2021 Fox Prize Meeting

June 21st, 2021

Are numerical algorithms accurate at large scale and at low precisions ?

Theo Mary

Sorbonne Université, CNRS, LIP6

Joint work with Nicholas J. Higham

Slides available at https://bit.ly/foxprize21

Floating-point arithmetic

- Standard model of floating-point arithmetic

$$\mathrm{fl}(x \mathbin{\mathrm{op}} y) = (x \mathbin{\mathrm{op}} y)(1+\delta), \quad |\delta| \le u, \text{ for } \mathrm{op} \in \{+, -, \times, \div\}$$

- Example: let $x, y \in \mathbb{R}^3$ and $s = x^T y$

$$\begin{aligned}\widehat{s} &= \Big[\big(x_1 y_1 (1+\delta_1) + x_2 y_2 (1+\delta_2)\big)(1+\delta_3) + x_3 y_3 (1+\delta_4)\Big](1+\delta_5) \\ &= x_1 y_1 (1+\delta_1)(1+\delta_3)(1+\delta_5) + x_2 y_2 (1+\delta_2)(1+\delta_3)(1+\delta_5) \\ &\qquad + x_3 y_3 (1+\delta_4)(1+\delta_5).\end{aligned}$$

- Backward error bound $\widehat{s} = (x + \Delta x)^T y$

Floating-point arithmetic

- Standard model of floating-point arithmetic

$$\mathrm{fl}(x \mathbin{\mathrm{op}} y) = (x \mathbin{\mathrm{op}} y)(1+\delta), \quad |\delta| \le u, \text{ for } \mathrm{op} \in \{+,-,\times,\div\}$$

- Example: let $x, y \in \mathbb{R}^3$ and $s = x^T y$

$$\begin{aligned}\widehat{s} &= \Big[\big(x_1y_1(1+\delta_1) + x_2y_2(1+\delta_2)\big)(1+\delta_3) + x_3y_3(1+\delta_4)\Big](1+\delta_5) \\ &= x_1y_1(1+\delta_1)(1+\delta_3)(1+\delta_5) + x_2y_2(1+\delta_2)(1+\delta_3)(1+\delta_5) \\ &\qquad + x_3y_3(1+\delta_4)(1+\delta_5).\end{aligned}$$

- Backward error bound $\widehat{s} = (x + \Delta x)^T y$

Fundamental lemma in backward error analysis

If $|\delta_k| \le u$ for $k = 1:n$ and $nu < 1$, then

$$\prod_{k=1}^{n}(1+\delta_k) = 1 + \theta_n, \quad |\theta_n| \le \gamma_n := \frac{nu}{1-nu} = nu + O(u^2)$$

Floating-point arithmetic

- Standard model of floating-point arithmetic

$$\mathrm{fl}(x \,\mathrm{op}\, y) = (x \,\mathrm{op}\, y)(1+\delta), \quad |\delta| \le u, \text{ for } \mathrm{op} \in \{+,-,\times,\div\}$$

- Example: let $x, y \in \mathbb{R}^3$ and $s = x^T y$

$$\begin{aligned}\widehat{s} &= \Big[\big(x_1 y_1 (1+\delta_1) + x_2 y_2 (1+\delta_2)\big)(1+\delta_3) + x_3 y_3 (1+\delta_4)\Big](1+\delta_5) \\ &= x_1 y_1 (1+\delta_1)(1+\delta_3)(1+\delta_5) + x_2 y_2 (1+\delta_2)(1+\delta_3)(1+\delta_5) \\ &\qquad + x_3 y_3 (1+\delta_4)(1+\delta_5).\end{aligned}$$

- Backward error bound $\widehat{s} = (x + \Delta x)^T y, \quad |\Delta x| \le \gamma_3$

Fundamental lemma in backward error analysis

If $|\delta_k| \le u$ for $k = 1 : n$ and $nu < 1$, then

$$\prod_{k=1}^{n} (1+\delta_k) = 1 + \theta_n, \quad |\theta_n| \le \gamma_n := \frac{nu}{1-nu} = nu + O(u^2)$$

