaço coluna.

Algebricamente, determina-se a equação normal pela multiplicação em ambos os lados da equação sem solução  $\mathbf{H}\mathbf{x} = \mathbf{y}$  por  $\mathbf{H}^T$ , que produz uma matriz quadrada  $\mathbf{H}^T\mathbf{H}$ ; obtendo-se  $n$  equações para os valores não conhecidos de  $x_1, \dots, x_n$ . Assume-se que a matriz  $\mathbf{H}$  tem posto cheio (*full rank*, todas as suas colunas são independentes), logo  $\mathbf{H}^T\mathbf{H}$  possui inversa e  $\hat{\mathbf{x}}$  é completamente determinado. Em aplicações práticas não é usual determinar  $\hat{\mathbf{x}}$  por meio da Equação 3.26, devido as dificuldades para o cálculo da inversa de  $\mathbf{H}^T\mathbf{H}$ . O cálculo dessas matrizes é feito por algoritmos estáveis (GOLUB; LOAN, 1989).

Agora, supõe-se que  $m$  medidas não são igualmente realizáveis e que as medidas  $\mathbf{y}$  não têm a mesma precisão. Isto é, supondo-se que a medida  $y_1$  seja mais precisa que a medida  $y_2$ , é natural atribuir maior peso para  $y_1$ . Assim, deve-se minimizar o erro

médio quadrático dado por  $\mathbf{W}\mathbf{e}^2$ , em que:

$$\|\mathbf{W}\mathbf{e}^2\| = w_1^2 e_1^2 + w_2^2 e_2^2 + \dots + w_m^2 e_m^2 \quad (3.27)$$

Se as medidas são independentes a matriz de pesos  $\mathbf{W}$  é diagonal, geralmente em mínimos quadrados  $\mathbf{W} = \mathbf{I}$ . Se as medidas são correlacionadas, isto é, o erro em  $y_i$  dependente de  $y_j$ , a matriz  $\mathbf{W}$  terá elementos fora da diagonal principal (STRANG, 1986).

A solução que minimiza  $\mathbf{W}\mathbf{H}\mathbf{x} = \mathbf{W}\mathbf{y}$  é dada por:

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{C} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{C} \mathbf{y} \quad (3.28)$$

em que  $\mathbf{C} = \mathbf{W}^T \mathbf{W}$ . Tomando

$$\mathbf{L} = (\mathbf{H}^T \mathbf{C} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{C} \quad (3.29)$$

tem-se  $\hat{\mathbf{x}} = \mathbf{L}\mathbf{y}$ . A solução de mínimos quadrados ponderados para um sistema sobredeterminado  $\mathbf{H}\mathbf{x} = \mathbf{y}$  depende linearmente de  $\mathbf{y}$ . Se  $\mathbf{W} = \mathbf{I}$  retorna-se ao problema de mínimos quadrados padrão apresentado na seção 3.1 (STRANG, 1986).

### 3.2.1 A escolha dos pesos

O próximo passo é escolher a matriz  $\mathbf{W}$ . Para tanto, extraem-se as propriedades estatísticas do experimento. Para o cálculo de  $\mathbf{W}$ , primeiro assume-se que o ruído da observação  $\mathbf{e} = \mathbf{y} - \mathbf{H}\mathbf{x}$  é não tendencioso. O valor esperado<sup>1</sup> é  $E[e_i] = 0$  e a variância é dada por  $E[e_i^2]$ . Apenas o desvio padrão, que é a raiz quadrada da variância, é necessário para determinar os pesos. Isto significa que, tendo uma medida e conhecendo o primeiro e o segundo momento estatístico, média e variância, respectivamente, é possível determinar as propriedades dos pesos no método de mínimos quadrados. Havendo mais do que uma medida, é necessário conhecer a dependência mútua entre os ruídos, que é medida pela sua covariância.

Segundo Strang (1986) para ruídos independentes, o que é mais comum na prática, a covariância é nula. Os pesos  $\mathbf{W}$  são dados por  $w_i = 1/\sigma_i$ , o que significa que, quanto menor for a variância  $\sigma_i$ , mais confiáveis são as observações e maiores os pesos  $w_i$ .

<sup>1</sup> $E[g(x)] = \int_{-\infty}^{+\infty} g(x)p(x)dx$ , em que  $p(x)$  é a densidade de probabilidade associada à variável  $x$ .

Então, para observações independentes, tem-se:

$$\|\mathbf{W}(\mathbf{y} - \mathbf{H}\mathbf{x})\|^2 = \|\mathbf{W}\mathbf{e}\|^2 = \frac{e_1^2}{\sigma_1^2} + \dots + \frac{e_m^2}{\sigma_m^2} \quad (3.30)$$

Observa-se que  $\mathbf{W}$  e  $\mathbf{C} = \mathbf{W}^T\mathbf{W}$  são diagonais,  $\mathbf{C}$  contém números  $1/\sigma_i$ . No caso em que, as variâncias são iguais, tem-se o método de mínimos quadrados padrão (STRANG, 1986). A matriz  $\mathbf{C}$  torna o estimador por mínimos quadrados ponderados o “melhor estimador”. Então, pode-se definir uma matriz de covariância dos erros de observação (dados medidos)  $\mathbf{R}$ , tal que  $\mathbf{C} = \mathbf{R}^{-1}$ . Os elementos de  $\mathbf{R}$ , denotados por  $R_{ii}$  são determinados pelas variâncias  $E[e_i^2]$  e os elementos fora da diagonal principal determinados por  $E[e_i e_j]$ . Como os produtos  $e_{ii}^2$  e  $e_i e_j$  aparecem na matriz quadrada  $\mathbf{e}\mathbf{e}^T$ , pode-se abreviar a matriz de covariância  $\mathbf{R}$  como o valor esperado de  $\mathbf{e}\mathbf{e}^T$ , isto é,

$$\mathbf{R} = E[\mathbf{e}\mathbf{e}^T] \quad (3.31)$$

Procura-se um  $\mathbf{C}$  tal que  $E[e_i] = 0$ . Para isso, o estimador  $\hat{\mathbf{x}} = \mathbf{L}\mathbf{y}$ , que estima o verdadeiro, mas não conhecido valor de  $\mathbf{x}$  das medidas  $\mathbf{y}$ , deve ser linear e não tendencioso se o valor esperado do erro de estimativa for zero, isto é:

$$E[\mathbf{x} - \hat{\mathbf{x}}] = E[\mathbf{x} - \mathbf{L}\mathbf{y}] = E[\mathbf{x} - \mathbf{L}(\mathbf{H}\mathbf{x} + \mathbf{e})] = E[(\mathbf{I} - \mathbf{L}\mathbf{H})\mathbf{x}] \quad (3.32)$$

deduzindo-se que  $\mathbf{L}$  é não tendencioso se  $\mathbf{L}\mathbf{H} = \mathbf{I}$ . Assumindo a hipótese de Gauss de que a matriz  $\mathbf{C}$  deve ser o inverso da matriz de covariância  $\mathbf{R}$  e fazendo  $\mathbf{L} = \mathbf{L}_0$ , tem-se da Equação 3.29 que:

$$\mathbf{L}_0 = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1} \quad (3.33)$$

a qual supõe-se ser a escolha ótima, pois satisfaz  $\mathbf{L}_0\mathbf{H} = \mathbf{y}$ . O melhor estimador linear não tendencioso (BLUE - *Best linear unbiased estimation*) é o que satisfaz  $\mathbf{C} = \mathbf{R}^{-1}$ . O estimador  $\hat{\mathbf{x}}$  e a matriz ótima  $\mathbf{L}_0$  satisfazem:

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} = \mathbf{L}_0\mathbf{y} \quad (3.34)$$

Esta escolha minimiza o erro esperado na estimativa, medida pela matriz de covariância do erro de estimativa  $\mathbf{P}$ , dado por:

$$\mathbf{P} = E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] \quad (3.35)$$

Estimar  $\mathbf{x}$  ótimo é equivalente a obter  $\mathbf{P}$  mínimo, portanto, minimiza-se a Equação 3.35, que ao se substituir o valor de  $\hat{\mathbf{x}}$  tem-se

$$E[(\mathbf{x} - \mathbf{LHx} - \mathbf{Le})(\mathbf{x} - \mathbf{LHx} - \mathbf{Le})^T] \quad (3.36)$$

Desde que  $\mathbf{LH} = \mathbf{I}$  e  $\mathbf{L}$  seja linear, usando a Equação 3.31 tem-se:

$$\mathbf{P} = E[(\mathbf{Le})(\mathbf{Le})^T] = \mathbf{L}E[\mathbf{e}\mathbf{e}^t]\mathbf{L}^T = \mathbf{LRL}^T \quad (3.37)$$

Para mostrar que  $\mathbf{L}_0$  é uma escolha ótima, escreve-se  $\mathbf{L}$  como  $\mathbf{L}_0 + (\mathbf{L} - \mathbf{L}_0)$ , que substituindo em  $\mathbf{P} = \mathbf{LRL}^T$ , resulta em:

$$\mathbf{P} = \mathbf{L}_0\mathbf{RL}_0^T + (\mathbf{L} - \mathbf{L}_0)\mathbf{RL}_0^T + \mathbf{L}_0\mathbf{R}(\mathbf{L} - \mathbf{L}_0)^T + (\mathbf{L} - \mathbf{L}_0)\mathbf{R}(\mathbf{L} - \mathbf{L}_0)^T \quad (3.38)$$

Para os termos do meio da Equação 3.38, tem-se:

$$(\mathbf{L} - \mathbf{L}_0)\mathbf{RL}_0^T = (\mathbf{L} - \mathbf{L}_0)\mathbf{RR}^{-1}\mathbf{H}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1} = 0 \quad (3.39)$$

pois,  $\mathbf{RR}^{-1}$  é a matriz identidade e  $(\mathbf{L} - \mathbf{L}_0)\mathbf{H} = \mathbf{I} - \mathbf{I} = 0$ . Além disso, o último termo da Equação 3.38 é simétrico e, pelo menos, positiva semidefinida (STRANG, 1986). Assim, a Equação 3.38 pode ser escrita como:

$$\mathbf{P} = \mathbf{L}_0\mathbf{RL}_0^T + (\mathbf{L} - \mathbf{L}_0)\mathbf{R}(\mathbf{L} - \mathbf{L}_0)^T \quad (3.40)$$

Esta expressão é mínima para  $\mathbf{L} = \mathbf{L}_0$ , então a Equação 3.40 se reduz a  $\mathbf{P} = \mathbf{L}_0\mathbf{RL}_0^T$ . Substituindo-se  $\mathbf{L}_0$  por  $\mathbf{L}$  em  $\mathbf{P} = \mathbf{LRL}^T$  obtém-se a seguinte expressão para a covariância do erro de estimativa dado por:

$$\mathbf{P} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1} \quad (3.41)$$

A derivação da matriz  $\mathbf{P}$  foi desenvolvida de acordo com Strang (1986). Uma outra maneira alternativa de derivação da matriz de covariância dos erros de estimação pode ser encontrada em Mendel (1995). De posse das deduções das equações para o método de mínimos quadrados ponderados (MQP) apresenta-se a seção a seguir o método de mínimos quadrados recursivos.

### 3.3 Mínimos quadrados recursivos

Supõe-se que se estimou o vetor  $\mathbf{x}$  tão bem quanto possível, a partir de um primeiro conjunto de dados observados  $\mathbf{y}_0$ . Aqui, deduz-se a expressão de mínimos quadrados, para estimar um estado, ao passo que se dispõe de novas medidas, sem a necessidade de se refazer os cálculos. Um maneira de expressar a ideia de mínimos quadrados recursivos (MQR) é considerar que se pretende estimar  $x_1$  a partir da estimativa  $x_0$ , agregando a nova observação  $y_1$ . Para determinar  $x_1$ , faz-se a mesma suposição feita para determinar  $x_0$ , isto é, a matriz  $\mathbf{C}$  que torna  $\hat{x}_1$  a melhor estimativa é  $\mathbf{C} = \mathbf{R}^{-1}$ , onde

$$\mathbf{R} = \begin{bmatrix} R_0 & 0 \\ 0 & R_1 \end{bmatrix} \quad (3.42)$$

é a matriz de covariância do erro de observação. O vetor de erro é dado por:

$$\mathbf{e} = \begin{bmatrix} e_0 & 0 \\ 0 & e_1 \end{bmatrix} \quad (3.43)$$

A matriz  $\mathbf{R}$  é diagonal por bloco, pois  $e_1$  é independente de  $e_0$ . Desta forma, a matriz  $\mathbf{H}^T \mathbf{C} \mathbf{H}$ , no cálculo de  $x_1$  será:

$$\mathbf{P}^{-1} = \begin{bmatrix} H_0^T & H_1^T \end{bmatrix} \begin{bmatrix} R_0^{-1} & 0 \\ 0 & R_1^{-1} \end{bmatrix} \begin{bmatrix} H_0 & H_1 \end{bmatrix} = \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{H}_0 + \mathbf{H}_1^T \mathbf{R}_1^{-1} \mathbf{H}_1 \quad (3.44)$$

Lembra-se que  $\hat{x}_1$  não é baseado apenas em  $y_1$ , e é melhor estimador para o sistema combinado  $H_0 x = y_0$ ,  $H_1 x = y_1$ . Uma vez que a equação normal é  $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{x} = \mathbf{H}^T \mathbf{R}^{-1} \mathbf{y}$ , nota-se que  $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{x}$  é exatamente  $\mathbf{P}^{-1}$ . Assim, o  $x_1$  ótimo é dado por:

$$x_1 = \mathbf{P}_1 \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \mathbf{R}^{-1} \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \mathbf{P}_1 (\mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{y}_0 + \mathbf{H}_1^T \mathbf{R}_1^{-1} \mathbf{y}_1) \quad (3.45)$$

Esta é a solução que, espera-se encontrar recursivamente, usando o valor de  $\mathbf{x}_0$  já calculado no lugar de  $\mathbf{y}_0$  na [Equação 3.45](#). A dificuldade é que o termo  $\mathbf{y}_0$  é multiplicado por  $\mathbf{P}_1$ . Portanto, atualiza-se a matriz  $\mathbf{P}$  com a estimativa de  $\mathbf{x}$  usando a [Equação 3.44](#).

$$\mathbf{P}_1^{-1} = \mathbf{P}_0^{-1} + \mathbf{H}_1^T \mathbf{R}_1^{-1} \mathbf{H}_1 \quad (3.46)$$

Isto fornece uma melhoria na aproximação da segunda medida. Há uma diminuição de  $\mathbf{P}$ , que mede a incerteza. Note novamente que a [Equação 3.46](#) não depende de uma medida particular  $\mathbf{y}_0$  ou  $\mathbf{y}_1$ , usa-se apenas as suas propriedades estatísticas e a propriedade estatística de  $\mathbf{x}_1$ .

A atual estimativa de  $\mathbf{x}_1$  deve ser baseada sobre o atual valor de  $\mathbf{y}_0$  e  $\mathbf{y}_1$ , que é dado pela [Equação 3.44](#) e todo o tema de mínimos quadrados recursivos é reescrito pela fórmula a seguir:

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{P}_1(\mathbf{P}_0^{-1}\mathbf{x}_0 + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{y}_1) \\ \mathbf{x}_1 &= \mathbf{P}_1(\mathbf{P}_1^{-1}\mathbf{x}_0 - \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1\mathbf{x}_0 + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{y}_1) \\ \mathbf{x}_1 &= \mathbf{x}_0 + \mathbf{K}_1(\mathbf{y}_1 - \mathbf{H}_1\mathbf{x}_0)\end{aligned}\tag{3.47}$$

A matriz  $\mathbf{K}_1 = \mathbf{P}_1\mathbf{H}_1^T\mathbf{R}_1^{-1}$  é denominada de matriz de ganho. Com esta manipulação a fórmula torna-se recursiva, usa-se  $x_0$  no lugar de  $y_0$ .

Suponha que uma nova medida é exatamente consistente com o  $x_0$  original, isto é,  $\mathbf{y}_1 = \mathbf{H}_1\mathbf{x}_0$ . Então, não há razão para mudar a estimativa de  $\mathbf{x}$ . A melhor hipótese é ainda  $x_1 = x_0$ , sempre que os novos pontos  $\mathbf{y}_1$  são determinados pelos pontos  $\mathbf{y}_0$ . Calcula-se a previsão do erro  $\|\mathbf{y}_1 - \mathbf{H}\mathbf{x}_0\|$ , denominada de inovação. Generalizando a [Equação 3.48](#) obtém-se a equação por MQR para  $\hat{\mathbf{x}}$  dadas as medidas  $\mathbf{y}_i$ :

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{K}_i(\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_{i-1})\tag{3.48}$$

e a matriz do erro de estimação é atualizada de acordo com:

$$\mathbf{P}_i^{-1} = \mathbf{P}_{i-1}^{-1} + \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i\tag{3.49}$$

em que  $\mathbf{K}_i = \mathbf{P}_i\mathbf{H}_i^T\mathbf{R}_i^{-1}$ . A [Equação 3.48](#) é um modo eficiente de calcular a estimativa da variável de estado  $\mathbf{x}$ , no instante  $i$ , dadas as medidas  $\mathbf{y}_{i-1}$ . Nesta seção, derivou-se o algoritmo de mínimos quadrados recursivos como uma extensão natural do método de mínimos quadrados. Na seção, a seguir introduz-se o Filtro de Kalman.

### 3.4 Filtro de Kalman

Nesta seção, deriva-se o método de Filtro de Kalman (FK) para processos em tempo discreto. Este método é um dos avanços mais importantes na teoria de previsão e filtragem. Esta ferramenta tem uma ampla área de aplicação. O que há em comum em

todas as áreas de aplicação é que quando os sinais chegam, eles devem ser processados e há a necessidade de se remover o ruído contido neles. Quando esses sinais são todos medidos de uma mesma quantidade, tem-se que a solução ótima é dada pela recursão representada pelas Equações 3.48 e 3.49. No entanto, se a quantidade que está sendo estimada muda assim que novas medidas estão disponíveis, então o filtro terá que calcular essa alteração. Portanto, o problema de filtragem ótima torna-se dinâmico (STRANG, 1986).

Neste caso, para estimar um vetor não conhecido  $\mathbf{x}_i$  em cada passo  $i$  de tempo precisam-se de duas equações para:

- Dados medidos  $y_0, y_1, \dots, y_m$  tomados no tempo  $i = m$ . Se as observações são exatas, elas serão relacionadas ao valor real por  $\mathbf{y}_i = \mathbf{H}_i \mathbf{x}_i$ . Em geral as observações não são exatas e as matrizes  $\mathbf{H}_i$  são retangulares.
- Uma lei conhecida  $\mathbf{x}_{i+1} = \mathbf{F}_i \mathbf{x}_i$  que governa as alterações nas variáveis de estado avançando no tempo. As matrizes  $\mathbf{F}_i$  mudam em cada passo de tempo e poderão conter erros denotado por  $\varepsilon_i$

Assim, as duas equações são representadas por:

$$\begin{aligned} \mathbf{y}_i &= \mathbf{H}_i \mathbf{x}_i \text{ com erro } e_i \\ \mathbf{x}_{i+1} &= \mathbf{F}_i \mathbf{x}_{i+1} \text{ com erro } \varepsilon_i \end{aligned} \tag{3.50}$$

Deseja-se combinar estas equações em um único sistema. Por uma questão prática, consideram-se disponíveis as observações  $y_0, y_1$  e  $y_2$  para estimar  $x_2$ . O sistema 3.50 pode ser reescrito como:

$$\begin{aligned} \mathbf{H}_i \mathbf{x}_i &= \mathbf{y}_i \text{ com erro } e_i \\ -\mathbf{F}_i \mathbf{x}_{i+1} + \mathbf{x}_{i+1} &= 0 \text{ com erro } \varepsilon_i \end{aligned} \tag{3.51}$$

em que  $i = 0, 1, 2$ . Então, o sistema de equações pode ser representado por:

$$\begin{bmatrix} H_0 & & & \\ -F_0 & I & & \\ & H_1 & & \\ & -F_1 & I & \\ & & H_2 & \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ 0 \\ y_1 \\ 0 \\ y_2 \end{bmatrix}^T \quad (3.52)$$

O objetivo da filtragem é encontrar a melhor solução para o sistema 3.52. Segundo Gauss, é possível resolver este sistema por mínimos quadrados, com sua matriz de ponderação. De acordo com Kalman, este sistema é resolvido recursivamente. Se  $x_0$  e  $x_1$  já são valores conhecidos na Equação 3.52, apenas  $x_2$  deverá ser determinado. Mas, este não é o caso. A solução do sistema 3.52 é afetada pela medida  $y_2$ . Calcula-se a melhor estimativa para  $x_0$  e  $x_1$ , bem como para  $x_2$ , baseada em todas as informações disponíveis até o tempo  $t = 2$ . O aperfeiçoamento sobre as medidas anteriores é denominada de *smoothing*, e a estimação de  $x_2$  é chamada de filtragem, conforme ilustração da Figura 3.1. Escrevem-se estas estimativas como  $x_{0|2}$ ,  $x_{1|2}$  e  $x_{2|2}$ .

A maioria das aplicações concentra-se no novo valor de  $x_{2|2}$ , que prediz  $x_{3|2} = F_2 x_{2|2}$  no tempo seguinte. Então  $y_3$  corrigirá o valor predito para um valor filtrado, depois de um passo a mais. Este é apenas um caso não provável que  $y_3$  corresponderá exatamente a predição  $H_3 x_{3|2}$ . A inovação, que é a diferença entre os dois, será zero, pois  $x_{3|3}$  concordará com o valor predito  $x_{3|2}$ . O problema é uma extensão direta do mínimo quadrado recursivo, no qual continuamente estima-se o mesmo vetor  $\mathbf{x}$  (STRANG, 1986).

No algoritmo do Filtro de Kalman, não é razoável supor que  $\mathbf{x}_{i+1} = \mathbf{F}_i \mathbf{x}_i$  é exato, visto que o modelo é sempre imperfeito. Os erros  $\varepsilon_i$  e  $e_i$  não possuem o mesmo tamanho, pois não têm a mesma medida e a mesma unidade. Frequentemente os erros  $e_i$  são independentes e com variância  $\sigma$ . Os erros  $\varepsilon_i$  também são independentes com variância diferente de  $\sigma/c$ . Então a ponderação correta para as equações no sistema 3.52 é escrita a seguir. As linhas em  $\mathbf{H}$  e  $\mathbf{y}$  são divididas por  $\sigma$ , e as linhas em

$\mathbf{F}$  e  $\mathbf{I}$  são divididas por  $\frac{\sigma}{c}$ . Então o sistema 3.52 pode ser escrito como:

$$\begin{bmatrix} H_0 & & & & \\ -cF_0 & c & & & \\ & & H_1 & & \\ & & -cF_1 & cI & \\ & & & & H_2 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ 0 \\ y_1 \\ 0 \\ y_2 \end{bmatrix}^T \quad (3.53)$$

Este é o sistema  $\mathbf{H}\mathbf{x} = \mathbf{y}$  que resolve o filtro por mínimos quadrados. A aproximação recursiva é usar tanto quanto possível o passo anterior, resumindo o problema de filtragem por Kalman e sua solução retoma-se o sistema 3.50. A variável de estado do modelo é descrita por:

$$\mathbf{x}_{i+1} = \mathbf{F}_i\mathbf{x}_i + \varepsilon_i \quad (3.54)$$

e os dados observados são relacionados com a variável de estado por:

$$\mathbf{y}_i = \mathbf{H}\mathbf{x}_i + e_i \quad (3.55)$$

em que  $\varepsilon_i$  e  $\mathbf{e}_i$  representam ruídos brancos e independentes. O estado inicial  $\mathbf{x}_0$  tem valor médio  $\hat{x}_{0|-1}$  e a matriz de covariância  $\mathbf{P}_{0|-1}$  é independente da posição e do ruído de medida. O ruído tem média zero e o segundo momento estatístico (variância) é descrita por:

$$E[e_i e_i^T] = R_i \delta_i \quad (3.56)$$

$$E[\varepsilon_i \varepsilon_i^T] = P_i \delta_i \quad (3.57)$$

$$E[e_i \varepsilon_i^T] = 0 \quad (3.58)$$

em que  $\delta_i$  é o delta de Kronecker. Uma estimativa  $\hat{\mathbf{x}}_{i|i}$  do estado  $\mathbf{x}_i$  é calculada a partir dos dados observados  $y_0, y_1, \dots, y_m$ , de modo a minimizar o erro quadrático médio na estimativa definido pela Equação 3.5. A estimativa que satisfaz isto é calculada como uma função implícita apenas da medida  $y_i$  e a melhor estimativa anterior  $\mathbf{x}_{i-1|i-1}$ . Esta aproximação conduz à uma aproximação recursiva que fornece uma estimativa equivalente para a estimativa obtida pelo processamento de todas as observações simultaneamente (SORENSEN, 1970). A estimativa do sinal  $\mathbf{y}_i = \mathbf{H}_i\mathbf{x}_i$  é dada por  $\hat{\mathbf{y}}_{i|i} = \mathbf{H}_i\hat{\mathbf{x}}_{i|i}$ .

Uma solução possível para o problema de mínimos quadrados, linear e recursivo

pode ser feita por meio do princípio da ortogonalidade<sup>2</sup>. A solução proposta por Kalman é dada por uma combinação linear da estimativa predita na ausência de um novo dado. O Filtro de Kalman calcula o melhor estimador linear não tendencioso no tempo  $i$ . Além disso, o filtro é capaz de estimar a matriz de covariância do erro dadas as medidas.

O processo ocorre em duas fases que são: atualização e propagação. Na fase de atualização, determina-se a estimativa  $\hat{\mathbf{x}}$  no instante  $i$  dadas as medidas  $\mathbf{y}_i$ . Na propagação calcula-se a estimativa  $\hat{\mathbf{x}}$  no instante  $i + 1$ , dadas as medidas  $\mathbf{y}_i$ . A seguir apresentam-se os passos do algoritmo.

1. Previsão a partir do modelo e cálculo da matriz de covariância da previsão

$$\mathbf{x}_{i+1}^p = \mathbf{F}_{i+1}\mathbf{x}_i^a \quad (3.59)$$

$$\mathbf{P}_{i+1}^p = \mathbf{F}_i\mathbf{P}_i^a\mathbf{F}_i^T + \mathbf{Q}_i \quad (3.60)$$

2. Cálculo do ganho de Kalman

$$\mathbf{K}_{i+1} = \mathbf{P}_{i+1}^p\mathbf{H}_{i+1}^T[\mathbf{R}_{i+1} + \mathbf{H}_{i+1}\mathbf{P}_{i+1}^p\mathbf{H}_{i+1}^T]^{-1} \quad (3.61)$$

3. Cálculo da estimativa

$$\mathbf{y}_{i+1}^p = \mathbf{H}_{i+1}\mathbf{x}_{i+1}^p \quad (3.62)$$

$$\mathbf{x}_{i+1}^a = \mathbf{x}_{i+1}^p + \mathbf{K}_{i+1}(\mathbf{y}_{i+1} - \mathbf{y}_{i+1}^p) \quad (3.63)$$

4. Cálculo da matriz de covariância da análise

$$\mathbf{P}_{i+1}^a = [\mathbf{I} - \mathbf{K}_{i+1}\mathbf{H}_{i+1}]\mathbf{P}_{i+1}^p \quad (3.64)$$

O sobreíndice  $p$ , referente à fase de propagação, corresponde a previsão;  $a$ , referente a fase de atualização, refere-se ao dado de análise no processo de assimilação de dados.

Na aplicação do algoritmo do Filtro de Kalman, assume-se que os ruídos e os resíduos  $(\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_i)$  são independentes. A distribuição de densidade de probabilidade do ruído

---

<sup>2</sup>Estabelece que o erro é mínimo se este for ortogonal aos  $x_i$  (KAY, 1993).

e das variáveis de estado inicial do modelo possuem distribuição gaussiana. Em cada passo de tempo as variáveis de estado e as observações manterão a distribuição gaussiana devido a linearidade do sistema.

O Algoritmo do Filtro de Kalman no contexto de assimilação de dados é mostrado no diagrama esquemático apresentado na [Figura 3.3](#).

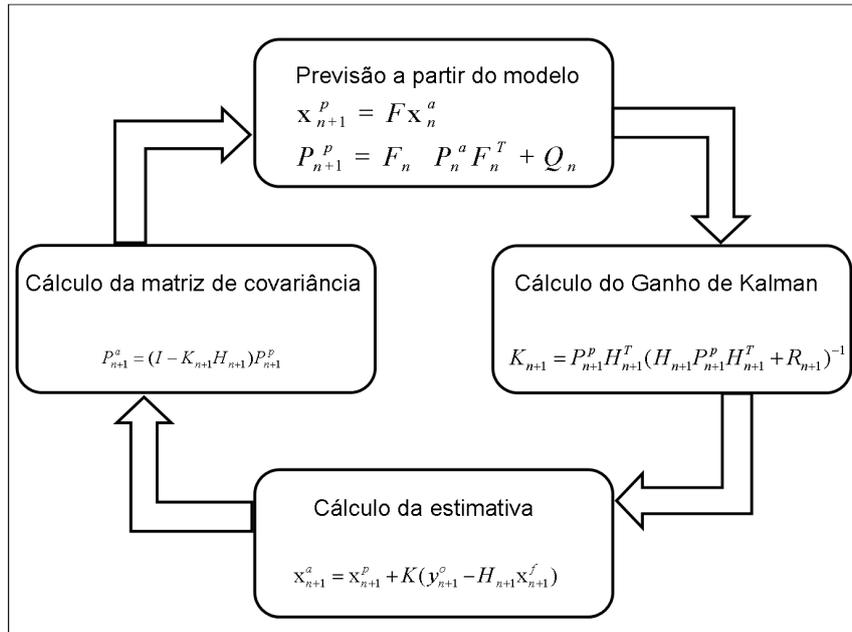


Figura 3.3 - Diagrama esquemático do Filtro de Kalman Linear.

Fonte: Adaptado de [Todling \(1999\)](#)

Segundo [Polavarapu \(2004\)](#) o Filtro de Kalman possui as seguintes propriedades.

1. O FK é um sistema dimensional finito linear em tempo discreto.
2. A entrada do filtro é o vetor  $\mathbf{y}_i$  e a saída é o vetor  $\hat{\mathbf{x}}_i$
3. As matrizes de covariância representas pelas equações 3.60 e 3.64 são independentes das observações e, portanto, as covariâncias são incondicionais. As matrizes de covariância representam as incertezas do modelo e das observações. Segundo [Bouttier e Courtier \(2002\)](#) um modo eficiente de se obter as matrizes de erro do modelo é realizar muitas execuções do modelo

com diferentes realizações para se obter as estatísticas.

4. A função densidade de probabilidade é completamente definida pela média e covariância. Assim, o FK descreve um procedimento para a atualização da  $fdp$  de  $\hat{\mathbf{x}}_i$ .
5. Na definição do ganho de Kalman, definida na Equação 3.61, é exigido o cálculo da inversa  $\mathbf{H}_i\mathbf{P}_i^p\mathbf{H}_i^T + \mathbf{R}_i$ . Uma vez que  $\mathbf{P}_i^p$  e  $\mathbf{R}_i$  são matrizes de covariância, elas devem ser semi-definidas positivas.
6. Se o modelo numérico de previsão é invariante no tempo e o ruído de entrada e de saída são estacionários,  $\mathbf{F}_i, \mathbf{H}_i, \mathbf{Q}_i, \mathbf{R}_i$  são constante. Então, o FK será em geral variante no tempo.

O FK é planejado para sistemas lineares. No entanto, na realidade lidam-se com problemas não apenas não lineares, mas com problemas de alta dimensão e complexo. Então esta ferramenta não é prática. Para o caso de problemas não lineares, a matriz  $\mathbf{F}$ , que é uma função das variáveis de estado do modelo, é linearizada. Tem-se então a versão do Filtro de Kalman Estendido (FKE), que é uma versão mais utilizada do Filtro (JAZWINSKI, 1970; KALNAY, 2003; KALMAN; BUCY, 1961). No trabalho de Brasseur e Verron (2006), apresenta-se o filtro SEEK (*Singular Evolutionary Extended Kalman*), que é usado na solução de problemas de assimilação de dados em oceanografia.

O problema principal consiste na evolução da matriz de covariância. Se o estado é de dimensão  $10^7$ , então a matriz de covariância tem  $10^{14}$  elementos e propagar essa matriz no tempo torna-se inviável. Estudos tem sido desenvolvido para reduzir o custo de computação dessa matriz (TANGBORN, 2004; WEI-MIN et al., 2010; BROQUET et al., 2008; KEPERT, 2011).

Os Filtros de Kalman baseados em conjunto, originalmente proposto por Evensen (1994), foi desenvolvido para possibilitar o cálculo da matriz de covariância do modelo em sistemas operacionais de previsão. Os métodos baseados em conjunto têm se tornado muito populares para assimilação de dados em modelos numéricos de fluidos oceânicos ou atmosféricos (BRASSEUR, 2011; SONG et al., 2012).

Os métodos de Filtro de Kalman enquadram-se na categoria de filtragem. O método descrito no Capítulo 4 a seguir classifica-se na categoria *smoothing* (JAZWINSKI, 1970; HAYKIN; KAILATH, 2002; CHUA; BENNETT, 2001; BENNETT, 1992).



## 4 MÉTODO VARIACIONAL: TÉCNICA DO REPRESENTANTE

O método variacional é uma técnica de assimilação de dados elegante baseada na minimização de funcionais. Porém, este desenvolvimento é uma formulação dependente do problema, ou seja, para diferentes problemas, ter-se-ão diferentes formulações (em alguns casos não é óbvio que se pode obter a formulação variacional do problema). De modo geral, para a formulação variacional do representante seguem-se os seguintes passos:

- Descrição do funcional de penalidade: composição ponderada entre a diferença quadrática existente entre dados de previsão e dados de observação;
- Incorporação de resíduos (restrições);
- Desenvolvimento das equações de Euler-Lagrange;
- Equação adjunta (multiplicadores de Lagrange);
- Cálculo do representante e da análise;

Neste Capítulo, descreve-se a metodologia variacional para os modelos apresentados no [Capítulo 2](#).

### 4.1 Método variacional: equação da onda 1D

Nesta seção, descreve-se o método variacional para o modelo descrito na [Seção 2.1](#) no [Capítulo 2](#).

A inclusão de observações no modelo representado pela [Equação 2.1](#), que são medidas imperfeitas da variável  $\eta(x, t)$  em pontos isolados no tempo e no espaço torna o problema sobredeterminado, ou seja, não existem soluções suaves que satisfaçam ao modelo e às observações simultaneamente ([BENNETT, 2004](#)). Portanto, o problema consiste em determinar a solução por meio de uma construção ponderada, realizada pelo ajuste de mínimos quadrados entre as observações e o modelo.

#### 4.1.1 Funcional de Penalidade

Para a construção do funcional de penalidade considera-se:  $F(x, t)$  a forçante,  $I(x)$  a condição inicial e a condição de contorno periódica. Para qualquer escolha de

$F(x, t) + f(x, t)$  e  $I(x) + i(x)$ , existe uma única solução real  $\eta(x, t)$  para a equação da onda linear (BENNETT, 2002). No entanto, tem-se apenas  $M$  valores de dados  $d_m$  para nos guiar e, então, os campos de erro ( $f(x, t)$ ,  $i(x)$  e  $\varepsilon_m$ ) não são conhecidos. O objetivo é procurar uma solução  $\hat{\eta}(x, t)$  que corresponda ao menor  $f(x, t)$ ,  $i(x)$  e  $\varepsilon_m$  em um sentido de mínimos quadrados ponderados. Assim, procura-se determinar o mínimo do funcional de penalidade quadrático a seguir:

$$J = J[\eta(x, t)] = W_f \int_0^T \int_0^L \{f(x, t)\}^2 dt dx + W_i \int_0^L \{i(x)\}^2 dx + w \sum_{m=1}^M \{\varepsilon_m\}^2 \quad (4.1)$$

em que  $W_f$ ,  $W_i$  e  $w$  são pesos positivos e constantes, escolhidos livremente. Estes pesos são operadores inversos de covariâncias *a priori* dos erros da dinâmica (modelo), da condição inicial e das observações, respectivamente (CHUA; BENNETT, 2001). Note a dependência de  $J$  sobre  $\eta$ , sendo que,  $\eta$  é um campo de valores para  $0 \leq x \leq L$  e  $0 \leq t \leq T$ , o funcional de penalidade  $J(\eta)$  é um único número para cada escolha de entrada do campo  $\eta$ . O lado direito da Equação 4.1 envolve apenas  $f, i$  e  $\varepsilon_m$ .

Reescrevendo a Equação 4.1 em função de  $f(x, t)$ ,  $i(x)$  e  $\varepsilon_m$  obtêm-se a dependência explícita sobre a variável  $\eta(x, t)$ , de acordo com a equação a seguir:

$$J = J[\eta(x, t)] \equiv W_f \int_0^T \int_0^L \left\{ \frac{\partial \eta}{\partial t} + c \frac{\partial \eta}{\partial x} - F(x, t) \right\}^2 dt dx + W_i \int_0^L \{\eta(x, 0) - I(x)\}^2 dx + w \sum_{m=1}^M \{\eta(x_m, t_m) - d_m\}^2 \quad (4.2)$$

O cálculo do funcional 4.2 é feito por meio do cálculo das variações. Define-se  $\hat{\eta}(x, t)$  como um extremo local, isto é, corresponde ao menor valor da função custo para o menor  $f(x, t)$ ,  $i(x)$  e  $\varepsilon_m$ . Ressalta-se que como o funcional de penalidade é quadrático, o extremo local é um extremo global e o funcional é não negativo, então o seu extremo é um mínimo global. Veja ilustração no gráfico 4.1.

