BHYWI-22: Semester-Fahrplan: 2019

Vorlesungen & Übungen

Kolditz		*Hydromechanik und numerische Methoden*
05 Apr 2019	V2-01	Einführungin die Veranstaltung
12 Apr 2019	V2-02	Hydromechanik / Numerische Methoden: Wiederholung
19 Apr 2019		Ostern
		Numerische Methoden
26 Apr 2019	V2-03	Einzugsgebiet: Übung
03 Mai 2019	V2-04	Numerik: Finite-Differenzen-Verfahren (2D): Vorlesung
10 Mai 2019	V2-05	Numerik: Finite-Differenzen-Verfahren (2D): Übung
17 Mai 2019	V2-06	Software: Objekt-Orientierte FDM
24 Mai 2019	V2-07	Selke-Modell: Übung
31 Mai 2019	V2-08	Numerik: Finite-Elemente-Verfahren: Vorlesung
07 Jun 2019	V2-09	Numerik: Finite-Elemente-Verfahren: Übung
14 Jun 2019	V2-10	Pfingsten
		Datenbasierte Methoden
21 Jun 2019	V1+2	*VISLAB Exkursion*
28 Jun 2019	V2-11	Hydrologeologische Modellierung: Datenbasierte Verfahren I
05 Jul 2019	V2-12	Hydrologeologische Modellierung: Datenbasierte Verfahren II
12 Jul 2019	V2-13	Klausurvorbereitung

Modellierung von Hydrosystemen

"Numerische und daten-basierte Methoden"

BHYWI-22-V2-05 @ 2019

Finite-Differenzen-Methode

Selke-Modell

Olaf Kolditz

*Helmholtz Centre for Environmental Research – UFZ

[1]Technische Universität Dresden – TUDD

[2]Centre for Advanced Water Research – CAWR

07.06.2019 - Dresden

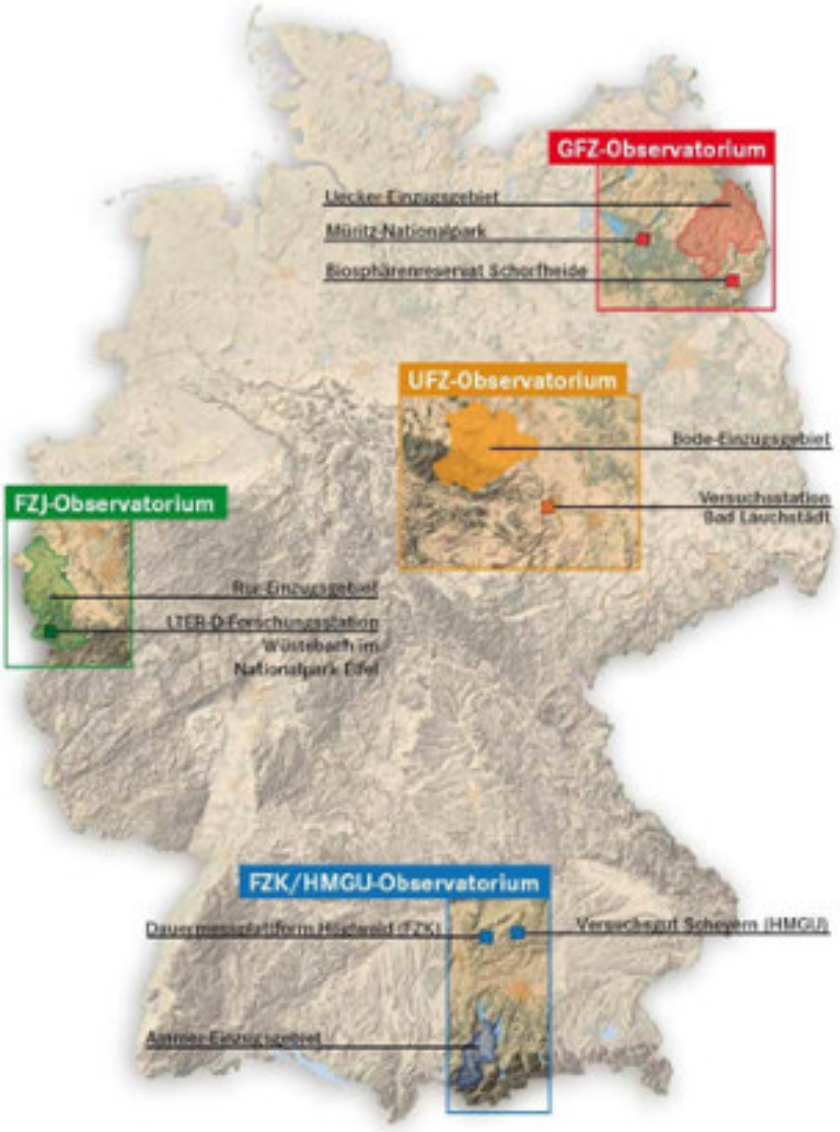
TECHNISCHE UNIVERSITÄT DRESDEN

Fahrplan für heute ...

