Contrôle dynamique de méthodes d'approximation

Fabienne Jézéquel

Habilitation à Diriger des Recherches

Laboratoire d'Informatique de Paris 6

Outline

1. Stochastic approach of round-off errors
   - Principles of stochastic arithmetics
   - Implementation of Discrete Stochastic Arithmetic
2. Dynamical control of approximations methods
   - Using sequences with a linear convergence
   - Using combined sequences
3. Dynamical control of multiple integrals computation
   - Using the principle of "iterated integrals"
   - Using cubature methods
4. Applications
   - In crystallography
   - In the neutron star theory

Round-off error analysis - 1/2
Several approaches

- Direct analysis
  estimation or bound of the direct error
  running error analysis, J.H. Wilkinson 1971
  SCALP, Ph. François 1989
- Inverse analysis
  based on the “ Wilkinson principle”: the computed solution is assumed to be the exact solution of a nearby problem
  LAPACK, E. Anderson et al. 1999
  PRECISE, F. Chaitin-Chatelin et al. 2000
- Methods based on algorithmic differentiation
  first order approximation of the global round-off error
  the CENA method, Ph. Langlois 2001

Round-off error analysis - 2/2

Several approaches

- Interval arithmetic
  guaranted bounds for each computed result
  XSC languages, U. Kulisch et al. 1990
  INTLAB, S.M. Rump 1998
  MPFI, N. Revol and F. Rouillier 2003
- Probabilistic approach
  uses a random rounding mode
  the CESTAC method, M. La Porte and J. Vignes 1974
  Monte Carlo Arithmetic, D.S. Parker 1997

The CESTAC method

- each arithmetical operation is performed N times using the random rounding mode
  $\Rightarrow$ for each arithmetical operation, N results R_i are computed.
- computed result: $\overline{R} = \frac{1}{N}\sum_{i=1}^{N} R_i$.
- the number $C_{\overline{R}}$ of exact significant digits is estimated by

$$C_{\overline{R}} = log_{10}\left(\frac{\sqrt{N}\ \left|\overline{R}\right|}{s\ \tau_\beta}\right) \quad \text{with} \quad s^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(R_i - \overline{R}\right)^2$$

τ_β being the value of the Student distribution for $N-1$ degrees of freedom and a probability level $(1-\beta)$.

In practice, $N = 2$ or $N = 3$ and $\beta = 0.05$.

The concept of computational zero

J. Vignes, 1986

Definition

During the run of a code using the CESTAC method, a result R is a computational zero, denoted by @.0, if

$$\forall i, R_i = 0 \text{ or } C_{\overline{R}} \leq 0.$$

- synchronous implementation of the CESTAC method
- concept of computational zero

$\Rightarrow$ continuous or discrete stochastic arithmetic

Continuous stochastic arithmetic

Using the CESTAC method, the results of each arithmetical operation can be considered as realizations of a Gaussian random variable.

Definition

A stochastic number X is a Gaussian random variable denoted by (m, σ^2), where m is the mean value of X and σ its standard deviation.

The number of significant digits common to all the elements of the confidence interval of m at $1-\beta$ and to m is lower bounded by

$$C_{\beta,X} = \log_{10}\left(\frac{|m|}{\lambda_\beta \sigma}\right)$$

with $\beta = 0.05$, $\lambda_\beta \approx 1.96$.

Stochastic arithmetical operations

Definition

We define the elementary operations on two stochastic numbers $X_1 = (m_1, \sigma_1^2)$ and $X_2 = (m_2, \sigma_2^2)$ by:

- $X_1 s + X_2 \stackrel{def}{=} \left(m_1 + m_2, \sigma_1^2 + \sigma_2^2\right)$
- $X_1 s - X_2 \stackrel{def}{=} \left(m_1 - m_2, \sigma_1^2 + \sigma_2^2\right)$
- $X_1 s \times X_2 \stackrel{def}{=} \left(m_1 * m_2, m_2^2\sigma_1^2 + m_1^2\sigma_2^2\right)$
- $X_1 s / X_2 \stackrel{def}{=} \left(m_1/m_2, \left(\frac{\sigma_1}{m_2}\right)^2 + \left(\frac{m_1\sigma_2}{m_2^2}\right)^2\right)$, *with* $m_2 \neq 0$.