Agora consideremos uma expansão em série de Taylor de  $J$  em torno do ponto  $\hat{\eta}$ , isto é:

$$J[\hat{\eta} + \delta\eta] = J[\hat{\eta}] + \frac{\partial}{\partial \hat{\eta}} J(\hat{\eta}) \delta\eta + \frac{1}{2!} \frac{\partial^2}{\partial^2 \hat{\eta}} J(\hat{\eta}) (\delta\eta)^2 + \dots \quad (4.3)$$

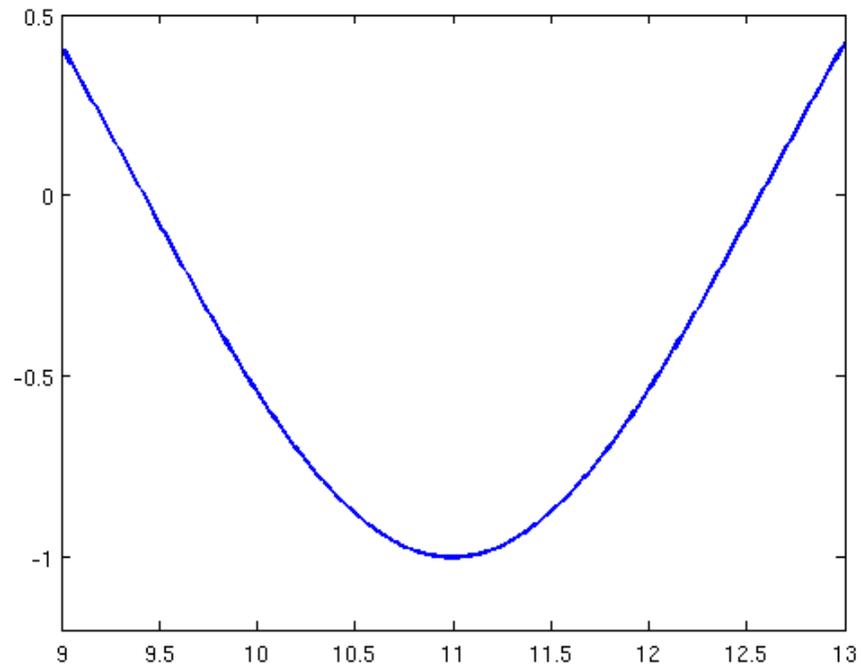


Figura 4.1 - Representação gráfica de um funcional de penalidade quadrático não negativo.

O funcional acima pode ser escrito como:

$$J[\hat{\eta} + \delta\eta] = J[\hat{\eta}] + \nabla J(\eta)\delta\eta + O(\delta\eta)^2 \quad (4.4)$$

Assim,

$$J[\hat{\eta} + \delta\eta] - J[\hat{\eta}] \cong \nabla J(\eta)\delta\eta \quad (4.5)$$

para uma pequena variação  $\delta\eta = \delta\eta(x, t)$ , em que  $\delta\eta = \nabla J(\eta)$ . Então uma primeira variação do funcional  $J$  pode ser escrita como:

$$\delta J = J[\hat{\eta} + \delta\eta] - J[\hat{\eta}] \quad (4.6)$$

De acordo com a definição do funcional 4.2 pode-se escrever  $J[\hat{\eta}]$  como:

$$J[\hat{\eta}] = W_f \int_0^T \int_0^L \left\{ \frac{\partial \hat{\eta}}{\partial t} + c \frac{\partial \hat{\eta}}{\partial x} - F(x, t) \right\}^2 dt dx + W_i \int_0^L \{\hat{\eta}(x, 0) - I(x)\}^2 dx + w \sum_{m=1}^M \{\hat{\eta}(x_m, t_m) - d_m\}^2 \quad (4.7)$$

e para  $J[\hat{\eta} + \delta\eta]$  tem-se:

$$J[\hat{\eta} + \delta\eta] = W_f \int_0^T \int_0^L \left\{ \frac{\partial \hat{\eta}}{\partial t} + \frac{\partial \delta\eta}{\partial t} + c \frac{\partial \hat{\eta}}{\partial x} + c \frac{\partial \delta\eta}{\partial x} - F \right\}^2 dt dx + W_i \int_0^L \{\hat{\eta}(x, 0) + \delta\eta(x, 0) - I(x)\}^2 dx + w \sum_{m=1}^M \{\hat{\eta}(x_m, t_m) + \delta\eta(x_m, t_m) - d_m\}^2 \quad (4.8)$$

Subtraindo as equações 4.7 de 4.8 e desprezando a variações de segunda ordem tem-se:

$$\delta J = 2W_f \int_0^T dt \int_0^L dx \left\{ \frac{\partial \hat{\eta}}{\partial t} c \frac{\partial \delta\eta}{\partial x} - F(x, t) \right\} \left\{ \frac{\partial \delta\eta}{\partial t} + c \frac{\partial \delta\eta}{\partial x} \right\} + 2W_i \int_0^L dx \{\hat{\eta}(x, 0) - I(x)\} \delta\eta(x, 0) + 2w \sum_{m=1}^M \{\hat{\eta}(x_m, t_m) - d_m\} \delta\eta(x_m, t_m) + O(\delta\eta)^2 \quad (4.9)$$

Portanto, tem-se a primeira variação para o funcional de penalidade 4.1 dado pela Equação 4.9. Uma vez definida a variação do funcional, o próximo passo é determinar a equação de Euler-Lagrange associada, o que é feito na próxima seção.

#### 4.1.2 Equação de Euler-Lagrange

Considera-se a primeira variação dada pela Equação 4.9, define-se o residual ponderado  $\lambda$  como:

$$\lambda(x, t) = W_f \left\{ \frac{\partial \hat{\eta}}{\partial t} + c \frac{\partial \hat{\eta}}{\partial x} - F(x, t) \right\} \quad (4.10)$$

Usando a definição do residual 4.10 e dividindo Equação 4.9 por 2 e distribuindo-se o primeiro integrando, tem-se:

$$\begin{aligned} \frac{\delta J}{2} = & \int_0^T dt \int_0^L dx \lambda \left\{ \frac{\partial \delta \eta}{\partial t} \right\} + \int_0^T dt \int_0^L dx \lambda \left\{ c \frac{\partial \delta \eta}{\partial x} \right\} + \\ & W_i \int_0^L dx \{ \hat{\eta}(x, 0) - I(x) \} \delta \eta(x, 0) + \\ & w \sum_{m=1}^M \{ \hat{\eta}(x_m, t_m) - d_m \} \delta \eta(x_m, t_m) + O(\delta \eta)^2 \quad (4.11) \end{aligned}$$

Se  $\hat{\eta}(x, t)$  é um extremo local de  $J$ , então  $\delta J = O(\delta \eta)^2$ . Assim, a soma dos termos restantes deve ser zero e procura-se encontrar o campo  $\hat{\eta}(x, t)$  que força  $\delta J$  a  $O(\delta \eta)^2$  indiferentemente do campo  $\delta \eta(x, t)$ . Então, se expressam todas as variações de  $\eta(x, t)$  inteiramente em termos de  $\delta \eta(x, t)$ , isto é, não aparecerá  $\frac{\partial \delta \eta}{\partial t}$ ,  $\frac{\partial \delta \eta}{\partial x}$  ou  $\delta \eta(x_m, t_m)$ .

Para eliminar  $\frac{\partial \delta \eta}{\partial t}$ , e  $\frac{\partial \delta \eta}{\partial x}$  de 4.11, usou-se integração por partes para deslocar a derivada sobre a variação do residual ponderado. Considerando o termo da derivada temporal, tem-se:

$$\begin{aligned} \int_0^L dx \int_0^T dt \lambda \left\{ c \frac{\partial \delta \eta}{\partial t} \right\} &= \\ \int_0^L dx \left\{ [\lambda(x, T) \delta \eta(x, T) + (-\lambda(x, 0) \delta \eta(x, 0))] + \int_0^T -\frac{\partial \lambda}{\partial t} \delta \eta dt \right\} &= \\ \int_0^T dt \int_0^L dx \left\{ -\frac{\partial \lambda}{\partial t} \delta \eta \right\} + \int_0^L dx \{ \lambda(x, T) \delta \eta(x, T) \} + \int_0^L dx \{ -\lambda(x, 0) \delta \eta(x, 0) \} & \quad (4.12) \end{aligned}$$

Do mesmo modo para a derivada espacial, tem-se:

$$\begin{aligned} \int_0^T dt \int_0^L dx \lambda \left\{ c \frac{\partial \delta \eta}{\partial x} \right\} &= \int_0^T dt \int_0^L dx \left\{ -c \frac{\partial \lambda}{\partial x} \delta \eta \right\} \\ &+ \int_0^T dt \{ c \lambda(L, t) \delta \eta(L, t) \} \\ &+ \int_0^T dt \{ -c \lambda(0, t) \delta \eta(0, t) \} \quad (4.13) \end{aligned}$$

Note que, os termos de contorno são iguais e opostos, pois a condição de contorno é

periódica. Assim, a Equação 4.13 pode ser escrita como:

$$\int_0^T dt \int_0^L dx \lambda \left\{ c \frac{\partial \delta \eta}{\partial x} \right\} = \int_0^T dt \int_0^L dx \left\{ -c \frac{\partial \lambda}{\partial x} \delta \eta \right\} \quad (4.14)$$

Usando a propriedade da função Delta de Dirac<sup>1</sup> para eliminar  $\delta(x_m, t_m)$  em 4.11 em favor de  $\delta \eta(x, t)$ , tem-se:

$$w \sum_{m=1}^M \{ \hat{\eta}(x_m, t_m) - d_m \} \delta \eta(x_m, t_m) = \int_0^T dt \int_0^L dx \sum_{m=1}^M \{ \hat{\eta}(x_m, t_m) - d_m \} \delta \eta(x, t) \delta(x - x_m) \delta(t - t_m) \quad (4.16)$$

em que o segundo e o terceiro  $\delta$  denota a função Delta de Dirac. Substituindo as equações 4.12, 4.14 e 4.16 em 4.11, obtém-se:

$$\begin{aligned} 0 = & \int_0^T dt \int_0^L dx \left\{ -\frac{\partial \lambda}{\partial t} \delta \eta \right\} + \int_0^L dx \{ \lambda(x, T) \delta \eta(x, T) \} \\ & + \int_0^L dx \{ -\lambda(x, 0) \delta \eta(x, 0) \} + \int_0^T dt \int_0^L dx \left\{ -c \frac{\partial \lambda}{\partial x} \delta \eta \right\} \\ & + W_i \int_0^L dx \{ \hat{\eta}(x, 0) - I(x) \} \delta \eta(x, 0) \\ & + \int_0^T dt \int_0^L dx w \sum_{m=1}^M \{ \hat{\eta}(x_m, t_m) - d_m \} \delta \eta(x, t) \delta(x - x_m) \delta(t - t_m) \quad (4.17) \end{aligned}$$

Rearranjando os termos da Equação 4.17, tem-se:

$$\begin{aligned} 0 = & \int_0^T dt \int_0^L dx \left\{ -\frac{\partial \lambda}{\partial t} - c \frac{\partial \lambda}{\partial x} + w \sum_{m=1}^M \{ \hat{\eta}(x_m, t_m) - d_m \} \delta(x - x_m) \delta(t - t_m) \right\} \delta \eta \\ & + \int_0^L dx \{ \lambda(x, T) \} \delta \eta(x, T) + \int_0^L dx \{ -\lambda(x, 0) + W_i (\hat{\eta}(x, 0) - I(x)) \} \delta \eta(x, 0) \quad (4.18) \end{aligned}$$

Então, o valor de  $\eta(x, t)$  que origina os coeficientes de  $\delta \eta$ ,  $\delta \eta(x, T)$  e  $\delta \eta(x, 0)$  para desaparecer  $\hat{\eta}(x, t)$  é um extremo de  $J$ . Definindo-se esses coeficientes iguais a zero,

<sup>1</sup>A função Delta de Dirac é dada por:

$$F(a) = \int_{-\infty}^{+\infty} F(x) \delta(x - a) dx, \text{ onde } \delta(x - a) = \begin{cases} 0 & \text{se } x \neq a, \\ \infty & \text{se } x = a. \end{cases} \quad (4.15)$$

tem-se:

$$-\frac{\partial \lambda}{\partial t} - c \frac{\partial \lambda}{\partial x} + w \sum_{m=1}^M \{\hat{\eta}(x_m, t_m) - d_m\} \delta(x - x_m) \delta(t - t_m) = 0 \quad (4.19)$$

com  $0 \leq x \leq L$  e  $0 \leq t \leq T$

$$\lambda(x, T) = 0 \quad (4.20)$$

$$-\lambda(x, 0) + W_i \{\hat{\eta}(x, 0) - I(x)\} = 0 \quad (4.21)$$

e finalmente, o residual ponderado já definido:

$$\lambda(x, t) = W_f \left\{ \frac{\partial \hat{\eta}}{\partial t} + c \frac{\partial \hat{\eta}}{\partial x} - F(x, t) \right\} \quad (4.22)$$

Em geral, as Equações 4.19 e 4.22 constituem a equação de Euler-Lagrange para um extremo local do funcional de penalidade. No [Apêndice A](#), apresenta-se um modelo mais simples para a derivação das equações de Euler-Lagrange.

Rearranjando, as equações de Euler-Lagrange 4.19 para um extremo local  $\hat{\eta}$  do funcional de penalidade,  $J[\eta]$  pode ser escrito como um problema “*backward*” também denominado de *Equação Adjunta*:

$$-\frac{\partial \lambda}{\partial t} - c \frac{\partial \lambda}{\partial x} = -w \sum_{m=1}^M \{\hat{\eta}(x_m, t_m) - d_m\} \delta(x - x_m) \delta(t - t_m) \quad (4.23a)$$

$$\lambda(x, T) = 0 \quad (4.23b)$$

$$\lambda(0, t) = \lambda(L, t) \quad (4.23c)$$

O problema “*forwards*” dado por:

$$\frac{\partial \hat{\eta}}{\partial t} + c \frac{\partial \hat{\eta}}{\partial x} = F(x, t) + \frac{\lambda(x, t)}{W_f} \quad (4.24a)$$

$$\hat{\eta}(x, 0) = I(x) + \frac{\lambda(x, 0)}{W_i} \quad (4.24b)$$

$$\hat{\eta}(0, t) = \hat{\eta}(L, t) \quad (4.24c)$$

Das equações em 4.24a–4.24c têm-se que a melhor estimativa para  $f$  e  $i$  é dado por:

$$\hat{f}(x, t) \equiv W_f^{-1}\lambda(x, t) \text{ e } \hat{i}(x, 0) \equiv W_i^{-1}\lambda(x, 0) \quad (4.25)$$

De posse das equações adjuntas 4.23a e da Equação 4.24a para modelo avançado, introduz-se o método *representante*, o qual é usado para resolver as equações de Euler-Lagrange (EL).

### 4.1.3 Método *Representante*

Deseja-se desacoplar as equações de Euler-Lagrange (EL), então permitem-se soluções sequenciais do problema *adjunto* ou para trás (*backward*) e do problema para frente (*forward*). O denominado representante-adjunto e funções *representantes* facilitam este desacoplamento. Existem  $M$  funções representantes, uma para cada escalar de dados (observações), e estas são denotadas por  $r_m(x, t)$  com  $1 \leq m \leq M$ . A  $m$ -ésima função representante tem um representante-adjunto satisfazendo a equação adjunta que forçado apenas com um único impulso localizado na posição espacial e temporal do  $m$ -ésimo dado escalar:

$$-\frac{\partial \alpha_m}{\partial t} - c \frac{\partial \alpha_m}{\partial x} = \delta(x - x_m)\delta(t - t_m) \text{ com } 0 \leq x \leq L, 0 \leq t \leq T \quad (4.26a)$$

$$\alpha_m(x, T) = 0 \quad (4.26b)$$

$$\alpha_m(0, t) = \alpha_m(L, t) \quad (4.26c)$$

Observa-se a mudança da variável  $\lambda_m(x, T)$  para  $\alpha_m(x, T)$ . Isso é para indicar a diferença entre a forçante e também para indicar a dependência dos *representantes-adjunto* sobre a localização espaço tempo do  $m$ -ésimo dado escalar. Desde que há impulso, a Equação 4.26a pode ser integrada para trás com a condição final 4.26b produzindo o *representante-adjunto*  $\alpha_m(x, t)$ .

O representante satisfaz a equação *forward*, isto é, a Equação 4.24a, exceto pela substituição do campo adjunto sobre o lado direito da Equação 4.26a com o campo *representante-adjunto* e com a estimativa *a priori* da forçante ou a condição inicial.

$$\frac{\partial r_m}{\partial t} + c \frac{\partial r_m}{\partial x} = \frac{\alpha_m(x, t)}{W_f}, \quad 0 \leq x \leq L, 0 \leq t \leq T \quad (4.27a)$$

$$r_m(x, 0) = \frac{\alpha_m(x, 0)}{W_i}, \quad 0 \leq x \leq L \quad (4.27b)$$

$$r_m(x, 0) = r_m(L, T), \quad 0 \leq t \leq T \quad (4.27c)$$

A solução ótima é assumida como a soma da estimativa *a priori* e uma combinação linear dos *representantes*, matematicamente expressado por:

$$\hat{\eta}(x, t) = \eta_F(x, t) + \sum_{m=1}^M \beta_m r_m(x, t) \quad (4.28)$$

em que  $\beta_m$  são coeficientes não conhecidos. Observa-se que, a resposta da solução ótima para a forçante  $F(x, t)$  e a condição inicial  $I(x)$  é efetuada pela solução *forward*  $\eta_F(x, t)$ , enquanto que os termos no somatório do *representante* são uma sequência de “correções” para cada dado de observação.

#### 4.1.4 Determinando o coeficiente do *representante*

O problema agora consiste em determinar os valores de  $M$  coeficientes  $\beta_m$  que satisfazem a solução explícita da Equação 4.28. Por uma proposta de notação, define-se o operador diferencial conforme a seguir:

$$D = \frac{\partial}{\partial t} + c \frac{\partial}{\partial x} \quad (4.29)$$

A equação adjunta 4.23a com condição final 4.23b pode ser escrita como:

$$-D[\lambda] = -w \sum_{m=1}^M \{\hat{\eta}(x_m, t_m) - d_m\} \delta(x - x_m) \delta(t - tm) \quad (4.30a)$$

$$\lambda(x, T) = 0 \quad (4.30b)$$

A Equação *forward* 4.24a e sua condição inicial 4.24b escrita como:

$$D[\hat{\eta}] = F(x, t) + \frac{\lambda(x, t)}{W_f} \quad (4.31a)$$

$$\hat{\eta}(x, 0) = I(x) + \frac{\lambda(x, 0)}{W_i} \quad (4.31b)$$

Substituindo a expressão para a solução ótima 4.28 em 4.31a tem-se:

$$D[\eta_F(x, t) + \sum_{m=1}^M \beta_m r_m(x, t)] = F(x, t) + \frac{\lambda(x, t)}{W_f} \quad (4.32)$$

Distribuindo o operador  $D$ , tem-se:

$$D[\eta_F(x, t)] + \sum_{m=1}^M \beta_m D[r_m(x, t)] = F(x, t) + \frac{\lambda(x, t)}{W_f} \quad (4.33)$$

Reconhecendo de 2.1, o problema *forward* que:

$$D[\eta_F(x, t)] = F(x, t) \quad (4.34)$$

Identificando da Equação 4.27a na Equação do *representante* e adjunta que:

$$D[r_m] = \frac{\alpha_m(x, t)}{W_f} \quad (4.35)$$

Substituindo as equações 4.34 e 4.35 em 4.33, tem-se:

$$F(x, t) + \frac{1}{W_f} \sum_{m=1}^M \beta_m \alpha_m(x, t) = F(x, t) + \frac{\lambda(x, t)}{W_f} \quad (4.36)$$

Subtraindo  $F(x, t)$  e multiplicando por  $W_f$  de ambos os lados da igualdade, tem-se:

$$\sum_{m=1}^M \beta_m \alpha_m(x, t) = \lambda(x, t) \quad (4.37)$$

Aplicando o operador diferencial na Equação 4.37 tem-se:

$$-D[\lambda(x, t)] = -\sum_{m=1}^M \beta_m D[\alpha_m(x, t)] \quad (4.38)$$

Identificando em 4.26a a Equação *representante-adjunto* e *representante* que:

$$-D[\alpha_m] = \delta(x - x_m)\delta(t - t_m) \quad (4.39)$$

Substituindo 4.39 em 4.38, tem-se:

$$-D[\lambda(x, t)] = \sum_{m=1}^M \beta_m \delta(x - x_m)\delta(t - t_m) \quad (4.40)$$

Da Equação 4.30a e 4.40, tem-se que:

$$-w \sum_{m=1}^M \{\hat{\eta}(x_m, t_m) - d_m\} \delta(x - x_m) \delta(t - t_m) = \sum_{m=1}^M \beta_m \delta(x - x_m) \delta(t - t_m) \quad (4.41)$$

Equacionando os coeficientes dos impulsos produz uma equação para a determinação dos coeficientes  $\beta_m$ , dada a seguir:

$$\beta_m = -w \{\hat{\eta}(x_m, t_m) - d_m\} \quad (4.42)$$

Tem-se, portanto, que a expressão 4.42 para a determinação do coeficiente  $\beta$ . No entanto, o valor de  $\hat{\eta}$  não é conhecido. Na próxima seção determina-se  $\hat{\eta}$ .

#### 4.1.5 Determinando uma solução explícita para a solução ótima

Uma solução explícita para  $\hat{\eta}(x_m, t_m)$  é obtida pela avaliação da expressão de  $\hat{\eta}(x_m, t_m)$  como uma soma da estimativa *a priori* e uma combinação linear dos *representantes* em  $(x_m, t_m)$ , usando um índice diferente sobre o somatório:

$$\hat{\eta}(x_m, t_m) = \eta_F(x_m, t_m) + \sum_{l=1}^M \beta_l r_l(x_m, t_m) \quad (4.43)$$

Por simplicidade de notação considere:  $\hat{\eta}(x_m, t_m) = \hat{\eta}_m$ ,  $\eta_F(x_m, t_m) = \eta_{Fm}$  e  $r_l(x_m, t_m) = r_{lm}$ .

A Equação 4.43 pode ser escrita como:

$$\hat{\eta}_m = \eta_{Fm} + \sum_{l=1}^M \beta_l r_{lm} \quad (4.44)$$

Substituindo 4.44 em 4.42 para  $\beta_m$ , tem-se:

$$\beta_m = -w \left\{ \eta_{Fm} + \sum_{l=1}^M \beta_l r_{lm} - d_m \right\} \quad (4.45)$$

Definindo  $h_m = d_m - \eta_{Fm}$ . Então a Equação 4.45 pode ser escrita como:

$$\beta_m = -w \left\{ \sum_{l=1}^M \beta_l r_{lm} - h_m \right\} \quad (4.46)$$

Dividindo ambos os lados de 4.46 por  $-w$  tem-se:

$$-w^{-1}\beta_m = \sum_{l=1}^M \beta_l r_{lm} - h_m \quad (4.47)$$

Portanto,

$$h_m = w^{-1}\beta_m + \sum_{l=1}^M \beta_l r_{lm} \quad (4.48)$$

Finalmente, introduzindo uma delta de Kronecker  $\delta_{lm}$  para colocar  $w^{-1}\beta_m$  dentro do somatório, tem-se:

$$h_m = \sum_{l=1}^M (w^{-1}\beta_l \delta_{lm} + \beta_l r_{lm}) \quad (4.49)$$

Em notação matricial, as  $M$  equações para o  $M$  coeficiente do *representante* tornam-se:

$$h = (\mathbf{R} + w^{-1}\mathbf{I})\beta \quad (4.50)$$

Então, a solução explícita para  $\hat{\eta}(x, t)$  é dada por:

$$\hat{\eta}(x, t) = \eta_F(x, t) + (\mathbf{d} - \eta_F)(\mathbf{R} + w^{-1})^{-1}r(x, t) \quad (4.51)$$

De posse de todas as equações necessárias para implementação do algoritmo para o representante para o modelo 1D, mostra-se na [Tabela 4.1](#) o algoritmo para o método de assimilação com o modelo de onda em uma dimensão.

Tabela 4.1 - Algoritmo *representante* para o modelo de onda linear.

---

**Algoritmo: *Representante* em uma dimensão**

---

1. Calcule  $\eta_F(x, t)$  pela integração numérica da Equação 2.1
  2. Calcule a inovação  $\mathbf{h}$ , de acordo com:  

$$h = \sum_{m=1}^M (d_m - \eta_F(x_m, t_m))$$
em que:  $\mathbf{d}_m$  representa o vetor de observações.
  3. Calcule o representante-adjunto  $\alpha_m$  de acordo com a Equação 4.26a.
  4. Calcule o representante  $r_m(x, t)$  para  $1 \leq m \leq M$   
de acordo com a Equação 4.27a.
  5. Calcule a matriz de covariância  
 $r_m(x_j, t_j)$ , com  $m = 1, 2, \dots, M$  e  $j = 1, 2, \dots, M \Rightarrow \mathbf{R}_{M \times M}$   
 $\mathbf{P} = (\mathbf{R} + \mathbf{w}^{-1}\mathbf{I})$
  6. Determine os coeficientes da expansão do incremento de análise  $\rho$ .  

$$\rho = \sum_{m=1}^M \beta_m r_m, \mathbf{b} = [\beta_1 \ \beta_2 \ \dots \ \beta_m]^T$$
resolvendo o sistema linear:  

$$\mathbf{P}\mathbf{b} = \mathbf{h}, \mathbf{h} = [h_1 \ h_2 \ \dots \ h_m]^T$$
  7. Calcule a análise dada pela equação a seguir:  

$$\hat{\eta}(x, t) = \eta_F(x, t) + \sum_{m=1}^M \beta_m r_m(x, t)$$
- 

Fonte: (BENNETT, 2004)

## 4.2 Método variacional: equação de água rasa 2D

As equações de Euler-Lagrange para o extremo do funcional de penalidade  $J$  para o modelo 2D é dado por:

$$\begin{aligned}
 J = J[u, v, q] = & W_f^u \int_0^T dt \int_0^X dx \int_0^Y dy (f^u(x, y, t))^2 + \\
 & W_f^v \int_0^T dt \int_0^X dx \int_0^Y dy (f^v(x, y, t))^2 + W_f^q \int_0^T dt \int_0^X dx \int_0^Y dy (f^q(x, y, t))^2 + \\
 & W_i^u \int_0^X dx \int_0^Y dy (i^u(x, y))^2 + W_i^v \int_0^X dx \int_0^Y dy (i^v(x, y))^2 + W_i^q \int_0^X dx \int_0^Y dy (i^q(x, y))^2 \\
 & W_b^v \int_0^T dt \int_0^X dx (b_0(x, t))^2 + W_b^v \int_0^T dt \int_0^X dx (b_Y(x, t))^2 + w \sum_{m=1}^M (\varepsilon)^2 \quad (4.52)
 \end{aligned}$$

O funcional de penalidade para o modelo 2.15, em termos da variável de estado é dado por:

$$\begin{aligned}
J = J[u, v, q] = & W_f^u \int_0^T dt \int_0^X dx \int_0^Y dy \left\{ \frac{\partial u}{\partial t} - f v + g \frac{\partial q}{\partial x} + r_u u - F_u \right\}^2 \\
& + W_f^v \int_0^T dt \int_0^X dx \int_0^Y dy \left\{ \frac{\partial v}{\partial t} + f u + g \frac{\partial q}{\partial y} + r_v v - F_v \right\}^2 \\
& + W_f^q \int_0^T dt \int_0^X dx \int_0^Y dy \left\{ \frac{\partial q}{\partial t} + H \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + r_q q \right\}^2 \\
+ W_i^u \int_0^X dx \int_0^Y dy & \{ u(x, y, 0) - I^u(x, y) \}^2 + W_i^v \int_0^X dx \int_0^Y dy \{ v(x, y, 0) - I^v(x, y) \}^2 \\
+ W_i^q \int_0^X dx \int_0^Y dy & \{ q(x, y, 0) - I^q(x, y) \}^2 + W_b^v \int_0^T dt \int_0^X dx \{ v(x, 0, t) \}^2 \\
& + W_b^v \int_0^T dt \int_0^X dx \{ v(x, Y, t) \}^2 + w \sum_{m=1}^M \{ q(x_m, y_m, t_m) - d_m \}^2 \quad (4.53)
\end{aligned}$$

em que  $d_m = q(x_m, y_m, t_m) + \epsilon_m$ . Os residuais são definidos como:

$$\lambda^u \equiv W_f^u \left( \frac{\partial \hat{u}}{\partial t} - f \hat{v} + g \frac{\partial \hat{q}}{\partial x} + r_u \hat{u} - F_u \right) \quad (4.54a)$$

$$\lambda^v \equiv W_f^v \left( \frac{\partial \hat{v}}{\partial t} + f \hat{u} + g \frac{\partial \hat{q}}{\partial y} + r_v \hat{v} - F_v \right) \quad (4.54b)$$

$$\lambda^q \equiv W_f^q \left( \frac{\partial \hat{q}}{\partial t} + H \left( \frac{\partial \hat{u}}{\partial x} + \frac{\partial \hat{v}}{\partial y} \right) + r_q q \right) \quad (4.54c)$$

#### 4.2.1 As equações de Euler-Lagrange

As equações de Euler-Lagrange para o modelo 2.15 são:

$$-\frac{\partial \lambda^u}{\partial t} + f \lambda^v - H \frac{\partial \lambda^q}{\partial x} + r_u \lambda^u = 0 \quad (4.55a)$$

$$-\frac{\partial \lambda^v}{\partial t} - f \lambda^u - H \frac{\partial \lambda^q}{\partial y} + r_v \lambda^v = 0 \quad (4.55b)$$

$$-\frac{\partial \lambda^q}{\partial t} - g \left( \frac{\partial \lambda^u}{\partial x} + \frac{\partial \lambda^v}{\partial y} + r_q \lambda^q \right) = -w \sum_{m=1}^M (\hat{q}(x_m, y_m, t_m) - d_m) \delta(x - x_m) \delta(y - y_m) \delta(t - t_m) \quad (4.55c)$$

com condições iniciais:

$$\lambda^u(x, y, T) = 0 \quad (4.56a)$$

$$\lambda^v(x, y, T) = 0 \quad (4.56b)$$

$$\lambda^q(x, y, T) = 0 \quad (4.56c)$$

Condições de contorno rígidas:

$$\lambda^u(x, 0, T) = \lambda^v(x, Y, T) = 0 \quad (4.57)$$

enquanto todos os campos são periódicos na direção  $x$ :

$$\lambda^u(x \pm X, y, t) = \lambda^u(x, y, t) \quad (4.58a)$$

$$\lambda^v(x \pm X, y, t) = \lambda^v(x, y, t) \quad (4.58b)$$

$$\lambda^q(x \pm X, y, t) = \lambda^q(x, y, t) \quad (4.58c)$$

#### 4.2.2 Equações adjunta e *representante*

As equações adjunta para o modelo 2.15 são:

$$-\frac{\partial \alpha_m^u}{\partial t} + f \alpha_m^v - H \frac{\partial \alpha_m^q}{\partial x} + r_u \alpha_m^u = 0 \quad (4.59a)$$

$$-\frac{\partial \alpha_m^v}{\partial t} - f \alpha_m^u - H \frac{\partial \alpha_m^q}{\partial y} + r_v \alpha_m^v = 0 \quad (4.59b)$$

$$-\frac{\partial \alpha_m^q}{\partial t} - g \left( \frac{\partial \alpha_m^u}{\partial x} + \frac{\alpha_m^v}{\partial y} \right) + r_q \alpha_m^q = \delta(x - x_m) \delta(y - y_m) \delta(t - t_m) \quad (4.59c)$$

sujeito as seguintes condições:

$$\alpha^u(x, y, T) = 0 \quad (4.60a)$$

$$\alpha^v(x, y, T) = 0 \quad (4.60b)$$

$$\alpha^q(x, y, T) = 0 \quad (4.60c)$$

e condições de contorno rígidas  $\lambda^u(x, 0, T) = \lambda^v(x, Y, T) = 0$ , e

$$\alpha_m^u(x \pm X, y, t) = \alpha_m^u(x, y, t) \quad (4.61a)$$

$$\alpha_m^v(x \pm X, y, t) = \alpha_m^v(x, y, t) \quad (4.61b)$$

$$\alpha_m^q(x \pm X, y, t) = \alpha_m^q(x, y, t) \quad (4.61c)$$

Finalmente, as equações do *representante* para o modelo de água rasa linear 2D são dadas por:

$$\frac{\partial r_m^u}{\partial t} + fr_m^v + g \frac{\partial r_m^q}{\partial x} + r_u r_m^u = [W_f^u]^{-1} \alpha_m^u \quad (4.62a)$$

$$\frac{\partial r_m^v}{\partial t} + fr_m^u + g \frac{\partial r_m^q}{\partial y} + r_v r_m^v = [W_f^v]^{-1} \alpha_m^v \quad (4.62b)$$

$$\frac{\partial r_m^q}{\partial t} + H \left( \frac{r_m^u}{\partial x} + \frac{r_m^v}{\partial y} \right) + r_q r_m^q = [W_f^q]^{-1} \alpha_m^q \quad (4.62c)$$

com:

$$r_m^u(x, y, 0) = [W_i^u]^{-1} \alpha_m^u(x, y, 0) \quad (4.63a)$$

$$r_m^v(x, y, 0) = [W_i^v]^{-1} \alpha_m^v(x, y, 0) \quad (4.63b)$$

$$r_m^q(x, y, 0) = [W_i^q]^{-1} \alpha_m^q(x, y, 0) \quad (4.63c)$$

com condições de contorno:

$$r_m^v(x, 0, t) = H[W_b^v]^{-1} \alpha(x, 0, t) \quad (4.64a)$$

$$r_m^v(x, Y, t) = -H[W_b^v]^{-1} \alpha(x, Y, t) \quad (4.64b)$$

$$r_m^u(x \pm X, y, t) = r_m^u(x, y, t) \quad (4.65a)$$

$$r_m^v(x \pm X, y, t) = r_m^v(x, y, t) \quad (4.65b)$$

$$r_m^q(x \pm X, y, t) = r_m^q(x, y, t) \quad (4.65c)$$

Assim, o extremo de  $J$  é calculado por:

$$\hat{u}(x, y, t) = u_F(x, y, t) + \sum_{m=1}^M \beta_m r_m^u(x, y, t) \quad (4.66a)$$

$$\hat{v}(x, y, t) = v_F(x, y, t) + \sum_{m=1}^M \beta_m r_m^v(x, y, t) \quad (4.66b)$$

$$\hat{q}(x, y, t) = q_F(x, y, t) + \sum_{m=1}^M \beta_m r_m^q(x, y, t) \quad (4.66c)$$

Tabela 4.2 - Algoritmo *representante* para o modelo de água rasa linear.

---

**Algoritmo: *Representante* em duas dimensões**

---

1. Calcule as variáveis de estado:

$$\Phi_{mod} = [u_F(x, y, t) \quad v_F(x, y, t) \quad q_F(x, y, t)]^T$$

por meio da integração numérica da equação 2.15

2. Calcule a inovação  $\mathbf{h}$ , de acordo com:

$$h = \sum_{m=1}^M (\Phi_{obs} - \Phi_{mod})$$

em que  $\Phi_{mod} = [u_{obs}(x_m, y_m, t_m) \quad v_{obs}(x_m, y_m, t_m) \quad q_{obs}(x_m, y_m, t_m)]^T$

3. Calcule o representante-adjunto  $\alpha_m$  de acordo com a equação 4.59.

4. Calcule o representante  $r_m^u(x_j, y_j, t_j)$ ,  $r_m^v(x_j, y_j, t_j)$  e  $r_m^q(x_j, y_j, t_j)$  para  $1 \leq m \leq M$  de acordo com a equação 4.62.

5. Calcule a matriz dos *representantes*

$$r_m^u(x_j, y_j, t_j), \text{ com } m = 1, \dots, M \text{ e } j = 1, \dots, M$$

$$r_m^v(x_j, y_j, t_j), \text{ com } m = 1, \dots, M \text{ e } j = 1, \dots, M$$

$$r_m^q(x_j, y_j, t_j), \text{ com } m = 1, \dots, M \text{ e } j = 1, \dots, M$$

$$\mathbf{P} = (\mathbf{R} + \mathbf{w}^{-1}\mathbf{I})$$

em que  $\mathbf{w}$  representa a matriz de covariância do erro de observação.

6. Calcule o coeficiente do representante

$$\mathbf{P}\mathbf{b} = \mathbf{h}, \quad \mathbf{b} = [\beta_1 \quad \beta_2 \quad \dots \quad \beta_M] \text{ e } \mathbf{h} = [h_1 \quad h_2 \quad \dots \quad h_M]$$

7. Calcule a análise para cada variável de estado dada pelas equações a seguir:

$$\hat{u}(x, y, t) = u_F(x, y, t) + \sum_{m=1}^M \beta_m r_m^u(x, y, t)$$

$$\hat{v}(x, y, t) = v_F(x, y, t) + \sum_{m=1}^M \beta_m r_m^v(x, y, t)$$

$$\hat{q}(x, y, t) = q_F(x, y, t) + \sum_{m=1}^M \beta_m r_m^q(x, y, t)$$


---

Fonte: (BENNETT, 2002)

### 4.3 Implementação do método representante

O algoritmo do representante fornece uma solução explícita para a solução das equações de Euler-Lagrange. Há um *representante* para cada observação. Para o cálculo de cada representante, faz-se necessário duas integrações, uma *backward* e outra *forward*. Se o conjunto de observações é muito grande, torna-se impraticável calcular um representante para cada valor de dado observado (BENNETT, 2002).

