On the evaluation of thermal expansion coefficients

Number SP-WA-17-TN-F38B	Rev. A
Date 29-Jun-2017	Hash 808c6c4
Document type Technical Note	Pages 8

Author
Jeremy Theler jeremy@seamplex.com
Reviewed by
Angus Ramsay angus_ramsay@ramsay-maunder.co.uk

Abstract

A potential issue with thermal expansion calculations in a privative commercial finite-element code was raised by Ramsay Maunder & Associates when employing material data from an ASME report. This technical note revisits the problem of thermal expansion coefficients and performs some mathematical operations on the ASME data to further investigate the potential issue. The open source tool wasora was used to numerically analyze the data and the relationships between different variables, illustrating how simple this task can be when employing the UNIX philosophy for carrying out engineering endeavors.

Keywords

ASME, thermal expansion, wasora

Revision history

Rev.	Date	Author	
A	29-Jun-2017	jeremy theler	First issue

1 Introduction

When solving thermal-mechanical problems it is customary to used thermal expansion coefficients to take into account the strains induced by changes in the material temperature. (see for example reference [1]). These coefficients α are defined as the partial derivative of the strain ϵ with respect to temperature T such that differential relationships like

$$d\epsilon = \frac{\partial \epsilon}{\partial T}\, dT = \alpha \cdot dT$$

can be written. This derivative α is called the *instantaneous* coefficient. If the strain is not linear with respect to the temperature (which is the most common case), then α depends on T. Therefore, when dealing with finite temperature increments $\Delta T = T - T_0$, mean values for the expansion coefficients have to be used:

$$\Delta\epsilon = \int_{\epsilon_0}^{\epsilon} d\epsilon' = \int_{T_0}^{T} \frac{\partial \epsilon}{\partial T'}\, dT' = \int_{T_0}^{T} \alpha(T')\, dT'$$

We can multiply and divide by ΔT to obtain

$$\int_{T_0}^{T} \alpha(T')\, dT' \cdot \frac{\Delta T}{\Delta T} = \bar{\alpha}(T) \cdot \Delta T$$

where the mean expansion coefficient for the temperature range $[T_0, T]$ is

$$\bar{\alpha}(T) = \frac{\displaystyle\int_{T_0}^{T} \alpha(T')\, dT'}{T - T_0}$$

This is of course the classical calculus result of the mean value of a continuous one-dimensional function in a certain range.

Usually, material properties are given in the form of instantaneous coefficients as a function of the temperature in the form of discrete data pairs. Computational codes ought to use these coefficients α to take into account the thermal strains by computing mean coefficients $\bar{\alpha}$. An issue has been recently raised regarding the way a certain non-free[1] closed-source commercial software—whose name we will not disclose because we stand firmly against privative software, especially regarding engineering analysis—performed this conversion [2]. Moreover, if such software had granted access to its source code, then a far more clear insight could have been provided.

This report investigates a set of thermal expansion data from ASME [3] and checks what the relationship between the different columns (namely thermal strain, instantaneous and mean coefficients) is by adaptively differentiating and integrating the interpolated ASME data with the open source UNIX-based tool wasora [4].

2 Problem statement

Let $\epsilon(T)$ be the linear thermal expansion of a given material in a certain direction when heating a piece of such material from an initial temperature T_0 up to T so that $\epsilon(T_0) = 0$. Let us define the following relationships:

$$\alpha(T) = \frac{\partial \epsilon}{\partial T} \tag{1}$$

$$\bar{\alpha}(T) = \frac{\displaystyle\int_{T_0}^{T} \alpha(T')\, dT'}{T - T_0} \tag{2}$$

$$\bar{\alpha}(T) = \frac{\epsilon(T)}{T - T_0} \tag{3}$$

$$\epsilon(T) = \int_{T_0}^{T} \alpha(T')\, dT' \tag{4}$$

Table 1 gives thermal expansion data for aluminum alloys. From note (b), we can see that

[1] Non-free as in “non-free speech,” see `https://www.gnu.org/philosophy/free-sw.en.html`.