- Kurzer Rückblick (Stabilitätskriterium)
- Vorstellen der Case Study: Selke Catchment (Bode)
- OGS DataExplorer
- Aktive und Inaktive Knoten
- FDM - erstmal QAD Programmierung (USA3)

TECHNISCHE UNIVERSITÄT DRESDEN

TERENO - Hydrologische Observatorien



- TERENO Projekt
- Steffen Zacharias
- Ute Wollschläger
- VISLab (Karsten Rink)

TECHNISCHE UNIVERSITÄT DRESDEN

Case Study: Bode Einzugsgebiet

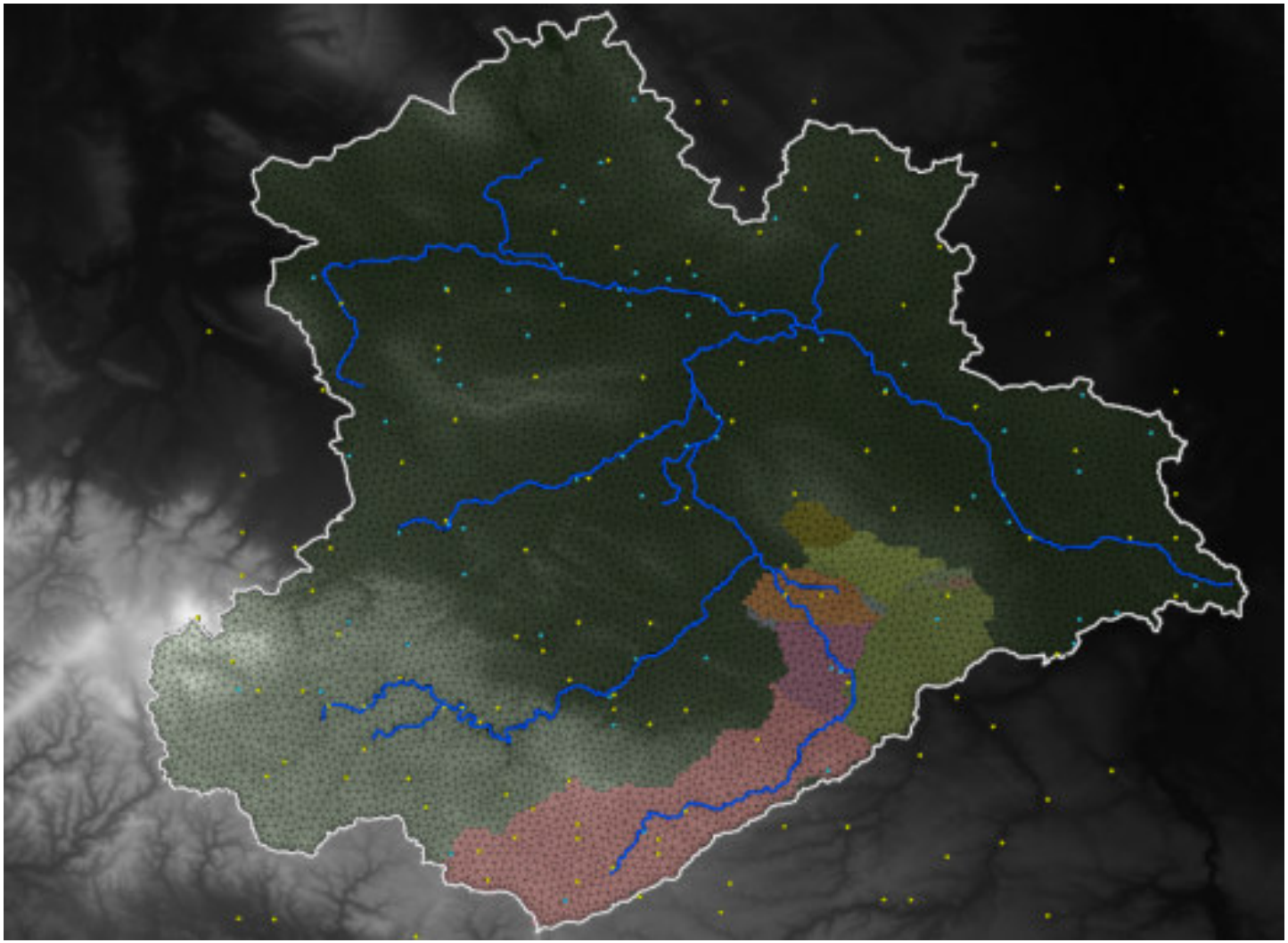


