

多倍長関係の最近の結果

渡部 善隆

<http://www.cc.kyushu-u.ac.jp/RD/watanabe/>

2003年9月19日

概要

概要

- 多倍長共役勾配法
 - 数値実験結果
 - 知見

概要

- 多倍長共役勾配法
 - 数値実験結果
 - 知見
- 4倍精度区間演算による計算機援用証明
 - Orr-Sommerfeld 問題
 - 検証結果
 - 知見

共役勾配法 (CG法)

set an initial vector \mathbf{x}_0 ; $\mathbf{r}_0 := \mathbf{b} - A\mathbf{x}_0$; $\mathbf{p}_0 := \mathbf{r}_0$;
for $k := 0, 1, \dots$ until $\mathbf{r}_k = \mathbf{0}$ do
begin

$$\alpha_k := \frac{(\mathbf{r}_k, \mathbf{p}_k)}{(\mathbf{p}_k, A\mathbf{p}_k)};$$

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k; \quad \mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k A\mathbf{p}_k;$$

$$\beta_k := -\frac{(\mathbf{r}_{k+1}, A\mathbf{p}_k)}{(\mathbf{p}_k, A\mathbf{p}_k)};$$

$$\mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$$

end

共役勾配法の収束定理

共役勾配法の収束定理

- \boldsymbol{x}^* を $A\boldsymbol{x} = \boldsymbol{b}$ の真の解, 目的関数を

$$\phi(\boldsymbol{x}) := (\boldsymbol{x} - \boldsymbol{x}^*)^T A(\boldsymbol{x} - \boldsymbol{x}^*)/2$$

とする. A が対称かつ**正定値**であれば, $\phi(\boldsymbol{x}_k)$ は単調に減少し, $\kappa = \lambda_N/\lambda_1 > 0$ に対して以下が成立する.

$$\phi(\boldsymbol{x}_k) \leq \phi(\boldsymbol{x}_0) \cdot 4 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2k} .$$

共役勾配法の収束定理

- x^* を $Ax = b$ の真の解 , 目的関数を

$$\phi(x) := (x - x^*)^T A(x - x^*)/2$$

とする . A が対称かつ**正定値**であれば , $\phi(x_k)$ は単調に減少し , $\kappa = \lambda_N/\lambda_1 > 0$ に対して以下が成立する .

$$\phi(x_k) \leq \phi(x_0) \cdot 4 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2k} .$$

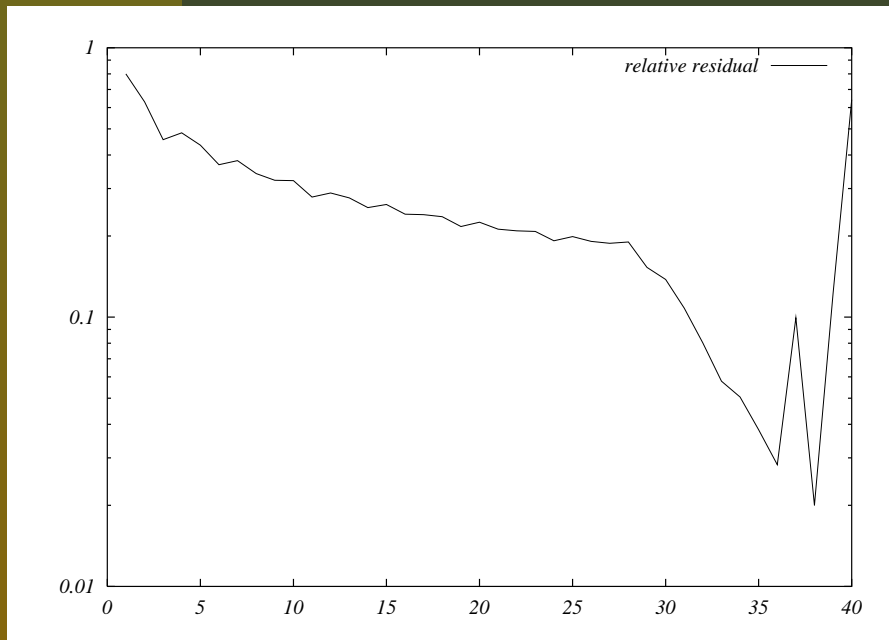
- A が対称かつ正則ならば , 共役勾配法はほとんどすべての初期ベクトル x_0 に対して**高々 N 回**の反復で厳密解を与える .

丸め誤差の影響

共役勾配法は，特に前処理を施すことにより次元数よりもはるかに少ない回数で十分な精度を得られる場合もある反面，停滞したり，ある回数を超えると収束が速くなったりなど，ムラの多いことで知られている．

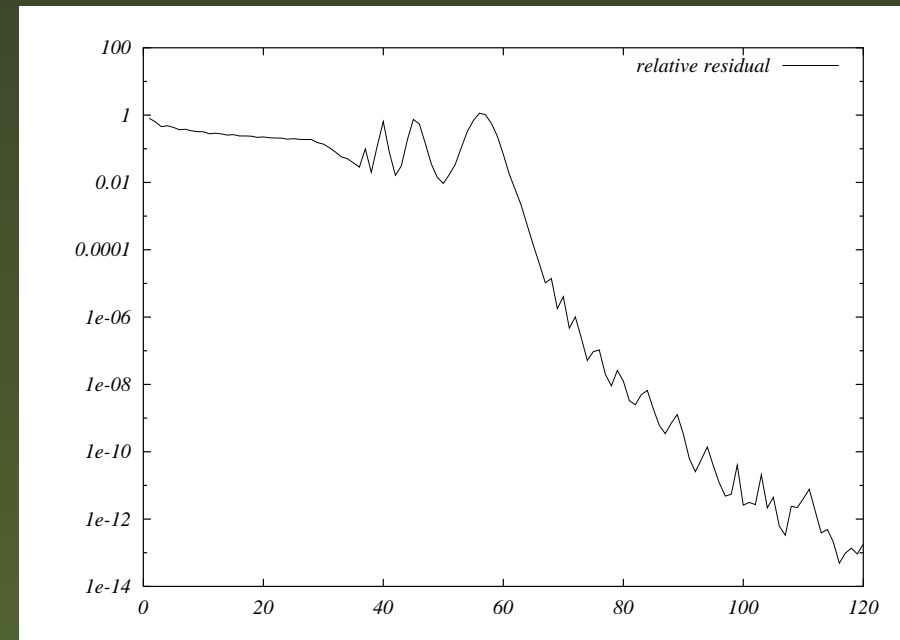
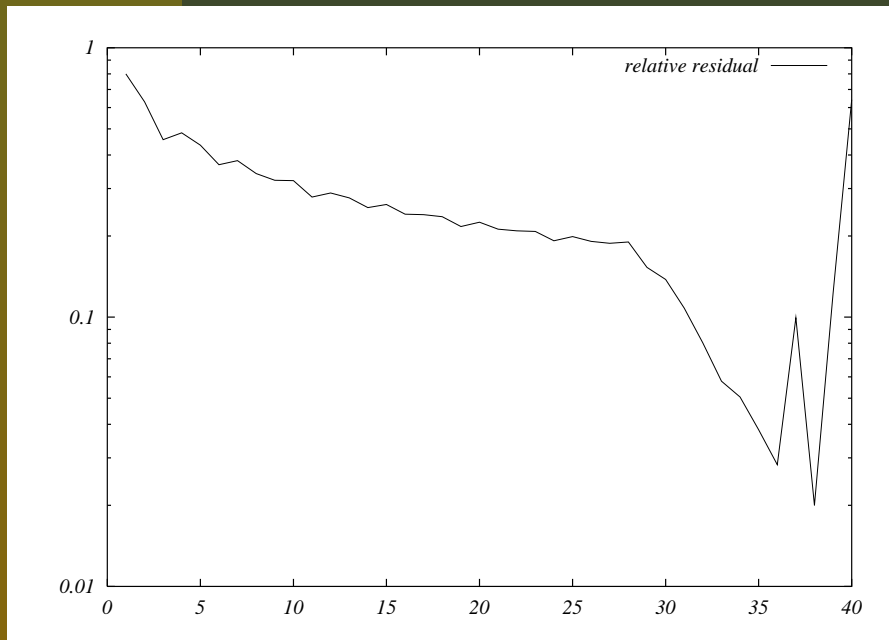
丸め誤差の影響

共役勾配法は，特に前処理を施すことにより次元数よりもはるかに少ない回数で十分な精度を得られる場合もある反面，停滞したり，ある回数を超えると収束が速くなったりなど，ムラの多いことで知られている．



丸め誤差の影響

共役勾配法は，特に前処理を施すことにより次元数よりもはるかに少ない回数で十分な精度を得られる場合もある反面，停滞したり，ある回数を超えると収束が速くなったりなど，ムラの多いことで知られている．