Backward error analysis

- Inner products $s = x^T y$:

$$\widehat{s} = (x + \Delta x)^T y, \qquad |\Delta x| \le \gamma_n |x|$$

- Matrix–vector products $y = Ax$:

$$\widehat{y} = (A + \Delta A)x, \qquad |\Delta A| \le \gamma_n |A|$$

- LU factorization $A = LU$:

$$\widehat{L}\widehat{U} = A + \Delta A, \qquad |\Delta A| \le \gamma_n |A|$$

- Solution to linear system $Ax = b$:

$$(A + \Delta A)\widehat{x} = b, \qquad |\Delta A| \le (3\gamma_n + \gamma_n^2)|A|$$

$\Rightarrow$ **Error grows as** nu **in NLA: should we worry ?**

Values of u

		Bits			
		Signif. (t)	Exp.	Range	$u = 2^{-t}$
fp64	D	53	11	$10^{\pm 308}$	1×10^{-16}
fp32	S	24	8	$10^{\pm 38}$	6×10^{-8}
fp16	H	11	5	$10^{\pm 5}$	5×10^{-4}
bfloat16	B	8	8	$10^{\pm 38}$	4×10^{-3}

Low precision increasingly supported by hardware:

- Fp16 used by NVIDIA GPUs, AMD Radeon Instinct MI25 GPU, ARM NEON, Fujitsu A64FX ARM
- Bfloat16 used by Google TPU, NVIDIA GPUs, Arm, Intel

Values of u

		Bits Signif. (t)	Exp.	Range	$u = 2^{-t}$
fp64	D	53	11	$10^{\pm 308}$	1×10^{-16}
fp32	S	24	8	$10^{\pm 38}$	6×10^{-8}
fp16	H	11	5	$10^{\pm 5}$	5×10^{-4}
bfloat16	B	8	8	$10^{\pm 38}$	4×10^{-3}

Low precision increasingly supported by hardware:

- Fp16 used by NVIDIA GPUs, AMD Radeon Instinct MI25 GPU, ARM NEON, Fujitsu A64FX ARM
- Bfloat16 used by Google TPU, NVIDIA GPUs, Arm, Intel

$nu > 1$ for $n > 2048$ in fp16 and for $n > 256$ in bfloat16!

Values of n

- Backward error analysis was developed by James Wilkison in the 1960s
- At that time, $n = 100$ was huge!

$\Rightarrow$ n was considered a "constant"

Hence traditional error analysis has paid little attention to n

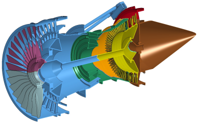
> *The* **constant** *terms in an error bound are the least important parts of error analysis. It is not worth spending much effort to minimize constants because the achievable improvements are usually insignificant.*
>
> *Nick Higham, ASNA 2ed (2002)*

Values of n

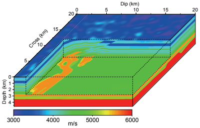
- The #1 computer in the latest TOP500 ranking (Nov. 2020) is there by having solved a linear system of 21 million equations (succesfully passing an accuracy check in double precision)

Values of n

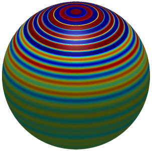
- The #1 computer in the latest TOP500 ranking (Nov. 2020) is there by having solved a linear system of 21 million equations (succesfully passing an accuracy check in double precision)
- Some problems we recently solved with the MUMPS sparse multifrontal solver (for these problems, error grows as $n^{2/3}$):



Jet engine
$n = 105$ millions
Double precision



Seismic imaging
$n = 130$ millions
Single precision



Helioseismology
$n = 384$ millions
Single precision

- Yet, all these problems were solved accurately. Why?

Probabilistic model of rounding errors

- Since the 1960s, researchers have tried modelling the δ_k as random variables to translate the intuition that δ_k of opposite sign cancel each other (von Neumann & Goldstine, Henrici, Hull & Swenson, . . .)

