Recent Improvements in LS-DYNA® Metal Forming Material Models

Jinglin Zheng and Xinhai Zhu
Livermore Software Technology Corporation

Abstract

This paper solves two numerical issues arising from the return mapping scheme when simulating metal forming processes with LS-DYNA: (1) the returning mapping process fails to converge in cases where the effective plastic strain increment is small; (2) even when full convergence is accomplished at each time step, the global solution may diverge under small time step settings. For the first issue, a stable iterative scheme with global convergence is implemented to substitute the original secant algorithm which is only locally convergent. For the second issue, a variable tolerance is introduced to control the local error when necessary and hence improve the global convergence. Two examples are given to demonstrate the effectiveness of these methods.

Background

In simulations involving plastic deformations, return mapping is an essential step to determine the final values of stress from a given strain increment. It starts from using the elastic equations and the total strain increment to form a trial stress, which is usually outside of the yield surface and hence inadmissible. Then by correcting the plastic strain increment, the stresses are relaxed back to the updated yield surface and become physically possible [1]. Usually, an iterative algorithm is used to find the correct plastic strain increment $\Delta\varepsilon^p$. Here we simplify this return mapping procedure as finding the root to the following non-linear equation:

$$f(\Delta\varepsilon_p) = 0 \tag{1}$$

where f is a non-linear function that measures the distance of a stress state to the yield surface. In LS-DYNA, the secant algorithm, which is outlined in equation (2), is applied to find the root:

$$\Delta\varepsilon^p_{k+1} = \Delta\varepsilon^p_k - f(\Delta\varepsilon^p_k)\cdot(\Delta\varepsilon^p_k - \Delta\varepsilon^p_{k-1})/(f(\Delta\varepsilon^p_k) - f(\Delta\varepsilon^p_{k-1})) \tag{2}$$

As shown by equation (2), the secant method is a variant of the classical Newton method which uses a linear approximation to replace the derivative term $f'(\Delta\varepsilon^p)$. It works well when f is well-behaved and delivers a decent convergence rate of 1.618. However, it inherits the local convergence nature from Newton's method, i.e., the initial guess has to be close enough to an isolated root for the solution to converge. But this condition is not guaranteed to be satisfied in numerical practice. For example, the initial guess of $\Delta\varepsilon^p$ is commonly set to be the total effective strain increment $\Delta\varepsilon$. But at the initial stage of plastic deformation, $\Delta\varepsilon^p$ may be well below $\Delta\varepsilon$, meaning the initial guess is actually far away from the true solution. In this case, the return mapping is highly likely to fail in convergence. Time step size is another simulation setting that the return mapping scheme is sensitive to and may have problems with. Sometimes even if we have accomplished full convergence at every time step, the solution still blows up when the time step size is too small and round-off error becomes dominant. As a proper time step size is determined from the element size and the sound speed of the material, we do not always have the option of increasing the time step size to avoid such problems. Therefore, this numerical issue also has to be resolved.

In this paper above two issues are fixed with the following approaches: (1) use an iterative scheme with global convergence by combining the secant method with bisection method (i.e., the Dekker's method) in the return mapping process; (2) introduce a variable tolerance criterion which controls the local error and improves the

overall numerical stability. Two examples are given to demonstrate that the implemented approaches indeed resolve these issues.

Iterative return mapping with global convergence

Fig. 1(a) shows a situation in which the original return mapping scheme, i.e., the secant algorithm, fails to converge. As shown, the *f* function tends to stagnate at small $\Delta\varepsilon^p$ which can be caused by limited machine precision. This leads to an extremely small denominator in equation (2). Consequently a substantially large step is taken in the iterative procedure which directs the solution away from the real root which is shown in Fig. 1(b). In Fig. 1(b), the left axis shows the *f* evaluations of each iterate and the right shows the values of iterates. Whenever the iterate gets into flat part indicated in Fig. 1(a), the algorithm makes a big adjustment to the iterate, leading to a large *f* in the next iteration. As a result, *f* keeps bouncing between four different values without a sign of convergence. In practice, LS-DYNA sets a maximum on the number of iterations and exits the secant algorithm with a warning message once the limit is reached, regardless of the convergence.

