nos diferentes valores x que não sejam overflow/underflow.

Numa máquina binária com precisão p e que faz arredondamentos ao mais próximo, a unidade de arredondamento é então $u = 2^{-p}$.

Outro valor que produz informação análoga à precisão da máquina é o conhecido como épsilon da máquina.

Definição 1.31. *Chama-se **épsilon da máquina** num sistema de cômputo em ponto flutuante o menor número positivo ϵ representável na máquina e tal que:*

$$\boxed{1 \oplus \epsilon > 1}$$

onde $1 \oplus \epsilon$ quer dizer a soma de 1 e ϵ tal como feita pela máquina para números em ponto flutuante, normalmente $fl(1+\epsilon)$.

1.4 Propagação do erro

A noção de valor aproximado pode entender-se também para funções aproximadas ou algoritmos.

Pensemos num problema matemático com solução perfeitamente determinada mas dependente de parâmetros de entrada. Por exemplo, pensemos no problema de encontrar a menor das raízes reais do polinómio $x^2 + b \cdot x + c$. Sabemos que existe uma solução deste problema, e que está determinada pela aplicação da fórmula resolvente:

$$\text{Menor raiz de } x^2 + b \cdot x + c \text{ é } z = \frac{-b - \sqrt{b^2 - 4c}}{2}$$

A solução do problema matemático está assim descrita por $z = f(b, c)$ onde f é uma função nos parâmetros b, c que determinam o problema.

Se temos um sistema numérico com finitos números e onde existe um procedimento para somar, multiplicar, dividir, e extrair raízes destes números em forma aproximada, poderíamos criar um algoritmo f^* como aproximação da fórmula resolvente, ou seja uma série de instruções que:

- Precisa da introdução de dois valores b^*, c^* existentes no sistema numérico.
- Através das instruções programadas, produz como resultado um valor f^*
- O valor f^* devolvido irá ser usado como aproximação de $f(b, c)$, sempre que (b^*, c^*) sejam considerados aproximação de (b, c)

Assim o problema matemático tem parâmetros de entrada x (no caso anterior $x = (b, c)$), e conhecido o parâmetro existe uma solução determinada (aqui $f(x) = \frac{-b-\sqrt{b^2-4c}}{2}$).

No computador temos um algoritmo $f^* \colon x^* \mapsto y^*$ que admite como parâmetros de entrada valores x^* representáveis no computador e que consegue devolver um valor $y^* = f^*(x^*)$ representável no computador.

Muitas questões na matemática exigem a utilização de determinados valores (parâmetros) Suponhamos que temos um problema a resolver, que este problema depende de parâmetros numéricos e que este problema é resolvido através da função $f \colon x \to f(x)$. Suponhamos que conhecemos um algoritmo numérico $f^* \colon x^* \mapsto f^*(x^*)$.

Existem duas questões pertinentes para ver se f^* é bom para representar f:

- Se tivéssemos parâmetros representáveis em forma exata no computador $x = x^*$, será que o valor devolvido pelo algoritmo $f^*(x^*)$ é a melhor aproximação existente de $f(x^*)$? Será que o computador não consegue distinguir $f(x^*)$ de $f^*(x^*)$? Ou seja: $f^*(x^*) = fl(f(x^*))$ para qualquer x^* do sistema numérico usado?

- Se consideramos a solução $f^*(x^*)$, será que esta solução resulta ser a solução exata $f(x)$ para algum valor concreto x que o computador não distingue de x^*? Ou seja: $x^* = fl(x)$ para algum x que satisfaz $f(x) = f^*(x^*)$?

A primeira questão está referida a uma análise do erro direta para o algoritmo f^*. A segunda questão está referida a uma análise do erro inversa para o algoritmo f^*.

Em qualquer dos casos indicados, devemos ficar satisfeitos. O sistema numérico do computador não vai permitir, por muito bom que for o algoritmo, encontrar melhor representação de f do que a dada através de f^*

Se partimos dum valor numérico x^* e dum algoritmo f^*, tomar $f^*(x^*)$ como aproximação de $f(x)$ contem um erro $\Delta(f^*(x^*), f(x))$. Podemos decompor este em duas partes, uma devida à aproximação de a e outra à aproximação de f:

$$\boxed{(f^*(a^*) - f(a)) = \underbrace{(f^*(a^*) - f(a^*))}_{(1)} + \underbrace{(f(a^*) - f(a))}_{(2)}}$$

Nesta soma a primeira parte conhece-se como **erro de computação associado ao algoritmo** (erro devido a que o algoritmo não produz $f(a^*)$ nem quando $a^* = a$), e a segunda parte como **erro propagado por** f (erro não relacionado com o algoritmo f^* senão com o comportamento de f ao usar um valor aproximado de partida).

