

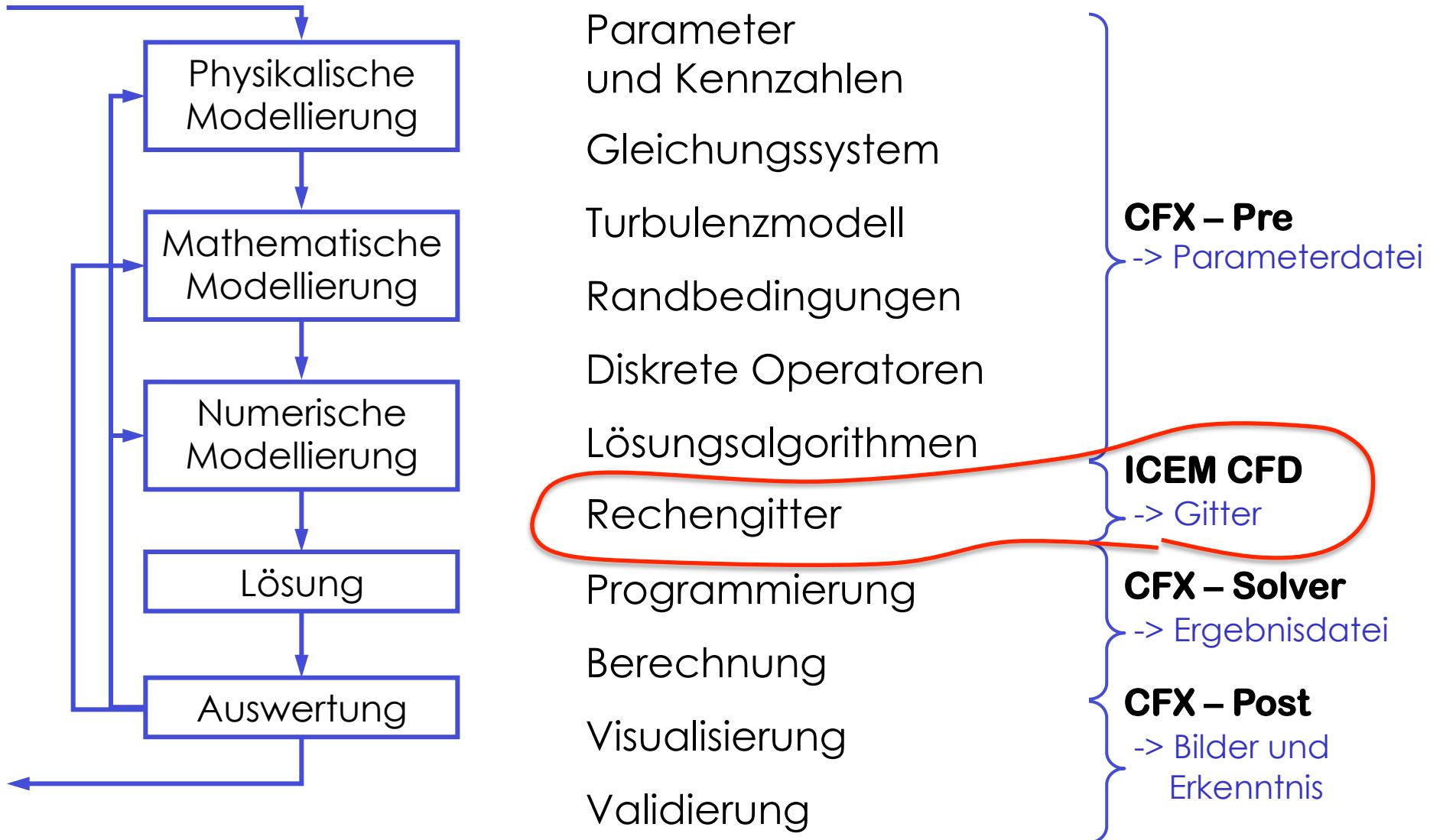
Angewandte Strömungssimulation

3. Vorlesung

Stefan Hickel

Gittergenerierung

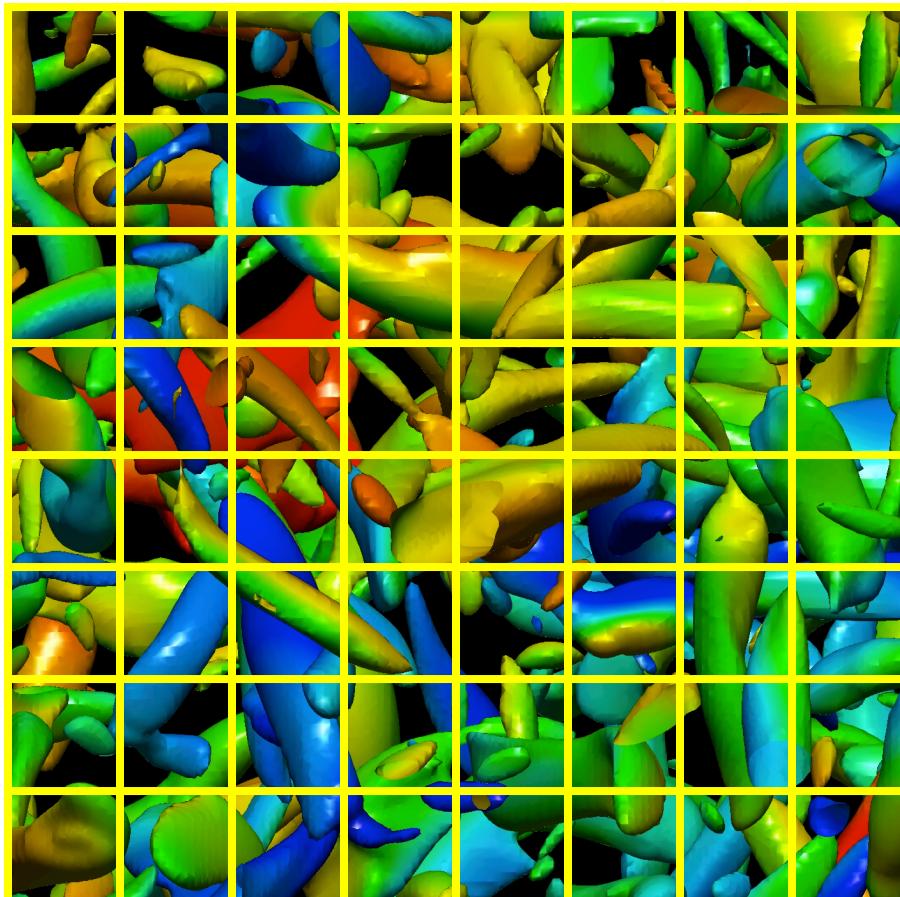
Numerische Strömungsberechnung



Diskretisierung

- Numerische Methoden erfordern eine diskrete Darstellung der Lösung. Hierzu wird das Rechengebiet üblicherweise in Zellen zerlegt.
- Die üblichsten Methoden betrachten die Lösung als
 - **Zellmittelwerte**
-> Finite-Volumen Diskretisierung (robust)
 - **Punktwerte** am Zellmittelpunkt oder an den Ecken
-> Finite-Differenzen Diskretisierung
 - **Koeffizienten** von Ansatzfunktionen
-> Finite-Elemente Diskretisierung (unüblich für CFD)
- **CFX basiert auf der Finite-Volumen Methode**

Finite-Volumen Methode

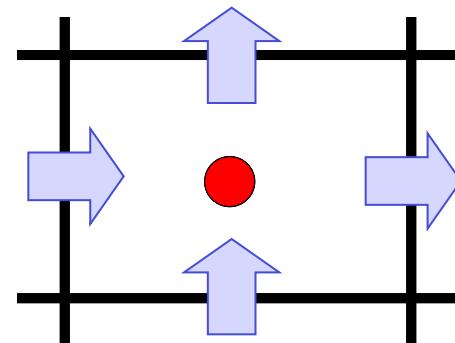


- Das Rechengebiet wird in nicht überlappendende Kontrollvolumina - finite Volumen (FV) - unterteilt.
- Für jedes FV wird nur ein Mittelwert betrachtet.
- Die Erhaltungsgleichungen für die FV einzeln ausgewertet.
- Volumenintegrale können mit dem Satz von Gauss in Oberflächenintegrale umgewandelt werden.

Finite-Volumen Methode

- Dazu werden **Flüsse** → über die FV-Oberfläche S bilanziert!

● Zellknoten
□ Finites Volumen



- Jedem Kontrollvolumen wird ein Knoten zugeordnet, an welchem die diskreten (Mittel-) Werte gespeichert werden.

Finite-Volumen Methode

- Die Erhaltungsgleichungen werden über die KV integriert

$$\frac{\partial}{\partial t} \varphi = \nabla \cdot \Psi \quad \rightarrow \quad \frac{\partial}{\partial t} \int_{\bar{V}} \varphi dV = \int_{\bar{V}} \nabla \cdot \Psi dV$$

- Volumenintegrale können mit dem Satz von Gauss in Oberflächenintegrale umgewandelt werden.

$$\int_{\bar{V}} \nabla \cdot \Psi dV = \int_{S_V} \mathbf{n} \cdot \Psi dS = \int_{S_{rechts}} \Psi dS - \int_{S_{links}} \Psi dS + \dots$$

- Beispiel: Impulserhaltungsgleichung in integraler Form

$$\frac{\partial}{\partial t} \int_{\bar{V}} \rho \underline{u} dV + \int_{S_V} \rho \underline{u} \underline{u} \cdot \underline{n} dS = - \int_{S_V} p dS + \int_{S_V} \underline{\underline{\tau}} \cdot \underline{n} dS + \int_{\bar{V}} \rho \underline{F}_V dV$$

Finite-Volumen Methode

- Mehrere **Approximationsvorschriften** werden zur numerischen Auswertung der Flüsse durch die Grenzflächen der Kontrollvolumina benötigt:
 - Zur Approximation der Feldgrößen an anderen Punkten als den Knotenpunkten (wo ja die Werte gespeichert sind) werden **Interpolation**vorschriften verwendet.
 - Die Flächenintegrale werden durch numerische **Quadratur** angenähert.
- Die diskretisierten Erhaltungsgleichungen werden für jedes KV ausgewertet.
- Zusammen mit geeigneten Randbedingungen entsteht ein algebraisches Gleichungssystem, welches numerisch gelöst werden kann.

Gittergenerierung

Wie kann man diese Kontrollvolumina (meist Zellen genannt) wählen?

- Damit beschäftigt sich die Gittergenerierung!

