

# **Getting Started with LS-DYNA**

**Copyright © 2002  
Livermore Software  
Technology Corporation  
All Rights Reserved**

## Table of Contents

page

<b>1. Introduction.....</b>	<b>1-1</b>
<b>2. Units.....</b>	<b>2-1</b>
<b>3. Getting Started.....</b>	<b>3-1</b>
<b>3.1 Problem Definition.....</b>	<b>3-1</b>
<b>3.2 Input File Preparation.....</b>	<b>3-1</b>
<b>3.3 LS-DYNA solution .....</b>	<b>3-4</b>
<b>4. The Next Step.....</b>	<b>4-1</b>
<b>4.1 Explicit Analysis (problem ex01.k).....</b>	<b>4-1</b>
<b>4.2 Implicit Analysis (problem im01.k).....</b>	<b>4-3</b>
<b>4.3 Heat Transfer Analysis (problem th01.k).....</b>	<b>4-5</b>
<b>4.4 Coupled thermal-stress analysis (problem cp01.k).....</b>	<b>4-7</b>

## List of Examples

page

<b>Example 4-1 Aluminum cube deformation, explicit method (file: ex01.k)</b>	<b>4-2</b>
<b>Example 4-2 Aluminum cube deformation, implicit method (file: im01.k)</b>	<b>4-3</b>
<b>Example 4-3 Aluminum cube transient heat transfer analysis (file: th01.k)</b>	<b>4-6</b>
<b>Example 4-4 Aluminum cube coupled thermal-stress solution (file: cp01.k)</b>	<b>4-8</b>

## **1. Introduction**

LS-DYNA is used to solve multi-physics problems including solid mechanics, heat transfer, and fluid dynamics either as separate phenomena or as coupled physics, e.g., thermal stress or fluid structure interaction. This manual presents “very simple” examples to be used as templates (or recipes).

This manual should be used side-by-side with the “LS-DYNA Keyword User’s Manual”. The keyword input provides a flexible and logically organized database. Similar functions are grouped together under the same keyword. For example, under the keyword, \*ELEMENT, are included solid, beam, and shell elements. The keywords can be entered in an arbitrary order in the input file. However, for clarity in this manual, we will conform to the following general block structure and enter the appropriate keywords in each block.

1. define solution control and output parameters
2. define model geometry and material parameters
3. define boundary conditions

## 2. Units

LS-DYNA requires a consistent set of units to be used. All parameters in this manual are in SI units.

### General

- length [m]
- mass [kg]
- temperature [K]
- time [sec]
- pressure [Pa]

### Mechanical units

- density [kg/m<sup>3</sup>]
- Modulus of elasticity [Pa]
- yield stress [Pa]
- coefficient of expansion [m/m K]

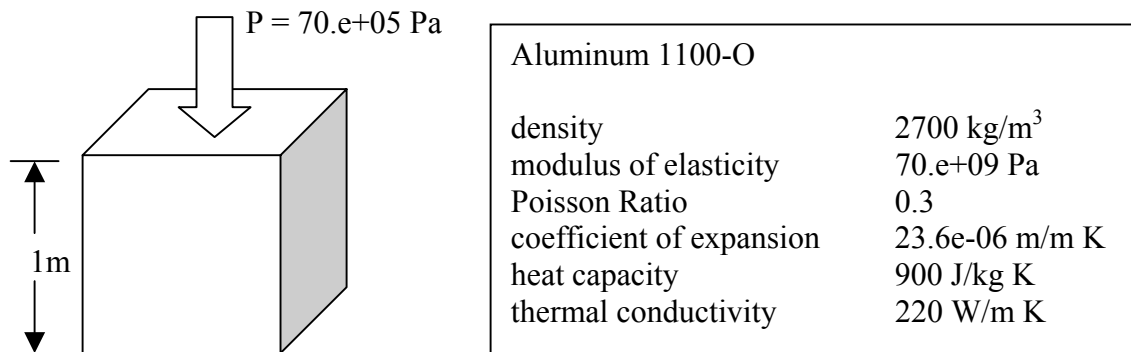
### Thermal units

- heat capacity [J/kg K]
- thermal conductivity [W/m K]
- heat generation rate [W/ m<sup>3</sup>]