Definição 1.32. *Diremos que uma função f de variável real x é uma* ***função bem condicionada*** *no cálculo de f no valor a se*

$$\Delta(a^*, a) \text{ pequeno} \Rightarrow \Delta(f(a^*), f(a)) \text{ pequeno}$$

Do ponto de vista do programador, este pode tratar de criar um algoritmo estável para o cálculo de f, manter sob controlo o erro de computação através duma boa escolha do algoritmo. Por outra parte o erro propagado não depende dele, senão da função matemática $f(x)$ dada no modelo matemático, e do valor aproximado a^* que pode ter erros com respeito do valor exato a.

Um elemento destacado no estudo do erro propagado é a noção de **aproximação linear** duma função $f(x)$ num ponto $a \in \mathbb{R}^n$.

Definição 1.33. *Consideremos um ponto $a \in \mathbb{R}^n$ e uma função $f(x)$ definida em todos os pontos de $\mathbb{R}^n$ que satisfazem $\|x - a\| \leq \epsilon$ (bola de raio ϵ centrada em a). Consideremos as possíveis* ***funções afins*** *em $\mathbb{R}^n$ (funções do tipo $p(x) = p(x_1, \ldots, x_n) = c_0 + c_1x_1 + c_2x_2 + \ldots + c_nx_n$, polinomiais de grau 1). Diremos que $p(x)$ é a* ***linearização de*** *$f(x)$* ***no ponto*** *a se $f(a) = p(a)$ e todas as sucessões $x(k) \neq a$ com limite a satisfazem:*

$$\lim_{k\to\infty} \frac{p(x(k)) - f(x(k))}{\|x(k) - a\|} = 0$$

Neste caso diremos que $f(x)$ é ***diferenciável no ponto*** *a, sendo $p(x)$ a sua aproximação linear.*

A condição do limite acima é uma forma de dizer que $p(x)$ é similar a $f(x)$. Indicaremos neste caso $f(x) \simeq p(x)$, ou com maior precisão:

$$f(x) = p(x) + o(x - a)$$

Definição 1.34. *Escrever $f(x) = g(x) + o((x-a)^q)$ irá significar, no sucessivo, que $f(a) = g(a)$ verificando ainda as sucessões $x(k) \neq a$ convergentes em a a seguinte propriedade:*

$$\lim_{k\to\infty} \frac{p(x(k)) - f(x(k))}{\|x(k) - a\|^q} = 0$$

A condição imposta exige em particular $p(a) = f(a)$. A linearização $p(x)$ tem assim a propriedade de coincidir com $f(x)$ no ponto a, e ainda se nos aproximarmos de a (através duma sucessão $x(k)$ de limite a), o erro de $p(x)$ como aproximação de $f(x)$ é pequeno (inferior a qualquer constante escolhida, multiplicada com $\|x(k) - a\|$).

Proposição 1.35. *Se $p(x)$ é a linearização de $f(x)$ no ponto a, então:*

$$\boxed{p(x) = f(a) + \partial_1 f(a) \cdot (x_1 - a_1) + \partial_2 f(a) \cdot (x_2 - a_2) + \ldots + \partial_n f(a) \cdot (x_n - a_n)}$$

onde

$$\partial_i f(a) = \lim_{h \to 0} \frac{f(a_1, \ldots, a_i + h, \ldots, a_n) - f(a)}{h}$$

é chamada a ***derivada parcial da função f, na componente i, no ponto a.***

Prova: Consideremos o ponto $a = (a_1, \ldots, a_n)$. Resulta simples ver que qualquer função afim pode ser escrita na forma: $p(x) = c_0 + c_1 \cdot (x_1 - a_1) + \ldots + c_n \cdot (x_n - a_n)$, para alguma escolha de constantes c_i.

Tratemos de determinar as componentes $c_0, c_1, \ldots, c_n$ no caso em que $p(x)$ seja linearização de $f(x)$ no ponto a. Como exigimos $f(a) = p(a)$ temos necessariamente $c_0 = f(a)$

Queremos ainda provar que que $c_i = h_i'(a_i)$, onde usamos a função real de variável real $h_i(t) = f(a_1, \ldots, t, \ldots a_n)$, cuja derivada em a_i é precisamente a definição de $\partial_i f(a)$

Fixamos nossa atenção num i concreto. Provar que $h_i'(a_i) = c_i$ é o mesmo que provar

$$\lim_{t \to a_i} \frac{h_i(t) - h_i(a_i)}{t - a_i} = c_i$$

ou seja provar que para cada sucessão $t(k) \neq a_i$ convergente em a_i temos:

$$\lim_{k \to \infty} \frac{h_i(t(k)) - h_i(a_i)}{t(k) - a_i} = c_i$$

Provar esta igualdade não é difícil. Tomamos a sucessão de pontos $x(k) = (a_1, \ldots, t(k), \ldots, a_n)$, todos com a mesma componente em cada posição, salvo no lugar i.Temos neste caso $\|(x(k) - a)\| = \|(0, \ldots, t(k) - a_i, 0, \ldots, 0)\| = |t(k) - a_i| \cdot \|(0, \ldots, 1, 0, \ldots, 0)\| = |t(k) - a_i| \cdot s_i$ onde $s_i \neq 0$ é uma constante, a norma do vetor $(0, \ldots, 1, 0, \ldots, 0)$.

