Recent improvements in LS-DYNA® hot stamping simulations

Jinglin Zheng, Xinhai Zhu and Houfu Fan
Livermore Software Technology Corporation, Livermore, CA 94551, USA

Abstract

In this study an improved numerical method is applied in the return mapping scheme of hot stamping simulations to fix the convergence issues with the original Newton-type scheme. Full convergence is achieved in cases where the original solver fails to converge, with a slight increase in computational time. On the other hand, the thermal solution efficiency is improved by adopting parallel computation in two heavily used procedures in the thermal-mechanical coupling analysis when using SMP: (1) temperature interpolation at each mechanical step between two consecutive thermal steps; (2) matrix-vector multiplication in the conjugate gradient solver. This implementation demonstrates computational time savings in thermal steps where it takes many (a few hundred or even more) iterations for the implicit step to converge.

I. Introduction

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 hence inadmissible. Then by correcting the plastic strain increment, the stresses are relaxed back to the updated yield surface and become physically possible. For simple situations, for example, the linear hardening von Mises plasticity case, a closed-form solution can be established. However, for hardening with high non-linearity, an iterative algorithm has to be used to find the correct plastic strain increment. Take material model 106 in LS-DYNA® for example. This is an elastic-visco-plastic material with thermal effects and is widely used in hot stamping simulations where how materials behave under different temperatures and load conditions matters a lot. The user can define a customized three-dimensional table where the stress-strain curves are functions of temperatures and strain rates. This 3-D table dictates material behaviors under hot stamping and can be highly non-linear. To solve for the plastic strain increment in the forming process, LS-DYNA® uses the secant method. This procedure can be simplified as the following non-linear

equation:

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

where $\Delta\varepsilon^p$ is incremental effective plastic strain and *f* is a non-linear function that measures the distance of a stress state to the yield surface. Note that though not explicitly shown, the yield surface itself is dependent on $\Delta\varepsilon^p$ hence has to be updated at every iteration.

The secant method is known as a variant of the classical Newton's method and comes in handy when it is expensive or impossible to evaluate the derivative of the function *f*. Assuming convergence, the derivative of *f* can be approximated by:

$$f'(\Delta\varepsilon^p) \approx \frac{f(\Delta\varepsilon_k^p) - f(\Delta\varepsilon_{k-1}^p)}{\Delta\varepsilon_k^p - \Delta\varepsilon_{k-1}^p} \tag{2}$$

By replacing *f'* in Newton's method with equation (2), we obtain the algorithm of the secant method as equation (3):

$$\Delta\varepsilon_{k+1}^p = \Delta\varepsilon_k^p - f(\Delta\varepsilon_k^p)\frac{\Delta\varepsilon_k^p - \Delta\varepsilon_{k-1}^p}{f(\Delta\varepsilon_k^p) - f(\Delta\varepsilon_{k-1}^p)} \tag{3}$$

As shown by equation (3), this method involves keeping the previous two iterates (and accordingly their *f* evaluations which are usually expensive to obtain) to proceed forward. The secant method delivers a decent convergence rate of 1.618, though not as fast as Newton's method which has a convergence rate of 2.

However, the secant method inherits the local convergence nature from Newton's method. As a result, the initial guess has to be close enough to an isolated root otherwise it will be very difficult for the solution to converge. Unfortunately, this happens a lot in plastic simulations involving return mapping, especially in the case where $\Delta\varepsilon^p$ is small as compared to the initial guess (the total effective strain increment). Fig. 1(a) shows a situation in which the secant algorithm fails under such circumstances. As shown, the *f* function tends to stagnate at small $\Delta\varepsilon^p$ which can be caused by limited machine precision or ill-behaved *f* defined by the material model. Either way this leads to an extremely small denominator in equation (3). 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.

