

Recently we have determined that the exiting definitions for theoretical models, data definitions, general definitions and instance models are not entirely consistent. Moreover, the terminology, especially the terminology that combines “theory” and “model” is confusing.

Our new terminology will be based on the concept of different “levels” of theories. We currently have what we are calling context theories, background theories, theories and final theories. Context theories are the starting point for a given scientific model. These theories are not defined, but their implicit presence is necessary for completeness. Typical examples include theories for arithmetic operations, differentiation, integration, vector calculus, etc. Following the context theories we have background theories. Background theories are defined, but not derived. Typically they will be the general forms of the conservation equations, like conservation of thermal energy or momentum, and constitutive equations. Assumptions are often invoked by The level of detail used to define background theories can vary, since we do not always need the full theory; we sometimes just need to know that the theory exists. The background theories are refined into other theories by making assumptions, like plane stress, or linear elasticity, or isothermal material properties, or laminar flow, etc. These theories are combined and refined until the point where we have final theories. All theories are part of the documentation, but the final theories are the ones that will be transformed into code.

We also have data definitions. A data definition is a label for part of a theory. Assumptions will be maintained from the previous terminology.

Below is an attempt to clarify the new concept of different levels of theories by re-writing the relevant parts of the NoPCM model using the new conceptual model of theories.

0.1 Terminology Definition

- *thermal flux* Provide the sign convention for thermal flux.

1 Assumptions

1.1 Assumptions for Background and Refined Theories

er-Coeffs-Constant: All heat transfer coefficients are constant over time. (RefBy: [BT:nwtnCooling.](#))

Temp-Across-Tank: The water in the tank is fully mixed, so the temperature of the water is the same throughout the entire tank. (RefBy: [RT:rocTempSimp.](#))

stant-over-Volume: The density of water has no spatial variation; that is, it is constant over their entire volume. (RefBy: [RT:rocTempSimp.](#))

stant-over-Volume: The specific heat capacity of water has no spatial variation; that is, it is constant over its entire volume. (RefBy: [RT:rocTempSimp.](#))

Cooling-Coil-Water: Newton’s law of convective cooling applies between the heating coil and the water. (RefBy: [RT:htFluxWaterFromCoil.](#))

- constant-over-Time: The temperature of the heating coil is constant over time. (RefBy: [LC:Temperature-Coil-Variable-Over-Day](#) and [RT:htFluxWaterFromCoil](#).)
- no-Temp-Discharge: The model only accounts for charging the tank, not discharging. The temperature of the water can only increase, or remain constant; it cannot decrease. This implies that the initial temperature is less than (or equal to) the temperature of the heating coil. (RefBy: [LC:Discharging-Tank](#).)
- water-Always-Liquid: The operating temperature range of the system is such that the material (water in this case) is always in liquid state. That is, the temperature will not drop below the melting point temperature of water, or rise above its boiling point temperature. (RefBy: [UC:Water-Fixed-States](#), [BT:sensHtE](#), [FT:heatEInWtr](#), and [FT:eBalanceOnWtr](#).)
- atmospheric-Pressure-Tank: The pressure in the tank is atmospheric, so the melting point temperature and boiling point temperature of water are 0°C and 100°C, respectively. (RefBy: [FT:heatEInWtr](#).)
- small-Coil-Negligible: When considering the volume of water in the tank, the volume of the heating coil is assumed to be negligible. (RefBy: [DD:waterVolume_nopcm](#).)
- [\[New: —SS\]](#) The assumptions from this point on are newly added.
- constant-Density-OverVol: The density of the liquid does not depend on the spatial location within the entire volume. ($\forall \mathbf{s}, \mathbf{s}' | \rho(\mathbf{s}, T) = \rho(\mathbf{s}', T)$) (RefBy: [RT:consThermE:ReduceDep](#).)
- constant-SpecHeat-OverVol: The specific heat of the liquid does not depend on the spatial location within the entire volume. (RefBy: [RT:consThermE:ReduceDep](#).)
- constant-molar-FormTemp-OverVol: The specific heat of the liquid does not depend on the spatial location within the entire volume. (RefBy: [RT:consThermE:ReduceDep](#).)
- constant-HeatGen-OverVol: The heat generation of the liquid does not depend on the spatial location within the entire volume. (RefBy: [RT:consThermE:ReduceDep](#).)
- constant-DensityIndepTemp: The density of the material does not depend on the temperature. (RefBy: [RT:consThermE:ReduceDep](#).)
- constant-SpecHeatIndepTemp: The specific heat capacity of the material does not depend on the temperature. (RefBy: [RT:consThermE:ReduceDep](#).)
- VolumeIsCompact: A compact volume has the property of compactness, which “generalizes the notion of a subset of Euclidean space being closed (containing all its limit points) and bounded (having all its points lie within some fixed distance of each other)” [[Wiki-CompactSpace2022](#)]. (RefBy: [RT:rocTempSimp](#).)
- piecewise-Smooth-Surface: The surface of the volume can be broken into separate surfaces which are all smooth, where the surface is smooth if its partial derivatives of every order exist at every point of the domain [[Wiki-DiffGeom2022](#)]. (RefBy: [RT:rocTempSimp](#).)

