

Die Methode der Finiten Elemente.

Dr. H. Ungricht

FEM:

- ① PDE in Divergenzform
- ② Bezug zum SESES-MANUAL
- ③ Konzept der schwachen Lösung
- ④ Diskretisierung
- ⑤ Numerische Methoden
- ⑥ ein einfaches Beispiel
 - a) mit gewöhnlichen FEM
 - b) mit Box-Methode und SGSA
 - c) Verallgemeinerung von b)
→ Hybrid-Methode.

(1) Partielle Dgl. in Divergenzform

$$(1) \quad \operatorname{div} \{ \mu(f) \cdot \operatorname{grad} f \} = g(f)$$

$\underline{f} := \mu(f) \operatorname{grad} f$ Stromdichte
 f primäre Variable
 \underline{f} sekundäre "

Alle in SESES implementierten Gleichungen sind vom Typ der Gleichung (1).

(1) ist die klassische Form der Erhaltungsgesetze der Physik.

(2) SE SES - Manual

p40 : Halbleitermodell

43

Drift-Diffusionsgleichungen, van Roosbroeck 1950

Elektronenstromdichte \underline{J}_n :

$$\underline{J}_n = q \cdot \mu_n \underbrace{(U_T \operatorname{grad} n - n \cdot \operatorname{grad} U)}_{\text{Diffusions-} \quad \text{Driftterm.}}$$

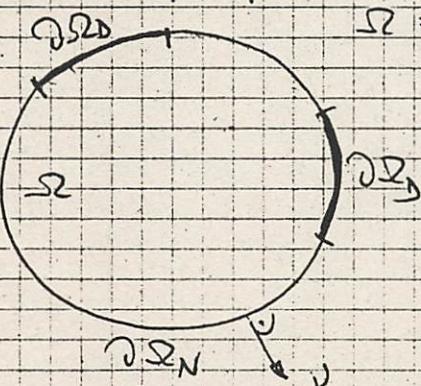
Diffusions- Driftterm.

Werden anstatt der physikalischen Variablen (U, n, p) die mathematischen Variablen (U, u, v) verwendet:

$$\left. \begin{array}{l} n = n; U_T e^{U/U_T} \\ p = n; U_T e^{-U/U_T} \end{array} \right\} \rightarrow \underline{J}_n = q \mu_n U_T n; e^{U/U_T} \operatorname{grad} u.$$

ohne Driftterm.

Randbedingungen



Ω = Simulationsgebiert $\Omega \subset \mathbb{R}^n$ $n=2$ oder 3

1) $\underline{\partial}\Omega_D$: f gegeben

$$\underline{\partial}\Omega_N = \underline{\partial}\Omega \setminus \underline{\partial}\Omega_D$$

2) $\underline{\underline{f}} \cdot \underline{\underline{n}} = 0$ auf $\underline{\partial}\Omega_N$.

$\underline{\underline{n}}$ = Einheitsnormalenvektor nach aussen
 in SESES sind allgemeinere RB möglich.

(3) Konzept der schwachen Lösung

Klassische Lösung: $\int \text{div} \{\mu(f) \cdot \text{grad } f\} = g(f) \quad \text{in } \Omega$

$$(1) \quad \left. \begin{array}{l} f \text{ gegeben auf } \partial \Omega_D \\ f \cdot v = 0 \text{ auf } \partial \Omega_N \end{array} \right\}$$

Schwache Lösung: $T = \text{Testraum von Funktionen}$
geeignet zu wählen, RB.

$$(2) \quad \left. \begin{array}{l} f \in T \\ \int_{\Omega} v \cdot \text{div} \{\mu(f) \cdot \text{grad } f\} d\Omega = \int_{\Omega} g(f) \cdot v d\Omega \end{array} \right\} \quad \text{für alle } v \in T$$

Bemerkungen:

1) (2) heißt schwache Formulierung von (1)

Lösung von (2) heißt schwache Lösung von (1)

Falls f eine schwache Lsg von (1) ist, ist es nicht klar ob
 f auch eine klassische Lsg von (1) darstellt.

f schwache Lsg $\xrightarrow{?} f$ klassische Lsg

Regularitätsbedingungen
für die schwache Lsg

Vorteil: Für (2) ist es einfacher die Existenz einer Lösung
zu beweisen, was für (1) relativ schwierig ist.

2) Zum Beweis der Existenz einer klassischen Lsg von (1) beginnt man
gewöhnlich mit einer schwachen Lsg von (1) und zeigt mit
 beachtlichem Aufwand, dass diese Lsg genügend regulär ist, um
eine klassische Lsg zu sein.

3) Nicht-lineare Probleme

Existenz einer klassischen Lsg zu beweisen ist extrem schwierig oder
praktisch unmöglich, liegegen liegt es immer noch im Bereich
des Möglichen die Existenz einer schwachen Lsg zu zeigen.

PDE ohne exakte Lsg.

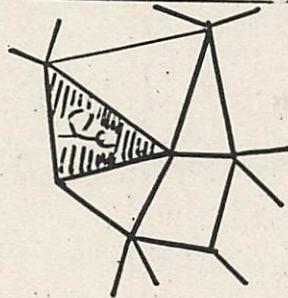
FEM

Dgl.
für u

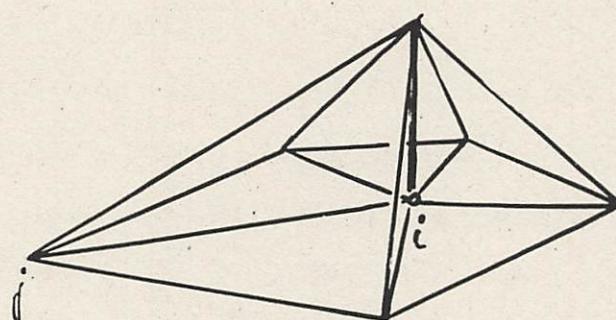
$\longleftrightarrow \left\{ \begin{array}{l} \text{Ritz: Variationsproblem} \\ \text{Galerkin: gewichtete} \\ \text{Residuen.} \end{array} \right.$

- ① Wahl von Ausatzfunktionen h_i $u_a = \sum_{i=1}^N u_i h_i$
 → System von N diskreten Gln.

- ② Zerlegung des Simulationsgebiets Ω in Elemente Ω_e



Ausätze h_i so, dass auf dem größten Teil des Simulationsgebiets: $h_i \equiv 0$.



h_i : linear

$$h_i(j) = \delta_{ij}$$

Ziel:

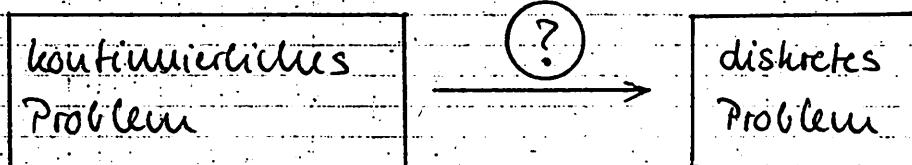
- a) h_i so, dass Rechenaufwand minimal
 b) h_i so, dass $u_a \xrightarrow[N \rightarrow \infty]{} u$

4.) Diskretisierung

Das gegebene Problem kann nicht analytisch gelöst werden.

Daher: numerische Approximation.

Diese Approximation macht Gebrauch vom Konzept der schwachen Lösung.



i. allg. nicht-lineares
Alg. System $Ax = b$, das linearisiert wird.

A: Jacobi-Matrix

b: RB

Für diesen Übergang gibt es verschiedene Methoden:

1) Boxmethode (control volume, finite volume method)

2) F.D. finite Differenzen:

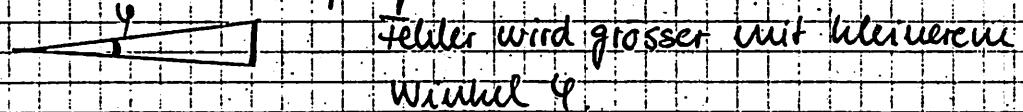
lineare Approximation für den Differentialoperator.

für die Kontinuitätsgleichung wird angepasst, da die Ladungsträgerdichte exponentiell variiert.

3) FEM finite element Methode (klassisch)

Auflösung der PDE wird ein äquivalentes Variationsproblem betrachtet. Das Simulationsgeliefert Ω wird in Elemente Ω_e zerlegt. Die Lsg des Variationsproblems wird approximiert durch Lsg des Variationsproblems für die einzelnen Elemente Ω_e . Die Ansatzfkt. u_i haben nur lokale Träger.

Vorteil: keine harte Einschränkung, vgl. Box-Methode
nur weiche Bedingungen:



Nachteil: interpolierende Fkt., Ansätze linear, quadratisch
schlecht geeignet, nur exponentiell variiierende
Ladungsträgerkonzentration nahezu bilden.
→ exponentielle Gewichtung SG

(5)

Numerische Methoden für PDE in Divergenzform

Methode der gewichteten Residuen:

Die gesuchte Funktion wird mit Hilfe von linear unabhängigen Ausatzfunktionen h_1, h_2, \dots, h_N approximiert:

$$(2) \quad \text{Ausatz} \quad f_a = \sum_{k=1}^N f_k \cdot h_k$$

gesucht: Koeffizienten f_1, f_2, \dots, f_N der Entwicklung (2)

Einsetzen von (2) in (1) liefert i. allg. ein Residuum.

