

Enthalpy Based Thermal Evolution of Loops (EBTEL): A Translation of the EBTEL Model into the C Programming Language

Will T. Barnes

Department of Physics and Astronomy, Rice University, Houston, TX 77251

`will.t.barnes@rice.edu`

1. Introduction

The original EBTEL model developed by Klimchuk et al. (2008) is coded in the Interactive Data Language (IDL), a proprietary software package used primarily for data analysis, particularly in the fields of astronomy and astrophysics. A single run of the EBTEL model using parameters specified in Cargill et al. (2012a) takes approximately 5-10 seconds. When doing a large parameter sweep, possibly over hundreds or thousands of different inputs, shorter runtimes are necessary. Thus, a translation to a faster, more readily available and portable platform is needed. A version of the EBTEL model in the C Programming Language (EBTEL-C from here on) is provided in an effort to allow wider availability, increase portability across operating systems, and allow for shorter runtimes. This new implementation can use both a simple Euler solver as well as a fourth-order Runge-Kutta routine. Through a judicious choice of the time step, a single run of the EBTEL-C model takes a maximum of half of a second and can even complete a full run in less than a one-hundredth of a second given the appropriate input parameters. Comparisons between EBTEL-C and the original IDL model will be shown.

2. Method

2.1. Loop Parameters

A complete derivation of the equations solved by the EBTEL model can be found in Klimchuk et al. (2008) & Cargill et al. (2012a). P , n , and T are computed by solving the following set of equations:

$$\frac{d\bar{P}}{dt} = \frac{2}{3} \left[\bar{Q} - \frac{1}{L} (\mathcal{R}_{TR} + \mathcal{R}_c) \right] \quad (1)$$

$$\frac{d\bar{n}}{dt} = -\frac{c_2}{5c_3 k L \bar{T}} (F_0 + \mathcal{R}_{TR}) \quad (2)$$

$$\bar{P} = 2\bar{n}k\bar{T} \quad (3)$$

where \bar{Q} is the *ad-hoc* heating function, L is the loop half-length, $\mathcal{R}_{TR,C}$ are the radiative cooling rates in the transition region (TR) and corona respectively, c_2 is the ratio of \bar{T} to the apex temperature T_a , c_3 is the ratio of the temperature at the coronal base T_0 to T_a , and F_0 is the heat flux at the coronal base. Note that the system of equations is closed by the equation of state which here is the ideal gas law.

At the onset of heating (i.e. when our heating function begins to increase past the background heating), the temperature begins to rise. The coronal portion of the loop, being very diffuse, is unable to radiate away all of the excess energy provided by the heating function. This in turn leads to a downward heat flux to the more dense TR. When the magnitude of the heat flux exceeds the TR cooling rate (\mathcal{R}_{TR}), the TR is no longer able to radiate away this excess heat and as a result an exchange of material occurs between the corona and TR in the form of plasma flow into the corona. This results in an increase in coronal density which slightly lags the temperature increase. As the coronal density increases, \mathcal{R}_C becomes a more efficient cooling mechanism, resulting in a decrease in temperature and subsequent drop in the heat flux into the TR.

When $|F_0| \leq \mathcal{R}_{TR}$, the coronal density reaches a maximum and then begins to decrease. This is due to the fact that the TR is now able to radiate away any excess energy and so plasma now flows downward from the corona to the TR, draining the coronal loop (chromospheric condensation) (Klimchuk et al. 2008).

The EBTEL model equates this exchange of material between the TR and corona with an enthalpy flux of the evaporating or condensing plasma. Quantitatively, this is seen in the integration of the 1-D hydrodynamic energy equation over the TR:

$$\frac{5}{2}P_0v_0 \approx -F_0 - \mathcal{R}_{TR} \quad (4)$$

Thus, excess heat flux results in a positive enthalpy flux and flow of material from the TR into the corona. Conversely, a deficient heat flux results in a negative enthalpy flux and correspondingly a flow of material back down into the TR.

2.2. Differential Emission Measure

The EBTEL model is also able to calculate the coronal and TR differential emission measure (DEM). The differential emission measure of a single strand of unit cross-sectional area is given by

$$\text{DEM}(T) = n^2 \left(\frac{\partial T}{\partial s} \right)^{-1} \quad (5)$$

where s is the field-aligned coordinate (Klimchuk et al. 2008). In order to solve for $\partial T/\partial s$, we again use the 1-D hydrodynamic energy equation which, after some simplification, can be written

$$\kappa_0 T^{3/2} \left(\frac{\partial T}{\partial s} \right)^2 - 5kJ_0 \frac{\partial T}{\partial s} - \left(\frac{\bar{P}}{2kT} \right) \Lambda(T) = 0 \quad (6)$$

This expression is quadratic in $\partial T/\partial s$. This, combined with the equation of state, allows for the calculation of the DEM in the TR using Eq. 5. An alternative method for the DEM calculation is given in Klimchuk et al. (2008). EBTEL-C can use either method to perform the calculation (see Table 2).

To calculate the coronal DEM, it is assumed that the total emission measure is distributed over the temperature interval $0.74T_a < T_C < T_a$ (Klimchuk et al. 2008). Thus, DEM_C can be approximated using Eq. 5

$$\text{DEM}_C \approx 2n^2 \frac{\delta s}{\delta T} \quad (7)$$

where the factor of 2 is introduced to account for both halves of the loop, $\delta s = L$, and δT is the difference between the endpoints of the DEM interval. Thus, this amounts to dividing the total emission measure, $2n^2L$, by the temperature interval δT .

3. Results

To ensure that the EBTEL-C model output matches that of the 2012 EBTEL code (see Cargill et al. 2012a), a series of test cases was used. These are summarized in Table 1. Both long (75 Mm) and short (25 Mm) loops were used as well as flare and nanoflare heating. Three different heating functions are implemented in the current version of EBTEL: triangular pulse, square pulse, and gaussian pulse. All were taken into account in the test cases. Additionally, all runs used the classical heat flux calculation as well as the $\text{DEM}_{TR}(T)$ calculation outlined in § 2.2. Heat flux limiting can also be used when calculating F_0 . An additional method for calculating $\text{DEM}_{TR}(T)$ can be found in the appendix of Klimchuk et al. (2008). Table 2 provides more information on how to use these alternate calculations.

Case	L (10^9 cm)	H_0 ($\text{erg cm}^{-3} \text{ s}^{-1}$)	t_s (s)	t_H (s)	$T(t=0)$ (MK)	$n(t=0)$ (10^8 cm^{-3})
1	75	1.5×10^{-3}	0	500	0.85	0.36
2	25	10^{-2}	0	200	0.78	1.85
3	25	2	0	200	2.1	18.5
4	25	10^{-2}	0	200	1.6	9.2
5	25	5×10^{-3}	100	200	0.9	1.4
6	25	5×10^{-3}	100	500	0.9	1.4
7	75	5×10^{-3}	100	500	0.9	1.4
			τ_H (s)	t_m (s)		
8	25	10^{-2}	40	200	0.9	1.4
9	25	10^{-2}	300	2000	0.9	1.4

Table 1: Initial conditions and input parameters for all test cases used to compare EBTEL and EBTEL-C solutions. Cases are outlined and discussed in Cargill et al. (2012a,b). Note that cases 1-4 use a triangular heating function, cases 5-7 use a square heating function, and cases 8 and 9 use a gaussian pulse heating function.

3.1. Plasma Properties

In Figs. 1-3, the plasma properties T, P, n , and n_a are shown in addition to the heating function h and the scaling between T and n . In this section, only cases 1, 5, and 8 are shown, each using a different heating function. The plots for the remaining cases can be found in § A. Near exact agreement between EBTEL-C and the IDL implementation of EBTEL can be seen for all cases using varying heating functions for a range of simulation times and initial conditions.

