

[Home \(/\)](#) / [Book \(/book/\)](#) / [Errata](#)

Errata

Corrections and clarifications to the text of the printed version of the book appear here. Special thanks to Stafford Baines, Stuart Anderson and Matthew Gillman for reporting so many of these.

p. 10 (Example E2.1) The second note ① is a bit careless: `int` truncates a floating point number towards zero rather than "rounding down" in casting it to an integer. For example, `int(-1.4)` is `-1`, not `-2`.

p. 17 (Table 2.4) The list of Python 3 keywords is incomplete: in addition to those given, `False`, `True`, `None`, `as` and `with` are reserved keywords. The identifier `print` is no longer a keyword in Python 3 but it does refer to the `print` function and so should be avoided as a variable name.

p. 34 (E2.13) There is no space after `Fortran` in the string `b`, so this word will not get an exclamation mark after it as a result of `c = b.replace(' ', '!\n')`. This is correctly shown when the string literal `c` is echoed back at the command prompt, but the `print(c)` call following that incorrectly shows `Fortran!` as the fourth line of output.

p. 41 (P2.3.4) The NASA page of planet symbols has moved to <https://solarsystem.nasa.gov/resources/680/solar-system-symbols/> (<https://solarsystem.nasa.gov/resources/680/solar-system-symbols/>).

p.55 (Q2.4.1) Some stray markup made its way into (d) which should read `print(*a, sep='')`

p. 56 (P2.4.7) The file defining the list of protein lengths to use in the second part of this exercise on Benford's Law is called `protein_lengths.py`, not `ex2-4_e_ii_protein_lengths.py`. It can be downloaded from the online version of the exercise. (</book/chapter-2-the-core-python-language-i/questions/problems/p24/benfords-law/>).

p. 57 (end of page) `if ... elif ... endif` should read `if ... elif ... else`. There is no `endif` in Python.

p. 76 (E2.28) There is a line of output missing between the two "in global scope" lines:

```
in global scope, b is 7 (id=264)
```

p. 76 (§2.7.4) "the it points to" → "the object it points to". This whole sentence should then read:

When a name is passed to a function, the "value" that is passed is, in fact, the object it points to.

p. 89 (P3.1.3) There is a missing minus sign in the exponential defining the Gaussian function, which should read

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right).$$

[Thanks to Ivan Yeung for spotting this typo].

p. 90 The voltages given are the rms values and should be 230 V (UK) and 120 V (US); to get the peak-to-peak values for the plot, they should be multiplied by $\sqrt{2}$. In the listing, instead of:

```
Vp_uk, Vp_us = 230, 110
```

use

```
Vp_uk, Vp_us = 230 * pylab.sqrt(2), 120 * pylab.sqrt(2)
```

(or `np.sqrt(2)` if using NumPy directly).

[Thanks to Toshio Iguchi for spotting this mistake].

p. 91 The URL in footnote 2 should be https://matplotlib.org/api/markers_api.html (https://matplotlib.org/api/markers_api.html).

p. 110 (P4.1.2) The code lines are missing their closing parantheses and should read:

```
>>> print(str_vector([-2, 3.5]))
>>> print(str_vector((4, 0.5, -2)))
```

p. 151 The last format specifier in the `print` statement is incorrect. This line should read

```
print('The balance of account number {:d} is {:s}{:.2f}'
      .format(self.account_number, self.currency, self.balance))
```

p. 207 The line

```
b=np.array([[0, 3, -2], [7, 1, 3], [4, 0, -1]])
```

should be

```
b = np.array([[0, 3, -2], [7, 1, 3], [4, 0, -1]])
```

p. 216 (P6.1.3) There is a missing minus sign in the exponential defining the Gaussian function, which should read

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

[Thanks to Ivan Yeung for spotting this typo].

p. 218 (E6.6) The "missing data" entries in the blood pressure column should be -/- instead of - for Listing 6.4 on the following page to work properly. [Thanks to Stafford Baines for spotting this].

p. 222(E6.7) The text file of data, `stroop.txt`, can be downloaded from <https://scipython.com/eg/aaj> (<https://scipython.com/eg/aaj>).

p. 232 (P6.3.2) `np.hist` should be `np.histogram`

p. 245 (P6.4.1) The exponent of time in the equation for $R(t)$ should be $+\frac{2}{5}$: $R(t) = CE^{\frac{1}{5}}\rho_{\text{air}}^{-\frac{1}{5}}t^{\frac{2}{5}}$

p. 247 (P6.4.4) The constant, $R = 8.314 \text{ J K}^{-1} \text{ mol}^{-1}$, should have been defined for this problem.

