

# Ein hybrider Ansatz für Festigkeitsnachweise von multiskaligen Strukturen

Hans-Peter Prüfer

## Die Motivation

Für Festigkeitsnachweise hat sich die Methode der Finiten Elemente (FEM) als *Goldstandard* etabliert. Zwar wird sowohl bei der Modellbildung als auch bei der Auswertung der Ergebnisse nach wie vor eine intellektuelle Eigenleistung gefordert, die Ergebnisse selbst sind aber unter dieser Voraussetzung zuverlässig und tendenziell reproduzierbar. Dank der Leistungsfähigkeit der heutigen Arbeitsplatzrechner werden zunehmend *große* Produkte betrachtet – Assemblies, die aus einer Vielzahl unterschiedlichster Parts bzw. Baugruppen bestehen. Hier begegnen wir einem neuen Phänomen. Es gibt oft Basisstrukturen, in denen Detailstrukturen enthalten sind, deren geometrische Abmessungen sich um mehrere Größenordnungen von den Gesamtabmessungen unterscheiden können. Eine gemeinsame Elementierung erweist sich dabei als wenig sinnvoll. Ebenso findet man oft eine große Anzahl von Gleichteilen, für die im Prinzip jeweils eine *Mustervernetzung* genügt. Selbst wenn die FE-Software dies zulassen sollte, bleibt das Problem der extrem unterschiedlichen Elementgrößen innerhalb eines Modells. Das häufig propagierte *defeaturing*, für das sogar Automatisierungsansätze existieren, ist ebenso wenig zielführend, weil es auf geometrische Details bezogen ist, die nicht notwendig physikalische Funktionselemente darstellen. Gerade die physikalischen Eigenschaften der Parts sollten ja erhalten bleiben und in die Analyse einfließen. In Einzelfällen werden physikalisch motivierte Vereinfachungen praktiziert; so werden Wellen auf Balkenstrukturen reduziert, wenn man sich nur für die mechanischen Eigenschaften von Rotoren interessiert. Eine Verallgemeinerung und Systematisierung solcher Individualansätze auf größere Klassen von Strukturkomponenten ist bisher nicht untersucht worden.

## Bekannte Lösungsideen

Derartige Aufgabenstellungen zur Vereinfachung sind keineswegs neu. Schon in der Frühzeit der kommerziellen Softwarepakete kannte man das Prinzip von Substrukturen und Makroelementen (Bathe 1986). Diese Ansätze sind FEM-seitig angesiedelt. Das bedeutet, dass die Kandidaten für Substrukturen und Makroelemente erst durch den FE-Modellierer erkannt werden müssen, der in der Regel ein fertiges Bauteilmodell in einem passenden Austauschformat erhält. An dieser Stelle ließe sich einwenden, dass Entwurf und Berechnung heute dank der integrierten Systeme in einer Hand liegen und damit die konstruktiven Absichten für den Berechnungsvorgang offensichtlich sind. Dem ist allerdings entgegenzuhalten, dass komplexere Nachweise im Allgemeinen so nicht gelingen können. Sobald nichtlineare Materialeigenschaften, Kontakte oder die Berücksichtigung von Temperatureinflüssen ins Spiel kommen, ist ein Konstrukteur, der sich nicht zusätzlich auf rechnergestützte Nachweise spezialisiert hat, überfordert.

Die übliche Vorgehensweise in der FEM-Modellierung ist die Erzeugung von *Teilstrukturen*. Sie werden als Elementierung generiert. Die Nutzung innerhalb des Gesamtmodells ist dann unmittelbar durch gegebenenfalls netzinkompatiblen Einbau der Teilstruktur möglich oder indirekt, indem das Verhalten der Teilstruktur auf Kopplungsknoten – im Sinne einer *Modalanalyse* – kondensiert wird. Die indirekte modale Vorgehensweise ist die Domäne der FE-Experten, da sie doch ein tieferes Verständnis der Mechanik erfordert. Die Auswahl von Kopplungsknoten ist nichttrivial, ebenso die Beantwortung der Frage, wieviele Moden benötigt werden (Nasdala 2010). Den Einbau von Teilstrukturen in Form von Elementierungen ist demgegenüber recht schlicht. Inkompatible Netze sind für die heutigen FE-Systeme kein Problem. Ein typisches Beispiel hierfür ist das Plugin GM BoltStudio für Abaqus Simulia (Wang et al. 2009). Es besteht aus einer Bibliothek, in der alle GM-eigenen Schraubenverbindungen mitsamt ihren Komponenten abgelegt sind. Wird eine Schraubenverbindung in einem CAD-Modell identifiziert, so kann mit wenigen zusätzlichen Benutzerinteraktionen eine automatische Vernetzung erzeugt und in das Modell eingefügt werden (Abb. 1). Der Vorteil dieser Vorgehensweise besteht natürlich darin, dass das FE-Modell der

Schraubenverbindung vollständig reproduzierbar wird, da hierbei keine Benutzereinflüsse mehr vorliegen. Nachteilig bleibt die manuelle Auswahl der Komponenten im CAD-Modell.