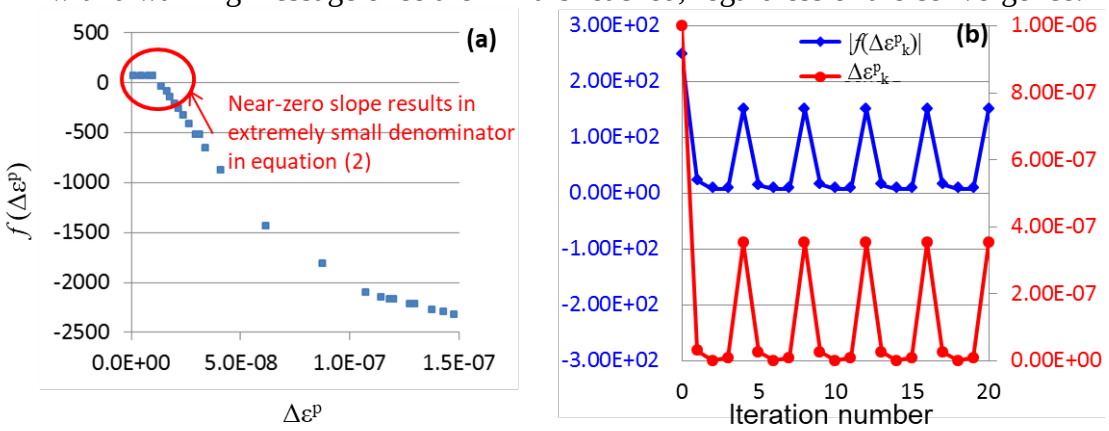


Figure 1 (a) *f* as a function of $\Delta\varepsilon_p$ in the scenario of divergence. (b) The divergence of the return mapping scheme: The red curve shows each iterate and the blue shows the corresponding *f* evaluations. The secant procedure would direct the solution away from the true root whenever the iterate gets into the flat part of the curve indicated in (a). Therefore the iterate keeps bouncing without a sign of convergence.

One key observation from this divergence scenario is that the plastic component of the effective strain increment is small and hence the *f* function is evaluated slightly positive when $\Delta\varepsilon^p = 0$ and becomes negative as $\Delta\varepsilon^p$gets larger. This suggests that we could actually bracket the real root between two iterates $\Delta\varepsilon_a^p$ and $\Delta\varepsilon_a^p$ if $f(\Delta\varepsilon_a^p) \cdot f(\Delta\varepsilon_b^p) < 0$. As long as *f* is continuous, we can then apply the bisection method to shrink down this bracket gradually and reach a satisfactory solution. As for the case plotted in Fig. 1(a), it is obvious that once we find the bracket, the bisection method, which only has a convergence rate of 1, would perform much better than the secant method.

To take advantage of both the efficiency of the secant method and the stability of the bisection method, a natural thought would be that we get started with the secant method and monitor the convergence on the solution. if $\left|\Delta\varepsilon_{k+1}^p - \Delta\varepsilon_k^p\right|$ is not shrinking down in a satisfactory manner, it suggests that the current iterate is not close enough to the real root and we should return back within the bracket and apply bisection for a few iterations to further tight down the search range of $\Delta\varepsilon^p$ such that the secant method would start working again [2]. This gives us Dekker's method and an implementation of this method in our return mapping algorithm can be outlined as follows:

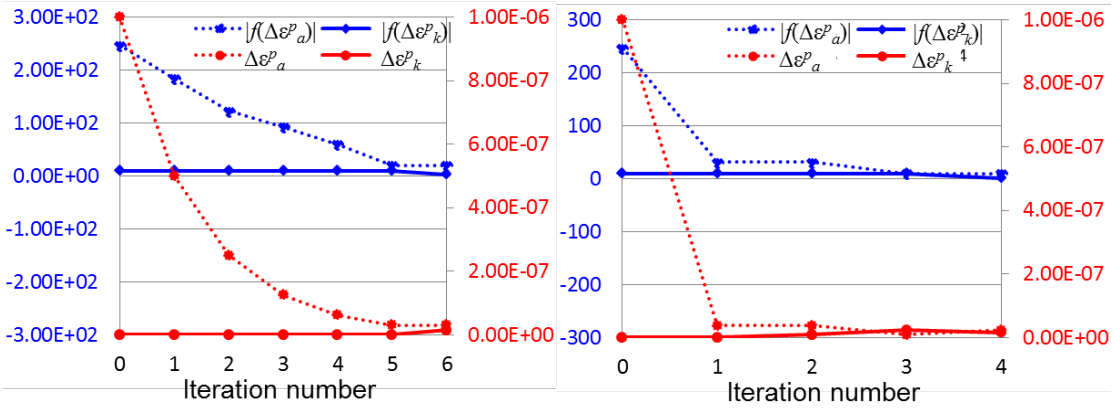
1. Start from $\Delta\varepsilon_0^p = 0$ and $\Delta\varepsilon_1^p = \Delta\varepsilon$ where $\Delta\varepsilon$ is the total effective strain increment at this time step.
2. If $f(\Delta\varepsilon_k^p) \cdot f(\Delta\varepsilon_{k-1}^p) < 0$, set bisection flag and set $\Delta\varepsilon_a^p = \Delta\varepsilon_{k-1}^p$, $\Delta\varepsilon_b^p = \Delta\varepsilon_k^p$.

