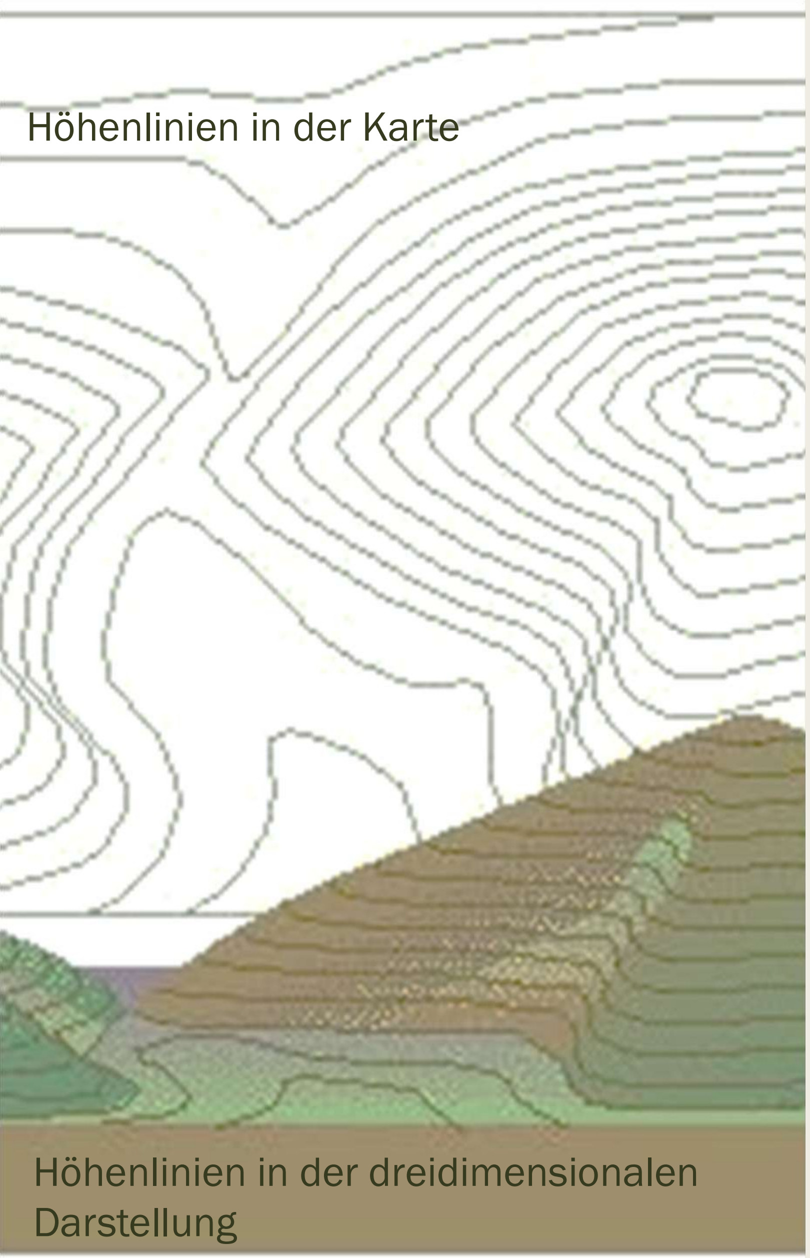
Computergrafik und Informationsvisualisierung

VISUALISIERUNG
TEIL 2

Prof. Dr. Elke Hergenröther

Visualisierung – Teil 2

- Isolinien und Isoflächen in 2D berechnen
- Isoflächen in 3D berechnen
- Visualisierungstechniken zur Darstellung von Dynamik



Begriffserklärung:

Was sind Isolinien?

- Mit einer Isolinie trennt man den Bereich von Werten, die gleich oder größer sind, vom Bereich, dessen Werte kleiner als der gesetzte Isowert ist.
- So kann man für abgestufte Isowerte, unterschiedliche Abstufungen in der Datenmenge visualisieren.
- Anschauliches Beispiel sind die Höhenlinen - eine Variante der Isolinien. Sie teilen das Gelände in zwei Bereiche. Einer ist niedriger und der andere Geländeteil ist höher als der angegebene Höhenwert.

Bild aus: https://sites.google.com/site/marcosbibliothek/home/geografie/karten-lesen/die-karte

Wie berechnet man Isolinien?

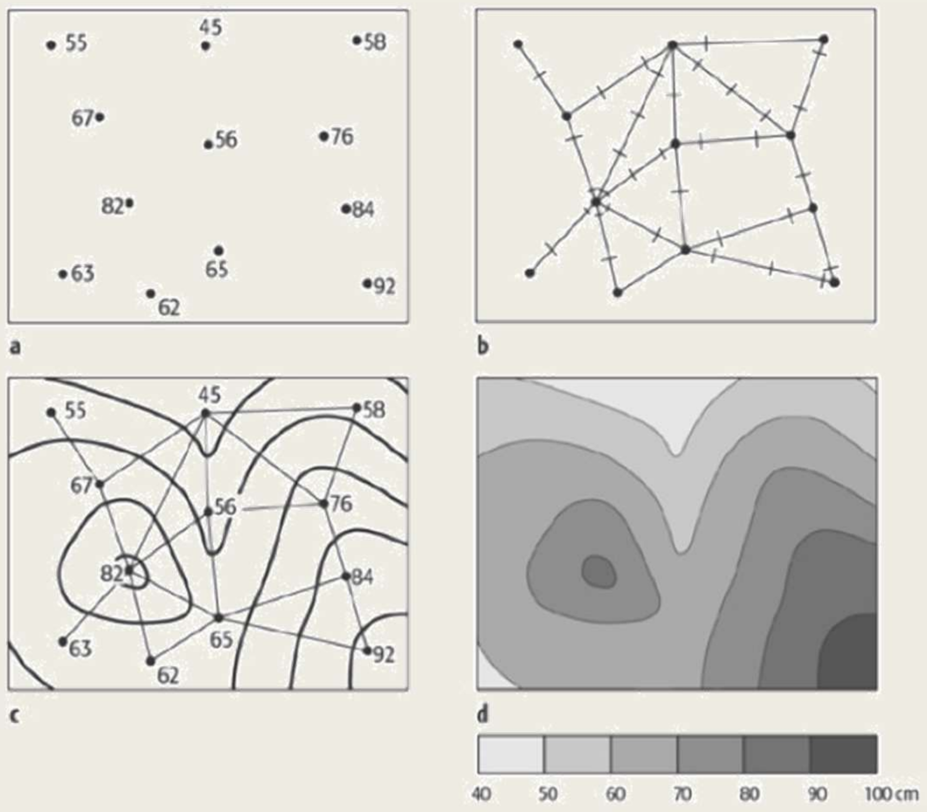
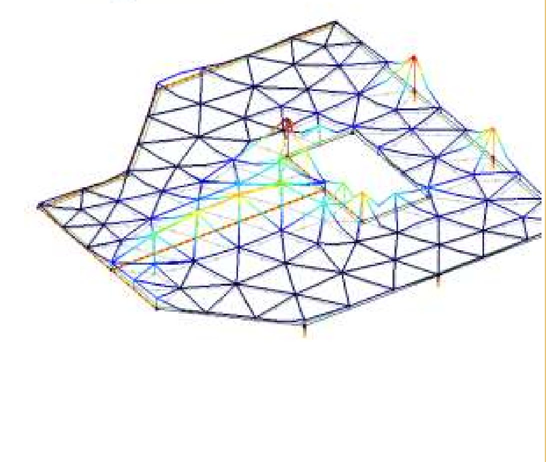


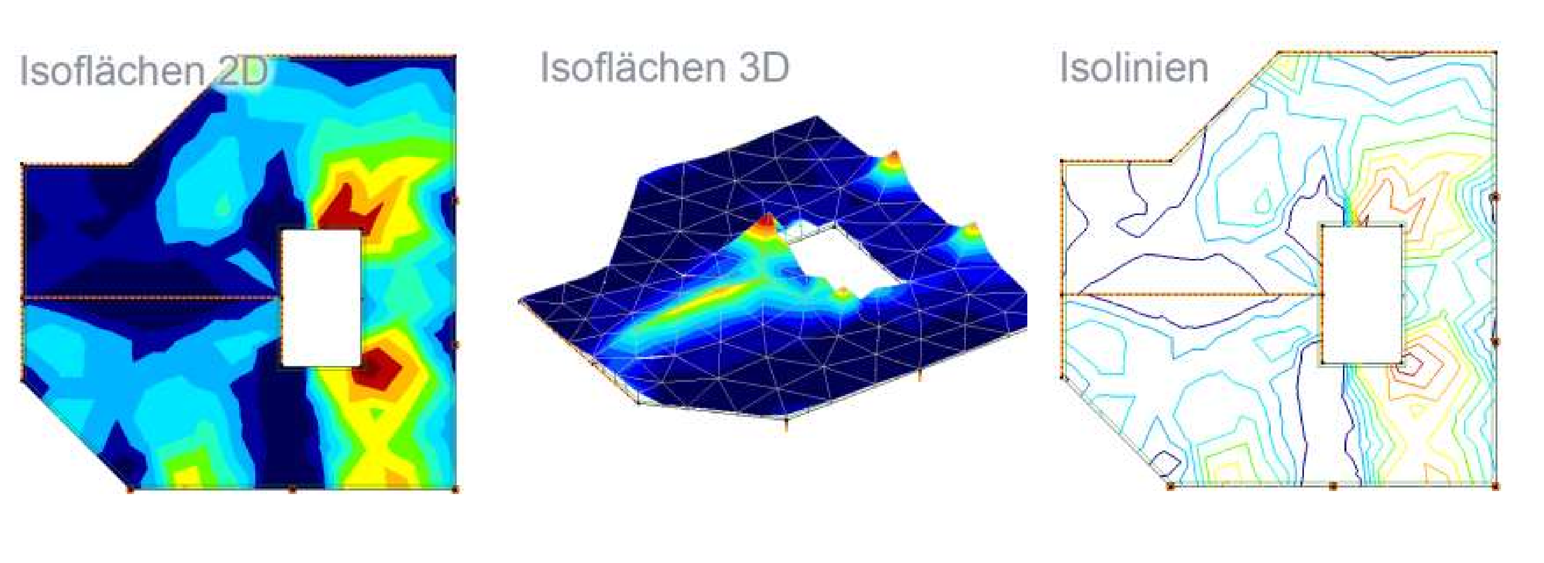
Bild aus: https://www.spektrum.de/lexikon/geographie/isolinien/3885

- Berechnet werden Isolinien durch lineare Interpolation, wie links in der Folge von vier Abbildungen dargestellt.
- Gegeben sind die Isowerte auf dessen Basis die Isolinien erzeugt werden sollen. Im Beispiel links sind das die Werte 40, 50, bis 100.

Unser Ziel: Isolinien und Isoflächen berechnen

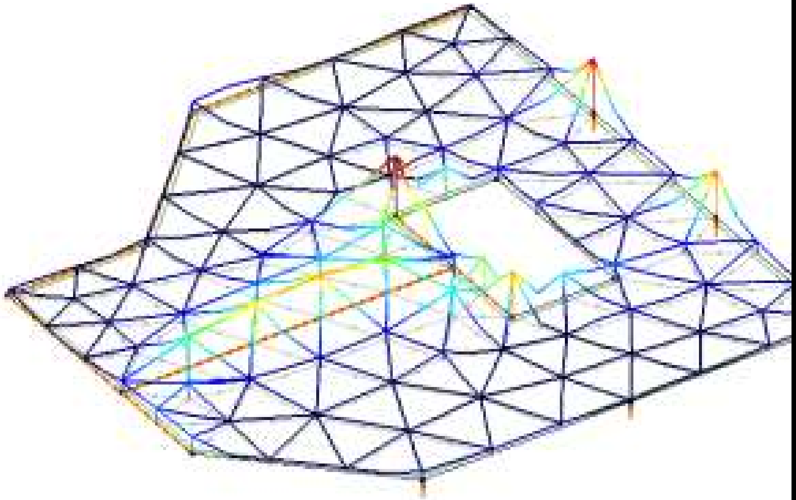
Ausgangssituation: Messwerte, die auf einen nicht regelmäßig strukturierten Gitter vorliegen.



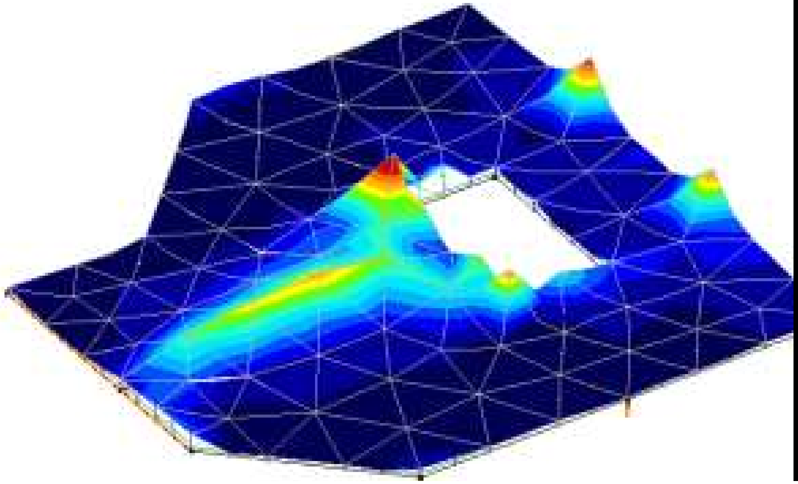


Bilder aus: https://www.ingware.ch/files/m3_10_preview.pdf

Start: Messwerte auf einem unstrukturierten Gitter



Ziel: Isoflächen im 3D-Raum

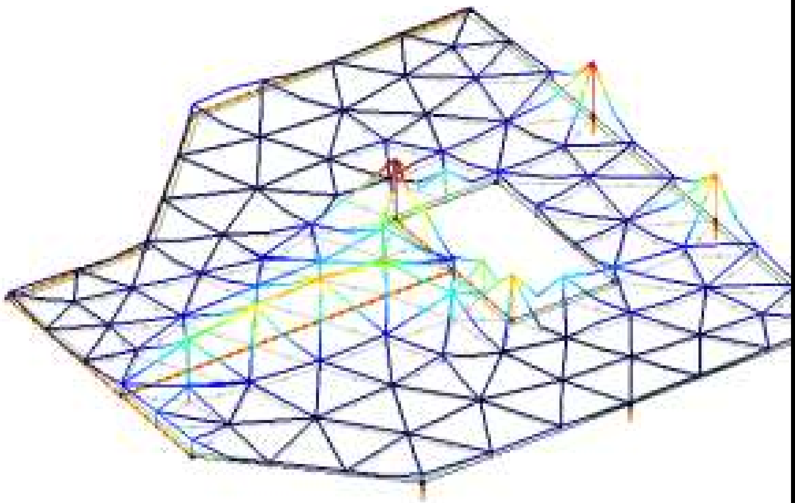


Fragestellung:

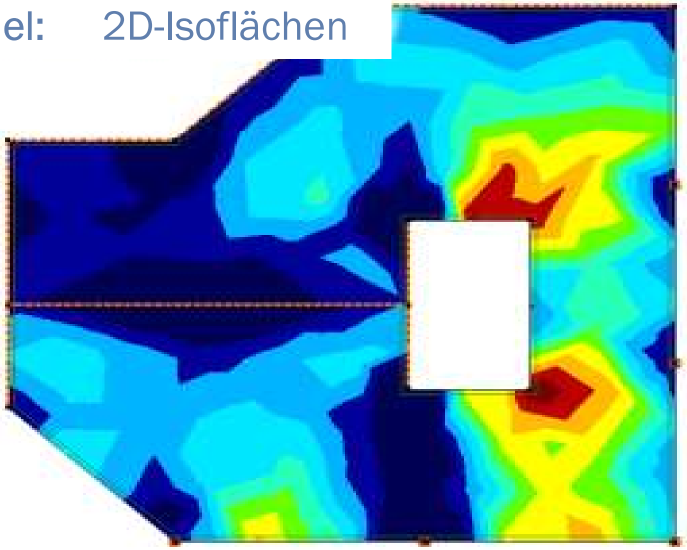
“Wie berechnet man 3D-Isoflächen für Messwerte, die auf einem unregelmäßig strukturierten Gitter vorliegen?“

Bilder aus: https://www.ingware.ch/files/m3_10_preview.pdf

Start: Messwerte auf einem unstrukturierten Gitter



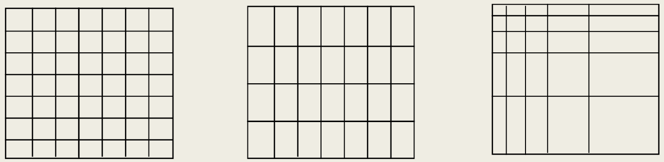
Ziel: 2D-Isoflächen



Komplexität des Problems reduzieren:

“Wie berechnet man 2D-Isoflächen für Messwerte, die auf einem strukturierten Gitter vorliegen?“

- Hierfür müssen wir nur die Höhenwerte auf Null setzen.
- Um die Isolinien und Isoflächen einfacher berechnen zu können, müssen wir die Daten in ein strukturiertes Gitter transformieren:



Oder anders herum Marching Squares und Marching Cubes Verfahren sind für rechtwinklige Gitter ausgelegt!

