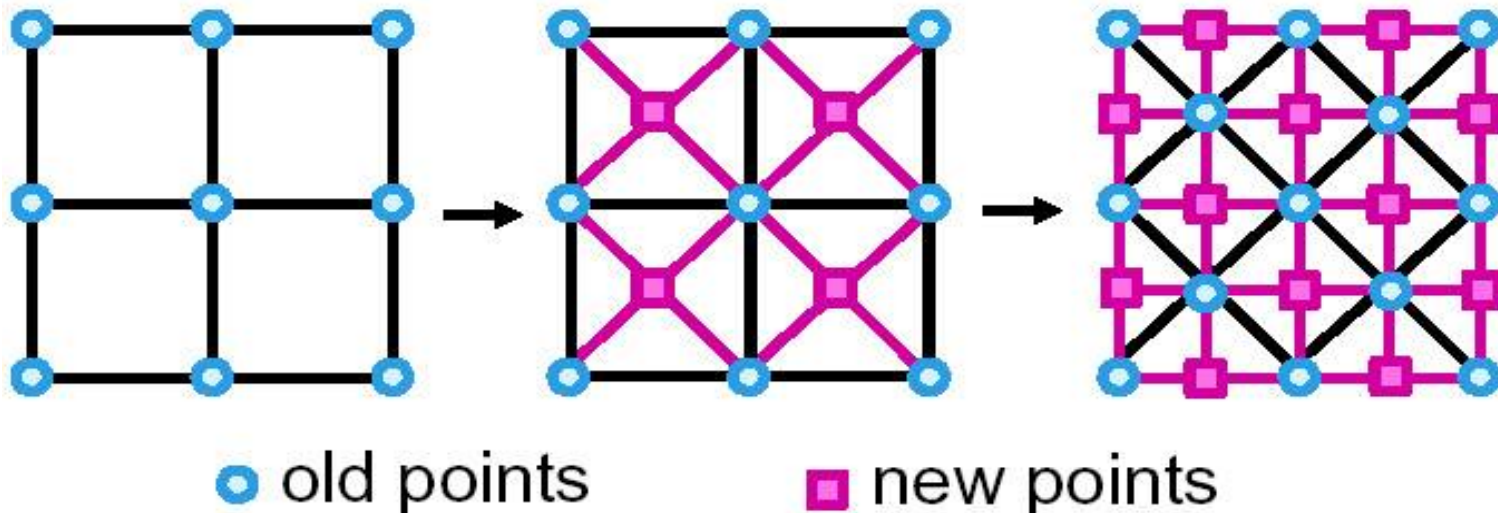


Fraktale Landschaften

Das Erstellen von Bergen und Wolken mit Hilfe von zwei verschiedenen Methoden

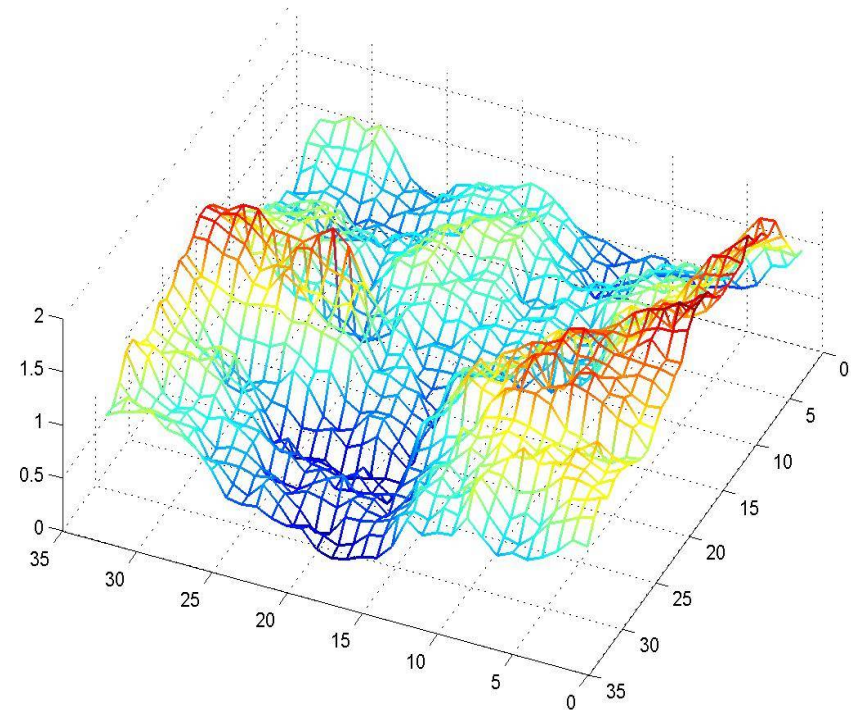
Midpoint Displacement

Zur Erzeugung des Grundgitters verwendet man das Midpoint Displacement. Dadurch kann eine $m \times m$ Matrix schrittweise verfeinert werden. Der Mittelpunkt eines Quadrates ergibt jeweils wieder einen neuen Gitterpunkt. Mit Hilfe eines Parameters kann die Feinheit des Gitters und somit der entstehenden Landschaft reguliert werden.



