Parallel Implicit Nonlinear Solvers in Large-Scale Computational Science

Lois Curfman McInnes
Computational Scientist
Mathematics and Computer Science Division
Argonne National Laboratory

2009 SIAM Annual Meeting
Denver, CO
July 9, 2009
Acknowledgments

- **U.S. Department of Energy – Office of Science**
  - Base applied math program

- **Collaborators**
  - B. Smith, S. Balay, W. Gropp, D. Kaushik, D. Keyes, M. Knepley, H. Zhang and other PETSc contributors
  - B. Norris, V. Bui, L. Li, R. Armstrong, D. Bernholdt and other TASCS collaborators
Outline

- Motivation
- Preconditioned Newton-Krylov methods
  - Algorithms
  - Software
- Scientific applications
- Ongoing challenges
- Conclusions
A few motivating applications

- Large-scale nonlinear equations
- Solve $F(u) = 0$, where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

Magma dynamics, R. Katz et al.

Reactive transport, P. Lichtner et al.

Bioelectric activity of the heart, L. Pavarino et al.

Core-edge fusion, J. Cary et al.
What are the algorithmic needs?

- Large-scale, nonlinear, PDE-based
  - Multirate, multiscale, multicomponent, multiphysics
  - Rich variety of time scales and strong nonlinearities
  - Ultimately want to do systematic parameter studies, sensitivity analysis, stability analysis, optimization

- Need
  - Fully or semi-implicit solvers
  - Multilevel algorithms
  - Support for adaptivity
  - Support for user-defined customizations (e.g., physics-informed preconditioners)
Target computer architectures

- Systems have increasingly deep memory hierarchy
- Time to reference main memory 100’s of cycles
- Additional complexities
  - multicore, manycore
  - GPUs, FPGAs
  - fault tolerance
  - etc.

IBM BG/P (4 cores)
**TOPS: A SciDAC Center for Enabling Technology**

- TOPS develops, demonstrates, and disseminates robust, quality engineered, solver software for high-performance computers.

Towards Optimal Petascale Simulations

PI: David Keyes, Columbia Univ.
www.scidac.gov/math/TOPS.html
**Overall scope of TOPS**

- **Design and implementation of “solvers”**
  - Linear solvers \( Ax = b \)
  - Eigensolvers \( Ax = \lambda Bx \)
  - Nonlinear solvers with sensitivity analysis \( F(x, p) = 0 \)
  - Time integrators with sensitivity analysis \( f(\dot{x}, x, t, p) = 0 \)
  - Optimizers \( \min_u \phi(x,u) \text{ s.t. } F(x,u) = 0, u \geq 0 \)

- **Software integration**

- **Performance optimization**

Primary emphasis of TOPS numerical software

Indicates dependence
Some popular nonlinear solution strategies

- **Splitting**
  - Often by equation or by coordinate direction
  - Motivated by desire to solve complicated problems with limited computer resources

- **Nonlinear Multigrid Methods**
  - E.g., Full approximation scheme (FAS) - performs relaxation on the full nonlinear problem on each successively coarsened grid

- **Newton-Krylov Methods**
  - Two levels of iteration: Newton on the outside and Krylov on the inside

*our emphasis*
Newton’s method

Based on multivariate Taylor expansion:

\[ F(u^{l+1}) = F(u^l) + F'(u^l)(u^{l+1} - u^l) + \text{higher order terms} \]

\[ F'(u^{l-1}) \, \delta u^l = -F(u^{l-1}) \]

\[ u^l = u^{l-1} + \lambda \, \delta u^l \]

- Can achieve quadratic convergence when sufficiently close to solution
- Can extend radius of convergence with line search, trust region, or continuation methods (e.g., pseudo-transient continuation, mesh sequencing)
**Krylov methods**

- Projection methods for solving linear systems, $Ax=b$, using the Krylov subspace

\[ K_j = \text{span}(r_0, Ar_0, Ar_0^2, \ldots, Ar_0^{j-1}) \]

- Require $A$ only in the form of matrix-vector products

- Popular methods include CG, GMRES, TFQMR, BiCGStab, etc.

