https://eventos.utfpr.edu.br//sicite/sicite2020

Utilização da biblioteca C-XSC na solução da Equação de Poisson 1D

Use of the C-XSC library in the solution of the Poisson Equation 1D

Guido Margonar Moreira
guidomoreira@alunos.utfpr.edu.br
Universidade Tecnológica Federal do paraná, Apucarana, paraná, Brasil

Ana Paula da Silveira Vargas
anavargas@utfpr.edu.br
Universidade Tecnológica Federal do paraná, Apucarana, paraná, Brasil

RESUMO

Este projeto tem por objetivo final a verificação de erros numéricos na solução de problemas clássicos em Engenharia. A equação utilizada foi a equação de Poisson 1D, discretizada por meio do método de diferenças finitas, e a solução do sistema de equações resultante utilizou-se um método direto, o TDMA e dois iterativos, método de Gauss-Jacobi e método de Gauss-Seidel. A verificação dos resultados numéricos foi realizada utilizando a biblioteca de alta precisão C-XSC, a qual se baseia na aritmética intervalar, aplicada com a linguagem C++. A título de iniciação científica foram explorados conceitos e definições pertinentes a linguagem adotada e no que se trata da biblioteca em questão. Os resultados mostram que, apesar do C++ puro ser extremamente mais rápido, o C-XSC consegue atingir valores mais acurados.

PALAVRAS-CHAVE: Equação de Poisson 1D**.** C++. Aritmética Intervalar. C-XSC. Verificação de Erros Numéricos.

ABSTRACT

This project has as final objective the verification of numerical errors in the solution of classic problems in Engineering. The equation used was the Poisson 1D equation, discretized using the finite difference method, and the solution of the resulting system of equations used a direct method, the TDMA and two iteratives, the Gauss-Jacobi method and the Gauss- Seidel. The verification of numerical results was carried out using the high-precision C-XSC library, which is based on interval arithmetic, applied with the C ++ language. As a scientific initiation, concepts and definitions relevant to the adopted language and the library in question were explored. The results show that, although pure C ++ is extremely faster, C-XSC can achieve more accurate values.

KEYWORDS: Poisson 1D equation. C ++. Interval Arithmetic. C-XSC. Checking Numerical Errors.

Recebido: 19 ago. 2020.
Aprovado: 01 out. 2020.

4 EDUCAÇÃO DE QUALIDADE

9 INDÚSTRIA, INOVAÇÃO E INFRAESTRUTURA

INTRODUÇÃO

O objetivo final de interesse científico é a validação de um modelo matemático que descreve um fenômeno físico em estudo, para isso a verificação se faz necessária. A validação deve ser precedida pela verificação do código e da solução. A verificação do código e a verificação da solução são processos distintos. A validação é definida como o processo que determina o grau em que um modelo está em representação acurada com o fenômeno real. A verificação é o processo usado para quantificar o erro numérico, e o seu objetivo é estabelecer a acurácia numérica, independente do fenômeno físico, isto é, o processo de verificação mede o quão bem o modelo matemático é resolvido numericamente (Knupp e Salari, 2003).

As principais fontes de erros em simulações numéricas são: erros de truncamento, erros de iteração, erros de arredondamento e erros de programação. Quando as fontes de erro de iteração, arredondamento e programação são minimizadas, o erro de truncamento passa a ser denominado erro de discretização, que é a diferença entre a solução analítica e a solução numérica exata das equações discretizadas (Vargas, 2013).

Para se chegar em resultados em que se possa considerar somente os erros de discretização, torna-se necessário, por exemplo, a redução dos erros de arredondamento, que a partir de um certo ponto comprometem a acurácia dos resultados.

Os erros de arredondamento podem ter efeitos catastróficos importantes. Conforme Petrov (2005), na aritmética de ponto-flutuante, uma aproximação apresenta acúmulo de erros de arredondamento e truncamento, e na aplicação de procedimentos para obter a acurácia das soluções, como já mencionado, os erros de arredondamento limitam a acurácia do resultado.

