

Supporting Information for “VESIcal Part I: An open-source thermodynamic model engine for mixed volatile (H₂O-CO₂) solubility in silicate melts”

K. Iacovino¹, S. Matthews^{2,3}, P.E. Wieser⁴, G.M. Moore¹, and F. Bégué⁵

¹Jacobs, NASA Johnson Space Center, Houston, TX 77058, USA

²Johns Hopkins University, Department of Earth and Planetary Sciences, Baltimore, MD 21218, USA

³University of Iceland, Institute of Earth Sciences, Askja, Sturlugata 7, 101 Reykjavik, Iceland

⁴University of Cambridge, Department of Earth Sciences, Downing Street, Cambridge CB2 3EQ, UK

⁵University of Geneva, Department of Earth Sciences, Geneva, Switzerland

Contents of this file

1. Text S1 to S3
2. Figures S1 to S21

Additional Supporting Information (Files uploaded separately)

1. Captions for Datasets S1 to S7
2. Captions for Jupyter Notebooks S1 to S10

Introduction This supplement contains text, figures, tables, and jupyter notebooks that serve to both enhance the user experience in VESIcal (e.g., with additional worked ex-

Corresponding author: Kayla Iacovino, Jacobs, NASA Johnson Space Center, Houston, TX 77058, USA

amples of how the code can be used) and give a complete and thorough background as to the thermodynamic calculations and methodologies employed both in VESICAL and in the models contained therein. Captions for the datasets and jupyter notebooks (all of which are uploaded as separate files) are given below.

Captions for Datasets S1 to S7

Seven Microsoft Excel files containing compositional data on various silicate melts are included. These files are as follows:

1. example_data.xlsx: Silicate melts from basalts to rhyolites. These data are used as the example data in the main manuscript.
2. alkaline.xlsx: Alkaline experimental samples from Iacovino et al. (2016), ranging from basanite to phonotephrite
3. basalts.xlsx: Basalts from Tucker et al. (2019) and Bennett et al. (2019)
4. cerro_negro.xlsx: Basaltic melt inclusions from Cerro Negro volcano from Roggen-sack (2001)
5. rhyolites.xlsx: Rhyolites from Myers et al. (2019) and Mercer et al. (2015)
6. liu_2005.xlsx: Calibration dataset of Liu et al. (2005); bulk composition data only
7. moore_1998.xlsx: Calibration dataset of Moore et al. (1998)

Captions for Jupyter Notebooks S1 to S10

Ten Jupyter Notebooks are included with this manuscript. These notebooks are given along with all necessary files to run them plus a PDF copy of the evaluated notebooks such that the code therein can be accessed without needing to run the notebook itself.

Each notebook and accessory files are located within a folder following the naming scheme described here:

1. Allison: For testing VESIcal's reproduction of the Allison et al. (2019) model
 - S1_Testing_Allison_et_al_2019.ipynb
 - S1_Testing_Allison_et_al_2019.pdf
 - S1_Testing_Allison_et_al_2019.xlsx
2. Dixon: For testing VESIcal's reproduction of the Dixon (1997) model (VolatileCalc)

- S2_Testing_Dixon_1997_VolatileCalc.ipynb
- S2_Testing_Dixon_1997_VolatileCalc.pdf
- S2_Testing_Dixon_1997_VolatileCalc.xlsx
- VolatileCalc_Test1.png
- VolatileCalc_Test2.png

3. Iacono-Marziano: For testing VESIcal's reproduction of the Iacono-Marziano (2012) model

- S3_Testing_Iacono-Marziano_et_al_2012.ipynb
- S3_Testing_Iacono-Marziano_et_al_2012.pdf
- S3_Testing_Iacono-Marziano_et_al_2012.xlsx

4. Liu: For testing VESIcal's reproduction of the Liu et al. (2005) model

- S4_Testing_Liu_et_al_2005.ipynb
- S4_Testing_Liu_et_al_2005.pdf
- S4_Testing_Liu_et_al_2005.xlsx

- Testing_Liu_img1.png
- Testing_Liu_img2.png
- Testing_Liu_img3.png

5. MagmaSat: For testing VESICAL's reproduction of MagmaSat (Ghiorso and Gualda, 2015) model

- S5_Testing_Magmasat.ipynb
- S5_Testing_Magmasat.pdf
- S5_Testing_Magmasat.xlsx
- Testing_MagmaSat_img1.png

6. Moore: For testing VESICAL's reproduction of the Moore et al. (1998) model

- S6_Testing_Moore_et_al_1998.ipynb
- S6_Testing_Moore_et_al_1998.pdf
- S6_Testing_Moore_et_al_1998.xlsx

7. Shishkina: For testing VESICAL's reproduction of the Shishkina et al. (2014) model

- S7_Testing_Shishkina_et_al_2014.ipynb
- S7_Testing_Shishkina_et_al_2014.pdf
- S7_Testing_Shishkina_et_al_2014.xlsx
- Testing_Shishkina_img1.png

8. CerroNegro-Calculations: For running calculations discussed in Section 4.1 of the manuscript

- S8_CerroNegro_isobar_comparison.ipynb

- S8_CerroNegro_isobar_comparison.pdf
- cerro_negro_satP_compare.xlsx

9. Isobar_Comparison_Plotter: For running calculations discussed in Section 4.2 of the manuscript

- S9_Isobar_Comparison_Plotter.ipynb
- S9_Isobar_Comparison_Plotter.pdf

10. Model_Hybridization: Notebook illustrating how to hybridize models, as discussed in Section 3.9 of the manuscript

- S10_Model_Hybridization.ipynb
- S10_Model_Hybridization.pdf

Text S1. VESICAL's verbose installation instructions

Below are instructions for installing the necessary prerequisite libraries and VESICAL on your computer in order to run the VESICAL code locally.

Step 1: Install dependencies. Dependencies are prerequisite files or libraries that you must have installed on your computer for VESICAL to run properly. VESICAL's dependencies are:

1. Python3 (<https://www.python.org/downloads/>)
 - (i) Python pandas (<https://pandas.pydata.org/>)
 - (ii) Numpy (<https://numpy.org/>)
 - (iii) SciPy (<https://www.scipy.org/>)
 - (iv) Matplotlib (<https://matplotlib.org/>)
2. ENKI thermoengine (<https://gitlab.com/ENKI-portal/ThermoEngine>)

Install Python3. First, determine if Python3 (or some other version of Python) is installed on your computer. On Mac or Linux systems, open the application called Terminal and type the following, which will start running python if you have it installed: `python`

If python is installed, you will see a message telling you which python is installed. If this message says “Python 3.x.x” where x can be any number, then you have Python3 already installed. If the message also says “Anaconda, Inc.” then that’s even better – you have Anaconda’s version of Python3, which means you have all of the basic dependencies already. To quit python, type: `exit()`

If python is not installed, or if you see a different python version, we suggest installing Python3 via Anaconda3 (<https://www.anaconda.com/products/individual>). Anaconda is Python plus all of the most commonly used libraries, many of which are required to run VESICAL. Pandas, Numpy, SciPy, and Matplotlib all come prepackaged in Anaconda.

Install ENKI thermoengine. Next, ENKI’s thermoengine (the python implementation of MELTS) must be installed. Because installation instructions for thermoengine may change, we will not replicate them here. Instead we refer you to the ENKI portal at <https://gitlab.com/ENKI-portal/ThermoEngine>. From there, download the thermoengine files and follow the installation instructions.

Step 2: Install VESICAL. Now that all dependencies are installed, you can install VESICAL in your terminal by typing: `pip install VESICAL`

Step 3: Use VESICAL. You are now ready to begin coding with VESICAL. At the beginning of your python script, type “`import VESICAL as v`” to use VESICAL in the same way shown in this manuscript.

Text S2. Text S2. Generic methods for calculating mixed-fluid properties

VESICAL provides a set of methods for calculating the properties of mixed CO₂-H₂O fluids, which can be used with any combination of H₂O and CO₂ solubility model. Using generic methods allows additional models to be added to VESICAL by defining only the (simpler) expressions describing pure fluid solubility. Non-ideality of mixing in the fluid or magma phases can be incorporated by specifying activity and fugacity models. We provide a summary here of how we approach calculating the properties of mixed-fluid systems.

S2.1 Finding the saturation pressures of pure fluids

Whilst the variables required for each pure-fluid solubility model are different, we take a very similar approach to calculating fluid properties for each. Most pure-fluid models provide analytical expressions for the concentration of a volatile species dissolved in a magma as a function of either its partial-pressure or fugacity. However, in most cases, an analytical expression does not exist for the inverse calculation (calculating the saturation pressure given the concentration of dissolved volatile species), it must be solved numerically.

Numerical methods obtain the answer by making guesses for the saturation pressure, checking whether the predicted concentration of the dissolved volatile species matches the measured value, and refining the guesses as appropriate. When the saturation pressure for a dissolved volatile concentration $[V]^L$ is found, this equality holds:

$$[V]_{measured}^L - [V]_{predicted}^L(P, \dots) = 0 \quad (1)$$

An advantage of using python is the ease with which other libraries can be incorporated, for example the scipy library (Virtanen et al., 2020), which contains optimized algorithms for finding solutions to problems like this. To solve for pure-fluid saturation pressures we employ the `scipy.optimize.root_scalar()` method.