- In practice, preconditioning typically needed for good performance
Cluster eigenvalues of the iteration matrix (and thus speed convergence of Krylov methods) by transforming $Ax=b$ into an equivalent form:

$$B^{-1}Ax = B^{-1}b \quad \text{or} \quad (AB)^{-1}(Bx) = b$$

where the inverse action of $B$ approximates that of $A$, but at a smaller cost

How to choose $B$ so that we achieve efficiency and scalability? Common strategies include:

- Lagging the evaluation of $B$
- Lower order and/or sparse approximations of $B$
- Parallel techniques exploiting memory hierarchy, e.g., additive Schwarz
- Multi-level methods
- User-defined custom physics-based approaches
The need for derivatives

\[ F'(u^{l-1}) \delta u^l = -F(u^{l-1}) \]

Solve approximately using a preconditioned Krylov method

- Newton-Krylov methods require derivatives in the form of Jacobian-vector products, \( F'(u)v \)
- Also typically require \( F'(u) \) (or a “cheaper” approximation) for use in preconditioning
- Options: Can provide either \( F'(u) \) or \( F'(u)v \) via
  - Analytic code (written by application developer)
  - Sparse finite difference approximation (FD)
  - Automatic differentiation (AD), see [www.autodiff.org](http://www.autodiff.org)
Matrix-free Jacobian-vector products

**Approaches**

- Finite differences (FD)
  - \( F'(x) v = \frac{[ F(x+hv) - F(x)]}{h} \)
  - costs approximately 1 function evaluation
  - challenges in computing the differencing parameter, \( h \); must balance truncation and round-off errors

- Automatic differentiation (AD)
  - costs approx 2 function evaluations, no difficulties in parameter estimation
  - e.g., ADIFOR & ADIC

**Advantages**

- Newton-like convergence without the cost of computing and storing the true Jacobian
- In practice, still typically perform preconditioning

**Reference**

Complementary TOPS nonlinear solver libraries

- **PETSc: SNES**
  - Scalable Nonlinear Equations Solvers

- **SUNDIALS: KINSOL**
  - [computation.llnl.gov/casc/sundials/](http://computation.llnl.gov/casc/sundials/)
  - Based on NKSOL

- **Trilinos: NOX**
  - [trilinos.sandia.gov/packages/nox/](http://trilinos.sandia.gov/packages/nox/)
  - Nonlinear Object Oriented Solutions

Primary emphasis of TOPS numerical software
Features of TOPS nonlinear solvers

- Emphasize Newton-Krylov methods
- Physicists want to concentrate on physics instead of solvers
  - Express nonlinear solver tasks at a level of mathematical abstraction
  - Exploit state-of-the-art linear solvers as these evolve under the interface
  - Run the same code on laptops, networks of workstations, and leadership-class machines
- Bonus: Sensitivity, optimization, parameter estimation, boundary control require the ability to apply the inverse action of the Jacobian: available in all Newton-like implicit methods
PETSc Background

- Supported “research” code
- Free for everyone, including industrial users
- Extensive documentation, many tutorial-style examples
- Support via email: petsc-maint@mcs.anl.gov
- Usable from Fortran 77/90, C, C++, Python

Long-term goals
- Provide software for the scalable (parallel) solution of algebraic systems arising from PDE-based problems
- Support interfaces to other solver packages (TOPS and more)
- Provide the building blocks for scalable optimization and eigenvalue computations
- Eliminate the MPI from MPI programming!
# PETSc numerical libraries

## Nonlinear Solvers
- Newton-based Methods
- Line Search
- Trust Region
- Others

## Time Steppers
- Euler
- Backward Euler
- Pseudo Time Stepping
- Others

## Krylov Subspace Methods
- GMRES
- CG
- CGS
- Bi-CG-STAB
- TFQMR
- Richardson
- Chebychev
- Others

## Preconditioners
- Additive Schwartz
- Block Jacobi
- Jacobi
- ILU
- ICC
- LU
- Redundant
- Others

## Matrices
- Compressed Sparse Row (AIJ)
- Blocked Compressed Sparse Row (BAIJ)
- Symmetric BAIJ (SBAIJ)
- Dense
- Matrix-free
- Others

## Distributed Arrays

## Vectors

## Index Sets
- Indices
- Block Indices
- Stride
- Others
Features of PETSc/SNES

- Preconditioned Newton-Krylov methods
  - Line search and trust region globalization strategies
  - Eisenstat-Walker approach for linear solver convergence tolerance

- Uses high-level abstractions for matrices, vectors, linear solvers
  - Easy to customize and extend, facilitates algorithmic experimentation
    - Supports matrix-free methods
  - Jacobians available via application, Finite Differences (FD) and Automatic Differentiation (AD)