- Wilkinson's rule of thumb: $nu \to \sqrt{n}u$

  > *In general, the statistical distribution of the rounding errors will reduce considerably the function of n occurring in the relative errors. We might expect in each case that this function should be replaced by something which is no bigger than its square root.*
  >
  > — *James Wilkinson, 1961*

Three limitations

Probabilistic analyses remained a “rule of thumb”: why?

Three limitations

Probabilistic analyses remained a "rule of thumb": why?

- **Lack of rigor**
  - First-order analyses
  - Asymptotic statements ("for sufficiently large n")
  - Unspecified probabilities ("with high probability")

Three limitations

Probabilistic analyses remained a "rule of thumb": why?

- **Lack of rigor**
  - First-order analyses
  - Asymptotic statements ("for sufficiently large *n*")
  - Unspecified probabilities ("with high probability")
- **Lack of generality**
  - Only applicable to specific algorithms

Three limitations

Probabilistic analyses remained a "rule of thumb": why?

- **Lack of rigor**
  - First-order analyses
  - Asymptotic statements ("for sufficiently large n")
  - Unspecified probabilities ("with high probability")
- **Lack of generality**
  - Only applicable to specific algorithms
- **Lack of understanding**

  Let us measure the actual backward error, which is given by

$$\eta = \min \left\{ \epsilon > 0 : \widehat{s} = (x + \Delta x)^T y, \quad |\Delta x| \le \epsilon |x| \right\} = \frac{|\widehat{s} - s|}{|x|^T |y|}$$

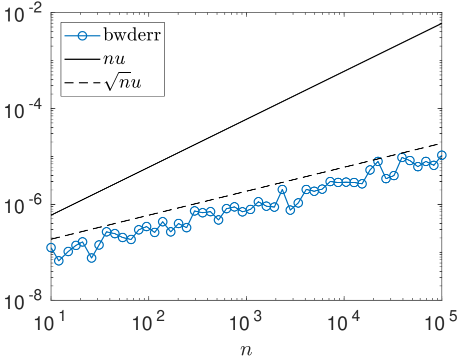
  and compare it to its bound γ_n

Three limitations

Probabilistic analyses remained a “rule of thumb”: why?

- **Lack of rigor**
- **Lack of generality**
- **Lack of understanding**

Inner product in single precision
with random uniform $[0, 1]$ vectors

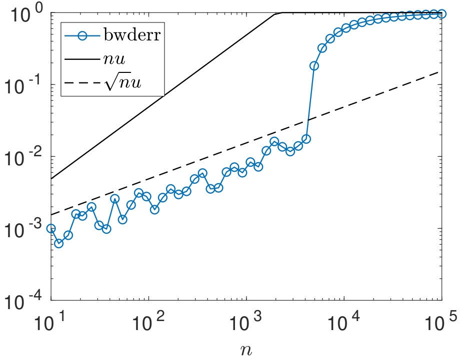


Three limitations

Probabilistic analyses remained a “rule of thumb”: why?

- **Lack of rigor**
- **Lack of generality**
- **Lack of understanding**

Inner product in half precision
with random uniform $[0, 1]$ vectors

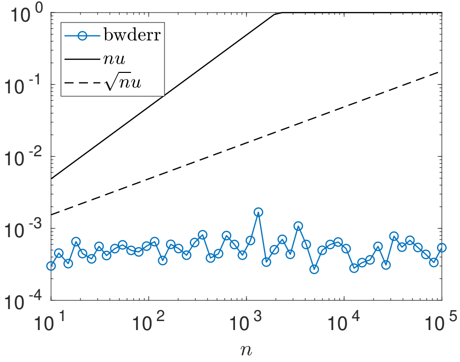


Three limitations

Probabilistic analyses remained a "rule of thumb": why?