They correspond to the first order terms in $\frac{\sigma}{m}$ of operations between two independent Gaussian random variables.

The concept of stochastic zero

Definition

A stochastic number X is a stochastic zero, denoted by $\underline{0}$, if

$$X = (0, 0) \;\; or \;\; C_{\beta,X} \leq 0.$$

In accordance with the concept of stochastic zero, a new equality concept and new order relations have been defined.

Definition

Let $X_1 = (m_1, \sigma_1^2)$ and $X_2 = (m_2, \sigma_2^2)$.

- Stochastic equality, denoted by $s=$, is defined as:
  $X_1 \; s= \; X_2$ if and only if $X_1 \; s- \; X_2 = \underline{0}$.
- Stochastic inequalities, denoted by $s>$ and $s\geq$, are defined as:
  $X_1 \; s> \; X_2$ if and only if $m_1 > m_2$ and $X_1 \; s\neq \; X_2$,
  $X_1 \; s\geq \; X_2$ if and only if $m_1 \geq m_2$ or $X_1 \; s= \; X_2$.

Discrete Stochastic Arithmetic (DSA)

With DSA, a real number becomes an N-dimensional set.

Any operation on these N-dimensional sets is performed element per element using the random rounding mode.

By identifying $C_{\beta,X}$ and $C_{\overline{R}}$, an equality concept and order relations have been defined for DSA.

Control of stochastic operations

Theorem

Let X_i be the approximation in stochastic arithmetic of a mathematical value x_i such that its exact significant bits are those of x_i up to p_i ($i = 1, 2$).

Let $\bigcirc$ be an arithmetical operator: $\bigcirc \in \{+, -, \times, /\}$ and $s\bigcirc$ the corresponding stochastic operator: $s\bigcirc \in \{s+, s-, s\times, s/\}$.

Then the exact significant bits of $X_1 \; s\bigcirc \; X_2$ are those of the mathematical value $x_1 \bigcirc x_2$, up to $max(p_1, p_2)$.

- proved for stochastic operations
- used in practice for results obtained in DSA

F. Jézéquel, *Dynamical control of converging sequences computation*, Applied Numerical Mathematics, 50(2): 147-164, 2004.

DSA on scalar architectures

DSA is implemented in the CADNA library.

A stochastic variable is an N-dimensional set of real numbers. Practically, N = 2 (or 3).

In Fortran 90, each stochastic variable A is represented by a structure consisting of 2 real variables: $A\%x$ and $A\%y$.

Each stochastic operation $A \, \Omega \, B$ is overloaded as:

$$(A\%x, A\%y) \ \Omega \ (B\%x, B\%y) = (A\%x \, \omega \, B\%x \ , \ A\%y \, \omega \, B\%y)$$

ω: arithmetic operation rounded up or down with probability $\frac{1}{2}$.

DSA on parallel architectures

Two types of versions of CADNA on parallel architectures:

- Numerical validation of parallel codes
  using PVM or MPI
- Parallelization of DSA
  to improve the performances of CADNA for sequential codes

(HDR J.-L. Lamotte 2004)

DSA on vector architectures

The problems :

1. vectorial processors do not always respect the IEEE standard
   - NEC SX5 respects the IEEE standard,
   - but not CRAY SV1
2. classical operation overloading inhibits vectorization.

Without the help of a vector preprocessor, we can only implement new array operators.

Run time (in seconds) of a code performing LU decomposition of a matrix of dimension 1000, without pivoting:

	classical code	with CADNA
PC (Pentium III-450)	73	199
CRAY SV1	7	395

Array implementation on Cray SV1

static rounding modes:

- additions and subtractions are rounded to zero,
- multiplications and divisions are rounded to the nearest.

For a stochastic vector operation $C = A \, \Omega \, B$, the elements of C must be randomly rounded up or down.

The last bit in the mantissa is changed using **vector** logical operations and an array of random 0 or 1.