3.2. Differential Emission Measure

In Fig. 4, the DEM is calculated for cases 1-4 using the expressions given in Eqs. 5, 6. The blue line indicates the DEM_{TR} , the red line indicates the coronal DEM, DEM_{cor} and the green line indicates the \log_{10} of the total DEM. Again, good agreement is seen between these solutions as expected given the agreement in the plasma properties.

3.3. Variable Time Step

In addition to the Euler method used in the original EBTEL model, EBTEL-C includes a fourth-order Runge-Kutta (RK) method option (see Table 2). Additionally, the functionality for an **adaptive method** has also been coded and is packaged with the EBTEL-C model though it is not currently integrated into the solver. Fig. 5 shows a comparison between various time steps for cases 1 and 8 (triangular and gaussian pulse) using both the Euler (Figs. 5(a), 5(c)) and the RK (Figs. 5(b), 5(d)) solvers. Using a larger time step has the advantage of even smaller computation times, allowing for large parameter sweeps in very small amounts of time (i.e. Case 1 with $\tau = 10\text{s}$ and the Euler solver takes just over 0.02 s). Here, it can be seen that using the RK solver for intermediate time steps allows for small computation times while still giving **reasonably good agreement**. To compute the error here (color-coded text in Fig. 5), Eq. B1 \times 100 is used with $\xi = T_{max}, n_{max}$. Timesteps that yield $\Delta\xi/\xi \times 100 \geq 1$ are not recommended as these solutions can deviate significantly from the original EBTEL model.

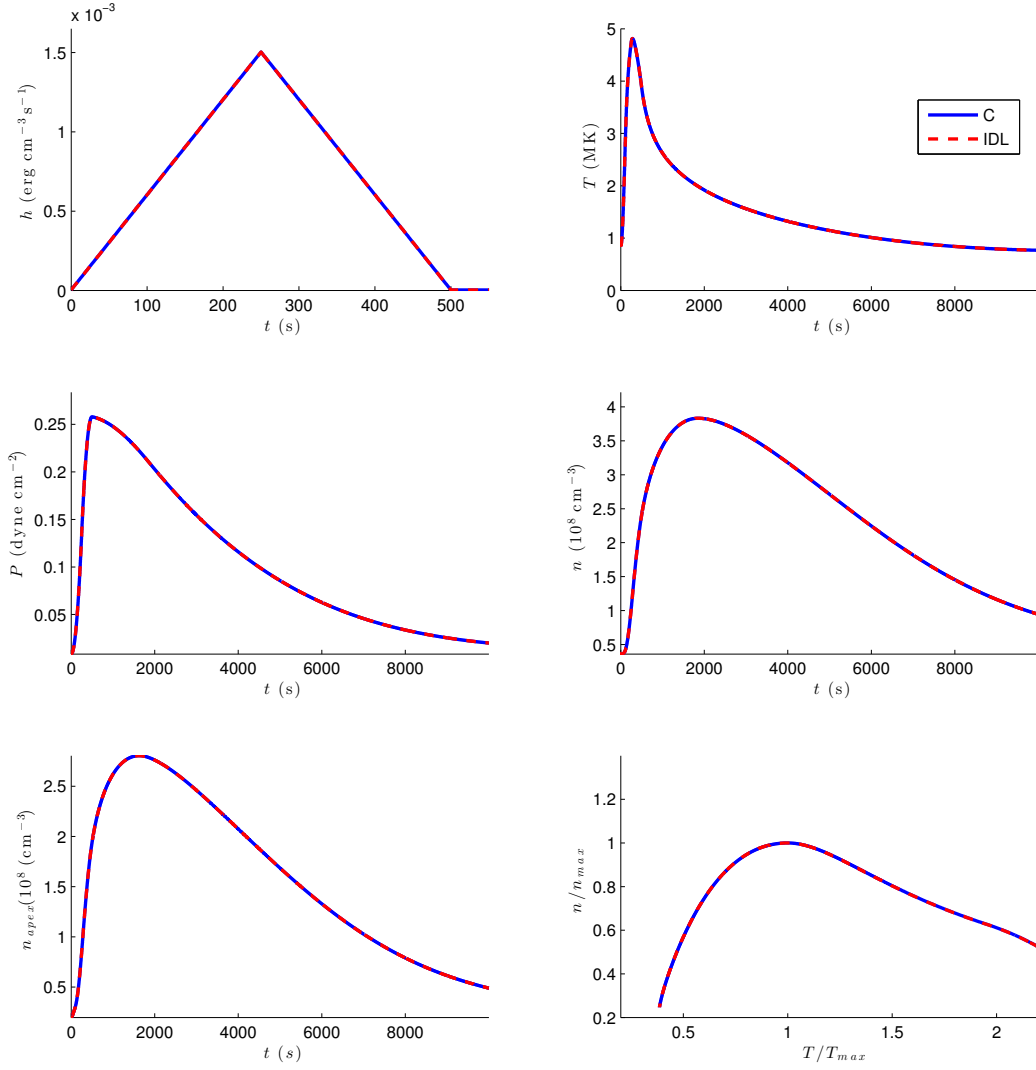


Fig. 1.— h, T, P, n, n_a as functions of time and the $n - T$ scaling are shown for Case 1. The dashed red line corresponds to the EBTEL (IDL) simulation while the blue line corresponds to the EBTEL-C simulation using the Euler solver with $\tau_{au} = 1$ s. Agreement within 10^{-6} (as seen in Fig. 13) is seen between the two models.

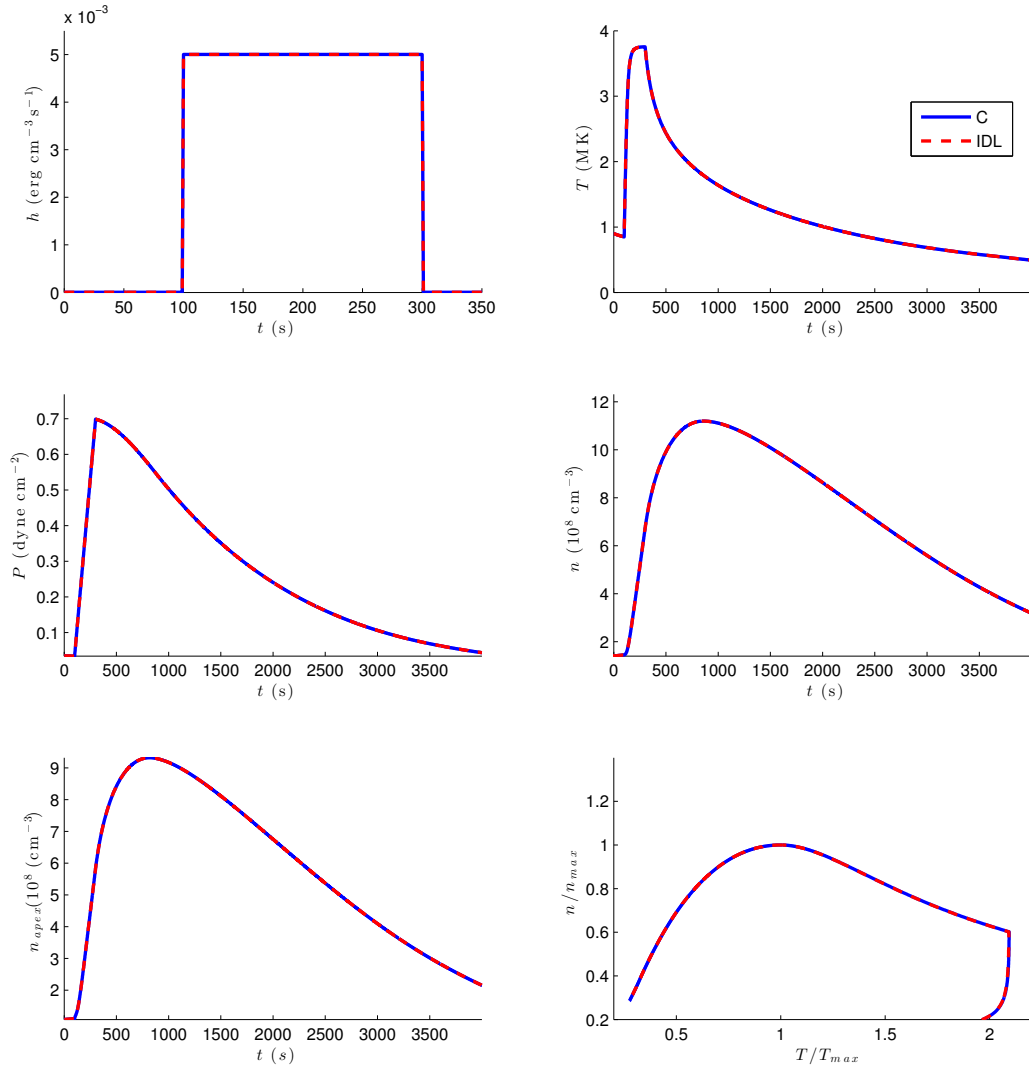


Fig. 2.— Same as Fig. 1 but for Case 5.

