جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Adaptive Nonlinear Preconditioning for PDEs with Error Bounds on Output Functionals

David Keyes, KAUST

From a nonlinear guy to a linear guy at his 71^{st}, with deep gratitude

8 July 2021

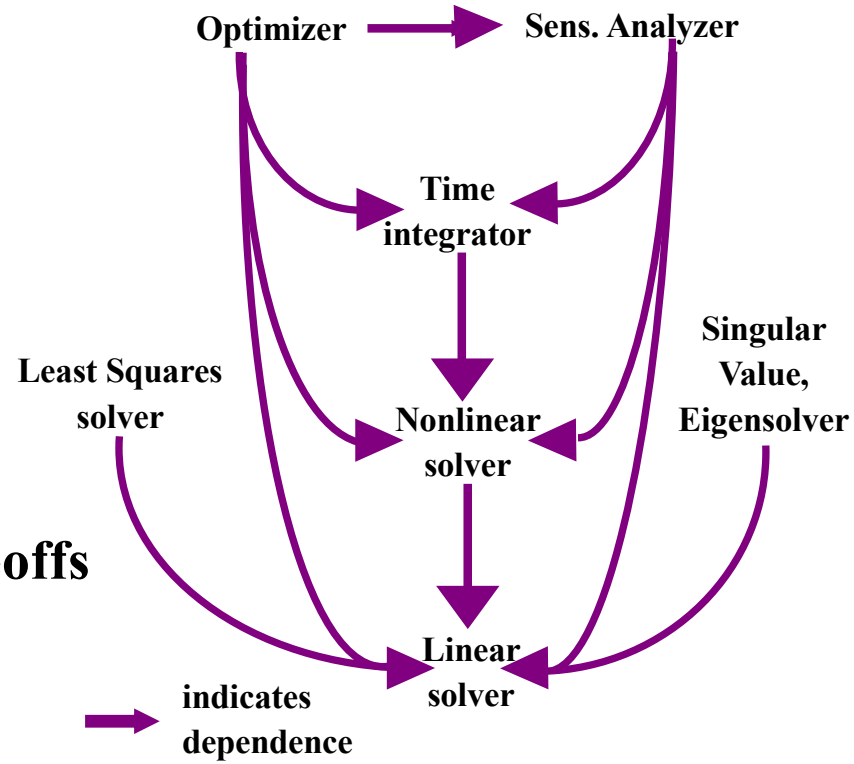
Solvers and their nestedness

To achieve the potential of emerging architectures for scientific applications, we need implementations of fast

- linear and least squares solvers
- eigensolvers & singular value solvers
- nonlinear and optimization solvers
- time integrators & sensitivity solvers
- stencil solvers

that

- offer tunable accuracy-time-space tradeoffs
- exploit data sparsity
- exploit hierarchy of precisions
- are highly concurrent
- minimize communication and synchronization
- are energy efficient



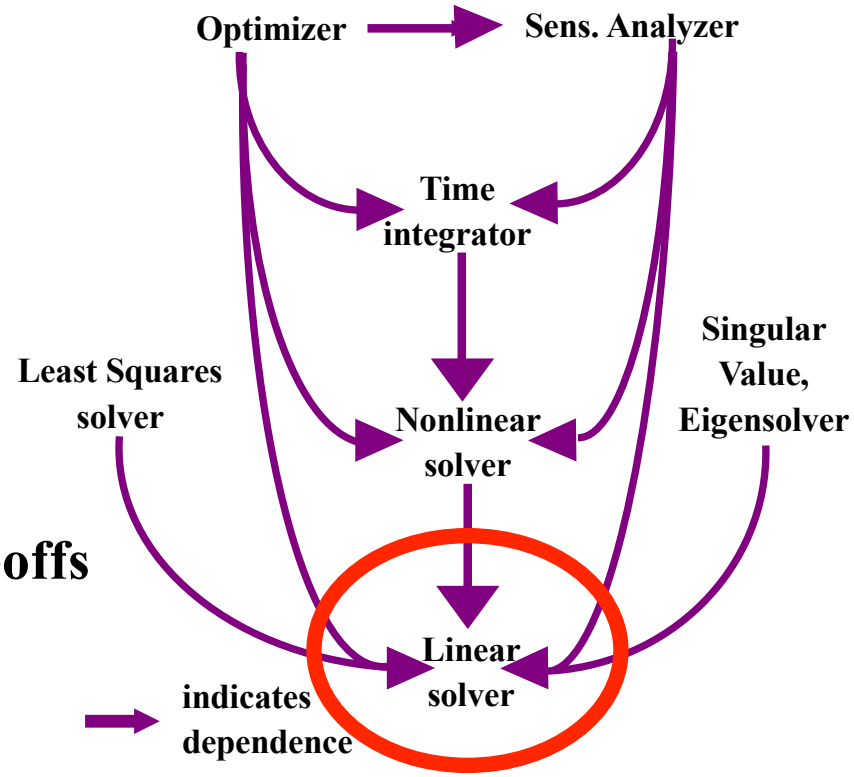
Solvers and their nestedness

To achieve the potential of emerging architectures for scientific applications, we need implementations of fast

- linear and least squares solvers
- eigensolvers & singular value solvers
- nonlinear and optimization solvers
- time integrators & sensitivity solvers
- stencil solvers

that

- offer tunable accuracy-time-space tradeoffs
- exploit data sparsity
- exploit hierarchy of precisions
- are highly concurrent
- minimize communication and synchronization
- are energy efficient



Terascale Optimal PDE Simulations

http://www.tops-scidac.org

SciDAC
Scientific Discovery through Advanced Computing

Who we are…

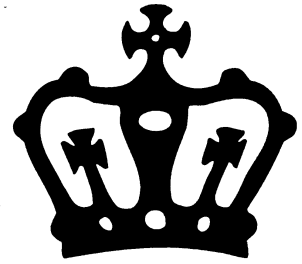
… the PETSc and TAO people

… the hypre and Sundials people

Berkeley Lab

… the SuperLU and PARPACK people

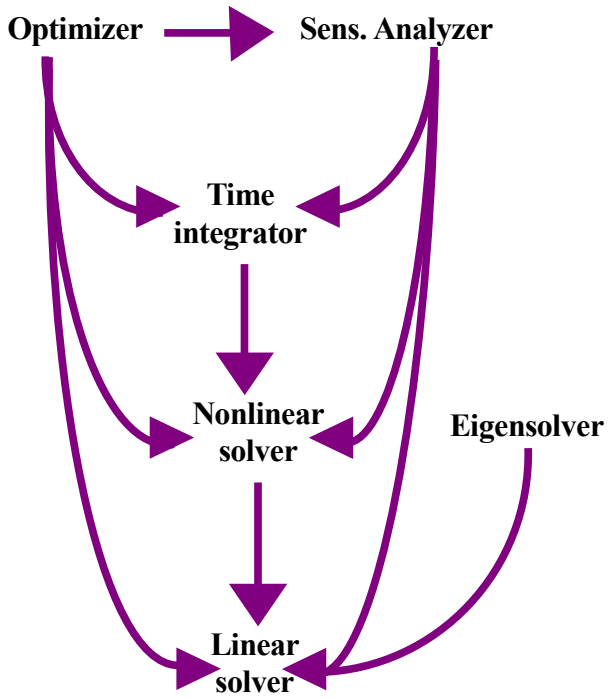
Plus some university collaborators …



… with a history of lab collaborations in high performance computing

Scope for TOPS

- **Design and implementation of "solvers"**
  - **Time integrators (w/ sens. anal.)** $f(\dot{x}, x, t, p) = 0$
  - **Nonlinear solvers (w/ sens. anal.)** $F(x, p) = 0$
  - **Optimizers** $\min_u \phi(x,u)\ s.t.\ F(x,u) = 0, u \geq 0$
  - **Linear solvers** $Ax = b$
  - **Eigensolvers** $Ax = \lambda Bx$



- **Software integration**
- **Performance optimization**

Scalable solvers for PDE-based and other simulations

Towards Optimal Petascale Simulations

Int J High Performance Computing Applications **25**:3-60 (2011)

INTERNATIONAL EXASCALE SOFTWARE PROJECT ROADMAP 1.0

Jack Dongarra, Pete Beckman, Terry Moore, Patrick Aerts, Giovanni Aloisio, Jean-Claude Andre, David Barkai, Jean-Yves Berthou, Taisuke Boku, Bertrand Braunschweig, Franck Cappello, Barbara Chapman, Xuebin Chi, Alok Choudhary, Sudip Dosanjh, Thom Dunning, Sandro Fiore, Al Geist, Bill Gropp, Robert Harrison, Mark Hereld, Michael Heroux, Adolfy Hoisie, Koh Hotta, Yutaka Ishikawa, Fred Johnson, Sanjay Kale, Richard Kenway, David Keyes, Bill Kramer, Jesus Labarta, Alain Lichnewsky, Thomas Lippert, Bob Lucas, Barney Maccabe, Satoshi Matsuoka, Michael E. Papka, Dan Reed, Mitsuhisa Sato, Ed Seidel, John Shalf, David Skinner, Marc Snir, Thomas Sterling, Rick Stevens, Fred Streitz, Matthias Mueller, Wolfgang Nagel, Hiroshi Nakashima, Paul Messina, Peter Michielse, Bernd Mohr, Bob Sugar, Shinji Sumimoto, William Tang, John Taylor, Rajeev Thakur, Anne Trefethen, Mateo Valero, Aad van der Steen, Jeffrey Vetter, Peg Williams, Robert Wisniewski, Kathy Yelick

