

Jussara Maria Marins*

SUMÁRIO

Este trabalho tem por finalidade apresentar as noções básicas dos algoritmos de quadratura adaptativas. Inicialmente mostra uma visão geral dos componentes das rotinas adaptativas. A seguir, dá uma fundamentação teórica simples para uma abordagem geral. A implementação da rotina didática é feita pela aplicação de regra de Simpson num procedimento em PASCAL. Finalmente 3 exemplos são apresentados para mostrar o comportamento típico da rotina.

ABSTRACT

This work is intend to present the basic notions of adaptative quadrature algorithms. Inicially, it shows a general view of the components of adaptive routines. Following this, it gives a simple the oretical foundation for a general approach. The implementation of the didactic routine is done by application of Simpson's rule, in a PASCAL procedure. Finally 3 exemples are presented to show the tipical behavior of the routine.

* Professora Assistente do Instituto Tecnológico de Aeronáutica (ITA); quadratura, integração numérica, programa para integração numérica.

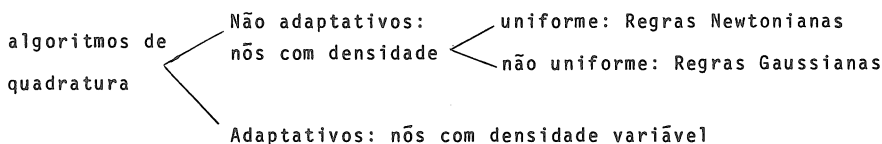
O problema da quadratura é determinar o valor de

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

O objetivo da quadratura numérica é calcular o valor de (1) usando exclusivamente operações aritméticas. Para estimar o valor de (1) vamos aproximá-la por uma combinação de valores da função em certos pontos x_i , $-\infty \leq a \leq x_1 \leq b \leq +\infty$, chamados de nós e certos valores w_i , que constituem pesos, ou seja:

$$(2) \quad \int_a^b f(x) dx \approx w_1 f(x_1) + w_2 f(x_2) + \dots + w_n f(x_n) + w_{n+1} f(x_{n+1})$$

De acordo com os valores dos pesos e com a escolha dos nós temos, no lado direito de (2) o que costumamos chamar de Regra de Integração. As diferentes filosofias surgem na maneira de fazer a escolha dos nós, que se refletem no tipo de algoritmo usado. Numa abordagem simples os algoritmos podem ser classificados como mostra o seguinte esquema:



O objetivo das quadraturas adaptativas é calcular o valor aproximado de (1) variando o tamanho dos subintervalos, de modo que, nos trechos onde a curva é mais abrupta, o número de subintervalos é maior, enquanto que tal número é menor onde a curva é mais suave. Para fazer o cálculo do valor aproximado de (1) usa-se uma ou mais regras básicas de quadraturas Newtonianas ou gaussianas.

Os algoritmos adaptativos são geralmente complicados pois podem fazer uma análise complexa do comportamento do integrando, dependendo dos limites de erro parciais que apresentem em cada subintervalo.

O objetivo deste trabalho é apresentar uma rotina adaptativa simples que possa ser ensinada num curso básico de cálculo numérico para graduação. Contudo, apesar da simplicidade esta rotina permite que faça novas estruturas ampliando os resultados.

Na seção 2 veremos com mais detalhes os diferentes modos de implementar rotinas adaptativas e um esquema geral para quadraturas adaptativas. Na seção seguinte veremos uma fundamentação simples sobre o estudo de um critério de convergência baseado em Forsythe [1]. Na seção 4 veremos os deta

lhes da implementação da rotina adaptativa didática e finalmente na seção 5 veremos exemplos mostrando o desempenho da rotina.

FILOSOFIAS ADAPTATIVAS (2)

Conforme vimos na introdução, uma rotina adaptativa é um algoritmo que usa uma ou mais regras básicas de quadratura e que conforme for o comportamento de f , escolhe dinamicamente o tamanho dos subintervalos e os pesos w_i de (2) de modo que o resultado computado encontre um ou mais critérios de exatidão pré-determinados. Podemos mostrar um exemplo gráfico simples na figura 1, que mostra a escolha dos tamanho dos subintervalos.

