Tree Power Notes

author* author*

May 7, 2020

ABSTRACT.

Key words:

Introduction

Literature review:

1 Assumptions

1. We assume all trees we consider are roughly the same. That means same height, same radius
2. With the previous assumption, we place all devices at the same height (therefore eliminating the z variable in the heat equation), and same depth

2 To do (April)

1. get heat programs running in Python3 (eliminate all syntax error)
   - write source terms from Potter 2002
   - write 2D heat code with both second order centered difference, and ADT method.
2. find appropriate parameters for testing (ask hardware group)
3. ways to validate computation (numerical analysis methods)
4. write a documentation (i.e., this file??)

3 PDE model of temperature within a tree stem

3.1 Parameters to collect and verify. Nick

Notations:

*University of Washington, WA

1. T: temperature (K)
2. ρ: density (kg/m^3)
3. c: specific heat (J/(kgK))
4. t: time (s)
5. k: thermal conductivity (W/(mK))
6. r: distance from center of the tree (m)
7. ϕ: azimuth angle, measured clockwise with south being $\phi = 0$ (radian/degree), [3] assumes $\partial k/\partial \phi = 0$
8. α: albedo of the surface

3.2 The heat equation in cylindrical coordinates

The standard heat equation for temperature distribution in cylindrical coordinates is

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) = \frac{1}{r}\frac{\partial}{\partial r}\left(kr\frac{\partial T}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial \phi}\left(\frac{k}{r}\frac{\partial T}{\partial \phi}\right) \tag{1}$$

heat

We add source terms for the diffusion equation to obtain

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) - \frac{1}{\Delta r}[H + (1-\alpha)(S_{dir} + S_{dif}) + (IR_{in} - IR_{out})] \tag{2}$$

In the following, we explain the source terms:

1. Free convective heat loss/gain happens when the tree surface temp is different from the ambient temp. Forced convection happens when there is wind. We denote the heat loss/gain by H, and
$$H = h(T_{sfc} - T_{air}), \tag{3}$$
with h (W/(m^2K)) being the convective heat transfer coefficient. Here
$$h = h_{free} - h_{forced}, \tag{4}$$
more details in 3.
2. $S_{dir} + S_{dif}$ represents direct solar radiation, plus diffusion solar radiation.
3. $IR_{in} - IR_{out}$ represents wave radiation from and to the tree. More details see [3].

Question to think about: Take part of the source term as boundary condition? What is the boundary condition with source term already implemented?

In this work, we solve the heat equation to model the temperature in tree trunks, for the purpose of energy harvesting. We modify above source terms to fit the parameters for trees in Washington state.

3.3 FD scheme for the heat equation

We solve the above equation numerically with the Finite-Difference (FD) method, as the FD method is simple to implement, and robust.

A detailed discussion on axi-symmetric heat equation is in [7]. We model the heat equation to depend on the azimuth angle ϕ, as trees in Washington state experience different sunlight and therefore growth in different direction. pg.251, section 3.5.6

For simplicity of discussion, we assume $\rho,\ c, k$ are all constants. This is a reasonable assumption, as the trees should not have parameters that change drastically within the trunk. Then the constant coefficient equation (3) simplifies to

$$\frac{\rho c}{k}\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r} + \frac{1}{r^2}\frac{\partial^2 T}{\partial \phi^2} \tag{5}$$

simple_heat

With centered FD discretization in space, and forward in time for example, we have

- $\dfrac{\partial T}{\partial t} \approx \dfrac{T^1_{i,j} - T^0_{i,j}}{\Delta t}$
- $\dfrac{\partial^2 T}{\partial r^2} \approx \dfrac{T^0_{i-1,j} + T^0_{i+1,j} - 2T^0_{i,j}}{(\Delta r)^2}$
- $\dfrac{\partial T}{\partial r} \approx \dfrac{T^0_{i+1,j} - T^0_{i-1,j}}{2\Delta r}$
- $\dfrac{\partial^2 T}{\partial \phi^2} \approx \dfrac{T^0_{i,j-1} + T^0_{i,j+1} - 2T^0_{i,j}}{(\Delta \phi)^2}$

Here $T^n_{i,j} \approx T(r_i, \phi_j, t_n)$ denotes the numerical solution at the point (r_i, ϕ_j, t_n). In general, we compute the heat equation iteratively in time with the general time forwarding scheme:

$$\frac{T^1_{i,j} - T^0_{i,j}}{\Delta t} = D\ T_{i,j}, \tag{6}$$

where the right hand side is a discretization in space, with D being the difference operator. Depending on D, the above *continuous differential equation* can be discretized in the following two ways:

3.3.1 Forward in time, centered difference Method (FTCD)

The FTCD is the most straightforward method, where we directly apply the above listed discretization to the heat equation.

