On the Experiences of Adding a Complex User-Material-Model to LS-DYNA®

Dr. Gurdip S. Kalsi

Computational Mechanics Consultant, AWE, Aldermaston, Berkshire, UK

Abstract

LSTC's DYNA codes (both SMP and MPP) have been extensively used at AWE over a period of many years since they represent state-of-the art computational analysis capability which is required for simulating most of our complex problems. Occasionally there is a requirement to enhance existing code capabilities. An example is a need to improve the simulation of the constitutive behaviours exhibited by Polymer-Bonded Explosives (PBXs). Most polymer-bonded explosives are dual-phase composites consisting of an explosive crystalline filler material bonded in an elastically softer polymer matrix, which results in a very complex heterogeneous material. This heterogeneity, and the non-linear properties of the matrix, can lead to very complicated constitutive responses being generated when a PBX is loaded. PBXs possess unequal properties in tension and compression, and show marked strain-rate and temperature dependency.

In the explosives arena, it is generally the energetic response of PBXs that forms the subject of in-depth studies, particularly from a viewpoint of safety assessments, e.g. accidental insults such as shock loading or fragment impact. The treatment of PBXs as structural, load bearing materials is less well investigated, for the primary purpose of a PBX is to act as an energetic source, rather than as a constructional material. However, there can be occasions when the structural response of a PBX needs to be well understood and controlled, and this new material model, known as GPM (Generic Polymer Material), was developed for this purpose.

The GPM model is a damage-based material model that can simulate some of the complex constitutive responses displayed by PBXs, with a particular emphasis on creep and viscous response. This paper will briefly describe the formulation and application of this model, but the focus will be on the implementation of this material model into the LS-DYNA codes as a user-material-model. This was a non-trivial exercise and its implementation required a good deal of effort, together with some internal code changes by LSTC to their codes in order for the model to function correctly, since the user-material-model interface was found to be inadequate to accommodate a complex, non-linear material model's requirements. Our experiences will be of potential benefit to other users who might find themselves in a similar situation.

Introduction

The LS-DYNA [1] codes have been extensively used at AWE over a period of many years since they represent state-of-the art computational analysis capability which is required for simulating most of our complex problems. These complexities arise from a variety of sources, primary ones being contact and non-linear material behaviour. Whilst the LS-DYNA codes do make available a very large and diversified library of material models, there are occasions when this capability is insufficient and a new material model needs to be incorporated into these codes by the user.

Such a need arose at AWE, which has funded and directed the development of a new material model to simulate the behaviour of Polymer-Bonded Explosives (PBXs). Most Polymer-Bonded Explosives are dual-phase composites consisting of an explosive crystalline filler material bonded in an elastically softer polymer matrix, making for a very complex heterogeneous material. This heterogeneity, and the non-linear properties of the matrix, can lead to very complicated constitutive responses being generated when a PBX is subjected to load, including strain-rate and temperature-dependency, as well as displaying different properties in tension and compression.

In the explosives arena, it is generally the energetic response of PBXs that forms the subject of in-depth studies, particularly from a viewpoint of safety assessments, e.g. accidental insults such as shock loading or fragment impact. The treatment of PBXs as *structural, load bearing* materials is less well investigated, for the primary purpose of a PBX is to act as an energetic source, rather than as a constructional material. However, there can be occasions when the structural response of a PBX needs to be well understood and controlled, and this new material model, known as GPM (Generic Polymer Material) [2,3], was developed for this purpose.

The GPM model is a damage-based material model that can simulate some of the complex constitutive responses displayed by PBXs, with a particular emphasis on creep and viscous response. This paper will briefly describe the formulation and application of this model, but the focus will be on the implementation of this material model into the LS-DYNA codes as a user-material-model. This was a non-trivial exercise and its implementation required a good deal of effort, together with some internal code changes by LSTC to their codes in order for the model to function correctly, since the user-material-model interface was found to be inadequate to accommodate a complex, non-linear material model's requirements. Our experiences will be of potential benefit to other users who might find themselves in a similar situation.

Mechanical Characteristics of a Selected Polymer-Bonded Explosive

Compared to the usual engineering materials, polymer-bonded explosives exhibit very complex constitutive responses. These arise from the composite nature of the material, and from the inherent viscoelastic/plastic behaviour of the polymeric binder. The properties of the explosive chosen for study are strongly influenced by temperature and strain-rate. By way of an example, Figure 1 shows the influence of temperature on the uniaxial compression response of this material at three different temperatures, when tested at a strain-rate of 5000microstrain/s.

