



OBTENÇÃO AUTOMATIZADA DA EXPRESSÃO DO ERRO DE TRUNCAMENTO PRESENTE NA DISCRETIZAÇÃO DE EQUAÇÕES DIFERENCIAIS PARCIAIS

Vanessa Portioli Rolnik

Universidade de São Paulo, Departamento de Engenharia Mecânica
Av. Dr. Carlos Botelho, 1465 São Carlos-SP CEP 13560-970
Telefone:(016) 273-9401 e-mail: vanessa@sc.usp.br

Armando de Oliveira Fortuna

Universidade de São Paulo, Departamento de Ciências de Computação e Estatística
Av. Dr. Carlos Botelho, 1465 São Carlos-SP CEP 13560-970
FAX (016)273-9751 telefone: (016)273-9660
e-mail: fortuna@lcad.icmc.sc.usp.br

***Resumo.** A equação modificada introduzida por Hirt nos anos 60, ainda é uma das ferramentas mais importantes para analisar os termos do erro de truncamento que surge da discretização de equações diferenciais parciais. Poucos anos depois Warming e Hyett formalizaram o conceito proposto por Hirt. Apesar disso, as características básicas do algoritmo, que requer extensa manipulação algébrica da equação discretizada, permaneceram as mesmas. As operações repetitivas que compõem o algoritmo fazem deste um candidato para automatização utilizando um computador. Pelo fato das operações necessitarem de manipulações simbólicas e diferenciações, não é conveniente a implementação do algoritmo em linguagens como C, Fortran ou Pascal. Para esse algoritmo, em particular, as linguagens apropriadas são aquelas capazes de realizar as manipulações simbólicas necessárias, como por exemplo, a linguagem interna do pacote Mathematica. Este trabalho descreve o pacote EQMOD, que é uma implementação do algoritmo de Hirt para obtenção da equação modificada. EQMOD está escrito na linguagem de programação interna do Mathematica. Alguns exemplos são apresentados para mostrar sua funcionalidade e aplicabilidade.*

***Palavras-chave:** EQMOD, Equações de diferenças finitas, Equações diferenciais parciais, Erro de truncamento, Equação modificada.*

1. INTRODUÇÃO

Equações diferenciais parciais (EDPs) regem importantes fenômenos físicos tais como propagação da onda, condução de calor e escoamento de fluido. Na maioria dos problemas práticos de engenharia, a determinação de soluções analíticas para essas EDPs não é viável. Assim sendo, métodos numéricos são frequentemente utilizados na solução dessas equações.

Um método numérico bastante utilizado para a resolução de problemas de equações diferenciais parciais é o método das diferenças finitas, pelo qual as derivadas parciais das EDPs são escritas em função dos valores de pontos discretos, na região onde se deseja encontrar a solução numérica da equação diferencial parcial. A relação entre os valores nesses pontos discretos é dada através de expressões algébricas. Por sua vez, essas expressões derivam de manipulações de expansões em série de Taylor da variável dependente. Obtém-se assim uma equação de diferenças finitas (EDF). Para a resolução numérica de uma EDF, utiliza-se métodos diretos ou, em geral, iterativos.

Uma consequência direta do processo de discretização de uma EDP é a presença de *erros de truncamento*. Estes surgem devido ao truncamento das expansões em séries de Taylor na discretização da EDP. Os erros de truncamento podem afetar significativamente a solução numérica da equação discretizada. Porém, com uma análise da expressão do erro de truncamento, informações podem ser obtidas sobre a ordem e o tipo de erro que irá predominar na solução numérica da EDP. Termos dominantes do erro de truncamento com derivadas de ordem par irão gerar erros do tipo *dissipativos*. Em engenharia, principalmente na área de mecânica dos fluidos computacional, erros dissipativos são conhecidos como "difusão numérica" ou "difusão artificial", uma vez que eles são equivalentes à termos difusivos. A presença desses erros pode comprometer significativamente a qualidade da solução numérica, já que a difusão numérica pode ser maior que a difusão real do fluido. Por outro lado, os erros *dispersivos* estão associados a termos dominantes com derivadas de ordem ímpar. Analisando o coeficiente do termo dominante do erro de truncamento é possível minimizar seus efeitos na solução numérica. Tal análise só é possível após a determinação da equação modificada.

