Formalisation[1] of the Lange and Rump's proof of error estimation for iterated sums of FP numbers

Laurence Rideau

May 2023

[1] Using Coq + Flocq library

Motivation (Iterated Sums)

- *Accurate calculation of Euclidean Norms using Double-word arithmetic* with V. Lefèvre, N. Louvet, J.-M. Muller, and J. Picot, TOMS, March 2023

Motivation (Iterated Sums)

- *Accurate calculation of Euclidean Norms using Double-word arithmetic* with V. Lefèvre, N. Louvet, J.-M. Muller, and J. Picot, TOMS, March 2023

  x *is a Double-word (DW) number also called double-double : represented by the sum* $x_h + x_\ell$ *of two FPs* x_h *and* x_ℓ *such that*
  $x_h = \mathrm{RN}(x)$

Motivation (Iterated Sums)

- *Accurate calculation of Euclidean Norms using Double-word arithmetic* with V. Lefèvre, N. Louvet, J.-M. Muller, and J. Picot, TOMS, March 2023

  x is a Double-word (DW) number also called double-double : represented by the sum $x_h + x_\ell$ of two FPs x_h and x_ℓ such that
  $$x_h = \mathrm{RN}(x)$$

- Euclidean Norm is the square root of a sum of squares

Motivation (Iterated Sums)

- *Accurate calculation of Euclidean Norms using Double-word arithmetic* with V. Lefèvre, N. Louvet, J.-M. Muller, and J. Picot, TOMS, March 2023

  x is a Double-word (DW) number also called double-double : represented by the sum $x_h + x_\ell$ of two FPs x_h and x_ℓ such that
  $x_h = \mathrm{RN}(x)$

- Euclidean Norm is the square root of a sum of squares
- iterated sums are used in many places in the accurate calculation of the Euclidean norm, with various conditions and arguments

FP Arithmetic: Basic Building Blocks

Algorithms computing binary operations on floating-point arguments:

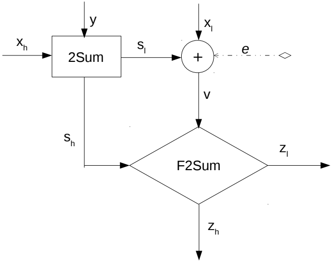
- 2Sum(a, b), Fast2Sum(a, b), Fast2Mult(a, b) treated by the Flocq library
- Exact result in the form of 2 FPs: rounding + *error* (a DW number)

Double-double numbers (DW) Arithmetic

- Different basic algorithms (addition, multiplication and division of a DW and a FP or of 2 DW) proposed by M. Joldes, J.-M. Muller and V. Popescu in 2017
- For each algorithm:
  - an approximation (double-double) of the operation on the operands x and y, such that $x = (x_h, x_\ell)$ is a DW and y is an FP or a DW (y_h, y_ℓ).
  - an error bound
- Formalized in Coq and amended with J.-M. Muller in 2020

Arithmetic for double-doubles: addition

DWPlusFP(x_h, x_ℓ, y): computes an approximation of $(x_h, x_\ell) + y$, with $x = (x_h, x_\ell)$ a DW and y an FP number .



- A relative error bound: $\frac{2 \cdot u^2}{1-2u} = 2u^2 + 4u^3 + 8u^4 + \cdots$
  where $u = 2^{-p} = ulp(1)$ denotes the roundoff error unit.
  ulp(x) is the distance between two consecutive FP numbers in the neighborhood of x.
- if x and y are positive, the bound becomes u^2.

Lange & Rump's Lemma

Lemma (Consequence of Lange and Rump's lemma)

Let $\mathbb{F}$ be an arbitrary subset of $\mathbb{R}$ and let $\tilde{+}$ be an operation in $\mathbb{F}$ with the only assumption that

$$\forall a, b \in \mathbb{F}, |(a \tilde{+} b) - (a + b)| \leq \min\{|a|, |b|\} \tag{1}$$

Let $x_1, x_2, \cdots, x_n$ be elements of $\mathbb{F}$ and define numbers s_i and ϵ_i as follows:

$$\begin{aligned} s_1 &= x_1, \\ s_i &= x_i \tilde{+} s_{i-1} = (x_i + s_{i-1})(1 + \epsilon_i) \text{ for } i = 2, \ldots, n. \end{aligned}$$