- **Lack of rigor**
- **Lack of generality**
- **Lack of understanding**

Inner product in half precision
with random uniform $[-1, 1]$ vectors



Three limitations

Probabilistic analyses remained a "rule of thumb": why?

- **Lack of rigor**
- **Lack of generality**
- **Lack of understanding**

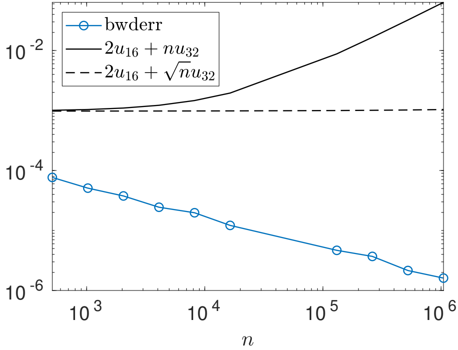
Inner product with tensor cores
with random uniform $[-1, 1]$ vectors
Error bound $2u_{16} + nu_{32}$

Three limitations

Probabilistic analyses remained a "rule of thumb": why?

- **Lack of rigor**
- **Lack of generality**
- **Lack of understanding**

Inner product with tensor cores
with random uniform $[-1, 1]$ vectors
Error bound $2u_{16} + nu_{32}$



Two papers

Nicholas J. Higham and T.M. A New Approach to Probabilistic Rounding Error Analysis, *SIAM J. Sci. Comput.* 41(5):A2815–A2835 (2019).

- First probabilistic backward error analysis, assuming independence of rounding errors

Nicholas J. Higham and T.M. Sharper Probabilistic Backward Error Analysis for Basic Linear Algebra Kernels with Random Data, *SIAM J. Sci. Comput.* 42(5):A3427–A3446 (2020).

- Replaces independence assumption by the weaker mean independence
- Explains difference between $[0, 1]$ and $[-1, 1]$ matrices
- New understanding into the behavior of tensor cores
- Probabilistic forward error bounds
- New algorithm based on shifting matrices in $[-1, 1]$

Probabilistic backward error analysis

Model M

In the computation of interest, the rounding errors δ_k are independent random variables of mean zero: $\mathbb{E}(\delta_k) = 0$.

Probabilistic backward error analysis

Model M

In the computation of interest, the rounding errors δ_k are independent random variables of mean zero: $\mathbb{E}(\delta_k) = 0$.

Probabilistic fundamental lemma

Let δ_k, $k = 1 : n$, satisfy Model M. Then, for any $\lambda > 0$, the relation

$$\prod_{k=1}^{n}(1 + \delta_k) = 1 + \theta_n, \quad |\theta_n| \le \gamma_{\lambda\sqrt{n}}$$

holds with probability at least $P(\lambda) = 1 - 2\exp(-\lambda^2(1-u)^2/2)$.

Probabilistic backward error analysis

Probabilistic fundamental lemma

Let δ_k, $k = 1 : n$, satisfy Model M. Then, for any $\lambda > 0$, the relation

$$\prod_{k=1}^{n}(1+\delta_k) = 1+\theta_n, \quad |\theta_n| \le \gamma_{\lambda\sqrt{n}}$$

holds with probability at least $P(\lambda) = 1 - 2\exp(-\lambda^2(1-u)^2/2)$.

Key features:

- valid to all orders
- valid for all n
- explicit probability $P(\lambda)$ (but pessimistic)
- can be applied **in a systematic way:** $\gamma_n \to \gamma_{\lambda\sqrt{n}}$

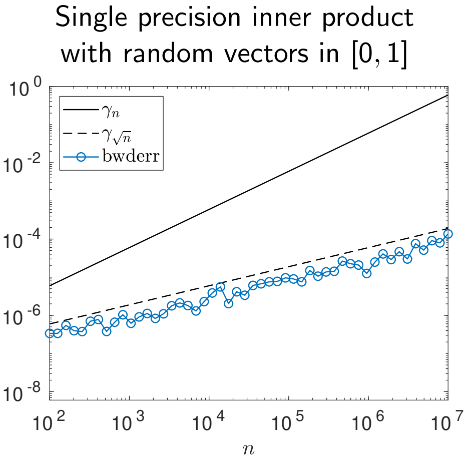
$$\widehat{s} = (x+\Delta x)^T y, \qquad |\Delta x| \le \gamma_{\lambda\sqrt{n}}|x|$$

$$\widehat{y} = (A+\Delta A)x, \qquad |\Delta A| \le \gamma_{\lambda\sqrt{n}}|A|$$

