MARCOS VINICIUS ANDRIOLO

Análise de Métodos não Lineares para Previsão de Vazões Médias Mensais

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre, no Programa de Pós-Graduação em Engenharia de Recursos Hídricos e Ambiental, do Setor de Tecnologia da Universidade Federal do Paraná.

Orientador: Prof. Eloy Kaviski

CURITIBA

Abril - 2006

AGRADECIMENTOS

Agradeço ao professor Eloy Kaviski pela orientação, acompanhamento e revisão deste trabalho, assim como pela sua especial contribuição a minha vida pessoal e profissional.

Meu agradecimento ao LACTEC/CEHPAR, especialmente a UTHG pelo incentivo e apoio.

A COPEL, que me apresentou à hidrologia.

Meu eterno agradecimento aos meus pais, José e Wânia, que contribuíram e me apoiaram em todos os momentos da minha vida.

Aos meus irmãos, Mirian e Anderson que estiveram presentes durante a realização deste trabalho.

A minha namorada Andréa, por estar sempre ao meu lado.

Aos meus amigos, por todos os momentos.

Ao Pe. Luiz Andriolo pela tradução do abstract.

E a todos, que direta ou indiretamente, contribuíram para a realização e divulgação deste trabalho.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

Para um planejamento eficaz da operação hidráulica de reservatórios de sistemas hidroelétricos, necessitam-se de técnicas que permitam realizar previsões de vazões de forma a minimizar as variações entre a previsão e os valores que realmente ocorreram no futuro.

Os objetivos propostos nesta dissertação são: implementar e comparar os modelos MARS (*Multivariate Adaptive Regression splines*), os métodos de interpolação Ótima e por Multiquadrícas para prever vazões afluentes médias mensais para o horizonte de 1 a 12 meses, e analisar a influência do período de calibração na qualidade das previsões de afluências médias mensais para o horizonte de 1 a 12 meses.

Neste trabalho foi realizado um estudo de caso usando-se as vazões médias mensais de 14 reservatórios do sistema hidroelétrico brasileiro, e compararam-se os resultados obtidos pelos métodos de previsão propostos com os resultados obtidos pelo modelo Periódico Auto-Regressivo PAR de ordem 6, que foi instituído como o resultado mínimo esperado, dado que o modelo PAR[6] é o modelo atualmente utilizado para prever vazões médias mensais no sistema elétrico brasileiro.

Os resultados obtidos pelos modelos implementados mostraram-se mais robustos quando comparados com o modelo PAR[6], sendo que o modelo MARS 2d (que utiliza apenas a vazão do mês anterior) com quatro trechos apresentou o melhor resultado.

ABSTRACT

To have an efficient hydraulic operation planning of reservoirs for hydroelectric systems, it needs techniques that give a possibility to realize inflow forecasting. Those techniques can help to minimize variation in between inflow forecast value and the real inflow that can happen in the future.

Aims put foreword in this dissertation are an attempt to implement and compare models: MARS (Multivariate Adaptive Regression Splines), Otimal interpolation and Multiquadric interpolation. These models can help to forecasting average monthly inflow for a period of 1 to 12 months. Those models are helpful to make analyses of influence on calibration period of monthlies average in quality inflow forecast for a period of 1 to 12 months.

In this dissertation has being studied the case of 14 reservoirs of the Brazilian hydroelectric system. This study was about monthly inflow average on these 14 reservoirs. Results obtained from those three models above have been compared with results reached by model PAR[6] (Auto-Regressive Periodical order 6^{th}). Results from this comparison has instituted as minimum expected results. It must be considered because PAR[6] model is used to forecasting monthly inflow average in Brazilian electrical system. Those models which have been implemented showed better results compared with PAR[6] model. Nevertheless, MARS 2d has showed the best results. This model is used to measure inflow average of a month before.

1 INTRODUÇÃO

1.1 CONDIÇÕES GERAIS

O escoamento superficial é um processo complexo influenciado por muitos fatores como: topografia, cobertura vegetal, tipo de solo, características do canal, escoamento subterrâneo, distribuição da precipitação entre outros fatores. O projeto de sistemas de recursos hídricos e a análise do impacto ambiental que estes sistemas podem causar requerem muitas vezes a estimativa de vazões e suas propriedades estatísticas (Karunanithi et al, 1994), sendo a modelagem de séries temporais um passo importante para planejar e operar os sistemas de recursos hídricos (Nayak et. al, 2004).

Tradicionalmente, modelos autoregressivos ou de médias móveis do tipo ARMA (p,q) são utilizados para modelar as séries temporais de vazões. Entretanto, esses modelos lineares podem não representar a dinâmica não linear inerente aos processos hidrológicos (Tokar e Johnson, 1999).

O problema básico da modelagem matemática de sistemas físicos é encontrar funções que consigam reproduzir com certa fidelidade a realidade. Uma forma comum da modelagem consiste em obter uma função estimadora que exprime a variável dependente y em relação a diversas variáveis explicativas $x_1, \cdots x_n$, e um erro e também chamado de componente estocástico, geralmente com valor esperado igual a zero ($E(e) = 0$).

$$y = f(x_1, \cdots, x_n) + e \tag{1.1}$$

Um dos propósitos desta função é obter a partir de uma série de valores para $x_1, \cdots, x_n$ prever os valores correspondentes de y, possibilitando planejar os impactos que estes valores possam causar.

Os três problemas clássicos de estimação estocástica são previsão, filtragem, e preenchimento de séries temporais. Isto corresponde a estimar o futuro, presente, e passado, respectivamente, baseando-se na avaliação de informações históricas disponíveis. (Eyink, 2001).

Uma das dificuldades encontradas para solucionar os problemas de previsão, filtragem e preenchimento de séries, consiste em determinar uma função “robusta” que represente, com boa aproximação, o comportamento da série histórica disponível. A escolha de um determinado modelo para representação da série hidrológica torna-se difícil, pois esta série representa apenas a realização de um processo que possui componentes estocásticos, caracterizando uma aleatoriedade das afluências.

Para um planejamento eficaz da operação hidráulica dos reservatórios do sistema hidrelétrico brasileiro, técnicas que permitam realizar previsões acuradas de vazões futuras podem contribuir significativamente para uma operação mais eficiente do sistema.

Um dos aspectos que interessa ao produtor de hidroeletricidade é a vazão futura, pois, para propósitos estratégicos, essas previsões tornam possível a programação eficiente da geração hidroelétrica e também estabelecer a programação das termelétricas (Mine, 1998).

Com o avanço tecnológico, está se tornando possível adotar modelos de previsão não lineares que permitem obter uma menor discrepância entre os valores previstos e observados. Porém, os modelos não lineares de previsão de vazões necessitam de estudos e desenvolvimento de métodos, baseados em um modelo estatístico fundamentado na dinâmica de sistemas não lineares.

1.2 OBJETIVOS

Um dos objetivos desta dissertação é utilizar modelos não lineares para prever vazões afluentes médias mensais para o horizonte de 1 a 12 meses, aplicado à reservatórios do sistema hidroelétrico, sendo analisado neste trabalho 14 locais de usinas do sistema hidroelétrico brasileiro, localizados nos rios Tocantins, São Francisco, Paraíba, Grande, Tietê, Paraná, Paranapanema, Iguaçu, Uruguai e Jacuí.

Os modelos utilizados nesta dissertação são o modelo Periódico Auto-regressivo PAR de ordem 6, o modelo de interpolação multiquadrático, o modelo de interpolação ótima e o modelo MARS.

O modelo PAR de ordem 6 e o modelo MARS são modelos de regressão e os modelos de interpolação multiquadrático e interpolação ótima são modelos de interpolação adaptados para prever vazões afluentes médias mensais.

O modelo *PAR(1)* pode ser visto como uma forma sintética de se representar um conjunto de 12 regressões, uma para cada mês do ano. Assumindo-se como variável explicativa para cada mês *i* a vazão do mês *i-1*, pode-se generalizar o conceito assumindo como variáveis explicativas as vazões dos *p* meses imediatamente antecedentes.

O modelo PAR foi utilizado porque é o modelo adotado no setor elétrico brasileiro para prever afluências, sendo que é esperado que os resultados dos modelos propostos sejam melhores que os resultados do modelo PAR. O valor previsto obtido através do modelo PAR é uma combinação linear das 6 vazões observadas anteriormente acrescido de um ruído aleatório.

Uma das técnicas usadas para ajustar regressões não lineares é o modelo estatístico MARS (*Multivariate Adaptive Regression Splines*), desenvolvido na Universidade de Stanford, USA, por Friedman (1990), que tem apresentado resultados melhores que outros métodos lineares e não lineares. Embora não se encontre literatura disponível sobre previsões de vazões utilizando o modelo MARS, existem estudos em outras áreas do conhecimento onde o método foi empregado (Chou et. al, 2004; Sephton, 2001; Xu et. al., 2004; Osei-Bryson e Ko, 2003), encontrando-se previsões mais próximas da realidade.

Os modelos de interpolação multiquadrático e interpolação ótima tratam a previsão de vazões por meio de uma superfície interpolável, sendo que a função de interpolação será adaptada de forma a obter uma superfície em função do tempo, ao invés de uma superfície espacial como no modelo tradicional.

O modelo multiquadrático é baseado nas funções multiquadráticas que tradicionalmente representam a distância espacial (latitude, longitude e altitude) sobre a variância desta distância (Kaviski, 1992).

O modelo ótimo utiliza a interpolação ótima que segundo Morin et. al. (1979) é aquela que determina os pesos pela minimização da variância dos erros de interpolação.

Um segundo objetivo desta dissertação é analisar a influência do período de calibração na qualidade das previsões de afluências médias mensais para o horizonte de 1 a 12 meses. O tamanho do período de calibração compreende espaços de 20 até 60 anos, entre os anos de 1931 até 1991.

1.3 RESUMO DO ESTUDO

No capítulo 2 serão apresentados os modelos de previsão, dividido em três partes. A primeira parte aborda os modelos lineares, na segunda parte são tratados os modelos não lineares, e na terceira parte são abordados os métodos de interpolação.

No capítulo 3 apresentam-se os modelos aplicados, sendo dividido em quatro partes. A primeira parte trata da aplicação do modelo Periódico Auto-Regressivo de ordem 6 (PAR[6]) , a segunda parte trata da aplicação do modelo de interpolação multiquadrático, a terceira parte trata do modelo de interpolação ótima, e a quarta parte trata do modelo MARS.

No capítulo 4 apresentam-se os resultados obtidos e a análise dos modelos, sendo dividido em duas partes. A primeira parte trata das usinas selecionadas, e a segunda parte dos resultados obtidos.

No capítulo 5 são apresentadas as conclusões e recomendações.

2 MODELOS DE PREVISÃO

2.1 GERAL

Os modelos utilizados para descrever séries temporais são algoritmos que geram séries temporais a partir da definição de um processo estocástico, isto é, processos controlados por leis probabilísticas (Morettin e Toloi, 1981).

Um modelo estocástico descreve a estrutura probabilística de uma seqüência de observações. Uma série temporal de *N* sucessivas observações $z = (z_1, z_2, \ldots, z_n)$ é uma realização amostral de uma população infinita de tais amostras, que foram geradas pelo processo. (Mine, 1984).

Os modelos estocásticos podem ser subdivididos em modelos estacionários e modelos periódicos. No primeiro grupo pode-se citar o modelo ARMA(p,q). Já entre os modelos periódicos citam-se o modelo PAR(p) (Mazer, 2003). Os modelos também são subdivididos em modelos lineares e não lineares.

Muitas vezes durante o processo de modelagem das séries hidrológicas temporais, faz-se necessário transformar séries ciclicamente não-estacionárias em estacionárias. Valendo-se do critério da parcimônia, ou seja, a utilização de um número mínimo de parâmetros, um procedimento simples consiste em realizar uma análise harmônica nos parâmetros periódicos (média, variância, covariância) e representá-los por uma função matemática, obtendo-se um modelo para previsão de afluências (SCEN/GTMC, 1980).

Um grande número de modelos matemáticos tem sido propostos para se fazer uma aproximação estocástica linear de um processo hidrológico, que na verdade é um processo complexo, não linear, e apresenta uma grande variabilidade espacial e temporal. Estudos recentes indicam que um sistema determinístico com interdependência não linear pode resultar em uma estrutura complexa, denominado caos determinístico. Evidências sobre a não-linearidade dos processos hidrológicos são apresentadas por Kavvas (2003). A não linearidade dos processos hidrológicos resulta da heterogeneidade e

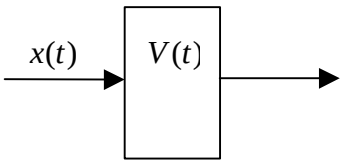
anisotropia das características das bacias hidrográficas e das equações não lineares da mecânica dos fluidos.

Este capítulo é subdividido em três partes, a primeira parte trata dos modelos lineares, a segunda parte trata dos modelos não lineares, e a terceira parte trata dos métodos de interpolação.

2.2 MODELO LINEAR

Muitas áreas do conhecimento, como engenharia e física envolvem o conceito de sistema dinâmico, caracterizado por uma série de entrada $x(t)$, uma série de saída $z(t)$ e uma função de transferência $V(t)$. Este sistema dinâmico é ilustrado pela *FIGURA 2.1*.

FIGURA 2.1 Sistema dinâmico.



Um sistema muito empregado em previsões e de particular importância são os sistemas lineares, onde a saída é relacionada com os dados de entrada através de um funcional linear envolvendo $V(t)$, sendo um exemplo típico:

$$z(t) = \sum_{t=0}^{\infty} V(t)x(t-t) \tag{2.1}$$

A condição necessária para um sistema possuir um comportamento linear (Barth et. al., 1987) está na validade do princípio da superposição. Considerando uma entrada $x(t)$, que produz uma saída $z(t)$:

$$x(t) \rightarrow z(t) \tag{2.2}$$

Se o princípio da superposição for válido a entrada $x_1(t) + x_2(t)$ produz uma saída $z_1(t) + z_2(t)$.

Outra característica de um sistema linear é a propriedade da homogeneidade, sendo que se existem *n* entradas no sistema, de forma que:

$$y_1 = y_2 = \ldots = y_n \tag{2.3}$$

O sistema (2.3) é linear quando ny_1 produz a saída nx_1 (Tucci,1998).

Quando as propriedades de superposição e homogeneidade são satisfeitas o sistema é linear. Matematicamente um sistema linear pode ser identificado, quanto a sua linearidade, pela equação diferencial que representa este sistema. Um exemplo de sistema linear é:

$$A_n \frac{d^n x}{dt^n} + A_{n-1} \frac{d^{n-1} x}{dt^{n-1}} + \ldots + A_1 \frac{dx}{dt} + A_0 x = z(t) \tag{2.4}$$

Onde *x(t)* é a função de saída, *y(t)* é a função de entrada, A_i, para *i=1,2,...,n* são os coeficientes da equação diferencial. Quando os coeficientes $A_i \neq f[x(t)]$ a equação diferencial é linear. Existindo um coeficiente A_i que dependa da função da variável dependente *x(t)* a equação torna-se não-linear.

2.2.1 Modelos Auto-Regressivos

Na literatura existem dois diferentes tipos de previsão (Ing e Wei, 2003). A previsão independente da realização, onde os valores futuros formam séries independentes com a mesma estrutura de probabilidade da série observada, sendo a análise matemática relativamente simples, e o segundo tipo de previsão é a com a mesma realização, isto é, prevê-se $x_{n+h}, h \geq 1$, com base nos valores anteriores de x_n.

Um modelo auto-regressivo de ordem infinita pode ser expresso como:

$$Z_t = \sum_{j=1}^{\infty} p_j Z_{t-j} + a_t \tag{2.5}$$

Onde Z_t é o valor observado no instante *t*, $p(B)$ é um operador que têm a forma polinomial em *B* sendo |*B*| *1*, a_t é o ruído branco.

A equação *(2.5)* representa um modelo auto-regressivo de ordem infinita que é pouco utilizado na prática, sendo substituído por um modelo de ordem finita:

$$(1 - f_1 B - \ldots - f_p B^p) Z_t = a_t$$

$$f(B) Z_t = a_t \qquad (2.6)$$

Um exemplo de um modelo de ordem um é:

$$(1-f_1B)Z_t = a_t \tag{2.7}$$

Devido ao polinômio $f(B)$ ser finito a equação (2.7) é incondicionalmente invertível, sendo esta propriedade assegurada a todos os modelos *AR* de ordem finita *p*. O modelo AR é equivalente a uma média móvel infinita estacionária, sendo que $|f_i| < 1$ para garantir sua estacionaridade.

A função de auto-correlação do modelo AR(p) pode ser obtida multiplicando-se a equação (2.8) por Z_{t-k}, obtendo-se o valor esperado expresso pela equação (2.9):

$$Z_t = f_1 Z_{t-1} + f_2 Z_{t-2} + \ldots + f_p Z_{t-p} + a_t \tag{2.8}$$

$$E[Z_{t-k} Z_t] = E\left[f_1 Z_{t-k} Z_{t-1} + f_2 Z_{t-k} Z_{t-2} + \ldots + f_p Z_{t-k} Z_{t-p} + Z_{t-k} a_t\right] \tag{2.9}$$

Tomando-se o valor esperado (2.9) do produto cruzado obtém-se:

$$g_k = f_1 g_{k-1} + f_2 g_{k-2} + \ldots + f_p g_{k-p} \tag{2.10}$$

Dividindo-se a equação (2.10) por g_0, ou seja, pela variância do processo, obtém-se a função de auto-correlação do modelo AR(p):

$$r_k = f_1 r_{k-1} + f_2 r_{k-2} + \ldots + f_p r_{k-p} \tag{2.11}$$

Sendo r a função de correlação.

Substituindo *k=1,2,...,p* na equação (2.11) obtém-se um conjunto de equações lineares conhecido como equações de Yule-Walker.

Dividindo o processo estacionário pela variância encontramos o sistema de equações:

$$\Phi = \Sigma^{-1} R \tag{2.12}$$

$$\Phi = \begin{bmatrix} f_1 \\ \vdots \\ f_p \end{bmatrix} \qquad R = \begin{bmatrix} r_1 \\ \vdots \\ r_p \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & 1 & r_1 & \cdots & r_{p-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & 1 \end{bmatrix} \tag{2.13}$$

Sendo que a função de correlação r_k deve satisfazer: $f(B)r_k = 0$, formando o operador polinomial:

$$f(B) = \prod_{i=1}^{p} (1 - G_i B) \tag{2.14}$$

Obtendo-se assim a solução geral:

$$r_k = A_1 G_1^k + A_2 G_2^k + \ldots + A_p G_p^k \tag{2.15}$$

Sendo $G_1^{-1}, \ldots, G_p^{-1}$ a raiz polinomial (equação característica $f(B)$).

O modelo auto-regressivo linear de ordem 1 é o modelo popularmente mais empregado para simulação e previsão de séries temporais hidrológicas (Bras e Rodrígues-Iturbe, 1985). Em hidrologia o modelo AR[1] é comumente chamado de modelo Thomas-Fiering (Mass et al., 1962).

Existe uma série de técnicas para identificar um sistema com séries contínuas, usualmente classificadas em aproximação direta e aproximação indireta (Larsson e Söderstrom, 2002).

Para definir o problema de identificação do processo *AR*, é necessário observar a série temporal no espaço descontínuo, conhecendo-se o momento da observação e o valor observado.

Com os dados é possível estimar os parâmetros auto-regressivos, assumindo que a distância entre duas amostras é um processo estocástico independente e identicamente distribuído, e também assumir que o ruído do sistema é independente.

2.2.2 Modelos ARMA

Na prática quando se necessita de um modelo com um número não muito grande de parâmetros, a inclusão de termos auto-regressivos e de médias móveis é a solução adequada (Morettin e Toloi, 1981), surgindo desta forma o modelo Auto-regressivo de médias móveis , o ARMA(p,q):

$$Z_t = f_1 Z_{t-1} + \ldots + f_p Z_{t-p} + a_t - q_1 a_{t-1} - \ldots - q_q a_{t-q} \tag{2.16}$$

Sendo $f(B)$ e $q(B)$ operadores auto-regressivos e de médias móveis, respectivamente.

O operador auto-regressivo estacionário de ordem *p:*

$$f(B) = 1 - f_1 B - f_2 B^2 - \ldots - f_p B^p \tag{2.17}$$

Podendo-se escrever:

$$f(B)\tilde{Z}_t = a_t \tag{2.18}$$

$$f(B)\tilde{Z}_t = q(B)a_t \tag{2.19}$$

Um modelo frequentemente utilizado é o ARMA(1,1), onde *p*=*q*=1, $f(B) = 1 - fB$ e $q(B) = 1 - qB$, ou seja (2.19) reduz-se a:

$$Z_t = fZ_{t-1} + a_t - qa_{t-1} \tag{2.20}$$

Os cálculos algébricos para determinação da função de auto-correlação do modelo geral ARMA(*p,q*) são bastantes complexos, em conseqüência, apresentar-se-á somente a função de auto-correlação do modelo ARMA(1,1), que tem sido a de maior interesse em hidrologia (O'Connell, 1974).

$$r_1 = \frac{(1 - f_1 q_1)(f_1 - q_1)}{(1 + f_1^{\,2} - 2 f_1 q_1)}$$

$$r_k = f_1 \, r_{k-1} \tag{2.21}$$

É possível a partir das equações acima observar que a presença do termo de médias móveis influencia somente o cálculo do coeficiente de *lag* k=1, e que a partir daí, somente os termos auto-regressivos contribuem para a função. Isso era esperado, uma vez que o modelo de MA(1) (médias móveis) tem uma função de auto-correlação que desaparece a partir do *lag* k=1 (Mine, 1984).

A função auto-correlação (*fac)* das funções *AR(p)*, *MA(q), ARMA(p,q)* apresentam características especiais.

Para o modelo *AR(p)* a *fac* decai de acordo com exponenciais e/ ou senóides amortecidas, infinita em extensão. Já o modelo *MA(q)* tem *fac* finita, no sentido de que ela apresenta um corte após o "*lag*" *q*. O modelo *ARMA(p,q)* tem *fac* infinita em extensão e que decai de acordo com exponenciais e/ou senóides amortecidas após o *"lag*" *q,p*.

2.2.3 Modelo PAR

Quando não se pode adotar o modelo estacionário, como no caso de modelagem de vazões mensais, é necessário efetuar uma adaptação no modelo estacionário, através do modelo de Thomas e Fiering ou *PAR(1)*, ou *AR(1)* periódico:

$$M_{t+1,t} = m_{t+1} + \frac{r_t s_{t+1}}{s_t}(m_{t,t} - m_t) + s_{t+1} + \sqrt{1 - r_t^2}\, Z_{t+1,t} \qquad (2.22)$$

Onde $M_{t,t}$ é a vazão do mês t , ano t ; $t = 1,\ldots,12$ é o índice do mês; $t = 1,2,\ldots$ é o índice do ano; $Z_{t,t}$ é o componente independente padrão; $r_t = Corr[M_{t+1,t}, M_{t,t}]$; $m_t = E[M_{t,t}]$; $s_t^{\,2} = VAR[M_{t,t}]$

O modelo Periódico Auto-Regressivo *PAR(1)* pode ser visto como uma forma sintética de se representar um conjunto de 12 regressões, uma para cada mês do ano. Assumindo-se como variável previsora para cada mês t a vazão do mês $t-1$, pode-se generalizar o conceito assumindo como variáveis

previsoras as vazões dos *p* meses imediatamente antecedentes, resultando no modelo *PAR(p)* expresso por:

$$M_{t,t} = af_t + \sum_{i=1}^{P_t} a_{it} M_{t-i,t} Z_{t,t} \quad (2.23)$$

2.2.4 Modelo ARIMA

O modelo auto-regressivo-integrado-médias-móveis *ARIMA* é uma extensão relativamente simples da teoria do modelo *ARMA*. O modelo permite manusear um número limitado de modelos não-estacionários, essencialmente processos de média não estacionária (Bras e Rodrígues-Iturbe, 1984).

A maioria das séries temporais em hidrologia apresenta um comportamento não estacionário que pode ser representado por uma descendência ou ascendência. Uma maneira de contornar o problema é proceder diferenciações na série observada, dando origem aos modelos ARIMA (p, d, q). A série diferenciada será estacionária desde que a série observada exiba um comportamento homogêneo, ou seja, a série não tem afinidade com nenhum valor médio em particular, mas seu comportamento é semelhante em diversos períodos de tempo. (Mine, 1984).

O modelo ARIMA pode ser representado por um operador auto-regressivo generalizado $f(B)$, tal-que:

$$y(B)Z_t = f(B)(1-B)^d Z_t = q(B)a_t \quad (2.24)$$

Ou:

$$f(B)\nabla^d z_t = q(B)a_t \quad (2.25)$$

Onde: ∇^d é um operador tal que:

$$\nabla^d z_t = (1-B)^d z_t \quad (2.26)$$

Sendo: *d* o grau de diferenciação utilizado para tornar a série estacionária na média, desde que $d \geq 1$.

De ordem *(p,d,q)* onde *p* e *q* são as ordens de $f(B)$ e $q(B)$ respectivamente.

Um modelo *ARIMA (p,d,q)* dado pela equação (2.25) pode ser representado de 3 formas:

a) Em termos de valores prévios de Z_t e do valor atual e prévio de a_t.

b) Em termos de valor atual e prévio de a_t.

c) Em termos de valores prévios de Z_t e do valor atual de a_t.

A construção de modelos *ARIMA* tem como fase crítica a identificação do particular modelo *ARIMA* a ser ajustado. Esta escolha é feita com base nas auto-correlações e auto-correlações parciais estimadas, que esperam representar adequadamente as respectivas quantidades teóricas que são desconhecidas.

O objetivo da identificação é determinar os valores *(p,d,q)* do modelo *ARIMA (p,d,q)*, além de estimativas preliminares dos parâmetros a serem usados no estágio de estimação.

O procedimento de identificação consiste em duas partes. Na primeira parte diferencia-se a série Z_t tantas vezes quanto necessário, para se obter uma série estacionária, de forma que o processo $\Delta^d Z_t$ seja reduzido a um ARMA(p,q). O número de diferenças *d* necessárias para que o processo se torne estacionário é alcançado quando a fac (função de auto-correlação) amostral $W_t = \Delta^d Z_t$ decresce rapidamente para zero. Na segunda parte identifica-se o processo ARMA(p,q) resultante, através da análise das auto-correlações e auto-correlações parciais estimadas, cujos comportamento devem imitar os comportamentos das respectivas quantidades teóricas.

As estimativas preliminares são obtidas através das auto-correlações amostrais da série $W_t = \Delta^d Z_t$ baseado em um procedimento interativo de estimativa de mínimos quadrados não linear. Os valores iniciais podem ser

obtidos através de um procedimento condicional, em que os valores iniciais desconhecidos são substituídos por valores supostos razoáveis. E outro procedimento é incondicional, em que os valores iniciais são estimados por uma amostra de dados.

Para conhecer a precisão dos estimadores encontrados devem-se construir intervalos de confiança, obtendo estimativas das variâncias dos estimadores da covariância entre estimadores. Através das estimativas das variâncias se obtêm intervalos de confiança para os parâmetros $n_i, i = 1, \ldots, k$:

$n_i = (f, q)$, de ordem *k x 1*, onde *k*=*p*+*q*.

Após encontrar os estimadores pode-se verificá-los através do método do super-ajustamento, ou seja, são estimados parâmetros extras para o modelo e examinando se estes parâmetros são significativos e se sua inclusão diminui de forma significativa a variância residual (ruído brando).

Outro teste é o de auto-correlação residual, que verifica se os valores estimados e observados são aproximadamente não correlacionados, sendo $\hat{r}_k$ um indício das auto-correlações dos resíduos $\hat{a}_t$ deve-se ter $\hat{r}_k \approx 0$, sendo:

$$\hat{r}_k \sim N(0, \frac{1}{n}) \tag{2.27}$$

Sendo a comparação de $\hat{r}_k$ com os limites $\pm \frac{2}{\sqrt{n}}$ fornece uma indicação geral da possível quebra de comportamento do ruído branco em a_t.

As formas básicas de previsão utilizando o modelo *ARIMA* são:

a) Forma de equação de diferenças:

$$Z_{t+h} = x_1 Z_{t+h-1} + \ldots + x_{p+d} Z_{t+h-p-d} - q_1 a_{t+h-1} - q_2 a_{t+h-2} - \ldots - q_q a_{t+h-q} + a_{t+h} \tag{2.28}$$

b) Forma de choques aleatórios:

$$Z_{t+h} = \sum_{j=-\infty}^{t+h} y_{\ t+h-j} a_j = \sum_{j=0}^{\infty} y_{\ j} a_{t+h-j} \tag{2.29}$$

c) Forma invertida:

$$Z_{t+h} = \sum_{j=1}^{\infty} \prod_j Z_{t+h-j} + a_{t+h} \tag{2.30}$$

2.3 MODELOS NÃO LINEARES

A idéia básica do modelo não linear é utilizar equações simples que se aproximem de $f(x)$, sendo normalmente uma equação polinomial de baixa ordem, definida com diferentes sub-regiões de domínio.

Um tipo muito popular de função polinomial são as splines, que são polinômios de ordem q, que produzem derivadas de ordem $q-1$, onde o tipo mais encontrado são as splines cúbicas.

Os modelos não lineares podem ser expressos por:

$$\hat{f}(x) = \sum_{j=1}^{j} \hat{g}_j(z_j) \tag{2.31}$$

Sendo z_j formado por diferentes sub-vetores $x_1, \cdots x_n$, e *g* uma função paramétrica simples.

Utilizando funções polinomiais com splines, e substituindo-as na variável z_j é possível obter o mínimo quadrado global como resposta de y, podendo formular o estimador $\hat{f}(x)$:

$$\hat{f}(x) = \arg\min_{\{g_j\}} \left\{ \sum_{i=1}^{n} \left[y_i - \sum_{j=1}^{j} g_j(z_{ij}) \right]^2 + \sum_{j=1}^{j} l_{\ j} R(gj) \right\} \tag{2.32}$$

Onde: *R(g)* é um funcional que é incrementando com o aumento da função *g(x)*, $l_{\ j}$ é um parâmetro que regula a relação entre o valor aproximado de da função *g* e o valor exato.

As funções não lineares são baseadas em expansões de baixa ordem com um passo *forward* (para frente), conseguindo na prática considerável sucesso. Os modelos não lineares não adaptáveis normalmente apresentam uma suave limitação na expansão de suas funções de baixa ordem.

Uma estratégia para trabalhar com funções de ordem elevada é introduzir uma adaptação computacional que promova o ajuste dinâmico com o intuito de resolver o problema em particular, criando desta forma um algoritmo adaptável.

Com este pensamento pode-se construir um estimador utilizando uma combinação linear de variáveis.

$$\hat{f}(x) = \sum_{m=1}^{m} f_m \left(\sum_{i=1}^{n} a_{im} x_i \right) \tag{2.33}$$

Onde: f_m é uma função univariada, a_{im} é um coeficiente e x_i é o valor observado.

Esta combinação reduz o tempo computacional, apresentando a desvantagem de dificultar a separação do efeito associado com as variáveis dependentes.

2.3.1 Regressão Parcial Recursiva

A regressão parcial recursiva dada pelo estimador $\hat{f}(x)$ quando $x \in R_m$:

$$\hat{f}(x) = g_m(x \mid \{a_j\}_1^p) \tag{2.34}$$

Sendo que $\{R_m\}_1^n$ representa as sub-regiões parciais pertencentes ao domínio D e a função g_m é uma generalização linear. Uma função constante muito comum é:

$$g_m(x \mid a_m) = a_m \tag{2.35}$$

O passo inicial para resolver esta regressão (2.34) é usar bons dados na estimativa simultânea nas sub-regiões, associando os parâmetros de forma separada para cada sub-região.

A regressão parcial recursiva utiliza constantes de aproximação simples, conferindo-lhe a habilidade de explorar funções de baixa ordem, e também ser utilizado em locais marginais. Além de ser adaptado a métodos com funções multivariadas.

2.3.2 Interpolação com Splines Cúbicas

A função spline permite a interpolação de curvas suaves com baixa probabilidade de exibir as grandes oscilações características das funções polinomiais de ordem elevada (Stoer e Bulirsch, 2002).

As funções splines são conectadas com a divisão:

$$\Delta : a = x_0 < x_1 < \ldots < x_n = b \tag{2.36}$$

no intervalo $[a,b]$ com os nós $x_i, i = 0,1,\ldots,n$, sendo sub-divisões da função polinomial $S : [a,b] \to \Re$.

A spline cúbica (Burdem e Faires, 2003) é um polinômio cúbico genérico que possui quatro constantes, tendo flexibilidade suficiente para assegurar que o interpolador não seja somente continuamente derivável no intervalo, mas também que tenha sua derivada segunda contínua, produzindo uma curvatura contínua (Boor, 1978).

$$p_{i-1}^{''}(t_i) = p_i^{''}(t_i) \tag{2.37}$$

Porém, o spline cúbico não garante que as derivadas do interpolador concordem com as da função que está sendo aproximada, mesmo em seus nós.

2.3.2.1 Fundamentação teórica

A função spline cúbica S_Δ é uma função real com as seguintes propriedades:

a) $S_\Delta \in C^2[a,b]$, estão S_Δ possui derivada segunda no intervalo $[a,b]$.

b) S_Δ coincide em todo o subintervalo $[x_i, x_{i+1}], i = 0,1,\ldots,n-1$, com o polinômio de grau três.

Sendo $S(x)$ um polinômio cúbico, indicado por $S_j(x)$, no subintervalo $[x_j, x_{j+1}]$ para cada *j=0,1,...,n-1*.

A função spline necessita de dois graus de liberdade, desta maneira é necessário considerar mais três requisitos:

a) $S_\Delta^{''}(Y;a) = S_\Delta^{''}(Y;b) = 0$ (2.38)

b) $S_\Delta^{(k)}(Y;a) = S_\Delta^{(k)}(Y;b)$ para $k = 0,1,2 : S_\Delta(Y;.)$ é periódica (2.39)

c) $S_\Delta^{'}(Y;a) = y_0^{'}, S_\Delta^{'}(Y;b) = y_n^{'}$ para os valores atribuídos a $y_0^{'}, y_n^{'}$ (2.40)

E, se $f \in K^3[a,b]$ então se pode definir:

$$\|f\|^2 = \int_a^b \left| f^{''}(x) \right|^2 dx \tag{2.41}$$

Se $f \in K^2(a,b), \Delta = \{a = x_0 < x_1 < \ldots < x_n = b\}$ é uma divisão do intervalo $[a,b]$, e se S_Δ é uma função spline com nós $x_i \in \Delta$, então:

$$\|f - S_\Delta\|^2 = \|f\|^2 - \|S_\Delta\|^2 - 2\begin{bmatrix} \left(f^{'}(x) - S_\Delta^{'}(x)\right) S_\Delta^{''}(x)\Big|_a^b - \\ \sum_{i=1}^n \left(f(x) - S_\Delta(x)\right) S_\Delta^{'''}\Big|_{x_{i-1}^+}^{x_i^-} \end{bmatrix} \tag{2.42}$$

Já para a função spline restrita temos que o espaçamento $\Delta = \{a = x_0 < x_1 < \ldots < x_n = b\}$ no intervalo $[a,b]$, possui valores $Y = (y_0, \ldots, y_n)$ e

a função $f \in K^2(a,b)$ com $f(x_i) = y_i$ para $i = 0,1,2,\cdots,n$ com $\|f\| \geq \|S_\Delta(Y,.)\|$ e mais precisamente:

$$\|f - S_\Delta(Y;.)\|^2 = \|f\|^2 - \|S_\Delta(Y;.)\|^2 \geq 0 \tag{2.43}$$

Sendo que a função spline restrita deve satisfazer os seguintes requisitos:

a) $S''_\Delta(Y;a) = S''_\Delta(Y;b) = 0$ (2.44)

b) $f \in K_p^2[a,b], S_\Delta(Y;.)$, periódico (2.45)

c) $S'_\Delta(Y;a) = f'(a), S'_\Delta(Y;b) = f'(b)$ (2.46)

2.3.2.2 Determinando a interpolação com funções spline cúbicas

Representando a função spline em relação aos seus momentos (derivadas segundas):

$$S_\Delta(Y;x) = a_j + b_j(x - x_j) + g_j(x - x_j)^2 + d_j(x - x_j)^3 \tag{2.47}$$

Para todo $x \in \lfloor x_j, x_{j+1} \rfloor$, sendo:

$$a_j = y_i,\ b_j = S'_\Delta(Y;x_j),\ d_j = \frac{S'''_\Delta(Y;x_j^+)}{6} \tag{2.48}$$

Para construir o spline cúbico para uma função *f* dada, as condições indicadas na definição (equações: 2.38, 2.39, 2.40, 2.44, 2.45, 2.46) são aplicadas aos polinômios cúbicos:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \tag{2.49}$$

Para *j=1, 2, ... , n-1*.

O sistema linear de equações resultantes é:

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1}) \quad (2.50)$$

Para *j=1, 2, ..., n-1*.

Sendo que:

$$h_j = x_{j+1} - x_j \quad (2.51)$$

$$a_n = f(x_n) \quad (2.52)$$

$$c_n = \frac{s''(x_n)}{2} \quad (2.53)$$

Se *f* é definida em $a = x_0 < x_1 < \ldots < x_n = b$, então *f* tem um único spline interpolador natural *S* nos nós $x_0, x_1, \ldots, x_n$, isto é, um spline interpolador que satisfaz as seguintes condições de contorno: $s''(a) = 0$, $s''(b) = 0$.

Aplicando estas condições de contorno na equação (2.50) obtém-se o sistema linear descrito por uma equação vetorial:

$$Ac=b \quad (2.54)$$

Sendo :

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \cdots & \cdots & 0 \\ 0 & h_1 & 2(h_1 + h_2) & h_2 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \cdots & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & \cdots & \cdots & 1 \end{bmatrix} \quad (2.55)$$

$$\mathbf{b}= \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix} \tag{2.56}$$

$$\mathbf{c}= \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} \tag{2.57}$$

O sistema descrito na equação (2.54) é um sistema linear tri-diagonal, com $n-2$ equações para n pontos, com diagonal horizontal dominante, resultando num sistemas de equações com solução exata (Boor, 1978).

Resolvendo este sistema linear de equações (2.54) é possível obter os coeficientes da spline cúbica.

As expressões (2.55), (2.56) e (2.57) representam um spline cúbico natural, dado quando a função possui condições de contorno livre, ou seja, $s''(a) = 0$, $s''(b) = 0$.

Quando temos condições de contorno restrito, $s'(a) = f'(a)$, $s'(b) = f'(b)$, a função spline é chamada de spline cúbico restrito, que de uma forma geral possui aproximações mais precisas, na medida em que elas incluem mais informações sobre a função. Porém, para utilizar este tipo de condição é necessário obter os valores das derivadas nos pontos extremos (*a, b*), ou uma aproximação destes valores.

O spline cúbico restrito, resulta no mesmo sistema de equações do spline cúbico natural (2.54) , porém, o sistema linear é construído de forma a incluir as condições de contorno, resultando no sistema abaixo:

$$\boldsymbol{A}= \begin{bmatrix} 2h_0 & h_0 & 0 & \cdots & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & \cdots & \cdots & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \cdots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & \cdots & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix} \quad (2.58)$$

$$\mathbf{b}= \begin{bmatrix} \frac{3}{h_0}(a_1-a_0)-3f'(a) \\ \frac{3}{h_1}(a_2-a_1)-\frac{3}{h_0}(a_1-a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n-a_{n-1})-\frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\ 3f'(b)-\frac{3}{h_{n-1}}(a_n-a_{n-1}) \end{bmatrix} \quad (2.59)$$

$$\mathbf{c}= \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (2.60)$$

A regressão utilizando funções spline pode ser expressa como uma combinação linear entre trechos de funções polinomiais básicas, que juntas alisam os extremos (nós), sendo os coeficientes das funções básicas estimando pelo método dos mínimos quadrados (Osei-Bryson e Ko, 2004).

2.3.3 Regressão Multivariada com Spline Adaptados (MARS)

O modelo MARS é um método estatístico relativamente recente proposto por Friedmam (1990) e agora desenvolvido pela Stanford Systems, tendo sua origem inspirada na regressão spline adaptada (Osei-Bryson e Ko, 2004).

Sendo o MARS um modelo de regressão flexível com uma alta dimensionalização dos dados. Permitindo a possibilidade de expandir suas

funções básicas associando automaticamente cada parâmetro com os dados utilizados. O modo de partição recursivo produz modelos contínuos com derivadas contínuas, fazendo com que o modelo MARS seja uma forma poderosa e flexível de relacionar a adição de interações com várias variáveis.

Um problema comum é adequar as funções de aproximação com muitas variáveis a um único valor de função em seus vários pontos no espaço, pois o problema aplicado matematicamente e de forma estatística acarreta em um erro associado à forma de aproximação adotada.

O MARS é um método flexível de regressão não linear envolvendo uma amostra moderada de dados com tamanhos variando de 50 N 1000 e um número de dimensões variando de 3 n 20, sendo descrito por uma função que envolve uma variável admensionalizada, seguido pelo somatório de todas as variáveis envolvendo uma dimensão (i), acrescido do somatório de todas as variáveis envolvendo duas dimensões (i, j), e assim sucessivamente até o número de dimensões que serão utilizadas no modelo.

O modelo MARS envolve números elevados de interações possibilitando a visualização de gráficos com vários pares de variáveis e valores complementares, construindo e usando pesados valores de respostas, construindo um círculo de funções de base, permitindo uma flexibilidade que reduz a parcialidade do modelo de estimativas devido à criação de parâmetros adicionais, ajustados para melhorar as aptidões dos dados processados.

O MARS é um método não linear que atende modelos com funções arbitrárias, porém, o modelo requer o conhecimento e estudo do sistema e dos seus dados específicos.

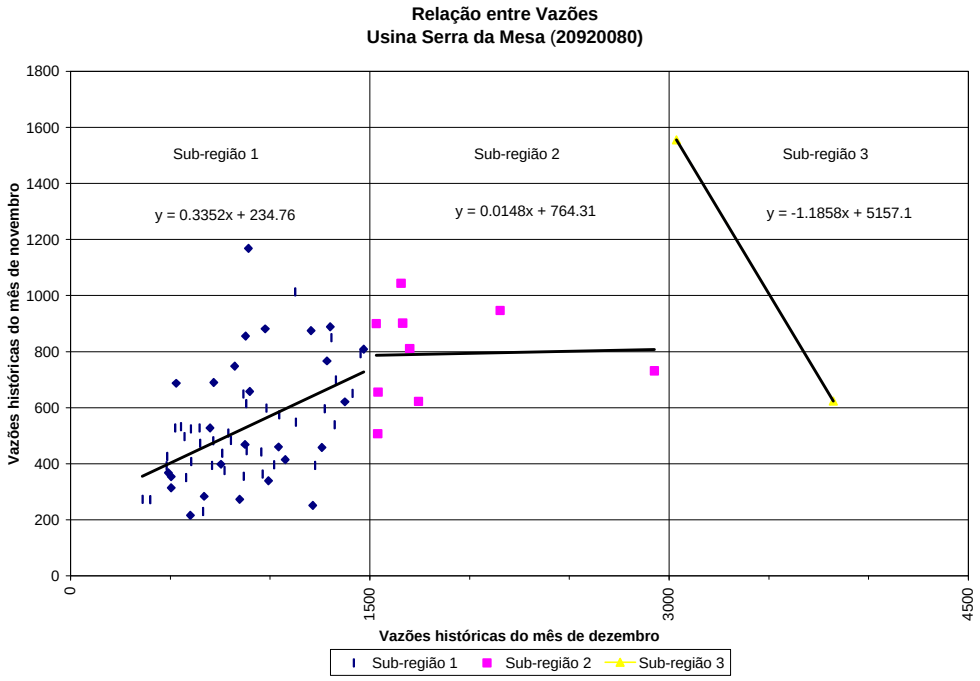
O método MARS combina uma divisão recursiva (divide o espaço amostral em trechos) das *splines* de uma forma a reter os aspectos positivos de suas propriedades, apresentando de forma adaptável à seleção de estratégias de subdivisão das variáveis locais, dependendo da associação de variáveis a funções complexas.

A vantagem do modelo MARS é que ele permite manusear funções complexas (não linear), suas relações e interações durante o processo de estipulação e interpretação do modelo (Briand et. al., 2004), tentando uma aproximação de séries de regressões com diferentes intervalos e com uma gama de variáveis independentes (sub-regiões do espaço variável e independente).

O modelo é baseado em vários princípios simplificados, muito flexíveis e adaptados de modo funcional, conveniente para a análise de dados.

Um dos principais aspectos do modelo MARS é que ele procura dividir o espaço amostral em sub-regiões com variáveis independentes, identificando interações de forma a evitar o excesso de aproximações dos dados, fazendo desta forma que em cada uma destas sub-regiões seja ajustada uma regressão linear independente (Briand et. al. , 2000).

FIGURA 2.2 Relação entre a série histórica de novembro e dezembro com regressão linear para sub-regiões.



A FIGURA 2.2 representa a relação entre a série histórica do mês de novembro com o de dezembro compreendida no período de 1931-2004 para a usina de Serra da Mesa (código ANEEL 20920080), ilustra como é elaborada a divisão de um espaço amostral bidimensional (eixo X e Y, contendo as variáveis independentes e dependentes respectivamente), que neste exemplo foi dividido em três sub-regiões e em cada uma destas sub-regiões foi ajustada uma regressão linear com coeficientes independentes na forma:

$$y = a + bX \tag{2.61}$$

A transformação das variáveis utilizada por Briand et. al. (2004) é um modelo log-linear de três parâmetros com 2 variáveis explicativas:

$$\ln(y) = a_0 + a_1 \ln(x_1) + a_2 \ln(x_2) \tag{2.62}$$

Um conceito chave do modelo MARS é inserir nas fronteiras das sub-regiões condições de contorno de forma a permitir que as regressões sejam contínuas, porém com diferentes coeficientes para cada um dos trechos, nesta

dissertação as condições de contorno são encontradas utilizando um algoritmo genético que será apresentado no capítulo *3.4 Aplicação do modelo MARS*.

Desta forma as funções básicas são reescritas de forma a acrescentar as seguintes condições de contorno:

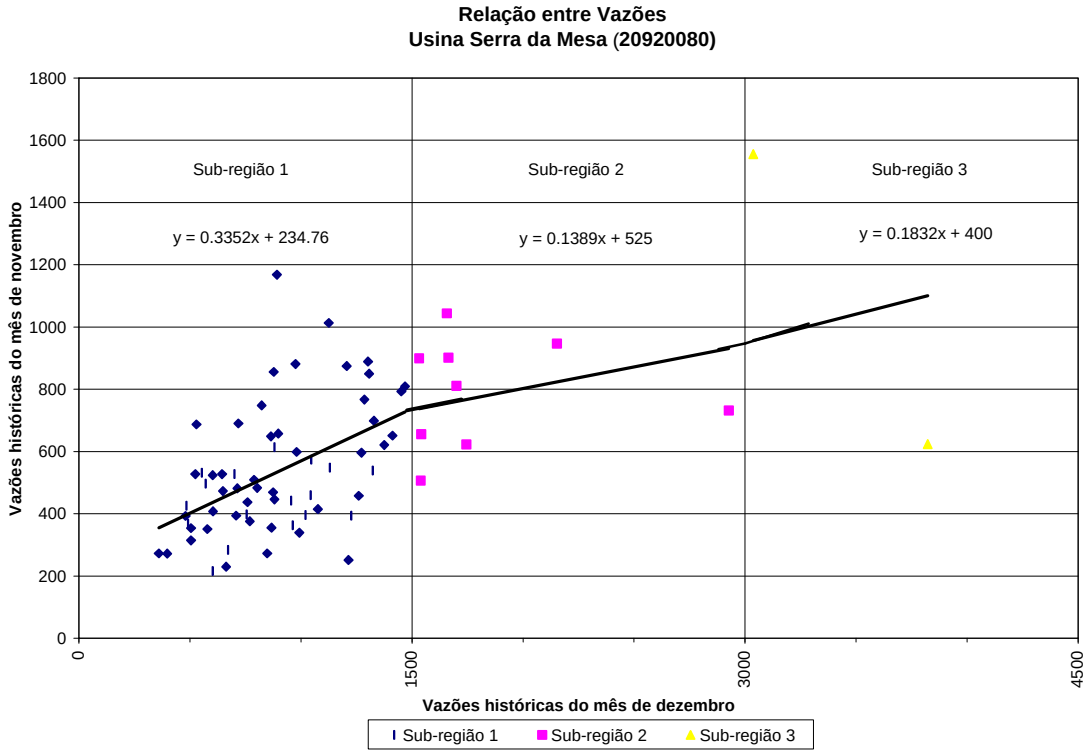
$$\max(0, X - c) \quad ou \tag{2.63}$$

$$\max(0, c - X) \tag{2.64}$$

Sendo X a variável independente e c uma constante (Briand et. al., 2000).

Após as condições de contorno serem determinadas, o resultado das funções básicas são usadas em novas variáveis independentes de forma a estimar novos modelos de regressão.

FIGURA 2.3 - Relação entre a série histórica de novembro e dezembro com regressão linear ajustada para sub-regiões.



A FIGURA 2.3 ilustra de forma bem simplificada como o modelo MARS efetua o ajuste nos coeficientes das regressões de forma a eliminar a descontinuidade nos limites de cada sub-região, obtendo desta forma uma continuidade em toda a amostra.

Segundo (Briand et. al., 2000) uma regra típica para aprovar a regressão em cada uma das sub-regiões é impor que em cada sub-região existam um número mínimo de pontos (por exemplo 10). No exemplo da FIGURA 2.3 o modelo MARS re-dividiria o espaço amostral de forma que em cada uma destas sub-regiões existam no mínimo 10 pontos para ajustar a regressão. De acordo com Friedman (1990), a escolha entre a precisão e a flexibilidade da aproximação da regressão é controlada pelo número de sub-regiões, e pela baixa ordem da derivada, permitida pela descontinuidade nas fronteiras das sub-regiões.

A regressão multivariada com splines adaptados (MARS) é uma técnica de regressão multivariada não linear que contém um vetor de variáveis dependentes $y(n\times 1)$ e uma matriz de variáveis explicativas $x(n\times p)$. O modelo MARS é representado por uma função que contém estas variáveis e um erro associado:

$$y = f(x) + e \tag{2.65}$$

Sendo e o vetor erro $(n\ X\ 1)$

O modelo MARS é considerado uma generalização do modelo CART (Xu et. al., 2004), sendo que o MARS possui o poder de superar as limitações do modelo CART (Classification and Regression Trees).

A idéia do modelo MARS é utilizar conceitos geométricos de sub-regiões com noções aritméticas multiplicativas, formando a função básica para o estimador $\hat{f}(x)$:

$$\hat{f}(x) = \sum_{m=1}^{m} a_m B_m(x) \tag{2.66}$$

Sendo a_m o coeficiente de expansão e B_m uma função que apresenta um argumento verdadeiro quando a função é positiva e valor igual a zero, caso contrário.

O primeiro algoritmo do modelo MARS é equivalente à regressão parcial recursiva (*2.3.1 Regressão Parcial Recursiva*), sendo sua função básica:

$$B_m(x) = \prod_{k=1}^{K_m} H\left[s_{km}\left(x_{v(k,m)} - t_{km}\right)\right] \tag{2.67}$$

Sendo Km o número de divisões (sub-regiões) de B_m, *H* a função de parada, s_{km} possuindo valor ± 1 associado à função de parada $v_{(k,m)}$ e a variável explicativa t_{km} representando o valor variável.

O primeiro algoritmo do modelo MARS produz funções contínuas, sendo que a função usada no algoritmo é um caso especial da função spline. Um problema inoportuno apresentado pela regressão parcial recursiva é a incapacidade de produzir boas aproximações para certas classes de funções.

Uma maneira de contornar este problema é substituir a função $H[\pm(c-t)]$ por uma função spline truncada $H[\pm(c-t)]_+^q$, não removendo a função $B_m^+(x)$ após a sua divisão, produzindo equações dependentes para as futuras divisões e restringindo o produto associado a estas funções básicas, com fatores que envolvem variáveis explicativas distintas.

Uma importante consideração sobre a generalização da regressão parcial é o grau de continuidade imposto à solução.

Um segundo algoritmo é acrescentado ao modelo MARS, incorporando a estratégia de truncar as funções exponenciais básicas, e substituí-las por fatores envolvendo variáveis distintas para o controle interno do primeiro algoritmo.

Removendo as funções básicas que não produzem descontinuidades no espaço de previsão, e que consequentemente não necessitam empregar subdivisões na função, é introduzido no modelo o terceiro algoritmo, que tem a

função de construir seqüências de modelos possuindo poucas funções básicas, apresentando como resultado um retorno melhor quando o algoritmo é executado.

Aplicando o resultado do segundo e do terceiro algoritmo obtém-se um modelo da forma:

$$\hat{f}(x) = a_0 + \sum_{m=1}^{m} a_m \prod_{k=1}^{km} \left[s_{km} \left(xv_{(k,m)} - l_{km} \right) \right]_+ \tag{2.68}$$

Sendo a_0 um coeficiente constante da função B_1, e o somatório a soma das funções B_m (2.67) produzidas pelo segundo algoritmo, s_{km} possuindo valor ± 1, a_m uma função constante.

A interpretação do modelo MARS é muito facilitada quando introduz-se no modelo a decomposição ANOVA (Analysis of Variance, Maidment, 1992):

$$\hat{f}(x) = a_0 + \sum_{km=1} f_i(x_i) + \sum_{km=2} f_{ij}(x_i, x_j) + \sum_{km=3} f_{ijk}(x_i, x_j, x_k) + \ldots \tag{2.69}$$

Sendo que:

O primeiro termo da equação (2.69) representa a função univariada:

$$f_i(x_i) = \sum_{\substack{km=1 \\ i \in V(m)}} a_m B_m(x_i) \tag{2.70}$$

O segundo termo da equação (2.69) representa a função bivariada:

$$f_{ij}(x_i, x_j) = \sum_{\substack{km=2 \\ i \in V(m)}} a_m B_m(x_i, x_j) \tag{2.71}$$

O terceiro termo da equação (2.69) representa a função trivariada:

$$f_{ijk}(x_i, x_j, x_k) = \sum_{\substack{km=3 \\ i \in V(m)}} a_m B_m(x_i, x_j, x_k) \tag{2.72}$$

A decomposição ANOVA identifica variáveis que entram no modelo unicamente de forma aditiva ou se estas variáveis estão envolvidas em

interações com outras variáveis, verificando seu nível de interação, e sua dependência com outras variáveis.

Entre os muitos aspectos contidos no modelo MARS, esta o critério chamado LOF (*lack-of-fit criterion*), presente no segundo e terceiro algoritmo, e o número máximo de funções de base $(M\max)$.

$$LOF(\hat{f}_m) = GCV(m) = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - \hat{f}_m(x_i)\right]^2 / \left[1 - \frac{C(m)}{N}\right]^2 \tag{2.73}$$

Sendo $\hat{f}$ (2.68), *N* o número de pontos, $y_1,\cdots,y_n$ as variáveis independentes, $x_1,\cdots,x_n$ as variáveis dependentes, e *C(m)* dada pela equação (2.76).

O GCV originalmente proposto por Craven e Wahba (1979) é a média quadrática do erro (resíduo).

Para explicar a função *GCV* (validação cruzada generalizada), é necessário primeiramente explanar sobre a função *CV* (validação cruzada), (Friedman e Silverman, 1989).

$$CV = \frac{1}{N}\sum_{i=1}^{n}\left[y_i - f_{-i}(x_i)\right]^2 \tag{2.74}$$

Sendo: f_{-i} a estimativa calculada com valores correntes dos parâmetros de controle (no caso do artigo de Frieman e Silvermam, 1989, o número de nós).

A função *CV* calcula o erro médio da previsão, sendo a função *GCV* uma aproximação da função *CV* com propriedades computacionais melhores.

Para otimizar o modelo é inserida uma função complexa $\tilde{C}(m)$ (2.75) na equação (2.73) com a finalidade de ajudar no ajuste dos dados, reduzindo a média quadrática do erro.

$$\tilde{C}(m) = C(m) + d.M \tag{2.75}$$

Sendo: M o número de variáveis da função e d o parâmetro de otimização de $C(m)$, tem-se a função complexa estimadora:

$$C(m) = trace\left(B\left(B^T B\right)^{-1} B^T\right) + 1 \tag{2.76}$$

Uma das principais vantagens do modelo MARS é a possibilidade de substituir o passo da função truncando a ordem da função spline.

Uma aproximação sugerida por Stone e Koo (1985) é modificar suavemente a função spline próxima ao contorno.

O modelo produzido pela equação (2.68) envolve a somatória de produtos:

$$b(x \mid s,t) = \left[s(x-t)\right]_+ \tag{2.77}$$

Outra estratégia para produzir modelos com contínuas derivadas, é truncar a função cúbica, obtendo-se:

$$C(x \mid s=+1, t_-, t, t_+) = \begin{cases} 0 & x \le t_- \\ p_+(x-t_-)^2 + r_+(x-t_-)^3 & t_- < x < t_+ \\ x-t & x \ge t_+ \end{cases} \tag{2.78}$$

$$C(x \mid s=-1, t_-, t, t_+) = \begin{cases} -(x-t) & x \le t_- \\ p_-(x-t_+)^2 + r_-(x-t_+)^3 & t_- < x < t_+ \\ 0 & x \ge t_+ \end{cases} \tag{2.79}$$

Com $t_- < x < t_+$ tem-se:

$$\begin{aligned} p_+ &= (2t_+ + t_- - 3t)/(t_+ - t_-)^2 \\ r_+ &= (2t + t_+ - t_-)/(t_+ - t_-)^3 \\ p_- &= (3t + 2t_- - t_+)/(t_- - t_+)^2 \\ r_+ &= (t_- + t_+ - 2t)/(t_- - t_+)^3 \end{aligned} \tag{2.80}$$

A decomposição ANOVA obtém todas as funções correspondentes a uma mesma variável.

No final do modelo é obtido o resultado dos coeficientes da função cúbica. Um importante limitante é que o fator básico nunca deve exceder o valor unitário (≤ 1).

Visando otimizar o segundo algoritmo é introduzido no modelo uma função j_{mv} e fixando o valor do coeficiente $\{a_j\}$, é possível generalizar esta função.

$$j_{mv}(xv) = E\left[B_m^{\,2} \left(\frac{R_{\backslash mv}}{B_m} \right) \mid x_v \right] / E\left[B_m^{\,2} \mid x_v \right] \tag{2.81}$$

Sendo $c_o, \cdots, c_j$ coeficientes do modelo, $R_{\backslash mv} = y - \hat{f}_{\backslash mv}$.

2.3.3.1 Considerações Computacionais do modelo MARS

Uma técnica muito popular é baseada na decomposição QR (Golub e Lour, 1983) que produz a matriz B:

$$B_{mi} = B_m(x_i) \tag{2.82}$$

As funções com média zero produzem uma matriz $B^T B$ proporcional a matriz de covariância das funções. Uma maneira simples de manter a proporcionalidade computacional é fazer uma aproximação de C:

$$C \approx nNM_{\max}^{\,4}(aN + bM_{\max})/L \tag{2.83}$$

Sendo a e b constantes de proporcionalidade e L o número de grupos da superfície.

$$L(a) = -\log_2\left[-\frac{1}{nN_m} \ln(1-a) \right] / 2.5 \tag{2.84}$$

Sendo N_m o número de observações, a um número de ajuste entre 0.05 e 0.10, e $1-a$ representando o número de valores positivos ou negativos compreendidos dentro do intervalo de confiança.

Uma propriedade muito especial é truncar a ordem das funções de grupos simples, representando aproximações splines com propriedades numéricas superiores, e minimizando a função GCV, reduzindo o tempo computacional de $O(M^3)$ para $O(M^2)$, fazendo com que o segundo algoritmo seja proporcional a:

$$C^* \approx nNM_{\max}{}^3(a + bM_{\max} / L) \tag{2.85}$$

Durante a execução do segundo algoritmo a função não necessita ser linearmente independente, portanto a matriz de covariância aparece normalmente como uma equação singular, sendo que a média quadrática do resíduo é obtida pela equação:

$$ASR(a) = \frac{1}{N}\left[\sum_{k=1}^{N}(y_k - \bar{y})^2 - \sum_{i=1}^{M+1} a_i(c_i + dD_{ii}a_i)\right] \tag{2.86}$$

Sendo *D* uma matriz diagonal *(M+1) x (M+1)* com elementos diagonais de *V*(matriz de covariância).

A equação (2.86) é usada no numerador da equação GCV (2.73), sendo d dado por um pequeno número usado para manter a estabilidade do modelo.'

2.3.4 Redes Neurais

As redes neurais são ferramentas matemáticas criadas a partir de observações do funcionamento do cérebro humano (Machado, 2005), procurando aproximar o processamento dos computadores ao cérebro, sendo que as redes neurais possuem um grau de interconexão similar a estrutura do cérebro (Rohn, 2002).

A rede neural é caracterizada por um número de processos operados paralelamente usando neurônios conectados em forma de circuito. Individualmente o desempenho dos neurônios é trivial, porém, funcionando coletivamente a rede neural possui a capacidade de resolver problemas complicados (Flood e Kartam, 1994).

Uma das características da rede neural é a capacidade de aprender, retendo o conhecimento dentro de si através da observação e experiência (Rohn, 2002).

As conexões entre neurônios procuram simular as conexões sinápticas biológicas através de uma variável chamada peso. O neurônio possui um ou mais sinais de entrada e um sinal de saída. Os pesos são valores que representam o grau de importância que determinada entrada possui em relação a determinado neurônio, ou seja, o peso muda em função da intensidade do sinal de entrada e dessa forma, muda o seu valor representativo para a rede (Rohn, 2002).

As redes neurais não podem garantir resultados absolutamente corretos, especialmente se os padrões são de alguma forma incompletos ou conflitantes. A maioria dos modelos de redes neurais possui alguma regra de treinamento, na qual o peso de suas conexões é apresentado de acordo com os padrões apresentados (Rohn, 2002).

Matematicamente uma rede neural artificial do tipo MLP (Multi Layer Perceptron, Machado, 2005) define uma função matemática cuja forma genérica é:

$$y_k = j\left(\sum_{j=1}^{q} w_{kj}\, j\left(\sum_{i=1}^{p} w_{ji}\, x_i + b_j\right) + b_k\right) \tag{2.87}$$

Onde *q* é o número de neurônios na camada intermediária e *p* é o número de neurônios na camada de entrada.

2.4 MÉTODOS DE INTERPOLAÇÃO

Este item tem a finalidade de descrever métodos de interpolação espacial que nesta dissertação serão adaptados para realizar previsões.

Os métodos de interpolação espacial têm a finalidade de encontrar uma função $f(x, y, z)$ que representem uma superfície espacial. Encontrando esta função é possível extrapolar os dados existentes para locais com dados

desconhecidos, sejam eles pontos de fronteira ou entre pontos conhecidos (Andriolo e Kaviski, 2005).

A interpolação espacial pode ser utilizada em regionalização hidrológica. Sendo a regionalização hidrológica qualquer processo de transferência de informações de estações com dados observados para outros locais em geral (sem observação), sendo ainda um mecanismo de transferência suficientemente geral para obter resultados em qualquer ponto de uma região (Barth et al., 1987).

A interpolação espacial é correntemente utilizada para extrapolar ou interpolar dados em função de coordenadas espaciais, sejam estes dados relativos a afluências, precipitação, insolação entre outros, com a finalidade de conhecer estes valores em pontos diferentes de sua origem.

Nesta dissertação será aproveitado o conceito de interpolação espacial para realizar previsões de afluências, sendo que os coeficientes da função que definem uma determinada malha espacial serão adaptados de forma que estes coeficientes representem uma malha temporal, onde será possível extrapolar os dados existentes de forma a estimar os valores futuros das afluências.

Com esta matriz serão determinados os coeficientes de uma função interpoladora, com estes coeficientes determinados, os mesmos serão utilizados para extrapolar os dados de afluência, aproveitando o fato de que as vazões de um mês *i* são dependentes da vazão do mês *i-1*.

2.4.1 Interpolação ótima

Antes de abordar a interpolação ótima propriamente dita, faz-se necessário uma introdução sobre os estimadores ótimos (Bras e Rodríguez-Iturbe,1984), com parâmetros conhecidas e desconhecidas.

2.4.1.1 Estimador ótimo com média e covariância conhecida

Definindo um sistema linear com um vetor aleatório ***X*** tem-se:

$$Z = HX + V \tag{2.88}$$

Sendo *Z* o vetor de observações, *V* o vetor de erro aleatório e a matriz *H* definindo o experimento. Em seguida as seguintes funções probabilísticas são avaliadas:

$$E[X] = m\,,$$

$$E[(X - m)(X - m)^T] = Var[X] = y\ ,$$

$$E[V] = 0\,, \tag{2.89}$$

$$E[VV^T] = R\,,$$

$$E[HXV^T] = 0\,,$$

O estimador ótimo $\hat{X}$ é dado por:

$$\hat{X} = \sum H^T R^{-1} Z + \sum y^{-1} m\,, \tag{2.90}$$

A matriz formada pelo erro médio quadrático deste estimador é:

$$\sum = E[(X - \hat{X})(X - \hat{X})^T]\,, \tag{2.91}$$

Sendo o valor esperado do estimador não-tendencioso $E[\hat{X}] = m$.

Usando-se a matriz inversa, o estimador ótimo é expresso:

$$\hat{X} = m + yH^T(HyH^T + R)^{-1}(Z - Hm) \tag{2.92}$$

$$\sum = y - yH^T(HyH^T + R)^{-1}Hy \tag{2.93}$$

Minimizar $\sum$ implica em outro estimador com erro médio quadrático $\hat{\sum}$, sendo $\sum - \hat{\sum}$ positivo semi-definido. Todos os elementos do estimador ótimo $\hat{X}$ correspondem a um elemento de *X*.

Tendo em vista que a melhor estimativa para uma equação linear é:

$Y = CX$, sendo a estimativa deste operador $\hat{Y} = C\hat{X}$.

Avaliando estas duas equações acima temos a estimativa do erro quadrático expresso por:

$$\sum_Y = C\sum_X C^T \tag{2.94}$$

Derivando a equação (2.90) e (2.91) obtém-se o estimador de mínima variância de Bayes:

$$C(X, \hat{X}) = (X - \hat{X}(Z))^T S(X - \hat{X}(Z)) \tag{2.95}$$

Sendo a matriz *S* simétrica e não negativa definida de maneira arbitrária.

Fazendo $f(X,Y)$ um conjunto de vetores aleatoriamente distribuídos em *X* e *Y* tem-se a equação de Bayes expressa:

$$B = \int_{-\infty}^{\infty}\{\int_{-\infty}^{\infty}(X - \hat{X}(Z))^T S((X - \hat{X}(Z))f(X \mid Z)dX\}f(Z)dZ \tag{2.96}$$

Sendo *f(X|Z)* uma distribuição condicionada de *X* dado *Z*.

Minimizando a integral da equação (2.96) em relação a *X*, e minimizando o valor esperado da estimativa dos erros temos:

$$\hat{X} = \int_{-\infty}^{\infty} Xf(X \mid Z)dX \tag{2.97}$$

Fazendo as variáveis aleatórias *X* e V da equação (2.88) assumir a distribuição Gaussiana, temos que a função densidade de probabilidade é expressa:

$$f(Z) = \frac{1}{(2p)^{m/2} \mid HyH^T + R \mid^{1/2}} \exp[-\frac{1}{2}(Z - Hm)^T (HyH^T + R)^{-1}(Z - Hm)] \qquad (2.98)$$

Substituindo a equação (2.98) na equação da função de densidade de probabilidade condicional (2.99) o estimador $\hat{X}$ é expresso pela equação (2.100).

$$f(X \mid Z) = \frac{f(Z \mid X) f(X)}{f(Z)} \qquad (2.99)$$

$$\hat{X} = (y^{-1} + H^T R^{-1} H)^{-1}(H^T R^{-1} Z + y^{-1} m) \qquad (2.100)$$

2.4.1.2 Estimador ótimo com constantes desconhecidas

Os estimadores descritos no item anterior são utilizados quando a média e a variância são conhecidas, porém em muitas aplicações estes parâmetros não estão disponíveis. Definindo o vetor *Z* de dados observados como:

$$Z = HX + V \qquad (2.101)$$

Sendo *X* completamente desconhecidos e *V* o erro aleatório, têm-se as seguintes funções probabilísticas:

$$E[V] = 0 \qquad (2.102)$$

$$E[VV^T] = R \qquad (2.103)$$

As constantes desconhecidas são interpretadas como uma variável aleatória com infinitas covariâncias, $y \rightarrow \infty I$. Usando este limite tem-se:

$$\hat{X} = \sum H^T R^{-1} Z \qquad (2.104)$$

$$\sum = E[(X - \hat{X})(X - \hat{X})^T] \qquad (2.105)$$

A não-tendenciosidade do estimador $\hat{X}$ implica novamente que $E[\hat{X}] = m$.

Não existindo ortogonalidade entre os erros e as observações têm-se:

$$E[(X-\hat{X})Z^T] = -[H^T R^{-1} H]^{-1} H^T \neq 0 \qquad (2.106)$$

Assumindo a distribuição normal do erro, obtém-se a função de probabilidade máxima, expressa como:

$$\frac{\partial f(X \mid Z)}{\partial X} = 0 \qquad (2.107)$$

Tendo-se então:

$$\hat{X} = (H^T R^{-1} H)^{-1} H^T R^{-1} Z \qquad (2.108)$$

2.4.1.2 Comentários gerais sobre o estimador ótimo

O estimador ótimo da variável aleatória é a combinação linear entre a média m e os desvios entre o valor observado *Z* e sua média. Dado que o peso das observações é expresso por:

$$W = (H^T R^{-1} H + y^{-1})^{-1} H^T R^{-1} \qquad (2.109)$$

A maior vantagem da estimativa linear é a independência do erro médio quadrático em relação às observações. Isto implica na medição acurada da base de dados.

A matriz do erro médio quadrático é interpretada posteriormente como a matriz de covariância da formulação de Bayesian.

2.4.1.3 A Interpolação Ótima

Considerando-se que é empregada a equação (2.110) para estimar a variável h(x), a aproximação por interpolação ótima, segundo Morin et. al. (1979), é aquela que determina os pesos pela minimização da variância dos erros de interpolação, de forma que resulta no seguinte sistema de equações:

$$h^*(\underline{x}) = \sum_{i=1}^{n} l_i(\underline{x}) h(\underline{x}_i). \qquad (2.110)$$

$$\sum_{i=1}^{n} l_i(\underline{x}) COV\left[h(\underline{x}_i), h(\underline{x}_j)\right] = COV\left[h(\underline{x}), h(\underline{x}_j)\right],\ j = 1,\ldots,\ n. \qquad (2.111)$$

Considerando-se que a estrutura de correlação espacial é homogênea e isotrópica, e fazendo-se as devidas substituições, o sistema de equações se reduz para:

$$\sum_{i=1}^{n} l_i(\underline{x})\, r\left[d(\underline{x}_i),(\underline{x}_j)\right] = r\left[d(\underline{x},\underline{x}_j)\right],\ j = 1,\ldots,n. \qquad (2.112)$$

Para que a estimativa h*(x) não seja tendenciosa, deve-se considerar que a soma dos pesos seja igual a um, resultando:

$$\sum_{i=1}^{n} l_i(\underline{x})\, r\left[d(\underline{x}_i),(\underline{x}_j)\right] + l_{n+1}(\underline{x}) = r\left[d(\underline{x},\underline{x}_j)\right],\ j = 1,\ldots,n. \qquad (2.113)$$

$$\sum_{i=1}^{n} l_i(\underline{x}) = 1. \qquad (2.114)$$

Para melhorar a regionalização das variáveis pode-se utilizar *x* em função de uma série de equações. Yevjevich e Karplus (1973) sugerem uma tabela (*Tabela 2.1 Modelos regionais matematicamente dependentes*) de modelos de regionalização matematicamente dependentes para componente estocásticas estacionárias para séries de precipitações mensais.

Tabela 2.1 Modelos regionais matematicamente dependentes.