II. A stable numerical scheme for return mapping

Here we aims at fixing the return mapping convergence issue from the root cause, the local convergence nature with Newton type schemes. One key observation from this divergence scenario is that the plastic component of the effective strain increment is small hence the f function is evaluated slightly positive when $\Delta\varepsilon^p = 0$ and becomes negative as $\Delta\varepsilon^p$ gets larger (which is physically impossible). This suggests that we could actually bracket the real root between two iterates $\Delta\varepsilon_a^p$ and $\Delta\varepsilon_b^p$ if $f(\Delta\varepsilon_a^p)\cdot f(\Delta\varepsilon_b^p) < 0$. As long as the 3-D table is continuous and our yield function is continuous, we can then apply the bisection method, which is well-known for its global convergence, to shrink down this bracket gradually. In this way, we are guaranteed with convergence to the real root. 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.

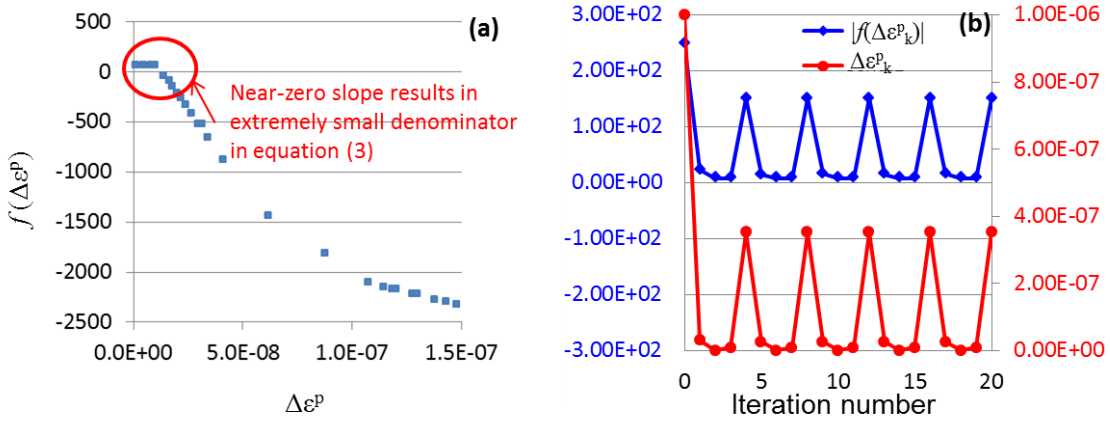


Figure 1 (a) f as a function of $\Delta\varepsilon_p$ in the scenario of divergence. (b) The divergence of the return mapping scheme: The red curve shows each iterate whereas the blue shows the corresponding f evaluations. The secant's procedure would direct the solution away from the true root whenever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

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. 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 set,

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 $\Delta\varepsilon_{m}^{p}=\dfrac{\Delta\varepsilon_{a}^{p}+\Delta\varepsilon_{b}^{p}}{2}$.

c) Calculate the secant guess $\Delta\varepsilon_{s}^{p}=\Delta\varepsilon_{k}^{p}-f(\Delta\varepsilon_{k}^{p})\dfrac{\Delta\varepsilon_{k}^{p}-\Delta\varepsilon_{k-1}^{p}}{f(\Delta\varepsilon_{k}^{p})-f(\Delta\varepsilon_{k-1}^{p})}$.

d) If $\Delta\varepsilon_{s}^{p}$ falls in $\Delta\varepsilon_{m}^{p}$ 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) $k=k+1$

