

## **AGRADECIMENTOS**

Os autores agradecem ao Senhor Engenheiro Técnico Agrário Dimas dos Reis o muito que com ele aprenderam no domínio das linguagens de programação e ao formador Senhor Jorge Vilar os conhecimentos que disponibilizou para melhorar a programação em linguagem *Visual Basic*.

## PREFÁCIO

Este livro destina-se a principiantes em técnicas de modelação matemática possuindo conhecimentos de matemática de nível universitário. Pensámos naqueles alunos, ou mesmo nos licenciados que confrontados com problemas de estabelecimento de modelos têm dificuldade em conjugar a concepção científica dos mesmos com a programação necessária para obtenção de resultados credíveis.

Olhando à mudança que as exigências comunitárias impõem aos nossos cursos universitários, exigindo do aluno um maior esforço de co-responsabilização na sua própria aprendizagem, pensámos que um livro desta natureza pode motivar aqueles que por si sós querem desbravar esta parte da optimização numérica aplicada.

O Visual Basic (versão 6) foi a linguagem de programação escolhida dada a fácil compatibilização com as técnicas de optimização numérica exigidas (Frenz, 2002).

A experiência dos autores no ensino da Hidrologia fez com que a maioria dos exemplos referidos sejam na área desta disciplina. Porém as informações fornecidas nesta obra, bem como a bibliografia citada, são suficientes para permitir ao leitor interessado desenvolver modelos semelhantes embora noutros domínios científicos.

## ÍNDICE DO TEXTO

Capítulo	Página
<b>1. INTRODUÇÃO</b> .....	1
<b>2. ESTIMAÇÃO PELO MÉTODO DOS MÍNIMOS QUADRADOS LINEARES</b> .....	5
<b>3. ESTIMAÇÃO PELO MÉTODO DOS MÍNIMOS QUADRADOS NÃO LINEARES</b> .....	11
<b>4. MÉTODOS DIRECTOS DE OPTIMIZAÇÃO</b> .....	19
4.1. Método de Hooke e Jeeves(Modificado por Monro) .....	20
4.2. Método de Nelder e Mead .....	25
4.3. Método de Rosenbrock.....	33
<b>5.EXEMPLOS DE APLICAÇÃO</b> .....	41
5.1. Determinação do Mínimo de uma Função .....	41
5.1.1. Aplicação à Função de Rosenbrock .....	41
Método de Monro.....	43
Método de Nelder e Mead .....	47

Método de Rosenbrock .....	50
5.1.2. Aplicação à Resolução de Sistemas de Equações .....	60
5.2. Ajustamento de uma Função Não Linear a Dados Experimentais .....	62
5.2.1. Caso de uma Variável Independente .....	63
Método dos Mínimos Quadrados Não Lineares .....	63
Método de Monro .....	69
Método de Nelder e Mead .....	76
Método de Rosenbrock .....	80
5.2.2. Caso de duas ou mais Variáveis Independentes .....	83
Método dos Mínimos Quadrados Não Lineares .....	84
Método de Monro .....	89
Método de Nelder e Mead .....	95
Método de Rosenbrock .....	98
5.2.3. Exemplos de Índole Hidrológica .....	101
Determinação da Curva Característica Humidade - Sucção .....	102
Determinação de um Modelo Para o Caudal de Ponta de Cheia .	106
Ajustamento de um Modelo de Semi-Variograma Esférico à Precipitação .....	110
Curva de Variação do Índice de Erosividade em Função do Tempo .....	113
5.3. Optimização dos Parâmetros de um Modelo de Escoamento .....	117
5.3.1. Modelo Mensal .....	118
5.3.2. Modelo Diário .....	126
<b>BIBLIOGRAFIA</b> .....	141
<b>LISTAGEM DOS PROGRAMAS</b> .....	145

# 1. INTRODUÇÃO

Entende-se por otimização o processo de obtenção dos melhores resultados sob certas condições. Pode consistir em minimizar um custo ou maximizar um lucro, por exemplo. Este custo ou lucro são expressos como funções de um determinado número de variáveis, pelo que a otimização se pode definir como o processo que calcula o máximo ou o mínimo valor de uma função.

Na Figura 1.1 pode ver-se que se  $X^*$  corresponde ao valor mínimo de  $f(x)$ , ele corresponderá também ao máximo valor do simétrico da função, isto é,  $-f(x)$ .

Assim, sem perda de generalidade, podemos desenvolver técnicas para minimização (ou maximização) pois é fácil fazer a respectiva conversão para a maximização (ou minimização).

A otimização pode tipificar-se deste modo:

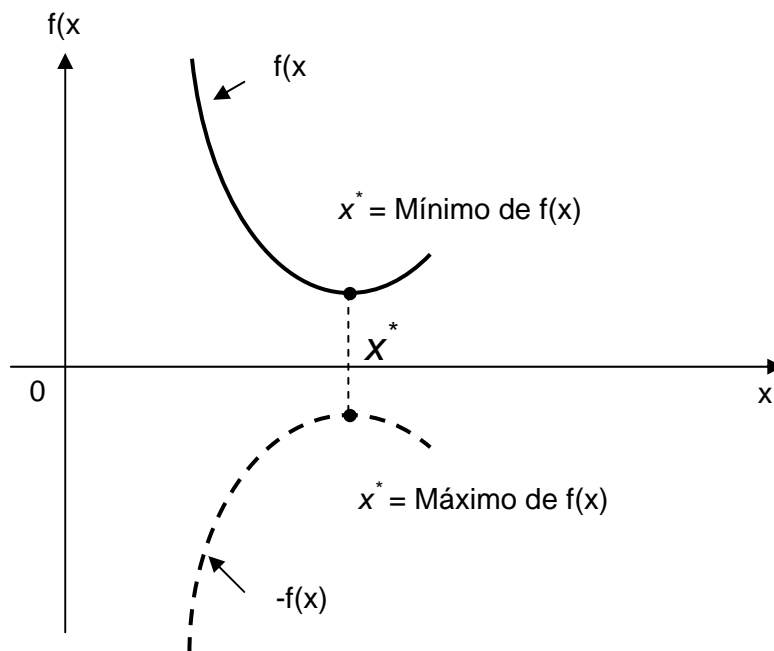
Determinar  $\mathbf{X} = [x_1 \ x_2 \dots \ x_n]^T$  que optimize  $f(x)$ , sujeita às restrições,

$$g_j(\mathbf{X}) \leq 0, \quad j = 1, 2, \dots, m$$

$$l_j(\mathbf{X}) = 0, \quad j = m + 1, m + 2, \dots, p,$$

sendo  $x$  um vector n-dimensional,  $f(x)$  a função objectivo,  $g_j(\mathbf{X})$  e  $l_j(\mathbf{X})$  restrições de desigualdade e/ou igualdade, respectivamente.

Se  $p = 0$  trata-se de um problema sem restrições.



**Figura 1.1** – Mínimo de  $f(x)$  ou máximo de  $-f(x)$ .

A função  $f(x)$  é também denominada função critério e a sua escolha está dependente do problema. Em alguns casos pode até haver mais de uma função critério. Assim se  $f_1(x)$  e  $f_2(x)$  são duas funções objectivo, pode construir-se uma nova função objectivo para ser optimizada (Rao, 1979),

$$f(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x)$$

onde  $\alpha_1$  e  $\alpha_2$  traduzem a importância relativa de cada uma delas.

O lugar geométrico de todos os pontos satisfazendo a  $f(x) = c$  (constante), forma uma hipersuperfície no espaço considerado, correspondendo a cada valor de  $c$  um elemento dessa família.

Para o caso bi-dimensional podemos esquematizar assim o problema com três restrições:

Minimizar  $f(x) = x_1^2 - 4x_1 + x_2^2 + 4$  sujeito a,

$$-x_1^2 + x_2 - 1 \geq 0$$

$$x_1 - x_2 + 2 \geq 0$$

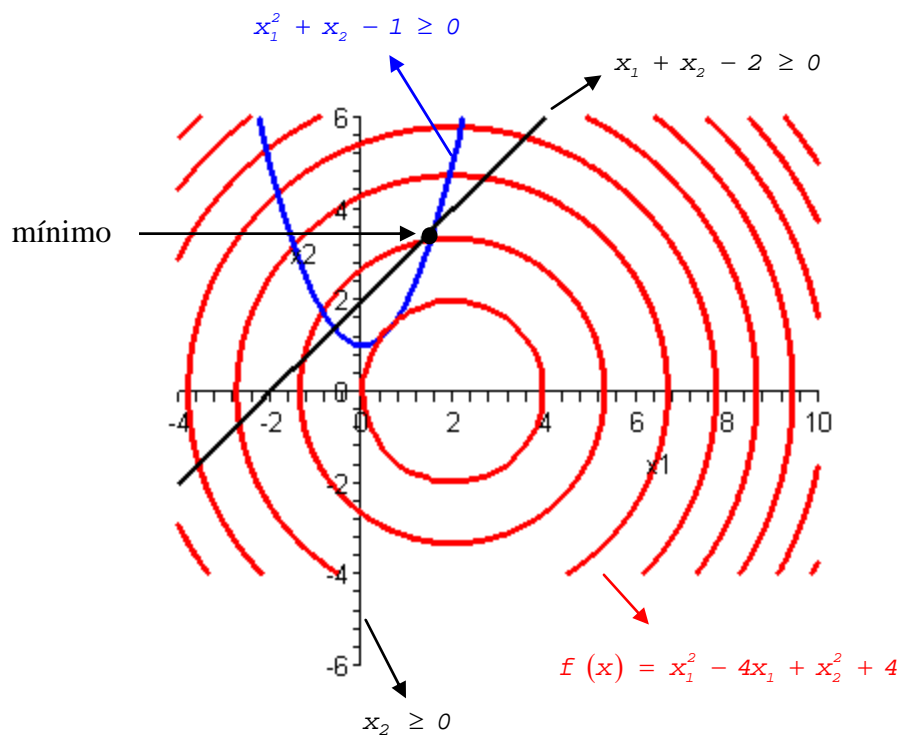
$$x_2 \geq 0,$$

que representamos graficamente na Figura 1.2.

Os métodos directos da análise diferencial para cálculo de máximos e mínimos de uma função  $f(x)$  exigem que ela seja de classe  $C^1$ . Quando houver restrições de igualdade, como no caso da grande maioria dos modelos de escoamento, o método dos multiplicadores de

Lagrange poderia ser usado, não fora a dificuldade da resolução do sistema de equações não lineares a que ele como regra conduz. Assim se compreende o enorme interesse dos métodos directos de optimização numérica que serão à frente abordados.

Fundamentação estritamente teórica para esta matéria pode encontra-se em Bonnans *et al* (2003) e Smirnov e Bushenkov (2005).



**Figura 1.2** – Mínimo de uma função objectivo com três restrições.



## 2. ESTIMAÇÃO PELO MÉTODO DOS MÍNIMOS QUADRADOS LINEARES

A calibração de modelos consiste na comparação dos valores observados com as saídas (output) geradas pelo modelo, para o que é necessário proceder à optimização dos seus parâmetros. Esta é feita tendo em vista o melhor ajustamento entre os valores observados e os valores calculados, o que se faz a partir de uma função objectivo, minimizada quando se tratar da diferença entre valores calculados e valores observados.

A estrutura do modelo deve ser, quer do ponto de vista matemático, quer do ponto de vista computacional, compatível com a técnica escolhida para optimizar determinada função objectivo. A técnica mais usual é a dos mínimos quadrados lineares, em que se minimiza a soma do quadrado dos desvios entre valores observados e valores calculados pelo modelo. Calculando a primeira derivada da função objectivo relativamente a cada parâmetro e igualando a zero as equações resultantes, obtemos o chamado sistema de equações normais, de fácil resolução, visto tratar-se de equações lineares.

Este método exige determinados pressupostos:

- Não haver erro na medição das variáveis independentes;
- As variáveis independentes serem estatisticamente não-correlacionadas;

- Os erros na variável dependente serem normalmente distribuídos, com média nula e variância constante.

Convém acentuar que quando as equações do modelo são lineares, a soma dos quadrados dos desvios é uma função quadrática e as suas derivadas são ainda lineares. Isto permite que as estimativas sejam obtidas por resolução de um sistema de equações lineares.

Quando dizemos que um modelo é linear estamos a referirmo-nos à linearidade nos parâmetros. O valor do expoente mais elevado da variável independente no modelo é denominado ordem do modelo. Por exemplo,

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon,$$

é um modelo de 2ª ordem (em  $x$ ) linear (nos parâmetros  $\beta$ ), sendo  $\varepsilon$  uma variável aleatória. De uma forma geral, para mais de uma variável independente, podemos escrever,

$$y = \beta_0 z_0 + \beta_1 z_1 + \beta_2 z_2 + \dots + \beta_p z_p + \varepsilon. \quad (2.1)$$

Se  $p = 1$ , fazendo  $z_1 = x$  na equação (2.1) temos o modelo linear de 1ª ordem com uma variável independente,

$$y = \beta_0 + \beta_1 x + \varepsilon.$$

Se  $p = k$ , com  $z_j = x_j$ , temos o modelo linear de 1ª ordem com  $k$  variáveis independentes,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon.$$

Se  $p = 2$ ,  $z_1 = x$ ,  $z_2 = x^2$  e  $\beta_2 = \beta_{11}$ , obtemos um modelo linear de 2ª ordem com uma variável independente,

$$y = \beta_0 + \beta_1 x + \beta_{11} x^2 + \varepsilon.$$

Se  $p = 5$ ,  $z_1 = x_1$ ,  $z_2 = x_2$ ,  $z_3 = x_1^2$ ,  $z_4 = x_2^2$ ,  $z_5 = x_1 x_2$ ,  $\beta_3 = \beta_{11}$ ,  $\beta_4 = \beta_{22}$  e  $\beta_5 = \beta_{12}$ , temos o modelo linear de 2ª ordem com duas variáveis independentes,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon$$

Se  $p = 3$ ,  $z_1 = x$ ,  $z_2 = x^2$ ,  $z_3 = x^3$ ,  $\beta_2 = \beta_{11}$  e  $\beta_3 = \beta_{111}$ , temos,

$$y = \beta_0 + \beta_1 x + \beta_{11} x^2 + \beta_{111} x^3 + \varepsilon,$$

que é um modelo linear de 3ª ordem com uma variável independente.

Vemos assim que é fácil obtermos modelos lineares de qualquer ordem a partir da equação (2.1).

Outras transformações poderiam ser mencionadas. O objectivo visado ao efectuar tais transformações é apenas a obtenção de um modelo de regressão mais simples.

Os modelos não lineares (nos parâmetros) podem classificar-se em *intrinsecamente lineares* e *intrinsecamente não-lineares*. Se um modelo é intrinsecamente linear, pode ser expresso, mediante transformações adequadas, na forma da equação (2.1). Caso contrário diz-se intrinsecamente não-linear. Por exemplo, o modelo multiplicativo,

$$y = \alpha x_1^\beta x_2^\gamma x_3^\delta \varepsilon,$$

onde  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$  são parâmetros desconhecidos e  $\varepsilon$  o erro multiplicativo aleatório, pode transformar-se na forma linear, tomando logaritmos neperianos em ambos os membros,

$$\ln y = \ln \alpha + \beta \ln x_1 + \gamma \ln x_2 + \delta \ln x_3 + \ln \varepsilon,$$

permitindo portanto a estimação dos parâmetros pelos métodos matriciais da regressão linear para mínimos quadrados, depois da verificação dos pressupostos exigidos. Os resultados obtidos com esta linearização não são necessariamente idênticos aos obtidos por solução do problema não-linear, embora a diferença seja aceitável do ponto de vista prático.

Já o modelo,

$$y = \alpha x_1^\beta x_2^\gamma x_3^\delta + \varepsilon,$$

uma vez que é intrinsecamente não-linear, não permite tratamento semelhante ao anterior. O método dos mínimos quadrados necessita de processos iterativos de estimação de  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\delta$ , como trataremos em seguida.



---

### 3. ESTIMAÇÃO PELO MÉTODO DOS MÍNIMOS QUADRADOS NÃO LINEARES

Vimos que um modelo linear nos parâmetros é do tipo,

$$y = \beta_0 + \beta_1 z_1 + \beta_2 z_2 + \dots + \beta_p z_p + \varepsilon, \quad (3.1)$$

(fazendo  $z_0 = 1$  na Equação 2.1).

Quando o modelo não tem a forma da equação (3.1) é chamado não linear nos parâmetros. Contudo, vamos exemplificar dois destes modelos com comportamento distinto. Um deles do tipo,

$$y = e^{\beta_1 + \beta_2 x^2 + \varepsilon},$$

e outro da forma,

$$y = \frac{\beta_1}{\beta_1 - \beta_2} \left[ e^{-\beta_2 x} - e^{-\beta_1 x} \right] + \varepsilon.$$

Enquanto que o primeiro, tomando logaritmos neperianos em ambos os membros, pode ser transformado num modelo linear,

$$\ln y = \beta_1 + \beta_2 x^2 + \varepsilon,$$

visto ser intrinsecamente linear, o segundo modelo é de impossível conversão na forma linear, dizendo-se assim intrinsecamente não-linear.

Para abordar a técnica dos mínimos quadrados nos casos não-lineares, façamos,

$$y = f(\xi_1, \xi_2, \dots, \xi_k; \theta_1, \theta_2, \dots, \theta_p) + \varepsilon. \quad (3.2)$$

com  $\boldsymbol{\xi} = [\xi_1, \xi_2, \dots, \xi_k]^T$ ,  $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_p]^T$ . Então podemos escrever a Equação 3.2 na forma abreviada,

$$y = f(\boldsymbol{\xi}, \boldsymbol{\theta}) + \varepsilon,$$

ou

$$E(y) = f(\boldsymbol{\xi}, \boldsymbol{\theta}),$$

assumindo que  $E(\varepsilon) = 0$ .

Admitiremos também que os erros estão não-correlacionados e que  $\sigma^2(\varepsilon) = \sigma^2$  com  $\varepsilon \sim N(0, \sigma^2)$ , o que acarreta a independência dos erros.

Quando existem  $n$  observações da forma  $y_u, \xi_{1u}, \xi_{2u}, \dots, \xi_{ku}$  para  $u = 1, 2, \dots, n$ , podemos escrever o modelo na forma alternativa,



$$y_u = f(\xi_{1u}, \xi_{2u}, \dots, \xi_{ku}; \theta_1, \theta_2, \dots, \theta_p) + \varepsilon_u,$$

onde  $\varepsilon_u$  é o erro da  $u$ -ésima ordem, com  $u = 1, 2, \dots, n$ .

Portanto,

$$y_u = f(\boldsymbol{\xi}_u, \boldsymbol{\theta}) + \varepsilon_u,$$

com  $\boldsymbol{\xi}_u = [\xi_{u1}, \xi_{u2}, \dots, \xi_{uk}]^T$ .

O pressuposto quanto à normalidade e independência dos erros pode agora escrever-se  $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, I\sigma^2)$ , onde  $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]^T$ ,  $\mathbf{0}$  é o vector nulo e  $I$  é a matriz identidade. Neste caso define-se a soma do quadrado dos erros por,

$$S(\boldsymbol{\theta}) = \sum_{u=1}^n [y_u - f(\boldsymbol{\xi}_u, \boldsymbol{\theta})]^2 \quad (3.3)$$

sendo portanto uma função de  $\boldsymbol{\theta}$ , já que  $y_u$  e  $\boldsymbol{\xi}_u$  são observações fixas.

A estimativa de mínimos quadrados para  $\boldsymbol{\theta}$  será indicada por  $\hat{\boldsymbol{\theta}}$  e é o valor que minimiza  $S(\boldsymbol{\theta})$ . Como Draper e Smith (1966) demonstraram, admitindo que  $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, I\sigma^2)$ , a estimativa de mínimos quadrados para  $\boldsymbol{\theta}$  é também uma estimativa de máxima verosimilhança.

Para determinar a estimativa de mínimos quadrados de  $\boldsymbol{\theta}$ , derivamos a Equação 3.3 em ordem a  $\hat{\boldsymbol{\theta}}$ . Isto conduz a  $p$  equações

normais que teremos de resolver em ordem a  $\hat{\theta}$ . Estas equações são da forma,

$$\sum_{u=1}^n [y_u - f(\xi_u, \hat{\theta})] \cdot \left[ \frac{\partial f(\xi_u, \theta)}{\partial \theta_i} \right]_{\theta=\hat{\theta}} = 0, \quad (3.4)$$

Para  $i = 1, 2, \dots, p$ .

Chamamos a atenção para o facto de que quando a função  $f(\xi_u, \theta)$  era linear a derivada em ordem aos parâmetros era apenas função de  $\xi_u$ . Quando o modelo é não-linear nos parâmetros também o mesmo sucede com as equações normais. Daí a complexidade do problema, como verificaremos com um exemplo.

Suponhamos  $f(\theta, t) = e^{-\theta t}$ , com  $n$  pares de observações disponíveis  $(y_1, t_1)(y_2, t_2) \dots (y_n, t_n)$ .

$$\frac{\partial f}{\partial \theta} = -te^{-\theta t}.$$

Aplicando a Equação 3.4 obteremos a única equação normal,

$$\sum_{u=1}^n [y_u - e^{-\hat{\theta}t_u}] \cdot [-t_u e^{-\hat{\theta}t_u}] = 0,$$

ou desenvolvendo,

$$\sum_{u=1}^n y_u t_u e^{-\hat{\theta} t_u} - \sum_{u=1}^n t_u e^{-2\hat{\theta} t_u} = 0.$$

Vemos assim que, mesmo com um só parâmetro e com um modelo não-linear relativamente simples, a determinação de  $\hat{\theta}$  por resolução de apenas uma equação normal não é fácil. Com mais parâmetros e modelos mais complicados, a solução do sistema de equações normais é extremamente difícil, pelo que se recorre, regra geral, a métodos iterativos, dos quais, será aqui abordado o método da linearização.

No processo de linearização (por truncatura da série de Taylor), utilizam-se os resultados dos mínimos quadrados lineares numa sucessão de etapas iterativas.

Suponhamos um modelo do tipo,

$$y_u = f(\boldsymbol{\xi}_u, \boldsymbol{\theta}) + \varepsilon_u, \quad (3.5)$$

e sejam  $\theta_{10}, \theta_{20}, \dots, \theta_{p0}$  os valores iniciais para os parâmetros. Estes valores iniciais podem ser melhorados por sucessivas iterações. Por desenvolvimento de  $f(\boldsymbol{\xi}_u, \boldsymbol{\theta})$  em série de Taylor, em torno de  $\theta_0$ , com  $\boldsymbol{\theta} = [\theta_{10}, \theta_{20}, \dots, \theta_{p0}]^T$  e limitando-nos às primeiras derivadas, por truncatura da série, podemos tomar para  $\theta$ , próximo de  $\theta_0$ ,

$$f(\boldsymbol{\xi}_u, \boldsymbol{\theta}) \cong f(\boldsymbol{\xi}_u, \boldsymbol{\theta}_0) + \sum_{i=1}^p \left[ \frac{\partial f(\boldsymbol{\xi}_u, \boldsymbol{\theta})}{\partial \theta_i} \right]_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} (\theta_i - \theta_{i0}).$$

Se fizermos,  $f_u^0 = f(\boldsymbol{\xi}_u, \boldsymbol{\theta}_0)$ ,  $\beta_i^0 = \theta_i - \theta_0$  e  $z_{iu}^0 = \left[ \frac{\partial f(\boldsymbol{\xi}_u, \boldsymbol{\theta})}{\partial \theta_i} \right]_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}$ ,

podemos dizer que o modelo (3.5) é aproximadamente da forma,

$$y_u - f_u^0 = \sum_{i=1}^p \beta_i^0 \cdot z_{iu}^0 + \varepsilon_u,$$

ou seja, da forma (3.1). Assim já podemos estimar os parâmetros  $\beta_i^0$ ,  $i = 1, 2, \dots, p$ , por aplicação do método dos mínimos quadrados lineares.

Fazendo,

$$\mathbf{Z}_0 = \begin{bmatrix} z_{11}^0 & z_{21}^0 & \dots & z_{p1}^0 \\ z_{12}^0 & z_{22}^0 & \dots & z_{p2}^0 \\ \vdots & \vdots & \vdots & \vdots \\ z_{1u}^0 & z_{2u}^0 & \dots & z_{pu}^0 \\ z_{1n}^0 & z_{2n}^0 & \dots & z_{pn}^0 \end{bmatrix} = \begin{bmatrix} z_{iu}^0 \\ n \times p \end{bmatrix},$$

$$\mathbf{b}_0 = \begin{bmatrix} b_1^0 \\ b_2^0 \\ \vdots \\ b_p^0 \end{bmatrix}$$

e

$$\mathbf{y}_0 = \begin{bmatrix} y_1 - f_1^0 \\ y_2 - f_2^0 \\ \vdots \\ y_u - f_u^0 \\ y_n - f_n^0 \end{bmatrix} = \mathbf{y} - \mathbf{f}^0,$$

temos as estimativas de  $\boldsymbol{\beta} = [\beta_1^0 \ \beta_2^0 \ \dots \ \beta_p^0]^T$  dadas por,

$$\mathbf{b}_0 = (\mathbf{z}_0^T \mathbf{z}_0)^{-1} \cdot \mathbf{z}_0^T \cdot (\mathbf{y} - \mathbf{f}^0).$$

Assim, o vector  $\mathbf{b}_0$  minimizará a soma do quadrado dos desvios,

$$SS(\boldsymbol{\theta}) = \sum_{u=1}^n \left[ y_u - f(\boldsymbol{\xi}_u - \boldsymbol{\theta}_0) - \sum_{i=1}^p \beta_i^0 z_{iu}^0 \right]^2, \quad (3.6)$$

relativamente a  $\beta_i^0$ ,  $i=1,2,\dots,p$  e onde  $\beta_i^0 = \theta_i - \theta_{i0}$ . Escrevendo  $\mathbf{b}_i^0 = \theta_{i1} - \theta_{i0}$ , podemos tomar  $\theta_{i1}$ ,  $i=1,2,\dots,p$ , como melhores estimativas revistas de  $\theta$ .

É bom chamar a atenção para a diferença entre a soma dos quadrados  $S(\boldsymbol{\theta})$  da equação 3.3, onde o modelo não-linear era usado, e a soma dos quadrados da equação (3.6) onde se utiliza uma aproximação linear do modelo.

Podemos agora tomar os valores  $\theta_{i1}$  como estimativas revistas e, de acordo com as equações anteriores e substituindo  $\theta_{i0}$  por estas

ltimas estimativas, tendo o cuidado de, nas equaces referidas, substituir os ndices zero por ndices um. Obtm-se assim novas estimativas revistas  $\theta_{i2}$  e os ciclos repetem-se.

Tem-se pois,

$$\boldsymbol{\theta}_{j+1} = \boldsymbol{\theta}_j + \mathbf{b}_j,$$

$$\boldsymbol{\theta}_{j+1} = \boldsymbol{\theta}_j + (\mathbf{Z}_j^T \cdot \mathbf{Z}_j)^{-1} \cdot \mathbf{Z}_j^T \cdot (\mathbf{y} - \mathbf{f}^j),$$

onde,  $\mathbf{Z}_j = [z_{iu}^j]$ ,  $\mathbf{f}^j = [f_1^j \ f_2^j \ \dots \ f_n^j]^T$  e  $\boldsymbol{\theta}_j = [\theta_{1j} \ \theta_{2j} \ \dots \ \theta_{pj}]^T$ .

O processo iterativo  continuado at se obter convergncia da soluo, isto , at que em sucessivas iteraes  $j$  e  $j+1$  se tenha,

$$\frac{|\boldsymbol{\theta}_{i(j+1)} - \boldsymbol{\theta}_{ij}|}{\boldsymbol{\theta}_{ij}} < \delta, \quad i = 1, 2, \dots, p$$

com  $\delta$  previamente fixado (por exemplo,  $10^{-6}$ ).

