FIELD STORAGE

Purpose

The classes of the menu *Field Storage* define the output points at which the field is calculated.

The output points may be grouped in a one-dimensional

Cut

or they may be grouped in a two-dimensional

Grid

Finally, field grids, irregular in two dimensions, may be defined in the class

Tabulated uv-Points

Command Types

The calculation of the specified field is activated by one of the commands:

Get Field

Add Field

Subtract Field

Further, the following commands may be applied to interpolate, add, subtract, replace or divide already calculated fields.

Get Pattern

Add Pattern

Subtract Pattern

Replace Pattern

Get Delta Pattern

These commands are used to calculate near fields inside regions:

Get Field Inside Device

Get Field Inside Region

Links

Classes→Electrical Objects→Field Storage

CUT

Purpose

Cut is a menu which contains classes that define regular field cuts or pattern cuts.

Field upon a surface in space:

Spherical Cut (includes far-field cuts),

Planar Cut

Cylindrical Cut

Field at the surface of a scatterer:

Surface Cut

Links

Classes→Electrical Objects→Field Storage→Cut

SPHERICAL CUT (spherical_cut)

Purpose

The class *Spherical Cut* defines points in cuts on a sphere at which the field is to be calculated. Polar as well as conical cuts can be specified, both in the near-field and in the far-field region.

Links

Classes→Electrical Objects→Field Storage→Cut→Spherical Cut

Remarks

Syntax

<object name> *spherical_cut*
(
coor_sys	: ref(<n>),
cut_type	: <si>,
theta_range	: struct(*start*:<r>, *end*:<r>, *np*:<i>),
phi_range	: struct(*start*:<r>, *end*:<r>, *np*:<i>),
e_h	: <si>,
polarisation	: <si>,
polarisation_modification	: struct(*status*:<si>, *coor_sys*:ref(<n>)),
near_far	: <si>,
near_dist	: <rl>,
file_name	: <f>,
file_format	: <si>,
comment	: <s>,
frequency	: ref(<n>)
)

where

<i> = integer

<n> = name of an object

<r> = real number

<rl> = real number with unit of length

<s> = character string

<f> = file name

<si>= item from a list of character strings

Attributes

`Coordinate System` (*coor_sys*) [name of an object], default: **blank**.

Reference to an object of one of the classes in *Coordinate System* defining the coordinate system (the output coordinate system) in which the field points will be calculated. Further, the origin serves as the phase reference for far-fields. The field polarisation components will also be expressed in this coordinate system unless otherwise specified in the attribute `Polarisation Modification`.

`Cut Type` (*cut_type*) [item from a list of character strings], default: **polar**.

Defines the direction of the cuts over the sphere. See also the remarks below.

polar

Polar cuts are cuts for which ϕ is constant and θ is varying. Polar cuts will pass through the pole of the sphere when $\theta = 0°$ is within the θ-range.

conical

Conical cuts are cuts for which θ is constant and ϕ is varying.

`theta-Range` (*theta_range*) [struct].

Defines the range of the polar angle θ (see the remarks below).

`Start` (*start*) [real number].

Start value of the polar coordinate θ, in degrees.

`End` (*end*) [real number].

End value of the polar coordinate θ, in degrees.

`Np` (*np*) [integer].

Number of θ-values. When the `Cut Type` is specified to 'polar' `Np` is the number of θ-values in each polar cut. When the `Cut Type` is specified to 'conical' `Np` is the number of conical cuts.

`phi-Range` (*phi_range*) [struct].

Defines the range of the azimuthal angle ϕ (see the remarks below).

`Start` (*start*) [real number].

Start value of the azimuthal coordinate ϕ, in degrees.

`End` (*end*) [real number].

End value of the azimuthal coordinate ϕ, in degrees.

`Np` (*np*) [integer].

Number of ϕ-values. When the `Cut Type` is specified to 'polar' `Np` is the number of polar cuts. When the `Cut Type` is specified to 'conical' `Np` is the number of ϕ-values in each conical cut.

`E/H-Field` (*e_h*) [item from a list of character strings], default: **e_field**.

Specifies whether the complex E-field or H-field shall be calculated.

e_field

The complex E-field is calculated.

h_field

The complex H-field is calculated.

`Polarisation` (*polarisation*) [item from a list of character strings], default: **linear**.