S2.2 Finding saturation pressure of mixed fluids Magmas in equilibrium with mixed-fluids can be modelled using a combination of pure-fluid CO₂ and H₂O models, provided the mole-fraction of each species ($X_{\text{CO}_2}^F$ and $X_{\text{H}_2\text{O}}^F$) in the vapour phase is specified. In an ideal gas the partial pressure of the vapour phase is given by:

$$P_{\text{CO}_2} = X_{\text{CO}_2}^F \times P \quad (2)$$

For non-ideal fluids, the H₂O and CO₂ fugacities are a function of $X_{\text{CO}_2}^F$ and $X_{\text{H}_2\text{O}}^F$, but will tend to zero as $X_{\text{CO}_2}^F$ and $X_{\text{H}_2\text{O}}^F$ tend to zero.

When calculating the saturation pressure of a mixed CO₂-H₂O fluid, we have an additional unknown: $X_{\text{H}_2\text{O}}^F$ ($X_{\text{CO}_2}^F$ is given by $[1 - X_{\text{H}_2\text{O}}^F]$). Like for the pure-fluid case described above, the problem must be solved numerically, but for two parameters simultaneously. The solution can be expressed as the vector equality:

$$\begin{pmatrix} [H_2O]_{\text{measured}}^L \\ [CO_2]_{\text{measured}}^L \end{pmatrix} - \begin{pmatrix} [H_2O]_{\text{predicted}}^L(P, X_{H_2O}, \dots) \\ [CO_2]_{\text{predicted}}^L(P, X_{H_2O}, \dots) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (3)$$

VESICAL then employs the `scipy.optimize.root()` method to obtain the value of the vector $\begin{pmatrix} P \\ X_{H_2O} \end{pmatrix}$ for which the above equality holds.

S2.3 Finding the equilibrium fluid composition

When calculating the dissolved volatile concentrations in a progressively degassing magma, we break the calculation into steps, each with a small decrement in pressure. As pressure decreases, the mass of fluid in equilibrium with the magma increases, and its com-

position will change. It follows that at each stage we must solve for the mass and composition of the fluid in equilibrium with the magma, given the total mass of volatiles in the system. This calculation is implemented in `MixedFluid.calculate_equilibrium_fluid_comp()`, which in turn is called by `MixedFluid.calculate_degassing_path()` for each increment of degassing magma. Here we summarize the methodology for the calculation.

Mass balance in the system can be expressed in terms of the mole fractions of H_2O ($X_{\text{H}_2\text{O}}$) and CO_2 (X_{CO_2}) in the silicate-liquid (L), the H_2O - CO_2 fluid (F), and the system total (T), and the fraction of fluid present (f).

$$X_{\text{CO}_2}^F + X_{\text{H}_2\text{O}}^F = 1 \quad (4)$$

$$(1 - f)X_{\text{CO}_2}^L + fX_{\text{CO}_2}^F = X_{\text{CO}_2}^T \quad (5)$$

$$(1 - f)X_{\text{H}_2\text{O}}^L + fX_{\text{H}_2\text{O}}^F = X_{\text{H}_2\text{O}}^T \quad (6)$$

The fraction of fluid (f) can be eliminated by substitution, obtaining Eqn (8) of Papale et al. (1999):

$$\frac{X_{\text{CO}_2}^T - X_{\text{CO}_2}^L}{X_{\text{CO}_2}^F - X_{\text{CO}_2}^L} - \frac{X_{\text{H}_2\text{O}}^T - X_{\text{H}_2\text{O}}^L}{X_{\text{H}_2\text{O}}^F - X_{\text{H}_2\text{O}}^L} = 0 \quad (7)$$

In principle, this equation allows us to numerically solve for mass balance; however, the equation is valid only for both $X_{\text{CO}_2}^F \neq X_{\text{CO}_2}^L$ and $X_{\text{H}_2\text{O}}^F \neq X_{\text{H}_2\text{O}}^L$. When $X_{\text{CO}_2}^F \rightarrow X_{\text{CO}_2}^L$ or $X_{\text{H}_2\text{O}}^F \rightarrow X_{\text{H}_2\text{O}}^L$ the equation tends to $\pm\infty$, which the `scipy.optimize.root_scalar()` routine will mistake for a root of the equation. The regions over which the expression is not infinite can be found; however we find a more reliable approach when using the `scipy.optimize` library is to use the equation in a different form.

Rearranging the CO₂ mass-balance expression allows the fraction of fluid (f) to be calculated:

$$f = \frac{X_{CO_2}^T - X_{CO_2}^L}{X_{CO_2}^F - X_{CO_2}^L} \quad (8)$$

This is valid for $X_{CO_2}^F \neq X_{CO_2}^L$. It is much more unlikely that a solution will exist near $X_{CO_2}^F = X_{CO_2}^L$ than near $X_{H_2O}^F = X_{H_2O}^L$. By not directly eliminating f by dividing through by both $(X_{CO_2}^F - X_{CO_2}^L)$ and $(X_{H_2O}^F - X_{H_2O}^L)$ we significantly increase the reliability of using this expression with `scipy.optimize.root_scalar()`.

In this expression, the only known value is $X_{CO_2}^T$. However, if a guess is provided for $X_{CO_2}^F$, the partial pressure of CO₂ can be calculated and the corresponding value for $X_{CO_2}^L$ obtained. Since $X_{H_2O}^F = 1 - X_{CO_2}^F$, when we provide a guess for $X_{CO_2}^F$ we also fix $X_{H_2O}^F$. This in turn fixes the partial pressure of H₂O, allowing us to calculate $X_{H_2O}^L$ consistent with that guess. We can then test whether our guess for $X_{CO_2}^F$ satisfies mass balance by checking for the equality:

$$(1 - f)X_{H_2O}^L + fX_{H_2O}^F - X_{H_2O}^T = 0 \quad (9)$$

The value of $X_{CO_2}^F$ that uniquely satisfies mass balance can then be found numerically by solving this equation using `scipy.optimize.root_scalar()`.

Text S3 - Code Validation and Testing

A series of tests have been performed to ensure that models included in VESICAL are implemented correctly. For each model, a Jupyter notebook is provided which runs through these tests, named "Testing_Model.pynb", which reads from an excel spreadsheet named "Testing_Model.xlsx". We detail these tests below:

Moore et al., 1998

- Test 1 compares the experimental pressures for the H₂O-only experiments in the calibration dataset of Moore et al. (1998) with the saturation pressures obtained from VESICAL using the "MooreWater" model (Fig. 1a).
- Test 2 compares the solubility of H₂O in the melt calculated using the excel spreadsheet of Moore et al. (1998) to the outputs of VESICAL for a synthetic array of inputs. The outputs match to within $\pm 0.5\%$ (Fig. 2).

Dixon (1997), as implemented in VolatileCalc (Newman and Lowenstern, 2002)

- Test 1 compares saturation pressures from VESICAL to the VolatileCalc spreadsheet, as well as an Excel Macro which has been used by a number of studies (e.g., Tucker et al., 2019; Fig. 4).
- Test 2 compares X_{H_2O} in the fluid phase at volatile saturation to that predicted by VolatileCalc and the Excel Macro (Fig. 6).
- Test 3 compares isobars to those produced by VolatileCalc for SiO₂=49 wt% (Fig. 7).

Liu et al., 2005

- Test 1 compares the experimental pressures for the H₂O-only experiments in the calibration dataset of Liu et al. (2005) with the saturation pressures obtained from VESIcal using the "LiuWater" model (Fig. 8).
- Test 2 recreates the model lines shown on Fig. 5 of Liu et al. (2005) by applying `calculate_saturation_pressure` to a synthetic array of inputs with variable H₂O contents (at three different temperatures, Fig. 9).
- Test 3 recreates the model lines shown on Fig. 7 of Liu et al. (2005) by applying `calculate_saturation_pressure` to a synthetic array of inputs with variable CO₂ contents (at 1050^{circ}C; Fig. 11).
- Test 4 recreates the isobars shown in Fig. 6b of Liu et al. (2005) using `calculate_isobars_and_isopleths`.

Iacono-Marziano et al., 2012

- Test 1 compares the experimental pressures of the H₂O-only experiments in their calibration dataset to the saturation pressures obtained from VESIcal using the "Iacono-MarzianoWater" model (Fig. 12a).
- Test 2 compares the experimental pressures of the H₂O-CO₂ experiments in their calibration dataset to the saturation pressures obtained from VESIcal for the "Iacono-Marziano" model (Fig. 12b).
- A lot of the scatter in the regression lines shown in Test 1 and Test 2 is due to experimental noise. Test 3 compares saturation pressures obtained from the web calculator hosted at <http://calcul-isto.cnrs-orleans.fr/> to VESIcal outputs for compositions in the calibration dataset (Fig. 13a).

- Test 4 compares saturation pressures obtained from the web calculator to VESICAL outputs for a synthetic dataset (Fig. 13b).

- Test 5 compares dissolved volatile contents calculated from the web app to those from VESICAL (Fig. 14).