Retomando a Equação 4.28 a solução do *representante* para a inversão para o modelo

em uma dimensão é dada por:

$$\hat{\eta}(x, t) = \eta_F(x, t) + \sum_{m=1}^M \hat{\beta}_m r_m(x, t) \quad (4.67)$$

em que

$$(\mathbf{R} + \mathbf{C}_\epsilon)\hat{\beta} = \mathbf{h} \equiv \mathbf{d} - L[\eta_F] \quad (4.68)$$

Portanto, a tarefa para o cálculo dos *representantes* consiste em:

1. Integra-se o modelo *forward* para  $\eta_F$  (1 integração) 2.1;
2. Integra-se o modelo *backward* para  $\alpha$  ( $M$  integrações) 4.26a;
3. Integra-se o modelo *forward* para  $r$  ( $M$  integrações) 4.27a

Isso resulta em um total de  $I = 2M + 1$  integrações. A parte *forward* e *backward* da equação de Euler-Lagrange são acoplados por  $M$  números de  $\hat{\eta}_1, \dots, \hat{\eta}_M$ . O coeficiente do vetor dos impulsos na equação adjunta, ou vetor de acoplamento é dado por:

$$\mathbf{C}_\epsilon^{-1}(\mathbf{d} - L[\hat{\eta}]) = \mathbf{C}_\epsilon^{-1}\{\mathbf{d} - L[\eta_F] - \mathbf{R}^T \hat{\beta}\} \quad (4.69)$$

$$= \mathbf{C}_\epsilon^{-1}\{\mathbf{h} - \mathbf{R}^T(\mathbf{R} + \mathbf{C}_\epsilon)^{-1}\mathbf{h}\} \quad (4.70)$$

$$= (\mathbf{R} + \mathbf{C}_\epsilon)^{-1}\mathbf{h}$$

$$= \hat{\beta}$$

Assim, o vetor de acoplamento é um vetor de coeficientes ***representantes***. Portanto, não é preciso armazenar o campo do vetor *representante*  $\mathbf{r}(x, t)$ . Precisa-se calcular  $\mathbf{r}(x, t)$  para obter a matriz *representante*  $R = L[\mathbf{r}^T]$ , calcular a Equação 4.68 para  $\hat{\beta}$ , integra-se a equação adjunta para a equação de Euler-Lagrange para  $\lambda(x, t)$  e integra-se a equação avançada para  $\hat{\eta}(x, t)$ .

A Figura 4.2 mostra a sequência em que devem ser implementadas as equações para o cálculo dos coeficientes representantes ( $\hat{\beta}$ ). A construção é feita de modo direto, isto é, com o cálculo da matriz  $\mathbf{R}$ , que contém os coeficientes *representantes*. A seta na vertical indica a sequência das execuções, que começa no topo da Figura 4.2. A Equação 4.67 não precisa ser assumida explicitamente, uma vez que  $\hat{\beta}$  é conhecido. A Equação 4.71 resolve o acoplamento na Equação 4.23a–4.23c e

na Equação 4.24a–4.24c. Então,  $\lambda$  e  $\hat{\eta}$  podem ser calculados com uma integração *backward* e *forward*, respectivamente. Os *representantes*  $r_m$ ,  $1 \leq m \leq M$ , não precisam ser armazenados. Se as inversas de  $W_f^{-1}$ ,  $W_i^{-1}$ ,  $w_b^{-1}$  não são diagonais, então a Equação 4.27a–4.27c e Equação 4.24a–4.24a requerem convolução.

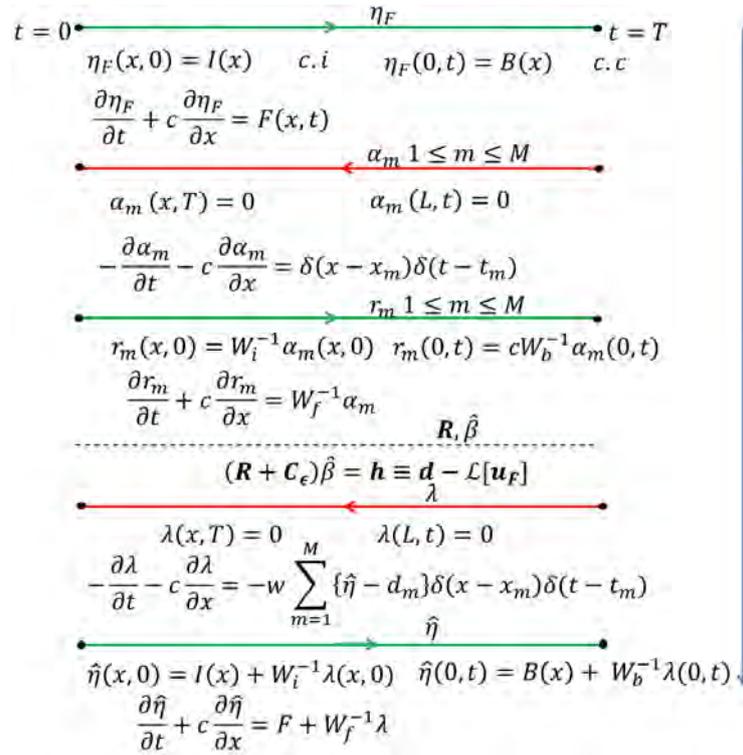


Figura 4.2 - Esboço para a implementação do algoritmo representante com o cálculo direto do coeficiente  $\hat{\beta}$ .

Uma outra forma de calcular os coeficientes do representante ( $\beta$ ) é por meio do algoritmo do representante indireto ilustrado na Figura 4.2. A inversão generalizada reduz exatamente à solução do sistema de dimensão finita:

$$(\mathbf{R} + \mathbf{C}_\epsilon)\hat{\beta} \equiv \mathbf{d} - L[\eta_F] \quad (4.71)$$

ou simplesmente resolver o sistema linear:

$$\mathbf{P}\hat{\beta} = \mathbf{h} \quad (4.72)$$

A solução indireta requer que  $\mathbf{P}$  e  $\mathbf{R}$  sejam conhecidos explicitamente. Portanto, a solução pode ser obtida de modo iterativo, fornecendo  $\mathbf{P}\phi$  podendo ser avaliada para algum  $\phi$ . Então, a solução iterativa padrão pode converter um *first-guess*  $\hat{\beta}$  dentro da solução  $\hat{\beta} = \mathbf{P}^{-1}\mathbf{h}$

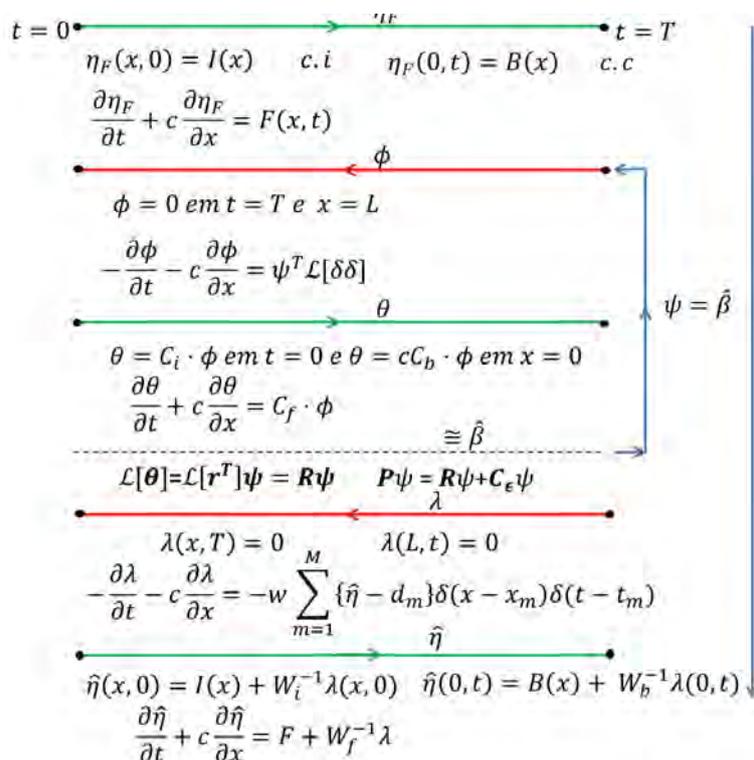


Figura 4.3 - Esboço para a implementação do algoritmo do representante com o cálculo indireto do coeficiente  $\hat{\beta}$ .

A Subseção 4.3.1 descreve o *software* IOM-*Inverse Ocean Modeling* que contém os códigos fontes do método do representante.

#### 4.3.1 Pacote de software IOM

Os códigos fontes dos modelos de previsão e assimilação de dados foram desenvolvidos em linguagem Fortran90. A linguagem de programação FORTRAN, acrônimo da expressão “*IBM Mathematical FORMula TRANslation System*”, é uma linguagem de programação procedural que possuem características que permitem suportar programação orientada a objetos. No núcleo do sistema IOM, uma parte do *software* foi desenvolvida em *Parametric Fortran*, que é um gerador de programas Fortran que

produz extensões para aplicações especializadas. Este compilador deve fazer parte do ambiente de instalação do software IOM.

O *software* IOM foi desenvolvido pelo grupo de pesquisa do Prof. Andrew F. Bennett do Colégio de Ciências Atmosféricas e Oceânicas de Oregon. Esse sistema é um auxiliar didático para que se possa realizar experimentos com o método do Representante.

Para instalação do *software* IOM, além dos compiladores Fortran, o ambiente para operação do sistema de assimilação de dados deve conter *softwares* como:

- PERL - acrônimo de *Practical Extraction and Report Language* - trata-se de uma linguagem de código aberto. O Perl suporta a programação orientada a objetos, é uma linguagem de script cuja principal característica é a manipulação de dados, repleta de truques de automatização. O *script* principal para executar o *software* IOM está escrito em Perl. O interpretador Perl deve ser instalado de acordo com o sistema operacional.
- XML, do inglês *eXtensible Markup Language*. Esta é uma linguagem de marcação, ou seja, apresenta um agregado de códigos que podem ser aplicados a dados ou textos para serem lidos por computadores ou pessoas. O XML é utilizado para padronizar uma sequência de dados com o objetivo de organizar, separar o conteúdo e integrá-lo com outras linguagens.
- Javascript - outra linguagem de script incorporada a um documento para a *web*. Esta linguagem de programação permite a execução de comandos do cliente, ou seja, em termos do navegador e não do servidor *web*.
- As bases de dados estão no formato NETCDF. Este formato de arquivo foi desenvolvido pela UNIDATA. O Network Common Data Format (netCDF) foi desenvolvido para prover um método único de acesso a tipos de dados variados, podendo incluir séries temporais, em grades regulares e informações/imagens de satélites ou campos meteorológicos. O Software netCDF é uma biblioteca de funções de I/O para várias linguagens. Esta biblioteca trata conjuntos de dados de forma independente da máquina utilizada. Desta forma fica garantida a compatibilidade dos dados quando usado em diferentes plataformas. Esta biblioteca de acesso para leitura e escrita de dados no formato netCDF é fornecida pela Unidata, de forma

livre, sem nenhuma forma de licenciamento e nenhuma outra importante restrição.

As opções de execução dos programas do IOM foram desenvolvidas para serem executadas por meio de uma interface gráfica construídas com as linguagens de marcação e de *web* ilustrada na [Figura 4.4](#). Para a opção do modelo de água rasa em duas dimensões, a interface gráfica não funcionou. Após a instalação do *software*, os

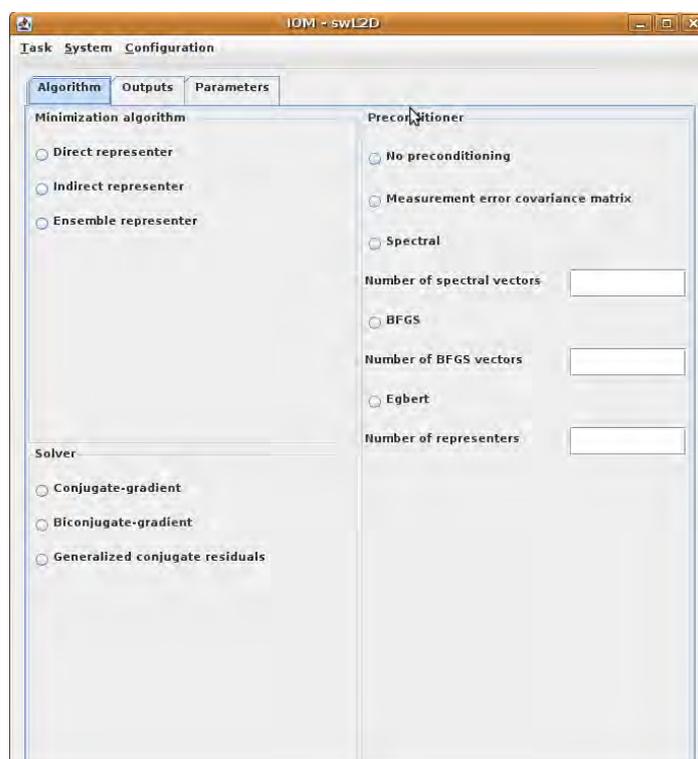


Figura 4.4 - Interface gráfica do software IOM.

programas foram rodados com os *scripts* programados em Perl e com parâmetros e opções no *script* escrito XML de acordo com o [Apêndice B](#). O software possui programas fontes para os modelos:

- Shallow water em duas dimensões (swL2D),
- equação da onda de uma dimensão (wave1D),
- equação KdV - Korteweg-de Vries ([MUCCINO; BENNETT, 2002](#)),

- modelo PEZHAT - *Primitive Equations Z-coordinate-Harmonic Analysis of Tides* (MUCCINO et al., 2008).

Existe uma estrutura de diretórios para cada modelo.

Nesta tese, trabalhou-se com os diretórios wave1D e swL2D. Para instalação dos núcleos destes modelos, deve-se compilar os programas que se encontram em cada estrutura de diretórios correspondente para obter os programas objetos que atendam às diversas opções constantes do arquivo XML da execução. Os diretórios com os códigos fontes e respectivos *scripts* de compilação são:

1. `cgdata` - contém arquivos com os *checkpoints* do método iterativo gradiente conjugado usado na solução do problema;
2. `inputs` - contém arquivos com as observações `inver_par_obs.nc` e com o desvio padrão do erros do modelo em pontos de grade `traj_model_std_err.nc`
3. `iom` - contém programas fontes e objetos de varias opções para execução do sistema de assimilação de dados do representante:
  - 3.1 `SRC_iom_bicg` - resolve o problema iterativo para chegar ao menor valor do funcional de penalidade utilizando o gradiente conjugado.
  - 3.2 `SRC_iom_preconditioner` - resolve o problema iterativo para chegar ao menor valor do funcional de penalidade utilizando o gradiente conjugado preconditionado;
  - 3.3 `SRC_iom_lapack_56` - resolve o problema iterativo para chegar ao menor valor do funcional de penalidade utilizando o gradiente conjugado da biblioteca matemática LAPACK;
  - 3.4 `SRC_iom_standard` - resolve o problema iterativo para chegar ao menor valor do funcional de penalidade utilizando o gradiente conjugado padrão;
  - 3.5 `SRC_iom_conv1` - resolve a análise, fazendo a convolução do incremento de análise na grade do modelo;
  - 3.6 `SRC_iom_cov` - resolve as matrizes de covariâncias de erros do modelo;

3.7 `SRC_iom_hand_written` - escreve os arquivos de observação e das penalidades;

3.8 `SRC_iom_meas_comb` fazem as interpolações do ponto de observação para grade com as devidas influências de observação e dos pontos de grade para as observações;

3.9 `SRC_iom_traj` - escreve os arquivos com as trajetórias dos modelos.

4. `ker` - contém programas que cria a grade de observação e de influência das observações a serem interpolados;
5. `model` - contém os programas para integrar as equações do modelo avançado (`foward`), do modelo adjunto (`adjoint`) e do modelo tangente linear;
6. `outputs` - contém os arquivos com os funcionais de penalidade calculados na execução e com as inovações (diferenças das observações e modelo);
7. `run` - contém os scripts em perl, xml e os arquivos textos (`namelists`) com parâmetros necessários para execução do software;
8. `sharedata` - contém os arquivos das trajetórias do modelo (inicial e análise) e dos vetores interpolados em pontos de observação;

O método iterativo utilizado para execução do modelo swL2D, constante da opção do *script* xml, foi o gradiente conjugado preconditionado. E na execução o diretório `ker` deve ser o primeiro a ser executado. Os arquivos do diretório `input` devem existir antes da execução dos programas, bem como, os *namelists* do diretório `run`. Para a execução dos modelos, é necessário escolher as opções no arquivo `inversion.xml`, em seguida dentro do diretório `run`, na linha de comando digitar: `./iom_script.pl inversion.xml`. A extensão “.pl” identifica o *script* na linguagem perl, que executa os seguintes passos:

1. Previsão: executando o comando `iom_comp_traj_prior`;
2. Calcula-se o vetor de inovação *h* executando o comando `iom_comp_vect`;
3. Calcula-se o funcional de penalidade executando o comando `iom_comp_penalty`;

4. Calcula-se os coeficientes do representante executando o comando:  
`iom_comp_coefficients_solver_bi_conjugate-gradient`
  - 4.1 faz-se o cálculo do gradiente conjugado para a minimização do funcional;
  - 4.2 calcula-se o comb (seta as observações no ponto de grade do modelo);
  - 4.3 integra-se o modelo adjunto;
  - 4.4 faz a convolução;
  - 4.5 integra-se o modelo representante executando o comando:  
`iom_get_VECT_Rp`;
5. Calcula-se a trajetória ótima executando o comando:  
`iom_comp_traj_open_loop`;
6. Faz-se a convolução executando o comando `iom_comp_convolve`;
7. O modelo faz a previsão com as correções da assimilação.

Para realizar os experimentos descrito nesta tese, foi necessário executar várias adaptações nos códigos (várias rotinas), tarefa que foi difícil de realizar devido ao uso de todas as linguagens (perl, XML, Java, NETCDF, parametric Fortran e Fortran 90) que trabalham em conjunto. Alterar parâmetros do modelo, por exemplo, implica em fazer modificações em diversos módulos do software, o que tornou a implementação difícil.



## 5 REDES NEURAIS ARTIFICIAIS

Neste capítulo apresenta-se a metodologia de Redes Neurais Artificiais (RNA), com as suas principais características. Essa metodologia é considerada uma nova abordagem em um contexto de assimilação de dados para “*emular*” as técnicas apresentadas no [Capítulo 3](#) e [Capítulo 4](#). O estudo de redes neurais artificiais tem uma área de aplicação muito abrangente, tornando-se uma técnica utilizada na solução de problemas em análise de séries temporais, reconhecimento de padrões, processamento de sinais, em assimilação de dados, que é o contexto de aplicação desse trabalho, em problemas inversos, que podemos citar os trabalhos de [Shiguemori \(2007\)](#) e [Paes \(2011\)](#), dentre outras aplicações. As definições aqui apresentadas serão feitas dentro do contexto multidisciplinar.

Os neurônios artificiais foram inspirados pelos neurônios biológicos, que, conseqüentemente, compõem uma rede neural biológica. Os neurônios são as células que formam o cérebro humano, um sistema complexo e distribuído, com capacidade de armazenamento e processamento de informações extremamente rápido. As redes neurais artificiais são compostas por neurônios artificiais, um modelo muito simples baseado no funcionamento do neurônio biológico. A seguir, descreve-se um neurônio biológico, ressaltando suas partes principais, com o objetivo de mostrar uma analogia entre a funcionalidade de um neurônio biológico e um neurônio artificial.

### 5.1 Neurônio Biológico

De acordo com [Kovacs \(1996\)](#), o neurônio ou célula nervosa, ilustrado na [Figura 5.1](#), aparece de diferentes formas e tamanhos no cérebro humano. O neurônio é composto basicamente por dendritos, axônios, sinapses e núcleo (ou soma), que é o centro dos processos metabólicos da célula nervosa. Os axônios têm a função de transmitir informações para outros neurônios, já os dendritos de receber informações em forma de sinais, que são pulsos elétricos conhecidos como impulsos *nervosos* ou *potenciais de ação*.

As sinapses podem ser consideradas como regiões eletroquimicamente ativas, compreendidas entre duas membranas celulares: a membrana *pré-sináptica*, por onde chega um estímulo proveniente de uma outra célula, e a membrana *pós-sináptica*, que é a membrana do dendrito. Nesta região intersináptica, o estímulo nervoso que chega à sinapse é transferido para a membrana dendrital através de substâncias conhecidas

como *neurotransmissores*. O resultado dessa transferência é uma alteração no potencial elétrico da membrana *pós-sináptica*. Dependendo do tipo de neurotransmissor, a conexão sináptica será excitatória ou inibitória, sendo que uma conexão excitatória provoca uma alteração no potencial da membrana que contribui para a formação de um impulso nervoso no axônio de saída, enquanto uma conexão inibitória age no sentido oposto.

O neurônio biológico pode ser visto como um dispositivo computacional elementar básico do sistema nervoso, com muitas entradas e uma saída. As entradas ocorrem através das conexões sinápticas, que conectam a árvore dendritica aos axônios de outras células nervosas. Os sinais que chegam por estes axônios são pulsos elétricos conhecidos como impulsos nervosos e constituem a formação que será processada pelo neurônio, para produzir como saída um impulso nervoso no seu axônio.

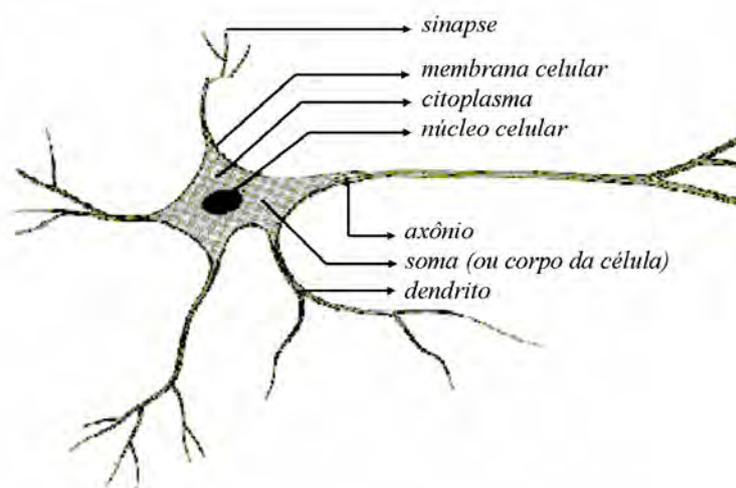


Figura 5.1 - Representação de um neurônio biológico.  
Fonte: Adaptado de [Tatibana e Kaetsu \(2012\)](#).

## 5.2 Neurônio Artificial

Os elementos do neurônio artificial possuem uma analogia direta com os elementos de um neurônio biológico. Segundo [Haykin \(2004\)](#) um neurônio artificial é uma unidade de processamento de informações, que é fundamental para a operação de uma rede neural. O modelo básico de um neurônio artificial foi definido por [McCulloch e Pitts](#)

(1943). A descrição matemática resultou em um modelo com  $n$  sinais de entrada  $x_1, x_2, \dots, x_n$ , representando os neurônios, e um sinal de saída  $y$ , que representa o axônio. Para *emular* o comportamento das sinapses, os sinais de entrada do neurônio artificial têm pesos acoplados  $\theta_1, \theta_2, \dots, \theta_n$ , que podem assumir valores positivos ou negativos. O efeito de uma sinapse particular em um neurônio  $i$  pós-sináptico é dado pela combinação linear  $x_i\theta_i$ . O corpo do neurônio é emulado por um mecanismo simples que faz a soma ponderada dos valores recebidos  $x_i\theta_i$ . A ativação do neurônio é obtida através de uma função de ativação. Alguns tipos de funções de ativação são descritas na [Subseção 5.2.1](#).

Identificam-se três elementos básicos de um modelo neural artificial.

- Um conjunto de *sinapses*, cada uma das quais é caracterizada por um peso correspondente. Especificamente, um sinal  $x_n$  na entrada da sinapse  $n$  conectada ao neurônio  $k$  é multiplicado pelo peso sináptico  $\theta_{nk}$ ;
- Um *somador* de sinais de entrada, ponderados pelas respectivas sinapses de cada neurônio;
- Uma *função de ativação*, para limitar a amplitude de saída do neurônio. Normalmente, a faixa de amplitude da saída de um neurônio é o intervalo  $[0, 1]$  ou  $[-1, 1]$ .

No modelo de um neurônio artificial, exemplificado na [Figura 5.2](#), está incluído um limiar  $b_k$ , que tem o efeito de acrescentar um grau de liberdade a cada neurônio.

Uma unidade de processamento em uma rede neural é uma combinação linear com vários pesos de entrada, seguido por uma função de ativação. O  $k$ -ésimo neurônio pode ser descrito pelo acoplamento de duas equações:

$$u_k = \sum_{j=1}^m \theta_{kj}x_j \quad (5.1)$$

$$y_k = \varphi(u_k + b_k) \quad (5.2)$$

em que  $x_1, \dots, x_m$  são sinais de entrada;  $\theta_{k_1}, \dots, \theta_{k_m}$  são os pesos sinápticos do neurônio  $k$ ;  $u_k$  é a saída do combinador linear devido aos sinais de entrada;  $b_k$  é o viés;  $\varphi(\cdot)$ , é a função de ativação e  $y_k$  é o sinal de saída do neurônio. O uso do viés ou

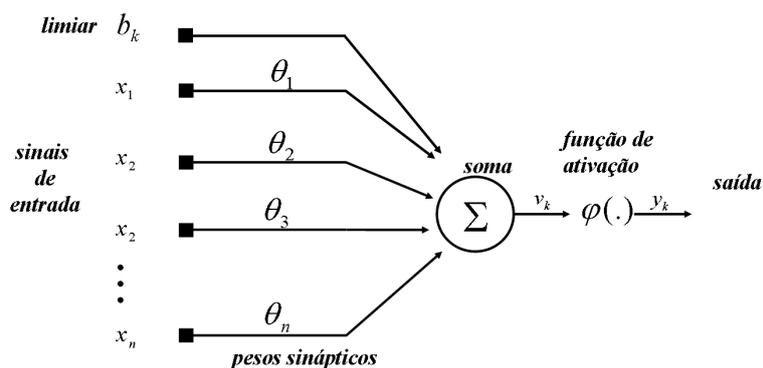


Figura 5.2 - Representação de um neurônio artificial.  
 Fonte: Adaptada de Haykin (2004).

limiar  $b_k$  tem o efeito de aplicar uma transformação afim à saída  $u_k$  do combinador linear do modelo da Figura 5.1, conforme

$$v_k = u_k + b_k = \sum_{j=1}^m \theta_{kj} x_j + b_k \quad (5.3)$$

### 5.2.1 Funções de ativação

A função  $\varphi(\cdot)$  representa a função de ativação, a qual restringe a amplitude do sinal na saída de um neurônio. O papel da função de ativação é simular características não lineares de um neurônio biológico. A seguir, identificam-se três tipos básicos de função de ativação:

1. *função degrau*: é uma função utilizada para valores binários. O primeiro neurônio usando um dispositivo binário foi introduzido por McCulloch e Pitts (1943), no qual a saída é pulso ou não pulso, sendo que suas entradas têm ganho arbitrário, podendo ser excitatório ou inibitório. Para determinar a saída do neurônio, calcula-se a soma ponderada das entradas com os respectivos pesos como fatores de ponderação, positivos, nos casos excitatórios, e negativos, nos casos inibitórios. Se esse resultado for maior ou igual a certo limiar, então a saída do neurônio é pulso, e caso contrário

é não pulso. Matematicamente a função é representada por:

$$\varphi(v) = \begin{cases} 1 & \text{se } v > 0 \\ 0 & \text{se } v \leq 0 \end{cases} \quad (5.4)$$

2. *função tangente hiperbólica*: é uma função sigmóide que varia entre  $[-1, 1]$  é dada por:

$$\varphi(v_j) = \tanh\left(\frac{av_j}{2}\right); \text{ para } a = 1 \quad (5.5)$$

3. *função exponencial*

$$\varphi(v_j) = \exp\left(-\frac{(v_j - \mu)^2}{2\sigma^2}\right); \text{ com } \sigma = 1, \mu = 0 \quad (5.6)$$

Os valores nas entradas devem ser normalizados para valores restritos ao intervalo da função de ativação utilizada e, posteriormente, deve ser feito o processo inverso para que as variáveis estimadas voltem ao intervalo da função original.

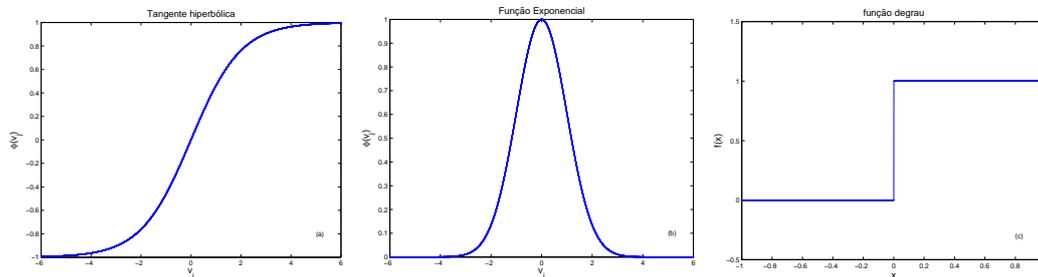


Figura 5.3 - Exemplos de funções de ativação mais utilizadas: (a) tangente hiperbólica; (b) exponencial; (c) função degrau.

### 5.3 Funcionamento das Redes Neurais

Uma rede neural artificial, segundo [Fausett \(1994\)](#), é caracterizada por:

- (i) O padrão de conexão entre os neurônios, o que determina a arquitetura da rede.
- (ii) O método de determinação dos pesos sobre as conexões (chamado de treinamento ou algoritmo de aprendizagem).

(iii) A função de ativação

Os neurônios podem ser dispostos de tal maneira a formar uma ou mais camadas. Neste texto, as entradas da rede são denominadas de camada de entrada, a primeira camada sem contato com o meio externo é chamada de camada escondida e os valores na saída da rede são chamados de camada de saída.

Nas redes onde o sinal de entrada projeta-se diretamente sobre a camada de saída, a rede é denominada de *feed-forward* (alimentação para frente). Quando há uma realimentação entre uma das camadas da rede, ela é chamada de rede recorrente.

O funcionamento de uma rede neural é realizado através de duas fases:

- aprendizagem, também conhecido como treinamento;
- ativação

No processo de aprendizagem um conjunto de exemplos são apresentados à rede, que extrai as características necessárias para representar a informação fornecida. Em outras palavras, no processo de aprendizagem há uma adaptação dos pesos sinápticos e do bias dos neurônios em resposta às entradas e à saída desejada. As características extraídas são armazenadas nos pesos sinápticos, que são utilizados posteriormente no processo de ativação da rede produzindo, portanto, respostas para o problema.

Um fato importante na fase de treinamento das RNA é adotar um critério de parada. Normalmente, a rede é treinada até que um número máximo de “*épocas*” de treinamento, que é definido *a priori*, seja atingido. Define-se uma *época* de treinamento como sendo a apresentação de todo um conjunto (ou padrões) de treinamento da rede. O critério de parada em função do erro objetivo pode ser aplicado somente por redes que utilizam aprendizagem supervisionada, pois o erro é calculado entre a saída desejada e a saída obtida da rede. Segundo Braga et al. (1998) os critérios de parada mais utilizados são:

- Encerrar o treinamento após  $N$  épocas;
- Encerrar o treinamento após o erro quadrático médio (EQM) estiver abaixo de uma constante  $\epsilon$  pré-definida;

- Encerrar o treinamento quando a porcentagem de classificações corretas estiver acima de uma constante  $\epsilon$ . Esta alternativa é mais indicada para saídas binárias;
- Combinação dos métodos acima.

Uma definição de aprendizagem, adaptada de Mendel e McLaren (1970), no contexto de redes neurais disponível em (HAYKIN, 2004) é:

*Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado de acordo com a modificação dos parâmetros da rede.*

### 5.3.1 Regras de Aprendizagem

Algoritmo de aprendizagem é um conjunto de procedimentos bem definidos para adaptação dos parâmetros de uma rede neural para que a mesma possa aprender uma determinada função. Os tipos de aprendizagem são: supervisionados ou não supervisionados. Na aprendizagem supervisionada, tanto a entrada quanto a saída da rede são conhecidas, sendo que a rede opera no sentido de encontrar pesos que minimizem a diferença entre a entrada e a saída desejada em um sentido estatístico, geralmente, o erro médio quadrático.

Já na aprendizagem não supervisionada, somente os padrões de entrada estão disponíveis na rede. A partir do momento em que a rede estabelece uma harmonia com as regularidades estatísticas da entrada de dados, desenvolve-se nela uma habilidade de formar representações internas para codificar características da entrada e criar novas classes ou grupos automaticamente (BRAGA et al., 2000). Neste tipo de aprendizagem, a rede se auto-organiza para que cada neurônio responda a diferentes conjuntos de entrada, um exemplo clássico desse tipo de abordagem, são os mapas topológicos de Kohonen (1995).

As regras usualmente aplicadas para o processo de aprendizagem supervisionado são: correção de erros e a regra delta, que foi generalizada para o treinamento do perceptron de múltiplas camadas, conhecido como algoritmo de retropropagação do erro. Na seção a seguir, descreve-se o algoritmo de retropropagação segundo (HAYKIN, 2004).

## 5.4 Algoritmo “*Backpropagation*” ou de Retropropagação

1. **Iniciar os pesos:** considerando que nenhuma informação prévia está disponível, os pesos sinápticos e limiares são iniciados com números aleatórios que seguem uma distribuição uniforme.
2. **Apresentação dos exemplos de treinamento:** apresenta-se uma época de exemplos de treinamento à rede. Para cada exemplo do conjunto de treinamento, executa-se os passos 3 e 4 apresentados a seguir:
3. **Propagação:** seja o conjunto de treinamento representado por  $(x(n), d(n))$ , em que  $x(n)$  é o vetor de entrada aplicado a primeira camada de nós sensoriais e o vetor de resposta desejada  $d(n)$  apresentado a camada de saída de nós computacionais. Calculam-se os campos locais induzidos e os sinais funcionais, camada por camada da rede. O campo local induzido  $v_j^{(l)}(n)$  para o neurônio  $j$  na camada  $l$  é dado por:

$$v_j^{(l)}(n) = \sum_{i=0}^{m_o} \theta_{ji}^{(l)}(n) y_i^{(l-1)}(n) \quad (5.7)$$

em que  $y_i^{(l-1)}(n)$  é o sinal de saída do neurônio  $i$  na camada anterior  $l - 1$ , na iteração  $n$ , e  $\theta_{ji}^{(l)}(n)$  é o peso sináptico do neurônio  $j$  da camada  $l$ , que é alimentado pelo neurônio  $i$  da camada  $(l - 1)$ . Para  $i = 0$ , temos  $y_0^{(l-1)}(n) = +1$  e  $\theta_{j0}^{(l)}(n) = b_{j0}^{(l)}(n)$  é o vies aplicado ao neurônio  $j$  na camada  $l$ . O sinal de saída do neurônio  $j$  na camada  $l$  é:

$$y_j^l = \varphi_j(v_j(n)) \quad (5.8)$$

Se o neurônio  $j$  está na primeira camada oculta (i.e.,  $l = 1$ ), faz-se  $y_j^{(0)}(n) = x_j(n)$ ,  $x_j(n)$  é o  $j$ -ésimo elemento do vetor de entrada  $\mathbf{x}(n)$ . Se o neurônio  $j$  está na camada de saída (i.e.,  $l = L$  em que  $L$  é denominado a profundidade da rede), se faz  $y_j^{(L)} = o_j(n)$ .

Calcula-se o sinal do erro

$$e_j^{(n)} = d_j^{(n)} - o_j^{(n)} \quad (5.9)$$

em que  $d_j^{(n)}$  é o  $j$ -ésimo elemento do vetor resposta desejada  $d(n)$ .

4. **Retropropagação:** Calculam-se os gradientes locais da rede definidos por:

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n)\varphi_j^{(l)}(v_l^{(L)}(n)) & \text{neurônio } j \text{ na camada} \\ & \text{de saída } L, \\ \varphi_j^{(l)}(v_l^{(L)}(n)) \sum_k \delta_k^{(l+1)}(n)\theta_{kj}^{(l+1)}(n) & \text{neurônio } j \text{ na camada} \\ & \text{oculta } l. \end{cases} \quad (5.10)$$

onde o apóstrofo  $\varphi_j^{(l)}(\cdot)$  representa a diferenciação em relação ao argumento. Assim, ajusta-se os pesos sinápticos da rede na camada  $l$  de acordo com a seguinte regra, conhecida como regra delta generalizada.

$$\theta_{ji}^{(n+1)} = \theta_{ji}^{(n)} + \alpha[\theta_{ji}^{(n)} - \theta_{ji}^{(n-1)}] + \eta\delta_j^{(l)}(n)y_i^{(l-1)}(n) \quad (5.11)$$

em que  $\eta$  é a taxa de aprendizagem e  $\alpha$  é a constante de momento.