As well as being strongly temperature and strain-rate sensitive, explosives typically possess different properties in tension and compression, both in terms of 'modulii' and peak strengths. These characteristics and the influence of strain-rate on material properties can be seen in Figure 2, which shows the data gathered from uniaxial tension and uniaxial compression tests at strain-rates of 5 to 5000microstrain/s, at decade intervals. Additionally, the explosive under study suffers continuing damage under load, the rate of damage accumulation being a function of load intensity and the rate of load application. Under load this material's properties suffer progressive degradation until ultimate failure occurs.

As is to be expected from the presence of a polymeric binder, most PBXs readily creep under load. Creep deformations are very low-rate, time dependent phenomena that can lead to undesirable consequences, so it is important to gain a good understanding of the creep behaviour of such materials. Creep strains can develop even under quite low stresses, which can exist in any engineering system, or even be generated anew by changes in the thermal environment. Over a prolonged period of time such strains can lead to undesirable changes in geometry, or even to failure of a component.

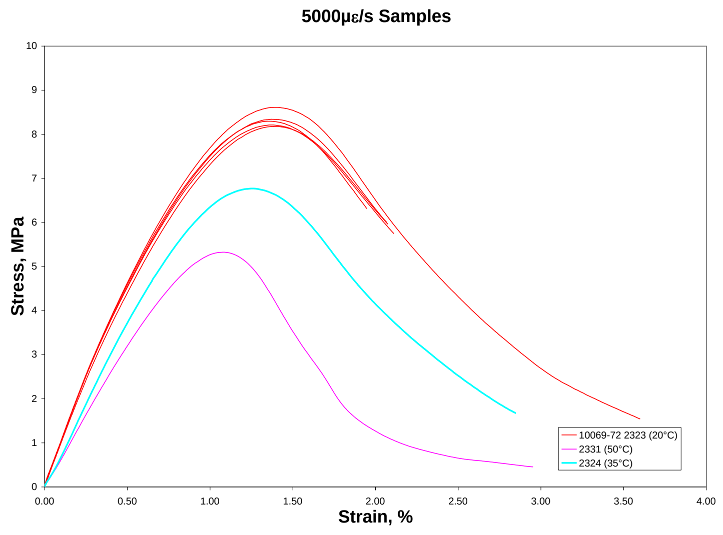

Figure 1: Influence of temperature on compression response

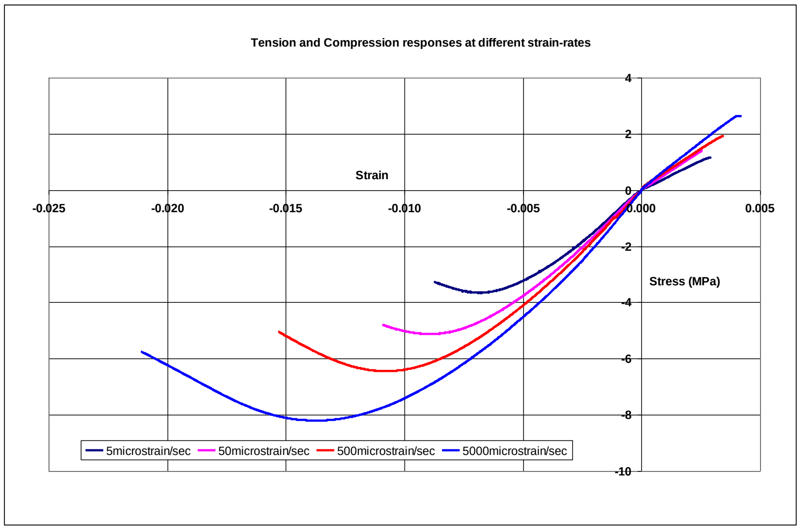

Figure 2: Influence of strain-rate on asymmetric material properties

It is vital to have sufficient experimental data of high quality to validate any numerical model, particularly so in the case of a heterogeneous and temperature, time and rate-dependent material such as the explosive under study. Therefore an extensive test programme was put under way to gather material property and response data over a wide variety of loading conditions. The test plan covered the following amongst other tests, at a range of temperatures:

strain-rate controlled tension and compression tests; load/unload cycling in tension and compression; creep testing in tension and compression; drop-weight tests to cover higher strain-rates; load-hold-release tests in tension and compression, fracture and triaxial tests, etc.

