FASTMath Nonlinear Solver and Time Integration Tools

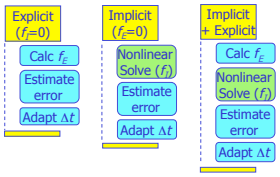
FASTMath Team Leads: Dan Reynolds [SMU], Andy Salinger [SNL], Barry Smith [ANL], Carol Woodward [LLNL]

The diverse systems of nonlinear equations and ODEs that come from science applications require semi-custom solution approaches, which can be composed from a small number of algorithmic building blocks. FASTMath provides the software and expertise in assembling solvers that balance robustness, performance, extensibility, and development effort.

Time Integration Tools

Time dependent problems arise in many contexts, but may typically be considered of the general form:

$$\frac{\partial y}{\partial t} = f_E(t,y) + f_I(t,y)$$



Caption: FASTMath packages provide a range of solvers for time-dependent problems, including explicit, implicit and mixed implicit/explicit (ImEx) approaches. Optionally, some of these may even provide information on sensitivity of the solution to problem parameters.

- The choice of integrator may depend on the time scales present in an application, the stiffness of an application, or even on the availability of optimal solvers for each application component.
- The implicit solve can leverage development of optimal nonlinear/ linear solvers throughout the FASTMath portfolio.

Algorithm Development

Accelerated Fixed-Point Methods for Nonlinear Solvers:

Fixed point iterations are simpler to implement within an application than Newton's method since no Jacobians are required, but they converge only linearly. Recently, an acceleration method by Anderson has proven to be competitive with Newton for some applications. We are equipping our nonlinear solvers and time integrators with robust and flexible options for accelerated fixed point and related methods.

Improved software modularity and agility transition to new architectures:

- Expedite adoption of semi-implicit and implicit algorithms by improving composablity of software, including improved mixed language interfaces
- Adopt architecture-aware kernels as they become available for the problem-independent vector operation underlying many algorithms

Development of time integrators for multi-physics and multi-scale problems:

- Development of explicit, implicit, and ImEx solvers via additive Runge-Kutta methods with automatic error control.
- Development of symplectic integrators via partitioned explicit and Gauss-Legendre implicit Runge-Kutta methods.

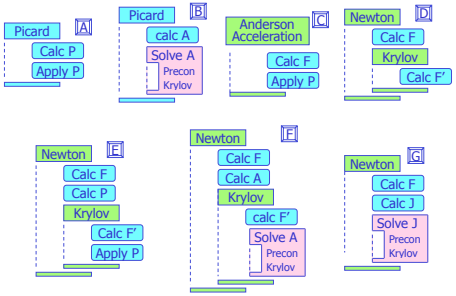
Scalable Differential Variational Inequality (DVI) solvers:

- Development of active set methods for variational inequalities with multigrid preconditioners.
- Coordination with differential algebraic equation (DAE) solvers in PETSc.

Flexible Nonlinear Solution Tools

FASTMath provides a broad array of algorithms and flexible software design. Depending on the needs, capabilities, and priorities of the science application, there are different nonlinear solver options available.

The choice of a nonlinear solver for a science application can depend on a balance between robustness, performance, development time, programming language, extensibility, interface requirements, and availability of derivative information.



Notation: F=Nonlinear Residual; F'=Nonlinear Residual at perturbed solution; J=Jacobian Matrix; A=Approximation to Jacobian; P=User Defined Preconditioning Operator.
Colors: Blue=Application Code; Green=Nonlinear Solver Code; Pink=FASTMath Linear Solvers

Caption: The variety of nonlinear solution procedures available to science applications include [A] Picard iteration; [B] Preconditioned Picard; [C] Anderson Acceleration; [D] Jacobian-Free Newton-Krylov (JFNK); [E] User-preconditioned JFNK; [F] Approximate-Jacobian Preconditioned JFNK; and [G] Newton-Krylov with analytic Jacobian.