Höhenwerte erstellen

- Es gibt verschiedene Algorithmen zur Generierung der Höhenwerte, wobei aber alle mit Zufallszahlen arbeiten.
- Die hier verwendete Methode beinhaltet eine $2^{\log 2}$ – Abhängigkeit von der Anzahl der Gitterpunkte.
- Die graphische Darstellung der erzeugten Matrix erfolgte mit einem Surface-Plot.



```
surf(A,'FaceColor','none','EdgeColor','interp');
```

MATLAB - Code

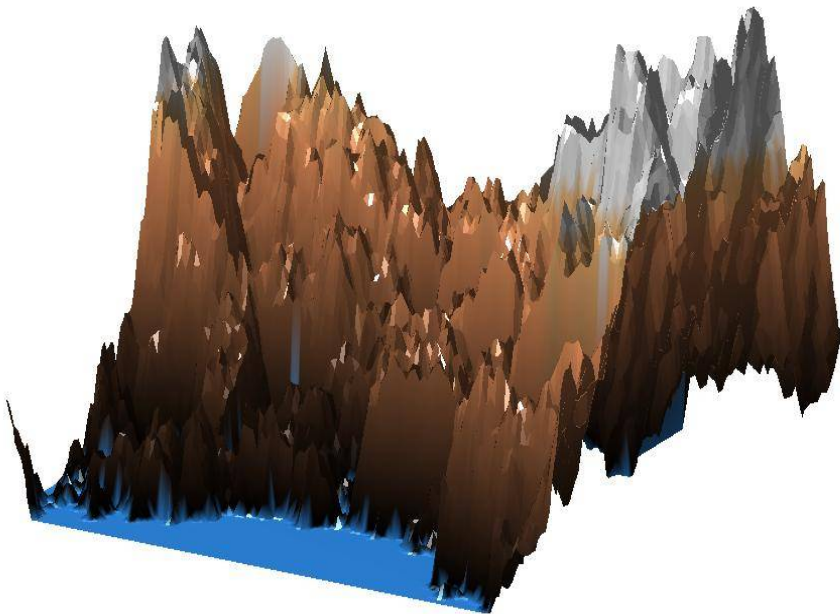
```
A=zeros(N,N);           % Ausgangsmatrix
A([1,N],[1,N])= 1;     % die Ecken duerfen nicht 0 sein

scalef=0.05*(2^(0.9*log2(N-1)));
d=(N+ 1)/2;
ran=rand*2-1;

A(d,d)=mean([A(1,1),A(1,N),A(N,1),A(N,N)])+ scalef*ran;
A(1,d)=mean([A(1,1),A(1,N)])+ scalef*ran;
A(d,1)=mean([A(1,1),A(N,1)])+ scalef*ran;
A(d,N)=mean([A(1,N),A(N,N)])+ scalef*ran;
A(N,d)=mean([A(N,1),A(N,N)])+ scalef*ran;

if N > 3
    A(1:d,1:d)=landschaft(A(1:d,1:d));
    A(1:d,d:N)=landschaft(A(1:d,d:N));
    A(d:N,1:d)=landschaft(A(d:N,1:d));
    A(d:N,d:N)=landschaft(A(d:N,d:N));
end
```