A história mostra o que esses efeitos podem ocasionar para a humanidade; por exemplo, em 25 de Fevereiro de 1991, durante a guerra do Golfo, um antimíssil denominado Patriot, lançado por tropas aliadas, falhou na intercepção para defesa e detecção de ataques inimigos e, como resultado, um míssil Scud iraquiano não poderia ser alvo e foi autorizado a explodir em um quartel, matando 28 pessoas. Essa falha grave no sistema de defesa se deu ao erro no radar do sistema Patriot e no software que o suportava, reduzindo-se no fim, a um erro de arredondamento no cálculo e na medição do tempo (Blair, 1992).

Outro exemplo da influência do erro de arredondamento foi o destino do foguete Ariane, lançado em 04 de junho de 1996 (Agência Espacial Europeia, 1996). Em 37 segundos de voo, a queda se deu por um erro de arredondamento causado ao converter um número em ponto flutuante com 64 dígitos binários (bits) para um número inteiro representado com 16 bits. O sistema de voo do foguete, controlado por computador, entrou em colapso. Os prejuízos foram estimados em várias centenas de milhões de euros.

No cálculo computacional da aritmética em ponto flutuante, cada operação elementar acrescenta erro de arredondamento por causa da codificação limitada de números reais, de forma que todo o resultado calculado é afetado pela propagação de erros de arredondamento.

Com base no trabalho de Wilkinson (1971), Higham (1989) apresenta uma análise de erro a posteriori envolvendo o número de condição de uma matriz triangular obtida por métodos diretos como LU, QR e Cholesky. Segundo ele, como os sistemas triangulares desempenham um papel fundamental no cálculo de matrizes, estes são desejáveis para compreender plenamente a sua solução na aritmética de ponto flutuante e discute em que condições se obtêm uma alta acurácia.

Wilkinson (1971) fornece um panorama histórico de estudos sobre erros de arredondamento começando com uma reavaliação do clássico artigo de Von Neumann e Goldstine (1947). Este, faz uma crítica com relação a adoção de análise de erro a priori e ressalta que limites de erro devem ser determinados por alguma forma de análise de erro a posteriori, pois verifica que aproveita ao máximo a distribuição estatística de erros de arredondamento, e de todas as características especiais, tais como a esparsidade de uma matriz. De qualquer forma, o autor considera que a análise de erro a priori tenha sido extremamente eficaz no sentido de encaminhar as análises na direção correta.

Análises e comparações entre os métodos Gauss-Jordan e eliminação de Gauss são tratados por Dekker e Hoffmann (1989). Segundo esses autores, se um problema é mal condicionado e numericamente instável, métodos utilizados para a solução de sistemas de equações, como a eliminação de Gauss tornam-se inúteis.

A acurácia dos resultados finais de um cálculo pode refletir uma combinação de erros, que podem ser amplificados pela natureza do problema a ser resolvido, ou pelo algoritmo a ser utilizado, ou ambos. Para Martel (2006), a propagação de erros de arredondamento e a introdução de novos erros em cada etapa do cálculo computacional é um fenômeno que quase nunca é estudado em uma estrutura adequada. Este propõe uma estrutura para operações de ponto flutuante, que descreve a propagação de erros de arredondamento ao longo de um cálculo (Martel, 2002).

Considerada uma das principais fontes de erro em uma simulação computacional, o erro de arredondamento pode ser atenuado pelo uso de uma quantidade maior de dígitos significativos no cálculo (Roy, 2005).

Como alternativa ao estudo da redução de erros como os erros de arredondamento, são analisados os resultados obtidos utilizando a biblioteca C-XSC. Esta biblioteca é utilizada no desenvolvimento de algoritmos numéricos com a geração de resultados com alta acurácia e verificados automaticamente. Ela fornece um grande número de tipos de dados numéricos e operadores predefinidos, permitindo, assim, a programação de alto nível de aplicações numéricas (Hölbig, 2005).

A biblioteca C-XSC tem como base a aritmética intervalar, que é uma teoria que vem ao encontro da redução de erros de arredondamento e truncamento, pois consiste no processamento de dados por meio de intervalos, resultando como resposta final um intervalo na qual o valor exato se encontra.

