



Data Article

PyNomo software dataset for sciences and engineering nomogram construction

Pedro Martínez-Pagán^{a,*}, Leif Roschier^b^aDepartment of Mining and Civil Engineering, Universidad Politécnica de Cartagena, Cartagena 30203 Spain^bVantaa, Finland

ARTICLE INFO

Article history:

Received 22 June 2022

Revised 24 September 2022

Accepted 3 October 2022

Available online 7 October 2022

Keywords:

Nomography

Alignment charts scripts

Python-based code

Sciences and engineering computing

PyNomo

ABSTRACT

This article examines different python-based codes to be run under PyNomo nomographer software to represent alignment charts or nomograms for formulas commonly used in sciences and engineering studies. PyNomo nomographer supports nine basic types of nomograms based on the format of the mathematical equation. The examined PyNomo codes cover the construction of the following types of nomogram: type 1, type 2, type 7, type 8, type 9, and type 10. Each PyNomo script includes some background and the underlying mathematics for the nomogram construction and how to implement each code into PyNomo software. Potential users can use these PyNomo codes, as template scripts, to customize and construct their own nomograms for a variety of technical applications. That is especially true since a considerable number of science and engineering equations fall into one of these covered nomogram types depending on the equation form involved.

© 2022 The Author(s). Published by Elsevier Inc.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

* Corresponding author.

E-mail address: p.martinez@upct.es (P. Martínez-Pagán).

Specifications Table

Subject	Computer Science
Specific subject area	Computer Graphics and Computer-Aided Design. Computer nomograph construction aided by PyNomo nomographer software for science and engineering applications.
Type of data	Python-based code (for PyNomo nomographer software). Images
How the data were acquired	All source codes written in Python language and run through PyNomo software to produce nomograms or abacs in a PDF vector form.
Data format	PyNomo monographer software scripts (.py extension).
Description of data collection	Most PyNomo scripts are listed in a single file. These files were run under PyNomo software to generate each graph that solves the mathematical equation from which the nomogram is based on.
Data source location	Institution: Department of Mining and Civil Engineering, Universidad Politécnica de Cartagena City/Town/Region: Cartagena Country: Spain
Data accessibility	With the article and in Zenodo repository (please refer to References section for Zenodo repository URL addresses and data identification name (DOI))
Related research article	P. Martínez-Pagán, L. Roschier, <i>Nomography: A renewed pedagogical tool to sciences and engineering high-education studies</i> , <i>Heliyon</i> . (2022). 10.1016/j.heliyon.2022.e09731 (Available online 15 June 2022, e09731).

Value of the Data

- These PyNomo codes have significant value to potential PyNomo users in 2 ways: 1) they enable to realize of the actual capability of nomography to solve science and engineering equations, and 2) they will help to grasp the basic to produce useful nomograms for a wide range of mathematical expressions of common use in science and engineering.
- Scientists, engineers and/or teachers could benefit from these datasets since they provide the basic skills to design and interpret nomograms as well as letting them include nomography as a valuable computational tool in an academic context, such as sciences and engineering studies.
- With minimal editing effort, these PyNomo codes can be easily modified to produce nomograms for other types of equations where they could benefit of PyNomo software to produce in seconds outstanding and customized nomograms.

1. Data Description

In nomography, one of the most difficult or demanding stages is drawing reliable scales or grids. Presumably that has been one of the main reasons for the decline of nomogram use in an academic environment, in which the appearance of more capable and powerful personal computers and handheld calculators helped to intensify this tendency. However, the introduction of PyNomo software [1] solves the most cumbersome point of manually constructing precise nomograms. Thus, this software, which has been written by one of the authors (LR), is a significant step forward to promote users to introduce nomography successfully into an academic context. PyNomo can be found at <http://lefakkomies.github.io/pynomo-doc/installation/installation.html>. PyNomo is a Python library and thus requires the installation of several other Python-based software packages, such as numpy, scipy and pyx that requires LaTeX-installation. From more recent PyNomo versions (presently, from 0.3.0 to 0.3.2), the software is compatible with Python 2 and Python 3. For editing PyNomo scripts any text browser works but integrated development environment (IDE) for Python can speed up developments (e.g. PyCharm community edition and

spyder). Also, comprehensive and detailed instructions for the successful installation of PyNomo on Windows, as well as OSX and Linux, platforms are found on the previously indicated website.

PyNomo scripts are written in text files saved with the suffix .py since they are interpreted by Python. Once the PyNomo script is run a PDF or EPS file is created, which is vector-based so it is totally scalable and suitable for additional artwork is desired. Moreover, PyNomo supports nine basic types of nomograms based simply on the format of the equation (Fig. 1).