Wasserfläche & Farbverteilung

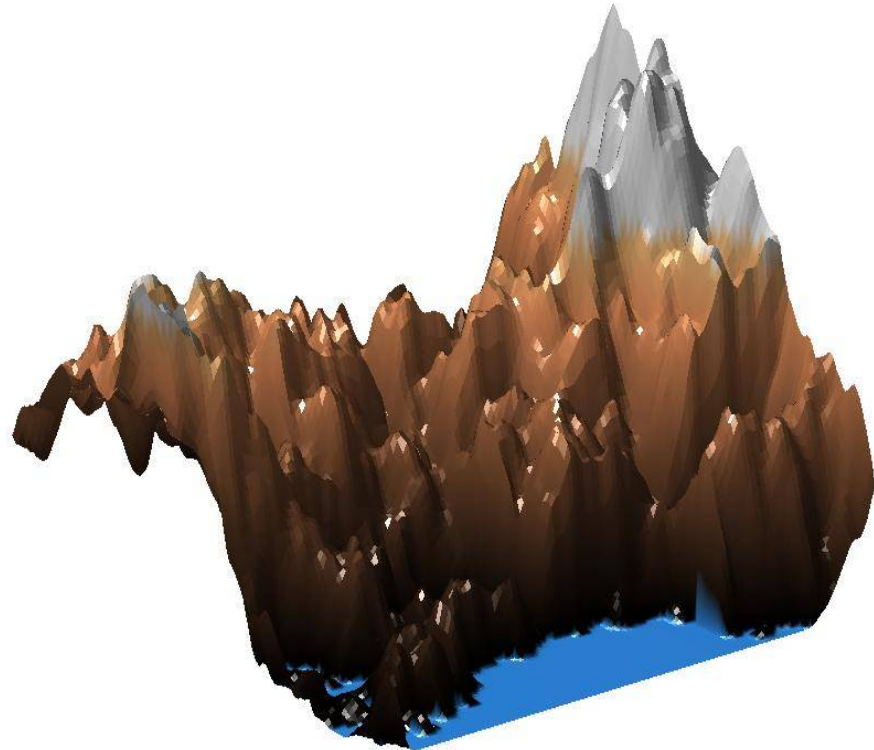


```
colormap([[.2 .6 1]; copper(512);  
         repmat([1 1 1],256,1)]);
```

- Um eine Wasserfläche zu erzeugen, wurden alle negativen Höhenwerte auf Null gesetzt.
- Das Aussehen einer realen Landschaft erhält die Darstellung nun vor allem durch eine realistische Farbgebung.
- Die Colormap enthält dazu einen Blau-Wert, einen Braun-Verlauf und mehrere Weiß-Werte.

Verwendung eines Filters

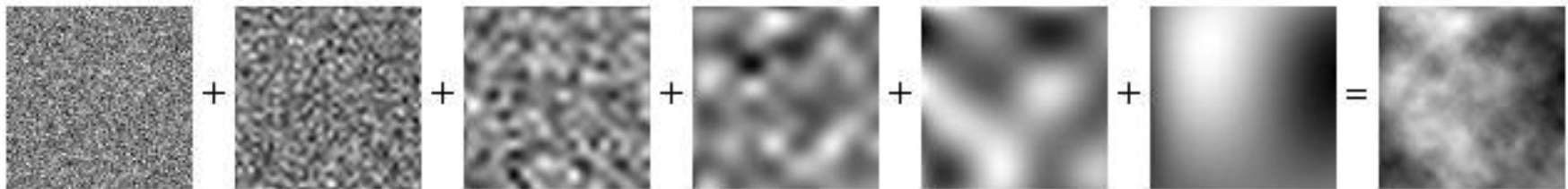
- Da das Bild oft von sehr schroffen Felsen geprägt war, wurde eine Glättung der Daten vorgenommen.
- Dies erfolgte mit Hilfe eines Filters: Matlab-Befehl `filter2`.
- Man erstellt dazu eine Filtermatrix, welche über die Daten gelegt wird und die überdeckten Daten einander annähert.



```
f = 3;  
F = ones(f,f) / f^2;  
AF = filter2(F,A,'valid');
```

Perlin Noise

- Zur Erzeugung realistischer Wolkenmuster benutzt man die Perlin-Noise-Methode.
- Dazu generiert man verschiedene Rauschfrequenzen und überlagert diese miteinander. Dabei werden die niedrigen Frequenzen stärker gewichtet als die hohen.
- Die Dichte und die Schärfe der auf diese Weise erzeugten Wolken, ist durch bestimmte Parameter veränderbar.



MATLAB - Code

```
m=200;      % Grösse des erzeugten Bildes
s=zeros(m);
w=m;       % fuer die Wichtung
i=0;       % Iterationen

while w > 3
    i=i+ 1;
    d=interp2(randn(w), i, 'spline');
    s=s+ i*d(1:m,1:m);
    w=ceil(w/2-1);
end

% Parametereinstellungen
de = 0.5;   % Dichte
sh = 0.1;   % Schaerfe der Wolken

s = 1 - exp(-(s - de)*sh);
s(s<0) = 0;
```


