

PAPERS | FEBRUARY 01 2025

Introducing key theoretical and data analysis tools in computational physics via Earth's temperature and climate

 **FREE**

David Syphers

 Check for updates

Am. J. Phys. 93, 150–156 (2025)

<https://doi.org/10.1119/5.0224293>



Articles You May Be Interested In

Modeling Hidden Circuits: An Authentic Research Experience in One Lab Period

Phys. Teach. (October 2016)

The pilot-wave perspective on quantum scattering and tunneling

Am. J. Phys. (April 2013)

Modeling the Motion of an Increasing Mass System

The Physics Teacher (April 2010)



Introducing key theoretical and data analysis tools in computational physics via Earth's temperature and climate

David Syphers^{a)}

Department of Chemistry, Biochemistry, and Physics, Eastern Washington University, Cheney,
Washington 99004

(Received 20 June 2024; accepted 4 November 2024)

This paper presents two thematically linked activities focused on temperature, climate, and climate change that can be used as engaging ways to introduce students to computational techniques. The first activity makes use of a non-equilibrium Earth undergoing climate change to introduce students to numerical solutions to differential equations via an ordinary first-order differential equation. This activity also introduces the concept of a toy model, and the important ideas of simulation validation and convergence. The second activity gives students several decades of local temperature data sampled hourly, introducing them to model fitting messy, real-world data, while also allowing them to see the effect of climate change. The amount of scaffolding for each activity is flexible, allowing instructors to adapt these activities to classes at advanced, intermediate, and even introductory levels. © 2025 Published under an exclusive license by American Association of Physics Teachers.

<https://doi.org/10.1119/5.0224293>

I. INTRODUCTION

Teaching computation prepares our undergraduate students not only for graduate school but also for private sector careers, many of which will be in fields related to computation, such as programming and data science.^{1,2} A perennial question facing teachers is how much to make our students do “by hand.” In computation, we have long made extensive use of programs written by others. This borrowing is efficient, but can lead to black boxes, where users do not know how the program actually works. The rise of machine learning, including artificial intelligence, has dramatically increased this black-box effect.

As physics teachers, we care about results, but we also care about derivations, because we care about understanding. We provide and ask for derivations in our analytical courses, and in traditional computation, we ask our students to write common algorithms prior to letting them use existing packages. There is somewhat less standardization when it comes to introducing students to data analysis, but it can be useful to have a broadly applicable and standardized approach here as well. Data and models about local temperature and climate change are well suited to this task, because they are topics about which students already have considerable intuition, while still providing enough complexity to thoroughly engage students.

The pair of activities I describe in the following, one in theory and one in data analysis, are linked only thematically and can be used separately, including with students at a wide variety of levels. My own target audience has been a broad group of undergraduate students who have all completed first-year physics but have little else in common. Many are physics majors, but others are majoring in math, chemistry, engineering, or other fields. About half of them have some prior programming experience, but the remaining half do not, and thus as a prerequisite to these activities, I spend a few weeks introducing them to Python. Any very high-level language suitable for scientific programming can be used, but Python stands out as one that is widely used in both scientific and corporate environments.

Adaptations can be made for students at different levels, as I discuss in Sec. IV. These activities can be used together

or separately. Jupyter Notebooks implementing these activities in Python, including full solutions, are provided as [supplementary material](#). These notebooks can be run locally on computers with Python and Jupyter installed (such as with the Anaconda distribution), or on a cloud-based service such as Google Colab.

II. TOY MODELS, AND DIFFERENTIAL EQUATIONS IN NON-EQUILIBRIUM SITUATIONS

Physics is stated in differential equations. While some situations are exactly solvable, other situations are not, particularly as students progress to more advanced and constrained versions of physical laws, such as Schrödinger’s equation or Einstein’s equations. Thus, we often resort to numerical solutions.

However, the equations of physics are almost always *second*-order equations, from the examples above to Maxwell’s equations, Lagrange’s equations, and even Newton’s second law. Computational approaches to second-order equations can require learning how to simultaneously solve two first-order equations and then conceptualizing one second-order equation as two first-order equations.³ While this is an important skill, it is a less-than-ideal instructional starting point.

However, there are situations that give rise to first-order equations, including temperature change resulting from an imbalance of energy flow. A salient example is climate change.

This exercise also reviews physics concepts and introduces the idea of a toy model. Importantly, for numerical simulations, it also introduces validation of programs by checking against known results, and the concept of a converged simulation.