p. 262 (P6.6.3) The classification of conic sections in terms of its matrix representation should read as follows:

- If $\det \mathbf{Q} = 0$, the conic is *degenerate* in one of the following forms:
 - if $\det \mathbf{Q}_{33} < 0$, the equation represents two intersecting lines,
 - if $\det \mathbf{Q}_{33} = 0$, the equation represents two parallel lines,
 - if $\det \mathbf{Q}_{33} > 0$, the equation represents a single point.
- If $\det \mathbf{Q} \neq 0$:
 - if $\det \mathbf{Q}_{33} < 0$ the conic is a *hyperbola*,
 - if $\det \mathbf{Q}_{33} = 0$ the conic is a *parabola*,
 - if $\det \mathbf{Q}_{33} > 0$, the conic is an *ellipse*:
 - if $A = C$ and $B = 0$, the ellipse is a *circle*.

[Thanks to Stuart Anderson for spotting this.]

p. 275 The first two lines of code at the bottom of this page:

```
In [x]: spec = 2/n * np.abs(F[:n/2])
In [x]: pylab.plot(freq[:n/2], spec, 'k')
```

should use integer division in the array slice:

```
In [x]: spec = 2/n * np.abs(F[:n//2])
In [x]: pylab.plot(freq[:n//2], spec, 'k')
```

p. 279 (P6.8.3) The line of code:

```
sample_rate, wav = wavfile.read(\emph{<filename>})
```

should read simply

```
sample_rate, wav = wavfile.read(<filename>)
```

where <filename> is the name of the .wav file to read in.

p. 281 (§7.1.1) The code line:

```
line_quad, = ax.plot(x, x**2 / 2)
```

should read:

```
line_quad, = ax.plot(x, 1 + x**2 / 2)
```

p. 293 (Paragraph under "Error bars" subheading): The `pyplot` function referred to is called `errorbar`, not `errorbars`.

p. 293 The three NumPy arrays, `x`, `y` and `yerr` should be created by calling `np.array` rather than simply `array` if NumPy has been imported as `np`, as recommended.

p. 297 (E7.8) The data should be loaded into an array with `dtype='f8'` (ie double precision floating point) to avoid a deprecation warning (earlier NumPy versions) or error (later NumPy versions) upon division by 1000 in the conversion from GWh to TWh. That is,

```
data = np.loadtxt('germany-energy-sources.txt', skiprows=2, dtype='i4')
```

should be

```
data = np.loadtxt('germany-energy-sources.txt', skiprows=2, dtype='f8')
```

[Thanks to Joseph Karpinski for pointing this out].

Some other recent changes to Matplotlib mean that the hatching looks better if the specifiers

```
hatch = ['o', '', 'xxxx', '/']
```

are replaced by

```
hatch = ['oo', '', 'xxxx', '///']
```

and the bar `edgecolor` is set explicitly in the following line:

```
bars[i] = ax.bar(years, bottom=bottom, height=data[:,i+1], color='w',  
                 hatch=hatch[i], align='center', edgecolor='k')
```

p. 302 (Listing 7.10) The diffusion equation is not implemented exactly as given in Example E7.10 (the factor of $1/4\pi$ should be $1/\sqrt{4\pi}$). That is, the code line

```
theta = theta0 + H/cp/A/np.sqrt(D*t) / 4/np.pi * np.exp(-x**2/4/D/t)
```

should be

```
theta = theta0 + H/cp/A/np.sqrt(D*t * 4*np.pi) * np.exp(-x**2/4/D/t)
```

This does not alter the output qualitatively, but has been corrected in the online example (</resources/examples/the-one-dimensional-diffusion-equation/>). [Thanks to Naresh Gurbuxani for spotting this error].

p. 305 The URL in footnote 9 should be https://matplotlib.org/api/text_api.html (https://matplotlib.org/api/text_api.html).

p. 310 (Listing 7.15) The `axisbg` argument to `Figure.add_subplot` was deprecated in Matplotlib version 2.0 and replaced with `facecolor`. The line

```
ax = fig.add_subplot(111, axisbg='k')
```

should therefore be replaced with

```
ax = fig.add_subplot(111, facecolor='k')
```

if using more recent versions of the Matplotlib library.

p. 312 (E7.16) The URL for the journal article by Heinz *et al.* no longer works, but it can be found at <https://doi.org/10.1080/10691898.2003.11910711> (<https://doi.org/10.1080/10691898.2003.11910711>). The necessary data file has also disappeared from the publisher's website, but can be downloaded here as [body.dat.txt](/static/media/problems/P7.2/body.dat.txt) (</static/media/problems/P7.2/body.dat.txt>).

