Head of the Institute: Prof. Dr.-Ing. H.-J. Christ

KinCorr
Modeling of Corrosion Phenomena

Ravisankar Naraparaju M.Sc

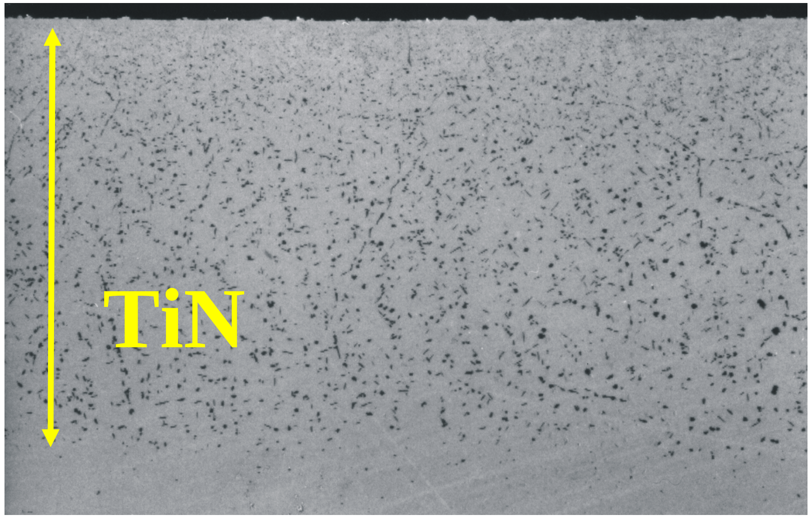
What is KinCorr?

- KinCorr - a computer- based program for simulation of high- temperature corrosion phenomena
- Use of numerical diffusion calculation in combination with thermodynamic equilibrium concepts
- The application KinCorr is parallelized along its functions (function-master, function- slave and function- matlab)

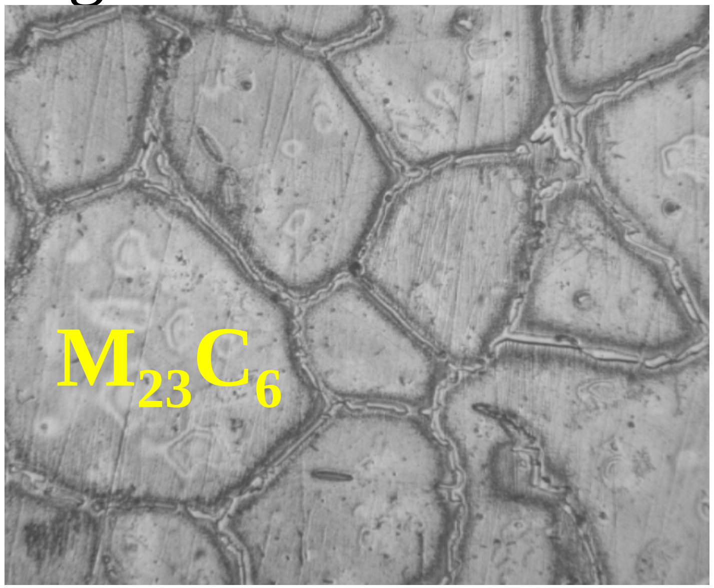
Physical Modeling
and
Computer-Based Simulation

Physical Modeling

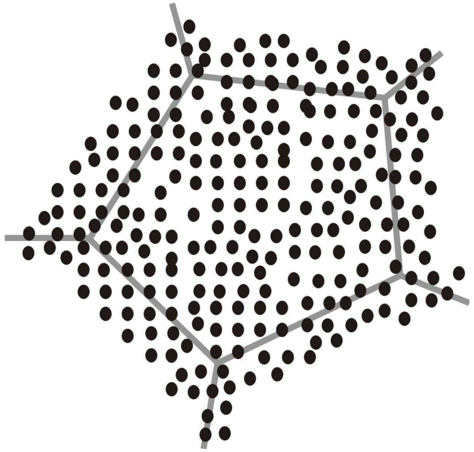
homogeneous internal attack | intergranular corrosion attack

