Created in COMSOL Multiphysics 5.6

Submarine Cable 6 — Thermal Effects

Introduction

This tutorial uses the *Inductive Effects* model from this series as a basis and adds thermal effects, including a temperature dependent conductivity (through linearized resistivity). It shows how to achieve a multiphysics coupling between electromagnetic fields and heat transfer, using the frequency-stationary study type (induction heating).

The influence of elevated temperatures on losses in the phases, screens, and armor is investigated (verification is included). The obtained temperature values are used in the *Inductive Effects 3D* tutorial, to apply a first-order temperature correction in 3D. Finally, the tutorial demonstrates how to match the resulting AC resistance to the one given by the IEC series of standards.

Model Definition

The geometry is the same as the one used in the *Inductive Effects* tutorial; see Figure 1. It describes a detailed cross section (as built in the *Introduction* tutorial). A large number of material properties is included for the metals, the polymers, and the sea bed.

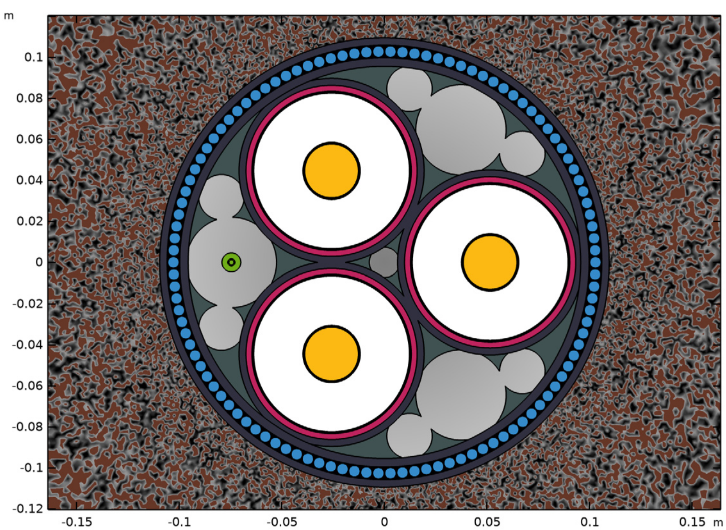


Figure 1: The cable's cross section, including the three phases (yellow), screens (red), the XLPE (white), the armor (blue), and the fiber (green).

THEORETICAL BASIS

In addition to Maxwell–Ampère's law (for which the theoretical basis has been discussed in the *Inductive Effects* tutorial), this 2D model solves the classical stationary heat transfer problem, known as Fourier's law. The following derivation uses the differential form, together with the SI unit system.

Fourier's Law

The derivation of Fourier's law starts with the notion that a temperature gradient ∇T in a thermal conductor leads to a flux of thermal energy:

$$\mathbf{q} = -\mathrm{k}\nabla T, \tag{1}$$

where $\mathbf{q}$ and k refer to the local heat flux density in $\mathrm{W/m^2}$ and the thermal conductivity in $\mathrm{W/(m{\cdot}K)}$, respectively. Secondly, the divergence criterion states that heat flux should be conserved; $\nabla \cdot \mathbf{q} = 0$. If the average divergence of $\mathbf{q}$ is *not* zero for a given volume, it means the sum of all heat entering or leaving that volume is not zero. In that case there is a heat source, or heat sink.

Electromagnetic Losses

In this model, the heat comes from electromagnetic losses. For the phases and the screens (copper and lead), magnetic and dielectric hysteresis losses have been neglected. What enters the heat equation is the resistive loss only (this is known as *Joule heating*):

$$Q_{\mathrm{rh}} = \frac{|\mathbf{J}|^2}{2\sigma} = \frac{\mathrm{Re}(\mathbf{J} \cdot \mathbf{E}^*)}{2}, \tag{2}$$

where $\mathbf{J}$ refers to the conduction current density (excluding displacement currents), and $\mathbf{E}^*$ refers to the complex conjugate of $\mathbf{E}$. For the armor, magnetic hysteresis losses are included as well. This is done by means of a *complex permeability*: The $\mathbf{B}$ and $\mathbf{H}$ fields will be slightly out-of-phase, resulting in a tilted ellipse on the B-H plane. The ellipse is not a fully accurate representation of the hysteresis loop at these frequencies, but it can be fitted to have the same surface area (to produce realistic loss values). The losses are then given by:

$$Q_{\mathrm{ml}} = \frac{\mathrm{Re}(\mathrm{j}\omega\mathbf{B} \cdot \mathbf{H}^*)}{2}. \tag{3}$$

Notice the similarity to the dielectric hysteresis loss term that appears when both the conduction currents and the displacements current are considered:

$$Q_{\mathrm{rh+dl}} = \frac{\mathrm{Re}(\mathbf{J}' \cdot \mathbf{E}^*)}{2} = \frac{\mathrm{Re}((\mathbf{J} + \mathrm{j}\omega\mathbf{D}) \cdot \mathbf{E}^*)}{2}, \tag{4}$$

where $\mathbf{J}'$ refers to the "total" current density, as introduced in the *Inductive Effects* tutorial. The imaginary term in front of **B** and **D** causes the losses to be zero, as long as **B** and **D** are in-phase with **H** and **E** (the energy is stored, rather than dissipated). This is true at low frequencies, and when no significant hysteresis effects are present. In addition to this, Equation 4 shows that in the frequency domain there is no clear mathematical distinction between resistive loss and dielectric hysteresis loss — resistive loss can be modeled using a *complex permittivity* — although the underlying cause is quite different.

Since Q_{rh} and Q_{ml} both represent the *cycle-average* or RMS loss (as opposed to the instantaneous loss — hence the division by two), we have implicitly assumed that the thermal response time is much larger than the cycle time used for the electric currents and the electromagnetic fields: $\tau_{ht} \gg 1/f_0 = T_0$, see On Frequency–Stationary Solving.

A Fully Coupled System

When substituting this together with the thermal flux definition (Equation 1) into the flux conservation law $\nabla \cdot \mathbf{q} = 0$, you get the following *2D partial differential equation* for the dependent variable T:

$$-\nabla \cdot (k\nabla T) = Q_h, \tag{5}$$

where Q_h refers to the total electromagnetic loss; $Q_h = Q_{rh} + Q_{ml}$. Note the resemblance to the current conservation law used in the *Capacitive Effects* model.

The Heat Transfer in Solids interface uses this conservation law to determine the value of T in the domains. Because the thermal conditions are assumed to be stationary, there exists a *thermal equilibrium* (no accumulation): The heat generated in the cable will have to leave through the external boundaries. The chosen external boundary condition is of the *Dirichlet* type; a constant predetermined temperature T_{ext}.

Lastly, the resulting temperature profile is fed back into the Magnetic Fields interface — into Maxwell–Ampère's law — by means of a temperature dependent conductivity[1]. Again, the thermal response time is assumed to be large; the temperature is assumed constant during the time-harmonic cycle:

$$-(\sigma(T) + j\omega\varepsilon)\mathbf{E} + \nabla \times (\mu^{-1}\mathbf{B}) = 0. \tag{6}$$

Together with Fourier's law (Equation 5), this gives a fully coupled system with a unique solution for both dependent variables; T and $\mathbf{A}$.

1. The magnetic material properties are assumed to be temperature-independent in this case, but this is by no means a restriction in the COMSOL Multiphysics software. You are free to choose any kind of dependency in your B(H) relation.

MODELING APPROACH

The tutorial starts by opening the model that results from the *Inductive Effects* tutorial, and restoring it to the point where it represents the *plain 2D* configuration. Then, a **Heat Transfer in Solids** physics interface is added, together with an **Electromagnetic Heating** multiphysics coupling. The frequency domain study is replaced by a **Frequency-Stationary** study, and a fixed temperature is used as an external boundary condition.

Although the initial rise in temperature seems plausible, the electromagnetic properties are virtually unchanged. This is because the conductivity of the most important materials (the metals) is not yet temperature dependent: The model effectively behaves like a one-way induction heating system. In a second step, a temperature dependence is included in the central conductors by means of a linearized resistivity[2]:

$$\sigma(T) = \frac{1}{\rho_0(1+\alpha(T-T_{\mathrm{ref}}))}, \tag{7}$$

where ρ_0 refers to the reference resistivity at T_{ref}, and α is the resistivity temperature coefficient in 1/K. With the linearized resistivity included, the losses and temperatures are analyzed once more. In a third step, the conductivity in the screens and the armor is made temperature dependent as well (if you are new to modeling inductive effects, the result may be counterintuitive). The resulting average temperatures in the phases, screens, and armor are evaluated, so they can be used in the *Inductive Effects 3D* tutorial.

Lastly, for the phases a conductivity σ_{coil} is found, such that the total effective phase AC resistance matches a certain specified value $R_{\mathrm{ac}}(T)$. Finding the conductivity is done by means of a global ODE (an ordinary differential equation). The expression for R_{ac} chosen in this case — as a proof of concept — is derived from Equation 7:

$$R_{\mathrm{ac}}(T) = \eta R_{\mathrm{dc}} = \frac{\eta}{\sigma(T)A} = \eta\rho_0(1+\alpha(T-T_{\mathrm{ref}}))/A, \tag{8}$$

where $R_{\mathrm{dc}} = 1/\sigma(T)A$ refers to the DC resistance per meter (the same as the one used in the *Inductive Effects* tutorial), and η refers to the ratio between the AC and DC resistance. In practice, however, for $R_{\mathrm{ac}}(T)$ any continuous, monotonic expression can be used, including ones based on measured data or coefficients provided by the IEC 60287 series of standards [1].

2. Note that, as this is COMSOL Multiphysics, you are free to choose any relation $\sigma(T)$, although not every relation will result in good convergence — or a (unique) solution for that matter. Having small or no higher-order terms in the relation $\sigma(T)$ will keep the computational effort low.