### 3. Getting Started

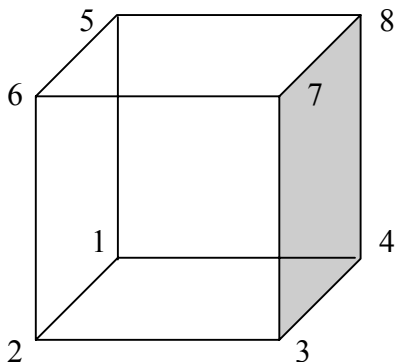
#### 3.1 Problem Definition

Consider the deformation of an aluminum block sitting on the floor with a pressure applied to the top surface.



#### 3.2 Input File Preparation

The first step is to create a mesh and define node points. Since we are just getting started, we will define the mesh as consisting of only 1 element and 8 node points as shown in the following figure. Also, we will use default values for many of the parameters in the input file, and therefore not have to enter them.



*The “LS-DYNA Keyword User’s Manual” should be read side-by-side with this manual.*

The following steps are required to create the finite element model input file.

\*KEYWORD

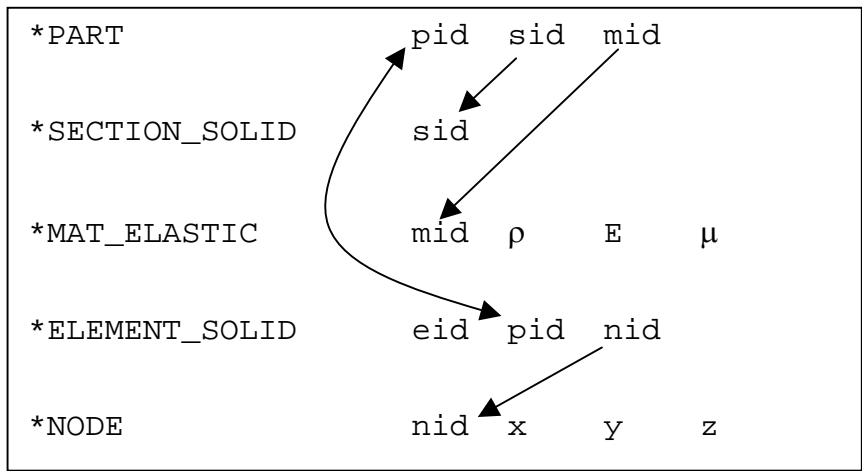
The first line of the input file must begin with \*KEYWORD. This identifies the file as containing the “keyword” format instead of the “structured” format which can also be used (see LS-DYNA Structured User’s Manual).

The first input block is used to define solution control and output parameters. As a minimum, the \*CONTROL\_TERMINATION keyword must be used to specify the problem termination time. We

will apply the pressure load as a ramp from 0 Pa to 70.e+05 Pa during a time interval of 1 second. Therefore, the termination time is 1 second. Additionally, one of the many output options should be used to control the printing interval of results (e.g., \*DATABASE\_BINARY\_D3PLOT). We will print the results every 0.1 seconds.

```
*CONTROL_TERMINATION
  1.
*DATABASE_BINARY_D3PLOT
  .1
```

The second input block is used to define the model geometry, mesh, and material parameters. The following description and map may help to understand the data structure in this block. We have 1 part, the aluminum block, and use the \*PART keyword to begin the definition of the finite element model. The keyword \*PART contains data that points to other attributes of this part, e.g., material properties. Keywords for these other attributes, in turn, point elsewhere to additional attribute definitions. The organization of the keyword input looks like this.



The LS-DYNA Keyword User Manual should be consulted at this time for a description of the keywords used above. A brief description follows:

- \*PART We have 1 part identified by part identification (pid=1). This part has attributes identified by section identification (sid=1) and material identification (mid=1).
- \*SECTION\_SOLID Parts identified by (sid=1) are defined as constant stress 8 node brick elements by this keyword.
- \*MAT\_ELASTIC Parts identified by (mid=1) are defined as an elastic material with a density  $\rho$ , a modulus of elasticity E, and a Poisson ratio of  $\mu$ .