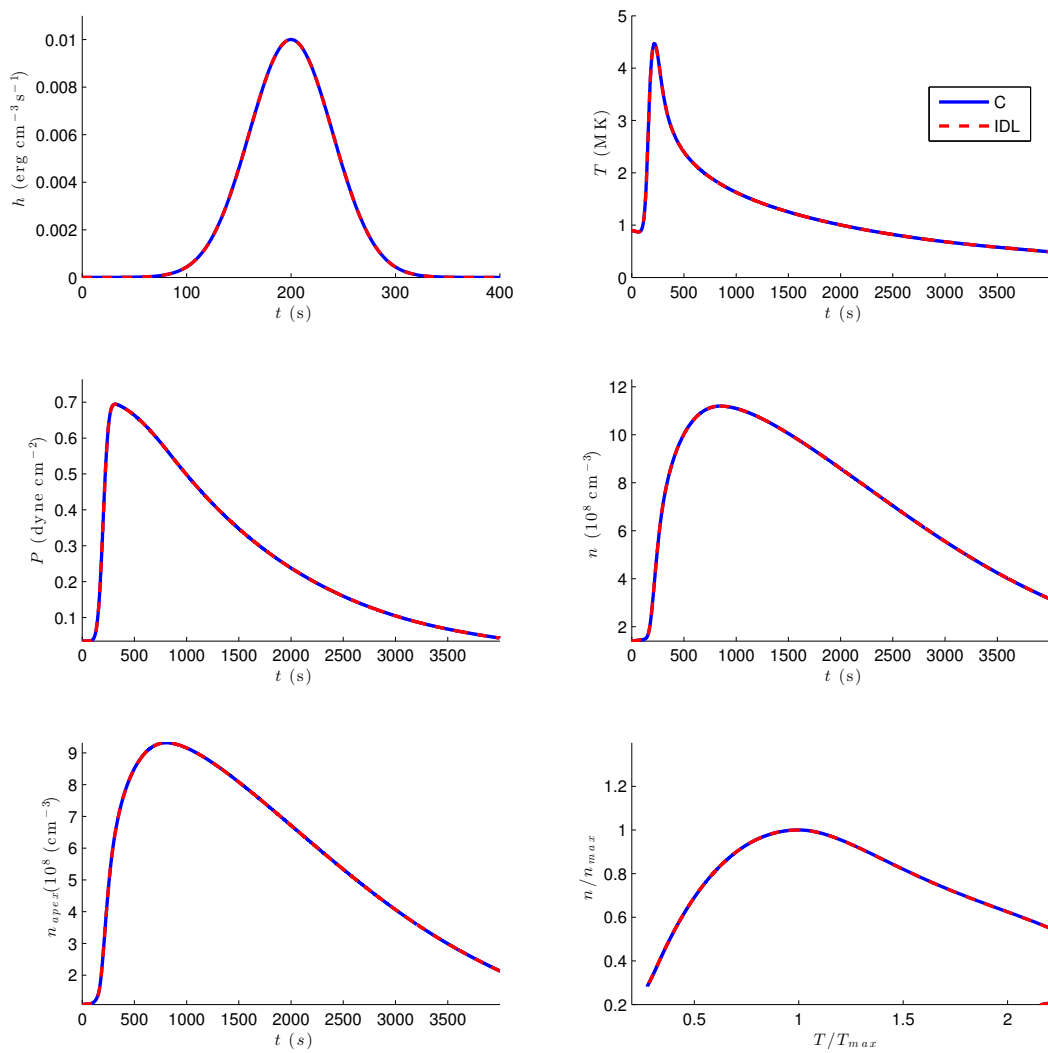


Fig. 3.— Same as Fig. 1 but for Case 8.

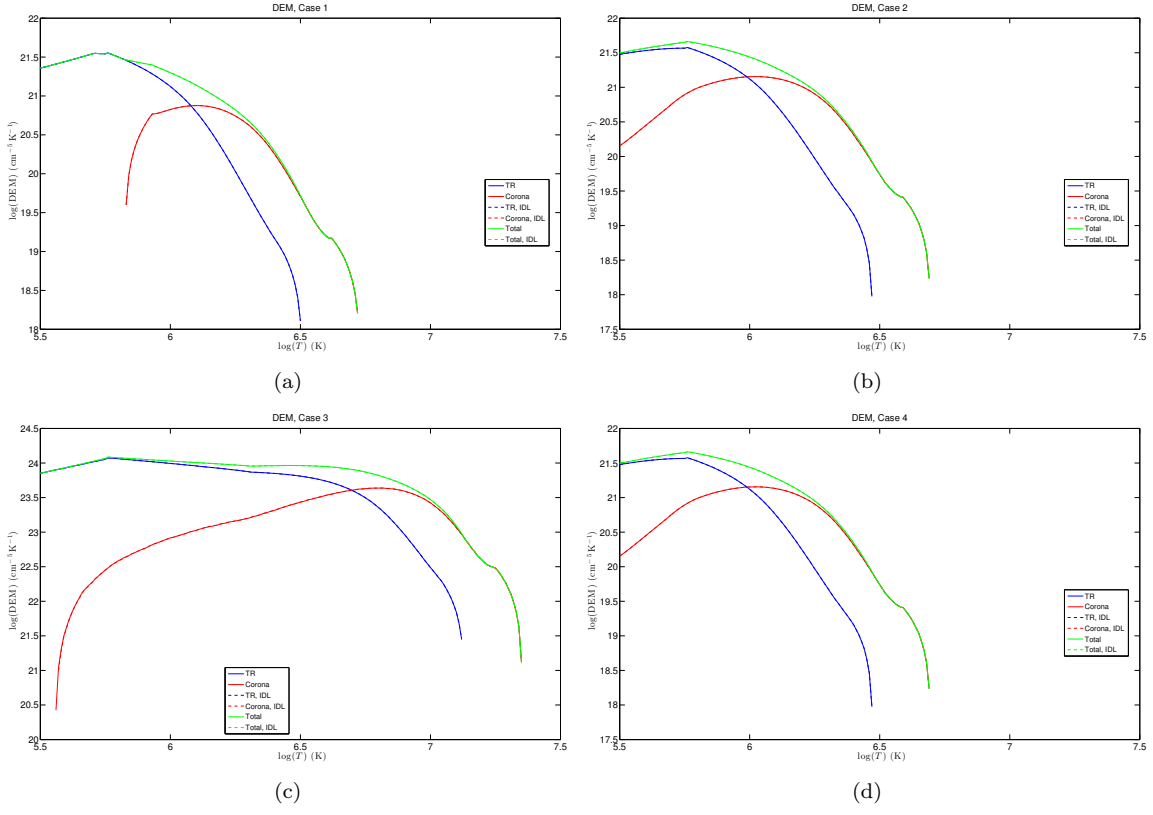


Fig. 4.— Differential emission measure (DEM) for cases 1-4.

