

DELPHIN 6 TUTORIAL

ERSTELLUNG UND AUSWERTUNG VON 3D SIMULATIONSMODELLEN

25. November 2019

Zusammenfassung

Dieses Tutorial zeigt am Beispiel eines idealisierten Erdkollektors, wie ein vertikal verbauter Kollektor in einem 2D Modell abgebildet werden kann. Danach wird das Modell um die dritte Dimension erweitert. Dabei kommen die Kommandozeilenprogramme `MoveToLayer.py` und `CmdDiscretize` zum Einsatz. Die Auswertung erfolgt im PostProc anhand von Schnittoperation. Im Tutorial wird hierbei die Verwendung der Schnitt-Extraktoren gezeigt. Weiterhin wird die Verwendung des Programms `DSixOutputConverter` gezeigt, um danach die 3D Ergebnisse in TecPlot auswerten zu können.

Inhaltsverzeichnis

1	Idealisiertes Erdkollektormodell	2
1.1	2D Modell	2
1.2	2D Ergebnisse	2
2	Transformation in das 3D Modell	3
2.1	Erstellen des 3D Simulationsmodells	3
2.1.1	Vervielfachen der 2D Schicht	4
2.1.2	Definition der z-Schicht-Dicken	4
2.1.3	Entfernen der Kollektor-Feldbedingung in Schicht 0 und 2	5
2.2	3D-Gittergenerierung	5
3	Durchführen und Prüfen der 3D Simulation	6
3.1	Solvereinstellungen für große 3D Projekte	6
3.2	Analyse des generierten Gitters und Sichtung der Ausgaben	6
3.3	Analyse mit TecPlot	7
3.4	Analyse mit PostProc 2	9

1 Idealisiertes Erdkollektormodell

1.1 2D Modell

Abbildung 1 zeigt das Modell eines Erdkollektors, wobei das Kollektorfeld idealisiert als Schicht eines Feldes dargestellt sind (siehe Datei `2D_Basisvariante.d6p`). Rechts- und links gibt es weitere Kollektoren im gleichen Abstand, weswegen rechts- und linksseitig Symmetriebedingungen gelten. Auf der Oberseite liegt konstant 5°C an, auf der Unterseite wird ebenso wie links und rechts keine Randbedingung angeordnet. Die Simulation ist explizit simpel (und unrealistisch) aufgesetzt, damit die Ergebnisse der 2D und 3D Rechnung einem Plausibilitätscheck unterworfen werden können.



Abbildung 1: Ausschnitt aus einem idealisierten vertikalen Kollektorfeld.

Im Kollektorfeld selbst (10x1000 mm Abmessung) wird eine konstante Entzugsleistung von -1000 W/m^3 definiert, welches einer längenbezogenen Last von -10 W/m entspricht (geht man von einer wiederholten Anordnung eines vertikalen Kollektors im Raster von ca. 0,5 m aus, so entspricht dies einer flächenbezogenen Entzugslast von -20 W/m^2). Die Simulation wird rein thermisch durchgeführt, beginnend von einer uniformen Anfangstemperatur von 5°C . Ein einzelner Temperatursensor ist in der Mitte des Feldes, direkt neben dem Kollektor angelegt.

1.2 2D Ergebnisse

Bei 2D Modell wird in z-Richtung von einem unendlich langen Kollektor ausgegangen, d.h. regenerierende Wärmeströme von „vorne“ und „hinten“ sind im Modell nicht berücksichtigt. Daher wird eine größere Temperaturabsenkung berechnet, als in der Realität zu erwarten wäre.