iformOverSurfaces: The projection of the thermal flux in the direction of the unit outward normal is uniform over each surface that encloses the volume (RefBy: [RT:rocTempSimp.](#))

ransCoeffOverSurf: The heat transfer coefficient over the surface is uniform. That is, the heat transfer coefficient does not depend on the location on the surface (RefBy: [RT:nwtnCooling:ReduceDep.](#))

ormTempOverSurf: The temperature at the surface is uniform. That is, the temperature does not depend on the location on the surface (RefBy: [RT:nwtnCooling:ReduceDep.](#))

iformEnviroTemp: The temperature at the environment around the surface is uniform. That is, the environment temperature does not depend on the location on the surface (RefBy: [RT:nwtnCooling:ReduceDep.](#))

nsCoeffIndepTime: The heat transfer coefficient does not change over time. (RefBy: [RT:nwtnCooling:ReduceDep.](#))

nsCoeffIndepTemp: The heat transfer coefficient is independent of temperature. (RefBy: [RT:nwtnCooling:ReduceDep.](#))

1.2 Assumptions for Final Theories

ermal-Energy-Only: The only form of energy that is relevant for this problem is thermal energy. All other forms of energy, such as mechanical energy, are assumed to be negligible. (RefBy: [BT:consThermE.](#))

niformFullyMixed: The water tank is fully mixed so that the material properties and temperature are uniform throughout its volume. (RefBy: [RT:htFluxWaterFromCoil.](#))

eneration-By-Water: No internal heat is generated by the water; therefore, the volumetric heat generation per unit volume is zero. (RefBy: [UC:No-Internal-Heat-Generation](#) and [FT:eBalanceOnWtr.](#))

ct-Insulation-Tank: The tank is perfectly insulated so that there is no heat loss from the tank. (RefBy: [LC:Tank-Lose-Heat](#) and [FT:eBalanceOnWtr.](#))

yIndepTempWater: The density of the water does not depend on the temperature. (RefBy: [FT:eBalanceOnWtr.](#))

tIndepTempWater: The specific heat capacity of the material does not depend on the temperature. (RefBy: [RT:consThermE:ReduceDep.](#))

ransCoeffOverCoil: The heat transfer coefficient over the surface of the coil is uniform. That is, the heat transfer coefficient does not depend on the location on the coil's surface (RefBy: [RT:htFluxWaterFromCoil.](#))

stant-over-Length: The temperature of the heating coil does not vary along its length. (RefBy: [LC:Temperature-Coil-Variable-Over-Length](#), [RT:htFluxWaterFromCoil](#).)

coeffCoilIndepTime: The heat transfer coefficient between the coil and the tank of water does not change over time. (RefBy: [RT:htFluxWaterFromCoil](#).)

coeffCoilIndepTemp: The heat transfer coefficient between the coil and the tank of water does not depend on the temperature. (RefBy: [RT:htFluxWaterFromCoil](#).)

2 Context Theories

Some theories do not have to be explicitly invoked. They are part of the context for the other theories, without having to be explicitly stated or defined. The context theories for this problem are as follows:

- arithmetic
- operations
- differentiation
- partial differentiation
- integration
- vector calculus (gradient operator, dot product)
- Gauss's divergence theorem [[Should this be a separate theory? Possibly a theory that is in the background, but not actually printed in the SRS document? —SS](#)]

3 Background Theories (BT)

Refname	BT:consThermE
Label	Conservation of thermal energy
Equation	$-\nabla \cdot \mathbf{q}(\mathbf{x}, t) + g(\mathbf{x}, t) = \rho(\mathbf{x}, T)C(\mathbf{x}, T)\frac{\partial T(\mathbf{x}, t)}{\partial t}$
Description	<p> ∇ is the gradient (Unitless) \mathbf{q} is the thermal flux vector ($\frac{\text{W}}{\text{m}^2}$) g is the volumetric heat generation per unit volume ($\frac{\text{W}}{\text{m}^3}$) ρ is the density ($\frac{\text{kg}}{\text{m}^3}$) C is the specific heat capacity ($\frac{\text{J}}{\text{kg}^\circ\text{C}}$) t is the time (s) T is the temperature ($^\circ\text{C}$) $[\mathbf{x}$ is the spatial position (m) —SS] </p>
Notes	<p> The above equation gives the law of conservation of energy for transient heat transfer in a given material. For this equation to apply, other forms of energy, such as mechanical energy, are assumed to be negligible in the system (A:Thermal-Energy-Only). [This form of the conservation equation is general. A specific coordinate system (Cartesian, polar, spherical etc) is not assumed. The spatial location is represented symbolically by \mathbf{x}. The divergence operator ($\nabla \cdot$) can be defined for whatever coordinate system is used. The thermal flux (\mathbf{q}), the volumetric heat generation (g) and temperature (T) depend on spatial position (\mathbf{x}) and time (t). The material properties, density (ρ) and specific heat capacity (C), depend on spatial position (\mathbf{x}) and temperature (T). —SS] </p>
Source	Fourier Law of Heat Conduction and Heat Equation
RefBy	RT:rocTempSimp , RT:consThermE:ReduceDep

Preconditions for **BT:consThermE**:

- **A:Thermal-Energy-Only**

Refname	BT:sensHtE
Label	Sensible heat energy (no state change)
Equation	$E = C^L m \Delta T$
Description	<p>E is the sensible heat (J) C^L is the specific heat capacity of a liquid ($\frac{\text{J}}{\text{kg}^\circ\text{C}}$) m is the mass (kg) ΔT is the change in temperature ($^\circ\text{C}$)</p>
Notes	<p>E occurs as long as the material does not reach a temperature where a phase change occurs, as assumed in A:Water-Always-Liquid. [This could actually be an assumption of no state change. This could be a generic assumption that applies for any state of matter. This particular instantiation applies for liquid, but with a change in the symbol for specific heat capacity it could be changed to a solid (for instance). —SS]</p>
Source	Definition of Sensible Heat
RefBy	FT:heatEInWtr