Temos em particular que $x(k)$ vai ser uma sucessão que nunca toma valor a (porque $t(k) \neq a_i$), mas tem limite a (porque $|t(k) - a_i|$ tem limite 0)

Mais ainda, devido à definição de h_i temos $f(x(k)) = h_i(t(k))$, $p(x(k)) = f(a) + c_i \cdot (t(k) - a_i) = h_i(a_i) + c_i \cdot (t(k) - a_i)$ e portanto como p é a linearização de f no ponto a temos:

$$0 = \lim_{k \to \infty} \frac{p(x(k)) - f(x(k))}{\|x(k) - a\|} = \lim_{k \to \infty} \frac{h_i(a_i) + c_i \cdot (t(k) - a_i) - h_i(t(k))}{|t(k) - a_i| \cdot s_i} =$$
$$= \lim_{k \to \infty} \left(c_i - \frac{h_i(t(k)) - h_i(a_i)}{t(k) - a_i} \right) \cdot \frac{t(k) - a_i}{|t_k - a_i| \cdot s_i}$$

Tendo em conta que o termo a multiplicar à direita é sempe $\frac{\pm 1}{s_i}$, o limite indicado só pode ser zero se as sucessões $t(k)$ indicadas satisfazem:

$$\lim_{k \to \infty} \frac{h_i(t(k)) - h_i(a_i)}{t(k) - a_i} = c_i$$

o qual é precisamente a definição para $\lim_{t \to a_i} \frac{h_i(t) - h_i(a_i)}{t - a_i} = c_i$, cada valor c_i é assim a derivada no ponto $t = a_i$ da função $h_i(t)$, como indicava a proposição. □

Nota 1.36. As derivadas parciais podem existir e no entanto o polinómio $p(x)$ indicado não satisfazer a condição para ser uma linearização de $f(x)$, ou seja, pode que a função não seja diferenciável. No entanto é sabido que quando as derivadas parciais existem em todos os pontos (não só em a), e são contínuas, o polinómio $p(x)$ indicado satisfaz sim a condição para ser uma linearização de $f(x)$ no ponto a, sendo portanto f diferenciável no ponto a.

Nota 1.37. A derivada parcial $\partial_i f(a)$ também é representada por $\frac{\partial f}{\partial x_i}(a)$, e coincide com o valor da derivada em $a_i \in \mathbb{R}$ da função real $f(a_1, \ldots, a_{i-1}, t, a_{i+1}, \ldots, a_n)$ na variável real t.

O cálculo da derivada parcial na componente i num ponto $(a_1, \ldots, a_n)$ exige assim considerar **quase todos** os valores $a_1, \ldots, a_n$ como **parâmetros fixos**, e o valor a_i como único variável, determinando então a derivada nesta variável. Podem assim aplicar-se todas as técnicas de derivação de funções reais de variável real.

Por exemplo, para $f(b,c) = \frac{-b-(b^2-4c)^{1/2}}{2}$, a derivada parcial na componente b seria:

$$\frac{\partial f}{\partial b}(b,c) = \frac{-1 - b \cdot (b^2-4c)^{-1/2}}{2} = \frac{-b - \sqrt{b^2-4c}}{2\sqrt{b^2-4c}}$$

e na componente c seria:

$$\frac{\partial f}{\partial c}(b,c) = \frac{2 \cdot (b^2-4c)^{-1/2}}{2} = \frac{1}{\sqrt{b^2-4c}}$$

O conjunto das derivadas parciais pode ser recolhido numa sequência

$$\boxed{\nabla_a f = (\partial_1 f(a), \partial_2 f(a), \ldots, \partial_n f(a)) \in \mathbb{R}^n}$$

o chamado **vetor gradiente de f no ponto a**.