Shishkina et al., 2014

- Test 1 compares the experimental pressures of the H₂O-only experiments in the calibration dataset of Shishkina et al. (2014) to the saturation pressures obtained from VESICAL using the "ShishkinaWater" model (Fig. 15a).

- Test 2 compares the experimental pressures of the CO₂-only experiments in the calibration dataset to the saturation pressures obtained from VESICAL using the "ShishkinaCarbon" model (Fig. 15b).

- Test 3 compares the experimental pressures for mixed H₂O-CO₂ experiments presented in their to the saturation pressures obtained from VESICAL using the "Shishkina" model (Fig. 16).

MagmaSat (Ghiorso and Gualda, 2015)

- Test 1 compares saturation pressures from Bennett et al. (2019), who used the Mac App, to those calculated using VESICAL (Fig. 17).

- Test 2 compares the isobars shown in Fig. 14 of Ghiorso and Gualda (2015) to those calculated with VESICAL. We note that although the figure caption says that the composition of the Late Bishop Tuff was used, VESICAL matches their isobars exactly if the composition of the Early Bishop Tuff is used (Fig. 18).

- Test 3 compares X_{H_2O} calculated using the 'Fluid+magma from bulk composition' option of the web app with the `calculate_equilibrium_fluid_comp` function of VESIcal for a set of synthetic inputs (Fig. 19).

Allison et al. (2019)

- Test 1 compares saturation pressures calculated in the spreadsheet provided by Allison et al. (2019) for the Sunset Crater model, using a synthetic array of inputs with variable CO_2 contents ($H_2O=0$ wt%; Fig. 20).

- Test 2 compares saturation pressures from the spreadsheet of Allison et al. (2019) to those calculated by VESIcal for all 6 composition for 100, 5000 and 10,000 ppm CO_2 (and $H_2O=0$ wt%; Fig. 21).

 Moore et al., 1998 - Test 1

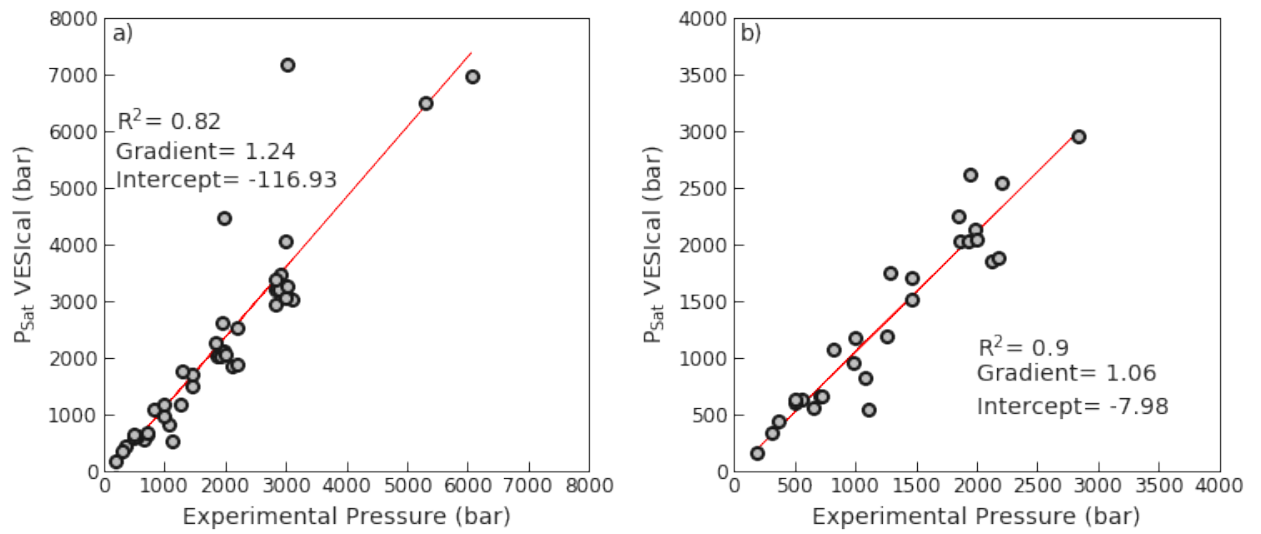


Figure 1. a) Comparison of the experimental pressures of the H₂O-only experiments in the calibration dataset of Moore et al. (1998) with saturation pressures calculated by VESICAL using 'MooreWater'. b) The correspondence is improved if experimental and saturation pressures lying outside the calibration range of 0–3000 bars are removed.

 Moore et al., 1998 - Test 2

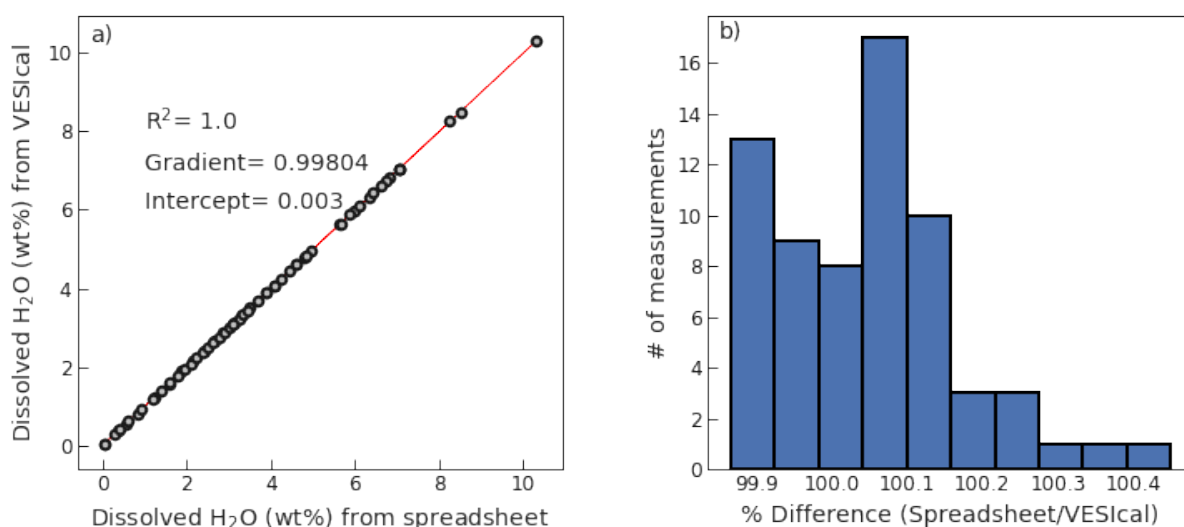


Figure 2. a) Comparison of the wt% H₂O dissolved in the melt calculated using the excel spreadsheet of Moore et al. (1998) with that calculated in VESICAL. The very small discrepancies vary as a function of pressure (see Fig. 3.)

 Moore et al., 1998 - Test 3

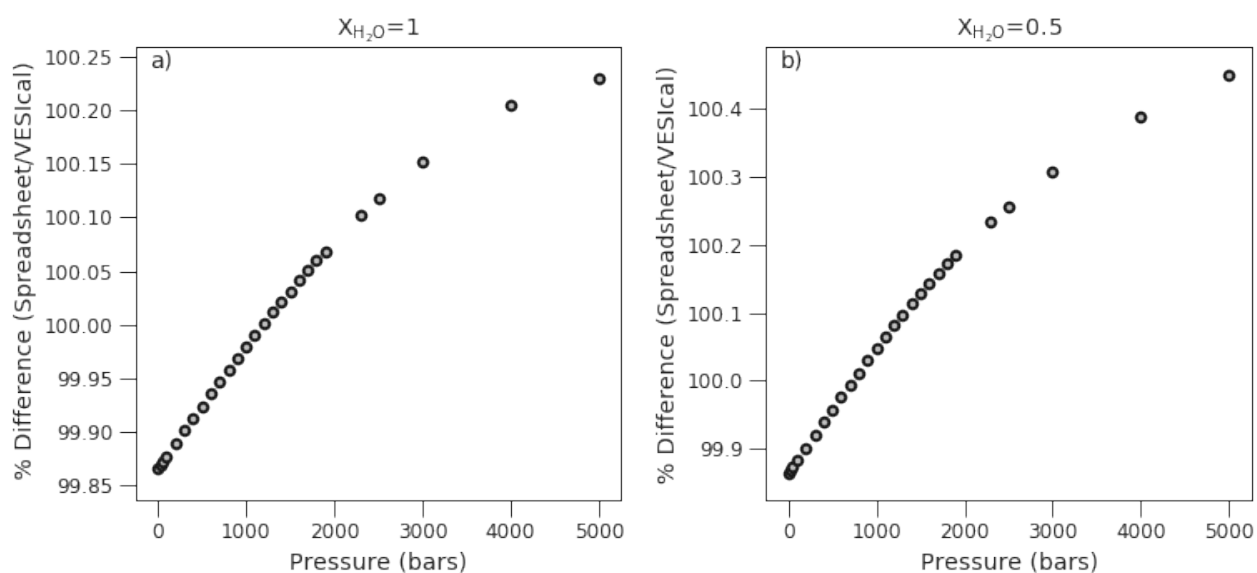


Figure 3. The discrepancy between the wt% H₂O in the melt calculated using the Moore spreadsheet and VESICAL varies with pressure, but is <0.15% for pressures within the calibration range (0-3000 bars).