ON FREQUENCY–STATIONARY SOLVING

Induction heating is a classical example where two different time scales meet. On one hand, the electromagnetic response time, $\tau_{em} = RC$, is on the order of microseconds, and the cycle time used for the time-harmonic excitation of the electromagnetic problem is twenty milliseconds. On the other hand, the thermal response time is on the order of hours or days: From the perspective of the electromagnetic problem, the thermal response time is close to "infinite" (the thermal properties are constant during the cycle). From the thermal perspective, the electromagnetic phenomena are oscillating so quickly, that only cycle-average properties are perceived (such as the RMS value for the resistive losses).

Maxwell–Ampère's law — although in this case being a frequency domain formulation — is still a stationary problem from a mathematical viewpoint: It does not contain any time derivatives or time dependencies. The **Frequency–Stationary** study step may therefore use a single stationary solver to solve a fully coupled system, containing both Fourier's law and Maxwell–Ampère's law.

ON THIN RESISTIVE LAYERS

Certain geometrical features may be very thin, but still have a large impact on physical phenomena — like a contact interface or a corrosion layer. From a numerical viewpoint, it is very costly to model these features in full detail (that is; include them in the geometry and mesh). Typically, one includes them in the form of a *boundary feature*: An entity that is geometrically infinitely thin, yet from a physical viewpoint shows a finite thickness.

For the group of *Poisson-type* physics (of which Fourier's law is part), there are two archetypes; the highly conductive layer (a **Thermally thin approximation**) and the thin resistive layer (a **Thermally thick approximation**). For the thin resistive layer, the material interface reduces to a potential (or temperature) discontinuity. The resistance in the layer is assumed to be very large compared to the one in the neighboring domains. Therefore, the potential drop across the interface is very sharp when compared to:

- Potential gradients normal to the boundary, in neighboring domains.
- Potential gradients tangential to the boundary.
- Geometrical features tangential to the boundary (curvature and such).

First of all, this means that at the scale of the potential drop, conditions tangential to the boundary may be considered constant (leaving only the normal direction in which $\mathbf{q}$ flows). Secondly, it means that the layer may be considered infinitely thin, when observed from the scale of the geometry as a whole.

What remains is a local temperature at the upside of the layer, T_u, and a local temperature at the downside of the layer, T_d, both existing at the same coordinates. When these two temperatures differ, a heat flux will flow normal to the boundary:

$$\mathbf{n}\cdot\mathbf{q} = \frac{(T_u - T_d)}{R_s}, \tag{9}$$

where R_s refers to the layer's thermal resistivity. When the thickness d_s and the material properties k_s are known, R_s is given by d_s/k_s. In many real-world applications, however, finding a correct value for R_s is not a trivial thing; see section On Accuracy.

Results and Discussion

Initially, the electromagnetic properties of the cable are not affected by the elevated temperatures. This is because the most dominant material properties are still temperature independent, resulting in what is essentially a *one-way* induction heating model. The cable heats up, but the phase, screen, and armor losses remain unchanged with respect to the ones determined in the *Inductive Effects* tutorial (for the *plain 2D* configuration): They are still 47 kW/km, 13 kW/km, and 7.6 kW/km, respectively (see Table 1).

The phase temperature is about 81°C; an increase of 61°C with respect to T_{ref}. An educated guess based on $1 + \alpha(T - T_{ref})$ and Equation 2 suggests that when linearized resistivity is included in the phases, an increase in losses of at least 24% is to be expected. This prediction turns out to be spot on: The losses in the screens and the armor are not strongly affected, but the losses in the phases are about 58 kW/km now, an increase of 24% indeed.

TABLE 1: RESULTS FROM THE DIFFERENT HEATING CONFIGURATIONS COMPARED.

	One-Way IH Plain 2D	**Lin.Res. Phases**	**Coupled IH Plain 2D**	**Preset Rac(T)**
Phase Temperature (°C)	81	92	90	90
Screen Temperature (°C)	-	-	83	83
Armor Temperature (°C)	-	-	70	70
Phase Losses (kW/km)	47	58	58	58
Screen Losses (kW/km)	13	13	11	12
Armor Losses (kW/km)	7.6	7.7	6.8	7.0
Phase AC Resistance (mΩ/km)	53	-	59	59
Rac(T)/Rdc(T) Ratio (-)	1.57	-	1.39	1.39
Phase Inductance (mH/km)	0.42	-	0.43	0.43

However, at the same time the temperature has risen to 92°C (see Figure 2), which should actually have caused a 29% increase in losses. The explanation for this apparent discrepancy, is found by adding linearized resistivity to the screens and the armor as well.

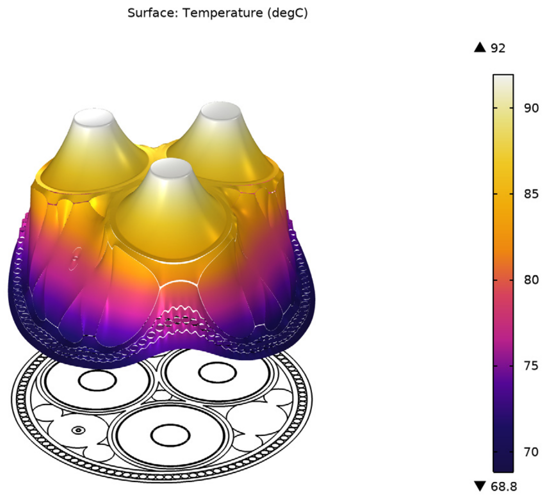


Figure 2: The temperature profile, with linearized resistivity applied in the phases only.

With the additional temperature dependence added, the configuration can be considered an actual *fully coupled* induction heating model[3]. Interestingly, this causes the losses in the screens and the armor to go *down* (see Table 1). The phase losses go down slightly, mainly because the phase temperature has lowered a bit, to 90°C; see Figure 3.

The reason for this, is that the currents in the screens and the armor are driven by the electromotive force (emf) produced by the currents in the phases. They are therefore *voltage-driven*. The phases themselves are current-driven[4]. The resistive losses scale with $|I|^2 R$ or, equivalently; $|V|^2/R$. Consequently, when the current is kept constant, the losses go up together with the resistance. When the voltage is kept constant, however, the opposite is true. Basically, the higher resistivity reduces parasitic effects — the skin depth becomes larger.

3. Note that the loss and the temperature dependence in the insulators are ignored. To investigate those, a model like the one in the *Capacitive Effects* tutorial would be more suitable.

4. The phase currents are based on the cable's official specifications (according to the IEC 60287 series of standards). They are preset, because the primary interest is in finding the maximum allowed continuous load current under normal operating conditions. The phase voltages are of less importance here, they are treated in more detail in the *Capacitive Effects* tutorial.

This also explains why the phase losses went up by only 24% in the second configuration, while the elevated phase temperature of 92°C implied a 29% increase: Only part of the phase loss is caused by current-driven phenomena, the rest of it is caused by parasitic effects (the skin- and proximity effects). For the screens a high conductivity is not even desired. On the contrary, they should conduct well enough to perform their duty as a screen — keeping the screen potential and the charging current losses close to zero[5] — but not much more than that.

Note: The two “mistakes” made in the previous text — expecting the phase losses to go up more than they should, and expecting the screen and armor losses to go up, while they should actually be going down — are caused by *stationary-electric reasoning*; forgetting about transient effects. Applying stationary-electric reasoning for approximations and basic understanding is common. Please be aware however, as frequencies go up, this reasoning loses validity. *A common pitfall is to apply static reasoning to dynamic phenomena without question.*

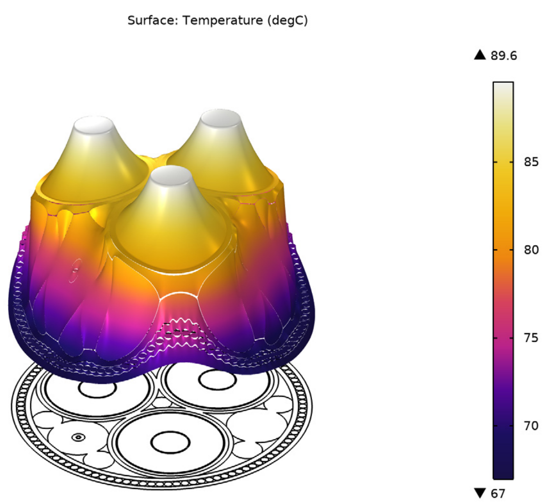


Figure 3: The temperature profile, with linearized resistivity applied in the phases, the screens, and the armor.

5. Please keep in mind that the charging currents are current-driven (as determined in the *Capacitive Effects* and *Bonding Capacitive* tutorials).

The error made by stationary-electric reasoning is nicely illustrated by η, the ratio between the AC resistance and the DC resistance: The DC resistance only reflects the stationary-Ohmic losses in the phases themselves. The AC resistance on the other hand, includes parasitic effects in the phases and the neighboring conductors too. As the temperature goes up, these secondary effects are suppressed; the value of η goes down (see Table 1).

For the fully coupled plain 2D configuration, some special care is taken to extract the phase, screen, and armor temperatures. These are used in the *Inductive Effects 3D* tutorial, to apply a first-order temperature correction to the materials: Linearized resistivity is used in 3D, based on the temperature profile from the 2D induction heating model.

Note: Second- or third-order temperature corrections are possible. Feeding the losses from the 3D model into a 2D thermal model, and coupling the resulting temperatures back to 3D even allows for a fully coupled, hybrid 2D/3D, induction heating model. In practice however, the first-order correction already covers 99% of the effect.

When switching to a preset AC resistance (as given by Equation 8), the results remain more or less the same. This does not say anything about the accuracy or validity of the model though; it only indicates that you have managed to correctly reverse the applied constraints (see section On Accuracy). The limited number of digits used for specifying η, and the modified conductor model used in the phases, is reflected by the results deviating from the third significant digit onward.