Fraktale Wolken

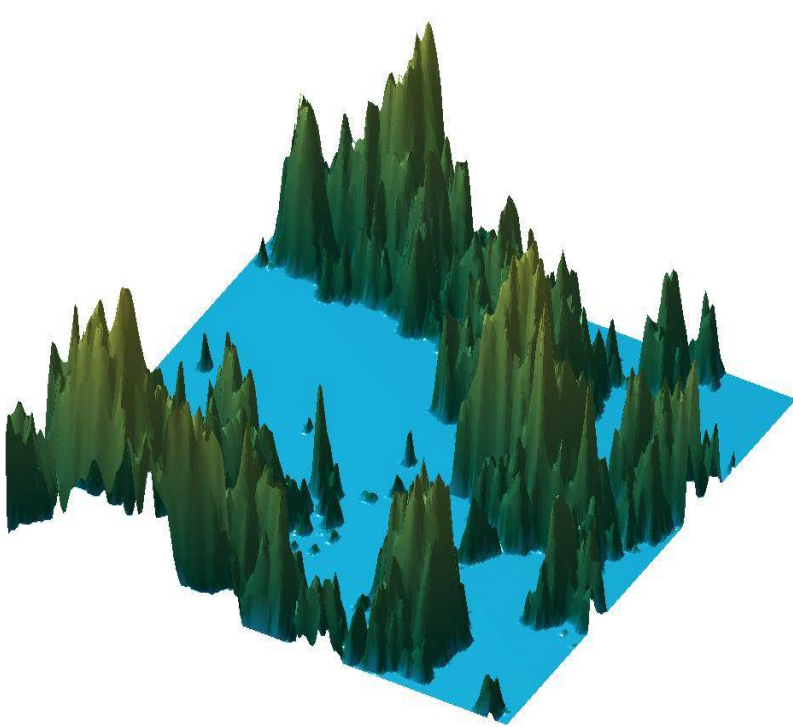


- Die graphische Darstellung der Wolken erfolgt mit dem `imagesc`-Befehl. Dieser skaliert die Daten auf den gesamte Colormapbereich und erzeugt ein 2D-Bild.
- Für eine realitätsnahe Farbgebung wurde eine eigene Colormap mit Blau- und Weiß-Werten erstellt.

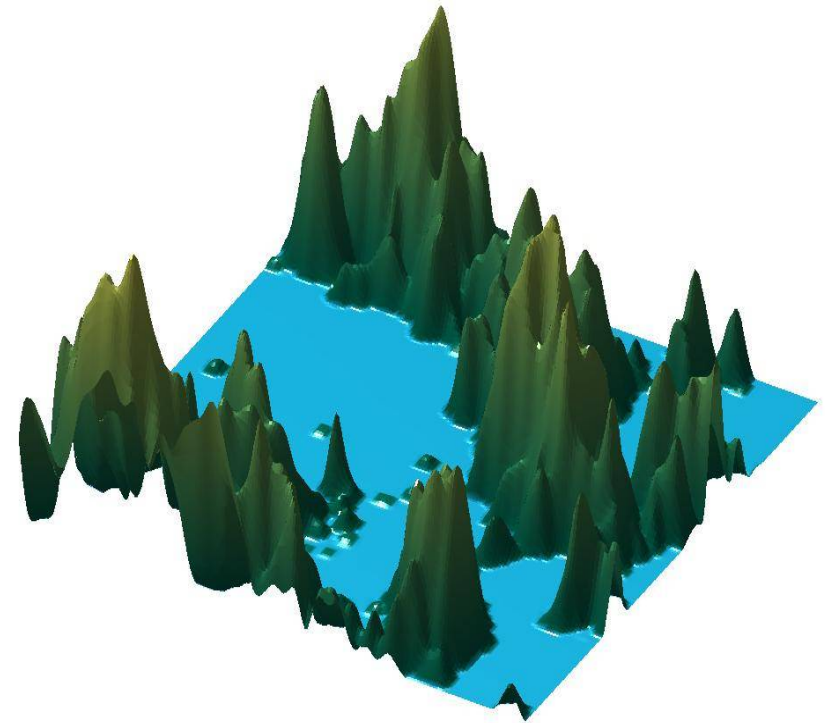
Landschaften mit Perlin Noise

- Der selbe Algorithmus, der für die Erzeugung der Wolken verwendet wurde, kann auch als Landschaftsgenerator benutzt werden.
- Die durch Perlin-Noise erzeugten Daten sind genauso wie bei der Midpoint Displacement Methode eine quadratische Matrix aus Höhenwerten.
- Statt mit dem Image-Befehl werden die Daten diesmal mit einem Surface-Plot dargestellt.
- Genauso wie man die Dichte der Woken regulieren konnte, ist es hier möglich das Verhältnis zwischen Bergen und Wasserfläche zu verändern.

Thailand



Perlin-Landschaft ohne Filter



Perlin-Landschaft mit Filter

Danke für eure Aufmerksamkeit

