Visualization and Error Analysis of Integrated Magnetic Field Lines from Magnetic Dipole and Magnetic Bottle

Ansh R. Gupta [1]

[1] *Steward Observatory, University of Arizona, 933 N Cherry Avenue, Tucson, AZ 85721, USA*

ABSTRACT

The magnetic field generated by a current distribution can be visualized by considering its corresponding field lines. In this work, a Runge–Kutta–Fehlberg integration method with an adaptive step size is used to visualize the magnetic field lines from a current loop and magnetic bottle. Further, an RK4 integration is carried out on a sample point from the magnetic bottle to analyze and characterize self-convergence and convergence order of the solution. Additionally, error for this integration scheme is estimated by constructing a higher order approximation to the solution using Richardson extrapolation. Visualizations qualitatively match the expected field lines from the magnetic field configurations and the RK4 solution is demonstrated to converge at fourth order as analytically expected. Error estimates from the extrapolation for each coordinate are on the order of 10^{-11}, affirming the accuracy of the solutions.

INTRODUCTION

Moving electric charges generate magnetic fields. To determine the magnitude and direction of the magnetic field at a given point from a current distribution, the Biot-Savart Law can be used. However, the integration required to analytically determine the field can be tedious and difficult (Griffiths 2005). A numerical approximation for the field may be used instead.

In this work, two magnetic field distributions are considered. The first is a simple dipole consisting of a loop of current in the x-y plane. The second is a "magnetic bottle", which consists of four current loops in the x-z plane. The components of the magnetic field from a loop of current in the x-y plane can be approximated by applying equations (1-3), where a is the radius of the current loop and I_0 is

the magnitude of the current. To determine the field from loops in the x-z plane, the same equations can be used with a variable change $y \rightarrow z$ and $z \rightarrow y$.

$$B_x(a, I_0, x, y, z) = \frac{3\pi a^2 I_0 xz(43a^4 + a^2(51(x^2+y^2) + 16z^2) + 8(x^2+y^2+z^2)^2)}{8(a^2+x^2+y^2+z^2)^{9/2}} \tag{1}$$

$$B_y(a, I_0, x, y, z) = \frac{3\pi a^2 I_0 yz(43a^4 + a^2(51(x^2+y^2) + 16z^2) + 8(x^2+y^2+z^2)^2)}{8(a^2+x^2+y^2+z^2)^{9/2}} \tag{2}$$

$$B_z(a, I_0, x, y, z) = \frac{\pi a^2 I_0(46a^6 + a^4(9(x^2+y^2) + 78z^2) + 3a^2(36z^2(x^2+y^2) - 15(x^2+y^2)^2 + 16z^4) - 8(x^2+y^2+z^2)^2}{8(a^2+x^2+y^2+z^2)^{9/2}} \tag{3}$$

Magnetic field lines follow the field vectors. To visualize the field lines, initial "seed points" can be chosen as a set of initial conditions. Magnetic fields have no source or sink, so if the position of a magnetic field line is integrated forward from an initial condition, a closed loop will form. This is easiest to accomplish by parameterizing the field line in terms of a time variable, t. Then the components of the position vector of the field lines, $x(t)$, $y(t)$, and $z(t)$ are the solutions to the following system of first order differential equations (ODEs).

$$\begin{aligned} \frac{dx}{dt} &= B_x \\ \frac{dy}{dt} &= B_y \\ \frac{dz}{dt} &= B_z \end{aligned} \tag{4}$$

To carry out this integration, the Runge–Kutta–Fehlberg method is used. This method utilizes an embedded pair of fourth and fifth order approximations. An embedded pair is a set of numerical approximations which use, at least in part, the same intermediate steps to minimize the total computational cost of the operation. With these two approximations, one of higher order, the truncation error of the lower order approximation can be estimated. By using a smaller step size when this error is larger and a larger step size when the error is low, the computational time can be concentrated

in regions of the integration where the numerical error is the largest. In practice, this can allow for numerically accurate integration to be carried out quickly and efficiently. The exact numerical coefficients used in the embedded pair are drawn from previous work (Fehlberg 1969).

This integration scheme was implemented in Python 3.10.4 and executed within a Jupyter Notebook (6.4.11). The numpy and matplotlib packages were used for numerical and plotting purposes.

