Implementing Taylor models arithmetic using floating-point arithmetic: bounding roundoff errors

Nathalie Revol
INRIA, Arenaire, LIP, ENS-Lyon, France
Nathalie.Revol@ens-lyon.fr

TMW'06, Boca Raton, Florida, 16-19 December 2006

Outline

- **introduction to Taylor models arithmetic**
- implementation using floating-point arithmetic
- details of various operations
  - addition of two Taylor models
  - multiplication of a Taylor model by a scalar
  - multiplication of two Taylor models
  - better multiplication of two Taylor models
- conclusion

Introduction to Taylor models arithmetic

A function f can be represented by a Taylor model (p, I)
where p is a polynomial and I is an interval if

$$\forall x \in D_f,\ f(x) \in p(x) + I.$$

(p, I) is a **Taylor model** for f.

Typically, p is the Taylor expansion of f and I encloses the truncation error of D_f, hence the name of **Taylor** model.
Assumption : interval $[-1, 1]$ as domain.

Operations on Taylor models : addition

Addition of two Taylor models :

$$(p, I) + (q, J) = (p + q, I + J).$$

If (p, I) is a Taylor model for f
and (q, J) is a Taylor model for g,
then $(p + q, I + J)$ is a Taylor model for $f + g$.

Example : $(1 + x, I) + (2 - 3x, J) = (3 - 2x, I + J)$.

Operations on Taylor models : multiplication by a scalar

Multiplication of a Taylor model by a scalar :

$$c \times (p, I) = (c \times p, c \times I).$$

If (p, I) is a Taylor model for f,
then $(c \times p, c \times I)$ is a Taylor model for $c \times f$.

Example : $5 \times (2 - 3x, I) = (10 - 15x, 5I)$.

Operations on Taylor models : multiplication

Multiplication of two Taylor models :

$$\begin{aligned}(p, I) \times (q, J) &= (trunc_n(p \times q), \\ &\quad \text{truncation error } + Rg(p) \times J + I \times Rg(q) + I \times J).\end{aligned}$$

If (p, I) is a Taylor model for f
and (q, J) is a Taylor model for g,
then $(p, I) \times (q, J)$ is a Taylor model for $f \times g$.

Example :

reminder : $x \in [-1, 1]$.

$$
\begin{aligned}
&(1 + x, [2, 3]) \times (2 - x, [-1, 0]) \\
&\quad = \quad (2 + x, Rg(-x^2) + Rg(1 + x) \cdot [-1, 0] + Rg(2 - x) \cdot [2, 3] \\
&\qquad\qquad + [2, 3] \cdot [-1, 0]) \\
&\quad = \quad (2 + x, [-1, 0] + [0, 2] \cdot [-1, 0] + [1, 3] \cdot [2, 3] + [2, 3] \cdot [-1, 0]) \\
&\quad = \quad (2 + x, [-4, 9])
\end{aligned}
$$

Outline

- introduction to Taylor models arithmetic
- **implementation using floating-point arithmetic**
- details of various operations
  - addition of two Taylor models
  - multiplication of a Taylor model by a scalar
  - multiplication of two Taylor models
  - better multiplication of two Taylor models
- conclusion

Implementation of Taylor models arithmetic

Cf. COSY.

Implementation of Taylor models using floating-point arithmetic :

- coefficients of the polynomial and endpoints of the interval
  $=$ floating-point numbers
- operations on Taylor models performed using floating-point arithmetic.

Advantage : benefit from the speed of floating-point arithmetic (implemented in hardware, thus very fast).

Implementation of Taylor models arithmetic

Roundoff errors must be taken into account.

Idea : for each computed coefficient,
bound the error on the computed coefficient by E
and add $[-E, E]$ to the interval remainder I.

I thus becomes a big "bin", enclosing every possible source of error (truncation error, roundoff error. . .).

Implementation of Taylor models arithmetic

Roundoff errors must be taken into account.