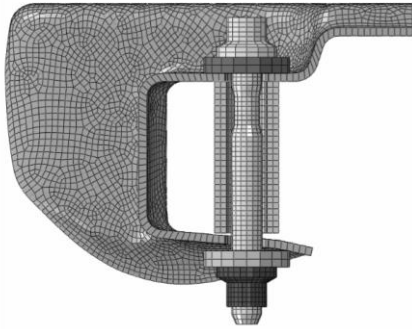


Abbildung 1: Typisches mit BoltStudio erzeugtes FE-Modell (Wang et al. 2009)

Ein anderer, dazu komplementärer Ansatz wurde im Projekt ParaFEM (Nickel et al. 2011) verfolgt. Das Hauptziel des Projekts besteht darin, die FE-Analyseergebnisse benutzerunabhängig zu machen. Bei der traditionellen Vorgehensweise ist das keineswegs gesichert, da hier die Modellierung durch FE-Experten mit durchaus unterschiedlichem Erfahrungsstand erfolgt. ParaFEM stellt eine weitgehend automatisierte FE-Umgebung dar. Die bearbeitbaren Modelle sind als Normteile und ihre Erweiterungen zu verstehen. Ein Beispiel möge dies verdeutlichen. Für Flanschberechnungen existiert eine Norm (DIN EN 1591), die prüffähige Nachweise liefert. Ihr Geltungsbereich endet jedoch bei Flanschen mit einer großen Anzahl von Schrauben, wie sie für Windenergieanlagen typisch sind. Obwohl sich die grundsätzlichen mechanischen Eigenschaften nicht signifikant ändern, kann nicht mehr nach Norm gerechnet werden, sondern man benötigt zum Nachweis eine FE-Analyse. ParaFEM soll in derartigen Grenzbereichen und Erweiterungen eine Anwendungslücke schließen, indem diese Nachrechnung, darin einer Norm entsprechend, stets die gleichen Ergebnisse liefert, was bei einer konventionellen, benutzergesteuerten FE-Analyse nur näherungsweise möglich ist (Prüfer 2018).

## Die konkretisierte Ausgangssituation

Im Rahmen der rechnergestützten Konstruktion entstehen zunehmend umfangreichere Modelle, die einen sehr hohen Detaillierungsgrad aufweisen. Sie werden üblicherweise als Assemblies von Parts erzeugt, wobei eine große Anzahl von Parts vorgefertigte Normteile bzw. Werknormteile sind, die so nicht jedesmal individuell neu erstellt werden müssen. Typische Beispiele für solche Standardparts sind Schraubenverbindungen aus Schrauben, Muttern und Unterlegscheiben, Wälzlager, kurz, die genormten Maschinenelemente, die als CAD-Bibliotheken bereitstehen (Traceparts 2019).

Für eine Analyse mittels FEM werden jedoch keineswegs stets alle Details benötigt. Möchte man beispielsweise einen Behälter nachweisen, so wird man nicht notwendig daran interessiert sein, sämtliche Schraubenverbindungen ebenfalls als kleinteilige FEM-Bereiche zu modellieren, sondern man wird auf Vereinfachungen zurückgreifen, die ausschließlich die charakteristischen Eigenschaften der Verbindung beschreiben. Im Fall von verschraubten Deckeln könnte man im Extremfall den Deckel mit dem zugehörigen Flansch durch ein *tying* verbinden oder bei etwas differenzierterer Betrachtung die Verbindung durch Kontakte und Kontaktkräfte ersetzen. Vergleichbare Überlegungen lassen sich für Getriebegehäuse anstellen. Hier ist man bei der eigentlichen Gehäusekonstruktion nicht an Details der Verzahnung interessiert, ebenso wenig interessieren Wellen und Lager. Aus dem Nachweis der Verzahnung kennt man die Lagerkräfte. Sind die Lager als genormte Parts erhalten, kennt man wiederum ihr Verhalten, und die Einleitung von Lagerkräften in eine FEM-Struktur ist eine wohlbekannte Aufgabe (Billenstein et al. 2016).

Die Partstruktur wird von den üblichen Netzgeneratoren allenfalls im Sinne der Geometrie verstanden. Zwar können die einzelnen Parts durchaus identifiziert werden, aber mehr als eine mögliche separate Vernetzung wird nicht angeboten. Sofern spezifische physikalische Eigenschaften von Parts ausgenutzt werden sollen, muss dies durch den Modellersteller geschehen. Da hierfür tiefere Kenntnisse der Mechanik erforderlich sind, ist es eine Aufgabe für Experten.

