ALGORÍTMOS DO TIPO ROMBERG
PARA QUADRATURA NUMÉRICA

LÚCIA HELENA BARROS MANARA

ORIENTADOR: PROFA. DRA. CÉLIA MARIA FINAZZI DE ANDRADE

Dissertação apresentada ao Instituto
de Ciências Matemáticas de São Car-
los, da Universidade de São Paulo,
para obtenção do título de "Mestre
em Ciências de Computação".

SÃO CARLOS - SP
1986

Aos meus pais
dedico este
trabalho.

AGRADECIMENTOS

- À Profa. Dra. Célia Maria Finazzi de Andrade, pela orientação na elaboração desta dissertação, proporcionando o meu desenvolvimento científico.
- Aos amigos, professores e funcionários do ICMSC, pela atenção a mim dispensada.
- Aos colegas do Departamento de Ciências da Universidade Estadual Paulista "Julio de Mesquita Filho", Campus de Ilha Solteira, pela colaboração e incentivo; em especial à amiga Profa. Neusa Augusto da Silva.
- À Profa. Maria José Romanatto, pela colaboração e atenção no desenvolvimento deste trabalho.
- Aos funcionários, Carmem Lúcia Pagadigorria e Valdir de Camargo Melchior, do Laboratório de Computação; José Luis de Souza Cabral e José Eduardo Amorin Pires, do Laboratório de Sistemas Digitais, por proporcionarem condições de execução deste trabalho.
- Às funcionárias da Seção de Pós-Graduação do ICMSC, Elizabeth Luisa Moretti e Silva, Laura Aparecida D. Ruy Turi e Sonia Regina Derigi Borges, pela atenção.
- Às bibliotecárias do ICMSC, pela presteza no atendimento.
- À Sra. Wilma Provinciali Vall, pela disposição e eficiência no trabalho de datilografia.
- Aos meus pais, pelo incentivo, apoio e compreensão em todas as etapas de meu trabalho.
- Ao Joel, pela presença e constante apoio.
- Ao CNPq e CAPES, pelo auxílio financeiro.

R E S U M O

Este trabalho tem por objetivo apresentar e analisar diferentes métodos de integração automática, baseados no método de Romberg. Os métodos podem ser adaptativos ou não adaptativos.

Experimentos numéricos são realizados com o intuito de comparar os diversos métodos quanto ao seu custo, precisão e confiança. Dois métodos adaptativos extras, baseados na regra de Simpson, são incluídos para uma comparação mais realista.

A B S T R A C T

This work introduces and analyses different methods, based on Romberg's quadrature, for automatic integration. The methods can be adaptives or not adaptives.

Numerical experiments are performed to compare costs, accuracy and confidence of the methods. Two extra adaptive Simpson's methods are included in order to have a more realistic comparison.

Í N D I C E

INTRODUÇÃO	1
I.1 - CONSIDERAÇÕES GERAIS	1
I.2 - APRESENTAÇÃO DO TRABALHO	2

C A P Í T U L O 1

RESENHA BIBLIOGRÁFICA	5
1.1 - REFERÊNCIAS SOBRE INTEGRAÇÃO DE ROMBERG	5
1.2 - REFERÊNCIAS SOBRE MODIFICAÇÕES DO MÉTODO DE ROMBERG	6
1.3 - REFERÊNCIAS SOBRE O MÉTODO DE ROMBERG USANDO EXTRAPOLAÇÃO RACIONAL	6
1.4 - REFERÊNCIAS SOBRE ALGUNS MÉTODOS DE INTEGRAÇÃO AUTOMÁTICA	7
1.5 - OUTRAS REFERÊNCIAS	8
1.5.1 - Sobre erro na quadratura de Romberg	8
1.5.2 - Sobre modificações e adaptações do método de Romberg	9
1.5.3 - Sobre o método de Romberg em integração multidimensional	10
1.5.4 - Sobre integração automática	11

C A P Í T U L O 2

PRÉ-REQUISITOS	13
----------------------	----

2.1 - NÚMEROS E POLINÔMIOS DE BERNOULLI	13
2.1.1 - Números de Bernoulli	13
2.1.2 - Polinômios de Bernoulli	16
2.1.3 - Funções periódicas relacionadas aos polinômios de Bernoulli	22
2.2 - FÓRMULA DO ERRO DE EULER-MacLAURIN PARA A REGRA DOS TRAPÉZIOS	23

C A P Í T U L O 3

QUADRATURA DE ROMBERG	25
3.1 - A IDÉIA BÁSICA	26
3.2 - A T-TABELA	30
3.3 - PROPRIEDADES DO MÉTODO DE ROMBERG	32
3.3.1 - Polinômios geradores da T-tabela	32
3.3.2 - Os coeficientes peso para o método de Romberg	37
3.4 - ESTUDO DO ERRO NO MÉTODO DE ROMBERG	40
3.4.1 - O termo exato do erro para $T_m^{(k)}$	40
3.4.2 - Outra expressão explícita para o termo do erro	44
3.4.3 - Cálculo numérico das funções $b_{2m+2}(x)$...	49
3.4.4 - Estimativa para o termo do erro $R_m^{(k)}$	52
3.5 - OUTRAS SUBDIVISÕES ALÉM DA BISSECÇÃO	53
3.6 - MÉTODO DE ROMBERG: EXTENSÃO PARA N-VARIÁVEIS ...	58
3.7 - CONCLUSÕES	60

C A P Í T U L O 4

MODIFICAÇÕES DO MÉTODO DE ROMBERG	62
4.1 - INTEGRAÇÃO DE ROMBERG COM O USO DE VALORES RETANGULARES	63

4.1.1 - Algoritmo de Krasun e Prager	63
4.1.2 - Separação da seqüência de aproximações calculadas pelo algoritmo de Romberg em duas seqüências	68
4.1.3 - Forma alternativa do método de Romberg baseada na transformação cosseno de Euler-MacLaurin e 2ª fórmula de Euler ..	73
4.1.4 - Derivação de expressões explícitas para o termo do erro no algoritmo de Romberg modificado	82
4.1.5 - Propriedades para os termos dos erros $R_m^{(k)}$ e $\bar{R}_m^{(k)}$	86
4.2 - INTEGRAÇÃO DE GAUSS-ROMBERG	90
4.2.1 - Algoritmo de Gauss-Romberg	90
4.2.2 - Obtenção da expressão para o termo do erro	99
4.3 - INTEGRAÇÃO DE ROMBERG COM EXTRAPOLAÇÃO RACIONAL ..	105
4.3.1 - Obtenção do algoritmo	105
4.3.2 - Limites assintóticos para o erro	109
4.4 - CONCLUSÕES	112

C A P Í T U L O 5

O MÉTODO DE ROMBERG E OUTROS NA INTEGRAÇÃO AUTO- MÁTICA	115
5.1 - INTEGRAÇÃO AUTOMÁTICA: CARACTERÍSTICAS GERAIS E FINALIDADES	116
5.2 - INTEGRAÇÃO DE ROMBERG: UM MÉTODO DE INTEGRAÇÃO AUTOMÁTICA	120
5.2.1 - Um método de Romberg adaptativo	120
5.2.2 - Um método de Romberg adaptativo e caute- loso: idéia geral	121
5.2.3 - Aplicação do método de Romberg adaptati- vo e cauteloso	127

5.2.4 - Testes e exemplos	134
5.3 - UMA INTEGRAÇÃO ITERATIVA ADAPTATIVA BASEADA NA REGRA DE SIMPSON	140
5.3.1 - Método de quadratura de Simpson adapta- tivo original	141
5.3.2 - Análise elementar do erro para rotinas de quadratura do tipo ASQ	147
5.4 - CONCLUSÕES	161

C A P Í T U L O 6

COMPARAÇÃO DOS VÁRIOS MÉTODOS DE QUADRATURA AU- TOMÁTICA ADAPTATIVA OU NÃO ADAPTATIVA	163
6.1 - ROTINAS DE QUADRATURA TESTADAS	163
6.2 - FUNÇÕES INTEGRANDO, TESTES E COMPARAÇÕES	164
6.3 - CONCLUSÕES	185

C A P Í T U L O 7

CONCLUSÕES FINAIS	186
APÊNDICE	188
A.1 - Função QUAD	189
A.2 - Função ROMINT	192
A.3 - Função INTEGRATOR	195
A.4 - Função GAUSS	199
A.5 - Função IRATEX	204
A.6 - Função PRAGER	209
A.7 - Função CADRE	211
A.8 - Função SIMPSON	223
A.9 - Função SQUANK	226
BIBLIOGRAFIA	233

I N T R O D U Ç Ã O

I.1 - CONSIDERAÇÕES GERAIS

Quadratura numérica consiste na obtenção do valor numérico de uma integral

$$I = \int_a^b f(x) dx \quad (I.1)$$

e é geralmente efetuada aproximando o integrando por partes, por polinômios, integrando-os exatamente. As regras de quadratura para aproximar (I.1) são, em geral, somas ponderadas dos valores do integrando em pontos do intervalo $[a,b]$ ou fora dele. Uma classe muito conhecida é a de Newton-Cotes (Krylov, 1962, capítulo 6), que usa pontos equidistantes.

Os aproximantes para (I.1), obtidos pelas fórmulas de Newton-Cotes de ordem crescente, não necessariamente convergem, mesmo que $f(x)$ seja contínua em todo o intervalo $[a,b]$. Este fato lança algumas dúvidas sérias sobre a filosofia de operar sobre funções arbitrárias pelo processo interpolatório e, certamente, impede o uso de fórmulas de Newton-Cotes de ordem mais elevada para quadratura numérica.

Existe uma outra classe de fórmulas de quadratura para integrais da forma mais geral $J = \int_a^b f(x)w(x)dx$, onde $w(x)$ é uma função peso. Estes métodos substituem $w(x)$ por uma distribuição de pesos, concentrada num número finito de pontos e, portanto, produzem a fórmula do seguinte tipo:

$$J \approx Q = \sum_{k=1}^n c_k f(x_k).$$

Os x_k e c_k são, respectivamente, as abscissas e os coeficientes peso para o particular intervalo e função peso $w(x)$ e são determinados de tal forma que a fórmula é exata para polinômios de mais alto grau possível.

Os métodos baseados neste princípio, tais como as fórmulas de Gauss, são numericamente estáveis, mas os c_k e x_k são, exceto para algumas funções peso especiais, muito difíceis de calcular. Isto não é uma desvantagem séria, mas desafia o cálculo recursivo de aproximantes de ordem crescente, o que seria de enorme eficiência para cálculo automático.

As desvantagens dos dois princípios mencionados são ambas evitadas por um método dado por Romberg em 1955, que inicialmente conquistou comparativamente pouca atenção. Entretanto, pelos anos 1960, o interesse no método brotou rapidamente, estimulado, pelo menos em parte, pela pesquisa intensiva para procedimentos de integração automática eficiente e pelo rápido desenvolvimento no campo de computação científica. Este método, baseado apenas na regra trapezoidal, é facilmente adaptado para cálculo automático e tem-se revelado um método iterativo de bastante sucesso

para integração numérica, tendo inclusive sido aplicado a problemas com singularidades.

I,2 - APRESENTAÇÃO DO TRABALHO

Este trabalho representa, principalmente, um estudo sobre os vários aspectos do método de Romberg, suas modificações e extensões introduzidas por outros autores, tendo como objetivo uma análise comparativa destes métodos entre si e também com outro tipo de método em termos do número de cálculos funcionais requeridos para se atingir uma determinada precisão e grau de confiabilidade.

No capítulo 1, apresentamos as principais referências nas quais se baseou o desenvolvimento deste trabalho, e damos outras referências que podem ser úteis ao conhecimento de um pesquisador.

No capítulo 2, damos alguns resultados especiais necessários para a construção da teoria de integração aproximada, apresentados somente para melhor entendimento dos outros capítulos.

No capítulo 3, descrevemos o método de Romberg, analisamos e obtemos uma estimativa do erro, usando polinômios de Bernoulli e suas funções periódicas relacionadas; demonstramos a convergência do método e apresentamos algumas de suas aplicações.

No capítulo 4, apresentamos algumas modificações do método de Romberg com a finalidade de reduzir erro de arredondamento, obtemos resultados teóricos sobre os li

mites superior e inferior para o valor verdadeiro da integral e derivamos expressões explícitas para o termo do erro. Introduzimos também o algoritmo de Gauss-Romberg. Apresentamos mais uma alternativa do método de Romberg usando extrapolação racional ao invés de polinomial.

No capítulo 5, fazemos uma avaliação dos princípios de integração automática e descrevemos os algoritmos automáticos de Romberg e Simpson.

No capítulo 6, através de resultados numéricos, apresentamos uma análise comparativa do método de Romberg, suas modificações e extensões introduzidas por outros autores.

No capítulo 7, expomos conclusões finais obtidas ao longo deste trabalho.

No Apêndice, damos uma breve descrição de cada rotina utilizada, com a respectiva listagem.

CAPÍTULO 1

RESENHA BIBLIOGRÁFICA

Este capítulo tem por objetivo mencionar e comentar trabalhos relevantes que tratam de integração de Romberg, suas extensões e modificações.

1.1 - REFERÊNCIAS SOBRE INTEGRAÇÃO DE ROMBERG

Bauer (1961) parece ter sido o primeiro a produzir uma análise rigorosa do erro do método de Romberg. Entretanto, uma análise mais completa foi apresentada por Bauer, Rutishauser e Stiefel (1963) num artigo básico sobre o assunto. Neste artigo são dadas as principais características do método e algumas de suas aplicações.

Håvie (1969), mostra que o termo do erro para o algoritmo de Romberg, pode ser expresso em termos dos polinômios de Bernoulli, suas funções periódicas relacionadas e dá uma estimativa para ele, aqui obtida de maneira mais simples que no artigo anterior [B.R.S. (1963)].

1.2 - REFERÊNCIAS SOBRE MODIFICAÇÕES DO MÉTODO DE ROMBERG

Krasun e Prager (1965), observam que erros de arredondamento no método de Romberg, podem ser reduzidos formando uma "tabela esqueleto" onde as aproximações iniciais são obtidas diretamente de valores retangulares.

Håvie (1966, 1967) mostra resultados teóricos sobre os limites superior e inferior para o valor verdadeiro da integral, obtidos com muito pouco cálculo adicional. Observa também que a transformação cosseno, aplicada por Smith (1965) para a regra de Simpson modificada, pode ser usada com o método de Romberg no cálculo de $\int_{-1}^1 f(x) dx$.

Mais tarde Håvie (1969), expressa o erro em termos de polinômios de Bernoulli e suas funções periódicas relacionadas, deduzindo propriedades para este erro.

Lyness (1972), observa que o método de Romberg pode ser aplicado para aproximações iniciais obtidas de regras de Gauss (Algoritmo de Gauss-Romberg).

Håvie (1972) obtém a expressão geral para o termo do erro no caso de uma subdivisão arbitrária do intervalo de integração.

1.3 - REFERÊNCIAS SOBRE O MÉTODO DE ROMBERG USANDO EXTRAPOLAÇÃO RACIONAL

Bulirsch e Stoer (1964) sugerem que, sob determinadas condições das aproximações iniciais $\pi(h)$ no lu-

ção racional. Bulirsch e Stoer (1967) a aplicam para calcular somas trapezoidais para vários valores de h , experimentam uma função racional de h^2 para estes valores e então extrapolam para $h=0$. Obtêm também limites assintóticos para o erro.

1.4 - REFERÊNCIAS SOBRE ALGUNS MÉTODOS DE INTEGRAÇÃO AUTOMÁTICA

Prager (1965), torna o método de Romberg mais flexível através de uma modificação adaptativa do método.

Boor (1971) implementa um método de Romberg adaptativo e cauteloso, o qual toma decisões apropriadas de acordo com o comportamento do integrando no intervalo de integração.

Lyness (1969) descreve a rotina de quadratura de Simpson adaptativa, dada por McKeeman (1962, 1963) e investiga, sob um plano elementar, as propriedades do erro da aproximação. Indica também algumas mudanças que podem melhorar a eficiência desta rotina.

Davis e Rabinowitz (1975), capítulo 6, fazem uma avaliação explícita dos princípios de integração automática e dão uma discussão completa de uma grande variedade de algoritmos de integração automática, dentre os quais citamos:

- Um método iterativo não adaptativo baseado em fórmulas de Kronrod [Cranley e Patterson (1971)], [Patterson (1973)].

- Um método adaptativo não iterativo para integração automática baseado na regra do ponto médio [Casaletto, Pickett e Rice (1969)];
- Um método de uma integração iterativa baseado na regra de Simpson, [Villars (1956)];
- Integração de Newton-Cotes adaptativa, [Hillstrom (1970)];
- Método adaptativo de O'Hara e Smith, [O'Hara e Smith (1969)];
- O método de Cranley-Patterson, [Cranley e Patterson (1971)];
- Integração automática usando polinômios de Tschebyscheff, [Clenshaw e Curtis (1960)]; [O'Hara e Smith, (1968, 1969)], [Hávie, (1969, 1971)];
- Integração automática de funções oscilatórias, [Chasse e Fosdick (1969)], [Linz (1972)];
- Integração automática em várias variáveis [Genz, (1972)], [Laurent (1964)], [Lyness e MacHugh (1963-1964)], [Stroud (1971)].

1.5 - OUTRAS REFERÊNCIAS

1.5.1 - SOBRE ERRO NA QUADRATURA DE ROMBERG

Stroud (1965) tabula constantes, em dois tipos de estimativas para o erro na quadratura de Romberg, produzindo uma base para comparação direta das fórmulas de Rom

berg com outras fórmulas de quadratura, por exemplo Gauss-Legendre. O primeiro tipo de estimativa usa limite sobre a derivada do integrando e o segundo são estimativas para funções analíticas numa certa região do plano complexo.

Ström (1967) dá dois procedimentos para a obtenção dos limites do erro exato na quadratura de Romberg. O procedimento mais simples usa somente quantidades já calculadas no método de Romberg e o mais geral é baseado no conceito de majorante.

1.5.2 - SOBRE MODIFICAÇÕES E ADAPTAÇÕES DO MÉTODO DE ROMBERG

Meir e Sharma (1965) sugerem o uso de fórmulas de Newton-Cotes de ordem mais elevada, para obter as aproximações iniciais do método de Romberg.

Hunter (1967) sugere que singularidades em $\int_a^b w(x)g(x)dx$ podem ser evitadas tomando $x = x(t)$, onde x é uma função monotônica de t tal que $x(A) = a$, $x(B) = b$ e $w(x)dx = W(t)dt$, com $W(t)$ sem singularidade em $[A,B]$, aplicando uma técnica baseada no método de Romberg.

Fox (1967) mostra como o método de Romberg pode ser modificado para tratar com integrandos infinitos e com primeira derivada infinita nos extremos do intervalo de integração, como por exemplo, $x^{1/2}g(x)$, $x \ln x g(x)$, $x^{1/2} \ln x g(x)$, $x^{1/2}(1-x)^{1/2}$ e $x^{-1/2}$.

Shanks (1972) constrói a tabela de Romberg para certas classes de integrandos singulares, usando uma

técnica mais simples e mais econômica que a empregada por Fox (1967), produzindo o mesmo resultado.

Ninham e Lyness (1967, 1969) investigam integrais com singularidades de qualquer tipo, sobre ou próximo do intervalo de integração.

1.5.3 - SOBRE O MÉTODO DE ROMBERG EM INTEGRAÇÃO MULTIDIMENSIONAL

Lyness e MacHugh (1963) sugerem o uso de extrapolação com uma regra simétrica de ordem baixa (por exemplo, a regra produto do ponto médio) em integração multidimensional (restrita a domínios hipercúbicos). O erro é sucessivamente reduzido usando um método baseado na idéia de extrapolação de Richardson. Em (1970) eles introduzem um "formalismo conciso" para obter o termo do resto para a fórmula da soma de Euler-MacLaurin N-dimensional.

Monegato e Lyness (1979) investigam a possibilidade de usar integração de Romberg bi-dimensional para aproximar integrais sobre o domínio $0 \leq x, y \leq 1$ de funções integrando da forma $g(x,y)/(x-y)$ onde $g(x,y)$ é, por exemplo, analítica em x e y . Derivam a expressão assintótica sobre a qual métodos de extrapolação podem ser baseados quando $g(x,y)$ tem um ponto de singularidade em direção à origem.

Outros usos de extrapolação em integração multidimensional são dados por Hodgson (1969) e Baker e Hodgson (1970).

1.5.4 - SOBRE INTEGRAÇÃO AUTOMÁTICA

Lyness (1969a) discute as consequências da omissão da possibilidade de ocorrência de erro de arredondamento ou outros erros estatísticos nos valores da função na construção de rotinas automáticas.

Lyness e Kaganove (1976) descrevem propriedades gerais de rotinas de quadratura automática e discutem em detalhe o conflito entre confiança e economia que surge quando se constrói uma rotina automática. Em (1977) eles dão uma técnica que pode ser usada na comparação de rotinas.

Laurie (1983) dá uma estimativa exata do erro em quadratura adaptativa. Ele usa duas regras de quadratura A, B que produzem aproximações A_1, B_1 para $I = \int_{x_0}^{x_1} f(x) dx$. Enquanto é comum estimar os erros em A_1, B_1 por $|A_1 - B_1|$ ele observa a deficiência deste processo e introduz uma nova estimativa para o erro, estabelecendo condições sob as quais ela é mais confiável, podendo ser estendida para integração múltipla sobre regiões apropriadas.

Lyness (1983) propõe ao usuário que não é um especialista em Matemática, condições de escolha entre uma rotina de quadratura que usa uma regra padrão, baseada diretamente sobre fórmula tradicional e uma rotina de quadratura automática para calcular integrais numericamente. Uma considerável parte do artigo é dedicada a uma descrição detalhada de como uma rotina de quadratura automática traba-

da para ter cada uma destas propriedades. É mostrado que enquanto em muitos problemas o uso de uma rotina automática pode salvar significativo tempo humano, em muitos outros, seu uso pode ser um desastre. Não há informação neste artigo sobre rotinas particulares; são dadas propriedades genéricas destas duas categorias fundamentalmente diferentes de software numérico.

CAPÍTULO 2

PRÉ-REQUISITOS

Apresentamos certos resultados selecionados dos tópicos matemáticos especiais seguintes: números de Bernoulli, polinômios de Bernoulli e suas funções periódicas relacionadas e a fórmula do erro de Euler McLaurin para a regra dos trapézios.

Estes tópicos são necessários para construir a teoria de integração aproximada e são apresentados somente para melhor entendimento dos outros capítulos. Os resultados desenvolvidos aqui podem ser encontrados em literatura especial, mas achamos útil apresentá-los para livrar o leitor da inconveniência de olhar referência de literatura.

Consideremos a função

$$g(t) = \frac{t}{e^t - 1} \quad (2.1.1.1)$$

onde t é um parâmetro complexo. Para k inteiro, os pontos

$t = 2k\pi i$ são os zeros do denominador. Todos são zeros simples, pois a derivada do denominador é e^t , diferente de zero para todo t finito e o ponto $t = 0$ não é um ponto singular de $g(t)$, pois $\lim_{t \rightarrow 0} \frac{t}{e^t - 1} = 1$.

A função $g(t)$ pode ser expandida numa série de fatores em t :

$$\frac{t}{e^t - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} t^n, \quad |t| < 2\pi \quad (2.1.1.2)$$

Os números B_n , definidos por esta equação, são chamados números de Bernoulli.

Se multiplicarmos ambos os lados de (2.1.1.2) por $e^t - 1 = \sum_{v=1}^{\infty} \frac{t^v}{v!}$, então obtemos a equação

$$\left(\frac{t}{1!} + \frac{t^2}{2!} + \frac{t^3}{3!} + \dots \right) \sum_{n=0}^{\infty} \frac{B_n}{n!} t^n = t,$$

válida para todo t no círculo $|t| < 2\pi$. Desta igualdade se

Esta equação nos permite calcular sequencialmente todos os números de Bernoulli. Podemos obter outras formas mais convenientes para o nosso propósito: multiplicando por $n!$ e adicionando B_n em ambos os lados, obtemos

$$\sum_{k=0}^n \frac{n!}{k!(n-k)!} B_k = B_n$$

Podemos facilmente verificar que todos os números de Bernoulli com índice ímpar maior que a unidade, são iguais a zero:

$$B_{2k+1} = 0, \quad k > 0 \quad (2.1.1.3)$$

Os valores dos números de Bernoulli, diferentes de zero, para $n \leq 10$ são:

$$B_0 = 1 \quad B_2 = \frac{1}{6} \quad B_6 = \frac{1}{42} \quad B_{10} = \frac{5}{66}$$

$$B_1 = -\frac{1}{2} \quad B_4 = -\frac{1}{30} \quad B_8 = -\frac{1}{30}$$

Os números de Bernoulli com índices pares são relacionados com os números naturais pela seguinte identidade:

$$B_{2k} = \frac{(-1)^{k-1} (2k)!}{2^{2k-1} \pi^{2k}} \cdot (1+2^{-2k}+3^{-2k}+4^{-2k}+\dots), \quad (2.1.1.4)$$

donde verificamos que para valores inteiros grandes de k , os números de Bernoulli B_{2k} crescem muito rapidamente, valendo aproximadamente

2.1.2 - POLINÔMIOS DE BERNOULLI

Definimos polinômios de Bernoulli por meio da função geradora:

$$g(x,t) = e^{xt} \frac{t}{e^t - 1} \quad (2.1.2.1)$$

que difere de (2.1.1.1) pelo fator e^{xt} , que não se anula; então $g(x,t)$ tem os mesmos pontos singulares que $g(t)$. Podemos expandi-la numa série de fatores em t :

$$g(x,t) = e^{xt} \frac{t}{e^t - 1} = \sum_{n=0}^{\infty} \frac{B_n(x)}{n!} t^n. \quad (2.1.2.2)$$

Veremos a seguir que as funções $B_n(x)$ são polinômios de grau n , chamados de polinômios de Bernoulli.

Se, em $g(x,t)$ o fator e^{xt} é substituído pela série

$$\sum_{v=0}^{\infty} \frac{x^v t^v}{v!} e^{\frac{t}{e^t - 1}}$$

é substituído pela expansão (2.1.1.2), obtemos a identidade

$$\sum_{v=0}^{\infty} \frac{x^v t^v}{v!} \sum_{k=0}^{\infty} \frac{B_k}{k!} t^k = \sum_{n=0}^{\infty} \frac{B_n(x)}{n!} t^n, \quad |t| < 2\pi.$$

Comparando os coeficientes de t^n , temos

$$\frac{B_n(x)}{n!} = \frac{x^n B_0}{n!} + \frac{x^{n-1} B_1}{(n-1)!1!} + \dots + \frac{B_n}{n!}$$

ou seja, $B_n(x)$ é o polinômio de grau n :

$$B_n(x) = \sum_{k=0}^n \frac{n!}{k!(n-k)!} B_{n-k} x^k \quad (2.1.2.3)$$

Precisamos estar familiarizados com certas propriedades dos polinômios de Bernoulli, dadas a seguir:

1. O valor de um polinômio de Bernoulli:

Para $x=0$, o valor de um polinômio de Bernoulli é o correspondente número de Bernoulli.

$$B_n(0) = B_n \quad (2.1.2.4)$$

que obtemos de (2.1.2.3).

2. Diferenciação e Integração de $B_n(x)$.

$$\frac{B'_n(x)}{n!} = \frac{B_{n-1}(x)}{(n-1)!}$$

ou

(2.1.2.5)

$$B'_n(x) = n B_{n-1}(x)$$

obtida diferenciando (2.1.2.2) com relação a x .

De (2.1.2.5) e (2.1.2.4) segue

$$B_n(x) = B_n + n \int_0^x B_{n-1}(t) dt \quad (2.1.2.6)$$

3. Multiplicação do argumento por uma constante.

$$B_n(mx) = m^{n-1} \sum_{s=0}^{m-1} B_n\left(x + \frac{s}{m}\right), \quad (2.1.2.7)$$

que para $m=2$ fornece:

Fazendo $x = 0$ vem:

$$B_n\left(\frac{1}{2}\right) = - \left[1 - \left(\frac{1}{2}\right)^{n-1} \right] B_n \quad (2.1.2.8)$$

4. Representações para os polinômios $B_n(x)$.

O estudo de $B_n(x)$ fica simplificado se usarmos a nova variável $z = x(1-x)$, valendo as afirmações que seguem.

Cada polinômio $B_n(x)$ de ordem par, $n = 2k$, pode ser expandido em termos de z :

$$(-1)^k \left[B_{2k}(x) - B_{2k} \right] = \sum_{v=0}^{k-2} F_{k,v} z^{k-v}, \quad (2.1.2.9)$$

onde $F_{k,0} = 1$ e $F_{k,v} > 0$; $v=1, 2, \dots, k-2$. Para polinômios de ordem ímpar, $n=2k-1$, teremos:

$$(-1)^k B_{2k-1}(x) = (1-2x) \sum_{v=0}^{k-2} H_{k,v} z^{k-v-1} \quad (2.1.2.10)$$

onde todos os coeficientes $H_{k,v}$; $v=0, 1, \dots, k-2$ são positivos.

Podemos obter sequencialmente $F_{k,v}$ pela relação de recorrência

$$H_{k,v} = \frac{(k-v)F_{k,v}}{2k}$$

5. Simetria de $B_n(x)$

Consideremos o ponto $x = \frac{1}{2}$ sobre o eixo dos

x . Os pontos x e $1-x$ estão situados simetricamente com relação a este ponto e o parâmetro $z=x(1-x)$ não muda de valor se substituirmos x por $1-x$. Portanto de (2.1.2.9) obtemos

$$B_{2k}(1-x) = B_{2k}(x) , \quad (2.1.2.11)$$

sendo o gráfico de $B_{2k}(x)$ simétrico com relação à linha $x = \frac{1}{2}$.

O fator $\sum_{v=0}^{k-2} H_{k,v} z^{k-v-1}$ em (2.1.2.10) tem o mesmo valor para os pontos x e $1-x$. O fator $(1-2x)$ tem o mesmo valor absoluto mas de sinais opostos para estes pontos. Portanto

$$B_{2k-1}(1-x) = -B_{2k-1}(x) , \quad (2.1.2.12)$$

sendo o gráfico de $B_{2k-1}(x)$ simétrico centralmente com relação ao ponto $x = \frac{1}{2}$.

6. O comportamento de polinômios de Bernoulli sobre o intervalo $[0,1]$.

Necessitamos de algumas propriedades dos polinômios

$$Y_n(x) = B_n(x) - B_n$$

que são essencialmente as mesmas de $B_n(x)$, mas mais convenientes para nossos propósitos. Para $n=2k$, de (2.1.2.9) obtemos

$$(-1)^k Y_{2k}(x) = \sum_{v=0}^{k-2} F_{k,v} z^{k-v}, \quad (2.1.2.14)$$

com raízes 0 e 1, de multiplicidade 2 para $k \geq 2$.

Para $0 < x < 1$ o parâmetro z permanecerá dentro dos limites $0 < z \leq \frac{1}{4}$, e desde que $F_{k,v} > 0$

$$(-1)^k Y_{2k}(x) > 0, \quad \text{para } 0 < x < 1.$$

No intervalo aberto $0 < x < 1$ o polinômio $Y_{2k}(x)$ não tem zeros e tem o mesmo sinal de $(-1)^k$.

Quando x varia de zero a $\frac{1}{2}$, a função $z = x(1-x)$ crescerá de zero a $\frac{1}{4}$ e quando x varia de $\frac{1}{2}$ a 1,

$$(-1)^k Y_{2k-1}(x) > 0 \quad \text{para } 0 < x < \frac{1}{2},$$

$$(-1)^k Y_{2k-1}(x) < 0 \quad \text{para } \frac{1}{2} < x < 1.$$

Daremos agora uma tabela dos cinco primeiros polinômios de Bernoulli.

$$B_0(x) = 1$$

$$B_1(x) = x - \frac{1}{2}$$

$$B_2(x) = x^2 - x + \frac{1}{6}$$

$$B_3(x) = x^3 - \frac{3}{2} x^2 + \frac{1}{2} x$$

$$B_4(x) = x^4 - 2x^3 + x^2 - \frac{1}{30}$$

$$B_5(x) = x^5 - \frac{5}{2} x^4 + \frac{5}{3} x^3 - \frac{1}{6} x$$

A Figura 1 ilustra o comportamento dos polinômios de Bernoulli $B_n(x)$ sobre o intervalo $[0,1]$.