2. Design, Materials and Methods

This section examines the methodology used for different python-based codes written to be run under PyNomo software to construct an appropriate representation of nomogram examples for equations commonly used in sciences and engineering studies. These examples enable potential users to grasp the importance and significant capabilities of PyNomo software in a science and engineering learning context. In line with that, the nomogram types covered with these examples will be type 1, type 2, type 7, type 8, type 9, and type 10. It is worth noting that these scripts are reusable since they can graphically represent a great number of common mathematical equations, by means of PyNomo software, with the appropriate code editing for the needs of each end-user.

The employ of these dataset to solve their associated nomograms by means of PyNomo software avoid the need of manually constructing the nomographic charts [2]. This point is relevant since the construction of nomographic charts is time demanding and require certain degree of skillfulness for obtaining them accurately [2]

2.1. Type 1 Nomogram Code

The code provided here below constructs Type 1, or 3 parallel scales nomogram. This type of script plots a considerable number of functions, commonly employed in engineering, of the form:

$$f_1(u_1) + f_2(u_2) + f_3(u_3) = 0 \quad (1)$$

Where u_1 , u_2 , and u_3 are the scale variables.

The code of this example enables to graph the slurry flow equation for pumped pipes, which is written as:

$$Q_{slurry} = 0.28 * \left(\frac{\%Water}{\%Solids} + \frac{1}{\rho_{solids}} \right) \quad (2)$$

Where Q_{slurry} is the slurry flow per t/h of solids, in l/s; $\%Water$ indicates the percentage of water in the slurry; $\%Solids$ indicates the percentage of solids in the slurry; and ρ_{solids} is the specific density, in g/cm^3 . Now, by arranging the terms of this formula, the form that satisfies the expression (1) is:

$$\frac{Q_{slurry}}{0.28} - \frac{100 - \%Solids}{\%Solids} - \frac{1}{\rho_{solids}} = 0 \quad (3)$$

Since:

$$f_1(Q_{slurry}) - f_2(\%Solids) - f_3(\rho_{solids}) = 0 \quad (4)$$

Below is the PyNomo script that makes this 3 parallel scales nomogram. The script has a typical and simple structure composed of a different set of parameters. Commonly, this set of parameters comprises: scale parameters, which define the function, variable title, etc., block parameters that define the scale name, its type and size, and main parameters constituted by the block names, paper size, equation title and transformations to apply. Thus, the scale parameters of the Type 1 nomogram code of this example are defined by "S_params", "P_params", and

Nomogram type	Form of equation	Representation
1	$f_1(u_1) + f_2(u_2) + f_3(u_3) = 0$	
2	$f_1(u_1) = f_2(u_2) * f_3(u_3);$ $f_1(u_1) = \frac{f_2(u_2)}{f_3(u_3)}$	
3	$f_1(u_1) + f_2(u_2) + \dots + f_n(u_n) = 0$	
4	$\frac{f_1(u_1)}{f_2(u_2)} = \frac{f_3(u_3)}{f_4(u_4)}$	
5	$f_1(v) = f_2(x, u)$	
6	$u = u$	
7	$\frac{1}{f_3(u_3)} = \frac{1}{f_1(u_1)} + \frac{1}{f_2(u_2)}$	
8	$y = f(u)$	
9	$\begin{vmatrix} f_1(u_1, v_1) & g_1(u_1, v_1) & h_1(u_1, v_1) \\ f_2(u_2, v_2) & g_2(u_2, v_2) & h_2(u_2, v_2) \\ f_3(u_3, v_3) & g_3(u_3, v_3) & h_3(u_3, v_3) \end{vmatrix}$	
10	$f_1(u_1) + f_2(u_2) * f_3(u_3) + f_4(u_3) = 0$	

Fig. 1. Types of nomograms supported by PyNomo software.

“V_params”. The block parameters are defined by “block_1_params”, and the main parameters by “main_params”. In the script, the variable u represents the variables of the functions for the Eq. (4). Also, the script includes the construction of an isopleth that helps interpret the nomogram. The code of this isopleth is written in “block_1_params” where the values of $\%Solids$ and ρ_{solids} , which are 40 and 2.65, respectively, will define the value of Q_{slurry} when the isopleth intersects the Q_{slurry} scale. The supplementary material provides this final nomogram under scalable format. Also, the script is available in the Zenodo repository [3]. It worth noting that this script could work as a guideline script to reproduce other equations of the form (1) with the appropriate modifications.