Finally, the phase resistance and phase inductance per kilometer evaluate to $59\ \mathrm{m\Omega}$ and $0.43\ \mathrm{mH}$, respectively. Compared to the plain 2D configuration at room temperature, the resistance has increased by about 13% (the inductive properties remain more or less the same). This increase is identical to the total increase in losses: If you evaluate the total electrical input power per meter, as given by $|I|^2 R_{ac}/2$, and compare the value to the total generated heat, you will see a good correspondence (up to 4–5 significant digits).

Apart from showing that the model correctly conserves energy, this confirms the statement that the AC resistance R_{ac} represents all losses, *including those occurring in the armor and the screens.*

ON ACCURACY

Using a Preset Resistance

For the fully coupled induction heating model, η is determined to be 1.39. Subsequently, that same value is used to get to the correct expression for the temperature-dependent AC resistance, Equation 8, under the same conditions. The fact that the results are similar does not tell you anything about the validity of the number 1.39 or the predictive value of the model in general.

It merely proves that the model is self-consistent and that the material properties of the phases can be set in several ways — based on conductivity measurements, resistance measurements, or the IEC reference coefficients, for example... After all, making the *value* of a preset resistance dictate the model's results also means making the *accuracy* of the resistance dictate the model's accuracy. This is true in the general case; a model is only as accurate as the data you put into it.

About Thin Resistive Layers

As stated in section On Thin Resistive Layers, finding a suitable value for the thermal resistivity of a material contact interface is not a trivial thing to do. As a proof of concept, in this tutorial a $20\ \mu m$ layer of air is used, neglecting effects such as contact roughness, pressure, and radiation.

Although for many applications a measured value or an educated guess may suffice, you might want to do some further investigation on this. If this topic is important to you, please investigate the tools provided by the *Heat Transfer Module*, including tutorial models such as the *Thermal Contact Resistance Between an Electronic Package and a Heat Sink* [3].

Modeling the Thermal Environment

One other important unknown is the *thermal environment*. While the cable itself is a well-known device with specified or to-be-determined properties, there is a general lack of knowledge when it comes to soil properties, ocean currents, or seasonal temperature variations (that is; as long as no specific location is chosen).

To include the environment is a challenge you will have to face for any model, regardless of the methods or the software used. When it comes to reproducing lab results, oftentimes you may enjoy a well-controlled environment (or at least a well-known one). Under real-world conditions, too, an educated guess may suffice. Examples of this are the ground and magnetic insulation conditions, chosen for the outer boundary in the *Capacitive Effects* and *Inductive Effects* tutorials. If the environment is not well-known, however, it is best, just to be pragmatic about it.

One option is to revert to official standards, as given by IEC, for example. They state precisely under what conditions a cable should be able to perform. They have been developed over many decades and take into account real-world conditions (including the necessary margins). For a manufacturer, meeting the standards is a way to communicate decency. For numerical modeling, it is a way to investigate *predictive value*.

Another option is to run the model for a wide range of external conditions, and check how sensitive it is to those conditions (that is, asking yourself whether you should worry about it in the first place). If you were to know precisely how the cable behaves for external temperatures varying from 20°C to 50°C, for example, the main thing left to do is to investigate what temperature actually occurs, and what margin of error may be expected.

For this, thermal numeric models can be used of course (2D or 3D), along with measurements. Assuming the behavior of the cable is precisely known for a range of temperatures, it can be included in the thermal model, simply as a temperature-dependent heat source — reducing the induction heating model to a local $Q_{\mathrm{h}}(T)$ lookup table. This greatly simplifies the thermal model and is also known as *submodeling*.

In the end, a good idea of what conditions will occur and what results are to be expected will depend on a decent understanding of the device's general behavior, preferably combined with experimental data from prototypes and literature. *Again, numerical models prove to be a perfect tool for achieving these goals.*

Reference

1. International Electrotechnical Commission, *Electric cables – Calculation of the Current Rating*; IEC 60287; IEC Press: Geneva, Switzerland, 2006.

2. J.C. del-Pino-López, M. Hatlo, and P. Cruz-Romero, "On Simplified 3D Finite Element Simulations of Three-Core Armored Power Cables," *Energies* 2018, 11, 3081.

3. Thermal Contact Resistance Between an Electronic Package and a Heat Sink, https://www.comsol.com/model/14659.

Application Library path: `ACDC_Module/Tutorials,_Cables/submarine_cable_06_thermal_effects`

Modeling Instructions

This tutorial will focus on thermal effects. The model is a modified version of the one resulting from the *Inductive Effects* tutorial (chapter 4). It is available in the Application Libraries folder as `submarine_cable_04_inductive_effects.mph`.

ROOT

You can start by opening this file and saving it under a new name.

1 From the **File** menu, choose **Open**.

2 Browse to the model's Application Libraries folder and double-click the file `submarine_cable_04_inductive_effects.mph`.

3 From the **File** menu, choose **Save As**.

4 Browse to a suitable folder and type the filename `submarine_cable_06_thermal_effects.mph`.

COMPONENT 1 (COMP1)

In the *Inductive Effects* tutorial, various twist configurations have been tested; *plain 2D*, *2.5D*, and *2.5D+milliken*. For a thermal analysis the plain 2D configuration seems most appropriate, as it does not underestimate the resistance and the losses as much as the other configurations do [2]. This will allow for an easier comparison with the results from the *Inductive Effects 3D* tutorial. What is more, we can use the thermal results obtained here as an estimate for the temperature profile in the 3D models (needed to apply a temperature correction).

Proceed by restoring the plain 2D configuration. That is; set the phases back to **Single conductor** and remove the coil group from the armor:

In the **Model Builder** window, expand the **Component 1 (comp1)** node.

MAGNETIC FIELDS (MF)

Phase 1

The settings window for the coil feature contains a lot of sections. For many of these, the default settings are sufficient. Collapse them to have a closer look at the important part; the **Coil** section.

1 In the **Model Builder** window, expand the **Component 1 (comp1)>Magnetic Fields (mf)** node, then click **Phase 1**.

2 Click to collapse the **Material Type** section, the **Coordinate System Selection** section, and the **Constitutive Relation** sections.

Next, continue by setting the conductor model.

3 In the **Settings** window for **Coil**, locate the **Coil** section.

4 From the **Conductor model** list, choose **Single conductor**.

Phase 2, Phase 3

Repeat this step for **Phase 2**, and **Phase 3**.

Cable Armor

In the **Model Builder** window, right-click **Cable Armor** and choose **Delete**.

GLOBAL DEFINITIONS

Some parameters have been prepared for doing the thermal analysis. You can load them from a file.

Thermal Parameters

1 In the **Home** toolbar, click **Pi** **Parameters** and choose **Add>Parameters**.

2 In the **Settings** window for **Parameters**, type `Thermal Parameters` in the **Label** text field.

3 Locate the **Parameters** section. Click **Load from File**.

4 Browse to the model's Application Libraries folder and double-click the file `submarine_cable_d_therm_parameters.txt`.

Twelve new parameters have been added. The last seven are used for temperature dependent material properties and `Tmext` refers to the external temperature.

The parameters `Tmcon`, `Tmpbs`, and `Tmarm` may require a further explanation. They are the expected average temperatures for the phases, screens and armor, respectively. They have been obtained from a model like this one, and will be validated later on. Their main purpose is to provide a good temperature estimate in 3D models with linearized resistivity, without actually having to solve a fully coupled 3D induction heating model (for more on this, see the *Inductive Effects 3D* tutorial).

Furthermore, `Ntcon` is used for the thermal properties of the stranded phase conductors (the model uses an *effective material*). Depending on the kind of insulation (if any) between the strands, the anisotropic structure may significantly influence the thermal conductivity. The thermal energy will have to cross barriers of insulating material in order to escape in the radial direction.

A thermal 2D model that resolves the strands in detail will show you the cross sectional thermal resistivity in the core may increase up to 13 times for certain types of insulation (as compared to solid copper). However, the same model shows that the resulting temperature increase is rather small (about 0.1°C). As it turns out, the temperature in the core is dictated by the properties of the thermal insulators surrounding it, rather than the thermal conductivity in the core itself. Even so, now that a good approximation of the value `Ntcon` is known, we might just as well make use of it.

Let us proceed by adding the **Heat Transfer in Solids** physics interface, and a multiphysics coupling:

ADD PHYSICS

1 In the **Home** toolbar, click **Add Physics** to open the **Add Physics** window.

2 Go to the **Add Physics** window.

3 In the tree, select **Heat Transfer>Heat Transfer in Solids (ht)**.

4 Click **Add to Component 1** in the window toolbar.

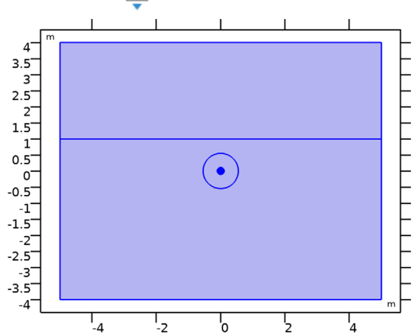
5 In the **Home** toolbar, click **Add Physics** to close the **Add Physics** window.