I start with the standard *equilibrium* calculation of a planet’s temperature. Such calculations are commonly covered when training climate scientists^{4,5} and astronomers,⁶ but do not actually require any specialized knowledge. First-year physics understanding of radiative heat transfer, based on the Stefan–Boltzmann equation, is all that is needed.^{7,8}

A. Equilibrium model

Students begin by deriving the solar constant (the flux from the Sun at Earth's orbit), $f_{\odot} = L_{\odot}/(4\pi d_{ES}^2)$, in terms of the provided constants of the Sun's luminosity (power output; L_{\odot}), and the Earth–Sun distance (d_{ES}). Equating the power received from the Sun with the radiated power out results in

$$\pi r_E^2 f_{\odot} R_{\text{in}}^{\text{eff}} = \sigma(4\pi r_E^2) T^4 R_{\text{out}}^{\text{eff}}. \quad (1)$$

The fact that there is a 4 on the righthand side of that equation is usually obvious, since the entire area of the Earth radiates, but the *absence* of the 4 on the lefthand side is often tricky for students. Some students instinctively use a 4 there as well, for the area of a sphere, and need to be reminded that the Sun does not shine everywhere on Earth at the same time. Others use a factor of 2, accounting for there being day and night sides of the Earth, but this assumes the Sun shines from directly overhead all day. Those with more physics and math background may start integrating the incoming radiation dotted with dA over the surface of a half sphere, which is fine, but they should be subtly encouraged to consider the far easier and more intuitive approach of asking how big a target Earth makes for the Sun's light.

I do discuss the dependence on emissivity (ϵ) and transmission coefficient (here denoted by R , since T is taken by temperature), but since these factors only occur multiplied together in this simple model, I instruct the students to combine these degenerate variables into a single effective transmission coefficient R^{eff} . As a pedagogical matter, it is worth emphasizing that the emissivities at visible wavelengths (which is most of the incoming radiation) and in the IR (most of the outgoing radiation) are different, as are the atmospheric transmission coefficients. More advanced students can appreciate that we are taking averages over more complicated, frequency-dependent transmission coefficients,^{8,9} just as the Stefan–Boltzmann law averages blackbody radiation over all frequencies.

Solving Eq. (1) results in the equilibrium temperature,

$$T_{\text{eq}} = \left(\frac{f_{\odot} R_{\text{in}}^{\text{eff}}}{4\sigma R_{\text{out}}^{\text{eff}}} \right)^{1/4}. \quad (2)$$

The first program the students write applies this analytical model to calculate equilibrium temperature for given transmission coefficients. This allows them to confirm that the transmission coefficients I provide do give a realistic global average temperature. It also lets them play with a variety of other scenarios, including the (low!) temperature of an Earth with no atmosphere, or the (also low) temperature of a “snowball Earth” that’s quite reflective due to very large-scale glaciation, such as the Marinoan glaciation period around 650 million years ago.¹⁰

When the students write the non-equilibrium program, which does *not* incorporate Eq. (2), the first test they run with the program is to verify that it returns a stable temperature if they start with the correct equilibrium temperature for the given transmission coefficients. This introduces the concept of validating code by checking the results in known special cases. (Questions and solutions are shown in the climate model supplementary file.)

B. Non-equilibrium model

The non-equilibrium situation is introduced as a toy model for climate change. I have students employ a step-function decrease of $R_{\text{out}}^{\text{eff}}$ at five years into the simulation, a very simple model of increased outward opacity due to greenhouse gasses being introduced to the atmosphere. This abrupt change in transmission may seem unphysical, and I do primarily choose it for its simplicity, but note that sudden forcing can result from the climate system leaving quasi-stable equilibrium.⁴ The purpose of having this change five years into the simulation is to force another check that the non-equilibrium code can reproduce the initial equilibrium situation.

Students at this level are often unfamiliar with the term “toy model,” so it’s worth emphasizing the utility of a model that captures some effect of interest even if it is not intended to realistically match data to any degree of accuracy. This is a single-zone model, using one temperature to characterize the Earth, and thus assumes perfect heat transfer around the globe. While clearly not the planet we live on, the toy model gives an example of how temperature increases can be “locked in” even if all new emissions of greenhouse gasses immediately cease.¹¹

Reminding the students that heat capacity $C = Q/\Delta T$, and that $Q = P_{\text{net}}\Delta t$, they come to the relationship

$$\begin{aligned} \Delta T &= \frac{1}{C} P_{\text{net}} \Delta t \\ \Rightarrow T_{i+1} &= T_i + \frac{P_{\text{net}}}{C} \Delta t, \end{aligned} \quad (3)$$

where $T_i = T(t_i)$, the temperature in the i th time step. P_{net} is temperature dependent, as it is calculated from the difference between the energy received from the Sun (the lefthand side of Eq. (1)) and the energy radiated out by the Earth (the righthand side of Eq. (1)). The two sides of that equation are no longer equal due to the system no longer being in equilibrium.

It is worth explicitly pointing out to the students that Eq. (3) is the finite version of the differential equation,

$$\frac{dT}{dt} = \frac{P_{\text{net}}}{C}. \quad (4)$$

The precise values of the transmission coefficients and heat capacity do not matter in this toy model, but make sure whatever combination you provide results in a correct initial equilibrium temperature, and a long-term temperature rise that is significant but not absurd. Instructors who have more time can introduce physically motivated values for these parameters. The heat capacity calculation can be motivated by taking the volume of water in the Earth’s oceans,¹² about $1.3 \times 10^{18} \text{ m}^3$, and calculating its heat capacity of about $5.4 \times 10^{24} \text{ J/(kg}^{\circ}\text{C)}$. The value I use in the code in my supplementary files is an order of magnitude smaller than this, and a more involved discussion could delve into the approximations we are making in this single-zone model (which assumes perfect mixing, something that does not happen in our oceans).

