An overview of mixed-precision methods in scientific computing

Matteo Croci

Center for Optimization and Statistical Learning Seminar.
Northwestern University, 6 October 2022

The University of Texas at Austin
Oden Institute for Computational Engineering and Sciences

Overview

1. Introduction and background
2. Optimization
3. Numerical linear algebra
4. Numerical solution of partial differential equations
5. Conclusions

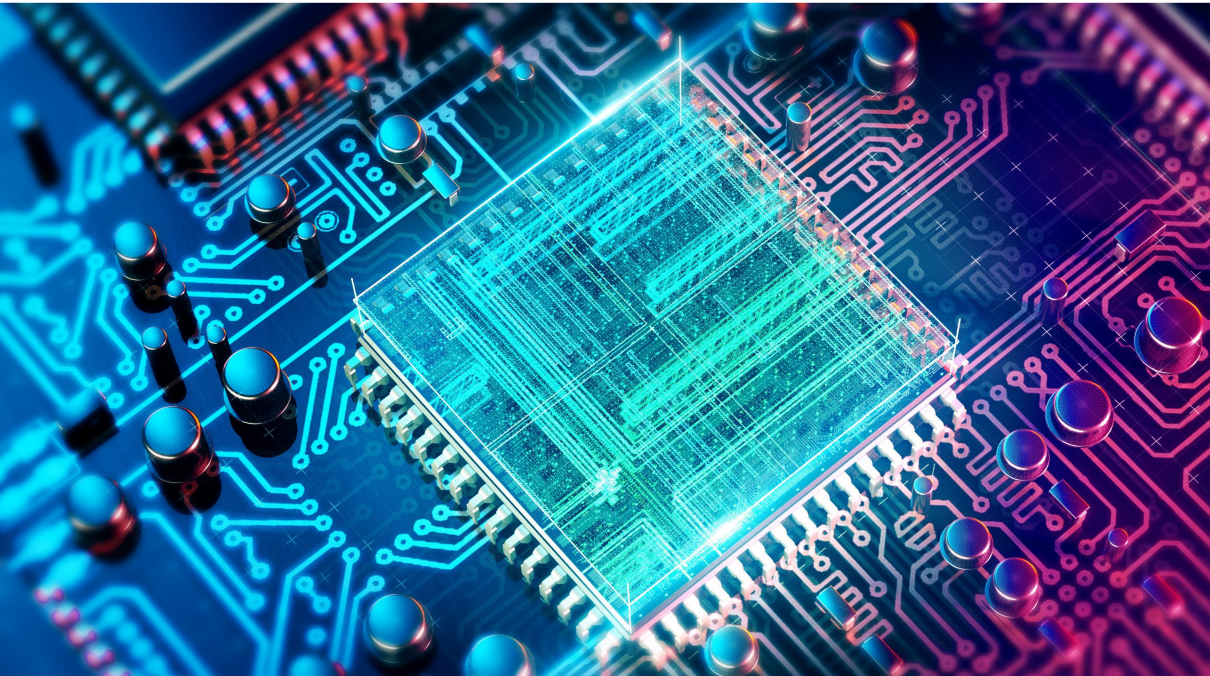
Note: too broad a field to include everything. I will present a few examples per topic.

1. Introduction and background

Main references:

- A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, et al. A survey of numerical linear algebra methods utilizing mixed-precision arithmetic. *The International Journal of High Performance Computing Applications*, 35(4):344–369, 2021
- N. J. Higham and T. Mary. Mixed precision algorithms in numerical linear algebra. *Acta Numerica*, 31:347–414, 2022
- M. Croci, M. Fasi, N. J. Higham, T. Mary, and M. Mikaitis. Stochastic rounding: implementation, error analysis and applications. *Royal Society Open Science*, 9:211631, 2022
- M. P. Connolly, N. J. Higham, and T. Mary. Stochastic rounding and its probabilistic backward error analysis. *SIAM Journal on Scientific Computing*, 43(1):566–585, 2021
- N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002

Reduced- and mixed-precision algorithms



Reduced- and mixed-precision algorithms

Reduced-precision algorithms

Reduced-precision algorithms obtain an as accurate solution as possible given the precision while avoiding catastrophic rounding error accumulation.

[1] Review articles: [Abdelfattah et al. 2021], [Higham and Mary 2021], [C. et al. 2021].

Reduced- and mixed-precision algorithms

Reduced-precision algorithms

Reduced-precision algorithms obtain an as accurate solution as possible given the precision while avoiding catastrophic rounding error accumulation.

Mixed-precision algorithms

Mixed-precision algorithms combine low- and high-precision computations in order to benefit from the performance gains of reduced-precision while retaining good accuracy.

[1] Review articles: [Abdelfattah et al. 2021], [Higham and Mary 2021], [C. et al. 2021].

Reduced- and mixed-precision algorithms

Reduced-precision algorithms

Reduced-precision algorithms obtain an as accurate solution as possible given the precision while avoiding catastrophic rounding error accumulation.

Mixed-precision algorithms

Mixed-precision algorithms combine low- and high-precision computations in order to benefit from the performance gains of reduced-precision while retaining good accuracy.

- This is now a very active field of investigation[1] with many new developments led mainly by the numerical linear algebra and machine learning communities.
- Many new RP/MP algorithms in scientific computing and data science.
- There is still much to discover on the topic.

[1] Review articles: [Abdelfattah et al. 2021], [Higham and Mary 2021], [C. et al. 2021].

Floating point formats

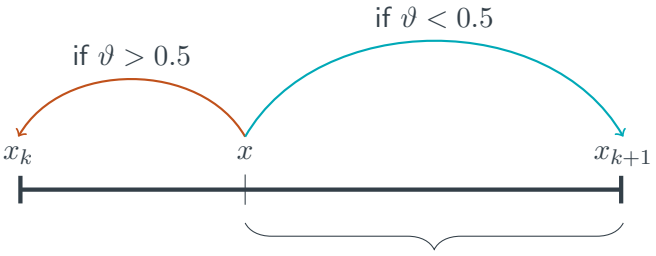
Format	unit roundoff u	Range
bfloat16 (half)	$2^{-8} \approx 3.91 \times 10^{-3}$	$10^{\pm 38}$
fp16 (half)	$2^{-11} \approx 4.88 \times 10^{-4}$	$10^{\pm 5}$
fp32 (single)	$2^{-24} \approx 5.96 \times 10^{-8}$	$10^{\pm 38}$
fp64 (double)	$2^{-53} \approx 1.11 \times 10^{-16}$	$10^{\pm 308}$

Important: don't just focus on u, range is an extremely important factor. Scaling and squeezing techniques are central for a correct reduced-precision implementation.

Recent trend in scientific computing: u is getting larger: all major chip manufacturers (AMD, ARM, NVIDIA, Intel, ...) have commercialized chips (CPUs, GPUs, TPUs, FPGAs, ...) supporting low-precision computations.

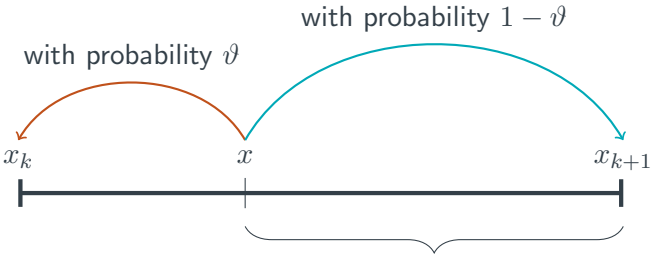
Half vs double max speedups: $\times 4$ on CPUs, $\times 32$ on A100 NVIDIA GPUs.

Round to nearest



$$\mathsf{fl}(x) = x(1+\delta), \quad \text{with} \quad |\delta| \leq u.$$

Stochastic rounding (review article [C. et al. 2022])



$\mathsf{sr}(x) = x(1+\delta(\omega)), \quad |\delta| \leq 2u, \quad$ and $\mathbb{E}[sr(x)] = x, \quad \mathbb{E}[\delta_i|\delta_1, \ldots, \delta_{i-1}] = \mathbb{E}[\delta_i] = 0.$

Limited (yet growing) hardware support. Many new applications in Sci. Comp. and ML.