Type 1 nomogram code. Slurry flow in pipes:

```

### Type1_nomogram.py ###

from pynomo.nomographer import *

S_params={
'u_min':10.0,
'u_max':80.0,
'function':lambda u:-(100-u)/u,  →  $-\frac{100-u}{u} = -\frac{100-\%Solids}{\%Solids}$ 
'title':r'$\%_{Solids}$',
'tick_levels':3,
'tick_text_levels':1,
'tick_side':'left',
}

P_params={
'u_min':0.0,
'u_max':3.0,
'function':lambda u:u/0.28,  →  $\frac{u}{0.28} = \frac{Q_{slurry}}{0.28}$ 
'title':r'$Q-1/s$',
'tick_levels':3,
'tick_text_levels':1,
'tick_side':'left',
}

V_params={
'u_min':1.0,
'u_max':5.0,
'function':lambda u:-1/u,  →  $-\frac{1}{u} = -\frac{1}{\rho_{solids}}$ 
'title':r'$\rho_{s}-gr/cm^3$',
'tick_levels':3,
'tick_text_levels':1,
}

main_params={
'filename': 'Type1_nomogram.pdf',
'paper_height':20.0,
'paper_width':10.0,
'block_params':[block_1_params],
'transformations':[('rotate',0.01),('scale paper',),('polygon',)],
'title_x':5.0,
'title_y':-1.0,
'title_box_width':10.0,
'title_str':r'$Q = 0.28 \times ((\%_{Solids}/\%_{Water})+(1/\rho_{s}))$',
'isopleth_params':[
{'color':'black',
'linewidth':'thick',
'linestyle':'dashed',
'circle_size':0.10,
'transparency':0.0},
],
]
}

Nomographer (main_params)

```

2.2. Type 2 Nomogram Code

This code constructs a Type 2, or N/Z nomogram, which graphically presents quite common functions in sciences and engineering studies that are under the form (included all their variations):

$$f_2(u_2) = \frac{f_1(u_1)}{f_3(u_3)} \quad (5)$$

Where u_1 , u_2 , and u_3 , are the scale variables.

Below is the PyNomo script for the index of refraction nomogram, whose equation satisfies the form (7):

$$I_R = \frac{\sin I}{\sin R} \quad (6)$$

Knowing that I_R is the index of refraction, dimensionless (from 0 to 10); I is the angle of incidence, in degrees (from 0° to 90°); and R is the angle of refraction, in degrees (from 0° to 90°).

In the script the variables u_1 , u_2 , and u_3 , represent I , I_R , and R . It has to be highlighted that PyNomo scripts require to formulate trigonometric functions in radians. The script has a typical and simple structure composed of a different set of parameters. Potential users can adjust this script to other similar formulations by making appropriate changes. The script is available in the Zenodo repository [4] to help accomplish this goal. Again, the supplementary material provides the scalable nomogram for this equation.

Type 2 nomogram code. Index of refraction:

```

### index_of_refraction.py ###
from pynomo.nomographer import *

N_params_1={
    'u_min':0,
    'u_max':90,
    'function':lambda u:sin(u*pi/180), →  $f_1(u_1) = \sin I$ 
                                     (in radians)
    'title':r'$I$',
    'tick_levels':3,
    'tick_text_levels':2,
    'tick_side':'left',
    }

N_params_2={
    'u_min':0,
    'u_max':10,
    'function':lambda u:u, →  $f_2(u_2) = I_R$ 
    'title':r'$index(IR)$',
    'tick_levels':3,
    'tick_text_levels':2,
    'title_y_shift':0.3,
    'title_x_shift':0.8,
    }

N_params_3={
    'u_min':0,
    'u_max':90,
    'function':lambda u:sin(u*pi/180), →  $f_3(u_3) = \sin R$ 
                                     (in radians)
    'title':r'$R$',
    'tick_levels':3,
    'tick_text_levels':2,
    'tick_side':'right',
    }

block_1_params={
    'block_type':'type_2',
    'width':10.0,
    'height':10.0,
    'f1_params':N_params_1,
    'f2_params':N_params_2,
    'f3_params':N_params_3,
    'isopleth_values':[['x',0.5,90]],
    }

main_params={
    'filename':'index_of_refraction.pdf',
    'paper_height':25.0,
    'paper_width':30.0,
    'block_params':[block_1_params],
    'transformations':[('rotate',0.01),('scale paper',)],
    'title_str':r'\LARGE$I_{R}=\frac{\sin I}{\sin R}$',
    'isopleth_params':[
        {'color':'Gray',
         'linewidth':'thin',
         'circle_size':0.05,
         'transparency':0.0,
        },
    ],
}

Nomographer(main_params)

```

2.3. Type 7 Nomogram Code

This code constructs a Type 7 nomogram, which graphically displays functions related by their reciprocals. Thus, the nomogram constructed with the script of this example represents the equation:

$$\frac{1}{F_1} - \frac{1}{F_2} = \frac{1}{F_3} \quad (7)$$

Where F_1 , F_2 , and F_3 , are variables.

