Tight Enclosure of Matrix Multiplication with Level 3 BLAS

K. Ozaki (Shibaura Institute of Technology)
joint work with
T. Ogita (Tokyo Woman's Christian University)

8th Small Workshop on Interval Methods
the Charles university, Prague, Czech Republic,
June 10th, 2015

Introduction

We focus on tight enclosure of matrix multiplication.

For matrices A and B we want to obtain $[\underline{C}, \overline{C}]$ which encloses AB by using only numerical computations (floating-point arithmetic), namely,

$$AB \in [\underline{C}, \overline{C}]$$

Introduction

- $\mathbb{F}$: set of floating-point numbers as defined by IEEE 754
- $\mathbb{IF}$: set of intervals (with floating-point numbers)
- $\mathbf{u}$: the relative rounding error unit (2^{-53} for binary64)
- $\mathrm{fl}(\cdots)$: a computed result by floating-point arithmetic.

Introduction

For $A \in \mathbb{F}^{m\times n}, B \in \mathbb{F}^{n\times p}$, the concern is to obtain $[C] \in \mathbb{IF}^{m\times p}$ s.t.

$$AB \in [C]$$

$\mathrm{fl}_{\triangle}(\cdots)$ and $\mathrm{fl}_{\triangledown}(\cdots)$ means a computed result with rounding upward and rounding downward, respectively. Then, we obtain $[C]$ by

$$[C] := [\mathrm{fl}_{\triangledown}(AB), \mathrm{fl}_{\triangle}(AB)]$$

Introduction

The code is very simple (by using INTLAB).

```
function C = MAT(A,B)
  setround(-1)
  Cd=A*B;
  setround(1)
  Cu=A*B;
  C=infsup(Cd,Cu);
end
```

Tight Enclosure of Matrix Multiplication

K. Ozaki, T. Ogita, S. Oishi: Tight and efficient enclosure of matrix multiplication by using optimized BLAS, Numerical Linear Algebra With Applications, Vol. 18:2 (2011), pp. 237-248.

The following is overview in the paper.

$$A = A^{(1)} + A^{(2)}, \quad B = B^{(1)} + B^{(2)}, \quad A^{(1)}B^{(1)} = \mathrm{fl}(A^{(1)}B^{(1)}).$$

Then,

$$AB = (A^{(1)} + A^{(2)})(B^{(1)} + B^{(2)}) = A^{(1)}B^{(1)} + A^{(1)}B^{(2)} + A^{(2)}B.$$

Accurate Matrix Multiplication

Let a constant β be

$$\beta = \lceil (\log_2 n - \log_2 \mathbf{u})/2 \rceil = \lceil (\log_2 n + 53)/2 \rceil .$$

Two vectors $\sigma \in \mathbb{F}^m$ and $\tau \in \mathbb{F}^p$ are defined as

$$\sigma_i = 2^{\beta} \cdot 2^{v_i}, \quad \tau_j = 2^{\beta} \cdot 2^{w_j}, \tag{1}$$

where two vectors $v \in \mathbb{F}^m$ and $w \in \mathbb{F}^p$ are defined as

$$v_i = \lceil \log_2 \max_{1 \le j \le n} |a_{ij}| \rceil, \quad w_j = \lceil \log_2 \max_{1 \le i \le n} |b_{ij}| \rceil. \tag{2}$$

Accurate Matrix Multiplication

We obtain $A^{(1)}$ and $A^{(2)}$ by

$$a_{ij}^{(1)} = \mathrm{fl}\left((a_{ij} + \sigma_i) - \sigma_i\right), \quad a_{ij}^{(2)} = \mathrm{fl}\left(a_{ij} - a_{ij}^{(1)}\right). \qquad (3)$$

Similarly, B is divided into $B^{(1)}$ and $B^{(2)}$ by

$$b_{ij}^{(1)} = \mathrm{fl}\left((b_{ij} + \tau_j) - \tau_j\right), \quad b_{ij}^{(2)} = \mathrm{fl}\left(b_{ij} - b_{ij}^{(1)}\right). \qquad (4)$$