Forderung: dieses Residuum soll im Inneren von Ω möglichst klein werden.

$$\int_{\Omega} G_j \left\{ \operatorname{div} \{ \mu(f_a) \operatorname{grad} f_a \} - g(f_a) \right\} d\Omega = 0$$

$j = 1, 2, \dots, N$ Gleichungen.

- dh. das Integral des Residuums, gewichtet mit einer Gewichtsfkt. G_j , über das Grundgebiet soll verschwinden.
- Da der Ausatz (2) N Parameter f_1, \dots, f_N enthält, kann diese Bedingung für N voneinander linear unabhängige Gewichtsfkt. G_j formuliert werden.
 $\rightarrow N$ Gleichungen für f_1, f_2, \dots, f_N .

Fragen:

- Wahl der G_j , h_i
- numerische Integration (Anzahl der Pkt., Wahl der Pkt.)

Methode von Galerkin:

$$G_j = h_j$$

Mit dieser Wahl der G_j wird erreicht, dass die Residuenfunktion orthogonal zum Funktionsraum ist, der durch h_1, \dots, h_N aufgespannt ist. Die resultierende Näherungslösung stellt in diesem Sinn die best mögliche Näherung im Raum der Ausatzfunktionen dar.

4-u-v Modell

$$\left\{ \begin{array}{l} \operatorname{div} \underline{J}_n = q R \\ \underline{J}_n = \boxed{q \cdot \mu_n \cdot u_T n_i e^{\frac{q u_T}{k T}}}; \operatorname{grad} u \end{array} \right.$$

stationäre Kontinuitätsgleichung für die Elektronenstromdichte:

$$\int_{\Omega} G_j [\operatorname{div} \underline{J}_n(u_a) - q \cdot R] d\Omega = 0$$

$j = 1, 2, \dots, N$ Gleichungen

Ausatz: $u_a = \sum_{i=1}^N u_i l t_i$ $l t_i = l t_i(x, y, z)$

$$\underline{J}_n = \boxed{a(x, y, z)}; \operatorname{grad} u_a$$

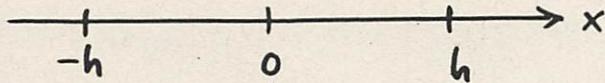
weitere Fragen - Approximation von R bzw. g

- Approximation der Differentialoperatoren
 div , grad (analyt, numer.)

4-u-v Modell

stationäre Kontinuitätsgleichung für Elektronenstromdichte

Bsp. 1-dim



Gleichung nach Vereinfachungen ($R=0$, $\mu_n=1$)
einzige Sicherheit: Faktor e^{ψ}

$$\frac{d}{dx} \left(e^{\psi} \frac{du}{dx} \right) = 0$$

$$RB: u(-h) = 0$$

$$u(h) = 1$$

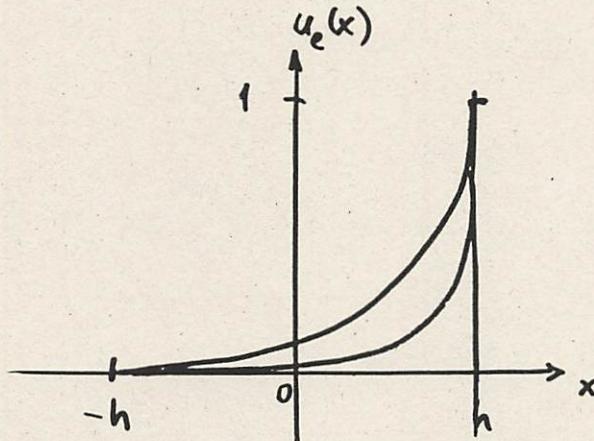
elekt. Potential $\psi(x) = \alpha \cdot x$

exakte Lösung:

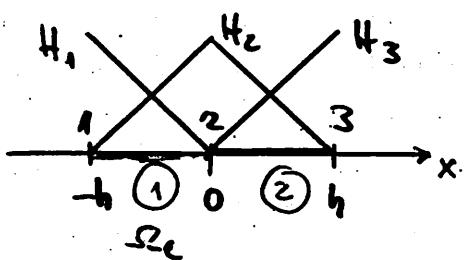
$$u_e(x) = \frac{e^{\alpha h} - e^{-\alpha x}}{e^{\alpha h} - e^{-\alpha h}}$$

Strom:

$$J_n = e^{\psi} \frac{du}{dx} = \frac{\alpha \cdot e^{\alpha h}}{e^{2\alpha h} - 1}$$



FEM:



$$\Omega = [-h, h]$$

Ausdruck für $u(x)$: $u_a(x) = u_1 H_1(x) + u_2 H_2(x) + u_3 H_3(x)$

$$RB: u_1 = 0 \quad u_3 = 1$$

Bestimmung von u_2 :

(Methode der gewichteten Residuen)

$$\int_{-h}^h H_2(x) (e^\alpha u_a'(x))' dx = 0 \Leftrightarrow \boxed{\int_{-h}^h H_2'(x) e^\alpha u_a'(x) dx = 0}$$

$$\Rightarrow u_2 = \frac{e^{\alpha h}}{e^{\alpha h} + 1} \quad \text{exakter Wert.}$$

Residuum links (= Strom im Knoten 1):

rechts

3

$$R_L = \int_{-h}^0 H_1'(x) J_n^{(1)} dx = \left(-\frac{1}{\alpha h^2}\right) \cdot \left(\frac{e^{\alpha h} - 1}{e^{\alpha h} + 1}\right)^2$$

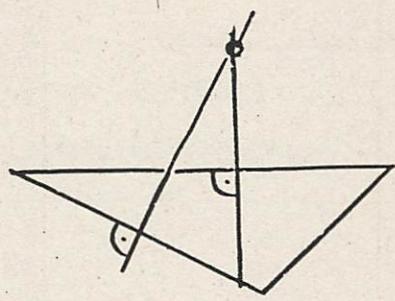
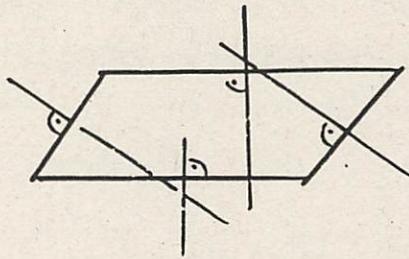
$$R_T = \int_0^h H_3'(x) J_n^{(2)} dx = \left(+\frac{1}{\alpha h^2}\right) \cdot \left(\frac{e^{\alpha h} - 1}{e^{\alpha h} + 1}\right)^2$$

Fehler (Knoten 3)

$$J_n^{(2)} - R_T = \frac{\frac{2}{\alpha} - \frac{e^{-\alpha h}}{e^{\alpha h} - 1}}{\frac{e^{\alpha h} - 1}{e^{\alpha h} + 1}} \cdot \left(\frac{h}{2} + h.o.t(\alpha, h)\right)$$

Frage: 1) Wie gross muss h sein bei gegebenem α , damit der Fehler $< \delta$?

2) Gibt es eine Diskretisierung, die dieses Bsp. mit 2 Elementen exakt nachbildet? (Knotenwert, Strom)



(2) Geometrie;

$$\int_{\Omega} \phi \bar{u} d\omega - \int_{\Omega} q R d\omega = \sum_{k=1}^n u_k j_k - \int_{\Omega} q R d\omega$$

Nutzfläche

1) Annahme: Show Nutzfläche Δu_k Boxfläche j_k !!

Show von x_i nutzt x_i : $\int_{\Omega} u_i \bar{u}_i d\omega = \int_{\Omega}$

Boxfläche mit SGS:

Klassische Boxmethode

$$\int_{\Omega} \bar{u} d\omega - \int_{\Omega} q R d\omega = 0$$

$$Grenz: \int_{\Omega} G_i d\omega \int_{\Omega} d\omega - \int_{\Omega} q R d\omega + \int_{\Omega} \bar{u} d\omega = 0$$

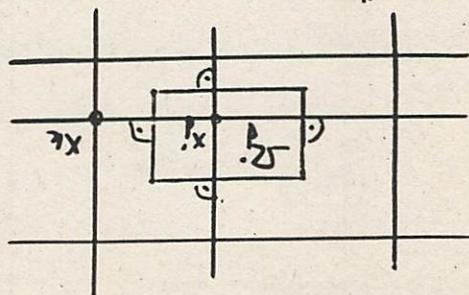
$$0 = \int_{\Omega} G_i (d\omega \int_{\Omega} \bar{u} d\omega - q \cdot R) d\omega$$

Wahl von G_i : $G_i = \begin{cases} 0 & \text{außerhalb} \\ 1 & \text{in } \Omega \end{cases}$

Stützfläche zu Elementflächen

Abstand der Fläche Null.

$$G_i = \text{Box um Pkt } x_i$$



Boxmethode (Varga 1962)

Schafffetter-Gummel-Shour Approximation SGSA (1969)
1-dimensional

Ann. \underline{J}_n elementweise konstant (auch wenn $R \neq 0$)

ψ " linear

μ_n konstant

(auch wenn $\mu_n \neq$ konst.)

$$\underline{J}_n = e^{\alpha} \frac{du}{dx} \text{ analytisch integrierbar}$$

$$u(x) = -\frac{1}{\alpha} e^{-\alpha x} \cdot \underline{J}_n + C_0$$

\underline{J}_n und C_0 eindeutig durch Knotenwerte u_i, u_{i+1} bestimmt.