Figure: Digitales Geländemodell (DEM) des Bode-Einzugsgebietes

TECHNISCHE UNIVERSITÄT DRESDEN

Case Study: Bode Einzugsgebiet

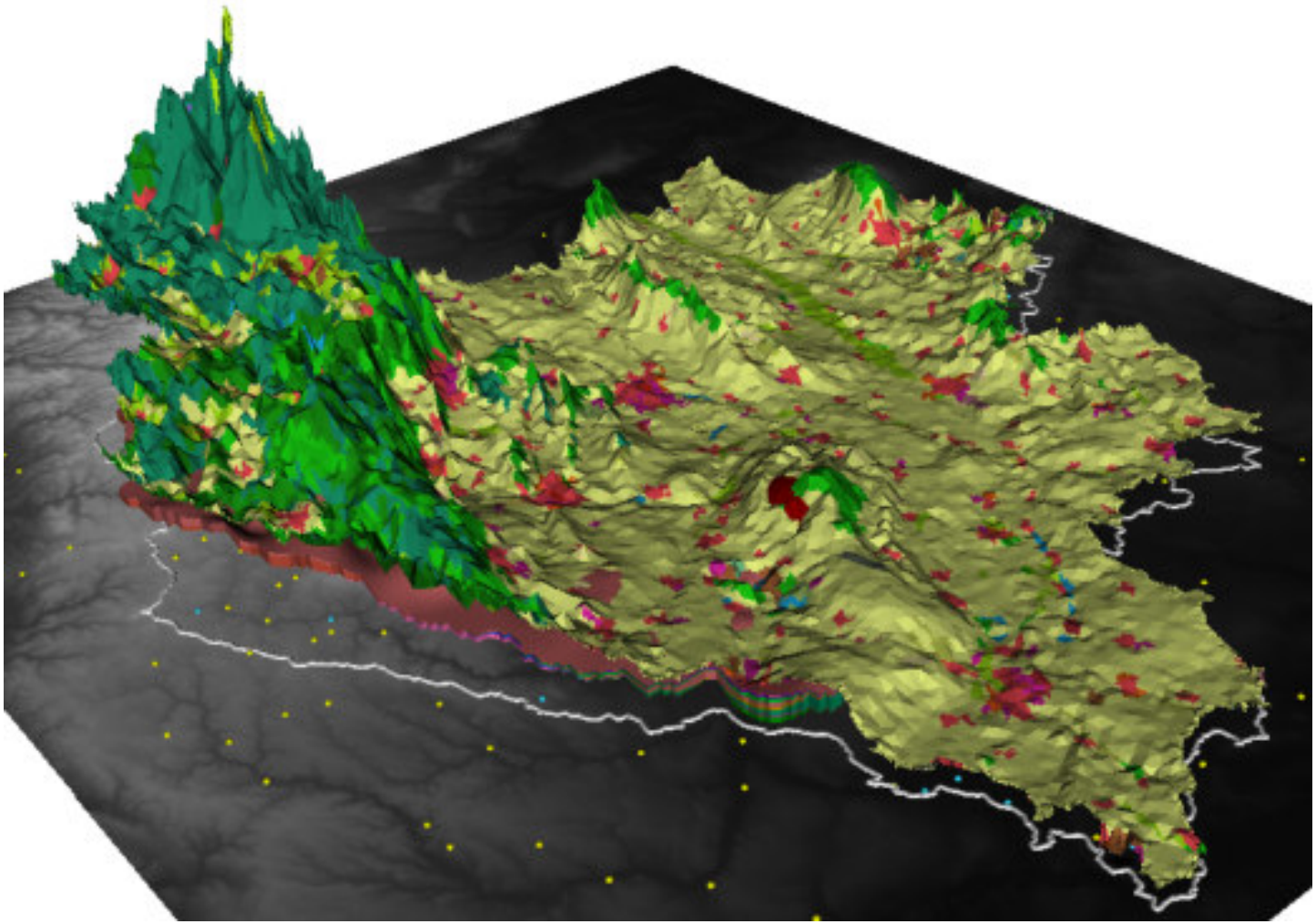


Figure: DEM mit Landnutzung

TECHNISCHE UNIVERSITÄT DRESDEN

Case Study: Bode Einzugsgebiet

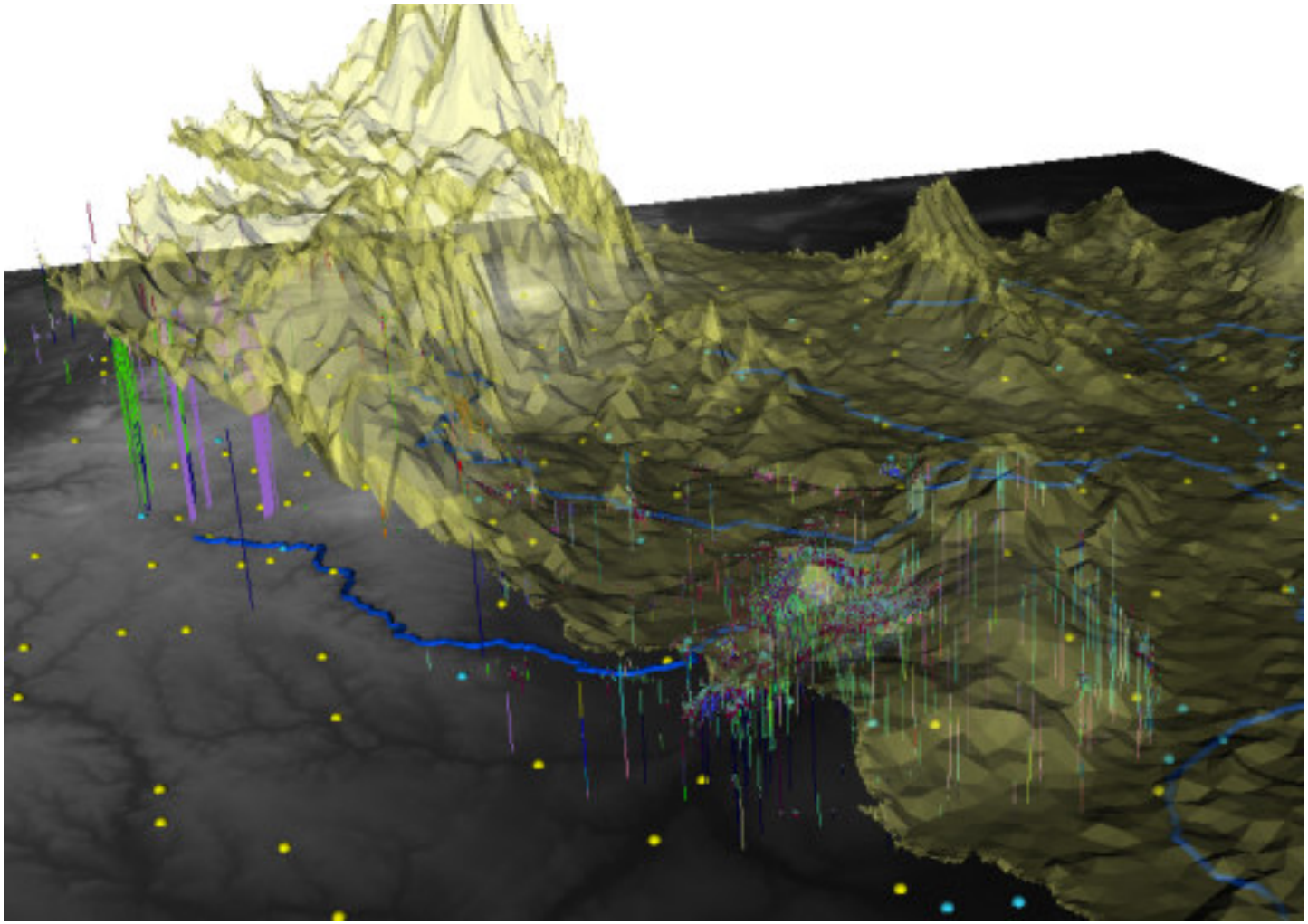


Figure: DEM mit geologischen Daten

TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet

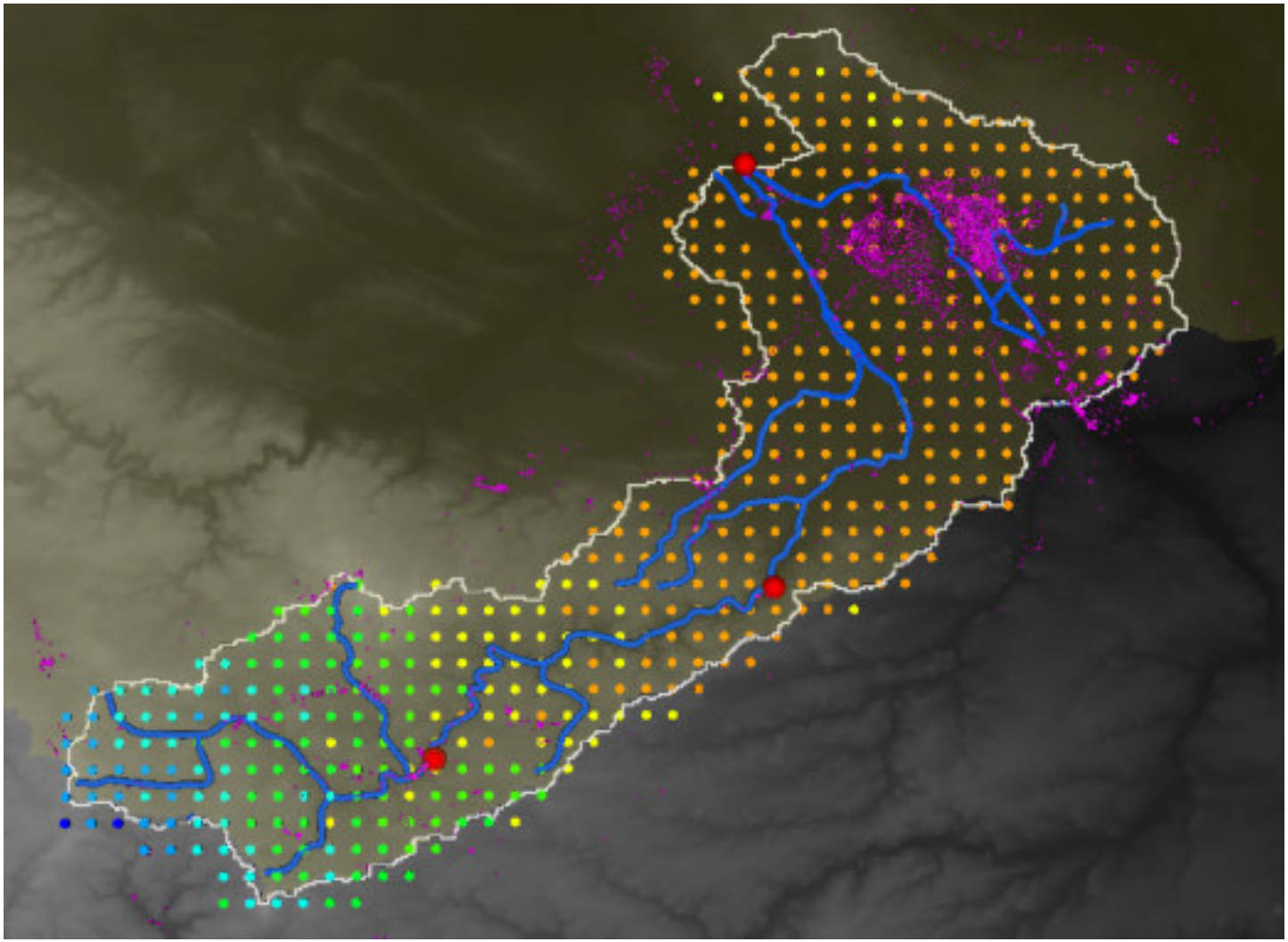