- Application provides to SNES
  - Residual: PetscErrorCode (*func) (SNES snes, Vec x, Vec r, void *ctx)
  - Jacobian (optional): PetscErrorCode (*func) (SNES snes, Vec x, Mat *J, Mat *M, MatStructure *flag, void *ctx)

- DMComposite: New support for multiphysics problems, see B. Smith, MS69, July 9, 5:00 pm
Outline

- Motivation
- Preconditioned Newton-Krylov methods
  - Algorithms
  - Software
- Scientific applications
- Ongoing challenges
- Conclusions
Application perspective on SNES

Solve \( F(u) = 0 \): Fully implicit matrix-free Newton-Krylov methods

Application Driver
(+ Timestepping + Parallel Partitioning)

Nonlinear Solvers (SNES)

Preconditioners
- SSOR
- ILU
- B-Jacobi
- ASM
- Multigrid
- Others...

Krylov Solvers
- GMRES
- TFQMR
- BCGS
- CGS
- BCG
- Others...

Matrices
- AIJ
- BAIJ
- SBAIJ
- Dense
- Matrix-free
- Others...

Vectors
- Sequential
- Parallel
- Others...

Function Evaluation

Jacobian Evaluation

PETSc

Application Initialization

Post-Processing

Application code
PETSc code
Application or PETSc for Jacobian

Argonne National Laboratory

L. C. McInnes, SIAM Annual Meeting, July 9, 2009
SNES usage: bioelectric activity of the heart

- **Developers:** Luca Pavarino (University of Milan, Italy) et al.

- **Background:** Reaction-diffusion system of degenerate parabolic PDEs

- **Discretization:** Finite elements in space + implicit (decoupled) time discretizations lead to $F(u) = 0$

- **Solvers:** Bidomain Newton-Krylov-Schwarz (multilevel overlapping Schwarz research)

- **References:**
SNES usage: magma dynamics

- **Developers:** R. Katz (U. of Oxford), M. Spiegelman (Columbia U.), assistance with PETSc issues by M. Knepley and B. Smith
- **Background:** Plate tectonics is linked to volcanism; continuum approach for magma dynamics: mantle convection + magmatic flow + phase transitions
- **Discretization:** Finite volume in space + semi-Lagrangian discretiz. of Lagrangian time derivatives lead to $F(u) = 0$
- **Solvers:** Newton-Krylov-Schwarz
- **References:**
Simulation of magmatic reactive flow

Magma “corrodes” mantle as it rises and becomes undersaturated in SiO$_2$. Movie courtesy of R. Katz.
SNES usage: reactive groundwater flow & transport

- **SciDAC project: PFLOTRAN**
  - PI P. Lichtner (LANL)
  - [https://software.lanl.gov/pflotran](https://software.lanl.gov/pflotran)
  - **Overall goal:** Continuum-scale simulation of multiscale, multiphase, multicomponent flow and reactive transport in porous media; applications to field-scale studies of geologic CO2 sequestration, contaminant migration

- **Model:** Fully implicit, finite volume discretization, multiphase flow, geochemical transport
  - Initial TRAN by G. Hammond for DOE CSGF practicum
  - Initial FLOW by R. Mills for DOE CSGF practicum
  - Initial multiphase modules by P. Lichtner and C. Lu

- **PETSc usage:** Preconditioned Newton-Krylov algorithms + parallel structured mesh management (B. Smith)
PFLOTRAN governing equations

Mass conservation: flow equations
\[
\frac{\partial}{\partial t} (\phi s_\alpha \rho_\alpha X_\alpha) + \nabla \cdot \left[ q_\alpha \rho_\alpha X_\alpha - \phi s_\alpha D_\alpha \rho_\alpha \nabla X_\alpha \right] = Q_\alpha
\]
\[
q_\alpha = -\frac{kk_\alpha}{\mu_x} \nabla (p_\alpha - W_\alpha \rho_\alpha g z)
\]

Energy conservation equation
\[
\frac{\partial}{\partial t} \left[ \phi \sum_\alpha s_\alpha \rho_\alpha U_\alpha + (1 - \phi) \rho_c c, T \right] + \nabla \cdot \left[ \sum_\alpha q_\alpha \rho_\alpha H_\alpha - \kappa \nabla T \right] = Q_e
\]

