Modellierung Hydrosysteme: Finite-Differenzen-Methode (FDM)

Prof. Dr.-Ing. habil. Olaf Kolditz

[1]Helmholtz Centre for Environmental Research – UFZ, Leipzig

[2]Technische Universität Dresden – TUD, Dresden

Dresden, 15. Mai 2015

Fahrplan für das Semester

Lehre Sommersemester 2015

Grundlagen der Hydroinformatik II (BHYWI 08)

Modellierung von Hydrosystemen (BHYWI 22)

		April	Mai	Juni	Juli	August
						Klausur
			01.05.2015	05.06.2015	03.07.2014	
2. DS	09:20-10:50		Feiertag	Kolditz	Kolditz	SCH/A251/H
4. DS	13:00-14:30			Kolditz	Kolditz	CHE/089/E
5. DS	14:50-16:20			Kolditz	Walther	CHE/089/E
6. DS	16:40-18:10			Sachse	Walther	CHE/089/E
			08.05.2015	12.06.2015	10.07.2014	
2. DS	09:20-10:50		Kolditz	Kolditz	Kolditz	SCH/A251/H
4. DS	13:00-14:30		Sachse	Kolditz	Kolditz	CHE/089/E
5. DS	14:50-16:20		Sachse	Sachse	Kalbacher	CHE/089/E
6. DS	16:40-18:10			Sachse		CHE/089/E
		17.04.2015	15.05.2015	19.06.2015	17.07.2014	
2. DS	09:20-10:50	Kolditz	Kolditz	Kolditz	Kolditz	SCH/A251/H
4. DS	13:00-14:30	Kolditz	Kolditz	Kolditz	VISLAB	CHE/089/E
5. DS	14:50-16:20	Kolditz	Sachse	Walther	Bilke, Rink	CHE/089/E
6. DS	16:40-18:10			Walther	Fischer	CHE/089/E
		24.04.2015	22.05.2015	26.06.2015	24.07.2015	
2. DS	09:20-10:50	Kolditz	Kolditz	Kolditz	Kolditz	SCH/A251/H
4. DS	13:00-14:30	Kolditz	Kolditz	Kolditz	VISLAB	CHE/089/E
5. DS	14:50-16:20	Sachse	Sachse	Walther	Bilke, Shao	CHE/089/E
6. DS	16:40-18:10			Walther	Böttcher	CHE/089/E
			29.05.2015	Feiertage	Klausur	
			Pfingsten	01.05.2015	Termin	
				14.05.2015		
			(DR China)	23-31.05.15		

Fahrplan für heute ...

- Grundlagen - GWE
- Grundlagen - TSE
- Übungen zur expliziten FDM (page 12)

Grundlagen - GWE

- PDE

$$S\frac{\partial h}{\partial t} - \frac{\partial}{\partial x}\left(K_x\frac{\partial h}{\partial x}\right) - \frac{\partial}{\partial y}\left(K_y\frac{\partial h}{\partial y}\right) = Q \qquad (1)$$

Grundlagen - TSE

in time

$$u_j^{n+1} = \sum_{m=0}^{\infty} \frac{\Delta t^m}{m!} \left[\frac{\partial^m u}{\partial t^m}\right]_j^n \tag{2}$$

in space

$$u_{j+1}^{n} = \sum_{m=0}^{\infty} \frac{\Delta x^m}{m!} \left[\frac{\partial^m u}{\partial x^m}\right]_j^n \tag{3}$$

Grundlagen - FDM

- Zeitableitung

$$\left[\frac{\partial u}{\partial t}\right]_j^n = \frac{u_j^{n+1} - u_j^n}{\Delta t} - \frac{\Delta t}{2}\left[\frac{\partial^2 u}{\partial t^2}\right]_j^n + 0(\Delta t^2) \tag{4}$$

- in space

$$\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} + \frac{\Delta x^2}{12}\left[\frac{\partial^4 u}{\partial x^4}\right]_j^n + \ldots \tag{5}$$