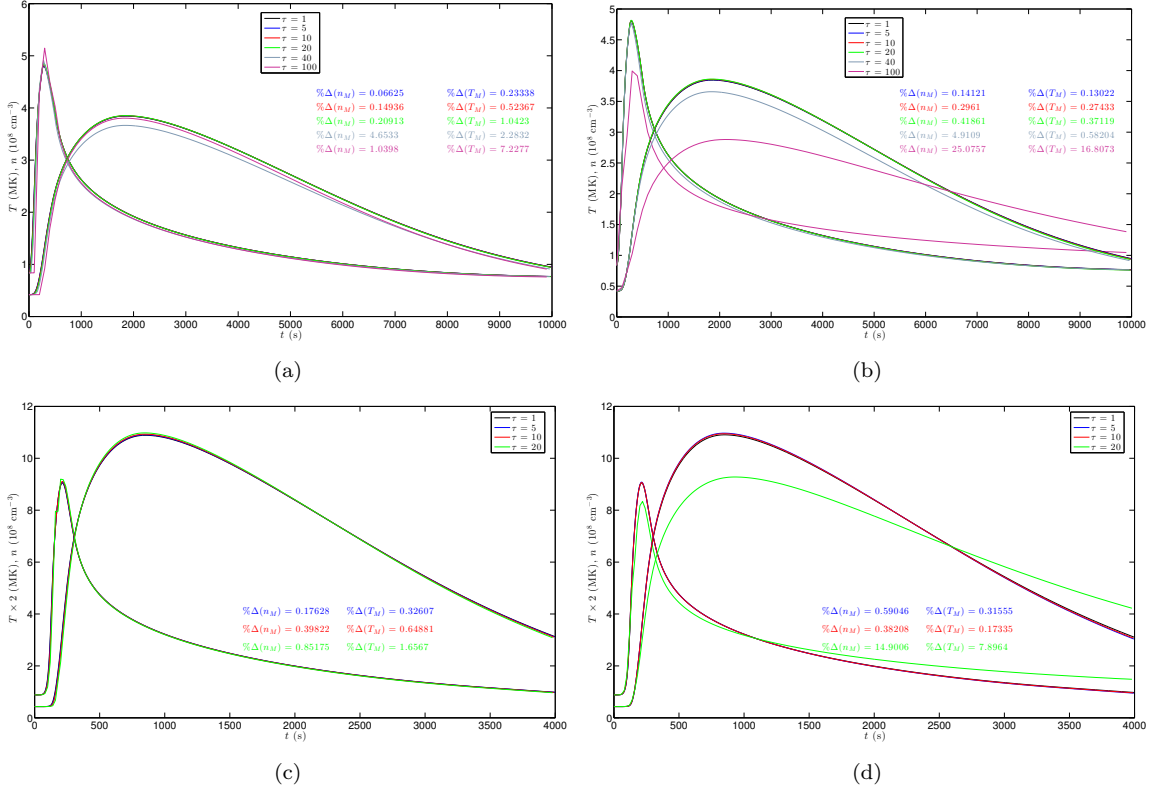


Fig. 5.— Density and pressure solution curves for 5(a) case 1 using the Euler solver and the 5(b) Runge-Kutta solver and 5(c) case 8 using the Euler solver and the 5(d) Runge-Kutta solver. The colors correspond to the varying time step with the black curve in each case corresponding to the $\tau = 1$ case.

A. Additional Parameter Plots

The plots for the remaining test cases are shown in Figs. 6-11. As in § 3.1, all EBTEL-C solutions match well with those solutions produced using the 2012 IDL implementation of EBTEL. Errors for these cases can be seen in Fig. 13. Additionally, the remaining DEM calculations for cases 5-9 are shown in Fig. 12.

B. Error Plots

Error is calculated between the C and IDL solutions. Given that ξ is some parameter which is a function of time t , $\Delta\xi/\xi$ is given by

$$\frac{\Delta\xi}{\xi} = \frac{\xi_C - \xi_{IDL}}{\xi_{IDL}} \quad (B1)$$

where $\xi_{C,IDL}$ are the computed EBTEL-C and EBTEL-IDL parameters respectively. Error plots for all cases outlined in Table 1 can be found in Fig. 13. The Euler solver in EBTEL-C was used when computing this error. Though the details of the the plots are not particularly enlightening, it is important to note that all are on the order of 10^{-6} and oscillate about zero with little linear behavior.

C. Compiling and Running the EBTEL-C Model

This version of the EBTEL model is meant to be much more portable than its IDL counterpart and thus should compile and run fairly easily on any machine with little architecture/OS dependence provided a standard C compiler is installed (i.e `gcc`, `clang`, etc.). The EBTEL-C model is composed of multiple source files each containing multiple functions. See the documentation at the top of each file and above each function for a brief summary of its purpose as well as a list of inputs and outputs. Additionally, the documentation in `ebtel_main.c` contains extensive information on input parameters as well as several other helpful comments.

Also included in the EBTEL-C directory is a `makefile` which can be executed by typing

```
$ make
```

at the command line while inside the EBTEL-C directory. This creates the executable `ebtel` which is run by typing

```
$ ./ebtel
```

at the command line on any Unix-based or Unix-like machine. This will run the EBTEL-C code using input parameters specified in `ebtel_parameters.txt`. See Table 2 for information on how to configure this input file.

REFERENCES

- J. A. Klimchuk, S. Patsourakos, and P. J. Cargill, The Astrophysical Journal **682**, 1351 (2008), ISSN 0004-637X, URL <http://adsabs.harvard.edu/abs/2008ApJ...682.1351K>.
- P. J. Cargill, S. J. Bradshaw, and J. A. Klimchuk, The Astrophysical Journal **752**, 161 (2012a), ISSN 0004-637X, URL <http://adsabs.harvard.edu/abs/2012ApJ...752..161C>.