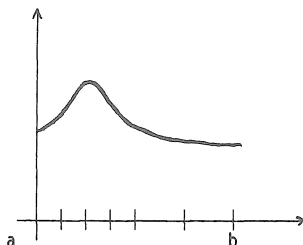


Figura 1

Diferentes tamanhos podem ser usados ao longo do intervalo de integração $[a, b]$ com subintervalos relativamente grandes onde o integrando é suave e de variação lenta e pequenos subintervalos em regiões onde a integração fica mais difícil. Observando a figura 1 é fácil verificar em que região a curva é mais abrupta, mas como verificar isto automaticamente? Pelo exemplo, também observamos na figura 1 que se dividirmos o intervalo $[a, b]$ em subintervalos de tamanho fixo e de tamanho pequeno teríamos um resultado que também prescreveria o critério de exatidão mas a um custo maior. Em tais rotinas de quadratura uma medida de eficiência do algoritmo é o número de avaliações de f que é feito para se atingir a exatidão exigida, o que está bastante ligado com o custo final do processamento. De um modo geral, se uma função é integrada por duas rotinas diferentes com a mesma exatidão a rotina que requer menor número de avaliações da função é vista como a mais eficiente para o problema particular. Usualmente o usuário de rotinas adaptativas ou automáticas não quer pensar qual a melhor regra para usar e qual o melhor número de subdivisões que deverá ser feita para que se tenha uma boa exatidão. Ele quer apenas fornecer a rotina para calcular $f(x)$ ou os pontos dados (x_i, y_i) ; os limites de integração, uma tolerância e para o limite de erro e um limite de segurança L para o número máximo de

avaliações de f . Sendo $I(f)$ o valor exato de I e $R(f)$ um resultado aproximado de I o usuário deseja que

$$(3a) \quad |I(f) - R(f)| \leq \epsilon_1 \quad \text{ou} \quad (3b) \quad \frac{|I - R(f)|}{|R(f)|} \leq \epsilon_2$$

onde ϵ_1 é a tolerância para o erro absoluto e ϵ_2 é a tolerância para o erro relativo. Em qualquer caso, o programa fornece o erro "teórico" que foi obtido e que é usado com controle da saída.

A rotina pode decidir que a exatidão prescrita não é atingível dentro do limite L , faz o melhor que pode e retorna com um resultado confiável ou um resultado parcial com alguma mensagem. É sempre possível achar integrandos que "enganam" a rotina produzindo um resultado errado, mas para as boas rotinas a classe de tais exemplos deve ser tão pequena quanto possível sem complicar em demasia a lógica da rotina fazendo-a ineficiente em casos razoáveis.

Para entendermos melhor as diferentes filosofias adaptativas precisamos de talhar os componentes básicos de uma rotina adaptativa segundo RICE [2]. São eles processador de intervalos; a coleção de intervalos; o gerenciador da coleção de intervalos e o controlador de algoritmo.

O processador de intervalos faz os cálculos necessários para prover as aproximações para o valor da integral através das regras de integração e calcula os limites de erro associados a estes intervalos. Aqui podemos ter:

- uma única regra aplicada a todo intervalo e a cada metade
- regras independentes da mesma ordem
- regras de ordem diferentes, e outras

O gerenciador da coleção de intervalos cria e mantém a coleção de intervalos numa determinada estrutura que pode ser uma pilha; uma lista ordenada (neste caso os intervalos são ordenados de acordo com os valores das estimativas de erro); uma fila; uma fila onde intervalos com estimativas de erro são descartados ou caixas.

O controlador do algoritmo pode ter dois componentes distintos: o estimador do limite de erro que é envolvido na decisão de terminar o algoritmo e uma coleção de procedimentos para detectar tipos especiais de comportamento como singularidades, descontinuidades e problemas com erro de arredondamento. Variando os elementos de cada componente podemos ter um número muito grande de algoritmos diferentes. É bastante difícil identificar dentre todos estes algoritmos quais são os realmente superiores. Para esta análise utilizam-se 5 ferramentas que iremos apenas citá-las.

Análise de erro e convergência, estudo da complexidade computacional; análise dos componentes dos algoritmos adaptativos; estudos da população de integrandos e estudos experimentais. Tais análises fogem ao objetivo des

te trabalho. Na próxima seção veremos uma fundamentação simples sobre as rotinas adaptativas.

