„Sharp Immersed Boundary“ Implementierung zur Ermöglichung von Strömungsakustiksimulationen um komplexe Geometrien

Jürgen Dierke[1]

[1] *Institut für Aerodynamik und Strömungstechnik - Technische Akustik, DLR Braunschweig, E-Mail: Juergen.dierke@dlr.de*

Einleitung

Seit vielen Jahren schon werden aeroakustische Probleme numerisch berechnet und obwohl das schnell teuer wird in Hinblick auf Rechnerressourcen ist eine der Hauptherausforderungen bei großen komplexen Problemen nicht der Rechenaufwand, sondern die Notwendigkeit das System zu vernetzen. Diese Aufgabe erfordert viel händische Arbeit und ist nicht wirklich parallelisierbar. Im schlimmsten Fall mag ein Fall auch nicht vernetzbar sein.

Ein Ausweg scheint die „Immersed Boundary“ zu sein. Die Konfiguration ist dabei separat vernetzt, muss jedoch nicht vom Rechengitter beschrieben werden. Die Geometrie schwebt quasi zwischen den Punkten des Rechengitters. Die Auswirkungen der Konfiguration auf die Simulation sind jedoch so, dass die Randbedingung am Ort der Geometrie erfüllt ist.

Die hier genutzte Art der Implementierung wurde von Cheng Zhao et al. [1] vorgestellt. Weiterhin wurde die Randbedingung nicht in einen rein kartesischen Code implementiert, sondern in den PIANO Code [3], der blockstrukturiert auf gekrümmten Gitter Gleichungssysteme löst. Dies hat Vor-, als auch Nachteile auf die im Weiteren eingegangen wird.

CAA Code PIANO

Die Randbedingung wurde in den CAA-Code PIANO [3] implementiert. PIANO ist ein blockstrukturierter finite Differenzen Code mit 4. Ordnung Diskretisierung in Raum (DRP [1]) und Zeit (LDDRK [5]) auf gekrümmten Gittern. Stabilität kann durch „Artifical Selective Damping“ (ASD [4]) oder Filter 6. Ordnung verbessert werden. Körperangepasste Wände besitzen einen „Ghostpoint“ für den fluktuierenden Druck oder die fluktuierende Dichte; zur Wand hin sind die Stencil asymmetrisch. Weiterhin sind hängende Knoten für eine 1:2 Verfeinerung nach Tam et al. [7] implementiert.

Immersed Boundary Konzept

Grundlegend ist erstmal, dass die IBC eine Randbedingung ist, die nicht an Gitterpunkte gebunden ist. Somit sind Rechengitter und Geometrie unabhängig! Die Beschreibung der Geometrie liegt im STL-Format vor.

Das Prinzip ist wie folgt:

- Identifikation der Punkte innerhalb und außerhalb der Geometrie. Ein Strahlsegment ist der Vektor von i zu i+1, respektive j und k. Für alle Strahlsegmente eines Blocks werden Kollisionen mit den Dreiecken der Geometrie überprüft. Sofern eine Kollision stattfindet wird der erste innere Punkt als virtueller Rechenpunkt markiert. Für eine Kollision aus jeder Richtung kann ein unterschiedlicher virtueller Rechenpunkt an derselben Stelle gefunden werden. Dies findet außerhalb der Zeitschleife statt (Initialisierung).
- Spiegelpunkte („Mirrorpoints“) der virtuellen Rechenpunkte (GP) relativ zur Geometrie werden berechnet und Position im Rechengitter bestimmt. (Initialisierung)
- Fluidwerte werden auf die Spiegelpunkte (MPs) interpoliert. Dargestellt in Abbildung 1 & Abbildung 4. (Zeitbereich)
- Abhängig von der gewünschten Randbedingung und der Spiegelpunktwerte werden die Werte der virtuellen Rechenpunkte gesetzt; siehe Abbildung 2. (Zeitbereich)
- Werte der virtuellen Rechenpunkte werden benutzt um wandnahe Ableitungen zu berechnen. (Zeitbereich)
    - Ableitungen sind immer zentral, aber von reduzierter Ordnung (Abbildung 3).