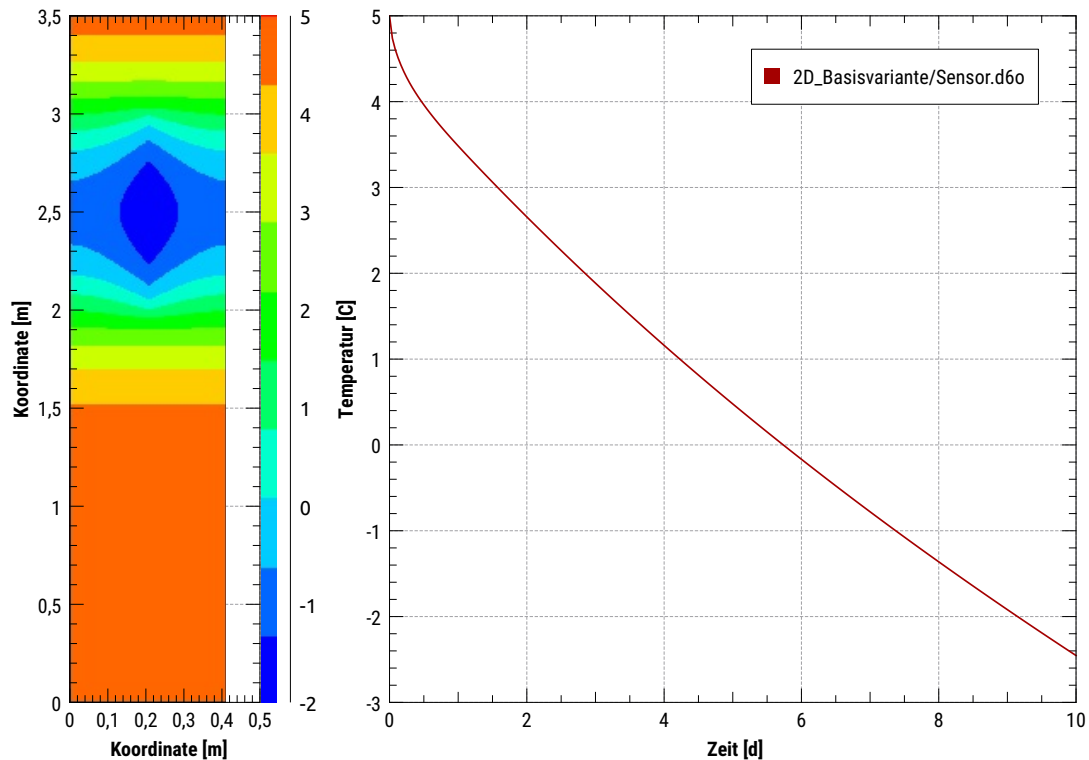


Abbildung 2: Temperaturfeld nach 10 Tagen und Sensortemperatur

2 Transformation in das 3D Modell

Das 3D Modell des Kollektors soll in z-Richtung vor und hinter dem Kollektorfeld je eine Bodenschicht von 4 m angeordnet werden. Das Kollektorfeld selbst soll eine Ausdehnung in z-Richtung von 5 m haben. Damit hat das 3D Modell also 3 Schichten in z-Richtung. Das einfachste Vorgehen ist daher, die bisherige Schicht einfach dreimal hintereinander zu definieren.

2.1 Erstellen des 3D Simulationsmodells

Zur Veranschaulichung der Schichten und Nummerierungen wird zunächst das Berechnungsraster aus der Basisvariante entfernt. Es sind folgende Schichten definiert:

```
<Discretization>
  <XSteps unit="m">0.2 0.01 0.2 </XSteps>
  <YSteps unit="m">2 1 0.5 </YSteps>
</Discretization>
```

mit den folgenden Zuweisungen:

```
<Assignment type="Output" location="Element">
  <Reference>Temperaturprofil</Reference>
  <Range>0 0 2 2</Range>
</Assignment>

<Assignment type="Material" location="Element">
  <Reference>Boden</Reference>
  <Range>0 0 2 2</Range>
</Assignment>

<Assignment type="Interface" location="Top">
  <Reference>Warm</Reference>
  <Range>0 2 2 2</Range>
</Assignment>

<Assignment type="FieldCondition" location="Element">
  <Reference>Konstante Entzugsleistung</Reference>
  <Range>1 1 1 1</Range>
</Assignment>

<Assignment type="Output" location="Coordinate">
```

```
<Reference>Sensor</Reference>
<IBK:Point3D>0.1995 2.5 0</IBK:Point3D>
</Assignment>
```

2.1.1 Vervielfachen der 2D Schicht

Im ersten Schritt wird diese einzige Schicht in z-Richtung (standardmäßig mit Index 0, welcher in den **Range**-Tags nicht angegeben wird) in zwei zusätzliche Schichten kopiert. Die mittlere Schicht (Index 1) behält den Kollektor, während die Schicht 0 (hinterste) und Schicht 2 (vorderste) den umlaufenden Boden abbilden.

