Kiva Documentation

Release 0.3.1

Neal Kruis

October 17, 2015

Contents

1 Getting Started **3**

2 Input Definition **5**

2.1 Simulation Control 5

2.2 Materials 6

2.3 Foundation 6

2.4 Foundation (Advanced) 27

3 Methodology **39**

4 Testing **41**

5 Bibliography **43**

Kiva is a free and open source ground heat transfer calculation tool written in C++. Specifically, Kiva is used to calculate heat loss and gain on a timestep basis from building foundations. The goal is to create a tool that can integrate the multi-dimensional heat transfer into standard building energy simulation engines.

Contents:

CHAPTER 1

Getting Started

Kiva is a command line tool. You can execute Kiva from the command line as follows:

```
kiva input.yaml weather.epw output.csv
```

Kiva will simulate the foundation defined in `input.yaml` using the weather conditions specfified in `weather.epw` and generate time series results in `output.csv`.

Input files are YAML formatted descriptions of building foundations and simulation control settings. This is described in more detail in the following sections.

Weather files are read in the EnergyPlus weather (EPW) file format.

Output files are a simple comma separated variable (CSV) file format, with results corresponding to output requested in the `input.yaml` file.

CHAPTER 2

Input Definition

Kiva input is specified in a hierarchical YAML file. This file has three objects at the highest level, `Simulation Control`, `Materials`, and the `Foundation` itself. All inputs are defined in standard International System (SI) units.

2.1 Simulation Control

This defines the settings of the simulation period and timestep.

Example:

```
Simulation Control:
  Start Date: 2015-Jan-1
                    End Date: 2015-Dec-31
                    Timestep: 60 # [min]
```

2.1.1 Start Date

Specifies the start date of the simulation. Simulation begins at 12:00am of this day. This is specified as a date string (e.g., YYYY-Mon-DD, YYYY/MM/DD).

Required:	Yes
Type:	Date string

2.1.2 End Date

Specifies the end date of the simulation. Simulation ends one timestep previous to 12:00am of the following day. This is specified as a date string (e.g., YYYY-Mon-DD, YYYY/MM/DD).

Required:	Yes
Type:	Date string

2.1.3 Timestep

Timestep duration in minutes used in calculations.

Required:	Yes
Type:	Integer
Units:	Minutes

2.2 Materials

Materials define the intrensic thermophysical properties of the solid components of the computational domain (eg: surrounding soil, structural concrete, insulation). All materials must fall within a single *Materials:* group.

Example:

```
Materials:
  Soil:
    Conductivity: 0.864 # [W/m-K]
    Density: 1510.0  # [kg/m3]
    Specific Heat: 1260.0  # [J/kg-K]
  Concrete:
    Conductivity: 1.98  # [W/m-K]
    Density: 1900.0  # [kg/m3]
    Specific Heat: 665.0  # [J/kg-K]
  XPS:
    Conductivity: 0.029  # [W/m-K]
    Density: 28.0  # [kg/m3]
    Specific Heat: 1450.0  # [J/kg-K]
```

Each instance of a material begins with a descriptive name (eg: Soil, Concrete or XPS) and contains the following attributes:

2.2.1 Conductivity

Thermal conductivity of the material.

Required:	Yes
Type:	Numeric
Units:	W/m-K

2.2.2 Density

Density of the material.

Required:	Yes
Type:	Numeric
Units:	kg/m^3

2.2.3 Specific Heat

Specific heat of the material.

Required:	Yes
Type:	Numeric
Units:	J/kg-K

2.3 Foundation

The description of the foundation design is provided within the two-dimensional context. This profile is applied along the entire perimeter of the foundation.

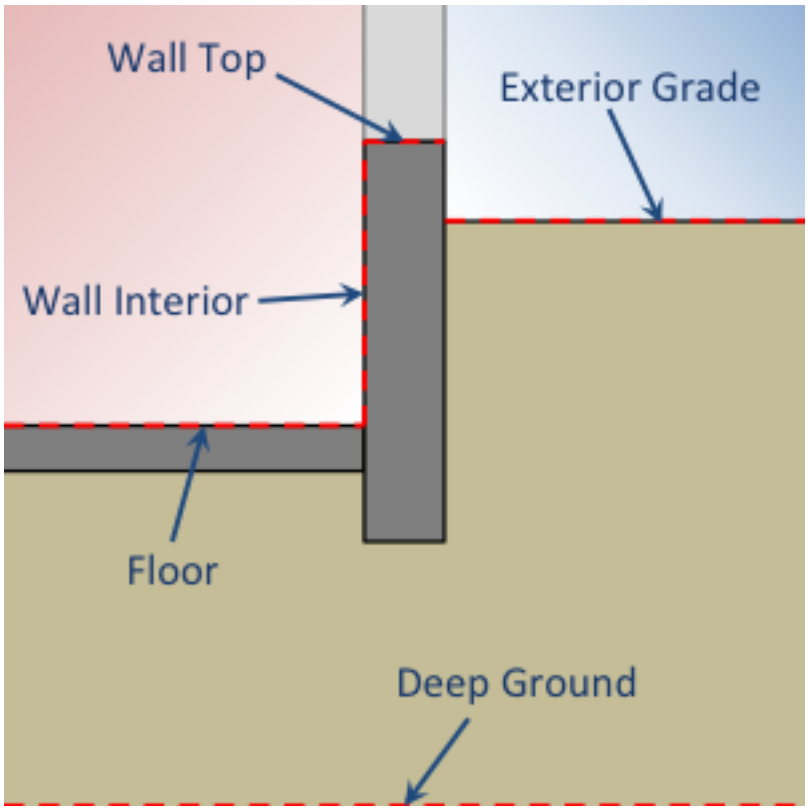

Fig. 2.1: Two-dimensional context for `Foundation` object definition

The foundation insulation and structural components are defined by (up to) six sub-objects. These objects allow the user to flexibly describe any foundation construction.

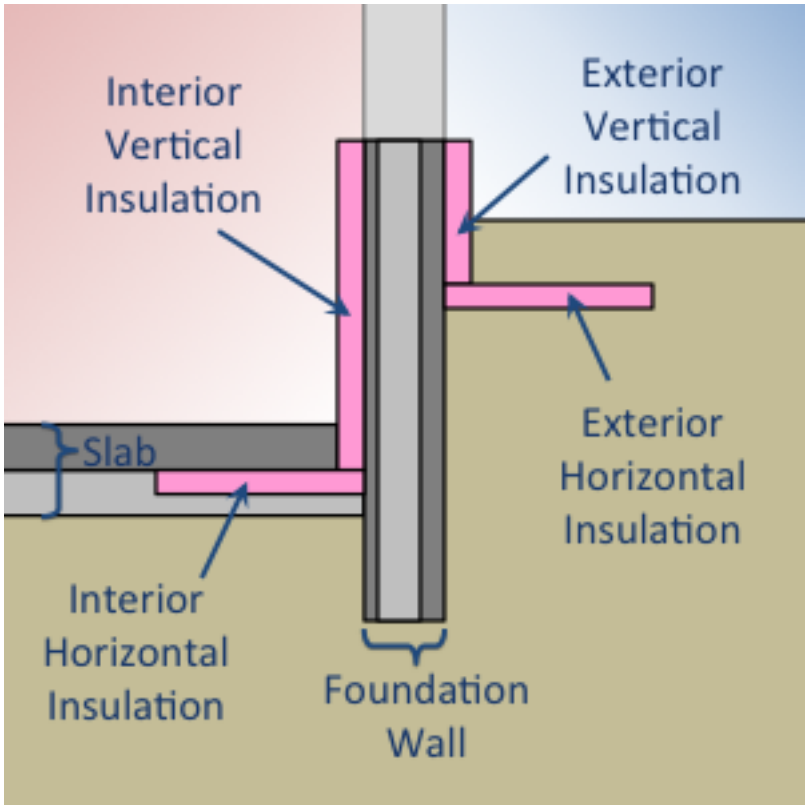

Fig. 2.2: Insulation and structural design components

Example:

```
Foundation:
  Foundation Depth: 0.0 # [m]
  Polygon:
    - [0, 0] # [m, m]
    - [0, 20] # [m, m]
    - [20, 20] # [m, m]
    - [20, 0] # [m, m]
  Soil: Typical Soil # Material reference
  Slab:
    Layers:
      -
        Material: Concrete # Material reference
        Thickness: 0.2032 # [m]
  Wall:
    Layers:
      -
        Material: Concrete # Material reference
        Thickness: 0.3048 # [m]
    Height Above Grade: 0.3048  # [m]
    Height: 0.508 # [m]
  Interior Horizontal Insulation:
    Depth: 0.2032 # [m]
    Width: 0.4064 # [m]
```

```
    Material: XPS # Material reference
    Thickness: 0.0508
  Interior Vertical Insulation:
    Depth: 0.2032 # [m]
    Material: XPS # Material reference
    Thickness: 0.0508 # [m]
  Indoor Air Temperature: 295.372 # [K]
```

2.3.1 Foundation Depth

`Foundation Depth` defines the distance from the wall top to the floor. This value is used to characterize the type of foundation (slab, crawlspace, or basement). For example, a value of zero would represent a sla and a value near 2 meters would represent a basement.