Motivating the transmission coefficients would be a more significant digression, appropriate only for an advanced class.^{9,13} This could be worthwhile if the class has already covered numerical integration, in which case the students

could integrate over an empirical atmospheric transmission function. You will need full coverage from the near UV through the thermal IR. Since no single instrument can record these data, the best approach might be an empirically based radiative transfer model, such as MODTRAN⁶,¹⁴ or other codes making use of the absorption database HITRAN.¹⁵ Also note that you must weight the actual transmission function by the emission spectrum to calculate a single effective number.

I have the students calculate how long it takes to reach the new equilibrium, to within 0.1 K, and they see a temperature rise due to greenhouse gas emissions from decades earlier (Fig. 1). I also have the students vary their time step, Δt , and check their results. At this early stage in their computational careers, they are tempted to choose time steps that seem like round numbers to them, not always distinguishing the large difference that can exist between $\Delta t = 1$ (second) and $\Delta t = 1$ (year).

The meaning of “convergence” is that the model produces the solution we care about to the precision we need. Remind students that when solving a differential equation, the solution is a *function*, not a number, and in our case, we are looking for the temperature function $T(t)$. This definition of convergence will likely be novel to the students, who will be more familiar with the idea of convergence, meaning that a function approaches a specific number for limiting values of the independent variable.¹⁶

Since we do not know the true solution, we can check for convergence by decreasing the time step Δt . If one sees a significant change in $T(t)$ when decreasing Δt , that means the simulation is not converged. Using smaller time steps makes the simulation more accurate (until numerical noise starts to dominate), but decreasing the time step means increasing the time it takes to run the simulation. Ideally, then, we should use time steps that are not much smaller than the level needed for convergence.

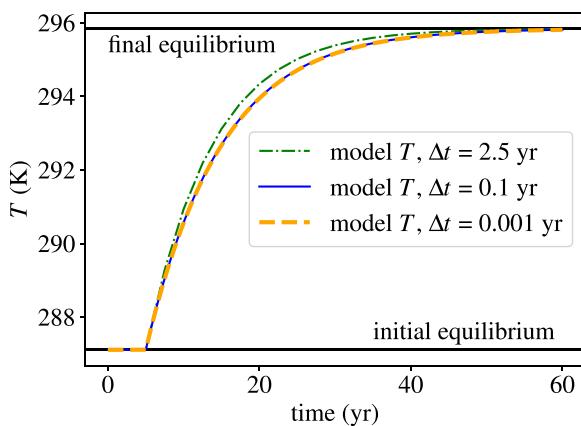


Fig. 1. Global temperature toy model with a one-time very large decrease in outward atmospheric transmission at five years into the simulation leading to a long period out of equilibrium. Time steps are varied to check convergence. The models with $\Delta t = 0.1$ year (solid blue line) and 0.001 year (dashed orange line) appear visually identical on this scale, although the former reaches within 0.1 K of the new equilibrium after 43.1 years, while the latter takes 43.308 years. Is being within 0.5% good enough? Depends on one’s goals. The 2.5-year time step (dot-dash green line) is not converged, since its $T(t)$ differs noticeably from that of the smaller time steps, and it takes also 40 years to reach equilibrium. Note that time steps that do not divide evenly into 5 years will not change the transmission coefficient at the right time, leading to a different kind of discrepancy.

In most of our classes, we try to solve things exactly, and it can be a challenge for students to let go of this mindset and aim for results that are “good enough.” It is also novel for the students to check their results not by comparing to the “right” answer, but rather by comparing two models when we do not know in advance that either is right (to within our tolerance).

Students find that this is a relatively gentle introduction to numerical differential equation solution, given the simplicity of the first-order differential equation and the sufficiency of the Euler approximation for solving the equation. The Euler approach, as implemented in Eq. (3), is only first-order accurate in the time step, and it is usually too inefficient to use in practice, but here it works fine. This is beneficial, as the Euler approach is straightforward to relate to the differential equation itself, Eq. (4), although this is not to be expected to be immediately obvious to students. From this basis in numerical approaches, instructors can lead into a discussion of higher-order Runge–Kutta techniques,³ as well as techniques to solve second-order ordinary and partial differential equations.

To tie this back to physics, it is worth closing by asking the students to explain why larger time steps always lead to *faster* temperature rises.

III. DATA ANALYSIS AND MODEL FITTING

Too often the data we initially give students to analyze is simplified, sanitized, or even mock data. There is use to this, to initially illustrate some points with no added complication, but I think we need to quickly move past this into working with messy data. Students often get to encounter such data in research projects, but it is useful to prepare students for this research experience with an exercise that is standardized, but still has data that are messy and can also be made unique for each campus.

To this end, I chose the temperature datasets that National Oceanic and Atmospheric Administration (NOAA) provides through Climate Data Online¹⁷ (CDO). Hourly temperature readings are available for weather stations across the United States, which provides data for more complex model fitting. In particular, there are diurnal variations in temperature and annual variations in temperature, and on top of that there might be the subtle signal of long-term rising temperatures from climate change. In addition, the data are imperfect and may not be formatted as we would like, providing students with an example of the complications of real-world data.