Segundo Moore et al. (2009), a propagação do erro dos dados iniciais e a propagação do erro de arredondamento em qualquer sequência finita de operações aritméticas podem ser ambas rigorosamente controladas, simplesmente pela utilização da aritmética intervalar.

Goldberg (1991) analisa a abordagem sobre o erro de arredondamento dentro do padrão IEEE (1985), e coloca que a aritmética intervalar, em contrapartida as clássicas estimativas de erros, é uma alternativa para a redução de erro de arredondamento.

Segundo Santos (2001), o objetivo da aritmética intervalar consiste em fornecer esquemas computacionais de modo que, em se efetuando vários cálculos, os erros de arredondamento e truncamento sejam reduzidos consideravelmente e, além disso, conhecidos.

Conforme Hölbig (2005), a importância do estudo dos métodos intervalares para a resolução de sistemas de equações incide no fato desses métodos produzirem resultados dentro de limites confiáveis (do intervalo solução) e provarem a existência ou não existência de soluções, portanto produzem resultados confiáveis, o que os métodos numéricos usuais podem não proporcionar sem uma análise exaustiva de erros.

Sabe-se que sistemas de equações podem apresentar tanto matrizes de coeficientes densas como esparsas. Análises sobre métodos numéricos aplicados para a solução nestes casos mostram que em sistemas cuja matriz de coeficientes é densa os métodos diretos são mais adequados, e que em sistemas cuja matriz de coeficientes é esparsa, do tipo banda tridiagonal, são resolvidos mais eficientemente por métodos iterativos.

Nesse sentido, novamente, uma alternativa para se reduzir o efeito dos erros de arredondamento é a aplicação da aritmética intervalar por meio da biblioteca C-XSC. Essa aplicação é uma versão intervalar de métodos numéricos usuais, denominados por Hölbig (1996) de métodos pontuais.

Hölbig (1996) coloca que a extensão intervalar dos métodos numéricos usuais (ou pontuais), não é simples e que o cálculo da solução é dispendioso por envolver vetores e matrizes de intervalos. Ele afirma que a aritmética intervalar é uma alternativa econômica na qual uma aproximação pontual é calculada e, a partir desta, melhorada por meio de métodos intervalares.

Com isso, a grande motivação para este trabalho encontra-se na importância da análise numérica e controle de erros, mais especificamente, buscam-se alternativas para o procedimento de redução dos erros de arredondamento, contribuindo assim, para a evolução de novas teorias, tecnologias e práticas no desenvolvimento de códigos e soluções dos problemas de interesse.

MATERIAL E MÉTODOS

Como já mencionado anteriormente, o modelo matemático abordado nesse estudo descreve o problema da condução de calor unidimensional com condições de Dirichlet, e é dado pelas Eq. (1) e Eq. (2)

$$-\frac{d^2u}{dx^2} = f(x) \tag{1}$$

$$u(0) = 0, u(1) = 0{,}5 \tag{2}$$

onde u é a temperatura, e o termo fonte é dado por f(x)=1. A solução analítica é Eq. (3)

$$u(x) = x - \frac{x^2}{2} \tag{3}$$

A discretização do modelo foi desenvolvida por meio do método de diferenças finitas centradas, em que a derivada segunda, em relação a x, é aproximada na Eq. (4)

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i-1}+u_{i+1}-2u_i}{h^2} \tag{4}$$

e com erro de truncamento resultante dado pela Eq. (5)

$$\varepsilon = -u^{iv}\frac{h^2}{12} - u^{vi}\frac{h^4}{360} - u^{viii}\frac{h^6}{20160} - u^{x}\frac{h^8}{1814400} - \cdots \tag{5}$$

A Eq. (5) se baseia na aproximação da derivada da função pelas respectivas equações de diferenças obtidas por meio da série de Taylor. Esta tem por objetivo, avaliar as ordens verdadeiras do erro de truncamento *a priori* e verificadas *a posteriori* com a utilização do método de multiextrapolações de Richardson (MER) (Vargas, 2013).

