



UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE MATEMÁTICA
CURSO DE LICENCIATURA PLENA EM MATEMÁTICA

**Avaliações de Acurácia e de Desempenho Computacional
da Diferenciação Com Passo Complexo**

ALEXANDRE COSTA GOMES

CASTANHAL - PA

2018

ALEXANDRE COSTA GOMES

Avaliações de Acurácia e de Desempenho Computacional da Diferenciação Com Passo Complexo

Trabalho de Conclusão de Curso apresentado à Faculdade de Matemática da Universidade Federal do Pará (UFPA), Campus Castanhal, como parte dos requisitos para a obtenção de Licenciado Pleno em Matemática.

Orientador: Prof. Valdelírio da Silva e Silva

CASTANHAL - PA

2018

ALEXANDRE COSTA GOMES

Avaliações de Acurácia e de Desempenho Computacional da Diferenciação Com Passo Complexo

Trabalho de Conclusão de Curso apresentado à Faculdade de Matemática da Universidade Federal do Pará (UFPA), Campus Castanhal, como parte dos requisitos para a obtenção de Licenciado Pleno em Matemática.

Orientador: Prof. Valdelírio da Silva e Silva

Aprovado em 13 de Agosto de 2018.

Conceito: Excelente

BANCA EXAMINADORA

Prof. Valdelírio da Silva e Silva
Faculdade de Matemática / UFPA - Presidente

Prof. Cícero Roberto Teixeira Régis
Instituto de Geociências / UFPA - Membro Externo

Prof. Fábio Nogueira Batista
Faculdade de Matemática / UFPA - Membro

*Ao meu Deus, Rei da minha vida,
à família e à futura companheira.*

Agradecimentos

Primeiramente, agradeço a Deus no nome do Filho Jesus mediante a pessoa do Espírito Santo, pela sua rica misericórdia e graça, que diariamente me alcança para o fim de ser seu filho.

A minha família, em especial, aos meus avós, José Ivo Gomes e Josefina Gomes, pelo apoio incondicional em todas as situações da minha vida.

A minha mãe Rosângela Costa Gomes, a qual sempre me tem apoiado.

Aos meus tios, Miguel Rodrigues de Melo e Rosiane Gomes de Melo, pelo carinho, apoio e incrível hospitalidade em toda minha trajetória acadêmica, sem dúvida, foram e sempre serão a minha segunda casa.

A minha melhor amiga, namorada e futura esposa Rafaeli Lobo, pelo companheirismo, compreensão, paciência, incentivo, atenção, carinho e amor que sempre me foi dado em todas as situações da minha vida.

Aos meus amigos particulares que Deus me deu na faculdade, Eyder Gonçalves, Levi Carvalho e Rafael Duarte, pelas trocas de experiências, momentos de descontração e ensinamentos que me proporcionaram como homens de Deus.

A todos os professores da faculdade pelo apoio na trajetória de minha vida acadêmica, em especial, ao Prof. Valdelírio da Silva e Silva, pela oportunidade de receber suas preciosas orientações, pela dedicação e competência no desenvolvimento deste trabalho, além de suas contribuições na minha caminhada acadêmica.

Resumo

Os processos de otimização são largamente usados na resolução de equações não lineares. Os casos realísticos não possuem em geral solução analítica, e aí todas as etapas dentro da técnica de otimização precisam ser o mais acuradas possível. Dentre aquelas baseados em gradiente de funções, o uso das matrizes jacobiana e hessiana são largamente usadas nos passos para determinação das soluções, e portanto quanto mais precisas e mais rápidas forem, mais otimizado é o processo. Apesar de pouco conhecida, o método de diferenciação com passo complexo é uma das técnicas que pode ser usada para as determinações das derivadas, e neste trabalho investigamos sua aplicabilidade, avaliamos o parâmetro passo e comparamos o desempenho computacional com a técnica de diferenças finitas, em duas situações. Uma com a tarefa de determinação de derivadas de primeira ordem em funções de duas variáveis, com e sem variação abrupta de amplitude; e outra, de busca de mínimos usando derivadas parciais de primeira ordem no método de Newton. Em ambos problemas a técnica de derivação com passo complexo foi mais regular na escolha do passo, relativamente inferior em tempo computacional e, com exceção de um caso com equidade, foi melhor com pelo menos seis ordens de grandeza de acurácia que a técnica de diferenças finitas.

Palavras-chave: Derivada com Passo Complexo. Diferenças Finitas. jacobiana. hessiana. Otimização. Método de Newton.

Lista de Tabelas

3.1	Resultados da derivada de segunda ordem da função (3.3) pelo método DPC usando $g(\epsilon) = \epsilon^{1/3}$ e método DF, programados no MATLAB	25
3.2	Resultados da derivada de segunda ordem da função (3.3) pelo método DPC com $g(\epsilon) = h$ e o método DF, programados no MATLAB	26
3.3	Resultados da derivada de segunda ordem da função (3.3) pelo método DF, DPC com $g(\epsilon) = h$ e DPC com precisão de máquina $g(\epsilon_m) = \epsilon^{1/3}$, programados no Fortran	27
3.4	Desempenho de DF e DPC na determinação da raiz $x = 0$ da função (3.2) pelo método iterativo de Halley, com tolerância absoluta de 1×10^{-15} , usando aproximação inicial $x = 5$ e passo fixo $h = 10^{-8}$	34

Lista de Figuras

3.1	Erros relativos da primeira derivada da função (3.1) em $x = -1,74$	22
3.2	Erros relativos da segunda derivada da função (3.1) em $x = -1,74$	22
3.3	Erros relativos da primeira derivada da função (3.2) em $x = 0$	23
3.4	Erros relativos de segunda derivada da função (3.2) na abscissa $x = 0$	24
3.5	Dados apresentados por [2], em que a equação 6 refere-se à derivada de segunda ordem por DF centrada, e a equação 40 é a mesma fórmula (2.39)	26
3.6	Função <i>gentle</i>	28
3.7	Máximos erros absolutos referentes à variação do passo h nas aproximações das derivadas parciais da função <i>gentle</i> sobre o domínio $[0, 1] \times [0, 1]$	28
3.8	Derivada parcial da função <i>gentle</i> em relação a x por DF e DPC (com passo $h = 10^{-8}$) e seus respectivos erros absolutos	29
3.9	Função <i>cloverleaf</i>	30
3.10	Máximos erros absolutos referentes à variação do passo h nas aproximações das derivadas parciais da função <i>cloverleaf</i> sobre o domínio $[0, 1] \times [0, 1]$	30
3.11	Derivada parcial da função <i>cloverleaf</i> em relação a x por DF e DPC (com passo $h = 10^{-8}$) e seus respectivos erros absolutos	31
3.12	Gráfico da função (3.2) na proximidade da raiz	33
3.13	Valores da primeira derivada da função (3.2) em x à direita próximo da abscissa de seu ponto de máximo	35
3.14	Parte imaginária da função (3.2) no plano complexo	36
3.15	Curvas de nível e singularidades da parte imaginária da função (3.2) no plano complexo	36
3.16	Valores médios de $\ \mathbf{x}\ $ da resolução do sistema 3.16 pelo método de Newton, obtidas com aproximação da jacobiana por DF centrada e DPC sob variação do incremento para as derivadas.	40
3.17	Valores médios do número de iterações na resolução do sistema 3.16 pelo método de Newton, obtidos com aproximação da jacobiana por DF centrada e DPC sob variação do incremento para as derivadas.	41

3.18 Valores médios de tempo de processamento na resolução do sistema 3.16 pelo método de Newton, obtidos com aproximação da jacobiana por DF centrada e DPC sob variação do incremento para as derivadas. 41

Sumário

1	Introdução	8
2	Métodos Computacionais de Diferenciação	9
2.1	Diferenças Finitas	9
2.1.1	Diferenciação com passo adiante e diferenciação com passo atrás	10
2.1.2	Diferenciação centrada	11
2.1.3	Diferenciação de segunda ordem	12
2.2	Diferenciação simbólica e diferenciação automática	13
2.3	Diferenciação com Passo Complexo	14
2.3.1	Função analítica e as condições de Cauchy-Riemann	14
2.3.2	Aproximação para a primeira derivada	15
2.3.3	Aproximação para a segunda derivada	17
2.3.4	Aproximações com a precisão de máquina	19
3	Avaliações	21
3.1	Em funções de uma variável	21
3.2	Em funções bidimensionais	27
3.3	Em método iterativo sobre função de uma variável	32
3.4	Em método iterativo de resolução de sistemas não-lineares	37
3.4.1	Método de Newton em sistemas $n \times n$ de equações não lineares	37
3.4.2	Newton aplicado em exemplo de sistemas não lineares	39
4	Considerações Finais	42

Capítulo 1

Introdução

Em Matemática aplicada computacional, a derivada é extremamente usada em problemas de otimização baseados em técnicas de gradiente. As determinações de derivadas com acurácia e precisão são requisitos fundamentais para que o processo seja realizado com eficácia. [18], *apud* [3], apresentam acurácia como sendo o grau de proximidade de uma estimativa com seu parâmetro (ou valor verdadeiro), enquanto precisão expressa o grau de consistência da grandeza medida em relação à sua média. Ou seja, nesses processos, busca-se derivadas que nos forneçam os resultados mais próximos da analítica. Entretanto, nem sempre podemos ter disponibilizadas, nestes processos, as derivadas analíticas. Desse modo, é imprescindível se utilizar de técnicas numéricas de derivação com melhores aproximações possíveis.

Entre as mais conhecidas técnicas de diferenciação se tem a Diferenciação Automática (DA) [8, 4], Diferenças Finitas (DF) [16] e, uma nem tão conhecida como as demais, a Diferenciação com Passo Complexo (DPC) [17, 14, 1, 6, 15], sobre a qual desenvolvemos neste trabalho.

Este texto está organizado em quatro capítulos. O primeiro refere-se a esta introdução. O segundo apresenta os principais métodos de diferenciação numérica, assim como também detalha o método DPC e suas particularidades, bem como alguns conceitos de derivadas nos números complexos, que irão dar suporte para entender a técnica DPC e sua obtenção. No capítulo três, iremos aplicar as técnicas DF e DPC ao cálculo de derivadas de funções de uma e duas variáveis, e outra, na obtenção da raiz de uma função pelo método iterativo de Halley, e a problemas de otimização com o método de Newton, mostrando a suscetibilidade dos resultados quanto à variação do passo e comparações de desempenho e acurácia entre elas em derivadas de primeira e segunda ordem. No quarto capítulo apresentamos nossas considerações finais sobre nossas investigações e recomendações para outras aplicações.

Capítulo 2

Métodos Computacionais de Diferenciação

Acurácia, precisão, custo computacional e facilidade de implementação são critérios importantes que costumam ser avaliados quando se deseja obter computacionalmente as derivadas. Para problemas complexos, os métodos computacionais são uma ferramenta escolhida.

No que tange aos métodos computacionais, os métodos numéricos correspondem a um conjunto de ferramentas ou técnicas usadas para se obter a solução de problemas matemáticos de forma aproximada.

Neste capítulo vamos apresentar o método computacional mais clássico para obtenção de derivadas, o de *Diferença Finitas* (DF); em seguida comentar sobre a *Diferenciação Automática* (DA); e por fim expor o método de diferenciação objeto de nossa pesquisa, o método de *Diferenciação Com Passo Complexo* (DPC), e mostraremos suas particularidades em relação à implementação, à acurácia e tempo computacional no capítulo de aplicações.

2.1 Diferenças Finitas

Fórmulas de diferenciação baseadas em DF são um método muito comum para aproximação de derivadas. Nesta seção iremos apresentar os tradicionais métodos de DF, diferenciação com passo “adiante”, centrada e “para trás”, as quais são as principais técnicas utilizáveis à obtenção de aproximação de derivada de primeira ordem; e também uma usada para encontramos a derivada de segunda ordem.

De início, podemos enfatizar que todas são obtidas a partir da expansão da série de Taylor de sobre uma função. A série de Taylor, ou polinômio de Taylor, é uma somatória infinita que representa qualquer função infinitamente diferenciável em torno de um ponto de abscissa a , do

intervalo que também contém x , pela expressão [7]:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)(x-a)^2}{2!} + \frac{f^{(3)}(a)(x-a)^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n. \quad (2.1)$$

2.1.1 Diferenciação com Passo Adiante e com Passo Atrás

Possivelmente a técnica mais comum utilizada para DF é a com passo adiante. Esta técnica é obtida da expansão de Taylor com um passo positivo, ou incremento h , na função $f(x)$ em torno da abscissa x , isto é:

$$f(x+h) = f(x) + hf'(x) + h^2 \frac{f''(x)}{2!} + h^3 \frac{f^{(3)}(x)}{3!} + \dots \quad (2.2)$$

Eliminado os termos a partir de h^2 e isolando $f'(x)$, obtemos:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}. \quad (2.3)$$

Essa é uma aproximação numérica para primeira derivada de uma função, que possui um erro de truncamento¹ da ordem h .

Em suma, a implementação do método de DF é bastante simples, basta calcular o valor da função no ponto x que nos interessa e em um ponto próximo, digamos, $x+h$, e em seguida subtraí-las e dividir o resultado pelo valor do incremento. Contudo, há algumas preocupações que devem ser tratadas para se considerar a eficiência da técnica. Em primeiro lugar, a escolha do comprimento h geralmente é feita de forma arbitrária, mas certamente não trivial [6]. Analiticamente um passo h muito pequeno conduziria a um excelente valor da derivada. Entretanto, numericamente isso não significa a mesma coisa, haja vista, que o computador possui um número finito de dígitos para representar h na sua memória, implicando em um limite de precisão numérica para o quão pequeno pode ser h .

Além disto, à medida que tomamos h muitíssimo pequeno podemos ser levados ao **erro de cancelamento subtrativo**, pois DF faz menção à subtração de funções, logo se h for extremamente pequeno, dependendo da função f , tal subtração resultaria em valor nulo, mas que verdadeiramente não retrataria a “inclinação” nula da função naquele ponto². Por outro lado, com o incremento h muito grande podemos ter prejuízos na obtenção da precisão da derivada, já que a série de Taylor toma um ponto com incremento, próximo para ser avaliado

¹Como bem escrito em (2.3), a expressão da primeira derivada é uma aproximação, advinda do truncamento da série de Taylor. Consequentemente a derivada utilizando-se essa fórmula contém o chamado **erro de truncamento** [7] por interromper um processo para consideração de um número pequeno, ou finito, de passos dentro de um com um número relativamente maior ou infinito de etapas.

²Na verdade, aqui o erro de cancelamento subtrativo é consequência da imprecisão da máquina ao representar a imagem da função de um ponto com incremento tão próximo do ponto de interesse na derivada. E sendo assim, este erro é resultado imediato do erro de arredondamento só que nas imagens da função!

na função. Por isso, o método de DF possui algumas deficiências que devem ser levadas em consideração no processo de aquisição da derivada.