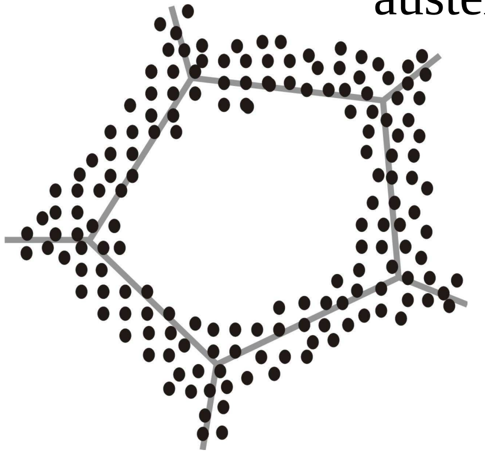


Ni-20Cr-2Ti



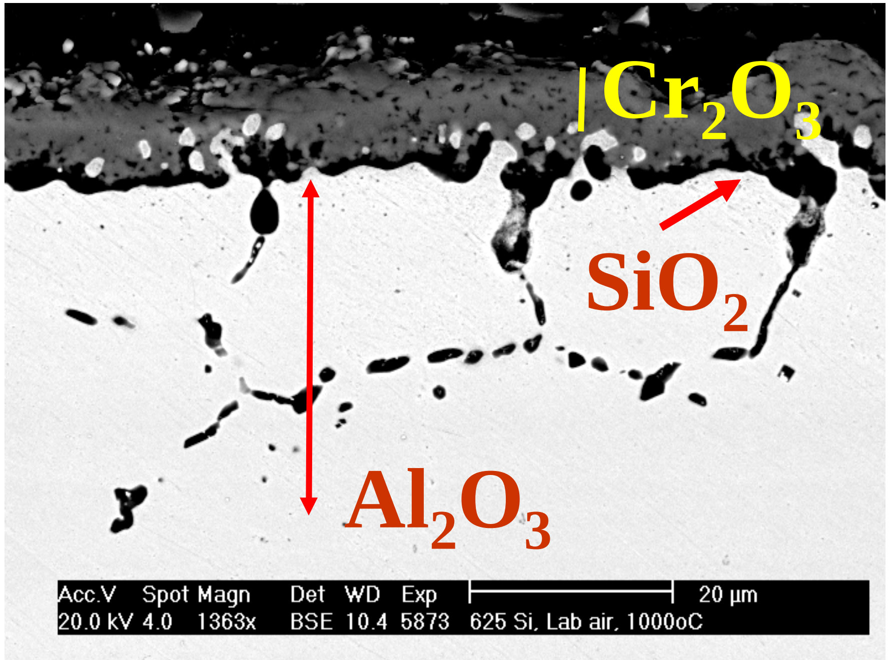
austenitic steel





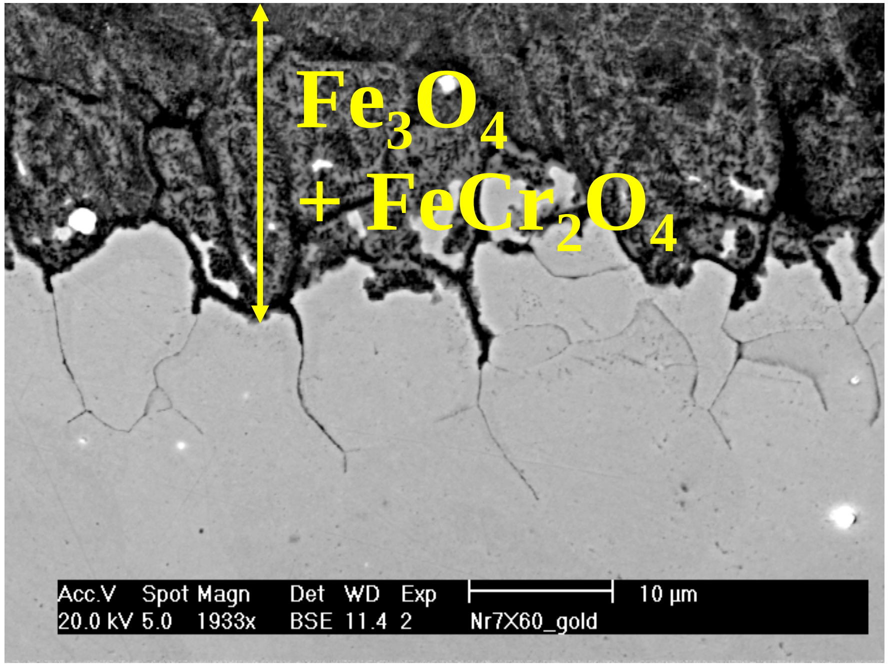
Physical Modeling

outer scale and internal attack



Inconel 625 Si

inward-growing scale



low-Cr steel

More realistic systems

ChemApp and data-bank

Diffusion in different phases (corrosion product and substrate) differentiation of diffusion along alloy grain boundary and volume

