Thermal Parameter Identification Code User Manual

Amit Singh*[1] and Ellad B. Tadmor†[1]

[1]Department of Aerospace Engineering and Mechanics, The University of Minnesota, Minneapolis, Minnesota 55455

October 9, 2014

*sing0335@umn.edu
†tadmor@aem.umn.edu

1 Introduction

This is the User Manual for the MATLAB implementations of the *Thermal Parameter Identification* (TPI) method referred to as the "TPI code". This document is provided as supplemental material to an article on the TPI method by A. Singh and E. B. Tadmor. Here, we give a brief overview of the method and describe the TPI code and its usage.

The TPI method uses a nonlinear regression analysis to find the thermal parameters of either a Fourier, Cattaneo-Vernotte (CV), or Jeffreys-type heat conduction model. The method takes as input the time history of the temperature of different regions (bins) along the length of a one-dimensional beam. The Jeffreys-type model also needs the time history of the heat current of all bins as input. Initially the temperatures of bins are set to a sinusoidal temperature distribution of the form:

$$\theta_n = \theta_0 + \Delta\theta_0 \cos(2\pi n/(N_{\text{bins}} - 1)), \quad 0 \leq n \leq N_{\text{bins}} - 1, \tag{1.1}$$

where θ_0 and $\Delta\theta_0$ are the base temperature and amplitude of the perturbation in the base temperature, respectively. Then the heated nanobeam is allowed to relax under periodic boundary conditions, $\theta(t)|_{n=0} = \theta(t)|_{n=N_{\text{bins}}-1}, t \geq 0$, by removing all thermostats. The simulations are repeated multiple times with different initial positions and momenta of the atoms corresponding to the initial condition described in Eq. (1.1) to obtain multiple realizations (ensembles). Having established a steady-state sinusoidal temperature distribution, the the positions and velocities of the atoms are stored 'nEnsembles' times at intervals of 10^4 MD time-steps to create initial conditions for 'nEnsembles' realizations. Then every realization of the system is allowed to relax under NVE conditions, and the average bin temperatures and average x-component of heat current vectors are recorded at each time step in files with the format specified below. The ensemble files are grouped into blocks (sets) and then a cosine average of temperature and a sine average of the x-component of the heat current vector along the length of the beam are computed. Then the thermal parameters are calculated for each block by performing a nonlinear regression analysis over the average temperature and heat current data. Non-negative thermal parameters obtained by this approach are retained and averaged to predict the best fit parameters. The operation of the TPI code is controlled through a parameter file called 'TPI.INPUT'.

The TPI code performs two processes:

1. **Consolidating temperature profiles**: Temperature profiles in the files provided by the user (in a format described below) are stored by averaging over a time interval 'deltat' defined in the input parameter file. Each file represents a particular realization (ensemble). To reduce noise, these files are grouped into blocks, and the temperature and heat current data is block-averaged. Then the cosine average of the temperature and the sine average of the x-component of the heat current vector along the length of the beam are computed for the available number of profiles.

2. **Parameter extraction process**: Thermal parameters are extracted using a nonlinear regression model. Only non-negative parameters are retained.

2 I/O Files and Parameters

This section describes the contents and format of the input and output files.

2.1 Input Files

The TPI code requires an input file, "TPI.INPUT", which contains the input parameters that define the program operation. This file also contains the name of the files, containing temperature and heat current profiles, to which the regression analysis is applied.

The input file "TPI.INPUT" must have the following lines:

```
model
Jeffreys
nemdFileCols
3
nEnsembles
50
nblocks
5
nemdFileOption
LAMMPS
```

Table 1: TPI code parameters and the corresponding values

Parameters	Values
model	Jeffreys or CV or Fourier
nemdFileCols	2 or 3
nEnsembles	An integer
nblocks	An integer
nemdFileOption	A string value
nprofs	An integer
deltat	A non-negative real number
theta0	A non-negative real number
dtheta0	A non-negative real number
nbins	An integer
binL	A non-negative real number
binVol	A non-negative real number
gamma	A non-negative real number

```
nprofs
100
deltat
0.5
theta0
36.2
dtheta0
3.62
nbins
11
binL
15.975
binVol
7247.69775
gamma
6.176189e-06
```

The file consists of pairs of lines, the TPI code parameter name followed on the next line by its value. The parameters and their possible values are summarized in Table 1 and described below.

model The continuum model used for the parameter extraction. Three

continuum models are supported: the Jeffreys-type, the Cattaneo-Vernotte (CV), and the Fourier model. The corresponding value for this parameter can be either 'Jeffreys', 'CV', or 'Fourier'.

nemdFileCols The total number of columns in the nonequilibrium Molecular Dynamics (NEMD) data files. This parameter must be either 2 or 3. The first, second, and third column correspond to the bin number, bin temperature, and x-component of the bin heat current vector (heat-flux times volume of the bin), respectively. This parameter must be set to 3 for the Jeffreys-type model, since both the temperature and heat current data are required for this model. For either the CV or the Fourier model, its value can be either 2 or 3, since these models only require temperature data. The heat current data (if supplied) is ignored for these models.