Se em vez de aplicarmos o incremento positivo h a partir de x usarmos h subtraindo x na série de Taylor, obteremos:

$$f(x - h) = f(x) - hf'(x) + h^2 \frac{f''(x)}{2!} - h^3 \frac{f^{(3)}(x)}{3!} + \dots \quad (2.4)$$

Da mesma forma que anteriormente, eliminado os termos a partir de h^2 e isolando $f'(x)$, ficaremos com:

$$f'(x) \approx \frac{f(x) - f(x - h)}{h}. \quad (2.5)$$

Essa é uma aproximação para primeira derivada de $f(x)$ chamada de diferenciação com passo atrás, e possui as mesmas vantagens e desvantagem que a técnica de diferenciação com passo adiante da fórmula (2.3), inclusive a mesma ordem de erro de truncamento.

2.1.2 Diferenciação Centrada

Se a diferença com passo adiante é o método mais comum de calcular derivada, a diferença centrada é a segunda técnica de DF mais utilizada [6]. A derivação obtida por meio desta formulação é construída a partir das derivações adiante e atrás. Subtraindo a equação (2.4) da (2.2), teremos:

$$f(x + h) - f(x - h) = f(x) - f(x) + hf'(x) + hf'(x) + h^2 \frac{f''(x)}{2!} - h^2 \frac{f''(x)}{2!} + \dots \quad (2.6)$$

O que, nos leva a:

$$f(x + h) - f(x - h) = 2hf'(x) + 2h^3 \frac{f^{(3)}(x)}{3!} + \dots \quad (2.7)$$

Desconsiderando os termos de ordem maior que um, obteremos uma aproximação da primeira derivada com um erro de truncamento na ordem de h^2 :

$$f'(x) \approx \frac{f(x + h) - f(x - h)}{2h}. \quad (2.8)$$

Encontramos duas referências que falam do incremento h sendo possivelmente um ótimo. O livro [20] sugere que o melhor valor é dado por $\sqrt{\epsilon_m} x_c$, em que ϵ_m é a precisão da máquina e $x_c = \sqrt{\frac{f}{f''}}$ é a escala de curvatura (sendo f'' a segunda derivada), a qual depende em que ponto do domínio será aplicado.

Esse valor ϵ_m , em precisão simples,³ se tem $\epsilon_m = 1,1920929 \times 10^{-7}$. Em precisão dupla

³Dados adquiridos com a função *epsilon(0)* da linguagem *Fortran*. Qualquer linguagem de programação

o computador tem $\epsilon_m = 2,2204460492503131 \times 10^{-16}$, e já com quádrupla precisão se tem $\epsilon_m = 1,92592994438723585305597794258493732 \times 10^{-34}$. Dito isto, dado um número real x , a precisão de máquina ϵ_m é um valor para o qual $x + \epsilon_m$ é o menor número real tal que seja maior que x .

No caso de desconhecimento da escala de curvatura, que compreende-se ao caso geral em aplicações de funções não dadas analiticamente, recomenda-se o uso de $x_c = x$ onde desejamos obter a derivada. Já [6] diz que se não houver nem mesmo a possibilidade de verificar empiricamente a escala de curvatura, uma boa escolha para o incremento é

$$h = \epsilon(1 + |x|), \quad (2.9)$$

sendo $\epsilon = 10^{-8}$ usado para a diferenciação adiante, enquanto que $h = 10^{-4}$ ele usou para seus estudos com diferenciação centrada.

Encontramos também outra orientação de que valores h seriam ótimos. O livro [11], página 248, afirma que $h = \sqrt{\epsilon_m}$ ou $h = \sqrt{\epsilon_m(1 + |x|)}$ são bons valores de forma a balancear os erros de cancelamento subtrativo e de arredondamento.

2.1.3 Diferenciação de Segunda Ordem por DF

Uma das fórmulas para derivadas de segunda ordem por DF pode ser obtida somando-se as equações (2.2) e (2.4):

$$f(x+h) + f(x-h) = f(x) + f(x) + hf'(x) - hf'(x) + h^2 \frac{f''(x)}{2!} + h^2 \frac{f''(x)}{2!} + h^3 \frac{f^{(3)}(x)}{3!} - h^3 \frac{f^{(3)}(x)}{3!} + \dots \quad (2.10)$$

Resultando em:

$$f(x+h) + f(x-h) = 2f(x) + \frac{2h^2 f''(x)}{2!} + \frac{2h^4 f^{(4)}(x)}{4!} + \dots \quad (2.11)$$

Desprezando os termos com derivadas de ordem igual ou superior a quatro obtemos a aproximação para a derivada de segunda ordem da forma diferença centrada:

$$f''(x) \approx \frac{f(x+h) + f(x-h) - 2f(x)}{h^2}. \quad (2.12)$$

Encontramos essa expressão ignorando as parcelas com h^2 , logo o erro de truncamento da aproximação é de $o(h^2)$.

disponibiliza essa função de encontrar o menor número ϵ tal que $x + \epsilon > x$.

2.2 Diferenciação Simbólica e Diferenciação Automática

Mais recentemente duas técnicas de diferenciação ganharam espaço no meio computacional, a diferenciação simbólica e a diferenciação automática. Ambas evitam o erro de truncamento obtido pelos métodos numéricos. Um caminho à obtenção do valor da derivada, sem erro de truncamento e numa forma automatizada, é a diferenciação simbólica (DS). Essa diferenciação baseia-se na observação fundamental de que qualquer função implementada por um programa de computador, pode ser escrita como uma sequência finita de operações elementares envolvendo um ou dois argumentos (soma, divisão e outras funções elementares, como por exemplo, trigonométricas e exponenciais). Assim, a DS também é uma composição de funções intrínsecas (seno, cosseno, exponencial, etc) com ou sem operadores (soma, subtração, multiplicação e divisão). Segundo [12], pag. 46:

A principal vantagem da DS é o fato do valor da derivada ser livre do erro de truncamento. Além disto, ela pode ser realizada automaticamente, ao contrário da determinação manual da expressão analítica. Entretanto, o custo computacional necessário para a diferenciação é bastante elevado. Além disto, a expressão obtida para a derivada pode ser muito complexa por possuir uma série de termos repetidos (já que nem sempre estes termos são identificados e eliminados pelo algoritmo de DS).

A diferenciação automática (DA), ou algorítmica, é uma técnica para avaliar derivadas de funções definidas por programa de computador de forma semelhante à DS mas que é inferior em custo computacional porque ela informa apenas o valor numérico da derivada e não a sua expressão, exigindo menos memória e tempo de CPU. No entanto, comparativamente às técnicas numéricas tradicionais, seu tempo computacional ainda é alto, testados em vários exemplos que realizamos, daí também não a avaliaremos quanto ao seu desempenho.

O princípio básico da DA, assim como da DS, consiste no fato de que a composição de uma funções, no computador, é sempre realizada por meio de uma sequência ordenada de funções elementares. Deste ponto de vista, o programa representa uma composição de várias funções simples, as quais possuem derivadas conhecidas. Além da vantagem de DA e DS não possuírem erro de truncamento em relação à DF, DA e DS também não recaem no erro subtrativo, como afirma [4].

2.3 Diferenciação com Passo Complexo

Um método de diferenciação numérica que tem se mostrado ser de fácil implementação e maior acurácia em comparação aos métodos padrões de diferenciação é o chamado *Diferenciação Com Passo Complexo* (DPC). Este método trata de forma simples o cálculo de primeira derivada para funções reais usando cálculo com funções complexas.

A ideia de usar números complexos à obtenção de derivadas surgiu por intermédio primeiramente de [13], *apud* [16], e que foram baseadas no teorema de Cauchy. Porém estudos seguintes mostraram que o método tinha um alto tempo computacional em sua execução, e relativamente complicado implementar. Entretanto, como declarou [15], [21] desenvolveram expressões muito mais simples e elegantes, que baseadas em etapas complexas de diferenciação para cálculos de derivadas numéricas de primeira ordem de uma função analítica, proporcionaram implementação fácil e de tempo computacional muito inferior à versão da aproximação anterior.

2.3.1 Função Analítica e as Condições de Cauchy-Riemann

Para podermos apresentar a metodologia da obtenção das derivadas de primeira e segunda ordens mediante a DPC, achamos conveniente recordar alguns elementos do cálculo com uma variável complexa para enunciar as condições necessárias à sua implementação.

A definição de derivada de uma função de variável complexa, com exceção do domínio e contradomínio, é convencionalmente a mesma que de uma função com variável real; isto é, seja $f : R \rightarrow \mathbb{C}$ uma função cujo domínio seja uma região aberta $R \subset \mathbb{C}$, e seja z um ponto de R , dizemos que f é *derivável* no ponto z se existe o limite [19]:

$$\lim_{\Delta z \rightarrow 0} \frac{f(z + \Delta z) - f(z)}{\Delta z}. \quad (2.13)$$

E tal limite é o número complexo denotado por $f'(z)$. Se além disso f é contínua em R , ela é chamada também como holomorfa, e se f , para cada $z_0 \in R$, possuir uma série de potências tal que f possa ser expandida localmente em z_0 , ela é então chamada analítica. Em outras palavras; uma função complexa f de variável z é dita analítica em um ponto z_0 , se for diferenciável em todos os pontos da vizinhança de z_0 . Dizemos então que f é analítica, se f é analítica em todos os pontos do seu domínio. Entretanto, da teoria de funções de uma variável complexa, a holomorfia e analiticidade são fatos equivalentes ([19]).

Agora seja $z = x + iy$ um ponto no domínio complexo e $f(z) = u(x, y) + i v(x, y)$ uma função analítica composta pelas funções reais u e v que são partes real e imaginária de f . Então as derivadas parciais em relação as partes real e imaginária, $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$, $\frac{\partial v}{\partial x}$ e $\frac{\partial v}{\partial y}$ também existem, e

como condição necessária e suficiente, satisfazem as equações de *Cauchy-Riemann* ([19]):

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad (2.14)$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \quad (2.15)$$

E mediante essas equações sendo satisfeitas é que se desenvolve a teoria das aproximações de derivadas numéricas com passo complexo. Então, mesmo se uma função f com domínio em um intervalo do conjunto dos reais, seja contínua e diferenciável até uma certa ordem, mas que isso não aconteça em âmbitos com domínio numa região dos complexos, não é possível aproximar numericamente as derivadas nas proximidades dos pontos de singularidades, como perto de pólos e pontos de ramificação existentes nesse domínio.

2.3.2 Primeira Derivada Com Passo Complexo

Se f é analítica, as equações de Cauchy-Riemann podem ser verificadas. Dessa forma, empregando a definição de derivada no lado direito da equação (2.14), temos:

$$\frac{\partial u}{\partial x} = \lim_{h \rightarrow 0} \frac{v(x, y+h) - v(x, y)}{h} \quad (2.16)$$

Nosso intuito é aplicar essa igualdade à função f real. Mas isso é possível somente se nossa função f for estendida a possuir uma parte real $u(x)$ e imaginária $v(x)$. Se tivermos f somente uma função com contra-domínio nos reais, nos é necessário expandir e admitir que exista uma parte imaginária nela, que convenientemente seja nula, i.e., $v(x) = 0$. Mas além, se f estiver definida com domínio nos reais, nos é necessário também expandir de modo que passe a ser definida com variável complexa $z = x + iy$. No entanto, assumir essas condições não nos implica em grandes problemas se f for analítica; e a aproximação da forma original por essa nova forma não resulta em grandes diferenças nas imagens se a parte imaginária do domínio for considerada um deslocamento vertical relativamente pequeno do eixo dos reais. Assim, para podermos usar a identidade (2.16) consideraremos $y = 0$ e $v(x) = 0$, de modo que tenhamos $v(x, h)$, com h relativamente pequeno, a fim de que f seja analítica em torno do ponto $x + ih$. Como v é a parte imaginária de $f(x + ih) = u(x, h) + i v(x, h)$, poderemos dizer que:

$$\frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{\text{Im}[f(x + ih)]}{h}, \quad (2.17)$$

Quando h é suficientemente pequeno, obteremos:

$$\frac{\partial f}{\partial x} \approx \frac{\text{Im}[f(x + ih)]}{h}. \quad (2.18)$$

Essa estimativa é a que nos fornece uma aproximação à primeira derivada de $f(x)$. Notemos de início, que esta aproximação mostra os seus benefícios em comparação à DF, pois é notório que neste caso não existe nenhuma subtração, o que representa que o erro de cancelamento subtrativo está ausente. E, em termos de número de avaliações, apenas uma avaliação da função $f(x + ih)$, mas, computacionalmente levada em conta não só o domínio do eixo real, como também o eixo imaginário.

Para obtemos o erro de truncamento envolvido nessa aproximação iremos repetir a derivação utilizada por [21], e que se baseia na expansão da série de Taylor. No entanto, em vez de usarmos um incremento h real na expansão (2.2), usaremos o passo sendo um imaginário puro ih . Nessa consideração, se f for uma função de variável real e analítica no domínio complexo podemos expandi-la na série Taylor em torno do ponto de abscissa x :

$$f(x + ih) = f(x) + ihf'(x) + (ih)^2 \frac{f''(x)}{2!} + (ih)^3 \frac{f^{(3)}(x)}{3!} + \dots \quad (2.19)$$

Sabendo que $i^2 = -1$, temos

$$f(x + ih) = f(x) + ihf'(x) - h^2 \frac{f''(x)}{2!} - ih^3 \frac{f^{(3)}(x)}{3!} + \dots \quad (2.20)$$

Então, a partir da equação (2.20), separando as partes reais e imaginárias, obteremos

$$\operatorname{Re}[f(x + ih)] = f(x) - h^2 \frac{f''(x)}{2} + h^4 \frac{f^{(4)}(x)}{4!} - \dots \quad (2.21)$$

$$\operatorname{Im}[f(x + ih)] = hf'(x) - h^3 \frac{f^{(3)}(x)}{3!} + h^5 \frac{f^{(5)}(x)}{5!} - \dots \quad (2.22)$$

Tomando a equação (2.22) e dividindo pelo incremento h , teremos:

$$f'(x) = \frac{\operatorname{Im}[f(x + ih)]}{h} + h^2 \frac{f^{(3)}(x)}{3!} - h^4 \frac{f^{(5)}(x)}{5!} + \dots \quad (2.23)$$

Portanto o erro de truncamento da aproximação para a primeira derivada, $E_{trunc}(h)$, é majorado por:

$$E_{trunc}(h) = \frac{h^2}{6} f^{(3)}(x). \quad (2.24)$$

Comparativamente às fórmulas (2.3), (2.5) e (2.8) de DF, vemos que somente a diferença centrada possui mesma a ordem de erro de truncamento, $o(h^2)$, da diferenciação com passo complexo para a primeira derivada.

Para o cálculo de derivadas parciais de primeira ordem de uma função com mais de uma variável $f(x_1, x_2, \dots, x_n)$ basta utilizarmos a aproximação (2.18) em cada uma das variáveis da

função:

$$\begin{aligned} \frac{\partial f}{\partial x_1} &\approx \frac{\operatorname{Im}[f(x_1 + ih, x_2, \dots, x_n)]}{h} \\ \frac{\partial f}{\partial x_2} &\approx \frac{\operatorname{Im}[f(x_1, x_2 + ih, \dots, x_n)]}{h} \\ &\vdots \\ \frac{\partial f}{\partial x_n} &\approx \frac{\operatorname{Im}[f(x_1, x_2, \dots, x_n + ih)]}{h} \end{aligned} \quad (2.25)$$

Essas equações são então utilizadas para aproximações de derivadas de primeira ordem nas funções vetoriais de várias variáveis, como na matriz jacobiana comumente usadas em métodos de otimização baseados em gradiente.

