

Modellierung Hydrosysteme: Finite-Differenzen-Methode (FDM)

Prof. Dr.-Ing. habil. Olaf Kolditz

¹Helmholtz Centre for Environmental Research – UFZ, Leipzig

²Technische Universität Dresden – TUD, Dresden

Dresden, 16. Juni 2017

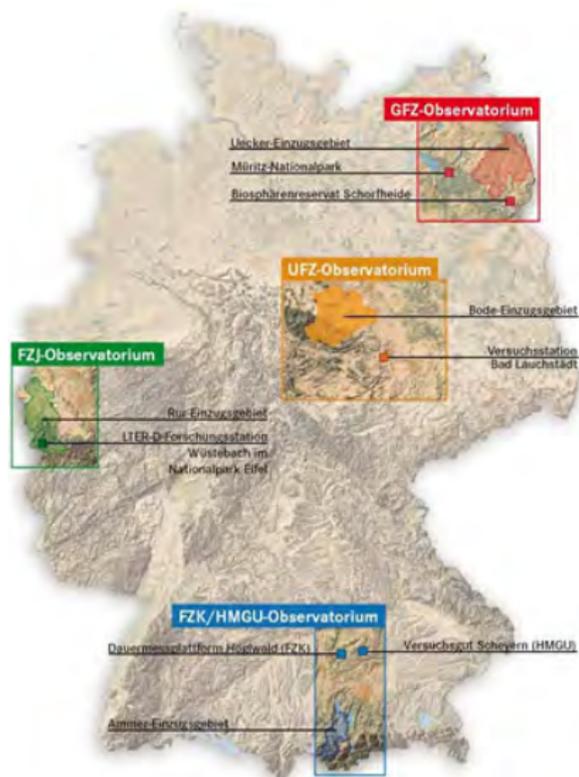
Fahrplan für das Semester ...

April	Mai	Juni	Juli			
	5.5.2017	6.2.2017	7.3.2014			Kolditz
				SCH/A 251/H	07.04.2017	1 Einführung / Hydrologische Modellierung
				CHE/089/E	21.04.2017	3 Hydrologische Modellierung II
				CHE/089/E	28.04.2017	5 DR DECOVALEX
				CHE/089/E	05.05.2017	7 Wdh. Hydromechanik / Übung Einzugsgebiet
					12.05.2017	9 DR China
					19.05.2017	11 Exkursion VisLab
4.7.2017	5.12.2017	6.9.2017	7.14.2017		26.05.2017	13 Finite-Differenzen-Verfahren 2D
	Qt Übung HW HW China	Pflingten		SCH/A 251/H	02.06.2017	15 Objekt-Orientierte FDM / Randbedingungen
				CHE/089/E	16.06.2017	17 FDM Anwendung Selke-Einzugsgebiet
				CHE/089/E	23.06.2017	19 Implizite FDM
					30.06.2017	21 Finite-Elemente-Methode I
4.14.2017	5.19.2017	6.16.2017			07.07.2017	23 Finite-Elemente-Methode II
Datern	Qt Übung VISLAB VISLAB China			SCH/A 251/H	14.07.2017	25 Zusammenfassung, Klausurvorbereitung
				CHE/089/E		Walther
				CHE/089/E	07.04.2017	2 V1 Einführung / Überblick
	Brücke ntag				21.04.2017	4 V2 Ursprung & Verbreitungswege Ü2 Fließwege
4.21.2017	5.26.2017	6.23.2017			28.04.2017	6 Ü2b Fließwege (als HA)
				SCH/A 251/H	05.05.2017	8 V3 Visualisierung Ü3 Visualisierung
				CHE/089/E	12.05.2017	10 V4 Detektion I
				CHE/089/E	19.05.2017	12 Exkursion VisLab? oder Ü3b Visualisierung (als HA)
				CHE/089/E	26.05.2017	14 V5 Detektion II Ü4 NDVI
		Wasserseminar			02.06.2017	16 V6 Exkurs Salzwasserintrusion
4.28.2017		6.30.2017	Termin		16.06.2017	18 V7 GW-Modellierung Ü5 GW analytisch
Qt Intro Ü1 Ü2 DECOVALEX				SCH/A 251/H	23.06.2017	20 V8 Benchmarking Ü6 OGS Einführung
				CHE/089/E	30.06.2017	22 V9 Meshing Ü6b Darcy num. (HA Meshing)
				CHE/089/E	07.07.2017	24 V10 Konservativer Stofftransport Ü7 Kons. ST
				CHE/089/E	14.07.2017	26 Zusammenfassung, Klausurvorbereitung

Fahrplan für heute ...

