Exception handling for the BLAS and LAPACK

Weslley da Silva Pereira, University of Colorado Denver

BLIS Retreat 2022

September 22, UT Austin

<□ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Coauthors ¹

- Jim Demmel, UC Berkeley,
- Jack Dongarra, U Tennessee, Knoxville,
- Mark Gates, U Tennessee, Knoxville,
- Greg Henry, Intel Corp.,
- Julien Langou, U Colorado, Denver,
- Xiaoye Li, LBNL,
- Piotr Luszczek, U Tennessee, Knoxville,
- Weslley Pereira, U Colorado, Denver,
- Jason Riedy, Lucata Corp.,
- Cindy Rubio-Gonzalez, UC Davis.

¹Proposed Exception Handling for the BLAS and LAPACK: https://www.arxiv.org/abs/2207.09281

Examples of bad exception handling

- Ariane 501, 1996. "[An] exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value."²
- Roborace, 2020. "the system somehow managed to produce a NaN (not a number) value and all verification logic was designed to work only with numbers." ³



²https://www-users.cse.umn.edu/~arnold/disasters/ariane.html
³https:

//www.reddit.com/r/formula1/comments/jk9jrg/comment/gai2951 = 🔊 ୯୯ 3/14

Agenda

<ロ > < @ > < E > < E > を を の へ で 4/14

 Inconsistencies in the Reference BLAS and LAPACK libraries (Some also in BLIS)

2 Proposed exception handling

3 Early tests for the proposed consistent standard

4 Concluding remarks

One of the goals of this talk: Seek feedback

Inconsistencies in the Reference BLAS

IxAMAX: Return the index of the largest entry.

- Inconsistency #1: isamax([0, NaN, 2]) = 3, isamax([NaN, 0, 2]) = 1,
 Inconsistency #2:
 - icamax([OV+i*OV, Inf+i*0]) = 1, icamax([.6*OV+i*.6*OV, .7*OV+i*.7*OV]) = 1, where OV is the overflow threshold.
- Proposed consistent fix:

Return index to the first NaN, else to the first Inf, else to the first largest finite entry.

Inconsistencies in the Reference BLAS

IxAMAX: Return the index of the largest entry.

• Inconsistency #1:

isamax([0, NaN, 2]) = 3, isamax([NaN, 0, 2]) = 1,

- Inconsistency #2: icamax([OV+i*OV, Inf+i*O]) = 1, icamax([.6*OV+i*.6*OV, .7*OV+i*.7*OV]) = 1, where OV is the overflow threshold.
- Proposed consistent fix:

Return index to the first NaN, else to the first Inf, else to the first largest finite entry.

BLIS 0.9.0 has inconsistency #2 only.

Inconsistencies in the Reference BLAS

 $\mathsf{xGER:} \ \mathbf{A} := \mathbf{A} + \alpha \mathbf{x} \mathbf{y}^{\mathsf{T}}.$

Current Reference BLAS: If α = 0, return A. If y_j = 0 then A_{ij} := A_{ij}. (What if x_i = Inf or NaN?) There is no check for x_i = 0.

xTRSV: Solve Tx = b for x.

 Current Reference BLAS: Skip trailing 0s in b. If x_j = 0 then skip update x_i := x_i - T_{ij}x_j. No checks for 0s when solving T^Tx = b.

Proposed consistent fix:

Keep check for $\alpha = 0$ but not for $x_i = 0$, $y_i = 0$ or $b_i = 0$.

<□ > < @ > < E > < E > ● E の Q @ 7/14

xGETF2 + xGETRS: Solve Ax = b for x.

• Example:
$$A = \begin{bmatrix} 1 & 0 \\ NaN & 2 \end{bmatrix}$$
 and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

• Combination of inconsistencies in the Reference BLAS:

• IXAMAX chooses $A_{11} = 1$, not $A_{21} = NaN$, as pivot.

$$A = \begin{bmatrix} 1 & 0 \\ NaN & 2 \end{bmatrix}$$

xGETF2 + xGETRS: Solve Ax = b for x.