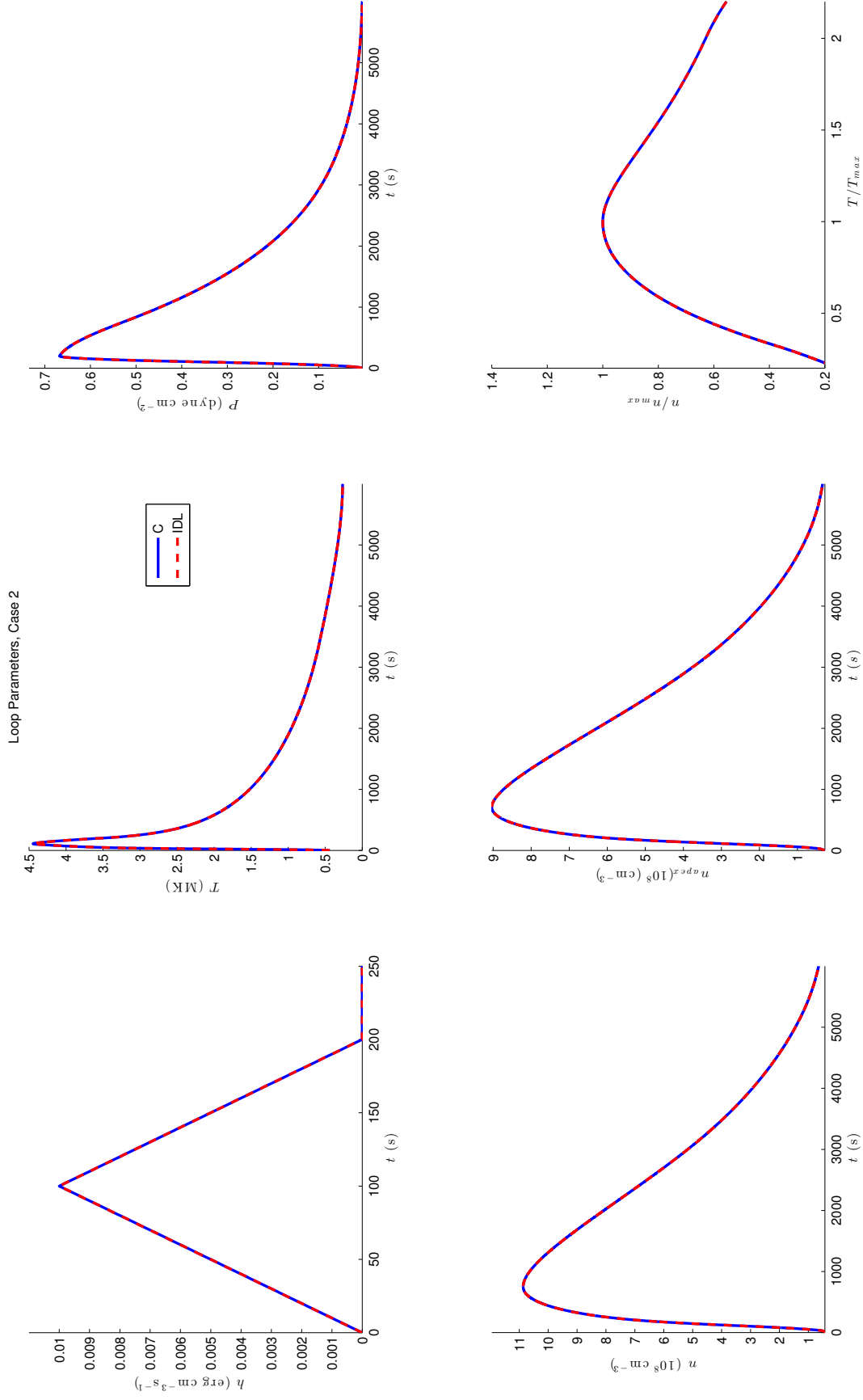


Fig. 6.— Same as Fig. 1 but for Case 2.

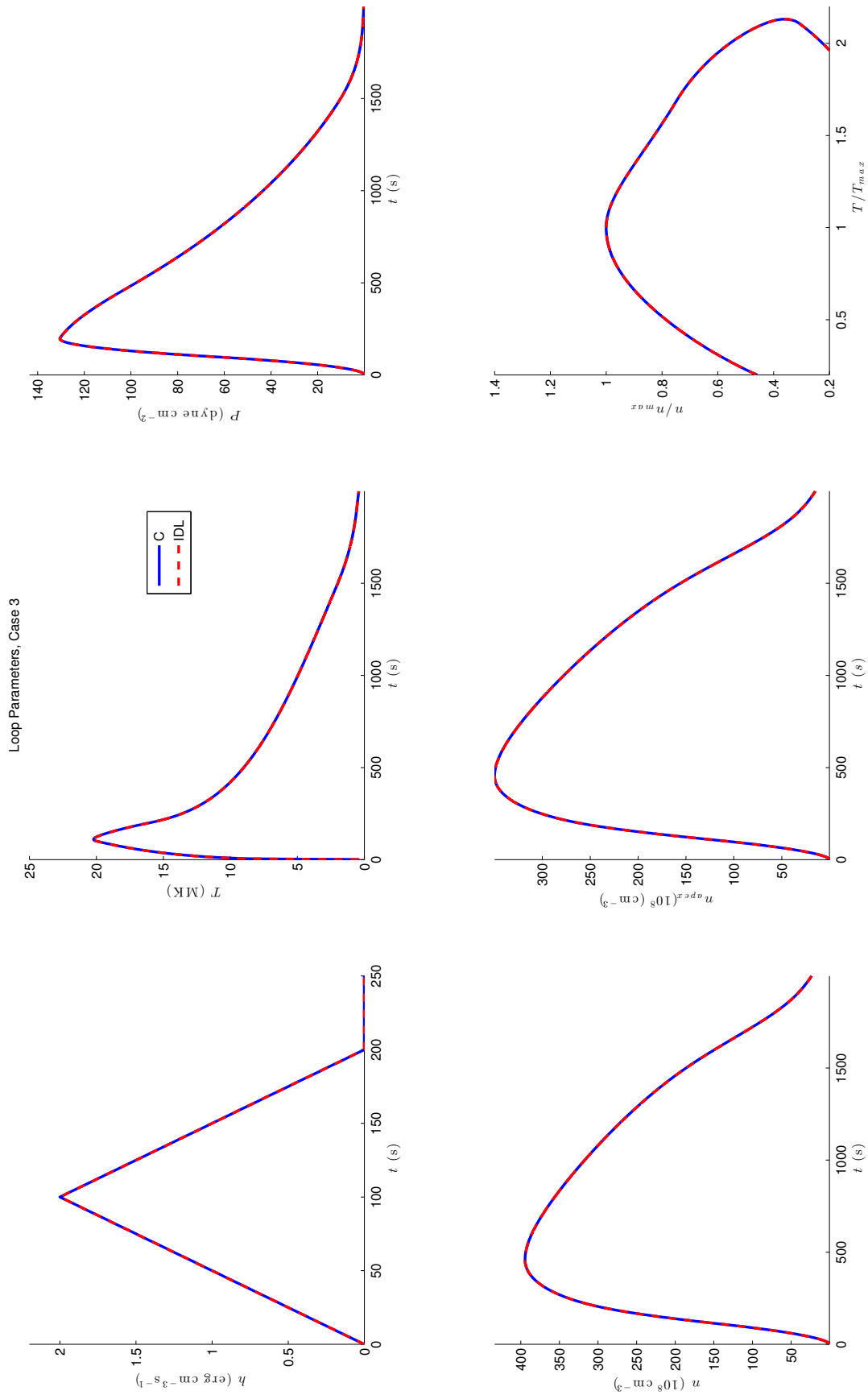


Fig. 7.— Same as Fig. 1 but for Case 3.