VolatileCalc implementation of Dixon (1997) - Test 1

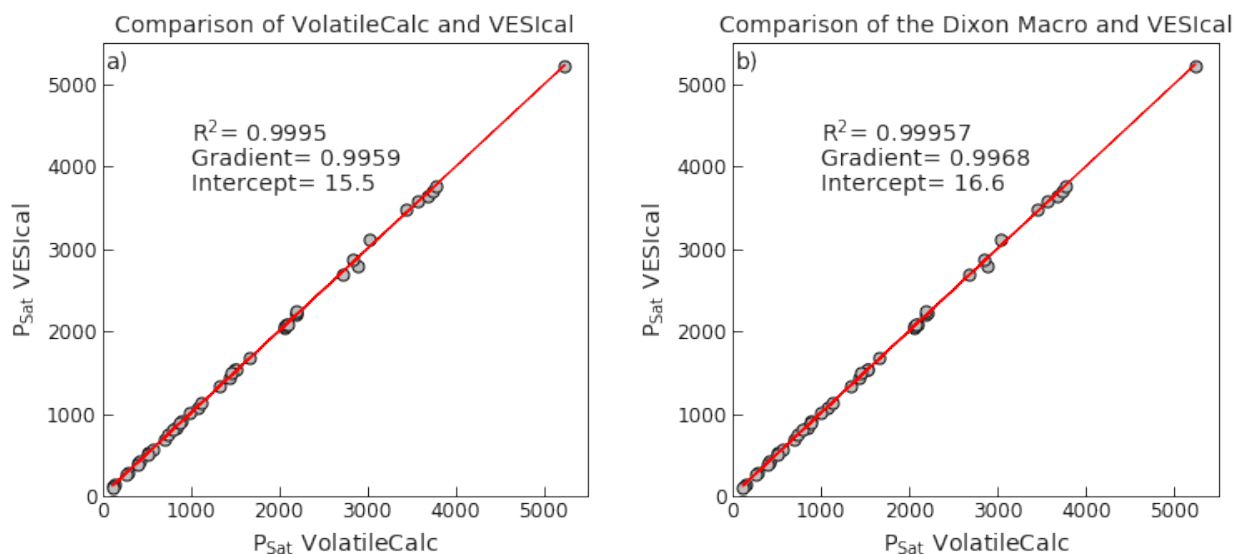


Figure 4. Comparison of the saturation pressures calculated using the VolatileCalc excel spreadsheet to those calculated in VESICAL, and a widely-used excel macro (e.g., Tucker et al., 2019).

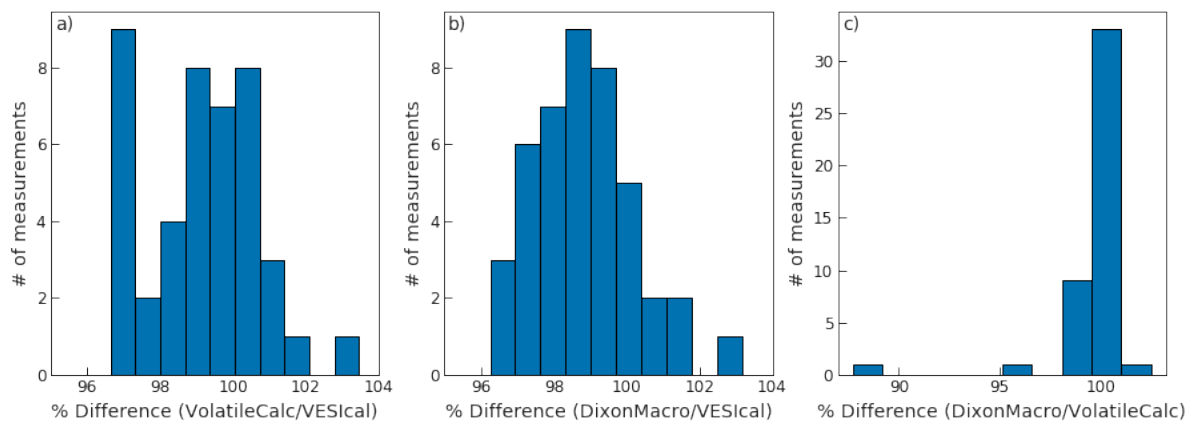


Figure 5. The percentage difference between saturation pressures from VESICAL vs. VolatileCalc (a) and Vesical vs. the excel macro (b) are similar in magnitude to those from VolatileCalc vs. the excel macro (c).

 VolatileCalc implementation of Dixon (1997) - Test 2

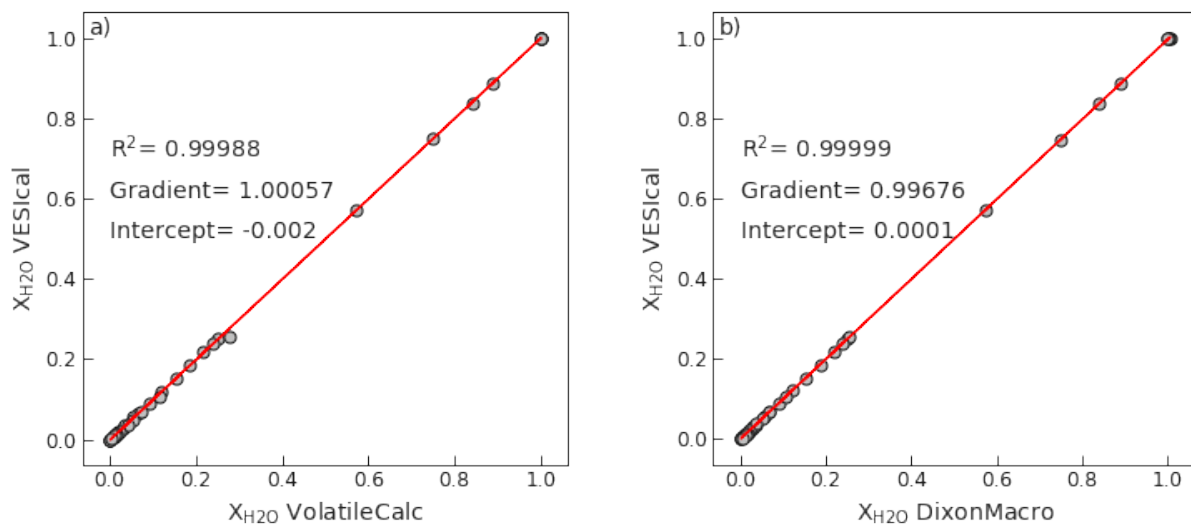


Figure 6. Comparison of X_{H_2O} in the fluid phase at volatile saturation calculated using the VolatileCalc excel spreadsheet to those calculated in VESICAL, and a widely-used excel macro (e.g., Tucker et al., 2019).

VolatileCalc implementation of Dixon (1997) - Test 3

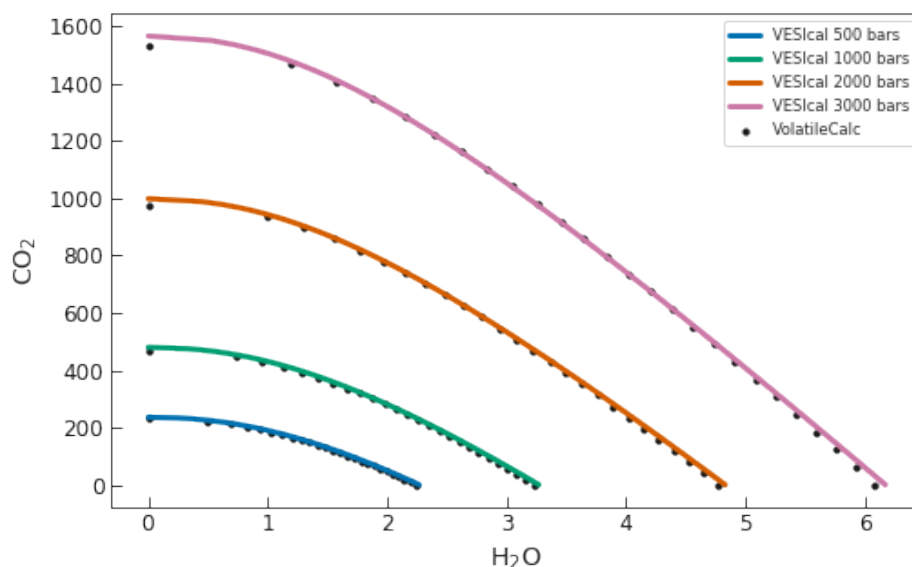


Figure 7. Comparison of isobars generated in VolatileCalc to those in VESICAL. The match is very good, with slight discrepancies at both end of the isobars (e.g., pure H₂O, or CO₂ fluids).