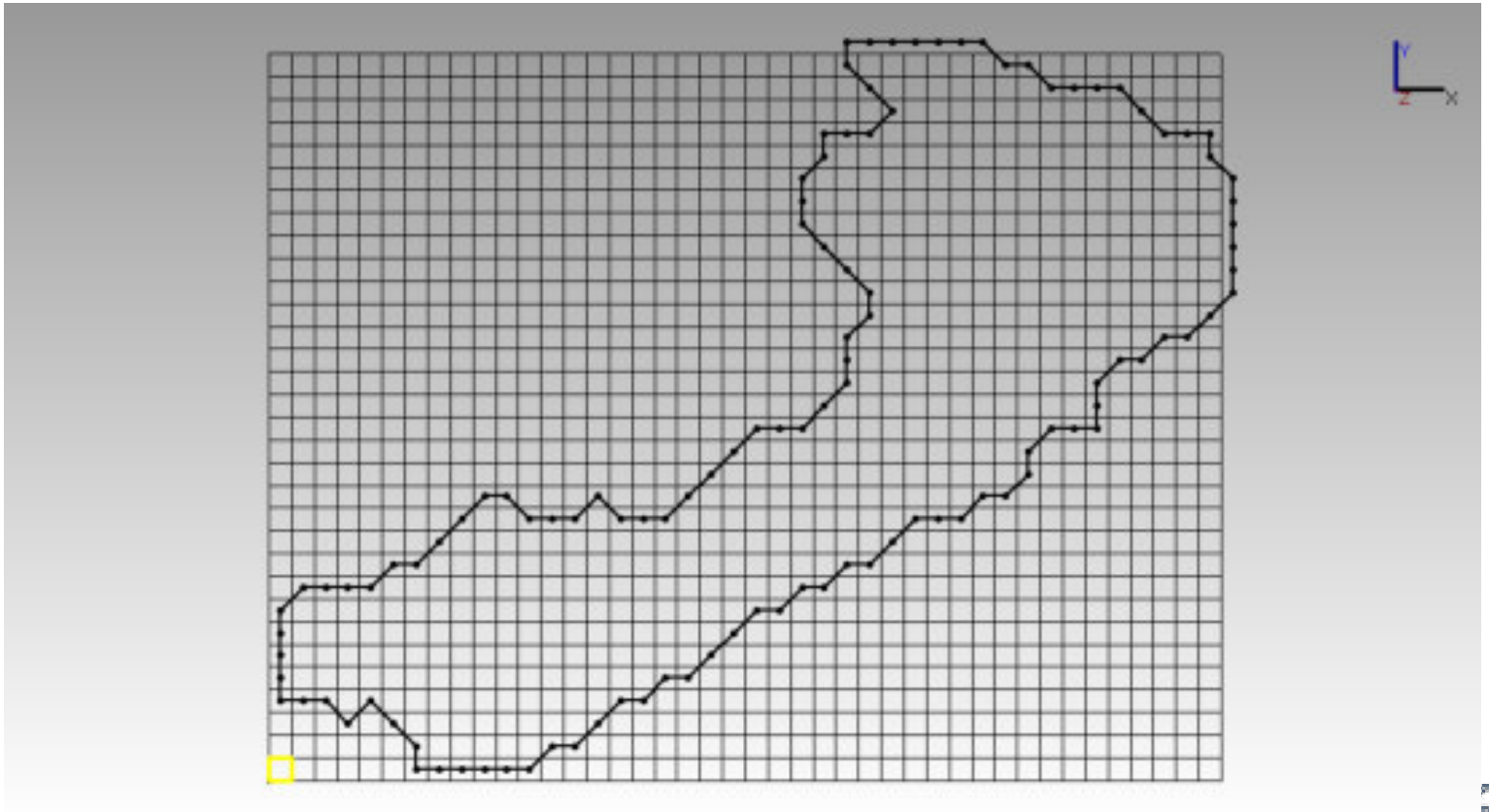
Figure: Untersuchungsgebiet - Selke

TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet

Die Abb. zeigt uns eine mögliche Approximation des Selke-Einzugsgebietes mit einer relativ geringen Anzahl von FD Zellen (32x42=1344).

- Aus wie vielen FD Knoten besteht das FD Mesh?



TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet

Wie bekommen wir aus unserem regelmäßigen rechteckigen Raster ein eher unregelmäßiges Catchment herausgeschnitten? Der Trick besteht darin, einzelne Zellen zu deaktivieren. Das klingt schon wieder nach Arbeit, ist aber machbar, dafür gibt's die nächste Übung (GW2). Die geometrische Analyse mit OGS liefert uns zunächst eine Liste von Gitterpunkten die ausserhalb des Catchments liegen (siehe Übung GW2):

- ExtractedSelkeMeshIDs.txt
- selke.gli

Diese Files können wir mal mit dem OGS-DatenExplorer (ogs-gui.exe) laden.