O sistema de equações obtido por meio da Eq. (4) substituída na Eq. (1), e considerando a Eq. (2), é resolvido por dois métodos iterativos, a saber, os métodos de Gauss-Jacobi e Gauss-Seidel, conforme encontrado em Burden e Faires, 2008.

Para a condição de parada utilizou-se dois recursos, o número de iterações máximas de 27000 e tolerância de no máximo 1.10^{-10}.

O funcionamento da biblioteca foi compreendido por meio do próprio site do C-XSC que possui a documentação nova e antiga, a ferramenta é dividida em várias sub-bibliotecas, para permitir acesso direto apenas ao conteúdo que o programa utiliza. Neste trabalho foram usadas $l\_rmatrix.hpp$, para manipulação de matrizes, e $l\_rmath.hpp$ funções matemáticas como $pow(x, y)$ para x^y e $\sin(x)$ para seno de x, ambas possuem o prefixo l_r que significa números reais de alta precisão.

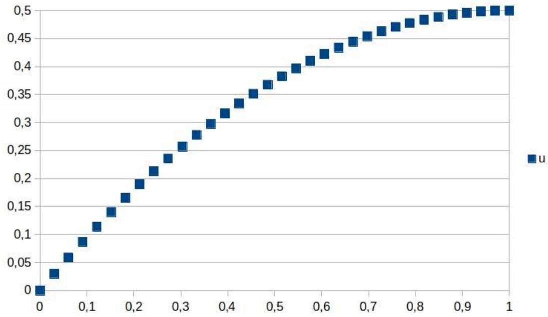
O C-XSC permite a definição prévia do número de casas decimais utilizadas pelo programa, porém como O C++ consegue chegar até 19 casas decimais, o C-XSC também foi definido para 19 casas decimais, desta maneira os resultados podem ser comparados, quando o erro do programa chegar a uma tolerância menor do que a definida no programa que é encerrado e todos os dados mandados para um arquivo utilizado para a criação dos gráficos.

Originalmente os testes seriam realizados no Windows e Linux, permitindo a comparação de ambos sistemas operacionais, porém ocorreram problemas durante as tentativas de instalação no Windows e vários arquivos não eram instalados. A biblioteca por ser formada de diversas partes foi de difícil importação em várias IDEs, no final foi decidido que todos os testes seriam realizados no linux utilizando o CLion, IDE para C e C++, que oferece uma licença gratuita para estudantes e professores, e possui um processo mais simples para importação do C-XSC.

RESULTADOS E DISCUSSÕES

Na Figura 1 pode-se verificar o resultado da solução com a biblioteca C-XSC.

Figura 1 – Solução da equação de Poisson 1D com a utilização da biblioteca C-XSC



Fonte: Autoria própria (2020).

Para realizar uma comparação de tempo computacional mais preciso, o cálculo foi realizado para 5 malhas, com 5, 10, 33, 50 e 100 nós, com número de iterações igual a 1000, como mostra o quadro 1.

Quadro 1 – Tempo computacional utilizado no cálculo das malhas.

Malhas	C++	C-XSC
n=5	0,002579 s	0,06624322222 s
n=10	0,00338944444 s	0,22753166666 s
n=33	0,01598155555 s	1,38409911111 s
n=50	0,02524666666 s	2,58811133333 s

Fonte: Autoria própria (2020).

O quadro 2 demonstra que apesar do C++ ser extremamente mais rápido, como visto na Tabela 1, o C-XSC consegue atingir valores mais precisos, se a tolerância fosse 10^{-20}, por exemplo. A linguagem C++ não teria espaço o bastante para armazenar esse valor e realizar as comparações para chegar na interação final, enquanto isso o C-XSC até mesmo encontrar o valor mínimo do erro, definindo a tolerância como10^{-30} o programa em C-XSC pode chegar no resultado com a seguinte precisão 2.2624639630813589068925165539825$9.10^{-30}$.

Quadro 2 – Tempo computacional de acordo com o número de casas decimais.