$$\begin{array}{l} (1) \\ \left\{ \begin{array}{l} u_2 = -\frac{1}{\alpha} \underline{J}_n + C_0 \\ u_1 = -\frac{1}{\alpha} e^{-\alpha h} \underline{J}_n + C_0 \end{array} \right. \end{array}$$

$$\Rightarrow \underline{J}_n^{(1)}(u_2, u_1) = \frac{\alpha(u_2 - u_1)}{e^{\alpha h} - 1}$$

$$\begin{array}{l} (2) \\ \left\{ \begin{array}{l} u_3 = -\frac{1}{\alpha} e^{-\alpha h} \underline{J}_n + C_0 \\ u_2 = -\frac{1}{\alpha} \underline{J}_n + C_0 \end{array} \right. \end{array}$$

$$\Rightarrow \underline{J}_n^{(2)}(u_3, u_2) = \frac{\alpha(u_3 - u_2)}{1 - e^{-\alpha h}}$$

Globale Gleichungen

Strombilanz (je 2 aneinanderstoßende Elemente)

$$\underline{J}_n^{(1)} = \underline{J}_n^{(2)} \Rightarrow u_2 = \frac{e^{\alpha h}}{e^{\alpha h} + 1}$$

exakte
Werte !!

$$\Rightarrow \underline{J}_n^{(1)} = \underline{J}_n^{(2)} = \frac{\alpha \cdot e^{\alpha h}}{e^{2\alpha h} - 1}$$

1-dimensional ist diese Methode Spitzenerreiter
alle übrigen Methoden werden daran gemessen.

Hybrides Elementmodell (Pian 1964)

allg.: Kombination von Gebiets- und Randmethode,
mehrere Lösungsausätze gleichzeitig

in jedem Element Δ_e :

$$\text{Ausatz für } \underline{J}_n: \quad \underline{J}_a = \sum_{j=1}^M \beta_j \cdot \underline{C}_j \cdot (x, y, z)$$

$$\text{Ausatz für } u: \quad u_a = \sum_{i=1}^N u_i \cdot h_i(x, y, z)$$

Kompatibilität von \underline{J}_a mit u_a in Δ_e nicht gewährleistet,
da beide Ausätze unabhängig voneinander verwendet werden.

I Lokal Stromkompatibilität in Δ_e :

$$\int_{\partial \Delta_e} \underline{W}_j \cdot (\underline{J}_a - \underline{J}_n(u_a)) d\mathcal{L} = 0 \quad j=1, \dots, M$$

$$\text{Gewicht } \underline{W}_j = \begin{bmatrix} \bar{a}'(x, y, z) \\ \underline{C}_j \cdot (x, y, z) \\ \bar{a}'(x, y, z) \neq 0 \end{bmatrix}$$

$$a(x, y, z) = q \mu_n U_T \vec{n}_i \cdot \vec{e}^{+4U_T} \quad \text{Gewichtung}$$

\underline{J}_a nicht stetig über $\partial \Delta_e$ hinweg.

II global Strombilanz:

$$\int_{\partial \Omega} H_i \cdot (\text{div } \underline{J}_a - qR) d\mathcal{L} = \sum_e \oint_{\partial \Delta_e} H_i \cdot \underline{v} \cdot \underline{J}_a dw - \int_{\Omega} H_i q R d\Omega$$

Bem.: Dieses Modell ist im 1-dim Fall mit den entsprechenden Voraussetzungen mit SGSA äquivalent!

Physik:

S. M. Sze

Physics of Semiconductor Devices
2nd Edition, J. Wiley 1981

S. M. Sze

Semiconductor Devices

Physics and Technology, J. Wiley 1985

A. S. Grove

Physics and Technology of Semiconductor Devices

J. Wiley 1967

J. D. Jackson

Classical Electrodynamics, 2nd Edition, J. Wiley 1975

Mathematik

P. A. Markowich, Chr. A. Ringhofer, Chr. Schmeiser

Semiconductor Equations, Springer 1990

P. A. Markowich

The stationary Semiconductor Device Equations

Springer 1986

M. S. Mock

Analysis of Mathematical Models of Semiconductor
Devices, Boole Press Dublin, 1983

S. Sellerkell

Analysis and Simulation of Semiconductor

Devices, Springer 1984

FEM

H. R. Schwarz

Methode der finiten Elemente, Teubner 1980

G. Strang, G. Fix

An Analysis of the FEM, Prentice Hall 1973

K. J. Bathe

Finite Element Methoden, Springer 1986

O. C. Zienkiewicz

The finite Element Method, 3rd Edition, McGrawHill 1977

O. C. Zienkiewicz, K. Morgan

Finite Elements and Approximations

J. Wiley 1983

Funktionsweise von SESES.

Dipl. phys. G. Sartoris

SESES Functionality (concise summary)

SESES provides 2D and 3D programs to solve one or more stationary elliptic partial differential equations of second order. Different program versions are available, each pair solving one specific problem:

- Poisson or Laplace equation for electrostatics (p2d, p3d)
- Heat transport equation with non linear diffusion term (t2d, t3d)
- Semiconductor drift-diffusion (Van Roosbroeck) equations extended to include magnetic field (Galvanometric) effects (f2d, f3d)
- Magnetic field computations in 2 dimension (v2d)
- Electric current continuity equation with magnetic field effects (r2d, r3d)

SESES is a Finite Element (FE) program. The discretization scheme used by any FE method is based on a partitioning of the simulation domain in subsets (called elements) whose union form the original domain and whose interiors are disjoint. Due to the high complexity of this task for 3D domains when using mixed shape element forms we have limited the choice to only topological equivalent brick shape. In a first step the user has to define a Macro Element (ME) mesh as tensor product of one dimensional (irregular) meshes with the possibility of shifting each ME z-joint coordinate. This allows us to properly model the layer structure arising in semiconductor manufacturing, less flexible is the modeling of shapes in the x-y plane. On the ME mesh the user defines material contacts and builds up a finer mesh in order to correctly compute a solution. Of course the ME mesh can also be used as a computing mesh. For refinement each single FE can be recursively subdivided in eight(four) new subelements such that neighbouring elements do not differ by more than one refinement level. This operation can produce joints in the middle of faces(edges), so that in order to assure C0-continuity of the solution on faces(edges) linear constraint equations have to be introduced. These set the value of the middle face(edge) joint to be the linear interpolation of the face(edge) joints values. The refinement is performed automatically by defining a ME subdomain region and an indicator function; on large or small values of the indicator the element is refined or unrefined.

SESES provides two program types:

- a) A computational kernel (one for each version: f2d f3d p2d p3d ...) to compute solutions, running in batch or interactive mode and using only the most standard IO and mathematical library functions. The interactive mode is mostly used to control long jobs, to build meshes or when experimenting with trial and error techniques.
- b) An X-Window based front end (mg2d mg3d) for pre and post-processing graphics and kernel monitoring.

Two files are needed to run a full simulation session:

- a) An <Initial> file specifying the device (geometry, materials, doping, ...). After parsing the file, the kernel builds the ME mesh and initializes all data structures and is ready to compute a solution.
- b) A <Command> file read after the <Initial> file, containing set commands specifying default set up and solution algorithms to be used and do commands specifying some actions to be performed. From this file the kernel builds up a work table, and begins sequentially the execution of each stored command:

```
Set LinearSolver Direct
Set Refine zone0 AllME
Do Refine zone0 Manually 1
Do Solve Stationary AtTime 1.0
```

When working in interactive mode, the user selects in the monitor of the front-end one or more commands by simply clicking on various menus describing each possible Set or Do command. As result a small command file is exchanged between the front-end and the kernel, the kernel in turn interrupt its work, reads the newly received command file and updates the worktable. Operations such as delete, replace, insert and append are available for the worktable after every executed command and linear solution step. In this way the user has at any time the possibility to display the actual solution state of the kernel, and has full control of it. The true sequence of executed commands is always available by the command:

```
Do ShowLogFile
```

Even if SESES is at present only solve stationary problems, we have introduced the concept of time. This is simply an internal variable of the kernel which can be formally referenced when defining time variables and time functions in the initial file.

```
TimeFunction t2 (time*time)
TimeFunction t3 t2*time
MagneticField Bx time By t2 Bz t3 Tesla
```

Contact values, material coefficients, magnetic field, (initial field values, ...) can all be defined to be time dependent. The time value is set by the user when a stationary solution is started and easily allows to compute a series of solutions with different parameter values.

```
Do Solve Stationary AtTime 1.0 StepLength 1.0 ToTime 10.0 .
```

When a series of solutions have to be computed it is useful (to reducing the running time) to produce a good guess for the start solution from the previously computed ones. Lagrange extrapolation of maximal fifth order is available in SESES where the order automatically determines the number of solutions to be retained by the kernel.

```
Set Extrapolation <doffieldname> <order:Linear Quadratic ... Quintic>
<State> Bulk files containing time and solution values within <Element> Bulk files describing the mesh fully determines the state of a SESES simulation. Loading of the <Element> and one or more <State> files allows exact recovering of previous kernel states:
```

```
Do Bulk Write Element File Mesh
Do Bulk Write State File State0
...
Do Bulk Read Element File Mesh
Do Bulk Read State File State0
```