SPONSORS

downloadable at
exascale.org/iesp

Check for updates

THE INTERNATIONAL JOURNAL of HIGH PERFORMANCE COMPUTING APPLICATIONS

The International Journal of High Performance Computing Applications
25(1) 3–60
© The Author(s) 2011
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1094342010391989
hpc.sagepub.com
SAGE

The International Exascale Software Project roadmap

Jack Dongarra, Pete Beckman, Terry Moore, Patrick Aerts, Giovanni Aloisio, Jean-Claude Andre, David Barkai, Jean-Yves Berthou, Taisuke Boku, Bertrand Braunschweig, Franck Cappello, Barbara Chapman, Xuebin Chi, Alok Choudhary, Sudip Dosanjh, Thom Dunning, Sandro Fiore, Al Geist, Bill Gropp, Robert Harrison, Mark Hereld, Michael Heroux, Adolfy Hoisie, Koh Hotta, Zhong Jin, Yutaka Ishikawa, Fred Johnson, Sanjay Kale, Richard Kenway, David Keyes, Bill Kramer, Jesus Labarta, Alain Lichnewsky, Thomas Lippert, Bob Lucas, Barney Maccabe, Satoshi Matsuoka, Paul Messina, Peter Michielse, Bernd Mohr, Matthias S. Mueller, Wolfgang E. Nagel, Hiroshi Nakashima, Michael E Papka, Dan Reed, Mitsuhisa Sato, Ed Seidel, John Shalf, David Skinner, Marc Snir, Thomas Sterling, Rick Stevens, Fred Streitz, Bob Sugar, Shinji Sumimoto, William Tang, John Taylor, Rajeev Thakur, Anne Trefethen, Mateo Valero, Aad van der Steen, Jeffrey Vetter, Peg Williams, Robert Wisniewski and Kathy Yelick

Abstract

Over the last 20 years, the open-source community has provided more and more software on which the world's high-performance computing systems depend for performance and productivity. The community has invested millions of dollars and years of effort to build key components. However, although the investments in these separate software elements have been tremendously valuable, a great deal of productivity has also been lost because of the lack of planning, coordination, and key integration of technologies necessary to make them work together smoothly and efficiently, both within individual petascale systems and between different systems. It seems clear that this completely uncoordinated development model will not provide the software needed to support the unprecedented parallelism required for peta/exascale computation on millions of cores, or the flexibility required to exploit new hardware models and features, such as transactional memory, speculative execution, and graphics processing units. This report describes the work of the community to prepare for the challenges of exascale computing, ultimately combing their efforts in a coordinated International Exascale Software Project.

Keywords

exascale computing, high-performance computing, software stack

Table of Contents

1. Introduction 6
2. Destination of the IESP Roadmap 7
3. Technology Trends and their Impact on Exascale 8
   3.1 Technology Trends 8
   3.2 Science Trends 9

University of Tennessee at Knoxville, USA

Corresponding author:
Jack Dongarra, University of Tennessee, at Knoxville, 1122 Volunteer Boulevard, Suite 203, Knoxville, TN 37996-3450, USA.
Email: dongarra@cs.utk.edu

Int J High Performance Computing Applications **34**:435-479 (2018)

downloadable at
exascale.org/bdec

International Journal of
HIGH PERFORMANCE
COMPUTING APPLICATIONS

Research Paper

The International Journal of High Performance Computing Applications
2018, Vol. 32(4) 435–479

DOI: 10.1177/1094342018778123
journals.sagepub.com/home/hpc
SAGE

Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry

M Asch, T Moore, R Badia, M Beck, P Beckman, T Bidot, F Bodin, F Cappello, A Choudhary, B de Supinski, E Deelman, J Dongarra, A Dubey, G Fox, H Fu, S Girona, W Gropp, M Heroux, Y Ishikawa, K Keahey, D Keyes, W Kramer, J-F Lavignon, Y Lu, S Matsuoka, B Mohr, D Reed, S Requena, J Saltz, T Schulthess, R Stevens, M Swany, A Szalay, W Tang, G Varoquaux, J-P Vilotte, R Wisniewski, Z Xu and I Zacharov

Abstract

Over the past four years, the Big Data and Exascale Computing (BDEC) project organized a series of five international workshops that aimed to explore the ways in which the new forms of data-centric discovery introduced by the ongoing revolution in high-end data analysis (HDA) might be integrated with the established, simulation-centric paradigm of the high-performance computing (HPC) community. Based on those meetings, we argue that the rapid proliferation of digital data generators, the unprecedented growth in the volume and diversity of the data they generate, and the intense evolution of the methods for analyzing and using that data are radically reshaping the landscape of scientific computing. The most critical problems involve the logistics of wide-area, multistage workflows that will move back and forth across the computing continuum, between the multitude of distributed sensors, instruments and other devices at the networks edge, and the centralized resources of commercial clouds and HPC centers. We suggest that the prospects for the future integration of technological infrastructures and research ecosystems need to be considered at three different levels. First, we discuss the convergence of research applications and workflows that establish a research paradigm that combines both HPC and HDA, where ongoing progress is already motivating efforts at the other two levels. Second, we offer an account of some of the problems involved with creating a converged infrastructure for peripheral environments, that is, a shared infrastructure that can be deployed throughout the network in a scalable manner to meet the highly diverse requirements for processing, communication, and buffering/storage of massive data workflows of many different scientific domains. Third, we focus on some opportunities for software ecosystem convergence in big, logically centralized facilities that execute large-scale simulations and models and/or perform large-scale data analytics. We close by offering some conclusions and recommendations for future investment and policy review.

Keywords

Big data, extreme-scale computing, future software, traditional HPC, high-end data analysis

1. Executive summary

Although the "big data" revolution first came to public prominence (circa 2010) in online enterprises like Google, Amazon, and Facebook, it is now widely recognized as the initial phase of a watershed transformation that modern society generally—and scientific and engineering research in particular—are in the process of undergoing. Responding to this disruptive wave of change, over the past 4 years,

Innovative Computing Laboratory, University of Tennessee, Knoxville, TN, USA

Corresponding author:
J Dongarra, University of Tennessee, Knoxville, TN 37996, USA.
Email: dongarra@icl.utk.edu

جامعة الملك عبدالله
للعلوم والتقنية

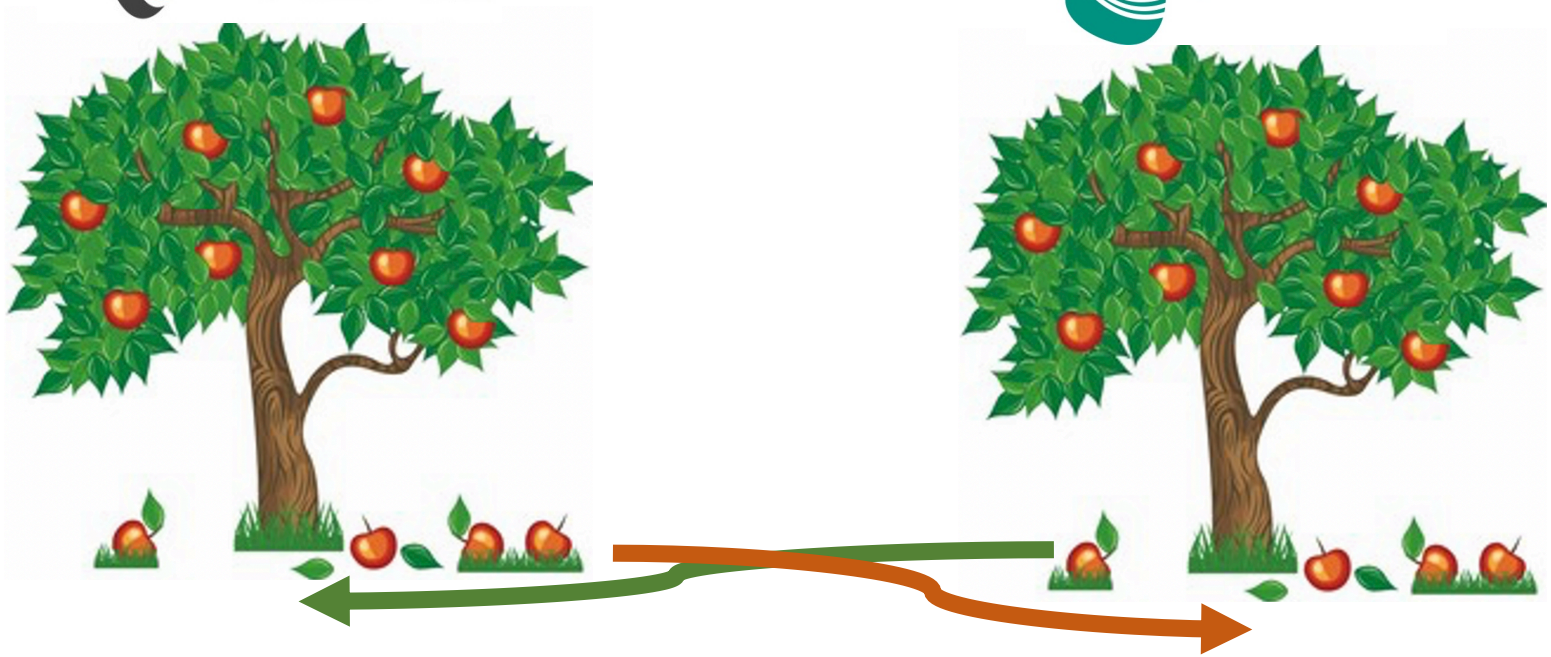
King Abdullah University of
Science and Technology