Das Konzept ermöglicht es auch Ecken berücksichtigen zu können (Abbildung 5). Weiterhin können dadurch auch dünne Geometrien ohne virtuelle Punkte innerhalb der Geometrie dargestellt werden (Abbildung 5).

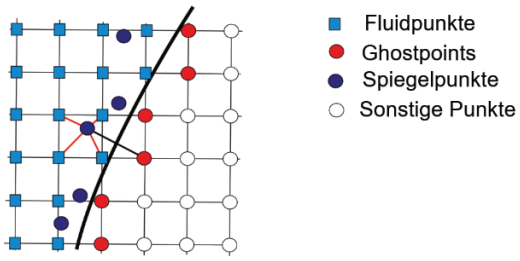


Abbildung 1: Skizze des Konzepts der IBC. Geometrie (schwarze Linie) verläuft zwischen den Rechengitterpunkten. Ghostpoints (rot) besitzen Spiegelpunkte (dunkel blau). Die Interpolation der Fluidpunktwerte auf einen Spiegelpunkt ist mit roten Linien angedeutet.

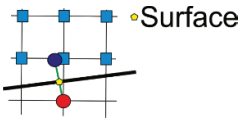


Abbildung 2: Skizze des Randbedingungkonzepts. Spiegelpunktwerte (lila) sind gegeben. Wenn der Wandwert (gelb) Null sein soll, so muss der Ghostpoint-Wert (rot) der negative Spiegelpunktwert sein. Soll der Gradient Null sein, so muss der Ghostpoint-Wert dem Spiegelpunktwert entsprechen. Denkbar sind jedoch auch alle anderen Zielwerte auf der Oberfläche z.B. für aktive Wände.

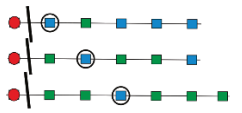


Abbildung 3: Skizze der Reduzierten Differenzensterne in der Nähe der IBC. Umkreist sind die Punkte für die die Ableitungen berechnet werden. Rot und grün sind die Punkte die einen Betrag zur Ableitung leisten.

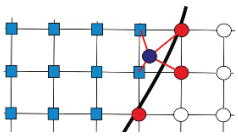


Abbildung 4: Skizze zeigt die Abhängigkeit der Spiegelpunktwerte (lila) von Ghostpoint-Werten (rot). Da diese wieder von den Spiegelpunktwerten abhängig sind, ergibt sich ein Gleichungssystem.

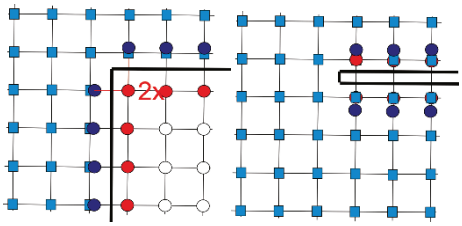


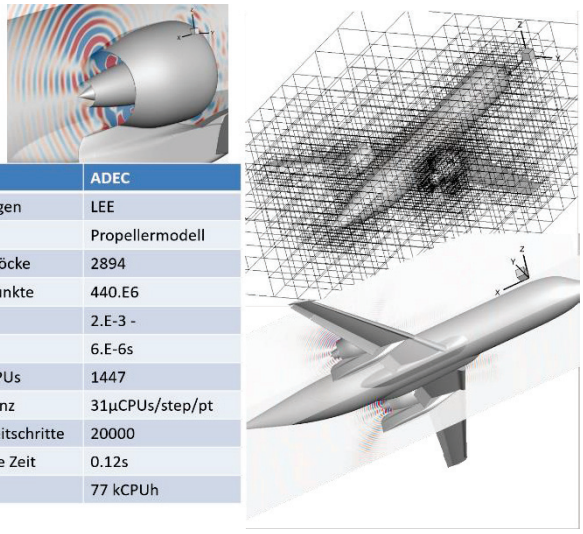
Abbildung 5: Skizze zur Verdeutlichung der Vorzüge von Gitterpunkten zugeordneter, jedoch nicht zum Rechengitter gehöriger Ghostpoints. Es sind sowohl scharfe Kanten, als auch dünne Geometrien möglich, die keine Ghostpoints beinhalten.