Liu et al., 2005 - Test 1

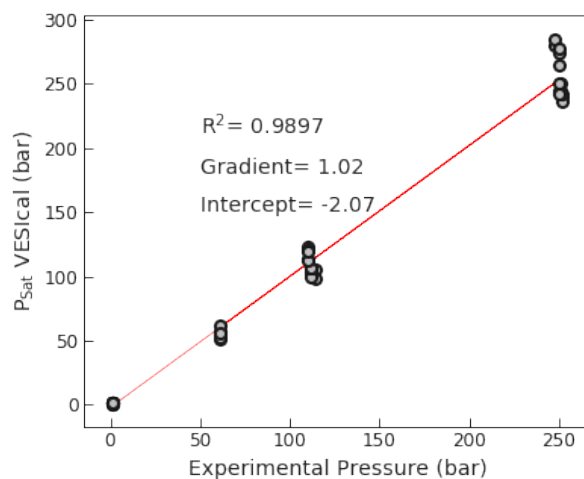


Figure 8. Comparison of the experimental pressures in the H₂O-only experiments in the calibration dataset of Liu et al. (2005) with saturation pressures calculated in VESICAL.

Liu et al., 2005 - Test 2

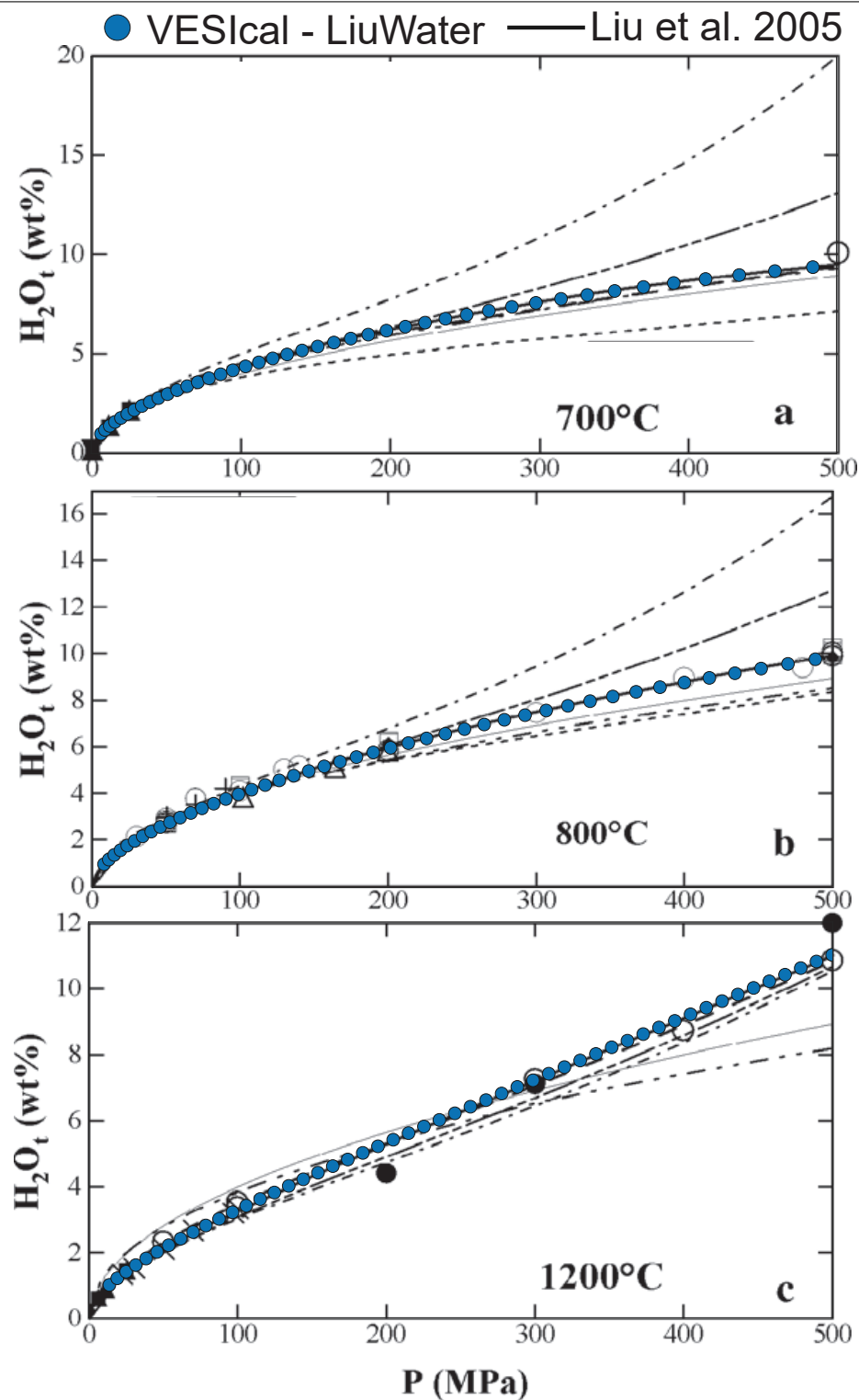


Figure 9. Relationship between pressure and dissolved H_2O content calculated in VESlcal for three different temperatures overlain on Fig. 5 of Liu et al. (2005).

Liu et al., 2005 - Test 3

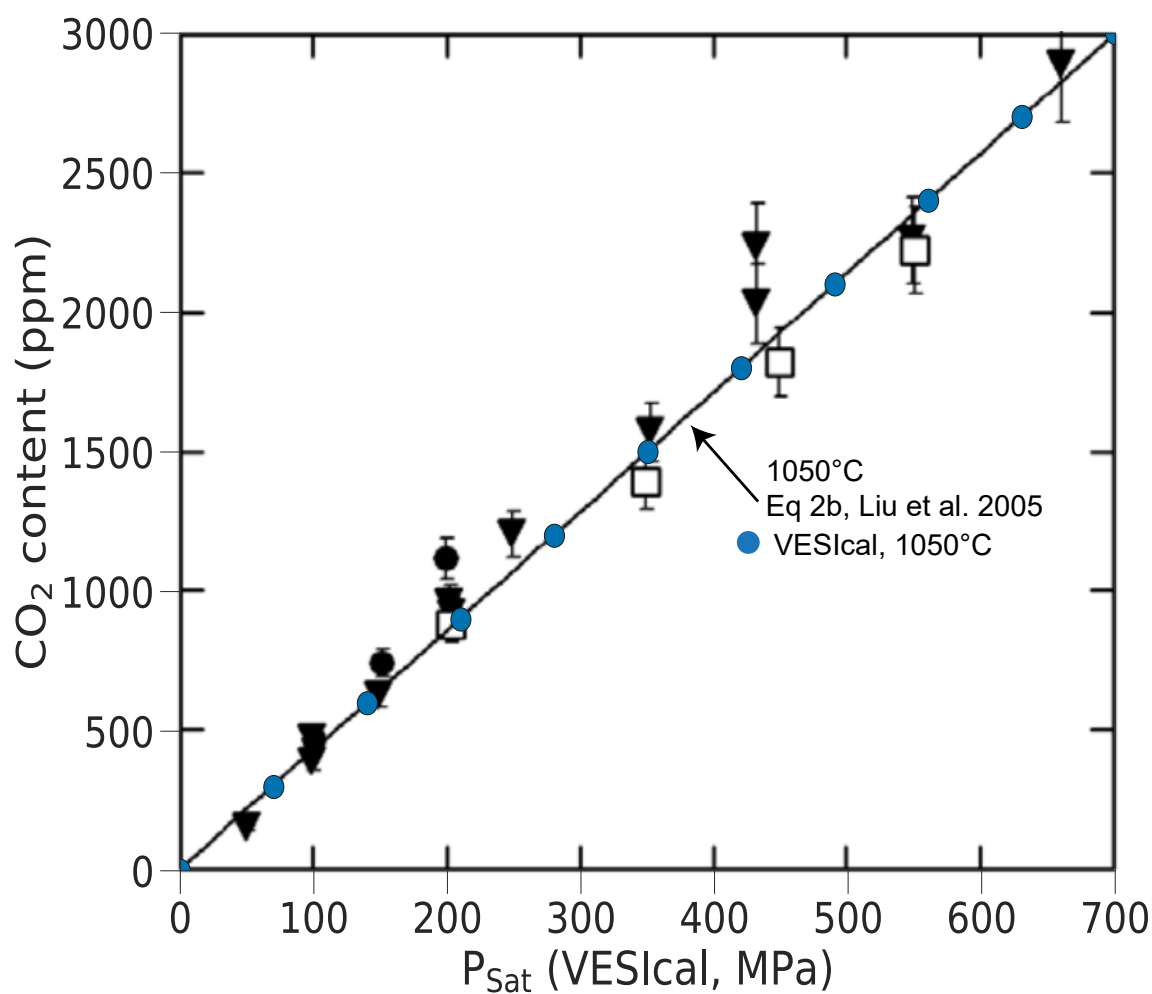


Figure 10. Relationship between pressure and dissolved CO_2 content calculated using VESIcal overlain on Fig. 7 of Liu et al. (2005)