LU decomposition
on CRAY SV1

Without pivoting, the dimension of the matrix is 1000.

Although the CADNA library is inlined, the code has also been written with manual inlining of the array operators.

	Run time	Perf.	Cost
classical vector processor	7.1 s	94 Mflops	–
CADNA, scalar version	395 s	1.7 Mstops	56
CADNA, vector version	49 s	14 Mstops	7
with manual inlining	27 s	24 Mstops	3.8

Mstops means *millions of stochastic operations per second*.

A real-life example: the ORCA code

- numerical simulation of all the ocean streams
- about 50 000 line codes in 112 files

7 subroutines have been rewritten with the array formulation so that the vector version of CADNA can be used.

For 10 iterations, they represent 65 % of the global classical run and 57 % of the stochastic run on CRAY SV1.

For 10 time iterations,

in the 7 routines	Run time	Perf.	Cost
classical	32 s	210 Mflops	–
CADNA, scalar	2868 s	2.3 Mstops	90
CADNA, vector	568 s	11.8 Mstops	18

Array implementation on NEC SX5

NEC SX5 respects the IEEE 754 standard
⇒ the 4 rounding modes defined in this standard are available.

The Fortran compiler has been updated
in order to vectorize some derived type statements.

Memory optimization:

From an 8-byte word to another, a step of length 1 is performed.
An even step-length is very time consuming.

In order to perform steps of odd length, stochastic arrays consist of:

- 2 real arrays in single precision
- 3 real arrays in double precision.

To save a switch of the rounding mode, the following property is used.

$$u *^{+} v = -((-u) *^{-} v) \quad \textit{and} \quad u *^{-} v = -((-u) *^{+} v)$$

Let $C = A * B$, where A, B and C are stochastic arrays of size n.

A random logical array L is used.

$where(L(1:n))$
$C\%x = A\%x * B\%x$
$C\%y = -((-A\%y) * B\%y)$
$elsewhere$
$C\%y = A\%y * B\%y$
$C\%x = -((-A\%x) * B\%x)$
$endwhere$

$$C = \begin{pmatrix} C(1)\%x & C(1)\%y \\ C(2)\%x & C(2)\%y \\ C(3)\%x & C(3)\%y \\ C(4)\%x & C(4)\%y \\ \vdots & \vdots \\ C(n)\%x & C(n)\%y \end{pmatrix}$$

To save a switch of the rounding mode, the following property is used.

$$u *^{+} v = -((-u) *^{-} v) \quad \textit{and} \quad u *^{-} v = -((-u) *^{+} v)$$

Let $C = A * B$, where A, B and C are stochastic arrays of size n.

A random logical array L is used.

$where(L(1:n))$
$C\%x = A\%x * B\%x$
$C\%y = -((-A\%y) * B\%y)$
$elsewhere$
$C\%y = A\%y * B\%y$
$C\%x = -((-A\%x) * B\%x)$
$endwhere$

$$C = \begin{pmatrix} C(1)\%x & C(1)\%y \\ C(2)\%x & C(2)\%y \\ C(3)\%x & C(3)\%y \\ C(4)\%x & C(4)\%y \\ \vdots & \vdots \\ C(n)\%x & C(n)\%y \end{pmatrix}$$

Updates of the Fortran compiler on NEC SX5

Inlining of an operation involving

- 2 arrays: since 2001
- scalar and array: since 2003

With a code performing 1 600 000 multiplications of double precision arrays of size 500:

	Run time	Perf.
Before update of 2001	13.0 s	62 Mstops
After update of 2001	3.5 s	229 Mstops
With global variable (instead of SIZE)	2.0 s	400 Mstops
With global variable and manual inlining	1.7 s	471 Mstops

LU decomposition
on NEC SX5

Without pivoting, the dimension of the matrix is 1000:

	Run time	Perf.	Cost
classical vector processor	0.55 s	1.221 Gflops	–
CADNA, scalar version	1207 s	0.56 Mstops	2194
CADNA, vector version	13.5 s	50 Mstops	25
with manual inlining	1.3 s	501 Mstops	2.4

Still problems of inlining with instructions involving:

- $>$ 2 variables
- 1D sub-arrays of 2D arrays

F. Jézéquel, J.-M. Chesneaux, *For reliable and powerful scientific computations*, Scientific Computing, Validated Numerics, Interval Methods, 367-378, 2001.