2.3.3 Segunda Derivada Com Passo Complexo

O cálculo da segunda derivada da função f pela aproximação com passo complexo é obtido diretamente da equação (2.21). A partir dela vemos que:

$$\begin{aligned} \operatorname{Re}[f(x + ih)] &= f(x) - h^2 \frac{f'(x)}{2} + h^4 \frac{f^{(4)}(x)}{4!} - \dots \\ f''(x) \frac{h^2}{2} &= f(x) - \operatorname{Re}[f(x + ih)] + h^4 \frac{f^{(4)}(x)}{4!} - \dots \\ f''(x) &= \frac{2}{h^2} \left\{ f(x) - \operatorname{Re}[f(x + ih)] + h^4 \frac{f^{(4)}(x)}{4!} - \dots \right\}. \end{aligned} \quad (2.26)$$

Desconsiderando os termos a partir daqueles com fator h^2 , ficaremos com:

$$f''(x) \approx \frac{2}{h^2} \{f(x) - \operatorname{Re}[f(x + ih)]\}, \quad (2.27)$$

e sendo:

$$E_{trunc}(h) = \frac{h^2}{12} f^{(4)}(x). \quad (2.28)$$

Notemos que, diferente da equação mostrada em (2.18), nesta aproximação da segunda derivada existe uma diferença de funções, por isso quando tomado um incremento h muitíssimo pequeno, pode conduzir a máquina a um possível erro de cancelamento subtrativo. Além disso, vimos que o erro de truncamento da segunda derivada de DF, dada pela aproximação (2.12), é da $o(h^2)$, sendo a mesma que a dada aqui por (2.28).

A obtenção das derivadas de segunda ordem em funções de mais de uma variável $f(x_1, x_2, \dots, x_n)$ é feita sob duas aproximações. Nas derivadas segundas sobre a mesma variável basta

usarmos a equação (2.27):

$$\begin{aligned}
 \frac{\partial^2 f}{\partial x_1^2} &\approx \frac{2}{h^2} \{f(x_1, x_2, \dots, x_n) - \operatorname{Re}[f(x_1 + ih, x_2, \dots, x_n)]\} \\
 \frac{\partial^2 f}{\partial x_2^2} &\approx \frac{2}{h^2} \{f(x_1, x_2, \dots, x_n) - \operatorname{Re}[f(x_1, x_2 + ih, \dots, x_n)]\} \\
 &\vdots \\
 \frac{\partial^2 f}{\partial x_n^2} &\approx \frac{2}{h^2} \{f(x_1, x_2, \dots, x_n) - \operatorname{Re}[f(x_1, x_2, \dots, x_n + ih)]\}
 \end{aligned} \tag{2.29}$$

Já no caso das derivadas cruzadas, usamos a expansão de Taylor para duas variáveis:

$$f(x + h, y + k) = f(x, y) + \frac{\partial f}{\partial x} h + \frac{\partial f}{\partial y} k + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2} h^2 + 2 \frac{\partial^2 f}{\partial x \partial y} hk + \frac{\partial^2 f}{\partial y^2} k^2 \right) + \dots \tag{2.30}$$

Quando a adaptamos com passos complexos ih e ik , reescrevemos como:

$$f(x + ih, y + ik) = f(x, y) + \frac{\partial f}{\partial x} ih + \frac{\partial f}{\partial y} ik + \frac{1}{2} \left[\frac{\partial^2 f}{\partial x^2} (ih)^2 + 2 \frac{\partial^2 f}{\partial x \partial y} ih ik + \frac{\partial^2 f}{\partial y^2} (ik)^2 \right] + \dots \tag{2.31}$$

No entanto já temos conhecidas as aproximações de $\frac{\partial f}{\partial x}$ e $\frac{\partial f}{\partial y}$ por (2.25); e, $\frac{\partial^2 f}{\partial x^2}$ e $\frac{\partial^2 f}{\partial y^2}$ por (2.29); de modo que quando as substituimos na equação (2.31) acima, desconsiderando as parcelas com termos de derivadas superiores à segunda, obteremos:

$$\begin{aligned}
 f(x + ih, y + ik) &\approx f(x, y) + \operatorname{Im}[f(x + ih, y)]i + \operatorname{Im}[f(x, y + ik)]i - \\
 &\left\{ f(x, y) - \operatorname{Re}[f(x + ih, y)] + \frac{\partial^2 f}{\partial x \partial y} hk + f(x, y) - \operatorname{Re}[f(x, y + ik)] \right\}.
 \end{aligned} \tag{2.32}$$

Como nossa função f é real, o que nos interessa nessa equação são as parcelas reais; daí, com essa consideração, obteremos as derivadas cruzadas ao isolarmos $\frac{\partial^2 f}{\partial x \partial y}$:

$$\frac{\partial^2 f}{\partial x \partial y} \approx \frac{\operatorname{Re}[f(x + ih, y)] + \operatorname{Re}[f(x, y + ik)] - f(x, y) - \operatorname{Re}[f(x + ih, y + ik)]}{kh} \tag{2.33}$$

A equação (2.29) e uma adequação da (2.33) compõem as aproximações das segundas derivadas de uma função f de várias variáveis mediante a técnica com passo complexo. A extensão para uma função vetorial qualquer é imediata e ela pode ser utilizada para a determinação da matriz hessiana nos problemas de otimização. No entanto, a expressão dessas derivadas cruzadas mostram que, além do valor $f(x, y)$ necessário, são exigidas também mais três outras avaliações, requerendo diretamente um custo computacional bem maior que a aproximação de primeira derivada.

2.3.4 Aproximações com a Precisão de Máquina

No trabalho de [1], o autor sugeriu novas aproximações para DPC. Ele formulou as derivadas de primeira e segunda ordem mediante adoção da expansão da série de Taylor sobre função complexa, mas com um acréscimo e um decréscimo de uma função $g(\epsilon_m)$ ao valor x do qual se deseja obter as derivadas. Em que ϵ_m é precisão de máquina e $g(\epsilon_m)$ é uma função composta por ϵ_m .

Denotemos z_{\pm} como sendo os números complexos com acréscimo e decréscimo ϵ sobre x e com passo complexo h , ou seja, $z_{\pm} = (x \pm g(\epsilon_m) + ih)$. Sobre estes pontos apliquemos a série de Taylor em torno de x_{\pm} da função f a qual desejamos derivar:

$$f(z_+) = f(x + g(\epsilon_m) + ih) = f(x_+ + ih) = f(x_+) + ihf'(x_+) - \frac{h^2}{2}f''(x_+) + \dots + \frac{(ih)^k}{k!}f^{(k)}(x_+) \quad (2.34)$$

$$f(z_-) = f(x - g(\epsilon_m) + ih) = f(x_- + ih) = f(x_-) + ihf'(x_-) - \frac{h^2}{2}f''(x_-) + \dots + \frac{(ih)^k}{k!}f^{(k)}(x_-) \quad (2.35)$$

Ao subtrairmos a segunda equação da primeira, e denotando $\delta f(z_{\pm}) = f(z_+) - f(z_-)$ essa diferença, teremos:

$$\begin{aligned} \delta f(z_{\pm}) = & f(x_+) - f(x_-) + ih [f'(x_+) - f'(x_-)] - \frac{h^2}{2} [f''(x_+) - f''(x_-)] + \dots \\ & + \frac{(ih)^k}{k!} [f^{(k)}(x_+) - f^{(k)}(x_-)] \end{aligned} \quad (2.36)$$

O ponto crucial para obtenção das fórmulas acontece ao se aplicar a série de Taylor em torno de x_{\pm} sobre as derivadas de ordem k , ou seja:

$$\begin{aligned} f^{(k)}(x_+) &= f^{(k)}(x) + \delta_x f^{(k+1)}(x) + \frac{\delta_x^2}{2} f^{(k+2)}(x) + \frac{\delta_x^3}{3!} f^{(k+3)}(x) + \dots \\ f^{(k)}(x_-) &= f^{(k)}(x) - \delta_x f^{(k+1)}(x) + \frac{\delta_x^2}{2} f^{(k+2)}(x) - \frac{\delta_x^3}{3!} f^{(k+3)}(x) + \dots \end{aligned}$$

E assim, para $k = 0, 1, \dots$; teremos $\delta f^{(k)}(x_{\pm}) = f^{(k)}(x_+) - f^{(k)}(x_-) = 2\delta_x f^{(k+1)}(x) + o(\delta_x^3)$. E conseqüentemente, substituindo em (2.36), dispensando a ordem de erro de truncamento, obteremos:

$$\delta f(z_{\pm}) = 2\delta_x \cdot f'(x) + ih \cdot 2\delta_x \cdot f''(x) - \frac{h^2}{2} \cdot 2\delta_x \cdot f^{(3)}(x) + \dots + \frac{(ih)^k}{k!} \cdot 2\delta_x \cdot f^{(k+1)}(x). \quad (2.37)$$

É possível então obtermos as aproximações para a primeira e segunda derivada em x como

sendo:

$$f'(x) \approx \frac{1}{2\delta_x} \operatorname{Re}\{\delta f(z_{\pm})\} \quad (2.38)$$

$$f''(x) \approx \frac{1}{2\delta_x h} \operatorname{Im}\{\delta f(z_{\pm})\}. \quad (2.39)$$

Onde a ordem do erro de truncamento das expressões (2.38) e (2.39) são de $h^2 \cdot \delta_x$:

$$E_{trunc}(h) = \frac{h^2}{2} \delta_x \cdot f^{(3)}(x). \quad (2.40)$$

$$E_{trunc}(h) = \frac{h^2}{3} \delta_x \cdot f^{(4)}(x). \quad (2.41)$$

Já que $\delta f(z_{\pm}) = f(x_+ + ih) - f(x_- + ih) = f(x + g(\epsilon_m) + ih) - f(x - g(\epsilon_m) + ih)$, a aproximação (2.38) é mais dispendiosa computacionalmente em comparação à primeira derivada obtida em (2.18), pois naquela precisamos apenas de uma avaliação da função no plano complexo. Em contrapartida, a aproximação (2.39) possui o mesmo número de avaliações que a aproximação da segunda derivada dadas por (2.27). No entanto, nos problemas de otimização com necessidade de determinação da matriz hessiana, o procedimento para determinação da aproximação (2.39) já inclui o cálculo do vetor gradiente por (2.38). É conveniente também ressaltar que as expressões (2.38) e (2.39) são sujeitas ao erro subtrativo, que podem acontecer na diferença das avaliações de $f(x_+ + ih)$ e $f(x_- + ih)$.

Capítulo 3

Avaliações

Neste capítulo apresentamos exemplos de aplicações da DPC e a comparamos com os resultados obtidos pela técnica de DF. Na primeira aplicação, investigamos as acurácias da DPC e DF analisando os erros relativos e absolutos de funções reais de apenas uma variável, sendo os erros nos cálculos de derivadas de primeira e segunda ordem. Como segunda aplicação, calculamos derivadas de funções reais de duas variáveis. Em seguida aplicamos as duas técnicas para obtenção da raiz de uma função real pelo método iterativo de Halley, do qual faz o uso de primeira e segunda derivadas. Por último, resolvemos um sistema de equações não-lineares pelo método de Newton, determinando a matriz jacobiana tanto com DF quanto DPC, a fim de fazermos estudo de desempenho computacional, e sobre acurácia obtida variando-se o passo.

3.1 Em Funções de Uma Variável

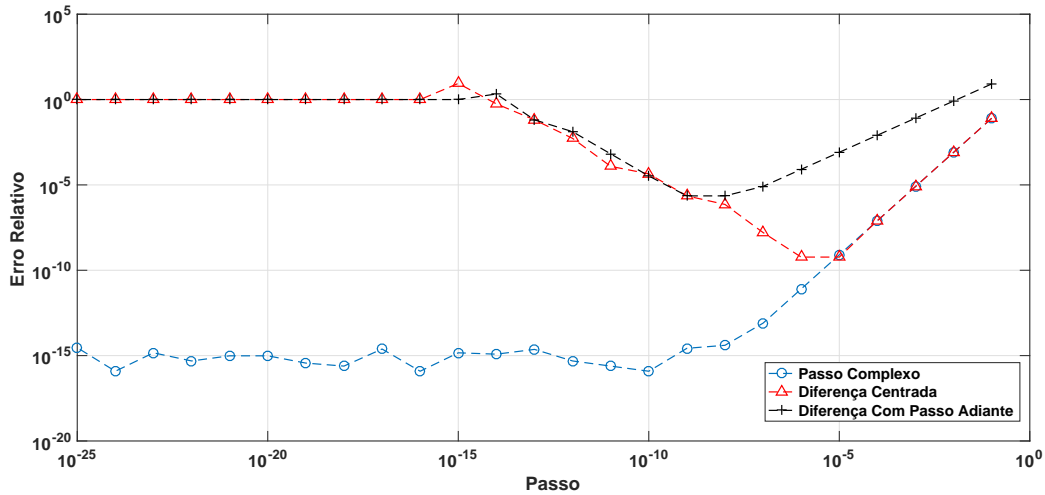
Nesta seção, apresentamos as investigações de acurácia das derivadas de primeira e segunda ordem sobre funções de uma variável. Qualquer resposta nesta seção refere-se ao uso da precisão de máquina ϵ_m de dupla precisão e os erros mensurados são do tipo relativo.

Como primeira função consideremos:

$$f(x) = e^x + \text{sen}(x). \quad (3.1)$$

Variamos os passos a partir de 10^{-25} a 10^{-1} . Na figura (3.1) apresentamos os erros relativos da primeira derivada de (3.1), quando aproximadas pelas fórmulas (2.18) de DPC, por (2.8) de DF centrada, e (2.3) de DF com passo adiante, calculadas na abscissa $x = -1,74$, o qual está próximo de um dos pontos de inflexão da função.

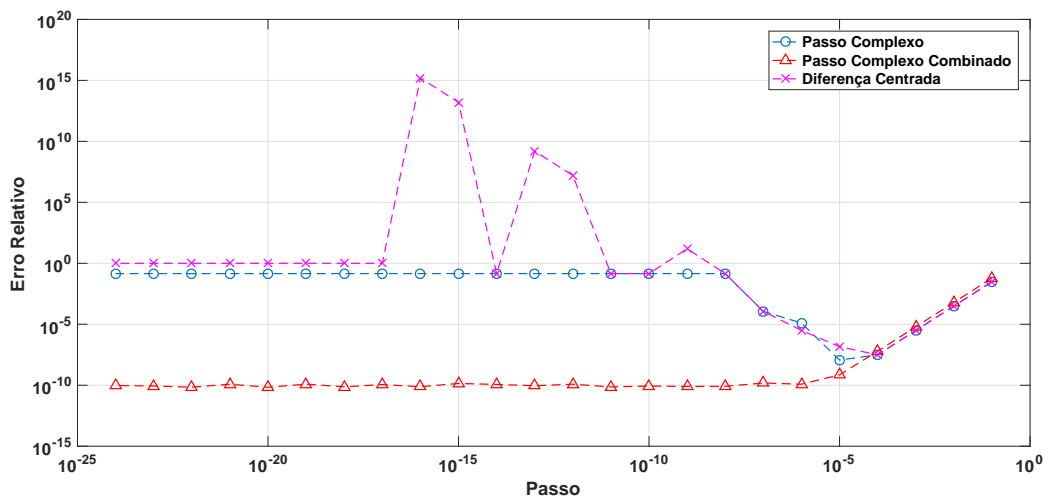
Figura 3.1: Erros relativos da primeira derivada da função (3.1) em $x = -1,74$



Podemos notar que diminuindo o passo até 10^{-5} , tanto DPC como DF centrada obtiveram bons resultados em relação a sua acurácia. Todavia, decrescendo h a partir de 10^{-5} , DPC mostra-se muito mais acurada, e mesmo DF centrada, que possuiu erro relativo mínimo perto de 10^{-10} , apresenta-se suscetível ao erros de cancelamento subtrativo nos menores passos, $f(x+h) = f(x-h) = f(x)$, ou erro de arredondamento caso numericamente se tenha $x+h = x$. Já os menores erros DPC estão na faixa de precisão de máquina.