2.1.3 - FUNÇÕES PERIÓDICAS RELACIONADAS AOS POLINÔMIOS DE BERNOULLI

Para estudarmos certas questões relacionadas aos polinômios de Bernoulli introduziremos as funções $B_n^*(x)$, de período 1, definidas pelas condições

$$B_n^*(x) = B_n(x) , \quad 0 \leq x < 1$$

$$B_n^*(x+1) = B_n^*(x) .$$

$B_0^*(x)$ é uma constante igual a 1, $B_1^*(x)$ é uma função descontínua com um salto de -1 para cada inteiro; para $n > 1$, $B_n^*(x)$ é uma função contínua.

bre $[0,1]$, então para qualquer $x \in [0,1]$ temos a relação

$$f(x) = \int_0^1 f(t) dt + \sum_{k=1}^{v-1} \frac{B_k(x)}{k!} \left[f^{(k-1)}(1) - f^{(k-1)}(0) \right] - \frac{1}{v!} \int_0^1 f^{(v)}(t) \left[B_v^*(x-t) - B_v^*(x) \right] dt \quad (2.1.3.2)$$

[A prova encontra-se em Krylov (1962), p. 16]

2.2 - FÓRMULA DO ERRO DE EULER-MACLAURIN PARA A REGRA DOS TRAPÉZIOS [Krylov, (1962)]

Considerando o problema de aumentar a precisão de uma fórmula de quadratura, removendo a parte principal do resto, o caminho mais apropriado para tal, depende

$$\rho_{2V}(f) = - \frac{nh^{2V+1}}{(2V)!} B_{2V} f^{(2V)}(\xi),$$

CAPITULO 3

QUADRATURA DE ROMBERG

Quadratura de Romberg é essencialmente uma aplicação do procedimento de extrapolação de Richardson para a fórmula da soma de Euler-MacLaurin. O nome Romberg é devido ao fato de que foi ele o primeiro a ver que o processo de eliminação do termo do erro em quadratura numérica podia ser

3.1 - A IDÉIA BÁSICA

O método de quadratura numérica que passamos a estudar provém de uma idéia muito bem formulada para aumentar a precisão de aproximações já disponíveis, dadas pela regra dos trapézios, por meio de um algoritmo de execução extremamente simples.

Sabemos do sensível aumento da precisão obtida na aproximação da integral quando, ao invés da regra dos trapézios, usamos o método de Simpson, para o mesmo número de pontos dados. Este não é um resultado do acaso, pois, com exame das fórmulas de erro dos dois processos, limitados respectivamente por:

$$\frac{h^2}{12} M_2 (x_n - x_0) ; \quad M_2 : \text{valor máximo de } |f''(x)|$$

$$\frac{h^4}{180} M_4 (x_n - x_0) ; \quad M_4 : \text{valor máximo de } |f^{IV}(x)|$$

aproximações ainda melhores para valores suficientemente pequenos de h .

A regra trapezoidal para calcular (I.1)

$$I = \int_a^b f(x) dx \approx h \sum_{k=0}^{n-1} f(a+kh) = T(h) \quad (3.1.1)$$

onde $h = (b-a)/n$ e $\sum$ indica que o primeiro e último termo são multiplicados por $1/2$, é essencialmente um h^2 -processo.

Como consequência, Romberg concluiu que poderia melhorar a precisão da regra trapezoidal aplicando a "extrapolação para o limite" de Richardson, que consiste em, combinar de modo especial, aproximações de certa ordem de um valor procurado, dadas por um certo método, para obter aproximações de ordem sucessivamente maior. No caso é fundamental o teorema seguinte.

ro tão pequeno quanto quisermos. No entanto, isto acarretará sempre um sensível acréscimo do esforço computacional, devido aos sucessivos cálculos de valores de $f(x)$, com o conseqüente acréscimo do erro de arredondamento. Usa-se então a extrapolação para o limite que, combinando poucas aproximações da regra dos trapézios, fornece ótimas aproximações com pequeno esforço computacional.

Se dividíssemos $[x_0, x_n]$ em $2n$ subintervalos teríamos a aproximação $T(h/2)$, dada por (3.1.2). Do teorema 3.1.1, segue que:

$$T(h) = T^{(0)}(h) = I + C_0 h^2 + o(h^4) \quad (3.1.3)$$

$$T^{(0)}(h/2) = I + C_0 (h/2)^2 + o(h^4) \quad (3.1.4)$$

onde C_0 é constante e $o(h^4)$ representa o resto da série (3.1.2).

Facilmente removemos o termo de ordem 2, a-

Uma verificação simples revela que o valor $T^{(1)}(h)$ obtido é idêntico ao valor resultante da aplicação da regra de Simpson, com intervalo h . Desde que a regra de Simpson é conhecida como um h^4 -processo, o princípio de Richardson pode ser aplicado novamente.

Se calculássemos agora outra aproximação $T^{(0)}(h/4)$ com a divisão de $[x_0, x_n]$ em $4n$ intervalos, poderíamos obter, analogamente, aproximações de ordem 4, $T^{(1)}(h/2)$:

$$T^{(1)}(h/2) = \frac{4T^{(0)}(h/4) - T^{(0)}(h/2)}{3}$$

valendo então:

$$T^{(1)}(h/2) = I + C_1' (h/2)^4 + O(h^6) \quad (3.1.7)$$

Combinando adequadamente (3.1.6) e (3.1.7)

temos:

onde C_2'' é obtido com convenientes simplificações.

O valor $T^{(2)}(h)$ obtido é novamente familiar, a saber, o valor obtido da fórmula de Newton-Cotes, sobre 5 pontos, de ordem 6.

O processo pode ser continuado num caminho

óbvio, chegando à fórmula de recursão geral

$$T^{(m)}(h) = \frac{4^m T^{(m-1)}(h/2) - T^{(m-1)}(h)}{4^m - 1} \quad (3.1.9)$$

No próximo passo com $m=3$, em (3.1.9), obtemos:

$$T^{(3)}(h) = \frac{64T^{(2)}(h/2) - T^{(2)}(h)}{63}$$

que produz um valor que não pode ser expresso em termos de

Estes valores são arranjados numa matriz triangular que chamaremos T-tabela:

$$T_0^{(0)}$$

$$T_0^{(1)}$$

$$T_1^{(0)}$$

$$T_0^{(2)}$$

$$T_1^{(1)}$$

$$T_2^{(0)}$$

$$T_0^{(3)}$$

$$T_1^{(2)}$$

$$T_2^{(1)}$$

$$T_3^{(0)}$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\ddots$$

onde cada entrada $T_m^{(k)}$ ($m \neq 0$) é calculada a partir da fórmula (3.1.9),

$$T_m^{(k)} = \frac{4^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{4^m - 1} ; \quad m = 1, 2, 3, \dots \quad (3.2.1)$$

$$k = 0, 1, \dots$$

Exemplo numérico [B.R.S. (1963)]

$$\int_1^2 x^{-1} dx = \ln(2) = 0.693147180\dots$$

la é obtida

.750 000 000

.708 333 333 .694 444 444

.697 023 809 .693 253 967 .693 174 603

.694 121 851 .693 154 532 .693 147 901 .693 147 479

.693 391 202 .693 147 652 .693 147 193 .693 147 182 .693 147 181

Já podemos observar que há convergência para $\ln 2$ nas colunas e nas diagonais.

A Função QUAD para integração clássica de Romberg é dada no Apêndice e foi usada para integrar as funções dadas na Tabela 6.2.1 (pp. 166-178).

3.3 - PROPRIEDADES DO MÉTODO DE ROMBERG

3.3.1 - POLINÔMIOS GERADORES DA T-TABELA

[B.R.S. (1963)]

Cada entrada $T_m^{(k)}$ da T-tabela é uma combinação linear dos valores da coluna T_0 :

$$c_{m,m-i} = \frac{4^m c_{m-1,m-i} - c_{m-1,m-1-i}}{4^{m-1}} \quad (3.3.1.2)$$

para $i = 0, 1, 2, \dots, m$ com $c_{m-1,m} = c_{m-1,-1} = 0$, $c_{0,0} = 1$.

Consideremos agora os polinômios geradores:

$$t_m(z) = \sum_{k=0}^m c_{m,k} z^k \quad (3.3.1.3)$$

Usando (3.3.1.2) e agrupando os termos de mesmo grau, obtemos:

$$\begin{aligned} t_m(z) &= c_{m,0} z^0 + c_{m,1} z^1 + c_{m,2} z^2 + \dots + c_{m,m} z^m = \\ &= \frac{1}{4^{m-1}} \left[4^m c_{m-1,0} z^0 + (4^m c_{m-1,1} - c_{m-1,0}) z^1 + \right. \\ &\quad + (4^m c_{m-1,2} - c_{m-1,1}) z^2 + \dots + (4^m c_{m-1,m-1} - \\ &\quad \left. - c_{m-1,m-2}) z^{m-1} - c_{m-1,m-1} z^m \right] = \frac{1}{4^{m-1}} \left[4^m (c_{m-1,0} z^0 + \right. \\ &\quad + c_{m-1,1} z^1 + c_{m-1,2} z^2 + \dots + c_{m-1,m-1} z^{m-1}) - \\ &\quad - c_{m-1,0} z^1 - c_{m-1,1} z^2 - \dots - c_{m-1,m-2} z^{m-1} - \\ &\quad \left. - c_{m-1,m-1} z^m \right] = \frac{1}{4^{m-1}} \left[4^m t_{m-1}(z) - z(c_{m-1,0} z^0 - \right. \\ &\quad \left. - c_{m-1,1} z^1 - \dots - c_{m-1,m-1} z^{m-1}) \right] \\ t_m(z) &= \frac{1}{4^{m-1}} \left[(4^m - z) + \dots \right] \end{aligned}$$

Temos assim uma fórmula de recorrência para os polinômios geradores, a qual, por indução, nos permite escrever:

$$t_m(z) = \frac{(1 - \frac{z}{4})(1 - \frac{z}{16}) \cdots (1 - \frac{z}{4^m})}{(1 - \frac{1}{4})(1 - \frac{1}{16}) \cdots (1 - \frac{1}{4^m})} \quad (3.3.1.4)$$

Extraímos as seguintes conclusões:

$$1. \quad \sum_{i=0}^m c_{m,i} = t_m(1) = 1, \quad (3.3.1.5)$$

2. Os $c_{m,i}$ são de sinais alternados com relação a i , isto é:

$$(-1)^i c_{m,i} > 0 \quad \text{para } i = 0, 1, 2, \dots, m \quad (3.3.1.6)$$

$$3. \quad \sum_{i=0}^m |c_{m,i}| = t_m(-1) < \prod_{k=1}^{\infty} \left(\frac{4^k + 1}{4^k - 1} \right) = 1,969 \dots < 2, \quad (3.3.1.7)$$

que garante a estabilidade numérica do método.

$$\sum_{k=0}^m c_{m,k} (4^k - 1) z^k = - \sum_{k=0}^{m-1} c_{m-1,k} z^{k+1},$$

a qual, pela equação dos coeficientes, produz $c_{m,k} =$

$$= - \frac{c_{m-1,k-1}}{4^k - 1}. \text{ Repetidas aplicações desta última equa-}$$

ção nos dão:

$$6. \quad c_{m,k} = \frac{(-1)^k c_{m-k,0}}{3 \cdot 15 \cdot 63 \cdot \dots \cdot (4^k - 1)} \quad (3.3.1.10)$$

Uma consequência imediata de (3.3.1.10) é:

$$7. \quad \lim_{m \rightarrow \infty} c_{m,m-k} = 0, \text{ para todo } k. \quad (3.3.1.11)$$

Além disso, desde que:

$$8. \quad \lim_{m \rightarrow \infty} c_{m,0} = 1 / \prod_{k=1}^{\infty} (1 - \frac{1}{4^k}) = c_0 = 1.452353642... \quad (3.3.1.12)$$

existe, segue de (3.3.1.10) que

$$9. \quad c_k = \lim_{m \rightarrow \infty} c_{m,k} = \frac{(-1)^k \lim_{m \rightarrow \infty} c_{m-k,0}}{3 \cdot 15 \cdot 63 \cdot \dots \cdot (4^k - 1)} \quad (3.3.1.13)$$

TEOREMA 3.3.1.1

A convergência da coluna T_0 da T-tabela implica na convergência de todas as demais colunas para o mesmo limite.

PROVA: Segue de (3.3.1.1) e (3.3.1.5) desde que, em cada coluna m é constante.

TEOREMA 3.3.1.2

A convergência da coluna T_0 da T-tabela implica na convergência de todas as seqüências diagonais, isto é, as seqüências $T_m^{(k)}$, k constante, $m \rightarrow \infty$, para o mesmo limite.

PROVA:

Primeiramente observamos que a fórmula (3.3.1.1) estabelece uma transformação linear, seqüência a seqüência, entre as seqüências $T_0^{(k)}$, $T_0^{(k+1)}$, $T_0^{(k+2)}$, ... e

Temos $\lim_{k \rightarrow \infty} T_0^{(k)} = I$ e também as propriedades

$$- \sum_{i=0}^m c_{m,i} = 1;$$

$$- \sum_{i=0}^m |c_{m,i}| < 2;$$

- $\lim_{m \rightarrow \infty} c_{m,m-k} = 0$, para todo k , a qual diz que toda coluna da matriz (3.3.1.15) converge para zero.

Então, segundo Toeplitz [Albrecht (1973), p. 186],

$$\lim_{m \rightarrow \infty} \sum_{i=0}^m c_{m,i} \cdot T_0^{(i)} = I, \text{ isto é,}$$

$$\lim_{m \rightarrow \infty} T_m^{(0)} = I, \text{ o que garante o teorema.}$$

Os teoremas 3.3.1.1 e 3.3.1.2 permitem concluir que, se $f(x)$ é Riemann integrável sobre $[a,b]$, o método

onde $h = (b-a)/2^{m+k}$, $k = 0, 1, \dots$

Para calcularmos os coeficientes d_j^m observa-

mos primeiramente que:

$$T_0^{(k+m)} = h \sum_{j=0}^{2^{k+m}} f(a+jh)$$

$$T_0^{(k+m-1)} = 2h \sum_{j=0}^{2^{k+m-1}} f(a+2jh) \quad (3.3.2.2)$$

$$\vdots$$

$$T_0^{(k)} = 2^m h \sum_{j=0}^{2^k} f(a+2^m jh)$$

De acordo com (3.3.1.1),

$$T_m^{(k)} = \sum_{i=0}^m c_{m,i} T_0^{(k+m-i)} = h \sum_{i=0}^m \sum_{j=0}^{2^{k+m-i}} c_{m,i} 2^i f(a+2^i jh) \quad (3.3.2.3)$$

$$= h \sum_{j=0}^{2^{k+m}} \left[c_{m,0} + 2c_{m,1} + 4c_{m,2} + \dots + 2^p c_{m,p} \right] f(a+jh),$$

onde 2^p é a maior potência de 2 que divide j . Portanto,

Valores numéricos dos d's.

m	j=1,3,5,7,9,...	j = 2,6,10,...	j = 4,12,20,...	j = 8,24,40,...	j = 16,48,80,...
1	1.333 333 333	.666 666 667			
2	1.422 222 222	.533 333 333	.622 222 222		
3	1.444 797 177	.496 649 029	.615 167 547	.612 345 679	
4	1.450 463 048	.487 264 929	.613 684 682	.609 922 190	
	↓	↓	↓	↓	
∞	1.452 353 642	.484 117 881	.613 215 982	.609 117 630	.690 944 322
					↓
					.690 149 774
				→	.690 149 711

Para qualquer m, os coeficientes $d_j^{(m)}$ são positivos e satisfazem

$$\frac{c_0}{3} = 0.484117 \dots \leq d_j^{(m)} \leq c_0 = 1.452353 \dots$$

isto é, nenhum dos $d_j^{(m)}$ pode ser mais que três vezes maior que qualquer outro coeficiente peso.

Portanto, vê-se o contraste favorável com as fórmulas de Newton-Cotes, nas quais os coeficientes peso podem ser negativos (fórmula de 9-pontos para frente).

Um algoritmo para calcular os coeficientes $d_j^{(m)}$ foi publicado por Welsh (1966).

3.4 - ESTUDO DO ERRO NO MÉTODO DE RONBERG

3.4.1 - O TERMO EXATO DO ERRO PARA $T_m^{(k)}$

[B.R.S. (1963)]

Restringindo o intervalo de quadratura para $[0,1]$, o termo do erro ou resto para a regra trapezoidal é dado por

$$T_0^{(0)} - \int_0^1 f(x) dx = \int_0^1 \frac{x(1-x)}{2} f''(x) dx$$

Seja $h = 1/2^k$.

$$\text{Definindo } b_2^k(x) = \begin{cases} \frac{x(1/2^k - x)}{2} & \text{em } 0 \leq x \leq 1/2^k \\ \text{periódica de período } 1/2^k \end{cases}$$

temos

$$T_0^{(k)} - \int_0^1 f(x) dx = \int_0^1 b_2^k(x) \cdot f''(x) dx.$$

Logo, como $b_2^k(x) = b_2(2^k x)/4^k$ vem:

$$T_0^{(k)} - \int_0^1 f(x) dx = 4^{-k} \int_0^1 b_2(2^k x) f''(x) dx \quad (3.4.1.1)$$

onde definimos

$$b_2(x) = \begin{cases} \frac{1}{2} x(1-x) & \text{em } 0 \leq x \leq 1 \\ \text{periódica de período } 1 \end{cases} \quad (3.4.1.2)$$

Destas fórmulas podemos extrair o termo do erro para $T_m^{(k)}$ por indução com relação a m . Assumimos:

$$T_{m-1}^{(k)} = \int_0^1 f(x) dx + 4^{-km} \int_0^1 b_{2m}(2^k x) f^{(2m)}(x) dx, \quad (3.4.1.3)$$

onde $b_{2m}(x)$ é uma função par com período 1, ainda a ser determinada. Esta hipótese é certamente verdadeira para $m = 1$ com $b_2(x)$ definida em (3.4.1.2).

Por substituição de (3.4.1.3) em (3.2.1) obtemos:

$$T_m^{(k)} = \int_0^1 f(x) dx + \frac{4^{-km}}{4^m - 1} \int_0^1 \left[b_{2m}(2^{k+1}x) - b_{2m}(2^k x) \right] f^{(2m)}(x) dx \quad (3.4.1.4)$$

Vamos agora definir as funções

$$\hat{b}_{2m}(x) = \frac{b_{2m}(2x) - b_{2m}(x)}{4^m - 1},$$

$$b_{2m+1}(x) = \int_0^x \hat{b}_{2m}(x) dx, \quad (3.4.1.5)$$

de de $b_{2m+1}(x)$ que, sendo uma integral de uma função par, deve ser ímpar. Portanto,

$$(c) \quad b_{2m+2}(1) = \int_0^1 b_{2m+1}(x) dx = \int_{-\frac{1}{2}}^{\frac{1}{2}} b_{2m+1}(x) dx = 0,$$

o que estabelece também a periodicidade de $b_{2m+2}(x)$. Esta função é par.

De acordo com (3.4.1.5) e as propriedades (a), (b), (c), integrando por partes duas vezes obtemos:

$$\int_0^1 \tilde{b}_{2m}(2^k x) f^{(2m)}(x) dx = 4^{-k} \int_0^1 b_{2m+2}(2^k x) f^{(2m+2)}(x) dx ,$$

que substituindo em (3.4.1.4), vem:

$$T_m^{(k)} = \int_0^1 f(x) dx + 4^{-(m+1)k} \int_0^1 b_{2m+2}(2^k x) f^{(2m+2)}(x) dx , \quad (3.4.1.6)$$

o que completa a indução.

Na verdade, para alcançarmos uma estimativa exata do erro, precisamos ainda de

$$J = \int_0^1 b_{2m+2}(x) dx ,$$

que pode ser derivada de (3.4.1.6) tomando-se $f(x) =$

(2.2.1) conduz, para $f(x) = x^{2m+2}/(2m+2)!$ a:

$$T_0^{(k)} = \int_0^1 f(x) dx = \sum_{p=1}^{m+1} \frac{B_{2p}}{(2p)!} \frac{4^{-kp}}{(2m-2p+3)!}$$

Então, por (3.3.1.1)

$$\begin{aligned} J &= \sum_{j=0}^m c_{m,j} \left[\sum_{p=1}^{m+1} \frac{B_{2p} 4^{-p(m-j)}}{(2p)! (2m-2p+3)!} \right] \\ &= \sum_{p=1}^{m+1} \left[\frac{B_{2p} 4^{-mp}}{(2p)! (2m-2p+3)!} \cdot \sum_{j=0}^m c_{m,j} 4^{pj} \right] \end{aligned}$$

Como $z = 4, 16, 64, \dots, 4^m$ são os zeros de $\sum_{j=0}^m c_{m,j} z^j = t_m(z)$, J se reduz ao termo correspondente a $p = m+1$, isto é,

$$J = \frac{B_{2m+2} 4^{-m(m+1)}}{(2m+2)!} \cdot t_m(4^{m+1}) = \frac{(-1)^m B_{2m+2}}{(2m+2)! 2^{m(m+1)}} \quad (3.4.1.8)$$

desde que, em virtude de (3.3.1.4);

$$t_m(4^{m+1}) = \frac{(1 - \frac{4^{m+1}}{4})(1 - \frac{4^{m+1}}{16}) \dots (1 - \frac{4^{m+1}}{4^m})}{(1 - \frac{1}{4})(1 - \frac{1}{16}) \dots (1 - \frac{1}{4^m})} =$$

para algum ξ no intervalo $0 \leq \xi \leq 1$, sendo os B 's os números de Bernoulli.

O resultado (3.4.1.10) estabelece o seguinte:

TEOREMA 3.4.1.1.

Se $f(x)$ tem $2m+2$ derivadas contínuas no in-

tervalo fechado $[a,b]$, então para $k \rightarrow \infty$:

$$T_m^{(k)} - \int_a^b f(x) dx = O(4^{-k(m+1)}) \quad (3.4.1.11)$$

Este resultado diz que a entrada $T_m^{(k)}$ é cerca de 4^{m+1} vezes mais precisa que a entrada imediatamente acima dela.

3.4.2 - OUTRA EXPRESSÃO EXPLÍCITA PARA O TERMO DO ERRO

[Hävie (1969)]

O estudo do termo do erro para o algoritmo de Romberg fica simplificado quando ele é expresso em termos de polinômios de Bernoulli e suas funções periódicas relacionadas. Derivamos expressões explícitas para as funções

$x = 0$ para $x = \frac{1}{2}$ e daí decrescem para zero.

PROVA: Usamos indução com relação a m .

A afirmação é obviamente verdadeira para $m=0$.

Se $b_{2m}(x)$ é monotonicamente crescente de $x=0$

para $x = \frac{1}{2}$ e tem a propriedade $b_{2m}(1-x) = b_{2m}(x)$ (que para u ma função par segue da periodicidade) temos:

$$(4^m - 1) \hat{b}_{2m}(x) = b_{2m}(2x) - b_{2m}(x) \quad \begin{cases} > 0 & \text{para } 0 < x < \frac{1}{3} \\ = 0 & \text{para } x = \frac{1}{3} \\ < 0 & \text{para } \frac{1}{3} < x < \frac{1}{2} \end{cases}$$

Por definição temos:

$$b_{2m+1}(x) = \int_0^x \hat{b}_{2m}(x) dx$$

Esta função é monotonicamente crescente de $x = 0$ até $\frac{1}{3}$ e então decresce até $x = \frac{1}{2}$ onde, sendo ímpar e periódica, com período 1, deve se anular. Portanto temos:

positiva entre 0 e 1, o que completa a prova.

Além disso, as funções $b_{2m+2}(x)$ satisfazem à equação diferencial de 2ª ordem

$$b_{2m+2}''(x) = \frac{b_{2m}(2x) - b_{2m}(x)}{4^m - 1} \quad (3.4.2.1)$$

com as condições:

$$b_{2m+2}(0) = b_{2m+2}'(0) = 0$$

Para $m = 0$ (conforme (3.4.1.2)), temos:

$$b_2(x) = \frac{1}{2}x(1-x) ; \quad 0 \leq x \leq 1.$$

Derivemos expressões explícitas para as funções $b_{2m+2}(x)$ em termos de funções de Bernoulli.

Para obtermos uma representação explícita para a função $b_{2m+2}(x)$ (Teorema 3.4.2.3) precisamos do resultado dado a seguir, que introduz outro polinômio gerador:

TEOREMA 3.4.2.2

Temos os polinômios geradores:

$$(-1)^m 2^{-m(m+1)} \prod_{j=1}^m \left[\frac{1 - (z/4^j)}{1 - (1/4^j)} \right] = \sum_{i=0}^m \frac{c_{m,m-i}}{4^{i(m+1)}} z^i \quad (3.4.2.2)$$

Utilizando este teorema e mais algumas propriedades das funções de Bernoulli dadas em (2.1.3) e algumas propriedades dos coeficientes $c_{m,i}$ dadas em (3.3.1), podemos provar o

TEOREMA 3.4.2.3

Para a função $b_{2m+2}(x)$ vale a representação explícita

$$b_{2m+2}(x) = \frac{(-1)^m B_{2m+2}(x)}{2^{m(m+1)} (2m+2)!} \left[1 - \sum_{i=0}^m c_{m,i} \frac{B_{2m+2}^*(2^i x)}{B_{2m+2}} \right] \quad (3.4.2.4)$$

onde os coeficientes $c_{m,i}$ obedecem a fórmula de recursão

$$c_{m,i} = \frac{4^m}{4^i} \frac{c_{m-1,i} - c_{m-1,i-1}}{4^m - 1} \quad (3.4.2.5)$$

$i = 0, 1, \dots, m$, com $c_{m-1,m} = c_{m-1,-1} = 0$ e $c_{0,0} = 1$, onde $B_{2m+2}^*(x)$ são as funções de Bernoulli.

PROVA: Para $m = 0$ calculamos

$$b_2(x) = \frac{B_2(x)}{2!} \left[1 - \frac{B_2^*(x)}{B_2} \right] = \frac{1}{2} x(1-x) \quad 0 \leq x \leq 1$$

$$\frac{b_{2m}(2x) - b_{2m}(x)}{4^m - 1} = - \frac{(-1)^m}{2^{m(m+1)}} \sum_{i=0}^{m-1} \frac{4^m c_{m-1,i}}{4^m - 1},$$

$$\frac{B_{2m}^*(2^i x) - B_{2m}^*(2^{i+1} x)}{(2m)!}$$

$$= - \frac{(-1)^m}{2^{m(m+1)}} \sum_{i=0}^m 4^i c_{m,i} \frac{B_{2m}^*(2^i x)}{(2m)!} = b_{2m+2}''(x),$$

usando a definição de $c_{m,i}$ dada pela equação (3.4.2.5) e a relação entre $c_{m,i}$ e $c_{m,m-i}$ dada pela equação (3.4.2.3). Isto completa a prova do teorema.

Damos a seguir alguns lemas decorrentes dos teoremas anteriores.

LEMA 3.4.2.1

O termo do erro (3.4.1.6), denotado daqui por diante por $R_m^{(k)}$, pode ser expresso nas formas alternativas

$$R_m^{(k)} = -2^{-(2m+1)k} \int_0^1 f^{(2m+1)}(x) b_{2m+2}'(2^k x) dx \quad (3.4.2.6)$$

LEMA 3.4.2.2

Temos a seguinte relação:

$$b_{2m+2}(x+1/2) - b_{2m+2}(x) = - \frac{(-1)^m c_{m,0}}{2^{m(m+1)}} \cdot \frac{B_{2m+2}^*(x+1/2) - B_{2m+2}^*(x)}{(2m+2)!} \quad (3.4.2.8)$$

O lema (3.4.2.2) é provado usando a equação (3.4.2.4).

3.4.3 - CÁLCULO NUMÉRICO DAS FUNÇÕES $b_{2m+2}(x)$

[Hâvie (1969)]

Para o cálculo numérico é mais conveniente tratarmos com as funções escalonadas $\tilde{b}_{2m+2}(x)$, definidas por

des dos polinômios e funções de Bernoulli, obtemos:

$$\tilde{b}_{2m+2}(1/2) = 2 \left[1 - \left(\frac{1}{4}\right)^{m+1} \right] c_{m,0} \quad (3.4.3.4)$$

As formas limites para as funções são facil-

mente obtidas da equação (3.4.3.3), observando que

$$\lim_{m \rightarrow \infty} \frac{B_{2m+2}^*(x)}{B_{2m+2}} = \cos(2 \pi x) , \quad (3.4.3.5)$$

e a equação (3.3.1.13).

Consequentemente obtemos,

$$\begin{aligned} \tilde{b}_{\infty}(x) = 1 - c_0 \left[\cos(2 \pi x) - \frac{\cos(4 \pi x)}{3} + \frac{\cos(8 \pi x)}{3 \cdot 15} - \right. \\ \left. - \frac{\cos(16 \pi x)}{3 \cdot 15 \cdot 63} + \dots \right] \end{aligned} \quad (3.4.3.6)$$

que é facilmente verificado usando (2.1.1.5).

Se definirmos o desvio relativo da forma limite

$$\tilde{d}_{2m+2}(x) = \left| \frac{\tilde{b}_{\infty}(x) - \tilde{b}_{2m+2}(x)}{\tilde{b}_{\infty}(x)} \right|$$

então segue de cálculos numéricos que para

$$m \geq 11, \quad \tilde{d}_{2m+2}(x) < 10^{-3}, \quad x \in [.05, .5]$$

e

$$m \geq 16, \quad \tilde{d}_{2m+2}(x) < 10^{-7}, \quad x \in [.05, .5]$$

Daremos uma tabela onde estão as funções $\tilde{b}_{2m+2}(x)$ para alguns valores de m .

Funções $\tilde{b}_{2m+2}(x)$ para alguns valores de m .

m					
-----	--	--	--	--	--

FIG. 3.4.3.1

3.4.4 - ESTIMATIVA PARA O TERMO DO ERRO $R_m^{(k)}$

[Hâvie (1969)]

O termo do erro $R_m^{(k)}$ pode ser escrito

$$R_m^{(k)} = \frac{(-1)^m 4^{-k(m+1)} B_{2m+2}}{2^{m(m+1)} (2m+2)!} \int_0^1 f^{(2m+2)}(x) \tilde{b}_{2m+2}(2^k x) dx \quad (3.4.4.1)$$

que segue da equação (3.4.1.6) quando expressamos $b_{2m+2}(x)$

Observamos aqui que, em casos muito excepcionais, o erro na regra trapezoidal não é da forma (3.1.2) e o método de Romberg pode não melhorar a convergência.

Em [B.R.S. (1963), pp. 213-214] há um exemplo ilustrando uma tal situação.

3.5 - OUTRAS SUBDIVISÕES ALÉM DA BISSECÇÃO

[B.R.S. (1963)]

Como vimos, o método clássico de Romberg provém de sucessivas bissecções no intervalo de integração. A vantagem é que usa todos os valores funcionais calculados para cada estágio. (De outra maneira, o número de valores calculados cresce exponencialmente).

Podemos verificar imediatamente que a fórmula (3.2.1) para calcular as entradas da T-tabela descrevem

Esta tabela não é muito harmônica, pois como vimos, o termo do erro da regra trapezoidal é uma função de h^2 o que claramente sugere um procedimento de extrapolação para obter o valor $T(0) = I$. Mas sugere também, que o método de Romberg não é o único método possível de extrapolação.

Na verdade, poderíamos montar qualquer sequência monotonicamente crescente de inteiros $n_0, n_1, n_2, \dots$, e então calcular os valores trapezoidais

$$T_0^{(k)} = h_k \sum_{j=0}^{n_k} f(a+jh_k) \quad (3.5.2)$$

onde $h_k = (b-a)/n_k$, $k=0,1,2,\dots$, e extrapolar $F(0)$ dos valores $F(n_k^{-2}) = T_0^k$, isto é, construir a correspondente tabela de Neville-Aitken.

O desejo de usar tal modificação do método de Romberg é sugerido pelo número rapidamente crescente de pontos, pois o número de valores calculados cresce exponencialmente. Pareceria proveitoso escolher uma sequência menos rapidamente crescente que $n_k = 2^k$. Consequentemente, variações do método de Romberg foram propostas, baseadas no seguinte resultado:

Seja $\{n_k\}$ uma sequência de inteiros tais que $n_{k+1}/n_k > c > 1$ para todo k . Pode-se mostrar que, se a sequência cresce pelo menos exponencialmente, as condições de

$$T_m^{(k)} = \frac{h_k^2 T_{m-1}^{(k+1)} - h_{k+m}^2 T_{m-1}^{(k)}}{h_k^2 - h_{k+m}^2} \quad (3.5.3)$$

as colunas e diagonais convergem para I.