2. Optimization

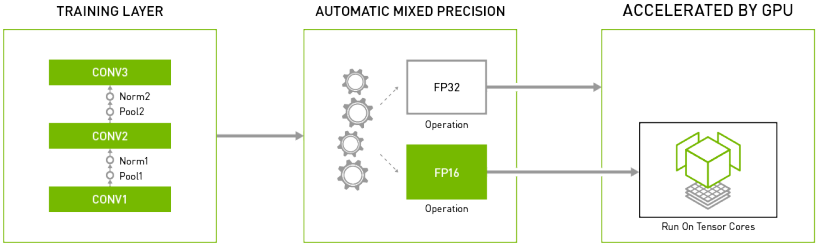
Note: Not my field of expertise. Post-seminar discussions are welcome!

Main references:

- N. Mellempudi, S. Srinivasan, D. Das, and B. Kaul. Mixed precision training with 8-bit floating point. *arXiv preprint arXiv:1905.12334*, 2019
- F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Fifteenth annual conference of the international speech communication association*. Microsoft, 2014
- Y. Xie, R. H. Byrd, and J. Nocedal. Analysis of the BFGS method with errors. *SIAM Journal on Optimization*, 30(1):182–209, 2020
- F. Tisseur. Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 22(4):1038–1057, 2001
- C. Kelley. Newton's method in mixed precision. *SIAM Review*, 64(1):191–211, 2022

Reduced- and mixed-precision first-order methods in machine learning

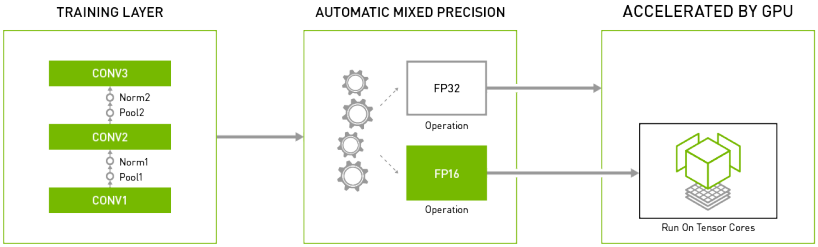
- The machine learning community has been the main driver of experimentation in this field and GPU tensor cores really help making this efficient.



`https://developer.nvidia.com/automatic-mixed-precision`.

Reduced- and mixed-precision first-order methods in machine learning

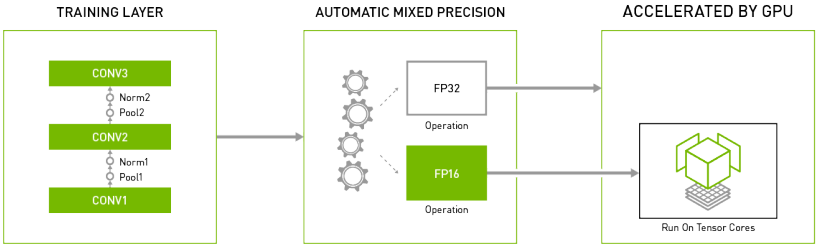
- The machine learning community has been the main driver of experimentation in this field and GPU tensor cores really help making this efficient.
- "Easy" to implement: a single line of code allows to switch to single/half mixed-precision in TensorFlow.



https://developer.nvidia.com/automatic-mixed-precision.

Reduced- and mixed-precision first-order methods in machine learning

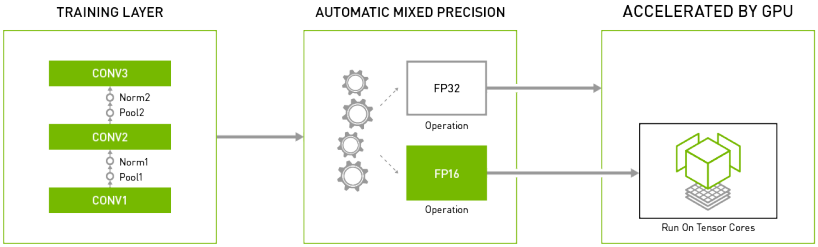
- The machine learning community has been the main driver of experimentation in this field and GPU tensor cores really help making this efficient.
- "Easy" to implement: a single line of code allows to switch to single/half mixed-precision in TensorFlow.
- Stochastic rounding has been successfully employed to squeeze stochastic gradient descent into quarter precision, see [Mellempudi et al. 2019].



`https://developer.nvidia.com/automatic-mixed-precision`.

Reduced- and mixed-precision first-order methods in machine learning

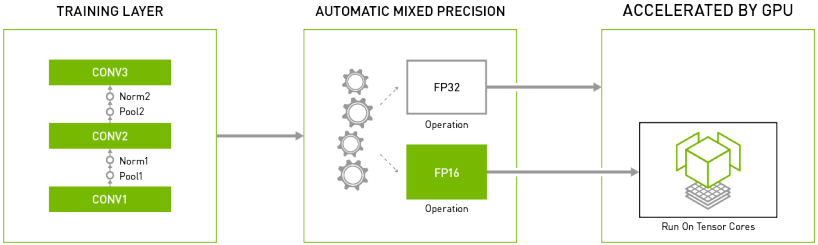
- The machine learning community has been the main driver of experimentation in this field and GPU tensor cores really help making this efficient.
- "Easy" to implement: a single line of code allows to switch to single/half mixed-precision in TensorFlow.
- Stochastic rounding has been successfully employed to squeeze stochastic gradient descent into quarter precision, see [Mellempudi et al. 2019].
- 1-bit Precision has been employed in sign gradient descent, cf. [Seide et al. 2014].



https://developer.nvidia.com/automatic-mixed-precision.

Reduced- and mixed-precision first-order methods in machine learning

- The machine learning community has been the main driver of experimentation in this field and GPU tensor cores really help making this efficient.
- "Easy" to implement: a single line of code allows to switch to single/half mixed-precision in TensorFlow.
- Stochastic rounding has been successfully employed to squeeze stochastic gradient descent into quarter precision, see [Mellempudi et al. 2019].
- 1-bit Precision has been employed in sign gradient descent, cf. [Seide et al. 2014].
- From a theoretical point of view: many open questions.



https://developer.nvidia.com/automatic-mixed-precision.

Reduced-precision second-order optimization $\subseteq$ optimization with noise?

Limited results in the optimization literature are specific to rounding errors. However, there is work on optimization with noise (see e.g. [Xie, Byrd & Nocedal 2020]).

Reduced-precision second-order optimization $\subseteq$ optimization with noise?

Limited results in the optimization literature are specific to rounding errors. However, there is work on optimization with noise (see e.g. [Xie, Byrd & Nocedal 2020]).

What I'd be curious to know: To what extent does the existing theory apply to inexact arithmetic? What are the implementation challenges?

Reduced-precision second-order optimization $\subseteq$ optimization with noise?

Limited results in the optimization literature are specific to rounding errors. However, there is work on optimization with noise (see e.g. [Xie, Byrd & Nocedal 2020]).

What I'd be curious to know: To what extent does the existing theory apply to inexact arithmetic? What are the implementation challenges?

Need to consider:

- Noisy function and derivative evaluations:

$$\hat{f}(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon_f(\boldsymbol{x}), \qquad \text{with} \quad |\varepsilon_0(\boldsymbol{x})| \leq \varepsilon_0, \, \forall \boldsymbol{x}.$$
$$\widehat{\nabla^i f}(\boldsymbol{x}) = \nabla^i f(\boldsymbol{x}) + \varepsilon_i(\boldsymbol{x}), \qquad \text{with} \quad \|\varepsilon_i(\boldsymbol{x})\| \leq \varepsilon_i, \, \forall \boldsymbol{x}, \; i = 1, 2.$$

- Inexact Newton system solves, linesearch, local models, subproblems, ...

How does the relative size of the errors affect convergence?
Which steps can I perform more or less accurately?

Designing mixed-precision optimization methods - some thoughts

- Stochastic rounding may be useful if theory assumes zero-mean independent errors.

Designing mixed-precision optimization methods - some thoughts

- Stochastic rounding may be useful if theory assumes zero-mean independent errors.
- Designing routines for reduced-/mixed-precision derivative evaluations may be problem-dependent and not straightforward in general.