疑問

疑問

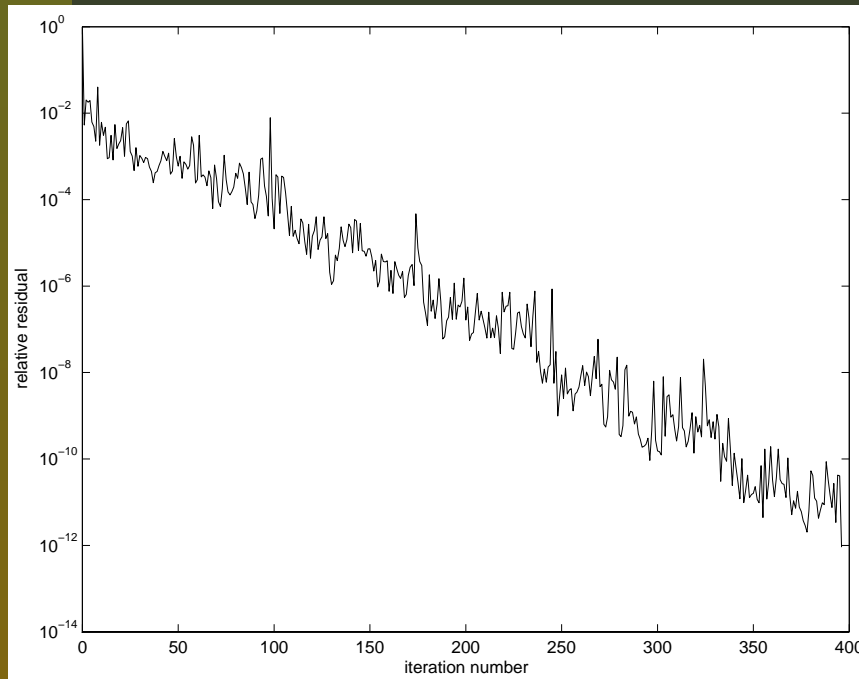
- 本当に N 回の反復で厳密解に到達するのか？

疑問

- 本当に N 回の反復で厳密解に到達するのか？
- 桁数を多く取るとどうなるか？

疑問

- 本当に N 回の反復で厳密解に到達するのか？
- 桁数を多く取るとどうなるか？



この「振動」は丸め誤差によるものか？

区間演算で共役勾配法を実行すると？

区間演算で共役勾配法を実行すると？

$$A = \begin{bmatrix} 2 & 1 & & & \\ 0 & 2 & 1 & & \\ \gamma & 0 & 2 & 1 & \\ & \gamma & 0 & 2 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \quad \mathbf{b} = [1, \dots, 1]^T$$

区間演算で共役勾配法を実行すると？

$$A = \begin{bmatrix} 2 & 1 & & & \\ 0 & 2 & 1 & & \\ \gamma & 0 & 2 & 1 & \\ & \gamma & 0 & 2 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \quad \mathbf{b} = [1, \dots, 1]^T$$

N	verifylss	区間CG法
10	2.2204e-16	10回目に破綻
20	2.2204e-16	10回目に破綻
100	2.2204e-16	12回目に破綻
1000	2.2204e-16	12回目に破綻

区間演算で共役勾配法を実行すると？

$$A = \begin{bmatrix} 2 & 1 & & & \\ 0 & 2 & 1 & & \\ \gamma & 0 & 2 & 1 & \\ & \gamma & 0 & 2 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \quad \mathbf{b} = [1, \dots, 1]^T$$

N	verifylss	区間CG法
10	2.2204e-16	10回目に破綻
20	2.2204e-16	10回目に破綻
100	2.2204e-16	12回目に破綻
1000	2.2204e-16	12回目に破綻



失敗！

多倍長計算

- 今井 仁司:
“応用解析における多倍長計算”
数学, Vol.55, No.3, pp.316–325 (2003).
- 幸谷 智紀:
“CG法の最短時間を探索する試み”
情報処理学会研究報告 2003-HPC-95, pp.1–6
(2003).

FMLIB

多倍長 Fortran モジュール FMLIB 1.2

- www.lmu.edu/acad/personal/faculty/dmsmith2/FMLIB.html
- 四則演算や Fortran の組込み関数の多くの多倍長演算が利用可能
- 計算環境
 - FUJITSU GP7000F model 900
 - Solaris 7
 - Fujitsu Fortran Compiler Driver Version 5.1

プログラム例

```
subroutine MATRIX_AND_VECTOR_MULTIPLY(A,b,x,N,DIGIT)
  use FMZM
  implicit none
  integer,intent(IN) :: N,DIGIT
  type(FM),dimension(N,N),intent(IN) :: A
  type(FM),dimension(N),intent(IN) :: b
  type(FM),dimension(N),intent(OUT) :: x
  integer :: i,j
  call FM_SET(DIGIT)
  do i=1,N
    x(i)=TO_FM(0)
    do j=1,N
      x(i)=x(i)+A(i,j)*b(j)
    end do
  end do
end subroutine MATRIX_AND_VECTOR_MULTIPLY
```

行列とベクトルの積に要する時間 ($N = 200$; ミリ秒)

演算	桁数	時間
単精度演算	約 6	1
倍精度演算	約 14	2
4 精度演算	約 33	40
多倍長演算	10	1338
	100	1337
	1000	1442
	10000	2519

実験 I

対称行列に対する共役勾配法を N 回反復させた時、
本当に真の解に到達するのか？

Hilbert 行列

$$A_{ij} = \frac{1}{i + j - 1}$$

- 正定値対称行列
- 条件数が約 $\exp(3.5N)$
- $b_i = \sum_{j=1}^N A_{ij}$
- 完全ピボット選択付き Gauss の消去法による直接解法 (4 倍精度) の計算結果とも比較

N 回反復時の残差/誤差

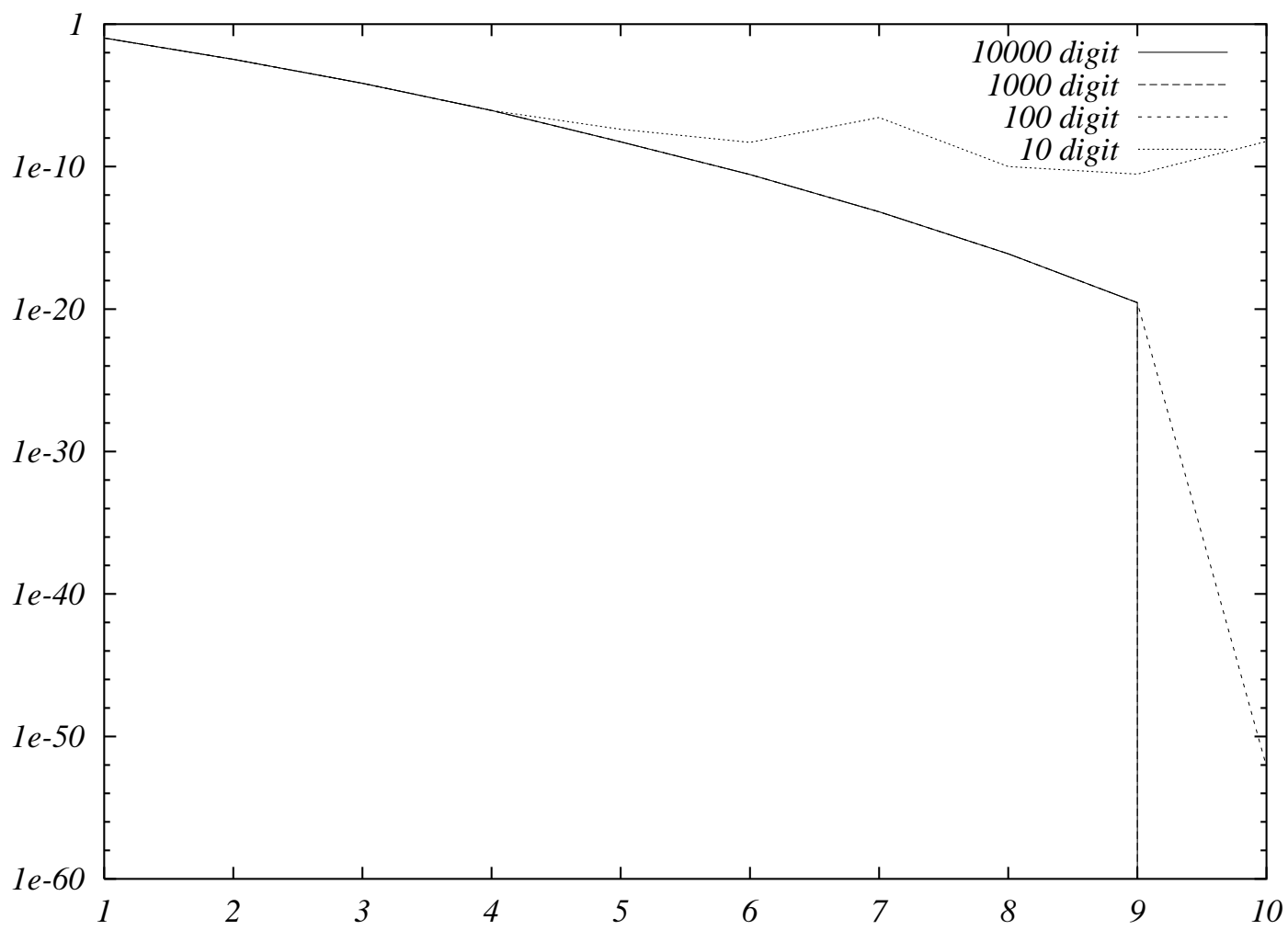
N	10桁	50桁	100桁	1000桁	5000桁	10000桁	LU 分解
10	0.57E-8	0.82E-19	0.76E-52	0.39E-955	0.11E-4951	0.98E-9950	—
	0.88E-4	0.75E-6	0.12E-51	0.65E-955	0.19E-4951	0.16E-9949	2.95E-22
20	0.51E-9	0.20E-21	0.39E-23	0.60E-779	0.74E-4776	0.31E-9773	—
	0.20E-3	0.39E-7	0.65E-10	0.11E-778	0.14E-4775	0.58E-9773	1.25E-7
30	0.24E-12	0.30E-23	0.65E-25	0.30E-482	0.19E-4480	0.10E-9476	—
	0.44E-4	0.38E-8	0.98E-11	0.61E-482	0.38E-4480	0.20E-9476	×
40	0.43E-13	0.15E-27	0.46E-32	0.49E-65	0.36E-4062	0.58E-9060	—
	0.32E-4	0.11E-8	0.13E-11	0.28E-42	0.75E-4062	0.12E-9059	×
50	0.26E-12	0.67E-23	0.22E-33	0.44E-91	0.25E-3524	0.35E-8523	—
	0.23E-4	0.25E-9	0.96E-13	0.11E-29	0.54E-3524	0.76E-8523	×
100	0.76E-9	0.46E-26	0.66E-35	0.11E-106	0.25E-203	0.31E-4029	—
	0.52E-1	0.34E-10	0.12E-13	0.65E-37	0.50E-67	0.75E-4029	×