Zum Verschieben einer Schicht (bzw. Anpassen der z-Indexe in den Zuweisungen) dient das Skript **MoveToLayer.py**, welches als Python-Skript wie folgt ausgeführt wird:

```
> python MoveToLayer.py -z 1 2D_BasisVariante.d6p 3DModell_z1.d6p
```

Hierbei gibt das Argument hinter **-z** an, welcher z-Schichtindex verwendet werden soll. Das Script selbst modifiziert lediglich die **<Assignment>** XML Tags. Achtung: wird ein anderer Schichtindex als 0 (der Standard) verwendet, ist die entstehende DELPHIN Simulationsdatei so noch nicht lauffähig, da die Definition der Schichtdicken in z-Richtung noch fehlt (siehe unten). Nach Verschieben in Schicht 2 sieht der Zuweisungsblock z.B. so aus:

```
<Assignment type="Output" location="Element">
  <Reference>Temperaturprofil</Reference>
  <Range>0 0 1 2 2 1</Range>
</Assignment>

<Assignment type="Material" location="Element">
  <Reference>Boden</Reference>
  <Range>0 0 1 2 2 1</Range>
</Assignment>

<Assignment type="Interface" location="Top">
  <Reference>Warm</Reference>
  <Range>0 2 1 2 2 1</Range>
</Assignment>

<Assignment type="FieldCondition" location="Element">
  <Reference>Konstante Entzugsleistung</Reference>
  <Range>1 1 1 1 1 1</Range>
</Assignment>

<Assignment type="Output" location="Coordinate">
  <Reference>Sensor</Reference>
  <IBK:Point3D>0.1995 2.5 0</IBK:Point3D>
</Assignment>
```

Die Zuweisungsbereiche haben nun das Format **<x_left, y_bottom, z_back, x_right, y_top, z_front>** (beachte, rechtshändisches Koordinatensystem, z-Achse zeigt nach vorne).

Das macht man nun 3 mal, für die Schicht-Indexe 0, 1 und 2. Es entstehen 3 Dateien (**3DModell_z0.d6p**, **3DModell_z1.d6p** und **3DModell_z2.d6p**). Man wählt eine Datei aus, und kopiert alle **<Assignment>** xml Tags der anderen Dateien zusammen in die Datei.

Nun kann man Duplikate herauslöschen, z.B. die Sensorausgabe, welche jetzt 3 Mal an der gleichen Koordinate zugewiesen wurde. Auch die Temperaturfeldausgabe lässt sich vereinfachen, indem die erste Zuweisung auf den gesamten z-Bereich ausgedehnt wird:

```
<Assignment type="Output" location="Element">
  <Reference>Temperaturprofil</Reference>
  <Range>0 0 0 2 2 2</Range>
</Assignment>
```

2.1.2 Definition der z-Schicht-Dicken

Als nächstes fügt man noch die Information der Schichtdicken in z-Richtung hinzu. Dazu wird das **Discretization**-XML-Tag erweitert:

```
<Discretization>
  <XSteps unit="m">0.2 0.01 0.2 </XSteps>
  <YSteps unit="m">2 1 0.5 </YSteps>
  <ZSteps unit="m">4 5 4</ZSteps>
</Discretization>
```

Zu diesem Zeitpunkt ist die Simulationsdatei vollständig und nach einer Diskretisierung (siehe Abschnitt 2.2) lauffähig. Als Test kann man die Simulation auch einmal durchlaufen lassen. Die berechneten Sensortemperaturen müssten mit denen der 2D Rechnung übereinstimmen.

