

DISS. ETH Nr. 10143

**AN IMPLEMENTATION OF THE ADAPTIVE FINITE ELEMENT METHOD
FOR SEMICONDUCTOR SENSOR SIMULATION**

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH
for the degree of
Doctor of Technical Sciences

Presented by
Jan Gerrit Korvink
B.Eng.(RAU), M.Sc.(UCT)
born on 10 January 1961
Citizen of Great Britain

Accepted on the recommendation of
Prof. Dr. E. Anderheggen, Examiner
Prof. Dr. H. Baltes, Co-Examiner
Prof. Dr. G. Gonnet, Co-Examiner

Zürich, 1993

Kurzfassung.

Bei der Entwicklung von Sensoren können durch den Einsatz zweckmässiger und realitätsnaher numerischer Simulationen wesentliche Kosteneinsparungen erreicht werden. Mittels numerischer Modelle soll die notwendige Anzahl Durchläufe vom Prototyp bis zum funktionsfähigen Sensor reduziert werden, indem gewisse Aspekte des Verhaltens eines Sensors vorausgesagt und das Verständnis des Entwicklers für physikalische Einsichten vertieft werden, welche beim Betrieb des Sensors versteckt ablaufen.

Um diese Produktivitätssteigerung zu erzielen muß die Simulationssoftware speziellen Voraussetzungen genügen. Konkret müssen folgende Bedingungen erfüllt sein:

- ① exakte Abbildung der Sensorgeometrie im Modell;
- ② Verarbeitung großer, lokal verfeinerbarer Maschen;
- ③ lösungsadaptive Anpassung der Masche;
- ④ robuste und schnelle Lösungsalgorithmen;
- ⑤ flexible und modulare Implementation der gekoppelten physikalischen Modelle;
- ⑥ graphische Benutzeroberfläche zur Kontrolle des Rechenprozesses und zur Darstellung der Resultate.

Diese Dissertation beschreibt eine Implementation der Adaptiven Finite Element Methode, das Programm **SESSES**, um Halbleitersensoren zu simulieren. Im ersten Kapitel werden die generelle Datenstruktur und die Anforderungen an die Algorithmen des adaptiven finite Element Systems unabhängig von einer Anwendung erklärt. Zur genaueren Spezifikation der Algorithmen, werden im zweiten Kapitel die Anforderungen erläutert, welche in Zusammenhang mit den physikalischen Modellen stehen. Im dritten Kapitel sind die Anforderungen einer konkreten Applikation zusammengestellt. Hier wird die Implementation von **SESSES** detailliert erörtert. Schließlich werden im vierten Kapitel anhand illustrativer Beispiele einige Möglichkeiten dieses Programmsystems präsentiert.

Synopsis.

The development of sensor devices stands to gain a major advance in cost reduction through the use of accurate and realistic numerical simulations. The numerical models are intended to reduce the number of prototype development cycles leading up to a functional sensor, by effectively predicting certain aspects of the behaviour of the device, and by accelerating a device developer's understanding of the – normally hidden – internal mechanisms that take place during device operation.

To achieve this productivity gain a number of issues must be addressed by the software that implements the simulator. In particular, it must allow for:

- ① Effective mapping of the device geometry onto a model geometry;
- ② The ability to handle large, locally refined meshes;
- ③ Adaptation of the discrete mesh to zones where more elements are required;
- ④ Robust, fast solution algorithms;
- ⑤ Flexible and modular implementation of (new) coupled physical models;
- ⑥ A graphical user interface that implements control of the computation and visualisation of the computed results.

This dissertation describes an implementation of the adaptive finite element method, the program system **SESES**, for the purpose of semiconductor sensor simulation. The first chapter addresses the general data structure and algorithmic requirements for an adaptive finite element system – independently from the application. To narrow the algorithmic specifications, the requirements that arise from specific physical models are addressed in the second chapter. In the third chapter the constraints arising from a concrete application are determined. At this point the implementation of **SESES** is discussed in more detail. Finally, the fourth chapter presents illustrative examples which indicate some of the possibilities of the program system.