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

TERENO - Hydrologische Observatorien



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

Case Study: Bode Einzugsgebiet



Abbildung: Digitales Geländemodell (DEM) des Bode-Einzugsgebietes

Case Study: Bode Einzugsgebiet

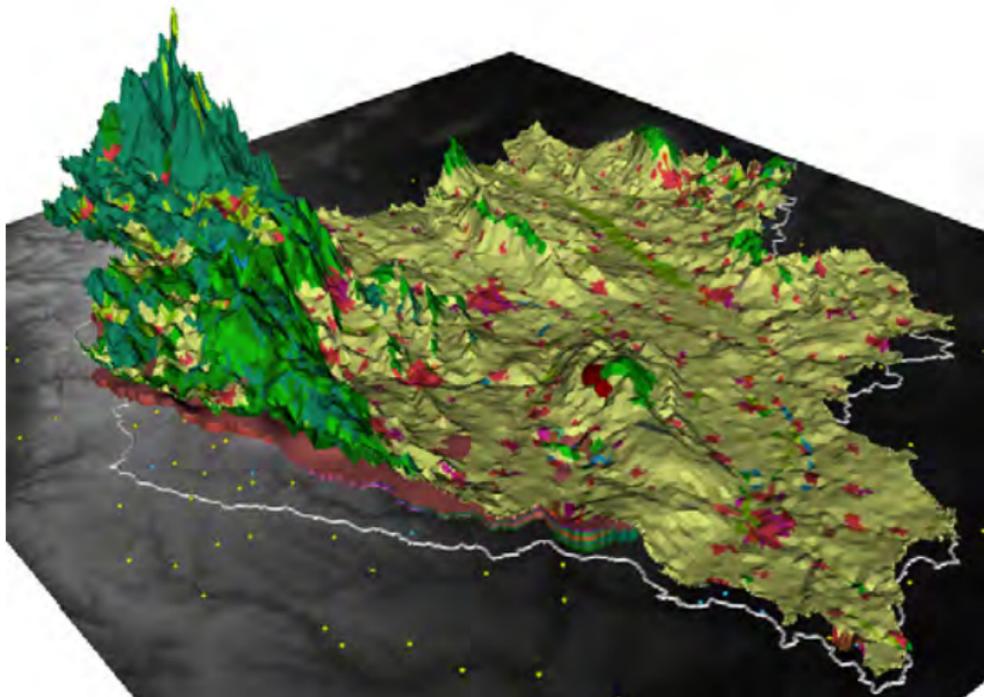


Abbildung: DEM mit Landnutzung

Case Study: Bode Einzugsgebiet