2.1.3 Entfernen der Kollektor-Feldbedingung in Schicht 0 und 2

Im endgültigen 3D-Modell dürfen die Bodenschichten hinter und vor dem Kollektor selbst keine Feldbedingungen zugewiesen bekommen. Daher müssen diese Zuweisungen entfernt werden. Zum Schluss darf nur noch eine einzige Feldbedingungs-zuweisung existieren. Wenn man auch die anderen Zuweisungen (Materialien, Randbedingung oben) sinnvoll zusammenfasst, ergibt sich folgende Zuweisungsliste:

```
<Assignment type="Output" location="Element">
  <Reference>Temperaturprofil</Reference>
  <Range>0 0 0 2 2 2</Range>
</Assignment>

<Assignment type="Material" location="Element">
  <Reference>Boden</Reference>
  <Range>0 0 0 2 2 2</Range>
</Assignment>

<Assignment type="Interface" location="Top">
  <Reference>Warm</Reference>
  <Range>0 2 0 2 2 2</Range>
</Assignment>

<Assignment type="Output" location="Coordinate">
  <Reference>Sensor</Reference>
  <IBK:Point3D>0.1995 2.5 7.5</IBK:Point3D>
</Assignment>

<Assignment type="FieldCondition" location="Element">
  <Reference>Konstante Entzugsleistung</Reference>
  <Range>1 1 1 1 1 1</Range>
</Assignment>
```

Bei der entstandenen 3x3x3 Geometrie wird die Feldbedingung nun genau dem Mittelelement zugewiesen.

Achtung: die z-Koordinate des Temperatursensors muss in die Mitte der mittleren Schicht, d.h. an Koordinate $4+5/2 = 7,5$ m gelegt werden. Sonst sind die Ergebnisse mit der 2D Basisvarianten gar nicht mehr vergleichbar.

2.2 3D-Gittergenerierung

Für die Gittergenerierung kommt nun das Kommandozeilentool CmdDiscretize zum Einsatz. Es bietet in etwa die Möglichkeiten der Gittergenerierung, wie die Programmoberfläche, kann aber individueller gesteuert werden. Die Argumente sind daher vielfältig, ein Aufruf mit `--help` gibt einen Überblick.

Hilfreich ist folgender Aufruf zum Entfernen eines eventuell vorhandenen Gitters (automatisch diskretisieren, minimale Schichtbreite wird so groß gesetzt, da keine Schicht diskretisiert werden darf).

```
> CmdDiscretize -m=auto -x=20 -y=20 -z=20 3D_Modell.d6p
```

Dabei wird die angegebene Projektdatei gelesen, verändert und wieder unter gleichem Namen geschrieben.

Eine sinnvolle Kommandozeile für das Diskretisieren dieses 3D Modells wäre (automatische Diskretisierung, variabel, maximale Elementdicken in x und y-Richtung je 10 cm, in z-Richtung 50 cm).

Tip: Das zusätzliche Argument `-o <Ausgabedatei>` sorgt dafür, dass die originale Projektdatei nicht verändert wird. Das ist hilfreich, wenn durch Angabe ungünstiger Parametrisierungseinstellungen gigantische Gitter und Projektdateien erstellt werden. Durch das Schreiben einer neuen Projektdatei wird so die Ausgangsprojektdatei nicht zerstört.

```
> CmdDiscretize -m=auto -X=0.1 -Y=0.1 -Z=0.5 3D_Modell.d6p -o=3D_ModellDisc.d6p
```

