An Overview of the IDEAS-ECP Project

Minimization Strategies for Solving Eigenvalue Problems

Osni Marques

Lawrence Berkeley National Laboratory

An Overview of the IDEAS-ECP Project

EXASCALE COMPUTING PROJECT

IDEAS productivity

Osni Marques

Lawrence Berkeley National Laboratory

current team:

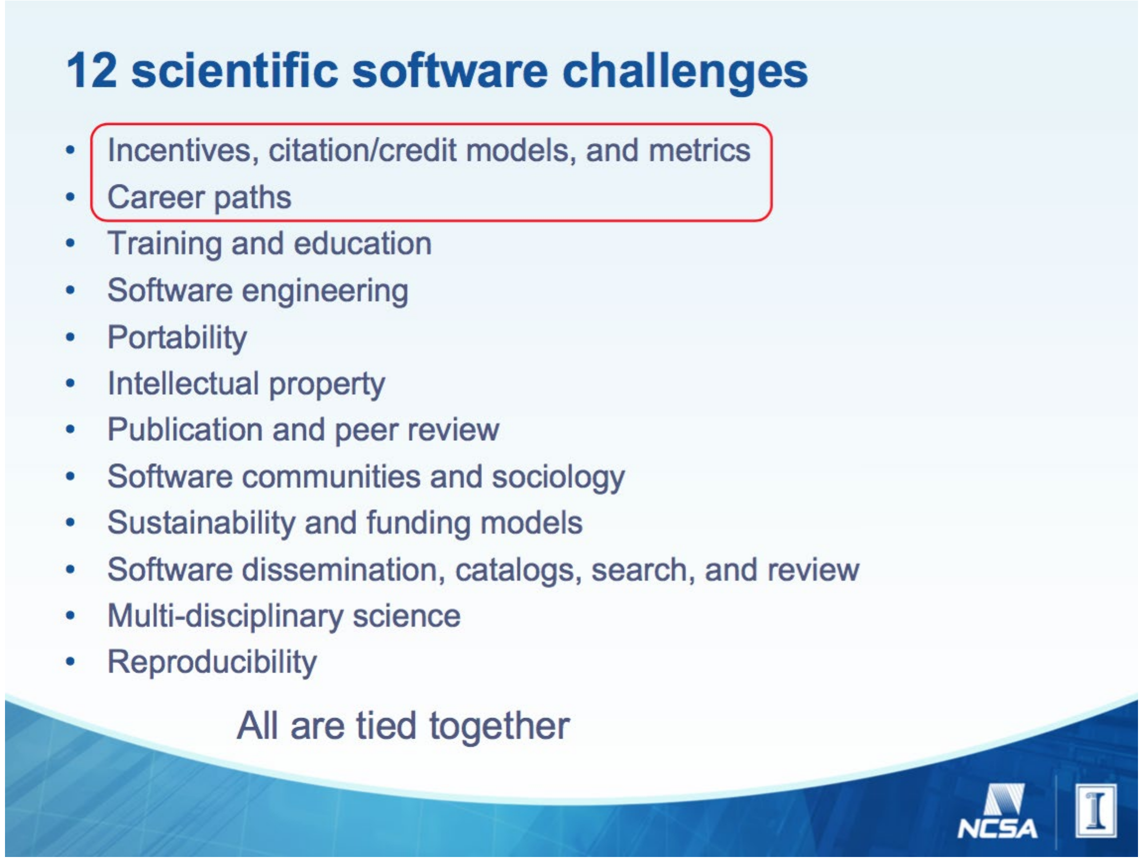
U.S. DEPARTMENT OF ENERGY | Office of Science

NNSA National Nuclear Security Administration

Motivation

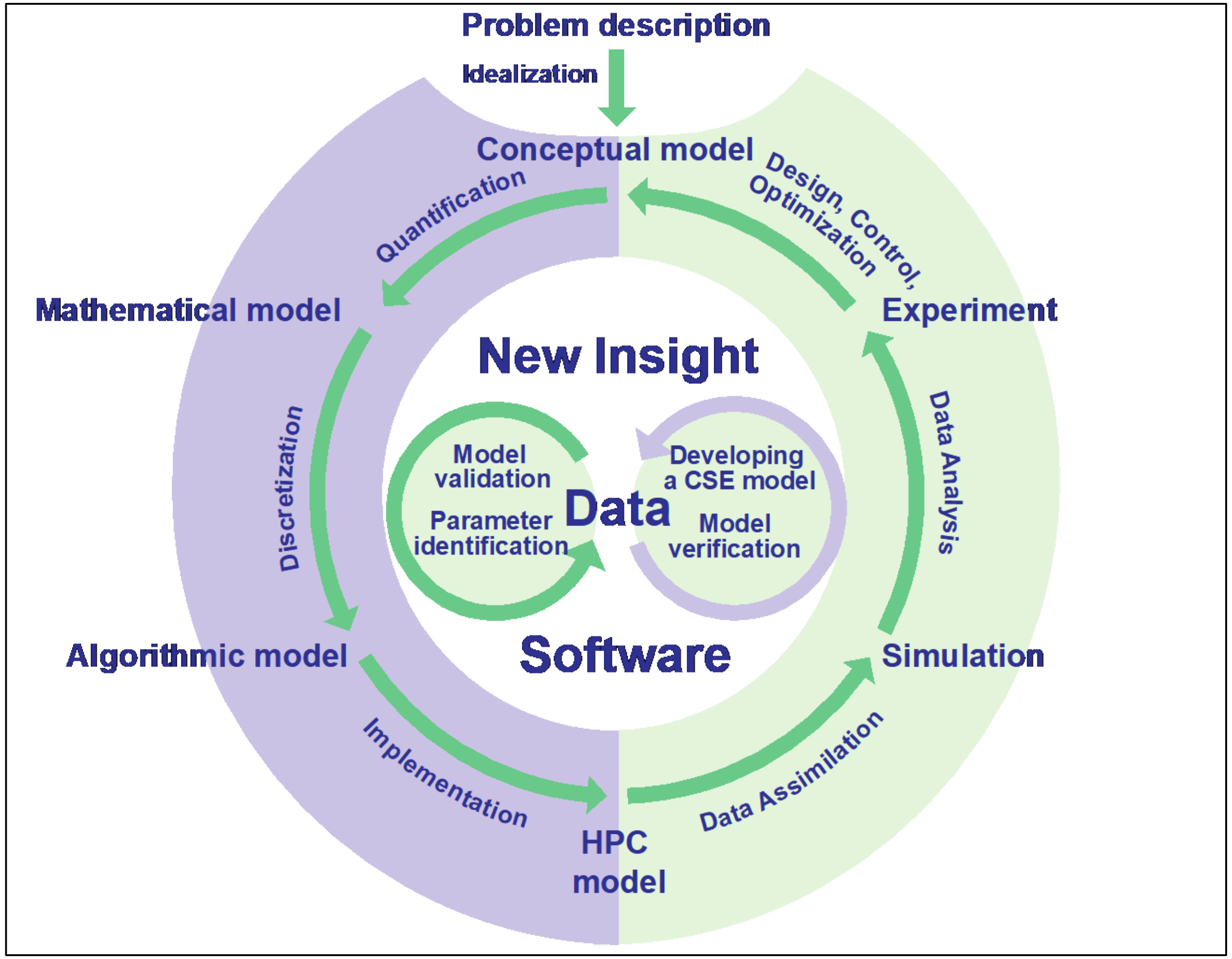
Challenges of High-Performance Scientific Software

- **Complex, intertwined challenges**
  - Disruptive changes in computer architectures
  - Increasing complexity of new scientific frontiers
  - Importance of reproducibility
  - Interdisciplinary, multi-institutional collaboration
  - Continually changing requirements
  - Competing priorities and incentives
  - Limited resources
- **Need community efforts**
  - Improve software quality and sustainability
  - Change research culture
  - Promote collaboration
  - etc.



Ref: Daniel Katz, Software in Research: Underappreciated and Underrewarded, 2017 eResearch Australasia conference, 2017, https://doi.org/10.6084/m9.figshare.5518933

Software is the foundation of sustained collaboration in HPC



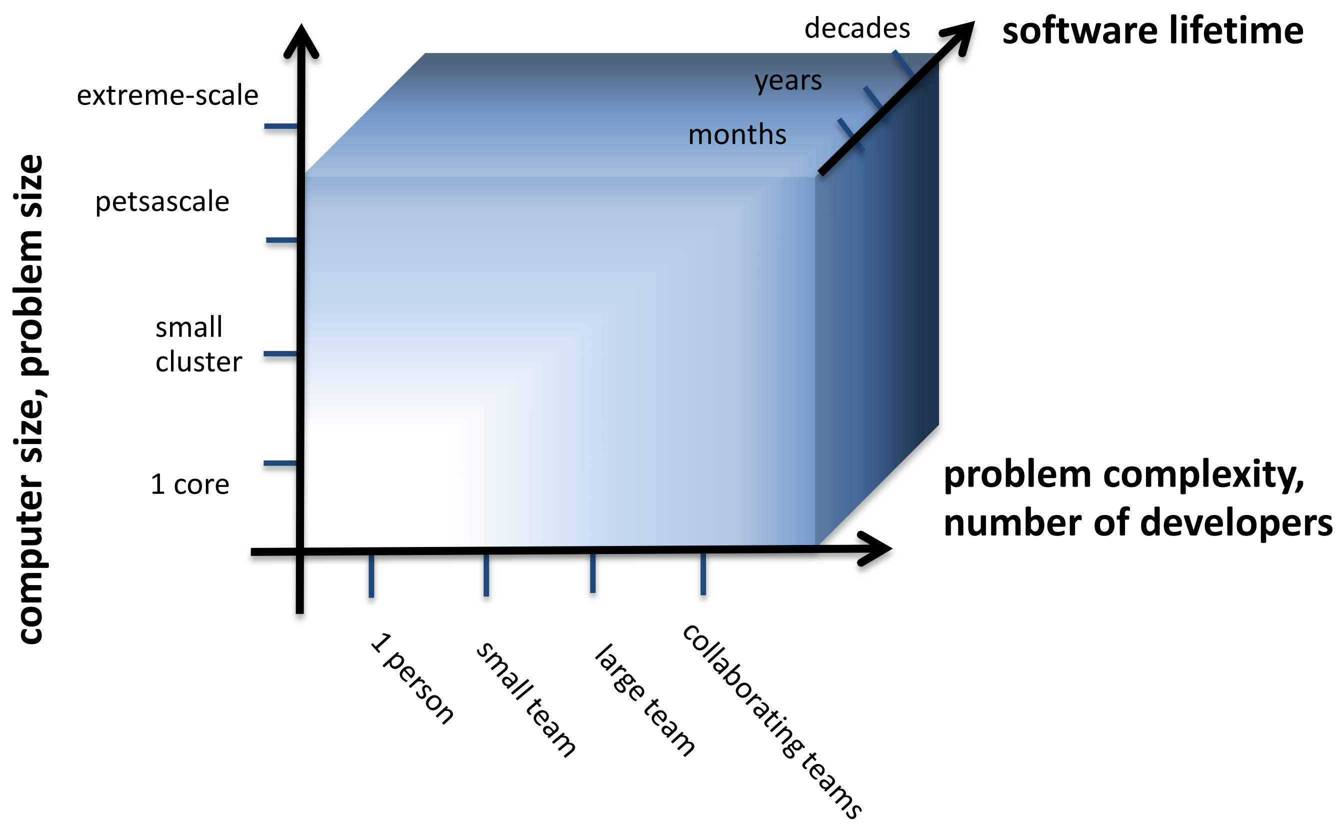
Research and Education in Computational Science and Engineering, U. Rüde, K. Willcox, L.C. McInnes, H. De Sterck, SIAM Review, 2018

computational science and engineering, data science, learning/AI, etc.