Por vezes ocorrem dificuldades na obteno da convergncia.

O esquema que seguiremos para estimaco dos parmetros pelo mtodo dos mnimos quadrados no-lineares  da autoria de Boratto (1984).

## 4. MÉTODOS DIRECTOS DE OPTIMIZAÇÃO

Os métodos que permitem resolver problemas de optimização de uma função  $F$  de  $n$  variáveis, sem restrições, são:

**a) Métodos que não implicam o cálculo de derivadas de  $F$ .**

- a1. Método de pesquisa em pontos pré-fixados.
- a2. Método de pesquisa segundo direcções pré-fixadas.
- a3. Método de **Hooke e Jeeves**.
- a4. Método de **Nelder e Mead**.
- a5. Método de **Rosenbrock**.
- a6. Método de Davis, Swann e Campey.
- a7. Método de Powel.

**b) Métodos que implicam o cálculo de valores de  $F$  e de primeiras derivadas de  $F$ .**

- b1. Método do declive máximo.
- b2. Método das direcções conjugadas.
- b3. Métodos baseados em direcções aproximadamente conjugadas.

**c) Métodos que implicam o cálculo de valores de  $F$ , de primeiras e segundas derivadas de  $F$ .**

- c1. Método do Newton-Raphson.

Apenas vamos considerar aqui os métodos que não implicam o cálculo das derivadas de  $F(x)$ . Estudaremos três métodos:

1. Método de Hooke e Jeeves.
2. Método de Nelder e Mead
3. Método de Rosenbrock

O esquema comum a qualquer destes métodos iterativos pode ver-se na Figura 4.1.

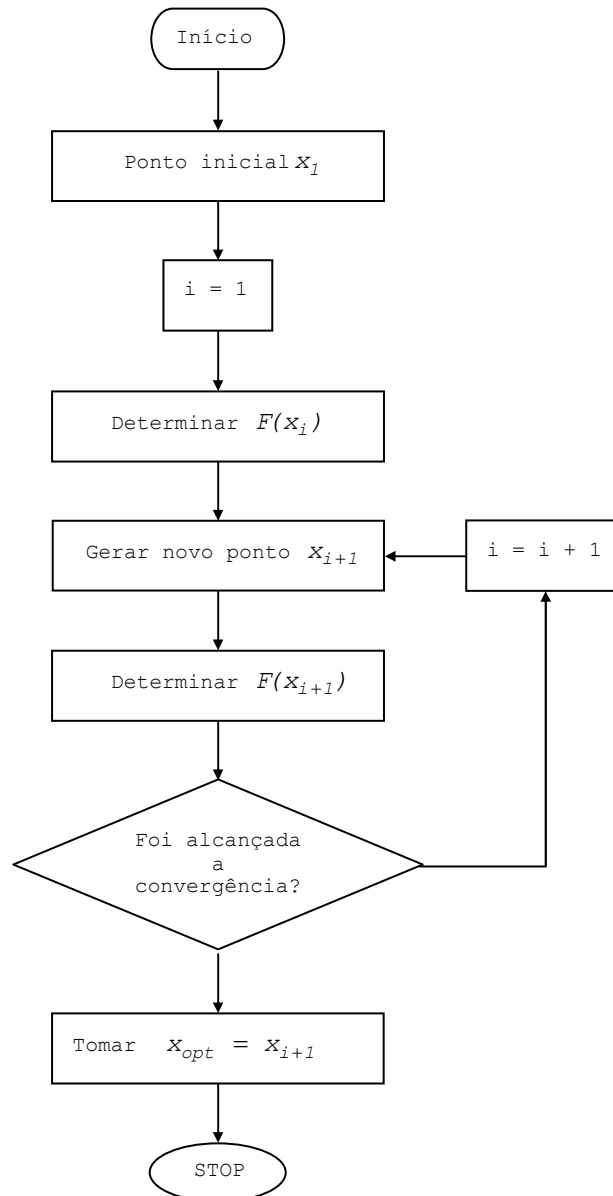
Os diferentes métodos que vamos passar a descrever diferem fundamentalmente no processo de geração do novo ponto  $x_{i+1}$ .

#### **4.1. Método de Hooke e Jeeves (Modificado por Monro)**

Seguiremos o trabalho de Monro (1971) e usaremos a terminologia de acordo com Hipólito (1985).

O processo inicia-se com um conjunto de valores para os parâmetros,  $A(l)$ , com  $l = 1, 2, \dots, N$ , ajustando-os sucessivamente e testando os resultados. Há dois tipos de ajustamentos denominados excursão local (LE) e direcção preferencial (PM). Enquanto que na excursão local o incremento  $\Delta$  é uma quantidade fixada ou uma percentagem do valor do coeficiente e só há modificação do valor do parâmetro se houver melhoria na função objectivo, no movimento ao longo da direcção preferencial a grandeza do ajustamento aplicado a cada coeficiente é geralmente muito superior à da excursão local e não se usa a função critério para aceitar ou rejeitar este movimento.





**Figura 4.1** – Esquema geral da optimização iterativa.

Na Figura 4.2 apresenta-se um exemplo extraído de Hipólito (1985). O valor inicial dos parâmetros  $A^0(1)$  e  $A^0(2)$  é representado pelo ponto  $A^0$ .

Verifica-se se o valor da função objectivo à direita de  $A^0$ , de coordenadas  $(A^0(1) + \Delta(1), A^0(2))$  é menor ou não do que em  $A^0$ . Se for,  $B^1(1) = A^0(1) + \Delta(1)$  passará a designar o valor do parâmetro 1. Caso contrário, testa-se o ponto à esquerda de  $A^0$ , passando a designar-se o novo valor do parâmetro 1 por  $B^1(1) = A^0(1) - \Delta(1)$ , se o teste for positivo, isto é, se à esquerda de  $A^0$  o valor da função objectivo for menor do que em  $A^0$ . Se o teste não for positivo o novo valor será  $B^1(1) = A^0(1)$ .

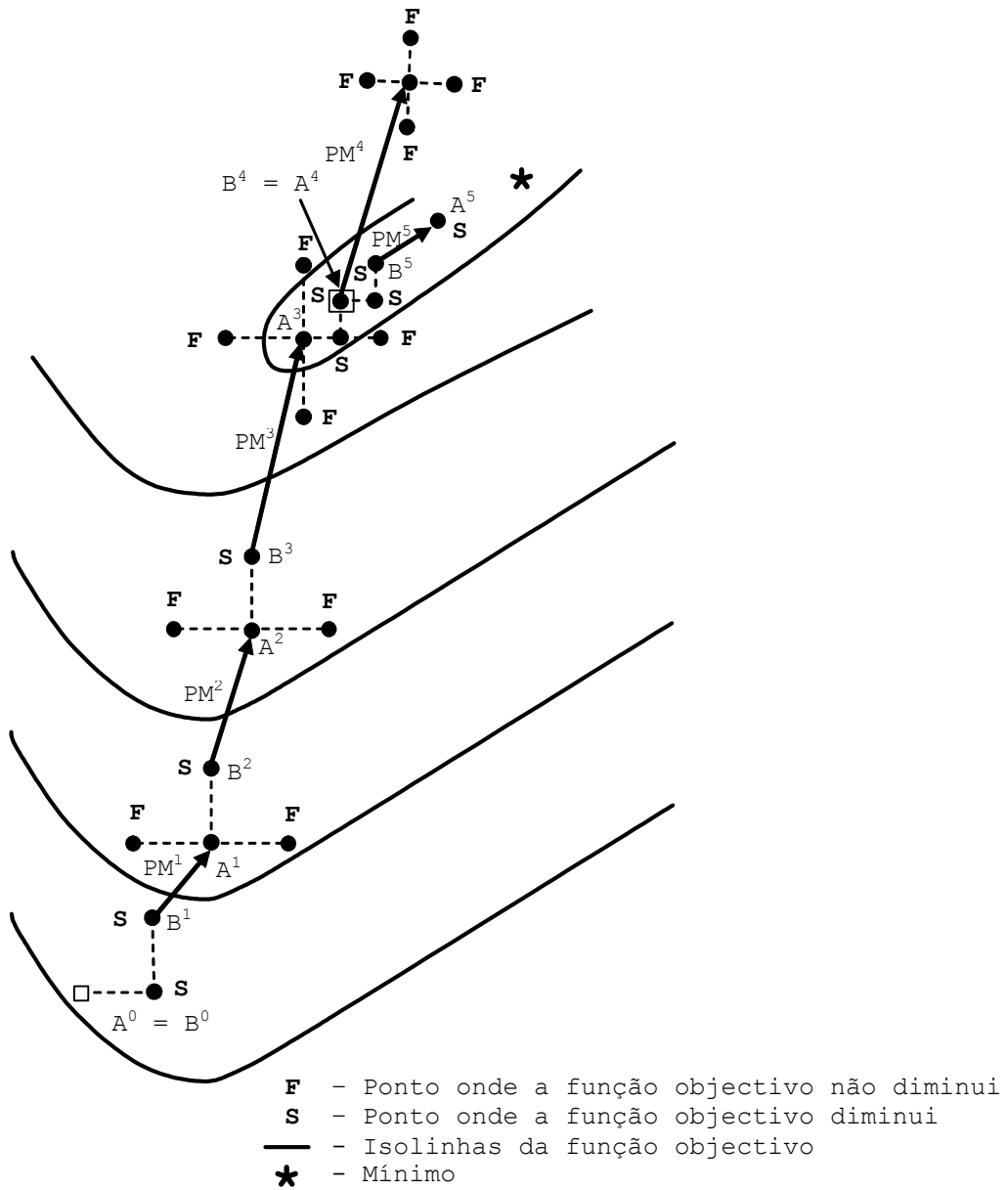
Repete-se o processo para o outro parâmetro, obtendo-se  $B^1(2)$ .

Determina-se assim a direcção preferencial representada pela recta que une  $B^0$  a  $B^1$ , ao longo da qual se fará um movimento,  $PM^1$ , de grandeza igual à distância entre os pontos  $B^1$  e  $B^0$ , definindo-se o ponto  $A^1$ .

$$A^1 = 2B^1 - B^0$$

A definição do ponto  $B^2$  é feita a partir de uma excursão local,  $LE^2$ , em torno de  $A^1$ .

A definição de  $A^2$  é feita, de novo, segundo a direcção que une os resultados das duas últimas excursões locais,  $B^1$  e  $B^2$ , num movimento  $PM^2$ , igual à distância entre os referidos resultados,  $B^1$  e  $B^2$ ,



**Figura 4.2** – Exemplificação das trajectórias de Hooke e Jeeves (Monro).

$$A^2 = 2B^2 - B^1$$

Diz-se que uma excursão local não teve sucesso quando a variação produzida nos valores dos parâmetros não permite encontrar para a função objectivo um valor inferior ao conseguido na excursão local anterior.

De um modo geral, pode dizer-se que após a  $i$ -ésima excursão local com sucesso e  $B^i \neq A^{i-1}$  a posição do ponto  $A^i$ , resultante do movimento segundo a direcção preferencial,  $PM^i$ , é definida por,

$$A^i = 2B^i - B^{i-1}$$

Quando o resultado da excursão local é  $B^i = A^i$  (Na Figura 4.2, após  $PM^3$ ), repete-se a excursão local para valores de  $\Delta$  reduzidos a metade,  $\Delta^{k+1} = \frac{\Delta^k}{2}$ , conservando-se estes valores até nova situação deste tipo.

Quando a excursão local não tem sucesso (na figura a excursão a seguir a  $PM^4$ ), realiza-se uma nova excursão em torno do ponto  $B$  da excursão local anterior, que passa a comportar-se como um ponto inicial.

Inicia-se a excursão local para cada parâmetro  $A^i(I)$  somando-lhe ou subtraindo-lhe  $\Delta^k$ , consoante se tenha obtido sucesso ou não na excursão anterior. Se nessa operação não se obtiver sucesso, pesquisa-se o valor da função objectivo no sentido contrário. Em  $A^0$  a excursão inicia-se somando  $\Delta^0$ .

O processo termina quer por especificação do número máximo de divisões de  $\Delta^0$ , máximo valor de  $k$  ( $k_c$ ), quer por especificação do número máximo de determinações do valor da função objectivo, sendo determinante o critério que primeiro ocorrer.

Na Figura 4.3 apresenta-se o fluxograma das operações descritas para este método e que servirá de base à elaboração do programa utilizado posteriormente.

## 4.2. Método de Nelder e Mead

O método de Nelder e Mead (Nelder e Mead, 1964) minimiza a função de  $n$  variáveis independentes utilizando  $n+1$  vértices dum simplex (poliedros regulares em  $\mathbb{R}^n$ ).

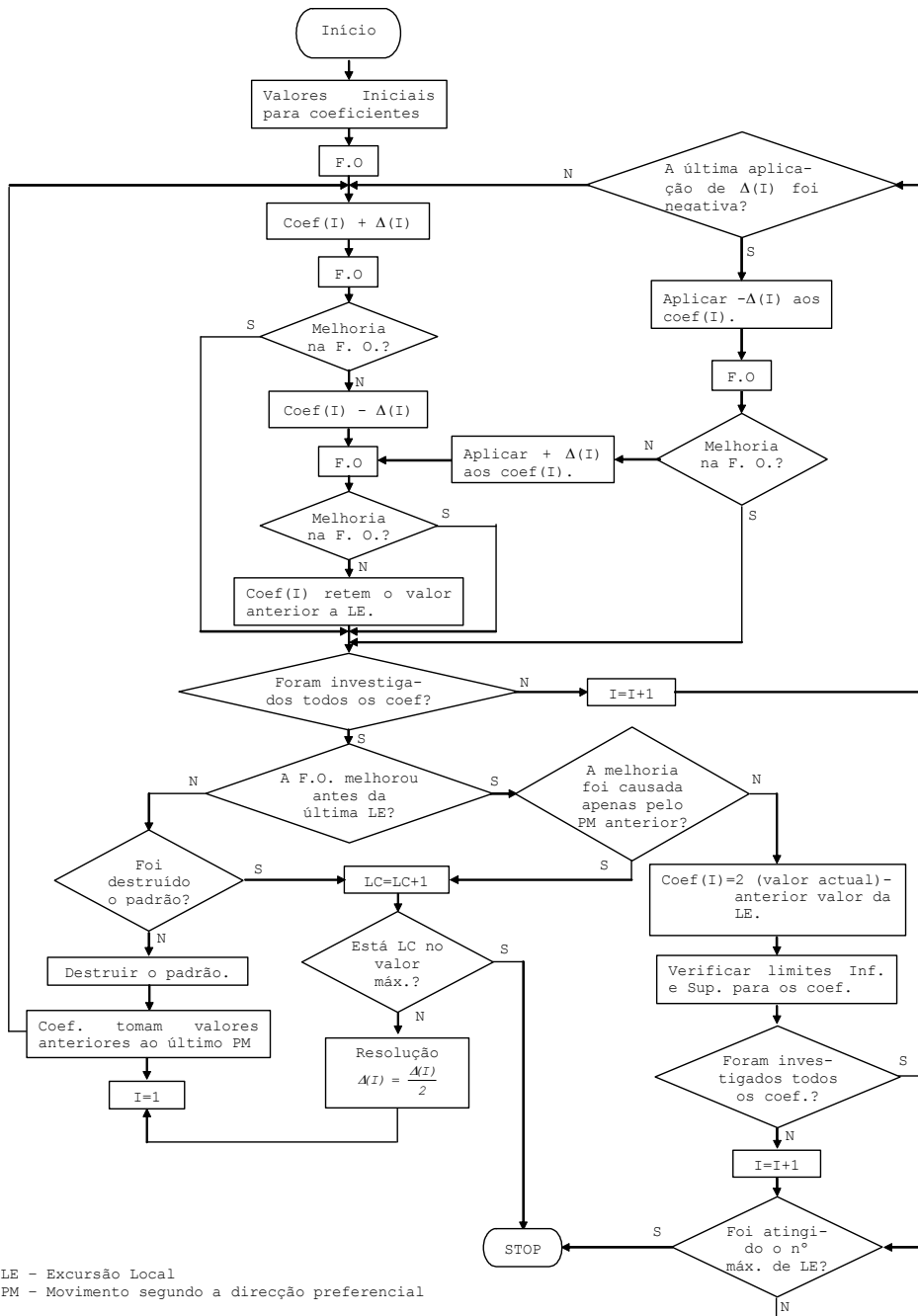
Cada vértice é definido por um vector  $X$ . O vértice onde a função objectivo toma o valor mais elevado é projectado através do centro de gravidade (centróide) dos restantes vértices.

Substituindo o ponto onde o valor de  $f(\mathbf{x})$  é mais elevado por melhores pontos, pode encontrar-se o mínimo de  $f(\mathbf{x})$ .

ado por Monro (1971).

Segue-se o algoritmo:

Seja  $\mathbf{x}_i^{(k)} = [x_{i1}^{(k)}, \dots, x_{ij}^{(k)}, \dots, x_{in}^{(k)}]^T$ ,  $i = 1, \dots, n+1$ , o vértice de ordem  $i$  em  $\mathbb{R}^n$  na etapa  $k$  da pesquisa,  $k = 0, 1, \dots$  e sejam  $f(x_i^{(k)})$  os valores da função objectivo em  $x_i^{(k)}$ .



**Figura 4.3** – Método de pesquisa directa de Hooke e Jeeves (1961) modificado por Monro (1971).

Procuremos os valores de  $X$  do poliedro onde  $f(\mathbf{x})$  atinge o máximo valor e o mínimo valor.

Sejam:

$$f(x_h^{(k)}) = \max [f(x_1^{(k)}), \dots, f(x_{n+1}^{(k)})]$$

com o correspondente  $x_i^{(k)} = x_h^{(k)}$  e

$$f(x_1^{(k)}) = \min [f(x_1^{(k)}), \dots, f(x_{n+1}^{(k)})]$$

com o correspondente  $x_i^{(k)} = x_1^{(k)}$ .

Como o poliedro em  $\mathbb{R}^n$  contem  $n+1$  vértices,  $x_1, \dots, x_{n+1}$ , seja  $x_{n+2}$  o centróide de todos os vértices excluindo  $x_h$ . As coordenadas do centróide são dadas por,

$$x_{n+2,j}^{(k)} = \frac{1}{n} \left[ \left( \sum_{i=1}^{n+1} x_{ij}^{(k)} \right) - x_{hj}^{(k)} \right] \quad (4.1)$$

onde  $j = 1, \dots, n$  designa cada direcção coordenada.

O poliedro inicial é geralmente um simplex regular (não tem obrigatoriamente de o ser) com ponto 1 na origem (centróide na origem).

O processo para encontrar um vértice em  $\mathbb{R}^n$  no qual  $f(\mathbf{x})$  tome um valor óptimo, envolve quatro operações (Himmelblau, 1972):

### 1 – Reflexão

Reflecte-se  $x_h^{(k)}$  através do centróide calculando:

$$x_{n+3}^{(k)} = x_{n+2}^{(k)} + \alpha (x_{n+2}^{(k)} - x_n^{(k)}) \quad (4.2)$$

onde  $\alpha > 0$  é o coeficiente de reflexão,  $x_{n+2}^{(k)}$  é o centróide calculado por (4.1) e  $x_h^{(k)}$  é o vértice no qual  $f(\mathbf{X})$  toma o maior valor dos  $n+1$  valores de  $f(\mathbf{X})$  na etapa  $k$ .

### 2 – Expansão

Se  $f(x_{n+3}^{(k)}) \leq f(x_l^{(k)})$ , expandir o vector  $(x_{n+3}^{(k)} - x_{n+2}^{(k)})$  calculando:

$$x_{n+4}^{(k)} = x_{n+2}^{(k)} + \gamma (x_{n+3}^{(k)} - x_{n+2}^{(k)}) \quad (4.3)$$

onde  $\gamma > 1$  é o coeficiente de expansão.

Se  $f(x_{n+4}^{(k)}) < f(x_l^{(k)})$ , substituir  $x_h^{(k)}$  por  $x_{n+4}^{(k)}$  e continuar a partir do primeiro passo, com  $k = k + 1$ . Caso contrário substituir  $x_h^{(k)}$  por  $x_{n+3}^{(k)}$  e continuar a partir do primeiro passo com  $k = k + 1$ .

### 3 – Contração

Se  $f(x_{n+3}^{(k)}) > f(x_l^{(k)})$  para todo  $i \neq h$ , contrair o vector  $(x_h^{(k)} - x_{n+2}^{(k)})$  calculando:



$$x_{n+5}^{(k)} = x_{n+2}^{(k)} + \beta(x_n^{(k)} - x_{n+2}^{(k)}) \quad (4.4)$$

onde  $0 < \beta < 1$  é o coeficiente de contracção.

Substituir  $x_h^{(k)}$  por  $x_{n+5}^{(k)}$ , voltar ao passo um e continuar a pesquisa na etapa  $k + 1$ .

#### 4 – Redução

Se  $f(x_{n+3}^{(k)}) > f(x_h^{(k)})$  reduzir os vectores  $(x_i^{(k)} - x_j^{(k)})$   $i = 1, \dots, n + 1$  de metade a partir de  $x_j^{(k)}$ , calculando:

$$x_i^{(k)} = x_j^{(k)} + 0,5(x_i^{(k)} - x_j^{(k)}) \quad (4.5)$$

com  $i = 1, \dots, n + 1$  e voltar ao passo um continuando a pesquisa na etapa  $n + 1$ .

O critério usado por Nelder e Mead para terminar a pesquisa é testar se

$$\left\{ \frac{1}{n+1} \sum_{i=1}^{n+1} [f(x_i^{(k)}) - f(x_{n+2}^{(k)})]^2 \right\}^{\frac{1}{2}} \leq \varepsilon \quad (4.6)$$

com  $\varepsilon$  arbitrariamente pequeno e onde  $f(x_{n+2}^{(k)})$  é o valor da função objectivo no centróide  $x_{n+2}^{(k)}$  tomado por ponto óptimo.

Nelder e Mead aconselham tomar  $\alpha = 1$ ,  $\beta = 0,5$  e  $\gamma = 2$ , em problemas de minimização não sujeita a restrições.

Para exemplificar considere-se a minimização da função,  $f(\mathbf{x}) = 4(x_1 - 5)^2 + (x_2 - 6)^2$  que tem um mínimo em  $\mathbf{x}^* = [5 \ 6]^T$ . Neste caso estão duas variáveis envolvidas logo o polígono inicial conterá três vértices em  $\mathbf{x}_1^{(0)} = [8 \ 9]^T$ ,  $\mathbf{x}_2^{(0)} = [10 \ 11]^T$  e  $\mathbf{x}_3^{(0)} = [8 \ 11]^T$ .

Na etapa de pesquisa  $k = 0$  temos,  $f(8,9) = 45$ ,  $f(10,11) = 125$  e  $f(8,11) = 65$ .  $\mathbf{x}_2^{(0)} = [10 \ 11]^T$  é reflectido através do centróide de  $\mathbf{x}_1^{(0)}$  e  $\mathbf{x}_3^{(0)}$ , isto é  $\mathbf{x}_4^{(0)}$  (calculado de acordo com 4.1),

$$\mathbf{x}_{4,1}^{(0)} = \frac{1}{2}[(8+10+8) - 10] = 8$$

$$\mathbf{x}_{4,2}^{(0)} = \frac{1}{2}[(9+11+11) - 11] = 10$$

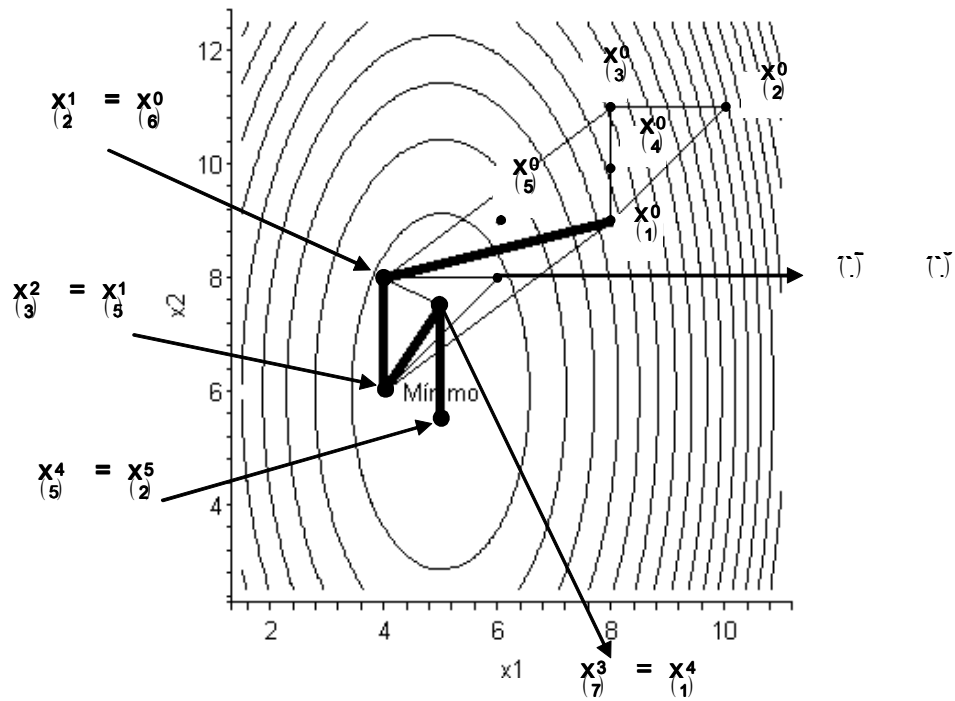
e  $f(8,10) = 52$ .

para se obter  $\mathbf{x}_5^{(0)}$  (calculado de acordo com 4.2),

$$\mathbf{x}_{5,1}^{(0)} = 8 + 1(8 - 10) = 6$$

$$\mathbf{x}_{5,2}^{(0)} = 10 + 1(10 - 11) = 9$$

e  $f(6,9) = 13$ .



**Figura 4.4** – Exemplificação da trajectória de Nelder e Mead.

Uma vez que  $f(6,9) = 13 < f(8,9) = 45$  vai proceder-se a uma expansão para se obter  $x_6^{(0)}$  (calculado de acordo com 4.3),

$$x_{6,1}^{(0)} = 8 + 2(6 - 8) = 4$$

$$x_{6,2}^{(0)} = 10 + 2(9 - 10) = 8$$

e  $f(4,8) = 8$ .

**Quadro 4.1** – Etapas de cálculo pelo método de Nelder e Mead para minimizar  $f(\mathbf{x}) = 4(x_1 - 5)^2 + (x_2 - 6)^2$ .

Etapa	Vértices		Valores de $f(\mathbf{x})$	Notas	
	$x_1$	$x_2$			
0	$x_1^{(0)}$	8	9	45	$x_l = x_1^{(0)}$
0	$x_2^{(0)}$	10	11	125	$x_h = x_2^{(0)}$
0	$x_3^{(0)}$	8	11	61	
0	$x_5^{(0)}$	6	9	13	Obtido por reflexão pela Eq. (4.2)
0	$x_2^{(1)} = x_6^{(0)}$	4	8	8	Obtido por expansão pela Eq. (4.3); substituir $x_2^{(0)}$ por $x_6^{(0)} = x_2^{(1)}$
1	$x_3^{(2)} = x_5^{(1)}$	4	6	4	Substituir $x_3^{(0)}$ por $x_5^{(1)} = x_3^{(2)}$
2	$x_1^{(3)} = x_7^{(2)}$	6	8	8	Substituir $x_1^{(0)}$ por $x_7^{(2)} = x_1^{(3)}$
3	$x_1^{(4)} = x_7^{(3)}$	5	7,5	2,25	Substituir $x_1^{(3)}$ por $x_7^{(4)}$
4	$x_2^{(5)} = x_5^{(4)}$	5	5,5	0,25	Substituir $x_2^{(1)}$ por $x_5^{(5)}$

Uma vez que  $f(4,8) = 8 < f(8,9) = 45$  substitui-se  $x_2^{(0)}$  por  $x_6^{(0)}$  que passa a designar-se por  $x_2^{(1)}$  na etapa de pesquisa seguinte.