KAUST Faculty PIs currently using Shaheen

- Aamir Farooq
- Ajay Jasra
- Alexandre Rosado
- Andrea Fratalocchi
- Arnab Pain
- Athanasios Tzavaras
- Atif Shamim
- Basem Shihada
- Bernard Ghanem
- Burton Jones
- Carlos Duarte
- Charlotte Hauser
- Cristian Picioreanu
- Daniel Peter
- Daniele Boffi
- David Ketcheson
- David Keyes
- Deanna Lacoste
- Dominik Michels
- Enzo Di Fabrizio
- Eric Feron
- Gabriel Wittum
- Geert Jan Witkamp
- Georgiy Stenchikov
- Hakan Bagci
- Hernando Ombao
- Himanshu Mishra
- Hong Im
- Hossein Fariborzi
- Hussein Hoteit
- Hussam Alshareef
- Ibrahim Hoteit
- Iman Roqan
- Jean-Marie Basset
- Jerry Schuster
- Jesper Tegner
- Johannes Vrouwenvelder
- Jorge Gascon
- Jr-Hau He
- Kuo-Wei Huang
- Lain-Jong Li
- Luigi Cavallo
- Magdy Mahfouz
- Magnus Rueping
- Mani Sarathy
- Marc Genton
- Marco Canini
- Mark Tester
- Markus Hadwiger
- Mario Lanza
- Martin Mai
- Matteo Parsani
- Matteo Ravasi
- Matthew McCabe
- Meriem Taous Laleg
- Min Suk Cha
- Mohamed Eddaoudi
- Mohamed Elhoseiny
- Mohammad Younis
- Nikos Hadjichristidis
- Noredine Ghaffour
- Omar Knio
- Omar Mohammed
- Panos Kalnis
- Pascal Saikaly
- Pedro Castano
- Peter Richtarik
- Peter Wonka
- Pierre Magistretti
- Raphael Huser
- Raul Tempone
- Ravi Samtaney
- Robert Hoehndorf
- Rod Wing
- Samir Hamdan
- Shuyu Sun
- Sigurdur Thoroddsen
- Slim Alouini
- Stefaan Dewolf
- Stefan Arold
- Suk Chung
- Tadeusz Patzek
- Takashi Gojobori
- Tareq AlNaffouri
- Tariq AlKhalifa
- Udo Schwingenschloegl
- Valerio Orlando
- Vladimir Bajic
- William Roberts
- Xiangliang Zhang
- Xiaohang Li
- Xin Gao
- Xixiang Zhang
- Ying Sun
- Yu Han
- Zhiping Lai

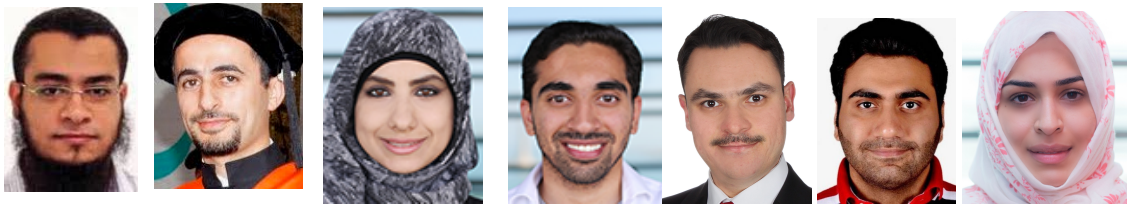
Does not include adjuncts

Hatem Ltaief
Bilel Hadri



Ahmad Abdelfattah
Ali Charara
Dalal Sukkari
Mohammed Farhan
Kadir Akbudak
Mustafa Abduljabbar
Rabab Alomairy

Hatem Ltaief
Bilel Hadri



Join us at Booth #1643

kaust.edu.sa/SC19

SC19

Denver, CO | hpc is now.

Satoshi Matsuoka
Director of the RIKEN Center for Computational Science

Jack Dongarra
Director of the Innovative Computing Center

David Keyes
Director of the Extreme Computing Research Center

Using Mixed Precision in Numerical Computations to Speedup Linear Algebra Solvers

Jack Dongarra, UTK/ORNL/U Manchester
Azzam Haidar, Nvidia
Nick Higham, U of Manchester
Stan Tomov, UTK

05 July 2020

Outline

1. Introduction
2. Nonlinear Elimination Preconditioned Inexact Newton Algorithms (NEPIN)
3. Multiplicative Schwarz Preconditioned Inexact Newton algorithm (MSPIN)
4. Adaptive Use of Nonlinear Preconditioning
5. Approximate error bounds on solutions of Nonlinearly Preconditioned PDEs
6. Conclusions

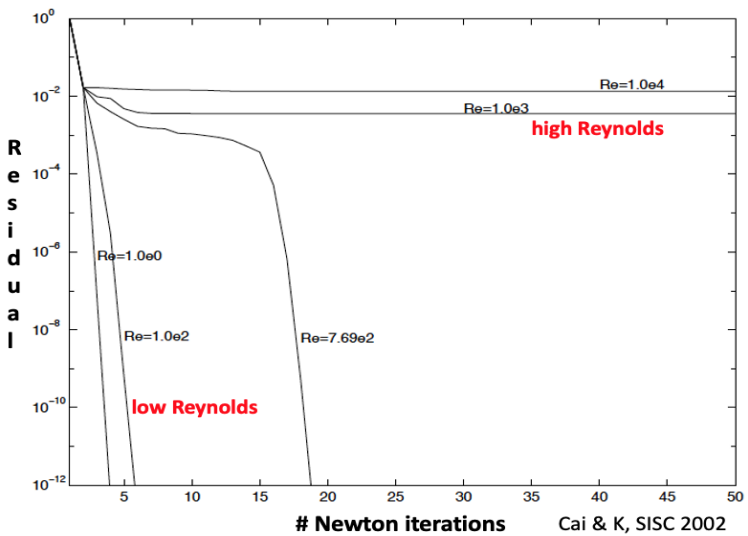
Introduction

Consider a nonlinear problem $F(x) = 0,\ F : \hat{D} \subset R^n \to R^n$.

Taking the Taylor expansion,

$$\boxed{F(x) = F(x_k) + F'(x_k)(x - x_k)} + O(\|x - x_k\|^2).$$

- When high-order terms dominate, the linear model is not a suitable approximation to $F(x)$.
- Strong nonlinearities result in a long plateau period of the residual $\|F(x_k)\|$.
- Only a small number of components of the solution may undergo significant updates in Newton corrections that are highly damped by linesearch backtracking or trust region globalization.



Newton methods may thus waste considerable computational resources solving global linear systems in problems that are "nonlinearly stiff" until they find the convergence domain. Examples include shocks, combustion fronts, recirculation bubbles, etc.

Enter nonlinear preconditioning

- A nonlinear "preconditioner" performs nonlinear relaxation within one or more subspaces, inside the context of an outer Newton method "accelerator."
  - Analogous to linear preconditioners, such as domain decomposition or multigrid, inside a Krylov accelerator.
  - Preconditioned Krylov methods are often used inside *both* the nonlinear subproblems and the global problem.
- A prime consideration in selecting a nonlinear preconditioner is whether the resulting outer problem is amenable to linear preconditioning.
- Left nonlinear preconditioners ASPIN and MSPIN complicate linear preconditioning by replacing a sparse Jacobian with a dense one.
- Right preconditioners like INB-NE retain the original Jacobian.
- Left preconditioner NEPIN (introduced here) can also employ the original Jacobian.