\*ELEMENT\_SOLID Eight node solid brick elements identified by element identification (eid=1) have the attributes of (pid=1) and are defined by the node list (nid)

\*NODE The node identified by (nid) has coordinates x,y,z.

Our finite element model consists of 1 element, 8 nodes, and 1 material. Keeping the above in mind, the data entry for this block looks like this.

```

*PART
aluminum block
  1      1      1
*SECTION_SOLID
  1
*MAT_ELASTIC
  1      2700.      70.e+09      .3
*ELEMENT_SOLID
  1      1      1      2      3      4      5      6      7      8
*NODE
  1      0.      0.      0.      0.      7      7
  2      1.      0.      0.      0.      5      0
  3      1.      1.      0.      0.      3      0
  4      0.      1.      0.      0.      6      0
  5      0.      0.      1.      1.      4      0
  6      1.      0.      1.      1.      2      0
  7      1.      1.      1.      0      0
  8      0.      1.      1.      1      0
    
```

The third input block is used to define boundary conditions and time dependent load curves. We are applying a load of 70.e+05 Pa to the top surface of the block defined by nodes 5-6-7-8. We will ramp the load up from 0 Pa to 70.e+05 Pa during a time interval of 1 second.

Note that the first data entry in \*LOAD\_SEGMENT is a load curve identification number (lcid=1) which points to the load curve defined by the keyword \*DEFINE\_CURVE having that same (lcid) identification number.

```

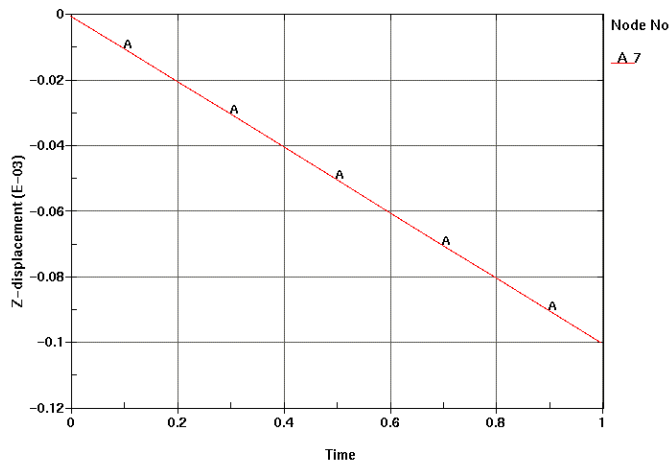
*LOAD_SEGMENT
  1      1      0.      5      6      7      8
*DEFINE_CURVE
  1
  0.      0.
  1.      70.e+05
    
```

\*END The last line in the input file must have the keyword \*END.

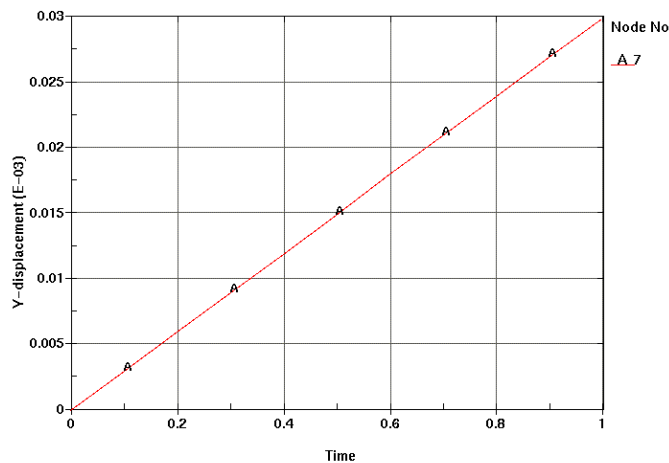


### 3.3 LS-DYNA solution

The vertical and horizontal displacement of node 7, calculated by LS-DYNA, are shown in the following 2 graphs. The solution to this simple problem can be calculated analytically. The LS-DYNA solution compares exactly with the analytical solution.