• Example:
$$A = \begin{bmatrix} 1 & 0 \\ NaN & 2 \end{bmatrix}$$
 and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

• Combination of inconsistencies in the Reference BLAS:

- IXAMAX chooses $A_{11} = 1$, not $A_{21} = NaN$, as pivot.
- xGER finds 0 in A_{12} and does not use the NaN. (Also on BLIS 0.9.0)

$$A_{22} := A_{22} - NaN \cdot 0$$

< □ > < □ > < □ > < Ξ > < Ξ > Ξ の < ?/14

< □ > < □ > < □ > < Ξ > < Ξ > Ξ の < ?/14

xGETF2 + xGETRS: Solve Ax = b for x.

• Example:
$$A = \begin{bmatrix} 1 & 0 \\ NaN & 2 \end{bmatrix}$$
 and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Combination of inconsistencies in the Reference BLAS:

- IXAMAX chooses $A_{11} = 1$, not $A_{21} = NaN$, as pivot.
- xGER finds 0 in A₁₂ and does not use the NaN.

• LU factorization:
$$L = \begin{bmatrix} 1 & 0 \\ NaN & 1 \end{bmatrix}$$
, $U = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

xGETF2 + xGETRS: Solve Ax = b for x.

• Example:
$$A = \begin{bmatrix} 1 & 0 \\ NaN & 2 \end{bmatrix}$$
 and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Combination of inconsistencies in the Reference BLAS:

- IXAMAX chooses $A_{11} = 1$, not $A_{21} = NaN$, as pivot.
- xGER finds 0 in A_{12} and does not use the NaN.
- LU factorization: $L = \begin{bmatrix} 1 & 0 \\ NaN & 1 \end{bmatrix}$, $U = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$
- The first call for xTRSV solves Ly = b and finds $y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. In the second iteration, $y_2 := 1 NaN \cdot y_1$ and $y_1 = 0$.

xGETF2 + xGETRS: Solve Ax = b for x.

• Example:
$$A = \begin{bmatrix} 1 & 0 \\ NaN & 2 \end{bmatrix}$$
 and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Combination of inconsistencies in the Reference BLAS:

• IXAMAX chooses $A_{11} = 1$, not $A_{21} = NaN$, as pivot.

xGER finds 0 in A₁₂ and does not use the NaN.

• LU factorization:
$$L = \begin{bmatrix} 1 & 0 \\ NaN & 1 \end{bmatrix}$$
, $U = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

• The first call for xTRSV solves Ly = b and finds $y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

• The final solution is
$$x = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$$

NaN does not propagate!

BLIS 0.9.0 propagates NaNs in this example.

Examples of unavoidable inconsistencies

Summation order: sum([OV,OV,-OV,-OV]) can be NaN, Inf, -Inf or O, where OV is the overflow threshold.

IEEE 754-2019 defines min and max that propagate NaNs. Compilers need time to adopt those rules. $^{\rm 4}$

```
1 min( qNaN, 1.0 ) = qNaN ! gfortran 12.2
2 min( 1.0, qNaN ) = 1.0 ! gfortran 12.2
```

Absolute value, product, and quotient of complex numbers:

- They are not specified by IEEE 754-2019.
- There is space for inconsistencies, e.g., ⁵

```
1 cmplx(Inf,0.0)*cmplx(Inf,Inf) = cmplx(Inf,Inf) ! ifort 2021
2 cmplx(Inf,0.0)*cmplx(Inf,Inf) = cmplx(NaN,NaN) ! gfortran 12
3
4 abs(subNormal) = subNormal ! ifort 2021
5 abs(cmplx(0.0,subNormal)) = subNormal ! ifort 2021 (-00)
6 abs(cmplx(0.0,subNormal)) = 0.0 ! ifort 2021 (-03)
```

Proposed exception handling

"If NaNs or Infs won't just disappear!"

- 1 The program will still complete.
- **2** Moreover, either:
 - NaNs and Infs propagate to the output.
 - NaNs and Infs are dealt with internally.
 - NaNs and Infs **do not propagate** to the output **in special cases**, e.g., $C := 0 \cdot AB + 0 \cdot C$ in GEMM.
- **3** LAPACK routines will report (using INFO and more).