Refname	BT:nwtnCooling
Label	Newton's law of cooling
Equation	$\mathbf{q}(\mathbf{s}, t) \cdot \hat{\mathbf{n}} = h(\mathbf{s}, t, T)(T(\mathbf{s}, t) - T_{\text{env}}(\mathbf{s}, t))$
Description	<p> \mathbf{q} is the thermal flux vector ($\frac{\text{W}}{\text{m}^2}$) $[\hat{\mathbf{n}}$ is a unit outward normal from a surface (m) —SS] $[\mathbf{s}$ is a location on the surface (m) —SS] t is the time (s) h is the convective heat transfer coefficient ($\frac{\text{W}}{\text{m}^2\text{C}}$) T is the temperature of the body ($^{\circ}\text{C}$) T_{env} is the temperature of the environment surrounding the body ($^{\circ}\text{C}$) </p>
Notes	<p>Newton's law of cooling describes convective cooling from a surface. The law is stated as: the rate of heat loss from a body is proportional to the difference in temperatures between the body and its surroundings. In the form given the thermal flux (\mathbf{q}), heat transfer coefficient (h), temperature of the body (T) and temperature of the environment (T_{env}) can all vary with the location on the surface (\mathbf{s}) and over time (t).</p>
Source	[incroperaEtAl2007]
RefBy	RT:htFluxWaterFromCoil

Refname	BT:density
Label	Density
Equation	$\rho = \frac{m}{V}$
Description	<p>ρ is the density of a material ($\frac{\text{kg}}{\text{m}^3}$)</p> <p>$m$ is the mass of the body (kg)</p> <p>V is the volume of the body (m^3)</p>
Notes	Density is the mass per unit volume.
Source	—
RefBy	BT:consThermE, RT:rocTempSimp, DD:waterMass

4 Refined Theories

This section collects the laws and equations that will be used to build the instance models.

Refname	RT:consThermE:ReduceDep
Label	Refine conservation of thermal energy to simplify the equation by reducing the dependencies of the variables on space and temperature.
Equation	$-\nabla \cdot \mathbf{q}(\mathbf{x}, t) + g(t) = \rho C \frac{dT(t)}{dt}$
Description	<p> ∇ is the gradient (Unitless) \mathbf{q} is the thermal flux vector ($\frac{\text{W}}{\text{m}^2}$) g is the volumetric heat generation per unit volume ($\frac{\text{W}}{\text{m}^3}$) ρ is the density ($\frac{\text{kg}}{\text{m}^3}$) C is the specific heat capacity ($\frac{\text{J}}{\text{kg}^\circ\text{C}}$) t is the time (s) T is the temperature ($^\circ\text{C}$) [\mathbf{x} is the spatial position —SS] </p>
Source	—
RefBy	RT:rocTempSimp

Preconditions for RT:consThermE:ReduceDep:

- A:Thermal-Energy-Only
- A:UniformDensity-OverVol,
- A:UniformSpecHeat-OverVol,
- A:UniformTemp-OverVol,
- A:UniformHeatGen-OverVol,
- A:DensityIndepTemp,
- A:SpecHeatIndepTemp

Detailed derivation of constant material properties conservation equation: Starting from **BT:consThermE**:

$$-\nabla \cdot \mathbf{q}(\mathbf{x}, t) + g(\mathbf{x}, t) = \rho(\mathbf{x}, T)C(\mathbf{x}, T)\frac{\partial T(\mathbf{x}, t)}{\partial t}$$

Apply the assumption that the density does not depend on the spatial location (**A:UniformDensity-OverVol**):

$$-\nabla \cdot \mathbf{q}(\mathbf{x}, t) + g(\mathbf{x}, t) = \rho(T)C(\mathbf{x}, T)\frac{\partial T(\mathbf{x}, t)}{\partial t}$$

Apply the assumption that the specific heat does not depend on the spatial location (**A:UniformSpecHeat-OverVol**):

$$-\nabla \cdot \mathbf{q}(\mathbf{x}, t) + g(\mathbf{x}, t) = \rho(T)C(T)\frac{\partial T(\mathbf{x}, t)}{\partial t}$$

Apply the assumption that the temperature does not depend on the spatial location (**A:UniformTemp-OverVol**):

$$-\nabla \cdot \mathbf{q}(\mathbf{x}, t) + g(\mathbf{x}, t) = \rho(T)C(T)\frac{dT(t)}{dt}$$

Apply the assumption that the heat generation does not depend on the spatial location (**A:UniformHeatGen-OverVol**):

$$-\nabla \cdot \mathbf{q}(\mathbf{x}, t) + g(t) = \rho(T)C(T)\frac{dT(t)}{dt}$$

Apply the assumption that the density is independent of the temperature (**A:DensityIndepTemp**):

$$-\nabla \cdot \mathbf{q}(\mathbf{x}, t) + g(t) = \rho C(T)\frac{dT(t)}{dt}$$

Apply the assumption that the specific heat is independent of the temperature (**A:SpecHeatIndepTemp**):

$$-\nabla \cdot \mathbf{q}(\mathbf{x}, t) + g(t) = \rho C\frac{dT(t)}{dt}$$

Refname	RT:rocTempSimp
Label	Simplified rate of change of temperature
Equation	$mC \frac{dT(t)}{dt} = - \sum_{i=0}^n q_i(t) A_i + g(t) V$
Description	<p> m is the mass (kg) C is the specific heat capacity ($\frac{\text{J}}{\text{kg}^\circ\text{C}}$) t is the time (s) T is the temperature ($^\circ\text{C}$) q_i is the heat flux input across surface i ($\frac{\text{W}}{\text{m}^2}$) [changed —SS] A_i is the surface area for surface i over which heat is transferred in (m^2) [changed —SS] g is the volumetric heat generation per unit volume ($\frac{\text{W}}{\text{m}^3}$) V is the volume (m^3) [n the number of surfaces over which heat transfer occurs —SS] </p>
Source	—
RefBy	FT:eBalanceOnWtr