Designing mixed-precision optimization methods - some thoughts

- Stochastic rounding may be useful if theory assumes zero-mean independent errors.
- Designing routines for reduced-/mixed-precision derivative evaluations may be problem-dependent and not straightforward in general.
- Barring underflow/overflow rounding errors are typically linear in u so noise constants are easy to estimate if evaluation routines are type-flexible.

Designing mixed-precision optimization methods - some thoughts

- Stochastic rounding may be useful if theory assumes zero-mean independent errors.
- Designing routines for reduced-/mixed-precision derivative evaluations may be problem-dependent and not straightforward in general.
- Barring underflow/overflow rounding errors are typically linear in u so noise constants are easy to estimate if evaluation routines are type-flexible.
- Mixed-precision NLA methods can be applied and incorporated, e.g. in Newton linear system solves, quasi-Newton updates, trust-region subproblems, ...

Newton's method in floating-point arithmetic [Tisseur 2001], [Kelley 2022]

Results from [Kelley 2022]. Analysis for "vanilla" Newton: no linesearch.

Newton's method in floating-point arithmetic [Tisseur 2001], [Kelley 2022]

Results from [Kelley 2022]. Analysis for "vanilla" Newton: no linesearch.

Assumptions:

1. Std assumptions for Newton local q-quadratic convergence. Lipschitz Hessian.
2. A backward error bound holds for linear solves (e.g. LU factorization is used).

Newton's method in floating-point arithmetic [Tisseur 2001], [Kelley 2022]

Results from [Kelley 2022]. Analysis for "vanilla" Newton: no linesearch.

Assumptions:

1. Std assumptions for Newton local q-quadratic convergence. Lipschitz Hessian.
2. A backward error bound holds for linear solves (e.g. LU factorization is used).

Newton step:

Newton's method in floating-point arithmetic [Tisseur 2001], [Kelley 2022]

Results from [Kelley 2022]. Analysis for "vanilla" Newton: no linesearch.

Assumptions:

1. Std assumptions for Newton local q-quadratic convergence. Lipschitz Hessian.
2. A backward error bound holds for linear solves (e.g. LU factorization is used).

Newton step:

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k - (\nabla^2 f(\hat{\boldsymbol{x}}_k) + \varepsilon_2(\hat{\boldsymbol{x}}_k) + \varepsilon_s(\hat{\boldsymbol{x}}_k))^{-1}(\nabla f(\hat{\boldsymbol{x}}_k) + \varepsilon_1(\hat{\boldsymbol{x}}_k)) + \varepsilon_a(\hat{\boldsymbol{x}}_k),$$

$$\|\varepsilon_1(\boldsymbol{x})\| \le \varepsilon_1, \quad \text{(gradient error)}, \quad \|\varepsilon_2(\boldsymbol{x})\| \le \varepsilon_2, \quad \text{(Hessian error)}, \qquad \forall \boldsymbol{x},$$
$$\|\varepsilon_a(\boldsymbol{x})\| \le \varepsilon_a, \quad \text{(update error)}, \quad \|\varepsilon_s(\boldsymbol{x})\| \le \varepsilon_s, \quad \text{(linear solve error)}, \quad \forall \boldsymbol{x}.$$

Theorem (Kelley 2022)

Under the above assumptions, the error $e_k = \hat{x}_k - x^*$ satisfies

$$\|\boldsymbol{e}_{k+1}\| = O\left(\|\boldsymbol{e}_k\|^2 + (\varepsilon_2 + \varepsilon_s)\|\boldsymbol{e}_k\| + \varepsilon_1 + \varepsilon_a\right)$$

Mixed-precision Newton

Theorem (Kelley 2022)

Under the above assumptions, the error $e_k = \hat{\boldsymbol{x}}_k - \boldsymbol{x}^*$ *satisfies*

$$\|\boldsymbol{e}_{k+1}\| = O\left(\|\boldsymbol{e}_k\|^2 + (\varepsilon_2 + \varepsilon_s)\|\boldsymbol{e}_k\| + \varepsilon_1 + \varepsilon_a\right)$$

Note: Inexact Hessian and linear solves impact convergence rate, but not limiting accuracy. Gradient and update errors do not harm rate, but affect limiting accuracy.

Warning: hidden constants proportional to $\|\nabla^2 f(\boldsymbol{x}^*)^{-1}\|$, $\kappa(\nabla^2 f)$, and problem size.

Mixed-precision Newton

Theorem (Kelley 2022)

Under the above assumptions, the error $e_k = \hat{\boldsymbol{x}}_k - \boldsymbol{x}^*$ satisfies

$$\|\boldsymbol{e}_{k+1}\| = O\left(\|\boldsymbol{e}_k\|^2 + (\varepsilon_2 + \varepsilon_s)\|\boldsymbol{e}_k\| + \varepsilon_1 + \varepsilon_a\right)$$

Note: Inexact Hessian and linear solves impact convergence rate, but not limiting accuracy. Gradient and update errors do not harm rate, but affect limiting accuracy.

Warning: hidden constants proportional to $\|\nabla^2 f(\boldsymbol{x}^*)^{-1}\|$, $\kappa(\nabla^2 f)$, and problem size.

Typical mixed-precision strategy: high-precision gradient evaluations and update and low-precision Hessian evaluation/approximation and inversion so that, e.g.

$$\varepsilon_1, \varepsilon_a = O(u^2);\ \varepsilon_2, \varepsilon_s = O(u) \quad \Longrightarrow \quad \|\boldsymbol{e}_{k+1}\| \approx O\left(\|\boldsymbol{e}_k\|^2 + u^2\right).$$

Since the reduction in the rate occurs when $\|\boldsymbol{e}_k\| \leq O(u)$ for which $\|\boldsymbol{e}_{k+1}\| = O(u^2)$.

3. Numerical linear algebra

Two topics:

1. Mixed-precision iterative refinement.
2. Mixed-precision Krylov subspace methods.

Review articles (citing all mentioned references)**:**

- A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, et al. A survey of numerical linear algebra methods utilizing mixed-precision arithmetic. *The International Journal of High Performance Computing Applications*, 35(4):344–369, 2021
- N. J. Higham and T. Mary. Mixed precision algorithms in numerical linear algebra. *Acta Numerica*, 31:347–414, 2022

Mixed-precision iterative refinement for $Ax = b$ [Langou et al. 2006], [Carson & Higham 2017-18]

Mixed-precision iterative refinement for $Ax = b$ [Langou et al. 2006], [Carson & Higham 2017-18]

Apply mixed-precision Newton to $A\boldsymbol{x} - \boldsymbol{b} = 0$. Use two precisions u, u^2.

Mixed-precision iterative refinement for $Ax = b$ [Langou et al. 2006], [Carson & Higham 2017-18]

Apply mixed-precision Newton to $A\boldsymbol{x} - \boldsymbol{b} = 0$. Use two precisions u, u^2.

Mixed-precision iterative refinement

Solve $A\boldsymbol{x}_0 = \boldsymbol{b}$ using LU factorization in precision u and store the LU factors.
For $k = 1, 2, \ldots$

1. Compute residual $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$ at precision u^2.
2. Solve $A\boldsymbol{d}_k = \boldsymbol{r}_k$ at precision u by re-using the LU factors.
3. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k$ at precision u^2.

Since $\|\boldsymbol{e}_0\| = O(u)$ the previous theorem gives that

$$\|\boldsymbol{e}_1\| = O(\|\boldsymbol{e}_0\|^2 + u\|\boldsymbol{e}_0\| + u^2) = O(u^2).$$

Mixed-precision iterative refinement for $Ax = b$ [Langou et al. 2006], [Carson & Higham 2017-18]

Apply mixed-precision Newton to $A\boldsymbol{x} - \boldsymbol{b} = 0$. Use two precisions u, u^2.

Mixed-precision iterative refinement

Solve $A\boldsymbol{x}_0 = \boldsymbol{b}$ using LU factorization in precision u and store the LU factors.
For $k = 1, 2, \ldots$

1. Compute residual $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$ at precision u^2.
2. Solve $A\boldsymbol{d}_k = \boldsymbol{r}_k$ at precision u by re-using the LU factors.
3. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k$ at precision u^2.