Bilder aus: https://www.ingware.ch/files/m3_10_preview.pdf

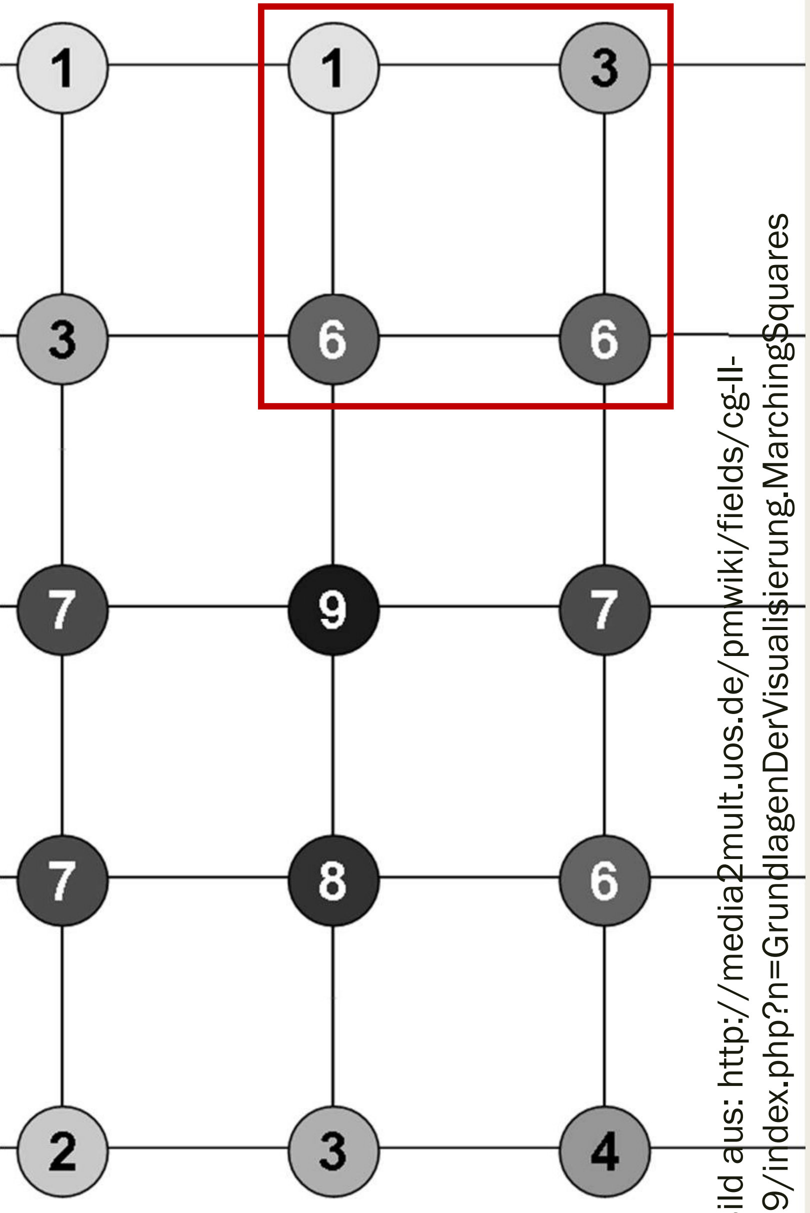


Bild aus: http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=GrundlagenDerVisualisierung.MarchingSquares

Wie berechnet man Isolinien algorithmisch?

- Um Isolinien automatisiert berechnen zu können, reduziert man die Komplexität. Ziel war ursprünglich die Visualisierung von Simulationsdaten.
- Statt einer *unregelmäßig strukturierten* Datenbasis wählt man in der klassischen Variante eine regelmäßig *strukturierte* Datenbasis. Am besten ein kartesisches Gitter, da hier die Abstände zwischen den Gitterpunkten immer gleich ist.
- Ebenfalls zur Komplexitätsreduzierung nutzt Marching Squares (ein Viereck ist beispielhaft rot eingezeichnet) eine Divide-and-Conquer-Strategie, um alle *Zellen* des Datensatzes unabhängig voneinander zu untersuchen.
- Bei jeder Zelle wird für sich entscheiden, *ob und wie diese durch die Isolinie durchsetzt* wird.
- Das algorithmische Verfahren heißt: „Marching Squares“

Marching Squares

Die Grundannahme der Methode ist: Wenn man auf die genaue Angabe der Schnittpunkte der Isolinie mit den Kanten einer Zelle verzichtet, gibt es nur eine endliche Anzahl an Möglichkeiten für den Verlauf der Isolinie in einer Zelle. Je nachdem, welche Eckpunkte der Zelle größer, gleich bzw. kleineren dem Iso-Wert sind. Es gibt 16 verschiedenen Möglichkeiten – siehe rechts.

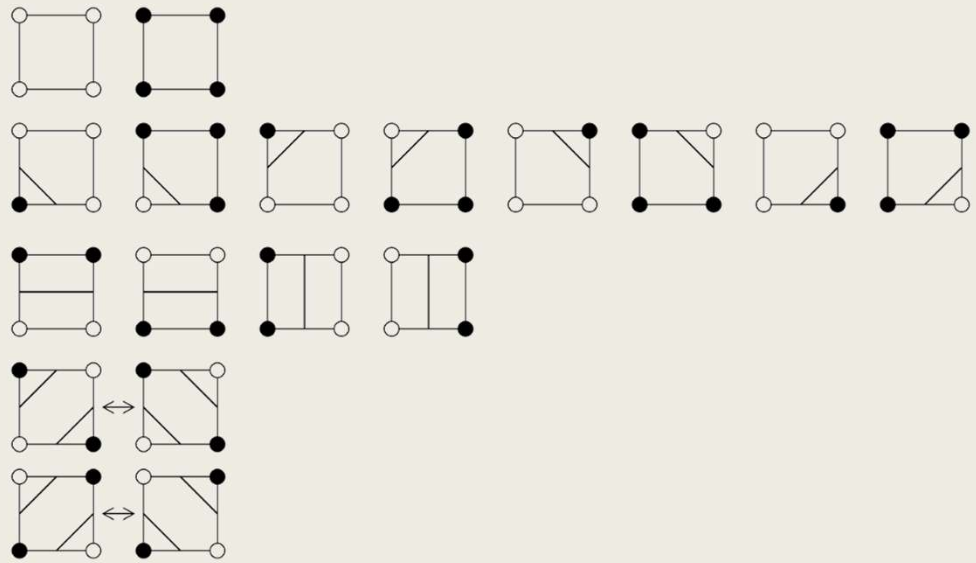


Bild aus: http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=GrundlagenDerVisualisierung.MarchingSquares

Marching Squares

Die schwarzen Punkte symbolisieren die Punkte deren Werte <u>über</u> dem Iso-Wert liegen oder gleich dem Iso-Wert sind und die weißen Punkte symbolisieren die Gitterpunkte deren Werte <u>unter</u> dem Iso-Wert liegen.

In der Abbildung sind alle Möglichkeiten inklusive der Symmetrien aufgelistet. Man sieht, dass in jedem Fall die schwarzen Punkte von den weißen (und umgekehrt) isoliert sind. Die unteren beiden Fälle sind offensichtlich nicht eindeutig, weil in diesen beiden Fällen beide Konturierungen des Pixels möglich sind.

Diese Fälle kann man bequem in einer Lookup-Tabelle speichern.

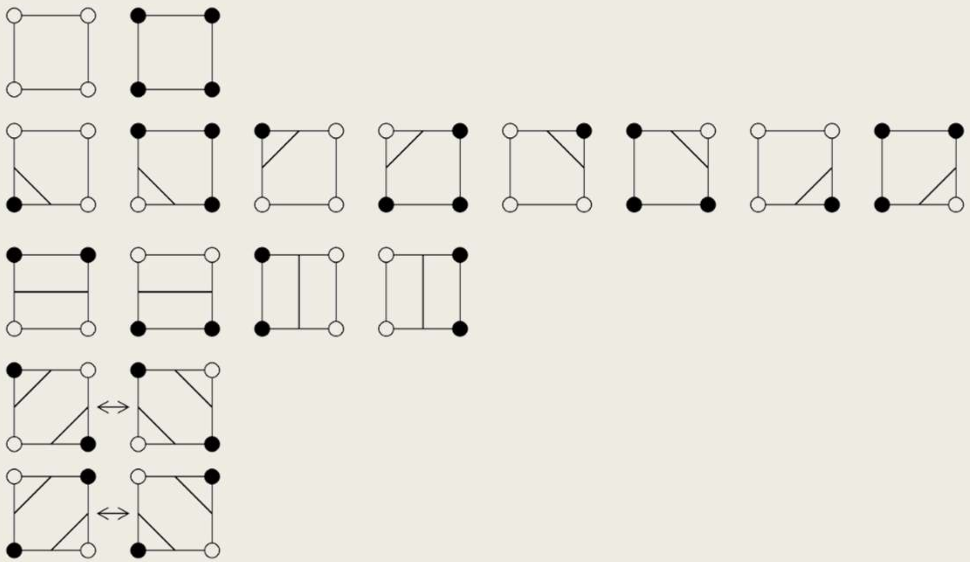
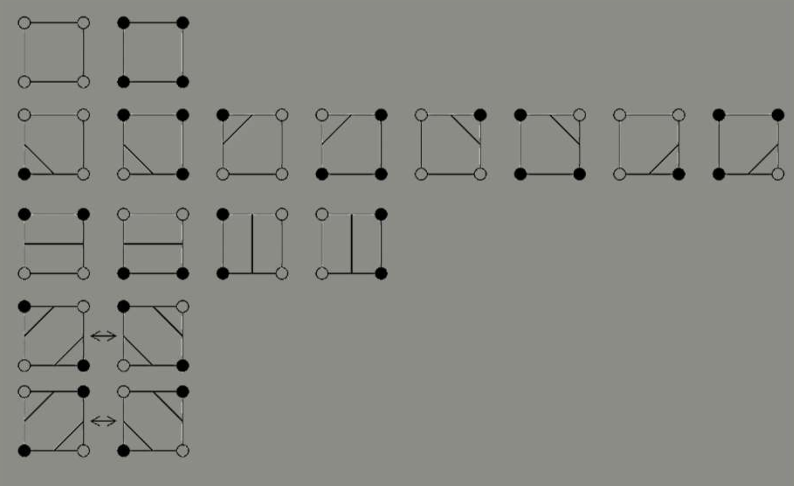


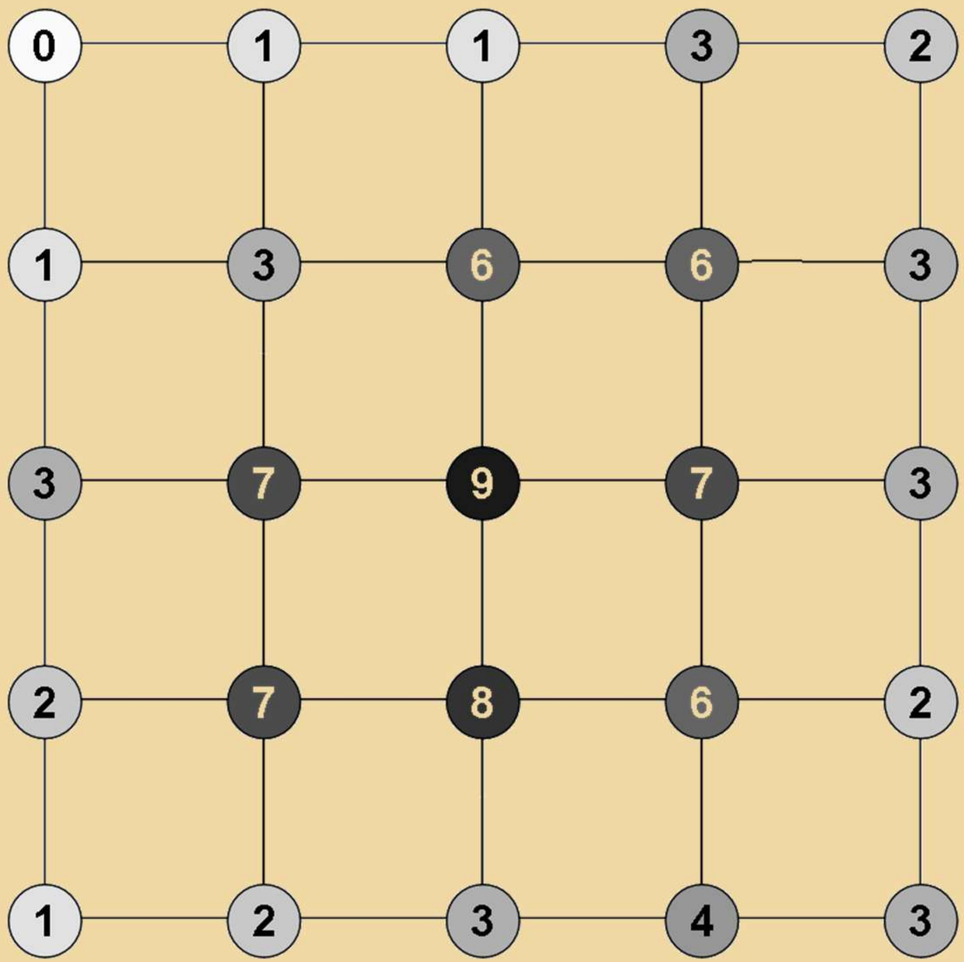
Bild aus: http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=GrundlagenDerVisualisierung.MarchingSquares

Mit Marching Squares Isolinien erzeugen:

mit Isowert = 6

Algorithmus mit Hilfe der in einer Lookup-Tabelle gespeicherten 16 Unterscheidungsfällen.