[Thanks to Joseph Karpinski for noticing this].

p. 314 (Listing 7.17) The `axisbg` argument to `Figure.add_subplot` was deprecated in Matplotlib version 2.0 and replaced with `facecolor`. The line

```
ax = fig.add_subplot(111, axisbg='k', aspect='equal')
```

should therefore be replaced with

```
ax = fig.add_subplot(111, facecolor='k', aspect='equal')
```

if using more recent versions of the Matplotlib library.

p. 316 (P7.1.5) Some readers may find it helpful to have the values of the following physical constants provided in this question:

```
# Physical constants in SI units: Planck's constant (J.s),
# the speed of light (m.s-1), Boltzmann's constant (J.K-1)
h, c, kB = 6.62606957e-34, 299792458, 1.3806488e-23
```

p. 317 The URL in footnote 14 no longer works. A suitable alternative is <https://matplotlib.org/tutorials/colors/colormaps.html> (<https://matplotlib.org/tutorials/colors/colormaps.html>).

p. 318 (Listing 7.18) Following a change to the Matplotlib API in v1.5.1 (<https://github.com/dfm/corner.py/issues/65>), the levels provided to `Axes.contour` must be in increasing order. To make the code in Example E7.18 work under versions v1.5.1+, the following line:

```
levels = list(-levels) + list(levels)
```

should be changed to:

```
levels = sorted(list(-levels) + list(levels))
```

p. 320 (and Listing 7.20) In recent versions of Matplotlib, `interpolation='nearest'` is the default and `interpolation='bilinear'` now has to be explicitly set in the call to `plt.imshow` if required. The online example (<https://scipython.com/book/chapter-7-matplotlib/examples/visualizing-a-matrix-with-imshow/>) has been updated to accommodate this change.

p. 322 (Listing 7.21) In recent versions of NumPy, indexing with non-integers is now an error. Replace the line:

```
ix, iy = width / 2 + x * width / 10, y * height / 12
```

with:

```
ix, iy = int(width / 2 + x * width / 10), int(y * height / 12)
```

p. 325 (Listing 7.23) The array `u` should be initialized to `u0`, not left empty, since we want to keep the boundary at T_{cool} . The code line:

```
u = np.empty((nx, ny))
```

should be:

```
u = u0.copy()
```

Also, the code line:

```
dt = dx2 * dx2 / (2 * D * (dx2 + dy2))
```

should be

```
dt = dx2 * dy2 / (2 * D * (dx2 + dy2))
```

(though it doesn't make any difference for the identical values of dx and dy chosen in this example).

p.346 (Example E8.7) `scipy.special.ellipse` (<https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.ellipse.html>) actually evaluates

$$E(m) = \int_0^{\pi/2} \sqrt{1 - m \sin^2 \theta} d\theta,$$

as given correctly on the previous page. Therefore, for the orbital perimeter we need

```
In [x]: pe = 4 * a * ellipse(e*e)
In [x]: print(pe)
939887967.974
```

and the difference from the circular orbit,

```
(pc - pe) / pe * 100
0.0067215663638305143
```

is only 0.0067% instead of 0.42%.

p.347 (Example E8.9) The expression for the normalized Gaussian profile is missing a square in the exponent. It should be:

$$G(x; \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

The accompanying code is correct. [Thanks to Chen Ying for spotting this].

p. 355 (P8.1.5) The Theis equation should read

$$s(r, t) = H_0 - H(r, t) = \frac{Q}{4\pi T} W(u)$$

(4π in the denominator, not 2π).

p. 359 The URL in footnote 7 should be <https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.quad.html> (<https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.quad.html>).

p. 370 (Q8.2.2.e.) Compare this integral with the value of $2\pi I_0(z)$, not $I_0(z)/2\pi$.

p.371 (P8.2.4: The Brusselator) The second of the scaled differential equations should read

$$\frac{dy}{dt} = bx - x^2y$$

(note the minus sign).

p. 384 (Definitions of the Jacobian and Hessian functions) The definitions for functions `dx` and `ddx` do not have the correct syntax; they should be

```
def df(x):
```

and

```
def ddf(x):
```

respectively. [Thanks to Alex Hogg for spotting this].

