Enabling mixed-precision with VerifiCarlo: Sharing CEEC experience

Roman Iakymchuk[1]
jointly with Yanxiang Chen (UmU), and
Pablo de Oliveira Castro (Paris-Saclay, UVSQ)

Umeå University and Uppsala University, Sweden
`riakymch@cs.umu.se`

CEEC Webinar
May 27, 2026

Centre of Excellence in Exascale CFD

[1]This research is partially funded by EuroHPC JU CoE CEEC (No. 101093393)

Outline

Outline

- Overview of Center of Excellence in Exascale CFD
- Methodology for energy-efficient computing
- Mixed-precision Nekbone and Neko

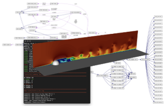
EuroHPC JU CoE CEEC

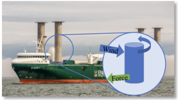
Center of Excellence in Exascale CFD

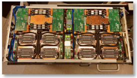
Overview

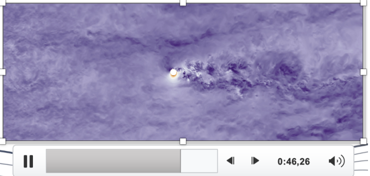
The main goal of CEEC is to address the extreme-scale computing challenge to enable the use of accurate and cost-efficient high fidelity computational fluid dynamics (CFD) simulations at exascale

- Implement **exascale-ready workflows** for addressing grand challenge scientific problems
- Develop **new or improved algorithms** that can efficiently exploit exascale systems.
- Significantly improve **energy efficiency** of simulations
- Demonstrate workflows on **lighthouse cases** relevant for both academia and industry









Courtesy of Niclas Janson, KTH

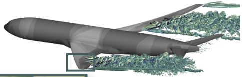
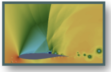
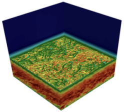
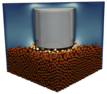
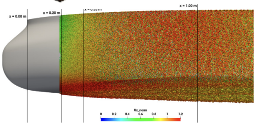
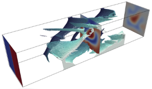
EuroHPC JU CoE CEEC

Center of Excellence in Exascale CFD

Lighthouse Cases

CEEC

- **Shock-boundary layer interaction and buffet on wings at the edge of the flight envelope**
  - Codes used: FLEXI
- **High fidelity aeroelastic simulation of the SFB 401 wing in flight conditions**
  - Codes used: Alya
- **Topology optimisation of static mixers**
  - Codes used: Neko
- **Localized erosion of an offshore wind-turbine foundations**
  - Codes used: waLBerla
- **Simulation of Atmospheric Boundary Layer flows**
  - Codes used: NekRS/Nek5000
- **Merchant ship hull**
  - Codes used: Neko

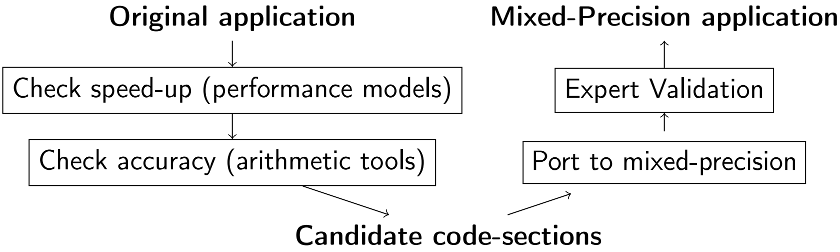
CEEC us out on `https://ceec-coe.eu`

CEEC

Outline

Methodology

Methodology[2] to enable mixed-precision algorithmic solutions in applications with accuracy guarantees.



[2] Chen et al. 'Enabling mixed-precision in spectral element codes'. Future Generation Computer Systems 174, 107990, 2026

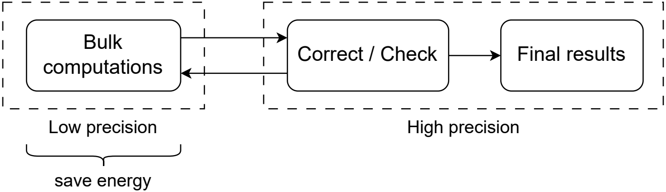
From mixed-precision to energy efficiency