Example : addition of $(\sum_{i=0}^n a_i x^i, I)$ **and** $(\sum_{j=0}^n b_j x^j, J)$**.**
Using exact arithmetic :

$$(\sum_{i=0}^{n} a_i x^i, I) + (\sum_{j=0}^{n} b_j x^j, J) = (c, K),$$

where

$$c = \sum_{k=0}^{n} c_k x^k \text{ with } c_k = a_k + b_k \text{ and } K = I + J.$$

Implementation of Taylor models arithmetic

Roundoff errors must be taken into account.

Example : addition of $(\sum_{i=0}^{n} a_i x^i, I)$ **and** $(\sum_{j=0}^{n} b_j x^j, J)$**.**

Using exact arithmetic :

$(\sum_{i=0}^{n} a_i x^i, I) + (\sum_{j=0}^{n} b_j x^j, J) = (c, K)$

where

$c = \sum_{k=0}^{n} c_k x^k$

with $c_k = a_k + b_k$

Using floating-point arithmetic :

$(\sum_{i=0}^{n} a_i x^i, I) \oplus (\sum_{j=0}^{n} b_j x^j, J) = (\hat{c}, \hat{K})$

where

$\hat{c} = \sum_{k=0}^{n} \hat{c_k} x^k$

with $\hat{c_k} = a_k \oplus b_k$

Implementation of Taylor models arithmetic

Elementary roundoff errors :

$$e_k = c_k - \hat{c_k}.$$

Let $E \geq \sum_{k=0}^{n} |e_k|$,
then when x varies in $[-1, 1]$,
the difference between $c(x)$ and $\hat{c}(x)$ lies in $[-E, E]$.

Roundoff errors are properly accounted for if

$$\hat{K} = K + [-E, E] = I + J + [-E, E].$$

Outline

- introduction to Taylor models arithmetic
- implementation using floating-point arithmetic
- **details of various operations**
  - **addition of two Taylor models**
  - multiplication of a Taylor model by a scalar
  - multiplication of two Taylor models
  - better multiplication of two Taylor models
- conclusion

Addition of two Taylor models using FP arithmetic

Addition of $(\sum_{i=0}^{n} a_i x^i, I)$ **and** $(\sum_{j=0}^{n} b_j x^j, J)$ using FP arithmetic :

$(\sum_{i=0}^{n} a_i x^i, I) \oplus (\sum_{j=0}^{n} b_j x^j, J) = (\hat{c}, \hat{K})$

where

$\hat{c} = \sum_{k=0}^{n} \hat{c}_k x^k$

with $\hat{c}_k = a_k \oplus b_k$

$$e_k = (a_k + b_k) - (a_k \oplus b_k)$$
$$E = (1 \oplus n\varepsilon) \odot \bigoplus_{k=0}^{n} |e_k|$$
$$\hat{K} = I + J + [-E, E]$$

Addition of two Taylor models using FP arithmetic

$e_k = (a_k + b_k) - (a_k \oplus b_k)$

for $k = 0$ to n, e_k is computed using the TwoSum algorithm

more precisely, $(\hat{c_k}, e_k) =$ TwoSum (a_k, b_k)

$E = (1 \oplus n\varepsilon) \odot \bigoplus_{k=0}^{n} |e_k|$

where ε is 1 ulp, $(1 + n\varepsilon)$ is computed exactly with FP arithmetic

and the factor $(1 + n\varepsilon)$ accounts for roundoff when computing E

$\hat{K} = I + J + [-E, E]$

$\hat{K}$ is computed using interval arithmetic.

Outline

- introduction to Taylor models arithmetic
- implementation using floating-point arithmetic
- **details of various operations**
  - addition of two Taylor models
  - **multiplication of a Taylor model by a scalar**
  - multiplication of two Taylor models
  - better multiplication of two Taylor models
- conclusion

Multiplication of a Taylor model by a scalar

Multiplication of $(\sum_{i=0}^{n} a_i x^i, I)$ by a scalar b using FP arithmetic :
$b \cdot (\sum_{i=0}^{n} a_i x^i, I) = (\hat{c}, \hat{K})$

where
$\hat{c} = \sum_{k=0}^{n} \hat{c_k} x^k$
with $\hat{c_k} = a_k \odot b$

$$e_k = (a_k \cdot b) - (a_k \odot b)$$
$$E = (1 \oplus n\varepsilon) \odot \bigoplus_{k=0}^{n} |e_k|$$
$$\hat{K} = I + J + [-E, E]$$