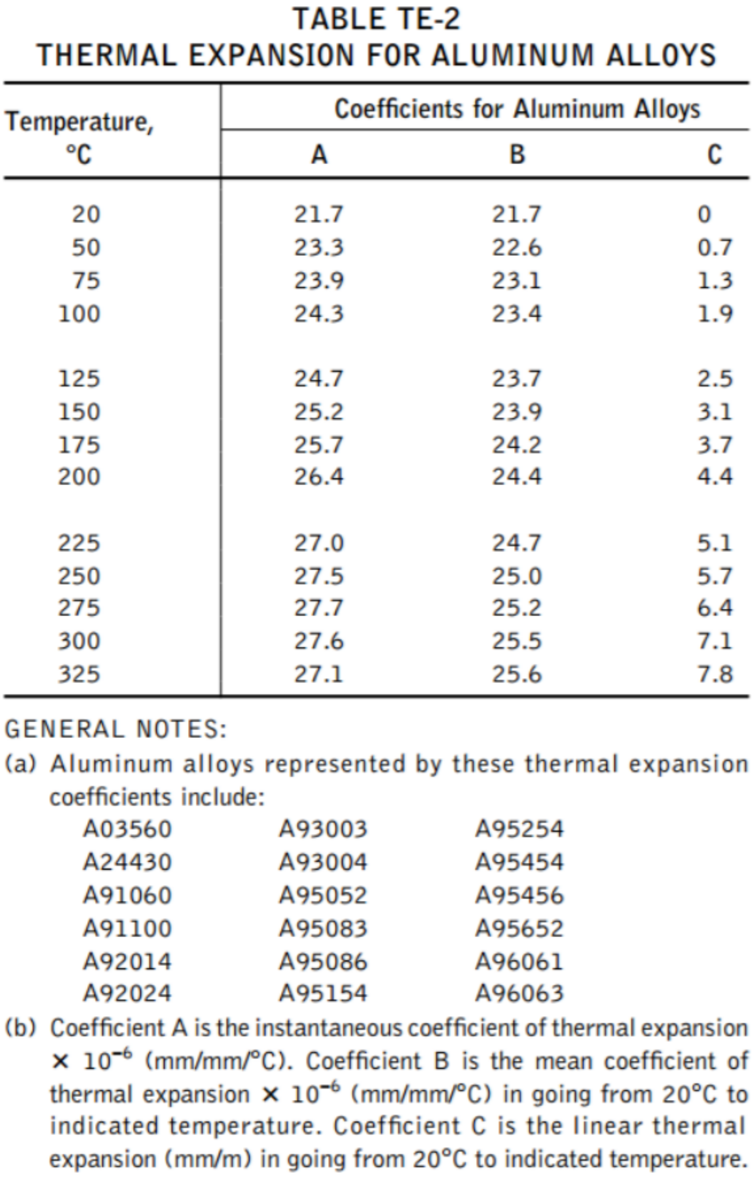

TABLE TE-2
THERMAL EXPANSION FOR ALUMINUM ALLOYS

Temperature, °C	Coefficients for Aluminum Alloys		
	A	B	C
20	21.7	21.7	0
50	23.3	22.6	0.7
75	23.9	23.1	1.3
100	24.3	23.4	1.9
125	24.7	23.7	2.5
150	25.2	23.9	3.1
175	25.7	24.2	3.7
200	26.4	24.4	4.4
225	27.0	24.7	5.1
250	27.5	25.0	5.7
275	27.7	25.2	6.4
300	27.6	25.5	7.1
325	27.1	25.6	7.8

GENERAL NOTES:

(a) Aluminum alloys represented by these thermal expansion coefficients include:

A03560	A93003	A95254
A24430	A93004	A95454
A91060	A95052	A95456
A91100	A95083	A95652
A92014	A95086	A96061
A92024	A95154	A96063

(b) Coefficient A is the instantaneous coefficient of thermal expansion × 10^{-6} (mm/mm/°C). Coefficient B is the mean coefficient of thermal expansion × 10^{-6} (mm/mm/°C) in going from 20°C to indicated temperature. Coefficient C is the linear thermal expansion (mm/m) in going from 20°C to indicated temperature.

Table 1: Temperature-dependent thermal expansion properties for aluminum alloys. Table from [3, p. 714]

$$
\begin{aligned}
A(T) &= \alpha(T) \\
B(T) &= \bar{\alpha}(T) \\
C(T) &= \epsilon(T)
\end{aligned}
$$

So we can compute column B out of any of the other two using equations (2) or (3) provided we interpolate the data in order to obtain continuous (and preferably differentiable) functions.