Como exemplo, consideremos a sequência $\{n_k\} =$

$\{1, 2, 3, 6, 9, 18, 27, 54, \dots\}$. Neste caso as fórmulas para gerar a T-tabela são

$$T_m^{(k)} = \begin{cases} \frac{3^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{3^m - 1} & ; \text{ m par} \\ \frac{4 \cdot 3^m T_{m-1}^{(k+1)} - 3 T_{m-1}^{(k)}}{4 \cdot 3^m - 3} & ; \text{ m ímpar e k par} \\ \frac{3 \cdot 3^m T_{m-1}^{(k+1)} - 4 T_{m-1}^{(k)}}{3 \cdot 3^m - 4} & ; \text{ m ímpar e k ímpar} \end{cases}$$

de fato:

Para m par e k ímpar

$$T_m^{(k)} = \frac{\frac{1}{2^2 \cdot 3^{k-1}} T_{m-1}^{(k+1)} - \frac{1}{2^2 \cdot 3^{k+m-1}} T_{m-1}^{(k)}}{\frac{1}{2^2 \cdot 3^{k-1}} - \frac{1}{2^2 \cdot 3^{k+m-1}}} = \frac{3^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{3^m - 1}$$

1b. Para m ímpar e k par

$$T_m^{(k)} = \frac{\frac{1}{3^k} T_{m-1}^{(k+1)} - \frac{1}{2^2 \cdot 3^{k+m-1}} T_{m-1}^{(k)}}{\frac{1}{3^k} - \frac{1}{2^2 \cdot 3^{k+m-1}}} = \frac{4 \cdot 3^m T_{m-1}^{(k+1)} - 3 T_{m-1}^{(k)}}{4 \cdot 3^m - 1}$$

1c. Para m ímpar e k ímpar

$$T_m^{(k)} = \frac{\frac{1}{2^2 \cdot 3^{k-1}} T_{m-1}^{(k+1)} - \frac{1}{3^{k+m}} T_{m-1}^{(k)}}{\frac{1}{2^2 \cdot 3^{k-1}} - \frac{1}{3^{k+m}}} = \frac{3 \cdot 3^m T_{m-1}^{(k+1)} - 4 T_{m-1}^{(k)}}{3 \cdot 3^m - 4}$$

Casualmente, o valor $T_3^{(0)}$ é o mesmo que seria obtido pela fórmula de Newton-Cotes de 7-pontos com seis subintervalos de todo o intervalo de integração.

$$T_3^{(0)} = \frac{b-a}{840} \left[41f_0 + 216f_1 + 27f_2 + 272f_3 + 27f_4 + 216f_5 + 41f_6 \right]$$

n_k	$T_0^{(k)}$				
1	.750 000 000				
2	.708 333 333	.694 444 444			
3	.7	.693 333 333	.693 194 444		
6	.694 877 345	.693 169 794	.693 149 352	.693 148 064	
9	.693 917 602	.693 149 808	.693 147 310	.693 147 204	.693 147 193

Podemos verificar que nenhuma vantagem sobre o método Romberg é encontrada.

Esta modificação é mais susceptível a erros de arredondamento, desde que a soma dos valores absolutos dos coeficientes que expressam as entradas da T-tabela como combinação linear de $T_0^{(k)}$ é aqui 3.5, portanto maior que para o método de Romberg onde é inferior a 2.0.

Outra sequência sugerida foi $n_k = k+1$, que causa séria instabilidade numérica devido a arredondamento e portanto, não pode ser usada praticamente. Contudo, produz uma T-tabela que converge, desde que $m \leq 15$.

3.6 - MÉTODO DE ROMBERG: EXTENSÃO PARA N-VARIÁVEIS

[Anders (1966)]

Nesta seção o método de integração numérica

de Romberg é estendido para integração multidimensional.

Como vimos, Romberg mostrou que a diferença entre a integral (I.1) e os elementos que compõem a primeira coluna da T-tabela é $O(h^2)$ e que esta ordem aumenta para a direita com a mesma potência,

Denotando $f(h_1, \dots, h_m) = O(h^m)$, queremos dizer que existe uma constante C para a qual $|f(h_1, \dots, h_n)| < Ch_1^{\beta_1} \dots h_n^{\beta_n}$, com $h_1, h_2, \dots, h_n$ tendendo simultaneamente para zero de tal modo que as razões h_i/h_j sejam limitadas ($i=1, \dots, n$). Os β_i são números reais não negativos satisfazendo $\beta_1 + \beta_2 + \dots + \beta_n = m$.

Aqui a T-tabela é aplicada para aproximar a

TEOREMA 3.6.1

Seja $f(x_1, \dots, x_n)$ com $2r+2$ derivadas parciais definidas e limitadas sobre $[a_i, b_i]$, $i=1, \dots, n$, em cada variável. Sejam os elementos da T-tabela usados para aproximar a integral J onde $T_o^{(k)}$ é definido por

$$T_o^{(k)} = \frac{b_1 - a_1}{2^{k+1}} \dots \frac{b_n - a_n}{2^{k+1}} \sum_{\lambda_i=0}^{2^k-1} \left[f(a_1 + \lambda_1 h_1, \dots, a_n + \lambda_n h_n) + f(a_1 + (\lambda_1 + 1) h_1, \dots, a_n + (\lambda_n + 1) h_n) \right],$$

$$k = 0, 1, \dots, \sum_{\lambda_i=0}^{2^k-1} \left[\dots \right] = \sum_{\lambda_1=0}^{2^k-1} \dots \sum_{\lambda_n=0}^{2^k-1}$$

e os $T_m^{(k)}$ são definidos exatamente como em (3.2.1).

Então $I - T_m^{(k)}$ é $O(h^{2m+2})$, $m=0, 1, \dots, r$ onde

$h_i = (b_i - a_i)/2^k$, $i=1, \dots, n$; $k=0, 1, \dots$

EXEMPLO 3.6.2

$$\int_0^{\frac{1}{2}} \int_0^{\frac{1}{2}} \int_0^{\frac{1}{2}} x^2 y^2 z^2 dx dy dz = 7.2338 \times 10^{-5}$$

T-tabela

24.414×10^{-5}		
10.300×10^{-5}	5.595×10^{-5}	
7.496×10^{-5}	6.561×10^{-5}	6.626×10^{-5}

Continuando desta maneira haverá convergência para o valor correto da integral.

3.7 - CONCLUSÕES

Observamos que no método de Romberg os pesos explícitos não são requeridos, mas são positivos (em alguns

te resultado, embora possam reduzir o tempo de cálculo, são mais susceptíveis a erros de arredondamento, causando séria instabilidade numérica.

Observamos também que o método de Romberg é de fácil implementação, ou seja, a regra trapezoidal é uma fórmula bastante simples e a fórmula de recursão do método é de manejo fácil e prático, permitindo, muitas vezes ter uma idéia do erro cometido.

Finalmente observamos que o método de Romberg pode ser facilmente estendido para integração em várias dimensões.

CAPÍTULO 4

MODIFICAÇÕES DO MÉTODO DE ROMBERG

Várias sugestões têm sido feitas objetivando possíveis melhoramentos e generalizações do método de Rom-

Damos também uma modificação do método de Romberg baseada em extrapolação racional com determinação de limites superior e inferior para o valor correto da integral.

4.1 - INTEGRAÇÃO DE ROMBERG COM O USO DE VALORES RETANGULARES

O acúmulo de erros de truncamento ou arredondamento no cálculo recursivo dos valores trapezoidais, na quadratura de Romberg, introduzem resíduos desnecessários para o cálculo de aproximações elevadas. Introduziremos agora aproximações que podem ser obtidas diretamente de valores retangulares.

$$T_0^{(k+1)} = \frac{1}{2} (T_0^{(k)} + M_0^{(k)}) ; \quad k = 0, 1, 2, \dots \quad (4.1.1.1)$$

Quadratura de Romberg tira partido da representação da diferença entre o valor trapezoidal $T_0^{(k)}$ e o valor verdadeiro I da integral, que, pela fórmula (3.1.2)

será:

$$T_0^{(k)} - I = 4^{-k} C_1 + 4^{-2k} C_2 + 4^{-3k} C_3 + \dots \quad (4.1.1.2)$$

onde os coeficientes C_j são constantes quando o integrando e o intervalo de integração são fixos.

Uma expressão análoga envolvendo $M_0^{(k)}$ é:

$$M_0^{(k)} - I = -\frac{1}{2} 4^{-k} C_1 - \frac{7}{8} 4^{-2k} C_2 - \frac{31}{32} 4^{-3k} C_3 - \dots \quad (4.1.1.3)$$

com os mesmos coeficientes C_j da (4.1.1.2)

ta, enquanto que os valores em negrito formarão a tabela-esqueleto.

Quando a tabela-esqueleto é calculada, o uso da equação (4.1.1.5) nos permite inserir os valores trapezoidais de uma dada coluna sem conhecer os valores trapezoidais ausentes em colunas mais para a esquerda. Para calcularmos a tabela-esqueleto precisamos da equação (4.1.1.4) e de uma outra obtida a seguir.

Eliminando C_1 entre (4.1.1.2) e (4.1.1.3) obtemos, com o uso de (3.2.1)

$$M_O^{(k)} + \frac{1}{3} (T_O^{(k)} - M_O^{(k)}) = 1 - \frac{1}{4} 4^{-2k} C_2 - \frac{5}{16} 4^{-3k} C_3 - \dots$$

$$= T_1^{(k)} \quad (4.1.1.6)$$

Em seguida eliminamos C_2 de (4.1.1.6) e a correspondente fórmula para $T_1^{(k)}$ para encontrar

TABELA DE ROMBERG COMPLETA E TABELA-ESQUELETO

$T_0^{(0)}$

$M_0^{(0)}$

$T_0^{(1)}$

$T_1^{(0)}$

$M_0^{(1)}$

$M_1^{(0)}$

$T_0^{(2)}$

$T_1^{(1)}$

$T_2^{(0)}$

$M_0^{(2)}$

$M_1^{(1)}$

$M_2^{(0)}$

$T_0^{(3)}$

$T_1^{(2)}$

$T_2^{(1)}$

$T_3^{(0)}$

$M_0^{(3)}$

$M_1^{(2)}$

$M_2^{(1)}$

$M_3^{(0)}$

$T_0^{(4)}$

$T_1^{(3)}$

$T_2^{(2)}$

$T_3^{(1)}$

$T_4^{(0)}$

TABELA 4.1.1.1

te antecipada dos valores trapezoidais aumentam para a direita na tabela de Romberg.

ILUSTRAÇÃO DO ACÚMULO DE ERROS NO ALGORÍTMO DE ROMBERG

.62500 00000

.44444 44444

.53472 22220	.50462 96296			
.48326 53061	.49620 55933			
.50899 37640	.50041 76114	.50013 68101		
.49554 79365	.49964 21466	.49987 12501		
.50227 08500	.50002 98787	.50000 40299	301	302
.49886 74899	.49997 40077	.49999 61317		
.50056 91695	.50000 19427	.50000 00803	809	810
.49971 55990	.49999 83020	.49999 99216		
.50014 23840	.50000 01222	.50000 00009	012	014
.49992 88192	.49999 98926	.49999 99986		
.50003 56015	.50000 00074	.49999 99998	999	000

TABELA 4.1.1.2

A FUNÇÃO ROMINT é usada para integrar as funções dadas na tabela 6.2.1 (pp. 166-178).

4.1.2 - SEPARAÇÃO DA SEQUÊNCIA DE APROXIMAÇÕES CALCULADAS PELO ALGORÍTMO DE ROMBERG EM DUAS SEQUÊNCIAS [Håvie (1966)]

Esta modificação do algoritmo de Romberg é praticamente a mesma dada em (4.1.1), só que, para qualquer passo durante os cálculos, obtemos ambos os limites superior e inferior para o valor "verdadeiro" da integral (I.1) e montamos duas tabelas, isto é, uma T-tabela e uma M-tabela.

Temos:

$$I = \int_a^b f(x) dx = T_m^{(k)} + O(h^{2m+2}) \quad (4.1.2.1)$$

$$\int_a^b f(x) dx \approx M_0^{(k)} - \sum_{q=1}^{\infty} \frac{(-1)^q \gamma_q B_{2q} h_k^{2q}}{(2q)!} \left[f^{(2q-1)}(b) - f^{(2q-1)}(a) \right] \quad (4.1.2.3)$$

onde $M_0^{(k)}$ são os valores retangulares. Os coeficientes γ_q são dados por:

$$\gamma_q = 1 - \frac{1}{2^{2q-1}} \quad (4.1.2.4)$$

Comparando a fórmula de Euler-MacLaurin com a 2ª fórmula de Euler, observamos que a primeira usa os valores nos pontos extremos, enquanto que a segunda usa os valores nos pontos médios do intervalo de integração, o que é um fato importante para a discussão a seguir. Além disso, observamos que a diferença na expansão dos erros de truncamento, a menos dos sinais, é somente o fator γ_q que tende para a unidade para valores crescentes de q .

Como o algoritmo de Romberg é estritamente relacionado à fórmula de Euler-MacLaurin, então pelo uso repetido da equação (4.1.2.2) e a fórmula de recorrência (3.2.1), derivamos a seguinte generalização desta fórmula:

$$\int_a^b f(x) dx \approx T_m^{(k)} + (-1)^m \sum_{q=m+1}^{\infty} \frac{(-1)^q \delta_{q,m} B_{2q} h_k^{2q}}{(2q)!} \left[f^{(2q-1)}(b) - f^{(2q-1)}(a) \right] \quad k = 0, 1, 2, \dots \quad (4.1.2.5)$$

Para $m=0$ temos $\delta_{q,0}=1$ e a equação (4.1.2.5)

torna-se equivalente à equação (4.1.2.2).

Conforme as similaridades entre a fórmula de Euler-MacLaurin e a 2ª fórmula de Euler, podemos aplicar a técnica anterior para esta fórmula também e derivamos facil

mente a seguinte generalização:

$$\int_a^b f(x) dx \approx M_m^{(k)} - (-1)^m \sum_{q=m+1}^{\infty} \frac{(-1)^q \delta_{q,m} \gamma_q B_{2q} h_k^{2q}}{(2q)!}.$$

$$\cdot \left[f^{(2q-1)}(b) - f^{(2q-1)}(a) \right] \quad k=0,1,2,\dots \quad (4.1.2.7)$$

Usando o algoritmo modificado podemos ter algumas vantagens comparando com o algoritmo de Romberg. Primeiramente, a convergência pode ser levemente mais rápida, devido ao fator $\gamma_q < 1$ no resto. Depois, se houver problema

seguinte desigualdade é válida para a linha i genérica:

$$T_m^{(k)} - \epsilon_m^{(k-1)} < \int_a^b f(x) dx < T_m^{(k)} + \epsilon_m^{(k-1)} \quad (4.1.2.8)$$

$$m = 0, 1, 2, \dots, i; \quad k = i, i-1, i-2, \dots, 0; \quad k+m = i$$

onde

$$\epsilon_m^{(k-1)} = \left| \frac{T_m^{(k-1)} - M_m^{(k-1)}}{2} \right| \quad (4.1.2.9)$$

EXEMPLO 4.1.2.1 [Hávie (1966)]

$$\text{Para } I = \int_0^{\pi/2} \cos x \, dx = 1.0$$

os resultados estão nas tabelas 4.1.2.1 e 4.1.2.2.

APROXIMAÇÕES $T_m^{(k)}$

$\begin{matrix} m \\ k \end{matrix}$	1	2	3	4
1	.78539818			
2	.94805946	1.00227988		
3	.98711580	1.00013459	.99999156	
4	.99679517	1.00000000	.99999999	1.00000000

EXEMPLO 4.1.2.2 [Hávie (1966)]

$$I = \int_0^{\infty} e^{-x^2} dx = \frac{1}{2} \sqrt{\pi} = 0.886226925...$$

Nos cálculos tomamos a integral de $x=0$ a

$x=4.3$. O erro verdadeiro devido a isto deveria ser da ordem 10^{-9} . As tabelas 4.1.2.3 e 4.1.2.4 contêm os resultados.

APROXIMAÇÕES $T_m^{(k)}$

$\begin{matrix} m \\ k \end{matrix}$	1	2	3	4
1	2.15000002			
2	1.09613063	.74484083		
3	.88657325	.81672079	.82151279	
4	.88622693	.88611150	.89073754	.89183634

TABELA 4.1.2.3

APROXIMAÇÕES $M_m^{(k)}$

$\begin{matrix} m \\ k \end{matrix}$	1	2	3	4

tre os dois exemplos. No primeiro a convergência é mais rápida. ao longo da diagonal principal, isto é, a precisão aumenta com a ordem de aproximação. No segundo exemplo a situação é oposta, a menor ordem de aproximação tem a convergência mais rápida. Neste caso o uso de um algoritmo do tipo Romberg é completamente desnecessário. Este comportamento é facilmente detectado usando o algoritmo de Romberg modificado, já que as estimativas do erro $|T_m^{(k)} - M_m^{(k)}|$ são monotonicamente crescentes para $m=1,2,3,\dots,k$, isto é, ao longo das colunas.

O algoritmo desta seção e o descrito a seguir (seção 4.1.3) são apresentados no Apêndice , através da Função Integrator que é usada para integrar as funções dadas na tabela 6.2.1 (pp. 166-178).

4.1.3 - FORMA ALTERNATIVA DO MÉTODO DE ROMBERG BASEADA NA TRANSFORMAÇÃO COSSENO DE EULER-MACLAURIN E 2ª FÓRMULA DE EULER [Hâvie (1967)]

rivado será semelhante àquele descrito em (4.1.2), fazendo o possível durante os cálculos para estimar limites superior e inferior para o valor "verdadeiro" da integral.

Já que o princípio envolvido no algoritmo de Romberg pode ser aplicado para a transformação cosseno de Euler-MacLaurin e 2ª fórmula de Euler, uma modificação do algoritmo pode ser derivada.

Considerando a integral $Y = \int_{-1}^1 f(x) dx$ e usando a transformação cosseno $x = \cos \phi$ obtemos a fórmula de Euler-MacLaurin.

$$\int_{-1}^1 f(x) dx \approx \tilde{T}_0^{(k)} + \sum_{q=1}^{\infty} \frac{(-1)^q B_{2q} h_k^{2q}}{(2q)!} \left[f^{(2q-1)}(1) - f^{(2q-1)}(-1) \right] \quad (4.1.3.1)$$

e a 2ª fórmula de Euler

$$\int_{-1}^1 f(x) dx \approx \tilde{M}_0^{(k)} - \sum_{q=1}^{\infty} \frac{(-1)^q \gamma_{2q} B_{2q} h_k^{2q}}{(2q)!} \left[f^{(2q-1)}(1) - f^{(2q-1)}(-1) \right] \quad (4.1.3.2)$$

onde $h_k = \frac{\pi}{2^k}$, isto é, o raio de integração na variável ϕ é suposto ser dividido em 2^k partes iguais,

$$\tilde{T}_0^{(0)}$$

$$\tilde{M}_0^{(0)}$$

$$\tilde{T}_0^{(1)} \quad \tilde{T}_1^{(0)}$$

$$\tilde{M}_0^{(1)} \quad \tilde{M}_1^{(0)}$$

$$\tilde{T}_0^{(2)} \quad \tilde{T}_1^{(1)} \quad \tilde{T}_2^{(0)}$$

$$\tilde{M}_0^{(2)} \quad \tilde{M}_1^{(1)} \quad \tilde{M}_2^{(0)}$$

$$\begin{array}{cccc} \tilde{T}_0^{(3)} & \tilde{T}_1^{(2)} & \tilde{T}_2^{(1)} & \tilde{T}_3^{(0)} \\ \vdots & \vdots & \vdots & \vdots \end{array}$$

TABELA 4.1.3.1

A vantagem deste algoritmo quando trabalhamos com integrais onde os integrandos comportam-se de um modo não polinomial é naturalmente devido ao fator $\sin \phi$ que ocorrerá no integrando depois da transformação. Do ponto de vista computacional mais trabalho parece estar envolvido neste

respectivamente as equações (4.1.1.5) e (4.1.3.4). A soma

trigonométrica da equação (4.1.3.4) pode ser calculada usando

do o seguinte algoritmo:

Calculamos a sequência de números b_i definida

da por

$$b_{i+2} - 2 \cosh_k b_{i+1} + b_i = f_i$$

$$i = 2^k, 2^k-1, \dots, 1,$$

começando com $b_{2^k+1} = b_{2^k+2} = 0$

Então,

$$\tilde{M}_0^{(k)} = h_k \left[(b_1 + b_2) \sin \left(\frac{h_k}{2} \right) \right] - \frac{h_k^2}{24} [f(-1) + f(1)]$$

Para calcularmos o argumento $x_i = \cos(i - \frac{1}{2})h_k$

No cálculo destes e dos exemplos seguintes o número máximo de colunas foi tomado igual a 15 na tabela

4.1.3.2 e a precisão relativa 10^{-6} . Nos casos onde o valor

final registrado é $\tilde{T}_m^{(0)}$ a precisão relativa dada é realmen

te aquela para $\tilde{T}_{m-1}^{(1)}$, mas isto não é de importância prática.

ILUSTRAÇÃO DO USO DO ALGORÍTMO COSSENO TRANSFORMADO

EXEMPLO	ELEMENTO	VALOR OBTIDO	PRECISÃO RELATIVA	ALGORÍTMO
4.1.3.1	$T_{15}^{(0)}$	9.999996×10^{-1}	5.79×10^{-7}	Romberg (mod. 4.1.2)
	$\tilde{T}_8^{(0)}$	9.999998×10^{-1}	9.50×10^{-7}	Cosseno
4.1.3.2	$T_{16}^{(0)}$	4.247040×10^{-1}	4.06×10^{-7}	Romberg (mod. 4.1.2)
	$\tilde{T}_9^{(0)}$	4.247041×10^{-1}	3.02×10^{-7}	Cosseno

TABELA 4.1.3.2

Em comunicação privada, Erskine forneceu a Hâvie informações segundo as quais os erros para $T_m^{(k+1)}$ e $M_m^{(k+1)}$ no algoritmo de Romberg modificado 4.2 valem

$$\alpha_{k,m} \int_0^1 b_{2m+2}(2^k x) f^{(2m+2)}(x) dx$$

e

$$- (1 - \frac{2}{4^{m+1}}) \alpha_{k,m} \int_0^1 c_{2m+2}(2^k x) f^{(2m+2)}(x) dx ,$$

Dois exemplos para os quais as estimativas do erro parcialmente falham são discutidos a seguir.

EXEMPLO 4.1.3.3 [Håvie (1967)]

A usual função de Bessel pode ser definida a través de

$$J_n(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x \sin \phi - n\phi) d\phi$$

Para $n=5$, $x=10$ temos $J_5(10) = -0.2340615\dots$

EXEMPLO 4.1.3.4 [Håvie (1967)]

$$\int_{0,05}^{1/\pi} \frac{\text{sen}(1/x)}{x} dx = S_i(20) - S_i(\pi) = -0.3036954\dots$$

onde S_i é a função integral seno.

Os resultados finais dos cálculos destes dois exemplos são dados na tabela a seguir.

ILUSTRAÇÃO DO USO DO ALGORÍTMO COSSENO TRANSFORMADO

EXEMPLO	ELEMENTO	VALOR OBTIDO	PRECISÃO RELATIVA	ALGORÍTMO
4.1.3.3	$T_0^{(5)}$	-2.340615×10^{-1}	8.70×10^{-10}	Romberg (mod. 4.1.2)
	$\tilde{T}_0^{(6)}$	-2.340615×10^{-1}	4.62×10^{-7}	Cosseno
4.1.3.4	$T_3^{(6)}$	-3.036954×10^{-1}	4.59×10^{-8}	Romberg (mod. 4.1.2)
	$\tilde{T}_1^{(6)}$	-3.036954×10^{-1}	6.86×10^{-7}	Cosseno

TABELA 4.1.3.3

Para o exemplo 4.1.3.3, as tabelas 4.1.3.4 e 4.1.3.5 contêm os valores de

$$\begin{bmatrix} T_m^{(k)} \\ M_m^{(k)} \end{bmatrix} \text{ e } \begin{bmatrix} \tilde{T}_m^{(k)} \\ \tilde{M}_m^{(k)} \end{bmatrix},$$

respectivamente. Todos os valores podem ser multiplicados pelo fator 1/2 para dar as aproximações verdadeiras. Setas indicam as posições nas tabelas onde a solução verdadeira não está entre T e M e $\tilde{T}$ e $\tilde{M}$.

FALHA PARCIAL DA ESTIMATIVA DO ERRO

$i \backslash j$	1	2	3	4	5
1	0 -1.088042				
2	-.544021 -1.002481 ←	-.725361 ← -.973961 ←			
3	-.773251 ← -.655449 ←	-.849661 ← -.539772 ←	-.857948 ← -.510826 ←		
4	-.714350 -.221896	-.694717 -.077378	-.684387 -.046552	-.681632 -.039182	
5	-.468123 -.468123	-.386047 -.550199	-.365469 -.581720	-.360407 -.590215	-.359147 -.592376

TABELA 4.1.3.4

$i \backslash j$	1	2	3	4	5	6
1	0 -1.709093					
2	-.854546 ← -1.594997 ←	-.911516 ← -1.587391 ←				
3	-1.224772 ← -1.012710 ←	-1.249453 ← -0.74059 ←	-1.254817 ← -0.65227 ←			

No caso do exemplo 4.1.3.2 o algoritmo de Romberg modificado (4.1.2) falha para as posições (i,j) dadas por:

$$(2,1), (2,2), (4,1) \text{ e } (4,2)$$

e o algoritmo cosseno transformado falha para as posições

$$(2,1) \text{ e } (2,2)$$

isto é, as setas, na tabela correspondente a este exemplo, indicariam estas posições.

Para integrais com função peso com um comportamento não polinomial influenciando na convergência do algoritmo, daremos a seguir uma modificação que diminui esta influência.

Consideremos a integral

$$J = \int_a^b w(x) f(x) dx \quad (4.1.3.5)$$

onde $w(x)$ é uma função peso satisfazendo $w(x) \geq 0, x \in (a,b)$ e $\int_a^b w(x) dx = b-a$.

A função $z(x) = a + \int_a^x w(u) du$ é uma função monotonicamente crescente no intervalo (a,b) satisfazendo

$$z(a) = a, \quad z(b) = b, \quad dz = w(x) dx \quad (4.1.3.6)$$

Além disso, para $z = z(x)$, temos

$$x = x(z), \quad (4.1.3.7)$$

$$Y = \int_a^b f(x(z)) dz \quad (4.1.3.8)$$

e podemos aplicar o algoritmo de Romberg diretamente para a equação (4.1.3.8) ao invés da equação (4.1.3.5), cuja vantagem é escapar da influência de um possível comportamento não polinomial de $w(x)$ sobre a convergência do algoritmo.

Como ilustração tomamos a mesma integral do exemplo 4.1.3.2

$$\int_0^1 x^{1/3} \cos\left(\frac{\pi}{2} x\right) dx = 0.424704\dots$$

Obtemos $w(x) = \frac{4}{3} x^{1/3}$, $z(x) = x^{4/3}$, $x = z^{3/4}$.

Os resultados são dados na tabela 4.1.3.6. A modificação descrita é muito melhor que o algoritmo ordinário dado em 4.1.2 (Exemplo 4.1.3.2) e o modo cosseno é o melhor.

ILUSTRAÇÃO DE CÁLCULO DE INTEGRAL COM FUNÇÃO PESO

EXEMPLO	ELEMENTO	VALOR OBTIDO	PRECISÃO RELATIVA	ALGORÍTMO
4.1.3.2	$T_8^{(o)}$	4.247041×10^{-1}	4.80×10^{-7}	Romberg (mod. 4.1.2)
	$\tilde{T}_6^{(o)}$	4.247041×10^{-1}	6.17×10^{-8}	Cosseno

TABELA 4.1.3.6

4.1.4 - DERIVAÇÃO DE EXPRESSÕES EXPLÍCITAS PARA O TERMO DO ERRO NO ALGORÍTMO DE ROMBERG MODI- FICADO [Håvie (1969)]

O termo do erro no algoritmo de Romberg modi-
ficado pode ser, da mesma maneira que o termo do erro do al-
goritmo de Romberg, expresso em termos dos polinômios de
Bernoulli e suas funções periódicas relacionadas.

Consideramos anteriormente, uma modificação
do algoritmo de Romberg definida por:

$$M_m^{(k)} = \int_0^1 f(x) dx - \bar{R}_m^{(k)} \quad (4.1.4.1)$$

onde $\bar{R}_m^{(k)}$ representa o erro de truncamento.

Os elementos $M_0^{(k)}$, $k \geq 1$, representam aproxi-
mações retangulares, com comprimento de passo $h_k = 2^{-k}$. Para
 $m > 0$ os elementos $M_m^{(k)}$ são dados pela mesma fórmula de re-
corrência como os elementos $T_m^{(k)}$ e, além disso, temos, com
o uso de (4.1.1.5)

$$M_m^{(k)} = 2 T_m^{(k+1)} - T_m^{(k)}$$

Ström (1967), expressou o erro de trunçamen-
to como

$$\bar{R}_m^{(k)} = 4^{-k(m+1)} \int_0^1 f^{(2m+2)}(x) C_{2m+2}(2^k x) dx, \quad (4.1.4.2)$$

$$C_2(x) = \begin{cases} \frac{1}{2}x^2 & ; \quad 0 \leq x \leq \frac{1}{2} \\ \frac{1}{2}(1-x)^2 & ; \quad \frac{1}{2} \leq x \leq 1 \end{cases} \quad (4.1.4.3)$$

Portanto, vamos derivar expressões explícitas para as funções $C_{2m+2}(x)$ em termos de polinômios de Bernoulli. Usando as propriedades destas funções obtemos, da mesma maneira que para $R_m^{(k)}$ (seção 3.4.2), muitos resultados para o termo do erro $\bar{R}_m^{(k)}$, de maneira simples.

Temos que:

A representação explícita para as funções $C_{2m+2}(x)$ é

$$C_{2m+2}(x) = - \frac{(-1)^m B_{2m+2}(1/2)}{2^{m(m+1)} (2m+2)!} \left[1 - \sum_{i=0}^m c_{m,i} \frac{B_{2m+2}^*(2^i x + \frac{1}{2})}{B_{2m+2}(\frac{1}{2})} \right] \quad (4.1.4.4)$$

O termo do erro $\bar{R}_m^{(k)}$ pode ser escrito nas formas alternativas

$$\bar{R}_m^{(k)} = - 2^{-(2m+1)k} \int_0^1 f^{(2m+1)}(x) C'_{2m+2}(2^k x) dx, \quad (4.1.4.5)$$

ou

$$\bar{R}_m^{(k)} = 4^{-mk} \int_0^1 f^{(2m)}(x) C''_{2m+2}(2^k x) dx, \quad m > 0 \quad (4.1.4.6)$$

e, temos a relação

Para o cálculo numérico das funções $C_{2m+2}(x)$, da mesma maneira que para $b_{2m+2}(x)$ (seção 3.4.3), é conve-

niente aqui tratarmos com a função escalonada $\tilde{C}_{2m+2}(x)$, definida por

$$C_{2m+2}(x) = \left[1 - \left(\frac{1}{2}\right)^{2m+1} \right] \alpha_{2m+2} \tilde{C}_{2m+2}(x), \quad (4.1.4.8)$$

onde α_{2m+2} é dado por (3.4.3.2)

Da equação (4.1.4.4) obtemos

$$\tilde{C}_{2m+2}(x) = 1 - \sum_{i=0}^m c_{m,i} \frac{B_{2m+2}^*(2^i x + \frac{1}{2})}{B_{2m+2}(\frac{1}{2})} \quad (4.1.4.9)$$

Também o valor máximo de $\tilde{C}_{2m+2}(x)$ ocorre para $x = \frac{1}{2}$, e usando a equação (4.1.4.7) obtemos

$$\tilde{C}_{2m+2}\left(\frac{1}{2}\right) = 2 \frac{1 - \left(\frac{1}{4}\right)^{m+1}}{2 - \left(\frac{1}{2}\right)^{2m+1}} \cdot c_{m,0} \quad (4.1.4.10)$$

A forma limite para as funções $\tilde{C}_{2m+2}(x)$ é a mesma que para $\tilde{b}_{2m+2}(x)$, isto é,

$$\tilde{C}_{\infty}(x) = 1 - c_0 \left[\cos(2\pi x) - \frac{\cos(4\pi x)}{3} + \frac{\cos(8\pi x)}{3 \cdot 15} - \dots \right]$$

mos alguns gráficos das funções $\tilde{C}_{2m+2}(x)$; observamos (comparando com a Figura 3.4.3.1) que a convergência para a forma limite é algumas vezes mais rápida para a função $\tilde{C}_{2m+2}(x)$ que para $\tilde{b}_{2m+2}(x)$.