First show students how to read in the data and turn it into two arrays of time (t) and temperature (T), as a physicist would expect. For time, I use the common astronomical standard of the modified Julian date (MJD), which is the floating-point number of days since midnight on July 17, 1858. Other formats, such as seconds since the Unix epoch, can work, although the CDO data can extend back well before 1970 for many stations. Note that the datasets provided through CDO are good, but imperfect. For example, in the dataset I have provided as an example, the year 1996 is entirely missing. Perhaps this weather station had a period of mourning after the end of the *Calvin and Hobbes* comic strip.¹⁸

In the [supplementary material](#), I provide the questions I pose and full solutions in the temperature and climate data analysis notebook, as well as an example data file for one particular weather station. I start this data file in 1973.

While this location has data back to the 1940s, it is increasingly sparse, and the weather station actually changed location during this period. There are nearly half a million data points, but I chose a plain text format to make it easily human readable. More advanced classes can introduce more efficient file formats.

A. Modeling periodicity

In preparation for fitting a model to the data, I introduce the students to `scipy.curve_fit()`. Prior to this, the students have been introduced to least-squares fitting, and understand that in this context, linear means linear *in the fit parameters*, not the independent variable. (They have also seen `scipy.stats.linregress()` for fitting models that are linear in the independent variable.) We discuss what might be a good first approximation for a dataset with double periodicity, and then they try to fit the model,

$$T(t) = A_1 \sin(\omega_1 t + \phi_1) + A_2 \sin(\omega_2 t + \phi_2) + C. \quad (5)$$

In fact, they are apt to omit the constant C on their first attempt, focused as they are on the periodicity. However, we spend much of the exercise plotting our data and our fits, as I emphasize making sure the results are sensible. They quickly see that omitting C does not yield good results, and then connect their implicit assumption that $C = 0$ to the physical realization that the global average temperature is not 0°C .

However, even with the full seven free parameters, students get extremely disappointing results from curve fitting with Eq. (5). Their plots show extremely poor matches to the data, as seen in the dot-dash line of Fig. 2.

At this point, I draw their attention back to two exercises I had them do earlier. The first was using brute-force calculation to find the least-squares best fit to a collection of lab measurements of the local acceleration due to gravity (g). To accomplish this, they generated an array of trial g values and calculated the χ^2 for each one. They then found the lowest χ^2 and selected the g value associated with it as their best

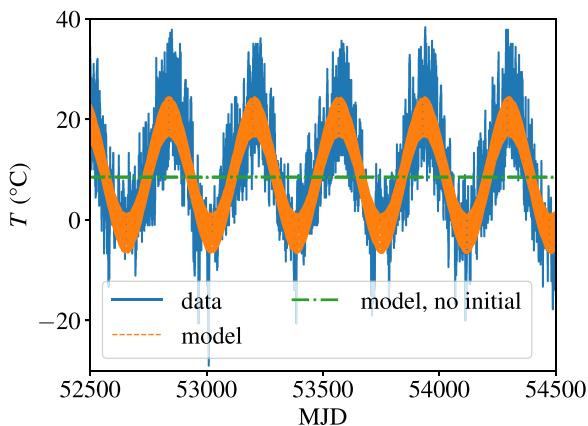


Fig. 2. A five-year segment of the temperature data with the model fit overplotted. Students can see from this plot that the phase is correct for the annual variation component of their model. The dot-dash line that is nearly constant shows the model fit when no initial guesses are given for the parameters, showing that the algorithm got stuck very far from the global minimum and produced a terrible fit. Rapid variation on the timescale of days causes the model to appear solid and very wide in temperature, although students will often think it is not wide enough because they can see the data well above and below. To resolve this, and check the fit on day timescales, it is necessary to plot shorter times as in Figs. 3 and 4.

estimate. This may seem like an absurd approach, as the optimal result is just the arithmetic mean of the data, but this exercise has several benefits. First, it allowed them to easily interpret χ^2 minimization visually, as you can make a 2D plot of χ^2 vs. trial g and see the clear minimum. Such plots become difficult and then impossible as the number of parameters increases. Second, they saw that the least-squares approach, done properly, produces an expected result that they can find independently. Third, they saw that it does not *quite* produce the expected result, because they only had a finite set of trial g values. They can increase the accuracy by using more trial g values, but they saw the tradeoff between accuracy and the cost of more memory and higher runtime.

