LA-UR-16-29341
Approved for public release; distribution is unlimited.

Title: iVCJ: A tool for Interactive Visualization of high explosives CJ states

Author(s): Wooten, Hasani Omar
Aslam, Tariq Dennis
Whitley, Von Howard

Intended for: Report

Issued: 2016-12-12

iVCJ: A tool for Interactive Visualization of high explosives CJ states

By

H. Omar Wooten [1]
Tariq Aslam [2]
Von H. Whitley [1]

[1] X Theoretical Design-Primary Physics (XTD-PRI)
[2] Shock and Detonation Physics (M-9)

Abstract

A graphical user interface (GUI) tool has been developed that facilitates the visualization and analysis of the Chapman-Jouguet state for high explosives gaseous products using the Jones-Wilkins-Lee equation of state.

Introduction

A commonly used model for the expansion of gaseous products from high explosives detonation is the Jones-Wilkins-Lee (JWL) equation of state (EOS). This analytical EOS describes the pressure of gaseous products as a sum of exponential components, using coefficients and exponential parameters determined from cylinder test experiments.

The JWL EOS model has been implemented into Los Alamos hydrocodes such as FLAG and PAGOSA using a formulation in which, the parameter C is omitted, but the energy released during detonation is provided. For some high explosives analysis, it is useful to compare plots of the pressure isentrope in P-V space, which requires the C term. Furthermore, use of JWLs in hydrocodes requires that the parameters of choice be "self-consistent," meaning that the interrelationship between the JWL parameters and the Chapman-Jouguet state (e.g., detonation velocity, detonation compression and pressure, and particle speeds) be consistent with conservation laws for mass, energy, and momentum.

We have developed a GUI application for visualization and comparison of JWL isentropes and computation of the Chapman-Jouguet state. The application, iVCJ (**I**nteractive **V**isual **CJ** Solver) allows users to input JWL parameters, supplying either C or E_0. If E_0 is supplied, and C is absent, the application solves for C using the energy provided, thereby allowing the isentrope to be plotted. iVCJ also provides real-time plotting and CJ state calculation, allowing the user to instantly observe how changes in JWL parameters affect both the isentrope and the CJ state.

Methods and Materials

The analytical JWL EOS and the isentrope through the CJ point are shown in Eqs. 1 and 2 respectively,

$$P(V,E) = A\left(1 - \frac{\omega}{R_1 V}\right) + B\left(1 - \frac{\omega}{R_2 V}\right) + \frac{\omega E}{V} \quad [1]$$

$$P(V) = Ae^{-R_1 V} + Be^{-R_2 V} + CV^{-(1+\omega)}, \quad [2]$$

where A, B, C, R_1, and R_2, and ω are parameters that are typically fit to cylinder test data, and V represents the inverse of relative compression, V = v/v_0, where $v_0 = 1/\rho_0$ and $v = 1/\rho$.

The CJ state is defined to be the point where the Rayleigh line of the unreacted explosive Hugoniot is equal and tangent to the Hugoniot of the gaseous products.[1] The Rayleigh line is represented by Eq. 3,

$$P_{Rayleigh} = P_0 - \frac{\Delta P}{\Delta V}(V_0 - V), \quad [3]$$

which is equivalent to

$$P_{Rayleigh} = -\frac{dP}{dV}(V_0 - V), \quad [4]$$

under the commonly used assumption that $P_0 = 0$.

From Eq. 2,

$$\frac{dP}{dV} = -AR_1 e^{-R_1 V} + BR_2 e^{-R_2 V} - C(1+\omega)V^{-(2+\omega)}. \quad [5]$$

This requires the CJ state to occur at V_{CJ} for which,

$$P(V) - \frac{dp}{dV}(V_0 - V) = 0. \quad [6]$$

Several important relationships may be derived from the application of conservation laws of mass and momentum at the front of a shock and at the CJ state (i.e., V = V_{CJ}),

$$\frac{V_0}{V_{CJ}} = \frac{D_{CJ}}{D_{CJ} - u_{CJ}} \quad \rightarrow \quad V_{CJ} = V_0 \frac{D_{CJ} - u_{CJ}}{D_{CJ}}, \quad [7]$$

$$P_{CJ} = \frac{u_{CJ} D_{CJ}}{V_0} \quad \rightarrow \quad u_{CJ} = \frac{P_{CJ} V_0}{D_{CJ}}, \quad [8]$$

where P_0 is again assumed to be zero.