3. If bisection flag is clear, continue one more step of secant's method and go back to (2). Otherwise:
   a) Swap $\Delta\varepsilon_a^p$ and $\Delta\varepsilon_b^p$, if $\left|f(\Delta\varepsilon_a^p)\right| < \left|f(\Delta\varepsilon_b^p)\right|$.
   b) Calculate the bisection guess $\Delta\varepsilon_m^p = (\Delta\varepsilon_a^p + \Delta\varepsilon_b^p)/2$.
   c) Calculate the secant guess
   $\Delta\varepsilon_s^p = \Delta\varepsilon_k^p - f(\Delta\varepsilon_k^p) \cdot (\Delta\varepsilon_k^p - \Delta\varepsilon_{k-1}^p)/f(\Delta\varepsilon_k^p) - f(\Delta\varepsilon_{k-1}^p)$
   d) If $\Delta\varepsilon_s^p$ falls between $\Delta\varepsilon_m^p$ and $\Delta\varepsilon_b^p$, $\Delta\varepsilon_{k+1}^p = \Delta\varepsilon_s^p$, otherwise, $\Delta\varepsilon_{k+1}^p = \Delta\varepsilon_m^p$.
   e) If $f(\Delta\varepsilon_k^p) \cdot f(\Delta\varepsilon_{k+1}^p) < 0$, set new contra-point $\Delta\varepsilon_a^p = \Delta\varepsilon_k^p$.
   f) $k = k + 1$ and go back to 3(a).

Note that with this method $\left|\Delta\varepsilon_{k+1}^p - \Delta\varepsilon_k^p\right|$ is guaranteed to shrink at least by a factor of 0.5 at each iteration. To get started, we need to run the secant method first until we can securely bracket our solution. However by our observation of the failure scenario, the Dekker's method would initiate from the very first step, namely, $\Delta\varepsilon_0^p = 0$ and $\Delta\varepsilon_1^p = \Delta\varepsilon$ due to the nature of the physical process.

Now we conduct the hot forming simulation on the same model with the Dekker's method being applied in the return mapping process. At exactly the same element where the secant method fails to converge, we are able to achieve convergence within 6 iterations which is shown in Fig. 2(a). Note that in Fig. 2(a) $\left|f(\Delta\varepsilon_k^p)\right|$ and $\Delta\varepsilon_k^p$ are plotted together with $\left|f(\Delta\varepsilon_a^p)\right|$ and $\Delta\varepsilon_a^p$ so that we can see how the bisection bracket is shrinking down as the iteration proceeds. It is clear that at iterations 1-5, bisection is used to ensure stability. The current iterate $\Delta\varepsilon_k^p$ is actually fixed at 0 yet $\Delta\varepsilon_a^p$ keeps going down to narrow down the search range. At iteration 6, we finally lock down our search to a range which is close enough to the real root. And with just one round of secant method, we achieved convergence with a $\left|f(\Delta\varepsilon_k^p)\right|$ smaller than the tolerance.

Dekker's method works well for the problem presented here but may converge much more slowly than the bisection method under certain circumstances. Brent proposed a more bullet-proof method later which has two major improvements from Dekker's method [3]: (1) it uses inverse quadratic interpolation (which is supposed to work better with smooth *f* functions) other than linear interpolation (which is adopted by the secant method); (2) it introduces more tests before the quadratic interpolated value can be accepted as the next iterate. Subjecting this algorithm to the same problem shown in Fig. 1, we are able to reach convergence within 4 iterations, as demonstrated in Fig. 2(b).



(a) Dekker's method (b) Brent's method

Figure 2 (a) Dekker's method: convergence is achieved within 6 iterations with 5 steps of bisection and 1 step of secant. (b) Brent's method: convergence is achieved within 4 iterations with a combination of bisection and inverse quadratic interpolation.