## Ein alternativer Lösungsansatz

Ein Kernproblem ist die Inkompatibilität von Geometrie- und Berechnungsmodellen. In der Theorie existieren zwar Vorschläge für Mastermodelle, aus denen die fachspezifischen Modelle abgeleitet werden können, hier stecken aber sowohl ihre Anwendung als auch die notwendige Standardisierung in den Kinderschuhen. Der hier vorgestellte Ansatz geht daher vom Geometrie-Modell aus, das als CAD-Modell in guter Qualität vorliegt.

Die Übertragung eines CAD-Modells in die FEM-Welt erfolgt traditionell über Schnittstellen. Es gibt zum einen den direkten Import aus CAD in FEM über individuell angepasste Schnittstellen, die auf die CAD-interne Datenbasis zugreifen, zum anderen die Zwischenschaltung systemneutraler Datenformate, von denen hier stellvertretend IGES und STEP genannt werden sollen. Während IGES ein reines Geometrieformat ist, kann STEP als Produktmodell aufgefasst werden. Schon in relativ einfachen Ausbaustufen enthält STEP die Partstruktur, wobei die Bezeichnungen der Parts erhalten bleiben. Sie sind damit identifizierbar.

In der Praxis wird die Tatsache, dass die Partstruktur bei Verwendung von Standardteilen implizite Informationen enthält, die über die bloße Geometrie hinausgehen und als Basis für eine weitergehende Wissensbasis dienen können, beim Datelexport meistens ignoriert. Standardteile sind funktionale Strukturen, deren Verhalten im zugehörigen Standard definiert und damit bekannt ist. Verfügt man also über eine aussagekräftige Bezeichnungskonvention, so kann man, einer Stückliste ähnlich, diese Parts als Objekte extrahieren und die innerhalb der Geometriedarstellung ausgeblendet bzw. eliminierten mechanischen Eigenschaften anhand des Standards wieder rekonstruieren. Mit diesem Wissen ist es dann möglich, externe mechanische Modelle, etwa FEM-konform, der Standardparts zu erstellen und als Teilstrukturen entsprechend den erwähnten Makros in einen FEM-Modellierer zu importieren. Die Idee dieses Vorgehens stellt sich somit als Hybrid aus dem FEM-Makrokonzept und der aus ParaFEM *geerbten* reproduzierbaren FEM-Modellbildung dar.

## Systemanforderungen

Bei der bisherigen Darstellung handelt es sich um eine ausschließlich theoretische Betrachtung. Für die konkrete Realisierung müssen zunächst die benötigten Softwarekomponenten auf ihre Eignung untersucht werden; ohne Zweifel wird es auch notwendig sein, ergänzende Module zu entwickeln.

### Geometrieerzeugung

Die vordergründig einfachste Komponente erstellt die Bauteilgeometrie. Das ist typischerweise ein CAD-System. Um den Programmieraufwand nicht ausufern zu lassen, beschränken wir uns zunächst auf den STEP-Export, der stets möglich ist. Für die CAD-Konstrukteure sind jedoch interne Richtlinien erforderlich. Der Import der Standardparts muss aus einer allgemeinverbindlichen Bibliothek erfolgen, damit in jedem Fall die gleichen Merkmale und Bezeichnungen verwendet werden. Da die Partbezeichnungen via STEP exportiert werden, kann man nutzbare Beziehungen zu einer *erweiterten* Partbibliothek herstellen. Wünschenswert ist eine Annotationsfunktion, mit der den Parts zusätzliche Merkmale zugewiesen werden können; diese müssen dann allerdings entweder ebenfalls in STEP oder in einer weiteren Exportfunktion zugänglich sein.

### FEM-Modellierung/Elementierung

Das Werkzeug, welches aus dem Geometriemodell ein FEM-Modell erzeugt, ist im üblichen Sinne der Preprozessor mit dem Netzgenerator. Dass der STEP-Import möglich sein muss, ist eine banale Forderung. Dass die Namen der Parts dabei genutzt werden können, ist weniger trivial. Alle gebräuchlichen Preprozessoren erlauben die Zerlegung in Teilvolumina, die damit auch individuell bearbeitet werden können. Darüber hinaus gehende Aktionen sind nur dann möglich, wenn der Preprozessor eine Programmierschnittstelle besitzt. Das wird meistens durch Skripte erreicht, die in softwarespezifischen eigenen Sprachen oder – was entschieden flexibler ist – in der *de-facto*-Standardsprache Python formuliert werden.

In der weiteren Vorgehensweise gibt es erhebliche Unterschiede. Unser Ziel ist es, die nicht automatisch zu elementierenden Parts in Ersatzmodule um-

zuwandeln, die dann über Kopplungsknoten an die Basisstruktur angebunden werden. Für die Bildung der Ersatzmodule ist in jedem Fall eine zusätzliche Softwarekomponente erforderlich, die mit der zu erstellenden Standardpartdatenbank/-wissensbasis zusammenarbeitet. Die daraus erzeugten Teilmodelle können dann fertige einbaubare Elementierungen oder eine Variante der so genannten Userelemente sein, letztere im Prinzip nichts anderes als Kopplungsknoten mit einer speziell definierten Übertragungsfunktion.