Modelo	Função
I	$r = Ae^{(Bd + Cd\cos 2q + Dd\sin 2q)}$
II	$r = Ae^{(Bd)}$
III	$r = Ae^{(Bd)} + 1 - A$
IV	$r = Ae^{(Bd)} + Cd + 1 - A$
V	$r = A + Bd + Cd^2 + Dd^3 + Ed^4 + Fd^5$
VI	$r = (1 + Ad)^{-1}$
VII	$r = (1 + Ad)^{-n}$
VIII	$r = e^{(Ad)}$
IX	$r = \dfrac{e^{(Ad)}}{(1 + Bd)}$
X	$r = (1 + Ad)^{-0,50}$

Os modelos da *Tabela 2.1* representam uma simplificação da relação entre estações em *Lag 0* através do coeficiente de correlação das distâncias entre as diversas estações em estudo.

Sendo *d* a distância entre as *i-th* e *j-th* estações. A matriz formada pelo valor de *d* com as $d_{i,j}$ entre as *M* estações podem ser obtidas para regiões onde existem estações ou para regiões com estações específicas interseccionando a malha cartesiana. Em alguns casos a dimensão *M* e os elementos $d_{i,j}$ determinam a matriz de correlação selecionando $r = y(d)$, definindo o espaço dependente do processo e_i, sendo r independente de f, e a distância $d_{i,j}$ dada por:

$$\mathrm{d_{i,j}} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2} \qquad (2.115)$$

O modelo *I* foi utilizado por Caffey (1963) para estudos da variação da precipitação anual e precipitação anual efetiva em estações nos E.U.A e Canadá. Sendo este modelo linear em *d*, simétrico em relação aos eixos (longitude e latitude) e positivo para todos os valores de *B, C, D*, sendo q a direção. Já o modelo *II* também implementado por Caffey (1963) é uma simplificação do modelo *I* com r independente da orientação entre as estações.

No modelo *III* o parâmetro *A* é introduzido para que r convirja para (*1-A)* e *d* para o infinito, contrariando a hipótese de $r = 0$ para $d \to \infty$. Este modelo pode ser usado apesar de r convergir para (*1-A)* pois:

a) O limite da região estudada pode ser insuficiente para provar que para grandes distâncias (*d)* os valores de r comportem-se como pontos extremos.

b) O modelo admite que $r = 0$ para *A=1* antecipando o valor de *A* e *d* para grandes distâncias, com *B* negativo.

c) A intercessão de r =1 para *d=0* é de grande importância, assim como r = *0* para grandes valores de *d*.

O modelo *IV* é similar ao modelo *III* acrescido de um termo linear *Cd* com a possibilidade de r negativo para grandes distâncias.

Já o modelo *V* é um polinômio de 5° ordem, sendo r =*A* para *d=0* e r não especificamente definido para *d* infinito.

O modelo *VI* é uma função de um parâmetro com uma transformação linear expressa por: $\frac{1}{r} = 1 + Ad$.

O modelo *VII* não requer o expoente *n* que pode ser adaptado como *n=1*, resultando simplesmente numa linearização do modelo.

O modelo *VIII* representa uma função com decaimento exponencial.

O modelo *IX* é uma combinação do modelo *VI* e *VIII* com a vantagem de que o decaimento exponencial para pequenos valores de *d* ser inversamente proporcional para grandes valores de *d*.

E o modelo *X* é uma adaptação forçada do modelo *VII* com *n=0,50*.

Os modelos ilustrados na *Tabela 2.1* permitem reduzir a variável aleatória de um grupo de variáveis estocásticas estacionárias, identicamente distribuídas e independentes no tempo e dependentes entre si.

2.4.2 Técnica de Kriging

Em uma expressão simples o método de Kriging é uma forma de analisar uma superfície de tendência baseada na decomposição de dados espaciais em uma componente de tendência e uma componente local. Entretanto, a derivada da interpolação espacial dos dados semelhantes à Kriging é retratada por propriedades estatísticas conceituadas diferentemente das características clássicas da análise da superfície de tendência.

O método de Kriging é um interpolador perfeito, pois: $\hat{Z}(x_i) = Z(x_i)$.

Sendo o objetivo do método de Kriging encontrar o menor estimador linear não tendencioso para a função linear $Z(u)$. O estimador $\hat{P}$ desta função pode ser definido:

a) Linearidade: O estimador $\hat{P}$ é formado pela combinação linear dos valores observados (u_i).

$$\hat{P} = \sum_{i=1}^{n} l_i Z(u_i) \tag{2.116}$$

Sendo o coeficiente l_i constante, porém desconhecido, e a função *Z* adotada similar a um dos modelos da *Tabela 2.1*.

b) Não-tendenciosidade: Esta condição requer que o valor esperado do estimador $\hat{P}$ seja igual ao valor esperado do valor observado:

$$E[\hat{P}] = E[P] \tag{2.117}$$

c) Melhor critério: O estimador é considerado ótimo pela estimativa da menor variância, sendo a estimativa das variâncias dada pela média quadrática do erro, definida como:

$$s_p^2 = E[(P - \hat{P})^2] = \text{var}[P - \hat{P}] \tag{2.118}$$

Para as variáveis aleatórias serem estacionárias de segunda ordem é necessário que as seguintes condições sejam satisfeitas:

Média: $E[Z(u)] = m(u) = m$ (2.119)

Variância: $\text{var}[z(u)] = s^2(u) = s^2$ (2.120)

Covariância: $\text{cov}(u_1, u_2) = \text{cov}(u_1 - u_2) = \text{cov}(v)$ (2.121)

Sendo $v = u_1 - u_2$, e os pontos u_1 e u_2 independentes da localização e dependentes da diferença entre a sua localização (distância).

Se o processo satisfizer as restrições (2.119), (2.120), (2.121) é possível obter a média e a variância para a diferença entre a localização dos pontos.

Média: $E[z(u_1) - z(u_2)] = m(v)$ (2.122)

Variância: $\text{var}[z(u_1) - z(u_2)] = 2g(v)$ (2.123)

Sendo que a média e a variância são independentes da atual localização u_1 e u_2, e dependentes apenas do vetor das diferenças entre suas localizações.

A equação (2.123) define o semivariograma, mais especificamente o semivariograma estacionário, definido diretamente pelas diferenças entre as coordenadas. Sendo o semivariograma uma função da variância no campo probabilístico, usado para expressar a dispersão dos dados (Sakata S., et al, 2004).

A estacionaridade de segunda ordem implica:

$$\text{var}[z(u_1) - z(u_2)] = \text{var}[z(u_2)] + \text{var}[z(u_1)] - 2\,\text{cov}(u_1, u_2) \quad (2.124)$$

Supondo uma estimativa do processo $h(x_0)$ em um ponto qualquer com coordenadas expressas pelo vetor x_0 possa ser representado por uma combinação linear ponderada dos valores observados $h(x_j)$:

$$\hat{h}(x_0) = \sum_{j=1}^{n} w_j h(x_j) \quad (2.125)$$

Onde w_j é o peso correspondente ao ponto x_j. Chamando $\hat{h}(x_0)$ a estimativa de $h(x_0)$ fornecida pela equação (2.125), a chamada interpolação

ótima (Tabios e Salas, 1985) determina os pesos pela minimização da variância do erro de interpolação s_e^2, que é dado por:

$$s_e^2 = \mathrm{var}[h(x_0) - \hat{h}(x_0)] = \mathrm{var}[h(x_0) - \sum_{j=1}^{n} w_j h(x_j) \quad (2.126)$$

Expandindo a equação (2.126), resulta:

$$s_e^2 = s^2 - 2\sum_{j=1}^{n} w_j \,\mathrm{cov}[h(x_0)h(x_j)] + \sum_{j=1}^{n} \sum_{i=1}^{n} w_i w_j \,\mathrm{cov}[h(x_i)h(x_j)] \quad (2.127)$$

Onde s^2 é a variância do processo $h(x_0)$ e $\mathrm{cov}[h(x_i)h(x_j)]$ representa a covariância entre $h(x_i)$ e $h(x_j)$. Minimizando a equação anterior com relação aos pesos w_j para $j = 1,\ldots,n$ estações, resulta:

$$\sum_{i=1}^{n} w_i \,\mathrm{cov}[h(x_i)h(x_j)] = \mathrm{cov}[h(x_0)h(x_j)] \qquad j = 1,\ldots,n \quad (2.128)$$

Considerando a homogeneidade nas variâncias, os termos de covariância da equação anterior podem ser substituídos por:

$$\mathrm{cov}[h(x_i)h(x_j)] = s_i s_j r[h(x_i)h(x_j)] = s^2 r[h(x_i)h(x_j)] \quad (2.129)$$

$$\mathrm{cov}[h(x_i)h(x_j)] = s^2 r[h(x_0)h(x_j)] \quad (2.130)$$

Onde $s^2 r[h(x_i)h(x_j)]$ e $s^2 r[h(x_0)h(x_j)]$ são coeficientes de correlação espacial.

Para estimar estes coeficientes de correlação, é necessário definir uma função de correlação espacial. Considerando uma estrutura de correlação espacial homogênea e isotrópica, $r[h(x_i)h(x_j)]$ pode ser escrito como uma função da distância apenas. Então, $r[h(x_i)h(x_j)]$ torna-se $r[d(x_i, x_j)]$, em que $d(x_i, x_j)$ é a distância entre os pontos x_i e x_j.

Assim, várias formas de técnica de Kriging têm sido propostas e aplicadas em estudos hidrológicos. Neste trabalho é descrito o método de

Kriging como apresentado por Loaiciga et. al. (1988), que avaliam os dados espaciais por meio de duas componentes: a de natureza regional e a de variação local. A variável regionalizada h($\underline{x}$) é modelada como a soma do componente de natureza regional m($\underline{x}$), com a componente aleatória ε($\underline{x}$):

$$h(\underline{x}) = m(\underline{x}) + e(\underline{x}), \tag{2.131}$$

$$m(\underline{x}) = \sum_{u=1}^{k} b_u f_u(\underline{x}). \tag{2.132}$$

Os coeficientes β_u são constantes, mas desconhecidos e as funções $f_u(\underline{x})$ são adotadas e dependem das coordenada de localização. A componente aleatória é assumida como estacionária de segunda ordem, com média zero e covariância definida em função da distância que separa dois pontos localizados no espaço orientado. Ou seja, que:

$$C(d_{ij}) = E\left[e(\underline{x}_i)e(\underline{x}_j)\right] = COV\left[h(\underline{x}_i), h(\underline{x}_j)\right] = \sum_{u=1}^{P} s_u{}^2 g_u(d_{ij}), \tag{2.133}$$

Sendo que $g_u(d_{ij})$ são adotadas e dependem da distância d_{ij}. Os parâmetros σ^2_u são desconhecidos e representam a variabilidade da diferença de escalas. Quando h($\underline{x}_i$) representa uma quantidade média à covariância é determinada por:

$$C(d_{ij}) = \left\{ \int_{v_i} dv \int_{v_j} E\left[e(v)e(u)\right] du \right\} / \left(V_i V_j\right), \tag{2.134}$$

Sendo que V_i e V_j, são os domínios médios, com centros em $\underline{x}_i$ e $\underline{x}_j$, respectivamente.

Para implementar o método é necessário que os parâmetros β_1,..., β_k, e as componentes da variância $a_1^2, ..., a_p^2$, sejam estimados. Loaiciga et. al. (1988) sugerem a utilização dos seguintes meios: i) mínimos quadrados generalizado para a estimativa dos parâmetros β_1,..., β_k; ii) máxima verossimilhança das informações totais para o cálculo simultâneo das componentes da variância e dos parâmetros β_1,..., β_k; iii) máxima

verossimilhança restrita para o cálculo das componentes da variância; e iv) estimativa de mínima invariante quadrática não tendenciosa das componentes da variância.

O modelo FIML (máxima verossimilhança das informações totais para o cálculo simultâneo das componentes da variância e dos parâmetros $\beta_1,..., \beta_k$) requer a consistência inicial do estimador $\hat{P}$ através de um processo interativo iniciado no local de máxima. Sendo que o modelo necessariamente não converge para o local ótimo.

O modelo RML (máxima verossimilhança restrita para o cálculo das componentes da variância) é uma adaptação do modelo FIML introduzida por Patterson e Thompson (1971) e citada por Loaiciga et. al. (1988), aproximando o vetor de dados $h(x)$ projetado dentro do subespaço com média $fb = 0$. Sendo a covariância e a variância componentes que podem ser aproximados pela inversa negativa da matriz Hessiana (matriz com derivadas segundas).

O método de MINQUE (estimativa de mínima invariante quadrática não tendenciosa das componentes da variância) é uma alternativa para estimar a matriz de variância e covariância de um modelo geral heteroquedástico, através da combinação linear $c^T s$ estimado pelo quadrado da função $z^T Az$, sendo que esta função deve ser invariável, não tendenciosa e o estimador um mínimo quadrático.

Cada componente da variância s^2 do processo MINQUE, é estimado pela combinação linear de todos os quadrados dos erros (resíduos) com pesos relativos à construção da matriz (Bera et. al., 2002), sendo que o método MINQUE apresenta a vantagem de resolver o sistema de equações através de uma combinação linear de equações, produzindo componentes da variância.

Segundo Lele e Taper (2002) o método MINQUE é um estimador ótimo apenas para pontos escolhidos próximos e antecedentes aos pontos de base, sendo os problemas não exclusivos.

2.4.3 Interpolação por superfície “spline”

Creutin e Obled (1982) apresentam uma função interpoladora proposta por Duchon, em 1976. A função interpoladora “spline” proposta, satisfaz um critério ótimo de alisamento, sendo única e descrita pela seguinte expressão:

$$h^*(\underline{x}) = a + \underline{b}^T \underline{x} + \sum_{i=1}^{n} j_i K(\underline{x}, \underline{x}_i), \tag{2.135}$$

sendo:

$$K(\underline{x}, \underline{x}_i) = \left\|\underline{x} - \underline{x}_i\right\|^2 \ln\left\|\underline{x} - \underline{x}_i\right\|^2. \tag{2.136}$$

Os coeficientes α, $\underline{\beta}$ e $\underline{\sigma}$, são obtidos pela solução do seguinte sistema de equações:

$$\underline{K}\,\underline{j} + a\underline{1}_n + \underline{X}\,\underline{b} = \underline{h}(\underline{x}), \tag{2.137}$$

$$\underline{1}_n^T \underline{j} = 0, \tag{2.138}$$

$$\underline{X}^T \underline{j} = 0, \tag{2.139}$$

Sendo que $\underline{K}$ é uma matriz simétrica nXn, cujo elemento (i,j) é igual a $k(\underline{x}_i,\underline{x}_j)$; $\underline{1}_n$ é um vetor unitário com n elementos; $\underline{X}$ é uma matriz com n colunas, cuja coluna i é igual ao vetor x_i; $\underline{h}(\underline{x})$ é um vetor com n elementos; e $\underline{0}$ é uma matriz nula com dimensão igual ao número de componentes do vetor $\underline{x}$.

2.4.4 Funções multiquadráticas

Supõe-se que as observações $h_{t,T}(\underline{x}_1),...,h_{t,T}(\underline{x}_n)$, do total precipitado em t horas e T anos de recorrência, são avaliadas em função de suas localizações, descritas pelos vetores cartesianos $\underline{x}_1,...,\underline{x}_n$, sendo que $\underline{x}_i$, representa posições no espaço de coordenadas x, y e z, correspondendo respectivamente a latitude, longitude e altitude, do local da estação pluviográfica.

A função de aproximação $h^*_{t,T}(\underline{x})$ é escrita como uma combinação linear de n funções multiquadráticas conhecidas $fi(\underline{x})$:

$$h^*_{t,T}(\underline{x}) = \sum_{i=1}^{n} b_i f_i(\underline{x}), \tag{2.140}$$

$$f_i(\underline{x}) = \left[(x_i - x)^2 / s_x^2 + (y_i - y)^2 / s_y^2 + (z_i - z)^2 / s_z^2\right]^{1/2}, \tag{2.141}$$

Sendo que s_x^2, s_y^2, s_z^2 representam a variância das latitudes, longitudes e altitudes, respectivamente, das localizações das estações pluviográficas.

Os coeficientes β_i, com i=1,...,n, são determinados com a condição de que a expressão (2.140) seja verdadeira para todos os *n* locais das estações pluviográficas, resultando num sistema de equações lineares.

Como se têm para cada estação pluviográfica (nt)(nT) variáveis $h_{t,T}$, uma para cada combinação de $t=t_1,...,t_{nt}$, e $T=T_1,...,T_{nT}$, deve-se definir (nt)(nT) conjuntos de parâmetros β, sendo um conjunto para cada variável.

2.4.5 Método da mínima Curvatura

O método da mínima curvatura (Smith e Wessel, 1990) é muito utilizado nas ciências da terra. O método interpola os dados do gride de superfície utilizando derivadas segundas contínuas e minimizando a curvatura quadrada total.

Em uma dimensão, o método da mínima curvatura utiliza a função com derivadas segundas contínuas, interpolando os dados com restrição exatamente como a curvatura mínima total é interpolada com as splines cúbicas naturais.

Já em duas dimensões, o método da mínima curvatura interpola uma spline bicúbica natural, com a mesma oscilação e pontos de inflexão unidimensionais.

O algoritmo da mínima curvatura é representado pela equação:

$$c = \iint \left(\nabla^2 x\right)^2 dxdy \tag{2.142}$$

A equação (2.142) representa aproximadamente a curvatura total de *z* quando $\left|\nabla z\right|$ é pequeno. Minimizando a equação (2.142) através da equação diferencial obtém-se:

$$\nabla^2\left(\nabla^2 z\right) = \sum_i f_i d\left(x - x_i, y - y_i\right) \tag{2.143}$$

Escrevendo as semelhanças $z \to z_i$ e $(x, y) \to (x_i, y_i)$ se obtém as condições de contorno:

$$\frac{\partial^2 z}{\partial n^2} = 0 \tag{2.144}$$

$$\frac{\partial}{\partial n}(\nabla^2 z) = 0 \tag{2.145}$$

$$\frac{\partial^2 z}{\partial x \partial y} = 0 \tag{2.146}$$

Aplicando as condições de contorno (2.144) (2.145) (2.146) e ajustando a escala de soluções com D e q, obtém-se:

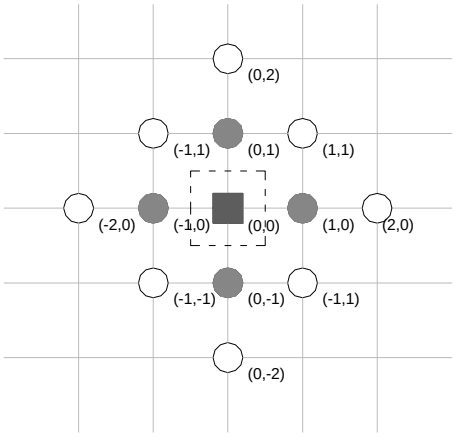
$$(1 - T_I)\nabla^2(\nabla^2 z) - T_I \nabla^2 z = \sum_i f_i d\left(x - x_i, y - y_i\right) \tag{2.147}$$

Sendo T_I o parâmetro de tensão, e o índice *I* indicando a tensão interna, f_i representa a relação entre *q/D*, sendo *D* a rigidez e *q* a tensão vertical normal.

Existem muitas maneiras de resolver as equações (2.143) e (2.144). Uma maneira é utilizar a combinação linear com funções de Green, construindo uma matriz de equações ***Gf*** =***d*** sendo ***G*** a matriz construída com as funções de Green, ***f*** o vetor com as incógnitas f_i e ***d*** o vetor com as restrições conhecidas $z(x_i, y_i) = z_i$.

Para resolver estas equações (2.143) e (2.147) é necessário recorrer ao método numérico, sendo possível expressar as equações (2.143) e (2.147) como diferenças finitas e construir uma malha com 12 nós.

FIGURA 2.4 Malha para resolução por diferenças finitas das equações (2.142) e (2.147).



Utilizando diferenças finitas de segunda ordem e expandindo as séries de Taylor, prevêem-se os valores da superfície interpoladora, e devido a não perfeita convergência do método é necessário associar a resolução um coeficiente de tolerância (erro) ao método.

Na solução do método da mínima curvatura é utilizada a interação por diferenças finitas, sendo z_{00} (2.148) referente à coordenada do ponto desejado. Aproximando as derivadas por diferenças finitas centrada, e assumindo $\Delta x = 1$

e considerando $a = \frac{\Delta y}{\Delta x}$, é possível resolver numericamente a equação (2.147), quando a mesma é igual à zero.

$$z_{00} = -\left[\left(6+8a^2+6a^4\right)\left(1-T_I\right)+2\left(1+a^2\right)T_I\right]^{-1}\times$$
$$\left\{\left(1-T_I\right)\left[\begin{matrix} z_{20}+z_{-20}+a^4\left(z_{02}+z_{0-2}\right)+ \\ 2a^2\left(z_{11}+z_{-11}+z_{1-1}+z_{-1-1}\right)\end{matrix}\right]-\left[4\left(1+a^2\right)\left(a-T_I\right)+T_I\right]\left[\begin{matrix} z_{10}+z_{-10}+ \\ a^2\left(z_{01}+z_{0-1}\right)\end{matrix}\right]\right\} \quad (2.148)$$

Sendo a equação (2.148) a expressão das diferenças para equações homogêneas.

Para resolver a expressão das diferenças incluindo restrição do datum é necessário utilizar a expansão de Taylor de segunda ordem, sendo necessário introduzir o ponto z_k (2.149) para calcular o ponto z_{00} (2.150).

$$z_k = z_{00} + x_k\frac{\partial z}{\partial x} + h_k\frac{\partial z}{\partial y} + \frac{1}{2}x_k{}^2\frac{\partial^2 z}{\partial x^2} + x_k h_k\frac{\partial^2 z}{\partial x \partial y} + \frac{1}{2}h_k{}^2\frac{\partial^2 z}{\partial y^2} \quad (2.149)$$

Multiplicando a expressão (2.149) por um número real b_k e normalizando a malha com Δx e a anisotrópico, sabendo que x e ah representam as distâncias fracionadas da malha, é possível encontrar o valor de z_{00}:

$$z_{00} = \left\{T_I\sum b_k - 2\left(1-T_I\right)\left[\left(1+a^4\right)-\left(1+a^2\right)\sum b_k\right]\right\}^{-1}\times$$
$$\left\{\left(1-T_I\right)\left\{\begin{matrix} z_{20}+z_{-20}+a^4\left(z_{02}+z_{0-2}\right)+2a^2\left(z_{11}+z_{1-1}+z_{-11}+z_{-1-1}\right) \\ -2\left(1+a^2\right)\times\left[z_{10}+z_{-10}+a^2\left(z_{10}+z_{0-1}\right)+\sum b_k z_k - T_I\sum b_k z_k\right]\end{matrix}\right\}\right\} \quad (2.150)$$

Sendo que no caso particular da mínima curvatura $T_I = 0$.

É possível resolver as equações (2.148) e (2.150) por métodos interativos (ex. Método de Gauss-Seidel), usando um fator de relaxação w. Sendo $1 < w < 2$ tendo como intuito acelerar a convergência. Utilizando este conceito podemos elaborar a equação de convergência:

$$z_{ij}^{new} = (1-w) z_{ij}^{old} + w z_{ij}^{new} \quad (2.151)$$

Para uma melhor convergência é necessário limitar o valor de x :

$$máx\left| z_{ij}^{new} - z_{ij}^{old} \right| < x \quad (2.152)$$

3 MODELOS APLICADOS

Este capítulo tem o objetivo de descrever como foram aplicados os modelos descritos no capítulo ***Erro! A origem da referência não foi encontrada.*** para realizar previsões de vazões médias mensais no horizonte de 1 a 12 meses.

3.1 MODELO PAR DE ORDEM 6

O modelo linear adotado neste trabalho foi o Periódico Auto-Regressivo de ordem 6 (PAR [6]), e tem como intuído fornecer um parâmetro de comparação entre um modelo linear amplamente conhecido e os modelos não-lineares desenvolvidos ao longo deste trabalho.

3.1.1 Aplicação do modelo PAR de ordem 6

O modelo periódico auto-regressivo de ordem 6 pode ser ilustrado como:

$$Z(t) = AZ(t-1) + BZ(t-2) + CZ(t-3) + DZ(t-4) + EZ(t-5) + FZ(t-6) + Ge(t) \quad (3.1)$$

Sendo *A,B,C,D,E,F* uma matriz de *nxn* parâmetros, o vetor *Z* composto por *n* parâmetros diferentes porém interdependentes, e o vetor *G* composto pelo erro aleatório do modelo.

Multiplicando a equação (3.1) pela sua transposta $Z^t(t)$:

$$E[Z(t)Z^t(t)] = AE[Z(t-1)Z^t(t)] + BE[Z(t-2)Z^t(t)] + \ldots + FE[Z(t-1)Z^t(t)] + GE[e(t)[Z^t(t-1)A^t + Z^t(t-2)B^t + \ldots + Z^t(t-6)F^t + e^t(t)G^t] \quad (3.2)$$

Definindo a matriz de covariância como:

$$M_0 = E[Z(t)Z^t(t)] \quad (3.3)$$

Multiplicando a equação (3.1) por $Z(t)^t$, e utilizando os valores esperados dos parâmetros, obtém-se a matriz de covariância M_0:

$$M_0 = AM_1^t + BM_2^t + CM_3^t + DM_4^t + EM_5^t + FM_6^t + GG^t \quad (3.4)$$

Para obter a covariância em *Lag 1* (M_1) multiplica-se a equação (3.1) por $Z^t(t-1)$, para a covariância em *Lag 2* (M_2) multiplica-se a equação (3.1) por $Z^t(t-2)$, e assim sucessivamente.

Preservando as médias mensais, a covariância entre a vazão do mês em análise com os seis meses anteriores e o desvio padrão destes meses, e com base na série histórica é possível calcular o valor da vazão no mês em análise em relação aos 6 meses anteriores, sejam estes dados anteriores observados ou previstos:

$$\left(X_{i,j} - m_j\right) = A_j\left(X_{i,j-1} - m_{j-1}\right) + B_j\left(X_{i,j-2} - m_{j-2}\right) + \ldots + F_j\left(X_{i,j-6} - m_{j-6}\right) + G_j e_{i,j} \quad (3.5)$$

Sendo *i* o ano, *j* o mês, m_j a vazão média do mês *j*, $X_{i,j}$ a vazão observada ou prevista do ano *i* e do mês *j*.

Assumindo que a auto-correlação dos resíduos (e) é igual a 0 tem-se:

$$Q_{i,j} = \begin{bmatrix} \Phi_{j-1} \\ \Phi_{j-2} \\ \vdots \\ \Phi_{j-6} \end{bmatrix} \times \begin{bmatrix} Y_{i,j-1} \\ Y_{i,j-2} \\ \vdots \\ Y_{i,j-6} \end{bmatrix} + U_j + V_{i,j} \quad (3.6)$$

Este modelo de previsão deve satisfazer as seguintes propriedades:

a) Preservar a média do vetor *Q*;

b) Preservar a variância do vetor *Y*;

c) Preservar o coeficiente de correlação entre Q_j e $Q_{j-1}, \cdots, Q_{j-6}$;

d) Preservar o coeficiente de correlação entre $Q_{j-1}, \cdots, Q_{j-6}$;

e) Preservar a relação cumulativa entre *X*=*CY*, desagregando os valores da componente anual.

Tendo em vista os conceitos descritos anteriormente (3.1) deve-se primeiramente elaborar uma matriz com todos os dados da série histórica disponíveis para um determinado local:

$$\begin{bmatrix} Q_{1,1} & \cdots & Q_{1,12} \\ \vdots & \ddots & \vdots \\ Q_{n,1} & \cdots & Q_{n,12} \end{bmatrix} \tag{3.7}$$

Sendo *n* o número de anos disponíveis para as vazões dos meses de janeiro a dezembro.

Em seguida é montada uma matriz com as correlações entre a série de vazão do mês em questão em relação ao mês anterior e assim sucessivamente, até obter a relação entre o mês em questão e este mesmo mês defasado em dois anos:

$$\begin{bmatrix}
Q_{1_n,12_{n-1}} & \cdots & Q_{1_n,1_{n-1}} & Q_{1_n,12_{n-2}} & Q_{1_n,11_{n-2}} & \cdots & Q_{1_n,10_{n-1}} & Q_{1_n,11_{n-1}} & Q_{1_n,12_{n-1}} & Q_{1_n,1_{n-2}} \\
Q_{2_n,1_n} & Q_{2_n,12_{n-1}} & \cdots & Q_{2_n,2_{n-1}} & Q_{2_n,1_{n-2}} & \cdots & Q_{2_n,11_{n-1}} & Q_{2_n,12_{n-1}} & Q_{2_n,1_{n-2}} & Q_{2_n,2_{n-2}} \\
Q_{3_n,2_n} & Q_{3_n,1_n} & Q_{3_n,12_{n-1}} & \cdots & Q_{3_n,2_n} & \cdots & Q_{3_n,12_{n-1}} & Q_{3_n,1_{n-2}} & Q_{3_n,2_{n-2}} & Q_{3_n,3_{n-2}} \\
Q_{4_n,3_n} & Q_{4_n,2_n} & Q_{4_n,1_n} & Q_{4_n,12_{n-1}} & \cdots & Q_{4_n,12_{n-1}} & Q_{4_n,1_{n-2}} & Q_{4_n,2_{n-2}} & Q_{4_n,3_{n-2}} & Q_{4_n,4_{n-2}} \\
Q_{5_n,4_n} & Q_{5_n,3_n} & Q_{5_n,2_n} & Q_{5_n,1_n} & Q_{5_n,12_{n-1}} & \cdots & Q_{5_n,2_{n-2}} & Q_{5_n,3_{n-2}} & Q_{5_n,4_{n-2}} & Q_{5_n,5_{n-2}} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
Q_{11_n,10_n} & Q_{11_n,9_n} & Q_{11_n,8_n} & Q_{11_n,7_n} & \cdots & Q_{11_n,1_n} & Q_{11_n,12_{n-1}} & \cdots & Q_{11_n,12_{n-2}} & Q_{11_n,11_{n-2}} \\
Q_{12_n,11_n} & Q_{12_n,10_n} & Q_{12_n,9_n} & Q_{12_n,8_n} & Q_{12_n,7_n} & \cdots & Q_{12_n,12_{n-1}} & \cdots & Q_{12_n,1_{n-1}} & Q_{12_n,12_{n-2}}
\end{bmatrix}$$

(3.8)

A matriz (3.8) é uma matriz 12x24 e ilustra a correlação entre o mês em questão (1..12) do ano *n* em relação ao mês anterior e assim sucessivamente, até encontrar a correlação entre o mês em questão no ano *n* em relação a este mesmo mês no ano *n-2*.

Obtendo-se esta matriz (3.8) é possível resolver o sistema de equações:

$$\begin{bmatrix} 1 & r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} & r_{1,5} \\ r_{1,1} & 1 & r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} \\ r_{1,2} & r_{1,1} & 1 & r_{1,1} & r_{1,2} & r_{1,3} \\ r_{1,3} & r_{1,2} & r_{1,1} & 1 & r_{1,1} & r_{1,2} \\ r_{1,4} & r_{1,3} & r_{1,2} & r_{1,1} & 1 & r_{1,1} \\ r_{1,5} & r_{1,4} & r_{1,3} & r_{1,2} & r_{1,1} & 1 \end{bmatrix} \times \begin{bmatrix} \Phi_{1,1} \\ \Phi_{1,2} \\ \Phi_{1,3} \\ \Phi_{1,4} \\ \Phi_{1,5} \\ \Phi_{1,6} \end{bmatrix} = \begin{bmatrix} r_{1,1} \\ r_{1,2} \\ r_{1,3} \\ r_{1,4} \\ r_{1,5} \\ r_{1,6} \end{bmatrix} \quad (3.9)$$

Sendo que em $r_{i,j}$ *i* representa a correlação do mês em questão e *j* representa a correlação do mês anterior.

Obtém-se desta maneira um vetor com 6 pesos $\Phi_{i,j}$ para cada um dos 12 meses, sendo possível montar uma matriz de 12x6.

Com os coeficientes $\Phi_{i,j}$ calcula-se a média m, e o desvio padrão s para cada um dos doze meses.

$$Z_i = \frac{V_i - m_i}{s_i} \quad (3.10)$$

Sendo V_i a vazão no mês *p-j*, onde *p* é o mês em que se deseja prever a vazão e *j=1..6* referentes ao *lag* do modelo PAR.

$$a = \sum_{i=1}^{6} \Phi_i \times Z_i \quad (3.11)$$

$$V_p = a \times s_p + m_p \quad (3.12)$$

Sendo *p* o mês para o qual se deseja prever a vazão, e V_p a vazão prevista para o mês *p*.

Como este trabalho tem o intuído de prever vazões no horizonte de 1 a 12 meses, a equação (3.12) deve ser calculada 12 vezes sendo que a vazão V_i equação (3.10) é a vazão no mês (1..6) anteriores a vazão prevista, podendo este valor V_i ser um valor observado ou previsto.

O código fonte do modelo PAR[6] adotado esta ilustrado no Apêndice F.1

3.2 APLICAÇÃO DO MODELO MULTIQUADRÁTICO

O método de interpolação utilizando funções multiquadráticas descrito no capítulo 2 foi adaptado de forma a possibilitar sua aplicação na previsão de vazões afluentes mensais.

Sendo assim, uma matriz *A* 240x240 é construída utilizando como parâmetros o mês e o ano:

$$A = \begin{bmatrix} 0 & a_{1,2} & a_{1,3} & \cdots & a_{1,240} \\ a_{2,1} & 0 & a_{2,3} & \cdots & a_{2,240} \\ a_{3,1} & a_{3,2} & 0 & \cdots & a_{3,240} \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ a_{240,1} & a_{240,2} & a_{240,3} & \ddots & a_{240,240} \end{bmatrix} \tag{3.13}$$

Sendo $a_{i,j} = \sqrt{k_a{}^2 + k_m{}^2}$ onde:

a) para $i \,(\mathrm{mod})\, 12 \leq 6$

$k_m = i(\mathrm{mod})12$ K_m referente ao mês (3.14)

$k_a = i(div)12$ K_a referente ao ano

Lembrando que *mod* é o operador resto da divisão e *div* é o operador de divisão entre número inteiro da divisão.

b) para $i \,(\mathrm{mod})\, 12 > 6$

$k_m = 12 - i(\mathrm{mod})12$ K_m referente ao mês (3.15)

$k_a = 1 + i(div)12$ K_a referente ao ano

Com $a_{i,j} = a_{j,i}$

Após a matriz *A* ser construída é obtido a sua inversa (A^{-1}) utilizando o método de Gauss-Jordan (Burden e Faires, 2003).

Em seguida é construído um vetor *H* com a série de vazões do mês a ser previsto.

$H = \lfloor Q_{i,1}, Q_{i,2}, \cdots, Q_{i,n} \rfloor$ sendo *i* o mês em que será feita a previsão e *1..n* o primeiro valor de vazão da série para o mês *i*, e *n* o último valor da vazão disponível para o mês *i*.

Para o vetor *H* é calculado a média, o desvio padrão e o coeficiente de assimetria.

O cálculo destes três parâmetros é executado de acordo com a seguinte seqüência:

a) Ordenam-se os valores das vazões do vetor *H* em ordem crescente;

b) Calcula-se a mediana $\tilde{x}$:

$$\tilde{x} = Q_{n/2-1} \qquad \text{para } n \text{ ímpar} \qquad (3.16)$$

$$\tilde{x} = \frac{1}{2}(Q_{n/2} + Q_{n/2+1}) \qquad \text{para } n \text{ par} \qquad (3.17)$$

Sendo *n* o número total de dados disponíveis (anos da série histórica).

c) em seguida é calculado o coeficiente de assimetria (*a*) a média *(med)* e o desvio padrão (*dpd*).

$$a = \frac{Q_1 \times Q_n - \tilde{x}^2}{Q_1 + Q_n - 2\tilde{x}} \qquad (3.18)$$

Onde Q_1 é a menor vazão da série e Q_n é a maior vazão da série.

$$med = \frac{\sum_{i=1}^{n} \ln|Q_i - a|}{n} \qquad (3.19)$$

$$dpd = \sqrt{\frac{\sum_{i=1}^{n} (\ln|Q_i - a|)^2}{n} - med^2} \tag{3.20}$$

Obtendo o coeficiente de assimetria, a média e o desvio padrão, é construída uma matriz *J* utilizando os dados observados dos últimos 20 anos.

$$J = \begin{bmatrix} J_{n-20,1} & J_{n-20,2} & \cdots & J_{n-20,12} \\ J_{n-19,1} & J_{n-19,2} & \cdots & J_{n-19,2} \\ \ddots & \ddots & \ddots & \ddots \\ J_{n,1} & J_{n,2} & \cdots & J_{n,12} \end{bmatrix} \tag{3.21}$$

$$J_{i,j} = \ln|Q_{i,j} - a_j| - \frac{med_j}{dpd_j} \tag{3.22}$$

Sendo *i* o ano, *j* o mês e *n* o último ano observado.

Com o inverso da matriz *J* $\left(J^{-1}\right)$ é calculado o vetor *X*:

$$X_j = \sum_{i=1}^{240} \left(J_{i,j}\right)^{-1} \times J_i \tag{3.23}$$

$$Q_i = a_i + \exp\left[med_i + dpd_i \times \sum_{j=1}^{240} X_j \sqrt{K_a{}^2 + K_b{}^2} \right] \tag{3.24}$$

Sendo que K_a e K_b dados pelas equações (3.14) e (3.15).

O código fonte desenvolvido para prever as vazões afluentes utilizando as funções multiquadráticas esta ilustrado no apêndice (*F.2 Modelo Multiquadrático)* deste trabalho.

3.3 APLICAÇÃO DO MODELO DE INTERPOLAÇÃO ÓTIMA

O método de interpolação ótima descrito no capítulo ***Erro! A origem da referência não foi encontrada.*** foi adaptado de forma a se obter uma previsão contínua utilizando os últimos 96 valores de vazão mensal observada para obter-se a vazão prevista.

Ao iniciar a calibração do modelo (obtenção dos parâmetros) o modelo de interpolação ótima inicia lendo toda a série histórica, desta maneira é elaborado um vetor com as vazões observadas de forma que:

$$\vec{Q} = [Q_1, Q_2, \ldots, Q_n] \tag{3.25}$$

Sendo Q_1 o primeiro dado de vazão observada e Q_n o último valor observado. Elaborando este vetor são selecionadas as vazões observadas relativas a determinado mês, sendo então montado um vetor *X*:

$$X_{i=1,j} = [X_1, X_2, \ldots, X_n] \tag{3.26}$$

Com: $X_1 = Q_1$, $X_2 = Q_{1+12}$, $X_3 = Q_{1+24}$, generalizando tem-se: $X_{i=1,j}^{i=120} = Q_{i+12(j-1)}$ com *j* representando os n_j anos existentes e *i* a localização do primeiro valor de vazão a ser utilizado.

Ilustrando $X_{i=1,j}^{i=120}$ de forma matricial tem-se:

$$\hat{X} = \begin{vmatrix} Q_1 & Q_{13} & Q_{25} & \cdots & Q_{1+12(n-1)<nvaz} \\ Q_2 & Q_{14} & Q_{26} & \cdots & Q_{2+12(n-1)<nvaz} \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ Q_{119} & Q_{131} & Q_{143} & & Q_{119+12(n-1)<nvaz} \\ Q_{120} & Q_{132} & Q_{144} & \cdots & Q_{120+12(n-1)<nvaz} \end{vmatrix} \tag{3.27}$$

Sendo *nvaz* o número de dados disponíveis na série histórica.

Para $X_{i=1}$ monta-se o vetor *X* desde o primeiro valor de vazão observada até o último valor do mesmo mês de $X_{i=1}$, $X_{i=2}$ elabora-se o vetor *X* a partir do

segundo valor observado até o último valor do mesmo mês de $X_{i=2}$ da série observada, e assim sucessivamente.

No vetor $X_{i,j}$ é selecionada sempre a vazão do mesmo mês do coeficiente *i*. Quando o vetor $X_{i,j}$ é montado, calcula-se a média, o desvio padrão e o coeficiente de assimetria com base na distribuição log-normal de três parâmetros (Maidment, 1992), ilustrada no apêndice *A.1*.

Os valores de vazão são ordenados de forma crescente $[Q_1 < Q_2 < \ldots < Q_n]$, em seguida é calculada a mediana $\tilde{x}$, sendo a média, o desvio padrão e o coeficiente de assimetria dado respectivamente por:

$$med = \sum_{i=1}^{J}\left[\ln\left(Q_i - \frac{x_1 \times x_n - \tilde{x}^2}{x_1 + x_n - \tilde{x}^2}\right)\right] \tag{3.28}$$

$$dpd = \sum_{i=1}^{J}\left[\ln\left(Q_i - \frac{x_1 \times x_n - \tilde{x}^2}{x_1 + x_n - \tilde{x}^2}\right)\right]^2 \tag{3.29}$$

$$a = \frac{x_1 \times x_n - \tilde{x}^2}{x_1 + x_n - \tilde{x}^2} \tag{3.30}$$

Com *med, dpd* e *a* é montado a matriz de correlação:

$$dt = \begin{bmatrix} 1 & r_{1,2} & r_{1,3} & \cdots r_{1,n} \\ r_{2,1} & 1 & r_{2,3} & \cdots r_{2,n} \\ \vdots & \ddots & \ddots & \ddots \vdots \\ r_{n,1} & r_{n,2} & r_{n,3} & \cdots\ 1 \end{bmatrix} \tag{3.31}$$

Com $1 \leq n \leq 120$

Sendo:

$$r_{i,j} = \frac{\left(\dfrac{\sum_{n=1}^{k}\ln(Q_{i1} - a_i) \times \ln(Q_{j1} - a_j)}{n} - med_i \times med_j\right)}{dpd_i \times dpd_j} \tag{3.32}$$

Sendo:

$i1 = i + (n-1) \times 12$

$j1 = j + (n-1) \times 12$

$K =$ número de dados que satisfaz a equação $i1 \leq nvaz \; e \; j1 \leq nvaz$

$nvaz =$ número de dados da série histórica

$r_{1,2} =$ Coeficiente de correlação entre a vazão do primeiro mês e a vazão do segundo mês.

$r_{4,5} =$ Coeficiente de correlação entre a vazão do quarto mês com a vazão do quinto mês.

Generalizando $r_{i,j} =$ Coeficiente de correlação entre a vazão do mês *i* com a vazão do mês *j*.

Após calcular os valores de r (3.32), da matriz dt (3.31), média (med) (3.28), o coeficiente de assimetria (a) (3.30) e o desvio padrão (dpd) (3.29), são armazenados de forma binária, formando a base de dados dos coeficientes do modelo.

A previsão de vazões pelo método de interpolação ótima leva em consideração as últimas 96 vazões observadas, que serão interpoladas pelo modelo baseado na matriz de coeficientes do modelo.

A matriz dos coeficientes possui dimensão de 120×120, sendo que cada uma de suas colunas representa o coeficiente de correlação de determinado mês, como é ilustrado em (3.35).

Ao iniciar a previsão de vazões pelo modelo de interpolação ótima, o último mês de vazão observada é lido e utilizado para localizar quais dados da matriz de correlação (3.35) serão utilizados.

Para ilustrar como funciona a localização destes dados podemos elaborar uma matriz de duas linhas sendo a primeira linha referente ao primeiro mês da matriz (3.33) e a segundo representando sua localização na matriz (3.34).

$$\begin{vmatrix} 1 & r_{1,2} & \cdots & r_{1,12} & \cdots & r_{1,97} & r_{1,98} & \cdots & r_{1,108} & r_{1,109} & r_{1,110} & r_{1,111} & \cdots & r_{1,119} & r_{1,120} \\ 1 & 2 & \cdots & 12 & \cdots & 97 & 98 & \cdots & 108 & 109 & 110 & 111 & \cdots & 119 & 120 \end{vmatrix} \quad (3.33)$$

O localizador identifica o mês em que será feita a previsão e constrói a matriz de correlação que será usada na previsão a partir deste mês. O localizador é um vetor com doze pontos sendo que, cada mês possui um código no localizador como é ilustrado abaixo:

Jan	*Fev*	*Mar*	*Abr*	*Mai*	*Jun*	*Jul*	*Ago*	*Set*	*Out*	*Nov*	*Dez*
109	98	99	100	101	102	103	104	105	106	107	108

(3.34)

Com o código do localizador é construída uma matriz *96x96* com os coeficientes de correlação da matriz (3.31), sendo que o último ponto desta matriz corresponde ao código do localizador.

Supondo que o mês de previsão seja abril, seu código de localização será 100. Então a matriz de previsão terá como seu último ponto este valor.

A matriz abaixo ilustra quais dados serão utilizados neste caso, sendo que a matriz pintada é a que será utilizada como a matriz de previsão, tendo dimensão de *96x96*.

$$
\begin{vmatrix}
1 & r_{1,2} & r_{1,3} & r_{1,4} & r_{1,5} & r_{1,6} & r_{1,7} & \cdots & r_{1,97} & r_{1,98} & r_{1,99} & r_{1,100} & \cdots & r_{1,119} & r_{1,120} \\
r_{2,1} & 1 & r_{2,3} & r_{2,4} & r_{2,5} & r_{2,6} & r_{2,7} & \cdots & r_{2,97} & r_{2,98} & r_{2,99} & r_{2,100} & \cdots & r_{2,119} & r_{2,120} \\
r_{3,1} & r_{3,2} & 1 & r_{3,4} & r_{3,5} & r_{3,6} & r_{3,7} & \cdots & r_{3,97} & r_{3,98} & r_{3,99} & r_{3,100} & \cdots & r_{3,119} & r_{3,120} \\
r_{4,1} & r_{4,2} & r_{4,3} & 1 & r_{4,5} & r_{4,6} & r_{4,7} & \cdots & r_{4,97} & r_{4,98} & r_{4,99} & r_{4,100} & \cdots & r_{4,119} & r_{4,120} \\
r_{5,1} & r_{5,2} & r_{5,3} & r_{5,4} & 1 & r_{5,6} & r_{5,7} & \cdots & r_{5,97} & r_{5,98} & r_{5,99} & r_{5,100} & \cdots & r_{5,119} & r_{5,120} \\
\vdots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \vdots \\
r_{n-1,1} & r_{n-1,2} & r_{n-1,3} & r_{n-1,4} & r_{n-1,5} & r_{n-1,6} & r_{n-1,7} & \cdots & r_{n-1,97} & r_{n-1,98} & r_{n-1,99} & r_{n-1,100} & \cdots & r_{n-1,119} & r_{n-1,120} \\
r_{n,1} & r_{n,2} & r_{n,3} & r_{n,4} & r_{n,5} & r_{n,6} & r_{n,7} & \cdots & r_{n,97} & r_{n,98} & r_{n,99} & r_{n,100} & \cdots & r_{n,119} & r_{n,120} \\
\vdots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \vdots \\
r_{98,1} & r_{98,2} & r_{98,3} & r_{98,4} & r_{98,5} & r_{98,6} & r_{98,7} & \cdots & r_{98,97} & 1 & r_{98,99} & r_{98,100} & \cdots & r_{98,119} & r_{98,120} \\
r_{99,1} & r_{99,2} & r_{99,3} & r_{99,4} & r_{99,5} & r_{99,6} & r_{99,7} & \cdots & r_{99,97} & r_{99,98} & 1 & r_{99,100} & \cdots & r_{99,119} & r_{99,120} \\
r_{100,1} & r_{100,2} & r_{100,3} & r_{100,4} & r_{100,5} & r_{100,6} & r_{100,7} & \cdots & r_{100,97} & r_{100,98} & r_{100,99} & 1 & \cdots & r_{100,119} & r_{100,120} \\
\vdots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \iddots & \vdots \\
r_{119,1} & r_{119,2} & r_{119,3} & r_{119,4} & r_{119,5} & r_{119,6} & r_{119,7} & \cdots & r_{119,97} & r_{119,98} & r_{119,99} & r_{119,100} & \cdots & 1 & r_{119,120} \\
r_{120,1} & r_{120,2} & r_{120,3} & r_{120,4} & r_{120,5} & r_{120,6} & r_{120,7} & \cdots & r_{120,97} & r_{120,98} & r_{120,99} & r_{120,100} & \cdots & r_{120,119} & 1
\end{vmatrix}
\tag{3.35}
$$

Com esta matriz que será chamada de *matriz de previsão* iniciam-se os procedimentos para prever a vazão do referido mês.

Primeiramente é construída uma matriz auxiliar que será utilizada para calcular a inversa da matriz de previsão. Considerando $z_{1,1}$ como o primeiro valor da matriz de previsão, extraída da matriz de correlação (3.31) podemos ilustrar esta matriz como:

$$\begin{vmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,95} & z_{1,96} & 1 \\ z_{2,1} & z_{2,2} & \cdots & z_{2,95} & z_{2,96} & 1 \\ z_{3,1} & z_{3,2} & \cdots & z_{3,95} & z_{3,96} & 1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ z_{96,1} & z_{96,2} & \cdots & z_{96,95} & z_{96,96} & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{vmatrix} \times \begin{vmatrix} l_1 \\ l_2 \\ l_3 \\ \vdots \\ l_{96} \\ 1 \end{vmatrix} = \begin{vmatrix} z_{96,1} \\ z_{96,2} \\ z_{96,3} \\ \vdots \\ z_{96,96} \\ 1 \end{vmatrix} \tag{3.36}$$

Sendo que para o exemplo anterior, com o mês de previsão sendo abril podemos escrever alguns exemplos:

$$z_{1,1} = r_{5,5}$$

$$z_{1,2} = r_{5,6}$$

$$z_{1,96} = r_{5,100} \tag{3.37}$$

$$z_{95,96} = r_{99,100}$$

$$z_{96,96} = r_{100,100}$$

Após o cálculo dos coeficientes l_i fazemos:

$$z = \sum_{i=1}^{i=96} \frac{\ln(Q_i) - a_i - med_i}{dpd_i} \times l_i \tag{3.38}$$

Sendo *i=96* representa o último valor de vazão observada, neste exemplo seria do mês de março e *i=1* representa a vazão do mesmo mês (abril), porém referente a 8 anos atrás. Como exemplo se quer prever a vazão do mês de abril de 2004, o primeiro dado (*i=1*) representa os dados de abril de 1996 e o último dado (*i=96*) representa os dados do mês de maço de 2004.

Obtendo o valor de *z* temos:

$$Q_{i=0} = z \times \exp(dpd_0) + \exp(med_0 + a_0) \tag{3.39}$$

Sendo: Q_0 a vazão que será prevista, neste exemplo a de abril de 2004; med_0 a média do mês que será previsto, conforme (3.28); dpd_0 o desvio padrão do mês que será previsto, conforme (3.29); a_0 o coeficiente de assimetria do mês que será previsto, conforme (3.30) e *z* dado pela equação (3.38).

O código fonte desenvolvido para prever as vazões afluentes utilizando o modelo de interpolação ótima esta ilustrado no Apêndice F.3.

3.4 APLICAÇÃO DO MODELO MARS

Conforme descrito no item ***Erro! A origem da referência não foi encontrada.*** o modelo MARS foi adaptado de forma a permitir que suas funções fossem utilizadas para prever vazões mensais. Neste trabalho foram desenvolvidos o modelo MARS em 2 e 3 dimensões, ou seja, para o MARS bi-dimensional utilizasse a vazão Q_{t-1}, e para o MARS tri-dimensional utilizasse as vazões Q_{t-1} e Q_{t-2}.

3.4.1 Aplicação do modelo MARS 2d

O modelo MARS 2d utiliza somente a vazão do mês anterior para realizar a previsão.

Ao iniciar a calibração do modelo (obtenção dos parâmetros) o modelo MARS inicia lendo toda a série histórica, desta maneira é construído um vetor com as vazões observadas de forma que:

$$\vec{Q} = [Q_1, Q_2, \ldots, Q_n] \tag{3.40}$$

Sendo Q_1 o primeiro dado de vazão observada (mês *t-1*) e Q_n o último valor observado. Elaborando este vetor são selecionadas as vazões observadas relativas ao mês *t* e *t-1*, sendo *t* o mês em que a vazão será prevista e *t-1* o mês que será utilizado para prever esta vazão, sendo que é montado um vetor X_{t-1}:

$$X_{i=t-1,\,j} = [X_1, X_2, \ldots, X_n] \tag{3.41}$$

Com: $X_1 = Q_1$, $X_2 = Q_{13}$, $X_3 = Q_{25}$, generalizando tem-se: $X_{i=1,\,j}^{\,i=120} = Q_{i+12(j-1)}$ com *j* representando os n_j anos existentes e *i* a localização do primeiro valor de vazão a ser utilizada (vazão *t-1*).

A série histórica de vazões para o mês *t* e *t-1* são organizadas de forma crescente, sendo que a vazão do mês *t* acompanha a posição do mês *t-1*. Na

Tabela 3.1 apresenta-se um exemplo:

Tabela 3.1 Série de vazões não organizadas

Q_{t-1} (m³/s)	Q_t (m³/s)
146	107
287	305
198	253
412	512
304	290

Organizando a Tabela 3.1 como descrito anteriormente prepara-se a Tabela 3.2:

Tabela 3.2 Série de vazões organizada pelo modelo MARS 2d

Q_{t-1} (m³/s)	Q_t (m³/s)
146	107
198	253
287	305
304	290
412	512

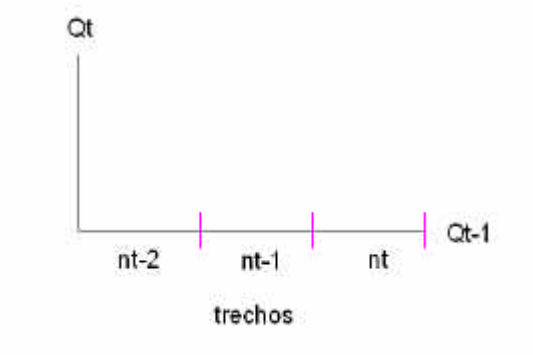
Define-se um vetor que se chamará Z_{t-1} para a vazão no mês *t-1* e outro vetor chamado Z_t para a vazão no mês *t*. Sendo que para o vetor Z_{t-1} tem-se:

$$Z_{t-1} = (Q_1 < Q_2 < \ldots < Q_{p-1} < Q_{nprec}) \quad (3.42)$$

Sendo: Q_1 a menor vazão observada para o mês *t-1* e Q_{nprec} a maior vazão observada para o mês *t-1*. Já o vetor Z_t tem suas vazões organizadas segundo a posição de suas respectivas vazões no mês *t-1*.

Após organizada a série de vazões para o mês t e *t-1* o modelo MARS define o número de trechos em que será dividida a série de vazões (neste trabalho foram analisados o modelo MARS utilizando 3 e 4 trechos). Além de determinar o número de trechos da série de vazões é definido o número mínimo de pontos por trecho, ou seja, a quantidade mínima de valores de vazões para cada trecho. Sendo que no presente trabalho foram determinados como número mínimo de pontos para o MARS com 3 trechos 10 pontos e para o MARS com 4 trechos foram determinados 7 pontos como o número mínimo.

FIGURA 3.1 Divisão dos trechos para o modelo MARS.



Determinado o número de trechos e o número mínimo de pontos por trecho, é elaborada uma divisão para determinar inicialmente quantos pontos cada trecho terá:

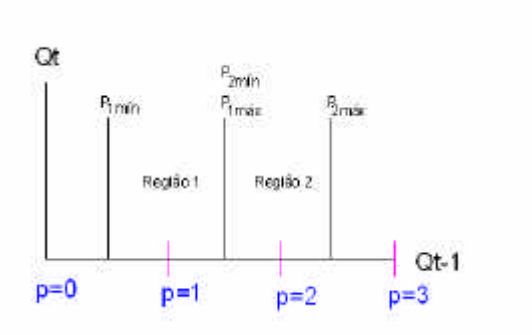
$$k = (N_q - 2p)\ div\ (2(nt-2)) \tag{3.43}$$

Sendo N_q o número de vazões existentes para o mês *t-1*, *p* o número mínimo de pontos, *nt* o número de trechos.

Definido o número de pontos por trecho é elaborada a divisão conforme ilustra a FIGURA 3.1. Esta é a pré-divisão do modelo MARS, em seguida é construído os limites de busca onde realmente serão estabelecidos os pontos da divisão. Cada trecho poderá ter um número variado de pontos, porém com o número mínimo de pontos estabelecido. A *FIGURA 3.2* ilustra como os trechos são divididos, sendo que o *ponto 0* (*p=0*) e o *ponto 3* (*p=3*) são fixos, sendo a vazão mínima e a vazão máxima observada para o mês *t-1*, respectivamente.

Já o *ponto1 (p=1)* deve estar compreendido na região 1, e o *ponto2 (p=2)* deve estar compreendido na região 2.

FIGURA 3.2 Limites de busca do modelo MARS para a vazão no mês *t-1*.



De uma forma matemática os pontos ilustrados na *FIGURA 3.2* podem ser expressos como:

$$P_0 = Q_1$$

$$P_{1mín} = Q_p + aprox$$

$$P_{1máx} = Q_{p+k} - aprox$$

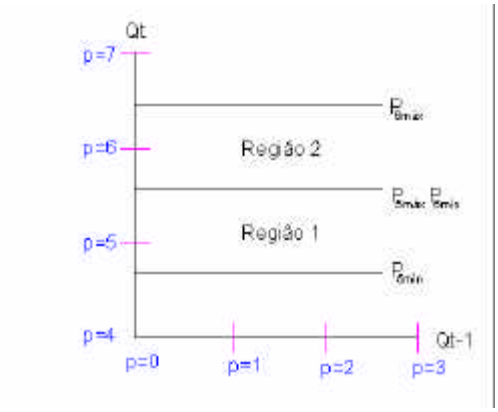
$$P_{2mín} = Q_{p+\{(2[nt-1]-1)k\}} + aprox \qquad (3.44)$$

$$P_{2máx} = Q_{nprec-p} - aprox$$

$$P_3 = Q_{nprec}$$

Determinado o intervalo de variação para a vazão no mês *t-1* conforme a equação (3.44) deve-se também determinar o intervalo de variação da vazão no mês *t*.

FIGURA 3.3 Limites de busca do modelo MARS para a vazão no mês *t*.



A *FIGURA 3.3* ilustra como os trechos são divididos, sendo que o *ponto 4* (*p=4*) e o *ponto 7* (*p=7*) são fixos, sendo a vazão mínima e a vazão máxima observada para o mês *t*, respectivamente. Já o *ponto 5 (p=5)* deve estar compreendido na região 1, e o *ponto6 (p=6)* deve estar compreendido na região 2.

De uma forma matemática os pontos ilustrados na *FIGURA 3.3* podem ser expressos como:

$$P_4 = Q_{1,t}$$

$$P_{5mín} = mín(Q_{1,t} \ldots Q_{p+k,t}) - aprox$$

$$P_{5máx} = máx(Q_{1,t} \ldots Q_{p+k,t}) + aprox$$

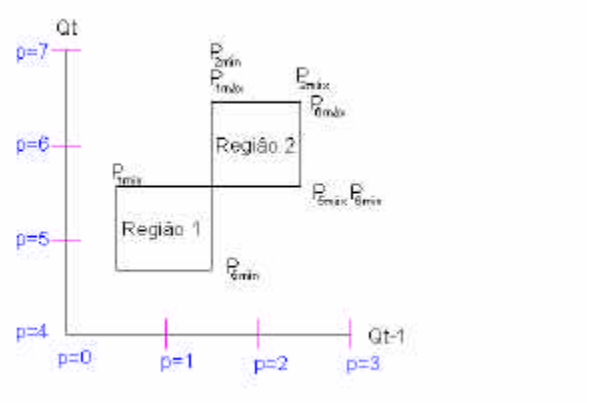
$$P_{6mín} = mín(Q_{nprec-p-k,t} \ldots Q_{nprec-p,t}) - aprox \qquad (3.45)$$

$$P_{6máx} = \max(Q_{nprec-p-k,t} \ldots Q_{nprec-p,t}) + aprox$$

$$P_7 = Q_{nprec,t}$$

Agrupando a *FIGURA 3.2* e a *FIGURA 3.3* tem-se a região em que os pontos de cada uma das retas do modelo MARS devem estar.

FIGURA 3.4 Limites de busca do modelo MARS para a vazão no mês *t e t-1*.



Após definido os extremos de busca para cada região é acionado um algoritmo genético (Kaviski, 2006) que é descrito no apêndice *Apêndice B Algoritmo Genético*. O algoritmo genético tem a finalidade de encontrar através da busca exaustiva o melhor par de pontos em cada região ilustrada na *FIGURA 3.4*. Ou seja, para a região 1 encontrasse um par de pontos formado pela vazão Q_t e Q_{t-1} , para a região 2 também encontrasse outro par de pontos. Em suma, o algoritmo genético efetua uma série de análises, em cada qual os pares de pontos são variados dentro da região, formando uma série de resultados. Para definir qual o melhor ponto (um para cada região) é feita uma análise baseada na soma dos erros ao quadrado, sendo que os pares de pontos que obtiver a menor somatória dos erros são eleitos para representar a previsão.

Como o modelo MARS é uma união de várias retas (*y=a+bx*), para efetuar a previsão é necessário entrar com a vazão do mês *t-1* para encontrar a vazão no mês *t*. Utilizando a *FIGURA 3.4* como referência, se a vazão no mês *t-1* estivar localizada entre os pontos *p=2* e *p=3*, por exemplo, a vazão no mês *t* será:

$$Q_t = \frac{P_7 - P_6}{P_3 - P_2}\left(Q_{t-1} - P_2\right) + P_6 \qquad (3.46)$$

Para analisar a soma dos erros ao quadrado é utilizada a seguinte equação:

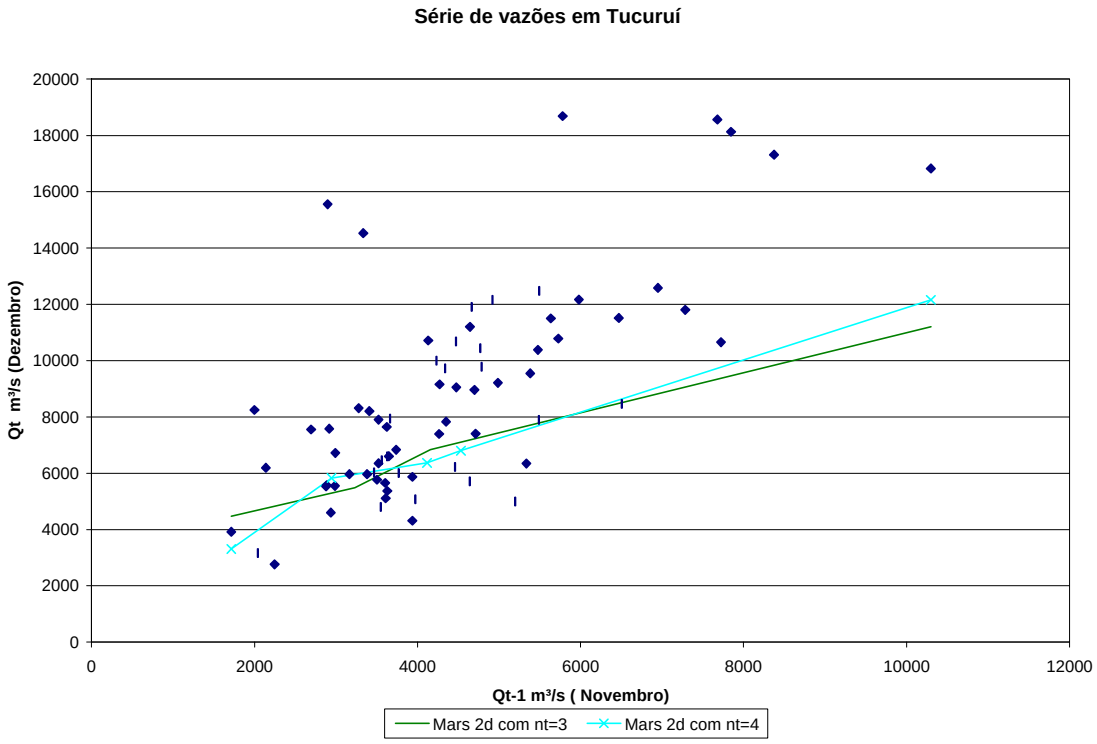
$$erro = \sum_{i=1}^{nprec} (Q_{obs} - Q_t)^2 \qquad (3.47)$$

Sendo: *nprec* o número de vazões existentes para o mês *t*, Q_{obs} valor da vazão observada para o mês *t*, Q_t o valor da vazão prevista no mês *t*.

Os pontos $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7$ que obtiverem o menor *erro* são os pontos de calibração e serão utilizados para fazer as previsões.

A *FIGURA* 3.5 ilustra a série histórica de vazões dos meses de novembro e dezembro da usina de Tucuruí, e as retas formadas pelo MARS 2d calculado para 3 e 4 trechos. Sabendo-se a vazão do mês *t-1* entra-se com este valor no eixo x e encontra-se seu correspondente no eixo *y*. Este é o valor da vazão prevista para o mês de dezembro baseada na vazão do mês de novembro, matematicamente se utiliza à equação *3.46*.

FIGURA 3.5 Equações formadas pelo MARS 2d para 3 e 4 trechos.



O código fonte desenvolvido para prever as vazões afluentes utilizando o modelo MARS 2d esta ilustrado no apêndice F.4 Modelo MARS 2d deste trabalho.

3.4.2 Aplicação do modelo MARS 3d

O modelo MARS 3d utiliza as vazões dos meses *t-1* e *t-2* para realizar a previsão.

Ao iniciar a calibração do modelo (obtenção dos parâmetros) o modelo MARS inicia lendo toda a série histórica, desta maneira é construído um vetor com as vazões observadas de forma que:

$$\vec{Q} = [Q_1, Q_2, \ldots, Q_n] \tag{3.48}$$

Sendo Q_1 o primeiro dado de vazão observada e Q_n o último valor observado. Elaborando este vetor são selecionadas as vazões observadas relativas ao mês *t*, *t-1*e *t-2*, sendo *t* o mês em que a vazão será prevista e *t-1* e *t*-2 os meses que serão utilizados para prever esta vazão, sendo que são montados dois vetores X_{t-1} e X_{t-2}:

$$X_{i=t-1,j} = [X_1, X_2, \ldots, X_n] \tag{3.49}$$

Com: $X_1 = Q_1$, $X_2 = Q_{13}$, $X_3 = Q_{25}$, generalizando tem-se: $X_{i=1,j}^{i=120} = Q_{i+12(j-1)}$ com *j* representando os n_j anos existentes e *i* a localização do primeiro valor de vazão a ser utilizada (vazão *t-1*).

Já para o mês *t-2* tem-se:

$$X_{i=t-2,j} = [X_1, X_2, \ldots, X_n] \tag{3.50}$$

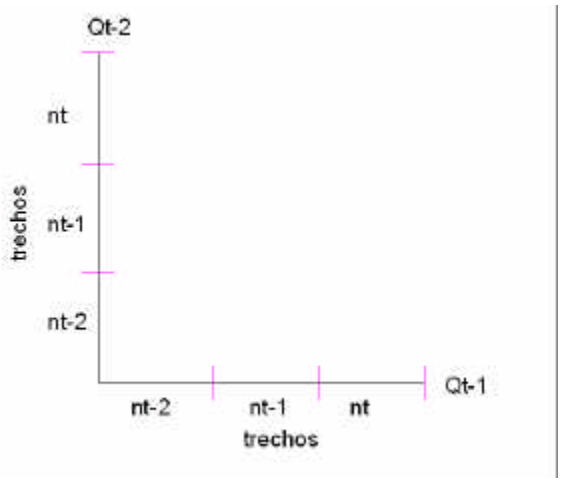
Com: $X_1 = Q_2$, $X_2 = Q_{14}$, $X_3 = Q_{26}$, generalizando tem-se: $X_{i=1,j}^{i=120} = Q_{i+12(j-1)}$ com *j* representando os n_j anos existentes e *i* a localização do primeiro valor de vazão a ser utilizada (vazão *t-2*).

A série histórica de vazões para o mês *t, t-1*e *t-2*, são organizadas de forma crescente, no MARS 3d ao contrário do MARS 2d a vazão do mês *t* não acompanha a posição do mês *t-1* e nem a posição do mês *t-2*.

Após organizada a série de vazões para o mês *t*, *t-1* e *t-2* o modelo MARS 3d define o número de trechos em que será dividida a série de vazões (neste trabalho foram analisados o modelo MARS utilizando 3 e 4 trechos). Além de determinar o número de trechos da série de vazões é definido o número mínimo de pontos por trecho, ou seja, a quantidade mínima de valores de vazões para cada trecho. Sendo que no presente trabalho foram determinados como número mínimo de pontos para o MARS 3d com 3 trechos 10 pontos e para o MARS com 4 trechos foram determinados 7 pontos como o número mínimo. Porém, este número mínimo de pontos se refere isoladamente ao eixo x e y, não sendo definido o número mínimo de pontos para cada quadrícula formada por esta divisão, por exemplo, com *nt=3* ter-se-á 9 quadrículas.

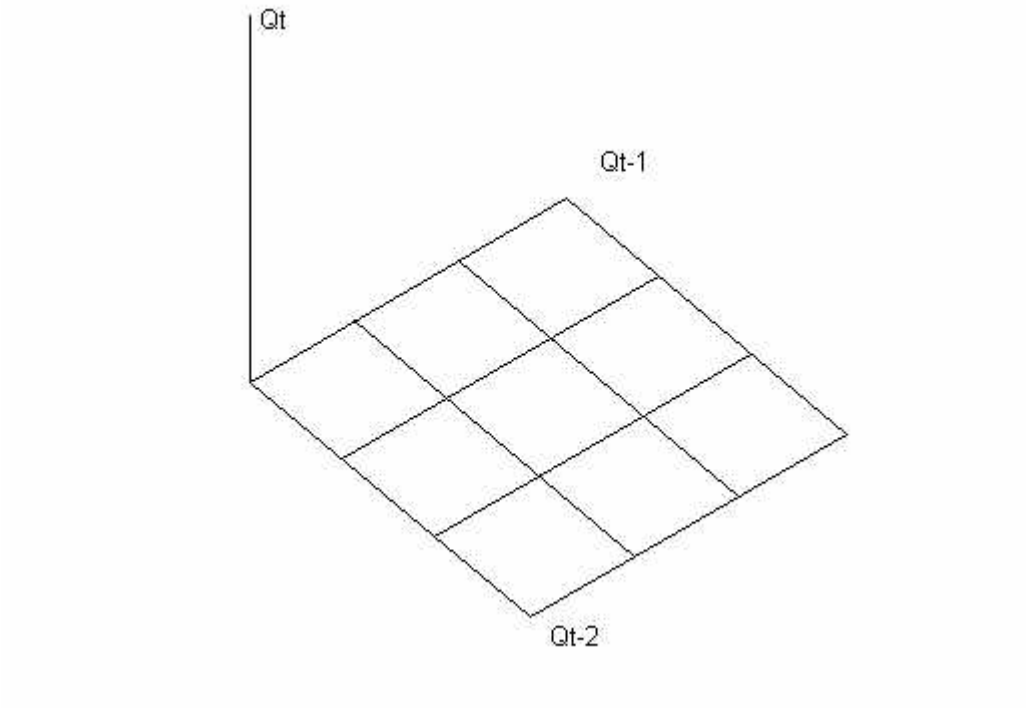
O modelo MARS 3d funciona como uma superfície, sendo que as vazões dos meses *t-1 e t-2* formam os pontos dos eixos *x e y* de um plano e a vazão do mês *t* formam os pontos do eixo *z* perfazendo desta maneira uma superfície.

FIGURA 3.6 Divisão dos trechos para o modelo MARS 3d.



A *FIGURA 3.6* ilustra como é realizada a divisão dos trechos para o modelo MARS 3d, sendo que o modelo MARS 3d trabalha de forma independente para o eixo *Qt-1* e *Qt-2*. Ou seja, após classificado de forma crescente as vazões dos meses *t-1* e *t-2* é construído os eixos *Qt-1 e Qt-2*.

FIGURA 3.7 Visão dos eixos x, y, z do MARS 3d.



A *FIGURA 3.7* ilustra como é construída a visão tri-dimensional do MARS 3d.