A figura (3.2) a seguir é referente aos erros relativos da segunda derivada da função (3.1), sobre a mesma abscissa e mesmos passos que experimentados para a primeira derivada. Nela, apresentamos resultados por meio da DF de segunda ordem dada por (2.12). Já DPC com a aproximação de segunda derivada dada por (2.27) (*Passo Complexo*), e por DPC com precisão de máquina dada em (2.39) (*Passo Complexo Combinado*) e tendo adotado $g(\epsilon_m) = (\epsilon_m)^{1/3}$.

Figura 3.2: Erros relativos da segunda derivada da função (3.1) em $x = -1,74$



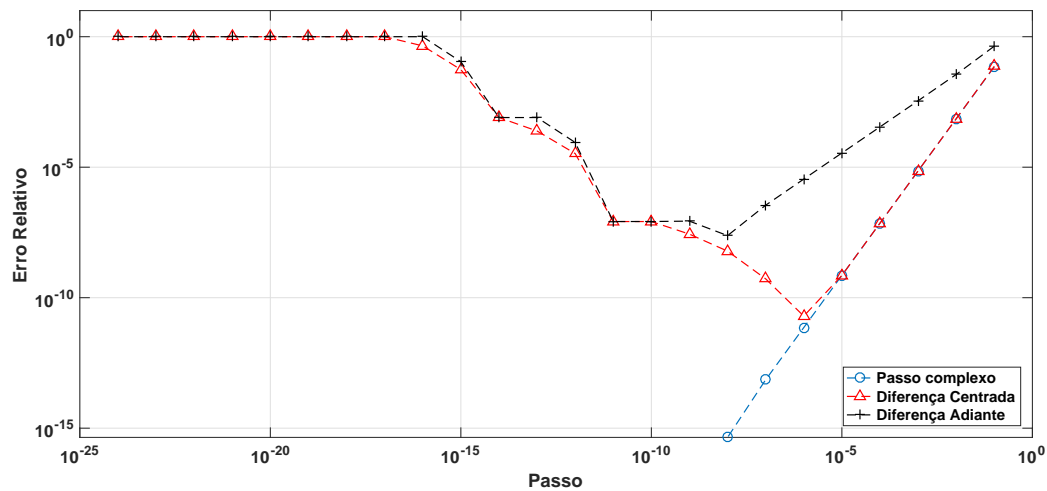
Podemos auferir que nos maiores passos, até 10^{-4} , as três aproximações tiveram resultados bem próximos e satisfatórios com $h = 10^{-4}$. Os resultados apresentados por DPC com $g(\epsilon_m)$ da precisão de máquina, a partir do passo de 10^{-5} foram ótimos em comparação a DF, o qual com passo menores do que 10^{-8} produzem erros cima de 100%. A perda de acurácia da DPC de segunda derivada em relação à de primeira é decorrente de que agora a aproximação (2.39) possui operação de diferença [1].

Agora consideremos a função que também foi utilizada no artigo [14]:

$$f(x) = \frac{e^{3x} (1 - e^x)}{\sqrt{\text{sen}^4(x) + \text{cos}^4(x)}} \quad (3.2)$$

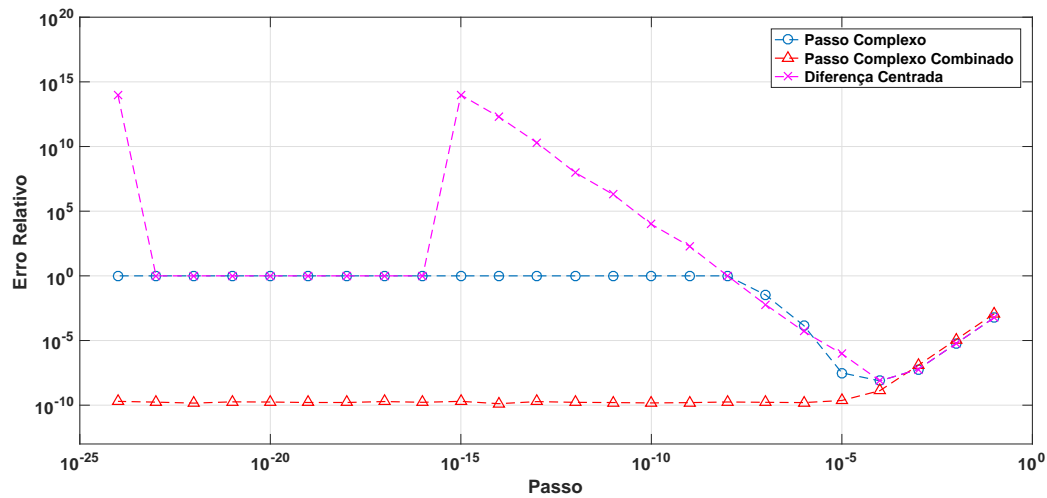
A figura (3.3) é referente aos erros relativos decorrente dos cálculos da primeira derivada de (3.2) gerados por DPC de (2.18), por DF centrada de (2.8) e por DF adiante de (2.3), e sendo calculada na abscissa $x = 0$.

Figura 3.3: Erros relativos da primeira derivada da função (3.2) em $x = 0$



Notamos que a partir dos maiores passos, até 10^{-6} , DPC e DF centrada tiveram resultados bem próximos e eficientes, contudo é notório que com passos menores do que este DF centrada tem acurácias deterioradas, e desde $h = 10^{-8}$ para baixo, tanto DF centrada quanto DF adiante possuem o mesmo nível de erro. Com passo inferior a 10^{-8} , o erro relativo da DPC foi zero, isso significa, que computacionalmente, o valor encontrado pelo método e o valor dado por (3.2) foi tão pequeno que a máquina considera iguais, logo gerando resultado zero no erro relativo, o que nos revela, pelo menos para este exemplo, o alto grau de potencialidade da DPC no cálculo da primeira derivada.

A figura (3.4) é alusiva ao cálculo da segunda derivada de (3.2), realizada pelos dois métodos DPC (fórmulas (2.27) e (2.39) com $g(\epsilon_m) = \epsilon_m^{1/3}$) e DF de segunda ordem (2.12), no mesmo ponto $x = 0$ e com os mesmos passos usados na análise da primeira derivada.

Figura 3.4: Erros relativos de segunda derivada da função (3.2) na abscissa $x = 0$ 

Podemos notar que até o passo de 10^{-4} todas as aproximações tiveram erros relativos bem próximos, mas, decrescendo deste, o método que mais uma vez tem ganho, com ordem 2, foi DPC com precisão de máquina. Assim como na função anterior, a segunda derivada com a simples DPC chega a erros de 100%, enquanto DF acima disso, quando experimentadas com pequenos incrementos h .

Observando as investigações sobre as funções (3.1) e (3.2), a simples DPC de segunda derivada (2.27) não possui o mesmo desempenho com passos pequenos que o da primeira derivada, e somente a DPC *combinada* garantiu acurácia satisfatória com esses incrementos. Este fato nos foi impressionante porque o artigo [2] faz várias comparações de precisão entre formulações de derivadas de primeira e segunda ordem, de DF e DPC, cita [1], mas não verifica a eficiência de sua formulação de segunda derivada (2.39) mediante estudo de valores do incremento nos reais, que aqui adotamos $g(\epsilon_m)$. Em consequência, avalia erros de acurácia entre as formulações de segunda derivada que ficam muito aquém das respostas que aqui encontramos. Pela desconfiança dos resultados apresentados pelos autores de [2], decidimos avaliar a segunda derivada da mesma função, calculada no mesmo ponto, e utilizada por eles:

$$f(x) = \frac{e^x}{\sin^3(x) + \cos^3(x)} \quad (3.3)$$

Numa primeira avaliação sobre ela usamos o método DPC com precisão da máquina fixa com $g(\epsilon_m) = \epsilon_m^{1/3}$ e DF centrada de segunda ordem. Mais uma vez procedemos com a análise comparativa entre essas metodologias adquirindo os dados à medida que variamos o passo, iniciando com 10^{-1} e decrescendo até 10^{-8} . Para $x = 1,5$ temos que $f''(x) = 14,568284268299991540251386342064$.

Segundo a tabela (3.1), notamos que até o passo de 10^{-4} ambos os métodos alcançam resul-

tados próximos um ao outro. Contudo, a partir desse passo, o método DPC ganha destaque na resolução alcançando melhores resultados. Mesmo não sendo mostrado na tabela, é importante destacarmos que a partir de 10^{-8} pra baixo, o método DF chega a 100% de erro. Já DPC, continua sendo estável e eficiente no resultado, mesmo com passo muitíssimos pequenos.

Tabela 3.1: Resultados da derivada de segunda ordem da função (3.3) pelo método DPC usando $g(\epsilon) = \epsilon^{1/3}$ e método DF, programados no MATLAB

Passo	Valor	Er_{DPC}	Valor	Er_{DF}
10^{-1}	14.2186	2.3996×10^{-2}	14.7500	1.2475×10^{-2}
10^{-2}	14.5647	2.4554×10^{-4}	14.5700	1.2282×10^{-4}
10^{-3}	14.5682	2.4559×10^{-6}	14.5683	1.2281×10^{-6}
10^{-4}	14.5682	2.4484×10^{-8}	14.5682	1.8257×10^{-8}
10^{-5}	14.5682	1.6304×10^{-10}	14.5682	6.0639×10^{-9}
10^{-6}	14.5682	7.6516×10^{-11}	14.5687	3.4757×10^{-5}
10^{-7}	14.5682	8.2517×10^{-11}	14.7437	1.2045×10^{-2}
10^{-8}	14.5682	7.7266×10^{-11}	17.7635	2.1933×10^{-1}

A próxima figura, (3.5), é referente aos dados apresentados por [2], em que sua equação (6) é a nossa segunda derivada por DF dada em (2.12); enquanto sua equação (40) refere-se à (2.39) sugerida por [1]. Diferentemente ao que usamos até aqui, $g(\epsilon_m) = \epsilon_m^{1/3}$, [2] usaram $g(\epsilon_m) = h$, e então relataram os resultados comparando-os com DF na variação de h . Comparativamente ao que encontramos, nem mesmo os resultados de DF nos fazem sentido algum com passo menor que 10^{-3} ! Para DPC os resultados muito distintos seriam talvez justificados na adoção de $g(\epsilon_m)$, e daí decidimos realizar a mesma investigação que [2].

A tabela (3.2) apresenta nossos valores da experimentação com as mesmas condições que [2] realizaram, mas na sintaxe MATLAB, enquanto os autores dizem ter obtido os seus em Fortran. Já que aqui $g(\epsilon) = h$, não podemos afirmar que as diferenças entre os resultados sejam da adoção de precisão de máquina simples, dupla ou quádrupla. Notamos que abaixo do passo 10^{-2} todos os resultados obtidos por DF são distintos aos de [2], e tendo quatro ordens de grandeza diferentes comparando-se os melhores resultados de ambas experimentações.

Sob comparação da DPC, as diferenças dos erros relativos são ainda maiores. Com $h = 0,01$ nosso programa retornou um erro de duas ordens inferior ao dos autores, sendo neste passo o melhor deles, na ordem de 10^{-6} . Já o erro relativo mínimo obtido por nós foi da ordem de 10^{-13} . Entretanto, diferentemente do que havíamos visualizado com $g(\epsilon_m) = \epsilon_m^{1/3}$ fixo, o erro relativo da segunda derivada de DPC não diminui e permanece praticamente constante com a diminuição do passo h . Isto nos pareceu indicar que a função de precisão de máquina $g(\epsilon_m)$ tem um papel muito importante para obtenção de excelentes resultados. Nosso uso por $g(\epsilon_m) = \epsilon_m^{1/3}$ foi decorrente de seguir o raciocínio argumentado por [1], que na verdade sugere usar $g(\epsilon_m) = \left(\frac{\epsilon_m}{2}\right)^{1/3}$, mas que o valor usado aqui foi mediante as investigações em

Figura 3.5: Dados apresentados por [2], em que a equação 6 refere-se à derivada de segunda ordem por DF centrada, e a equação 40 é a mesma fórmula (2.39)

Table 7

Second order derivative validations: $f(x) = \frac{e^x}{\sin^3(x) + \cos^3(x)}$, $x = 1.5$, $f''(x) = 14.5683$.

h	Eq. (6)	ERROR	Eq. (40)	ERROR	Eq. (41)	ERROR	Eq. (42)	ERROR	Eq. (43)	ERROR
0.1	14.75	0.012476	14.5571	0.00076655	14.5571	0.00076655	14.5571	0.00076655	14.5571	0.00076655
0.01	14.57	0.00012088	14.5683	1.083e-06	14.5683	1.083e-06	14.5683	1.083e-06	14.5683	1.083e-06
0.001	14.5697	9.4661e-05	14.569	4.6748e-05	14.569	4.6748e-05	14.569	4.6748e-05	14.569	4.6748e-05
0.0001	14.5731	0.0003319	14.5707	0.00016596	14.5707	0.00016596	14.5707	0.00016596	14.5707	0.00016596
1e-05	14.6079	0.0027185	14.5881	0.001358	14.5881	0.001358	14.5881	0.001358	14.5881	0.001358
1e-06	13.2498	0.090501	13.8934	0.046326	13.8934	0.046326	13.8934	0.046326	13.8934	0.046326
1e-07	20.8722	0.43271	17.3667	0.19209	17.3667	0.19209	17.3667	0.19209	17.3667	0.19209
1e-08	0	1	0	1	0	1	0	1	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1e-25	0	1	0	1	0	1	0	1	0	1

Tabela 3.2: Resultados da derivada de segunda ordem da função (3.3) pelo método DPC com $g(\epsilon) = h$ e o método DF, programados no MATLAB

Passo	Valor	Er_{DPC}	Valor	Er_{DF}
10^{-1}	14.5570	7.6678×10^{-4}	14.7500	1.2475×10^{-2}
10^{-2}	14.5680	7.6741×10^{-8}	14.5700	1.2282×10^{-4}
10^{-3}	14.5680	7.7955×10^{-12}	14.5680	1.2281×10^{-6}
10^{-4}	14.5680	4.7990×10^{-13}	14.5680	1.8257×10^{-8}
10^{-5}	14.5680	1.0220×10^{-11}	14.5680	6.0639×10^{-9}
10^{-6}	14.5680	1.0955×10^{-10}	14.5690	3.4757×10^{-5}
10^{-7}	14.5680	6.6810×10^{-10}	14.7440	1.2045×10^{-2}
10^{-8}	14.5680	4.5100×10^{-9}	17.7635	2.1933×10^{-1}
10^{-9}	14.5680	9.9400×10^{-8}	0.0000	1.0000×10^0
10^{-10}	14.5680	1.7750×10^{-7}	$2.6645 \times 10^{+5}$	$1.8290 \times 10^{+4}$
10^{-11}	14.5680	9.7600×10^{-7}	$2.6645 \times 10^{+7}$	$1.8290 \times 10^{+6}$
10^{-12}	14.5700	1.1575×10^{-4}	$8.8820 \times 10^{+8}$	$6.0970 \times 10^{+7}$

todas as nossas avaliações. Então nos vieram os questionamentos: o que fez gerar resultados tão discrepantes, tanto em DF quanto em DPC, entre os apresentados por [2] e os nossos? Esses diferentes resultados foram decorrentes da programação em linguagens diferentes? A nós somente foi possível investigar a última pergunta. Daí confeccionamos um código em Fortran para a mesma experimentação.