VISUALIZATION

Magnetic Dipole

The first visualization considered the magnetic field from a single current loop, which is a simple magnetic dipole. The first set of initial conditions consisted of points located on two circles of radius 3 at $z = 2$ and $z = 5$. Each circle was discretized into 24 seed points by evenly distributing points azimuthally. The results are shown in a 3 dimensional plot in Figure 1.

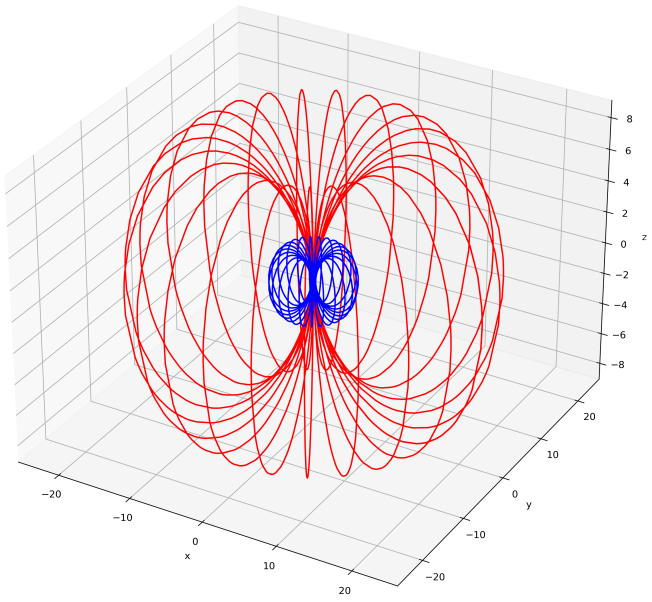

Figure 1. 3 dimensional plot showing magnetic field lines from a magnetic dipole. The field lines in blue show the integration of the first set of initial conditions at $z = 2$, while the lines in red show the integration of the second set ($z = 5$).

The second visualization looked at the same dipole, but used a set of seed points from two lines of length 3 (centered at $x = 0$) parallel to the x-axis. Both lines were at $y = 0$, with the first set to $z = 2$ and the second to $z = 5$. By inspecting equation (2), it can be seen that the y-component will remain 0 and the field lines will lie completely within the x-z plane. A 2 dimensional plot of the results of this integration are shown in Figure 2.