GPM Constitutive Model

The long-term goal behind the development of our new GPM material model is to be able to simulate both quasi-static, short-duration as well as extremely long duration problems within the same theoretical framework, and with an equal degree of fidelity. Such a model needs to be able to operate in a stable fashion with material constants that change with the sense of loading, particularly in the case of cyclic loads that change in sign during a cycle. Equally, where different parts of a structure may experience compressive or tensile stresses (e.g. a beam in bending), it is imperative that the model can use differing material constants (e.g. 'modulii') without encountering computational difficulties.

The GPM model used the early ViscoScram model [4] as a starting point, but it has been significantly extended to enable more realistic simulation of complex material behaviours. AWE funded the development of our new GPM model in a commercially available code called LUSAS [5]. At AWE the GPM model has been implemented as a user-material-model by Kalsi [6] for use with both the SMP and MPP versions of DYNA. These codes, particularly the MPP more so than the SMP version, provide a greater degree of scalability and hence speed than is available in LUSAS, which is an extremely important consideration. This is especially so when analysing very large finite element models over extended analysis times, such as simulation periods lasting a number of years.

A basic rheological representation of the new model is shown in Figure 3. It consists of a viscoelastic component that is constructed from a generalized Maxwell model, placed in series with an external linear spring and the ViscoScram damage component. Whilst this damage formulation simulates the effects of material degradation in compression quite well, it is not able to simulate brittle failure in tension.

In order to simulate secondary creep which occurs at a constant rate, an Eyring dashpot [7], which is based on rate process theory, has been included in this model and is also shown in Figure 3. Unlike Newtonian dashpots, the Eyring dashpot behaves nonlinearly and can in theory also account for temperature and strain-rate effects.

The GPM model allows different material properties to be specified in tension and compression for all but the damage component. Furthermore the material properties can vary with temperature, and this is an important requirement in any material model that may be used to study the influence of different thermal environments upon or within a system. In theory, any number of Maxwell units may be employed within the model, being limited really only by the data available to characterise their properties.

Brief Mathematical Details

The deviatoric stress rate for the viscoelastic component represented by the generalised Maxwell model consisting of N Maxwell units in parallel may be written as

$$\dot{s}_{ij} = \sum_{n=1}^{N} \left(2G^{(n)} \dot{e}_{ij}^{ve} - \frac{s_{ij}^{(n)}}{\tau^{(n)}} \right) \tag{1}$$

where $\tau^{(n)}$ is the relaxation period for the nth Maxwell unit, and $G^{(n)}$ is the corresponding shear modulus. The deviatoric strain rate ($\dot{e}_{ij}$) in the model consists of viscoelastic $\left(\dot{e}_{ij}^{ve}\right)$, permanent deformation/creep $\left(\dot{e}_{ij}^{eyr}\right)$, damage $\left(\dot{e}_{ij}^{dam}\right)$ and the external spring $\dot{e}_{ij}^{spr}$ components, i.e.

$$\dot{e}_{ij} = \dot{e}_{ij}^{ve} + \dot{e}_{ij}^{eyr} + \dot{e}_{ij}^{dam} + \dot{e}_{ij}^{spr} \tag{2}$$

so that $\dot{e}_{ij}^{ve} = \dot{e}_{ij} - \dot{e}_{ij}^{eyr} - \dot{e}_{ij}^{dam} - \dot{e}_{ij}^{spr}$ (3)

where $\dot{e}_{ij}^{spr} = \frac{1}{2G_{spr}} \dot{s}_{ij}$.

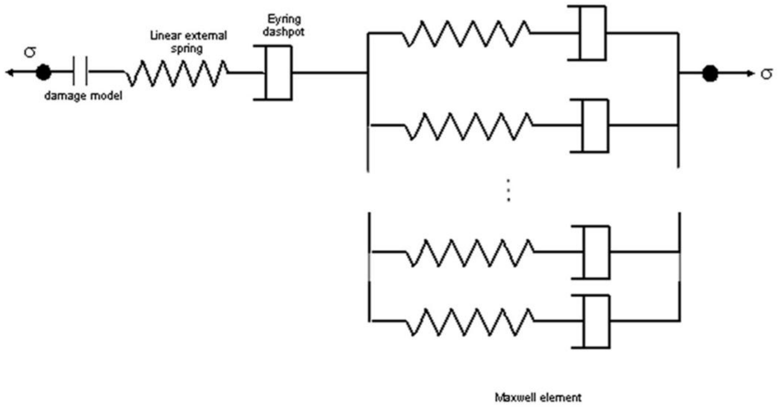

Figure 3: Mechanical analogue of GPM Model

By substituting equation (3) into equation (1) the deviatoric stress rate for the material model may be written as

$$\dot{s}_{ij} = \sum_{i=1}^{N} \left(2G^{(n)} \left(\dot{e}_{ij} - \dot{e}_{ij}^{eyr} - \dot{e}_{ij}^{dam} - \dot{e}_{ij}^{spr} \right) - \frac{s_{ij}^{(n)}}{\tau^{(n)}} \right) \tag{4}$$

