

OBSERVAÇÕES SOBRE UM PROBLEMA NUMÉRICO SIMPLES

Mário M. Graça

Departamento de Matemática
Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais 1049-001 Lisboa, Portugal
e-mail: mgraca@math.tecnico.ulisboa.pt

Resumo: Propomos um problema simples de cálculo de raízes em \mathbb{R}^2 ao qual o sistema *Mathematica* não dá resposta satisfatória. Resolvemos o problema construindo certas funções iteradoras que nos permitem localizar globalmente um grande número dessas raízes. O método proposto usa apenas valores da função evitando assim o cálculo de matrizes jacobianas ou suas aproximações por diferenças finitas convencionais.

Abstract: A simple root finding problem in \mathbb{R}^2 is proposed to which the *Mathematica* system does not give a satisfactory answer. We solve the problem constructing certain iteration maps enabling a global localization of a great number of these roots. The proposed method only uses function values and so avoiding the computation of Jacobian matrices or their approximations by conventional finite differences.

palavras-chave: Raíz; ponto fixo; função iteradora; método quasi-Newton; diferença centrada.

keywords: Root; fixed point, iteration function; quasi-Newton method; centered difference.

1 Introdução

O problema que nos propomos resolver consiste em aproximar *globalmente* os zeros da função $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ definida por $f(x, y) = (f_1(x, y), f_2(x, y))$, com

$$\begin{cases} f_1(x, y) = \sin [4(x - 5)^2 + 1/3(y - 5)^2 - 1] \\ f_2(x, y) = \cos [-3/2(x - 4)^2 + 4/3(y - 5)^2 - 1], \end{cases} \quad (1)$$

onde (x, y) pertence ao domínio convexo $A = [-4, 8] \times [-4, 8] \subset \mathbb{R}^2$. O objectivo principal é separar os zeros de f tendo em vista, posteriormente, calcular simultaneamente aproximações desses zeros, aplicando apenas uma vez uma certa função iteradora que iremos construir. Note-se que as expressões das componentes da função em (1) são formalmente simples e f é

de classe $C^\infty(\mathbb{R}^2)$. Por isso, esta função pode ser considerada ideal como modelo para as questões de que nos iremos ocupar, a saber:

- (A) Como localizar globalmente os zeros de f no domínio A ;
- (B) Como obter esses zeros simultaneamente, sem necessidade de se escolher explicitamente aproximações iniciais ‘suficientemente próximas’ desses zeros. Note-se que tais aproximações são indispensáveis por exemplo quando se aplica o método de Newton clássico;
- (C) Como evitar o cálculo da matriz jacobiana da função f (imprescindível no referido método de Newton), ou aproximações dessa matriz através de diferenças finitas convencionais – técnica utilizada nomeadamente nos métodos designados por quasi-Newton;
- (D) No caso de não ser possível aproximar globalmente *todos* os zeros da função f , calcular simultaneamente aproximações de um grande número deles.

O nosso objectivo inicial consiste em definir uma função iteradora capaz de resolver a questão primordial (A) usando a função f definida em (1) como modelo de teste.

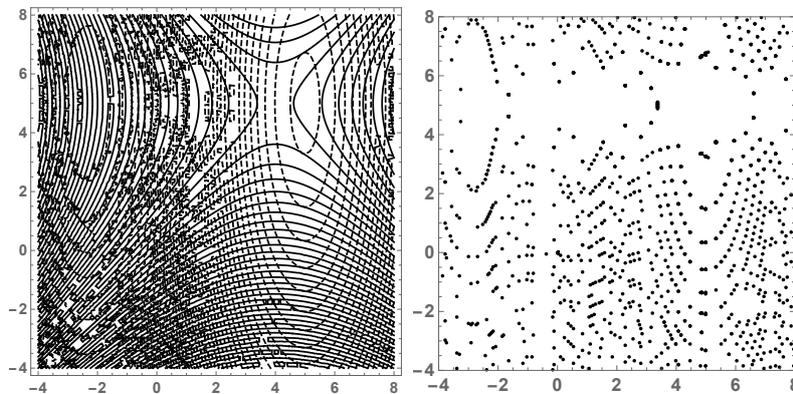


Figura 1: (Esquerda) – Curvas de nível zero para f ;
(Direita) – Localização global de um grande número de zeros de f , aplicando uma função iteradora composta n^2 (cf. Secção 3).