相対残差 (上段) と相対誤差 (下段)

LU 分解は 4 倍精度

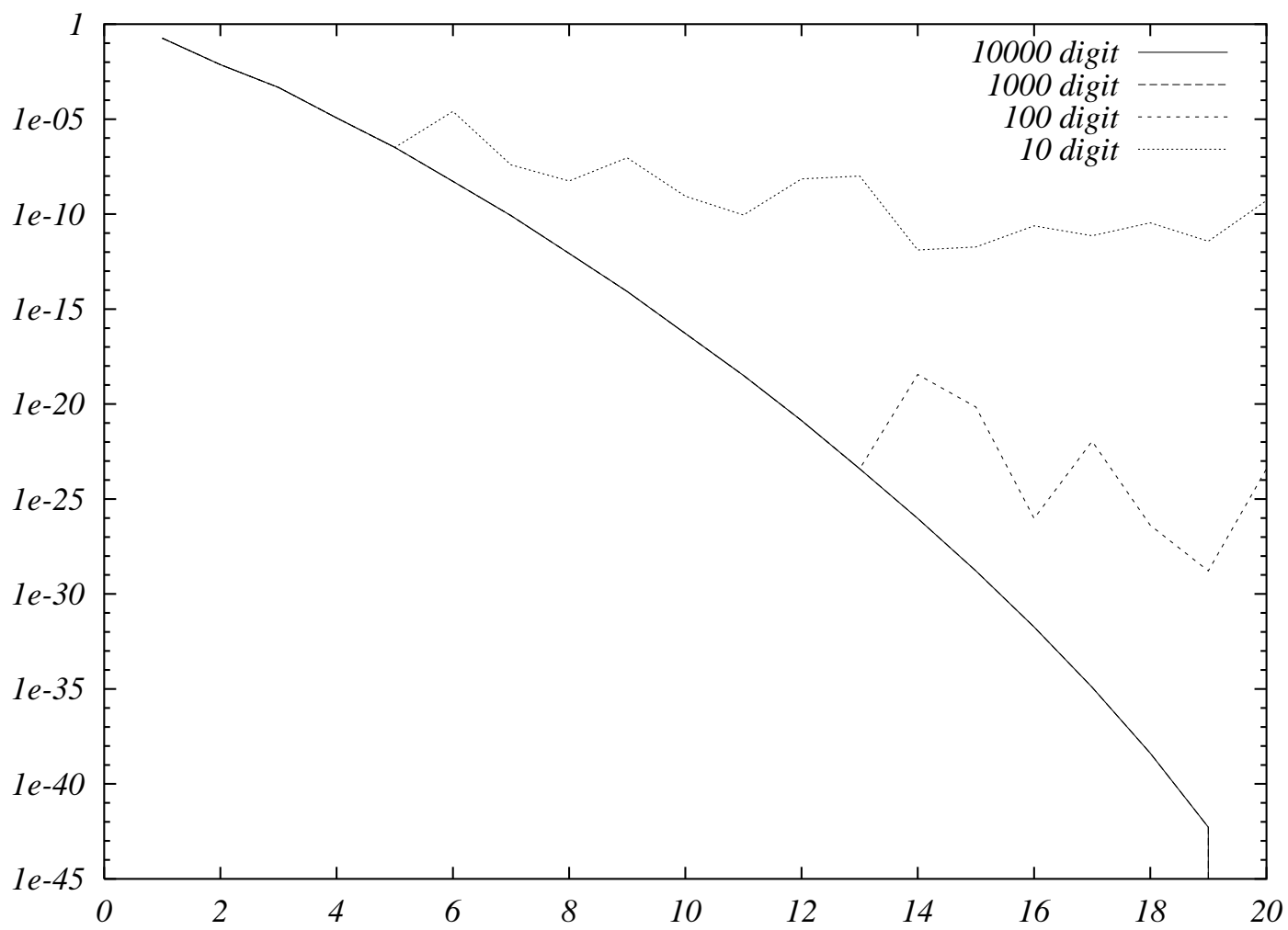
相对残差履歴

相对残差履歴



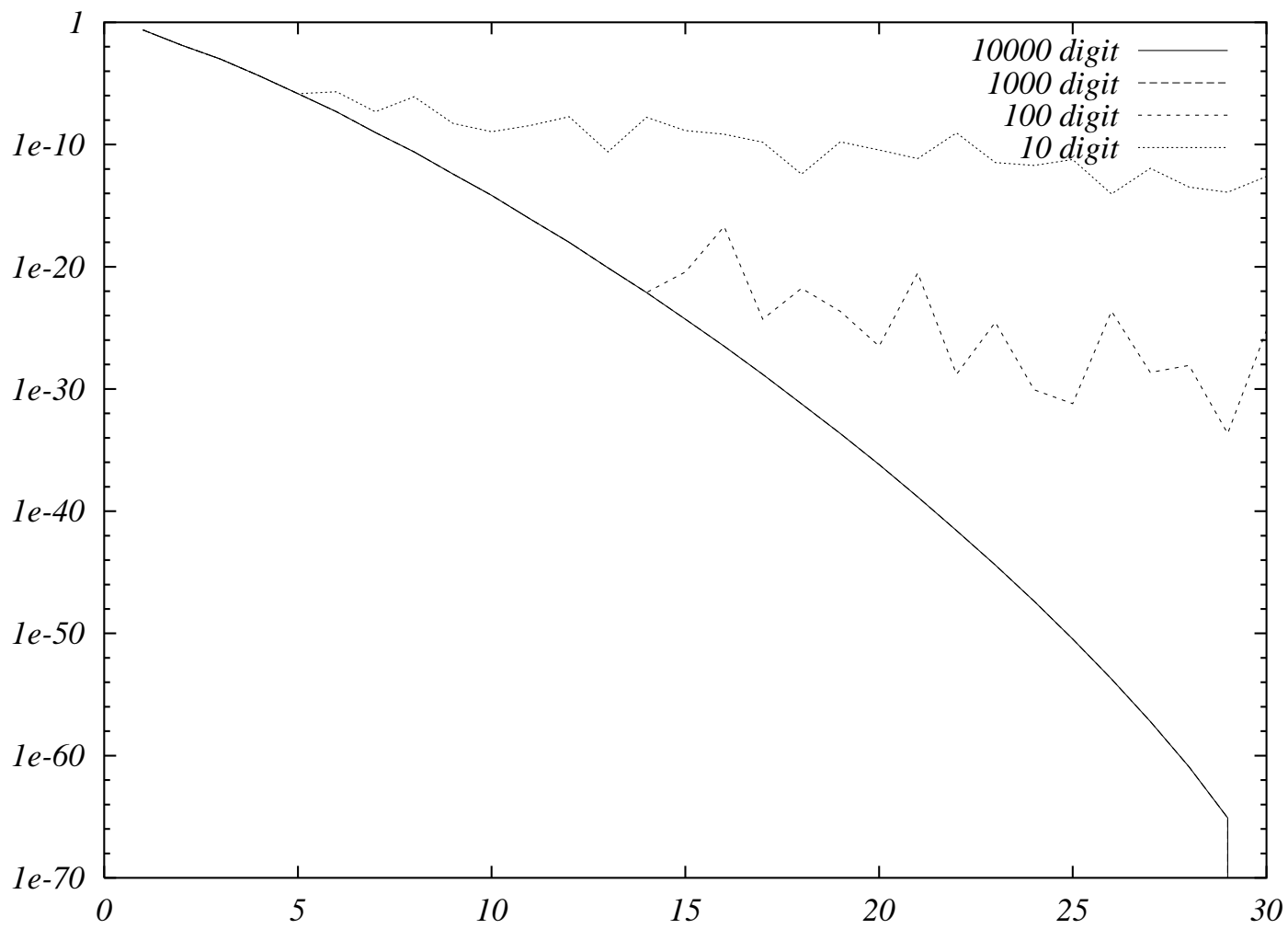
$$N = 10$$

相对残差履歷



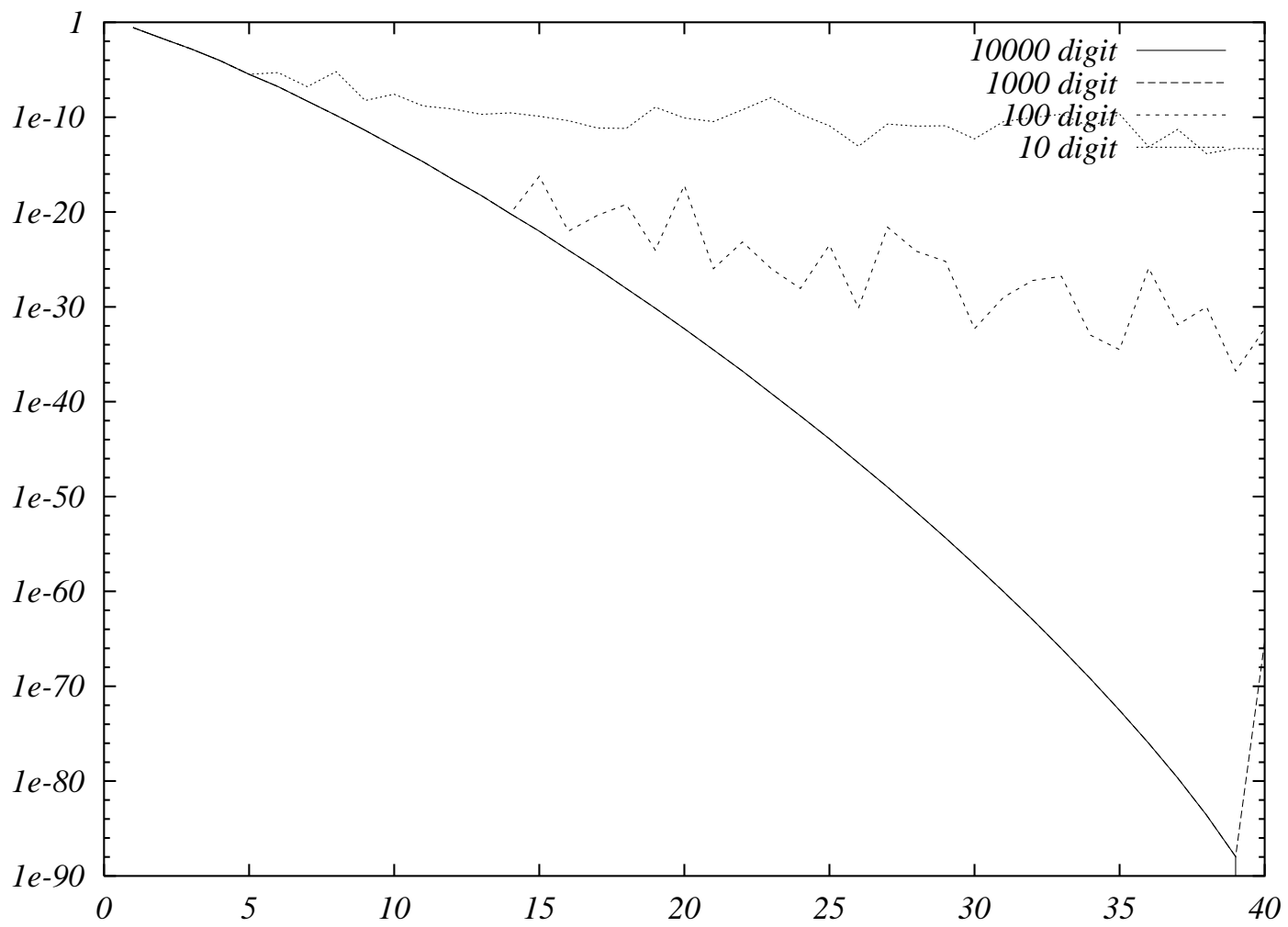
$N = 20$

相对残差履歷



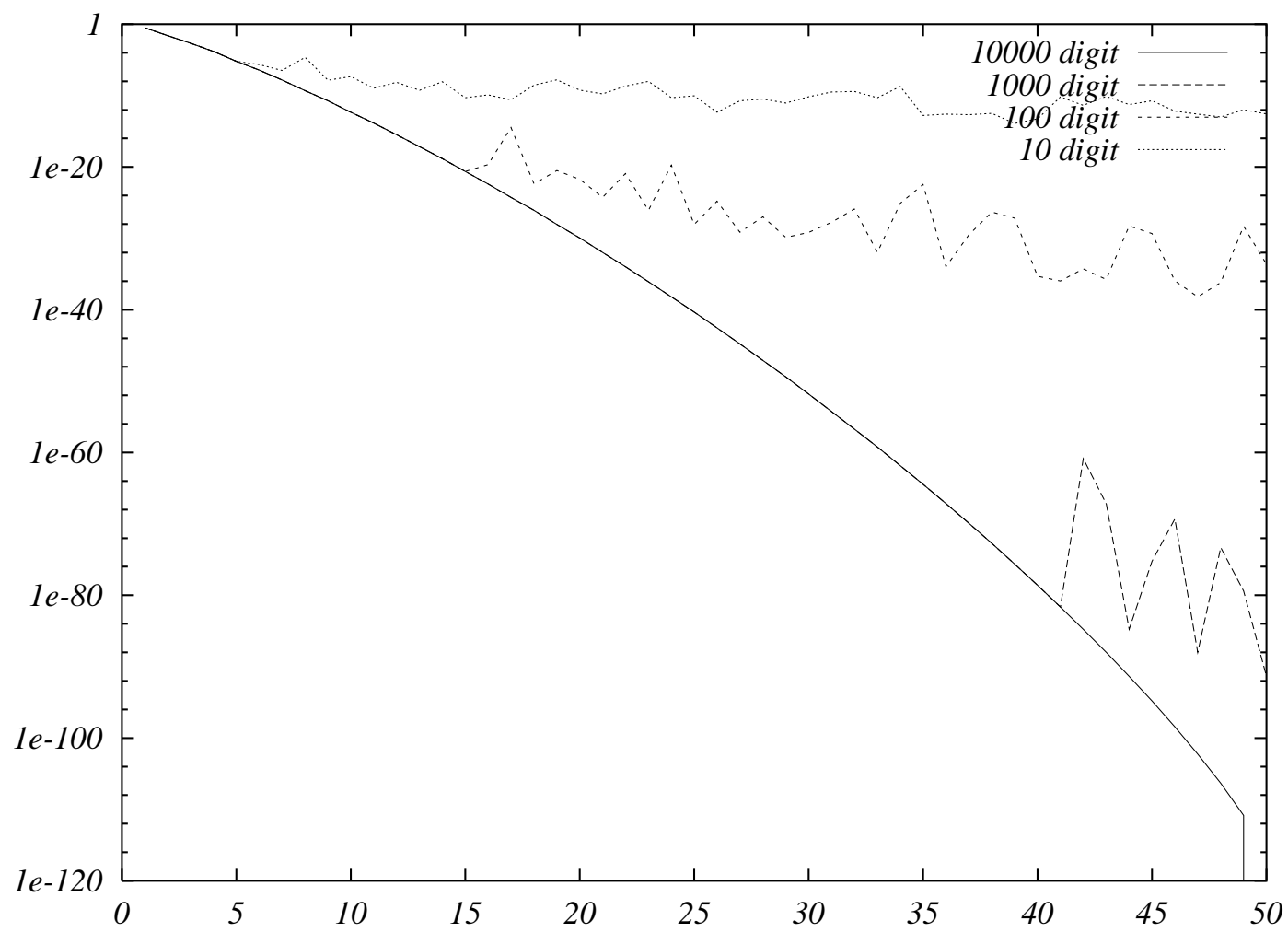
$$N = 30$$

相对残差履歷



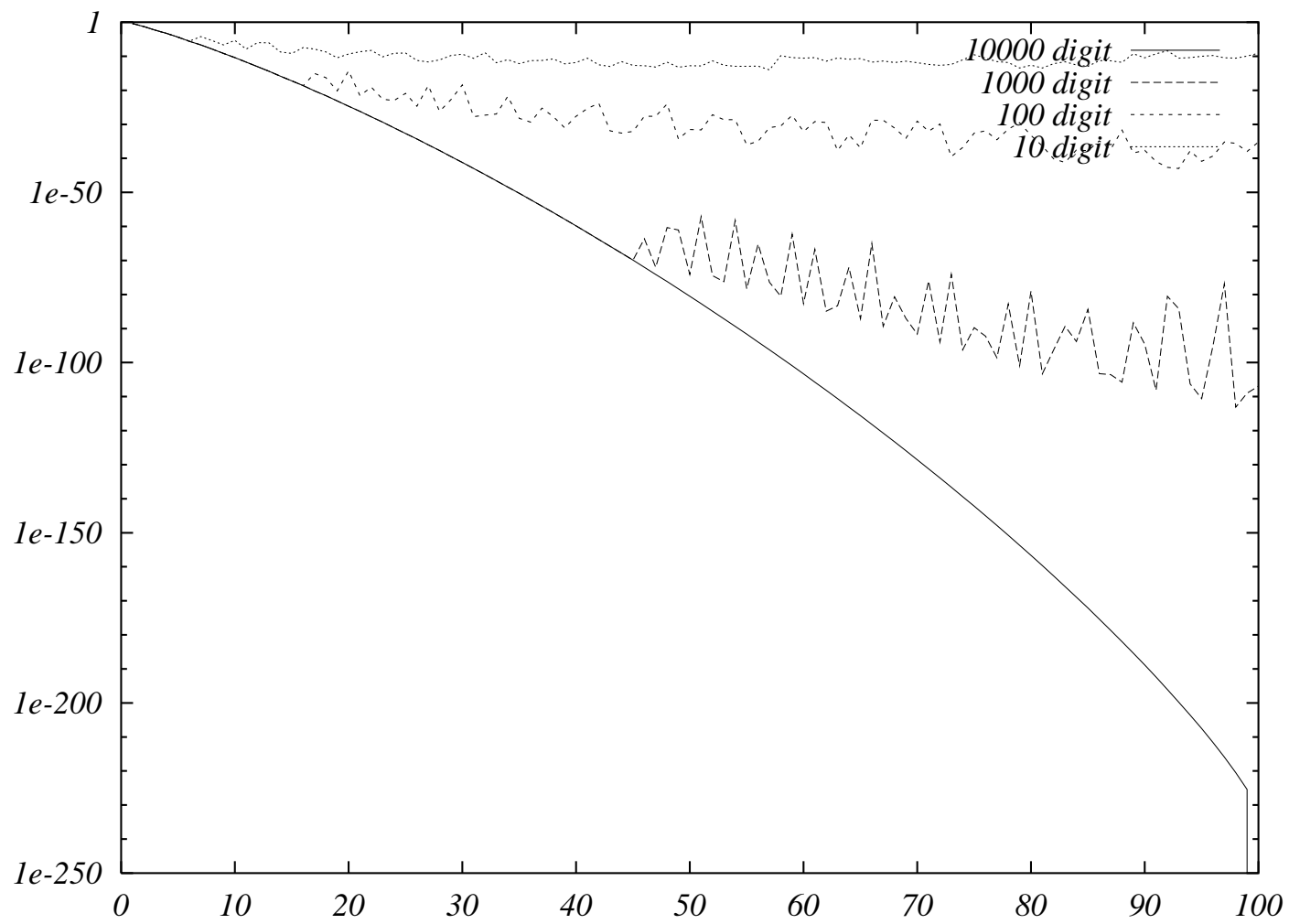
$$N = 40$$

相对残差履歴



$N = 50$

相对残差履歴



$$N = 100$$

実験II

弾性梁の曲げに関する行列

$$A = \begin{bmatrix} 5 & -4 & 1 & & & & & & & & \\ -4 & 6 & -4 & 1 & & & & & & & \\ 1 & -4 & 6 & -4 & 1 & & & & & & \\ & 1 & -4 & 6 & -4 & 1 & & & & & \\ & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ & & & & 1 & -4 & 6 & -4 & 1 & & \\ & & & & & 1 & -4 & 6 & -4 & 1 & \\ & & & & & & 1 & -4 & 6 & -4 & \\ & & & & & & & 1 & -4 & 5 & \end{bmatrix}$$

- $\mathbf{b} = [1, 0, \dots, 0]^T$
- 真の解 $x = [x_j]$ は

$$x_j = \frac{j}{6(N+1)}(N+1-j)(2N+2-j)$$

N 回反復時の残差/誤差

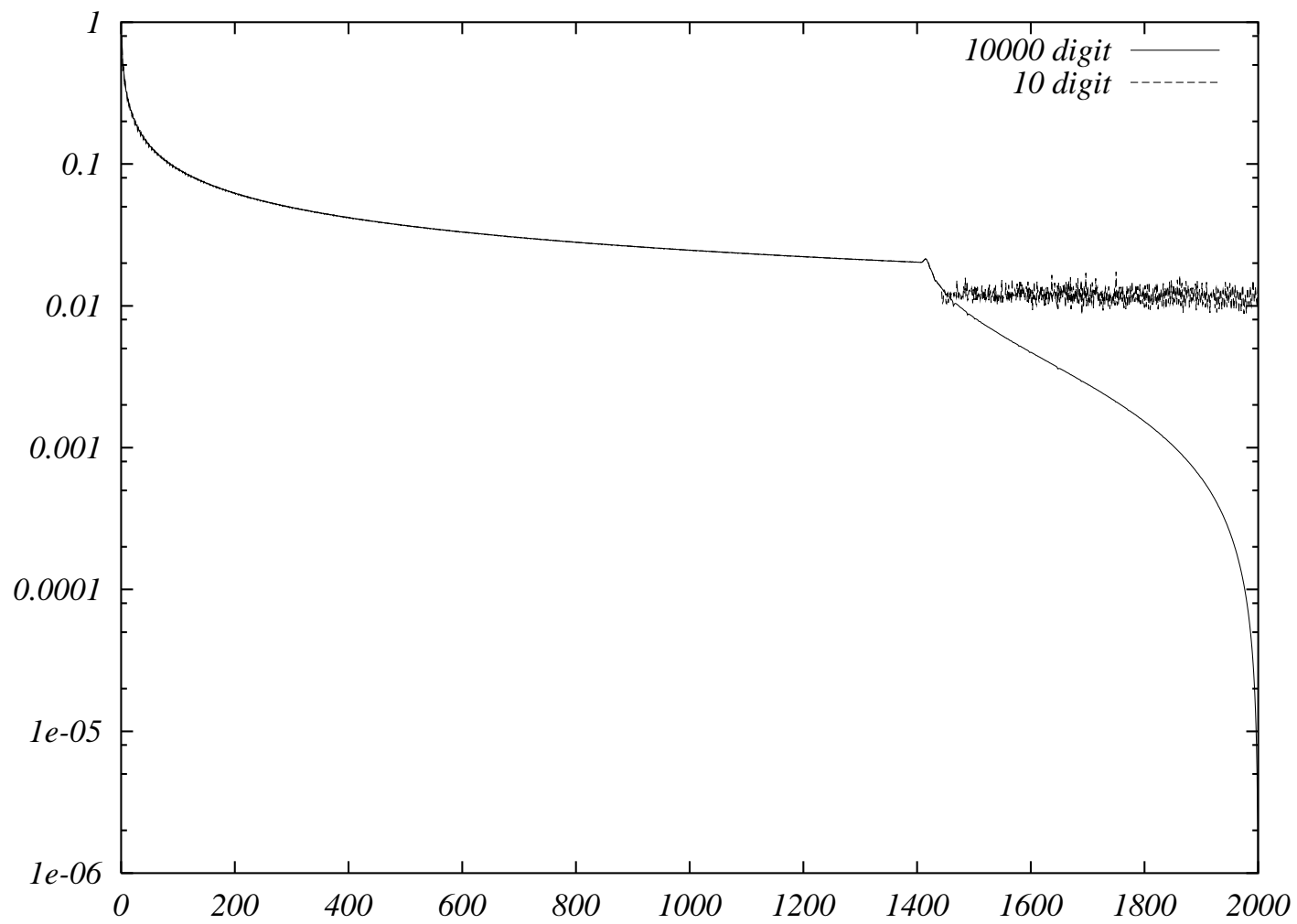
N	10桁	50桁	100桁	500桁	1000桁	10000桁	LU 分解
10	0.54E-9	0.13E-50	0.37E-99	0.11E-498	0.13E-1002	0.14E-9997	—
	0.64E-11	0.13E-52	0.34E-101	0.12E-500	0.13E-1004	0.12E-9999	0.14E-33
20	0.23E-1	0.42E-43	0.48E-92	0.23E-491	0.37E-995	0.18E-9990	—
	0.10E-3	0.10E-45	0.11E-94	0.63E-494	0.89E-998	0.47E-9993	0.55E-31
100	0.32E-1	0.38E+0	0.26E-31	0.20E-430	0.48E-934	0.13E-9929	—
	0.99E+0	0.10E+1	0.25E-35	0.19E-434	0.46E-938	0.13E-9933	0.62E-27
200	0.42E-1	0.82E-1	0.29E-2	0.53E-354	0.41E-858	0.33E-9853	—
	0.10E+1	0.10E+1	0.10E+1	0.12E-358	0.10E-862	0.81E-9858	0.27E-25
1000	0.20E-1	0.13E-1	0.14E-1	0.97E-1	0.18E-246	0.32E-9241	—
	0.10E+1	0.10E+1	0.10E+1	0.99E+0	0.18E-252	0.31E-9247	0.58E-21
2000	0.14E-1	0.14E-1	0.81E-2	0.24E-1	0.42E-1	0.25E-8476	—
	0.10E+1	0.10E+1	0.10E+1	0.10E+1	0.10E+1	0.61E-8483	0.37E-19

相対残差 (上段) と相対誤差 (下段)