As previously mentioned, the creep component is modelled using an Eyring dashpot. The strain rate for it in three dimensions is defined as

$$\dot{e}_{ij}^{eyr} = \frac{3A}{2} \frac{s_{ij}}{\sigma_e} \sinh\left(v_t \sigma_e\right) \tag{5}$$

where A and v_t are material parameters, and σ_e is the effective deviatoric stress. It is noted that the Eyring function can quickly become unstable at high stresses, so care is needed in its use.

After further mathematical development, and employing the Crank-Nicolson [8] scheme for solution, the governing differential equation for the new model may finally be expressed as

$$\sigma_e\left(t+\frac{1}{2}\Delta t\right)\frac{\Delta s_{ij}}{\Delta t}+\left[g_1\sinh\left(v_t\sigma_e\left(t+\frac{1}{2}\Delta t\right)\right)+g_4\sigma_e\left(t+\frac{1}{2}\Delta t\right)\right]\left(s_{ij}+\frac{1}{2}\Delta s_{ij}\right)$$

$$= \quad g_2\sigma_e\left(t+\frac{1}{2}\Delta t\right)\frac{\Delta e_{ij}}{\Delta t}-g_3\sigma_e\left(t+\frac{1}{2}\Delta t\right)\lambda_{ij} \qquad (6)$$

where
$$g_1=\frac{3A\overline{G}}{\left[1+(c/a)^3+\left(\overline{G}/G_{spr}\right)\right]}, \qquad g_2=\frac{2\overline{G}}{\left[1+(c/a)^3+\left(\overline{G}/G_{spr}\right)\right]},$$

$$g_3=\frac{1}{\left[1+(c/a)^3+\left(\overline{G}/G_{spr}\right)\right]}, \qquad g_4=\frac{3(c/a)^2(\dot{c}/a)}{\left[1+(c/a)^3+\left(\overline{G}/G_{spr}\right)\right]},$$

c is the average crack radius, 'a' is the limiting flaw size, and sum of all the Maxwell shear modulii is approximated as

$$\overline{G}=\sum_{n=1}^{N}G^n\ , \qquad \text{and the history term } \lambda_{ij}=\sum_{n=1}^{N}\frac{s_{ij}^{(n)}}{\tau^{(n)}}$$

Equation (6) is a non-linear equation and needs to be solved for both the incremental stresses and the consistent tangent modulus matrix [9]. Details of these procedures are not presented here for brevity.

On Porting of Material Models

Generally a material model needs the newly calculated strain increments, together with all the associated history and state variables, as input to be supplied by the calling program. It then calculates the corresponding stress increments and updated history and state variables and returns these to the analysis program.

To make a material model *easily* portable between different programs, the following requirements are essential:

- The material model should be as fully self-contained as possible, and not reference any coding, other than standard functions or calls, which does not reside within the material model subroutines
- The interface through which the analysis code exchanges information with the material model should give access to the same collection of variables in both the donor and the host program
- The overall program flow should be virtually identical for the solution of a given type of problem

- Both analysis programs should be either serial or parallel in operation

The first requirement can be readily met by careful design of the software, and the second one is usually fulfilled for the majority of material models that do not stray too far from the theory or format of existing models. However, when complex or advanced materials are developed that differ from such formats, then serious difficulties and delays can arise in porting from the donor to the host code. This circumstance has caused the most serious problems in porting the model from LUSAS to LS-DYNA.

Although LUSAS and LS-DYNA both solve implicit problems, the way in which the initializations are done in the first two-three cycles of LS-DYNA at start-up time are different from the way the logic in LUSAS flows, and from the logic flow expected based on prior experience of implicit codes. Teasing out and understanding the LS-DYNA logic without any supporting documentation, and overcoming problems caused by this rather surprising difference caused significant difficulties and delays during the early porting attempts.