Se conhecemos $f(a)$ e $\nabla_a f$, a linearização de f no ponto a é a seguinte:

$$p(x) = f(a) + (\nabla_a f) \cdot (x-a), \qquad f(x) = f(a) + (\nabla_a f) \cdot (x-a) + o(x-a)$$

Definição 1.38. *Diz-se que **um problema que depende de n parâmetros** e que é resolvido por uma função $f\colon \mathbb{R}^n \to \mathbb{R}$ está **bem condicionado** se a função f estiver bem condicionada, se variações pequenas nos dados introduzidos levam a variações pequenas nos resultados obtidos:*

Desde o ponto de vista do erro absoluto:

$$\|a^* - a\| \text{ pequeno} \Rightarrow |f(a^*) - f(a)| \text{ pequeno ?}$$

O coeficiente com que esta noção de "pequeno" é medida é o valor K que satisfaz:

$$|f(a^*) - f(a)| < K \cdot \|a^* - a\|$$

para valores de $\|a^* - a\|$ pequenos.

Desde o ponto de vista do erro relativo:

$$\frac{\|a^* - a\|}{\|a\|} \text{ pequeno} \Rightarrow \frac{|f(a^*) - f(a)|}{|f(a)|} \text{ pequeno ?}$$

que também pode ser medido com um valor κ que satisfaz:

$$\frac{|f(a^*) - f(a)|}{|f(a)|} < \kappa \cdot \frac{\|a^* - a\|}{\|a\|}$$

para valores de $\delta = \frac{\|a^* - a\|}{\|a\|}$ pequenos.

Pensemos num ponto a e em todas as aproximações a^* onde o erro absoluto seja $\epsilon = \Delta(a^*, a) > 0$. Estas aproximações são todos os pontos da forma $a^* = a + \epsilon \cdot u$ com u unitário. Então se substituimos f pela sua linearização no ponto a temos:

$$\Delta(f(a^*), f(a)) \simeq |f(a) + (\nabla_a f) \cdot (a^* - a) - f(a)| = \epsilon \cdot |(\nabla_a f) \cdot u|$$

Se agora aplicamos a propriedade $|x \cdot y| \leq \|x\|_p \cdot \|y\|_q$ e tirando os termos quadráticos em ϵ, o máximo erro que podemos cometer é:

$$\begin{array}{c} \Delta_2(a^*, a) = \epsilon \Rightarrow |f(a^*) - f(a)| \leq \epsilon \cdot \|\nabla_a f\|_2 \\ \Delta_1(a^*, a) = \epsilon \Rightarrow |f(a^*) - f(a)| \leq \epsilon \cdot \|\nabla_a f\|_\infty \\ \Delta_\infty(a^*, a) = \epsilon \Rightarrow |f(a^*) - f(a)| \leq \epsilon \cdot \|\nabla_a f\|_1 \end{array}$$

(estas desigualdades são salvo termos em ϵ^2)

Pensemos agora em $a \neq 0$ com $f(a) \neq 0$ e nas aproximações a^* onde o erro relativo seja $\delta = \delta(a^*, a) > 0$. Nestas aproximações o erro absoluto é $\delta \cdot \|a\|$ e portanto estamos a falar de pontos da forma $a^* = a + \delta \cdot \|a\| \cdot u$ com u unitário. Novamente se substituímos f pela sua linearização no ponto a temos:

$$\delta(f(a^*), f(a)) \simeq \frac{|f(a) + (\nabla_a f) \cdot (a^* - a) - f(a)|}{|f(a)|} = \frac{\delta \cdot \|a\| \cdot |(\nabla_a f) \cdot u|}{|f(a)|}$$

Assim tirando os termos quadráticos em δ, o máximo erro que podemos cometer é:

$$\begin{array}{c} \delta_2(a^*, a) = \delta \Rightarrow \delta(f(a^*), f(a)) \leq \delta \cdot \dfrac{\|a\|_2 \cdot \|\nabla_a f\|_2}{|f(a)|} \\ \delta_1(a^*, a) = \delta \Rightarrow \delta(f(a^*), f(a)) \leq \delta \cdot \dfrac{\|a\|_1 \cdot \|\nabla_a f\|_\infty}{|f(a)|} \\ \delta_\infty(a^*, a) = \delta \Rightarrow \delta(f(a^*), f(a)) \leq \delta \cdot \dfrac{\|a\|_\infty \cdot \|\nabla_a f\|_1}{|f(a)|} \end{array}$$

(estas desigualdades são salvo termos em δ^2)

Para qualquer função $f(x)$ diferenciável num ponto a, para qualquer valor $p \in [1, +\infty]$ e para o valor complementar (aquele q com $\frac{1}{p} + \frac{1}{q} = 1$) temos:

$$\begin{array}{c} \Delta(f(a^*), f(a)) \leq \|\nabla_a f\|_q \cdot \Delta_p(a^*, a) \\ \delta(f(a^*), f(a)) \leq \dfrac{\|a\|_p \cdot \|\nabla_a f\|_q}{|f(a)|} \cdot \delta_p(a^*, a) \end{array}$$

Estas fórmulas são chamadas as **fórmulas aproximadas da propagação de erros numa vizinhança dum ponto**, e devem ser interpretadas salvo termos da ordem $(\delta_p(a^*, a))^2$

Os números de condição estudados foram obtidos a partir duma linearização. Assim temos desigualdades no limite, quando nos aproximamos dum ponto a, e válidas salvo termos quadráticos.