Variable tolerance to improve time step sensitivity

There are cases in which the global solution fails in convergence even when the convergence criterion is met at each time step. Fig. 3 gives a simple example to illustrate this. Consider a dog bone sample under a tensile test, as shown in Fig. 3(a). The sample is subject to a uniaxial tension with the left end fixed and the right end being stretched at a constant rate of 0.02 m/s. A transient mechanical simulation is conducted to solve this simple elastic-plastic deformation problem. For such an analysis, one of the most important settings is the time step size *dt*. As a rule of thumb, we need the time step size to be small enough such that the update of a nodal solution is completed before the node is hit by the wave from neighboring nodes. Therefore, a proper time step size is dependent on the element size in the model, as well as the sound speed of the model material. However, in this case we are actually stuck in a dilemma where the solution becomes unstable when we decrease the time step size for more accuracy. Fig. 3(b) shows the time history of the axial force calculated by LS-DYNA under three different time step sizes: *dt* = 7e-6 s, *dt* = 7e-7 s and *dt* = 7e-8 s. Fig. 3(c) shows the time history of effective plastic strain (ε_p). Because a couple of obvious discontinuities are observed from the curve of *dt* = 7e-6 s in Fig. 3(b), a natural thought is to improve the solution by further reducing *dt*. However, when we decrease *dt* to 7e-7 s and further down to 7e-8 s, both axial force and effective plastic strain solutions blow up, as shown by the red and green curves in Figs. 3(b) and (c).