Das Tool liefert dabei folgende Ausgabe:

```
DELPHIN 6 Project Discretiser/Grid Generator, Version 2.1
Bauklimatik Dresden Software GmbH
https://www.bauklimatik-dresden.de
All rights reserved.
Using automatic discretization.
Discretizing range: [0, 0, 0] - [2, 2, 2]
Using stretch factors: 1.25/1.25/1.25
Discretising in all directions
Using minimum element widths: 0.001/0.001/0.001
Using maximum element widths: 0.1/0.1/0.5
Updating assignment grid...
Discretization range after removal of existing grid: [0, 0, 0] - [2, 2, 2]
Discretising now...
Running in auto-discretization mode
  Discretising in X-direction
    Updating boundary ID grid
    Discretization columns
      Discretizing column 0 (dx=0.2) with discMode 2 into 19 elements
      Discretizing column 1 (dx=0.01) with discMode 3 into 7 elements
      Discretizing column 2 (dx=0.2) with discMode 1 into 19 elements
    Discretization range after discretization step: [0, 0, 0] - [44, 2, 2]
    Updating assignment grid...
  Discretising in Y-direction
    Updating boundary ID grid
    Discretization rows
      Discretizing row 0 (dx=2) with discMode 2 into 37 elements
      Discretizing row 1 (dx=1) with discMode 3 into 44 elements
      Discretizing row 2 (dx=0.5) with discMode 3 into 37 elements
    Discretization range after discretization step: [0, 0, 0] - [44, 117, 2]
    Updating assignment grid...
  Discretising in Z-direction
    Updating boundary ID grid
    Discretization stack planes
      Discretizing stack plane 0 (dx=4) with discMode 2 into 32 elements
      Discretizing stack plane 1 (dx=5) with discMode 3 into 58 elements
      Discretizing stack plane 2 (dx=4) with discMode 1 into 32 elements
    Discretization range after discretization step: [0, 0, 0] - [44, 117, 121]
    Updating assignment grid...
Grid successfully discretised, updating assignment grid and counting elements...
Elements (all/used): 647820/647820
Min/Max element dimensions in discretisation range:
  min: 0.001/0.001/0.001
  max: 0.0226101/0.0981449/0.484012
Min/Max element dimensions in entire grid:
  min: 0.001/0.001/0.001
  max: 0.0226101/0.0981449/0.484012
Writing project to '3D_ModellDisc.d6p'
```

Es sind hier schon beeindruckend viele Elemente. Waren es bei der 2D Berechnung nur 5330, sind es jetzt bereits 647820 Elemente. Dennoch ist das für DELPHIN bei korrekt eingestellten Numerikparametern kein Problem.

3 Durchführen und Prüfen der 3D Simulation

3.1 Solvareinstellungen für große 3D Projekte

Während bei einer rein thermischen 2D Rechnung fast immer KLU als direkter linearer Gleichungssystemlöser am schnellsten ist, so ist bei Elementzahlen > 50000 kein direktes Verfahren mehr zielführend. Es sollte daher folgendes in den Simulationseinstellungen eingestellt sein:

- Löser für das lineare Gleichungssystem: BiCGStab
- Vorkonditionierer: ILU

Und für den Fall, dass 3D Felddaten erzeugt werden, sollte auf jeden Fall das Schreiben von binären Ausgabedateien aktiviert werden. Die 10-tägige Simulation unter konstanten Randbedingungen dauert bei 6 Kernen damit 100 Sekunden (zum Vergleich: die 2D Rechnung war nach 0.3 Sekunden fertig).

3.2 Analyse des generierten Gitters und Sichtung der Ausgaben

Man kann natürlich sofort die Sensorwerte analysieren, und erhält erstaunlich ähnliche Ergebnisse.

3.3 Analyse mit TecPlot

TecPlot ist nur eines der Analyseprogramme für 3D Felder. Für die Konvertierung der 3D Ergebnisdaten von DELPHIN kann das Kommandozeilentool **DSixOutputConverter** verwendet werden. Zunächst kann der Datensatz analysiert werden:

```
> DSixOutputConverter list 3D_Modelldisc/results/Temperaturprofil.d6b
```

zeigt folgende Analyse:

```
Reading data file '3D_Modelldisc/results/Temperaturprofil.d6b'.
 1 ms for reading input DataIO file.
 15 ms for parsing 7 datasets.
Reading geometry file '3D_Modelldisc/results/3D_Modelldisc_3673700722.g6b'.
Reading      : 100% [#####] 68 ms for reading input
geometry file.

--- 3D_Modelldisc/results/Temperaturprofil.d6b ---
Output type   : FIELD
Quantity      : Temperature
Space type    : SINGLE
Time type     : NONE
Value unit    : C
Time unit     : h
Start year    : 2007
Data format   : 5D [[time, matrix],[...],...]
Elements      : 647820
Materials     : 1
Geometry      : 3D structure, 45 columns, 118 rows
Time points   : 7
-----
```

Die Datei kann mit folgendem Befehl ins TecPlot Format kopiert werden (--3DFE gibt dabei an, dass Blockelemente (Brick) verwendet werden sollen).

```
> DSixOutputConverter tecplot --3DFE 3D_Modelldisc/results/Temperaturprofil.d6b
```