HEAT TRANSFER IN SOLIDS (HT)

1 In the **Settings** window for **Heat Transfer in Solids**, locate the **Domain Selection** section.

2 From the **Selection** list, choose **Thermal Domains**.

3 Click the **Zoom to Selection** button in the **Graphics** toolbar.



MULTIPHYSICS

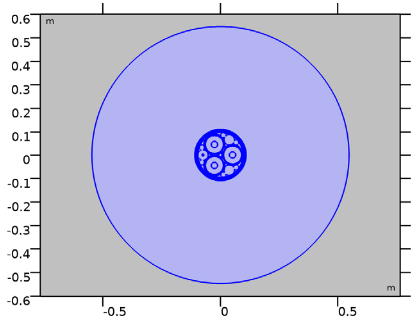
Electromagnetic Heating 1 (emh1)

1 In the **Physics** toolbar, click **Multiphysics Couplings** and choose **Domain> Electromagnetic Heating**.

2 In the **Settings** window for **Electromagnetic Heating**, locate the **Domain Selection** section.

3 From the **Selection** list, choose **Electromagnetic Domains**.

4 Click the **Zoom to Selection** button in the **Graphics** toolbar.



5 Locate the **Boundary Selection** section. Click **Clear Selection**.

The **Electromagnetic Heating** multiphysics coupling sees to it that electromagnetic losses result in generation of heat. Furthermore, it makes temperature dependent properties in the Magnetic Fields interface susceptible to the temperature field established by the Heat Transfer in Solids interface (see section Theoretical Basis).

RESULTS

Before adding a new study, you can remove the old one. That will automatically clean unnecessary clutter from the **Results** part of the model. This is because removing **Study 1**, *will also remove everything referring to it* — either directly or indirectly (plot groups, domain integrals, ... and such). The items in **Derived Values** you want to keep, however. To this end, you should decouple them from the dataset.

Phase Losses

1 In the **Model Builder** window, expand the **Results>Derived Values** node, then click **Phase Losses**.

2 In the **Settings** window for **Surface Integration**, locate the **Data** section.

3 From the **Dataset** list, choose **None**.

Screen and Armor Losses, Phase AC Resistance and Inductance

Repeat these steps for **Screen Losses**, **Armor Losses**, **Phase AC Resistance**, and **Phase Inductance**.

Next, remove the existing study and create a new one.

STUDY 1

In the **Model Builder** window, right-click **Study 1** and choose **Delete**.

ADD STUDY

1 In the **Home** toolbar, click **Add Study** to open the **Add Study** window.

2 Go to the **Add Study** window.

3 Find the **Studies** subsection. In the **Select Study** tree, select **Preset Studies for Selected Multiphysics>Frequency-Stationary**.

4 Right-click and choose **Add Study**.

5 In the **Home** toolbar, click **Add Study** to close the **Add Study** window.

STUDY 1

Step 1: Frequency-Stationary

1 In the **Settings** window for **Frequency-Stationary**, locate the **Study Settings** section.

2 In the **Frequency** text field, type `f0`.

The frequency-stationary study type is very well-suited for electromagnetic heating, see section On Frequency–Stationary Solving. The electromagnetic part is done in the frequency domain. Cycle-average (RMS) values for the losses are used as a heat source in the thermal part. The thermal part assumes stationary conditions — depending on how you look at it; a thermal equilibrium, large thermal inertia, or *infinite relaxation time*.

MATERIALS

Most of the material properties have already been prepared in the *Introduction* tutorial and the *Inductive Effects* tutorial. The effective thermal conductivity of the main conductors has not yet been set however, as it depends on the parameter `Ntcon`.

Copper (mat11)

1 In the **Model Builder** window, expand the **Component 1 (comp1)>Materials** node, then click **Copper (mat11)**.

2 In the **Settings** window for **Material**, locate the **Material Contents** section.

3 In the table, enter the following settings:

Property	Variable	Value	Unit
Thermal conductivity	k_iso ; kii = k_iso, kij = 0	`Ntcon*400[W/(m*K)]`	W/(m·K)

In case you are wondering; *the material properties for linearized resistivity will be discussed later*.

In order to model the thermal contact resistance between the cable's components, a **Thin Layer** boundary feature will be used. For this, a boundary material is added.

Thermal Contact Layer

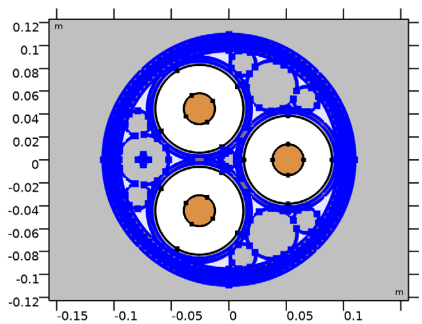
1 In the **Model Builder** window, under **Component 1 (comp1)>Materials** right-click **Air (mat1)** and choose **Duplicate**.

2 In the **Settings** window for **Material**, type `Thermal Contact Layer` in the **Label** text field.

3 Locate the **Geometric Entity Selection** section. From the **Geometric entity level** list, choose **Boundary**.

4 From the **Selection** list, choose **Thermal Contact**.

5 Click the **Zoom to Selection** button in the **Graphics** toolbar.



Notice that for the air, some of the material properties are given by a function, rather than a constant. You can find and plot these functions by expanding the material's node in the **Model Builder** tree.

Please be aware that in the end, many of the material properties used in this model will be either ignored, overridden or proven to be insignificant (as seen in the *Capacitive* and *Inductive Effects* tutorials). In case of this particular study type for example, the density and the heat capacity are not strictly needed (as transient thermal effects are neglected). For the sake of completeness, they are listed anyway.

Finally, observe that there is no such thing as "*the thermal conductivity of the sea bed*". The thermal conductivity will strongly depend on soil properties; local measurements are therefore important. For a further reflection on this, see section Modeling the Thermal Environment. Without referring to any specific location, the best thing we currently have is the sea bed thermal resistivity used when determining continuous current ratings according to the IEC 60287 series of standards [1].

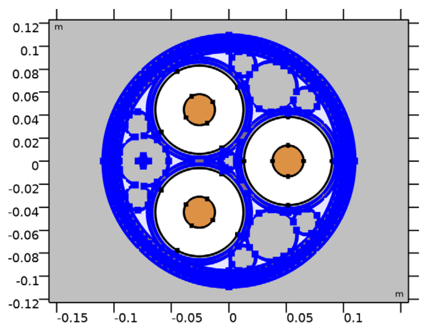
Now that the materials have been set and double-checked, let us have a look at the physics.

HEAT TRANSFER IN SOLIDS (HT)

In the **Model Builder** window, under **Component 1 (comp1)** click **Heat Transfer in Solids (ht)**.

Thin Layer 1

1. In the **Physics** toolbar, click **Boundaries** and choose **Thin Layer**.
2. In the **Settings** window for **Thin Layer**, locate the **Boundary Selection** section.
3. From the **Selection** list, choose **Thermal Contact**.



4. Locate the **Shell Properties** section. From the **Shell type** list, choose **Nonlayered shell**. In the L_{th} text field, type `20[um]`.

   Here, we intend to approximate a *thermal contact resistance* by adding a 20 µm thick layer of air (the **Thermally thick approximation** is valid for thin, highly resistive layers).

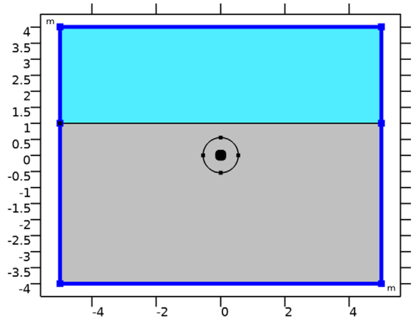
Admittedly, this is a bit of a crude way to model a contact resistance. When you have access to the *Heat Transfer Module*, much better would be to use the **Thermal Contact** boundary condition, and do some investigation on where it is applicable, and to what extent.

As thermal contact is a sophisticated phenomenon — of which the details lie outside the scope of this tutorial series — we will do with an approximation for now. The important thing to remember, is that one should not underestimate this phenomenon. For more details on this, see sections On Thin Resistive Layers, and About Thin Resistive Layers.

Temperature 1

1. In the **Physics** toolbar, click **Boundaries** and choose **Temperature**.
2. Click the **Zoom Extents** button in the **Graphics** toolbar.

3 Select Boundaries 1–3, 5, 10, and 11 only.



4 In the **Settings** window for **Temperature**, locate the **Temperature** section.

5 In the T_0 text field, type `Tmext`.

As you might have noticed, from a thermal viewpoint the model is rather simple. Effects like convection (buoyancy, ocean currents) are not included here. For more information on heat transfer and flow modeling, please have a look at one of the tutorial models in the COMSOL *Application Gallery* specifically addressing those topics. Proceed by disabling the default plots and computing the solution.

STUDY 1

1 In the **Model Builder** window, click **Study 1**.

2 In the **Settings** window for **Study**, locate the **Study Settings** section.

3 Clear the **Generate default plots** check box.

4 In the **Home** toolbar, click **Compute**.

RESULTS

Magnetic Flux Density Norm (mf)

1 In the **Home** toolbar, click **Add Plot Group** and choose **2D Plot Group**.

2 In the **Settings** window for **2D Plot Group**, type `Magnetic Flux Density Norm (mf)` in the **Label** text field.

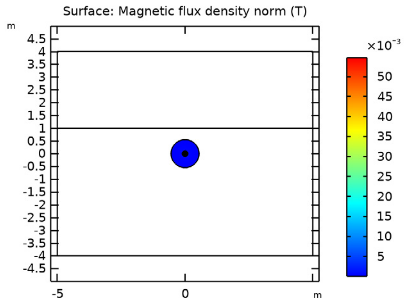
Surface 1

1 Right-click **Magnetic Flux Density Norm (mf)** and choose **Surface**.

2 In the **Settings** window for **Surface**, locate the **Coloring and Style** section.

3 From the **Color table** list, choose **RainbowLight**.

4 In the **Magnetic Flux Density Norm (mf)** toolbar, click **Plot**.



The **Magnetic Flux Density Norm** plot contains a lot of empty space, since the important bit (the cable) comprises only a small part of the total model. Let us focus on the cable by applying a selection to the solution.

Study 1/Solution 1 (sol1)

In the **Model Builder** window, expand the **Results>Datasets** node, then click **Study 1/ Solution 1 (sol1)**.