Acontece que é muito elevado o número de zeros da função f no domínio A considerado, como pode ser observado na Figura 1 (esquerda) onde estão

traçadas as curvas de nível zero das funções componentes de f , respectivamente a cheio e a tracejado. Os pontos de intersecção dessas curvas correspondem aos zeros da função f . Pode ser verificado que, mesmo para esta função simples, rotinas sofisticadas para o cálculo global de raízes, como é o caso de *Reduce* ou *FindInstance* do sistema *Mathematica* [8], não dão resposta em tempo razoável (por exemplo, em menos de 20 segundos), ou não conseguem mesmo responder ao pretendido se quisermos calcular simultaneamente aproximações de um grande número de raízes da equação $f(x, y) = (0, 0)$.

Na Figura 1 à direita é evidenciada a localização de um certo número de zeros da função modelo. Os pontos a negro visíveis nessa figura resultam de conglomerados de aproximações numéricas de zeros da função, os quais foram calculados simultaneamente através de uma determinada função iteradora baseada em diferenças finitas centradas generalizadas propostas neste trabalho (ver Definições 2.1, 2.2 e 2.3). Na Secção 2 descrevemos a construção de funções iteradoras centradas e na Secção 3 apresentamos alguns resultados numéricos da sua aplicação ao problema modelo.

As funções iteradoras que propomos, adiante designadas como *iteradoras centradas*, além de servirem para dar resposta às questões (A) e (B), foram definidas de modo a satisfazer a questão (C). Tais funções iteradoras baseiam-se numa generalização dos operadores diferenças finitas tradicionalmente usados para aproximar derivadas de funções num ponto, os quais dependem da fixação prévia de um certo ‘passo’ constante h . Em particular, na expressão da função iteradora de Newton clássica $\mathcal{N}_f = x - J_f^{-1}(x) f(x)$, substituímos a matriz jacobiana $J_f(x)$ por um operador $D_f(x)$, que definimos adiante em (4)-(5), o qual dispensa o uso do referido passo h adoptado nos métodos quasi-Newton tradicionais. Provamos (ver Proposição 2.1) que para funções f continuamente diferenciáveis numa vizinhança de um zero z , o operador $D_f(x)$ aproxima a matriz jacobiana $J_f(x)$. As funções iteradoras centradas são depois convenientemente modificadas de modo a ser bem definidas em todo o espaço \mathbb{R}^2 e evidenciarem apenas os pontos do domínio A da função de partida f que são relevantes para a finalidade em vista, ou seja, a separação global dos zeros de f nesse domínio, tal como pretendido na questão (A).

Dado que nas aplicações podem ocorrer funções bivariadas f de complexidade muito superior à da função modelo (1), ou mesmo funções cujas componentes não são conhecidas explicitamente, inviabilizando portanto a construção directa da matriz jacobiana $J_f(x)$, conclui-se que mesmo a resolução parcial da referida questão (A) aqui ensaiada quanto ao problema

simples que enunciámos, assume um grande valor prático em problemas reais. Esperamos que as funções iteradoras aqui sugeridas possam ser úteis nesse contexto mais vasto.

Na Secção 3 respondemos finalmente à questão (D), calculando simultaneamente um número elevado de aproximações dos zeros da função modelo no domínio considerado, por aplicação de uma certa função iteradora centrada composta. Para selecção desses zeros aproximados foi usado como critério de resíduo $\|f(x)\|_\infty < 10^{-7}$. Apresentamos na Secção 4 código *Mathematica* que desenvolvemos a fim de testar algumas funções iteradoras centradas aplicadas ao problema modelo e a uma bateria de funções de teste para as quais não é possível o cálculo simultâneo de raízes mediante as referidas rotinas do sistema *Mathematica*.

A discussão detalhada de alguns factos subentendidos ou apenas aflorados nesta nota bem como questões relativas à convergência de métodos iterativos resultando das funções iteradoras centradas aqui introduzidas será efectuada noutro lugar. A técnica sugerida de substituição da matriz jacobiana $J_f(x)$ pode ser facilmente generalizada a derivadas de ordens superiores, originando funções iteradoras de ordem de convergência superior à do método de Newton clássico.