A expressão do erro de truncamento está presente na *equação modificada*. A equação modificada é uma equação diferencial parcial composta da equação diferencial parcial original mais a expressão do erro de truncamento relativa a discretização escolhida e pode ser determinada através do *algoritmo da equação modificada*.

Introduzido em 1974 por (Warming&Hyett,74), o algoritmo da equação modificada é direto, porém, requer extensas manipulações algébricas, o que dificulta a obtenção manual da expressão do erro de truncamento. Estas manipulações também dificultam a implementação do algoritmo em linguagens como C, Fortran e Pascal. Decidiu-se pelo pacote Mathematica (Wolfram, 91) que possui linguagem de manipulação simbólica.

Como exemplo de que a equação modificada está sendo atualmente utilizada na solução de problemas práticos, pode-se citar:

- determinação de equações de diferenças finitas com maior ordem de precisão (Shubin&Bell,87) (Tuomela,95);
- estudo de esquemas de diferenças finitas não lineares (Goodman&Majda,85);
- explicação da instabilidade observada em uma aplicação do método MAC (Marker and Cell) (Hirt,68).

Outros pesquisadores já desenvolveram aplicações da equação modificada em computador. Uma dessas aplicações é a implementação de um sistema que analisa a estabilidade de equações de diferenças finitas através do método de von Neumann ou da equação modificada (Liska, 84). Porém, esse pacote trata apenas de equações com duas variáveis independentes, além de outras restrições. Decidiu-se implementar o algoritmo da equação modificada com menos restrições.

Este trabalho descreve o pacote EQMOD que é a implementação do algoritmo da equação modificada, implementado na linguagem interna do Mathematica e baseado no algoritmo proposto por Hirt. O programa EQMOD determina a equação modificada começando da equação de diferenças finitas que é fornecida pelo usuário. O programa não só calcula a

equação modificada como que, opcionalmente exibe as operações efetuadas durante sua execução, tornando, assim, uma ferramenta útil no ensino do método da equação modificada.

A continuação deste artigo está dividida como se segue: Seção 2 apresenta o algoritmo da equação modificada enquanto que a Seção 3 mostra seu cálculo em um problema modelo. A Seção 4 descreve o programa EQMOD, seus dados de entrada e saída e a funcionalidade do programa como ferramenta de pesquisa e ensino. A Seção 5 contém exemplos e resultados e a Seção 6 apresenta as conclusões.

2. DESCRIÇÃO DO ALGORITMO

A equação modificada é obtida através de um procedimento de dois passos (Warming&Hyett,74):

1. expansões em série de Taylor de cada termo da equação discretizada e
2. eliminação das derivadas temporais (se existirem) presentes na expressão do erro de truncamento.

Esses passos são mostrados abaixo para o caso de uma EDP dependente do tempo em uma dimensão espacial. A extensão para problemas três com dimensões espaciais é direta.

2.1 Expansões em série de Taylor

O primeiro passo é expandir cada termo da equação discretizada em série de Taylor em torno da função incógnita u_i^n , isto é, em torno do ponto $(x = i\Delta x, t = n\Delta t)$ da malha (u é a função incógnita). Após todas as expansões terem sido feitas, elas são substituídas na equação discretizada resultando em uma equação diferencial parcial, aqui denominada **Eq1**, que contém derivadas espaciais e temporais.

2.2 Eliminação das derivadas temporais

O segundo passo consiste em eliminar as derivadas temporais de **Eq1**, com exceção daquelas presentes na EDP original. Essas eliminações são necessárias para analisar os efeitos do erro de truncamento na dimensão espacial. Para a eliminação das derivadas temporais existe um procedimento análogo à eliminação gaussiana, que consiste em:

- selecionar o termo que deve ser eliminado em **Eq1**;
- encontrar um “operador” adequado que aplicado sobre **Eq1**, elimina o termo selecionado;
- somar o resultado à **Eq1**.