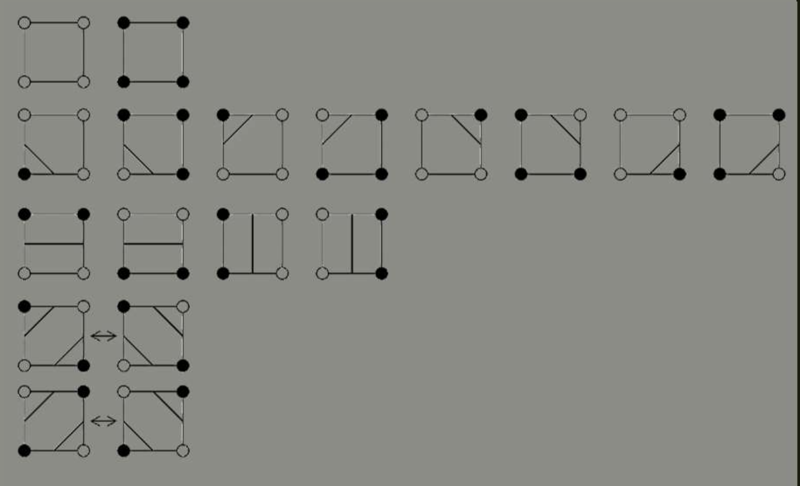


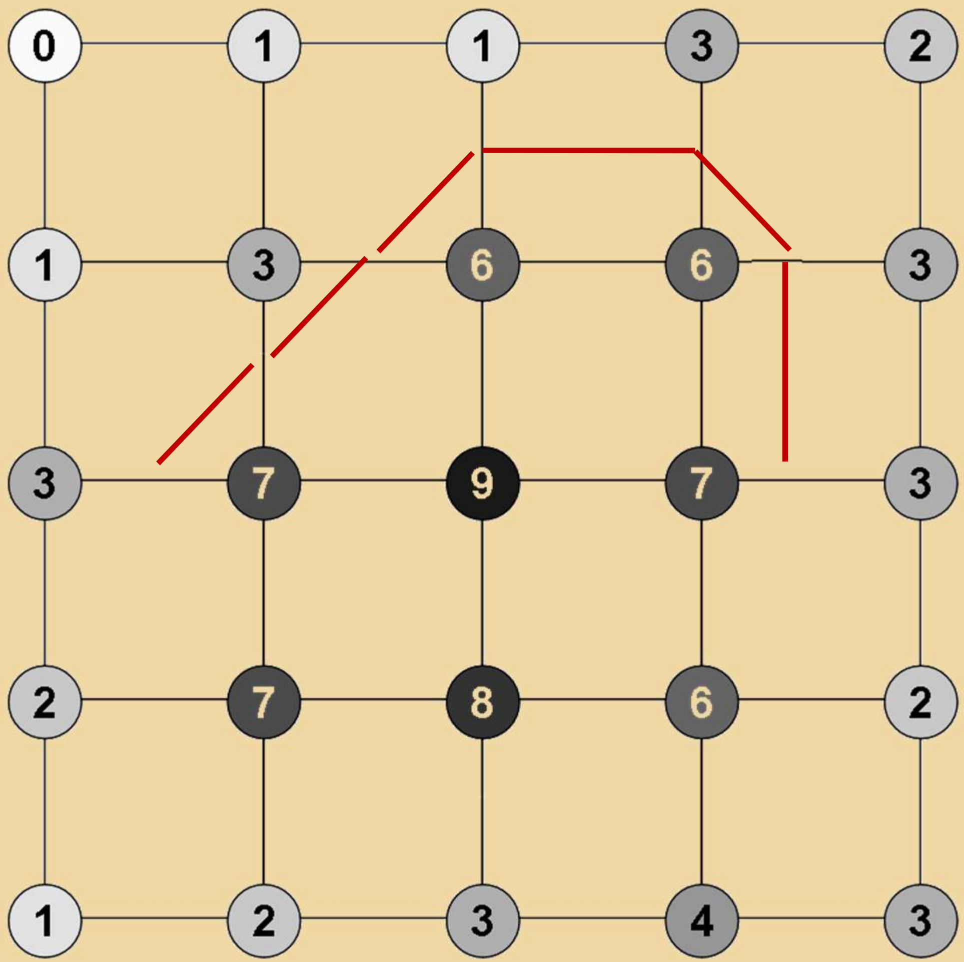
Bilder aus: http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=GrundlagenDerVisualisierung.MarchingSquares

Mit Marching Squares Isolinien erzeugen:

mit Isowert = 6

Algorithmus mit Hilfe der in einer Lookup-Tabelle gespeicherten 16 Unterscheidungsfällen.





Bilder aus: http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=GrundlagenDerVisualisierung.MarchingSquares

Ergebnis des Marching Squares

mit Isowert = 6

Die schwarzen Punkte haben Werte größer, gleich 6 und die weißen haben kleiner Werte.

Der Marching Squares Algorithmus berechnet nur, durch welche Kanten die Isolinie geht, ohne die konkreten Schnittpunkte anzugeben.

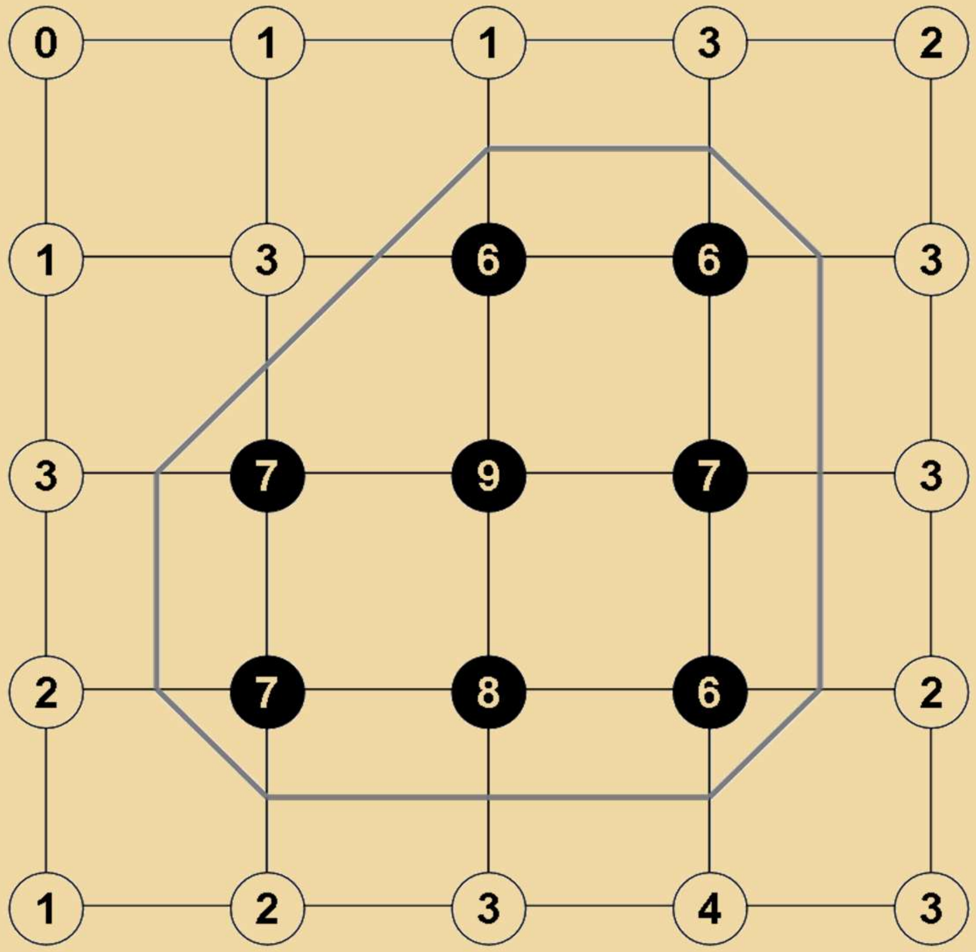


Bild aus: http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=GrundlagenDerVisualisierung.MarchingSquares

Ergebnis des Marching Squares

mit Isowert = 6

Zur Berechnung der tatsächlichen Schnittpunkte (rot) ist eine lineare Interpolation nötig (angedeutet durch die Pfeile). Diese Schnittpunkte bestimmen nun den endgültigen Verlauf der Isolinie (grüne).

1.

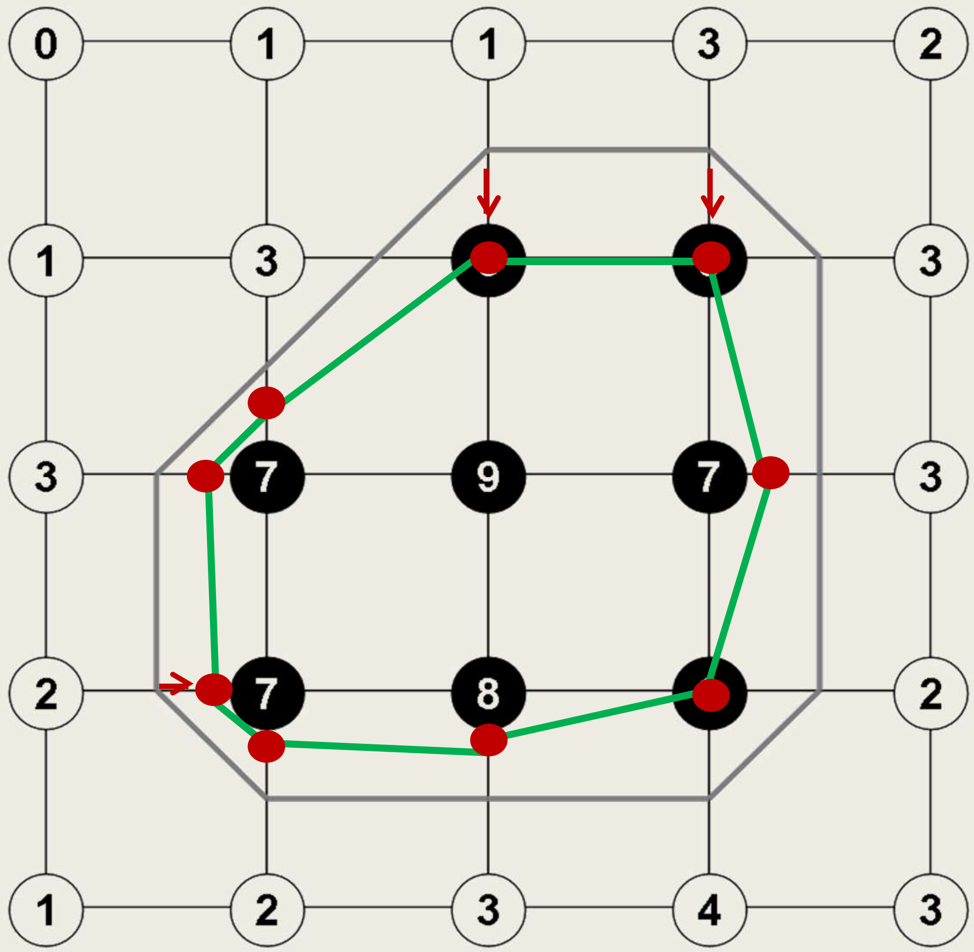


Bild aus: http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=GrundlagenDerVisualisierung.MarchingSquares

Ergebnis des Marching Squares

mit Isowert = 6

Anschließend kann der Verlauf der Isolinie mit einem Spline, der zwischen den Stützpunkten interpoliert, geglättete werden.

Vorsicht! Durch den Verlauf der Spline kann der Verlauf der Isolinie evtl. etwas verfälscht werden.

Ein Vorteil der Splines ist, dass man bei einigen Verfahren die Anzahl der zu berechnenden Stützpunkte einstellen kann.

1.

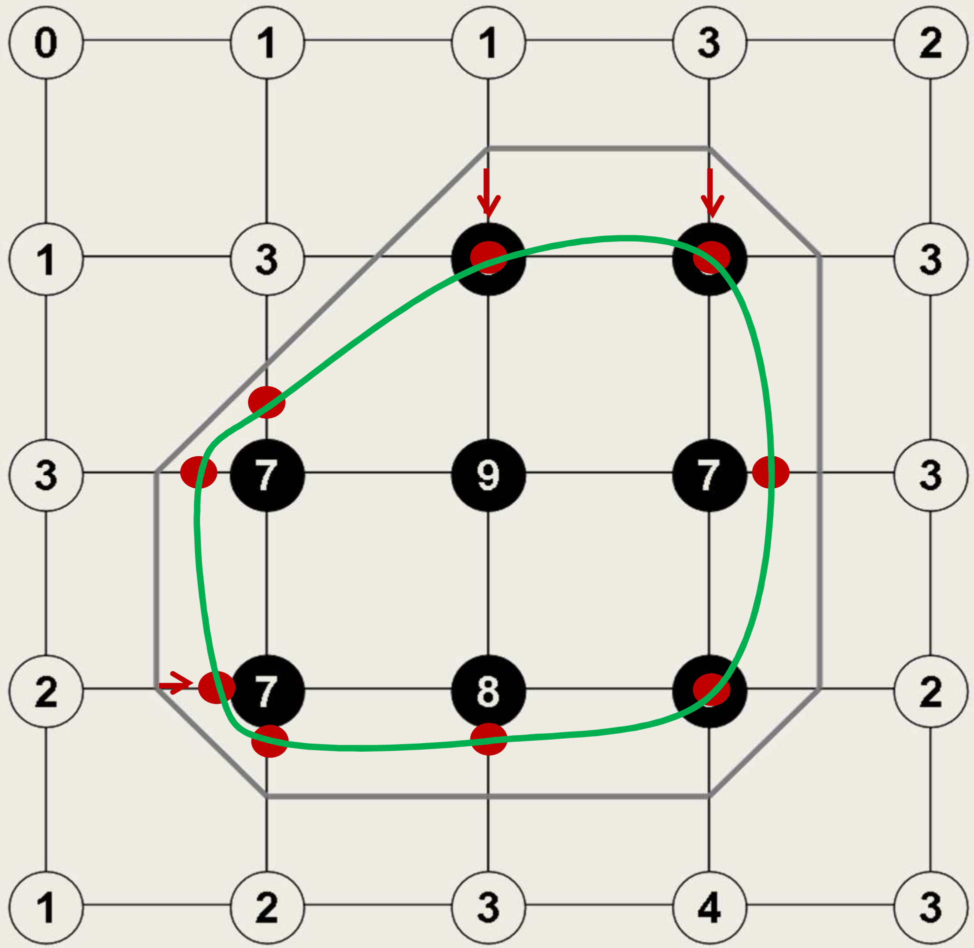
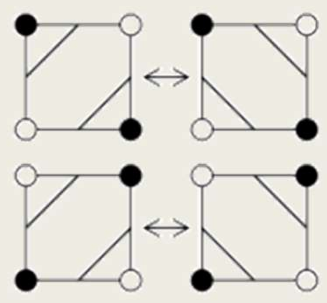
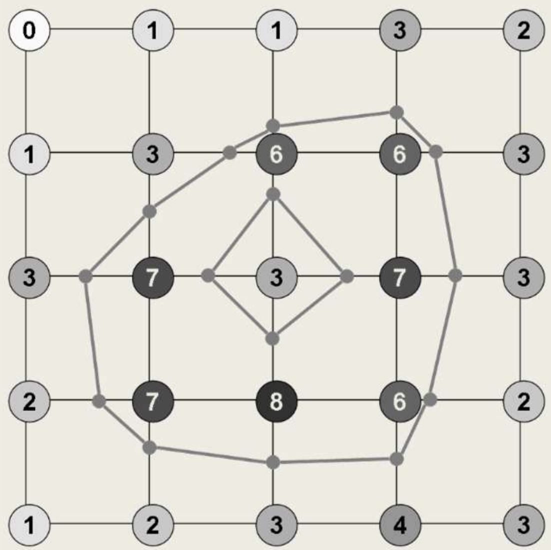
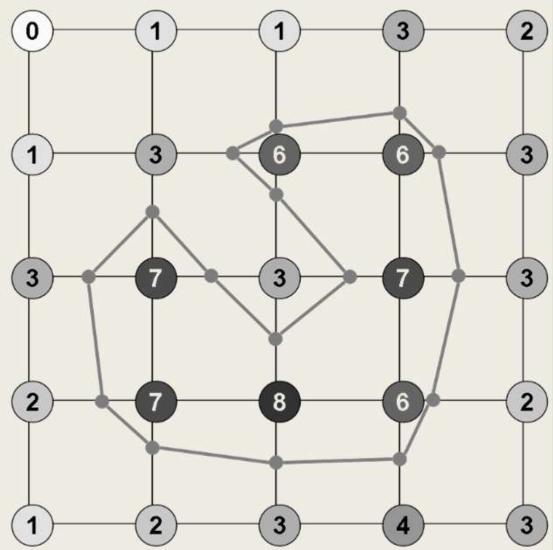