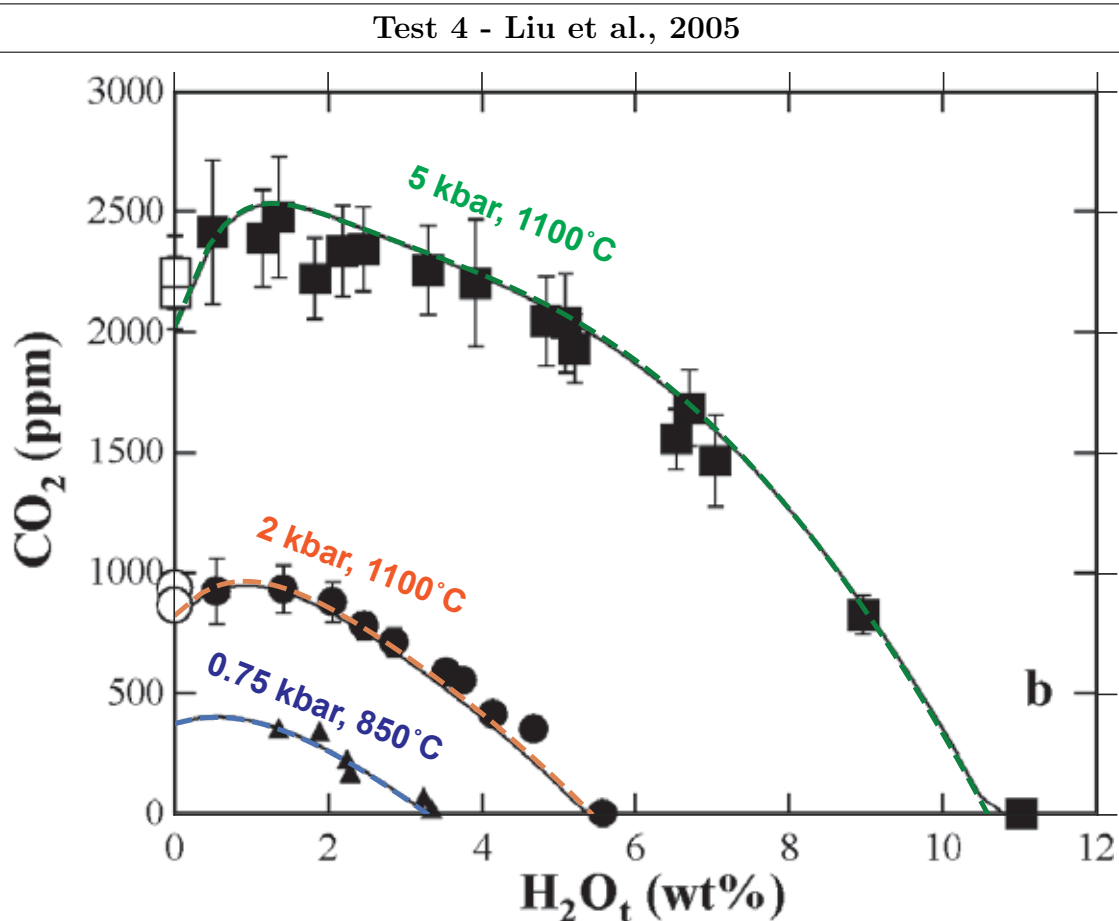


Figure 11. Isobars calculated using VESICAL overlain on Fig. 6b of Liu et al. (2005). For the 5 kbar isobar, H₂O and CO₂ contents are multiplied by 0.98 following their description in the figure caption.

Iacono-Marziano et al., 2012 - Test 1 and 2

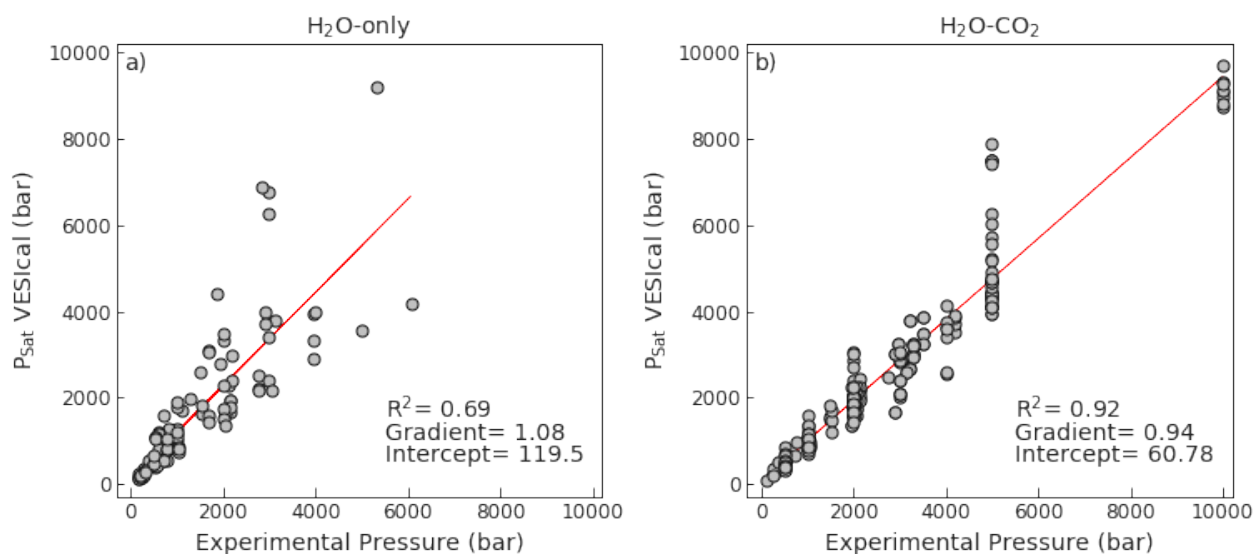


Figure 12. Comparison of the experimental pressures in H₂O-only experiments (a) and the H₂O-CO₂ experiments (b) in the calibration dataset of Iacono-Marziano with the saturation pressures calculated in VESICAL.

Iacono-Marziano et al., 2012 - Test 3 and 4

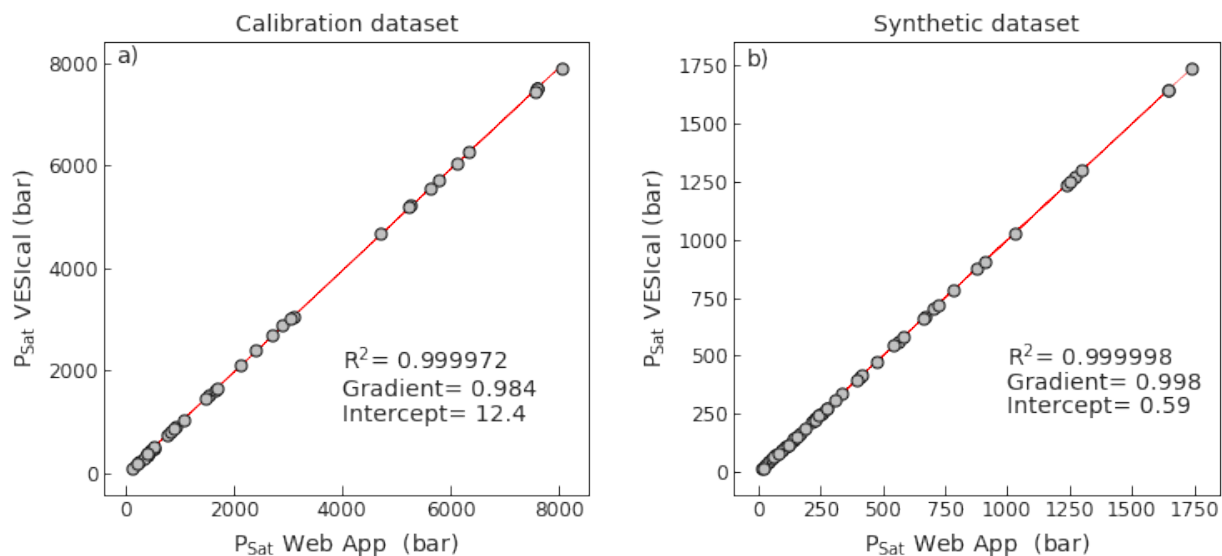


Figure 13. a) Comparison of saturation pressures from the web app of Iacono-Marziano (<http://calcul-isto.cnrs-orleans.fr/apps/h2o-co2-systems/>) and VESICAL for the major element and volatile contents of the calibration dataset (a) and a synthetic array of inputs (b).

Iacono-Marziano et al., 2012 - Test 5

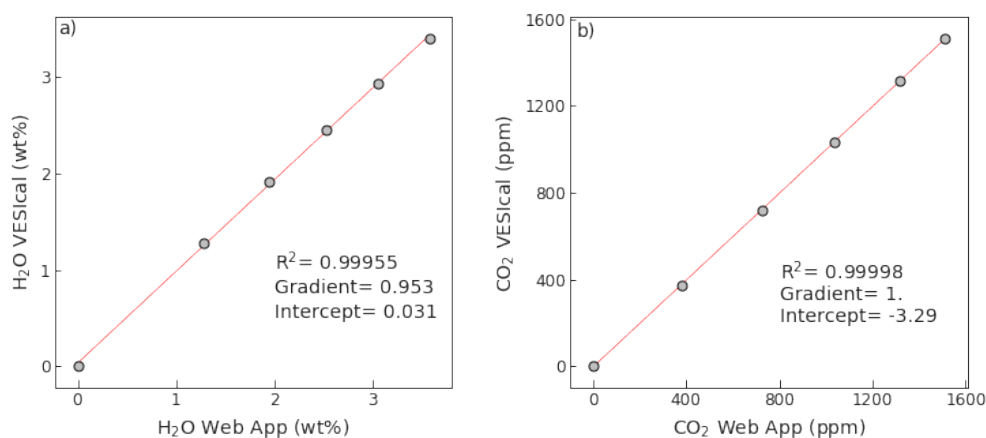


Figure 14. Comparison of dissolved volatiles calculated using the web app and VESICAL for a synthetic dataset.