Finalmente, uma vez que,

$$\frac{1}{3} \left[ (45 - 52)^2 + (61 - 52)^2 + (8 - 52)^2 \right]^{\frac{1}{2}} > 10^{-6}$$

$$\frac{1}{3} \left[ 7^2 + 9^2 + 44^2 \right]^{\frac{1}{2}} > 10^{-6}$$

dá-se início à etapa de pesquisa  $k = 1$ .

A Figura 4.4 ilustra a pesquisa nestas etapas iniciais e o Quadro 4.1 mostra as coordenadas dos vértices e os valores de  $f(\mathbf{x})$  para quatro etapas de pesquisa adicionais.

Na Figura 4.5 apresenta-se o fluxograma do processo iterativo e que servirá de base à elaboração do programa utilizado posteriormente.

### 4.3. Método de Rosenbrock

Antes de descrever o método para o caso de  $n$  variáveis, vamos detalhadamente exemplificar a sua aplicação no caso de duas variáveis (Rosenbrock, 1960). Para tal consideremos as isolinhas dos valores de uma função objectivo conforme Figura 4.6.

Antes de se iniciar o processo de minimização define-se um conjunto de incrementos iniciais  $\lambda_1$  e  $\lambda_2$ , a praticar segundo as direcções  $D_1$  e  $D_2$ , inicialmente coincidentes com os eixos coordenados.

Parte-se de um ponto arbitrariamente escolhido  $x_1$  (ponto 1 na figura) na primeira etapa de minimização.

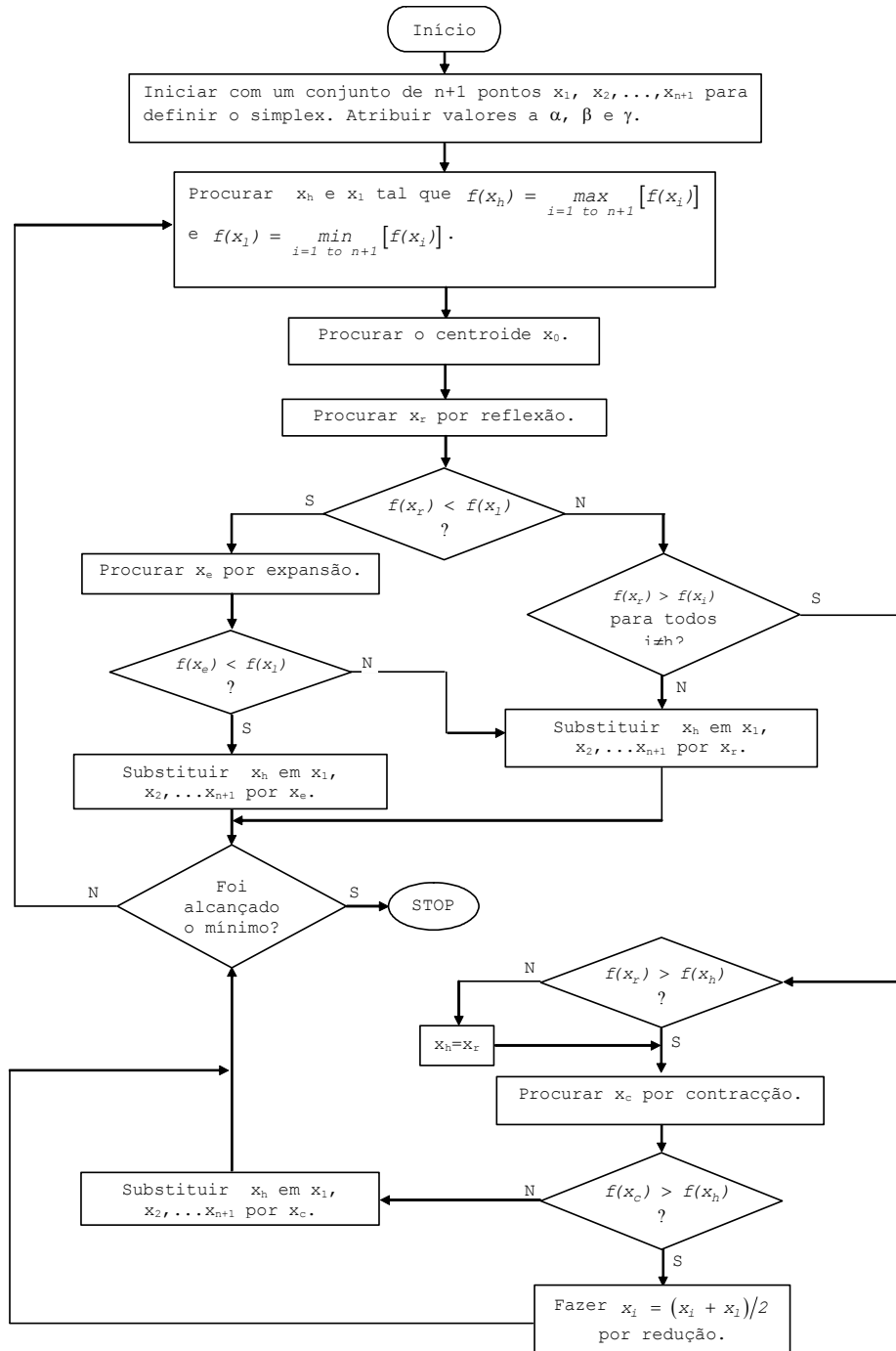


Figura 4.5 – Método de Nelder e Mead.

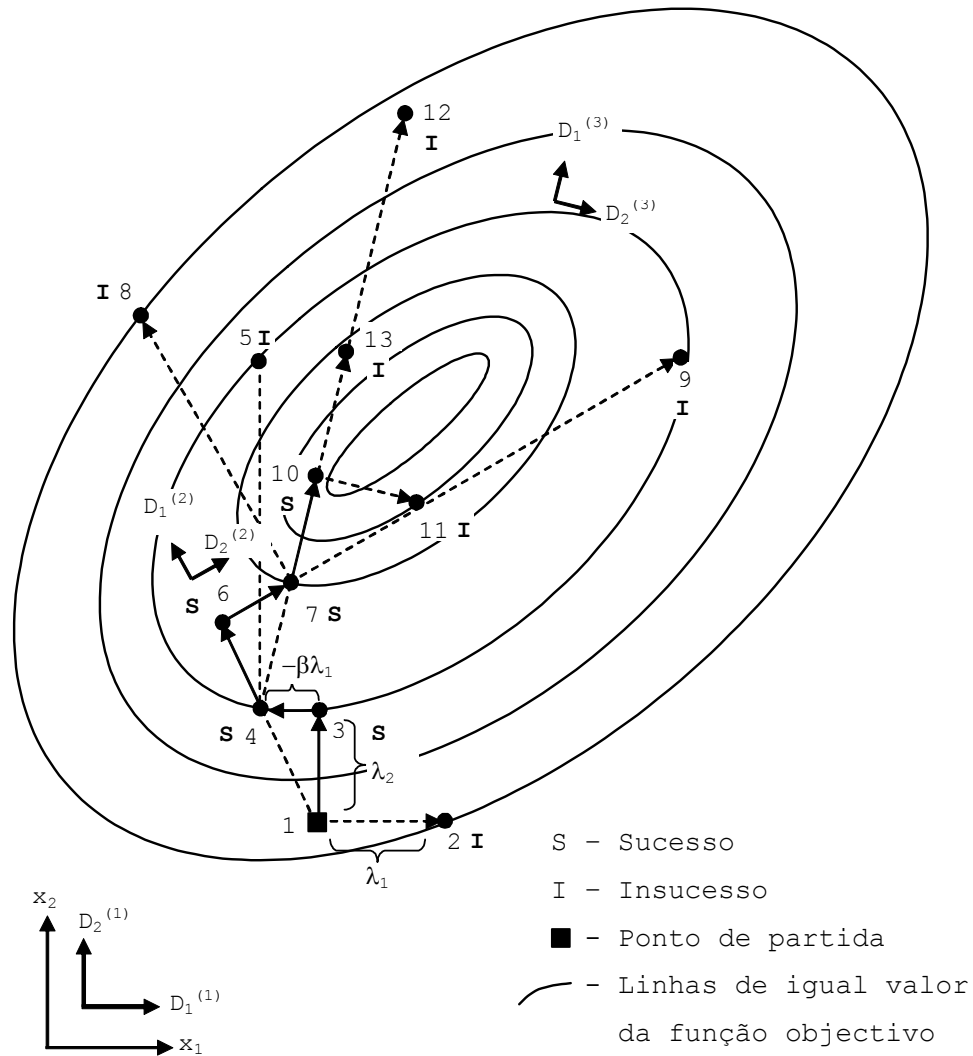


Figura 4.6 – Exemplificação da trajectória de Rosenbrock.

Na pesquisa do mínimo ao longo da primeira direcção  $D_1^{(1)}$ , incrementa-se o ponto 1 de uma quantidade  $\lambda_1$  e obtém-se o ponto 2. Neste caso temos  $f(x_2) > f(x_1)$  e portanto, um insucesso.

Quando se obtém um insucesso, o incremento, na mesma direcção, para a nova iteração, será  $-\beta\lambda_1$  com  $0 < \beta < 1$ . No caso de sucesso este incremento será de  $\alpha\lambda_1$ , com  $\alpha > 1$ .

Os valores recomendados por Rosenbrock são respectivamente de  $\beta = 1/2$  e  $\alpha = 3$ .

No exemplo figurado, o incremento de pesquisa segundo  $D_1^{(1)}$  na iteração seguinte é  $-\beta\lambda_1$ . A pesquisa é agora feita a partir do ponto 1 segundo  $D_2^{(1)}$ , obtendo-se o ponto 3 e verificando-se um sucesso.

Assim, na próxima iteração, o incremento segundo  $D_2^{(1)}$  será igual a  $\alpha\lambda_2$ . Tomando o ponto 3 como novo ponto base e incrementando-o de  $-\beta\lambda_1$  segundo  $D_1^{(1)}$ , obtemos um sucesso, definindo o ponto 4. Na próxima iteração o incremento a praticar na mesma direcção é  $\alpha(-\beta\lambda_1)$ . A partir do ponto 4 e segundo  $D_2^{(1)}$ , incrementando-o de  $\alpha\lambda_2$ , obtendo-se o ponto 5 que é um insucesso. Então conservamos o ponto 4 como melhor ponto nesta etapa.

Como já obtivemos um sucesso e um insucesso em cada uma das direcções  $D_1^{(1)}$  e  $D_2^{(1)}$ , terminamos esta etapa.

Ao iniciar-se uma nova etapa de minimização é preciso modificar as direcções  $D_1^{(1)}$  e  $D_2^{(1)}$  para  $D_1^{(2)}$  e  $D_2^{(2)}$ , segundo o processo de ortogonalização de Gram-Schmidt. Geometricamente, a nova direcção



$D_1^{(2)}$  será a da linha unindo o ponto base do início da primeira etapa, ponto 1, com o último ponto desta etapa, ponto 1.

Parte-se do ponto 4 como novo ponto base. Incrementando-o de  $\lambda_1$  segundo  $D_1^{(2)}$ , obtemos um sucesso, com o ponto 6. A partir deste ponto e com um incremento de  $\lambda_2$  segundo  $D_2^{(2)}$  define-se o ponto 7. Como ambos são sucesso os incrementos passarão agora a ser iguais a  $\alpha\lambda_1$  e  $\alpha\lambda_2$ . Assim, a partir do ponto 7, e com um incremento de  $\alpha\lambda_1$  segundo  $D_1^{(2)}$  definimos o ponto 8. Como é um insucesso partimos do ponto 7, com um incremento de  $\alpha\lambda_2$  segundo  $D_2^{(2)}$  e alcançamos o ponto 9 que é também um insucesso. Assim, novamente a partir do ponto 7, e uma vez que em cada direcção já obtivemos um insucesso, definimos as novas direcções  $D_1^{(3)}$  e  $D_2^{(3)}$  e continuamos com a terceira etapa de minimização.

Este procedimento é continuado até se alcançar um número de iterações pré determinado, ou até que a distância entre pontos de partida e de finalização de qualquer etapa seja inferior a certo valor muito pequeno  $\varepsilon$  previamente fixado.

Vamos agora generalizar o método de Rosenbrock para o caso da minimização de uma função a  $n$  variáveis (Figura 4.7). Antes de iniciar o processo de minimização escolhemos um conjunto de incrementos  $\lambda_1, \lambda_2, \dots, \lambda_n$  a serem praticados ao longo das direcções de pesquisa  $D_1, D_2, \dots, D_n$ , respectivamente. Para a  $j$ -ésima etapa podemos resumir o processo deste modo:

a) O conjunto de direcções  $D_1^{(j)}, D_2^{(j)}, \dots, D_n^{(j)}$  e o ponto base são conhecidos no começo da etapa.

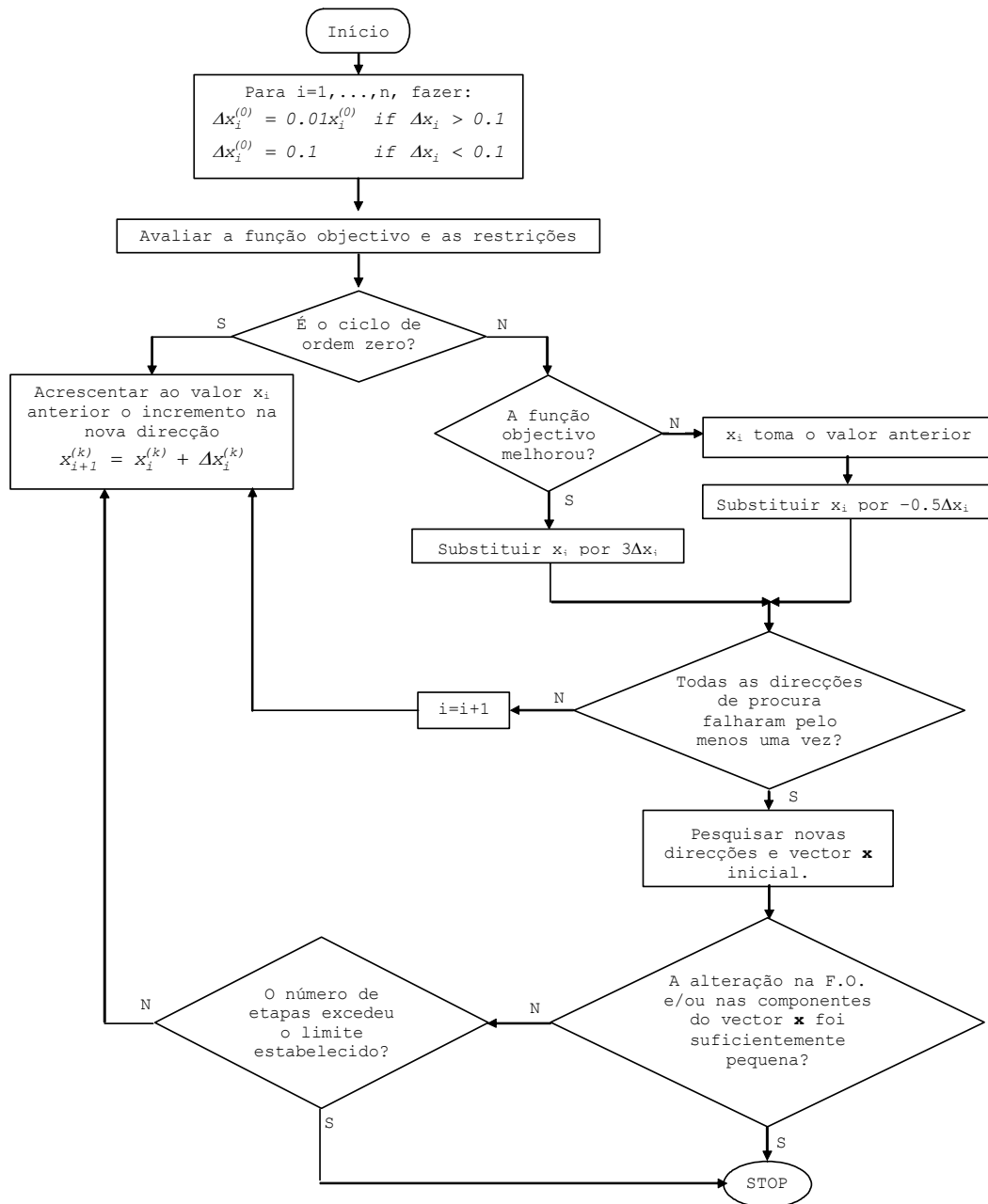


Figura 4.7 – Fluxograma do algoritmo de Rosenbrock.

Um incremento  $\lambda_1$  é tomado na direcção  $D_1^{(j)}$  a partir do ponto base. Se houver sucesso,  $\lambda_1$  é multiplicado por  $\alpha$ , o novo ponto é mantido e anota-se o sucesso. Se houver insucesso,  $\lambda_1$  é multiplicado por  $-\beta$  e regista-se um insucesso. Os valores de  $\alpha$  e  $\beta$  são usualmente 3,0 e 0,5 respectivamente.

b) Continua-se a pesquisa sequencialmente segundo as direcções  $D_1^{(j)}, D_2^{(j)}, \dots, D_{n-1}^{(j)}, D_n^{(j)}, D_1^{(j)}, D_2^{(j)}, \dots$  até se obter pelo menos um sucesso e um insucesso em cada uma das  $n$  direcções  $D_1^{(j)}, D_2^{(j)}, \dots, D_n^{(j)}$ .

c) Calcular o novo conjunto de direcções  $D_1^{(j+1)}, D_2^{(j+1)}, \dots, D_n^{(j+1)}$  para utilizar na próxima etapa  $j+1$ , pelo processo de ortogonalização de Gram-Schmidt.

d) Tomar o melhor ponto da etapa  $j$ -ésima como ponto base para a próxima etapa e repetir o processo desde a).

e) Assumir a convergência quer após um número de iterações pré determinado ou após a distância entre os pontos iniciais e de chegada se tornar inferior a  $\varepsilon$ , fixado previamente.



## **5. EXEMPLOS DE APLICAÇÃO**

Vamos mostrar aplicações práticas destes métodos, para o que previamente se fizeram adaptações de programas existentes, tendo escolhido como linguagem de programação o Visual Basic (6.0). Os referidos programas encontram-se num CD em Anexo.

Começa-se com o cálculo do mínimo de uma função com mais de uma variável.

Segue-se o ajustamento de uma função a dados experimentais, quer ela seja de uma ou mais variáveis e também exemplos de índole hidrológica.

Finalmente apresentamos a optimização dos parâmetros de um modelo para conversão da precipitação em escoamento.

Para cada um destes casos serão discutidos os diferentes métodos utilizados.

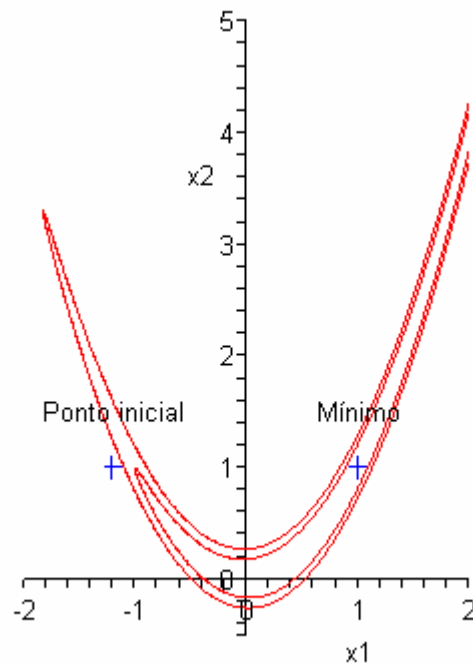
### **5.1. Determinação do Mínimo de uma Função**

#### **5.1.1. Aplicação à Função de Rosenbrock**

A função de Rosenbrock é dada por,

$$y = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

cuja dificuldade é apontada por vários autores e de que apresentamos na Figura 5.1 dois contornos correspondentes às isolinhas  $f(x) = 8$  e  $f(x) = 4$ , respectivamente.



**Figura 5.1** – Isolinhas da função de Rosenbrock.

Geometricamente  $f(X)$  é interpretada como um “vale” cujo mínimo é igual a 0, isto é,  $f(X^*) = 0$ , para o ponto  $(1,1)$ .

Partimos do ponto inicial  $(-1,2;1)$ .

## Método de Monro

Para a aplicação deste método à minimização da função de Rosenbrock utilizamos o programa “MONROMINIMO” onde esta função aparece na forma,

$$y = 100 \left[ A(2) - A(1)^2 \right]^2 + \left[ 1 - A(1) \right]^2.$$

Aqui, como se vê, os parâmetros  $A(1)$  e  $A(2)$  substituem as variáveis  $x_1$  e  $x_2$ .

Os dados que o programa necessita são:

A(I) - Valores iniciais dos parâmetros, com  $I = 1,2$ ;

DDELTA (I) - Valores iniciais dos acréscimos, com  $I = 1,2$ ;

CHECKL(I) - Limite inferior dos parâmetros, com  $I = 1,2$ ;

CHECKH(I) - Limite superior dos parâmetros, com  $I = 1,2$ ;

NUMA – Número de parâmetros a serem otimizados;

NPER = 1 se os acréscimos são tomados em percentagem;

NPER = 0 se os acréscimos são tomados em valor absoluto;

KC - Número máximo de vezes que os acréscimos podem ser reduzidos para metade;

MAXN - Número máximo de vezes que o programa pode chamar a otimização.

Na Figura 5.2 mostra-se a subrotina do programa “MONROMINIMO” onde é possível alterar os dados iniciais e a função a otimizar.

```
#####
##          PROGRAMA MONROMINIMO          ##
## Programa para determinação do mínimo de uma função      ##
## pelo método de Hooke e Jeeves modificado por Monro      ##
#####
Private Sub Form_Load()
' Abertura do ficheiro para gravação dos resultados
Open "MONROMIN.TXT" For Output As #1
#####
'          DADOS A ALTERAR
' NUMA - Nº de parâmetros a otimizar
' NPER = 0 - Acréscimos são tomados em valor absoluto
' NPER = 1 - Acréscimos são tomados em percentagem
' KC - Nº máximo de vezes que os acréscimos podem ser reduzidos
'   para metade
' MAXN - Nº máximo de vezes que o programa pode chamar a rotina OPT
' A(I) - Valores iniciais dos parâmetros
' DDELTA(I) - Valores iniciais dos acréscimos
' CHECKL(I) - Limite inferior dos parâmetros
' CHECKH(I) - Limite superior dos parâmetros
' Y - Função a minimizar
#####
NUMA = 2
NPER = 0
KC = 20
MAXN = 250
A(1) = -1.2
A(2) = 1
DDELTA(1) = 0.05
DDELTA(2) = 0.05
CHECKL(1) = -9
CHECKL(2) = -9
CHECKH(1) = 10
CHECKH(2) = 10
NSTART = 0
8000:
Y = 100 * (A(2) - A(1) ^ 2) ^ 2 + (1 - A(1)) ^ 2
OPTIM = Y
Call opt
GoTo 8000
End
End Sub
```

**Figura 5.2** – Subrotina Form Load do Programa “MONROMINIMO”.



Partindo dos dados apresentados no Quadro 5.1 e para avaliar a influência do número de chamadas à otimização (MAXN), ensaiámos os três casos apresentados no Quadro 5.2.

No Quadro 5.3 apresentam-se os resultados obtidos para os diferentes valores de MAXN considerados. A sequência detalhada dos resultados pode ser vista na janela “Immediate” em cada corrida do programa “MONROMINIMO” ou no ficheiro de texto “MONROMIN.TXT” criado pelo programa.

Confrontando os resultados obtidos com os verdadeiros resultados de  $f(X^*) = 0$  para  $x_1 = 1$  e  $x_2 = 1$ , pensamos que com MAXN = 400 já se obtém um valor de  $f(X)$  bastante satisfatório, embora com MAXN = 500 se obtenha um melhor resultado.

**Quadro 5.1** – Dados iniciais

Ordem	Parâmetro	Acréscimo	Limite Inferior	Limite Superior
I	A(I)	DDELTA(I)	CHECKL(I)	CHECKH(I)
1	-1,2	0,01	-9,0	10,0
2	1,0	0,01	-9,0	10,0

**Quadro 5.2** – Casos ensaiados

NUMA	NPER	KC	MAXN
2	0	20	250
2	0	20	400
2	0	20	500

**Quadro 5.3 – Resultados**

<b>MAXN</b>	$f(X)$	$x_1$	$x_2$
<b>250</b>	1,3389E-04	1,0115	1,0232
<b>400</b>	4,0218E-08	1,0002	1,0004
<b>500</b>	4,1807E-10	1,0000	1,0000

A fim de avaliar a influência dos valores iniciais dos acréscimos, (DDELTA(I), I = 1, 2) ensaiámos, para os três casos apresentados anteriormente, três valores diferentes para estes acréscimos conforme se apresenta no Quadro 5.4. No Quadro 5.5 apresentam-se os resultados obtidos.

**Quadro 5.4 – Dados iniciais**

<b>Ordem</b>	<b>Parâmetro</b>	<b>Acréscimo</b>	<b>Limite Inferior</b>	<b>Limite Superior</b>
<b>I</b>	<b>A(I)</b>	<b>DDELTA(I)</b>	<b>CHECKL(I)</b>	<b>CHECKH(I)</b>
1	-1,2	0,01	-9,0	10,0
2	1,0	0,01	-9,0	10,0
1	-1,2	0,10	-9,0	10,0
2	1,0	0,10	-9,0	10,0
1	-1,2	0,05	-9,0	10,0
2	1,0	0,05	-9,0	10,0

Analisando os resultados, podemos verificar que nem sempre o maior valor de MAXN conduz ao melhor resultado. Assim, tomando MAXN = 250, o melhor resultado obtém-se com DDELTA(I) = 0,10, mas com MAXN = 400 ou MAXN = 500 já o melhor resultado é obtido com DDELTA(I) = 0,05. O resultado mais próximo do verdadeiro mínimo é obtido, para estes casos, com MAXN = 500 e DDELTA(I) = 0,05.

**Quadro 5.5** – Resultados

MAXN	DDELTA(I)	$f(X)$	$x_1$	$x_2$
250	0,01	1,3389E-04	1,0115	1,0232
	0,05	2,2699E-04	1,0139	1,0285
	0,10	3,7266E-05	1,0061	1,0122
400	0,01	4,0218E-08	1,0002	1,0004
	0,05	1,6357E-10	1,0000	1,0000
	0,10	2,0142E-08	1,0000	0,9997
500	0,01	4,1807E-10	1,0000	1,0000
	0,05	8,7576E-11	1,0000	1,0000
	0,10	1,4844E-09	1,0000	1,0000

## Método de Nelder e Mead

Para a minimização da função de Rosenbrock por aplicação do método de Nelder e Mead, utilizámos o programa “SIMPLEXMINIMO”

sendo necessário escrever a função na subrotina *SUMR*. Na Figura 5.3 mostra-se esta subrotina onde a função a minimizar aparece na forma,

$$SUM(NN) = 100 [Z(2) - Z(1)^2]^2 + [1 - Z(1)]^2.$$

Neste caso os parâmetros  $Z(1)$  e  $Z(2)$  substituem as variáveis  $x_1$  e  $x_2$ .

```
Public Sub SUMR(ByVal NN As Integer, Z())
*****
'          ROTINA SUMR
*****
C = C + 1
*** NA FUNÇÃO OS PARÂMETROS X(I) APARECEM COMO Z(I) ***
SUM(NN) = 100 * (Z(2) - Z(1) ^ 2) ^ 2 + (1 - Z(1)) ^ 2
*** Imprime na janela IMEDIATE ***
Debug.Print "C="; C
*** Imprime no ficheiro SIMPLEXMINIMO.TXT ***
Print #1, "C="; C
End Sub
```

**Figura 5.3** – Subrotina SUMR do Programa “SIMPLEXMINIMO”.

Os dados que o programa necessita são:

NX – Número de parâmetros a otimizar;

X(I) - Valores iniciais dos parâmetros, com  $I = 1, 2$ ;

STEPE - Passo inicial;

E - Constante arbitrariamente pequena para critério de paragem.

