Paralelização de Solvers Intervalares para a Resolução de Sistemas Lineares

Andriele Busatto do Carmo, Leila Diane Wentz,
Carlos Amaral Hölbig

Curso Ciência da Computação - Universidade de Passo Fundo
Campus I – BR 285 – Bairro São José CEP 91501-970 – Passo Fundo – RS - Brasil
{62408, 62384}@lci.upf.br , holbig@upf.br

Introdução

A resolução de sistemas de equações lineares é um dos problemas numéricos mais comuns em aplicações científicas. Tais sistemas surgem, por exemplo, em conexão com a solução de equações diferenciais parciais, determinação de caminhos ótimos em redes (grafos) e interpolação de pontos, dentre outros [BOR 03]. Os problemas que podem ser resolvidos através desses sistemas lineares envolvem, por exemplo, a determinação de potenciais em certas redes elétricas, o cálculo do estresse numa armação de construção ou de uma estrutura de ponte, o cálculo do padrão de escoamento num sistema hidráulico com ramos interconectados ou o cálculo das estimativas das concentrações de reagentes sujeitos a simultâneas reações químicas.

Esta pesquisa tem por meta paralelizar os *solvers* verificados LSS (*Linear System Solver*) e BAND. O *solver* LSS foi implementado em C-XSC [KLA 93], resolvendo sistemas lineares quadrados, sobre-determinados (ou seja, com mais equações do que incógnitas) e sub-determinados (com mais incógnitas do que equações) para valores reais, números complexos, intervalos e intervalos complexos. O cálculo da aproximação da inversa para as matrizes quadradas e da pseudo-inversa para as matrizes sobre e sub-determinadas, é igualmente abordado pelo LSS.

Já o solver BAND foi implementado com o objetivo de solucionar sistemas lineares com matrizes esparsas do tipo banda. Todos esses algoritmos foram implementados com técnicas que possibilitam a Computação Verificada.

Computação Verificada

O cálculo numérico com a verificação de resultados é de fundamental importância para muitas aplicações como, por exemplo, para a simulação e modelagem matemática. Na análise clássica do erro em algoritmos numéricos, o erro em cada operação em ponto-flutuante é estimado. Na verdade, a possibilidade do resultado estar errado não é normalmente considerada. Do ponto de vista matemático, o problema da correção dos resultados é de grande importância para garantir a alta velocidade computacional atingida atualmente. Isto torna possível ao usuário distinguir entre inexatidão nos cálculos e as reais propriedades do modelo. No sentido de tornar possível o manuseio de milhões de adições e subtrações com o máximo de exatidão, é evidente que as capacidades da aritmética de ponto-flutuante tradicional têm que ser aumentadas. Dadas as possibilidades atuais, não há razão para que isto não possa ser feito em um *chip*, simulado por *software* ou pela combinação dos dois.

O grande objetivo da Computação Verificada é possibilitar que o próprio computador possa rapidamente estabelecer se o cálculo realizado é ou não correto e útil para o problema que se quer solucionar. Neste caso, o programa pode escolher um algoritmo alternativo ou repetir o processo usando uma maior precisão. Técnicas similares de computação verificada podem ser aplicadas para muitas áreas de problemas algébricos, tais como a solução de sistemas de equações lineares e não lineares, o cálculo de raízes, a obtenção de autovalores e de autovetores de matrizes, problemas de otimização, etc. Em particular, a validade e a alta exatidão da avaliação de expressões aritméticas e de funções no computador está incluída. Estas rotinas também funcionam para problemas com dados intervalares. É importante ressaltar que para que se desenvolvam programas computacionais que tenham o paradigma da Computação Verificada é obrigatório o uso de todas as suas características, ou seja, o uso da aritmética de alta exatidão, dos métodos intervalares de inclusão e da convergência garantida pelo Teorema de Ponto Fixo de Brouwer, além do uso de algoritmos apropriados. Com a Computação Verificada é possível desenvolver métodos numéricos para a realização de computações científicas com verificação automática do resultado [ALE 83].

Solver LSS (*Linear System Solver*)

O *solver* LSS – *Linear System Solver* (e suas extensões[1]) é uma ferramenta que auxilia na resolução dos seguintes problemas:

1. Solução de Sistemas Lineares Sobre-determinados (m x n, com m > n);
2. Solução de Sistemas Lineares Quadrados (n x n);
3. Solução de Sistemas Lineares Sub-determinados (m x n, com m < n);
4. Calculo da Inversa A-1 de A, no caso do Sistema Quadrado;
5. Calculo da Pseudo-inversa A+ de A, no caso do Sistema Sobre-determinado;
6. Calculo da Pseudo-inversa A+ de A, no caso do Sistema Sub-determinado.

Detalhes sobre o solver LSS e a sua implementação em seqüencial poderão ser encontrados em [ALC 04], [MOR 04] e [HÖL 05].

[1] *solver LSS* – para dados de entrada do tipo real, *solver ILSS* – para dados de entrada do tipo intervalo, *solver CLSS* – para dados de entrada com números complexos e *solver CILSS* – para dados de entrada com números complexos intervalares

Solver BAND

O *solver BAND* é um *solver* para a resolução de sistemas de equações lineares com matrizes esparsas, em especial para matrizes do tipo banda. Sistemas com matrizes bandas são um caso particular de sistemas esparsos, onde os elementos não nulos encontram-se em torno da diagonal principal. O algoritmo utilizado para resolver sistemas densos não é muito eficiente neste caso, uma vez que é necessária grande quantidade de memória para alocação dos sistemas e há um aumento no tempo de execução. Por se tratar de um caso particular de sistemas esparsos, as matrizes bandas podem ser alocadas em faixas (*bandwidths*) onde os elementos não nulos estão presentes. Neste *solver* empregou-se a Computação Verificada na elaboração de algoritmos numéricos auto-validáveis e com controle automático de erros.