FIGURA 4.1.4.1

FUNÇÕES $\tilde{C}_{2m+2}(x)$ PARA ALGUNS VALORES DE m

$x \backslash m$	0	1	5	10	15
0.00	0	0	0	0	0
0.05	0.030	$8.571429 \cdot 10^{-4}$	$8.533478 \cdot 10^{-7}$	$7.434875 \cdot 10^{-7}$	$7.434164 \cdot 10^{-7}$
0.10	0.120	$1.371429 \cdot 10^{-2}$	$9.063636 \cdot 10^{-4}$	$8.924587 \cdot 10^{-4}$	$8.294470 \cdot 10^{-4}$
0.15	0.270	$6.942857 \cdot 10^{-2}$	$2.308620 \cdot 10^{-2}$	$2.229795 \cdot 10^{-2}$	$2.299787 \cdot 10^{-2}$
0.20	0.480	$2.194286 \cdot 10^{-1}$	$1.493333 \cdot 10^{-1}$	$1.491502 \cdot 10^{-1}$	$1.491500 \cdot 10^{-1}$
0.25	0.750	$5.357143 \cdot 10^{-1}$	$4.843149 \cdot 10^{-1}$	$4.841181 \cdot 10^{-1}$	$4.841179 \cdot 10^{-1}$
0.30	1.080	1.065143	1.046858	1.046754	1.046754
0.35	1.470	1.692286	1.730283	1.730342	1.730342
0.40	1.920	2.276571	2.350667	2.350850	2.350850

Damos agora a estimativa para o termo do erro

ro $\bar{R}_m^{(k)}$.

O termo do erro pode ser escrito

$$\bar{R}_m^{(k)} = \frac{-(-1)^m 4^{-k(m+1)} B_{2m+2}(\frac{1}{2})}{2^{m(m+1)} (2m+2)!} \int_0^1 f^{(2m+2)}(x) \tilde{C}_{2m+2}(2^k x) dx \quad (4.1.4.11)$$

que segue da equação (4.1.4.2) quando expressamos as funções $C_{2m+2}(x)$ em termos das funções escalonadas $\tilde{C}_{2m+2}(x)$, equação (4.1.4.9). Também observamos que os coeficientes das integrais na equação (4.1.4.11) são positivos e $\tilde{C}_{2m+2}(x)$ são positivas para $0 < x < 1$ e têm a mesma forma limite que $\tilde{b}_{2m+2}(x)$.

Obtemos da equação (4.1.4.11), usando o teorema do valor médio, a seguinte estimativa exata do erro para um integrando $f(x)$, com $2m+2$ derivadas contínuas.

$$\bar{R}_m^{(k)} = - \frac{(-1)^m 4^{-k(m+1)} B_{2m+2}(\frac{1}{2})}{2^{m(m+1)} (2m+2)!} f^{(2m+2)}(\eta) ; \quad 0 \leq \eta \leq 1 \quad (4.1.4.12)$$

4.1.5 - PROPRIEDADES PARA OS TERMOS DOS ERROS $R_m^{(k)}$ E

$\bar{R}_m^{(k)}$ [Hâvie (1969)]

TEOREMA 4.1.5.1

Para os termos dos erros $R_m^{(k)}$ e $\bar{R}_m^{(k)}$ temos:

- (i) Se $f^{(2m+2)}(x)$ tem sinal constante no intervalo $0 < x < 1$, então $R_m^{(k)}$ e $\bar{R}_m^{(k)}$ terão o mesmo sinal;
- (ii) Se $f^{(2m+1)}(x)$ é contínua para $0 \leq x \leq 1$ e, se $f^{(2m+1)}(1) \neq f^{(2m+1)}(0)$, então para valores crescentes de k , $R_m^{(k)}$ e $\bar{R}_m^{(k)}$ terão o mesmo sinal.

PROVA

- (i) Segue das formulas (3.4.4.1) e (4.1.4.11) lembrando que $B_{2m+2}(\frac{1}{2})$ e B_{2m+2} têm sinais opostos (2.1.2.8) e que $\tilde{b}_{2m+2}(x)$ e $\tilde{C}_{2m+2}(x)$ são positivas para $0 < x < 1$.
- (ii) Primeiramente reescrevemos a integral da equação (3.4.4.1),

$$\int_0^1 f^{(2m+2)}(x) \tilde{b}_{2m+2}(2^k x) dx = \frac{1}{2^k} \sum_{j=1}^{2^k} \int_0^1 f^{(2m+2)}(x_{j-1} + \frac{t}{2^k}) \tilde{b}_{2m+2}(t) dt$$

Para a integral da equação (4.1.4.11) obtemos, pelo mesmo caminho:

$$\int_0^1 f^{(2m+2)}(x) \tilde{C}_{2m+2}(2^k x) dx = \frac{1}{2^k} \sum_{j=1}^{2^k} f^{(2m+2)}(\eta_j) +$$

$$\int_0^1 f^{(2m+2)}(x) dx = f^{(2m+1)}(1) - f^{(2m+1)}(0)$$

$$x_{j-1} \leq \eta_j \leq x_j.$$

Obtemos para a diferença entre as duas integrais, usando o teorema da média, pois, $f^{(2m+2)}(x)$ é contínua em $0 \leq x \leq 1$,

$$\left| \int_0^1 f^{(2m+2)}(x) \tilde{b}_{2m+2}(2^k x) dx - \int_0^1 f^{(2m+2)}(x) \tilde{C}_{2m+2}(2^k x) dx \right| =$$

$$= \frac{1}{2^k} \sum_{j=1}^{2^k} \left| f^{(2m+2)}(\xi_j) - f^{(2m+2)}(\eta_j) \right| \leq \frac{L}{2^k} \sum_{j=1}^{2^k} \left| \xi_j - \eta_j \right| \leq \frac{L}{2^k},$$

onde L é a constante de Lipschitz, que para $f^{(2m+3)}(x)$ limitada no intervalo $0 < x < 1$ pode ser tomada independente dos índices j, k . Portanto, para $k \rightarrow \infty$, a diferença entre as duas integrais vai para zero. Isto prova (ii), visto que as duas integrais tendem para o mesmo valor limite para

O principal empecilho do erro limitado, equação (4.1.5.2), do ponto de vista computacional, é que ele pode superestimar o erro de truncamento "verdadeiro" $R_m^{(k+1)}$ relacionado ao elemento $T_m^{(k+1)}$. Isto ocorre quando $T_m^{(k+1)}$ é uma aproximação altamente melhorada para a integral, compa-

rada com os elementos $T_m^{(k)}$ e $M_m^{(k)}$.

A relação entre os elementos T e M, equação (4.1.1.5) dá

$$R_m^{(k+1)} = \frac{R_m^{(k)} - \bar{R}_m^{(k)}}{2} \quad (4.1.5.3)$$

Das equações (3.4.1.6) e (4.1.4.1) segue

$$T_m^{(k)} - M_m^{(k)} = R_m^{(k)} + \bar{R}_m^{(k)}, \text{ isto é,}$$

4.2 - INTEGRAÇÃO DE GAUSS-ROMBERG

[Lyness (1972)]

Métodos do tipo Romberg baseados em regras de quadratura gerais foram investigados por vários auto-

res. Lyness (1972) observou que o método de Romberg pode ser aplicado com aproximações iniciais obtidas de regras de Gauss. Håvie (1972), obteve a expressão para o termo do erro.

4.2.1 - ALGORITMO DE GAUSS-ROMBERG

Usamos o algoritmo de Neville-Romberg para construir a convencional T-tabela, com o propósito de "eliminar" os primeiros termos da expansão

Descrevemos o processo de integração de Romberg em termos da T-tabela, cuja primeira coluna consiste simplesmente na regra trapezoidal. As entradas nas colunas subsequentes podem ser definidas de vários modos. A definição conveniente aqui é dada pela equação (3.3.1.1) onde os coeficientes $c_{m,j}$ são tomados tais que

$$T_m^{(k)} - I = 0, \quad (4.2.1.1)$$

quando $f(x)$ é um polinômio de grau $2m+1$ ou menor.

Como vimos $c_{m,j}$ de (3.3.1.1) satisfaz à equação (3.3.1.5) e também à condição

$$\sum_{j=m}^{m+k} c_{m,j} / p_j^{2s} = 0, \quad s = 1, 2, \dots, m, \quad (4.2.1.2)$$

facilmente determinada pela substituição dos valores trapezoidais $T_0^{(k)}$ e (3.3.1.5) na expansão dada pela fórmula de Euler-MacLaurin.

$$T_m^{(k)} - I \approx O(p^{-(2m+2)}), \quad (4.2.1.5)$$

quando $p \rightarrow \infty$. É necessário cuidado em interpretar (4.2.1.5)

quando os raios p_j são substituídos por $kp_0, kp_1, kp_2, \dots$

e $k \rightarrow \infty$,

Como vimos, a T-tabela pode ser calculada recursivamente. É conveniente apresentarmos a relação de recorrência em uma forma algébrica independente desta aplicação particular.

TEOREMA 4.2.1.1

Dados $p_j, F_j, j=0,1,2,\dots$ e definindo $c_{m,j}$ por (3.3.1.5) e (4.2.1.2) e $N_m^{(k)}$ por

$$N_m^{(k)} = \sum_{j=k}^{m+k} c_{m,j} F_j, \quad (4.2.1.6)$$

então $N_m^{(k)}$ pode ser calculada recursivamente por meio do seguinte algoritmo.

interpolação e extrapolação o teorema é estritamente algébrico. F_j não precisa ser uma regra de quadratura. F_j é arbitrário.

No caso em que F_j é a regra trapezoidal, o

algoritmo de Neville fornece uma generalização da T-tabela

de Romberg. Portanto temos, com $F_j = T_0^{(j)}$:

ALGORÍTMO DE NEVILLE-ROMBERG

$$N_0^{(j)} = T_0^{(j)}, \quad j = 0, 1, \dots, K$$

Para cada $m = 1, 2, \dots, K$ (4.2.1.8)

$$T_m^{(k)} = (p_{k+m}^2 T_{m-1}^{(k+1)} - p_k^2 T_{m-1}^{(k)}) / (p_{k+m}^2 - p_k^2)$$

uma p-versão é definida como a regra obtida pela divisão do intervalo de integração em p subintervalos, de comprimento $\frac{1}{p}$, e aplicando a regra original para cada subintervalo. Especificamente,

$$G^{(p)} f = (p \times G) f = \frac{1}{p} \sum_{j=0}^{p-1} \sum_{i=1}^n w_i f\left(\frac{j+x_i}{p}\right), \quad (4.2.1.11)$$

que tem uma expansão precisamente da mesma forma que (4.2.1.3), isto é,

$$G^{(p)} f = I \approx d_2/p^2 + d_4/p^4 + \dots + d_{2m}/p^{2m} + O(p^{-2m-2}), \quad (4.2.1.12)$$

onde os coeficientes são novamente da forma $d_{2s} = e_{2s} I f^{(2s)}$, mas os valores numéricos dos e_{2s} dependem dos coeficientes w_i e x_i , em (4.2.1.10), que definem a regra G.

Se a regra G corresponde a um polinômio de grau $2t+1$ segue que

tre a expansão (4.2.1.3) e (4.2.1.14). É mais conveniente

denotarmos o j -ésimo elemento da primeira coluna desta tabe

la por G_t^j (e não G_0^j).

A G-tabela de Romberg

$$G_t^{(0)}$$

$$G_t^{(1)}$$

$$G_{t+1}^{(0)}$$

$$G_t^{(2)}$$

$$G_{t+1}^{(1)}$$

$$G_{t+2}^{(0)}$$

As constantes $\lambda_{m-t,j}$ satisfazem condições análogas a (3.3.1.5) e (4.2.1.2), isto é,

$$\sum_{j=k}^{k+m-t} \lambda_{m-t,j} = 1 \quad (4.2.1.18)$$

$$\sum_{j=k}^{k+m-t} \lambda_{m-t,j} / p_j^{2s} = 0 \quad ; \quad s = t+1, t+2, \dots, p$$

mas não necessariamente coincidem com $c_{m,j}$, a não ser para $t = 0$. Então, não podemos usar diretamente o algoritmo de Neville-Romberg para calcular os elementos da G-tabela, pois as constantes $\lambda_{m,j}$ não coincidem sempre com $c_{m,j}$. Portanto

Agora vamos obter uma expressão para $G_m^{(k)}$

($m \geq t$). Em vista de (4.2.1.14) podemos escrever

$$p_j^{2t} G^{(p_j)} f - p_j^{2t} I \approx \sum_{s=1}^{m-t} \frac{e_{2s+2t} I f^{(2s+2t)}}{p_j^{2s}} + O(p_j^{-(2m-2t+2)}) \quad (4.2.1.19)$$

Introduzindo as quantidades $c_{m,j}$ definidas

por (3.3.1.5) e (4.2.1.2), temos

$$\begin{aligned} \sum_{j=k}^{k+m-t} c_{m-t,j} p_j^{2t} G^{(p_j)} f - \sum_{j=k}^{k+m-t} c_{m-t,j} p_j^{2t} I &\approx \\ &\approx \sum_{s=1}^{m-t} e_{2s+2t} I f^{(2s+2t)} \sum_{j=k}^{k+m-t} c_{m-t,j} / p_j^{2s} + \\ &+ O(p^{-(2m-2t+2)}), \end{aligned} \quad (4.2.1.20)$$

TEOREMA 4.2.1.2

O elemento (k, m) na G-tabela de Romberg é

$$G_m^{(k)} = w_{m-t}^{(k)} / x_{m-t}^{(k)}, \quad m \geq t. \quad (4.2.1.24)$$

Este teorema é provado mostrando que $G_m^{(k)}$ da

do por (4.2.1.24) satisfaz (4.2.1.15), (4.2.1.16) e (4.2.1.17)

Agora retornamos à construção do algoritmo para calcular $G_m^{(k)}$. Para este propósito recorreremos ao teorema 4.2.1.1, que diz que qualquer quantidade da forma $\sum_{j=k}^{m+k} c_{m,j} F_j$, pode ser calculada como o (k, m) elemento da T-tabela usando o algoritmo de Neville (4.2.1.7), os elementos na primeira coluna sendo F_j . O teorema 4.2.1.2 expressa

no Apêndice . A Função Gauss integra as funções dadas na
tabela 6.2.1 (pp. 166-178).

4.2.2 - OBTENÇÃO DA EXPRESSÃO PARA O TERMO DO ERRO

[Hávie (1972)]

Vamos aqui obter o termo do erro relacionado a um método de extrapolação do tipo Romberg baseado no uso de uma fórmula de quadratura arbitrária.

Primeiramente formulamos, como um teorema, o cálculo da primeira coluna na tabela de Romberg, então faremos uma pequena discussão do termo do erro para o algoritmo de Gauss-Romberg.

$$H_j = H_{m-j} \quad , \quad a_j = 1 - a_{m-j} \quad (4.2.2.3)$$

$$\sum_{j=0}^m H_j = 1 \quad , \quad \sum_{j=0}^m H_j B_{2q}(a_j) = 0 \quad , \quad q = 1, \dots, n$$

e onde

$$z_{n,0}^{(k)} = \frac{1}{p_k^{2n+2}} \int_0^1 f^{(2n+2)}(\xi) d_{2n,2}(p_k \xi) d\xi \quad , \quad (4.2.2.4)$$

com o núcleo $d_{2n,2}(\xi)$ dado por ,

$$d_{2n,2}(\xi) = \sum_{j=0}^m \frac{H_j B_{2n+2}(a_j)}{(2n+2)!} \left[1 - \frac{B_{2n+2}^*(\xi - a_j)}{B_{2n+2}(a_j)} \right] \quad (4.2.2.5)$$

onde $B_q(x)$ é o polinômio de Bernoulli de ordem q e $B_q^*(x)$ é a função periódica relacionada.

Usando a equação (4.2.2.6) para formarmos $y_{n,0}^{(0)}$ como definido pela equação (4.2.2.2), e impondo que a fórmula de integração (4.2.2.1) deve ser exata para polinômios de ordem $2n+1$ inclusive, então as equações (4.2.2.3) -

(4.2.2.5) seguem diretamente fazendo uso das equações

(2.1.2.11), (2.1.2.12), (2.1.2.6) e $B_{2q}^*(x) = B_{2q}^*(-x)$. Isto finaliza a prova do teorema 4.2.2.1.

Considerando o conjunto de fórmulas de extrapolação obtidas usando a fórmula de quadratura do teorema 4.2.2.1 e tomando

$$p_k = 2^k, \quad k = 0, 1, 2, \dots, \quad (4.2.2.7)$$

Vamos finalmente dar uma pequena discussão do termo do erro relacionado ao uso geral do algoritmo de Gauss-Romberg como proposto por Lyness (1972).

Com o intervalo de integração básico $[0,1]$ dividido em $t_0, t_1, t_2, \dots$ intervalos iguais, construímos

a tabela de Romberg, com os elementos da primeira coluna como descrito no teorema 4.2.2.1.

TEOREMA 4.2.2.2

Seja $y_{n,m}^{(k)}$ um elemento arbitrário na tabela de Romberg, calculado usando o algoritmo geral proposto por Lyness (1972); então o erro de truncamento $z_{n,m}^{(k)}$ (ver equação (4.2.2.8)), é dado por:

PROVA

Desde que a aproximação $Y_{n,m}^{(k)}$ tem que ser exata para polinômios de ordem $2n+2m+1$, segue, integrando a equação (4.2.2.10) por partes, que temos que exigir

$$d_{k,2n,2m+2}^{(s)}(0) = d_{k,2n,2m+2}^{(s)}(1) = 0 ; \quad s=1, \dots, 2m \quad (4.2.2.13)$$

Diferenciando a equação (4.2.2.11) obtemos

$$\begin{aligned} d_{k,2n,2m+2}^{(s)}(\xi) = \\ = - \sum_{j=0}^p H_j \sum_{i=0}^m \frac{c_{m,m-i}^{(n)}}{t_{k+i}^{2n+2m-s+2}} \frac{B_{2n+2m-s+2}^*(t_{k+1}\xi - a_j)}{(2n+2m-s+2)!} \end{aligned} \quad (4.2.2.14)$$

Desde que

Usando as equações (4.2.2.4) e (4.2.2.5) re-
escrevemos a equação (4.2.2.16)

$$y_{n,m}^{(k)} = \sum_{i=0}^m c_{m,m-i}^{(n)} \int_0^1 f(\xi) d\xi +$$

$$+ \int_0^1 f^{(2n+2)}(\xi) \left[\sum_{j=0}^p H_j \sum_{i=0}^m \frac{c_{m,m-i}^{(n)}}{t_{k+i}^{2n+2}} \frac{B_{2n+2}(a_j) - B_{2n+2}^*(t_{k+i}\xi - a_j)}{(2n+2)!} \right] d\xi,$$

da qual concluimos a necessidade de exigir:

(a) a validade da equação (3.3.1.5)

(b) $d_{k,2n,2m+2}^{(2m)}(\xi) =$

$$= \sum_{j=0}^p H_j \sum_{i=0}^m \frac{c_{m,m-i}^{(n)}}{t_{k+i}^{2n+2}} \frac{B_{2n+2}(a_j) - B_{2n+2}^*(t_{k+i}\xi - a_j)}{(2n+2)!}$$

4.3 - INTEGRAÇÃO DE ROMBERG COM EXTRAPOLAÇÃO RACIONAL

[Bulirsch e Stoer (1967)]

Anteriormente vimos que o método de Romberg e suas modificações são métodos de extrapolação baseados sobre

polinômios de interpolação. Bulirsch e Stoer sugeriram que extrapolação racional poderia ser usada no lugar de extrapolação polinomial. As fórmulas de recursão são mais complicadas mas podem ser simplificadas escolhendo $\mu = \left[\frac{m}{2} \right]$ e $\nu = m - \left[\frac{m}{2} \right]$ (*)

Bulirsch e Stoer calcularam somas trapezoidais para vários valores de h , experimentaram uma função racional de h^2 para estes valores e então extrapolaram para

$$\hat{T}_m^{(i)}(h_j) = T(h_j) \quad ; \quad j = i, i+1, \dots, i+m \quad (4.3.1.2)$$

$$\text{onde } h_j = \frac{b-a}{n_j} \quad ; \quad j = 0, 1, 2, \dots$$

e $n = \{n_0, n_1, n_2, \dots\}$, é uma sequência arbitrária de nú

meros naturais satisfazendo

$$\frac{n_i}{n_{i+1}} \leq \beta < 1 \quad ; \quad i = 0, 1, 2, \dots$$

para alguma constante β .

Extrapolando agora para zero, temos então que $T_m^{(i)} = \hat{T}_m^{(i)}(0)$ são aproximações da integral (I.1).

Os valores $T_m^{(i)}$, que podem ser dispostos em uma tabela triangular, a exemplo da T-tabela.

$$\Delta_k^m = \frac{v_{k-1}^m \delta_{k-1}^m}{\left(\frac{h_{m-k}}{h_m}\right)^2 \Delta_{k-1}^{m-1} - v_{k-1}^m}$$

$$v_k^m = \frac{(h_{m-k}/h_m)^2 \Delta_{k-1}^{m-1} \delta_{k-1}^m}{\left(\frac{h_{m-k}}{h_m}\right)^2 \Delta_{k-1}^{m-1} - v_{k-1}^m}$$

(4.3.1.4)

$$T_k^{(m-k)} = \sum_{\ell=0}^k \Delta_{\ell}^m$$

onde $\delta_{k-1}^m = v_{k-1}^m - \Delta_{k-1}^m$

Se o denominador de (4.3.1.4) se anula, Δ_k^m e v_k^m são tomados como zero.

Estas fórmulas são calculadas sucessivamente para $m = 0.1.2....$

$$0 \leq \text{err} < \left| \text{eps} \cdot w \cdot \int_a^b |f(x)| dx \right| \quad \text{onde } 4 \leq w \leq 144.$$

Este é um critério devido a Rutishauser e o número w é determinado durante a execução da Função IRATEX de tal modo que o erro absoluto é da mesma ordem de magnitude que o número

$$\left| \text{eps} \cdot \int_b^a |f(x)| dx \right|$$

A Função IRATEX é usada para integrar as funções dadas na tabela 6.2.1 (pp. 166-178).

Na tabela a seguir são dados exemplos de extrapolação racional, com o uso da sequência $\eta = \{L, \frac{L}{2}, \frac{L}{3}, \frac{L}{4}, \frac{L}{5}, \frac{L}{6}, \frac{L}{7}, \frac{L}{8}, \frac{L}{9}, \frac{L}{10}\}$

4.3.2 - LIMITES ASSINTÓTICOS PARA O ERRO

De Gragg (1965) temos que o erro $T_m^{(i)} - I$ pa-

ra a integral I pode ser expresso na forma

$$T_m^{(i)} - I = h_i^2 \dots h_{i+m}^2 \sigma_{m+1}(h_i, \dots, h_{i+m}), \quad (4.3.2.1)$$

$$\text{com } \sigma_{m+1}(h_i, \dots, h_{i+m}) = (-1)^m \left[e_{m+1} + O(h_i^2) \right]$$

e

$$e_{2q} = \frac{H_0^{(q+1)}}{H_0^{(q)}}, \quad e_{2q+1} = \frac{H_1^{(q+1)}}{H_1^{(q)}},$$

onde $H_p^{(q)}$ denota o determinante de Hankel.

Nestes casos o algoritmo não trabalhará satisfatoriamente.

Formando uma combinação linear apropriada de $T_m^{(i)}$ e $T_m^{(i+1)}$ é possível derivar limites superior e inferior para a integral I.

Definindo

$$M_m^{(i)} = 2 T_m^{(i+1)} - T_m^{(i)}$$

Temos

$$\begin{aligned} M_m^{(i)} - I &= -h_i^2 \dots h_{i+m}^2 [\sigma_{m+1}(h_i, \dots, h_{i+m}) - \\ &\quad - 2 \frac{h_{i+m-1}^2}{h_i^2} \sigma_{m+1}(h_{i+1}, \dots, h_{i+m+1})] \end{aligned} \quad (4.3.2.2)$$

Isto é, se h_i é suficientemente pequeno (i suficientemente grande) e $\sigma_{m+1} \neq 0$ então para i grande

$$\frac{h_{i+m+1}^2}{h_i^2} \frac{|T_m^{(i)} - M_m^{(i)}|}{2}$$

desde que i seja suficientemente grande, isto é, desde que

$$|\sigma_{m+1}(h_i, \dots, h_{i+m})| \approx |\sigma_{m+1}(h_{i+1}, \dots, h_{i+m+1})| \quad (4.3.2.6)$$

Temos as consequências que seguem.

Se (4.3.2.6) é válida, então (4.3.2.3) superestima o erro verdadeiro por um fator

$$2^{m+1} \approx \frac{h_i^2}{h_{i+m+1}^2}.$$

Apesar de $\frac{E_m^{(i)}}{2^{m+1}}$ dar uma excelente estimativa da ordem de magnitude do erro, esta estimativa não é muito segura e portanto $E_m^{(i)}$ é escolhido como limite para o erro

4.4 - CONCLUSÕES

As modificações descritas neste capítulo tiveram o intuito de apresentar os possíveis melhoramentos e generalizações do método de Romberg e pudemos observar que:

A modificação sugerida por Krasun e Prager (1965), tem a vantagem de podermos inserir valores trapezoidais de uma dada coluna sem conhecer os valores trapezoidais ausentes em colunas mais para a esquerda, quando calculamos a "tabela-esqueleto". Também é vantajosa no que concerne ao acúmulo de erros de arredondamento e truncamento.

Na modificação sugerida por Hâvie (1966), podemos obter estimativas do erro com muito pouco cálculo adicional, através de limites inferior e superior para o valor verdadeiro da integral; além disso observamos que o método-M deve ter convergência levemente mais rápida. Entretanto, há situações em que o uso do algoritmo é completamente desnecessário (ver exemplo 4.1.2.2).

Em Hâvie (1967), observamos que o uso do algoritmo cosseno-transformado tem vantagem quando lidamos com integrais onde os integrandos se comportam de maneira não polinomial. No caso de integrais da forma $\int_a^b w(x)g(x)dx$, uma modificação na qual os efeitos indesejáveis devido à função peso $w(x)$ podem ser contornados.

Hâvie (1969), expressou os erros em $T_m^{(k)}$ e $M_m^{(k)}$ em termos de polinômios de Bernoulli e suas funções periódicas relacionadas e introduziu uma estimativa assintótica

No algoritmo de Gauss-Romberg, dado por Lyness (1972), a divisão do intervalo de integração é arbitrária e deve ser usado somente quando se requer uma sequência de raios que não é geométrica. Nestas condições, a constatação de que é um processo de integração competitivo, depende de uma investigação detalhada do erro ou talvez de resultados de experimentos numéricos. Mas podemos observar que, o método de Gauss-Romberg é uma combinação da precisão e economia de fórmulas de Gauss com a simplicidade da extrapolação iterativa.

Håvie (1972), obteve o termo do erro relacionado a um método de extrapolação do tipo Romberg, baseado no uso de uma fórmula de quadratura arbitrária e estendeu para o caso do algoritmo proposto por Lyness (1972).

Vale observar que, a construção de fórmulas de quadratura usando a expansão de uma função em termos dos polinômios de Bernoulli, dá aproximações muito gerais.

Na modificação dada por Bulirsch e Stoer (1967) é difícil verificarmos se o erro obtido

Existem outras modificações do método de Romberg que não foram aqui estudadas, por fugirem do âmbito deste trabalho. No entanto, é reconhecido que o seu estudo, bem como a sua utilização, podem oferecer apoio aos usuários que necessitem resolver problemas envolvendo integrais particulares.

CAPÍTULO 5

O MÉTODO DE ROMBERG E OUTROS NA INTEGRAÇÃO AUTOMÁTICA

Integração automática tem se tornado o pas
satempo favorito de muitos programadores e, desde 1960, dú
zias de tais métodos publicados e não publicados, têm esta-
do em toda parte.

Uma avaliação explícita dos princípios de in
tegração automática aparece no livro Numerical Integration
por Davis e Rabinowitz (1975), onde uma discussão completa

mos atenção especial a um método de extrapolação de Romberg adaptativo, que foi implementado por Boor (1971) e também uma das primeiras e, certamente uma das mais bem sucedidas rotinas automáticas deste tipo, que é a rotina adaptativa de Simpson. Esta última foi publicada em forma de algoritmo

por McKeeman (1962), com modificações dadas por Lyness (1969), que também investigou (1969a) as conseqüências da não consideração, na construção de tais rotinas, da possibilidade de erro de arredondamento ou outro erro estatístico em valores da função.

5.1 - INTEGRAÇÃO AUTOMÁTICA: CARACTERÍSTICAS GERAIS E FINALIDADES

[Davis e Rabinowitz (1975)]

Um integrador automático é um programa para

(4) um limite superior N para o número de cálculos da função.

- O programa então sai:

(a) com um valor I da integral, que é supostamente

correto dentro da tolerância, isto é,

$$\left| I - \int_a^b f(x) dx \right| \leq \epsilon \text{ ou } \frac{\left| I - \int_a^b f(x) dx \right|}{\left| \int_a^b f(x) dx \right|} \leq \epsilon$$

ou

(b) com um indicador de que o limite superior N foi alcançado, mas não a tolerância ϵ ; a saída pode incluir o "melhor" valor da integral, determinado no processo.

Em cada caso, o programa pode imprimir o er-

zer um trabalho razoável.

Se houver falha, restam ao usuário diversas alternativas:

(1) aumentar, se possível, o número de cálculos de va-

lores da função, permitido;

(2) aumentar a tolerância ϵ para o erro;

(3) subdividir o intervalo $[a,b]$ em dois ou mais subintervalos, preferencialmente de uma maneira aleatória e, aplicar o método de integração separadamente em cada subintervalo;

(4) tentar um método de integração automática diferente, talvez um com características especiais próprias para o integrando em questão e,

(5) tratar a integral analiticamente usando alguma técnica.

Podemos também pensar em integração automática como um processo por meio do qual um programa para um integrando f e dois limites a, b é convertido em um número $AI(f; a, b, \epsilon, N)$. Para um mesmo ambiente computacional e para ϵ e N fixados, podemos obter AI como um funcional operando

grandos automáticos, as quantidades n , a_k , x_k usadas na formação desta soma podem todas depender de f , a , b , ϵ e N .

Os métodos de integração automática existentes, podem ser classificados como adaptativo ou não adaptativo e iterativo ou não iterativo.

Em integração adaptativa, os pontos para os quais o integrando é calculado são tomados de maneira que depende da natureza do integrando e na integração não adaptativa os pontos são escolhidos de um modo fixo independente da natureza do integrando; o número de tais pontos, entretanto, dependerá geralmente do integrando.

Em um método iterativo, aproximações sucessivas para a integral são calculadas até que exista concordância com o limite de tolerância dado. Em um método não iterativo, informação de uma primeira aproximação é usada para gerar uma segunda aproximação, que é então tomada como uma resposta final.

Um integrador automático é geralmente escrito com uma certa classe F de integrandos em mente. A estima

5.2 - INTEGRAÇÃO DE ROMBERG: UM MÉTODO DE INTEGRAÇÃO AUTOMÁTICA

[Davis e Rabinowitz (1975)]

Como o método iterativo não adaptativo para integração automática, o algoritmo de Romberg tem provado sucesso na prática e pode ser visto (Tabela 6.2.1, pp. 166-178) que é competitivo com a regra de Simpson adaptativa.