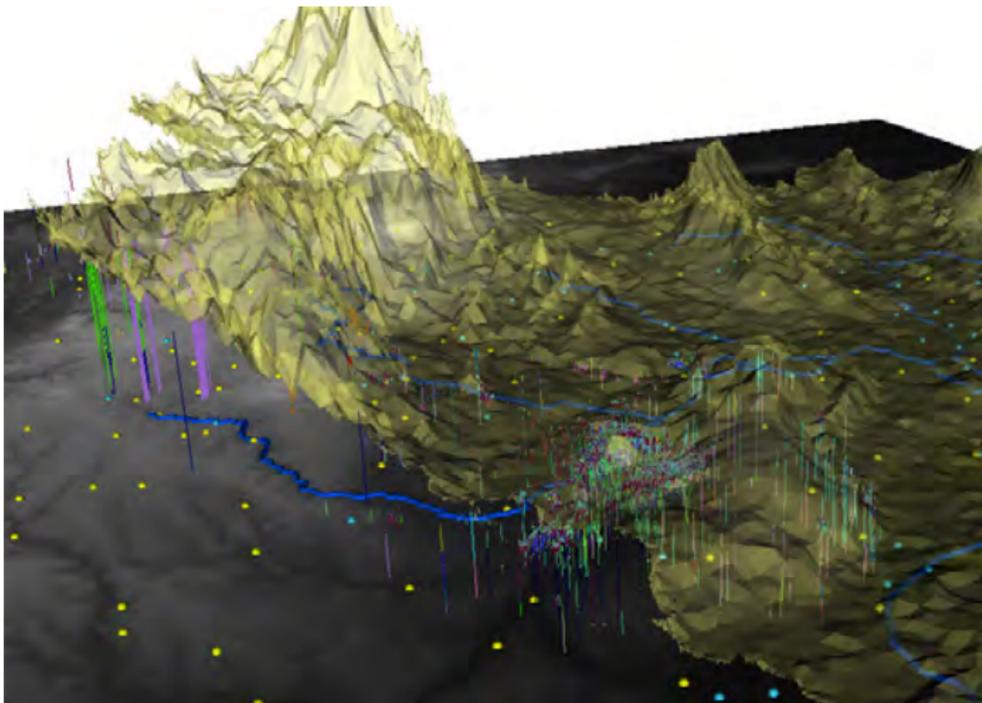


Abbildung: DEM mit geologischen Daten

Selke Einzugsgebiet

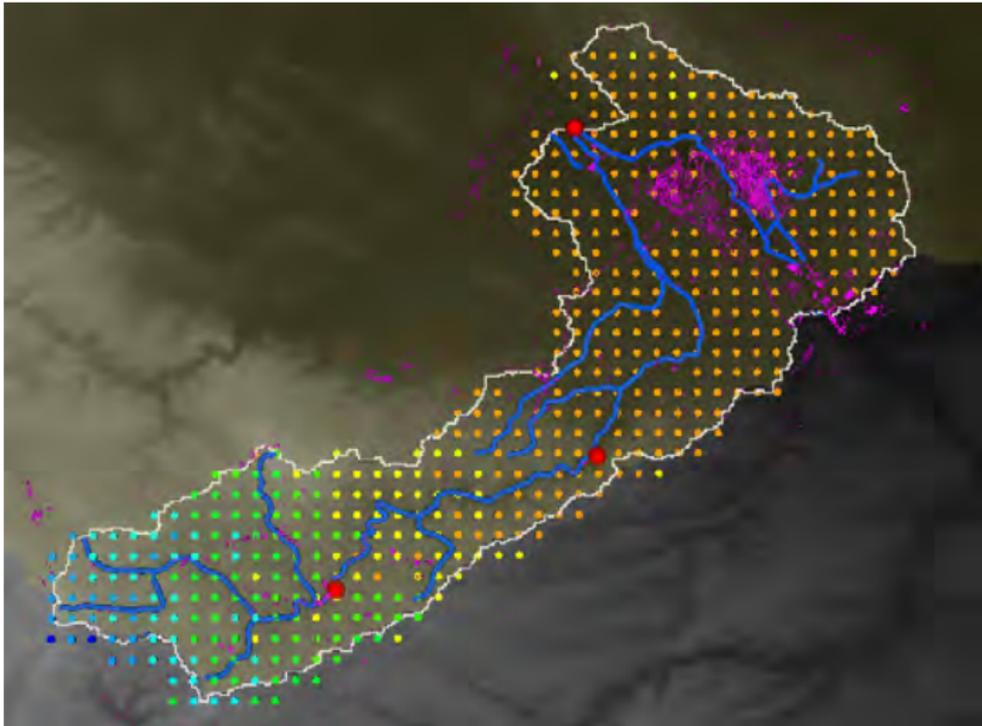
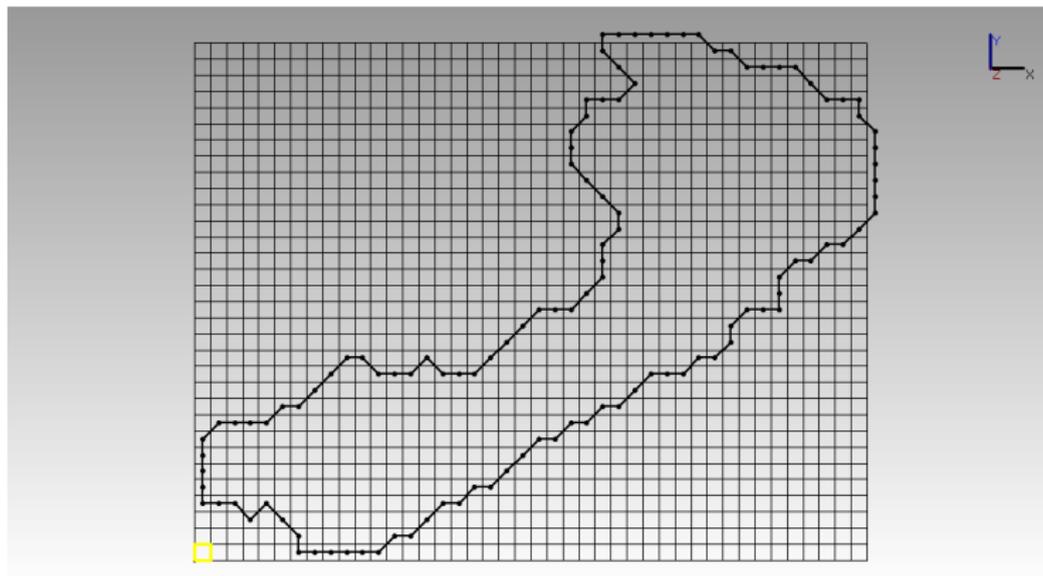


Abbildung: Untersuchungsgebiet - Selke

Selke Einzugsgebiet

Die Abb. zeigt uns eine mögliche Approximation des Selke-Einzugsgebietes mit einer relativ geringen Anzahl von FD Zellen ($32 \times 42 = 1344$).