Defines how the calculated field shall be decomposed. In the near field the r-component of the field is calculated as a third component. All components refer to the polarisation coordinate system as defined under the attribute `Polarisation Modification`.

linear

>   Linear components are calculated according to Ludwig's 3rd definition, with the first component (E_{Lx}) along x and the second component (E_{Ly}) along y (at $\theta = 0°$).

circular

>   Circular components are calculated based on the linear components defined above. The first component is the right hand circular (E_{rhc}) and the second is left hand circular component (E_{lhc}).

theta_phi

>   The field is decomposed along the θ- and ϕ-unit vectors with the θ-component (E_{θ}) being the first component and the ϕ-component being the second (E_{ϕ}).

major_minor

>   The field is de-composed along the major and minor axes of the polarisation ellipse. The first field component is parallel to the major axis (E_{maj}) and the second to the minor axis (E_{min}).

linear_xpd

>   The ratios E_{Lx}/E_{Ly} and E_{Ly}/E_{Lx}, where E_{Lx} and E_{Ly} are the first and second components as defined for the '*polarisation*: linear' above. This is the linear cross-polar discrimination ratio.

circular_xpd

>   The ratios E_{rhc}/E_{lhc} and E_{lhc}/E_{rhc}, where E_{rhc} and E_{lhc} are the first and second components as defined for the '*polarisation*: circular' above. This is the circular cross-polar discrimination ratio.

theta_phi_xpd

>   The ratios E_{θ}/E_{ϕ} and E_{ϕ}/E_{θ}, where E_{θ} and E_{ϕ} are the first and second components as defined for the '*polarisation*: theta_phi' above.

major_minor_xpd

>   The axial ratios E_{maj}/E_{min} and E_{min}/E_{maj}, where E_{maj} and E_{min} are the major and minor axes as defined for the '*polarisation*: major_minor' above.

power

> The first component is the amplitude of the field, $|\mathbf{E}|$, (i.e. the square root of the power) and the second component is the complex square root $\sqrt{E_{rhc}/E_{lhc}}$. The phase of the latter component is the rotation angle of the polarisation ellipse.
>
> In the far field the square root of the power is determined from
>
> $$|\mathbf{E}| = \sqrt{|E_{Lx}|^2 + |E_{Ly}|^2}$$
>
> and in the near field it is determined from all three field components: $|\mathbf{E}| = \sqrt{|E_{Lx}|^2 + |E_{Ly}|^2 + |E_r|^2}$, E_r being the r-component of the field.

Polarisation Modification (*polarisation_modification*) [struct].

> Defines a coordinate system in which the field polarisation components are determined (if different from the output coordinate system defined). See also the remarks section below.

> Status (*status*) [item from a list of character strings], default: **off**.
>
> > Determines if the polarisation modification shall be performed:
> >
> > off
> >
> > > No polarisation modification, the polarisation is defined in the above defined output coordinate system and the polarisation coordinate system is identical to the output coordinate system.
> >
> > on
> >
> > > The polarisation is defined in the coordinate system defined next.
>
> Coordinate System (*coor_sys*) [name of an object], default: **blank**.
>
> > Reference to an object of one of the classes in *Coordinate System* defining the coordinate system (the polarisation coordinate system) in which the polarisation components of the calculated field vectors will be expressed. Shall only be specified for Status: on.

Near Far (*near_far*) [item from a list of character strings], default: **far**.

> The value specifies if a near or a far field is to be calculated.

> far
>
> > A far field is calculated. In this case only two field components are calculated according to the specified Polarisation as described in the remarks below.

> near
>
> > The three field components of a near field is calculated. These are the two components defined under the attribute Polarisation below and the third component is the radial component.

Near Dist (*near_dist*) [real number with unit of length], default: **0**.

> Defines the radius of a near-field sphere. For a far field this attribute has no effect and needs not to be specified.

`File Name` (*file_name*) [file name].

> Name of a file, to which the calculated field values shall be written.

`File Format` (*file_format*) [item from a list of character strings], default: **TICRA**.

> The file format used to store the field data:
>
> TICRA
>
> > The field is written in TICRA Tools units and stored as an ASCII-file according to the TICRA-format described in *Field Data in Cuts*. The recommended file extension is *.cut*.
>
> `EDX (Obsolete)` (*EDX*)
>
> > The field is written in SI units in a format according to the Electromagnetic Data Exchange (EDX) standard. The recommended file extension is *.cut*.
>
> `EDI (Obsolete)` (*EDI*)
>
> > Obsolete file format name. The same as EDX.