Numerical accuracy of approximation methods

When an approximation $L(h)$ such that $\lim\limits_{h \to 0} L(h) = L$ is computed, it is affected by:

- a truncation error $e_m(h)$
- a round-off error $e_c(h)$.

$e_m(h) \longrightarrow$

If h decreases, $L(h)$:

s	exponent	mantissa

$\longleftarrow e_c(h)$

As long as $e_c(h) < e_m(h)$, decreasing h brings reliable information to the mantissa.

The optimal step is reached when $e_c(h) \approx e_m(h)$.

1. How to determine dynamically the optimal step ?
2. Which digits in the approximation obtained are in common with L ?

Significant digits common to two real numbers

Definition

Let a and b be two real numbers, the number of significant digits that are common to a and b can be defined in $\mathbb{R}$ by

1. for $a \neq b$, $C_{a,b} = \log_{10} \left| \dfrac{a+b}{2(a-b)} \right|$,
2. $\forall a \in \mathbb{R},\ C_{a,a} = +\infty$.

Example:

if $a = 2.4599976$ and $b = 2.4600012$, then $C_{a,b} \approx 5.8$.

On sequences with a linear convergence

Theorem

Let (I_n) be a sequence converging linearly to I, i.e. which satisfies $I_n - I = K\alpha^n + o(\alpha^n)$ where $K \in \mathbb{R}$ and $0 < |\alpha| < 1$, then

$$C_{I_n, I_{n+1}} = C_{I_n, I} + \log_{10}\left(\frac{1}{1-\alpha}\right) + o\left(1\right).$$

If the convergence zone is reached,
the significant decimal digits common to I_n and I_{n+1}, are those of I, up to $\log_{10}\left(\frac{1}{1-\alpha}\right)$.

If $-1 < \alpha \leq \frac{1}{2}$, then $-1 < \log_2\left(\frac{1}{1-\alpha}\right) \leq 1$.
In this case, the significant bits common to I_n and I_{n+1} are those of I, up to one.

Applications in DSA

Let us assume that the convergence zone is reached.

If $I_n - I_{n+1} = @.0$,
the difference between I_n and I_{n+1} is due to round-off errors.

Further iterations are useless.

Consequently

- the optimal iterate I_{n+1} can be dynamically determined
- if $\alpha \leq \frac{1}{2}$, the exact significant bits of I_{n+1} are those of I, up to one.

F. Jézéquel, *Dynamical control of converging sequences computation*, Applied Numerical Mathematics, 50(2): 147-164, 2004.

Dynamical control of approximation methods

Theorem

Let $L(h)$ be an approximation of order p of L, i.e.

$$L(h) - L = Kh^p + \mathcal{O}\left(h^q\right) \text{ with } 1 \leq p < q,\ K \in \mathbb{R}.$$

If L_n is the approximation computed with the step $\frac{h_0}{2^n}$, then

$$C_{L_n,L_{n+1}} = C_{L_n,L} + \log_{10}\left(\frac{2^p}{2^p - 1}\right) + \mathcal{O}\left(2^{n(p-q)}\right).$$

If the convergence zone is reached and $L_n - L_{n+1} = @.0$, the exact significant bits of L_{n+1} are those of L, up to one.

Dynamical control of the trapezoidal rule and Simpson's rule

Corollary

If I_n is the approximation of $I = \int_a^b f(x)dx$ computed with step $h = \frac{b-a}{2^n}$ using the trapezoidal rule or Simpson's rule, then

$$C_{I_n,I_{n+1}} = C_{I_n,I} + \log_{10}(\beta) + \mathcal{O}\left(\frac{1}{4^n}\right)$$

- trapezoidal rule: $\beta = \frac{4}{3}$ $(p = 2)$
- Simpson's rule: $\beta = \frac{16}{15}$ $(p = 4)$

If the convergence zone is reached and $I_n - I_{n+1} = @.0$, the exact significant bits of I_{n+1} are those of I, up to one.

