

Finite-Volumen-Verfahren auf unstrukturierten Gitter

K. Benkert¹, A. Stock²

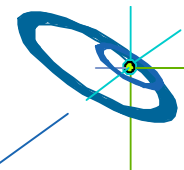
¹High Performance Computing Centre Stuttgart

www.hlrs.de

Universität Stuttgart

²Institut für Aerodynamik und Gasdynamik (IAG)

www.iag.uni-stuttgart.de



Grundidee Finite-Volumen-Verfahren (I)

Die Differenzialgleichung

$$u_t + \nabla \cdot f(u) = 0$$

stellt Anforderungen an Differenzierbarkeit, die häufig nicht erfüllt sind.
Dies gilt nicht für die integrale Form der Erhaltungsgleichung, hier für V_i :

$$\int_{t^n}^{t^{n+1}} \int_{V_i} u_t \, dV \, dt + \int_{t^n}^{t^{n+1}} \int_{V_i} \nabla \cdot f(u) \, dV \, dt = 0$$

Durch die Anwendung des Gaußschen Satzes erhält man:

$$\int_{t^n}^{t^{n+1}} \int_{V_i} u_t \, dV \, dt + \int_{t^n}^{t^{n+1}} \oint_{\partial V_i} f(u) \cdot \vec{n} \, dS \, dt = 0$$



Grundidee Finite-Volumen-Verfahren (II)

Ein wesentlicher Schritt zu einem numerischen Verfahren ist nun, Mittelwerte über Volumina V zu betrachten. Dann ist

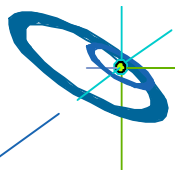
$$\int_{t^n V_i}^{t^{n+1}} \int u_t dV dt = V_i \int_{t^n}^{t^{n+1}} (u_i)_t dt$$

und man erhält:

$$V_i \int_{t^n}^{t^{n+1}} (u_i)_t dt + \int_{t^n}^{t^{n+1}} \oint_{\partial V_i} f(u) \vec{n} dS dt = 0$$

Wir definieren nun

$$R = -\frac{1}{V} \int_{t^n}^{t^{n+1}} \oint_{\partial V} f(u) \vec{n} dS dt$$



Grundidee Finite-Volumen-Verfahren (III)

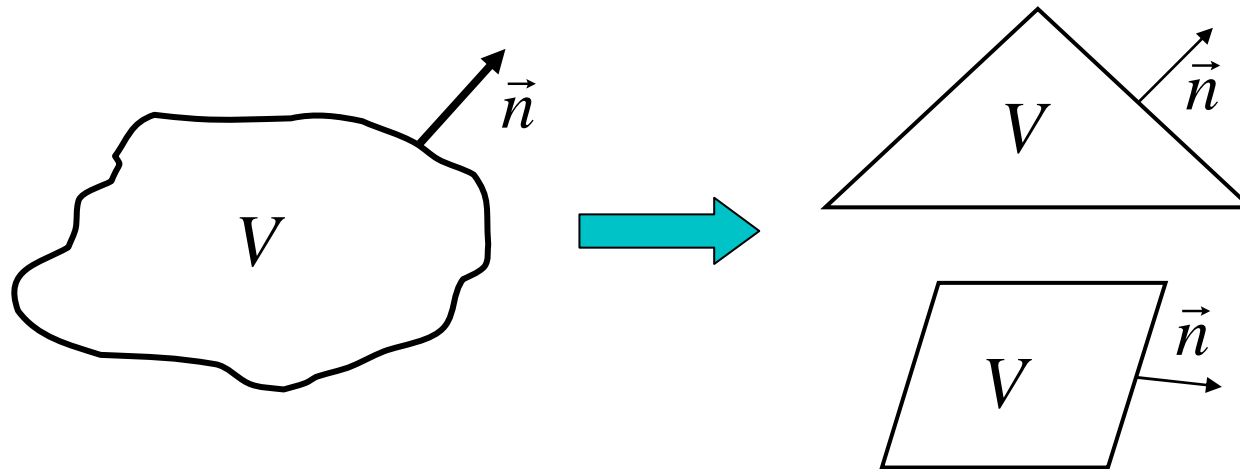
Bei R handelt es sich um das sog. Residuum, also die Änderung des integralen Mittelwerts eines Kontrollvolumens über eine Zeitspanne Δt . Ziel des Finite-Volumen-Verfahrens ist es nun, in jeder Iteration das Residuum zu bestimmen. Hierzu muss also folgender Ausdruck numerisch berechnet werden:

$$R = -\frac{1}{V} \int_{t^n}^{t^{n+1}} \oint_{\partial V} f(u) \vec{n} dS dt$$



Kontrollvolumen

Um nun ein numerisches Verfahren konstruieren zu können, welches das Oberflächenintegral löst, wird nun die Forderung gestellt, dass der Rand eines Kontrollvolumens aus stückweise glatten Elementen besteht:

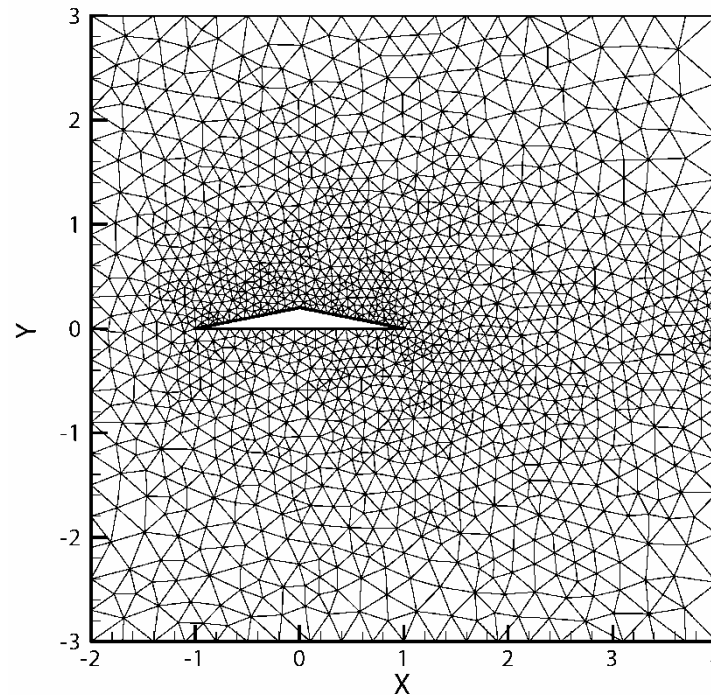


Es können beliebige Polyeder verwendet werden.



Rechengitter

Um nun ein komplettes Rechengebiet auf diese Art und Weise berechnen zu können, wird das komplette Gebiet mit einem Gitter überdeckt, dessen Gitterzellen die Kontrollvolumina darstellen:



Raumdiskretisierung - Ansatz

Unter Ausnutzung der vereinfachten Geometrie kann man nun den Ausdruck für das Residuum folgendermaßen räumlich diskretisieren:

$$R_i = -\frac{1}{V_i} \int_{t^n}^{t^{n+1}} \sum_{e_{ij} \subset \partial V_i} \oint f(u) \vec{n} dS dt$$

Die Integration erfolgt mittels Gaussquadratur, für Verfahren erster und zweiter Ordnung mit einer Stützstelle an der Seitenmitte.

$$\Rightarrow R_i = -\frac{1}{V_i} \int_{t^n}^{t^{n+1}} \sum_{e_{ij} \subset \partial V_i} |e_{ij}| g(u_i, u_j; \vec{n}_{ij}) dt$$

Hierbei sind e_{ij} die Randkanten von V_i und g die numerische Flussfunktion, also eine geeignete Approximation an f .



Die Eulergleichungen in zwei Raumdimensionen

Die Eulergleichungen in zwei Raumdimensionen lauten:

$$u_t + f_1(u)_x + f_2(u)_y = 0$$

$$u = \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ e \end{pmatrix} \quad f_1(u) = \begin{pmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ v_1(e + p) \end{pmatrix} \quad f_2(u) = \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ v_2(e + p) \end{pmatrix}$$

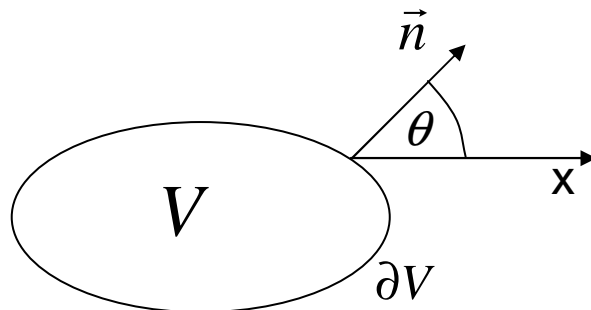
$$\text{Zustandsgleichung: } p = (\gamma - 1)\rho\varepsilon \Leftrightarrow p = (\gamma - 1)\left(e - \frac{1}{2}\rho v^2\right)$$

Die Eulergleichungen sind **rotationsinvariant**.