Wichtige Definitionen:

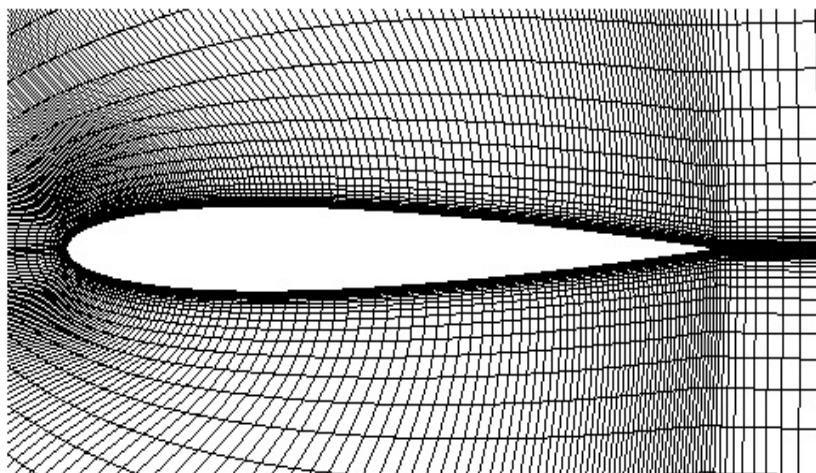
Topologie: Beziehungen der Nachbarschaft, der Überschneidung und ähnlichem.

Geometrie: Form und Maße.

Topologie

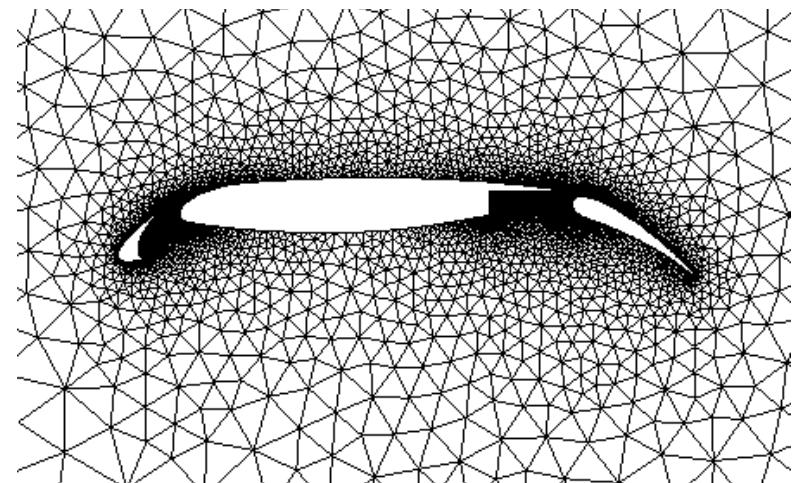
Strukturierte Gitter

- innenliegende Elemente sind topologisch ähnlich
- die Verbindung zum lokalen Nachbarn ist die selbe für alle inneren Punkte



Unstrukturierte Gitter

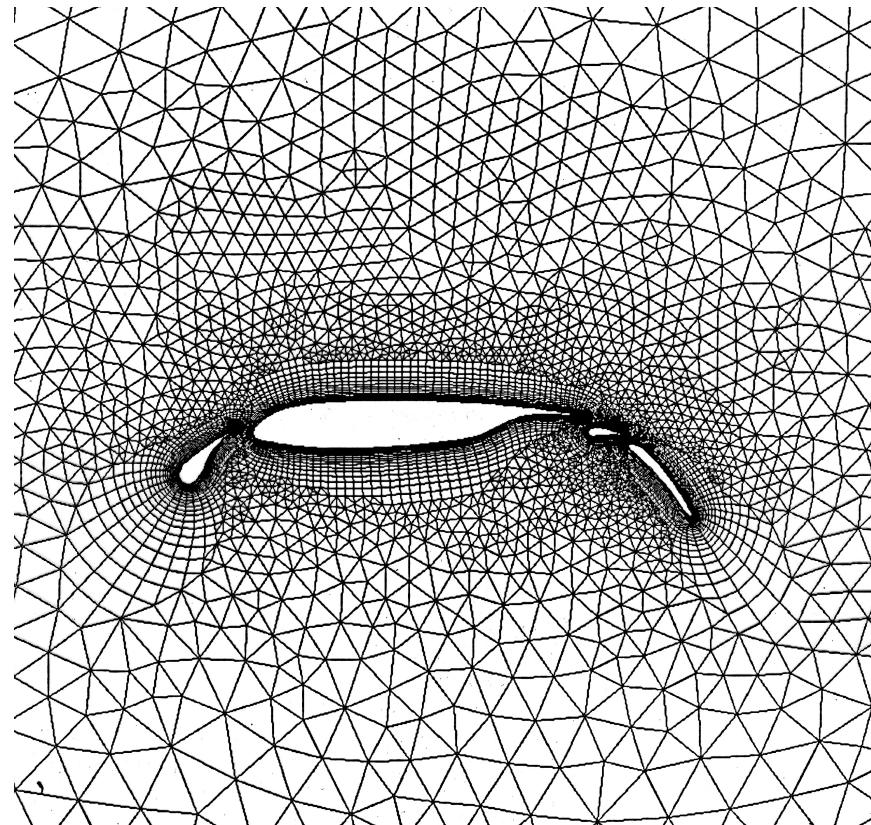
- Elemente haben Nachbarn beliebiger Struktur
- die Verbindung zum lokalen Nachbarn ist im Allgemeinen unterschiedlich



Topologie

Hybride Gitter

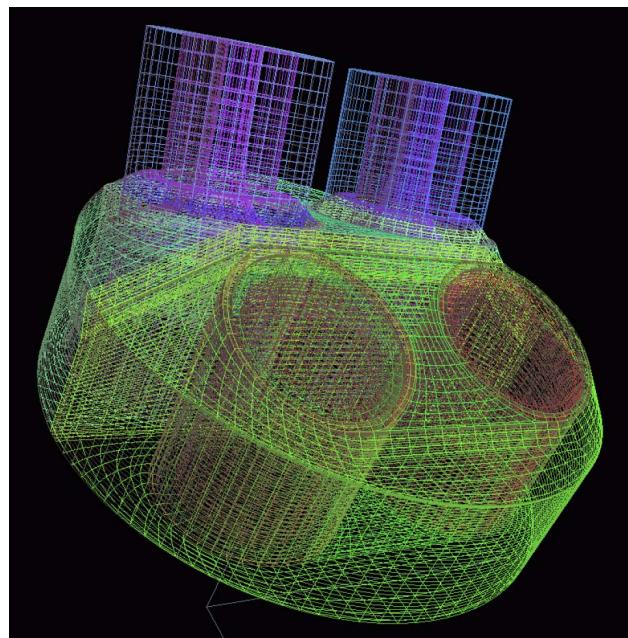
- Eine Mischung von strukturierten und unstrukturierten Gittern
- Implementierung erfolgt dann meist unstrukturiert oder als Multiblock Gitter



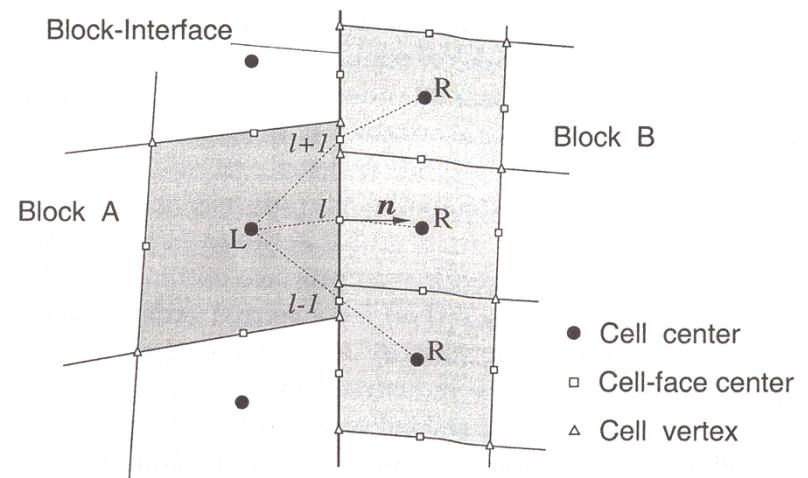
Topologie

Multi-Block Gitter

- Das gesamte Gebiet wird mehrfach unterteilt
- Die Teilbereiche werden entweder strukturiert oder unstrukturiert vernetzt
- auch komplexe Geometrien können hierdurch mit strukturierten Gittern modelliert werden



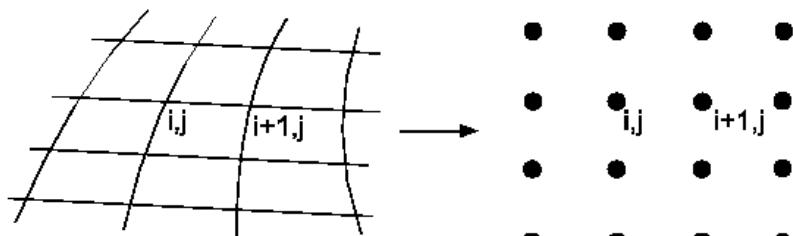
Problem der Definition von
Übergangsbedingungen an Blockgrenzen



Topologie

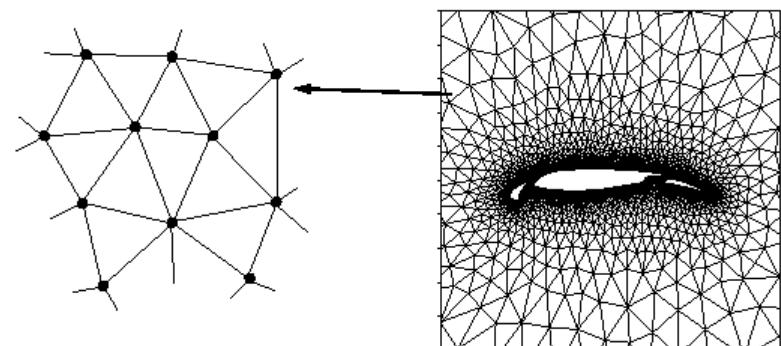
Strukturierte Gitter

- (unter-) Gebiete können auf kartesische Gebiete abgebildet werden
- hauptsächlich Vierecke (2-D) Hexaeder (3-D)



Unstrukturierte Gitter

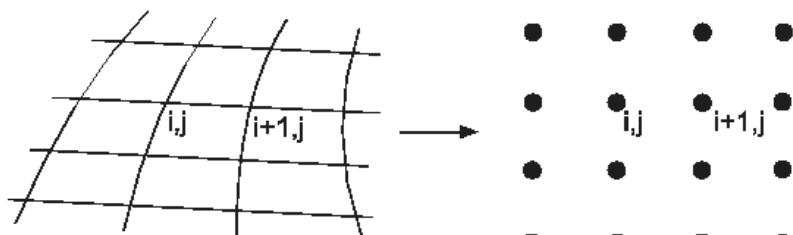
- (unter-) Gebiete können i.A. nicht auf kartesische Gebiete abgebildet werden
- hauptsächlich Dreiecke (2-D), Tetraeder (3-D)



Topologie

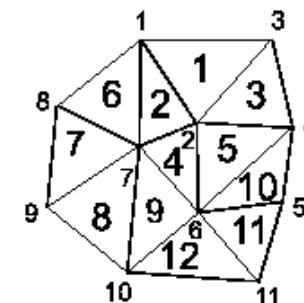
Strukturierte Gitter

- im Normalfall einfache Datenstruktur
(z.B. 2-D/3-D Matrizen)



Unstrukturierte Gitter

- im Normalfall kompliziertere Datenstruktur
- Connectivity-Matrix** muss mitgeführt werden



"Connectivity Matrix"

Element	Knoten		
1	1	2	3
2	1	7	2
3	2	4	3
4	2	7	6
5	6	4	2
6	7	1	8
7	7	8	9
:	.	.	.