Required:	Yes
Type:	Numeric
Units:	m

2.3.2 Polygon

The foundation shape is defined by the description of a single polygon. The perimeter of this polygon defines the location of the interior surface of the foundation wall. The positioning of the foundation insulation and structural components are translated into three-dimensional space internally.

The polygon is specified by a list of x-y Cartesian vertices tracing the foundation perimeter in a clockwise fashion. When simulating in three-dimensions, the polygon must be rectilinear (comprised only of right angles).

Example:

```
Polygon:
  - [0, 20]
  - [15, 20]
  - [15, 30]
  - [30, 30]
  - [30, 17]
  - [22, 17]
  - [22, 0]
  - [12, 0]
  - [12, 10]
  - [0, 10]
```

Required:	Yes
Type:	List [N] of lists [2] of numerics
Units:	m

2.3.3 Soil

Represents the soil surrounding the building foundation.

Required:	Yes
Type:	Material reference

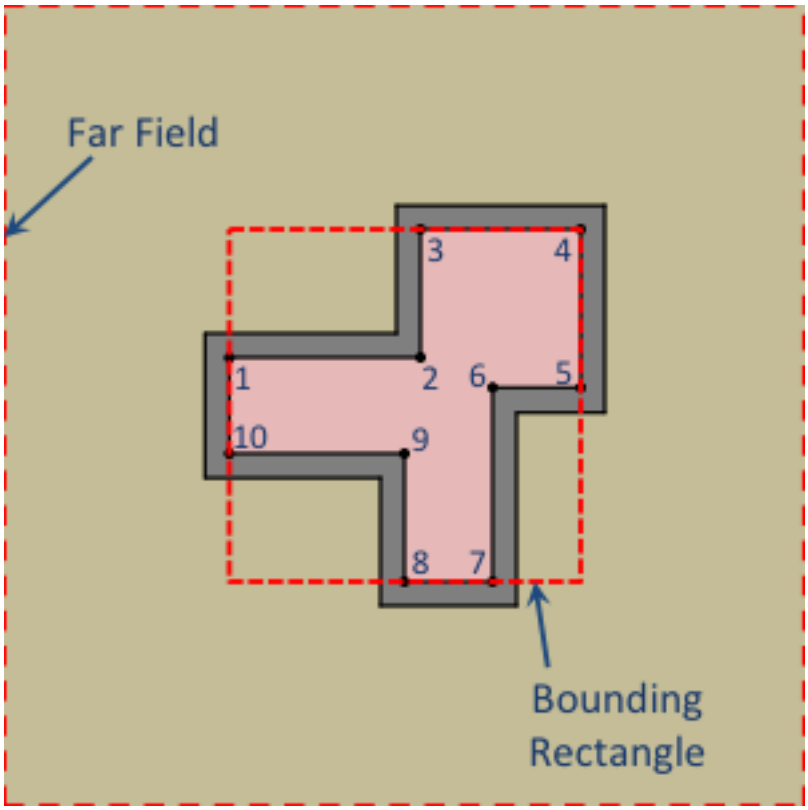

Fig. 2.3: Plan view illustrating foundation shape vertex definition and far-field boundary.

2.3.4 Slab

This defines the costruction of the floor slab in the foundation. This is not required. If there is no slab defined for a given foundation, then the floor is exposed soil.

Example:

```
Slab:
  Layers:
    -
      Material: XPS # Material reference
      Thickness: 0.0508 # [m]
    -
      Material: Concrete # Material reference
      Thickness: 0.2032 # [m]
```

Required:	No
Type:	Compound object

Layers

Layers are specified as a list of material references, and thickness values from the outtermost layer to the innermost layer (at the floor surface). A layer of insulation can be added to model whole-slab insulation.

Required:	Yes
Type:	List of layers (a material and a thickness)

Material

Material composing the layer.

Required:	Yes
Type:	Material reference

Thickness

Thickness of the layer.

Required:	Yes
Type:	Numeric
Units:	m

Emissivity

Interior emissivity of the slab used for interior long-wave radiation calculations.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	0.8

2.3.5 Wall

This defines the costruction of the foundation wall. This is not required. If there is no wall defined for a given foundation, then the wall is exposed soil.

Example:

```
Wall:
  Height: 2.95 # [m]
  Height Above Grade: 0.3048  # [m]
  Layers:
    -
      Material: XPS # Material reference
      Thickness: 0.0508 # [m]
    -
      Material: Concrete # Material reference
      Thickness: 0.2032 # [m]
    -
      Material: XPS # Material reference
      Thickness: 0.0508 # [m]
```

Required:	No
Type:	Compound object

Height

The height of the wall describes the distance from the wall top to the bottom of the foundation footer (the footer is not modeled separately). This value should generally be greater than that of the *Foundation Depth* combined with the total thickness of the slab.

Required:	Yes
Type:	Numeric
Units:	m

Height Above Grade

The height of the wall top relative to the grade (z = 0).

Required:	Yes
Type:	Numeric
Units:	m

Layers

Layers are specified as a list of material references, and thickness values from the outtermost layer to the innermost layer (at the interior wall surface).

Material

Material composing the layer.

Required:	Yes
Type:	Material reference

Thickness

Thickness of the layer.

Required:	Yes
Type:	Numeric
Units:	m

Interior Emissivity

Interior emissivity of the wall used for interior long-wave radiation calculations.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	0.8

Exterior Emissivity

Exterior emissivity of the wall used for exterior long-wave radiation calculations.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	0.8

Exterior Absorptivity

Exterior absorptivity of the wall used for calculating absorbed solar radiation.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	0.8

2.3.6 Interior Horizontal Insulation

This defines the position, dimensions, and material of interior horizontal insulation. Interior horizontal insulation begins at the wall's interior surface and extends inward and downward to a user-specified width and thickness at a user-specified depth at or below the *Foundation Depth*.

Example:

```
Interior Horizontal Insulation:
  Material: XPS # Material reference
  Thickness: 0.0508 # [m]
  Depth: 0.2032  # [m]
  Width: 0.4064 # [m]
```

Required:	No
Type:	Compound object

Material

Insulation material reference.

Required:	Yes
Type:	Material reference

Thickness

Thickness of the insulation.

Required:	Yes
Type:	Numeric
Units:	m

Depth

Depth of the insulation measured from the wall top to the top of the insulation.

Required:	Yes
Type:	Numeric
Units:	m

Width

Width of the insulation extending from the interior wall surface.

Required:	Yes
Type:	Numeric
Units:	m

2.3.7 Interior Vertical Insulation

This defines the position, dimensions, and material of interior vertical insulation. Interior vertical insulation begins at the wall top and extends downward and inward to a user-specified depth and thickness. The depth can be specified to model partial interior wall insulation.

Example:

```
Interior Vertical Insulation:
  Material: XPS # Material reference
  Thickness: 0.0508 # [m]
  Depth: 0.6096  # [m]
```

Required:	No
Type:	Compound object

Material

Insulation material reference.

Required:	Yes
Type:	Material reference

Thickness

Thickness of the insulation.

Required:	Yes
Type:	Numeric
Units:	m

Depth

Depth of the insulation measured from the wall top to the bottom of the insulation.

Required:	Yes
Type:	Numeric
Units:	m

2.3.8 Exterior Horizontal Insulation

This defines the position, dimensions, and material of exterior horizontal insulation. Exterior horizontal insulation begins at the wall's exterior surface and extends outward and downward to a user-specified width and thickness at a user-specified depth at or below the grade level.

Example:

```
Exterior Horizontal Insulation:
  Material: XPS # Material reference
  Thickness: 0.0508 # [m]
  Depth: 0.3048  # [m]
  Width: 0.6096 # [m]
```

Required:	No
Type:	Compound object

Material

Insulation material reference.

Required:	Yes
Type:	Material reference

Thickness

Thickness of the insulation.