Emerging exascale architectures and systems will provide a sizable increase in raw computing power for science. To ensure the full potential of these new and diverse architectures, as well as the longevity and sustainability of science applications, we need to embrace software ecosystems as first-class citizens.

McInnes, L.C., Heroux, M.A., Draeger, E.W. *et al. How community software ecosystems can unlock the potential of exascale computing. Nat Comput Sci* **1,** 92–94 (2021). https://doi.org/10.1038/s43588-021-00033-y

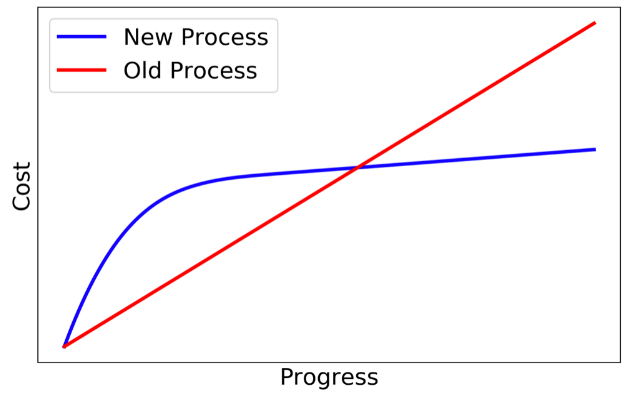
Community software ecosystems require high-quality software



Technical debt: The implied cost of additional rework caused by choosing an easy (limited) solution now instead of using a better approach that would take longer.

- Wikipedia

Improving developer productivity and software sustainability: nurturing a culture of continual improvement in software practices



IDEAS-ECP

Interoperable Design of Extreme-scale Application Software

Objectives and Major Areas of Work

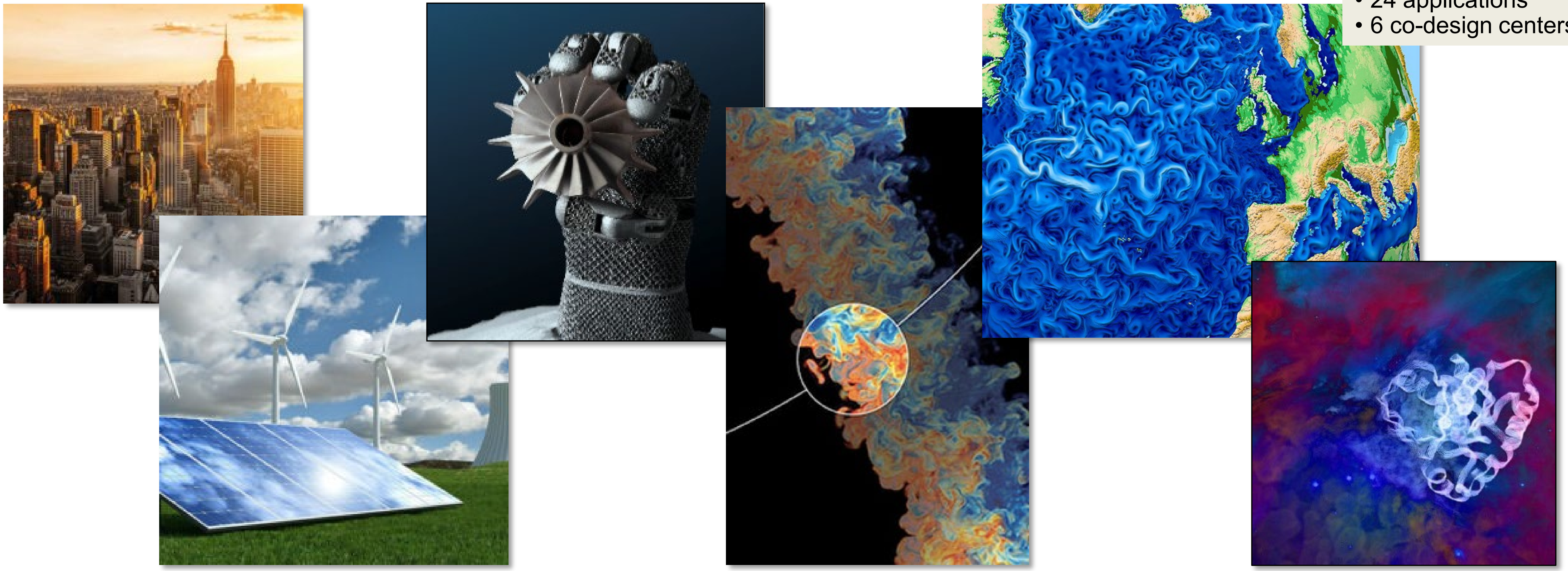
https://ideas-productivity.org

IDEAS-ECP: objectives

- *Address confluence of trends in hardware and increasing demands for predictive multiscale, multiphysics simulations.*
- *Respond to trend of continuous refactoring with efficient agile software engineering methodologies & improved software design.*
- *Promote software quality as a essential component for quality science.*
- *IDEAS began in 2014 as a DOE ASRC/BER partnership to improve application software productivity, quality, and sustainability.*
- *In 2017 the DOE Exascale Computing Project began supporting IDEAS to help application teams improve developer productivity and software sustainability while making major changes for exascale.*

Science and beyond: Applications and discovery in ECP

And more:
- 24 applications
- 6 co-design centers



National security	Energy security	Economic security	Scientific discovery	Earth systems	Health care

IDEAS-ECP team works with the ECP community to improve developer productivity and software sustainability as key aspects of increasing overall scientific productivity

1 Customize and curate methodologies

- Target scientific software productivity and sustainability
- Use workflow for best practices content development

2 Incrementally and iteratively improve software practices

- Determine high-priority topics for improvement and track progress
- *Productivity and Sustainability Improvement Planning (PSIP)*

3 Establish software communities

- Determine community policies to improve software quality and compatibility
- Create Software Development Kits (SDKs) to facilitate the combined use of complementary libraries and tools

4 Engage in community outreach

- Broad community partnerships
- Collaboration with computing facilities
- Webinars, tutorials, events
- *WhatIs* and *HowTo* docs
- Better Scientific Software site (https://bssw.io)

Productivity and Sustainability Improvement Planning (PSIP)

https://bssw.io/psip

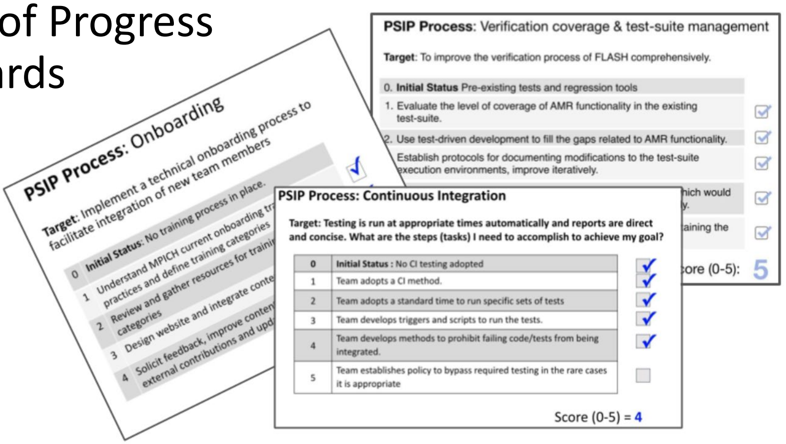
psip

A lightweight iterative workflow, where teams identify their most urgent software bottlenecks and track progress to overcome them.

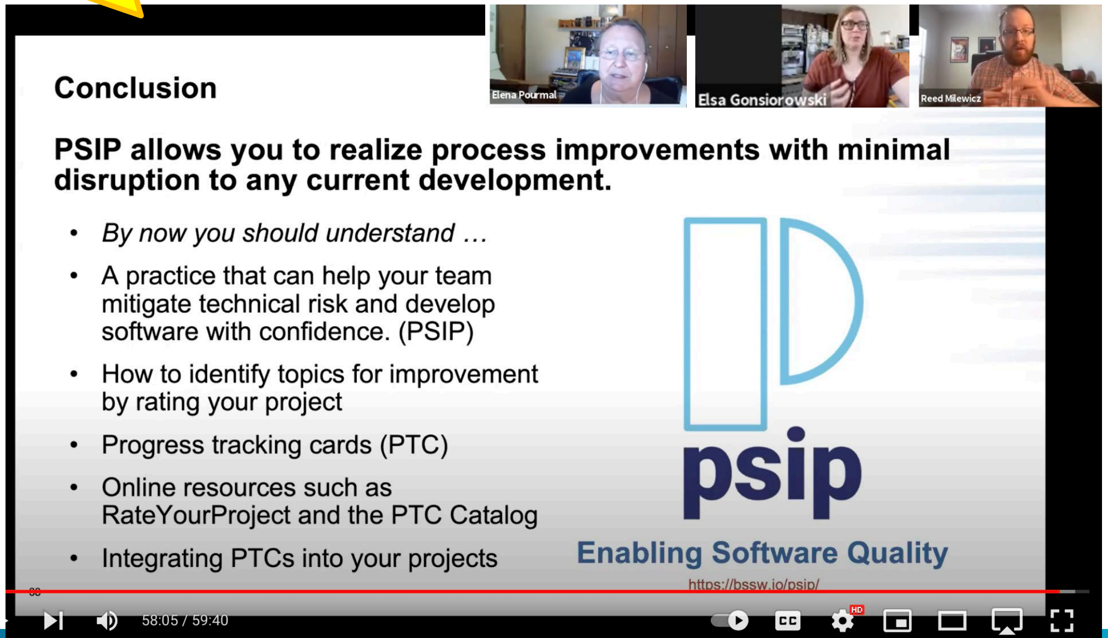
The PSIP workflow helps a team identify areas for improvement, select a specific area and topic for a single improvement cycle, and then develop those improvements with specific metrics for success.