Topologie

Strukturierte Gitter

- erlaubt Gittergenerierung "von Hand" -> optimale Ergebniskontrolle.
- einfache Topologie und schnelle Implementierung
- effiziente Algorithmen anwendbar
- Gitterverfeinerung aufwendig

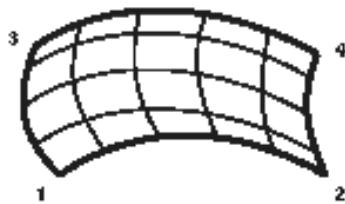
Unstrukturierte Gitter

- komplexe Geometrien können sehr leicht modelliert werden
- einfache Gitterverfeinerung durch lokales Hinzufügen von Punkten im Bereich starker Gradienten
- keine globale Struktur
- ständiges Zugreifen auf die Connectivity-Matrix
- höhere Rechenzeiten

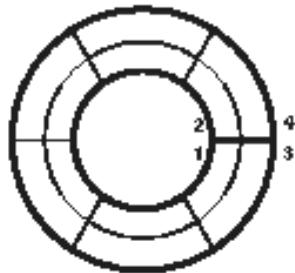
Strukturierte Gitter

Man unterscheidet drei grundlegende Gittertypen (Topologien)

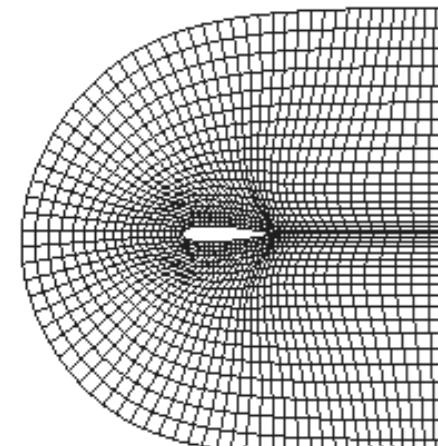
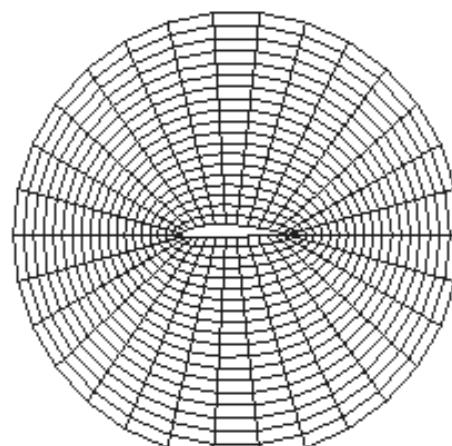
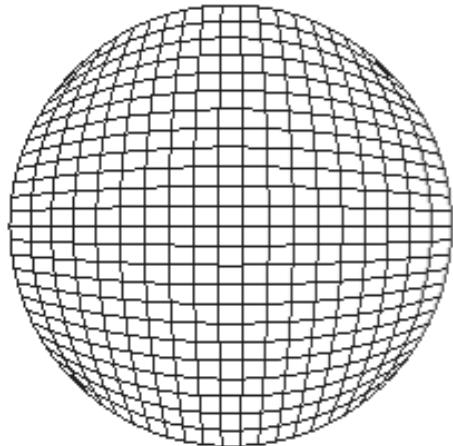
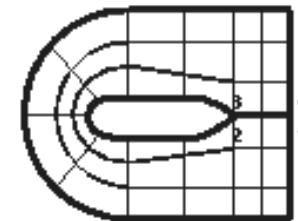
H-Gitter



O-Gitter



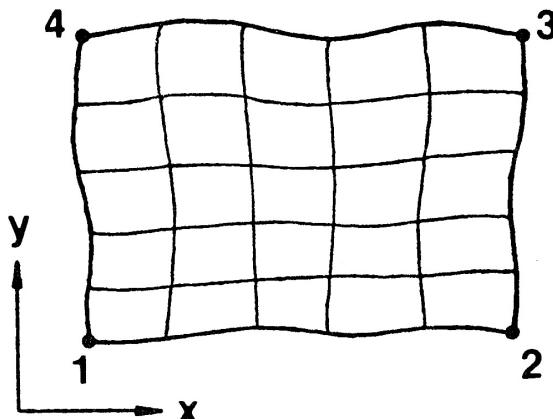
C-Gitter



Strukturierte Gitter

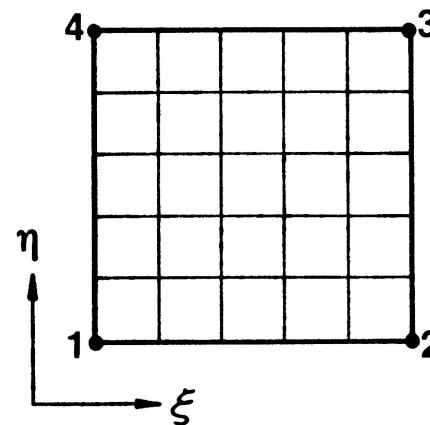
- Raumpunkte werden in direkter Weise auf die Elemente einer Matrix abgebildet.
- Transformation der Gleichungen aus dem physikalischen Raum in den numerischen Raum.