The second exercise is to use brute-force calculation to do a linear fit to some data, calculating χ^2 over a 2D parameter space of slope and intercept. They chose 1-D arrays of slope values and of intercept values and generated a 2D array of paired slopes and intercepts. They calculated the χ^2 for each pair of slopes and intercepts, and then, as with the first problem, they found the minimum χ^2 and selected the associated slope and intercept as their best fit. Again this method has no practical application, given that the linear problem can be exactly solved with matrix inversion, but it showed them the computational side of problems that cannot be solved exactly. The slopes and intercepts they found were *good*, but not quite right, because the sampling that they could do was somewhat course-grained in a 2D space. Here, the cost of increasing accuracy is much larger than in the first problem of finding g , because the cost of the calculation goes like N^2 rather than like N , where N is the number of points in the 1D arrays. They could still plot χ^2 as a function of the model parameters, and they saw that while some regions were easily ruled out, there was a valley of fairly similar χ^2 values. While the slope and intercept were not degenerate, there was an approximate degeneracy. With an upward sloping line, a decrease in intercept can approximately be compensated for by an increase in slope. Thus, this exercise showed them the increased difficulty of brute-force minimization with increased dimensionality and also showed that minimization can be tricky when multiple parameters are involved and can change in correlated ways.

This background, while not necessary, equips the students to understand what had gone wrong with `scipy.curve_fit()`. They have 10^5 – 10^6 data points and a seven-dimensional parameter space. Brute-force χ^2 minimization is essentially impossible, certainly at a fine enough level to get adequately precise results. We discuss the concept of a local minimum, and how easy it is for an algorithm to get stranded far from the optimal point in parameter space.

I have them read the help page for `scipy.curve_fit()`, and they notice that you can specify initial guesses for parameters. After some thought, they realize they can give *extremely* good initial guesses for ω_1 and ω_2 , and reasonable guesses for A_1 , A_2 , and C . It is possible to guess the phases as well, but it is not necessary; with the information from the periods, `scipy.curve_fit()` is able to find a good fit. Students will discover that the routine does not obey physics conventions. For example, it often returns negative amplitudes, which students may only recently have had drilled out of them in first-year physics.

The model plot with the true best-fit parameters still can look wrong to students. On annual scales, the amplitude looks too small, for example, as seen with the model fit in Fig. 2. Plots that zoom in on few-day regions can help

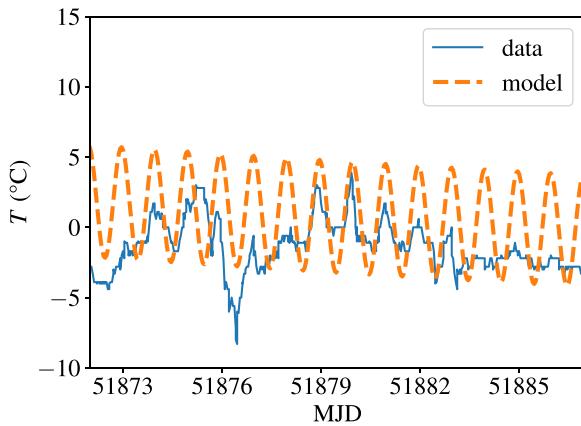


Fig. 3. Best-fit model overplotted on a 15-day segment of data, showing that sometimes there is very little day–night temperature variation and the model overpredicts it, since the model describes *average* variation.

explain what is happening. Some time periods show very little diurnal variation (Fig. 3), while others show much more (Fig. 4).

I have students use their temperature model to predict the temperature for our last class meeting. This is not entirely trivial, as they have to make sure their model (in UTC) is correctly shifted to match our time zone, and many will shift the wrong direction. However, this is a fun test, the like of which students may not have experienced in their classes before, where they are predicting something that is genuinely unknown but can be tested easily and viscerally quickly thereafter. The first time my students did this, their prediction was closer to reality than what the National Weather Service predicted 24 h in advance. This was obviously luck, given that our model knows nothing about current weather conditions, but it is a good example for students about how hard prediction is with complicated systems—even advanced computer models taking into account recent data can be inaccurate for complicated systems with many physical scales.

While I use an explicit two-sine model, I do mention Fourier analysis to my students and stress the utility of such an approach for processes where they do not have good guesses at the periods involved. The power spectrum thus

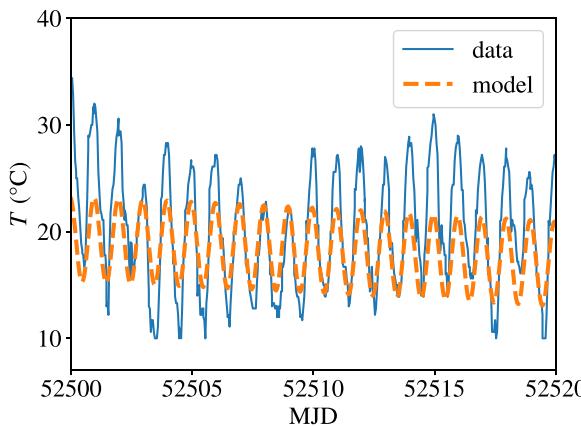


Fig. 4. The model can also *underpredict* the diurnal temperature variation, as seen in this 20-day segment of data. While this is not surprising, given that the model combines only two pure frequency modes, one for annual variation and one for diurnal variation, it still often surprises students new to data analysis. Plots like this are also necessary for students to check that the diurnal phase in their fit is correct.