J.-M. Chesneaux, F. Jézéquel, *Dynamical control of computations using the trapezoidal and Simpson's rules*, J. of Universal Computer Science, 4(1): 2-10, 1998.

The Gauss-Legendre method

The approximation of $\int_{-1}^{1} f(x)dx$ by the Gauss-Legendre method with ν points is

$$\sum_{i=1}^{\nu} C_i f(x_i)$$

where for $i = 1, ..., \nu$,

- $\{x_i\}$ are the roots of the ν-degree Legendre polynomial P_ν
- $C_i = \frac{2}{(1-x_i^2)(P'_\nu(x_i))^2}$.

For the computation of $I = \int_a^b g(t)dt$, a change of variable is required:

$$I = \frac{(b-a)}{2} \int_{-1}^{1} g\left(\frac{(b-a)x + (b+a)}{2}\right) dx.$$

Dynamical control of the Gauss-Legendre method

Theorem

Let $I = \int_a^b g(t)dt$. If $[a, b]$ is partitioned into 2^n subintervals of same length on which the Gauss-Legendre method with ν points is applied and I_n is the sum of the 2^n approximations obtained, then

$$I_n - I = \frac{K_\nu}{4^{n\nu}} + \mathcal{O}\left(\frac{1}{2^{n(2\nu+1)}}\right).$$

Corollary

$$C_{I_n,I_{n+1}} = C_{I_n,I} + log_{10}\left(\frac{4^\nu}{4^\nu - 1}\right) + \mathcal{O}\left(\frac{1}{2^n}\right).$$

If the convergence zone is reached and $I_n - I_{n+1} = @.0$, the exact significant bits of I_{n+1} are those of I, up to one.

Romberg's method

The approximation of $I = \int_a^b f(x)dx$ with Romberg's method, requires the following computations ($h = \frac{b-a}{M}$, $M \geq 1$):

$$\begin{array}{cccccc}
T_1(h) & T_1(\frac{h}{2}) & .... & T_1(\frac{h}{2^{n-3}}) & T_1(\frac{h}{2^{n-2}}) & T_1(\frac{h}{2^{n-1}}) \\
T_2(h) & T_2(\frac{h}{2}) & .... & T_2(\frac{h}{2^{n-3}}) & T_2(\frac{h}{2^{n-2}}) & \\
T_3(h) & T_3(\frac{h}{2}) & .... & T_3(\frac{h}{2^{n-3}}) & & \\
\vdots & \vdots & & & & \\
T_{n-1}(h) & T_{n-1}(\frac{h}{2}) & & & & \\
T_n(h) & & & & &
\end{array}$$

The first row is computed using the trapezoidal rule with step $\frac{h}{2^j}$.

For $r = 2, ..., n$ and $j = 0, ..., n - r$,

$$T_r\left(\frac{h}{2^j}\right) = \frac{1}{4^{r-1} - 1}\left(4^{r-1}\,T_{r-1}\left(\frac{h}{2^{j+1}}\right) - T_{r-1}\left(\frac{h}{2^j}\right)\right).$$

Dynamical control of Romberg's method

Theorem

If $T_n(h)$ is the approximation of $I = \int_a^b f(x)dx$ computed with n iterations of Romberg's method using the initial step $h = \frac{b-a}{M}$, then

$$C_{T_n(h),T_{n+1}(h)} = C_{T_n(h),I} + \mathcal{O}\left(\frac{1}{n^2}\right).$$

If the convergence zone is reached and $T_n(h) - T_{n+1}(h) = @.0$, the exact significant digits of $T_{n+1}(h)$ are those of I.

Dynamical control of combined sequences

Let (u_m) be a sequence converging linearly to u.

For all m, let $(u_{m,n})$ be a sequence converging linearly to u_m.

$(u_{m,n})$ is computed until, in the convergence zone, the difference between two successive iterates is @.0.