Thermodynamics

Diffusion

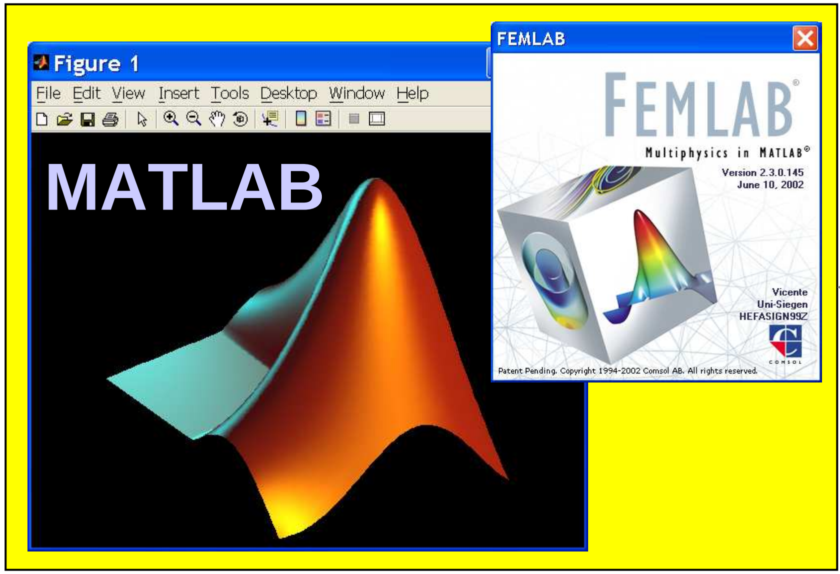
Computer

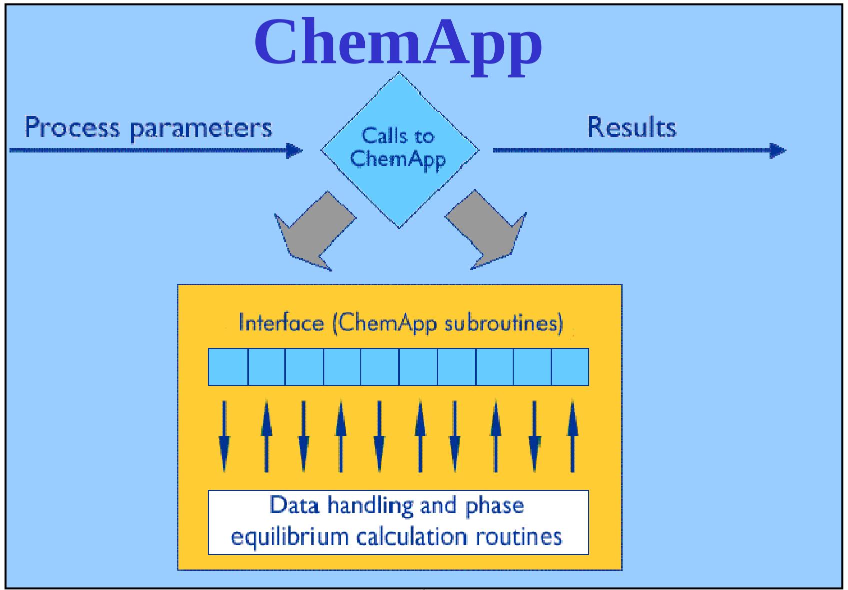
Description of corrosion kinetics

moving boundary conditions

Life-cycle of power plant materials

Modeling





solid state diffusion

representation

high-temperature corrosion

InCorr

local thermodynamic equilibrium

G = Minimum

Mathematical Modeling

One-Dimensional

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial c}{\partial x}\right) \xrightarrow{D = f(T)} \frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}$$

Two-Dimentional

$$\frac{\partial c}{\partial t} = D_x\frac{\partial^2 c}{\partial x^2} + D_y\frac{\partial^2 c}{\partial y^2}$$

One-Dimensional Problem – explicit finite-difference method

$$\frac{\partial c}{\partial t} \approx \frac{c_{i,j+1} - c_{i,j}}{\Delta t} \qquad \frac{\partial^2 c}{\partial x^2} \approx \frac{c_{i+1,j} - 2c_{i,j} + c_{i-1,j}}{(\Delta x)^2}$$