`Comment` (*comment*) [character string], default: **Field data in cuts**.

> A line of text which will be written as a header in the file specified by `File Name` above.

`Frequency` (*frequency*) [name of an object], default: **blank**.

> Reference to a *Frequency* object, defining the frequencies for which the field is calculated. The reference must be to the same object as specified in the `Frequency` attribute of the source generating the field. A `Frequency` needs not to be specified. In that case, the frequencies in the *Frequency* object of the source generating the field will be applied. More information can be found in the "Remarks" section.

Command Types

The *Spherical Cut* class is derived from the class *Field Storage*. See this class for available commands.

The calculation of the specified field is activated by one of the commands:

> *Get Field*
>
> *Add Field*
>
> *Subtract Field*

Remarks

The section introduces the position of the field points and the definition of a polarisation coordinate system.

Field Points

Field points in a *Spherical Cut* are defined in usual spherical (θ, ϕ)-coordinates given in the output coordinate system. For far fields, (θ, ϕ) defines a direction

$$\hat{r} = \hat{x} \sin\theta \cos\phi + \hat{y} \sin\theta \sin\phi + \hat{z} \cos\theta$$

and for near fields, (θ, ϕ) defines a point

$$\bar{R} = R(\hat{x} \sin\theta \cos\phi + \hat{y} \sin\theta \sin\phi + \hat{z} \cos\theta)$$

where $R = |\bar{R}|$ is the radius (given by `Near Dist`) of the near-field sphere. In a polar cut (see Figure 1), ϕ is fixed and θ takes on the values

$$\theta_i = \theta_{start} + \Delta\theta \cdot (i-1), \qquad i = 1, 2, ..., n_\theta \tag{1}$$

with

$$\Delta\theta = (\theta_{end} - \theta_{start})/(n_\theta - 1)$$

where θ_{start} and θ_{end} are the members `Start` and `End`, respectively, of the attribute `theta-Range`, and n_θ is the member `Np` of the same attribute.

The ϕ-angle is increased equidistantly from one cut to the next by the values

$$\phi_j = \phi_{start} + \Delta\phi \cdot (j-1), \qquad j = 1, 2, ..., n_\phi \tag{2}$$

with

$$\Delta\phi = (\phi_{end} - \phi_{start})/(n_\phi - 1)$$

where ϕ_{start} and ϕ_{end} are the members `Start` and `End`, respectively, of the attribute `phi-Range`, and n_ϕ is the member `Np` of the same attribute.

In a conical cut (see Figure 2) θ is fixed and ϕ runs through the values given by Eq. (2). From one conical cut to the next, the θ-angle is increased according to Eq. (1).

The Polarisation Coordinate System

When the field polarisation is requested in another coordinate system than the output coordinate system then the attribute `Polarisation Modifica-tion` shall be applied with `Status` set to 'on' followed by a reference to the polarisation coordinate system.

The attribute `Polarisation` may be specified as 'linear' (E_{Lx}- and E_{Ly}-components), 'circular' (E_{rhc}- and E_{lhc}-components) or 'theta_phi' (E_θ- and E_ϕ-components). In the near field also an E_r-component will be present. In case of 'theta_phi' polarisation the electric field is given by

$$\bar{E} = E_r \hat{r} + E_\theta \widehat{\theta} + E_\phi \widehat{\phi}$$

where $\hat{r}$, $\widehat{\theta}$ and $\widehat{\phi}$ are the polarisation vectors in the polarisation coordinate system. The spherical cut in which the field is determined is always given in the output coordinate system.