### Solver und Postprozessor

Im Falle des Solvers ist die Frage zu beantworten, an welcher Stelle der Einbau der Teilmodelle erfolgt. Im Falle vorgefertigter Elementierungen wird dies sicher eher im Preprocessor geschehen. Grundsätzlich ist das ein Vorgang, der beim Zusammenbau der Gesamtsteifigkeitsmatrix stattfindet. Postprocessorseitig sind die knotenbezogenen Resultate unproblematisch. Die Erzeugung von Elementdaten müsste dagegen für Userelemente nachprogrammiert werden, es sei denn, sie werden bereits solverseitig erzeugt.

### Ideen zur prototypischen Realisierung

Für einen zukünftigen industriellen Einsatz müssen die erforderlichen Softwarekomponenten in das bestehende Ökosystem passen. Kein Unternehmen wird sein CAD-System und seine FEM-Software aufgeben können. Damit ist die endgültige Softwarekonfiguration eine individuelle Anpassung.

Ein erstes, allerdings nicht abgeschlossenes Realisierungskonzept, welches in Teilen auf ParaFEM zurückgreift, nutzt als CAD-Komponente Siemens NX 7.5 aufgrund spezifischer Verfügbarkeit sowie CalculiX (Dhondt 2018), welches bereits bei ParaFEM eingesetzt wurde. Hierbei stand die zukünftige industrielle Anwendung im Primärfokus. Ferner wurde es so möglich, die in ParaFEM vorgefertigten FE-Modelle nutzen, was allerdings nur einen Teil der vorgesehenen Vereinfachungstechniken abdeckt. Als Schnittstelle zwischen den einzelnen Komponenten wurde STEP gewählt.

Für einen vorläufigen *Proof of Concept* bestehen hinsichtlich der Software keine derartigen Einschränkungen. Wir betrachten daher – auch im Sinne einfacher Verfügbarkeit – vorzugsweise quellenoffene Software. Hier bieten

sich OpenCascade (OCCT 2018) als Geometrikern sowie PythonOCC (PythonOCC 2018) als Programmierschnittstelle an. Als CAD-Komponente wollen wir zunächst FreeCAD (FreeCAD 2018) einsetzen. FreeCAD hat gegenüber anderen freien System den Vorzug, dass sogar kommerzielle Partbibliotheken ohne wesentliche Einschränkungen nutzbar sind. Die interessierenden Standardparts stehen also zur Verfügung. FEM-seitig empfiehlt sich Salome-Meca (Salome-Meca 2018). Geometriebearbeitung und Netzgenerierung sind mittels Python-Skripts programmierbar, und bei Code\_Aster handelt es sich um ein validiertes Softwareprodukt. Als FEM-Solver kommen alternativ auch noch Aurora/Z88 und CalculiX in Frage. Alle genannten Systeme sind im Übrigen auch in der Windowswelt verfügbar.

Ein typischer Workflow stellt sich wie folgt dar, wobei die erforderlichen Softwarekomponenten in eckigen Klammern benannt werden, ggf. mit Alternativvorschlägen.

- Generierung eines CAD-Modells, Export der Modelldaten [FreeCAD]
- Analyse der geometriebezogenen Modelldaten [Salome-Meca bzw. PythonOCC]
- Extraktion von Standardteilen [dito, mit zusätzlichem Skript]
- Erzeugung von Partialmodellen für die Standardteile [*Physikmodellierer*]
- Zusammenführen der Basisstruktur mit den Partialmodellen [Salome-Meca]
- FEM-Analyse [Code\_Aster, alternativ CalculiX]
- Auswertung [ParaView, mit Skript]

Bei der Voruntersuchung dieser Auswahl erwies sich übrigens PythonOCC, welches eine besonders umfangreiche Programmierschnittstelle bietet, als (noch) nicht erfolgreich installierbar, so dass nur Salome-Meca übrigblieb. Beide Systeme nutzen OpenCascade und Python, sind jedoch nicht kompatibel.

### **Ideen zur praktischen Anwendung**

Für die erste Demonstration der Realisierungsfähigkeit zeigen wir im Folgenden beispielhaft die Startschritte bei der Bearbeitung eines Flanschmodells (Abb. 2). Dazu betrachten wir einen Flansch mit Deckel einschließlich der



Schraubenverbindungen, die ihrerseits selbstverständlich aus Gleichteilen bestehen. Der Kern der Betrachtungen liegt dabei im Bereich der CAD-Generierung und den Möglichkeiten der Übergabe an die externe Softwarekomponente zur Modellreduktion. Gerade für den CAD-Teil und die nachfolgende FE-Modellierung besteht der größte Handlungsbedarf, zumal entgegen allen Ideen zur Automatisierung hier der größte Anteil manueller Aktionen erforderlich sein dürfte.