Esse processo de eliminação resulta em uma nova equação diferencial parcial, **Eq2**, que não mais contém o termo selecionado. Este passo pode ser resumido através da expressão **Eq2** = **Eq1** + “operador”[**Eq1**]

O operador é determinado pelo negativo do coeficiente do termo que está sendo eliminado, multiplicado por uma derivada conveniente, de maneira que quando a equação resultante for somada à **Eq1**, o termo escolhido seja eliminado. Por exemplo, suponha-se que a

EDP contenha a derivada temporal: $\frac{\partial^m u}{\partial t^m}$ e que o termo que se deseja eliminar é do tipo

$$A \frac{\partial^p u}{\partial t^q \partial x^{p-q}}, \text{ com } p > m. \text{ O operador utilizado para eliminar tal termo é: } -A \frac{\partial^{p-m} u}{\partial t^{q-m} \partial x^{p-q}}$$

É importante observar que a EDP original não deve ser usada para eliminar as derivadas temporais, pois, em geral, sua solução não satisfaz a EDF, devido aos erros de truncamento.

Assim, para eliminar-se as derivadas temporais, deve-se usar a equação diferencial parcial obtida no primeiro passo (**Eq1**) (Warming&Hyett,74).

Como a expressão do erro de truncamento possui derivadas de ordem arbitrária da função incógnita u , o número de termos de uma equação modificada varia conforme a diferenciabilidade dessa função. Com isso, a equação modificada pode ter infinito número de termos. Porém, na prática, somente os termos iniciais (dominantes) são necessários para uma análise adequada, já que os termos com derivadas de ordens superiores são multiplicados por coeficientes $(\Delta x)^n$ ou $(\Delta t)^n$. Quando Δx e Δt forem menores que um, esses coeficientes tendem a zero rapidamente, reduzindo o valor do respectivo termo (supondo que as derivadas sejam limitadas). Assim, a aplicação do algoritmo para obtenção da equação modificada pára após a determinação dos primeiros termos da expressão do erro de truncamento. A obtenção da E.M. pode ser resumida nos passos abaixo. Dada uma equação de diferenças finitas que aproxima uma EDP:

1. expanda cada termo da EDF em série de Taylor, obtendo uma equação diferencial parcial, $Eq(i=1) = Eq1$;
2. encontre *termo(i)* de $Eq(i)$ aquele que contém a derivada temporal de menor ordem de $Eq(i)$, (sem considerar aquelas que estão presentes na EDP original);
3. determine o *operador(i)* conveniente para eliminar o *termo(i)*;
4. faça $Eq(i+1) = Eq(i) + \text{operador}(i)[Eq1]$;
5. se $Eq(i+1)$ ainda contém derivadas temporais além das originalmente presentes na EDP, volte ao passo 2;
6. *Equação Modificada* = $Eq(i+1)$.

3. APLICAÇÃO DO ALGORITMO

Esta seção ilustra a processo descrito acima para obtenção da equação modificada. Considere a equação da onda de primeira ordem:

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, \quad a > 0 \quad (3.1)$$

Existem diversas aproximações por diferenças finitas para a equação acima. Neste exemplo será usado o método de Lax-Wendroff de segunda ordem no tempo e no espaço. A aplicação desse método à equação acima resulta na seguinte equação de diferenças finitas:

$$u_j^{n+1} - u_j^n + a \frac{\Delta t}{2\Delta x} (u_{j+1}^n - u_{j-1}^n) - \frac{1}{2} \left(a \frac{\Delta t}{\Delta x} \right)^2 (u_{j+1}^n - 2u_j^n + u_{j-1}^n) = 0 \quad (3.2)$$

Expandindo u_j^{n+1} , u_{j+1}^n e u_{j-1}^n em série de Taylor em torno de u_j^n , substituindo as expansões na Eq. (3.2) e simplificando o resultado, tem-se:

$$\begin{aligned} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} + \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} - a^2 \frac{\Delta t}{2} \frac{\partial^2 u}{\partial x^2} + \frac{(\Delta t)^2}{6} \frac{\partial^3 u}{\partial t^3} + a \frac{(\Delta x)^2}{6} \frac{\partial^3 u}{\partial t^3} + \\ + \frac{(\Delta t)^3}{24} \frac{\partial^4 u}{\partial t^4} - a^2 \frac{(\Delta t)(\Delta x)^2}{24} \frac{\partial^4 u}{\partial t^4} + \dots = 0 \end{aligned} \quad (3.3)$$