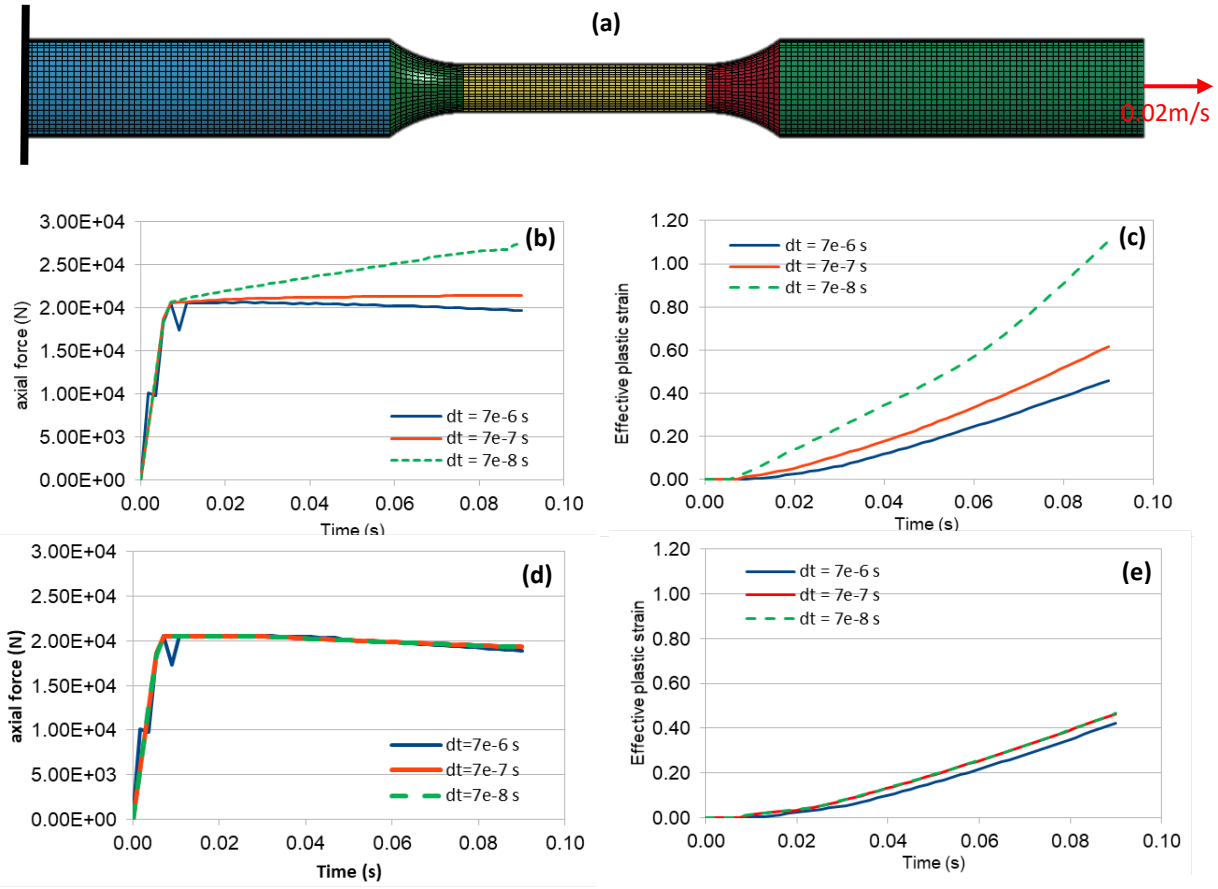


Figure 3 Example uniaxial tension problem (a) Tensile test on a dog bone sample (b) Before fix: time history of axial force under three different time step settings: *dt* = 7e-6 s, 7e-7 s, 7e-8 s (c) Before fix: time history of effective plastic strain under three different time step settings: *dt* = 7e-6 s, 7e-7 s, 7e-8 s (d) After fix: time history of axial force under three different time step settings: *dt* = 7e-6 s, 7e-7 s, 7e-8 s (e) After fix: time history of effective plastic strain under three different time step settings: *dt* = 7e-6 s, 7e-7 s, 7e-8 s

This phenomenon is not rare in numerical practice. In general, the numerical solution is expected to converge to the real one when the time step size decreases. But excessive reduction in time step size will result in the dominance of round-off error which can accumulate and mess up the entire solution. In the specific case of Figs. 3(b) and (c), the problem occurs at the iterative return mapping process, i.e., the process of solving equation (1). The non-linear nature of *f* prohibits us from obtaining an analytical solution. Instead, we use Dekker's method to find an approximate solution. To determine if an approximate solution is good enough, a fixed relative tolerance is applied as the convergence criterion, i.e., the searching process terminates if $\left|f(\Delta\varepsilon_p)\right| \leq tol \cdot Y(\Delta\varepsilon_p)$. Here *tol* is a fixed number (usually *tol* is set to be 0.001) and *Y* is a yield surface function dependent on $\Delta\varepsilon_p$. Because equation (1) is solved numerically, we actually introduces a local error to the mechanical strain at every time step. The precision of the solved $\Delta\varepsilon_p$ is up to the specified tolerance. Extremely small time steps could lead to small strain increment which is likely to generate big round-off errors that is ignored by the fixed tolerance criteria. Accordingly, the accumulated global error can become quite significant and eventually mess up the entire solution, as shown in Figs. 3(b) and (c).

Here we developed a simple fix to this problem by introducing a flexible tolerance which tightens up the accuracy requirement when the local error needs to be controlled for the purpose of convergence. The key is how to determine whether the tolerance needs to be adjusted and how to adjust it. One option is to scale the tolerance with the time step size. However, as mentioned above, an applicable time step size varies with models significantly. Thus it is difficult to find one single reference time step size that works for all models. On the other hand $\Delta\varepsilon_p$ might be a good choice because it reduces with time step size for the same model and unlike the time step size, strain increment itself is a relative measure. The only problem is that $\Delta\varepsilon_p$ is unknown (we are actually trying to solve $\Delta\varepsilon_p$ with the return mapping algorithm). However, we do have a very good candidate to serve as $\Delta\varepsilon_p$ in an iterative scheme, which is the guess of $\Delta\varepsilon_p$.

Figs. 3(d) and (e) show the results when the flexible tolerance is applied in return mapping. As shown in Fig. 3(d), time history of the axial force becomes smooth when *dt* is reduced to 7e-7 s and below. In the meantime, the result converges successfully as *dt* decreases, in contrast to the diverging trend in Fig. 3(b). Similarly, the effective plastic strain curves in Fig. 3(e) also shows a nice trend of convergence as *dt* reduces toward 7e-8 s. Thus the conflict between time step size and numerical stability is resolved.

Conclusion

In this paper, we made two improvements to the iterative return mapping scheme used in metal forming simulations to address the convergence issues. In the process of finding $\Delta\varepsilon_p$, we replaced the original secant method by combining the newton-type scheme with the bisection method. This method effectively fixes the convergence failure encountered by elements with very small $\Delta\varepsilon_p$. Furthermore, we also introduced a flexible tolerance criterion which scales with the initial guess of $\Delta\varepsilon_p$ such that when the time step size needs to be ultrafine to meet the courant criterion, we can still effectively control the local error and achieve a smooth and accurate global solution.

References

1. Plasticity and the Return-Mapping Algorithm: http://mooseframework.org/wiki/PhysicsModules/TensorMechanics/ReturnMap/
2. Uri M. Ascher and Chen Grief, *A First Course in Numerical Methods*, Society for Industrial and Applied Mathematics, 2011
3. Brent's method: https://en.wikipedia.org/wiki/Brent%27s_method