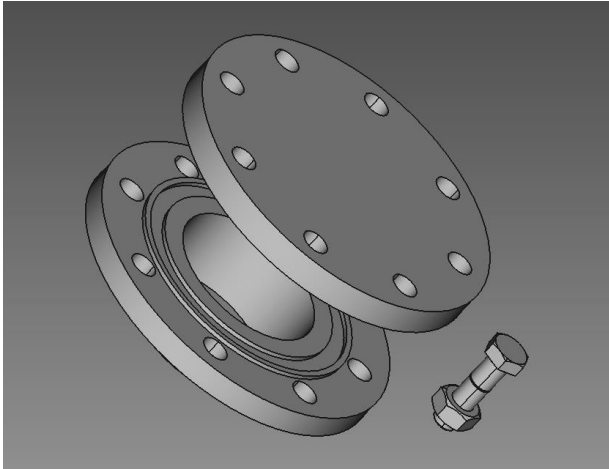


Abbildung 2: Normteile: Flansch mit Blindflansch und Verbindungselement

Das gesamte Assembly unseres Beispiels ist aus Normteilen aufgebaut. Im Strukturbaum findet man die Benennungen der Parts, die aus den Namen der importierten Dateien abgeleitet sind (Abb. 3). Wir begegnen hier der ersten Schwierigkeit. Die Gleichteile können nicht gleich benannt sein, weil sie im Sinne der Objektidentität als Teilobjekte unterscheidbar sein müssen. Eine unmittelbare Nutzung der Partnamen ist damit ausgeschlossen. Abhängig von den jeweiligen systemspezifischen Konventionen bietet es sich an, die Parts im Sinne einer Anwendungsrichtlinie bereits bei der Bauteilgenerierung mit sinnvollen Bezeichnern zu versehen. Insbesondere sollten alle Parts, die als Gleichteile mit gleichartiger Funktionalität verwendet werden, beispielsweise nach dem Schema *Basisname\_Nummer* (siehe ebenfalls Abb. 3) bezeichnet werden. Danach wird das Gesamtbauteil als STEP exportiert.

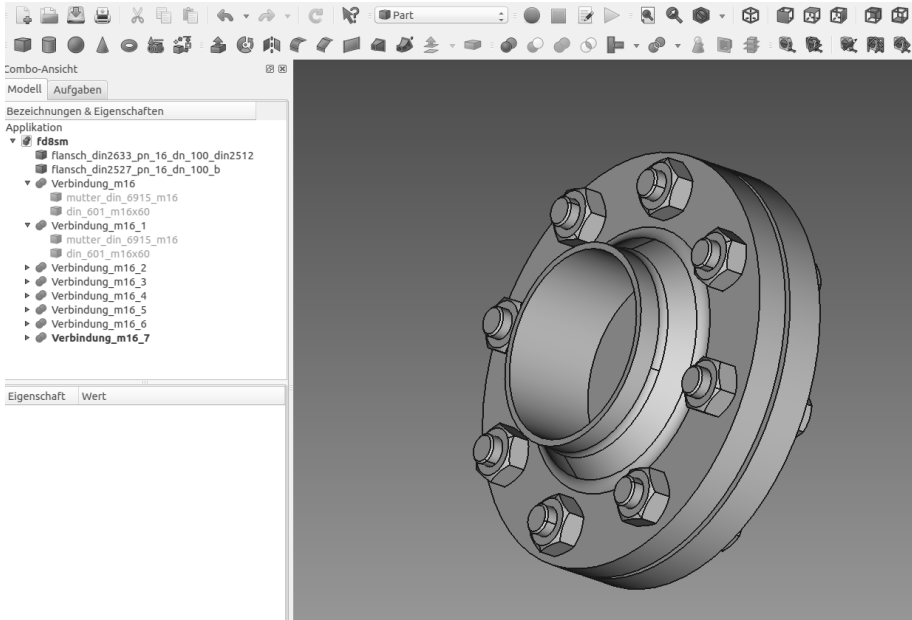


Abbildung 3: Struktur mit importierten Benennungen und Gleichteilbenennungen

Eine oberflächliche Analyse der STEP-Datei zeigt die nächste Schwierigkeit. Werden alle Standardparts separat exportiert, so muss beispielsweise der Zusammenhang von Schrauben, Muttern, Gewindebohrungen etc. *ex post* hergestellt werden. Dazu müsste man geometrische Beziehungen verwenden und würde einen Teil der Identifikation von der reinen CAD-Seite in die STEP-Analyse verlagern. Daraus ergibt sich eine weitere Richtlinie: Es ist zumindest für die erste Realisierungsphase günstiger, nicht einfach nur elementare Standardparts zu betrachten, sondern auf Funktionsstrukturen zurückzugreifen, die natürlich ihrerseits als zusammengesetzte Parts definiert sein können. Im Fall der Kombination von Schrauben und Muttern würde man diese beiden Teile jeweils zu einem Verbindungselement zusammenfassen.