2 Operador diferenças centradas generalizadas

Diferenças finitas são usadas com sucesso nas mais diversas aplicações e em particular nos chamados *métodos quasi-Newton* para resolução de equações em \mathbb{R}^k (ver, por exemplo, [1] e [6]). Sob hipóteses bastante gerais, dada uma função $f : \mathbb{R}^k \rightarrow \mathbb{R}^k$ definida num convexo $A \subset \mathbb{R}^k$ e a correspondente função iteradora clássica de Newton $\mathcal{N}_f(x) = x - J_f^{-1}(x)f(x)$ (cujas propriedades são discutidas por exemplo em [5], [7]), sabe-se que substituindo a matriz jacobiana $J_f(x)$ por uma diferença finita de “passo” h , obtém-se um método do tipo $\tilde{\mathcal{N}}_f(x) = x - B^{-1}(x)f(x)$ habitualmente designado por *quasi-Newton*, o qual é localmente convergente desde que o passo h seja criteriosamente escolhido ([2], pag. 95). A matriz quadrada B em $\tilde{\mathcal{N}}_f$ supõe-se invertível numa vizinhança de um zero $z = (z_1, \dots, z_k) \in \mathbb{R}^k$ da função f .

Se nada se disser a respeito da dimensão k , admitimos a partir de agora que uma função genérica $f(x) = (f_1(x), f_2(x), \dots, f_k(x))$ é dada num domínio convexo $A \subset \mathbb{R}^k$, para $k \geq 1$. Relembremos o conceito de diferença finita de passo $h > 0$. Entre várias possibilidades para aproximar derivadas de uma função num ponto, de primeira ou ordem superior, avultam as chamadas

diferenças progressivas, regressivas e centradas. Neste trabalho cingimo-nos apenas a diferenças *centradas* para estimar a primeira derivada de uma função num ponto x , mediante generalização abaixo detalhada. Em \mathbb{R}^k , com $k \geq 2$, tais diferenças podem ser usadas como aproximantes de derivadas parciais de funções $f(x) = (f_1(x), \dots, f_k(x))$, tal como se indica a seguir.

Considerem-se as diferenças *centradas* clássicas, de passo $h > 0$:

$$\Delta_{i,j} f(x) = \frac{f_i(x + h e_j) - f_i(x - h e_j)}{2h}, \quad i, j = 1, \dots, k \quad (2)$$

onde $x = (x_1, \dots, x_k)$ e $e_j = (0, 0, \dots, 1, 0, \dots, 0)$ designa o j -ésimo vector da base canónica de \mathbb{R}^k . Sendo $f \in C^3(\mathbb{R}^k)$ e $h > 0$ suficientemente pequeno, pode concluir-se que as derivadas parciais $\frac{\partial f_i}{\partial x_j}$ satisfazem ([4]. Ch. 6),

$$\frac{\partial f_i}{\partial x_j}(x) = \Delta_{i,j} f(x) + \mathcal{O}(h^2). \quad (3)$$

A fim de construirmos uma função iteradora no qual seja dispensável a consideração explícita do parâmetro $h > 0$ em (2), definimos a seguir o operador $D_f(x) = [\delta_{i,j} f(x)]_{i,j=1}^k$, que designamos por *operador diferenças generalizadas*. Como neste trabalho usaremos somente diferenças do tipo centrado o operador $D_f(x)$ dir-se-á um operador às diferenças centradas. Em substituição do passo $h > 0$ usado na definição de diferenças finitas convencionais, consideramos $h(x) = \|f(x)\|_2^2$, onde $\|\cdot\|_2$ designa a norma euclidiana em \mathbb{R}^k .

Definição 2.1. (*Operador diferença centrada*)

Seja $h(x) = \|(f_1(x), \dots, f_k(x))\|_2^2 = f_1^2(x) + \dots + f_k^2(x)$. O operador diferença centrada (generalizada) é definido pela matriz quadrada $D_f(x)$, cujas entradas $\delta_{i,j} f(x)$, para $i, j = 1, \dots, k$, são as seguintes:

$$\begin{cases} \delta_{i,j} f(x) = \delta_{i,j}, & \text{se } h(x) = 0 \\ \delta_{i,j} f(x) = \frac{f_i(x + h(x) e_j) - f_i(x - h(x) e_j)}{2h(x)}, & \text{se } h(x) \neq 0 \end{cases} \quad (4)$$

onde δ_{ij} designa o símbolo de Kronecker ($\delta_{ii} = 1$, $\delta_{ij} = 0$, para $i \neq j$).