Determinado o número de trechos e o número mínimo de pontos por trecho, é elaborada uma divisão para determinar inicialmente quantos pontos cada trecho terá.

$$k = (N_q - 2p)\ div\ (2(nt - 2)) \tag{3.51}$$

Sendo N_q o número de vazões existentes para o mês *t-1 e t-2*; *p* o número mínimo de pontos, *nt* o número de trechos.

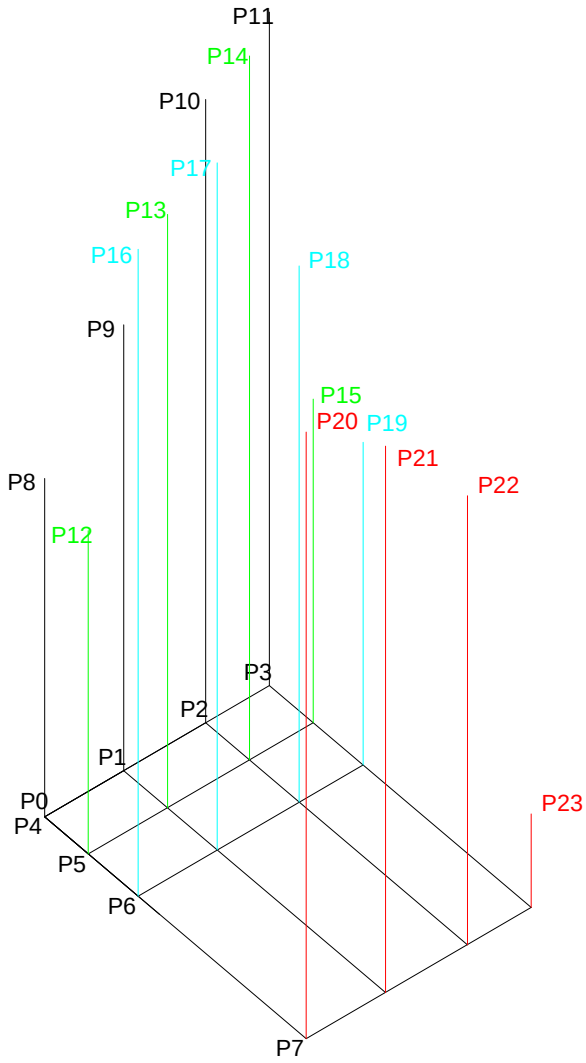
Em cada um dos nós ilustrados na *FIGURA 3.7* existe um eixo vertical (eixo z) que representa a vazão no mês *t*. O número total de pontos existentes no modelo MARS 3d com 3 trechos são oito pontos nos eixos *x* e *y* (sendo que os extremos não são considerados como pontos, já que são o ponto máximo e mínimo respectivamente) e mais 16 pontos no eixo *z*, perfazendo um total de 24 pontos para 3 divisões.

De uma forma genérica o número total de pontos é dado por:

$$nv = 2(nt+1)+(nt+1)^2 \qquad (3.52)$$

Sendo: *nv:* número total de pontos, *nt* número total de trechos.

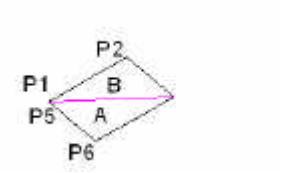
FIGURA 3.8 Visão dos pontos do MARS 3d para 3 trechos.



A *FIGURA 3.8* ilustra a visão tri-dimensional do MARS 3d, sendo que cada quadrícula no plano x, y representa um modelo de previsão. O intuído do MARS 3d é efetuar a previsão através de um plano. A princípio se for selecionado este plano tem-se uma superfície formada pelas vazões *Qt-1* e *Qt-2* e pela vazão *Qt.* O modelo MARS 3d trabalha com plano por triângulos. Ou seja, sabendo-se que a vazão *Qt-1* esta entre os pontos P_1 e P_2 e a vazão *Qt-2*

esta entre os pontos P_5 e P_6 , o modelo MARS 3d traça uma linha de forma a dividir este quadrado em dois triângulos, como mostra a *FIGURA 3.9*.

FIGURA 3.9 Divisão do plano quadrilátero em dois triângulos pelo MARS 3d.

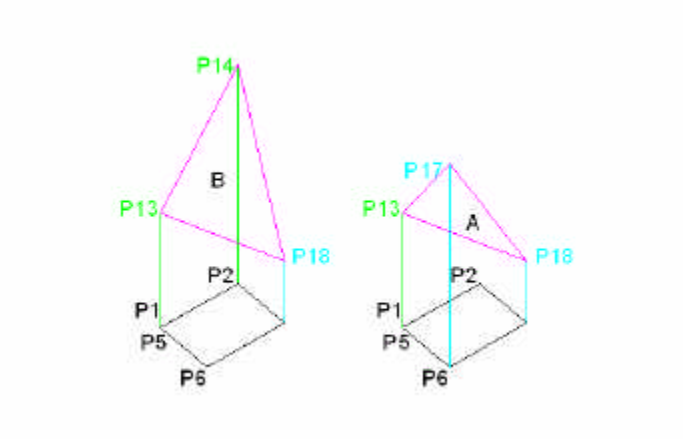


A divisão ilustrada na *FIGURA 3.9* tem como intuito formar um plano por triângulos com os pontos atribuídos a vazão *Qt*. Sendo que para saber em que parte do plano (*A* ou *B*) esta localizada a vazão *Qt-2* é necessário resolver a seguinte equação:

$$h = P_5 + \frac{P_6 - P_5}{P_2 - P_1}\left(Q_{t-1} - P_1\right) \tag{3.53}$$

Se *Qt-2* < h, então a vazão *Qt-2* esta localizada na parte *B* do plano ilustrado na *FIGURA 3.9*. Caso *Qt-2* >= h então *Qt-2* estará localizada na parte *A* do plano.

FIGURA 3.10 Plano por triângulos formado pelo MARS 3d.



A *FIGURA 3.10* ilustra como fica a divisão dos planos por triângulos. Definido o plano que será utilizado para fazer a previsão, é utilizada a equação do plano passando pelos pontos P_{13}, P_{14}, P_{18} para a parte *B* do plano e os pontos P_{13}, P_{17}, P_{18} para a parte *A* do plano. Considerando os ponto *1, 2,* 3 e suas coordenadas *x, y,* z tem-se:

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = 0 \tag{3.54}$$

Sabendo que *x* representa a vazão *Qt-1*, *y* representa a vazão *Qt-2* e *z* representa a vazão *Qt*, e que os índices *1, 2, 3* são parâmetros de calibração, tem-se que a incógnita é o ponto *z*.

$$z = z_1 + \frac{\left\{ \begin{matrix} (x - x_1)[(y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1)] + \\ (y - y_1)[(z_2 - z_1)(x_3 - x_1) - (z_3 - z_1)(x_2 - x_1)] \end{matrix} \right\}}{(x_3 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_3 - y_1)} \tag{3.55}$$

Sendo que a incógnita *z* é a vazão que será prevista.

Caso a vazão *t-1 ou t-2* seja menor ou maior do que a existente no plano, ou seja, esta vazão esta fora do greide mostrado na *FIGURA 3.8,* o modelo MARS irá procurar qual o plano mais próximo dos valores *Qt-1* e *Qt-2*. Para isto o modelo MARS procura em toda a malha qual a menor distância entre o baricentro dos planos por triângulos *x, y* e do ponto formado pela vazão observada *t-1* e *t-2*.

Para os planos por triângulos inferiores (ex.: P_{10}, P_{11}, P_{16}) a distância é dada por:

$$dis\tan cia = \sqrt{\left(Q_{obs,t-1} - (Q_3 - Q_2) \times \frac{2}{3} \right)^2 + \left(Q_{obs,t-2} - (Q_5 - Q_4) \times \frac{1}{3} \right)^2} \tag{3.56}$$

Para os planos por triângulos superiores (ex.: P_{10}, P_{15}, P_{16}) a distância é dada por:

$$dis\tan cia = \sqrt{\left(Q_{obs,t-1} - (Q_3 - Q_2) \times \frac{1}{3} \right)^2 + \left(Q_{obs,t-2} - (Q_5 - Q_4) \times \frac{2}{3} \right)^2} \tag{3.57}$$

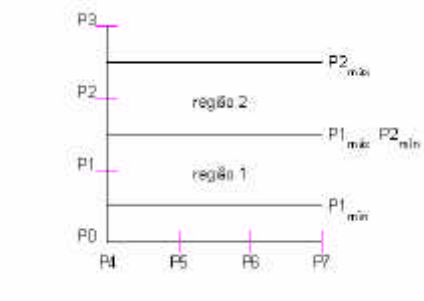
No caso do MARS com 3 trechos, possuindo 9 quadrículas e conseqüentemente 18 plano por triângulos, o modelo procura a menor distância entre todos estes planos, e escolhe os pontos pertencentes ao plano

com a menor distância, e para calcular a vazão prevista para o mês *t* é usada a equação 3.55.

Sabendo como o modelo MARS 3d faz a previsão e tendo em vista como é elaborada a divisão, inicia-se a entrada dos parâmetros para calibração, que segue a idéia do MARS 2d.

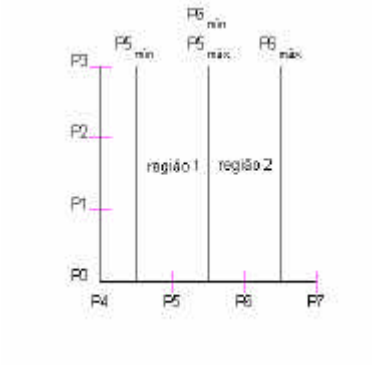
Primeiramente divide-se o eixo x (vazão *t*-1) em trechos conforme ilustrado na *FIGURA 3.6*, em seguida se estabelece de acordo com o número mínimo de pontos os limites de busca e ilustrado na *FIGURA 3.11*.

FIGURA 3.11 Limites de busca do eixo x (*t-1*) para o modelo MARS 3d.



Da mesma maneira, o eixo *y* (vazão *t-2*) é dividido por igual número de trechos que do eixo *x*, tem seus limites de busca estabelecidos conforme o número mínimo de pontos e é ilustrada na *FIGURA 3.12*.

FIGURA 3.12 Limites de busca do eixo y (*t-2*) para o modelo MARS 3d.



De uma forma matemática podemos definir os pontos do eixo *x* (vazão *t-1*) como:

$$P_0 = Q_{t-1,1}$$

$$P_{1mín} = Q_{t-1,p} + aprox$$

$$P_{1máx} = Q_{t-1,p+k} - aprox$$

$$P_{2mín} = Q_{t-1,,\,p+\{(2[nt-1]-1)k\}} + aprox \qquad (3.58)$$

$$P_{2máx} = Q_{t-1,nprec-p} - aprox$$

$$P_3 = Q_{t-1,nprec}$$

Para o eixo *y* (vazão *t-2*) pode-se definir os pontos de forma matemática como:

$$P_4 = Q_{t-2,1}$$

$$P_{5mín} = Q_{t-2,p} + aprox$$

$$P_{5máx} = Q_{t-2,p+k} - aprox$$

$$P_{6mín} = Q_{t-2,\,p+\{(2[nt-1]-1)k\}} + aprox \qquad (3.59)$$

$$P_{6máx} = Q_{t-2,nprec-p} - aprox$$

$$P_6 = Q_{t-2,nprec}$$

Para o eixo *z* (vazão *t*) classificamos as vazões em ordem crescente e definimos os limites dos pontos como:

$$P_{8mín} = P_{9mín} = P_{10mín} = \cdots = P_{23mín} = Q_{t,p} - aprox$$

$$P_{8máx} = P_{9máx} = P_{10máx} = \cdots = P_{23máx} = Q_{t,nprec-p} + aprox \qquad (3.60)$$

Definido os extremos de busca para cada região dos eixos *x* e *y,* e os limites de busca no eixo *z* é acionado o algoritmo genético descrito no *Apêndice B*. Este algoritmo tem a finalidade de encontrar através da busca exaustiva a melhor localização do grupo de pontos ($P_0 \ldots P_{23}$). Em suma, o algoritmo genético efetua uma série de análises, em cada qual o grupo de pontos dos eixos *x e y,* é variado dentro da região, e no eixo *z* os valores são variados dentro do intervalo especificado na *equação 3.60*, formando desta maneira uma série de resultados para cada iteração. Para definir qual o melhor grupo de pontos é feita uma análise baseada na soma dos erros ao quadrado, o grupo de pontos que obtiver a menor somatória dos erros são selecionados para representar a previsão.

Para analisar a soma dos erros ao quadrado é utilizada a seguinte equação:

$$erro = \sum_{i=1}^{nprec} (Q_{obs} - Q_t)^2 \qquad (3.61)$$

Sendo: *nprec* o número de vazões existentes para o mês *t*, Q_{obs} valor da vazão observada no mês *t*, Q_t o valor da vazão prevista no mês *t*.

Os pontos $P_0, P_1, \ldots, P_{22}, P_{23}$ que obtiverem o menor *erro* são os pontos de calibração e serão utilizados para fazer as previsões do modelo MARS 3d.

O código fonte desenvolvido para prever as vazões afluentes utilizando o modelo MARS 3d esta ilustrado no apêndice *F.5 Modelo MARS 3d* deste trabalho.

4 APLICAÇÕES

Para verificar a qualidade das previsões de vazões médias mensais para o horizonte de 1 a 12 meses utilizando os modelos descritos no capítulo 3, foram selecionados diversos locais de usinas, perfazendo 14 locais, geograficamente diferentes, com potência outorgada de 28.381,2 Mw, localizados nos rios Tocantins, São Francisco, Paraíba, Grande, Tietê, Paraná, Paranapanema, Iguaçu, Uruguai e Jacuí. Possuindo magnitudes de vazões e características hidrológicas diversas.

4.1 USINAS SELECIONADAS

Após o nome da usina é colocado entre parênteses o código desta usina na Aneel (Agência Nacional de Energia Elétrica), o rio em que esta usina esta localizada, os municípios, a potência outorgada e a área de drenagem.

I. A usina de Serra da Mesa (20920080) localizada no rio Tocantins, nos municípios de Cavalcante-GO e Minaçu-GO possui potência outorgada de 1275 Mw e área de drenagem de 50975 km^2.

II. A usina de Tucuruí (29680080) localizada no rio Tocantins, no município de Tucuruí-PA, possui potência outorgada de 8125 Mw e área de drenagem de 758000 km^2.

III. A usina de Três Marias (40990080) localizada no rio São Francisco, no município de Três Marias-MG, possui potência outorgada de 396 Mw e área de drenagem de 50600 km^2.

IV. A usina São Simão (60877080) localizada no rio Paraíba, nos municípios de Santa Vitória-MG e São Simão-GO, possui potência outorgada de 1710 Mw e área de drenagem de 83600 km^2.

V. A usina de Furnas (61661000) localizada no rio Grande, no município de Alpinópolis-MG, possui potência outorgada de 1216Mw e área de drenagem de 50464 km^2.

VI. A usina de Água Vermelha (61998080) localizada no rio Grande, nos municípios de Indiaporã-SP e Iturama-MG, possui potência outorgada de1396,2 Mw e área de drenagem de 139900 km^2.

VII. A usina de Três Irmãos (62900080) localizada no rio Tietê, no município de Pereira Parreto-SP, possui potência outorgada de 1292 Mw e área de drenagem de 71510 km^2.

VIII. A usina de Porto Primavera (63995080) localizada no rio Paraná, nos municípios de Anaurilândia-MS e Teodoro Sampaio-SP, possui potência outorgada de 1540 Mw e área de drenagem de 574000 km^2.

IX. A usina de Capivara (64516080) localizada no rio Paranapanema, nos municípios de Porecatu-PR e Taciba-SP, possui potência outorgada de 640 Mw e área de drenagem de 85000 km^2.

X. A usina de Itaipu (64918980) localizada no rio Paraná, no município de Foz do Iguaçu na parte brasileira, possui potência outorgada de 6300 Mw (parte brasileira) e potência total de 12600 Mw (incluindo a parte paraguaia) e área de drenagem de 822150 km^2.

XI. A usina de Foz do Areia (65774403) localizada no rio Iguaçu, no município de Pinhão-PR, possui potência outorgada de 1676 Mw e área de drenagem de 29800 km^2.

XII. A usina de Salto Caxias (65973500), localizada no rio Iguaçu, no município de Capitão Leônidas Marques-PR, possui potência outorgada de 1240 Mw e área de drenagem de 57970 km^2.

XIII. A usina de Ita (73200080), localizada no rio Uruguai, nos municípios de Aratiba-RS e Ita-SC, possui potência outorgada de 1450 Mw e área de drenagem de 44500 km^2.

XIV. A usina Dona Francisca (85398000) localizada no rio Jacuí, nos municípios de Agudo-RS e Nova Palma-RS, possui potência outorgada de 125Mw e área de drenagem de 13200 km^2.

A FIGURA 4.1 ilustra o mapa brasileiro com a localização das 14 usinas selecionadas.

FIGURA 4.1 Mapa brasileiro com a localização das usinas em análise.


N
Rio Amazonas
Atlântico
Trecho Norte
Rio São Francisco
Rio Tocantins
Atlântico
Trecho Leste
Rio Paranaíba
Rio Paraná
Rio Grande
Incremental
Rio Tietê
Rio Iguaçu
Rio Paranapanema
Rio Uruguai
Atlântico
Trecho Sudeste
1 U.H. DONA FRANCISCA
2 U.H. ITÁ
3 U.H. GBM
4 U.H. SALTO CAXIAS
5 U.H. CAPIVARA
6 U.H. TRÊS IRMÃOS
7 U.H. FURNAS
8 U.H. ÁGUA VERMELHA
9 U.H. SÃO SIMÃO
10 U.H. PORTO PRIMAVERA
11 U.H. ITAIPÚ
12 U.H. TRÊS MARIAS
13 U.H. SERRA DA MESA
14 U.H. TUCURUÍ

4.1.1 Vazões médias mensais aos locais das usinas

Será apresentado os dados estatísticos de todas as usinas utilizadas para analisar os resultados obtidos pelos modelos de previsão. Esta estatística será a MLT (média de longo termo), o desvio padrão, a vazão mínima e vazão máxima histórica, para cada mês, no período de 1931 a 2004.

As vazões médias mensais foram fornecidas pela GRHI (Gerência de Recursos Hídricos) da Copel Geração.

A MLT é dada por:

$$MLT = \frac{\sum_{i=1}^{n} Q_i}{n} \tag{4.1}$$

Onde: Q_i é a vazão no ano *i*; *n* é o número de anos.

O desvio padrão (*DP*) é dado por:

$$DP = \sqrt{\frac{\sum_{i=1}^{n} \left(Q_i - \overline{Q}\right)^2}{n-1}} \tag{4.2}$$

Onde: Q_i é a vazão no ano *i*; $\overline{Q}$ é a média da vazão; *n* é o número de anos.

A MLT, o desvio padrão, a mínima vazão registrada e a máxima vazão registrada em escala mensal, para cada uma das 14 usinas estão tabeladas no *Apêndice E*.

4.2 ANÁLISE DOS RESULTADOS

Para analisar a qualidade dos resultados obtidos através dos modelos de previsão, foram efetuados vários cenários de previsão, variando o período de calibração e o período de verificação.

As etapas de análise são duas: na primeira é determinado o período de calibração e na segunda é determinado o período de verificação, sendo que o primeiro ano de verificação se inicia 1 ou 5 anos após determinado o período de calibração. Este período para início da previsão (1 ou 5 anos) tem como intuito verificar a influência de ter-se uma matriz de correlação atualizada como base para a previsão.

Paralelamente foi construído um cenário onde para cada previsão a matriz de calibração era atualizada, preservando desta maneira a distância temporal entre o período de calibração e o período de verificação (a distância temporal permanecia constante entre 1 e 5 anos em todo o período de verificação, e não somente em relação ao primeiro ano de verificação, como no primeiro cenário), verificando desta forma a influência de ter-se a matriz de correlação atualizada durante todo o período de verificação, ou ter-se apenas uma matriz atualizada até determinado ano e se fazer todas as previsões a partir desta base.

O resultado obtido pela previsão foi comparado com os valores observados, sendo que para cada mês e em cada local de usina foi determinada uma média dos erros:

$$erro_i = \frac{\sum_{j=1}^{n} \frac{\left|\hat{Q}_{i,j} - Q_{i,j}\right|}{Q_{i,j}}}{n} \qquad (4.3)$$

Sendo: $\hat{Q}_{i,j}$ a vazão prevista no mês *i* no ano *j* e $Q_{i,j}$ a vazão observado no mês *i* no ano *j* e *n* o número de anos analisados.

O $erro_i$ é o erro médio obtido para cada mês e em cada local de usina, sendo que será apresentado a média do erro em todos os locais de usinas.

Com o intuito de facilitar a demonstração dos resultados, o subtítulo do período de calibração e dos anos de previsão (período de verificação) será descrito como:

a) Lag-0 de calibração: significa que para cada ano de previsão o primeiro ano de calibração não foi atualizado.

b) Lag-1 de calibração: significa que para cada ano de previsão o primeiro ano de calibração foi atualizado em um ano.

c) Lag-0 de verificação: significa que para cada ano de previsão o último ano de calibração não foi atualizado.

d) Lag-1 de verificação: significa que para cada ano de previsão o último ano de calibração foi atualizado em um ano.

Para facilitar a visualização do funcionamento do Lag de Calibração e do Lag de Verificação, foi construído um exemplo para o período de calibração 1931-1941 e período de verificação de 1942-1951 com distância temporal =1, sendo ilustrado na Figura 4.2 e na Figura 4.3 é ilustrada a distância temporal = 5.

Para efetuar estas análises foi desenvolvido um programa em Delphi 5, sendo que o código fonte esta ilustrado no Apêndice F. Este programa retorna um arquivo para cada uma das usinas analisadas, sendo que os gráficos ilustrados na seqüência foram elaborados em Excel.

Figura 4.2 Exemplo do funcionamento do Lag-0 e Lag-1 de Calibração e Lag-0 e Lag-1 de verificação para distância temporal entre o período de calibração e verificação = 1

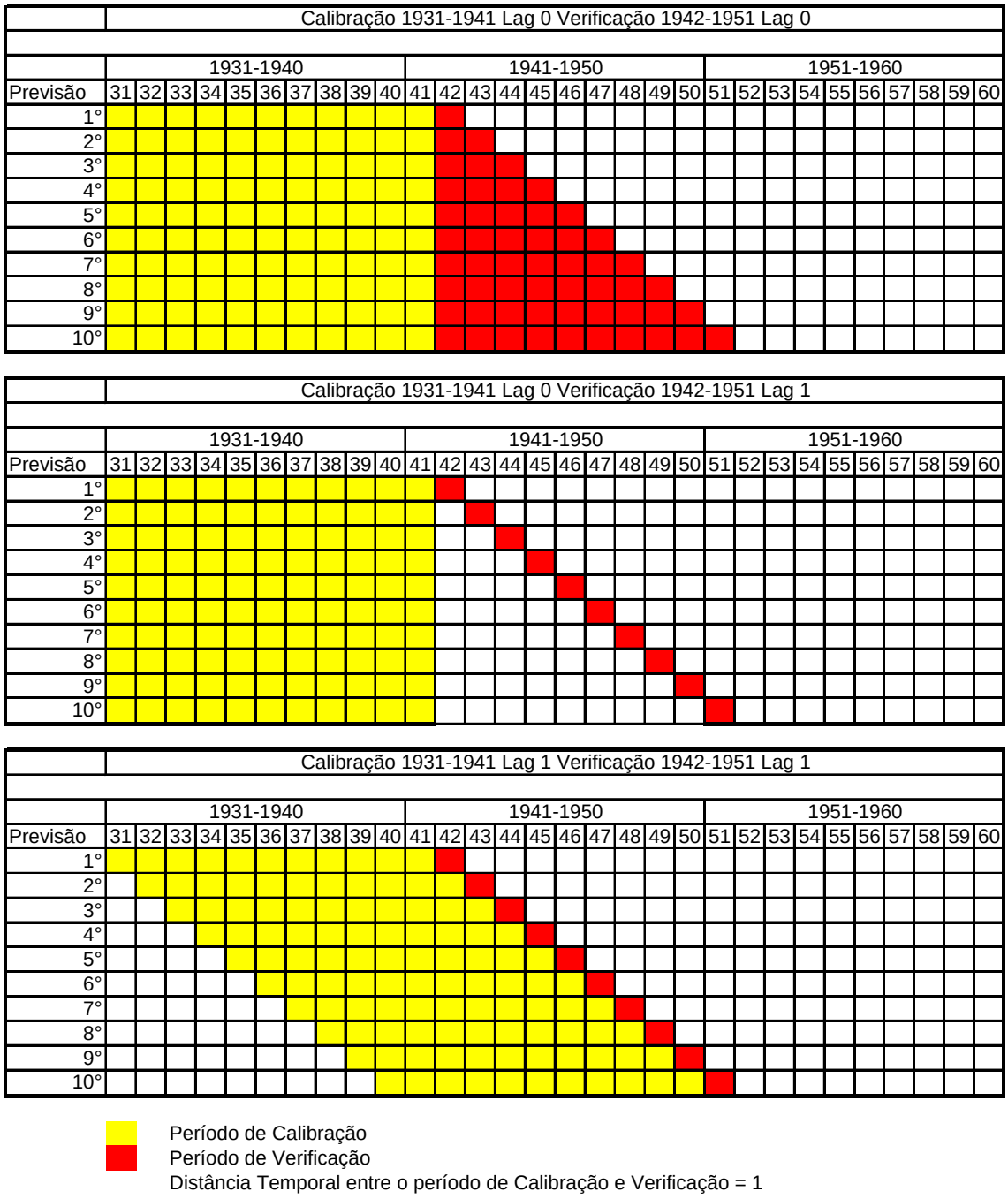
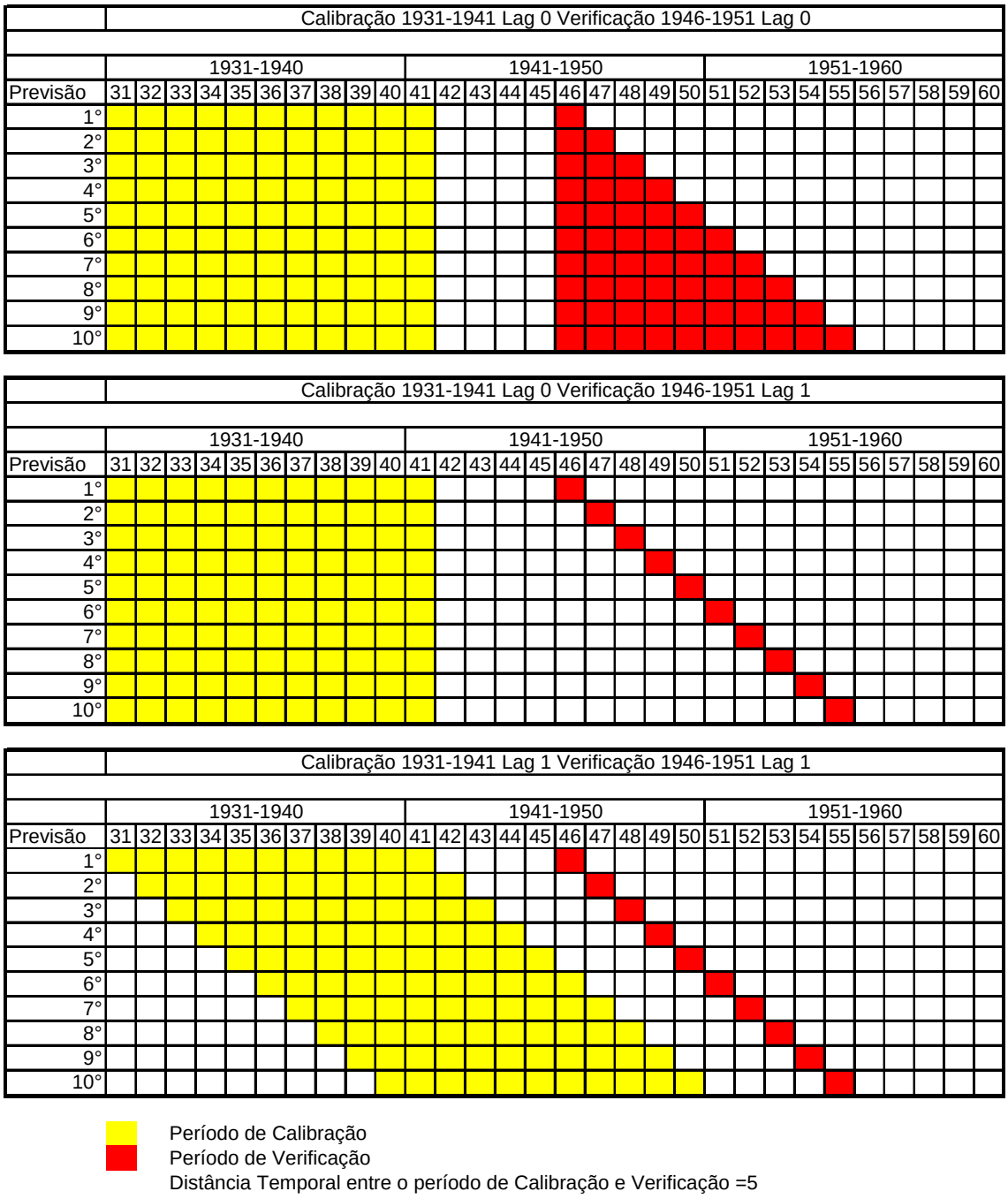


Figura 4.3 Exemplo do funcionamento do Lag-0 e Lag-1 de Calibração e Lag-0 e Lag-1 de verificação para distância temporal entre o período de calibração e verificação = 5

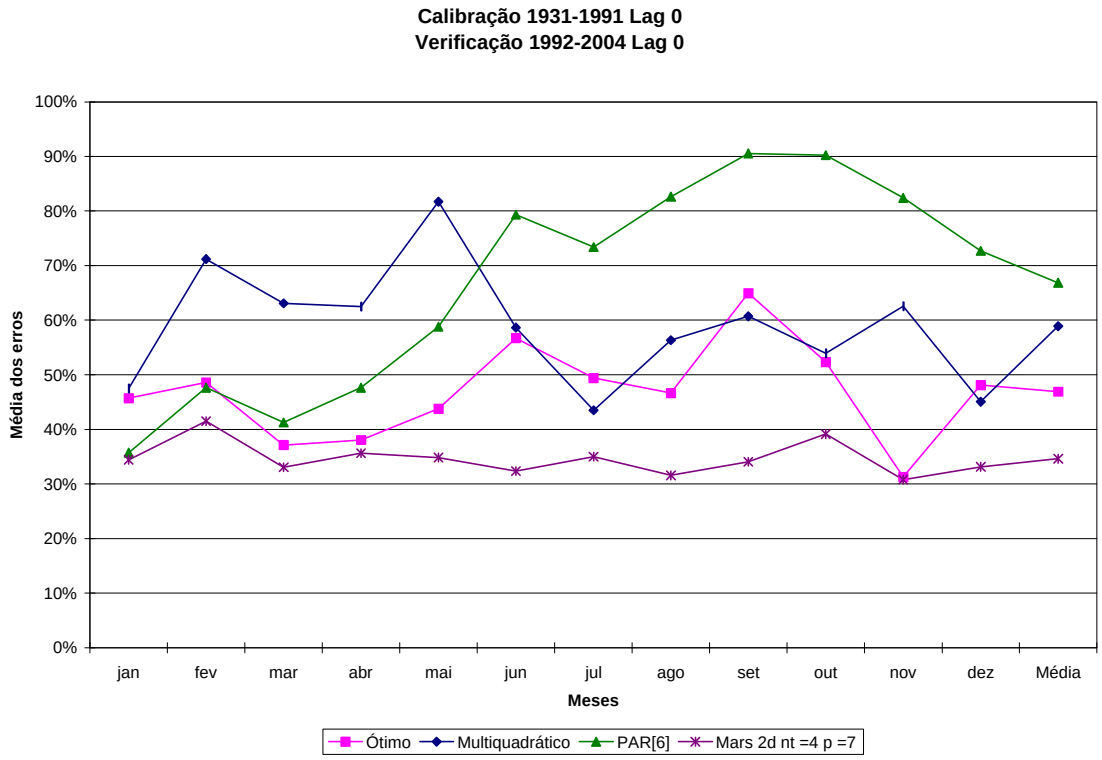


4.3 RESULTADOS OBTIDOS COM INTERVALO DE UM ANO

Este item tem como finalidade apresentar os resultados obtidos pelos modelos de previsão de vazões quando o intervalo entre o último ano de calibração e o primeiro ano de verificação é igual a 1. Nos gráficos ilustrados na seqüência é apresentado uma média do erro de todos os locais de usinas englobados, sendo que foi elaborado uma média para cada mês de previsão.

4.3.1 Resultados obtidos em Lag 0 de calibração e Lag 0 de verificação

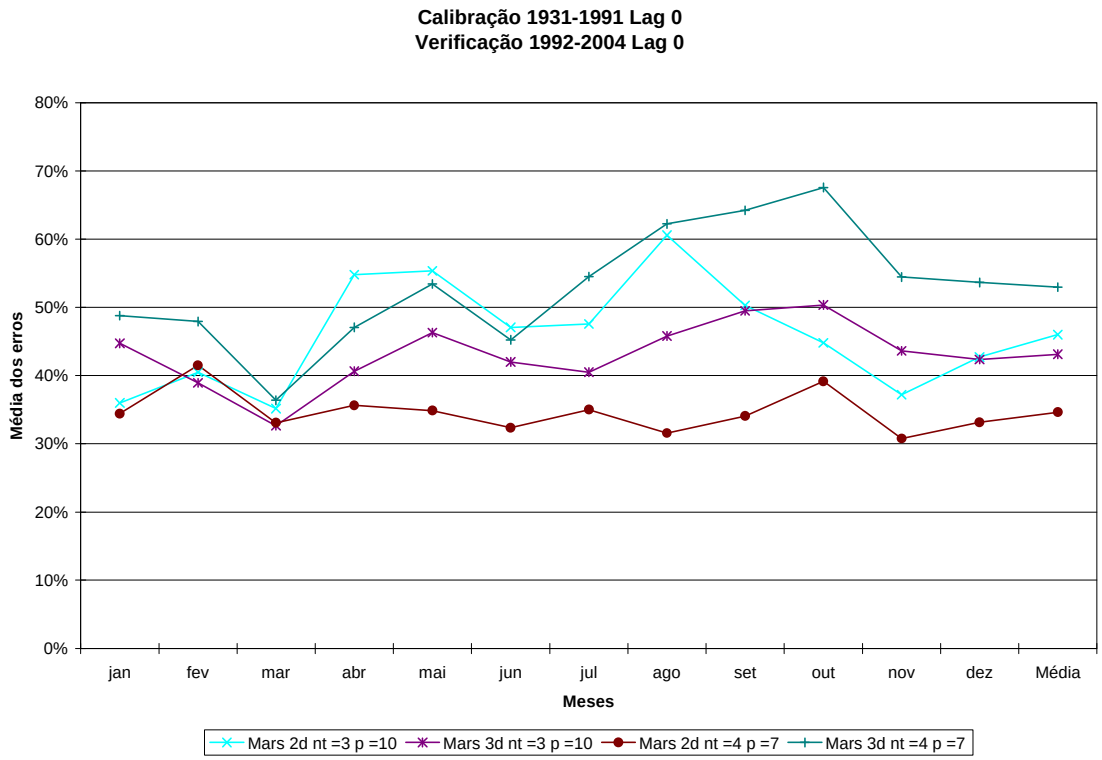
FIGURA 4.4 Distância temporal 1 Calibração 1931-1991 lag 0 Verificação 1992-2004 lag 0.



Como é possível visualizar na *FIGURA 4.4* o erro relativo entre a vazão prevista e a vazão observada para os diversos modelos, com Lag-0 de calibração e lag-0 de verificação, apresenta um comportamento relativamente parecido, porém, a magnitude dos erros é bem diferente, sendo que a modelo PAR[6] apresenta um erro relativo bem superior ao modelo ótimo e na maioria dos meses têm um erro relativo maior que o modelo multiquadrático. Já o modelo MARS 2d com 4 trechos apresentou o menor erro, sendo que seus resultados mostram-se relativamente constantes para todos os meses.

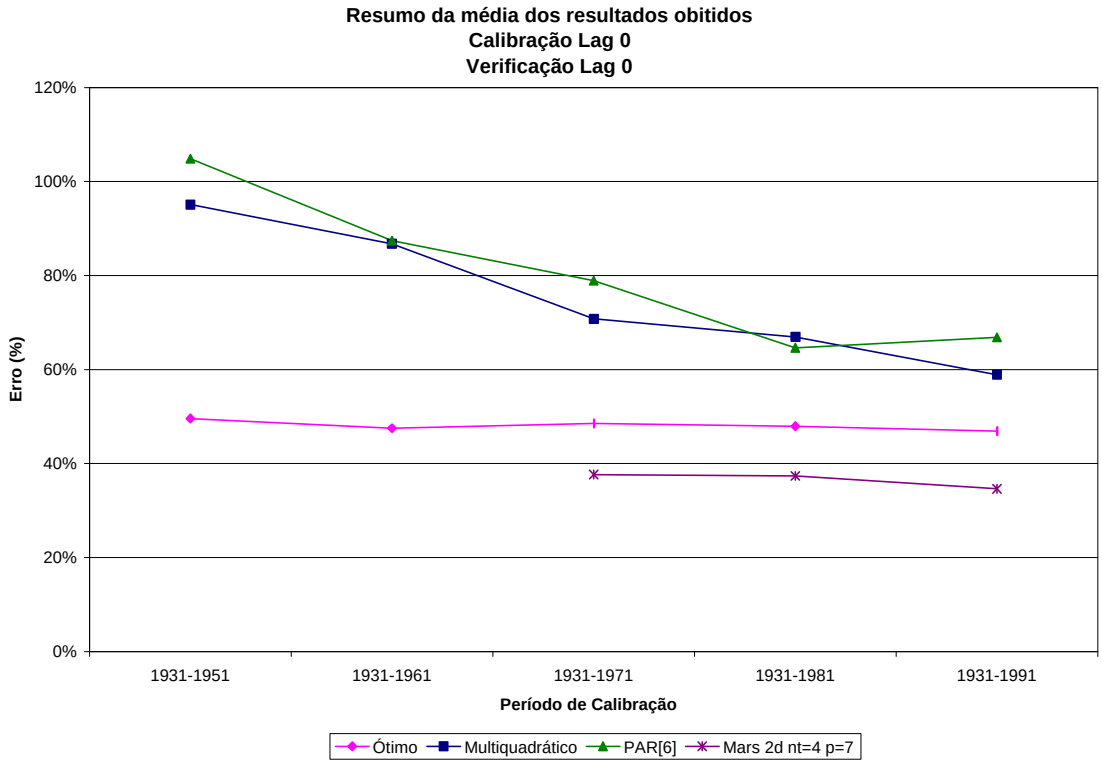
Nos meses de fevereiro a maio o modelo multiquadrático apresenta um erro relativo bem superior ao modelo ótimo e um pouco superior ao modelo PAR[6]. Nos meses de julho, setembro e dezembro o modelo multiquadrático obteve um erro menor que o modelo ótimo. A *FIGURA 4.4* apresenta o erro relativo para o período de calibração de 1931-1991 e período de verificação de 1992-2004, nas figuras *FIGURA* C.1 a *FIGURA C.6* na página 152 é apresentado a previsão para lag 0 de calibração e lag 0 de verificação para outros períodos (1931-1951, 1931-1961, 1931-1971, 1931-1981).

FIGURA 4.5 Distância temporal 1 Calibração 1931-1991 lag 0 Verificação 1992-2004 lag 0 utilizando o modelo MARS.



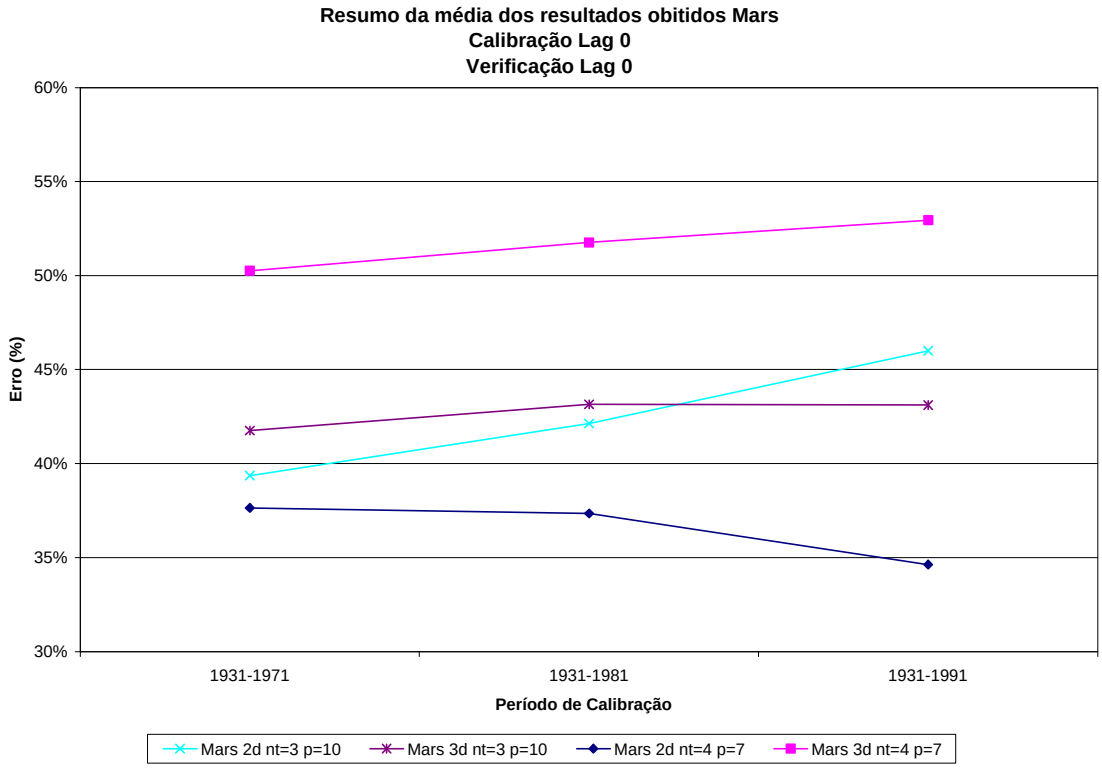
A *FIGURA 4.5* ilustra os erros das previsões utilizando o modelo MARS 2d e MARS 3d utilizando três e quatro trechos, com Lag-0 de calibração e lag-0 de verificação. O modelo MARS 2d com 4 trechos apresentou o melhor resultado para praticamente todos os meses, já utilizando 3 divisões o modelo MARS 3d apresentou na média o segundo melhor resultado. O MARS 3d com quatro divisões apresentou um resultado ruim, isto porque utilizando quatro divisões o número de pontos em cada quadrícula é muito pequeno, ocasionando um erro maior.

FIGURA 4.6 Resumo dos resultados distância temporal 1, Calibração lag 0 e Verificação lag 0.



A *FIGURA 4.6* apresenta a influência do período de calibração sobre o erro relativo para os quatro modelos de previsão. Com base nesta figura é possível concluir que o período de calibração tem uma forte influência sobre a qualidade da previsão para o modelo PAR[6] e multiquadrático, e pouca influência para o modelo ótimo. Já a qualidade da previsão para o modelo multiquadrático tem uma leve melhora para um período de calibração utilizando uma série superior a 40 anos, para períodos maiores a qualidade da previsão não tem uma melhora significativa. Para o modelo MARS 2d com quatro divisões o período de calibração maior melhorou um pouco os resultados.

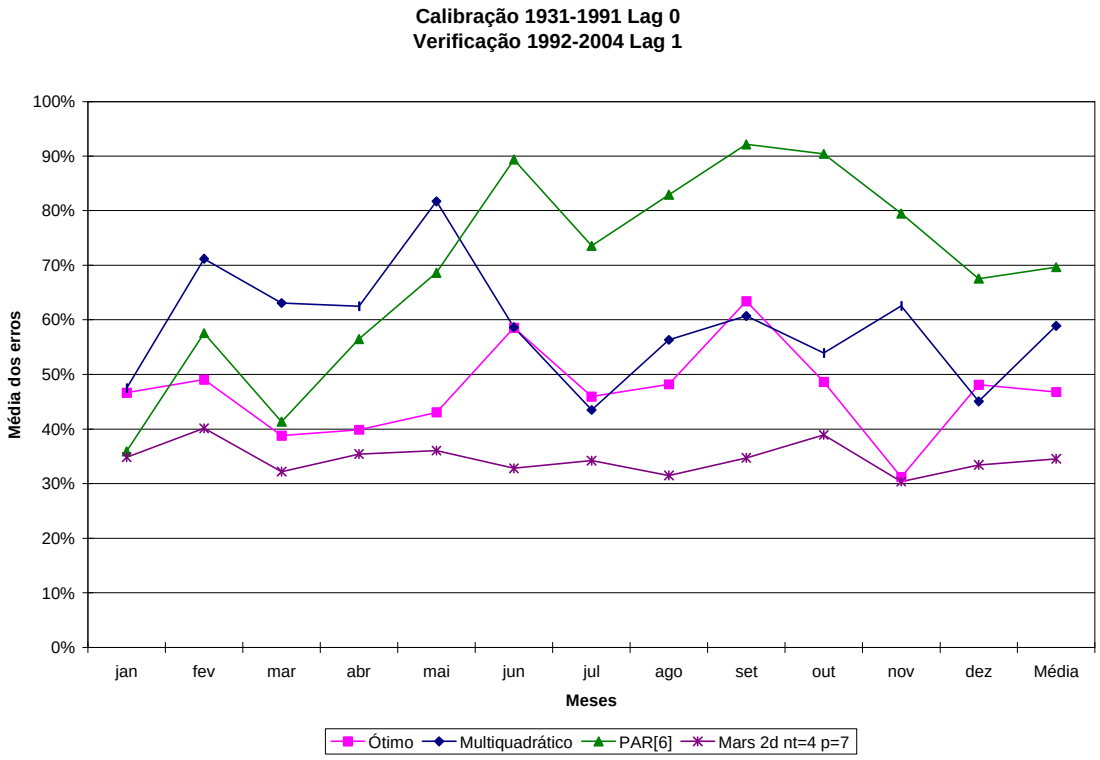
FIGURA 4.7 Resumo dos resultados distância temporal 1, Calibração lag 0 e Verificação lag 0 utilizando o modelo MARS.



A *FIGURA 4.7* apresenta a influência do período de calibração sobre o erro relativo para o modelo MARS 2d e 3d com 3 e 4 trechos. Para o MARS 2d com nt=4 (4 trechos) o aumento do período de calibração diminuiu o erro, já para os outros modelos MARS o aumento do período de calibração piorou a previsão.

4.3.2 Resultados obtidos em Lag 0 de calibração e Lag 1 de verificação

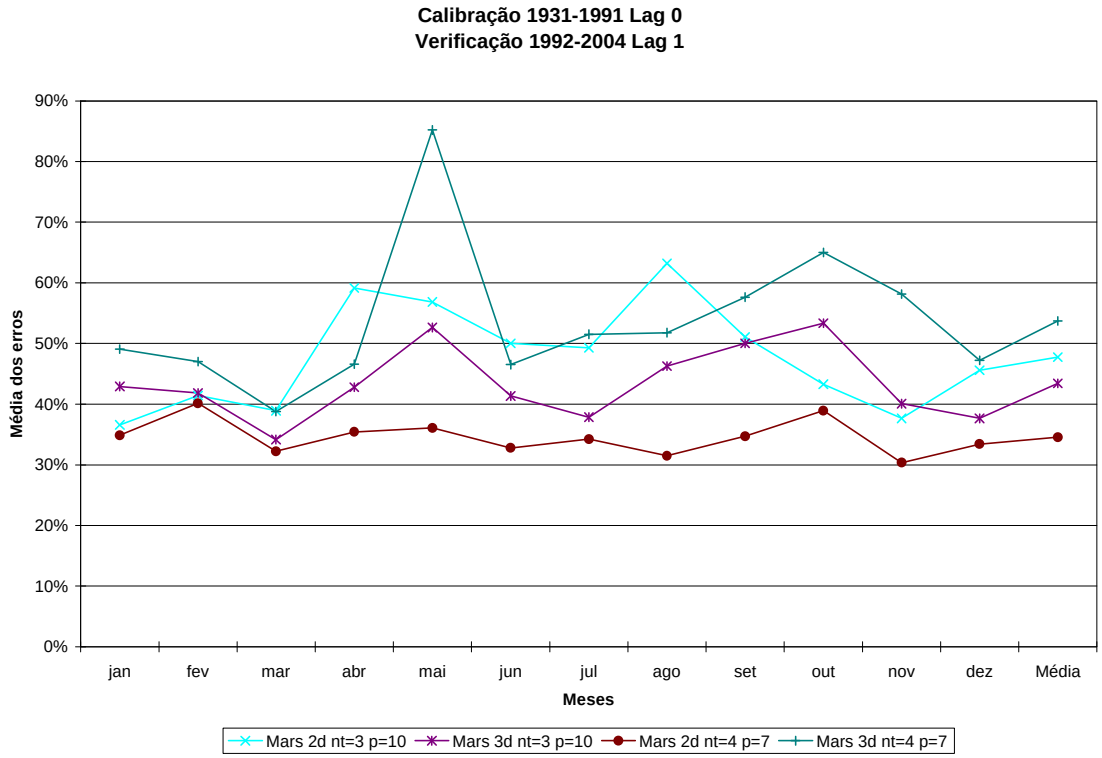
FIGURA 4.8 Distância temporal 1 Calibração 1931-1991 lag 0 Verificação 1992-2004 lag 1.



A *FIGURA 4.8* apresenta o erro relativo para o período de calibração de 1931-1991 e período de verificação de 1992-2004, na *FIGURA C.7* a *FIGURA* C.12 na página 156 é apresentado a previsão para lag 0 de calibração e lag 1 de verificação para outros períodos.

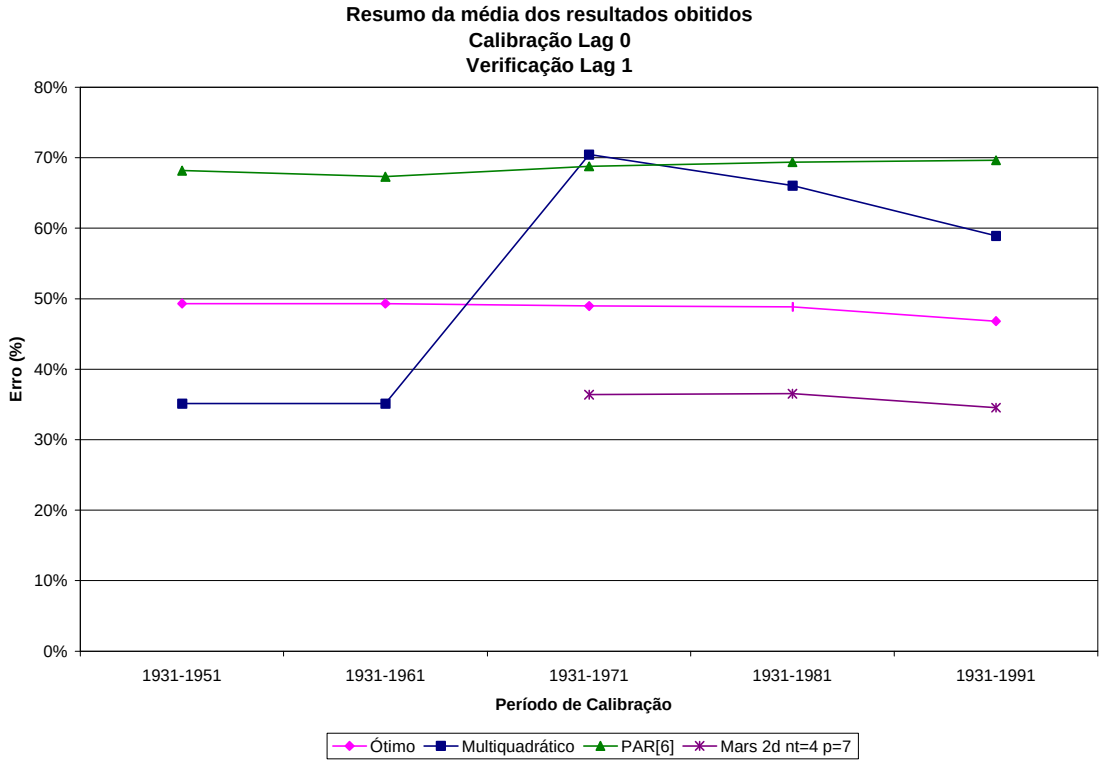
Como é possível visualizar na *FIGURA 4.8* o erro relativo entre a vazão prevista e a vazão observada para os quatro modelos de previsão, com Lag-0 de calibração e lag-1 de verificação, apresenta um comportamento relativamente parecido, porém, a magnitude dos erros é bem diferente, sendo que o modelo PAR[6] apresenta um erro relativo bem superior ao modelo ótimo e apenas nos meses de janeiro a maio têm um erro relativo menor que o modelo multiquadrático. Nos meses de fevereiro a maio o modelo multiquadrático apresenta um erro relativo bem superior ao modelo ótimo e um pouco superior ao modelo PAR[6]. Nos meses de julho e setembro o modelo multiquadrático obteve um erro menor que o modelo ótimo. Já o modelo MARS 2d com nt = 4 obteve em todos os meses um erro menor.

FIGURA 4.9 Distância temporal 1 Calibração 1931-1991 lag 0 Verificação 1992-2004 lag 1 para o modelo MARS.



A *FIGURA 4.9* ilustra os erros das previsões utilizando o modelo MARS 2d e MARS 3d com três e quatro trechos, com Lag-0 de calibração e lag-1 de verificação. O modelo MARS 2d com 4 trechos apresentou o melhor resultado para todos os meses, já utilizando 3 divisões o modelo MARS 3d apresentou na média o segundo melhor resultado. O MARS 3d com quatro divisões apresentou um resultado ruim, isto porque utilizando quatro divisões o número de pontos em cada quadrícula é muito pequeno, ocasionando um erro maior. Para o mês de maio o modelo MARS 3d com nt = 4 apresentou um resultado muito ruim quando comparado com os outros modelos MARS.

FIGURA 4.10 Resumo dos resultados distância temporal 1, Calibração lag 0 e Verificação lag 1



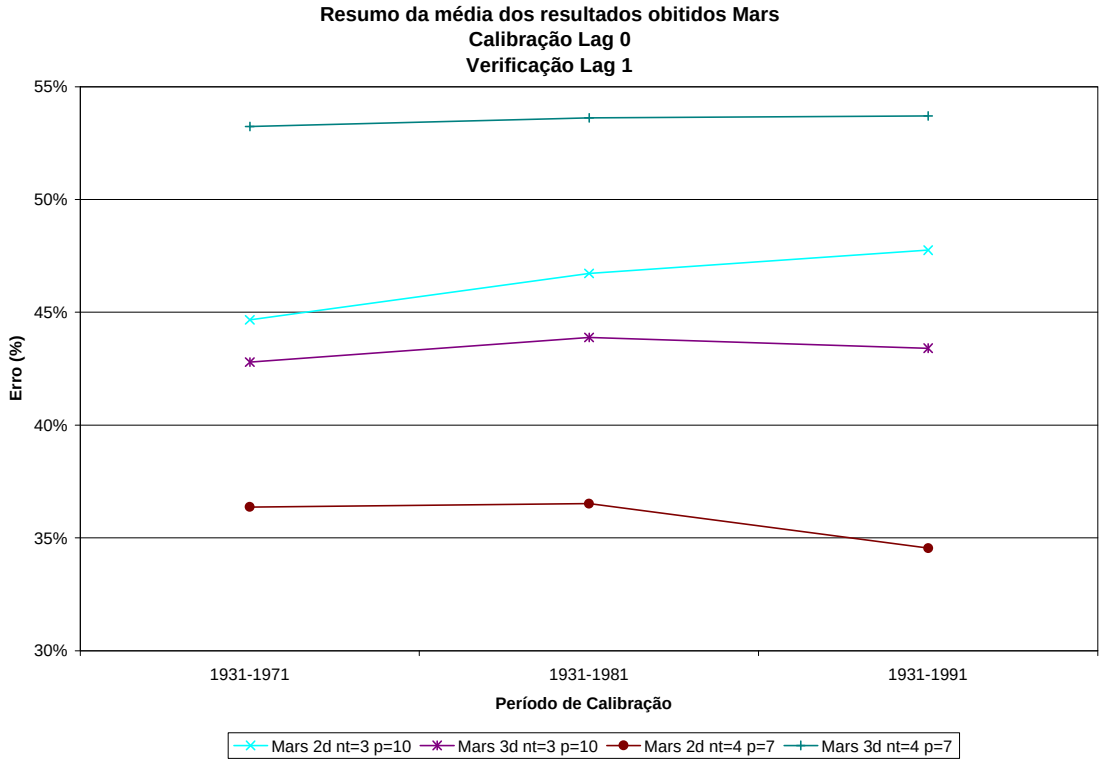
A *FIGURA 4.10* apresenta a influência do período de calibração sobre o erro relativo para os quatro modelos de previsão. Com base nesta figura é possível concluir que o período de calibração tem uma pequena influência sobre a qualidade da previsão para o modelo PAR[6], no período de calibração de 1931 até 1961 o aumento da série melhora o resultado obtido pelo modelo PAR[6], já para o período de 1931-1991 o modelo PAR[6] tem uma piora no resultado.

No modelo ótimo o comportamento do erro é praticamente igual para os diversos períodos de calibração, porém para o período de calibração de 1931-1991 o erro relativo teve uma sensível melhora, diminuindo em torno de 3 pontos percentuais o erro relativo.

Já a qualidade da previsão para o modelo multiquadrático tem uma leve melhora para um período de calibração utilizando uma série superior a 40 anos.

Para o modelo MARS 2d com nt=4 o aumento do período de calibração melhora o resultando, sendo que o MARS obteve o melhor resultado em todo o período analisado.

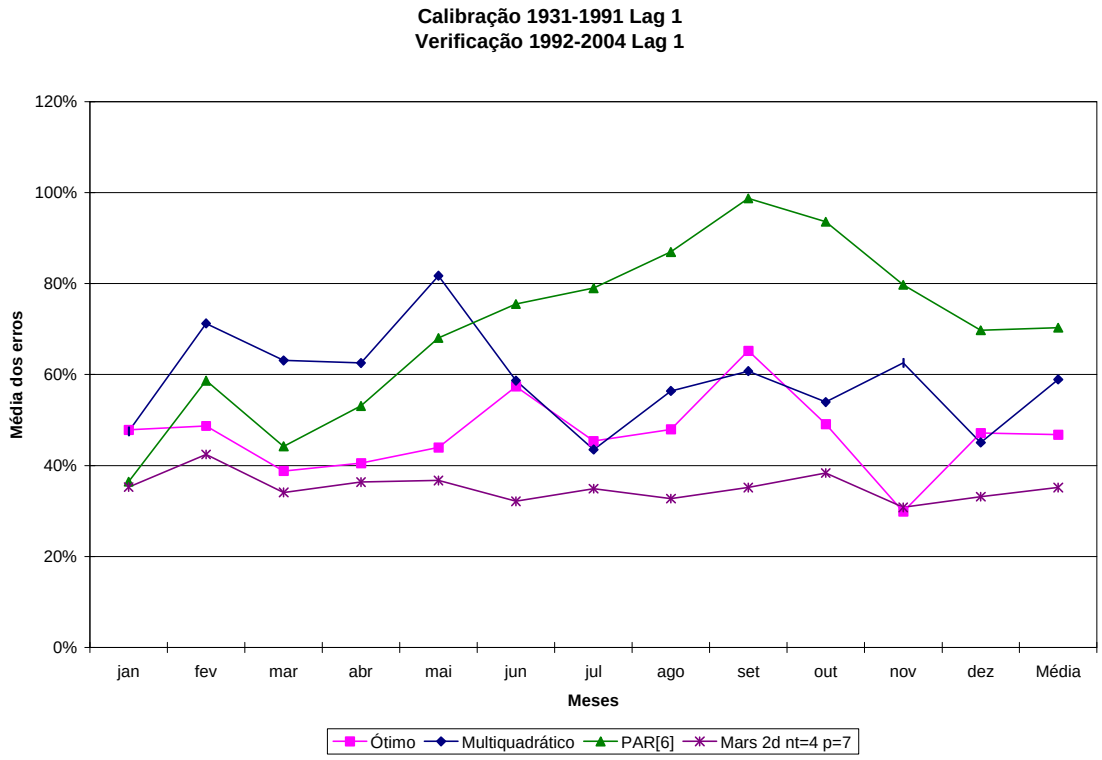
FIGURA 4.11 Resumo dos resultados distância temporal 1, Calibração lag 0 e Verificação lag 1 utilizando o modelo MARS.



A *FIGURA 4.11* apresenta a influência do período de calibração sobre o erro relativo para o modelo MARS 2d e 3d com 3 e 4 trechos. Para o MARS 2d com nt=4 (4 trechos) o aumento do período de calibração diminuiu o erro, já para o MARS 2d com nt = 3 o aumento do período de calibração piorou consideravelmente a previsão. Para o MARS 3d com nt = 3 e nt=4 o período de calibração não influenciou na qualidade da previsão de forma significativa.

4.3.3 Resultados obtidos em Lag 1 de calibração e Lag 1 de Verificação

FIGURA 4.12 Distância temporal 1 Calibração 1931-1991 lag 1 Verificação 1992-2004 lag 1.



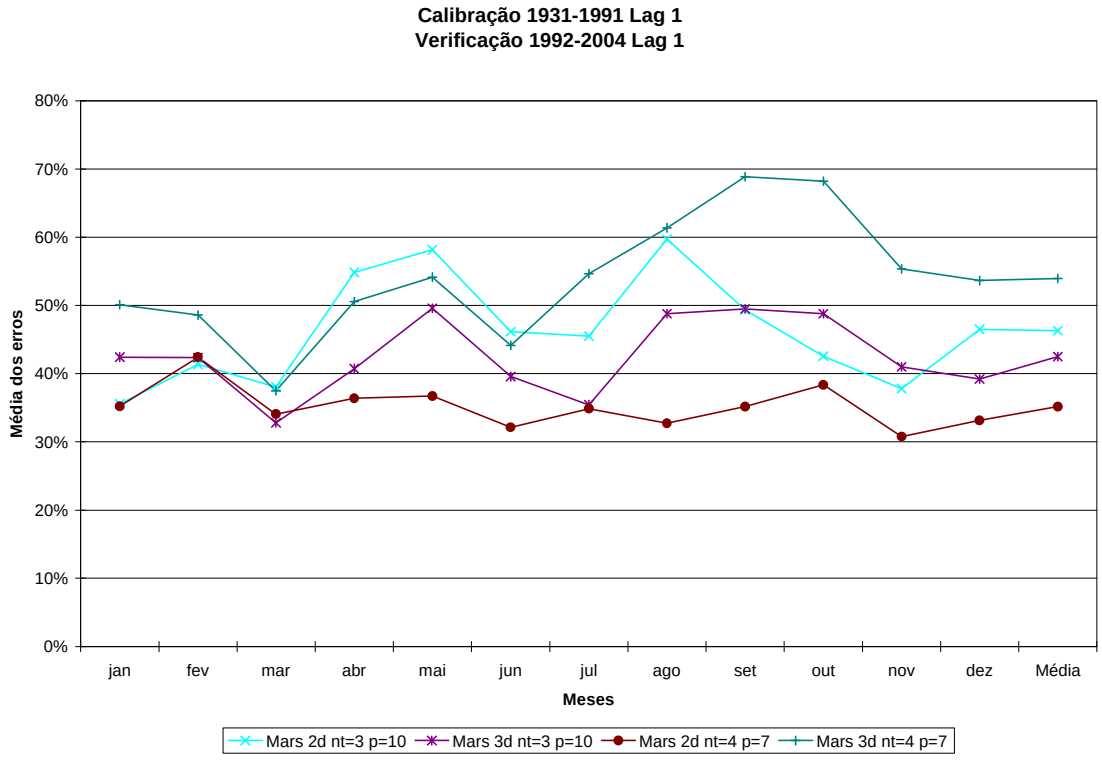
A *FIGURA 4.12* apresenta o erro relativo para o período de calibração de 1931-1991 e período de verificação de 1992-2004, na *FIGURA C.13* a *FIGURA C.18* na página 161 é apresentado a previsão para lag 1 de calibração e lag 1 de verificação para outros períodos.

Como é possível visualizar na *FIGURA 4.12* o erro relativo entre a vazão prevista e a vazão observada para os diversos modelos de previsão, com lag-1 de calibração e lag-1 de verificação, apresenta um comportamento relativamente parecido, porém, a magnitude dos erros é bem diferente, sendo que o modelo PAR[6] apresenta um erro relativo bem superior ao modelo ótimo e apenas nos meses de janeiro a maio têm um erro relativo inferior ao modelo multiquadrático.

Nos meses de fevereiro a maio o modelo multiquadrático apresenta um erro relativo bem superior ao modelo ótimo. Já nos meses de julho, setembro e dezembro o modelo multiquadrático obteve um erro menor que o modelo ótimo.

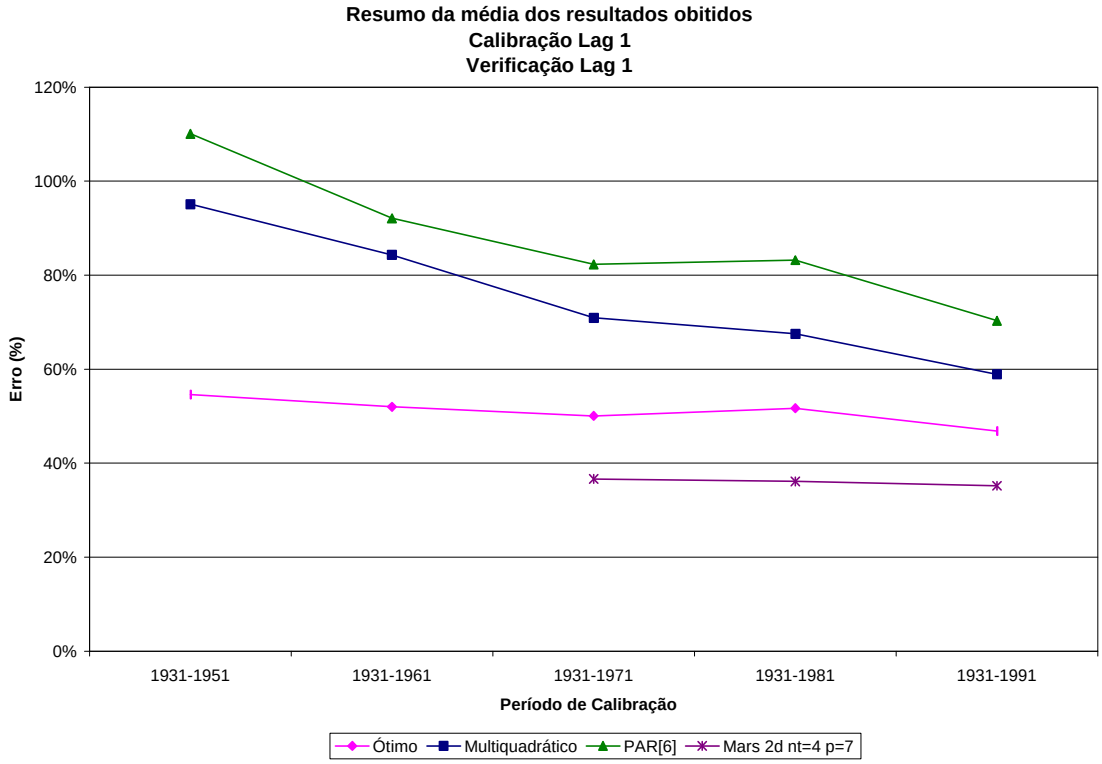
O modelo MARS 2d com nt=4 apresentou na média o melhor resultado, sendo superado pelo modelo ótimo apenas no mês de novembro e em todos os outros 11 meses apresentou um resultado bem superior.

FIGURA 4.13 Distância temporal 1 Calibração 1931-1991 lag 1 Verificação 1992-2004 lag 1 utilizando o modelo MARS.



A *FIGURA 4.13* ilustra os erros das previsões utilizando o modelo MARS 2d e MARS 3d com três e quatro trechos, com lag-1 de calibração e lag-1 de verificação. O modelo MARS 2d com 4 trechos apresentou o melhor resultado para todos os meses exceto fevereiro e março, já utilizando 3 divisões o modelo MARS 3d apresentou na média o segundo melhor resultado. O MARS 3d com quatro divisões apresentou um resultado ruim, isto porque utilizando quatro divisões o número de pontos em cada quadrícula é muito pequeno, ocasionando um erro maior.

FIGURA 4.14 Resumo dos resultados distância temporal 1, Calibração lag 1 e Verificação lag 1

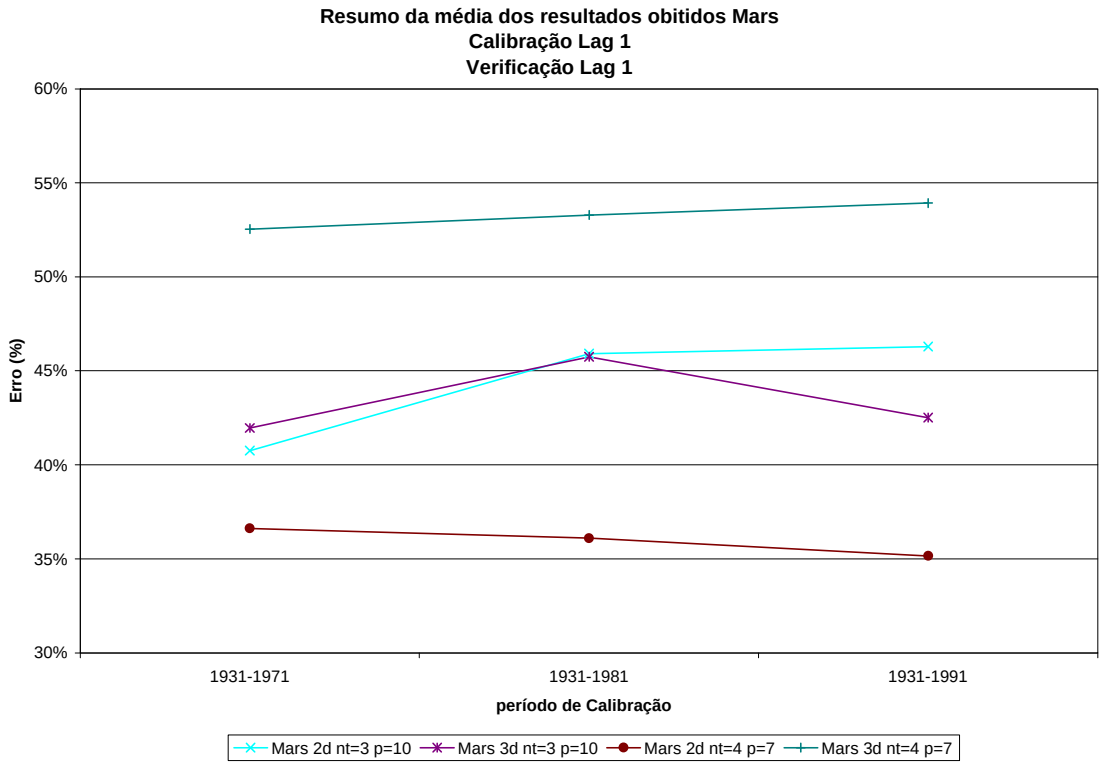


A *FIGURA 4.14* apresenta a influência do período de calibração sobre o erro relativo para os quatro modelos de previsão. Com base nesta figura é possível concluir que o período de calibração tem uma forte influência sobre a qualidade da previsão para o modelo PAR[6] e multiquadrático, sendo que para cada aumento no tamanho da série de calibração a qualidade da previsão aumenta consideravelmente.

No modelo ótimo o comportamento do erro é praticamente igual para os diversos períodos de calibração, porém para o período de calibração de 1931-1991 o erro relativo teve uma sensível melhora.