Precision impact on energy

- **Higher precision** → more accurate, but more energy-hungry
- **Lower precision** → less accurate, but:
  - more FLOPs per Watt
  - smaller memory footprint, less data transfer

Mixed-precision

- Bulk operations in low precision → cheaper & lower memory traffic
- Critical operations in high precision → maintain accuracy/stability



Outline

Test case

- **Nek5000/ Neko** are the high order, incompressible Navier-Stokes solvers based on the spectral element method

$\rightarrow$ pressure solver using Jacobi preconditioned Conjugate Gradient

$\rightarrow$ matrix-free

- **Nekbone** is a mini-app of Nek5000 and it solves a Poisson equation using the Conjugate Gradient method with the spectral element multigrid preconditioner

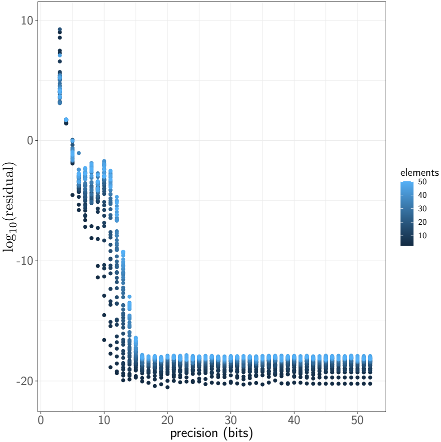
Nekbone w Vprec

$$\boxed{Ax = b}$$

while $(\tau > \tau_{\max})$

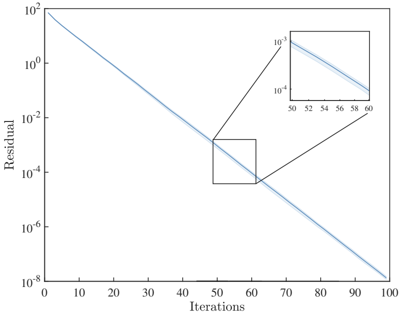
Step	Operation
$S1:$	$w := Ad$
$S2:$	$\rho := \beta / < d, w >$
$S3:$	$x := x + \rho d$
$S4:$	$r := r - \rho w$
$S5:$	$z := M^{-1} r$
$S6:$	$\beta := < z, r >$
$S7:$	$d := (\beta/\beta_{old}) d + z$
$S8:$	$\tau := < r, r >$

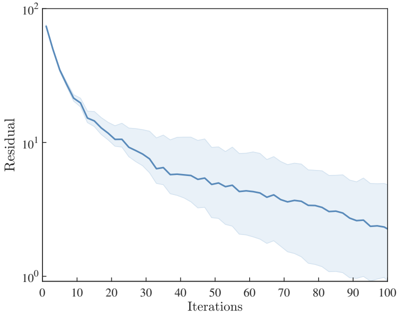
end while



Nekbone w MCA for FP32

Entire program, no preconditioner

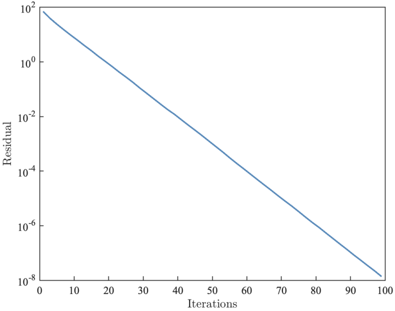


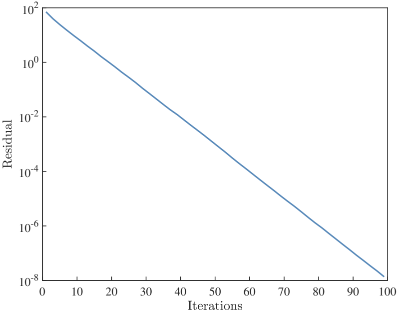


- Random Rounding (rr) mode (left)
- MCA (mca) mode (right)
- Issue in initialization $10^9 \cos(x) \rightarrow$ focus on the solver only