The vertical displacement due to a  $70.0e+05$  Pa pressure load can be calculated by

$$\Delta l = \frac{Pl}{E} = \frac{(70e+05)(1)}{(70e+09)} = 1.0e-04 \text{ m}$$


The horizontal displacement is

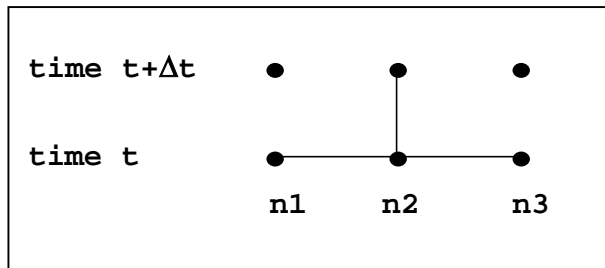
$$\Delta l_h = \mu \Delta l = (0.3)(0.0001) = 3.0e-05 \text{ m}$$

## 4. The Next Step

This chapter builds on the simple example presented in the previous chapter. First, more detail is given about solving this problem using explicit analysis in section 4.1. Explicit analysis is well suited to dynamic simulations such as impact and crash analysis, but it can become prohibitively expensive to conduct long duration or static analyses. Static problems, such as sheet metal spring back after forming, are one application area for implicit analysis. Implicit analysis is presented in section 4.2. The difference between explicit and implicit is described. The problem is then presented as a heat transfer problem in section 4.3 and finally as a coupled thermal-stress problem in section 4.4.

### 4.1 Explicit Analysis (problem ex01.k)

Explicit refers to the numerical method used to represent and solve the time derivatives in the momentum and energy equations. The following figure presents a graphical description of



explicit time integration. The displacement of node n2 at time level t+Δt is equal to known values of the displacement at nodes n1, n2, and n3 at time level t. A system of explicit algebraic equations are written for all the nodes in the mesh at time level t+Δt. Each equation is solved in-turn for the unknown node point displacements. Explicit methods are computational fast but are conditionally stable. The time step, Δt, must be less than a critical value or computational errors will grow resulting in a bad solution. The time step must be less than the length of time it takes a signal traveling at the speed of sound in the material to traverse the distance between the node points. The critical time step for this problem can be calculated by

$$\Delta t \leq \frac{\Delta x}{c} = \frac{\Delta x}{\sqrt{\frac{E}{\rho}}} = \frac{1.}{\sqrt{\frac{70. * 10^9}{2700.}}} = 1.96 * 10^{-4} \text{ sec}$$

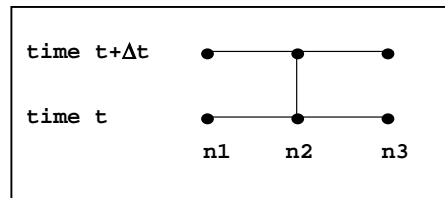
To be safe, the default value used by LS-DYNA is 90% of this value or 1.77e-04 sec. Therefore, this problem requires 5,658 explicit time steps as compared with 10 implicit time steps (see section 4.2). Note that the time step and scale factor can be set using the keyword `*CONTROL_TIMESTEP`.

The input file for the 1-element aluminum cube example problem, presented in Chapter 3, is duplicated below. The keyword `*TITLE` has been added for problem identification.



## 4.2 Implicit Analysis (problem im01.k)

Implicit refers to the numerical method used to represent and solve the time derivatives in the momentum and energy equations. The following figure presents a graphical description of



implicit time integration. The displacement of node n2 at time level  $t+\Delta t$  is equal to known values of the displacement at nodes n1, n2, and n3 at time level t, and also the unknown displacements of nodes n1 and n3 at time level  $t+\Delta t$ . This results in a system of simultaneous algebraic equations that are solved using matrix algebra (e.g., matrix inversion). The advantage of this approach is that it is unconditionally stable (i.e., there is no critical time step size). The disadvantage is the large numerical effort required to form, store, and invert the system of equations. Implicit simulations typically involve a relatively small number of computationally expensive time steps.