FDM - explizites Schema

$$S_{i,j}\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t} - K_{i,j}^{x}\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^2} - K_{i,j}^{y}\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\Delta y^2} = Q_{i,j} \tag{6}$$

$$\begin{aligned} u_{i,j}^{n+1} &= u_{i,j}^{n} \\ &+ \frac{K_{i,j}^{x}}{S_{i,j}}\frac{\Delta t}{\Delta x^2}u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n} \\ &+ \frac{K_{i,j}^{y}}{S_{i,j}}\frac{\Delta t}{\Delta y^2}u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n} \\ &+ \frac{Q_{i,j}}{S_{i,j}} \end{aligned} \tag{7}$$

Grundlagen - FDM

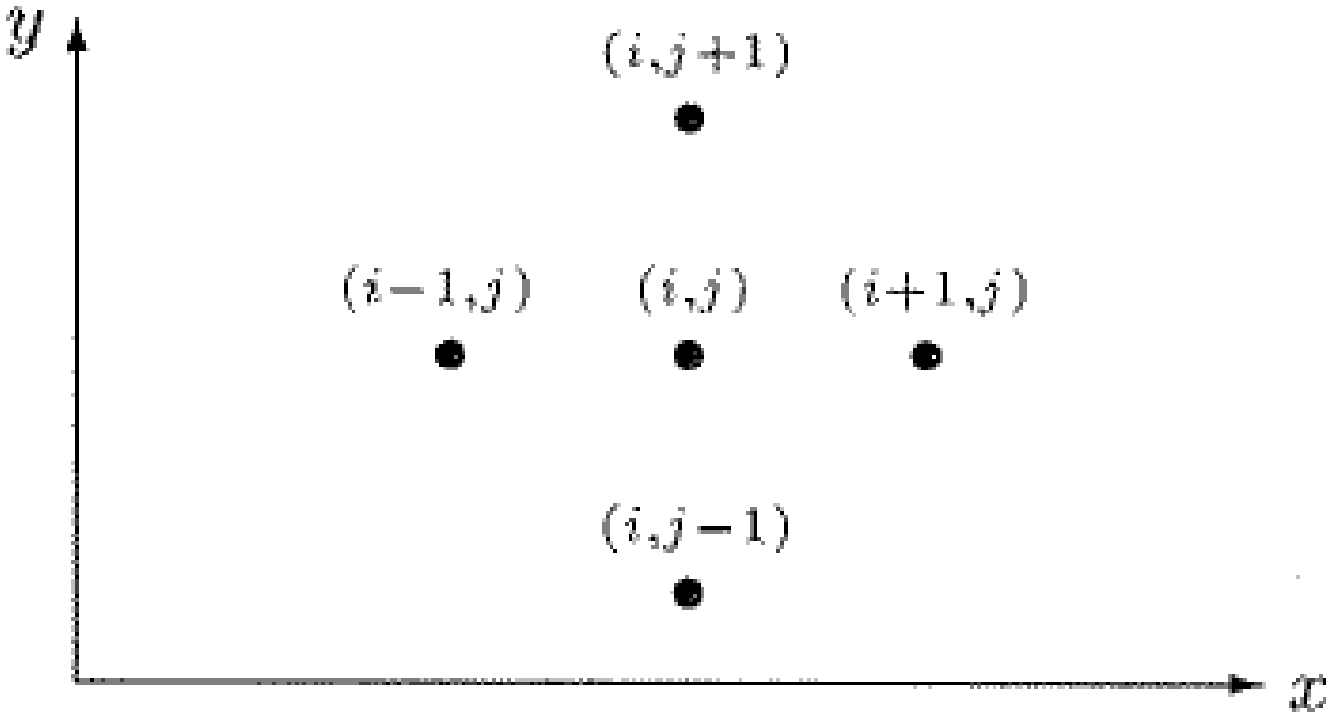


Abbildung: 5-Punkte-Stern (Knabner und Angermann 2000)

Übung - USA2 - QAD

- Datenstrukturen

Tabelle:

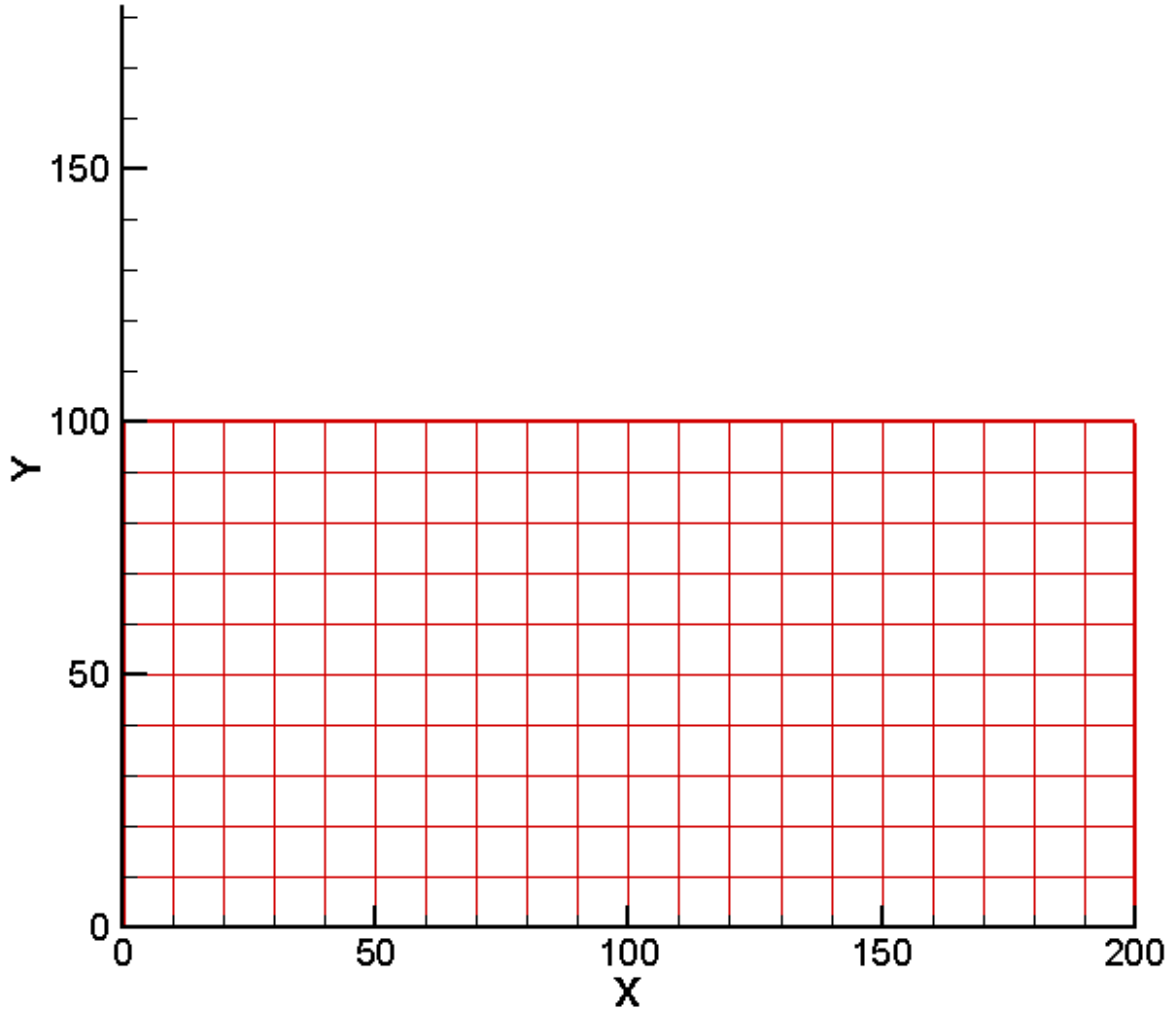
Feldgröße	u
Physikalische Parameter	S, K, Q
Numerische Parameter	$\Delta t, \Delta x, \Delta y$

Übung - USA2 - QAD

Die Minimal-Datenstrukturen für die Programmierung der Gleichung (7) sind damit:

```
std::vector<float>u_new;
std::vector<float>u_old;
float S0,Kf,Q;
float dx,dy,dt;
```