Shishkina et al., 2014 - Test 1 and 2

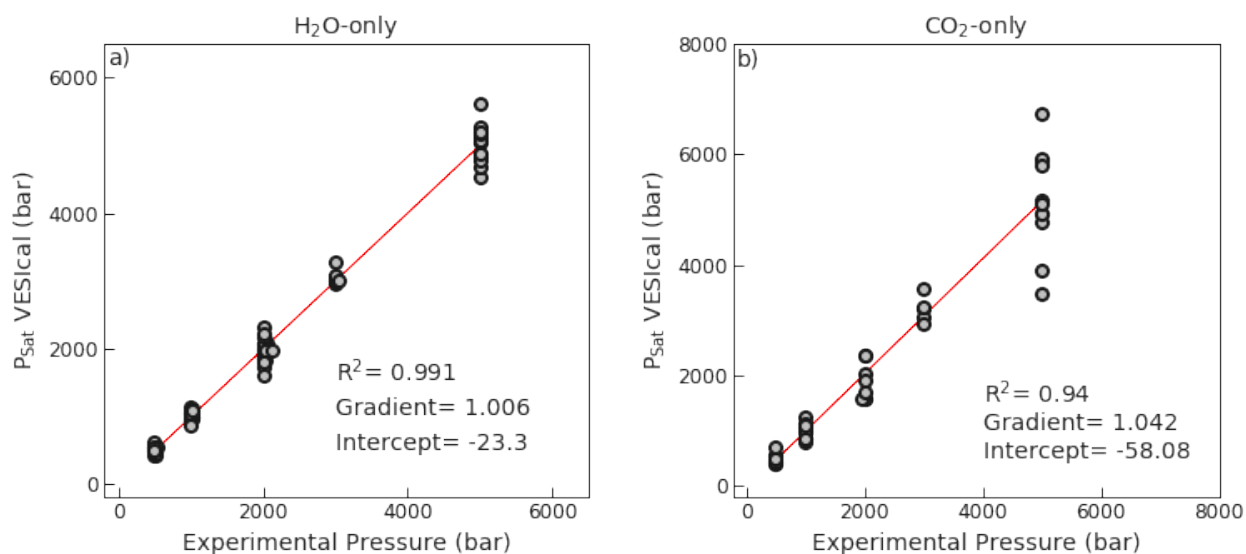


Figure 15. Comparison of the experimental pressures in H₂O-only experiments (a) and the CO₂ experiments (b) in the calibration dataset with the saturation pressures calculated by VESICAL.

Shishkina et al., 2014 - Test 3

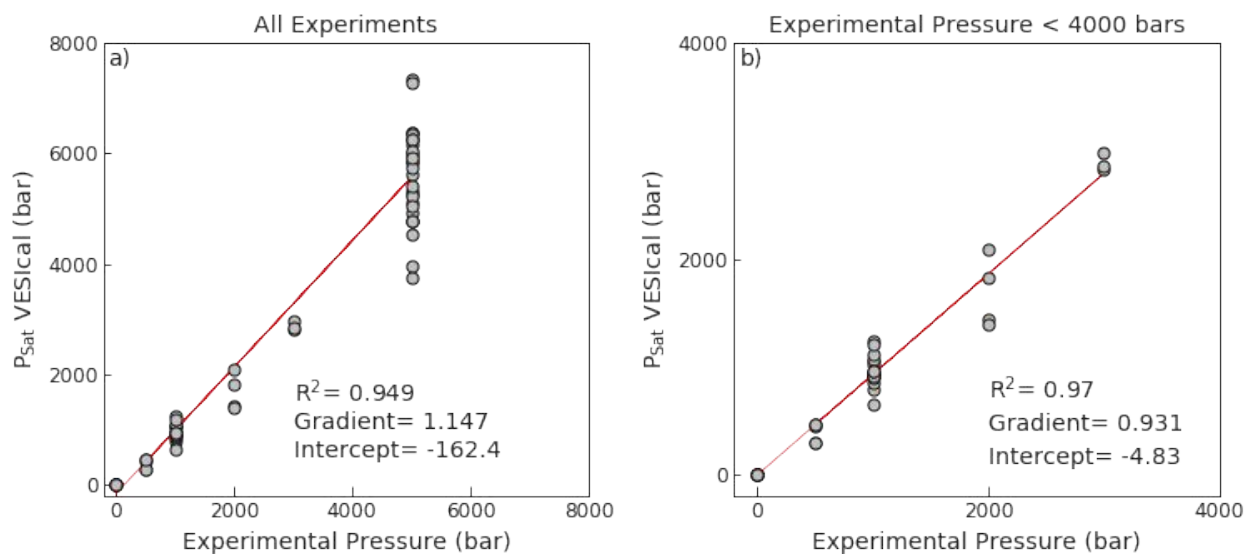


Figure 16. a) Comparison of the experimental pressures for the H₂O-CO₂ experiments presented in Table 2 of the main text with the saturation pressures calculated by VESICAL. b) Regression excluding the experiments run at >4000 bars, which show considerable scatter.

MagmaSat (Ghiorso and Gualda, 2015) - Test 1

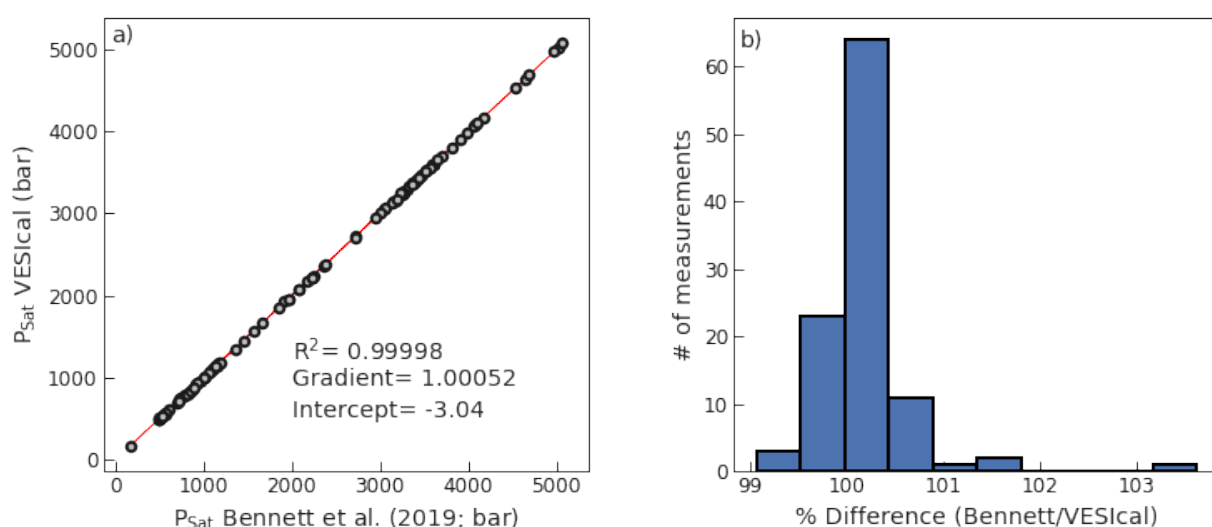


Figure 17. Comparison of saturation pressures calculated in VESICAL to those published by Bennett et al. (2019, who used the Mac App). These results agree to within 1% (within error of the rounding within VESICAL to the nearest MPa).