Bild aus: http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=GrundlagenDerVisualisierung.MarchingSquares

Nicht eindeutige Fälle des
Marching Square







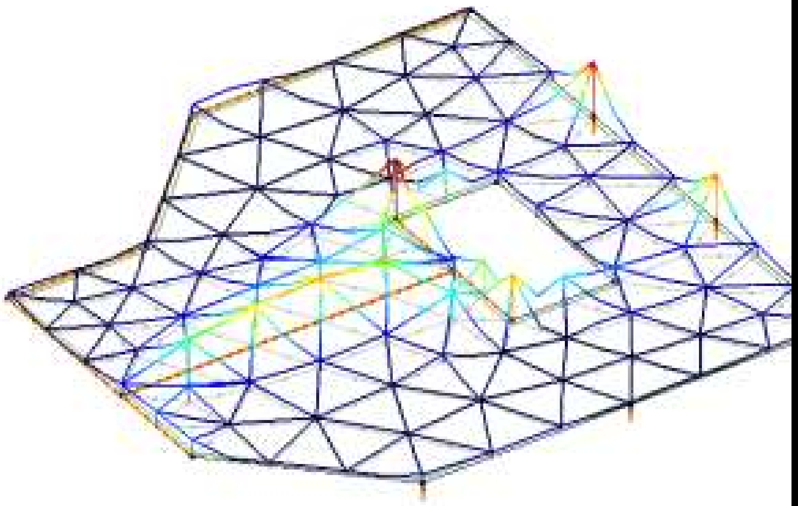
Bilder aus: http://media2mult.uos.de/pmwiki/fields/cg-II-09/index.php?n=GrundlagenDerVisualisierung.MarchingSquares

Nicht eindeutige Fallunterscheidungen:

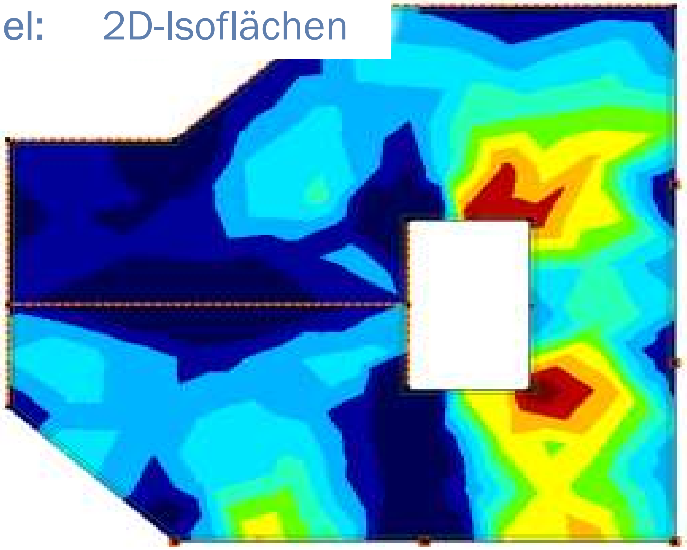
Wie bereits erwähnt, gibt es zwei Fälle die nicht eindeutig sind. Wenn man in dem vorherigen Beispiel die 9 in der Mitte durch eine 3 ersetzt (wodurch neue vier Schnittpunkte um diese 3 entstehen), dann kann die Konturierung zwei unterschiedliche Ergebnisse liefern. Egal welche Variante ausgewählt wird, die Isolinen bleiben geschlossen.

Isowert ist immer noch 6.

Start: Messwerte auf einem unstrukturierten Gitter



Ziel: 2D-Isoflächen



„Wie berechnet man 2D-Isoflächen für Messwerte, die auf einem unstrukturierten Gitter vorliegen?“

- Zunächst brauchen wir eine Methode, um die Messwerte, die auf einem unregelmäßig strukturierten Gitter vorliegen, auf ein regelmäßiges, am besten kartesisches Gitter zu übertragen oder wir nehmen die ursprünglichen Messwerte, die meistens auf einem kartesischen oder regelmäßig strukturierten Gitter angeordnet sind.
- Auf Basis der Gitterwerte können wir mit Marching Squares Isolinien berechnen kann.
- Überlegen Sie sich einen Lösungsweg, um das Ziel, die 2D-Isoflächen zu visualisieren, zu erreichen.

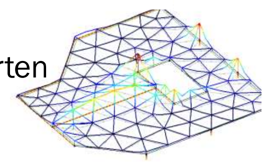
Bilder aus: https://www.ingware.ch/files/m3_10_preview.pdf

Simpler Lösungsweg

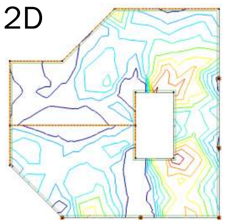
2D-Isoflächen berechnen:

- Isolinien anhand der vorgegebenen Isowerte in 2D berechnen.
- Aus den Punkten einer Isolinie ein gefülltes Polygon erzeugen. In OpenGL verwenden wir dazu das Primitiv GL_POLYGON. Die Triangulierung übernimmt OpenGL.
- So erhält man Isoflächen. Die Dreiecke einer Isofläche haben alle die gleiche Farbe. Der passende Farbwert ist wird aus einer Farbtabelle ausgelesen. Falls der exakte Wert nicht darin enthalten ist, wird der passende Farbwert interpoliert.
- Dem Painterverfahren entsprechend, kann man nun zuerst die Polygone mit den niedrigsten Isowert rendern und dann schrittweise die Isoflächen mit den höheren Werten. Die letzten gerenderten Isoflächen im Beispiel unten sind die dunkelroten. Das Ziel Isoflächen in 2D-Raum ist erreicht.

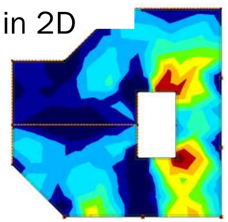
Ausgangssituation: Werte im unstrukturierten Gitter



Isolinien in 2D berechnen

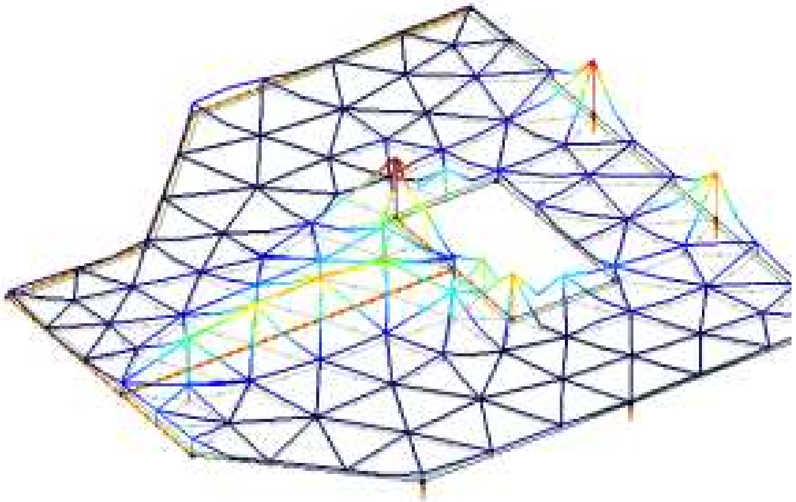


Isoflächen in 2D berechnen

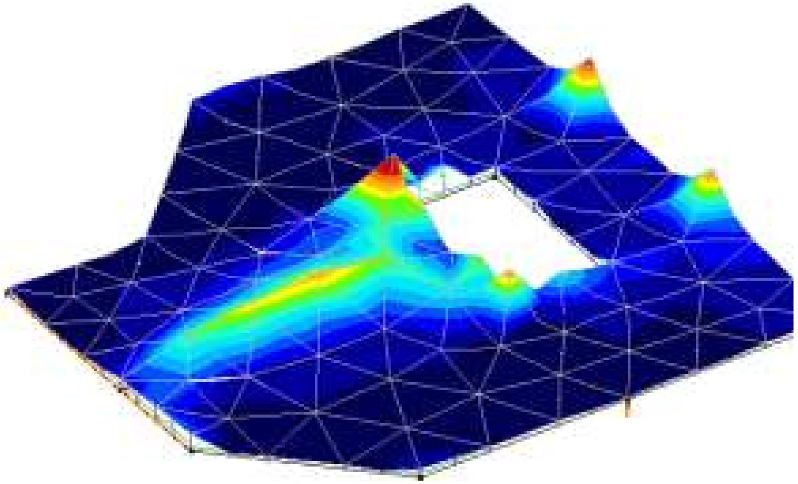


Bilder aus: https://www.ingware.ch/files/m3_10_preview.pdf

Start: Messwerte auf einem unstrukturierten Gitter



Ziel: Isoflächen im 3D-Raum



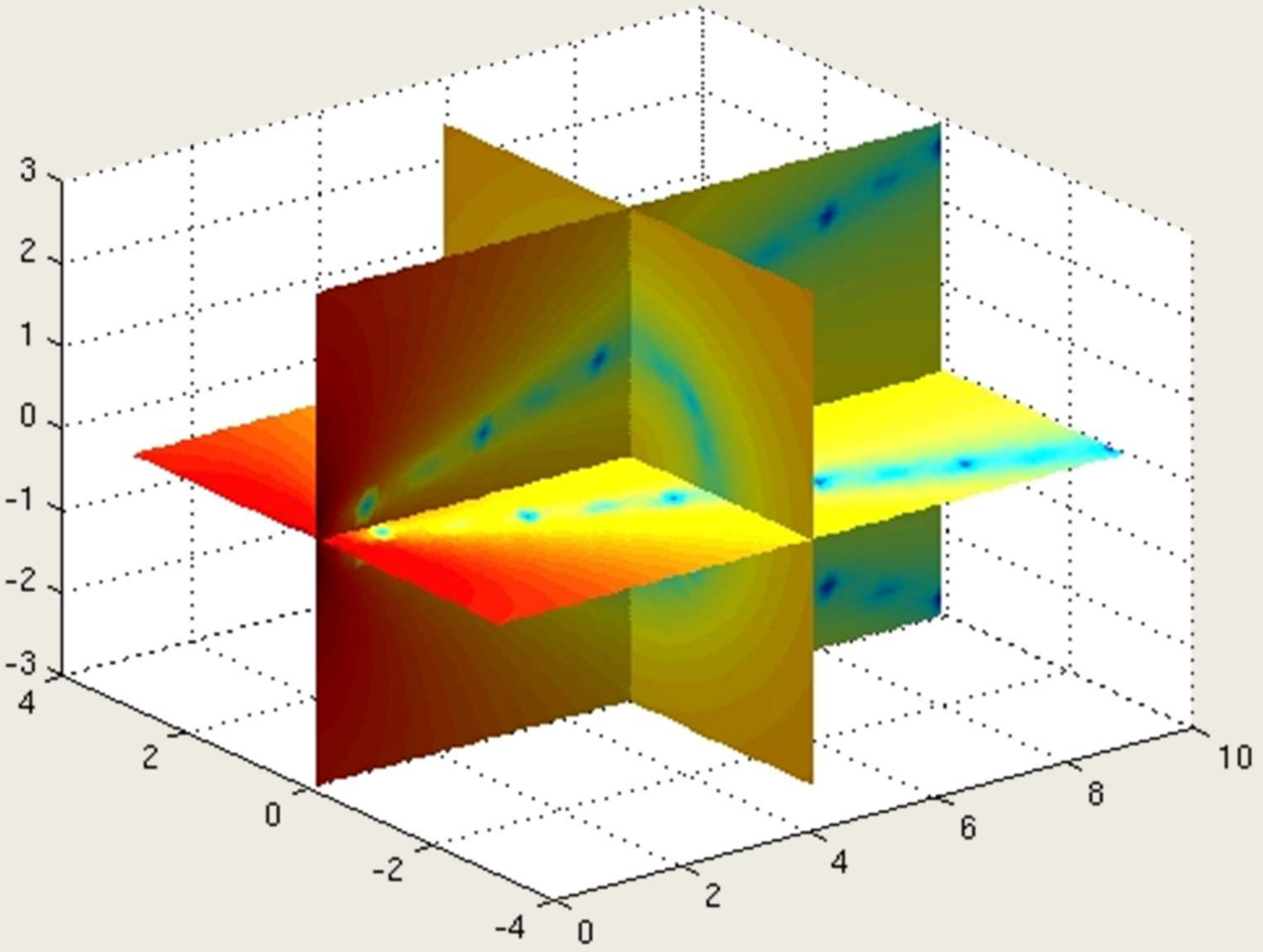
"Wie berechnet man 3D-Isoflächen für Messwerte (inkl. Höhenwerte), die auf einem regelmäßigen kartesischen Gitter vorliegen?"

- Zur Lösung des Problem hilft, dass die Isolinien immer ineinander geschachtelt sind:

  Ein Beispiel: Wenn die Isolinien im Abstand 10 erstellt werden, dann liegen die Isolinien mit dem Wert 40 innerhalb der Isolinien mit dem Wert 50. Isolinen mit dem Wert 30 werden immer innerhalb der 40er und nie innerhalb der 50er-Linien liegen.

- Weiterhin hilft vielleicht die Annahme, dass Sie die Auflösung der Isolinien (durch Spline-Interpolation) beliebig steuern können.