Below is the PyNomo script that makes a three scales reciprocal type nomogram. In the script, the variable u represents the functions of the Eq. (9) that in this example are F_1 , ranging from 10 to 100 (N_params_1), F_2 ranging from 10 to 300 (N_params_2), and F_3 ranging from 0 to 100 (N_params_3). Also, the script includes the construction of an isopleth that helps interpret the nomogram. The code of this isopleth is written in "block_1_params" where the values of F_1 and F_3 , which are 50 and 80, respectively, will define the value of F_2 when the isopleth intersects the F_2 scale. The supplementary material provides the created nomogram by PyNomo software. Also, the script ready to be run under PyNomo software is available in the Zenodo repository [5].

Type 7 nomogram code:

```

### Code_Type7.py ###
from pynomo.nomographer import *

N_params_1={
    'u_min':10,
    'u_max':100.0,
    'function':lambda u:u, →  $u = F_1$ 
    'title':r'$F_1$',
    'tick_levels':3,
    'tick_text_levels':1,
    ...
}

N_params_2={
    'u_min':10,
    'u_max':300.0,
    'function':lambda u:-u, →  $-u = -F_2$ 
    'title':r'$F_2$',
    'title_x_shift':-0.8,
    'title_y_shift':-0.05,
    'tick_levels':3,
    'tick_side':'left',
    'tick_text_levels':1,
    ...
}

N_params_3={
    'u_min':0,
    'u_max':100.0,
    'function':lambda u:u, →  $u = F_3$ 
    'title':r'$F_3$',
    'title_x_shift':0.2,
    'title_y_shift':0.2,
    'tick_levels':3,
    'tick_side':'left',
    'tick_text_levels':1,
    ...
}

block_1_params={
    'block_type':'type_7',
    'width':10.0,
    'height':10.0,
    'f1_params':N_params_1,
    'f2_params':N_params_2,
    'f3_params':N_params_3,
    'angle_u':30.0,
    'angle_v':70.0,
    'isopleth_values':[[50,'x',80]],
}

main_params={
    'filename':'Code_Type7.pdf',
    'paper_height':10.0,
    'paper_width':10.0,
    'block_params':[block_1_params],
    'transformations':[('rotate',0.01),('scale paper',)],
    'title_str':r'$1/F_1-1/F_2=1/F_3$',
    'title_x':6.0,
    'title_y':3.0,
    'debug':False,
}

Nomographer(main_params)

```

2.4. Type 1/Type 8 Combined Nomogram Code

This example examines the combination of two types of nomograms to represent the earthquake magnitude determination: type 1 and type 8 nomogram. This nomogram combination is especially useful when is required variable conversion taking place into a nomogram of any type. The variable conversion is obtained with the code provided for type 8 nomograms, which is named as single, simple function scale. It provides a means of aligning (or overlaying) a scale right on top of another scale using the parameter “tag”. Thus, scales with the same “tag” parameter will be aligned (overlaid) in the final nomogram. In this example the magnitude conversion between earthquake Richter magnitude and moment magnitude. The earthquake Richter magnitude function can be written as:

$$M_L - \log_{10}(A_{\max}) + 0.000008 \times D^2 - 0.01 \times D - 1.8783 = 0 \quad (8)$$

Where M_L is the Richter magnitude, dimensionless; D is the distance between the point of earthquake occurrence and the seismic recording site, in km; A_{\max} is the maximum amplitude, in mm.

Thus, the Eq. (9) fulfils a type 1 nomogram of the form:

$$f_1(M_L) + f_2(A_{\max}) + f_3(D) = 0 \quad (9)$$

On the other hand, the expression to convert Richter magnitude values to moment magnitude (M_w) values is given by Mereu [6]:

$$M_w = 0.59 \times M_L + 1.32 \quad (10)$$

Now the Eqs. (8) and (10) satisfy type 1 and type 8 nomogram expressions, respectively, so the code can be easily written. This finished code is provided below where “block_2_params” defines the code for creating type 8 nomograms, as it is described by Doerfler [7], and where the “tag” parameter is named ‘ML’. Also, the script is available in the Zenodo repository [8].