Required:	Yes
Type:	Numeric
Units:	m

Depth

Depth of the insulation measured from the wall top to the top of the insulation.

Required:	Yes
Type:	Numeric
Units:	m

Width

Width of the insulation extending from the interior wall surface.

Required:	Yes
Type:	Numeric
Units:	m

2.3.9 Exterior Vertical Insulation

This defines the position, dimensions, and material of exterior vertical insulation. Exterior vertical insulation begins at the wall top and extends downward and outward to a user-specified depth and thickness.

Example:

```
Exterior Vertical Insulation:
  Material: XPS # Material reference
  Thickness: 0.0508 # [m]
  Depth: 2.0  # [m]
```

Required:	No
Type:	Compound object

Material

Insulation material reference.

Required:	Yes
Type:	Material reference

Thickness

Thickness of the insulation.

Required:	Yes
Type:	Numeric
Units:	m

Depth

Depth of the insulation measured from the wall top to the bottom of the insulation.

Required:	Yes
Type:	Numeric
Units:	m

2.3.10 Indoor Air Temperature Method

Allows the user to choose between having a constant indoor temperature for the duration of the simulaiton or to reference temperatures from a file.

Required:	No
Type:	Enumeration
Values:	`FILE` or `CONSTANT`
Default:	`CONSTANT`

2.3.11 Indoor Air Temperature File

If *Indoor Air Temperature Method* is `FILE` the indoor dry-bulb temperature (in Kelvin) will be set using hourly values defined in a comma separted value (CSV) file.

Example:

```
Indoor Air Temperature File:
  Name: ../path/to/file.csv
  Index: [1,1]
```

Required:	No
Type:	Compound object

Name

Path (relative or absolute) file.

Required:	Yes
Type:	File Path

Index

A list of two values corresponding to the row and column where the hourly data begins in the file. A value of `[0, 0]` starts at the first row and first column. A value of `[0,1]` starts at the first row and second column.

Required:	Yes
Type:	List [2] of integers

2.3.12 Indoor Air Temperature

If *Indoor Air Temperature Method* is `CONSTANT` the indoor dry-bulb temperature will be set using this value. If *Indoor Air Temperature Method* is `FILE`, then this is not required.

Required:	Depends
Type:	Numeric
Units:	K

2.3.13 Outdoor Air Temperature Method

Allows the user to choose between having a constant outdoor temperature for the duration of the simulaiton or to reference temperatures from the weather file.

Required:	No
Type:	Enumeration
Values:	`WEATHER-FILE` or `CONSTANT`
Default:	`WEATHER-FILE`

2.3.14 Outdoor Air Temperature

If *Outdoor Air Temperature Method* is `CONSTANT` the outdoor dry-bulb temperature will be set using this value. If *Outdoor Air Temperature Method* is `WEATHER-FILE`, then this is not required.

Required:	Depends
Type:	Numeric
Units:	K

2.3.15 Soil Absorptivity

Solar absorptivity of the soil or grade surface.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	0.8

2.3.16 Soil Emissivity

Long-wave emissivity of the soil or grade surface.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	0.8

2.3.17 Surface Roughness

Represents the relief of the surface. This value is used to calculate forced convection and the wind speed near the grade surface. Roughness values in the table below are converted from the more qualitative rougness values used in DOE-2 and EnergyPlus. Estimates for soil, gravel, and grass are also shown.

Example Surface	Roughness [m]
Glass	0.0000
Smooth Plaster	0.0044
Clear Pine	0.0052
Concrete	0.0208
Brick	0.0268
Stucco	0.0468
Soil	0.0500
Gravel	0.1200
Grass	0.3000

Required:	No
Type:	Numeric
Units:	m
Default:	0.3

2.3.18 Local Boundary Layer Thickness

Local boundary layer thickness used for calculating local wind speeds from weather file wind speeds.

Required:	No
Type:	Numeric
Units:	m
Default:	370

2.3.19 Local Terrain Exponent

Local terrain exponent used for calculating local wind speeds from weather file wind speeds.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	0.22

2.3.20 Orientation

Defines the orientation of the building clockwise relative to North (East = $\pi/2$, South = π, West = $3\pi/2$). This is used to calculate the solar incidence and wind direction relative to exterior vertical foundation surfaces.

Required:	No
Type:	Numeric
Units:	radians
Default:	0.0

2.3.21 Perimeter Surface Width

This value is used to define a portion of the slab's perimeter separately from the slab core. This will affect the meshing of the slab, but is intended primarily for separate output reporting for each region.

Required:	No
Type:	Numeric
Units:	m
Default:	0.0

2.3.22 Output Report

The output report defines what variables are written to the CSV output file and how often they are written.

Example:

```
Output Report:
  Minimum Reporting Frequency: 60 # [min]
  Reports:
    - 0 # Slab Core Average Heat Flux [W/m2]
    - 1 # Slab Core Average Temperature [K]
    - 2 # Slab Core Average Effective Temperature [C]
    - 3 # Slab Core Total Heat Transfer Rate [W]
    - 4 # Slab Perimeter Average Heat Flux [W/m2]
    - 5 # Slab Perimeter Average Temperature [K]
    - 6 # Slab Perimeter Average Effective Temperature [C]
    - 7 # Slab Perimeter Total Heat Transfer Rate [W]
    - 8 # Slab Average Heat Flux [W/m2]
    - 9 # Slab Average Temperature [K]
    - 10 # Slab Total Heat Transfer Rate [W]
    - 11 # Wall Average Heat Flux [W/m2]
    - 12 # Wall Average Temperature [K]
    - 13 # Wall Average Effective Temperature [C]
    - 14 # Wall Total Heat Transfer Rate [W]
    - 15 # Foundation Average Heat Flux [W/m2]
    - 16 # Foundation Average Temperature [K]
    - 17 # Foundation Total Heat Transfer Rate [W]
```

Required:	No
Type:	Compound object

Minimum Reporting Frequency

Kiva can provide timeseries output at the same interval as the timestep. This input will override to delay output and write it a a lower frequency. This helps to reduce the output size when running at very small timesteps.

Required:	No
Type:	Integer
Units:	min
Default:	60

Reports

This is a list of report ID numbers that Kiva will write to the CSV output file. The IDs and there corresponding output are listed in the table below:

When *Perimeter Surface Width* is not specified, the entire slab is considered to be "Core".

"Effective Temperature" is used for preprocessed ground temperatures in whole-building simulation engines. These values represent the effective temperature on the ground's side of the slab core, slab perimeter, or wall layers. When used in a whole-building simulation, the construction in the whole-building model should be the same as the layers defined for the respective surface in Kiva (ignoring any insulation objects).

Required:	No
Type:	List [N] of integers
Default:	No reports

2.3.23 Output Snapshots

Output snapshots are used to graphically visualize domain temperatures and/or heat fluxes. Each series of snapshots is part of a list within the *Output Snapshots* object. A series consists of potentially many snapshots taken of a slice of

the domain at a user-specified frequency between a start and end date.