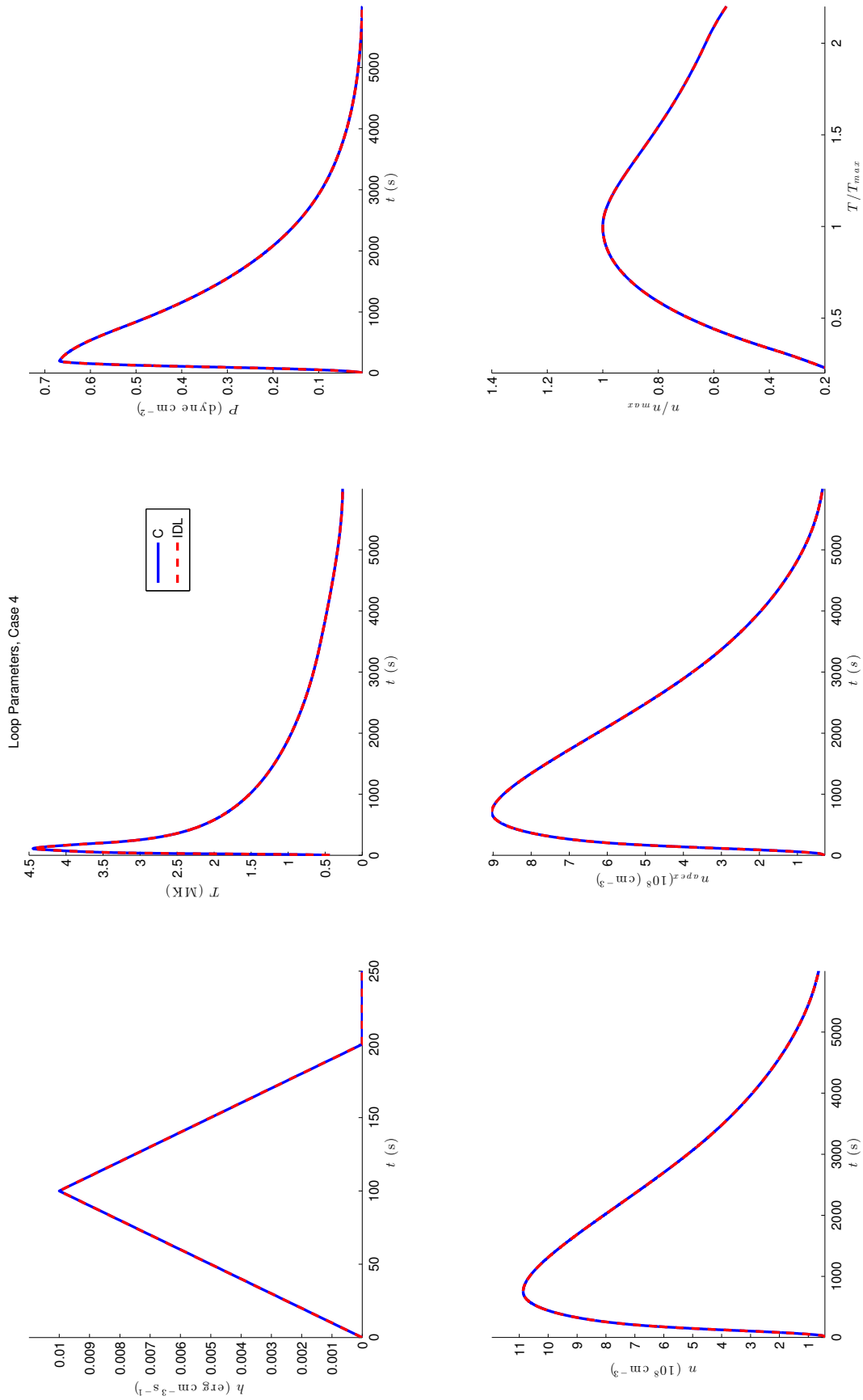


Fig. 8.— Same as Fig. 1 but for Case 4.

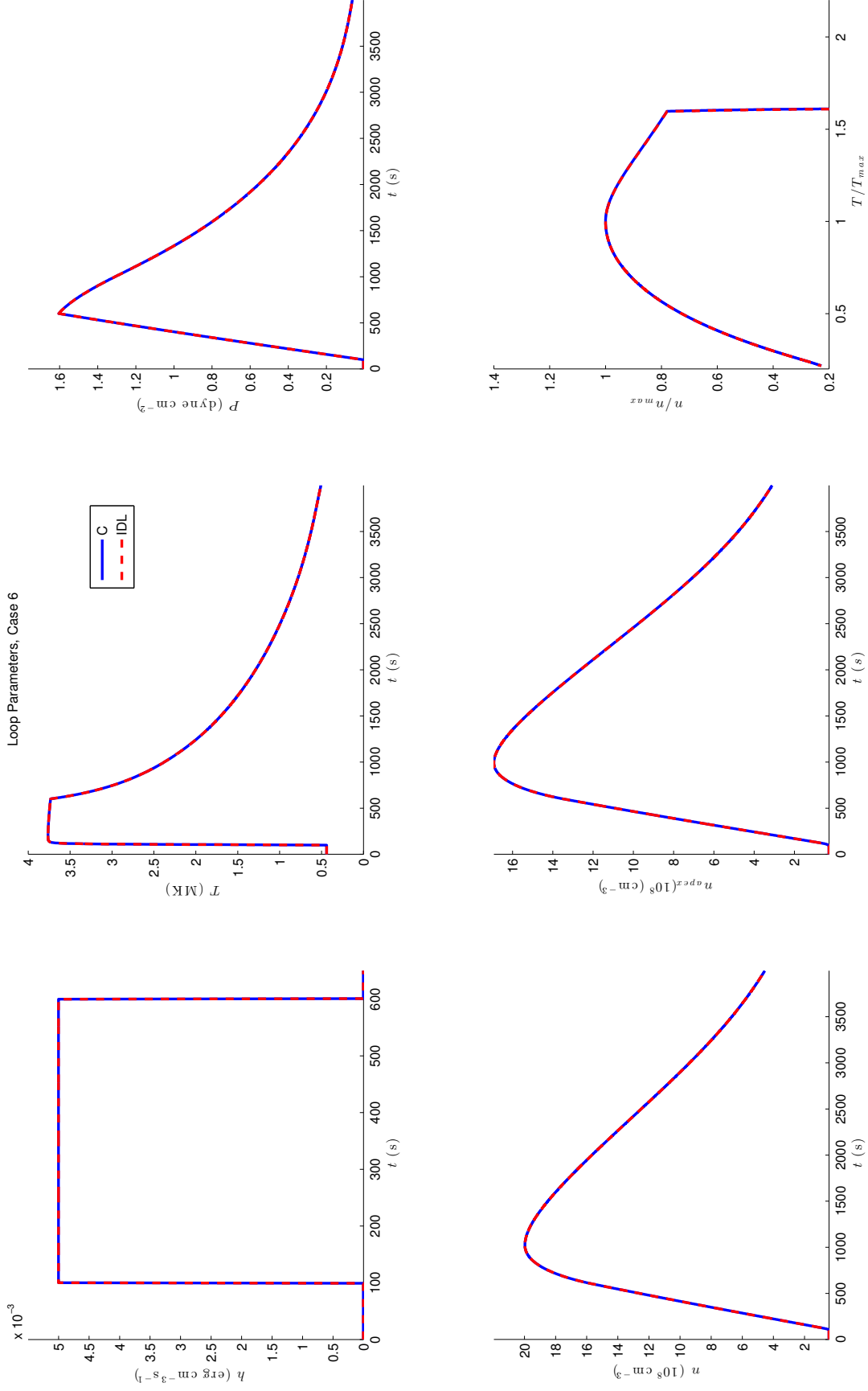


Fig. 9.— Same as Fig. 1 but for Case 6.

