基盤研究(B)(1)「科学技術計算に現れる超大規模線形方程式の数理的諸問題と高速解法の総合的開発」

手軽に反復解法

長谷川 秀彦（筑波大学 図書館情報メディア研究科）

最近の話題から

- 適切なアルゴリズムを選択するには？
- ユーザに合った反復法ライブラリとは？
- 多様な環境に適応するプログラムとは？
- 非対称行列に CG 法を適用するには？
- 共有メモリ上の固有値解析の並列化

Iterative Solvers

- Key component of Scientific computation
- The best algorithm for the problem?
- The best storage format?
- Various Computing Environments
- Accurate and robust code
- Quick implementation and ease of use
- Future Extension

これだけでいいのか？

- $\boldsymbol{x} = (1, 1, 1, \ldots, 1)^T$
- $\boldsymbol{b} = (1, 1, 1, \ldots, 1)^T$
- $\boldsymbol{x}_0 = (0, 0, 0, \ldots, 0)^T$

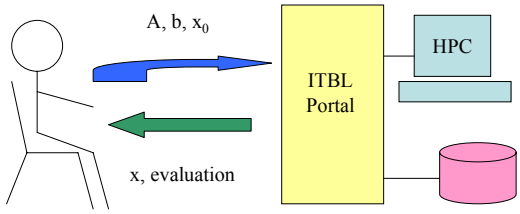
For productive scientific simulation

- TiS evaluates algorithms based on data
- Lis provides various choice to your code
- SILC supports many computing environments

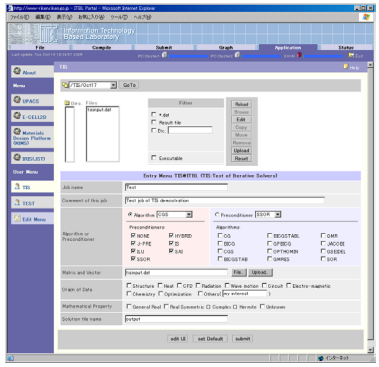
To find the best algorithm

- To choose the best choice of Iterative Solver and Preconditioner is difficult problem. What should I do?
- Use TiS (Test of Iterative Solvers);
  TiS suggests the best combination of Iterative Solver and Preconditoner based on your own problem with no programming effort.

TiS: Test of Iterative Solvers

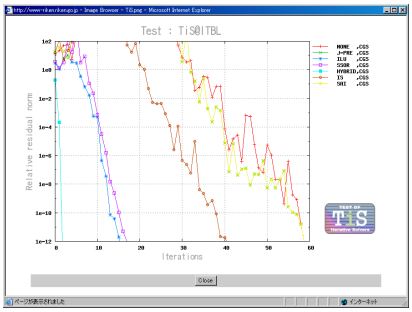


Evaluation of iterative solvers and preconditioners for Ax=b



Selection Menu of TiS

Comparison result with marker



To choose the best implementation

- How should I decide which data storage format is the best for the computing environment?
- Lis (a Library of Iterative Solvers for linear systems) provides various iterative solvers, various preconditioners, and various storage formats.

LiS

- 10 Iterative Solvers
- 10 Preconditioners
- 10 Storage Formats

Over 1,000 Combinations on OpenMP/MPI environments

- Double and Fast Quadruple on SSE2

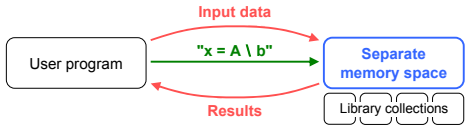
To port your code for anywhere else

- Will you use your code in many computing environments?
- SILC (Simple Interface of Library Collections) is the solution. The user code of SILC run any computing environments with no change to your code.

SILC 構想の全体像

SILC: Simple Interface for Library Collections

- Basic ideas
  - **Data transfer** and **a request for computation**
  - **Mathematical expressions** for the request
  - **A separate memory space** for the computation



関連研究：MATLAB と SILC

- MATLAB
  - 既存のプログラミング言語を置き換える統合開発環境
  - 演算子・関数とライブラリ関数の対応関係が一定
    - \ の解法は行列構造に応じて一定の選択肢から選ばれる
    - svd, rcond などの関数は LAPACK の呼出しに変換
- SILC
  - 従来のライブラリ利用法に代わるミドルウェア
    - 既存のプログラムの一部のみに SILC の方法を適用できる
  - 演算子・関数とライブラリ関数の対応関係が可変
    - さまざまなライブラリ、計算環境を容易に試せる

ライブラリの考慮点（緑は対応済み）

- フレームワーク、精度、性能、メモリサイズ、インタフェース、データ構造、アルゴリズム、数値精度、逐次、共有並列、分散並列、アーキテクチャ、バージョンアップ、バグ修正、課金、標準化、インタプリタ・コンパイラ、バッチ・TSS、多様性、頑健性、移行性、アルゴリズムのフィッティング、チューニング、 …

Collaborators and Acknowledgement

- TiS on ITBL
  Y. Fukui (JAXA)
  K. Suzuki (Fujitsu)
  Y. Sakaguchi (Fujitsu)
- Lis
  H. Kotakemori (JST/U Tokyo)
  A. Fujii (Kogakuin U)
  K. Nakajima (U Tokyo)
  A. Nishida(Chuo U/JST)
- SILC
  T. Kajiyama (JST/U Tokyo)
  A. Nukada (JST/U Tokyo)
  R. Suda (U Tokyo)
  A. Nishida (Chouo U/JST)
- Lis and SILC are parts of SSI project which is funded by JST/CREST

Information resources

- TiS http://amazon.slis.tsukuba.ac.jp/TiS/
  hasegawa@slis.tsukuba.ac.jp
- Lis http://ssi.is.s.u-tokyo.ac.jp/lis/
- SILC http://ssi.is.s.u-tokyo.ac.jp/silc/