Die folgenden Betrachtungen nutzen zur besseren Veranschaulichung ausschließlich die GUI von Salome-Meca. Alle hier benutzten Funktionen stehen

jedoch in gleicher Weise als Module in der zugehörigen Python API zur Verfügung.

Der Import von STEP in Salome-Meca liefert zunächst die vollständige exportierte Bauteilgeometrie als ein einziges Volumen. Die Partstruktur ist dabei nicht unmittelbar zugänglich, sondern muss aufgedeckt werden. Innerhalb von Salome-Meca gibt es hierzu die Anweisung *explode*, mit der man ein Geometrieobjekt in Unterobjekte zerlegen kann. In unserem Fall erhalten wir den Aufbau des Gesamtvolumens aus *solids*, die den Parts aus dem CAD-Modell entsprechen. Eine weitere volumenartige Zerlegung ist danach nicht mehr möglich, es können nur noch niedrigdimensionale Bestandteile gefunden werden, weil der STEP-Export der Parts aus FreeCAD keine Subparts erzeugt (Abb. 4).

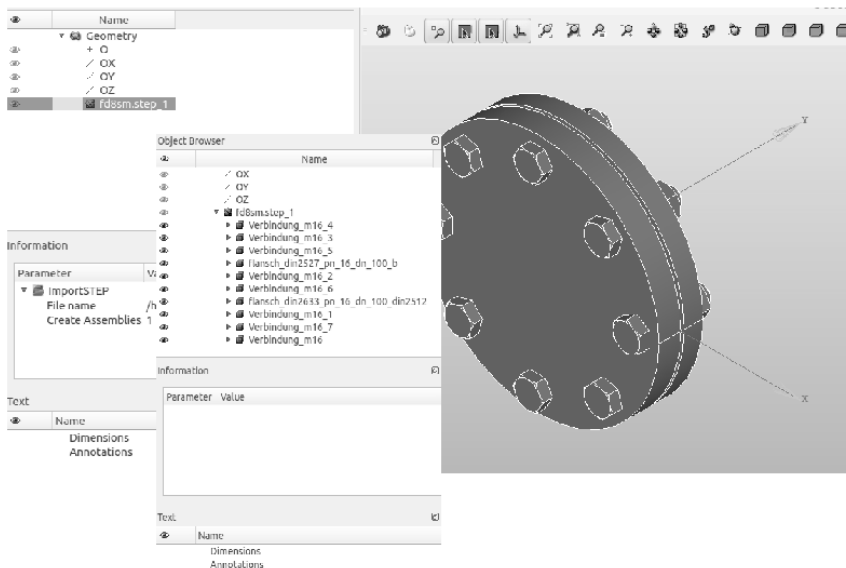


Abbildung 4: STEP-Import in Salome-Meca und Strukturbaum nach *explode*

Die Teilsolids aus dem Explodeprozess tragen die Namen, die sie während der CAD-Generierung erhalten haben. Damit sind sie eindeutig identifizierbar. Sofern die Benennung entsprechend den besprochenen Richtlinienge-

danken vergeben worden ist, können Gleichteile bzw. gleichartige Funktionsgruppen mit Hilfe eines einfachen textuellen *pattern matching* erkannt werden.

Für die Generierung eines physikalischen Modells, das als Vereinfachung in das FE-Modell eingefügt werden soll, sind weitere Informationen erforderlich. Zu jedem Volumenobjekt kann man den Schwerpunkt und die Trägheitsmomente finden (Abb. 5). Hilfreich kann dabei auch die Angabe der *bounding box* sein (Abb. 6). Prinzipiell ergeben sich daraus hinreichende Positionierungsinformationen. Die als Standardparts identifizierte Objekte können danach intern oder extern weiter verarbeitet werden. Intern wäre bei ausschließlicher Verwendung von Salome-Meca ebenso wie bei konventionellen Preprozessoren nur eine einfache Vernetzung möglich. Für die externe Bearbeitung genügt es, die Benennung und die Positionierungsdaten für jedes Exemplar dieses Typs in einer separaten Datei zu speichern, auf die dann der *Physikmodellierer* zurückgreift, da die Geometrie *a priori* aus dem Standard bekannt ist. Ein großer Teil der im Rahmen dieses Konzepts durchzuführenden Vorarbeiten betrifft deshalb selbstverständlich die Erfassung der Eigenschaften der anwendungsspezifischen Standardteile, damit diese korrekt in Makros oder Submodelle umgewandelt werden können.