Beispielfälle

Anhand einiger Beispiele wird in diesem Abschnitt gezeigt, dass komplexe Fälle nun vernetzbar sind, was mit körperangepassten strukturierten Gittern nicht möglich gewesen wäre und bei unstrukturierten Lösern einen erheblichen Geometrieaufbereitungsaufwand bedeutet hätte. Die Bilder beinhalten immer Informationen über das gelöste Gleichungssystem, die Art der Quelle, Anzahl der strukturierten Blöcke, Anzahl der Gitterpunkte, dimensionslose und dimensionsbehaftete Zeitschritt, sowie Leistungsdaten abhängig von der Anzahl genutzter CPUs, als auch Informationen zu den Rechenkosten in Abhängigkeit der simulierten Zeit.

Beispiel 1: Flugzeug mit Klappen

Dieser Testfall ist eine Machbarkeitsstudie für große Fälle, als auch ein Test ob bewegte Quellen Nahe der IBC für Probleme sorgen. Die Größe des Falls, sowie oben gelisteter Parameter und eine Übersicht über das Rechengitter sind in Abbildung 6 gegeben. Es hat sich herausgestellt, dass die Performanz trotz des blockübergreifenden Ghostpoint-Systems nicht gelitten hat und auch keine Störungen an der IBC entstehen. Eine strukturierte Vernetzung mit einem körperangepassten Gitter wäre mit Sicherheit nur durch erhebliche Geometrievereinfachungen möglich gewesen und dann auch nur unter erheblichem Arbeitsaufwand. Die Vernetzung für die IBC hat weniger als 10 Minuten gedauert!

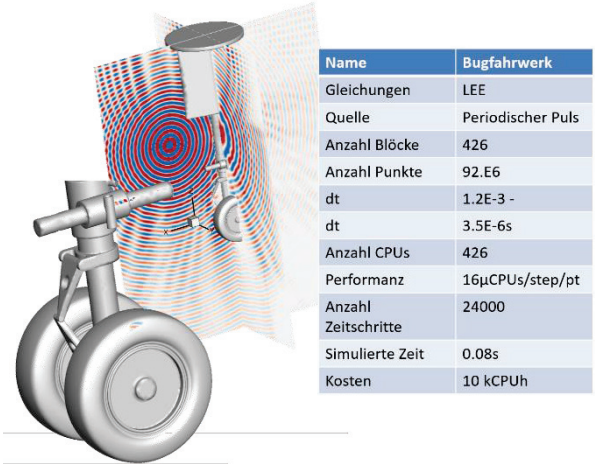


Name	ADEC
Gleichungen	LEE
Quelle	Propellermodell
Anzahl Blöcke	2894
Anzahl Punkte	440.E6
dt	2.E-3 -
dt	6.E-6s
Anzahl CPUs	1447
Performanz	31µCPUs/step/pt
Anzahl Zeitschritte	20000
Simulierte Zeit	0.12s
Kosten	77 kCPUh

Abbildung 6: Beispiel eines Flugzeugs mit ausgeschlagenen Klappen unter Nutzung eines Propellermodels als Testschallquelle im Triebwerk. In blau/rot dargestellt ist der fluktuierende Druck.

Beispiel 2: Komplexes Bugfahrwerk

Der Testfall des komplexen Bugfahrwerks ist gewählt worden, weil er einerseits körperangepasst überhaupt nicht vernetzbar war, und andererseits, weil die filigranen Strukturen häufig auf unterschiedliche Weise die Möglichkeit der IBC dünne Geometrien darzustellen ausreizt. Der Testfall und dessen Parameter sind in Abbildung 7 dargestellt.