O próximo passo é a eliminação dos termos contendo derivadas temporais que aparecem na Eq. (3.3), com exceção de $\frac{\partial u}{\partial t}$ (pois este termo pertence à EDP original), pelo procedimento descrito anteriormente. O processo de eliminação se inicia pelo termo $\frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2}$, pois este termo contém a derivada temporal de menor ordem (dentre as que serão eliminadas). Para eliminar este termo, aplica-se o operador $-\frac{\Delta t}{2} \frac{\partial}{\partial t}$ à Eq. (3.3) e, em seguida, soma-se o resultado à Eq. (3.3), obtendo-se:

$$\begin{aligned} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} - a \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t \partial x} - a^2 \frac{\Delta t}{2} \frac{\partial^2 u}{\partial x^2} - \frac{(\Delta t)^2}{12} \frac{\partial^3 u}{\partial t^3} + a^2 \frac{(\Delta t)^2}{4} \frac{\partial^3 u}{\partial t \partial x^2} + \\ a \frac{(\Delta x)^2}{6} \frac{\partial^3 u}{\partial x^3} - \frac{(\Delta t)^3}{24} \frac{\partial^4 u}{\partial t^4} - a^2 \frac{(\Delta t)(\Delta x)^2}{12} \frac{\partial^4 u}{\partial t \partial x^3} - a^2 \frac{(\Delta t)(\Delta x)^2}{24} \frac{\partial^4 u}{\partial x^4} + \dots = 0 \end{aligned} \quad (3.4)$$

Observe que a Eq. (3.4) já não contém o termo $\frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2}$. A operação acima pode ser representada pela expressão Eq. (3.4) = Eq. (3.3) + $\left(-\frac{\Delta t}{2}\right) \frac{\partial}{\partial t}$ [Eq. (3.3)].

O procedimento apresentado para a eliminação das derivadas temporais é aplicado continuamente até que a ordem desejada para as derivadas espaciais no erro de truncamento seja alcançada. Os termos remanescentes da expressão do erro de truncamento são descartados. Para o método de Lax-Wendroff de segunda ordem, dado pela Eq. (3.2), obtém-se:

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = -\frac{a}{6} [(\Delta x)^2 - a^2 (\Delta t)^2] \frac{\partial^3 u}{\partial x^3} - \frac{a^2 \Delta t}{8} [(\Delta x)^2 - a^2 (\Delta t)^2] \frac{\partial^4 u}{\partial x^4} + \dots \quad (3.5)$$

Observe que o lado esquerdo da Eq. (3.5) é a equação original (equação da onda - 3.1) e lado direito corresponde ao erro de truncamento introduzido no processo de discretização da EDP. A Eq. (3.5) é conhecida como *Equação Modificada*. A correspondente “saída” (*output*) do programa EQMOD é mostrada na Fig. 1.

```
variables: {j, dx, n, dt}
order of the last derivative in the M.E.: 4
detail level: 1
maximum number of eliminations:100
-----
Modified Equation
(0,1)      (1,0)      3 2      2      (3,0)      4 3 2 2      (4,0)
u      + a u      = (----- - -----) u      + (----- - -----) u      +
          6          6          8          8
      5 4 3 2 2      4
      a dt a dt dx a dx      (5,0)
      (----- - ----- - -----) u      + ...
          20          24          120

Total number of eliminations 15
```

Figura 1: Tela de saída do programa EQMOD

Observe que os resultados fornecidos pelo código não estão na notação usual matemática. Infelizmente, a versão do Mathematica usada para implementação do programa EQMOD não permite a conversão do resultado fornecido pelo programa para notação usual. Em todos os exemplos mostrados neste artigo, $u^{(x_1, x_2)}$ significa que a função incógnita u foi derivada x_i vezes em relação a variável i . No exemplo acima, $u^{(3,0)}$ significa $\frac{\partial^3 u}{\partial x^3}$. A ordem das variáveis independentes na função incógnita é a mesma fornecida pelo usuário nos dados de entrada (vide Seção 4), mas essa ordem não faz diferença para o programa.