LUSAS was originally a serial code, but it has some parallel capabilities. LS-DYNA has a shared memory parallel version which has been shown to have only limited scalability. Without appropriate changes to the material model code, this operating system difference can lead to clashes between processors in multi-processor runs. To make the serial version of the GPM model run correctly in multi-processor mode required the use of certain OPENMP constructs in some subroutines.

Code Changes

One of the files that LSTC makes available to code developers who wish to integrate their own computational developments or software into LS-DYNA is called 'DYN21.f'. It is to this subroutine that user-material-models can be added for integration with the rest of the LS-DYNA code capabilities. The details of how this may be done are explained briefly in Appendix A of the LS-DYNA User Manual.

The GPM model has been defined as material number 41 in LS-DYNA. Two new subroutines, UMAT41 and UTAN41, have been written to manage interactions between LS-DYNA and this user material model. Briefly, UMAT41 is called in the post-solution (i.e. calculation of stress increments, etc.) phase, whilst UTAN41 is called in the pre-solution phase, during the calculation of element and global matrices. The GPM model is a complex material model, and as coded in LUSAS it consists of over 70 subroutines, including quite a few utility subroutines. Major problems were caused by the restricted list of variables that can be transferred between a user-material-model and LS-DYNA via the user-material-model interface, and by the fixed sizes of some arrays in LS-DYNA.

In LS-DYNA input material properties are conventionally read into the 'CM' array, which at the time this work was carried out was fixed at size 50, i.e. CM(50), and could not be modified by users. Whilst this array size is satisfactory for use with most material models, it is insufficient for the GPM model where the number of constants increases as the number of Maxwell elements is increased. For example for a 9-spring model which was used in analysis the total number of material variables adds up to 228, for material properties at three different temperatures. So the existing logic in LS-DYNA could not be used to read in the long list of GPM material constants.

The first ever call by LS-DYNA to the GPM model (or to any other user-material-model) is made via subroutine UMATnn, in our case UMAT41. So a section of coding was developed there to read the GPM material constants via a new input file named 'GPM_PROPS' using file unit number 95. These constants are then stored in a local array during analysis. Ideally these constants should have been transferred to LS-DYNA so that they would then be available to a subsequent restart run via the standard dump file, but the limited size of the CM array prevented this happening across the user-material-model interface. Therefore what was done was to immediately write out the newly-read constants to a file named 'GPM_RESTART', on file unit 97, so that LS-DYNA could pick up the material constants during a restart analysis and transfer them into the local material arrays. The input material data is echoed to the 'd3hsp' so that the user can check that the constants have been read in correctly.

Since LS-DYNA is a multi-processor code, reading of these material data files in the SMP version required the use of OPENMP constructs ('c$omp critical', 'c$omp end critical') around the relevant pieces of coding. These constructs were needed to ensure that only one processor reads the 'GPM_PROPS' or 'GPM_RESTART' file, before transferring these constants to the local arrays. Similar constructs were also needed elsewhere in quite a few of the GPM model subroutines to prevent multiple processors trying to access the same disk files at the same time.

For reasons to do with initialisations, and manipulation of basic material properties, the GPM model needs to distinguish between the *start* of a new loadstep, and equilibrium iterations *within* this loadstep. Further, it also needs to be able to distinguish between the very *first* loadstep at time zero, and subsequent loadsteps that begin a *new* cycle of calculations. Whilst the difference between these situations was clearly defined in LUSAS and the information easily passed to the GPM model, this was more difficult to accomplish in LS-DYNA. This is because this kind of information is not passed by LS-DYNA via the user-material-model-interface, and because of the difference in the sequencing of the calls made to material models by LS-DYNA and LUSAS during pre- and post-solution phases. So a careful study was required to figure out the way in which LS-DYNA calls the pre- and post-solution portions of a user material model. These calling sequences are slightly different at analysis start-up compared to those at the beginning of a subsequent new loadstep.

The lack of program documentation for LS-DYNA was a real impediment at this juncture of the work. A quite convoluted piece of coding needed to be written into subroutines UMAT41 and UTAN41 to help distinguish between these distinct analysis conditions. Abstracts of this coding are shown on the next page. A local array named RINC common to both these subroutines was needed to track the progression of each element through the solution process, and this information was written into LS-DYNA's state array HSV at every exit from subroutines UMAT41 and UTAN41. The updated state and history variables, as well as updated stresses, are transferred back to LS-DYNA upon exit from UMAT41.