É automático no sentido de que o número de cálculos dos valores da função depende do comportamento do integrando sobre todo o intervalo de integração. Não é adaptativo, pois avalia o integrando em um conjunto fixo de pontos. Em geral, há maior eficiência nos métodos adaptativos em termos de cálculos de valores da função. Contudo, o conjunto fixo de pontos usado na integração de Romberg é vantajoso, quando desejamos integrar uma função $f(\alpha, x)$ para diversos valores do parâmetro α . Em muitos casos, $f(\alpha, x)$ po

com m não maior que 4. Se $T_m^{(0)}$ concorda com $T_{m-1}^{(0)}$ nos n primeiros dígitos, a construção da T -tabela para $T_m^{(0)}$ é acei-

ta como a integral sobre o intervalo. Se ocorre concordân-

cia com $m=1$, h é incrementado para $1.5h$. Se ocorre com $m=4$,

h é decrementado para $0.6h$; caso contrário h permanece o

mesmo. Em todos os casos a integração então procede sobre o próximo intervalo. Se $T_4^{(0)}$ não concorda com $T_3^{(0)}$ em n dígitos, h é decrementado para $0.6h$, e é feita a integração sobre o intervalo reduzido.

Existe uma indicação de erro quando o número de intervalos sobre os quais a integração é realizada, excede $8(b-a)/h$, sugerida por Davis e Rabinowitz (1975).

A implementação deste método é dada no Apêndice através da FUNÇÃO PRAGER, que foi usada para inte-

Para estimarmos

$$I = S(o_+) = \lim_{h \rightarrow 0^+} S(h) ,$$

habitualmente calculamos $S(h)$ para vários valores positivos de h e então extrapolamos destes valores para $h = 0$. Esta quantidade combina os valores calculados de $S(h)$ de tal modo que, os primeiros $A_i h^{\gamma_i}$ dos termos do erro em (5.2.2.1) são eliminados.

Desde que, por exemplo,

$$0 = S(h) - S(2h) + \sum_{i=1}^k A_i h^{\gamma_i} (1-2^{\gamma_i}) + O(h^{\gamma_k})$$

temos, para qualquer $r \neq 1$,

$$I = S(h) + (S(h) - S(2h)) / (r-1) + \sum_{i=1}^k A_i h^{\gamma_i} (r-2^{\gamma_i}) / (r-1) + O(h^{\gamma_k}) .$$

Conseqüentemente, com a particular escolha $r = 2^{\gamma_1}$, a função

$$\bar{S}(h;r) = S(h) + (S(h) - S(2h)) / (r-1)$$

satisfaz

$$I = \bar{S}(h;r) + \sum_{i=2}^k A'_i h^{\gamma_i} + O(h^{\gamma_k}) , \quad \text{com}$$

$$A'_i = A_i (r-2^{\gamma_i}) / (r-1) .$$

Já que $\gamma_1 < \gamma_2 < \dots$, a estimativa $\bar{S}(h;r)$ é de ordem mais elevada que a estimativa $S(h)$. O princípio "ordem mais elevada é melhor", implica que $\bar{S}(h;r)$ é uma estimativa melhor

para I que $S(h)$. Uma vez que isto é aceito, prosseguimos naturalmente. Escolhendo $r_i = 2^{\gamma_i}$, $i=1, \dots, k$, calculamos

$$\begin{aligned} \bar{S}(h; r_1, \dots, r_j) &= \bar{S}(h; r_1, \dots, r_{j-1}) + \\ &+ \frac{\bar{S}(h; r_1, \dots, r_{j-1}) - \bar{S}(2h, r_1, \dots, r_{j-1})}{r_{j-1}} \end{aligned}$$

e encontramos que

$$I = \bar{S}(h; r_1, \dots, r_j) + \sum_{i=j+1}^k A_i^{(j)} h^{\gamma_i} + o(h^{\gamma_k})$$

$$\text{com } A_i^{(j)} = A_i \prod_{n=1}^j \frac{r_n - 2^{\gamma_i}}{r_n - 1}.$$

Infelizmente, esta propriedade permanece somente até usarmos extrapolação no cálculo de fato. Então torna-se penosamente claro que $\bar{S}(h, r_1)$ é uma estimativa melhor para I que $S(h)$ somente se o termo do erro $A_1 h^{\gamma_1}$ eliminado é responsável pela maior parte da discrepância entre $S(h)$ e I . Se isto não ocorrer, então o aumento nos coeficientes dos termos de ordem mais elevada pode, na verdade, tornar $\bar{S}(h, r_1)$ uma estimativa pior para I que $S(h)$. Coisas surpreendentes podem acontecer por trás de "0".

Contra tais aborrecimentos Lynch (1967) propôs o critério a seguir.

Se o termo do erro "dominante" $A_j^{(j-1)} h^{\gamma_j}$ para $\bar{S}(h; r_1, \dots, r_{j-1})$ é de fato dominante, isto é, se

$$I \approx \bar{S}(h; r_1, \dots, r_{j-1}) + A_j^{(j-1)} h^{\gamma_j}, \quad \text{para } h \leq 4h_0$$

então, para $h \leq 2h_0$,

$$\bar{S}(h; r_1, \dots, r_{j-1}) - \bar{S}(2h; r_1, \dots, r_{j-1}) \approx A_j^{(j-1)} h^{\gamma_j} (2^{\gamma_j} - 1) \quad (5.2.2.3)$$

portanto, para $h \leq h_0$

$$R_{j-1}(h) = \frac{\bar{S}(2h; r_1, \dots, r_{j-1}) - \bar{S}(4h; r_1, \dots, r_{j-1})}{\bar{S}(h; r_1, \dots, r_{j-1}) - \bar{S}(2h; r_1, \dots, r_{j-1})} \approx 2^{\gamma_j} \quad (5.2.2.4)$$

Uma pessoa cautelosa, aceitará $\bar{S}(h; r_1, \dots, r_j)$ como uma estimativa para I , que é melhor que $\bar{S}(h; r_1, \dots, r_{j-1})$, somente se

$$R_{j-1}(h) \approx 2^{\gamma_j}.$$

Além do mais, sob esta condição, aceitará (5.2.2.2) e portanto acredita que

$$\begin{aligned} \left| I - \bar{S}(h; r_1, \dots, r_j) \right| &\leq \left| \bar{S}(h; r_1, \dots, r_j) - \bar{S}(h; r_1, \dots, r_{j-1}) \right| \\ &= \left| \bar{S}(h; r_1, \dots, r_{j-1}) - \bar{S}(2h; r_1, \dots, r_{j-1}) \right| / (r_{j-1}) \end{aligned}$$

Isto é a essência da extrapolação cautelosa.

Até agora assumimos que os expoentes $\gamma_1, \dots, \gamma_k$ são conhecidos. Então o uso computacional de extrapolação cautelosa corresponde a tomar menores e menores h , até que finalmente

$$R_0(h) = \frac{S(2h) - S(4h)}{S(h) - S(2h)} \approx 2^{\gamma_1}$$

Devemos testar

$$\left| R_0(h) - 2^{\gamma_1} \right| \leq 2^{\gamma_1}/10$$

e, quando isto finalmente ocorrer, acreditamos que

$$\left| I - \bar{S}(h, 2^{\gamma_1}) \right| \leq \left| S(h) - S(2h) \right| / (2^{\gamma_1} - 1).$$

Se este limite do erro na estimativa $\bar{S}(h, 2^{\gamma_1})$ para I não for suficientemente pequeno, tomamos h menor até que

$$\left| R_1(h) - 2^{\gamma_2} \right| \leq 2^{\gamma_2}/10$$

e, quando isto finalmente ocorrer, acreditamos que a estimativa

$$\left| I - \bar{S}(h; 2^{\gamma_1}, 2^{\gamma_2}) \right| \leq \left| S(h; 2^{\gamma_1}) - S(2h, 2^{\gamma_1}) \right| / (2^{\gamma_2} - 1), \text{ etc.}$$

Se os expoentes não são anteriormente conhecidos, podemos tentar calculá-los de vários valores calculados de $S(h)$. Isto pode ser feito eficientemente usando o ϵ -algoritmo de Peter Wynn (1963), se estivermos dispostos a assumir que o 0-termo em (5.2.2.1) é, de fato, zero. Mais ingenuamente, devemos tomar menores e menores h até que $R_0(h)$ tenha convergido para um ou dois dígitos, e então usar $r = R_0(h)$ para extrapolação. Pode-se facilmente verificar que o valor extrapolado resultante $S(h; r)$ coincide com o resultado de aplicações do Δ^2 -processo de Aitken para três números $S(4h)$, $S(2h)$ e $S(h)$. Infelizmente, a validade do limite

é agora mais duvidosa, como mostra o seguinte lema.

LEMA 5.2.2.1

Suponhamos que $S(h)$ satisfaça

$$I = S(h) + Ah^\alpha + Bh^\beta ; \quad h > 0$$

onde $0 < \alpha, \beta$. Se $|R_0(h) - 2^\alpha| = C$

então

$$\left| I - \bar{S}(h; 2^\alpha) \right| = \frac{|S(h) - S(2h)|}{2^{\alpha-1}} \cdot \frac{C}{2^{\beta-1}}.$$

A dificuldade surge quando

$$\alpha \gg \beta \quad \text{e} \quad |A| \gg |B|$$

tal que o termo de maior ordem é (inicialmente) dominante.

Em tal caso, $R_0(h)$ estará relativamente mais próximo de 2^α ,

de modo que extrapolação com $r \approx 2^\alpha$ parece indicada; ainda

a existência de um termo de ordem menor (despercebido) des-

trói a estimativa (5.2.2.5). Para dar um exemplo simples to

memos

$$I = S(h) + 10^3 h^6 + h^{1/2}$$

e $h = 1/4$. Então $R_0(1/4) - 2^6 = 0.8 = C$ de modo que $R_0(1/4)$

está relativamente próximo de $2^6 = 64$. Mas $C/(2^\beta - 1) = 2$, por-

tanto o erro em $\bar{S}(1/4; 2^6)$ previsto por (5.2.2.5) (com $r = 2^6$) é somente metade do erro de fato.

Portanto, quando usamos valores de $S(h)$ para

mações disponíveis sobre os γ_i . Na verdade devemos escapar da extrapolação com $R_0(h) \approx 2^\alpha$ enquanto a expansão do erro para $S(h)$ ainda contém termos h^β com $\beta \ll \alpha$.

Para extrapolação cautelosa correspondente à aproximação da soma trapezoidal composta para I , baseada sobre os subintervalos de comprimento $h=(b-a)/2^k$, $S(h)$ coincide com $T(h)$ do capítulo 3, isto é,

$$S(h) = T(h) = T(2h)/2 + h \sum_{i=1}^{2^{k-1}} f[a+(2i-1)h]$$

uma vez que conhecemos $T(2h)$.

5.2.3 - APLICAÇÃO DO MÉTODO DE ROMBERG ADAPTATIVO E CAUTELOSO

[Boor (1971)]

Um método de extrapolação cautelosa de Romberg adaptativo foi implementado por Carl de Boor, no programa CADRE (Apêndice).

O método procura resolver o seguinte problema:

Dada uma função real f , os dois extremos do intervalo de integração a e b , e o desejado erro absoluto (ϵ_A) e relativo (ϵ_R), encontrar uma solução aproximada EST tal que

$$\left| \int_a^b f(x) dx - EST \right| \leq \max(\epsilon_A, \epsilon_R \cdot \left| \int_a^b f(x) dx \right|)$$

meio do qual o valor da integral é encontrado como a soma das estimativas para a integral de $f(x)$, sobre pequenos subintervalos apropriados do intervalo de integração dado.

Começando com o intervalo de integração $[a,b]$ como o primeiro de tais subintervalos, o método procura encontrar uma estimativa aceitável sobre o intervalo dado por extrapolação de Romberg cautelosa. Se esta procura falha, o subintervalo é dividido em dois subintervalos de comprimentos iguais, considerados agora separadamente. Por motivo de economia, valores de $f(x)$, uma vez calculados, são novamente usados em estimativas sobre alguns subintervalos aos quais pertencem.

Seja $n = b-a$, $a < b$.

Na verdade para estimar a integral (I.1) o método usa a soma trapezoidal composta. Para certas classes de integrandos, a soma trapezoidal composta exhibe um conhecido e característico comportamento de convergência que o método procura detectar e aproveita-se da extrapolação cautelosa.

O programa CADRE foi escrito para tomar distintas ações, apropriadas para algumas possibilidades do comportamento do integrando, as quais vamos descrever a seguir.

$$I = T(h) + A_1 h^2 + A_2 h^4 + \dots + A_k h^{2k} + o(h^{2k})$$

tal que a extrapolação cautelosa aplica-se com $\gamma_i = 2^i$,
 $i = 1, \dots, k$.

O programa não ocupa-se diretamente com os
números $\bar{T}(h, 2^2, \dots, 2^{2j})$.

Mais propriamente, é montada a T-tabela usual (Seção 3.2) com

$$h_k = \frac{b-a}{2^k}$$

Mas é claro que

$$(b-a) T_m^{(k)} = \bar{T}(h; 4, 16, \dots, 2^{2m}).$$

Procura-se inicialmente por uma razão

$$R_o(h) = (T_o^{(k+1)} - T_o^{(k)}) / (T_o^{(k+2)} - T_o^{(k+1)})$$

de aproximadamente $4 = 2^2$ na primeira coluna das somas trapezoidais. Uma vez encontrada tal razão, então examinamos as demais colunas da T-tabela de uma maneira cautelosa, como descrito anteriormente.

de que a aproximação da soma trapezoidal é exata para constantes, podemos assumir que $f(x)$ é da forma

$$f(x) = \begin{cases} S, & x < \xi \\ 0, & x > \xi \end{cases}$$

isto é, $f(x)$ tem um "jump" de tamanho S em $x = \xi$. Com $T(h)$ a aproximação da soma trapezoidal sobre 2^{k+1} pontos, temos $h = (b-a)/2^k$, e a diferença entre I e $T(h)$ ocorre somente em algum intervalo.

$$[a + jh, a + (j+1)h]$$

que contém o ponto de "jump" ξ . Se escrevermos ξ como uma fração binária

$$\xi = c_1 2^{-1} + c_2 2^{-2} + \dots$$

$$\text{então } j = c_1 2^{k-1} + c_2 2^{k-2} + \dots + c_k 2^0$$

Integrando $f(x)$ sobre este subintervalo obtemos o número

$$Sh(\xi 2^{k-j}) = Sh(c_{k+1} 2^{-1} + c_{k+2} 2^{-2} + \dots)$$

$$= \frac{1}{2} \text{Sh} \left[2(c_k + c_{k+1} 2^{-1} + \dots) - 2 - \right.$$

$$\left. - (c_{k+1} + c_{k+2} 2^{-1} + \dots) + 1 \right]$$

$$= \frac{1}{2} \text{Sh} (2c_k - 1)$$

$$= \begin{cases} \frac{1}{2} \text{Sh} & , \quad c_k = 1 \\ \frac{-1}{2} \text{Sh} & , \quad c_k = 0 \end{cases}$$

Consequentemente,

$$R_0(h) = \pm 2 \quad (5.2.3.1)$$

e, com $h = (b-a)/2^k$,

$$|s| = |T(h) - T(2h)| 2^{k+1} / (b-a) \quad (5.2.3.2)$$

mos mais somas trapezoidais para o intervalo $[a,b]$ se o limite do erro (5.2.3.3) é insatisfatório. Parece então melhor subdividir mais o intervalo.

C. Integrando com singularidade nos pontos extremos

Suponhamos que $f(x)$ é da forma

$$f(x) = (x-a)^\alpha g(x) \quad , \quad (\text{ou} \quad f(x) = (b-x)^\alpha g(x)) \quad (5.2.3.4)$$

onde $\alpha \in (-1, 1)$, $\alpha \neq 0$, e $g(x)$ é $(2k+1)$ vezes continuamente diferenciável. Além disso, para α negativo, assumimos que $f(a) = 0$ (ou $f(b) = 0$).

Sob estas hipóteses, Lyness e Ninham (1967) mostraram que a aproximação da soma trapezoidal $T(h)$ satisfaz

$$I = T(h) + \sum_{1 \leq i \leq [2k+1-\alpha]} C_i h^{i+\alpha} + \sum_{i=1}^k D_i h^{2i} + o(h^{(2k+1)}), \quad (5.2.3.5)$$

onde os C_i e D_i não dependem de h .

Se o expoente α é conhecido, extrapolação

Mas a entrada para este programa não contém tais dados. Consequentemente, o algoritmo deve de algum modo identificar que o integrando é da forma (5.2.3.4) e, além disso, estimar razoavelmente o expoente antes da extrapolação cautelosa poder ser usada. Isto é feito no programa como segue. De acordo com (5.2.3.5), tem-se, para um integrando da forma (5.2.3.4), que

$$R_0(h) \xrightarrow{h \rightarrow 0} 2^{1+\alpha}$$

Consequentemente, se $R_0(h)$ parece convergir para algum número entre 1 e 4 (diferente de 2), suspeita-se que o integrando tem a forma (5.2.3.4), isto é, enquanto es tima-se α (ou melhor, o número 2^α), a T-tabela de Romberg usual é construída, extrapolando com $\gamma_i = 2i$, $i=1, 2, \dots$. Isto elimina os termos h^{2i} em (5.2.3.5), deixando o termo $A_1 h^{1+\alpha}$ como o termo dominante (se h é suficientemente peque no). Consequentemente, se o termo $A_1 h^{1+\alpha}$ está presente de fato na expansão do erro, então as razões das várias colu-

características poderiam ser adicionadas - dentro de limites. Portanto, não é realmente necessário reconhecer um "jump" de descontinuidade na derivada de quarta ordem de um integrando apenas para calcular a integral corretamente.

Além disso, se o integrando tem singularidade para $a + \zeta(b-a)$ onde $0 < \zeta < 1$, então o algoritmo reconhecerá esta singularidade somente se a fração binária para ζ tem uma série de zeros ou de uns. Isto é o mesmo que dizer que o local de singularidade está muito próximo dos pontos extremos de um dos subintervalos considerados.

Finalmente, o algoritmo não está muito bem equipado para tratar com integrandos que são relativamente inadequados para os dados requerimentos de erro.

5.2.4 - TESTES E EXEMPLOS

[Boor (1971)]

(ii) singularidade nos pontos extremos

$$f(x, \alpha) = \begin{cases} 0 & , \quad x=0 \\ x^\alpha & , \quad x>0 \end{cases} \quad \text{sobre } [0,1]$$

(iii) oscilação, como mostrada por

$$f(x, \alpha) = \cos(\alpha \pi x) + 1 \quad \text{sobre } [0,1]$$

Todos os cálculos foram executados no CDC-6500, que tem mantissa de 48 bits em aritmética de ponto flutuante. Consequentemente, erro de arredondamento não foi considerado um problema. Em máquinas de palavra menor, deve-se executar a soma no cálculo de somas trapezoidais e o acúmulo dos resultados de vários subintervalos em precisão dupla.

O caso da função (i),

$f(x; \alpha) = 2^{-\alpha} / (4^{-\alpha} + x^2)$ sobre $[-1, 1]$ é relativamente desinteressante desde que CADRE realiza bem. Observe que o pico corresponde a $f(0, \alpha) = 2^\alpha$ e a curvatura no pico au-

portamento da rotina em torno de $\alpha = 24$ e que CADRE falha com $IFLAG=5^{(*)}$ para $\alpha = 31$, como mostra a tabela 5.2.4.1.

Desempenho de CADRE para $\int_{-1}^1 2^{-\alpha} / (4^{-\alpha} + x^2) dx$ com $EPS = 10^{-6}$

α	VALOR OBTIDO	ERRO FORNECIDO PELA ROTINA	NÚMERO DE CÁLCULOS DA FUNÇÃO	IFLAG
5	3.07911299234380	$7.42042225317075 \cdot 10^{-7}$	225	1
10	3.13963953972054	$1.24411487369168 \cdot 10^{-6}$	385	1
15	3.14153164191550	$1.74496546564959 \cdot 10^{-6}$	473	1
20	3.14159077325646	$2.00997290318271 \cdot 10^{-6}$	553	1
23.8	3.14159254747618	$2.91320177761051 \cdot 10^{-6}$	617	1
23.9	3.14159255835157	$2.89818134897998 \cdot 10^{-6}$	625	1
24	3.14159256150266	$2.01798724099758 \cdot 10^{-6}$	617	1
24.1	3.14159257915310	$2.32543552960969 \cdot 10^{-6}$	601	1
24.2	3.14159258658121	$2.80143476883185 \cdot 10^{-6}$	617	1
31	$2.68435456365603 \cdot 10^8$	$5.68862165520331 \cdot 10^{-7}$	349	5

TABELA 5.2.4.1

O caso de singularidade para os pontos ex-

$2^{1+\alpha}$). Em tal caso, é política do programa subdividir mais o subintervalo, embora para este integrando fosse melhor calcular mais linhas da T-tabela de Romberg para o subintervalo. O retorno de IFLAG=5 para α próximo de -1 é, entretanto, realista.

Através da tabela 5.2.4.2 (pag.138), comprovamos a análise feita por Boor, sobre o comportamento de CADRE para esta função.

O terceiro caso mostra como CADRE trabalha com a função oscilatória,

$$f(x, \alpha) = \cos(\alpha \pi x) + 1 \quad \text{sobre } [0,1]$$

O aspecto mais interessante aqui, vem da dificuldade usual com integrandos oscilatórios: Quando somente alguns valores da função são considerados (como acontece para o começo de CADRE), o integrando se assemelha às vezes com uma função bem diferente. Isto é claramente indicado

Desempenho de CADRE para $\int_0^1 x^\alpha dx$ com EPS = 10^{-6}

α	VALOR OBTIDO	ERRO FORNECIDO PELA ROTINA	NÚMERO DE CÁLCULOS DA FUNÇÃO	IFLAG
-0.99	$2.09039227157288 \cdot 10^1$	$1.24456310632108 \cdot 10^{-6}$	489	5
-0.98	$1.87196289904369 \cdot 10^1$	$1.09891799434658 \cdot 10^{-6}$	489	5
-0.97	$1.68395783278324 \cdot 10^1$	$1.83463858801636 \cdot 10^{-6}$	489	5
-0.96	$1.52161086876026 \cdot 10^1$	$2.97466971949841 \cdot 10^{-6}$	489	5
-0.95	$1.38095204322620 \cdot 10^1$	$3.55722661700142 \cdot 10^{-6}$	489	5
-0.9	9.04239973157808	$3.21655691522630 \cdot 10^{-6}$	489	5
0.0	1.0	0.0	7	1
0.5	$6.66666699013236 \cdot 10^{-1}$	$4.43387630077269 \cdot 10^{-8}$	33	2
0.6	$6.25000022660639 \cdot 10^{-1}$	$3.24540540543998 \cdot 10^{-8}$	33	2
0.7	$5.88235308497228 \cdot 10^{-1}$	$2.16102481975354 \cdot 10^{-8}$	33	2
0.8	$5.55555563432807 \cdot 10^{-1}$	$1.24868426340032 \cdot 10^{-8}$	33	2
0.9	$5.26315792623625 \cdot 10^{-1}$	$5.29970608023199 \cdot 10^{-9}$	33	2
0.95	$5.12820510674576 \cdot 10^{-1}$	$5.34648877421850 \cdot 10^{-7}$	145	5
0.96	$5.10204079966493 \cdot 10^{-1}$	$4.20203075848706 \cdot 10^{-7}$	145	5
0.97	$5.07614211985306 \cdot 10^{-1}$	$3.09589127226303 \cdot 10^{-7}$	145	5
0.98	$5.05050504266059 \cdot 10^{-1}$	$2.02733171823879 \cdot 10^{-7}$	145	5
0.99	$5.02512561697382 \cdot 10^{-1}$	$3.47187231696749 \cdot 10^{-7}$	141	5
1.0	0.5	0.0	7	1
1.1	$4.76190477653686 \cdot 10^{-1}$	$2.98676100746769 \cdot 10^{-7}$	149	5
1.2	$4.54545456776473 \cdot 10^{-1}$	$4.84872058680020 \cdot 10^{-7}$	140	5

Desempenho de CADRE para $\int_0^1 [\cos(\alpha \pi x) + 1] dx$ com EPS = 10^{-6}

α	VALOR OBTIDO	VALOR CORRETO DA INTEGRAL	ERRO FORNECIDO PELA ROTINA	NÚMERO DE CÁLCULOS DA FUNÇÃO	IFLAG
15.9	1.98363164307582	$9.93813637589874 \cdot 10^{-1}$	$1.30057226173837 \cdot 10^{-8}$	9	1
16	1.00000000484898	1.00000000013057	$1.27322166917713 \cdot 10^{-6}$	496	1
16.1	1.98363164307582	1.00610951343446	$1.30057226173837 \cdot 10^{-8}$	9	1
23.9	$9.95884388045697 \cdot 10^{-1}$	$9.95884386555333 \cdot 10^{-1}$	$9.63401280786441 \cdot 10^{-7}$	497	1
24	1.00000000287651	1.00000000013057	$1.33218816736983 \cdot 10^{-6}$	512	1
24.1	1.00408145878201	1.00408145922358	$7.00912555736769 \cdot 10^{-7}$	513	1
31.9	1.98363164307582	$9.96916515350028 \cdot 10^{-1}$	$1.30057226173837 \cdot 10^{-8}$	9	1
32	1.00000000242449	1.00000000013057	$2.05911956684049 \cdot 10^{-6}$	912	1
32.1	1.98363164307583	1.00306427315519	$1.30057226173837 \cdot 10^{-6}$	9	1
39.9	$9.97534751084445 \cdot 10^{-1}$	$9.97534757911262 \cdot 10^{-1}$	$1.28419846487255 \cdot 10^{-6}$	529	1
40	1.00000000191227	1.00000000013057	$1.64379647391941 \cdot 10^{-6}$	768	1
40.1	1.00245294887440	1.00245294686472	$1.35599695523606 \cdot 10^{-6}$	817	1
47.9	1.98363164307582	$9.97946489387324 \cdot 10^{-1}$	$1.30057226469897 \cdot 10^{-8}$	9	1
48	1.00000000457081	1.00000000013057	$1.98168982445552 \cdot 10^{-6}$	976	1
48.1	1.98363164307582	1.00204497235486	$1.30057226913986 \cdot 10^{-8}$	9	1
63.9	1.98363164307582	$9.98460670479495 \cdot 10^{-1}$	$1.30057226173837 \cdot 10^{-8}$	9	1
64	1.0	1.00000000013057	$2.84501746449940 \cdot 10^{-6}$	1664	1
64.1	1.98363164307583	1.00153452686826	$1.30057226321867 \cdot 10^{-8}$	9	1
71.9	$9.98631940454728 \cdot 10^{-1}$	$9.98631944987944 \cdot 10^{-1}$	$2.24953938576620 \cdot 10^{-6}$	1025	1
72	$9.99999993917165 \cdot 10^{-1}$	1.00000000013057	$2.11698583753722 \cdot 10^{-6}$	1040	1
72.1	1.00136425302997	1.00136426037793	$1.98854258084772 \cdot 10^{-6}$	1025	1
127.9	1.98363164307582	$9.99230937072615 \cdot 10^{-1}$	$1.30057227358075 \cdot 10^{-8}$	9	1
128	$9.99999998484694 \cdot 10^{-1}$	1.00000000013057	$3.53267812494807 \cdot 10^{-6}$	3048	1
128.1	1.98363164307583	1.00076786245280	$1.30057227210045 \cdot 10^{-8}$	9	1
255.9	1.98363164307582	$9.99615618864723 \cdot 10^{-1}$	$1.30057228098224 \cdot 10^{-8}$	9	1

colha regular dos pontos nos quais o integrando é examinado. A razão da deficiência pode, entretanto, ser diminuída dramaticamente, sem nenhum sacrifício extra de cálculo, em casos de integrandos oscilatórios,

$$\int_a^b f(x) dx \quad \text{com} \quad \int_a^c f(x) dx + \int_c^b f(x) dx$$

onde c é um número escolhido aleatoriamente entre a e b .

Podemos novamente, através da tabela 5.2.4.3 (pag. 139), comprovar a análise, dada por Boor, a respeito do comportamento de CADRE para esta função.

5.3 - UMA INTEGRAÇÃO ITERATIVA ADAPTATIVA BASEADA NA REGRA DE SIMPSON

Damos uma breve descrição da rotina de quadratura de Simpson adaptativa (ASQ) em sua forma original, Seção 5.3.1, o que proporciona embasamento apropriado, bem como uma notação matemática consistente para uso na subse

(ii) uma mudança que produz um resultado de maior precisão, mas sem aumento do número de cálculos da função.

(iii) uma mudança que produz um resultado de mesma precisão com redução no número de cálculos de função.

5.3.1 - MÉTODO DE QUADRATURA DE SIMPSON ADAPTATIVO ORIGINAL

[McKeeman (1962), (1963)]

Na sua forma original este método consiste em calcular uma aproximação AS para $I_f = \int_a^{a+h} f(x) dx$ com precisão ϵ , no sentido:

$$\left| AS - I_f \right| \leq \epsilon \int_a^{a+h} |f(x)| dx \quad (5.3.1.1)$$

Para evitar introdução de complicações desnecessárias, descrevemos o método na forma modificada: a magnitude absoluta do erro não deve exceder ϵ . Isto é

O operador $S^{(m)}[a, a+h]f(x)$ pode ser definido analogamente como o resultado de dividir o intervalo $[a, a+h]$ em m subintervalos iguais e aplicar a mesma regra S para cada um. Todas estas expressões são aproximações para I tal que quanto maior é o valor de m , melhor é a aproximação (esperançosamente) e o maior número de valores da função requerido para construir esta aproximação é $(2m+1)$. A rotina de quadratura de Simpson adaptativa (ASQ) produz um resultado da seguinte forma:

O intervalo $[a, a+h]$ é denotado como o único intervalo de nível 0. Este é dividido em 3 intervalos iguais $[a, a+\frac{h}{3}]$, $[a+\frac{h}{3}, a+\frac{2h}{3}]$ e $[a+\frac{2h}{3}, a+h]$ que são denominados intervalos de nível 1 tendo comprimento $h/3$. Alguns, mas não necessariamente todos, destes intervalos de nível 1 podem ser subdivididos em 3 intervalos de nível 2 tendo comprimento $h/9$. Procedendo desta maneira, o intervalo $[a, a+h]$ é finalmente subdividido em n subintervalos $[a_i, a_{i+1}]$, $i=0, 1, \dots, n-1$. Portanto

A subdivisão, equação (5.3.1.5) que depende dos valores da função encontrados e da aplicação de um critério baseado nestes valores, é a parte adaptativa da rotina. A intenção geral é obter intervalos pequenos (nível alto) onde a função está modificando rapidamente e intervalos grandes (nível baixo) onde a função varia somente superficialmente.

Na rotina ASQ um conjunto de erros

$$\varepsilon(r) = \varepsilon/3^r \quad (5.3.1.8)$$

é assinalado para cada nível, ε sendo um limitante para o erro (ver 5.3.2.1). Em um particular estágio i do cálculo, o intervalo $[a_i, a_{i+h}]$ está sendo considerado. Para este estágio, os números $a_0, a_1, \dots, a_{i-1}$ foram calculados e a aproximação

$$\sum_{j=1}^i S^{(3)}[a_{j-1}, a_j] f(x) \approx \int_a^{a_i} f(x) dx \quad (5.3.1.9)$$

está disponível. Os valores restantes para a última subdivisão, isto é, o valor de n e os números $a_{i+1}, a_{i+2}, \dots, a_n$, não foram calculados. (Contudo, um subconjunto destes números, juntamente com correspondentes valores da função, está

adicionais da função. A magnitude da diferença entre estas duas aproximações para $I[a_i, a_i+h]f(x)$ é comparada com $\epsilon(r)$. Se é menor que $\epsilon(r)$, isto é, se

$$S[a_i, a_i+h]f(x) - S^{(3)}[a_i, a_i+h]f(x) = \theta\epsilon(r), \quad |\theta| \leq 1$$

(5.3.1.10)

então admitimos que "este intervalo convergiu". Definimos $h_{i+1} = h$, adicionamos $S^{(3)}[a_i, a_i+h]f(x)$ à soma (5.3.1.9), e assim consideramos o próximo intervalo à direita de a_{i+1} onde

$$a_{i+1} = a_i + h_{i+1} \quad (5.3.1.11)$$

Por outro lado, se a diferença entre estas duas aproximações é maior que $\epsilon(r)$, isto é $|\theta| > 1$ em (5.3.1.10) subdividimos o intervalo $[a_i, a_i+h]$ em 3 intervalos $[a_i, a_i+h/3]$, $[a_i+h/3, a_i+2h/3]$, $[a_i+2h/3, a_i+h]$. Desde que o intervalo $[a_i, a_i+h]$ é um intervalo de nível r , cada um destes subintervalos é um intervalo de nível $r+1$. Também no cálculo de $S^{(3)}[a_i, a_i+h]$ calculamos $S[a_i, a_i+h/3]f(x)$ e as correspondentes quantidades para os outros dois subintervalos. Esta última informação é armazenada para uso futuro e passamos a tratar o intervalo $[a_i, a_i+h/3]$ da mesma maneira descrita para o intervalo $[a_i, a_i+h]$.

to obtém-se a integração da esquerda para a direita. Quando o intervalo de nível r , $[a_i, a_{i+h}]$, está sendo tratado, a aproximação para a parte anterior do intervalo (5.3.1.9) já está disponível, e informações correspondentes aos intervalos 0, 1 ou 2 para cada um dos níveis $r, r-1, \dots, 1$, correspondentes a intervalos à direita de a_{i+h} , são armazenadas para subsequente atenção.