Entretanto, o uso de intervalos acarreta em um efeito conhecido como *wrapping effect*. O *wrapping effect* é um efeito que faz o diâmetro dos intervalos resultantes aumentar rapidamente com o uso de intervalas, ou seja, o intervalo calculado como resposta diverge bastante da resposta exata. Para tentar minimizar esse efeito, o programa foi desenvolvido baseado na resolução de equações diferenciais, o que mostrou bastante eficiência no tratamento deste efeito. Após essa experiência, um programa para a resolução de sistemas lineares triangulares com matrizes bandas foi desenvolvido. O algoritmo implementado no *solver* foi baseado no estreito relacionamento existente entre as matrizes com estrutura banda e equações diferenciais. De acordo com esse relacionamento, as equações diferenciais podem ser reescritas equivalentemente como um sistema linear triangular com matrizes banda. Similarmente, pode-se reescrever um sistema triangular com matrizes banda como equações diferenciais. O cálculo da aproximação da inversa através do método de Gauss (utilizado para resolver sistemas densos) foi substituído pelo cálculo da aproximação através da Decomposição LU da matriz A. Detalhes sobre o solver BAND e a sua implementação em seqüencial poderão ser encontrados em [ALC 04] e [HÖL 05a].

Conclusões e Trabalhos Futuros

Os *solvers* LSS e BAND surgiram da implementação dos algoritmos do *Numerical Toolbox for Verified Computing* II em C-XSC [KRÄ 93]. Essas implementações serviram como um primeiro passo rumo a um objetivo maior. A meta agora é paralelizar as rotinas intervalares que compõem os *solvers* LSS e BAND, para que eles possam ser executados em ambientes de alto desempenho, do tipo *clusters* de computadores. Para atingir esse objetivo pretende-se paralelizar os programas implementados em C-XSC com o uso da biblioteca MPI no *cluster* Colorado do grupo ComPaDi/UPF. Ao final da pesquisa, uma análise do desempenho e da exatidão obtidos com a versão paralela dos *solvers* deverá ser realizada. Paralelizados, os *solvers* LSS e BAND ficarão disponíveis para o uso nas aplicações desenvolvidas em ambientes *clusters* de computadores.

Referências

[ALC 04] ALCALDE, B. F. K.; MORANDI JÚNIOR, P. S.; HÖLBIG, C. A.; DIVERIO, T. A. Resolução de Sistemas Lineares com Alta Exatidão. Revista Eletrônica de Iniciação Científica (online), Porto Alegre, v.1, 2004.

[ALE 83] ALEFELD, G.; HERZBERGER, J. An Introduction to Interval Computations. New York: Academic Press, 1983.

[BOR 03] BORTOLI, A.; CARDOSO, C.; FACHIN, M.; CUNHA, R. **Introdução ao Cálculo Numérico – 2a. Edição**. Porto Alegre, Brasil: Universidade Federal do Rio Grande do Sul, 2003.

[HÖL 04] HÖLBIG, C. A.; CLAUDIO, D. M.; DIVERIO, T. A. Resolução de Sistemas Lineares com Alta Exatidão no Ambiente de Agregados. In: ESCOLA REGIONAL DE ALTO DESEMPENHO, IV, 2004, Pelotas, Brasil. Anais. . . Pelotas: SBC, 2004. p.153–154.

[HÖL 05] HÖLBIG, C. A.; MORANDI JÚNIOR, P. S.; ALCALDE, B. F. K.; DIVERIO, T. A.; CLAUDIO, D. M. Solving Linear Systems on Cluster Computers with High Accuracy. Copenhagen, Denmark: Department of Informatics and Mathematical Modelling of the Technical University of Denmark, 2005. (IMM-Technical report-2005-09).

[HÖL 05a] HÖLBIG, C. A.; MORANDI JÚNIOR, P. S.; CLAUDIO, D.M.; DIVERIO, T. A. Solving Real Life Applications With High Accuracy. In: INTERNATIONAL CONFERENCE ON PARALLEL COMPUTING, PARCO 2005, 2005, Málaga, Spain. **Anais. . .** Málaga: Universidad de Málaga, 2005. p.98.

[KRA 03] KRAMER, W.; KULISCH, U.; LOHNER, R.: Numerical Toolbox for Verified Computing II, Advanced Numerical Problems, http://www.xsc.de. Springer-Verlag, Heidelberg, New York, 1993.

[KLA 93] KLATTE, R.; KULISCH, U.; WIETHOFF, A.; LAWO, C.; RAUCH, M.: C-XSC – A C++ Class Library for Extended Scientific Computing , Springer-Verlag, Heidelberg, 1993.

[MOR 04] MORANDI JÚNIOR, P. S.; ALCALDE, B. F. K.; HÖLBIG, C. A.; DIVERIO, T. A. A Integração da Biblioteca de Alta Exatidão C-XSC em Ag regados de Computadores. Revista Eletrônica de Iniciação Científica (online), Porto Alegre, v.2, 2004.