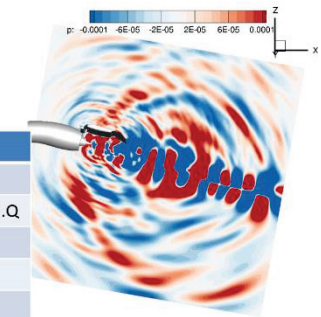


Name	Bugfahrwerk
Gleichungen	LEE
Quelle	Periodischer Puls
Anzahl Blöcke	426
Anzahl Punkte	92.E6
dt	1.2E-3 -
dt	3.5E-6s
Anzahl CPUs	426
Performanz	16µCPUs/step/pt
Anzahl Zeitschritte	24000
Simulierte Zeit	0.08s
Kosten	10 kCPUh

Abbildung 7: Beispiel eines komplexen Bugfahrwerks. Die Quelle ist eine harmonische Monopolquelle und dargestellt ist der fluktuierende Druck.

Beispiel 3: Strahl-Klappen-Interaktion

Der Strahl-Klappen-Interkations Fall ist wieder ein Beispiel das nur mit starken Vereinfachungen klassisch vernetzt werden könnte. Dargestellt ist das Beispiel in Abbildung 8. Die Windkanaldüsengeometrie des UHBR-Triebwerks ist links im Bild. Der Vorflügel des Hochauftriebsflügels ist gespalten und nur spannweitig verschoben zur Triebwerksposition ausgeschlagen; somit werden auch Seitenkanten mitsimuliert.

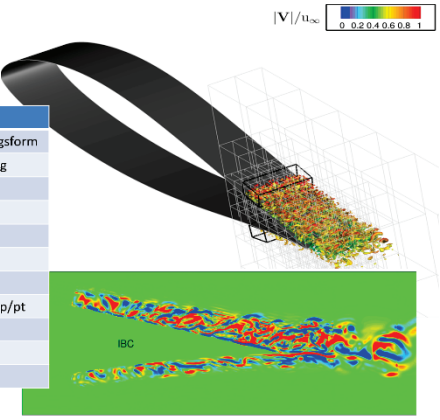


Name	JetFlap
Gleichungen	LEE
Quelle	Instabilität+Stoch.Q
Anzahl Blöcke	51
Anzahl Punkte	18.E6
dt	5.E-4 -
dt	1.5E-6s
Anzahl CPUs	52
Performanz	11µCPUs/step/pt
Anzahl Zeitschritte	20000
Simulierte Zeit	0.03s
Kosten	1100 kCPUh

Abbildung 8: Beispiel eines Strahl-Klappen-Interaktion Problems. Die stochastische Quelle bildet das Tam & Auriault-Strahllärmmodell [6,8] ab. Es ergeben sich großskalige physikalische Strukturen durch Nutzung der LEE über einer RANS-Hintergrundströmung.

Die Nutzung der LEE über einer RANS-Hintergrundströmung liefert die Fähigkeit aufklingende Strömungsinstabilitäten zu simulieren, die in Abbildung 8 zu finden sind. Unphysikalische, numerische Instabilitäten in IBC-Nähe werden jedoch nicht beobachtet.

Beispiel 4: Zonale LES für Hinterkante eines Profils



Name	ZLES DU96
Gleichungen	NS in Störungsform
Quelle	Stoch. Forcing
Anzahl Blöcke	51
Anzahl Punkte	5.E6
dt	5.E-4 -
dt	1.5E-6s
Anzahl CPUs	52
Performanz	11µCPUs/step/pt
Anzahl Zeitschritte	20000
Simulierte Zeit	0.03s
Kosten	30 kCPUh

Abbildung 9: Beispiel einer zonalen LES um die Hinterkante eines DU96-180 Windkraftanlagenprofils. In den schwarzen Kästen wird stochastisch isotrope Turbulenz eingeimpft, die schnell in anisotrope Turbulenz übergeht.

Beispiel 4 zeigt die Möglichkeit die IBC auch für Zonale LES [8] zu nutzen (Abbildung 9). Es wird stochastisch isotrope Turbulenz auf Saug- und Druckseite geimpft die schnell in realistische anisotrope Turbulenz übergeht; zu erkennen an den länglichen Strukturen im Bild. Das Ziel realistische Turbulenz an der Hinterkante zu erhalten um Hinterkantenschall zu simulieren wird mit der IBC erreicht ohne Störschall zu generieren.

Fazit Beispiele

Alles in allem ist das Verfahren robust und schnell für die exemplarisch gezeigten Rechenfälle und das für verschiedene Gleichungssysteme. Komplexe Fälle sind schnell und einfach zu vernetzen und auch die Rechenzeit wächst nicht an im Vergleich zu klassischen Randbedingungen.