We have:

$$\left| s_n - \sum_{i=1}^{n} x_i \right| \leq \sum_{i=2}^{n} |\epsilon_i| \cdot \sum_{i=1}^{n} |x_i|.$$

Lange & Rump's Lemma [2]

Lemma (Lange & Rump)

Let $\mathbb{F}$ be an arbitrary subset of $\mathbb{R}$ and let $\tilde{+}$ be an operation in $\mathbb{F}$ Let x_1, $x_2, \cdots, x_n$ be elements of $\mathbb{F}$ and define numbers s_i and ϵ_i as follows:

$$\begin{aligned} s_1 &= x_1, \\ s_i &= x_i \tilde{+} s_{i-1} = (x_i + s_{i-1})(1+\epsilon_i) \quad \text{for } i = 2, \ldots, n. \end{aligned}$$

Let $\delta_i = (x_i + s_{i-i}) \cdot \epsilon_i$
and for $2 \leq k \leq n$, $\xi_k = \frac{|\delta_k|}{\sum_{i=1}^{k} |x_i| + \sum_{i=1}^{k-1} |\delta_i|}$
Assuming

$$|\delta_k| \leq \left(1 + \sum_{i=1}^{k} \xi_i\right) |x_k| \tag{2}$$

We have:

$$\left| s_n - \sum_{i=1}^{n} x_i \right| \leq \sum_{i=2}^{n} |\epsilon_i| \cdot \sum_{i=1}^{n} |x_i|.$$

[2] *Error estimates for the summation of real numbers with application to FP summation* (2017)

Note that it's easy to show that the property (1)

$$\forall a, b \in \mathbb{F}, |(a \tilde{+} b) - (a + b)| \leq \min\{|a|, |b|\}$$

implies the (2) assumption : $|\delta_k| \leq \left(1 + \sum_{i=1}^{k} \xi_i\right) |x_k|$

Note that it's easy to show that the property (1)

$$\forall a, b \in \mathbb{F}, |(a \tilde{+} b) - (a + b)| \leq \min\{|a|, |b|\}$$

implies the (2) assumption : $|\delta_k| \leq \left(1 + \sum_{i=1}^{k} \xi_i\right) |x_k|$
Operator $\tilde{+}$ examples:

- $a \tilde{+} b = RN(a + b)$ with a, b FP numbers
- $a \tilde{+} b = (DWPlusFP\ \ a\ \ b)$ with a a DW and b an FP
- $a \tilde{+} b = (SloppyDWPlusDW\ \ a\ \ b)$ with a and b DW

NB: $\tilde{+}$ has to verify the necessary condition (1) of the Lange&Rump lemma: RN ✓, DWPlusFP ✓, SloppyDWPlusDW ✗

Sketch of the proof

Prove that:

$$\left| s_n - \sum_{i=1}^{n} x_i \right| \leq \sum_{i=2}^{n} |\epsilon_i| \cdot \sum_{i=1}^{n} |x_i|$$

Sketch of the proof

Prove that:

$$\left| s_n - \sum_{i=1}^{n} x_i \right| \leq \sum_{i=2}^{n} |\epsilon_i| \cdot \sum_{i=1}^{n} |x_i|$$

Let $\Delta_n = s_n - \sum_{i=1}^{n} x_i$

- $\Delta_n = \sum_{i=1}^{n} \delta_i$ and $|\Delta_n| \leq \sum_{i=1}^{n} |\delta_i|$

Sketch of the proof

Prove that:

$$\left| s_n - \sum_{i=1}^{n} x_i \right| \leq \sum_{i=2}^{n} |\epsilon_i| \cdot \sum_{i=1}^{n} |x_i|$$

Let $\Delta_n = s_n - \sum_{i=1}^{n} x_i$

- $\Delta_n = \sum_{i=1}^{n} \delta_i$ and $|\Delta_n| \leq \sum_{i=1}^{n} |\delta_i|$
- $\sum_{i=1}^{n} |\delta_i| \leq \sum_{i=1}^{n} \xi_i \cdot \sum_{i=1}^{n} |x_i|$ *by induction on n*