4. PROGRAMA EQMOD

O programa EQMOD determina a equação modificada de EDP para uma dada discretização e foi implementado na linguagem de programação interna do pacote Mathematica. A escolha do Mathematica deve-se ao fato desse estar disponível para uma grande variedade de computadores (portanto aumenta a portabilidade) e possui todos os recursos necessários para essa aplicação, como por exemplo manipulações algébricas e derivações. O usuário deve fornecer os seguintes dados ao programa EQMOD:

- equação de diferenças finitas;
- variáveis: constantes e incógnitas. A ordem das variáveis não é relevante para a execução do programa mas deve ser observada para posterior leitura do resultado (devido a notação de derivadas parciais usada pelo Mathematica);
- ordem da derivada espacial de maior ordem na equação modificada;
- nível de detalhamento (1,2 ou 3) (vide abaixo);
- número máximo de eliminações permitido.

Os itens acima são solicitados ao usuário através de janelas. A equação de diferenças finitas é escrita no formato da linguagem de programação. Veja Fig. 2. O programa EQMOD pode imprimir os resultados na tela, ou, então, em arquivos no formato Mathematica (arquivo .m) ou Latex (arquivo .tex).

Durante a execução do programa EQMOD, o usuário escolhe a quantidade de informação (nível de detalhamento) que ele quer visualizar durante o cálculo da equação modificada. Esta opção torna o programa EQMOD uma ferramenta não só para pesquisa mas também para o ensino e demonstração do cálculo da equação modificada. São três os níveis de detalhamento apresentados pelo programa EQMOD:

- Nível 1: exhibe somente os dados fornecidos pelo usuário e a equação modificada encontrada;
- Nível 2: exhibe o mesmo que o nível 1 mais a EDP obtida após expansões em série de Taylor e os termos eliminados;
- Nível 3: exhibe o mesmo que o nível 2 mais textos explicativos sobre cada passo executado pelo programa.

Os níveis 2 e 3 permitem ao usuário acompanhar passo a passo a execução do programa. Na Seção 5 pode-se ver um exemplo com nível de detalhamento 3.

5. EXEMPLOS

Os exemplos abaixo ilustram algumas das capacidades do programa EQMOD. O programa encontra a equação modificada para EDPs lineares com um número arbitrário de dimensões espaciais e que contenham (ou não) derivada de qualquer ordem em relação ao

tempo. As discretizações para as derivadas temporais podem ser implícitas ou explícitas. Porém, a versão atual do programa não calcula a equação modificada de EDPs não lineares e nem sistemas de equações. Os exemplos abaixo foram obtidos pelo programa EQMOD com opção de nível de detalhamento igual a 1.

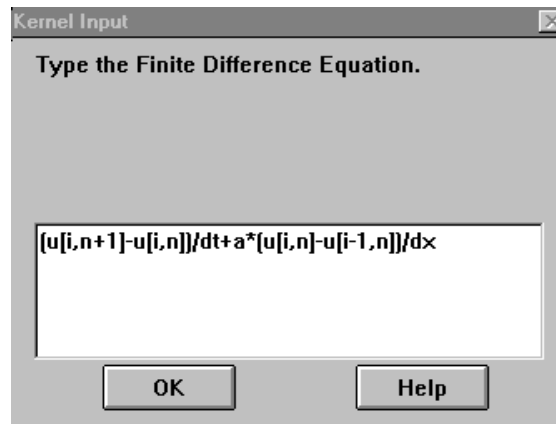


Figura 2: Janela para o fornecimento da equação de diferenças finitas

Considere a equação da convecção em uma dimensão dada por:

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, \quad a > 0 \quad (5.1)$$

discretizada pelo seguinte esquema de diferenças finitas:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a \frac{u_j^n - u_{j-1}^n}{\Delta x} = 0 \quad (5.2)$$

A Eq. (5.2) é conhecida como “upwind”. A Fig. 3 mostra o resultado do programa quando a equação acima é fornecida (a Fig. 2 mostra como a Eq. (5.2) deve ser fornecida). Como esperado, a saída do programa EQMOD indica a primeira ordem (no espaço e no tempo) da discretização (veja o primeiro termo da expressão do erro de truncamento). A *consistência* da discretização também pode ser verificada, observando que o erro de truncamento vai para zero quando dx , dt tendem para zero. O termo de menor ordem do erro de truncamento é uma derivada segunda, sugerindo que os resultados numéricos serão afetados por difusão artificial, que é realmente o caso (Hoffmann, 89). Pode-se verificar, com um pouco de álgebra, que, para $a \frac{dt}{dx} = 1$ (número de Courant), o erro de truncamento se anula. Para a equação de Laplace discretizada pela fórmula de 5 pontos, o programa EQMOD fornece a equação modificada mostrada na Figura 4.