Multicomponent reactive transport equations
\[
\frac{\partial}{\partial t} \left[ \phi \sum_\alpha s_\alpha \Psi_\alpha \right] + \nabla \cdot \left[ \sum_\alpha \Omega_\alpha \right] = -\sum_m v_{jm} I_m + Q_j
\]

Total concentration
\[
\Psi_\alpha = \delta_\alpha d C_j + \sum_i v_{ji} C_i
\]

Total solute flux
\[
\Omega_\alpha = (-\tau \phi s_\alpha D_\alpha \nabla + q_\alpha) \Psi_\alpha
\]

Mineral mass transfer equation
\[
\frac{\partial \phi_m}{\partial t} = V_m I_m \quad \quad \quad \phi + \sum_m \phi_m = 1
\]

PDEs for PFLOW and PTRAN have general form
\[
\frac{\partial A}{\partial t} + \nabla \cdot F = s
\]

Dominant computation of each can be expressed as:

Solve \( F(u) = 0 \)
Hanford 300 benchmark on Jaguar (Cray XT5 at ORNL)

Modeled uranium plume at the Hanford site; computed on 32,000 cores of the Cray XT5

- 37538 quad-core 2.3 GHz Opteron compute nodes (150152 CPU compute cores)
- Additional nodes to handle OS services (I/O, etc.)
- 1.4 petaflops theoretical peak performance
- 300 terabytes aggregate RAM; 10,000 terabytes parallel disk storage
SNES / PFLOTRAN scalability

- Results courtesy of R. Mills (ORNL). More details + multiphysics issues: See R. Mills, MS69, July 9, 4:30 pm
- Inexact Newton w. line search using BiCGStab + Block Jacobi/ILU(0)
- PETSc/SNES design facilitates algorithmic research: reduced synchronization BiCGStab
SNES usage: FACETS fusion

- SciDAC project: FACETS: Framework Application for Core-Edge Transport Simulations
  - PI John Cary, Tech-X Corp,
  - [https://www.facetsproject.org/facets/](https://www.facetsproject.org/facets/)
  - **Overall goal:** Develop a tight coupling framework for core-edge-wall fusion simulations

- **Initial solvers focus:** Incorporated SNES into
  - **UEDGE** (T. Rognlien et al., LLNL): 2D plasma/neutral transport
  - **New core solver** (A. Pletzer et al., Tech-X)
Nonlinear PDEs in core and edge

**Core:** 1D conservation laws:

\[ \frac{\partial q}{\partial t} + \nabla \cdot F = s \]

where \( q = \{ \text{plasma density, electron energy density, ion energy density} \} \)

\( F = \) fluxes, including neoclassical diffusion, electron/ion temperature, gradient induced turbulence, etc.

\( s = \) particle and heating sources and sinks

**Challenges:** highly nonlinear fluxes

**Edge:** 2D conservation laws: Continuity, momentum, and thermal energy equations for electrons and ions:

\[ \frac{\partial n}{\partial t} + \nabla \cdot (n_{e,i}v_{e,i}) = S_{e,i}^p, \text{ where } n_{e,i} \text{ & } v_{e,i} \text{ are electron and ion densities and mean velocities} \]

\[ nm_{e,i} \frac{\partial v_{e,i}}{\partial t} + m_{e,i}n_{e,i}v_{e,i} \cdot \nabla v_{e,i} = \nabla p_{e,i} + qn_{e,i}(E + v_{e,i} \times B/c) - \nabla \cdot \Pi_{e,i} - R_{e,i} + S_{e,i}^m \]

where \( m_{e,i}, p_{e,i}, T_{e,i} \) are masses, pressures, temperatures

\( q, E, B \) are particle charge, electric & mag. fields

\( \Pi_{e,i}, R_{e,i}, S_{e,i}^m \) are viscous tensors, thermal forces, source

\[ \frac{3}{2} n \frac{\partial T_{e,i}}{\partial t} + \frac{3}{2} n v_{e,i} \cdot \nabla T_{e,i} + p_{e,i} \nabla \cdot v_{e,i} = -\nabla \cdot q_{e,i} - \Pi_{e,i} \cdot \nabla v_{e,i} + Q_{e,i} \]

where \( q_{e,i}, Q_{e,i} \) are heat fluxes & volume heating terms

Also neutral gas equation

**Challenges:** extremely anisotropic transport, extremely strong nonlinearities, large range of spatial and temporal scales

Dominant computation of each can be expressed as nonlinear PDE: Solve \( F(u) = 0, \) where \( u \) represents the fully coupled vector of unknowns
UEDGE: 2D plasma/neural transport