Selection

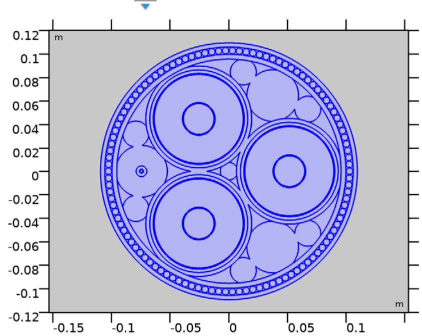
1 In the **Results** toolbar, click **Attributes** and choose **Selection**.

2 In the **Settings** window for **Selection**, locate the **Geometric Entity Selection** section.

3 From the **Geometric entity level** list, choose **Domain**.

4 From the **Selection** list, choose **Cable Domains**.

5 Click the **Zoom to Selection** button in the **Graphics** toolbar.



Magnetic Flux Density Norm (mf)

1 In the **Model Builder** window, click **Magnetic Flux Density Norm (mf)**.

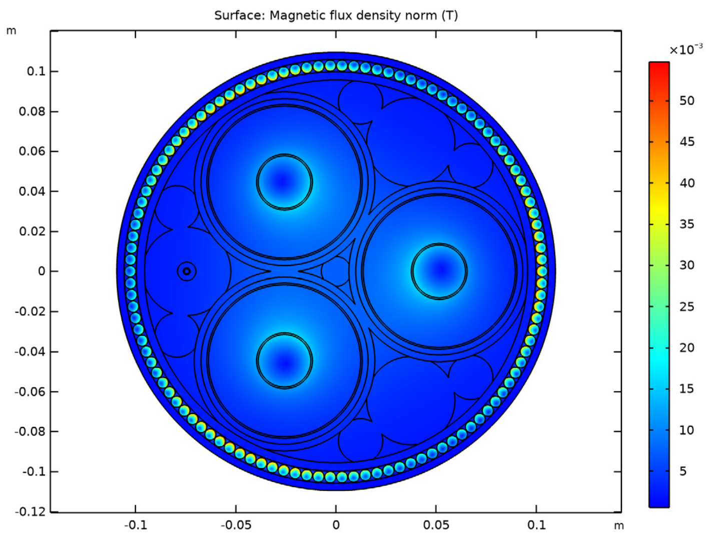
The second thing to notice is that the plot is zoomed-in quite a bit. This is because it is still locked to the camera settings used in the geometry and the mesh. You can give it separate view settings.

2 In the **Settings** window for **2D Plot Group**, locate the **Plot Settings** section.

3 From the **View** list, choose **View 2D 2**.

4 In the **Magnetic Flux Density Norm (mf)** toolbar, click **Plot**.

5 Click the **Zoom Extents** button in the **Graphics** toolbar.



The plot looks identical to the one we had initially in the *Inductive Effects* tutorial. We will not spend more time on it here. Next, proceed by investigating the temperatures.

Temperature (ht)

1 In the **Home** toolbar, click **Add Plot Group** and choose **2D Plot Group**.

2 In the **Settings** window for **2D Plot Group**, type `Temperature (ht)` in the **Label** text field.

3 Locate the **Plot Settings** section. From the **View** list, choose **View 2D 2**.

4 Locate the **Color Legend** section. Select the **Show maximum and minimum values** check box.

Surface 1

1 Right-click **Temperature (ht)** and choose **Surface**.

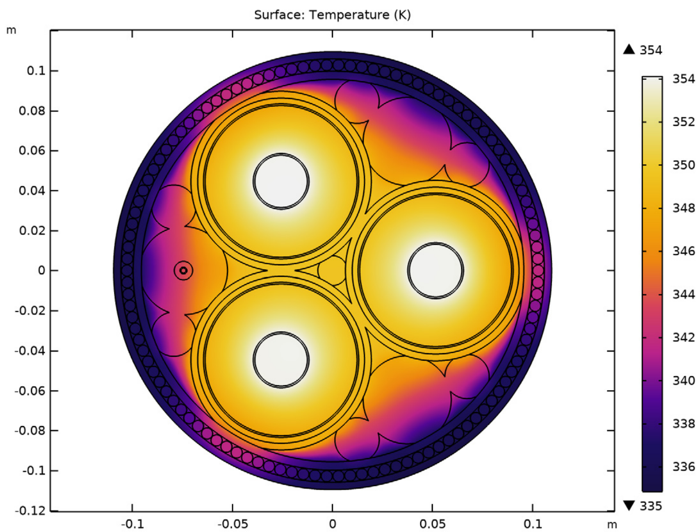
2 In the **Settings** window for **Surface**, locate the **Expression** section.

3 In the **Expression** text field, type `T`.

4 Locate the **Coloring and Style** section. From the **Color table** list, choose **HeatCameraLight**.

5 In the **Temperature (ht)** toolbar, click **Plot**.

6 Click the **Zoom Extents** button in the **Graphics** toolbar.



This is a good start. Let us make it more insightful (that is, *more fancy*).

7 Locate the **Expression** section. From the **Unit** list, choose **degC**.

8 Click to expand the **Quality** section. From the **Resolution** list, choose **Fine**.

Height Expression 1

1 Right-click **Surface 1** and choose **Height Expression**.

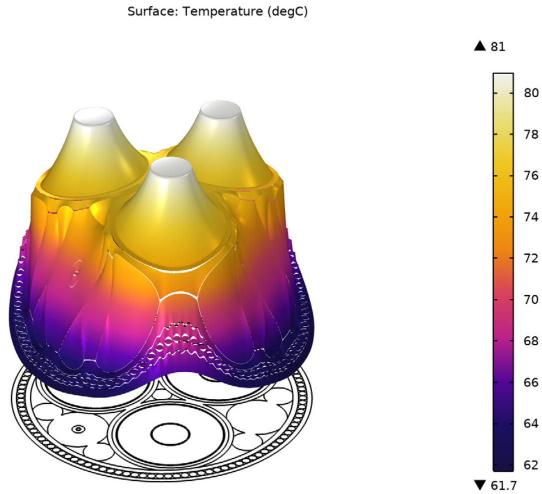
2 In the **Settings** window for **Height Expression**, locate the **Expression** section.

3 From the **Height data** list, choose **Expression**.

4 In the **Expression** text field, type `T-55[degC]`.

5 In the **Temperature (ht)** toolbar, click **Plot**.

6 Click the **Go to Default View** button in the **Graphics** toolbar.



This shape shows some similarities with the **Electric Potential** plot in the *Capacitive Effects* tutorial, in the sense that the metals tend to behave like equipotentials.

The **Thin Layer** feature creates small slits (discontinuities) in the temperature field, about 0.2–0.5°C in size. This is because it is a finite thermal resistor, while the boundary representing the contact interface in the geometry is infinitely thin. It is a classical boundary condition for diffusion (or, *Poisson-type*) physics, see section On Thin Resistive Layers.

The poorest thermal conductor seems to be the air. Furthermore, we see the maximum temperature (about 81°C), is not unreasonable: These cables can be loaded continuously up to temperatures of about 90°C. You can continue by checking the effects of this temperature raise on the electromagnetic properties of the cable.

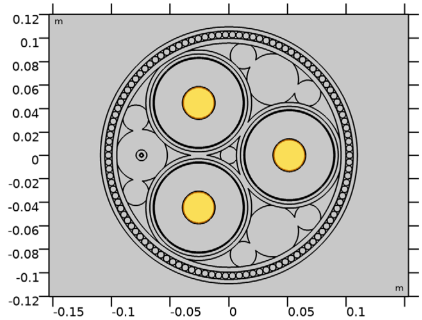
Phase Losses

1 In the **Model Builder** window, click **Phase Losses**.

2 In the **Settings** window for **Surface Integration**, locate the **Data** section.

3 From the **Dataset** list, choose **Study 1/Solution 1 (sol1)**.

4 Locate the **Selection** section. From the **Selection** list, choose **Phases**.



5 Locate the **Expressions** section. In the table, enter the following settings. That is; replace "`W/m`" (the default) with "`W/km`", and "`2.5D+milliken`" with "`one-way ih`":

Expression	**Unit**	**Description**
`mf.Qh`	`W/km`	`Phase losses (one-way ih)`

6 Click **Evaluate**.

TABLE

1 Go to the **Table** window.

The result should be about 47 kW/km.

RESULTS

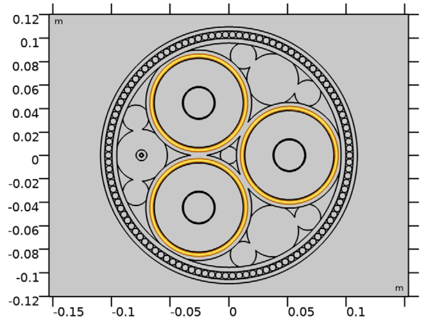
Screen Losses

1 In the **Model Builder** window, under **Results>Derived Values** click **Screen Losses**.

2 In the **Settings** window for **Surface Integration**, locate the **Data** section.

3 From the **Dataset** list, choose **Study 1/Solution 1 (sol1)**.

4 Locate the **Selection** section. From the **Selection** list, choose **Screens**.



5 Locate the **Expressions** section. In the table, enter the following settings:

Expression	**Unit**	**Description**
`mf.Qh`	`W/km`	`Screen losses (one-way ih)`

6 Click **Evaluate**.

TABLE

1 Go to the **Table** window.

The result should be about 13 kW/km.