LU 分解は 4 倍精度

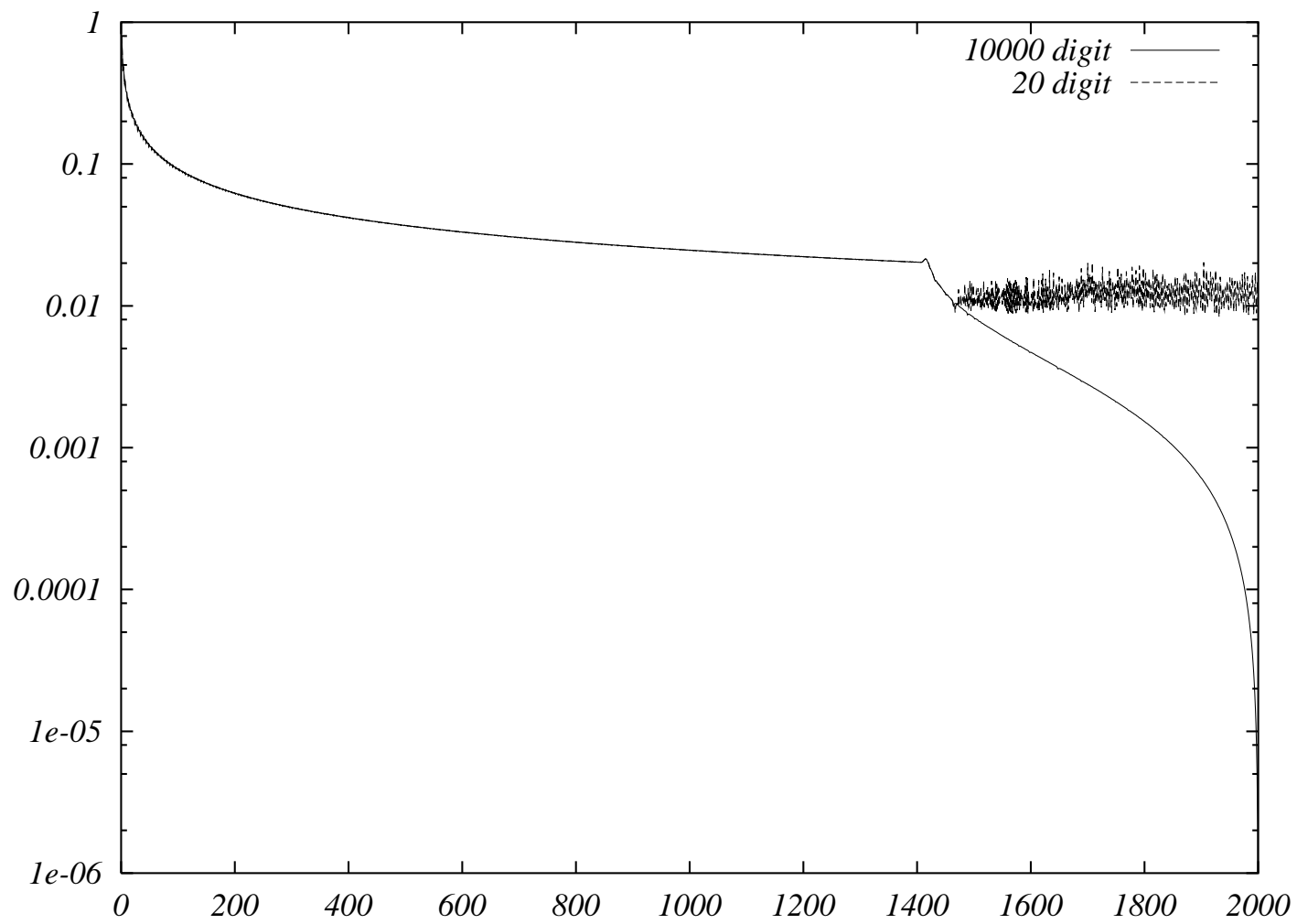
相对残差履歷

相对残差履歴



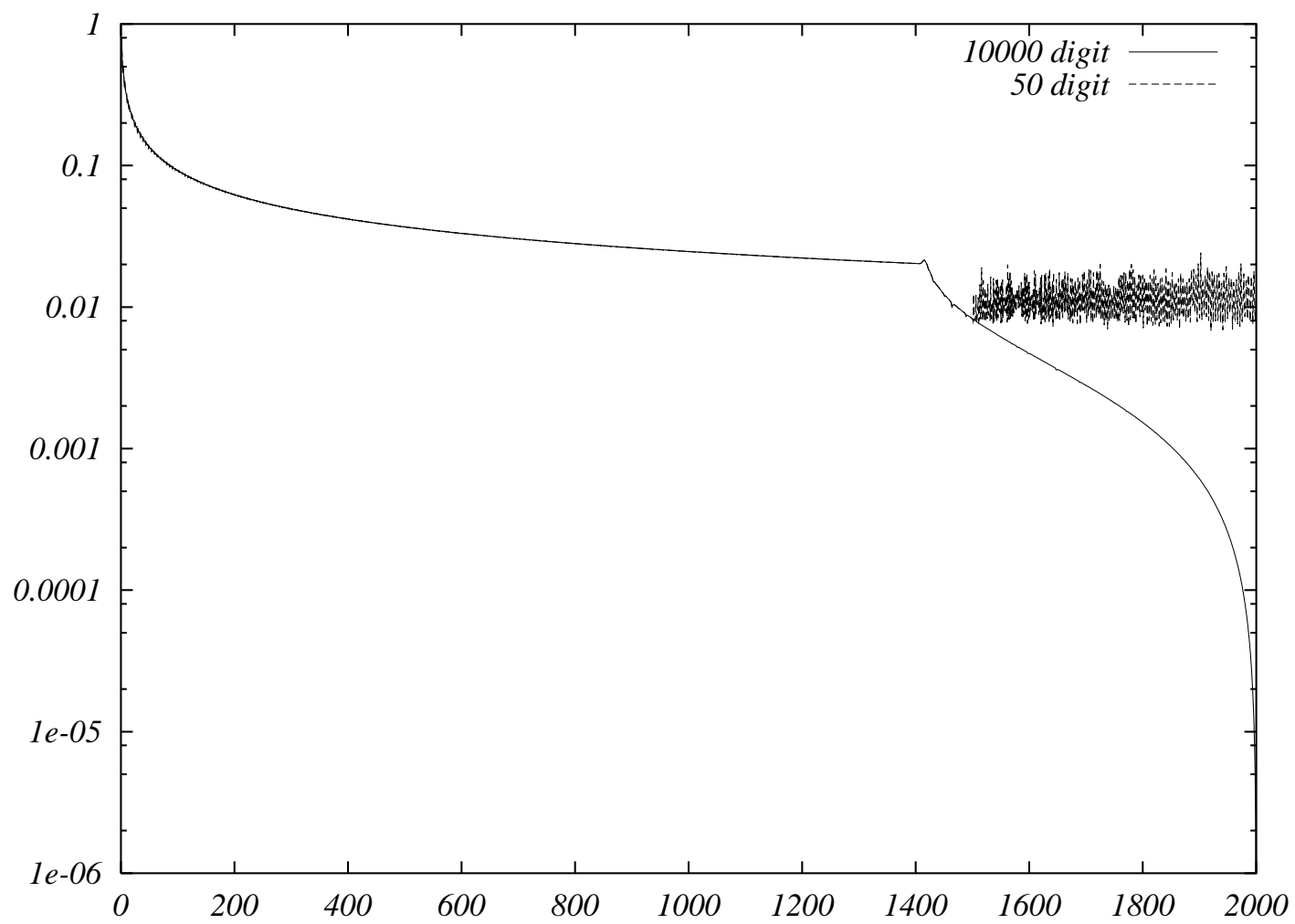
10000桁と10桁

相对残差履歴



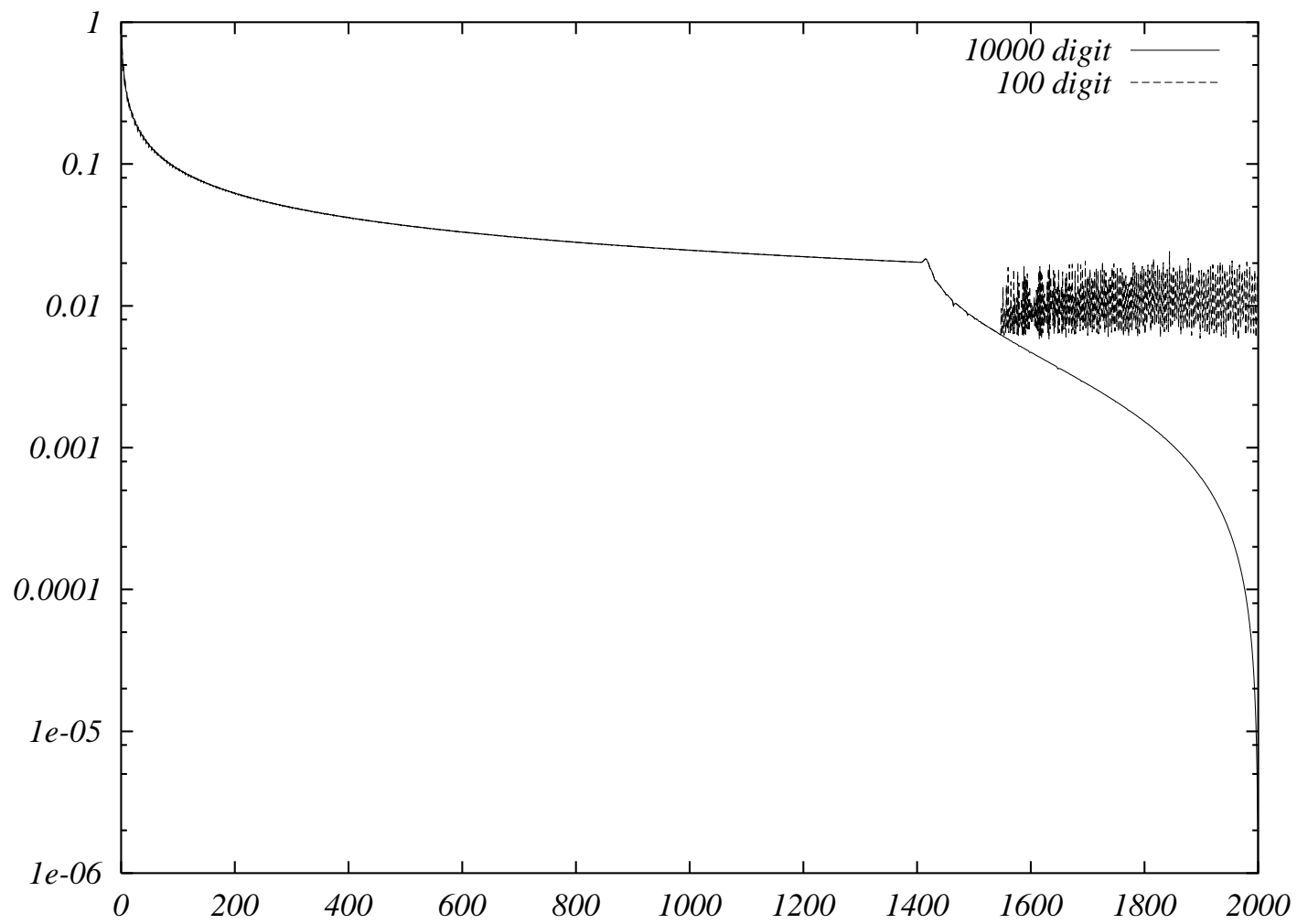
10000桁と20桁

相对残差履歴



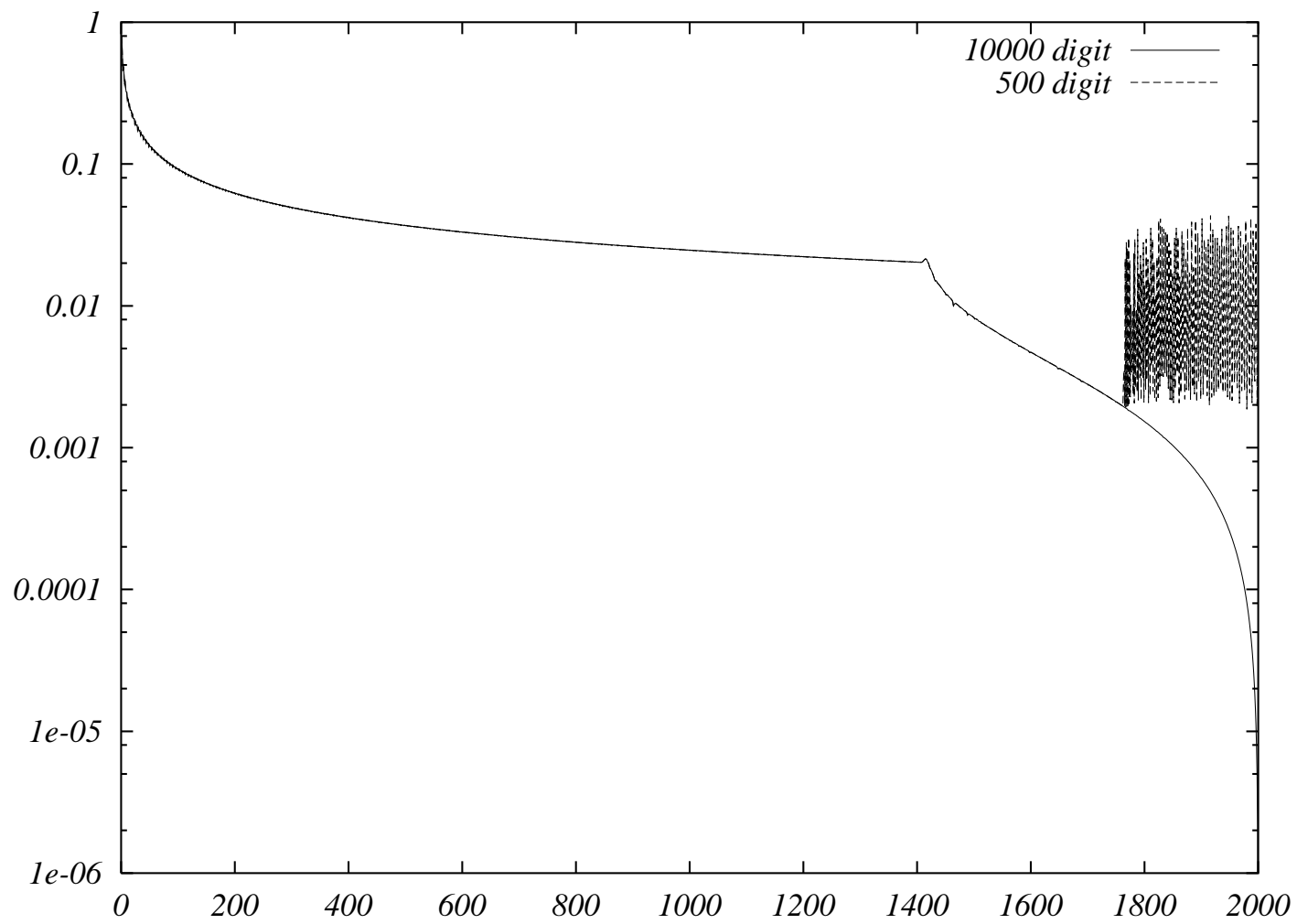
10000桁と50桁

相对残差履歴



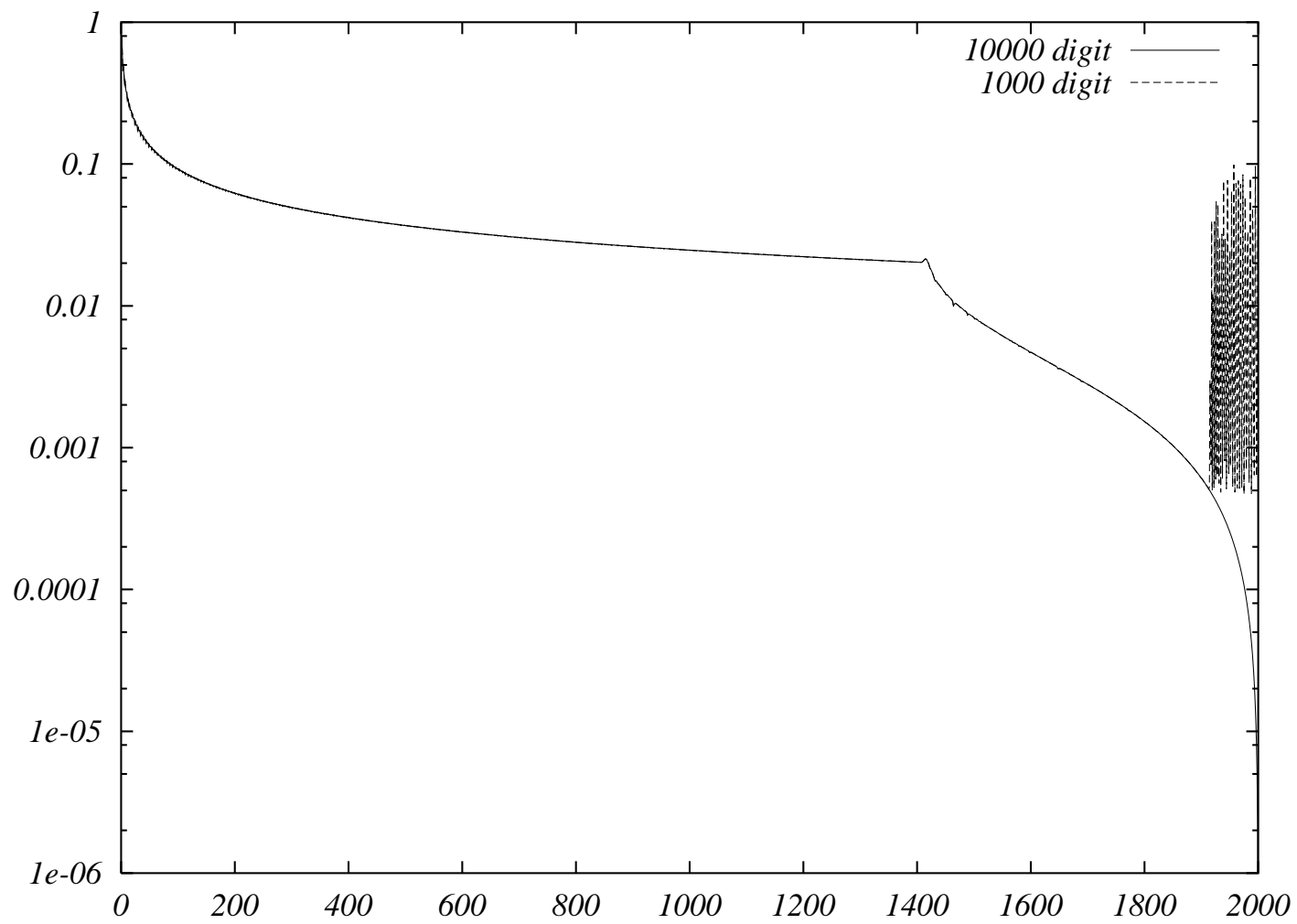
10000桁と100桁

相对残差履歴



10000桁と500桁

相对残差履歴



10000桁と1000桁