Type 1/Type 8 combined nomogram code. Earthquake seismic magnitude:

```

### Richter_Moment_Magnitudes2.py ###
from pynomo.nomographer import *
D_params={
    'u_min':0.0,
    'u_max':500,
    'function':lambda u:0.000008*u*u-0.01*u-1.8783,
    'title':r'$D$',
    'tick_levels':3,
    'tick_text_levels':2,
    'tick_side':'left',
}
P_params={
    'tag':'ML',
    'u_min':0.0,
    'u_max':10.0,
    'function':lambda u:u,
    'title':r'$M_L$',
    'title_x_shift':-0.2,
    'title_y_shift':0.2,
    'tick_levels':3,
    'tick_text_levels':1,
    'tick_side':'left',
}
V_params={
    'u_min':0.01,
    'u_max':100,
    'function':lambda u:-log10(u),
    'scale_type':'log',
    'title':r'$A_{max}$',
    'tick_levels':3,
    'tick_text_levels':1,
}
block_1_params={
    'block_type':'type_1',
    'width':10.0,
    'height':10.0,
    'f1_params':D_params,
    'f2_params':P_params,
    'f3_params':V_params,
    'mirror_y':True,
    'isopleth_values':[[192,'x',23]],
}
M_params={
    'tag':'ML',
    'u_min':2,
    'u_max':7,
    'function':lambda u:0.59*u+1.32,
    'align_func':lambda u:(u-1.32)/0.59,
    'title':r'$M_w$',
    'title_x_shift':0.5,
    'tick_levels':3,
    'tick_text_levels':1,
    'tick_side':'right',
}
block_2_params={
    'block_type':'type_8',
    'width':10.0,
    'height':10.0,
    'f_params':M_params,
    'isopleth_values':[['x']],
}
main_params={
    'filename':'Richter_Moment_Magnitudes2.pdf',
    'paper_height':10.0,
    'paper_width':10.0,
    'block_params':[block_1_params,block_2_params],
    'transformations':[('rotate',0.01),('scale paper',)],
    'make_grid':False,
    'title_x':5.0,
    'title_y':-1.0,
    'title_box_width':10.0,
    'title_str':r'$M_L=log_{10}\{A_{max}\}-0.000008*D^2+0.01*D-1.8783$',
    'extra_texts':[
        {
            'x':3,
            'y':-2,
            'text':r'$M_w=0.59*M_L+1.32$',
            'width':5,
        }
    ],
}
Nomographer (main_params)

```

$f_1(D) = 0.000008 \times u^2 - 0.01 \times u - 1.8783$

$f_2(M_L) = u$

$f_3(A_{max}) = -\log_{10} u$

$M_w = 0.59 \times u + 1.32$

2.5. Type 8 Wheel Nomogram Code

This code consists of a set of circular scales based on Type 8 nomogram to convert pressure units in a very convenient way, which are of common use in the fields of hydraulics, geotechnics, physics, etc. Type 8 or single scale nomogram, which another example was described in [Section 2.4](#), is the most basic nomogram type satisfying the basic form of:

$$y = f(u) \tag{11}$$

This Type 8-based circular nomogram was constructed following the comprehensive indications provided by Doerfler [7] to deal with this type of curved nomogram.

Below is provided the script block to construct the atmosphere unit curved scale from 0.1 to 9.99 atms, in which the *function_x* and *function_y* parameters allow the scale curvature to be defined by functions of *u* in terms of *x* and *y*. In this example, a linear scale from 0.1 to 9.99 atms to be laid out in a circle of radius 7.0 cm, should be written as:

$$\begin{aligned} 'function_x' : \text{lambda}u : 7.0 * \cos(2 * \pi * (\log_{10}(u) - \log_{10}(0.1)) / (\log_{10}(9.99) - \log_{10}(0.1))), \\ 'function_y' : \text{lambda}u : 7.0 * \sin(2 * \pi * (\log_{10}(u) - \log_{10}(0.1)) / (\log_{10}(9.99) - \log_{10}(0.1))), \end{aligned} \tag{12}$$

By using the same radius but with the ticks on the opposite side, two scales can overlay. The other scales provide the conversion of PSI, mm Hg, kN/m², kPa, kg/cm², Bar, and m.c.a units. All the scales repeat the same basic set of parameters, making it easy to write the script. For sake of simplicity only the atmosphere unit script block is provided, however the complete script is available in the Zenodo repository [9] to help to obtain the circular nomogram.

Type 8 nomograma code. Atmosphere unit curved scale block:

```

atm_params={
'u_min':0.1,
'u_max':9.99,
'base_start':0.1,
'base_stop':10.0,
'function_x':lambda u:7.0*cos(2*pi*(log10(u)-log10(0.1))/(log10(9.99)-log10(0.1))),
'function_y':lambda u:7.0*sin(2*pi*(log10(u)-log10(0.1))/(log10(9.99)-log10(0.1))),
'title':r'\LARGE $Atm$',
'title_x_shift':-0.10,
'title_y_shift':-1.3,
'tick_levels':5,
'tick_text_levels':5,
'scale_type':'linear smart',
'tick_side':'left',
'turn_relative':True,
'grid_length_0':1.20,
'grid_length_1':0.90,
'grid_length_2':0.60,
'grid_length_3':0.45,
'grid_length_4':0.30,
'grid_length_5':0.25,
'text_distance_0':1.25,
'text_distance_1':0.95,
'text_distance_2':0.65,
'text_distance_3':0.45,
'text_distance_4':0.35,
'text_distance_5':0.30,
'text_size_0':text.size.scriptsized,
'text_size_1':text.size.tiny,
'extra_angle':90,
'text_horizontal_align_center':True,
'full_angle':True,
'axis_color':color.cmyk.Red,
'text_color':color.cmyk.Red,
'title_color':color.cmyk.Red,
}
atm_block_params={
'block_type':'type_8',
'f_params':atm_params,
'width':15.0,
'height':15.0,
}