Na Figura 5.4 mostra-se parte da subrotina do programa “SIMPLEXMINIMO” onde é possível alterar os dados iniciais.

```

#####
##          PROGRAMA SIMPLEXMINIMO          ##
## Programa para determinação do mínimo de uma função      ##
## pelo método de Nelder e Mead                    ##
#####

Private Sub Form_Load()
Dim X1(50, 50), X(50)
' Abertura do ficheiro para gravação dos resultados
Open "SIMPLEXMINIMO.TXT" For Output As #1
#####
'          DADOS A ALTERAR
' NX - Nº de parâmetros a otimizar
' STEPE - Passo inicial
' X(I) - Valores iniciais dos parâmetros
' E - Constante arbitrariamente pequena para critério de paragem
' SUM(NN) - Função a minimizar, alterada na subrotina SUMR
#####
NX = 2
STEPE = 0.01
X(1) = -1.2: X(2) = 1
E = 0.0000001
C = 0
C1 = 0
ALFA = 1
BETA = 0.5
GAMA = 2
DIFER = 0
XNX = NX
INA = 1
Call SUMR(INA, X())

```

**Figura 5.4** – Subrotina Form Load do Programa “SIMPLEXMINIMO”

Partindo de  $X(1) = -1,2$  e  $X(2) = 1$  como valores iniciais para os parâmetros e com o objectivo de avaliar a influência do STEPE, ensaiámos os três casos apresentados no Quadro 5.6.

No Quadro 5.7 apresentam-se os resultados obtidos. A sequência detalhada dos resultados pode ser vista na janela “Immediate” em cada corrida do programa “SIMPLEXMINIMO” ou no ficheiro de texto “SIMPLEMINIMO.TXT” criado pelo programa.

**Quadro 5.6** – Casos ensaiados

<b>NX</b>	<b>E</b>	<b>STEPE</b>
2	1E-07	0,01
2	1E-07	0,05
2	1E-07	0,10

**Quadro 5.7** – Resultados

<b>STEPE</b>	$f(X)$	$x_1$	$x_2$
0,01	1.6171E-07	1,0000	1,0000
0,05	5.8095E-08	1,0002	1,0005
0,10	1.3568E-07	0.9999	1,0000

Analisando os resultados, podemos verificar que o resultado mais próximo do verdadeiro mínimo é obtido com um STEPE = 0,05.

## Método de Rosenbrock

Para a minimização da função de Rosenbrock com o método de Rosenbrock utilizámos o programa “ROSENBROCKMINIMO” sendo necessário escrever a função na subrotina FUNCAO. Na figura 5.5 mostra-se esta subrotina onde esta função aparece na forma,

$$F1 = 100 \left[ X(2) - X(1)^2 \right]^2 + \left[ 1 - X(1) \right]^2.$$

Aqui, como se vê, os parâmetros  $X(1)$  e  $X(2)$  substituem as variáveis  $x_1$  e  $x_2$ .

```

Sub FUNCAO()
'*****
'          ROTINA FUNÇÃO
'*****
ITERC = K5
For I = 1 To NOPT
  IK = IOPT(I)
  X(IK) = Y(I)
Next I
F1 = 100 * (X(2) - X(1) ^ 2) ^ 2 + (1 - X(1)) ^ 2
For I = 1 To NOPT
  IK = IOPT(I)
  Y(I) = X(IK)
Next I
End Sub

```

**Figura 5.5** – Subrotina FUNCAO do Programa “ROSENBROCKMINIMO”.

Os dados que o programa necessita são:

NK – Número de iterações;

NP – Número total de parâmetros;

NOPT – Número de parâmetros a otimizar;

X(I) – Valores iniciais dos parâmetros, com  $I = 1,2$ ;

XI(I) – Limite inferior dos parâmetros, com  $I = 1,2$ ;

XS(I) – Limite superior dos parâmetros, com  $I = 1,2$ .

Na Figura 5.6 mostra-se parte da subrotina do programa “ROSENBROCKMINIMO” onde é possível alterar os dados iniciais.

```

#####
##          PROGRAMA ROSENBROCKMINIMO          ##
## Programa para determinação do mínimo de uma função pelo      ##
## método de Rosenbrock                                     ##
#####

Private Sub Form_Load()
' Abertura do ficheiro para gravação dos resultados
Open "ROSENBROCKMINIMO.TXT" For Output As #1
#####
'
'          DADOS A ALTERAR
' NK - N° de iterações
' NP - N° total de parâmetros
' NOPT - N° de parâmetros a otimizar
' X(l) - Valores iniciais dos parâmetros
' XI(l) - Limite inferior dos parâmetros
' XS(l) - Limite superior dos parâmetros
' F1 - Função a minimizar, alterada na Subrotina FUNCAO
#####
NK = 2500
NP = 2
NOPT = 2
X(1) = -1.2
X(2) = 1
XI(1) = -50
XI(2) = -50
XS(1) = 50
XS(2) = 50

```

**Figura 5.6** – Subrotina Form Load do Programa “ROSENBROCKMINIMO”.

Partindo de  $X(1) = -1,2$  e  $X(2) = 1$  como valores iniciais para os parâmetros e com o objectivo de avaliar a influência do número de iterações (NK), ensaiámos diversos valores de NK. Os resultados obtidos estão apresentados no Quadro 5.8.

É surpreendente que para se obter  $f(\mathbf{X}) = 5,2532 \times 10^{-7}$  sejam necessárias 20000 iterações que conduziram à solução,  $x_1 = 0,9993$  e  $x_2 = 0,9985$ . Com 40000 iterações os resultados melhoram substancialmente aproximando-se da verdadeira solução  $f(\mathbf{X}) = 5,5127 \times 10^{-12}$  com  $x_1 = 1,0000$  e  $x_2 = 1,0000$ .



Para tentar melhoria nos resultados alterámos os valores dos incrementos  $\alpha$  e  $\beta$ . Neste caso apenas modificámos  $\beta$  de  $\beta = 1,5$  para  $\beta = 2,0$ . No quadro 5.9 apresentam-se os resultados.

**Quadro 5.8** – Resultados

<b>NK</b>	$f(X)$	$x_1$	$x_2$
500	2,2760	-0,5047	0,2749
1000	0,1408	0,6249	0,3915
2500	2,3862E-02	0,8456	0,7165
10000	1,6348E-04	0,9872	0,9872
15000	9,1883E-06	0,9970	0,9939
20000	5,2532E-07	0,9993	0,9985
30000	1,7136E-09	1,0000	0,9999
40000	5,5127E-12	1,0000	1,0000
50000	1,8108E-14	1,0000	1,0000
60000	5,3214E-17	1,0000	1,0000

**Quadro 5.9** – Resultados

<b>NK</b>	$f(X)$	$x_1$	$x_2$
500	0,4100	0,3697	0,1114
1000	6,1397E-03	1,0943	0,8977
2500	8,0157E-04	0,9712	0,9441
10000	8,0157E-04	0,9712	0,9441

Como se pode verificar da comparação dos Quadros 5.8 e 5.9, os resultados para o mesmo número de iterações e até 2500 são substancialmente melhores. No entanto neste último caso o aumento do número de iterações não conduz à alteração da função objectivo enquanto que no primeiro caso o aumento do número de iterações conduz a uma melhoria considerável mas lenta da função objectivo.

No intuito de obviar este inconveniente, construímos um programa mais simples apenas para minimizar funções com duas variáveis, que passa a designar-se por “ROSENBROCKMINIMO\_2X”.

Os dados que este programa necessita são apenas:

NK – Número de iterações;

$X(I)$  – Valores iniciais dos parâmetros, com  $I = 1, 2$ ;

$DELTA(I)$  – Acréscimos iniciais na pesquisa das variáveis, com  $I = 1, 2$ ;

Na figura 5.7 apresenta-se parte da subrotina onde se pode alterar estes dados.

Partindo dos mesmos valores iniciais,  $X(1) = -1,2$  e  $X(2) = 1$  e com  $DELTA(1) = DELTA(2) = 0,1$ , ensaiámos também diversos valores de NK. Os resultados estão apresentados no quadro 5.10.

Como se pode verificar, a solução óptima é atingida apenas com 1000 iterações pelo que para a minimização de funções com duas variáveis sugerimos a utilização deste último programa.

```

#####
##          PROGRAMA ROSENBROCKMINIMO_2x          ##
##  Programa para determinação do mínimo de uma função  ##
##  com duas variáveis pelo método de Rosenbrock      ##
#####

Private Sub Form_Load()
' Abertura do ficheiro para gravação dos resultados
Open "ROSENBROCKMINIMO_2X.TXT" For Output As #1
#####
'          DADOS A ALTERAR
' NK - Nº de iterações
' X0(I) - Valores iniciais dos parâmetros
' DELTA(I) - Acréscimos iniciais na pesquisa das variáveis
' F1 - Função a minimizar
#####
Dim X(15), DELTA(7), S(2, 2), SU(7), FA(7), X0(15), XB(15)
Dim A1(2), A2(2), B1(2), B2(2)
Dim DIR
CONT = 0
NK = 250
X0(1) = -1.2
X0(2) = 1
DELTA(1) = 0.1
DELTA(2) = 0.1
k = 0
Call FUN(F, X0())
F0 = F
Print #1, "f="; F
Debug.Print "f="; F
'DIRECÇÕES
S(1, 1) = 1: S(1, 2) = 0: S(2, 1) = 0: S(2, 2) = 1
'J=ETAPA
'K=CICLO
10: J = 1
Print #1, "F0="; F

```

**Figura 5.7** – Subrotina Form Load do Programa “ROSENBROCKMINIMO\_2X”.

**Quadro 5.10** – Resultados

NK	$f(\mathbf{x})$	$x_1$	$x_2$
250	2,5009E-04	0,9846	0,96991
500	6,3456E-18	1,0000	1,0000
1000	0	1	1

Como esta modificação exige determinados passos não detalhados anteriormente, julgámos útil incluir um cálculo pormenorizado de uma etapa de cálculos que conduziram aos resultados acima citados.

Na etapa inicial ( $J = 0$ ), partimos dos valores iniciais  $x_1^{(0)} = -1,2$  e  $x_2^{(0)} = 1$ , isto é,  $\mathbf{x}^{(0)} = [-1,2 \quad 1,0]^T$  onde  $f[\mathbf{x}(0)] = 24,2$  e tomámos os acréscimos iniciais  $\Delta_1^{(0)} = 0,1$  e  $\Delta_2^{(0)} = 0,1$ .

As direcções de pesquisa foram as dos eixos coordenados, isto é,

$$\mathbf{s}_1^{(0)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ e } \mathbf{s}_2^{(0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Para a iteração  $k = 1$  vem,

$$x_1^{(1)} = -1,2 + 0,1 \times 1 = -1,1$$

e

$$x_2^{(1)} = 1,0 + 0,1 \times 0 = 1,0,$$

com  $f(\mathbf{x}^{(1)}) = 8,82$ .

Uma vez que  $f(\mathbf{x}^{(1)}) = 8,82 < f(\mathbf{x}^{(0)}) = 24,2$  é verdadeiro, obtemos um SUCESSO e o novo óptimo é  $f(\mathbf{x}^{(1)}) = 8,82$ .

Na iteração  $K = 2$  temos,

$$x_1^{(2)} = -1,1 + 0,1 \times 0 = -1,1$$

e

$$x_2^{(2)} = 1,0 + 0,1 \times 1 = 1,1,$$

com  $f(\mathbf{x}^{(2)}) = 5,62$ .

Como  $f(\mathbf{x}^{(2)}) = 5,62 < f(\mathbf{x}^{(1)}) = 8,82$  é verdadeiro obtemos um novo SUCESSO e o novo óptimo é  $f(\mathbf{x}^{(2)}) = 5,62$ .

Na iteração seguinte,  $K = 3$ , os acréscimos são alterados,

$$\Delta_1^{(1)} = 3 \times 0,1 = 0,3$$

e

$$\Delta_2^{(1)} = 3 \times 0,1 = 0,3.$$

Logo, vem,

$$x_1^{(3)} = -1,1 + 0,3 \times 1 = -0,8$$

e

$$x_2^{(3)} = 1,1 + 0,3 \times 0 = 1,1,$$

com  $f(\mathbf{x}^{(3)}) = 24,4$ .

Como  $f(\mathbf{x}^{(3)}) = 24,4 < f(\mathbf{x}^{(2)}) = 5,62$  é falso obtemos um INSUCESSO continuando o óptimo a ser  $f(\mathbf{x}^{(2)}) = 5,62$ .

Para  $K = 4$  vem,

$$x_1^{(4)} = -1,1 + 0,3 \times 0 = -1,1$$

e

$$x_2^{(4)} = 1,1 + 0,3 \times 1 = 1,4,$$

com  $f(\mathbf{x}^{(4)}) = 8,02$ .

Sendo falso  $f(\mathbf{x}^{(4)}) = 8,02 < f(\mathbf{x}^{(2)}) = 5,62$  obtemos um novo INSUCESSO continuando o óptimo a ser  $f(\mathbf{x}^{(2)}) = 5,62$ .

Esquematizando,

Direcções	
$s_1$	$s_2$
SUCESSO	SUCESSO
INSUCESSO	INSUCESSO

Quando, como se vê acima, um SUCESSO é seguido de um INSUCESSO em cada uma das direcções, a etapa termina e a nova etapa ( $J = 1$ ) exige que se proceda à rotação dos eixos. Para tal vejamos qual foi o ponto óptimo obtido e o ponto de partida inicial (ponto base),

$$\mathbf{x}^{(2)} = [-1,1 \quad 1,1]^T = \text{ponto óptimo}$$

e

$$\mathbf{x}^{(0)} = [-1,2 \quad 1,0]^T = \text{ponto base}.$$

As novas direcções são calculadas tomando  $s_1^{(1)}$  colinear com o vector  $[\text{ponto óptimo} - \text{ponto base}]^T$ .  $s_2^{(1)}$  será ortonormal a  $s_1^{(1)}$ .

Primeiro calculamos  $A_1^{(0)}$  e  $A_2^{(0)}$ ,

$$\mathbf{A}_1^{(0)} = [\mathbf{x}^{(2)} - \mathbf{x}^{(0)}]^T = [-1, 1 - (-1, 2) \quad 1, 1 - 1]^T = [0, 1 \quad 0, 1]^T$$

e

$$\mathbf{A}_2^{(0)} = [0 - \mathbf{A}_1^{(0)}]^T = [0 \quad 0, 1]^T,$$

para depois se calcular  $s_1^{(1)}$ ,

$$\mathbf{s}_1^{(1)} = \frac{[0, 1 \quad 0, 1]^T}{\sqrt{0,1^2 + 0,1^2}} = \frac{[0, 1 \quad 0, 1]^T}{\sqrt{0,02}} = \frac{[0, 1 \quad 0, 1]^T}{0,1\sqrt{2}} = \left[ \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right]^T.$$

Para determinarmos  $s_2^{(1)}$  necessitamos previamente de calcular  $B_2^{(0)}$  de acordo com,

$$\begin{aligned} \mathbf{B}_2^{(0)} &= \mathbf{A}_2^{(0)} - \left\{ [\mathbf{A}_2^{(0)}]^T \times [\mathbf{s}_1^{(1)}] \right\} \times \mathbf{s}_1^{(1)} \\ &= \begin{bmatrix} 0 \\ 0,1 \end{bmatrix} - \left\{ [0 \quad 0,1] \times \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \right\} \times \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0,1 \end{bmatrix} - \frac{0,1}{\sqrt{2}} \times \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0,1 \end{bmatrix} - \begin{bmatrix} 0,05 \\ 0,05 \end{bmatrix} \\ &= \begin{bmatrix} -0,05 \\ +0,05 \end{bmatrix} \end{aligned}$$

Assim,

$$\mathbf{S}_2^{(1)} = \frac{\mathbf{B}_2^{(0)}}{\|\mathbf{B}_2^{(0)}\|} = \frac{\begin{bmatrix} -0,05 \\ +0,05 \end{bmatrix}}{\sqrt{(-0,05)^2 + (0,05)^2}} = \frac{\begin{bmatrix} -0,05 \\ +0,05 \end{bmatrix}}{0,05\sqrt{2}} = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ +\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Dar-se-ia início agora à nova etapa ( $J = 1$ ) onde o novo ponto base seria o último ótimo, neste caso  $[-1,1 \ 1,1]^T$  e os novos acréscimos seriam,

$$\Delta_1^{(1)} = -\frac{1}{2} \times 0,3 = -0,15$$

e

$$\Delta_2^{(1)} = -\frac{1}{2} \times 0,3 = -0,15.$$

As etapas suceder-se-iam e terminariam de acordo com o critério de paragem estabelecido, neste caso, número máximo de iterações.

### 5.1.2. Aplicação à Resolução de Sistemas de Equações

Quando temos um sistema de equações,



$$f(x,y) = 0$$

$$g(x,y) = 0$$

podemos resolvê-lo substituindo-o por uma função a minimizar,

$$S(x,y) = [f(x,y)]^2 + [g(x,y)]^2.$$

De facto,  $S(x,y)$  tem o mínimo de zero apenas quando  $f = g = 0$ .

Aplicando este procedimento ao sistema extraído de Scheid (1991),

$$x = \text{sen}(x + y)$$

$$y = \text{cos}(x - y)$$

vem:

$$S(x,y) = f^2 + g^2 = [x - \text{sen}(x + y)]^2 + [y - \text{cos}(x - y)]^2$$

Por utilização do programa "MONROMINIMO" com a função na forma,

$$y = \{A(1) - \text{sen}[A(1) + A(2)]\}^2 + \{A(2) - \text{cos}[A(1) - A(2)]\}^2,$$

com os dados iniciais apresentados no Quadro 5.11 e com  $NUMA = 2$ ,  $NPER = 0$ ,  $KC = 40$  e  $MAXN = 500$  obtivemos os resultados que se

apresentam no Quadro 5.12. Comparando estes resultados com os resultados obtidos pelo autor,  $x = 0,934$  e  $y = 0,998$  com  $S(x,y) = 0,000002$  verifica-se que a soma dos quadrados por nós obtida está muito mais próxima de zero.

**Quadro 5.11** – Dados iniciais

Ordem	Parâmetro	Acréscimo	Limite Inferior	Limite Superior
I	A(I)	DDELTA(I)	CHECKL(I)	CHECKH(I)
1	0,5	0,1	-9,0	10,0
2	0,5	0,1	-9,0	10,0

**Quadro 5.12** – Resultados

$S(x,y)$	$x$	$y$
5,7739E-27	0,9351	0,9980

## 5.2. Ajustamento de uma Função Não Linear a Dados Experimentais

Os programas que vamos apresentar consistem fundamentalmente em minimizar uma função objectivo (FO), que aqui é

a soma do quadrado do erro, diferença entre valores observados da variável dependente ( $Y$ ) e os valores calculados pela função ( $\hat{Y}$ ),

$$FO = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, i = 1, 2, \dots, n .$$

### 5.2.1. Caso de uma Variável Independente

Para exemplificar o ajustamento de uma função a uma variável a dados experimentais, escolhemos a função (Boratto, 1984),

$$Y = A + Be^{\left(\frac{-X}{C}\right)},$$

onde  $A$ ,  $B$  e  $C$  são parâmetros a determinar. Para tal dispomos dos pares de pontos apresentados no quadro 5.13.

Escolhemos  $A = 5$ ,  $B = 10$  e  $C = 500$  como valores iniciais dos parâmetros.

### Método dos Mínimos Quadrados Não Lineares

Utilizou-se o programa "BORATO\_1\_VARIAVEL" que é uma modificação do programa original desenvolvido por Boratto (1984).

Neste programa a função aparece na forma,

$$S2 = T(1) + T(2) * \exp\left(-\frac{X(J)}{T(3)}\right),$$

onde T(1), T(2) e T(3) correspondem respectivamente a A, B e C e X(J), com  $J \leq N1$ , onde N1 é o número de pares de pontos, se refere aos diferentes valores que a variável independente pode tomar.

**Quadro 5.13** – Pontos experimentais

I	X	Y
1	30,8	23,3
2	92,3	19,1
3	153,8	16,0
4	215,3	13,4
5	276,8	11,3
6	369,1	9,0
7	492,1	7,0
8	615,1	5,5
9	738,2	4,6
10	861,2	4,0
11	984,2	3,5
12	1107,3	3,1

Os dados necessários para correr o programa são:

M - Número de parâmetros a otimizar;

N1 – Número de pares de pontos experimentais;

$X(J \leq N1)$  - Valor da variável independente. Neste caso (12 pares de pontos),  $X(1), \dots, X(12)$ ;

$X(J > N1)$  - Valor da variável dependente. Neste caso  $X(13), \dots, X(24)$ ;

M1 – Número de iterações;

T(I) - Valores iniciais dos parâmetros, com  $I = 1, \dots, M$ ;

C(I) - Valores iniciais dos acréscimos, com  $I = 1, \dots, M$ ;

Na Figura 5.8 mostra-se parte da subrotina do programa “BORATO\_1\_VARIAVEL” onde é possível alterar os dados iniciais.

Na Figura 5.9 mostra-se a subrotina *FUNCAO* onde se deve introduzir a função cujos parâmetros pretendemos ajustar.

Partindo dos dados iniciais apresentados no Quadro 5.14 e para avaliar a influência do número de iterações (M1), ensaiámos os três casos apresentados no Quadro 5.14.

No Quadro 5.15 apresentam-se os resultados obtidos para os diferentes valores de M1 considerados. A sequência detalhada dos resultados pode ser vista na janela “Immediate” em cada corrida do programa “BORATO\_1\_VARIAVEL” e no ficheiro de texto “BORATO\_1.TXT” criado pelo programa.

Pela análise dos resultados apresentados no quadro 5.15, pode verificar-se que à medida que o número de iterações aumenta o desvio quadrático médio diminui.

```

#####
'##          PROGRAMA BORATO_1_VARIABEL          ##
'## Programa para determinação dos parâmetros de uma função          ##
'## com uma variável independente pelo método dos Mínimos          ##
'## Quadrados Não Lineares          ##
#####

Private Sub Form_Load()
' Abertura do ficheiro para gravação dos resultados
Open "BORATO_1.TXT" For Output As #1
#####
'          DADOS A ALTERAR
' M - Número de parâmetros a otimizar
' N1 -Número de pares de pontos experimentais
' M1 - Número de iterações
' T(J) - Valores iniciais dos parâmetros
' C(J) - Valores iniciais dos acréscimos (deltas)
' X(J<=N1) - Variável independente (X)
' X(J>N1) - Variável dependente (Y)
' S - Função a otimizar, alterada na Subrotina FUNCAO
#####
M = 3
N1 = 12
M1 = 20
T(1) = 5
T(2) = 10
T(3) = 500
C(1) = 0.1
C(2) = 0.1
C(3) = 0.1
X(1) = 30.8: X(13) = 23.3
X(2) = 92.3: X(14) = 19.1
X(3) = 153.8: X(15) = 16#
X(4) = 215.3: X(16) = 13.4
X(5) = 276.8: X(17) = 11.3
X(6) = 369.1: X(18) = 9#
X(7) = 492.1: X(19) = 7#
X(8) = 615.1: X(20) = 5.5
X(9) = 738.2: X(21) = 4.6
X(10) = 861.2: X(22) = 4#
X(11) = 984.2: X(23) = 3.5
X(12) = 1107.3: X(24) = 3.1

```

**Figura 5.8** – Subrotina *Form Load* do Programa “BORATO\_1\_VARIABEL”.

```

Sub FUNCAO()
*****
'          ROTINA FUNCAO
*****
S1 = 0
K9 = N1 + 1
For J = 1 To N1
  S2 = 0
  S2 = T(1) + T(2) * Exp(-X(J) / T(3))
  T1(J) = S2
  F2(J) = S2 - X(K9)
  K9 = K9 + 1
  S1 = S1 + F2(J) * F2(J)
Next J
Exit Sub
End Sub

```

**Figura 5.9** – Subrotina *FUNCAO* do Programa “BORATO\_1\_VARIAVEL”.

**Quadro 5.14** – Dados iniciais e casos ensaiados

Ordem	Parâmetro	Acréscimo	M1
I	T(I)	C(I)	
1	5	0,1	5
2	10	0,1	10
3	500	0,1	20

**Quadro 5.15** – Resultados

M1	Desvio Quadrático Médio	A	B	C
5	0,6271	1,4848	22,4823	365,6831
10	0,1159	2,8341	22,6728	283,8210
20	0,1159	2,8342	22,6729	283,8159

A fim de avaliar a influência dos valores iniciais dos acréscimos,  $(C(I), I = 1, 2, 3)$  ensaiámos, para os três casos apresentados anteriormente, três valores diferentes para estes acréscimos conforme se apresenta no Quadro 5.16.

Analisando os resultados (Quadro 5.17), podemos verificar que maiores valores de M1 conduzem a melhores resultados em termos de desvio quadrático médio. Assim, tomando  $M1 = 20$ , o melhor resultado obtém-se com  $C(I) = 0,10$  ou com  $C(I) = 0,05$ . Com valores de  $C(I) = 0,01$  obtêm-se valores de desvio quadrático médio superiores.

**Quadro 5.16** – Dados iniciais e casos ensaiados

M1	Ordem	Parâmetro	Acréscimo
	I	T(I)	C(I)
5	1	5	0,1
	2	10	0,1
	3	500	0,1
10	1	5	0,01
	2	10	0,01
	3	500	0,01
20	1	5	0,05
	2	10	0,05
	3	500	0,05



**Quadro 5.17 – Resultados**

<b>C(I)</b>	<b>M1</b>	<b>Desvio Quadrático Médio</b>	<b>A</b>	<b>B</b>	<b>C</b>
	<b>5</b>	0,6271	1,4848	22,4823	365,6831
	<b>10</b>	0,1159	2,8341	22,6728	283,8210
<b>0,1</b>	<b>20</b>	0,1159	2,8342	22,6729	283,8159
	<b>5</b>	1,2589	-0,4312	22,6410	499,6110
<b>0,01</b>	<b>10</b>	1,1973	-0,5154	23,1463	486,6065
	<b>20</b>	1,1000	-0,1013	22,9658	461,3605
	<b>5</b>	1,0503	0,1035	22,8663	449,3741
	<b>10</b>	0,7451	1,1387	22,5684	386,2584
<b>0,05</b>	<b>20</b>	0,1159	2,8342	22,6729	283,81824

## Método de Monro

Tal como já referimos anteriormente, os ajustamentos que se seguem consistem em minimizar a função objectivo soma do quadrado do erro, diferença entre valores observados da variável dependente ( $Y$ ) e os valores calculados pela função ( $\hat{Y}$ ). Aqui, o método utilizado na minimização foi o método de Monro. Para tal utilizou-se o programa “monro\_1\_variavel”.

