Accelerated Inexact Newton Method for Casting Simulations

Andrew P. Kuprat (T-1)

This year T-1 implemented a fast nonlinear solver in the 3D Truchas casting simulation code. Truchas is the main product of the ASC-funded TELLURIDE project (a collaboration involving MST, CCS, T, and other divisions). Truchas simulates the entire casting process: flow of molten alloy, heat transfer, solidification of alloy, induced stresses, etc.

Truchas has to solve for the heat transfer and elastic displacement fields at each time step, and for both of these fields this amounts to solving a large nonlinear system of equations

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}\ ,$$

where $\mathbf{f}(\mathbf{x})$ and $\mathbf{x}$ are vectors with dimension of order the number of cells in the simulation—typically hundreds of thousands.

The classic Newton iteration to solve this problem could be written as

Do until done
$\quad \mathbf{u}_i \leftarrow \mathrm{IJ}[\mathbf{f}(\mathbf{x}_i)]$

$\quad \mathbf{x}_{i+1} \leftarrow \mathbf{x}_i - \mathbf{u}_i$

Here, $\mathrm{IJ}[\mathbf{f}(\mathbf{x}_i)]$ means take the residual vector $\mathbf{f}(\mathbf{x}_i)$ and multiply it on the left by the Inverse of the Jacobian of $\mathbf{f}$ evaluated at $\mathbf{x}_i$.

Although Newton's method has the advantage of being rapidly convergent in a neighborhood of the root, it is typically very expensive to evaluate the Jacobian matrix. Indeed, evaluation of the residual function $\mathbf{f}(\mathbf{x})$ and its Jacobian $\mathbf{f}'(\mathbf{x})$ may involve evaluating expensive "subscale" models. For example, if $\mathbf{f}(\mathbf{x})$ is the residual for heat transfer, it may require evaluation of a microscale phase change model if a casting is undergoing solidification. Evaluation of the Jacobian $\mathbf{f}'(\mathbf{x})$ will be in general even more expensive than evaluation of $\mathbf{f}(\mathbf{x})$ itself, and so the classic Newton's method is seen to be relatively expensive.

In the 1990's, Carlson and Miller [1] designed a method, now called the Accelerated Inexact Newton (AIN) Method, that reduced or eliminated the need for evaluation of the exact Jacobian. This method can be described as

Do until done
$\quad \mathbf{u}_i \leftarrow \mathrm{AIJ}[\mathbf{f}(\mathbf{x}_i)]$

$\quad \mathbf{x}_{i+1} \leftarrow \mathbf{x}_i - \mathrm{FPA}(\mathbf{u}_i)$

Here, $\mathrm{AIJ}[\mathbf{f}(\mathbf{x}_i)]$ signifies application of an Approximate Inverse Jacobian matrix to the residual $\mathbf{f}(\mathbf{x}_i)$. This approximate Inverse Jacobian could be an "old" exact Inverse Jacobian evaluated at a point $\tilde{\mathbf{x}}$ close to but not necessarily equal to $\mathbf{x}_i$. In fact, the Approximate Inverse Jacobian may be the application of a "preconditioner" subroutine that doesn't correspond to any exact Inverse Jacobian evaluated at any point whatsoever. The reason this iteration converges is the presence of "FPA" which means "Fixed Point Accelerator."

FPA monitors changes in the inputs $\mathbf{u}_i$ fed to it, and deduces ways to correct the $\mathbf{u}_i$'s in order for the composition FPA ∘ AIJ to produce corrections closer what would have been produced by the true Inverse Jacobian IJ if it had been available.

We have implemented the AIN method for heat transfer and thermomechanics solves, with the AIJ operations being "preconditioners" for the two respective systems. (The preconditioners amount to approximate evaluation of the Jacobian, and then a small number of sweeps of Symmetric Successive Over-Relaxation (SSOR), in order to approximately invert this approximate Jacobian.) The result of implementing the AIN method in Truchas has been a speed-up of the nonlinear heat transfer and thermomechanical displacement solves in both serial and parallel. Speed-ups of up to 4x have been observed when compared to the existing Newton-Krylov method used by the Truchas code. In Figs. 1 and 2, we see elastic displacement field components computed by the Truchas code.