Snapshots of Progress Tracking Cards

By EXAALT, ExaStar, MPICH



Using the PSIP Toolkit to Achieve Your Goals – A Case Study at The HDF Group, E. Pourmal, R. Milewicz, E. Gonsiorowski, webinar, June 2020



EXASCALE COMPUTING PROJECT

Outreach

Slides and videos available via
https://ideas-productivity.org/events

Webinar Series: Best Practices for HPC Software Developers

- *Wrong Way: Lessons Learned and Possibilities for Using the "Wrong" Programming Approach on Leadership Computing Facility Systems* (Jan 12, 2022)

  ⋮
- *Migrating to Heterogeneous Computing Lessons Learned in the Sierra and El Captan Centers of Excellence*
- *Software Design for Longevity with Performance Portability*
- *Testing & Code Review Practices in Research Software Development*
- *Accelerating Numerical Software Libraries with Multi-Precision Algorithms*
- *Discovering and Addressing Social Challenges in the Evolution of Scientific Software Projects*

  ⋮

and many more …

50+ webinars, 8800+ registrations in total. Average: 169 registrations (48 ECP-affiliated), 82 attendees

Tutorials

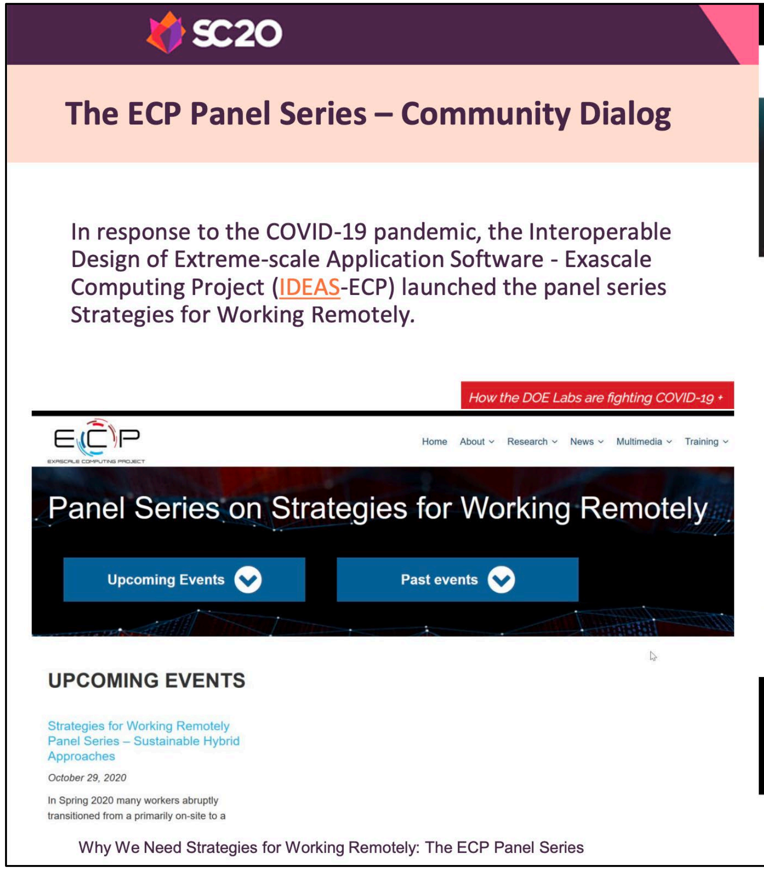
- ***Overview of Best Practices in HPC Software Development***
- ***Agile Project Management***
- ***Git Workflows***
- ***Software Design***
- ***Software Testing***
- ***Code Coverage and Continuous Integration***
- ***Software Refactoring***
- ***Continuous Integration***
- ***Reproducibility***
- ***An Introduction to Software Licensing***

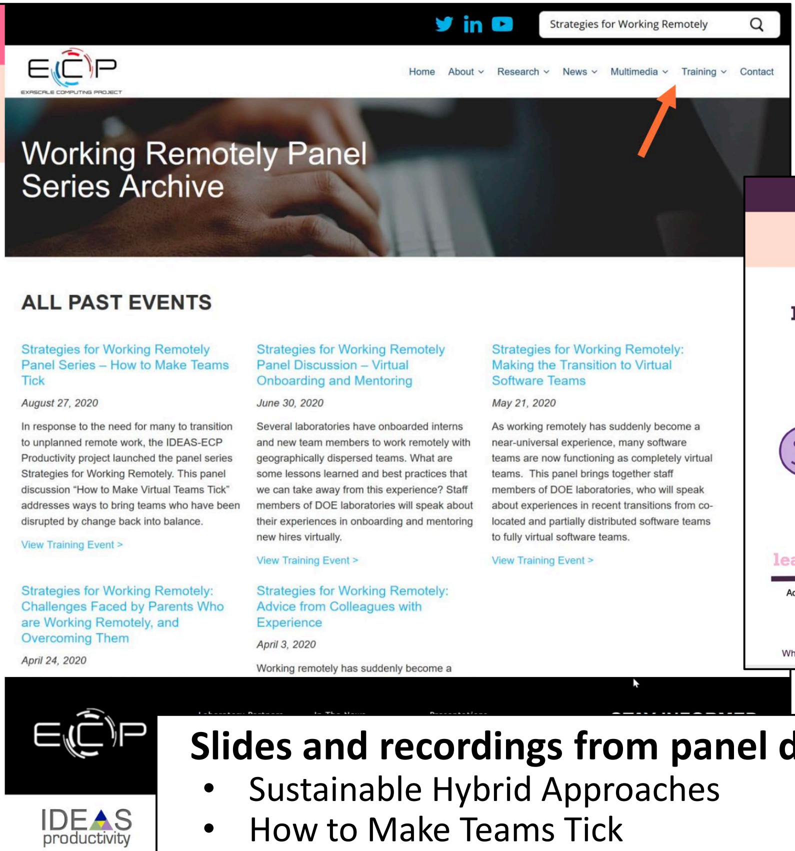
Panel Series

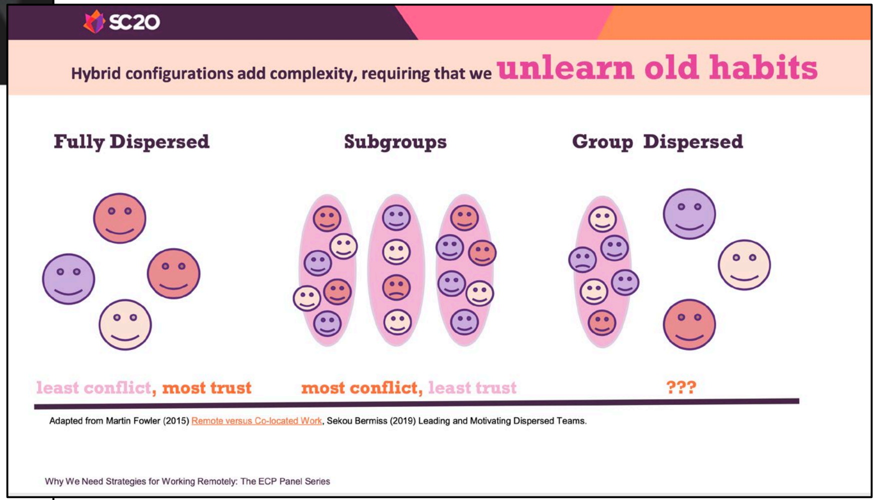
- ***Performance Portability & ECP***
- ***Strategies for Working Remotely***

Panel Series: Strategies for Working Remotely

https://www.exascaleproject.org/strategies-for-working-remotely







Slides and recordings from panel discussions:

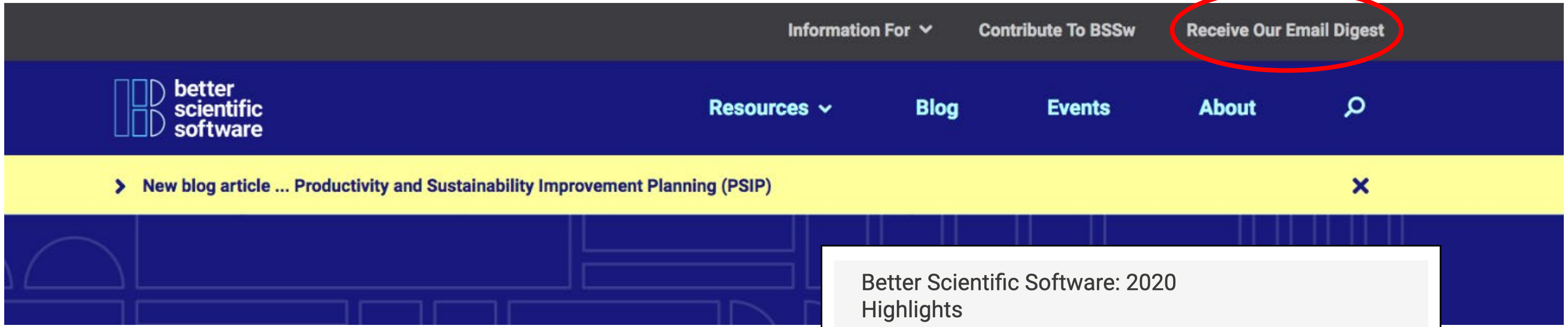
- Sustainable Hybrid Approaches
- How to Make Teams Tick
- Virtual Onboarding and Mentoring
- Making the Transition to Virtual Software Teams
- Challenges Faced by Parents Who Are Working Remotely, & Overcoming Them
- Advice from Colleagues with Experience

https://bssw.io/items/tips-for-producing-online-panel-discussions

bssw.io

better scientific software

So your code will see the future.