p. 391 (Demonstration of the `leastsq` function in Section 8.4.2) The variable `pfit` is never assigned. It should hold the list of best-fit parameters, returned as `plsq[0]` and output earlier on the page. Use:

```
pfit = plsq[0]
pylab.plot(t, f(t, *pfit), c='k', label='Fit')
```

[Thanks to Javier Elexpuru for spotting this].

p. 394 (Example E8.22) The function `jac` should simply return the derivatives,

```
return -da, -de
```

(i.e. remove the following line, which is never executed, `return np.array((-da, -de)).T`.)

p. 396 (§8.4.3) The bracketing interval $[a, b]$ should be such that $\text{sgn}[f(a)] = -\text{sgn}[f(b)]$. That is, $f(a)$ and $f(b)$ should bracket the root and have opposite signs. It is not necessary, of course, that $f(a) = -f(b)$.

p. 397 (Caption to Figure 8.23): The function plotted is $f(x) = \frac{1}{5} + x \cos(3/x)$ (NB not \sin). [Thanks to Alex Hogg for pointing this out].

p. 400 (Q8.4.3): The cosine in the formula for z in this question should be squared:

$$z = x \tan \theta_0 - \frac{g}{2v_0^2 \cos^2 \theta_0} x^2.$$

p. 433 (Q6.1.5e): The description of this slicing operation should be: "in the third block, for each row take (backwards) the items in all but the first column".

Minor typos

p. 64 (P2.5.5): the tomorrow's date → tomorrow's date

p. 65 (P2.5.10): Consider region → Consider the region

p. 66 (§2.6): to acheive → to achieve

- p. 78 (E2.29): there is a stray close parenthesis, ")", after $f(2) = 4$.
- p. 103 (§4.1.2) an syntactically → a syntactically
- p. 112 (§4.2.2) keys to to be copied → keys to be copied
- p. 122 (§4.3.1) L9: acheived → achieved
- p. 125 (E4.10) is that if these functions → is that these functions
- p. 127 an loop → a loop
- p. 130 (Footnote) S. Wofram → S. Wolfram
- p. 136 (P4.4.2) The lower case r in the haversine formula should be R .
- p. 137 (P4.4.2) considered as a a sphere → considered as a sphere
- p. 150 (Listing 4.6) A abstract base class → An abstract base class
- p. 151 (E4.16) described in class diagram → described in the class diagram
- p. 155 (Listing 4.7, comment) Update or center of mass sum → Update our center of mass sum
- p. 186 (Penultimate line) "often often" → "often"
- p. 189 (Table 6.2) "floatng" → "floating"
- p. 190 (Final line) "respecively" → "respectively"
- p. 194 "which, tries to return" → "which tries to return"
- p. 194 (Code comment) "create and independent" → "create an independent"
- p. 208: "takes a, *sorted* array, a and" → "takes a *sorted* array, a , and"
- p. 212 "respecively" → "respectively"
- p. 217 "picked off an assigned" → "picked off and assigned"
- p. 233: "define a the object" → "define the object"
- p. 239: in Figure 6.7 the caption should read $n = 0, 1, 2, 3, 4$ instead of $x = 0, 1, 2, 3, 4$
- p. 255 (P6.5.3): "**MM*** and **MM*** " → "**MM*** and **M*****M**"
- p. 269: "Poison probability distribution" → "Poisson probability distribution"
- p. 269: "an one-dimensional sequence" → "a one-dimensional sequence"
- p. 272 (P6.7.3) L5: "process known as chemotaxis" → "a process known as chemotaxis"
- p. 278 (Figure 6.16): "applied its" → "applied to its"
- p. 286: "you may wish to pass to is `legend` " → "you may wish to pass to `legend()` " p. 296 (Table 7.9): The explanation of the argument `height` appears twice in this table.
- p. 338 (Figure 8.1 caption): "gravitationa" → "gravitational"
- p. 339 (Last bullet point): `jn_zeros(n, nt)` appears twice.
- p. 341 (Example E8.4) There should be no period after "1953".
- p. 345 (Footnote) "instead m " → "instead of m "

p. 366 (top of the page): $\frac{dx_2}{dt_2}$ should be $\frac{dx_2}{dt}$

p. 368 "Drag consant" → "Drag constant"

p. 390 (§8.4.2) The function to be fit is $f(t) = Ae^{-t/\tau} \cos 2\pi\nu t$, so τ is positive.

p. 447 (Index): `plt.errorbars` → `plt.errorbar`