physikalischer Raum



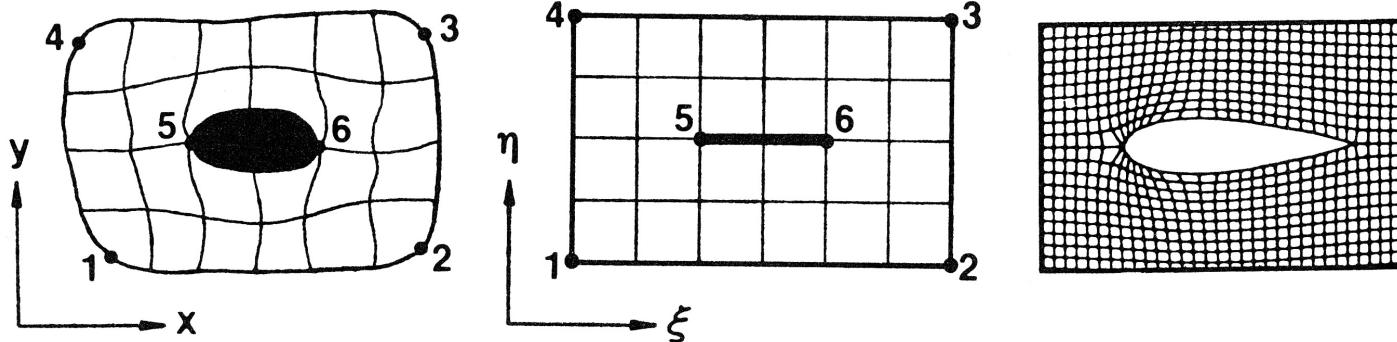
numerischer Raum

$$x = x(\xi, \eta)$$
$$y = y(\xi, \eta)$$



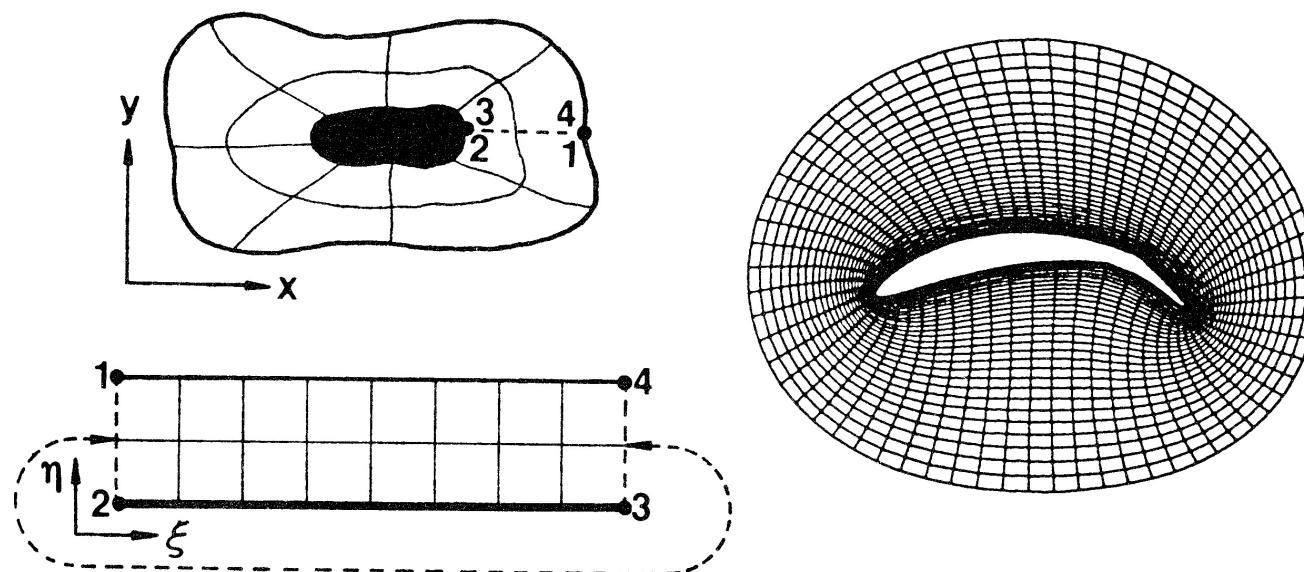
Strukturierte Gitter

H-Gitter



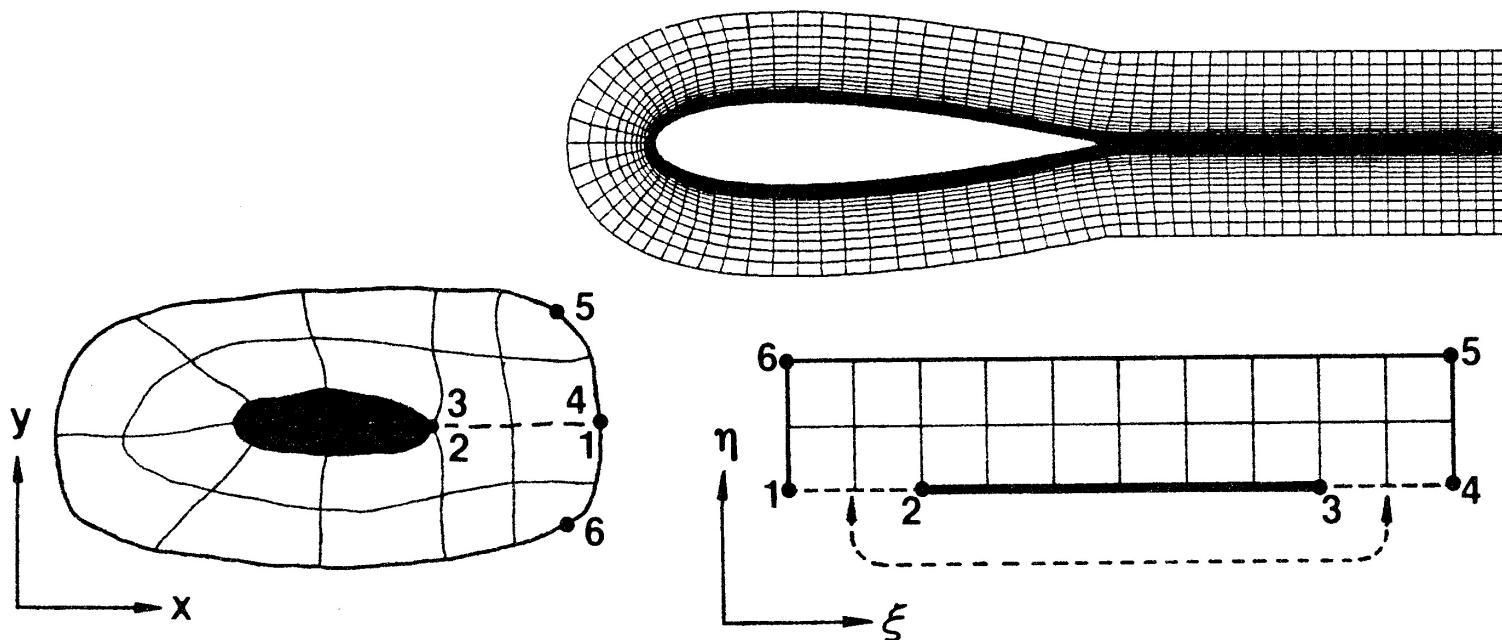
Strukturierte Gitter

O-Gitter



Strukturierte Gitter

C-Gitter



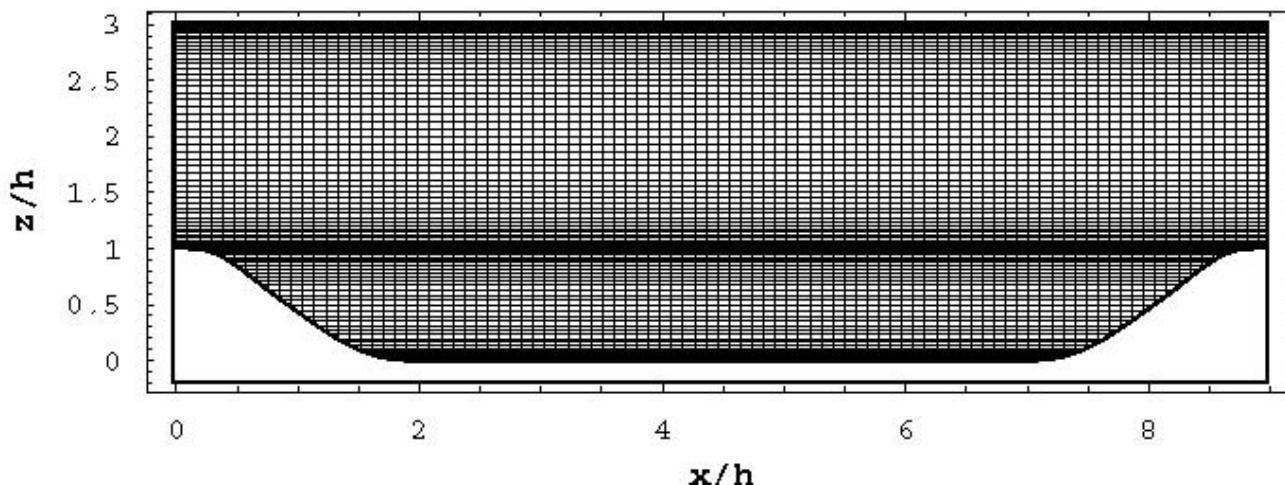
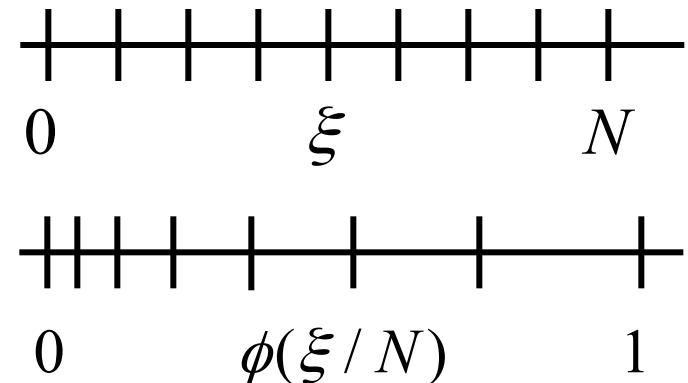
Anforderungen an Gitter

- Das generierte Gitter darf keine überschneidenden Gitterlinien besitzen, Zellen müssen positives Volumen haben.
- Die Möglichkeit, Gitterpunkte und Gitterlinien in beliebig wählbaren Bereichen zu verdichten, sollte vorhanden sein.
- Lage der Punkte auf der Berandung sollte vorgebbar sein, um eine genaue Wiedergabe der Geometrie zu gewährleisten.
- Gitter soll glatt sein.
- Möglichst orthogonales Gitter.
- Kurze Rechenzeiten (wenig Zellen, schnelle Konvergenz, usw.)

Erzeugung strukturierter Gitter

Algebraische Verdichtungsfunktionen

- Definiert auf Blockgrenzen
- Regeln für Punktverteilung
 - geometrische Streckung
 - exponentielle Streckung
 - hyperbolische Streckung

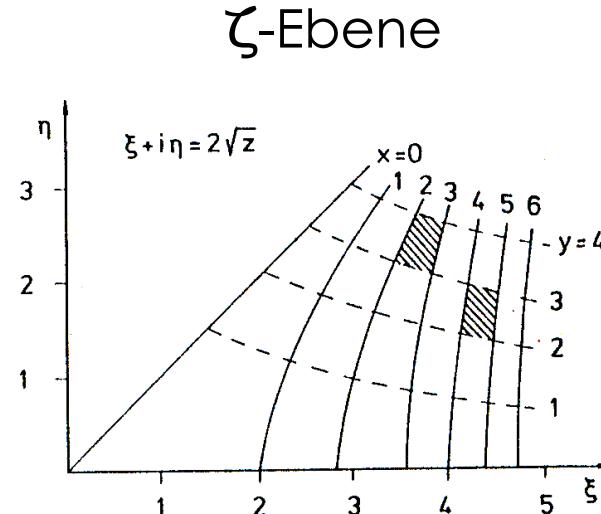
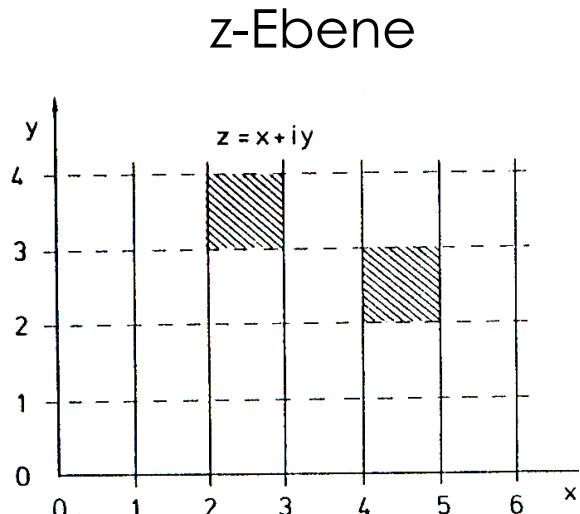


$$z(i) = -\frac{\tanh(\gamma(1 - \frac{2i}{N_z}))}{\tanh(\gamma)}$$
$$i=0 \dots N_z$$
$$\gamma=1.5$$

Erzeugung strukturierter Gitter

Konforme Abbildung

- Definiert für den gesamten Block, bzw. eine 2-D Ebene
- Komplexe Transformation $\zeta = \zeta(z)$
bildet ein Gebiet der komplexen z -Ebene ($z=x+iy$) auf ein
Gebiet in der komplexen ζ -Ebene ($\zeta=\xi+i\eta$) ab



$$\zeta = 2\sqrt{z}$$

Erzeugung strukturierter Gitter

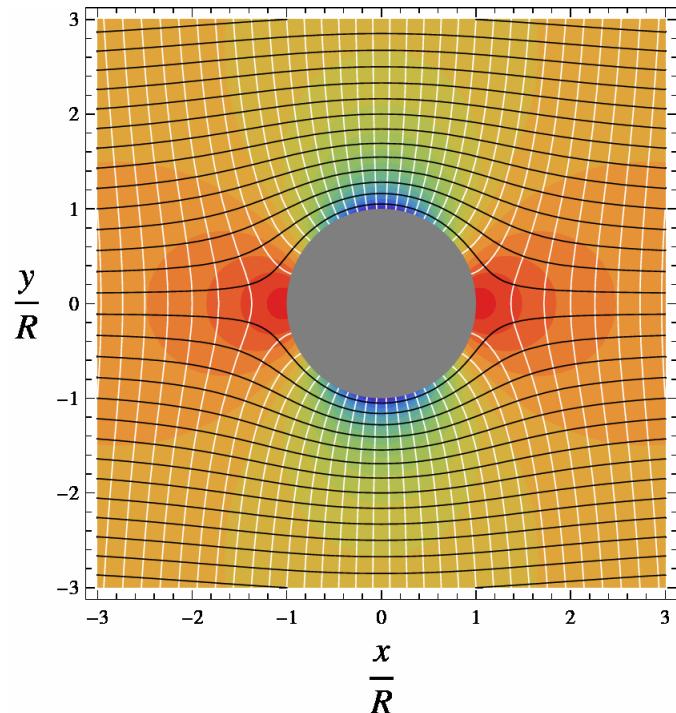
Elliptische Gittergenerierung

- Formulierung der Gittergenerierung als Randwertproblem
- Grundidee analog Potentialströmung:
 - Äquipotentiallinien ($\Phi=\text{const.}$) und Stromlinien ($\Psi=\text{const.}$) bilden ein orthogonales Netz, d.h.