Preconditions for **RT:rocTempSimp:**

- **A:Thermal-Energy-Only**,
- **A:UniformDensity-OverVol**,
- **A:UniformSpecHeat-OverVol**,
- **A:UniformTemp-OverVol**,
- **A:UniformHeatGen-OverVol**,
- **A:DensityIndepTemp**,

- **A:SpecHeatIndepTemp**,
- **A:VolumeIsCompact**,
- **A:VolHasPiecewise-Smooth-Surface**,
- **A:ThermalFlux-UniformOverSurfaces**

Detailed derivation of simplified rate of change of temperature: Integrating **RT:consThermE:ReduceDep** over a volume (V), we have:

$$-\int_V \nabla \cdot \mathbf{q}(\mathbf{x}, t) dV + \int_V g(t) dV = \int_V \rho C \frac{dT(t)}{dt} dV$$

Moving all terms that do not vary over space outside of the volume integrals and integrating the remaining volume integrals we have:

$$-\int_V \nabla \cdot \mathbf{q}(\mathbf{x}, t) dV + g(t)V = \rho CV \frac{dT(t)}{dt}$$

Assuming the volume is compact (**A:VolumeIsCompact**) and has a piecewise smooth surface (**A:VolHasPiecewise-Smooth-Surface**), we can apply Gauss's Divergence Theorem to the first term over the surface S of the volume, with \mathbf{q} as the thermal flux vector from the surface, \mathbf{s} as the location on the surface, and $\hat{\mathbf{n}}$ as a unit outward normal vector from the surface:

$$-\int_S \mathbf{q}(\mathbf{s}, t) \cdot \hat{\mathbf{n}} dS + g(t)V = \rho CV \frac{dT(t)}{dt}$$

The surface S of the volume V can be divided into n separate surfaces S_i , each of area A_i . For each of these surfaces S_i we assume $q_i(t) = \mathbf{q}(\mathbf{s}, t) \cdot \hat{\mathbf{n}}$ is uniform. That is, the projection of the thermal flux q_i in the direction of the unit outward normal is uniform over the surface S_i (**A:ThermalFlux-UniformOverSurfaces**). Since the normal thermal flux does not vary over the surface, it can be moved outside of the integration. The integral over the surface then is simply the surface area A_i . With this information the previous equation can be rewritten as:

$$-\sum_{i=0}^n q_i(t)A_i + g(t)V = \rho CV \frac{dT(t)}{dt}$$

Substituting in $\rho=m/V$ (**BT:density**), the above equation can be rewritten as:

$$mC \frac{dT(t)}{dt} = -\sum_{i=0}^n q_i(t)A_i + g(t)V$$

When applying the above equation, remember that the sign of q_i is determined relative to the unit outward normal each surface. If the thermal flux is travelling out of the body, in

the direction of the unit outward normal, then the sign of q_i is positive. If the thermal flux is travelling into the body, then the direction is opposite to the unit outward normal and the sign will be negative.

Refname	RT:nwtncooling:ReduceDep
Label	Refine Newton's law of cooling to simplify the relation by reducing the dependencies of the variables on location and temperature.
Equation	$q(t) = h(T(t) - T_{\text{env}}(t))$
Description	<p>q is the thermal flux normal to the surface ($\frac{\text{W}}{\text{m}^2}$)</p> <p>$t$ is the time (s)</p> <p>h is the convective heat transfer coefficient ($\frac{\text{W}}{\text{m}^2\text{C}}$)</p> <p>$T$ is the temperature of the body ($^{\circ}\text{C}$)</p> <p>T_{env} is the temperature of the environment surrounding the body ($^{\circ}\text{C}$)</p>
Source	—
RefBy	[? —SS]

Preconditions for RT:nwtncooling:ReduceDep:

- A:UniformHeatTransCoeffOverSurf,
- A:UniformTempOverSurf,
- A:UniformEnviroTemp,
- A:HeatTransCoeffIndepTime,
- A:HeatTransCoeffIndepTemp

Detailed derivation of simplified Newton's Law of Cooling: Starting from **BT:nwt-nCooling**:

$$\mathbf{q}(\mathbf{s}, t) \cdot \hat{\mathbf{n}} = h(\mathbf{s}, t, T)(T(\mathbf{s}, t) - T_{\text{env}}(\mathbf{s}, t))$$

First introduce q as the normal flux to the surface $q(\mathbf{s}, t) = \mathbf{q}(\mathbf{s}, t) \cdot \hat{\mathbf{n}}$ and substitute this into the above equation for:

$$q(\mathbf{s}, t) = h(\mathbf{s}, t, T)(T(\mathbf{s}, t) - T_{\text{env}}(\mathbf{s}, t))$$