Nonlinear preconditioning

- **Left preconditioning:** solve equivalent system with better balanced nonlinearities

$$\mathcal{F}(x) = G(F(x)) = 0,$$

  - e.g., Additive Schwarz Preconditioned Inexact Newton (ASPIN), Cai & K (SISC, 2002), Multiplicative Schwarz Preconditioned Inexact Newton (MSPIN), Liu & K (SISC, 2015), Restricted Additive Schwarz Preconditioned Exact Newton (RASPEN), Dolean *et al.* (SISC, 2016), Nonlinear Elimination Preconditioning Inexact Newton (NEPIN), Liu *et al.* (2021, submitted)

- **Right preconditioning:** start from a better initial guess by correction within a subspace:

$$F(G(\tilde{x})) = 0, x = G(\tilde{x}),$$

  - e.g., Nonlinear FETI-DP and BDDC, Klawonn *et al.* (SISC, 2014), Nonlinear Elimination (NE), Hwang *et al.* (Comp & FI, 2015), Luo *et al.* (SISC, 2020)

Co-authors and references for this talk

2015	Liu & K	Field-split preconditioned inexact Newton algorithms	SISC
2016	Liu & K	Convergence analysis for the multiplicative Schwarz preconditioned inexact Newton algorithm	SINUM
2018	Liu, K & Krause	A note on adaptive nonlinear preconditioning techniques	SISC
2021	Liu & K	Approximate error bounds on solutions of nonlinearly preconditioned PDEs	SISC (to appear)
2021	Liu, Hwang, Luo, Cai & K	A nonlinear elimination preconditioned inexact Newton algorithm	(submitted)

Short-cuts for a 10-minute peek

- We illustrate on standard 2D PDE models
  - transonic potential flow over an airfoil
  - velocity-vorticity incompressible Navier-Stokes in a cavity
  - velocity-vorticity-energy incompressible Boussinesq in a cavity
- Some other applications, not discussed here:
  - porous media flows, arterial flows, two-phase flows, combustion
- Discretizations are suppressed, being primitive
  - second-order finite differences
  - 2-point upwinding from Boeing in transonic potential example
- Derivations are suppressed
  - please see references
- Parallel scaling and parameter tuning are suppressed
  - PETSc, on KAUST's Shaheen Cray SC-40
  - nonlinear preconditioning has imbalance issues not yet addressed in our software, but Newton-Krylov scaling is decent within a subproblem or outer Newton iteration
  - inexact Newton uses loose linear convergence tolerances
  - nonlinear convergence tolerances and thresholds for "bad" / "good" component selection can be nontrivial

A simple algebraic example in R^2

$$F(x_1, x_2) = \begin{bmatrix} F_1(x_1, x_2) \\ F_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} (x_1 - x_2^3 + 1)^5 - x_2^5 \\ x_1 + 2x_2 - 3 \end{bmatrix} = 0 \quad (1)$$

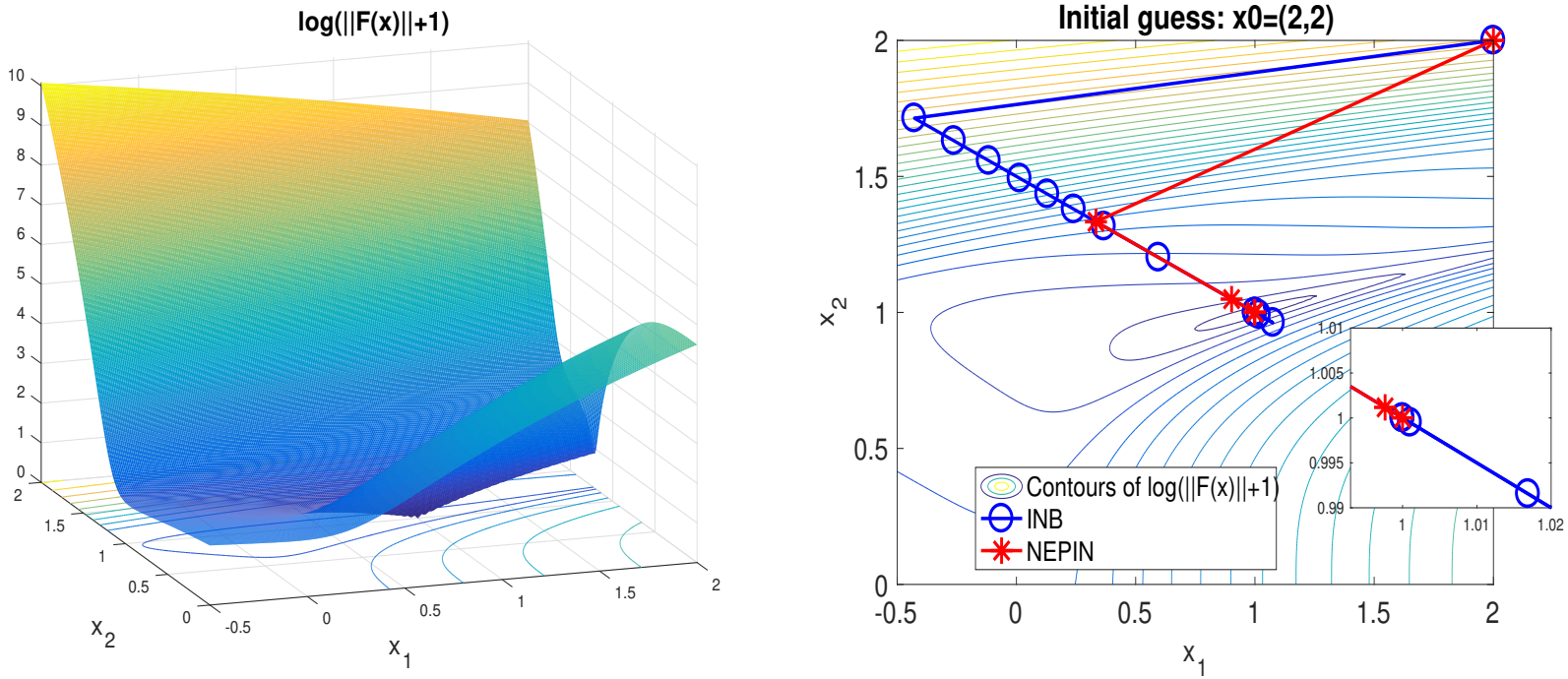


Figure: Contours of $\log(\|F(x)\| + 1)$ for and the path using Inexact Newton with Backtracking (INB) (blue circles) and Nonlinear Elimination Preconditioning Inexact Newton (NEPIN) (red stars) from the same starting point $x^0 = [2, 2]^T$.

Outline

1. Introduction
2. Nonlinear Elimination Preconditioned Inexact Newton Algorithms (NEPIN)
3. Multiplicative Schwarz Preconditioned Inexact Newton algorithm (MSPIN)
4. Adaptive Use of Nonlinear Preconditioning
5. Approximate error bounds on solutions of Nonlinearly Preconditioned PDEs
6. Conclusions

Nonlinear Elimination PIN for unbalanced nonlinearities

The components of the nonlinear system $F(x) = 0$ are partitioned heuristically into two groups, "bad" and "good," labeled as F_b and F_g, respectively, according to the degree of nonlinear "stiffness." This is often successfully associated with the components whose absolute residual exceeds some threshold, or for which some physical feature exceeds some threshold. The unknowns principally associated with each equation are split conformally into $x = [x_b, x_g]^T$:

$$F(x) = F(x_b, x_g) = \begin{bmatrix} F_b(x_b, x_g) \\ F_g(x_b, x_g) \end{bmatrix}, \tag{2}$$

where x_b and x_g are "bad" and "good" components, respectively.

Nonlinear preconditioned function

For a given partitioning, the nonlinear elimination preconditioned function

$$\mathcal{F}(x) = \mathcal{F}(x_b, x_g) = \begin{bmatrix} T_b(x_b, x_g) \\ F_g(x_b, x_g) \end{bmatrix} = \begin{bmatrix} T_b(x) \\ F_g(x) \end{bmatrix}, \tag{3}$$

is obtained by solving the subsystem

$$F_b(x_b - T_b(x), x_g) = 0. \tag{4}$$

for $T_b(x)$. The Jacobian of $\mathcal{F}(x)$ can be written in the form of

$$\mathcal{J}(x) = \begin{bmatrix} \left(\frac{\partial F_b}{\partial u_b}\right)^{-1} & \\ & I_g \end{bmatrix} \begin{bmatrix} \frac{\partial F_b}{\partial u_b} & \frac{\partial F_b}{\partial x_g} \\ \frac{\partial F_g}{\partial x_b} & \frac{\partial F_g}{\partial x_g} \end{bmatrix}, \quad \text{where} \quad u_b = x_b - T_b(x). \tag{5}$$

Then the Newton correction step

$$\mathcal{J}(x)\hat{d} = \mathcal{F}(x) = \begin{bmatrix} T_b(x) \\ F_g(x) \end{bmatrix} \tag{6}$$

is equivalent, upon multiplying the upper block row through by $J_b = R_b J(u_b, x_g) R_b^T$, to

$$J(u_b, x_g)\hat{d} = \begin{bmatrix} J_b T_b(x) \\ F_g(x) \end{bmatrix}. \tag{7}$$