- ▶ Aus wie vielen FD Knoten besteht das FD Mesh?



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.

OpenGeoSys - Datenexplorer (OGS-DE)

- ▶ Download von der Lehre-Seite
- ▶ Manual
- ▶ Übung

OGS-DE: Selke Einzugsgebiet

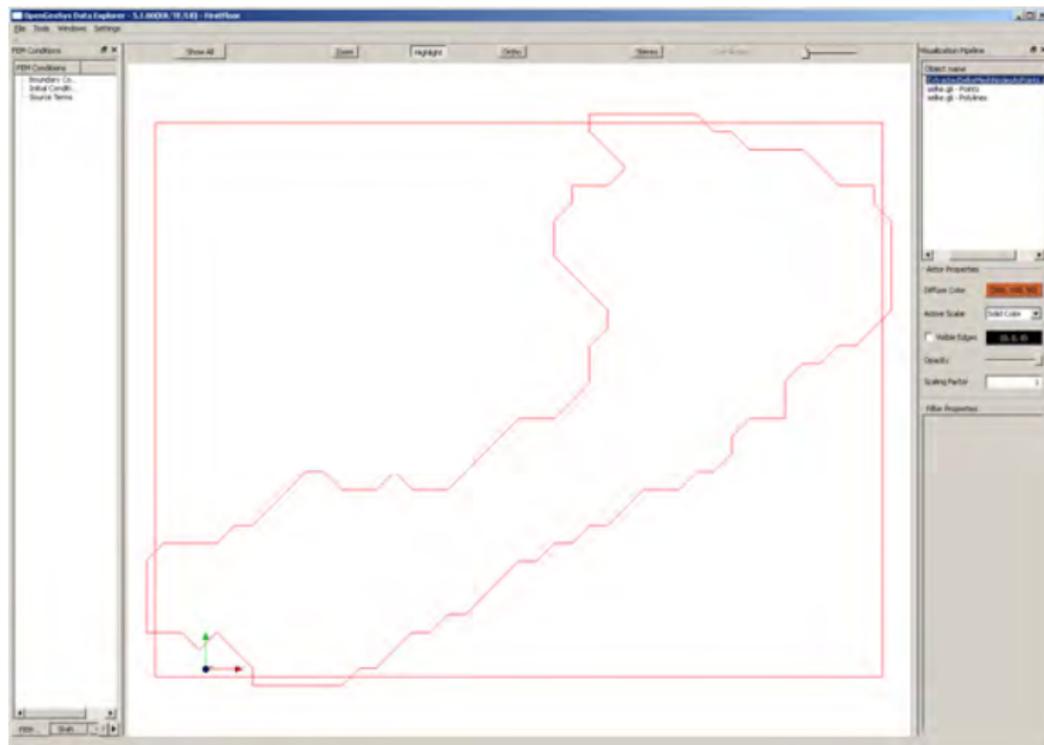


Abbildung: Das sind zwar die Daten, sieht aber noch nach nix aus ...