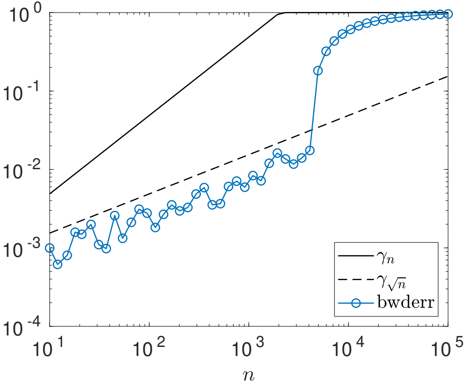
$$\widehat{L}\widehat{U} = A+\Delta A, \qquad |\Delta A| \le \gamma_{\lambda\sqrt{n}}|A|$$

$$(A+\Delta A)\widehat{x} = b, \qquad |\Delta A| \le (3\gamma_{\lambda\sqrt{n}} + \gamma_{\lambda\sqrt{n}}^2)|A|$$

Typical result



Half precision inner product
with random vectors in $[0, 1]$



- Summation of a very large number of nonnegative terms ($n \gg 10^3$ in half precision) eventually violates Model M
- Issue known as stagnation: small increments get obliterated by large partial sum

A refined model

Model M'

Let the computation of interest generate rounding errors δ_1, δ_2, ... in that order, with $|\delta_k| \le u$. The δ_k are (possibly dependent) random variables of mean zero and mean independent of the previous δ_1, ..., δ_{k-1}, i.e., $\mathbb{E}(\delta_k \mid \delta_1, \dots, \delta_{k-1}) = \mathbb{E}(\delta_k) = 0$.

A refined model

Model M'

Let the computation of interest generate rounding errors δ_1, δ_2, ... in that order, with $|\delta_k| \leq u$. The δ_k are (possibly dependent) random variables of mean zero and mean independent of the previous δ_1, ..., δ_{k-1}, i.e., $\mathbb{E}(\delta_k \mid \delta_1, \dots, \delta_{k-1}) = \mathbb{E}(\delta_k) = 0$.

Probabilistic fundamental lemma

Let δ_k, $k = 1 : n$, satisfy Model M'. Then, for any $\lambda > 0$, the relation

$$\prod_{k=1}^{n}(1 + \delta_k) = 1 + \theta_n, \quad |\theta_n| \leq \gamma_{\lambda\sqrt{n}}$$

holds with probability at least $P(\lambda) = 1 - 2\exp(-\lambda^2(1-u)^2/2)$.

Proof

Martingale: a sequence of random variables satisfying

- $\mathbb{E}(|S_k|) < \infty$
- $\mathbb{E}(S_{k+1} \mid S_0, \ldots, S_k) = S_k$

Proof

Martingale: a sequence of random variables satisfying

- $\mathbb{E}(|S_k|) < \infty$
- $\mathbb{E}(S_{k+1} \mid S_0, \ldots, S_k) = S_k$

Azuma–Hoeffding inequality

Let $S_0, \ldots, S_n$ be a martingale such that $|S_{k+1} - S_k| \leq c$. Then

$$|S_n - S_0| \leq \lambda\sqrt{n}c$$

holds with probability at least $P(\lambda) = 1 - \exp(-2\lambda^2)$.