Observe-se que o operador $D_f(x)$ está bem definido para todo o $x \in \mathbb{R}^k$. Notar que quando x é um ponto de \mathbb{R}^k próximo de um zero z da função f , ter-se-á $\|f(x)\| \simeq 0$, para qualquer norma definida em \mathbb{R}^k , pelo que nesse caso a função $h(x)$ está próxima do valor escalar nulo 0. Assim, no

contexto do problema de cálculo aproximado de zeros de uma função, o operador $D_f(x)$ é porventura a generalização mais natural que se pode conceber em substituição do correspondente operador utilizando diferenças centradas convencionais como em (2). Surpreendentemente, tanto quanto sabemos, na literatura dedicada aos algoritmos de aproximação de raízes de equações, por exemplo em [5], [7], [6],[2], [4], apenas são consideradas aproximações da matriz jacobiana J_f usando diferenças finitas convencionais, nas quais uma escolha criteriosa do respectivo passo constante h é crucial nomeadamente para garantir a estabilidade dos respectivos algoritmos.

Tendo em conta a Definição 2.1, neste trabalho a matriz jacobiana $J_f(x) = [\partial f_i(x)/\partial x_j]_{i,j=1}^k$ será aproximada por uma das seguintes matrizes:

$$\left\{ \begin{array}{l} D_f(x) = I_k, \quad \text{se } h(x) = 0. \\ \\ D_f(x) = \begin{bmatrix} \delta_{1,1} f_1(x) & \delta_{1,2} f_1(x) & \cdots & \delta_{1,k} f_1(x) \\ \delta_{2,1} f_2(x) & \delta_{2,2} f_2(x) & \cdots & \delta_{2,k} f_2(x) \\ \vdots & \vdots & \vdots & \vdots \\ \delta_{k,1} f_k(x) & \delta_{k,2} f_k(x) & \cdots & \delta_{k,k} f_k(x) \end{bmatrix}, \quad \text{se } h(x) \neq 0 \end{array} \right. \quad (5)$$

onde I_k designa a matriz identidade ($k \times k$) e $0 \in \mathbb{R}$. Consequentemente, a matriz $D_f(x)$ está definido para todo o $x \in \mathbb{R}^k$.

A correspondente função iteradora *centrada* é assim definida:

Definição 2.2. (*Função iteradora centrada*)

$$g(x) = x - D_f^{-1}(x) f(x), \quad (6)$$

onde D_f é a matriz dada por (5).

Na Proposição 2.1 a seguir mostra-se que para uma função $f \in C^1(V_z)$, onde V_z denota uma vizinhança de um seu zero z , o operador (5), com $h(x) \neq 0$, aproxima $f'(x) = J_f(x)$, pelo que a função iteradora centrada (6) deverá ser considerada como uma função geradora de um método iterativo do tipo quasi-Newton.

Proposição 2.1. *Seja $z \in A$ um zero da função $f : A \subset \mathbb{R}^k \rightarrow \mathbb{R}^k$ continuamente diferenciável numa vizinhança de z . Considere-se o operador $D_f(x)$ definido em (4)-(5), com $h(x) = \|f(x)\|_2^2 \neq 0$. Então,*

$$\lim_{x \rightarrow z} D_f(x) = J_f(z), \quad (7)$$

onde J_f designa a matriz jacobiana de f . Em (7), $\lim_{x \rightarrow z} D_f(x)$ tem o seguinte significado: trata-se da matriz dos limites das entradas da matriz $D_f(x)$, definida em (5), quando x tende para z .

Demonstração. Fixados $i, j \in \{1, 2, \dots, k\}$, denote-se por $u(x)$ a função $u(x) = x - h(x) e_j \Leftrightarrow x = u(x) + h(x) e_j$. Assim, quando x tende para z , tem-se $f(x) \rightarrow 0$, $h(x) \rightarrow 0$ e $u(x) \rightarrow z$. Atendendo a que

$$\begin{aligned} \delta_{i,j} f(x) &= \frac{f_i(x + h(x) e_j) - f_i(x - h(x) e_j)}{2 h(x)} \\ &= \frac{f_i(x + h(x) e_j) - f_i(x) + f_i(x) - f_i(x - h(x) e_j)}{2 h(x)} \\ &= \frac{1}{2} \left[\frac{f_i(x + h(x) e_j) - f_i(x)}{h(x)} + \frac{f_i(u(x) + h(x) e_j) - f_i(u(x))}{h(x)} \right], \end{aligned}$$

uma vez que f é continuamente diferenciável em z , existem os limites a seguir e é válida a igualdade

$$\lim_{x \rightarrow z} \delta_{i,j} f(x) = \frac{1}{2} \left[\lim_{x \rightarrow z} \frac{f_i(x + h(x) e_j) - f_i(x)}{h(x)} + \lim_{x \rightarrow z} \frac{f_i(u(x) + h(x) e_j) - f_i(u(x))}{h(x)} \right].$$