$$\nabla\Phi \bullet \nabla\Psi = 0$$

- Es werden entweder Laplace- oder Poisson-Gleichungen gelöst:

$$\begin{array}{ccc} \Delta\Phi = 0 & & \Delta\Psi = 0 \\ & \downarrow & \\ \frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = 0 & & \frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = 0 \end{array}$$



Erzeugung strukturierter Gitter

- Das Ziel der elliptischen Gittergenerierung ist es, zu jedem Wertepaar (ξ, η) ein Wertepaar (x, y) zu finden. Deshalb muss die Laplace-Gleichung invertiert werden.

$$\alpha \frac{\partial^2 x}{\partial \xi^2} - 2\beta \frac{\partial^2 x}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 x}{\partial \eta^2} = 0$$

$$\alpha \frac{\partial^2 y}{\partial \xi^2} - 2\beta \frac{\partial^2 y}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 y}{\partial \eta^2} = 0$$

$$\alpha = \left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2$$

$$\beta = \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta}$$

$$\alpha = \left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2$$

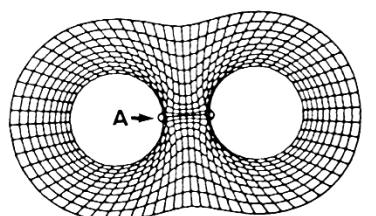
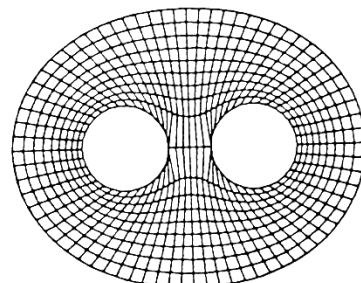
- Die Gleichungen für die Gitterpunktkoordinaten werden numerisch gelöst, als Randbedingungen dienen die Werte von x und y entlang der Randkurven.
- Elliptische Gittergenerierung glättet Unstetigkeiten wird oft zur Nachbearbeitung manuell erzeugter Gitter eingesetzt

Erzeugung strukturierter Gitter

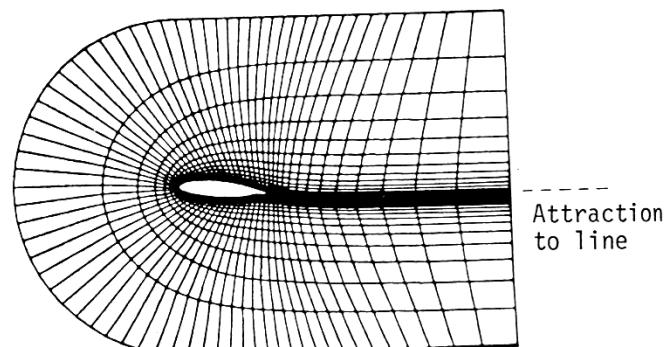
- Kontrollfunktionen (P und Q) können eingesetzt werden, um das Gitter in bestimmten Bereichen zu verdichten:

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = P(\xi, \eta)$$

$$\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = Q(\xi, \eta)$$



Verdichtung
um einen Punkt

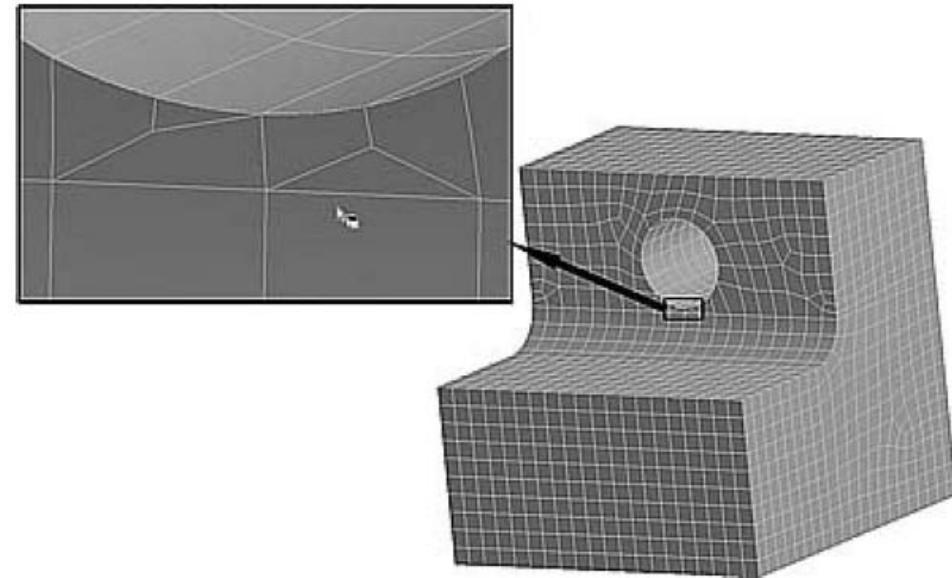
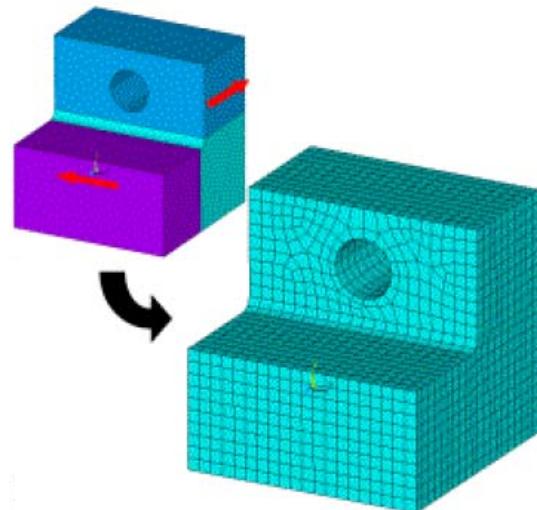


Verdichtung entlang
einer Linie

Erzeugung strukturierter Gitter

Automatische Generierung strukturierter Gitter?

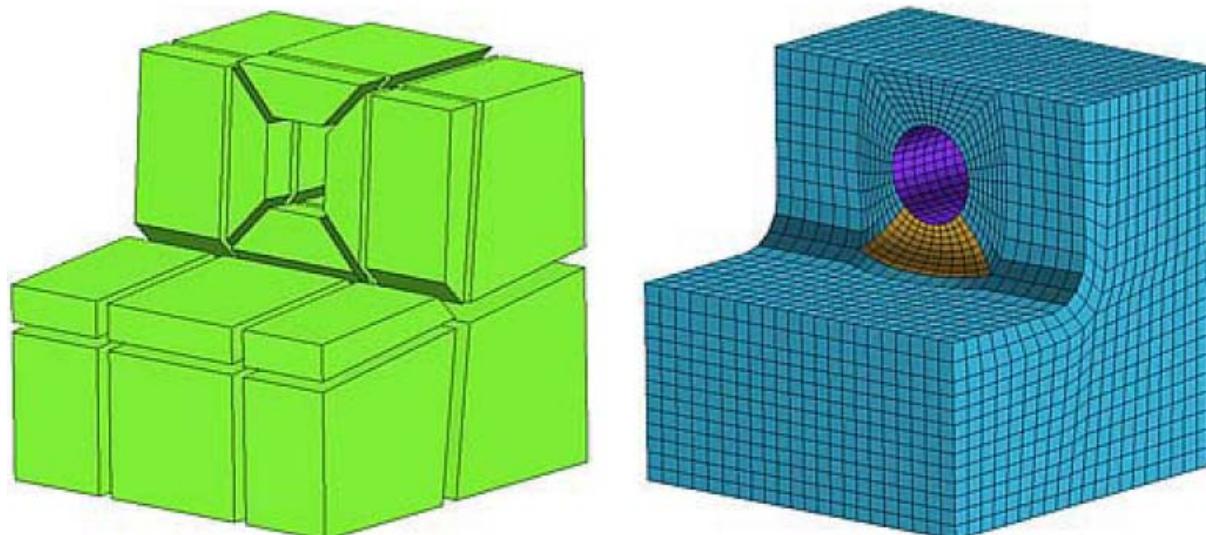
- Idee: Geometrie wird in mehrere Gebiete zerlegt und automatisch strukturiert vernetzt.
- Funktioniert bei einfachen Geometrien.
- Probleme bei komplexeren Berandungen.



Erzeugung strukturierter Gitter

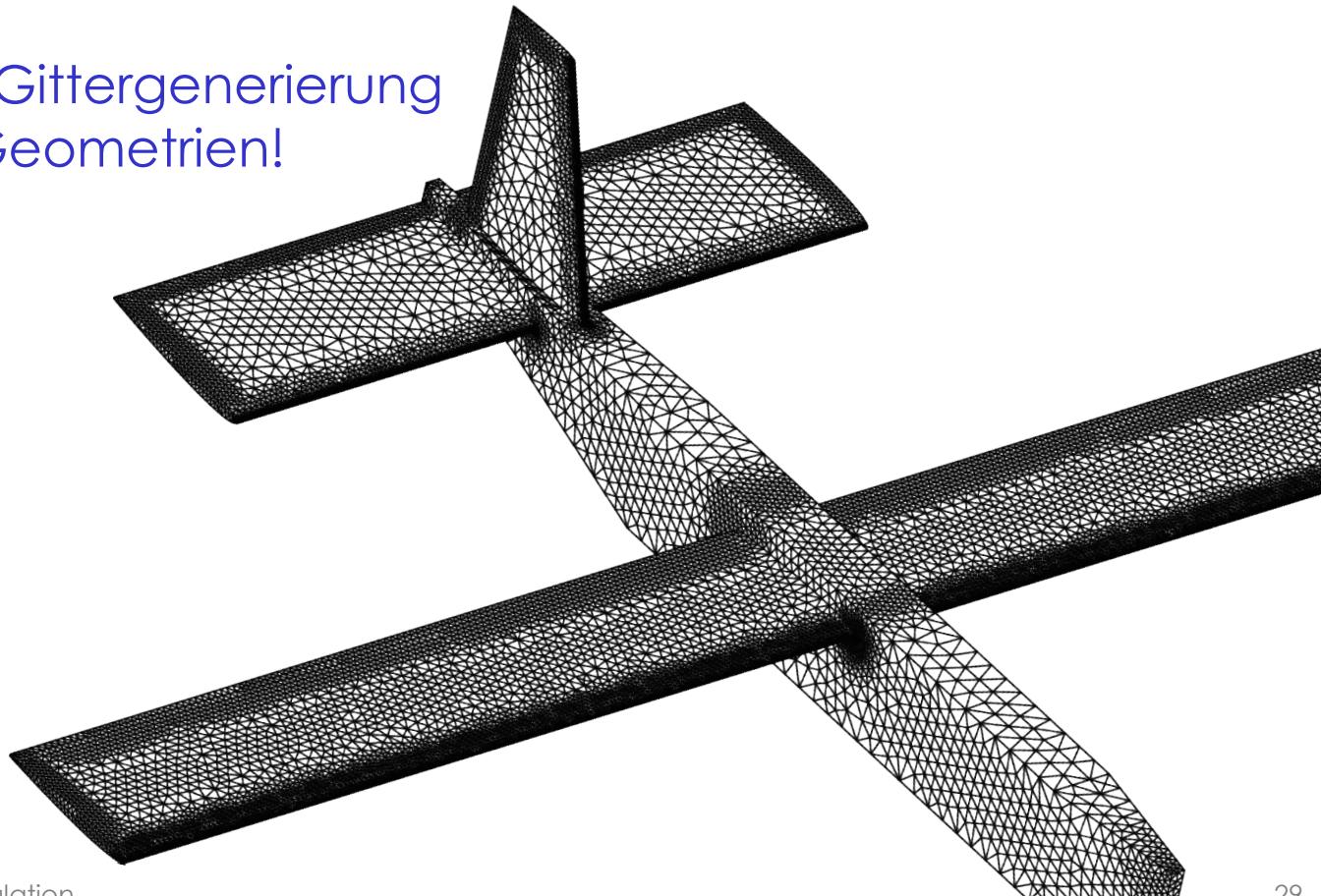
ICEM CFD

- Der Anwender zerteilt, streckt und verändert einen Block, welcher anschließend auf die Geometrie projiziert wird
- Vorteil: funktioniert auch bei komplexen Geometrien und beim Geometrieimport aus CAD-Programmen sehr zuverlässig (schlechte Qualität der Oberflächen)