Since $\|\boldsymbol{e}_0\| = O(u)$ the previous theorem gives that

$$\|\boldsymbol{e}_1\| = O(\|\boldsymbol{e}_0\|^2 + u\|\boldsymbol{e}_0\| + u^2) = O(u^2).$$

Advantages: LU factorization performed only once in low precision. Limiting accuracy dictated by u^2 provided $\kappa_\infty(A)$ is small enough.

GMRES-IR [Carson & Higham 2017-18, Amestoy et al. 2021]

Now use three precisions: $u_l \geq u \geq u^2$. In [Amestoy et al. 2021] they use five.

GMRES-IR

Solve $A\boldsymbol{x}_0 = \boldsymbol{b}$ using LU factorization in precision u_l and store the LU factors.
For $k = 1, 2, \ldots$

1. Compute residual $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$ at precision u^2.
2. Solve $A\boldsymbol{d}_k = \boldsymbol{r}_k$ at precision u by using GMRES with $U^{-1}L^{-1}$ as preconditioner and matrix-vector products performed at precision u^2.
3. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k$ at precision u.

GMRES-IR [Carson & Higham 2017-18, Amestoy et al. 2021]

Now use three precisions: $u_l \geq u \geq u^2$. In [Amestoy et al. 2021] they use five.

GMRES-IR

Solve $A\boldsymbol{x}_0 = \boldsymbol{b}$ using LU factorization in precision u_l and store the LU factors.
For $k = 1, 2, \ldots$

1. Compute residual $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$ at precision u^2.
2. Solve $A\boldsymbol{d}_k = \boldsymbol{r}_k$ at precision u by using GMRES with $U^{-1}L^{-1}$ as preconditioner and matrix-vector products performed at precision u^2.
3. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k$ at precision u.

Result:

- Provided that $\kappa_\infty(A) \ll u^{-1}$ we obtain a limiting accuracy of $O(u)$ where the hidden constant is independent from $\kappa_\infty(A)$.
- This approach is efficient since again the LU factorization is performed only once and in low precision, and GMRES typically converges in a handful of iterations.
- GMRES-IR is more robust to ill-conditioning than LU-based iterative refinement.

Mixed-precision iterative refinement in the literature

Mixed-precision iterative refinement is at the heart of many recent mixed-precision developments in numerical linear algebra, including:

- Sparse approximate factorizations (e.g. replace LU with a sparse approximation), cf. [Amestoy et al. 2022].
- Least square problems (see e.g. [Carson et al. 2020]).
- Eigenvalue problems (see e.g. [Tisseur 2001]).
- Multigrid (see e.g. [Tamstorf et al. 2021] and [McCormick et al. 2021]).
- Krylov subspace methods, cf. [Anzt et al. 2010, Lindquist et al. 2021].

Mixed-precision Krylov methods and preconditioning

Mixed-precision Krylov methods and preconditioning

Complex theory: the theory describing the finite precision behavior of iterative methods is extensive and complex. Review on Lanczos-CG: [Meurant & Strakoš 2006].

Practical methods: much work focuses on showing what improves performance in practice rather than on theoretical results.

Mixed-precision Krylov methods and preconditioning

Complex theory: the theory describing the finite precision behavior of iterative methods is extensive and complex. Review on Lanczos-CG: [Meurant & Strakoš 2006].

Practical methods: much work focuses on showing what improves performance in practice rather than on theoretical results.

Three approaches: (see review articles for more details and info):

1. **Iterative refinement.** Use lower precision in inner solver.

Mixed-precision Krylov methods and preconditioning

Complex theory: the theory describing the finite precision behavior of iterative methods is extensive and complex. Review on Lanczos-CG: [Meurant & Strakoš 2006].

Practical methods: much work focuses on showing what improves performance in practice rather than on theoretical results.

Three approaches: (see review articles for more details and info):

1. **Iterative refinement.** Use lower precision in inner solver.
2. **MP preconditioning.** Apply/implement preconditioner in low precision.

Mixed-precision Krylov methods and preconditioning

Complex theory: the theory describing the finite precision behavior of iterative methods is extensive and complex. Review on Lanczos-CG: [Meurant & Strakoš 2006].

Practical methods: much work focuses on showing what improves performance in practice rather than on theoretical results.

Three approaches: (see review articles for more details and info):

1. **Iterative refinement.** Use lower precision in inner solver.
2. **MP preconditioning.** Apply/implement preconditioner in low precision.
3. **MP iterative methods.** Adaptively change precision of inner products/matvecs.

4. Numerical solution of partial differential equations

Main references:

- M. Croci, M. Fasi, N. J. Higham, T. Mary, and M. Mikaitis. Stochastic rounding: implementation, error analysis and applications. *Royal Society Open Science*, 9:211631, 2022
- M. Klöwer, S. Hatfield, M. Croci, P. D. Düben, and T. N. Palmer. Fluid simulations accelerated with 16 bits: Approaching 4x speedup on A64FX by squeezing ShallowWaters.jl into Float16. *Journal of Advances in Modeling Earth Systems*, 2021
- M. Croci and M. B. Giles. Effects of round-to-nearest and stochastic rounding in the numerical solution of the heat equation in low precision. *IMA Journal of Numerical Analysis*, 2022. URL https://doi.org/10.1093/imanum/drac012
- M. Croci and G. R. de Souza. Mixed-precision explicit stabilized Runge–Kutta methods for single-and multi-scale differential equations. *Journal of Computational Physics*, 2022

4a. Towards climate simulations in half precision

Joint with: M. Klöwer and T. N. Palmer (University of Oxford),
S. Hatfield and P. D. Düben (European Centre for Medium-Range Weather Forecasts).

Algorithm type: reduced-precision (half).

Main references:

- M. Klöwer, S. Hatfield, M. Croci, P. D. Düben, and T. N. Palmer. Fluid simulations accelerated with 16 bits: Approaching 4x speedup on A64FX by squeezing ShallowWaters.jl into Float16. *Journal of Advances in Modeling Earth Systems*, 2021

Towards climate simulations in half precision [Klöwer et al. 2021]

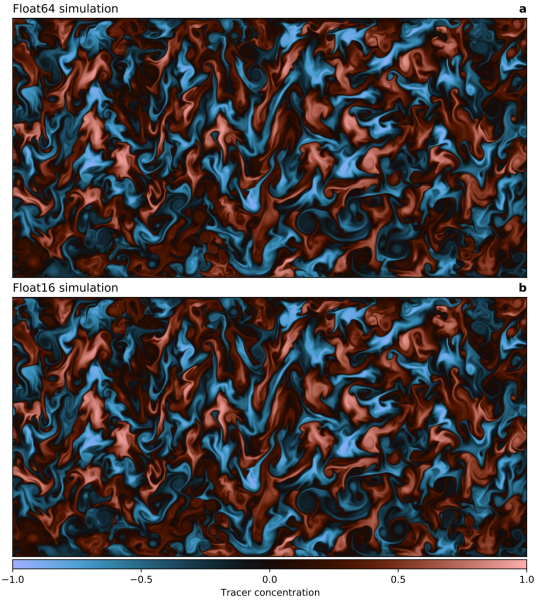
Shallow-water eqs for 2D oceanic flow:

$$\begin{cases} \dot{\boldsymbol{v}} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} + \hat{\boldsymbol{z}} \times \boldsymbol{v} = -\nabla \eta + \Delta^2 \boldsymbol{v} - \boldsymbol{v} + \boldsymbol{F}, \\ \dot{\eta} + \nabla \cdot (\boldsymbol{v} h) = 0, \\ \dot{q} + \boldsymbol{v} \cdot \nabla q = -\tau (q - q_0). \end{cases}$$

Numerical scheme: explicit 4th-order timestepping on a staggered grid.

Techniques used for fp16 simulations:

- Scaling and squeezing.
- Kahan compensated summation.
- Performed using A64FX chips on Fugaku (1st in TOP500).



Note: all other results in this part of the talk use *precision emulation* in software.