3 Proposed solution

Although slightly different results are expected with different interpolation methods, the overall behavior should be consistent. We can investigate this issue with the free and open source tool wasora [4]. We start with the data from table 1 in a file called `asme.dat`:

Listing 1: asme.dat

```
# temp  A       B       C
20      21.7    21.7    0
50      23.3    22.6    0.7
75      23.9    23.1    1.3
100     24.3    23.4    1.9
125     24.7    23.7    2.5
150     25.2    23.9    3.1
175     25.7    24.2    3.7
200     26.4    24.4    4.4
225     27.0    24.7    5.1
250     27.5    25.0    5.7
275     27.7    25.2    6.4
300     27.6    25.5    7.1
325     27.1    25.6    7.8
```

Then we use the following input file for wasora which reads the data, interpolates using a certain method (provided in the command line) and computes $B(T)$ out of $A(T)$ and $C(T)$:

```
# the commandline argument tells us how to interpolate
DEFAULT_ARGUMENT_VALUE 1 linear

# read columns from file and interpolate
FUNCTION A(T) FILE_PATH asme.dat COLUMNS 1 2 INTERPOLATION $1
FUNCTION B(T) FILE_PATH asme.dat COLUMNS 1 3 INTERPOLATION $1
FUNCTION C(T) FILE_PATH asme.dat COLUMNS 1 4 INTERPOLATION $1

VAR T'     # dummy variable for integration
T0 = A_a   # lowest definition point for function A(T)

# compute B from A and C
B_fromA(T) := integral(A(T'), T', T0, T)/(T-T0)
B_fromC(T) := 1e3*C(T)/(T-T0)    # C is in mm/m, hence the 1e3

# compute other relationships
C_fromA(T) := integral(A(T'), T', T0, T)
A_fromC(T) := 1e3*derivative(C(T'), T', T)

# write results (the 0.1 is to avoid division by zero)
PRINT_FUNCTION A B C A_fromC B_fromA B_fromC  C_fromA MIN A_a+0.1 MAX A_b NSTEPS 1000
```

Listing 2: mean.was

Wasora uses the GNU Scientific Library [5] for the low-level numerics, so we can use any of the one-dimensional interpolation methods it provides. In particular we will try out:

1. linear
2. splines
3. akima
4. steffen

Taking into account all these considerations and exploiting the script-friendly nature of wasora and the UNIX philosophy it relies on, we can use wasora to compute the results and Pyxplot to plot them with this script:

Listing 3: run.sh

```
for i in linear splines akima steffen; do
  wasora mean.was $i > $i.dat
  sed s/xxx/$i/g mean.ppl | pyxplot
done
```

Figures 1–4 show the results. We can see that $B(T)$ computed from $A(T)$ with equation (2) gives reasonable results, whilst using (3) to compute $B(T)$ out of $C(T)$ does not give the expected results. Therefore, the column C in table 1 is not consistent with the other two.

4 Conclusions

The obtained results are essentially the same as the ones reported with the closed-source privative software in the reference that raised the original issue [2]. Even more, it seems that such software is using plain linear interpolation to compute mean coefficients out of the thermal strain data. The apparent contradiction comes from the fact that the ASME data is not fully consistent in the sense that the relationships (1)–(4) do not hold exactly. We do not have technical details about how table 1 was built, probably the strains were measured and then the coefficients were computed. Probably the reported data does not give enough decimal figures to allow more accurate numerical operations on the interpolated data. Or possibly the uncertainties of the experimental methods are such that the first two columns needed a particular processing that we could not reproduce in this note. A further comment by Ramsay Mauder & Associates can be found in reference [6].

In either case, although it seems that the provided data is not consistent, the errors made by the condensation of coefficients out of strains (or conversely) are within the order of magnitude of the expected uncertainties associated to the rest of the material properties and to the numerical methods employed to solve the differential equations that model the original problem, which in turn rely on further assumptions and approximations.

References

[1] Jeremy Theler. *Thermo-elastic expansion of finite cylinders*. Benchmark problem SP-FI-17-BM-5460. Version A. Seamplex, Apr. 12, 2017.