Sketch of the proof

Prove that:

$$\left| s_n - \sum_{i=1}^{n} x_i \right| \leq \sum_{i=2}^{n} |\epsilon_i| \cdot \sum_{i=1}^{n} |x_i|$$

Let $\Delta_n = s_n - \sum_{i=1}^{n} x_i$

- $\Delta_n = \sum_{i=1}^{n} \delta_i$ and $|\Delta_n| \leq \sum_{i=1}^{n} |\delta_i|$
- $\sum_{i=1}^{n} |\delta_i| \leq \sum_{i=1}^{n} \xi_i \cdot \sum_{i=1}^{n} |x_i|$ *by induction on n*
  case $n = 1$ trivial, then 2 cases
  - $|x_n| < \xi_n \cdot \sum_{i=1}^{n-1} |x_i|$ uses the (2) hypothesis
  - $\xi_n \cdot \sum_{i=1}^{n-1} |x_i| \leq |x_n|$

Sketch of the proof

Prove that:

$$\left| s_n - \sum_{i=1}^{n} x_i \right| \leq \sum_{i=2}^{n} |\epsilon_i| \cdot \sum_{i=1}^{n} |x_i|$$

Let $\Delta_n = s_n - \sum_{i=1}^{n} x_i$

- $\Delta_n = \sum_{i=1}^{n} \delta_i$ and $|\Delta_n| \leq \sum_{i=1}^{n} |\delta_i|$
- $\sum_{i=1}^{n} |\delta_i| \leq \sum_{i=1}^{n} \xi_i \cdot \sum_{i=1}^{n} |x_i|$ *by induction on n*
  case $n = 1$ trivial, then 2 cases
  - $|x_n| < \xi_n \cdot \sum_{i=1}^{n-1} |x_i|$ uses the (2) hypothesis
  - $\xi_n \cdot \sum_{i=1}^{n-1} |x_i| \leq |x_n|$
- For each k, $\xi_k \leq |\epsilon_k|$

Sequential calculation of a sum of squares in DW arithmetic

Algorithm 1 Sequential computation of $\sum_{i=0}^{n-1} a_i^2$ assuming no under/overflow.

1. For $i = 0 \ldots n-1$, compute $(y_i^h, y_i^\ell) = \text{Fast2Mult}(a_i, a_i)$.
   (gives $a_i^2 = y_i^h + y_i^\ell$).

Sequential calculation of a sum of squares in DW arithmetic

Algorithm 1 Sequential computation of $\sum_{i=0}^{n-1} a_i^2$ assuming no under/overflow.

1. For $i = 0 \ldots n-1$, compute $(y_i^h, y_i^\ell) = \text{Fast2Mult}(a_i, a_i)$.
   (gives $a_i^2 = y_i^h + y_i^\ell$).
2. Accumulate the terms y_i^h in DW arithmetic: starting from
   $(x_1^h, x_1^\ell) = \text{2Sum}(y_0^h, y_1^h)$,
   for $i = 2 \ldots n-1$, compute $(x_i^h, x_i^\ell) = \text{DWPlusFP}(x_{i-1}^h, x_{i-1}^\ell, y_i^h)$.

Sequential calculation of a sum of squares in DW arithmetic

Algorithm 1 Sequential computation of $\sum_{i=0}^{n-1} a_i^2$ assuming no under/overflow.

1. For $i = 0 \dots n-1$, compute $(y_i^h, y_i^\ell) = \text{Fast2Mult}(a_i, a_i)$.
   (gives $a_i^2 = y_i^h + y_i^\ell$).
2. Accumulate the terms y_i^h in DW arithmetic: starting from
   $(x_1^h, x_1^\ell) = \text{2Sum}(y_0^h, y_1^h)$,
   for $i = 2 \dots n-1$, compute $(x_i^h, x_i^\ell) = \text{DWPlusFP}(x_{i-1}^h, x_{i-1}^\ell, y_i^h)$.
3. Accumulate the terms y_i^ℓ in FP arithmetic:
   for $i = 0 \dots n-2$, compute $\sigma_{i+1} = \text{RN}(\sigma_i + y_{i+1}^\ell)$, with $\sigma_0 = y_0^\ell$.