O modelo MARS 2d com nt = 4 apresentou o melhor resultado para os três períodos analisados, sendo que a influência do aumento do período de calibração é relativamente pequeno.

FIGURA 4.15 Resumo dos resultados distância temporal 1, Calibração lag 1 e Verificação lag 1 utilizando o modelo MARS.



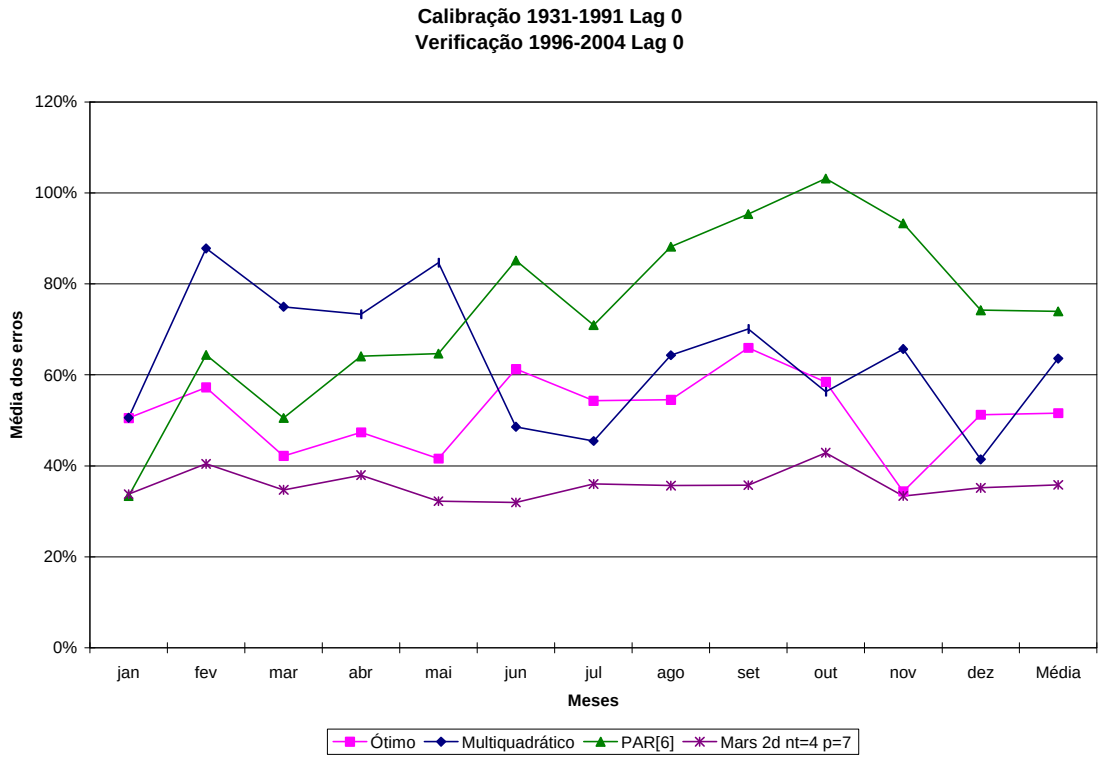
A *FIGURA 4.15* apresenta a influência do período de calibração sobre o erro relativo para o modelo MARS 2d e 3d com 3 e 4 trechos. Para o MARS 2d com nt=4 (4 trechos) o aumento do período de calibração diminuiu o erro, já para o MARS 2d com nt = 3 o aumento do período de calibração piorou consideravelmente a previsão. Para o MARS 3d com nt = 3 e nt=4 o período de calibração não influenciou na qualidade da previsão de forma significativa. E para o MARS 3d com nt = 3 o período de calibração 1931-1981 piorou consideravelmente a qualidade da previsão.

4.4 RESULTADOS OBTIDOS COM INTERVALO DE CINCO ANOS

Este item tem como finalidade apresentar os resultados obtidos pelos modelos de previsão de vazões quando o intervalo entre o último ano de calibração e o primeiro ano de verificação é igual a 5, para os modelos PAR[6], multiquadrático, ótimo e para os modelos MARS 2d e 3d com três e quatro trechos.

4.4.1 Resultados obtidos em Lag 0 de calibração e Lag 0 de verificação

FIGURA 4.16 Distância temporal 5 Calibração 1931-1991 lag 0 Verificação 1996-2004 lag 0.



A *FIGURA 4.16* apresenta o erro relativo para o período de calibração de 1931-1991 e período de verificação de 1996-2004, na *FIGURA* C.19 a *FIGURA C.24* na página 165 é apresentado a previsão para lag 0 de calibração e lag 0 de verificação para outros períodos.

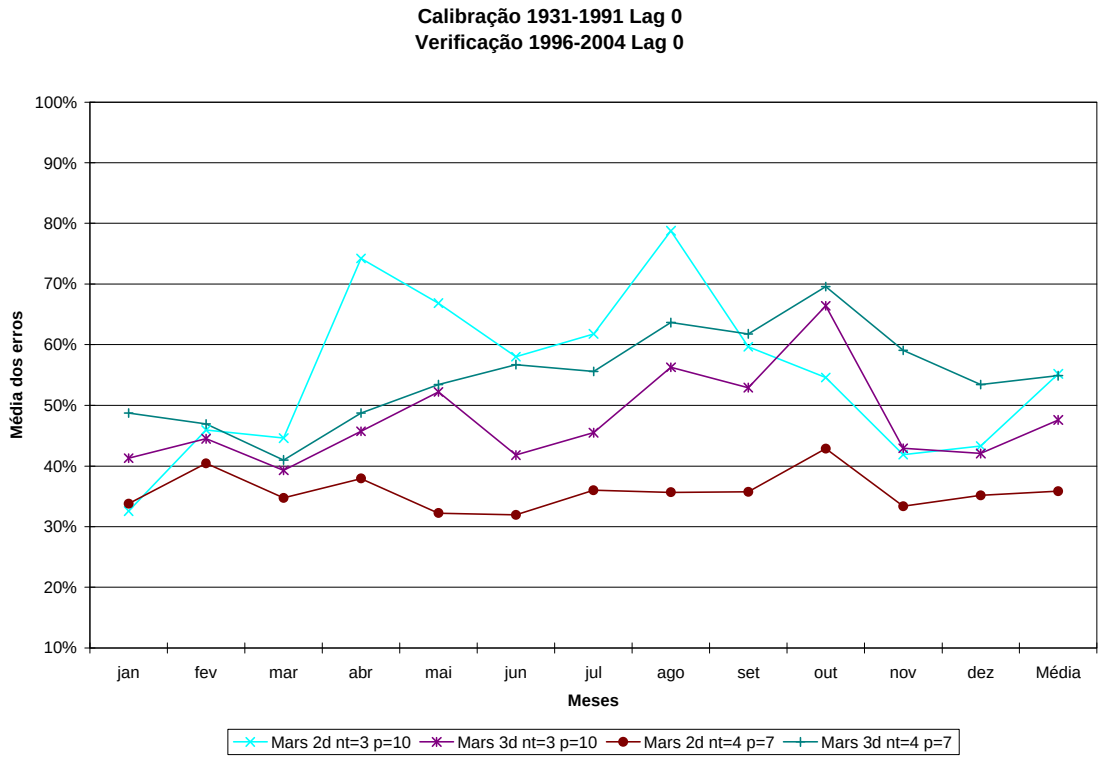
Como é possível visualizar na *FIGURA 4.16* o erro relativo entre a vazão prevista e a vazão observada para os quatro modelos de previsão, com Lag-0 de calibração e lag-0 de verificação, apresenta um comportamento relativamente parecido, porém, a magnitude dos erros é bem diferente, sendo

que o modelo PAR[6] apresenta um erro relativo bem superior ao modelo ótimo e apenas nos meses de janeiro a maio têm um erro relativo inferior ao modelo multiquadrático.

Nos meses de fevereiro a maio o modelo multiquadrático apresenta um erro relativo bem superior ao modelo ótimo. Já nos meses de junho, julho, outubro e dezembro o modelo multiquadrático obteve um erro menor que o modelo ótimo.

A previsão utilizando o modelo MARS 2d com nt = 4 apresentou um menor erro em quase todos os meses, sendo superada em janeiro pelo modelo PAR[6].

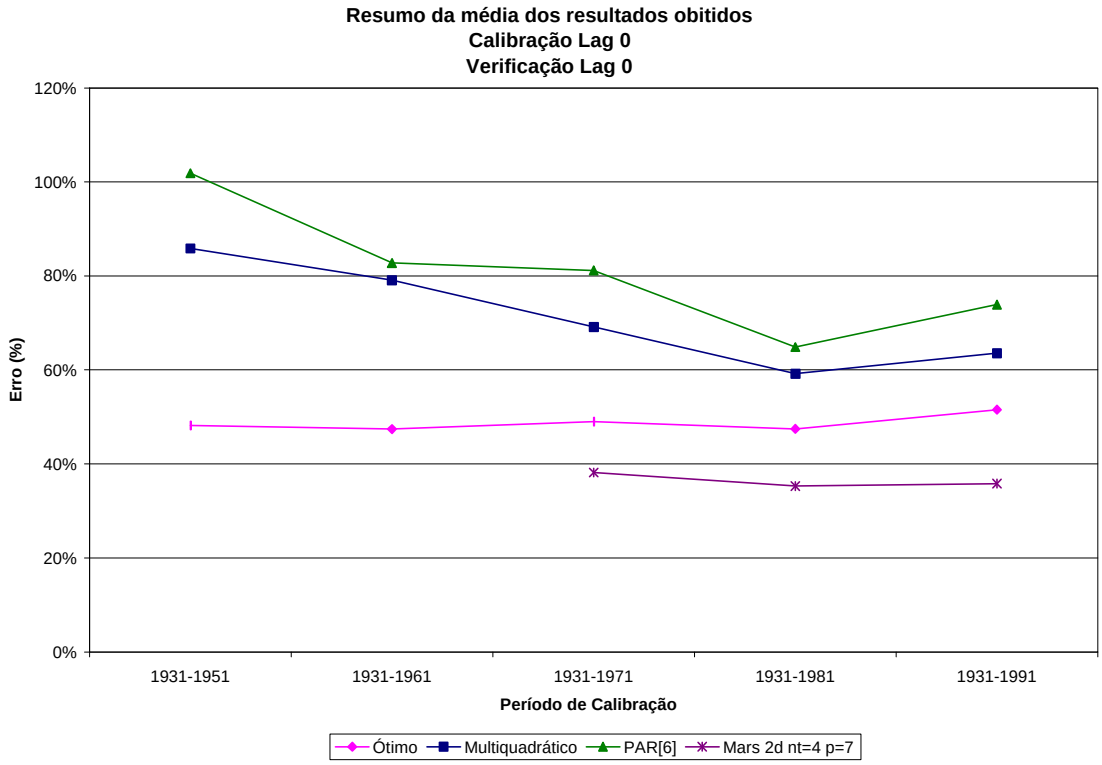
FIGURA 4.17 Distância temporal 5 Calibração 1931-1991 lag 0 Verificação 1996-2004 lag 0 utilizando o modelo MARS.



A *FIGURA 4.18* ilustra os erros das previsões utilizando o modelo MARS 2d e MARS 3d com três e quatro trechos, com Lag-0 de calibração e lag-0 de verificação. O modelo MARS 2d com 4 trechos apresentou o melhor resultado para todos os meses com exceção de janeiro em que o modelo MARS 2d com 3 trechos apresentou um resultado um pouco melhor, já utilizando 3 divisões o

modelo MARS 3d apresentou na média o segundo melhor resultado. O MARS 3d com quatro divisões apresentou um resultado ruim na média, porém em alguns meses obteve um resultado melhor que o MARS 2d com nt=3.

FIGURA 4.18 Resumo dos resultados distância temporal 5 Calibração lag 0 e Verificação lag 0.



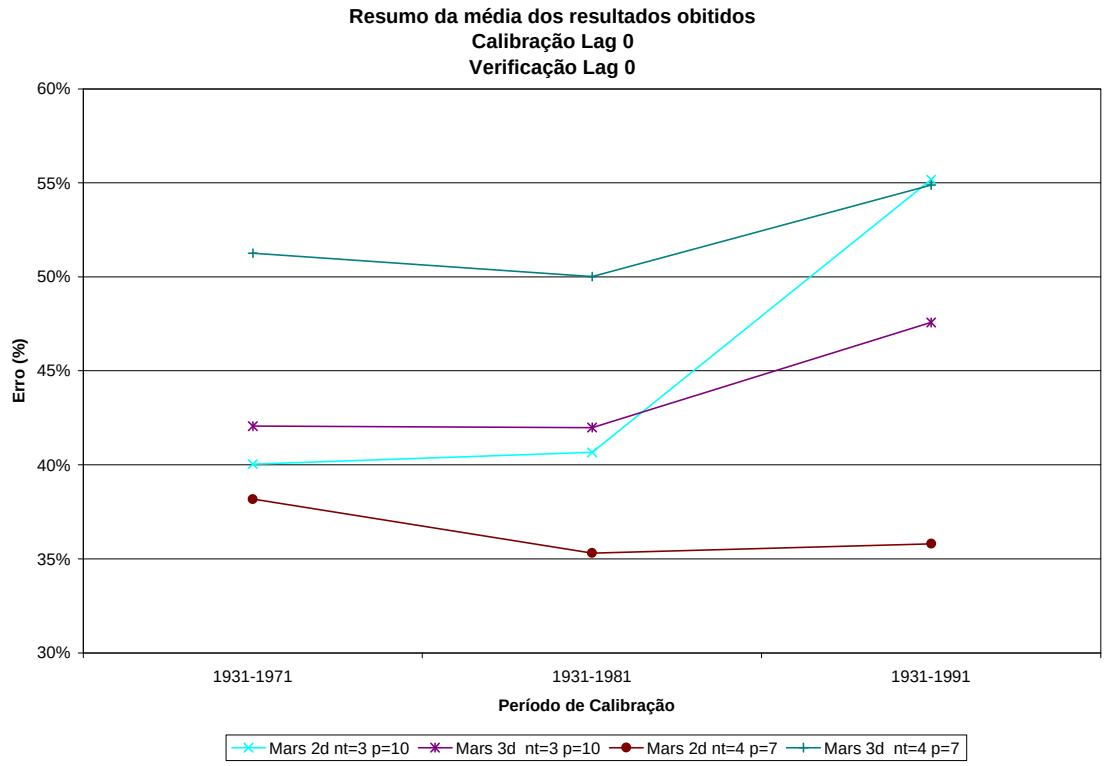
A *FIGURA 4.18* apresenta a influência do período de calibração sobre o erro relativo para os quatro modelos de previsão. Com base nesta figura é possível concluir que o período de calibração tem uma forte influência sobre a qualidade da previsão para o modelo PAR[6] e multiquadrático.

No modelo ótimo o comportamento do erro é praticamente igual para os diversos períodos de calibração, porém para o período de calibração de 1931-1991 o erro relativo teve uma leve piora.

Já a qualidade da previsão para o modelo multiquadrático tem uma leve melhora para um período de calibração utilizando uma série de 50 anos, e uma piora significativa para período de calibração de 60 anos.

A previsão utilizando o modelo MARS 2d com nt = 4 apresentou o melhor resultado para todos os períodos analisados, sendo que o aumento do período de calibração melhorou um pouco o resultado.

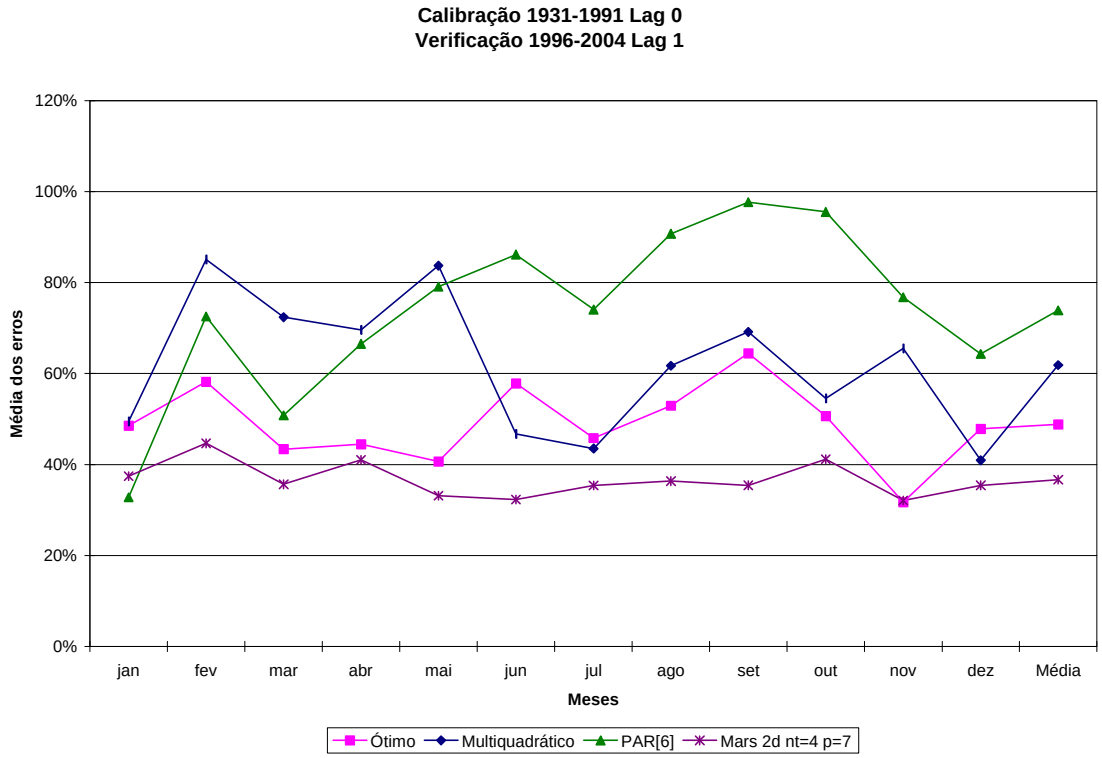
FIGURA 4.19 Resumo dos resultados distância temporal 5, Calibração lag 0 e Verificação lag 0 utilizando o modelo MARS.



A *FIGURA 4.19* apresenta a influência do período de calibração sobre o erro relativo para o modelo MARS 2d e 3d com 3 e 4 trechos. Para o MARS 2d com nt = 4 (4 trechos) o aumento do período de calibração diminuiu o erro, já para o MARS 2d com nt = 3 o aumento do período de calibração piorou consideravelmente a previsão. Para o MARS 3d com nt = 3 e nt = 4 o período de calibração de 1931-1991 piorou consideravelmente o erro.

4.4.2 Resultados obtidos em Lag 0 de calibração e Lag 1 de verificação

FIGURA 4.20 Distância temporal 5 Calibração 1931-1991 lag 0 Verificação 1996-2004 lag 1.



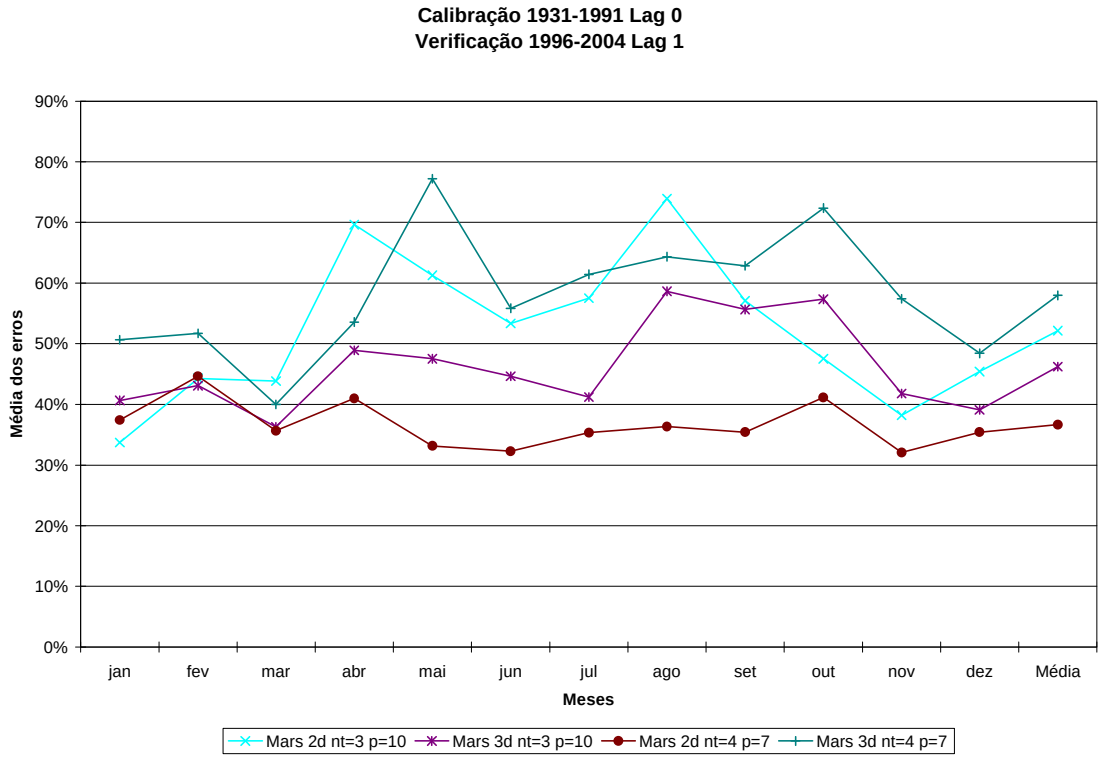
A *FIGURA 4.20* apresenta o erro relativo para o período de calibração de 1931-1991 e período de verificação de 1996-2004, na *FIGURA* C.25 a *FIGURA C.30* na página 171 é apresentado a previsão para lag 0 de calibração e lag 1 de verificação para outros períodos.

Como é possível visualizar na *FIGURA 4.20* o erro relativo entre a vazão prevista e a vazão observada para os diversos modelos de previsão, com Lag-0 de calibração e lag-1 de verificação, apresenta um comportamento relativamente parecido, porém, a magnitude dos erros é bem diferente, sendo que o modelo PAR[6] apresenta um erro relativo bem superior ao modelo ótimo, e apenas nos meses de janeiro a maio tem um erro relativo inferior ao modelo multiquadrático.

Nos meses de fevereiro a maio o modelo multiquadrático apresenta um erro relativo bem superior ao modelo ótimo. Já nos meses de junho, julho e dezembro o modelo multiquadrático obteve um erro menor que o modelo ótimo.

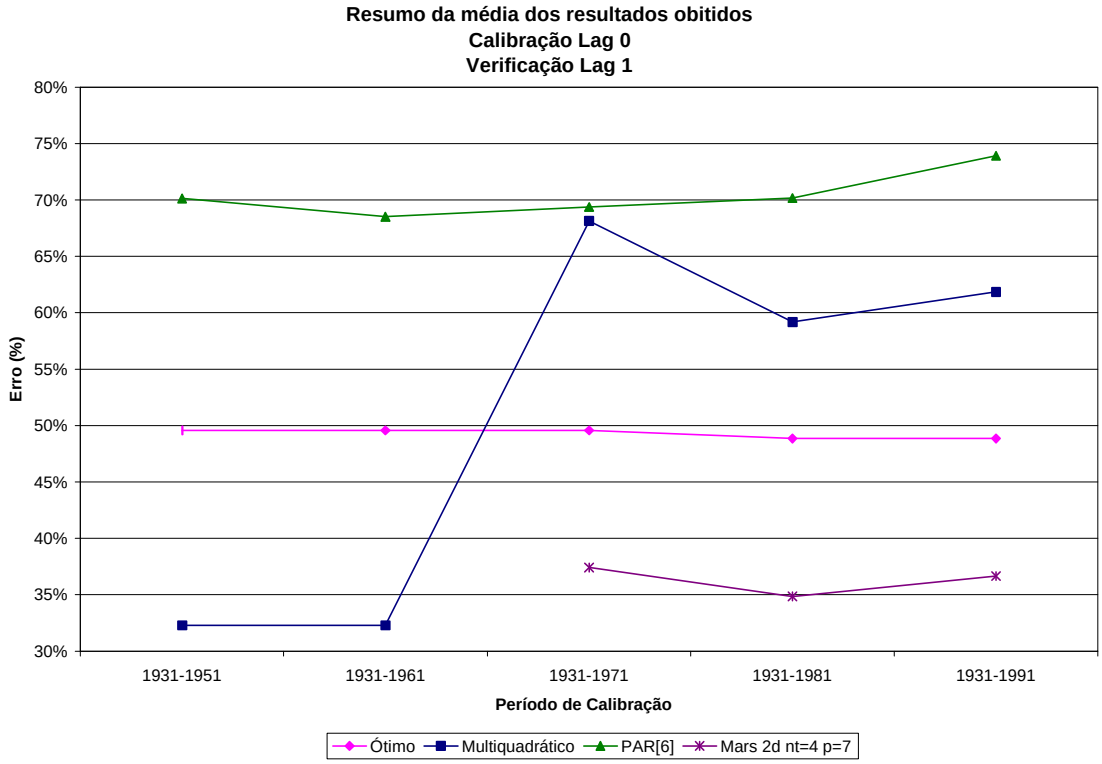
O modelo MARS 2d com nt = 4 apresentou na média o melhor resultado, sendo que no mês de janeiro o modelo PAR[6] apresentou um resultado um pouco melhor e no mês de novembro o modelo ótimo apresentou um resultado superior, para os outros meses o modelo MARS 2d com nt=4 obteve um resultado superior.

FIGURA 4.21 Distância temporal 5 Calibração 1931-1991 lag 0 Verificação 1996-2004 lag 1 utilizando o modelo MARS.



A *FIGURA 4.21* ilustra os erros das previsões utilizando o modelo MARS 2d e MARS 3d com três e quatro trechos, com Lag-0 de calibração e lag-1 de verificação. O modelo MARS 2d com 4 trechos apresentou o melhor resultado para todos os meses com exceção de janeiro e fevereiro em que o modelo MARS 2d com 3 trechos apresentou um resultado um pouco melhor, já utilizando 3 divisões o modelo MARS 3d apresentou na média o segundo melhor resultado. O MARS 3d com quatro divisões apresentou um resultado ruim na média, porém em alguns meses obteve um resultado melhor que o MARS 2d com nt=3.

FIGURA 4.22 Resumo dos resultados distância temporal 5, Calibração lag 0 e Verificação lag 1



A *FIGURA 4.22* apresenta a influência do período de calibração sobre o erro relativo para os quatro modelos de previsão.

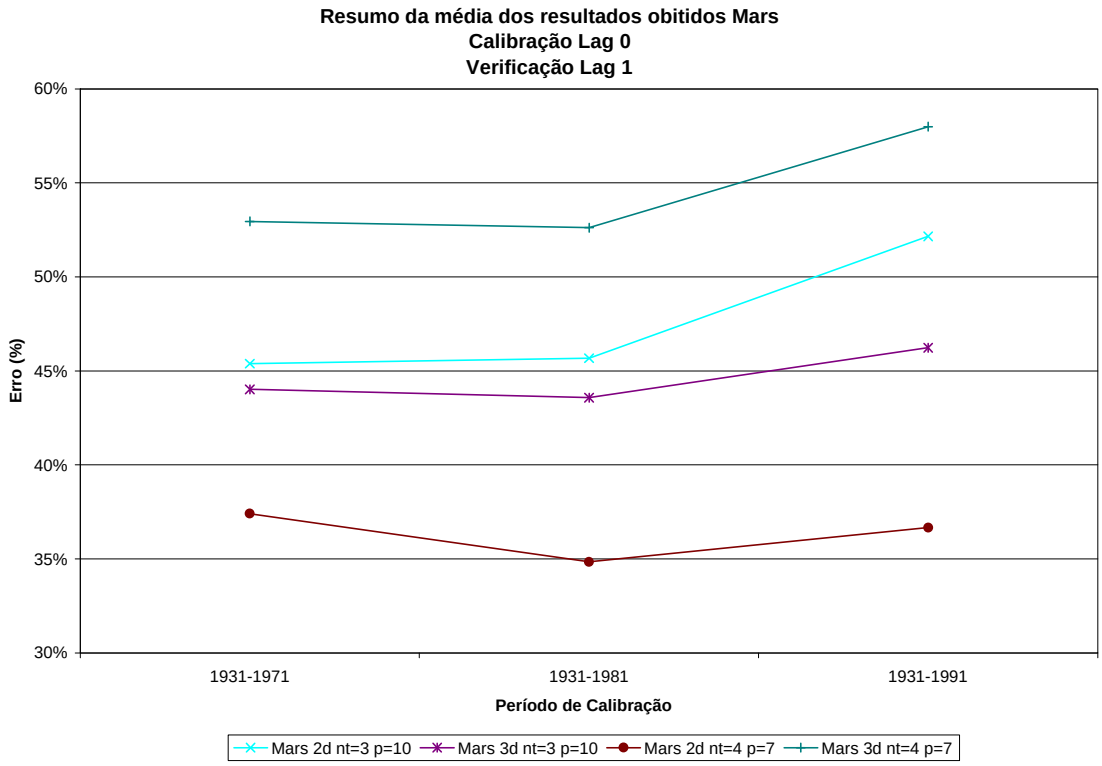
No modelo ótimo o comportamento do erro é praticamente igual para os diversos períodos de calibração, sendo que a diferença entre utilizar um período de calibração de 20 ou 70 anos é praticamente igual.

Já a qualidade da previsão para o modelo multiquadrático tem uma leve melhora para um período de calibração utilizando uma série de 50 anos, e uma piora significativa para período de calibração de 60 anos.

O modelo multiquadrático obteve um bom resultado para um período de calibração de 20 e 30 anos, e um resultado ruim para um período de calibração de 40 anos.

Para o modelo MARS 2d com nt = 4 o menor período de calibração apresentou o pior resultado. De uma forma geral o MARS 2d com nt = 4 obteve um resultado bem superior quando comparado com os outros modelos.

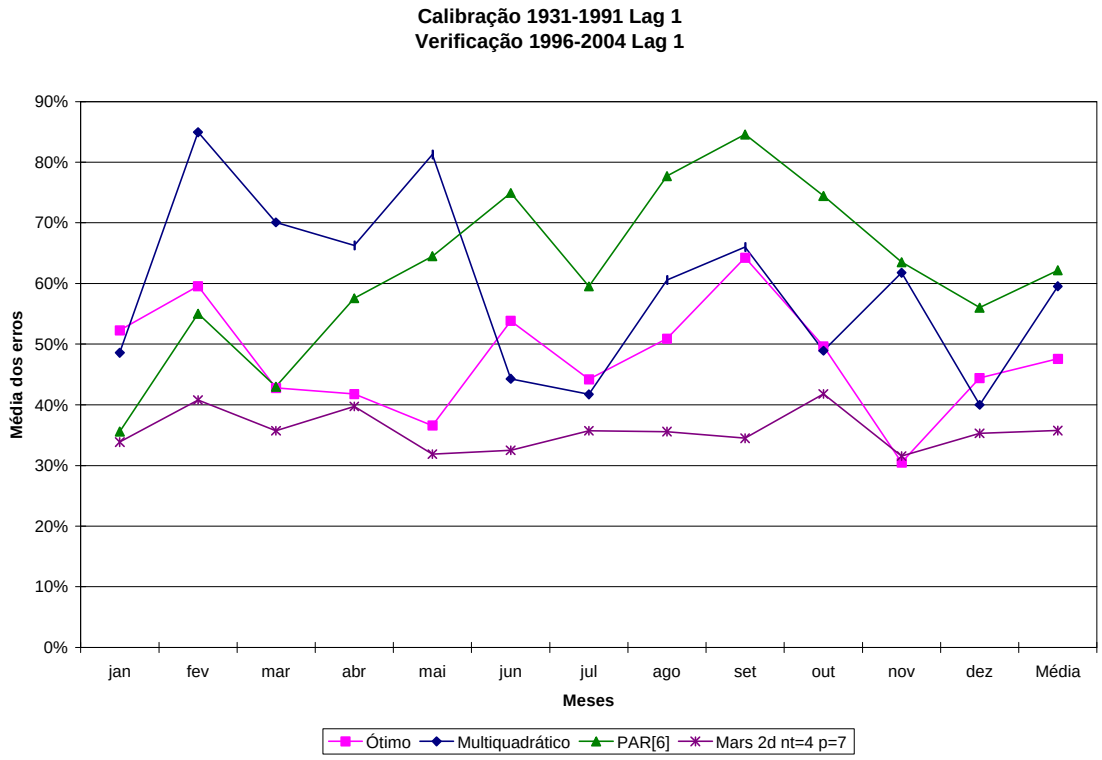
FIGURA 4.23 Resumo dos resultados distância temporal 5, Calibração lag 0 e Verificação lag 1 utilizando o modelo MARS.



A *FIGURA 4.23* apresenta a influência do período de calibração sobre o erro relativo para o modelo MARS 2d e 3d com 3 e 4 trechos. Para o MARS 2d com nt=4 (4 trechos) o aumento do período de calibração diminuiu o erro, porém, quando o período de calibração é 1931-1991 o modelo obteve um resultado um pouco pior que para o período de 1931-1981. Já para o MARS 2d com nt = 3 e o MARS 3d com nt=4 o aumento do período de calibração piorou consideravelmente a previsão. Para o MARS 3d com nt = 3 o período de calibração de 1931-1991 piorou um pouco o resultado.

4.4.3 Resultados obtidos em Lag 1 de calibração e Lag 1 de verificação

FIGURA 4.24 Distância temporal 5 Calibração 1931-1991 lag 1 Verificação 1996-2004 lag 1.



A *FIGURA 4.24* apresenta o erro relativo para o período de calibração de 1931-1991 e período de verificação de 1996-2004, na *FIGURA* C.31 a *FIGURA C.36* na página 175 é apresentado a previsão para lag 1 de calibração e lag 1 de verificação para outros períodos.

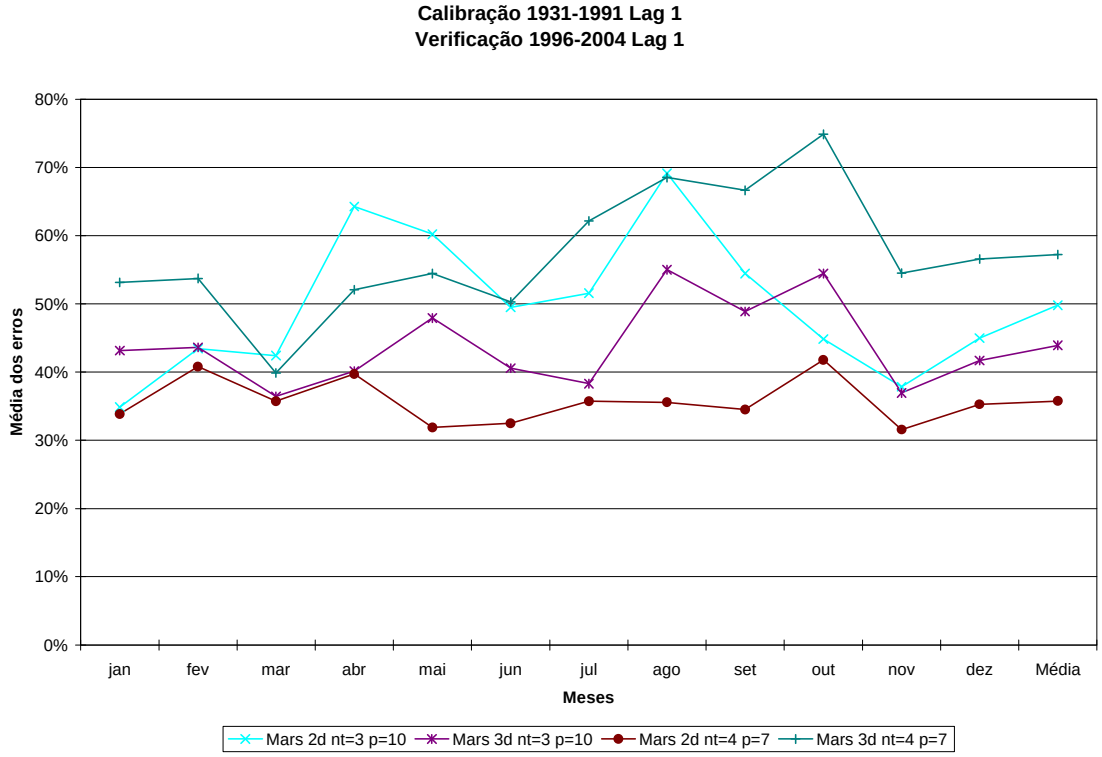
Como é possível visualizar na *FIGURA 4.24* o erro relativo entre a vazão prevista e a vazão observada para os quatro modelos de previsão, com Lag-1 de calibração e lag-1 de verificação apresenta um comportamento relativamente parecido, porém, a magnitude dos erros é bem diferente, sendo que o modelo PAR[6] apresenta um erro relativo bem superior ao modelo ótimo, e apenas nos meses de janeiro a maio tem um erro relativo inferior ao modelo multiquadrático, e em janeiro e fevereiro um erro inferior ao modelo ótimo.

Nos meses de fevereiro a maio o modelo multiquadrático apresenta um erro relativo bem superior ao modelo ótimo. Já nos meses de junho, julho,

outubro e dezembro o modelo multiquadrático obteve um erro menor que o modelo ótimo.

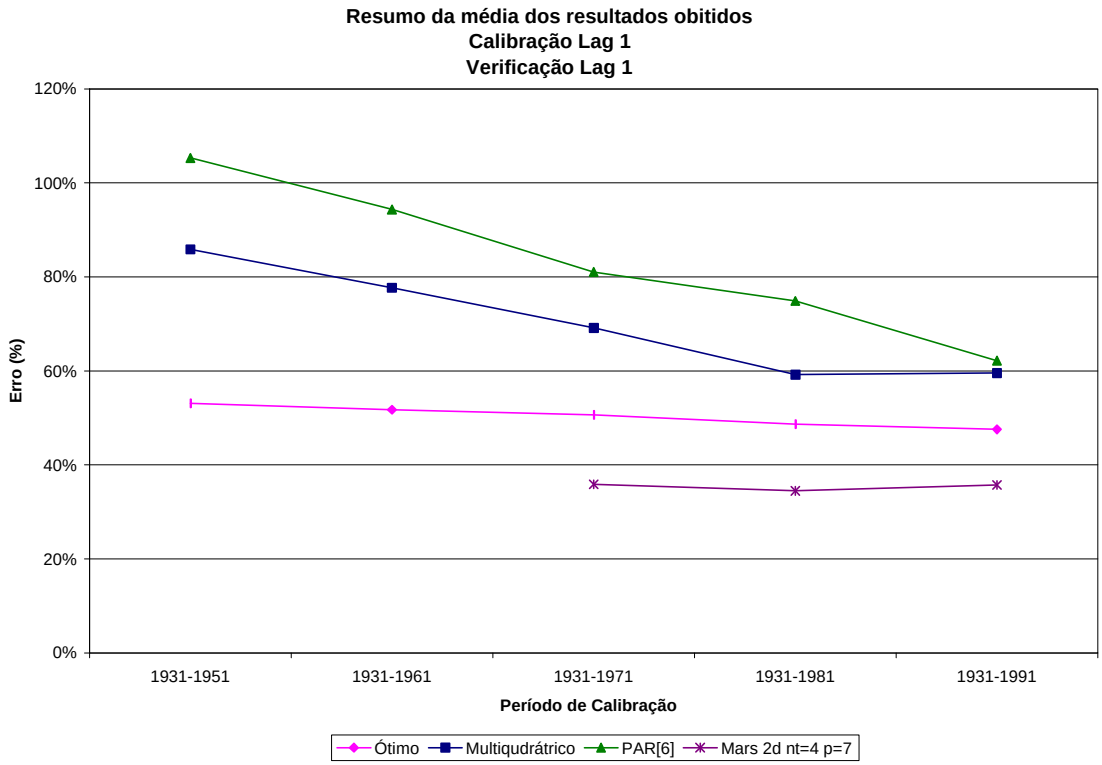
O modelo MARS 2d com nt = 4 apresentou no geral um resultado bem melhor que os outros modelos de previsão, e de uma maneira geral a variação de seus resultados entre os vários meses é bem menor.

FIGURA 4.25 Distância temporal 5 Calibração 1931-1991 lag 1 Verificação 1996-2004 lag 1 utilizando o modelo MARS.



A *FIGURA 4.25* ilustra os erros das previsões utilizando o modelo MARS 2d e MARS 3d com três e quatro trechos, com Lag-1 de calibração e lag-1 de verificação. O modelo MARS 2d com 4 trechos apresentou o melhor resultado para todos os meses, já utilizando 3 divisões o modelo MARS 3d apresentou na média o segundo melhor resultado. O MARS 3d com quatro divisões apresentou um resultado ruim na média, porém em alguns meses obteve um resultado melhor que o MARS 2d com nt=3.

FIGURA 4.26 Resumo dos resultados distância temporal 5 Calibração lag 1 e Verificação lag 1



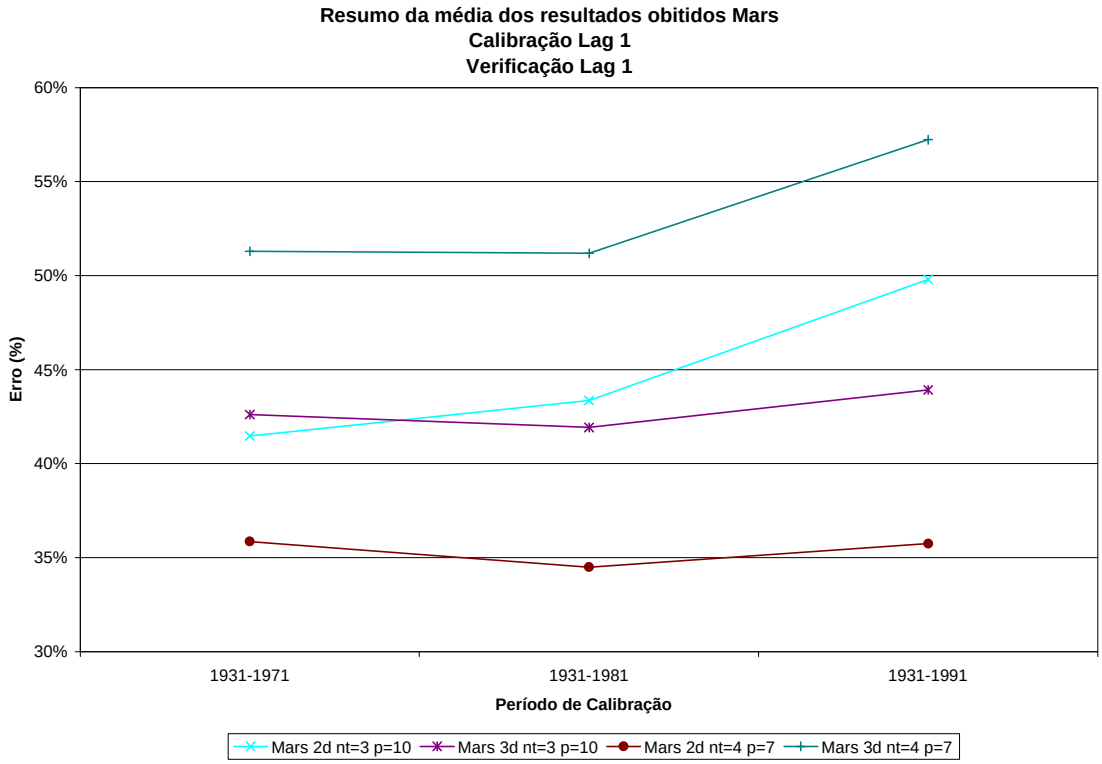
A FIGURA 4.26 apresenta a influência do período de calibração sobre o erro relativo para os quatro modelos de previsão. Com base na FIGURA 4.26 é possível concluir que o período de calibração tem forte influência sobre a qualidade da previsão para o modelo PAR[6] e multiquadrático, diminuindo o erro com o aumento do período de calibração.

No modelo ótimo o comportamento do erro é praticamente igual para os diversos períodos de calibração, sendo que a o aumento do tamanho da série de calibração melhora o resultado levemente em relação a uma série menor.

Já a qualidade da previsão para o modelo multiquadrático teve uma leve piora para período de calibração de 60 anos (1931-1991).

Para o modelo MARS 2d com nt = 4 o período de calibração influenciou pouco a qualidade dos resultados, sendo que o MARS obteve o melhor resultado entre os modelos avaliados.

FIGURA 4.27 Resumo dos resultados distância temporal 5, Calibração lag 1 e Verificação lag 1 utilizando o modelo MARS



A *FIGURA 4.27* apresenta a influência do período de calibração sobre o erro relativo para o modelo MARS 2d e 3d com 3 e 4 trechos. Para o MARS 2d com nt = 4 (4 trechos) o aumento do período de calibração diminuiu o erro, porém, quando o período de calibração é 1931-1991 o modelo obteve um resultado um pouco pior que para o período de 1931-1981. Já para o MARS 2d com nt = 3 e o MARS 3d com nt=4 o aumento do período de calibração piorou consideravelmente a previsão. Para o MARS 3d com nt = 3 o período de calibração de 1931-1991 piorou um pouco o resultado.

No Apêndice C é apresentado outras figuras com os resultados obtidos para distância temporal 1 e 5, para os períodos de calibração de 1931-1951, 1931-1961, 1931-1971, 1931-1981.

No Apêndice D é apresentado os resultados obtidos pelos modelos de previsão para cada uma das 14 usinas analisadas.

5 CONCLUSÕES E RECOMENDAÇÕES

5.1 GERAL

Nesta dissertação foram analisados quatro modelos de previsão de vazões médias mensais no horizonte de 1 a 12 meses. Os modelos são os seguintes: modelo periódico auto regressivo de ordem 6 (PAR[6]); modelo não linear MARS com duas e três dimensões; e modelos baseados nos métodos de interpolação espacial adaptados para prever vazões médias mensais. Sendo estes modelos baseados no método de interpolação ótima, e no modelo de interpolação com funções multiquadráticas. O nome do terceiro modelo é modelo ótimo e o quarto modelo é chamado de modelo multiquadrático.

Os objetivos propostos nesta dissertação são: implementar os modelos MARS, Ótimo e Multiquadrático para prever vazões afluentes médias mensais para o horizonte de 1 a 12 meses, e utilizar como estudo de caso 14 reservatórios do sistema hidroelétrico brasileiro, e comparar os resultados obtidos por esses três modelos com os resultados obtidos pelo modelo PAR de ordem 6, que foi instituído como o resultado mínimo esperado, dado que o modelo PAR[6] é o modelo atualmente utilizado para prever vazões médias mensais no setor elétrico brasileiro; e analisar a influência do período de calibração na qualidade das previsões de afluências médias mensais para o horizonte de 1 a 12 meses.

Para alcançar estes objetivos foi desenvolvido um programa em Delphi 5 que integrou em um único software todos os modelos de previsão, sendo permitido calibrar o modelo de acordo com as necessidades, escolhendo-se o período que será utilizado para calibração e prever a vazão para determinado período de verificação.

Utilizando um banco de dados com as vazões médias mensais das 14 usinas no período de 1931 à 2004, o software permite uma maneira amigável de prever-se vazões afluentes médias mensais a estas usinas.

O programa desenvolvido nesta dissertação também permite analisar a qualidade da previsão em função de um período definido para avaliação. Um

dos produtos de saída do programa é uma tabela contendo a média do erro relativo para cada usina, em cada mês e para cada modelo analisado.

Pelo que foi apresentado nesta dissertação comprova-se que os propósitos e objetivos foram totalmente atingidos. A seguir apresentam-se as principais conclusões e recomendações para a realização de trabalhos futuros.

5.2 CONCLUSÕES

As principais conclusões deste trabalho são:

i) Os modelos de previsão MARS, Ótimo e Multiquadrático obtiveram um erro médio menor que o modelo PAR[6].

ii) O modelo MARS de duas dimensões e com 4 trechos obteve o menor erro entre as vazões observadas e previstas entre os modelos analisados. Em seguida os modelos que obtiveram as melhores previsões são: o modelo Ótimo e o Multiquadrático. O modelo PAR[6] que serve como referência obteve o pior resultado entre os modelos analisados.

iii) A qualidade das previsões obtidas pelo modelo MARS de duas dimensões e com quatro trechos é superior a 100% quando comparado com o modelo PAR[6].

iv) O modelo MARS com duas dimensões utiliza apenas a vazão do mês anterior para efetuar a previsão, enquanto o modelo PAR[6] utiliza as últimas seis vazões observadas para efetuar a previsão.

v) O período de calibração é muito importante para a qualidade das previsões de vazões médias mensais no modelo PAR[6], sendo que a diferença entre o erro médio quando utiliza-se 20 anos de calibração e quando utiliza-se 60 anos é superior a 100%.

vi) O período de calibração para o modelo ótimo não é uma variável muito importante, já que a diferença entre a vazão prevista e observada é pequena.

vii) O modelo MARS necessita de no mínimo 40 anos de dados observados para obter resultados satisfatórios.

viii) O modelo MARS com três dimensões e com 3 trechos obteve bons resultados. Para o modelo com quatro trechos obteve-se resultados ruins, resultado explicado pelo pequeno número de vazões observadas (a série analisada tem no máximo 60 anos).

ix) Para séries de vazões muito curtas, menores do que 30 anos, utilizando o modelo Ótimo obteve-se o melhor resultado.

x) O período de calibração atualizado não mostrou-se uma variável tão importante, embora a qualidade das previsões melhore em função desta análise.

5.3 RECOMENDAÇÕES

A partir do que foi analisado nesta dissertação, constatou-se uma série de problemas e questões a serem analisadas em estudos futuros:

i) No modelo MARS foi utilizado um método linear para prever afluências, a utilização de equações não lineares como as transformação logarítmica e splines cúbicas pode ser uma alternativa para melhorar a previsão.

ii) Analisar os modelos propostos para pequenas bacias hidrográficas.

iii) Avaliar os modelos propostos para reconstituição de séries de vazões naturais.

iv) Neste trabalho foi usada a série de vazões mensais para efetuar a previsão de vazões médias mensais. Sugere-se a consideração de outras variáveis como precipitação, e outras variáveis hidrometeorológicas na modelagem multivariada.

6 REFERÊNCIAS BIBLIOGRÁFICAS

Andriolo, M. V.; Kaviski, E., (2005): **Projeto HG-211, Revisão, atualização e aperfeiçoamento do sistema de avaliação da evaporação líquida dos reservatórios do sistema interligado nacional – SisEvapo 2.0, relatório nº 4.** LACTEC/CEHPAR.

Barth, F. T.; Pompeu C. T.; Fill H. D. O. A.; Tucci C. E. M.; Braga B. P. F., (1987): **Modelos para gerenciamento de recursos hídricos**. ABRH.

Bera, A. K.; Suprayitno T.; Premaratne G., (2002): **On some heteroskedasticity-robust estimators of variance-covariance matriz of lest-squares estimators.** Journal of Statistics Planning and Inference, 108, p.121-136.

Boor, C. (1978): **A Practical Guide to Spline**. Spring-Verlag New York

Bras, R. L.; Rodríguez-Iturbe, I. (1984): **Random functions and hydrology.** Addison-Wesley Publishing Company.

Briand, L. C.; Freimut, B.; Vollei, F. (2000): **Using multiple adaptive regression splines to understand trens in inspection data and identify optimal inspection rates**. International Software Engineering Research Network. Technical report ISERN-00-07.

Briand, L. C.; Freimut, B.; Vollei, F. (2004): **Using multiple adaptive regression splines to support decision making in code inspections**. The Journal of Systems and Software, 73, p. 205-217.

Burden R. L., Faires J. D. (2003): **Análise numérica**, Ed. Pioneira Thomson Learning. P. 126-138.

Caffey J. E. (1963): **Inter-Station correlations in annual precipitation and in annual effective precipitation**. Colorado State University, Hydrology Papers, 6, pag. 1-47.

Cao, Y.J., Jiang, L., Wu, Q.H. (2000): **An evolutionary programming approach to mixed-variable optimization problems**. Appl. Math. Modelling 24, 931.

Chou, S.; Lee T.; Chen Y. E. S. (2004): **Mining the breast cancer pattern using artificial neural networks and multivariate adaptive regression splines.** Expert Systems with Applications, article in press.

Craven, P.; Wahba, G. (1979): **Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation.** Numerische Mathematik, 31, p. 317-403.

Creutin, J. D.; Obled C. (1982): **Objective analysis and mapping techniques for rainfall fields: an objective**. Water Resourses Research, v.18, n. 2, p.413-431, apr.1982.

Eyink, G. L., (2001): **Variational Formulation of Optimal Nonlinear Estimation**. arXiv: physics, Vol. 2, p. 1-58, mar. 2001.

Flood, B. I.; Kartam, N., (1994): **Neural networks in civil engineering. I: Principles and understanding.** Journal of Computing in Civil Engineering, vol.8, n° 2, apr. 1994.

Friedman, J. H., Silverman, B. W., (1989): **Flexible Parsimonious Smoothing and Additive Modeling**. Technometrics, Vol. 31, n°1, p. 3-39, fev. 1989.

Friedman, J. H., (1990): **Multivariate Adaptive Regression SPLINES. Annals of Statistics**, 19(1), p. 1-67, ago. 1990

Golup G. H., Van Loan C, F., (1983): **Matrix Computations**. The Johns Hopkins University Press, Baltimore, MD.

Ing, C-K. ; W, C-Z. (2003): **On same-realization prediction in an infinite-order auto regressive process.** Journal of Multivariate Analysis, 85, p. 130-155.

Karunanithi, N. et. Al. (1994): **Neural networks for river flow prediction.** Journal of Computing in Civil Engineering. Vol. 8, n° 2, Pág. 201-220

Kaviski E., (1992): **Métodos de regionalização de eventos e parâmetros hidrológicos**. Dissertação de mestrado. Universidade Federal do Paraná, Curitiba. Pág. 78-84, 125-130.

Kaviski E., (2006): **Solução de problemas de fenômenos de transporte pelo método de Monte Carlo**. Tese de doutorado a ser defendida. Universidade Federal do Paraná, Curitiba.

Kavvas, M. L. (2003): **Nonlinear hydrologic processes: Conservation equations for determining their means and probability distributions.** Journal of Hydrologic Engineering, vol 8, n.2, p. 44-53.

Ko, M.; Osei-Bryson, K. (2003): **The productivity impacto f information technology in the healthcare industry: na empirical study using a regression spline-based approach.** Information and Software Technology, 46, p.65-73

Larsson, E. K.; Söderström, T., (2002): **Identification of continuous-time AR processes from unevenly sampled data**. Aumomatica, 38, p.709-718.

Lacerda, E. G. M.; Carvalho, A. C. P. L. (1999): **Introdução aos algoritmos genéticos.** Em XIX Congresso Nacional da Sociedade Brasileira de Computação. Anais v. II, p. 51-125.

Lele, S.; Taper, M. L. (2002): **A composite likelihood approach to (co)variance components estimation.** Journal of Statistics Planning and Inference, 103, p.117-135.

Loaciga, H. A. et. al. (1988): **Linear spatial interpolation: Analysis with an application to San Joaquim Valley.** Stochastic Hydrology Hydraulics, 2, p. 113-136.

Machado, F. W., (2005): **Modelogem chuva-vazão mensal utilizando redes neurais artificiais.** Dissertação de mestrado. Universidade Federal do Paraná, Curitiba.

Maidment, D. R.; editor chief, (1992): **Handbook of Hydrology.** McGraw-Hill.

Mass, A. et. al. (1962): **Design of water-resource system.** Cambridge, Mass.: Havard University Press.

Mazer, W. (2003): **Método não linear para previsões de vazões** .Dissertação de mestrado. Universidade Federal do Paraná, Curitiba.

Mine, M. R. M. (1984): **Modelos estocásticos lineares para previsão de cheias em tempo real.** Dissertação de mestrado. Escola Politécnica da USP, São Paulo.

Mine, M. R. M. (1998): **Método determinístico para minimizar o conflito entre gerar energia e controlar cheias.** Tese de doutorado. Universidade Federal do Rio Grande do Sul, Porto Alegre.

Morettin P. A., Toloi C. M. C., (1981): **Modelos para previsão de séries temporais**. 13 Colóquio brasileiro de matemática.

Morin, G.; Fortin, J. P.; Sochanska, W.; Lardeau, J. P. (1979): **Use of principal component analysis to identify homogeneous precipitation stations for optimal interpolation**. Water Resources Reserch, v.15, n. 6, p. 1841-1850, Dec 1979.

Nayak, P. C. et al. (2004): **A neuron-fuzzy computing technique for modeling hydrological time serie.** Journal of Hydrology. 291, pág. 52-66.

Nguyen-Cong V., Dang G. V., Rode B. M., (1996): **Using multivariate adaptive regression splines to QSAR studies of dihydroartemisinin derivatives**. Eur J. Med. Chem, 31 p.797-803.

O'Connell, P. E. (1974): **Stochastic modeling of long term persistence in streamflow sequences.** London: Phd thesis presented to the

Civil Engineering Department, Imperial College.

Osei-Bryson, K.-M.; Ko, M., (2004): **Exploring the relationship between information technology investiments and firm performance using regression splines analysis.** Information & management, 42, p.1-13.

Patterson, H. D.; Thompson, R. (1971): **Recovery of interblock information when block sizes are unequal.** Biometrika, 58, p.545-554.

Reed, P., Minsker, B., Goldberg, D.E.(2000): **Designing a competent simple genetic algorithm for search and optimization**. Water Resour. Res. 36 (12), 3757.

Rohn, M. C. (2002): **Uma aplicação das redes neurais artificiais à previsão de chuvas de curtíssimo prazo.** Dissertação de mestrado. Universidade Federal do Paraná, Curitiba.

Sakata S., Ashida F., Zako M., (2004): **Na efficient algorithm for Kriging approximation and optimization with large-scale sampling data**, Computer methods in applied mechanics and engineering, 193, p. 385-404.

SCEN/GTMC (1980): **Sistema para previsão de vazões médias mensais para o programa de operação.** Grupo coordenador para operadoção interligada Região Sul/Sudeste – Subcomitê de estudos energéticos, Curitiba.

Sephton, P. (2001): **Forecasting Recessions: Can we do better on MARS?**. Federal Reserve Bank of St. Louis, mar-abr, p. 39-49.

Smith W. H. F., Wessel P., (1990): **Gridding with continuous curvature splines in tension, Geophysics**, Vol. 55, n°3, pag. 293-305.

Stoer, J.; Bulirsch, R. (2002): **Introduction to numerical analysis.** Volume 12. Springer-Verlag, New York.

Stone C.J. and Koo, Cha-Yong, (1985). **Additive splines in statistics**. Proceedings, Annual Meeting of Amer. Assoc., Statist., August, p.45-48.

Sutton ;Boyden. (1994): **Genetic algorithm: A general search procedure.** Am. J. Phys, 62(6), p. 549-552.

Tabios, III G.Q.; Salas, J. D. (1985): **A comparative analysis of techniques for spatial interpolation on precipitation**. Water Resources Bulletin, v.21, n. 3, p.365-380, june 1985.

Tokar, A. S.; Johnson, P. A. (1999): **Rainfall runoff modeling using artificial neural network.** Journal of Hydrologic Engineering, 4, pág. 232-239.

Tucci, C.M. (1998): **Modelos Hidrológicos.** Ed. UFRGS/ ABRH.

Yevjevich V.; Karplus A. K., (1973). **Area-time structure of the monthly precipitation process.** Colorado State University, Hydrology Papers, 64, pag. 1-45.

Xu, Q. –S., et. al. (2004). **Multivariate adaptive regression splines – studies of HIV reverse transcriptase inhibitors.** Chemometrics and intelligent laboratory systems, 72, p.27-34.

Zeiri Y. (1995): **Prediction of the lowest energy structure of clusters using a genetic algorithm.** Phys. Rev. E. Sl, R2769.

APÊNDICE A DISTRIBUIÇÃO LOG-NORMAL DE TRÊS PARÂMETROS

A distribuição log-normal de três parâmetros tem muitas aplicações em hidrologia. Em muitos casos os logaritmos da variável aleatória (*X*) não são normalmente distribuídos, porém subtraindo o parâmetro x da variável *X* antes de calcular-se o logaritmo desta variável é possível resolver este problema:

$$Y = \ln(X - x) \tag{A.1.1}$$

Modelando *Y* em relação a *X* tem-se a distribuição normal:

$$X = x + e^{Y} \tag{A.1.2}$$

Sendo a probabilidade no intervalo de confiança *p* da variável x_p dado por:

$$x_p = x + \exp(m_y + s_y Z_p) \tag{A.1.3}$$

Neste caso a média m_x e a variância s_x^2 são dados por:

$$m_x = x + \exp\left(m_y + \frac{1}{2} s_y^2\right) \tag{A.1.4}$$

$$s_x^2 = \left[\exp(2m_y + s_y^2)\right]\left[\exp(s_y^2) - 1\right] \tag{A.1.5}$$

$$g_x = 3f + f^3 \tag{A.1.6}$$

$$f = \left[\exp(s_y^2) - 1\right]^{0.5} \tag{A.1.7}$$

Como o logaritmo de base *10* é empregado, tem-se $W = \log(X - x)$, sendo que o valor de x e g_x não é afetado, porém tem-se:

$$m_x = x + 10^{mw + \ln(10) s_w^2 / 2} \quad \text{(A.1.8)}$$

$$s_x^2 = (m_x - x)^2 f^2 \quad \text{(A.1.9)}$$

$$f = \left(10^{\ln(10) s_w^2 / 2} - 1\right)^{0.5} \quad \text{(A.1.10)}$$

O método do momento para estimadores da distribuição Log-Normal de três parâmetros é relativamente ineficiente, por isso, um estimador simples e eficaz $\hat{x}$ pode ser ilustrado como:

$$\hat{x} = \frac{x_1 x_n - \tilde{x}^2}{x_1 + x_n - 2\tilde{x}} \quad \text{(A.1.11)}$$

$$\tilde{x} = \begin{cases} x_{k+1} & \text{se n é ímpar} \quad e \quad n = 2k+1 \\ \dfrac{x_k + x_{k+1}}{2} & \text{se n é par} \quad e \quad n = 2k \end{cases}$$

Onde $x_1 + x_n - 2\tilde{x} > 0$ e x_1 , x_n são respectivamente o maior e o menor valor observado e $\tilde{x}$ é a mediana da série. Quando $x_1 + x_n - 2\tilde{x} < 0$ a fórmula prova que $\ln(x - x)$ é normalmente distribuída.

Dado x é possível estimar m_y, s_y^2 usando a média e a variância da amostra com $y_i = \ln(x_i - \hat{x})$ e $w_i = \log(x_i - \hat{x})$.

O desempenho do estimador $\hat{x}$ com o resultado da estimativa de $m_y \; e \; s_y^2$ é melhor que o método dos momentos e é tão robusto quanto o

método da máxima verossimilhança dos estimadores.

Para a distribuição Log-Normal de 2 e 3 parâmetros o segundo momento L é dado por:

$$l_2 = \exp\left(m_y + \frac{s_y^2}{2}\right) erf\left(\frac{s_y}{2}\right) = 2\exp\left[\left(m_y + \frac{s_y^2}{2}\right)\left(f\frac{s_y}{\sqrt{2}} - \frac{1}{2}\right)\right] \qquad \text{(A.1.12)}$$

APÊNDICE B ALGORITMO GENÉTICO

B.1. INTRODUÇÃO

Algoritmos genéticos são métodos de otimização global inspirados nos mecanismos de evolução de populações de seres vivos. O nome é derivado da habilidade do algoritmo para simular o processo de criação da vida e através da adptação ao meio ambiente. Os algoritmos genéticos tem sido aplicados com sucesso numa grande variedade de problemas (Sutton e Boyden, 1994; Zeiri, 1995; Cao e outros, 2000; Reed e outros, 2000). Os algoritmos genéticos diferem dos tradicionais métodos de otimização em 4 importantes aspectos: (i) usam um código para as variáveis de controle (geralmente correntes de bits chamados de genes) em vez das próprias variáveis; (ii) pesquisam numa população a solução para uma outra população, ao invés de pesquisar individualmente; (iii) usam somente a informação da função objetivo e não as suas derivadas; e (iv) usam regras de transição probabilísticas e não determinísticas.

B.2. DESCRIÇÃO DO ALGORITMO GENÉTICO

O problema de otimização examinado neste apêndice consiste em minimizar uma função $f(\boldsymbol{x})$, sendo $\boldsymbol{x}$ um vetor com dimensão p, cujos componentes estão compreendidos em intervalos finitos: $a_i \le x_i \le b_i$, i=1,...,p. Para aplicar o algoritmo genético o vetor $\boldsymbol{x}$ é transformado numa corrente de bits, usando-se a seguinte expressão:

$$z = \sum_{i=1}^{p} \left\lfloor \frac{x_i - a_i}{b_i - a_i}(2^{t-1} - 1) + \frac{1}{2} \right\rfloor 2^{(i-1)t}, \qquad \text{(B.1)}$$

sendo t o número de bits considerado para representar cada um dos componentes do vetor $\boldsymbol{x}$, e z é uma variável binária com $s=pt$ bits. Computacionalmente a variável z pode ser representada em um vetor com p números inteiros que ocupam t/8 bytes (geralmente t=32 bits).

Conhecendo-se a variável *z* pode-se determinar os componentes do vetor ***x***, usando-se a seguinte expressão:

$$x_i = a_i + (b_i - a_i)\frac{\left\lfloor z / 2^{(i-1)t} \right\rfloor \bmod 2^t}{2^{t-1} - 1}, \qquad i=1,...,p. \tag{B.2}$$

O algoritmo genético pode ser aplicado considerando-se os seguintes passos:

(i) Gera-se aleatoriamente uma população z_j (j=1,...,n), sendo n o tamanho da população. Parte desta população (30%) geralmente é escolhida de forma que a função objetivo seja inferior a um limite máximo que deve ser previamente estabelecido. Para cada valor de z_j, pode-se determinar $\boldsymbol{x}_j$ usando-se a expressão (B.2).

(ii) A população deve ser classificada em ordem crescente em função dos valores da função objetivo $f(\boldsymbol{x}_j)$, j=1,...,n. Para cada elemento da população associa-se uma probabilidade, estimada por:

$$p(\boldsymbol{x}_j) = \frac{[(n - j + 1) / f(\boldsymbol{x}_j)]^2}{S}, \qquad j=1,...,n, \tag{B.3}$$

sendo que: $f(\boldsymbol{x}_{j-1}) \leq f(\boldsymbol{x}_j)$, j=2,...,n; e $S = \sum_{j=1}^{n} [(n - j + 1) / f(\boldsymbol{x}_j)]^2$.

Se $f(\boldsymbol{x}_1)$ é inferior a um limite mínimo aceitável o processo iterativo é encerrado, adotando-se como solução o vetor $\boldsymbol{x}_1$.

(iii) Uma nova população é formada a partir da população existente por meio de 3 processos:

Reprodução: Usando-se a distribuição de probabilidades discreta definida em (B.3) sorteiam-se aleatoriamente *n* valores $\boldsymbol{x}'_j$ (*j*=1,...,*n*). Através da expressão (B.1) determinam-se as variáveis z'_j correspondentes.

Recombinação: Sorteiam-se aleatoriamente *r* pares de elementos da população. Geralmente adota-se *r* como um percentual de *n*, por exemplo: *r*=0.6*n*. Considera-se que cada par de elementos pode acasalar-se uma vez. Para cada par de variáveis sorteadas escolhe-se aleatoriamente um bit *m* compreendido entre 1 e *s*. Por exemplo, considerando-se que um par de elementos é constituído pelas variáveis z'_7 e z'_{31}, a recombinação é realizada transferindo-se o conteúdo dos primeiros 10 bits (neste caso *m*=10) de z'_7 para z'_{31}, e de z'_{31} para z'_7:

Antes da recombinação: z'_7 0011011101 | 00111···011110101111111

z'_{31} 0111101110 | 01101···101100001110010

Após a recombinação: z'_7 0111101110 | 00111···011110101111111

z'_{31} 0011011101 | 01101···101100001110010

Mutação: Para cada par de elementos da população que foram recombinados considera-se que pode ocorrer um processo de mutação. A mutação é simulada sorteando-se aleatoriamente um número inteiro *l* entre 0 e *u*, sendo *u* o número máximo de mutações que podem ocorrer num determinado elemento. Geralmente *u* é adotado como um percentual de *s*, por exemplo *u*=0.05*s*. Quando $l > 0$, sorteiam-se *l* números aleatórios entre 1 e *s*, que identificam os bits em que ocorrem as mutações. Em cada um dos bits selecionados aplica-se o operador $\neg$ (não lógico). Por exemplo, supondo-se que para a variável z'_7, $l = 2$, e que os bits sorteados são iguais a 7 e 121 (neste caso $s = 128$), a mutação é processada da seguinte forma:

Antes da mutação: z'_7 011110111000111···011110101111111

↓ ↓

Após a mutação: z'_7 011110011000111···011110111111111

↑ ↑ ↑

Bit 7 Bit 121 Bit 128

(iv) A população identificada pelas variáveis z_j (*j*=1,...,*n*) é substituída pelas variáveis z'_j. Usando-se a expressão (F.2) determinam-se as variáveis correspondentes $\boldsymbol{x}_j$. O processamento do algoritmo prossegue retornando-se ao passo (ii).

B.3. TESTE DE DEJONG

Para verificar os resultados obtidos com o algoritmo descrito no item anterior foram realizados testes para um conjunto de funções propostas por DeJong, em 1975 (Lacerda e Carvalho, 1999). O conjunto é formado por 5 funções simples com vários tipos de superfícies. A tabela B.1 apresenta as funções investigadas. Todas as funções de teste de DeJong representam problemas de minimização.

Os testes aplicados comprovaram a validade do algoritmo genético descrito neste apêndice.

Tabela B.1 Funções de teste de DeJong (Lacerda e Carvalho, 1999).

Função	**Limites**	**Observações**
$x_1^2 + x_2^2 + x_3^2$	$-5.12 \le x_1, x_2, x_3 \le 5.12$	
$100(x_1^2 - x_2)^2 + (1 - x_1)^2$	$-2.048 \le x_1, x_2 \le 2.048$	
$\sum_{i=1}^{5} \left\lfloor x_i + \frac{1}{2} \right\rfloor$	$-5.12 \le x_i \le 5.12$ i=1,...,25	
$\sum_{i=1}^{30} i\, x_i^4 + h$	$-1.28 \le x_i \le 1.28$ i=1,...,30	h é um número aleatório normal com média 0 e variância unitária.
$0.002 + \sum_{j=1}^{25} \frac{1}{j + (x_1 - a_{1j})^6 + (x_2 - a_{2j})^6}$	$-65.536 \le x_1, x_2 \le 65.536$	$a_{1j} = a_{2j} = 10$ j=1,...,25

APÊNDICE C ANÁLISE DAS PREVISÕES DE VAZÕES

C.1 RESULTADOS OBTIDOS COM INTERVALO DE UM ANO

Este apêndice tem como finalidade apresentar os resultados obtidos pelo programa de previsão quando o intervalo entre o último ano de calibração e o primeiro ano de verificação é igual a 1, utilizando a metodologia descrita no item 4. Sendo que as previsões utilizando o modelo MARS foram analisadas a partir de 1971, compreendendo um período de calibração mínimo de 40 anos.

C.1.1 Resultados obtidos em Lag 0 de calibração e Lag 0 de verificação

FIGURA C.1 Distância temporal 1 Calibração 1931-1951 lag 0 Verificação 1952-2004 lag 0.