Temporarily ignoring the P_0 assumption and combining the mass and momentum equations, solving Eq. 7 for u_{CJ} gives,

$$\frac{V_0}{V_{CJ}} = \frac{D_{CJ}}{D_{CJ}-u_{CJ}} \quad \rightarrow \quad u_{CJ} = D_{CJ}\frac{V_0-V_{CJ}}{V_0}. \qquad [9]$$

Substitution into the momentum equation gives,

$$P_{CJ} - P_0 = \frac{D_{CJ}}{V_0}\left(D_{CJ}\frac{V_0-V_{CJ}}{V_0}\right) = \frac{D_{CJ}^2}{V_0^2}\left(V_0 - V_{CJ}\right), \qquad [10]$$

such that,

$$\frac{P_{CJ}-P_0}{V_{CJ}-V_0} = \frac{\Delta P}{\Delta \mathrm{V}} = \frac{dP}{dV} = -\frac{D_{CJ}^2}{V_0^2}, \qquad [11]$$

where we include the Rayleigh line equation as appropriate.

In hydrocodes, it is often useful to specify the chemical energy released during detonation as a specific parameter, E_0, in addition to A, B, R_1, and R_2, and ω, while omitting C, allowing it to be internally computed and hidden by the code. However, knowledge of C is instructive for visual inspection of the isentrope, sensitivity analyses, data fitting, etc.

E_0 is computed as described in more detail by Weseloh[2] and Wooten[3] as the integral of Eq. 2 from V_{CJ} to infinity, minus the portion of energy minus the energy lost to the mechanical compression of the explosive by the shock. This is represented as,

$$E_0 = -\int_{V_{CJ}}^{\infty} P(V)dV - \frac{1}{2}P_{CJ}\left(1 - V_{CJ}\right) \qquad [12]$$

which, when expanded, becomes

$$E_0 = \frac{A}{R_1}e^{-R_1 V} + \frac{B}{R_2}e^{-R_2 V} + \frac{C}{\omega}V^{-\omega} - \frac{1}{2}P_{CJ}\left(1 - V_{CJ}\right). \qquad [13]$$

With this, we have Eqs. 2, 11, and 13 for unknowns P_{CJ}, D_{CJ}, and C, and Eqs. 7 and 8 relating V_{CJ} and u_{CJ}.

iVCJ first solves Eq. 6 to determine the CJ state variables, then, if it is absent, solves Eqs. 2, 7, 8, 11, and 13 to determine C. It then has all of the necessary inputs required for plotting isentropes.

iVCJ was written in the scripting language Python Anaconda[4] (ver 2.7.11), and is therefore highly portable and requires neither building nor compiling. The Tkinter module was employed for GUI components.

The CJ solver algorithm employs the Nelder-Mead downhill simplex minimization method[5] (scipy.optimize.fmin) with a tolerance of 1×10^{-6}. The C-solver employs a root finding algorithm (scipy.optimize.fsolve) based on MINPACK[6], developed at Argonne National Laboratory. iVCJ reports CJ state variables are reported with a precision of 1×10^{-6}.

The performance of iVCJ's algorithms was verified by computing E and C JWL parameters and CJ state variables V_{CJ}, P_{CJ}, and D_{CJ} for four high explosives with a range of CJ properties (Composition-B, TNT, PBX 9501, and PBX 9404). iVCJ's results were compared to reference data from the LLNL High Explosives Reference Guide[7] for 1-inch diameter cylinder tests.

Results

Table I shows a comparison of iVCJ's calculations for JWL parameters E and C, and CJ state variables V_{CJ}, P_{CJ}, and D_{CJ} for Composition-B, TNT, PBX 9501 and PBX 9404. In all cases, only minor differences occur due to rounding and precision differences between the reference data and iVCJ's precision of 1×10^{-6}. Thus, iVCJ agrees well with reference data.

Table I. Comparison of iVCJ calculations relative to data from LLNL High Explosives Reference Guide (HE-RG) for Composition B, TNT, PBX 9501 and PBX 9404.

Composition B	HE-RG	iVCJ	Ratio
ρ (g/cm³)	1.694		
A (Mbar)	5.797578		
B (Mbar)	0.114037		
R1	4.5		
R2	1.5		
ω	0.28		
C (Mbar)	0.01913448	0.019134	0.99997
E_0 (Mbar)	0.11	0.11	1.00000
V_{CJ}	0.737318	0.737318	1.000000
P_{CJ} (Mbar)	0.27603	0.276029	0.999996
D_{CJ} (cm/μs)	0.7876	0.7876	1.000000