Nekbone w MCA for FP32

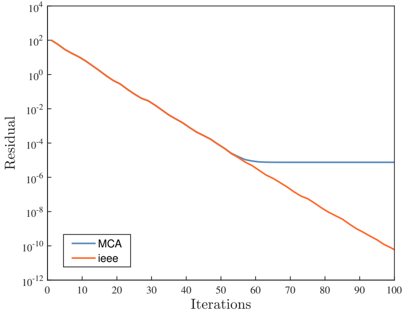
Only the CG loop, no preconditioner

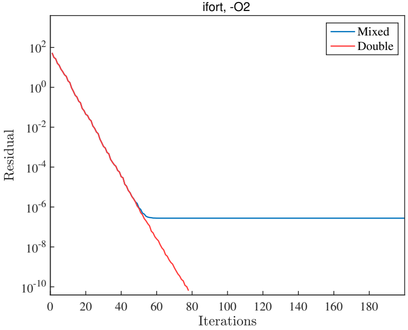




- Random Rounding (rr) mode (left)
- MCA (mca) mode (right)

Nekbone: CG with preconditioner





- Verificarlo predicts stagnation (left plot)
- Stagnation in the mixed-precision Nekbone (right plot)
- Stagnation occurs after 61st iteration w residual $r = 7.94 \times 10^{-6}$

Nekbone: CG with preconditioner

Mixed-precision strategies

Mixed-precision strategy evaluation

GSO mode	MPI	Precond. Ops	Precond. GSO	PCG Ops	PCG GSO	Conv.
pairwise	< 4	fp32	fp32	fp32	fp32	stagnates
		fp64	fp64			converges
		fp32+fp64sqrt	fp32			converges
		fp32	fp64			converges
	≥ 4	fp64	fp64		fp32	stagnates
		fp32+fp64sqrt	fp32			stagnates
		fp32	fp64			stagnates
		fp32+fp64sqrt	fp64		fp64	converges
		fp64	fp64		fp64	converges

- Gather-Scatter Operations have strong impact on convergence
- dot product impacts convergence too

Mixed-precision Nekbone: time-to-solution

CG with multigrid preconditioner

MPI ranks	8	20	40	80
Double-1[a]	5.79	8.98	13.33	24.02
Double-2	7.55	8.62	10.84	13.71
Mixed-1[b]	4.79	5.99	9.42	14.85
Gain-1	1.21x	1.50x	1.42x	1.62x
Mixed-2[c]	4.88	6.02	8.37	10.11
Gain-2	1.55x	1.43x	1.3x	1.36x

[a] Double-1: allreduce, Double-2: pairwise

[b] fp64+fp32+fp64sqrt (fp32 GS, allreduce)

[c] fp64+fp32+fp64sqrt (fp64 GS, pairwise)

- 128 elements per MPI rank
- On MN5: 2x Intel Sapphire Rapids 8460Y+ @2.3Ghz

CEEC

Mixed-precision Nekbone: energy-to-solution

CG with multigrid preconditioner

MPI ranks	8	20	40	80
Double-1[a]	1960	2865	7147	18 014
Double-2	1877	2150	4457	4920
Mixed-1[b]	939	2088	3133	7428
Gain-1	2.09x	1.37x	2.28x	2.43x
Mixed-2[c]	934	1566	2560	3367
Gain-2	2.01x	1.37x	1.74x	1.46x

[a] Double-1: allreduce, Double-2: pairwise

[b] fp64+fp32+fp64sqrt (fp32 GS, allreduce)

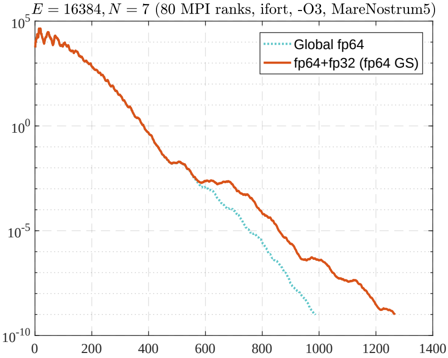
[c] fp64+fp32+fp64sqrt (fp64 GS, pairwise)

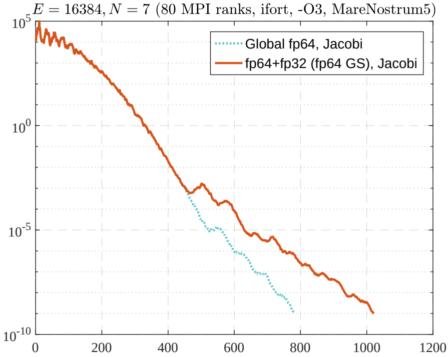
- 128 elements per MPI rank
- Energy measurements with EAR in joules
- On MN5: 2x Intel Sapphire Rapids 8460Y+ @2.3Ghz