Apply the assumption that the heat transfer coefficient is uniform over the surface (**A:UniformHeatTransCoeffOverSurf**):

$$q(\mathbf{s}, t) = h(t, T)(T(\mathbf{s}, t) - T_{\text{env}}(\mathbf{s}, t))$$

Apply the assumption that the temperature is uniform over the surface (**A:UniformTempOverSurf**):

$$q(\mathbf{s}, t) = h(t, T)(T(t) - T_{\text{env}}(\mathbf{s}, t))$$

Apply the assumption that the temperature of the environment is uniform over the surface (**A:UniformEnviroTemp**):

$$q(\mathbf{s}, t) = h(t, T)(T(t) - T_{\text{env}}(t))$$

Since the right hand side of the equation does not depend on the location on the surface (\mathbf{s}), this dependence can be removed from the left hand side of the equation.:

$$q(t) = h(t, T)(T(t) - T_{\text{env}}(t))$$

Apply the assumption that the heat transfer coefficient is independent of time (**A:HeatTransCoeffIndepTime**):

$$q(t) = h(T)(T(t) - T_{\text{env}}(t))$$

Apply the assumption that the heat transfer coefficient is independent of temperature (**A:HeatTransCoeffIndepTemp**):

$$q(t) = h(T(t) - T_{\text{env}}(t))$$

Refname	RT:htFluxWaterFromCoil
Label	Heat flux into the water from the coil
Units	$\frac{W}{m^2}$
Equation	$q_C(t) = h_C (T_C - T_W(t))$
Description	<p>q_C is the heat flux into the water from the coil ($\frac{W}{m^2}$)</p> <p>h_C is the convective heat transfer coefficient between coil and water ($\frac{W}{m^2 \cdot ^\circ C}$)</p> <p>$T_C$ is the temperature of the heating coil ($^\circ C$)</p> <p>T_W is the temperature of the water ($^\circ C$)</p> <p>t is the time (s)</p>
Notes	<p>q_C is found by assuming that Newton's law of cooling applies (A:Newton-Law-Convective-Cooling-Coil-Water).</p>
Source	[koothoor2013]
RefBy	FT:eBalanceOnWtr

Preconditions for **RT:htFluxWaterFromCoil:**

- **A:UniformHeatTransCoeffOverCoil**,
- **A:Temp-Heating-Coil-Constant-over-Length**,
- **A:UniformFullyMixed**,
- **A:HeatTransCoeffCoilIndepTime**,

- [A:HeatTransCoeffCoilIndepTemp](#),
- [A:Temp-Heating-Coil-Constant-over-Time](#)

Detailed derivation of heat flux into the water from the coil: We assume that Newton's law of cooling applies between the coil and the water ([A:Newton-Law-Convective-Cooling-Coil-Water](#)). To use the simplified form of Newton's law of cooling ([RT:nwtnCooling:ReduceDep](#)) we need to map the coil and water specific variables to the general form of the equation and we need to show that the pre-conditions are satisfied.

The beginning equation is from [RT:nwtnCooling:ReduceDep](#):

$$q(t) = h(T(t) - T_{\text{env}}(t))$$

We apply this equation with the surface as the surface of the coil in the tank of water. The environment is the tank of water and the body under consideration is the heating coil. Therefore, to apply Newton's law of cooling in the current context, we use the following mapping: $q(t) = q_C(t)$, $h = h_C$, $T(t) = T_C$ and $T_{\text{env}}(t) = T_W(t)$. With these substitutions we have the following equation:

$$q_C(t) = h_C (T_C(t) - T_W(t))$$

To use this equation, we need to show that the pre-conditions are satisfied by our system. We will cover each assumption from [RT:nwtnCooling:ReduceDep](#) in turn.

- We can use [A:UniformHeatTransCoeffOverSurf](#) because we make the assumption that the heat transfer coefficient over the surface of the coil is uniform ([A:UniformHeatTransCoeffOverCoil](#)).
- We can apply [A:UniformTempOverSurf](#) because of our assumption that the temperature of the coil is the same throughout its length ([A:Temp-Heating-Coil-Constant-over-Length](#)).
- We can apply the assumption that the environment temperature of the tank is uniform over the surface ([A:UniformEnviroTemp](#)) because we are assuming that the tank is fully mixed ([A:UniformFullyMixed](#)).
- We can apply [A:HeatTransCoeffIndepTime](#) because we assume that the heat transfer coefficient for the coil does not change over time ([A:HeatTransCoeffCoilIndepTime](#)).
- We can apply [A:HeatTransCoeffIndepTemp](#) because we assume for the coil that the heat transfer coefficient does not vary with the temperature ([A:HeatTransCoeffCoilIndepTemp](#)).

In addition to the above assumptions we make one more simplifying assumption. We assume that the temperature in the heating coil is constant over time ([A:Temp-Heating-Coil-Constant-over-Time](#)). Our rationale is that the simulations are for one heating event

where the conditions will not overly vary. With T_C now independent of temperature, the equation can be rewritten as:

$$q_C(t) = h_C (T_C - T_W(t))$$

5 Definitions

This section collects and defines all the data needed to build the instance models.