[2] Angus Ramsay. *ANSYS/ASME: Potential Issue with Thermal Expansion Calculations*. Tech. rep. Ramsay Mauder & Associates,.

[3] ASME. *ASME Boiler & Pressure Vessel IICode Part D*. Tech. rep. July 1, 2010.

[4] J. Theler. *Description of the computational tool wasora*. Technical Description SP-WA-15-TD-9E3D. Version B. Seamplex, 2016.

[5] M. Galassi et al. *GNU Scientific Library Reference Manual*. 3rd. ISBN: 0954612078.

[6] Angus Ramsay. *Accuracy of Thermal Expansion Properties in ASME B&PV Code*. Tech. rep. Ramsay Mauder & Associates,.

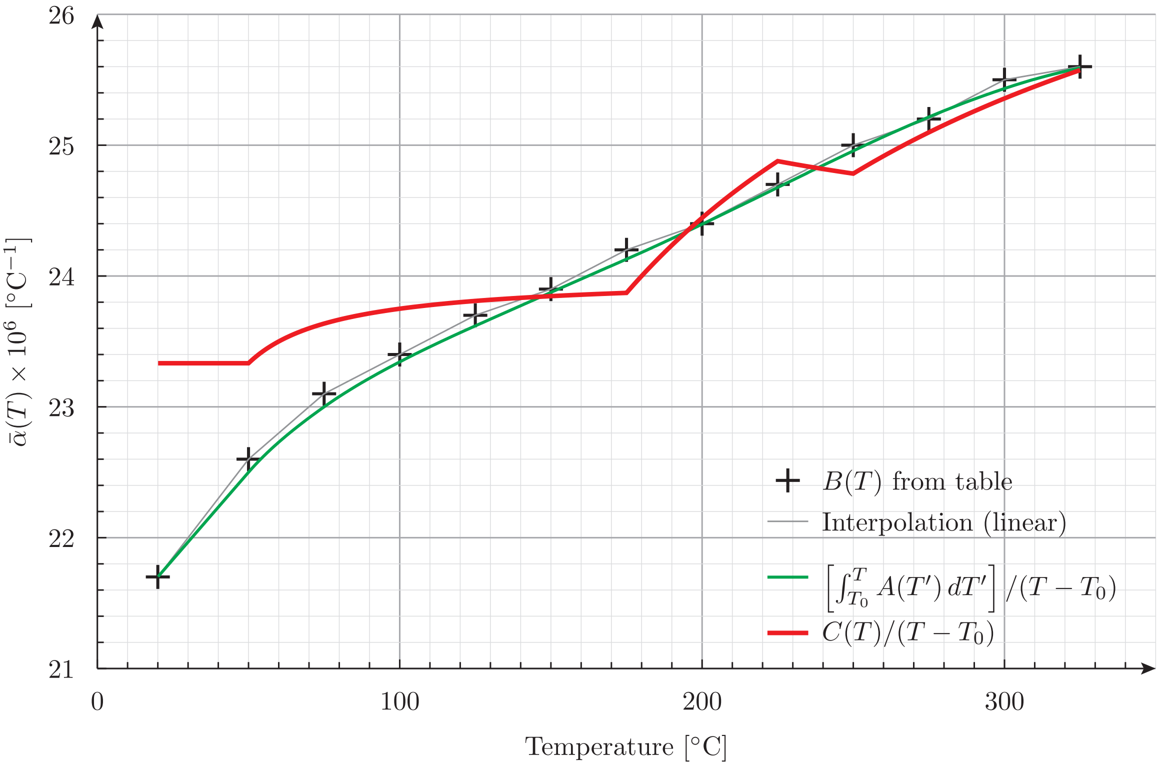

Figure 1: Results using linear interpolation.