Vamos tratar de dar uma fórmula que permita majorar o erro de maneira real, sem deixar de lado termos quadráticos, e portanto sem termos que ficar próximos dum ponto a.

Consideremos dois pontos a, a^* de $\mathbb{R}^n$ quaisquer. Consideremos um conjunto $I \subseteq \mathbb{R}^n$, produto de intervalos e que contém a, a^*, e de maneira que as derivadas parciais de f existem não só no ponto a senão em todos os pontos $x \in I$. Então o teorema de Lagrange permite provar o seguinte resultado:

Proposição 1.39 (Fórmula fundamental da propagação do erro absoluto). *Seja $I \subset \mathbb{R}^n$ um produto de intervalos, seja $f(x)$ uma função definida em I, onde sabemos que existem*

as derivadas parciais $\partial_i f(x)$ em cada ponto $x \in I$. Se a, a^* ão pontos de I, existem pontos $p_i \in I$ onde:

$$\begin{aligned} f(a^*) - f(a) = &\partial_1 f(p_1) \cdot (a_1^* - a_1) + \\ &\partial_2 f(p_2) \cdot (a_2^* - a_2) + \\ &\partial_3 f(p_3) \cdot (a_3^* - a_3) + \\ &\dots \\ &\partial_n f(p_n) \cdot (a_n^* - a_n) \end{aligned}$$

Em particular se existe um valor M que serve como limite superior de todos os valores $|\partial_i f(p)|$ em todos os pontos $p \in I$ indicados, tem-se:

$$\Delta(f(a^*), f(a)) \leq M \cdot (|a_1^* - a_1| + |a_2^* - a_2| + \ldots + |a_n^* - a_n|)$$

Observe que a soma indicada representa $\Delta_1(a^*, a) = \|a^* - a\|_1$, e se chamamos $\|\nabla f\|_I$ o valor supremo dos possíveis valores $|\partial_i f(p)|$ nos pontos $p \in I$ temos:

$$\boxed{\Delta(f(a^*), f(a)) \leq \|\nabla f\|_I \cdot \Delta_1(a^*, a)}$$

Demonstração. Basta escrever:

$$\begin{aligned} f(a^*) - f(a) = &f(a_1^*, a_2, \dots, a_n) - f(a_1, a_2, \dots, a_n) + \\ &f(a_1^*, a_2^*, a_3, \dots, a_n) - f(a_1^*, a_2, \dots, a_n) + \\ &f(a_1^*, a_2^*, a_3^*, \dots, a_n) - f(a_1^*, a_2^*, a_3, \dots, a_n) + \\ &\dots + \\ &f(a_1^*, a_2^*, \dots, a_n^*) - f(a_1^*, a_2^*, \dots, a_{n-1}^*, a_n) \end{aligned}$$

e aplicar em cada uma destas diferenças o teorema do valor médio de Lagrange, para a função $h(t) = f(a_1^*, \dots, t, \dots, a_n)$ no intervalo de valores t limitado por a_i^* e por a_i. □

A fórmula indicada proporciona um limite superior para os erros absolutos cometidos. Uma desvantagem é que para ser aplicada a fórmula seria necessário encontrar um majorante das derivadas parciais e no fim temos um erro não é "preciso", sendo possível que o erro esteja limitado por valores ainda menores do que o indicado. Uma vantagem é que esta fórmula não faz intervir termos do tipo $o(\epsilon)$,

Quando trabalhamos com arredondamentos a^* com erros pequenos respeito dum valor a, aplicar f supõe obter valores $f(a^*)$ que, vistos como arredondamento de $f(a)$, levam um erro proporcional ao erro original. A constante de proporcionalidade é chamada número de condição.

Definição 1.40. *Chamamos **número de condição duma função f para o erro absoluto**, medido com respeito duma norma $\|\cdot\|$ o valor:*

$$\lim_{d \to 0^+} \sup_{\Delta(a^*, a) = d} \frac{\Delta(f(a^*), f(a))}{\Delta(a^*, a)}$$

*Chamamos **número de condição duma função f para o erro relativo**, medido com respeito duma norma $\|\cdot\|$ o valor:*

$$\lim_{\delta \to 0^+} \sup_{\delta(a^*, a) = \delta} \frac{\delta(f(a^*), f(a))}{\delta(a^*, a)}$$

Se a função for diferenciável no ponto a e usamos a norma-p, estes números de condição são, respetivamente, os valores

$$K_p(a) = \|\nabla_a f\|_q$$
$$\kappa_p(a) = \frac{\|a\|_p \cdot \|\nabla_a f\|_q}{|f(a)|}$$

Quando se tem a^* um valor aproximado de a com erro relativo δ, a solução proporcionada por f pode ter um erro relativo de até $\kappa \cdot \delta$. A resolução do problema produz resultados onde o erro relativo vê-se multiplicado por κ respeito do erro dos dados introduzidos.