O programa EQMOD também trata discretizações implícitas. Considere a equação parabólica em duas dimensões aproximada por uma fórmula de diferenças finitas totalmente implícita

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \alpha \left(\frac{u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{(\Delta y)^2} \right) \quad (5.3)$$

$$\begin{aligned}
 &\text{Modified Equation} \\
 &u^{(0,1)} + a u^{(1,0)} = \left(\frac{-(a \, dt)^2}{2} + \frac{a \, dx}{2} \right) u^{(2,0)} + \\
 &\quad \left(\frac{-(a \, dt)^3}{3} + \frac{a \, dt \, dx}{2} - \frac{a \, dx^2}{6} \right) u^{(3,0)} + \dots
 \end{aligned}$$

Figura 3: Equação modificada obtida pelo programa EQMOD usando a equação de diferenças finitas 5.2.

$$\begin{aligned}
 &\text{Modified Equation} \\
 &u^{(0,2)} + u^{(2,0)} = \frac{-(dy \, u)^2}{12} - \frac{dy^4 \, u^{(0,6)}}{360} - \frac{dx^2 \, u^{(4,0)}}{12} - \\
 &\quad \frac{dx^4 \, u^{(6,0)}}{360} + \dots
 \end{aligned}$$

Figura 4: Equação modificada da equação de Laplace discretizada pela fórmula de 5 pontos.

Para o exemplo mostrado na Seção 3, a escolha do nível de detalhamento 3 gera o resultado mostrado nas Figuras 6 e 7 (compare-o com a Fig. 1). Inicialmente, o programa imprime a equação diferencial obtida após as expansões em série de Taylor em torno do ponto onde a EDP foi discretizada:

$$\begin{aligned}
 &\text{Modified Equation} \\
 &u^{(0,0,1)} - a u^{(0,2,0)} - a u^{(2,0,0)} = \\
 &\quad \left(\frac{a \, dt^2}{2} + \frac{a \, dy^2}{12} \right) u^{(0,4,0)} + a \, dt^2 \, u^{(2,2,0)} + \left(\frac{a \, dt^2}{2} + \frac{a \, dx^2}{12} \right) u^{(4,0,0)} + \dots
 \end{aligned}$$

Figura 5: Equação modificada da EDP parabólica em duas dimensões aproximada por uma fórmula totalmente implícita.

1.
Expand each term of the discretized equation in Taylor series around the point where it was discretized. After algebraic simplifications, we obtain Equation(1)

$$u^{(0,1)} + a u^{(1,0)} = \frac{-(dt \, u)^2}{2} - \frac{dt^2 \, u^{(0,3)}}{6} - \frac{dt^3 \, u^{(0,4)}}{24} - \dots$$

Figura 6: Exemplo de execução do programa com nível de detalhamento 3 – primeiro passo do algoritmo.

A cada termo eliminado, uma tela similar a Fig. 7 é mostrada para o usuário, que pode, assim, acompanhar a execução do algoritmo. Os textos explicativos tornam mais clara a seqüência de operações que estão sendo feitas.


```

2.
Eliminate the terms containing partial derivatives with
respect to Equation(1). The eliminated terms
and the operations performed to eliminate them are listed below:

Term to be eliminated:
      (0,2)
-(dt u
-----
      2
Perform the following operations:
Differentiate Equation(1):
0-times with respect to variable 1 ;
1-times with respect to variable 2 ;
      dt
Multiply it by --
      2
Add the result to Equation(1), obtaining Equation(2)

      2 (0,3)   3 (0,4)   4 (0,5)
      dt u     dt u     dt u
(0,1)  (1,0)  = ----- + ----- + ----- +
u      + a u   12         24         80

```

Figura 7: Exemplo de execução do programa com nível de detalhamento 3 – eliminação das derivadas temporais e textos explicativos.