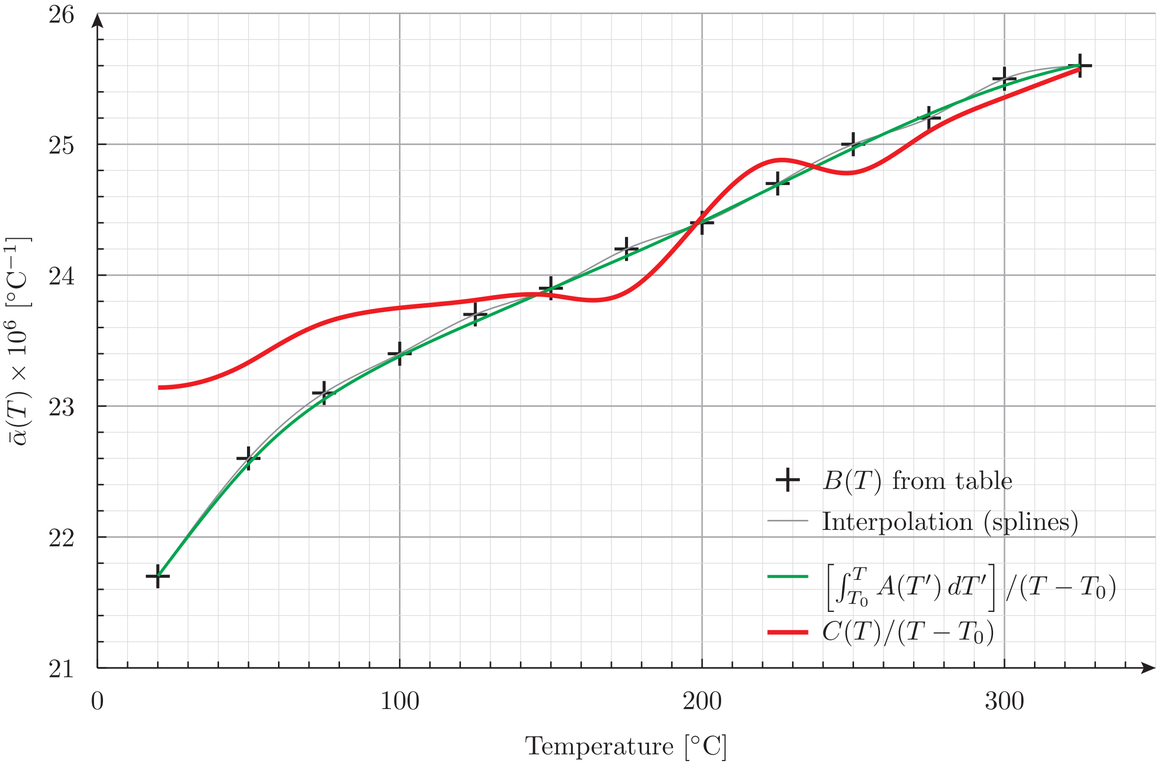

Figure 2: Results using splines interpolation.