5. **Iteração:** repetem-se os passos 3 e 4, apresentando a rede novos exemplos de treinamento até que o critério de parada seja satisfeito.

## 5.5 Perceptron de Múltiplas Camadas

A rede Perceptron de Múltiplas Camadas (PMC) é uma das redes mais utilizadas na literatura para a solução de problemas. Esta rede possui treinamento supervisionado realizado por meio do algoritmo de retropropagação do erro apresentado na seção 5.4.

Em uma rede multicamadas são identificadas a camada de entrada, uma ou mais camadas ocultas e a camada de saída de nós computacionais, ilustrada na [Figura 5.4](#). Seu algoritmo de treinamento, aprendizagem por retropropagação do erro, consiste em dois passos através das diferentes camadas da rede: um passo para frente, a propagação e um passo para trás, a retropropagação. Na fase de propagação da rede, o sinal de entrada se propaga para frente, camada por camada até que um conjunto de saídas seja obtido na última camada. Na primeira camada escondida, multiplica-se cada unidade de entrada por um peso correspondente a cada neurônio. O número de camadas ocultas e a quantidade de neurônios na camada oculta variam de acordo com a aplicação.

A camada de entrada é formada por padrões que serão apresentados à rede durante a fase de treinamento (também chamados de conjuntos sensoriais ou nós de fonte). Segundo Haykin (2004), a PMC tem três características distintas:

- O modelo de cada neurônio da rede inclui uma função de ativação não linear, sendo que esta função de ativação deve ser diferenciável em todos os pontos;
- A rede contém uma ou mais camadas de neurônios ocultos. Estes neurônios ocultos capacitam a rede a aprender tarefas complexas extraindo progressivamente as características mais significativas dos padrões de entrada;
- A rede exibe alto grau de conectividade, determinada por suas sinapses.

Os valores na entrada devem ser normalizados para valores restritos ao intervalo da função de ativação utilizada e, posteriormente, deve ser feito o processo inverso para que as variáveis estimadas voltem ao intervalo da função original.

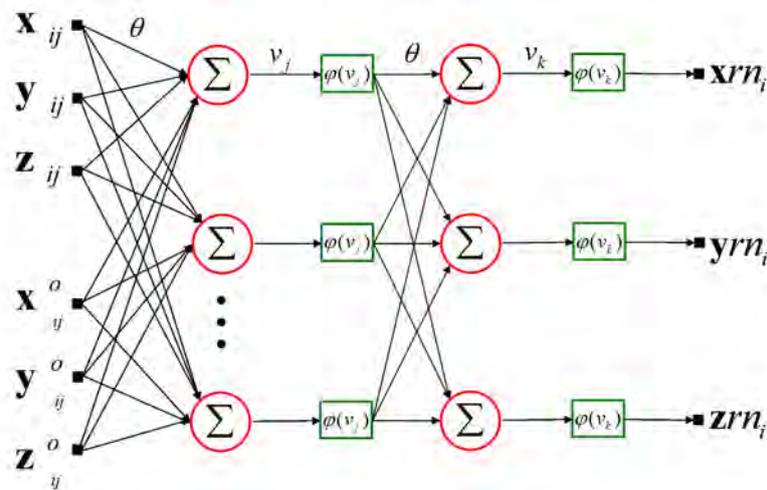


Figura 5.4 - Ilustração da Rede Perceptron de Múltiplas Camadas

## 5.6 Rede neural artificial auto-configurável

Salienta-se que no presente trabalho as configurações da rede neural foram determinadas de maneira empírica. O enfoque atual desta pesquisa é a aplicação da

ferramenta de redes neurais artificiais para o problema de assimilação de dados. Mais especificamente, foram realizados diversos experimentos numéricos envolvendo o ajuste dos seguintes parâmetros:

- o número de camadas escondidas;
- a quantidade de neurônios em cada camada;
- a taxa de aprendizagem  $\eta$ ;
- a constante de momento  $\alpha$ ;
- a função de ativação

O processo empírico é um procedimento padrão para encontrar uma boa arquitetura para uma rede neural artificial. Algumas topologias preliminares são definidas e testadas. A desvantagem desse procedimento é o período de tempo gasto para conseguir uma arquitetura capaz de produzir o resultado esperado. Isto exige um esforço contínuo por parte do especialista (desenvolvedor da RNA).

Uma solução alternativa é formular o problema de identificação de uma arquitetura ótima de RNA como um problema de otimização. Assim, a formulação para uma rede auto-configurável tem como finalidade determinar o melhor conjunto de parâmetros da RNA que otimiza uma função objetivo (TEIXEIRA et al., 2000; COSTA et al., 2003; ROCHA, 2012).

Em Sambatti et al. (2012), utilizou-se o método de colisão de partículas (MPCA-Multi-Particle Collision Algorithm), desenvolvido por Luz (2012), para encontrar a solução ótima. A função objetivo empregada é dada por:

$$f_{obj} = penalty \times \left( \frac{\rho_1 \times E_{train} + \rho_2 \times E_{gen}}{\rho_1 + \rho_2} \right) \quad (5.12)$$

em que  $\rho_1$  e  $\rho_2$  são termos que modificam a relevância atribuída ao erro de generalização e de treinamento. Essa formulação foi desenvolvida por Carvalho (2011). Para o caso em que  $\rho_1 > \rho_2$ , é dado maior importância para a capacidade da rede em memorizar os padrões apresentados, caso contrário, a ênfase será dada para a capacidade de generalização, isto é, identificar padrões similares, mas não idênticos

aos apresentados durante a fase de treinamento. O termo *penalty* prioriza arquiteturas menores, evitando o *overfitting*. O *overfitting* ocorre quando a rede memoriza os padrões de treinamento, ao invés de, aprender/extrair informações.

O termo  $E_{train}$  é o erro de treinamento. Esse termo é importante na medida que nos fornece uma indicação quantitativa do nível de treinamento da rede. O termo  $E_{gen}$  é o erro de generalização. Este critério se refere a à capacidade da RNA de identificar e responder a padrões que são similares mas não idênticos aos padrões com os quais a rede foi treinada. Ele revela-se mais importante que o erro de treinamento como parâmetro indicativo da performance do modelo de RNA na maioria das aplicações. O fator de penalidade *penalty* é expresso por:

$$penalty = c_1 e^{x^2} + c_2 y + 1 \quad (5.13)$$

em que  $x$  é o número mínimo de neurônios,  $y$  corresponde ao número de épocas de treinamento. Essa abordagem para determinar uma arquitetura ótima de uma RNA foi avaliada no modelo de propagação de onda em um contexto de assimilação de dados em Sambatti et al. (2012) e comparados com os resultados preliminares de Furtado et al. (2011).

## 6 ASSIMILAÇÃO DE DADOS COM REDES NEURAIAS

As técnicas apresentadas nos Capítulos 3, 4 e 5 foram avaliadas na equação da onda em uma dimensão e no modelo de água rasa em duas dimensões descritos no Capítulo 2. Neste Capítulo, apresenta-se os resultados e discussões dessas metodologias como uma técnica de assimilação de dados.

### 6.1 Redes neurais emulando o Filtro de Kalman

A teoria da estimação abrange recursos usados para estimar o estado de um sistema pela combinação de todo o conhecimento disponível sobre ele, incluindo as medidas, que são obtidas a partir de experimentos e modelos teóricos. Essa combinação de informações usualmente é feita por meio de aproximações estatísticas. A hipótese *a priori* e o critério de estimação são cruciais no processo de previsão, uma vez que eles influenciam na dinâmica do sistema e dos dados observados, isto é, precisa-se conhecer as hipóteses estatísticas *a priori* para o ruído do modelo e os erros das observações.

Em outras palavras, a teoria da estimação deseja conhecer a função densidade de probabilidade (*fdp*) *a posteriori*, isto é, a *fdp* condicional das variáveis de estado do sistema dado as observações. Na prática, é difícil determinar a *fdp* completa, portanto, pode-se tentar estimar algum parâmetro da *fdp* que pode servir como uma estimativa melhorada da variável de estado. Vários princípios conduzem a alguns estimadores. O estimador de Mínima Variância (MV), por exemplo, que é a média condicional das variáveis de estado, considerando as observações.

O método desenvolvido por Kalman (1960), o denominado Filtro de Kalman (FK), é uma ferramenta importante na teoria de estimação, que foi aplicado em modelos lineares discretos com observações linearmente relacionadas ao estado do modelo. O FK é uma simplificação da estimação Bayesiana para o caso de sistemas lineares. Para modelos lineares com estatística gaussiana, este método fornece a solução ótima para o problema da estimação da variável de estado do modelo. A versão do método para tempo contínuo foi desenvolvido por Kalman e Bucy (1961).

A técnica de Filtros de Kalman tem sido considerada para aplicação em meteorologia e oceanografia, mas é geralmente considerada muito cara para implementações operacionais devido a alta dimensão do problema (GHIL; MALANOTTE-RIZZOLI, 1991). Isto ocorre devido as operações com matrizes existentes no método. O problema

principal é a evolução da matriz de covariância. Se o vetor de variáveis de estado é de dimensão  $10^7$ , então a matriz de covariância tem  $10^{14}$  elementos e a propagação dessa matriz no tempo é impossível (POLAVARAPU, 2004). Portanto, na implementação prática do método, geralmente requer alguma simplificação das equações de evolução da covariância do erro de previsão ou análise.

Muitas técnicas alternativas foram propostas usando o método de Filtro de Kalman, por exemplo, o Filtro de Kalman por conjunto denominado *Ensemble Kalman Filter* (EnKF). O EnKF é um método sequencial de assimilação de dados proposto por Evensen (1994).

Nas seções a seguir, apresentam-se os resultados do FK e do PMC emulando o FK, aplicado à equação da onda linear e ao modelo de água rasa em duas dimensões.

### 6.1.1 Equação da onda 1D

Nesta seção, são apresentados os resultados para o modelo de uma dimensão, a equação da onda linear. O modelo foi integrado com o método de Crank Nicholson (LYNCH, 2004) com derivada espacial de quarta ordem. As observações assimiladas neste trabalho foram geradas artificialmente. Os dados foram gerados a partir da integração do modelo, adicionando-se um ruído aleatório de variância 0,5. Assim, a curva de referência de *verdade* para o método de assimilação é a curva obtida a partir da integração da equação da onda sem ruído, o que significa que quanto mais próximo as estimativas obtidas com os métodos de assimilação estiverem da dinâmica do sistema, melhor será a estimativa obtida por eles.

Para a implementação do Filtro de Kalman, foram usadas as seguintes matrizes:  $\mathbf{Q}_t = 0,1\mathbf{I}$ ;  $\mathbf{R}_t = 0,5\mathbf{I}$ ;  $\mathbf{H} = \mathbf{I}$ ; matriz de covariância do erro de modelagem, matriz de covariância do erro de observação e o operador que representa o sistema de observação, respectivamente. A matriz de covariância do erro de previsão é inicializada por:

$$\mathbf{P}_0^f = \begin{cases} 10(x_0^f)_i^2 & \text{para } i = j, \\ 0 & \text{para } i \neq j. \end{cases} \quad (6.1)$$

A matriz de dinâmica do sistema  $\mathbf{F}$ , ilustrada na Figura 6.1, para o modelo de uma dimensão é de dimensão  $128 \times 128$ , pois foram usados 128 pontos na discretização espacial.

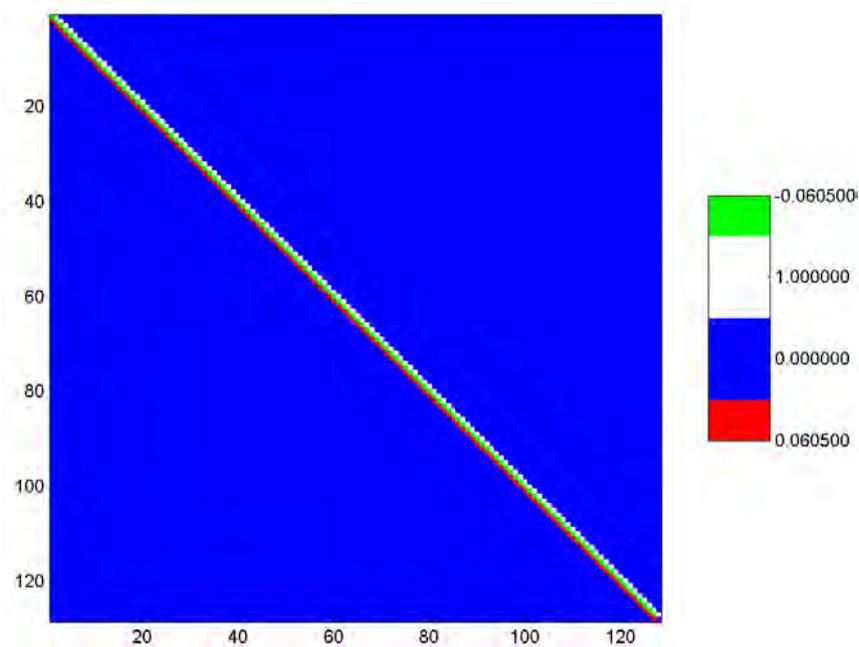


Figura 6.1 - Ilustração da matriz de dinâmica usada no Filtro de Kalman para o modelo 1D.

As Figuras 6.2, 6.3, 6.4 e 6.5 apresentam os resultados para a assimilação realizada com o Filtro de Kalman (FK). As observações foram inseridas a cada 10 passos de tempo. Na Figura 6.2, tem-se a amplitude da onda no tempo  $t = 200$ . Na Figura 6.3, a amplitude em todos os tempos de integração do modelo, a condição inicial usada para integração do modelo é dada pela Equação 2.2, que foi definido como experimento 1.

As Figuras 6.4 e 6.5 mostram os resultados para o experimento 2, em que utilizou-se uma função seno como condição inicial. Na Figura 6.4, ilustra-se a amplitude da onda no tempo  $t = 500$ , as observações foram inseridas a cada 20 passos de tempo. Na Figura 6.5, tem-se a amplitude da onda em todos os tempos.

Para as Figuras 6.2 e 6.4, a curva azul representa a referência de *verdade* e a curva vermelha representa o estimado por meio do Filtro de Kalman. Considera-se como *verdade* a integração do modelo sem ruído.

Este trabalho apresenta a técnica de Redes Neurais Artificiais, apresentada no Capítulo 5, como uma ferramenta para assimilação de dados. O funcionamento desta arquitetura de rede é realizado por meio de dois passos que são: o treinamento e a

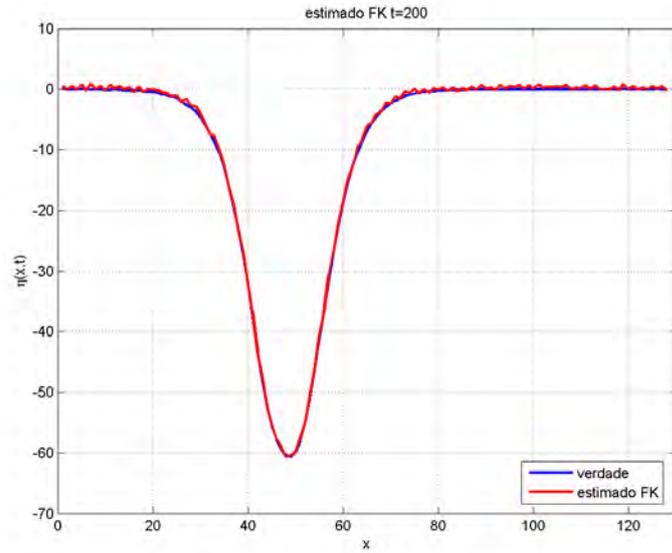


Figura 6.2 - Assimilação de dados realizada a cada 10 passos de tempo com o Filtro de Kalman. Amplitude da onda no tempo  $t = 200$ . Experimento 1: condição inicial usada um sóliton KdV.

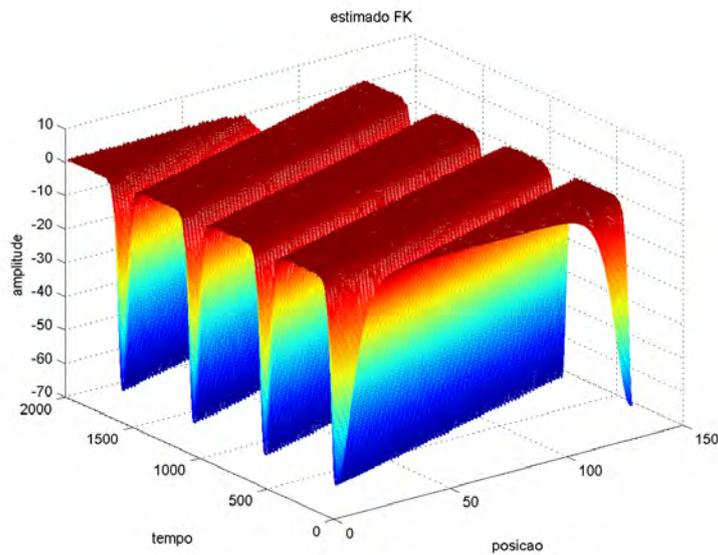


Figura 6.3 - Assimilação de dados a cada 10 passos de tempo com o Filtro de Kalman. Amplitude da onda em todos os tempos. Experimento 1: condição inicial usada um sóliton KdV.

ativação. Na fase de treinamento, apresentam-se os conjuntos de treinamentos, que são os dados de entrada e os dados de saída desejada. Neste caso, o conjunto de

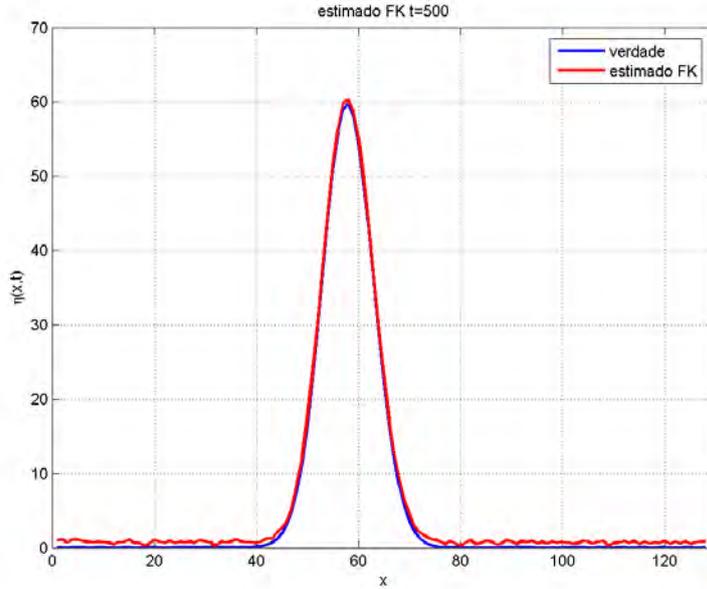


Figura 6.4 - Assimilação de dados realizada a cada 20 passos de tempo com o Filtro de Kalman. Amplitude da onda no tempo  $t = 500$ . Experimento 2: função seno como condição inicial.

dados de saída desejada é composto pela estimativa obtida por meio do Filtro de Kalman. A Figura 6.6 mostra o gráfico do erro de treinamento da RNA. A Figura 6.6(a) curva do erro de treinamento para o experimento 1 e a Figura 6.6(b) a curva do erro de treinamento para o experimento 2.

Matematicamente, o processo de assimilação de dados baseada em redes neurais artificiais (RNA) por meio de Perceptron de Múltiplas camadas descrito na Seção 5.5 do Capítulo 5 é dado por:

$$x_t^a = F_{RNA/PMC}(x_t^p, y_t^o) \tag{6.2}$$

em que  $x_t^a$  representa a análise calculada,  $F_{RNA/PMC}$  representa o processo de assimilação por RNA/PMC,  $x_t^p$  representa as variáveis de estado do modelo físico-matemático e  $y_t^o$  representa o vetor de observações.

A arquitetura PMC exige que o seu treinamento seja supervisionado. Assim, precisa-se um outro conjunto de entrada, representado aqui por  $x_t^{as}$  que é a análise obtida por uma outra técnica de assimilação de dados. Nesta seção,  $x_t^{as}$  é obtida com o método do Filtro de Kalman.

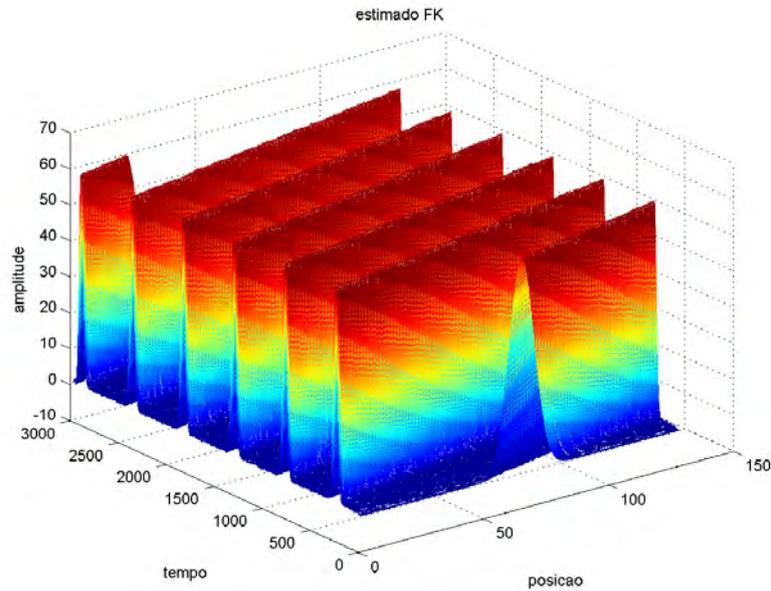
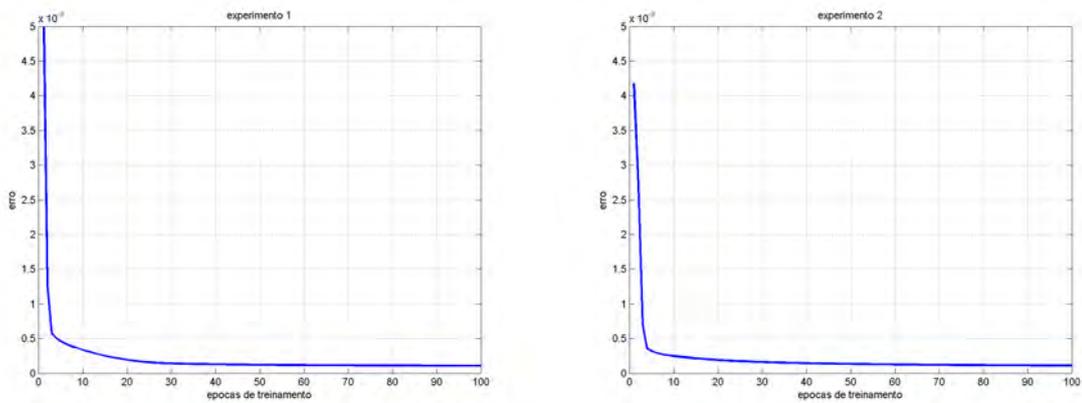


Figura 6.5 - Assimilação de dados realizada a cada 20 passos de tempo com o Filtro de Kalman. Amplitude da onda em todos os tempos nas coordenadas  $(x, y, z)$ . Experimento 2: função seno como condição inicial.



(a) condição inicial dada pela equação KdV (2.2)    (b) condição inicial dada por uma gaussiana.

Figura 6.6 - Curvas do erro de treinamento para os dados de treinamento para o modelo 1D.

A Figura 6.7 ilustra a arquitetura da rede utilizada para o cálculo da análise. Essa arquitetura é constituída por dois neurônios na camada de entrada, três na camada intermediária (oculta) e um neurônio na camada de saída. Para a camada de entrada, tem-se que  $\eta^m$  é o dado do modelo,  $\eta^o$  é o dado observado e  $\eta^a$  é o dado de análise ou

condição inicial, o parâmetro que é estimado para reiniciar a integração do modelo. A taxa de aprendizagem utilizada foi 0,9 e a função de ativação foi a tangente hiperbólica. Esses parâmetros foram encontrados de modo empírico e usados no experimento 1 e 2 para o modelo de uma dimensão.

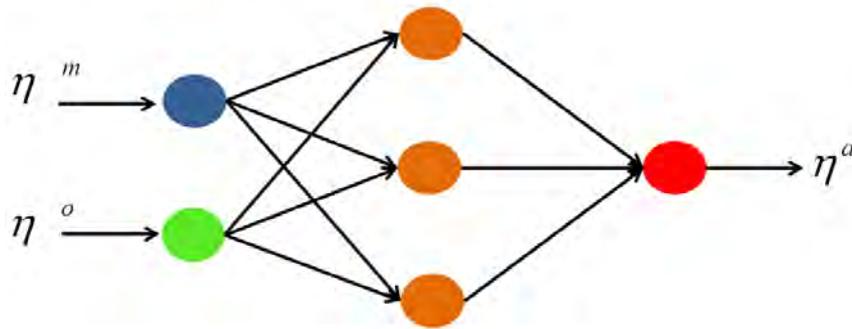


Figura 6.7 - Arquitetura da RNA empregada para o modelo 1D.

Para o treinamento da rede, utilizou-se o algoritmo de aprendizagem de retropropagação do erro. Esse é um processo supervisionado, em que a rede recebe os vetores de entrada com as suas respectivas respostas desejadas. No processo de treinamento, determinam-se os pesos sinápticos ótimos que mapeiam os dados de entrada aos dados de saída desejados.

A Figura 6.8 ilustra como foram selecionados os dados para o treinamento e generalização da rede para o experimento 1 do modelo de uma dimensão. O modelo foi integrado em 2000 passos de tempo com 128 pontos na discretização espacial. Utilizou-se os dados integrados até o passo de tempo ( $nk = 100$ ) para o treinamento. As observações foram inseridas a cada 10 passos de tempo. Os pontos verdes representam as observações inseridas no ponto de grade do modelo. Os dados a partir de  $nk = 101$  até  $nk = 2000$  foram usados para generalização da rede conforme mostra a Figura 6.8.

Os pesos sinápticos ótimos são obtidos após a fase de treinamento. Em seguida, ativam-se os neurônios da rede com novos dados de entrada que não foram usados no treinamento verificando a capacidade de generalização da informação de uma rede neural artificial. Na fase de treinamento de uma RNA, há a atualização dos pesos em um sentido de mínimos quadrados. Na fase de ativação, os pesos sinápticos são fixos, uma vez que, o peso ótimo já foi determinado na fase anterior.

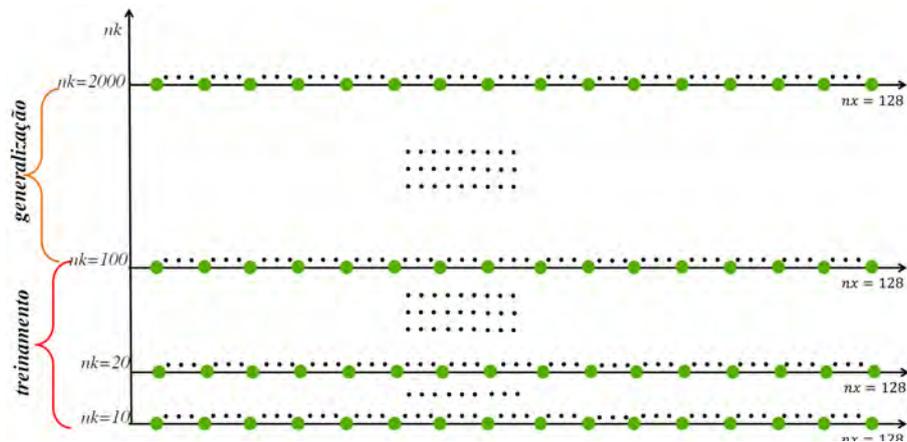


Figura 6.8 - Conjunto de dados de treinamento para o experimento 1 do modelo em uma dimensão.

Para o modelo de uma dimensão, também foi realizado um outro teste com um outro conjunto de dados de treinamento. O modelo foi integrado em 3000 passos de tempo. A condição inicial usada para integração do modelo foi uma função gaussiana. A assimilação foi realizada a cada 20 passos de tempo. O conjunto de dados usados na fase de treinamento fora selecionados conforme ilustrado na Figura 6.9.

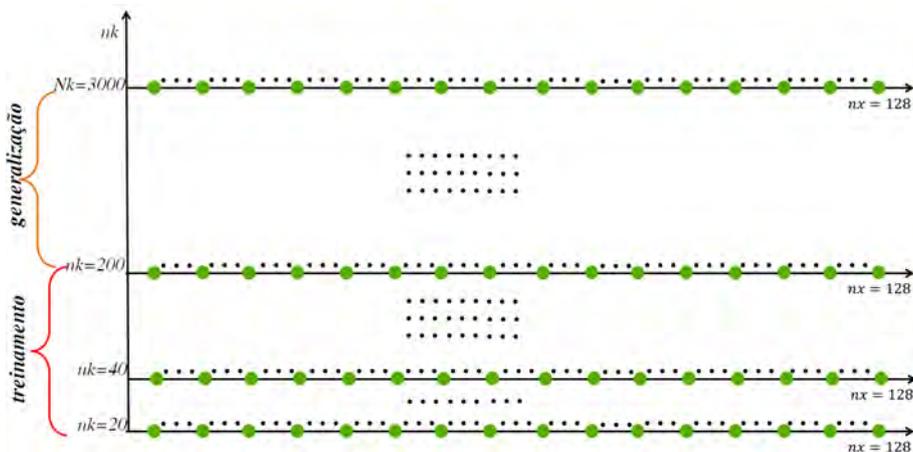


Figura 6.9 - Conjunto de dados de treinamento para o experimento 2 do modelo em uma dimensão.

As Figuras 6.10, 6.11, 6.12 e 6.13 apresentam os resultados de assimilação de dados por meio de RNA. Para as Figuras 6.10 e 6.11, usou-se a condição inicial dada por

um s3liton KdV, representada pela Equa3o3o 2.2, que denominou-se de experimento 1. Para as Figuras 6.10 e 6.12, a curva azul representa a refer4ncia de *verdade* e a curva verde representa o estimado por meio de RNA no tempo  $t = 200$ . Na Figura 6.11, tem-se a amplitude da onda em todos os tempos.

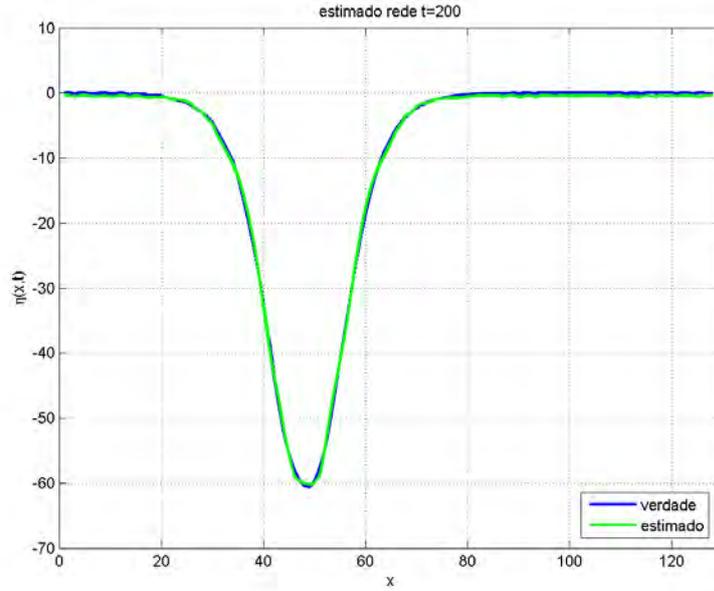


Figura 6.10 - Assimila3o3o de dados realizada a cada 10 passos de tempo com RNA. Amplitude da onda no tempo  $t = 200$ . Experimento 1: condi3o3o inicial usada um s3liton KdV.

A Tabela 6.1 apresenta os resultados da m4dia do erro de assimila3o3o. O erro 4 calculado pela diferen3a absoluta entre o valor estimado ( $\phi_i^{est}$ ) e a refer4ncia de *verdade* ( $\phi_i^{verd}$ ) de acordo com a Equa3o3o 6.3. Matematicamente, o erro 4 dado por:

$$\epsilon = \int_0^{T_{final}} \int_{\Omega} |\phi_i^{est} - \phi_i^{verd}|$$

em que  $\Omega \in [0, L_x]$ .

Tabela 6.1 - Resultados do erro de assimila3o3o para FK e o PMC para o modelo 1D.

	FK	RNA-PMC
<i>experimento1</i>	0,5331	0,1171
<i>experimento2</i>	0,5275	0,1740

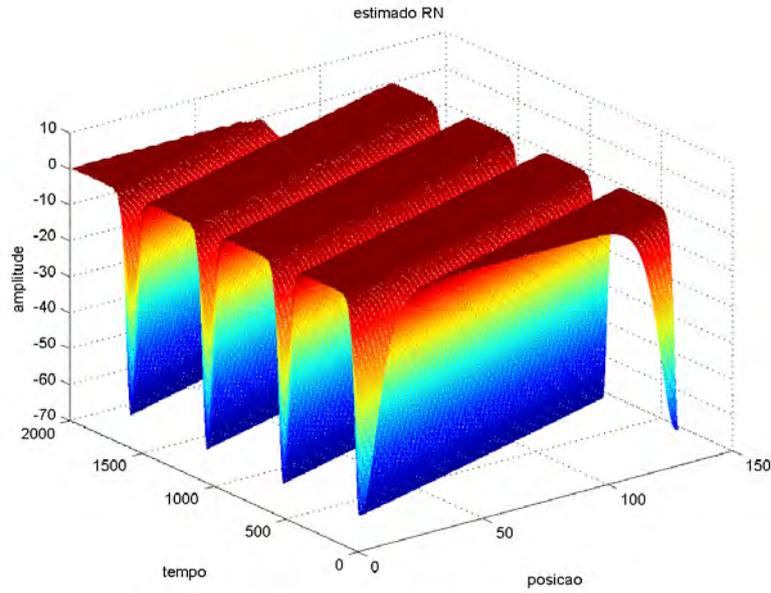


Figura 6.11 - Assimilação de dados realizada a cada 10 passos de tempo com RNA. Amplitude da onda em todos os tempos. Experimento 1: condição inicial usada um sóliton KdV.

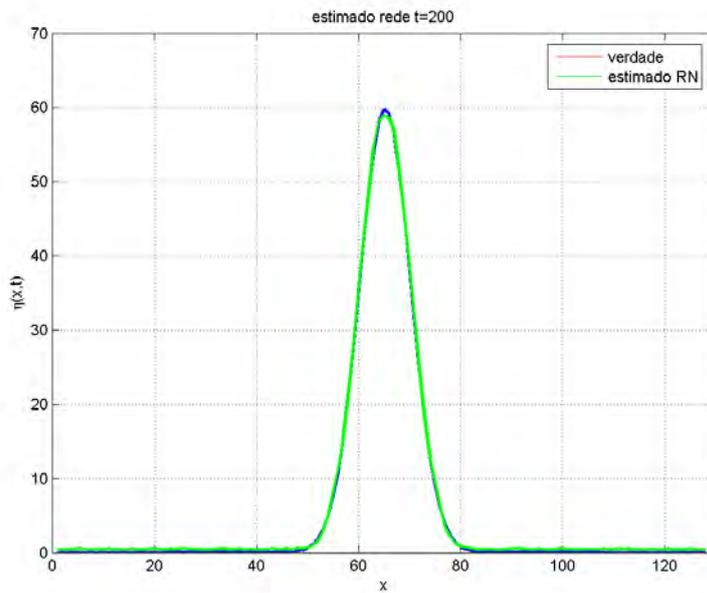


Figura 6.12 - Assimilação de dados realizada a cada 20 passos de tempo com RNA. Amplitude da onda no tempo  $t = 200$ .

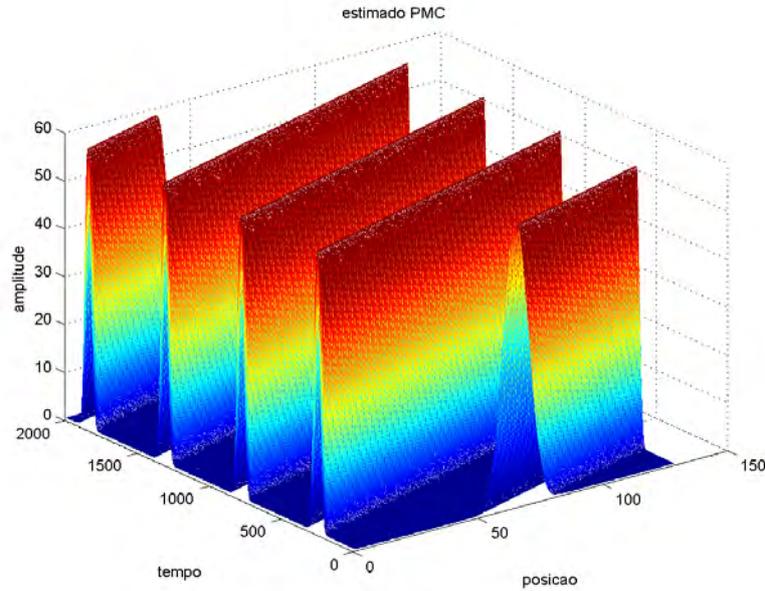


Figura 6.13 - Assimilação de dados realizada a cada 20 passos de tempo com RNA. Amplitude da onda em todos os tempos.