Bilder aus: https://www.ingware.ch/files/m3_10_preview.pdf



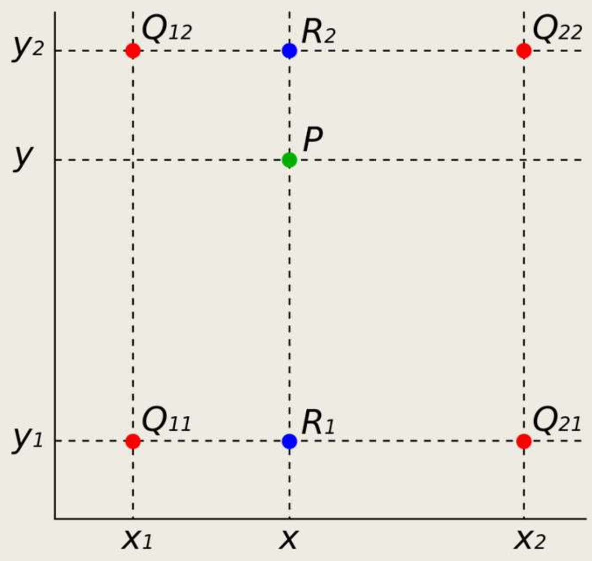
Isolinen und Isoflächen

Häufig nutzt man Isolinen und Isoflächen, um Schnitte durch 3D-Volumen zu berechnen.

Zur Berechnung benötigt man die Methode der Bi- und Trilinearen Interpolation

Bild aus: https://www.bu.edu/tech/support/research/training-consulting/online-tutorials/visualization-with-matlab/

Bilineare Interpolation im 2D Raum



Aufgabe: Wert für P anhand der Werte die auf den Gitterpunkten Q_{12} bis Q_{22} liegen zu berechnen.

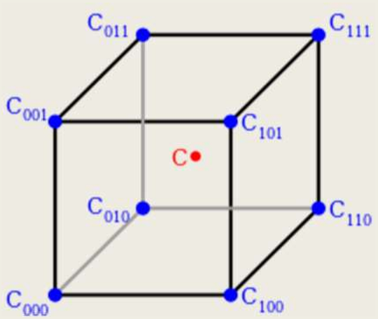
Erste Idee: Abstandsberechnung – teure, wegen Wurzelberechnung

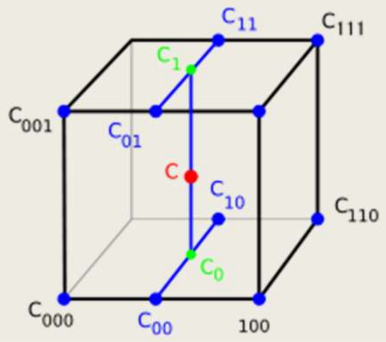
Bessere Idee: Bilineare Interpolation

- auf Q_{12} bis Q_{22} sind die Messwerte lokalisiert, die interpoliert werden sollen
- P ist der Punkt für den die Werte interpoliert werden sollen.
- Zuerst werden die Werte von Q_{12} und Q_{22} für R_1 sowie Q_{11} und Q_{21} für R_2 linear interpoliert
- Danach werden die Werte auf R_2 und R_1 für P linear interpoliert.

Aus https://de.wikipedia.org/wiki/Bilineare_Filterung

Trilineare Interpolation im 3D Raum



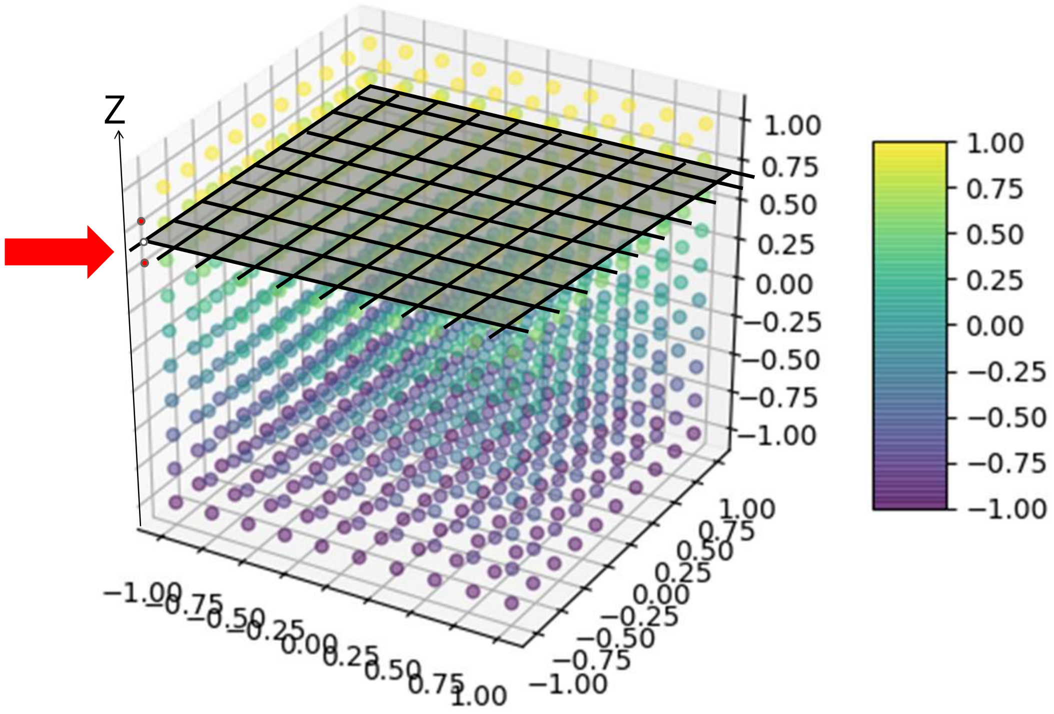


Acht Eckpunkte auf einem Würfel, die den Interpolationspunkt C umgeben.

- Zunächst werden die Wert für C_1 und C_0 mittels Bilinearer Interpolation ermittelt.
- Danach werden die Werte von C_1 und C_0 für C linear interpoliert.

Bilder aus: https://en.wikipedia.org/wiki/Trilinear_interpolation

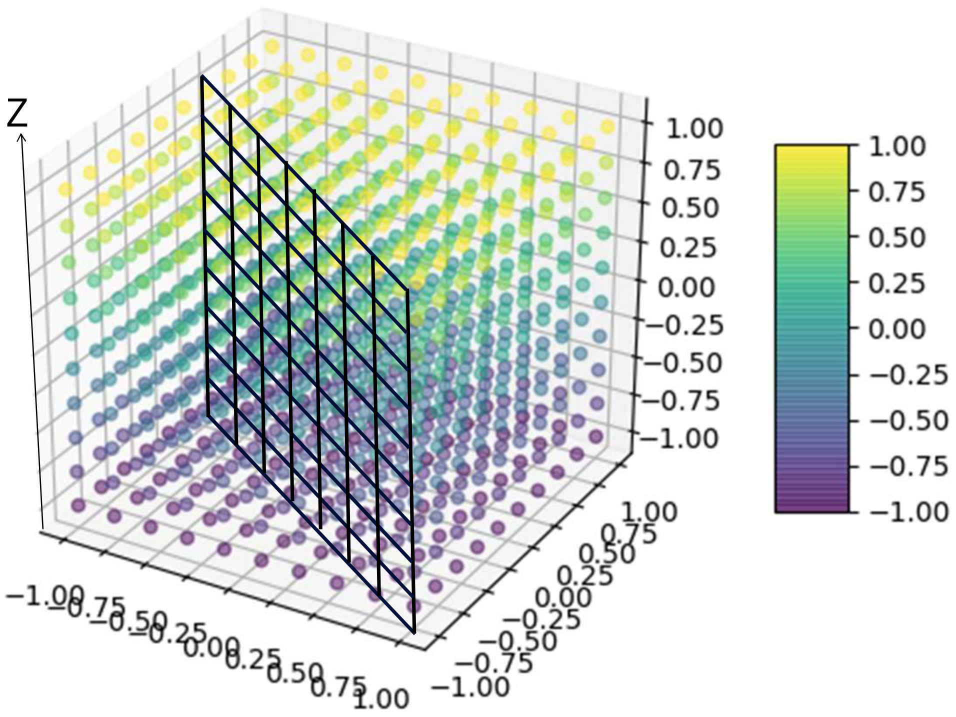
Bild aus https://stackoverflow.com/questions/22715746/plot-3d-cartesian-grid-with-python



Einfache lineare Interpolation

- Horizontale Schnitte, die zwischen zwei Gitter-schichten liegt und die gleiche Auflösung, wie das 3D-Gitter haben, werden linear berechnet.
- Dabei werden die Höhen-werte linear interpoliert.
- Hat man die Gitterpunkte auf der Schnittebene berechnet, kann man die Isolinien oder Isoflächen berechnen.

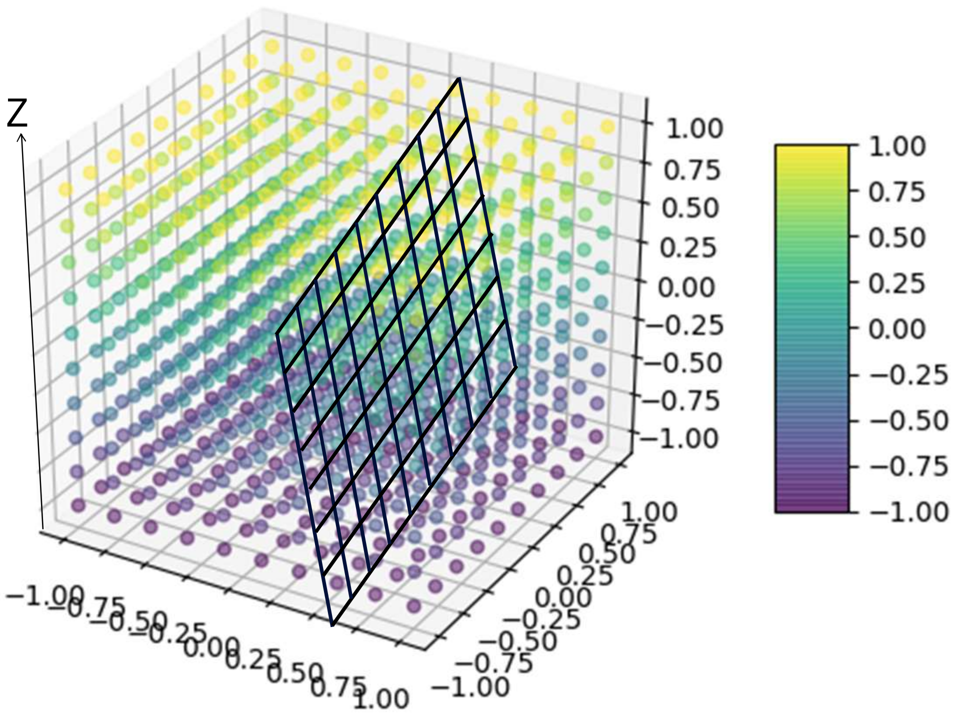
Bild aus https://stackoverflow.com/questions/22715746/plot-3d-cartesian-grid-with-python



Bilineare-Interpolation

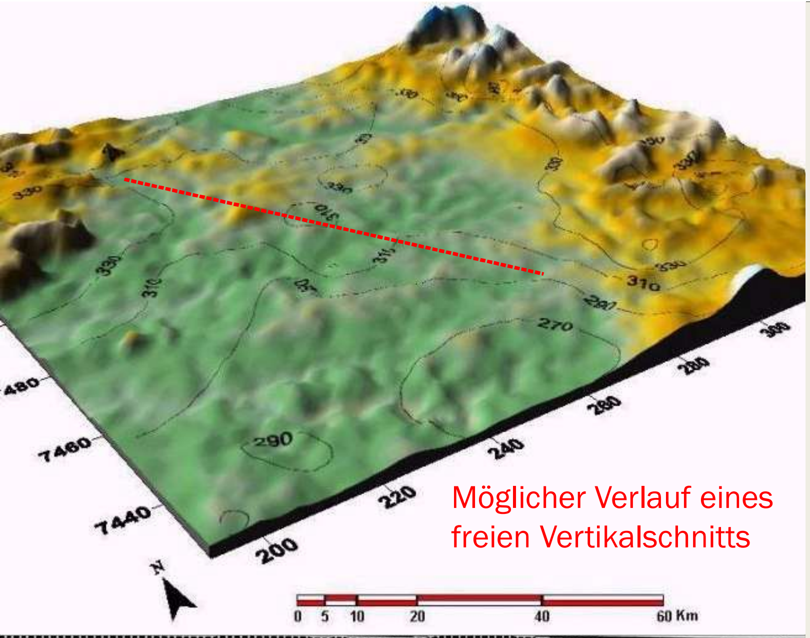
- Eine Bilineare Interpolation kann genutzt werden, wenn die Höhenwerte des Schnitts, den Höhenwerten des Datengitters entsprechen.
- Mit Bilinearer-Interpolation werden die Punkte, die den Verlauf des frei gewählten vertikalen Schnitts durch das 3D-Datenvolumen angeben, berechnen.

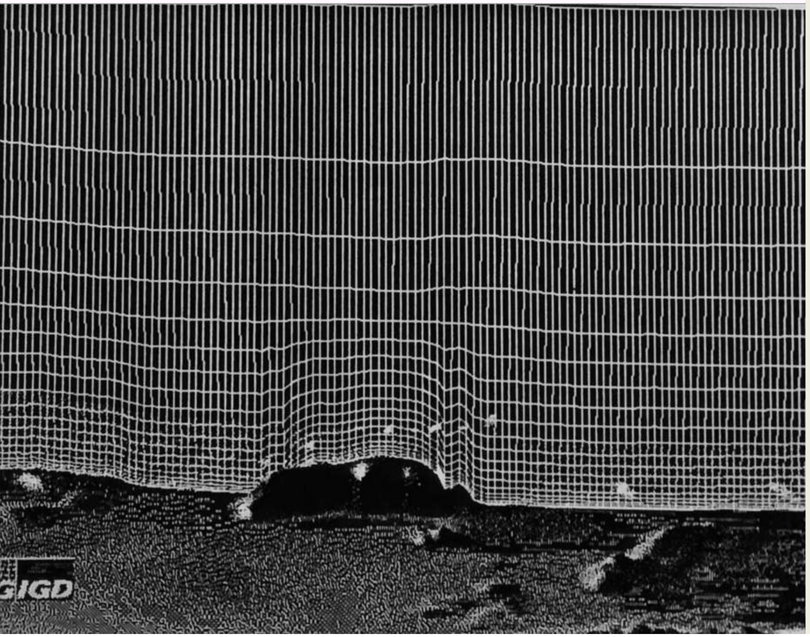
Bild aus https://stackoverflow.com/questions/22715746/plot-3d-cartesian-grid-with-python



Trilineare-Interpolation

- Liegt der Schnitt „irgendwie“ im Raum, braucht man zur Berechnung der einzelnen Gitterpunkte der Schnittfläche die Trilineare Interpolation





Trilineare-Interpolation

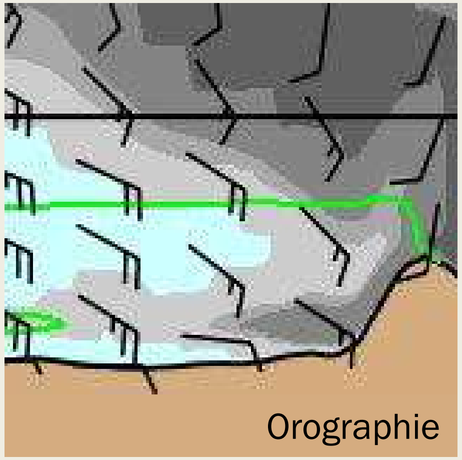
Noch ein Beispiel für Trilineare:

- Interpolation eines Schnittes durch ein Gitter mit nicht regelmäßigen Höhenwerten.
- Während die xy-Position der Schnittebene mit Bilinearer Interpolation berechnet werden, müssen in diesem Fall die Gitterwerte, wegen des unregelmäßigen Verlaufs der Orographie mit trilinearer Interpolation berechnet werden.