- Edge-plasma region key for integrated modeling of fusion devices
  - Edge-pedestal temperature has large impact on fusion gain
  - Plasma exhaust can damage walls
  - Impurities from wall can dilute core fuel and radiate substantial energy
  - Tritium transport key for safety

- UEDGE features
  - Multispecies plasma; var. \( n_{i,e}, u_{\parallel i,e}, T_{i,e} \) for particle density, parallel momentum, and energy balances
  - Reduced Navier-Stokes or Monte Carlo neutrals
  - Multi-step ionization and recombination
  - Finite volume discretization; non-orthogonal mesh

References:
UEDGE: More complete parallel Jacobian data enables robust solution for problems with strong nonlinearities

- **New capability:** Computing parallel Jacobian using matrix coloring for finite differences
  - More complete parallel Jacobian data enables more robust parallel preconditioners

- **Impact:** Enables inclusion of neutral gas equation (difficult for highly anisotropic mesh, not possible in prior parallel UEDGE approach); useful for cross-field drift cases

---

**UEdge parallel partitioning**

**Missing Jacobian elements**

Previous parallel UEDGE Jacobian (Block Jacobi only)

Recent progress: Complete parallel Jacobian data

**5 equations:** ion density, ion velocity, gas density diffusion, electron temp, ion temp

**8 proc:** Matrix-free Newton w. GMRES: Block Jacobi stagnates; complete Jacobian data enables convergence
Idealized view: Surficial couplings

- Core: collisionless, 1D transport system with local, only-cross-surface fluxes
- Edge: collisional 2D transport system
- Wall: beginning of a particle trapping matrix
Core + edge in FACETS framework

- Beginning simulations of pedestal buildup of DIII-D experimental discharges
- Further details: J. Cary, MS80, July 10, 11:30 am
Software challenges for high-performance computational science

- Exploiting emerging leadership-class systems for multi-model simulations at extreme scale
  - Scalability in problem size/complexity, resources (CPUs, memory)
  - Also software size/complexity/rate of change, human factors

- Software is a team effort
  - Geographically distributed
  - Multidisciplinary

- Software must be robust, but flexible
  - Much larger than any single contributor can deeply understand
  - Much longer lifetime than specific platforms, contributors
  - Must allow effective integration of many contributions
  - Must allow evolution of algorithms, scientific focus
Components help manage software complexity

- Object-oriented techniques are useful for building individual components by relatively small teams; component technologies facilitate sharing of code developed by different groups by addressing issues in
  - **Language interoperability**
    - Via interface definition language (IDL)
  - **Common interfaces**
    - Enable “plug-and-play”
  - **Dynamic composability**

- Can easily convert from an object orientation to a component orientation

TASCS: A SciDAC Center for Enabling Technology

- TASCS leads development of the Common Component Architecture (CCA), a component architecture specially designed for high-performance scientific computing
  - Supports parallel and distributed computing
  - Supports mixed language programming
    - Currently: C, C++, Fortran, Java, Python
  - Support for platforms, data types, etc. important to HPC
  - Support for legacy software

- TASCS Institutions: ANL, LLNL, ORNL, PNNL, SNL, Binghamton U, Indiana U, Tech-X, U Maryland, U Oregon, Virginia State U
TOPS (non)linear solver components

- **Components**
  - Units of software functionality
  - Interact only through well-defined interfaces
  - Can be composed into applications based on their interfaces

- **Ports**
  - Interfaces through which components interact
  - Follow a provides/uses pattern
  - **Provided** ports are implemented by a component
  - **Used** ports are functionality a component needs to call

- **Frameworks**
  - Hold components while applications are assembled and executed
  - Control the connections of ports
  - Provide standard services to components

Wiring diagram from Ccaffeine framework’s GUI (SNL)
TASCS component technology initiatives address new research challenges

■ Emerging HPC Hardware and Software Paradigms
  – Fully harness unprecedented computing power of massively parallel, heterogeneous architectures (multiple levels of parallelism, hybrid hardware, fault tolerance)

■ Software Quality and Verification
  – Lightweight runtime enforcement of behavioral semantics to help in correct software usage

■ Computational Quality of Service (CQoS) & Adaptivity
  – Exploit component automation to help scientists choose among algorithmic implementations & parameters, thereby creating opportunities for enhanced performance
    • collaboration with SciDAC Performance Engineering Research Institute (PERI)

Facilitate enhancements to nonlinear solvers for multidisciplinary applications on emerging leadership-class machines
Summary

- Parallel implicit nonlinear solvers are effective for solving many large-scale PDE-based applications.
- TOPS researchers pay careful attention to:
  - Usability and robustness
  - Portability
  - Algorithmic efficiency (optimality)
  - Implementation efficiency (within a processor and in parallel)
- Users can readily employ TOPS nonlinear solver libraries:
  - Application simply provides code for nonlinear function, $F(u)$
  - Then can extend/customize based on individual priorities.
Summary (cont.)