RESULTS

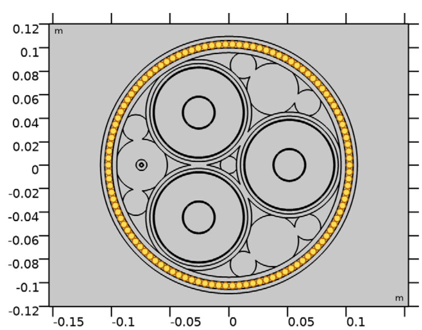
Armor Losses

1 In the **Model Builder** window, under **Results>Derived Values** click **Armor Losses**.

2 In the **Settings** window for **Surface Integration**, locate the **Data** section.

3 From the **Dataset** list, choose **Study 1/Solution 1 (sol1)**.

4 Locate the **Selection** section. From the **Selection** list, choose **Cable Armor**.



5 Locate the **Expressions** section. In the table, enter the following settings:

Expression	Unit	Description
`mf.Qh`	`W/km`	`Armor losses (one-way ih)`

6 Click **Evaluate**.

TABLE

1 Go to the **Table** window.

This should be about 7.6 kW/km.

Notice that these values are virtually identical to the ones in the *Inductive Effects* tutorial (for the *plain 2D* configuration): The electromagnetic part has not been affected at all by the raise in temperature. This is because for the most dominant lossy domains (the metals), the model does not yet include temperature dependent material properties. Before implementing this, let us first have a look at the thermal parameters and do some reflection on what should happen.

Modeling Instructions — Linearized Resistivity (Phases)

GLOBAL DEFINITIONS

Thermal Parameters

The temperature coefficient for copper and lead is about 0.004 1/K. This means the raise in temperature of about 61°C (with respect to `Tmref`) should lead to an increase in resistivity of about 24%, as given by Equation 7. Considering loss due to joule heating scales with I^2R and assuming the current is constant, the losses in our model should increase by 24% as well.

Proceed by reproducing this result. To this end, you can introduce a temperature dependence for the three main conductors.

MAGNETIC FIELDS (MF)

Phase 1

1 In the **Model Builder** window, under **Component 1 (comp1)>Magnetic Fields (mf)** click **Phase 1**.

2 In the **Settings** window for **Coil**, locate the **Constitutive Relation Jc-E** section.

3 From the **Conduction model** list, choose **Linearized resistivity**.

Phase 2, Phase 3

Repeat these steps for **Phase 2**, and **Phase 3**.

MATERIALS

Now, you will see that COMSOL starts detecting missing material properties. The properties that should be added are listed in the following table. Please check all of them for the correct value, even the ones that are already filled in. Note that for cases like this, *a convenient option is to copy-paste the values directly from this *.pdf file to COMSOL.*

1 In the **Model Builder** window, under **Component 1 (comp1)>Materials**, add the following material properties:

	Label	rho0 [ohm*m]	alpha [1/K]	Tref [K]
mat11	`Copper`	`R0cup/Ncon`	`ALcup`	`Tmref`
mat12	`Lead`	`R0pbs`	`ALpbs`	`Tmref`
mat13	`Galvanized steel`	`R0arm`	`ALarm`	`Tmref`

The reference resistivity for copper is divided by `Ncon`. This is because the phase conductors consist of compacted strands, rather than solid copper. For more information on this, see the *Inductive Effects* tutorial.

Although the material data listed for the lead and steel is not yet used at this point, you can fill them in anyway. They will become useful later on. Now, let us check the results.

STUDY 1

In the **Home** toolbar, click **Compute**.

RESULTS

Phase Losses

1 In the **Model Builder** window, under **Results>Derived Values** click **Phase Losses**.

2 In the **Settings** window for **Surface Integration**, locate the **Expressions** section.

3 In the table, update the description. Type `Phase losses (linres phases)`, that is; replace “`one-way ih`” with “`linres phases`”.

4 In the **Settings** window for **Surface Integration**, click **Evaluate**.

Screen Losses, Armor Losses

Repeat these steps for **Screen Losses** and **Armor Losses**.

TABLE

1 Go to the **Table** window.

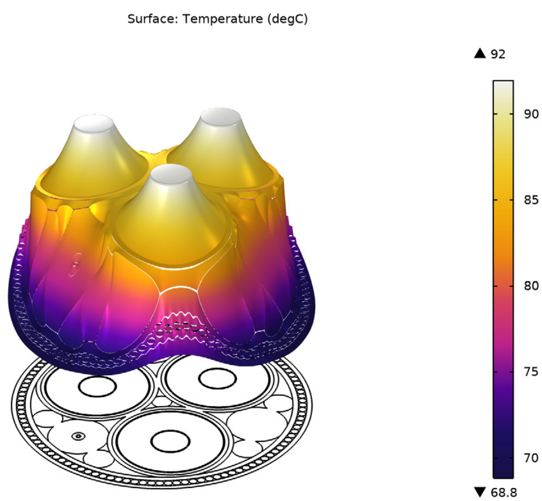
The losses per kilometer should still be about 13 kW and 7.7 kW for the screens and the armor respectively (virtually unchanged). The phase losses however, should be about 58 kW/km now, an increase of 24%, as predicted.

Temperature (ht)

1 In the **Model Builder** window, click **Temperature (ht)**.

2 In the **Temperature (ht)** toolbar, click **Plot**.

3 Click the **Go to Default View** button in the **Graphics** toolbar.



Getting an increase in losses equal to our crude initial guess may look great at first, but when you think about it, it does raise questions. As you can see, due to the additional losses the temperature has risen further; from 81°C to 92°C. With that kind of temperature raise, the increase in losses should actually have been `(92-20)*0.004`, which evaluates to 29%. Let us investigate further, by adding linearized resistivity to the screens and armor as well.

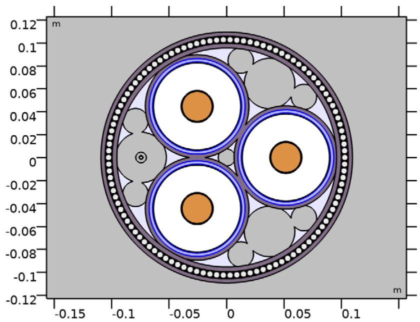
MAGNETIC FIELDS (MF)

Screens

1 In the **Physics** toolbar, click **Domains** and choose **Ampère's Law**.

2 In the **Settings** window for **Ampère's Law**, type `Screens` in the **Label** text field.

3 Locate the **Domain Selection** section. From the **Selection** list, choose **Screens**.



The settings window for the **Ampère's Law** feature contains a lot of sections. For many of these, the default settings are sufficient. Collapse them to have a closer look at the important part; the **Constitutive Relation Jc-E** section.

4 Click to collapse the **Material Type** section, the **Coordinate System Selection** section, the **Constitutive Relation B-H** section, and the **Constitutive Relation D-E** section.

Next, proceed by setting the conduction model.

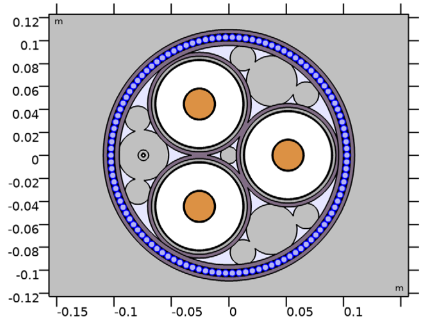
5 Locate the **Constitutive Relation Jc-E** section. From the **Conduction model** list, choose **Linearized resistivity**.

Cable Armor

1 Right-click **Screens** and choose **Duplicate**.

2 In the **Settings** window for **Ampère's Law**, type `Cable Armor` in the **Label** text field.

3 Locate the **Domain Selection** section. From the **Selection** list, choose **Cable Armor**.



With elevated temperatures and increased resistance in both the phases, screens, and armor, surely you would expect even more losses and even higher temperatures. Let us see if this reasoning is correct.

STUDY 1

In the **Home** toolbar, click **Compute**.

RESULTS

Phase Losses

1 In the **Model Builder** window, under **Results>Derived Values** click **Phase Losses**.

2 In the **Settings** window for **Surface Integration**, locate the **Expressions** section.

3 In the table, update the description. Type `Phase losses (coupled ih)`, that is; replace "`linres phases`" with "`coupled ih`".

4 In the **Settings** window for **Surface Integration**, click **Evaluate**.

Screen Losses, Armor Losses

Repeat these steps for **Screen Losses** and **Armor Losses**.

TABLE

1 Go to the **Table** window.

As opposed to what you might have expected, the losses in the screens and armor actually went *down* (by about 14–15% and 11–13% respectively). The phase losses went down as well, albeit slightly. This is because the currents in the screens and armor are induced by the electromotive force (emf) coming from the currents in the central conductors. The currents in the screens and armor are therefore *voltage driven*: When the resistance goes up, the currents and losses go down.

With this new insight, we are also able to explain why our initial guess for the increase in losses was spot on (24%). In the phase conductors there are two competing effects at work: On one hand the increase in losses causes a further rise in temperature, causing more losses and so on. On the other hand, the higher resistivity will suppress voltage driven induction currents. Basically, it means *parasitic effects* are reduced (the skin depth becomes larger).

In our initial guess we used *stationary-electric reasoning*; forgetting about transient effects. For R_{dc} it would have worked perfectly, and for homogenized multiturn coils too (see section *On Coil Domains* in the *Inductive Effects* tutorial). For this case it just happened to "work", because we forgot about the additional raise in temperature as well.

Applying stationary-electric reasoning for approximations and basic understanding is common. Please be aware however, as frequencies go up, this reasoning loses validity. *A common pitfall is to apply static reasoning to dynamic phenomena without question.* Note for example that the ratio between the AC resistance and the DC resistance η should decrease, as the temperature increases. Let us investigate.