Vorteile gegenüber klassischen Randbedingungen

Die Vorteile der IBC Vernetzung sind sowohl gegenüber der klassisch körperangepassten strukturierten Vernetzung, als auch im Vergleich zur unstrukturierten Vernetzung vorhanden (Abbildung 10 & Abbildung 11).

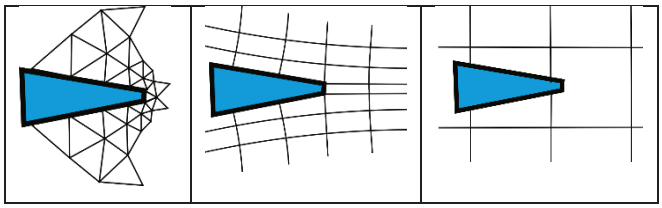


Abbildung 10: Einfluss eines kleinen geometrischen Details auf die Vernetzung. (links: unstrukturierte Vernetzung erzeugt Vielzahl kleiner Elemente, die expliziten Zeitschritt stark reduzieren; Mitte: klassische strukturierte Vernetzung erzeugt kleinere Elemente, die expliziten Zeitschritt stark reduzieren; rechts: strukturierte Vernetzung für IBC ist unabhängig von geometrischen Details.

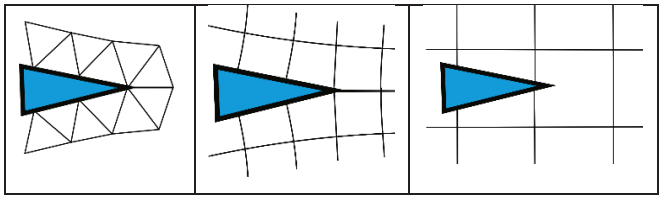


Abbildung 11: Aufbereitete kleine geometrische Details z.B. durch zusammenkneifen von Hinterkanten sorgen für gleichmäßige Zellen für unstrukturierte Gitter (links), sowie strukturierte Gitter ohne und mit IBC (Mitte und rechts). Es erfordert aber viel manuellen Aufwand.

Zum einen können kleine geometrische Strukturen für sehr kleine Zellen und damit verbundene kleine Zeitschritte sorgen und zum anderen wird häufig ein großer Aufwand benötigt um Geometriebeschreibung aufzubereiten um eine nutzbare Oberflächenbeschreibung zu bekommen (Abbildung 12). Beide Probleme sind für die hier beschriebene Randbedingung nicht vorhanden.

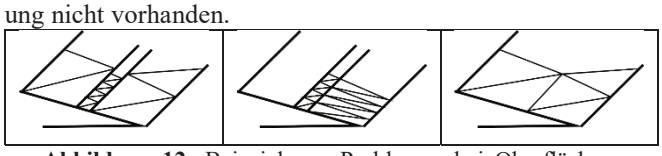


Abbildung 12: Beispiel von Problemen bei Oberflächenvernetzung. Bei der Oberflächenvernetzung kann es passieren, dass Panel die eine Oberfläche beschreiben ungleichmäßig vernetzt werden (links, Mitte) und dann erst aufbereitet werden müssen um ein gleichmäßiges Gitter (rechts) zu erhalten.

Vor- und Nachteile einer IBC in einem krummlinigen gegenüber einem kartesischen Code

Eine Besonderheit der Implementierung in den krummlinigen PIANO Code ist, dass das ein Stretching des Gitters möglich

ist um lokale, auch anisotrope Verfeinerungen zu bekommen und dass eine Kombination mit klassischen körperangepassten Wänden oder anderen existierenden Randbedingungen möglich ist. Ein kartesischer Code kann das nicht und für Verfeinerungen stehen nur 1:2 Verfeinerungen zur Verfügung, die schnell teuer werden können. Die Initialisierung, also die Suche der virtuellen Rechenpunkte (GPs) ist hingegen trivial für kartesische Codes und teuer für krummlinige Codes. Eine Suche im Zeitbereich ist also für einen kartesischen Code denkbar, nicht jedoch für einen krummlinigen.