Na Tabela 6.2, apresenta-se o custo computacional para ambos os experimentos. Na fase de treinamento da Rede Neural Artificial para o modelo 1D, foram usados os dados do modelo integrado em 1000 passos de tempo. A ativação foi realizada em 2000 passos de tempo adiante.

Tabela 6.2 - Custo computacional para FK e o PMC para o modelo 1D.

	FK	RNA-PMC
<i>experimento1</i>	29,61 s	12,48 s
<i>experimento2</i>	44,56 s	19,51 s

Na Subseção 6.1.2, apresentam-se os resultados usando as mesmas metodologias apresentadas nesta seção, aplicadas no modelo de água rasa em duas dimensões.

### 6.1.2 Modelo de água rasa linear 2D

Nesta seção, apresentam-se os resultados para o modelo de água rasa linear em duas dimensões. As técnicas de assimilação de dados utilizadas foram o Filtro de Kalman e a técnica de Redes Neurais Artificiais emulando o Filtro de Kalman. Esse

modelo foi descrito na [Seção 2.2](#). As equações de água rasa são frequentemente usadas para testar problemas em meteorologia e oceanografia, pois descrevem características presentes no fluxo da atmosfera e oceano.

Para avaliar a capacidade dos métodos em realizar assimilação de dados para o modelo de água rasa em duas dimensões, foram realizados dois experimentos para o Filtro de Kalman e a RNA emulando o FK. O experimento **A** para o modelo 2D caracteriza-se por:

- inserção de observações realizadas a cada 10 passos de tempo;
- o modelo foi integrado em 60 passos de tempo;
- a grade do modelo possui dimensão  $40 \times 40$ ;
- foram utilizadas 25 observações inseridas no ponto de grade do modelo numérico;
- a variável  $q$  foi inicializada com uma função gaussiana;
- as variáveis  $u$  e  $v$  tiveram valor zero na condição inicial;

O experimento **B** diferenciou-se nos seguintes aspectos:

- todas as variáveis foram inicializadas com uma função gaussiana;
- o modelo foi integrado em 100 passos de tempo;
- 100 observações foram assimiladas no ponto de grade do modelo numérico em cada 10 passos de tempo;

A [Figura 6.14](#) ilustra a inserção de observação no ponto de grade para ambos os experimentos. A Figura da esquerda mostra as 25 observações assimiladas no experimento **A** e a Figura da direita mostra as 100 observações assimiladas no ponto de grade para o experimento **B**. As observações estão representadas pelos quadrados amarelos.

Para realizar assimilação de dados com o método Filtro de Kalman, o primeiro passo a ser feito é determinar a matriz de dinâmica do sistema  $\mathbf{F}$  que depende do modelo

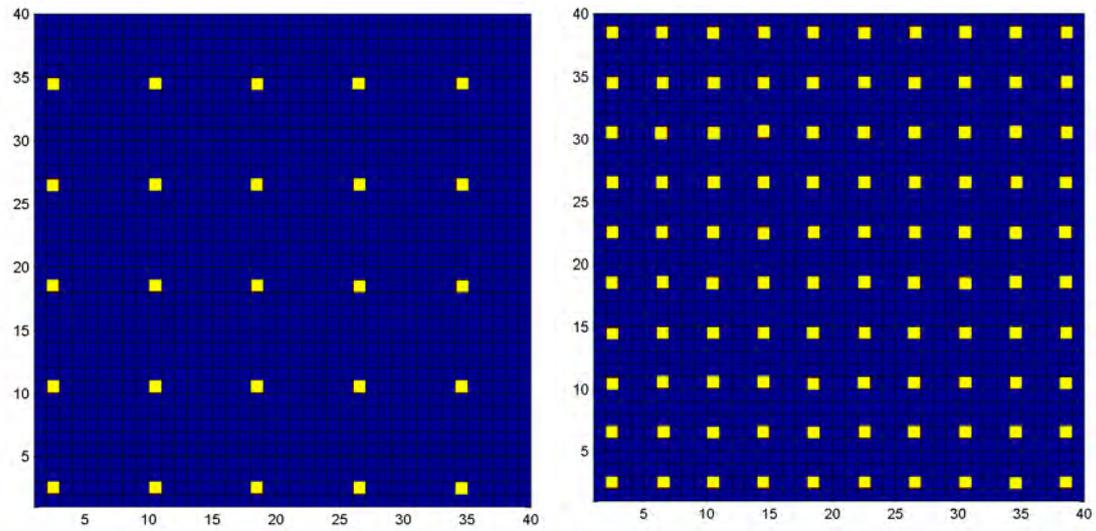


Figura 6.14 - (lado esquerdo) experimento **A**; (lado direito) experimento **B**. Os quadrados amarelos representam as observações no ponto de grade para o modelo 2D.

que está sendo usado. Para o modelo de água rasa linear em duas dimensões e com três variáveis do modelo tem-se:

$$\Psi^{n+1} = \mathbf{F}\Psi^n$$

em que  $\Psi$  é dado por:

$$\Psi = \begin{bmatrix} \mathbf{Q} \\ \mathbf{U} \\ \mathbf{V} \end{bmatrix}$$

e cada variável do modelo  $q$ ,  $u$  e  $v$  é um vetor dado por:

$$\mathbf{Q} = \begin{bmatrix} \begin{bmatrix} q_{11} \\ q_{21} \\ \vdots \\ q_{nx,1} \end{bmatrix} \\ \begin{bmatrix} q_{12} \\ q_{22} \\ \vdots \\ q_{nx,2} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} q_{nx,1} \\ q_{nx,2} \\ \vdots \\ q_{nx,ny} \end{bmatrix} \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} \begin{bmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{nx,1} \end{bmatrix} \\ \begin{bmatrix} u_{12} \\ u_{22} \\ \vdots \\ u_{nx,2} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} u_{nx,1} \\ u_{nx,2} \\ \vdots \\ u_{nx,ny} \end{bmatrix} \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \\ \vdots \\ v_{nx,1} \end{bmatrix} \\ \begin{bmatrix} v_{12} \\ v_{22} \\ \vdots \\ v_{nx,2} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} v_{nx,1} \\ v_{nx,2} \\ \vdots \\ v_{nx,ny} \end{bmatrix} \end{bmatrix}$$

A matriz de dinâmica  $\mathbf{F}$  é definida por meio das equações do modelo de água rasa. Determina-se a matriz  $\mathbf{F}$  de acordo com os coeficientes das equações do modelo discretizadas. As equações discretizadas encontram-se na [Subseção 2.2.1](#). A matriz  $\mathbf{F}$  é dada por:

$$\mathbf{F} = \begin{bmatrix} (1 - \Delta tr_q)\mathbf{I} & \mathbf{A}_1 & -\mathbf{B}_1 \\ -\mathbf{A}_1 & (1 + \Delta tr_u)\mathbf{I} & \mathbf{B}_2 \\ \mathbf{B}_1 & -\mathbf{B}_2 & (1 + \Delta tr_v)\mathbf{I} \end{bmatrix} \quad (6.3)$$

em que:

$$\begin{aligned} (\mathbf{A}_1)_{ii} &= H \frac{\Delta t}{\Delta x} \\ (\mathbf{A}_1)_{i,i+3} &= H \frac{\Delta t}{\Delta x} \\ (\mathbf{B}_1)_{ii} &= -(\mathbf{B}_1)_{i,i+1} = g \frac{\Delta t}{\Delta y} \\ (\mathbf{B}_2)_{ii} &= -(\mathbf{B}_2)_{i,i+1} = f \frac{\Delta t}{4} \end{aligned}$$

Sendo que  $\Delta x$  e  $\Delta y$  são os tamanhos da malha para a discretização espacial,  $\Delta t$  é o espaçamento temporal;  $r_q$ ,  $r_u$  e  $r_v$  são coeficientes de amortecimento;  $H$  é a profundidade média do oceano;  $f$  é o parâmetro de Coriolis e  $g$  é a aceleração da

gravidade. Os valores desses parâmetros estão definidos na [Tabela 2.2](#). Assim, é possível estimar as variáveis de estado do modelo  $u$ ,  $v$  e  $q$  no instante  $n + 1$  de acordo com:

$$\Psi^{n+1} = \mathbf{F}\Psi^n$$

A dimensão da matriz [6.3](#) dependerá do número de pontos usados para a discretização espacial. Pode-se perceber que o método de assimilação por meio do Filtro de Kalman é muito custoso computacionalmente. Além da matriz de dinâmica do modelo  $\mathbf{F}$ , tem-se as matrizes de covariância do modelo e das observações que são atualizadas em cada passo de tempo. As operações com matrizes, multiplicação e cálculo da inversa de matrizes, realizadas para determinar a matriz de ganho de Kalman tornam esta ferramenta pesada ao se aplicar em modelos de grande dimensão. Essas dificuldades estimulam a investigação da ferramenta de RNA como um método de assimilação de dados.

Nesta Seção, compara-se o desempenho da assimilação de dados realizada com o Filtro de Kalman e com a RNA no modelo de água rasa 2D. A [Figura 6.15](#) ilustra como foram selecionados os dados para o treinamento da rede para o experimento **A**. Integrou-se o modelo em 60 passos de tempo. Para o treinamento da rede utilizou-se os dados até o passo 40 e para a generalização do passo 41 em diante.

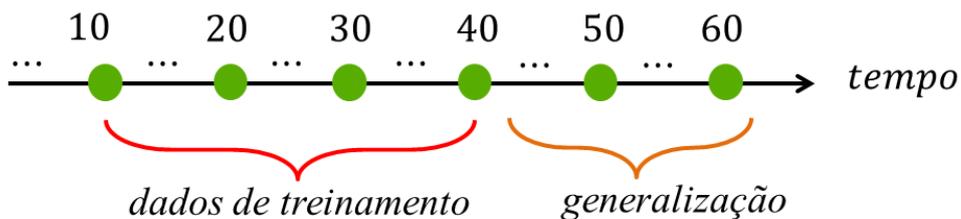


Figura 6.15 - Conjunto de dados para o treinamento da rede para o experimento **A** do modelo 2D.

A [Figura 6.16](#) ilustra como foram selecionados os dados para o treinamento da rede para o experimento **B**. O modelo foi integrado em 100 passos de tempo. Para o treinamento da rede utilizou-se os dados até o passo 40 e para a generalização do passo 41 em diante. Nesse experimento, verifica-se que a rede foi capaz de obter uma

estimativa satisfatória até o passo de tempo  $nk = 100$ . Aqui 100 observações foram assimiladas a cada 10 passos de tempo. A inserção da observação está representada pelo ponto verde ilustrado na Figura 6.16.

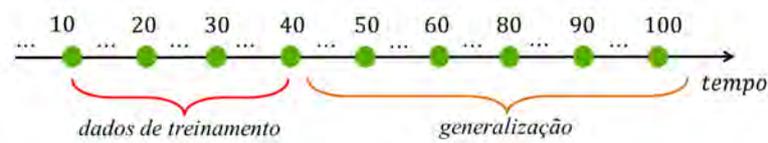


Figura 6.16 - Conjunto de dados para o treinamento da rede para o experimento **B** do modelo 2D.

Para o modelo de água rasa, utilizou-se uma rede neural para cada variável conforme ilustrado na Figura 6.17. Os dados de entrada da rede são os dados do modelo e o dado observado e a saída da rede é o dado de análise ou condição inicial.

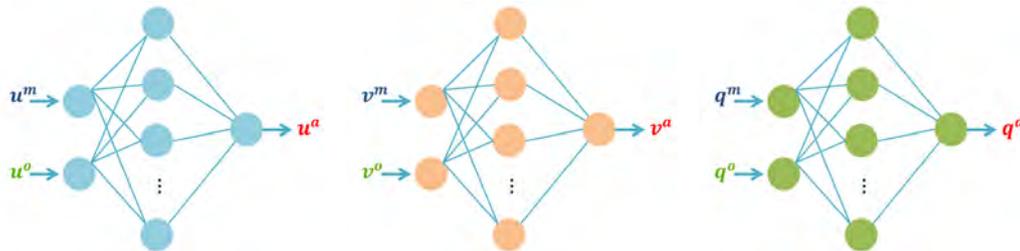


Figura 6.17 - Arquitetura das redes para as variáveis  $u$ ,  $v$  e  $q$ . Os sobrescritos  $m$ ,  $o$  e  $a$  significam modelo, observação e análise, respectivamente.

A Tabela 6.3 apresenta os parâmetros da rede para o experimento **A** e a Tabela 6.4 para o experimento **B** realizado no modelo 2D. O parâmetro  $nco$  indica o número de neurônios na camada oculta,  $\eta$  é a taxa de aprendizagem e  $a$  é o parâmetro da tangente hiperbólica dado pela Equação 5.5 no Capítulo 5, com as suas respectivas épocas de treinamento. As Tabelas 6.3 e 6.4 constam também o tempo computacional que foi necessário para se determinar os pesos ótimos. Esses pesos são usados na fase de ativação. Em ambos os experimentos a variável  $q$  foi a que obteve o menor tempo de treinamento.

Tabela 6.3 - Parâmetros da rede PMC para o experimento **A** do modelo 2D.

variável	$nco$	$\eta$	$a$	épocas	tempo
$u$	15	0,2	1	290	6,7 <i>min</i>
$v$	15	0,7	2	270	5,0 <i>min</i>
$q$	10	0,7	1	681	4,1 <i>min</i>

Tabela 6.4 - Parâmetros da rede PMC para o experimento **B** do modelo 2D.

variável	$nco$	$\eta$	$a$	épocas	tempo
$u$	10	0,2	1	1000	7,2 <i>min</i>
$v$	10	0,7	1	1000	6,8 <i>min</i>
$q$	10	0,7	1	630	4,4 <i>min</i>

As Tabelas 6.5 e 6.6 apresentam o erro para as variáveis  $u, v$  e  $q$  para a assimilação de dados realizada com o Filtro de Kalman e com a RNA, para os experimento **A** e **B**, respectivamente. Para ambos os experimentos e para todas a variáveis do modelo, a assimilação de dados realizada por meio de RNA obtiveram o menor erro de assimilação. Para estes experimentos aqui apresentados, constata-se a eficiência da assimimilação de dados por meio de redes neurais artificiais tanto na qualidade da assimilação quanto no tempo necessário para o cálculo da estimativa (ver Tabela 6.7). Estes resultados são animadores e estimulam o estudo dessa metodologia como um possível método operacional para assimilação de dados.

O erro de assimilação é calculado para cada variável do modelo de acordo com a equação a seguir:

$$\epsilon = \int_0^{T_{final}} \int_{\Omega} |\phi_i^{est} - \phi_i^{verd}|$$

em que:  $\Omega \in [0, L_x] \times [0, L_y]$ . Os sobrescritos *est* e *verd* significam estimado e verdade, respectivamente.

De acordo com a Tabela 6.7, para o experimento **A**, a RNA é aproximadamente 30 vezes mais rápida que o método usual do Filtro de Kalman e para o experimento **B** a RNA é aproximadamente 16 vezes mais rápida computacionalmente.

As Figuras 6.18 e 6.19 exibem a evolução temporal do ponto  $(x, y)$  para  $x = y = 7$

Tabela 6.5 - Resultados do erro de assimilação para FK e o PMC para o experimento **A** do modelo 2D.

variável	RNA-PMC	FK
$u$	0,0199	0,0296
$v$	0,7794	0,8554
$q$	0,1460	0,5969

Tabela 6.6 - Resultados do erro de assimilação para FK e o PMC para o experimento **B** do modelo 2D.

variável	RNA-PMC	FK
$u$	0,0211	0,0290
$v$	0,8441	0,8515
$q$	0,1971	0,8951

Tabela 6.7 - Custo computacional para FK e o PMC.

	FK	RNA-PMC
<i>experimento1</i>	42 <i>min</i>	1,39 <i>min</i>
<i>experimento2</i>	1 h e 19 <i>min</i>	5,00 <i>min</i>

para o experimento **A** e **B**, respectivamente. A curva azul representa a referência de *verdade*, a curva vermelha o estimado pelo Filtro de Kalman e a curva verde é a estimativa obtida por meio da RNA. Por meio dos gráficos, é possível ver que a estimativa obtida com a RNA (curva verde) acompanha a *verdade* (curva azul) de modo mais suave. A estimativa obtida com o Filtro de Kalman tem um comportamento mais oscilatório com relação a *verdade*, pois este método tende a seguir as observações.

As Figuras 6.20 e 6.21 mostram a projeção em três dimensões da variável  $q$ , para os passos de tempo 60 e 100, respectivamente. Essa estimativa foi realizada com a RNA.

Houve uma evolução significativa no emprego da metodologia de assimilação de dados por meio de redes neurais artificiais. O desenvolvimento no algoritmo consiste em determinar a análise em cada ponto de grade (HARTER, 2004; HÄRTER; CAMPOS

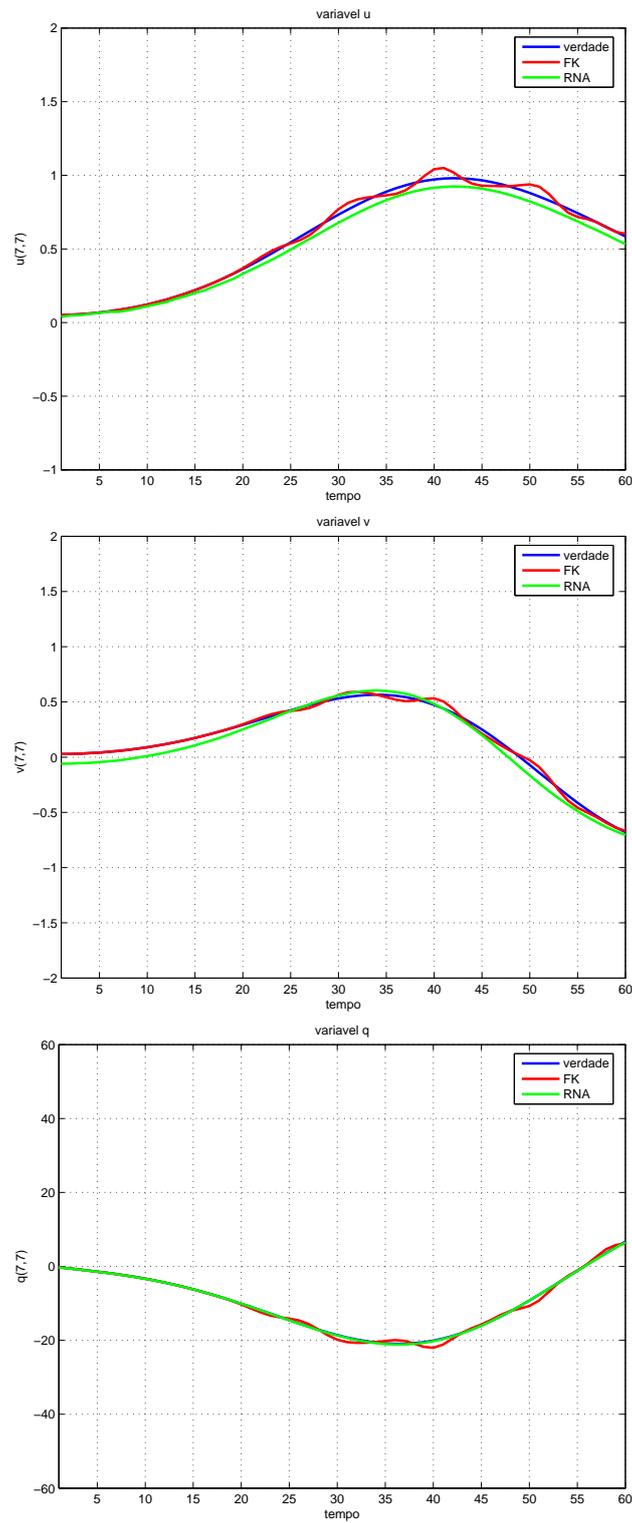


Figura 6.18 - Evolução temporal dos pontos  $u(7, 7)$ ,  $v(7, 7)$  e  $q(7, 7)$  experimento **A**.

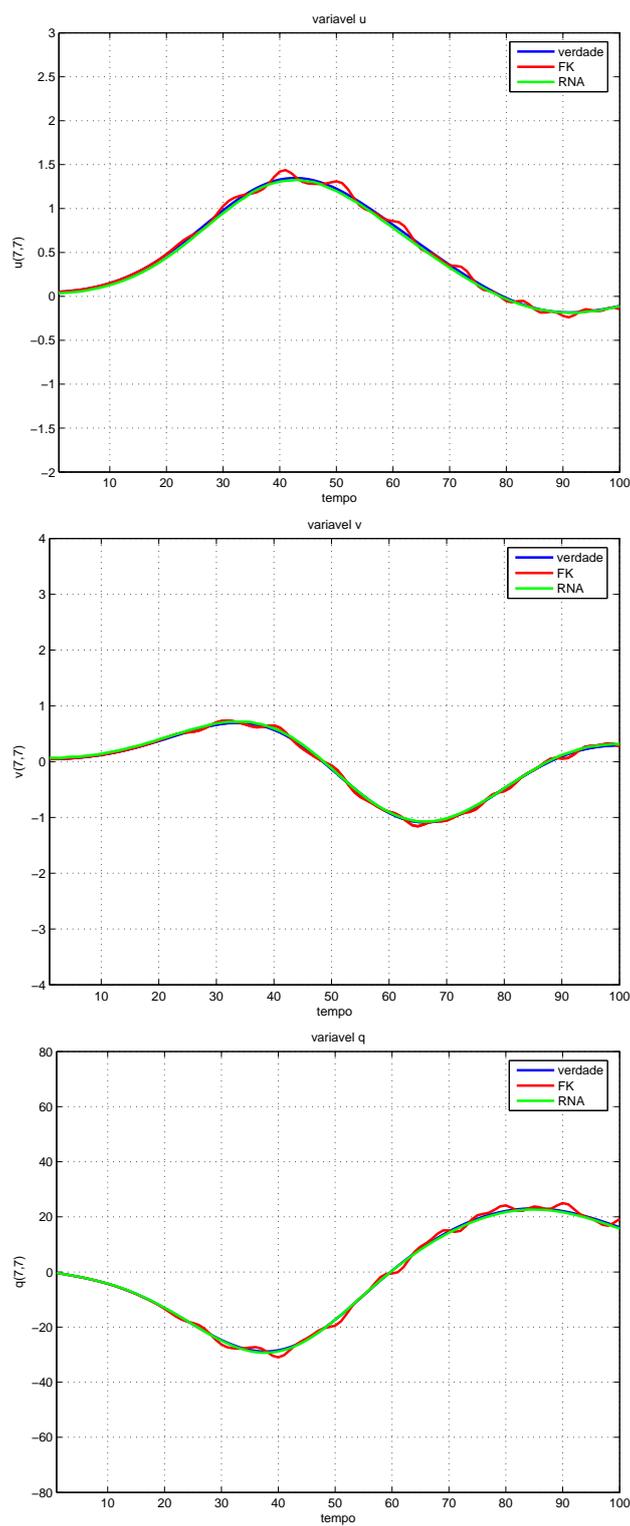


Figura 6.19 - Evolução temporal dos pontos  $u(7, 7)$ ,  $v(7, 7)$  e  $q(7, 7)$  experimento **B**.

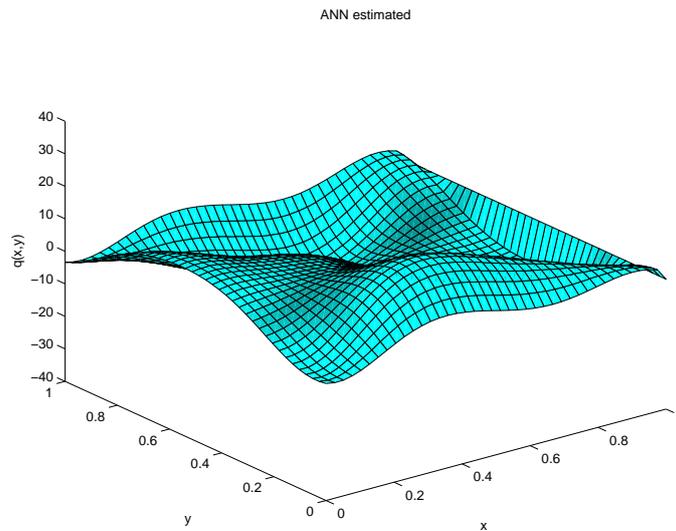


Figura 6.20 - Projeção da variável  $q$  em três dimensões no passo de tempo 60.

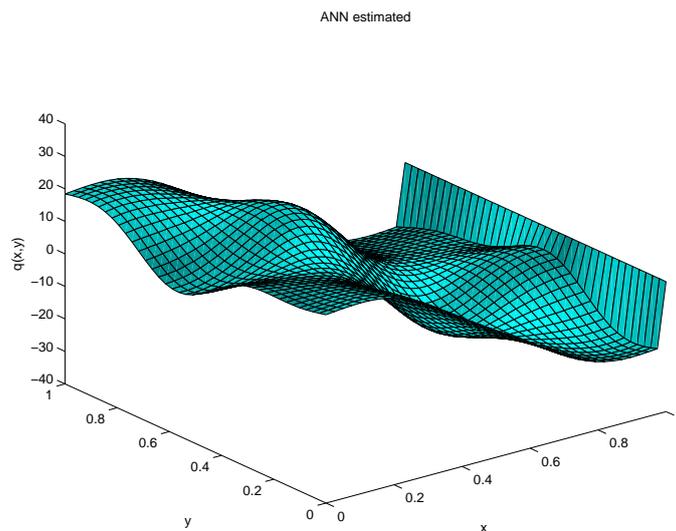


Figura 6.21 - Projeção da variável  $q$  em três dimensões no passo de tempo 100.

VELHO, 2012). No trabalho pioneiro de Nowosad (2001), a análise era determinada sobre todo o domínio. O ganho com a diminuição do espaço de busca foi um resultado significativo da pesquisa de Harter (2004), pois em problemas de minimização de gradiente em aplicações de grande dimensão, diminuir o espaço de busca pode ser o fator a tornar a bordagem operacionalmente viável.

O método do Filtro de Kalman e as RNA para o modelo de uma dimensão e o modelo

de água rasa 2D foram implementadas em matlab. Os programas foram executados em um processador Intel(R) core(TM) i5-2310 CPU@ 2.9 GHz com 8 GB de memória RAM. A rede neural artificial por meio de um Perceptron de Múltiplas Camadas (PMC) apresentou um menor erro de assimilação em todos os casos aqui avaliados. Mesmo considerando o tempo de treinamento mais o tempo de generalização da rede a técnica de RNA é computacionalmente mais rápido que o método do Filtro de Kalman.

## 6.2 Redes neurais emulando o Método do Representante

O método do Representante é uma abordagem da técnica variacional usada em modelos de circulação oceânica (CHUA; BENNETT, 2001; KALNAY, 2003). Essa metodologia foi apresentada no Capítulo 4. A comparação do método de redes neurais artificiais com o método do Representante é um resultado significativo para a validação da ferramenta de RNA no contexto de assimilação de dados, pois é uma metodologia voltada para aplicação em circulação oceânica.

A ferramenta de Redes Neurais Artificiais teve um bom desempenho comparado com o Método Variacional, Filtro de Kalman e Filtro de Partículas (FURTADO et al., 2008; FURTADO et al., 2011; FURTADO et al., 2011). Essa metodologia foi aplicada no sistema de Lorenz, que possui características semelhantes ao comportamento da atmosfera (LORENZ, 1963; LORENZ, 1965). Nesta seção, apresenta-se o resultado do método do Representante comparado com a RNA aplicado a equação da onda 1D e no modelo de água rasa em duas dimensões.

### 6.2.1 Equação da onda 1D

A equação da onda apresentada nesta seção foi integrada com o método FTCS (Forward-Time Central-Space). A condição inicial e a condição de contorno são periódicas. Os dados foram assimilados a cada 10 passos de tempo em 4 pontos da grade espacial. Os dados observados usados no processo de assimilação são dados sintéticos. Cada dado observado foi gerado a partir da integração do modelo mais um ruído adicionado com variância 0,04. A referência de *verdade* neste experimento é a curva obtida a partir integração do modelo sem ruído.

A arquitetura da RNA implementada para este experimento foi com uma camada oculta composta por três neurônios, duas entradas e um neurônio na camada de saída, de acordo com a Figura 6.7. O algoritmo de treinamento utilizado foi o de

retropropagação do erro (HAYKIN, 2004) descrito na Seção 5.4. A função de ativação usada neste experimento foi a tangente hiperbólica, a taxa de aprendizagem  $\eta = 0,7$ . A constante de momento  $\alpha = 0$ . A comparação dessas duas metodologias constam em Furtado et al. (2012).

A Figura 6.22 mostra a assimilação realizada com o método Representante (lado esquerdo) e a assimilação com a RNA (lado da direito) para o tempo  $t = 10$ . A condição inicial usada para a integração do modelo foi uma função seno. A Figura 6.23 mostra a comparação entre as duas metodologias. Pode-se observar que no ponto  $x = 12$  a estimativa obtida por meio da RNA teve uma melhor aproximação com a *verdade* que o método do Representante.

O experimento realizado com o método do representante aplicado no modelo de uma dimensão foi feito para que houvesse um entendimento da metodologia e avaliar se a RNA teria a capacidade de emular essa ferramenta. Tendo em vista que os resultados foram positivos o passo seguinte foi explorar essa abordagem em um modelo de duas dimensões que será exposto na Subseção 6.2.2 a seguir.

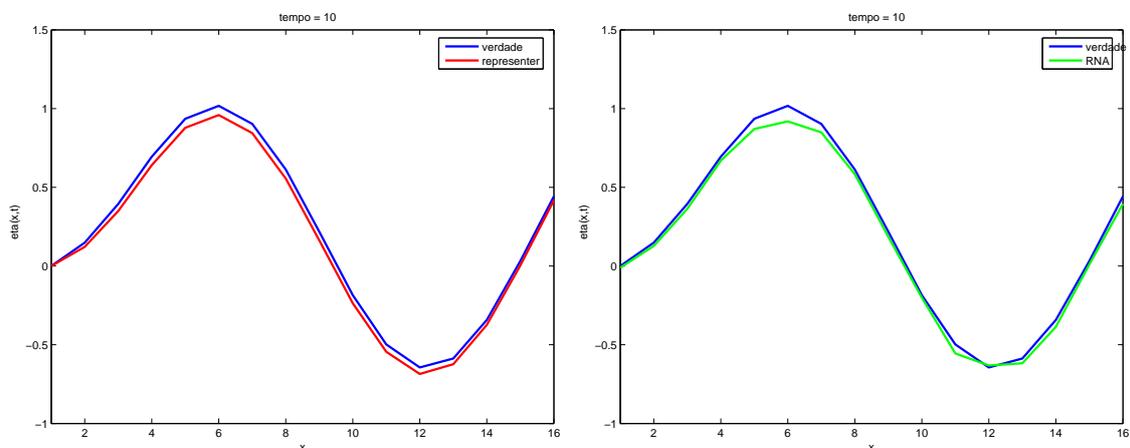


Figura 6.22 - (lado esquerdo) Assimilação com o Representante; (lado direito) Assimilação com a RNA. Curva azul: *verdade*; curva vermelha: estimado com o representante; curva verde: estimado com a RNA.

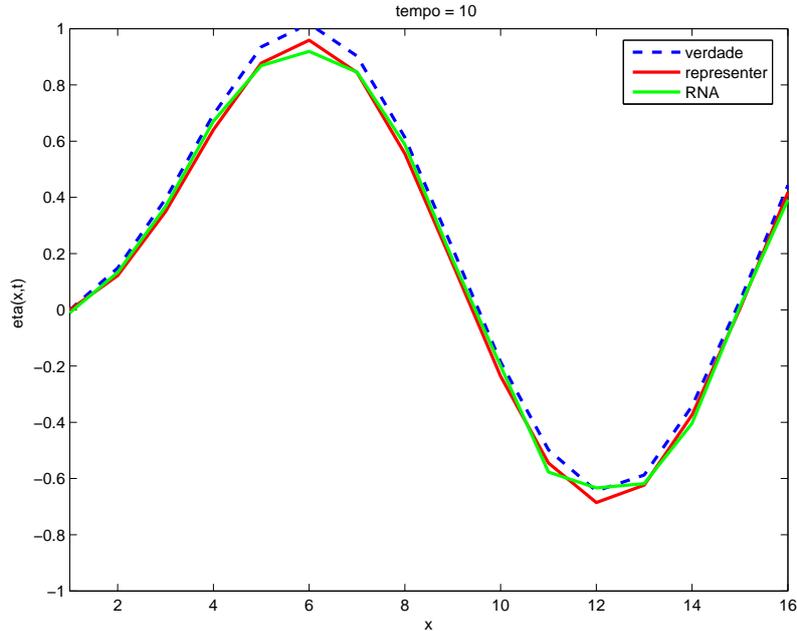


Figura 6.23 - Comparação entre o método Representer e a RNA. Curva azul: *verdade*; curva vermelha: estimado com o representer; curva verde: estimado com a RNA.

### 6.2.2 Modelo de água rasa 2D

Nesta seção, apresenta-se os resultados para o método do Representante, descrito no [Capítulo 4](#), e para a RNA, descrito no [Capítulo 5](#), emulando essa metodologia. Essas técnicas são testadas no modelo de água rasa em duas dimensões descrito na [Seção 2.2 do Capítulo 2](#). Ressalta-se que emular o método do Representante significa que a rede neural tem como saída desejada para o treinamento a estimativa obtida com essa metodologia.

Retomemos a formulação variacional para o problema de assimilação de dados, em que as equações do modelo são dadas por:

$$\frac{\partial u}{\partial t} - fv + g \frac{\partial q}{\partial x} + r_u u = F_u + f^u \quad (6.4a)$$

$$\frac{\partial v}{\partial t} + fu + g \frac{\partial q}{\partial y} + r_v v = F_v + f^v \quad (6.4b)$$

$$\frac{\partial q}{\partial t} + H \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + r_q q = F_q + f^q \quad (6.4c)$$

Com condições iniciais:

$$u(x, y, 0) = I^u(x, y) + i^u(x, y) \quad (6.5a)$$

$$v(x, y, 0) = I^v(x, y) + i^v(x, y) \quad (6.5b)$$

$$q(x, y, 0) = I^q(x, y) + i^q(x, y) \quad (6.5c)$$

com condições de contorno periódicas, dadas a seguir:

$$u(x \pm X, y, t) = u(x, y, t) \quad (6.6a)$$

$$v(x \pm X, y, t) = v(x, y, t) \quad (6.6b)$$

$$q(x \pm X, y, t) = q(x, y, t) \quad (6.6c)$$

e com condições de contorno rígida dadas por:

$$v(x, 0, t) = b_0(x, t) \quad (6.7)$$

$$v(x, Y, t) = b_y(x, t)$$

O dado observado para a variável  $q$  do modelo é representado por:

$$d_m = q(x_m, y_m, t_m) + \epsilon_m, \quad 1 \leq m \leq M \quad (6.8)$$

em que  $\epsilon_m$  é o ruído da observação e  $M$  é a quantidade de dados observados. O problema de assimilação para este problema consiste em determinar os seguintes parâmetros:

- a forçante:  $f^u$ ,  $f^v$  e  $f^q$
- a condição inicial:  $i^u(x, y)$ ,  $i^v(x, y)$  e  $i^q(x, y)$
- a condição de contorno:  $b_0(x, t)$  e  $b_y(x, t)$

Neste caso, a RNA deve emular o método do representante para estimar estes três parâmetros. Para o modelo de água rasa em duas dimensões estima-se a condição inicial, a condição de contorno e a forçante. A estimativa desses parâmetros com o método do Representante foi feita com o *software* IOM - *Inverse Ocean Modeling*

(BENNETT, 2002; BENNETT et al., 2008) que encontra-se disponível na *homepage* <http://iom.asu.edu>.

O algoritmo do representante descrito na Seção 4.3 para minimização do funcional 4.53 é iterativo. O IOM minimiza o funcional de penalidade ou função custo por um ajuste de mínimos quadrados ponderados (BENNETT et al., 2008).

O modelo de água rasa em duas dimensões representado pelas Equações 6.4 possui três variáveis de estado, as componentes de velocidade do fluido  $(u, v)$  e  $q$  a perturbação de superfície livre (elevação da superfície). O processo de assimilação para esse modelo considerou-se três núcleos (*ker*) de observações:

- 1) perturbação em um ponto espaço-tempo em  $x = 550000$ ,  $y = 550000$ ,  $z = 0$ ,  $t = 1350$ , em que a distância é dada em metros e o tempo em segundos.
- 2) uma média das perturbações em dois pontos espaço-tempo, isto é

$$ker_2 = 0,5(q_1 + q_2) \quad (6.9)$$

em que  $q_1$  está definido nas coordenadas  $x = 600000$ ,  $y = 600000$ ,  $z = 0$ ,  $t = 1350$  e  $q_2$  em  $x = 600000$ ,  $y = 700000$ ,  $z = 0$ ,  $t = 1350$ .

- 3) uma integral de velocidade tangencial ao longo de um segmento de reta com  $45^\circ N$ , em um tempo aproximado numericamente pela regra de Simpson e quatro pontos de acordo com:

$$ker_3 = 0,5(u_1 + v_1) + (u_2 + v_2) + (u_3 + v_3) + 0,5(u_4 + v_4) \quad (6.10)$$

em que os subscritos da Item 6.10 correspondem aos valores dados na Tabela 6.8.