Bild oben: Bild links : von Daniel Candido aus Econt von pt.wikipedia auf Commons übertragen.

Bild unten: Diplomarbeit Hermin Aftahi, „Visualisierung meteorologischer Daten“, 1994

Freie Vertikalschnitte nutzt man beispielsweise, um die Wetterinformation entlang einer Flugroute zu berechnen

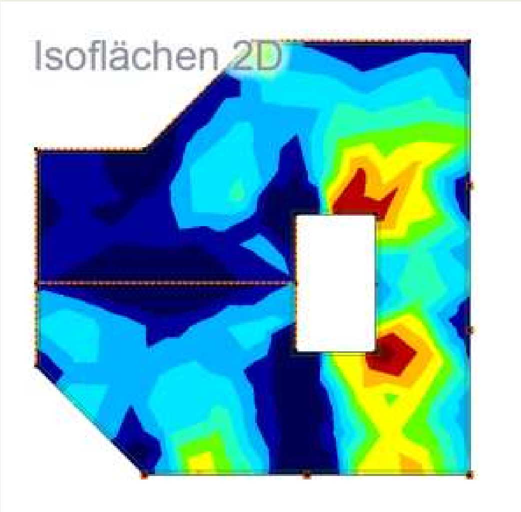


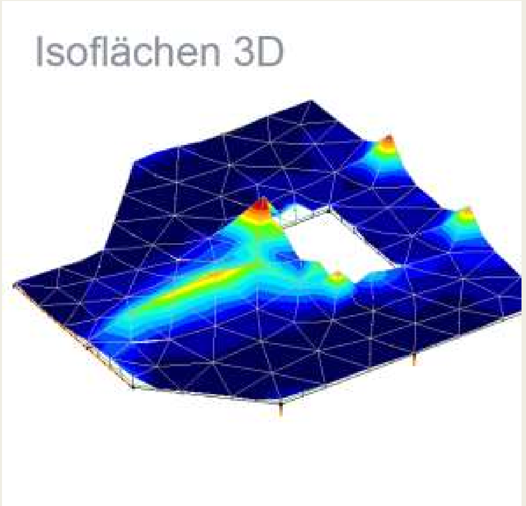
Schnittebene zeigt das Flugwetter von Frankfurt nach Münster.

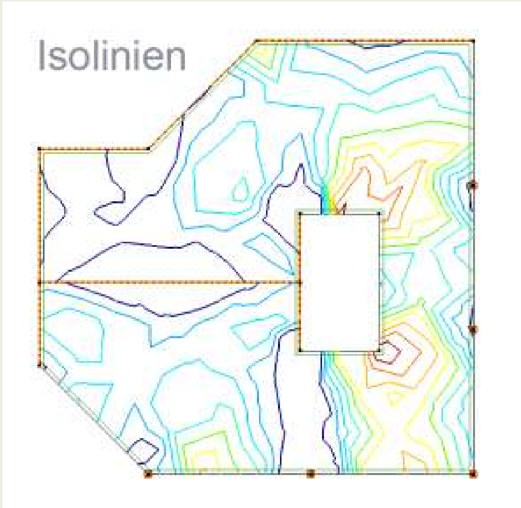
Quelle: https://www.dwd.de/DE/fachnutzer/luftfahrt/download/produkte/cross_section/cross_section_download.html;jsessionid=EA7CB57ACD903A417141B1872A7025F6.live11042?nn=371800

Wir kennen jetzt:

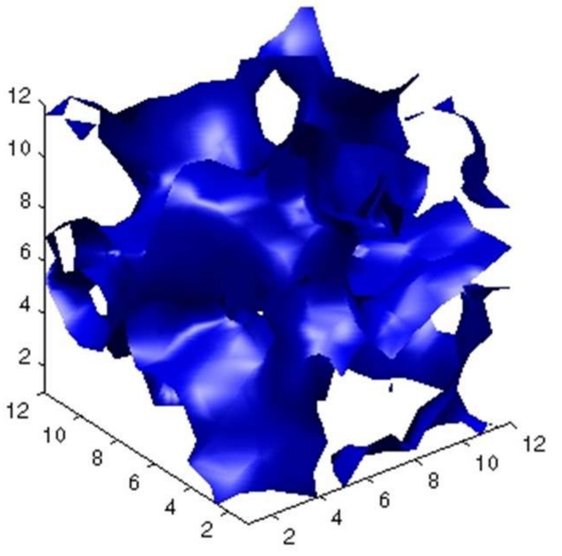
Isolinien und Isoflächen auf 2D- und 3D-Ebenen

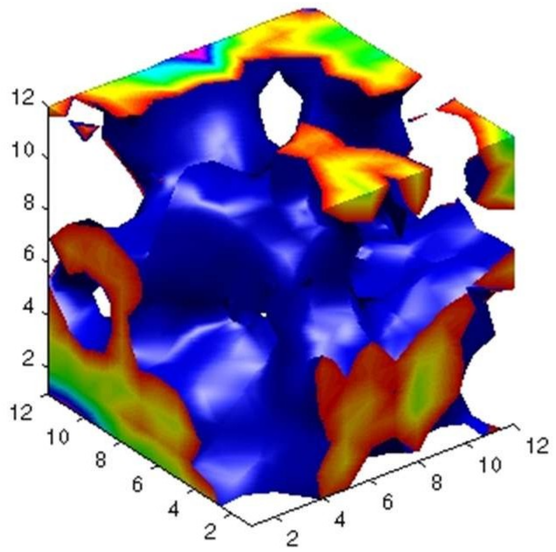






Bilder aus: https://www.ingware.ch/files/m3_10_preview.pdf

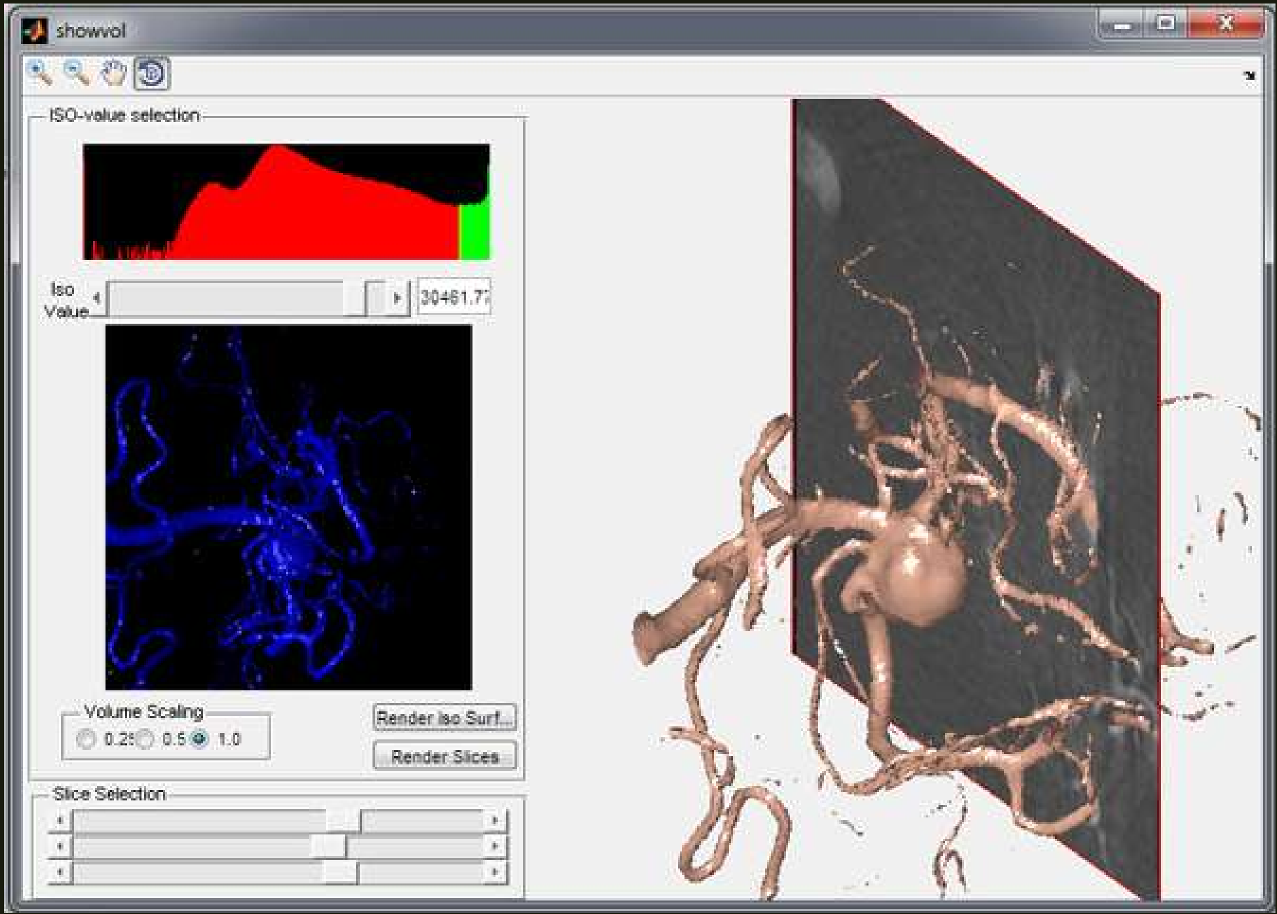




3D-Isoflächen im 3D-Datenvolumen berechnen

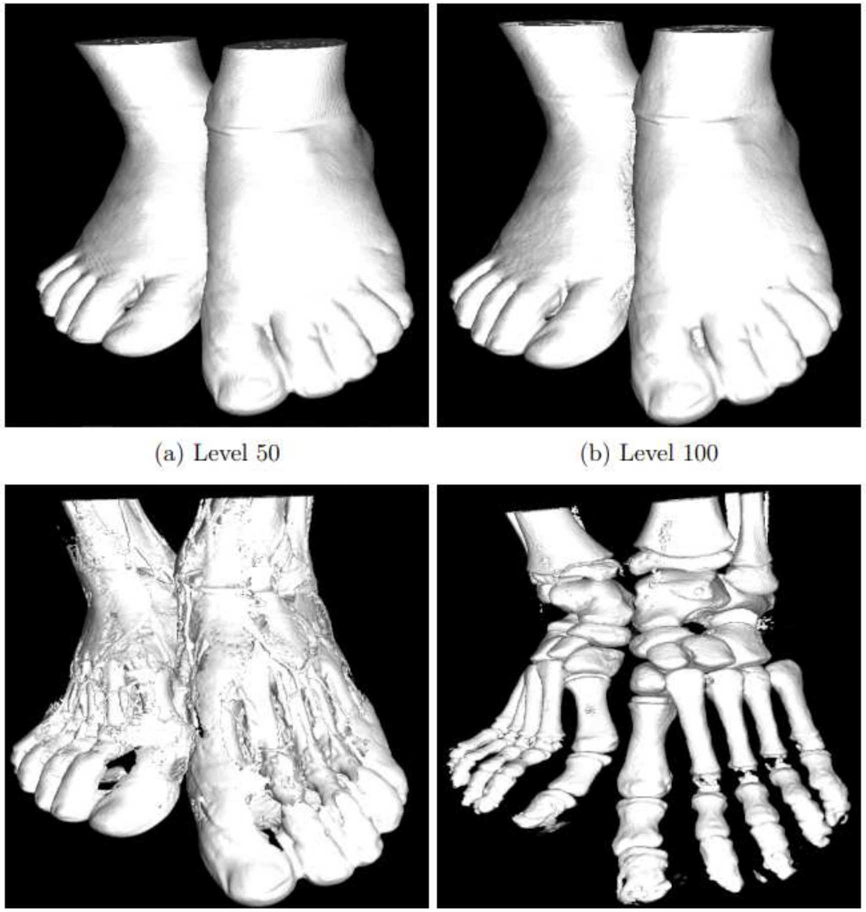
- Oben: 3D-Isofläche im 3D-Volumen
- Unten: 3D-Isoflächen auch auf den Schnittebenen

Bilder aus: https://de.mathworks.com/help/matlab/visualize/isocaps-add-context-to-visualizations.html



3D-Isoflächen

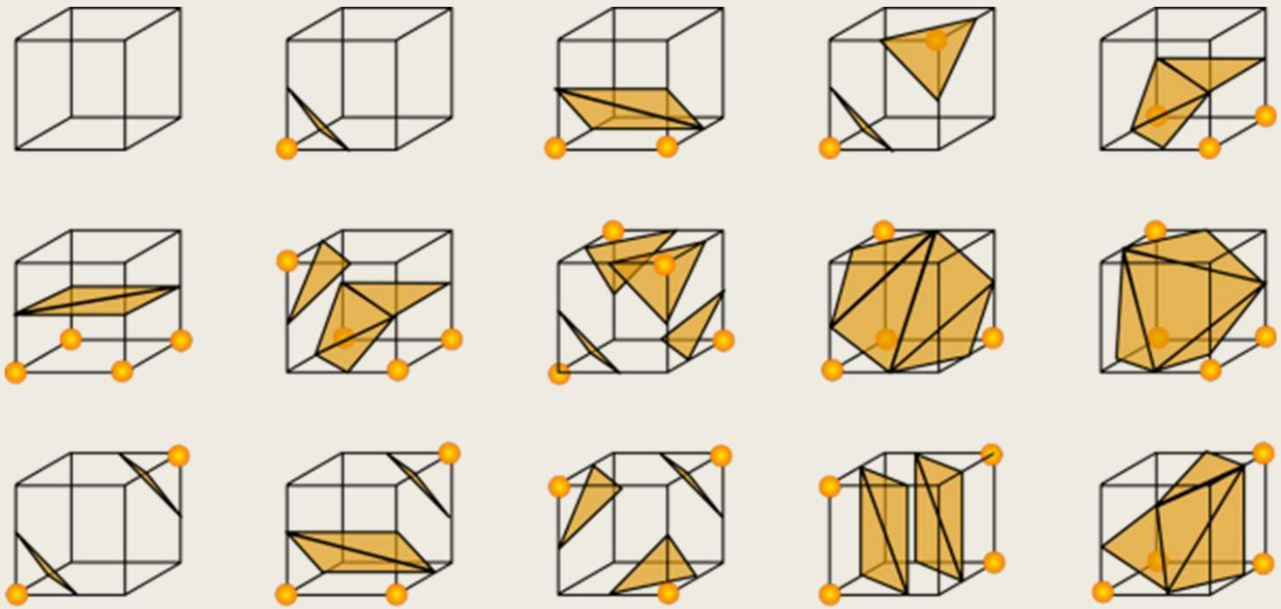
- Visualisierung von CT-Daten mittels von Isofflächen
- Beispielsweise Adern
- Bild aus. https://de.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/25987/versions/2/screenshot.png