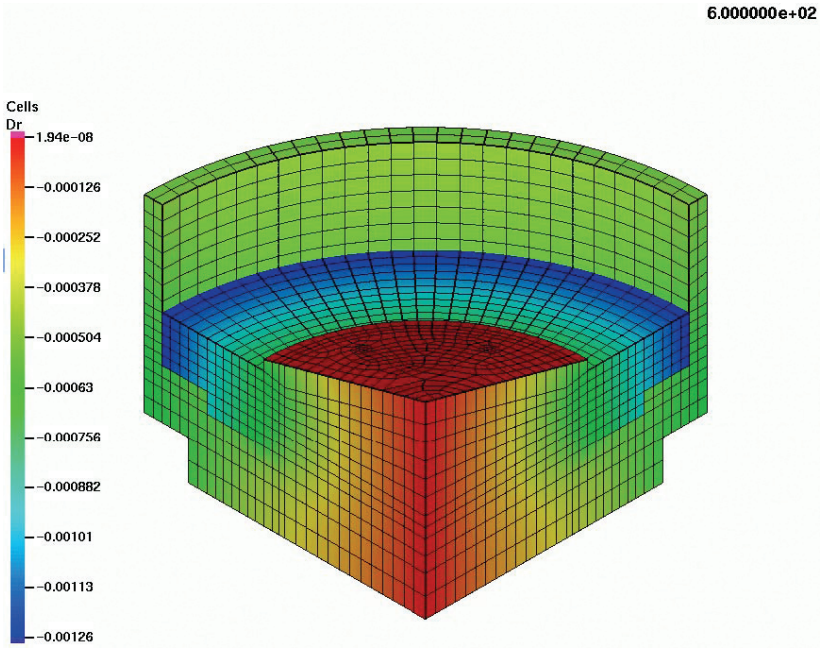

Figure 1— Radial displacement field computed by the Truchas code on a typical metal casting. Courtesy Kin Lam, ESA-WR.

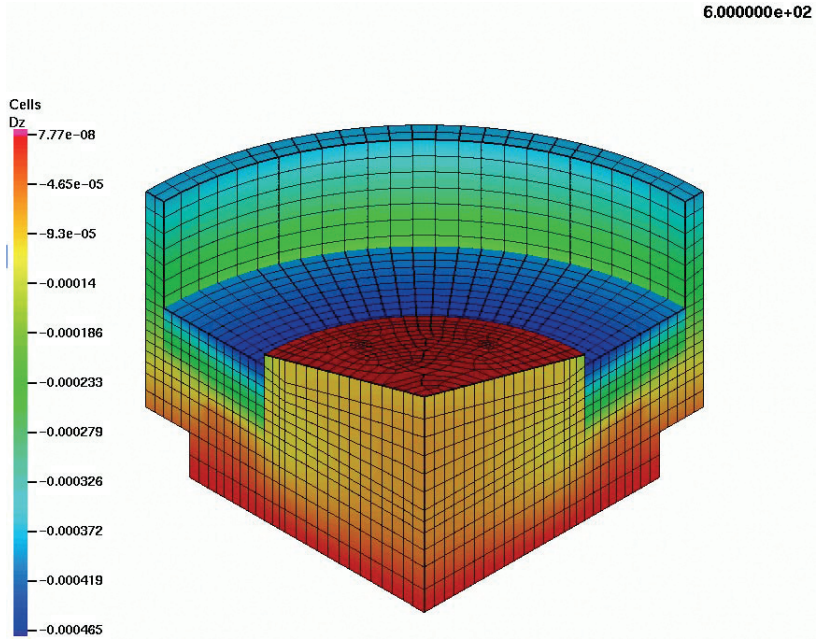

Figure 2— Axial displacement field, same casting.

[1] Neil N. Carlson and Keith Miller, "Design and Application of a Gradient-Weighted Moving Finite Element Code I: In One Dimension," *SIAM J. Sci. Comput.* **19**, 3, pp. 728–765, 1998. (See Section 9.)

For more information, contact Andrew P. Kuprat (kuprat@lanl.gov).

Acknowledgements
I would like to acknowledge NNSA's Advanced Simulation and Computing (ASC), Advanced Applications Program for financial support.