実験 III

Harwell-Boeing Sparse Matrix Collection

<http://math.nist.gov/MatrixMarket/>

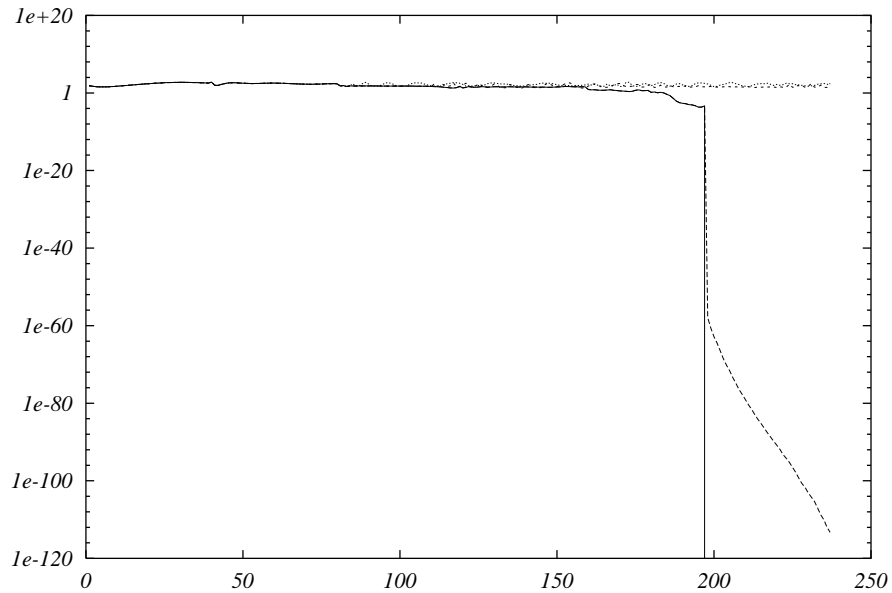
- LANPRO (NOS1)
 - Finite element approximations to problems in structural engineering
 - Biharmonic operator on a beam with one end free and one end fixed
- positive definite matrix
- 237×237
- $\mathbf{b} = [1, \dots, 1]^T$