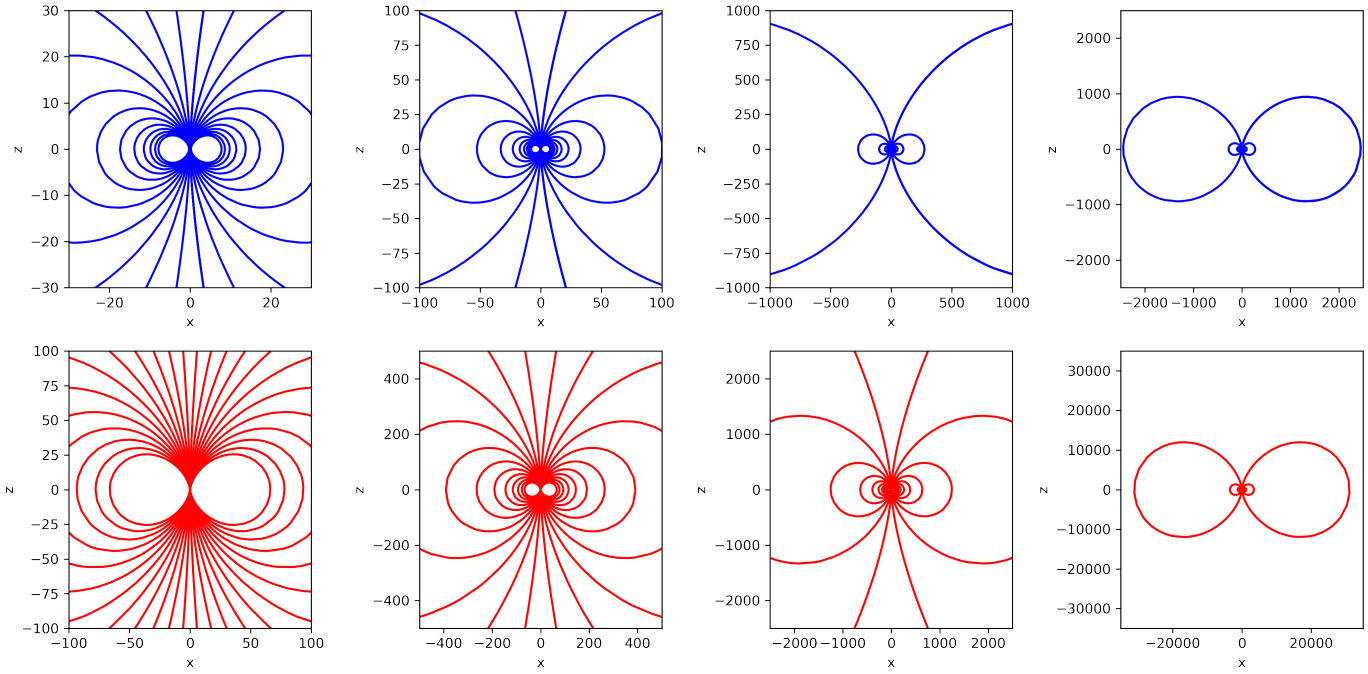

Figure 2. 2 dimensional plot showing magnetic field lines from a magnetic dipole at varying scales. The top row, in blue, shows the results of the integration of the first set of initial conditions ($z = 2$), while the second, in red, shows the second set ($z = 5$).

To ensure that the magnetic field loops properly closed, the distance to the initial seed point was measured at each step along the integration. If the distance is within a certain small value, ϵ, and smaller than the previous step, then it can be assumed that the loop is converging towards the start point. Whenever this condition is triggered, the integration is halted and the results returned.

Magnetic Bottle

A magnetic bottle consists of four separate current loops located at different positions. The total field of the bottle can be found by taking the superposition of the fields from each of the loops.

Specifically, the bottle in this case has symmetry along the y-axis, so the loops lie along the x-z plane. The field for each of these loops can be found using equations (1-3) and making the variable change $y \rightarrow z$ and $z \rightarrow y$ as previously described. In this example, the four loops have parameters as shown in Table 1.

Loop Number	a	I_0	y
1	3	10	15
2	30	2	5
3	30	2	-5
4	3	10	-15

Table 1. Parameters for component loops of magnetic bottle.

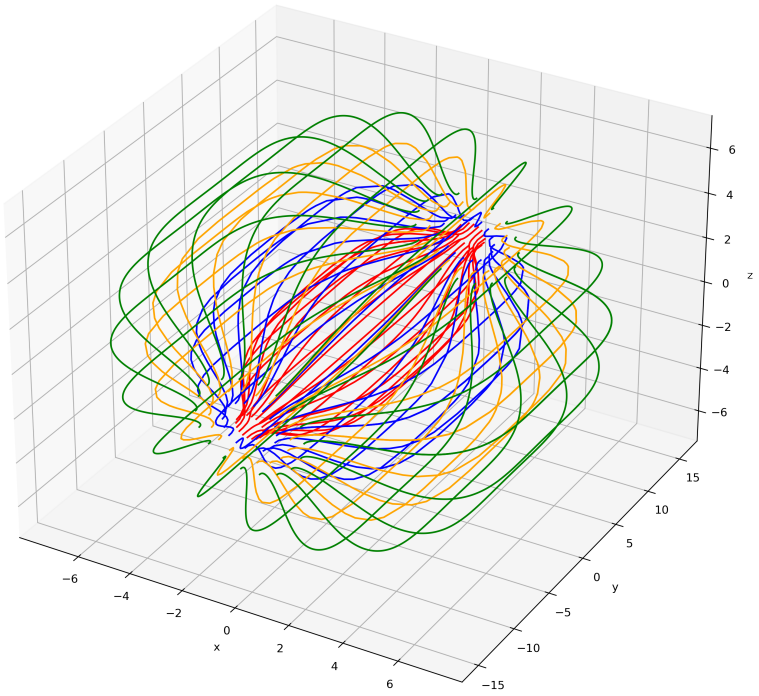

Figure 3. 3 dimensional plot showing magnetic field lines from a magnetic bottle. The field lines in red, blue, orange, and green correspond with the set of initial conditions with the circle radius of 0.5, 1, 1.5, and 2 units, respectively.