The number of state and history variables associated with the GPM model is quite large, and increases as the number of Maxwell elements or springs is increased. For our 9-spring model the number of state and history variables is 177. At the time this work was done the HSV array size had been increased from the previously fixed size of 142 to 200 at AWE's request, hence limiting our use of the GPM model to 9 Maxwell springs at that time. Since then, LSTC have enabled the size of the HSV array to be set by any code developer, which is helpful.

```
              ……
CGK     to determine whether new load increment (NEWINC) or not
        GKDTIME = DTIME
        GKRSPTM = RSPTM - GKDTIME
        NEWINC    = .FALSE.
C--- FIRST INCREMENT …….
              IF(GKRSPTM .LT. 1.0E-12) THEN
                 IF(RINC(NEL) .GT. 1.5) THEN
                   NEWINC = .FALSE.
                 ELSEIF (RINC(NEL) .EQ. 1.0) THEN
                   NEWINC = .TRUE.
                   RINC(NEL) = RINC(NEL) +1.5
                 ELSEIF (RINC(NEL) .LE. 0.5) THEN
                   NEWINC = .TRUE.
                   RINC(NEL) = -1.5
                ENDIF
C
              ELSE
                 RINC(NEL) = RINC(NEL) + 1.0
                 IF(RINC(NEL) .EQ. 2.0 .OR. RINC(NEL) .EQ. 3.0) NEWINC = .TRUE.
              ENDIF
C--- to determine whether analysis is on the first time step and on a new increment
              INTIAL = NEWINC .AND. EQL(GKRSPTM,R0)
              REPAS = .TRUE.
              IF(INTIAL) REPAS = .FALSE.
              IF(EQL(GKRSPTM, R0) .AND. RINC(NEL) .GT. 1.5) REPAS = .FALSE.
              PRESOL = .FALSE.
C
              CALL …..
              ……
```

Within the GPM model, the calculation of creep deformations uses the Eyring dashpot formulation, and this calculation can on occasion fail to converge. In this case what is required is for the analysis program to re-initialise to the converged state at the end of the previous load increment, and re-enter the current loadstep with a smaller load increment. When this situation arises, a flag is set to 'TRUE'. In LUSAS this flag is passed back by the GPM model and the analysis can continue with a smaller loadstep size. However the interface in LS-DYNA does not permit the transfer of this information from the GPM subroutines back to LS-DYNA. Neither has it been possible to find out whether this information can be passed back via one of the common blocks in LS-DYNA. So currently in LS-DYNA the GPM model has been altered to write out an informative error message and it then halts the analysis, which is a serious impediment that requires to be overcome with LSTC's help. The alternative at present is to restart from the latest dump files with a reduced loadstep size, but this can cause delays, especially if such a condition occurs early during a weekend run, etc.

For the present, the GPM model has been activated for use only with 3D solid elements, and subroutines URMATHN and URTANH in 'DYN21.f' have been suitably modified. Basically, URMATHN makes calls to subroutine UMAT41 to compute stress increments, etc. The GPM driver needs to know the number of Maxwell elements being used, and also the external or user defined element number for which the call is being made. Subroutine URMATHN in 'DYN21.f' was suitably modified to obtain this information, two additional parameters were added to the default call to UMAT41, and the 'if' statement around the call to 'usr_temps' was permanently de-activated so as to always obtain the current element temperature. URTANH calls subroutine UTAN41 to compute element matrices which are required to form the global stiffness matrix.

Changes similar to those in URMATHN were made here. Additional subroutines that have been modified are URMATS, URMATB, URMATD, URMATT, UMAT41V, URTANS, and UTAN41V to prevent these subroutines making calls to the GPM model via the UTAN41 and UMAT41 subroutines.

Input Changes

LS-DYNA expects to read a minimum amount of data when a user-material-model is used in an analysis. As already explained, the CM array was not large enough to accept all the material constants that are needed to define a reasonable-sized GPM model, in terms of the number of Maxwell elements. So the standard input format employed by LS-DYNA has been slightly modified in order to transfer additional data via the CM array to the GPM subroutines, as this information is required to be available within LS-DYNA before any calls are made to the GPM model.

In order to obtain the 177 or as many as requested of the history and state variables for post-processing, the following command must be included in the LS-DYNA input file:

```
$
*DATABASE_EXTENT_BINARY
177,0,0,1
0
$
```

The 177 variables are then output in the usual way by LS-DYNA to the ‘d3plot’ family of files that can then be examined with LS-PrePost®. There is a related bug that has been reported to LSTC, but for which there is apparently no easy fix available. What happens is that LS-DYNA now writes out 177 values for *all* materials used in the model, even if these materials have far fewer, say 3 or 4, actual state variables. This obviously increases the size of the ‘d3plot’ files very, very significantly, and can lead to disks filling up during analysis of large models that dump data frequently, and cause analyses to crash.