Se $\kappa = 8$, valores de entrada x^* com precisão de p algarismos binários significativos produzem resultados de saída $f(x)$ que podem ter até 8 vezes o erro relativo original, portanto com uma precisão de menos 3 algarismos binários significativos (porque o erro relativo poderia ficar multiplicado com $8 = 2^3$)

A função $f(x)$ diz-se **mal condicionada** num ponto a se o número de condição neste ponto for grande. Para problemas mal condicionados não importa o algoritmo programado, devemos esperar a possibilidade de que com dados de entrada muito precisos possamos recuperar valores de saída muito imprecisos.

O número de condição também pode ser estudado componente a componente (de maneira similar às derivadas parciais). Se trabalhamos com f num ponto a e consideramos a componente a_i como variável, temos uma função $f(a_1, \ldots, t, \ldots, a_n)$ numa única variável t que irá ter um número de condição no ponto a_i dado por:

$$\kappa_{[i]}(a) = \frac{|a_i| \cdot |\partial_i f(a)|}{|f(a)|}$$

Se queremos estudar a soma indicada na proposição antes dada componente a componente, também podemos usar os valores $\|\delta_i f\|_I$, supremos de $\partial_i f(p)$ em todos os pontos $p \in I$ e escrever:

$$\Delta(f(a^*), f(a)) \leq \|\partial_1 f\|_I \cdot |a_1^* - a_1| + \ldots + \|\partial_n f\|_I \cdot |a_n^* - a_n|$$

Ao dividir entre $|f(a)|$ e tendo em conta $|a_i^* - a_i| = |a_i| \cdot \delta(a_i^*, a_i)$ podemos também escrever:

$$\delta(f(a^*), f(a)) \leq \kappa_{[1]}(I) \cdot \delta(a_1^*, a_1) + \ldots + \kappa_{[n]}(I) \cdot \delta(a_n^*, a_n)$$

onde $\kappa_{[i]}(I) = \sup_{p \in I} \frac{|a_i| \cdot |\partial_i f(p)|}{|f(a)|}$.

Esta fórmula de propagação do erro pode resultar útil se os erros relativos conhecidos nos dados de entrada são diferentes duma componente para outra, ou se os números de condição da função, em cada componente, são muito diferentes duma componente para outra. Assim:

$$\delta(a_i^*, a_i) \leq \delta \Rightarrow \delta(f(a^*), f(a)) \leq \delta \cdot \sum \kappa_{[i]}(I)$$

$$\kappa_{[i]}(I) \leq \kappa \Rightarrow \delta(f(a^*), f(a)) \leq \kappa \cdot \sum \delta(a_i^*, a_i)$$

Exemplo

Consideremos a função $f(b,c) = \frac{-b-\sqrt{b^2-4c}}{2}$, fórmula resolvente que determina a menor das raízes no polinómio $x^2 + b \cdot x + d$.

Quando introduzimos os valores (b,c) num computador, se usamos a norma-∞ sabemos que o par (b,c) é introduzido na forma $(b^*, c^*) = (fl(b), fl(c))$, com um erro relativo não superior a 2^{-p} (onde p é a precisão). Podemos esperar que $f(b^*, c^*)$ seja uma boa aproximação da raiz $f(b,c)$? Depende de qual é o número de condição relativo (com a norma ∞) da função f no ponto (b,c).

$$\frac{\partial f}{\partial b}(b,c) = \frac{-b - \sqrt{b^2-4c}}{2\sqrt{b^2-4c}}, \qquad \frac{\partial f}{\partial c}(b,c) = \frac{1}{\sqrt{b^2-4c}}$$

$$\|\nabla_{(b,c)} f\|_1 = \left| \frac{-b - \sqrt{b^2-4c}}{2\sqrt{b^2-4c}} \right| + \left| \frac{1}{\sqrt{b^2-4c}} \right| = \frac{2 + b + \sqrt{b^2-4c}}{2\sqrt{b^2-4c}}$$

(nesta última igualdade assumimos $b > 0$ para poder retirar o valor absoluto, dado que $b > \sqrt{b^2-4c} > 0$)