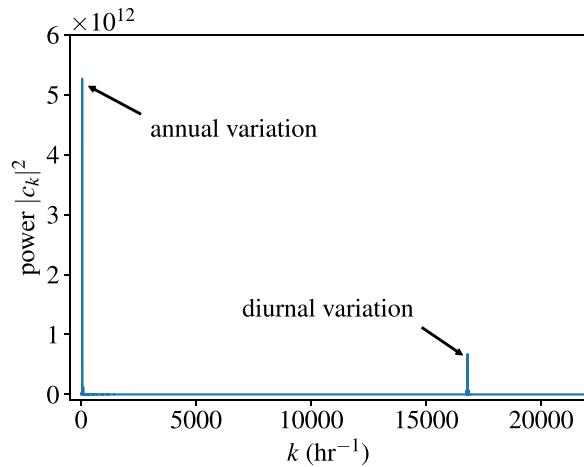


Fig. 5. The Fourier power spectrum of the temperature data. Periods are given by N/k , where N is the number of data points (403 200 in my example dataset). The stronger peak, at $k = 46$ in this plot, thus, has a period of 365.2 days, while the weaker peak, at $k = 16\,800$, has a period of 24.00 h.

obtained, shown in Fig. 5, confirms the two periods that we knew *a priori* were present, gives some information on their relative strengths, and confirms that no other periods of any significance occur in the data. My class has no physics or math prerequisites that would cover Fourier analysis, so, for me, this is just a demonstration, but other instructors could modify this based on their students' backgrounds.

B. Detecting warming

When students plot the residuals after fitting this model to it, they look like noise. However, I introduce the students to the idea of smoothing the data, and they write a running-average smoothing routine that, when averaging over enough data points, shows that there is some residual linear structure to the data. They can perform a linear fit to the residuals and find a significant increase in temperature over the five decades our data cover (Fig. 6). However, I also have them go back and perform a fit to the raw data with an updated version of Eq. (5), adding on a linear portion,

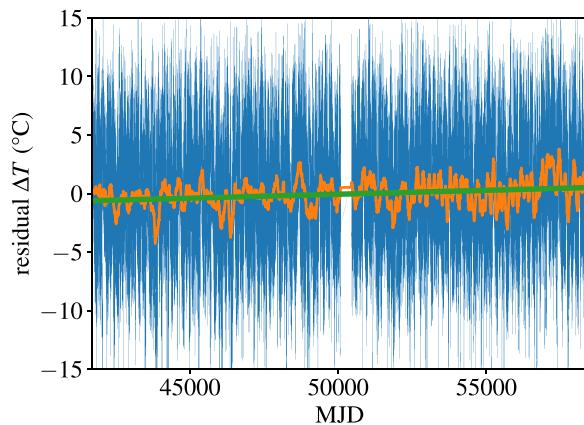


Fig. 6. The lightweight blue line shows the raw residuals after model fitting the data with Eq. (5). When smoothing the residuals with a 4001-h running average (about five and a half months), shown in the heavy orange line, some residual structure is visible in a slow increase in temperature. A linear fit to the residuals, shown in the straight heavy green line tracing across the residuals, shows a rise of 1.1 °C over the time period of these data, from 1973 to 2018. Note that 1996 is missing from this dataset.

$$T(t) = A_1 \sin(\omega_1 t + \phi_1) + A_2 \sin(\omega_2 t + \phi_2) + \alpha t + C. \quad (6)$$

The slope they get seems small until they appreciate the units. With the weather station I use, we find $\alpha = (6.81 \pm 0.14) \times 10^{-5} \text{ }^{\circ}\text{C}/\text{day}$ in the fit to the residuals, but this leads to a $1.1 \text{ }^{\circ}\text{C}$ rise from 1973 through 2018. (When directly fitting the original data, we find a slightly different number, but it is consistent: $\alpha = (6.74 \pm 0.14) \times 10^{-5} \text{ }^{\circ}\text{C}/\text{day}$.)

I stress to the students that a single temperature station is not enough to establish a global trend like climate change, and that satellite observations are crucial for whole-Earth temperature analysis. Nonetheless, it is nice that in this sample dataset, the rise is nearly exactly equal to the global mean temperature rise over this period.

IV. CONCLUSION

These activities are best completed as a combination of in-class work and homework, but the exact proportion of each is flexible and can be adapted to the circumstances of particular courses.

The sample code provided has a specific amount of scaffolding that is appropriate for students who have completed first-year physics but not necessarily anything beyond that and have received a brief introduction to programming, plotting, and statistics (perhaps in the current class, or perhaps earlier). However, the amount of scaffolding is very flexible. Remove some of the supports, and the problems become more challenging—indeed, the data analysis one would be appropriate to challenge graduate students if it had very little scaffolding.

Scaffolding can also be increased to allow use with lower-level students. For example, an instructor could use the temperature residuals after fitting out annual and diurnal variation to discuss climate change with a class that has only seen linear regression. When asking students to write a function, an instructor who wants to provide an intermediate level of scaffolding can provide the docstring, specifying the input and output, and allow the students to create the requisite code.