Unstrukturierte Gitter

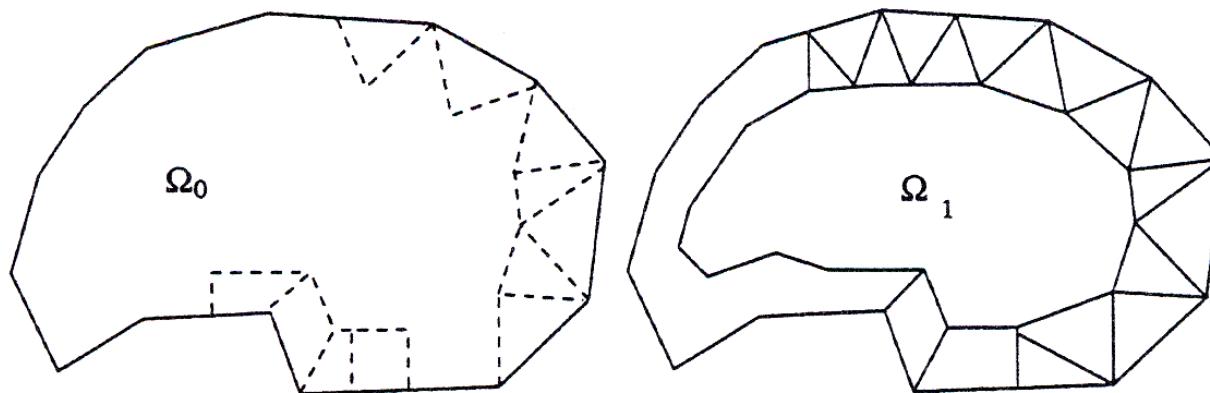
- Unstrukturierte Gitter bieten ein hohes Maß an Flexibilität.
- Tetraeder können einfacher in eine Geometrie eingepasst werden als Hexaeder.
- Automatische Gittergenerierung für komplexe Geometrien!



Unstrukturierte Gitter

Advancing-Front Methode

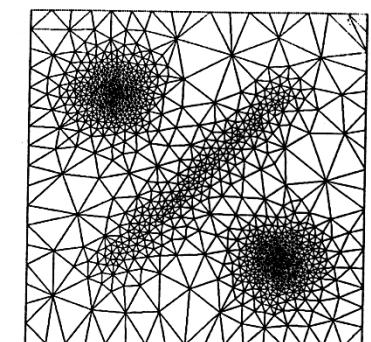
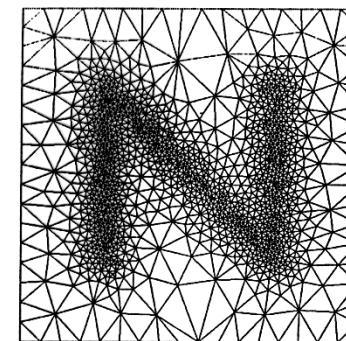
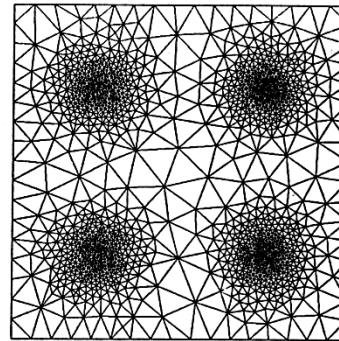
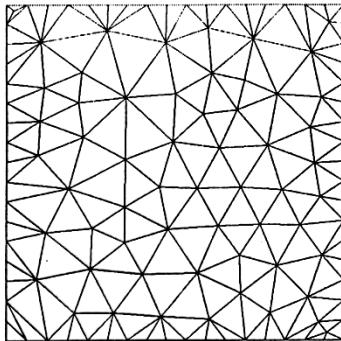
- Punkteverteilung auf dem Rand des Gebiets wird vorgegeben
- Entlang des Randes wird eine Schicht aus Elementen erzeugt
- Damit wird ein neuer geschlossener Rand definiert, die „advancing front“, welche das verbleibende Gebiet einschließt



- Das Verfahren wird fortgesetzt bis das gesamte Gebiet ausgefüllt ist

Unstrukturierte Gitter

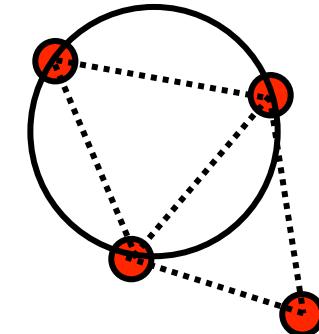
- In der Regel wird anschließend eine Glättung durchgeführt (siehe elliptische Gittergenerierung)
- Möglichkeit der Verwendung eines Hintergrundnetzes bei der zur Schrittweitensteuerung.



Unstrukturierte Gitter

Delaunay-Triangulation

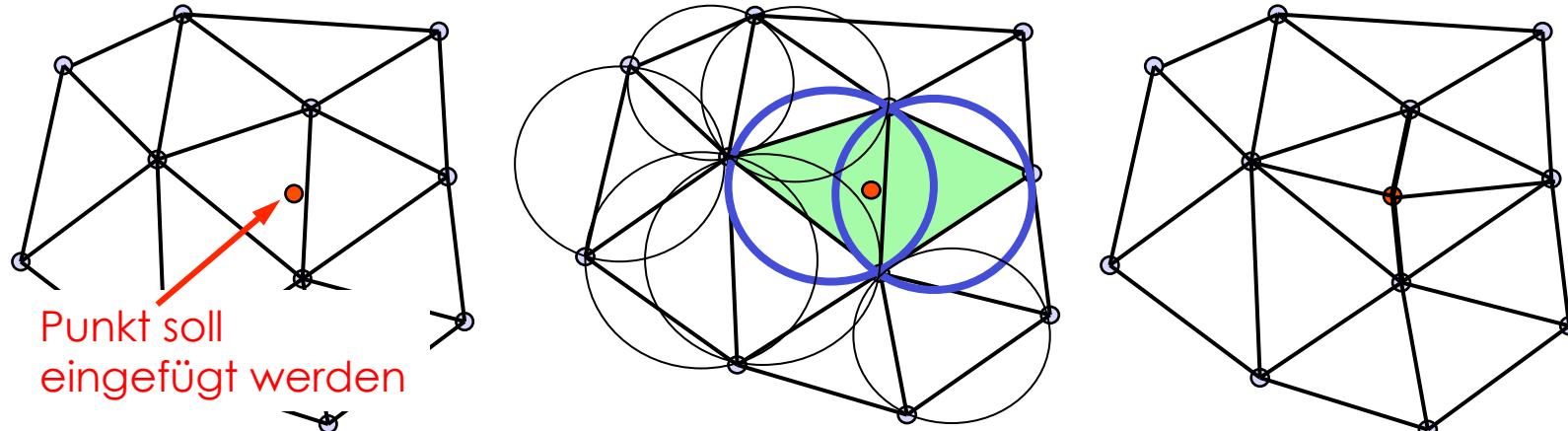
- Gegeben ist eine Punktwolke oder eine Verteilung von Punkten
- Gesucht ist die Zuordnungsmatrix (Connectivity-Matrix)
- Der Umkreis jedes Dreiecks enthält keine weiteren Punkte
- Der kleinste Winkel der Dreiecke wird maximal (Delaunay-Eigenschaft)



Triangulation: Erzeugen eines Dreiecksnetzes aus einer gegebenen Punktwolke.

Unstrukturierte Gitter

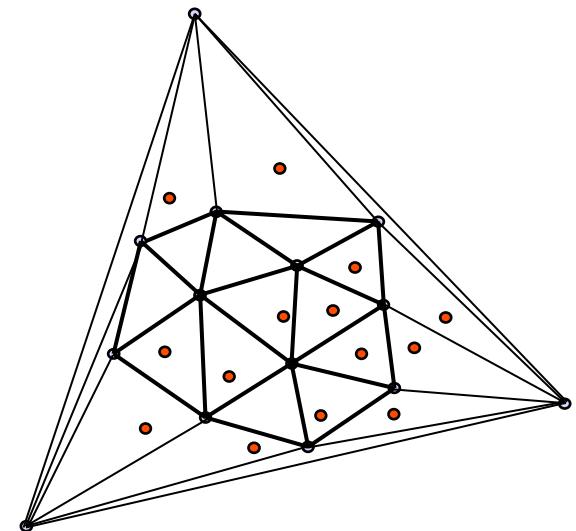
- Algorithmus Delaunay-Triangulation:
 1. Berechnung der Umkreise aller Dreiecke.
 2. Löschen derjenigen Dreiecke deren Umkreis den neuen Punkt enthält. Es entsteht ein Hohlraum der immer konvex ist.
 3. Verbinden des neuen Punktes mit den Eckpunkten des Hohlraums.



Unstrukturierte Gitter

- Woher die Punkte?