CEEC

Neko: CG with preconditioners

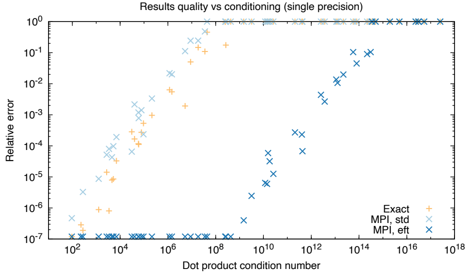
Convergence history

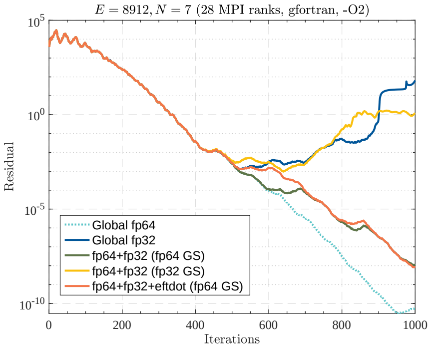




- Results from the real-world Neko code
- Comparable convergence for mixed-precision
- On MN5: 2x Intel Sapphire Rapids 8460Y+ @2.3Ghz

Dot product emulation via 2xfp32





- In CG, each dot product can be computed via dot2 with two fp32
  - $[c, e] =$ `twoprod`(a, b) $\{c = a * b, e = fma(a, b, -c)\}$ plus `twosum`
- Implementation from `github.com/ffevotte/libeft` 2018

Neko: time- and energy-to-solution

Identity preconditioner, -O3

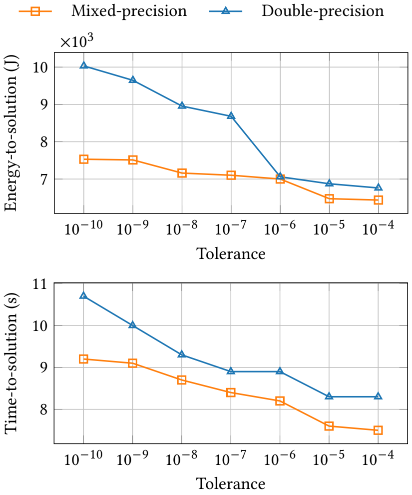
	Time (secs)	Energy (joules)
Double	2.01	10067
Mixed	1.71	8251
Gain	**1.18x**	**1.22x**

Jacobi preconditioner, -O3

	Time (secs)	Energy (joules)
Double	1.69	9634
Mixed	1.50	8313
Gain	**1.13x**	**1.16x**

- Energy measurements with EAR in joules
- On MN5: 2x Intel Sapphire Rapids 8460Y+ @2.3Ghz

Neko: Accuracy vs Energy vs Time



- Neko with the Jacobi preconditioned CG
- On MN5: 2x Intel Sapphire Rapids 8460Y+ @2.3Ghz

Summary

- Use computer arithmetic tools to
  - reduce precision, verify sensitivity, CI/CD
- Enabled mixed-precision in Nekbone and Neko:
  - use tools: gprof, Verificarlo, Intel Advisor
  - target the most time-consuming part
  - mixed-precision Nekbone: reduce time-to-solution by roughly 1.62x and energy-to-solution by 2.43x on MareNostrum 5
  - mixed-precision Neko: gain is up to 1.3x in both time and energy
- Ongoing work
  - porting to GPUs plus velocity solver with GMRES
  - adaptive mixed-precision

Summary

- Use computer arithmetic tools to
  - reduce precision, verify sensitivity, CI/CD
- Enabled mixed-precision in Nekbone and Neko:
  - use tools: gprof, Verificarlo, Intel Advisor
  - target the most time-consuming part
  - mixed-precision Nekbone: reduce time-to-solution by roughly 1.62x and energy-to-solution by 2.43x on MareNostrum 5
  - mixed-precision Neko: gain is up to 1.3x in both time and energy
- Ongoing work
  - porting to GPUs plus velocity solver with GMRES
  - adaptive mixed-precision

Thank you for your attention !

This research is partially supported by EuroHPC JU CoE CEEC (No. 101093393)