TECHNISCHE UNIVERSITÄT DRESDEN

OpenGeoSys - Datenexplorer (OGS-DE)

- Download von der Lehre-Seite
- Manual
- Übung

OGS-DE: Selke Einzugsgebiet

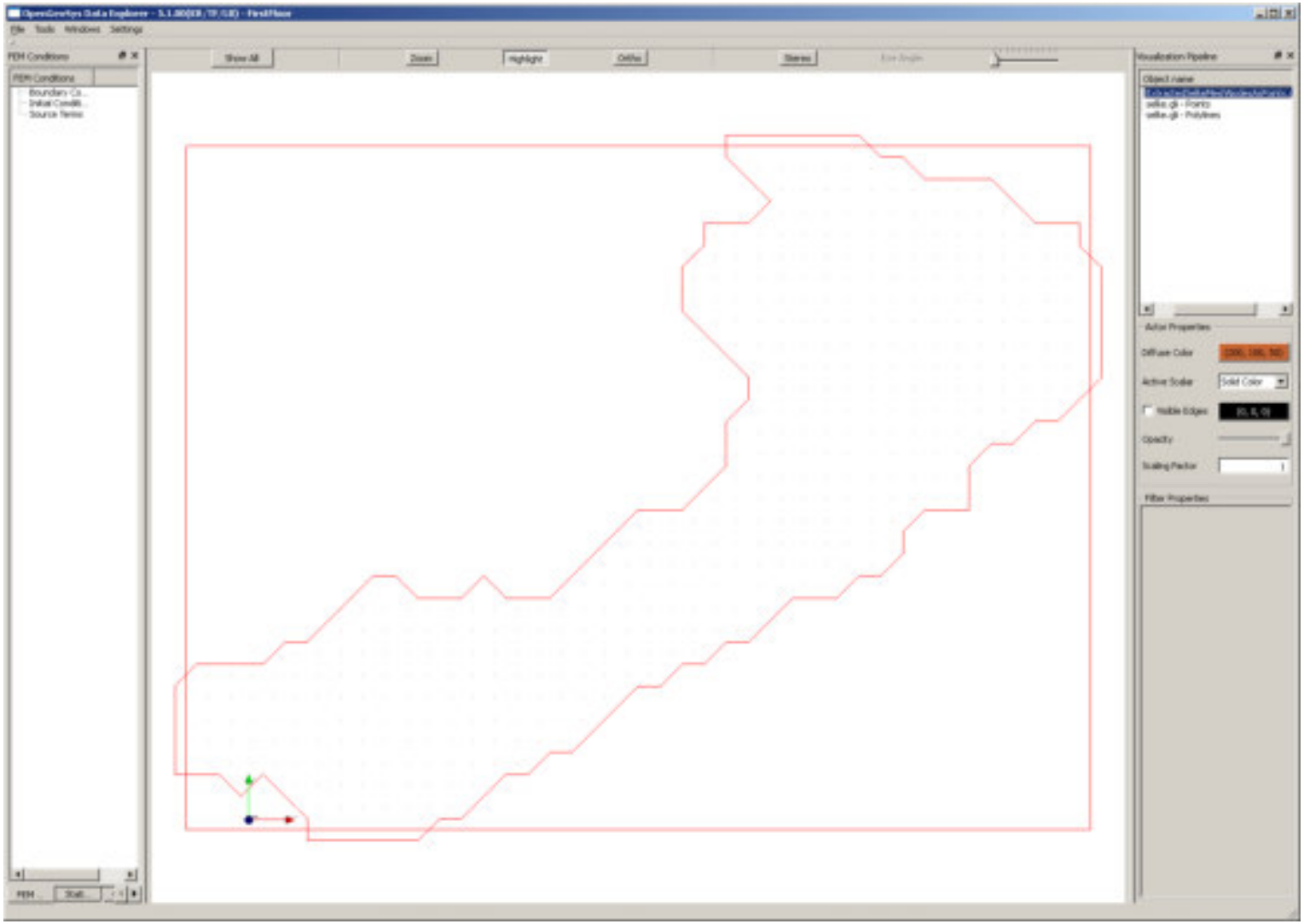


Figure: Das sind zwar die Daten, sieht aber noch nach nix aus ...