An example is shown in Figure 3, the cut is the polar cut shown in red and the field shall be determined at one of the output points, P. The output coordinate system is denoted $x_0 y_0 z_0$ and the polarisation coordinate system is $x_p y_p z_p$. In
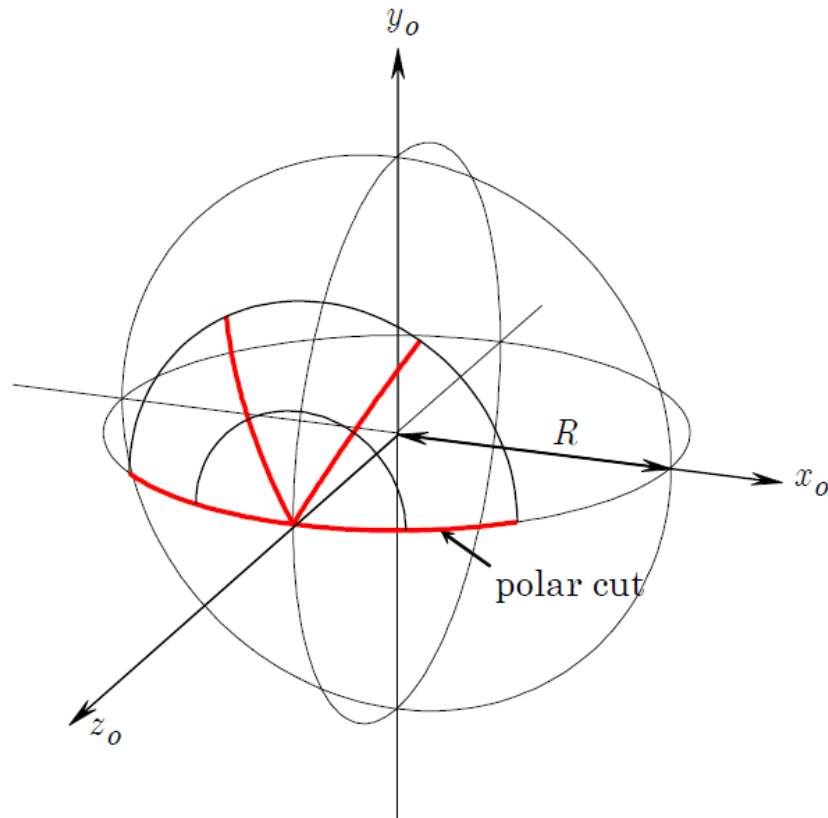
Figure 1   4 polar cuts for $0° \leq \theta \leq 45°$, and $\phi = 0°$, $60°$, $120°$ and $180°$.

the figure the latter is a simple rotation of the former around the y-axis but any coordinate system may be chosen as polarisation coordinate system.

Internally in TICRA Tools the field is calculated in Cartesian components in the output coordinate system

$$\bar{E} = E_{ox}\widehat{x}_0 + E_{oy}\widehat{y}_0 + E_{oz}\widehat{z}_0$$

The field is then converted to the new components

$$\bar{E} = E_{px}\widehat{x}_p + E_{py}\widehat{y}_p + E_{pz}\widehat{z}_p$$

before the Cartesian components are converted to (in this case) the polar components in the usual way.

The resulting θ_p-component (shown in blue in Figure 3) is pointing away from the z_p-axis in the same way as the standard θ_0-component will point away from the z_0-axis (along θ_0).

Using `Polarisation Modification` can give problems when converting fields in $\theta\phi$-components to a new coordinate system. At the poles $\theta = 0°$ and $\theta = 180°$ the value of ϕ becomes undefined and thus $\phi = 0°$ is assumed as default. This may lead to discontinuities in the patterns at $\theta = 0°$ and $\theta = 180°$.