Application in LS-DYNA

The GPM model has been extensively used at AWE for large, complex analyses. It has shown itself to be a robust, efficient and stable model, producing good match with experimental data. Figure 4 shows the results of a simulation wherein a compression creep test was simulated using this model. The creep test was carried out over a period of two months. There is seen to be a good correspondence between test and computed results.

The real advantage to be gained from this model transfer into LS-DYNA is one of execution speed and reduction in turnaround times. Our experiences with the SMP version show that LS-DYNA does not scale well much beyond 4 processors for our implicit analyses. However, this still gives significant gains compared to any serial program.

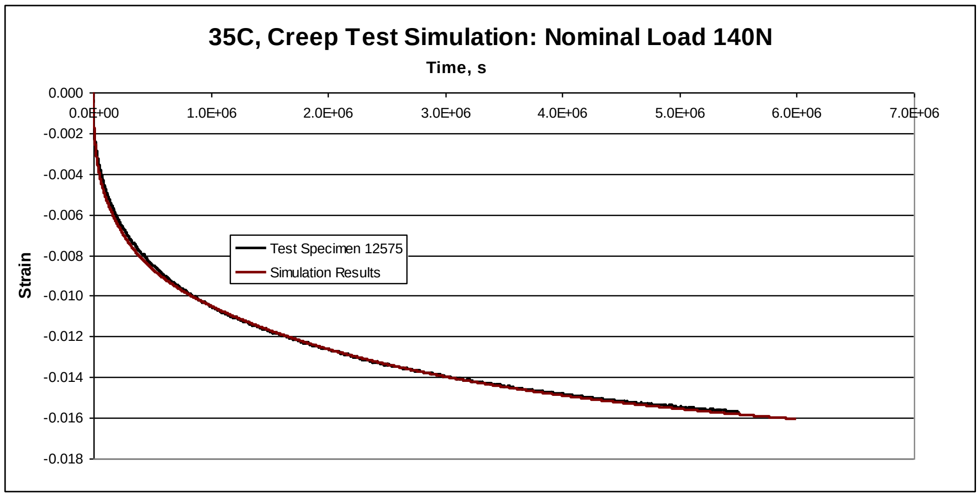

Figure 4: Comparison of Computed Results and Test Data for a 2-month Creep Test

Some of our recent implicit analyses have often taken many tens of days to run to completion with the SMP code. So there is clear advantage in applying the MPP version of LS-DYNA to our analyses for improved scaling with larger numbers of processors. However because of the small fixed size of the CM array it has not been possible until very recently, when LSTC finally gave us an increased-size array, to use the GPM model with the MPP version of LS-DYNA. This was due to the inability in MPP of ensuring that only one processor reads the input material file 'GPM_PROPS' before storing the input constants in the local array: to our knowledge, system constructs equivalent to those used in the SMP version could not be used in MPP based on the information available from LS-DYNA to the material model programmer. Thus any attempt to read in material data caused the program to crash due to processor conflict over file access.

Figure 5 shows the speedup in turnaround time when using many more processors. This study was carried out with the MPP version using a contact-dominated problem made up of about 1.5 million elements. Compared to 4 processors, there is a maximum speedup of about 4.5 times with 192 processors and probably beyond, for this example. Results in Figure 5 suggest that using many more processors will not reduce the turnaround time much further in this case. Compared to the run time using the SMP version of LS-DYNA on 4 processors, the speed-up is around 7. However the benefits of porting the GPM model, with the aim of reducing turnaround times for our current large models and even larger future models, are self-evident particularly with the MPP version of LS-DYNA.

Conclusions and Recommendations

This paper briefly described the GPM material model, which has been developed to model the complex behaviour exhibited by polymer-bonded explosives, whose responses are strongly influenced by strain-rate and temperature. In order to take advantage of the very powerful features and multi-processing capabilities of LS-DYNA, the GPM model has been ported to this state-of-the-art code. The greatest difficulties arose from the lack of versatility in the material modelling interface, and the small fixed sizes of the CM and HSV arrays. This interface needs to be modified to allow more variables to be transferred between LS-DYNA and a user-material-model.