A tabela (3.3) abaixo apresenta os resultados de DF, DPC com $g(\epsilon_m) = h$ e DPC com $g(\epsilon_m) = \epsilon_m^{1/3}$ na derivada de segunda ordem da função (3.3) em $x = 1,5$, sendo todos os métodos

programados no Fortran. Todos os valores confirmam aqueles obtidos pelos programas feitos no MATLAB. E daí podemos tecer conclusões. Uma delas é que, pelo menos para essa função aplicada, os resultados não dependem da linguagem de programação, pois não são muitos diferentes os erros relativos. Outra, é que foi infeliz a adoção dos autores do artigo [2] em somente apresentar as resposta do incremento real, $g(\epsilon_m)$, sendo igual ao próprio passo h da parte imaginária. Os resultados dão impressão que a DPC de segunda derivada (2.39) tem comportamento análogo ao da derivada por DF, ou seja, que existiria um passo não muito pequeno que retornaria um melhor erro de aproximação! Onde na verdade, fixado $g(\epsilon_m)$, os erros diminuem e se permanecem praticamente constantes qualquer que seja o passo muito pequeno que se use. E, sendo tal erro, com $g(\epsilon_m) = \epsilon_m^{(1/3)}$, bastante inferior aquele melhor que se obtém por DF. Há no entanto uma coisa que não conseguimos justificar: os diferentes resultados entre os de [2] e os nossos em DF de segunda ordem nos passos de 10^{-7} a 10^{-4} .

Tabela 3.3: Resultados da derivada de segunda ordem da função (3.3) pelo método DF, DPC com $g(\epsilon) = h$ e DPC com precisão de máquina $g(\epsilon_m) = \epsilon^{1/3}$, programados no Fortran

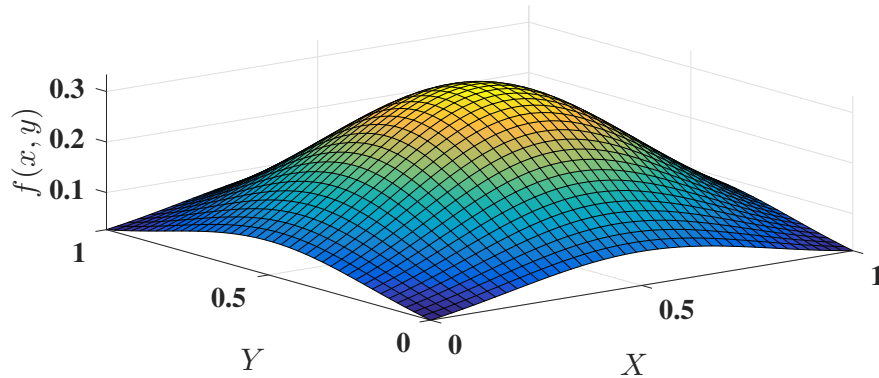
Passo	Er_{DF}	$Er_{DPC} (g(\epsilon_m) = h)$	$Er_{DPC} (g(\epsilon_m) = \epsilon_m^{1/3})$
10^{-1}	1.2191×10^{-2}	7.6679×10^{-4}	2.3997×10^{-2}
10^{-2}	1.2282×10^{-4}	7.6741×10^{-8}	2.4554×10^{-4}
10^{-3}	1.2281×10^{-6}	7.8058×10^{-12}	2.4560×10^{-6}
10^{-4}	1.8257×10^{-8}	8.4719×10^{-13}	2.4484×10^{-8}
10^{-5}	1.0640×10^{-9}	1.0223×10^{-11}	1.6304×10^{-10}
10^{-6}	3.4557×10^{-5}	9.5015×10^{-11}	7.8918×10^{-11}
10^{-7}	1.2045×10^{-2}	3.0471×10^{-10}	7.6518×10^{-11}
10^{-8}	2.1933×10^{-1}	2.2390×10^{-9}	8.1019×10^{-11}
10^{-9}	1.0000×10^0	1.1359×10^{-7}	8.1956×10^{-11}
10^{-10}	1.8288×10^4	1.7747×10^{-7}	7.7854×10^{-11}
10^{-11}	1.8290×10^6	1.2420×10^{-6}	8.4447×10^{-11}
10^{-12}	6.0966×10^7	1.1575×10^{-4}	8.2615×10^{-11}

3.2 Em Funções Bidimensionais.

As duas funções que tomamos para fazer estudo nesta aplicação foram retiradas do trabalho de [10]. Nessas duas funções uma malha uniforme de 36×36 pontos, no domínio $[0, 1] \times [0, 1]$, foi usada para determinar numericamente as derivadas. Apesar de podermos apresentar as respostas tanto da diferenciação parcial em x quanto em y , vimos que os resultados comparativos foram semelhantes, e por essa razão omitimos as avaliações para derivadas em y .

A primeira função é denominada de *gentle*, e cuja expressão analítica é:

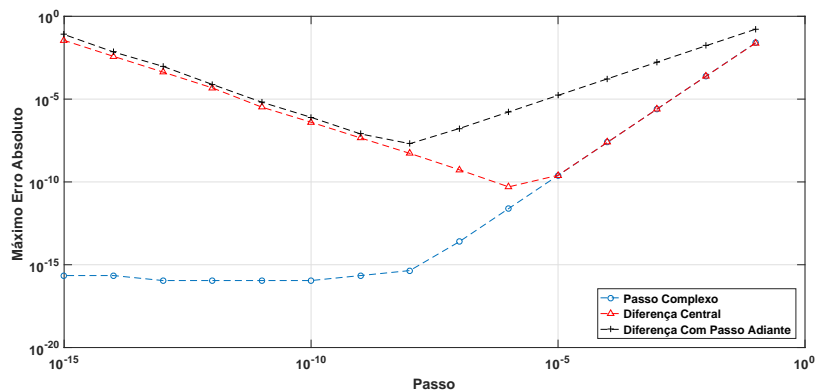
$$f(x, y) = \frac{1}{3} e^{-5,0625 \cdot [(x-0,5)^2 + (y-0,5)^2]} \quad (3.4)$$

Figura 3.6: Função *gentle*

Podemos notar através da figura (3.6) que essa função não possui grande variação de amplitude em seu pequeno intervalo de domínio, e também não possui mudanças abruptas em sua imagem; podemos então dizer que ela tem um comportamento suave.

Pelo fato de podermos avaliar os erros nas obtenções das derivadas mediante a variação do passo h , investigamos o menor dentre todos os maiores erros absolutos obtidos nas aproximações de todos os pontos do grid do domínio¹. Esse estudo está apresentado na figura (3.7) abaixo. Com o incremento máximo de $h = 1 \times 10^{-1}$, vemos que diminuindo, até chegar a 10^{-8} , cada vez mais acurada fica a DPC, e a partir disso, diminuindo h , praticamente o erro máximo fica constante. Em contrapartida, DF é irregular, e o valor de passo 10^{-8} foi o de menor máximo erro absoluto em DF adiante, e sendo DF centrada com passo de 10^{-6} . Apesar desses menores máximos erros absolutos de DF estarem retratando respostas relativamente aceitáveis, os passos correspondentes foram obtidos empiricamente comparando-se com as soluções analíticas das derivadas.

Figura 3.7: Máximos erros absolutos referentes à variação do passo h nas aproximações das derivadas parciais da função *gentle* sobre o domínio $[0, 1] \times [0, 1]$



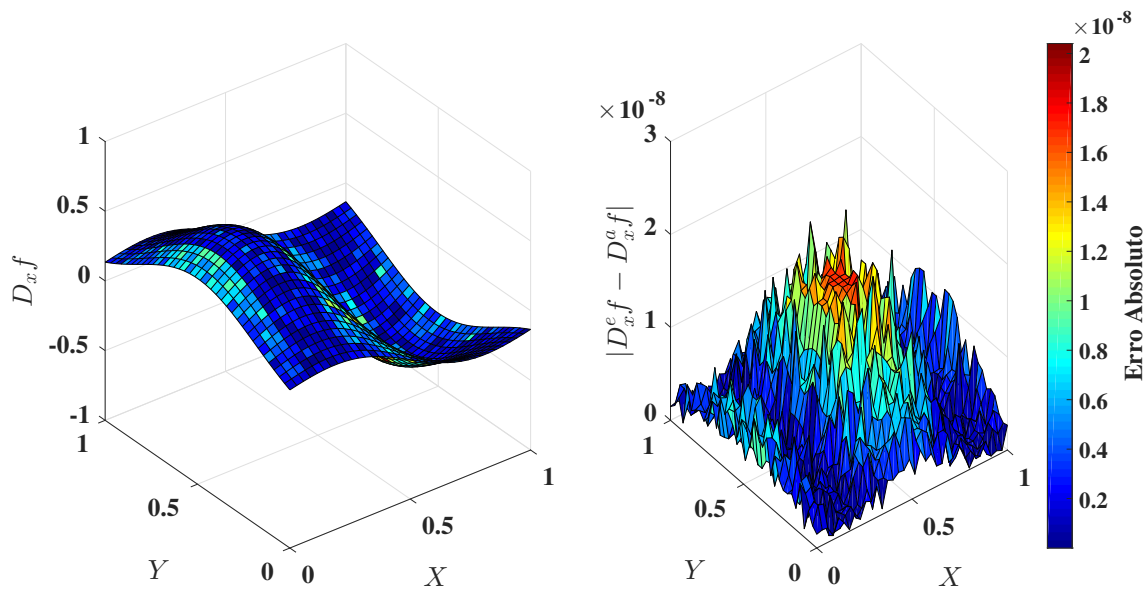
A figura (3.8) a seguir apresenta as superfícies obtidas como sendo das derivadas da função

¹Não usamos o erro relativo nesta aplicação porque as derivadas analíticas são nulas em alguns pontos!

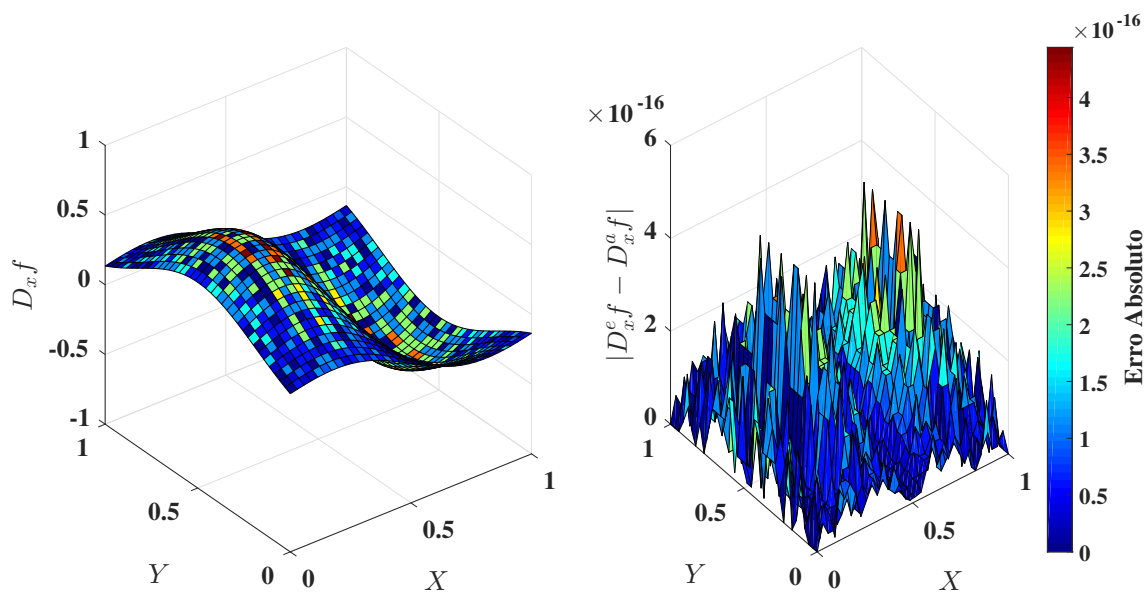
gentle em relação a variável x , com DF adiante e DPC, respectivamente; acompanhadas dos erros absolutos dessas aproximações quando são assumidas em ambas técnicas o passo $h = 1 \times 10^{-8}$. Vemos que a diferença entre os maiores erros absolutos entre as técnicas foi de praticamente 8 ordens de grandeza, o que retrata a grande acurácia de diferenciação com DPC.

Figura 3.8: Derivada parcial da função *gentle* em relação a x por DF e DPC (com passo $h = 10^{-8}$) e seus respectivos erros absolutos

(a) Aproximação de $\frac{\partial f}{\partial x}$ (*f gentle*) por DF e os erros absolutos.



(b) Aproximação de $\frac{\partial f}{\partial x}$ (*f gentle*) por DPC e os erros absolutos.

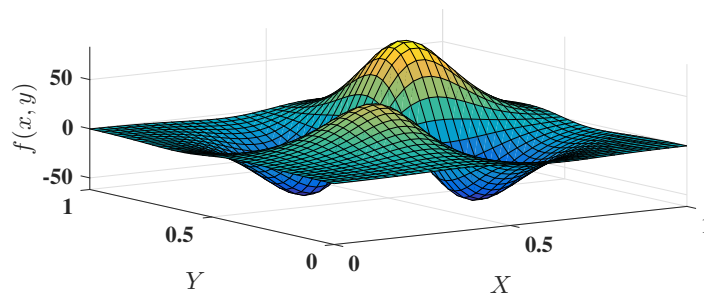


A fim de verificar o que ocorre em funções com mudanças abruptas na imagem, consideramos a função *cloverleaf* abaixo:

$$f(x, y) = 6,4 \times 10^7 \cdot \left[e^{\frac{1}{3}(10-20x)} - \frac{2}{e^{\frac{1}{3}(10-20x)} + 1} \right] \cdot \left[e^{\frac{1}{3}(10-20y)} - \frac{2}{e^{\frac{1}{3}(10-20y)} + 1} \right] \cdot e^{\frac{2}{3}(10-20x) + \frac{2}{3}(10-20y)} / \left\{ 729 \left[e^{\frac{1}{3}(10-20x)} + 1 \right]^5 \cdot \left[e^{\frac{1}{3}(10-20y)} + 1 \right]^5 \right\} \quad (3.5)$$

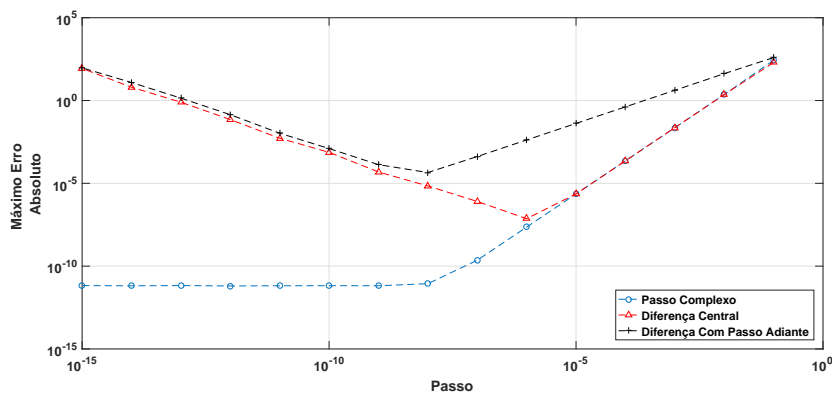
A figura (3.9) apresenta a superfície dessa função e nela vemos uma grande variação das imagens no seu pequeno intervalo de domínio, o que pode levar a imprecisões nos cálculos das derivadas, principalmente nas regiões de maior inclinação da função.