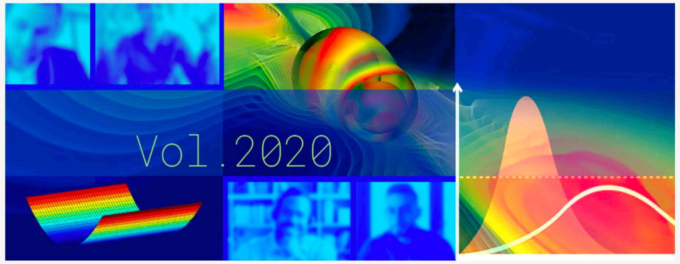
- **BSSw**: community-based hub for sharing information on practices, techniques, and tools to improve developer productivity and software sustainability for computational science.
- **We want and *need* contributions from the community!**
- **Types of content**
  - Informative articles
  - Curated links
  - Events
  - WhatIs, HowTo docs
  - Blog articles

Recent articles:

- Best Practices for HPC Software Developers: The First Five Years of the Webinar Series, O. Marques & D. Bernholdt
- The Contributions of Scientific Software to Scientific Discovery, K. Keahey & R. Gupta
- Effectively Integrating Interns into Research Teams, J. Lofstead
- Improving Team Practices with RateYourProject.org, G. Watson

Better Scientific Software: 2020 Highlights

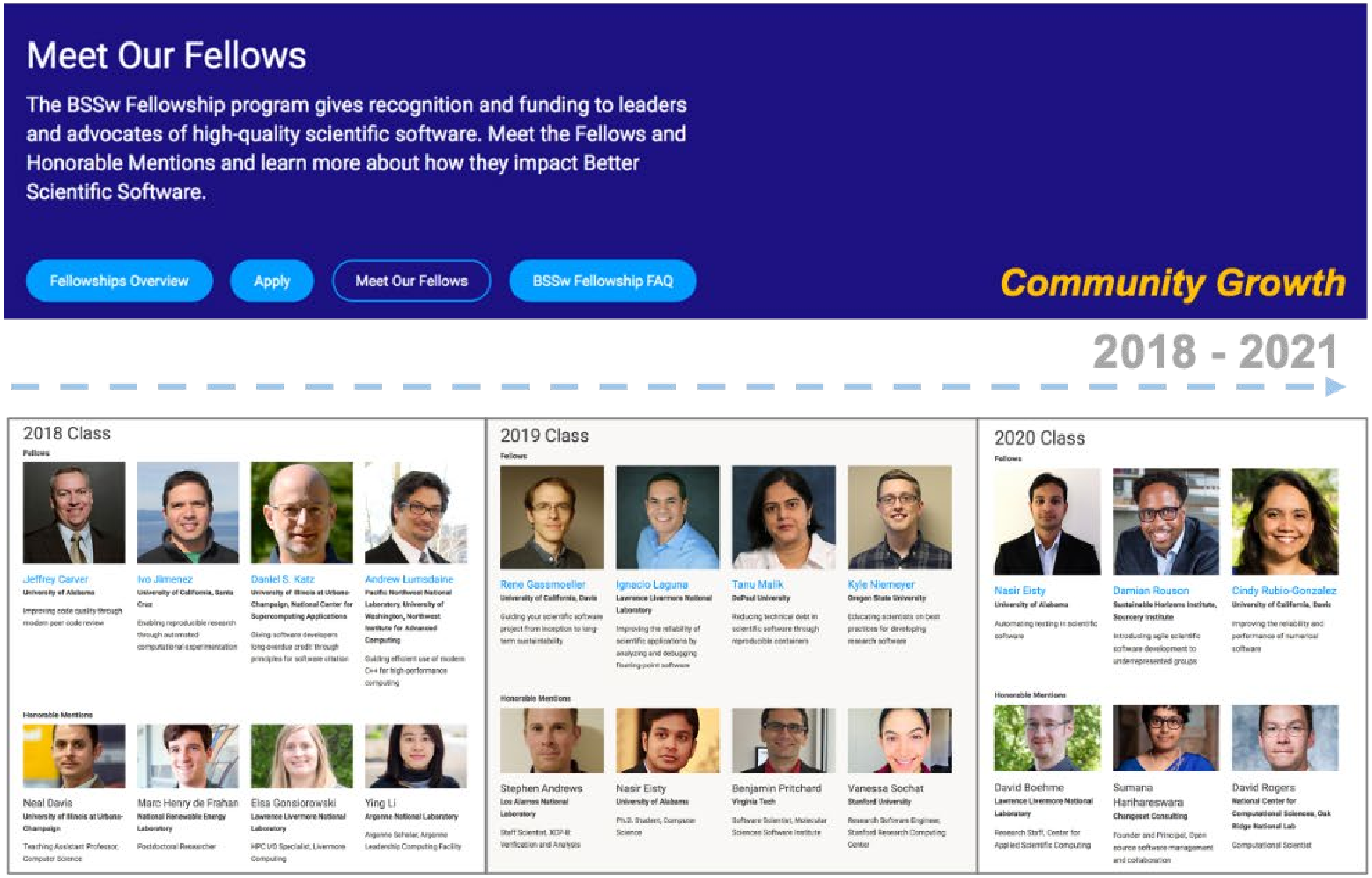
SHARE

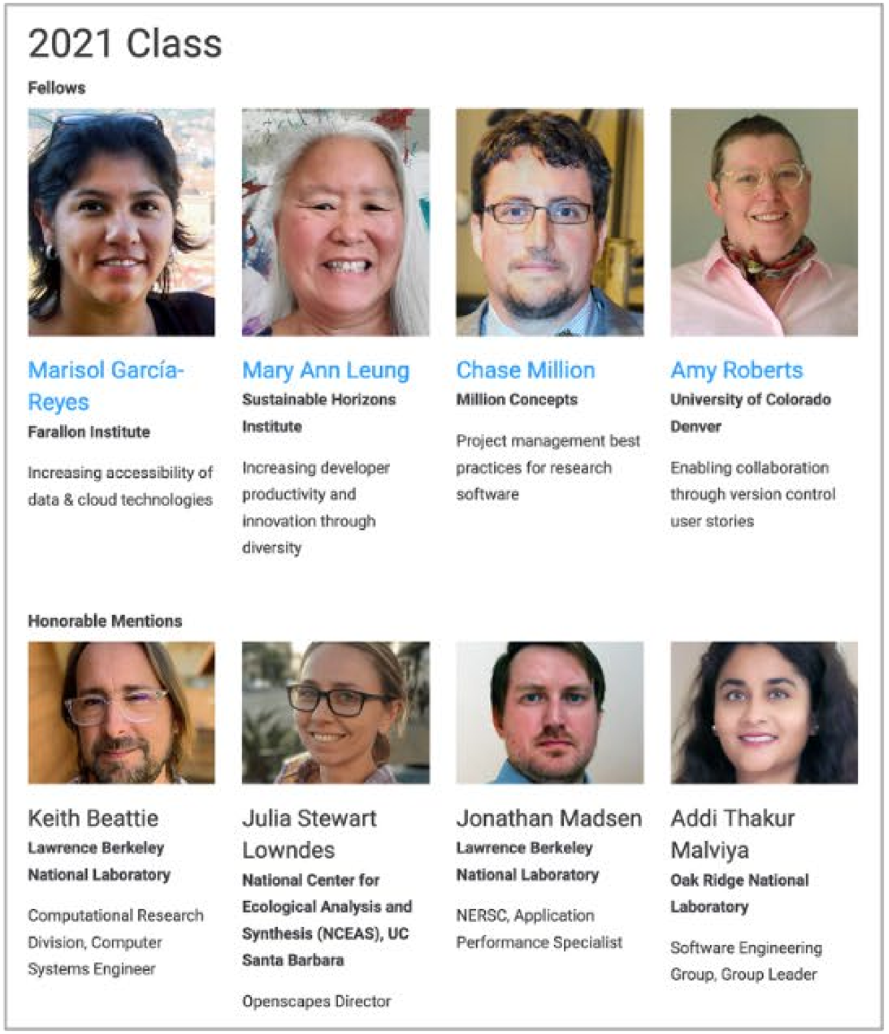


- Unit Testing C++ with Catch, M. Dewing
- The Art of Writing Scientific Software in an Academic Environment, H. Anzt
- FLASH5 Refactoring and PSIP, A. Dubey & J. O'Neal
- Software Sustainability in the Molecular Sciences, T. Windus & T.D. Crawford
- Working Effectively with Legacy Code, R. Bartlett
- Building Community through Software Policies, P. Luszczek & U.M. Yang
- Continuous Technology Refreshment: An Introduction Using Recent Tech Refresh Experiences on VisIt, M. Miller & H. Auten

EXASCALE COMPUTING PROJECT

BSSw Fellowship: Meet the Fellows





Final Remarks

Software quality is a critical component of quality science

- **Do you develop and use HPC software?**
  - Investigate resources for software improvement
  - Advocate for and lead change in your projects
  - Disseminate insights about software improvement from your own work (blogs, presentations, posters, papers, etc)
  - Check out community activities, such as the Research Software Engineering (RSE) movement
- **Do you lead projects or organizations where teams develop and use HPC software?**
  - Encourage continual software quality improvement
  - Provide clear career paths and mentoring for scientific software professionals, such as research software engineers
- **Are you a stakeholder or supporter of projects that develop and use HPC software?**
  - Incorporate expectations of software quality and sustainability, including funding for people to do this important work
  - Incorporate expectations for transparency and reproducibility
- **Everyone**
  - Work toward changes in software citations/credit models, metrics
  - Work toward changes in incentives, training and education

IDEAS mission: promote software quality as essential for quality science

- Evaluates and disseminates best practices and methodologies to improve developer productivity, software sustainability, and scientific reproducibility
- Help to improve developer productivity and software sustainability
  - Reduce technical risk by building a firmer foundation for computational science
  - Change requires investment but pays off over time
- Help ECP teams to achieve:
  - Better: Science, portability, robustness, composability
  - Faster: Execution, development, dissemination
  - Cheaper: Fewer staff hours and lines of code

- Advancing Scientific Productivity through Better Scientific Software: Developer Productivity & Software Sustainability Report
  - Explains the IDEAS approach, outcomes, and impact of work (in partnership with the ECP and broader computational science community).
  - Target those who care about the quality and integrity of scientific discoveries based on simulation and analysis.


EXASCALE COMPUTING PROJECT

ECP-U-RPT-2020-0001

Advancing Scientific Productivity through Better Scientific Software:
Developer Productivity and Software Sustainability Report

IDEAS-ECP Team and Collaborators

January 28, 2020


Home About Research News Podcast Videos Training

BETTER SCIENTIFIC PRODUCTIVITY THROUGH BETTER SCIENTIFIC SOFTWARE: THE IDEAS REPORT

01/30/20

The US Department of Energy's Exascale Computing Project (ECP) provides a unique opportunity to advance computational science through an accelerated growth phase in extreme-scale computing. However, disruptive changes in computer architectures and the complexities of tackling new frontiers in extreme-scale modeling, simulation, and analysis present daunting challenges to the productivity of software developers and the sustainability of software artifacts.