Dabei wird die Datei **Temperaturprofil.dat** erstellt. Man kann auch nur den letzten Zeitpunkt konvertieren. In der Ergebnisdatei gibt es genau 7 Zeitpunkte (der letzte hat den TimeIndex=6). Nun können wir den letzten Zeitpunkt ausschneiden und in das TecPlot Format konvertieren:

```
> DSixOutputConverter tecplot --3DFE 3D_Modelldisc/results/Temperaturprofil.d6b "--timeindex=6,6"
```

Hinweis: das Argument "--timeindex=6,6" muss in Anführungszeichen stehen, da unter Linux/MacOS der Befehlszeileninterpreter dies sonst als Kommando versteht.

Die erstellte **.dat**-Datei kann nun in TecPlot eingelesen werden. Neben den üblichen Profilschnitten sind 2 Dinge interessant:

- ob das Gitter sinnvoll erstellt wurde
- ob die räumliche Temperaturverteilung sinnvoll ist

Ersteres erhält man durch Ausgabe des Berechnungsrasters (Abbildung 3).

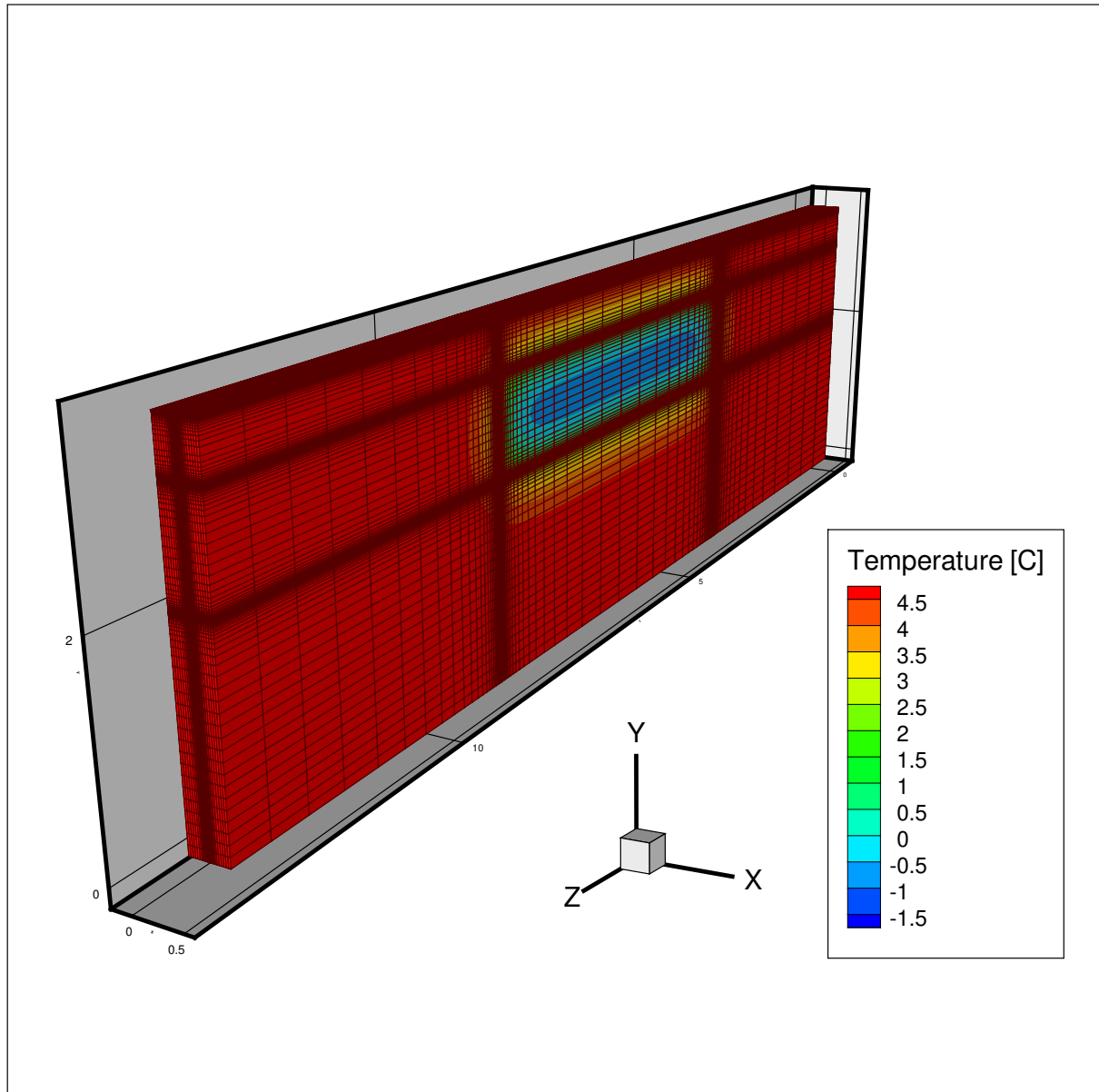


Abbildung 3: Berechnungsraster und außenseitige Temperaturverteilung

Man kann auch Isoprofile zeichnen, z.B. bei -0.5 °C und 4 °C , sowie vertikale Schnitte (z.B. am hinteren und vorderen Ende des Kollektors, um den Einfluss auf den umliegenden Boden zu verdeutlichen (Abbildung 4).