TNT	HE-RG	iVCJ	Ratio
ρ (g/cm³)	1.612		
A (Mbar)	4.213169		
B (Mbar)	0.0946951		
R1	4.5		
R2	1.5		
ω	0.28		
C (Mbar)	0.0128139	0.012814	1.00001
E_0 (Mbar)	0.078	0.078	1.00000
V_{CJ}	0.73559	0.735588	0.999997
P_{CJ} (Mbar)	0.20423	0.204225	0.999976
D_{CJ} (cm/μs)	0.6922	0.6922	1.000000

PBX 9501	HE-RG	iVCJ	Ratio
ρ (g/cm³)	1.834		
A (Mbar)	7.781315		
B (Mbar)	0.208618		
R1	4.5		
R2	1.5		
ω	0.28		
C (Mbar)	0.01499822	0.014998	0.99999
E_0 (Mbar)	0.118	0.118	1.00000
V_{CJ}	0.736	0.735702	0.999595
P_{CJ} (Mbar)	0.37537	0.375368	0.999995
D_{CJ} (cm/μs)	0.88	0.88	1.000000

PBX 9404	HE-RG	iVCJ	Ratio
ρ (g/cm³)	1.843		
A (Mbar)	7.624336		
B (Mbar)	0.224251		
R1	4.5		
R2	1.5		
ω	0.28		
C (Mbar)	0.0154693	0.015469	0.99998
E_0 (Mbar)	0.122	0.122	1.00000
V_{CJ}	0.733	0.732805	0.999734
P_{CJ} (Mbar)	0.37961	0.379614	1.000011
D_{CJ} (cm/μs)	0.878	0.878	1.000000

Features and operation of iVCJ

The main features of iVCJ are described below. Fig. 1 shows the initial window of iVCJ, with details listed in Table II. Output from the CJ solver is described in Table III.

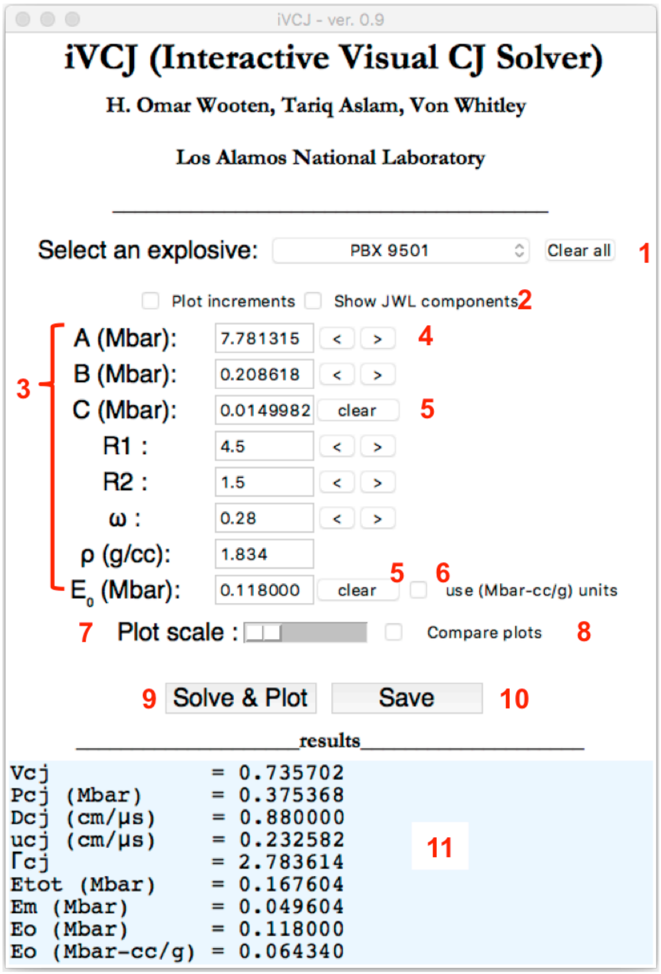

Figure 1. iVCJ main window. Labels are described in Table II.

To begin, the user may enter JWL parameter values manually, or select a pre-defined model from the pull-down menu, for which parameters will be automatically populated. Parameters may be adjusted as desired, either manually, or in increments of 0.1 with the modification < > buttons.

Parameters A, B, R1, R2, ω, and ρ are required. In addition, either E_0 or C must also be supplied by the user. If C is absent, it is computed according to the methods described in the previous section of this report. If both E_0 and C are entered, E_0 takes precedence, and iVCJ will re-compute C. The user may enter E_0 in units of either Mbar or Mbar-cm^3/g.

If the "Plot increments" checkbox is selected, the parameter modification buttons, in addition to changing the parameters, will update the isentrope plots in real-time.

Selecting the “Show JWL components” checkbox will display the components of the JWL isentrope (Eq. 2) in addition to the total isentrope.