Flussberechnung in mehreren Raumdimensionen

Für ein allgemeines Kontrollvolumen V kann man den Normalenvektor auf dem Rand folgendermaßen definieren:



$$\vec{n} = (n_1, n_2) = (\cos \theta, \sin \theta)$$

Der Integrand $f(u) \cdot \vec{n}$ kann dann folgendermaßen umgeformt werden:

$$f(u) \cdot \vec{n} = \cos \theta f_1(u) + \sin \theta f_2(u)$$

Dieser Ausdruck kann über die Ausnutzung der Rotationsinvarianz folgendermaßen umgeformt werden:

$$\cos \theta f_1(u) + \sin \theta f_2(u) = T^{-1} f(Tu)$$



Die Matrix T ist eine Drehmatrix:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & n_1 & n_2 & 0 \\ 0 & -n_2 & n_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & n_1 & -n_2 & 0 \\ 0 & n_2 & n_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Wir erhalten somit für das Residuum:

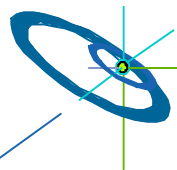
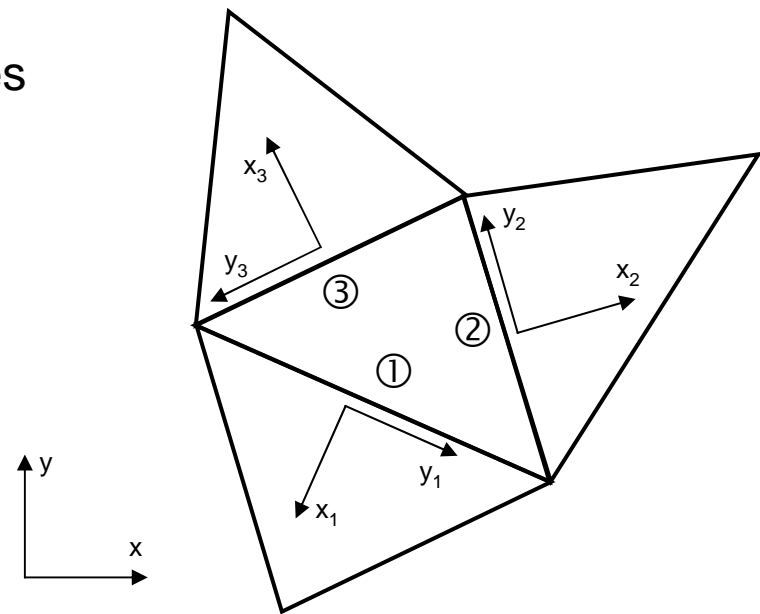
$$\Rightarrow R_i = -\frac{1}{V_i} \int_{t^n}^{t^{n+1}} \sum_{e_{ij} \subset \partial V_i} |e_{ij}| [T^{-1} g(Tu_i, Tu_j; (1,0)^T)] dt$$



Flussberechnung - Zusammenfassung

Das Vorgehen lässt sich folgendermaßen zusammenfassen:

- An jeder Zellgrenze wird der Zustand der beiden angrenzenden Zellen in ein lokales Koordinatensystem gedreht
- Das resultierende 1D-Riemannproblem an der entsprechenden Kante wird gelöst (Übergabe an die Flussfunktion)
- Rückrotation des berechneten Flusses ins globale Koordinatensystem und Berechnung des Residuums der Zelle
- Integration des Flusses über die Kante mit Mittelpunktsregel
- Fluss in Nachbarzelle updaten



Flussberechnung – Hinweise zur Implementierung

- Die Schleife zur Flussberechnung geht über alle Kanten und nicht über alle Zellen um zu vermeiden, den Fluss zwei mal berechnen zu müssen
 - Rechenzeiterparnis
 - Konservativität wird erhalten
- Für jede Zellkante wird nur ein Normalenvektor abgespeichert, d.h. der Normalenvektor zeigt für eine Zelle in die falsche Richtung.
- Je nach Richtung des Normalenvektors geht der Fluss positiv oder negativ in die jeweilige Zelle ein.