Embora o objectivo visado não seja o aperfeiçoamento de técnicas de programação, mas sim a sua interacção com a optimização numérica, achámos interessante mostrar como com boas técnicas de

programação se podem tornar muito mais aliciantes os programas a utilizar. Escolhemos um caso para demonstrar isto, deixando ao leitor, com melhores bases de programação, a modificação dos outros programas. Daí que o programa que se segue apresente um modo de programação distinto de todos os referidos neste texto.

No programa “monro\_1\_variavel” todos os dados necessários são introduzidos na *Form*, conforme se mostra na Figura 5.10.

Valores iniciais dos acréscimos

Pontos experimentais

Função a ajustar

NUMA

NPER

MAXN

KC

M

Limites superior e inferior dos parâmetros

Valores iniciais dos parâmetros

Função com uma variável independente (método de Hooke e Jeeves - modificado por Monro)

Ficheiro Ajuda

ATENÇÃO: O Separador decimal é a vírgula

Escreva a função:

f= [a(1)+a(2)\*exp(-x(i)/a(3))] sdif= [f-v(i)]^2

	Lim. Inf.	a	Lim. Sup.	Acréscimo	x	y
1	-5	5	1000	0,1		
2	-5	10	1000	0,1		
3	-5	500	1000	0,1		
4						
5						
6						
7						
8						
9						
10						
11						
12						

Outros Dados

NUMA 3 NPER 0

KC 100 Maxn 500

M 12

Opções de Exportação

Texto

Excel

Word

Iniciar

Resultado:

Critério: Parâmetros:

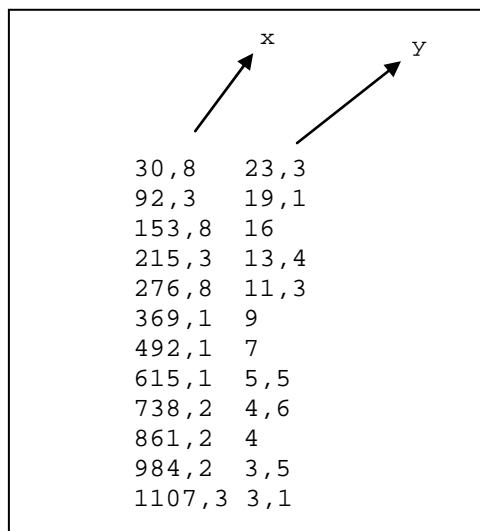
**Figura 5.10** - *Form* do Programa “monro\_1\_variavel”.

Os elementos a fornecer para correr o programa são:

f - Função a ajustar, que neste caso aparece do seguinte modo,

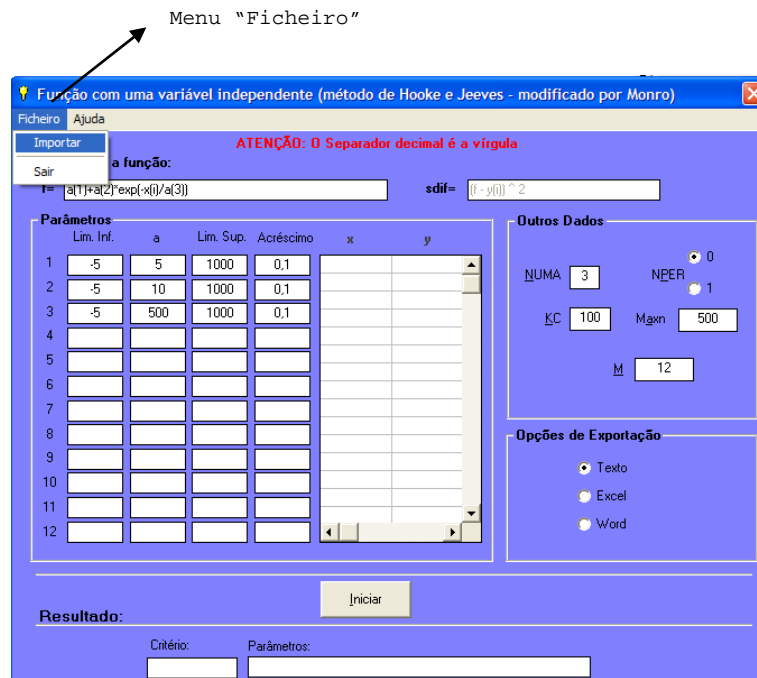
$$f = a(1) + a(2) * \exp\left(-\frac{X(i)}{a(3)}\right),$$

onde  $a(1)$ ,  $a(2)$  e  $a(3)$  correspondem a  $A$ ,  $B$  e  $C$  respectivamente e  $X(i)$  se refere aos diferentes valores que a variável independente pode tomar.  $X(i), Y(i)$  - Pares de pontos experimentais que são lidos a partir de um ficheiro em formato "TXT". Neste ficheiro, os dados estão dispostos em colunas separadas por espaços como se mostra na Figura 5.11. Para que o programa leia o ficheiro de dados é necessário aceder ao *Menu Ficheiro* e dentro deste ao *Menu Importar* (Figura 5.12) e seleccionar o ficheiro que contem os dados.



30,8	23,3
92,3	19,1
153,8	16
215,3	13,4
276,8	11,3
369,1	9
492,1	7
615,1	5,5
738,2	4,6
861,2	4
984,2	3,5
1107,3	3,1

**Figura 5.11** - Ficheiro de dados.



**Figura 5.12** – Leitura dos pares de pontos experimentais.

$a(1)$ ,  $a(2)$  e  $a(3)$  - Valores iniciais para os parâmetros.

$DDELTA(1)$ ,  $DDELTA(2)$  e  $DDELTA(3)$  - Valores iniciais dos acréscimos.

$CHECKL(1)$ ,  $CHECKL(2)$  e  $CHECKL(3)$  – Limite inferior dos parâmetros.

$CHECKH(1)$ ,  $CHECKH(2)$  e  $CHECKH(3)$  – Limite superior dos parâmetros.

NUMA – Número de parâmetros a serem otimizados.

NPER = 1 se os acréscimos são tomados em percentagem.

NPER = 0 se os acréscimos são tomados em valor absoluto.

KC - Número máximo de vezes que os acréscimos podem ser reduzidos para metade;

MAXN - Número máximo de vezes que o programa pode chamar a otimização.

M – Número de pares de pontos experimentais.

Para correr o programa basta premir a caixa “Iniciar”. Como resultados o programa apresenta na *Form*, os parâmetros otimizados e o critério, isto é a função objectivo (Figura 5.13).

Para além disto, em cada corrida do programa, a sequência detalhada dos cálculos é apresentada na janela “Immediate”. O programa permite ainda a gravação de todos os cálculos num ficheiro. Este ficheiro pode ser gravado em três formatos diferentes: Texto, Word, Excel, conforme a escolha das opções de exportação como se mostra na Figura 5.13.

Partindo dos dados apresentados no Quadro 5.18 e para avaliar a influência do número de chamadas à optimização (MAXN), ensaiámos os três casos apresentados no Quadro 5.19.

Função Objectivo

Parâmetros otimizados

Opções para gravação do ficheiro de resultados

Início da corrida do programa

**Figura 5.13** – Apresentação dos Resultados.

**Quadro 5.18** – Dados iniciais

Ordem	Parâmetro	Acréscimo	Limite Inferior	Limite Superior
I	A(I)	DDELTA(I)	CHECKL(I)	CHECKH(I)
1	5	0,1	-5	1000
2	10	0,1	-5	1000
3	500	0,1	-5	1000

**Quadro 5.19** – Casos ensaiados

NUMA	NPER	KC	MAXN
3	0	100	500
3	0	100	2000
3	0	100	5000

No Quadro 5.20 apresentam-se os resultados obtidos para os diferentes valores de MAXN considerados. Pela análise dos resultados pode verificar-se que a partir de MAXN = 2000 os resultados são praticamente iguais.

**Quadro 5.20** – Resultados

MAXN	Soma do Quadrado dos Erros	A	B	C
500	0,1221	2,8125	22,6625	284,9750
2000	0,1210	2,8342	22,6729	283,8147
5000	0,1210	2,8342	22,6729	283,8147

A fim de avaliar a influência dos valores iniciais dos acréscimos, (DDELTA(I), I = 1, 2, 3) ensaiámos, para os três casos apresentados anteriormente, três valores diferentes para estes acréscimos conforme se apresenta no Quadro 5.21.

**Quadro 5.21** – Dados iniciais e casos ensaiados

Ordem	Parâmetro	Acréscimo	Limite Inferior	Limite Superior
I	A(I)	DDELTA(I)	CHECKL(I)	CHECKH(I)
1	5	0,10	-5	1000
2	10	0,10	-5	1000
3	500	0,10	-5	1000
1	5	0,01	-5	1000
2	10	0,01	-5	1000
3	500	0,01	-5	1000
1	5	0,05	-5	1000
2	10	0,05	-5	1000
3	500	0,05	-5	1000

No Quadro 5.22 apresentam-se os resultados obtidos. Analisando os resultados podemos verificar que com MAXN = 500 os resultados são bastante influenciados pelos valores iniciais dos acréscimos. Neste caso o melhor resultado é obtido com DDELTA(I) = 0,1. Para valores mais elevados de MAXN, os resultados praticamente não são influenciados pelos DDELTA(I).

**Quadro 5.22** – Resultados

MAXN	DDELTA(I)	Soma do Quadrado dos Erros	A	B	C
500	0,10	0,1221	2,8125	22,6625	284,9750
	0,01	12,0127	-0,3400	23,0900	475,4100
	0,05	4,6762	1,2500	22,6000	381,5000
2000	0,10	0,1210	2,8342	22,6729	283,8147
	0,01	0,1210	2,8345	22,6729	283,8036
	0,05	0,1210	2,8342	22,6729	283,8150
5000	0,10	0,1210	2,8342	22,6729	283,8147
	0,01	0,1210	2,8342	22,6729	283,8147
	0,05	0,1210	2,8342	22,6729	283,8147

## Método de Nelder e Mead

Utilizou-se o programa “SIMPLEX\_1\_VARIAVEL” onde é necessário escrever a função na subrotina *SUMR*. Na Figura 5.14 mostra-se esta subrotina onde a função aparece na forma,

$$F = Z(1) + Z(2) * \text{Exp} \left[ \frac{-Y1(I)}{Z(3)} \right].$$

Neste caso Z(1), Z(2) e Z(3) correspondem respectivamente a A, B e C e Y1(I) refere-se aos diferentes valores que a variável



independente pode tomar, com  $I = 1, 2, \dots, NP$ , onde NP é o número de pontos experimentais.

Os dados que o programa necessita são:

$Y1(I)$  e  $Y(I)$  – Valores das variáveis independente e dependente, respectivamente;

NP - Número de pontos experimentais;

NX – Número de parâmetros a otimizar;

$X(I)$  - Valores iniciais dos parâmetros, com  $I = 1, 2, 3$ ;

STEPE - Passo inicial;

E - Constante arbitrariamente pequena para critério de paragem.

```
Public Sub Sumr(ByVal NN As Integer, Y(), Y1(), Z())
*****
'
      ROTINA SUMR
*****
C = C + 1
SU = 0
*** NA FUNÇÃO OS PARÂMETROS X(I) APARECEM COMO Z(I) ***
For I = 1 To NP
  F = Z(1) + Z(2) * Exp(-Y1(I) / Z(3))
  SDIF = (F - Y(I)) ^ 2
  SU = SU + SDIF
Next I
SUM(NN) = SU
**** Imprime na janela IMMEDIATE ****
Debug.Print "C="; C
**** Imprime no ficheiro SIMPLEX_1_VARIAVEL.TXT ****
Print #1, "C="; C
End Sub
```

**Figura 5.14** – Subrotina *SUMR* do Programa “SIMPLEX\_1\_VARIAVEL”.

Na Figura 5.15 mostra-se parte da subrotina do programa “SIMPLEX\_1\_VARIAVEL” onde é possível alterar os dados iniciais.

```

#####
##          PROGRAMA SIMPLEX_1_VARIAVEL          ##
## Programa para determinação dos parâmetros de uma função com ##
## uma variável independente pelo método de Nelder e Mead      ##
#####

Private Sub Form_Load()
Dim X1(50, 50), X(50)
Dim Y(50), Y1(50)
' Abertura do ficheiro para gravação dos resultados
Open "SIMPLEX_1_VARIAVEL.TXT" For Output As #1
#####
'          DADOS A ALTERAR
' Y(I), Y1(I) - Variável dependente e variável independente
' NP - Nº de pontos experimentais
' NX - Nº de parâmetros a otimizar
' STEPE - Passo inicial
' X(I) - Valores iniciais dos parâmetros
' E - Constante arbitrariamente pequena para critério de paragem
' F - Função a otimizar, alterada na subrotina SUMR
#####
Y(1) = 23.3: Y1(1) = 30.8
Y(2) = 19.1: Y1(2) = 92.3
Y(3) = 16.: Y1(3) = 153.8
Y(4) = 13.4: Y1(4) = 215.3
Y(5) = 11.3: Y1(5) = 276.8
Y(6) = 9: Y1(6) = 369.1
Y(7) = 7: Y1(7) = 492.1
Y(8) = 5.5: Y1(8) = 615.1
Y(9) = 4.6: Y1(9) = 738.2
Y(10) = 4: Y1(10) = 861.2
Y(11) = 3.5: Y1(11) = 984.2
Y(12) = 3.1: Y1(12) = 1107.3
NP = 12
NX = 3
STEPE = 0.1
X(1) = 5: X(2) = 10: X(3) = 500
E = 0.0000001
C = 0
C1 = 0
ALFA = 1
BETA = 0.5
GAMA = 2
DIFER = 0
XNX = NX
INA = 1
Call Sumr(INA, Y(), Y1(), X())

```

**Figura 5.15** – Subrotina *Form Load* do Programa “SIMPLEX\_1\_VARIAVEL”.

Partindo de  $X(1) = 5$ ,  $X(2) = 10$  e  $X(3) = 500$  como valores iniciais para os parâmetros e com o objectivo de avaliar a influência do STEPE, ensaiámos os três casos apresentados no Quadro 5.23.

No Quadro 5.24 apresentam-se os resultados obtidos. A sequência detalhada dos resultados pode ser vista na janela “Immediate” em cada corrida do programa “SIMPLEX\_1\_VARIABEL” ou no ficheiro de texto “SIMPLEX\_1\_VARIABEL.TXT” criado pelo programa.

**Quadro 5.23** – Casos ensaiados

<b>NX</b>	<b>E</b>	<b>STEPE</b>
3	1E-07	0,01
3	1E-07	0,05
3	1E-07	0,10

**Quadro 5.24** – Resultados

<b>STEPE</b>	<b>Soma do Quadrado dos Erros</b>	<b>A</b>	<b>B</b>	<b>C</b>
0,01	0,1210	2,8342	22,6732	283,8145
0,05	0,1210	2,8341	22,6729	283,8272
0,10	0,1210	2,8344	22,6731	283,8039

## Método de Rosenbrock

Neste caso utilizámos o programa "ROSENBROCK\_1\_VARIAVEL" sendo necessário escrever a função na subrotina FUNCAO. Na Figura 5.16 mostra-se esta subrotina onde esta função aparece na forma,

$$YEST(I) = X(1) + X(2) * \text{Exp} \left[ \frac{-X1(I)}{X(3)} \right],$$

onde  $X(1)$ ,  $X(2)$  e  $X(3)$  substituem os parâmetros A, B e C, respectivamente e  $X1(I)$ , com  $I = 1, 2, \dots, PARES$ , onde PARES é o número de pontos experimentais, se refere aos diferentes valores que a variável independente pode tomar.

Os dados que o programa necessita são:

PARES - Número de pares de pontos experimentais;

$X1(I)$  e  $Y1(I)$  – Valores das variáveis independente e dependente, respectivamente;

NK - Número de iterações;

NP - Número total de parâmetros;

NOPT - Número de parâmetros a otimizar;

$X(I)$  - Valores iniciais dos parâmetros, com  $I = 1, 2, 3$ ;

$XI(I)$  - Limite inferior dos parâmetros, com  $I = 1, 2, 3$ ;

$XS(I)$  - Limite superior dos parâmetros, com  $I = 1, 2, 3$ .

```
Sub FUNCAO()
*****
'          ROTINA FUNÇÃO
*****
ITERC = K5
For I = 1 To NOPT
  IK = IOPT(I)
  X(IK) = Y(I)
Next I
F1 = 0
For I = 1 To PARES
  YEST(I) = X(1) + X(2) * Exp(-(X1(I) / X(3)))
  F1 = F1 + (Y1(I) - YEST(I)) ^ 2
Next I
For I = 1 To NOPT
  IK = IOPT(I)
  Y(I) = X(IK)
Next I
End Sub
```

**Figura 5.16** – Subrotina *FUNCAO* do Programa “ROSENBROCK\_1\_VARIAVEL”.

Na Figura 5.17 mostra-se parte da subrotina do programa “ROSENBROCK\_1\_VARIAVEL” onde é possível alterar os dados iniciais.

Partindo de  $X(1) = 5$ ,  $X(2) = 10$  e  $X(3) = 500$  como valores iniciais para os parâmetros e com o objectivo de avaliar a influência do número de iterações (NK), ensaiámos diversos valores de NK. Os resultados obtidos estão apresentados no Quadro 5.25 onde podemos verificar que com  $NK = 500$  já se obtêm resultados bastante aceitáveis e que a partir de  $NK = 1000$  os resultados permanecem inalteráveis.

```

#####
##          PROGRAMA ROSENBROCK_1_VARIAVEL          ##
## Programa para determinação dos parâmetros de uma função      ##
## com um variável independente pelo método de Rosenbrock      ##
#####

Private Sub Form_Load()
' Abertura do ficheiro para gravação dos resultados
Open "ROSENBROCK_1.TXT" For Output As #1
#####
'
'          DADOS A ALTERAR
' PARES - Número de pares de pontos experimentais
' Y(I), X1(I) - Variável dependente e variável independente
' NK - Nº de iterações
' NP - Nº total de parâmetros
' NOPT - Nº de parâmetros a otimizar
' X(I) - Valores iniciais dos parâmetros
' XI(I) - Limite inferior dos parâmetros
' XS(I) - Limite superior dos parâmetros
' YEST - Função a otimizar, alterada na subrotina FUNCAO
#####
PARES = 12
X1(1) = 30.8: Y1(1) = 23.3
X1(2) = 92.3: Y1(2) = 19.1
X1(3) = 153.8: Y1(3) = 16
X1(4) = 215.3: Y1(4) = 13.4
X1(5) = 276.8: Y1(5) = 11.3
X1(6) = 369.1: Y1(6) = 9
X1(7) = 492.1: Y1(7) = 7
X1(8) = 615.1: Y1(8) = 5.5
X1(9) = 738.2: Y1(9) = 4.6
X1(10) = 861.2: Y1(10) = 4
X1(11) = 984.2: Y1(11) = 3.5
X1(12) = 1107.3: Y1(12) = 3.1
NK = 500
NP = 3
NOPT = 3
X(1) = 5
X(2) = 10
X(3) = 500
XI(1) = -1000
XI(2) = -1000
XI(3) = -1000
XS(1) = 1000
XS(2) = 1000
XS(3) = 1000

```

**Figura 5.17** - Subrotina *Form Load* do Programa "ROSENBROCK\_1\_VARIAVEL".

**Quadro 5.25 – Resultados**

NK	Soma do			
	Quadrado dos Erros	A	B	C
500	0,1210	2,8338	22,6729	283,8396
1000	0,1210	2,8343	22,6729	283,8147
2500	0,1210	2,8343	22,6729	283,8147
10000	0,1210	2,8343	22,67289	283,8147

### 5.2.2. Caso de duas ou mais Variáveis Independentes

A função não linear com que vamos exemplificar os diferentes métodos é a função a duas variáveis,

$$F(X, Y) = e^{\left( -AX_1 e^{-\frac{B}{X_2}} \right)},$$

extraída de Bard (1974), onde A e B são parâmetros a determinar. Para tal dispomos dos pares de pontos apresentados no quadro 5.26.

Escolhemos  $A = 750$  e  $B = 1200$  como valores iniciais dos parâmetros.

**Quadro 5.26** – Pontos experimentais

I	$X_1$	$X_2$	Y
1	0,10	100	0,9800
2	0,20	100	0,9830
3	0,30	100	0,9550
4	0,40	100	0,9790
5	0,50	100	0,9930
6	0,05	200	0,6260
7	0,10	200	0,5440
8	0,15	200	0,4550
9	0,20	200	0,2256
10	0,25	200	0,1670
11	0,02	300	0,5660
12	0,04	300	0,3170
13	0,06	300	0,0340
14	0,08	300	0,0160
15	0,10	300	0,0660

### Método dos Mínimos Quadrados Não Lineares

O programa que vamos utilizar é uma modificação do programa “BORATO\_1\_VARIAVEL” descrito em 5.2.1. Este programa para duas variáveis foi designado por “BORATO\_2\_VARIAVEL”.



Neste programa a função aparece na forma,

$$S2 = \text{Exp} \left[ -T(1) * X(J) * \text{Exp} \left( -\frac{T(2)}{X(J+N1)} \right) \right],$$

onde T(1) e T(2) correspondem respectivamente aos parâmetros A e B, X(J) corresponde aos valores da variável independente  $X_1$  e X(J+N1) aos valores da variável independente  $X_2$ , com  $J = 1, 2, \dots, N1$  sendo N1 o número de pares de pontos experimentais.

Os dados necessários para correr o programa são:

M - Número de parâmetros a otimizar;

N1 – Número de pares de pontos experimentais;

$X(J \leq N1)$  - Valor da variável independente  $X_1$ . Neste caso (15 pares de pontos),  $X(1), \dots, X(15)$ ;

$X(N1 > J > N1 * 2)$  - Valor da variável independente  $x_2$ . Neste caso  $X(16), \dots, X(30)$ ;

$X(J > N1 * 2)$  - Valor da variável dependente. Neste caso  $X(31), \dots, X(45)$ ;

M1 – Número de iterações;

T(I) - Valores iniciais dos parâmetros, com  $I = 1, \dots, M$ ;

C(I) - Valores iniciais dos acréscimos, com  $I = 1, \dots, M$ ;

Na Figura 5.18 mostra-se parte da subrotina *Form Load*, do programa “BORATO\_2\_VARIAVEL”, onde é possível alterar os dados

iniciais e na Figura 5.19 mostra-se a subrotina *FUNCAO*, onde se deve introduzir a função cujos parâmetros pretendemos ajustar.

```
#####
##          PROGRAMA BORATO_2_VARIAVEL          ##
## Programa para determinação dos parâmetros de uma função com      ##
## duas ou mais variáveis independentes pelo método dos Mínimos    ##
## Quadrados Não Lineares                                           ##
#####

Private Sub Form_Load()
' Abertura do ficheiro para gravação dos resultados
Open "BORATO_2.TXT" For Output As #1
#####
'          DADOS A ALTERAR
' M - Número de parâmetros a otimizar
' N1 -Número de pares de pontos experimentais
' M1 - Número de iterações
' T(J) - Valores iniciais dos parâmetros
' C(J) - Valores iniciais dos acréscimos (deltas)
' X(J<=N1) - Variável independente (X1)
' X(N1<J>=N1*2) - Variável independente (X2)
' X(J>N1*2) - Variável dependente (Y)
' S - Função a otimizar, alterada na subrotina FUNCAO
#####
M = 2
N1 = 15
M1 = 20
T(1) = 750
T(2) = 1200
C(1) = 0.1
C(2) = 0.1
X(1) = 0.1: X(16) = 100: X(31) = 0.98
X(2) = 0.2: X(17) = 100: X(32) = 0.983
X(3) = 0.3: X(18) = 100: X(33) = 0.955
X(4) = 0.4: X(19) = 100: X(34) = 0.979
X(5) = 0.5: X(20) = 100: X(35) = 0.993
X(6) = 0.05: X(21) = 200: X(36) = 0.626
X(7) = 0.1: X(22) = 200: X(37) = 0.544
X(8) = 0.15: X(23) = 200: X(38) = 0.455
X(9) = 0.2: X(24) = 200: X(39) = 0.2256
X(10) = 0.25: X(25) = 200: X(40) = 0.167
X(11) = 0.02: X(26) = 300: X(41) = 0.566
X(12) = 0.04: X(27) = 300: X(42) = 0.317
X(13) = 0.06: X(28) = 300: X(43) = 0.034
X(14) = 0.08: X(29) = 300: X(44) = 0.016
X(15) = 0.1: X(30) = 300: X(45) = 0.066
```

**Figura 5.18** – Subrotina *Form Load* do Programa “BORATO\_2\_VARIAVEL”.

```
Sub FUNCAO()  
*****  
'          ROTINA FUNCAO  
*****  
S1 = 0  
K9 = 2 * N1 + 1  
For J = 1 To N1  
  S2 = 0  
  S2 = Exp(-T(1) * X(J) * Exp(-T(2) / X(J + N1)))  
  T1(J) = S2  
  F2(J) = S2 - X(K9)  
  K9 = K9 + 1  
  S1 = S1 + F2(J) * F2(J)  
Next J  
Exit Sub  
End Sub
```

**Figura 5.19** – Subrotina *FUNCAO* do Programa “BORATO\_2\_VARIAVEL”.

Partindo dos dados iniciais apresentados no Quadro 5.27 e para avaliar a influência do número de iterações ensaiámos três casos,  $M1 = 5$ ,  $M1 = 10$ ,  $M1 = 20$ .

No Quadro 5.28 apresentam-se os resultados obtidos para os diferentes valores de  $M1$  considerados. A sequência detalhada dos resultados pode ser vista na janela “Immediate” em cada corrida do programa “BORATO\_2\_VARIAVEL” e no ficheiro de texto “BORATO\_2.TXT” criado pelo programa.

Pela análise dos resultados apresentados no quadro 5.28, pode verificar-se que à medida que o número de iterações aumenta o desvio quadrático médio diminui e, neste caso, com  $M1 = 20$  já se obtêm resultados satisfatórios.

A fim de avaliar a influência dos valores iniciais dos acréscimos,  $(C(I), I = 1, 2, 3)$  ensaiámos, para os três casos apresentados anteriormente, três valores diferentes para estes acréscimos conforme se apresenta no Quadro 5.29.

**Quadro 5.27** – Dados iniciais e casos ensaiados

<b>Ordem</b>	<b>Parâmetro</b>	<b>Acréscimo</b>
<b>I</b>	<b>T(I)</b>	<b>C(I)</b>
1	750	0,1
2	1200	0,1

**Quadro 5.28** – Resultados

<b>M1</b>	<b>Desvio</b>	<b>A</b>	<b>B</b>
	<b>Quadrático Médio</b>		
<b>5</b>	0,13198869	794,0767	1057,3645
<b>10</b>	5,53034E-02	816,2350	961,6455
<b>20</b>	5,53033E-02	814,3442	961,1905

**Quadro 5.29** – Dados iniciais e casos ensaiados

<b>M1</b>	<b>Ordem</b>	<b>Parâmetro</b>	<b>Acréscimo</b>
	<b>I</b>	<b>T(I)</b>	<b>C(I)</b>
5	1	750	0,1
	2	1200	0,1
10	1	750	0,01
	2	1200	0,01
20	1	750	0,05
	2	1200	0,05

Analisando o Quadro 5.30, podemos verificar que maiores valores de M1 conduzem a melhores resultados em termos de desvio quadrático médio. Assim, tomando  $M1 = 20$ , o melhor resultado obtém-se com  $C(l) = 0,10$ . Com valores de  $C(l) = 0,01$  obtêm-se valores de desvio quadrático médio superiores.