FUNDAMENTAÇÃO (3)

A base da idéia adaptativa é subdividir o intervalo de integração num número maior de subintervalos onde a variação de f é mais acentuada. Para que se faça isso automaticamente tomamos duas aproximações da integral num subintervalo e as comparamos para obter uma estimativa da sua exatidão. Se esta é aceitável uma das duas é tomada como valor da integral sobre o intervalo. Se a exatidão não é aceitável o subintervalo é dividido em duas ou mais partes e o processo é repetido nos subintervalos menores. Para reduzir o número de avaliações da função usaremos a bissecção do intervalo de modo que a segunda regra pode ser calculada usando os pontos utilizados pela primeira regra. Vamos aplicar uma única regra aplicada em todo subintervalo que vamos denotá-lo por $[x_i, x_{i+1}]$. A numeração é arranjada de modo que $x_1 = a$ e a rotina determina n tal que, no final do processo, $x_{n+1} = b$. Seja h_i o tamanho de cada intervalo, ou seja $h_i = x_{i+1} - x_i$. Vamos denotar por P_i a aplicação da regra no subintervalo $[x_i, x_{i+1}]$ e Q_i a soma da aplicação mesma em cada metade deste subintervalo. Tanto P_i como Q_i são aproximações para

$$(4) \quad I_i = \int_{x_i}^{x_{i+1}} f(x) dx$$

Vamos aplicar regras newtonianas para o cálculo de P_i de modo que P_i dê a resposta exata se o integrando é um polinômio de grau $p-1$, ou seja, que $f^{(p)}(x)$ é identicamente nula. Pode ser mostrado que

$$(5) \quad I_i - P_i = c h_i^{p+1} f^{(p)}\left(x_i + \frac{h_i}{2}\right) + \dots$$

Como Q_i é a soma dos dois P_i 's de subintervalos de comprimento $h_i/2$ segue que:

$$(6) \quad I_i - Q_i = c \left(\frac{h_i}{2}\right)^{p+1} \left[f^{(p)}\left(x_i + \frac{h_i}{2}\right) + f^{(p)}\left(x_i + \frac{3h_i}{4}\right) \right] + \dots$$

desde que

$$(7) \quad f^{(p)}\left(x_i + \frac{h_i}{4}\right) + f^{(p)}\left(x_i + \frac{3h_i}{4}\right) = 2f^{(p)}\left(x_i + \frac{h_i}{2}\right) + \dots$$

e reescrevendo (6), usando (5) e (7) temos

$$(8) \quad I_i - Q_i = \frac{1}{2^p} (I_i - P_i),$$

indicando que, bisseccionando o intervalo o erro diminui por um fator de 2^p . Resolvendo para a incôgnita I_i e rearranjando temos:

$$(9) \quad Q_i - I_i = \frac{1}{2^{p-1}} (P_i - Q_i),$$

em outras palavras, o erro em Q_i é aproximadamente $\frac{1}{2^{p-1}}$ vezes a diferença entre duas aproximações.

Portanto se queremos que $I(f) - R(f) < \epsilon$ e se tomamos

$$(10) \quad R(f) = \sum_{i=1}^n Q_i$$

teremos que a subdivisão de um subintervalo em outros subintervalos menores deve ser feita até que se atinja.

$$(11) \quad |p_i - Q_i| < \frac{(2^{p-1})h_i \epsilon}{b-a} = F(\epsilon, p, a, b, h)$$

onde ϵ é o critério de exatidão fornecido. Mostramos agora que se (11) é válida em todos subintervalos então (3a) é satisfeita. Para isso vejamos.

$$\begin{aligned} |R(f) - I(f)| &= \left| \sum_{i=1}^n (Q_i - I_i) \right| \\ &\leq \sum_{i=1}^n |Q_i - I_i| \end{aligned}$$

que por (9) ignorando os termos de mais alta ordem temos

$$|R(f) - I(f)| \leq \frac{1}{2^{p-1}} \sum_{i=1}^n |P_i - Q_i|$$

Como (11) é satisfeita em cada subintervalo temos

$$(12) \quad |R(f) - I(f)| \leq \frac{1}{2^{p-1}} \cdot \frac{(2^{p-1})}{b-a} \epsilon \cdot \sum_{i=1}^n h_i \leq \epsilon.$$

Esta análise requer que $f^{(p)}(x)$ seja contínua e que o erro de truncamento seja proporcional a $h_i^{(p+1)}$ $f^{(p)}(x)$. Esta condição nem sempre será satisfeita para todo integrando, contudo conforme for implementado o controlador do algoritmo poderemos ter um resultado confiável atingindo a exatidão desejada.