Proof

Martingale: a sequence of random variables satisfying

- $\mathbb{E}(|S_k|) < \infty$
- $\mathbb{E}(S_{k+1} \mid S_0, \ldots, S_k) = S_k$

Azuma–Hoeffding inequality

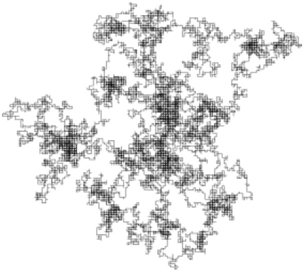
Let $S_0, \ldots, S_n$ be a martingale such that $|S_{k+1} - S_k| \leq c$. Then

$$|S_n - S_0| \leq \lambda\sqrt{n}c$$

holds with probability at least $P(\lambda) = 1 - \exp(-2\lambda^2)$.

- Let $S_n = \prod_{k=1}^{n}(1 + \delta_i) = 1 + \theta_n$
- S_n is martingale (with $S_0 = 1$)
- $|S_{k+1} - S_k| \leq |\delta_{k+1} S_k| \leq u(1 + |\theta_n|) =: c$
- Azuma–Hoeffding: $|\theta_n| = |S_n - S_0| \leq \lambda\sqrt{n}u(1 + |\theta_n|)$
- $|\theta_n| \leq \frac{\lambda\sqrt{n}u}{1-\lambda\sqrt{n}u} = \gamma_{\lambda\sqrt{n}}$

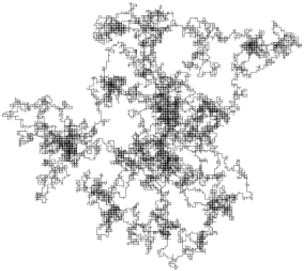
Random walks



Let S_k be the position at step k

- S_{k+1} depends on S_k
- However, identical chance of going in any direction $\Rightarrow \mathbb{E}(S_{k+1} \mid S_0, \ldots S_k) = S_k$

Random walks

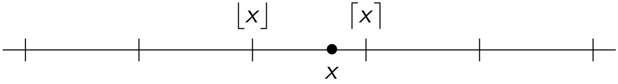


Let S_k be the position at step k

- S_{k+1} depends on S_k
- However, identical chance of going in any direction
  $\Rightarrow \mathbb{E}(S_{k+1} \mid S_0, \ldots S_k) = S_k$

- Model M' identifies finite-precision computations to random walks
  - Allows rounding errors at a given step to depend on previous errors
  - Only assumes the expected error (conditioned by previous errors) to be zero

Stochastic rounding



- With stochastic rounding

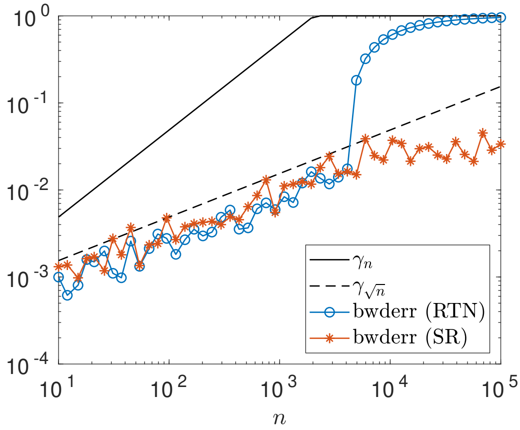
$$\mathrm{fl}(x) = \begin{cases} \lceil x \rceil & \text{with probability } p = \frac{x - \lfloor x \rfloor}{\lceil x \rceil - \lfloor x \rfloor} \\ \lfloor x \rfloor & \text{with probability } 1 - p = \frac{\lceil x \rceil - x}{\lceil x \rceil - \lfloor x \rfloor} \end{cases}$$

where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote the operators that round down and up

Connolly, Higham, and M. (2021): rounding errors produced by SR satisfy Model M' (with $u \leftarrow 2u$)

$\Rightarrow$ Probabilistic $\gamma_{\lambda\sqrt{n}}$ bound holds unconditionally: the rule of thumb is a rule for SR