4b. Solving parabolic PDEs in half precision

Joint with: M. B. Giles (University of Oxford)

Algorithm type: reduced-precision (half), using **stochastic rounding**.

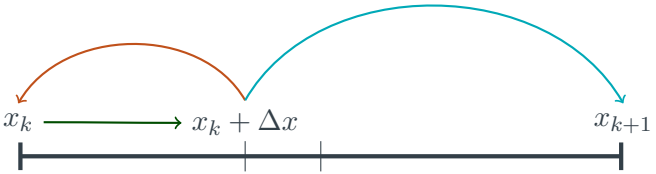
Main references:

- M. Croci and M. B. Giles. Effects of round-to-nearest and stochastic rounding in the numerical solution of the heat equation in low precision. *IMA Journal of Numerical Analysis*, 2022. URL https://doi.org/10.1093/imanum/drac012
- M. Croci, M. Fasi, N. J. Higham, T. Mary, and M. Mikaitis. Stochastic rounding: implementation, error analysis and applications. *Royal Society Open Science*, 9:211631, 2022

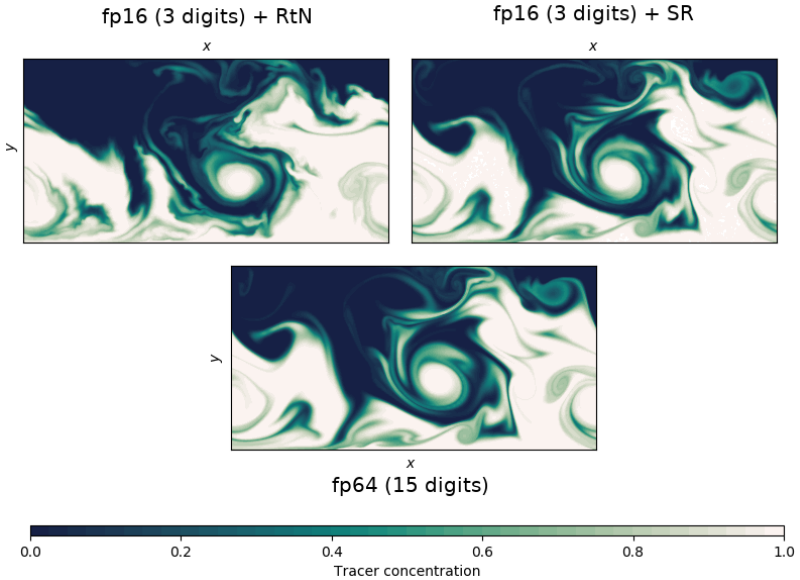
RtN might cause stagnation

RtN might cause stagnation

SR is resilient to stagnation



Interesting results by Milan Klöwer (University of Oxford)



Note: not just due to stagnation, SR decorrelates errors and causes error cancellation!

RtN vs SR

Why is RtN in low precision bad for parabolic PDEs?

a) Stagnation:

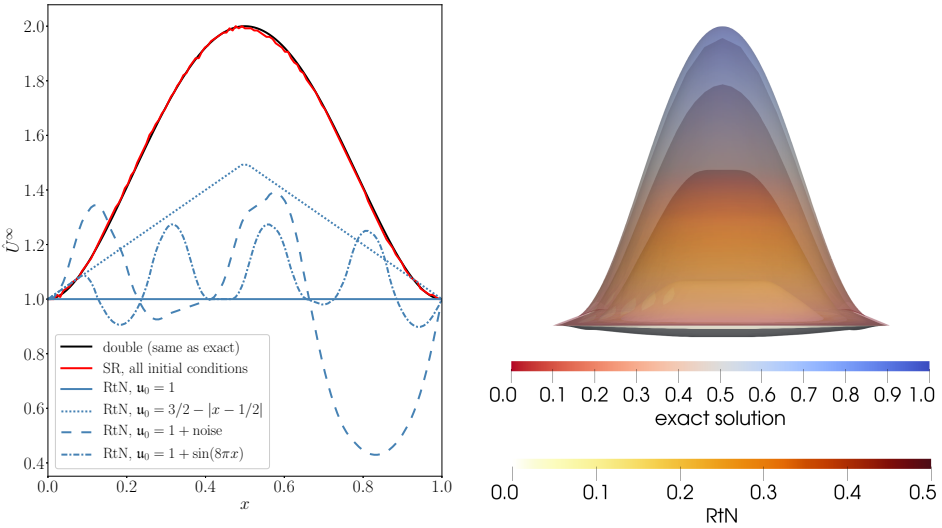
- RtN always stagnates for sufficiently small Δt.

b) Global error:

- RtN rounding errors are strongly correlated and grow rapidly until stagnation.

SR fixes all these issues!

a) Stagnation (heat equation, left 1D, right 2D)



RtN computations are discretization and initial condition dependent. SR works!

b) Global rounding errors [C. and Giles 2020]

Let $\varepsilon^n \in \mathbb{R}^K$ be the vector containing all rounding errors introduced at time step n.
Define the global rounding error $\boldsymbol{E}^n = \hat{\boldsymbol{U}}^n - \boldsymbol{U}^n$. It can be shown that

$$\boldsymbol{E}^{n+1} = S\boldsymbol{E}^n + \varepsilon^n.$$

Traditional results for ODEs [Henrici 1962-1963, Arató 1983]: ε^n is $O(\Delta t^2)$.

We can distinguish two cases:

RtN: we can only assume the worst-case scenario, $|\varepsilon_i^n| = O(u)$ for all n, i.

SR: the ε_i^n are zero-mean, independent in space and mean-independent in time.

b) Global rounding errors [C. and Giles 2020]

Let $\varepsilon^n \in \mathbb{R}^K$ be the vector containing all rounding errors introduced at time step n. Define the global rounding error $\boldsymbol{E}^n = \hat{\boldsymbol{U}}^n - \boldsymbol{U}^n$. It can be shown that

$$\boldsymbol{E}^{n+1} = S\boldsymbol{E}^n + \varepsilon^n.$$

Traditional results for ODEs [Henrici 1962-1963, Arató 1983]: ε^n is $O(\Delta t^2)$.

We can distinguish two cases:

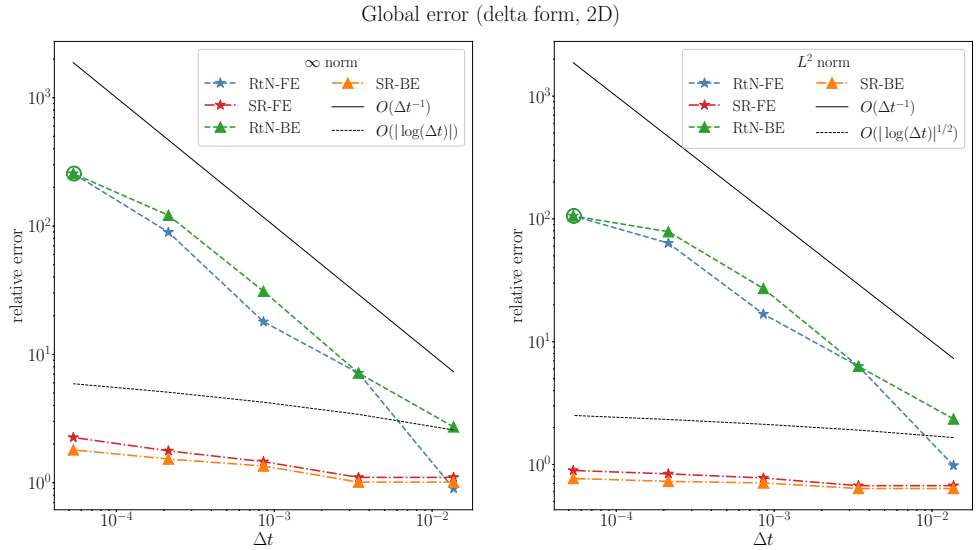
RtN: we can only assume the worst-case scenario, $|\varepsilon_i^n| = O(u)$ for all n, i.

SR: the ε_i^n are zero-mean, independent in space and mean-independent in time.