Frequency

Specifying `Frequency` in *Spherical Cut* is useful when working with the *Get Pattern*, *Add Pattern*, *Subtract Pattern* and *Replace Pattern* commands that
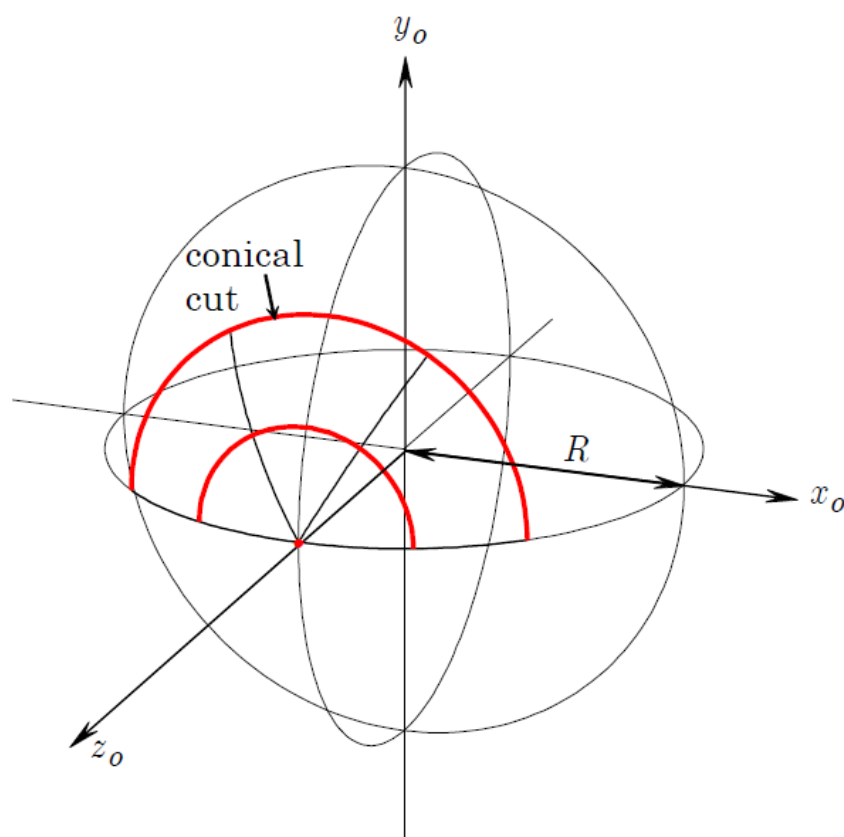
Figure 2   3 conical cuts for $\theta = 0°$, $22.5°$ and $45°$, and $0° \leq \phi \leq 180°$.

are used to interpolate, add, subtract and replace, respectively, *already-calculated fields*. There is no frequency information stored in the output files, so specifying `Frequency` in these cases can help to make sure that, for example, only fields originally computed at the same frequency are added or subtracted. On the other hand, the *Get Field*, *Add Field* and *Subtract Field* commands are used to calculate, add and subtract, respectively, *fields from sources* that always have their own well-defined frequency. Thus, in these cases specifying `Frequency` in *Spherical Cut* is not necessary.
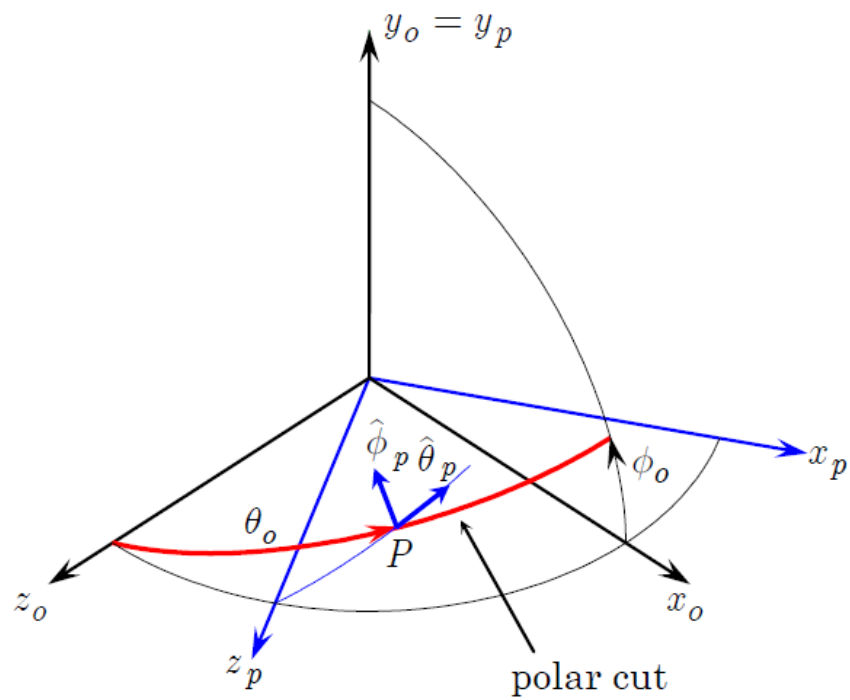
Figure 3   In red is shown a single polar cut in the output coordinate
system given by $x_0y_0z_0$. A field point P at the direction
(θ_0, ϕ_0) is illustrated. The polarisation is expressed in
$\theta\phi$-components (`Polarisation` set to 'theta_phi') along $\widehat{\theta}_p$
and $\widehat{\phi}_p$ in the polarisation coordinate system $x_py_pz_p$.