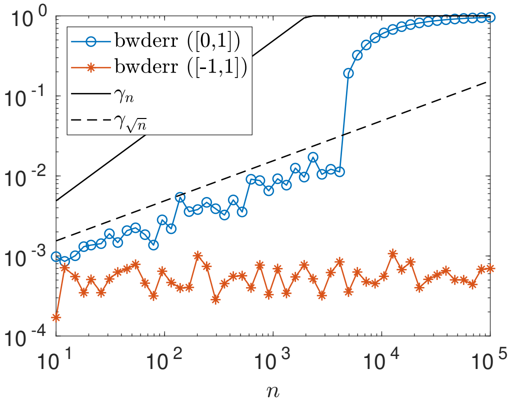
Same example, now with SR



- Stagnation explains success of SR in neural network training (Gupta et al., 2015)
- SR also prevents stagnation in PDEs (Croci & Giles, 2021)

$[-1, 1]$ data

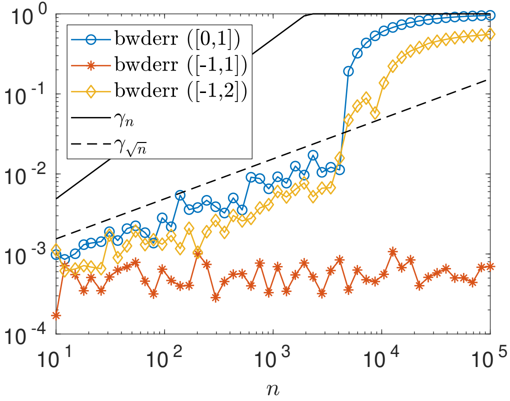
Previous results for $[0, 1]$ random uniform data.
What about $[-1, 1]$ data ?



$[0, 1]$ vectors only have positive elements $\Rightarrow$ too special ?

$[-1, 1]$ data

Previous results for $[0, 1]$ random uniform data.
What about $[-1, 1]$ data ?



$[0, 1]$ vectors only have positive elements $\Rightarrow$ too special ?
No! $[-1, 1]$ vectors are the special ones!

Role of the data, intuitively

- Recall that $\eta = \frac{|\widehat{s}-s|}{|x|^T|y|}$
- Under Model M', $|\widehat{s} - s| \le \lambda\sqrt{n}u \max_k |s_k|$, where s_k is the partial inner product of the first k elements of x and y
- Because of cancellation, cannot bound $|s_k|$ by $|x^T y|$ but only by $|x|^T|y|$ in general. But what about specific x_i, y_i?
  - $x_i, y_i \in \text{Unif}([0,1]) \Rightarrow |s_k| = O(n)$
  - $x_i, y_i \in \text{Unif}([-1,1]) \Rightarrow |s_k| = O(\sqrt{n})$

$\Rightarrow$ Backward error smaller by a factor $\sqrt{n}$

Role of the data, more formally

Model M''

In addition to the assumptions of Model M', assume that in the inner product $s = x^T y$, x_i and y_i are random independent variables such that $\mathbb{E}(x_i y_i) = \mu$, $\mathbb{E}(|x_i y_i|) = \mu_+$, and $|x_i y_i| \le C$.

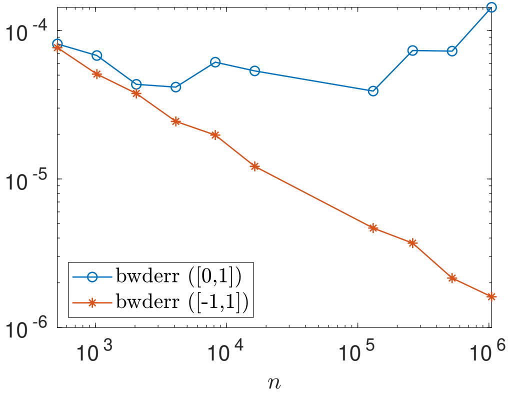
Probabilistic bwd error bound for random inner products