**Quadro 5.30 – Resultados**

<b>C(l)</b>	<b>M1</b>	<b>Desvio Quadrático Médio</b>	<b>A</b>	<b>B</b>
	<b>5</b>	0,13198869	794,0767	1057,3645
	<b>10</b>	5,53034E-02	816,2350	961,6455
<b>0,1</b>	<b>20</b>	5,53033E-02	814,3442	961,1905
	<b>5</b>	0,2773	753,8817	1188,0024
<b>0,01</b>	<b>10</b>	0,2648	757,7270	1175,9934
	<b>20</b>	0,2391	765,3115	1151,9425
	<b>5</b>	0,2263	769,0071	1140,2235
<b>0,05</b>	<b>10</b>	0,1582	787,1910	1080,1712
	<b>20</b>	5,5316E-02	820,3127	963,7768

## Método de Monro

O mesmo exemplo foi resolvido por um programa adaptado de Monro (1971) e que designamos por “MONRO\_2\_VARIAVEL”. Aqui, a função aparece com a seguinte forma,

$$F = \text{Exp} \left[ -A(1) * X1(I) * \text{Exp} \left( -\frac{A(2)}{X2(I)} \right) \right],$$

onde, como se vê, os parâmetros  $A(1)$  e  $A(2)$  correspondem a A e B,  $X1(I)$  corresponde aos valores da variável independente  $X_1$  e  $X2(I)$  aos valores da variável independente  $X_2$ .

Na Figura 5.20 mostra-se a subrotina *Form Load* do programa “MONRO\_2\_VARIAVEL” onde é possível alterar os dados iniciais e a função a otimizar.

Os dados que o programa necessita são:

M – Número de pontos experimentais;

$X1(I)$  - Valor da variável independente  $X_1$ ,  $I = 1, 2, \dots, M$ ;

$X2(I)$  - Valor da variável independente  $X_2$ ,  $I = 1, 2, \dots, M$ ;

$Y(I)$  - Valor da variável dependente,  $I = 1, 2, \dots, M$ ;

NUMA – Número de parâmetros a serem otimizados;

NPER = 1 se os acréscimos são tomados em percentagem;

NPER = 0 se os acréscimos são tomados em valor absoluto;

KC - Número máximo de vezes que os acréscimos podem ser reduzidos para metade;

MAXN - Número máximo de vezes que o programa pode chamar a otimização;

$A(I)$  - Valores iniciais dos parâmetros, com  $I = 1, 2$ ;

DDELTA (I) - Valores iniciais dos acréscimos, com  $I = 1, 2$ ;

CHECKL(I) - Limite inferior dos parâmetros, com  $I = 1, 2$ ;

CHECKH(I) - Limite superior dos parâmetros, com  $I = 1, 2$ ;

```

#####
##          PROGRAMA MONRO_2_VARIAVEL          ##
## Programa para determinação dos parâmetros de uma função com ##
## duas ou mais variáveis independentes pelo método de Hooke ##
## e Jeeves modificado por Monro ##
#####

Private Sub Form_Load()
' Abertura do ficheiro para gravação dos resultados
Open "MONRO_2.TXT" For Output As #1
#####
'          DADOS A ALTERAR
' Y(I), X1(I), X2(I) - Variável dependente e variáveis independentes
' M - Número de pontos experimentais
' NUMA - Nº de parâmetros a otimizar
' NPER = 0 - Acréscimos são tomados em valor absoluto
' NPER = 1 - Acréscimos são tomados em percentagem
' KC - Nº máximo de vezes que os acréscimos podem ser reduzidos
' para metade
' MAXN - Nº máximo de vezes que o programa pode chamar a rotina OPT
' A(I) - Valores iniciais dos parâmetros
' DDELTA(I) - Valores iniciais dos acréscimos
' CHECKL(I) - Limite inferior dos parâmetros
' CHECKH(I) - Limite superior dos parâmetros
' F - Função a otimizar
#####
Y(1) = 0.98: X1(1) = 0.1: X2(1) = 100
Y(2) = 0.983: X1(2) = 0.2: X2(2) = 100
Y(3) = 0.955: X1(3) = 0.3: X2(3) = 100
Y(4) = 0.979: X1(4) = 0.4: X2(4) = 100
Y(5) = 0.993: X1(5) = 0.5: X2(5) = 100
Y(6) = 0.626: X1(6) = 0.05: X2(6) = 200
Y(7) = 0.544: X1(7) = 0.1: X2(7) = 200
Y(8) = 0.455: X1(8) = 0.15: X2(8) = 200
Y(9) = 0.2256: X1(9) = 0.2: X2(9) = 200
Y(10) = 0.167: X1(10) = 0.25: X2(10) = 200
Y(11) = 0.566: X1(11) = 0.02: X2(11) = 300
Y(12) = 0.317: X1(12) = 0.04: X2(12) = 300
Y(13) = 0.034: X1(13) = 0.06: X2(13) = 300
Y(14) = 0.016: X1(14) = 0.08: X2(14) = 300
Y(15) = 0.066: X1(15) = 0.1: X2(15) = 300
M = 15
NUMA = 2
NPER = 1
KC = 100
MAXN = 2000
A(1) = 750#
A(2) = 1200#
DDELTA(1) = 0.1
DDELTA(2) = 0.1
CHECKL(1) = -5

```

**Figura 5.20** – Subrotina *Form Load* do Programa “MONRO\_2\_VARIAVEL”.

```

CHECKL(2) = -5
CHECKH(1) = 2000
CHECKH(2) = 2000
NSTART = 0
8000:
SUM = 0
For I = 1 To M
  F = Exp(-A(1) * X1(I) * Exp(-A(2) / X2(I)))
  SDIF = (F - Y(I)) ^ 2
  SUM = SUM + SDIF
Next I
OPTIM = SUM
Call Opt
GoTo 8000
End
End Sub

```

**Figura 5.20 (Cont.)** – Subrotina *Form Load* do Programa “MONRO\_2\_VARIAVEL”.

Partindo dos dados apresentados no Quadro 5.31 e para avaliar a influência do número de chamadas à otimização (MAXN), ensaiámos os três casos apresentados no Quadro 5.32. No Quadro 5.33 apresentam-se os resultados obtidos. A sequência detalhada dos resultados pode ser vista na janela “Immediate” em cada corrida do programa “MONRO\_2\_VARIAVEL” ou no ficheiro de texto “MONRO\_2.TXT” criado pelo programa.

**Quadro 5.31** – Dados iniciais

Ordem	Parâmetro	Acréscimo	Limite Inferior	Limite Superior
I	A(I)	DDELTA(I)	CHECKL(I)	CHECKH(I)
1	750,0	0,1	-5	2000
2	1200,0	0,1	-5	2000



**Quadro 5.32 – Casos ensaiados**

<b>NUMA</b>	<b>NPER</b>	<b>KC</b>	<b>MAXN</b>
2	0	10	500
2	0	100	2000
2	0	100	5000

**Quadro 5.33 – Resultados**

<b>MAXN</b>	<b>Soma do Quadrado dos Erros</b>	<b>A</b>	<b>B</b>
<b>500</b>	0,0398	812,1773	960,4687
<b>2000</b>	0,0398	814,4526	961,2214
<b>5000</b>	0,0398	814,4526	961,2214

Analisando o Quadro 5.33 pensamos que com MAXN = 500 já se obtém um valor de da soma do quadrado dos erros bastante satisfatório.

A fim de avaliar a influência dos valores iniciais dos acréscimos, (DDELTA(I), I = 1, 2) ensaiámos, para os três casos apresentados anteriormente, três valores diferentes para estes acréscimos conforme se apresenta no Quadro 5.34.

No Quadro 5.35 apresentam-se os resultados obtidos.

**Quadro 5.34 – Casos ensaiados**

Ordem	Parâmetro	Acréscimo	Limite Inferior	Limite Superior
I	A(I)	DDELTA(I)	CHECKL(I)	CHECKH(I)
1	750,0	0,1	-5	2000
2	1200,0	0,1	-5	2000
1	750,0	0,01	-5	2000
2	1200,0	0,01	-5	2000
1	750,0	0,05	-5	2000
2	1200,0	0,05	-5	2000

**Quadro 5.35 – Resultados**

MAXN	DDELTA(I)	Soma do Quadrado dos Erros	A	B
	<b>0,10</b>	0,0398	812,1773	960,4687
<b>500</b>	<b>0,05</b>	0,0398	813,4572	960,8923
	<b>0,01</b>	0,0398	814,4609	961,2324
	<b>0,10</b>	0,0398	814,4526	961,2214
<b>2000</b>	<b>0,05</b>	0,0398	814,4526	961,2214
	<b>0,01</b>	0,0398	814,4525	961,2214
	<b>0,10</b>	0,0398	814,4526	961,2214
<b>5000</b>	<b>0,05</b>	0,0398	814,4526	961,2214
	<b>0,01</b>	0,0398	814,4525	961,2214

## Método de Nelder e Mead

Utilizou-se o programa “SIMPLEX\_2\_VARIAVEL” onde é necessário escrever a função na subrotina *SUMR*. Na Figura 5.21 mostra-se esta subrotina onde a função aparece na forma,

$$F = \text{Exp} \left[ -Z(1) * Y1(I) * \text{Exp} \left( -\frac{Z(2)}{Y2(I)} \right) \right].$$

Neste caso  $Z(1)$  e  $Z(2)$  correspondem respectivamente aos parâmetros  $A$  e  $B$ ,  $Y1(I)$ , com  $I = 1, 2, \dots$ , se refere aos diferentes valores que a variável independente pode tomar.

```
Public Sub Sumr(ByVal NN As Integer, Y(), Y1(), Y2(), Z())
*****
'          ROTINA SUMR
*****
C = C + 1
SU = 0
**** NA FUNÇÃO OS PARÂMETROS X(I) APARECEM COMO Z(I) ****
For I = 1 To NP
    F = Exp(-Z(1) * Y1(I) * Exp(-Z(2) / Y2(I)))
    SDIF = (F - Y(I)) ^ 2
    SU = SU + SDIF
Next I
SUM(NN) = SU
**** Imprime na janela IMEDIATE ****
Debug.Print "C="; C
**** Imprime no ficheiro SIMPLEX_2_VARIAVEL.TXT ****
Print #1, "C="; C
End Sub
```

**Figura 5.21** – Subrotina *SUMR* do Programa “SIMPLEX\_2\_VARIAVEL”.

Os dados que o programa necessita são:

Y1(I), Y2(I) e Y(I) – Valores das variáveis independentes e dependente, respectivamente;

NP - Número de pontos experimentais;

NX – Número de parâmetros a otimizar;

X(I) - Valores iniciais dos parâmetros, com  $I = 1, 2, 3$ ;

STEPE - Passo inicial;

E - Constante arbitrariamente pequena para critério de paragem.

Na Figura 5.22 mostra-se parte da subrotina do programa “SIMPLEX\_2\_VARIAVEL” onde é possível alterar estes dados.

Com o objectivo de avaliar a influência do STEPE, ensaiámos os três casos apresentados no Quadro 5.36.

No Quadro 5.37 apresentam-se os resultados obtidos. A sequência detalhada dos resultados pode ser vista na janela “Immediate” em cada corrida do programa “SIMPLEX\_2\_VARIAVEL” ou no ficheiro de texto “SIMPLEX\_2\_VARIAVEL.TXT” criado pelo programa.

**Quadro 5.36** – Casos ensaiados

NX	E	STEPE
2	1E-07	0,10
2	1E-07	0,05
2	1E-07	0,10

```

#####
'##          PROGRAMA SIMPLEX_2_VARIAVEL          ##
'## Programa para determinação dos parâmetros de uma função          ##
'## com duas ou mais variáveis independentes pelo método de          ##
'## Nelder e Mead          ##
#####

Private Sub Form_Load()
Dim X1(50, 50), X(50)
Dim Y(50), Y1(50), Y2(50)
' Abertura do ficheiro para gravação dos resultados
Open "SIMPLEX_2_VARIAVEL.TXT" For Output As #1
#####
'          DADOS A ALTERAR
' Y(l), Y1(l), Y2(l) - Variável dependente e variáveis independentes
' NP - Nº de pontos experimentais
' NX - Nº de parâmetros a otimizar
' STEPE = Passo inicial
' X(l) - Valores iniciais dos parâmetros
' E - Constante arbitrariamente pequena para critério de paragem
' F - Função a otimizar, alterada na subrotina SUMR
#####
Y(1) = 0.98: Y1(1) = 0.1: Y2(1) = 100
Y(2) = 0.983: Y1(2) = 0.2: Y2(2) = 100
Y(3) = 0.955: Y1(3) = 0.3: Y2(3) = 100
Y(4) = 0.979: Y1(4) = 0.4: Y2(4) = 100
Y(5) = 0.993: Y1(5) = 0.5: Y2(5) = 100
Y(6) = 0.626: Y1(6) = 0.05: Y2(6) = 200
Y(7) = 0.544: Y1(7) = 0.1: Y2(7) = 200
Y(8) = 0.455: Y1(8) = 0.15: Y2(8) = 200
Y(9) = 0.2256: Y1(9) = 0.2: Y2(9) = 200
Y(10) = 0.167: Y1(10) = 0.25: Y2(10) = 200
Y(11) = 0.566: Y1(11) = 0.02: Y2(11) = 300
Y(12) = 0.317: Y1(12) = 0.04: Y2(12) = 300
Y(13) = 0.034: Y1(13) = 0.06: Y2(13) = 300
Y(14) = 0.016: Y1(14) = 0.08: Y2(14) = 300
Y(15) = 0.066: Y1(15) = 0.1: Y2(15) = 300
NP = 15
NX = 2
STEPE = 0.1
X(1) = 750: X(2) = 1200
E = 0.0000001
C = 0
C1 = 0
ALFA = 1
BETA = 0.5
GAMA = 2
DIFER = 0
XNX = NX
INA = 1
Call Sumr(INA, Y(), Y1(), Y2(), X())

```

**Figura 5.22** – Subrotina *Form Load* do Programa “SIMPLEX\_2\_VARIAVEL”.

**Quadro 5.37** – Resultados

STEPE	Soma do Quadrado dos Erros	A	B
0,10	0,0398	814,1403	961,1818
0,05	0,0398	813,1642	960,8801
0,01	0,0402	909,0088	985,7898

Analisando o Quadro 5.37 podemos dizer que o menor valor de da soma do quadrado dos erros se obtém com STEPE = 0,10 ou STEPE = 0,05. Com STEPE = 0,01 obtém-se uma soma do quadrado dos erros ligeiramente superior e portanto pior.

## Método de Rosenbrock

Neste caso utilizámos o programa “ROSENBROCK\_2\_VARIAVEL” sendo necessário escrever a função na subrotina FUNCAO. Na Figura 5.23 mostra-se esta subrotina onde esta função aparece na forma,

$$YEST(I) = \text{Exp} \left[ -X(1) * X1(I) * \text{Exp} \left( -\frac{X(2)}{X2(I)} \right) \right],$$

onde  $X(1)$  e  $X(2)$  substituem os parâmetros A e B, respectivamente,  $X1(I)$  e  $X2(I)$ , com  $I = 1, 2, \dots$ , onde PARES é o número de pontos experimentais, se referem aos diferentes valores das duas variáveis independentes.

Os dados que o programa necessita são:

PARES - Número de pares de pontos experimentais;

X1(I), X2(I) e Y1(I) – Valores das duas variáveis independentes e da variável dependente, respectivamente;

NK - Número de iterações;

NP - Número total de parâmetros;

NOPT - Número de parâmetros a otimizar;

X(I) - Valores iniciais dos parâmetros, com  $I = 1, 2, 3$ ;

XI(I) - Limite inferior dos parâmetros, com  $I = 1, 2, 3$ ;

XS(I) - Limite superior dos parâmetros, com  $I = 1, 2, 3$ .

Na Figura 5.24 mostra-se parte da subrotina do programa “ROSENBROCK\_1\_VARIAVEL” onde é possível alterar os dados iniciais.

```

Sub FUNCAO()
*****
'          ROTINA FUNÇÃO
*****
ITERC = K5
For I = 1 To NOPT
  IK = IOPT(I)
  X(IK) = Y(I)
Next I
F1 = 0
For I = 1 To PARES
  YEST(I) = Exp(-X(1) * X1(I) * Exp(-(X(2) / X2(I))))
  F1 = F1 + (Y1(I) - YEST(I)) ^ 2
Next I
For I = 1 To NOPT
  IK = IOPT(I)
  Y(I) = X(IK)
Next I
End Sub

```

**Figura 5.23** – Subrotina *FUNCAO* do Programa “ROSENBROCK\_2\_VARIAVEL”.

```

#####
##          PROGRAMA ROSENBROCK_2_VARIAVEL          ##
## Programa para determinação dos parâmetros de uma função      ##
## com duas ou mais variáveis independentes pelo método de      ##
## Rosenbrock                                                  ##
#####

Private Sub Form_Load()
' Abertura do ficheiro para gravação dos resultados
Open "ROSENBROCK_2.TXT" For Output As #1
'#####
'          DADOS A ALTERAR
' PARES - Número de pares de pontos experimentais
' Y(l), X1(l), X2(l) - Variável dependente e variáveis independentes
' NK - Nº de iterações
' NP - Nº total de parâmetros
' NOPT - Nº de parâmetros a otimizar
' X(l) - Valores iniciais dos parâmetros
' Xl(l) - Limite inferior dos parâmetros
' Xs(l) - Limite superior dos parâmetros
' YEST - Função a otimizar, alterada na subrotina FUNCAO
'#####
PARES = 15
Y1(1) = 0.98: X1(1) = 0.1: X2(1) = 100
Y1(2) = 0.983: X1(2) = 0.2: X2(2) = 100
Y1(3) = 0.955: X1(3) = 0.3: X2(3) = 100
Y1(4) = 0.979: X1(4) = 0.4: X2(4) = 100
Y1(5) = 0.993: X1(5) = 0.5: X2(5) = 100
Y1(6) = 0.626: X1(6) = 0.05: X2(6) = 200
Y1(7) = 0.544: X1(7) = 0.1: X2(7) = 200
Y1(8) = 0.455: X1(8) = 0.15: X2(8) = 200
Y1(9) = 0.2256: X1(9) = 0.2: X2(9) = 200
Y1(10) = 0.167: X1(10) = 0.25: X2(10) = 200
Y1(11) = 0.566: X1(11) = 0.02: X2(11) = 300
Y1(12) = 0.317: X1(12) = 0.04: X2(12) = 300
Y1(13) = 0.034: X1(13) = 0.06: X2(13) = 300
Y1(14) = 0.016: X1(14) = 0.08: X2(14) = 300
Y1(15) = 0.066: X1(15) = 0.1: X2(15) = 300
NK = 5000
NP = 2
NOPT = 2
X(1) = 750
X(2) = 1200
Xl(1) = -1000
Xl(2) = -1000
Xs(1) = 1000
Xs(2) = 1000

```

**Figura 5.24** – Subrotina *Form Load* do Programa “ROSENBROCK\_2\_VARIAVEL”.



Com a finalidade de avaliar a influência do número de iterações, ensaiámos diversos valores de NK. Os resultados obtidos estão apresentados no Quadro 5.38.

Pela análise do Quadro 5.38 podemos verificar que com NK = 2000 já se obtêm resultados bastante aceitáveis e que a partir de NK = 1000 os resultados permanecem praticamente inalteráveis.

**Quadro 5.38** – Resultados

NK	Soma do		
	Quadrado dos Erros	A	B
500	0,03989	866,17895	975,21533
1000	0,03976	820,3656	962,90817
2000	0,03976	814,50627	961,23669
5000	0,03976	814,45277	961,22143

### 5.2.3. Exemplos de Índole Hidrológica

Para tornar mais objectiva a aplicação destes métodos, recorreremos a alguns exemplos no domínio da Hidrologia.

## Determinação da Curva Característica Humidade-Sucção

M. van Genuchten (1980) estabeleceu uma curva característica de humidade-sucção, em que os parâmetros  $\theta_r$ ,  $\theta_s$ ,  $a_1$ ,  $a_2$  e  $a_3$ , permitem tornar a curva bastante flexível adaptando-se aos diferentes tipos de solo. A curva é dada por,

$$\theta = \theta_r + \frac{(\theta_s - \theta_r)}{\left[1 + |a_1 \psi|^{a_2}\right]^{a_3}}$$

onde  $\theta$  é o teor de humidade do solo ( $L^3.L^{-3}$ ),  $\theta_s$  é o teor de humidade à saturação ( $L^3.L^{-3}$ ),  $\theta_r$  é o teor de humidade residual ( $L^3.L^{-3}$ ),  $\psi$  é a sucção (L),  $a_1$  ( $L^{-1}$ ) e  $a_2$  são parâmetros a determinar, já que segundo Mualem (1976)  $a_3 = 1 - \frac{1}{a_2}$ .

Enquanto que  $\theta_s$  é fácil de determinar experimentalmente, já o mesmo não se consegue com  $\theta_r$  que em muitos casos se torna um parâmetro mal definido. M. van Genuchten (1980) definiu-o como o teor de humidade para o qual o gradiente  $\frac{d\theta}{d\psi}$  se torna nulo, excluindo a região vizinha da saturação.

O parâmetro  $a_1$  está inversamente relacionado com o valor de entrada de ar e o parâmetro  $a_2$  está inversamente relacionado com a distribuição da largura dos poros.

Para a determinação dos parâmetros escolhemos o programa “BORATO\_1\_VARIAVEL” e utilizámos os dados do autor. Assim, considerámos  $\theta_s$  constante e igual a 0,5 e os 12 pares de valores de humidade ( $\theta$ ) e de sucção ( $\psi$ ) apresentados no Quadro 5.39.

**Quadro 5.39** – Valores de humidade ( $\theta$ ) e de sucção ( $\psi$ )

I	$\theta$ (cm <sup>3</sup> .cm <sup>-3</sup> )	$\psi$ (cm)
1	0,50	0,00
2	0,49	45,58
3	0,46	96,90
4	0,42	150,00
5	0,38	204,00
6	0,34	266,70
7	0,30	346,40
8	0,26	458,30
9	0,22	635,90
10	0,18	979,80
11	0,14	1990,00
12	0,11	7998,00

Para correr o programa é necessário introduzir a equação da curva na subrotina *FUNCAO*, que neste caso se escreve,

$$S2 = T(1) + \frac{(0,5 - T(1))}{\left[1 + |T(2) * X(J)|^{T(3)}\right]^{\left(1 - \frac{1}{T(3)}\right)}}$$

onde  $T(1)$ ,  $T(2)$  e  $T(3)$  correspondem respectivamente a  $\theta_r$ ,  $a_1$  e  $a_2$  e  $X(J)$ , com  $J \leq N1$  e  $N1$  é o número de pares  $(\theta, \psi)$ , se refere aos diferentes valores que a variável  $\psi$  pode tomar.

É necessário ainda introduzir, para além dos valores das variáveis  $\theta$  e  $\psi$ , os valores para  $M$ ,  $N1$ ,  $M1$ ,  $T(I)$  e  $C(I)$ , com  $I = 1, \dots, M$ . No Quadro 5.40 e 5.41 apresentam-se estes valores. No Quadro 5.42 apresentam-se os resultados.

**Quadro 5.40** – Valores de  $M$ ,  $N1$  e  $M1$

<b>M</b>	<b>N1</b>	<b>M1</b>
3	12	20

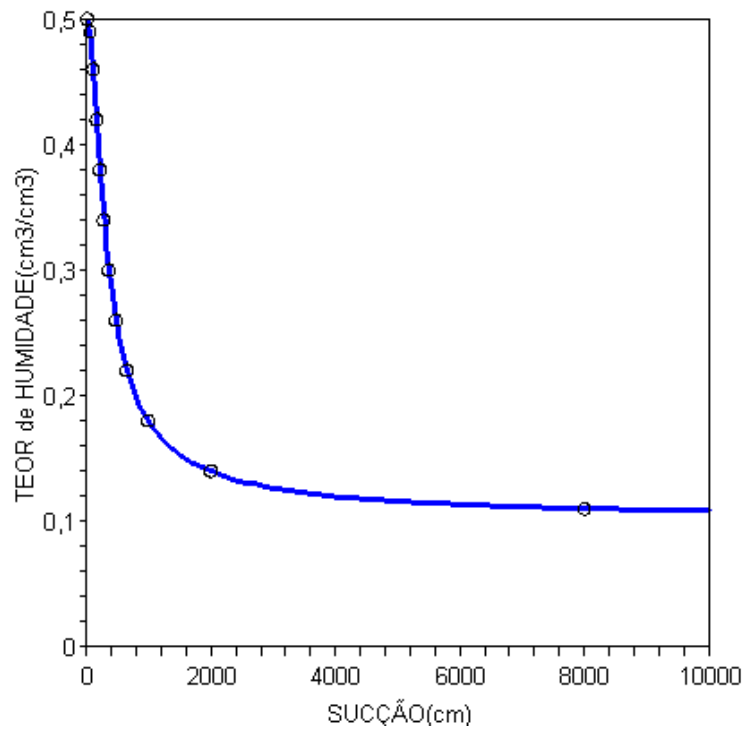
**Quadro 5.41** – Parâmetros e acréscimos iniciais

<b>Ordem</b>	<b>Parâmetro</b>	<b>Acréscimo</b>
<b>I</b>	<b>T(I)</b>	<b>C(I)</b>
1	0,08	0,005
2	0,01	0,001
3	3,00	0,1

**Quadro 5.42 – Resultados**

<b>Desvio Quadrático Médio</b>	$\theta_r$	$a_1$	$a_2$
1,517E-05	0,100	0,005	2,000

Na Figura 5.25 pode apreciar-se a boa aderência da curva aos dados experimentais.

**Figura 5.25 - Curva característica umidade-sucção.**

## Determinação de um Modelo Para o Caudal de Ponta de Cheia

Neste exemplo, extraído de Maccuen e Snyder (1986) consideram-se diferentes bacias hidrográficas do estado de Indiana para as quais se dispõem dos dados apresentados no Quadro 5.43.

Pretende obter-se o caudal de ponta através do modelo potência,

$$Y = a_1 X_1^{a_2} X_2^{a_3} X_3^{a_4} X_4^{a_5} X_5^{a_6} X_6^{a_7} X_7^{a_8},$$

onde  $Y$  é o caudal de ponta para um período de retorno de 10 anos ( $\text{ft}^3 \cdot \text{s}^{-1}$ ),  $X_1$  é a área da bacia ( $\text{mi}^2$ ),  $X_2$  é o declive da linha de água ( $\text{ft} \cdot \text{mi}^{-1}$ ),  $X_3$  é o comprimento da linha de água (mi),  $X_4$  é um índice de precipitação (in),  $X_5$  é a altura da bacia (ft),  $X_6$  é a densidade de drenagem ( $\text{mi} \cdot \text{mi}^{-2}$ ),  $X_7$  é o coeficiente de escoamento e  $a_1, a_2, a_3, a_4, a_5, a_6, a_7$  e  $a_8$  são os parâmetros a otimizar.

Utilizámos o programa “**MONRO\_2\_VARIAVEL**” onde esta função deve ser introduzida na seguinte forma,

$$F = A(1)X1(I)^{A(2)} X2(I)^{A(3)} X3(I)^{A(4)} X4(I)^{A(5)} X5(I)^{A(6)} X6(I)^{A(7)} X7(I)^{A(8)}$$

Para correr o programa é necessário fornecer os valores das variáveis dependente e independentes que constam no Quadro 5.43, os valores de M, NUMA, NPER, KC, MAXN, A(I), DDELTA (I), CHECKL(I) e CHECKH(I). Todos estes valores estão apresentados nos Quadros 5.44 e 5.45.