<Field> Bulk files are used to display the specified field by the front end program. The ascii option can eventually be used to assure portability when one has to generate files that have to be transferred between computers with different binary representations. This however is becoming an uncommon operation today since the advent of X-window based user graphics interfaces which allow displaying of data on computers other than the one computing.

```
Do Bulk Write Field <fieldname: Psi, E-Field, Jn-Current, Res-I, ...>
```

Data on field values is also available in ascii formatted form. The user has first to define a regular lattice of points over the domain where the field has to be computed.

```
Set Lattice lattice0 Line Coordinates 0. 0. 0. 1. 1. 1. 10
Do Write lattice0 <fieldname: Psi, E-Field, Jn-Current, Res-I, ...>
```

Before computing a solution (but also in afterwards) the user has to define a good mesh in order to properly compute a numerical solution. First step is to define a region where refinement has to take place, secondly a manual or an adaptive refinement technique is selected. By the first one it is intended that all elements in the region are refined by one level, by the second an error indicator is evaluated on each element of the region, and refinement takes place only on those elements having an indicator value higher than a given threshold. The same argument exactly applies to unrefinement techniques.

```
Set Refine zone1 AllME  
Set Refine zone2 Indices 0 1 0 2 0 3  
Do Refine zone1 Manually 1  
Do Refine zone2 Adaptive 1 Statistical 0.0 IndGradPsi
```

A special talk is devoted to SESES functionality when solving the full set of equations so that we will not treat this issue here.

SESES runs concurrently on more than one host. The hosts have to have the same binary format representation and run a BSD–Unix system or a similar defining sockets (abstract objects from which data can be send and received over the network). The distributed network topology is a stellar one with one master host running the usual SESES kernel connected by a full–duplex connection to the slave hosts running a reduced kernel program stripped of all unnecessary functions. Actually only the element assembling which takes the most time for 3D simulations has been parallelized. Due to the small bandwidth of local area network links (10Mbits/s for Ethernet) results on achieved speed–up are not excellent but somehow good. Better speed–up will be obtained by parallelizing the linear iterative solver.

Errors are handled differently in batch and interactive mode. Numerical errors (division by zero or some floating over/underflow) are handled as severe ones and always kill the kernel. This avoids the expensive checking of values and in general only appears when the solution algorithm has already begun to diverge so that these kills are not critical. Non-converging solution strategies (maximal number of iterations exceeded) or insufficient memory have the result in interactive mode of aborting the executing command and proceeding with the next one, in batch mode the kernel is exited and a positive value is returned to the shell program. This value in turn can be used to develop shell programs for trial and error and optimizing algorithms.

Problembeschreibung, Initial-, Bulk-Files.

Dipl. Ing. J. Korvink

Übung I: Lineare Probleme

Uebung 1

PRELIMINARY: SESES FILE STRUCTURE

A Seses simulation is governed by four file types:

One Initial file:

The Initial file contains a problem description for a KERNEL and FRONT END program:

 Mesh geometry

 Fields

 Boundary conditions

 Material properties

The user has to type in an Initial file using an ascii EDITOR program.

One or more Command files:

Each Command file contains instructions to a KERNEL:

 Name of the Initial file

 Solution strategy

 Refinement strategy

 Postprocessing instructions

The user has to type in a Command file using an ascii EDITOR program, OR generate it with a FRONT END program.

An optional SesesSetUp file:

A SesesSetUp file defines individual startup defaults for a FRONT END program:

 Graphics window size

 Colours used in the graphical display

 Menu button choices

 Printer type, etc

The user has to type in a SesesSetUp file using an ascii EDITOR program, OR generate it with a FRONT END program (the second choice is advisable).

Any number of Bulk files:

Bulk files communicate simulation data between the KERNEL and the FRONT END:

The refined mesh's finite elements

The mesh solution vector incidences which connects mesh elements with the corner joints

The joint field vectors

The element field vectors

An instantaneous solution state

The KERNEL produces bulk files when instructed by a Command file instruction. The FRONT END reads and displays the contents of Bulk files. The KERNEL reads Bulk files to enable a computation restart.

UEBUNG 1

In the next section, we are going to produce the files necessary for a linear simulation problem. We will also have the opportunity to use each of the SESES programs, to familiarise ourselves with the functionality of SESES. In particular, we will:

Type in an Initial problem description file

Display and check the Initial file with the FRONT END

Type in a KERNEL Command file

Do a batch simulation with the KERNEL Command file

Later on, we will:

Interactively create and execute a KERNEL Command file in an interactive simulation session

Display the simulation result Bulk files with the FRONT END

Automatically create a SesesSetUp file

Produce a PostScript (tm) graphics output file with the FRONT END, and display (and print) it with another program

a) Problem statement

Investigate the interconnect capacitance of two crossing conductors realised with a 3 μ CMOS process. The process results in a layer structure, given in the table below, and starting from the silicon substrate.

Process Layers:

Material	Thickness	Permittivity
Silicon substrate	500.00 μ	11.7
SiO ₂	0.05 μ	3.9
Polysilicon	0.3 μ	NA
SiO ₂	0.035 μ	3.9
Aluminium	1.1 μ	NA
SiO ₂	1.0 μ	3.9
Air	10.0 μ	1.0

Device Layout:

See the figure attached at the end.

b) Problem specification in the Initial file:

It is assumed that you are already logged on. Edit the Initial file:

Invoke the text editor using the right mouse button with the mouse pointer placed anywhere on the background panel. Select the File popdown and then the Load File option. Type the Initial file's title: InitialUebung1

OR

On the command line, type "e" followed by the filename and a return: e InitialUebung1

The text editor will appear on the screen, displaying the contents of the file and ready to accept typed input. Simply move the mouse pointer into the text editor window. You have been given an Initial file syntax diagram. The Initial file conforms to this diagram. Follow the instructions in the Initial file's comments, which are bounded by (*) and *).

c) Solution instructions in the Command file:

Use the editor to open the file entitled "CommandUebung1". Analogously to the Initial file, we follow the syntax of the Command file for the file statements.

d) Solution by batch processing:

To start the electrostatic computational KERNEL p3d, we type its name followed by the Command file name, and press return:

p3d CommandUebung1

The KERNEL program echos to the user the steps that it takes. View the resulting bulk files with the interactive FRONT END. In particular, have a look at where the error indicator IndDelta has shown the mesh to be insufficiently refined.

Take some time to alter the "Intial" file, e.g. the device dimensions as well as the number of ME's used. Recompute the solution and observe the effect that this has on the capacitance.

e) Solution by interactive processing:

For an interactive SESES simulation session, we start the FRONT END as usual:

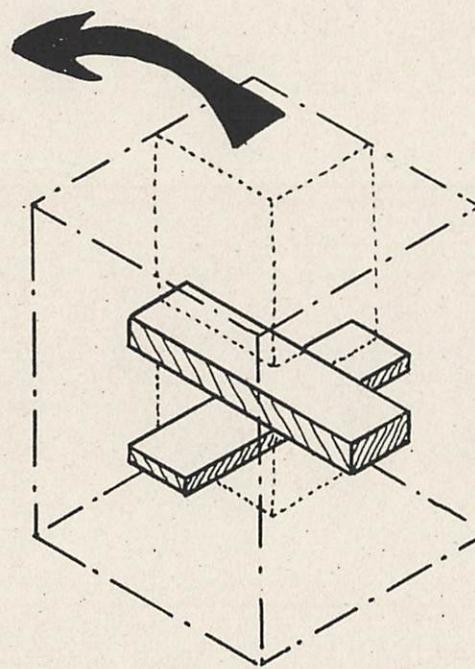
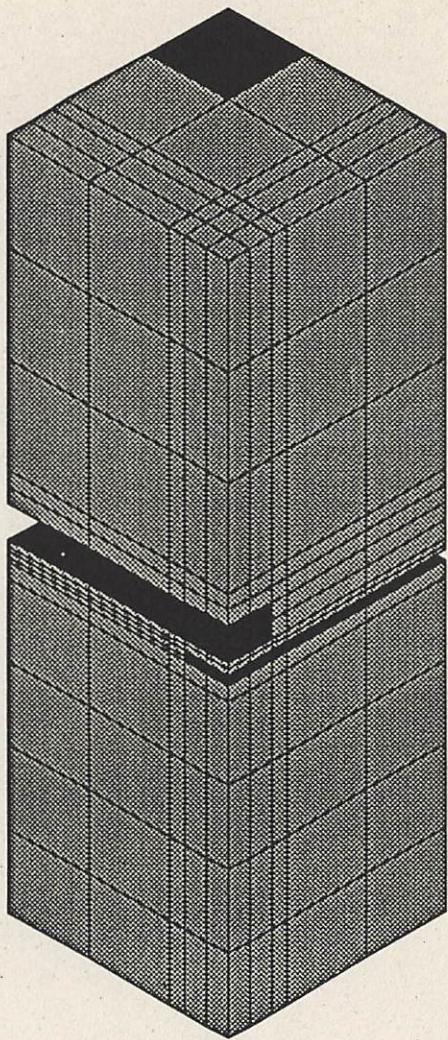
mg3d InitialUebung1

We select the MONITOR button on the top menu. This opens a horizontal area with many buttons. Select the correct KERNEL program on the top menu bar - POISSON. Now start the KERNEL with the START KERNEL button. Look at the shell window for an effect. The KERNEL will echo some data as it reads the Initial file. The KERNEL now lies waiting dormant, ready to execute commands.