Übung - USA2 - QAD



Übung - USA2 - QAD - V1

Um dieses Gitter "abtasten" zu können, schreiben wir folgende doppelte Schleife.

```
for(j=0;j<jy;j++)
{
  nn = j*ix;
  for( i=0;i<ix;i++)
  {
    n = nn+i;
    u_new[n] = u[n] \
             + Kf/S0*dt/dx2 * (u[n+1]-2*u[n]+u[n-1]) \
             + Kf/S0*dt/dy2 * (u[(j+1)*ix+i]-2*u[n]+u[(j-1)*ix+i]) \
             + Q/S0;
  }
}
```

Übung - USA2 - QAD - V2

Um dieses Gitter "abtasten" zu können, schreiben wir folgende doppelte Schleife.

```
for(int j=0;j<jy;j++)
{
  nn = j*ix;
  for(int i=0;i<ix;i++)
  {
    n = nn+i;
    if(IsBCNode(n,bc_nodes))
      continue;
    u_new[n] = u[n] \
             + Kf/S0*dt/dx2 * (u[n+1]-2*u[n]+u[n-1]) \
             + Kf/S0*dt/dy2 * (u[(j+1)*ix+i]-2*u[n]+u[(j-1)*ix+i]) \
             + Q/S0;
  }
}
```

Übung - USA2 - QAD

Dabei ist j der Laufindex über die y Richtung und i der Laufindex über die x Richtung. Ganz wichtig ist natürlich, den Speicher für die Vektoren bereitzustellen, bevor es los geht.

```
u.resize(ix*jy);
u_new.resize(ix*jy);
```

- Welche Rolle spielen `ix` und `jy` bei der Speicherreservierung?

Übung - USA2 - QAD

Natürlich müssen auch die Parameter vor der Berechnung initialisiert werden

```
ix = 21;
jy = 11;
dx = 10.;
dy = 10.;
dt = 0.25e2;
S0 = 1e-5;
Kf = 1e-5;
Q  = 0.;
u0 = 0.;
```

- Welche Einheiten haben die einzelnen Parameter?

Übung - USA2 - QAD

Das mit den Anfangsbedingungen ist eine einfache Sache. Mit der Doppelschleife über alle Knoten, können wir sehr einfach einen Wert u0 als Anfangsbedingung überall zuweisen.

```
for(int i=0;i<ix;i++)
  for(int j=0;j<jy;j++)
  {
    u[j*(ix+1)] = u0;
    u_new[j*(ix+1)] = u0;
  }
}
```

Übung - USA2 - QAD

Mit den Randbedingungen ist es etwas kniffliger ...

```
//top and bottom
int l;
for(int i=0;i<ix;i++)
{
  bc_nodes.push_back(i); u[i] = u_top  u_new[i] = u_top;
  l = ix*(jy-1)+i;
  bc_nodes.push_back(l); u[l] = u_bottom;  u_new[l] = u_bottom;
}
//left and right side
for(int j=1;j<jy-1;j++)
{
  l = ix*j;
  bc_nodes.push_back(l); u[l] = u_left;  u_new[l] = u_left;
  l = ix*j+ix-1;
  bc_nodes.push_back(l); u[l] = u_right;  u_new[l] = u_right;
}
```

Übung - USA2 - QAD

Sie sehen, dass wir für die Zuweisung der Randbedingungen eine neue Datenstruktur eingeführt haben.

```
std::vector<float>u_bc;
```

Übung - USA2 - QAD

Das Einbauen der Randbedingungen integrieren wir direkt in die Doppelschleife zur Berechnung der Knotenwerte. Dabei kommt eine neue Funktion `IsBCNode` ins Spiel, die wir uns gleich noch näher anschauen. `IsBCNode` soll eigentlich nichts anderes machen, als beim Auftreten einer Randbedingung nichts zu tun (i.e. `continue`). Randbedingungswerte sind gesetzt, müssen also nicht gerechnet werden.

```
for(int j=0;j<jy;j++)
{
  nn = j*ix;
  for(int i=0;i<ix;i++)
  {
    n = nn+i;
    if(IsBCNode(n,bc_nodes))
      continue;
    ...
  }
}
```