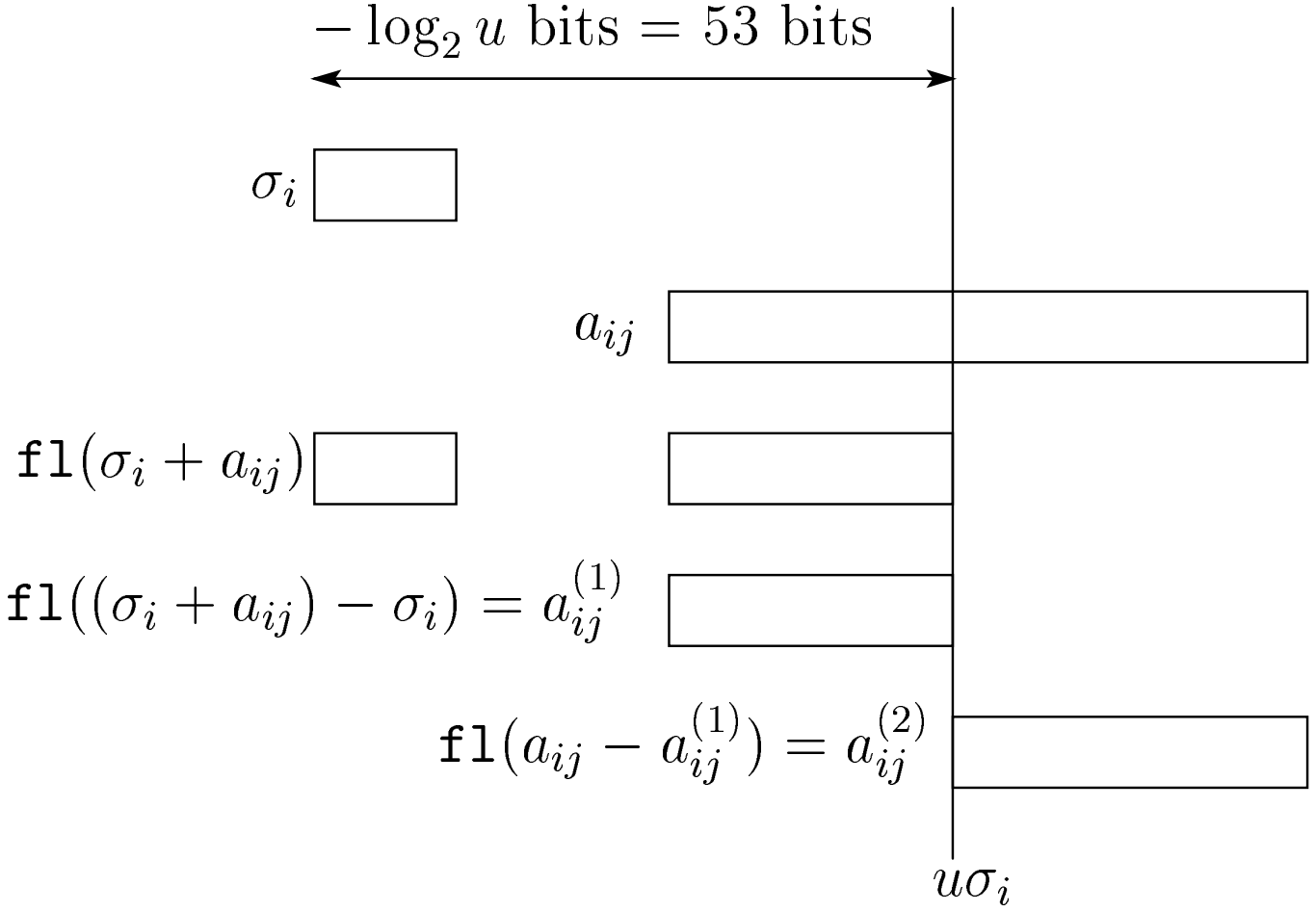

Figure 1: Image of Error-Free Splitting

Error-Free Splitting

$$|a_{ij}^{(1)}| \le 2^{-\beta}\sigma_i, \quad a_{ij}^{(1)} \in \mathbf{u}\sigma_i\mathbb{Z}, \quad |a_{ij}^{(2)}| \le \mathbf{u}\sigma_i$$
$$|b_{ij}^{(1)}| \le 2^{-\beta}\tau_j, \quad b_{ij}^{(1)} \in \mathbf{u}\tau_j\mathbb{Z}, \quad |b_{ij}^{(2)}| \le \mathbf{u}\tau_j$$

They yield

$$\mathbf{u}^2\sigma_i\tau_j\mathbb{Z} \ni \sum_{k=1}^{n} a_{ik}^{(1)}b_{kl}^{(1)} \le \sum_{k=1}^{n} |a_{ik}^{(1)}||b_{kl}^{(1)}| \le n2^{-2\beta}\sigma_i\tau_j \le \mathbf{u}\sigma_i\tau_j.$$

Therefore, there is no rounding error in the dot product!