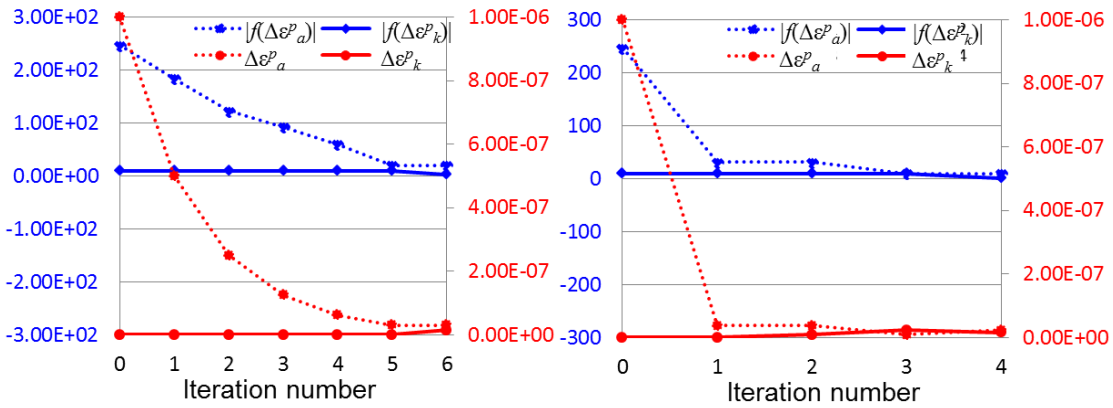
If bisection flag is clear, continue one more step of secant's method and go back to (2).

Note that with this method $\left|\Delta\varepsilon_{k+1}^{p}-\Delta\varepsilon_{k}^{p}\right|$ is guaranteed to shrink by a factor of 0.5 (or less if the secant guess is applied) at each iteration. To get started, we need to run the secant method first until we can securely bracket our solution. However in the scenario we studied, the Dekker's method actually initiates 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 stamping 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) $f(\Delta\varepsilon_{k}^{p})$ and $\Delta\varepsilon_{k}^{p}$ are plotted together with $f(\Delta\varepsilon_{a}^{p})$ 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 $f(\Delta\varepsilon_k^p)$ 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: (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. In a similar manner, the algorithm of Brent's method is also implemented in the return mapping scheme of material model 106. Subjecting this algorithm to the same problem shown in Fig. 1, we are also 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.

The secant method has a convergence rate of 1.618 and the inverse quadratic interpolation 1.839. As the bisection method has a convergence rate of 1, both Dekker's and Brent's methods are expected to converge super linearly. Depending on the nature of function *f* and how good our initial guess is, they may be slower or faster than the original secant method. It also should be noted that due to the extra tests to ensure stability, it will take more time for Dekker's or Brent's

method to find the next iterate. However both methods are rewarded with much better convergence, as we can see from later numerical tests.

3. Paralleling computations in the thermal solution

In the common practice of hot stamping simulations, an efficient choice is to adopt a weak thermo-mechanical coupling scheme in which the mechanical deformation process is advanced through an explicit time scheme whereas an implicit scheme is employed in the thermal analysis. Therefore, the inherent stability with the implicit time scheme allows a much larger thermal time step size, compared to the mechanical time step size. The thermal solutions within each large thermal time step are obtained through interpolation in order to determine the thermal conditions for the mechanical solution. As the mechanical time step size is much smaller compared to the thermal time step size, this interpolation is one of the most heavily used procedures during hot stamping simulations hence worth being paralleled. Another computation paralleling is implemented in the conjugate gradient (CG) solver, the default solver choice for the thermal linear equation system. At each CG iteration, a matrix-vector multiplication procedure is repeated several times to calculate the residual and correction vector, which takes up a substantial amount of time especially when lots of iterations are needed to find the implicit solution. Note that CG is guaranteed to converge within n iterations where n is the size of the linear system. So if the model is significantly large and the convergence is relatively slow, paralleling this operation can save lots of clock time. These two parallel implementations are only effective when using SMP.

4. Simulation results and discussion

Next we conducted three full-length hot-stamping simulations using the secant method, Dekker's method with parallel thermal solver (labeled as Dekker) and Brent's method with parallel thermal solver (labeled as Brent). In these tests, we observe if convergence is accomplished in each element at every time step and compare the final results in terms of the computation cost and key plastic deformation results such as shell thickness and effective plastic strain. The computational cost is measured by the clock time spent on the return mapping scheme, temperature interpolation, conjugate gradient solver and the entire simulation. The results are summarized in Table 1.