Casas decimais	C-XSC
16	14,2911012 s
32	17,3609956 s
128	39,0979056 s
320	9,5242322 s

Fonte: Autoria própria (2020).

CONCLUSÕES

Foram realizados testes com e sem a biblioteca C-XSC. Ambos os programas, para C++ com e sem C-XSC apresentaram resultados coerentes e acurados.

Mesmo com o custo computacional extra, é possível chegar a resultados extremamente mais precisos fazendo uso da biblioteca.

REFERÊNCIAS

BLAIR, M. GAO/IMTEC-92-26 **Patriot Missile Software Problem**. < http://www.fas.org/spp/starwars/gao/im92026.htm > Acesso em 05 nov. 2010.

BURDEN, R. L., FAIRES, J. D. **Análise Numérica.** Cengage Learning, São Paulo, 2008.

DEKKER, T. J., HOFFMANN, W. **Rehabilitation of the Gauss-Jordan Algorithm.** Numerische Mathematik, Springer-Verlag, 1989.

GOLDBERG, D. **What every computer scientist should know about floating-point arithmetic**. Association for Computing Machinery, Inc., 1991.

HIGHAM, N. J. **The Accuracy of Solutions to Triangular Systems. SIAM Journal on Numerical Analysis**, v. 26, No. 5, pp. 1252-1265, 1989.

HÖLBIG, C. A. **Métodos intervalares para a resolução de Sistemas de Equações Lineares**. Dissertação de Mestrado, Porto Alegre – RS - UFRGS, 1996.

HÖLBIG, C. A. **Ambiente de Alto desempenho com Alta Exatidão para a resolução de problemas**. Tese de Doutorado, Porto Alegre – RS - UFRGS, 2005.

KNUPP, P. M., SALARI, K. **Verification of computer codes in computational science and engineering**, Chapman & Hall/CRC, New York, 2003.

MARCHI, C. H. **Verificação De Soluções Numéricas Unidimensionais em Dinâmica dos Fluidos.** Tese de Doutorado, Florianópolis – Santa Catarina – UFSC, 281 p, 2001.

MARTEL, M. **Propagation Of Roundoff Errors In Finite Precision Computations: A Semantics Approach**. In 11th European Symposium on Programming, ESOP'02, number 2305 in Lecture Notes in Computer Science, pages 194–208. Springer - Verlag, 2002.

MARTEL, M. **Semantics of roundoff error propagation in finite precision computations.** Journal of Higher Order and Symbolic Computation, pp 7-30, 2006.

MOORE, R. E., KEARFOTT, R. B., CLOUD, M. J. **Introduction to interval analysis**. SIAM, Philadelphia, 2009.

PETROV, N. I. **Criterion for Richardson's extrapolations of risk technical systems. Academic Open Internet Journal**, v. 16, 2005.

ROY, C. J. **Review of code and solution verification procedures for computational simulation.** Journal of Computational Physics 205, pp. 131–156, 2005.

SANTOS, J. M. **Em Direção A Uma Representação Para Equações Algébricas: Uma Lógica Equacional Local**. Dissertação de Mestrado, Natal - Rio Grande do Norte – UFRN, 2001.

University of Wuppertal. C-XSC Languages. **Como instalar c-xsc no linux e windows**. Disponível em: <http://www2.math.uni-wuppertal.de/wrswt/xsc/cxsc/README >. Acesso em: 29 de jul. de 2020.

VARGAS, A. P. da S. **Multiextrapolação de Richardson e Esquemas de 1ª e 2ª Ordens, Mistos e Crank-Nicolson sobre as Equações 2D de Advecção-Difusão e Fourier**, Tese de Doutorado, PGMEC-UFPR, Curitiba, 2013.

VON NEUMANN, J., GOLDSTINE, H. H. **Numerical inverting of matrices of high order**. Bull. Amer. Math. Soc., 53, pp 1021-1099, 1947.

WILKINSON, J. H. **Modern error analysis.** SIAM Review, v. 13, no. 4, pp. 548-568, 1971.