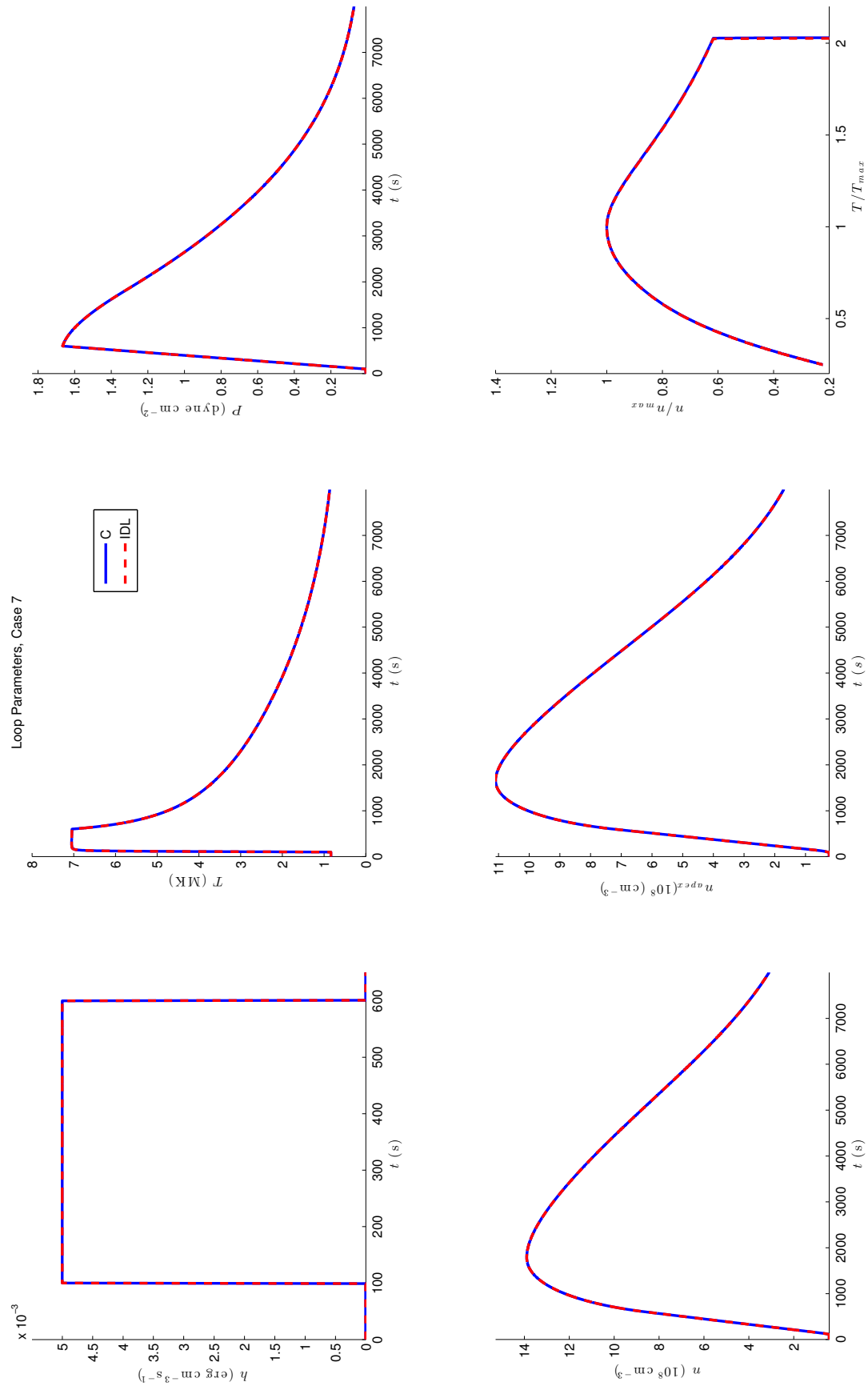


Fig. 10.— Same as Fig. 1 but for Case 7.

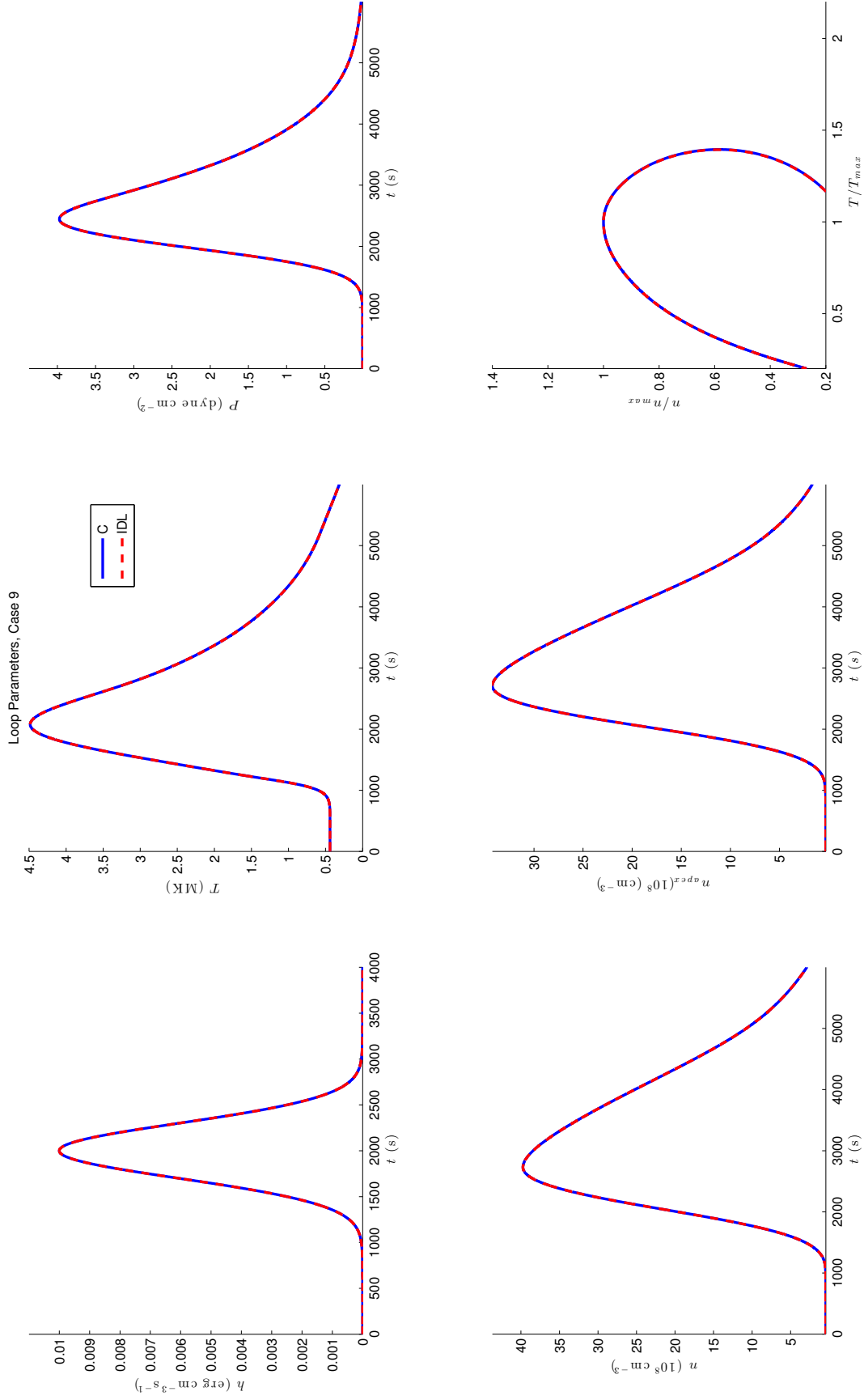


Fig. 11.— Same as Fig. 1 but for Case 9.

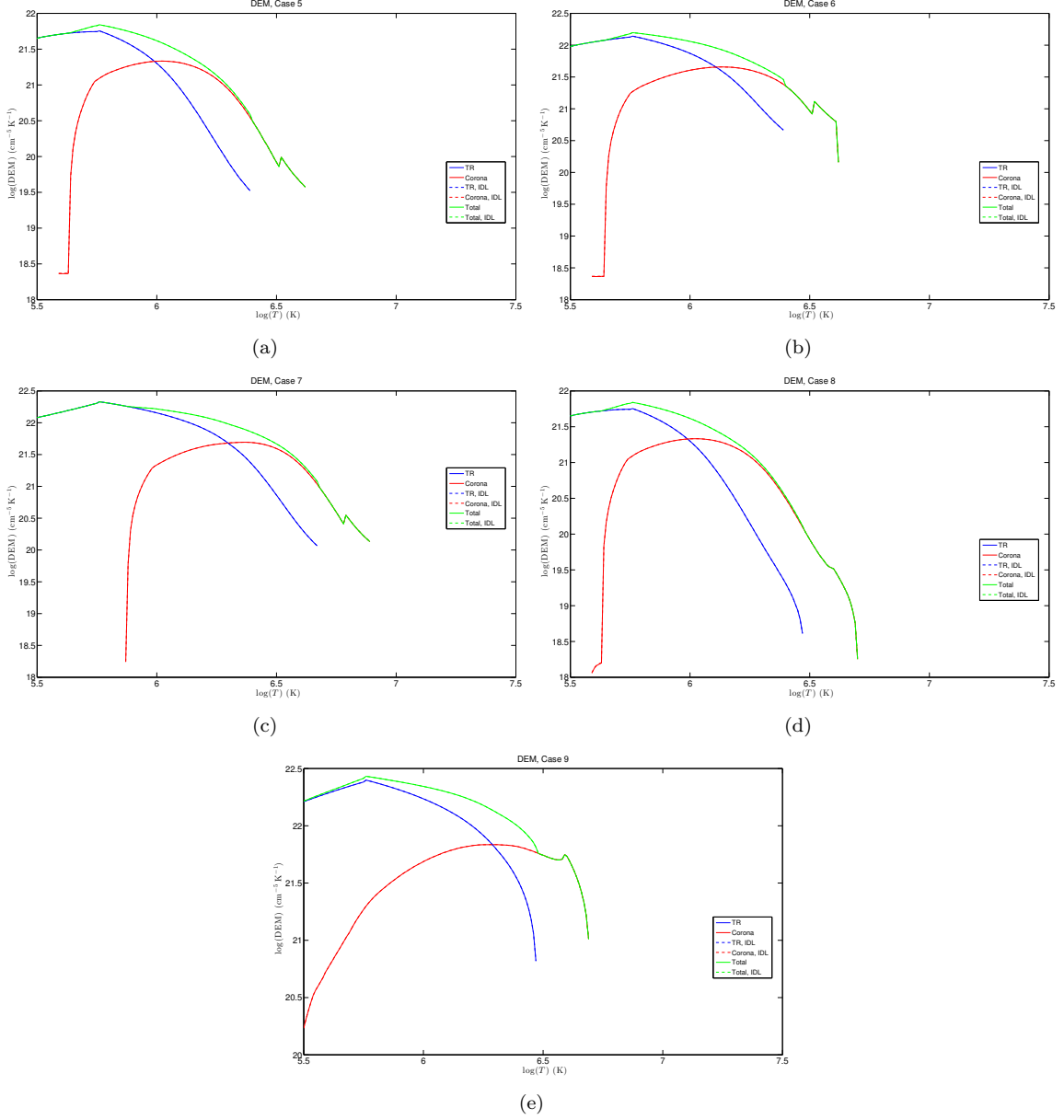


Fig. 12.— Same as Fig. 4 but for cases 5-9.