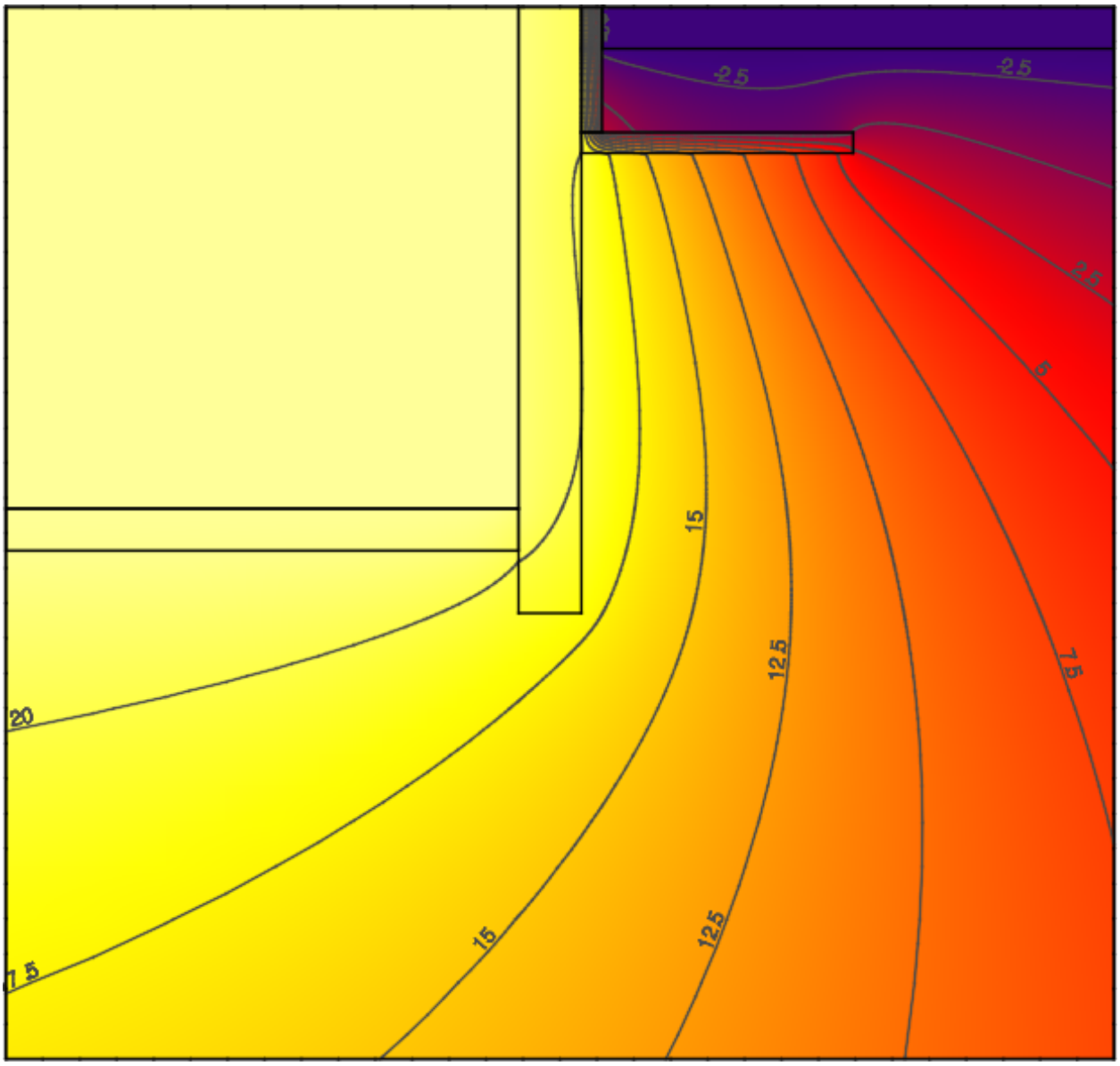

Fig. 2.4: Example profile snapshot

Example:

```
Output Snapshots:
  -
   Directory: Output/Profile
   Size: 800
   Frequency: 1
   Start Date: 2015-Dec-21
   End Date: 2015-Dec-21
   X Range: [0, 30]
   Z Range: [-30, 0.3048]
```

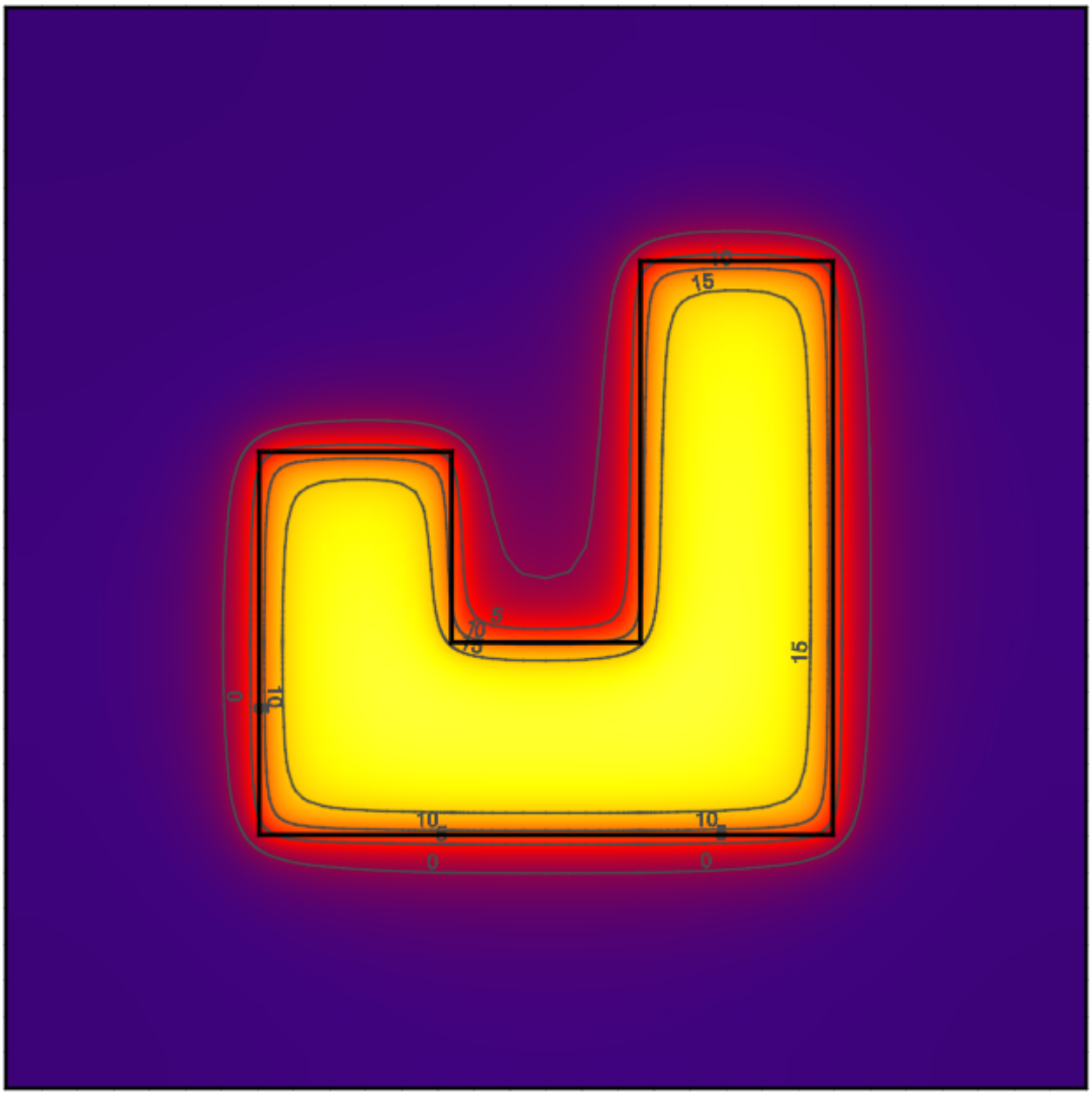

Fig. 2.5: Example plan snapshot

Required:	No
Type:	List [N] of compound objects

Directory

Directory where snapshots are created. An ordered file name, `XXXX.png`, identifies each snapshot within a series. For example, the 134th snapshot in a series with a directory name of `Profile` will be created as `Profile/0134.png`.

Required:	Yes
Type:	Directory Path

Size

The size in pixels of each snapshot file. Outputs are all generated as square images.

Required:	No
Type:	Integer
Units:	pixels
Default:	800

Frequency

The frequency, in hours, at which new snapshots are taken. The default is 36 hours so that the snapshots capture both nighttime and daytime output.

Required:	No
Type:	Integer
Units:	hours
Default:	36

Start Date

Specifies the start date of the snapshots. Snapshots begin at 12:00am of this day. This is specified as a date string (e.g., YYYY-Mon-DD, YYYY/MM/DD).

Required:	No
Type:	Date string
Default:	Simulation start date

End Date

Specifies the end date of the snapshots. Snapshots end before 12:00am of the following day. This is specified as a date string (e.g., YYYY-Mon-DD, YYYY/MM/DD).

Required:	No
Type:	Date string
Default:	Simulation end date

X Range

Defines the range the domain captured in the snapshot in the "X"-direction (`[Xmin, Xmax]`). By default the *X Range* will show the entire extents of the "X" direction, and may not show the detail where heat is flowing near the foundaiton. For three-dimensional solutions, a slice along a plane in the "X"-direction can be specified by giving both `Xmin` and `Xmax` the same value.

The snapshot will round the range to the next cell division.

Required:	No
Type:	List [2] of numerics
Units:	m
Default:	"X" extents of the domain

Y Range

Defines the range the domain captured in the snapshot in the "Y"-direction (`[Ymin, Ymax]`). By default the *Y Range* will show the entire extents of the "Y" direction, and may not show the detail where heat is flowing near the foundaiton. For three-dimensional solutions, a slice along a plane in the "Y"-direction can be specified by giving both `Ymin` and `Ymax` the same value. For two-dimensional simulations this should not be included.