6. CONCLUSÕES

A equação modificada é uma ferramenta para análise dos erros que surgem na discretização de equações diferenciais parciais. Encontra aplicação, principalmente, entre usuários de métodos numéricos através de técnicas que discretizem essas equações. Embora seja direto, o algoritmo da equação modificada necessita de extensas manipulações algébricas, o que torna laborioso seu cálculo manual. Porém, pode ser implementado em uma linguagem de programação que possua primitivas de manipulação simbólica. Neste trabalho, escolheu-se a linguagem de programação interna do Mathematica.

O programa EQMOD determina a equação modificada de EDPs lineares elípticas, parabólicas e hiperbólicas, com até três dimensões espaciais, aproximadas por discretizações explícitas e implícitas, com um ou mais níveis de tempo.

O EQMOD é útil para pesquisadores na área de métodos numéricos, como uma ferramenta no estudo dos erros presentes nas discretizações de equações diferenciais parciais. Além disso, como o programa possui três níveis de detalhamento, ele pode ser empregado na demonstração de como a equação modificada é obtida, sendo também, portanto, uma ferramenta de ensino. O código está disponível gratuitamente em: <http://www.lcad.icmc.sc.usp.br/~fortuna>.

Agradecimentos

Agradecemos as contribuições para este trabalhos dadas pelos professores do ICMC/USP Dr. José Alberto Cuminato e Dr. Hermínio Cassago Jr. Agradecemos também o apoio financeiro do CNPq para realização deste trabalho.

REFERÊNCIAS

- Goodman, J., Majda, A., 1985, The Validity of the Modified Equation for Nonlinear Shock Waves. *Journal of Computational Physics* 58, 336-348.
- Hirt, C. W., 1968, Heuristic Stability Theory for Finite-Difference Equations. *Journal of Computational Physics* 2, 339-355.
- Hoffmann, K. A., 1989, *Computational Fluid Dynamics for Engineers*. Engineering Education System.
- Liska, R., 1984, Program for Stability and Accuracy Analysis of Finite Difference Methods. *Computer Physics Communications* 34, 175-186.
- Shubin, G. R., Bell, J. B., 1987, A Modified Equation Approach to Constructing fourth order Methods for Acoustic Wave Propagation. *SIAM J. Sci. Stat. Comput.* 8, 153-151.
- Tuomela, J., 1995, A Note on High Order Schemes for the One Dimensional Wave Equation *BIT* 35, 3, 394-405.
- Warming, R. F.; Hyett, B. J., 1974, The Modified Equation Approach to the Stability and Accuracy Analysis of Finite-Difference Methods. *Journal of Computational Physics* 14, 159-179.
- Wolfram, S., 1991, *Mathematica: a System for Doing Mathematics by Computer*, 2nd edition, Addison-Wesley.

A MATHEMATICA PROGRAM TO EVALUATE THE TRUNCATION ERROR ARISING FROM THE DISCRETIZATION OF PARTIAL DIFFERENTIAL EQUATIONS

Abstract: *The modified equation, introduced by Hirt in the late sixties, still is one of the most important tools for evaluating the truncation error terms arising from the discretization of partial differential equations. A few years later, Warming and Hyett placed the modified equation in a solid mathematical basis. However, the basic characteristic of the algorithm, namely that it requires extensive algebraic manipulations of the discretized equation, remained unchanged. The repetitive operations performed by the algorithm make it a good candidate for automation using a computer. But since these operations require symbolic manipulation and differentiation, the algorithm is unsuited for implementation in "standard" programming languages such as Pascal, Fortran or C. For this particular algorithm, appropriate programming languages are those capable of performing such symbolic manipulations. One such language comes with the software package Mathematica. This work describes EQMOD, an implementation of Hirt's modified equation algorithm. EQMOD is written in Mathematica's native programming language. Some simple examples are presented to demonstrate EQMOD's capabilities.*

Keywords: *EQMOD, Finite differences equation, Partial differential equations, Truncation error, Modified equation.*