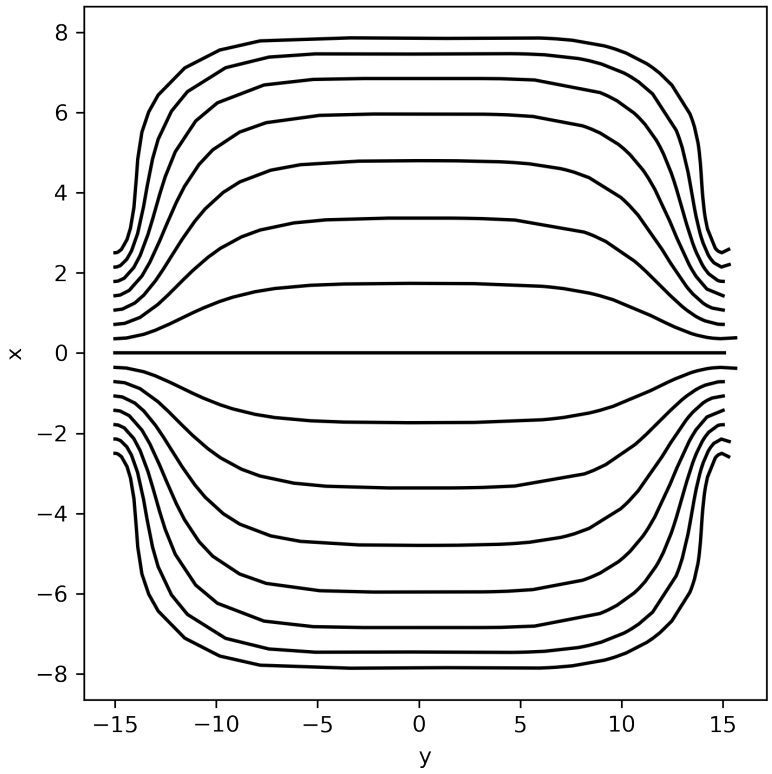

Figure 4. 2 dimensional plot showing magnetic field lines from a magnetic bottle with the set of initial conditions from the line parallel to the x-axis.

The first visualization of the magnetic bottle field considered lines passing through seed points on circles parallel to the x-z plane with radius 0.5, 1, 1.5, and 2. These initial conditions were all set to $y = -15$ and integrated through to $y = 15$. Each circle was discretized into 19 seed points by evenly distributing points azimuthally. The results are shown in a 3 dimensional plot in Figure 3.

The second visualization of the magnetic bottle used a set of seed points from a line of length 5 parallel to the x-axis (centered at $x = 0$) with $y = -15$ and $z = 0$. The line was discretized into 9 sets of initial conditions, and the results are shown in a 2 dimensional plot in Figure 4. Similarly to the 2 dimensional case in the dipole, this set of initial conditions generates field lines solely in the x-y plane.

SELF-CONVERGENCE

As mentioned in the introduction, finding an analytical solution to the system of first order ODEs can be challenging, so there is no way to show convergence to a "real" solution. However, by using various step sizes, the convergence of the numerical solution to a specific value for finer and finer subdivisions can be quantified. This process can be used to find the order of convergence of the solution, which can be compared to the theoretical order for an ODE solving algorithm. Because this method requires a constant step size, from this point onwards a standard RK4 method was used instead of an embedded RK45 pair with adaptive step size.

The standard RK4 method has a convergence order of $O(h^4)$ (Landau et al. 2007). This means that the method approximates the numerical solution exactly up to the fourth order, with a fifth order truncation error term.

$$y_{numerical}(h) = y_{exact}(h) + Ch^4 + O(h^5) \tag{5}$$

Ignoring the truncation error term, consider the difference in approximations between two separate values of the step size, h_1 and h_2, and a third, h_3.

$$y_{numerical}(h1) - y_{numerical}(h3) = C(h_1^4 - h_3^4) \tag{6}$$

$$y_{numerical}(h2) - y_{numerical}(h3) = C(h_2^4 - h_3^4) \tag{7}$$

Taking the ratio of equations (6) and (7) then eliminates the constant term in the fourth order error term, C, leaving the right hand side in terms of known values at the analytic rate of convergence. Considering a variable $h1 \equiv h$ and a fixed $h2 = 2h3$ then reduces the right hand side into simple terms by diving through by $\frac{h3}{h3}$.

