9 — PHY 494: Homework assignment (35 points total)

Due Friday, April 28, 2017, 11:59pm.

Submission is to your **private GitHub repository**.

Read the following instructions carefully. Ask if anything is unclear.

1. `cd` into your assignment repository (change *YourGitHubUsername* to your GitHub username) and run the update script `./scripts/update.sh` (replace *YourGitHubUsername* with your GitHub username):

   ```
   cd  assignments-2017-YourGitHubUsername
   bash ./scripts/update.sh
   ```

   It should create three subdirectories[1] `assignment_09/Submission`, `assignment_09/Grade`, and `assignment_09/Work`.

2. You can try out code in the `assignment_09/Work` directory but you don't have to use it if you don't want to. Your grade with comments will appear in `assignment_09/Grade`.

3. Create your solution in `assignment_09/Submission`. Use Git to `git add` files and `git commit` changes.

   You can create a PDF, a text file or Jupyter notebook inside the `assignment_09/Submission` directory as well as Python code (if required). **Name your files `hw09.pdf` or `hw09.txt` or `hw09.ipynb`**, depending on how you format your work. Files with code (if requested) should be named exactly as required in the assignment.

4. When you are ready to submit your solution, do a final `git status` to check that you haven't forgotten anything, commit any uncommited changes, and `git push` to your GitHub repository. Check on *your* GitHub repository web page[2] that your files were properly submitted.

   You can push more updates up until the deadline. Changes after the deadline will not be taken into account for grading.

Homeworks must be legible and intelligible or may otherwise be returned ungraded with 0 points.

Failure to adhere to the following requirements may lead to homework being returned ungraded with 0 points for the problem.

- Only submit code.
- All code should be in a file `hw_09.py`.

[1] If the script fails, file an issue in the Issue Tracker for PHY494-assignments-skeleton and just create the directories manually.

[2] `https://github.com/ASU-CompMethodsPhysics-PHY494/assignments-2017-`*`YourGitHubUsername`*

- Code will be tested against the unit tests in `test_heat.py`. The grade will be approximately proportional to the number of tests that pass successfully so your code *must* be able run under the tests.

9.1 Temperature distribution in a wall of a house in Phoenix [20 points]

We want to determine the temperature distribution $T(x,t)$ (1d) in the wall of a house that is kept at a constant temperature of $T_{\text{in}} = 293$ K (20°C, 68°F) on the inside. On the outside we want to model the diurnal temperature variation of a typical Phoenix day in summer (July) with average lows of 83°F (28.3°C, 301.4 K) and average highs of 106°F (41.1°C, 314.1 K) [3].

We need to solve the heat equation for the brick wall of thickness L,

$$\frac{\partial T(x,t)}{\partial t} = \frac{K}{C\rho}\frac{\partial^2 T(x,t)}{\partial x^2} \tag{1}$$

under boundary conditions

$$T(0,t) = \bar{T} + \frac{1}{2}(T_{\text{max}} - T_{\text{min}})\cos[\omega(t - t_{\text{max}})] \tag{2}$$

$$\text{with} \quad \bar{T} := \frac{1}{2}(T_{\text{max}} + T_{\text{min}}) \tag{3}$$

$$\text{and} \quad \omega = \frac{2\pi}{t_{\text{day}}} \tag{4}$$

$$T(L,t) = T_{\text{in}}. \tag{5}$$

Eq. 2 roughly models the change in temperature during a day $t_{\text{day}} = 24 \times 60 \times 60$ s, fluctutating between the minimum and the maximum temperature and peaking around noon, i.e., $t_{\text{max}} = 12 \times 60 \times 60$ s.

As initial condition at $t = 0$ (midnight) we can choose any sensible value or distribution but a simple choice is the outside temperature on the left boundary and the inside temperature everywhere else[4]

$$T(0,0) = T_{\text{min}} \tag{6}$$

$$T(x,0) = T_{\text{in}}, \quad x > 0. \tag{7}$$

To solve this problem, write a Python module `hw09.py` that contains the functions

[3] Data from http://www.usclimatedata.com/climate/phoenix/arizona/united-states/usaz0166

[4] An alternative condition is to have everywhere the outside temperature

$$T(x,0) = T(0,t=0) \equiv T_{\text{min}}.$$

In principle, the solution at larger times does not depend sensitively on the initial choice. However, the *tests* were written based on a solution that implements Equations 6 and 7 and thus these initial conditions are prescribed in this problem.

material	K ($\mathrm{W \cdot m^{-1} \cdot K^{-1}}$)	C ($\mathrm{J \cdot kg^{-1} \cdot K^{-1}}$)	ρ ($\mathrm{kg \cdot m^{-3}}$)
brick	1	900	2000
aluminium	205	870	2700
glass window	0.96	840	2600
wood	0.147	250	500

Table 1: Thermal parameters for selected building materials. Source: http://www.engineeringtoolbox.com

T_diurnal(t, Tmin, Tmax, t_max=12*3600) This function takes the current time t (in seconds), the min and max temperatures $T_{\min}$ and $T_{\max}$ and the time at which the temperature peaks ($t_{\max}$) as parameters. It returns the current temperature, as described by Eq. 2.

CrankNicholson_T(L, Dx, Dt, step, t_max, Kappa, CHeat, rho, Tin, Tmin, Tmax) This function solves Eq. 1 with the given boundary conditions using the Crank-Nicholson algorithm. It returns a tuple `(T_plot, parameters)` with the saved temperature profile every `step` iterations and `parameters = (Dx, Dt, step)`.

Use the provided skeleton code to get started and to implement the required function calls. Unit tests are provided in `test_hw09.py` (which you can run in the usual way). Grading of this assignment will primarily rely on all tests passing.

(a) Assume that the brickwall has a thickness $L = 0.3$ m and has the thermal parameters given in Table. 1. Compute the temperature distribution $T(x, t)$ over three days. Plot $T(x, t)$. Submit your code and a plot (use the provided function `plot_surface(T_plot, *parameters)`)(Do you see how the brick wall stores heat during the night and then provides heat to its surroundings?) **[20 points]**

(b) Bonus: Investigate other materials with parameters in Table 1. Assume realistic lengths.[5] **[bonus +4*]**

(c) Bonus: Investigate *heat flow* through the wall by using Fourier's law (in 1D)

$$H(x, t) = -K \frac{\partial T(x, t)}{\partial x} \tag{8}$$

and plot $H(x, t)$. Can you now see explicitly when the wall returns heat to the house? [6] **[bonus +5*]**

[5] No unit tests, free form submission. Submission of PDF figures with titles that identify what you did will be sufficient.

[6] No unit tests, just do it if you're curious