A newly released report introduces work by the IDEAS project within ECP (called IDEAS-ECP) to foster and advance software productivity and sustainability for extreme-scale computational science, as a key aspect of improving overall scientific productivity. The report explains the IDEAS approach, outcomes, and impact of work (in partnership with the ECP and broader computational science community).

Target readers are all those who care about the quality and integrity of scientific discoveries based on simulation and analysis. While the difficulties of extreme-scale computing intensify software challenges, issues are relevant across all computing scales, given universal increases in complexity and the need to ensure the trustworthiness of computational results.

The report may be obtained from the ECP website.

https://exascaleproject.org/better-scientific-productivity-through-better-scientific-software-the-ideas-report

Community organizations: changing the culture in which research software is developed and sustained

D.S. Katz, L.C. McInnes, D.E. Bernholdt, A.Cabunoc Mayes, N.P. Chue Hong, J. Duckles, S. Gesing, M.A. Heroux, S. Hettrick, R.C. Jimenez, M. Pierce, B. Weaver, N. Wilkins-Diehr, 2019, special Issue of IEEE Computing in Science and Engineering (CiSE) on Accelerating Scientific Discovery with Reusable Software, DOI:10.1109/MCSE.2018.2883051, arXiv:1811.08473

Resources and opportunities to get involved:

- WSSSPE: http://wssspe.researchcomputing.org.uk
  - International community-driven organization that promotes sustainable research software
- NUMFocus: https://www.numfocus.org
  - Umbrella nonprofit that supports and promotes open-source scientific computing
- Software Carpentry: http://software-carpentry.org
  - Volunteer non-profit dedicated to teaching basic computing skills to researchers.
  - Lessons: https://software-carpentry.org/lessons
- Software Sustainability Institute: http://www.software.ac.uk
  - Institute to support UK's research software community: cultivating better, more sustainable, research software to enable world-class research
  - Guides: https://www.software.ac.uk/resources/guides-everything

Thank you !

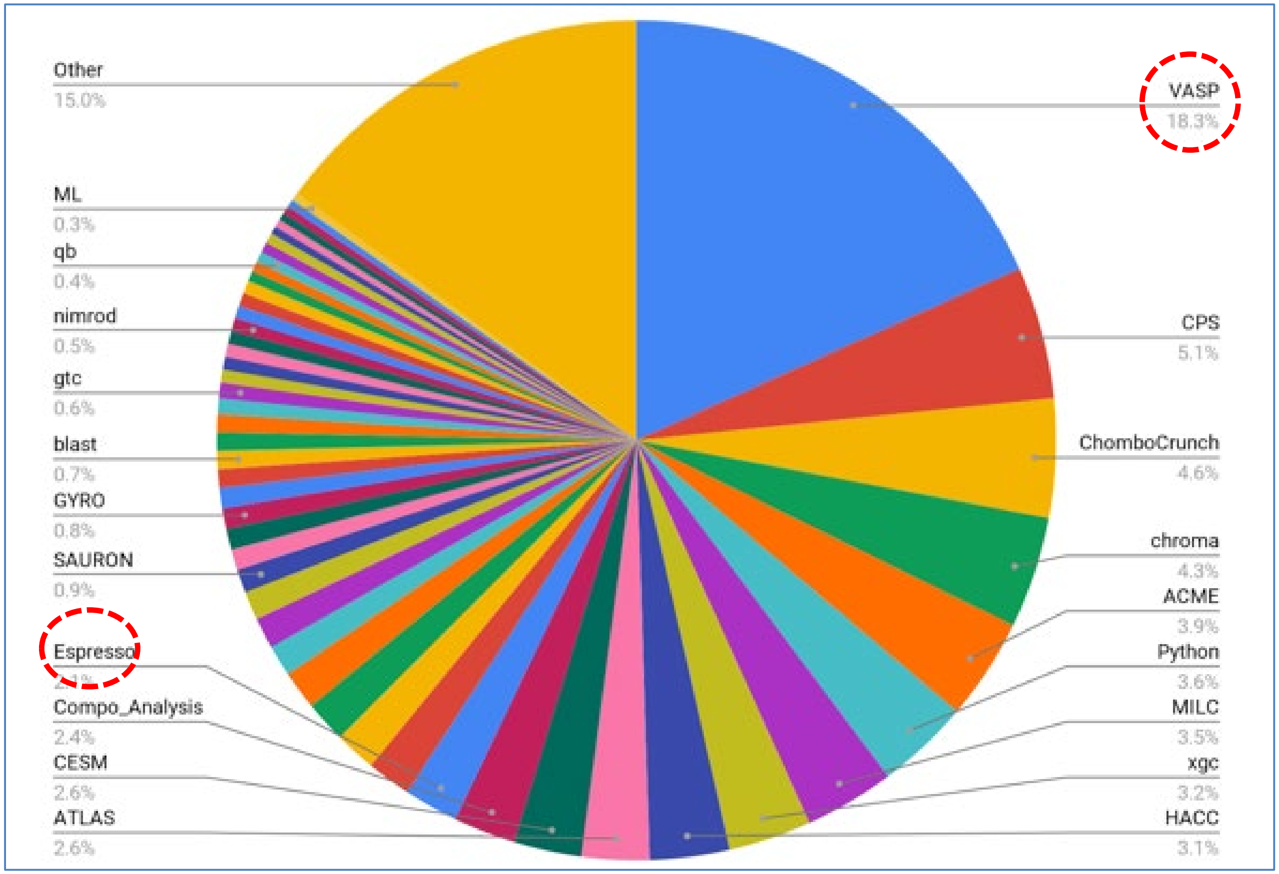
Minimization Strategies for Solving Eigenvalue Problems

Osni Marques
Lawrence Berkeley National Laboratory
oamarques@lbl.gov

Joint work with Doru Thom Popovici, Mauro Del Ben and Andrew Canning
funding from DOE's SciDAC FASTMath

How are computer cycles used?

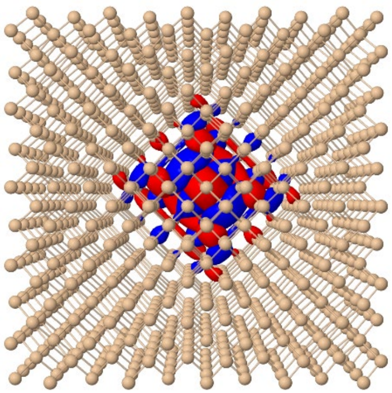
NERSC System Utilization (Aug’17 - Jul’18)

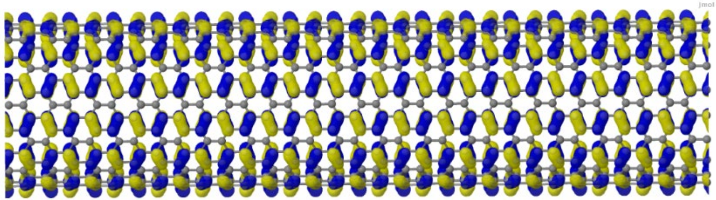


- electronic structure DFT eigenvalue problems ~ 25% of the workload
- 10 codes > 50% of the workload
- 35 codes > 75% of the workload
- Over 600 codes comprise the remaining 25% of the workload.

Electronic Structure of Materials

- Schrödinger equation: $\hat{H}\Psi = E\Psi$, $\Psi(\vec{r}_1, \ldots, \vec{r}_n)$
  - Many-particle equation
  - Very expensive to be solved (exponential)
  - Unpractical for large systems
- Density Functional Theory (DFT): $H\psi_i = E_i\psi_i$
  - Kohn and Pople, Nobel Prize in Chemistry, 1998
  - Maps the many-particle problem into a single-particle problem
  - Accurate results for structural and electronic properties of materials
  - Need to be solved self–consistently
  - $O(N^3)$ scaling with system size





Self-Consistency: Nonlinear Eigenvalue Problem

initial guess $\{\psi_i\}$

↓

calculate density

$$\rho(\vec{r}) = \sum_{i=1}^{N} |\psi_i(\vec{r})|^2$$

↓

update $H(\rho)$

↓

solve $H\psi_i = E_i\psi_i$

new set $\{\psi_i\}$

Direct Methods

- ScaLAPACK
- EigenExa
- ELPA

Iterative Methods

- only a small fraction (2-10%) of (smallest) eigenpairs is required
- limited/poor parallel performance for conventional diagonalization and/or reorthogonalization, $O(N^3)$

$$H\psi_i(r) = \left[-\frac{1}{2}\nabla^2 + V\right]\psi_i(r) = \varepsilon_i\psi_i(r)$$

$$\psi_i(r) = \sum_{j=1}^{m} c_{ji}\varphi_j(r)$$

Iterative Methods for $H\psi_i = E_i\psi_i$

- (Jacobi-)Davidson
- Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG)
- (Polynomial filtered) Lanczos
- Conjugate gradient minimization of $\psi_i^* H \psi_i$

References

- *Sameh and Wisniewski, A trace minimization algorithm for the generalized eigenvalue problem, SIAM Journal on Numerical Analysis, 19, 1243-1259 (1982)*
- *Teter, Payne, Allen, Solution of Schrodinger's equation for large systems, Phys. Rev. B 40, 12255 (1994)*
- *Mauri and Galli, Electronic-structure calculations and molecular-dynamics simulations with linear system-size scaling, Phys. Rev. B 50, 4316 (1994)*
- *Kresse and Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, Phys. Rev. B 54, 11169 (1996)*
- *Pfrommer, Demmel and Simon, Unconstrained energy functionals for electronic structure calculations, J. Comput. Phys., 150 (1999)*
- *Saad, Chelikowsky and Shontz, Numerical methods for electronic structure calculations of materials, SIAM Review 52, 3–54 (2010)*
- *Jordan, Marsman, Kim and Kresse, Fast iterative interior eigensolver for millions of atoms, J. Comput. Phys., 231 (2012)*
- *Corsetti, The orbital minimization method for electronic structure calculations with finite-range atomic basis sets, Comput. Phys. Comm., 185 (2014)*
- *Levitt and Torrent, Parallel eigensolvers in plane-wave Density Functional Theory, Comput. Phys. Comm., 187 (2014)*

Standard (constrained) iterative CG eigensolver versus unconstrained iterative CG eigensolver