Por definição de derivada parcial de f no ponto z , obtém-se

$$\lim_{x \rightarrow z} \delta_{i,j} f(x) = \frac{1}{2} \left(\frac{\partial f_i(z)}{\partial x_j} + \frac{\partial f_i(z)}{\partial x_j} \right) = \frac{\partial f_i(z)}{\partial x_j}.$$

Dado que as igualdades anteriores ocorrem para qualquer índice i, j , é válida a relação (7). \square

O exemplo a seguir ilustra uma das vantagens computacionais do operador (4)-(5) e da correspondente função iteradora centrada (6), no caso particular de funções reais, isto é, para $k = 1$. Com efeito, apesar de no respectivo domínio a função f escolhida não ser diferenciável, tanto o operador $D_f(x)$ como $g(x)$ estão bem definidos. Em particular, no Exemplo (2.1) é tratada uma função f para a qual a função iteradora g tem por imagem a (única) raiz $z = 1$ da equação $f(x) = 0$ num certo intervalo $A \subset \mathbb{R}$.

Exemplo 2.1. *Sejam $k = 1$ e*

$$f(x) = |x - 1|, \quad A = [0, 2].$$

A função real f possui obviamente um único zero $z = 1$ no intervalo A . Por conseguinte, a função iteradora clássica de Newton $\mathcal{N}_f(x) = x - f(x)/f'(x)$

não está definida em z pelo que não deverá ser utilizada para aproximar o zero em causa. No entanto o correspondente operador diferenças centradas $D_f(x)$ está bem definido em A . De facto, para $k = 1$ e de acordo com (5), tem-se:

$$\begin{aligned} h(x) &= f^2(x) = |x - 1|^2, \\ D_f(1) &= 1, \\ D_f(x) &= \frac{|x + h(x) - 1| - |x - h(x) - 1|}{2h(x)}, \quad x \neq 1. \end{aligned}$$

Donde se conclui

$$D_f(x) = \begin{cases} -1, & 0 \leq x \leq 1 \\ 1, & 1 \leq x \leq 2. \end{cases}$$

A correspondente função iteradora centrada (6) escreve-se

$$g(x) = x - D_f^{-1} f(x) = 1 \quad \forall x \in A.$$

Assim, neste caso a função g pode ser considerada ideal no sentido de que, em primeiro lugar, a imagem por g de qualquer ponto do domínio A é precisamente o zero z procurado; em segundo lugar a função iteradora g está bem definida apesar da função f considerada não ser diferenciável no domínio considerado.

Uma vez que o operador em (4)-(5) está definido para todo o $x \in \mathbb{R}^k$, a função iteradora centrada g em (6) apenas não está definida em pontos de \mathbb{R}^k onde $D_f(x)$ é matriz singular. Ora, no que respeito ao nosso objectivo de separação dos zeros de uma função f num domínio A , interessa-nos que a função iteradora associada g esteja definida quando o argumento x está “próximo” de um zero $z \in A$ ou, equivalentemente, de um ponto fixo $z = g(z)$, independentemente da matriz $D_f(x)$ ser ou não singular nesse ponto x .

Assim, qualquer ponto de A para o qual a função g não esteja definida ou produza uma imagem mais afastada de z do que o ponto x de partida não nos interessa tendo em vista o nosso objectivo fundamental. Por conseguinte, a função g será modificada de modo a construir uma outra função iteradora definida em \mathbb{R}^k , que passamos a designar por $n(x)$, tal que:

- (i) À imagem $n(x)$ é atribuído o vector nulo sempre que $D_f(x)$ é matriz singular;
- (ii) À imagem $n(x)$ é atribuído o vector $y = g(x)$ no caso de serem satisfeitas uma ou mais condições que nos assegurem proximidade de y relativamente

a um certo ponto fixo z da função g , no domínio A (ponto fixo em geral desconhecido).