Numerous additional capabilities, such as sensitivity analysis, globalization strategies, and performance enhancements are available in the tool sets, as are interfaces to many FASTMath Linear Solvers and preconditioners.

Software Tools and Contacts

SUNDIALS Webpage: computation.llnl.gov/casc/sundials/
POC: Carol Woodward (woodward6@llnl.gov)
POC: Dan Reynolds (reynolds@smu.edu)

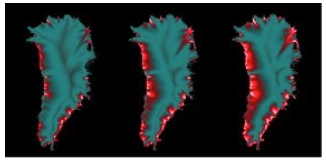
PETSc: Webpage: www.mcs.anl.gov/petsc
POC: Barry Smith (bsmith@mcs.anl.gov)

Trilinos/NOX: Webpage: trilinos.sandia.gov
POC: Andy Salinger (agsalin@sandia.gov)

Use in Applications

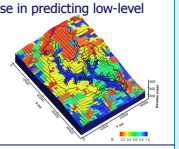
Climate Science: Under the SciDAC SEACISM project, we transitioned the Glimmer Community Ice Sheet Model from a serial code using Picard iteration (algorithm [A]) to a parallel solver from SciDAC-TOPS2 (using algorithm [B]), and then to a preconditioned JFNK (algorithm [F]). Science runs on Jaguar on up to 20000 cores are being used to predict ranges of mass loss of the Greenland Ice Sheet. This new CISM 2.0 code is being incorporated in the CESM (Comm. Earth System Model).

The new PISCEES BER-SciDAC will also use FASTMath solvers and unstructured grid tools and rely on Newton approach (algorithm [G]) and link to UQ technology from the QUEST institute. [Ack: Evans, Price, Lipscomb, Worley, et al.]



Subsurface Science: Using Newton-Krylov with multigrid preconditioners (algorithm [E]), we transitioned the subsurface flow component of the NOAH land-surface model used in WRF to include deep, lateral flow through use of the ParFlow subsurface flow simulator. The more accurate subsurface flow simulations provided detailed information on surface/subsurface feedbacks for use in predicting low-level winds as needed near wind farms [Ack: Maxwell, Kollet].

The coupled ParFlow / CLM (common Land model) simulator demonstrated a proof of concept for regional scale hydrologic simulations of 10^3 km^2 modeled at hydrologic resolutions on over 16,000 processors of a BGL machine. [WRR Featured Paper]

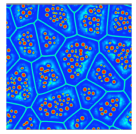


Material Science: Using active set variational inequality solvers with Newton-Krylov-multigrid we have developed scalable algorithms that use phase field models (Allen-Cahn and Cahn-Hillard) to simulate microstructure at the mesoscale for radiation induced damage to materials.

DVIs arise whenever both dynamics and inequalities/switching appear in models producing a mixture of differential equations and variational inequalities

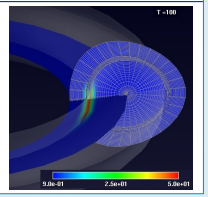
$$\begin{aligned} y' &= f(t, y(t), x(t)) \\ x(t) &\in SOL(K; F(t, y(t), \cdot)) \\ y(0) &= y_0 \end{aligned}$$

$$x \in SOL(K; F(t, y, \cdot)) \Leftrightarrow (\tilde{x} - x)^T F(t, y, x) \geq 0, \forall \tilde{x} \in K$$



Fusion Energy: Under the SciDAC TOPS-2 project, we performed fully implicit simulations of resistive magnetohydrodynamics in both Cartesian and tokamak geometry. Starting with a parallel explicit algorithm we constructed a JFNK algorithm with physics-based preconditioning (algorithm [E]).

Non-preconditioned implicit tests show runtimes 6x faster than explicit, which preconditioning improves by another factor of 4x. [Ack: Samtaney]



More Information: http://www.fastmath-scidac.org, or contact team members identified above.