- Constrained CG method for iterative eigensolver
  - $\min_{\Psi} \mathrm{Tr}\,[\Psi^T H \Psi],\ \Psi = [\psi_1, \psi_2, \ldots \psi_N],\ \Psi^T\Psi = I$
  - CG steps followed by reorthogonalization with ScaLAPACK
  - Typically matrix size 100,000 to millions (dimension of H)
  - Operations on H and ψ_i (matrix vector for CG steps) scale well
  - Operations on small subspace scale poorly (reorthogonalization)
- Unconstrained CG method for iterative eigensolver (simplest form)
  - $\min_{\mathrm{X}} \mathrm{Tr}\,[\mathcal{S}^{-1}\mathrm{X}^T H \mathrm{X}],\ \mathcal{S} = \mathrm{X}^T\mathrm{X},\ \Psi = \mathrm{X}\mathcal{S}^{-\frac{1}{2}}$
  - $\mathcal{S}^{-1} \approx (2I - \mathcal{S})$ (1^{st} order expansion)
  - Functional has same minimum as constrained functional (trial eigenvectors orthogonal at minimum)
  - No operations on subspace matrix (scales to large core counts)
  - Convergence properties different from constrained functional

Standard (constrained) iterative CG eigensolver versus unconstrained iterative CG eigensolver

Constrained CG Method

- minimize

$$\min_{\Psi} Tr[\Psi^T \cdot H \cdot \Psi]\ s.t.\ \Psi^T \cdot \Psi = I$$

- gradient

$$\nabla = 2 \cdot H \cdot \Psi$$

- update

$$\Psi_{new} = \Psi_{old} \cdot \cos(\theta) + P_{search} \cdot \sin(\theta)$$

- orthogonalize

$$\Psi^T \cdot \Psi = I$$

Unconstrained CG Method

- minimize

$$\min_{\Psi} Tr[(2 \cdot I - \Psi^T \cdot \Psi) \cdot \Psi^T \cdot H \cdot \Psi]$$

- gradient

$$\nabla = 4 \cdot H \cdot \Psi - 2 \cdot H \cdot \Psi \cdot \Psi^T \cdot \Psi - 2 \cdot \Psi \cdot \Psi^t \cdot H \cdot \Psi$$

- update

$$\Psi_{new} = \Psi_{old} + \alpha \cdot P_{search}$$

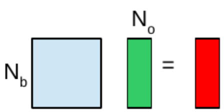
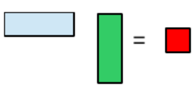
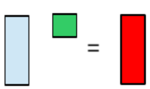
Data Movement

Computation

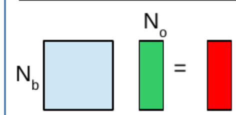
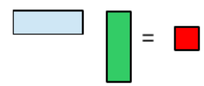
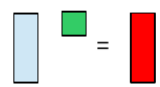
Operations for constrained iterative CG eigensolver and unconstrained iterative CG eigensolver

N_b = matrix dimension, N_o = number of eigenpairs (1-10% of N_b), p = number of processors

Constrained Solver

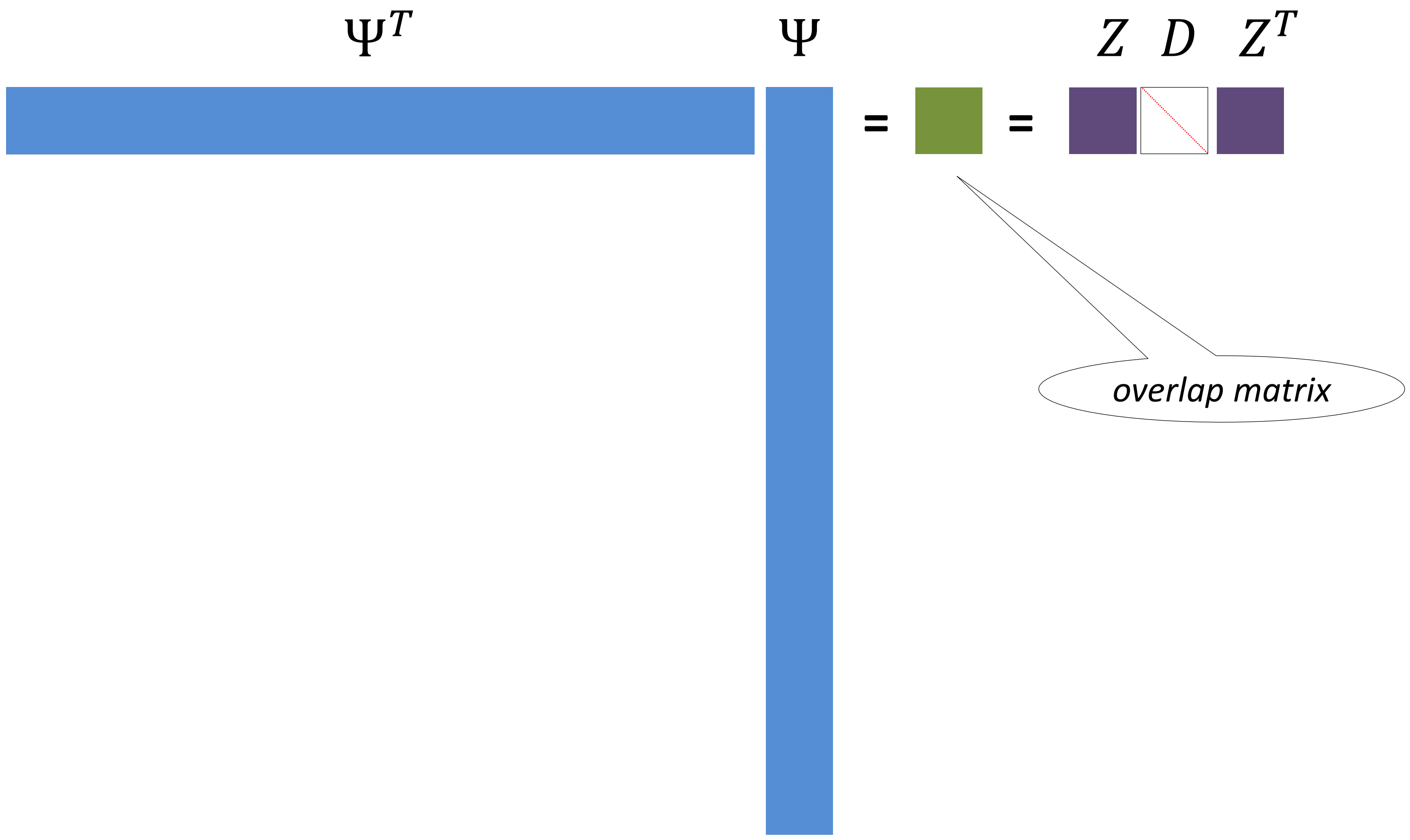
Operation	Label	Comput.	Commun.
	HX	$\frac{N_b^2 \cdot N_o}{p}$	$\frac{N_b \cdot (N_o + N_b)}{\sqrt{p}}$
	XtP	$\frac{N_b \cdot N_o^2}{p}$	$\frac{2N_b \cdot N_o}{\sqrt{p}}$
	XB	$\frac{N_b \cdot N_o^2}{p}$	$\frac{N_o \cdot (N_b + N_o)}{\sqrt{p}}$
	Transf.	$\frac{N_o^3}{p}$	$\frac{N_o^2}{\sqrt{p}}$

Unconstrained Solver

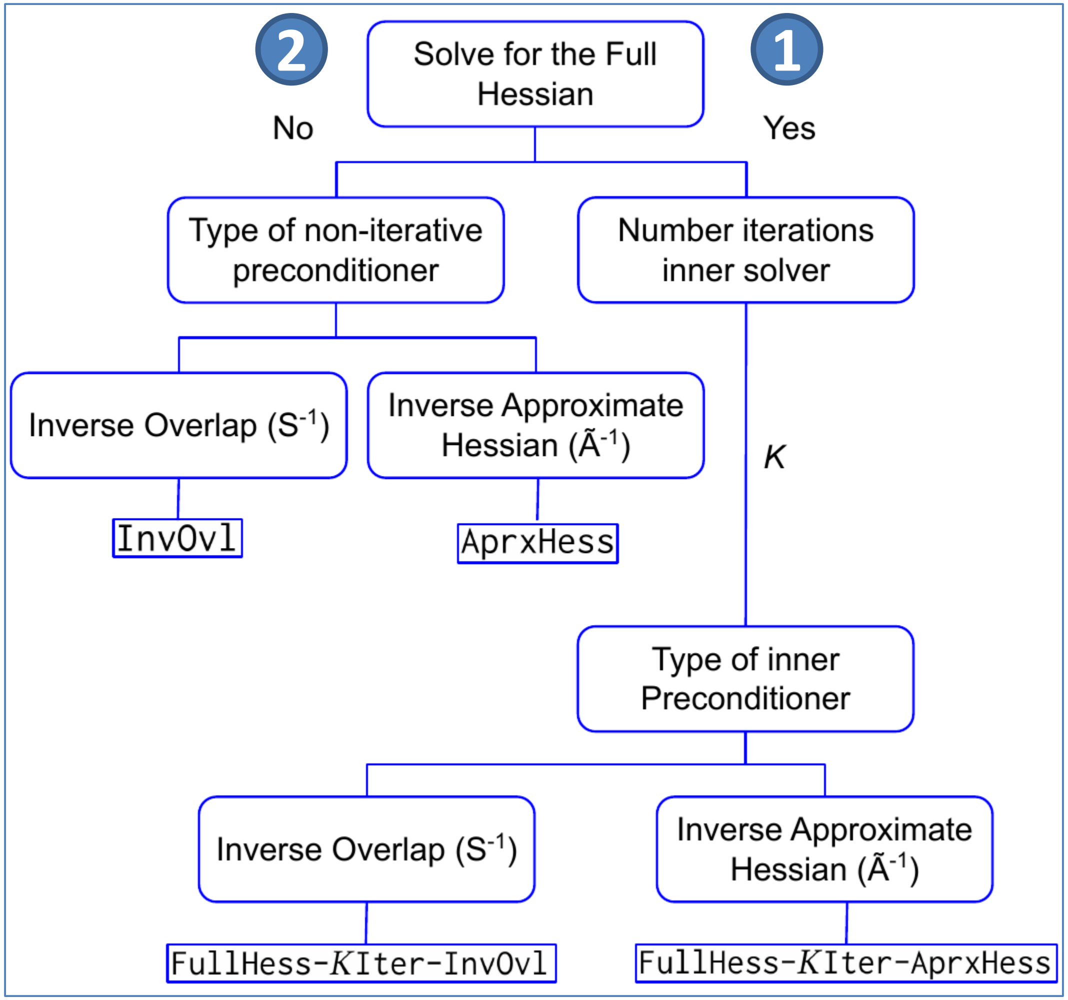
Operation	Label	Comput.	Commun.
	HX	$\frac{N_b^2 \cdot N_o}{p}$	$\frac{N_b \cdot (N_o + N_b)}{\sqrt{p}}$
	XtP	$\frac{N_b \cdot N_o^2}{p}$	$\frac{2N_b \cdot N_o}{\sqrt{p}}$
	XB	$\frac{N_b \cdot N_o^2}{p}$	$\frac{N_o \cdot (N_b + N_o)}{\sqrt{p}}$