$$\frac{y_{numerical}(h) - y_{numerical}(h3)}{y_{numerical}(h) - y_{numerical}(h3)} = \frac{h^4 - h_3^4}{h_2^4 - h_3^4} \tag{8}$$

$$\frac{y_{numerical}(h) - y_{numerical}(h3)}{y_{numerical}(h) - y_{numerical}(h3)} = \frac{(h/h_3)^4 - 1}{2^4 - 1} \tag{9}$$

To test self convergence, an arbitrary initial seed point (the third discretized point from the circle with r=1.5 on the x-z plane) was chosen from the 3 dimensional magnetic bottle visualization. A

maximum integration time of 1 unit was chosen. The total number of steps, N, was varied in intervals of 100 from $N = 300$ to $N = 700$ to determine the step sizes h to be used. Then h2 was calculated with a total of 800 steps and h3 with 1600 steps. After carrying out the integration from the given initial condition with each value of h, as well as h2 and h3, the values for each coordinate were inputted into equation (9). For each coordinate (x, y, z), a plot was made of the left hand side of the equation against the step size, and the right hand side was also shown in each plot at the given h values, shown in Figure 5. The value of equation (9) is here referred to as the "Convergence Value".

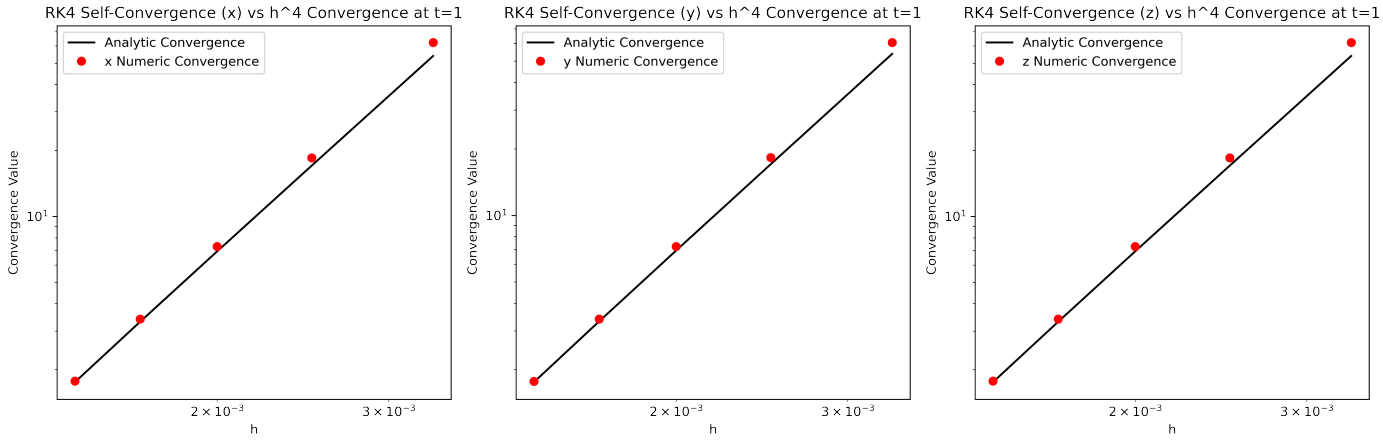

Figure 5. Plot of the "Convergence Value" against h for the solution from the single initial condition described in section 3 for each coordinate. The solid line indicates the analytical rate of convergence, with the red dots indicating the numerically determined values.

Figure 5 shows that the numerically determined "Convergence Values" for the RK4 method are approximately in line with the expected analytical convergence rate of n=4. Thus, the solution to the system of ODEs converges as expected, which validates the solutions obtained.

RICHARDSON EXTRAPOLATION

As mentioned in section 3, the standard RK4 method has a truncation error on the order of $O(h^5)$. To estimate the magnitude of this error, a higher order estimate of the solution can be made using a process known as Richardson extrapolation. Using two evaluations of the RK4 integration, one with half the step size of the error, the lowest order error term can be cancelled.