The snapshot will round the range to the next cell division.

Required:	No
Type:	List [2] of numerics
Units:	m
Default:	"Y" extents of the domain

Z Range

Defines the range the domain captured in the snapshot in the "Z"-direction (`[Zmin, Zmax]`). By default the *Z Range* will show the entire extents of the "Z" direction, and may not show the detail where heat is flowing near the foundaiton. For three-dimensional solutions, a slice along a plane in the "Z"-direction can be specified by giving both `Zmin` and `Zmax` the same value.

The snapshot will round the range to the next cell division.

Required:	No
Type:	List [2] of numerics
Units:	m
Default:	"Z" extents of the domain

Plot Type

Defines the type of output plotted. Options are `TEMPERATURE` and `HEAT-FLUX`. For `HEAT-FLUX`, the user may also specify a *Flux Direction* for output.

Required:	No
Type:	Enumeration
Values:	`TEMPERATURE` or `HEAT-FLUX`
Default:	`TEMPERATURE`

Flux Direction

When *Plot Type* is `HEAT-FLUX`, the snapshots show the magnitude of heat flux throughout the domain. This input allows the user to specify whether they want to display the overall magnitude, `MAG`, or the magnitude in a given direciton, `X`, `Y`, or `Z`.

Required:	No
Type:	Enumeration
Values:	`MAG`, `X`, `Y`, or `Z`
Default:	`MAG`

Unit System

Defines the units used in the output snapshots. Options are `IP` (Inch-Pound), and `SI` (International System). Keep in mind that regardless of this value, all other inputs are still defined in the SI unit system.

Required:	No
Type:	Enumeration
Values:	`IP` or `SI`
Default:	`SI`

Output Range

Specifies the range of output shown in the snapshots. The units of the range depend on the value of *Plot Type* and *Unit System*.

Required:	No
Type:	List [2] of numerics
Units:	Depends
Default:	"Z" extents of the domain

Color Scheme

Specifies the color scheme used within the *Output Range*. Options are:

- `CMR`, best color scheme where colors progress in brightness with magnitude (prints in black-and-white),
- `JET`, like a rainbow(!), but doesn't print well,
- `NONE`, do not show any output. This can be used to illustrate meshing independent of results.

Required:	No
Type:	Enumeration
Values:	`CMR`, `JET`, or `NONE`
Default:	`CMR`

Mesh

Enables the display of the mesh (discretized cells).

Required:	No
Type:	Boolean
Default:	False

Axes

Enables the display of the spatial axes, and the colorbar.

Required:	No
Type:	Boolean
Default:	True

Timestamp

Enables the display of the timestamp.

Required:	No
Type:	Boolean
Default:	True

Gradients

Enables the display of gradients.

Required:	No
Type:	Boolean
Default:	False

Contours

Enables the display of contours.

Required:	No
Type:	Boolean
Default:	True

Contour Labels

Enables the display of contour labels.

Required:	No
Type:	Boolean
Default:	False

Number of Contours

Specifies the number of countours to generate between the values specified in *Output Range*.

Required:	No
Type:	Integer
Default:	13

2.4 Foundation (Advanced)

These are additional inputs that can be used to further define the *Foundation* object. These inputs are largely intended for advanced research or testing purposes.

2.4.1 Far-Field Width

Distance from the interior wall surface to the edge of the domain.

Required:	No
Type:	Numeric
Units:	m
Default:	40

2.4.2 Deep-Ground Depth

Distance from the grade level to the bottom of the domain.

Required:	No
Type:	Numeric
Units:	m
Default:	40

2.4.3 Deep-Ground Boundary Condition

Specifies the type of boundary condition to apply at the deep-ground boundary. Options are:

- `ZERO-FLUX`, which applies a zero heat flux boundary,
- `AUTO`, which applies a constant temperature equal to the average outdoor dry-bulb temperature from the weather file, and
- `CONSTANT-TEMP`, which applies a user-specified constant temperature (see *Deep-Ground Temperature*).

Required:	No
Type:	Enumeration
Values:	`ZERO-FLUX`, `AUTO`, or `CONSTANT-TEMP`
Default:	`ZERO-FLUX`

2.4.4 Deep-Ground Temperature

If *Deep-Ground Boundary Condition* is `CONSTANT-TEMP`, then this value specifies the temperature applied to the deep-ground boundary.

Required:	Depends
Type:	Numeric
Units:	K

2.4.5 Convection Calculation Method

Specifies how convection coefficients are calculated. Options are:

- `AUTO`, which calculates dynamic convection coefficients based on temperature difference, wind speed, and wind direction.
- `CONSTANT`, which applies a user-specified convection coefficients to interior and exterior surfaces (see *Interior Convection Coefficient* and *Exterior Convection Coefficient*). This is used primariliy for IEA BESTEST calculations.

Required:	No
Type:	Enumeration
Values:	`AUTO` or `CONSTANT`
Default:	`AUTO`

2.4.6 Interior Convection Coefficient

If *Convection Calculation Method* is `CONSTANT`, then this value specifies the convection coefficient applied to interior surface boundaries (slab floor, interior foundation wall, and interior insulation).

Required:	Depends
Type:	Numeric
Units:	W/m^2-K

2.4.7 Exterior Convection Coefficient

If *Convection Calculation Method* is `CONSTANT`, then this value specifies the convection coefficient applied to exterior surface boundaries (grade, exterior foundation wall, and exterior insulation).

Required:	Depends
Type:	Numeric
Units:	W/m^2-K

2.4.8 Wall Top Boundary Condition

Specifies how the boundary condition along the wall top is calculated. Options are:

- `ZERO-FLUX`, which applies a zero heat flux boundary condition along the wall top. This implies that heat flux above the wall top is one dimensional and does not flow through the wall top boundary.
- `LINEAR-DT`, which applies a linear change in temperature across the wall top (see *Wall Top Temperature Difference*). This is used only to represent the constraints of the IEA BESTEST analytical solution in case GC10a.

Required:	No
Type:	Enumeration
Values:	`ZERO-FLUX` or `LINEAR-DT`
Default:	`ZERO-FLUX`

2.4.9 Wall Top Temperature Difference

If *Wall Top Boundary Condition* is `LINEAR-DT`, then this value specifies the change in temperature across the wall. This is used only to represent the constraints of the IEA BESTEST analytical solution in case GC10a. The actual temperatures are determined based on the values of *Indoor Air Temperature* and *Outdoor Air Temperature*.

Required:	Depends
Type:	Numeric
Units:	K

2.4.10 Number of Dimensions

Switch between two- and three-dimensional simulations. **Warning:** Three-dimensional simulations can be very computationally intensive. Kiva does not impose any limitations, but be warned: some three-dimensional simulations may require more memory than most computers have available.

Required:	No
Type:	Integer
Constraints:	2 or 3
Default:	2

2.4.11 Coordinate System

Allows the user to specify the coordinate system used for calculations. For Three-dimensional simulations, this must be `CARTESIAN`.

Required:	No
Type:	Enumeration
Values:	`CARTESIAN` or `CYLINDRICAL`
Default:	`CARTESIAN`

2.4.12 Two-Dimensional Approximation

These are methods of approximating three-dimensional foundation heat transfer using a two-dimensional coordinate system. Options are:

- `AP`: Creates an infinite rectangle (*Coordinate System* = `CARTESIAN`) or a circle (*Coordinate System* = `CYLINDRICAL`) with the same *area-to-perimeter* ratio as the three-dimensional *Polygon*.
- `RR`: Creates the straight section (*Coordinate System* = `CARTESIAN`) or a rounded cap (*Coordinate System* = `CYLINDRICAL`) of a rounded rectangle with the same area and perimeter as the three-dimensional *Polygon*.
- `BOUNDARY`: Creates an infinite rectangle (*Coordinate System* = `CARTESIAN`) or a circle (*Coordinate System* = `CYLINDRICAL`) with an adjusted *area-to-perimeter* ratio from the three-dimensional *Polygon*. Adjustments are made to represent concave corners of the *Polygon*.
- `CUSTOM`: Creates an infinite rectangle (or parallel infinite rectangles) (*Coordinate System* = `CARTESIAN`) or a circle (or concentric circles) (*Coordinate System* = `CYLINDRICAL`) based on the specification of *Length 1* and *Length 2*. This is an experimental feature where the three-dimensional *Polygon* is not used directly to define the approximation.