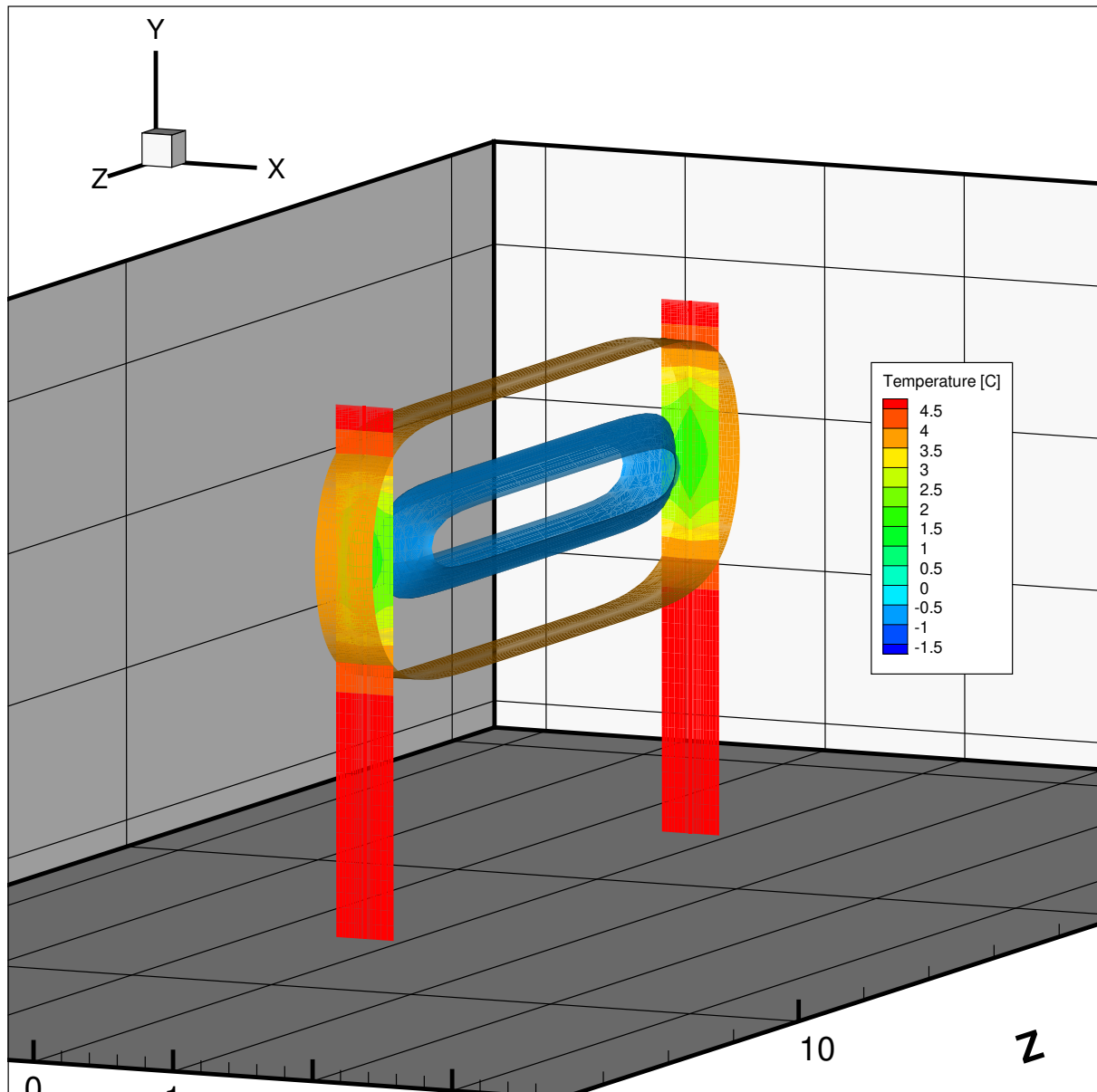


Abbildung 4: Isoprofile ($-0,5\text{ °C}$ und 4 °C) und vertikale Profilschnitte bei $z=4\text{m}$ und $z=9\text{m}$ (an den jeweiligen Enden des Kollektors). Der Einfluss des Kollektors auf das umliegende Erdreich ist am Ende der 10 Tage eher gering, da die Abkühlphase noch nicht abgeschlossen ist.

3.4 Analyse mit PostProc 2