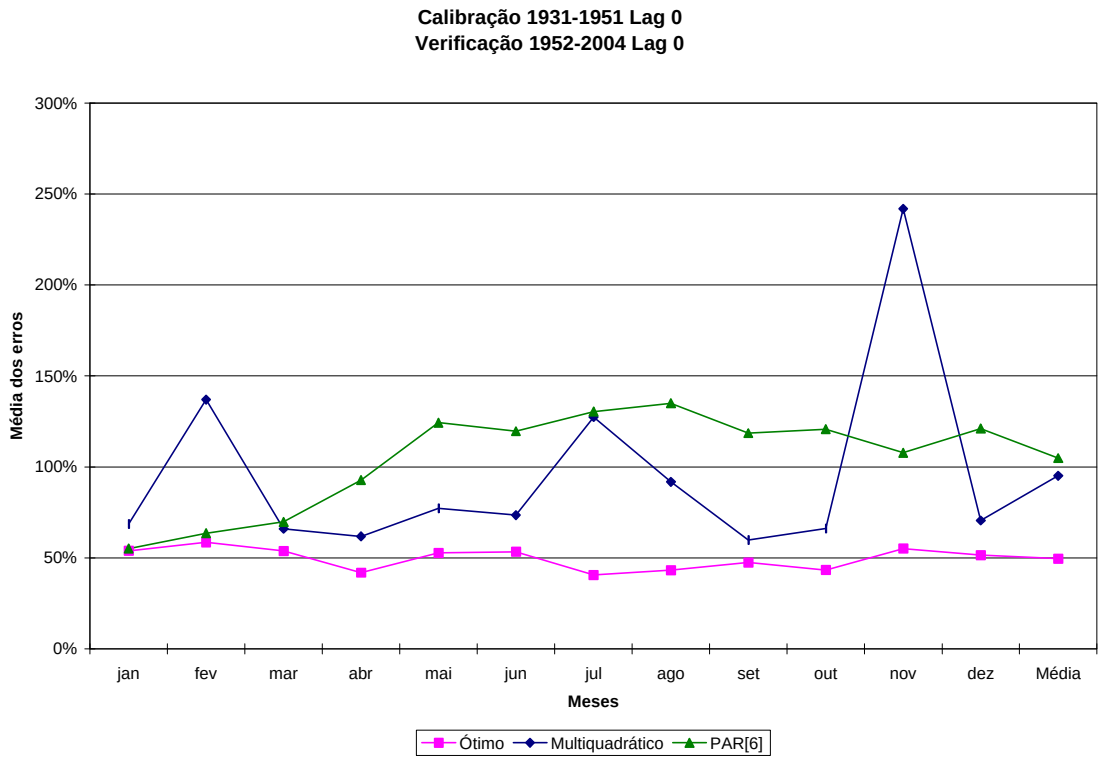


FIGURA C.2 Distância temporal 1 Calibração 1931-1961 lag 0 Verificação 1962-2004 lag 0.

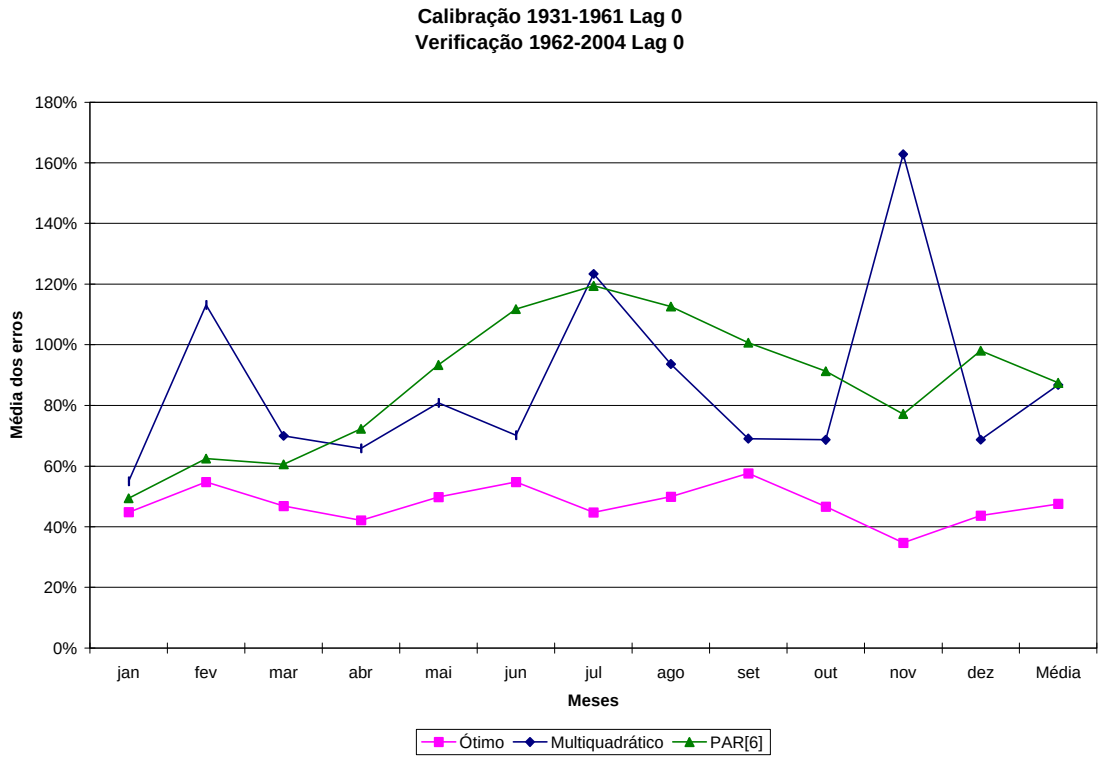


FIGURA C.3 Distância temporal 1 Calibração 1931-1971 lag 0 Verificação 1972-2004 lag 0.

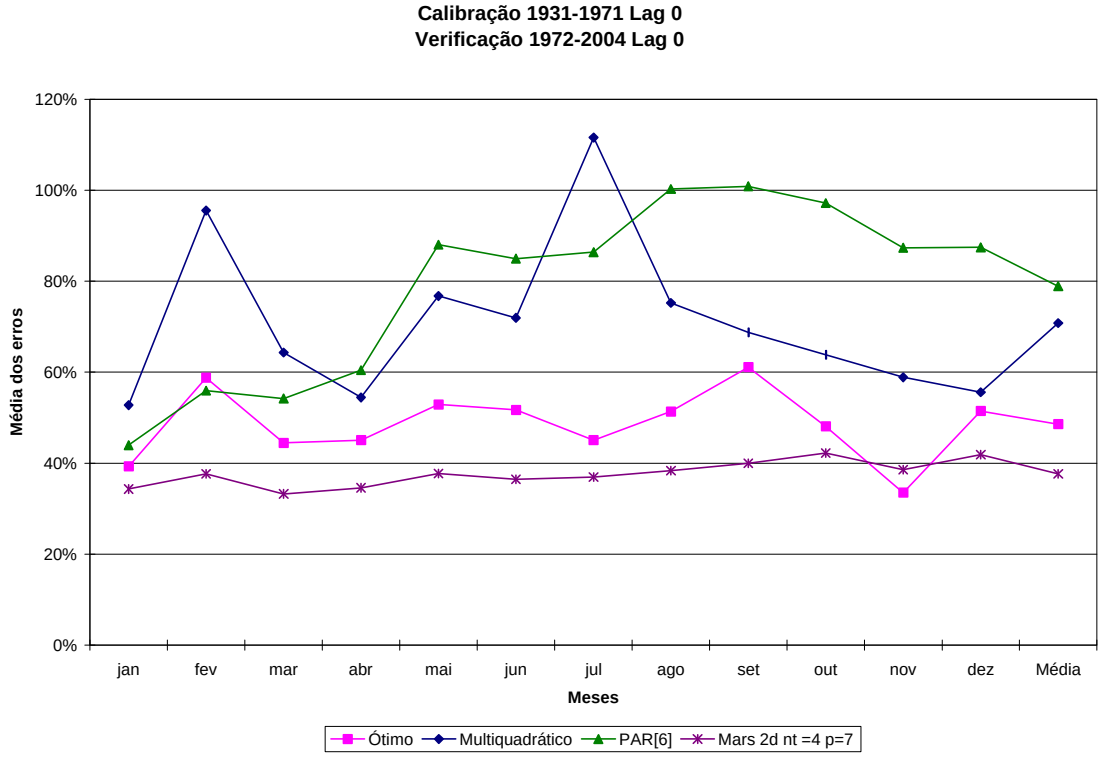


FIGURA C.4 Distância temporal 1 Calibração 1931-1971 lag 0 Verificação 1972-2004 lag 0 utilizando o modelo MARS.

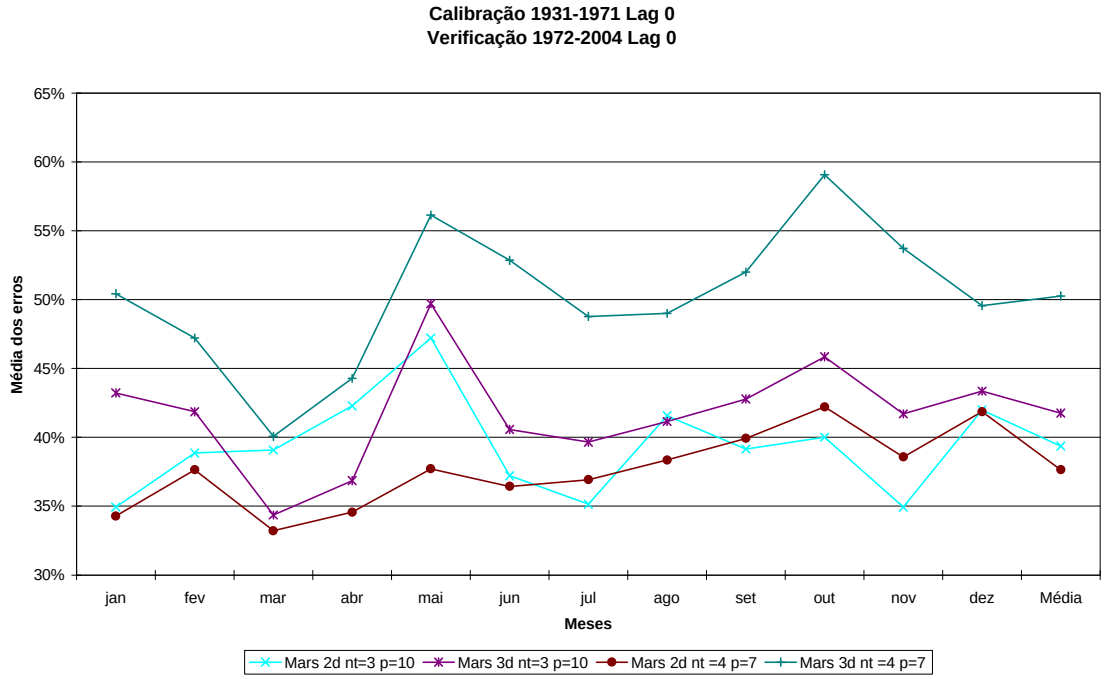


FIGURA C.5 Distância temporal 1 Calibração 1931-1981 lag 0 Verificação 1982-2004 lag 0.

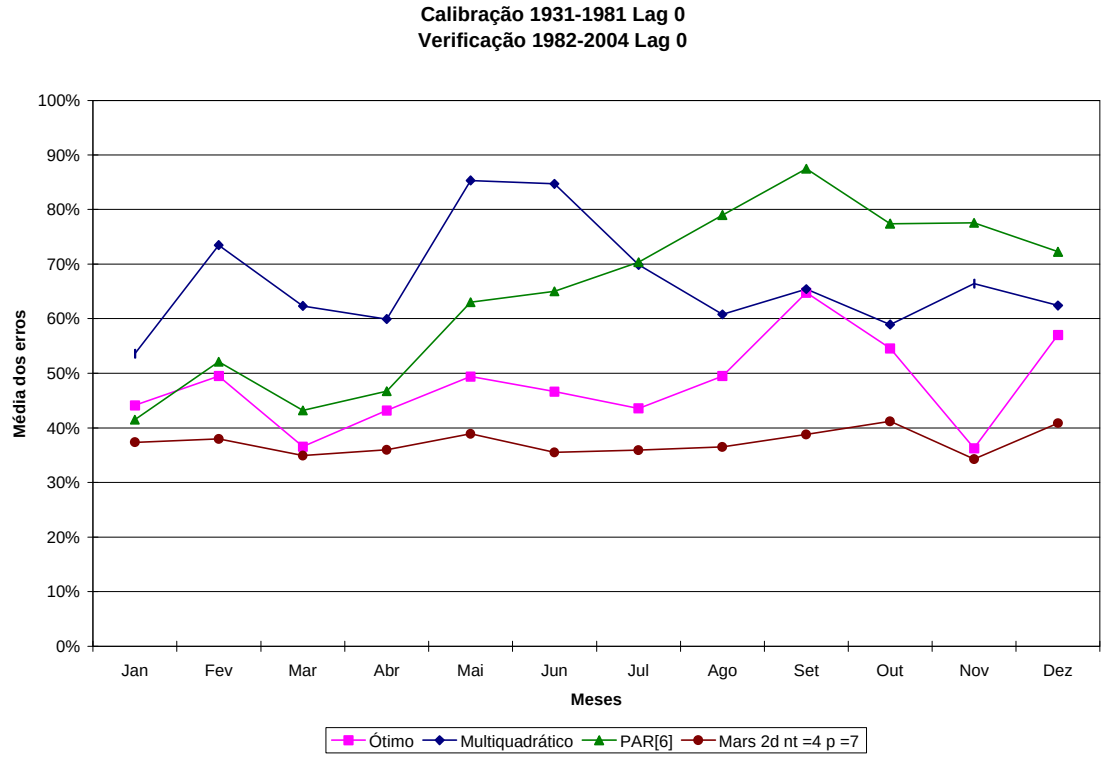
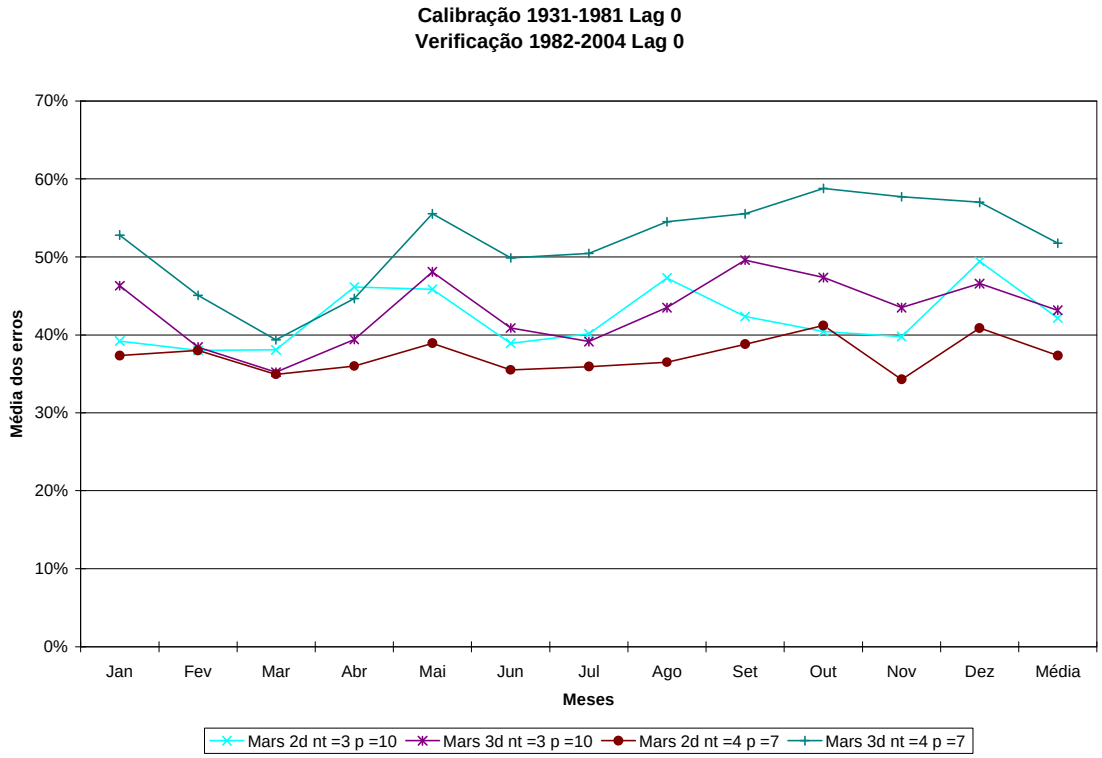


FIGURA C.6 Distância temporal 1 Calibração 1931-1981 lag 0 Verificação 1982-2004 lag 0 utilizando o modelo MARS.



C.1.2 Resultados obtidos em Lag 0 de calibração e Lag 1 de verificação

FIGURA C.7 Distância temporal 1 Calibração 1931-1951 lag 0 Verificação 1952-2004 lag 1.

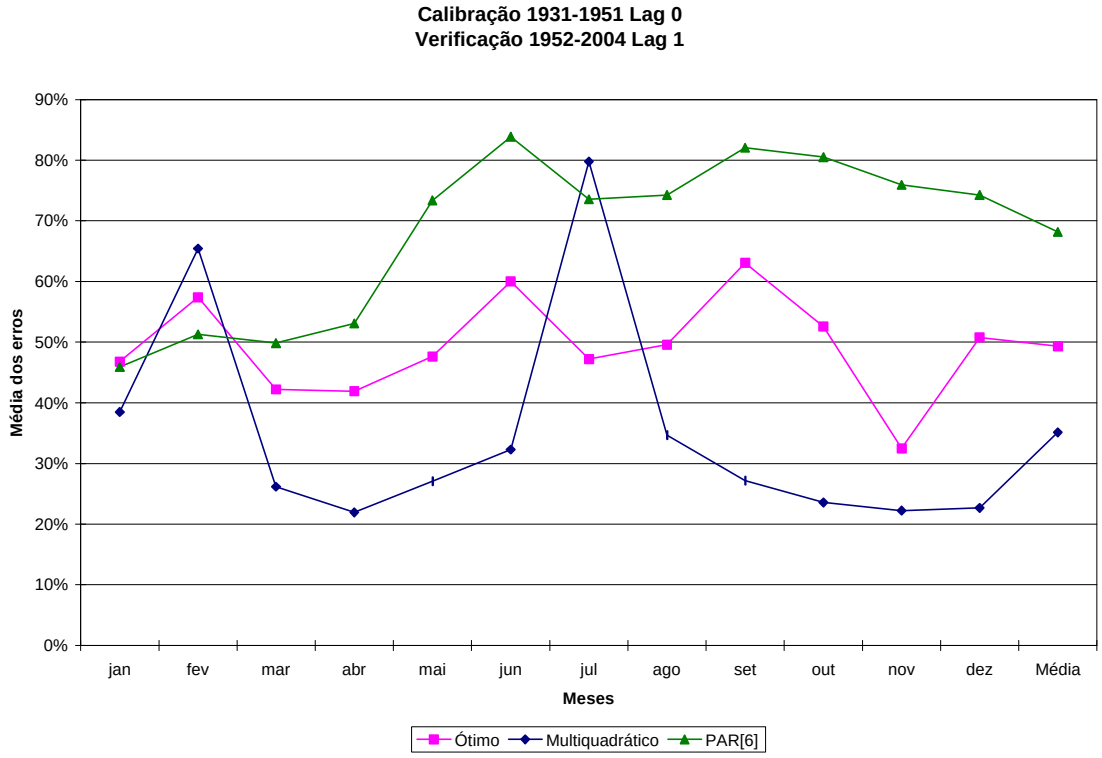


FIGURA C.8 Distância temporal 1 Calibração 1931-1961 lag 0 Verificação 1962-2004 lag 1.

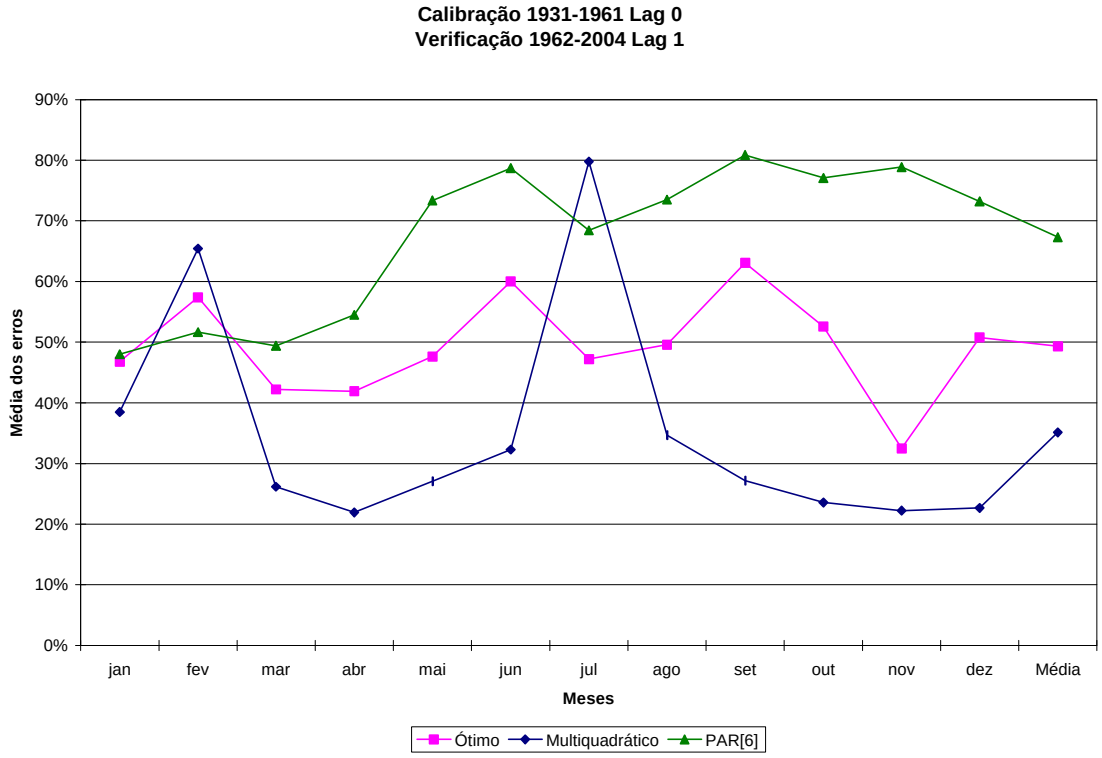


FIGURA C.9 Distância temporal 1 Calibração 1931-1971 lag 0 Verificação 1972-2004 lag 1.

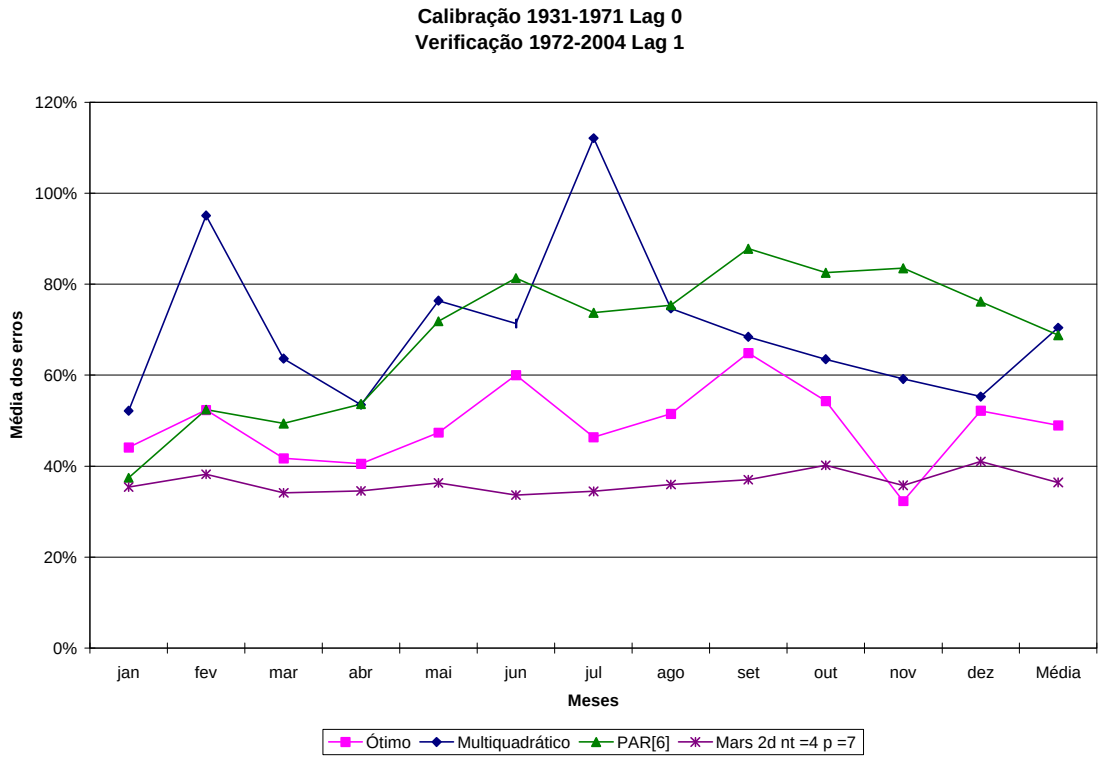


FIGURA C.10 Distância temporal 1 Calibração 1931-1971 lag 0 Verificação 1972-2004 lag 1 utilizando o modelo MARS.

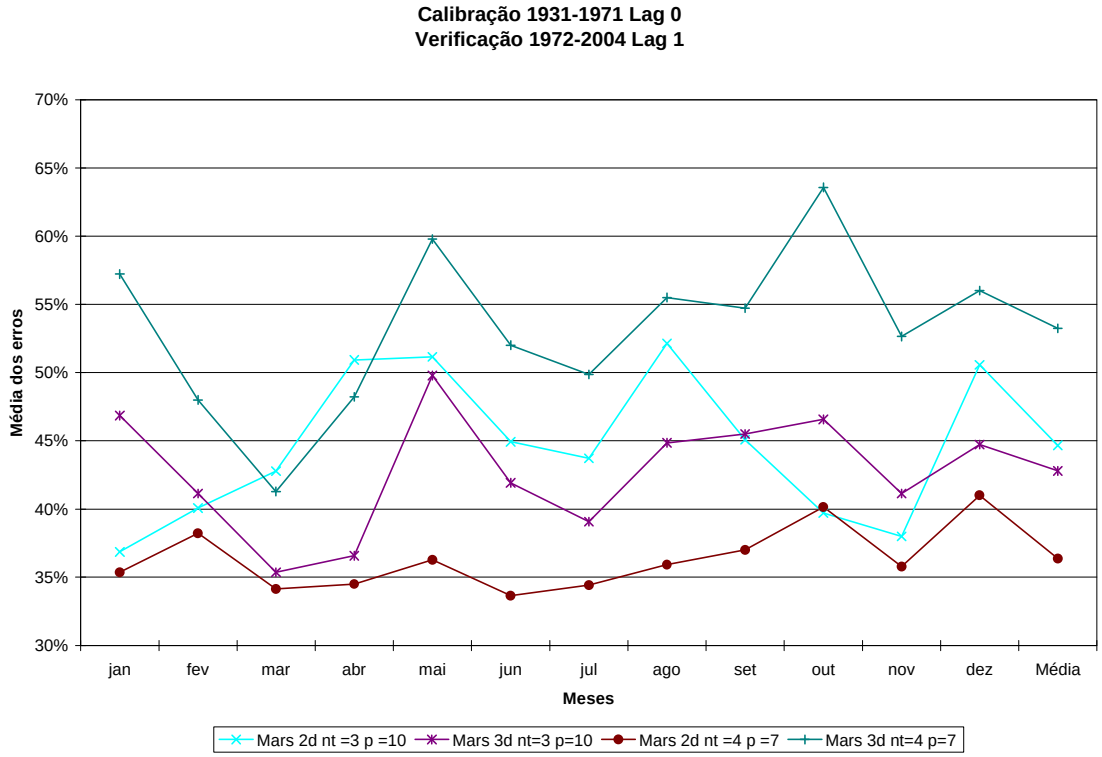


FIGURA C.11 Distância temporal 1 Calibração 1931-1981 lag 0 Verificação 1982-2004 lag 1.

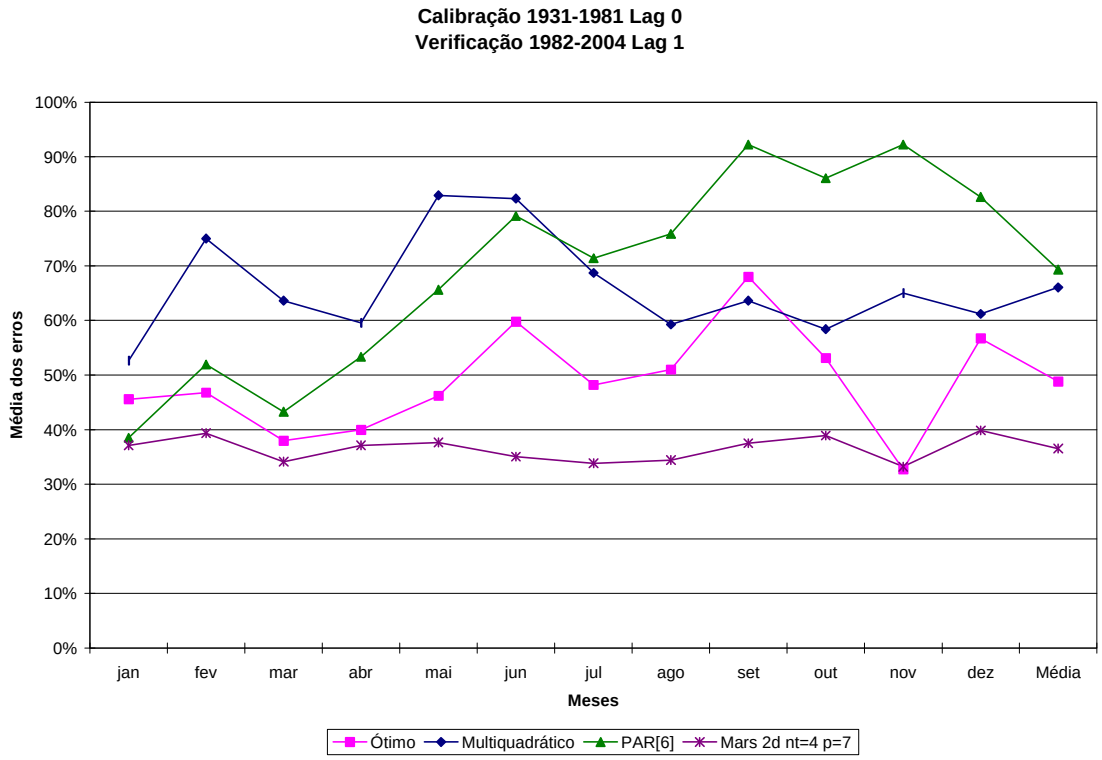
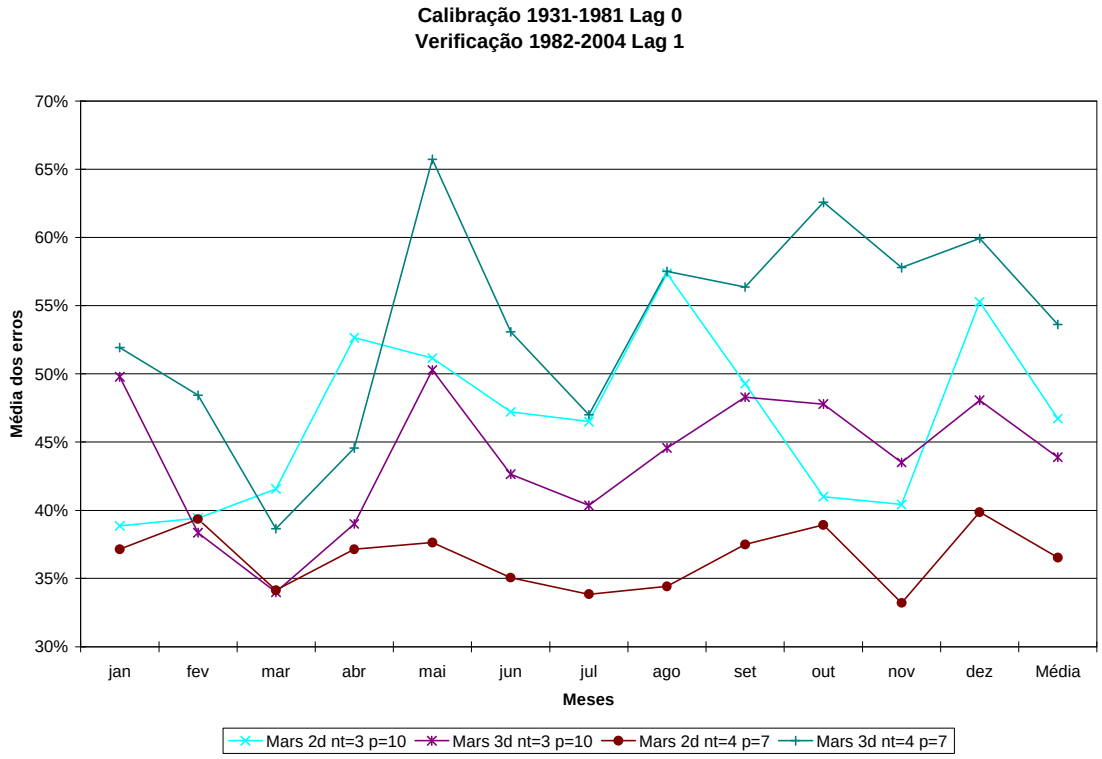


FIGURA C.12 Distância temporal 1 Calibração 1931-1981 lag 0 Verificação 1982-2004 lag 1 utilizando o modelo MARS.



C.1.3 Resultados obtidos em Lag 1 de calibração e Lag 1 de verificação

FIGURA C.13 Distância temporal 1 Calibração 1931-1951 lag 1 Verificação 1952-2004 lag 1.

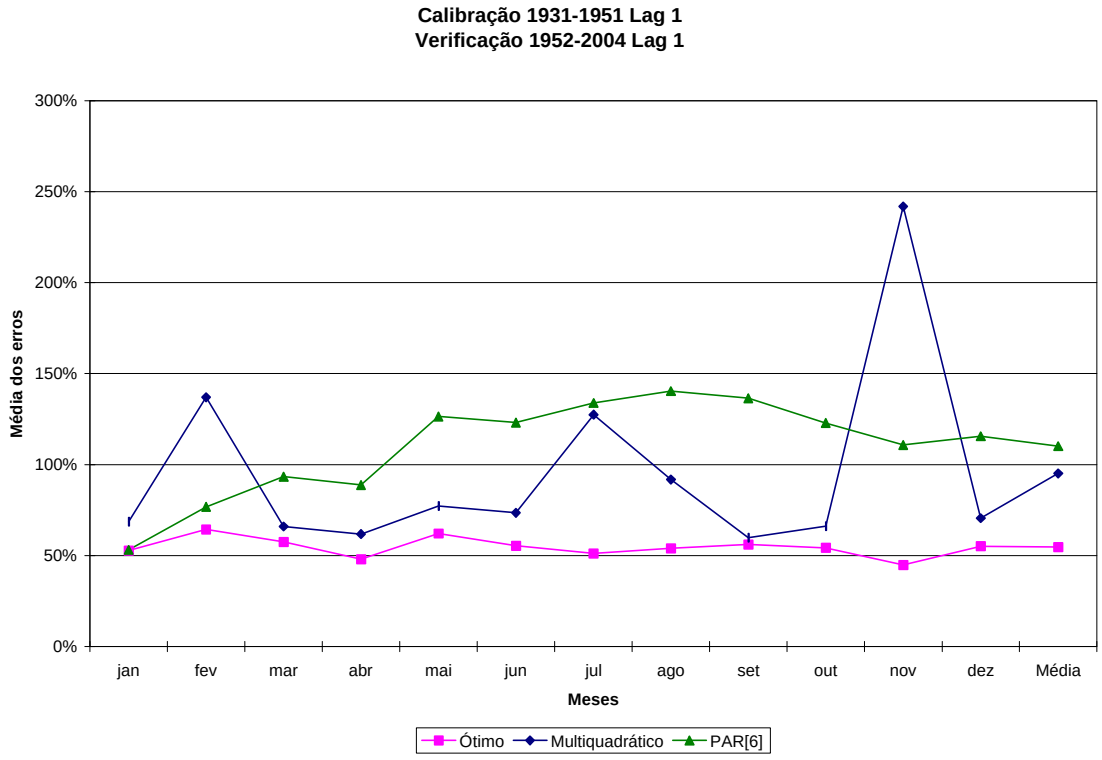


FIGURA C.14 Distância temporal 1 Calibração 1931-1961 lag 1 Verificação 1962-2004 lag 1.

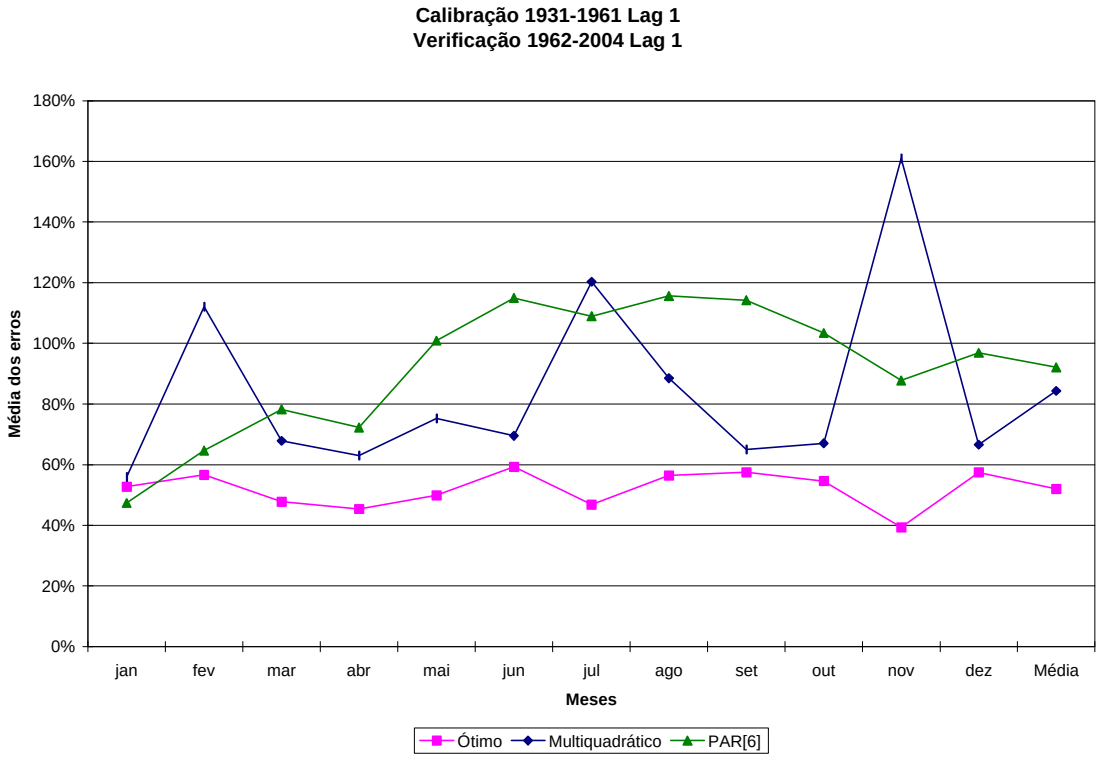


FIGURA C.15 Distância temporal 1 Calibração 1931-1971 lag 1 Verificação 1972-2004 lag 1.

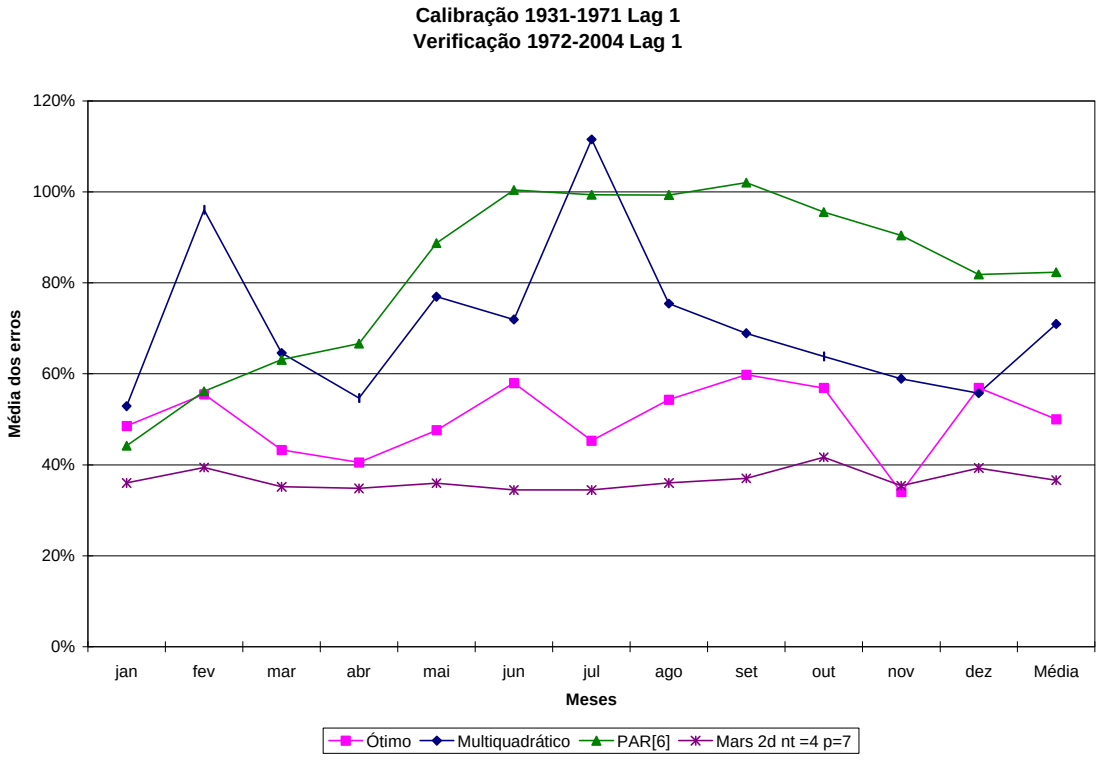


FIGURA C.16 Distância temporal 1 Calibração 1931-1971 lag 1 Verificação 1972-2004 lag 1 utilizando o modelo MARS.

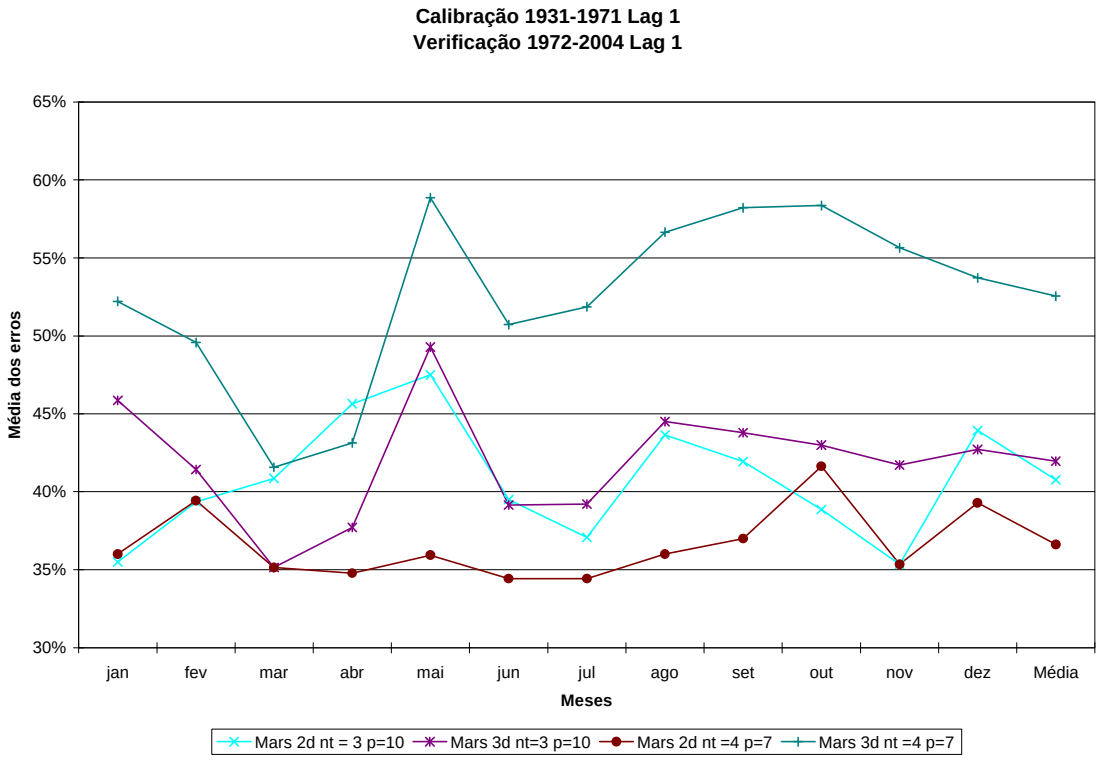


FIGURA C.17 Distância temporal 1 Calibração 1931-1981 lag 1 Verificação 1982-2004 lag 1.

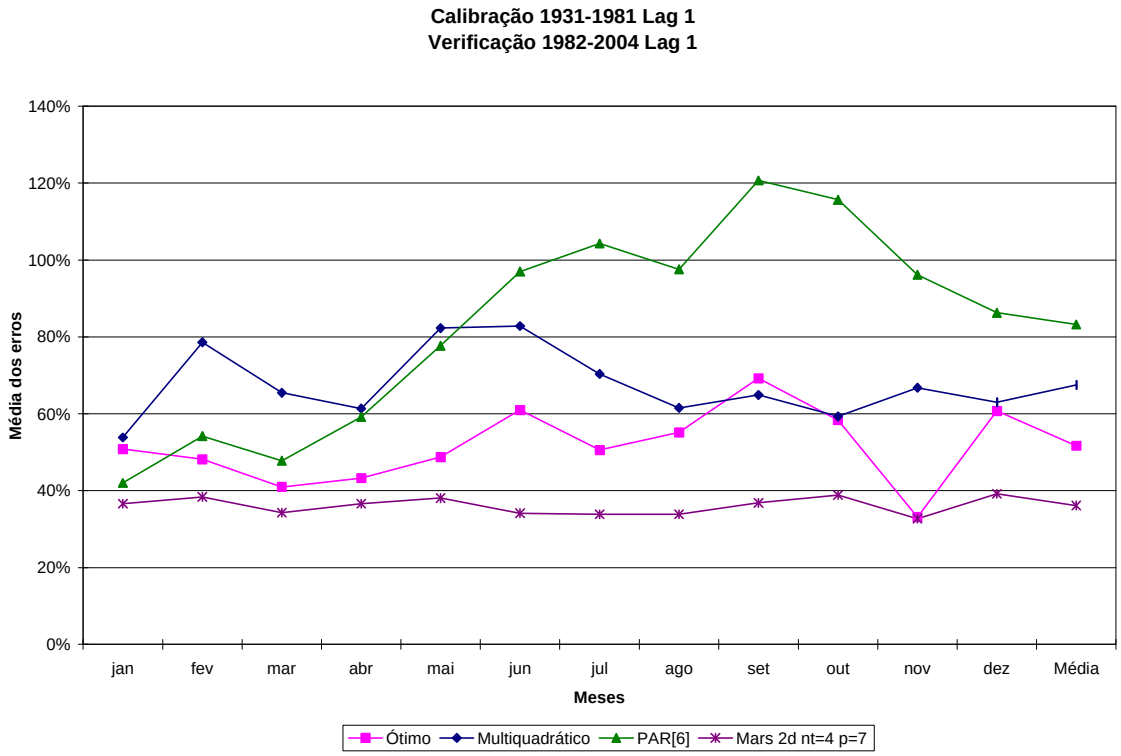
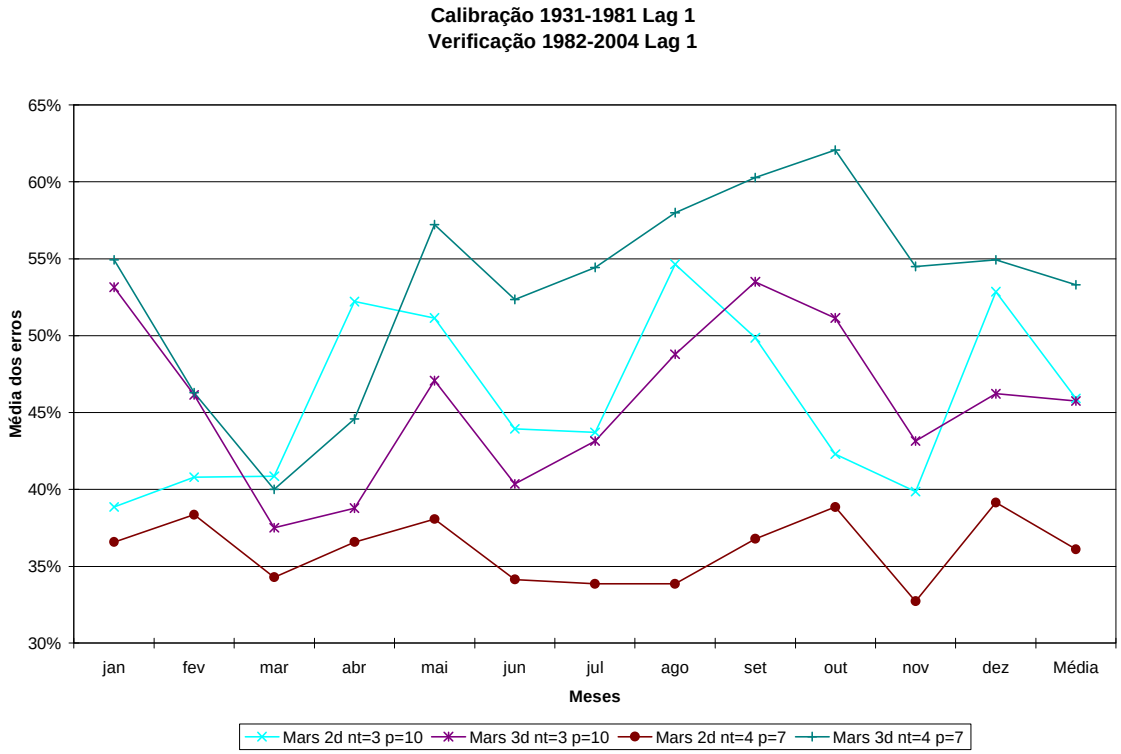


FIGURA C.18 Distância temporal 1 Calibração 1931-1981 lag 1 Verificação 1982-2004 lag 1 utilizando o modelo MARS.



C.2 RESULTADOS OBTIDOS COM INTERVALO DE CINCO ANOS

Este apêndice tem como finalidade apresentar os resultados obtidos pelo programa de previsão quando o intervalo entre o último ano de calibração e o primeiro ano de verificação é igual a 5, utilizando a metodologia descrita no item 4. Sendo que as previsões utilizando o modelo MARS foram analisadas a partir de 1971, compreendendo um período de calibração mínimo de 40 anos.

C.2.1 Resultados obtidos em Lag 0 de calibração e Lag 0 de verificação

FIGURA C.19 Distância temporal 5 Calibração 1931-1951 lag 0 Verificação 1956-2004 lag 0.

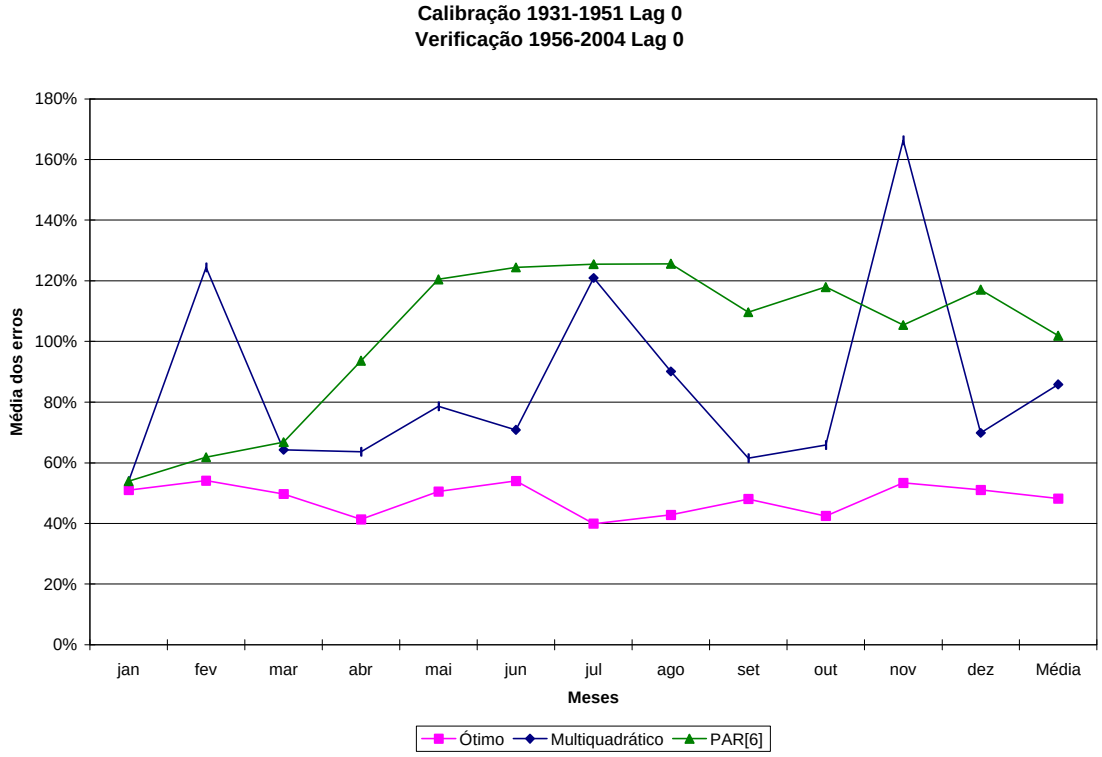


FIGURA C.20 Distância temporal 5 Calibração 1931-1961 lag 0 Verificação 1966-2004 lag 0.

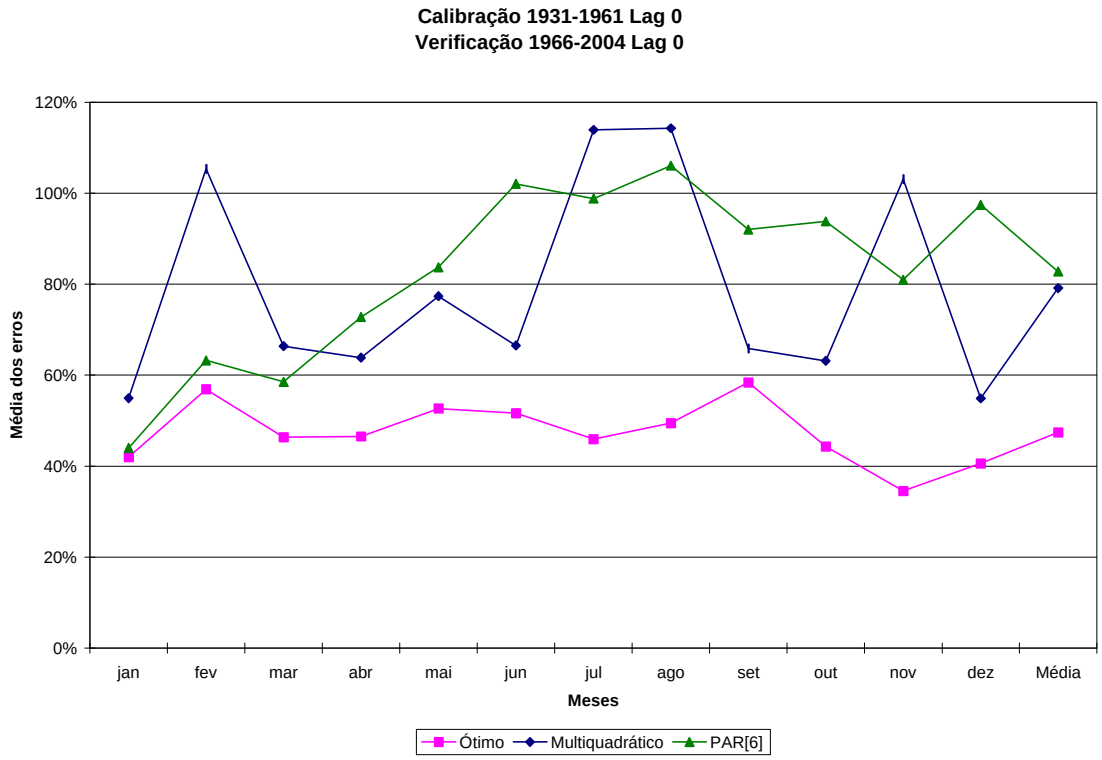


FIGURA C.21 Distância temporal 5 Calibração 1931-1971 lag 0 Verificação 1976-2004 lag 0.

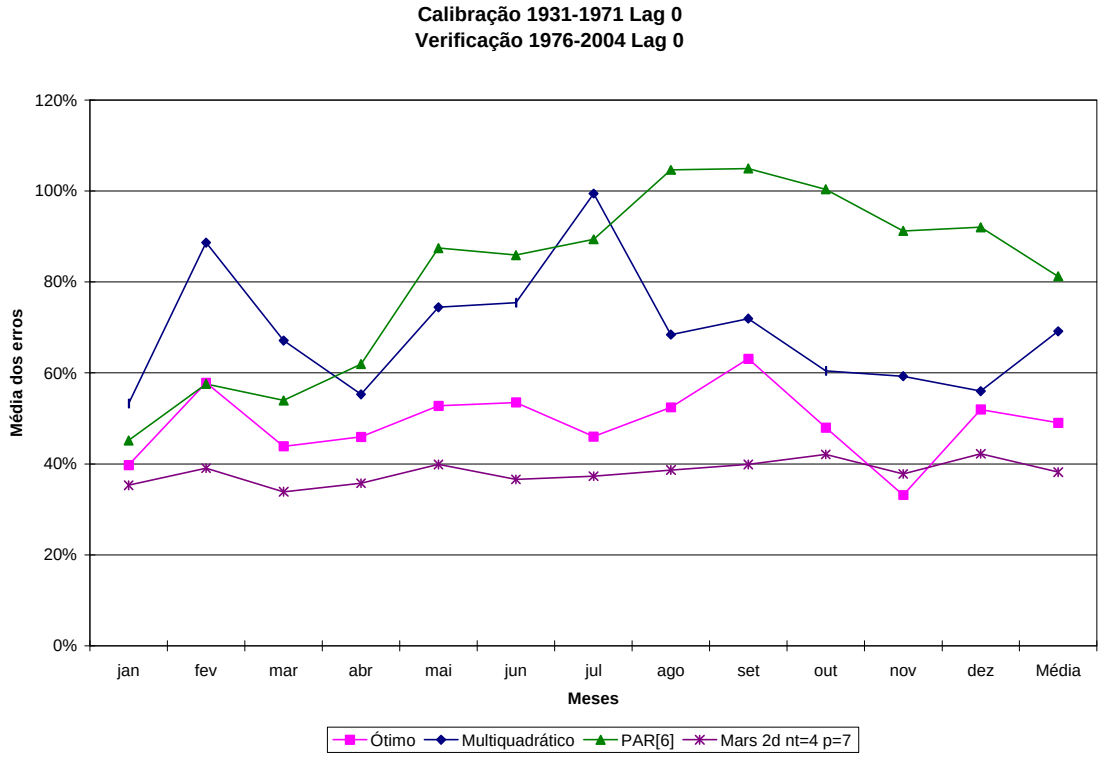


FIGURA C.22 Distância temporal 5 Calibração 1931-1971 lag 0 Verificação 1976-2004 lag 0 utilizando o modelo MARS.

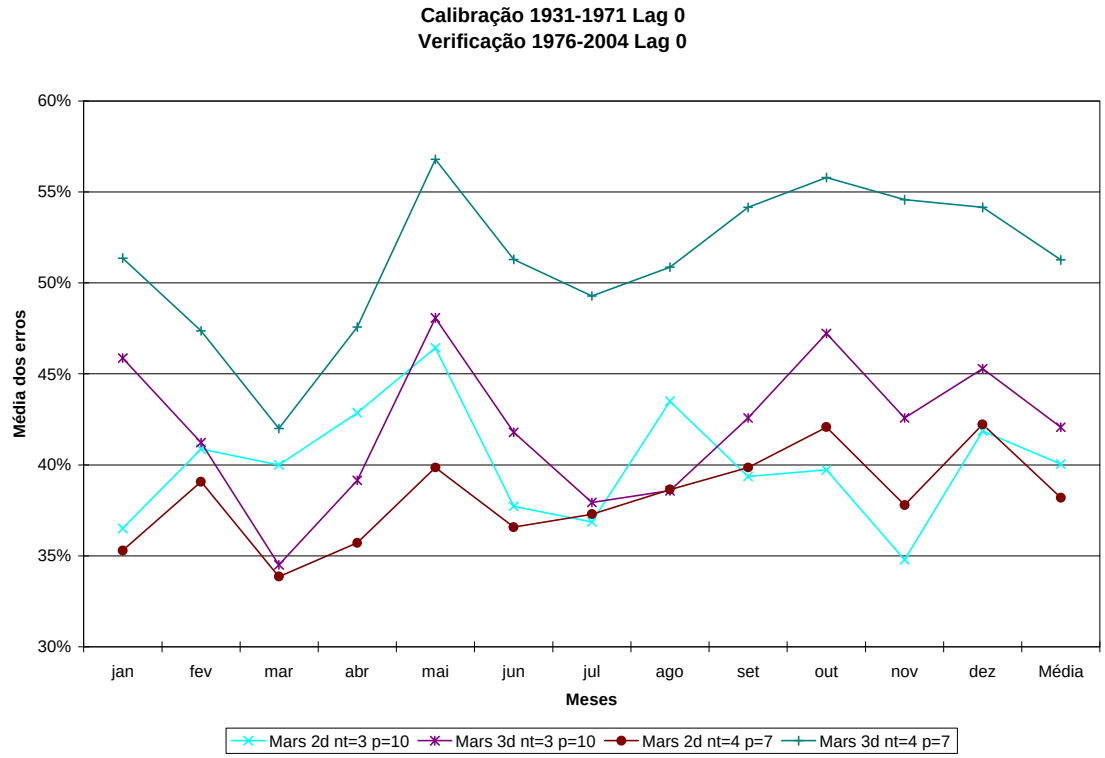


FIGURA C.23 Distância temporal 5 Calibração 1931-1981 lag 0 Verificação 1986-2004 lag 0.

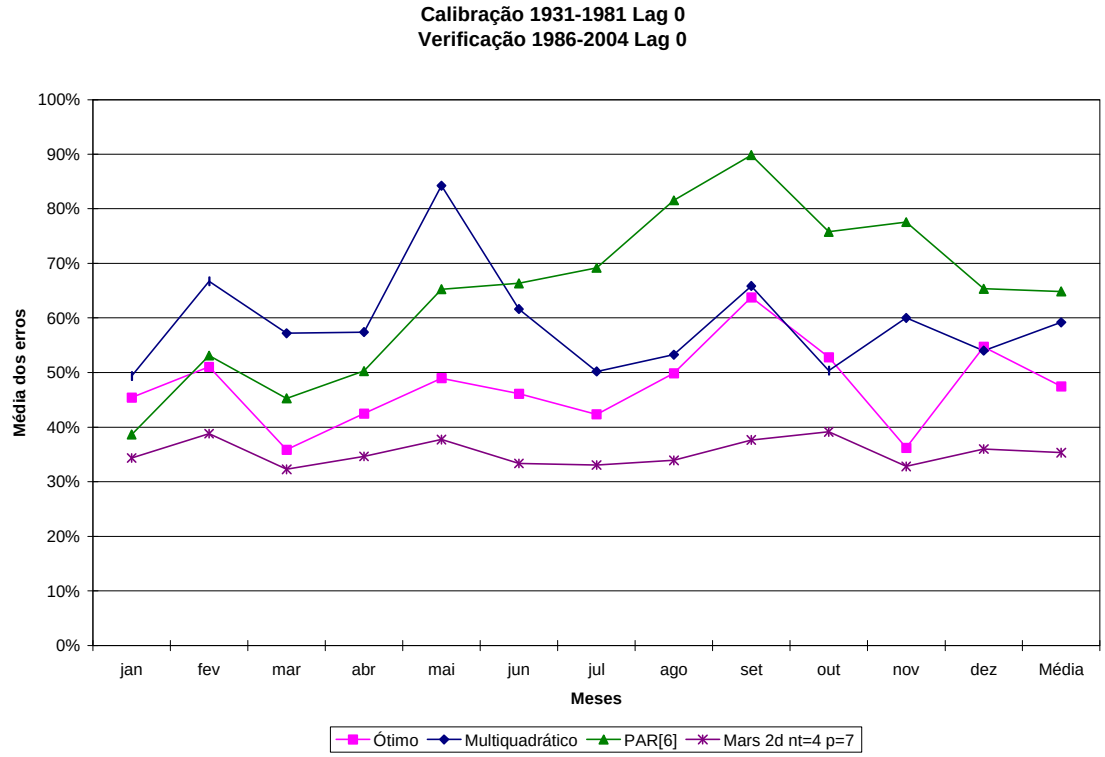
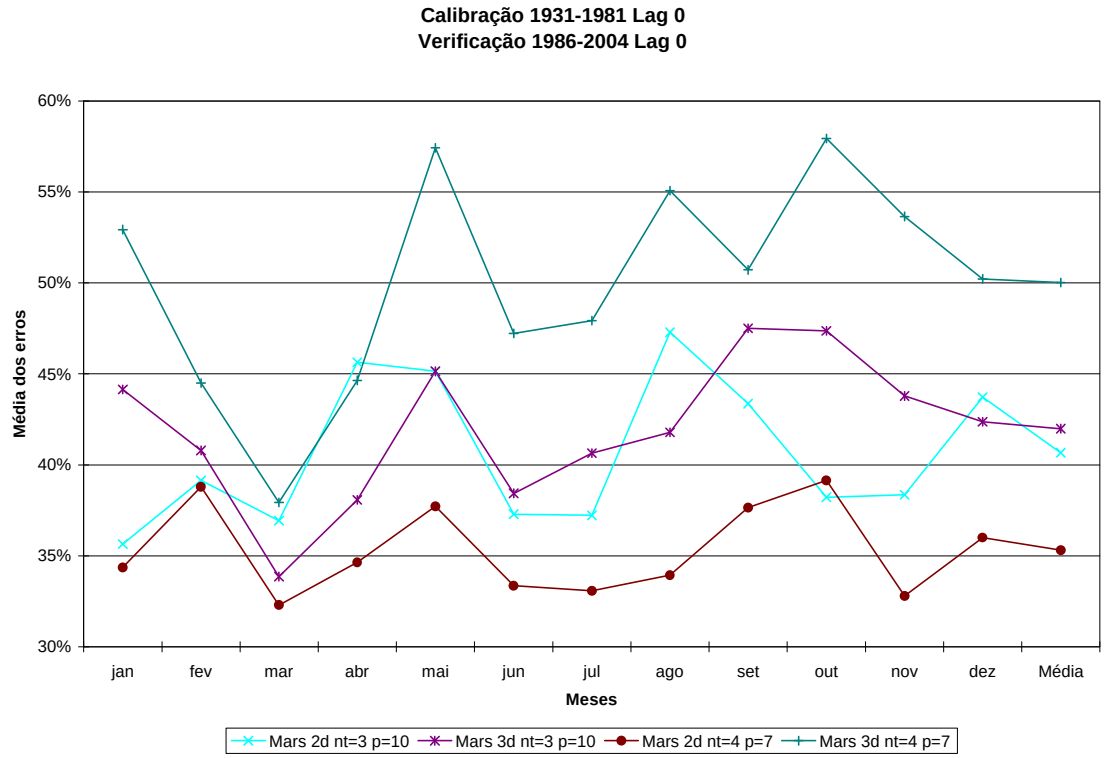


FIGURA C.24 Distância temporal 5 Calibração 1931-1981 lag 0 Verificação 1986-2004 lag 0 utilizando o modelo MARS.



C.2.2 Resultados obtidos em Lag 0 de calibração e Lag 1 de verificação

FIGURA C.25 Distância temporal 5 Calibração 1931-1951 lag 0 Verificação 1956-2004 lag 1.

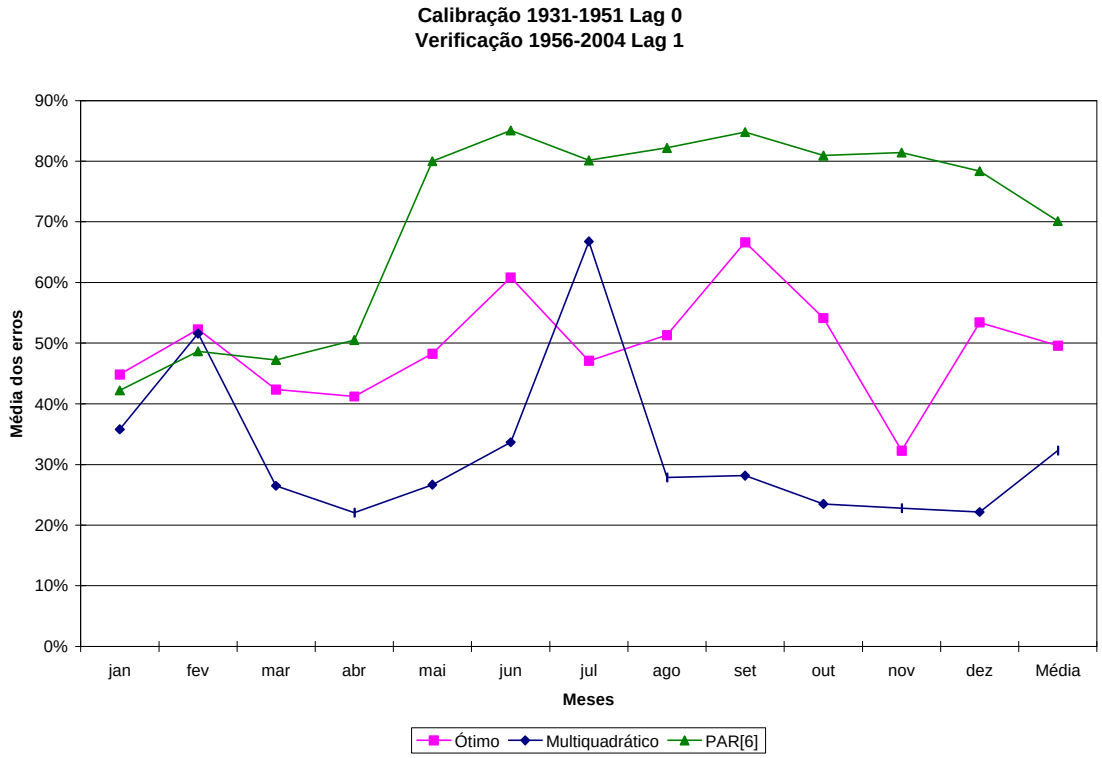


FIGURA C.26 Distância temporal 5 Calibração 1931-1961 lag 0 Verificação 1966-2004 lag 1.

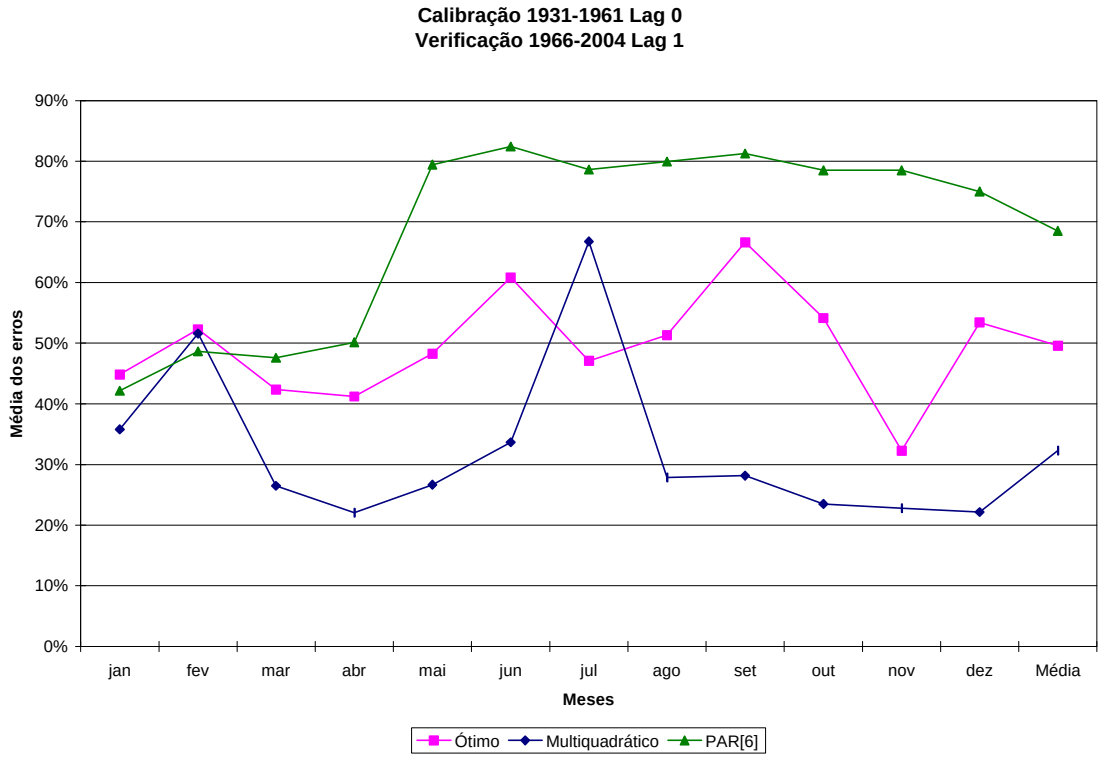


FIGURA C.27 Distância temporal 5 Calibração 1931-1971 lag 0 Verificação 1976-2004 lag 1.