NEPIN algorithm basic step

1. Solve the subspace problem:

$$F_b(z^{(k)}) = F_b(x_b^{(k)} - T_b^{(k)}, x_g^{(k)}) = 0, \tag{8}$$

2. Form the global residual

$$g^{(k)} = \begin{bmatrix} J_b(z^{(k)})T_b^{(k)} \\ F_g(x^{(k)}) \end{bmatrix}, \quad J_b(z^{(k)}) = R_b J(z^{(k)}) R_b^T. \tag{9}$$

3. Solve inexactly for the Newton direction $d^{(k)}$ in

$$J(z^{(k)})d^{(k)} = g^{(k)}, \quad \text{where} \quad J(x) = F'(x) \tag{10}$$

4. Compute the new approximate solution

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} d^{(k)}. \tag{11}$$

Example: Transonic full potential flow

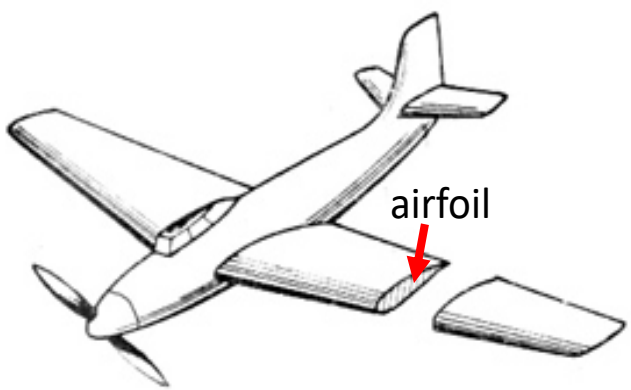
Consider transonic flow around an airfoil, which is described by the scalar full potential equation, derived for inviscid, irrotational, isentropic compressible flow as:

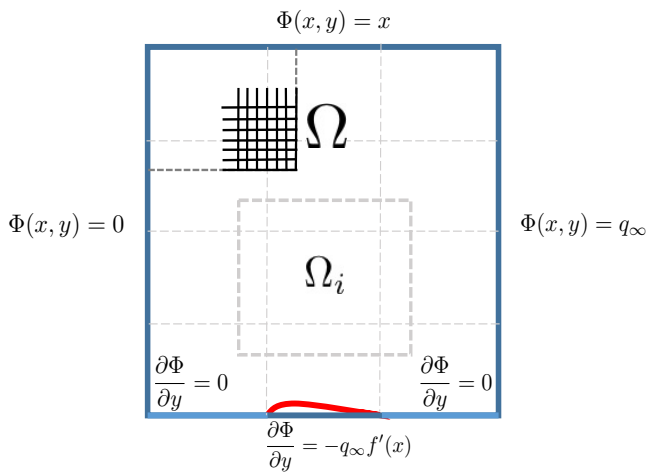
$$\nabla \cdot (\rho(\Phi)\nabla\Phi) = 0, \tag{12}$$

where Φ is the velocity potential, and $\nabla\Phi = [u, v]^T$ is the velocity field. The density function ρ is computed by

$$\rho(\Phi) = \rho_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty^2 \left(1 - \frac{\|\nabla\Phi\|_2^2}{q_\infty^2}\right)\right)^{\frac{1}{\gamma-1}} \tag{13}$$

with suitable upwinding, as in the Boeing TRANAIR code [Young *et al.*, JCP 1991].





Example: Transonic full potential flow

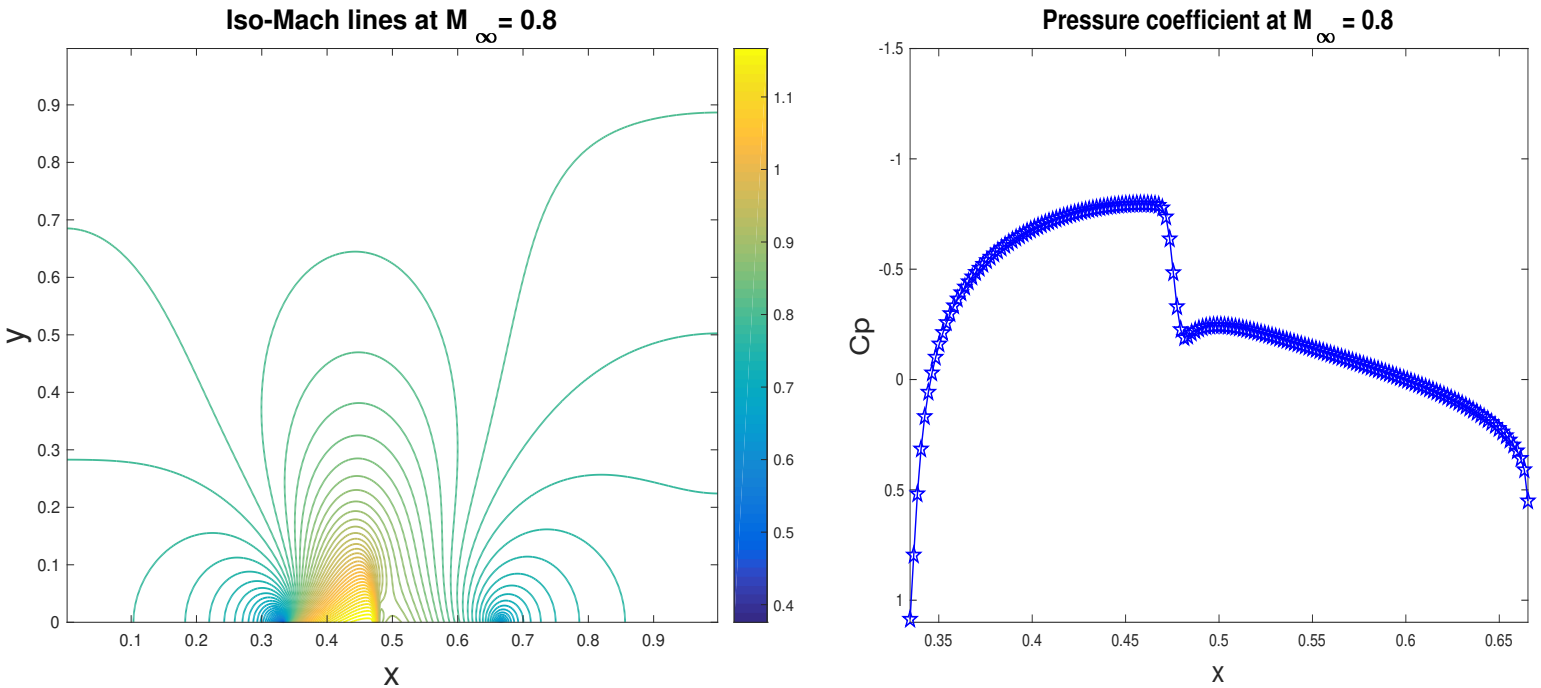


Figure: Mach number countours (left) and the pressure coefficient C_p curve (right) obtained by the final solution at $M_\infty = 0.8$ on a uniform 512×512 mesh.

The distribution of the "bad" components

Define "bad" components as those where the local velocity exceeds a certain cut-off Mach number, $M(x, y) > M_c$.