(a) Level 50
(b) Level 100
(c) Level 150
(d) Level 235

3D-Isoflächen

- Extraktion unterschiedlicher Dichte-Werte

Bild aus: Bachelorarbeit „Oberflächenextraktion mittels des Marching Cubes Algorithmus “ Georg Seibt, https://www.fim.uni-passau.de/fileadmin/dokumente/fakultaeten/fim/lehrstuhl/sauer/geyer/BA_MA_Arbeiten/BA-SeibtGeorg-201410.pdf


Quelle: http://de.wikipedia.org/wiki/Marching_Cubes

Marching Cube zur Erzeugung von 3D-Isoflächen

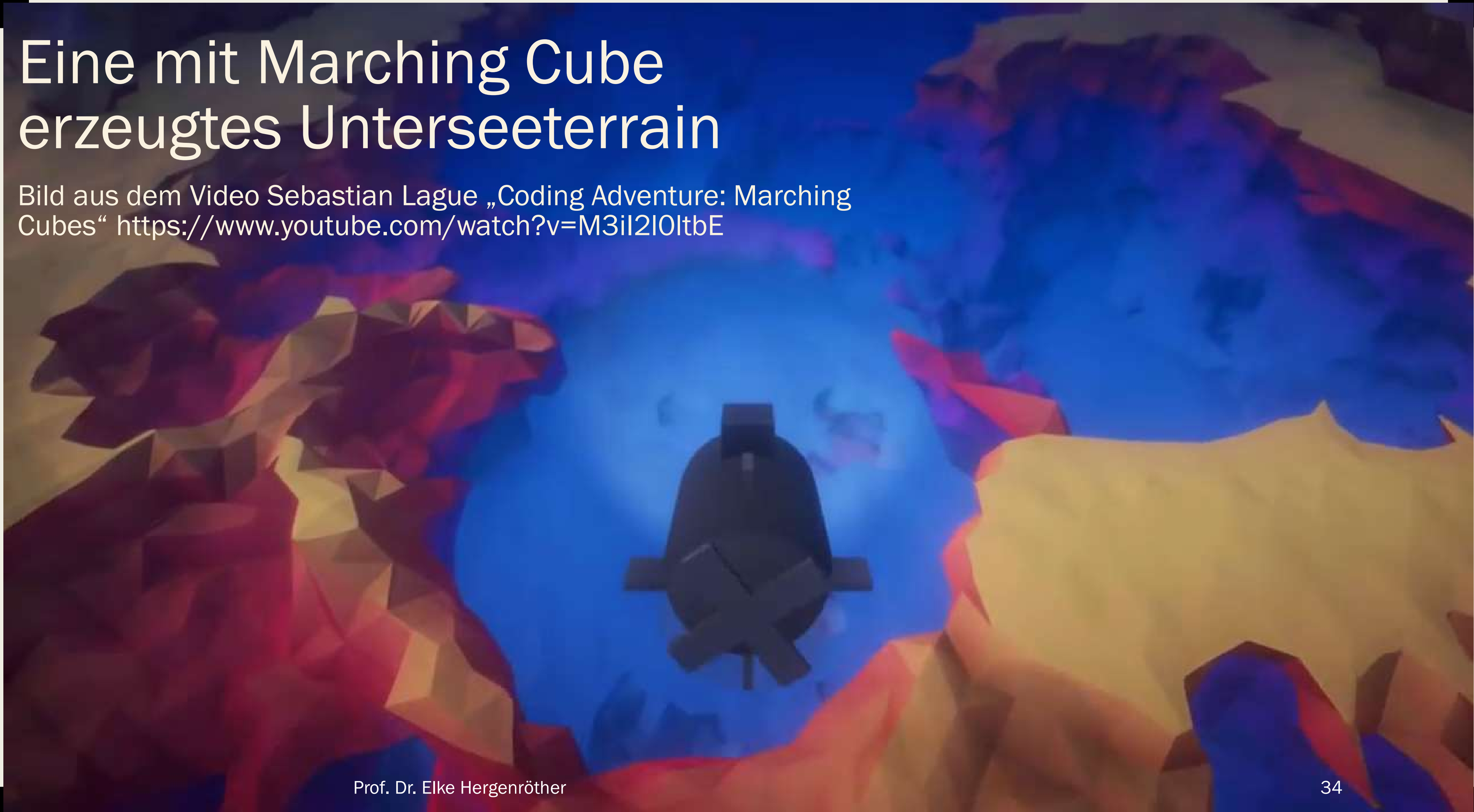
- Funktioniert wie der Marching Squares-Algorithmus im 3D-Raum
- Auch hier gibt es uneindeutige Situationen
- Es gibt eine Reihe von Paper, wie man diese uneindeutigen Fälle auflösen kann
- Beispielsweise: Gregory M. Nielson. »On marching cubes«. In: Visualization and Computer Graphics, IEEE Transactions on 9.3 (2003), S. 283–297.

Eine mit Marching Cube erzeugte Geländefläche

Bild aus dem Video Sebastian Lague „Coding Adventure: Marching Cubes“ https://www.youtube.com/watch?v=M3iI2l0ltbE

Eine mit Marching Cube erzeugtes Unterseeterrain

Bild aus dem Video Sebastian Lague „Coding Adventure: Marching Cubes“ https://www.youtube.com/watch?v=M3iI2l0ltbE



3D-Isofläche, die Geländepunkte verbindet: generiert mit Marching Cubes

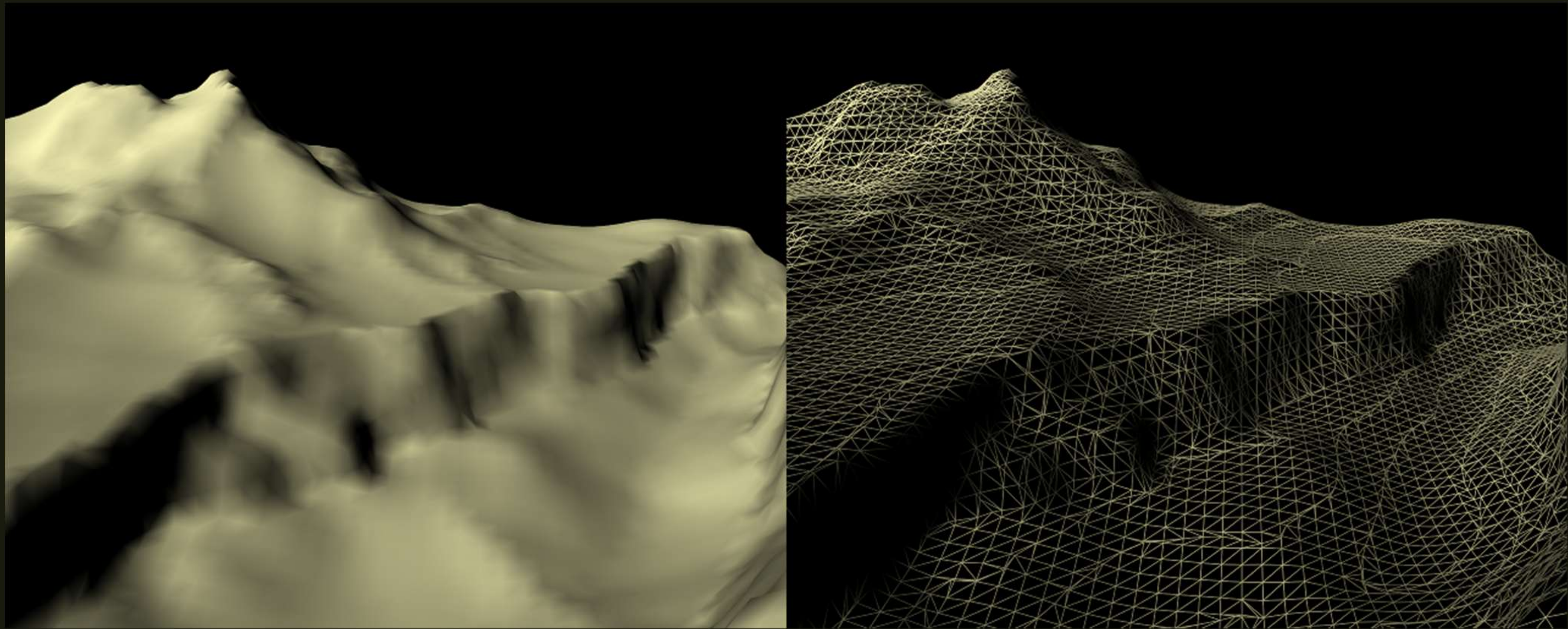
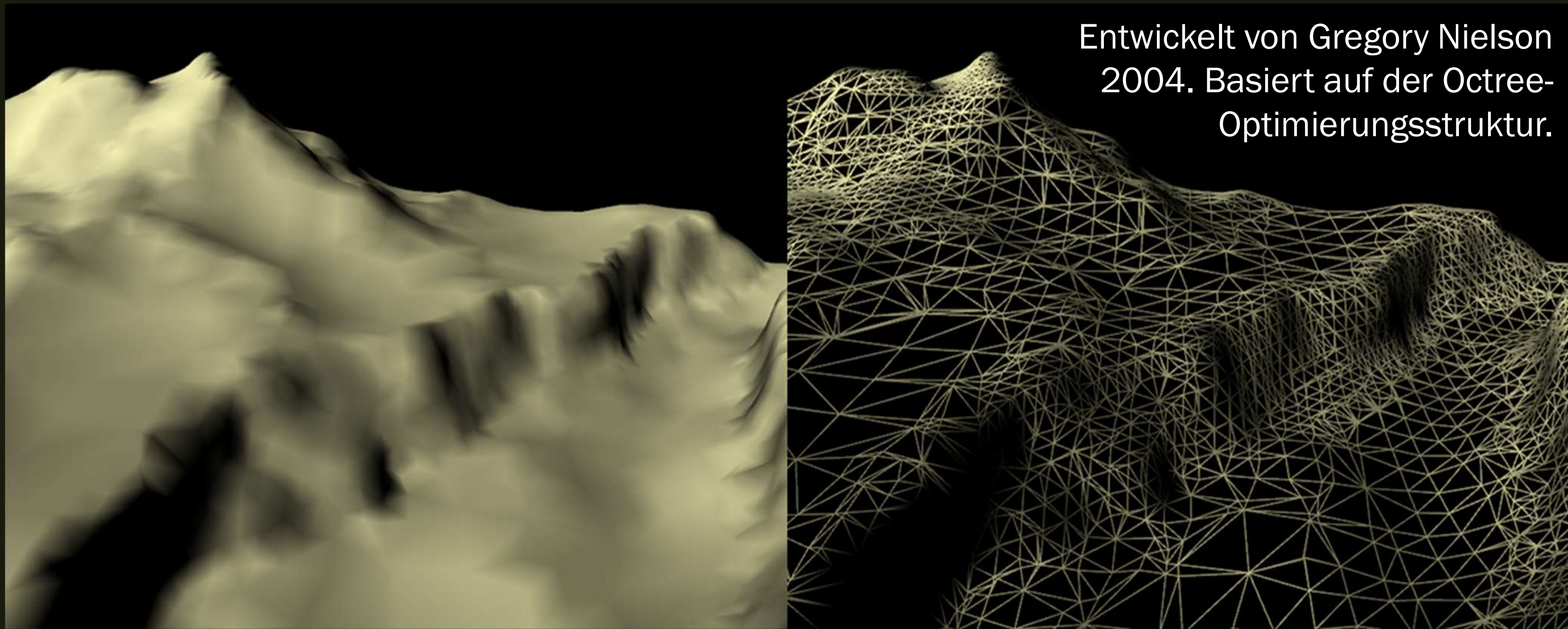


Bild aus: https://www.volume-gfx.com/wp-content/uploads/2013/02/mcTerrain.png

3D-Isofläche, die Geländepunkte verbindet: generiert mit Dual Marching Cubes

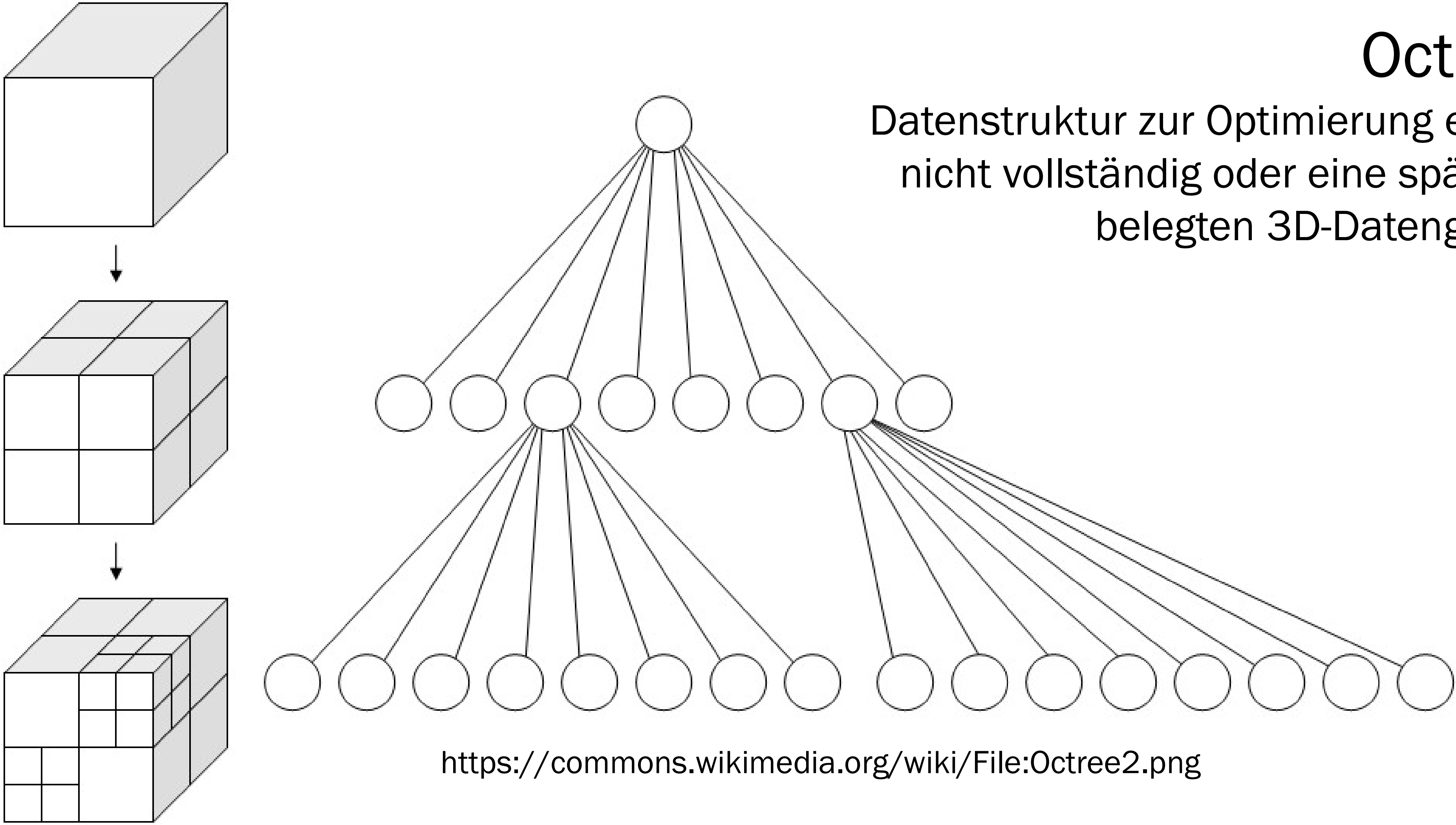


Entwickelt von Gregory Nielson 2004. Basiert auf der Octree-Optimierungsstruktur.