OGS-DE: Selke Einzugsgebiet

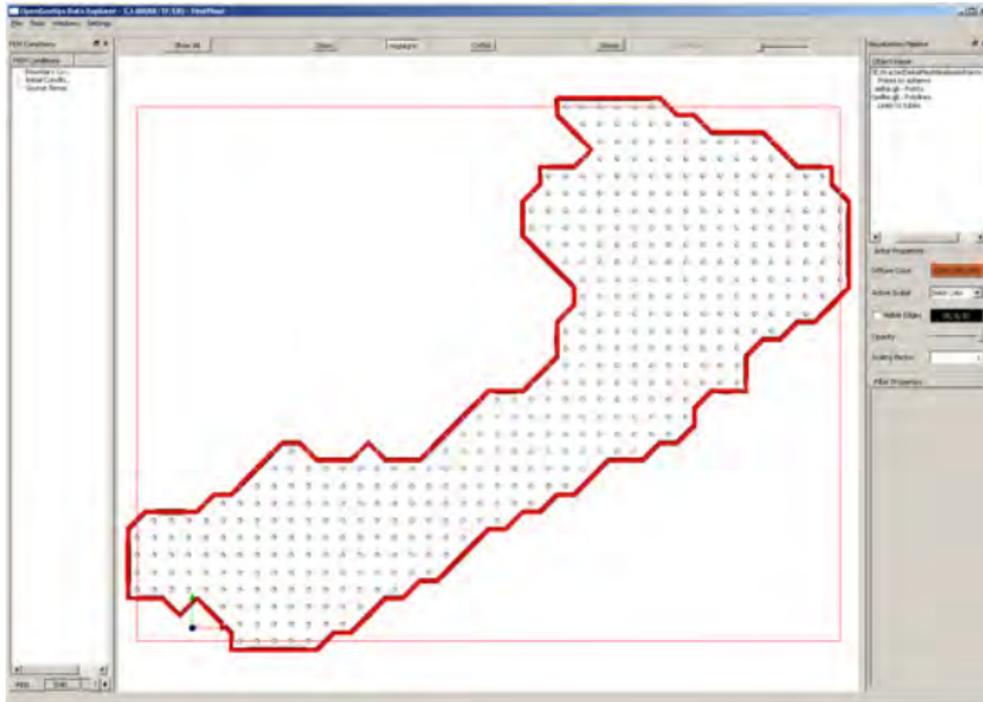
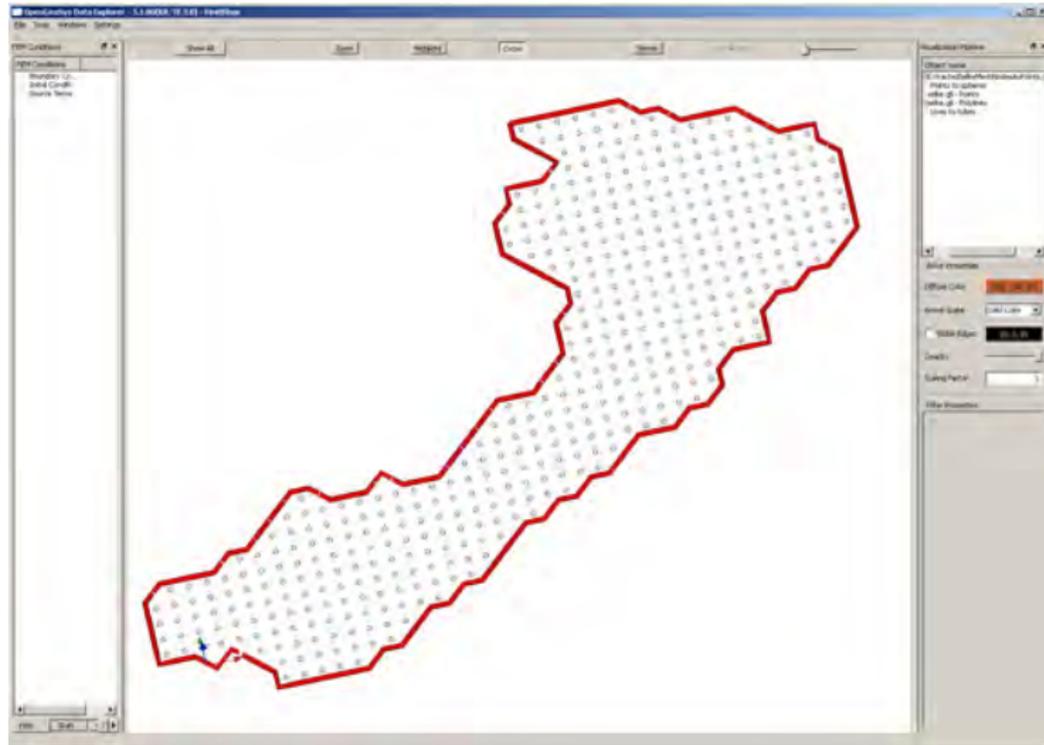


Abbildung: OGS hat ein paar nette VTK-Filter, um geometrische Objekte herauszuheben

OGS-DE: Selke Einzugsgebiet



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!

Selke Einzugsgebiet Modellieren

▶ ActiveNodes.txt

173

216

259

302

345

174

217

260

303

346

175

218

261

304

347

...

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();
```

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();
```

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.

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);
    }
}
```

Selke Einzugsgebiet Modellieren

Die nützliche Hilfs-Funktion, die alle Knoten raussucht, die NICHT in nodes_activestehen.

```
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();
```

OGS-DE: Selke Einzugsgebiet

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