**Quadro 5.43 – Dados**

I	Y	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$
1	13300	121	12,8	19,5	11	341	9,5	0,8
2	12600	38,2	26	16,6	13	449	11	0,9
3	14100	188	5,5	25,1	15	265	10,5	0,9
4	13100	129	6,3	33,2	16	438	8,7	0,8
5	5500	41,9	15,4	14	17,5	435	12	0,8
6	3200	85,6	4,7	15,6	9,5	182	5,8	0,5
7	7700	133	4,6	20,1	10	150	7,5	0,8
8	600	113	3,6	22,7	7	95	3,4	0,3
9	460	35	5,5	9,6	8,5	77	2,2	0,3
10	2350	152	2	19,1	8	82	2,3	0,7
11	4100	146	3,3	24,1	9	100	2,8	0,8
12	4800	162	3,3	29,6	9	111	3	0,8
13	740	24,7	4,5	12,7	9,5	72	4,5	0,7
14	13100	132	7,2	35,9	12	252	5,4	0,7
15	16800	133	11,4	29,1	13	344	5,4	0,7
16	1580	35,5	10,2	12,5	10,5	165	5,7	0,7
17	3540	131	4	27,1	10	121	5,4	0,7
18	2400	40,4	6,2	15	9,5	110	4,6	0,7
19	3750	18,5	18,7	6,4	9,5	110	6,6	0,7
20	5800	169	7,2	31,8	11,5	273	5,3	0,7
21	1600	42,4	6,7	19,3	10	127	4,5	0,7
22	9000	103	15,2	17,4	10,5	153	7,5	0,7
23	11200	174	6,8	35,1	10,5	264	7,9	0,7
24	1820	23,9	18,8	9,2	11	189	4,5	0,7
25	3900	28,8	10,6	11,5	11	131	6	0,7
26	4150	14,6	19,8	7,6	13	291	11,5	0,8
27	9600	59	12,6	18,3	12,5	283	7	0,7
28	7800	184	5,8	30,8	12	251	5,2	0,7
29	8200	107	4,3	30,1	12	195	7	0,7

**Quadro 5.43 (Cont.) – Dados**

I	Y	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$
30	8200	91,4	10,3	33,7	12,5	401	7,4	0,7
31	14200	155	8,9	42,3	13,5	410	9,8	0,9
32	14000	77,2	9,4	32,6	14	318	1,5	1
33	13400	85,9	12,2	29,1	13,5	391	9	1
34	29000	198	9,2	43,2	14	448	9,1	1
35	57000	38,2	13	10,6	14	312	10	0,8
36	7400	76,1	11,6	17,7	13,5	400	8	0,7
37	11700	120	9	34,7	13,5	436	9,5	0,7
38	8400	48,8	19,1	13,7	14,5	383	7,5	0,8
39	6300	60,7	12,7	16,5	15	390	7	0,8
40	5600	171	2,6	52,8	15,5	466	13,7	0,8
41	2300	69,2	7,4	22,6	6,5	159	4	0,5
42	2700	125	3,6	29,8	7	131	3,2	0,5
43	2400	160	3,2	36	8	173	3,1	0,4
44	2080	62,9	6,2	14,8	12	262	8	0,4
45	1800	78,7	4,7	22,5	8	206	6,6	0,4
46	530	80,5	5,2	22,9	4,5	120	3,2	0,3
47	650	142	3,9	26,6	5,5	130	3,5	0,3
48	1270	87,3	8	20,1	4,5	193	5,1	0,5
49	640	116	1,2	23,2	5,5	100	3	0,3
50	1420	132	5	12,9	7	65	2,5	0,5
51	1150	123	3,2	22,2	7,5	137	2,5	0,4
52	1600	54,7	2,3	21,1	7	102	6,5	0,4
53	340	35,6	2,5	8,9	7,5	44	2,5	0,4
54	1220	144	2,9	15,7	8	58	3,5	0,4
55	1670	203	2,5	18,6	8	63	4,5	0,4
56	750	21,8	6,4	10,4	8,5	82	3,4	0,7
57	2100	83,7	2,2	13,2	8,5	95	2,5	0,7
58	1970	44,8	6,4	21,4	8,5	193	3	0,7



**Quadro 5.44** – Valores de M, NUMA, NPER, KC e MAXN

M	NUMA	NPER	KC	MAXN
58	8	1	1000	10000

**Quadro 5.45** – Parâmetros, acréscimos iniciais e limites dos parâmetros

Ordem	Parâmetro	Acréscimo	Limite Inferior	Limite Superior
I	A(I)	DDELTA (I)	CHECKL(I)	CHECKH(I)
1	1	0,05	-5	2000
2	1	0,05	-5	2000
3	1	0,05	-5	2000
4	1	0,05	-5	2000
5	1	0,05	-5	2000
6	1	0,05	-5	2000
7	1	0,05	-5	2000
8	1	0,05	-5	2000

No Quadro 5.46 apresentam-se os resultados obtidos.

**Quadro 5.46** – Resultados

Soma do Quadrado dos Erros	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$
162500464,06	2,6979	0,8401	0,8564	0,2602	0,7995	-0,0038	-0,0888	1,0850

Comparando estes resultados com os obtidos pelos autores verificamos que a soma do quadrado dos erros sofreu uma diminuição de 7,5%.

## Ajustamento de um Modelo de Semi-Variograma Esférico à Precipitação

Maccuen e Snyder (1986) referem os trabalhos de Karlinger e Skrivan (1981) para a precipitação anual, numa dada bacia, com base em 60 estações.

O modelo a ajustar para o semi-variograma (Webster e Burges, 1983) é do tipo,

$$\gamma(h) = \begin{cases} \gamma_n + (\gamma_r - \gamma_n) \left[ 1,5 \left( \frac{h}{r} \right) - 0,5 \left( \frac{h}{r} \right)^3 \right] & \text{se } h \leq r \\ \gamma_r & \text{se } h > r \end{cases},$$

onde  $\gamma_n$  é o chamado efeito de pepita,  $\gamma_r$  é o patamar,  $r$  é o raio de influência e  $h$  é a distância. Os valores de  $\gamma_n$ ,  $\gamma_r$  e  $r$  estimaram-se a partir dos dados apresentados no Quadro 5.47.

Para tal escolhemos o programa “**SIMPLEX\_1\_VARIAVEL**”, onde a função deve ser escrita na forma,

$$F = Z(1) + (Z(2) - Z(1)) * \left[ 1,5 \left( \frac{Y1(I)}{Z(3)} \right) - 0,5 \left( \frac{Y1(I)}{Z(3)} \right)^3 \right]$$

onde  $Z(1)$ ,  $Z(2)$  e  $Z(3)$  correspondem respectivamente a  $\gamma_n$ ,  $\gamma_r$  e  $r$  e  $Y1(I)$ , com  $I = 1, 2, \dots, NP$ , onde  $NP$  é o número de pares  $[h, \gamma(h)]$ , se refere aos diferentes valores que a variável independente  $h$  pode tomar.

**Quadro 5.47** – Dados

$I$	$h$ (mi)	$\gamma(h)$ (in <sup>2</sup> )
1	17,2	3,94
2	42,5	3,38
3	59,4	4,06
4	76,6	5,16
5	87,5	4,16
6	109,4	5,03
7	128,1	4,34
8	145,3	4,94
9	160,9	4,53
10	187,5	7,05
11	237,5	4,53

Para correr o programa devemos introduzir, para além dos dados das variáveis que constam no Quadro 5.47, valores para  $NP$ ,  $NX$ ,  $STEPE$ ,  $X(I)$  e  $E$ . Estes valores encontram-se no Quadro 5.48 e 5.49.

**Quadro 5.48** – Valores de NP, NX, STEPE, e E

NP	NX	STEPE	E
11	3	0,1	1E-07

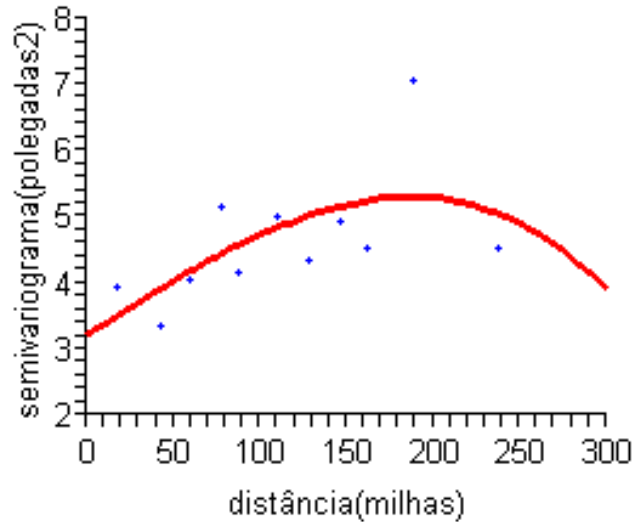
**Quadro 5.49** – Parâmetros iniciais

Ordem	Parâmetro
I	X(I)
1	3
2	2
3	200

No Quadro 5.50 apresentam-se os resultados obtidos e na Figura 5.26 mostra-se a curva ajustada, sendo a soma do quadrado dos erros ligeiramente inferior à obtida pelos autores citados.

**Quadro 5.50** – Resultados

Soma do			
Quadrado dos	$\gamma_n$	$\gamma_r$	$r$
Erros			
5,5909	3,21198	5,2914	186,9498



**Figura 5.26** – Semi-Variograma para a precipitação anual.

### **Curva de Variação do Índice de Erosividade em Função do Tempo**

Aplicámos o método de Rosenbrock ao ajustamento de uma curva logística para modelar o índice de erosividade em função do tempo, à escala mensal. Assim e de acordo com Maccuen e Snyder (1986), pretende-se determinar, com base nos dados apresentados no Quadro 5.51, os parâmetros de equação,

$$Y(t) = \frac{k_1}{1 + e^{-k_2(t-t_0)}},$$

sendo  $k_1$  e  $k_2$  parâmetros de escala e  $t_0$  um parâmetro de localização.

**Quadro 5.51** – Dados

I	t	Y(t)
1	0,0	0
2	3,0	5
3	4,0	15
4	4,5	22
5	5,0	31
6	5,5	55
7	6,0	80
8	6,5	90
9	7,0	105
10	8,0	130
11	9,0	147
12	10,0	158
13	12,0	160

Para a determinação dos parâmetros utilizamos o programa “ROSENBROCK\_1\_VARIAVEL” onde é necessário escrever a equação na subrotina *FUNCAO*, que neste caso toma a forma,

$$YEST(I) = \frac{X(1)}{1 + e^{-X(2) * [X(1) - X(3)]}}$$

onde  $X(1)$ ,  $X(2)$  e  $X(3)$  correspondem respectivamente a  $k_1$ ,  $k_2$  e  $t_0$  e  $X(I)$ , com  $I = 1, 2, \dots, PARES$ , onde PARES é o número de pontos experimentais, se refere aos diferentes valores que a variável independente  $t$  pode tomar.

Para correr o programa devemos introduzir, para além dos dados das variáveis que constam no Quadro 5.51, valores para PARES, NK, NOPT, NP,  $X(I)$ ,  $XI(I)$  e  $XS(I)$ . Estes valores encontram-se nos Quadros 5.52 e 5.53.

No Quadro 5.54 apresentam-se os resultados obtidos e na Figura 5.27 apresenta-se a curva ajustada aos pontos experimentais.

**Quadro 5.52** – Valores de PARES, NK, NP, NOPT

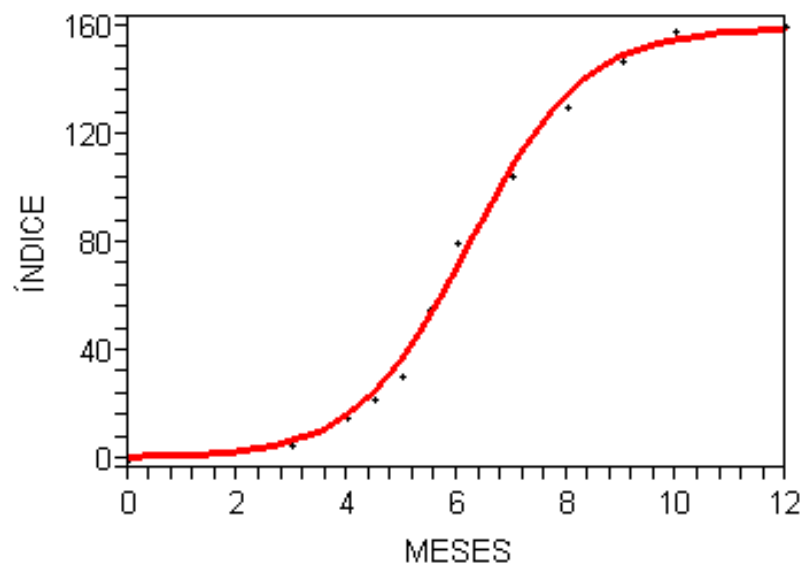
PARES	NK	NP	NOPT
13	500	3	3

**Quadro 5.53** – Valores iniciais e limites dos parâmetros

Ordem	Parâmetro	Limite Inferior	Limite Superior
I	$X(I)$	$XI(I)$	$XS(I)$
1	100	-1000	1000
2	1	-1000	1000
3	1	-1000	1000

**Quadro 5.54** – Resultados

Soma do Quadrado dos Erros	$k_1$	$k_2$	$t_0$
189,59283	158,91119	0,97109	6,22935

**Figura 5.27** – Índice de erosividade mensal.

Analisando os resultados apresentados no Quadro 5.54 podemos verificar que para os mesmos valores iniciais obtivemos uma redução na soma do quadrado dos erros ligeiramente inferior à obtida pelos autores.



### **5.3. Optimização dos Parâmetros de um Modelo Hidrológico**

Os modelos hidrológicos têm sido desenvolvidos tendo em vista a resolução de diferentes problemas como erosão, disponibilidade de águas superficiais ou subterrâneas, controlo de cheias, qualidade da água, ordenamento agrícola, etc.

Os modelos de simulação de escoamentos para uma bacia hidrográfica que vamos abordar são do tipo conceptual, onde múltiplas equações procuram interpretar as diferentes fases do ciclo hidrológico, fazendo intervir para tal um certo número de parâmetros.

Os parâmetros das funções dos diversos sub-sistemas destes modelos conceptuais são determinados recorrendo a métodos de optimização numérica. Regra geral, trata-se de encontrar um conjunto de parâmetros do modelo, portanto de um sistema não linear, tornando mínimo um critério de desvio entre uma sucessão de valores observados e uma sucessão de valores calculados.

A formulação do critério é um factor de grande importância a nível de resultados. De facto, se um critério tipo quadrado dos desvios favorece a reconstituição dos caudais de cheia, um critério de desvio relativo irá, pelo contrário, dar preferência aos débitos fracos.

A calibração do modelo faz-se em duas etapas distintas: optimização dos parâmetros e validação do modelo. Começa-se por dividir a sucessão de dados disponíveis em duas partes. Na primeira (com cerca de 70% dos dados) definem-se os valores iniciais para os parâmetros e os valores observados, quer para a precipitação quer para

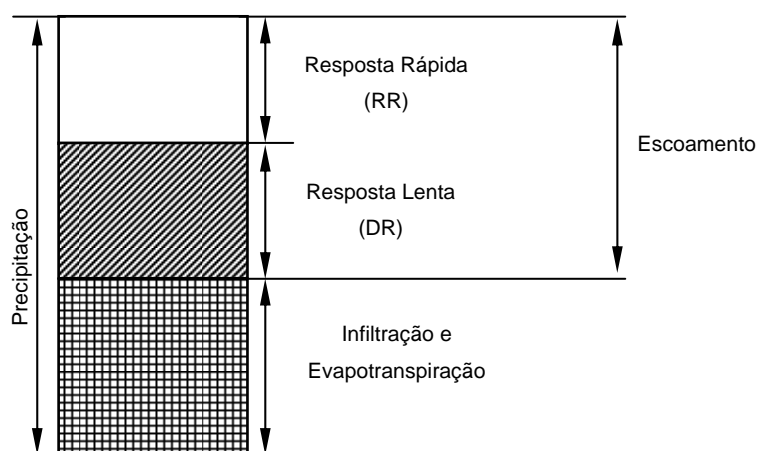
os escoamentos e evapotranspiração potencial. Comparam-se os valores calculados para os escoamentos com os valores observados através de um critério julgado adequado.

O algoritmo de otimização numérica vai modificando os valores dos parâmetros e o processo repete-se até se alcançar um valor aceitável como ótimo. Conseguida a convergência do processo de otimização o modelo é aplicado à segunda parte do conjunto de dados (aproximadamente 30% dos dados) e os resultados são apreciados completando-se assim o processo de validação.

### **5.3.1. Modelo Mensal**

O modelo de escoamento mensal que vamos apresentar é da autoria de Mccuen e Snyder (1986) e tem como finalidade mostrar como se interligam os códigos do modelo com os de otimização. Dado o escasso número de meses disponíveis apenas apresentamos a aplicação do método de Monro para a determinação dos parâmetros na fase de otimização.

Dispomos de 22 valores mensais de precipitação e de 18 valores de escoamentos observados. Esquemáticamente o modelo consiste na partição da precipitação em duas partes, uma correspondente à infiltração e evapotranspiração e a outra referente ao escoamento. Este, por sua vez, vai ser decomposto numa componente que é a resposta rápida à precipitação e numa componente que dá a resposta lenta e diferida no tempo. A Figura 5.28 esquematiza o modelo.



**Figura 5.28** – Partição da precipitação.

O escoamento medido na secção de referência em dado mês é a soma da resposta rápida à precipitação desse mês mais a resposta lenta à precipitação do mesmo mês acrescida das respostas às precipitações ocorridas nos 4 meses anteriores.

Assim, representando por  $RO(t)$  o escoamento para um dado mês  $t$ , por  $RR(t)$  e  $DR(t)$  as respostas rápida e lenta para um dado mês e por  $RC(t)$  os coeficientes de recessão para o mesmo mês, podemos escrever:

$$RO(t) = RR(t) + RC(1) \times DR(t) + RC(2) \times DR(t-1) + RC(3) \times DR(t-2) + RC(4) \times DR(t-3) + RC(5) \times DR(t-4)$$

No modelo intervêm 6 parâmetros  $x(i)$ ,  $i = 1, \dots, 6$ . Os coeficientes  $RC(t)$  serão calculados pela expressão,

$$RC(t) = \frac{\exp[-x(1) \times t]}{\sum_{t=1}^5 \exp[-x(1) \times t]} \quad t = 1, \dots, 5$$

cujos valores constam do Quadro 5.55

**Quadro 5.55** – Coeficientes  $RC(t)$ .

<b>t</b>	<b><math>RC(t)</math></b>
1	0,452
2	0,261
3	0,150
4	0,087
5	0,050

A infiltração e a evapotranspiração são calculadas de acordo com a expressão sinusoidal:

$$ST(t) = x(2) + x(3) \times \text{sen}[30 \times 0,01745 \times T \times x(4)]$$

Sendo T o resto da divisão inteira de  $(t + 1)$  por 12 adicionado de 1. t deve começar em Setembro para que os primeiros escoamentos calculados digam respeito a Janeiro.

Em termos de linguagem de Visual Basic vem  $T = (t + 1) \text{mod} 12 + 1$ .

O coeficiente  $k(t)$  com

$$RR(t) = k(t).ST(t)$$

e

$$DR(t) = [1 - k(t)].ST(t)$$

calcula-se a partir de

$$k(t) = x(5) + x(6).ST(t)$$

Deste modo podemos escrever,

$$\begin{aligned} RO(t) = & [k(1) \times ST(t)] + [RC(1) \times [1 - k(1)] \times ST(t)] + \\ & [RC(2) \times [1 - k(2)] \times ST(t - 1)] + \\ & [RC(3) \times [1 - k(3)] \times ST(t - 2)] + \\ & [RC(4) \times [1 - k(4)] \times ST(t - 3)] + \\ & [RC(5) \times [1 - k(5)] \times ST(t - 4)] + \end{aligned}$$

Para valores iniciais dos parâmetros utilizamos os valores propostos pelos autores do modelo que apresentamos no Quadro 5.56.

A interligação dos códigos do modelo com o do método de otimização de Monro é feita conforme se esquematiza na Figura 5.29.

**Quadro 5.56** – Valores iniciais para os parâmetros.

<b>Parâmetro</b>	<b>Valor</b>
$X_1$	0,5500
$X_2$	2,4900
$X_3$	1,8100
$X_4$	0,9000
$X_5$	0,1030
$X_6$	0,0390

```

Private Sub command1_click()
#####
'##          PROGRAMA MONRO_mensal          ##
#####
'*** Abertura do ficheiro para gravação dos resultados ***
Open "mensal.txt" For Output As #2
'*** Valores iniciais dos parâmetros ***
X(1) = 0.55: X(2) = 2.49: X(3) = 1.81
X(4) = 0.9: X(5) = 0.103: X(6) = 0.039
'*** Valores para a optimização ***
numa = 6: nper = 1: kc = 100: maxn = 500
***** ROTINA mensal *****
nstart = 0
8000:
'*** Dados ***
pp(1) = 8.71: pp(2) = 1.86: pp(3) = 4.48: pp(4) = 2.88
pp(5) = 1.3: pp(6) = 2.92: pp(7) = 6.67: pp(8) = 3.9
pp(9) = 4.13: pp(10) = 1.5: pp(11) = 5.08: pp(12) = 1.69
pp(13) = 4.5: pp(14) = 1.52: pp(15) = 3.14: pp(16) = 2.62
pp(17) = 7.18: pp(18) = 3.76: pp(19) = 5.13: pp(20) = 5.84
pp(21) = 0.13: p(22) = 4.81
eo(5) = 1.78: eo(6) = 1.45: eo(7) = 3.65: eo(8) = 3.05:
eo(9) = 1.19: eo(10) = 0.22: eo(11) = 0.42: eo(12) = 0.03
eo(13) = 0.12: eo(14) = 0.04: eo(15) = 0.28: eo(16) = 0.26
eo(17) = 5.88: eo(18) = 4: eo(19) = 3.1: eo(20) = 4.92
eo(21) = 0.13: eo(22) = 0.3
For i = 1 To 6

```

**Figura 5.29** – Programa “MONRO\_Mensal”.

```

Print #2, "x("; i, ")="; X(i);
Next i
s = 0
For j = 1 To 5
  r(j) = Exp(-X(1) * (j))
  s = s + r(j)
Next j
Debug.Print s
For j = 1 To 5
  rc(j) = r(j) / s
  Debug.Print "rc("; j, ")="; rc(j)
Next j
For t = 1 To 22
  se(t) = X(2) + X(3) * Sin((30 * ((t + 1) Mod 12 + 1) * 0.01745) + X(4))
  Debug.Print "se("; t, ")="; se(t)
  st(t) = pp(t) - se(t)
  Debug.Print "st("; t, ")="; st(t)
  k(t) = X(5) + X(6) * st(t)
  Debug.Print "k("; t, ")="; k(t)
  'resposta rapida
  rr(t) = k(t) * st(t)
  dr(t) = st(t) * (1 - k(t))
  Debug.Print "dr("; t, ")="; dr(t)
  Debug.Print "rr("; t, ")="; rr(t)
  cdr(t) = dr(t) * rrc(t)
  Debug.Print "cdr("; t, ")="; cdr(t)
Next t
For t = 5 To 22
  ro(t) = (k(t) * st(t)) + (rc(1) * (1 - k(t)) * st(t)) + (rc(2) * (1 - k(t - 1)) * st(t - 1)) + (rc(3) * (1 - k(t - 2)) * st(t - 2)) + (rc(4) * (1 - k(t - 3)) * st(t - 3)) + (rc(5) * (1 - k(t - 4)) * st(t - 4))
  Debug.Print "ro("; t, ")="; ro(t)
Next t
Call dif(F2, eo(), ro())
optim = F2
Call opt(X())
GoTo 8000
End
End Sub

```

**Figura 5.29 (Cont.)** – Programa “MONRO\_Mensal”.

Efectuámos 500 iterações e os resultados da optimização conduziram aos valores expressos no Quadro 5.57.

No Quadro 5.58. apresentam-se, para os diferentes meses, os valores de precipitação, de escoamento observado e de escoamento calculado.

**Quadro 5.57** – Resultados da otimização.

<b>x(i)</b>	<b>Valor Inicial</b>	<b>Valor Final</b>
$x_1$	0,5500	0,41183
$x_2$	2,4900	2,10051
$x_3$	1,8100	1,66551
$x_4$	0,9000	0,91843
$x_5$	0,1030	0,00033
$x_6$	0,0390	0,07063
<b>Função Critério</b>	12,8812	8,1068

Embora haja uma melhoria na função objectivo, os valores calculados para o escoamento são muito semelhantes aos calculados com os valores iniciais dos autores.

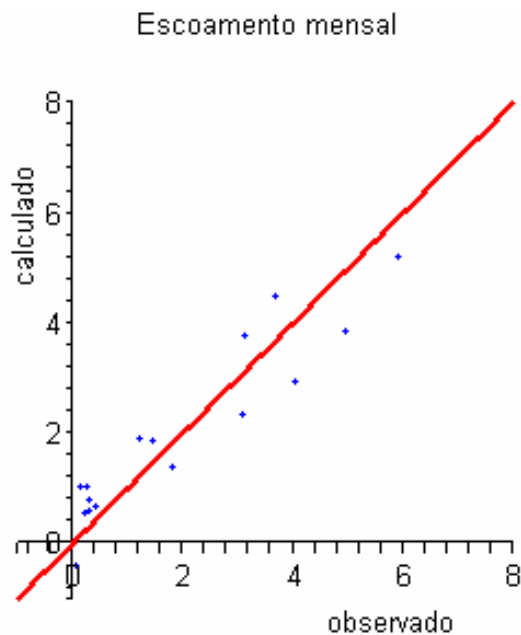
Na Figura 5.30 apresentam-se os valores de escoamento observado *versus* escoamento calculado com a optimização dos parâmetros.

Como exemplo de um modelo mensal de escoamento muito mais elaborado e complexo do que aquele que apresentámos, podemos citar o modelo de Fernandes (1990).



**Quadro 5.58 – Resultados.**

<b>Mês</b>	<b>Precipitação (in)</b>	<b>Escoamento Observado (in)</b>	<b>Escoamento Calculado (in)</b>
1	8,71	-	-
2	1,86	-	-
3	4,48	-	-
4	2,88	-	-
5	1,30	1,78	1,43
6	2,92	1,45	1,89
7	6,67	3,65	4,52
8	3,90	3,05	2,38
9	4,13	1,19	1,92
10	1,50	0,22	0,56
11	5,08	0,42	0,69
12	1,69	0,03	-0,38
13	4,50	0,12	0,11
14	1,52	0,04	-0,36
15	3,14	0,28	0,61
16	2,62	0,26	1,07
17	7,18	5,88	5,21
18	3,76	4,00	2,97
19	5,13	3,10	3,80
20	5,84	4,92	3,89
21	0,13	0,13	1,05
22	4,81	0,30	-0,62



**Figura 5.30** – Escoamento observado versus escoamento calculado.

### 5.3.2. Modelo Diário

O modelo matemático de simulação de escoamentos diários que vamos abordar é uma modificação do modelo CREC (Centre de Recherches et d'études de Chatou), elaborado em 1969 por um grupo de investigadores da Universidade de Montpellier (Cormary e Guilbot, 1969) e mais tarde modificado por Mello (1987).

Trata-se de um modelo conceptual de três reservatórios que põe em jogo, segundo um esquema estabelecido, os diferentes processos intervenientes no escoamento de uma bacia de drenagem (Figura 5.31).

O primeiro reservatório S, superficial, cujo nível tem em conta a humidade do solo, alimenta a evapotranspiração. É ele também que permite calcular a chuva eficaz, designação que no modelo se refere à chuva disponível para o escoamento que irá alimentar os reservatórios H, hipodérmico, e G, subterrâneo. O esvaziamento do reservatório H produz um escoamento rápido enquanto que o do reservatório G origina um escoamento lento. É também este reservatório S que controla a redução da evapotranspiração potencial.