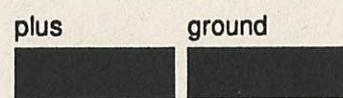
Further instructions are given during the lecture.

Copyright by Professur fuer Informatik
ETH-Hoenggerberg
CH-8093 Zuerich
September 1992

y
z
x



DEVICE LAYOUT FOR ÜBUNG 1



+-----+
| 'Initial' file |
+-----+

(* -----
"Initial" file with notes for Uebung1 of the SESES course given on
18 September 1992 at the ETH Zuerich. File created by J.G.Korvink.
Copyright by Professur fuer Informatik
ETH-Hoenggerberg
CH-8093 Zuerich
September 1992 ----- *)

- (* -----
0. This sentence is not read by the "Initial" file parser. It is a comment.
The parser ignores all text written between the (*) and *) brackets. Note
that comments can be nested, else the above line would not have been
possible!
1. The problem is given a title:
----- *)

Title Interconnect Capacitance Simulation

- (* -----
2. We define some useful constants which we will use in the
mesh specification later on:
----- *)

Define (* In the plane:

Length (mu)	No. of ME's	
		*)
lmet = 1.5	nmet = 2	
lbou = 6.0	nbou = 3	

(* For each layer:

Thickness (mu)	No. of ME's	Top ME no.	
			*)
lair = 10.00	nair = 3	tair = 0	
lpas = 1.00	npas = 2	tpas = tair + nair	
lalu = 1.10	nalu = 2	talu = tpas + npas	
lox1 = 0.35	nox1 = 2	tox1 = talu + nalu	
lpol = 0.30	npol = 2	tpol = tox1 + nox1	
lox2 = 0.50	nox2 = 1	tox2 = tpol + npol	
lsub = 10.00	nsub = 4	tsub = tox2 + nox2	

nx = nbou+nmet (* the no. of ME in the x direction *)
ny = nbou+nmet (* the no. of ME in the y direction *)

- (* -----
3. We define the material constants. The metals are not modelled...
----- *)

Material Si Coefficient Epsrel 11.7
Material Air Coefficient Epsrel 1.0
Material SiO2 Coefficient Epsrel 3.9

- (* -----
4. We specify the field to be solved for:
----- *)

Field Psi

(*) -----
5. Here the macro element geometry is defined. This entails giving the numbers and sizes for each coordinate direction:

```
XME nbou-1 ((lbou - (lmet/3))/(nbou-1)) 1 (lmet/3)
  nmet (lmet/nmet)
YME nbou-1 ((lbou - (lmet/3))/(nbou-1)) 1 (lmet/3)
  nm̄et (lmet/nmet)
ZME nair (lair/nair)
  npas (lpas/npas)
  nalu (lalu/nalu)
  nox1 (lox1/nox1)
  npol (lpol/npol)
  nox2 (lox2/nox2)
  nsub (lsub/nsub)
```

(*) -----
6. On the next line is the file termination symbol, followed by a return. The "Initial" file parser only reads the file up to and including the Finish statement. All other characters are ignored. The interactive FRONT END can display an incomplete "Initial" file that at least is complete up to the ME statements. The present "Initial" file should be viewed with the FRONT END, by typing:

```
... mg3d InitialUebung1 &
```

The '&' sign starts the FRONT END in background mode. It is a UNIX (tm) option. Before continuing with the instructions in this file, familiarise yourself with the FRONT END. Try out the 'FRAME' as well as the 'ME BOUND' options. Look at slices of the mesh with the 'VIEW' submenu. Do NOT quit the FRONT END. Simply move it out of the way and return to editing this file.

The Finish statement should be removed and the file saved again. A way of removing the Finish statement is by surrounding it by comment brackets. Save the file after the alteration.

Finish

(*) -----
7. The metallic deposits are 'removed' from the mesh. We do not simulate the interior of the metals (Further on we will 'metallise' the surfaces of the holes using a contact statement):

```
DeleteME 0 nx
  nbou (nbou+nmet)
  talu (talu + nalu)
DeleteME nbou (nbou+nmet)
  0 ny
  tpol (tpol + npol)
```

(*) -----
8. Go back to the FRONT END and press on the RESTART button. The mesh will now be displayed with certain macro elements removed. The following Finish statement should be removed once the deletions are observed to be correct. Save the altered file.

Finish

(* -----
9. We now assign materials to the different regions of our simulation domain. This is done macro element wise:
----- *)

```
Material Si Global  
Material Air 0 nx  
0 ny  
tair (tair + nair)  
Material SiO2 0 nx  
0 ny  
tpas (tpas + npas + nalu)  
Material SiO2 0 nx  
0 ny  
tox1 (tox1 + nox1 + npol + nox2)
```

(* -----
10. Go back to the FRONT END and press on the RESTART button. The mesh will now be displayed with the materials assigned. The following Finish statement should be removed once the materials are observed to be correctly assigned. Save the altered file.

Finish

(* -----
11. In step 7 we deleted macro elements to remove the metallic deposits. We now associate each of these exposed inner surfaces with Dirichlet contacts (equipotential surfaces):
----- *)

```
Contact plus IJType 0 nx nbou (nbou+nmet) talu  
IJType 0 nx nbou (nbou+nmet) (talu + nalu)  
IKType 0 nx talu (talu + nalu) nbou  
IKType 0 nx talu (talu + nalu) (nbou+nmet)  
Basic Psi Dirichlet 1.0 Volt  
Contact ground IJType nbou (nbou+nmet) 0 ny tpol  
IJType nbou (nbou+nmet) 0 ny (tpol + npol)  
JKType 0 ny tpol (tpol + npol) nbou  
JKType 0 ny tpol (tpol + npol) (nbou+nmet)  
Basic Psi Dirichlet 0.0 Volt
```

(* -----
12. Go back to the FRONT END and press on the RESTART button. The mesh will now be displayed with the contacts coloured. The following Finish statement is truly the end of the file and should NOT be removed. The "Initial" file is now complete.

If some of the dimensions did not appeal to you (i.e. your manufacturing process results in different layer thickness, or is a 6 mu process, or you would prefer different macro element subdivisions), these can now be adjusted simply and redisplayed with the FRONT END.
----- *)

Finish

+-----+
| 'Command' file |
+-----+

(* -----
"Command" file with notes for Uebung1 of the SESES course given on
18 September 1992 at the ETH Zuerich. File created by J.G.Korvink.
Copyright by Professur fuer Informatik
ETH-Hoenggerberg
CH-8093 Zuerich
September 1992 ----- *)

(* -----
0. On the first line, we specify to the kernel that a new problem is to be
calculated from our "Initial" file: ----- *)
New InitialFile InitialUebung1

(* -----
1. Here we specify how accurate the equation solver should solve the linear
system. This is necessary, because we use an iterative method. The
default value is 1.0E-2 ----- *)
Set LinearSolver Tolerance Absolute 1.0E-9

(* -----
2. These statements will be activated later on, see 8. below. ----- *)
(* Set Refine zonel AllME
Set Info ContactSummary *)

(* -----
3. The next statement specifies a linear solution step: ----- *)
Do Solve Stationary

(* -----
5. These statements will be activated later on, see 8. below. ----- *)
(* Do Refine zonel Adaptive 1 Statistical 0.0 IndDelta Do Solve Stationary
Do Refine zonel Adaptive 1 Statistical 0.0 IndDelta Do Solve Stationary
Do Refine zonel Adaptive 1 Statistical 0.0 IndDelta Do Solve Stationary *)

(* -----
6. This set of statements cause a series of "Bulk" files, necessary for
graphical output, to be created. The "Bulk" files are in Ascii format. ----- *)
Do Bulk Write Element Ascii
Do Bulk Write Incidence Ascii
Do Bulk Write Field Psi Ascii
Do Bulk Write Field IndDelta Ascii
Do Bulk Write Field E-Field Ascii

(* -----

7. The file terminator symbol, followed by a return, completes the "Command" file:

Finish

- (* -----*)
8. The next step is to adaptively form a refined mesh. This is done with statements specifying

- a) Refinement zones. (A Set statement)

A name is associated with a range of macro elements. This enables selected adaptation in user-defined regions. We select the whole mesh here.

- b) Contact summary table. (A Set statement)

This ensures that we are given a table of the contact charges, as they were computed after each solution step, right at the end of the simulation. We are able to judge whether the solution is converging as the mesh is refined.

- c) A number of adaptive steps. (Here we show three sets of Do statements)

We both adaptively refine as well as solve the system.

For refinement it is necessary to specify a zone name, a direction, how many times, and for adaptive methods a indicator as well as how the indicator is to be evaluated.

To activate this section, uncomment the two statements under point 2 above, as well as the three statements of point 5 above, then save this "Command" file.

(* -----*)

SESES Algorithmen.