A rotina acaba quando o intervalo $[a_{n-1}, b]$ converge. Desde que cada intervalo $[a_i, a_{i+1}]$ convergiu, podemos escrever

$$S[a_i, a_{i+1}]f(x) - S^{(3)}[a_i, a_{i+1}]f(x) = \theta_{i+1} \epsilon(r),$$

(5.3.1.12)

onde $|\theta_{i+1}| < 1$. Desde que

$$h_{i+1} = a_{i+1} - a_i = h3^{-r}$$

(5.3.1.13)

A segunda expressão do lado esquerdo de (5.3.1.15) é o resultado (5.3.1.7) retornado pela rotina baseada sobre $6n+1$ valores da função. Este difere evidentemente de um resultado similar baseado sobre $2n+1$ valores da função por uma quantidade em módulo menor que ϵ . Na ausên-

cia de qualquer evidência contrária, de um ponto de vista prático, existe uma forte suposição de que a integral verdadeira difere do resultado baseado sobre $6n+1$ valores da função por menos que ϵ .

Este método descrito em detalhe foi estabelecido na prática para ser "super cauteloso"; isto é, uma pre

5.3.2 - ANÁLISE ELEMENTAR DO ERRO PARA ROTINAS DE

QUADRATURA DO TIPO ASQ

[Lyness (1969)]

Fazemos uma análise elementar do erro para rotinas de quadratura do tipo ASQ. O propósito desta análise é justificar modificações para estas rotinas; não serão dados limites do erro.

Apresentamos aqui uma teoria válida para todas as funções $f(x)$ que, juntamente com suas cinco primeiras derivadas, são contínuas no intervalo de integração; isto é, $f(x) \in C^{(n)}[a,b]$, $n=5$. Se $n > 5$, vários termos estabelecidos como $O(h^5)$ são de fato $O(h^6)$.

Uma ferramenta útil para tratar com erro de quadratura é a fórmula da soma de Euler MacLaurin. Na sua forma convencional esta refere somente à regra trapezoidal. Contudo, existe uma fórmula similar para qualquer regra. Em particular, um caso especial ($q=5$) da fórmula apropriada (para regra de Simpson) pode ser estabelecido como segue:

Se $f(x) \in C^{(n)}[a, a+h]$, $n \geq 5$, então

$$S^{(m)}[a, a+h] f(x) - \int_a^{a+h} f(x) dx = \frac{C_4 h^4}{m^4} \int_a^{a+h} f^{(4)}(x) dx +$$

$$g_5(x) = \frac{4}{3} 2^{-5} \phi_5(2x) - \frac{1}{3} \phi_5(x)$$

$$\phi_5(x) = B_5 - B_5^*(1-x)/5!$$

Segue trivialmente de (5.3.2.1) que

$$\begin{aligned} S[a, a+h] f(x) - S^{(m)}[a, a+h] f(x) &= \frac{C_4 h^4 (m^4 - 1)}{m^4} \int_a^{a+h} f^{(4)}(x) dx \\ &+ \frac{h^5}{m^5} \int_a^{a+h} \left\{ m^5 g_5\left(\frac{x-a}{h}\right) - g_5\left(m \frac{x-a}{h}\right) \right\} f^{(5)}(x) dx \end{aligned} \quad (5.3.2.2)$$

O lado esquerdo de (5.3.2.1) é o erro que gostaríamos de limitar. O lado esquerdo de (5.3.2.2) é a diferença entre duas aproximações sucessivas, que pode ser calculada e cujo valor é usado como um critério de convergência. Os termos restantes de (5.3.2.1) e (5.3.2.2) são $O(h^5)$.

Nas rotinas adaptativas de Simpson quando o critério de convergência é aplicado, uma idéia grosseira das magnitudes das várias quantidades envolvidas nos pontos, pode ser obtida se nestas fórmulas os termos da ordem de h^6 são desprezados. Na versão descrita em 5.3.1 comparando dois casos da equação (5.3.2.1), isto é, $m=1$ e $m=3$, vemos que a segunda delas dá um resultado cerca de $1/81$ da primeira. De (5.3.2.2) vemos que a diferença entre estas a-

cluimos que quando a convergência em todos os intervalos é alcançada, a precisão obtida é $\epsilon/80$ e não ϵ , isto a um custo de aproximadamente três vezes o número de cálculos da função, necessários para obter a precisão requerida ϵ . Uma vez examinada, esta propriedade super cautelosa da rotina pode ser removida por uma modificação simples.

Modificação 1

Na modificação 1 substituímos (5.3.1.8) por

$$\epsilon(r) = 80\epsilon/3^r = (m^4 - 1) \epsilon / m^r$$

Modificação 2

A segunda modificação consiste em substituir $m=3$ por $m=2$. Nas versões dadas em 5.3.1, se convergência num intervalo não é obtida, a rotina triplica o número de cálculos da função para obter um resultado 81 vezes mais preciso. Se $m=2$, a rotina dobraria o número de cálculos da função para obter um resultado 16 vezes mais preciso. Isto parece ser obviamente preferível em intervalos nos quais o critério de convergência falha somente por uma margem pequena. Em outros casos, não parece fazer qualquer diferença profetizável.

É interessante observar que um método adaptativo baseado em bissecção ao invés de trisseção é mais fácil de implementar como um código automático.

Uma rotina baseada na rotina de Simpson

A diferença essencial é que (5.3.1.6),

(5.3.1.8), (5.3.1.13) e (5.3.1.14) são substituídas por

$$a_i - a_{i-1} = h_i = h 2^{-r}; \quad \epsilon(r) = 15 \epsilon 2^{-r} \quad (5.3.2.3)$$

que conduz à definição

$$AS[a, a+h] f(x) = \sum_{i=0}^{n-1} S^{(2)}[a_i, a_{i+1}] f(x) \quad (5.3.2.4)$$

aqui

$$AS[a, a+h] f(x) - \int_a^{a+h} f(x) dx = \frac{1}{15} \sum_{i=0}^{n-1} \theta_{i+1} \epsilon(r) + \\ + \sum_{i=0}^n \frac{h_i^5}{15} \int_{a_{i-1}}^{a_i} \left\{ \frac{1}{2} g_5 \left(2 \frac{x-a_{i-1}}{h_i} \right) - g_5 \left(\frac{x-a_{i-1}}{h_i} \right) \right\} f^5(x) dx, \quad (5.3.2.5)$$

onde

$$\theta_{i+1} \epsilon(r) = S[a_i, a_{i+1}] f(x) - S^{(2)}[a_i, a_{i+1}] f(x) \quad (5.3.2.6)$$

e

$$|\theta_i| < 1 \quad (5.3.2.7)$$

Concluindo observamos o seguinte:

- (i) A aproximação $AS[a, a+h] f(x)$ dada por (5.3.2.4) baseada sobre $4n+1$ valores da função ($m=2$) difere de uma aproximação similar baseada sobre $2n+1$ valores da função ($m=1$) por menos que 15ϵ (não é como em 5.3.1).

tegral exatamente por menos que ϵ . Este resultado não é garantido, como na verdade os termos de ordem h^6 em (5.3.2.5) podem tomar qualquer valor.

(iii) A teoria vista aqui aplica-se a integrandos $f(x) \in C^{(5)}[a, a+h]$, não assume limite superior sobre o possível nível r , e descarta o efeito de arredondamento.

Devemos também lembrar que tem-se eventualmente obtido sucesso ao integrar através de singularidades de funções. Um exemplo é

$$\int_{-a}^a |\operatorname{sen}(1/x)|^{-1/2} dx .$$

A teoria dada aqui não permite qualquer esclarecimento sobre esta questão.

Nos casos em que $f(x)$ é finita, mas suas primeiras derivadas têm um número finito de descontinuidades para pontos conhecidos, um procedimento padrão é dividir o intervalo de integração nestes pontos, em subintervalos e tratar com cada um separadamente. Se estes pontos coincidem com pontos usados para calcular a função para um primeiro nível e se $f(x)$ é contínua, não há necessidade de equili-

erro consiste do termo

$$a = \frac{1}{15} \sum \theta_{i+1} \epsilon(r) \sim O(h_i^5) \quad (5.3.2.8)$$

que é menor em magnitude que ϵ , juntamente com um termo da ordem h_i^6 . Contudo, os valores atuais dos termos indivi-

duais $\theta_{i+1} \epsilon(r)$ em (5.3.2.8) são conhecidos, pois o cálculo do lado direito de (5.3.2.6) foi executado para verificar a convergência do intervalo.

Uma quantidade insignificante de esforço adicional é requerida para tornar o valor atual do termo (5.3.2.8) disponível para o final do cálculo. Consequentemente, no lugar de retornar um resultado contendo um erro $a + O(h_i^6)$ onde $|a| < \epsilon$, já que o valor atual de a pode ser precisamente calculado, parece mais razoável ajustar o resultado apropriadamente e retornar um resultado cujo erro é simplesmente $O(h_i^6)$.

A rotina deveria retornar o resultado

$$Q_{AS}[a, a+h] f(x) = AS[a, a+h] f(x) - \frac{1}{15} \sum_{i=0}^{n-1} \theta_{i+1} \epsilon(r) \quad (5.3.2.9)$$

ção baseia-se no fato de que a aproximação de Newton - Cotes de 5-pontos é a aproximação da integração de Romberg de grau 5 baseada em duas subdivisões sucessivas. De fato, se a modificação 1 é usada para definir $\epsilon(r)$, o critério de convergência para cada intervalo é idêntico ao critério de convergência de Romberg padrão. Em outras palavras o resultado final para cada intervalo é o mesmo que poderia ter sido obtido se tivéssemos estabelecido a tabela de Romberg (T-tabela) para cada intervalo e reconhecido convergência se $|T_2^{(0)} - T_1^{(1)}| < \epsilon h$. Mas neste caso, se este critério não é satisfeito, no lugar de calcular a aproximação de grau mais elevado $T_3^{(0)}$, como seria o caso na integração de Romberg, dividimos o intervalo em duas partes.

Uma rotina tendo esta modificação retorna um resultado consideravelmente mais preciso que o ϵ requerido, mas não existe critério disponível prontamente para determinar, a priori, quanto mais preciso ele é.

Modificação 4

(Erro de arredondamento nos valores da função)

Até agora, o efeito do erro de arredondamento foi descartado. Mas, em qualquer rotina automática o e-

$$D[a,b]f(x) = \frac{12}{b-a} \left[S[a,b]f(x) - S^{(2)}[a,b]f(x) \right] \quad (5.3.2.10)$$

que comparamos com

$$E = 180\epsilon/(b-a) \quad (5.3.2.11)$$

Isto é equivalente à comparação (5.3.1.10) ou (5.3.2.6); o uso destas quantidades preferivelmente àquelas em (5.3.2.6) torna possível escapar de excessiva divisão por 2.

A expressão (5.3.2.10) pode ser expandida na forma

$$D[a,b]f(x) = \frac{1}{4} \left\{ f_0 - 4f_1 + 6f_2 - 4f_3 + f_4 \right\} \quad (5.3.2.12)$$

onde $f_i = f(a+(b-a)i/4)$.

Como o intervalo $[a,b]$ torna-se pequeno, a redução no valor de $D[a,b]f(x)$ depende do cancelamento de quantidades aproximadamente iguais. Se uma expressão tal co

Num exemplo dado em Lyness (1969a), um cálculo que deveria ter requerido cerca de 120 valores da função usou 152.997 valores da função simplesmente porque o erro de arredondamento impediu convergência. Devemos observar que $\epsilon_{r,0}$ não deve ser confundido com o parâmetro de precisão de máquina

ϵ_m . Se $f(x) = 10^{\pm 6} \cos x$ e a sub rotina função cosseno é codificada para precisão de máquina, $\epsilon_{r,0} \approx 10^{\pm 6} \epsilon_m$ é uma estimativa razoável. Na prática, o usuário pode não conhecer a precisão da função e esta precisão pode ser diferente em diferentes intervalos da variável x .

Como outro exemplo, Lyness resolveu a integral

$$I_f = \int_{-1}^1 (x^2 + 10^{-6})^{-1} dx \approx 3 \times 10^3$$

em dois computadores diferentes, com parâmetros de precisão de máquina $\epsilon_M = 10^{-11}$ e $\epsilon_M = 10^{-7}$, respectivamente. Esta integral foi resolvida com várias tolerâncias ϵ . Lyness selecionou alguns resultados (*) tabelados a seguir.

$\epsilon_M = 10^{-11}$				$\epsilon_M = 10^{-7}$		
ϵ	ϵ'	E_{act}	N	ϵ'	ϵ_{act}	N
10^{-3}	10^{-3}	-2.4×10^{-5}	1081	2.4×10^{-3}	9.1×10^{-4}	889
10^{-9}	0.0×10^{-8}	0.0×10^{-10}	10000	0.0×10^{-3}	0.0×10^{-4}	...

Observe que o pico do integrando é de magnitude 10^6 e, portanto, a precisão no cálculo da função próximo a ele é cerca de 10^{-5} ou 10^{-1} , respectivamente. Naturalmente, a máquina de palavra menor produziu um resultado menos preciso, mas com menor número de cálculos da função.

Descartando a proteção contra erro de arredondamento, para $\epsilon \leq 10^{-7}$ a rotina requereu 577.197 valores da função, mas o valor resultante foi praticamente o mesmo.

Analizamos este exemplo, com os cálculos efetuados no XT-2002 e SQUANK implementada na linguagem Pascal e obtivemos os seguintes resultados

ϵ	ϵ'	E_{act}	N
10^{-1}	$1.1 * 10^{-1}$	$-4.6 * 10^{-6}$	27.051
10^{-2}	$1.2 * 10^{-2}$	$-1.8 * 10^{-6}$	29.029
10^{-3}	$1.1 * 10^{-3}$	$-2.8 * 10^{-8}$	81.531

Em princípio, é fácil para a rotina determinar o nível de erro de arredondamento do integrando. Ela pode calcular uma tabela de diferenças-finitas e usá-la para calcular a precisão do integrando em todo o intervalo seguindo métodos padrões. Contudo, não existe necessidade de fazer isto explicitamente na rotina ASQ, uma vez que podem ser usados os teoremas seguintes.

TEOREMA 5.3.2.2

Se $f(x) \in C^{(4)}[a,b]$ e $f^{(4)}(x)$ é de sinal

constante neste intervalo, então

$$\left| D[a, (a+b)/2] f(x) \right| \leq \left| D[a, b] f(x) \right|$$

As provas destes teoremas encontram-se em [Lyness (1969), pp. 494-495].

Os teoremas podem ser utilizados como segue.

Supondo que um particular intervalo não converge, o valor $D_j = \left| D[a, b] f(x) \right| > E$. A rotina divide o intervalo em duas partes e calcula $D_{j+1} = \left| D[a, (a+b)/2] f(x) \right|$. Em vista do teorema 5.3.2.1, devemos esperar que $D_{j+1} \approx D_j/16$ sobre a média e devemos certamente esperar que $D_{j+1} < D_j$. Se encontrarmos $D_{j+1} \geq D_j$, uma das possibilidades ocorreu.

a) existe um zero de $f^{(4)}(x)$ no intervalo $[a, b]$; ou

para um valor diferente ϵ' , quando e quanto parecer apropriado. Definimos $E' = 180 \epsilon' / (b-a)$ em analogia a (5.3.2.11).

Baseados nos comentários feitos sistematizamos a modificação 4:

- (i) se um intervalo $j+1$ não convergiu, o intervalo anterior j não convergiu e $D_{j+1} \geq D_j$, então aumentamos o valor corrente da tolerância E' para $E' = D_{j+1}$;
- (ii) se um intervalo convergiu e $D_{j+1} \neq 0$, então ajustamos o valor corrente da tolerância para $E' = \max(E, D_{j+1})$.

As possibilidades (i) e (ii) estão sujeitas aos ajustamentos mencionados a seguir.

Na prática, se esta modificação é usada na situação atual a rotina reajusta a tolerância em cada flutuação estatística pequena do aparente nível do erro de arredondamento. Ela não protege contra a situação (a) acima. Portanto é necessário arranjarmos uma série de caminhos diferentes para a rotina prosseguir.

Em geral, um grande aumento inesperado no nível E' é impedido, e depois de qualquer aumento moderado, é simulado que o intervalo não convergiu. Isto ajuda a remediar uma situação na qual E' é aumentado simplesmente porque

f(4), (a) muda de sinal

so. Numa execução normal, na qual erro de arredondamento é insignificante, o custo adicional é simplesmente uma comparação singular depois de considerar cada subintervalo, isto é,

$$"E = E' ?" \text{ ou } "D_{j+1} < D_j ?"$$

conforme o intervalo anterior convergiu ou não. A resposta é invariavelmente "sim" em uma execução normal, e a rotina procede normalmente.

Se a tolerância ϵ é aumentada ou diminuída durante o cálculo, o usuário pode ser informado do efeito acumulativo. Se houve $n-1$ ajustamentos para $x = x_1, x_2, \dots, x_{n-1}$, o intervalo foi efetivamente dividido em n subintervalos $[x_{i-1}, x_i]$, $i=1, 2, \dots, n$ (onde $a = x_0$ e $b = x_n$), e uma tolerância diferente ϵ_i ($i=1, 2, \dots, n$) foi usada em cada um. É claro que para cada subintervalo a tolerância é $\epsilon_i (x_i - x_{i-1}) / (b-a)$. Portanto, a tolerância correspondente ao intervalo de entrada é

$$\epsilon_{\text{eff}} = \frac{1}{b-a} \sum_{i=1}^n \epsilon_i (x_i - x_{i-1}) \quad (5.3.2.13)$$

Para a rotina, é simples calcular ϵ_{eff} . A quantidade apropriada é simplesmente adicionada cada vez que a tolerância é alterada.

Finalmente, observamos que se a tolerância

Podemos agora completar as afirmações da modificação 4.

(iii) a rotina retorna ϵ_{eff} dado por (5.3.2.13), superposto ao parâmetro de entrada ϵ .

(iv) se a modificação 3 é incluída, o ajustamento de 5ª ordem deve ser incluído somente para seções nas quais $\epsilon' = \epsilon$.

(v) a modificação 1 deve ser aplicada somente para seções nas quais $\epsilon' = \epsilon$.

Numa rotina tendo a modificação 4, é possível para o usuário especificar $\epsilon = 0$. Neste caso, a rotina produz um "resultado possivelmente melhor" e retorna uma estimativa de sua precisão.

Observação geral

As modificações sugeridas aqui são independentes uma da outra e cada uma pode ser incorporada a qualquer rotina de Simpson adaptativa.

A Modificação 1 é uma alternativa para a modificação dada por McKeeman (1963). A regra para determinar $\epsilon(r)$ é mudada a fim de produzir um resultado menos pre-

3º grau por um do 5º grau. Portanto, ela melhora a precisão (esperançosamente) sem aumentar o trabalho para obter o resultado.

A Modificação 4, protege contra erro de arredondamento. Em casos onde este é insignificante, o efeito é produzir um resultado igualmente preciso com considerável diminuição nos cálculos da função.

Estas modificações são consideradas melhoramentos e não afetam a idéia básica subordinada à rotina de Simpson adaptativa original.

Experimentos numéricos têm mostrado que, na maior parte das vezes os resultados são aperfeiçoados em relação aos da rotina original. Consequentemente, cabe ao usuário decidir, baseado no exposto, quais destas modificações são convenientes para um particular propósito.

No Apêndice damos a Função SQUANK que é a implementação das modificações dadas aqui da rotina de Simpson adaptativa. Esta Função que foi implementada por Lyness (1970), é usada para integrar as funções dadas na tabela 6.2.1 (pp. 166-178).

5.4 - CONCLUSÕES

Existem muitos métodos de quadratura automática e mais estão sendo diariamente construídos.

A razão principal para esta proliferação é

qualquer indivíduo que constrói uma rotina automática pode encontrar alguns problemas para os quais ela é mais eficiente que uma rotina disponível, já existente. Geralmente a estrutura de rotinas de quadratura automática é suficientemente complicada para prevenir a possibilidade de comparação ou cálculo por meios analíticos e experimentos numéricos devem fazer este papel. Portanto, no capítulo seguinte comparamos os métodos discutidos neste capítulo, através de uma análise sobre os resultados numéricos obtidos com a execução das correspondentes rotinas implementadas (Apêndice).

Além dos métodos aqui discutidos, outros existem e alguns foram citados na seção 1.4, não sendo aqui discutidos e nem analisados por fugirem do escopo deste trabalho; podem entretanto, ser úteis a um usuário que necessita resolver uma integral à qual um deles se mostre adequado.

CAPÍTULO 6

COMPARAÇÃO DOS VÁRIOS MÉTODOS DE QUADRATURA AUTOMÁTICA ADAPTATIVA OU NÃO ADAPTATIVA

Acabamos de observar que a estrutura de rotinas de quadratura automática é complicada o suficiente para impedir uma comparação por meios analíticos e que experimentos numéricos têm que fazer este papel; com eles o sucesso relativo ou falha de um método pode ser normalmente medido em termos da relação entre o número de valores da função requerido (custo) e a precisão atingida.

Usando este critério, neste capítulo comparamos os vários métodos de quadratura descritos nos capítulos anteriores, aplicados a uma coleção de 38 funções integrandos. Também é feita uma comparação em termos de confiabilidade.

integração; um parâmetro indicando a precisão requerida na
integração aproximada (precisão absoluta e/ou relativa) e
parâmetros de saída; um dando o número de cálculos da fun-
ção requerido pelo programa para estimar a integral e outro
dando o erro fornecido pela rotina. Além destes parâmetros

comuns existem outros adicionais (descritos nas respectivas

listagens) em algumas rotinas que dão maiores informações
sobre os resultados obtidos.

6.2 - FUNÇÕES INTEGRANDO, TESTES E COMPARAÇÕES

Trinta e oito funções são usadas como inte-
grandos, vinte e uma delas sugeridas por Kahaner (1971) e
outras pelos autores dos métodos. Estas funções foram sele-
cionadas como sendo representativas de funções repetidamen-
te usadas na prática; são na grande maioria diferentes, em-
bora muitas tenham características em comum. Estão listadas
na Tabela 6.2.1, (pp. 166-178).

Na rotina QUAD original e na GAUSS o erro é mais rigoroso, ou melhor, é a soma de duas diferenças consecutivas do tipo recém citado da Tabela de Romberg e de Gauss-Romberg. Por esta razão, a presença de QUAD* e GAUSS* na Tabela 6.2.1 é justificável, pois achamos útil para comparação com as outras rotinas, que apresentam os erros citados anteriormente, menos rigorosos.

Como mencionado na seção 4.1.2, a Função INTEGRATOR engloba duas modificações do método de Romberg, da das por Håvie, diferenciadas através do parâmetro de entrada IOP: se $IOP = 0$, modificação 4.1.2; se $IOP = 1$, modificação 4.1.3.

Na Função PRAGER, quando o valor obtido apresenta um (*) significa que o critério de saída, sugerido por Davis e Rabinowitz (1975), foi substituído por 32767, número este que representa o máximo de subdivisões do intervalo de integração.

INTEGRATOR (IOP = 1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUAREX
66667E-01	6.666671E-01	6.666678E-01	6.665212E-01	6.666664E-01	6.66667E-01 IFLAG = 2	6 - 666667E-01	6.666666E-01
77108E-07	7.10144E-07 1.06521E-06	8.69315E-07	9.472113E-05	6.39723E-07	4.43388E-08	3 - 46021E-07	2.55289E-07
17	2120	1160	49	361	33	199	105

INTEGRATOR (IOP = 1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUAREX
00000E-01	3.999998E-01	3.999996E-01	4.000128E-01	4.000001E-01	4.000003E-01 IFLAG = 1	4 - 00000E-01	4.000001E-01
24425E-09	8.80379E-07 2.20095E-06	7.15316E-07	2.02392E-05	9.61826E-07	4.59762E-07	3 - 46021E-07	-2.53610E-07
33	224	144	13	81	65	91	29

INTEGRATOR (IOP = 1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUAREX
931472E-01	6.931472E-01	6.931472E-01	6.931472E-01	6.931472E-01	6.931472E-01 IFLAG = 1	6 - 931472E-01	6.931472E-01
16773E-07	1.14960E-07 1.65853E-07	2.74847E-08	1.12272E-06	6.58289E-07	4.58522E-07	3 - 46021E-07	-1.90402E-07
17	62	32	9	18	17	55	21

continua . .

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANX
3.798855E-01	3.798855E-01	3.798855E-01	3.798855E-01	3.798855E-01 IFLAG = 1	3.798855E-01	3 - 798855E-01
3.88570E-08 1.02286E-07	3.65959E-08	2.11882E-07	6.41920E-07	1.24939E-07	5.88235E-07	1 - 24939E-07
32	20	7	14	9	19	9

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANX
.775046E-01	7.775046E-01	7.775046E-01	7.775046E-01	7.775046E-01 IFLAG = 1	7.775046E-01	7 - 775046E-01
.11483E-07 .57852E-07	5.09611E-07	1.35402E-08	3.53032E-07	4.03874E-08	5.88235E-07	4 - 03874E-08
20	12	7	10	9	19	9

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANX
.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00 IFLAG = 1	1.000000E+00	1 - 000000E+00
.40660E-07 .40660E-07	8.60661E-08	4.03614E-06	8.75416E-08	1.93441E-09	3.46021E-07	-5 - 18589E-07
62	32	7	18	17	55	17

cont inua - -

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
8.862269E-01	8.862270E-01	8.862269E-01	8.862269E-01	8.862269E-01 IFLAG=1	8.862269E-01	8.862519E-01
5.55507E-07	5.02413E-07	4.18426E-06	1.99043E-07	2.44489E-07	3.46021E-07	1.99065E-08
6.26822E-07						
224	194	25	55	66	199	45

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
1.718282E+00	1.718282E+00	1.718282E+00	1.718282E+00	1.718282E+00 IFLAG=1	1.718282E+00	1.718282E+00
1.63497E-07		4.36966E-07		1.45377E-07	3.46021E-07	-1.45377E-07
9.51517E-08	9.08083E-08		8.00687E-09			
32	20	7	18	17	55	17

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
1.154701E+00	1.154701E+00	1.154700E+00	1.154701E+00	1.154701E+00 IFLAG=1	1.154701E+00	1.154700E+00
7.41288E-08		4.98787E-06		4.10030E-07	1.19730E-07	-8.01655E-08
6.41974E-08	6.37858E-08		8.04921E-07			
872	680	33	413	409	883	377

GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANX
0 3.376859E+00	3.376914E+00	3.376859E+00	3.376859E+00 IFLAG = 1	3 - 376859E+00	3 - 376859E+00
3.36095E-07	2.33328E-04	8.45554E-07	9.05483E-07	2 - 43701E-08	-2.84846E-07
2120	385	601	257	571	403

GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANX
1 1.402585E+01	1.402582E+01	1.402585E+01	1.402585E+01 IFLAG = 1	1 - 402585E+01	1 - 402585E+01
2.81985E-07	3.64624E-04	5.45140E-07	2.79476E-07	3 - 46021E-07	3.52528E-07
144	17	98	81	115	77

GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANX
5.770734E+00	5.769792E+00	5.770726E+00	5.770726E+00 IFLAG = 2	5 - 770726E+00	5 - 770726E+00
7.66725E-07	6.65469E-04	8.75750E-07	2.73722E-07	3 - 46021E-07	-8.07911E-08
5576	257	711	145	415	353

Continua . . .

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
18878E-10	3.45792E-09	0.000000E+00	4.10185E-10	4.10210E-10 IFLAG=1	4-960221E-10	4.099133E-10
1337E-08 1750E+01	1.79371E-07	1.04978E-10	7.50551E-08	5.28859E+12	1-000000E-06	-1.22837E-12
120	224	13	134	68	7	17

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
92308E-02	7.69230E-02	7.692308E-02	7.692308E-02	7.692308E-02 IFLAG=3	7-692308E-02	7-692315E-02
2831E-07 3080E-06	4.04346E-07	5.02210E-11	4.31328E-08	4.90899E-08	1-19730E-07	-3.51194E-07
120	80	33	50	69	223	45

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
00000E+00	1.800000E+00	1.800081E+00	1.800000E+00	1.800000E+00 IFLAG=1	1-800000E+00	1.800000E+00
3256E-07 0698E-07	2.14040E-07	2.21367E-04	9.44519E-07	8.40120E-07	3-46021E-07	-4.53627E-07
1160	872	97	287	105	247	125

Continua - -

R	INTEGRATOR (IOP - 1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
01	3.036954E-01	3.036954E-01	3.036954E-01	3.037022E-01	3.036953E-01	3.036954E-01 IFLAG - 1	3.036953E-01	3.036954E-01
8	5.03357E-07	1.12803E-07 3.71436E-07	1.12202E-07	1.53907E-04	8.19914E-07	5.27230E-07	2.03542E-07	-1.36303E-07
	129	872	680	129	310	177	559	277

R	INTEGRATOR (IOP - 1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
01	8.669730E-01	8.669730E-01	8.669731E-01	8.669793E-01	8.669730E-01	8.669730E-01 IFLAG - 1	8.669730E-01	8.669730E-01
7	1.02732E-07	5.41285E-07 6.24339E-07	4.41757E-07	1.18303E-05	3.46054E-08	5.24399E-08	3.46021E-07	-1.42572E-07
	17	80	62	19	34	17	67	29

R	INTEGRATOR (IOP - 1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
01	5.000000E-01	5.000000E-01	4.999999E-01	5.000001E-01	5.000000E-01	5.000000E-01 IFLAG - 1	5.000000E-01	5.000000E-01
1	1.27046E-07	1.21311E-07 2.42622E-07	4.77480E-07	6.84559E-06	2.43181E-07	1.21377E-07	3.46021E-07	-3.07244E-07
	33	62	32	9	26	33	79	25

Continua - - -

IRATOR -1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
19E-01	1.000001E+00	1.000001E+00	9.998373E-01	9.999996E-01	1.000000E+00 IFLAG = 2	1-000000E+00	1000000E-01
19E-07	8.50611E-07 8.50611E-07	7.07679E-07	9.96633E-05	8.60127E-07	8.25539E-08	3-46021E-07	3-46119E-07
9	5576	2888	129	480	33	235	157

IRATOR -1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
41E-01	4.247048E-01	4.247052E-01	4.246259E-01	4.247036E-01	4.247044E-01 IFLAG = 2	4-247041E-01	4-247041E-01
9E-07	9.58168E-07 2.25608E-06	5.30760E-07	4.40343E-05	7.80918E-07	1.59381E-07	3-46021E-07	4-99298E-08
	4040	2888	193	437	49	259	153

2x 0.75) , x # 0

RATOR -1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
42E-01	4.247043E-01	4.247045E-01	4.247035E-01	4.247041E-01	4.247039E-01 IFLAG = 1	4-247041E-01	4-247041E-01
0E-09	8.14592E-07 1.91802E-06	6.61862E-07	1.00109E-06	4.81501E-07	4.25404E-07	3-46021E-07	2-74074E-07
	224	144	33	85	65	91	29

CONTINUA . . .