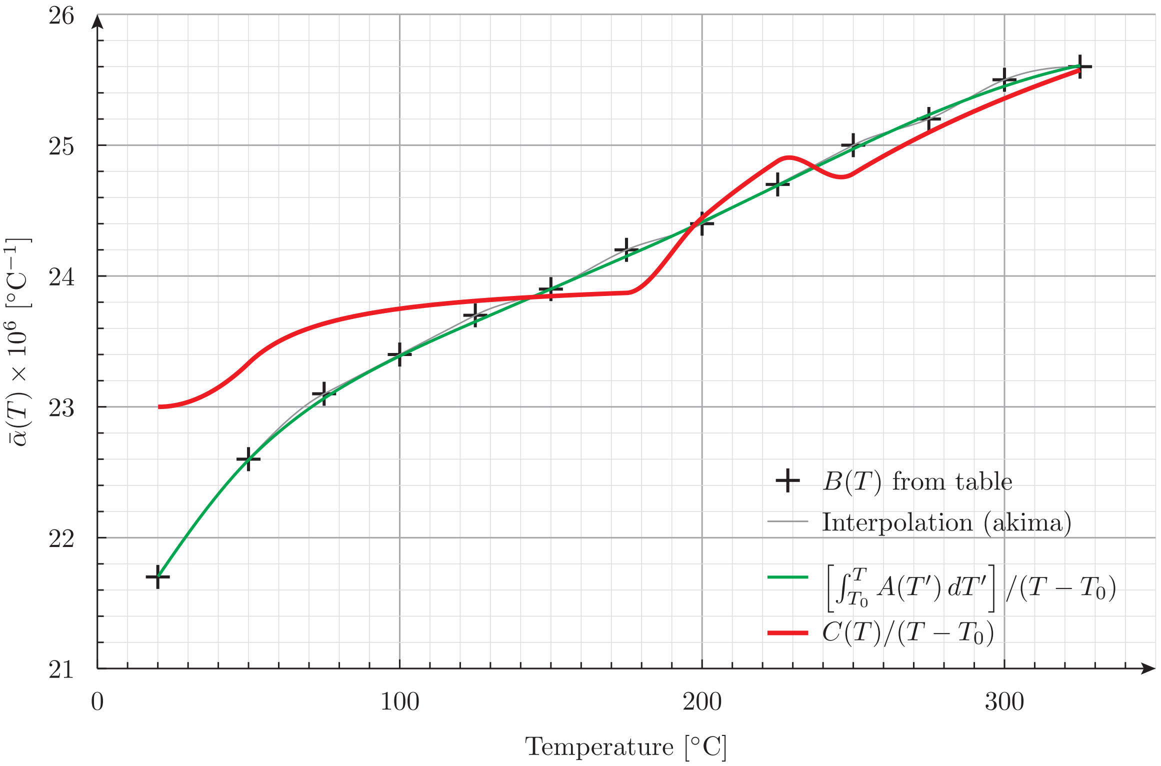

Figure 3: Results using akima interpolation.

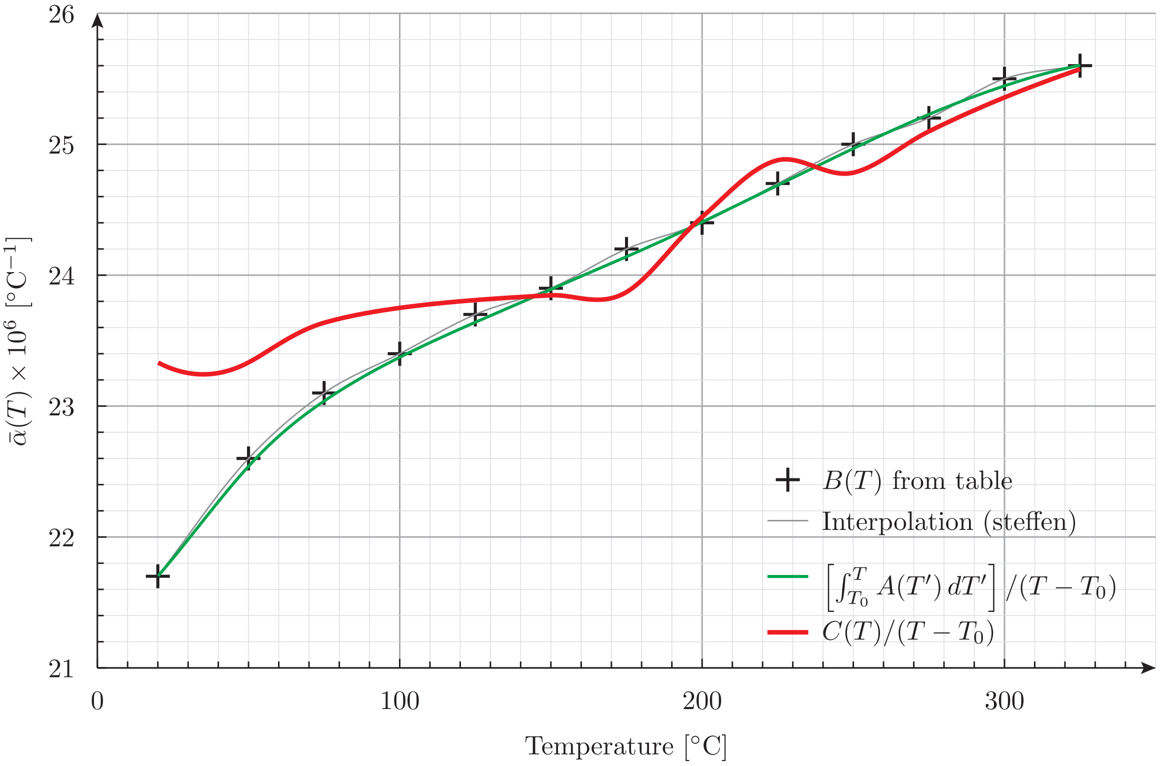

Figure 4: Results using steffen interpolation.