

Escolhendo Ambientes Intervalares sob diferentes critérios

Lucas M. Tortelli¹ **Maurício D. C. Balboni¹** **Mariline Lorini¹**

Alice F. Finger² **Aline B. Loreto²**

Universidade Federal de Pelotas

¹Curso Ciência da Computação/CDTec

²Programa de Pós-graduação em Computação/CDTec

CEP 96001-970, Campus Porto, Pelotas-RS

E-mail: baalbis@gmail.com, {lmtortelli, mlorini, affinger, aline.loreto@inf.ufpel.edu.br}

RESUMO

Problemas numéricos na computação científica originam-se primordialmente da impossibilidade de se operar com os números reais diretamente, pois tem-se que representar uma grandeza contínua (a reta real) de forma discreta (palavras de máquina). O sistema de ponto flutuante [2] é uma aproximação prática dos números reais.

Os intervalos foram definidos com o objetivo inicial de automatizar a análise do erro computacional. Através da utilização de intervalos, tem-se um controle automático de erros com limites confiáveis [4]. A aritmética intervalar utiliza intervalos reais para representar valores infinitos, valores desconhecidos ou para representar valores contínuos que podem ser conhecidos ou não. Os intervalos servem para representar dados inexatos, aproximações e erros de truncamento de procedimentos.

Para obtenção de resultados com maior exatidão, cálculos numéricos devem ser suportados pela matemática intervalar e pela aritmética de exatidão máxima, o que implica que em computadores sejam realizados por meio das linguagens ou bibliotecas que tenham definidos o tipo intervalo e as operações sobre o tipo, usualmente denominadas de linguagens XSC (*eXtended Scientific Computation*) [3].

Existem diversos ambientes computacionais com suporte para a matemática intervalar, dentre os quais destacam-se: Maple Intervalar, IntLab, IntPy, C-XSC, Fortan-XSC, Pascal-XSC e Java-XSC. Destes, dois são softwares que fazem o uso da metodologia de aritmética intervalar, como os pacotes Maple Intervalar e o IntLab, os demais são linguagens de programação que suportam o tipo intervalo.

Diante de várias opções de ambientes intervalares e da dificuldade em escolher a melhor ou mais adequado para utilizar em futuros trabalhos que demandem exatidão nas soluções, o presente trabalho analisa tais ambientes considerando diferentes critérios: qualidade de software, avaliação de linguagens de programação e critérios de qualidade do intervalo.

Segundo Rocha [1], os objetivos de qualidade de um software determinam as propriedades gerais que o produto deve possuir. Os fatores de qualidade do produto determinam a qualidade do ponto de vista dos diferentes usuários do produto, são eles: manutenibilidade, operacionalidade, portatibilidade, reutilizabilidade, eficiência, rentabilidade, avaliabilidade.

As linguagens de programação são classificadas de acordo como são escritas e como são executadas. De acordo com Sebesta [5], os critérios para avaliação de linguagens de programação são: legibilidade, manutenção, simplicidade, ortogonalidade, suporte a abstração, verificação de tipos e manipulação de excessões.

Quando o foco é a exatidão da solução considera-se a verificação da qualidade do intervalo como um critério a ser analisado. A computação com utilização de intervalos fornece as seguintes estimativas para o erro [6]:

- Diâmetro: $w = \bar{x} - \underline{x}$

- Erro Absoluto: $|x - m(\mathbf{X})| < w(\mathbf{X})/2$;
- Erro Relativo: $\frac{x-m(\mathbf{X})}{x} < \frac{w(\mathbf{x})}{2\min|\mathbf{X}|}$

Quando se trabalha com números de ponto flutuante o resultado obtido é apenas uma aproximação de um valor real e erros são gerados por arredondamentos ou por algoritmos instáveis, levando algumas vezes a resultados incorretos.

Segundo Ratschek [6], os computadores utilizam aritmética de ponto flutuante, onde números reais são aproximados por um subconjunto de números reais chamados representação numérica da máquina. Devido a esta representação, são gerados dois tipos de erros: quando uma entrada de valor real é aproximada por um número de máquina e quando o erro é causado por resultados intermediários aproximados pelos números de máquina. A aritmética intervalar fornece uma ferramenta para estimar e controlar esses erros automaticamente.

A partir dos critérios de qualidade de software, de linguagens de programação e de qualidade do intervalo, o presente trabalho visa analisar e escolher o melhor ambiente de programação intervalar.

Em relação aos critérios para avaliação de linguagens de programação, considerando linguagens compiladas, o C-XSC se destaca por apresentar bons conceitos de abstração, manipulação de exceções, manutenibilidade e legibilidade. Porém, a linguagem interpretada Python é altamente abstrata e expressiva, uma vez que suas instruções são simples e a combinação destas definem instruções mais complexas. Em se tratando de exatidão, verificada com os critérios de qualidade do intervalo, Python apresentou os melhores resultados em comparação a linguagem C-XSC e aos pacotes Maple Intervalar e IntLab.

Sobre os critérios técnicos, verifica-se que os softwares que são proprietários, como o IntLab e o Maple Intervalar, são de boa atualização, fácil instalação e possuem manual de utilização. Já os que são gratuitos, como C-XSC e IntPy, também possuem manual de utilização, mas suas atualizações não são tão boas quanto dos softwares proprietários. Salienta-se que o IntPy é de fácil instalação porém o C-XSC não. Não deixando de citar que no IntLab existe uma grande exatidão nos resultados e que o Maple Intervalar é o de mais fácil utilização.

Assim, conclui-se que os melhores ambientes analisados quanto aos critérios utilizados no presente trabalho são: C-XSC e IntPy. Salienta-se que o ambiente a ser adotado em futuras pesquisas será IntPy, devido a exatidão dos resultados e por possuir um pacote Pypy que supre a necessidade do Python ser previamente compilada e otimizada antes de ser executada.

Garantindo a qualidade do intervalo solução e conhecendo o ambiente intervalar que retorna o intervalo com melhor qualidade, tem-se uma ferramenta intervalar confiável para ser utilizada em diversas aplicações.

Palavras-chave: *Ambientes intervalares, qualidade de intervalo, exatidão numérica, análise de ambientes intervalares, aritmética intervalar*

Referências

- [1] Rocha, A. R.; de Campos, G. H. B, "Avaliação da qualidade de software educacional," *Em Aberto*, no.57, 1993.
- [2] Goldberg, D, "What Every Computer Scientist Should Know About Floating-Point Arithmetic", *ACM Computing*, pp. 5-48, 1991.
- [3] Klatte, R.; Kulisch, U.; Wiethoff, A.; Lawo, C.; Rauch, M, "C-XSC - A Class Librart for Extended Scientific Computing, *Springer-Verlag*, 1993.
- [4] Moore, R. E, "Interval Analysis", Englewood Cliffs: Prentice Hall, 1966.
- [5] Sebesta, R. W, "Conceps of Programming Languages", *Colorado Spring: Addison Wesley Longman*, pp. 17-49, 1999.
- [6] Ratschek, H.; Rokne, R, "New Computer Methods for Global Optimization", Ellis Horwood, 1988.