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
4.794282E-01	4.794282E-01	4.794282E-01	4.794282E-01	4.794282E-01 IFLAG=1	4-794282E-01	4-794282E-01
1.03122E-10 1.46658E-09	6.23962E-10	1.64576E-07	9.90209E-07	4.00051E-08	3-46021E-07	-1.99560E-08
80	62	9	18	33	55	25

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
582233E+00	1.582233E+00	1.582233E+00	1.582233E+00	1.582233E+00 IFLAG=1	1-582233E+00	1-582233E+00
54539E-07 76716E-08	5.99617E-07	6.33459E-06	6.14674E-07	2.96449E-07	2-03542E-07	5-82130E-09
224	144	17	59	49	163	65

GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
998309E+00	1.998309E+00	1.946596E+00	1.999998E+00	2.000000E+00 IFLAG=2	2-000000E+00	1-999981E+00
00404E-04 50502E-04	1.81905E-04	9.71998E-03	9.22290E-07	3.76451E-07	3-46021E-07	-2-54940E-07
123080	123080	385	1846	129	1507	1153

CONTINUA ...

GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
.098637E-03	1.076757E-01	9.089649E-03	9.098637E-03 IFLAG-1	3-917186E-09	9-098646E-03
.98569E-07	3.33284E-05	6.10465E-07	2.18102E-06	5-882235E-07	9-461465E-09
1544	33	1397	1449	19	2549

GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
.000000E+00	4.998841E-01	5.000000E-01	5.000000E-01 IFLAG-1	5.000000E-01	5-000000E-01
.000000E+00	1.78832E-03	2.55573E-07	4.54248E-07	5-882235E-07	-6-13960E-09
6	385	345	91	333	169

GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
408347E-10	1.000002E+00	1.000000E+00	1.000000E+00 IFLAG-3	1.000000E+00	1-000000E+00
40835E-10	7.89171E-05	3.72685E-07	5.62950E-07	5-882235E-07	-2-88931E-08
6	193	302	128	343	213

CONTINUA - -

USS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
634E-01	4.993633E-01	4.479765E-01	4.993634E-01	4.993635E-01 IFLAG-1	4-993634E-01	4.993636E-01
87E-07	2.24342E-07	4.91726E-03	8.94885E-07	1.12156E-06	3-46021E-07	-5.15197E-07
58E-07						
52	7880	385	487	145	547	273

USS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
393E-01	1.121393E-01	1.121393E-01	1.121392E-01	1.121393E-01 IFLAG-1	1-121393E-01	1.124968E-01
9E-08	8.80530E-08	1.77732E-05	9.93003E-07	2.12291E-06	7-04296E-08	-2.59621E-08
38E-07						
10	1544	385	909	1237	3211	697

16763234E-01

SS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
63E-01	8.386763E-01	8.386765E-01	8.386763E-01	8.386763E-01 IFLAG-1	8-386763E-01	8.386763E-01
5E-07	1.78128E-07	1.10124E-04	6.85667E-07	4.40650E-07	1-19730E-07	6.01134E-08
0E-07						
	440	65	271	177	547	301

Continua - - -

GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
-9.999976E-01	-9.946251E-01	9.999993E-01	1.000000E+00 IFLAG = 2	1 - 000000E+00	-9.9999993E-01
1.13237E-06	1.84684E-03	9.77990E-07	7.88460E-07	3 - 46021E-07	5.57374E-07
123080	385	982	233	583	237

GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
1.564396E+00	1.564396E+00	1.564396E+00	1.564396E+00 IFLAG = 1	1 - 564396E+00	1.564396E+00
1.82382E-07	1.93821E-05	3.56115E-08	1.82610E-08	2 - 03542E-07	3.45647E-07
144	17	59	33	163	49

GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
3.139592E+03	3.330723E+03	3.139593E+03	3.139593E+03 IFLAG = 1	3 - 139597E+03	Para Precisão de 10-6 não obtivemos resultado com um milhão de cálculos da função (ver tabela pág. 156)
4.62423E-07	7.35843E-02	9.65062E-07	1.38073E-06	3 - 46021E-07	
30920	385	1078	865	1099	

continua . .

TABELA 6.2.1 (continuação)

$\text{INTEGRAL N}^\circ 33 \int_{-1.0}^{1.0} f(x)dx = 7.00000000E-01, f(x) = \begin{cases} 0, & x < 0.3 \\ 1, & x \geq 0.3 \end{cases}$												
ROTINA:	QUAD	QUAD*	ROMINT	INTEGRATOR (IOP = 0)	INTEGRATOR (IOP = 1)	GAUSS	GAUSS*	IRATEX	FRAGER	CADRE	SIMPSON	SQUANK
VALOR OBTIDO:	6.999699E-01	6.999699E-01	7.000048E-01	6.999699E-01	6.999597E-01	7.000370E-01	7.000370E-01	6.977555E-01	7.000015E-01	7.000000E-01 IFLAG = 1	7.000000E-01	7.018229E-01
ERRO FORNEC. PELA ROTINA:												
- ABSOLUTO :	1.50574E-04	1.09604E-05	3.49033E-05			2.06427E-04	9.75684E-05	3.19619E-03	7.96367E-07	2.38419E-07	5.88235E-07	0.00000E+00
- RELATIVO :	2.15111E-04					2.94883E-04						
Nº CÁLCULOS DA FUNÇÃO :	32769	32769	65537	32769	32769	123080	123080	385	873	125	295	33
$\text{INTEGRAL N}^\circ 34 \int_{0.0}^{1.0} f(x)dx = -4.342944819E-01, f(x) = \begin{cases} \text{Log}(x), & x > 1.E-15 \\ 0, & x \leq 1.E-15 \end{cases}$												
ROTINA:	QUAD	QUAD*	ROMINT	INTEGRATOR (IOP = 0)	INTEGRATOR (IOP = 1)	GAUSS	GAUSS*	IRATEX	FRAGER	CADRE	SIMPSON	SQUANK
VALOR OBTIDO:	-4.342418E-01	-4.342418E-01	-4.342667E-01	-4.342418E-01	-4.342944E-01	-4.342934E-01	-4.342924E-01	-4.319602E-01	-4.342937E-01	-4.342945E-01 IFLAG = 2	-4.342945E-01	-4.342939E-01
ERRO FORNEC. PELA ROTINA:												
- ABSOLUTO :	1.35736E-04	4.71109E-05	2.49544E-05			1.03530E-06	9.83592E-07	8.02072E-04	7.86576E-07	1.08400E-06	3.46021E-07	2.42064E-07
- RELATIVO :	3.12603E-04					2.38387E-06						
Nº CÁLCULOS DA FUNÇÃO :	32769	32769	65537	32769	8193	123080	61640	385	888	241	583	235
$\text{INTEGRAL N}^\circ 35 \int_0^{\pi} \frac{1}{\pi} \cos(10 \sin(x) - 5x) dx = -2.340615283E-01$												
ROTINA:	QUAD	QUAD*	ROMINT	INTEGRATOR (IOP = 0)	INTEGRATOR (IOP = 1)	GAUSS	GAUSS*	IRATEX	FRAGER	CADRE	SIMPSON	SQUANK
VALOR OBTIDO:	-2.340615E-01	-2.340615E-01	-2.340615E-01	-2.340615E-01	-2.340615E-01	-2.340615E-01	-2.340615E-01	-2.340615E-01	-2.340615E-01	-2.340615E-01 IFLAG = 1	-2.340615E-01	-2.340615E-01
ERRO FORNEC. PELA ROTINA:												
- ABSOLUTO :	2.46235E-07	2.45995E-07	2.45991E-07			1.73642E-07	1.72787E-07	1.12738E-06	8.50728E-07	6.56162E-07	7.04296E-08	-7.17227E-08
- RELATIVO :	1.05201E-06					7.41865E-07						
Nº CÁLCULOS DA FUNÇÃO :	513	257	257	33	65	536	440	65	200	177	739	269

continua .

INTEGRAL Nº 36 $\int_{1.0}^e \frac{1}{x} dx = 1.000000000E+00$

ROTINA:	QUAD	QUAD*	ROMINT	INTEGRATOR (IOP = 0)	INTEGRATOR (IOP = 1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
VALOR OBTIDO:	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000001E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00
ERRO FORNEC. PELA ROTINA :												
- ABSOLUTO :	8.01610E-08		7.97651E-08			3.26590E-07						
- RELATIVO :	8.01569E-08	7.98431E-08		7.97651E-08	1.09030E-07	3.26590E-07	2.96970E-07	2.67959E-05	4.18090E-07	2.64447E-07	3.46021E-07	-3.90570E-07
Nº CÁLCULOS DA FUNÇÃO :	65	33	33	33	33	80	62	9	26	33	79	29

INTEGRAL Nº 37 $\int_{0.0}^{\pi/2} \sin(x) dx = 1.000000000E+00$

ROTINA:	QUAD	QUAD*	ROMINT	INTEGRATOR (IOP = 0)	INTEGRATOR (IOP = 1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
VALOR OBTIDO:	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00	1.000000E+00
ERRO FORNEC. PELA ROTINA :												
- ABSOLUTO :	8.14799E-09	8.14600E-09	8.11418E-09			1.40660E-07	8.60661E-08	4.03614E-06	8.75416E-08	1.93441E-09		
- RELATIVO :	8.14962E-09			8.11418E-09	7.78394E-09	1.40660E-07					3.46021E-07	-5.18569E-07
Nº CÁLCULOS DA FUNÇÃO :	33	17	17	17	33	62	32	7	18	17	55	17

INTEGRAL Nº 38 $\int_{0.01}^{1.1} x^{-3} dx = 4.99958678E+03$

ROTINA:	QUAD	QUAD*	ROMINT	INTEGRATOR (IOP = 0)	INTEGRATOR (IOP = 1)	GAUSS	GAUSS*	IRATEX	PRAGER	CADRE	SIMPSON	SQUANK
VALOR OBTIDO:	4.999500E+03	4.999587E+03	4.999500E+03	4.999500E+03	4.999587E+03	4.999587E+03	4.999587E+03	5.000174E+03	4.999587E+03	4.999587E+03	4.999587E+03	4.999587E+03
ERRO FORNEC. PELA ROTINA :												
- ABSOLUTO :	5.01598E-05		8.60041E-08			2.25346E-04						
- RELATIVO :	1.00322E-08	2.22464E-08		1.00158E-08	9.80156E-07	4.50729E-08	4.42790E-08	3.34023E+00	9.77596E-07	3.85292E-07	5.88235E-07	3.60374E-07
Nº CÁLCULOS DA FUNÇÃO :	8193	4097	8193	4097	257	4040	2888	385	710	401	487	2225

De acordo com os resultados apresentados na Tabela 6.2.1, podemos observar que:

- O número de cálculos de valores da função diminui sensivelmente de QUAD para QUAD* e de GAUSS para GAUSS*.
- Os resultados dados por QUAD ou QUAD* parecem mostrar que para funções bem comportadas (isto é, funções contínuas sem muitas oscilações), obtém-se um resultado com erro inferior à precisão requerida. Em vários testes exemplos a precisão foi significativamente melhorada depois das primeiras sete iterações. Podemos verificar que QUAD é competitiva com SIMPSON adaptativa, em vários exemplos testes. Para integrandos da forma $f(x)g(\alpha, x)$, QUAD é em geral mais eficiente.
- Em geral ROMINT e INTEGRATOR (IOP = 0) não apresentam melhorias em relação à QUAD*. INTEGRATOR (IOP = 1), no entanto, apresenta melhor desempenho em vários exemplos.
- A forma sugerida por Håvie para integrais com função peso com comportamento não polinomial, realmente diminui a influência deste na convergência de um método.
- A Função GAUSS em geral não apresenta vantagem em relação a QUAD, portanto a idéia de Lyness em combinar economia e precisão de regra de Gauss com a simplicidade da extrapolação iterativa não resultou em um método eficiente. Talvez fosse necessário investigar melhor a sequência arbitrária de raios m_j , ou efetuar algumas modificações na rotina.
- IRATEX regularmente não produz bons resultados quando a função integrando não é pelo menos 3 a 4 vezes diferenciá

vel. Em geral, para estes casos, após poucos cálculos da função sai com um resultado sem atingir a precisão requerida. O erro fornecido por esta rotina não dá, realmente, a idéia do resultado obtido. Em muitos casos este erro não atinge a precisão requerida, mas o resultado está dentro dos limites desejáveis.

- PRAGER, que é uma combinação de integração de Romberg com método adaptativo, apresenta melhorias com relação a QUAD e alguns métodos não adaptativos, mas é menos eficiente que os métodos adaptativos, há um número razoável de casos onde é melhor que SIMPSON adaptativa.
- CADRE em geral é superior para funções com singularidades e está competindo com as rotinas não adaptativas e adaptativas, exceto para algumas funções suaves, quando dá informação errada.
- SIMPSON adaptativa em alguns casos é melhor que as não adaptativas em outros menos eficiente; é em geral menos eficiente que as outras adaptativas. Em vários casos SIMPSON necessita mais iterações que QUAD, em outros QUAD foi mais vagarosa.
- SQUANK é em geral competitiva, em alguns casos é mais econômica que as outras rotinas, em outros é menos. Embora não esperado, deu bons resultados na maioria das funções que não têm as quatro primeiras derivadas contínuas. Segundo seu autor, SQUANK é eficiente para integrandos com as propriedades:
 - . $f(x)$ e suas primeiras quatro derivadas são contínuas;
 - . $f(x)$ não tem oscilações altas.

- Em integrais da forma $\int_{-1}^1 (x^2+p^2)^{-1}dx$, ou seja, se o integrando tem pico, uma regra adaptativa é mais eficiente.
- Em alguns casos as rotinas não adaptativas são preferíveis às adaptativas com relação ao número de cálculos da função, ou seja, são mais econômicas e em outros ocorre o contrário, dependendo diretamente do tipo do integrando.

Um resumo da Tabela 6.2.1 é dado através da Tabela 6.2.2, onde NF representa o número de cálculos de valores da função. Para cada rotina estão os dados comparativos a seguir:

- NF mínimo: quantas, das 39 integrais, cada rotina calculou (dentro da tolerância especificada 10^{-6}) com menor número de avaliações do integrando que qualquer outra.
- NF máximo: quantas, das 39 integrais, cada rotina calculou (dentro da tolerância requerida), com maior número de cálculos da função que qualquer outra.
- Funções integradas com a tolerância requerida: quantas, das 39 integrais, foram integradas com pelo menos a tolerância requerida (10^{-6}).
- Funções integradas incorretamente dentro da tolerância requerida: quantas, das 39 integrais, cada rotina forneceu um resultado errado.

ROTINA Nº DE INTEGRAIS	QUAD	QUAD*	ROMINT	INTEGRATOR (IOP = 0)	INTEGRATOR (IOP = 1)	GAUSS	GAUSS*			IRATEX	PRAGER	CADRE	SIMPSON	SQUARK
							FUNÇÃO Nº 25	Nº CÁLC. 12	Nº CÁLC. 6					
COM Nº MÍNIMO	--	4	4	7	11	--	--	5	2	15	1	7		
COM Nº MÁXIMO	15	--	3	1	1	13	1	--	2	--	6	--		
OBTIDAS CORRETAMENTE (COM TOLERÂNCIA 10 ⁻⁶)	29	29	28	26	33	30	27	6	34	34	35	32		
OBTIDAS INCORRETAMENTE	--	--	--	FUNÇÃO Nº 13	--	FUNÇÃO Nº 25	Nº CÁLC. 12	FUNÇÃO Nº 25 26	Nº CÁLC. 6	FUNÇÃO Nº 24	Nº CÁLC. 33	--	FUNÇÃO Nº 24	Nº CÁLC. 19

TABELA 6.2.2

OBSERVAÇÃO: Em todos os casos só aceitamos resultados para os quais o erro fornecido pela rotina satisfazia à tolerância requerida (10^{-6}). Lembramos que, em certos casos, IRATEX fornece erro $>10^{-6}$ e resultado com 6 ou mais dígitos corretos e em outros ocorre o inverso.

Quanto à economia, verificamos que a rotina CADRE, mesmo sendo adaptativa e por natureza mais dispendiosa, no nosso caso, evidenciou-se como a mais econômica, ou melhor, das 39 integrais testadas, executou 15 com o mínimo de cálculos do integrando. Verificamos, ainda, que as mais dispendiosas são as não adaptativas QUAD e GAUSS, devido ao tipo de erro que elas utilizam, o que é claramente evidenciado pela comparação com QUAD* e GAUSS*.

Dentre as não adaptativas destaca-se INTEGRATOR (IOP=1); entre 39 funções, executou 11 com menor número de cálculos da função integrando. Como adaptativa mais dispendiosa temos SIMPSON.

Existe um equilíbrio entre todas as rotinas quanto à precisão do resultado obtido, exceto IRATEX, que, segundo seu autor, é eficiente somente para funções integrando três a quatro vezes diferenciáveis.

Poucas foram as funções integradas erradamente e também, poucas as rotinas que o fizeram. Isto se deve ao fato destas rotinas não serem apropriadas para estas funções ou vice-versa.

A Tabela 6.2.3, a seguir, demonstra o desempenho das rotinas quanto à confiança sobre o resultado obtido para a tolerância requerida.

R O T I N A	PORCENTAGEM DE CONFIANÇA PARA TOLERÂNCIA DE 10^{-6}
QUAD	74,36%
QUAD*	74,36%
ROMINT	71,79%
INTEGRATOR (IOP = 0)	66,67%
INTEGRATOR (IOP = 1)	84,62%
GAUSS	76,92%
GAUSS*	69,23%
IRATEX	15,38%
PRAGER	87,18%
CADRE	87,18%
SIMPSON	89,74%
SQUANK	82,05%

TABELA 6.2.3.

Todas as rotinas testadas apresentaram uma boa porcentagem de confiabilidade, exceto IRATEX pois, como já observamos, o erro fornecido por ela não dá uma idéia precisa do resultado obtido. Dentre as adaptativas salienta-se a rotina SIMPSON, embora CADRE, PRAGER e SQUANK sejam satisfatoriamente confiáveis. Dentre as não adaptativas salienta-se INTEGRATOR (IOP = 1), que se sobrepõe à adaptativa SQUANK, aproximando-se fortemente de CADRE e PRAGER.

6.3 - CONCLUSÕES

Toda rotina construída pelo ser humano é falível em algum sentido ou contexto. Portanto, rotinas de quadratura automática também são passíveis de falha. É, pois, importante que uma rotina automática possua um indicador que informe ao usuário se o resultado obtido é impreciso (caso de CADRE), dê uma estimativa do erro cometido no resultado obtido (apresentado por todas). Também é importante que uma rotina automática tenha uma estimativa do nível de ruído ("noise") em valores da função para ser usada em fugas de situações de erro de arredondamento (caso de SQUANK).

Qualquer decisão na escolha de uma rotina automática deveria ser devidamente influenciada pela flexibilidade e confiança adicional resultante de sua competência para retornar indicações de falha. Existe um custo x confiança. Qualquer rotina automática pode ser alterada para tornar-se significativamente mais confiável se, ao mesmo tempo estivermos preparados para torná-la mais cara. Portanto, a escolha de uma rotina automática, quanto ao custo ou confiança, dependerá da predisposição do usuário.

CAPÍTULO 7

CONCLUSÕES FINAIS

Quadratura numérica é requerida como um componente em grande escala em muitos projetos científicos. Frequentemente os cientistas enfrentam uma escolha: recorrem ou a uma das rotinas de quadratura automática ou a uma das rotinas tipo padrão, baseadas diretamente em fórmulas tradicionais (procedimentos clássicos).

O uso de métodos de integração automática tem sido um atrativo para o usuário; ele necessita somente preparar um programa para o integrando e especificar os limites de integração e a tolerância.

Os atributos desejáveis geralmente aceitos, de uma rotina de quadratura, são confiança e economia simultâneas. Gostaríamos que ela fornecesse sempre um resultado correto (dentro dos limites da tolerância) e que fizesse isso com pouco esforço ou custo, tanto quanto possível. Estes dois atributos desejados são reciprocamente exclusivos: uma rotina mais confiante implica em menor economia.

Dentre as rotinas automáticas estão as roti-

nas adaptativas, as quais, em regiões onde o integrando parece ser mais difícil, utilizam mais valores da função que em regiões onde ele é aparentemente mais fácil. São elementos-chaves de uma rotina de quadratura adaptativa a estraté

gia de partição do intervalo; a quadratura local estimada e

estimativa do erro local. Contudo, o sofisticado trabalho associado a este tipo de rotina pode tornar partes do programa mais dispendiosas que o necessário. Portanto, enquanto em muitos casos o uso de uma rotina adaptativa pode salvar significativo tempo humano, em muitos outros seu uso pode ser um desastre. Mas, se o usuário é confrontado com uma integral isolada, e ele pode sacrificar economia por segurança, uma rotina adaptativa parece ser a melhor escolha.

Se vasta quantidade de computação é requerida, a integração automática (adaptativa ou não) pode tornar-se dispendiosa e análises preliminares conduzindo ao uso de procedimentos clássicos podem resultar em compensação considerável.

Uma das motivações para este trabalho é proporcionar espaço para crescimento. Não acreditamos que nossa escolha de funções seja a mais perfeita, nem acreditamos que as melhores rotinas foram as escolhidas. Se escolhessemos outras funções e outras rotinas poderíamos chegar também a outras conclusões. Isto não necessariamente atrasa o desenvolvimento do processo numérico, meramente implica que ele é um processo dinâmico e que conclusões correntes são passíveis de modificações ou alterações.

APENDICE

Neste apendice, são apresentados os algoritmos QUAD, ROMINT, INTEGRATOR, GAUSS, IRATEX, PRAGER, CADRE, SIMPSON e SQUANK, escritos como subprogramas Funcoes, em linguagem Pascal. Um resumo-descrição de cada função é dado para possibilitar ao leitor um melhor entendimento, assim como facilitar seu uso em casos particulares. A integral definida (I.1) aqui referenciada, é a citada na Introdução do Trabalho.


```

begin
  err:=1;
  era:=1;
  nf:=0;

  h:=b-a;

  fcna:=fun(a);
  fcnb:=fun(b);
  nf:=nf+2;

  t:=h*(fcna+fcnb)/2;
  nx:=1;
  n:=0;
  flag:=false;
  repeat
    n:=n+1;
    h:=h/2;
    sum:=0;
    scorr:=0;
    for i:=1 to nx do
      begin
        xi:=2*i-1;
        ax:=a+xi*h;
        fcncxi:=fun(ax);
        nf:=nf+1;
        fcncxi:=fcncxi+scorr;
        ss:=sum+fcncxi;
        scorr:=(sum-ss)+fcncxi;
        sum:=ss;
      end;
    t:=t/2+h*sum;
    tabs:=abs(t);
    q[n]:=2*(t+h*sum)/3;
    if n>=2 then
      begin
        f:=4;
        for j:=2 to n do

```

```

if n>=3 then
begin
  if n>=min then
    begin
      x:=abs(q[1]-qx2)+abs(qx2-qx1);
      if tabs<>0 then
        begin
          err:=x/tabs;
          era:=x;
          end;
        end;
      end;
      qx1:=qx2;
      end;
      qx2:=q[1];
      nx:=nx*2;
      flag:=(err<=(eta+0.14901161e-7)) or (era<=eps);
      until (n=15) or flag;
      quad:=q[1];
      end;

```

(*

A.2 - FUNCAO ROMINT

*)

(* ROMINT calcula o valor aproximado da integral (I.1) e um limite de *)

(* erro ERR para a aproximacao obtida, isto e, $|APROX - I| \leq ERR$. O inte- *)

(* grando F(x) deve ser dado como um subprograma FUNCAO FUN(X). *)

(* A aproximacao e obtida de uma forma modificada da quadratura de Rom- *)

(* berg que e menos sensitiva ao acumulo de erros de arredondamento. O nume- *)

(* ro maximo de extrapolacoes e 15 (MAXE=15). Os valores retangulares sao *)

(* calculados usando uma tecnica sugerida por Rutishauser (1967) que reduz o *)

(* efeito de erros de arredondamentos. A funcao ROMINT e deixada quando *)

(* $|T_m^{(0)} - M_m^{(0)}| / 2 \leq EPS$, onde EPS e a precisao desejada, ou quando o numero *)

(* maximo de extrapolacoes permitido foi alcancado. Esta rotina pode ser u- *)

(* sada para estimar a integral (I.1), com F(x) tres ou quatro vezes dife- *)

(* renciavel (pelo menos) e nao periodica com periodo b-a. Esta funcao e ba- *)

(* seada na escrita por Fairweather (1969), em FORTRAN. *)

(*

Parametros de Entrada

*)

(* a,b - extremos do intervalo de integracao *)

(* eps - tolerancia para o erro absoluto *)

(*

Parametros de Saida

*)

(* err - erro absoluto no valor obtido *)

(* n - numero de calculos da funcao requerido para se obter o valor aproxi- *)

(* mado *)

(* mfin = m, (numero de colunas calculadas na tabela completa) se o valor *)

(* foi obtido dentro dos limites da tolerancia eps *)

(* 16, se o numero maximo de extrapolacoes foi alcancado *)


```

Function romint(var a,b,eps,err: real; var n,mfin: integer): real;
var sh,r,t,bb,s,h,s0,s1,aux,aux1: real;
    k,n0,n1,k2,kk,kkk,l,k1,k0,j,maxe: integer;
    rm:array[1..16]of real;
    flag:boolean;

```

```

begin

```

```

    sh:=b-a;

```

```

    t:=sh*(fun(a)+fun(b))*0.5;

```

```

    aux:=(a+b)*0.5;

```

```

    rm[1]:=sh*fun(aux);

```

```

    n:=2;

```

```

    r:=4;

```

```

    err:=1;

```

```

    k:=0;

```

```

    maxe:=15;

```

```

    flag:=false;

```

```

repeat

```

```

    k:=k+1;

```

```

    bb:=(r*0.5-1)/(r-1);

```

```

    t:=rm[1]+bb*(t-rm[1]);

```

```

    n:=2*n;

```

```

    s:=0;

```

```

    h:=sh/n;

```

```

    if n<=32 then

```

```

        begin

```

```

            n0:=n;

```

```

            n1:=1;

```

```

        end

```

```

    else

```

```

        begin

```

```

            n0:=32;

```

```

            if n<=512 then n1:=n

```

```

            else n1:=512;

```

```

        end;

```

```

    k2:=1;

```

```

    k1:=k2;
    while k1<=kk do
        begin
            s0:=0;

            kkk:=k1+n0-1;

            k0:=k1;

            while k0 <= kkk do
                begin
                    aux1:=a+k0*h;
                    s0:=s0+fun(aux1);
                    nf:=nf+1;
                    k0:=k0+2;
                end;

                s1:=s0+s1;
                k1:=k1+32;
            end;

            s:=s+s1;
            k2:=k2+512;
        end;

        rm[k+1]:=2*h*s;
        r:=4;
        for j:=1 to k do
            begin
                l:=k+1-j;
                rm[l]:=rm[l+1]+(rm[l+1]-rm[l])/(r-1);
                r:=4*r;
            end;

            err:=abs(t-rm[1])*0.5;
            flag:=(err<=eps);
        until (k=maxe) or flag;
        romint:=(t+rm[1])*0.5;
        n:=n+1;
        if k>= maxe then mfin:=16
        else mfin:=k;
    end;
end;

```

(*

A.3 - FUNCAO INTEGRATOR

*)

(* Esta funcao e baseada na funcao dada por Havie (1967) que e basica- *)
(* mente o algoritmo 257 dado por Kubik (1965), baseado na modificacao do *)
(* algoritmo de Romberg descrito em Havie (1966). Na versao dada por Havie *)

(* (1967) e incorporado um modo de calculo sobre um algoritmo similar derl- *)
(* vado da transformacao cosseno de Euler Maclaurin e segunda formula de Eu- *)
(* ler. Para algumas importantes classes de integrais, esta transformacao a- *)
(* celera os calculos. *)
(* Um subprograma FUNCAO FUN(X), correspondente ao integrando, deve ser *)
(* suprido pelo usuario. *)

(*

Parametros de Entrada e/ou Saida

*)

(* a,b - extremos do intervalo de integracao *)
(* eps - precisao relativa requerida e quando retorna ao programa de chama- *)
(* da e igual a estimativa da precisao relativa obtida. *)
(* m - o numero maximo de linhas a serem calculadas na M-tabela e m e na T- *)
(* tabela e m+1. O calculo e terminado quando a precisao relativa obtida *)
(* e $\leq$ EPS ou quando o numero maximo de linhas na T-tabela e obtido. *)
(* Quando retorna ao programa de chamada m e igual ao numero atual de *)
(* linhas na T-tabela. *)
(* iop - modo de calculo; iop=0, metodo normal e iop=1 metodo baseado sobre *)
(* a transformacao cosseno. Quando retorna ao programa de chamada, iop *)
(* sera um inteiro identificando o elemento na m-esima linha na T- *)
(* tabela que e tomado como a aproximacao final para a integral. Se *)

```

Function integrator(var a,b,eps:real;var nf,m,iop:integer):real;
type arr=array[1..16] of real;
var x,alf,bet,endpts,sum,sum1,error,ar,h,d,z,cof,accur:real;
    cosh,sinh,cosh,sink,const1,const2,const3,store1,store2:real;
    n,i,j,iacc:integer;
    t,u:arr;
    flag:boolean;

(* aqui a U-tabela corresponde a M-tabela *)

begin
    alf:=0.5*(b-a);
    bet:=0.5*(b+a);
    ar:=0.4112335167;
    i:=0;
    m:=15;
    n:=1;
    h:=2;
    x:=a;
    t[1]:=fun(x);
    x:=b;
    t[1]:=t[1]+fun(x);
    nf:=nf+2;
    if iop = 1 then
        begin
            endpts:=t[1];
            t[1]:=0.8224670334*endpts;
            h:=3.141592654;
        end;
    flag:=false;

```

```

    x:=alf*(z+cof)+bet;
    sum:=sum+fun(x);
    cof:=cof+h;
    nf:=nf+1;

end;

sum:=h*sum;

end

else (*calculo de u(1) no caso ordinario*)
begin(*calculo de u(1) no caso nao ordinario*)
(*calculo inicial*)
    sink:=sin(0.5*h);
    cosk:=-cos(0.5*h);
    sinh:=-2*sink*cosk;
    cosh:=(cosk*cosk)-(sink*sink);
    cof:=2*cosh;
    sum:=sink;
    const2:=0;
    const3:=0;
    for j:=n downto 1 do
        begin
            x:=alf*cosk+bet;
            const1:=fun(x)+cof*const2-const3;
            const3:=const2;
            const2:=const1;
            store1:=cosk*cosh+sink*sinh;
            store2:=sink*cosh-cosk*sinh;
            cosk:=store1;
            sink:=store2;
            nf:=nf+1;
        end;
        sum:=h*(const2+const3)*sum-ar*endpts;
    end;
end;

```

```

    sum1:=sum+(sum-u[j-1])/(d-1);
    u[j-1]:=sum;
    sum:=sum1;
    d:=4*d;
end;

u[i]:=sum1;

end;

(* calculo da i-esima coluna na u-tabela terminado *)
(* comeco do calculo da (i+1)-esima coluna da t-tabela *)
(*teste se o calculo e terminado*)
    accur:=1;
    iacc:=1;
    for j:=1 to l do
    begin
        sum:=0.5*(t[j]+u[j]);
        error:=0.5*abs((t[j]-u[j])/sum);
        if error <= accur then
            begin
                accur:=error;
                iacc:=j;
            end;
        t[j]:=sum;
    end;
    if iacc=i then iacc:=i+1;
    t[i+1]:=t[i]-(t[i]-u[i])/(d-1);
    n:=2*n;
    h:=0.5*h;
(* calculo do proxima linha na tabela u *)
    flag:=accur<=eps;
    until (i=m) or flag;
    eps:=accur;
    m:=i+1;
    integrator:=alf*t[iacc];
end;

```

(*

A.4 - FUNCAO GAUSS

*)

(* Esta funcao e usada para aproximar a integral definida (I.1) pelo me- *)
 (* todo de Gauss-Romberg, dado por Lyness (1972) Usamos a sequencia de *)
 (* raios m_j , $\{m_j\} = \{1, 2, 3, 4, 6, 8, 12, \dots\}$ (gerada na rotina) e substituímos a *)
 (* regra trapezoidal pela regra de Gauss-Legendre de $t+1$ pontos, para o cal- *)
 (* culo das aproximacoes iniciais (primeira coluna da G-tabela). Esta ultima *)
 (* regra corresponde a Funcao Gaule, apresentada a seguir, onde $1 \leq t \leq 5$. *)
 (* Para $t=1$ tenta-se obter I com a G-tabela contendo ate 5 linhas ($j \leq 5$). *)
 (* Caso nao se consiga a precisao requerida, repetimos o processo sucessiva- *)
 (* mente com $t=2, 3, 4$. Se ainda nao houve sucesso, com $t=5$ tentamos a G-tabe- *)
 (* la ate 25 linhas ($j \leq 25$). Caso nao obtenha o valor aproximado da inte- *)
 (* gral dentro dos limites das tolerancias especificadas, o programa sai com *)
 (* o ultimo valor obtido e FLAG=FALSE. O erro absoluto e o relativo sao os *)
 (* mesmos que os utilizados na Funcao Quad. A funcao FUN(X) para calcular o *)
 (* integrando, deve ser providenciada pelo usuario, assim como a funcao *)
 (* expon(u,v) para calcular as potencias v de u, onde v e um inteiro. *)