Sequential calculation of a sum of squares in DW arithmetic

Algorithm 1 Sequential computation of $\sum_{i=0}^{n-1} a_i^2$ assuming no under/overflow.

1. For $i = 0 \dots n-1$, compute $(y_i^h, y_i^\ell) = \text{Fast2Mult}(a_i, a_i)$.
   (gives $a_i^2 = y_i^h + y_i^\ell$).
2. Accumulate the terms y_i^h in DW arithmetic: starting from
   $(x_1^h, x_1^\ell) = \text{2Sum}(y_0^h, y_1^h)$,
   for $i = 2 \dots n-1$, compute $(x_i^h, x_i^\ell) = \text{DWPlusFP}(x_{i-1}^h, x_{i-1}^\ell, y_i^h)$.
3. Accumulate the terms y_i^ℓ in FP arithmetic:
   for $i = 0 \dots n-2$, compute $\sigma_{i+1} = \text{RN}(\sigma_i + y_{i+1}^\ell)$, with $\sigma_0 = y_0^\ell$.
4. Obtain the approximation to (S_h, S_ℓ) to $\sum_{i=0}^{n-1} a_i^2$ as

$$(S_h, S_\ell) = \text{DWPlusFP}(x_{n-1}^h, x_{n-1}^\ell, \sigma_{n-1}).$$

Block calculation of sum of squares in DW arithmetic

- the a_i are separated into k blocks of m numbers, with $n = k \times m$;
- parallelizing the calculation & obtaining a more accurate result;
- block j ($j = 0, \ldots, k-1$) contains $a_{mj}, a_{mj+1}, \ldots, a_{m(j+1)-1}$.

Algorithm 2 Blockwise computation of $\sum_{i=0}^{n-1} a_i^2$ assuming no under/overflow.

1. for $j = 0, 1, \ldots, k-1$, compute an approximation (Z_j^h, Z_j^ℓ) to $\sum_{i=mj}^{m(j+1)-1} a_i^2$ using the sequential summation algorithm applied to $a_{mj}, a_{mj+1}, a_{mj+2}, \ldots, a_{m(j+1)-1}$;
2. accumulate the terms Z_j^h in DW arithmetic, i.e., starting from $\left(\Sigma_1^h, \Sigma_1^\ell\right) = 2\mathrm{Sum}(Z_0^h, Z_1^h)$, iteratively compute, for $j = 2 \ldots k-1$ the terms $\left(\Sigma_j^h, \Sigma_j^\ell\right) = \mathrm{DWPlusFP}\left(\Sigma_{j-1}^h, \Sigma_{j-1}^\ell, Z_j^h\right)$;
3. accumulate the terms Z_j^ℓ using the conventional "recursive" summation, i.e., for $j = 0 \ldots k-2$, compute $\tau_{j+1} = \mathrm{RN}(\tau_j + Z_{j+1}^\ell)$, with $\tau_0 = Z_0^\ell$;
4. obtain the approximation (S_h, S_ℓ) to $\sum_{i=0}^{n-1} a_i^2$ as

$$(S_h, S_\ell) = \mathrm{DWPlusFP}\left(\Sigma_{k-1}^h, \Sigma_{k-1}^\ell, \tau_{k-1}\right).$$

Formalisation

- Proof of the *general* Lange & Rump lemma (with parametrised bounds and function), refined by use in various cases.
- Proof that the condition (1) implies the condition (2) of the *general* Lange & Rump lemma

Formalisation

- Proof of the *general* Lange & Rump lemma (with parametrised bounds and function), refined by use in various cases.
- Proof that the condition (1) implies the condition (2) of the *general* Lange & Rump lemma
- Some "mistakes" detection
  - SloppyDWPlusDW algorithm: the proof of the condition (1) was wrong
  - error bound of the step 4. of the algorithm

NB: During this work, the formalization was carried out at the same time as the development of the algorithms and the (paper) proofs of correction of the algorithms and the error bounds.

Conclusion

Formalisation for double-double arithmetic :

- addition,multiplication and division of a DW and an FP or of 2 DW
- new error bound for the DWPlusFP algorithm when the arguments are positive
- square root for the double-double numbers
- Lange & Rump Lemma
  - general case with condition (2)
  - proof that (2) is implied by the condition (1) used in the euclidean norms paper

Thank you!