次数回反復後の相対残差

	10桁	20桁	50桁	100桁	1000桁	10000桁
前処理なし	0.11E+3	0.29E+3	0.34E+2	0.30E+3	0.69E-617	0.24E-9612
対角スケールング	0.10E+2	0.31E-1	0.23E-26	0.49E-75	0.20E-978	0.19E-9973

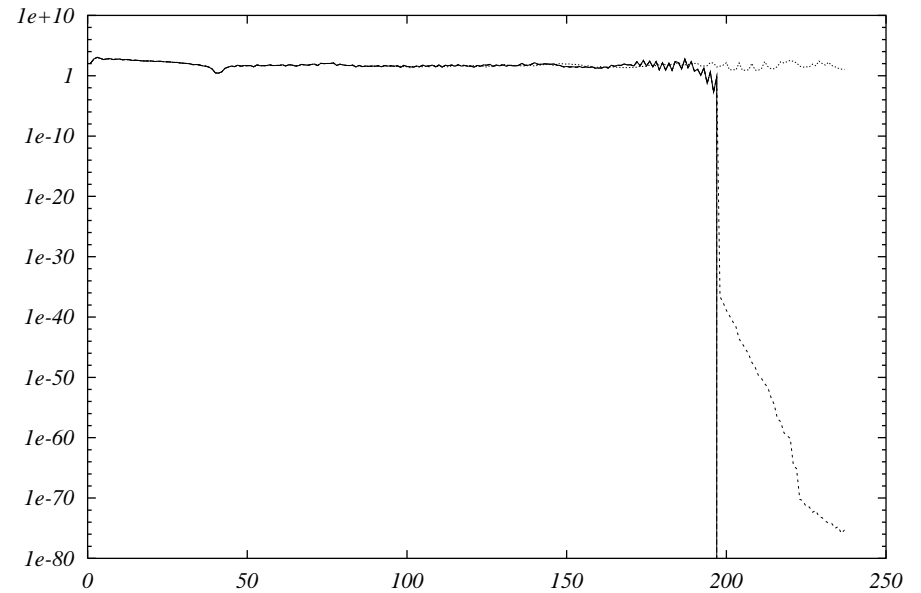
次数回反復後の相対残差

	10桁	20桁	50桁	100桁	1000桁	10000桁
前処理なし	0.11E+3	0.29E+3	0.34E+2	0.30E+3	0.69E-617	0.24E-9612
対角スケーリング	0.10E+2	0.31E-1	0.23E-26	0.49E-75	0.20E-978	0.19E-9973



1000 digit ——— 500 digit - - - - - 100 digit 10 digit - · - - - -

前処理なし



1000 digit ——— 500 digit - - - - - 100 digit 10 digit - · - - - -

対角スケーリング

平面 Poiseuille 流れの安定性

平面 Poiseuille 流れの安定性

$$\left\{ \begin{array}{l} (-D^2 + a^2)^2 u + iaR[V(-D^2 + a^2) + V'']u = \lambda(-D^2 + a^2)u \\ u(x_1) = u(x_2) = u'(x_1) = u'(x_2) = 0 \end{array} \right.$$

平面 Poiseuille 流れの安定性

$$\left\{ \begin{array}{l} (-D^2 + a^2)^2 u + iaR[V(-D^2 + a^2) + V'']u = \lambda(-D^2 + a^2)u \\ u(x_1) = u(x_2) = u'(x_1) = u'(x_2) = 0 \end{array} \right.$$

平面 Poiseuille 流れ:

$$V = 1 - x^2, \quad x_1 = -1, \quad x_2 = 1.$$

平面 Poiseuille 流れの安定性

$$\begin{cases} (-D^2 + a^2)^2 u + iaR[V(-D^2 + a^2) + V'']u = \lambda(-D^2 + a^2)u \\ u(x_1) = u(x_2) = u'(x_1) = u'(x_2) = 0 \end{cases}$$

平面 Poiseuille 流れ:

$$V = 1 - x^2, \quad x_1 = -1, \quad x_2 = 1.$$

基本流の安定性

λ の実部が正ならば流れ関数 ψ は減衰するため安定,
負ならば不安定.

Orr-Sommerfeld 方程式の特徴

$$(-D^2 + a^2)^2 u + iaR[V(-D^2 + a^2) + V'']u = \lambda(-D^2 + a^2)u$$

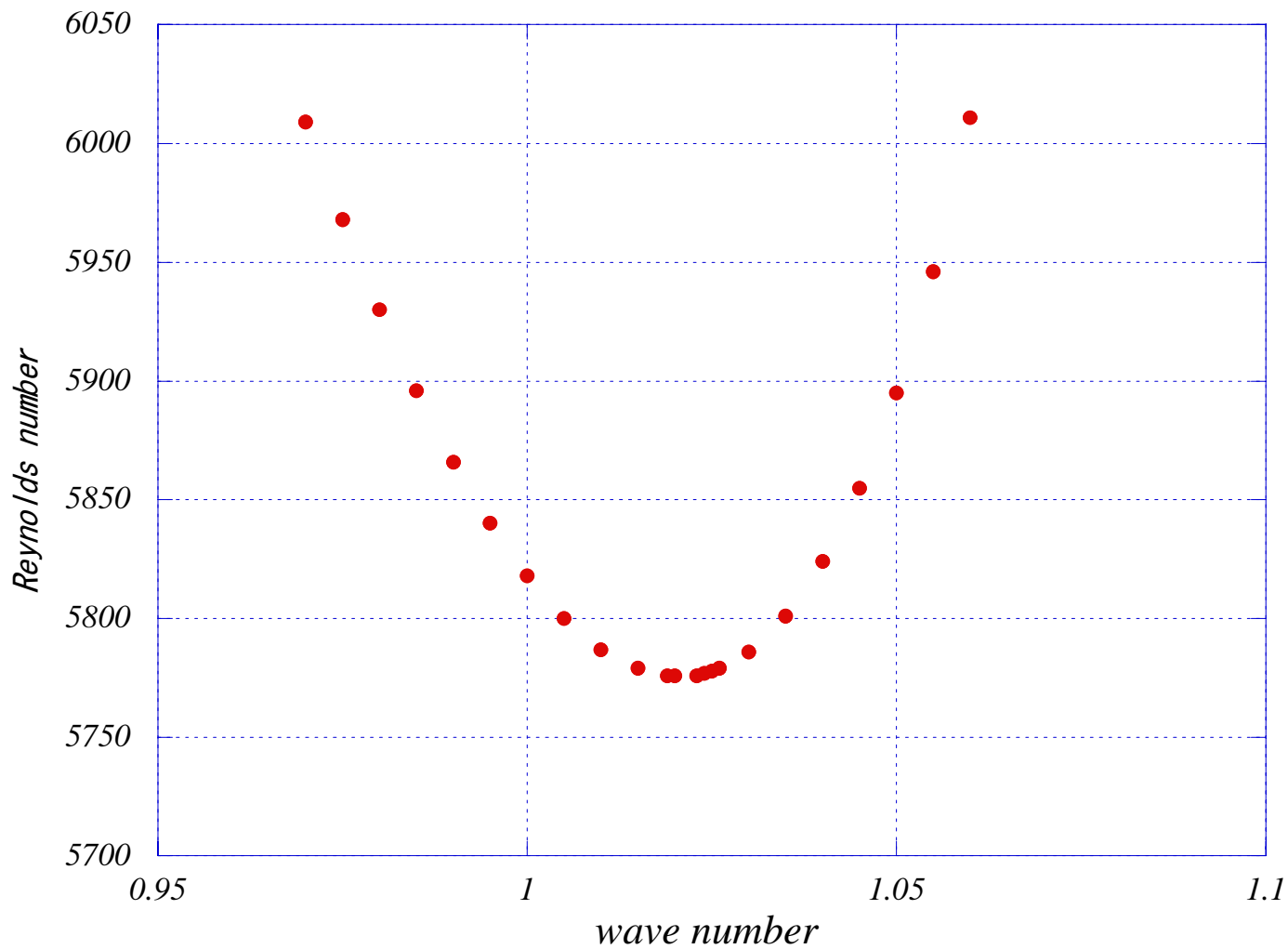
- 4 階微分方程式
- 複素固有値問題
- 2 階微分の項を含む
- 非自己共役
- 境界条件: $u(-1) = u(1) = u'(-1) = u'(1) = 0$
- Reynolds 数 R : 5000 ~ 6000
 - 丸め誤差の影響

計算環境

- Sun マイクロシステムズ社 Forte Fortran
<http://www.sun.co.jp/workshop/>
- 4倍精度区間変数

コンパイラ	Forte Fortran Desktop Edition 6 update 1
OS	SunOS 5.7
計算機	FUJITSU GP7000F model 900 SUN Ultra5 model 360

臨界曲線



$Re(\lambda) < 0$ が検証された $[a, R]$

まとめ

まとめ

- 多倍長計算環境は(比較的)簡単に構築できる

まとめ

- 多倍長計算環境は(比較的)簡単に構築できる
- プログラムの移植は容易 (組込み関数の自作がなければ)

まとめ

- 多倍長計算環境は(比較的)簡単に構築できる
- プログラムの移植は容易 (組込み関数の自作がなければ)
- 実験数学の道具として有効?

まとめ

- 多倍長計算環境は(比較的)簡単に構築できる
- プログラムの移植は容易 (組込み関数の自作がなければ)
- 実験数学の道具として有効?
- 丸め誤差を (できるだけ) 取り除いた世界が見える

まとめ

- 多倍長計算環境は(比較的)簡単に構築できる
- プログラムの移植は容易 (組込み関数の自作がなければ)
- 実験数学の道具として有効?
- 丸め誤差を (できるだけ) 取り除いた世界が見える
- 現状, 計算時間がネック