Übung - USA2 - QAD

Wie funktioniert nun IsBCNode?

```
bool IsBCNode(int n,std::vector<int>bc_nodes)
{
  bool is_node_bc = false;
  for(int k=0;k<(size_t)bc_nodes.size();k++)
  {
    if(n==bc_nodes[k])
    {
      is_node_bc = true;
      return is_node_bc;
    }
  }
  return is_node_bc;
}
```

Übung - USA2 - QAD

Struktur der Funktion:

- Rückgabewert: logischer Wert wahr oder falsch
- Parameter: aktueller Gitterpunkt und Randbedingungsknotenvektor

Die Funktion überprüft, ob der Gitterpunkt n ein Randbedingungsknoten ist und gibt den entsprechenden logischen Wert zurück.

Übung - USA2 - QAD

Das Ergebnis der finite Differenzen Simulation sehen wir in der Abb.

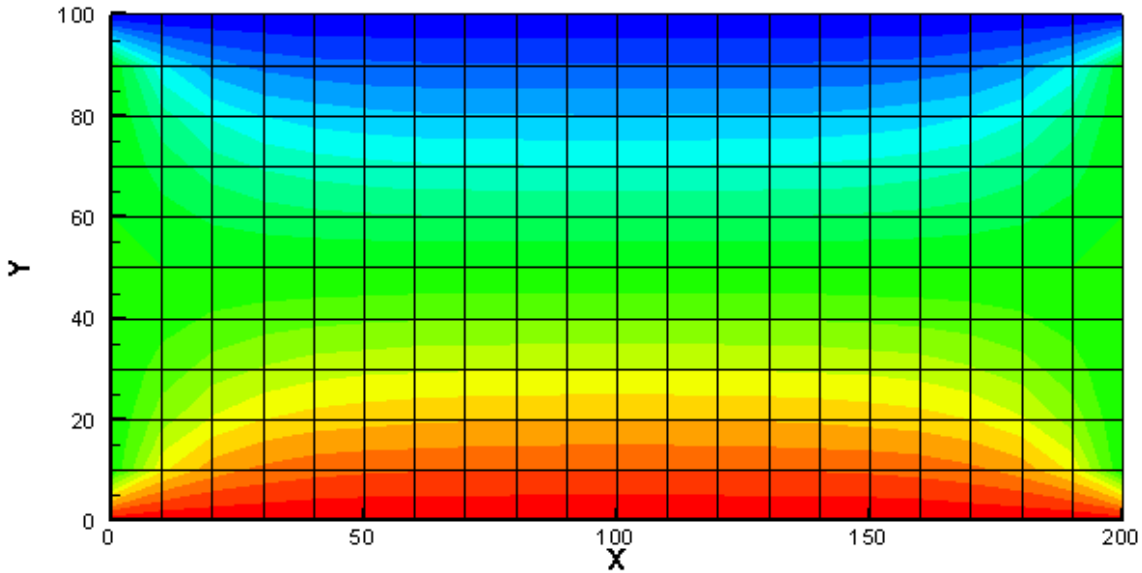


Abbildung: Berechnete Druckverteilung im Rechteck-Aquifer nach 100 Zeitschritten $\Delta t = 25$ sec

Übung - USA2 - QAD

Jetzt werden wir mutig und vergrößern mal den Zeitschritt, sagen wir mal verdoppeln: $\Delta t = 50$ sec. Das Maleur sehen wir in der Abb. Was ist hier los?