Das PostProzessing ist in der Lage, Profilschnitte in allen 3 Ebenen aus einer 5D Ergebnisdatei ($x, y, z, \text{Zeit}, \text{Werte}$) herauszuschneiden. Um zum Beispiel einen Schnitt direkt längs durch den Kollektor zu erhalten, muss ein XT-Schnitt an Koordinate $x=0,205\text{m}$ definiert werden.

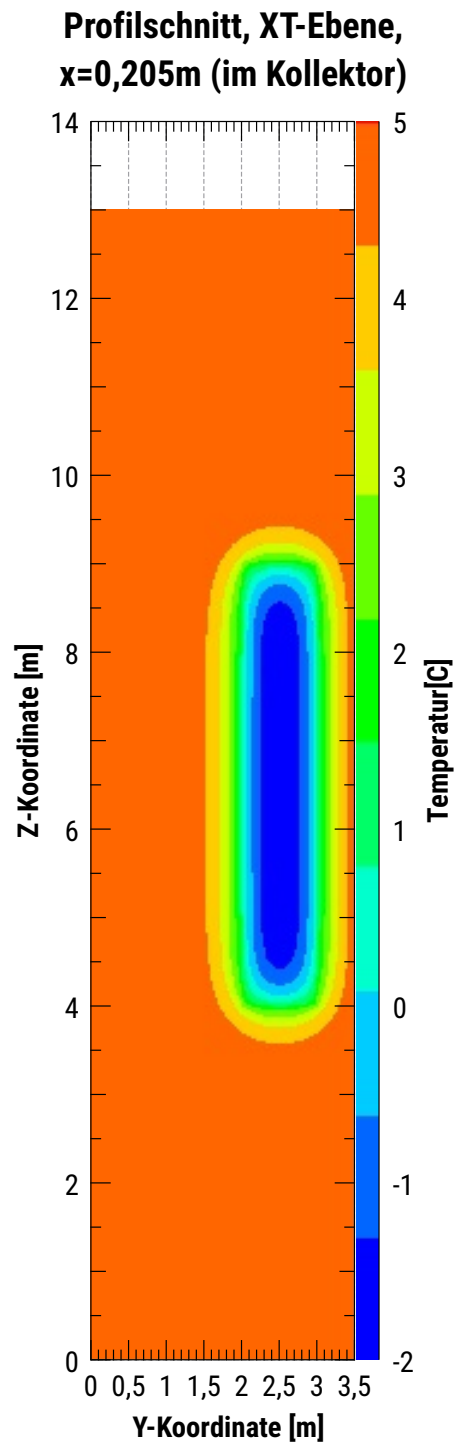


Abbildung 5: Profilschnitt durch den Kollektor in YZ-Ebene, an Stelle $x=0,205\text{ m}$ und zum Zeitpunkt $t=9\text{d}$.