Current design ⁶ takes into account:

- Several user requests.
- Different stakeholders.
- Relevant xSDK Community Policies for Library Development.

⁶https://www.arxiv.org/abs/2207.09281 < □ > < ♂ > < ₹ > < ₹ > ₹ < ? > \$ 9/14

The EC (Error Checking) interfaces

- Example: SGESV_EC(N, NRHS, A, LDA, IPIV, B, LDB, INFO, FLAG_REPORT, INFO_ARRAY, CONTEXT)
- 3 more arguments at end:
 - FLAG_REPORT(1:2) FLAG_REPORT(1): WHAT to report. FLAG_REPORT(2): HOW to report.
 - INFO_ARRAY(FIXED_SIZE) can report INFO + details on Infs and NaNs on input, output and internal calls.
 - CONTEXT pointer to opaque data. Can report, for example, exceptions on different threads.
- Legacy interface will be maintained as a wrapper.

Some early testBLAS ⁹ results

Expected output from IxAMAX: Return index to the first NaN, else to the first Inf, else to the first largest finite entry.

BLAS library ⁷	I{S,D}AMAX	I{C,Z}AMAX
Apple Accelerate 12.2.1	X 8	Х
BLIS 0.9.0	Pass	Х
IBM ESSL 6.3.0	Х	Х
Intel MKL 2022.1.0	Pass	Х
LAPACK 3.9.1	Х	Х
LAPACK 3.10.1	Х	Х
LAPACK 3.11-beta	Pass	Pass
OpenBLAS 0.3.8	Х	Х

⁷Apple Accelerate on macOS Monterey 12.2.1 using Apple clang version 13.1.6. IBM ESSL on Summit node using GNU compiler v9.1.0. Others on Ubuntu 20.04.4 LTS using GNU compiler v9.4.0.

⁸X: Does not follow proposed standard.

Some early testBLAS ¹² results

Expected output from xNRM2: NaN if in input, else Inf if in input, else "accurate answer" (possibly Inf).

BLAS library ¹⁰	{S,D}NRM2 (finite input)	{S,D}NRM2
Apple Accelerate 12.2.1	X ¹¹	Х
BLIS 0.9.0	Х	Х
IBM ESSL 6.3.0	Pass	Pass
Intel MKL 2022.1.0	Pass	Pass
LAPACK 3.9.1	Х	Х
LAPACK 3.10.1	Pass	Pass
LAPACK 3.11-beta	Pass	Pass
OpenBLAS 0.3.8	Pass	Pass

¹⁰Apple Accelerate on macOS Monterey 12.2.1 using Apple clang version 13.1.6. IBM ESSL on Summit node using GNU compiler v9.1.0. Others on Ubuntu 20.04.4 LTS using GNU compiler v9.4.0.

¹¹X: Does not follow proposed standard.

¹²https://www.github.com/tlapack/testBLAS → () → (

Proposed tasks

- During installation of LAPACK, check abs, multiplication and division of complex, and min and max behave as expected. (Partially done in LAPACK 3.10.1)
- 2 Modify LAPACKE drivers, so that they return an error flag if input has NaNs or Infs.
- Odify LAPACK, so that routines that compute norms of complex matrices signal NaNs and Infs on input.
- Fix Reference BLAS and implement test code. (The new IxAMAX is planned to be in LAPACK 3.11)
- **5** Validate the EC interfaces for a few LAPACK routines.
- 6 Design more general test code.
- Implement the EC interfaces to the rest of LAPACK.

Thank you!

Questions? weslley.pereira@ucdenver.edu

Proposed Exception Handling for the BLAS and LAPACK: https://www.arxiv.org/abs/2207.09281 (Shorter version will be available in the SC'22)

Funded by:

NSF under the project Basic ALgebra LIbraries for Sustainable Technology with Interdisciplinary Collaboration (BALLISTIC).



National Science Foundation