Mode	Norm	1D	2D	3D
RtN	L^2, ∞	$O(u\Delta t^{-1})$	$O(u\Delta t^{-1})$	$O(u\Delta t^{-1})$
SR	$\mathbb{E}[\|\|\cdot\|\|_\infty^2]^{1/2}$	$O(u\Delta t^{-1/4}\ell(\Delta t)^{1/2})$	$O(u\ell(\Delta t))$	$O(u\ell(\Delta t)^{1/2})$
SR	$\mathbb{E}[\|\|\cdot\|\|_{L^2}^2]^{1/2}$	$O(u\Delta t^{-1/4})$	$O(u\ell(\Delta t)^{1/2})$	$O(u)$

Asymptotic global rounding error blow-up rates; $\ell(\Delta t) = |\log(\Delta t)|$.

b) Global rounding errors (2D heat equation)



Note: relative error = error $\times$ $(u||\boldsymbol{U}^N||)^{-1}$

4c. Mixed-precision explicit Runge-Kutta methods

Joint with: G. Rosilho De Souza (USI Lugano).

Algorithm type: mixed-precision (double/bfloat16) using **round-to-nearest**.

Main reference:

- M. Croci and G. R. de Souza. Mixed-precision explicit stabilized Runge–Kutta methods for single-and multi-scale differential equations. *Journal of Computational Physics*, 2022

Framework and objective

We consider mixed-precision **explicit** RK schemes for the solution of ODEs in the form

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad \boldsymbol{y}(0) = \boldsymbol{y}_0,$$

where $\boldsymbol{f}(t, \boldsymbol{y})$ is sufficiently smooth, and from now on set $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{y}(t))$ for simplicity.

Objective

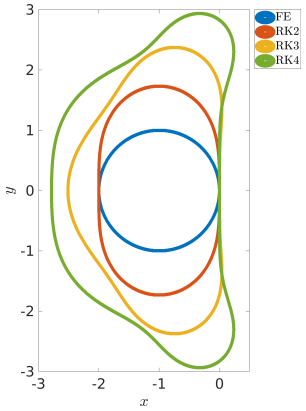
Evaluate $\boldsymbol{f}$ in low-precision as much as possible without affecting accuracy or stability.

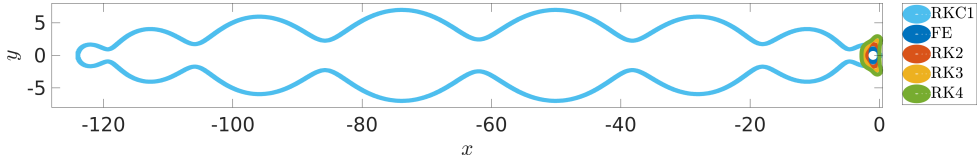
Note: in this part of the talk we only use RtN.

Absolute stability

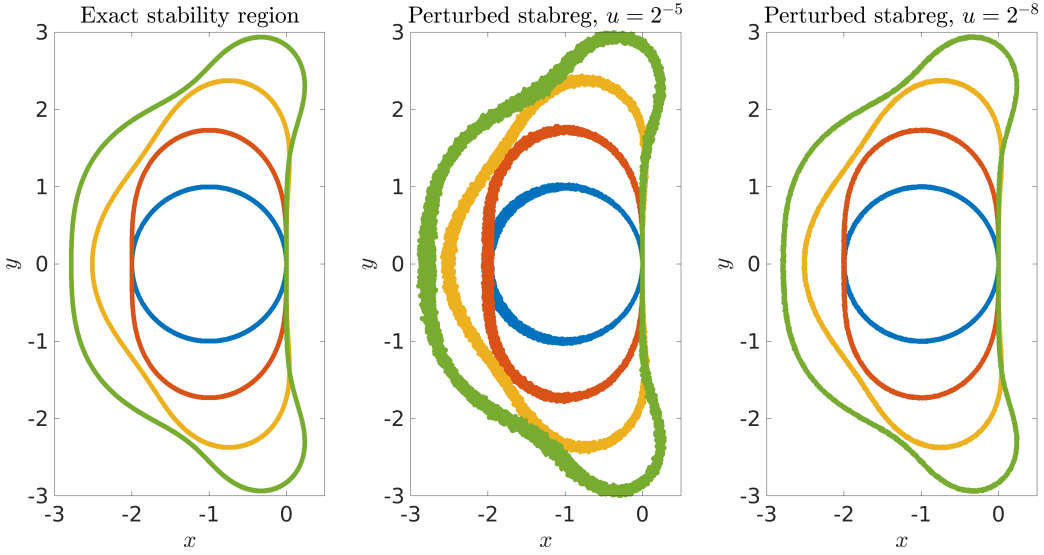
Dahlquist's test problem: $y' = \lambda y$, $y(0) = 1$.

s-stage RK method $y^n = R_s(z)^n$, where $z = \Delta t \lambda = x + iy$. Stable if $|R_s(z)| < 1$.

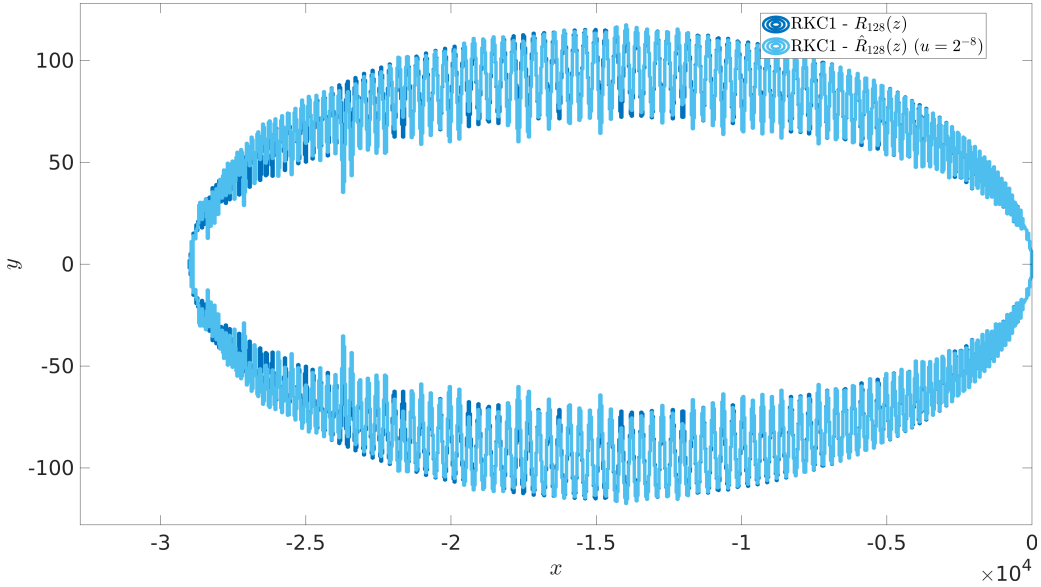




Linear stability for RK methods (in practice)



Linear stability for RKC (in practice, $s = 128$, $u = 2^{-8}$)



Order-preserving mixed-precision RK methods

Assumption

Operations performed in high-precision are exact.

Definition (Order-preserving mixed-precision RK method)

A p-th order mixed-precision RK method is q-order-preserving ($q \in \{1, \dots, p\}$), if it converges with order q under the above assumption.

We saw that RP methods do not converge, hence they are not order-preserving.

Order-preserving mixed-precision RK methods

Assumption

Operations performed in high-precision are exact.

Definition (Order-preserving mixed-precision RK method)

A p-th order mixed-precision RK method is q-order-preserving ($q \in \{1, \dots, p\}$), if it converges with order q under the above assumption.

We saw that RP methods do not converge, hence they are not order-preserving.

Our idea: store solution in high precision and use only q high-precision function evaluations to obtain a q-order-preserving mixed-precision RK method.

We can construct q-order preserving RK methods for any q for linear problems, and for $q = 1, 2$ for nonlinear problems. We can prove both stability and convergence.

Order-preserving mixed-precision RK methods

Assumption

Operations performed in high-precision are exact.

Definition (Order-preserving mixed-precision RK method)

A p-th order mixed-precision RK method is q-order-preserving ($q \in \{1, \dots, p\}$), if it converges with order q under the above assumption.