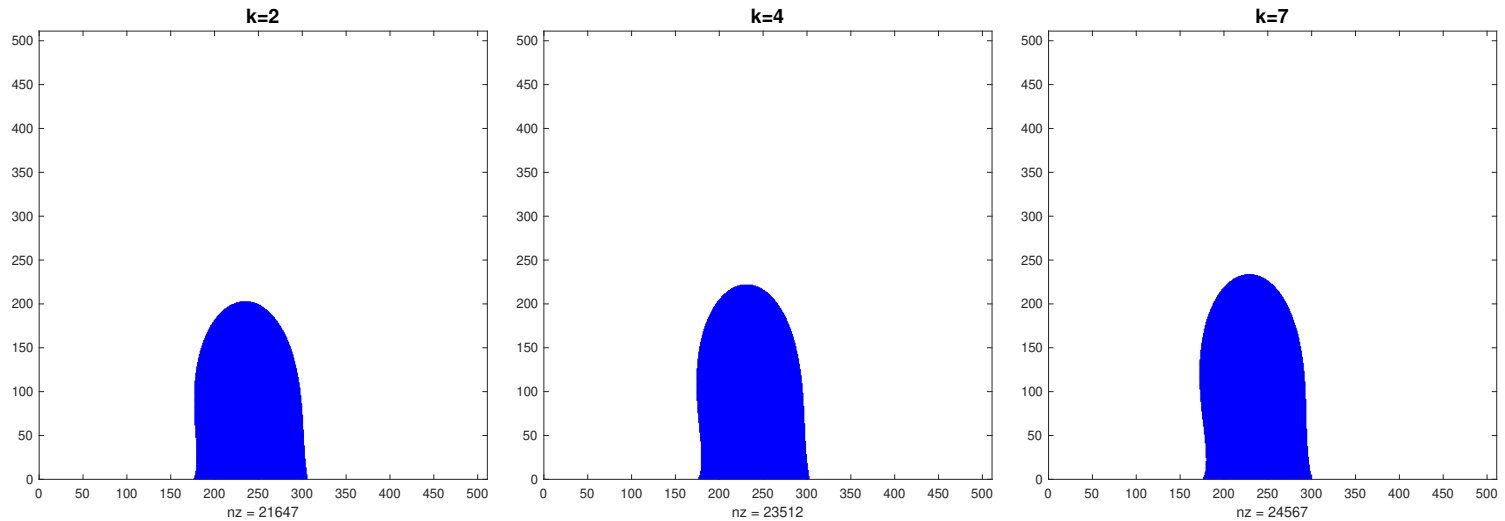
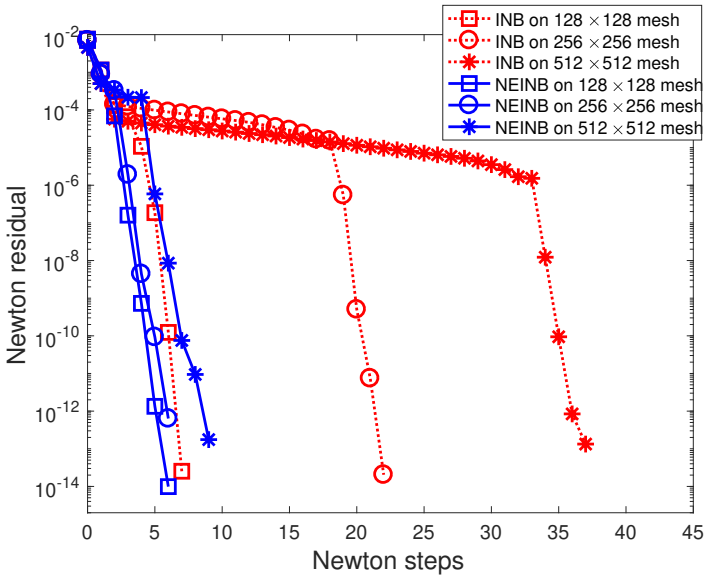
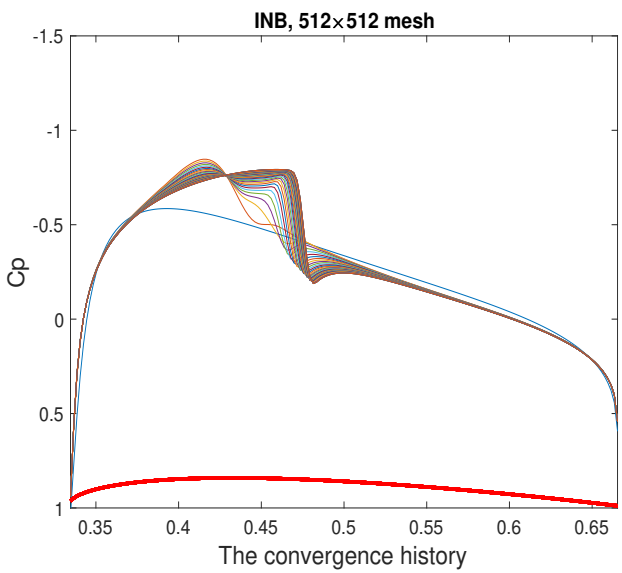
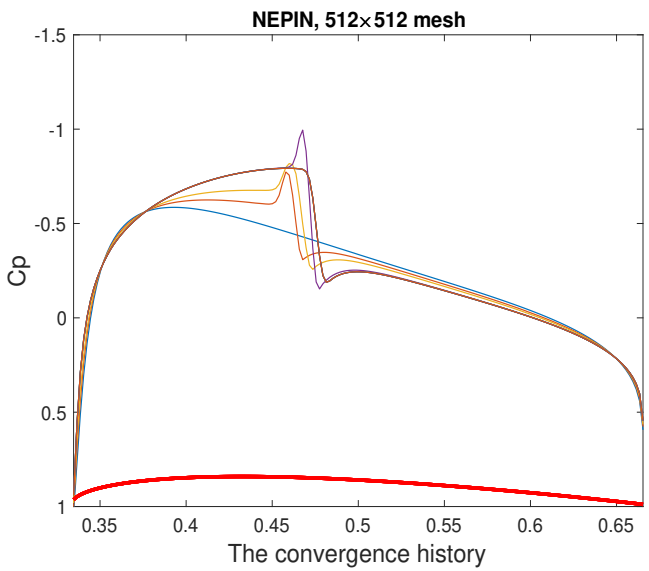


Figure: The evolution of the "bad" component region using NEPIN for $M_\infty = 0.8$ and $M_c = 0.82$, on a uniform 512×512 mesh, on the second, the fourth and the seventh global Newton iterations. Number of bad components: 21,647 at iteration 2; then 23,512 at iteration 4; then 24,567 at iteration 7.

Convergence history for varying mesh size ($M_c = 0.82$)







Outline

1. Introduction
2. Nonlinear Elimination Preconditioned Inexact Newton Algorithms (NEPIN)
3. Multiplicative Schwarz Preconditioned Inexact Newton algorithm (MSPIN)
4. Adaptive Use of Nonlinear Preconditioning
5. Approximate error bounds on solutions of Nonlinearly Preconditioned PDEs
6. Conclusions

Multiplicative Schwarz Preconditioned Inexact Newton algorithm (MSPIN)

A form of nonlinear Gauss-Seidel:

The nonlinear function $F(x)$ is split conformally into 2 nonoverlapping components, representing distinct fields or physical features, as

$$F(x) = F(u, v) = \begin{bmatrix} G(u, v) \\ H(u, v) \end{bmatrix} = 0. \tag{14}$$

The preconditioned system comes from solving subspace nonlinear problems:

$$\mathcal{F}(u, v) = \begin{bmatrix} g(u, v) \\ h(u, v) \end{bmatrix} \tag{15}$$

1. Solve for g in
$$G(u - g, v) = 0$$
2. Solve for h with the new g in
$$H(u - g, v - h) = 0$$

MSPIN, 2-component, non-overlapping

The Jacobian matrix of the preconditioned system is

$$\mathcal{J}(u,v) = \begin{bmatrix} g_u & g_v \\ h_u & h_v \end{bmatrix} = \begin{bmatrix} \frac{\partial G}{\partial p} & \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}, \tag{16}$$

where $p = u - g(u,v)$ and $q = v - h(u,v)$.
In practice, since (p,q) approaches (u,v) as the solution converges locally, the preconditioned Jacobian is locally well approximated by the readily computable

$$\hat{\mathcal{J}}_{MSPIN}(u,v) = \begin{bmatrix} G_p & \\ H_p & H_v \end{bmatrix}^{-1} \begin{bmatrix} G_p & G_v \\ H_p & H_v \end{bmatrix} = \begin{bmatrix} G_p & \\ H_p & H_v \end{bmatrix}^{-1} J(p,v). \tag{17}$$

Generalization to 3 or more components is straightforward.

Example: lid-driven cavity

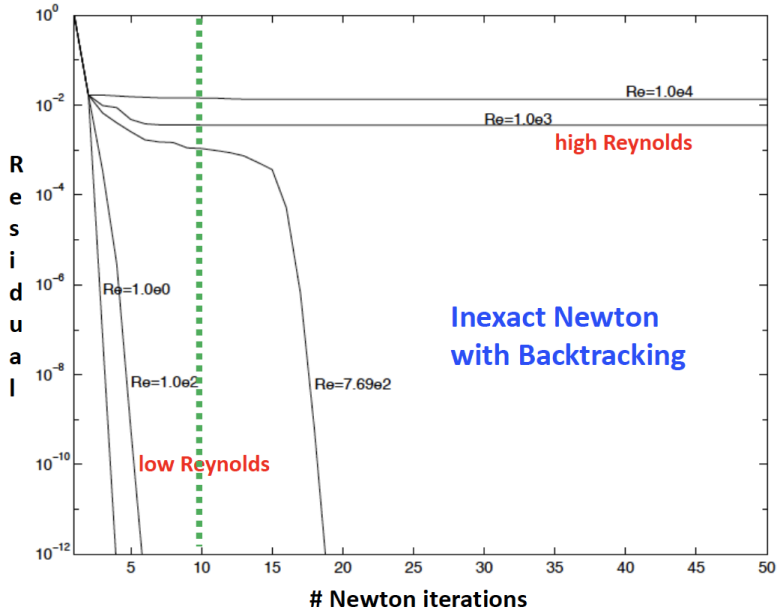
The governing system consists of the nondimensional steady-state incompressible Navier-Stokes equations in vorticity-velocity form:

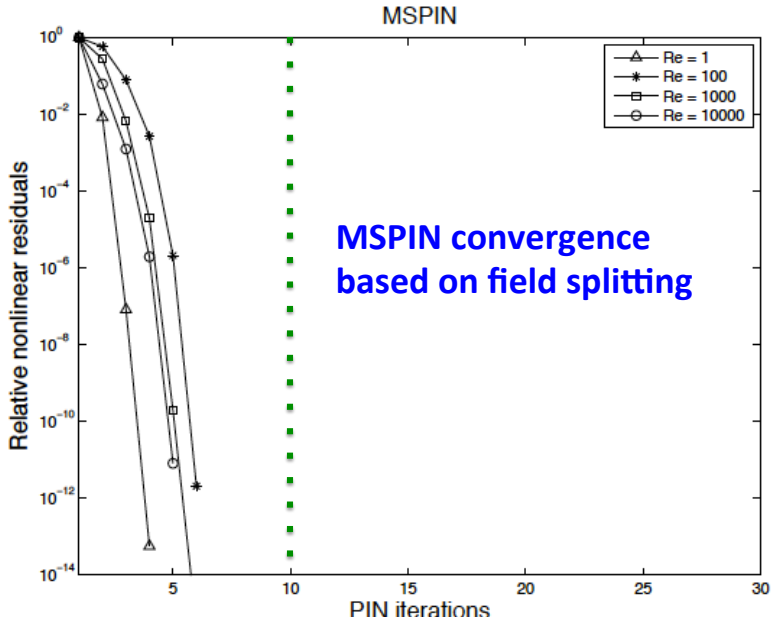
$$\begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \\ -\Delta \omega + Re(u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y}) = 0, \end{cases} \tag{18}$$

There are three unknowns: the velocity fields (u, v) in the (x, y) directions, and the vorticity ω. The parameter Re controls the system's only nonlinearity.

We employ MSPIN as the nonlinear preconditioner with two subsystems: the two velocity equations as one subsystem and the vorticity equation as the other.

Example: lid-driven cavity





Some analysis

[Liu & K, SISC, 2015] $F(x)$ and $\mathcal{F}(x)$ are equivalent in the sense that they have the same solution in a neighborhood of x^* in D.

[Liu & K, SINUM, 2016] MSPIN's local convergence is guaranteed

- superlinear if the forcing tolerance approaches 0
- quadratically if the forcing tolerance approaches 0 like $\mathcal{O}(\|\mathcal{F}(\cdot)\|)$

Outline

1. Introduction
2. Nonlinear Elimination Preconditioned Inexact Newton Algorithms (NEPIN)
3. Multiplicative Schwarz Preconditioned Inexact Newton algorithm (MSPIN)
4. Adaptive Use of Nonlinear Preconditioning
5. Approximate error bounds on solutions of Nonlinearly Preconditioned PDEs
6. Conclusions

Adaptive preconditioning motivation

- Asymptotically, nonlinear preconditioned Newton approaches Newton.
- A preconditioned Newton step is more expensive because of the nonlinear subiterations.
- Nonlinear preconditioning should be "on" only when needed and "off" when not.
- A scalar manipulation of norms available as by-products of the global iterations can be the switch.
- η_k below reflects the agreement between $F(x)$ and its local linear model at the previous step:

$$\eta_k = \frac{\left| \|F(x_k)\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\| \right|}{\|F(x_{k-1})\|}, \quad k = 1, 2, \ldots, \tag{19}$$

Adaptive preconditioning algorithm

Set initial iterate $x^{(0)}$
Set $\eta_0 = 1$ and switch tolerance ϵ
While $k = 0, 1, 2, \ldots$ until convergence
    Update $x^{(k+1)}$ starting from $x^{(k)}$ based on η_k:
    If $\eta_k < \epsilon$
        Implement one step of plain INB
    Else
        Implement one step of nonlinearly preconditioned INB
    EndIf
    Step 2. Compute η_{k+1}
    Step 3. $k \leftarrow k + 1$
EndWhile

Example: lid-driven cavity

We compare the number of nonlinear iterations using MSPIN and MSPIN-adapt at different Reynolds numbers. MSPIN-adapt suspends preconditioning for the terminal step(s), but converges in the same number of Newton iterations as MSPIN.

INB fails to converge from the same "cold" start for large Reynolds numbers.

Number of global Newton iterations				
Algorithm	$Re = 100$	$Re = 1000$	$Re = 5000$	$Re = 10000$
INB	5	-	-	-
MSPIN	4	3	3	3
MSPIN-adapt	4	3	3	3

Example: lid-driven cavity

History of η_k for the lid-driven cavity with $\epsilon = 0.2$.

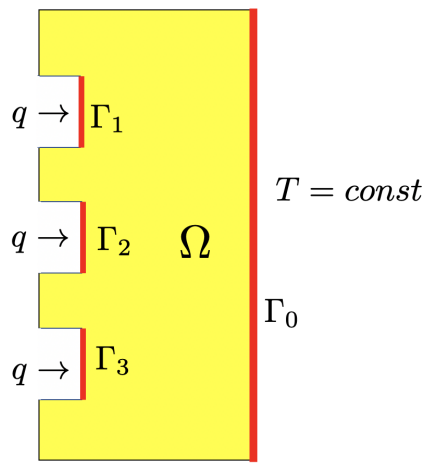
$Re = 100$		
Iter	η_k	step
0	1.0	MSPIN
1	0.580156	MSPIN
2	0.132696	INB
3	0.028887	INB
$Re = 1000$		
Iter	η_k	step
0	1.0	MSPIN
1	0.272078	MSPIN
2	0.025135	INB
$Re = 10000$		
Iter	η_k	step
0	1.0	MSPIN
1	0.061412	INB
2	0.034303	INB

Outline

1. Introduction
2. Nonlinear Elimination Preconditioned Inexact Newton Algorithms (NEPIN)
3. Multiplicative Schwarz Preconditioned Inexact Newton algorithm (MSPIN)
4. Adaptive Use of Nonlinear Preconditioning
5. Approximate error bounds on solutions of Nonlinearly Preconditioned PDEs
6. Conclusions

Motivation

- In applications, we are are often happy with selected functionals of the overall field solution.
- Some of these, particularly integral functionals, may converge faster than the primitive variables.
- Can we directly bound the functionals with by-products of the iteration and stop early?



The problem of cooling electronic components in a computer by natural convection of air in the enclosure. $\partial\Omega / \bigcup_{i=0}^{3} \Gamma_i$ is isolated.

- The field solution $[\boldsymbol{u}, p, T]^T$
- Given an input heat flux q, we wish to determine whether

$$T_{mean} = \int_{\Gamma_1} T ds.$$

is within an acceptable design interval.

Output linear functional of the coupled solution

As in MSPIN, the nonlinear system $F(x)$ is split as

$$F(x) = F(u,v) = \begin{bmatrix} G(u,v) \\ H(u,v) \end{bmatrix} = 0, \quad x = [u,v]^T, \tag{20}$$

We are interested in a linear functional of the coupled solution:

$$J(u,v) = \langle \psi_1, u \rangle + \langle \psi_2, v \rangle, \tag{21}$$

for prescribed $\psi_1 \in R^{n_1}$ and $\psi_2 \in R^{n_2}$.

Approximate error bounds for MSPIN

We can bound the error in the linear functional in terms of the component residuals and some Jacobian blocks

$$\begin{aligned} & J(u,v) - J(\hat{u}^{(k)}, \hat{v}^{(k)}) \\ = \; & \langle \psi_1, u - \hat{u}^{(k)} \rangle + \langle \psi_2, v - \hat{v}^{(k)} \rangle \\ \approx \; & -\langle \psi_1, R_1^{(k)} \rangle + \langle \psi_1, {\mathcal{B}_{11}^{(k)}}^{-1} \mathcal{B}_{12}^{(k)} (I - \mathcal{C}_2^{(k)})^{-1} R_2^{(k)} \rangle \\ & -\langle \psi_2, (I - \mathcal{C}_2^{(k)})^{-1} R_2^{(k)} \rangle, \end{aligned}$$

where $\mathcal{C}_2^{(k)} = {\mathcal{B}_{22}^{(k)}}^{-1} \mathcal{B}_{21}^{(k)} {\mathcal{B}_{11}^{(k)}}^{-1} \mathcal{B}_{12}^{(k)}$.

Approximate error bounds for MSPIN

In the MSPIN algorithm, the submodels are solved sequentially for the physical variable corrections, which implicitly forms the preconditioned system. For any given $x = [u, v]^T \in R^n$, the preconditioned nonlinear system

$$\mathcal{F}(x) = \begin{bmatrix} g(u, v) \\ h(u, v) \end{bmatrix} = 0 \tag{22}$$

is obtained by solving

$$G(u - g, v) = 0, \tag{23}$$

for g. With values of u, v, g, the system

$$H(u - g, v - h) = 0, \tag{24}$$

is solved for h.