The keyword `*CONTROL_IMPLICIT_GENERAL` is used to activate the implicit method. The second entry on this card is the time step. For this example the time step is 0.1 sec. Therefore, a total of 10 implicit time steps will be taken to solve this problem. The results are identical to those obtained by the explicit method as shown in Sec 3.3.

For small problems, such as this 1 element example, most of the computer time is spent performing IO operations in reading the data and writing the output files. Very little CPU time is spent solving the problem. No conclusions should be made concerning the execution speed of explicit versus implicit methods on this problem.

### Example 4-2 Aluminum cube deformation, implicit method (file: im01.k)

```

*KEYWORD
*TITLE
im01.k - implicit analysis - problem 1
$
$----- implicit solution keywords -----
$
*CONTROL_IMPLICIT_GENERAL 1 .1
$
$----- define solution control and output parameters -----
$
*CONTROL_TERMINATION
1.
*DATABASE_BINARY_D3PLOT
.1

```

← This keyword turns on the implicit method

```

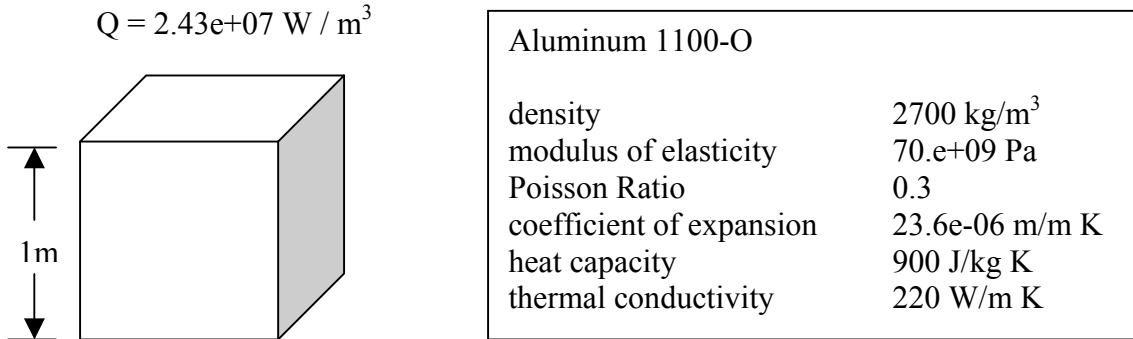
$$$$
$$$$----- define model geometry and material parameters -----
$$$$
*PART
aluminum block
  1          1          1
*SECTION_SOLID
  1
*MAT_ELASTIC
  1      2700.    70.e+09      .3

*NODE
  1          0.          0.          0.          7          7
  2          1.          0.          0.          5          0
  3          1.          1.          0.          3          0
  4          0.          1.          0.          6          0
  5          0.          0.          1.          4          0
  6          1.          0.          1.          2          0
  7          1.          1.          1.          0          0
  8          0.          1.          1.          1          0
*ELEMENT_SOLID
  1          1          1          2          3          4          5          6          7          8
$$$$
$$$$----- define boundary conditions and load curves -----
$$$$
*LOAD_SEGMENT
  1          1.          0.          5          6          7          8
*DEFINE_CURVE
  1
  1          0.          0.
  1          1.          70.e+05
*END

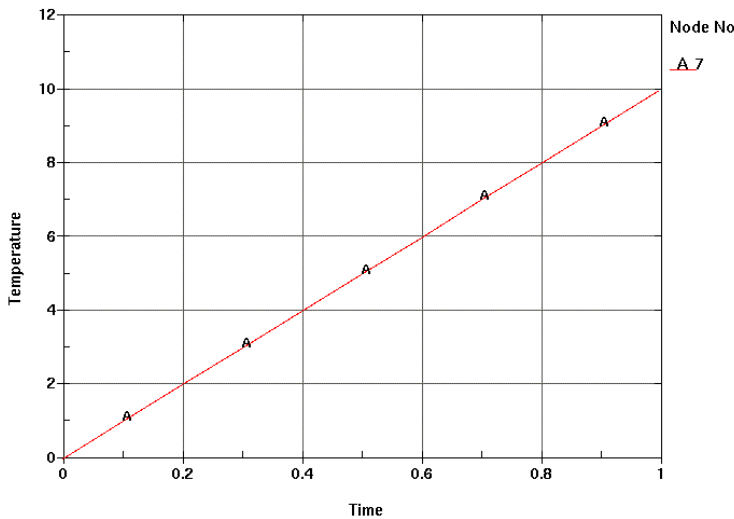
```