Uma função n nas condições (i) e (ii) oferece desde logo uma vantagem computacional relevante. Com efeito, existindo vários pontos fixos de g em A , a atribuição do vector nulo como referido em (i) tem desde logo a vantagem de automaticamente os separar. Como $n(x)$ apenas retém os pontos $x \in A$ considerados favoráveis mediante o critério (ii), a composição da nova função iteradora $n(x)$ consigo própria surge como o processo mais natural e económico de obtenção de novas funções iteradoras que se espera sejam mais interessantes do que a função iteradora de partida, no sentido das iteradoras compostas serem capazes de produzir globalmente e com pouco esforço computacional melhores aproximações dos pontos fixos z de g , em todo o domínio A . Além disso, como definimos $n(0) = 0$, o cálculo da função composta $n^r(x)$ apenas faz intervir os pontos favoráveis em A , pelo que a sua complexidade computacional será muito menor do que no caso da composição de funções iteradoras em geral. Funções iteradoras deste tipo, convenientemente ‘educadas’, capazes de separar zeros de funções reais num intervalo, são consideradas em [3].

Visto que para $x \in A$, em (4) é calculado $h(x) = \|f(x)\|_2^2$, um critério de proximidade como referido em (ii) consiste em assumir uma dada tolerância ϵ e definir como ponto favorável para a iteradora $n(x)$ qualquer ponto de A satisfazendo $\|f(x)\| \leq \epsilon$. Um outro critério favorável que iremos adoptar diz respeito à proximidade entre $x \in A$ e $y = g(x)$. Dado um *deslocamento máximo* $d > 0$ (por exemplo $d = 0.5$), consideramos $x \in A$ favorável para a iteradora $n(x)$ caso $y = g(x) \neq x$ (se $y = x$ então x é ponto fixo de g) e $\|y - x\| \leq d$. Isso significa que a função iteradora $n(x)$ apenas será não nula nos pontos x para os quais $y = g(x)$ se encontre a uma distância (para a norma considerada) não superior a d . Somos assim levados a definir a seguinte função iteradora $n : \mathbb{R}^k \rightarrow \mathbb{R}^k$ associada à função iteradora (6):

Definição 2.3. (*Função iteradora centrada associada a g*)

Dados $\epsilon > 0$ e $d > 0$ considere-se a função g definida em (6). A função iteradora associada $n(x)$ é assim definida:

$$\begin{aligned} n(0) &= 0 \\ n(x) &= \begin{cases} g(x), & \text{se } x \in A, g(x) \in \mathbb{R}^k, \|f(x)\| \leq \epsilon \text{ e } \|y - x\| \leq d \\ 0, & \text{caso contrário.} \end{cases} \end{aligned} \quad (8)$$

Em (8) o símbolo 0 em $n(0) = 0$ refere-se ao vector nulo e $\|\cdot\|$ denota uma qualquer norma em \mathbb{R}^k .

Na prática, não é necessário calcular a matriz inversa $D_f^{-1}(x)$ em (6), visto que o vector $g(x)$ em (8) pode ser obtido considerando

$$\begin{aligned} s(x) &= g(x) - x, \quad x \in A \\ D_f(x) s(x) &= -f(x). \end{aligned} \tag{9}$$

Assim, para $x \in A$, o vector $s(x)$ em (9) é solução do sistema linear $D_f(x) s(x) = -f(x)$. Existindo tal solução resulta $g(x) = x + s(x)$. No caso da matriz $D_f(x)$ ser singular o vector $g(x)$ em (9) não pode ser calculado (logo $g(x) \notin \mathbb{R}^k$), daí a atribuição $n(x) = 0$ nesse caso. Conclui-se que, tal como o operador $D_f(x)$ definido em (5), a função iteradora $n(x)$ calculada a partir do sistema linear referido em (9) está bem definida em \mathbb{R}^k , e por conseguinte a função composta $n^r(x) = (n \circ \dots \circ n)(x)$, para $r \geq 1$ está igualmente bem definida em \mathbb{R}^k . No programa *Mathematica* do Anexo 4 é dado código para a função centrada associada $n(x)$.

É claro que os critérios de proximidade (ii) anteriormente referidos só terão utilidade prática caso a função iteradora g possua propriedades adequadas tendo em vista a separação global e automática dos seus pontos fixos. Isso acontece quando, em particular, a função g é *contractiva* numa vizinhança de um determinado ponto fixo $z \in A$. Se g for contractiva numa vizinhança \mathcal{V}_z , sabemos desde logo que para pontos dessa vizinhança acontece $\|g(x) - z\| < \|x - z\|$, ou seja que $y = g(x)$ é atraído para o ponto fixo. Por conseguinte, para $x \simeq z$ espera-se que $\|g(x) - x\| < \|x - z\|$, caso em que fixados parâmetros adequados d (deslocamento) e ϵ (tolerância), a condição $\|g(x) - x\| < d$ é satisfeita por algum ponto favorável do domínio A . De igual modo, para um ponto x satisfazendo o critério anterior quando $x \simeq z$, $\|f(x)\| < \epsilon$ pelo que a função $n(x)$ produzirá uma aproximação do ponto fixo em causa.