OGS-DE: Selke Einzugsgebiet

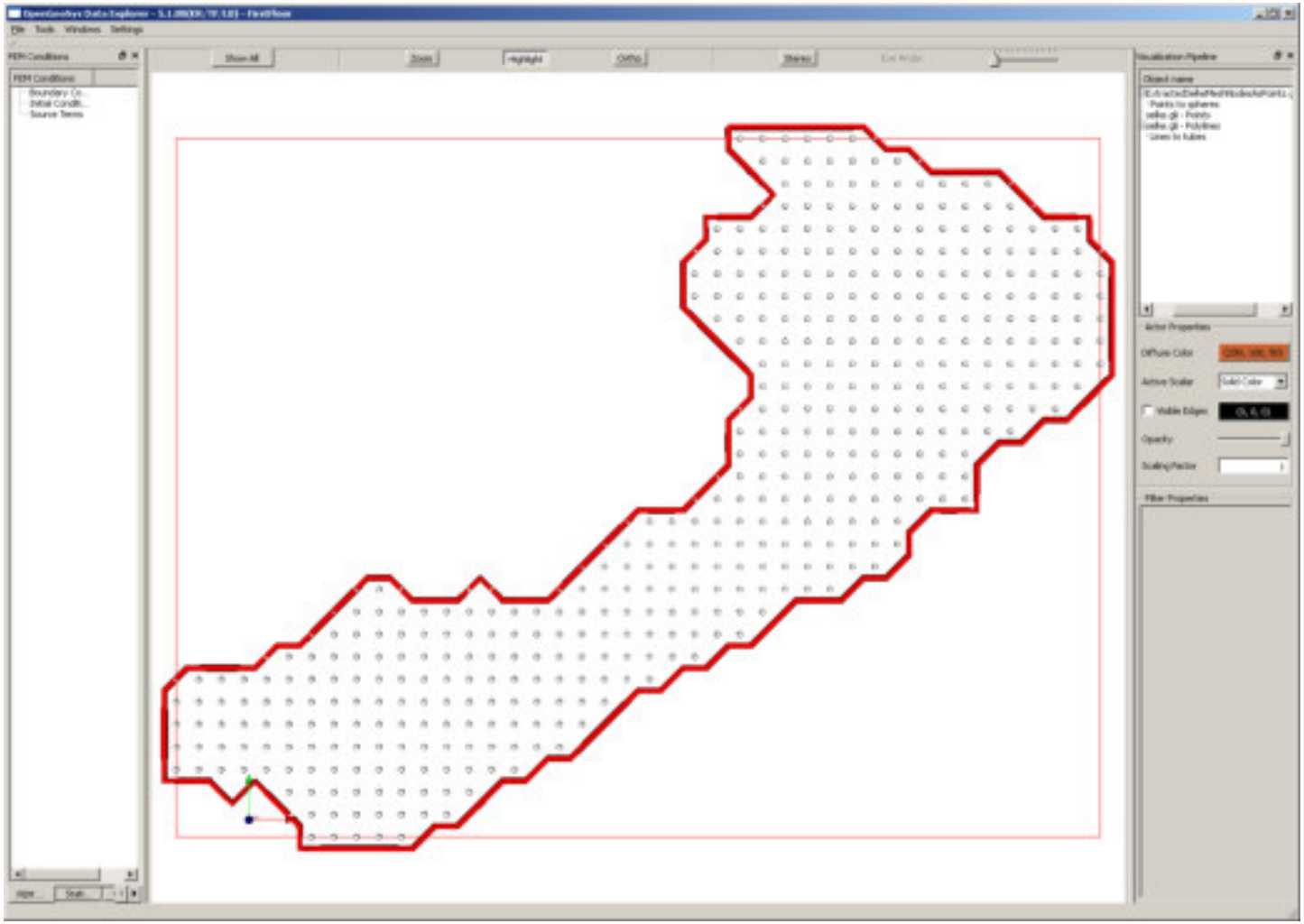
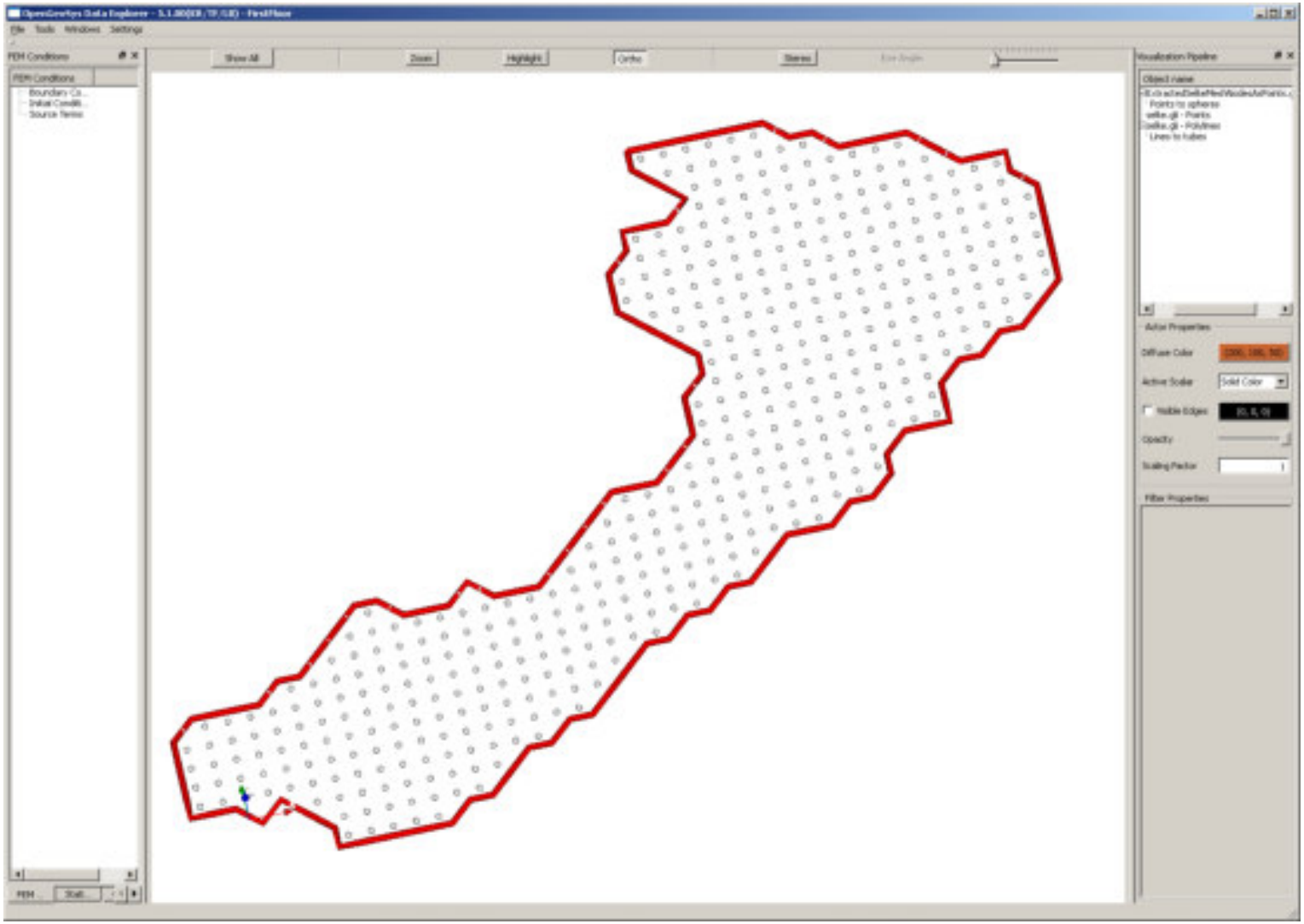