### 4.3 Heat Transfer Analysis (problem th01.k)

LS-DYNA can solve steady state and transient heat transfer problems. Steady state problems are solved in one step, while transient problems are solved using an implicit method. Our 1-element problem will now be re-defined as a transient heat transfer problem as shown below:



We will solve for the temperature response of the cube as the result of internal heat generation,  $Q$ . All the surfaces of the cube are perfectly insulated. Therefore, all the heat generation goes into increasing the internal energy of the cube. The temperature response of the cube calculated by LS-DYNA is shown in the figure below. The analytical solution is shown in the box.



$$T = \frac{Qt}{\rho V c_p}$$

$$T = \frac{(2.43e + 07)(t)}{(2700.)(1)(900.)}$$

$$T = 10t$$

The keyword input for this problem is shown below. Important things to note are:

- The default initial condition is  $T=0$  for all nodes.
- The default thermal boundary condition is adiabatic (i.e. perfectly insulated). Therefore, no thermal boundary conditions need be specified for this problem.
- The \*CONTROL\_SOLUTION keyword is used to specify this problem as thermal only.
- The entry on the \*PART keyword points to the definition of thermal property data
- The entry on the \*PART keyword points to the definition of thermal property data.

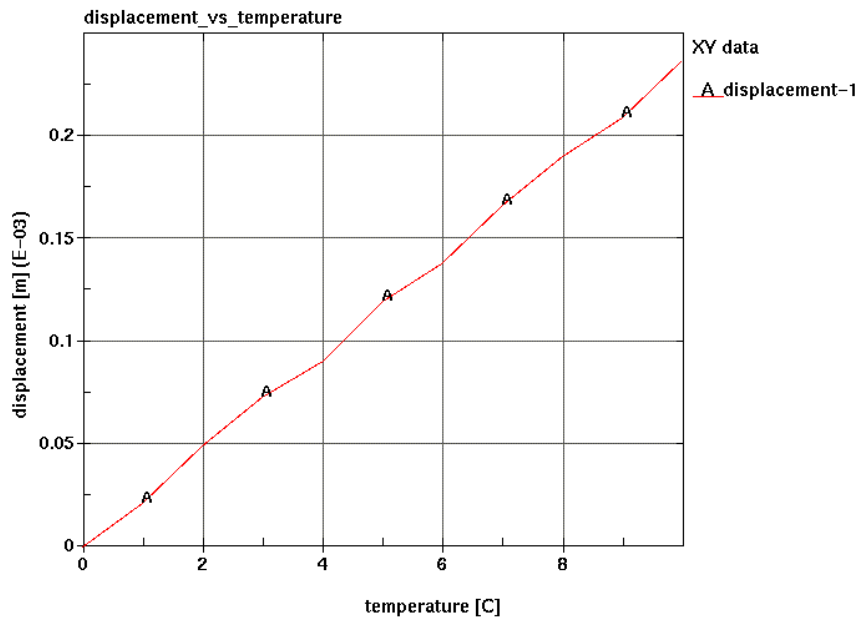
**Example 4-3 Aluminum cube transient heat transfer analysis (file: th01.k)**

```

*KEYWORD
*TITLE
th01.k - heat transfer - problem 1
$
$----- thermal solution keywords -----
$
*CONTROL_SOLUTION 1
*CONTROL_THERMAL_SOLVER 1
*CONTROL_THERMAL_TIMESTEP 0 1. .1
*DATABASE_TPRINT .1
$
$----- define solution control and output parameters -----
$
*CONTROL_TERMINATION 1.
*DATABASE_BINARY_D3PLOT .1
$
$----- define model geometry and material parameters -----
$
*PART
aluminum block
1 1
*SECTION_SOLID
1
*MAT_THERMAL_ISOTROPIC
1 2700. 0 2.43e+07
904. 222.
*NODE
1 0. 0. 0. 0. 7 7
2 1. 0. 0. 0. 5 0
3 1. 1. 0. 0. 3 0
4 0. 1. 0. 0. 6 0
5 0. 0. 0. 1. 4 0
6 1. 0. 0. 1. 2 0
7 1. 1. 1. 0 0
8 0. 1. 1. 1. 1 0
*ELEMENT_SOLID
1 1 1 2 3 4 5 6 7 8
*END
    
```