Required:	No
Type:	Enumeration
Values:	`AP`, `RR`, `BOUNDARY`, or `CUSTOM`
Default:	`BOUNDARY`

2.4.13 Length 1

Represents the outter width of an infinite rectangle (*Coordinate System* = `CARTESIAN`) or outer radius of a circle (*Coordinate System* = `CYLINDRICAL`).

Required:	Depends
Type:	Numeric
Units:	m

2.4.14 Length 2

Represents the inner width of parallel infinite rectangles (*Coordinate System* = `CARTESIAN`) or inner radius of concentric circles (*Coordinate System* = `CYLINDRICAL`).

Required:	Depends
Type:	Numeric
Units:	m

2.4.15 Use Symmetry

For three-dimensional simulations Kiva can detect planes of symmetry and automatically reduce the simulation domain by half (for a single plane of symmetry), or three quarters (for two planes of symmetry). If this is the case, then direction dependent boundary conditions such as incident solar and wind driven convection are averaged for the symmetric unit.

Required:	No
Type:	Boolean
Default:	True

2.4.16 Mesh

The mesh is a compound object that defines the refinement of cells within the calculation domain. Cells are defined within four distinct region types of the domain. The region bounding the foundation wall and insulation elements defines the near-field region. All other regions are defined either laterally (interior and far-field regions) or vertically (deep-field region) relative to the near-field region.

Cells grow geometrically towards the far-field, deep-ground, and symmetry boundaries. Cells grow towards the center of each interior region and within each structural or insulation component. Example meshes are shown in the following figures.

Example:

```
Mesh:
  Minimum Cell Dimension: 0.02
  Maximum Near-Field Growth Coefficient: 1.5
  Maximum Deep-Field Growth Coefficient: 1.5
  Maximum Interior-Field Growth Coefficient: 1.5
  Maximum Far-Field Growth Coefficient: 1.5
```

Required:	No
Type:	Compound object

Minimum Cell Dimension

The minimum cell dimension defines the smallest possible dimension of a cell within the domain. This defines the allowable number of cells between element or region boundaries. The cells' dimensions are allowed to increase to fit within a component or region boundaries according to the growth and distribution of the cells.

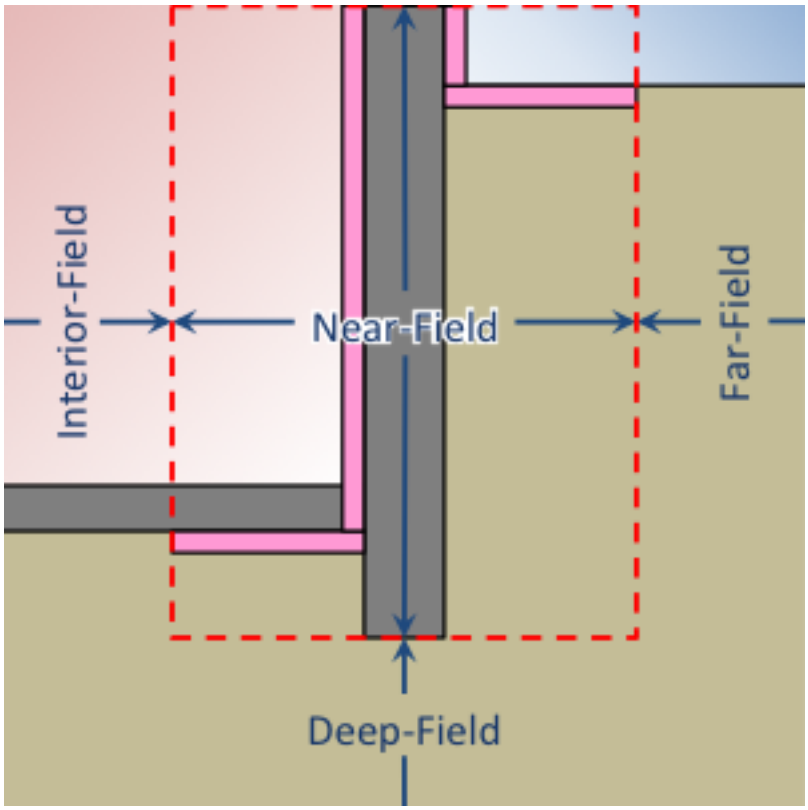

Fig. 2.6: Illustration of regions (profile view)

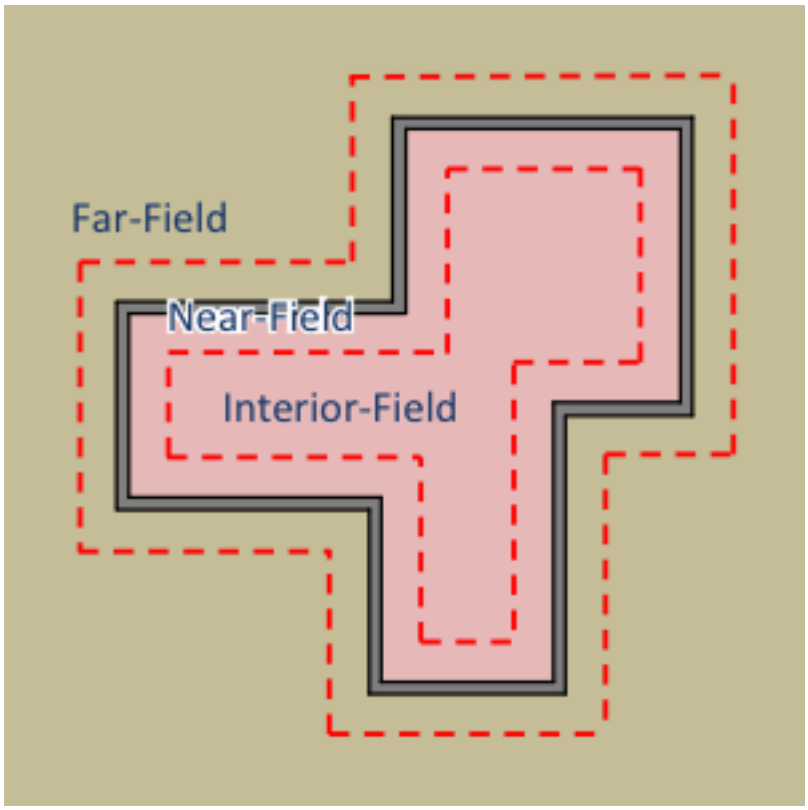

Fig. 2.7: Illustration of regions (plan view)