Figura 3.9: Função *cloverleaf*



A mesma estratégia que usamos para a função *gentle* fizemos também para a função *cloverleaf* quanto à avaliação dos melhores passos em DF para apresentação da acurácia da derivada primeira. Vemos na figura (3.10) que à medida que nos afastamos de $h = 10^{-8}$ DF adiante piora em acurácia. Esse é o mesmo comportamento em DF centrada, mas sendo o melhor resultado com $h = 10^{-6}$. Já em DPC, a diminuição no maior erro absoluto acontece com a diminuição do passo até o nível de 10^{-8} , e a partir daí ele se apresenta praticamente constante.

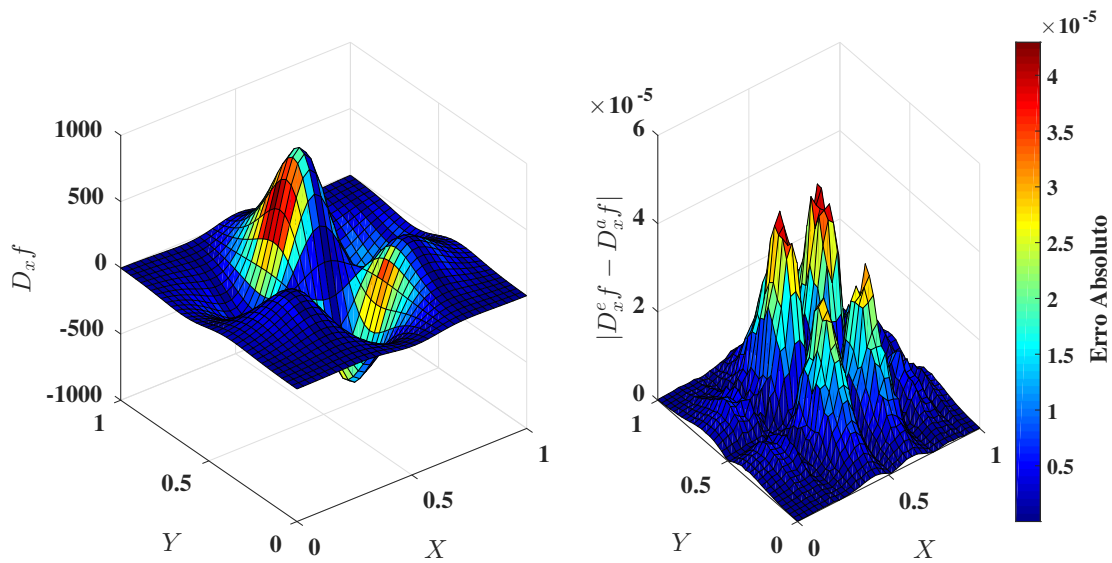
Figura 3.10: Máximos erros absolutos referentes à variação do passo h nas aproximações das derivadas parciais da função *cloverleaf* sobre o domínio $[0, 1] \times [0, 1]$



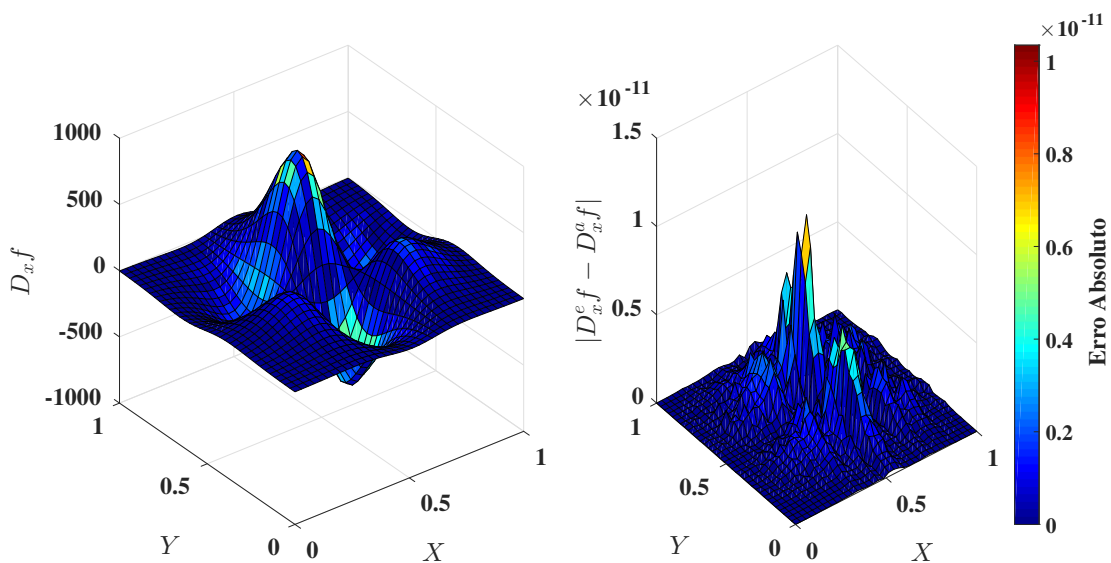
Comparativamente à função *gentle*, vemos na figura (3.11) que as derivadas da função *cloverleaf* apresentam maiores erros absolutos. Ambas técnicas decaem em acurácia, demandadas principalmente nas regiões de maior declividade da função. Ainda assim, com passo $h = 10^{-8}$ o maior erro produzido por DF adiante foi de $4,5 \times 10^{-5}$, enquanto que para DPC, com o mesmo passo, o máximo erro absoluto foi de 10^{-11} , ou seja, superior em acurácia em 6 ordens de grandeza.

Figura 3.11: Derivada parcial da função *cloverleaf* em relação a x por DF e DPC (com passo $h = 10^{-8}$) e seus respectivos erros absolutos

(a) Aproximação de $\frac{\partial f}{\partial x}$ (f *cloverleaf*) por DF e os erros absolutos.



(b) Aproximação de $\frac{\partial f}{\partial x}$ (f *cloverleaf*) por DPC e os erros absolutos.



3.3 Em Método Iterativo sobre Função de Uma Variável

Com intuito de avaliar não só acurácia, mas também tempo computacional de DF e DPC para as derivadas de primeira e segunda ordem, decidimos, inicialmente, aplicar um método iterativo que requeira essas derivadas utilizando-se apenas uma função escalar de uma variável.

Pensando na técnica de DF, possivelmente nos seria conveniente saber de antemão qual o melhor passo a ser usado para obtenção da primeira derivada e para obtenção da segunda de uma dada função. No caso de coincidência de valor do passo h esperaríamos que o procedimento iterativo retornasse a maior acurácia possível, e também no menor tempo possível dentro de DF. No entanto, nas aplicações em geral sequer dispomos de expressões analíticas da função a ser usada. Por isso, para a primeira derivada aproximada usamos a formulação DF centrada da equação (2.8), por se mostrar a melhor aproximação de derivada nessa ordem nas investigações das seções 3.2 e 3.1; enquanto para a segunda derivada de DF, como só consideramos uma única aproximação, usaremos (2.12).

Vimos na seção 3.1 acima que a segunda derivada por DPC é mais acurada quando computada com a aproximação de precisão de máquina (referendada como *Passo Complexo Combinado*) dada pela fórmula (2.39) e usando $g(\epsilon_m)$ fixo, e por isso a usamos aqui. Já para a primeira derivada com DPC, permaneceremos com o uso da fórmula (2.18).

Como também utilizaram os autores do artigo [14], usaremos o método iterativo de Halley de determinação de raízes de funções $f(x)$ não lineares:

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2[f'(x_n)]^2 - f(x_n)f''(x_n)}, \quad (3.6)$$

sendo x_n o valor de aproximação de uma raiz na n -ésima iteração e x_{n+1} na $n+1$ -ésima iteração.

Agora, pensando em número de avaliações de funções, dadas por DF e DPC, em cada iteração do método de Halley, verificamos que são três para DF ($f(x)$, $f(x+h)$ e $f(x-h)$), e são quatro em DPC ($f(x)$, $f(x+ih)$, $f(x+g(\epsilon_m)+ih)$ e $f(x-g(\epsilon_m)+ih)$). E assim, se verificarmos que outras aproximações que usam passo complexo tem mais requerimentos avaliativos de funções, de modo que os resultados são similares ou piores que os encontrados por nós, mostramos que nossa adoção possui vantagem computacional em tempo. Não podemos afirmar o mesmo na comparação de DPC com qualquer aproximação por DF, porque do ponto de vista computacional, usamos em DPC não só operações aritméticas no eixo real.

Os autores do artigo [14] formularam para DPC as aproximações²

$$f'(x) \approx \frac{\text{Im} \left\{ 8[f(x + h\sqrt{i}/2) - f(x + h\sqrt{i^5}/2)] - [f(x + h\sqrt{i}) - f(x + h\sqrt{i^5})] \right\}}{3h\sqrt{2}} \quad (3.7)$$

$$f''(x) \approx \frac{\text{Im} \left\{ 64[f(x + h\sqrt{i}/2) + f(x + h\sqrt{i^5}/2)] - [f(x + h\sqrt{i}) + f(x + h\sqrt{i^5})] \right\}}{15h^2} \quad (3.8)$$

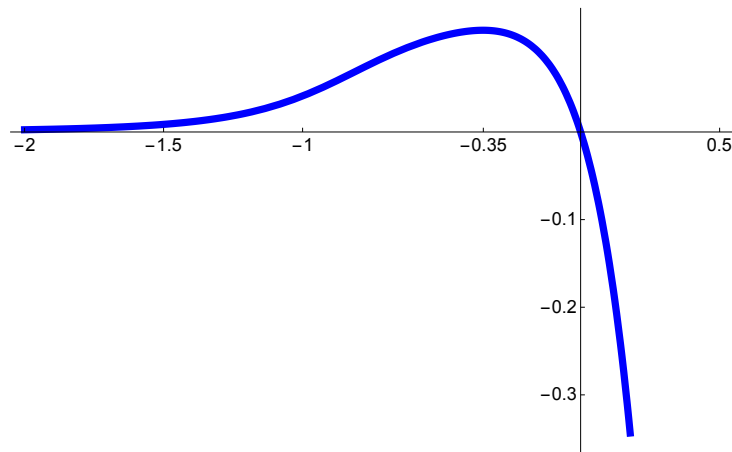
e comparam seus resultados com DF de derivada de primeira e segunda ordens dadas pela regra simétrica de quatro pontos e de cinco pontos³:

$$f'(x) \approx \frac{f(x - 2h) - 8f(x - h) + 8f(x + h) - f(x + 2h)}{12h} \quad (3.9)$$

$$f''(x) \approx \frac{-f(x - 2h) + 16f(x - h) - 30f(x) + 16f(x + h) - f(x + 2h)}{12h^2} \quad (3.10)$$

A função testada foi a (3.2), com gráfico na figura (3.12), da qual facilmente notamos que $x = 0$ é raiz. Com solução inicial $x = 5$ e uma tolerância absoluta de 10^{-15} , os autores afirmaram que com passos decrescentes de 10^{-1} a 10^{-7} , ambos métodos convergem ao usar (3.7), (3.8), (3.9) e (3.10), e sendo a convergência mais rápida com DPC. Com nossas aproximações de derivadas formuladas aqui, realizamos o mesmo teste e verificamos que a convergência é praticamente a mesma, e inclusive com o mesmo número de iterações.

Figura 3.12: Gráfico da função (3.2) na proximidade da raiz



Fixado $h = 10^{-8}$, o trabalho [14] apresenta uma tabela com os valores resultantes das 15 primeiras iterações de DPC e DF, mostrando que DF com (3.9) e (3.10) se quer tem convergência no método de Halley. No entanto, nós avaliamos esse caso e nos certificamos que DFs por nós adotadas convergem e com uma iteração a menos que a DPC aqui formulada (verifique a tabela

²Os erros de truncamento dessas aproximações são respectivamente iguais a $\frac{h^4 f^{(5)}(x)}{120}$ e $\frac{h^8 f^{(10)}(x)}{1814400}$

³cujos erros de truncamento são $\frac{h^4 f^{(5)}(x)}{30}$ e $\frac{h^6 f^{(6)}(x)}{90}$, respectivamente

3.4). E além disso, nossas fórmulas de DPC retornam, até a décima terceira iteração do método de Halley, com as mesmas quatro casas decimais do artigo; significando portanto que o custo computacional demandado pela formulação dos autores é um pouco maior que com a formulação adotada aqui, já que com uso (3.7) e (3.8) dentro do método de Halley se tem 5 avaliações de funções em cada iteração. O mesmo podemos dizer quanto à adoção das fórmulas (3.9) e (3.10) em DF, elas requerem junto a cada iteração no método Halley 5 avaliações de funções, contra apenas três demandadas no uso da formulação que aqui usamos!

Tabela 3.4: Desempenho de DF e DPC na determinação da raiz $x = 0$ da função (3.2) pelo método iterativo de Halley, com tolerância absoluta de 1×10^{-15} , usando aproximação inicial $x = 5$ e passo fixo $h = 10^{-8}$

Iteração	x_{dpc}	x_{df}
1	4.5246	4.6050
2	3.8886	4.3417
3	3.4971	4.1193
4	3.0442	3.6718
5	2.4493	3.4800
6	2.02073	1.9807
7	1.6061	1.7586
8	1.0975	1.3824
9	5.9466×10^{-1}	9.5138×10^{-1}
10	2.9241×10^{-1}	-3.3984×10^{-2}
11	6.6074×10^{-2}	-2.72133×10^{-4}
12	1.2732×10^{-3}	-4.2227×10^{-9}
13	1.0464×10^{-8}	-4.7412×10^{-17}
14	4.4778×10^{-17}	-4.7412×10^{-17}

Há também comentário no artigo [14] que diminuindo o passo h , de 10^{-8} a 10^{-15} o método de Halley converge com as fórmulas (3.7) e (3.8) com menos de 15 iterações, enquanto que com $h = 10^{-16}$ se obtém a convergência em torno de 40. Mais resultados bastante distintos do que encontramos! Em DPC usada aqui, decrescendo h a partir de 10^{-5} e indo até 10^{-25} , todos alcançaram convergência com critério do erro absoluto em somente 14 iterações. Já com DF, o melhor resultado aconteceu com 10^{-8} , também em 14 iterações; e com 10^{-9} se chegou a 1000 iterações sem convergir; mas, com 10^{-10} , 10^{-11} e 10^{-12} , DF convergiu novamente, tendo 193, 408 e 362 iterações, respectivamente. Passando disso chegaram-se a 1000 iterações ou as investigações retornaram *NaN* (not a number).