Table II. Description of the iVCJ main window, shown in Fig.1.

	Description
1.	Pull-down menu of common HE models, and clear all plots and data
2.	User may select to have plots automatically updated in real time as parameters are incrementally changed with modification buttons (4), and choose to display JWL exponential components on the plot.
3.	JWL isentrope parameters are automatically filled when model is selected from pull-down, or entered manually
4.	Modification < > buttons may be used to change JWL parameters in increments of 0.1. If Plot Increment checkbox is selected, modification buttons will also update plots in real time
5.	Parameters C and E_0 may be cleared
6.	Default E_0 units is Mbar, but Mbar-cc/g may be used, if desired.
7.	The scale (maximum V/V_0) of the isentrope plots may be increased from 5 up to 100
8.	Compare plots as parameters are adjusted
9.	Compute CJ parameters and plot the isentrope. The <Return> key also calls the solver.
10.	Save CJ results to text file and add parameters to the HE model pull-down data file.
11.	CJ state results

The “Plot scale” slider bar changes the scale of the isentrope plots out to a maximum V/V_0 = 100.

If the “Compare plots” checkbox is selected, the current and previous isentrope plots will be displayed. As parameters are updated, the current and previous plots are also updated. In addition, the differences plots for the pressure isentrope and integrated energy are also displayed.

After modifying parameters, the CJ solver and plotter may be called by clicking the “Solve and Plot” button, or by pressing the <Return> key on the keyboard. This will produce CJ state variables to be displayed in the results area of the window, and the plots window to appear. The solver produces CJ state output data according to Table II. The plots include:

1. The pressure isentrope
2. Natural log of the pressure isentrope
3. The chemical energy released between V_{CJ} and each V/V_0 according to Eq. 12 (with V/V_0 replacing the upper limit).
4. If plot comparison is selected, differences between the current and previous isentropes and integrated energy are also displayed.

Figure 2 show a comparison of PBX 9502 (black) and a previous PBX 9501 (red-dashed) isentropes and energy integrals.

Results and parameters may be saved to a text file, as described in the following section, by clicking the "Save" button.

To clear all plots and data, click "Clear all." To exit iVCJ, press the <Esc> key on the keyboard.