Como é sabido, dada uma função $f \in C^2(\mathbb{R}^k)$, tal que numa vizinhança de um zero de f a matriz jacobiana $J_f(x)$ é não singular, o método de Newton clássico converge localmente e a convergência é supralinear ([5], Ch. 5), verificando-se portanto a contractividade desta função iteradora uma vez escolhidos pontos x suficientemente próximos de z . Nas mesmas condições pode provar-se que o método iterativo associado à função $n(x)$ dada em (8)-(9) goza dessa propriedade de convergência. Em particular, tal justifica podermos usar a função iteradora centrada (8) para separar zeros da função modelo (1) usada nesta nota bem como as iteradoras compostas dela resultantes, tal como é ilustrado na Secção 3.

(x, y)	$\ f(x, y)\ _\infty \leq 10^{-7}$
$(-3.8000751, -2.9564962)$	$3.76 * 10^{-8}$
$(-3.8000751, 7.5994545)$	$7.99 * 10^{-9}$
$(-3.8000751, 2.4005455)$	$7.93 * 10^{-9}$
$(-3.8000751, -0.060308236)$	$6.48 * 10^{-8}$
$(-3.5492605, 1.5574901)$	$9.13 * 10^{-9}$
\vdots	\vdots
$(7.8503105, 1.9699305)$	$7.76 * 10^{-8}$
$(7.9993853, 7.8533476)$	$2.05 * 10^{-11}$
$(7.9993853, 2.1466524)$	$2.02 * 10^{-11}$
$(7.9993853, -0.19530061)$	$3.85 * 10^{-11}$
$(7.9993853, 0.80877458)$	$1.32 * 10^{-9}$

Tabela 1: Aproximações de 274 zeros da função modelo e respectivos resíduos, após uma aplicação de $n^2(x)$.

3 Funções iteradoras aplicadas ao exemplo modelo

Retomemos a função modelo (1). Tanto para a localização dos seus zeros como para o cálculo aproximado de um número elevado desses zeros, iremos aplicar uma versão discreta da função iteradora centrada $n(x) = x - D^{-1}(x)f(x)$ definida por (8) e (9). A Figura 1 (à direita) resultou da aplicação da função composta $n^2(x) = (n \circ n)(x)$ a um conjunto de pontos do domínio $A = [-4, 8] \times [-4, 8]$. Fazendo $x_{min} = -4$, $x_{max} = 8$, $y_{min} = -4$, $y_{max} = 8$ e acréscimo $dx = dy = 0.05$, consideraram-se 58 081 pontos (x_i, y_i) em A , com $x_i = -4 + i dx$, $y_i = -4 + i dy$. A imagem pela função iteradora centrada $n^2(x)$ desses pontos foi calculada (cf. lista $t = ParallelTable[]$ na parte final do código dado no Anexo 4). Os pontos da lista das imagens calculadas por $n^2(x)$ aparecem a negro na referida figura e dão-nos uma indicação visual da distribuição de zeros de f no domínio considerado. Confirma-se assim a existência de um grande número de zeros de f , tal como é sugerido pelas curvas de nível que constam da Figura 1 à esquerda.

Com esforço computacional insignificante, pode verificar-se automaticamente que dos 58 081 pontos inicialmente considerados 56 702 têm por imagem o vector nulo, ou seja, não são pontos relevantes para a função centrada $n(x)$ relativamente aos parâmetros que foram adoptados nesta experiência, ou seja, um “deslocamento” $d = 0.5$ e tolerância $\epsilon = 0.1$.