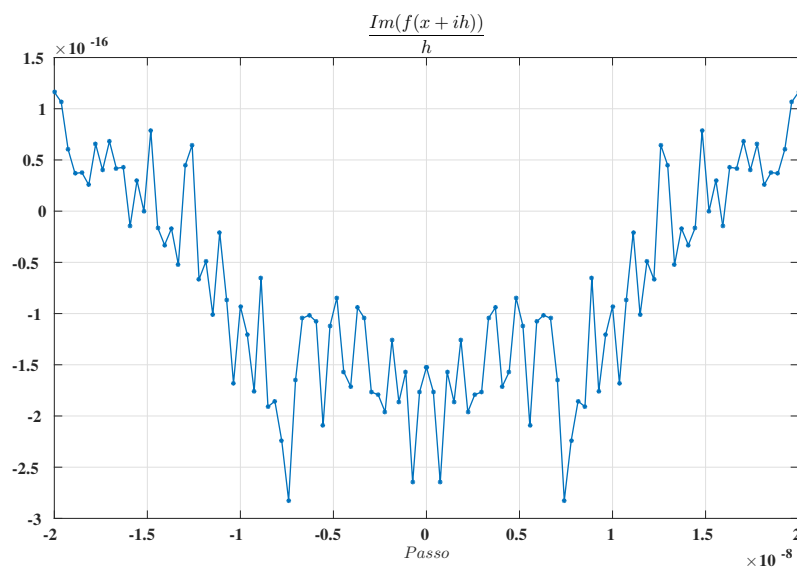
Nessa experimentação, até onde ambas técnicas convergiram ($h = 10^{-8}$), as soluções estavam na mesma ordem de grandeza e alcançadas com praticamente os mesmos números de iterações, tendo portanto pequeníssimas diferenças de tempo computacional. Isso aconteceu mesmo quando experimentamos dar como solução inicial positiva um valor mais distante da

solução verdadeira.

Observando mais atentamente a expressão do método de Halley com a função a qual desejamos encontrar a raiz, poderemos averiguar mais detalhadamente a técnica de diferenciação sobre a primeira derivada com DPC. O ponto de máximo da função (3.2) fica com abscissa próxima a 0.35, na verdade, o máximo ocorre aproximadamente quando $x = -0.3511657776411328596$. Vemos nesse ponto que $f(x)$ é positiva, e também nesse ponto a concavidade da curva é para baixo, ou seja, a segunda derivada aí é negativa. Sendo assim, na proximidade do ponto de máximo, o denominador da segunda parcela do método de Halley é positivo, e aí, dependendo do sinal da primeira derivada sobre a abscissa x_n levemente à direita ou à esquerda do ponto de máximo, o método de Halley pode convergir ou não, pois o valor x_{n+1} de uma próxima iteração seria em direção a raiz nula da função, ou se afastando dela.

Avaliamos a convergência do método de Halley usando DF e DPC para um valor inicial da raiz como sendo $x_1 = -0.3511657776411327$, que fica a direita da abscissa de ponto de máximo. Verificamos que diminuindo o passo, começando de 10^{-1} e indo até 10^{-6} , DF convergia para a solução, mas DPC nem mesmo até 10^{-7} . Somente com passo menor do que ou igual a 10^{-8} a raiz foi encontrada por DPC. A figura (3.13) apresenta a razão pela qual o método de Halley com DPC teve tal comportamento. Vemos que h um pouco acima de 10^{-8} faz com que os valores da primeira derivada sejam positivos; e dentro do intervalo $[-10^{-8}, 10^{-8}]$ a primeira derivada é sempre aproximada por valor negativo, que é o que corresponde corretamente à inclinação da reta tangente à direita do ponto de máximo da função. Isso ilustra a justificativa dos resultados mostrados até agora das aplicações com a DPC; as melhores aproximações acontecem quando o passo é muito pequeno.

Figura 3.13: Valores da primeira derivada da função (3.2) em x à direita próximo da abscissa de seu ponto de máximo



A adoção do passo muito pequeno também pode sustentar a validade da aproximação das derivadas em um ponto z_0 , já que a DPC só é válida se em torno desse ponto a função não possuir singularidades. Essa função (3.2) também é um bom exemplo da aplicação da DPC. Como mostra a figura (3.14) ela não é analítica no plano complexo, possui descontinuidades em várias regiões, mas é analítica em torno do ponto de máximo da função quando definida nos reais. É até mais; conforme vemos na figura (3.15), mesmo que os sinais da parte imaginária da função mudem bruscamente na vizinhança do eixo real, ela é contínua nessa região. Os pontos de ramificação, onde se apresentam as descontinuidades, destacadas em vermelho, estão relativamente longe do eixo real, de modo que usando h muito pequeno não corremos riscos de não validade da DPC.

Figura 3.14: Parte imaginária da função (3.2) no plano complexo

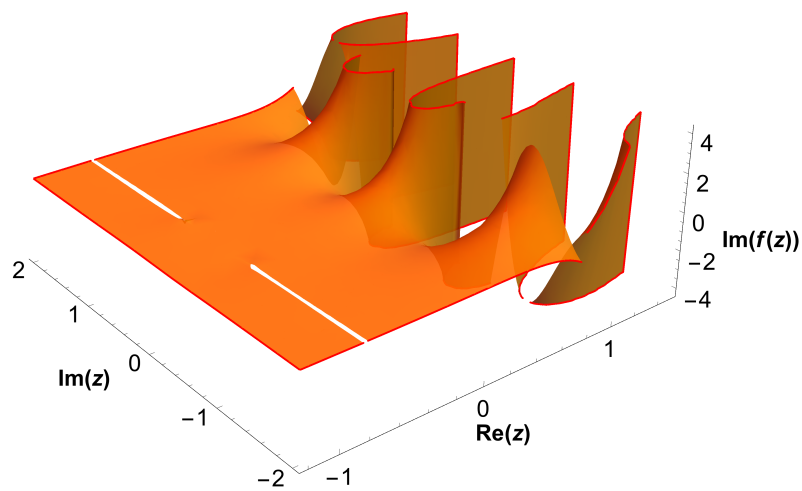
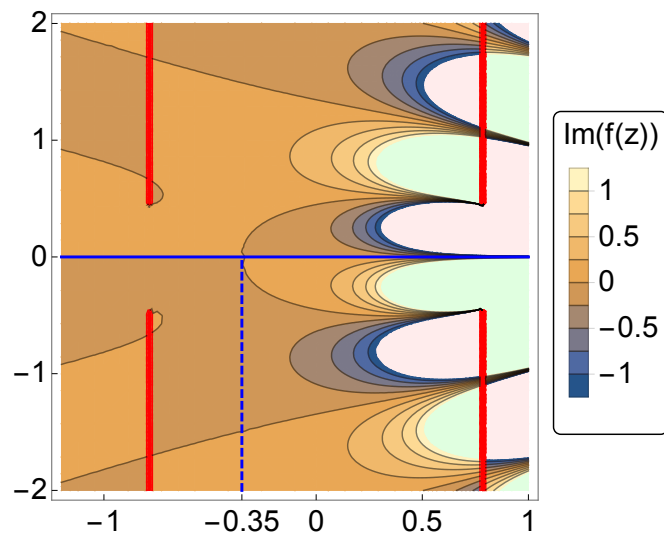


Figura 3.15: Curvas de nível e singularidades da parte imaginária da função (3.2) no plano complexo



3.4 Em Método Iterativo de Resolução de Sistemas Não-lineares

Nesta seção aplicamos o método de DPC, e DF, na obtenção das derivadas parciais de primeira ordem para construção de matrizes jacobianas. A fim de investigar o tempo demandado por cada método, decidimos avaliá-los em um procedimento que requeira, de certo modo, um maior custo computacional que o aplicado na seção 3.3 anterior. Para isso, usamos o método de Newton na resolução de sistemas não lineares.

No método de Newton, a matriz jacobiana tem papel fundamental no desenvolvimento do método. A jacobiana é a matriz formada pelas derivadas parciais de primeira ordem de uma função vetorial $F : R^n \rightarrow R^m$, com $F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ e $\mathbf{x} = (x_1, x_2, \dots, x_n)$ sendo o vetor das n variáveis. A representação matricial da derivada de F em cada variável, quando existe, é então definida como sendo a matriz jacobiana J :

$$J(F(x_1, \dots, x_n)) = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}' = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \frac{\partial f_m(x)}{\partial x_2} & \dots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix} \quad (3.11)$$

Para cômputo nas técnicas de DF e DPC, cada elemento então da jacobiana será aproximado com a aplicação das fórmulas desenvolvidas por cada técnica, sendo o passo, costumeiramente o mesmo adotado em cada derivada parcial.

3.4.1 Método de Newton em sistemas $n \times n$ de equações não lineares

O Método de Newton é provavelmente o método mais conhecido para a obtenção da solução de sistemas de equações não-lineares. Consideremos um sistema de n funções escalares f_1, f_2, \dots, f_n , que compõem $F : R^n \rightarrow R^n$, em x_1, x_2, \dots, x_n , e tais que:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \quad (3.12)$$

Qualquer método numérico para obtenção da solução desse sistema consiste então em obter variáveis aproximadas x'_1, x'_2, \dots, x'_n que satisfaçam esse conjunto de equações. Uma forma usual

para obtenção do método de Newton é através da expansão de Taylor. Tomando a série de Taylor de funções vetoriais em torno de uma aproximação \mathbf{x}_k teremos:

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x}_k) + J(\mathbf{x}_k)\mathbf{h} + O(\|\mathbf{h}\|_2^2) \quad (3.13)$$

em que $\mathbf{h} = \mathbf{x} - \mathbf{x}_k$, e cuja essa expressão foi tomada até o termo linear. Então, em vez de buscar a solução do sistema $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ dado em 3.12, o método de Newton, de aproximação de primeira ordem, consiste em solucionar o sistema linearizado:

$$\mathbf{F}(\mathbf{x}_k) + J(\mathbf{x}_k)\mathbf{h} = \mathbf{0}. \quad (3.14)$$

Agora pensando num processo iterativo, obtemos um passo \mathbf{h} na direção de uma melhor aproximação solucionando a equação matricial $J(\mathbf{x}_k)\mathbf{h} = -\mathbf{F}(\mathbf{x}_k)$, ou, em termos de uma melhor aproximação \mathbf{x}_{k+1} dentro processo iterativo podemos escrever:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{J}^{-1}(\mathbf{x}_k)\mathbf{F}(\mathbf{x}_k) \quad (3.15)$$

Esse é então o procedimento iterativo do método de Newton para resolução de sistemas não lineares do tipo 3.12.

O método de Newton pode convergir rapidamente para a solução pois converge de forma quadrática ([9]). Mas pode nem obter soluções se não for dada uma boa aproximação inicial. Ele é um método de otimização de busca de mínimo, ou máximo local e assim depende fortemente da aproximação inicial informada. Além disso, cuidados computacionais nas derivadas devem ser levados em conta, pois em cada iteração o método de Newton requer a inversa da matriz jacobiana, que pode se apresentar sendo próxima de singular, ou mesmo singular, acarretando em soluções ruins ou mesmo sem obtenção de solução.

Como qualquer método iterativo, na implementação do método de Newton devemos usar pelo menos um critério de parada na obtenção da solução. Considerando que \mathbf{x}_k e \mathbf{x}_{k+1} são duas aproximações sucessivas, podemos encontrar os seguintes critérios de parada na literatura:

- i) Do erro absoluto: O processo é interrompido se após uma iteração $k + 1$ as soluções aproximadas \mathbf{x}_k e \mathbf{x}_{k+1} satisfazem a norma de suas diferenças ser menor do que um ε_{abs} , ou seja⁴:

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 < \varepsilon_{abs}, \text{ com } \varepsilon_{abs} > 0.$$

Evitamos este critério porque caso uma solução exista em um ponto muito distante da origem e a tolerância ε_{abs} seja muito exigente, é possível que o processo requeira um número de iterações muito grande, sendo que as iterações anteriores já deem aproximações

⁴Mas é claro que se já conhecermos o vetor solução, o critério é aplicado fazendo-se $\mathbf{x}_k = \mathbf{x}_{sol}$.

bem satisfatórias⁵. Isto é, o erro absoluto não leva em consideração a ordem de grandeza do valor calculado ([5]).

ii) Do erro relativo: O processo é interrompido quando satisfizer:

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2}{\|\mathbf{x}_{k+1}\|} < \varepsilon_{rel}, \text{ com } \varepsilon_{rel} > 0$$

Neste caso, para evitar problemas com a norma do vetor nulo, ou próximo do nulo, no denominador, podemos usamos a forma alternativa:

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 < \varepsilon_{rel} \cdot \max(1, \|\mathbf{x}_{k+1}\|_2).$$

Se caso o vetor nulo apareça, essa forma recai na verificação do erro absoluto.

iii) Como o sistema 3.12 tem matricialmente o vetor nulo como termo independente, podemos requerer que uma boa solução aproximada \mathbf{x}_{k+1} seja quando encontrarmos $\mathbf{F}(\mathbf{x}^{k+1})$ próximo do vetor nulo, isto é:

$$\|\mathbf{F}(\mathbf{x}_{k+1})\| < \varepsilon_{nf}, \text{ com } \varepsilon_{nf} > 0.$$

iv) Caso o processo ainda não satisfaça nenhum dos critérios anteriores é prudente exigir que o método iterativo seja interrompido para um número máximo de iterações.

Na implementação computacional do método de Newton fizemos uso dos critérios (ii), (iii) e (iv).

3.4.2 Aplicação em Exemplo de Sistemas Não Lineares

Consideremos o seguinte sistema não linear:

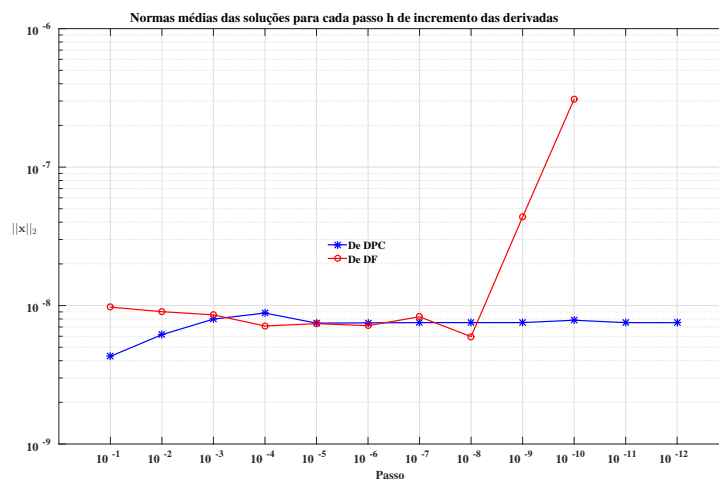
$$\begin{cases} e^{x^2+y^2} - 1 = 0 \\ e^{x^2-y^2} - 1 = 0 \end{cases} \quad (3.16)$$

Facilmente vemos que a solução é $\mathbf{x} = [0; 0]^t$. Para comparação entre os resultados obtidos pelo método de Newton com a jacobiana calculada com DF e DPC, usamos como aproximação inicial $\mathbf{x} = [3,5; 3,5]^t$, o número máximo de iterações permitido $n_{iter} = 100$, a norma mínima para a função vetorial, de pelo menos $nf = 10^{-12}$ e a tolerância relativa, que neste caso é também a absoluta do passo, de $\varepsilon_{rel} = 10^{-14}$.