Tabela 6.8 - Valores das coordenadas para a observação 3.

<i>subscritos</i>	$x$	$y$	$z$	$t$
1	500000	500000	0	1800
2	600000	600000	0	1800
3	700000	700000	0	1800
4	800000	800000	0	1800

A Figura 6.24 ilustra a grade utilizada neste experimento. A dimensão da grade é  $20 \times 11$  com  $\Delta x = \Delta y = 10^6$ . A observação está definida na coordenada (6, 6) para todas as variáveis do modelo, os demais pontos verdes definidos na grade são os pontos que tiveram influência da observação. A assimilação de dados é realizada nos passos de tempo 10 e 11. O ponto verde representa a influência observação no passo de tempo 10 e o círculo vermelho representa a influência da observação no passo de tempo 11.

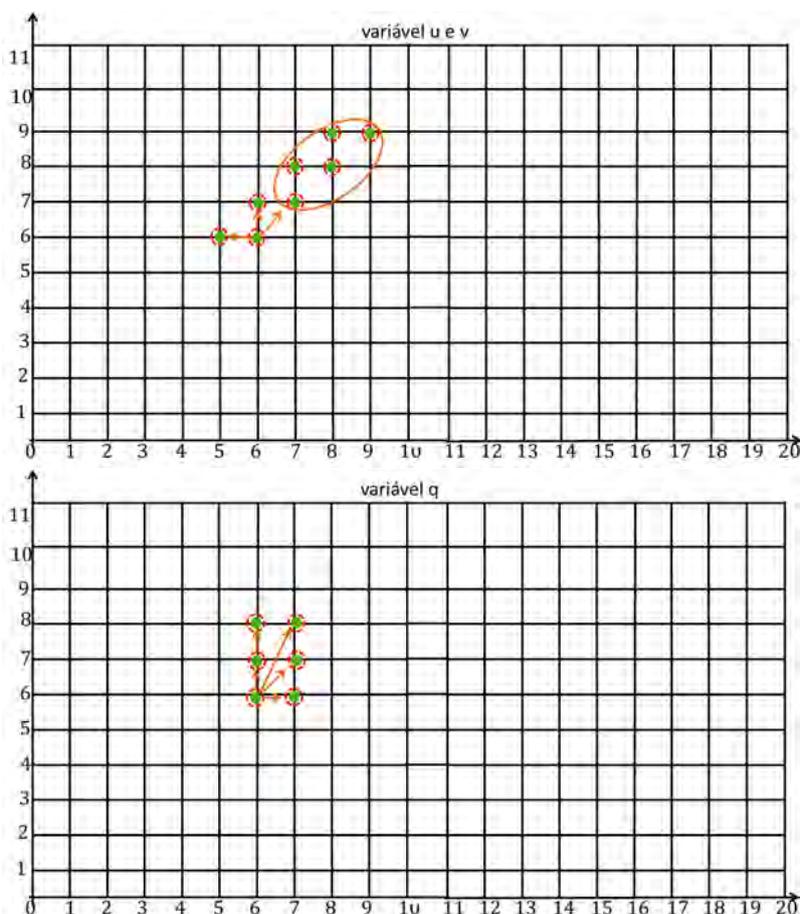


Figura 6.24 - Observações interpoladas no ponto de grade do modelo para  $u, v$  e  $q$ ; ponto verde: observação no tempo 10; círculo vermelho: observação no tempo 11.

Um dos objetivos deste trabalho é avaliar a capacidade da RNA emular o método do representante. Para isto, adotou-se a seguinte estratégia. Realizou-se a assimilação de dados com o método do representante em 20 “*janelas*” para compor o conjunto de treinamento.

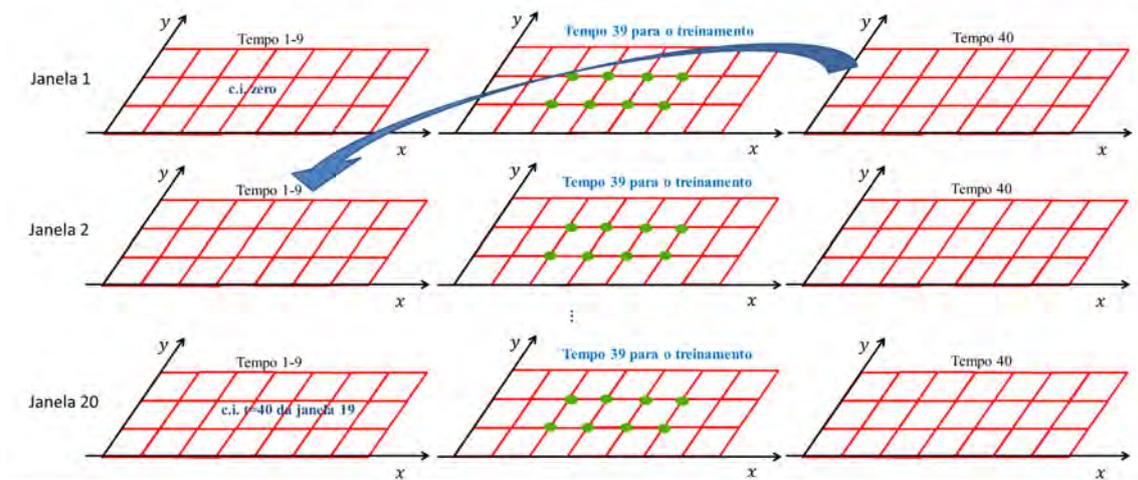


Figura 6.25 - Janelas de integração do modelo de água rasa 2D. Os pontos verdes indicam o ponto da observação para a variável  $u$  e  $v$ .

O que denominou-se de “*janela*” corresponde a integrar o modelo em 40 passos de tempo, com grade  $20 \times 11$ . Para a primeira “*janela*”, as variáveis  $u$ ,  $v$  e  $q$  tiveram valor inicial igual a zero. A “*janela*” subsequente iniciou a integração com o campo do último passo de tempo obtido da “*janela*” anterior. O processo se repete para as demais “*janelas*” para todos os parâmetros ( $u, v, q, f, b$ ), conforme ilustrado nas Figuras 6.25 e 6.26. Esta abordagem foi feita para se obter o conjunto de padrões para o treinamento da rede neural artificial. A diferença entre as Figuras 6.25 e 6.26 está na quantidade de observações assimiladas. Na primeira, usou-se 8 e na segunda 6 observações. Para compor o conjunto de treinamento, pegou-se apenas os dados do tempo  $nk = 39$ .

A quantidade de dados que compõem os conjuntos dos dados de treinamento para a estimação de cada parâmetro está definido na Tabela 6.9 a seguir:

Tabela 6.9 - Número de dados que formam os conjuntos de treinamento.

parâmetros	dados de entrada	saída desejada
c. i. $p/u$ e $v$	$180 \times 2$	$180 \times 1$
c. i. $p/u$ e $v$	$180 \times 2$	$160 \times 1$
Forçante	$180 \times 2$	$160 \times 1$
condição de contorno	$20 \times 21$	$20 \times 20$

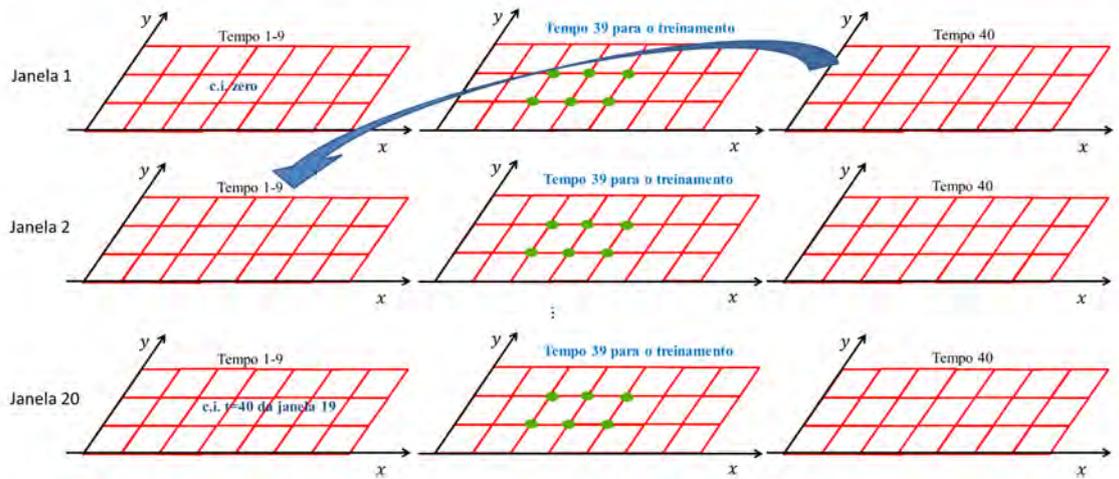


Figura 6.26 - Janelas de integração do modelo de água rasa 2D. Os pontos verdes indicam o ponto da observação para a variável  $q$ .

O conjunto de dados de treinamento para as respectivas redes constam na Tabela 6.10. A saída desejada corresponde a estimativa obtida com o método do Representante.

Tabela 6.10 - Conjuntos de treinamento.

Rede	dado de entrada	saída desejada
Condição inicial	modelo e observação	$i^u, i^v$ e $i^q$
Forçante	modelo e observação	$f^u, f^v$ e $f^q$
Condição de contorno	cond. cont. do modelo e observação	$b_0$ e $b_Y$

Para a estimação das condições iniciais (Equação 6.5), forçantes (Equação 6.4) e condição de contorno (Equação 6.8) foram implementadas redes PMC (Perceptron de Múltiplas Camadas) para cada parâmetro de acordo com as Tabelas 6.11, 6.12 e 6.13. Em que:  $n_{co}$  é o número de neurônios na camada oculta,  $\eta$  é a taxa de aprendizagem e  $a$  é o valor da tangente hiperbólica.

Para testar a capacidade de generalização da rede, integrou-se o modelo em mais uma janela conforme ilustrado na Figura 6.27. Na generalização, utilizam-se dados que não fizeram parte do conjunto de treinamento. O tempo  $t = 1$  para a janela 21 tem como condição inicial o campo do tempo  $t = 40$  da janela 20. As observações

Tabela 6.11 - Parâmetros da rede PMC para a estimação da condição inicial.

variável	$nco$	$\eta$	$a$
$u$	45	0,01	1
$v$	40	0,01	1
$q$	40	0,01	1

Tabela 6.12 - Parâmetros da rede PMC para a estimação da forçante.

variável	$nco$	$\eta$	$a$
$u$	35	0,005	1
$v$	35	0,003	1
$q$	35	0,005	1

Tabela 6.13 - Parâmetros da rede PMC para a estimação da condição de contorno.

variável	$nco$	$\eta$	$a$
$b_0$	40	0,007	1
$b_Y$	40	0,001	1

foram inseridas nos tempos  $t = 10$  a  $t = 39$ .

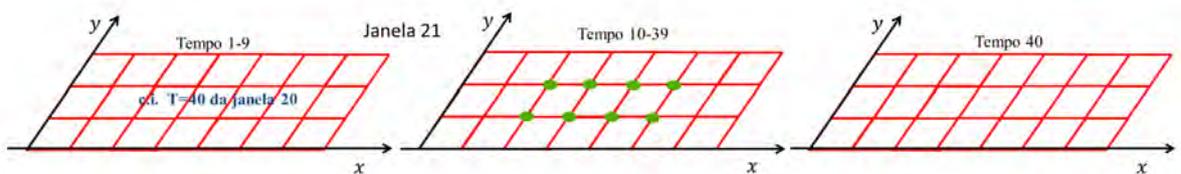


Figura 6.27 - Generalização da RNA; janela 21

A Figura 6.28 mostra o resultado para a variável  $u$  (vento zonal em  $m/s$ ) para o passo de tempo  $t = 10$ . A curva azul é a referência de *verdade* e a curva verde é o estimado com a RNA. Na Figura 6.29, tem-se a estimativa obtida com o método do Representante.

No gráfico da Figura 6.28, pode-se verificar que na região marcada em vermelho a rede não conseguiu acompanhar a dinâmica da verdade, no restante do domínio a rede obteve uma estimativa próxima da verdade. A estimativa obtida com o Representante, representada pela curva vermelha na Figura 6.29, acompanha em todo o domínio a dinâmica da *verdade*, sendo que, é difícil diferenciar as duas dinâmicas. Isso é confirmado no gráfico das diferenças mostrado na Figura 6.30. O erro para a rede é da ordem de  $10^{-3}$  e para o método do representante é da ordem de  $10^{-7}$ .

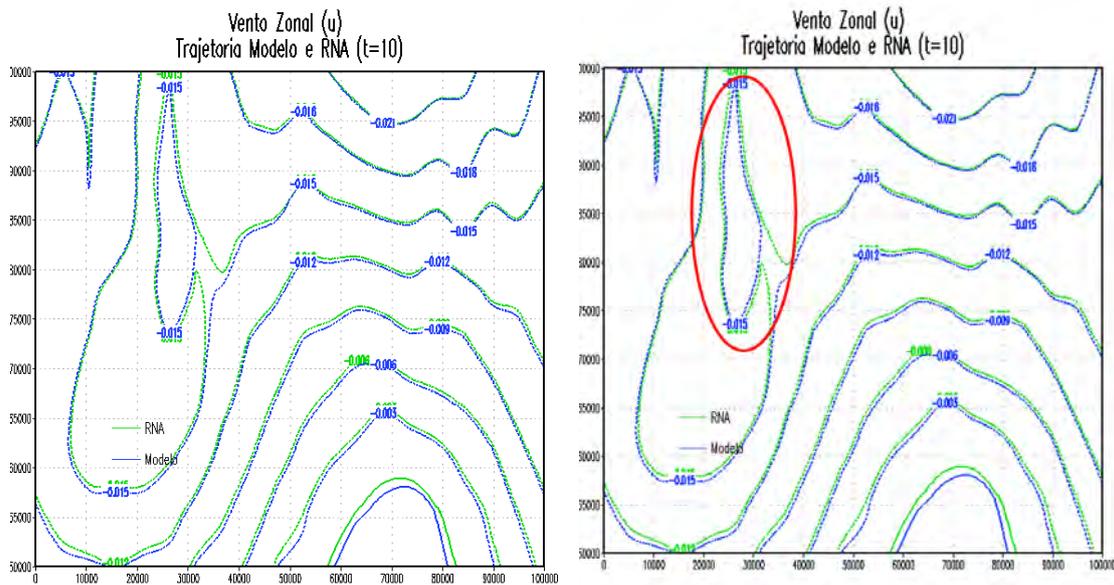


Figura 6.28 - Variável  $u$  (vento zonal em  $m/s$ ); estimativa obtida com a RNA (curva verde); *verdade* (curva azul).

A Figura 6.31 apresenta as estimativas obtidas pela RNA (gráfico à esquerda) e por meio do método do Representante (gráfico à direita). Neste caso, a rede obteve uma boa estimativa acompanhando a dinâmica do modelo *verdade* por todo o domínio. A ordem do erro da estimativa para a rede é de  $10^{-5}$  e para o método do representante é da ordem de  $10^{-09}$  (veja Figura 6.32).

A Figura 6.33 apresenta o resultado da estimativa obtida com a RNA para a variável  $q$ . O gráfico à direita da Figura 6.33 ressalta com um círculo vermelho a região do domínio em que a houve uma divisão de um estrutura em duas partes. Nesta região, a estimativa obtida por meio da RNA (curva verde) desacoplou da dinâmica do modelo

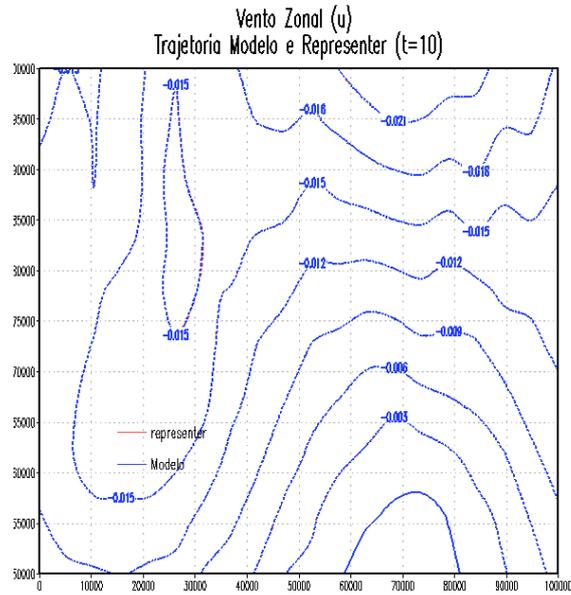


Figura 6.29 - Variável  $u$  (vento zonal em  $m/s$ ); estimativa obtida com o método do Representante (curva vermelha); *verdade* (curva azul).

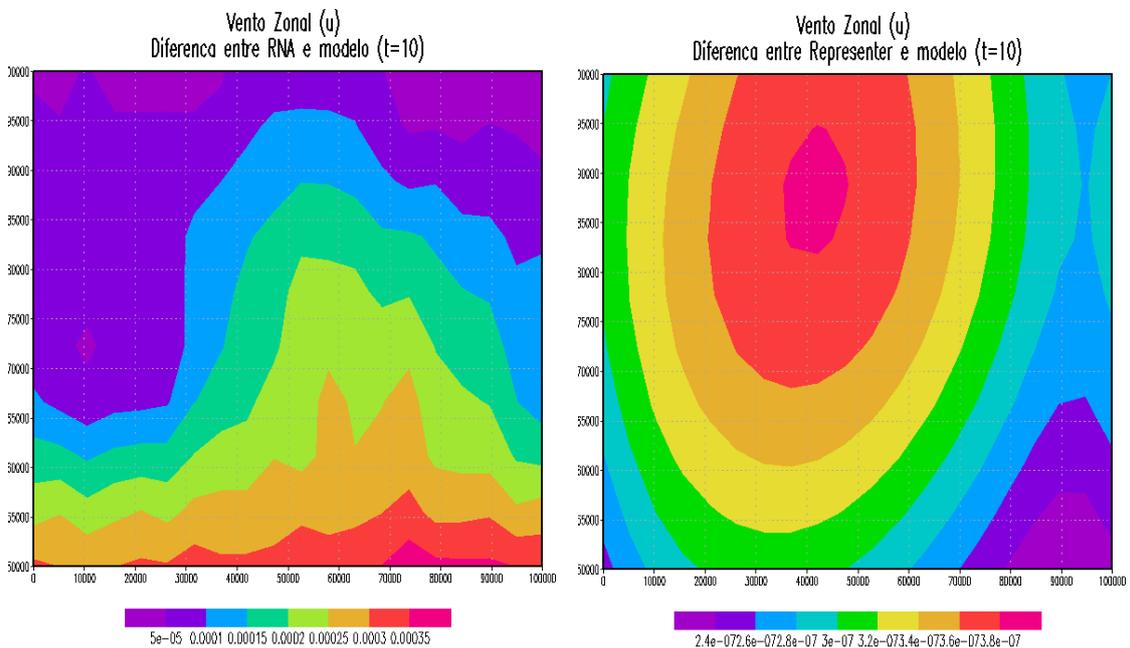


Figura 6.30 - Gráficos das diferenças para a variável  $u$  e  $t = 10$ . lado esquerdo: rede x modelo (*verdade*); lado direito: representante x modelo (*verdade*).

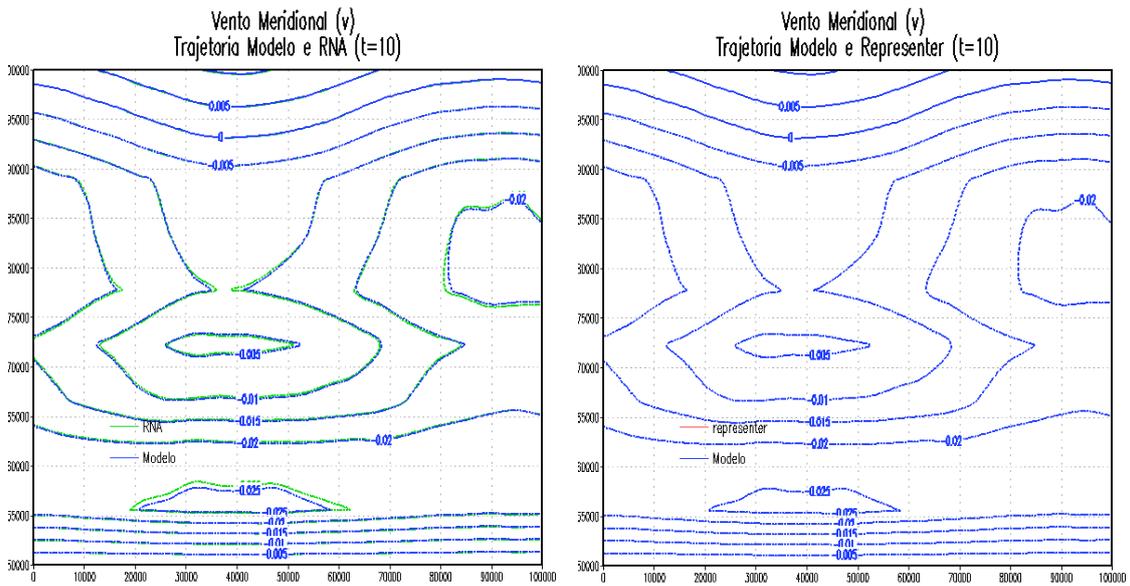


Figura 6.31 - Variável  $v$  ( $m/s$ ) no tempo  $t = 10$ ; lado direito: RNA x verdade; lado esquerdo: Representante x verdade.

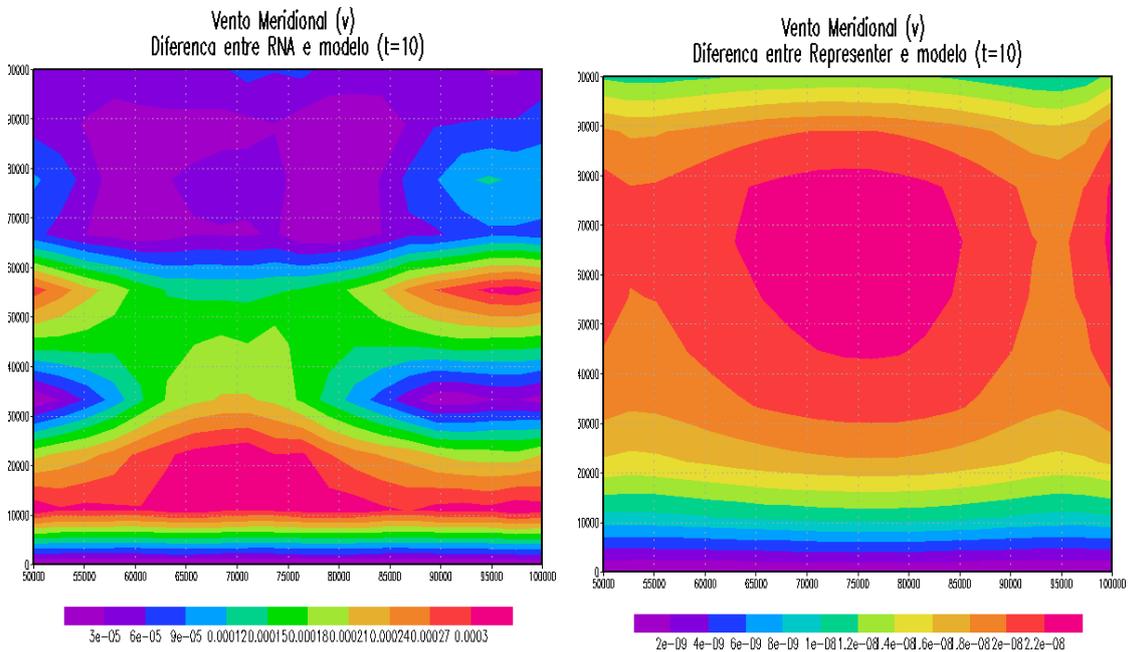


Figura 6.32 - Gráficos das diferenças para a variável  $v$  e  $t = 10$ . lado esquerdo: rede x modelo (*verdade*); lado direito: representante x modelo (*verdade*).

(*verdade*) (curva azul). A Figura 6.34 mostra o resultado da estimativa obtida com o método do representante. O gráfico da diferença para os dois métodos é mostrado na Figura 6.35.

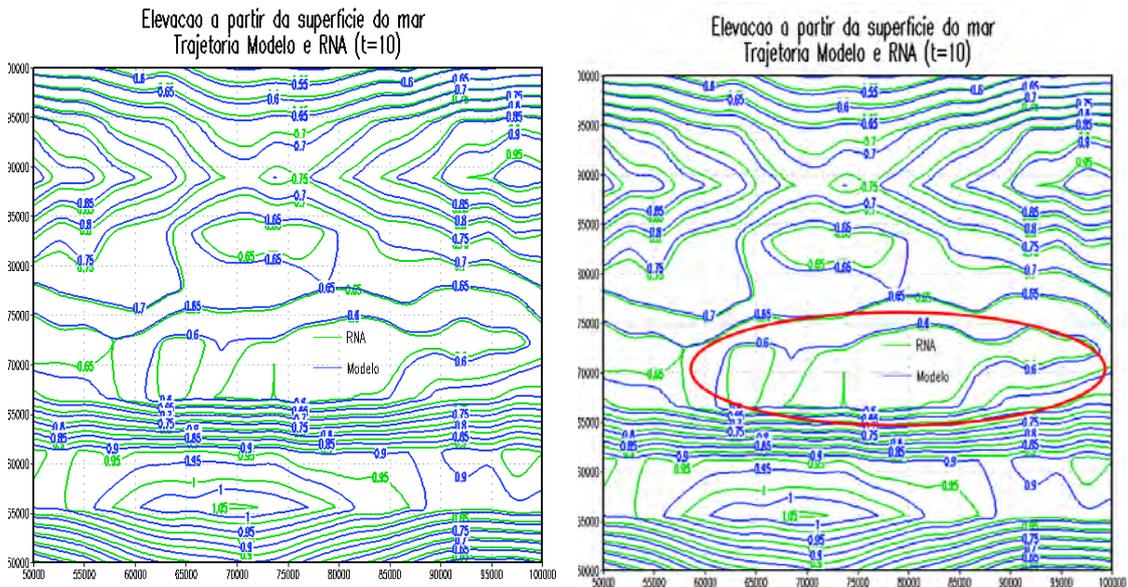


Figura 6.33 - Variável  $q$  ( $m$ ) no tempo  $t = 10$ ; estimativa obtida com a RNA; curva azul: modelo (*verdade*); curva verde: estimado.

A Figura 6.36 mostra os resultados com o método do Representante (gráfico à direita) e RNA (gráfico à esquerda) para a variável  $u$  no tempo  $t = 20$ . Neste instante de tempo, percebe-se o desacoplamento da estimativa obtida com a rede com relação a *verdade*. Pode-se verificar que no gráfico a direita da Figura 6.38, a região estimada com o representante é muito próxima da verdade.

O gráfico da diferença é mostrado na Figura 6.37, à esquerda é a diferença entre a rede e o modelo (*verdade*) e a direita a difereça entre o Representante e o modelo (*verdade*).

A Figura 6.38 mostra a estimativa obtida com a rede neural artificial para a variável  $v$  no tempo  $t = 20$ . Pode-se verificar que no gráfico à direita da Figura 6.38 a região circulada em vermelho resalta uma estrutura identificada pela RNA que não existe. No entanto, no restante do domínio a rede fez uma boa estimativa que conseguiu acompanhar a dinâmica do modelo (*verdade*).

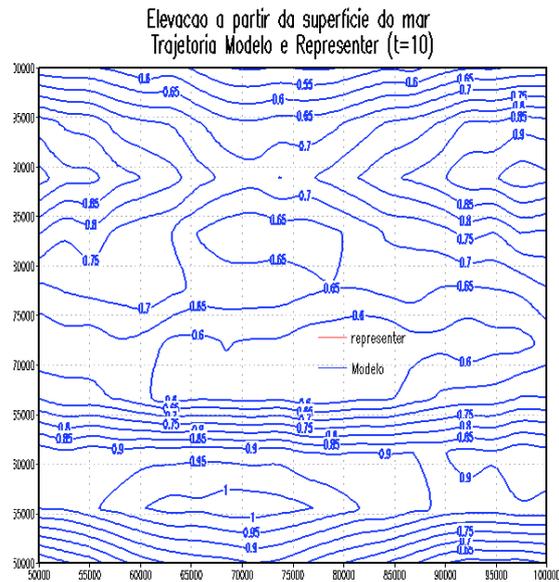


Figura 6.34 - Variável  $q$  (m) no tempo  $t = 10$ ; estimativa obtida com o representante; curva azul: modelo (*verdade*); curva vermelha: estimado

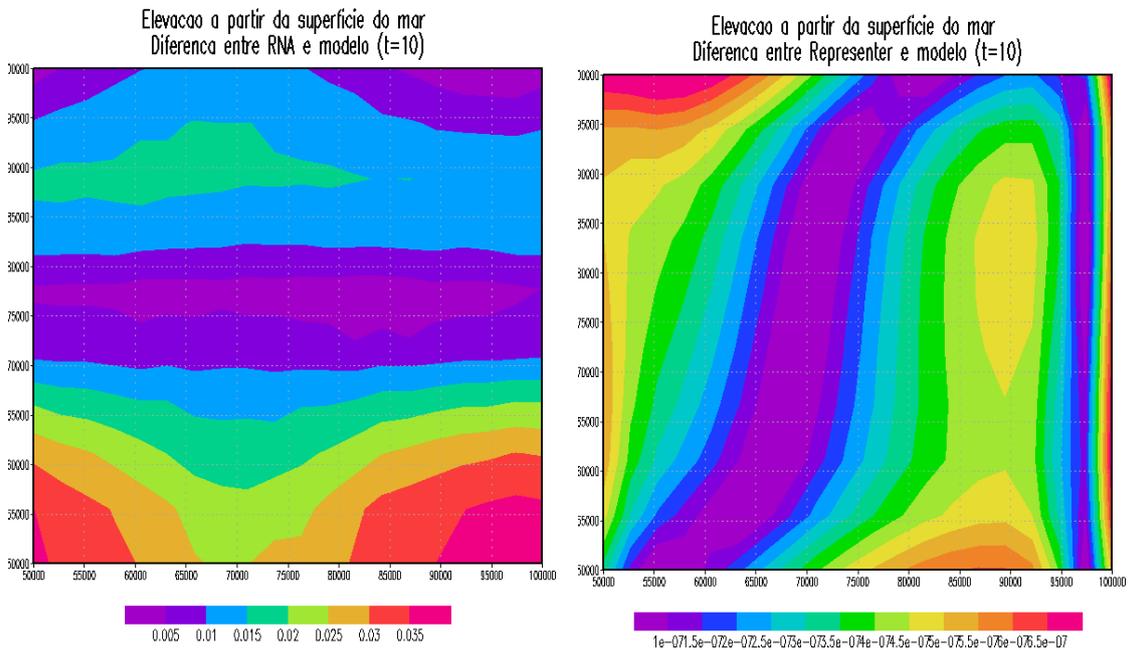


Figura 6.35 - Gráficos das diferenças para a variável  $q$  e  $t = 10$ . lado esquerdo: rede x modelo (*verdade*); lado direito: representante x modelo (*verdade*).

Na Figura 6.39, tem-se a estimativa obtida com o método do Representante. As-

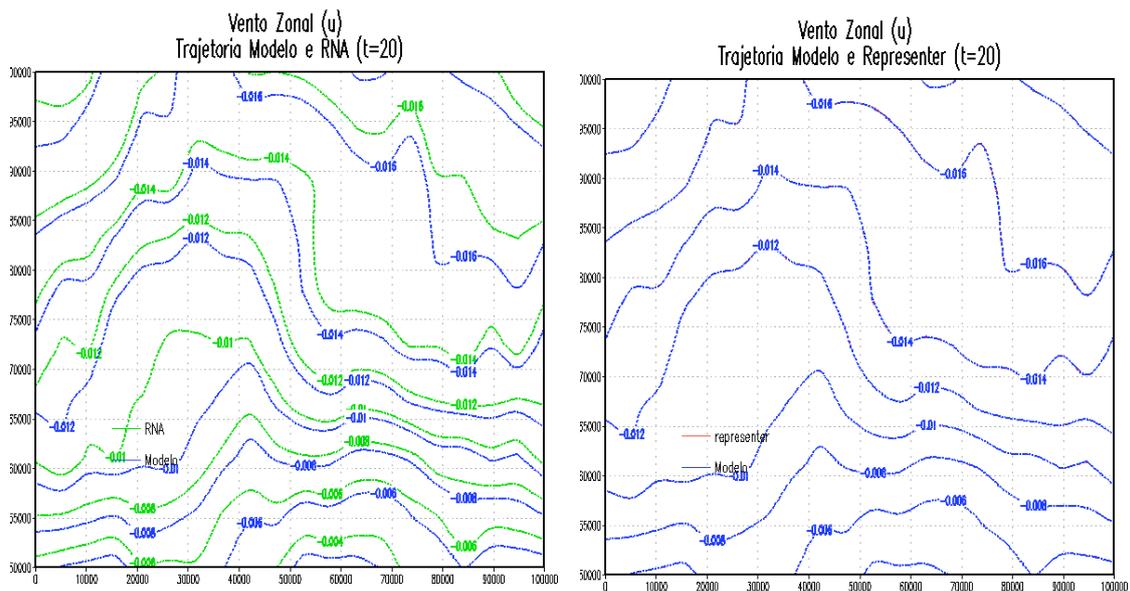


Figura 6.36 - Variável  $u$  (vento zonal em  $m/s$ ) no tempo  $t = 20$ ; lado direito: RNA x verdade; lado esquerdo: Representante x verdade.

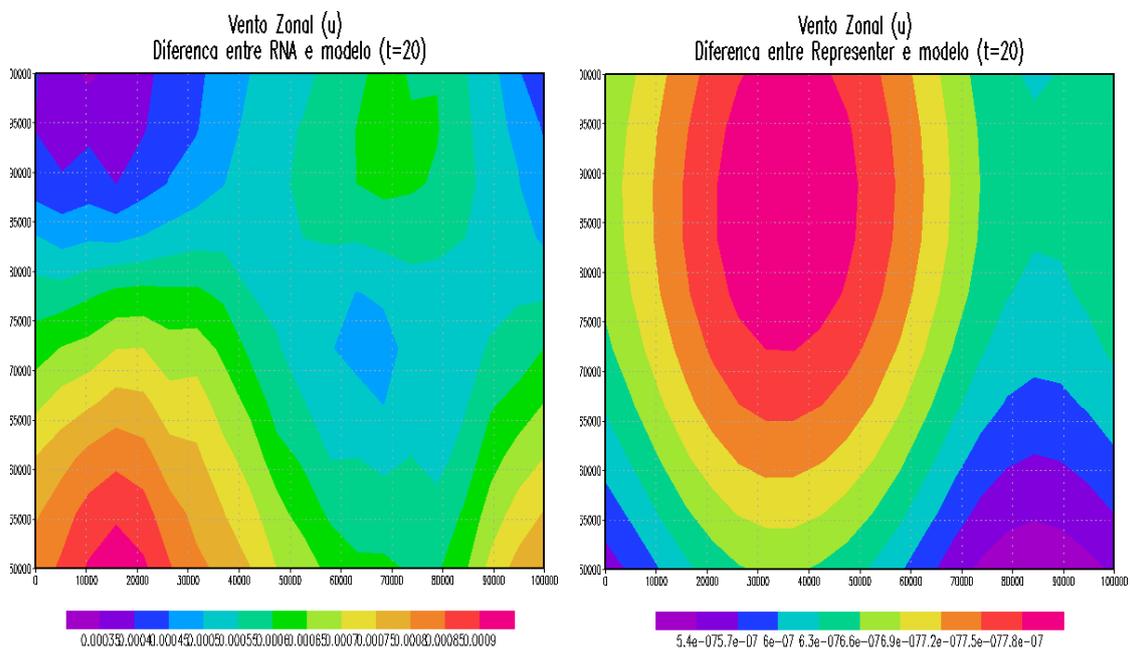


Figura 6.37 - Gráficos das diferenças para a variável  $u$  em  $t = 20$ ; lado esquerdo: rede x modelo (*verdade*); lado direito: representante x modelo (*verdade*).