Multiplication of a Taylor model by a scalar

$e_k = (a_k \cdot b) - (a_k \odot b)$

for $k = 0$ to n, e_k is computed using the TwoMult algorithm

more precisely, $(\hat{c}_k, e_k) =$ TwoMult (a_k, b)

$E = (1 \oplus n\varepsilon) \odot \bigoplus_{k=0}^{n} |e_k|$

where again ε is 1 ulp, $(1 + n\varepsilon)$ is computed exactly with FP arithmetic

and the factor $(1 + n\varepsilon)$ accounts for roundoff when computing E

$\hat{K} = I + J + [-E, E]$

$\hat{K}$ is computed using interval arithmetic.

Outline

- introduction to Taylor models arithmetic
- implementation using floating-point arithmetic
- **details of various operations**
  - addition of two Taylor models
  - multiplication of a Taylor model by a scalar
  - **multiplication of two Taylor models**
  - better multiplication of two Taylor models
- conclusion

Multiplication of two Taylor models using FP arith.

Multiplication of $(\sum_{i=0}^{n} a_i x^i, I)$ **by** $(\sum_{j=0}^{n} b_j x^j, J)$ using FP arith. :
$(\sum_{i=0}^{n} a_i x^i, I) \cdot (\sum_{j=0}^{n} b_j x^j, J) = (\hat{c}, \hat{K})$

where
$\hat{c} = \sum_{k=0}^{n} \hat{c_k} x^k$
with $\hat{c_k} = \bigoplus_{i+j=k} a_i \odot b_j$

$e_k = c_k - \hat{c_k}$
$E = (1 \oplus n\varepsilon) \odot \bigoplus_{k=0}^{n} |e_k|$
$\hat{K} = I + J + [-E, E]$

Multiplication of two Taylor models using FP arith.

$e_k = \sum_{i=0}^{k} a_i \cdot b_{k-1} - \bigoplus_{i=0}^{k} a_i \odot b_{k-i}$

for each operation ($\oplus$ or $\odot$),

the roundoff error is computed using either a TwoSum or a TwoMult

finally, e_k is computed by summing (using $\oplus$) all these terms

and by multiplying by a security factor (of the kind $(1 + 2k\varepsilon)$).

$E = (1 \oplus n\varepsilon) \odot \bigoplus_{k=0}^{n} |e_k|$

where the factor $(1 + n\varepsilon)$ accounts for roundoff when computing E

$\hat{K} = I + J + [-E, E]$

$\hat{K}$ is computed using interval arithmetic.

Outline

- introduction to Taylor models arithmetic
- implementation using floating-point arithmetic
- **details of various operations**
  - addition of two Taylor models
  - multiplication of a Taylor model by a scalar
  - multiplication of two Taylor models
  - **better multiplication of two Taylor models**
- conclusion

Multiplication of two Taylor models using FP arith.

Multiplication of $(\sum_{i=0}^{n} a_i x^i, I)$ **by** $(\sum_{j=0}^{n} b_j x^j, J)$ using FP arith. :
$(\sum_{i=0}^{n} a_i x^i, I) \cdot (\sum_{j=0}^{n} b_j x^j, J) = (\hat{c}, \hat{K})$

where
$\hat{c} = \sum_{k=0}^{n} \hat{c_k} x^k$
with $\hat{c_k} = \bigoplus_{i+j=k} a_i \odot b_j$
or equivalently $c_k = \{(a_i)^t \odot (b_{k-i})\}$ is a FP dot product

$e_k = c_k - \hat{c_k}$
$E = (1 \oplus n\varepsilon) \odot \bigoplus_{k=0}^{n} |e_k|$
$\hat{K} = I + J + [-E, E]$