⁵A título de ilustração pensemos no caso escalar de determinação da raiz 2000 da função $f(x) = (x-1)(x-2000)$, usando ε_{abs} muito pequeno! ([9])

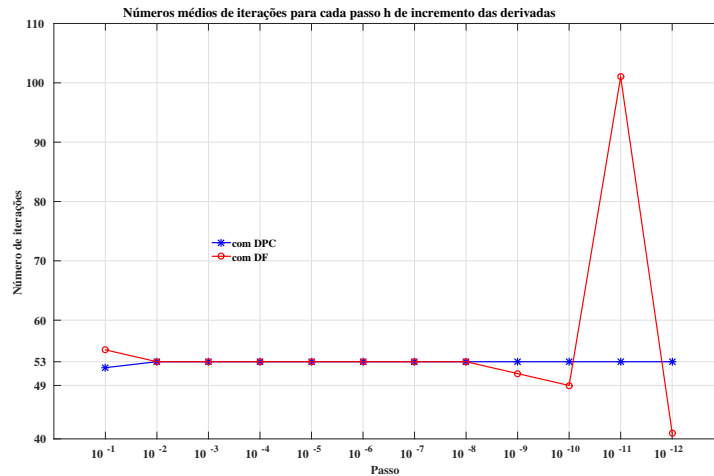
Variando-se o passo h de incremento para as derivadas, de 10^{-12} a 10^{-1} , rodamos os programas do método de Newton 100 vezes para cada passo h , em cada aproximação da jacobiana obtida por DF centrada e obtida por DPC. As figuras (3.16) e (3.17) apresentam as médias das normas da solução do sistema 3.16, e as médias de iterações obtidas com os cálculos dessa investigação. Até o incremento $h = 10^{-8}$ o método de Newton alcançou tanto com DF quanto DPC uma solução aproximada sob o critério de parada do erro absoluto sendo nulo, ou seja, quando \mathbf{x}_{k+1} foi igual \mathbf{x}_k . Vemos que quando o incremento foi 10^{-9} e 10^{-10} a solução usando DF diminuiu sua acurácia de uma e duas ordens, respectivamente. Nestes casos, para o último passo dentro do método de Newton não foi possível obter a inversa da jacobiana, já que ela era singular, e o código então retornou uma solução anterior a esse acontecimento. Para h sendo 10^{-11} e 10^{-12} a jacobiana foi, em algumas das 100 repetições com uso de DF, singular ou mal-condicionada e assim não se obteve solução aproximada; com o código alcançando o número máximo de iterações com $h = 10^{-11}$, e média de número de iterações um pouco maior que 40 quando $h = 10^{-12}$. Em contrapartida, registramos que o método de Newton calculando a jacobiana com DPC, sempre teve o critério de parada conseguido pela norma nula do passo dentro da busca de mínimo da solução aproximada. Vemos que suas soluções foram sempre com norma na mesma ordem de grandeza que as melhores respostas obtidas com cálculos da jacobiana por DF centrada, e tendo também médias de mesmo números de iterações na extensão das respostas confiáveis obtidas por DF.

Figura 3.16: Valores médios de $\|\mathbf{x}\|$ da resolução do sistema 3.16 pelo método de Newton, obtidas com aproximação da jacobiana por DF centrada e DPC sob variação do incremento para as derivadas.



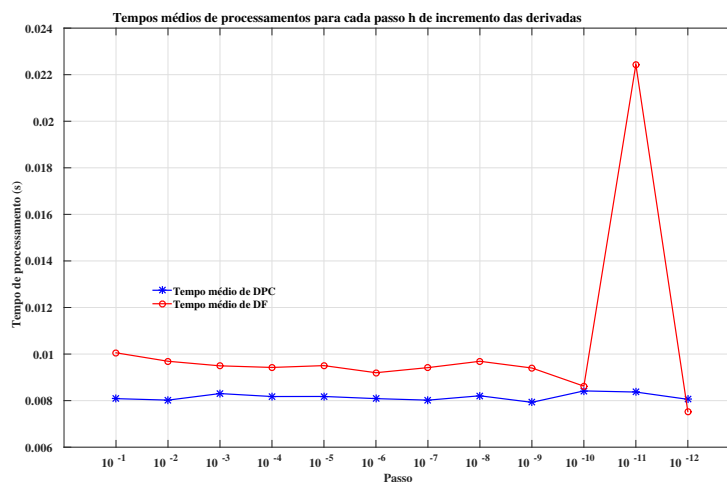
O artigo de [1] diz que por conta da aritmética de operações no conjunto dos números complexos envolver várias operações no conjunto dos números reais, o método DPC tem em média uma demanda de tempo de cerca de uma vez e meia do tempo de processamento efetuado para operações com derivadas por DF. O autor relata que outro trabalho apresenta custos ainda

Figura 3.17: Valores médios do número de iterações na resolução do sistema 3.16 pelo método de Newton, obtidos com aproximação da jacobiana por DF centrada e DPC sob variação do incremento para as derivadas.



superiores, com duas a três vezes do tempo com DF; assim como um outro artigo mencionado, que diz que DPC é em torno de 12% mais lento que DF. A figura (3.18) apresenta as médias de tempo de processamento requerido em 100 repetições de execução dos programas com DF e DPC. Nela vemos que, diferentemente do que afirma [1], DPC teve demanda de tempo inferior ao se usar DF. Um custo menor em torno de 10% foi alcançado com DPC, mas, nossa experimentação na resolução do sistema 3.16 requer uma relativa pequena demanda computacional, visto que as diferenças de tempo de processamento utilizando as duas técnicas são muito pequenas.

Figura 3.18: Valores médios de tempo de processamento na resolução do sistema 3.16 pelo método de Newton, obtidos com aproximação da jacobiana por DF centrada e DPC sob variação do incremento para as derivadas.



Capítulo 4

Considerações Finais

Neste trabalho, nos propomos a mostrar um não tão recente método de diferenciação numérica, mas que é pouco conhecido no meio acadêmico a nível de graduação. Fizemos algumas comparações com o método de Diferenças Finitas já bastante utilizado. Tendo em vista a grande importância do estudo e aplicações da derivada, seja no ambiente acadêmico fazendo menção do estudo da derivada e de métodos computacionais de diferenciação, até a sua aplicação em processos de otimização; sendo parte essencial na construção de matrizes jacobianas e hessianas, muito utilizadas em processos de inversão na resolução de sistemas lineares, não lineares e de mínimos quadrados. Por isso em tais processos o uso de derivadas acuradas é fundamental.

Tivemos a oportunidade de estudar e implementar a Diferenciação Com Passo Complexo e conseguimos notar a sua poderosa forma de entregar derivadas computacionais de forma mais acurada, com tempo computacional aproximadamente igual ao de Diferenças Finitas e de forma bastante simples para implementação.

Nos teste efetuados, tivemos a nítida percepção de que em relação à acurácia, a DPC foi extraordinária. Vimos que entre as aproximações de DF, a centrada retorna maior acurácia para derivadas de primeira ordem que DF adiante, mas mesmo assim, a DPC a supera nesse quesito. Quanto à segunda derivada com DPC, mostramos que a aproximação com uso do incremento real dado em função da precisão de máquina é muito superior que a simples aproximação com incremento apenas na parte imaginária da função.

Dentro dessas descobertas caímos nos questionamentos sobre dois artigos que tratam de DPC. Um deles apresenta resultados de aproximação de segunda derivada com adoção do incremento da parte real sem ter o cuidado de apresentar, ou pelo menos comentar, que valores seriam ótimos para essa aproximação de segunda ordem. Por conta disso mostra resultados muito aquém daqueles dos quais conseguimos com uso da função de precisão de máquina adequada. Sobre outro artigo pudemos avaliar que mesmo aproximações com erros de truncamento inferior perdem em acurácia em relação as que aqui adotamos para DPC.

Dentro de todas as investigações mostramos que o método de DF apresenta acurácia muito

diminuída com pequenos passos, decorrente de estar sujeita ao erro de cancelamento subtrativo. Além disso, o melhor passo em uma dada aplicação não necessariamente é o mesmo para outra; deixando assim DF sujeita a avaliação com testes do tipo tentativa-e-erro na ocasião de desconhecimento de expressões analíticas.

Quanto à DPC, para os pequeníssimos valores dos passos, verificamos que o cancelamento subtrativo não parece comprometer nem mesmo a aproximação de segunda ordem. Passos cada vez menores não fazem com que a aproximação se deteriore, ou mesmo melhore. E, adoção de passos de ordem muito pequena podem evitar a não validade da aproximação, que é aceitável somente se em torno do ponto de interesse a função for é analítica.

Nas aplicações com métodos iterativos, cremos que os exemplos adotados não foram do porte necessário para requerer grande custo computacional das técnicas testadas. No método de Halley, praticamente o tempo de processamento de ambas foram os mesmos. No caso da resolução do sistema não-linear pelo método de Newton, tivemos uma maior exigência computacional, mas não tão grande. Nas obtenções de soluções válidas por uso de DF, os tempos de processamento requerido por DPC foi em torno de 10% menor. Isso contrasta com o que foi declarado por alguns artigos. E que, de certo modo, justificam o tempo demandado por DPC ser maior que DF, porque a primeira utiliza de operações matemáticas tanto no domínio real quanto complexo, já a segundo, só operações no conjunto dos números reais.

Na aplicação aplicação do método de Newton, usamos derivadas de primeira ordem na obtenção da matriz jacobiana, e, mais uma vez detectamos o perigo em que valor do passo usar em DF. Com a diminuição de apenas uma ordem na grandeza do incremento para a derivada, atestamos que a jacobiana passou a ser singular, inviabilizando a determinação de uma solução do sistema.

Apesar de termos apresentado as expressões de derivadas cruzadas que podem ser utilizadas para obtenção da matriz hessiana via DPC, não realizamos nenhuma aplicação a fim de retornar informações de acurácia e desempenho computacional quando comparadas com DF. Uma forma natural para o aparecimento dessa matriz é da aproximação até a segunda ordem do método de Newton sobre solução de mínimos quadrados não lineares. Apesar de termos começado sua implementação computacional com DPC, não pudemos apresentar algumas investigações, e por isso elencamos como de trabalho futuro em apresentação de eventos acadêmicos.

Esse trabalho não somente se apresenta como um requisito para obtenção de um título acadêmico. Pensamos que decorrente da simplicidade da formulação e facilidade de implementação computacional, este assunto deve ser apresentado nas cadeiras dos cursos de matemática numérica. Certamente aqui não abrangemos todas as possibilidades de aplicação ou de avaliação de mérito computacional, mas dispomos diversas análises que ajudam o entendimento e valorização da técnica de diferenciação com passo complexo.

Referências Bibliográficas

- [1] Abokhodair, Abdulwahab A.: *Complex differentiation tools for geophysical inversion*. Geophysics, 74(2):H1–H11, 2009.
- [2] Abreu, Rafael, Daniel Stich e José Morales: *On the generalization of the Complex Step Method*. Journal of Computational and Applied Mathematics, 241:84 – 102, 2013, ISSN 0377-0427.
- [3] Ackermann, Edward M Mikhail; Friedrich E: *Observations and Least Squares*.
- [4] Alexe, M. *et al.*: *Automatic differentiation of codes in nuclear engineering applications*. dezembro 2009.
- [5] Arenales, Selma e Artur Darezzo: *Cálculo Numérico: Aprendizagem com Apoio de Software*. Thomson Learning, 2008, ISBN 9788522106028.
- [6] Campbell, Alan Robert: *Numerical Analysis of Complex-Step Differentiation in Spacecraft Trajectory Optimization Problems*. Tese de Mestrado, The University of Texas at Austin, 2011.
- [7] Cunha, M. Cristina C.: *Métodos Numéricos*. UNICAMP, 2000, ISBN 9788526808775.
- [8] Forth, Shaun *et al.*: *Recent Advances in Algorithmic Differentiation*. Springer, 2012, ISBN 978-3-642-30023-3.
- [9] Franco, Neide Bertoldi: *Cálculo Numérico*. Pearson Prentice Hall, São Paulo, São Paulo, 2006, ISBN 978-85-7605-087-2.
- [10] Franke, R. e Naval Postgraduate School Monterey CA: *A Critical Comparison of Some Methods for Interpolation of Scattered Data*. Final report. Defense Technical Information Center, 1979.
- [11] Gander, W., M.J. Gander e F. Kwok: *Scientific Computing - An Introduction using Maple and MATLAB*. Texts in Computational Science and Engineering. Springer International Publishing, 2014, ISBN 9783319043258.

- [12] Junior, Esly Ferreira Costa: *Resolução Automática de Equações Algébrico-diferenciais de Índice Superior*. Unpublished, 2003.
- [13] Kim, Jongrae, Declan G. Bates e Ian Postlethwaite: *Nonlinear robust performance analysis using complex-step gradient approximation*. *Automatica*, 42(1):177–182, 2006.
- [14] Lai, K. L. e J.L. Crassidis: *Extensions of the first and second complex-step derivative approximations*. *Journal of Computational and Applied Mathematics*, 219(1):276 – 293, 2008.
- [15] Lantoine, Gregory e Ryan P. Russel: *Using Multicomplex Variables for Automatic Computation of High-Order Derivatives*. *ACM Transactions on Mathematical Software*, 38(3):21, 2012.
- [16] Lyness, J. N. e C. B. Moler: *Numerical differentiation of analytic functions*. *SIAM Journal of Numerical Analysis*, 38(4):202–210, 1967.
- [17] Martins, Joaquim R. R. A., Peter Sturdza e Juan J. Alonso: *The Complex-step Derivative Approximation*. *ACM Trans. Math. Softw.*, 29(3):245–262, setembro 2003, ISSN 0098-3500.
- [18] MONICO, JOÃO FRANCISCO GALERA. *et al.: ACURÁCIA E PRECISÃO: REVENDO OS CONCEITOS DE FORMA ACURADA*. 15:469– 483, 2009, ISSN 0377-0427.
- [19] Neto, Alcides Lins: *Funções de uma variável complexa*. Projeto Euclides. IMPA, 2008, ISBN 9788524400872.
- [20] Press, W.H. *et al.: Numerical Recipes in FORTRAN 77: The Art of Scientific Computing*, volume 1. Cambridge University Press, 1986.
- [21] Squire, William e George Trapp: *Using Complex Variables to Estimate Derivatives of Real Functions*. *SIAM Rev.*, 40(1):110–112, março 1998, ISSN 0036-1445.