We saw that RP methods do not converge, hence they are not order-preserving.

Our idea: store solution in high precision and use only q high-precision function evaluations to obtain a q-order-preserving mixed-precision RK method.

We can construct q-order preserving RK methods for any q for linear problems, and for $q = 1, 2$ for nonlinear problems. We can prove both stability and convergence.

Note: We mainly focused on stabilized methods since they are low-order, but use a lot of function evaluations to maximize stability.

Linear problems, i.e. $\boldsymbol{f}(\boldsymbol{y}) = A\boldsymbol{y}$

Consider the exact solution at $t = \Delta t$ and its corresponding p-th order RK approximation:

$$\boldsymbol{y}(\Delta t) = \exp(\Delta t A)\boldsymbol{y}_0 = \sum_{j=0}^{\infty} \frac{(\Delta t A)^j}{j!}\boldsymbol{y}_0,$$

$$\boldsymbol{y}_1 = \sum_{j=0}^{p} \frac{(\Delta t A)^j}{j!}\boldsymbol{y}_0 + O(\Delta t^{p+1}).$$

Giving a local error of $\tau = \Delta t^{-1}||\boldsymbol{y}(\Delta t) - \boldsymbol{y}_1|| = O(\Delta t^p)$.

Linear problems, i.e. $\boldsymbol{f}(\boldsymbol{y}) = A\boldsymbol{y}$

Consider the exact solution at $t = \Delta t$ and its corresponding p-th order RK approximation:

$$\boldsymbol{y}(\Delta t) = \exp(\Delta t A)\boldsymbol{y}_0 = \sum_{j=0}^{\infty} \frac{(\Delta t A)^j}{j!}\boldsymbol{y}_0,$$

$$\boldsymbol{y}_1 = \sum_{j=0}^{p} \frac{(\Delta t A)^j}{j!}\boldsymbol{y}_0 + O(\Delta t^{p+1}).$$

Giving a local error of $\tau = \Delta t^{-1}||\boldsymbol{y}(\Delta t) - \boldsymbol{y}_1|| = O(\Delta t^p)$.

Evaluating the scheme in finite precision yields:

$$\hat{\boldsymbol{y}}_1 = \varepsilon + \boldsymbol{y}_0 + \sum_{j=1}^{p} \frac{\Delta t^j}{j!} \left(\prod_{k=1}^{j} (A + \Delta A_k) \right) \boldsymbol{y}_0 + O(\Delta t^{p+1}).$$

Linear problems

$$\tau = \Delta^{-1}||\hat{\boldsymbol{y}}_1 - \boldsymbol{y}_1|| = \Delta t^{-1} \left|\left| \varepsilon + \sum_{j=1}^{p} \frac{\Delta t^j}{j!} \left(\prod_{k=1}^{j} (A + \Delta A_k) - A^j \right) \boldsymbol{y}_0 \right|\right| + O(\Delta t^p).$$

Linear problems

$$\tau = \Delta^{-1}||\hat{\boldsymbol{y}}_1 - \boldsymbol{y}_1|| = \Delta t^{-1} \left|\left| \varepsilon + \sum_{j=1}^{p} \frac{\Delta t^j}{j!} \left(\prod_{k=1}^{j} (A + \Delta A_k) - A^j \right) \boldsymbol{y}_0 \right|\right| + O(\Delta t^p).$$

Let us consider the following scenarios:

1. We have $\varepsilon = O(u)$ and we get $\tau = O(u\Delta t^{-1} + \Delta t^p)$. Rapid error growth!

Linear problems

$$\tau = \Delta^{-1}||\hat{\boldsymbol{y}}_1 - \boldsymbol{y}_1|| = \Delta t^{-1} \left|\left| \varepsilon + \sum_{j=1}^{p} \frac{\Delta t^j}{j!} \left(\prod_{k=1}^{j} (A + \Delta A_k) - A^j \right) \boldsymbol{y}_0 \right|\right| + O(\Delta t^p).$$

Let us consider the following scenarios:

1. We have $\varepsilon = O(u)$ and we get $\tau = O(u\Delta t^{-1} + \Delta t^p)$. Rapid error growth!
2. Exact vector operations: $\varepsilon = 0$ so $\tau = O(u + \Delta t^p)$. $O(u)$ limiting accuracy and loss of convergence.

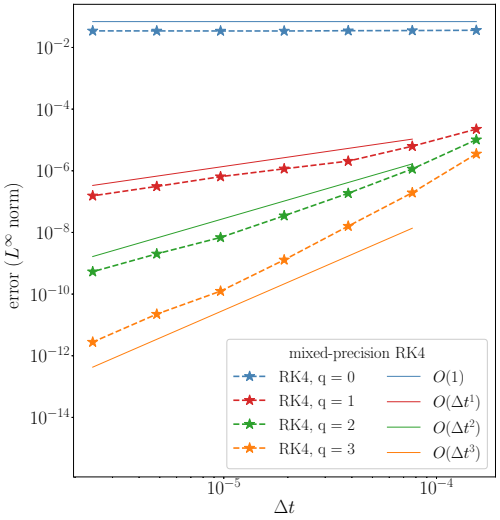
Linear problems

$$\tau = \Delta^{-1}||\hat{\boldsymbol{y}}_1 - \boldsymbol{y}_1|| = \Delta t^{-1} \left|\left| \varepsilon + \sum_{j=1}^{p} \frac{\Delta t^j}{j!} \left(\prod_{k=1}^{j} (A + \Delta A_k) - A^j \right) \boldsymbol{y}_0 \right|\right| + O(\Delta t^p).$$

Let us consider the following scenarios:

1. We have $\varepsilon = O(u)$ and we get $\tau = O(u\Delta t^{-1} + \Delta t^p)$. Rapid error growth!
2. Exact vector operations: $\varepsilon = 0$ so $\tau = O(u + \Delta t^p)$. $O(u)$ limiting accuracy and loss of convergence.
3. First $q \geq 1$ matvecs exact. Now $\varepsilon = 0$ and $\Delta A_k = 0$ for $k = 1, \ldots, q$, so $\tau = O(u\Delta t^q + \Delta t^p)$. Recover q-th order convergence!

Numerical results - convergence (3D heat eqn)

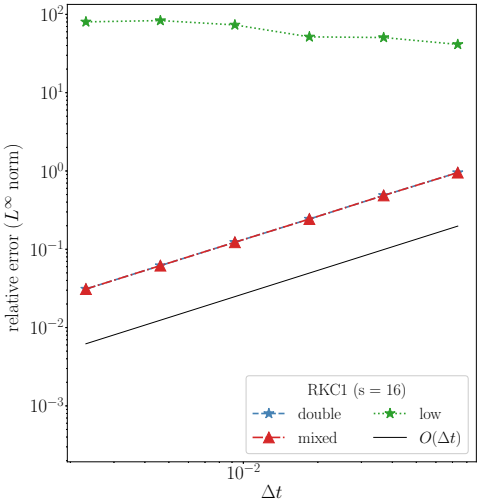


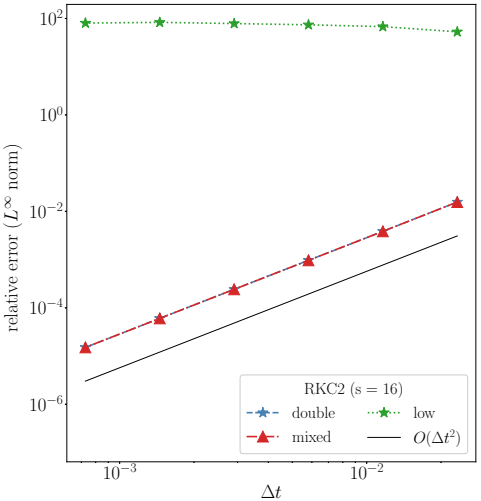
The transition from order p to order q happens roughly when $\Delta t = O(||A||^{-1} u^{\frac{1}{p-q}})$