- Ausgangspunkt ist ein „Superdreieck“, und/oder eine Punktverteilung auf dem Rand des Gebietes.
- Sukzessives Einfügen neuer Punkte in die vorhandene Triangulation.
- Löschen der Punkte des Superdreiecks
 - bereits eingefügt
 - noch nicht eingefügt



Unstrukturierte Gitter

Advancing-Front

- Punkte und Elemente werden erzeugt
- mehrere Lösungen
- Eigenschaften der Elemente individuell kontrollierbar
- nicht robust, fehleranfällig (z.B. bei Inselbildung)
- Glättung notwendig

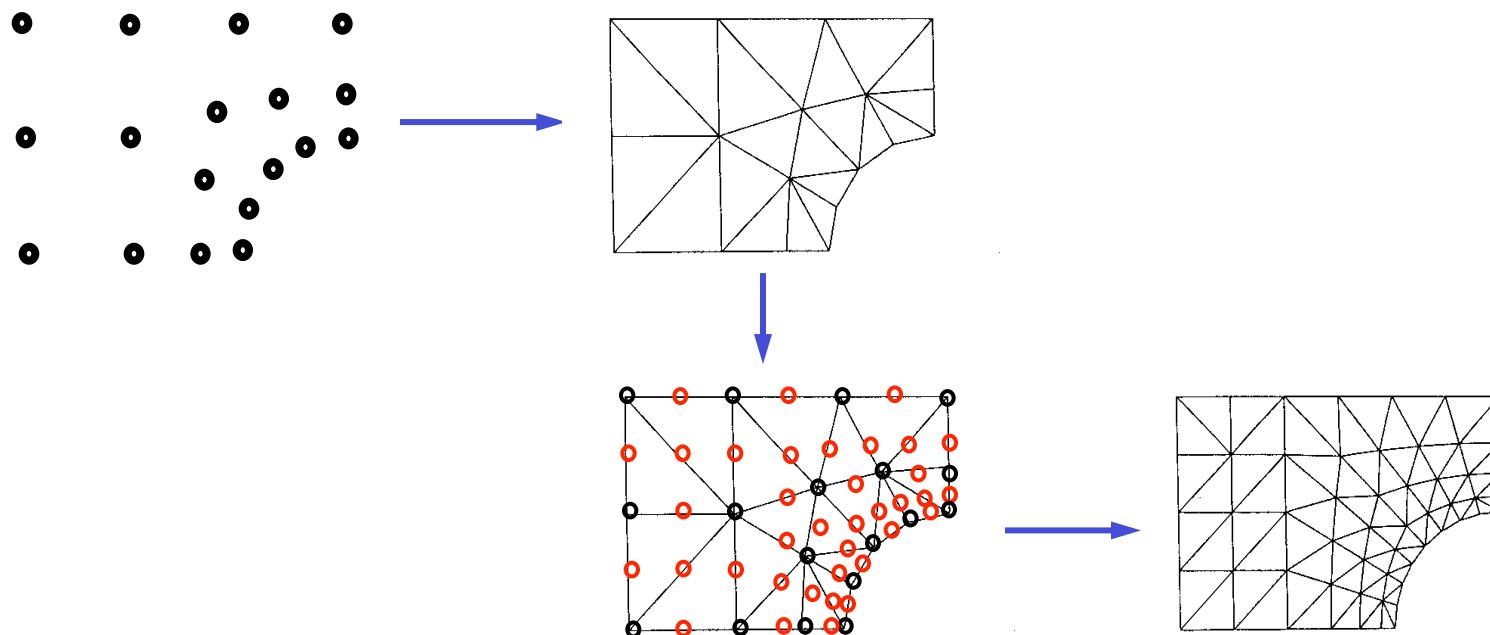
Delaunay-Triangulation

- Punkteverteilung ist vorgegeben
- Elemente werden erzeugt
- Nur eine optimale Lösung

Unstrukturierte Gitter

Möglichkeiten zur Gitterverfeinerung

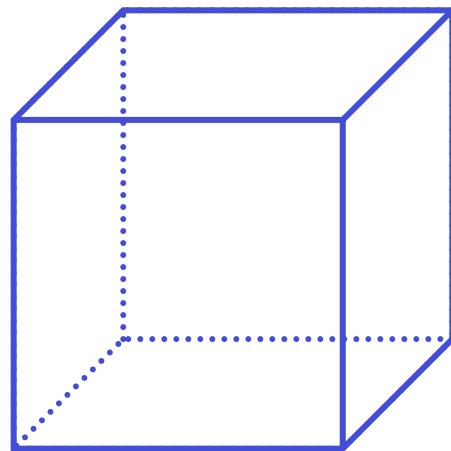
- Erzeugung neuer Punkte
 - am Dreiecksmittelpunkt,
 - am Schwerpunkt oder
 - auf dem Rand bestehender Dreiecke



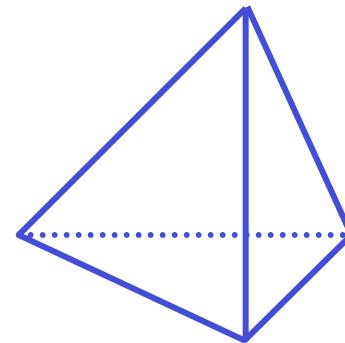
Warum strukturierte Gitter?

Speicherbedarf und Rechenzeit in 3-D

1 Hexaeder



= 6 Tetraeder bei gleicher
kleinster Kantenlänge



Genauigkeit der diskreten Approximation

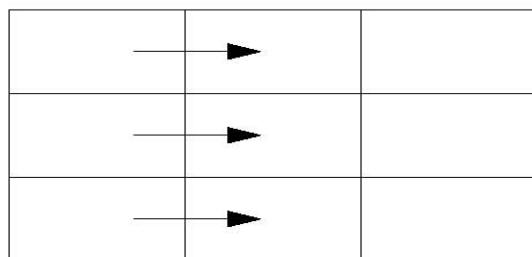
Numerische Diffusion

wird durch den Abbruchfehler des numerischen Diskretisierungsverfahrens verursacht und verfälscht das Simulationsergebnis.

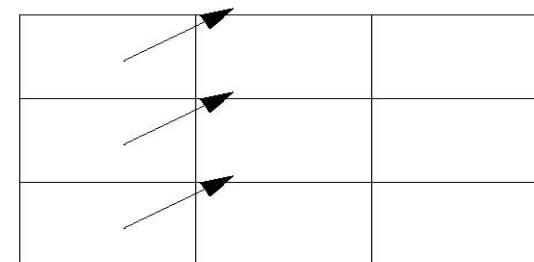
Numerische Fehler

Strukturiertes Gitter

- Verringerung der numerischen Diffusion
 - Zellflächen möglichst orthogonal anströmen
 - Gitternetzlinien entlang der Stromlinien ausrichten
- Sehr genaue Lösungen für Strömungen mit einer ausgeprägten Hauptströmungsrichtung (z.B. in der Nähe von Wänden) möglich.



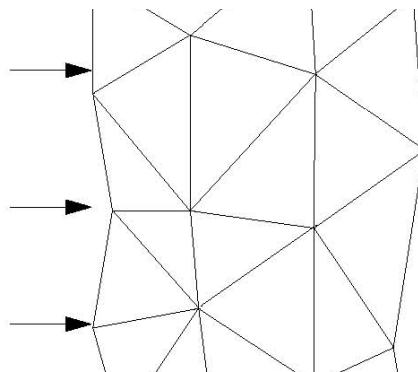
besser als



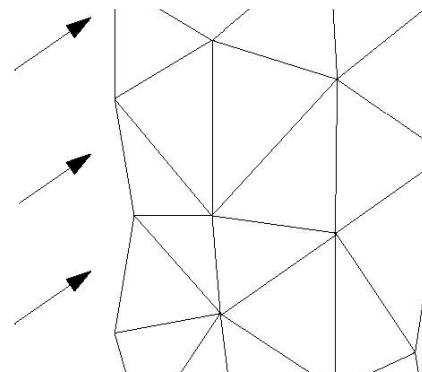
Numerische Fehler

Unstrukturiertes Gitter

- Es existiert keine Vorzugsrichtung in welcher der numerische Fehler minimal wird.



genauso
gut bzw.
schlecht



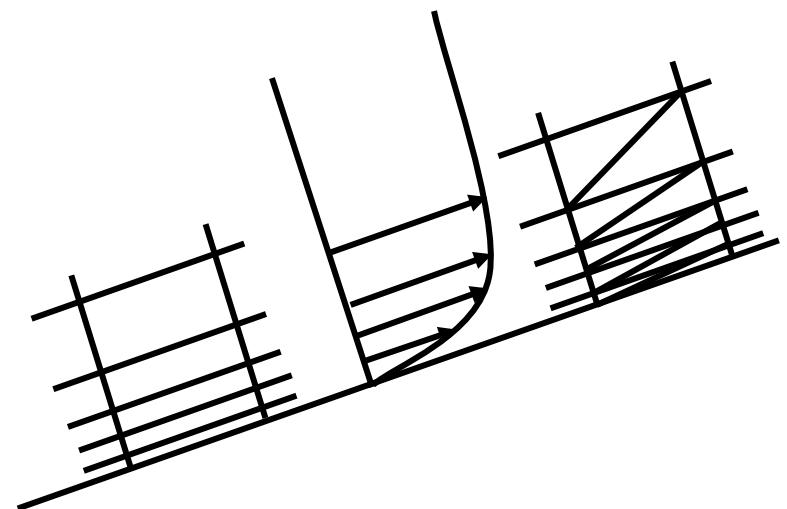
- Der numerische Fehler ist insgesamt größer als bei der Verwendung eines an die Strömung angepassten Hexaedergitters.
- Die Diskretisierungsfehler liegen aber im gesamten Gebiet in der gleichen Größenordnung.

Numerische Fehler

- Netz mit Tetraeder-Elementen und Upwind-Diskretisierung (UDS) erzeugt eine stärkere numerische Diffusion als ein Netz mit Hexaeder-Elementen.
- Verschiedene Untersuchungen zeigen für **stationäre** Probleme kaum Unterschiede hinsichtlich der Genauigkeit strukturierter und unstrukturierter Gitter (Aber beim numerischen Aufwand).
- Der Fehler nimmt schnell ab, wenn man eine Diskretisierung höherer Ordnung (CDS) verwendet oder das Gitter verfeinert.