Dr. H. Schwarzenbach

SESES Algorithmen

A. Methode von Newton Raphson

Residualgleichungen:

$$R_i := \int_{\Omega} H_i \left\{ \underbrace{\operatorname{div} F(f_a, t) - \rho(f_a, t)}_{=0} \right\} dv$$

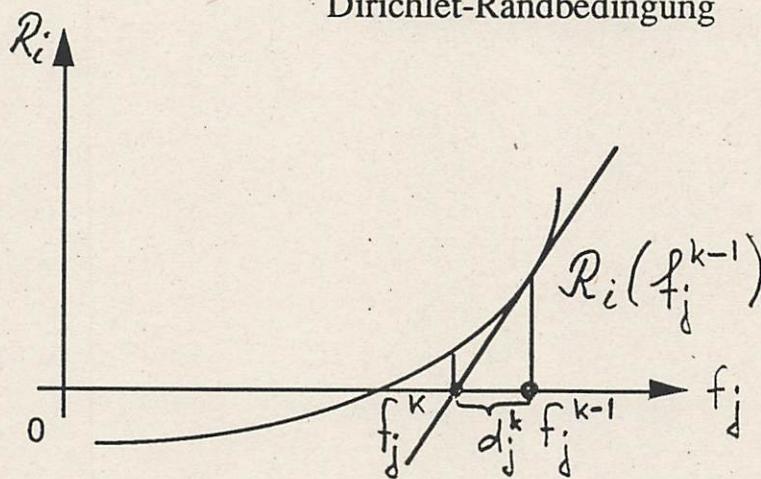
Differentialgleichung

$$- \int_{\partial\Omega} H_i \left\{ \underbrace{F(f_a, t) v - \alpha(t) f_a - \beta(t)}_{=0} \right\} d\sigma$$

"nicht"-Dirichlet-Randbedingung

z. B. $F(f_a, t) = \mu(f_a) \operatorname{grad} f_a$

Ansatz: $f_a(t) = \sum_{i=1}^n f_i(t) H_i(x, y, z) + \underbrace{\sum_{i=n+1}^N f_i(t) H_i(x, y, z)}_{\text{Dirichlet-Randbedingung}}$



Taylor: $R = R(f^{k-1}) + \underbrace{\delta R / \delta f(f^{k-1})}_{\text{Jacobimatrix}} (f^k - f^{k-1}) + \dots$
 $d^k \text{ (Inkrement)}$

Lineare Gleichungen: $0 = R(f^{k-1}) + \delta R / \delta f(f^{k-1}) d^k$
(k-te Iteration)

$$f^k = f^{k-1} + d^k$$

SESES - Commands:

Do Solve Stationary

Do Solve Stationary StepLength 0.1 NoSteps 3
Do Solve Stationary StepLength 0.1 ToTime 0.5

Do Solve Stationary StartTime 0.6 StepLength 0.1 NoSteps 3
Do Solve Stationary StartTime 0.6 StepLength 0.1 ToTime 0.8

Abbruchbedingungen:

Set Tolerance dof_name RelIncNorm
Set Tolerance dof_name RelResidNorm
Set Tolerance dof_name AbsIncNorm
Set Tolerance dof_name AbsResidNorm

B. Gauss-Seidel - voller Newton Raphson

Residualgleichungen:

$$\psi R = \psi R(\psi_a, n_a, p_a, t)$$

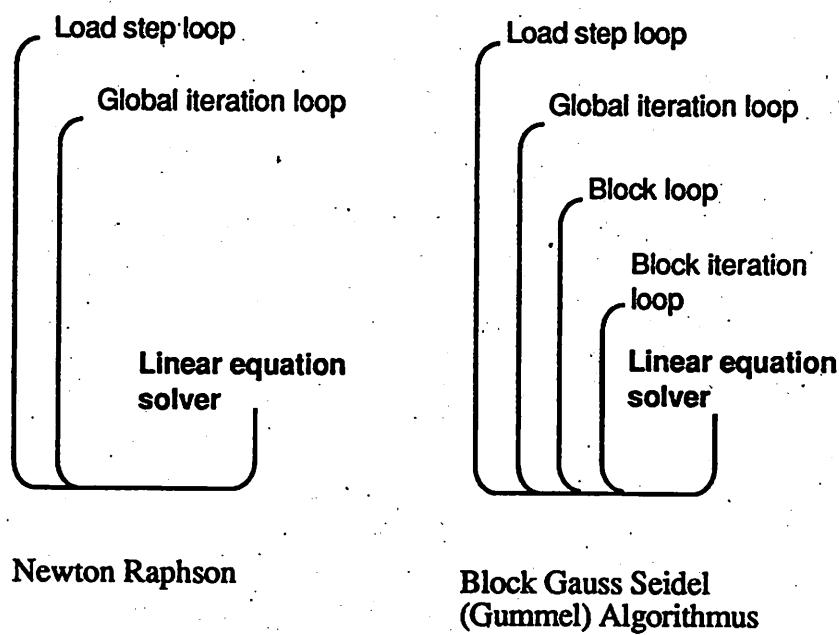
$$n R = n R(\psi_a, n_a, p_a, t)$$

$$p R = p R(\psi_a, n_a, p_a, t)$$

Ansatz: $\psi_a(t) = \sum_{i=1}^{n1} \psi_i(t) H_i(x,y,z) + \sum_{i=n1+1}^{N1} \psi_i(t) H_i(x,y,z)$

$n_a(t) = \sum_{i=1}^{n2} n_i(t) H_i(x,y,z) + \sum_{i=n2+1}^{N2} n_i(t) H_i(x,y,z)$

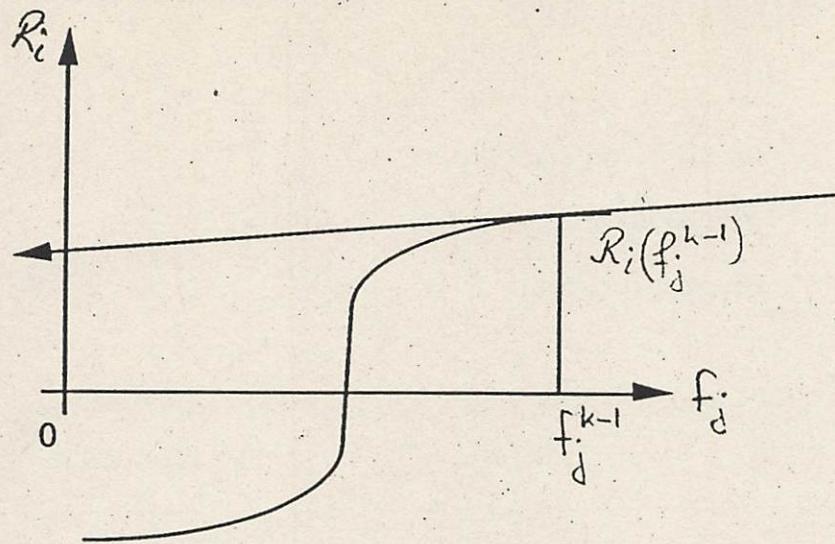
$p_a(t) = \sum_{i=1}^{n3} p_i(t) H_i(x,y,z) + \sum_{i=n3+1}^{N3} p_i(t) H_i(x,y,z)$



SESES - Commands:

Set BlockStruct Uncoupled
Set BlockStruct Block Psi Block U Block V
Set BlockStruct Coupled

C. Modifikation Inkrement



SESES - Commands:

Set Increment UpperLimit Field_name Enabled value
Set Increment UpperLimit Field_name Disabled

Set Increment LowerLimit Field_name Enabled value

Set Increment Scale Field_name Enabled value

Set Increment UpdateN Enabled
Set Increment UpdateP Enabled

Set Increment LowerLimitField Field_name Enabled value
Set Increment UpperLimitField Field_name Enabled value

D. Unabhängige Variable

- $\{ \psi, n, p \}$ Ladungsträgerdichten,
 $\{ \psi, u, v \}$ Slotboom Variable, $n = n_i \cdot u e^{\psi/U_T}$, $p = n_i \cdot v e^{-\psi/U_T}$;
 $\{ \psi, \phi_n, \phi_p \}$ Energie Variable $u = e^{-\phi_n/U_T}$, $v = e^{\phi_p/U_T}$.

Variable	Vorteile	Nachteile
$\{ \psi, n, p \}$	linear *)	unsymmetrische Matrizen
$\{ \psi, u, v \}$	linear *) symmetrische Matrizen	grosser Wertebereich
$\{ \psi, \phi_n, \phi_p \}$	kleiner Wertebereich	nichtlinear

*) : Falls die Mobilitäten unabhängig von den Ladungsträgerdichten sind und die Rekombination /Generation homogen linear von den Ladungsträgerdichten abhängen, dann sind die Kontinuitätsgleichungen in den entsprechenden Fällen linear.

SESES Initial File:

```

Field Psi U V
Field Psi N P
Field Psi PhiN PhiP
  
```

Problemlösung Beispiel Wärmeleitung

Übung II: Nichtlineare Probleme

Dr. M. Roos

UEBUNG 2

REFINEMENT UND PROBLEME MIT SCHWACHER NICHTLINEARITAET

In dieser Uebung soll der Umgang mit einem SESES-Kernel vertieft werden, d.h. die Beschreibung eines Loesungsalgorithmus fuer ein konkretes Problem mittels Command Files.

a) Problembeschreibung

Wir wollen uns mit einem Gassensor befassen, der zur Detektion eines bestimmten Gases eine erhieltte Oxidschicht verwendet. Bei unserer Simulation konzentrieren wir uns auf das Waermeleitungsproblem. Insbesondere wollen wir die folgenden Frage beantworten:

- i) Zur Erwaermung der aktiven Sensorschicht wird ein Heizwiderstand verwendet, der eine einstellbare Heizleistung abgeben kann. In erster Linie wollen wir die Temperaturverteilung und den Zusammenhang von abgegebener Leistung und erreichbarer Sensortemperatur modellieren.