Let U_m be the approximation of u_m obtained.

- $\Rightarrow$ the bits common to u_m and u_{m+1} are those of u, up to p.
- $\Rightarrow$ the exact significant bits of U_m are those of u_m, up to q.
- $\Rightarrow$ if $U_m - U_{m+1} = @.0$,
  the exact significant bits of U_{m+1} are those of u, up to $p + q$.

Dynamical control of integrals on an infinite domain

Let $g = \int_0^\infty \phi(x)dx$ and $g_m = \sum_{j=0}^{m} f_j$ with $f_j = \int_{jL}^{(j+1)L} \phi(x)dx$.

We assume that (g_m) converges linearly to g.

An approximation of each integral can be computed in DSA,
such that its exact significant bits are those of f_j, up to 1.

Let G_m be the approximation of g_m computed in DSA.

$\Rightarrow$ the exact significant bits of G_m are those of g_m, up to 1.

$\Rightarrow$ if the convergence zone is reached,
the significant bits common to g_m and g_{m+1} are those of g, up to p.

$\Rightarrow$ if $G_m - G_{m+1} = @.0$,
the exact significant bits of G_{m+1} are those of g, up to $p+1$.

Dynamical control of multiple integrals computation

PhD M. Charikhi, Jan. 2005

$$I = \int_\Omega f(\mathbf{x})d\mathbf{x} \text{ with } \Omega \subset \mathbb{R}^{\mathrm{N}}$$

can be approximated by:

$$Q[f] = \sum_{j=1}^{\nu} a_j f(\mathbf{x_j}) \text{ with } a_j \in \mathbb{R} \text{ and } \mathbf{x_j} \in \Omega.$$

The approximation Q is called cubature formula if $N \geq 2$.

- polynomial-based methods
- Monte Carlo methods

Cubpack, R. Cools et al. 1992
VANI, C.-Y. Chen 1998
CLAVIS, S. Wedner 2000

Approximation using the principle of “iterated integrals”
Computation of 2-dimensional integrals

$$s = \int_a^b \int_{y_1(x)}^{y_2(x)} f(x,y)dxdy = \int_a^b g(x)dx \text{ with } g(x) = \int_{y_1(x)}^{y_2(x)} f(x,y)dy.$$

$\forall x \in [a,b]$, an approximation $G(x)$ can be computed in DSA such that its exact significant bits are those of $g(x)$, up to δ.

Let $S_n = \phi(\{G(x_i)\})$ be the approximation of s computed in DSA and $s_n = \phi(\{g(x_i)\})$.

- $\Rightarrow$ the exact significant bits of S_n are those of s_n, up to δ
- $\Rightarrow$ if the convergence zone is reached, the significant bits common to s_{n-1} and s_n are common with s, up to δ
- $\Rightarrow$ if $S_{n-1} - S_n = @.0$,
  the exact significant bits of S_n are those of s, up to 2δ.

The exact significant bits of the approximation obtained are those of the mathematical value of the integral, up to $N\delta$.

- With Romberg's method, $\delta = 0$.
- With the trapezoidal rule, $N\delta$ represents:
  - one bit if $N \leq 2$
  - one decimal digit if $N \leq 8$.
- With Simpson's rule, $N\delta$ represents one bit if $N \leq 35$.
- With the Gauss-Legendre method with 6 points, $N\delta$ represents one bit if $N \leq 2838$.

Computation of an integral involved in crystallography

$$g(a) = \int_0^{+\infty} f(x)dx,$$

with $f(x) = [\exp(x) + \exp(-x)]^a - \exp(ax) - \exp(-ax)$ and $0 < a < 2$.

$g(5/3) \approx 4.45$ (W. Harrison 1981)
$g(5/3) \approx 4.6262911$ (SIAM review 1996)

$g(a)$ can be expressed as a series expansion:

$$g(a) = \sum_{n=1}^{+\infty} \frac{\Pi_{i=0}^{n-1}(a-i)}{(n!)(2n-a)} - \frac{1}{a}.$$

F. Jézéquel, J.-M. Chesneaux, *Computation of an infinite integral using Romberg's method*, Numerical Algorithms, 36(3): 265-283, 2004.