- Important questions for constrained and unconstrained eigensolvers:
  - Convergence rate
  - Parallel scaling
  - Stability
- Unconstrained formulation can be applied to other matrices
  - Tested on Harwell-Boeing matrices

PCG for $Ax = \lambda x$: *orthogonality versus scalability*



Novel Preconditioners for PCG



$$\min_{X} \mathrm{Tr}\,[\mathcal{S}^{-1} X^T H X]$$

$$\mathcal{S}^{-1} \approx (2I - \mathcal{S})$$

$$G = 4HX - 2\mathcal{S}X\mathcal{H} - 2HX\mathcal{S}$$

$$\mathcal{H} = X^T H X$$

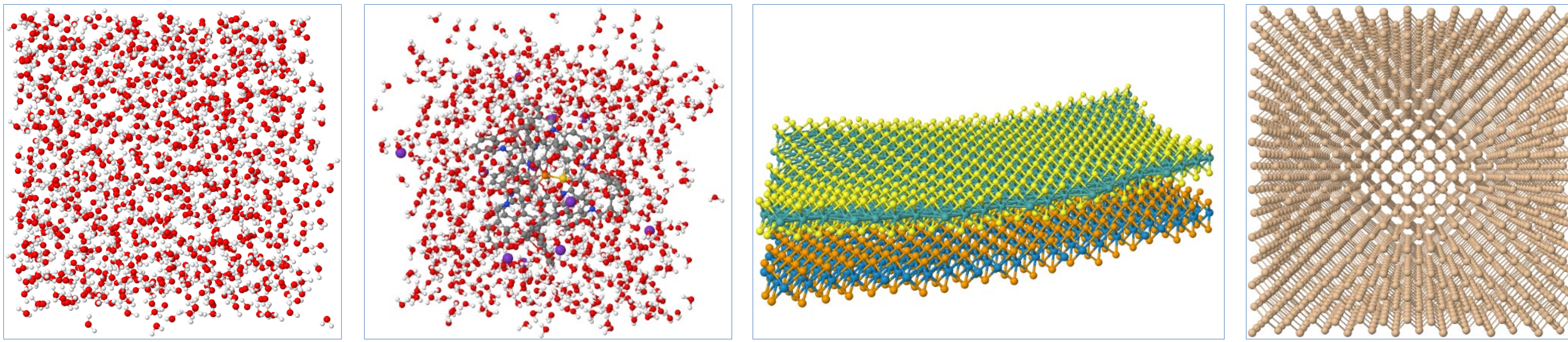
- **Option 1**: Hessian A of the unconstrained functional to precondition the gradient, $A^{-1}G$ (quasi Newton step)
  - solve $AP = G$ iteratively
  - S^{-1}or $\tilde{A}^{-1}$ as preconditioner for the inner solver (with K iterations)
  - $\tilde{A} \approx A$
- **Option 2**: use S^{-1}or $\tilde{A}^{-1}$ to precondition the unconstrained functional minimization

Numerical Experiments

- CP2K
  - quantum chemistry and solid state physics package
  - DFT using mixed Gaussian and plane waves approaches
  - Non-orthogonal basis, generalized eigenvalue problem $HC = SCE$
- Cray XC40 system (cori @ NERSC)
  - 2,388 Intel Xeon 16-core Intel Xeon Haswell
  - 9,688 68-core Intel Xeon Phi Knights Landing (KNL)
  - Hybrid MPI+OpenMP implementation
  - Intel compiler, MKL, ELPA, and LIBXSMM (latest available releases)

Del Ben, Marques and Canning, Improved Unconstrained Energy Functional Method for Eigensolvers in Electronic Structure Calculations, ICPP 2019, Kyoto, Japan.

Systems Used in the Numerical Experiments



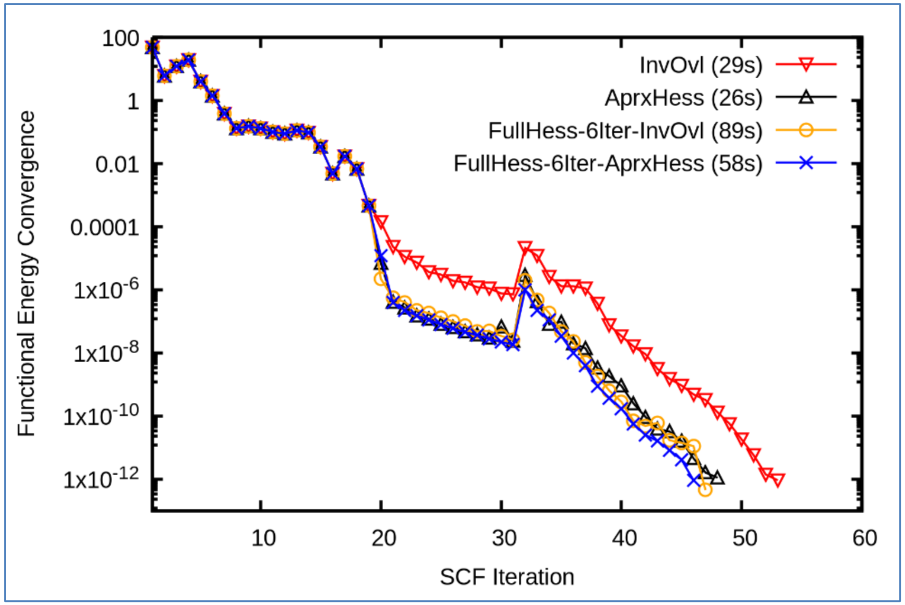
Systems used in the numerical experiments, in increasing order of "complexity" for convergence: 1024 molecules of bulk liquid water, supramolecular catalyst gold(III)-complex, bilayer of MoS_2-WSe_2, and divacancy point defect in silicon. The number of atoms range from 2,247 to 12,288.

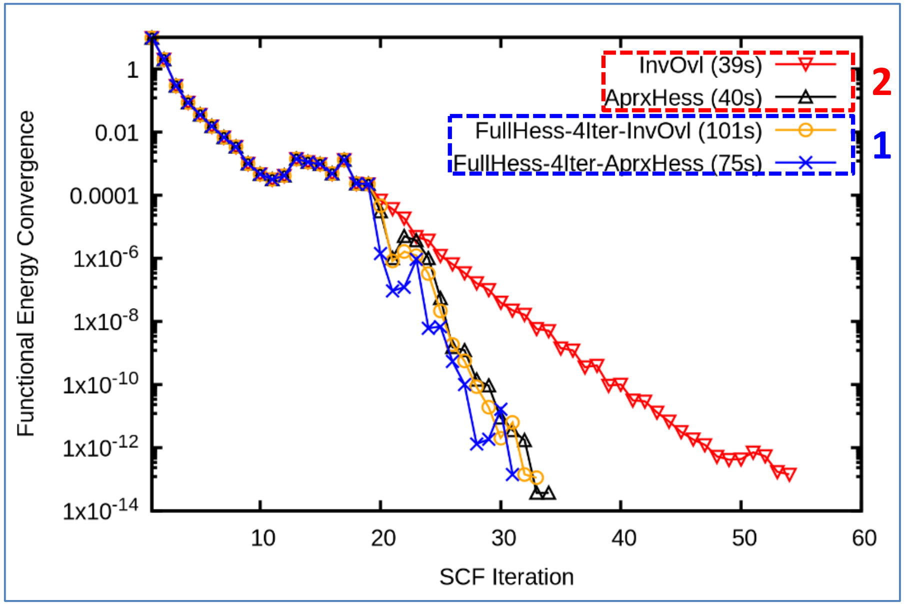
Physical and Computational Parameters

System	Label	Atoms	Basis	N_b	N_o	N_b/N_o	gap(AU)
Bulk liquid water	Water-1024	3,072	TZVP	29,696	4,096	0.14	0.128
	Water-2048	6,144	TZVP	59,392	8,192	0.14	
	Water-4096	12,288	TZVP	118,784	16,384	0.14	
Solvated catalyst complex	Complex	2,590	TZVP	26,339	3,605	0.14	0.052
MoS_2-WSe_2 bilayer	BiLayer	2,247	TZVP	51,681	9,737	0.19	0.035
Divacancy defect in silicon	SiDivac	2,742	TZVP	46,614	5,484	0.12	0.013
	SiDivac-SZV	2,742	SZV	10,968	5,484	0.5	
	SiDivac-DZVP	2,742	DZVP	35,646	5,484	0.15	
	SiDivac-TZV2P	2,742	TZV2P	79,518	5,484	0.07	

Basic physical and computational parameters of the systems employed in the numerical experiments. N_b is the basis set size, N_o is the number of eigenvectors to be computed (number of wavefunctions needed to build the electronic density), and gap is the energy difference between eigenvalues N_o and N_o+1 in atomic units (AU), the unit employed to express H.