(*

Parametros de Entrada

*)

(* a,b - extremos do intervalo de integracao *)
 (* eps,eta - tolerancia para o erro absoluto e relativo, respectivamente *)

(*

Parametros de Saída

*)

(* err,era - erro relativo e absoluto cometidos no calculo do valor aproxi- *)
 (* mado, respectivamente *)
 (* t - $t+1$ e o grau do polinomio de Legendre com o qual se obteve o valor *)
 (* aproximado *)
 (* nf - numero de calculos da funcao requerido para se obter o valor aproxi- *)
 (* mado *)

```
Function gauss (var a,b,eps,eta,err,era:real;var t,nf:integer;var flag:boolean):real;  
type arr=array[1..30] of real;
```

```
var mj,x,w:arr
```

```
    g:array[1..6200] of real;
```

```
    nc,j,mjj,k,l,i,t1,t2:integer;
```

```
    aux,aux1,gx1,gx2,sum,aj,bj,hj,gabs,err1:real;
```

```
begin
```

```
    j:=0;
```

```
    mj[1]:=1;
```

```
    repeat
```

```
        j:=j+2;
```

```
        aux:=expon(2,(j-2) div 2);
```

```
        mj[j]:=2*aux;
```

```
        mj[j+1]:=3*aux;
```

```
    until j=24;
```

```
    nc:=5;
```

```
    nf:=0;
```

```
    err:=1;
```

```
    era:=1;
```

```
    t1:=0;
```

```
    repeat
```

```
        flag:=false;
```

```
        t1:=t1+1;
```

```
        t:=t1;
```

```
        if t=5 then nc:=25;
```

```
        j:=0;
```

```
        repeat
```

```
            j:=j+1;
```

```
            sum:=0;
```

```
            aj:=a;
```

```
            hj:=(b-a)/mj[j];
```

```
            mjj:=trunc(mj[j]);
```

```
            for l:=1 to mjj do
```


g[i]:=gaule(aj,bj,t,nf);

aj:=bj;

sum:=sum+g[i];

end;

t2:=2*t;

aux:=expon(mj[j],t2);

x[j]:=aux;

gabs:=abs(sum);

g[j]:=sum;

w[j]:=aux*g[j];

if j>=2 then

begin

l:=0;

for k:=2 to j do

begin

l:=l+1;

i:=j+1-k;

aux1:=(mj[j]/mj[j-1])-1;

w[i]:=w[i+1]+(w[i+1]-w[i])/aux1;

(* FUNCAO GAULE

(* Parametros de Entrada

(* aj,bj - extremos do intervalo de integracao

(* k - k+1 e o numero de pontos base

(* Parametros de Saída

(* nf - numero de calculos da funcao

Function gaule(var aj,bj:real;var k,nf:integer):real;

type arr=array[1..11] of real;

var xi,wi:arr;

ky:array[1..6] of integer;

c,d,soma,waux:real;

ji,jf,jj:integer;

begin

ky[1]:=1;

ky[2]:=2;

ky[3]:=4;

ky[4]:=6;

ky[5]:=9;

ky[6]:=12;

xi[1]:=0.577350269189626;

xi[2]:=0.0;

xi[3]:=0.774596669241483;

xi[4]:=0.339981043584856;

xi[5]:=0.861136311594053;

xi[6]:=0.0;

xi[7]:=0.538469310105683;

xi[8]:=0.906179845938664;

xi[9]:=0.238619186083197;

xi[10]:=0.661209386466265;

xi[11]:=0.932469514203152;

wi[12]:=1.0;

wi[2]:=0.888888888888889;

wi[3]:=0.555555555555556;

wi[4]:=0.652145154862546;

wi[5]:=0.347854845137454;

wi[6]:=0.568888888888889;

wi[7]:=0.478628670499366;

wi[8]:=0.236926885056189;

wi[9]:=0.467913934572691;

wi[10]:=0.360761573048139;

wi[11]:=0.171324492379170;

ji:=ky[k];

jf:=ky[k+1]-1;

c:=(bj-aj)*0.5;

d:=(bj+aj)*0.5;

(*

A.5 - FUNCAO IRATEX

*)

(* Esta funcao e usada para aproximar a integral definida (I.1), dentro *)

(* dos limites da precisao EPS especificada, correspondente ao metodo de *)

(* Bulirsch e Stoer(1967). E baseada na rotina do mesmo nome IRATEX, dada em *)

(* Davis e Rabinowitz (1975), apendice 2, a qual por sua vez baseia-se no *)

(* procedimento ALGOL "Quadrature", dado por Bulirsch e Stoer. Um subprogra- *)

(* ma FUNCAO FUN(x), correspondente ao integrando, deve ser suprido pelo u- *)

(* suario. ETA e um parametro dependente da maquina utilizada: se ela tem *)

(* representacao com D posicoes decimais na mantissa, entao $ETA=1.E(-D+1)$, *)

(* com o correspondente valor de D inserido. *)

(*

Parametros de Entrada

*)

(* a,b - extremos do intervalo de integracao

*)

(* eps - precisao requerida

*)

begin

if eps < eta then eps:=eta;

n:=2;

nm:=3;

ba:=b-a;

t1:=0;

gr:=0;

sm:=0;

t2a:=(fun(a)+fun(b))*0.5;

nf:=2;

t2:=t2a;

tb:=abs(t2a);

c:=t2*ba;

dt[0]:=c;

odd:=true;

bu:=false;

ind:=1;

for m:=1 to 15 do

begin

bo:=m >= 7;

hm:=ba/n;

else

begin

i:=1;

while i <= n do

begin

w:=i*hm;

aux:=a+w;

aux1:=b-w;

t1:=fun(aux)+fun(aux1)+t1;

nf:=nf+2;

i:=i+6;

end;

nt:=t1+t2a;

t2a:=t2;

d[1]:=9/4;

d[3]:=9;

d[5]:=36;

end;

ddt:=dt[0];

if den $\neq$ 0 then

begin

e:=e/den;

v:=nt*e;

nt:=dl*e;

t:=v+t;

end

else

begin

v:=0;

nt:=0;

end;

ddt:=dt[1];

dt[1]:=v;

end;

ta:=c;

c:=t;

int:=t;

if not bo then t:=-v+t;

bo:=false;

10:v:=eta*tab;

if err < v then err:=v;

if not bo then ind:=0;

iratex:=int;

end;

(*

A.6 - FUNCAO PRAGER

*)

(* PRAGER e usada para aproximar a integral definida (I.1) pelo algorit- *)

(* mo adaptativo sugerido por Prager (1965) e descrito na secao 5.2.1. *)

(* A funcao PRAGER faz uso da funcao QUAD (ja descrita), com algumas altera- *)

(* coes, dentre as quais: o tipo de erro para $|APROX2 - APROX1| \leq$ *)

(* $EPS \cdot \max(1, APROX2)$; O numero maximo de passos executados para 5 e a inclu- *)

(* sao de um parametro N que contem a posicao do ultimo elemento calculado *)

(* da diagonal principal da T-tabela, onde a precisao requerida foi satis- *)

(* feita. O usuario deve providenciar uma FUNCAO FUN(x), correspondente ao *)

(* integrando, e uma FUNCAO AMAX2(u,v) que calcula o maior entre dois reais *)

(* u,v. *)

(*

Parametros de Entrada

*)

```

begin
    errm:=0;
    nf:=0;
    nmax:=(8*(b-a))/h1+0.1;

    cont:=0;
    sum:=0;

    a1:=a;

    b1:=a+h1;
repeat
    vquad:=quad(a1,b1,eps,err,n,nf,flag);
    if flag then
        begin
            sum:=sum+vquad;
            errm:=amax2(errm,err);

            if n=2 then h1:=1.5*h1
                else if n=5 then h1:=0.6*h1;

            a1:=b1;
            b1:=a1+h1;
            cont:=cont+1;
        end
    else
        begin
            h1:=0.6*h1;
            b1:=a1+h1;
        end;
    if b1>b then b1:=b;
    if a1=b1 then b1:=b1+1;
until (b1>b) or (cont>nmax);
if cont>nmax then writeln(1st,'nro de intervalos excede o maximo permitido ');
prader:=sum;

```

(*

A.7 - FUNCAO CADRE

*)

(* CADRE calcula uma aproximacao para a integral definida (I.1) para uma *)

(* precisao relativa RERR ou uma precisao absoluta AERR. Uma funcao FUN(X) *)

(* correspondente ao integrando, deve ser providenciada pelo usuario, assim *)

(* como as funcoes AMAX2 e AMAX3 que determinam o maior elemento de um con- *)

(* junto de 2 e 3 elementos, respectivamente e a funcao AMIN1 que determina *)

(* o menor elemento de um conjunto de dois elementos. TOLMCH e um parametro *)

(* dependente da maquina, depende do comprimento da mantissa de ponto flu- *)

(* tuante. CADRE foi escrito por Boor (1971), em FORTRAN, e se adapta ao *)

(* integrando, isto e, conforme seu comportamento. *)

(* Uma aproximacao para a integral e calculada como a soma de aproximacoes *)

(* sobre pequenos intervalos, apropriadamente. Em cada subintervalo uma es- *)

(* timativa VINT, com erro absoluto ERRER e encontrada por extrapolacao *)

(* cautelosa de Romberg. Se ERRER e pequeno o suficiente, VINT e aceita e *)

(* adicionada a variavel CADRE e ERRER e adicionada a ERROR. Caso contrario *)

(* o subintervalo e dividido, e cada metade e considerada separadamente, *)

(* sendo as informacoes sobre a outra metade temporariamente esquecidas. *)

(*

Parametros de Entrada

*)

(* a,b - extremos do intervalo de integracao

*)

(* nome para a funcao integrando

Function cadre (var a,b,aerr,rerr,error: real; var iflag,nf: integer): real;

label 5,6,10,12,18,21,22,30,31,33,37,50,60,70,71,72,80,90,91,94,95,900,960,999;

label 950;

type rad=array[1..10] of real;

bfe=array[1..30] of real;

const tolmch=2.e-13;

aitlow=1.1;

h2tol=0.15;

aittol=0.1;

jump1=0.01;

maxts=2049;

maxtbl=10;

mxstge=30;

var r,ait,dif: rad;

begn,finis,est:bfe;

t:array[1..10,1..10] of real;

ts:array[1..2049] of real;

rn:array[1..4] of real;

reglsv:array[1..30] of boolean;

ibegs:array[1..30] of integer;

h2conv,aitken,right,reglar:boolean;

alg402,lenght,err,errs,stepmn,stage,curest,fnsz,prever,beg,fbeg: real;

pend,fpnd,step,astep,fn,hovn,sum,cumabs,tabs,abs: real; nint: integer;

```
if lenght<>0 then
```

```
begin
```

```
err:=amin1(0.1,amax2(abs(rerr),10*tolmch));
```

```
erra:=abs(aerr);
```

```
stepmn:=amax2(lenght/expon(2,30),amax3(lenght,abs(a),abs(b))*tolmch);
```

```
stage:=0.5;
```

```
!stage:=1;
```

```
currest:=0;
```

```
fnsiz:=0;
```

```
prever:=0;
```

```
reglar:=false;
```

```
(* o intervalo de integracao e o primeiro considerado *)
```

```
beg:=a;
```

```
fbeg:=fun(beg)/2;
```

```
ts[1]:=fbeg;
```

(* calcula a proxima soma trapezoidal $T(L,1)$, que e baseada so-*)

(* bre $N2+1$ pontos equidistantes. Aqui $N2=N*2=2*(L-1)$ *)

10:1m1:=1;

1:=1+1;

n2:=n*2;

fn:=n2;

istep:=(iend-ibeg) div n;

if istep <= 1 then

begin

ii:=iend;

iend:=iend+n;

if iend > maxts then goto 900

else begin

hovn:=step/fn;

iii:=iend;

i:=1;

while i <= n2 do

begin

ts[iii]:=ts[ii];

```
tabs:=tabs/2+sumabs/fn;
```

```
absi:=astep*tabs;
```

```
n:=n2;
```

```
it:=1;
```

```
vint:=step*t[1,1];
```

```
tabt1m:=tabs*tolmch;
```

```
fnsizel:=amax2(fnsizel,abs(t[1,1]));
```

```
ergoal:=amax2(astep*tolmch*fnsizel,stage*amax2(erra,errr*abs(curest+vint)));
```

```
fextrp:=1;
```

```
for i:=1 to 1m1 do
```

```
begin
```

```
    fextrp:=fextrp*4;
```

```
    t[1,1]:=t[1,i]-t[1-1,i];
```

```
    t[1,i+1]:=t[1,i]+t[i,1]/(fextrp-1);
```

```
end;
```

```
errr:=astep*abs(t[1,1]);
```

```
if 1 <= 2 then
```

```
begin
```

```
    if(abs(t[1,2]) <= tabt1m) then goto 60
```

```
    else goto 10;
```

```
end;
```

else

begin

if regular then goto 18

else

begin

if l=4 then goto 10

else goto 18;

end;

end;

end;

18:if(error <= ergoal) then goto 70

else goto 91;

end;

(* estima VINT (valor aproximado) assumindo integrando suave

if h2conv then goto 21

else

begin

aitken:=false;

h2conv:=true;

else

begin

if aitken then goto 31

else

begin

h2conv:=false;

aitken:=true;

end;

end;

31:fextrp:=t[1-2,1m1];

if(fextrp > 4.5) then goto 21

else

begin

if(fextrp < aitlow) then goto 91

else

begin

if(abs(fextrp-t[1-3,1m1])/t[1,1m1] > h2tol) then goto 91

else

begin

sing:=fextrp;

it:=it+1;

if (it < lm1) then

begin

if (it <= 3) then

begin

h2next:=4;

singnx:=2*sing;

end;

if (h2next >= singnx) then

begin

fextrp:=singnx;

singnx:=2*singnx;

goto 37

end;

fextrp:=h2next;

h2next:=4*h2next;

37:for i:=it to lm1 do

begin

r[i+1]:=0;

if(abs(dif[i+1]) > tabt1m) then r[i+1]:=dif[i]/dif[i+1];

end;

(* soma trapezoidal corrente e os valores extrapolados resul-*)

(* tantes nao dao ERRER pequeno suficiente. Se menos que 5 *)

(* somas trapezoidais foram calculadas ate entao, tente pro- *)

(* xima soma trapezoidal; caso contrario decide se sai ou se *)

(* subdivide como segue *)

fextrp:=amax2(prever/error,aitlow);

prever:=error;

if 1 < 5 then goto 10

else

begin

if ((1-it) > 2) and (istage < mxstge) then goto 90

else

begin

if(error/expon(fextrp,maxtbl-1)) < ergoal then goto 10

else goto 90;

end;

end;

(* setias HUNT (consequencia) calculada para = 1.0

goto 80;

(* estima VINT (aproximacao) assumindo variacao no integrando *)

(* ("noise" parece ser dominante) *)

70:slope:=(fpend-fbeg)*2;

fbeg2:=fbeg*2;

i:=1;

71:diff:=abs(fun(beg+rn[i]*step)-fbeg-rn[i]*slope);

nf:=nf+1;

72:error:=amax2(error,astep*diff);

if(error > ergoal) then goto 91

else

begin

i:=i+1;

if i <= 4 then goto 71

else iflag:=3;

end;

end;

(* integracao sobre o subintervalo corrente bem sucedida *)

```
curest:=curest+vint;
```

```
stage:=stage*2;
```

```
lend:=beg;
```

```
lbeg:=lbegin[stage];
```

```
pend:=beg;
```

```
beg:=begin[stage];
```

```
fpend:=fbeg;
```

```
fbeg:=ts[lbeg];
```

```
goto 5;
```

```
(* Integracao sobre o subintervalo corrente nao foi bem sucedida *)
```

```
(* marca subintervalo para mais subdivisao e toma o proximo sub- *)
```

```
(* intervalo *)
```

```
90:regular:=true;
```

```
91:if lstage < mxstge then
```

```
begin
```

```
if right then goto 95
```

```
else
```

ts[ii]:=ts[i];

end;

for i:=ibeg to li do

begin

ts[iii]:=ts[i];

iii:=iii+1;

end;

iend:=iend+1;

ibeg:=iend-nnleft;

fpend:=fbeg;

fbeg:=ts[ibeg];

finis[istage]:=pend;

pend:=beg;

beg:=begn[istage];

begn[istage]:=pend;

reglsv[istage]:=reglar;

istage:=istage+1;

(*)

A.8 - FUNCAO SIMPSON

(*)

(* SIMPSON dada aqui em Pascal, foi escrita por McKeeman e Tesler (1963) *)

(* em Algol e e uma traducao nao recursiva do algoritmo145 dado por McKeeman *)

(* (1962). Esta modificacao adaptativa da regra de Simpson, esta descrita na *)

(* secao 5.3.1 e tenta minimizar o numero de calculos da funcao usando pe- *)

(* quenas subdivisoes do intervalo, somente onde necessario para atingir a *)

(* tolerancia especificada. *)

(* Parametros de Entrada *)

(* a,b - extremos do intervalo de integracao *)

(* eps - tolerancia requerida *)

(* Parametros de Saida *)

begin

lvl:=0;

absar:=0;

est:=0;

da:=b-a;

fa:=fun(a);

aux:=(a+b)*0.5;

fm:=4.*fun(aux);

fb:=fun(b);

nf:=3;

10:lvl:=lvl+1;

dx[lvl]:=da/3;

sx:=dx[lvl]/6;

aux:=0.5*dx[lvl]+a;

f1:=4.*fun(aux);

x2[lvl]:=a+dx[lvl];

f2[lvl]:=fun(x2[lvl]);

x3[lvl]:=x2[lvl]+dx[lvl];

f3[lvl]:=fun(x3[lvl]);

fm:=f1;

fb:=f2[lv1];

eps:=epsp[lv1]/1.7;

est:=est1;

goto 10;

30:case 1 of

1:begin

nrtr[lv1]:=2;

da:=dx[lv1];

fa:=f2[lv1];

fm:=fmp[lv1];

fb:=f3[lv1];

eps:=epsp[lv1]/1.7;

est:=est2[lv1];

a:=x2[lv1];

goto 10;

end;

2:begin

nrtr[lv1]:=3;

da:=dx[lv1];

fa:=f3[lv1];

fm:=f4[lv1];

fb:=fbp[lv1];

eps:=epsp[lv1]/1.7;

(*

A.9 - FUNCAO SQUANK

*)

(* A construçao desta rotina e descrita em detalhe na secção 5.3. Em re- *)

(* sumo, ela e baseada nas ideias da rotina de Simpson adaptativa, engloban- *)

(* do quatro modoficacoes importantes: um diferente assinalamento de obten- *)

(* cao de erro para o intervalo e um criterio de convergencia do intervalo *)

(* diferente; bisseccao do intervalo no lugar de trisseccao; inclusao de um *)

(* termo de ajustamento para dar um resultado de polinomio de grau 5 no lu- *)

(* gar de grau 3 e uma protecao contra os efeitos de excessivos erros de ar- *)

(* redondamento nos valores da funcao. Esta rotina foi dada por Lyness(1970) *)

(* em FORTRAN, aqui apresentamos a mesma na linguagem Pascal. *)

(* Parametros de Entrada *)

(* a,b - extremos do intervalo de integracao *)

(* error-tolerancia requerida (erro absoluto) *)

(* Parametros de Saida *)

(* fifth-termo de ajustamento de quinta ordem, que pode ser usada como uma *)

(* estimativa do erro, em casos onde o erro de arredondamento nao e *)

(* significante *)

(* rum - tolerancia obtida para o erro de arredondamento ajustado; e normal- *)

(* mente o mesmo que a tolerancia requerida, exceto em casos onde o *)

(* erro de arredondamento e significante, quando e maior que toleran- *)

(* cia requerida *)

(* no - numero de calculos da funcao requerido para obter o valor aproxima- *)

(* do *)

Function squank(var a,b,error,fifth,rum:real;var no:integer):real;

Label 700, 705, 710, 715, 720, 725, 730, 735, 740, 745, 750, 755, 760, 765, 770, 775, 780, 785, 790, 795, 800, 805, 810, 815, 820, 825, 830, 835, 840, 845, 850, 855, 860, 865, 870, 875, 880, 885, 890, 895, 900, 905, 910, 915, 920, 925, 930, 935, 940, 945, 950, 955, 960, 965, 970, 975, 980, 985, 990, 995

```
begin
```

```
(* estagio um *)
```

```
(* inicializa todas as quantidades requeridas p/ calculo central-estagio3-*)
```

```
sum:=0;
```

```
sim:=0;
```

```
cepsf:=180.*error/(b-a);
```

```
ceps:=cepsf;
```

```
adiff:=0;
```

```
levtag:=-1;
```

```
facerr:=1.0;
```

```
xzero:=a;
```

```
efact:=0;
```

```
nim:=1;
```

```
lev:=0;
```

```
(* primeiro intervalo *)
```

```
x1:=a;
```

```
x5:=b;
```

```
x3:=0.5*(a+b);
```

```
fx1:=fun(x1);
```

```
fx3:=fun(x3);
```

```
fx5:=fun(x5);
```

```

        if fx5=0 then ceps:=epmach;
    end;

end;

end;

end;

qceps:=0.25*ceps;

(* inicializacao completa *)

(* estagio tres *)

(* calculo central, requer x1,x3,x5,fx1,fx3,fx5,est,adiff *)

300:x2:=0.5*(x1+x3);
    x4:=0.5*(x3+x5);
    fx2:=fun(x2);
    fx4:=fun(x4);
    no:=no+2;
    est1:=fx1+4.0*fx2+fx3;
    est2:=fx3+4.0*fx4+fx5;
    adiff1:=adiff;
    diff:=est+est-est1-est2;
    if lev<30 then
        begin
            adiff:=abs(diff);
            crit:=adiff-ceps;
            if crit>0 then
                begin
                    (* fim do loop central *)

```

```
if adiff1=adiff then
```

```
begin
```

```
ceps:=adiff;
```

```
goto 445;
```

```
end;
```

```
ceps:=2.0*ceps;
```

```
if levtag<3 then levtag:=2;
```

```
445:qceps:=0.25*ceps;
```

```
end;
```

```
end;
```

```
(* estagio cinco *)
```

```
(* nenhuma convergencia natural*)
```

```
(* armazenando os elementos do lado direito
```

```
nim:=2*nim;
```

```
lev:=lev+1;
```

```
estst[lev]:=est2;
```

```
x3st[lev]:=x4;
```

```
x5st[lev]:=x5;
```

```
fx3st[lev]:=fx4;
```

```
fx5st[lev]:=fx5;
```

```
predif[lev]:=adiff;
```

```
(* estagio seis *)
```

```
(* quantidades para calculo central *)
```

```
(* levtag=2 ceps e maior que cepsf.ceps previamente questionado por um jump *)
```

```
(*          b.mas nao obtido *)
```

```
(* levtag=3 ceps e maior que cpsf.ceps previamente teve um jump b *)
```

```
if levtag>=0 then
```

```
begin
```

```
(* numa execucao normal com nenhum problema de erro de arredondamento,*)
```

```
(* levtag=-1 e o resto do estagio sete e omitido *)
```

```
cepst:=15.0*ceps;
```

```
(* cepst aqui e o valor facerr*current de ceps *)
```

```
if crit<0 then
```

```
begin
```

```
if levtag=2 then
```

```
begin
```

```
levtag:=0;
```

```
if adiff<=0 then
```

```
begin
```

```
ceps:=adiff1;
```

```
goto 775;
```

```
end else goto 725;
```

```
end;
```

```
if levtag>2 then
```

```
begin
```

```
levtag:=0;
```

else

begin

levtag:=1;

facerr:=1;

ceps:=cepsf;

775:efact:=efact+cepst*(x1-xzero);

xzero:=x1;

780:qceps:=0.25*ceps;

(* estagio oito *)

(* convergencia de fato em intervalo anterior. Adicionados incrementos p/ *)

(* execucao de somas - adiciona para sum e sim *)

end;

end;

end;

end;

end;

end;

end;

end;

sum:=sum+(est1+est2)*(x5-x1);

if levtag<0 then

begin

(* adicionamos para sim so se estamos certos de nivel de arredondamento*)

sim:=sim+100*(x5-x1);

```
(* estagio dez *)
```

```
(* pondo quantidades para calculo central
```

```
  x1:=x5;
```

```
  fx1:=fx5;
```

```
  x3:=x3st[lev];
```

```
  x5:=x5st[lev];
```

```
  fx3:=fx3st[lev];
```

```
  fx5:=fx5st[lev];
```

```
  est:=estst[lev];
```

```
  adiff:=predif[lev];
```

```
  goto 300;
```

```
end
```

```
(* estagio onze *)
```

```
(* calculo completo - finalizacao- *)
```

```
  else
```

```
    begin
```

```
      efact:=efact+ceps*(b-xzero)*facerr;
```

```
      rum:=efact/180;
```

```
      third:=sum/12;
```

```
      fifth:=-sim/180;
```

```
      squank:=third+fifth;
```

```
    end;
```


BIBLIOGRAFIA

- (1973) - ALBRECHT, P. *Análise Numérica - Um Curso Moderno*, Livros Técnicos e Científicos Editora S.A., Editora da Universidade de São Paulo, Rio de Janeiro.
- (1966) - ANDERS, E.B. *An Extension of Romberg Integration Procedures to N Variables*, JACM 13, 505-510.
- (1971) - _____ *Asymptotic Expansions for Integration Formulas in one or more Dimensions*, SIAM J. Numer. Anal. 8, 473-480.
- (1961) - BAUER, F.L. *La Méthode d'Integration Numérique de Romberg*, Colloq. Anal. Numér., 119-129. Librairie Universitaire, Louvain.

(1967) - BULIRSCH, R., and STOER, J. Handbook Series Nume-

rical Integration: Numerical Quadrature by Interpolation, Numer. Math. 9, 271-278.

(1969) - CASALETTO, J., PICKETT, M., and RICE, J.R. A Com

parison of some Numerical Integration Programs,

SIGNUM Newsletter 4, Nº 3, 30-40.

(1969) - CHASE, S.M., and FOSDICK, L.D. *An Algorithm for Filon Quadrature, CACM 12, 453-457.*

(1960) - CLENSHAW, C.W., and CURTIS, A.R. *A Method for Numerical Integration on an Automatic Computer, Numer. Math. 2, 197-205.*

(1971) - CRANLEY, R., and PATTERSON, T.N.L. *On the Numerical Evaluation of Definite Integrals, Comput J. 14, 189-198.*

(1977) - CONTE, S.D. *Elementos de Analise Numérica*, Editora Globo S.A., Porto Alegre - Rio Grande do Sul, 3ª Edição.

(1975) - DAVIS, P.J., and RABINOWITZ, P. *Numerical Integration, Academic Press, New York, San Francisco*

(1966) - HAVIE, T. On a Modification of Romberg's Algorithm, BIT 6, 24-30.

(1967) - _____ On the Practical Application of the Modified Romberg Algorithm, BIT 7, 103-113.

(1969) - _____ Derivation of Explicit Expressions for the Error Terms in the Ordinary and the Modified Romberg Algorithms, BIT 9, 18-29.

(1969) - _____ On a Modification of the Clenshaw-Curtis Quadrature Formula, BIT 9, 338-350.

(1971) - _____ Some Methods for Automatic Integration and their Implementation on the CERN CDC 65/6600 Computers, Rep. 71-26. CERN, Geneva.

(1972) - _____ Error Derivation in Romberg Integration, BIT 12, 516-527.

(1970) - HILLSTROM, K.E. Comparison of Several Adaptive Newton-Cotes Quadrature Routines in Evaluating Definite Integrals with Peaked Integrands. II, CACM 13, 362-365.

(1969) - HODGSON, G.S. Numerical Integration, MSc Thesis,

(1965) - KUBIK, R.N. ALGORITHM 257. *Hävie Integrator*,
CACM 8, 381 (Remark: CACM 9 (1966) 795, 871).

(1964) - LAURENT, P.J. *Formules de Quadrature Aproximée
sur Domaines Rectangulaires Convergents pour
toute Fonction Intégrable Riemann*, C.R. Acad.
Sci. Paris 258, 798-801.

(1983) - LAURIE, D.P. *Sharper Error Estimates in Adaptive
Quadrature*. BIT 23, N° 2, 258-261.

(1972) - LINZ, P. ALGORITHM 427. *Fourier Cosine Inte-
gral*, CACM 15, 358-360.

(1967) - LYNCH, R.E. *Generalized Trapezoid Formulas and
Errors in Romberg Quadrature*, In Blanch Anniv.
Vol. (Mond. B., ed.), 215-229. Office of Aeros-
pace Res., USAF, Washington, D.C.

(1969) - LYNESS, J.N. *Notes on the Adaptive Simpson Qua-
drature Routine*, JACM 16, 483-495.

(1969a) - _____ *The Effect of Inadequate Convergence
Criteria in Automatic Routines*, Comput. J. 12,
279-281.

(1970) - _____ ALGORITMO 379. *SQUANK (Simpson Qua-
drature used Adaptively-Moise Kibodi)* CACM 13

Quadrature Routines, Comput. J. 20, N° 2, 170-177.

(1963-1964)- LYNESS, J.N., and MacHUGH, J.J. *Integration over Multidimensional Hypercubes. I: A Progressive Procedure, Comput. J. 6, 264-270.*

(1970) - _____, _____ *On the Remainder Term in the N-Dimensional Euler-MacLaurin Expansion. Numer. Math., 15, 333-444.*

(1962) - McKEEMAN, W.M. ALGORITHM 145. *Adaptive Numerical Integration by Simpson's Rule. CACM 15, N° 12, 604.*

(1963) - _____ CERTIFICATION OF ALGORITHM 145. *Adaptive Numerical Integration by Simpson's Rule. CACM 6, N° 4, 167-168.*

(1963) - _____, and TESLER, L. ALGORITHM 182. *Nonrecursive Adaptive Integration. CACM 6, N° 6, 315.*

(1965) - MEIR, A., and SHARMA, A. *On the Method of Rom-*
berg. CACM 8, N° 2, 250.

the Clenshaw-Curtis Quadrature Formula. *Comput. J.* 11, 213-219.

(1969) - O'HARA, H., and SMITH, F.J. The Evaluation of de
finite Integrals by Interval Subdivision. Com
put. J., 12, 179-182.

(1971) - OLIVER, J. The Efficiency of Extrapolation Me-
thods for Numerical Integration. *Numer. Math.*
17, 17-32.

(1973) - PATTERSON, T.N.L. ALGORITHM 468. Algorithm for
Automatic Numerical Integration Over a Finite
Interval. *CACM* 16, 694-699.

(1965) - PRAGER, W. Introduction to Basic Fortran Pro-
gramming and Numerical Methods. GINN(Blaissdell),
Boston, Massachusetts. Cap. IX, 100-124.

(1967) - RUTISHAUSER, H. Description of Algol 60. In
Handbook for Automatic Computation, Vol. 1,
Springer-Verlag, Berlin. Part. a, 105-106.

(1972) - SHANKS, J.A. Romberg Tables for Singular Inte-
grands. *Comput. J.* 15, N° 4, 360-361.

*tum Mechanical Calculations II. Overlap Inte-
gral.* Navord Rep. 5257, U.S. Naval Ordnance
Test Station, China Lake, California.

(1956) - VILLARS, D.S. *"Simultaneous Multiple Integration,
Floating Point"*, Share File N° 0704-0240 NO

SIG.

(1966) - WELSH, J.H. ALGORITHM 281. *Abscissas and
Weights for Romberg Quadrature.* CACM 9, N° 4,
271-272 (Remark: CACM 10, 188).

(1963) - WYNN, P. *Acceleration Techniques in Numerical
Analysis, with Particular Reference to Problems
in one Independent Variable, "Information Pro-
cessing 1962"* (C.M. Popplewell, ed.). North-
Holland Publ., Amsterdam.