Approximate error bounds for MSPIN

Let $p = u - g(u, v)$ and $q = v - h(u, v)$. We define

$$\mathcal{B} = \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix} = \begin{bmatrix} \mathcal{B}_{11} & \mathcal{B}_{12} \\ \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix}. \tag{25}$$

The derivatives of g and h with respect to u and v are written as

$$\frac{\partial g}{\partial u} = I_u, \quad \frac{\partial g}{\partial v} = \mathcal{B}_{11}^{-1}\mathcal{B}_{12}, \tag{26}$$

$$\frac{\partial h}{\partial u} = 0, \quad \frac{\partial h}{\partial v} = I_v - \mathcal{B}_{22}^{-1}\mathcal{B}_{21}\mathcal{B}_{11}^{-1}\mathcal{B}_{12}, \tag{27}$$

where I_u and I_v are the identity matrices that have the same dimension as the u and v blocks, respectively.

Approximate error bounds for MSPIN

Theorem

At the k-th iteration in the MSPIN algorithm, the approximate solution is denoted by $\hat{x}^{(k)} = [\hat{u}^{(k)}, \hat{v}^{(k)}]^T$ and let $\mathcal{C}_2^{(k)} = \mathcal{B}_{22}^{(k)^{-1}} \mathcal{B}_{21}^{(k)} \mathcal{B}_{11}^{(k)^{-1}} \mathcal{B}_{12}^{(k)}$. We define the error induced in the linear functional (21) as

$$\Delta J_k = |J(u, v) - J(\hat{u}^{(k)}, \hat{v}^{(k)})|. \tag{28}$$

Then the approximate error bound is given by

$$\begin{aligned} \Delta J_k \quad &\lesssim \quad |\langle \psi_1, R_1^{(k)} \rangle| + \|(I - \mathcal{C}_2^{(k)})^{-1} R_2^{(k)}\| \|\mathcal{B}_{12}^{(k)^T} \mathcal{B}_{11}^{(k)^{-T}} \psi_1\| \\ &+ \quad \|R_2\| \|(I - \mathcal{C}_2^{(k)})^{-T} \psi_2\|. \end{aligned} \tag{29}$$

If $\|\mathcal{C}_2^{(k)}\| < 1$, we further derive

$$\Delta J_k \lesssim |\langle \psi_1, R_1^{(k)} \rangle| + \|R_2^{(k)}\| (\|\mathcal{B}_{12}^{(k)^T} \mathcal{B}_{11}^{(k)^{-T}} \psi_1\| + \|\psi_2\|) \frac{1}{1 - \|\mathcal{C}_2^{(k)}\|}. \tag{30}$$

Example: buoyancy-driven cavity

The governing system consists of the nondimensional steady-state incompressible Navier-Stokes equations in vorticity-velocity form with Boussinesq bouyancy and the energy equation:

$$\begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \\ -\Delta \omega + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} - Gr\frac{\partial T}{\partial x} = 0, \\ -\Delta T + Pr(u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y}) = 0, \end{cases} \tag{31}$$

There are four unknowns: the velocity fields (u, v) in the (x, y) directions, the vorticity ω and the temperature T.

Example: buoyancy-driven cavity

For this example, we are interested in the following four linear functionals of u, v, ω and T:

$$J_1 = \int_0^1 u(0.5, y)dy, \tag{32}$$

$$J_2 = v_x(0.5, 0.5), \tag{33}$$

$$J_3 = \omega(0.5, 0.5), \tag{34}$$

$$J_4 = \int_0^1 \int_0^1 T dx dy. \tag{35}$$

- J_1 is the mass flux across the vertical line through geometric center of the cavity;
- J_2 is the partial derivative of v along the horizontal line through the center point;
- J_3 is the vorticity at the center point;
- J_4 is the average temperature in the cavity.

Error bounds using MSPIN with different Gr

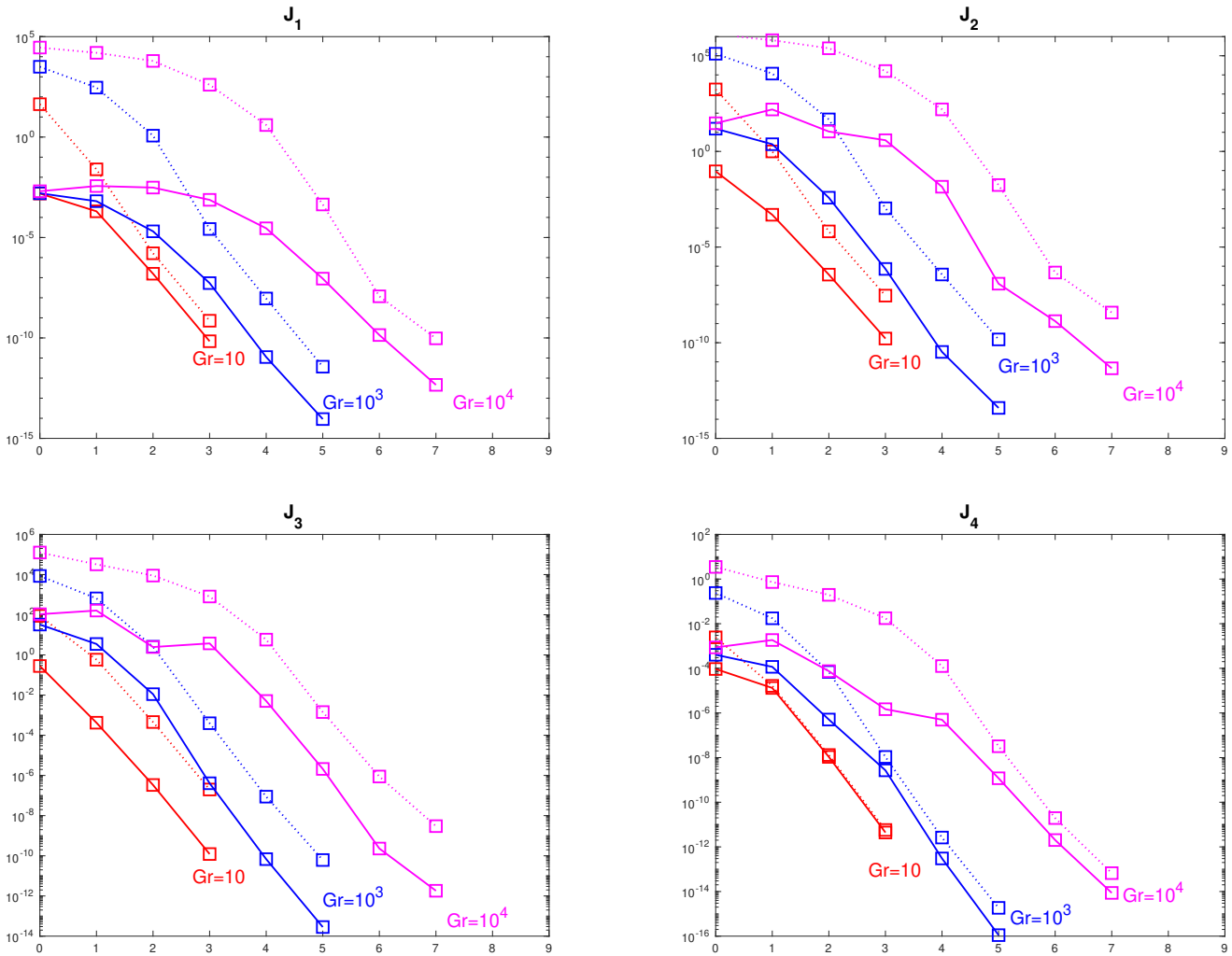


Figure: The absolute errors (solid) in J_1, J_2, J_3 and J_4 with $Gr = 10, 10^3, 10^4$, $Pr = 1$ and the lid velocity $V_{lid} = 0.1$ on uniform 256×256 mesh, with the corresponding error bounds (dotted).

Outline

1. Introduction
2. Nonlinear Elimination Preconditioned Inexact Newton Algorithms (NEPIN)
3. Multiplicative Schwarz Preconditioned Inexact Newton algorithm (MSPIN)
4. Adaptive Use of Nonlinear Preconditioning
5. Approximate error bounds on solutions of Nonlinearly Preconditioned PDEs
6. Conclusions

Conclusions

- Experiments demonstrate that left-preconditioned methods NEPIN (threshold-split) and MSPIN (field-split) can each be effective in improving on the convergence of global Inexact Newton with Backtracking (INB) iterations.
- Nonlinear preconditioning expends extra local computational cost for the solution of nonlinear subproblems to reduce the computation, communication, and synchronization costs of the global outer iterations, by reducing their number.
- A simple adaptive framework is useful to switch nonlinear preconditioning on and off during the outer Newton iterations.
- *A posteriori* approximate error bounds on the linear functionals of interest are available using by-products of the nonlinear preconditioning split systems.

Future Work

- Use of dynamic runtime systems to better sequence the irregularity of the nonlinear subiterations
- More automated identification of "bad" components systems, perhaps using machine learning


Thank you!

Questions