Several numerical problems may occur in the computation of $g(a)$:

- for high values of x, the computation of $f(x)$ may generate cancellations,
- the upper bound of the integral is infinite,
- the quadrature method used, e.g. Romberg's method, generates both a truncation error and a round-off error.

Computation of an integral involved in crystallography

Dynamical control of the computation

In order to avoid cancellations, the same expression of the integrand is not used at both bounds of the interval.

$$g(a) \approx \int_0^l f_1(x)dx + \sum_{j=1}^{k} \int_{jl}^{(j+1)l} f_2(x)dx,$$

where

$$f_1(x) = \exp(ax)\left[(1+\exp(-2x))^a - 1 - \exp(-2ax)\right]$$

$$f_2(x) = \exp(ax)u(x) - \exp(-ax),$$

$$u(x) = \lim_{n\to\infty} u_n(x) \text{ with } u_n(x) = \sum_{i=1}^{n-1} \frac{\exp(-2ix)}{i!}\Pi_{j=0}^{i-1}(a-j).$$

Dynamical choice of several parameters:

- n such that $u_n(x) \approx u(x)$
- k such that $\int_l^{kl} f_2(x)dx \approx \int_l^{\infty} f_2(x)dx$
- the number of iterations with Romberg's method

Proposition

One can compute an approximation $G(a)$ such that its exact significant digits are those of $g(a)$, up to $\delta = \log_{10}\left(\frac{2}{1-\exp^{-l\min(a,2-a)}}\right)$.

a	$\delta \approx$		$g(a)$
0.5	0.34	exact:	-1.694426169587958E+000
		DSA:	-1.69442616958795E+000
$5/3$	0.39	exact:	4.62629111983995E+000
		DSA:	4.626291119833E+000
1.9999	3.6	exact:	1.999899986776092E+004
		DSA:	1.99989997358E+004

The exact significant digits of $G(a)$ are in common with $g(a)$, up to $\lceil \delta \rceil$.

Study of an integral involved in the neutron star theory

$$\tau(\varepsilon, v) = \frac{1}{\omega(\varepsilon)} \int_0^{\frac{\pi}{2}} d\theta \sin(\theta) \int_0^{\infty} dn\, n^2 \int_0^{\infty} dp\, h(n, p, \theta, \varepsilon, v)$$

$(\varepsilon, v) \in [10^{-4}, 10^4] \times [10^{-4}, 10^3]$

ω is a normalization function

$$\begin{aligned} h(n, p, \theta, \varepsilon, v) = \;& \psi(z)\Gamma(n - \varepsilon - z) + \psi(-z)\Gamma(n - \varepsilon + z) \\ & -\psi(z)\Gamma(n + \varepsilon - z) - \psi(z)\Gamma(n + \varepsilon + z) \end{aligned}$$

with $z = \sqrt{p^2 + (v \sin(\theta))^2}$, $\psi(x) = \frac{1}{\exp(x)+1}$, $\Gamma(x) = \frac{x}{\exp(x)-1}$.

F. Jézéquel, F. Rico, J.-M. Chesneaux, M. Charikhi, *Reliable computation of a multiple integral involved in the neutron star theory*, submitted to “Mathematics and Computers in Simulation”.

The numerical problems:

- two infinite bounds
  $\int_0^\infty \ldots$ is replaced by $\sum_{j=0}^{k} \int_{jL}^{(j+1)L} \ldots$
  $\Rightarrow$ Dynamical choice of k
- $\Gamma(x) = \frac{x}{\exp(x)-1}$ generates cancellations if $x \approx 0$.
  a series expansion of $\Gamma(x)$ is used: $\Gamma(x) \approx \frac{1}{1+\frac{x}{2}+\ldots+\frac{x^{n-1}}{n!}}$
  $\Rightarrow$ Dynamical choice of n
- With the principle of "iterated integrals", the Gauss-Legendre method is used and generates both a truncation error and a round-off error
  $\Rightarrow$ Dynamical control of the Gauss-Legendre method

The numerical problems:

- two infinite bounds
  $\int_0^{\infty}$ ... is replaced by $\sum_{j=0}^{k} \int_{jL}^{(j+1)L}$ ...