Numerical results - convergence

1D Brussellator model for chemical autocatalytic reactions (with Dirichlet BCs):

$$\begin{cases} \dot{\mathrm{u}} = \alpha \Delta \mathrm{u} + \mathrm{u}^2 \mathrm{v} - (b+1)\,\mathrm{u} + a \\ \dot{\mathrm{v}} = \alpha \Delta \mathrm{v} - \mathrm{u}^2 \mathrm{v} + b\,\mathrm{u} \end{cases}$$

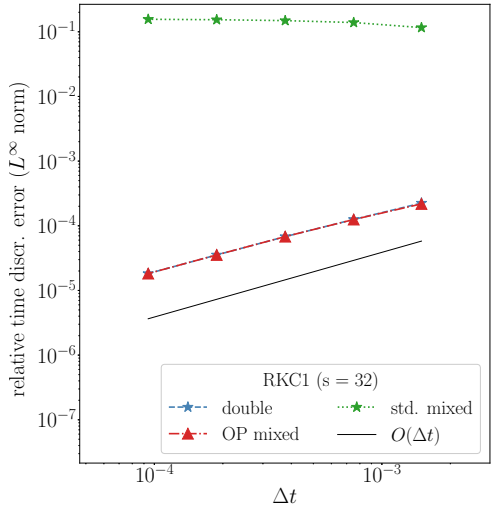


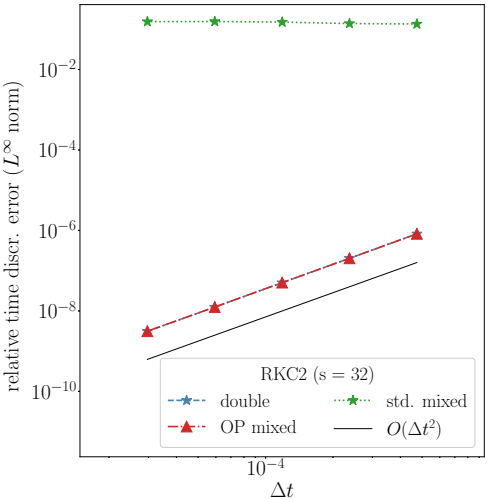


Numerical results - convergence

Nonlinear diffusion model, 1D 4-Laplace diffusion operator (with Dirichlet BCs):

$$\dot{u} = \nabla \cdot (\|\nabla u\|_2^2 \nabla u) + f$$





4. Conclusions

Outlook

To sum up

- Reduced-/mixed-precision algorithms require a careful implementation, but can bring significant memory, cost, and energy savings.
- Many new reduced and mixed-precision algorithms for scientific computing and data science were developed in recent years. Hardware support is growing.
- **Advice for new developers:** find which operations are more costly or more sensitive to rounding errors before designing a mixed-precision method.
- **Advice for new practitioners:** keep GPU and FPGA applications in mind as that's where most savings can currently be obtained.

Outlook

To sum up

- Reduced-/mixed-precision algorithms require a careful implementation, but can bring significant memory, cost, and energy savings.
- Many new reduced and mixed-precision algorithms for scientific computing and data science were developed in recent years. Hardware support is growing.
- **Advice for new developers:** find which operations are more costly or more sensitive to rounding errors before designing a mixed-precision method.
- **Advice for new practitioners:** keep GPU and FPGA applications in mind as that's where most savings can currently be obtained.

Thank you for listening!

Papers, slides, and more info at: `https://croci.github.io`
Email: `matteo.croci@austin.utexas.edu`

References I

[1] A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, et al. A survey of numerical linear algebra methods utilizing mixed-precision arithmetic. *The International Journal of High Performance Computing Applications*, 35(4):344–369, 2021.

[2] N. J. Higham and T. Mary. Mixed precision algorithms in numerical linear algebra. *Acta Numerica*, 31: 347–414, 2022.

[3] M. Croci, M. Fasi, N. J. Higham, T. Mary, and M. Mikaitis. Stochastic rounding: implementation, error analysis and applications. *Royal Society Open Science*, 9:211631, 2022.

[4] M. P. Connolly, N. J. Higham, and T. Mary. Stochastic rounding and its probabilistic backward error analysis. *SIAM Journal on Scientific Computing*, 43(1):566–585, 2021.

[5] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002.

[6] N. Mellempudi, S. Srinivasan, D. Das, and B. Kaul. Mixed precision training with 8-bit floating point. *arXiv preprint arXiv:1905.12334*, 2019.

[7] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Fifteenth annual conference of the international speech communication association*. Microsoft, 2014.

[8] Y. Xie, R. H. Byrd, and J. Nocedal. Analysis of the BFGS method with errors. *SIAM Journal on Optimization*, 30(1):182–209, 2020.

[9] F. Tisseur. Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 22(4):1038–1057, 2001.

[10] C. Kelley. Newton's method in mixed precision. *SIAM Review*, 64(1):191–211, 2022.

References II

[11] M. Klöwer, S. Hatfield, M. Croci, P. D. Düben, and T. N. Palmer. Fluid simulations accelerated with 16 bits: Approaching 4x speedup on A64FX by squeezing ShallowWaters.jl into Float16. *Journal of Advances in Modeling Earth Systems*, 2021.

[12] M. Croci and M. B. Giles. Effects of round-to-nearest and stochastic rounding in the numerical solution of the heat equation in low precision. *IMA Journal of Numerical Analysis*, 2022. URL `https://doi.org/10.1093/imanum/drac012`.

[13] M. Croci and G. R. de Souza. Mixed-precision explicit stabilized Runge–Kutta methods for single-and multi-scale differential equations. *Journal of Computational Physics*, 2022.

[14] E. Carson and N. J. Higham. Accelerating the solution of linear systems by iterative refinement in three precisions. *SIAM Journal on Scientific Computing*, 40(2):A817–A847, 2018.

[15] E. Carson and N. J. Higham. A new analysis of iterative refinement and its application to accurate solution of ill-conditioned sparse linear systems. *SIAM Journal on Scientific Computing*, 39(6):A2834–A2856, 2017.

[16] J. Langou, J. Langou, P. Luszczek, J. Kurzak, A. Buttari, and J. Dongarra. Exploiting the performance of 32 bit floating point arithmetic in obtaining 64 bit accuracy (revisiting iterative refinement for linear systems). In *SC'06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, pages 50–50. IEEE, 2006.

[17] E. Carson, N. J. Higham, and S. Pranesh. Three-precision GMRES-based iterative refinement for least squares problems. *SIAM Journal on Scientific Computing*, 42(6):A4063–A4083, 2020.

[18] P. Amestoy, A. Buttari, N. J. Higham, J.-Y. l'Excellent, T. Mary, and B. Vieuble. Combining sparse approximate factorizations with mixed precision iterative refinement. 2022. URL `eprints.maths.manchester.ac.uk/2845/`.

References III

[19] R. Tamstorf, J. Benzaken, and S. F. McCormick. Discretization-error-accurate mixed-precision multigrid solvers. *SIAM Journal on Scientific Computing*, 43(5):S420–S447, 2021.

[20] S. F. McCormick, J. Benzaken, and R. Tamstorf. Algebraic error analysis for mixed-precision multigrid solvers. *SIAM Journal on Scientific Computing*, 43(5):S392–S419, 2021.

[21] H. Anzt, V. Heuveline, and B. Rocker. Mixed precision iterative refinement methods for linear systems: Convergence analysis based on Krylov subspace methods. In *International Workshop on Applied Parallel Computing*, pages 237–247. Springer, 2010.

[22] N. Lindquist, P. Luszczek, and J. Dongarra. Accelerating restarted GMRES with mixed precision arithmetic. *IEEE Transactions on Parallel and Distributed Systems*, 33(4):1027–1037, 2021.

[23] G. Meurant and Z. Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

[24] S. Gratton, E. Simon, D. Titley-Peloquin, and P. Toint. Exploiting variable precision in GMRES. *arXiv preprint arXiv:1907.10550*, 2019.

[25] M. Arioli and I. S. Duff. Using FGMRES to obtain backward stability in mixed precision. *Electronic Transactions on Numerical Analysis*, 33:31–44, 2009.

[26] J. G. Verwer, W. H. Hundsdorfer, and B. P. Sommeijer. Convergence properties of the Runge-Kutta-Chebyshev method. *Numerische Mathematik*, 57:157–178, 1990.