Accurate Matrix Multiplication

The matrix multiplication AB is transformed to

$$\begin{aligned} AB &= A^{(1)}B^{(1)} + A^{(1)}B^{(2)} + A^{(2)}B \\ &= \mathrm{fl}(A^{(1)}B^{(1)}) + A^{(1)}B^{(2)} + A^{(2)}B. \end{aligned}$$

Then, the interval matrix is obtained by

$$\begin{aligned} AB \in\ & \mathrm{fl}(A^{(1)}B^{(1)}) + [\mathrm{fl}_{\nabla}(A^{(1)}B^{(2)}),\ \mathrm{fl}_{\triangle}(A^{(1)}B^{(2)})] \\ & + [\mathrm{fl}_{\nabla}(A^{(2)}B),\ \mathrm{fl}_{\triangle}(A^{(2)}B)]. \end{aligned}$$

MATLAB Program (12 lines)

```
function C1 = MAT2(A,B)
  n=size(A,2);
  mu=max(abs(A),[],2);
  temp=2.^(ceil(log2(mu)+ceil(53+log2(n))/2));
  sigma=repmat(temp,1,n);
  A1=(A+sigma)-sigma;
  A2=A-A1;
  mu=max(abs(B));
  temp=2.^(ceil(log2(mu))+ceil((53+log2(n))/2));
  tau=repmat(temp,n,1);
  B1=(B+tau)-tau;
  B2=B-B1;
  C1=A1*B1+(intval(A1)*B2+intval(A2)*B);
end
```

Applications

Andreas Frommer, Behnam Hashemi, Verified error bounds for solutions of Sylvester matrix equations, Linear Algebra and its Applications, Volume 436, Issue 2, 15 January 2012, Pages 405–420.

Shinya Miyajima, Fast enclosure for solutions of Sylvester equations, Linear Algebra and its Applications, Volume 439, Issue 4, 15 August 2013, Pages 856–878.

Extension

Both A and B are divided into an unevaluated sum of three floating-point matrices, i.e.

$$A = A^{(1)} + A^{(2)} + A^{(3)}, \quad B = B^{(1)} + B^{(2)} + B^{(3)}$$

Then, AB is transformed to

$$\begin{aligned} AB = {} & A^{(1)}B^{(1)} + A^{(2)}B^{(1)} + A^{(1)}B^{(2)} \\ & + A^{(1)}B^{(3)} + A^{(2)}(B^{(2)} + B^{(3)}) + A^{(3)}B. \end{aligned}$$

Error-Free Splitting

The reason for the error-free is that

$$\mathbf{u}^2\sigma_i\tau_j\mathbb{Z} \ni \sum_{k=1}^{n} a_{ik}^{(1)}b_{kl}^{(1)} \leq \sum_{k=1}^{n} |a_{ik}^{(1)}||b_{kl}^{(1)}| \leq n2^{-2\beta}\sigma_i\tau_j \leq \mathbf{u}\sigma_i\tau_j.$$

However, it is very rare to find that the upper bound of the dot product becomes $\mathbf{u}\sigma_i\tau_j$.

We want to make β more small but ...

New Error-Free Transformation

Let $1 \leq \beta' \leq \beta$.

Replacing β with β' in (1) we define two vectors

$$\tilde{\sigma}_i = 2^{\beta'} \cdot 2^{v_i^{(k)}}, \quad \tilde{\tau}_j = 2^{\beta'} \cdot 2^{w_j^{(l)}},$$

where $v_i^{(k)}$ and $w_j^{(l)}$ are defined by

$$v_i^{(k)} = \lceil \log_2 \max_{1 \leq j \leq n} |a_{ij}^{(k)}| \rceil, \quad w_j^{(l)} = \lceil \log_2 \max_{1 \leq i \leq n} |b_{ij}^{(l)}| \rceil.$$

We divide A and B as follows:

$$\tilde{a}_{ij}^{(1)} = \mathrm{fl}((a_{ij} + \tilde{\sigma}_i) - \tilde{\sigma}_i) \quad , \quad \tilde{b}_{ij}^{(1)} = \mathrm{fl}((b_{ij} + \tilde{\tau}_j) - \tilde{\tau}_j),$$
$$\tilde{a}_{ij}^{(2)} = \mathrm{fl}(a_{ij} - \tilde{a}_{ij}^{(1)}) \quad , \quad \tilde{b}_{ij}^{(2)} = \mathrm{fl}(b_{ij} - \tilde{b}_{ij}^{(1)}).$$