Figure: OGS hat ein paar nette VTK-Filter, um geometrische Objekte herauszuheben

TECHNISCHE UNIVERSITÄT DRESDEN

OGS-DE: Selke Einzugsgebiet



TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet Modellieren

Das ist zwar alles schön und gut, was wir aber brauchen sind die Knoten ausserhalb des Catchments, damit wir diese für das FD Verfahren deaktivieren können (Tafelbild). Also müssen wir doch selber ran. Unser Plan ist wie folgt:

1. Aktive Knoten lesen und speichern.
2. Aktive Knoten sortieren (Gruß an Hydroinformatik I - Hantieren mit Listen)
3. (das Zwischenergebnis zur Sicherheit mal rausschreiben)
4. Alle Knoten rausfischen, die NICHT aktiv sind.
5. Dabei kommt eine neue Hilfs-Funktion `NodeInList` ins Spiel (siehe unten).
6. Wir überzeugen uns vom Ergebnis (File schreiben) ...
7. ... und natürlich graphisch, wozu haben wir denn Visual C++ gelernt!

TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet Modellieren

- ActiveNodes.txt

```
173
216
259
302
345
174
217
260
303
346
175
218
261
304
347
...
```

TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet Modellieren

- Aktive Knoten lesen und speichern.
- Aktive Knoten sortieren (Gruß an Hydroinformatik I - Hantieren mit Listen)

```
std::list<int>nodes_active;
std::ifstream active_nodes_file;
active_nodes_file.open("ActiveNodes.txt");
int na;
while(!active_nodes_file.eof())
{
  active_nodes_file >> na;
  nodes_active.push_back(na);
}
nodes_active.sort();
```

TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet Modellieren

- Das Zwischenergebnis zur Sicherheit mal rausschreiben

```
std::ofstream active_nodes_file_test;
active_nodes_file_test.open("ActiveNodesSorted.txt");
list<int>::const_iterator p = nodes_active.begin();
while(p!=nodes_active.end())
{
  active_nodes_file_test << *p << endl;
  ++p;
}
active_nodes_file_test.close();
```

TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet Modellieren

- ActiveNodesSorted.txt

```
50
51
52
53
54
93
94
95
96
97
98
...
```

- E: Überzeugen sie sich, ob unser Unterfangen erfolgreich war, indem sie die Elementanzahl der beiden Listen bestimmen und rausschreiben.

TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet Modellieren

- Alle Knoten rausfischen, die NICHT aktiv sind.
- Dabei kommt eine neue Hilfs-Funktion `NodeInList` ins Spiel (siehe unten).

```
for(j=0;j<jy;j++)
{
  nn = j*ix;
  for( i=0;i<ix;i++)
  {
    n = nn+i;
    if(!NodeInList(n,nodes_active))
      nodes_inactive.push_back(n);
  }
}
```

TECHNISCHE UNIVERSITÄT DRESDEN

Selke Einzugsgebiet Modellieren

Die nützliche Hilfs-Funktion, die alle Knoten raussucht, die NICHT in `nodes_active`stehen.

```
bool NodeInList(int n,std::list<int>nodes_active)
{
  list<int>::const_iterator p = nodes_active.begin();
  while(p!=nodes_active.end())
  {
   if(n==*p)
     return true;
    ++p;
  }
  return false;
}
```

Selke Einzugsgebiet Modellieren

- Wir überzeugen uns vom Ergebnis (File schreiben) ...

```
std::ofstream inactive_nodes_file;
inactive_nodes_file.open("InactiveNodes.txt");
for(i=0;i<nodes_inactive.size();i++)
{
  inactive_nodes_file << nodes_inactive[i] << endl;
}
inactive_nodes_file.close();
```

TECHNISCHE UNIVERSITÄT DRESDEN

OGS-DE: Selke Einzugsgebiet

- ... und natürlich graphisch, wozu haben wir denn Visual C++ gelernt!