Nesta situação temos como número de condição com a norma-∞ o seguinte:

$$\kappa_\infty(b,c) = \frac{2(2+b+\sqrt{b^2-4c})}{2\sqrt{b^2-4c}(b+\sqrt{b^2-4c})} \cdot \max(|b|,|c|) = \frac{2+b+\sqrt{b^2-4c}}{b^2-4c+b\sqrt{b^2-4c}} \cdot \max(|b|,|c|)$$

Exemplo

Pensemos no caso particular da procura da maior raiz no polinómio $x^2/2+a\cdot x+5=0$, dependente do parâmetro $a \in \mathbb{R}$, e cuja solução está dada pela função $f(a) = -a + \sqrt{a^2-10}$. Sempre que a esteja um pouco longe da $\sqrt{10}$ (digamos $10/a^2$ pequeno), tem valores de κ pequenos:

$$k = \left|\frac{a \cdot f'(a)}{f(a)}\right| = \left|\frac{-a}{\sqrt{a^2-10}}\right| = \frac{1}{\sqrt{1-(10/a^2)}} \simeq 1$$

Para estes valores de a, o nosso é um problema bem condicionado. Nas cercanias de $\sqrt{10}$, no entanto, o problema está mal condicionado. Calcular a raiz mais elevada do polinómio $x^2/2 + a \cdot x + 5 = 0$, para valores a próximos de $\sqrt{10}$ pode levar a resultados muito diferentes, com uma pequena alteração de a. Assim, para $a = 3.16229$ a raiz do polinómio é aproximadamente -3.1534557, para $a = 3.16228$ a raiz do polinómio é aproximadamente -3.158433 (observamos que só foi alterado o sexto algarismo significativo, e o valor do resultado tem um erro no terceiro algarismo significativo), e para $a = 3.16227$, nem sequer existem raízes reais deste polinómio.

Análise de erros do algoritmo

Temos estudado a parte do erro $\Delta(f(a^*), f(a))$, devida ao arredondamento de a através de a^*. Este é um erro devido à natureza do problema, e não tem nada a ver com o algoritmo utilizado na sua resolução numérica. Há outra componente de erro devida ao algoritmo numérico: $\Delta(f^*(a^*), f(a^*))$.

De maneira imprecisa diremos que um algoritmo é estável se erros pequenos nos dados introduzidos levam a erros pequenos nos resultados numéricos obtidos através do algoritmo.

O algoritmo não é uma função definida sobre todos os números reais, senão só sobre aqueles representáveis na máquina. Portanto, uma caracterização mais precisa desta definição exige um estudo diferente do que o feito para o condicionado do problema.

Esta componente $\Delta(f^*(a^*), f(a^*))$ é uma fonte de erros e podemos pretender limitar os mesmos. A análise do erro introduzido pelo algoritmo pode ser feito desde dois pontos de vista: a análise de erros direta e a análise de erros inversa.

Análise de erros direta

A análise de erros direta dum algoritmo é similar a análise já feita. é o estudo de qual é o erro (absoluto ou relativo) cometido ao tomarmos $f^*(a^*)$ como aproximação de $f(a^*)$:

$$\delta(f^*(a^*), f(a^*)) = \left|\frac{f^*(a^*) - f(a^*)}{f(a^*)}\right|$$

Um algoritmo será tanto melhor se este valor é sempre menor do que a unidade de arredondamento u. De facto, se para todo o a^* representável na máquina o algoritmo satisfaz:

$$f^*(a^*) = f(a^*) \cdot (1+\delta), \qquad \delta < u$$

então a substituição de f pelo algoritmo f^* não produz erro nenhum que não fosse já um erro intrínseco do sistema de arredondamento em ponto flutuante (o sistema não sabe distinguir $f^*(a^*)$ de $f(a^*)$ se $\delta(f^*(a^*), f(a^*)) < u$). Podemos assumir que f^* produz como resultado precisamente o mesmo do que f. Se $\delta < u$ ou δ é duma ordem próxima a u, diz-se que o algoritmo é estável (desde o ponto de vista da análise de erros direta).

Análise de erros inversa

Nesta perspetiva, a pergunta que fazemos não é se $f^*(a^*)$ é aproximadamente igual do que $f(a^*)$ para o sistema de arredondamento da máquina, senão que nos perguntamos se $f^*(a^*)$ é exatamente $f(a)$ para algum a que a máquina percebe como sendo aproximadamente igual do que a^* (isto é, um a tal que $a = a^* \cdot (1+\delta)$ com $\delta < u$).