Then,

$$A = \tilde{A}^{(1)} + \tilde{A}^{(2)}, \quad B = \tilde{B}^{(1)} + \tilde{B}^{(2)},$$

but

$$\mathrm{fl}(\tilde{A}^{(1)}\tilde{B}^{(1)}) = \tilde{A}^{(1)}\tilde{B}^{(1)}?$$

New Error-Free Transformation

First, we define a constant $c \in \mathbb{R}$ by

$$c = 2^r,\ r \in \mathbb{N}, \quad \mathrm{fl}\left(\frac{c}{2}\right) \neq \mathtt{Inf}, \quad \mathrm{fl}(c) = \mathtt{Inf}$$

For example, c is 2^{1024} for binary64. Next, we set two constants d_1 and d_2 with powers of two:

$$d_1 d_2 = c\mathbf{u}, \quad d_1 = 2^{486} \text{ and } d_2 = 2^{485} \text{ for binary64}$$

New Error-Free Transformation

Let two vectors $x \in \mathbb{F}^m$ and $y \in \mathbb{F}^p$ be

$$x_i = \frac{d_1}{\mathbf{u}\tilde{\sigma}_i}, \quad y_i = \frac{d_2}{\mathbf{u}\tilde{\tau}_i}.$$

Then, diagonal scalings for $\tilde{A}^{(1)}$ and $\tilde{B}^{(1)}$ are applied as follows:

$$\dot{A} = \mathtt{diag}(x)\tilde{A}^{(1)}, \quad \dot{B} = \tilde{B}^{(1)}\mathtt{diag}(y).$$

New Error-Free Transformation

After the diagonal scaling, we have

$$\dot{a}_{ij} \in d_1\mathbb{Z}, \quad \dot{b}_{ij} \in d_2\mathbb{Z}.$$

Therefore,

$$\dot{a}_{ij}\dot{b}_{ij} \in d_1 d_2 \mathbb{Z} = \mathbf{u}c\mathbb{Z}$$

For binary64,

$$\dot{a}_{ij}\dot{b}_{ij} \in 2^{971}\mathbb{Z}$$

New Error-Free Transformation

Theorem 1 *Let* $T := \mathrm{fl}(\dot{A}\dot{B})$. *If*

$$t_{ij} \notin \{\mathtt{Inf}, -\mathtt{Inf}, \mathtt{NaN}\}$$

is satisfied for all pairs of (i, j), *then* $\dot{A}\dot{B} = \mathrm{fl}(\dot{A}\dot{B})$.

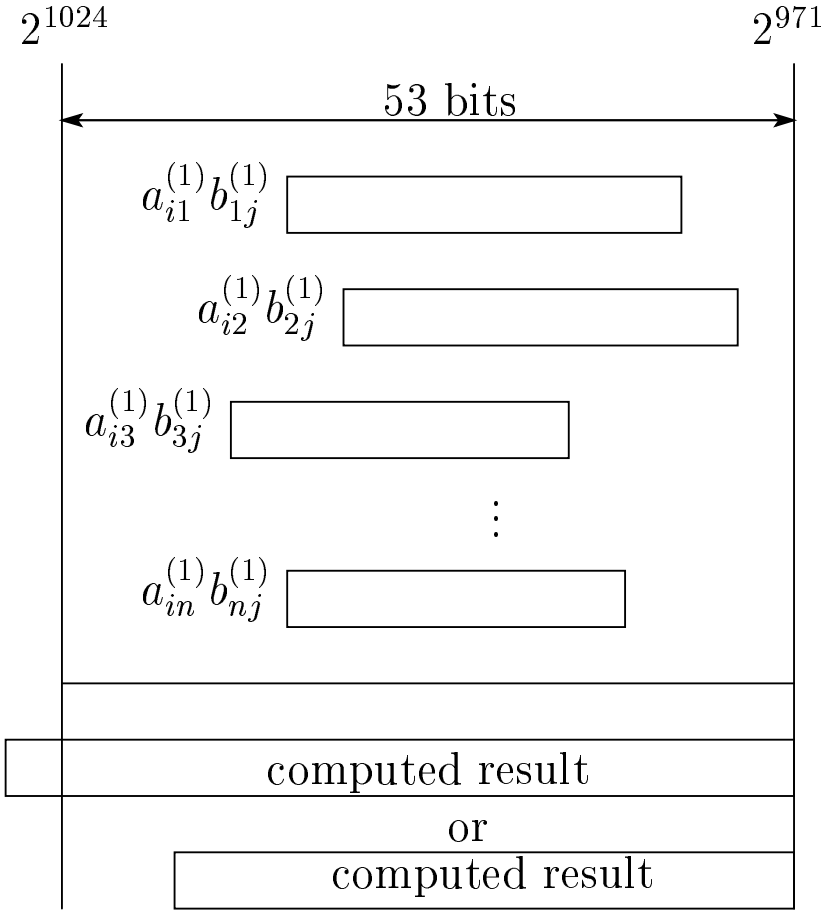

Figure 2: Image of ’error-free’