MagmaSat (Ghiorso and Gualda, 2015) - Test 2

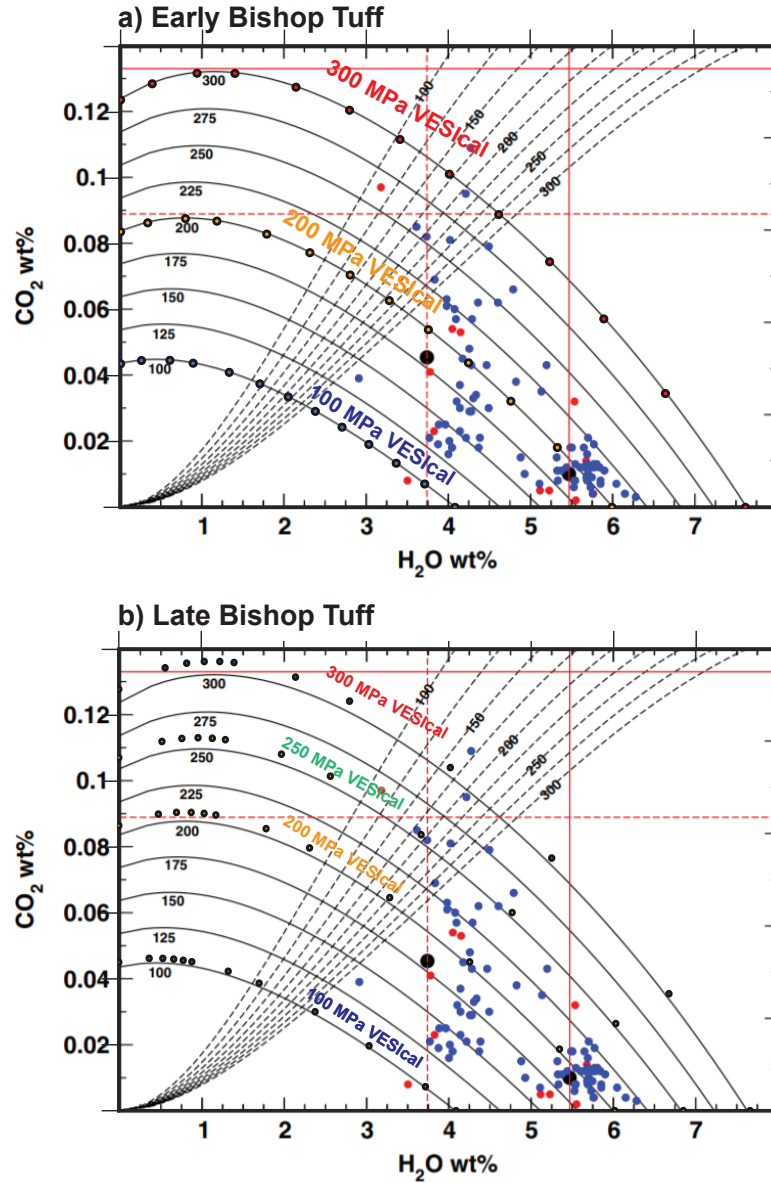


Figure 18. Comparison of isobars produced in VESICAL for the Bishop Tuff (colored dots) to those shown in Fig. 14 of Ghiorso and Gualda (lines). The best match uses the Early Bishop Tuff composition from the table in their paper.

MagmaSat (Ghiorso and Gualda, 2015) - Test 3

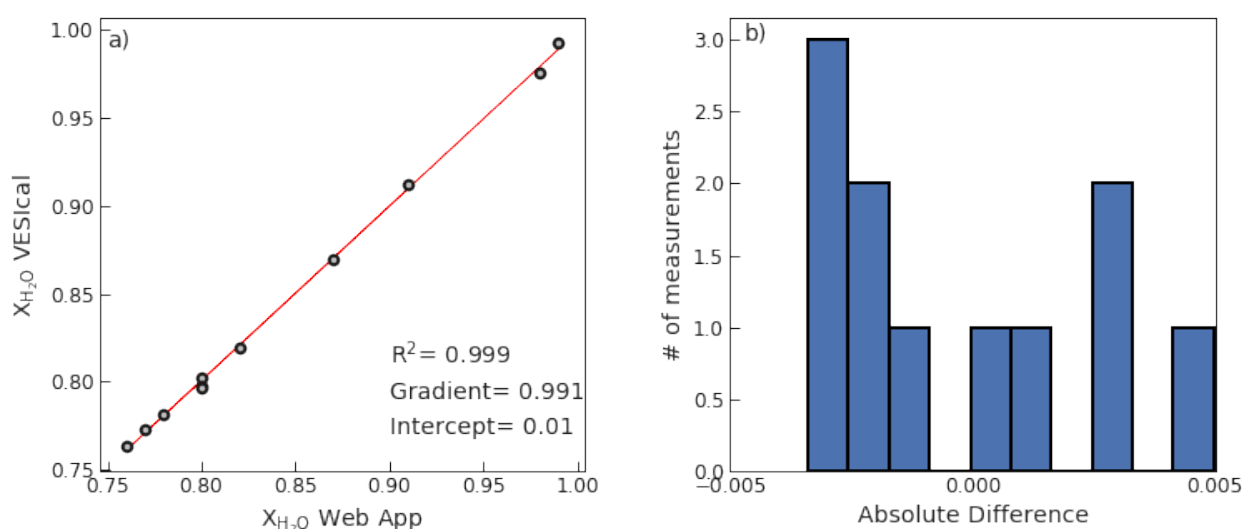


Figure 19. Comparison of X_{H_2O} from VESICAL and the web app (<http://209.180.202.74:8080/H2O-CO2BxApp/>) for a synthetic array of inputs. The web app only returns values to the nearest 0.01. The discrepancies between VESICAL and the web app lie within this rounding error.

Allison et al. (2019) - Test 1

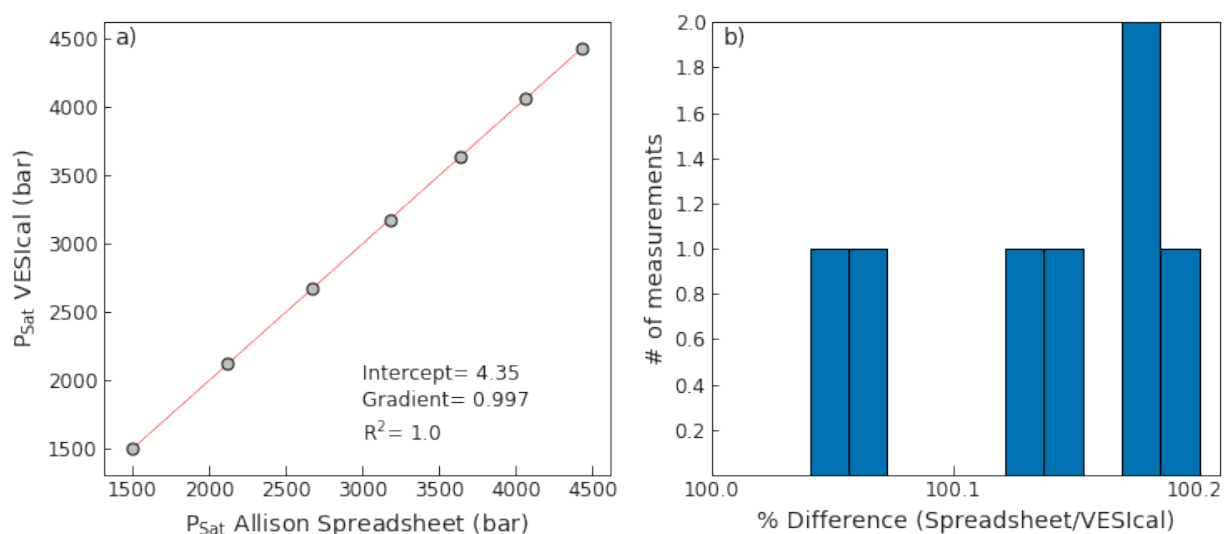


Figure 20. a) Comparison of saturation pressures from the Excel spreadsheet provided by Allison et al. (2019) for the Sunset Crater Model for a synthetic array of inputs.

Allison et al. (2019) - Test 2

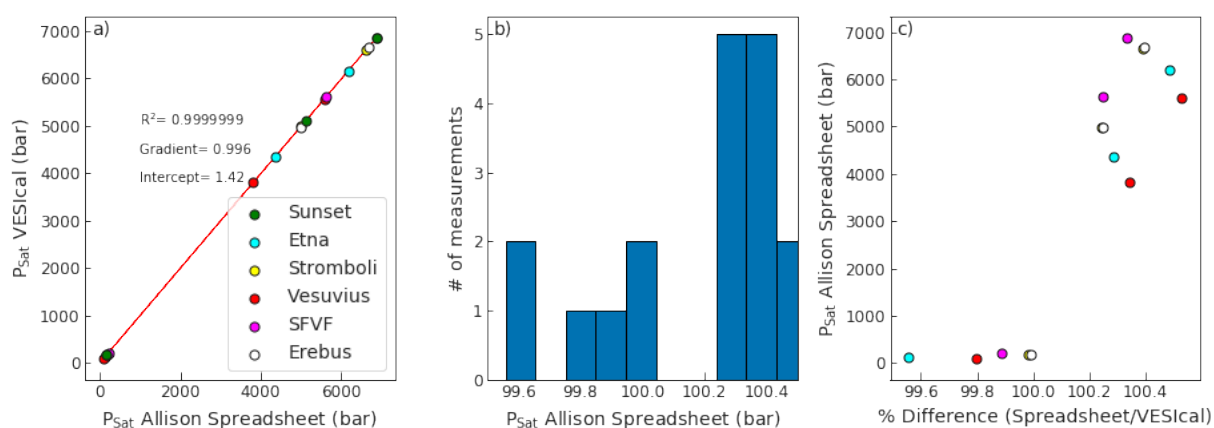


Figure 21. a) Comparison of saturation pressures from the Excel spreadsheet provided by Allison et al. (2019) for all 6 model compositions for a synthetic array of inputs. b) VESICAL matches the spreadsheet to within $\pm 0.5\%$. c) The percentage difference increases with increasing pressure.