The activities discussed here are also ripe targets for extending into further projects. The simplest extension of the climate model is to assume that the transmission coefficient changes slowly over time, to mimic more realistic increases of greenhouse gasses in the atmosphere. The effective transmission in (which includes the Earth's albedo) could be changed as a function of temperature, for example modeling the transition of sea ice to open sea with its lower albedo. Implementing models with multiple zones to the Earth, with different temperatures and albedos, would be an interesting but more challenging extension. Of course there are many other extensions possible along the road to a professional climate model, such as taking into account some of the effects of convection, and the atmosphere's nonlinear and wavelength-dependent opacity response to additional greenhouse gasses,¹³ accounting for feedback,⁷ creating a 1D atmospheric model,⁹ and so on.⁵

The temperature data have many extensions. For example, in my class, students are introduced to confidence intervals. Thus, I have the students calculate 95% confidence intervals on the average temperature at a specific day and time using both Gaussian statistics and a nonparametric bootstrap. (Daylight saving time can be nontrivial here, as some dates

have been on standard time some years and daylight saving time other years.) They can also find normal temperature ranges that could be used in calculations of expected heating and cooling loads for buildings in the area. It would also be interesting to use data from many different weather stations to check for consistency on climate change, or compare the temperature stability of various regions.

The Fourier analysis, presented to my students just to let them know that such approaches exist, would form the beginning of the analysis for students familiar with such approaches. Do be careful here. Discrete Fourier transform algorithms tend to assume uniformly sampled data, and real weather station data are nearly but not quite uniform. In the example data I provide, 99.923% of the data are separated by 1 h, but you need 100%. I use spline interpolation, which my students have briefly seen before, to regularize my data.

Scientific data analysis is making increased use of machine learning, including AI, which presents opportunities but also pitfalls. Machine learning approaches create their own algorithms or statistical weights that were not coded by any person, replacing boxes that were black to most users (like `scipy.optimize.curve_fit()`) with boxes fairly black to *all* users. These tools are powerful, and necessary to make cutting-edge progress in science, but, as educators, we must work to lift the veil and give students at least a glimpse of what is going on behind the scenes. We must also make sure that students think critically about the results they get, from any routine or model, and build the skills to assess whether or not things make sense. This has become an issue of great importance in the era of generative AI, where probabilistic output means you can no longer say an algorithm is clearly correct or incorrect, but rather each output must be assessed independently.

Temperature data are well suited to introducing both theoretical techniques and data analysis because of its familiarity to the students. This reduces the cognitive load when it comes to the data and allows them to focus on the novel aspects of numerical simulation and data analysis.

SUPPLEMENTARY MATERIAL

See the [supplementary material](#) online for a text data file containing temperature information for one weather station from 1973 to 2018. Two Jupyter Notebooks of text and Python code are also included. The first describes the climate model of Sec. II. The second goes through the data analysis and model fitting of Sec. III.

AUTHOR DECLARATIONS

Conflict of Interest

The author has no conflicts to disclose.

^aElectronic mail: dsyphers@ewu.edu, ORCID: 0009-0003-8934-0836.

¹American Institute of Physics, *Field of Employment for New Physics Bachelors* (American Institute of Physics, 2023).

²American Institute of Physics, *Common Job Titles for New Physics Bachelors* (American Institute of Physics, 2023).

³M. Newman, *Computational Physics*, revised ed. (CreateSpace Independent Publishing Platform, 2012), pp. 331–339 and 343–354.

⁴F. W. Taylor, *Elementary Climate Physics* (Oxford U.P., Oxford, 2005), Vol. 184, pp. 108–110.

⁵R. T. Pierrehumbert, *Principles of Planetary Climate* (Cambridge U.P., Cambridge, 2010), pp. 143–146.

⁶B. Ryden and B. M. Peterson, *Foundations of Astrophysics* (Pearson, 2010), pp. 197–200.

⁷J. R. Barker and M. H. Ross, "An introduction to global warming," *Am. J. Phys.* **67**(12), 1216–1226 (1999).

⁸S. E. Schwartz, "Resource letter GECC-1: The greenhouse effect and climate change: Earth's natural greenhouse effect," *Am. J. Phys.* **86**(8), 565–576 (2018).

⁹D. J. Wilson and J. Gea-Banacloche, "Simple model to estimate the contribution of atmospheric CO₂ to the Earth's greenhouse effect," *Am. J. Phys.* **80**(4), 306–315 (2012).

¹⁰A. D. Rooney, J. V. Strauss, A. D. Brandon, and F. A. Macdonald, "A cryogenian chronology: Two long-lasting synchronous neoproterozoic glaciations," *Geology* **43**(5), 459–462 (2015).

¹¹A. H. MacDougall, T. L. Frölicher, C. D. Jones, J. Rogelj, H. D. Matthews, K. Zickfeld, V. K. Arora, N. J. Barrett, V. Brovkin, F. A. Burger, M. Eby, A. V. Eliseev, T. Hajima, P. B. Holden, A. Jeltsch-Thömmes, C. Koven, N. Mengis, L. Menviel, M. Michou, I. I. Mokhov, A. Oka, J. Schwinger, R. Séférian, G. Shaffer, A. Sokolov, K. Tachiiri, J. Tjiputra, A. Wiltshire, and T. Ziehn, "Is there warming in the pipeline? A multi-model analysis of the zero emissions commitment from CO₂," *Biogeosciences* **17**(11), 2987–3016 (2020).