Parallel implicit solvers offer a foundation for addressing increasing challenges in large-scale computational science:

- Quote from D.E. Keyes: “We [TOPS team] aim to carry users from “one-off” solutions to the full scientific agenda of sensitivity, stability, and optimization (from heroic point studies to systematic parametric studies) all in one software suite.”


Many diverse and challenging research issues

- E.g., exploring nonlinear multigrid issues
- See speakers in MS56, MS69, MS80: Implicit Nonlinear Solvers in Multimodel Simulations
Minisymposium: Implicit Nonlinear Solvers in Multimodel Simulations

Part I: MS56: July 9, 10:30-12:30 am

- 10:30-10:55 Nonlinear Solvers and Their Multiphysics Applications, C. Woodward, LLNL
- 11:00-11:25 Application of Newton-Krylov Methods to the Implicit Solution of Problems in Radiation Hydrodynamics, E. Myra, U of Michigan; D. Swesty, SUNY Stony Brook
- 11:30-11:55 Schur-Complement and Block-Preconditioned Iterative Techniques for Coupled Subsurface Flow and Geomechanics, J. White and R. Borja, Stanford University
- 12:00-12:25 Implicitly Coupled Solvers for the Simulation of Fluid-Structure Interaction, A. Barker and X.-C. Cai, U of Colorado at Boulder

Part 2: MS69: July 9, 4:00-6:00 pm

- 4:00-4:25 Towards Fully-Implicit Parallel Adaptive Solution of Mantle Convection Problems, L. Wilcox, C. Burstedde, O. Ghattas, U of Texas at Austin; M. Gurnis, Caltech; G. Stadler, UT Austin; Eh Tan, Caltech; T. Tu and J. Worthen, UT Austin; Shijie Zhong, U of Colorado at Boulder
- 4:30-4:55 Parallel Implicit Solvers in Multiphase Flow and Reactive Transport in Porous Media, R. Mills, ORNL
- 5:00-5:25 Software Development of Composite Solvers for Electrical Power Systems, B. Smith and H. Zhang, ANL; S. Abhyankar, Illinois Institute of Technology
Minisymposium: Implicit Nonlinear Solvers in Multimodel Simulations (cont.)

Part 3: MS80: July 10, 10:30-12:30 am

- 10:30-10:55 *Algorithms and Software for Multiphysics Computational Nuclear Engineering*, D. Knoll and R. Park, Idaho National Laboratory


- 12:00-12:25 *A Multiscale Preconditioner for Nonlinear Multiphysics Problems in Porous Media*, T. Wildey, U of Texas at Austin; M. Wheeler, UTA; Ivan Yotov, U of Pittsburgh
Thanks again to:

- **SIAM**
- **U.S. Department of Energy – Office of Science**
- **Collaborators**
  - B. Smith, S. Balay, W. Gropp, D. Kaushik, D. Keyes, M. Knepley, H. Zhang and other PETSc contributors
  - B. Norris, V. Bui, L. Li, R. Armstrong, D. Bernholdt and other TASCS collaborators
- **All PETSc users**
Special thanks to James M. Ortega

- Ph.D., Stanford University, 1962
- U of Maryland, Mathematics, 1964-1973
- Director and Founder, ICASE, 1973-1977
- Prof. and Head, Math, NC State, 1977-1979
- Charles Henderson Prof., UVA, 1979-1998
  - Chair, Applied Math and CS, 1979-1984
  - Associate Dean, Engineering, 1980-1982
  - Chair, Applied Math, 1984-1989
  - Director, Institute for Parallel Computation, 1990-1993
  - Chair, Computer Science, 1993-1996
- Professor Emeritus, UVA, 1998-2008
- Directed 19 Ph.D. theses
- Author/co-author of 9 books, 40+ papers

Favorite saying: “Work hard and think smart.”