#### 4.4 Coupled thermal-stress analysis (problem cp01.k)

In this problem, the cube is allowed to expand due to the temperature increase from internal heat generation. Keywords from the mechanical problem defined in section 4.1 and the thermal problem defined in section 4.3 are combined to define this thermal-stress problem. The keyword `*MAT_ELASTIC_PLASTIC_THERMAL` is used to define a material with a thermal coefficient of expansion. For this problem,  $\alpha = 23.6 \cdot 10^{-6}$  m/m C. The aluminum blocks starts out at 0C (the default initial condition) and heats up 10C over the 1 second time interval (see section 1.3 above). The x displacement of node 7 versus temperature increase as calculated by LS-DYNA is shown in the figure below. The curve is not smooth due to numerical noise in the solution because we are only using 1 element. The analytical solution is shown in the box.



$$\begin{aligned} \Delta x &= \alpha \Delta T \\ &= (23.6e-06)(10) \\ &= 23.6e-05 \end{aligned}$$

The keyword input for this problem is shown below. Important things to note are:

- `*CONTROL_SOLUTION` is set to 2. This defines the problem as a coupled thermal stress analysis.
- Defining both mechanical and thermal properties.
- Using a mechanical constitutive model (`*MAT_ELASTIC_PLASTIC_THERMAL`) that allows entry of a thermal coefficient of expansion and mechanical properties that are a function of temperature.
- The mechanical and thermal time steps are independent. For this problem, we are using the default explicit mechanical time step calculated by the code and have specified a thermal time step of 0.1 sec. Therefore, many mechanical time steps will be taken for every thermal time step.



**Example 4-4 Aluminum cube coupled thermal-stress solution (file: cp01.k)**

```

*KEYWORD
*TITLE
cp01.k - coupled thermal stress analysis - problem 1
$
$----- define solution control and output parameters -----
$
$
*CONTROL_SOLUTION 2
*CONTROL_TERMINATION 1.
*DATABASE_BINARY_D3PLOT
  .1
$
$----- thermal solution keywords -----
$
$
*CONTROL_THERMAL_SOLVER
  1
*CONTROL_THERMAL_TIMESTEP
  0 1. .1
*DATABASE_TPRINT
  .1
$
$----- define model geometry and material parameters -----
$
$
*PART
aluminum block
  1 1 1
*SECTION_SOLID
  1
*MAT_ELASTIC_PLASTIC_THERMAL
  1 2700.
  0. 100.
  70.e+09 70.e+09
  .3 .3
  23.6e-06 23.6e-06
*MAT_THERMAL_ISOTROPIC
  1 2700. 0 2.44e+07
  904. 222.
*NODE
  1 0. 0. 0. 7 7
  2 1. 0. 0. 5 0
  3 1. 1. 0. 3 0
  4 0. 1. 0. 6 0
  5 0. 0. 1. 4 0
  6 1. 0. 1. 2 0
  7 1. 1. 1. 0 0
  8 0. 1. 1. 1 0
*ELEMENT_SOLID
  1 1 1 2 3 4 5 6 7 8
*END

```