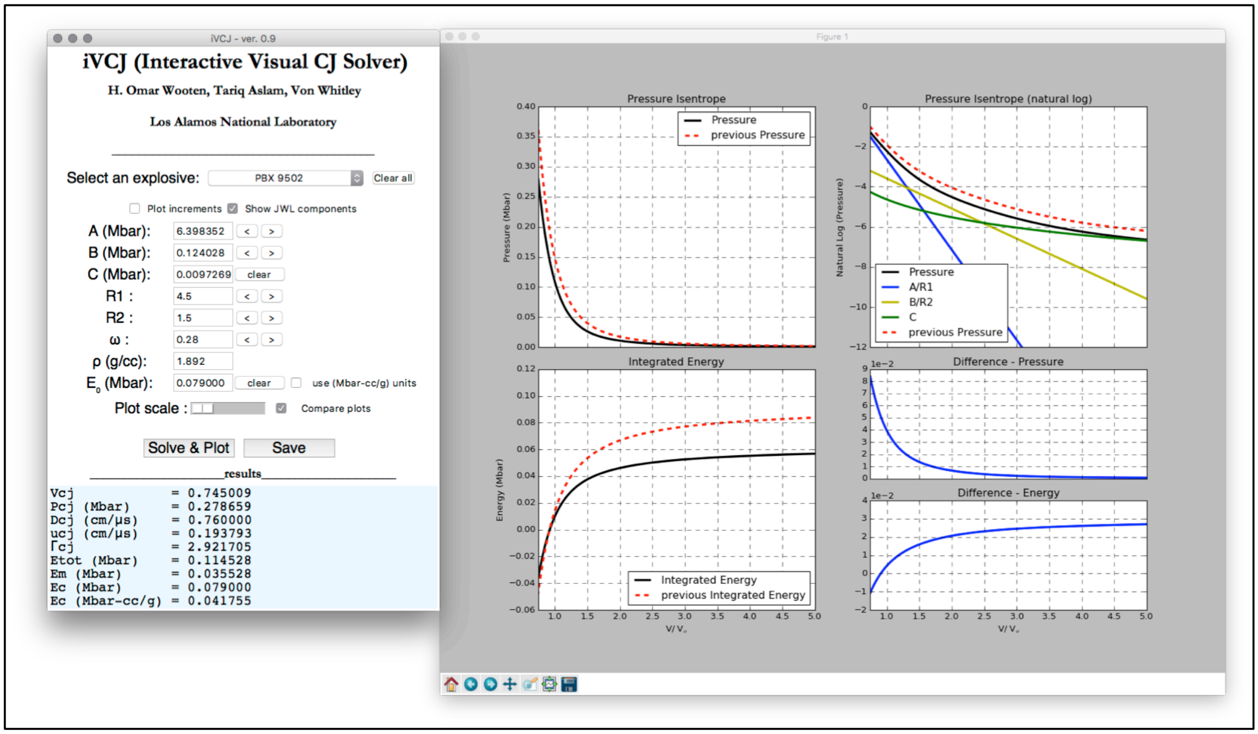

Figure 2. Pressure isentropes for PBX 9502 (black curve), PBX 9501 (red-dashed curve), and differences between the two (blue curves)

Table III. CJ state variables computed by iVCJ

Variable	Description
V_{CJ}	Compression at the CJ state, normalized to V_0 (dimensionless)
P_{CJ}	Pressure at the CJ state (Mbar)
D_{CJ}	Detonation velocity at the CJ state (cm/μs)
u_{CJ}	Particle speed at the detonation front (cm/μs)
Γ_{CJ}	Grüneisen gamma (dimensionless)
E_{tot}	Total energy (area under the isentrope curve) between V_{CJ} and ∞
E_m	The mechanical energy done by the shock to compress the explosive to the CJ state
E_0	$E_{tot} - E_m$, the chemical energy released by the explosive

iVCJ data files

iVCJ comprises a single Python script, `ivcj.py`. Within this script is a function that creates a file, `jwl_in.txt`, if it is not already present within the same directory, that includes the data used to populate the pull-down menu of pre-defined high explosives.

After a calculation, the user may save the parameters and CJ state results to a file `jwl_out.txt`, by clicking the "Save" button. Doing so will also add the parameters as an additional record in `jwl_in.txt`. This allows the user to recall this simulation from the pull-down menu the next time iVCJ is run.

Users may add records to `jwl_in.txt` manually. Figure 3 shows the format of `jwl_in.txt` for which a record has been added for PBX 9404 that uses E_0, and for which C is blank.

```
TITLE           A           B           C               E       R1      R2      omega   density(g/cc)
==========================================================================================================
PBX 9404        7.624336    0.224251    0.01546925      -       4.5     1.5     0.28    1.843
PBX 9501        7.781315    0.208618    0.01499822      -       4.5     1.5     0.28    1.834
PBX 9502        6.398352    0.124028    0.00972697      -       4.5     1.5     0.28    1.892
PBX 9404E       7.624336    0.224251    -               0.0662  4.500   1.500   0.280   1.843
```

Figure 3. Example of `jwl_in.txt` manually edited with a record of PBX 9404.

```
User_1 JWL parameters
A    = 7.624336
B    = 0.224251
C    = 0.015469
R1   = 4.500000
R2   = 1.500000
w    = 0.280000
Vcj           = 0.732805
Pcj  (Mbar)   = 0.379614
Dcj  (cm/us)  = 0.878000
ucj  (cm/us)  = 0.234597
Gcj           = 2.742588
Etot  (Mbar)  = 0.172715
Emech  (Mbar)= 0.050715
Echem  (Mbar)= 0.122000
```

Figure 4. Example of `jwl_out.txt`.

Conclusions

A user-friendly GUI application has been developed in Python for analyzing JWL isentropes. iVCJ allows users to input parameters from other sources, or from hydro code simulations, and solves the nonlinear equations necessary to compute all parameters required for plotting P(V), and computes the CJ state variables based on the parameters supplied by the user. iVCJ provides real-time feedback to the user, re-computing CJ state variables and updating plots as parameters are adjusted. iVCJ agrees well with reference data, with differences arising due to reported precisions and rounding.

References

1. Cooper, P., "Explosives Engineering," Wiley-VCH, New York, 1996. Page 255.

2. Weseloh, W., "JWL in a Nutshell," LA-UR-14-24318, Los Alamos National Laboratory, 2014.

3. Wooten, H., "Analysis of historical and recent PBX 9404 cylinder tests using FLAG," LA-UR-16-24389, Los Alamos National Laboratory, 2016.

4. www.python.org

5. Nelder, John A., Mead, R., "A simplex method for function minimization". Computer Journal. **7**: 308–313, 1965.

6. J. J. Moré, B. S. Garbow, and K. E. Hillstrom, User Guide for MINPACK-1, Argonne National Laboratory Report ANL-80-74, Argonne, Ill., 1980

7. https://hereference.llnl.gov