Step

1. Compute $\tilde{A}^{(1)},\ \tilde{A}^{(2)}, \tilde{B}^{(1)},\ \tilde{B}^{(2)}$
2. Diagonal scaling $\dot{A} = \mathtt{diag}(x)\tilde{A}^{(1)},\ \dot{B} = \tilde{B}^{(1)}\mathtt{diag}(y)$
3. Check $T = \mathrm{fl}(\dot{A}\dot{B})$ contains $\pm\mathtt{Inf}$ or $\mathtt{NaN}$
4. If not, interval computations for

$$\mathtt{diag}(x)^{-1} \cdot T \cdot \mathtt{diag}(y)^{-1} + \tilde{A}^{(1)}\tilde{B}^{(2)} + \tilde{A}^{(2)}B$$

Numerical Example

We compare the tightness of the enclosure of

- M1: $[\mathrm{fl}_{\nabla}(AB),\ \mathrm{fl}_{\Delta}(AB)]$
- M2: The original method
- M3: A posteriori Validation

for $B = \mathtt{gallery}('randsvd', n, cnd, 3), \quad A = \mathtt{inv}(B).$

Numerical Example

Table 1: Maximum Radius for $n = 1000$ ($\beta' = \left\lceil (\log_2 2\sqrt{n} - \log_2 \mathbf{u})/2 \right\rceil$)

cnd	M1	M2	M3
10^2	2.1886e-14	2.2204e-16	1.1102e-16
10^4	1.1438e-12	2.2204e-16	1.6653e-16
10^6	8.1741e-11	2.2204e-16	2.2204e-16
10^8	6.3854e-09	1.1979e-14	2.9039e-15
10^{10}	5.4939e-07	9.1551e-13	2.2889e-13
10^{12}	4.7074e-05	9.1188e-11	2.1964e-11
10^{14}	4.0933e-03	7.7183e-09	1.9427e-09

Numerical Example

Table 2: Maximum Radius for $n = 3000$ ($\beta' = \left\lceil (\log_2 \sqrt{n} - \log_2 \mathbf{u})/2 \right\rceil$)

cnd	M1	M2	M3
10^2	1.6470e-14	2.2204e-16	1.3323e-16
10^4	8.1070e-13	2.2204e-16	2.2204e-16
10^6	5.6153e-11	4.4409e-16	2.2204e-16
10^8	4.4313e-09	1.9826e-14	2.4057e-15
10^{10}	3.6058e-07	1.5753e-12	1.9344e-13
10^{12}	3.0595e-05	1.1507e-10	1.4501e-11
10^{14}	2.6405e-03	1.1664e-08	1.4252e-09

Numerical Example

Table 3: Maximum Radius for $n = 5000$ ($\beta' = \left\lceil (\log_2 \sqrt{n/2} - \log_2 \mathbf{u})/2 \right\rceil$)

cnd	M1	M2	M3
10^2	1.5932e-14	2.2204e-16	1.1102e-16
10^4	7.2014e-13	2.2204e-16	2.2204e-16
10^6	4.8332e-11	4.4409e-16	2.2204e-16
10^8	3.7116e-09	1.7172e-14	2.1559e-15
10^{10}	3.0865e-07	1.2709e-12	1.5906e-13
10^{12}	2.5739e-05	1.2665e-10	1.5930e-11
10^{14}	2.2455e-03	9.8396e-09	1.2168e-09

Numerical Example

Table 4: Maximum Radius for $n = 10000$ ($\beta' = \left\lceil (\log_2 \sqrt{2n} - \log_2 \mathbf{u})/2 \right\rceil$)

cnd	M1	M2	M3
10^2	1.5954e-14	2.2204e-16	1.5543e-16
10^4	6.1814e-13	2.2204e-16	2.2204e-16
10^6	4.2197e-11	6.6613e-16	2.2204e-16
10^8	3.2493e-09	3.1193e-14	3.9090e-15
10^{10}	2.5984e-07	2.8944e-12	3.5628e-13
10^{12}	2.1861e-05	2.1475e-10	2.7377e-11
10^{14}	1.9012e-03	1.7609e-08	2.2116e-09

Conclusion

We proposed the tight enclosure method for matrix multiplication.

The tightness of the computed interval matrix is improved without much additional cost.

Thank you very much
for your attention!