Atendendo a que por defeito a função centrada $n(x)$ atribui o vector nulo sempre que x é o vector nulo, a complexidade algorítmica quanto ao cálculo da função composta n^2 fica reduzida apenas aos pontos favoráveis da função $n(x)$, em número de 1379, ou seja cerca de 2.4% dos pontos inicialmente considerados. Na versão 10.3 do sistema *Mathematica* e cálculos em precisão dupla (num computador pessoal 2GHz Intel Core i7), a referida lista de pontos imagem de $n^2(x)$ quando aplicada à lista de 58 081 pontos do domínio A foi calculada em menos de um segundo, aplicando o código dado a seguir. Extraíndo os pontos (x, y) não nulos satisfazendo o critério de resíduo $\|f(x, y)\|_\infty \leq 10^{-7}$ resultam 274 pontos do domínio A , os quais são zeros aproximados da função f , satisfazendo um tal critério. Na Tabela 1 apresentam-se os primeiros e últimos cinco pontos assim calculados bem como o respectivo resíduo para a norma $\|f(x, y)\|_\infty = \max\{|f_1(x, y)|, |f_2(x, y)|\}$.

Note-se que as rotinas *Reduce* e *FindInstance* do *Mathematica* foram substancialmente melhoradas na versão 11.0 do sistema (Agosto 2016). Todavia, o método aqui proposto é competitivo, conforme o leitor poderá comprovar experimentalmente.

4 Anexo (código *Mathematica*)

A nomenclatura adoptada no código *Mathematica* a seguir é semelhante à utilizada ao longo do texto. O operador $D(x)$ definido em (4)-(5) é aqui denotado por $\delta[x]$. Os comentários inseridos no código destinam-se a facilitar a sua compreensão.

Para resolver o sistema linear (9) é invocada a rotina *LinearSolve* do sistema. Embora apenas se use a dimensão $k = 2$, o mesmo código poderá ser facilmente modificado para funções $f : \mathbb{R}^k \rightarrow \mathbb{R}^k$, com $k > 2$.

```
k = 2; (* dimensão *)
id = IdentityMatrix[k]; (* matriz identidade *)
e[j_] := id[[j]]; (* j-ésimo vector canónico *)
null = Table[0, {k}]; (* vector nulo *)
g[i_][x_List] := f[x][[i]]; (* i-ésima componente da função f *)
```

```

(* operador diferenças centradas generalizadas: *)
δ[{0, 0}] = id;
δ[x_List] := Block[{xx, fx, h, hj, fxi, v},
xx = N[x]; fx = f[xx]; h = fx.fx;
v = 1/(2h) Table[hj = he[j]; g[i][x + hj] - g[i][x - hj], {i, k}, {j, k}];
If[N[fx ≠ null], v, id];

(* iteradora diferenças centradas : *)
n[{0, 0}] = null;
n[x_List] := n[x] = Quiet[Block[{xx, res, y},
xx = N[x]; (* res= solução sistema linear: *)
res = LinearSolve[δ[xx], -f[xx]];
If[VectorQ[res], y = xx + res, y = null]; (* caso δ[x] singular *)

Which[xx ≠ null && xx = N[y], y, (* y é ponto fixo máquina *)
VectorQ[res] == False, null,
(* vector y fora do domínio A: )*
y[[1]] < xmin || y[[1]] > xmax, null,
y[[2]] < ymin || y[[2]] > ymax, null,
Norm[y - xx, Infinity] ≤ 10-3, y, (* xx é ponto fixo aprox. *)
(* d= parâmetro deslocamento; ε= tolerância : *)
Norm[res, Infinity] ≤ d && Norm[f[y], Infinity] ≤ ε, y,
True, null ]];

```

Aplicação à função modelo (1):

```

f[{x_, y_}] := {Sin[4(x - 5)2 + 1/3(y - 5)2 - 1],
Cos[-3/2(x - 4)2 + 4/3(y - 5)2 - 1]};
xmin = -4; xmax = 8; ymin = -4; ymax = 8;
dx = dy = 0.05; d = 0.5; ε = 10-1;
t = ParallelTable[n[n[{x, y}]], {x, xmin, xmax, dx}, {y, ymin, ymax, dy}];

```

Referências

- [1] J. E. Dennis and J. J. Moré, Quasi-Newton methods, motivation and theory, *SIAM Review*, (1977), 46-89.
- [2] J. E. Dennis, R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, *SIAM*, Philadelphia, 1983.

- [3] M. M. Graça, Maps for global separation of roots, *Electronic Trans. Numer. Anal.* (2016), 241-256.
- [4] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, John Wiley and Sons, New York, 1966.
- [5] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.
- [6] W. C. Rheinboldt, *Methods for Solving Systems of Nonlinear Equations*, SIAM, 2nd ed., Philadelphia, 1998.
- [7] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, 1964.
- [8] S. Wolfram, *The Mathematica Book*, Wolfram Media, fifth ed., 2003.