```

2.6. Type 9 Nomogram Code

Type 9 nomogram code enables to construct of those mathematical expressions that can be put into determinant form as follows:

$$\begin{vmatrix} f_1(u_1, v_1) & g_1(u_1, v_1) & h_1(u_1, v_1) \\ f_2(u_2, v_2) & g_2(u_2, v_2) & h_2(u_2, v_2) \\ f_3(u_3, v_3) & g_3(u_3, v_3) & h_3(u_3, v_3) \end{vmatrix} = 0 \quad (13)$$

Where u_i , and v_i , are variables.

Accordingly, the trajectory of a projectile could be determined through a type 9 nomogram, where its expression is given by [10]:

$$y = x \cdot \tan \alpha - \frac{g \cdot x^2}{2 \cdot V_0^2 \cdot \cos^2 \alpha} \quad (14)$$

Knowing that Eq. (14) can be put under the determinant form that justifies the standard expression for Type 9 nomogram:

$$\begin{vmatrix} \frac{g}{2 \cdot V_0^2} & 0 & 1 \\ y & 1 & 1 \\ \frac{x \cdot \tan \alpha \cdot \cos^2 \alpha}{x^2 + \cos^2 \alpha} & \frac{\cos^2 \alpha}{x^2 + \cos^2 \alpha} & 1 \end{vmatrix} = 0 \tag{15}$$

Next the complete script is provided, in which 'f' represents the function in u , given by the first column in the determinant row corresponding to the scale, 'g' represents the function in u , given by the second column in the determinant row corresponding to the scale, and 'h' represents the function in u , given by the third column in the determinant row corresponding to the scale. Consequently, following that criterion functions in the determinant (15) can be expressed as:

$$\begin{aligned} f_1(u_1) &= \frac{g}{2 \cdot V_0^2}; g_1(u_1) = 0; h_1(u_1) = 1 \\ f_2(u_2) &= y; g_2(u_2) = 1; h_2(u_2) = 1 \\ f_3(u_3, v_3) &= \frac{x \cdot \tan \alpha \cdot \cos^2 \alpha}{x^2 + \cos^2 \alpha}; g_3(u_3, v_3) = \frac{\cos^2 \alpha}{x^2 + \cos^2 \alpha}; h_3(u_3, v_3) = 1 \end{aligned} \tag{16}$$

Therefore, these functions are written accordingly in the script provided below, noting that angles are to be expressed in radians. Also, the complete script is available in the Zenodo repository [11] to help to obtain the type 9 based projectile trajectory nomogram using PyNomo software.

Type 9 nomogram code. The projectile trajectory:

```

### Type9_Grid2.py ###
from pynomo.nomographer import *
V_params={
'u_min':0.1,
'u_max':10.0,
'f':lambda u:0.00491/(u**2),
'g':lambda u:0,
'h':lambda u:1.0,
'title':r'\Large $V_0$ ',
'title_x_shift':0.4,
'title_y_shift':0.0,
'scale_type':'log',
'tick_levels':3,
'tick_text_levels':1,
'grid':False,
}
Y_params={
'u_min':0.0,
'u_max':0.5,
'f':lambda u:u,
'g':lambda u:1.0,
'h':lambda u:1.0,
'title':r'\Large $y$ ',
'title_x_shift':0.4,
'tick_levels':3,
'tick_text_levels':1,
'tick_side':'left',
'grid':False,
}
X_A_params={
'u_min':0.1,
'u_max':5.0,
'f_grid':lambda u,v:u*tan(v**2*pi/360.0)*(cos(v**2*pi/360.0)**2)/(u**2+(cos(v**2*pi/360.0)**2)),
'g_grid':lambda u,v:((cos(v**2*pi/360.0)**2)/(u**2+(cos(v**2*pi/360.0)**2)),
'h_grid':lambda u,v:1.0,
'u_start':0.1,
'u_stop':5.0,
'v_start':0.0,
'v_stop':60,
'u_values':[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1.5,2,3,4,5],
'v_values':[0,5,10,20,30,40,45,50,60],
'u_texts':['0.1','0.2','0.3','0.4','0.5','0.6','0.7','0.8','0.9','1','1.5','2','3','4','5'],
'v_texts':['0','5','10','20','30','40','45','50','60'],
'v_texts_u_start':True,
'u_texts_v_start':True,
'u_texts_v_stop':False,
'text_distance':0.5,
'u_line_color':color.cmyk.Green,
'u_text_color':color.cmyk.Green,
'v_line_color':color.cmyk.Red,
'v_text_color':color.cmyk.Red,
'text_prefix_u':r'$x$',
'text_prefix_v':r'$\alpha$',
'grid':True,
}
block_params={
'block_type':'type_9',
'f1_params':V_params,
'f2_params':Y_params,
'f3_params':X_A_params,
'transform_inl':False,
'isopleth_values':[['x',0.28,[0.5,30]]]
}
main_params={
'filename':'Type9_Grid2.pdf',
'paper_height':20.0,
'paper_width':30.0,
'block_params':[block_params],
'transformations':[['rotate',0.01],('scale paper',)],
'make_grid':False,
'title_x':8,
'title_y':-2,
'title_box_width':10,
'title_str':r'$\huge y=x*\tan\alpha-\{g*x^2\}/\over{2*V_0^2*\cos^2\alpha}$',
'isopleth_params':[
{'color':'black',
'linewidth':'thin',
'linestyle':'dashed',
'transparency':0.0,
'circle_size':0.08,
}],
},
}
Nomographer(main_params)