In erster Ordnung ist der Waermeleitungskoeffizient kappa von der Temperatur unabhaengig, d.h. wir haben ein lineares Problem vor uns. Damit ist die Heizleistung und die Temperatur an einem festen Punkt durch folgendens Gesetz beschrieben:

$$(1) \quad T = T_0 + \alpha * P,$$

wo P die Heizleistung bedeutet.

- ii) Falls fuer die Sensoroperation die Temperatur genau geregelt werden muss, ist es wichtig zu wissen, ob der Zusammenhang (1) fuer ein temperaturabhaengiges κ stark modifiziert wird. In einer zweiten Phase der Uebung studieren wir daher mit einem nichtkonstanten Modell fuer $\kappa(T)$ den Einfluss der Nichtlinearitaet. Dabei untersuchen wir die Loesungsstrategie fuer ein schwach nichtlineares Probleme an einem einfachen Beispiel.

b) Devicestruktur

Der aktive Sensor teil besteht aus einer duennen Siliziumfaeche die teilweise mit einem Oxid beschichtet ist (am Interface von Oxid und Si befindet sich der Heizwiderstand). Diese Sandwichstruktur ist mit duennen Oxidbalken am

Substrat aufgehaengt, das aus Plausibilitaetsgrunden fuer unsere Fragestellung nur teilweise modelliert werden muss. Die gesamte Struktur besitzt die Symmetriegruppe eines Quadrates, weshalb nur ein Viertel der Struktur simuliert wird (im Prinzip wuerde sogar ein Achtel genuegen).

Device Layout:

Figur 2

Die Waerme fliesst ueber die Oxidbruecken zum Substrat, welches als Waermesenke wirkt. Wir nehmen eine konstante Temperatur an der Unterseite des Substrates an. Diese Layout wird gewaehlt, weil es hervorragende Waermeisolationseigenschaften hat.

Das Modell fuer den nichtkonstanten Waermeleitungskoeffizienten des Oxsids wird durch das folgende Gesetz beschrieben

$$(2) \quad \kappa(T) = \frac{\kappa_0 (1 + \kappa_A T + \kappa_B T^2)}{(1 + \kappa_C T + \kappa_D T^2)}$$

durch Wahl der vier Konstanten $\kappa_A \dots \kappa_D$ koennen sehr vielfaeltige Temperaturabhaengigkeiten von κ approximiert werden.

In Aufgabe 4 wollen wir den Einfluss von (2) auf die Linearitaet der Temperatur-Heizleistungskurve untersuchen. Als Temperatursignal verwenden wir die Temperaturdifferenz an zwei ausgewahlten Punkten auf dem Sensor ($\rightarrow T_1, T_2, dT := T_2 - T_1$). Die Heizleistung wird durch die time-Variable, die im Initial File benuetzt wurde, bestimmt und ermoeglicht so, dass in einer und derselben Rechnung verschiedene Punkte der Temperatur-Heizleistungskurve berechnet werden koennen.

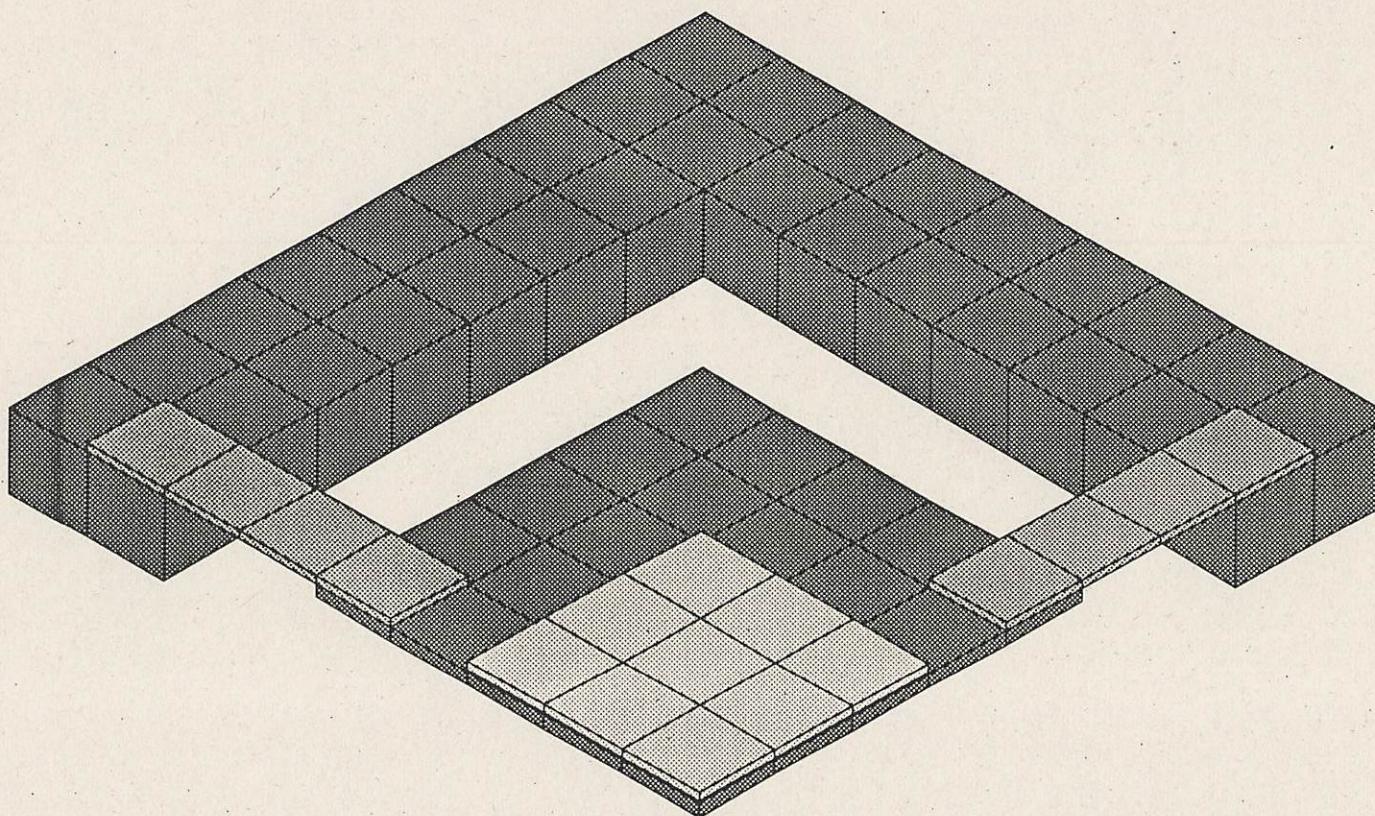
c) Vorgehen bei der Loesung der Aufgaben

Oeffnen Sie das Initial File "InitialUebung2" und die Command Files "RefinementCommand" sowie "NonLinearCommand". Die meisten Statements sind mit Kommentaren versehen, so dass Sie sich einfach zurechtfinden duerften.

c1) Refinement

In einem ersten Schritt wird eine optimale Masche fuer die endgueltige Loesung erzeugt. Verwenden Sie dazu das Command File "RefinementCommand" und loesen Sie die Aufgaben 1-3, welche am Ende dieses Files beschrieben sind.

y
x
z



Silicon SiO₂ TiN

Source BulkTemp

Figur 2

c2) Nichtlineare Loesungstrategie

Die oben erhaltene Masche wird im zweiten Schritt fuer die Loesung der eigentlichen Aufgabe verwendet. Der Command "Do Bulk Read Element" liest die zuvor erzeugte Masche ein, so dass Sie dafuer sorgen muessen, dass das File "Element" vor dem Starten des Kernels t3d wirklich die gewuenschte Masche enthaelt.

Um Ihnen die Analyse der Resultate zu erleichtern, habe ich ein Filterprogramm bereitgestellt, das es erlaubt die Resultate der Temperatur-Heizleistungskurve graphisch darzustellen. Rufen Sie dazu das Kernel-Programm folgendermassen auf:

```
t3d NonLinearCommand | tee Logfile
```

Der UNIX Command "| tee <file>" bedeutet, dass der Output von t3d in das File <file> geschrieben wird, zusaetlich zur Anzeige auf dem Bildschirm. Das ermoeglicht es Ihnen, dieses File weiterzubearbeiten. Tippen Sie dazu nach erfolgter Loesung den Befehl

```
graph Logfile
```

auf die Commandzeile. Es erscheint eine graphische Darstellung. Insbesondere werden die von den Textoutputstatements (-> Do Write MiddlePoint Temp) erzeugten Werte herausgefiltiert und zusammen mit einer linearen Extrapolation der ersten beiden TimeSteps aufgezeichnet.

Weitere Instruktionen werden im Uebungsraum gegeben.