Exemplo

Suponhamos que temos o polinómio dependendo de a já estudado $x^2/2+a\cdot x+5$ e que procuramos, para cada valor de a, qual é a raiz maior do polinómio. Suponhamos que a resposta é obtida, para cada a^*, através dum algoritmo que devolve:

$$f^*(a^*) = \frac{2\cdot(20\cdot(a^*)^2 - 139\cdot a^* - 205)}{729}$$

(este é o polinómio de Taylor da função $f(a)$ centrado em $11/2$ e truncado no terceiro passo)

Se queremos estudar o erro cometido pelo algoritmo em $a^* = 5.4$, o método direto diz-nos:

$$\delta = \delta(f^*(5.4), f(5.4)) = 1.08796\cdot 10^{-3}$$

O método inverso exige computar um valor a com $f^*(5.4) = f(a)$, que é $f^*(5.4) = f(5.40476802)$. O erro relativo segundo esta perspetiva é

$$\delta(a^*, a) = \delta(5.4, 5.40476802) = 8.82967\cdot 10^{-4}$$

A análise de erros inversa desenvolveu-se com posterioridade à análise de erros direta e permitiu determinar como, ainda que certos algoritmos pareciam fracos quando se estudava o seu erro pelo método direto, o estudo com o método inverso permite justifica-los.

Se o valor de δ com o método inverso é menor do que a unidade de arredondamento u, podemos afirmar que o nosso algoritmo não produz nenhum erro que não fosse já um erro intrínseco do sistema de arredondamento dos dados em ponto flutuante (o sistema não sabe distinguir a^* de a se $\delta(a^*, a) < u$). Podemos assumir que f^* produz a solução exata. Se $\delta < u$ ou δ é duma ordem próxima a u, diz-se que o algoritmo é estável (desde o ponto de vista da análise de erros inversa).

Definição 1.41. *Um **algoritmo** f^* que aproxima uma função f diz-se que é **estável em** a^* **para análise de erros direta** se:*

$$\delta(f^*(a^*), f(a^*)) < c_1 \cdot u$$

para uma constante c_1 não demasiado grande.

*Um **algoritmo** f^* que aproxima uma função f diz-se que é **estável em** a^* **para análise de erros inversa** se existe um a com $f(a) = f^*(a^*)$ e tal que:*

$$\delta(a^*, a) < c_2 \cdot u$$

para uma constante c_2 não demasiado grande.

Se $fl(x)$ representa o valor do sistema numérico que aproxima x com menor erro, o ideal no cálculo de f seria um algoritmo que produz sempre como resposta $f^*(a^*) = fl(f(a^*))$

Erros das operações aritméticas

Pensemos nas operações aritméticas elementares, como são a soma e o produto. Assumimos que conseguimos programar um algoritmo de soma estável, por exemplo $a^* \oplus b^* = fl(a^* + b^*)$. Isto resolve a questão do erro que o algoritmo introduz, muito próximo da unidade de arredondamente. Ainda temos a questão de se a soma $s(a,b) = a + b$ é uma função bem

condicionada. Para esta função $s(x,y) = x + y$ temos $\nabla_{(a,b)} s = (1,1)$, vetor com norma-1 simples de calcular: $\|(1,1)\|_1 = 2$. Portanto o número de condição para erros relativos (com a norma infinito) é:

$$\kappa_\infty(a,b) = \frac{\|(a,b)\|_\infty \cdot 2}{|a+b|}$$

este número de condição é elevado se $a + b$ for uma ordem de grandeza muito inferior aos valores $|a|$, $|b|$. Por exemplo, em $(a,b) = (1001, -1000.9)$ temos número de condição associado $1001/0.1 = 10010$

A operação de soma está mal condicionada quando é feita com valores de soma próxima a 0 (onde "próxima" é em comparação à ordem de grandeza dos elementos que são somados). Dizemos que executar uma soma de elementos quase opostos (ou a diferença de dois números próximos) é uma operação com **cancelamento catastrófico**, no sentido que pequenos erros nos dados de entrada implicam grandes erros no resultado da operação.

Para o produto podemos fazer o mesmo tipo de análise. No entanto vamos estudar a propagação do erro com ajuda do número de condição em cada componente. Se temos $f(x,y) = x \cdot y$, sabemos $\partial_x f(a,b) = b$, $\partial_y f(a,b) = a$. Portanto:

$$\kappa_x(a,b) = \frac{|a| \cdot |b|}{|a \cdot b|} = 1, \qquad \kappa_y(a,b) = \frac{|a| \cdot |b|}{|a \cdot b|} = 1$$

e para erros relativos podemos afirmar:

$$\delta(a^*, a) \leq \delta_x \wedge \delta(y^*, y) \leq \delta_y \Rightarrow \delta(a^* b^*, ab) = \delta(f(a^*, b^*), f(a,b)) \leq \kappa_x(a,b) \cdot \delta_x + \kappa_y(a,b) \cdot \delta_y = \delta_x + \delta_y$$

se as componentes não trazem um erro relativo grande, o produto não irá ter um erro relativo grande.