```

$$f(u) = \frac{g}{2 * V_0^2} = \frac{0.00491}{u^2}$$

$$g_1(u) = 0$$

$$h_1(u) = 1$$

$$f_2(u_2) = y$$

$$g_2(u_2) = 1$$

$$h_2(u_2) = 1$$

$$f_3(u, v) = \frac{x \cdot \tan \alpha \cdot \cos^3 \alpha}{x^2 + \cos^2 \alpha}$$

$$g_3(u, v) = \frac{\cos^2 \alpha}{x^2 + \cos^2 \alpha}$$

$$h_3(u, v) = 1$$

2.7. Type 10 Nomogram Code

A Type 10 nomogram consists of two straight scales and one curved scale. This type of nomogram represents equations of the form:

$$f_1(u_1) + f_2(u_2) * f_3(u_3) + f_4(u_4) = 0 \quad (17)$$

Where u_i are variables.

The Type 10 nomogram code is used in this example to graph the chemical equation of Van Der Waals, which is commonly written as:

$$\left(p + \frac{a}{v^2}\right) * (v - b) = R * (t + 273.15) \quad (18)$$

Here p is the pressure, in Atm; v is the volume, in litres, L; R is a gas constant, in L-Atm/mol-°K; t is the absolute temperature, in centigrades degrees, °C; a and b are constants which depends on the type of gas. In this example, the carbon dioxide (CO₂) gas will be considered, whose R , a , and b constants are 0.00366, 0.00717, and 0.00191, respectively.

Thus, Eq. (18) can be put under a form that satisfies expression (17) as follows:

$$f_1(p) - f_2(t) * f_3(v) + f_4(v) = 0 \quad (19)$$

Therefore, the expression that is obtained once the different variables of Eq. (19) are properly arranged, in relation to the expression (17), will be:

$$p - t * \frac{0.00366}{v - 0.00191} - \frac{273.15 * 0.00366}{v - 0.00191} + \frac{0.00717}{v^2} = 0 \quad (20)$$

As shown below the complete script to construct the nomogram for the Van Der Waals chemical equation. Moreover, introducing the appropriate changes, that script can be reused to construct the nomogram for other formulas that could satisfy Eq. (17). Consequently, this reusable script is available in the Zenodo repository [12] to help to obtain any Type 10 nomogram through PyNomo software.

Type 10 nomogram code. Van Der Waals chemical equation:

```

""" Van_Der_Waals.py """
from pynomo.nomographer import *

p_params_1={
    'u_min':0.0,
    'u_max':50.0,
    'function':lambda u:u, →  $f_1(u_1) = f_1(p) = p$ 
    'title':r'$p$',
    'tick_levels':3,
    'tick_text_levels':2,
    'tick_side':'left'
}

t_params_2={
    'u_min':-100.0,
    'u_max':100.0,
    'function':lambda u:-u, →  $f_2(u_2) = f_2(t) = -t$ 
    'title':r'$t$',
    'tick_levels':2,
    'tick_text_levels':1,
    'tick_side':'right',
}

v_params_3={
    'u_min':0.0024,
    'u_max':0.1,
    'function_3':lambda u:(0.00366/(u-0.00191)),
    'function_4':lambda u:-((273.15*0.00366)/(u-0.00191))+0.00717/u**2,
    'title':r'$v$',
    'tick_levels':3,
    'tick_text_levels':2,
    'title_draw_center':True,
    'title_opposite_tick':False,
}

block_1_params={
    'block_type':'type_10',
    'width':10.0,
    'height':10.0,
    'f1_params':p_params_1,
    'f2_params':t_params_2,
    'f3_params':v_params_3,
    'isopleth_values':[[40,50,'x']],
}

main_params={
    'filename':'Van_Der_Waals.pdf',
    'paper_height':15.0,
    'paper_width':20.0,
    'block_params':[block_1_params],
    'transformations':[('rotate',0.01),('scale paper',)],
    'title_str':r'$p+\frac{a}{v^2}*(v-b)=R*(t+273.15)$'
}

Nomographer(main_params)