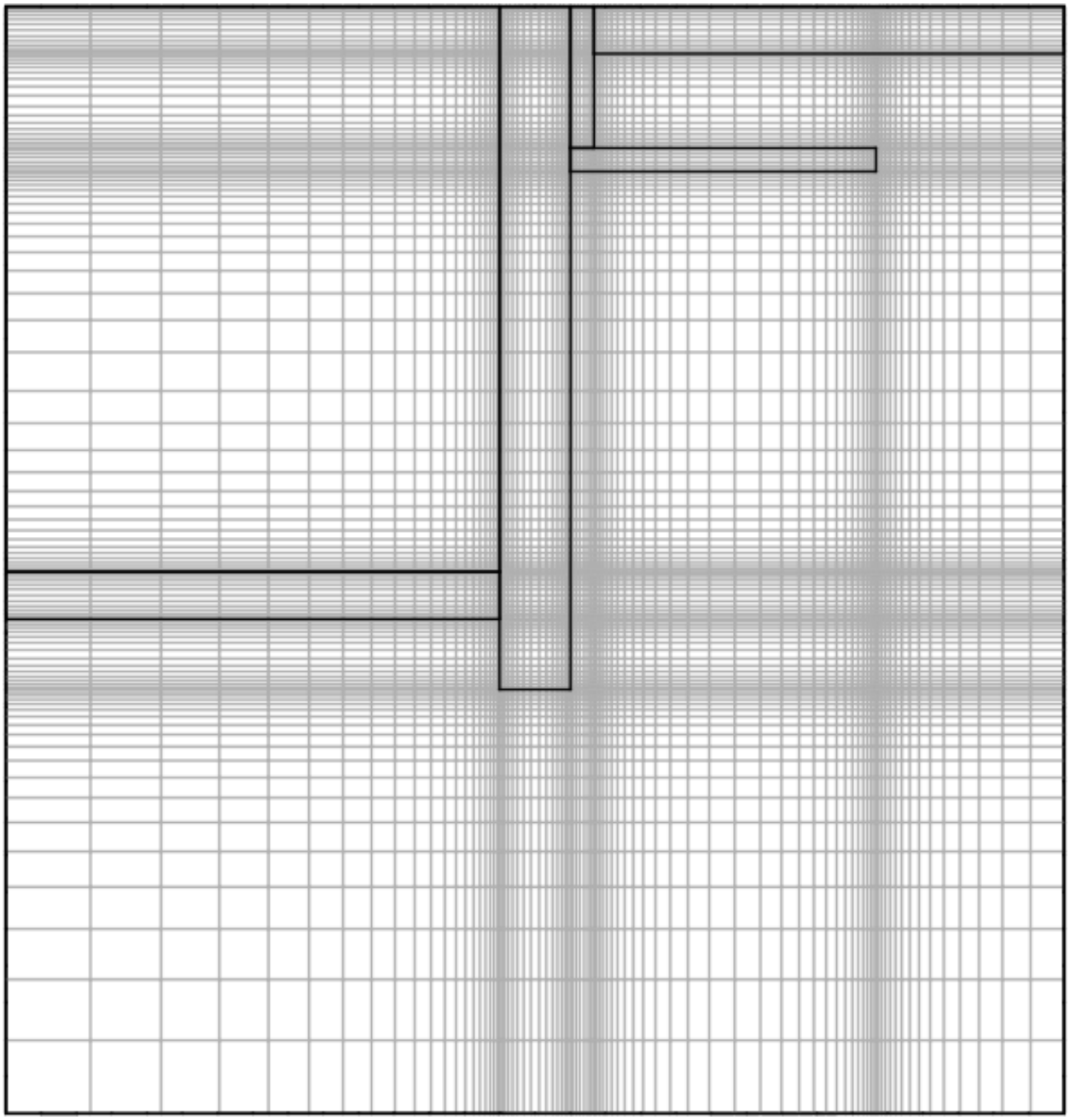

Fig. 2.8: Profile view of a basement mesh

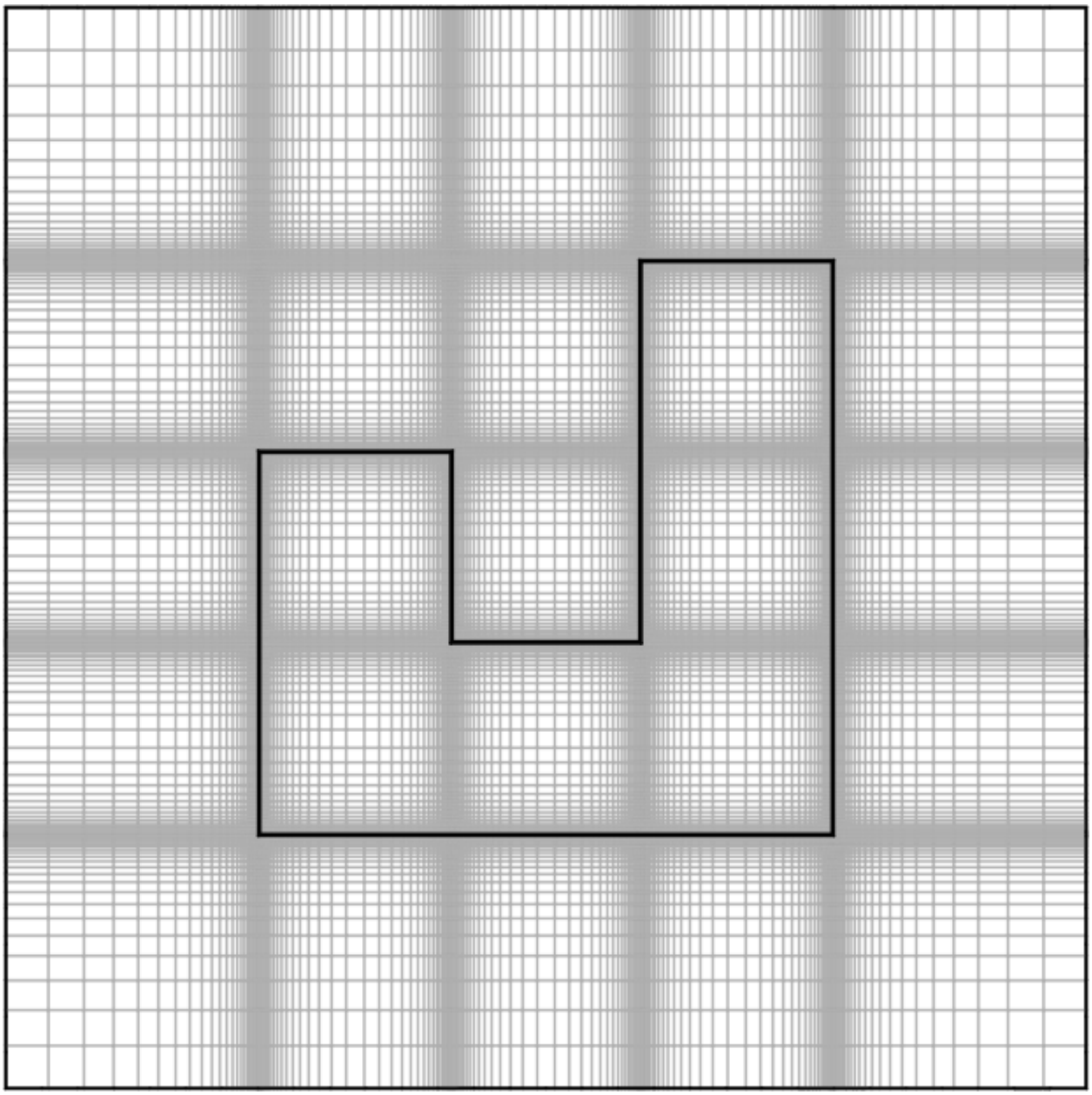

Fig. 2.9: Plan view of a foundation mesh

Required:	No
Type:	Numeric
Units:	m
Default:	0.02

Maximum Near-Field Growth Coefficient

The maximum size increase between neighboring cells within the near-field region.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	1.5

Maximum Deep-Field Growth Coefficient

The maximum size increase between neighboring cells within the deep-field region.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	1.5

Maximum Interior-Field Growth Coefficient

The maximum size increase between neighboring cells within the interior-field region.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	1.5

Maximum Far-Field Growth Coefficient

The maximum size increase between neighboring cells within the far-field region.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	1.5

2.4.17 Numerical Scheme

This defines the numerical scheme used for calculating domain temperatures for successive timesteps. Options are:

- `IMPLICIT`, a fully implicit scheme with unconditional stability using an iterative solver,
- `EXPLICIT`, an explicit scheme with conditional stability,
- `CRANK-NICOLSON`, a partially implicit scheme with unconditional stability using an iterative solver (may exhibit oscillations),

- `ADI`, a scheme that solves each direction (X, Y, and Z) implicitly for equal sized sub-timesteps. The other two directions are solved explicitly. This allows for an exact solution of the linear system of equations without requiring an iterative solver. This scheme is extremely stable,
- `ADE`, a scheme that sweeps through the domain in multiple directions using known neighboring cell values. This scheme is very stable,
- `STEADY-STATE`, domain temperatures are calculated independently of previous timesteps using a steady-state solution from an iterative solver. This is often slower and less accurate than other methods.

Required:	No
Type:	Enumeration
Values:	`IMPLICIT`, `EXPLICIT`, `CRANK-NICOLSON`, `ADI`, `ADE`, or `STEADY-STATE`
Default:	`ADI`

2.4.18 f-ADI

When *Numerical Scheme* is `ADI`, this defines the weighting between the implicit, and explicit solutions in the sub-timesteps. In general, it is best to make this number very small.

Required:	No
Type:	Numeric
Units:	dimensionless
Default:	0.00001

2.4.19 Solver

Kiva uses Lis [1] (Library of Iterative Solvers) to solve any schema that requires an iterative solution. Refer to their user guide for details on the value options.

Required:	No
Type:	Enumeration
Values:	See [3].
Default:	`bicgstab`

2.4.20 Preconditioner

Preconditioners are used to help solvers find solutions faster. Again the options here come from the Lis documentaiont [3].

Required:	No
Type:	Enumeration
Values:	See [3].
Default:	`ilu`

2.4.21 Maximum Iterations

Maximum number of iterations allowed in search for a solution.

Required:	No
Type:	Integer
Default:	100000

[1] The Scalable Software Infrastructure Project. 2014. *Lis User Guide*. The Scalable Software Infrastructure Project, Fukuoka, Japan.

2.4.22 Tolerance

Tolerance is defined as the relative ℓ^2-norm of the residual when solving the linear system of equations.