Refname	DD:waterMass
Label	Mass of water
Symbol	m_W
Units	kg
Equation	$m_W = V_W \rho_W$
Description	m_W is the mass of water (kg) V_W is the volume of water (m^3) ρ_W is the density of water ($\frac{\text{kg}}{\text{m}^3}$)
Notes	Density (ρ) is defined in BT:density
Source	—
RefBy	FR:Find-Mass

Refname	DD:waterVolume.nopcm
Label	Volume of water
Symbol	V_W
Units	m^3
Equation	$V_W = V_{\text{tank}}$
Description	V_W is the volume of water (m^3) V_{tank} is the volume of the cylindrical tank (m^3)
Notes	Based on A:Volume-Coil-Negligible . V_{tank} is defined in DD:tankVolume .
Source	—
RefBy	FR:Find-Mass

Refname	DD:tankVolume
Label	Volume of the cylindrical tank
Symbol	V_{tank}
Units	m^3
Equation	$V_{\text{tank}} = \pi \left(\frac{D}{2} \right)^2 L$
Description	<p>V_{tank} is the volume of the cylindrical tank (m^3)</p> <p>π is the ratio of circumference to diameter for any circle (Unitless)</p> <p>D is the diameter of tank (m)</p> <p>L is the length of tank (m)</p>
Source	—
RefBy	DD:waterVolume_nopcm and FR:Find-Mass

Refname	DD:balanceDecayRate
Label	ODE parameter for water related to decay time
Symbol	τ_W
Units	s
Equation	$\tau_W = \frac{m_W C_W}{h_C A_C}$
Description	τ_W is the ODE parameter for water related to decay time (s) m_W is the mass of water (kg) C_W is the specific heat capacity of water ($\frac{J}{kg^\circ C}$) h_C is the convective heat transfer coefficient between coil and water ($\frac{W}{m^2^\circ C}$) A_C is the heating coil surface area (m ²)
Source	[koothoor2013]
RefBy	FR:Output-Input-Derived-Values and FT:eBalanceOnWtr

6 Final Theories

This section transforms the problem defined in the problem description into one which is expressed in mathematical terms. It uses concrete symbols defined in the [data definitions](#) to replace the abstract symbols in the models identified in [theoretical models](#) and [general definitions](#).

The goal GS:Predict-Water-Temperature is met by [FT:eBalanceOnWtr](#) and the goal GS:Predict-Water-Energy is met by [FT:heatElInWtr](#).

Refname	FT:eBalanceOnWtr
Label	Energy balance on water to find the temperature of the water
Input	$T_C, T_{\text{init}}, t_{\text{final}}, A_C, h_C, C_W, m_W$
Output	T_W
Input Constraints	$T_C \geq T_{\text{init}}$
Output Constraints	
Equation	$\frac{dT_W(t)}{dt} = \frac{1}{\tau_W} (T_C - T_W(t))$
Description	<p>t is the time (s)</p> <p>T_W is the temperature of the water (°C)</p> <p>τ_W is the ODE parameter for water related to decay time (s)</p> <p>T_C is the temperature of the heating coil (°C)</p>
Notes	<p>[The initial temperature of the water in the tank is T_{init}; that is, $T_W(0) = T_{\text{init}}$. —SS] τ_W is calculated from DD:balanceDecayRate.</p> <p>The above equation applies as long as the water is in liquid form, $0 < T_W < 100$ (°C) where 0 (°C) and 100 (°C) are the melting and boiling point temperatures of water, respectively (A:Water-Always-Liquid).</p>
Source	[koothoor2013]
RefBy	UC:No-Internal-Heat-Generation, FR:Find-Mass, and FR:Calculate-Temperature-Water-Over-Time

Preconditions for FT:eBalanceOnWtr:

- A:Thermal-Energy-Only,
- A:UniformFullyMixed
- A:No-Internal-Heat-Generation-By-Water,
- A:Perfect-Insulation-Tank,
- A:DensityIndepTempWater,
- A:SpecHeatIndepTempWater,
- A:UniformHeatTransCoeffOverCoil,
- A:Temp-Heating-Coil-Constant-over-Length,
- A:HeatTransCoeffCoilIndepTime,
- A:HeatTransCoeffCoilIndepTemp

Detailed derivation of the energy balance on water: To derive the ODE that defines the temperature of the water over time, we start with the simplified rate of change of temperature equation (RT:rocTempSimp):

$$mC \frac{dT(t)}{dt} = - \sum_{i=0}^n q_i(t) A_i + g(t) V$$

The volume under consideration is the water tank. There are two surfaces ($n = 2$) that encompass the volume: the walls of the tank and the surface of the heating coil. We will represent the thermal flux through the tank walls by q_{tank} and the thermal flux through the heating coil by q_C . The respective surface areas are A_{tank} and A_C . To complete the application of RT:rocTempSimp to the water tank and coil system, we add the following symbols:

- m_W for the mass of the water in the tank, $m = m_W$.
- C_W for the specific heat capacity of the water, $C = C_W$.
- T_W for the temperature of the water in the tank, $T = T_W$.
- g_W for the per unit heat generation of the water in the tank, $g = g_W$.

Substituting the above variables into **RT:rocTempSimp**, we have:

$$m_W C_W \frac{dT_W(t)}{dt} = -q_{\text{tank}}(t) A_{\text{tank}} - q_C(t) A_C + g_W(t) V$$

Apply the assumption that there is no internal heat generation for the water ($g_W = 0$) (**A:No-Internal-Heat-Generation-By-Water**):

$$m_W C_W \frac{dT_W(t)}{dt} = -q_{\text{tank}}(t) A_{\text{tank}} - q_C(t) A_C$$