Table 1 itemized the clock time of the three different methods. In terms of the clock time spent

in return mapping algorithm for the three methods, using the secant method as a reference, a slight increase is seen in the new schemes, but the increase is no more than 3%. This in general makes sense because the idea of Dekker's and Brent's methods is to sacrifice part of the superlinear convergence rate (by inserting some bisection steps) to guarantee convergence when the *f* function does not behave well.

Table 1 Comparison of clock time spent in return mapping, interpolation and CG

	Simulation time	Secant	Dekker	Brent
Case I	Return mapping	Reference	+3%	+2%
	interpolation	Reference	-48%	-50%
	Conjugate gradient	Reference	-37%	-38%
	total	Reference	-5%	-5%
Case II	Return mapping	Reference	+0.2%	+1%
	interpolation	Reference	-45%	-36%
	Conjugate gradient	Reference	-46%	-45%
	total	Reference	-24%	-1%
Case III	Return mapping	Reference	+1%	+3%
	interpolation	Reference	-44%	-38%
	Conjugate gradient	Reference	+75%	+76%
	total	Reference	+1%	8%

On the other hand, paralleling computation in temperature interpolation accomplishes a substantial reduction in clock time in all test cases, whereas in 2 out of 3 cases, paralleling CG brings in efficiency benefits. In terms of the overall simulation time, cases I and II see a cut, which means the time benefit of paralleling is able to compensate the efficiency loss in stable return mapping, whereas case III shows an increase in the total simulation time. The reason that case 3 didn't see a time benefit is that the thermal solution of most time steps in this case take less than 2 iterations. In this case, the paralleled CG algorithm takes even more time because the action of paralleling itself cost additional computation resources. However, note that case 3 is a relative simple case with linear thermo-physical properties and constant boundary conditions. In practice, we would need to consider more complicated scenarios which are expected to increase the difficulty of solving thermal equation systems hence the total number of iterations. Therefore, we

are expecting to see time benefit from parallel computations in large and complicated simulation jobs.

Now let's take a look at the convergence behavior and metal forming results, which are listed in Table 2. As shown, both Dekker's and Brent's methods achieve full convergence in all test cases. On the other hand, the original secant method fails to accomplish convergence for more than 50 times. Regarding the metal forming results, namely shell thickness and effective plastic strain, the difference between the three methods are very small (mostly below 1%) in the test cases. However this doesn't necessarily mean that un-converged plasticity algorithm doesn't matter. As we are not able to explore all the possible simulation scenarios, having convergence at each step always gives us peace of mind that the simulations are on the right track.

Table 2 Comparison of convergence and metal forming results

			Secant	Dekker	Brent
Case I	Converged?		No	Yes	Yes
	Shell thickness (mm)	Minimum	0.049	0.049	0.049
		Maximum	2.090	2.133	2.239
	Effective plastic strain	Minimum	0.003	0.003	0.003
		Maximum	3.713	3.711	3.711
Case II	Converged?		No	Yes	Yes
	Shell thickness (mm)	Minimum	0.148	0.148	0.148
		Maximum	1.297	1.299	1.297
	Effective plastic strain	Minimum	0.0005	0.0004	0.0004
		Maximum	2.916	2.914	2.914
Case III	Converged?		No	Yes	Yes
	Shell thickness (mm)	Minimum	1.519	1.519	1.519
		Maximum	2.158	2.158	2.158
	Effective plastic strain	Minimum	0.001	0.001	0.001
		Maximum	0.474	0.474	0.474

5. Conclusion

Two improvements are made to the current hot stamping simulation schemes to improve convergence and efficiency: (1) Dekker's and Brent's methods are used to fix the convergence

issues in return mapping schemes of the material routine. Full convergence is accomplished in all test cases where the original secant method fails to achieve convergence, with slight increase in the computation time. (2) Parallel computation is used to increase the efficiency of two heavily used procedures in thermal solutions: temperature interpolation and conjugate gradient equation solver. This implementation helps to cut down the thermal solution time in cases where the implicit scheme is difficult to converge.