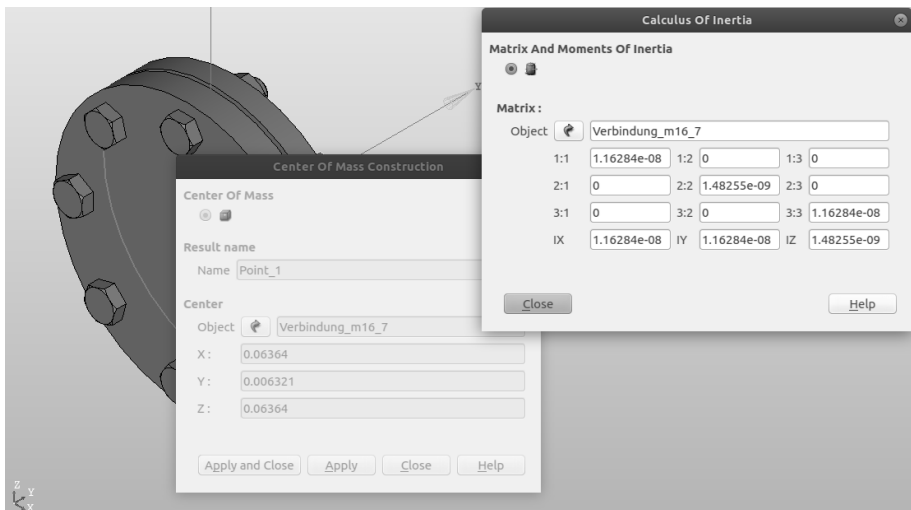


Abbildung 5: Schwerpunkt, Trägheitsmomente

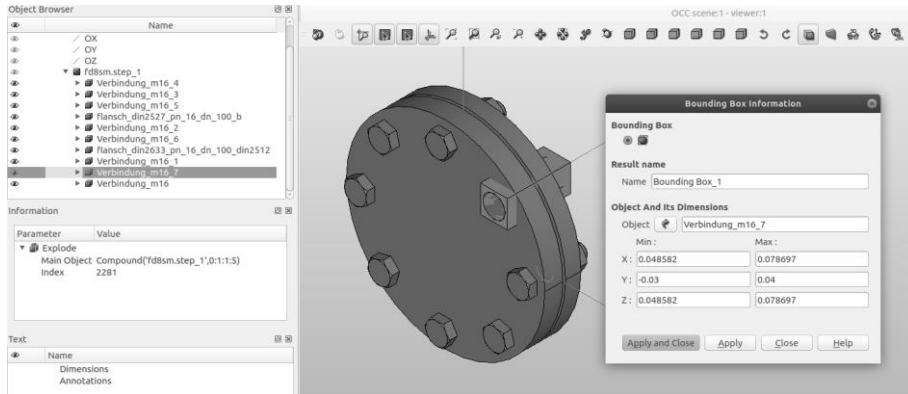


Abbildung 6: Bounding Box

Im Sinne der vereinfachten Bearbeitung multiskaliger Modelle werden selbstverständlich stark reduzierte Modelle angestrebt. So sollten beispielsweise Schraubenverbindungen und Wellen in Balkenmodelle mit ihren zugehörigen Anbindungen an die Grundstruktur umgewandelt werden. Diese spezielle Konversion ist der physikalisch-numerische Kern des vorgestellten Konzepts. Sie wird hier jedoch zunächst noch nicht weiter verfolgt, da sie eng an den genutzten Standardkatalog und die Fähigkeiten der zu nutzenden FEM-Software gebunden ist.

Schließlich sei angemerkt, dass die Extraktion der Parteieigenschaften nicht notwendig anhand eines Geometrieexports erfolgen muss. Alle gängigen CAD-Systeme verfügen über Programmierschnittstellen, mit denen die gewünschten Daten bereits im Rahmen der Geometriegenerierung erzeugt und bei Bedarf in externen Dateien abgespeichert werden können. Der Weg über STEP wurde beispielhaft gewählt, weil sich auf diese Weise die Nutzung einer systemneutralen Schnittstelle demonstrieren lässt.

## Einige abschließende Gedanken

Das hier vorgestellte Grundkonzept ist weder als endgültige Lösung noch als fertiger Softwareentwurf zu verstehen. Unsere Untersuchung ist zunächst beschränkt auf die Demonstration der Möglichkeit einer physikalisch sinnvollen Modellvereinfachung, die bereits während des CAD-Entwurfs vorbereitet wird. Sie entsteht durch die Verbindung von FE-Modellen für die interessierenden Basisstrukturen mit Ersatzmodellen für Detailkomponenten, die sich

als Übertragungsfunktionen bzw. Userelemente verstehen lassen. Diese werden in der externen Softwarekomponente *Physikmodellierer* erzeugt und als Makroelemente in die Grundstruktur zurückgekoppelt.