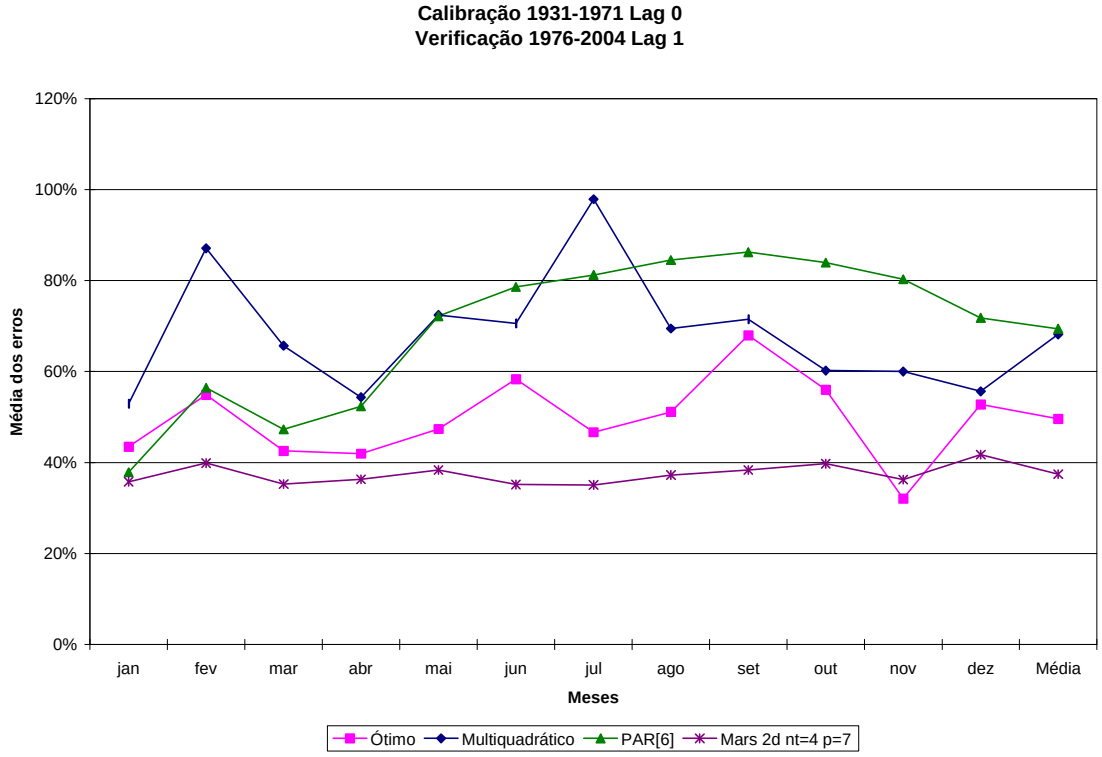


FIGURA C.28 Distância temporal 5 Calibração 1931-1971 lag 0 Verificação 1976-2004 lag 1 utilizando o modelo MARS.

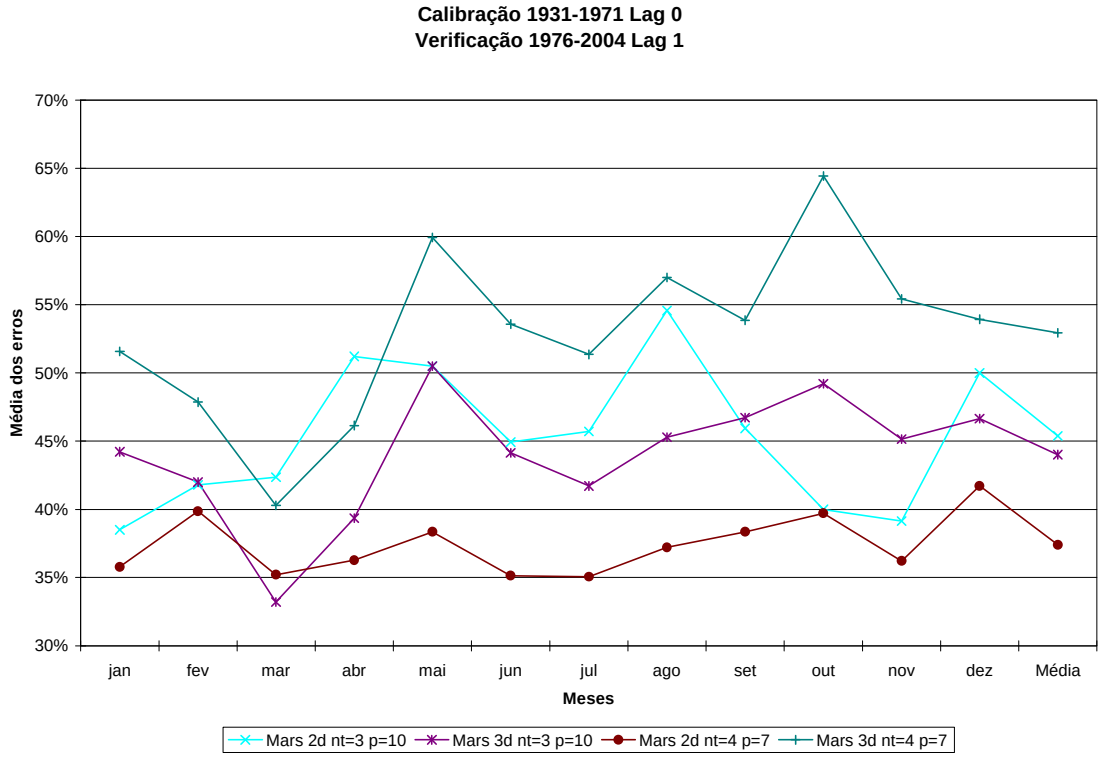


FIGURA C.29 Distância temporal 5 Calibração 1931-1981 lag 0 Verificação 1986-2004 lag 1.

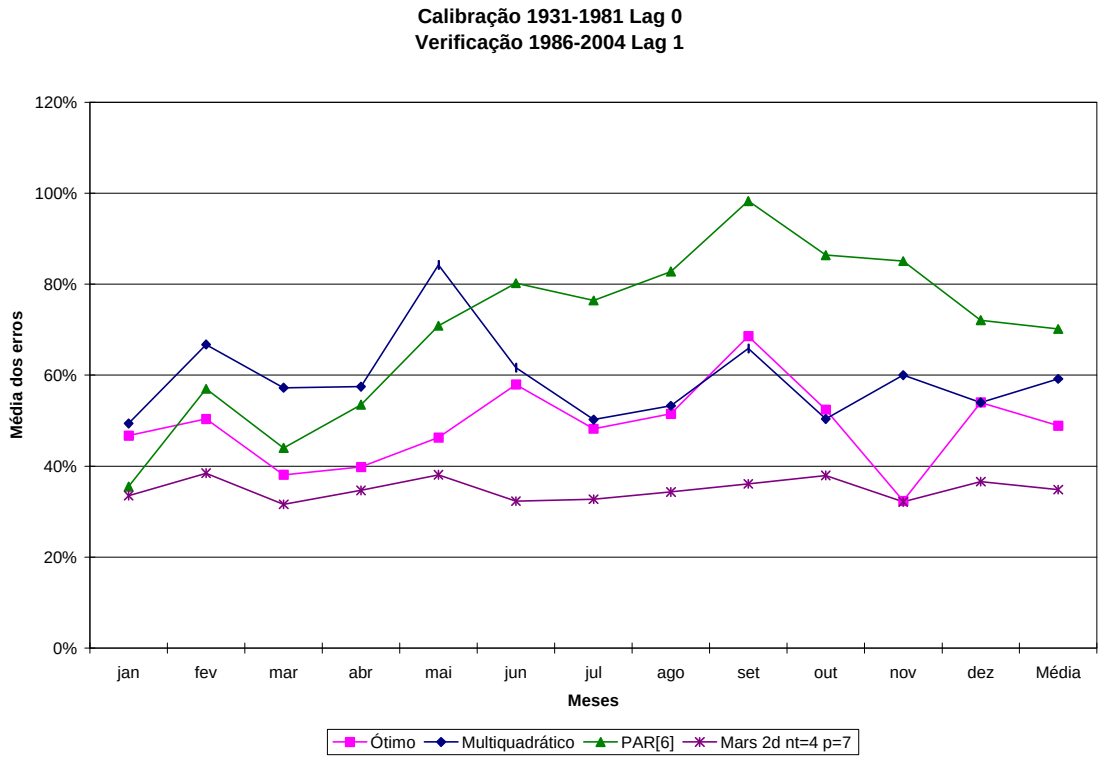
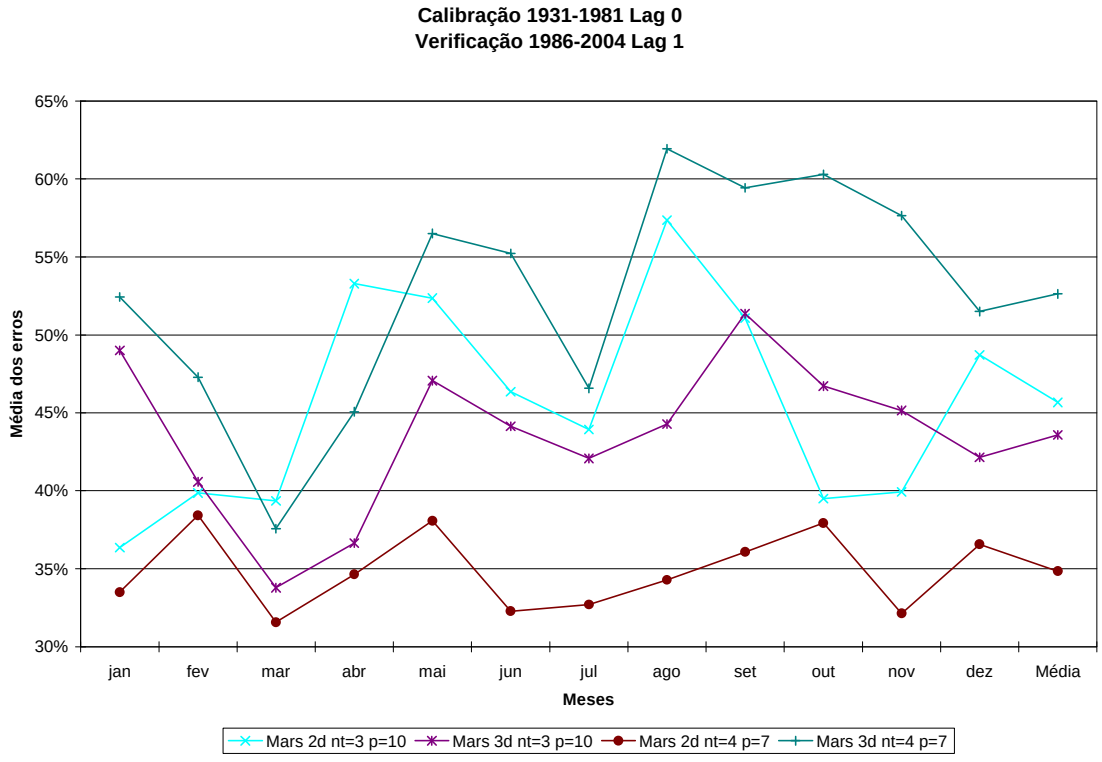


FIGURA C.30 Distância temporal 5 Calibração 1931-1981 lag 0 Verificação 1986-2004 lag 1 utilizando o modelo MARS.



C.2.3 Resultados obtidos em Lag 1 de calibração e Lag 1 de verificação

FIGURA C.31 Distância temporal 5 Calibração 1931-1951 lag 1 Verificação 1956-2004 lag 1.

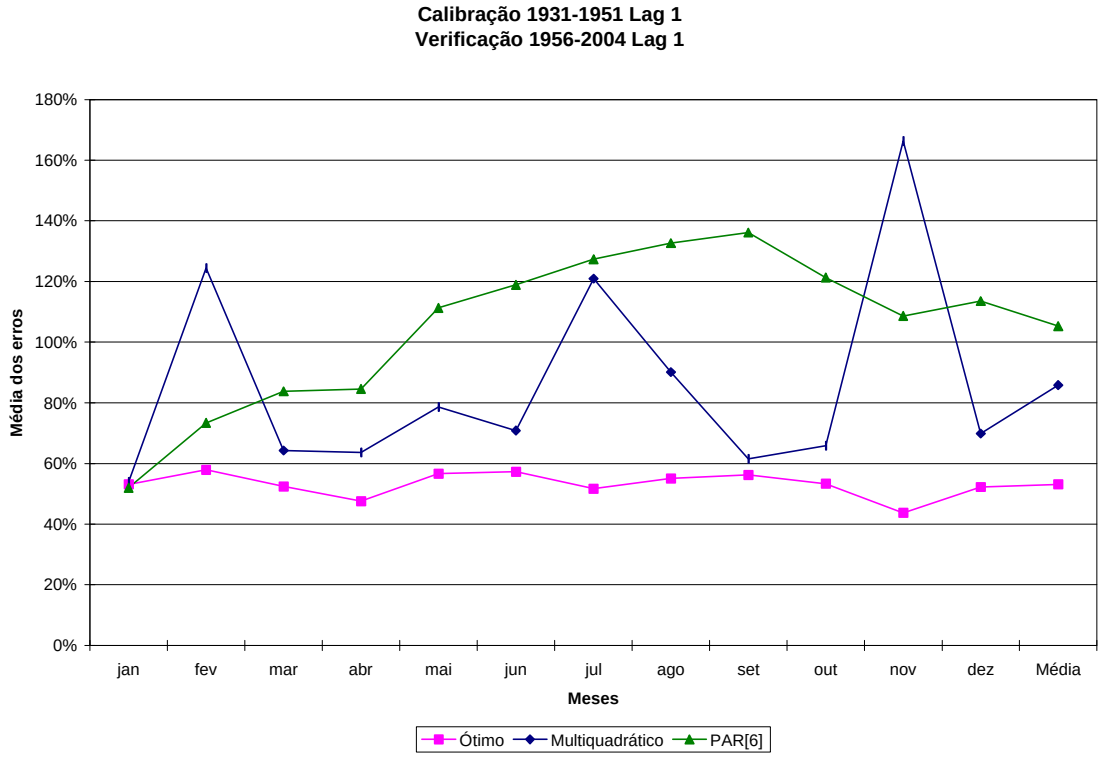


FIGURA C.32 Distância temporal 5 Calibração 1931-1961 lag 1 Verificação 1966-2004 lag 1.

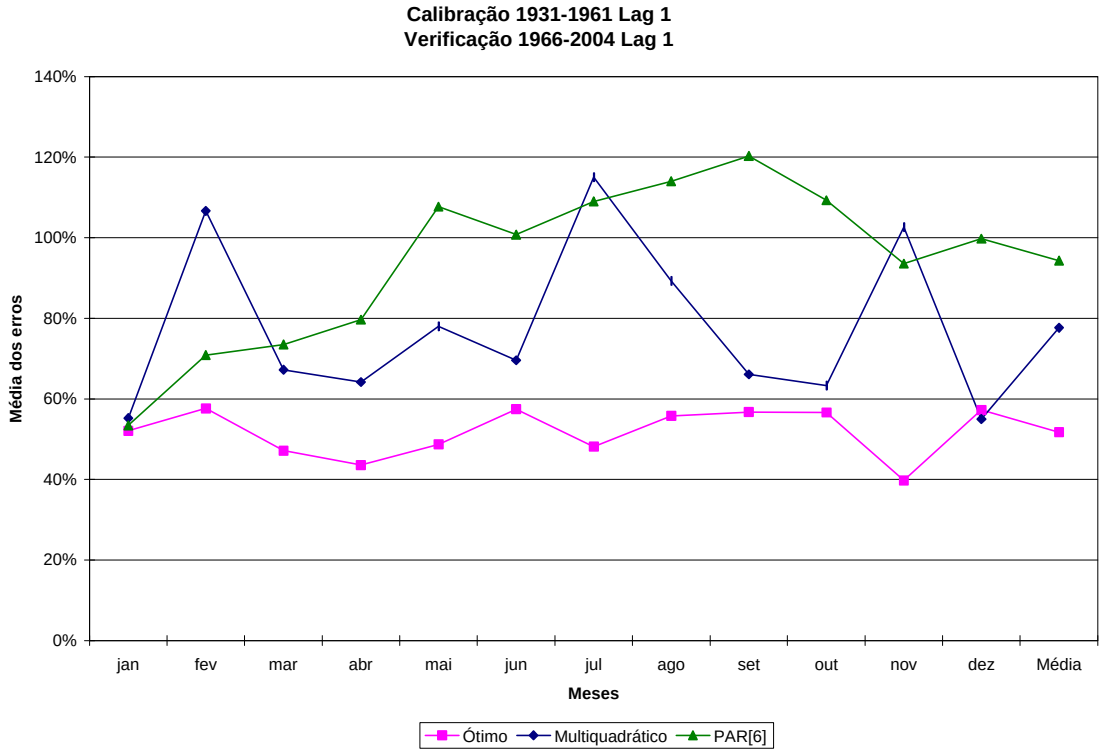


FIGURA C.33 Distância temporal 5 Calibração 1931-1971 lag 1 Verificação 1976-2004 lag 1.

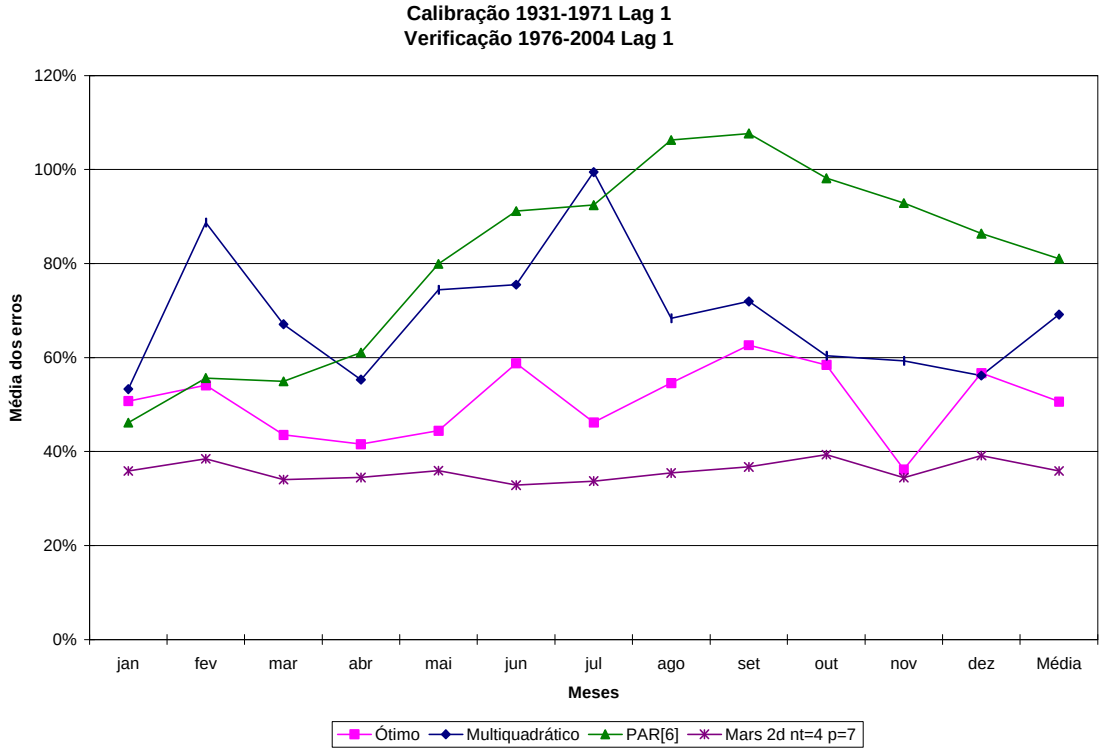


FIGURA C.34 Distância temporal 5 Calibração 1931-1971 lag 1 Verificação 1976-2004 lag 1 utilizando o modelo MARS.

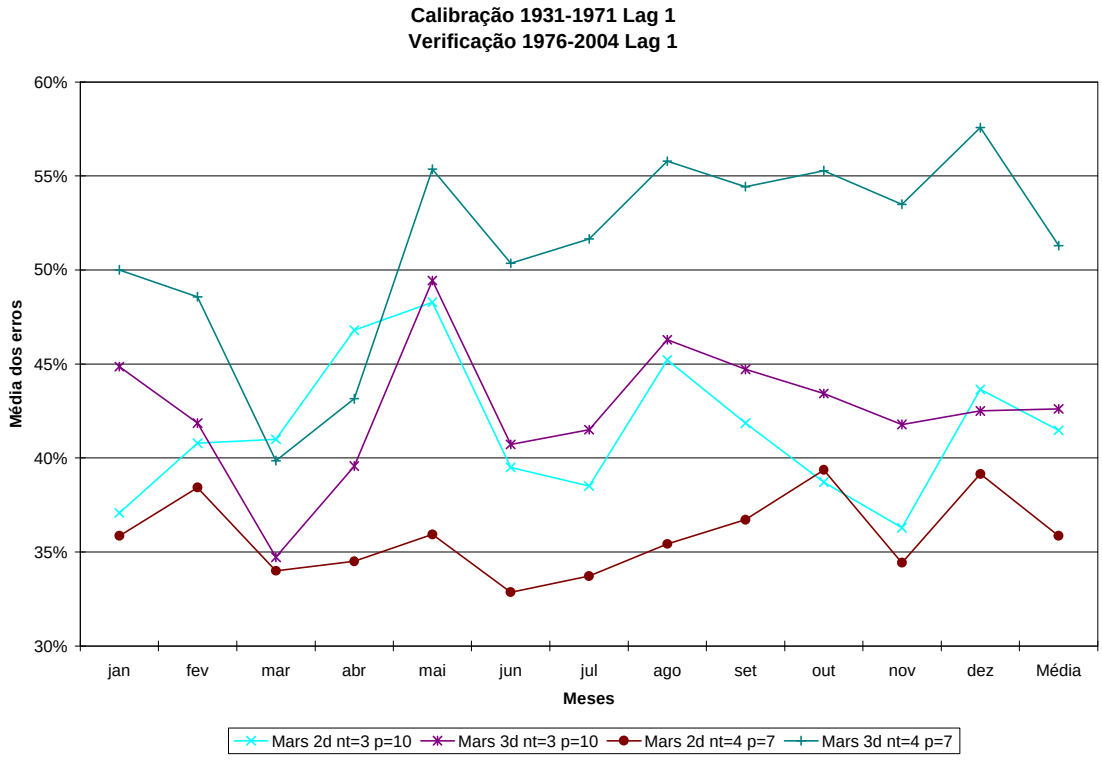


FIGURA C.35 Distância temporal 5 Calibração 1931-1981 lag 1 Verificação 1986-2004 lag 1.

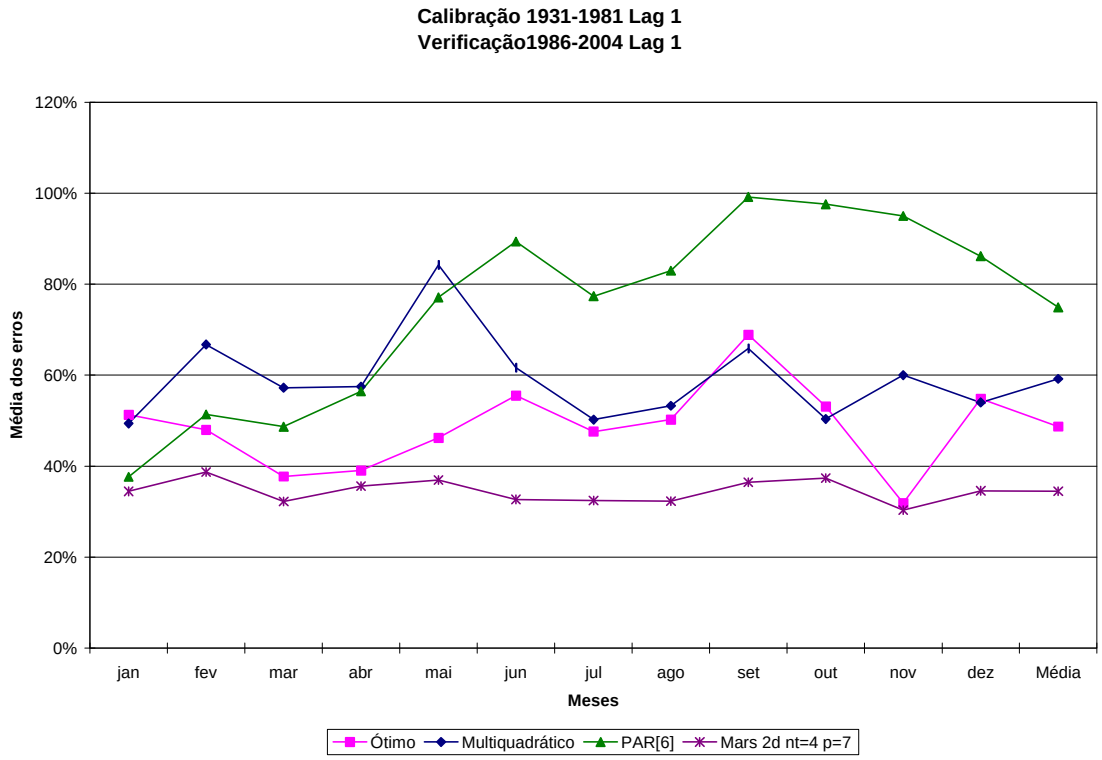
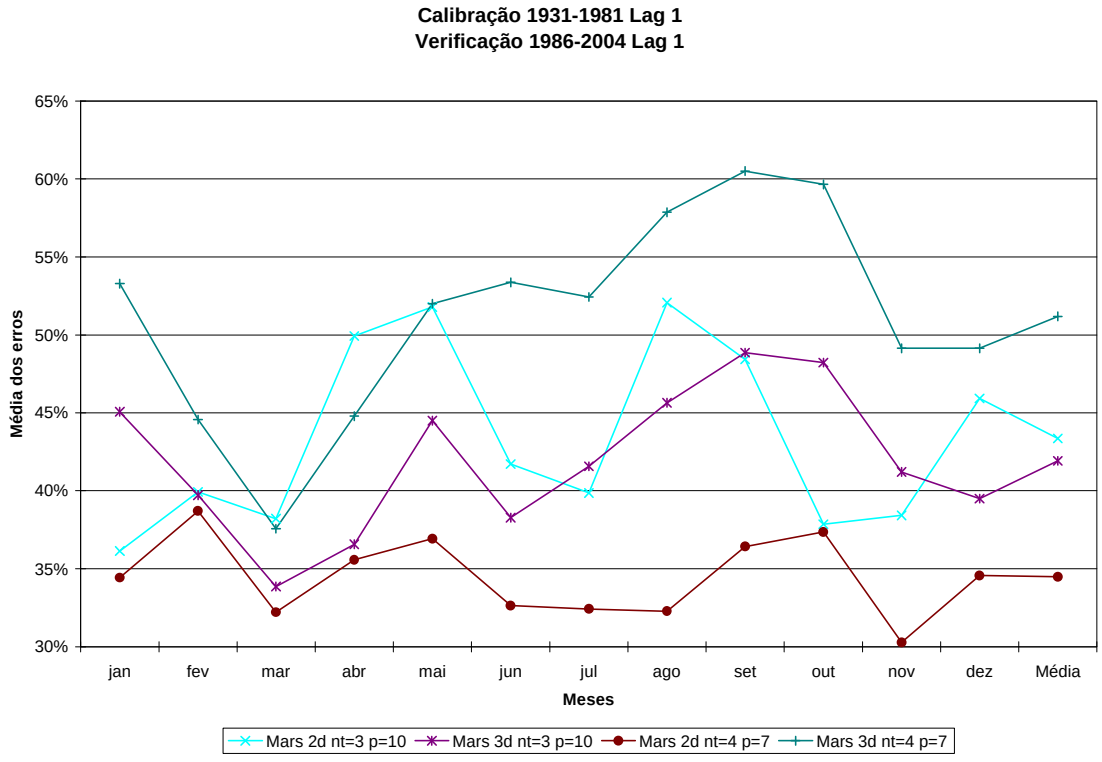


FIGURA C.36 Distância temporal 5 Calibração 1931-1981 lag 1 Verificação 1986-2004 lag 1 utilizando o modelo MARS.



APÊNDICE D ANÁLISE DA PREVISÃO DE VAZÕES POR USINA

Neste apêndice será ilustrado os resultados obtidos pelos modelos de previsão de vazões médias mensais no horizonte de um a doze meses para cada uma das 14 usinas analisadas. Em cada tabela será ilustrado os resultados médios para o período analisado. A análise que será apresentada neste apêndice é para Lag 0 de calibração e Lag 0 de verificação, com intervalo de um ano. A descrição do significado de Lag de calibração e do Lag de verificação encontra-se no capítulo 4 desta dissertação.

As tabelas são para distância temporal 1, período de calibração de 1931-1981 em lag 0 e período de verificação de 1982-2004 em lag 0.

Além do erro médio é apresentado o coeficiente de correlação *r* para cada mês e para cada modelo de previsão. O coeficiente de correlação é dado por:

$$r = \frac{\sum (x_i - \bar{x}) \times (y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \times \sum (y_i - \bar{y})^2}} \qquad \text{(D.1)}$$

Onde: x_i é a vazão mensal prevista para o ano *i,* $\bar{x}$ é a média da vazão mensal prevista, y_i é a vazão mensal observada para o ano *i* e $\bar{y}$ é a média da vazão mensal observada.

Tabela D.1 Resultados obtidos para a usina de Serra da Mesa (20920080)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.33	-0.39	0.48	-0.10	0.42	0.27	0.34	0.15	0.40	0.47	0.34	0.26	0.57	0.14
fevereiro	0.53	-0.07	0.99	-0.17	0.90	0.13	0.41	-0.27	0.61	-0.05	0.44	-0.21	0.54	-0.01
março	0.42	-0.32	0.68	-0.34	0.54	0.16	0.44	-0.31	0.38	-0.01	0.37	-0.32	0.40	0.24
abril	0.44	0.14	0.67	-0.22	0.69	0.11	0.69	-0.41	0.52	-0.02	0.44	-0.31	0.48	0.24
maio	0.54	0.34	0.48	0.12	0.46	0.48	0.35	0.05	0.38	0.06	0.30	0.10	0.51	-0.06
junho	0.44	0.40	0.49	0.25	0.70	0.40	0.42	0.00	0.64	-0.25	0.34	0.12	0.79	-0.05
julho	0.42	0.41	0.51	0.07	0.60	0.20	0.49	-0.07	0.47	0.23	0.36	-0.17	0.75	0.31
agosto	0.48	0.57	0.54	0.19	1.32	0.16	0.58	-0.06	0.63	0.22	0.40	0.09	1.27	-0.16
setembro	0.47	0.47	0.37	0.46	0.69	0.41	0.47	0.06	0.89	-0.59	0.29	0.41	0.91	-0.50
outubro	0.49	0.56	0.33	0.39	0.73	0.38	0.46	0.09	0.63	-0.11	0.32	0.20	1.02	0.04
novembro	0.26	-0.05	0.53	-0.13	1.28	0.13	0.29	0.00	0.56	-0.22	0.25	0.06	0.68	0.24
dezembro	0.40	-0.22	0.81	-0.27	1.21	0.16	0.42	0.15	0.60	-0.10	0.41	-0.27	0.54	0.01
média	0.44	0.11	0.57	0.07	0.78	0.08	0.45	0.05	0.54	0.07	0.36	0.07	0.70	0.06

Tabela D.2 Resultados obtidos para a usina de Tucuruí (29680080)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.25	-0.15	0.33	0.32	0.19	0.81	0.19	0.78	0.34	0.40	0.19	0.81	0.40	0.44
fevereiro	0.41	-0.10	0.48	-0.22	0.61	-0.01	0.28	0.16	0.27	0.35	0.28	0.00	0.30	0.48
março	0.18	-0.46	0.43	-0.20	0.87	-0.02	0.30	-0.09	0.25	-0.46	0.30	0.21	0.33	-0.09
abril	0.33	-0.15	0.42	-0.16	0.81	0.02	0.28	-0.54	0.21	0.25	0.29	0.19	0.34	-0.16
maio	0.39	-0.13	0.58	-0.31	0.81	0.00	0.28	-0.21	0.37	0.24	0.27	0.36	0.50	0.45
junho	0.21	-0.08	0.44	-0.07	0.99	-0.02	0.25	-0.31	0.69	-0.10	0.21	0.20	0.98	-0.01
julho	0.17	-0.04	0.28	-0.13	0.49	0.11	0.18	-0.13	0.64	0.06	0.15	-0.38	0.80	0.50
agosto	0.14	0.08	0.24	-0.14	0.79	0.04	0.21	-0.32	0.31	-0.16	0.13	0.09	0.41	0.09
setembro	0.26	0.23	0.21	0.19	0.74	-0.04	0.19	-0.43	0.24	0.25	0.17	0.06	0.45	-0.36
outubro	0.20	0.36	0.23	0.54	0.74	0.04	0.17	-0.33	0.30	0.14	0.24	0.05	0.70	0.12
novembro	0.15	-0.07	0.28	0.09	0.69	-0.28	0.19	-0.35	0.35	-0.31	0.20	-0.24	0.85	-0.08
dezembro	0.37	-0.12	0.56	-0.25	1.18	-0.27	0.31	-0.12	0.46	-0.02	0.29	0.05	0.71	-0.25
média	0.25	0.06	0.37	0.07	0.75	0.06	0.24	0.11	0.35	0.06	0.23	0.09	0.56	0.09

Tabela D.3 Resultados obtidos para a usina de Três Marias (40990080)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.87	0.03	0.63	-0.07	0.47	0.10	0.39	0.21	0.50	0.27	0.41	0.13	0.73	-0.03
fevereiro	0.77	-0.14	0.99	0.09	0.67	-0.21	0.58	-0.14	0.57	-0.18	0.49	-0.17	0.49	0.19
março	0.63	-0.09	0.87	-0.20	0.41	0.21	0.58	-0.03	0.40	-0.42	0.41	-0.07	0.37	0.23
abril	0.92	0.25	0.84	-0.09	0.64	-0.02	0.95	-0.06	0.45	0.12	0.45	0.03	0.48	0.15
maio	0.62	0.39	0.56	0.24	0.70	-0.09	0.69	0.34	0.44	0.09	0.33	0.22	0.43	-0.09
junho	1.20	0.33	1.11	0.81	0.99	-0.14	0.73	-0.05	0.37	-0.21	0.35	0.14	0.36	-0.48
julho	1.22	0.33	0.97	0.63	1.84	-0.26	0.81	-0.34	0.37	0.05	0.37	-0.16	0.61	0.19
agosto	0.70	0.29	0.48	0.26	2.22	-0.21	0.92	-0.37	0.45	0.00	0.49	-0.54	0.36	-0.01
setembro	0.39	0.49	0.43	0.40	1.29	-0.23	0.66	-0.26	0.36	-0.19	0.37	-0.29	0.41	0.05
outubro	1.16	0.49	0.67	0.15	1.90	-0.31	0.86	-0.27	0.76	0.01	0.65	0.10	0.69	-0.14
novembro	0.51	-0.05	0.99	-0.03	2.35	-0.11	0.66	-0.11	0.48	0.00	0.44	-0.12	0.84	-0.35
dezembro	1.53	0.10	0.68	0.18	1.42	0.15	0.46	0.00	0.43	0.05	0.32	0.02	0.61	-0.25
média	0.86	0.06	0.77	0.10	1.24	0.05	0.71	0.06	0.47	0.05	0.42	0.06	0.52	0.06

Tabela D.4 Resultados obtidos para a usina de São Simão (60877080)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.42	-0.23	0.31	0.14	0.28	0.33	0.24	0.41	0.26	0.42	0.26	0.37	0.31	0.32
fevereiro	0.58	-0.04	0.47	0.12	1.42	0.30	0.32	0.18	0.32	-0.50	0.34	-0.03	0.48	-0.45
março	0.37	-0.10	0.53	-0.04	0.79	0.14	0.24	0.00	0.27	0.07	0.26	0.09	0.31	0.43
abril	0.27	0.28	0.42	0.11	0.47	0.26	0.26	0.09	0.32	-0.10	0.26	0.28	0.34	0.32
maio	0.23	0.36	0.33	0.38	0.61	0.44	0.22	0.12	0.24	0.17	0.21	0.35	0.38	-0.38
junho	0.25	0.26	0.30	0.43	0.60	0.45	0.26	0.20	0.24	0.18	0.18	0.18	0.44	-0.04
julho	0.37	0.36	0.32	0.61	0.38	0.56	0.22	0.38	0.24	0.18	0.18	-0.21	0.30	-0.30
agosto	0.18	0.52	0.19	0.54	0.28	0.59	0.17	0.27	0.23	0.16	0.21	-0.10	0.24	0.06
setembro	0.22	0.40	0.24	0.52	0.40	0.45	0.19	0.21	0.35	-0.16	0.23	-0.28	0.39	-0.07
outubro	0.33	0.56	0.32	0.37	0.27	0.67	0.24	0.20	0.34	-0.15	0.29	-0.30	0.49	-0.14
novembro	0.21	0.02	0.48	0.08	0.72	-0.07	0.26	-0.12	0.32	0.23	0.27	-0.10	0.49	0.02
dezembro	0.64	-0.03	0.48	0.09	0.54	-0.05	0.27	-0.21	0.36	-0.36	0.34	0.46	0.34	-0.07
média	0.34	0.09	0.37	0.10	0.56	0.12	0.24	0.07	0.29	0.07	0.25	0.08	0.38	0.06

Tabela D.5 Resultados obtidos para a usina de Furnas (61661000)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.32	0.03	0.46	0.15	0.32	0.40	0.29	0.41	0.31	0.61	0.31	0.42	0.52	0.35
fevereiro	0.82	-0.36	0.56	-0.06	0.40	0.25	0.31	0.18	0.34	-0.19	0.35	0.07	0.31	0.44
março	0.25	0.09	0.50	-0.06	0.28	0.57	0.35	0.27	0.28	-0.17	0.27	0.27	0.34	0.09
abril	0.56	0.24	0.45	0.08	0.32	0.39	0.42	0.18	0.23	0.21	0.27	0.42	0.31	-0.05
maio	0.31	0.59	0.29	0.47	0.58	0.25	0.42	0.18	0.28	-0.07	0.23	0.32	0.30	-0.20
junho	0.28	0.33	0.48	0.16	0.67	0.02	0.46	0.18	0.27	0.20	0.29	-0.23	0.34	-0.16
julho	0.22	0.41	0.34	0.28	1.32	0.06	0.46	0.11	0.31	-0.06	0.25	-0.12	0.29	-0.26
agosto	0.26	0.58	0.26	0.45	1.25	0.15	0.47	0.15	0.25	0.07	0.25	-0.30	0.36	0.14
setembro	0.41	0.08	0.54	0.11	1.76	0.04	0.40	0.01	0.29	0.16	0.31	-0.06	0.64	0.12
outubro	0.36	0.34	0.68	0.06	2.08	-0.09	0.37	0.24	0.46	-0.12	0.33	0.36	0.63	0.34
novembro	0.48	0.07	0.63	0.10	1.53	0.12	0.31	-0.05	0.67	-0.08	0.28	0.12	0.60	0.06
dezembro	0.64	0.47	0.43	0.40	1.39	0.10	0.30	-0.08	0.39	0.04	0.31	-0.04	0.55	0.14
média	0.42	0.10	0.47	0.07	0.98	0.06	0.36	0.06	0.34	0.06	0.29	0.08	0.42	0.07

Tabela D.6 Resultados obtidos para a usina de Água Vermelha (61998080)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.28	0.00	0.37	0.20	0.29	0.46	0.31	0.52	0.41	-0.09	0.33	0.52	0.42	0.10
fevereiro	0.68	-0.34	0.53	-0.12	0.36	-0.10	0.34	0.05	0.39	-0.27	0.33	0.05	0.43	-0.24
março	0.37	-0.09	0.42	-0.10	0.25	0.05	0.25	0.30	0.28	-0.17	0.25	0.22	0.27	-0.07
abril	0.43	0.16	0.40	0.02	0.24	-0.19	0.29	0.20	0.24	-0.02	0.20	0.18	0.36	0.09
maio	0.23	0.50	0.30	0.37	0.26	-0.48	0.26	0.07	0.27	-0.17	0.20	0.03	0.30	-0.15
junho	0.21	0.27	0.43	0.15	0.60	-0.17	0.29	0.25	0.26	0.08	0.21	0.24	0.40	-0.06
julho	0.19	0.37	0.31	0.28	0.40	-0.13	0.30	0.35	0.26	0.00	0.18	-0.15	0.40	-0.36
agosto	0.22	0.53	0.22	0.47	0.39	-0.07	0.31	0.28	0.26	0.07	0.18	0.66	0.28	0.30
setembro	0.59	0.15	0.40	0.22	0.47	-0.11	0.26	0.18	0.27	0.13	0.23	0.38	0.29	0.39
outubro	0.45	0.40	0.49	0.13	0.76	-0.03	0.29	0.08	0.48	0.06	0.26	0.26	0.41	0.05
novembro	0.43	0.09	0.51	0.10	0.80	-0.07	0.23	0.17	0.68	0.15	0.24	0.26	0.84	-0.19
dezembro	0.58	0.45	0.34	0.53	0.69	-0.04	0.26	0.27	0.48	-0.27	0.29	0.06	0.74	-0.33
média	0.39	0.09	0.39	0.08	0.44	0.06	0.28	0.07	0.36	0.04	0.24	0.08	0.43	0.07

Tabela D.7 Resultados obtidos para a usina de Três Irmãos (62900080)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.56	0.06	0.37	0.26	0.30	0.43	0.26	0.50	0.50	-0.29	0.30	0.47	0.35	0.29
fevereiro	0.32	-0.36	0.58	-0.25	0.29	0.23	0.30	0.53	0.31	0.10	0.33	0.37	0.49	-0.16
março	0.23	-0.16	0.45	-0.10	0.24	0.32	0.24	0.27	0.24	0.47	0.26	0.29	0.32	-0.04
abril	0.21	-0.07	0.37	-0.12	0.23	-0.05	0.19	0.27	0.36	0.04	0.21	0.19	0.28	0.04
maio	0.16	-0.06	0.36	0.01	0.25	0.07	0.15	0.50	0.35	-0.16	0.28	0.20	0.36	-0.08
junho	0.56	0.04	1.04	0.00	0.33	0.20	0.26	0.59	0.36	0.03	0.29	0.21	0.42	0.07
julho	0.37	0.10	0.41	0.10	0.32	0.18	0.22	0.38	0.27	0.32	0.26	-0.11	0.29	0.20
agosto	0.60	-0.04	0.31	0.13	0.33	0.08	0.13	0.13	0.27	-0.02	0.33	-0.09	0.35	0.04
setembro	0.66	-0.02	0.37	0.26	0.30	0.25	0.22	0.19	0.35	0.19	0.37	-0.19	0.39	0.15
outubro	0.57	0.26	0.40	0.18	0.32	0.31	0.32	-0.17	0.32	-0.03	0.44	-0.03	0.42	0.17
novembro	0.26	0.30	0.33	0.31	0.24	0.16	0.32	-0.09	0.28	-0.16	0.41	-0.46	0.31	-0.42
dezembro	0.24	0.15	0.34	0.27	0.26	-0.07	0.40	-0.09	0.36	0.05	0.50	-0.11	0.48	-0.27
média	0.39	0.05	0.44	0.06	0.28	0.06	0.25	0.10	0.33	0.06	0.35	0.08	0.37	0.06

Tabela D.8 Resultados obtidos para a usina de Porto Primavera (63995080)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.41	-0.11	0.51	0.54	0.34	0.46	0.39	0.36	0.44	0.17	0.37	0.44	0.44	0.17
fevereiro	0.46	-0.28	0.75	0.54	0.27	0.07	0.30	0.17	0.21	0.26	0.30	0.14	0.33	-0.07
março	0.26	-0.12	0.31	-0.01	0.22	0.06	0.21	0.24	0.25	0.09	0.21	0.19	0.38	-0.09
abril	0.20	0.23	0.26	0.12	0.20	0.10	0.17	0.26	0.23	0.04	0.19	0.02	0.43	0.21
maio	0.16	0.45	0.23	0.35	0.17	-0.08	0.15	0.25	0.30	-0.02	0.20	-0.17	0.40	-0.03
junho	0.17	0.26	0.38	0.26	0.19	0.10	0.17	0.07	0.23	0.18	0.21	0.11	0.29	0.19
julho	0.14	0.38	0.23	0.42	0.26	0.09	0.13	0.16	0.26	0.51	0.20	0.02	0.66	-0.02
agosto	0.18	0.51	0.19	0.36	0.16	0.48	0.13	0.20	0.44	-0.19	0.22	0.09	0.66	-0.13
setembro	0.34	0.23	0.23	0.49	0.20	0.40	0.16	0.18	0.33	0.15	0.26	0.00	0.38	-0.02
outubro	0.25	0.42	0.30	0.27	0.23	0.28	0.21	-0.05	0.28	0.17	0.30	0.63	0.66	-0.34
novembro	0.20	0.20	0.30	0.22	0.34	0.22	0.19	-0.18	0.34	0.16	0.26	-0.10	0.57	-0.21
dezembro	0.30	0.29	0.29	0.44	0.22	0.43	0.24	-0.24	0.36	0.04	0.31	0.15	0.60	-0.21
média	0.26	0.09	0.33	0.11	0.23	0.06	0.20	0.06	0.31	0.06	0.25	0.07	0.46	0.05

Tabela D.9 Resultados obtidos para a usina de Capivara (64516080)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.46	0.14	0.78	0.15	0.60	0.11	0.35	0.26	0.51	0.45	0.37	0.23	0.50	-0.20
fevereiro	0.35	-0.14	0.48	-0.18	0.37	0.02	0.41	0.00	0.33	0.12	0.40	0.08	0.43	0.05
março	0.25	0.09	0.44	0.18	0.47	0.00	0.29	0.16	0.35	0.32	0.31	0.27	0.37	0.06
abril	0.41	0.14	0.50	0.13	0.42	-0.17	0.30	0.11	0.38	-0.28	0.34	0.37	0.45	-0.22
maio	0.82	-0.17	0.66	-0.08	0.39	0.10	0.33	0.22	0.32	0.37	0.46	0.47	0.47	0.29
junho	0.63	0.09	1.48	-0.05	1.00	0.31	0.32	0.49	0.43	0.03	0.44	0.35	0.42	0.19
julho	0.56	0.02	0.91	0.02	0.47	0.58	0.35	0.35	0.32	0.12	0.42	0.11	0.47	-0.02
agosto	0.60	0.01	0.44	0.09	0.34	0.26	0.37	0.11	0.38	-0.50	0.32	-0.10	0.60	0.01
setembro	0.86	-0.11	0.68	0.16	0.94	0.10	0.38	0.46	0.64	-0.23	0.37	0.09	0.63	0.43
outubro	0.60	0.10	0.86	0.25	0.76	-0.05	0.43	-0.13	0.40	-0.05	0.44	0.08	0.75	-0.19
novembro	0.35	0.14	0.65	0.13	0.40	0.42	0.35	-0.34	0.38	0.06	0.36	0.16	0.56	-0.06
dezembro	0.57	0.03	0.65	-0.06	0.42	0.00	0.37	-0.09	0.36	0.37	0.41	0.26	0.47	-0.17
média	0.54	0.03	0.71	0.04	0.55	0.07	0.35	0.06	0.39	0.06	0.39	0.07	0.50	0.06

Tabela D.10 Resultados obtidos para a usina de Itaipu (64918980)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR(6)		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.19	0.03	0.27	0.16	0.19	0.39	0.21	0.44	0.31	0.30	0.22	0.49	0.24	0.55
fevereiro	0.21	-0.37	0.61	0.55	0.20	0.20	0.27	0.30	0.29	0.00	0.27	0.35	0.35	-0.01
março	0.16	-0.13	0.28	-0.19	0.17	0.31	0.19	0.32	0.26	0.20	0.24	0.46	0.46	-0.16
abril	0.22	0.12	0.22	0.16	0.17	0.38	0.17	0.41	0.29	-0.19	0.25	0.62	0.37	0.14
maio	0.40	0.09	0.29	0.17	0.18	0.22	0.17	0.16	0.26	0.06	0.33	0.50	0.37	0.12
junho	0.39	0.11	0.49	0.07	0.21	0.25	0.20	0.14	0.39	0.10	0.32	0.45	0.45	0.07
julho	0.23	0.15	0.37	0.20	0.18	0.37	0.19	-0.06	0.26	-0.15	0.32	0.19	0.40	0.02
agosto	0.36	0.13	0.25	0.18	0.17	0.20	0.19	-0.43	0.36	-0.17	0.32	0.03	0.47	-0.07
setembro	0.64	-0.07	0.43	0.17	0.26	-0.11	0.22	-0.31	0.42	0.11	0.35	0.02	0.52	0.06
outubro	0.42	0.18	0.40	0.31	0.29	0.03	0.28	0.18	0.34	0.00	0.43	0.46	0.43	0.29
novembro	0.20	0.26	0.35	0.21	0.24	0.51	0.31	-0.06	0.34	0.16	0.38	0.21	0.45	0.14
dezembro	0.29	0.31	0.34	0.45	0.24	0.34	0.32	0.01	0.35	-0.35	0.38	0.16	0.44	-0.17
média	0.31	0.05	0.36	0.08	0.21	0.09	0.23	0.06	0.32	0.05	0.32	0.11	0.41	0.06

Tabela D.11 Resultados obtidos para a usina de Foz do Areia (65774403)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR[6]		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.46	0.22	0.73	0.01	0.67	-0.11	0.56	0.03	0.73	0.21	0.46	0.09	0.60	0.17
fevereiro	0.44	0.30	0.66	0.09	0.44	0.02	0.45	-0.03	0.43	-0.02	0.45	-0.08	0.55	-0.24
março	0.35	0.23	0.72	0.06	0.41	0.31	0.42	0.10	0.43	-0.01	0.45	0.01	0.45	0.36
abril	0.38	-0.20	0.86	-0.09	0.54	-0.47	0.67	0.08	0.60	-0.17	0.45	0.05	0.55	0.09
maio	0.54	-0.05	1.51	-0.09	0.93	-0.24	0.89	-0.15	0.96	-0.47	0.62	-0.23	1.27	-0.13
junho	0.54	-0.03	1.30	0.00	1.37	-0.08	0.56	-0.32	0.65	-0.17	0.53	-0.13	0.61	0.27
julho	0.61	0.19	1.64	-0.10	1.80	-0.13	0.69	0.00	0.66	-0.35	0.64	-0.29	0.66	-0.22
agosto	0.56	0.21	1.33	0.39	1.23	-0.16	1.10	-0.12	0.68	0.28	0.57	-0.06	0.75	0.35
setembro	0.82	-0.12	1.50	-0.08	1.62	0.03	0.84	-0.09	0.83	0.13	0.62	-0.03	0.91	-0.10
outubro	1.01	0.31	0.95	0.38	0.84	0.16	0.61	0.06	0.71	0.13	0.55	-0.32	0.68	-0.11
novembro	0.51	0.09	0.87	0.09	0.56	0.11	0.56	0.29	0.37	0.33	0.37	0.11	0.55	0.04
dezembro	0.64	0.18	0.86	0.10	0.85	0.08	1.12	0.02	0.78	0.02	0.65	0.04	0.54	0.31
média	0.57	0.06	1.08	0.05	0.94	0.06	0.71	0.04	0.64	0.07	0.55	0.05	0.67	0.06

Tabela D.12 Resultados obtidos para a usina de Salto Caxias (65973500)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR[6]		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.54	0.07	0.89	-0.05	0.84	-0.09	0.89	-0.05	0.75	0.11	0.79	-0.09	1.11	-0.41
fevereiro	0.43	0.26	0.62	0.05	0.42	0.26	0.46	-0.05	0.37	0.59	0.48	-0.13	0.54	-0.36
março	0.62	0.07	0.78	-0.01	0.40	0.10	0.51	0.04	0.41	0.39	0.46	0.03	0.49	0.08
abril	0.44	-0.23	0.82	-0.07	0.56	-0.42	0.59	-0.04	0.42	0.00	0.46	-0.09	0.53	-0.22
maio	0.69	-0.35	1.73	-0.14	1.00	-0.14	0.91	0.09	0.83	-0.02	0.67	0.06	1.01	0.26
junho	0.51	-0.08	1.16	0.04	0.67	0.07	0.51	0.00	0.63	0.08	0.53	0.23	0.65	-0.06
julho	0.52	0.20	1.54	-0.10	0.83	-0.02	0.53	0.47	0.47	0.04	0.55	0.37	0.63	-0.21
agosto	0.50	0.17	1.21	0.36	0.74	-0.01	0.89	0.42	0.48	0.05	0.49	-0.05	0.67	-0.07
setembro	0.63	-0.19	1.51	-0.05	1.94	-0.11	0.73	0.08	0.67	0.05	0.63	0.02	0.61	-0.09
outubro	0.93	0.34	0.91	0.33	0.81	0.13	0.53	-0.03	0.65	-0.05	0.56	-0.08	0.61	0.04
novembro	0.47	0.11	0.83	0.16	0.45	0.37	0.51	-0.26	0.37	0.43	0.41	0.12	0.41	-0.05
dezembro	0.58	0.32	0.86	0.18	0.73	0.28	0.87	-0.16	0.60	0.12	0.59	-0.01	0.74	0.15
média	0.59	0.06	1.07	0.05	0.78	0.06	0.66	0.06	0.54	0.07	0.55	0.04	0.64	0.06

Tabela D.13 Resultados obtidos para a usina de Itá (73200080)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR[6]		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.62	0.19	0.76	-0.25	0.55	0.01	0.64	0.17	0.60	-0.08	0.49	0.15	0.62	-0.06
fevereiro	0.48	0.05	1.31	-0.22	0.62	0.04	0.41	0.26	0.46	-0.04	0.50	0.11	0.53	0.00
março	0.49	-0.09	1.07	-0.17	0.53	-0.28	0.74	0.25	0.56	0.09	0.59	0.23	0.49	0.10
abril	0.57	-0.39	1.37	-0.23	0.61	-0.12	0.87	-0.12	0.68	-0.31	0.58	0.08	0.65	-0.19
maio	1.07	-0.18	2.77	-0.27	1.68	0.20	1.03	0.02	1.05	0.33	0.78	0.23	0.94	0.29
junho	0.52	-0.26	1.64	-0.03	0.67	0.00	0.49	-0.28	0.61	-0.06	0.51	-0.24	0.47	0.09
julho	0.56	0.24	1.03	0.13	0.66	-0.14	0.49	-0.38	0.62	-0.15	0.54	0.20	0.45	-0.31
agosto	1.28	0.28	1.66	0.54	1.12	0.11	0.66	-0.35	0.85	0.06	0.66	0.09	0.79	0.09
setembro	1.79	-0.34	1.32	-0.37	1.07	-0.15	0.72	-0.59	0.94	-0.02	0.75	-0.17	0.78	-0.16
outubro	0.47	-0.05	0.73	-0.16	0.64	-0.41	0.47	0.11	0.67	-0.30	0.46	-0.39	0.63	0.14
novembro	0.53	-0.16	0.93	-0.19	0.59	-0.29	0.68	-0.26	0.45	-0.36	0.44	0.08	0.52	-0.43
dezembro	0.54	0.36	1.06	-0.03	0.65	0.24	0.91	0.15	0.58	-0.21	0.55	-0.32	0.65	-0.03
média	0.74	0.07	1.30	0.07	0.77	0.06	0.68	0.06	0.65	0.06	0.57	0.06	0.63	0.06

Tabela D.14 Resultados obtidos para a usina de Dona Francisca (85398000)

							número de trechos = 3 c/ 10 pontos				número de trechos = 4 c/ 7 pontos			
	Ótimo		Multiquadrático		PAR[6]		MARS 2d		MARS 3D		MARS 2d		MARS 3D	
	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r	erro	r
janeiro	0.47	0.56	0.61	-0.06	0.35	0.56	0.43	0.72	0.42	0.32	0.39	0.58	0.58	0.20
fevereiro	0.45	-0.48	1.26	-0.24	0.43	-0.10	0.48	0.80	0.48	-0.20	0.36	0.59	0.54	-0.36
março	0.54	-0.51	1.24	-0.44	0.47	-0.11	0.57	0.47	0.57	-0.27	0.51	0.34	0.53	0.36
abril	0.67	-0.17	0.79	-0.12	0.64	-0.18	0.61	0.35	0.60	-0.10	0.65	-0.13	0.68	-0.09
maio	0.76	0.07	1.86	-0.16	0.80	0.54	0.57	-0.04	0.68	-0.10	0.57	-0.07	0.53	0.22
junho	0.63	0.03	1.12	0.16	0.61	0.19	0.53	0.26	0.45	0.36	0.55	0.03	0.66	-0.05
julho	0.54	0.22	0.92	0.13	0.61	0.08	0.56	-0.05	0.63	0.19	0.59	-0.30	0.65	-0.30
agosto	0.90	0.07	1.19	0.30	0.73	-0.08	0.49	0.20	0.60	-0.11	0.52	-0.19	0.62	0.20
setembro	0.80	0.29	0.93	-0.02	0.69	-0.02	0.49	0.28	0.46	0.40	0.48	-0.38	0.66	-0.24
outubro	0.41	-0.09	0.98	-0.24	0.49	0.10	0.42	-0.18	0.49	-0.34	0.48	-0.03	0.41	0.11
novembro	0.52	-0.24	1.62	-0.26	0.47	-0.04	0.51	-0.26	0.50	-0.43	0.49	-0.26	0.51	-0.26
dezembro	0.47	0.01	1.04	0.01	0.42	0.37	0.67	-0.03	0.41	-0.12	0.37	-0.35	0.57	0.01
média	0.60	0.09	1.13	0.06	0.54	0.08	0.53	0.11	0.52	0.08	0.50	0.09	0.56	0.07

APÊNDICE E DADOS ESTATÍSTICOS

Neste apêndice será ilustrado a MLT, o desvio padrão, a vazão mínima e a vazão máxima (em m³/s), para o período de 1931 a 2004 nos locais das usinas utilizadas para verificar a qualidade da previsão utilizando os modelos descritos no capítulo 3 e analisados no capítulo 4.

Tabela E.1 Dados estatísticos da usina de Serra da Mesa

Dados Estatísticos da Usina Serra da Mesa (20920080) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	1444	1686	1529	1078	573	396	299	232	223	324	577	1066
Desvio Padrão	735	896	653	478	226	146	112	90	89	136	244	566
Mínimo	434	432	554	330	215	145	111	94	99	96	216	360
Máximo	3330	6163	3827	3524	1689	976	717	543	517	847	1566	3823

Tabela E.2 Dados estatísticos da usina de Tucuruí

Dados Estatísticos da Usina de Tucuruí (29680080) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	15401	20626	24086	23928	15367	7361	4291	3047	2398	2682	4401	8663
Desvio Padrão	6146	7573	7449	6759	4753	2265	1089	792	620	836	1597	3669
Mínimo	5249	7196	10319	12956	7242	3772	2278	1667	1392	1269	1715	2764
Máximo	35804	44250	51539	49445	31611	14345	7742	5559	4379	5642	10298	18684

Tabela E.3 Dados estatísticos da usina de Três Marias

Dados Estatísticos da Usina Três Marias (40990080) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	1459	1378	1125	747	452	339	273	226	221	304	610	1094
Desvio Padrão	675	799	547	371	181	140	105	80	80	140	324	478
Mínimo	221	219	287	157	137	64	98	80	94	74	207	163
Máximo	3503	4436	2716	2095	1297	1062	747	487	531	957	1849	2496

Tabela E.4 Dados estatísticos da usina de São Simão

Dados Estatísticos da Usina São Simão (60877080) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	4041	4259	4171	3158	2080	1619	1304	1064	971	1153	1765	2916
Desvio Padrão	1376	1712	1520	1069	692	453	339	266	311	372	627	1084
Mínimo	1172	960	1626	1324	912	759	576	512	469	450	762	748
Máximo	7857	9931	7938	5957	3708	2995	2406	1915	2082	2639	3961	6076

Tabela E.5 Dados estatísticos da usina de Furnas

Dados Estatísticos da Usina Furnas (61661000) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	1742	1662	1478	1011	742	616	507	419	440	517	730	1241
Desvio Padrão	680	633	603	347	233	250	156	123	230	225	313	463
Mínimo	594	357	477	401	310	276	240	204	214	198	295	334
Máximo	3621	3230	3757	2327	1572	2303	1308	921	1889	1822	1989	3123

Tabela E.6 Dados estatísticos da usina de Água Vermelha

Dados Estatísticos da Usina Água Vermelha (61998080) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	3622	3834	3453	2499	1787	1461	1218	1022	1014	1166	1534	2467
Desvio Padrão	1313	1455	1205	819	516	526	354	273	416	513	568	844
Mínimo	1258	853	1191	1048	788	646	570	494	541	578	684	784
Máximo	7005	7938	6862	5105	3651	4748	2995	2236	3379	4041	4154	5348

Tabela E.7 Dados estatísticos da usina de Três Irmãos

Dados Estatísticos da Usina Três Irmãos (62900080) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	1260	1423	1241	852	666	643	521	457	486	575	609	893
Desvio Padrão	513	656	463	312	247	438	204	172	226	264	225	363
Mínimo	255	377	439	324	204	251	202	189	186	189	271	331
Máximo	2720	3759	2403	2075	1693	3761	1466	1100	1596	1629	1541	2093

Tabela E.8 Dados estatísticos da usina de Porto Primavera

Dados Estatísticos da Usina Porto Primavera (63995080) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	11157	12252	11597	8841	6407	5600	4532	3884	3801	4335	5470	8057
Desvio Padrão	3530	3822	3376	2449	1611	1680	1175	959	1242	1383	1546	2245
Mínimo	3042	3600	4895	4151	3311	3125	2627	2236	1881	2298	2740	3150
Máximo	22795	26596	20786	16647	13015	17548	10374	7802	9601	11155	11775	14959

Tabela E.9 Dados estatísticos da usina de Capivara

Dados Estatísticos da Usina Capivara (64516080) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	1450	1537	1269	956	981	1042	934	788	867	1065	977	1081
Desvio Padrão	946	715	540	440	643	892	641	408	575	742	508	601
Mínimo	301	441	466	361	279	368	317	229	275	254	192	339
Máximo	5977	4252	3094	3023	3591	7334	3205	2323	3489	4371	2833	4101

Tabela E.10 Dados estatísticos da usina de Itaipu

Dados Estatísticos da Usina Itaipú (64918980) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	14498	16874	14845	11650	9430	8607	7255	6095	6255	7448	8267	10752
Desvio Padrão	4375	4651	4249	3343	3222	3666	2648	1988	2779	3238	2660	3407
Mínimo	6357	5910	6927	5856	4641	4428	3624	2969	2939	2849	3117	3796
Máximo	26648	31630	26706	22358	22494	31574	20056	12228	15961	18144	17265	21304

Tabela E.11 Dados estatísticos da usina de Foz do Areia

Dados Estatísticos da Usina Foz do Areia (65774403) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	589	675	609	471	566	675	723	601	709	911	686	574
Desvio Padrão	397	396	321	337	475	527	722	507	541	608	425	353
Mínimo	128	150	207	164	92	116	101	87	104	143	187	82
Máximo	2224	2196	1809	2418	2010	2845	5150	3182	3036	3415	2295	1892

Tabela E.12 Dados estatísticos da usina de Salto Caxias

Dados Estatísticos da Usina Salto Caxias (65973600) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	1109	1275	1077	993	1274	1537	1517	1224	1418	1682	1466	1161
Desvio Padrão	757	771	600	791	1137	1168	1461	967	1105	1259	919	699
Mínimo	204	276	205	264	195	228	200	148	191	197	403	208
Máximo	4904	3770	3212	6077	5798	5957	10798	5934	5804	6325	5443	3466

Tabela E.13 Dados estatísticos da usina de Ita

Dados Estatísticos da Usina Itá (73200080) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	656	836	658	698	950	1133	1367	1399	1568	1561	959	756
Desvio Padrão	416	631	444	566	656	662	1271	1144	1004	930	643	535
Mínimo	79	118	153	95	86	117	172	111	183	286	199	129
Máximo	1981	3442	2205	2974	3705	5096	9408	5360	4197	4519	3850	2270

Tabela E.14 Dados estatísticos da usina de Dona Francisca

Dados Estatísticos da Usina Dona Francisca (85398000) 1931-2004												
Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
MLT	193	194	168	228	309	386	422	420	464	460	317	252
Desvio Padrão	115	143	115	186	305	252	268	247	267	314	249	193
Mínimo	17	35	31	20	34	41	86	47	49	87	56	14
Máximo	602	1028	598	895	1771	1077	1390	1259	1246	1515	1578	958

APÊNDICE F CÓDIGO FONTE DOS PROGRAMAS

Neste apêndice é mostrado os códigos fontes dos programas desenvolvidos neste trabalho e descritos no item 3 , sendo que este código fonte é dividido em duas partes para cada modelo utilizado, sendo a primeira parte referente a coeficientes de calibração e a segunda parte ao programa de previsão. Também será ilustrado o código fontes do programa que efetua as análises descritas no capítulo 4.