Phase AC Resistance

1 In the **Model Builder** window, click **Phase AC Resistance**.
2 In the **Settings** window for **Global Evaluation**, locate the **Data** section.
3 From the **Dataset** list, choose **Study 1/Solution 1 (sol1)**.
4 Locate the **Expressions** section. In the table, enter the following settings. That is; replace "`Ω/m`" (the default) with "`mohm/km`", replace "`2.5D+milliken`" with "`coupled ih`", and rewrite the row for `Rcon` altogether:

Expression	**Unit**	**Description**
`(mf.RCoil_1/1[m]+mf.RCoil_2/1[m]+ mf.RCoil_3/1[m])/3`	`mohm/km`	`Phase AC resistance (coupled ih)`
`Rcon*(1+ALcup*(Tmcon-Tmref))`	`mohm/km`	`Main conductor DC resistance per phase, at 90°C (analytic)`

5 Click **Evaluate**.

Phase Inductance

1 In the **Model Builder** window, click **Phase Inductance**.
2 In the **Settings** window for **Global Evaluation**, locate the **Data** section.
3 From the **Dataset** list, choose **Study 1/Solution 1 (sol1)**.

4 Locate the **Expressions** section. In the table, enter the following settings. That is; replace "`H/m`" (the default) with "`mH/km`", and "`2.5D+milliken`" with "`coupled ih`":

Expression	Unit	Description
`(mf.LCoil_1/1[m]+mf.LCoil_2/1[m]+ mf.LCoil_3/1[m])/3`	`mH/km`	`Phase inductance (coupled ih)`

5 Click **Evaluate**.

6 Go to the **Table** window.

The phase AC resistance per kilometer for the fully coupled induction heating model should be about 59 mΩ. The AC/DC ratio η for 20°C is given by dividing the table columns **AC resistance, plain 2D**, and **DC resistance, 20°C**, which should evaluate to about 1.57. At 90°C, this ratio (**AC resistance, coupled ih** divided by **DC resistance, 90°C**) has been reduced to 1.39, confirming our reasoning. Finally, the inductance per kilometer should be about 0.43 mH — the increased resistivity has little effect on it.

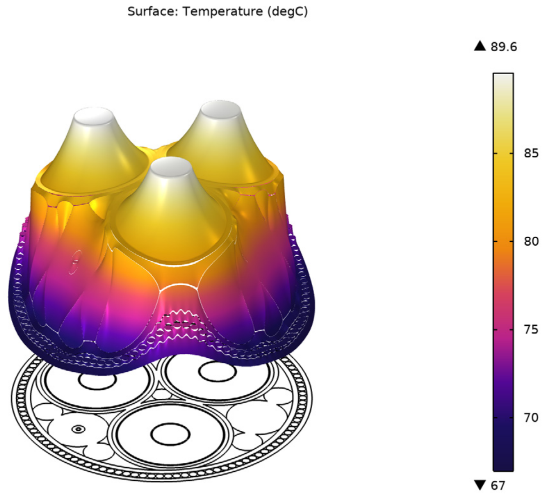
The expression for R_{dc} at 90°C has been derived directly from the expression for linearized resistivity; Equation 7. You may have wondered about the use of `Tmcon` in this expression. This is the expected average temperature for the main conductors. Continue by checking if its value is correct.

Temperature (ht)

1 In the **Model Builder** window, click **Temperature (ht)**.

2 In the **Temperature (ht)** toolbar, click **Plot**.

3 Click the **Go to Default View** button in the **Graphics** toolbar.



Already from the plot, you should be able to see the value is about 90°C. Notice this is a *decrease* with respect to last time. It is in agreement with the reduction in losses we have seen just now.

Note that the results obtained here are of *particular importance*. The plain 2D configuration at room temperature is known to agree well with the 3D twist models at room temperature (when it comes to overall losses and resistance). The plain 2D configuration with fully coupled induction heating and elevated temperatures, should therefore provide a realistic temperature profile for the 3D model.

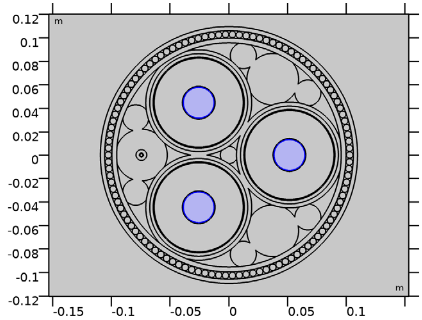
So instead of having to solve a fully coupled 3D induction heating model, you can probe the average temperatures here and use them as a "predetermined temperature distribution" in a 3D model with linearized resistivity applied. To this end, proceed by evaluating the average temperatures for the phases, screens, and armor.

Average Temperature

1 In the **Results** toolbar, click 8.85 e-12 **More Derived Values** and choose **Average>Surface Average**.

2 In the **Settings** window for **Surface Average**, type `Average Temperature` in the **Label** text field.

3 Locate the **Selection** section. From the **Selection** list, choose **Phases**.

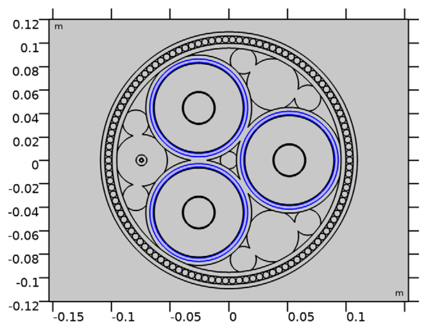


4 Locate the **Expressions** section. In the table, enter the following settings:

Expression	Unit	Description
T	degC	Temperature (phases)

5 Click **Evaluate**.

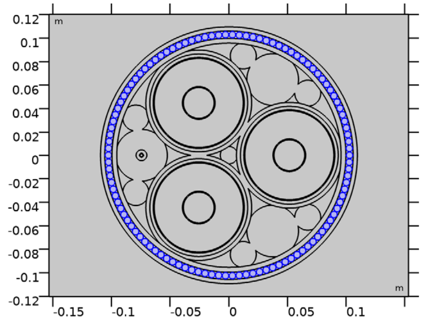
6 Locate the **Selection** section. From the **Selection** list, choose **Screens**.



7 Locate the **Expressions** section. In the table, update the description. Type `Temperature (screens)`, that is; replace “`phases`” with “`screens`”.

8 Click **Evaluate**.

9 From the **Selection** list, choose **Cable Armor**.



10 In the table, update the description. Type `Temperature (armor)`, that is; replace "`screens`" with "`armor`".

11 Click **Evaluate**.

12 Go to the **Table** window.

The average temperatures should be about 90°C, 83°C, and 70°C for the phases, screens, and armor respectively (and in agreement with `Tmcon`, `Tmpbs`, and `Tmarm`).

Finally, let us see how the conductivity can be set, such that the cable assumes a certain specified resistance.

Modeling Instructions — Preset Resistance

In the *Inductive Effects* tutorial, we have seen that the chosen conductor model (**Single conductor**, or **Homogenized multiturn**) makes a difference when it comes to the losses. In practice, it may not always be clear which one best approximates the actual conditions.

As this is a standardized cable however, you can make use of the official temperature dependent AC resistance according to the IEC 60287 series of standards [1]: We will look for the value of σ_{coil}, needed to obtain the "correct" phase resistance — and therefore, the correct total loss, see section Modeling Approach.

Start by creating a component coupling, to probe for the average phase temperature.

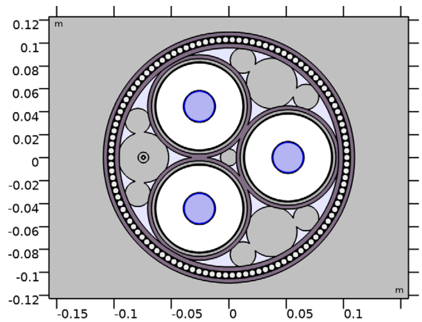
DEFINITIONS

Average 1 (aveop1)

1 In the **Definitions** toolbar, click **Nonlocal Couplings** and choose **Average**.

2 In the **Settings** window for **Average**, locate the **Source Selection** section.

3 From the **Selection** list, choose **Phases**.



When this operator is used in *global expressions* (not related to any particular domain or boundary), it will return the average value of some locally evaluated quantity in the phase domains. For example: "`aveop1(T)`" will return the average phase temperature. This temperature can then be inserted in a known or measured $R_{ac}(T)$ relation, to obtain the desired R_{ac} value.

Proceed by adding an analytic fuction for $R_{ac}(T)$.

Linearized Resistance

1 In the **Definitions** toolbar, click f(x) **Analytic**.

2 In the **Settings** window for **Analytic**, type `Linearized Resistance` in the **Label** text field.

3 In the **Function name** text field, type `Rac`.

4 Locate the **Definition** section. In the **Expression** text field, type `1.39*Rcon*(1+ALcup*(T-Tmref))`.

5 In the **Arguments** text field, type `T`.

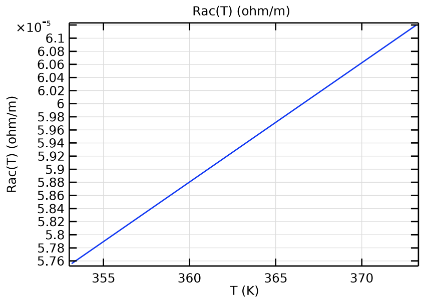
6 Locate the **Units** section. In the **Arguments** text field, type `K`.

7 In the **Function** text field, type `ohm/m`.

8 Locate the **Plot Parameters** section. In the table, enter the following settings:

Argument	Lower limit	Upper limit
T	80[degC]	100[degC]

9 Click **Plot**.



To be clear, this is *not* the temperature dependent AC resistance as derived from the coefficients provided by the IEC standard. It is based on the 90°C DC resistance, derived from the expression for linearized resistivity that we used earlier (see section Modeling Approach). The factor `1.39` is the ratio η, as discussed before. For the sake of simplicity, η is assumed to be temperature independent. You could consider this a linearisation of $R_{ac}(T)$ around 90°C.

In other words, this exercise serves as a *proof of concept*: Feel free to substitute your own temperature dependent resistance here — it does not even have to be an analytic relation, it could be an interpolating curve based on measured data, for example. The true curve will depend on the cable type and the operating conditions.

Next, introduce a **Global ODEs and DAEs** interface to determine the effective copper conductivity.