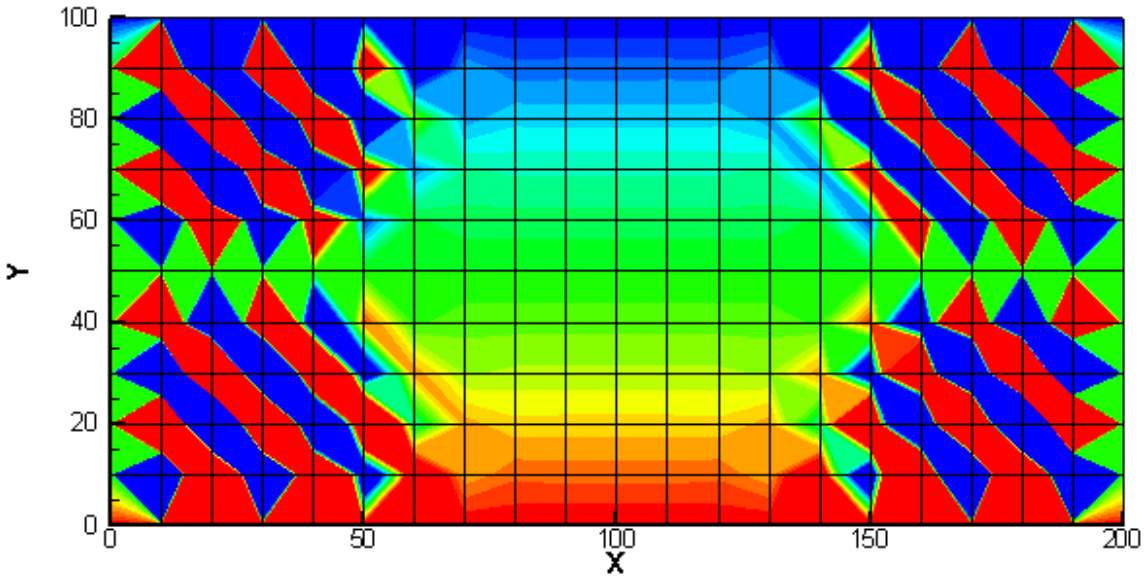


Abbildung: Berechnete Druckverteilung im Rechteck-Aquifer nach 100 Zeitschritten $\Delta t = 50$ sec

Übung - USA2 - QAD

Wir erinnern uns noch dunkel daran, dass der Preis für das einfache explizite FDM ein strenges Stabilitätskriterium war (siehe Hydroinformatik, Teil II, Abschn. 3.2.2 und Abschn. 4.1). Dabei muss die Neumann-Zahl kleiner einhalb sein.

$$Ne = \alpha \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{8}$$

Übung - USA2 - QAD

Prima, aber was ist jetzt α und warum steht nur Δx und nicht auch Δy in der Gleichung? Zur bestimmung des α schreiben wir die Grundwassergleichung in eine Diffusionsgleichung wie folgt um.

$$\frac{\partial h}{\partial t} = \frac{K_x}{S}\frac{\partial^2 h}{\partial x^2} + \frac{K_y}{S}\frac{\partial^2 h}{\partial y^2} + \frac{Q}{S} \tag{9}$$

Übung - USA2 - QAD

Wir sehen, dass es eigentlich zwei α-s gibt, für jede Richtung eins.

$$\alpha_x = \frac{K_x}{S} \tag{10}$$
$$\alpha_y = \frac{K_y}{S}$$

- Welche Einheit hat unser Grundwasser-α ?

Übung - USA2 - QAD

Der richtige Zeitschritt für unser explizites FD Verfahren ergibt sich somit zu:

$$\Delta t \leq \frac{\min(\Delta x^2, \Delta y^2)}{2\alpha} \tag{11}$$

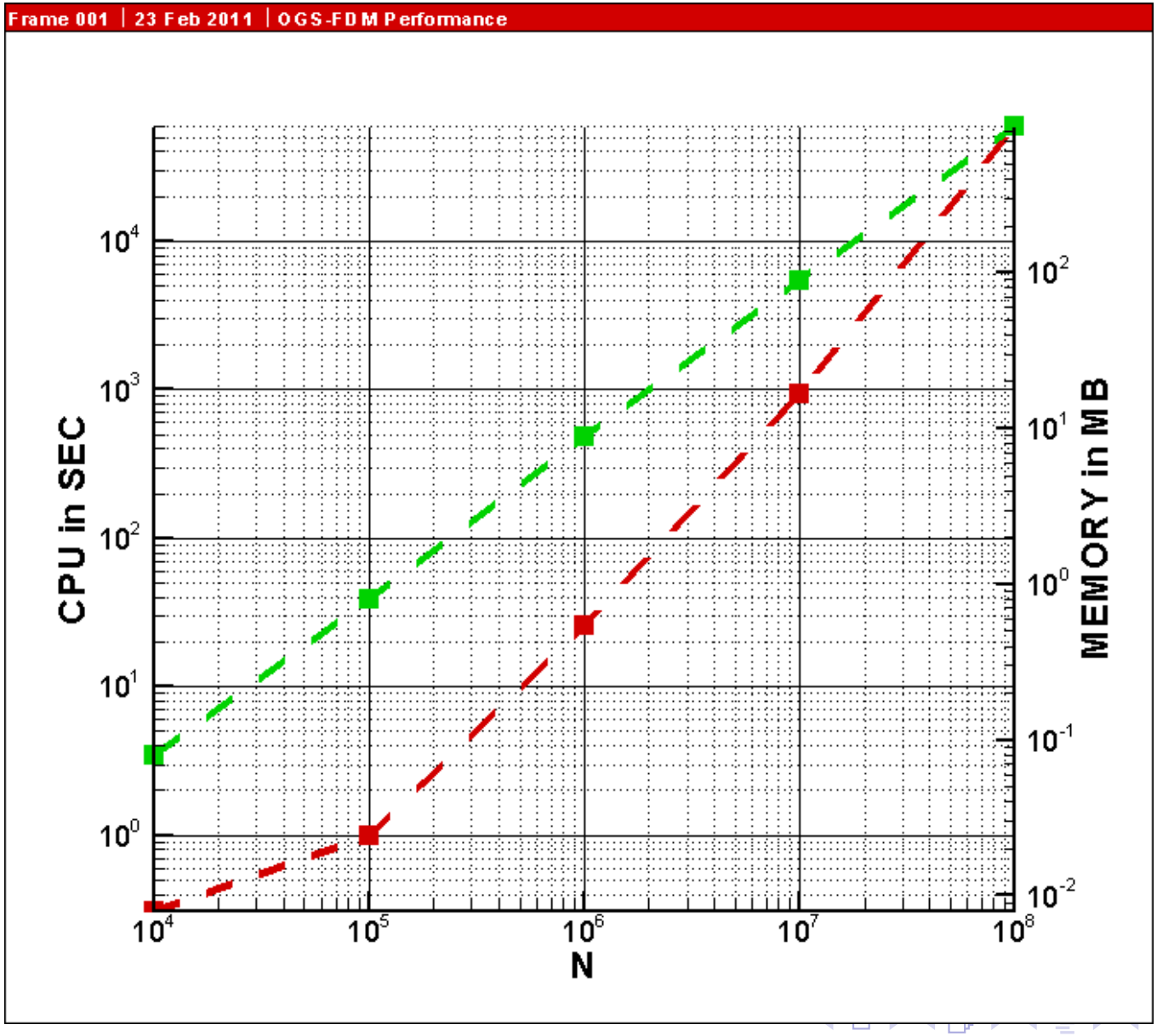
Übung - USA2 - QAD

Die numerischen und hydraulischen Parameter sind in der nachfolgenden Tabelle **??** zu finden. Glücklicherweise sind die Ortdiskretisierungen und die hydraulischen Leitfähigkeiten in beide Koordinatenrichtungen gleich (isotropes Problem).

Übung - USA2 - QAD

$$\Delta t \leq \frac{100 m^2}{2 \times 1 m^2/s} = 50 s \tag{12}$$

Übung - USA2 - QAD



Übung - USA2 - QAD

Wie stellen wir eine Zeitmessung in einem Programm an.

```
clock_t start, end; // Definitionen
...
start = clock();    // Beginn Zeitmessung
...
end = clock();      // Ende Zeitmessung
...
time= (end-start)/ (double)(CLOCKS_PER_SEC); // Differenz
```

Übung USA2 Der Quelltext für diese Übung befindet sich in `EXERCISES\USA2`.