ICEM

- zur genauen Simulation von Grenzschichten Prismen-Schicht einfügen
- Kombination der Vorteile von strukturierter und unstrukturierter Vernetzung



Auflösung von Grenzschichten

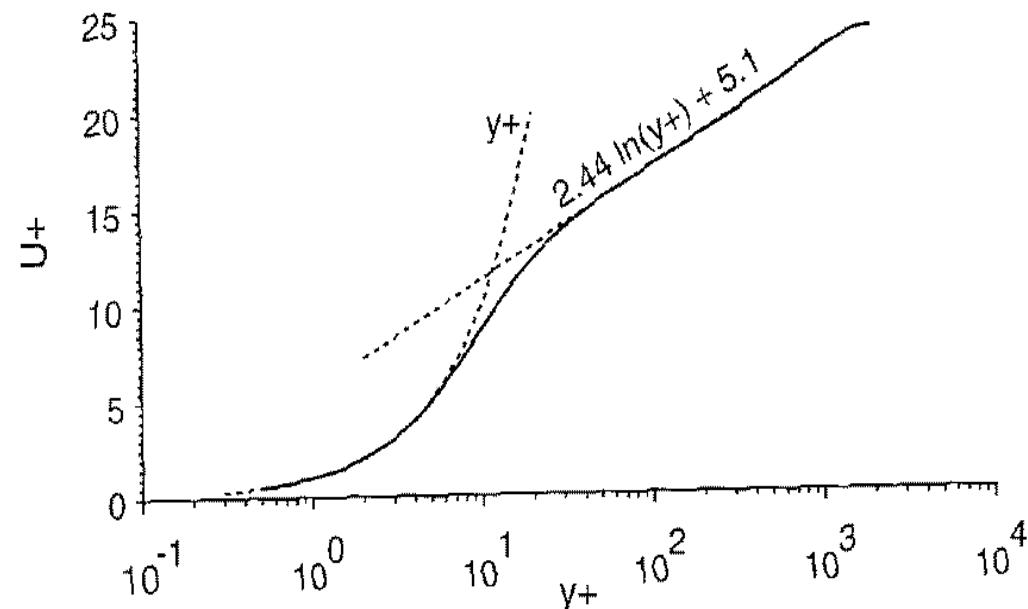
Empfohlener Wandabstand

- An Wänden herrschen große Gradienten.
- Turbulenzproduktion passiert zum großen Teil in $y^+ < 20$.
- Nur $y^+ \approx 1,0$ für den Abstand des ersten Gitterpunktes von der Wand liefert erfahrungsgemäß reproduzierbare Ergebnisse.

$$y^+ = \frac{u_\tau}{\nu} y$$

$$u_\tau = \sqrt{\frac{\bar{\tau}_w}{\rho}}$$

$$u^+ = \frac{u}{u_\tau}$$



Auflösung von Grenzschichten

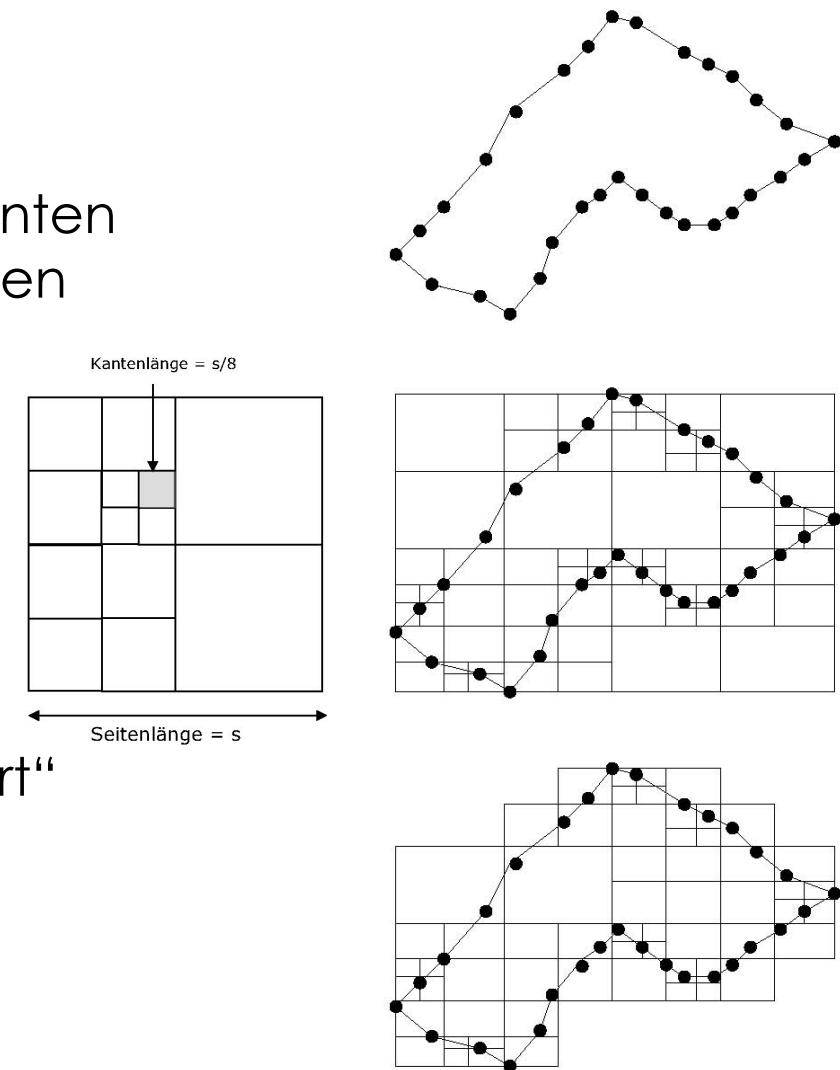
Wandmodelle

- Bei gutmütiger Strömung und einer speziellen Wandbehandlung kann y^+ auch deutlich höher liegen.
- CFX hat eine automatische Wandbehandlung optimiert für den logarithmischen Bereich des Geschwindigkeitsprofils.
 - Zum Auflösen der Wand müssen sogenannte „low-Re“ Modelle gewählt werden.
 - Bei Zweigleichungsmodellen beste Ergebnisse mit ω basierten Modellen.
- Viele Wandmodelle versagen bei Krümmung, Druckgradient und Nichtgleichgewicht.

Unstrukturierte Hexaedergitter

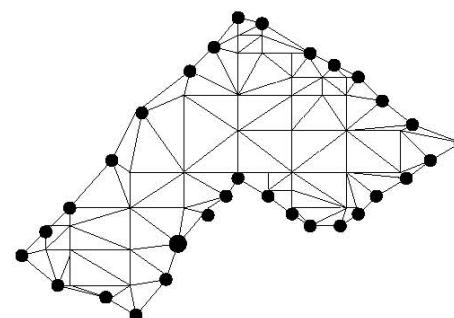
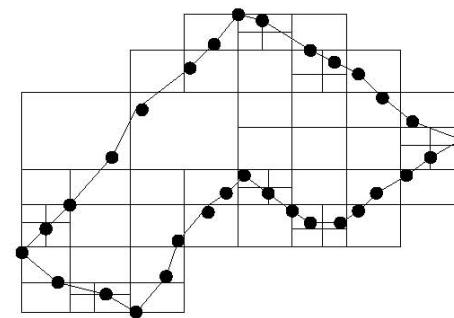
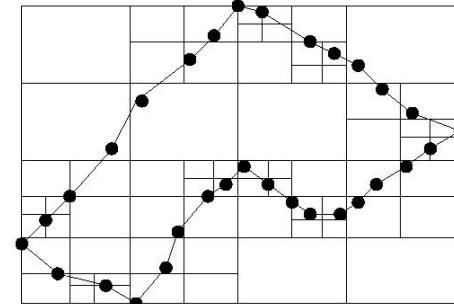
Octree-Verfahren

- Die Geometrie wird mit Elementen einer vorgegebenen maximalen Größe gefüllt.
- Wenn kleinere Elemente erforderlich sind, werden die ursprünglichen Zellen geteilt.
- Der Oktree wird „ausbalanciert“ und nicht innerhalb des Volumens liegende Punkte werden gelöscht.



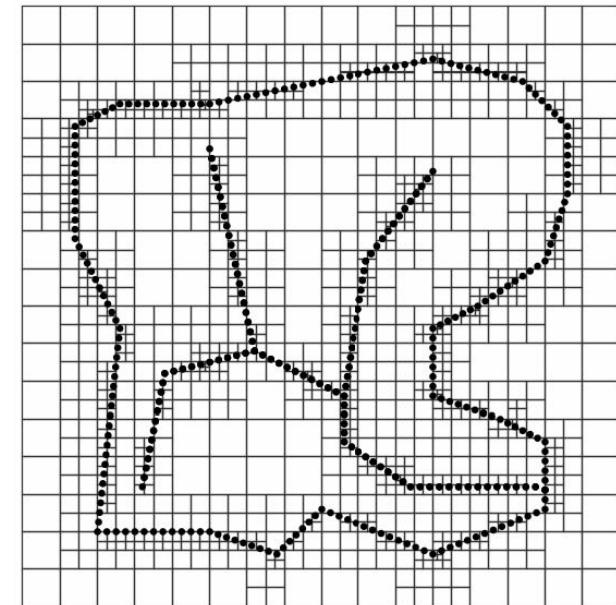
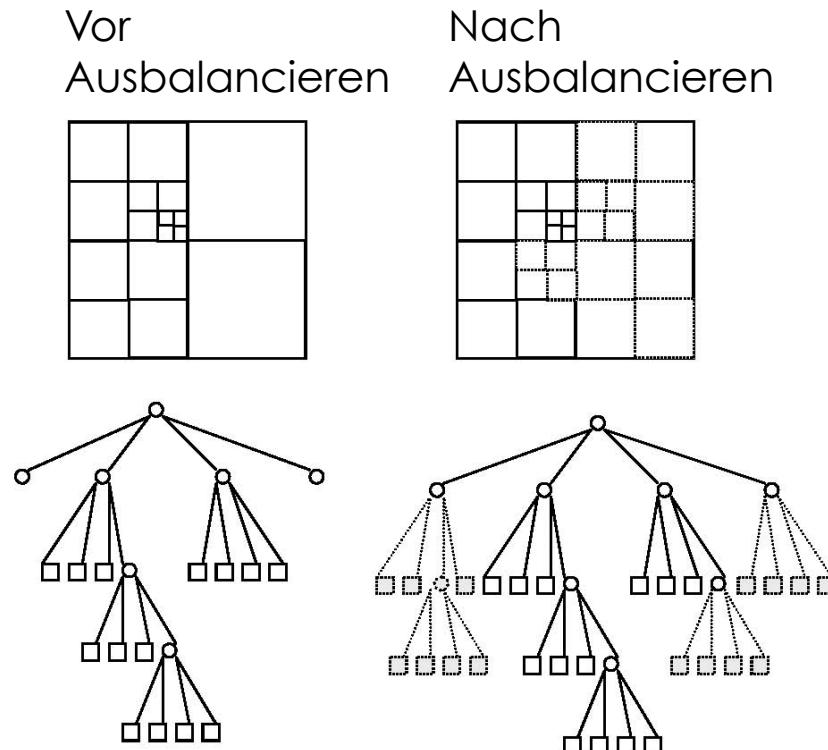
Unstrukturierte Hexaedergitter

- Octree Algorithmus:
 1. Grob Auffüllen
 2. Teilen bis Auflösung stimmt
 - Verschiedene Kriterien können kombiniert werden.
 3. „Ausbalancieren“
 - Nach der Verfeinerung können große Quadrate von beliebig kleinen Quadraten benachbart sein.
 - Zwei Nachbarquadrate dürfen sich nur um eine Tiefenebene unterscheiden.



Unstrukturierte Hexaedergitter

- Daten können in einer Baumstruktur gespeichert werden.
- Jedes Blatt des Baumes stellt dann einen strukturierten Gitterblock dar.
- Unstrukturierte Interpolationsvorschriften werden nur an den Blockgrenzen benötigt.



Unstrukturierte Hexaedergitter