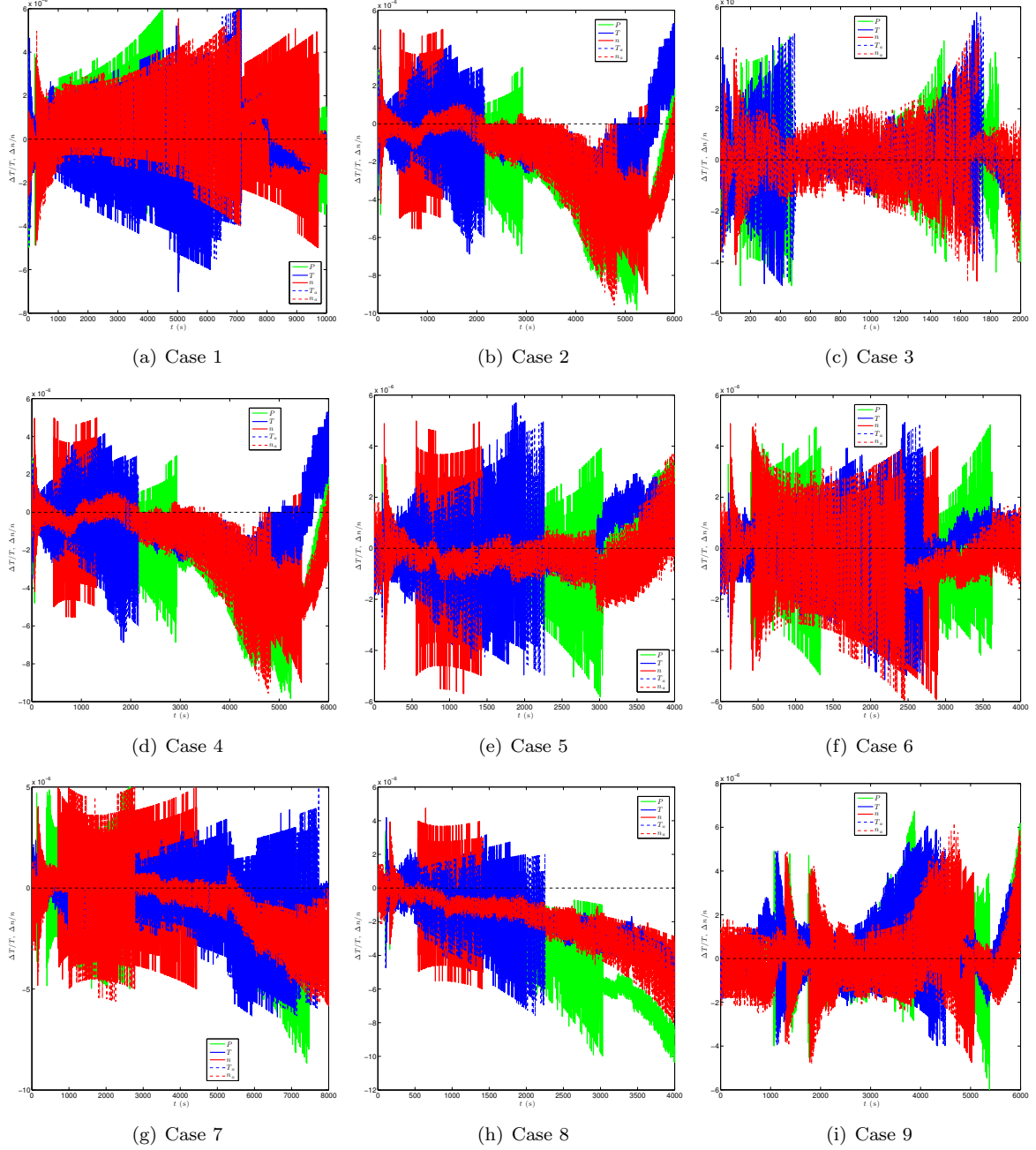


Fig. 13.— Normalized differences for P, n, n_a, T and T_a for all cases versus time t .

Table 2: Structure and description of the parameter input file `ebtel.parameters.txt`.

Parameter	Type	Description
total time	double	Total time in seconds over which the loop will evolve.
τ	double	Step size in time. This will also determine the number of steps taken.
heating function shape	int	Specifies which heating function will be used to drive evolution in the loop. EBTEL currently supports three options: (1) triangular pulse, (2) square pulse, or (3) Gaussian pulse.
loop length	int	Half-length of the loop in Mm
usage	int	This option specifies the type of output EBTEL produces. (1) TR and Coronal DEM calculated, (2) DEM calculation not included, (3) Non-thermal electron beam calculation included, and (4) TR and Coronal DEM calculated as well as flux and radiation ratios.
radiative loss function option	int	This option specifies whether to use the Rosner-Tucker-Vaiana(RTV) loss function (1) or the Raymond-Klimchuk(RK) loss function (0). Details regarding the RK loss function can be found in Klimchuk et al. (2008).
heat flux option	int	This option allows the user to choose either to use a classical (1) heat flux calculation or a dynamic (0) heat flux calculation.
DEM _{TR}	int	This option allows the user to specify how differential emission measure (DEM) of the transition region (TR) is calculated. Either thermal conduction parameters and the cooling phase can be used (old,1) or the modified energy equation is used to quadratically solve for the DEM in the TR (new,0).
solver	int	This option allows the user to choose whether to use (0) an Euler method, (1) a fourth-order Runge-Kutta routine, or (2) an adaptive Runge-Kutta routine to solve the modified hydrodynamic equations.
mode	int	This option allows the user to specify whether or not to force initial values (1) for T and n or compute them using an algorithm (0) built in to EBTEL. Additionally, the initial parameters can be computed using the scaling laws (2).
H_0	double	Maximum heating in units of $\text{erg cm}^{-3} \text{s}^{-1}$. This is the maximum value of the <i>ad-hoc</i> heating function.
$t_{H,1/2}$	double	Time between onset of heating to maximum value of the heating. Two times this value is the total width of the heating pulse in time (i.e. $t_H = 2(t_{H,1/2})$).
t_s	double	Time at which the heating function is turned on
DEM index	int	This index defines the range over which the DEM is calculated. A typical value is 451 which yields the range $10^4 \leq T_{\text{DEM}} \leq 10^8$. However, this number can be decreased to increase computation time.
err	double	Allowed truncation error. Used only when an adaptive method is used (i.e. <code>solver</code> = 2)
$T(t=0)$	double	Initial temperature. This parameter only used if mode=1.
$n(t=0)$	double	Initial density. This parameter only used if mode=1.

P. J. Cargill, S. J. Bradshaw, and J. A. Klimchuk, The Astrophysical Journal **758**, 5 (2012b), ISSN 0004-637X, URL <http://adsabs.harvard.edu/abs/2012ApJ...758....5C>.