¹²NOAA, *How Much Water is in the Ocean?* (NOAA, 2024).

¹³P. C. Nelson, "Effects of greenhouse gases on Earth, Venus, and Mars: Beyond the one-blanket model," *Am. J. Phys.* **91**(9), 721–730 (2023).

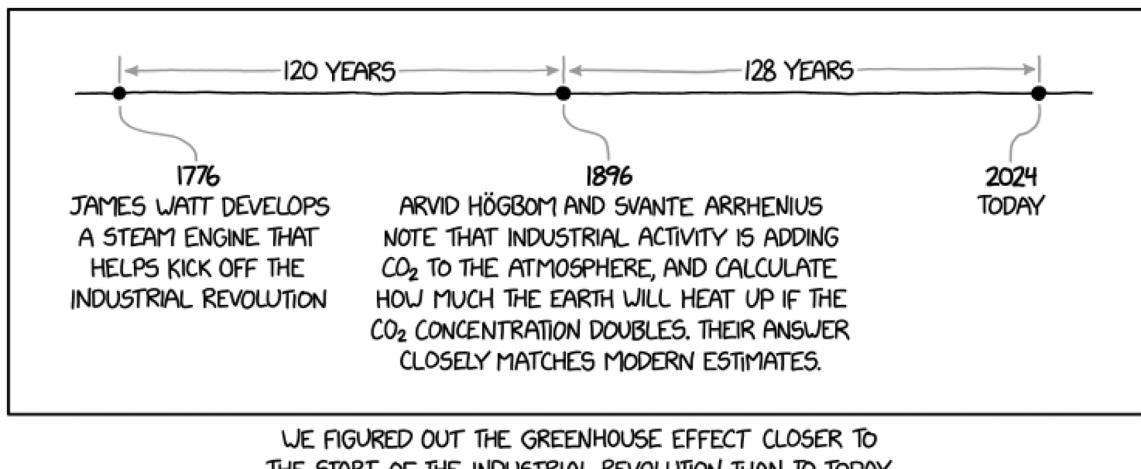
¹⁴A. Berk, P. Conforti, R. Kennett, T. Perkins, F. Hawes, and J. van den Bosch, "MODTRAN® 6: A major upgrade of the MODTRAN® radiative transfer code," in 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2014.

¹⁵I. Gordon, L. Rothman, R. Hargreaves, R. Hashemi, E. Karlovets, F. Skinner, E. Conway, C. Hill, R. Kochanov, Y. Tan, P. Wcislo, A. Finenko, K. Nelson, P. Bernath, M. Birk, V. Boudon, A. Campargue, K. Chance, A. Coustenis, B. Drouin, J.-M. Flaud, R. Gamache, J. Hodges, D. Jacquemart, E. Mlawer, A. Nikitin, V. Perevalov, M. Rotger, J. Tennyson, G. Toon, H. Tran, V. Tyuterev, E. Adkins, A. Baker, A. Barbe, E. Canè, A. Császár, A. Dudarionok, O. Egorov, A. Fleisher, H. Fleurbaey, A. Foltynowicz, T. Furtenbacher, J. Harrison, J.-M. Hartmann, V.-M. Horneman, X. Huang, T. Karman, J. Karns, S. Kassi, I. Kleiner, V. Kofman, F. Kwabia-Tchana, N. Lavrentieva, T. Lee, D. Long, A. Lukashevskaya, O. Lyulin, V. Makhnev, W. Matt, S. Massie, M. Melosso, S. Mikhailenko, D. Mondelain, H. Müller, O. Naumenko, A. Perrin, O. Polyansky, E. Raddaoui, P. Raston, Z. Reed, M. Rey, C. Richard, R. Tóbiás, I. Sadiek, D. Schwenke, E. Starikova, K. Sung, F. Tamassia, S. Tashkun, J. Vander Auwera, I. Vasilenko, A. Vigasin, G. Villanueva, B. Vispoel, G. Wagner, A. Yachmenev, and S. Yurchenko, "The HITRAN2020 molecular spectroscopic database," *J. Quant. Spectrosc. Radiat. Transfer* **277**, 107949 (2022).

¹⁶The sense of convergence used for simulations is just a generalization of this, where a set of functions $T(t, dt)$ converges to a specific $T(t)$ as $dt \rightarrow 0$. The idea of entire functions as "values" will be most familiar to physics students if they have already seen quantum mechanics.

¹⁷NOAA, *NOAA Climate Data Online* (NOAA, 2024).

¹⁸Understandable, really.



Once he had the answer, Arrhenius complained to his friends that he'd "wasted over a full year" doing tedious calculations by hand about "so trifling a matter" as hypothetical CO₂ concentrations in far-off eras (quoted in Crawford, 1997).

(Source: <https://xkcd.com/2889>)