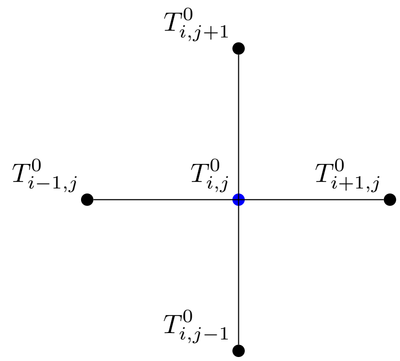
$$\begin{aligned} T^1_{i,j} =& \frac{k\Delta t}{\rho c}\Bigg[\underbrace{\frac{1}{(\Delta r)^2}\Big(T^0_{i-1,j} + T^0_{i+1,j} - 2T^0_{i,j}\Big)}_{\frac{\partial^2 T}{\partial r^2}} + \underbrace{\frac{1}{2r_i\Delta r}\Big(T^0_{i+1,j} - T^0_{i-1,j}\Big)}_{\frac{1}{r}\frac{\partial T}{\partial r}} \\ &+ \underbrace{\frac{1}{r_i^2(\Delta\phi)^2}\Big(T^0_{i,j-1} + T^0_{i,j+1} - 2T^0_{i,j}\Big)}_{\frac{1}{r^2}\frac{\partial^2 T}{\partial \phi^2}}\Bigg] + T^0_{i,j}, \end{aligned} \tag{7}$$

with $k_1 = \dfrac{k\Delta t}{\rho c(\Delta r)^2}$, $k_2 = \dfrac{k\Delta t}{2\rho c r_i \Delta r}$, and $k_3 = \dfrac{k\Delta t}{\rho c r_i^2(\Delta\phi)^2}$, above equation can be implemented as

$$T^1_{i,j} = k_1\Big(T^0_{i-1,j} + T^0_{i+1,j} - 2T^0_{i,j}\Big) + k_2\Big(T^0_{i+1,j} - T^0_{i-1,j}\Big) + k_3\Big(T^0_{i,j-1} + T^0_{i,j+1} - 2T^0_{i,j}\Big) + T^0_{i,j}, \tag{8}$$

where we obtain $T^1_{i,j}$ each time step with T values in several locations from previous step. This system of equations can be solved directly. See implementation (that needs modification) in `heatPolar.py`. Here we can denote the general scheme to be

$$\frac{T^1_{i,j} - T^0_{i,j}}{\Delta t} = D\ T^0_{i,j}. \tag{9}$$



Space information at each time step.

Imagine one more axis normal to the paper, pointing outwards above the blue dot. That would be $T^1_{i,j}$.

3.3.2 Crank-Nicolson Method (CNM)

We also solve the heat equation with the Crank-Nicolson method, as it is easy to modify given the FTCD method. To put simply, the general scheme is modified by averaging the space information of the current step and the next step:

$$\frac{T^1_{i,j} - T^0_{i,j}}{\Delta t} = \frac{1}{2} D\ T^0_{i,j} + \frac{1}{2} D\ T^1_{i,j}. \tag{10}$$

The specific scheme is

$$\begin{aligned} &T^1_{i,j} - 0.5k_1\left(T^1_{i-1,j} + T^1_{i+1,j} - 2T^1_{i,j}\right) - 0.5k_2\left(T^1_{i+1,j} - T^1_{i-1,j}\right) - 0.5k_3\left(T^1_{i,j-1} + T^1_{i,j+1} - 2T^1_{i,j}\right) \\ &= 0.5k_1\left(T^0_{i-1,j} + T^0_{i+1,j} - 2T^0_{i,j}\right) + 0.5k_2\left(T^0_{i+1,j} - T^0_{i-1,j}\right) + 0.5k_3\left(T^0_{i,j-1} + T^0_{i,j+1} - 2T^0_{i,j}\right) + T^0_{i,j}, \end{aligned} \tag{11}$$

here $k_1,\ k_2,\ k_3$ are previously defined.