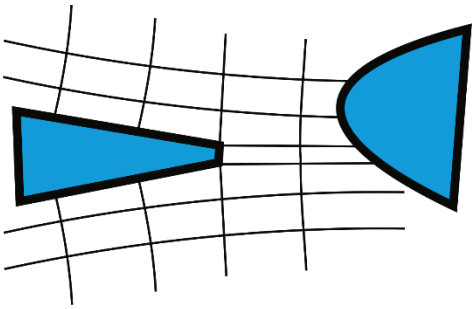


Abbildung 13: Skizzierte Möglichkeit körperangepasste Gitter mit IBC zu kombinieren.

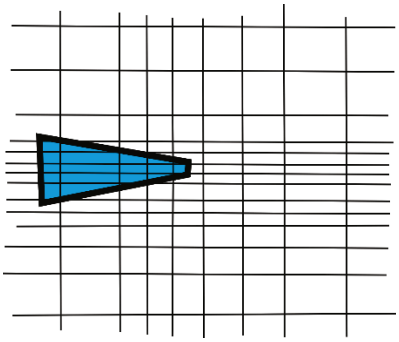


Abbildung 14: Lokale direktionale Gitterverfeinerung ohne den Zwang einer 2:1 Verfeinerung.

Zusammenfassung

Die zunehmende Notwendigkeit für einfachere Vernetzung bei großen und komplexen aeroakustischen Problemen wurde dargelegt. Die Idee und Umsetzung der „Immersed Boundary“ wurde beschrieben.

Anhand von Beispielen wurde die Nutzbarkeit für große Fälle dargestellt, was ja auch das Ziel der Implementierung war. Für akademischere Fälle zur Bestimmung der Genauigkeit sei erstmal auf [2] verwiesen.

Schlussendlich ist die IBC leicht zu handhaben und es ist leicht die Eigenschaften einzustellen. Durch das Setzen der Ghostpoint-Werte abhängig von den Spiegelpunktwerten lassen sich einfach Haft- und Nicht-Haftbedingungen erzeugen.

Literatur

[1] C. K. W. Tam and J. C. Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. J. Computational. Phys., 107:262–281, 1993. doi: 10.1006/jcph.1993.1142.

[2] Cheng Zhao, Yan Yang, Tao Zhang, Haibo Dong, Guoxiang Hou, A sharp interface immersed boundary method for flow-induced noise prediction using acoustic perturbation equations, Computers & Fluids, Volume 227, 2021, 105032, ISSN 0045-7930, https://doi.org/10.1016/j.compfluid.2021.105032.

[3] Delfs, J., Bauer, M., Ewert, R., Grogger, H., Lummer, M., and Lauke, T., Numerical simulation of aerodynamic noise with DLR’s aeroacoustic code PIANO - PIANO manual version 5.2, Braunschweig, Germany, 2007

[4] K. W. Tam, J. C. Webb, and Z. Dong. A study of the short wave components in computational acoustics. Journal of Computational Acoustics, 1:1–30, 1993.

[5] W. Hu, M. Y. Hussaini, and J. L. Manthey. Low-dissipation and low dispersion Runge-Kutta schemes for computational acoustics. Journal of Computational Physics, 124:177–191, 1996

[6] Tam, C.K.W. and Auriault, L., Jet mixing noise from fine-scale turbulence, AIAA J. Vol. 37, No. 2, February 1999.

[7] Kurbatskii, K. and C. Tam. “Multi-size-mesh multi-time-step dispersion-relation-preserving scheme for multiple-scales aeroacoustics problems.” (2002).

[8] Ewert, R., Broadband slat noise prediction based on CAA and stochastic sound sources from a fast random particle-mesh (RPM) method, Computers & Fluids, Vol. 37, Issue 4, 2008, pp. 369-387

[8] Akkermans, Rinie und Bernicke, Paul und Ewert, Roland und Dierke, Jürgen (2018) Zonal Overset-LES with stochastic volume forcing. International Journal of Heat and Fluid Flow, 70, Seiten 336-347. Elsevier. ISBN 0142-727X. ISSN 0142-727X