Apply the assumption that the tank is a perfect insulator ($q_{\text{tank}} = 0$) (**A:Perfect-Insulation-Tank**):

$$m_W C_W \frac{dT_W(t)}{dt} = -q_C(t) A_C$$

We find the value for q_C from (**RT:htFluxWaterFromCoil**): $q_C(t) = h_C (T_C - T_W(t))$. However before substitution into the energy balance equation, we need to change the sign of q_C because it is in the opposite direction to the unit outward normal for the surface. The thermal flux is from the coil to the water because of the assumption that the coil is warmer than the water tank ($T_C \geq T_{\text{init}}$). Therefore, the equation can be rewritten as:

$$m_W C_W \frac{dT_W(t)}{dt} = h_C (T_C - T_W(t)) A_C$$

Move A_C :

$$m_W C_W \frac{dT_W(t)}{dt} = h_C A_C (T_C - T_W(t))$$

Dividing both sides of the above equation by $m_W C_W$, we obtain:

$$\frac{dT_W(t)}{dt} = \frac{h_C A_C}{m_W C_W} (T_C - T_W(t))$$

By substituting τ_W (from **DD:balanceDecayRate**), this can be rewritten as:

$$\frac{dT_W(t)}{dt} = \frac{1}{\tau_W} (T_C - T_W(t))$$

To use this equation, we need to show that the pre-conditions from **RT:rocTempSimp** and **RT:htFluxWaterFromCoil** are satisfied.

- **A:Thermal-Energy-Only**: We will satisfy this assumption by making the same assumption.
- **A:UniformDensity-OverVol**: we make this assumption because we assume that the tank is fully mixed (**A:UniformFullyMixed**).

- **A:UniformSpecHeat-OverVol**: we make this assumption because we assume that the tank is fully mixed (**A:UniformFullyMixed**).
- **A:UniformTemp-OverVol**: we make this assumption because we assume that the tank is fully mixed (**A:UniformFullyMixed**).
- **A:UniformHeatGen-OverVol**: we make this assumption because we assume that heat generation for the water is zero over the volume (**A:No-Internal-Heat-Generation-By-Water**).
- **A:DensityIndepTemp**: we assume that for the water in the tank, and the temperature bounds of the simulation, the density does not change with temperature (**A:DensityIndepTempWater**). [We could cite actually data for this. —SS]
- **A:SpecHeatIndepTemp**: we assume that for the water in the tank, and the temperature bounds of the simulation, the specific heat capacity does not change with temperature (**A:SpecHeatIndepTempWater**). [We could cite actually data for this. —SS]
- **A:VolumeIsCompact**: The shape of the tank is compact, so this assumption is satisfied. [Do we need more here? —SS]
- **A:VolHasPiecewise-Smooth-Surface**: The shape of the surface of the tank is smooth, as is the surface of the heating coil. [Do we need more here? —SS]
- **A:ThermalFlux-UniformOverSurfaces**: This assumption is satisfied because of the following assumptions that are pre-conditions for **RT:htFluxWaterFromCoil**: **A:UniformHeatTransCoeffOverCoil**, **A:Temp-Heating-Coil-Constant-over-Length**, **A:UniformFullyMixed**.
- **A:UniformHeatTransCoeffOverCoil**: We will satisfy this assumption by making the same assumption.
- **A:Temp-Heating-Coil-Constant-over-Length**: We will satisfy this assumption by making the same assumption.
- **A:UniformFullyMixed**: We will satisfy this assumption by making the same assumption.
- **A:HeatTransCoeffCoilIndepTime**: We will satisfy this assumption by making the same assumption.
- **A:HeatTransCoeffCoilIndepTemp**: We will satisfy this assumption by making the same assumption.
- **A:Temp-Heating-Coil-Constant-over-Time**: We will satisfy this assumption by making the same assumption.

Refname	FT:heatEInWtr
Label	Heat energy in the water
Input	$T_{\text{init}}, m_{\text{W}}, C_{\text{W}}, m_{\text{W}}$
Output	E_{W}
Input Constraints	
Output Constraints	
Equation	$E_{\text{W}}(t) = C_{\text{W}}m_{\text{W}}(T_{\text{W}}(t) - T_{\text{init}})$
Description	<p>E_{W} is the change in heat energy in the water (J)</p> <p>t is the time (s)</p> <p>C_{W} is the specific heat capacity of water ($\frac{\text{J}}{\text{kg}^{\circ}\text{C}}$)</p> <p>$m_{\text{W}}$ is the mass of water (kg)</p> <p>T_{W} is the temperature of the water ($^{\circ}\text{C}$)</p> <p>T_{init} is the initial temperature ($^{\circ}\text{C}$)</p>
Notes	<p>The above equation is derived using BT:sensHtE.</p> <p>The change in temperature is the difference between the temperature at time t (s), T_{W} and the initial temperature, T_{init} ($^{\circ}\text{C}$).</p> <p>This equation applies as long as $0 < T_{\text{W}} < 100^{\circ}\text{C}$ (A:Water-Always-Liquid, A:Atmospheric-Pressure-Tank).</p>
Source	[koothoor2013]
RefBy	FR:Calculate-Change-Heat_Energy-Water-Over-Time

6.0.1 Data Constraints

The **Data Constraints Table** shows the data constraints on the input variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The column for software constraints restricts the range of inputs to reasonable values.

Var	Physical Constraints	Software Constraints	Typical Value	Uncert.
A_C	$A_C > 0$	$A_C \leq A_C^{\max}$	0.12 m ²	10%
C_W	$C_W > 0$	$C_W^{\min} < C_W < C_W^{\max}$	4186 $\frac{\text{J}}{\text{kg}^\circ\text{C}}$	10%
D	$D > 0$	$AR_{\min} \leq D \leq AR_{\max}$	0.412 m	10%
h_C	$h_C > 0$	$h_C^{\min} \leq h_C \leq h_C^{\max}$	1000 $\frac{\text{W}}{\text{m}^2^\circ\text{C}}$	10%
L	$L > 0$	$L_{\min} \leq L \leq L_{\max}$	1.5 m	10%
T_C	$0 < T_C < 100$	—	50 °C	10%
T_{init}	$0 < T_{\text{init}} < 100$	—	40 °C	10%
t_{final}	$t_{\text{final}} > 0$	$t_{\text{final}} < t_{\text{final}}^{\max}$	50000 s	10%
t_{step}	$0 < t_{\text{step}} < t_{\text{final}}$	—	0.01 s	10%
ρ_W	$\rho_W > 0$	$\rho_W^{\min} < \rho_W \leq \rho_W^{\max}$	1000 $\frac{\text{kg}}{\text{m}^3}$	10%

Table 1: Input Data Constraints

6.0.2 Properties of a Correct Solution

The **Data Constraints Table** shows the data constraints on the output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable.