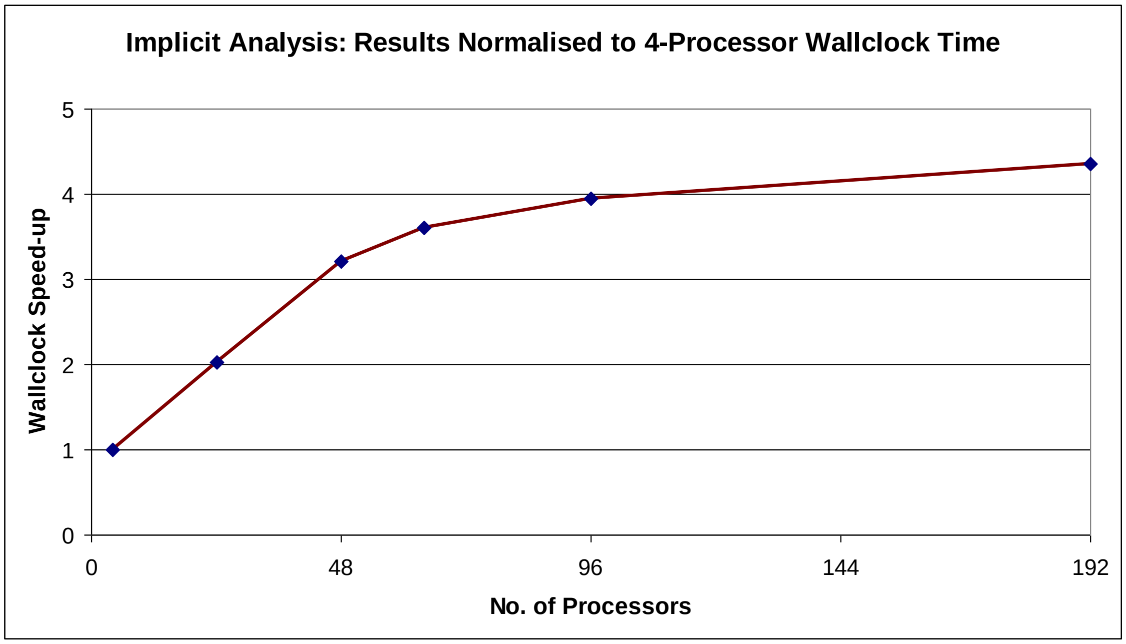

Figure 5: Wallclock Speedup with MPP version for Contact-Dominated Problem

The problem of the GPM model flagging to LS-DYNA that the current loadstep needs to be re-calculated, with a smaller load increment size, requires help from LSTC in order to be fixed. To reduce the risks of 'disk-full' crashes, LSTC also need to stop very large-sized history files being produced for *all* materials in an analysis, simply because the GPM model needs to write out many more state variables. Preliminary results presented here show that significantly reduced turnaround times can be obtained with the MPP version of LS-DYNA when simulating very large, complex, implicit problems. The solvers need to be further parallelised so that run times can be reduced even more for these types of problems.

References

1. Livermore Software Technology Corporation –LS-DYNA, A Program for Nonlinear Dynamic Analysis of Structures in Three Dimensions, Revision 58153, V. ls971d R5 beta, 02/10/2010.
2. Kalsi, GS - Modelling the Creep Response of a Polymer Bonded Explosive. NAFEMS World Congress, June 16-19, 2009, Crete, Greece.
3. Kalsi, GS – Experimental Characterisation and Modelling of the Mechanical Behaviour of Polymer Bonded Explosives. NAFEMS World Congress, May 23-26, 2011, Boston, USA.
4. Robert MH, Joel GB – An Implicit Finite Element Model for Energetic Particulate Composite Materials, International Journal for Numerical Methods in Engineering, 49, 1191-1209, 2000.
5. LUSAS Finite Element System, User Manual, FEA Ltd., Surrey, UK.
6. Kalsi GS – TN20/10, A User Guide to the GPM Model in GPM_LSDYNA3Dv1.8, AWE Report No. 610/10, July 2010.
7. Glasstone S, Laidler K, Eyring H – The Theory of Rate Processes, McGraw-Hill Book Company Inc, New York, NY, 1941.
8. Smith, GD - Numerical Solution of Partial Differential Equations – Finite Difference Methods, 2nd Edition, 1978, Clarendon Press, Oxford.
9. Bathe, K-J. 2003. Finite Element Procedures, 7th Printing, Prentice Hall.