Bild aus: https://www.volume-gfx.com/wp-content/uploads/2013/02/mcTerrain.png

Octree

Datenstruktur zur Optimierung eines nicht vollständig oder eine spärlich belegten 3D-Datengittes



https://commons.wikimedia.org/wiki/File:Octree2.png

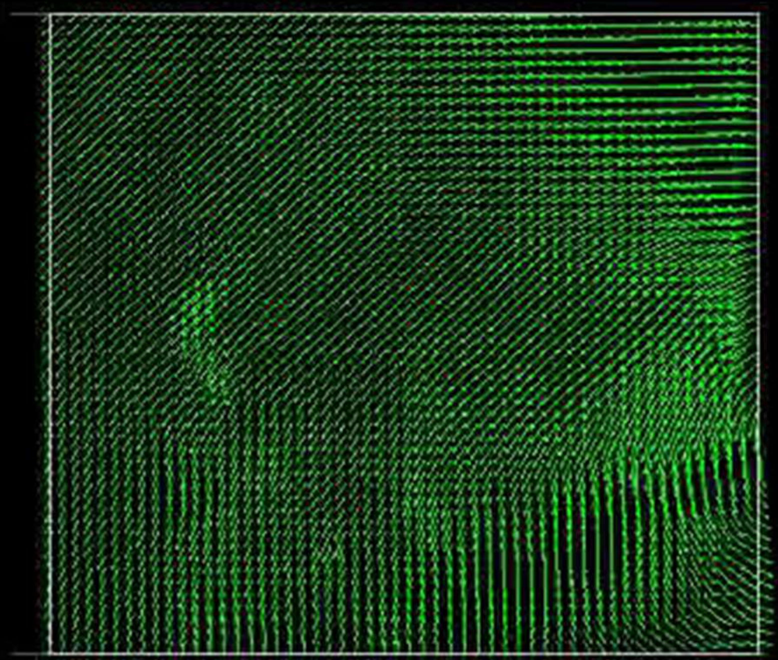
Es gibt noch mehr Techniken neben den Isoflächen und –linien, um Simulations- oder Messwerte darzustellen:

Beispielsweise:

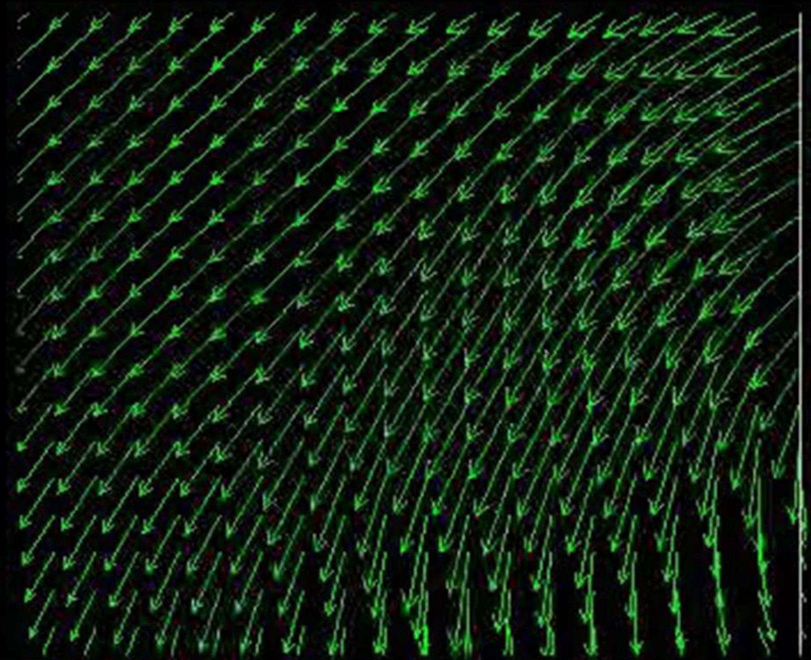
Pfeile und Linien, die Isolinien- und Flächen gegebenenfalls ergänzen können.

Dynamische Daten

- Bisher nur statische Daten einer Momentaufnahme visualisiert.
- Die Visualisierung von dynamischen Daten erfolgt am besten in einer Animation.
- Aber auch 2D-Schnitte (Slices) kann man so visualisieren, dass sie Bewegung anzeigen.



Visualisierung mehrere Zeitschritte mit normierten Pfeilen

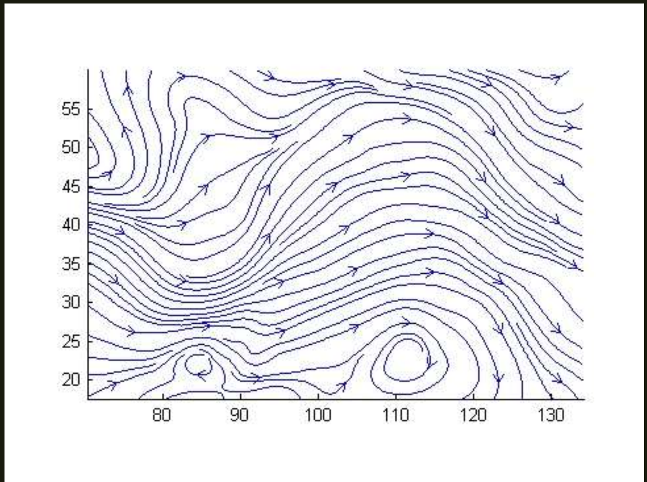


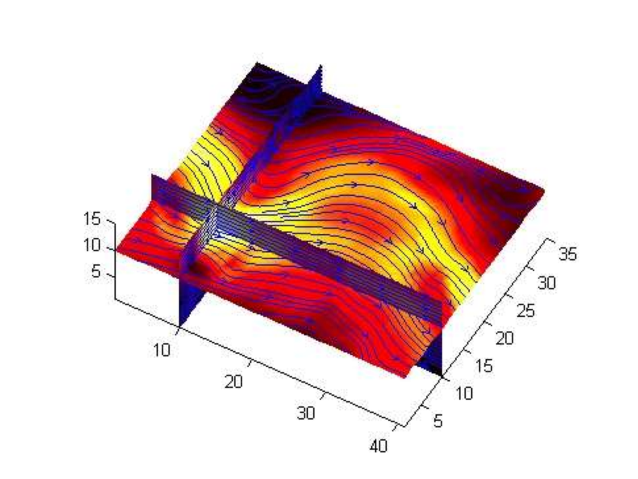
Visualisierung mehrere Zeitschritte mit nicht normierten Pfeilen

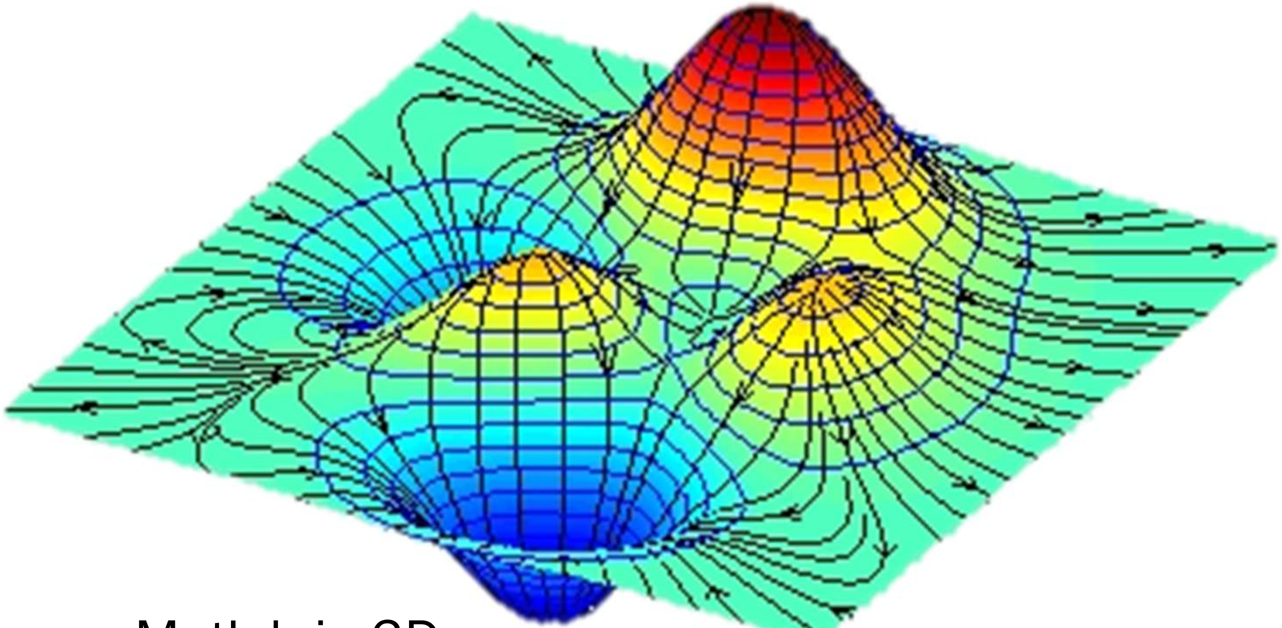
WEITERE, DYNAMIK ZEIGENDE VISUALISIERUNGSTECHNIKEN AUF SCHNITTEBENEN (SLICES)

Streamlines aus Matlab

Bilder aus:
http://matlab.izmiran.ru/help/techdoc/ref/streamslice.html

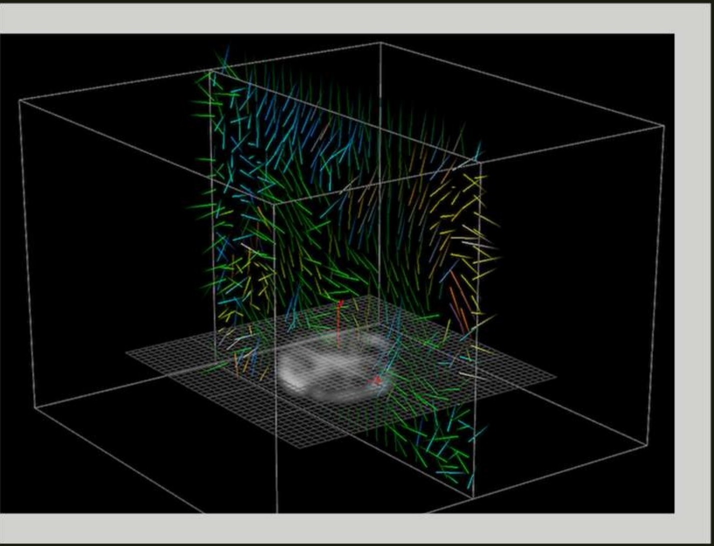


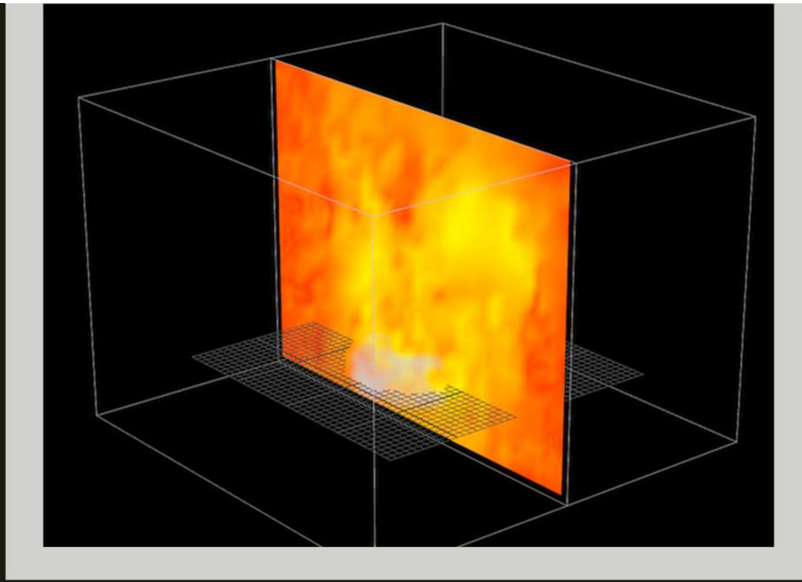


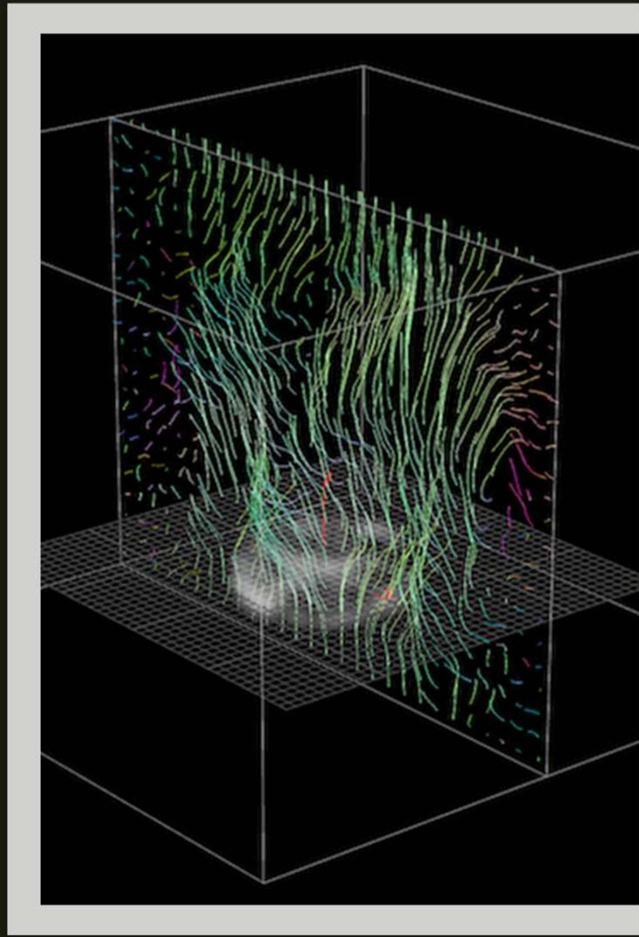


Streamlines aus Matlab in 3D

Bilder aus:
http://matlab.izmiran.ru/help/techdoc/ref/streamslice.html







Weitere Visualisierungstechniken auf Schnittebenen, um Strömung darzustellen:

Links: Länge und Form zeigt Geschwindigkeit und Verlauf an

Mitte: Farbe symbolisiert Geschwindigkeit

Rechts: Partikelspur, die den Verlauf anzeigt

Alle Bilder von Houdini aus:
https://www.sidefx.com/docs/houdini/visualizers/volume.html

ZUSAMMENFASSUNG: VISUALISIERUNG TEIL 2

Behandelte Themen:

- Isolinien und Isoflächen in 2D berechnen
- Isoflächen in 3D berechnen

Nächste Vorlesung

- 3D-Wahrnehmung erzeugen
- Relfexionsberechnung