3.4 Implementation in Python and numerical results

With the FTCD method, we need to worry about numerical stability. By the von Neumann stability analysis, we write $T(r,\phi,t_n) = e^{ikr+il\phi}$, and $T(r,\phi,t_{n+1}) = Ge^{ikr+il\phi}$. We force $G \leq 1$. Plug $T(r,\phi,t_n) = T^0_{i,j}$ and $T(r,\phi,t_{n+1}) = T^1_{i,j}$ into the Finite Difference equation, we get (we will temporarily denote $c = \dfrac{k}{\rho c}$)

$$G = c\Delta t\left[\frac{1}{(\Delta r)^2}(e^{-ik\Delta r} + e^{ik\Delta r} - 2) + \frac{1}{2r_i\Delta r}(e^{ik\Delta r} - e^{-ik\Delta r}) + \frac{1}{(r_i\Delta\phi)^2}(e^{-il\Delta\phi} + e^{il\Delta\phi} - 2)\right] + 1 \tag{12}$$

As $r_i \geq \Delta r$, we replace all r_i with Δr, and we get

$$G \leq c\Delta t\left[\frac{1}{(\Delta r)^2}(2\cos(k\Delta r)-2)+\frac{1}{2(\Delta r)^2}2i\sin(k\Delta r)+\frac{1}{(\Delta r\Delta\phi)^2}(2\cos(l\Delta\phi)-2)\right]+1 \tag{13}$$

We request $|G| \leq 1$, and it comes down to the following inequality:

$$\left[1-\frac{c\Delta t}{(\Delta r)^2}4\sin^2(k\Delta r/2)-\frac{c\Delta t}{(\Delta r\Delta\phi)^2}4\sin^2(l\Delta\phi/2)\right]^2+\left[\frac{c\Delta t}{(\Delta r)^2}\sin(k\Delta r)\right]^2 \leq 1 \tag{14}$$

Worst case is when $k\Delta r = l\Delta\phi = \pi$, and we get the stability requirement:

$$\left(1-\frac{c\Delta t}{(\Delta r)^2}4-\frac{c\Delta t}{(\Delta r\Delta\phi)^2}4\right)^2 \leq 1 \tag{15}$$

For vectorization, we rewrite the above equation as

$$T^1_{i,j} = (k_1-k_2)T^0_{i-1,j}+(k_1+k_2)T^0_{i+1,j}+(1-2k_1-2k_3)T^0_{i,j}++k_3T^0_{i,j-1}+k_3T^0_{i,j+1} \tag{16}$$

Read Chapter 9 of Randy LeVeque.

Reference BAD FORMAT. FOR convience only

1. Yan. D. et al, Time-Dependent Finite-Volume Model of Thermoelectric Devices, IEEE, Transactions on industry applications, Vol. 50, No.1, Jan/Feb 2014 [thermoelectric]

2. MIT notes online [mit]

3. Potter, A., Andresen, J., A finite-difference model of temperatures and heat flow within a tree stem, Can. J. For. Res. **32**, 548-555 (2002) [heat]

4. Bownman, W. et al, Sapwood temperature gradients between lower stems and the crown do not influence estimate of stand-level stem CO_2 efflux, Tree Physiology, 28, 1553-1559 [sapwood]

5. Chen, J., et al, An empirical model for predictin diurnal air-temperature gradients from edge into old-growth Douglas-fir forest, Ecological Modeling, Vol. 67, Issues 2-4, pg 179-198, June 1993 [airtempforest]

6. Tanja, S. et al., Air temperature triggers the recovery of evergreen boreal forest photosynthesis in spring, Global Change Biology, vol 9, issue 10, pg 1410-1426, Oct 2003 [evergreen]

7. Linge S., Langtangen H.P. (2017) Diffusion Equations. In: Finite Difference Computing with PDEs. Texts in Computational Science and Engineering, vol 16. Springer, Cham [2017book]