Os reservatórios H e G representam respectivamente a zona insaturada e a zona saturada do perfil do solo. O reservatório H tem várias funções. É em relação ao teor em água deste reservatório que se define a fracção da precipitação que participará nos escoamentos hipodérmicos e de base.

Os reservatórios H e G simulam o escoamento por esvaziamento em paralelo. Uma parte do escoamento da bacia pode resultar da fracção da precipitação que é escoada superficialmente, desde que a intensidade da chuvada ultrapasse determinado valor crítico.

O conjunto de operações efectuadas no modelo comporta duas partes essenciais:

- a primeira calcula a fracção da precipitação que participará nos escoamentos hipodérmico e de base (parte de produção);
- a segunda reparte esta chuva eficaz pelos reservatórios H e G e calcula os respectivos esvaziamentos ao nível diário (parte de transferência).

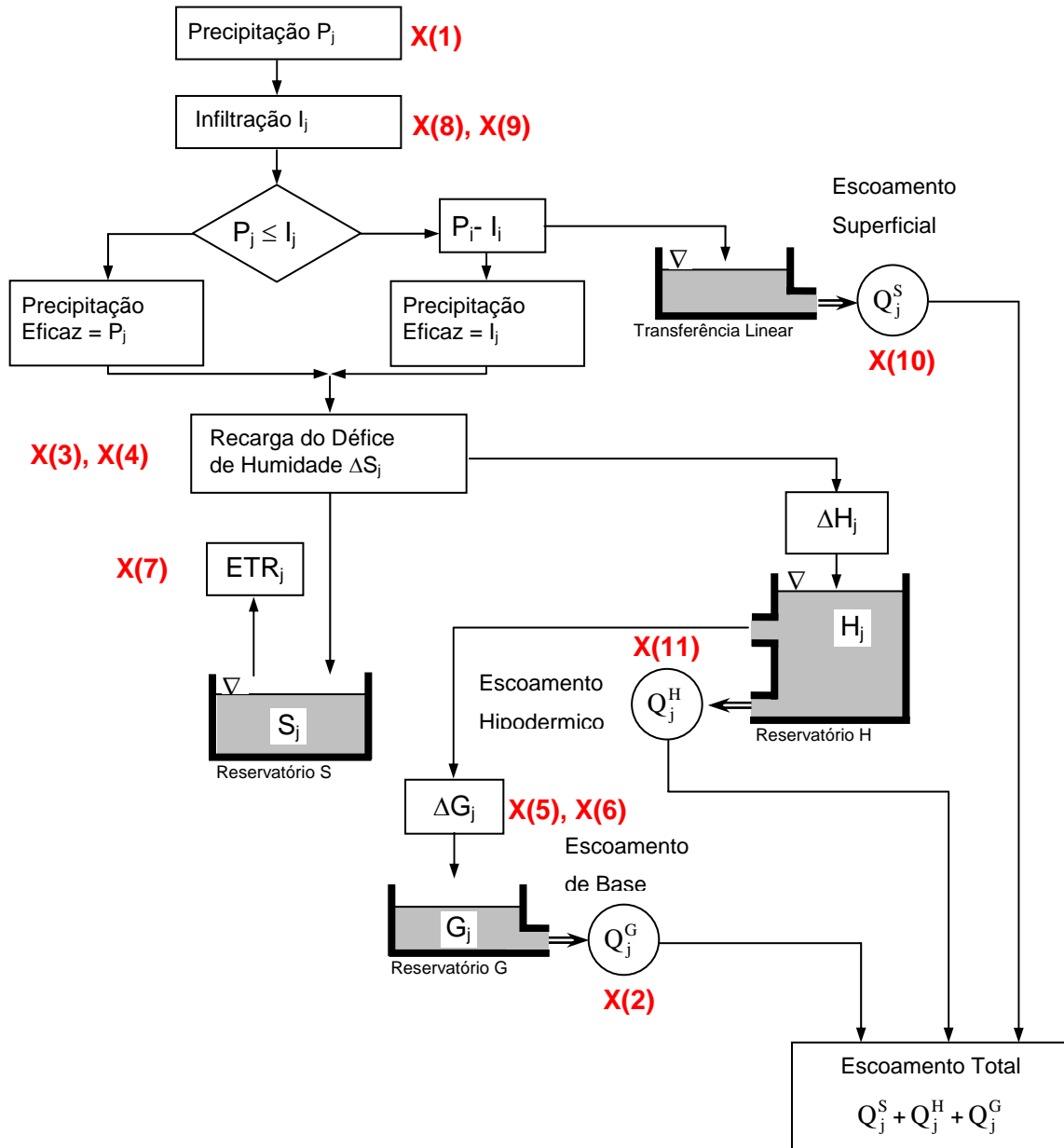


Figura 5.31 – Representação esquemática do modelo CREC.

Os símbolos utilizados na Figura 5.31 têm o seguinte significado:

$P_j$  - Chuva diária ;

$I_j$  - Infiltração potencial diária;

$\Delta S_j$  - Alimentação diária do reservatório S;

$ETR_j$  - Evapotranspiração real diária;

$S_j, S_{j-1}$  - Níveis do reservatório S em dias consecutivos;

$H_j, H_{j-1}$  - Níveis do reservatório H em dias consecutivos;

$G_j, G_{j-1}$  - Níveis do reservatório G em dias consecutivos;

$\Delta H_j$  - Alimentação diária do reservatório H;

$\Delta G_j$  - Alimentação diária do reservatório G;

$Q_j^S$  - Escoamento diário devido ao escoamento superficial;

$Q_j^H$  - Escoamento diário hipodérmico;

$Q_j^G$  - Escoamento diário de base;

$X_i$  ( $i=1, \dots, 11$ ) - Parâmetros do modelo.

Nas expressões analíticas das diferentes funções que caracterizam as partes de produção e de transferência do modelo aparecem parâmetros que serão determinados numericamente por processos de otimização.

Na Figura 5.32 apresentamos um esquema da estruturação do programa CREC, que consta das seguintes partes:

- Programa principal;
- Subrotina OPTIMI;
- Subrotina CREC;

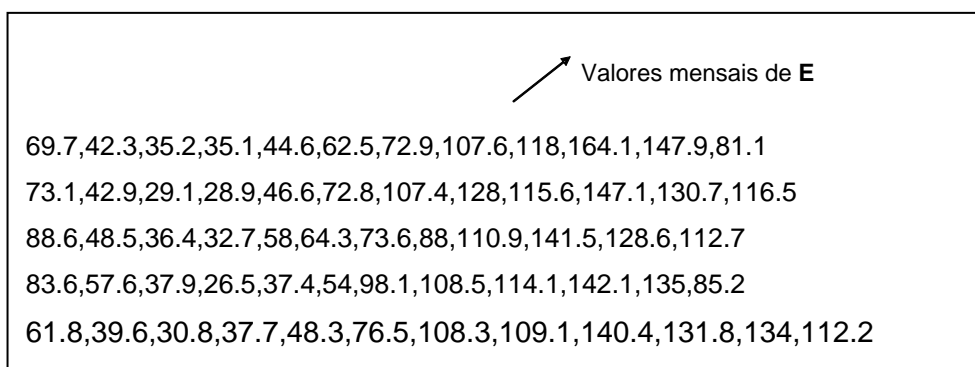
- Subrotina PROPAG;
- Subrotina DIF.

O programa principal é essencialmente um programa para leitura de dados, nomeadamente:

- NA – Número de anos;
- ANO – Primeiro ano civil de ensaio;
- NK – Número máximo de iterações de optimização;
- NOPT – Número de parâmetros a optimizar;
- Z – Área da bacia (km<sup>2</sup>);
- COMEÇOANO\$ - Ano hidrológico;
- CHUVA – Precipitação diária (mm);
- DEB – Caudal diário (m<sup>3</sup>s<sup>-1</sup>);
- E – evapotranspiração mensal (mm);
- X – Valor inicial dos parâmetros;
- XS – Limite superior de variação dos parâmetros;
- XI – Limite inferior de variação dos parâmetros.

Os valores de precipitação, caudal e evapotranspiração são lidos a partir de um ficheiro em formato "TXT". Nos ficheiros de dados da precipitação e do caudal o primeiro valor refere-se ao ano hidrológico, os dados seguintes referem-se aos valores diários da precipitação e caudal, na sequência dos meses, como se exemplifica na Figura 5.33. No ficheiro de dados relativo à evapotranspiração potencial mensal estes valores estão dispostos em linha, correspondendo cada linha a um ano, como se mostra na Figura 5.34.





**Figura 5.34** – Ficheiro de dados de evapotranspiração.

A subrotina OPTIMI desenvolve o método de optimização de Rosenbrock.

A subrotina CREC consta do programa da conversão da precipitação em escoamento.

A subrotina PROPAG consiste na propagação do escoamento superficial.

Finalmente a subrotina DIF calcula o valor da função objectivo em cada iteração da optimização.

O programa principal chama a subrotina de optimização e esta, por cada iteração, faz três chamadas à subrotina CREC. As subrotinas PROPAG e DIF são chamadas uma vez pela subrotina CREC.

Na Figura 5.35 mostra-se uma parte do código do modelo, onde se podem ver as entradas pedidas e os parâmetros.

Todos os detalhes de natureza hidrológica podem encontrar-se em Mello (1987).

Apresentamos um exemplo da aplicação deste modelo à bacia do rio Simonte em Louçainha. Utilizámos cinco anos consecutivos (1968/69 a 1972/73) e escolhemos o método de Rosenbrock para optimização dos



parâmetros, que segundo Ibbrit e O'Donnell (1971) é dos mais apropriados para a modelação hidrológica.

Devido às alterações que introduzimos no modelo, tentámos uma melhoria na qualidade da optimização, partindo dos valores óptimos a que tínhamos chegado e utilizando como função critério a soma do quadrado da diferença entre os valores simulados e observados.

Após 300 iterações os valores óptimos atingidos foram os apresentados no Quadro 5.59.

```

Private Sub command1_click()
#####
##          PROGRAMA ROSEN_CREC          ##
#####
*** Abertura do ficheiro para gravação dos resultados ***
Open "Crec.txt " For Output As #2
'DIAS DE CADA MÊS DO ANO HIDROLÓGICO
MES(1) = 31: MES(2) = 30: MES(3) = 31: MES(4) = 31: MES(5) = 28
MES(6) = 31: MES(7) = 30: MES(8) = 31: MES(9) = 30: MES(10) = 31
MES(11) = 31: MES(12) = 30
NA = 5
ANO = 1968
For i = 1 To NA
  K(i) = ANO + i
  Print #2, "k("; i; ")="; K(i)
Next i
'ENTRADA DAS CHUVAS
Open "CHUVA.txt" For Input As #1
For AN = 1 To NA
  Input #1, COMEÇOANOS$
  For M = 1 To 12
    MES(5) = 28
    If K(AN) Mod 4 = 0 Then
      MES(5) = 29
    End If
    For DI = 1 To MES(M)
      Input #1, CHUVA(DI, M, AN)
    Next DI
  Next M
Next AN
Close #1
'ENTRADA DE CAUDAIS
Open "CAUDAL.txt" For Input As #1
For AN = 1 To NA

```

**Figura 5.35** – Programa “ROSEN\_CREC”.

```

Input #1, COMEÇANO$
For M = 1 To 12
  MES(5) = 28
  If K(AN) Mod 4 = 0 Then
    MES(5) = 29
  End If
  For DI = 1 To MES(M)
    Input #1, DEB(DI, M, AN)
  Next DI
Next M
Next AN
Close #1
' ENTRADA DA EVAPOTRANSPIRACAO
Open "EVAPORA.txt" For Input As #1
For AN = 1 To NA
  If K(AN) Mod 4 = 0 Then
    MES(5) = 29
  Else
    MES(5) = 28
  End If
  For M = 1 To 12
    Input #1, E(AN, M)
    E(AN, M) = E(AN, M) / MES(M)
  Next M
Next AN
Close #1
' VALORES INICIAS PARA OS PARAMETROS
X(1) = 0.0331: X(2) = 0.5157: X(3) = 0.0569: X(4) = 0.0569
X(5) = 1: X(6) = 49.1637: X(7) = 0.3359: X(8) = 36.4554
X(9) = 30.5192: X(10) = 0.093: X(11) = 0.6992
' Área da bacia em km2
Z = 4
PV = 0
DV = 0
' SE nk = -1 NÃO FAZ OPTIMIZAÇÃO
NK = 300
MAXIT = NK
ITERC = 0
NOPT = 11
For i = 1 To 7
  XI(i) = 0.0001
Next i
XI(8) = 30: XI(9) = 30: XI(10) = 0.0001: XI(11) = 0.0001
For i = 1 To 4
  XS(i) = 0.9999
Next i
XS(5) = 50: XS(6) = 50: XS(7) = 150: XS(8) = 200
XS(9) = 200: XS(10) = 0.9999: XS(11) = 0.9999
N = NOPT
Call Optimizacao(X(), XI(), XS())
Close #2
End
End Sub

```

**Figura 5.35 (Cont.)** – Programa “ROSEN\_CREC”.

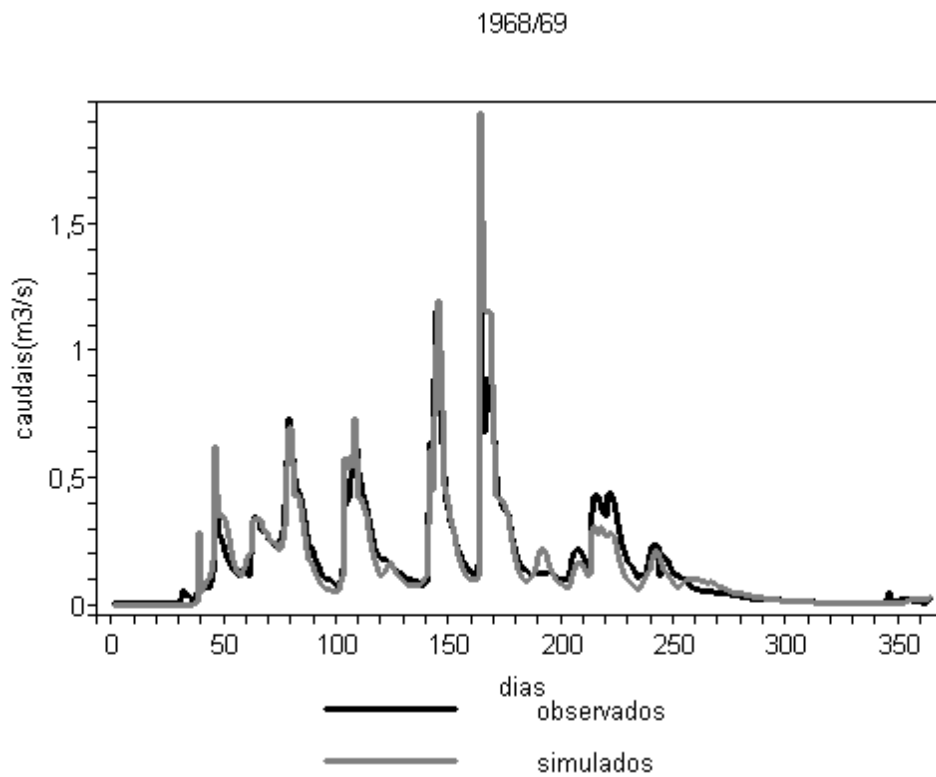
**Quadro 5.59** – Valores óptimos dos parâmetros.

<b>Ordem do Parâmetro</b>	<b>Valor inicial</b>	<b>Valor final</b>
1	0,0331	0,03348
2	0,5157	0,64657
3	0,0569	0,00428
4	0,0569	0,04037
5	1,0	2,04374
6	49,1637	49,42724
7	0,3359	0,40455
8	36,4554	38,16524
9	30,5192	30,43982
10	0,0930	0,16569
11	0,6992	0,68494
<b>Função Objectivo</b>	0,0079465	0,0043855

Nas Figuras 5.36 e 5.37 podem ver-se os bons resultados obtidos pela simulação dos caudais diários para o rio Simonte (Louçainha), respectivamente para as fases de calibração (ano 1968/69) e para a fase de validação (ano 1971/72).

Embora o trabalho citado (Mello, 1987) tenha utilizado o método de optimização de Rosenbrock, apresentamos também (CD em anexo) o código equivalente para a aplicação do método de Monro (“MONRO\_CREC”).

Chamamos a atenção para o facto da interligação das subrotinas ser aqui diferente, sendo o programa Monro chamado a partir do modelo, conforme se resume na Figura 5.38.



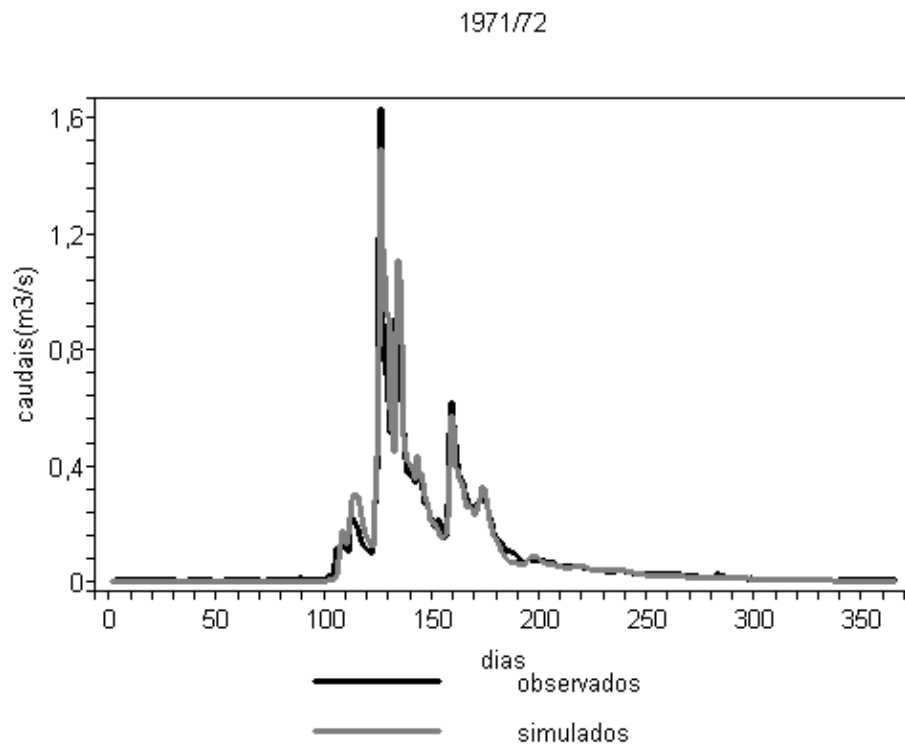
**Figura 5.36** – Caudais diários para a fase de calibração.

No código apresentado, e resumindo etapas, pode ver-se,

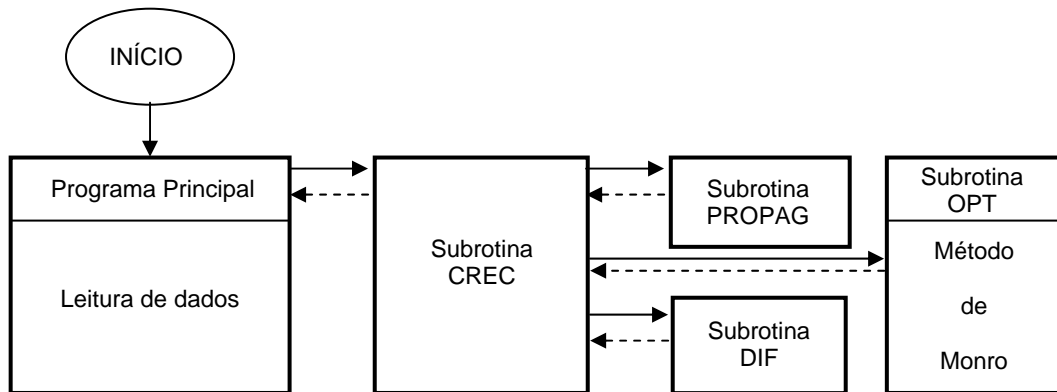
```
NSTART = 0  
8000: Modelo  
Optim = F  
Call Opt  
Goto 8000
```

É curioso observar o que se passa com 300 iterações efectuadas com o programa Monro partindo dos mesmos parâmetros atrás

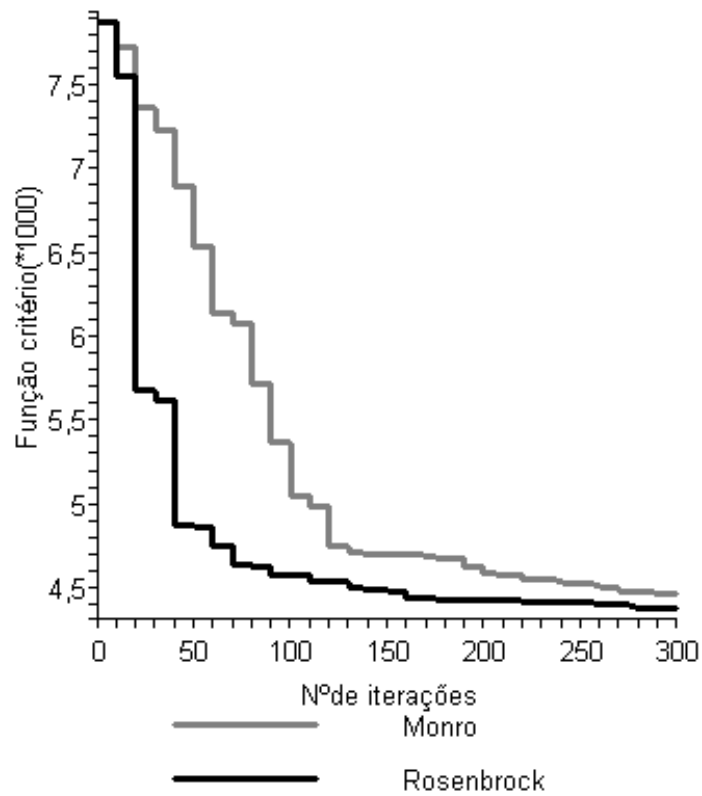
mencionados (Quadro 5.59) para o método de Rosenbrock. A função objectivo, tem o decrescimento que indicamos na Figura 5.39, onde o método de Rosenbrock apresenta, para o mesmo número de iterações, uma quebra mais acentuada.



**Figura 5.37** – Caudais diários para a fase de validação.



**Figura 5.38** – Esquema da estruturação do programa MONRO\_CREC.



**Figura 5.39** – Evolução da função objectivo para os dois métodos.

Em Hipólito (1985) pode ver-se um modelo diário aplicado também a bacias hidrográficas portuguesas com utilização do método de optimização de Monro.

Muito se poderia comentar sobre os modelos apresentados, mas o nosso objectivo ao estudar a “optimização numérica” apenas nos levou a mostrar como intervém neles o processo de optimização e não um estudo aprofundado no campo da hidrologia.





## BIBLIOGRAFIA

- Bard, Y. (1974). *Nonlinear Parameter Estimation*. Academic Press, New York.
- Bonnans, J. F., Gilbert, J. C., Lemaréchal, C. e Sagastizábal, C. A. (2003). *Numerical Optimization. Theoretical and Practical Aspects*. Springer-Verlag, Berlin.
- Boratto, F. (1984). *BASIC para Engenheiros e Cientistas*. Livros Técnicos e Científicos Editora, Rio de Janeiro.
- Cormary, Y. e Guilbot, A. (1969). Relations Pluie-Débit sur le Bassin de la Sioule. *Fac. Sci. Montpellier, Rapp. D.G.R.S.T.*, 30.
- Draper, N. R. e Smith, H. (1966). *Applied Regression Analysis*. John Wiley and Sons, New York.
- Fernandes, A. L. (1990). *Modelação Matemática da Transformação da Precipitação em Escoamento*. Dissertação de Mestrado, Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa.
- Frenz, C. M. (1972). *Visual Basic and Visual Basic.NET for Scientists and Engineers*. APress.
- Himmelblau, D. (2002). *Applied Nonlinear Programming*. McGraw-Hill, New York.
- Hipólito, J. N. R. (1985). *NWSIST: Um Sistema de Simulação Contínua de Processos Hidrológicos*. Tese de Doutoramento, Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa.

- Hooke, R. e Jeeves, T. A. (1961). Direct Search Solution of Numerical and Statistical Problems. *JACM*, 8.
- Ibbitt, R. P. e O' Donnell, T. (1971). Fitting Methods for Conceptual Catchments Models . *Journal of the Hydraulic Division, ASCE*, 97(HY9), 1331-1342.
- Karlinger, M. R. e Skrivan, J. A. (1981). Kriging analysis of mean annual precipitation, Powder River Basin, Montana and Wyoming. *U.S. Geological Survey Water-Resources Investigations* 80-50.
- McCuen, R. e Snyder, W. (1986). *Hydrologic Modelling. Statistical Methods and Applications*. Prentice Hall, New Jersey.
- Mello, F. M. (1987). *Um Modelo Global de Simulação Contínua para Conversão da Precipitação em Escoamento*. Tese de Doutoramento, Universidade de Évora, Évora.
- Monro, J. C. (1971). Direct Search Optimization in Mathematical Modelling and Watershed Model Application. *NOAA Technical Memorandum NWS, Hydro* 12.
- Mualem, Y. (1976). A New Model for Predicting Hydraulic Conductivity of Unsaturated Porous Média. *Water Resources Research*, 12(13), 513-522.
- Nelder, J. e Mead, R. (1964). A Simplex Method for Function Minimization. *Computer J.*, 7, 308-315.
- Rao, S. S. (1979). *Optimization. Theory and Application*. Wiley Eastern, New Delhi.
- Rosenbrock, H. H. (1960). An Automatic Method for Finding the Greatest on Least Value of a Function. *Computer J.*, 73, 175-180.
- Scheid, F. (1991). *Análise Numérica*. McGraw-Hill de Portugal (Schaum), Lisboa.

- 
- Smirnov, G. e Bushenkov, V. (2005). *Curso de Optimização. Programação Matemática, Cálculo de Variações, Controlo Ótimo*. Escolar Editora, Lisboa.
- van Genuchten, M. (1980). A Closed-Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Sci. Soc. Am. J.*, 44(5), 892-898.
- Webster, R. e Burges, T. M. (1983). Spatial Variation in Soil and the Role Kriging. *Agricultural Water Management*, 6, 111-122.



## LISTAGEM DOS PROGRAMAS

(Referentes ao CD ANEXO)

Apresenta-se seguidamente a lista dos programas utilizados e respectivas designações.

### A) Programas para Minimizar uma Função

1. Método de Monro – *MONROMINIMO*
2. Método de Nelder e Mead – *SIMPLEXMINIMO*
3. Método de Rosenbrock – *ROSENBROCKMINIMO*

### B) Programas para Ajustamento de uma Função com uma Variável Independente

4. Método dos Mínimos Quadrados Não Lineares –  
*BORATO\_1\_VARIAVEL*
5. Método de Monro – *MONRO\_1\_VARIAVEL*
6. Método de Nelder e Mead – *SIMPLEX\_1\_VARIAVEL*
7. Método de Rosenbrock – *ROSENBROCK\_1\_VARIAVEL*

**C) Programas para Ajustamento de uma Função com duas ou mais Variáveis Independentes**

8. Método dos Mínimos Quadrados Não Lineares – *BORATO\_2\_VARIABEL*
9. Método de Monro – *MONRO\_2\_VARIABEL*
10. Método de Nelder e Mead – *SIMPLEX\_2\_VARIABEL*
11. Método de Rosenbrock – *ROSENBROCK\_2\_VARIABEL*

**D) Programas para Optimização dos Parâmetros de um Modelo**

12. Modelo Mensal – *MONRO\_mensal*
13. Modelo Diário – *ROSEN\_CREC*
14. Modelo Diário – *MONRO\_CREC*