ADD PHYSICS

1 In the **Home** toolbar, click **Add Physics** to open the **Add Physics** window.

2 Go to the **Add Physics** window.

3 In the tree, select **Mathematics>ODE and DAE Interfaces>Global ODEs and DAEs (ge)**.

4 Click **Add to Component 1** in the window toolbar.

5 In the **Home** toolbar, click **Add Physics** to close the **Add Physics** window.

GLOBAL ODES AND DAES (GE)

Global Equations 1

1 In the **Model Builder** window, under **Component 1 (comp1)>Global ODEs and DAEs (ge)** click **Global Equations 1**.

2 In the **Settings** window for **Global Equations**, locate the **Global Equations** section.

3 In the table, enter the following settings:

Name	f(u,ut,utt,t) (1)	Initial value (u_0) (1)
Scup2	Rac(aveop1(T))-(mf.RCoil_1/1[m]+mf.RCoil_2/1[m]+ mf.RCoil_3/1[m])/3	Scup

4 Locate the **Units** section. Click **Define Dependent Variable Unit**.

5 In the **Dependent variable quantity** table, enter the following settings:

Dependent variable quantity	Unit
Custom unit	S/m

6 Click **Define Source Term Unit**.

7 In the **Source term quantity** table, enter the following settings:

Source term quantity	Unit
Custom unit	ohm/m

This **Global Equation**, will introduce an additional *dependent variable*: `Scup2`. Dependent variables are not predefined, nor are they derived from other variables (as is the case for *derived variables*). They are the variables to solve for, the unknowns of the system of equations.

The new dependent variable comes equipped with a *global constraint* (the expression of the form: `Rac(aveop1(T))-(mf.RCoil_1...`) and an *initial value*. Starting with that initial value, COMSOL will look for the value `Scup2` that satisfies the constraint (that sets it to zero). In other words, solving the model means looking for the value of `Scup2` that makes the average phase AC resistance equal to `Rac(aveop1(T))`.

Using equations like this requires a bit of insight. After all there may be infinitely many values of `Scup2` that are valid, or there may be none at all. In those cases the model may give an arbitrary result, or it may not converge in the first place (depending on the solver settings chosen). The next steps are needed to find a *unique solution*.

ROOT

1 Click the **Show More Options** button in the **Model Builder** toolbar.

2 In the **Show More Options** dialog box, in the tree, select the check box for the node **Physics>Advanced Physics Options**.

3 Click **OK**.

GLOBAL ODES AND DAES (GE)

Global Equations 1

1. In the **Model Builder** window, under **Component 1 (comp1)>Global ODEs and DAEs (ge)** click **Global Equations 1**.
2. In the **Settings** window for **Global Equations**, click to expand the **Discretization** section.
3. From the **Value type when using splitting of complex variables** list, choose **Real**.

   This sets `Scup2` to a strictly *real* value, as is the case for the temperature `T`, for example.

If you do not use this setting, `Scup2` will be interpreted as a *complex* value, like any other variable in the frequency domain. The problem with this, is that the constraint `Rac(T)-RCoil` is only affected by the real part of `Scup2` (as the *resistance* is the real part of the *impedance*), leaving the imaginary part completely undetermined. In addition to this, we did not intend `Scup2` to be complex in the first place.

Let us use the new `Scup2` variable in the Magnetic Fields interface and recompute. Start by removing the linearized resistivity, and switch back to the **Homogenized multiturn** conductor model.

MAGNETIC FIELDS (MF)

Phase 1

1. In the **Model Builder** window, under **Component 1 (comp1)>Magnetic Fields (mf)** click **Phase 1**.
2. In the **Settings** window for **Coil**, locate the **Constitutive Relation Jc-E** section.
3. From the **Conduction model** list, choose **Electrical conductivity**.
4. Locate the **Coil** section. From the **Conductor model** list, choose **Homogenized multiturn**.
5. Locate the **Homogenized Multiturn Conductor** section. In the σ_{coil} text field, type `Scup2`.

Phase 2, Phase 3

Repeat these steps for **Phase 2**, and **Phase 3**.

The multiturn conductor model is chosen, because it sets the current directly. This is different from the **Single conductor** conductor model, that excites the phase by means of an electric field. It will then search for the electric field magnitude required to get the desired current (this is in a way, very similar to looking for the value of `Scup2` required to get the desired resistance).

Having two things to search for at the same time (two global unknowns) will make this induction heating model rather unstable. The multiturn conductor model therfore provides a more robust alternative. For more on coil domains, see section *On Coil Domains* in the *Inductive Effects* tutorial.

In case you are wondering, the chosen conductor model affects the loss distribution by one per cent or so: The multiturn model lowers the losses in the phases a bit, and therefore — as the total loss is predetermined by $R_{ac}(T)$ — raises the screen and armor losses. The effects on the temperature distribution are small however, as you will see in a minute.

STUDY 1

In the **Home** toolbar, click **Compute**.

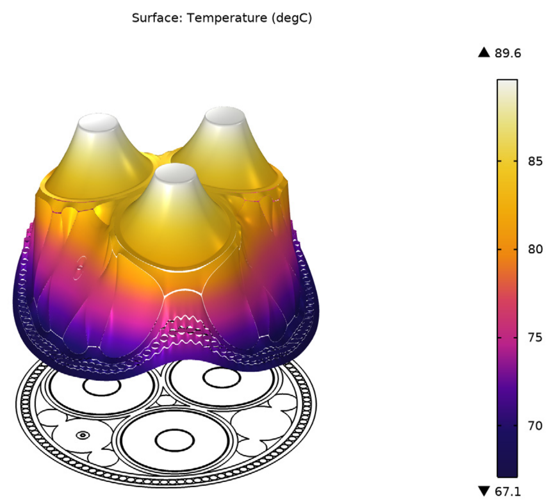
RESULTS

Temperature (ht)

1 In the **Model Builder** window, under **Results** click **Temperature (ht)**.

2 In the **Temperature (ht)** toolbar, click **Plot**.

3 Click the **Go to Default View** button in the **Graphics** toolbar.



So the average phase temperature is still about 90°C (and the screen and armor temperatures should not really be affected either). You have just used a different method to obtain the same results — *even after having switched to a different conductor model* — see section Using a Preset Resistance.

The third significant digit might be different this time. This kind of accuracy is expected, as the ratio between the AC resistance and the DC resistance is set, up to three significant digits only. Moreover, the same value is used for all three phases, while the actual figure should be slightly different for each one of them.

Now, let us do some sanity checks: For this temperature, the phase AC resistance should be 59 mΩ/km, as given by `Rac(90[degC])`. Additionally, the total thermal energy generated in the cable should be 76 kW/km, as given by `3*(I0^2/2)*Rac(90[degC])`. Here, "3" comes from the fact that we have three phases, and "1/2" comes from the peak-to-RMS conversion.

Let us see if these assumptions are correct.

Phase Losses

1 In the **Model Builder** window, click **Phase Losses**.

2 In the **Settings** window for **Surface Integration**, locate the **Expressions** section.

3 In the table, update the description. Type `Phase losses (preset Rac)`, that is; replace "`coupled ih`" with "`preset Rac`".

4 In the **Settings** window for **Surface Integration**, click **Evaluate**.

Screen Losses, Armor Losses

Repeat these steps for **Screen Losses** and **Armor Losses**.

TABLE

1 Go to the **Table** window.

If you now add the phase, screen and armor losses, the result will equate to 76 kW/km indeed. Therefore, *energy is conserved*: electric power in, equals thermal power out. Hint; you can get even better accuracy by comparing `3*(I0^2/2)*Rac(aveop1(T))` to the integral of `mf.Qh` over all electromagnetic domains, directly.

Finally, reevaluate the resistance and the inductance.

Phase AC Resistance

1 In the **Model Builder** window, click **Phase AC Resistance**.

2 In the **Settings** window for **Global Evaluation**, locate the **Expressions** section.

3 In the table, enter the following settings. That is; replace "`coupled ih`" with "`preset Rac`", and remove the second row alogether:

Expression	Unit	Description
`(mf.RCoil_1/1[m]+mf.RCoil_2/1[m]+mf.RCoil_3/1[m])/3`	`mohm/km`	`Phase AC resistance (preset Rac)`

4 Click **Evaluate**.

Phase Inductance

1 In the **Model Builder** window, click **Phase Inductance**.
2 In the **Settings** window for **Global Evaluation**, locate the **Expressions** section.
3 In the table, update the description. Type `Phase inductance (preset Rac)`, that is; replace "`coupled ih`" with "`preset Rac`".
4 In the **Settings** window for **Global Evaluation**, click **Evaluate**.
5 Go to the **Table** window.

So when compared to the plain 2D configuration at room temperature, the cable's resistance has risen with about 13%, the exact same figure with which the total losses have increased. Feel free to investigate this further by comparing the results from this model to those from the *Inductive Effects* tutorial. Lastly, observe the cable's inductive properties are barely affected by the increase in temperature (as they should).

So far, all investigations in this series have been done in 2D. This is understandable, as 2D and 2.5D models can provide pretty accurate results when it comes to the total loss, resistance, and inductance (and with very little computational effort, too). However, even though 2D models are able to provide a good insight into the general behavior of the device, capturing the precise interaction between the phases, screens, and armor will still require a full *3D twist model*.

When developing advanced numerical models in 3D with many degrees of freedom (a high *DOF count*), one of the major challenges is to prepare a good mesh — notice that for the 2D models, we have not spent any time on this at all. The next tutorial will therefore provide a detailed treatment of the 3D twist mesh (and geometry) as needed for chapter 8: The *Inductive Effects 3D* tutorial.

You have now completed this tutorial, subsequent tutorials will refer to the resulting file as `submarine_cable_06_thermal_effects.mph`.

From the **File** menu, choose **Save**.