Mit den dargestellten Techniken können nur solche Funktionselemente behandelt werden, die sich als volumenhafte Parts beschreiben lassen. Damit ist es noch nicht möglich, Gewindesacklöcher anstelle von Muttern bei Schraubenverbindungen zu erfassen.

Die bisher keineswegs vollständig betrachteten Features gerade des Open-Cascade-Kerns eröffnen allerdings noch wesentlich weitergehende Möglichkeiten. Bei entsprechendem Programmieraufwand kann man natürlich darauf verzichten, Schrauben und Muttern zu Funktionselementen zusammenzubauen. Auf der Basis der geometrischen Attribute wie Schwerpunkte etc. lassen sich zusammengehörende Elementarparts eindeutig identifizieren. Das entscheidende Kriterium bleibt allerdings die Identifizierbarkeit der Standardteile als solche, wozu die CAD-seitige Benennung gewiss den Schlüssel darstellt. Diese Arbeit wird man dem Konstrukteur nicht ersparen können; sie wird aber erleichtert, weil der Import derartiger Teile bereits eine grundsätzliche Namenskonvention enthält.

Noch zu erstellen bleibt die Softwarekomponente *Physikmodellierer*, die die gewünschten Vereinfachungen/Reduktionen erzeugt. Sie besteht aus einem allgemeingültigen Modellierkern und einer Ausgabekomponente, die die Ersatzelemente passend zu den Konventionen des gewählten FEM-Systems erstellt. Hierbei kann teilweise auf die Erfahrungen aus ParaFEM zurückgegriffen werden.

## Dank

Ein ganz besonderer Dank gebührt meinem ehemaligen Kollegen Andreas Nickel, der maßgeblich an ParaFEM mitgearbeitet hat. Auf seine Vorarbeiten geht ein Teil des hier gezeigten Konzepts zurück.

## Literaturverzeichnis

Bathe, K. 1986: Finite-Elemente-Methoden. Berlin, Springer-Verlag

Nasdala, L. 2010: FEM-Formelsammlung Statik und Dynamik. Wiesbaden: Vieweg + Teubner

- Wang, C., Zhou, M. & Dunn, J. 2009: GM BoltStudio: A Suite of Extensions to Abaqus/CAE for Simulating Bolted Assemblies at General Motors. In: 2009 Simulia Customer Conference
- Nickel, A., Benke, K. 2011: ParaFEM – innovative Zukunftsstrategie zur effektiven Nutzung von numerischen Methoden mit integrierter Parametrisierung für Konstruktionsaufgaben. In: KT 2011, 9. Gemeinsames Kolloquium Konstruktionstechnik 2011, 06./07.10.2011, Rostock, S. 115-122, ISBN 978-3-8440-0381-9
- DIN EN 1591: Flansche und Flanschverbindungen – Regeln für die Auslegung von Flanschverbindungen mit runden Flanschen und Dichtung, 2011
- Prüfer, HP 2018: Norm oder nicht Norm – wie können FEM-Resultate verstanden werden? In: KT 2018, 16. Gemeinsames Kolloquium Konstruktionstechnik 2018, 11./12.10.2018, Bayreuth, S. 284-293, ISBN 978-3-00-059609-4
- Traceparts 2019: Bauteile-Bibliothek. Kauf- und Normteilsammlung von 3D- und 2D-Daten.  
<https://www.traceparts.com/de/>
- Billenstein, D., Diewisch, P., Dinkel, C., Rieg, F. 2016: Numerische Betrachtung von Zwangsbedingungen in der Finite Elemente Analyse. In: KT 2016, 14. Gemeinsames Kolloquium Konstruktionstechnik 2016, 06./07.10.2018, Rostock, S. 302-309, ISBN 978-3-8440-4732-5
- Dhondt, G. 2018: CalculiX CrunchiX USER'S MANUAL version 2.15. [http://www.dhondt.de/ccx\\_2.15.pdf](http://www.dhondt.de/ccx_2.15.pdf). Abgerufen 6.3.2019
- OCCT 2018: <https://www.opencascade.com/doc/occt-7.3.0/overview/html/index.html>. Abgerufen 6.3.2019
- PythonOCC 2018: <http://www.pythonocc.org>. Abgerufen 6.3.2019
- FreeCAD 2018: [https://www.freecadweb.org/wiki/Getting\\_started/de](https://www.freecadweb.org/wiki/Getting_started/de). Abgerufen 6.3.2019
- Salome-Meca 2018: <https://www.code-aster.org/spip.php?article303>. Abgerufen 6.3.2019

## Kontakt

Dr.-Ing. Dipl.-Math. Hans-Peter Prüfer  
Hasselbrinkstraße 36  
44892 Bochum