```

$$f_3(u_3) = f_3(v) = \frac{0.00366}{v - 0.00191}$$

$$f_4(u_3) = f_4(v) = -\frac{273.15 * 0.00366}{v - 0.00191} + \frac{0.00717}{v^2}$$

Ethics Statements

Authors state that the data supporting their paper meet the data requirements of the journal, ensuring that they have written a original work, not submitting this paper for consideration in another journal. Authors, also, declare not conflict of interest, or any financial of this work. Moreover, authors declares not potential ethical issues in the paper or during its writing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Please declare any financial interests/personal relationships which may be considered as potential competing interests here.

Data Availability

Type 10 nomogram code for Pynomo software. Van Der Waals equation (Original data) (Zenodo).

Type 9 nomogram code for Pynomo software. Projectile trajectory formula (Original data) (Zenodo).

Type 8 nomogram code for Pynomo software. Pressure unit wheel conversion (Original data) (Zenodo).

Type 1/Type 8 combined nomogram code for Pynomo software. Seismic magnitude of earthquakes (Original data) (Zenodo).

Type 7 reciprocal nomogram code for Pynomo software (Original data) (Zenodo).

Type 2 nomogram code for Pynomo software. Index of refraction formula (Original data) (Zenodo).

Type 1 nomogram code for Pynomo software. Slurry flow formula (Original data) (Zenodo).

CRedit Author Statement

Pedro Martínez-Pagán: Conceptualization, Data curation, Methodology, Writing – original draft; **Leif Roschier:** Conceptualization, Methodology, Software, Validation, Writing – review & editing.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Supplementary Materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.dib.2022.108661](https://doi.org/10.1016/j.dib.2022.108661).

References

- [1] L. Roschier, PyNomo 0.3.2 (2022). Retrieved from <http://lefakkomies.github.io/pynomo-doc/installation/installation.html>. Accessed March 15, 2022. doi:[10.5281/zenodo.6060368](https://doi.org/10.5281/zenodo.6060368).
- [2] D.P. Adams, in: *Nomography*, Archon Books, Connecticut, 1964, p. 198.
- [3] P. Martínez-Pagán, L. Roschier, Type 1 nomogram code for Pynomo software. Slurry flow formula, (2022). doi:[10.5281/zenodo.6060368](https://doi.org/10.5281/zenodo.6060368).
- [4] P. Martínez-Pagán, L. Roschier, Type 2 nomogram code for Pynomo software. Index of refraction formula, (2022). doi:[10.5281/zenodo.6059117](https://doi.org/10.5281/zenodo.6059117).
- [5] P. Martínez-Pagán, L. Roschier, Type 7 reciprocal nomogram code for Pynomo software, (2022). doi:[10.5281/zenodo.6053187](https://doi.org/10.5281/zenodo.6053187).
- [6] R.F. Mereu, A study of the relation between ML, Me, Mw, apparent stress, and fault aspect ratio, *Phys. Earth Planet. Inter.* 298 (2020), doi:[10.1016/j.pepi.2019.106278](https://doi.org/10.1016/j.pepi.2019.106278).

- [7] R. Doerfler, Creating nomograms with the PyNomo software, 2009. Retrieved from <http://www.myreckonings.com/wordpress>. Accessed January 19, 2022.
- [8] P. Martínez-Pagán, L. Roschier, Type 1/Type 8 combined nomogram code for Pynomo software. Seismic magnitude of earthquakes, (2022). doi:[10.5281/zenodo.6167730](https://doi.org/10.5281/zenodo.6167730).
- [9] P. Martínez-Pagán, L. Roschier, Type 8 nomogram code for Pynomo software. Pressure unit wheel conversion, (2022). doi:[10.5281/zenodo.6319907](https://doi.org/10.5281/zenodo.6319907).
- [10] R.P. Hoelscher, J.N. Arnold, S.H. Pierce, in: *Graphic Aids in Engineering Computation*, McGraw-Hill Book Company, New York, 1952, p. 216.
- [11] P. Martínez-Pagán, L. Roschier, Type 9 nomogram code for Pynomo software. Projectile trajectory formula, (2022). doi:[10.5281/zenodo.6320126](https://doi.org/10.5281/zenodo.6320126).
- [12] P. Martínez-Pagán, L. Roschier, Type 10 nomogram code for Pynomo software. Van Der Waals equation, (2022). doi:[10.5281/zenodo.6330876](https://doi.org/10.5281/zenodo.6330876).