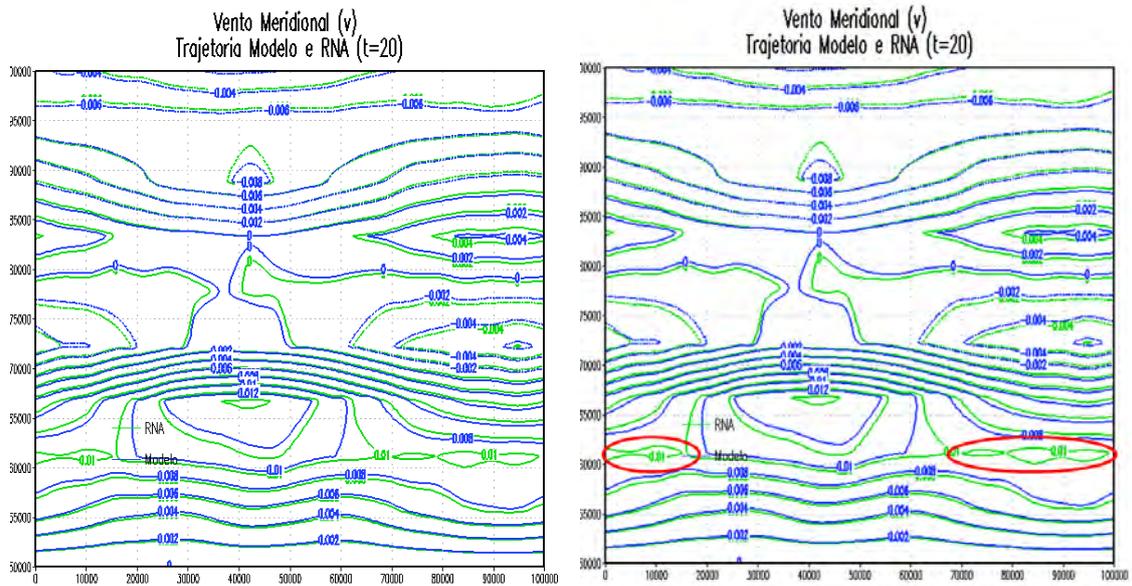


Figura 6.38 - Variável  $v$  no tempo  $t = 20$ ; estimativa obtida com a RNA; curva azul: modelo (*verdade*); curva verde: estimado.

sim como para a variável  $u$ , esse método obteve uma estimativa muito próxima da dinâmica do modelo (*verdade*) para a variável  $v$ . Visualmente não possível diferenciar a dinâmica da estimativa obtida com o método do representante com a dinâmica do modelo. Quantitativamente, pode-se verificar a diferença na Figura 6.40 no gráfico à direita tem-se a diferença absoluta entre os campos da estimativa com o modelo (*verdade*). O erro para a estimativa obtida com a rede neural é da ordem de  $10^{-3}$  e para a estimativa obtida com o Representante é da ordem de  $10^{-8}$ .

As estimativas obtidas para a variável  $q$  do modelo de água rasa são mostradas nas Figuras 6.41 e 6.42 para o tempo  $t = 20$ . Na Figura 6.41, a curva verde representa o estimado pela RNA e a curva azul é a referência de *verdade* (modelo). Neste instante de tempo, percebe-se que a análise com redes neurais inseriu uma estrutura (destacada pelo círculo vermelho no gráfico à direita) que não corresponde a dinâmica do modelo. O erro de assimilação para rede neural para esta variável é da ordem de  $10^{-3}$ , como pode ser visto na Figura 6.43 no gráfico à esquerda. A estimativa obtida com o método do Representante possui erro da ordem de  $10^{-6}$ .

A rede neural ao emular o método do Representante para o modelo 2D não reproduziu o mesmo desempenho como apresentado no modelo 1D. Mais estudos são

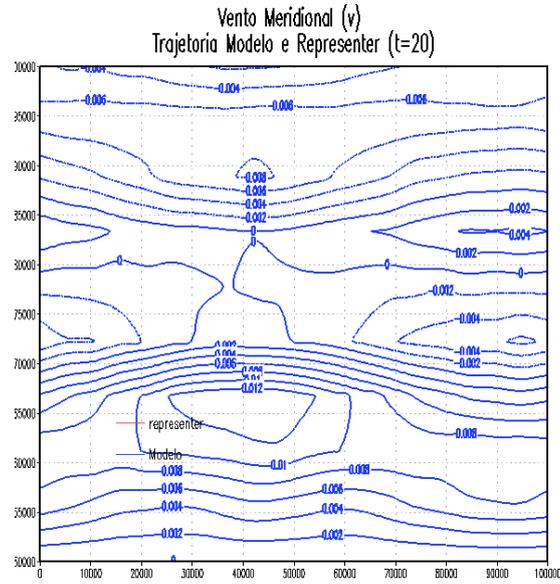


Figura 6.39 - Variável  $v$  no tempo  $t = 20$ ; estimativa obtida com o método do representante; curva azul: modelo (*verdade*); curva vermelha: estimado.

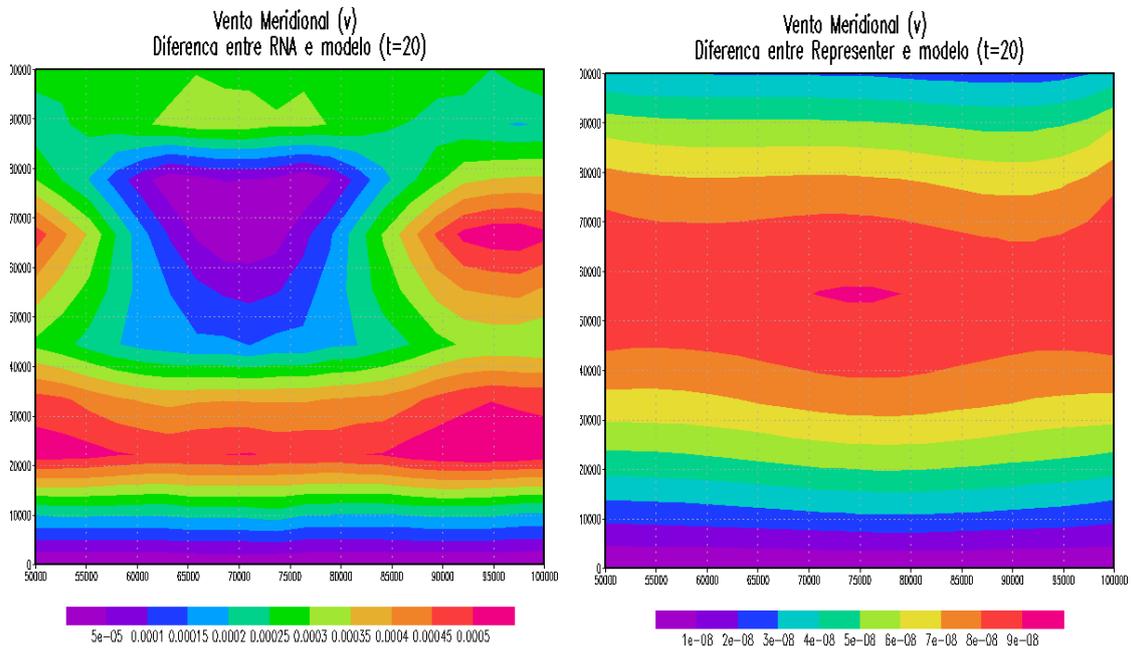


Figura 6.40 - Gráficos das diferenças para a variável  $v$  em  $t = 20$ ; estimativa obtida com a RNA; curva azul: verdade; curva verde: estimado.

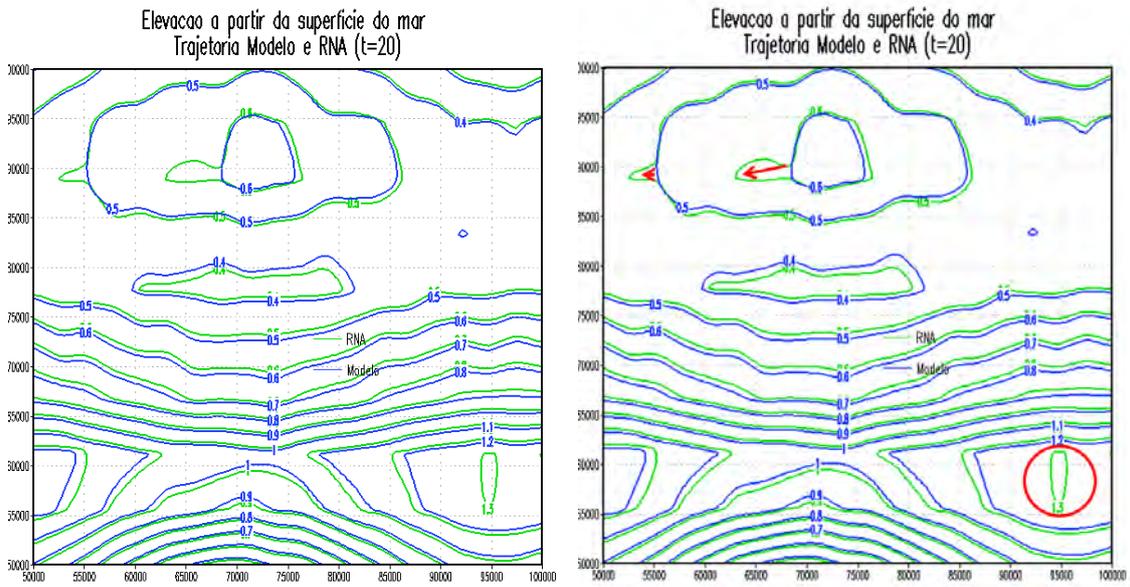


Figura 6.41 - Variável  $q$  no tempo  $t = 20$ ; estimativa obtida com a RNA; curva azul: modelo (*verdade*); curva verde: estimado.

necessários. Entretanto, a primeira estratégia a ser investigada é aumentar o conjunto de dados para o treinamento da rede, pois com maiores informações sobre a dinâmica do modelo, a rede terá capacidade de seguir a dinâmica do modelo por todo o domínio e em qualquer instante de tempo. Outra alternativa, é alterar a zona de influência da observação aquela adotada pelo método do Representante, ilustrada pela Figura Figura 6.24.

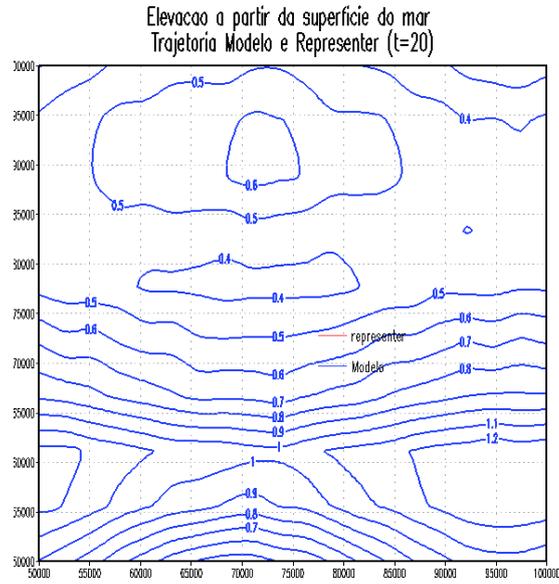


Figura 6.42 - Variável  $q$  no tempo  $t = 20$ ; estimativa obtida com o método do representante; curva azul: modelo (*verdade*); curva vermelha: estimado.

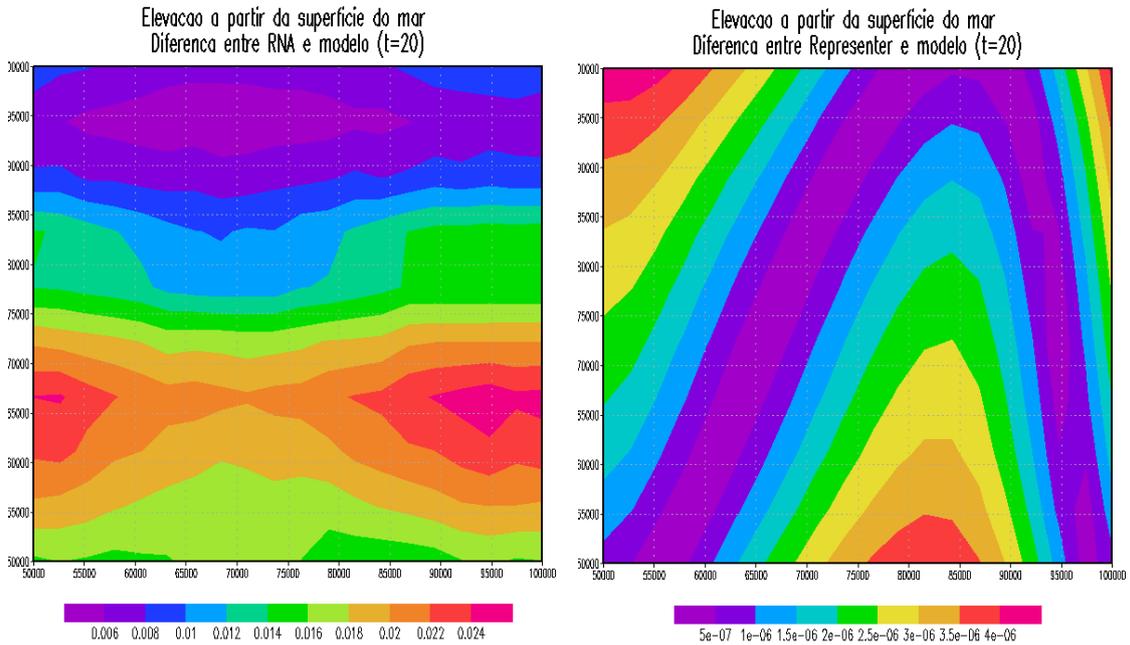


Figura 6.43 - Gráficos das diferenças para a variável  $q$  em  $t = 20$ ; lado esquerdo: rede  $\times$  modelo (*verdade*); lado direito: representante  $\times$  modelo (*verdade*).

## 7 CONCLUSÃO

Assimilação de dados é um processo essencial em todos os sistemas operacionais de previsão. A necessidade foi primeiramente notada em previsão numérica do tempo, pois foram os meteorologistas os primeiros a utilizar um modelo computacional para realizar previsão. A assimilação constitui-se no emprego de uma técnica para calcular a partir de medidas observacionais de um sistema real a melhor condição inicial para um modelo de previsão, ou seja, o cálculo da melhor análise. Com a quantidade crescente da resolução dos modelos numéricos e com o crescimento exponencial dos dados de observação, a assimilação de dados tornou-se um grande desafio científico, pois deve-se aliar precisão com rapidez na computação da condição inicial.

Dados provenientes dos modelos numéricos em um futuro próximo estarão na ordem de  $10^9$ , enquanto os dados observados estarão na ordem de pelo menos  $10^7$ , mas tendem a superar em muito o número de dados de pontos de grade. Para um cenário de uma quantidade crescente de pontos de grade e observações, uma questão importante é: como produzir uma boa análise na janela de tempo disponível na prática operacional?

Neste trabalho, avaliou-se o desempenho de alguns métodos de assimilação de dados: filtro de Kalman, método variacional na formulação do representante e a técnica de Redes Neurais Artificiais (RNA) – Perceptron de Múltiplas Camadas (PMC). Os métodos foram testados em um modelo de onda linear 1D e um modelo de água rasa 2D. O [Capítulo 6](#) mostra que a metodologia de RNA poder ser uma alternativa viável para o cálculo da análise, satisfazendo uma precisão aceitável, isto é, não compromete a qualidade da previsão, e o cálculo é realizado em um intervalo de tempo bastante inferior aos de outros métodos.

A [Tabela 7.1](#) reproduz os tempos utilizados para o cálculo da assimilação com diferentes técnicas. Nota-se que a assimilação com redes neurais é quase 16 vezes mais rápida do que o filtro de Kalman (com 100 observações), enquanto que, a RNA é 113 vezes mais rápida do que o método do representante.

De acordo com [Lorenç \(1986\)](#), sob certas condições, há uma equivalência entre o filtro de Kalman e o método 4D-Var. Todavia, a formulação variacional demanda um trabalho adicional na dedução da equação adjunta, por exemplo. Há alguns modelos em que sequer se sabe se é possível deduzir a equação de Euler-Lagrange

Tabela 7.1 - Tempo de CPU para diferentes técnicas de assimilação com diferentes modelos de previsão, para vários experimentos numéricos.

Modelo	FK	Rep	RNA-PMC-FK	RNA-PMC-Rep
Onda 1D: 10 passos tempo	29,61 s	-	12,48 s	-
Onda 1D: 20 passos tempo	44,56 s	-	19,51 s	-
Agua rasa 2D: 25 Obs	42 min	-	1,39 min	-
Agua rasa 2D: 100 Obs	79 min	-	5,00 min	-
Agua rasa 2D: 3 Obs	-	4 min	-	0 min 2 s

associada.

A versão do método variacional proposta por Bennett e colaboradores é ainda mais sofisticada, pois é necessário a derivação do adjunto do representante, isto é, são necessários cálculos adicionais (ver Equações 4.59, 4.62).

Contudo, mesmos nas versões mais sofisticadas do filtro de Kalman e no método variacional, há uma questão em aberto: o cálculo da matriz de co-variância do erro de modelagem. As metodologias propostas para endereçar esta questão são: filtro de Kalman adaptativo (JAZWINSKI, 1970; DEE et al., 1985; NOWOSAD et al., 2000), equação de Fokker-Planck (BELYAEV; TANAJURA, 2005) e filtro de Kalman por ensemble (EVENSEN, 1994; KALNAY, 2003). Há propostas híbridas, combinando o EnKF com o método variacional.

Uma estratégia que vem sendo investigada é a aplicação do filtro de partículas (GORDON et al., 1993; CHORIN; KRAUSE, 2004). Embora, esta técnica não suponha hipóteses de gaussianidade do filtro de Kalman, e nem de linearidade (algumas vezes empregadas tanto para o filtro de Kalman, quanto para o método variacional), esta técnica é a de maior custo computacional. Além disso, a função de verossimilhança desempenha um papel expressivo no filtro de partículas. Para garantir que o resultado do filtro de de partículas possa representar distribuições vinculadas ao teorema central do limite na sua forma padrão (PAPOULIS, 1984), bem como o teorema central do limite na versão de Levy-Genedenko, recentemente foi introduzida um novo filtro de partículas adaptativo (FURTADO; Campos Velho, 2011). Na formulação de Gordon et al. (1993) do filtro de partículas, Furtado et al. (2008) mostraram que as RNAs podem também emular o filtro de partícula.

Os trabalhos anteriores em assimilação com redes neurais foram aplicados em modelos de baixa dimensão (Lorenz - (NOWOSAD et al., 2000; NOWOSAD, 2001; HARTER, 2004; FURTADO et al., 2008; CINTRA et al., 2010); dinâmica solar (HÄRTER et al., 2008) ou modelos vinculados a dinâmica da atmosfera (DYNAMO, água rasa 1D (HÄRTER; CAMPOS VELHO, 2012) e modelo SPEEDY 3D (CINTRA, 2010).

Nesta tese, aplicam-se redes neurais para emular métodos de assimilação empregados em modelos mais próximos à dinâmica de circulação oceânica. Os resultados foram muito animadores, apresentando uma drástica redução no custo computacional.

Como trabalhos futuros, sugere-se realizar o treinamento das redes neurais com um conjunto que contenha maior número de dados observados e dados do modelo físico-matemático. Provavelmente, os resultados obtidos com as redes poderão ser mais próximos dos resultados obtidos com o método do representante (ver Seção 6.2). A aplicação da metodologia de redes auto-cofiguráveis foi efetiva para o modelo de onda 1D (SAMBATTI et al., 2012). Deve-se investigar se a estratégia permanece válida para o modelo de água rasa 2D empregado nesta tese. Outra linha de pesquisa interessante, é a aplicação de redes neurais não supervisionadas, em que não será necessário o processamento de outro método de assimilação de dados.

Finalmente, um tema importante é o uso de computação híbrida, onde parte do processamento é executada em *hardware*. As redes neurais podem ser implementadas em FPGA (*Field-programmable gate array*). Assim, a FPGA configurada para atuar como uma rede neural, o processo de assimilação de dados é realizado por um neuro-computador. Os resultados de Shigemori (2007) e Gomes (2012) mostram a viabilidade deste recurso, que aumentaria ainda mais a eficiência computacional da técnica de redes neurais.



## REFERÊNCIAS BIBLIOGRÁFICAS

ARAVEQUIA, J. A.; DIAS, P. L. d. S. Ferramenta de análise e prognóstico de anomalias climáticas usando funções de green de um modelo barotrópico divergente. In: CONGRESSO BRASILEIRO DE METEOROLOGIA, 13., 13., 2004, Fortaleza. **Anais... SBMET**, 2004. Disponível em:

<<http://urlib.net/cptec.inpe.br/walmeida/2004/09.16.14.46>>. Acesso em: 25 fev. 2012. 17

BELYAEV, K. P.; TANAJURA, C. A. On the correction of perturbations due to data assimilation in ocean circulation models. **Applied Mathematical Modelling**, v. 29, n. 7, p. 690 – 709, 2005. ISSN 0307-904X. Disponível em:

<<http://www.sciencedirect.com/science/article/pii/S0307904X04001490>>. 124

BENNETT, A. F. **Inverse methods in physical oceanography**. 1. ed. UK: Cambridge University Press, 1992. 234 p. ISBN 9780511600807. 9, 43

\_\_\_\_\_. **Inverse modeling of the ocean and atmosphere**. 1. ed. New York: Cambridge University Press, 2002. 234 p. ISBN 0-521-81373-5. 3, 9, 18, 19, 20, 21, 46, 61, 108

\_\_\_\_\_. **Linear one-dimensional problem**. 2004. Notas de aula. Disponível em: <<http://iom.asu.edu/index.cfm?fuseaction=instruction.lessons>>. Acesso em: 26 fevereiro 2007. 45, 57

BENNETT, A. F.; BAUGH, J. R. A parallel algorithm for variational assimilation in oceanography and meteorology. **Journal of Atmospheric and Oceanic Technology**, American Meteorological Society, v. 9, n. 4, p. 426–433, August 1992. 9

BENNETT, A. F.; CHUA, B. S.; PFLAUM, B. L. The inverse ocean modeling system. part i: Implementation. **Journal of Atmospheric and Oceanic Technology**, American Meteorological Society, v. 25, p. 1608–1622, 2008. 108

BOUTTIER, F.; COURTIER, P. **Data assimilation concepts and methods**. 2002. 59 p. Meteorological Training Course Lecture Series. Disponível em: <[http://www.ecmwf.int/newsevents/training/rcourse\\_notes/pdf\\_files/Assim\\_concepts.pdf](http://www.ecmwf.int/newsevents/training/rcourse_notes/pdf_files/Assim_concepts.pdf)>. Acesso em: 28 de maio de 2010. 42

BRAGA, A.; LUDERMIR, T. B.; CARVALHO, A. C. P. L. F. **Redes neurais artificiais**: teoria e aplicações. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 2000. 77

BRAGA, A. P.; CARVALHO, A. P. L.; LUDERMIR, T. B. **Fundamentos de redes neurais artificiais**. Rio de Janeiro: DCC/IM, COPPE/Sistemas,NCE/UFRJ: Cambridge: MIT Press, 1998. 76

BRASSEUR, P. Ensemble-based data assimilation methods. In: SCHILLER, A.; BRASSINGTON, G. B. (Ed.). **Operational oceanography in the 21st century**. [S.l.]: Springer Netherlands, 2011. p. 381–393. ISBN 10.1007/978 – 94 – 007 – 0332 – 2 – 15. 43

BRASSEUR, P.; VERRON, J. The seek filter method for data assimilation in oceanography: a synthesis. **Ocean Dynamics**, Springer Berlin / Heidelberg, v. 56, p. 650–661, 2006. ISSN 1616-7341. 10.1007/s10236 – 006 – 0080 – 3. Disponível em: <<http://dx.doi.org/10.1007/s10236-006-0080-3>>. 43

BROQUET, G.; BRASSEUR, P.; ROZIER, D.; BRANKART, J.-M.; VERRON, J. Estimation of model errors generated by atmospheric forcings for ocean data assimilation: experiments in a regional model of the bay of biscay. **Ocean Dynamics**, Springer Berlin / Heidelberg, v. 58, p. 1–17, 2008. ISSN 1616-7341. Disponível em: <<http://dx.doi.org/10.1007/s10236-007-0128-z>>. 43

CAMPOS VELHO, H. F.; CINTRA, R. S.; FURTADO, H. C. M. **Introdução à assimilação de dados**. 2007. Disponível em: <[www.lac.inpe.br/~haroldo/Curso-DataAssimilacion/Curso\\_Assim\\_Dados-INPE-2.pdf](http://www.lac.inpe.br/~haroldo/Curso-DataAssimilacion/Curso_Assim_Dados-INPE-2.pdf)>. Acesso em: 28 de maio de 2012. 3

CAMPOS VELHO, H. F.; CLAEYSSSEN, J. Singular value decomposition in the integration of an atmospheric model. In: CONGRESSO ÍBERO-LATINO-AMERICANO DE MÉTODOS COMPUTACIONAIS PARA A ENGENHARIA, 13., 1992, Porto Alegre, Brasil. **Proceedings...** Porto Alegre: CILAMCE, 1992. p. 344–353. 32

CARDOSO, A. d. O.; FERRAZ, S. E. T.; DIAS, P. L. S.; ARAVEQUIA, J. A. Interação trópicos extratropicos - um estudo utilizando modelo de água rasa. In: SIMPOSIO BRASILEIRO DE METEOROLOGIA. **Anais...** 2002. p. 1087–1093. Disponível em: <<http://urlib.net/sid.inpe.br/iris@1915/2005/04.11.17.57>>. 17

- CARVALHO, A. R. **Uso de redes neurais otimizadas para recuperação do perfil de concentração de gases traço atmosféricos a partir de dados de satélites**. 208 p. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011-06-27 2011. Disponível em: <<http://urlib.net/sid.inpe.br/mtc-m19/2011/06.06.12.20>>. Acesso em: 14 nov. 2012. 81
- CHORIN, A. J.; KRAUSE, P. Dimensional reduction for a bayesian filter. **PNAS**, v. 101, n. 42, p. 15013–15017, August 2004. 3, 124
- CHUA, B. S.; BENNETT, A. F. An inverse ocean modeling system. **Ocean Modeling**, v. 3, p. 137–165, March 2001. 11, 43, 46, 104
- CINTRA, R. S.; VELHO, H. F. C.; TODLING, R. Nova implementação em filtro do kalman estendido para assimilação de dados com redes neurais. **Learning and Nonlinear Model**, v. 7, p. 30–37, 2010. 125
- CINTRA, R. S. C. **Assimilação de dados com redes neurais artificiais em modelo de circulação geral da atmosfera**. 200 p. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2010. Disponível em: <<http://urlib.net/sid.inpe.br/mtc-m19/2010/09.20.14.46>>. Acesso em: 09 fev. 2012. 3, 6, 8, 9, 125
- COSTA, M. A.; BRAGA, A. P.; MENEZES, B. R.; TEIXEIRA, A. R.; PARMA, G. G. Training neural networks with a multi-objective sliding mode control algorithm. **Neurocomputing**, Elsevier, Holanda, v. 51, p. 467–473, 2003. 81
- DALEY, R. **Atmospheric data analysis**. New York: Cambridge University Press, 1993. 3, 4
- DEE, D. P.; COHN, S. E.; DALCHER, A. D.; GHIL, M. An efficient algorithm for estimating noise covariances in distributed systems. **Automatic Control, IEEE Transactions**, AC-3, n. 11, p. 1057–1065, November 1985. 124
- EVENSEN, G. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. **Journal Geophysical Research**, v. 99, n. C5, p. 10143–10162, May 1994. 43, 84, 124
- FAUSETT, L. **Fundamentals of neural networks: architectures, algorithms, and applications**. New Jersey: Prentice Hall, 1994. 75

FURTADO, H.; Campos Velho, H. de. Adaptive particle filter for stable distribution. In: CONSTANDA, C.; HARRIS, P. J. (Ed.). **Integral methods in science and engineering**. First. Boston: Birkhäuser, 2011. p. 419. ISBN 978-0-8176-8237-8. 10.1007/978-0-8176-8238-5. 124

FURTADO, H.; Campos Velho, H. de; MACAU, E. Data assimilation by neural network emulating representer method applied to the wave equation. In: INTERNATIONAL SYMPOSIUM ON UNCERTAINTY QUANTIFICATION AND STOCHASTIC MODELING, 1., 2012, Maresias, São Sebastião. **Proceedings...** São Paulo: USP, 2012. 105

FURTADO, H.; VELHO, H. de C.; MACAU, E. Data assimilation: particle filter and artificial neural networks. **Journal of Physics: Conference Series**, v. 135, p. 012073, 2008. Institute of Physics Publishing. 7, 104, 124, 125

\_\_\_\_\_. **Assimilação de dados com redes neurais artificiais em equações diferenciais**. agosto 2011. 10<sup>a</sup> Conferência Brasileira de Dinâmica Controle e Aplicações. 82, 104

FURTADO, H. C. M. **Redes neurais e diferentes métodos de assimilação de dados em dinâmica não linear**. 125 p. Dissertação (Mestrado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2008-02-22 2008.

Disponível em: <<http://mtc-m17.sid.inpe.br/sid.inpe.br/mtc-m17@80/2008/02.07.10.49>>. 4, 5, 7, 8

FURTADO, H. C. M.; CAMPOS VELHO, H. F. de; MACAU, E. E. N. Neural networks for emulation variational method for data assimilation in nonlinear dynamics. **Journal of Physics: Conference Series**, v. 285, n. 1, p. 11, 2011. Institute of Physics Publishing. Disponível em: <[http://iopscience.iop.org/1742-6596/285/1/012036/pdf/\\$1742-6596\\_285\\_1\\_012036.pdf\\$](http://iopscience.iop.org/1742-6596/285/1/012036/pdf/$1742-6596_285_1_012036.pdf$)>. 7, 104

GHIL, M.; MALANOTTE-RIZZOLI, P. Data assimilation in meteorology and oceanography. **Advances in Geophysics**, n. 33, p. 141-266, 1991. 83

GOLUB, G.; LOAN, V. **Matrix computations**. Baltimore: Johns Hopkins University Press, 1989. 476 p. ISBN 64295204. 32

GOMES, V. C. F. **Fast Poisson solver para sistema híbrido reconfigurável**. 105 p. Dissertação (Mestrado) — Instituto Nacional de Pesquisas Espaciais

(INPE), São José dos Campos, 2012-05-25 2012. Disponível em:  
<<http://urlib.net/sid.inpe.br/mtc-m19/2012/05.10.19.52>>. Acesso em: 21  
set. 2012. 125

GORDON, N. J.; SALMOND, D.; SMITH, A. Novel approach to  
nonlinear/non-gaussian bayesian state estimation. **IEE Proceedings**, v. 140, n. 2,  
p. 107–113, Abril 1993. 3, 124

GRIFFITH, A. K. **Data assimilation for numerical weather prediction  
using control theory**. Tese (Doutorado) — University of Reading, Reading,  
Inglaterra, 1997-04- 1997. Disponível em:  
<[http://www.reading.ac.uk/web/FILES/maths/Ak\\_griffith.pdf](http://www.reading.ac.uk/web/FILES/maths/Ak_griffith.pdf)>. Acesso  
em: 28 maio 2010. 1, 4

HARTER, F. P. **Redes neurais recorrentes aplicadas a assimilação de  
dados em dinâmica não-linear**. 2004-12-16. 138 p. p. (INPE-14222-TDI/1123).  
Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas  
Espaciais, São José dos Campos, 2004. 3, 6, 7, 100, 103, 125

HÄRTER, F. P.; CAMPOS VELHO, H. F. Data assimilation procedure by  
recurrent neural network. **Engineering Applications of Computational Fluid  
Mechanics**, v. 6, n. 2, p. 224–233, 2012. 100, 103, 125

HÄRTER, F. P.; REMPEL, E. L.; Campos Velho, H. F. de; CHIAN, A.  
Application of artificial neural networks in auroral data assimilation. **Journal of  
Atmospheric and Solar - Terrestrial Physics**, v. 70, n. 10, p. 1243–1250,  
março 2008. 3, 125

HAYKIN, S. **Redes neurais princípios e prática**. Porto Alegre: Editora  
Bookman, 2004. 72, 74, 77, 80, 105

HAYKIN, S.; KAILATH, T. **Adaptive filter theory**. Fourth edition. Hamilton,  
Ontario, Canada: Pearson Education, 2002. 24, 25, 43

HÄRTER, F. P.; CAMPOS VELHO, H. F. de. New approach to applying neural  
network in nonlinear dynamic model. **Applied Mathematical Modelling**, v. 32,  
n. 12, p. 2621 – 2633, 2008. ISSN 0307-904X. Disponível em:  
<[http://www.sciencedirect.com/science/article/B6TYC-4PSC2CD-3/2/  
6808bbd6a3ea3626a06094d607a2b25a](http://www.sciencedirect.com/science/article/B6TYC-4PSC2CD-3/2/6808bbd6a3ea3626a06094d607a2b25a)>. 7

HSIEH, W. W.; TANG, B. Applying neural network models to prediction and data analysis in meteorology and oceanography. **American Meteorology Society**, v. 79, n. 9, p. 1855–1870, September 1998. 6

ISMAIL-ZADEH, A.; TACKLEY, P. J. **Computational methods for geodynamics**. First. United States of America: Cambridge University Press, 2010. 332 p. ISBN 978-0-521-86767-2. Disponível em: <[http://www.cambridge.org/gb/knowledge/isbn/item2711755/?site\\_locale=en\\_GB](http://www.cambridge.org/gb/knowledge/isbn/item2711755/?site_locale=en_GB)>. 1, 11, 13

JAZWINSKI, A. H. **Stochastic processes and filtering theory**. New York and London: Academic Press, 1970. 24, 32, 43, 124

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Journal of Basic Engineering**, n. 82, p. 35–45, 1960. 2, 83

KALMAN, R. E.; BUCY, R. S. New results in linear filtering and prediction theory. **Transactions of the AMSE**, n. 83, p. 35–44, 1961. Series D. 43, 83

KALNAY, E. **Atmospheric modeling, data assimilation and predictability**. 2. ed. New York: Cambridge University Press, 2003. 3, 4, 6, 43, 104, 124

KAY, S. M. **Fundamentals of statistical signal processing estimation theory - Upper Saddle River**. NJ: Prentice-Hall Ptr, 1993. ISBN 595301078. 41

KEPERT, J. Balance-aware covariance localisation for atmospheric and oceanic ensemble kalman filters. **Computational Geosciences**, Springer Netherlands, v. 15, p. 239–250, 2011. ISSN 1420-0597. 10.1007/s10596 – 010 – 9188 – 0. 43

KLEEMAN, L. **Understanding and applying Kalman filtering**. 2012.

Disponível em:

<<http://www.ecse.monash.edu.au/centres/irrc/LKPub/Kalman.PDF>>.

Acesso em: 10 de dez de 2012. 28

KOHONEN, T. **Self-organizing maps**. Berlin; Heidelberg; New-York: Springer, 1995. ISBN 3-540-58600-8. 77

KOVACS, Z. **Redes neurais artificiais: princípio e aplicações**. São Paulo: Collegium Cognito, 1996. 71

LAWSON, C. L.; HANSON, R. J. **Solving least squares problems**. Englewood Cliffs: NJ: Prentice-Hall Series in Automatic Computation, 1974. ISBN 34071496.