IMPLEMENTAÇÃO (4)

Como esta é uma rotina didática e não competitiva com as demais existentes usaremos o que há de mais simples em cada um dos componentes de um al

goritmo adaptativo citado em 2. O processador de intervalos utilizará a regra de Simpson simples aplicada em todo intervalo $[x_i, x_{i+1}]$ e a mesma regra composta aplicada em cada metade do subintervalo.

Teremos então

$$(13) P_i = \frac{h_i}{6} [f(x_i) + 4f(x_i + \frac{h_i}{2}) + f(x_i + h_i)]$$

e

$$(14) Q_i = \frac{h_i}{12} [f(x_i) + 4f(x_i + \frac{h_i}{4}) + 2f(x_i + \frac{h_i}{2}) + 4f(x_i + \frac{3h_i}{4}) + f(x_i + h_i)]$$

que pode ser visualizado na seguinte figura 2.

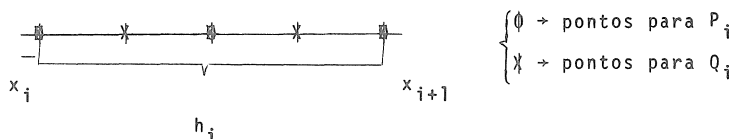


Figura 2

Como para a regra de Simpson $p = 4$ então o estimado de limite de erro será dado por (11) ou seja, cada intervalo será bisseccionado enquanto

$$(15) |P_i - Q_i| > \frac{16 h_i \epsilon}{(b-a)}$$

Nesta rotina não forneceremos um limite de erro para cada subintervalo mas ele pode ser facilmente obtido. Esta rotina não tem nenhum mecanismo especial para detectar singularidades ou erros de arredondamentos. Toda vez que (11) não é satisfeita então o subintervalo, assim como as informações já obtidas são colocados numa coleção de intervalos que obedece a uma estrutura de pilha. Esta estrutura é finita e pode armazenar um número pre-determinado de subintervalos que não atingiram (11). Então é necessário um controle no número de subdivisões feitas para não causar overflow na pilha. Vamos especificar que o número máximo de subdivisões permitidas é N . Tal número deve ser escolhido em função da precisão que a máquina onde será implementado o algoritmo pode oferecer. Outro fator a ser considerado é que embora tal limite possa não ser atingido é preciso limitar o número de avaliações da função que será indicado por L . Vejamos agora o esquema simplificado para uma rotina adaptativa.

1. entrada $\{f, a, b, N, L, \epsilon\}$
2. inicializa: $h_i \leftarrow b-a$; $e \leftarrow a$; $x_0 \leftarrow a$; $R \leftarrow 0$; $Q \leftarrow 0$.
3. calcular P_i e Q_i
4. se $|P_i - Q_i| < F(\epsilon, p, a, b)$

então $\left\{ \begin{array}{l} Q \leftarrow Q + Q_i \\ \text{se a pilha está vazia} \\ \quad \text{então } R \leftarrow Q \text{ e pãra} \\ \quad \text{senão desempilha e volta a 2} \end{array} \right.$

Senão $\left\{ \begin{array}{l} \text{empilha } [e + \frac{h_i}{2}, e + h_i] \\ h_i \leftarrow h_i/2 \\ \text{volta a 2.} \end{array} \right.$

A rotina INARS (Integração Numérica Aadaptativa pela Regra de Simpson) implementada em PASCAL no IBM-4341 é dada no final deste trabalho.

DESEMPENHO (5)

Vejamos 3 exemplos que mostram o desempenho adaptativo da rotina INARS veremos exemplos com um número pequeno de subdivisão para não tornar cansativo os exemplos.

Exemplo 1 $\int_0^1 \frac{1}{1+x} dx$ cujo gráfico é dado na figura 3

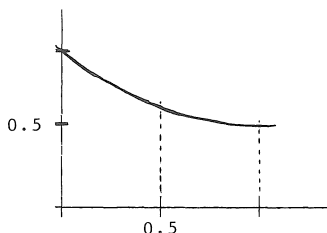


Figura 3

e o valor exato é dado por $\ell_n^2 - \ell_n^1 = 0.693\ 147\ 180\ 6$

Executando passo a passo a rotina temos:

$h = 1.0$, $N = 0$, $E = 0.0$, $F(\epsilon, p, a, b, h) = 15 \epsilon h_i$
 $P = 6.944\ 444\ 4E-1$

$$Q = 6.932\ 539\ 6\ E-1$$

$$|P-Q| = 0.001\ 190\ 48 \quad \dagger \quad 15 * 1.0\ E-1 = 0.000\ 15$$

empilha: [0.5, 1.0]

$$h = 0.5, \quad E = 0.0 \quad P = 4\ 055\ 555\ 5\ E-1 \quad Q = 4.054\ 713\ 8\ E-1$$

$$|P-Q| = 0.000\ 084\ 17 \quad \dagger \quad 15 * 0.5 * 1.0\ E-1 = 0.000\ 075$$

empilha [0.25, 0.5]

$$h = 0.25, \quad E = 0.0, \quad P = 2.231\ 481\ 4\ E-1, \quad Q = 2.231\ 438\ 4\ E-1$$

$$|P-Q| = 0.000\ 004\ 34 \quad < \quad 15 * 0.25 * 1.0\ E-5 = 0.000\ 037\ 5$$

$$R = 2.231\ 438\ 4\ E-1$$

desempilha [0.25, 0.5]

$$h = 0.25, \quad E = 0.25 \quad P = 1.823\ 232\ 3\ E-1$$

$$Q = 1.823\ 216\ 6\ E-1$$

$$|P-Q| = 0.000\ 001\ 57 \quad < \quad 15 * 0.25 * \epsilon = 0.000\ 037\ 5$$

$$R = 0.405\ 465\ 5$$

desempilha:

$$h = 0.5 \quad e = 0.5 \quad P = 2.876\ 984\ 12\ E-1$$

$$Q = 2.876\ 838\ 5\ E-1$$

$$|P-Q| = 0.000\ 015\ 162 \quad < \quad 15 * 0.5 * \epsilon = 0.000\ 075$$

$$R = 0.693\ 148\ 65$$

E daí temos que $|R(f) - I(f)| = 1.46\ E-6 < 1.0\ E-5$.

Tal resultado foi obtido com 13 avaliações do integrando f , conforme está representado graficamente na figura 4, o que está de acordo com o esperado.

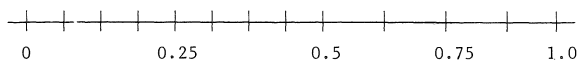
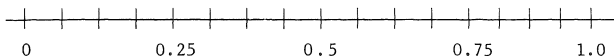


Figura 4

Para obter a mesma exatidão com uma rotina não-adaptativa precisamos de 17 avaliações de f .

Exemplo 2 $\int_0^1 \sin \pi x \, dx$

Neste caso não faremos os cálculos detalhadamente mostrando apenas na figura 5 a distribuição dos pontos no intervalo $[0, 1]$, que é uniforme pois f é suave em toda extensão de $[0, 1]$.



O resultado exato é $2/\pi = 0.636\ 619\ 772\ 4$ enquanto que o resultado aproximado é $0.636\ 625\ 064$.

Exemplo 3 $\int_0^1 \sqrt{x} \, dx$

Neste caso, temos que f não tem primeira derivada limitada, no entanto, o resultado aproximado obtido é 0.666 663 238 com 61 avaliações do integrando f pois esta função tem um crescimento muito rápido em torno do ponto zero onde f' não é definida. O menor valor de h_i foi de 6.103 52 E-5.

Neste caso com uma quadratura fixa precisaríamos de um número de avaliações de f muito maior.