Using equation (5) with a step size h and another with h/2, the coefficient C which determines the lowest order error term can be found.

$$y_{numerical}(h) = y_{exact}(h) + Ch^4 + O(h^5) \tag{10}$$

$$y_{numerical}(h/2) = y_{exact}(h/2) + C(h/2)^4 + O(h^5) \tag{11}$$

If the truncation error term is ignored, then the exact analytic value is no longer exact, but an estimate with lowest error term $O(h^5)$. This new value can be called y_{new}. Then by rearranging to solve for C, (10) and (11) can be equated. Since y_{exact} doesn't depend on the step size, the second term in (10) and (11) are equal.

$$C(h)^4 = y_{numerical}(h) - y_{new} \tag{12}$$

$$C/16(h)^4 = y_{numerical}(h/2) - y_{new} \tag{13}$$

$$C(h)^4 = 16y_{numerical}(h/2) - 16y_{new} \tag{14}$$

$$y_{numerical}(h) - y_{new} = 16y_{numerical}(h/2) - 16y_{new} \tag{15}$$

$$15y_{new} = 16y_{numerical}(h/2) - y_{numerical}(h) \tag{16}$$

$$y_{new} = \frac{16}{15}y_{numerical}(h/2) - \frac{1}{15}y_{numerical}(h) \tag{17}$$

As demonstrated above, by taking a linear combination of an RK4 integration with step size h and h/2, a new approximation for the solution can be constructed with lowest error term $O(h^5)$. So the magnitude of the truncation error of the RK4 solution can be determined by taking the difference between this new, higher order estimate and the original solution.

The same initial conditions as in section 3 were used, with improved estimates made for each coordinate solution. The results of the Richardson extrapolation and estimated truncation error are summarized in Table 2.

Coordinate	Original Solution	Improved Estimate	Truncation Error Estimate
x	3.7671629437995704	3.7671629437800704	1.949995720451625e-11
y	-10.905907298924221	-10.905907298843692	8.052936095737095e-11
z	3.161025036726944	3.1610250367105737	1.6370460542702858e-11

Table 2. Original solution, Richardson extrapolation derived estimates, and estimated local truncation error for RK4 solution at single seed point.

CONCLUSION

The visualization of the magnetic field lines from a dipole and magnetic bottle appeared as qualitatively expected from their respective sources (Stöhr & Siegmann 2006). The order of convergence of the RK4 method for the 3 dimensional single seed point was approximately $O(h^4)$ as analytically expected. Furthermore, Richardson extrapolation was successfully employed to generated a higher order estimate for the solution from the same seed point, and error estimates were obtained. The results of this project demonstrate how relatively mundane applications of ODE solving methods can be used to compute and analyze problems that would be difficult to do so analytically. Future work could build upon this project by investigating the magnetic field lines of different current distributions, performing further Richardson extrapolation to quantify higher order error terms, or optimize the solution/solving method to increase the efficiency of the process.

Thank you Elisabeth Krause, Associate Professor at the University of Arizona Departments of Astronomy and Physics, and Associate Astronomer at Steward Observatory (https://www.as.arizona.edu/people/faculty/elisabeth-krause) for instructing the Fall 2022 semester PHYS 305 class, and for answering questions about and facilitating this project.

Software: Python (Van Rossum et al. 1995), NumPy (Van Der Walt et al. 2011), Matplotlib (Hunter 2007), SciPy (Virtanen et al. 2020), IPython (Pérez & Granger 2007), Jupyter Notebook (Kluyver et al. 2016)

REFERENCES

Fehlberg, E. 1969, Computing, 4, 93–106, doi: 10.1007/bf02234758

Griffiths, D. J. 2005, Introduction to electrodynamics, American Association of Physics Teachers

Hunter, J. D. 2007, Computing in science & engineering, 9, 90

Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. 2016, in IOS Press (IOS Press), 87–90, doi: 10.3233/978-1-61499-649-1-87

Landau, R. H., Páez, M. J., & Bordeianu, C. C. 2007, Computer Languages, 7, 2

Pérez, F., & Granger, B. E. 2007, Computing in science & engineering, 9, 21

Stöhr, J., & Siegmann, H. C. 2006, Solid-State Sciences. Springer, Berlin, Heidelberg, 5, 236

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, Computing in science & engineering, 13, 22

Van Rossum, G., Drake, F. L., et al. 1995, Python reference manual (Centrum voor Wiskunde en Informatica Amsterdam)

Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nature methods, 17, 261