Required:	No
Type:	Numeric
Units:	Dimensionless
Default:	1.0e-6

2.4.23 Initialization Method

The initialization method determines how the initial temperatures in the domain are set. Options are:

- `CONSTANT`, spatially-constant initial temperature,
- `KUSUDA`, a one-dimensional analytical solution developed by that provides temperature variation as a function of depth driven by an annual harmonic temperature fluctuation. There is no temperature variation in horizontal dimensions,
- `STEADY-STATE`, a steady-state solution scheme initializes the temperatures with the first timestep's boundary conditions. This provides an initial condition temperature variation in all dimensions.

Required:	No
Type:	Enumeration
Values:	`CONSTANT`, `KUSUDA`, or `STEADY-STATE`
Default:	`STEADY-STATE`

2.4.24 Initial Temperature

When *Initialization Method* is `CONSTANT` this specifies the temperature to use.

Required:	Depends
Type:	Numeric
Units:	K

2.4.25 Accelerated Initialization Timestep

An accelerated initialization begins with the user-defined *Initialization Method* and calculates new domain temperatuers prior to the beginning of the simulation using long timesteps (on the order of days, weeks, or months). These timesteps are calculated using a fully implicit, unconditionally stable numerical scheme. This allows the simulation to build a history of temperatures without requiring a signficant amount of additional calculations. The defualt, one week, was found to give very accurate initial temperatures.

Required:	No
Type:	Integer
Units:	days
Default:	168

2.4.26 Number of Accelerated Initialization Timesteps

This specifies the number of timesteps (of the size specified by *Accelerated Initialization Timestep*) to calculate prior to the beginning of the simulation.

Required:	No
Type:	Integer
Default:	12

2.4.27 Number of Warmup Days in Initialization

Additional days of initialization can be calculated using the *Timestep* and *Numerical Scheme* defined by the user. This input specifies the number of days the domain is simulated under these conditions after the accelerated initialization timesteps, but prior to the *Start Date* specified in the *Simulation Control*.

CHAPTER 3

Methodology

Coming sometime (soon?)! In the meantime, refer to Kruis (2015) [1].

[1] Kruis, N. 2015. *Development and Application of a Numerical Framework for Improving Building Foundation Heat Transfer Calculations*. Ph.D. Dissertation. Department of Civil, Environmental, and Architectural Engineering. University of Colorado, Boulder.

CHAPTER 4

Testing

Kiva has been tested against the *International Energy Agency Building Energy Simulation Test (IEA BESTEST) in-depth diagnostic cases for ground coupled heat transfer related to slab-on-grade construction* [1]. The results match very closely with the analytical truth standard and the verified numerical truth standards:

Steady-State Results:

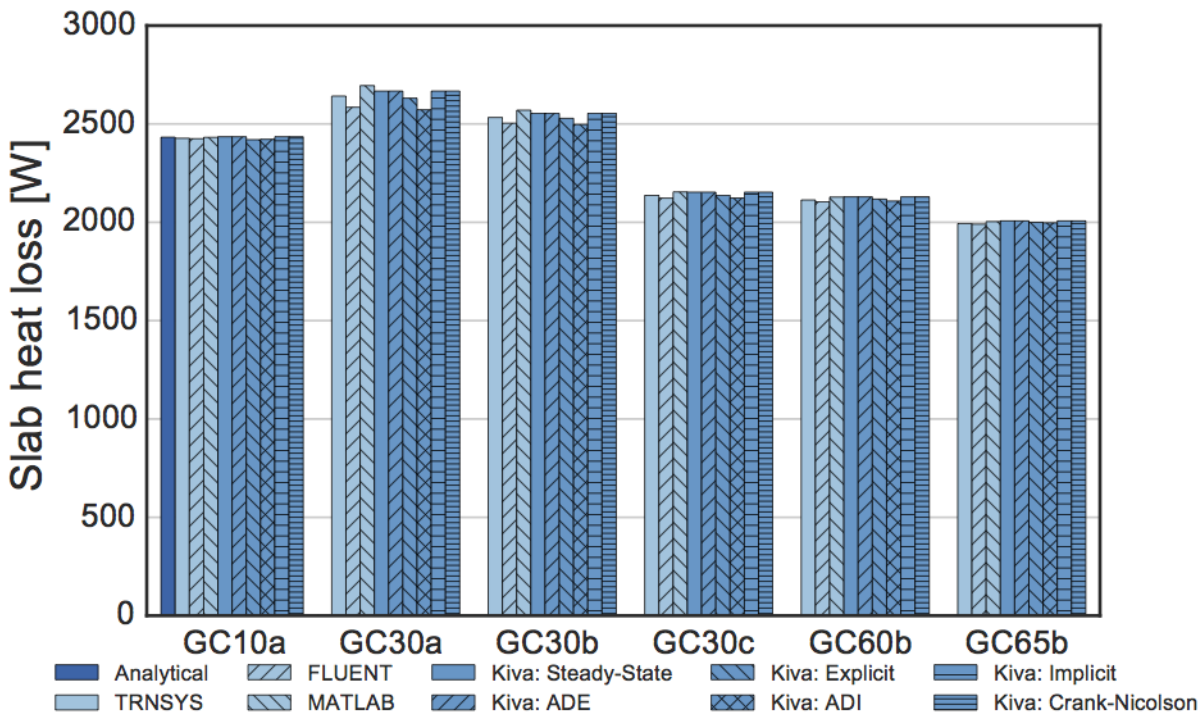

Fig. 4.1: IEA BESTEST Steady-State Test Case Results

Harmonic (Unsteady) Results:

[1] Neymark J., Judkoff R. 2008. *International Energy Agency Building Energy Simulation Test and Diagnostic Method (IEA BESTEST): In-Depth Diagnostic Cases for Ground Coupled Heat Transfer Related to Slab-on-Grade Construction*. Technical Report September, National Renewable Energy Laboratory, Golden, Colorado.

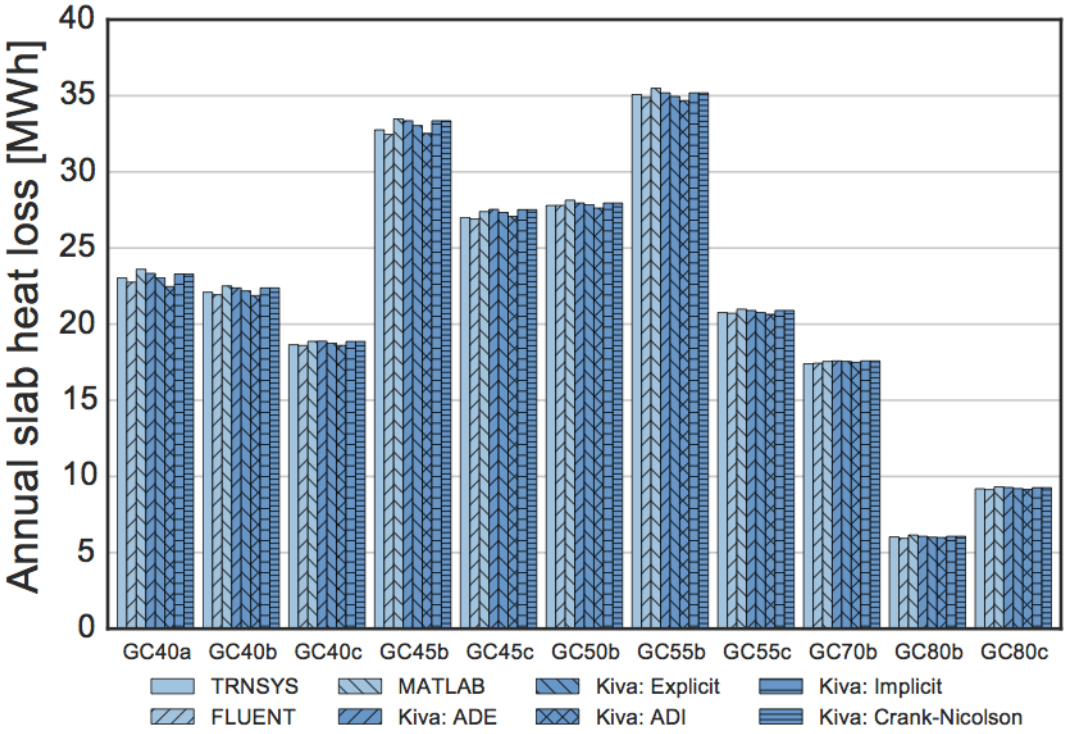

Fig. 4.2: IEA BESTEST Harmonic Test Case Results

CHAPTER 5

Bibliography