BIBLIOGRAFIA

- |1| FORSYTHE, G.E.; MALCOLM, M.A. & MOLER C.B. Computer methods for mathematical computations. Prentice-Hall, INC, 1977.
- |2| RICE, J.A. Metalgorithm for adaptative quadrature. J.ACM 22, 1(Jan 1975) 61-82.
- |3| LYNESS, J.N. Notes on the adaptative Simpson quadrature routine. J.ACM 16, (July 1969), 483-495.

```

PROCEDURE INARS(FUNCTION F(P REAL) REAL
  A,P,EPS REAL
  VAR RES REAL
  NF INTEGER)
VAR E, PHI,P,X , P , C REAL
N,NMAX,NFMAX,I,K INTEGER
CONTINUA BOOLEAN
XI
XAXX APPAY (.1..5.) CF REAL
FAUX APPAY (.1..20.) OF REAL
FAUX APPAY (.1..42.) OF REAL
(*
* INICIALIZAÇÃO
*)
BEGIN
WRITELN('CALCULO COM EPSILON =',EPS)
E = A
RES = 0.0
NF = 5
N = 0
H = E - A
NMAX = 200
NFMAX = 1000
PHI = 15.0 * EPS / (P - A)
CONTINUA = TRUE
FOR I = 1 TO 5 DO
  BEGIN
    X = E + (I - 1) * H / 4
    XI(I.) = F(X)
  END
(*
* LAÇO PRINCIPAL
*)
WHILE CONTINUA DO
  BEGIN
    (*
    * CALCULO DAS APROXIMAÇÕES
    *)
    P = (H/6) * (XI(1.) + 4 * XI(2.) + XI(5.))
    Q = (H/12) * (XI(1.) + 4 * XI(2.) + 2 * XI(3.) + 4 * XI(4.) +
      XI(5.))
    WRITELN(' H = ',H 15 10, ' E = ',E 15 10, ' P = ',P, ' Q = ',Q)
    (*
    * VERIFICA SE SUBINTERVALO CONVERGIU
    *)
    IF ABS(P - Q) < PHI * H THEN
      BEGIN
        (* ACUMULA RESULTADOS PARCIAIS *)
        RES = RES + Q
        WRITELN(' NIVEL = ',K 5, ' RES. PARC = ',RES)
        (*
        * VERIFICA SE PILHA ESTA VAZIA
        *)
        IF N = 0 THEN
          (*
          * PILHA ESTA VAZIA
          *)
          BEGIN
            (*
            * FIM NORMAL
            *)
            WRITELN('RESULTADO = ',RES, ' COM ',NF 5, ' AVALIAÇÕES ',
              'DE F(X) ')
            CONTINUA = FALSE
          END
          FALSE
        BEGIN
          (*
          * PILHA NAO ESTA VAZIA
          *)
          (*
          * DESEMPILHA
          *)
          WRITELN('DESEMPILHA ')
          WRITELN(' 15 10, ' H = ',H 15 10, ' XINIC = ',E + H)
          F = E + H
          H = XAXX(K.) - E
          XI(1.) = XI(5.)
          XI(3.) = FAUX(N.)
          XI(5.) = FAUX(N-1.)
          N = N - 2
          K = K DIV 2

```

```

(**
* VERIFICA SE ATINGIU NUMERO MAXIMO DE AVALIACOES *
*)
IF NF NFMAX THEN
  BEGIN
    WRITELN('EXCEDEU ',NFMAX 5,'AVALIACOES DE',
    ' DE F(X) ')
    CONTINUA =FALSE
  END
  ELSE
    BEGIN
      (*
      * REFAZ CALCULOS *
      *)
      NF =NF+2
      X =E+H/4
      XI(.2.) =F(X)
      X =E+3*H/4
      XII(.4.) =F(X)
    END
  END
END
ELSE
  BEGIN
    (*
    * SUBINTERVALO NAO CONVERGIU *
    *
    * VERIFICA SE ATINGIU NUMERO MAXIMO DE SUBDIVISOES *)
    IF (N DIV 2) NMAX THEN
      BEGIN
        WRITELN('EXCEDEU NUMERO MAXIMO DE SUBDIVISOES')
        CONTINUA =FALSE
      END
      ELSE
        BEGIN
          (*
          * NAO ATINGIU NUMERO MAXIMO DE SUBDIVISOES *
          *
          * EMPILHA *
          *)
          WRITELN('EMPILHA')
          N =N+2
          K =N DIV 2
          FAUX(.N.) =XI(.4.)
          FAUX(.N-1.) =XI(.5.)
          XAUX(.K.) =E+H
          H =H/2
          (*
          * TRANSLADA *
          *)
          XI(.5.) =XI(.3.)
          XII(.2.) =XI(.2.)
          (*
          * VERIFICA SE EXCEDEU NUMERO MAXIMO DE AVALIACOES *
          *)
          IF NF NFMAX THEN
            BEGIN
              WRITELN('EXCEDEU ',NFMAX 5,'AVALIACOES',
              ' DE F(X) ')
              CONTINUA =FALSE
            END
            ELSE
              BEGIN
                (*
                * REFAZ CALCULOS *
                *)
                NF =NF+2
                X =E+H/4
                XII(.2.) =F(X)
                X =E+3*H/4
                XII(.4.) =F(X)
              END
            END
          END
        END
      END
    END
  END
  (* WHILE *)
END (* INITS *)

```