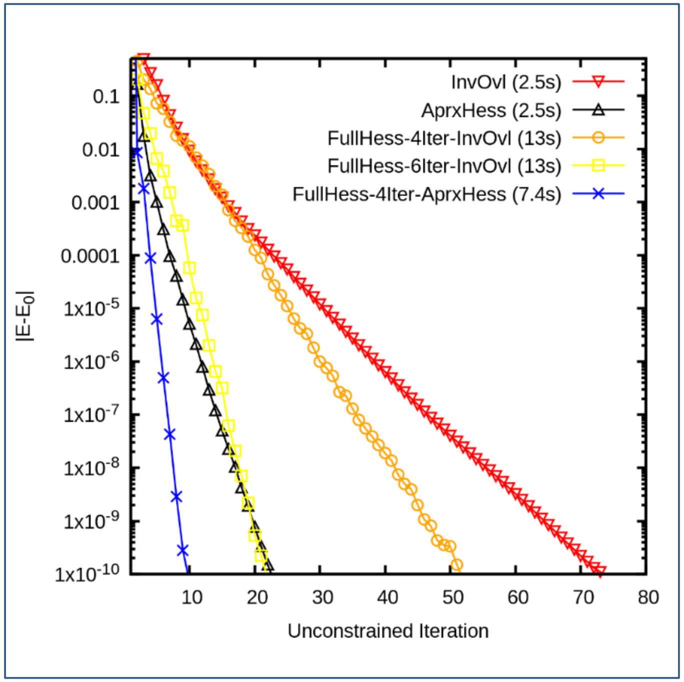
Convergence of the SCF Procedure

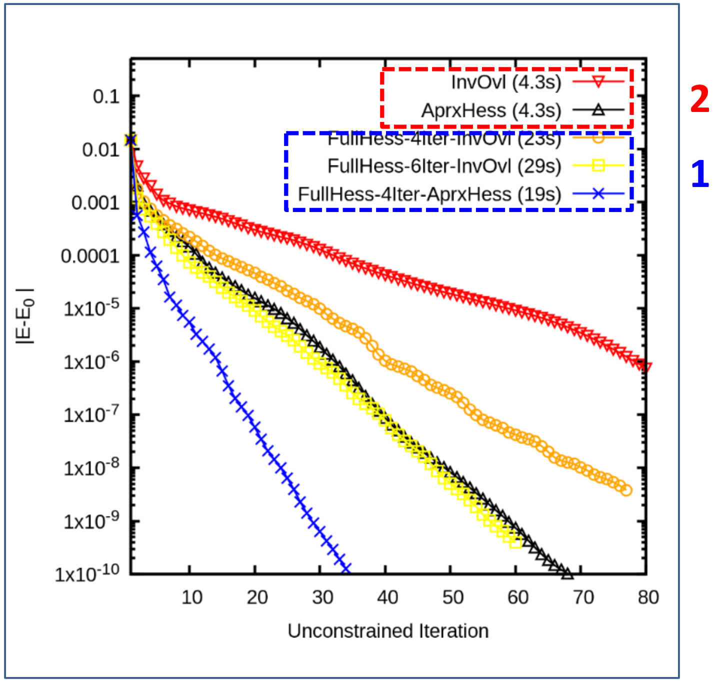




Convergence of the SCF procedure. Four setups; the average time for a single SCF step is given in parenthesis. Left: Complex. Right: SiDivac.

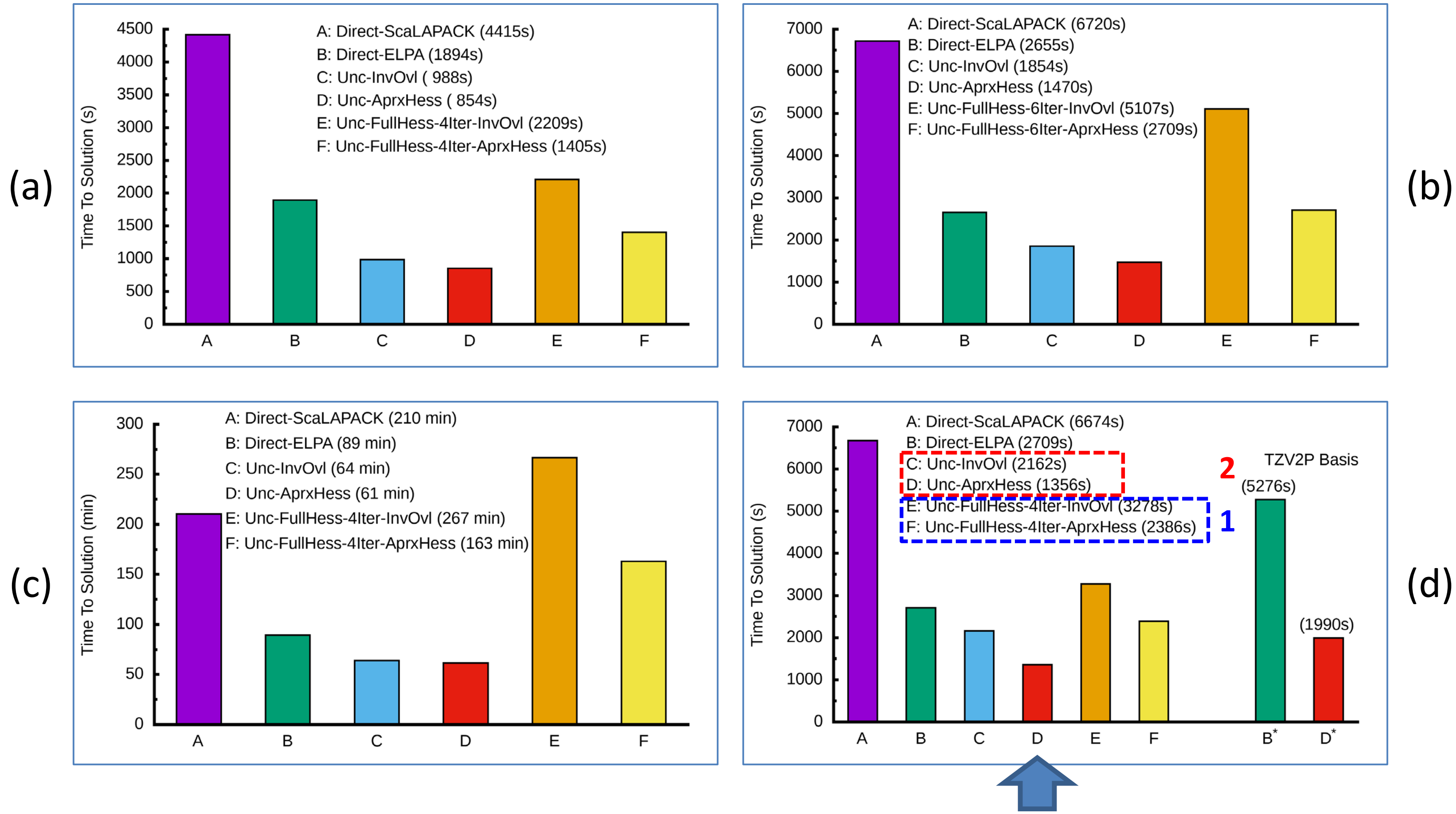
Convergence of the Energy





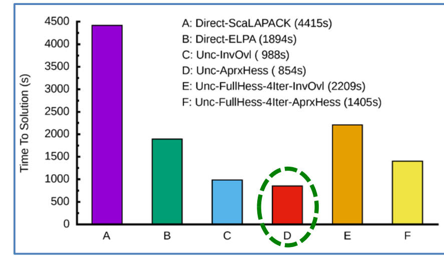
Convergence of the energy (unconstrained objective function) for a single unconstrained functional diagonalization (unconstrained subspace minimization). Five setups; the time for a single unconstrained-PCG iteration is given in parenthesis. Left: Complex. Right: SiDivac.

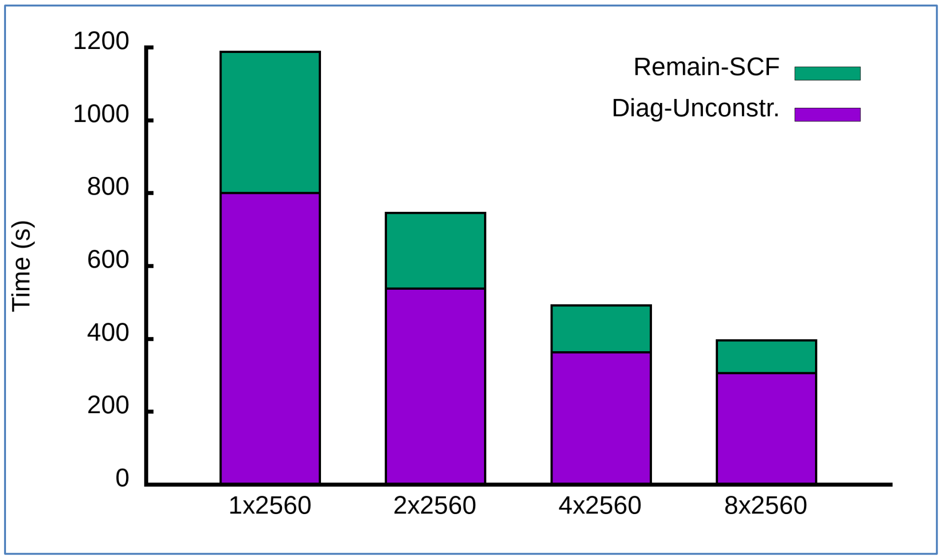
Time to Solution for Full SCF

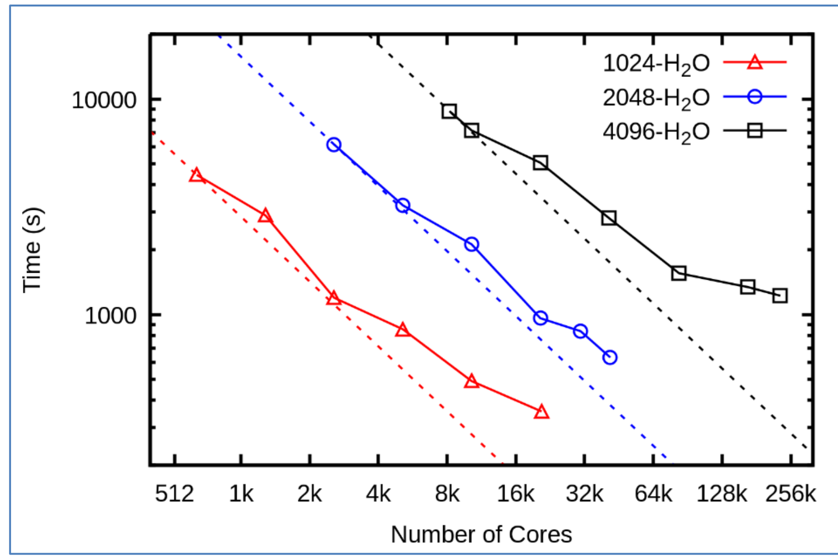


Time to solution for full SCF convergence compared to direct solvers (ScaLAPACK and ELPA). (a) Water-1024, (b) Complex, (c) BiLayer and (d) SiDivac. Actual times are given in parenthesis. For SiDivac, B^* and D^* are times obtained with a larger basis (about 1.7 times larger than in B and D, with 160 KNL nodes).

Strong Scaling







Left: OpenMP threads per MPI task for a fixed number of MPI tasks (2560), Water-1024 (method D in the figure above). Right: time to solution for bulk liquid water with 1024, 2048 and 4096 molecules (method D in the previous slide).

Summary

Main conclusions:

- Unconstrained CG offers good parallel scalability and outperforms standard diagonalization
- Implementation within a localized basis set allows for efficient sparse and dense linear algebra implementations

Ongoing and future work:

- Sub-group parallelization for small matrix multiplication
- Implementation in a plane wave basis framework (http://qboxcode.org)
- GPU implementation and comparisons with other iterative strategies
- Mixed precision and automatic tuning
- Applications of unconstrained minimization in other areas

Thank you !