F.1 MODELO PAR[6]

```
unit Ulinear;

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs, math;

type
  TFLinear = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FLinear: TFLinear;
  procedure prevlinear;
  procedure coeficiente_linear;
  procedure leitura(nom:string;tipo:integer);
{esta função lê os arquivos com a série
histórica}
implementation
uses pricipal,UCoefMult;

{$R *.DFM}
type
   complex = record
           a,b : single;
           end;

  mat = array[1..12,1..6,1..6] of double;
  vet1 = array[1..1500] of complex;
  vet = array[0..1500] of single;
  vet2 = array[1..12,1..6] of double;
  vet3 = array[1..1500] of integer;
  vet4 = array[1..12] of double;
  mat_12x24 = array[1..12,1..24] of real;
  mat_i_12x6 = array[1..12,1..6] of integer;
  mat_12x6x6 = array[1..12,1..6,1..6] of
double;
  mat_6x6 = array[1..6,1..6] of real;

   arq_mat2 = record
        r_cod        : string[8];
        r_mat_12x6x6 : mat_12x6x6;
           end;

var
  na    : array[1..12] of word;
  wk,
  dat,
  vz  : ^vet;
  x   : ^vet1;
  iwk : ^vet3;
  fi  : mat;
  mmes,
  smes: vet4;
  sig,
  nome,
  nom,dated : string;
  s8  : string[8];
  aaa,
  ark,
  aaq,
  arq : text;
  dt,
  dt1,
  ndf,
  ndd,
  i,
  an,
  aa  : word;
  vpr,
  med,
  dpv,
  cav,
  max,
  min,
  cv     : single;
  arqfi  :  file of mat;
```

```
  arq1,
  arq2   :  text;
  cod    :  array [1..20] of string[8];
  ii,jj,kk,
  u,n    :  integer;
  conterro: integer;
{-------------------------------------------------------
-------------}
procedure Det_coef_correl(lag,mes:integer;
num_dados:mat_i_12x6;Coef_correl:mat_1
2x24; var Curv_coef_correl:mat_6x6);
{ determinação dos coeficientes da curva
teórica de correlação }
var
i,j,k,
dad_mes                        { salva o valor
referente a matriz num_dados}
               : integer;
vet_correl      : array [1..24] of real;
mat_correl      : mat2020; { utilizado este
tipo de variável para ser compatível com o
método de gaujor}
retorno_sel_gau : vet20;
rc              : byte;
aux_coef_correl : array[1..12,-24..24] of
real;
begin
  rc:=0;
  for i:=1 to 12 do
     for j:=1 to 24 do
       begin
          aux_coef_correl[i,j]:=coef_correl[i,j];
          aux_coef_correl[i,-j]:=coef_correl[i,j];
       end;
  for i:=1 to 12 do
    aux_coef_correl[i,0]:=0;
  for i:= 1 to 6 do
    begin
      dad_mes:=num_dados[mes,i];
      for j:=1 to 300 do
         new(mat_correl[j]);

      for j:=1 to dad_mes do

mat_correl[j,dad_mes+1]:=aux_coef_correl[
mes,j];

      for j:=1 to dad_mes do
        for k:=1 to dad_mes do

mat_correl[j]^[k]:=aux_coef_correl[mes,j-k];

      for j:= 1 to dad_mes do
         mat_correl[j,j]:=1;
      UcoefMult.selgau( dad_mes,
mat_correl, retorno_sel_gau, rc );
      for j:=1 to 6 do
         begin
           if dad_mes >= j then
              begin
                 if
(abs(retorno_sel_gau[j])>(power(10,-10)))
and

(abs(retorno_sel_gau[j])<(power(10,+10)))
then

Curv_coef_correl[i,j]:=retorno_sel_gau[j]
                 else
                   Curv_coef_correl[i,j]:=0;
              end
              else
                 Curv_coef_correl[i,j]:=0;
         end;
       for j:=1 to 300 do
         dispose(mat_correl[j]);

    end;
end;
{-------------------------------------------------------
-------------}
procedure
var_correl(lag,mes:integer;entrada_correl:m
at_12x24;
          ent_conta_correl:mat_i_12x6; var
saida: mat_12x24);
{calcula a variância das correlações }
var
i               : integer;
aux_var         : array[1..24] of real;
aux             : real;
begin
    aux_var[1]:=0;
    for i:= 2 to lag do
        aux_var[i]:=aux_var[i-1] +
power(entrada_correl[mes][i-1],2);

    for i:=1 to lag do
       begin
          aux:=1/ent_conta_correl[mes,i];
          saida[mes,i]:=aux*(1+2*aux_var[i]);
       end;

end;
{-------------------------------------------------------
-------------}
procedure
correlacao(nd:integer;vazao_i,vazao_j:vet;
var cor:real);
{ calcula o coeficiente de correlação entre
duas séries de vazões}
var
i,j,
n_validos       : integer;
soma_i,
soma_j,
media_i,
media_j,
correla
                :real;
```

```
begin
  soma_i  :=0;
  soma_j  :=0;
  media_i :=0;
  media_j :=0;
  n_validos:=0;
   for i:=1 to nd do
       if (vazao_i[i]>0) and
(vazao_i[i]<999999) and
          (vazao_j[i]>0) and
(vazao_j[i]<999999) then
             begin
               inc(n_validos);
               soma_i:=soma_i+vazao_i[i];
               soma_j:=soma_j+vazao_j[i];
             end;

  media_i:=soma_i/n_validos;
  media_j:=soma_j/n_validos;
  n_validos:=0;
  soma_i  :=0;
  soma_j  :=0;
  correla:=0;
   for i:=1 to nd do
       if (vazao_i[i]>0) and
(vazao_i[i]<999999) and
          (vazao_j[i]>0) and
(vazao_j[i]<999999) then
             begin
               inc(n_validos);
               soma_i:=soma_i+sqr(vazao_i[i]-
media_i);
               soma_j:=soma_j+sqr(vazao_j[i]-
media_j);
               correla:=correla+(vazao_i[i]-
media_i)*(vazao_j[i]-media_j);
             end;
   correla:=correla/(sqrt(soma_i*soma_j));
   cor:=correla;

end;
{-------------------------------------------------------
-------------}

procedure mdpc(n:word; var v:vet; var
md,dp,ca,min,max,cv:single);
{
 Cálculo da média, desvio padrão e
coeficiente de assimetria
 da série histórica
}
  var
   s1,
   s2  : single;
   i,j : word;
  begin
   s1 := 0;
   s2 := 0;
   j := 0;
   min := 100000;
   max := 0;

   for i := 1 to n do
     if (v[i] < 99999.00)and(v[i] > 0.0)
       then begin
            j := j + 1;
            s1 := s1 + v[i];
            s2 := s2 + sqr(v[i]);
            if v[i] < min
              then min := v[i];
            if v[i] > max
              then max := v[i];
            end;
   md := s1/j;
   dp := (s2/j)-md*md;
   dp := sqrt(j*dp/(j-1));
   cv := dp/md;

   s1 := 0;
   for i := 1 to n do
     if v[i] < 99999.00
       then s1 := s1 + sqr(v[i]-md)*(v[i]-md);
   ca := s1/(dp*dp*dp)/(n-3);
   end;
{-------------------------------------------------------
-------------}

procedure leitura(nom:string;tipo:integer);
{
 Leitura do arquivo de dados, com a série
histórica
 tipo 1: lê todos os dados até a data de
previsão
 tipo 2: lê os dados dentro do período de
calibração
}
  var
   i,j,a        : word;
   s2           : array[1..12] of double;
   aaa          : text;
   arqfile      : textfile;
   text         : string;
   Ano_ini_cal,         { ano inicial de
calibração }
   ano_fim_cal,         { ano final de
calibração }
   anoprevisao          { ano de previsão }
                : integer;
  begin
assignfile(arqfile,dir+'\configuracao.ini');
for i:=0 to 1500 do
   vz^[i]:=0;
if fileexists(dir+'\configuracao.ini') then
 begin
  reset(arqfile);
  while not eof (arqfile) do
    begin
      readln(arqfile,text);
      ano_ini_cal:=strtoint(copy(text,22,4));
      readln(arqfile,text);
```

```
      ano_fim_cal:=strtoint(copy(text,22,4));
      readln(arqfile,text);
      anoPrevisao:=strtoint(copy(text,22,4));
    end;
 end
else
 begin
    ano_ini_cal:=0000;
    ano_fim_cal:=9999;
    anoprevisao:=9999;
 end;

closefile(arqfile);

  assign(aaa,nom);
  reset(aaa);

  for i := 1 to 2 do
    readln(aaa);

  for i := 1 to 12 do
    begin
    na[i] := 0;
    mmes[i] := 0;
    smes[i] := 0;
    s2[i] := 0;
    end;

  j := 1;
  while not eof(aaa) do
    begin
    read(aaa,a);
       if tipo = 1 then
            if a<anoPrevisao then
              begin
                for i := 1 to 12 do
                  begin
                    dat^[j] := a*100+i;
                    read(aaa,vz^[j]);
                    if (vz^[j] < 0.1)or(vz^[j] >
90000.0)
                      then vz^[j] := 999999.00;

                    {Calculo da media mensal}

                    if vz^[j] < 99999.00 then
                      begin
                          na[i] := na[i] + 1;
                          mmes[i] := mmes[i] +
vz^[j];
                          s2[i] := s2[i] + vz^[j]*vz^[j];
                       end;
                       j:=j+1;
                    end;
                end;
       if tipo = 2 then
            if (a>=ano_ini_cal) and
(a<=ano_fim_cal) then
              begin
                for i := 1 to 12 do
                  begin
                    dat^[j] := a*100+i;
                    read(aaa,vz^[j]);
                    if (vz^[j] < 0.1)or(vz^[j] >
90000.0)
                      then vz^[j] := 999999.00;

                    {Calculo da media mensal}

                    if vz^[j] < 99999.00 then
                      begin
                          na[i] := na[i] + 1;
                          mmes[i] := mmes[i] +
vz^[j];
                          s2[i] := s2[i] +
vz^[j]*vz^[j];
                      end;
                    j:=j+1;
                  end;
              end;
    readln(aaa);
    end;
  close(aaa);
  for i := 1 to 12 do
    begin
    mmes[i] := mmes[i]/na[i];
    smes[i] := (s2[i]/na[i])-mmes[i]*mmes[i];
    smes[i] := sqrt(smes[i]*na[i]/(na[i]-1));
    if smes[i] > 99999.9
      then smes[i] := 99999.99;
    if mmes[i] > 99999.9
      then mmes[i] := 99999.99;
    end;

  ndd := j;
  end;
{-------------------------------------------------------
------------}
procedure previsao(k,n:word; var v:vet;
fi:mat; var vp:single);
var
 a,
 z             : single;
 i,p,
 d,
 j,
 erro          : integer;
 zzz           : array [1..6] of real;
begin
 p := 6;
 repeat
   a := 0.0;
   erro:=0;
   for i :=1 to 6 do
     begin
     j := n+1-i; {ok}
     d := k-i; {ok}
     if d < 1
       then d := 12+d;
       zzz[i]:=v[j];
```

```
    if smes[d] < (power(10,-10)) then
       smes[d]:= 0.000001     { evita bug
Marcos 26/07/05}
    else
       z := (v[j]-mmes[d])/smes[d];

    a := a + fi[k,p,i]*z;
    end;
  vp := a*smes[k]+mmes[k];
  if (vp < 0) or (vp>5*mmes[k])
    then begin
         p := p - 1;
         {m := m - 1;}
         end;
  if  vp>5*mmes[k] then
  begin
       inc(conterro);
       inc(erro);
  end;
  until ((vp > 0) and (vp<5*mmes[k]) or
(p=1)) or ((vp>0) and (erro>2));
 if vp < 0
   then vp := mmes[k];
 end;



{-----------------------------------------------------------
------------}
procedure prevlinear;
var
arqg            : textfile;
arq_mat         : file of arq_mat2;
text_mat2       : arq_mat2;
i,j,k           : integer;
begin
assign(arq,dir+'\usinas.txt');
reset(arq);
conterro:=0;

while not eof(arq) do
  begin
    readln(arq,nom);

    new(vz);
    new(dat);
    new(wk);
    new(iwk);
    new(x);

    assign(arqg,dir+'\sigla.dad');
    reset(arqg);
    read(arqg,sig);
    close(arqg);


    assign(aaa,'usinasc.txt');
    reset(aaa);
    repeat
      readln(aaa,nome);
      s8 := nome;
    until (nome = nom)or eof(aaa);
    close(aaa);

    ndd := 0;
  {
    Leitura dos dados a serem analisados
  }
    leitura(dir+'\vazoes\'+nom+'.prn',1);

    mdpc(ndd,vz^,med,dpv,cav,min,max,cv);
    ndf := ndd;

  { -------------------------------------------------
    Análise Linear - Modelo Par(6)
    -------------------------------------------------}
    {Leitura dos coeficientes para ajuste}

    assign(arq2,dir+'\linear\'+'autocp.txt');
    rewrite(arq2);


assignfile(arq_mat,dir+'/linear/coef_linear.bi
n');
    reset(arq_mat);
      while not eof(arq_mat) do
        begin
          read(arq_mat,text_mat2);
          if text_mat2.r_cod = nom then
            begin
              for i:=1 to 12 do
                for j:=1 to 6 do
                  for k:=1 to 6 do
                    fi[i,7-
j,k]:=text_mat2.r_mat_12x6x6[i,j,k];
            end;
        end;
    closefile(arq_mat);

    writeln(arq2,nom);
    for ii :=1 to 12 do
      for jj:=1 to 6 do
        begin
          writeln(arq2);
          write (arq2,ii:2,jj:2);
          for kk:=1 to 6 do
            write (arq2,fi[ii,jj,kk]);
          end;

    close (arq2);


    {Impressão do cabeçalho do relatório da
análise linear}

    assign(ark,dir+'\linear\'+nom+'.rel');
    rewrite(ark);

    writeln(ark);
    dated:=DateTimeToStr(Now);
    writeln(ark,dated);
```

```
   writeln(ark,'COPEL - ',sig:10,'LACTEC -
CEHPAR':50);
   writeln(ark);
   writeln(ark,'ANÁLISE LINEAR':50);
   write(ark,' ':36);
   for i := 1 to 14 do
     write(ark,'-');
   writeln(ark);
   writeln(ark);
   writeln(ark,'Posto: ',nome);
   writeln(ark);
   writeln(ark,'Estatística da Série
Amostral');
   for i := 1 to 29 do
     write(ark,'-');
   writeln(ark);
   writeln(ark,'Média: ':18,med:8:2);
   writeln(ark,'Desvio Padrão: ':18,dpv:8:2);
   writeln(ark,'Coef. Variação: ':18,cv:8:4);
   writeln(ark,'Coef. Assimetria: ',cav:8:5);
   writeln(ark,'Máximo Obs.: ':18,max:8:2);
   writeln(ark,'Minimo Obs.: ':18,min:8:2);
   writeln(ark,'Número Pts Obs.: ':18,ndf:8);
   writeln(ark);
   writeln(ark,'Valores Mensais');
   for i := 1 to 127 do
     write(ark,'-');
   writeln(ark);
   write(ark,'mes:':7);
   for aa := 1 to 12 do
     write(ark,aa:10);
   writeln(ark);

   write(ark,'media:':7);
   for aa := 1 to 12 do
     write(ark,mmes[aa]:10:2);
   writeln(ark);

   write(ark,'d.pad:':7);
   for aa := 1 to 12 do
     write(ark,smes[aa]:10:2);
   writeln(ark);

   write(ark,'nanos:':7);
   for aa := 1 to 12 do
     write(ark,na[aa]:10);
   writeln(ark);

   for i := 1 to 127 do
   write(ark,'-');
   writeln(ark);

   {Determinação da data a ser feita a
previsão}

   i := 0;
   while (vz^[ndd-i] > 90000)or(vz^[ndd-i] =
0) do
     i := i + 1;
   ndd := ndd-i;

   dt := 1 + round(dat^[ndd]) mod 100;
   dt1 := dt;
   an := round(dat^[ndd]) div 100;
   if dt > 12
     then dt := dt - 12;

   for i := 1 to 6 do
     begin
     aa := dt - i;
     if aa < 1
       then aa := 12 - aa;
     if vz^[ndd-i] > 90000
       then vz^[ndd-i] := mmes[aa];
     end;

   writeln(ark);
   writeln(ark,'VALORES PREVISTOS');
   writeln(ark,'MES':3,'ANO':5,'VALOR':9);
   for i := 1 to 18 do
     write(ark,'-');
   writeln(ark);

   assign(aaq,dir+'\grafico\'+nom+'.grf');
   append(aaq);
   writeln(aaq,'Média');
   for aa:=1 to 12 do
      begin
          write(aaq,mmes[dt]:10:2);
          dt:=dt+1;
          if dt>12
            then dt:=dt-12;
      end;
   writeln(aaq);
   for aa:=1 to 12 do
      begin
          write(aaq,smes[dt]:10:2);
          dt:=dt+1;
          if dt>12
            then dt:=dt-12;
      end;
   writeln(aaq);
   writeln(aaq,'Linear');
   close(aaq);

   {Chamada da rotina de previsão do
modelo linear}

   for aa := 1 to 12 do
     begin
     previsao(dt,ndd,vz^,fi,vpr);
     ndd := ndd+1;
     vz^[ndd] := vpr;
     if (dt = 1)
       then an := an + 1;
        writeln(ark,dt:3,an:5,vpr:10:2);

     assign(aaq,dir+'\grafico\'+nom+'.grf');
     append(aaq);
```

```
      if vpr < 900000
        then writeln(aaq,dt:3,vpr:10:2);

      close(aaq);

      dt := dt + 1;
      if dt > 12
        then dt := dt - 12;
      end;

    for i := 1 to 18 do
      write(ark,'-');
    writeln(ark);
    close(ark);

    dispose(vz);
    dispose(dat);
    dispose(wk);
    dispose(iwk);
    dispose(x);

  end;
close(arq);

end;

procedure coeficiente_linear;
{ Função que calcula os coeficientes para a
previsão linear }

var
arq             : textfile;
arq_mat         : file of arq_mat2;
text_mat2       : arq_mat2;     {variável que
armazena os coeficientes da matriz }
nome_usi        : string[8];
i,j,n,k,
n_lag           : integer;
vazao_mes_i,
vazao_mes_j     : ^vet;
correl          : real;
varian_correl,
matriz_correl   : mat_12x24;
m_conta_correl  : mat_i_12x6;
retorno         : mat_6x6;
Curv_coef_correl: mat_12x6x6;
begin
  new(vazao_mes_i);
  new(vazao_mes_j);
  assign(arq,dir+'\usinasb.txt');
  reset(arq);
  for i:= 1 to 12 do
     for j:=1 to 6 do
        for k:=1 to 6 do
           curv_coef_correl[i,j,k]:=0;
  for i:=1 to 12 do
     for j:=1 to 24 do
          matriz_correl[i][j]:=0;

  for j:=1 to 6 do
    begin
      m_conta_correl[1][j]:=7-j;
      for i:= 2 to 12 do

m_conta_correl[i][j]:=m_conta_correl[1][j];
    end;

assignfile(arq_mat,dir+'\linear\coef_linear.bi
n');
  rewrite(arq_mat);
  while not eof(arq) do
    begin
       new(vz);
       new(dat);
       new(wk);
       new(iwk);
       new(x);

     readln(arq,nome_usi);

leitura(dir+'\vazoes\'+nome_usi+'.prn',2);
      for i:=1 to 12 do
        begin
          n:=i;
          j:=0;
          n_lag:=0;
          repeat
            if (vz^[n] > 0) and (vz^[n]<999999)
then
              begin
                vazao_mes_i^[j]:=vz^[n];
                inc(j);
              end;
              n:=n+12;
          until n>=ndd;
          for k:=(i-1) downto (1) do
            begin
              n:=k;
              j:=0;
              repeat
                if (vz^[n] > 0) and
(vz^[n]<999999) then
                  begin
                    vazao_mes_j^[j]:=vz^[n];
                    inc(j);
                  end;
                 n:=n+12;
              until n>=ndd;
              inc(n_lag);

correlacao(j,vazao_mes_i^,vazao_mes_j^,c
orrel);
              matriz_correl[i][n_lag]:=correl
            end;
          for k:=12 downto (1) do
            begin
              n:=k;
              vazao_mes_j^[0]:=0;
              j:=1;
              repeat
```

```
              if (vz^[n] > 0) and
(vz^[n]<999999) then
                begin
                  vazao_mes_j^[j]:=vz^[n];
                  inc(j);
                end;
              n:=n+12;
            until (n+12)>=ndd;
            inc(n_lag);

correlacao(j,vazao_mes_i^,vazao_mes_j^,c
orrel);
            matriz_correl[i][n_lag]:=correl
          end;
        for k:=12 downto i do
          begin
            n:=k;
            vazao_mes_j^[0]:=0;
            vazao_mes_j^[1]:=0;
            j:=2;
            repeat
              if (vz^[n] > 0) and
(vz^[n]<999999) then
                begin
                  vazao_mes_j^[j]:=vz^[n];
                  inc(j);
                end;
              n:=n+12;
            until (n+24)>=ndd;
            inc(n_lag);

correlacao(j,vazao_mes_i^,vazao_mes_j^,c
orrel);
            matriz_correl[i][n_lag]:=correl
          end;

Det_coef_correl(24,i,m_conta_correl,matriz
_correl,retorno);
        for j:=1 to 6 do
          for k:=1 to 6 do

curv_coef_correl[i,j,k]:=retorno[j,k];

       end; { end do for }
     text_mat2.r_cod:=nome_usi;

text_mat2.r_mat_12x6x6:=curv_coef_correl
;
     write(arq_mat,text_mat2);
     dispose(vz);
     dispose(dat);
     dispose(wk);
     dispose(iwk);
     dispose(x);
   end;
closefile(arq_mat);
dispose(vazao_mes_i);
dispose(vazao_mes_j);
end;{ fim da procedure coeficiente_linear}
end.
```

F.2 MODELO MULTIQUADRÁTICO

```
unit UCoefMult; { esta unit constrói a matriz
de calibração do modelo multiquadrático }

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs;

type
  TFCoefMulti = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

type
    vet12    = array[1..12]   of double;
    vet20    = array[1..300]  of double;
    mat2020  = array[1..300]  of ^vet20;
    mat1200s = array[1..1200] of single;
    mat1200b = array[1..1200] of byte;
var
  FCoefMulti: TFCoefMulti;
  procedure matrizk;
  procedure gaujor( n:integer; var
a:mat2020; var rc:byte );
  procedure selgau( n:integer; var
a:mat2020; var x:vet20; var rc:byte );

implementation

uses pricipal;

{$R *.DFM}
    procedure gaujor( n:integer; var
a:mat2020; var rc:byte );

    { Inversao de matrizes - metodo de
Gauss-Jordan }

      var
        i,
        j,
        k,
        r,
        hi  : integer;
        max,
        hr  : extended;
        hv  : array[1..300] of extended;
        p   : array[1..300] of integer;

      begin

        for j := 1 to n do
          p[j] := j;
```

```
      for j := 1 to n do
        begin

          max   := abs(a[j]^[j]);
          r     := j;
          for i := j+1 to n do
            if abs(a[i]^[j]) > max
              then
                begin
                  max := abs(a[i]^[j]);
                  r   := i;
                  end;

          if max < 1.0e-30
            then
              begin
                rc := 1;
                exit;
                end;

          if r > j
            then
              begin

                for k := 1 to n do
                  begin
                    hr       := a[j]^[k];
                    a[j]^[k] := a[r]^[k];
                    a[r]^[k]   := hr;
                    end;

                hi   := p[j];
                p[j] := p[r];
                p[r] := hi;

                end;

          hr := 1.0/a[j]^[j];
          for i := 1 to n do
            a[i]^[j] := hr*a[i]^[j];

          a[j]^[j] := hr;
          for k := 1 to n do
            if k <> j
              then
                begin

                  for i := 1 to n do
                    if i <> j
                      then
                        a[i]^[k] := a[i]^[k] -
a[i]^[j]*a[j]^[k];

                  a[j]^[k] := -hr*a[j]^[k];
                  end;

          end;


        rc := 0;

        for i := 1 to n do
          begin

            for k := 1 to n do
              hv[p[k]] := a[i]^[k];

            for k := 1 to n do
              a[i]^[k] := hv[k];

            end;

          end;
      procedure selgau( n:integer; var
a:mat2020; var x:vet20; var rc:byte );

      { sistema de equacoes lineares - metodo
de eliminacao de Gauss }

        var
          i,j,k,
          l,c     : integer;
          max,
          aux     : extended;
          p       : array[1..300] of integer;
          s       : array[1..300] of extended;


        begin

          for i := 1 to n do
            p[i] := i;

          for i := 1 to n do
            begin
              s[i] := abs(a[i]^[1]);
              for j := 2 to n do
                if s[i] < abs(a[i]^[j])
                  then
                    s[i] := abs(a[i]^[j]);
              end;


          for i := 1 to n do
            begin

              c   := i;
              l   := i;
              max := abs(a[i]^[i])/s[i];
              for k := i to n do
                for j := i to n do
                  if max < abs(a[k]^[j])/s[k]
                    then
                      begin
                        max := abs(a[k]^[j])/s[k];
                        c   := j;
                        l   := k;
                        end;

              if max < 1.0e-30
```

```
            then
              begin
                rc := 1;
                exit;
                end;

            if l > i
              then
                for j := 1 to n+1 do
                  begin
                    aux    := a[i]^[j];
                    a[i]^[j] := a[l]^[j];
                    a[l]^[j] := aux;
                    end;

            if c > i
              then
                begin
                  k    := p[i];
                  p[i] := p[c];
                  p[c] := k;
                  for k := 1 to n do
                    begin
                      aux      := a[k]^[i];
                      a[k]^[i] := a[k]^[c];
                      a[k]^[c] := aux;
                      end;
                  end;

            for j := i+1 to n do
              begin
                aux := a[j]^[i]/a[i]^[i];
                for k := i+1 to n+1 do
                  a[j]^[k] := a[j]^[k] - aux*a[i]^[k];
                end;

            end;


          if abs(a[n]^[n]) < 1.0e-30
            then
              begin
                rc := 1;
                exit;
                end;

          for i := n downto 1 do
            begin
              x[p[i]] := a[i]^[n+1];
              for j := i+1 to n do
                x[p[i]] := x[p[i]] - a[i]^[j]*x[p[j]];
              x[p[i]] := x[p[i]]/a[i]^[i];
              end;

          rc := 0;

          end;

      procedure matrizk;
      {
        Inverte a matriz de distâncias.
        Ajuste de hiper-superfícies com
      multiquádricas.
      }


        var
          auxd : double;
          auxl : longint absolute auxd;
           sol          : file of vet20;
           n            : integer;
           a,inva       : mat2020;
           i,j,k,km,ka  : integer;
           rc           : byte;

        begin
          for i := 1 to 300 do
            begin
              new(a[i]);
              new(inva[i]);
              end;

          n := 240;
          for i := 1 to n-1 do
            for j := i+1 to n do
              begin
                k  := abs(i-j);
                km := k mod 12;
                ka := k div 12;
                if km > 6
                  then
                    begin
                      km := 12 - km;
                      ka :=  1 + ka;
                      end;
                a[i]^[j] := sqrt(1.0*sqr(ka) +
      1.0*sqr(km));
                a[j]^[i] := a[i]^[j];
              end;

          for i := 1 to n do
            a[i]^[i] := 0.0;

          inva := a;
          gaujor( n, inva, rc );


          assign ( sol,
      dir+'\naolinear\mat_k_ai.bin');
          rewrite( sol  );

          for i := 1 to 300 do
            write(sol,a[i]^);

          for i := 1 to 300 do
            write(sol,inva[i]^);

          for i := 1 to 300 do
             dispose(inva[i]);
```

```
    close(sol);

  end;

end.


unit Uprevmqx; {Esta unit faz  a previsão de
vazões utilizando funções multiquadráticas}

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs;

type
  TFprevmqx = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Fprevmqx: TFprevmqx;
  procedure naolinear;

    type
    vet12    = array[1..12]   of double;
    vet300   = array[1..300]  of double;
    mat300   = array[1..300]  of ^vet300;
    vet1200s = array[1..1200] of single;
    vet1200b = array[1..1200] of integer;


implementation

uses
pricipal;

{$R *.DFM}
  var
     sol,par,
     arq,aaq    : text;
     vaz        : ^vet1200s;
     ano,mes    : ^vet1200b;
     med,dpd,
     e,avz      : vet12;
     nvaz,n,na,
     inic,i1,i2 : integer;
     a,inva     : mat300;
     h,f,r,x,z  : ^vet300;
     i,j,k,
     ka,km,d,nd : integer;
     hest,hobs  : double;
     s          : string;
     s8,nom     : string[8];
     aa, dt     : word;

procedure selamr;

  var
    i,j,k,it : integer;
    zmax,s   : double;

  begin

    for i := 1 to n do
      begin
        x^[i] := 0.0;
        for j := 1 to n do
          x^[i] := x^[i] + inva[i]^[j]*h^[j];
        end;

    exit;

    it := 0;
    repeat

      it := it + 1;

      for i := 1 to n do
        begin
          s := 0.0;
          for j := 1 to n do
            s := s + a[i]^[j]*x^[j];
          r^[i] := h^[i] - s;
          end;

      zmax := 0.0;
      for i := 1 to n do
        begin
          z^[i] := 0.0;
          for j := 1 to n do
            z^[i] := z^[i] + inva[i]^[j]*r^[j];
          if abs(z^[i]) > zmax
            then
              zmax := abs(z^[i]);
          end;

      for i := 1 to n do
        x^[i] := x^[i] + z^[i];

      writeln(it:5,' ',zmax);
      until it = 10;


    end;


procedure lermat;

  var
    i   : integer;
    mat : file of vet300;

  begin
```

```
assignfile(mat,dir+'\naolinear\mat_k_ai.bin');
    reset (mat);


    for i := 1 to 300 do
      read(mat,a[i]^);

    for i := 1 to 300 do
      read(mat,inva[i]^);

    close(mat);

    end;


procedure param(nd:integer; var x:vet300;
var a,med,dpd:double);

  var
    i,j,
    max : integer;
    aux : double;

  begin

    max := nd;
    repeat

      i := 0;
      for j := 1 to max-1 do
        if x[j] > x[j+1]
          then
            begin
              aux    := x[j];
              x[j]   := x[j+1];
              x[j+1] := aux;
              i      := j;
              end;

      max := i;
      until i = 0;

    if (nd mod 2) <> 0
      then
        aux := x[(nd - 1) div 2]
      else
        begin
          j   := nd div 2;
          aux := 0.5*(x[j] + x[j+1]);
          end;

    a   := x[1]*x[nd] - sqr(aux);
    aux := x[1] + x[nd] - 2.0*aux;
    if aux = 0 then
      aux:=1;
    a   := a/aux;

    med := 0.0;
    dpd := 0.0;

    for i := 1 to nd do
      begin
        aux := ln(abs(x[i]-a));
        med := med + aux;
        dpd := dpd + sqr(aux);
        end;

    med := med/nd;
    dpd := sqrt(dpd/nd - sqr(med));

    end;


procedure naolinear;
var
  codusi          : array[1..300] of string[8];
  jj              : string[4];    // verifica se o ano
não esta vazio,
  jjj             : string[1];    // pois se não é lido
o vazio como ano 0
  cont,ii,i,erro,
  anoprevisao     : integer;      { ano para
prever}
  arqfile         : textfile;
  text            : string;
begin  {inicio 1.1}
    new(h);
    new(x);
    new(r);
    new(z);
    new(vaz);
    new(ano);
    new(mes);
     for i := 1 to 300 do
       begin {inicio 1.2}
         new(a[i]);
         new(inva[i]);
         end; {fim 2.2}

    lermat;
    assignfile(arqfile,dir+'\configuracao.ini');
    if fileexists(dir+'\configuracao.ini') then
     begin  {inicio 1.3 }
      reset(arqfile);
      while not eof (arqfile) do
        begin { 1.4 }
          readln(arqfile);
          readln(arqfile);
          readln(arqfile,text);

anoPrevisao:=strtoint(copy(text,22,4));
        end; {2.4 }
     end { 2.3 }
    else
        anoprevisao:=9999;

    closefile(arqfile);

    assignfile (arq,dir+'\usinas.txt');
    reset (arq);
```

```
   cont := 0;
   while not eof(arq) do
     begin    {1.5}
       inc(cont);
       readln(arq, codusi[cont]);
     end;     {2.5}
   for ii:=1 to cont do
     begin    {2.6}
       nom:=codusi[ii];
       s8:=nom;
       assignfile
(sol,dir+'\naolinear\'+nom+'.txx');
       rewrite(sol);

       assignfile
(par,dir+'\vazoes\'+nom+'.prn');
       reset  (par);

       assignfile
(aaq,dir+'\grafico\'+nom+'.grf');
       rewrite (aaq);

       readln(par,s);
       readln(par,s);
       nvaz := 0;

       repeat

         read(par,jj);
         if (jj<>'') and
(strtoint(jj)<anoprevisao) then    { este if tem
o intuido de excluir os dados posteriores
ao}
           begin   {1.8 }              { ano em
que será realizada a previsão                  }
             erro:=0;
             for i:=1 to 4 do
             begin  {1.9}
               jjj:=copy(jj,i,1);
               if (jjj = '') or (JJJ=' ') then
                  erro:= erro
               else
                  inc(erro);
             end;  {2.9}
             if erro>0 then
             begin  {1.10}
               j:=strtoint(jj);

               i := 0;
               repeat  {1.11}

                 i := i + 1;

                 nvaz       := nvaz + 1;
                 ano^[nvaz] := j;
                 mes^[nvaz] := i;

                 read(par,vaz^[nvaz]);

                 if vaz^[nvaz] < 0.0001 then
                   begin
                   end;

                until (i = 12) or eof(par);
              end   {2.10}
              else
              readln(par);
           end    {2.8}
           else
            readln(par);
       until eof(par);

       closefile(par);

     for i := nvaz+1 to nvaz+12 do
       if mes^[i-1] < 12
         then
           begin  {1.12}
             mes^[i] := mes^[i-1] + 1;
             ano^[i] := ano^[i-1];
             end   {2.12}
         else
           begin   {1.13}
             mes^[i] := 1;
             ano^[i] := ano^[i-1] + 1;
             end;   {2.13}



     for i := 1 to 12 do
       begin       {1.14}
         e[i] := 0.0;
         nd   := 0;
         for j := 1 to nvaz do
           if mes^[j] = i
             then
               begin    {1.15}
                 nd     := nd + 1;
                 h^[nd] := vaz^[j];
                 end;    {2.15}
         param(nd,h^,avz[i],med[i],dpd[i]);
         end;    {2.14}

     na   := 20;
     n    := 12*na;

     j := nvaz - n;
     for i := 1 to n do
       begin     {1.15}
         j     := j + 1;
         h^[i] := (ln(abs(vaz^[j]-avz[mes^[j]])) -
med[mes^[j]])/dpd[mes^[j]];
         end;    {2.15}

     selamr;

     writeln(sol);
     writeln(sol,'COPEL - ',nom:10,'LACTEC
- CEHPAR':50);
     writeln(sol);
```

```
     writeln(sol,'ANÁLISE NÃO-LINEAR':50);
     writeln(sol,' ':36);

     writeln(aaq,'ANÁLISE NÃO-
LINEAR':50);
     writeln(aaq,nom:10);


     for i := 1 to 12 do
       begin    {1.16}

         d    := i + n;
         hest := 0.0;
         for j := 1 to n do
           begin  {1.17}
             k  := abs(d-j);
             km := k mod 12;
             ka := k div 12;
             if km > 6
               then
                 begin  {1.18}
                   km := 12 - km;
                   ka :=  1 + ka;
                   end;  {2.18}
             hest := hest + x^[j]*sqrt(1.0*sqr(ka)
+ 1.0*sqr(km));
             end;       {2.17}

         j := i + nvaz;
         hest := med[mes^[j]] +
hest*dpd[mes^[j]];
         hest := abs(avz[mes^[j]]+exp(hest));
         if hest <
(avz[mes^[j]]+exp(med[mes^[j]] -
8*dpd[mes^[j]])) then
            hest:=
(avz[mes^[j]]+exp(med[mes^[j]] -
8*dpd[mes^[j]]));
         if hest >
(avz[mes^[j]]+exp(med[mes^[j]] +
8*dpd[mes^[j]])) then
            hest:=
(avz[mes^[j]]+exp(med[mes^[j]] +
8*dpd[mes^[j]]));

         if hest > 200000 then
            hest:= (avz[mes^[j-
1]]+exp(med[mes^[j-1]] + 8*dpd[mes^[j-1]]));
{ evita absurdos na previsão embora raros}
         k := ano^[j];
         writeln(sol,k:6,mes^[j]:4,hest:12:3);
    // escreve a média, desvio padrão e
previsão para o gráfico
     writeln(aaq,k:6,mes^[j]:4,hest:12:3);

       end;         {2.16}

       closefile(sol);
       closefile(aaq);
     end;
       dispose(h);
       dispose(x);
       dispose(r);
       dispose(z);
       dispose(vaz);
       dispose(ano);
       dispose(mes);
       for i := 1 to 300 do
        begin      {1.18}
          dispose(a[i]);
          dispose(inva[i]);
        end;       {2.18}

     closefile(arq);

end;
end.
```

F.3 MODELO ÓTIMO

```
unit UKriging;

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs, math;

type
  TfKriging = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fKriging: TfKriging;
procedure Coef_Krigign;
procedure previsao_krigign;

implementation

uses
  pricipal, UCoefMult, uLinear;

{$R *.DFM}
  type
    v1 = array[1..1200] of single;
    v2 = array[1..120]  of double;
    v3 = array[1..12] of real;
    v4 = array[1..120]  of ^v2;

    tpar = record
      a,                              { assimetria }
      med,                             { média }
      dpd,                            { desvio
padrão parâmtros }
      invert    : ^v2;                  { inversa da
matriz }
      dt        : ^v4;                  { matriz de
correlações }
      m,md      : integer;
      q         : array[1..12] of integer;  {
posição na matriz dt do mês k a ser
previsto }
      end;
 var
     par         : tpar;
     vaz         : ^v1;          { série de vazões }
     nvaz        : integer;      { número de
dados de vazões }
     aux         : ^v2;          { auxilia para gravar
dados }
     aux2020     : ^mat2020;     { auxilia para
usar a função selgau }
     coefinv     : ^vet20;       { retorna os
coeficientes da matriz }
     lixo        : real;

procedure
nl_det_par(nome:string;pos:byte);
{
  Modelo não-linear
  Determinação dos parâmetros

  dados de entrada: nvaz - tamanho do
vetor com a série histórica de vazões
                    vaz  - vetor contendo as
vazoes (1..nvaz)
                    mes  - mês para cada posição
de vaz
                    ano  - ano para cada posição
de vaz.
}

  var
    arq_v2      : file of v2;
    arq         : textfile;
    i,j,n,
    i1,j1       : integer;
    x           : ^v2;
    r           : double;
    namearq     : string;

  const
    m           : integer = 120;
    md          : integer =  96;

procedure ler_dad(namearq:string);

  var
    arq         : textfile;
    i           : integer;
    lixo        : single;
    anoInicio,                  { ano inicial para
calibração }
    anoFim,                     { ano final para
calibração }
    anoPrevisao : integer;      { ano para
prever}
    text        : string;
  begin
    assignfile(arq,dir+'\configuracao.ini');
    if fileexists(dir+'\configuracao.ini') then
     begin
      reset(arq);
      while not eof (arq) do
        begin
          readln(arq,text);
          anoInicio:=strtoint(copy(text,22,4));
          readln(arq,text);
          anoFim:=strtoint(copy(text,22,4));
          readln(arq,text);

anoPrevisao:=strtoint(copy(text,22,4));
```

```
      end;
    end
   else
    begin
      anoInicio:=0000;
      anoFim:=9999;
      anoprevisao:=9999;
    end;
    { como esta função é usada diversas
vezes e o bloqueo de anos só pode ser
    feito na calibração este if tem o intuito
de permitir que todos os dados sejam
    lidos para outros procedimentos}
    if pos<>0 then
    begin
      anoInicio:=0000;
      anoFim:=9999;
    end;
  closefile(arq);


assignfile(arq,dir+'\vazoes\'+namearq+'.prn'
);
   reset(arq);
   nvaz:=0;
   readln(arq);
   readln(arq);

   repeat
     read(arq,lixo); { a variável lixo lê o
primeiro valor que é o ano}
     if (lixo>=anoInicio) and (lixo<=anofim)
and (lixo<anoprevisao) then
       begin
         i:=0;
         repeat
           inc(nvaz);
           inc(i);
           read(arq,vaz^[nvaz]);
         until eof(arq) or (i = 12);
       end
     else
       readln(arq);
   until eof (arq);

   { Este if desconsidera vazões nulas
como último dado}
   while vaz^[nvaz]<= 0 do
   begin
     nvaz:=nvaz-1;
     i:=i-1;
   end;
   {Este IF faz com que o programa
desconsidere o último ano se
      este for incompleto}
    if pos <> 2 then
    begin
      if i<12 then
        nvaz:=nvaz - i;
    end;
   closefile(arq);

 end;

procedure param(nd:integer; var
a,med,dpd:double);

 var
  i,j,k,
  max    : integer;
  aux    : single;
  bux    : double;

 begin

  max := nd;
  repeat

   i := 0;
   for j := 1 to max-1 do
    if x^[j] > x^[j+1]
     then
      begin
       aux     := x^[j];
       x^[j]   := x^[j+1];
       x^[j+1] := aux;
       i       := j;
       end;

  max := i;
  until i = 0;

  k := nd div 2;
  if (k+k) = nd
   then
    aux := 0.5*(x^[k] + x^[k+1])
   else
    aux := x^[k+1];

  bux :=  x^[1] + x^[nd] - sqr(aux);
  a   := (x^[1]*x^[nd] - sqr(aux))/bux;

  med := 0.0;
  dpd := 0.0;
  for i := 1 to nd do
   begin
    bux := ln(x^[i]-a);
    med := med + bux;
    dpd := dpd + sqr(bux);
    end;

  med := med/nd;
  dpd := sqrt(dpd/nd - sqr(med));

  end;

procedure ler_coefKriging(nome1:string);
 var
  i         : integer;
 begin
```

```
    { carrega os valores da variável par tipo
v2}

namearq:=dir+'\kriging\Kriging_par'+nome1
+'.bin';
    assignfile(arq_v2,namearq);
    reset(arq_v2);
    read(arq_v2,par.a^);
    read(arq_v2,par.med^);
    read(arq_v2,par.dpd^);

    for i:=1 to 120 do
      read(arq_v2,par.dt[i]^);

      read(arq_v2,par.invert^);

    closefile(arq_v2);

    { carrega os valores da variável par tipo
ì nteger}

namearq:=dir+'\kriging\Kriging_int'+nome1+
'.bin';
    assignfile(arq,namearq);
    reset(arq);
    readln(arq,par.m);
    readln(arq,par.md);
    for i:=1 to 12 do
      readln(arq,par.q[i]);
    closefile(arq);

  end;

{-----------------------------------------------------------
----}
  begin

  if (pos = 0)  then      { inicializa as variáveis
dinâmicas p/ estes casos}
  begin
      new(vaz);
      new(x);
      new(par.a);
      new(par.med);
      new(par.dpd);
      new(par.dt);
      new(aux);
      new(coefinv);
      new(aux2020);
      for i := 1 to m do
        new(par.dt^[i]);
      for i:=1 to 300 do
        new(aux2020^[i]);
      new(par.invert);
  end;

    if pos = 0 then
    begin
      par.m  := m;
      par.md := md;

      ler_dad(nome);

      for i := 1 to m do
        begin
          j  := 0;
          i1 := i;
          repeat
            j := j + 1;
            x^[j] := vaz^[i1];
            i1 := i1 + 12;
            until i1 > nvaz;

param(j,par.a^[i],par.med^[i],par.dpd^[i]);
          end;

      for i := 1 to m do
        begin
          par.dt^[i]^[i] := 1.0;
          for j := i+1 to m do
            begin
              r  := 0.0;
              i1 := i;
              j1 := j;
              n  := 0;
              repeat
                r  := r + ln(vaz^[i1]-
par.a^[i])*ln(vaz^[j1]-par.a^[j]);
                i1 := i1 + 12;
                j1 := j1 + 12;
                n  := n  +  1;
                until (i1 > nvaz) or (j1 > nvaz);
              r := r/n - par.med^[i]*par.med^[j];
              r := r/(par.dpd^[i]*par.dpd^[j]);
              par.dt^[i]^[j] := r;
              par.dt^[j]^[i] := r;
              end;
          end;

      par.q[1] := 109;
      for i := 2 to 12 do
        par.q[i] := 96 + i;

      for i:= 1 to m do  {monta a matriz para
ser invertida utilizando o método de gaujor}
          for j:=1 to m do
            aux2020[i][j]:=par.dt[i][j];

      for i:=1 to m do
      begin
          aux2020[i][m+2]:=(ln(vaz^[nvaz-
m+i])-par.med^[i])/par.dpd^[i];    { AxB=Lam}
          par.invert[i]:=  (ln(vaz^[nvaz-m+i])-
par.med^[i])/par.dpd^[i];
          aux2020[m+1][i]:=1;
          aux2020[i][m+1]:=1;
      end;
          aux2020[m+1][m+1]:=0;
          aux2020[m+1][m+2]:=1;
```

```
   //
ucoefmult.selgau(m+1,aux2020^,coefinv^,rc
);   { função que encontra os coeficientes da
matrix}
       lixo:=coefinv^[m+1];

       { grava as variáveis dinâmicas do tipo
v2}

namearq:=dir+'\kriging\Kriging_par'+nome+'
.bin';

       if fileexists(namearq)
         then
           deletefile(namearq);

       assignfile(arq_v2,namearq);
       rewrite(arq_v2);

       write(arq_v2,par.a^);
       write(arq_v2,par.med^);
       write(arq_v2,par.dpd^);
       for i:=1 to 120 do
         write(arq_v2,par.dt[i]^);


         write(arq_v2,par.invert^);

       closefile(arq_v2);

       { grava as variáveis tipo integer da
variável par}

namearq:=dir+'\kriging\Kriging_int'+nome+'.
bin';

       if fileexists(namearq)
         then
           deletefile(namearq);

       assignfile(arq,namearq);
       rewrite(arq);
       writeln(arq,par.m);
       writeln(arq,par.md);
       for i:=1 to 12 do
         writeln(arq,par.q[i]);

       closefile(arq);

       end;



       if pos=1 then
         ler_coefKriging(nome);

       if pos = 2 then
         ler_dad(nome);


     if (pos = 0)  then      { inicializa as
variáveis dinâmicas p/ estes casos}
       begin
         for i := 1 to m do
           dispose(par.dt^[i]);
         for i:=1 to 300 do
           dispose(aux2020^[i]);

         dispose(vaz);
         dispose(x);
         dispose(par.a);
         dispose(par.med);
         dispose(par.dpd);
         dispose(par.dt);
         dispose(aux);
         dispose(coefinv);
         dispose(aux2020);
         dispose(par.invert);
       end;

end;


procedure coef_krigign;
{ Este procedimento calcula os parâmetros
para o método de
interpolação ótima }
  var
    arq         : textfile;
    textarq     : string[8];

  begin
      assignfile(arq,dir+'\usinasb.txt');
      reset(arq);

      repeat

        readln(arq,textarq);
        nl_det_par(textarq,0);    {0 calcula os
coeficientes}

      until eof(arq);

      closefile(arq);
  end;
procedure previsao_krigign;

  var
    arq,
    arq_pre,                  { arquivo que salva
os valores previstos }
    arqteste    : textfile;
    codusi      : string[8];
    prev        : string;       { texto utilizado
para salvar as vazões previstas }
    mes,                      { mês a ser previsto }
    loc,                      { localização do mês a
ser previsto }
    i,j,
    k,m,
```

```
   mes_pre,                  { conta o mês de
previsão }
   mes_ini,                  { último mês com
dados observados }
   loc_ini     : integer;     { posiciona o vetor
no últiomo mês observado }
   vaz_pre,
   vaz_pre2    : v3;
   z,a,b       : real;        { variáveis auxiliares
}
   rc          : byte;
   aux3030     : ^mat2020;    { variável
auxiliar para calcular pesos das vazões }
   co_vazoes   : ^vet20;      { retorna os
coeficientes da matriz em relação as
vazões }

  begin
      par.m:= 120;
      par.md:=  96;
      rc:=0;
      assignfile(arq,dir+'\usinas.txt');   {
Arquivo contendo o código das usinas
selecionadas}
      reset(arq);

      repeat
       new(vaz);  { inicializa variáveis
dinâmicas }
       new(par.a);
       new(par.med);
       new(par.dpd);
       new(par.dt);
       new(aux);
       for i := 1 to par.m do
          new(par.dt^[i]);
       new(par.invert);

       readln(arq,codusi);
       nl_det_par(codusi,1);         { 1 lê os
coeficientes }
       nl_det_par(codusi,2);         { 2 lê a
série histórica de vazões }
       mes_ini:=nvaz mod 12;         {
posiciona no último mês de vazão
observada }
       if mes_ini = 0 then
          loc_ini:=par.q[12]
       else
          loc_ini:= par.q[mes_ini];


assignfile(arq_pre,dir+'\kriging\'+codusi+'.pr
e');    { arquivo que salva as vazões
previstas }
       rewrite(arq_pre);

       for mes_pre:=1 to 12 do
        begin
          mes:=mes_ini + mes_pre;
          if mes> 12 then
            mes:=mes-12;

          loc:= par.q[mes];
{ inicializa variável auxiliar dinâmica }
          new(coefinv);
          new(co_vazoes);
          new(aux2020);
          new(aux3030);
          for i:=1 to 300 do
            begin
              new(aux2020^[i]);
              new(aux3030^[i]);
            end;
          m:=par.md+1;
          for i:= loc_ini downto (loc_ini-
par.md+1) do  {monta a matriz para ser
invertida utilizando o método de gaujor}
           begin
             m:=m-1;
             k:=par.md+1;
             for j:=loc_ini downto (loc_ini-
par.md+1) do
               begin
                 k:=k-1;
                 aux2020^[m]^[k]:=par.dt^[i]^[j];
               end;

aux2020^[m]^[par.md+2]:=par.dt^[loc]^[i];
           end;

          for i:=1 to par.md do       {
montagem da matriz Ok}
           begin
            aux2020^[par.md+1]^[i]:=1;
            aux2020^[i]^[par.md+1]:=1;
           end;
          aux2020^[par.md+1]^[par.md+1]:=0;
          aux2020^[par.md+1]^[par.md+2]:=1;
{ este for constrói uma matrix espelho para
encontar o peso das vazões }
          for i:=1 to (par.md+1) do
            for j:=1 to (par.md+1) do
                aux3030^[i]^[j]:=aux2020^[i]^[j];

          for i:=1 to par.md do

aux3030^[i]^[par.md+2]:=par.invert^[i];
            aux3030^[par.md+2]^[par.md]:=1;

{ função que encontra os coeficientes da
matrix em relação ao mês da previsão}

ucoefmult.selgau(par.md+1,aux2020^,coefi
nv^,rc);

{ função que encontra os coeficientes da
matrix em relação a série de vazões}
```

```
//ucoefmult.selgau(par.md+1,aux3030^,co_
vazoes^,rc);

                { imprime matrix pra testes}
              {
assignfile(arqteste,dir+'/'+codusi+'.tes');
                rewrite(arqteste);
                for i:= 1 to 120 do
                 begin
                  //write(arqteste,inttostr(i));
                  for j:=1 to 120 do

write(arqteste,par.dt[i][j]:10:3);

                  writeln(arqteste);
                 end;
                closefile(arqteste);


assignfile(arqteste,dir+'/'+codusi+'.aux');
                rewrite(arqteste);
                for i:= 1 to 120 do
                 begin
                  write(arqteste,inttostr(i));
                  for j:=1 to 120 do

write(arqteste,aux2020[i][j]:10:3);

                  writeln(arqteste);
                 end;
                closefile(arqteste); }

           j:=0;
           a:=0;
           b:=0;
           k:=par.md;
           for i:=(loc_ini-par.md+1) to (loc_ini)
do
             begin
               k:=k-1;
               inc(j);
               if (par.dpd^[i]) >(power(10,-10))
then
                  z:=(ln(vaz^[(nvaz-k)])-par.a^[i]-
par.med^[i])/(par.dpd^[i])
               else
                  z:=(ln(vaz^[(nvaz-k)])-par.a^[i]-
par.med^[i])/(power(10,-10));
               a:=a+z*coefinv^[j];
             end;
           vaz_pre2[mes_pre]:=
a*exp(par.dpd^[loc])+exp(par.med^[loc]+par
.a^[loc]);
           if
(vaz_pre2[mes_pre])>5*(exp(par.med^[loc])
) then

vaz_pre2[mes_pre]:=5*(exp(par.med^[loc]))
;
           write(arq_pre,mes:3);

writeln(arq_pre,vaz_pre2[mes_pre]:10:2);
           for i:=1 to 300 do
             begin
               dispose(aux2020^[i]);
               dispose(aux3030^[i]);
             end;

           dispose(coefinv);            { finaliza
variável auxiliar dinâmica }
           dispose(aux2020);
           dispose(aux3030);
           dispose(co_vazoes);

         end; { end do for Mes_pre }

         closefile(arq_pre);

         { destrói variáveis dinâmicas }
        for i := 1 to par.m do
          dispose(par.dt^[i]);

         dispose(vaz);
         dispose(par.a);
         dispose(par.med);
         dispose(par.dpd);
         dispose(par.dt);
         dispose(aux);
         dispose(par.invert);

       until eof(arq); { fim do repeat }

       closefile(arq);

  end;

end.
```

F.4 MODELO MARS 2D

```
unit MARS;

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs,
  u_MARS,MARS_uar, math, pricipal;
  type
    vet1000 = v1;

  var
    i,p,
    nv,
    inic,
    nvaz,
    nprec       : integer;
    prec,
    vazr,
    serie_vazao : ^vet1000;
    aprox       : double;
    a           : mat63;
    txt         : text;
    codusi      : string;
    Qx_max,
    Qx_min,
    Qy_max,
    Qy_min      : real;

    procedure coef_MARS;
  const
  nt:integer = numero_trechos;
implementation


procedure
ajust_par(pmin,nv,nt,nprec,inic,nger,percm,
npp,tmax:integer;
                  eps:double; var ab:mat63; var
prec,vazr:vet1000);
{
  pmin  - numero mínimo de pontos em
cada trecho;
  nv    - número de variáveis de decisão
(=2*nt);
  nt    - número de trechos de spline;
  nprec - número de elementos da amostra;
  inic  - ponto de início do cálculo da FOB;
  nger  - número de gerações no alg.
genético;
  percm - perc. máximo inicial na busca
exaustiva (decresce de 1% até atingir 1%);
  npp   - número de pontos por variável na
busca exaustiva;
  tmax  - tempo máximo de processamento
(minutos);
  eps   - valor mínimo admissível no cálculo
da FOB;
  a[*,1]=min; a[*,2]=max; a[*,3]=ótimo.
}
  var
    fmin,
    uimax : double;
    a     : mat63;
    yo,zo : vetd;
    i,j   : byte;
    vazc  : ^vet1000;
    t_inic,
    t_proc : double;

  const
    imax : double = 2147483647.e0;
    fobm : double = 1.0e90;

  function num_ptos(x:double):integer;

    var
      i : integer;

    begin

      for i := 1 to nprec do
        if prec[i] > x
          then
            begin
              num_ptos := i-1;
              exit;
              end;

      num_ptos := nprec;

      end;

  procedure temproc(var tinic,tproc:double);

    begin

      if tinic < 0
        then
          begin
            tproc := 0;
            tinic := now;
            exit;
            end;

      tproc := now - tinic;

      end;

  function valord(tipo:byte; var nv,nt:integer;
var x:vet; var z,y:vetd):double;
  {
    tipo=1 ---> Entrada com parâmetros do
tipo vet  (  0 < x < 2**31-1),
        =2 ---> Entrada com parâmetros do
tipo vetd (0.0 < z < 1.0).
```

```
}
  var
    i,
    cont_maior,
    j1,j2       : integer;
    r,
    aux         : double;
    Coeficiente : v2;
    s_erro      : real;

 function anapar:boolean;

   var
     i : byte;

   begin

     if tipo = 2
       then
         for i := 1 to nv do
           x[i] := trunc(z[i]*imax);

     lerxf(nv,x,r);
     if r > 0.0
       then
         begin
           valord := r;
           anapar := true;
           exit;
           end;

     if tipo = 1
       then
         for i := 1 to nv do
           z[i] := x[i]*uimax;

     for i := 1 to nt-1 do
       y[i] := a[i,1] + (a[i,2] - a[i,1])*z[i];

       // ver se esta correto
   {  repeat
        cont_maior:=0;
          for i:=nt to nv do
               if z[i]<z[i-1]
                 then
                   begin
                     aux:=z[i-1];
                     z[i-1]:=z[i];
                     z[i]:=aux;
                     inc(cont_maior);
                   end;
     until cont_maior=0;}   //

     for i := nt to nv do
       y[i+1] := a[i,1] + (a[i,2] - a[i,1])*z[i];

     anapar := false;

     end;

   begin {inicio valord}

     if anapar
       then
         exit;

     j1 := num_ptos(y[1]);
     for i := 2 to nt-2 do
       begin
         j1 := num_ptos(y[i-1]);
         j2 := num_ptos(y[i]);
         if j2-j1 < pmin
           then
             begin
               r := 1.0e100;
               graxf(nv,x,r);
               valord := r;
               exit;
               end;
         j1 := j2;
         end;
       u_MARS.s_erro_linear( prec,
vazr,nprec,nt,y, s_erro);
       r := s_erro;

       graxf(nv,x,r);

       valord := r;

       end;


   procedure listad;

     var
       i    : byte;

     begin

       for i := 1 to nv do
         begin
           end;



       end;

procedure pesqtot;

   var
     y,z,zi : vetd;
     xx     : vet;
     delta,
     v      : double;
     perc,
     i,n    : integer;
     fim    : boolean;

procedure mudalim;

   var
```

```
  aux,
  bux : double;
  i   : byte;

begin

  aux   := 1.0e-2*perc;
  delta := (aux + aux)/npp;
  for i := 1 to nv do
    begin
      zi[i] := zo[i] - aux;
      if zi[i] < 0.0e0
        then
          zi[i] := 0.0e0;
      bux := zi[i] + aux + aux;
      if bux > 1.0e0
        then
          zi[i] := 1.0e0 - aux - aux;
    end;


  end;


procedure busca(var fim:boolean);

  var
    i    : byte;
    p    : array[1..20] of byte;
    aux  : double;
    sair : boolean;

  begin

    fim := false;

    for i := 1 to nv do
      begin
        p[i] := 0;
        z[i] := zi[i];
        end;

    repeat

      v := valord(2,nv,nt,xx,z,y);

      if v < fmin
        then
          begin
            fmin := v;
            zo   := z;
            yo   := y;
            end;

      if (fmin < eps) or (ngmx < 0)
        then
          begin
            fim := true;
            exit;
            end;

        i    := 1;
        sair := false;
        while (not sair) and (i <= nv) do
          begin
            p[i] := p[i] + 1;
            if p[i] > npp
              then
                begin
                  p[i] := 0;
                  z[i] := zi[i];
                  i    := i + 1;
                  end
              else
                begin
                  z[i] := z[i] + delta;
                  sair := true;
                  end;
            end;

        until i > nv;

      temproc(t_inic,t_proc);
      if t_proc > tmax
        then
          fim := true;

      end;


  begin

    y[0]  := prec[1];
    y[nt] := prec[nprec];
    perc := percm;
    repeat
      mudalim;
      busca(fim);
      perc := perc - 1;
      until (perc < 1) or fim;

    listad;

    end;


procedure genetic;

  type
    mat     = array[1..300] of vet;
    vetd300 = array[0..300] of double;

  var
    iter,
    niter,
    ger,
    i,j,k,
    i1,i2,p  : integer;
    aux      : longint;
    auxd     : double;
```

```
    x,y      : ^mat;
    xmin     : vet;
    z,zr     : vetd;
    fx,fy,px : ^vetd300;
    r,s      : double;
    d        : array[0..30]  of longint;
    n,n3,nb  : integer;
    sair     : boolean;

function rlgm1:longint;

  var
    h : integer;
    l : longint;

  const
    k : longint =      16807;
    m : longint = 2147483647;
    q : longint =     127773;
    r : longint =       2836;
    y : longint =     524287;

  begin

    h := y div q;
    l := y mod q;

    y := k*l - r*h;
    if y <= 0
      then
        y := y + m;


    rlgm1 := y;

    end;


procedure ordena;

  var
    aux : double;
    bux : vet;
    max,
    i,
    j   : integer;

  begin

    max := n;

    repeat

      i := 0;

      for j := 1 to max-1 do
        if fx^[j] > fx^[j+1]
          then
            begin
              aux      := fx^[j];
              fx^[j]   := fx^[j+1];
              fx^[j+1] := aux;

              bux     := x^[j];
              x^[j]   := x^[j+1];
              x^[j+1] := bux;

              i       := j;

              end;

      max := i;

      until i = 0;

    end;


procedure bit(var j,p:integer);

  begin

    p := random(nb);

    j := 1;
    while p > j*32-1 do
      j := j + 1;

    p := p - 32*(j-1);

    end;


procedure finaliza;

  var
    i : byte;

  begin

    for i := 1 to nv do
      zo[i] := xmin[i]*uimax;

    yo[0]:=prec[1];
    yo[nt]:=prec[nprec];

    for i := 1 to nt-1 do
      yo[i] := a[i,1] + (a[i,2] - a[i,1])*zo[i];
    for i := nt to nv do
      yo[i+1] := a[i,1] + (a[i,2] - a[i,1])*zo[i];

    listad;

    dispose(x);
    dispose(y);
    dispose(fx);
    dispose(fy);
    dispose(px);
```

```
    end;


  begin

    new(x);
    new(y);
    new(fx);
    new(fy);
    new(px);
    zr[0]  := prec[1];            // insere o valor
mínimo de Qt
    zr[nt] := prec[nprec];        // insere o valor
máximo de Qt
    d[0] := 1;
    for i := 1 to 30 do
      d[i] := 2*d[i-1];

    fmin := 1.0e150;

    n     :=  300;
    i     :=    0;
    iter  :=    0;
    niter := 50*n;
    repeat

      i := i + 1;

      repeat

        for j := 1 to nv do
          x^[i][j] := rlgm1;

        fx^[i] := valord(1,nv,nt,x^[i],z,zr);

        iter := iter + 1;
        until fx^[i] < fobm;

      until (i >= n) or (iter > niter);

    if i < n
      then
        n := i;

    n3    :=  10*n;
    nb    := 32*nv;
    niter :=    10;

    for ger := 1 to nger do
      begin

        ordena;

        if fmin > fx^[1]
          then
            begin
              xmin := x^ [1];
              fmin := fx^[1];
              end
          else
            begin
              i := 0;
              repeat
                i := i + 1;
                until (xmin[i] <> x^[1][i]) or (i >=
nv);
              if (i < nv) or (xmin[nv] <> x^[1][nv])
                then
                  begin
                    for i := n downto 2 do
                      begin
                        x^ [i] := x^ [i-1];
                        fx^[i] := fx^[i-1];
                        end;
                    x^ [1] := xmin;
                    fx^[1] := fmin;
                    end;
              end;

        temproc(t_inic,t_proc);

        if (fmin < eps) or (ngmx < 0) or (t_proc
> tmax)
          then
            begin
              finaliza;
              exit;
              end;

        for i := 1 to n do
          px^[i] := sqr((n-i+1.0)/fx^[i]);

        s := 0.0;
        for i := 1 to n do
          s := s + px^[i];

        px^[0] := 0.0;
        for i := 1 to n do
          px^[i] := px^[i-1] + px^[i]/s;

        y^[1]  :=  x^[1];
        fy^[1] := fx^[1];
        for i := 2 to n do { seleção }
          begin

            r := random;

            j := 1;
            while (r > px^[j]) and (j < n) do
              j := j + 1;

            y^[i]  :=  x^[j];
            fy^[i] := fx^[j];

            end;

        x^  := y^;
        fx^ := fy^;

        for i := 1 to n do { reprodução }
```

```
        begin

          i1 := 1 + random(n);
          i2 := 1 + random(n);

          iter := 0;
          sair := false;
          repeat

            bit(j,p);

            for k := j+1 to nv do
              begin
                aux       := x^[i1][k];
                x^[i1][k] := x^[i2][k];
                x^[i2][k] := aux;
                end;

            if p < 31
              then
                begin
                  aux       :=  x^[i1][j];
                  x^[i1][j] := (aux       mod d[p]) +
d[p]*(x^[i2][j] div d[p]);
                  x^[i2][j] := (x^[i2][j] mod d[p]) +
d[p]*(aux       div d[p]);
                  end;

            auxd := valord(1,nv,nt,x^[i1],z,zr);
            if auxd < fobm
              then
                begin
                  fx^[i1] := auxd;
                  sair    := true;
                  end
                else
                  x^[i1]  := y^[i1];

            auxd := valord(1,nv,nt,x^[i2],z,zr);
            if auxd < fobm
              then
                begin
                  fx^[i2] := auxd;
                  sair    := true;
                  end
              else
                x^[i2]  := y^[i2];

            iter := iter + 1;
            until sair or (iter > niter);

          end;

        for i := 1 to n3 do { mutação }
          begin

            i1 := 1 + random(n);

            iter := 0;
            sair := false;
            repeat

              bit(j,p);

              if p < 31
                then
                  begin
                    aux := x^[i1][j] and d[p];
                    if aux = 0
                      then
                        x^[i1][j] := x^[i1][j] + d[p]
                      else
                        x^[i1][j] := x^[i1][j] - d[p];
                    end;

              auxd := valord(1,nv,nt,x^[i1],z,zr);
              if auxd < fobm
                then
                  begin
                    fx^[i1] := auxd;
                    sair    := true;
                    end
                else
                  x^[i1]  := y^[i1];

              iter := iter + 1;
              until sair or (iter > niter);

            end;

        end;

    finaliza;

    end;

    begin { ajust_par }

      t_inic := -1;
      temproc(t_inic,t_proc);

      a := ab;

      new(vazc);

      inixf;

      uimax := 1.0e0/imax;


      genetic;
      pesqtot;
      dispose(vazc);

      fimxf;

      writeln(txt,nt:3,fmin);
      for i := 0 to nt do
        writeln(txt,i:3,yo[i],yo[i+nt+1]);
    end;
```

```
function min(n1,n2:integer; var
y:vet1000):double;

  var
    i  : integer;
    yx : double;

  begin

    yx := y[n1];
    for i := n1+1 to n2 do
      if yx > y[i]
        then
          yx := y[i];

    min := yx;

    end;


function max(n1,n2:integer; var
y:vet1000):double;

  var
    i          : integer;
    yx         : double;
  begin

    yx := y[n1];
    for i := n1+1 to n2 do
      if yx < y[i]
        then
          yx := y[i];

    max := yx;

    end;

procedure coef_MARS;
  var
     mes,
     i , k,m    :integer;
     arq_usi    : textfile;
     aux_vazao  : ^vet1000;

  begin

    assignfile(arq_usi,dir+'\usinasa.txt');   {
Arquivo contendo o código das usinas que
seram previstas}
    reset(arq_usi);

     repeat
           readln(arq_usi,codusi);
           assignfile(txt,dir+'\MARS\'+
codusi+'.MARS');
           rewrite(txt);
           new(serie_vazao); { série completa
de vazões }

u_MARS.ler_dad(codusi,0,serie_vazao^,nv
az);          { 2 lê a série histórica de vazões }
           new(prec);   { vazão n-1 }
           new(vazr);   { vazão n }
           new(aux_vazao); // teste;
           for mes:=1 to 12 do
             begin
               u_MARS.organiza
(serie_vazao^, mes, nvaz ,prec^,
vazr^,nprec);
                  { calcula o valor máximo e
mínimo de acordo cam a distribuição log-
normal }

                aux_vazao^ := vazr^;  //teste
                u_MARS.classifica_vetor
(aux_vazao^, nprec); // teste
                p:=num_minimo_pontos; {
número mínimo de pontos por trecho }
                aprox := 0.1;

                 for m:=nt  to nt do   { número de
trechos }
                  begin
                    k := (nprec-2*p) div (2*(nt-2));

                    nv := nt + nt;


                    a[1,1] := prec^[p] + aprox;
                    a[1,2] := prec^[p+k] - aprox;

                  {    a[nt,1] := min(1,p+k,vazr^) -
aprox;  // eixo y
                    a[nt,2] := max(1,p+k,vazr^) +
aprox;     }

                    for i := 2 to nt-2 do
                      begin
                        a[i,1] := prec^[p+(2*i-3)*k]+
aprox;
                        a[i,2] := prec^[p+(2*i-1)*k]-
aprox;
                      end;

                    a[nt-1,1] := prec^[p+(2*(nt-2)-
1)*k] + aprox;
                    a[nt-1,2] := prec^[nprec-p] -
aprox;


                  {  a[nv,1] := min(nprec-p-
k,nprec-p,vazr^) - aprox;
                    a[nv,2] := max(nprec-p-
k,nprec-p,vazr^) + aprox;
                    for i:=nt+1 to nv-1 do
                      begin
```

```
                        a[i,1]:= a[nt,1];
                        a[i,2]:=a[nv,2];
                      end; }
                  //teste
                  a[nt,1]:= aux_vazao^[1] +
aprox;
                  a[nt,2]:= aux_vazao^[p] -
aprox;
                  a[nt+1,1]:= aux_vazao^[p] +
aprox;
                  a[nt+1,2]:= aux_vazao^[p+k] -
aprox;
                  for i:=3 to nt do    //teste
                     begin
                        a[nt+i-1,1] :=
aux_vazao^[p+k*(i-2)]+ aprox;
                        a[nt+i-1,2] :=
aux_vazao^[p+k*(i-1)]- aprox;
                     end;
                  a[nv,1]:= aux_vazao^[nprec-
p] + aprox;
                  a[nv,2]:= aux_vazao^[nprec] -
aprox;

                  // fim teste

ajust_par(p,nv,nt,nprec,1,100,5,6,15,1.0,a,p
rec^,vazr^);
               end;

            end;
         close(txt);
         dispose(prec);   { vazão n-1 }
         dispose(vazr);   { vazão n }
         dispose(serie_vazao);
         dispose(aux_vazao) //teste
   until eof(arq_usi); { fim do repeat }
   closefile(arq_usi);

  end;

end.
```

Para efetuar a previsão de vazões foi criada uma unit chamada U_MARS que compartilha diversas funções com o MARS 3d, e é apresentada no apêndice *F.5 Modelo MARS 3d*.