nEnsembles Total number of realizations (ensembles) available.

nblocks Number of blocks (sets) that the ensembles will be grouped into. The thermal parameters are extracted for each block and then a mean value is taken for all those blocks which successfully achieve nonlinear minimization.

nemdFileOption This parameter specifies the manner in which the NEMD files containing the temperature data and possibly heat current data for N_{bins} bins are parsed to the code. The parameter can take either 'LAMMPS' or 'MANUAL' as its value. When its value is 'LAMMPS' then the names of the files must be either temp.%d.ensemble or tempCurrent.%d.ensemble, depending upon whether the value of nemdFileCols is 2 or 3. Here '%d' refers to the ensemble number, which can vary from 1 to nEnsembles. Using the LAMMPS script, which can be found with the supplementary materials, these files are automatically generated when the option 'LAMMPS' is used for the NEMD simulation. The format for these files is explained below. When the value of this parameter is 'MANUAL' then the names of all 'nEnsembles' files should be linewise listed followed by the string MANUAL. An example case has been provided with the supplementary materials. It has the 'TPI.INPUT' files for both options – 'LAMMPS' and 'MANUAL'.

nprofs The number of NEMD temperature profiles, N_{profiles}, available in each NEMD ensemble file.

deltat Time difference in picoseconds between two consecutive temperature profiles recorded in the NEMD ensemble files.

theta0 Base or mean temperature, θ_0, of the initial sinusoidal temperature distribution in degrees Kelvin.

dtheta0 Amplitude, $\Delta\theta_0$, of the sinusoidal temperature distribution in degrees Kelvin. A typical value is 10% of θ_0.

nbins Total number of bins N_{bins}.

binL The length of a bin in Å.

binVol The volume of a bin in Å^3.

gamma Volumetric specific heat, γ, in $\text{eV}/(\text{Å}^3\text{K})$.

2.2 Format for the NEMD data file

An NEMD ensemble file begin with three commented lines serving as the title for the file. These lines are ignored by the TPI code for any processing. The file then contains a series of temperature and possibly heat current profiles at consecutive time steps separated by blank lines. Each profile provides for all n bins along the beam, the bin number, bin temperature (in degrees Kelvin), and possibly the x-component of heat current vector of the bin ($J_x = q_x V$, in eVÅ/ps, where q_x is the heat-flux and V is the bin volume), in two or three columns. Thus the file has the following format:

```
#<Title 1>
#<Title 2>
#<Title 3>
1  temperature_1  heat_current_1
2  temperature_2  heat_current_2
.  . . . . . .
.  . . . . . .
n  temperature_n  heat_current_n
<blank line>
1  temperature_1  heat_current_1
2  temperature_2  heat_current_2
.  . . . . . .
```

```
.  . . . . . . . . . . . . . . . .
n  temperature_n  heat_current_n
<blank line>
etc.
```

The third column need not be there when either the CV or the Fourier model are chosen.

2.3 Output Files

Following the successful execution of the code, one of the following output files is generated: "paramJeffreys.txt", "paramCV.txt", or "paramFourier.txt", depending on the thermal model chosen, which contains the extracted thermal parameters. For the Jeffreys-type model, the file has the following format:

```
Printing Jeffreys parameters for each block (set)
=========================================
Case  kappa  tau_T  tau_q  residual norm
=========================================
..     ..     ..     ..     ..
..     ..     ..     ..     ..
================================================
Case 1, 2, 3 refers to (lambda_2)^2 > 0, < 0, = 0.
Case 0 means minimum was not found for the block.
================================================

Printing final Jeffreys-type parameters:

kappa  std_kappa  tau_T  std_tau_T  tau_q  std_tau_q
=====================================================
  ..      ..        ..        ..     ..       ..
=====================================================
```

The first line contains the text "Printing Jeffreys parameters for each block (set)". The second and fourth line are separators. The third line consists of column headings. The first column is the case obtained after nonlinear minimization (see Section III.C of the article for details). The cases 1, 2,

3 refer to $\lambda_2^2 > 0, < 0, = 0$, respectively. Case 0 means that minimization was not possible for the particular block. This is explained in the later lines within separators. Thermal parameters are in the second to fourth column. The last column denotes the residual norm of the nonlinear curve fitting. At the bottom of the file, the averages of all non-negative parameters with their standard errors are reported over all blocks for which minimization was possible. The files "paramCV.txt" and "parametersFourier.txt" have the same format, except that for the CV model only the thermal conductivity and one relaxation time are included, and for the Fourier model only the thermal conductivity is included.

3 Further discussion

It is possible that the TPI code will fail to provide thermal parameters for the given number of blocks and number of ensemble files. The following suggestions may help to obtain results:

- The values of the parameters 'nblocks' and 'nEnsembles' can be changed.
- The initial guesses for the thermal parameters can be changed. Nonlinear regression can be highly sensitive to the initial guesses. A grid of initial guesses is formed for this purpose and nonlinear minimization is performed for all grid values. However, it may occur that these grid values do not lead to acceptable minimization results. In this case, changing the grid values may help.