Copyright by Professur fuer Informatik
ETH-Hoenggerberg
CH-8093 Zuerich
September 1992

Problemlösung Beispiel Halbleiterproblem

Übung III: Nichtlineare Probleme

Dipl. phys. G. Sartoris

Computing Semiconductor devices with SESES

The f2d f3d SESES kernel programs solve the stationary drift-diffusion (Van Roosbroeck) equations for semiconductors including magnetic field effects as given by:

$$\begin{aligned}\vec{\nabla}_\varepsilon \vec{\nabla} \psi &= q(n - p - C) \\ \vec{\nabla} \vec{J}_n &= qR; \quad \vec{J}_{0n} = qD_n e^{(\psi/V_T)} \vec{\nabla}(e^{(-\psi/V_T)} n) \\ \vec{\nabla} \vec{J}_p &= qR; \quad \vec{J}_{0p} = qD_p e^{(-\psi/V_T)} \vec{\nabla}(e^{(\psi/V_T)} p) \\ \vec{J}_a &= \frac{1}{1 + \mu_{Ha}} \left[\vec{J}_{0a} + \mu_{Ha} \vec{J}_{0a} \times \vec{B} + \mu_{Ha}^2 (\vec{J}_{0a} \vec{B}) \vec{B} \right]; \quad a = n, p\end{aligned}$$

The above equations are transformed by applying a one-to-one mapping to the set the carrier variables (ψ, n, p). Well-known are the Slotboom variables (ψ, u, v):

$$n = n_i e^{(\psi/V_T)} u \quad p = n_i e^{(-\psi/V_T)} v$$

and the quasi-Fermi variables (ψ, ϕ_n, ϕ_p):

$$n = n_i e^{(\psi - \phi_n)/V_T} \quad n = n_i e^{(\phi_p - \psi)/V_T}$$

All three sets of equation are implemented in SESES, whereas their use and choice is bound to the specific problem to be solved for. When computing solutions near thermodynamic equilibrium the Slotboom variables are to be preferred, else the carrier variables are pertinent. The quasi-Fermi variables are useful when solving the three set of equations in a coupled manner.

The discretized form of the above equations with an hybrid finite element method are given for the Slotboom variables by:

$$\begin{aligned}\vec{R}_I &= G^T F^{-1} (1/\varepsilon) G \vec{\psi} - q \int_{\Omega} (N_i e^{(\psi_h/V_T)} u_h - N_i e^{(-\psi_h/V_T)} v_h - C) \vec{H} d\Omega \\ \vec{R}_{II} &= q G^T F^{-1} \left(M_n^{-1} \frac{e^{-\psi_h/V_T}}{D_n} \right) G \vec{u} + q \int_{\Omega} R \vec{H} d\Omega \\ \vec{R}_{III} &= q G^T F^{-1} \left(M_p^{-1} \frac{e^{\psi_h/V_T}}{D_p} \right) G \vec{v} + q \int_{\Omega} R \vec{H} d\Omega\end{aligned}$$

with $\vec{H} := (H_0, H_1, \dots)$, $\psi_h := \vec{H} \vec{\psi}$, $u_h := \vec{H} \vec{u}$, $v_h := \vec{H} \vec{v}$ and

$$\vec{H} := (H_0, H_1, \dots) \quad F(w) := \int_{\Omega} (w \vec{C}_i) \vec{C}_j d\Omega \quad G := \int_{\Omega} \vec{C}_i \vec{n} H_j df; \quad M_a^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \mu_{Ha} \begin{pmatrix} 0 & -B_z & B_y \\ B_z & 0 & -B_x \\ -B_y & B_x & 0 \end{pmatrix}$$

The vectors: $\vec{\psi}, \vec{u}, \vec{v}$ are the unknowns of the problem and represent the values of the fields: ψ, u, v

at each mesh joint. The set of scalar $\{H_0, H_1, \dots\}$ and vector functions $\{\vec{C}_0, \vec{C}_1, \dots\}$ are approximations of the spaces where the function ψ, u, v and $\epsilon \nabla \psi, \vec{J}_n, \vec{J}_p$ are sought. Most of the problems arising when using the above discretization scheme are of a numerical nature when numerical integration is performed. All three equations contain integrals of exponential functions that, when computed on each element, reduce to the general form:

$$\int_{[-\frac{1}{2}, \frac{1}{2}]} P_n[x, y, z] \exp^{(a + bx + cy + dz + exy + fzx + gyz + hxyz)} dx dy dz$$

where $P_n[x, y, z]$ is a multivariate polynomial of degree n. Such integrals are numerically very hard to compute due to the large values of the constant a, b, c, \dots . SESES computes these integrals in one of two ways: other Gauss-Legendre quadrature rule or by discarding the terms: $exy + fzx + gyz + hxyz$ and exact integration of the remaining ones. By the second algorithm it is easy to see that if we divide the brick integration domain in subbrick domains we can perform the same integration with new values of the constants a, b, c, d . In the limit with infinite subdivision the integral will be exactly computed.

Problems associated with numerical integration and choice of an appropriate mesh can be easily investigated by solving the Poisson equation alone as is the case when computing the solution for thermodynamic equilibrium. For this task a typical SESES Command file will be:

```
(* Command file to compute the thermodynamic equilibrium solution *)
Set Increment      Psi           LowerLimit -0.5   Enabled
Set Increment      Psi           UpperLimit  0.5   Enabled
Set Model          IntExpPoisson Enabled
Set Parameter      ElemSubdivPoisson 3
Set BlockStruct    Block Psi
Set Tolerance      Psi           AbsIncrNorm 1.E-8
Set LinearSolver   Direct (* Only for small problems !? *)
Do   Solve         Initialize
Do   Solve         BCLocked
```

The upper and lower limitation of the increments are in general unnecessary because it is rare to see divergence when solving only the Poisson equation. The next two set commands determine the choice of the integration algorithm for the Poisson equation and their influence on the convergence rate for a given mesh will be investigated in the first part of the exercise session. A block containing the Poisson equation only is defined and the convergence tolerance for the Newton-Rapson method is set. If the thermodynamic equilibrium is only used as start solution for other bias points a less stringent convergence criteria can of course be used to save computing time. The Do Solve Initialize command, initializes the solution to the free space charge condition defined by: $\nabla \epsilon \nabla \psi = q(n - p - C) = 0$ and the Do Solve BCLocked solves the defined block structure by locking the boundary condition i.e. the values of the fields along any contact are left unchanged.

As second exercise we will investigate solutions using the Slotboom variables which are appropriate for an applied voltage in the range $-8 \dots +8$ Volt. If we further seek a solution which is near to thermodynamic equilibrium the uncoupled algorithm of Gummel can be used and recombination can be switched off. After having computed a thermodynamic equilibrium solution for the given device the next command reads:

```

(* Command file for solutions near thermodynamic equilibrium *)
Set Model           ZeroRecomb      Enabled
Set Parameter       ElemSubdivPoisson 3
Set Parameter       ElemSubdivCont   3
Set Model           IntExpPoisson   Enabled
Set Model           IntExpCont     Enabled
Set BlockStruct    Block U  Block V  Block Psi
Set Increment      U LowerLimitField Enabled 1.E-100
Set Increment      V LowerLimitField Enabled 1.E-100
Set Tolerance      U
Set Tolerance      V

Set Parameter       BlockIteration Psi 1
Set Parameter       GlobalIteration 15

Set Extrapolation  Psi Linear

Set Tolerance      Psi AbsIncrNorm 1.E-4
Do Solve           Stationary 0.0001
Set Tolerance      Psi AbsIncrNorm 1.E-2
Do Solve           Stationary StartTime 0.1
                           StepLength 0.2 ToTime 1
Set Tolerance      Psi AbsIncrNorm 1.E-10
Do Solve           Stationary StartTime 1.0

```

The first command defines as before the integration algorithm for the Poisson and current continuity equations, then the Gummel algorithm is defined by declaring three blocks each one containing one equation. The Slotboom variables U and V have strictly positive functions, however during the solution process it can happen that negative values appear. This is avoided by the command Set Increment U LowerLimitField Enabled 1.E-100 which modifies the increment such that the field will not have values less than the given limit. If recombination is not considered both continuity equations are linear, and in this case we switch off all convergence criteria for the fields U and V such that convergence is governed only by the Poisson equation. If a good integration algorithm has been chosen the Newton algorithm for the Poisson equation will show nearly quadratic convergence, however the Gummel algorithm does not. In most case only one block iteration for the Poisson equation is thus needed to make the Gummel loop convergent and also to execute it with minimum time. This is selected by the command Set Parameter BlockIteration Psi 1, the next statement limiting the number of global iterations is simply used to trap non convergence. Experience shows that when computing a solution with Slotboom variables this has to be done in steps by incrementing the voltage at the contacts of at maximum one Volt until the chosen applied voltage is reached. The convergence tolerance for all intermediary solutions has to be selected so that the next iterations will also converge. The best time performances are also achieved by selecting a linear extrapolation function for the Psi fields, U and V do not need it because the equations are linear. After having computed the solution at thermodynamic equilibrium we compute another one extremely near to this one such that few iterations are needed. Now with two solutions we can extrapolate an initial guess for the third one, and for the latter solution the convergence criteria is set to normal values.

As the last exercise we will investigate speed up which can be achieved when running SESES concurrently on a number of machines connected by the same network. For this task use:

```

Set Parameter NetworkHost 2 <hostname1> <hostname2>
Set Info          TimeSummary

```

The second command prints at the end of the computation, a summary of the time spent to assemble the system, to solve it, and other tasks.