F.5 MODELO MARS 3D

```
unit MARS_3d;

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs,
  u_MARS,MARS_uar, math, pricipal;
  type
    vet1000 = v1;

  var
    i,p,
    nv,
    inic,
    nvaz,
    nprec,
    mes               : integer;
    prec,
    vazr,
    serie_vazao       : ^vet1000;
    prec_3,         {vazão do mês i, i-1, i-2}
    vazOrg_3          : ^a3_v1; {vazão
ordenada individualmente do mês i, i-1, i-2}
    prec0, prec1, prec2,
    Qorg0, Qorg1, Qorg2 : ^v1;

    aprox       : double;
    a           : mat63;
    txt         : text;
    codusi      : string;
    Qx_max,
    Qx_min,
    Qy_max,
    Qy_min      : real;

    procedure coef_MARS_3d;
  const
  nt:integer = numero_trechos; {  número de
trechos da função MARS }
implementation


procedure
ajust_par(pmin,nv,nt,nprec,inic,nger,percm,
npp,tmax:integer;
                  eps:double; var ab:mat63; var
prec,vazr:a3_v1);
{
  pmin  - numero mínimo de pontos em
cada trecho;
  nv    - número de variáveis de decisão
(=2*nt);
  nt    - número de trechos de spline;
  nprec - número de elementos da amostra;
  inic  - ponto de início do cálculo da FOB;
  nger  - número de gerações no alg.
genético;
  percm - perc. máximo inicial na busca
exaustiva (decresce de 1% até atingir 1%);
  npp   - número de pontos por variável na
busca exaustiva;
  tmax  - tempo máximo de processamento
(minutos);
  eps   - valor mínimo admissível no cálculo
da FOB;
  a[*,1]=min; a[*,2]=max; a[*,3]=ótimo.
}
  var
    fmin,
    uimax : double;
    a     : mat63;
    yo,zo : vetd;
    i,j   : byte;
    vazc  : ^vet1000;
    t_inic,
    t_proc : double;
    Vazao,
    VazaoOrg  :a3_v1;

  const
    imax : double = 2147483647.e0;
    fobm : double = 1.0e90;

  function num_ptos(x:double;
k:integer):integer;
  { k : vetor referência }
    var
      i : integer;

    begin
      k:=k+1;  { isto porque k=1 é o mês Qt,
k=2 Qt-1, k=3 Qt-2}
      for i := 1 to nprec do
        if vazOrg_3^[k]^[i] > x
          then
            begin
              num_ptos := i-1;
              exit;
              end;

      num_ptos := nprec;

      end;

  procedure temproc(var tinic,tproc:double);

    begin

      if tinic < 0
        then
          begin
            tproc := 0;
            tinic := now;
            exit;
            end;
```

```
      tproc := now - tinic;

      end;

  function valord(tipo:byte; var nv,nt:integer;
var x:vet; var z,y:vetd):double;
  {
    tipo=1 ---> Entrada com parâmetros do
tipo vet  (  0 < x < 2**31-1),
        =2 ---> Entrada com parâmetros do
tipo vetd (0.0 < z < 1.0).
  }
    var
      i,k,
      cont_maior,
      j1,j2       : integer;
      r,
      aux         : double;
      Coeficiente : v2;
      s_erro      : real;

   function anapar:boolean;

     var
       i : byte;

     begin

       if tipo = 2
         then
           for i := 1 to nv do
             x[i] := trunc(z[i]*imax);

       lerxf(nv,x,r);
       if r > 0.0
         then
           begin
             valord := r;
             anapar := true;
             exit;
             end;

       if tipo = 1
         then
           for i := 1 to nv do
             z[i] := x[i]*uimax;

       for i := 1 to nt-1 do
         y[i] := a[i,1] + (a[i,2] - a[i,1])*z[i];

       for i := nt to (2*nt-2) do
          y[i+2] := a[i,1] + (a[i,2] - a[i,1])*z[i];

       for i := (2*nt-1) to nv do
          y[i+3] := a[i,2] + (a[i,3] - a[i,2])*z[i];

       anapar := false;

       end;

    begin {inicio valord}

      if anapar
        then
          exit;

      for k:=1 to 2 do
        begin
          j1 := num_ptos(y[1],k);
          for i := (k+nt*(k-1)) to (k*(nt+k-1)-1)
do
            begin
              j1 := num_ptos(y[i-1],k);
              j2 := num_ptos(y[i],k);
              if j2-j1 < pmin
                then
                  begin
                    r := 1.0e100;
                    graxf(nv,x,r);
                    valord := r;
                    exit;
                    end;
              j1 := j2;
            end;
        end;

      u_MARS.s_erro_linear_3D( prec_3^,
                          nprec,nt,y, s_erro);
      r:=s_erro;

      graxf(nv,x,r);

      valord := r;

      end;


  procedure listad;

    var
      i    : byte;

    begin

      for i := 1 to nv do
        begin
          end;


      end;

procedure pesqtot;

  var
    y,z,zi : vetd;
    xx     : vet;
    delta,
    v      : double;
    perc,
```

```
  i,n    : integer;
  fim    : boolean;

procedure mudalim;

 var
  aux,
  bux : double;
  i   : byte;

 begin

  aux   := 1.0e-2*perc;
  delta := (aux + aux)/npp;
  for i := 1 to nv do
   begin
    zi[i] := zo[i] - aux;
    if zi[i] < 0.0e0
     then
      zi[i] := 0.0e0;
    bux := zi[i] + aux + aux;
    if bux > 1.0e0
     then
      zi[i] := 1.0e0 - aux - aux;
   end;


  end;


 procedure busca(var fim:boolean);

  var
   i    : byte;
   p    : array[1..50] of byte;
   aux  : double;
   sair : boolean;

  begin

   fim := false;

   for i := 1 to nv do
    begin
     p[i] := 0;
     z[i] := zi[i];
     end;

   repeat

    v := valord(2,nv,nt,xx,z,y);

    if v < fmin
     then
      begin
       fmin := v;
       zo   := z;
       yo   := y;
       end;

    if (fmin < eps) or (ngmx < 0)
     then
      begin
       fim := true;
       exit;
       end;

    i    := 1;
    sair := false;
    while (not sair) and (i <= nv) do
     begin
      p[i] := p[i] + 1;
      if p[i] > npp
       then
        begin
         p[i] := 0;
         z[i] := zi[i];
         i    := i + 1;
         end
       else
        begin
         z[i] := z[i] + delta;
         sair := true;
         end;
      end;

     until i > nv;

   temproc(t_inic,t_proc);
   if t_proc > tmax
    then
     fim := true;

   end;


begin

  y[0]  := vazOrg_3^[2]^[1];
  y[nt] := vazOrg_3^[2]^[nprec];
  y[nt+1]:=vazOrg_3^[3]^[1];
  y[2*nt+1]:=vazOrg_3^[3]^[nprec];

  perc := percm;
  repeat
   mudalim;
   busca(fim);
   perc := perc - 1;
   until (perc < 1) or fim;

  listad;

  end;


procedure genetic;

 type
  mat     = array[1..300] of vet;
  vetd300 = array[0..300] of double;
```

```
  var
    iter,
    niter,
    ger,
    i,j,k,
    i1,i2,p  : integer;
    aux      : longint;
    auxd     : double;
    x,y      : ^mat;
    xmin     : vet;
    z,zr     : vetd;
    fx,fy,px : ^vetd300;
    r,s      : double;
    d        : array[0..50]  of longint;
    n,n3,nb  : integer;
    sair     : boolean;

function rlgm1:longint;

  var
    h : integer;
    l : longint;

  const
    k : longint =      16807;
    m : longint = 2147483647;
    q : longint =     127773;
    r : longint =       2836;
    y : longint =     524287;

  begin

    h := y div q;
    l := y mod q;

    y := k*l - r*h;
    if y <= 0
      then
        y := y + m;


    rlgm1 := y;

    end;


procedure ordena;

  var
    aux : double;
    bux : vet;
    max,
    i,
    j   : integer;

  begin

    max := n;

    repeat

      i := 0;

      for j := 1 to max-1 do
        if fx^[j] > fx^[j+1]
          then
            begin

              aux      := fx^[j];
              fx^[j]   := fx^[j+1];
              fx^[j+1] := aux;

              bux     := x^[j];
              x^[j]   := x^[j+1];
              x^[j+1] := bux;

              i       := j;

              end;

      max := i;

      until i = 0;

    end;


procedure bit(var j,p:integer);

  begin

    p := random(nb);

    j := 1;
    while p > j*32-1 do
      j := j + 1;

    p := p - 32*(j-1);

    end;


procedure finaliza;

  var
    i : byte;

  begin

    for i := 1 to nv do
      zo[i] := xmin[i]*uimax;

    yo[0]:=Qorg1^[1];
    yo[nt]:=Qorg1^[nprec];
    yo[nt+1]:=Qorg2^[1];
    yo[2*nt+1]:=Qorg2^[nprec];

    for i := 1 to nt-1 do
      yo[i] := a[i,1] + (a[i,2] - a[i,1])*zo[i];
```

```
    for i := nt to (2*nt-2) do
      yo[i+2] := a[i,1] + (a[i,2] - a[i,1])*zo[i];

    for i := (2*nt-1) to nv do
      yo[i+3] := a[i,2] + (a[i,3] - a[i,2])*zo[i];

    listad;

    dispose(x);
    dispose(y);
    dispose(fx);
    dispose(fy);
    dispose(px);

    end;


  begin

    new(x);
    new(y);
    new(fx);
    new(fy);
    new(px);
    zr[0]  := Qorg1^[1];            // insere o
valor mínimo de Qt
    zr[nt] := Qorg1^[nprec];        // insere o
valor máximo de Qt
    zr[nt+1]:=Qorg2^[1];
    zr[2*nt+1]:=Qorg2^[nprec];

    d[0] := 1;
    for i := 1 to 30 do
      d[i] := 2*d[i-1];

    fmin := 1.0e150;

    n     :=  300;
    i     :=    0;
    iter  :=    0;
    niter := 50*n;
    repeat

      i := i + 1;

      repeat

        for j := 1 to nv do
          x^[i][j] := rlgm1;

        fx^[i] := valord(1,nv,nt,x^[i],z,zr);

        iter := iter + 1;
        until fx^[i] < fobm;

      until (i >= n) or (iter > niter);

    if i < n
      then
        n := i;

    n3    :=  10*n;
    nb    := 32*nv;
    niter :=    10;

    for ger := 1 to nger do
      begin

        ordena;

        if fmin > fx^[1]
          then
            begin
              xmin := x^ [1];
              fmin := fx^[1];
              end
          else
            begin
              i := 0;
              repeat
                i := i + 1;
                until (xmin[i] <> x^[1][i]) or (i >=
nv);
              if (i < nv) or (xmin[nv] <> x^[1][nv])
                then
                  begin
                    for i := n downto 2 do
                      begin
                        x^ [i] := x^ [i-1];
                        fx^[i] := fx^[i-1];
                        end;
                    x^ [1] := xmin;
                    fx^[1] := fmin;
                    end;
              end;

        temproc(t_inic,t_proc);

        if (fmin < eps) or (ngmx < 0) or (t_proc
> tmax)
          then
            begin
              finaliza;
              exit;
              end;

        for i := 1 to n do
          px^[i] := sqr((n-i+1.0)/fx^[i]);

        s := 0.0;
        for i := 1 to n do
          s := s + px^[i];

        px^[0] := 0.0;
        for i := 1 to n do
          px^[i] := px^[i-1] + px^[i]/s;

        y^[1]  :=  x^[1];
        fy^[1] := fx^[1];
        for i := 2 to n do { seleção }
```

```
        begin

          r := random;

          j := 1;
          while (r > px^[j]) and (j < n) do
            j := j + 1;

          y^[i]  :=  x^[j];
          fy^[i] := fx^[j];

          end;

        x^  := y^;
        fx^ := fy^;

        for i := 1 to n do { reprodução }
          begin

            i1 := 1 + random(n);
            i2 := 1 + random(n);

            iter := 0;
            sair := false;
            repeat

              bit(j,p);

              for k := j+1 to nv do
                begin
                  aux       := x^[i1][k];
                  x^[i1][k] := x^[i2][k];
                  x^[i2][k] := aux;
                  end;

              if p < 31
                then
                  begin
                    aux       :=  x^[i1][j];
                    x^[i1][j] := (aux       mod d[p]) +
d[p]*(x^[i2][j] div d[p]);
                    x^[i2][j] := (x^[i2][j] mod d[p]) +
d[p]*(aux       div d[p]);
                    end;

              auxd := valord(1,nv,nt,x^[i1],z,zr);
              if auxd < fobm
                then
                  begin
                    fx^[i1] := auxd;
                    sair    := true;
                    end
                  else
                    x^[i1]  := y^[i1];

              auxd := valord(1,nv,nt,x^[i2],z,zr);
              if auxd < fobm
                then
                  begin
                    fx^[i2] := auxd;
                    sair    := true;
                    end
                  else
                    x^[i2]  := y^[i2];

              iter := iter + 1;
              until sair or (iter > niter);

            end;

        for i := 1 to n3 do { mutação }
          begin

            i1 := 1 + random(n);

            iter := 0;
            sair := false;
            repeat

              bit(j,p);

              if p < 31
                then
                  begin
                    aux := x^[i1][j] and d[p];
                    if aux = 0
                      then
                        x^[i1][j] := x^[i1][j] + d[p]
                      else
                        x^[i1][j] := x^[i1][j] - d[p];
                    end;

              auxd := valord(1,nv,nt,x^[i1],z,zr);
              if auxd < fobm
                then
                  begin
                    fx^[i1] := auxd;
                    sair    := true;
                    end
                  else
                    x^[i1]  := y^[i1];

              iter := iter + 1;
              until sair or (iter > niter);

            end;

        end;

    finaliza;

    end;

    begin { ajust_par }

      t_inic := -1;
      temproc(t_inic,t_proc);

      a := ab;
```

```
    new(vazc);

    inixf;

    uimax := 1.0e0/imax;


    genetic;
    pesqtot;
    dispose(vazc);

    fimxf;

    writeln(txt,nt:3,fmin);
    for i := 0 to (nv+3) do
      writeln(txt,'mes ',mes:2,+i:3,yo[i]);
  end;

function min(n1,n2:integer; var
y:vet1000):double;

  var
    i  : integer;
    yx : double;

  begin

    yx := y[n1];
    for i := n1+1 to n2 do
      if yx > y[i]
        then
          yx := y[i];

    min := yx;

    end;


function max(n1,n2:integer; var
y:vet1000):double;

  var
    i          : integer;
    yx         : double;
  begin

    yx := y[n1];
    for i := n1+1 to n2 do
      if yx < y[i]
        then
          yx := y[i];

    max := yx;

    end;

procedure coef_MARS_3d;
  var
     i , k,m    :integer;
     arq_usi    : textfile;
     q,v        : array[1..3,1..1000] of single;
     // variáveis de teste
     arq        : textfile;
  begin

    assignfile(arq_usi,dir+'\usinasa.txt');   {
Arquivo contendo o código das usinas que
seram previstas}
    reset(arq_usi);

     repeat
          readln(arq_usi,codusi);
          assignfile(txt,dir+'\MARS_3d\'+
codusi+'.MARS');
          rewrite(txt);

          new(serie_vazao); { série completa
de vazões }

u_MARS.ler_dad(codusi,0,serie_vazao^,nv
az);          { 2 lê a série histórica de vazões }
          new(prec);   { vazão n-1 }
          new(vazr);   { vazão n }
          new(prec0);  { vazão n }
          new(prec1);  { vazão n-1 }
          new(prec2);  { vazão n-2 }
          new(Qorg0);  { vazão ordenada n }
          new(Qorg1);  { vazão ordenada n-1
}
          new(Qorg2);  { vazão ordenada n-2
}
          new(prec_3);
          new(vazOrg_3);

          for k:=1 to 3 do
            begin
              new(prec_3^[k]);
              new(vazOrg_3^[k]);
            end;
          for mes:=1 to 12 do  // TESTE
            begin
              u_MARS.organiza_3D
(serie_vazao^,mes,nvaz,prec0^, prec1^,
prec2^,
                                 Qorg0^, Qorg1^,
Qorg2^,nprec);

              prec_3^[1]^:=prec0^;
              prec_3^[2]^:=prec1^;
              prec_3^[3]^:=prec2^;
              vazOrg_3^[1]^:=Qorg0^;
              vazOrg_3^[2]^:=Qorg1^;
              vazOrg_3^[3]^:=Qorg2^;

              p:=num_minimo_pontos; {
número mínimo de pontos por trecho }
              aprox := 0.1;

               for m:=nt  to nt do   { número de
trechos }
```

```
                begin
                  k := (nprec-2*p) div (2*(nt-2));

                  nv := 2*(nt-
1)+trunc(power((nt+1),2));


                  a[1,1] := vazOrg_3^[2]^[p] +
aprox;
                  a[1,2] := vazOrg_3^[2]^[p+k] -
aprox;

                  a[nt,1] := vazOrg_3^[3]^[p] +
aprox;
                  a[nt,2] := vazOrg_3^[3]^[p+k]
- aprox;

                  for i := 2 to nt-2 do
                    begin
                      a[i,1] :=
vazOrg_3^[2]^[p+(2*i-3)*k]+ aprox;
                      a[i,2] :=
vazOrg_3^[2]^[p+(2*i-1)*k]- aprox;
                      a[nt+i-1,1] :=
vazOrg_3^[3]^[p+(2*i-3)*k]+ aprox;
                      a[nt+i-1,2] :=
vazOrg_3^[3]^[p+(2*i-1)*k]- aprox;

                    end;

                  a[nt-1,1] :=
vazOrg_3^[2]^[p+(2*(nt-2)-1)*k] + aprox;
                  a[nt-1,2] :=
vazOrg_3^[2]^[nprec-p] - aprox;
                  a[2*nt-2,1] :=
vazOrg_3^[3]^[p+(2*(nt-2)-1)*k] + aprox;
                  a[2*nt-2,2] :=
vazOrg_3^[3]^[nprec-p] - aprox;


                  a[2*nt-1,1] := min(p,nprec-
p,vazOrg_3^[1]^) - aprox;
                  a[2*nt-1,2] := max(p,nprec-
p,vazOrg_3^[1]^) + aprox;

                  for i:=2*nt to nv do
                    begin
                      a[i,1]:= a[2*nt-1,1];
                      a[i,2]:= a[2*nt-1,2];
                    end;


ajust_par(p,nv,nt,nprec,1,100,5,6,15,1.0,a,v
azOrg_3^,prec_3^);
                end;

              end;
          close(txt);
          dispose(prec);   { vazão n-1 }
          dispose(vazr);   { vazão n }
          dispose(prec0);  { vazão n }
          dispose(prec1);  { vazão n-1 }
          dispose(prec2);  { vazão n-2 }
          dispose(Qorg0);  { vazão ordenada
n }
          dispose(Qorg1);  { vazão ordenada
n-1 }
          dispose(Qorg2);  { vazão ordenada
n-2 }

          for k:=1 to 3 do
            begin
              dispose(prec_3^[k]);
              dispose(vazOrg_3^[k]);
            end;

          dispose(prec_3);
          dispose(vazOrg_3);

      until eof(arq_usi); { fim do repeat }
      closefile(arq_usi);

    end;


end.



unit U_MARS;

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs, math,
  MARS_uar;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

  type
    v1 = array[1..1500] of real;
    v2 = array[1..100] of real;
    v_mes_2 = array[1..12] of ^v2;
    a3 = ^v1;
    a3_v1 = array[1..3] of a3;
var
  Form1: TForm1;
  procedure ler_dad(namearq:string;
pos:integer; var vazao :v1; var
nvaz:integer);
  procedure organiza (var vazao: v1; mes,
ndados:integer ; var mes_0, mes_1: v1;var
npares:integer);
```

```
  procedure organiza_3D (var vazao: v1;
mes, ndados:integer ;
                    var Qmes0,Qmes1,Qmes2 :
v1;
                    var Qord0,Qord1,Qord2 :
v1; var npares:integer);
  procedure coef_MARS_Min_quad(
Vazao_0, Vazao_1:v1; npares,
grau:integer;
                      var Coeficiente:v2; var
s_erro:real);
  procedure s_erro_linear( Vazao_0,
Vazao_1:v1; npares, nt:integer;
p_pontos:vetd;
                    var s_erro:real);
  procedure s_erro_linear_3D( Vazao
:a3_v1;
                     npares,nt:integer;
p_pontos:vetd; var s_erro:real);

  procedure prev_MARS_linear;
  procedure log_normal_3(vazao :
v1;nun_pontos:integer; var max, min:real);
  procedure prev_MARS_linear_3d;
  procedure classifica_vetor (var vazao : v1;
npar:integer);
implementation

{$R *.DFM}
uses
   pricipal, UCoefMult, MARS;
var  { variáveis globais para a unit
u_MARS}
     vaz              : ^v1;

const
     pmin       : integer =
num_minimo_pontos;
     nv         : integer = numero_trechos ;
     nt         : integer = 2*numero_trechos ;
     inic       : integer = 1;
     nger       : integer = 100;
     percm      : integer = 5;
     npp        : integer = 6;
     tmax       : integer = 15;
     eps        : integer = 1;
{-----------------------------------------------------
---------------------
    esta procedure lê as vazões de
determinada usina no banco de dados
{-----------------------------------------------------
---------------------}

procedure classifica_vetor (var vazao :
vet1000; npar:integer);
var
  i,j     : integer;
  aux     : real;
begin
   repeat
      i:=0;
      for j:=1 to (npar-1) do
        begin
          if vazao[j]>vazao[j+1] then
            begin
              aux := vazao[j];
              vazao[j]:=vazao[j+1];
              vazao[j+1]:=aux;
              inc(i);
            end;
        end;
   until i = 0;
end;
procedure ler_dad(namearq:string;
pos:integer; var vazao :v1; var
nvaz:integer);

  var
    arq         : textfile;
    i           : integer;
    lixo        : single;
    anoInicio,                { ano inicial para
calibração }
    anoFim,                   { ano final para
calibração }
    anoPrevisao : integer;      { ano para
prever}
    text        : string;
  begin
  anoInicio:=0000;
  anoFim:=9999;
  anoprevisao:=9999;
  i:=0;
    assignfile(arq,dir+'\configuracao.ini');
    if fileexists(dir+'\configuracao.ini') then
     begin
      reset(arq);
      while not eof (arq) do
        begin
          readln(arq,text);
          anoInicio:=strtoint(copy(text,22,4));
          readln(arq,text);
          anoFim:=strtoint(copy(text,22,4));
          readln(arq,text);

anoPrevisao:=strtoint(copy(text,22,4));
        end;
     end;

     { como esta função é usada diversas
vezes e o bloqueo de anos só pode ser
     feito na calibração este if tem o intuito
de permitir que todos os dados sejam
     lidos para outros procedimentos}
     if pos<>0 then
     begin
       anoInicio:=0000;
       anoFim:=9999;
     end;
  closefile(arq);
```

```
  new(vaz);  { inicializa variáveis dinâmicas }


assignfile(arq,dir+'\vazoes\'+namearq+'.prn'
);
    reset(arq);
    nvaz:=0;
    readln(arq);
    readln(arq);

    repeat
       read(arq,lixo); { a variável lixo lê o
primeiro valor que é o ano}
       if (lixo>=anoInicio) and (lixo<=anofim)
and (lixo<anoprevisao) then
         begin
           i:=0;
           repeat
             inc(nvaz);
             inc(i);
             read(arq,vaz^[nvaz]);
           until eof(arq) or (i = 12);
         end
       else
         readln(arq);
    until eof (arq);

    { Este if desconsidera vazões nulas
como último dado}
    while vaz^[nvaz]<= 0 do
    begin
       nvaz:=nvaz-1;
       i:=i-1;
    end;
    {Este IF faz com que o programa
desconsidere o último ano se
        este for incompleto}
     if pos <> 2 then  { lê ano inteiro}
     begin
        if i<12 then
          nvaz:=nvaz - i;
     end;
    closefile(arq);
    vazao:=vaz^;
    dispose(vaz);
  end;
{-------------------------------------------------------
---------------------
   organiza as vazões em ordem crescente
   vazao  = vetor de entrada com a série
histórica
   mes    = mês de referência
   ndados = número de vazões
   mes_0  = vetor em ordem crescente do
mês (n-1) (variável independente(x))
   mes_1  = vetor com o mês (n) (variável
dependente (y))
   ndados = número de pares de valores
 -------------------------------------------------------
---------------------}
procedure organiza (var vazao: v1; mes,
ndados:integer ; var mes_0, mes_1: v1;var
npares:integer);
  var
    i,j         : integer;
    auxilia     : single;
  begin

    i := mes;
    j:=0;
    while i<= ndados do
      begin
        inc(j);
        if i =1 then
          i:=1+12; { se a vazão for referente
ao primeiro valor não existe como
correlacionar com a vazão anterior}

        mes_0[j] := vazao[i-1];
        mes_1[j] := vazao[i];
        i:=i+12;
      end;
    npares:=j;
    repeat
    j:=0;
       for i:=1 to npares-1 do
         if mes_0[i]> mes_0[i+1] then
           begin
             auxilia := mes_0[i];
             mes_0[i] := mes_0[i+1];
             mes_0[i+1] := auxilia;
             auxilia := mes_1[i];
             mes_1[i] := mes_1[i+1];
             mes_1[i+1] := auxilia;
             inc(j);
           end;
    until j = 0;

  end;
{-------------------------------------------------------
---------------------
   organiza as vazões em ordem crescente
dos meses Qt-1, Qt-2
   monta um vetor com as  vazões Qt, Qt-1,
Qt-2
   vazao  = vetor de entrada com a série
histórica
   mes    = mês de referência
   ndados = número de vazões
   Qmes   = matriz com as vazões do mes i,
i-1, i-2
   Qord   = vazões dos    meses i, i-1, i-2
ordenadas individualmente em ordem
            crescente (não relacionadas)
   ndados = número de pares de valores
 -------------------------------------------------------
---------------------}
procedure organiza_3D (var vazao: v1;
mes, ndados:integer ;
```

```
                    var Qmes0, Qmes1,Qmes2 :
v1;
                    var Qord0, Qord1,Qord2 :
v1; var npares:integer);
  var
    i,j,k       : integer;
    auxilia     : real;
    arq         : textfile;
    localiza    : array [1..14] of integer;
    Qmes,
    Qord        : a3_v1;
  begin
    for i:=1 to 12 do
      localiza[i+2]:=i;

    localiza[1]:=11;
    localiza[2]:=12;

    for i:=1 to 3 do
      begin
        new(Qmes[i]);
        new(Qord[i]);
      end;
    i := mes;
    j:=0;
    npares:=0;

    while i<= ndados do
     begin
       inc(npares);
       inc(j);
       if i < 3 then
          i:=i+12; { se a vazão for referente ao
mês 1 e 2 não existe como correlacionar
com a vazão anterior}
        for k:=1 to 3 do
          begin
            Qmes[k,j]:=vazao[i-k+1];
            Qord[k,j]:=vazao[i-k+1];
          end;
       i:=i+12;
     end;


    repeat
    i:=0;
       for j:=1 to npares-1 do
         for k:=1 to 3 do
           begin
              if Qord[k,j]> Qord[k,j+1] then
                begin
                  auxilia := Qord[k,j];
                  Qord[k,j] := Qord[k,j+1];
                  Qord[k,j+1] := auxilia;
                  inc(i);
                end;
           end;
    until i = 0;

    for i:=1 to npares do
      begin
        Qmes0[i]:=Qmes[1,i];
        Qmes1[i]:=Qmes[2,i];
        Qmes2[i]:=Qmes[3,i];
        Qord0[i]:=Qord[1,i];
        Qord1[i]:=Qord[2,i];
        Qord2[i]:=Qord[3,i];

      end;
   for i:=1 to 3 do
     begin
       dispose(Qmes[i]);
       dispose(Qord[i]);
     end;

 {              assignfile(arq,dir+'/teste.txt');
                rewrite(arq);
                for i:=1 to npares do
                  begin
                    writeln(arq);
                    for k:=1 to 3 do
                      write(arq,Qord[k,i]);
                  end;
                closefile(arq); }

  end;

{-----------------------------------------------------
---------------------
       Esta função calcula a soma dos erros
ao quadrado da série
-----------------------------------------------------
---------------------}
function soma_erro( Vazao_0, Vazao_1:v1;
npares, grau:integer;
                        Coeficiente:v2):real;
   var
     i,j       : integer;
     erro,
     ycalc     : real;  { valor calculado de y
através dos coeficientes da regressão}

   begin
     erro:=0;
     for i:=1 to npares do
       begin
         ycalc:=0;
           for j:=1 to grau+1 do

ycalc:=ycalc+coeficiente[j]*power(vazao_0[i
],j-1);

          erro:=erro+power(vazao_1[i]-
ycalc,2);
        end;
     soma_erro:=erro;
   end;
{-----------------------------------------------------
---------------------
```

```
      Esta procedure calcula a soma dos
erros ao quadrado para o MARS linear
      Vazao_1 - variável independente (x)
      Vazao_0 - variável dependente   (y)
      npares  - número de pares de valores
      s_erro  -  soma dos erros ao quadrado
da série

-------------------------------------------------------------
-------------------}
procedure s_erro_linear( Vazao_0,
Vazao_1:v1; npares,nt:integer;
p_pontos:vetd;
                    var s_erro:real);
  var
    i, j      : integer;
    a, b,       { coeficientes da reta linear }
    vazao     : real;
    encontrado: boolean;
  begin
    s_erro:=0;

    for i:=1 to npares do
      begin
        j:=2;
        encontrado:=false;
        repeat
          if (vazao_1[i]<=p_pontos[j]) or (j =
nt) then
            begin
              encontrado:=true;
              a:=p_pontos[j+nt];
              b:=(p_pontos[j+nt+1]-
p_pontos[j+nt])/(p_pontos[j]-p_pontos[j-1]);
              vazao:=a+b*(vazao_1[i]-
p_pontos[j-1]);
            end
            else
              inc(j);

        until encontrado;

        s_erro:=s_erro+power(vazao_0[i]-
vazao,2);
       end;
  end;
{-------------------------------------------------------------
---------------------
      Esta procedure calcula a soma dos
erros ao quadrado para o MARS linear 3D
      Vazao_1 - variável independente (x)
      Vazao_0 - variável dependente   (y)
      npares  - número de pares de valores
      s_erro  -  soma dos erros ao quadrado
da série

-------------------------------------------------------------
-------------------}
 procedure s_erro_linear_3D( Vazao
:a3_v1;
                    npares,nt:integer;
p_pontos:vetd; var s_erro:real);
  var
    i, j, k,
    k1,m,n,
    int_localiza
          : integer;
    localiza  : boolean;
    y,        { localiza se o ponto z(Qt) esta
no quadrante superior ou inferior}
    b       : real;
  {Estas variáveis são utilizadas para o
cálculo da vazão simulada Qt}
    x0, x1, x2, x3,
    y0, y1, y2, y3,
    z0, z1, z2, z3,
    Qprevisto,
   { variáveis para o cálculo das
coordenadas do baricentro do quadrante}
    xb, yb,          {x do baricentro, y do
baricentro}
    distancia,       { distância do quadrante
em relação a Qt-1 e Qt-2}
    distancia1
         : real;
  begin
    s_erro:=0;
    for k:=1 to npares do
      begin
        localiza:=false;
        int_localiza:=0;

        i:=1;
        repeat
          if (p_pontos[i] >= vazao[2,k]) and
(p_pontos[i-1] < vazao[2,k])
            then
              localiza:=true
            else
              inc(i);
        until (i > nt) or (localiza);

        if not localiza then
          int_localiza:=1;

        localiza:=false;
        j:=1;

        repeat
          if (p_pontos[nt+j] >= vazao[3,k])
and (p_pontos[nt+j-1] < vazao[3,k])
            then
              localiza:=true
            else
              inc(j)
        until (j > nt+1) or (localiza);
        if not localiza then
          int_localiza:=int_localiza+2;

        y:= (p_pontos[nt+j]-p_pontos[nt+j-1])/
```

```
            (p_pontos[i]-p_pontos[i-
1])*(vazao[2,k]-p_pontos[i-1])
            +p_pontos[nt+j-1];

         if int_localiza > 0 then { a vazão esta
fora do greide de coeficientes}
            begin
              localiza:=false;
              { calcula qual o quadrante com a
menor distância em relação a Qt-1 Qt-2}

distancia1:=99999999999999.99999;
              { calculo da distancia para o
quadrante inferior ex. 10-11-16}
              for m:=1 to nt do  { eixo Qt-1 }
                for n:=1 to nt do  { eixo Qt-2 }
                  begin
                    xb:=p_pontos[n-
1]+(p_pontos[n]-p_pontos[n-1])*2/3;

yb:=p_pontos[nt+m]+(p_pontos[nt+m+1]-
p_pontos[nt+m])*1/3;

distancia:=power(power(vazao[2,k]-xb,2)+
                    +power(vazao[3,k]-
yb,2),0.5);
                    if distancia < distancia1
then
                      begin
                        x0 := vazao[2,k];
                        y0 := vazao[3,k];
                        x1 := p_pontos[m-1];
                        y1 := p_pontos[nt+n];
                        z1 :=
p_pontos[(2*nt+1)+(n-1)*(nt+1)+m];
                        x2 := p_pontos[m];
                        y2 := p_pontos[nt+n];
                        z2 :=
p_pontos[(2*nt+1)+(n-1)*(nt+1)+m+1];
                        x3 := p_pontos[m];
                        y3 := p_pontos[nt+n+1];
                        z3 :=
p_pontos[(2*nt+1)+(n)*(nt+1)+m+1];
                        distancia1:=distancia;
                      end;
                  end;
              { calculo da distancia para o
quadrante superior ex. 10-15-16}
              for m:=1 to nt do  { eixo Qt-1 }
                for n:=1 to nt do  { eixo Qt-2 }
                  begin
                    xb:=p_pontos[n-
1]+(p_pontos[n]-p_pontos[n-1])*1/3;

yb:=p_pontos[nt+m]+(p_pontos[nt+m+1]-
p_pontos[nt+m])*2/3;

distancia:=power(power(vazao[2,k]-xb,2)+
                    +power(vazao[3,k]-
yb,2),0.5);
                    if distancia < distancia1
then
                      begin
                        x0 := vazao[2,k];
                        y0 := vazao[3,k];
                        x1 := p_pontos[m-1];
                        y1 := p_pontos[nt+n];
                        z1 :=
p_pontos[(2*nt+1)+(n-1)*(nt+1)+m];
                        x2 := p_pontos[m];
                        y2 := p_pontos[nt+n+1];
                        z2 :=
p_pontos[(2*nt+1)+(n)*(nt+1)+m+1];
                        x3 := p_pontos[m-1];
                        y3 := p_pontos[nt+n+1];
                        z3 :=
p_pontos[(2*nt+1)+(n)*(nt+1)+m];
                        distancia1:=distancia;
                      end;
                  end;

            end;
{se localiza  = false quer dizer que a vazao
não esta no intervalo superior}
          if localiza then
             if  (y <= vazao[3,k]) then
             { se esta condição é valida
significa que a vazao (Qt-1 e Qt-2)
              esta na parte superior do
quadrante }
                begin
                  x0 := vazao[2,k];
                  y0 := vazao[3,k];
                  x1 := p_pontos[i-1];
                  y1 := p_pontos[nt+j-1];
                  z1 := p_pontos[(2*nt+1)+(j-
2)*(nt+1)+i];
                  x2 := p_pontos[i];
                  y2 := p_pontos[nt+j];
                  z2 := p_pontos[(2*nt+1)+(j-
1)*(nt+1)+i+1];
                  x3 := p_pontos[i-1];
                  y3 := p_pontos[nt+j];
                  z3 := p_pontos[(2*nt+1)+(j-
1)*(nt+1)+i];

                end
                else
                  begin
                  x0 := vazao[2,k];
                  y0 := vazao[3,k];
                  x1 := p_pontos[i-1];
                  y1 := p_pontos[nt+j-1];
                  z1 := p_pontos[(2*nt+1)+(j-
2)*(nt+1)+i];
                  x3 := p_pontos[i];
                  y3 := p_pontos[nt+j];
                  z3 := p_pontos[(2*nt+1)+(j-
1)*(nt+1)+i+1];
                  x2 := p_pontos[i];
```

```
                y2 := p_pontos[nt+j-1];
                z2 := p_pontos[(2*nt+1)+(j-
2)*(nt+1)+i+1];

                end;
          Qprevisto:=((x0-x1)*(y2-y1)*(z3-
z1)+(y0-y1)*(z2-z1)*(x3-x1)-
                  (y3-y1)*(z2-z1)*(x0-x1)-(z3-
z1)*(x2-x1)*(y0-y1))/
                  ((x2-x1)*(y3-y1)-(x3-x1)*(y2-
y1))+z1;
          s_erro:=s_erro+(power(qprevisto-
vazao[1,k],2));
        end;
  end;


{-----------------------------------------------------
---------------------
      Esta procedure calcula os coeficientes
do modelo MARS pelo método
      dos mínimos quadrados para um
polinômio de grau n
      Vazao_0 - variável independente (x)
      Vazao_1 - variável dependente   (y)
      npares  - número de pares de valores
      grau    - grau do poninômio
      coeficiente - coeficientes do polinômio
Pn(x) = An.X^n+An-1.X^n-1+...+a0
      s_erro  -  soma dos erros ao quadrado
da série
      limite grau  = 50
-----------------------------------------------------
--------------------}
procedure coef_MARS_Min_quad(
Vazao_0, Vazao_1:v1; npares,
grau:integer;
                        var Coeficiente:v2; var
s_erro:real);
  var
    i,j      : integer;
    x,yx     : ^v2;  { calcula o valor de x^n
x[1] = x^0, x[2]=x^1}
    a        : ^mat2020; { coeficientes da
matriz para gaujor }
    retorno  : ^vet20;   { retorno dos
coeficientes de gaujor }
    rc       : byte;

  begin
    new(a);       { inicialização das
variáveis dinâmicas }
    new(retorno);
    new(x);
    new(yx);
    for i:=1 to 300 do
      new(a^[i]);

    for i:=1 to 300 do  { zera matriz }
      for j:=1 to 300 do
        a[i][j]:=0;

    for i:=1 to 100 do
      begin
        x^[i]:=0;
        yx^[i]:=0;
        coeficiente[i]:=0;
      end;

    for i:=1 to (2*(grau+1)) do
      for j:=1 to npares do  { somatória }
        x^[i] := x^[i] + power(vazao_0[j],(i-
1));

    for i:=1 to (grau+1) do
      for j:=1 to npares do  { somatória }
        yx^[i]:= yx^[i] +
vazao_1[j]*power(vazao_0[j],(i-1));

    for i:=1 to grau+1 do
      begin
        for j:=1 to grau+1 do
          a[i]^[j]:=x^[i-1+j];

        a[i]^[grau+2]:=yx^[i];
      end;


ucoefmult.selgau(grau+1,a^,retorno^,rc);

    for i:=1 to grau+1 do
      coeficiente[i]:=retorno^[i];

    s_erro:=soma_erro(Vazao_0,
Vazao_1,npares, grau,Coeficiente);
     dispose(retorno);  { destrói as variáveis
dinâmicas }
     dispose(x);
     dispose(yx);
     for i:=1 to 300 do
       dispose(a^[i]);
     dispose(a);

  end;
{-----------------------------------------------------
---------------------
      Esta procedure faz a previsão
utilizando o modelo MARS linear 1
dimensão
-----------------------------------------------------
--------------------}

procedure prev_MARS_linear;
  var
    nvaz,
    i, j,  mes,
    mes_pre                { mes de previsão }
                  : integer;
    codusi,
    coeficiente      : string;
```

```
      arq_usi,
      arq_coe,
      arq_pre          : textfile;
      vazao            : ^v1;
      coef             : array[1..12,0..13] of
double;
      vaz_prev         : array [1..12] of real;
      a,b,
      aux_vazao        { capta o último valor
da vazão observada}
                       : real;
      localiza_par     : boolean;      { indica
se o parâmetro foi encontrado}
   const
      nt: integer = numero_trechos;
   begin
    assignfile(arq_usi,dir+'\usinas.txt');   {
Arquivo contendo o código das usinas
selecionadas as usinas}
     reset(arq_usi);
     new(vazao);
     while not eof (arq_usi) do
       begin
          readln(arq_usi,codusi);
          ler_dad(codusi, 2, vazao^, nvaz);    {
lê a série de vazões }

assignfile(arq_coe,dir+'\MARS\'+codusi+'.M
ARS'); { lê os coeficientes da regressão
MARS }
          reset(arq_coe);
          for mes:=1 to 12 do
             for i:=0 to 13 do
                coef[mes,i]:=0;
          mes:=0;

          while not eof (arq_coe) do  { lê os
coeficientes do modelo MARS }
             begin
                inc(mes);
                     readln(arq_coe,coeficiente);
                     for i:=0 to nt do
                        begin

readln(arq_coe,coeficiente);

coef[mes,i]:=strtofloat(copy(coeficiente,5,22
));

coef[mes,i+nt+1]:=strtofloat(copy(coeficient
e,28,22));

                        end;
             end;
          closefile(arq_coe);
          mes_pre:= nvaz div 12;
          mes_pre:=nvaz mod mes_pre+1;
          for i:=1 to 12 do  { zera o vetor de
previsão}
             vaz_prev[i]:=0;

          assignfile(arq_pre,
dir+'\MARS\'+codusi+'.pre'); { grava a
previsão}
          rewrite(arq_pre);
          aux_vazao:=vazao^[nvaz];
          for i:=1 to 12 do
            begin
              j:=1;
              localiza_par:=false;
              if mes_pre > 12 then
                mes_pre:=1;
              repeat
                 if i=1 then
                   begin
                      if aux_vazao <=
coef[mes_pre,j]
                        then
                          localiza_par:=true
                        else
                          inc(j);
                   end
                   else
                     begin
                       if vaz_prev[i-1] <=
coef[mes_pre,j]
                          then
                            localiza_par:=true
                          else
                            inc(j);
                     end;
              until (localiza_par) or (j =
numero_trechos);

              a:=coef[mes_pre,j+nt];
              b:=(coef[mes_pre,j+nt+1]-
coef[mes_pre,j+nt]) /
                   (coef[mes_pre,j]-
coef[mes_pre,j-1]);
              if i=1
                then
                  vaz_prev[i]:=a+b*(aux_vazao-
coef[mes_pre,j-1])
                else
                  vaz_prev[i]:=a+b*(vaz_prev[i-
1]-coef[mes_pre,j-1]);

              if vaz_prev[i]< 0 then
                 begin
                   localiza_par:=false;
                   j:=1;
                   repeat
                      a:=coef[mes_pre,j+nt];
                      b:=(coef[mes_pre,j+nt+1]-
coef[mes_pre,j+nt]) /
                           (coef[mes_pre,j]-
coef[mes_pre,j-1]);
                      if i=1
                        then
```

```
vaz_prev[i]:=a+b*(aux_vazao-
coef[mes_pre,j-1])
                        else

vaz_prev[i]:=a+b*(vaz_prev[i-1]-
coef[mes_pre,j-1]);
                      if vaz_prev[i]> 0 then
                        localiza_par:=true;
                      inc(j);
                    until  (localiza_par) or (j =
numero_trechos)
                  end;
              write(arq_pre,mes_pre:3);
              writeln(arq_pre,vaz_prev[i]:10:2);
              inc(mes_pre);

            end;   { fim do for i }
          closefile(arq_pre);
      end; { fim do while lê usinas para
previsão }
    closefile(arq_usi);
    dispose(vazao);
  end;
{-------------------------------------------------------
--------------------
       Esta procedure faz a previsão
utilizando o modelo MARS linear 3
dimensões
-------------------------------------------------------
--------------------}
procedure prev_MARS_linear_3d;
  var
    nvaz,
    i, j,  mes,
    mes_pre,                { mes de previsão
}
    n_equacoes
                    : integer;
    codusi,
    coeficiente      : string;
    arq_usi,
    arq_coe,
    arq_pre          : textfile;
    vazao            : ^v1;
    coef             : array[1..12,0..50] of
double;
    vaz_prev         : array [1..12] of real;
    a,b,
    aux_vazao,         { capta o último valor
da vazão observada}
    lixo             : real;
    localiza_par     : boolean;      { indica
se o parâmetro foi encontrado}
    vazao1_2         : ^a3_v1;
    coeficiente_MARS  :vetd;
  const
    nt: integer = numero_trechos;
  begin
    assignfile(arq_usi,dir+'\usinas.txt');   {
Arquivo contendo o código das usinas
selecionadas as usinas}
    reset(arq_usi);
    new(vazao);
    while not eof (arq_usi) do
      begin
        readln(arq_usi,codusi);
        ler_dad(codusi, 2, vazao^, nvaz);    {
lê a série de vazões }

assignfile(arq_coe,dir+'\MARS_3d\'+codusi
+'.MARS'); { lê os coeficientes da regressão
MARS }
        reset(arq_coe);
        for mes:=1 to 12 do
          for i:=0 to (2*(nt-
1)+trunc(power((nt+1),2))) do
            coef[mes,i]:=0;
        mes:=0;

        while not eof (arq_coe) do  { lê os
coeficientes do modelo MARS }
          begin
            inc(mes);
                readln(arq_coe,coeficiente);
                for i:=0 to (2*(nt-
1)+trunc(power((nt+1),2))+3) do
                  begin

readln(arq_coe,coeficiente);

coef[mes,i]:=strtofloat(copy(coeficiente,11,2
8));

                  end;
          end;
        closefile(arq_coe);
        mes_pre:= nvaz div 12;
        mes_pre:=nvaz mod mes_pre+1;
        for i:=1 to 12 do  { zera o vetor de
previsão}
          vaz_prev[i]:=0;

        assignfile(arq_pre,
dir+'\MARS_3d\'+codusi+'.pre'); { grava a
previsão}
        rewrite(arq_pre);

        for i:=1 to 12 do
          begin
            new(vazao1_2);
            for j:=1 to 3 do
              new(vazao1_2^[j]);
            n_equacoes:= 2*(nt-
1)+trunc(power((nt+1),2))+3;
            for j:=1 to n_equacoes+1 do
              coeficiente_MARS[j-1] :=
coef[mes_pre,j-1];
```

```
              if i = 1 then  { válido só para o
primeiro mês}
                for j:=1 to 2 do

Vazao1_2[j+1]^[1]:=vazao^[nvaz-j+1];
               a:=Vazao1_2[2]^[1];
               b:=Vazao1_2[3]^[1];
              Vazao1_2[1]^[1]:=0;
              lixo:=a+b;
              s_erro_linear_3D(
Vazao1_2^,1,nt,coeficiente_MARS, lixo);
              vaz_prev[i]:=power(lixo,0.5);
              write(arq_pre,mes_pre:3);
              writeln(arq_pre,vaz_prev[i]:10:2);
              { faz com que a vazão prevista
se torne Qt-1 e Qt-1 = Qt-2}
              Vazao1_2[2]^[3] :=
Vazao1_2[2]^[2];
              Vazao1_2[2]^[2] := vaz_prev[i];

              for j:=1 to 3 do
                dispose(vazao1_2[j]);
              if mes_pre = 12 then
                mes_pre:=1
                else
                  inc(mes_pre);
            end;   { fim do for i }
          closefile(arq_pre);
       end; { fim do while lê usinas para
previsão }
    closefile(arq_usi);
    dispose(vazao);
   end;

{-----------------------------------------------------------
----------------------
   Esta procedure retorna o valor máximo e
mínimo da vazão segundo a distibuição
   Log-normal a 3 parâmetros para um
tempo de recorrência = +-5000 anos
-----------------------------------------------------------
----------------------}

procedure log_normal_3(vazao :
v1;nun_pontos:integer; var max, min:real);
   var
     serie_vazao       : array of double;
     media, variancia,
     skew, kurtosis,
     m3, m4            : extended;
     i                 : integer;
   begin
     SetLength(serie_vazao, nun_pontos);

     for i:=0 to nun_pontos-1 do
       serie_vazao[i] := log10(vazao[i+1]);

     MomentSkewKurtosis(serie_vazao,
media, variancia, m3, m4, skew, kurtosis);
max:=power(10,media+power(variancia,0.5
)*3.5);
     min:=power(10,media-
power(variancia,0.5)*3.5);
     if min<0 then
       min:=0;
   end;
end.
```

F.6 ANÁLISE DOS RESULTADOS

```
unit U_Afericao;

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs,
  StdCtrls, Mask, math;

type
  TFAfericao = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    E_Ano_ini_cal: TMaskEdit;
    E_Ano_fim_cal: TMaskEdit;
    E_Lag_cal: TMaskEdit;
    E_Ano_ini_pre: TMaskEdit;
    E_Ano_fim_pre: TMaskEdit;
    E_Lag_pre: TMaskEdit;
    BExecutar: TButton;
    procedure BExecutarClick(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;


var
  FAfericao: TFAfericao;

implementation

uses
ulinear, UKriging,
UCoefMult,uprevmqx,fmxutils, U_Mars,
Mars, mars_3d;

{$R *.DFM}
type
   vaz_usinas = record
      codigo    : string[8];
      vazao     : array[1..12] of real;
      end;
   vet_vaz_usinas  = array [1..100] of
^vaz_usinas;
   vet_name_usinas = array [1..100] of
string[8];
procedure erro_variancia(vaz_pre_otimo,
vaz_pre_din,vaz_pre_linear,vaz_pre_mars,
                         vaz_pre_mars_3d,
vaz_rel_otimo,vaz_rel_din,

vaz_rel_linear,vaz_rel_mars,vaz_rel_mars_
3d,
                         vaz_obs:vet_vaz_usinas
;cont_pre:integer;
                         nome_arquivo:string);
var
i,j            : integer;
arq            : textfile;
soma_otimo,
soma_din,
soma_linear,
soma_mars,
soma_mars_3d,
soma_obs,
media_otimo,
media_din,
media_linear,
media_mars,
media_mars_3d,
media_obs,
r_otimo,                    { coeficiente de
correlação ótimo }
r_din,                     { coeficiente de
correlação dinâmico }
r_linear,                   { coeficiente de
correlação linear }
r_mars,                     { coeficiente de
correlação Mars }
r_mars_3d,                    { coeficiente de
correlação Mars 3d }
r_obs,                     { coeficiente de
correlação observado }
s_otimo,                    { soma  otimo }
s_din,                     { soma  dinâmico }
s_linear,                   { soma  linear }
s_mars,                     { soma  mars }
s_mars_3d,                    { soma  mars 3d }
s_obs,                     { soma observado }
e_otimo,                    { erro  otimo }
e_din,                     { erro  dinâmico }
e_linear,                   { erro  linear }
e_mars,                     { erro mars }
e_mars_3d,                    { erro mars 3d }
COE                        { coeficiente de
eficiência }
               : real;
begin
   assignfile(arq,nome_arquivo);
   if fileExists(nome_arquivo)
      then append(arq)
      else rewrite(arq);

   writeln(arq,'Resumo da média dos erros e
coeficiente de correlação');
   writeln(arq,'Método ótimo, Método
dinâmico, Método linear, Método Mars');
```

```
        for i:=1 to 12 do
          begin
            soma_otimo:=0;
            soma_din:=0;
            soma_linear:=0;
            soma_mars:=0;
            soma_mars_3d:=0;
            soma_obs:=0;
            r_otimo:=0;
            r_din:=0;
            r_linear:=0;
            r_mars:=0;
            r_mars_3d:=0;
            r_obs:=0;
            s_otimo:=0;
            s_din:=0;
            s_linear:=0;
            s_mars:=0;
            s_mars_3d:=0;
            s_obs:=0;
            e_otimo:=0;
            e_din:=0;
            e_linear:=0;
            e_mars:=0;
            e_mars_3d:=0;
            for j:=1 to cont_pre do
              begin

soma_otimo:=soma_otimo+vaz_pre_otimo[j
].vazao[i];

soma_din:=soma_din+vaz_pre_din[j].vazao
[i];

soma_linear:=soma_linear+vaz_pre_linear[j
].vazao[i];

soma_mars:=soma_mars+vaz_pre_mars[j].
vazao[i];

soma_mars_3d:=soma_mars_3d+vaz_pre_
mars_3d[j].vazao[i];

soma_obs:=soma_obs+vaz_obs[j].vazao[i];
              end;

media_otimo:=soma_otimo/cont_pre;
            media_din:=soma_din/cont_pre;

media_linear:=soma_linear/cont_pre;

media_mars:=soma_mars/cont_pre;

media_mars_3d:=soma_mars_3d/cont_pre;
            media_obs:=soma_obs/cont_pre;

            for j:=1 to cont_pre do
              begin
                { calcula coeficiente de
correlação entre as previsões e os valores
observados }
                r_otimo := r_otimo +
(vaz_pre_otimo[j].vazao[i]- media_otimo)*
                    (vaz_obs[j].vazao[i]-
media_obs);
                s_otimo := s_otimo +
power(vaz_pre_otimo[j].vazao[i]-
media_otimo,2);

                r_din := r_din +
(vaz_pre_din[j].vazao[i]- media_din)*
                    (vaz_obs[j].vazao[i]-
media_obs);
                s_din :=s_din +
power(vaz_pre_din[j].vazao[i]-
media_din,2);

                r_linear := r_linear +
(vaz_pre_linear[j].vazao[i]- media_linear)*
                    (vaz_obs[j].vazao[i]-
media_obs);
                s_linear :=s_linear +
power(vaz_pre_linear[j].vazao[i]-
media_linear,2);

                r_mars := r_mars +
(vaz_pre_mars[j].vazao[i]- media_mars)*
                    (vaz_obs[j].vazao[i]-
media_obs);
                s_mars := s_mars +
power(vaz_pre_mars[j].vazao[i]-
media_mars,2);

                r_mars_3d := r_mars_3d +
(vaz_pre_mars_3d[j].vazao[i]-
media_mars_3d)*
                    (vaz_obs[j].vazao[i]-
media_obs);
                s_mars_3d := s_mars_3d +
power(vaz_pre_mars_3d[j].vazao[i]-
media_mars_3d,2);

                s_obs := s_obs +
power(vaz_obs[j].vazao[i]- media_obs,2);

              {
r_otimo:=r_otimo+power((vaz_pre_otimo[j].
vazao[i]-vaz_obs[j].vazao[i]),2);

r_din:=r_din+power((vaz_pre_din[j].vazao[i]
- vaz_obs[j].vazao[i]),2);

r_linear:=r_linear+power((vaz_pre_linear[j].
vazao[i]- vaz_obs[j].vazao[i]),2);

r_mars:=r_mars+power((vaz_pre_mars[j].v
azao[i]- vaz_obs[j].vazao[i]),2);
```

```
r_mars_3d:=r_mars_3d+power((vaz_pre_m
ars_3d[j].vazao[i]- vaz_obs[j].vazao[i]),2);

r_obs:=r_obs+power((vaz_obs[j].vazao[i]-
media_obs),2);
              }

e_otimo:=e_otimo+abs(vaz_pre_otimo[j].va
zao[i]- vaz_obs[j].vazao[i])
                        /vaz_obs[j].vazao[i];

e_din:=e_din+abs(vaz_pre_din[j].vazao[i]-
vaz_obs[j].vazao[i])
                        /vaz_obs[j].vazao[i];

e_linear:=e_linear+abs(vaz_pre_linear[j].va
zao[i]- vaz_obs[j].vazao[i])
                        /vaz_obs[j].vazao[i];

e_mars:=e_mars+abs(vaz_pre_mars[j].vaz
ao[i]- vaz_obs[j].vazao[i])
                        /vaz_obs[j].vazao[i];

e_mars_3d:=e_mars_3d+abs(vaz_pre_mar
s_3d[j].vazao[i]- vaz_obs[j].vazao[i])
                        /vaz_obs[j].vazao[i];

              end;

          e_otimo:=e_otimo/cont_pre;
          e_din:=e_din/cont_pre;
          e_linear:=e_linear/cont_pre;
          e_mars:=e_mars/cont_pre;
          e_mars_3d:=e_mars_3d/cont_pre;


write(arq,(format('%4.2f',[e_otimo])):10);

coe:=r_otimo/power(s_otimo*s_obs,0.5);
//1-r_otimo/r_obs;
          write(arq,format('%4.2f',[coe]):10);


write(arq,format('%4.2f',[e_din]):10);

coe:=r_din/power(s_din*s_obs,0.5);
//coe:=1-r_din/r_obs;
          write(arq,format('%4.2f',[coe]):10);


write(arq,format('%4.2f',[e_linear]):10);

coe:=r_linear/power(s_linear*s_obs,0.5);
//coe:=1-r_linear/r_obs;
          write(arq,format('%4.2f',[coe]):10);


write(arq,format('%4.2f',[e_mars]):10);
          if  s_mars <> 0 then

coe:=r_mars/power(s_mars*s_obs,0.5)
//coe:=1-r_mars/r_obs;
              else
                coe:=9999;
          write(arq,format('%4.2f',[coe]):10);


write(arq,format('%4.2f',[e_mars_3d]):10);
          if s_mars_3d <> 0 then

coe:=r_mars_3d/power(s_mars_3d*s_obs,0
.5) //coe:=1-r_mars_3d/r_obs;
              else
                coe:=9999;

writeln(arq,format('%4.2f',[coe]):10);


        end;
  closefile(arq);
end;
procedure
TFAfericao.BExecutarClick(Sender:
TObject);
var
k             : byte;       { le dado
descartável}
ano_ini_cal,                { ano inicial de
calibração }
ano_fim_cal,                { ano final de
calibração }
lag_cal,                    { lag de calibração }
ano_ini_pre,                { ano inicial de
previsão }
ano_fim_pre,                { ano final de
previsão }
lag_pre,                    { lag de previsão }
ano_previsao,
cont_usinas,                { conta o número
de usinas existentes }
cont_pre,                   { conta o número
de previsões realizadas }
ano_leitura,                { ano de leitura }
cont,
nvaz,i,j,
cont_ano,                   { conta o número
de anos }
lag_cal_pre                 { lag entre a
calibração e a previsão inicial }
              : integer;
arq,
arq_cor       : textfile;
arq_text,
dir,
arq_usi_comp,               { arquivo
contendo todas as usinas }
arq_usi_pre,                { arquivo com as
usinas utilizadas na previsão }
saux
              : string;
```

```
nome_usi       : ^vet_name_usinas;  {
armazena o código das usinas cadastradas
p/ previsão}

vaz_obs,                    { vetor com as
vazões observadas }
vaz_pre_otimo,              { vetor com as
vazões previstas pelo método ótimo }
vaz_pre_linear,             { vetor com as
vazões previstas pelo método linear }
vaz_pre_din,                { vetor com as
vazões previstas pelo método dinâmico }
vaz_pre_mars,               { vetor com as
vazões previstas pelo método Mars }
vaz_pre_mars_3d,            { vetor com as
vazões previstas pelo método Mars 3d }
vaz_rel_otimo,              { vetor com a
relação entre q previstas e q obs pelo
método ótimo }
vaz_rel_linear,             { vetor com a
relação entre q previstas e q obs pelo
método linear }
vaz_rel_din,                { vetor com a
relação entre q previstas e q obs pelo
método dinâmico }
vaz_rel_mars,               { vetor com a
relação entre q previstas e q obs pelo
método Mars }
vaz_rel_mars_3d             { vetor com a
relação entre q previstas e q obs pelo
método Mars 3d }
            : ^vet_vaz_usinas;

begin
dir:= getcurrentdir;
ano_ini_cal:=strtoint(e_ano_ini_cal.text);
ano_fim_cal:=strtoint(e_ano_fim_cal.text);
lag_cal:=strtoint(e_lag_cal.text);
ano_ini_pre:=strtoint(e_ano_ini_pre.text);
ano_fim_pre:=strtoint(e_ano_fim_pre.text);
lag_pre:=strtoint(e_lag_pre.text);
ano_previsao:=ano_ini_pre;
lag_cal_pre:=ano_ini_pre-ano_fim_cal;
while ano_previsao <= ano_fim_pre do
  begin

     new(vaz_obs);
     new(vaz_pre_otimo);
     new(vaz_pre_linear);
     new(vaz_pre_din);
     new(vaz_pre_mars);
     new(vaz_pre_mars_3d);
     for j:=1 to 50 do
       begin
         new(vaz_obs^[j]);
         new(vaz_pre_otimo^[j]);
         new(vaz_pre_linear^[j]);
         new(vaz_pre_din^[j]);
         new(vaz_pre_mars^[j]);
         new(vaz_pre_mars_3d^[j]);
       end;
         new(nome_usi);

     { altera os parâmetros de calibração e
previsão no arquivo de configuração}
     arq_usi_comp:=dir+'/usinasa.txt';
     arq_usi_pre:=dir+'/usinas.txt';
     copyfile(arq_usi_comp,arq_usi_pre);  {
copia para arquivo de previsão todas as
usinas cadastradas}
     assignfile(arq,dir+'\configuracao.ini');
     rewrite(arq);
     write(arq,'anoinicial':15);
     writeln(arq,ano_ini_cal:10);
     write(arq,'anofinal':15);
     writeln(arq,ano_fim_cal:10);
     write(arq,'anoprevisao':15);
     writeln(arq,ano_previsao:10);
     closefile(arq);

     {aciona funções de calibração }
     uKriging.Coef_Krigign;
     ulinear.coeficiente_linear;
     UCoefMult.matrizk;
        { caso a série seja menor que 30 o
método mars e mars 3d não é executado}
     if (ano_fim_cal - ano_ini_cal)>=40 then
       begin
         Mars_3d.coef_mars_3d;          {
Calibração do modelo Mars 3d}
         Mars.coef_Mars;               {
Calibração do modelo Mars  }
       end;

     { aciona funções de previsão }
     uprevmqx.naolinear;            { Previsão
de vazões utilizando funções
multiquadráticas}
     ulinear.prevlinear;            { Previsão de
vazões utilizando modelo AR(6)}
     UKriging.previsao_krigign;     {
Previsão de vazões utilizando funções
multiquadráticas }


        { caso a série seja menor que 30 o
método mars e mars 3d não é executado}
     if (ano_fim_cal - ano_ini_cal)>=40 then
       begin
         U_Mars.prev_mars_linear;       {
Previsão de vazões utilizando o modelo
Mars }
         U_Mars.prev_mars_linear_3d;
{ Previsão de vazões utilizando o modelo
Mars }
       end;

     { lê o código das usinas no cadastro de
previsão }
     assignfile(arq,dir+'/usinas.txt');
     reset(arq);
```

```
      cont_usinas:=0;
      while not eof (arq) do
        begin
          inc(cont_usinas);
          readln(arq,nome_usi[cont_usinas]);
        end;
      closefile(arq);

      cont_pre:=1;

      while cont_pre<=cont_usinas do
        begin
          {Lê a série observadas no ano de
previsão }  {Ok}

assignfile(arq,dir+'\vazoes\'+nome_usi[cont
_pre]+'.prn');
          reset(arq);
          nvaz:=1;
          readln(arq);  { lê cabeçalho }
          readln(arq);  { lê cabeçalho }

vaz_obs[cont_pre].codigo:=nome_usi[cont_
pre];

vaz_pre_linear^[cont_pre].codigo:=nome_u
si[cont_pre];

vaz_pre_otimo^[cont_pre].codigo:=nome_u
si[cont_pre];

vaz_pre_din^[cont_pre].codigo:=nome_usi[
cont_pre];
           repeat
             read(arq,ano_leitura);
               if ano_leitura = ano_previsao
then
                 begin
                   cont:=0;
                   repeat
                     inc(cont);
                     if cont>1 then

vaz_obs^[cont_pre].vazao[cont]:=vaz_obs^[
cont_pre].vazao[cont-1];

read(arq,vaz_obs^[cont_pre].vazao[cont]);
                   until eof(arq) or (cont = 12);
                 end;
                   readln(arq);
           until eof (arq);
           closefile(arq);

           { Lê as vazões previstas pelo
método linear }
          cont:=0;

assignfile(arq,dir+'\linear\'+nome_usi[cont_
pre]+'.rel');
          reset(arq);
          if
(fileexists(dir+'\linear\'+nome_usi[cont_pre]
+'.rel')) then
            begin
              for i :=1 to 30 do  { Lê
parâmetros iniciais não usados }
                readln(arq);

              for i := 1 to 12 do   { Lê as
previsões lineares }
              begin
                readln(arq,arq_text);

vaz_pre_linear^[cont_pre].vazao[i]:=strtoflo
at(copy(arq_text,11,9));
              end;

          end;
          closefile(arq); { fim da leitura da
preisão linear }

          { Lê previsão de Kriging}

assignfile(arq,dir+'\Kriging\'+nome_usi[cont
_pre]+'.pre');
          reset(arq);

          for i:=1 to 12 do

readln(arq,k,vaz_pre_otimo^[cont_pre].vaza
o[i]);

          closefile(arq);  { fim da leitura
previsão de kriging}

          { Lê previsão de Mars}
                  { caso a série seja menor
que 30 o método mars não é executado}
          if (ano_fim_cal - ano_ini_cal)>=40
then
            begin

assignfile(arq,dir+'\Mars\'+nome_usi[cont_p
re]+'.pre');
              reset(arq);

              for i:=1 to 12 do

readln(arq,k,vaz_pre_mars^[cont_pre].vaza
o[i]);

              closefile(arq);  { fim da leitura
previsão de Mars}
            end;
          { Lê previsão de Mars 3d }
                  { caso a série seja menor
que 30 o método mars não é executado}
          if (ano_fim_cal - ano_ini_cal)>=40
then
```

```
            begin

assignfile(arq,dir+'\Mars_3d\'+nome_usi[co
nt_pre]+'.pre');
              reset(arq);

              for i:=1 to 12 do

readln(arq,k,vaz_pre_mars_3d^[cont_pre].v
azao[i]);

              closefile(arq);  { fim da leitura
previsão de Mars}
            end;

          { Lê a previsão dinâmica }

assignfile(arq,dir+'\naolinear\'+nome_usi[co
nt_pre]+'.txx');
          reset(arq);
          for i:=1 to 5 do
            readln(arq);

          for i:=1 to 12 do

readln(arq,k,k,vaz_pre_din^[cont_pre].vaza
o[i]);

          inc(cont_pre); { conta o número de
usinas lidas }
          closefile(arq);
        end; { dim do while
cont_pre<=cont_usinas}



        for i:=1 to cont_usinas do
          begin
            { salva o resumo das previsões em
um arquivo txt }

assignfile(arq,dir+'/afericao/lag_'+inttostr(la
g_cal_pre)+'/'+(nome_usi[i])+'.pre');
            if
fileExists(dir+'/afericao/lag_'+inttostr(lag_cal
_pre)+'/'+(nome_usi[i])+'.pre') then
              append(arq)
            else
              rewrite(arq);

            if i = 1 then { imprime cabeçalho}
              begin
                writeln(arq,'----------------------------
----------------------');
                writeln(arq,'   Previsão de
vazões ');
                writeln(arq,'   Ano de calibagem
'+inttostr(ano_ini_cal)+' - '
                      + inttostr(ano_fim_cal));
              end;
            writeln(arq);

write(arq,vaz_pre_otimo^[i].codigo:10);
            write(arq,ano_previsao:5);
            write(arq,'Otimo':10);
            for j:=1 to 12 do

write(arq,vaz_pre_otimo^[i].vazao[j]:10:2);

            writeln(arq);

write(arq,vaz_pre_otimo^[i].codigo:10);
            write(arq,ano_previsao:5);
            write(arq,'E. % Otimo':10);
            for j:=1 to 12 do

write(arq,((vaz_pre_otimo^[i].vazao[j]-
vaz_obs^[i].vazao[j])*100/vaz_obs^[i].vazao
[j]):10:2);

            writeln(arq);
            write(arq,nome_usi[i]:10);
            write(arq,ano_previsao:5);
            write(arq,'Dinâmica':10);
            for j:=1 to 12 do

write(arq,vaz_pre_din^[i].vazao[j]:10:2);

            writeln(arq);
            write(arq,nome_usi[i]:10);
            write(arq,ano_previsao:5);
            write(arq,'E. % Dinâ.':10);
            for j:=1 to 12 do

write(arq,((vaz_pre_din^[i].vazao[j]-
vaz_obs^[i].vazao[j])*100/vaz_obs^[i].vazao
[j]):10:2);

            writeln(arq);
            write(arq,nome_usi[i]:10);
            write(arq,ano_previsao:5);
            write(arq,'Linear':10);
            for j:=1 to 12 do

write(arq,vaz_pre_linear^[i].vazao[j]:10:2);

            writeln(arq);
            write(arq,nome_usi[i]:10);
            write(arq,ano_previsao:5);
            write(arq,'E. % Line.':10);
            for j:=1 to 12 do

write(arq,((vaz_pre_linear^[i].vazao[j]-
vaz_obs^[i].vazao[j])*100/vaz_obs^[i].vazao
[j]):10:2);

            writeln(arq);
            write(arq,nome_usi[i]:10);
            write(arq,ano_previsao:5);
            write(arq,'Mars':10);
            for j:=1 to 12 do
```

```
          { caso a série seja menor que 30 o
método mars não é executado}
             if (ano_fim_cal -
ano_ini_cal)>=40 then

write(arq,vaz_pre_mars^[i].vazao[j]:10:2)
               else
                 write(arq,0:10);

          writeln(arq);
          write(arq,nome_usi[i]:10);
          write(arq,ano_previsao:5);
          write(arq,'E. %  Mars':10);
          for j:=1 to 12 do
          { caso a série seja menor que 30 o
método mars não é executado}
             if (ano_fim_cal -
ano_ini_cal)>=40 then

write(arq,((vaz_pre_mars^[i].vazao[j]-
vaz_obs^[i].vazao[j])*100/vaz_obs^[i].vazao
[j]):10:2)
               else
                 write(arq,9999:10);

          writeln(arq);
          write(arq,nome_usi[i]:10);
          write(arq,ano_previsao:5);
          write(arq,'Mars_3d':10);
          for j:=1 to 12 do
          { caso a série seja menor que 30 o
método mars 3d não é executado}
             if (ano_fim_cal -
ano_ini_cal)>=40 then

write(arq,vaz_pre_mars_3d^[i].vazao[j]:10:2
)
               else
                 write(arq,0:10);

          writeln(arq);
          write(arq,nome_usi[i]:10);
          write(arq,ano_previsao:5);
          write(arq,'E.%Mars_3d':10);
          for j:=1 to 12 do
          { caso a série seja menor que 30 o
método mars 3d não é executado}
             if (ano_fim_cal -
ano_ini_cal)>=40 then

write(arq,((vaz_pre_mars_3d^[i].vazao[j]-
vaz_obs^[i].vazao[j])*100/vaz_obs^[i].vazao
[j]):10:2)
               else
                 write(arq,9999:10);

          writeln(arq);
          write(arq,nome_usi[i]:10);
          write(arq,ano_previsao:5);
          write(arq,'Observada':10);
             for j:=1 to 12 do

write(arq,vaz_obs^[i].vazao[j]:10:2);
             writeln(arq);
             closefile(arq);

          end;
        { procedimento que calcula o erro e a
variância }

     for j:=1 to 50 do
       begin
         dispose(vaz_obs^[j]);
         dispose(vaz_pre_otimo^[j]);
         dispose(vaz_pre_linear^[j]);
         dispose(vaz_pre_din^[j]);
         dispose(vaz_pre_mars^[j]);
         dispose(vaz_pre_mars_3d^[j]);
       end;
     dispose(vaz_obs);
     dispose(vaz_pre_otimo);
     dispose(vaz_pre_linear);
     dispose(vaz_pre_din);
     dispose(vaz_pre_mars);
     dispose(vaz_pre_mars_3d);

     ano_previsao:=ano_previsao+1;
     ano_ini_cal:=ano_ini_cal+lag_cal;
     { mantém a distância entre a matrix de
correlação e os valores previstos}
     ano_fim_cal:=ano_fim_cal+lag_pre;

     {}
   end; { fim do while ano_previsao <=
no_fim_pre }
  for i:=1 to  cont_usinas do
     begin

assignfile(arq,dir+'/afericao/lag_'+inttostr(la
g_cal_pre)+
                          '/'+(nome_usi[i])+'.pre');
         reset(arq);
         cont_ano:=1;
         new(vaz_obs);
         new(vaz_pre_otimo);
         new(vaz_pre_linear);
         new(vaz_pre_din);
         new(vaz_pre_mars);
         new(vaz_pre_mars_3d);
         new(vaz_rel_otimo);
         new(vaz_rel_linear);
         new(vaz_rel_din);
         new(vaz_rel_mars);
         new(vaz_rel_mars_3d);

          for j:=1 to 100 do
            begin
              new(vaz_obs^[j]);
              new(vaz_pre_otimo^[j]);
              new(vaz_pre_linear^[j]);
```

```
            new(vaz_pre_din^[j]);
            new(vaz_pre_mars^[j]);
            new(vaz_pre_mars_3d^[j]);
            new(vaz_rel_otimo^[j]);
            new(vaz_rel_linear^[j]);
            new(vaz_rel_din^[j]);
            new(vaz_rel_mars^[j]);
            new(vaz_rel_mars_3d^[j]);
            for k:=1 to 12 do
              begin
                vaz_obs^[j].vazao[k]:=0;
                vaz_pre_otimo^[j].vazao[k]:=0;
                vaz_pre_linear^[j].vazao[k]:=0;
                vaz_pre_din^[j].vazao[k]:=0;
                vaz_pre_mars^[j].vazao[k]:=0;

vaz_pre_mars_3d^[j].vazao[k]:=0;
                vaz_rel_otimo^[j].vazao[k]:=0;
                vaz_rel_linear^[j].vazao[k]:=0;
                vaz_rel_din^[j].vazao[k]:=0;
                vaz_rel_mars^[j].vazao[k]:=0;

vaz_rel_mars_3d^[j].vazao[k]:=0;
              end;
          end;

        while not eof (arq) do
          begin
            readln(arq,arq_text);
            if copy(arq_text,3,8) = nome_usi[i]
then
              begin
                if copy(arq_text,16,10)  ='
Observada' then
                  for j:=1 to 12 do

vaz_obs^[cont_ano].vazao[j]:=strtofloat(cop
y(arq_text,(16+10*j),10));

                if copy(arq_text,16,10)  ='
Otimo' then
                  for j:=1 to 12 do

vaz_pre_otimo^[cont_ano].vazao[j]:=strtoflo
at(copy(arq_text,(16+10*j),10));

                if copy(arq_text,16,10)  ='
Dinâmica' then
                  for j:=1 to 12 do

vaz_pre_din^[cont_ano].vazao[j]:=strtofloat(
copy(arq_text,(16+10*j),10));

                if copy(arq_text,16,10)  ='
Linear' then
                  for j:=1 to 12 do

vaz_pre_linear^[cont_ano].vazao[j]:=strtoflo
at(copy(arq_text,(16+10*j),10));;

                if copy(arq_text,16,10)  ='
Mars' then
                  for j:=1 to 12 do

vaz_pre_mars^[cont_ano].vazao[j]:=strtoflo
at(copy(arq_text,(16+10*j),10));;

                if copy(arq_text,16,10)  ='
Mars_3d' then
                  for j:=1 to 12 do

vaz_pre_mars_3d^[cont_ano].vazao[j]:=strt
ofloat(copy(arq_text,(16+10*j),10));;

                if copy(arq_text,16,10)  ='E. %
Otimo' then
                  for j:=1 to 12 do

vaz_rel_otimo^[cont_ano].vazao[j]:=strtoflo
at(copy(arq_text,(16+10*j),10));

                if copy(arq_text,16,10)  ='E. %
Dinâ.' then
                  for j:=1 to 12 do

vaz_rel_din^[cont_ano].vazao[j]:=strtofloat(
copy(arq_text,(16+10*j),10));

                if copy(arq_text,16,10)  ='E. %
Line.' then
                  for j:=1 to 12 do

vaz_rel_linear^[cont_ano].vazao[j]:=strtoflo
at(copy(arq_text,(16+10*j),10));;

                if copy(arq_text,16,10)  ='E. %
Mars' then
                  for j:=1 to 12 do

vaz_rel_mars^[cont_ano].vazao[j]:=strtofloa
t(copy(arq_text,(16+10*j),10));;

                if copy(arq_text,16,10)
='E.%Mars_3d' then
                  for j:=1 to 12 do

vaz_rel_mars_3d^[cont_ano].vazao[j]:=strto
float(copy(arq_text,(16+10*j),10));;


vaz_pre_otimo^[cont_ano].codigo:=nome_u
si[i];
              { caso a série seja menor que
30 o método mars não é executado}
                if ((ano_fim_cal -
ano_ini_cal)>=40) and

((vaz_obs^[cont_ano].vazao[1]<>0)  and
```

```
(vaz_pre_otimo^[cont_ano].vazao[1]<>0)
and

(vaz_pre_din^[cont_ano].vazao[1]<>0) and

(vaz_pre_linear^[cont_ano].vazao[1]<>0)
and

(vaz_pre_mars^[cont_ano].vazao[1]<>0)
and

(vaz_pre_mars_3d^[cont_ano].vazao[1]<>0
) and

(vaz_rel_otimo^[cont_ano].vazao[1]<>0)
and

(vaz_rel_din^[cont_ano].vazao[1]<>0) and

(vaz_rel_linear^[cont_ano].vazao[1]<>0))
and

(vaz_rel_mars^[cont_ano].vazao[1]<>0)
and

(vaz_rel_mars_3d^[cont_ano].vazao[1]<>0)
then
                        inc(cont_ano);
                if ((ano_fim_cal -
ano_ini_cal)<40) and

((vaz_obs^[cont_ano].vazao[1]<>0)  and

(vaz_pre_otimo^[cont_ano].vazao[1]<>0)
and

(vaz_pre_din^[cont_ano].vazao[1]<>0) and

(vaz_pre_linear^[cont_ano].vazao[1]<>0)
and

(vaz_rel_otimo^[cont_ano].vazao[1]<>0)
and

(vaz_rel_din^[cont_ano].vazao[1]<>0) and

(vaz_rel_linear^[cont_ano].vazao[1]<>0))
then
                        inc(cont_ano);

            end;
        end; {fim do while not eof (arq)}
      erro_variancia(vaz_pre_otimo^,
vaz_pre_din^, vaz_pre_linear^,
                  vaz_pre_mars^,
vaz_pre_mars_3d^, vaz_rel_otimo^,
                  vaz_rel_din^,
vaz_rel_linear^, vaz_rel_mars^,
                  vaz_rel_mars_3d^,
vaz_obs^,cont_ano-1,

dir+'/afericao/lag_'+inttostr(lag_cal_pre)+
                  '/'+(nome_usi[i])+'.cor');

      for j:=1 to 100 do
        begin
          dispose(vaz_obs^[j]);
          dispose(vaz_pre_otimo^[j]);
          dispose(vaz_pre_linear^[j]);
          dispose(vaz_pre_din^[j]);
          dispose(vaz_pre_mars^[j]);
          dispose(vaz_pre_mars_3d^[j]);
          dispose(vaz_rel_otimo^[j]);
          dispose(vaz_rel_linear^[j]);
          dispose(vaz_rel_din^[j]);
          dispose(vaz_rel_mars^[j]);
          dispose(vaz_rel_mars_3d^[j]);
        end;
      dispose(vaz_obs);
      dispose(vaz_pre_otimo);
      dispose(vaz_pre_linear);
      dispose(vaz_pre_mars);
      dispose(vaz_pre_mars_3d);
      dispose(vaz_pre_din);
      dispose(vaz_rel_otimo);
      dispose(vaz_rel_linear);
      dispose(vaz_rel_din);
      dispose(vaz_rel_mars);
      dispose(vaz_rel_mars_3d);

    end;
  closefile(arq);
  showmessage('Processo de aferição
terminado');
end;

end.
```