Accurate dot product by Ogita, Rump and Oishi (2004)

$$
\begin{aligned}
&\texttt{function [res, err] = DotErr1}(x, y)\\
&\quad [p, s] = \texttt{TwoMult}(x_1, y_1)\\
&\quad \texttt{err} = |\texttt{s}|\\
&\quad \texttt{for } i = 2 : n\\
&\qquad [\texttt{h,r}] = \texttt{TwoMult}(x_i, y_i)\\
&\qquad [\texttt{p,q}] = \texttt{TwoSum(p,h)}\\
&\qquad \texttt{s} = \texttt{s} \oplus (\texttt{q} \oplus \texttt{r})\\
&\qquad \texttt{err} = \texttt{err} \oplus (|\texttt{q}| \oplus |\texttt{r}|)\\
&\quad \texttt{res} = \texttt{p} \oplus \texttt{s}\\
&\quad \texttt{err} = \texttt{err} \oslash (1 - (n+2)\varepsilon)
\end{aligned}
$$

Multiplication of $(\sum_{i=0}^{n} a_i x^i, I)$ **by** $(\sum_{j=0}^{n} b_j x^j, J)$

$(\sum_{i=0}^{n} a_i x^i, I) \cdot (\sum_{j=0}^{n} b_j x^j, J) = (\hat{c}, \hat{K})$

where

$\hat{c} = \sum_{k=0}^{n} \hat{c_k} x^k$

with $(\hat{c_k}, e_k) =$ DotErr1 $((a_i), (b_{k-i}))$

$E = (1 \oplus n\varepsilon) \odot \bigoplus_{k=0}^{n} |e_k|$

where the factor $(1 + n\varepsilon)$ accounts for roundoff when computing E

$\hat{K} = I + J + [-E, E]$

$\hat{K}$ is computed using interval arithmetic.

Outline

- introduction to Taylor models arithmetic
- implementation using floating-point arithmetic
- details of various operations
  - addition of two Taylor models
  - multiplication of a Taylor model by a scalar
  - multiplication of two Taylor models
  - better multiplication of two Taylor models
- **conclusion**

Conclusion

- **quality :**
  - better, tighter bounds for roundoff errors
  - thus interval remainder should contain only "true" error
- **price :**
  - a few extra operations, especially in the presence of a FMA
  - but maybe not much more than in existing COSY
  - maybe even better in practice, since no test and branching

Possible improvements

- **assumption :**
  - algorithms work only with rounding to nearest
  - cf. Christoph Lauter's talk
    algorithms exist that work for any faithful rounding mode
- **even higher precision (double-double, triple-double) :**
  - use of (truncated) expansions
  - care must be taken to bound tightly the roundoff errors
- **arbitrary precision :**
  - more expensive
  - resort to more naive error bounds for efficiency reason

Disclaimer

I did not prove totally yet the algorithms given here. What might be slightly modified are the safety factors of the kind $1 + n\varepsilon$, which may be something like $1 + (n + 2)\varepsilon \ldots$

Bibliography

- Dekker T. J., *A floating point technique for extending the available precision*, *Numerische Mathematik*, vol 18, no 3, pp 224-242, 1971.
- Higham N., *Accuracy and stability of numerical algorithms (2nd edition)*, SIAM, 2002. Chapter 4 : Summation.
- Lauter C., *Basic building blocks for a triple-double intermediate format*, Research report LIP, ENS Lyon RR2005-38, 2005. http://www.ens-lyon.fr/LIP/Pub/Rapports/RR/RR2005/RR2005-38.pdf
- Ogita T., Rump S. and Oishi S., *Accurate sum and dot product*, *SIAM Journal on Scientific Computing*, 2004.
- Priest D., Algorithms for arbitrary precision floating point arithmetic, Kornerup P. and Matula D., *10th Symposium on Computer Arithmetic*, Grenoble, France, pp 132–144, 1991. http://www.cs.cmu.edu/afs/cs/project/quake/public/papers/related/Priest.ps
- Shewchuk J. R., Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates, *Discrete and Comput. Geometry*, vol 18, pp 305-363, 1997.
- Sterbenz P. H., *Floating Point Computation*, Prentice Hall, 1974.