Var	Physical Constraints
T_W	$T_{\text{init}} \leq T_W \leq T_C$
E_W	$E_W \geq 0$

Table 2: Output Data Constraints

7 Rationale

[We might want to bring in the goal of the simulation, likely related to the purpose section. Some purposes do not require high fidelity simulations. —SS]

- **A:UniformHeatTransCoeffOverCoil**: The rationale for this assumption is that heating coils are constructed so that the effective heat transfer coefficient will be uniform.
- **A:Temp-Heating-Coil-Constant-over-Length**: The rationale for this is that although the temperature of the coil will drop over its length, the change will not be large enough to matter because new heated water will continually be moved into the coil.
- **A:UniformFullyMixed**: Our rationale for assuming that the tank is fully mixed is that the temperature changes are relatively slow giving convective and conductive heat transfer enough time to even out the temperature in the tank. [Maybe the rationale should go with the assumption, since assumptions may be reused in other contexts. —SS] [Assumptions are sometimes made even though they are known to likely be inaccurate, but the scientist wants to explore the model. —SS]
- **A:HeatTransCoeffCoilIndepTime**: Our rationale for this assumption is that the physical characteristics of the heating coil are not going to change appreciably over the length of any given simulation.
- **A:HeatTransCoeffCoilIndepTemp**: Our rationale for this is that within the operating range of the coil the change in the heat transfer coefficient with temperature is relatively small.
- **A:Thermal-Energy-Only**
- **A:UniformFullyMixed**
- **A:No-Internal-Heat-Generation-By-Water**
- **A:Perfect-Insulation-Tank**
- **A:DensityIndepTempWater**
- **A:SpecHeatIndepTempWater**

8 Requirements

This section provides the functional requirements, the tasks and behaviours that the software is expected to complete, and the non-functional requirements, the qualities that the software is expected to exhibit.

8.1 Functional Requirements

This section provides the functional requirements, the tasks and behaviours that the software is expected to complete.

- Input-Initial-Values: Input the following values described in the table for **Required Inputs**, which define the tank parameters, material properties, and initial conditions.
- Find-Mass: Use the inputs in **FR:Input-Initial-Values** to find the mass needed for **FT:eBalanceOnWtr**, using **DD:waterMass**, **DD:waterVolume_nopcm**, and **DD:tankVolume**.
- Physical_Constraints: Verify that the inputs satisfy the required **physical constraints**.
- Output-Derived-Values: Output the input values and derived values in the following list: the values (from **FR:Input-Initial-Values**), the mass (from **FR:Find-Mass**), and τ_W (from **DD:balanceDecayRate**).
- Water-Over-Time: Calculate and output the temperature of the water ($T_W(t)$) over the simulation time (from **FT:eBalanceOnWtr**).
- Water-Over-Time: Calculate and output the change in heat energy in the water ($E_W(t)$) over the simulation time (from **FT:heatEInWtr**).

Symbol	Description	Units
A_C	Heating coil surface area	m^2
A_{tol}	Absolute tolerance	—
C_W	Specific heat capacity of water	$\frac{\text{J}}{\text{kg}^\circ\text{C}}$
D	Diameter of tank	m
h_C	Convective heat transfer coefficient between coil and water	$\frac{\text{W}}{\text{m}^2^\circ\text{C}}$
L	Length of tank	m
R_{tol}	Relative tolerance	—
T_C	Temperature of the heating coil	$^\circ\text{C}$
T_{init}	Initial temperature	$^\circ\text{C}$
t_{final}	Final time	s
t_{step}	Time step for simulation	s
ρ_W	Density of water	$\frac{\text{kg}}{\text{m}^3}$

Table 3: Required Inputs following **FR:Input-Initial-Values**

9 Likely Changes

This section lists the likely changes to be made to the software.

- Variable-Over-Day: **A:Temp-Heating-Coil-Constant-over-Time** - The temperature of the heating coil will change over the course of the day, depending on the energy received from the sun.
- Variable-Over-Length: **A:Temp-Heating-Coil-Constant-over-Length** - The temperature of the heating coil will actually change along its length as the water within it cools.
- Discharging-Tank: **A:Charging-Tank-No-Temp-Discharge** - The model currently only accounts for charging of the tank. That is, increasing the temperature of the water to match the temperature of the coil. A more complete model would also account for discharging of the tank.
- Tank-Lose-Heat: **A:Perfect-Insulation-Tank** - Any real tank cannot be perfectly insulated and will lose heat.

10 Unlikely Changes

This section lists the unlikely changes to be made to the software.

- Water-Fixed-States: **A:Water-Always-Liquid** - It is unlikely for the change of water from liquid to a solid, or from liquid to gas to be considered.
- l-Heat-Generation: **A:No-Internal-Heat-Generation-By-Water** - Is used for the derivations of **FT:eBalanceOnWtr**.