  $\Rightarrow$ Dynamical choice of k

- $\Gamma(x) = \frac{x}{\exp(x)-1}$ generates cancellations if $x \approx 0$.
  a series expansion of $\Gamma(x)$ is used: $\Gamma(x) \approx \frac{1}{1+\frac{x}{2}+\ldots+\frac{x^{n-1}}{n!}}$

  $\Rightarrow$ Dynamical choice of n

- With the principle of "iterated integrals", the Gauss-Legendre method is used and generates both a truncation error and a round-off error

  $\Rightarrow$ Dynamical control of the Gauss-Legendre method

$\tau(\varepsilon, v)$ has been computed using DSA in single precision for 5752 points (ε, v) defined by:

$$\begin{cases} \varepsilon = 10^a & \textit{with} \quad a = -4.0, -3.9, -3.8, \ldots, 4.0 \\ v = 10^b & \textit{with} \quad b = -4.0, -3.9, -3.8, \ldots, 3.0. \end{cases}$$

The run time of the code varies from 45 s to 3347 s depending on the values of ε and v, the average run time being 389 s.

Proposition

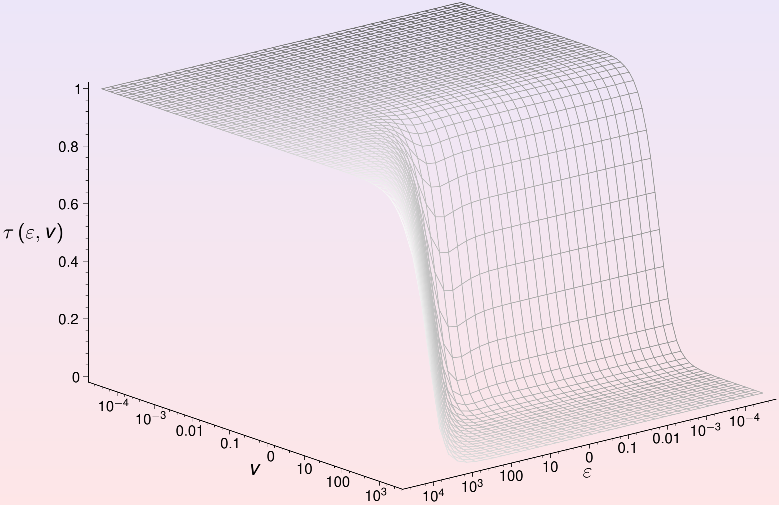
One can compute an approximation of $\tau(\varepsilon, v)$ such that its exact significant digits are those of $\tau(\varepsilon, v)$, up to 2.

nb. of exact significant digits	occurrence
3	1
4	217
5	665
6	3347
7	1522

$\Rightarrow$ we can guarantee 1 to 5 significant digits in the results obtained.

Study of an integral involved in the neutron star theory

Numerical results



Conclusion and perspectives - 1/2

Dynamical control of converging sequences computation

Let $u = \lim_{n \to \infty} u_n$. From two iterates in the convergence zone, one can determine the first digits of u.

If $u_n - u_{n+1} = @.0$, one can determine which significant digits of u_{n+1} are in common with u.

Combination of theoretical results if several sequences are involved

For the approximation of an integral, one has to take into account:

- the dimension of the integral
- the number of improper bounds
- the possible approximation of the integrand by its series expansion
- the convergence speed of the sequences involved

Conclusion and perspectives - 2/2

- Adaptive strategies
- Other approximation methods
- Approximation of multiple integrals
  - other cubature methods
  - singular integrals
  - Monte Carlo methods
- Dynamical control of vector sequences computation
  PhD R. Adout
  - acceleration of the restarted GMRES method
  - dynamical control of the dimension of the Krylov subspace
- Automatic methods for round-off error analysis
  - DSA for MATLAB
  - compiler with DSA features
  - linear algebra library
  - grid computing: new methodologies