31

LORENC, A. C. Analysis methods for numerical weather prediction. **Quarterly Journal of the Royal Meteorological Society**, v. 112, p. 1177–1194, 1986. 123

LORENZ, E. N. Deterministic nonperiodic flow. **Journal of the Atmospheric Sciences**, v. 20, n. 2, p. 130–141, january 1963. 3, 104

\_\_\_\_\_. A study of the predictability of a 28-variable atmospheric model. **Tellus**, v. 17, p. 321–333, 1965. 104

LUZ, E. F. P. d. **Meta-heurísticas paralelas na solução de problemas inversos**. 155 p. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011-03-08 2012. Disponível em: <<http://urlib.net/sid.inpe.br/mtc-m19/2012/02.22.17.13>>. Acesso em: 14 nov. 2012. 81

LYNCH, L. **Numerical integration of linear and nonlinear wave equations**. 149 p. Dissertação (Mestrado) — Florida Atlantic University, Jupiter, Florida, December 2004. 13, 84

LYNCH, P. The slow equations. **Quarterly Journal of the Royal Meteorological Society**, v. 115, p. 201–219, jan 1989. Provided by the SAO/NASA Astrophysics Data System. Disponível em: <<http://adsabs.harvard.edu/abs/1989QJRMS.115..201L>>. 6

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115–133, 1943. 73, 74

MENDEL, J. **Lessons in estimation theory for signal processing communicatins, and control**. Prentice Hall: Englewood Cliffs:, 1995. 561 p. 35

MENDEL, J. M.; MCLAREN, R. W. Adaptive, learning and pattern recognition systems: Theory and applications. In: MENDEL, J. M.; FU, K. S. (Ed.). **Reinforcement learning control and pattern recognition systems**. New York: Academic Press, 1970. p. 287–318. 77

MESINGER, F.; ARAKAWA, A. Numerical methods used in atmospheric models. **GARP Publication**, WMO/ICSU Joint Organizing Committee, Geneva, Switzerland, p. 64 pp, 2002. Disponível em: <[http://www.atmos.ucla.edu/~brianpm/numerical\\_methods.html](http://www.atmos.ucla.edu/~brianpm/numerical_methods.html)>. Acesso em: 15 de Março de 2011. 19

MOLTENI, F. Atmospheric simulations using a gcm with simplified physical parametrizations. i: model climatology and variability in multi-decadal experiments. **Climate Dynamics**, Springer Berlin / Heidelberg, v. 20, p. 175–191, 2003. ISSN 0930-7575. 10.1007/s00382-002-0268-2. Disponível em: <<http://dx.doi.org/10.1007/s00382-002-0268-2>>. 8

MUCCINO, J. C.; ARANGO, H. G.; BENNETT, A. F.; CHUA, B. S.; CORNUELLE, B.; LORENZO, E. D.; EGBERT, G.; HAIDVOGEL, D.; LEVIN, J. C.; LEVIN, J. C.; LUO, H.; MILLER, A. J.; MOORE, A. M.; ZARON, E. D. The inverse ocean modeling system. part ii: Applications. **American Meteorological Society**, p. 1623–1637, 2008. 67

MUCCINO, J. C.; BENNETT, A. F. Generalized inversion of the korteweg-de vries equation. **Dynamics of atmospheres and oceans**, 05 April 2002. Disponível em: <[http://www.sciencedirect.com/science?\\_ob=ArticleURL&\\_udi=B6VCR-45S9CNJ-1&\\_user=972035&\\_rdoc=1&\\_fmt=&\\_orig=search&\\_sort=d&view=c&\\_acct=C000049643&\\_version=1&\\_urlVersion=0&\\_userid=972035&md5=38faf9cd9dee0e9f847cd97396448757](http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6VCR-45S9CNJ-1&_user=972035&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000049643&_version=1&_urlVersion=0&_userid=972035&md5=38faf9cd9dee0e9f847cd97396448757)>. Acesso em: 15 de Março de 2009. 66

NOWOSAD, A.; Campos Velho, H.; NETO, A. R. Neural network as a new approach for data assimilation. In: CONGRESSO BRASILEIRO DE METEOROLOGIA, 1., 2000, Rio de Janeiro. **Proceedings...** Rio de Janeiro: SBMET, 2000. p. 3078–3086. ISBN PT00002. 6, 124, 125

NOWOSAD, A. G. **Novas abordagens de assimilação de dados meteorológicos**. 2001-03-14. 120 p. p. (INPE-8481-TDI/779). Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais, 2001. 3, 7, 103, 125

PAES, F. F. **Neuro estimador para o fluxo de gases entre a superfície terrestre e a atmosfera**. 151 p. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011-11-21 2011. Disponível em: <<http://urlib.net/sid.inpe.br/mtc-m19/2011/10.24.11.20>>. Acesso em: 28 maio 2012. 71

PAPOULIS, A. **Probability, random variables, and stochastic processes**. New York and London: McGraw-Hill, 1984. 124

PETRY, A.; SOUZA, J. R.; Campos Velho, H. F. de; PEREIRA, A. G.; BAILEY, G. J. Operational ionospheric dynamics prediction for south america. In:

INTERNATIONAL CONFERENCE ON INTEGRAL METHODS IN SCIENCE AND ENGINEERING IMSE2012, 12., 2012, Bento Gonçalves, Rio Grande do Sul, Brazil. **Proceedings...** Porto Alegre: IMSE, 2012. 3

POLAVARAPU, S. **Introduction to estimation theory**. 2004. Lectures notes. Disponível em: <<http://www.atmosph.physics.utoronto.ca/PHY2509/>>. Acesso em: 28 de julho de 2012. 23, 24, 25, 32, 42, 84

RANDALL, D. A. 2006. Department of Atmospheric Science. Disponível em: <<http://kiwi.atmos.colostate.edu/group/dave/pdf/ShallowWater.pdf>>. Acesso em: 06 abril 2009. 17

ROBINSON, A. R.; LERMUSIAUX, P. F. J. **Overview of data assimilation**. 2000. Harvard Reports in Physical/Interdisciplinary. Disponível em: <[http://robinson.seas.harvard.edu/PAPERS/red\\_report\\_62.html](http://robinson.seas.harvard.edu/PAPERS/red_report_62.html)>. Acesso em: 12 abril 2012. 1, 2, 23, 24

ROCHA, H. P. **Abordagens multi-objetivo para o treinamento de redes neurais e seleção de características**. 80 p. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Minas Gerais, Brasil, Março 2012. 81

SAMBATTI, S. B. M.; FURTADO, H. C. M.; ANOCHI, J. A.; LUZ, E. F. P. da; Campos Velho, H. F. de. Automatic configuration of an artificial neural network with application to data assimilation. In: INTERNATIONAL CONFERENCE ON INTEGRAL METHODS IN SCIENCE AND ENGINEERING IMSE2012, 12., 2012, Bento Gonçalves, Rio Grande do Sul, Brazil. **Proceedings...** Porto Alegre: IMSE, 2012. 81, 82, 125

SAMPSON, J. **Some solutions of the shallow water wave equations**. Tese (Thesis) — Faculty of Engineering and Industrial Sciences Swinburne University of Technology Melbourne, Australia, Maio 2008. Disponível em: <<http://researchbank.swinburne.edu.au/vital/access/manager/Repository/swin:8483>>. 17

SCHERLIESS, L.; SCHUNK, R. W.; SOJKA, J. J.; THOMPSON, D. C. Development of a physics-based reduced state kalman filter for the ionosphere. **Radio Science**, v. 39, n. RS1S04, June 2004. Doi:10.1029/2002RS002797. 3

SHIGUEMORI, E. H. **Recuperação de perfis de temperatura e umidade da atmosfera a partir de dados de satélite - abordagens por redes neurais**

**artificiais e implementação em hardware.** Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2007-05-22 2007.

Disponível em:

<<http://urlib.net/sid.inpe.br/mtc-m17@80/2007/07.31.12.19>>. Acesso em: 19 julho. 2012. 71, 125

SONG, H.; EDWARDS, C. A.; MOORE, A. M.; FIECHTER, J. Incremental four-dimensional variational data assimilation of positive-definite oceanic variables using a logarithm transformation. **Ocean Modelling**, v. 54-55, n. 0, p. 1 – 17, 2012. ISSN 1463-5003. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S146350031200090X>>. 43

SORENSEN, H. W. Least-squares estimation: from gauss to kalman. **IEEE Spectrum**, v. 7, p. 63–68, July 1970. 23, 29, 32, 40

STRANG, G. **Introduction to applied mathematics.** Wellesley, Massachusetts: Wellesley-Cambridge Press, 1986. Massachusetts Institute of Technology. ISBN 0-9614088-0-4. 29, 32, 33, 34, 35, 38, 39

TALAGRAND, O.; COURTIER, P. Variational assimilation of meteorological observations with the adjoint vorticity equation. **Quart. J. Roy. Meteor. Soc.**, v. 113, p. 1311–1328, 1987. 3

TANGBORN, A. Wavelet approximation of error covariance propagation in data assimilation. **Tellus**, v. 56, n. 1, 2004. Disponível em: <<http://journals.sfu.ca/coaction/index.php/tellusa/article/view/14388>>. Acesso em: 04 de outubro de 2012. 43

TATIBANA, C. Y.; KAETSU, D. Y. **Homepage de Redes Neurais.** 2012. Disponível em: <<http://www.din.uem.br/ia/neurais/>>. Acesso em: 28 de maio 2012. 72

TEIXEIRA, R. de A.; BRAGA, A. de P.; TAKAHASHI, R. H.; SALDANHA, R. R. A multi-objective optimization approach for training artificial neural networks. **Neural Networks, Brazilian Symposium on**, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 168, 2000. ISSN 1522-4899. 81

TODLING, R. **Estimation theory and foundations of atmospheric data assimilation.** 1999. Technical Report DAO Office Note 1999-01, Data

Assimilation Office, Goddard Space Flight Center, 1999. Disponível em:  
<[citeseer.ist.psu.edu/todling99estimation.html](http://citeseer.ist.psu.edu/todling99estimation.html)>. 42

VIJAYKUMAR, N. L.; Campos Velho, H. F.; STEPHANY, S.; PRETO, A. J.;  
NOWOSAD, A. G. A neural network implementation for data assimilation using  
mpi, application of high performance computing in engineering. In: BREBIA, C. A.;  
MELLI, P.; ZANASI, A. (Ed.). **Application of high performance computing  
in engineering**. Southampton: WIT Press, 2002. p. Section 5, 211–220.

Disponível em: <Disponível em: <<http://urlib.net/sid.inpe.br/mtc-m17@80/2007/08.14.13.09>>. Acesso em: 22abr.2008.>. 6

WEI-MIN, Z.; XIAO-QUN, C.; JUN-QIANG, X. Q. nong S.; XIAO-QIAN, Z.;  
SHU-CHANG, W. Variational data assimilation using wavelet background error  
covariance: initialization of typhoon kaemi. **Journal of Tropical Meteorology**,  
v. 16, n. 4, december 2010. Acesso em: 04 de outubro de 2012. 43

ZARON, E. D. **Introduction to ocean data assimilation**. 2009. Disponível em:  
<[ftp:  
//ftp.marine.csiro.au/pub/GODAE-2010/Edward20Zaron\\_Intro\\_Rev2.pdf](ftp://ftp.marine.csiro.au/pub/GODAE-2010/Edward20Zaron_Intro_Rev2.pdf)>.  
Acesso em: 12 abril 2009. 1



## APÊNDICE A - DERIVANDO A EQUAÇÃO DE EULER LAGRANGE

Considere o problema avançado dado pela EDO a seguir:

$$\frac{d\eta}{dt} = 1 \quad 0 \leq t \leq 4 \quad (\text{A.1})$$

com condição inicial  $\eta(0) = 0$  e um dado de observação  $\eta(1) = 3$ . Para a formulação do problema de estimação, a equação de evolução é redefinida como:

$$\frac{d\eta}{dt} = 1 + f(t) \quad 0 \leq t \leq 4; \quad \eta(0) = i \quad (\text{A.2})$$

$$\eta(1) = 3 + \varepsilon \quad (\text{A.3})$$

onde  $f(t)$ ,  $i$  e  $\varepsilon$  são residuais na dinâmica, condição inicial e observação, respectivamente. A hipótese de nulidade, consiste da seguinte definição em torno da média e covariância dos residuais, dados por:

$$\langle f(t) \rangle = \langle i \rangle = \langle \varepsilon \rangle = 0 \quad (\text{A.4a})$$

$$\langle f(t)f(s) \rangle = V_f \delta(t-s); \quad \langle ii \rangle = V_i; \quad \langle \varepsilon \varepsilon \rangle = V_\varepsilon \quad (\text{A.4b})$$

$$\langle fi \rangle = \langle f\varepsilon \rangle = \langle i\varepsilon \rangle = 0 \quad (\text{A.4c})$$

em que  $\langle \rangle$  indica o valor esperado e  $\delta(t-s)$  é a função delta de Dirac. As variâncias ( $V_f, V_i, V_\varepsilon$ ) são assumidas serem conhecidas e constantes. O objetivo é minimizar os erros existentes no modelo e na observação, ou seja,

$$f(t) = \frac{d\eta}{dt} - 1 \quad \text{e} \quad (\text{A.5a})$$

$$\varepsilon = \eta(1) - 3 \quad (\text{A.5b})$$

Essa redução é feita por um ajuste ponderado de mínimos quadrados, isto é:

$$J = J[\eta(t)] = W_f \int_0^T dt \left\{ \frac{d\eta}{dt} - 1 \right\}^2 + W_i \{i\}^2 + w \{\eta(1) - 3\}^2 \quad (\text{A.6})$$

A minimização do funcional A.6 é feita por meio do cálculo das variações. Deseja-se determinar  $\hat{\eta}(t)$  tal forma que o funcional  $J$  seja mínimo, isto é:  $\delta J = 0 \Rightarrow \nabla J = 0$ . Expandindo-se o funcional  $J$  em torno do ponto extremo local  $\hat{\eta}(t)$ , obtém-se:

$$J[\hat{\eta} + \delta\eta] = J[\hat{\eta}] + O(\delta\eta)^2 \quad (\text{A.7})$$

Pode-se perceber que  $W_i = 0$ , pois o funcional precisa ser minimizado. Deste modo,

$$J[\hat{\eta}] = W_f \int_0^T dt \left\{ \left( \frac{d\hat{\eta}}{dt} \right)^2 - 2 \frac{d\hat{\eta}}{dt} + 1 \right\} + w \{ \hat{\eta}^2(1) - 6\hat{\eta}(1) + 9 \} \quad (\text{A.8})$$

Analogamente,

$$J[\hat{\eta} + \delta\eta] = W_f \int_0^T dt \left\{ \left( \frac{d\hat{\eta}}{dt} \right)^2 + \left( \frac{d\delta\eta}{dt} \right)^2 + 2 \frac{d\hat{\eta}}{dt} \frac{d\delta\eta}{dt} - 2 \frac{d\hat{\eta}}{dt} - \frac{d\delta\eta}{dt} + 1 \right\} + w \{ \hat{\eta}^2(1) + \delta\eta^2(1) + 2\hat{\eta}(1)\delta\eta(1) - 6\hat{\eta}(1) - 6\delta\eta(1) + 9 \} \quad (\text{A.9})$$

A primeira variação para o funcional  $J$  pode ser descrita como:

$$\delta J = J[\hat{\eta} + \delta\eta] - J[\hat{\eta}] \quad (\text{A.10})$$

Assim, substituindo as equações A.8 e A.9 em A.10, obtém-se a primeira variação para  $J$  dada por:

$$\delta J = W_f \int_0^T dt \left\{ \left( \frac{d\delta\eta}{dt} \right)^2 + 2 \frac{d\hat{\eta}}{dt} \frac{d\delta\eta}{dt} - 2 \frac{d\delta\eta}{dt} \right\} + w \{ 2\hat{\eta}(1)\delta\eta(1) - 6\delta\eta(1) \} \quad (\text{A.11})$$

Desprezando os termos de segunda ordem, tem-se:

$$\delta J = W_f \int_0^T dt \left\{ 2 \frac{d\hat{\eta}}{dt} \frac{d\delta\eta}{dt} - 2 \frac{d\delta\eta}{dt} \right\} + w \{ 2\hat{\eta}(1)\delta\eta(1) - 6\delta\eta(1) \} + O(\delta\eta)^2 \quad (\text{A.12})$$

O funcional A.12 pode ser escrito como:

$$\delta J = W_f \int_0^T dt \left\{ \frac{d\hat{\eta}}{dt} - 1 \right\} \left\{ 2 \frac{d\delta\eta}{dt} \right\} + w \{ (\hat{\eta}(1) - 3)(2\delta\eta(1)) \} + O(\delta\eta)^2 \quad (\text{A.13})$$

Define-se o residual ponderado  $\lambda(t)$  por:

$$\lambda(t) \equiv W_f \left\{ \frac{d\hat{\eta}}{dt} - 1 \right\} \quad (\text{A.14})$$

Substituindo a equação A.14 em A.13 tem-se:

$$\delta J = \int_0^T dt \lambda(t) \left\{ 2 \frac{d\delta\eta}{dt} \right\} + w \{ [\hat{\eta}(1) - 3](2\delta\eta(1)) \} + O(\delta\eta)^2 \quad (\text{A.15})$$

Por simplicidade, defini-se  $W_f = w = 1$ . Integrando por partes a equação A.15,

$$\int_0^T \lambda(t) \left\{ 2 \frac{d\delta\eta}{dt} \right\} dt = \lambda(T)2\delta\eta(T) - \lambda(0)2\delta\eta(0) - \int_0^T \frac{d\lambda}{dt} 2\delta\eta dt \quad (\text{A.16})$$

Substituindo A.16 em A.15, tem-se:

$$\delta J = - \int_0^T \frac{d\lambda}{dt} 2\delta\eta dt + \lambda(T)2\delta\eta(T) - \lambda(0)2\delta\eta(0) + \{(\hat{\eta}(1) - 3)(2\delta\eta(1))\} + O(\delta\eta)^2 \quad (\text{A.17})$$

Empregando-se a propriedade função delta de Dirac:

$$\int_0^T \{[\hat{\eta}(1) - 3](2\delta\eta)\delta(t - 1)\} dt = \{[\hat{\eta}(1) - 3]2\delta\eta(1)\} . \quad (\text{A.18})$$

Substituindo A.18 em A.17, tem-se:

$$\delta J = - \int_0^T \left\{ \left[ \frac{d\lambda}{dt} + (\hat{\eta}(1) - 3)\delta(t - 1) \right] 2\delta\eta \right\} dt + \lambda(T)2\delta\eta(T) - \lambda(0)2\delta\eta(0) \quad (\text{A.19})$$

Portanto,

$$\frac{d\lambda}{dt} = -[\hat{\eta}(1) - 3]\delta(t - 1) \quad (\text{A.20})$$

Com o residual definido anteriormente:

$$\frac{d\hat{\eta}}{dt} = \lambda(t) + 1 \quad (\text{A.21})$$

definem as equações de Euler Lagrange.



## APÊNDICE B - Estrutura de chamada dos programas do *software* IOM

```

<?xml version="1.0"?>
<!-- inverse model name and type -->
<inversion name="swL2D" type="linear">
<!-- directories -->
  <iom_comp_dir>../iom</iom_comp_dir>
  <iom_run_dir>../run</iom_run_dir>
  <iom_sharedata_dir>../sharedata</iom_sharedata_dir>
  <iom_cgdata_dir>../cgdata</iom_cgdata_dir>
  <iom_cgdata_save_dir>../cgdata_save</iom_cgdata_save_dir>
  <iom_outputs_dir>../outputs</iom_outputs_dir>
  <iom_log_dir>../log</iom_log_dir>
<!-- user-supplied components -->
  <ker_filename>kernel_obs.bin</ker_filename>
  <ker_dir>../ker</ker_dir>
  <vect_d_filename>inver_par_obs.nc</vect_d_filename>
  <vect_d_dir>../inputs</vect_d_dir>
  <vect_d_var>observation_vector</vect_d_var>
  <matr_Ceps_filename>inver_par_obs.nc</matr_Ceps_filename>
  <matr_Ceps_dir>../inputs</matr_Ceps_dir>
  <matr_Ceps_var>measurement_error_covariance_matrix</matr_Ceps_var>
  <matr_Ceps_type>diagonal</matr_Ceps_type>
  <traj_std_err_filename>traj_model_std_err.nc</traj_std_err_filename>
  <traj_std_err_dir>../inputs</traj_std_err_dir>
<!-- user-supplied components (for both linear and nonlinear model) -->
  <component name="fwdExt" supplied="user" dir="../model" exe="swL2D_fwdExt">
    <input_file>traj_convolved.nc</input_file>
    <output_file>traj.nc</output_file>
  </component>
  <component name="adj" supplied="user" dir="../model" exe="swL2D_adj">
    <input_file>traj_comb.nc</input_file>
    <output_file>traj_adjoint.nc</output_file>
  </component>
  <component name="meas" supplied="iom" dir="../iom" exe="iom_meas">
  </component>
  <component name="comb" supplied="iom" dir="../iom" exe="iom_comb">
  </component>
  <component name="convol" supplied="iom" dir="../iom" exe="iom_convol">
  </component>
  <component name="convol_half" supplied="iom" dir="../iom"
exe="iom_convol_half">
    <input_file>traj_adjoint_direct_representer_1.nc</input_file>
  </component>
  <component name="convol_initial" supplied="iom" dir="../iom"
exe="iom_convol_initial"/>
  <component name="user_convol_initial" supplied="user" dir="../model"
exe="swL2D_convol_initial">
    <output_file_dir>../sharedata</output_file_dir>
    <output_file>swL2D_convolved_initial.nc</output_file>
  </component>
<!-- iom-supplied components -->
  <component name="comp_penalty" supplied="iom" dir="../iom"
exe="iom_comp_penalty"/>
  <component name="comp_vect" supplied="iom" dir="../iom"
exe="iom_comp_vect"/>
  <component name="comp_matr" supplied="iom" dir="../iom"
exe="iom_comp_matr"/>
  <component name="form_matr" supplied="iom" dir="../iom"
exe="iom_form_matr"/>
  <component name="comp_traj" supplied="iom" dir="../iom"
exe="iom_comp_traj"/>
  <component name="form_traj" supplied="iom" dir="../iom"
exe="iom_form_traj"/>

```

opção do modelo  
swL2D ou wave1D

arquivo das observações

opções do IOM

```

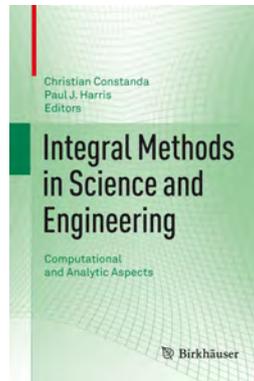
<number_of_samples>10</number_of_samples>
<number_of_inner_iterates>3</number_of_inner_iterates> número de observações >8</number_of_inner_iterates>
<stopping_tolerance>1.0d-5</stopping_tolerance>
<number_of_test_samples></number_of_test_samples>
<inner_solver_restart_index></inner_solver_restart_index>
<number_of_spectral_vectors></number_of_spectral_vectors>
<number_of_bfgs_vectors></number_of_bfgs_vectors>
<number_of_representers></number_of_representers>
<var name="lambda_u" type="correlated">
  <space_len_dc idx="1">100.0d4</space_len_dc>
  <space_len_dc idx="2">100.0d4</space_len_dc>
  <time_len_dc>1800.0</time_len_dc>
</var>
<var name="lambda_v" type="correlated">
  <space_len_dc idx="1">100.0d4</space_len_dc>
  <space_len_dc idx="2">100.0d4</space_len_dc>
  <time_len_dc>1800.0</time_len_dc>
</var>
<var name="lambda_q" type="correlated">
  <space_len_dc idx="1">100.0d4</space_len_dc>
  <space_len_dc idx="2">100.0d4</space_len_dc>
  <time_len_dc>1800.0</time_len_dc>
</var>
<var name="xi_u" type="correlated">
  <space_len_dc idx="1">100.0d4</space_len_dc>
  <space_len_dc idx="2">100.0d4</space_len_dc>
</var>
<var name="xi_v" type="correlated">
  <space_len_dc idx="1">100.0d4</space_len_dc>
  <space_len_dc idx="2">100.0d4</space_len_dc>
</var>
<var name="xi_q" type="correlated">
  <space_len_dc idx="1">100.0d4</space_len_dc>
  <space_len_dc idx="2">100.0d4</space_len_dc>
</var>
<var name="gamma_v_0" type="correlated">
  <space_len_dc idx="1">100.0d4</space_len_dc>
  <time_len_dc>1800.0</time_len_dc>
</var>
<var name="gamma_v_L" type="correlated">
  <space_len_dc idx="1">100.0d4</space_len_dc>
  <time_len_dc>1800.0</time_len_dc>
</var>
</parameters>
<!-- inversion outputs -->
<outputs>
  <iom_cg_cold_start/>
  <iom_compute_penalty_posterior/>
  <iom_compute_penalty_prior/>
  <iom_compute_representer_coefficients/>
  <iom_compute_traj_optimal/>
  <iom_compute_traj_prior/>
  <iom_compute_vect_h/>
  <iom_method_indirect_representer/>
  <iom_solver_conjugate_gradient/>
  <iom_compute_multivariate_cov_prior/>
</outputs>
</inversion>

```

parâmetros para convolução

opções da assimilação realizada pelo IOM

## ANEXO A - Publicações



### Adaptive Particle Filter for Stable Distribution

H.F. de Campos Velho and H.C. Morais Furtado

#### 1 Introduction

Estimation theory is a central issue for several applications: filtering, signal analysis, image processing, control theory, inverse problem, and data assimilation. There is intensive research into developing a set of techniques for estimating quantities, for example: the least squares approximation, Kalman filter, variational method, Bayesian approach and, more recently, schemes based on artificial intelligence: neural network and fuzzy logic.

For all the methods cited, there are some constraints to be considered. For example, in the context of inverse problem, it is important to include some *a priori* information to apply the least square solution. Some authors use the expression *generalized least squares* to characterize the latter condition. Kalman filter is another alternative, and it was developed for a linear dynamical system under Gaussian assumptions for describing the probability density function (PDF) for the variables involved. The extended Kalman filter is an adaptation to employ the filter to nonlinear problems. But this strategy can fail under strong nonlinear regimes. Variational methods are standard procedures in estimation problems. The goal is to calculate a minimum of an objective function (a functional), in a similar way to the least square approach, but the gradient of the functional is computed by using the adjoint function associated to the problem.

For Kalman filter and the variational method, one quantity remains unknown: the covariance matrix for representing the modeling error. Some schemes were designed to identify this covariance matrix: the use of an estimating operator, as a secondary

---

H.F. de Campos Velho  
Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP, Brazil.  
e-mail: [haraldo@iac.inpe.br](mailto:haraldo@iac.inpe.br)

H.C. Morais Furtado  
Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP, Brazil.  
e-mail: [helaine.furtado@iac.inpe.br](mailto:helaine.furtado@iac.inpe.br)

C. Constanda, P.J. Harris (eds.), *Integral Methods in Science and Engineering*.

47

## Data assimilation: particle filter and artificial neural networks

Helaine Cristina Morais Furtado,  
Haroldo Fraga de Campos Velho,  
Elbert Einstein Nehrer Macau

Laboratório Associado de Computação e Matemática Aplicada (LAC), Instituto Nacional de Pesquisas Espaciais (INPE), P. O. Box 515, 12245-970, São José dos Campos, SP, Brazil

E-mail: helaine.furtado@lac.inpe.br; haroldo@lac.inpe.br; elbert@lac.inpe.br

**Abstract.** The goal of this work is to present the performance of the Neural Network Multilayer Perceptrons trained to emulate a Particle Filter in the context of data assimilation. Techniques for data assimilation are applied for the Lorenz system, which presents a strong nonlinearity and chaotic nature. The cross validation method was used for training the network. Good results were obtained applying the multilayer perceptrons neural network.

### 1. Introduction

Data assimilation is an essential feature, emerging from operational process for prediction systems. Such operational procedures were identified as a fundamental issue for the early time in the centers for the numerical weather prediction (NWP). Today, data assimilation is a routine process in the NWP [1, 2], ocean circulation forecasting [3], air monitoring [4], and space weather models [5, 6]. This is a science to produce a nice combination of data from a mathematical model and from observations for the numerical prediction systems.

From a mathematical point of view, the data assimilation process can be represented by

$$w^a = w^f + K p [y - H(w^f)] \quad (1)$$

where  $w^a$  is the value of the analysis;  $w^f$  is the forecasting (from the mathematical model, also known as background field);  $K$  is the weight matrix;  $y$  denotes the observation;  $H$  represents the observation system; the difference  $\{y - H(w^f)\}$  is the innovation; and  $p[\cdot]$  is a discrepancy function. Another approach for solving the data assimilation is by computing the minimum solution for the cost function:

$$J(w) = \langle (w - w^f), C(w - w^f) \rangle + \langle [y - H(w^f)], S[y - H(w^f)] \rangle \quad (2)$$

where  $C$  and  $S$  denote the inverse covariance matrices of the background (in general such matrix is represented by  $B$  or  $Q$  for the data assimilation or for the control theory communities – respectively) and of the measurement errors (denoted by  $R$ ), and  $\langle u, v \rangle$  expresses the internal product between vectors  $u$  and  $v$ . For the Kalman filter (KF), the matrix  $K$  (Kalman gain) is computed from a formula involving matrices  $B$  and  $R$ . Variational scheme is linked to the optimization problem with the objective function given by expression (2).

# Neural networks for emulation variational method for data assimilation in nonlinear dynamics

Helaine Cristina Morais Furtado

Haroldo Fraga de Campos Velho

Elbert E. N. Macau

Laboratório Associado de Computação e Matemática Aplicada, São José dos Campos, Brasil  
E-mail: helaine.furtado@lac.inpe.br, haroldo@lac.inpe.br, elbert@lac.inpe.br

**Abstract.** Description of a physical phenomenon through differential equations has errors involved, since the mathematical model is always an approximation of reality. For an operational prediction system, one strategy to improve the prediction is to add some information from the real dynamics into mathematical model. This additional information consists of observations on the phenomenon. However, the observational data insertion should be done carefully, for avoiding a worse performance of the prediction. Technical data assimilation are tools to combine data from physical-mathematics model with observational data to obtain a better forecast. The goal of this work is to present the performance of the Neural Network Multilayer Perceptrons trained to emulate a Variational method in context of data assimilation. Techniques for data assimilation are applied for the Lorenz systems; which presents a strong nonlinearity and chaotic nature.

## 1. Introduction

Data assimilation is nowadays prevailing process in the Numerical Weather Prediction (NWP)[1, 2], ocean circulation forecasting [3], air monitoring [4] and space weather models [5, 6]. Technical data assimilation can be defined as a set of methods that can be used to combine data from physical-mathematics model with observational data to obtain a better forecast.

From filtering point of view, data assimilation process can be represented by:

$$w^a = w^f + Kp[y - H(w^f)] \quad (1)$$

where  $w^a$  is the value of the analysis;  $w^f$  is the forecasting (from the mathematical model, also known as background field);  $K$  is the weight matrix;  $y$  denotes the observation;  $H$  represents the observation system; the difference  $\{y - H(w^f)\}$  is the innovation; and  $p[\cdot]$  is a discrepancy function. Another approach for solving the data assimilation is by computing the minimum solution for the cost function:

$$J(w) = \langle (w - w^f), C(w - w^f) \rangle + \langle [y - H(w^f)], S[y - H(w^f)] \rangle \quad (2)$$



**DINCON 2011**  
**10ª Conferência Brasileira de Dinâmica,  
Controle e Aplicações**  
**28 de agosto a 1º de setembro de 2011**



**ASSIMILAÇÃO DE DADOS COM REDES NEURAIS ARTIFICIAIS EM EQUAÇÕES DIFERENCIAIS**

*H. C. M. Furtado, H. F. de Campos Velho, E. E. N. Macau*

Instituto Nacional de Pesquisas Espaciais (INPE), Laboratório Associado de Computação (LAC)

São José dos Campos, Brasil, e-mail: [helaine.furtado, haroldo, elbert]@lac.inpe.br

**Resumo:** Assimilação de dados é uma ferramenta que combina de modo eficiente dados de observação com dados de um modelo físico-matemático para a determinação do dado de análise, que é usado para executar o modelo de previsão. Esse processo é fundamental, por exemplo, na prática operacional da previsão numérica do tempo e em circulação oceânica. Com o desenvolvimento do sistema de observação e a evolução dos modelos numéricos, a quantidade de dados disponíveis para serem assimilados é da ordem de  $10^5$  a  $10^7$ , enquanto que, o número de pontos de grade do modelo é da ordem de  $10^6$  a  $10^8$ . Isso representa uma computação intensiva e é um grande desafio desenvolver técnicas que possam produzir um resultado de análise o mais preciso possível, para produzir boa capacidade de previsão, o mais eficiente possível e tornar o custo de computação operacionalmente viável, em um contexto em que o número de pontos de grades e de observações são exponencialmente crescentes. Este trabalho tem como objetivo apresentar a ferramenta de redes neurais artificiais como uma técnica de assimilação de dados

**Palavras-chave:** assimilação de dados, redes neurais artificiais, Filtro de Kalman, sistema dinâmico de Lorenz, equação da onda em uma dimensão

## 1. INTRODUÇÃO

Descrever fenômenos físicos a partir de equações diferenciais (parcial ou ordinária), é uma atividade passível de erro, uma vez que o modelo matemático é sempre uma aproximação da realidade. Um modo de “melhorar” esta descrição é adicionando informações ao modelo. Estas informações consistem de *observações*, que são dados provenientes de medidas sobre o fenômeno que se deseja modelar. No entanto, ao usar observações para estimar alguma propriedade do modelo matemático (condições iniciais, condições de contorno, propriedades do material e/ou da dinâmica, termos fontes) o problema de estimação é frequentemente um prob-

lema *mal posto* [1].

Para determinar condições iniciais, a alternativa para a solução do problema é combinar de modo eficiente as observações com o modelo matemático, por meio da utilização de técnicas de assimilação de dados, que podem ser baseadas em teoria da estimação (como: filtro de Kalman [2], filtro de partículas [3, 4]); baseadas no cálculo variacional [5]; ou ainda, baseadas em inteligência artificial, com o uso de Redes Neurais Artificiais [7].

Como já bem conhecido, previsão numérica do tempo (PNT) é um problema de valor inicial [8]:

*“Dado uma estimativa do estado atual da atmosfera (condições iniciais), superfície apropriada e condições de contorno laterais, o modelo simula (prever) a sua evolução.”* No entanto, mesmo considerando por hipótese que, os modelos determinísticos e as observações fossem perfeitas, a natureza *caótica* da atmosfera limitaria o tempo de previsibilidade. Essa questão foi primeiramente observado no trabalho de Lorenz [9], em que a solução das equações semelhantes às que governam os movimentos atmosféricos apresentaram dependência sensível com relação a condição inicial, conforme ilustrado na Figura 1. A Figura 2 ilustra a aplicação da técnica de assimilação de dados. Nota-se que à medida que são inseridas as observações no sistema, a dinâmica da estimativa (curva vermelha) está próxima da referência de verdade (curva azul), uma vez que é interrompido a inserção de observações, ocorre o desacoplamento das dinâmicas. Portanto, as observações, quando combinadas de modo eficiente com dados do modelo, servem para aproximar a estimativa da condição inicial ao valor desejado.

A base dos algoritmos de assimilação de dados podem ser expressas como uma correção ou seqüência de correções dos estados do modelo. Este processo consiste de duas fases que são: a fase de *previsão* e a fase de *análise* ou *incremento*, representados matematicamente por:

## Data assimilation by neural network emulating representer method applied to the wave equation

Helaine C. M. Furtado<sup>1</sup>, Haroldo F. de Campos Velho<sup>2</sup>, Elbert E. N. Macau<sup>2</sup>

<sup>1</sup> Pos Applied Computing, helaine.furtado@lac.inpe.br

<sup>2</sup> Computing and Applied Mathematics Laboratory (LAC), haroldo@lac.inpe.br and elbert@lac.inpe.br  
National Institute For Space Research - INPE  
Avenida do Astronautas, 1758, São José dos Campos, SP, Brazil

**Abstract.** Description of a physical phenomenon through differential equations has errors involved, since the mathematical model is always an approximation of reality. For an operational prediction system, one strategy to deal with uncertainties from the modeling and observation errors is to add some information from the real dynamics into mathematical model. This additional information consists of observations on the phenomenon. However, the observational data insertion should be done carefully, for avoiding a worse performance of the prediction. Technical data assimilation are tools to combine data from physical-mathematics model with observational data to obtain a better forecast. Two data assimilation methods are compared here: the Representer method, and artificial neural network. Artificial neural networks appear as a novel method in the context for data assimilation. The performance of the methods is evaluated under application to wave propagation model (Bennet, 2002).

**Keywords.** data assimilation, neural network, variational method, representer method, wave equation.

### 1 INTRODUCTION

In recent years, data assimilation methods have been developed and used in many research areas, such as: numerical weather prediction (NWP) (Daley, 1993; Kalnay, 2003); ocean circulation forecasting (Bennett, 2002); air monitoring (Zannetti, 1990); and space weather models (Garner, 1999; Schunk et.al, 2004). However, the first application of data assimilation techniques was in meteorology, and today it is a key component of numerical weather and climate forecasts. Satellites and *in situ* measurements are routinely providing new atmospheric and oceanographic data and bringing the daily practice of physical oceanography closer to that of dynamic meteorology (Belyaev, 2000). Essentially, NWP consists of the integration of Navier-Stokes equation using numerical procedures. Therefore, after some time-steps, there is a disagreement due to small uncertainties in the a specification of the initial conditions and model errors. In other words, the sensitive dependence on the initial conditions causes the forecasting error to grow exponentially with time (Grebogi, 1987). The problem of estimating the initial conditions is called *data assimilation* that improves the estimate of the ocean and atmosphere physical state by combining the data from measurements and from dynamic models in an optimal way.

Many methods have been developed for data assimilation (Kalnay, 2003; Härter, 2008). They have different strategies to combine the forecasting (*background*) and observations. From a mathematical point view, the assimilation process can be represented by:

$$\mathbf{x}^a = \begin{cases} \mathbf{x}^f + \mathbf{K}(\mathbf{y} - \mathbf{H}[\mathbf{x}^f]), & \text{estimation theory} \\ \nabla J(\mathbf{x}), & \text{variational method} \\ \mathbf{F}_w(\mathbf{x}^f, \mathbf{y}^o), & \text{artificial intelligence} \\ \mathbf{x}^f + \sum_{m=1}^M \beta_m r_m(x, t), & \text{representer method} \end{cases} \quad (1)$$

For methods based on the estimation theory, it can be cited the Optimal interpolation (Daley, 1993), Kalman Filter (KF) (Kalnay, 2003), and Particle Filter (PF) (Nakano, 2007). In the analysis  $\mathbf{x}^a$  for KF,  $\mathbf{x}^f$  is the forecasting (from mathematical model, also known as *background* field);  $\mathbf{K}$  is the weight matrix (Kalman gain), generally computed from the covariance matrix of the predictions errors from forecasting and observation;  $\mathbf{y}$  denotes the observation;  $\mathbf{H}$  represents the observation system. For the variational methods, the analysis is obtained by the minimizing a functional. This method will be described in the section 3.1. For the method based on artificial intelligence analysis is determined by mapping the input data with the desired output data that described in the section 3.2. The representer method is present in the section 3.1.

### 2 WAVE EQUATION: A TESTING MODEL

Data assimilation is a necessary process for operational prediction systems. The insertion of observational data can degenerate the prediction. Therefore, techniques allowing a soft introduction of the observations in a mathematical prediction model are called data assimilation. Two schemes will be compare for data assimilation, and the testing model is the linear wave equation, following the tests described by Bennet (2002).



## **PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE**

### **Teses e Dissertações (TDI)**

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

### **Manuais Técnicos (MAN)**

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

### **Notas Técnico-Científicas (NTC)**

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

### **Relatórios de Pesquisa (RPQ)**

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

### **Propostas e Relatórios de Projetos (PRP)**

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

### **Publicações Didáticas (PUD)**

Incluem apostilas, notas de aula e manuais didáticos.

### **Publicações Seriadas**

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

### **Programas de Computador (PDC)**

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.

### **Pré-publicações (PRE)**

Todos os artigos publicados em periódicos, anais e como capítulos de livros.