Let $s = x^T y$. Under Model M'', for any $\lambda > 0$, the backward error bound

$$\eta = \frac{|\widehat{s} - s|}{|x|^T |y|} \le \frac{\lambda \mu \sqrt{n} + \lambda^2 C}{\mu_+ - \lambda C / \sqrt{n}} \cdot u + O(u^2)$$

holds with probability $P(\lambda) = 1 - 2(n+1)\exp\left(-\lambda^2/2\right)$

Tensor cores



Round x and y to fp16, then compute $s = x^T y$ in fp32 arithmetic

$$\eta \le \frac{\left|\sum_{i=1}^{n} x_i y_i \epsilon_i\right|}{|x|^T|y|} + n u_{32}, \quad |\epsilon_i| \le 2u_{16} + u_{16}^2$$

$$\le \frac{u_{16}}{\sqrt{n}} + n u_{32} \quad \text{under Model M'' for zero-mean vectors}$$

Shifting to zero mean for accuracy

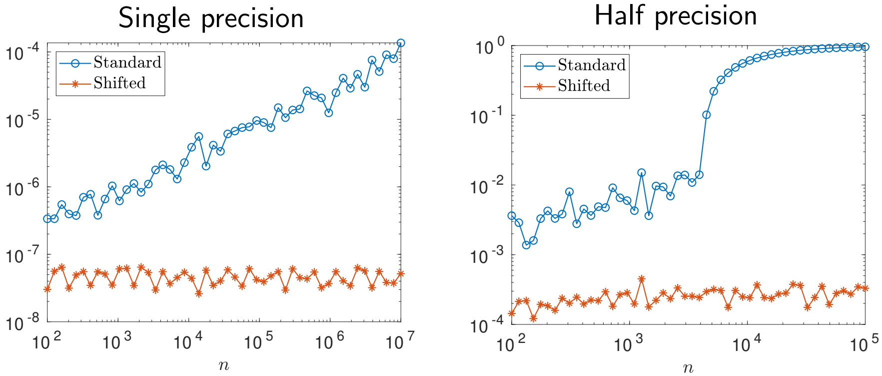
Idea: given x_i, y_i of mean $\mu \neq 0$, let $z_i = x_i - \mu$ and compute $s = z^T y + n\mu$, then $\eta \leq cu$ for some c independent of n

Cost: $2n$ flops but for $C = AB$, where $A, B, C \in \mathbb{R}^{n\times n}$ the cost of the algorithm below is in $O(n^2)$ instead of $O(n^3)$

$$\widetilde{A} \leftarrow A - xe^T$$
$$C \leftarrow \widetilde{A}B + x(e^T B)$$

where x_i = mean of ith row of A and e is the vector full of ones

Backward error (for [0, 1] data)



Conclusions

$$\gamma_n \to \gamma_{\lambda\sqrt{n}} \quad \text{with probability } P(\lambda)$$

- Accuracy guarantees for larger problems/lower precisions
  - In probabilistic sense
  - Under some assumptions, which are enforced by SR

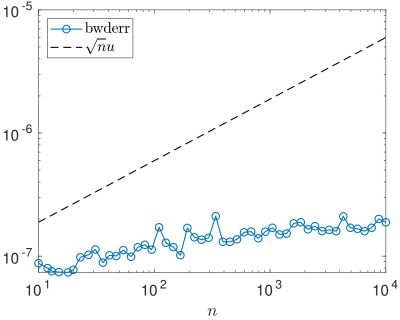
- New insights and understanding into the behavior of finite-precision computations
  - Stagnation
  - Rounding mode
  - Mean of the data
  - Tensor cores

Open problem: LU factorization and linear systems

Doolittle's formula for $A = LU$

$$\ell_{ik} = \Big(a_{ik} - \sum_{j=1}^{k-1} \ell_{ij} u_{jk}\Big)/u_{kk}, \qquad u_{kj} = a_{kj} - \sum_{i=1}^{k-1} \ell_{ki} u_{ij}$$

The inner products arising in LU factorization are not random! And yet. . .



Thanks! Questions?