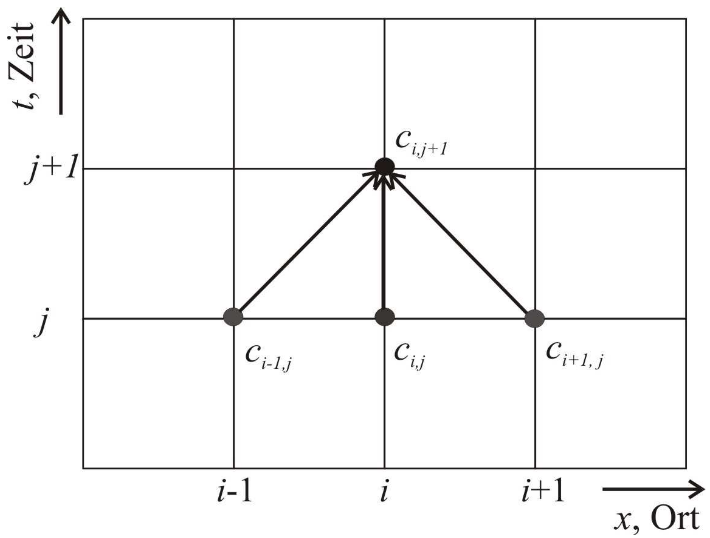
$$r = \frac{D\Delta t}{(\Delta x)^2}$$

$$r \le \frac{1}{2}$$

$$c_{i,j+1} = rc_{i-1,j} + (1-2r)c_{i,j} + rc_{i+1,j}$$

$$\boldsymbol{c}_{j+1} = \boldsymbol{A}\boldsymbol{c}_j + \boldsymbol{b}_j$$

$$\boldsymbol{A} = \begin{pmatrix} 1-2r & r & 0 & 0 & 0 & \dots & 0 \\ r & 1-2r & r & 0 & 0 & \dots & 0 \\ 0 & r & 1-2r & r & 0 & \dots & 0 \\ 0 & 0 & r & 1-2r & r & \dots & 0 \\ 0 & 0 & 0 & r & . & \dots & 0 \\ . & . & . & . & . & . & r \\ 0 & 0 & 0 & 0 & 0 & r & 1-2r \end{pmatrix} \qquad \boldsymbol{b}_j = \begin{pmatrix} rc^s \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$$



One-Dimensional problem – implicit finite-difference method

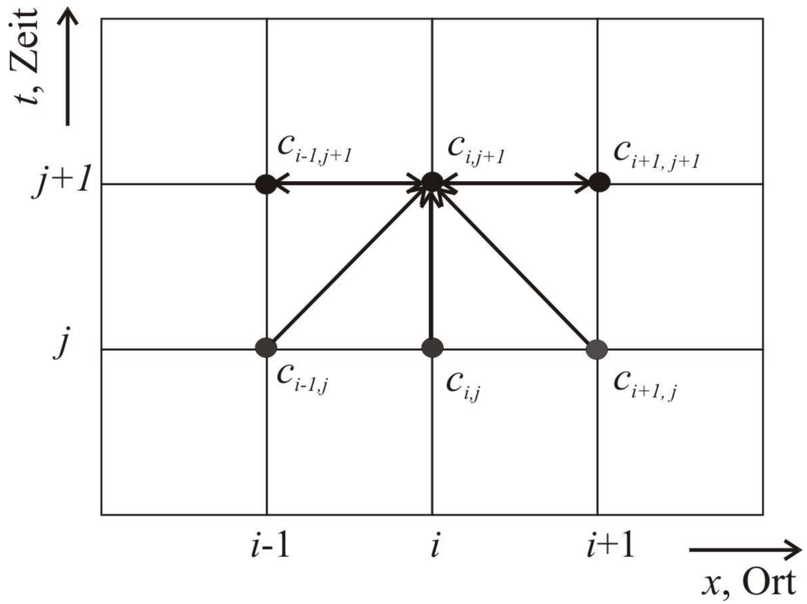
$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}$$

$$\begin{cases} \dfrac{\partial c}{\partial t} \approx \dfrac{c_{i,j+1} - c_{i,j}}{\Delta t} \\ \dfrac{\partial^2 c}{\partial x^2} \approx \dfrac{1}{2(\Delta x)^2}(c_{i+1,j} - 2c_{i,j} + c_{i-1,j} + c_{i+1,j+1} - 2c_{i,j+1} + c_{i+1,j+1}) \end{cases}$$

$$\boldsymbol{c}_{j+1} = (2\boldsymbol{I} + \boldsymbol{N})^{-1}(2\boldsymbol{I} - \boldsymbol{N})\boldsymbol{c}_j + (2\boldsymbol{I} + \boldsymbol{N})^{-1}\boldsymbol{b}_j$$

$$\boldsymbol{N} = r\boldsymbol{M}_x$$

$$\boldsymbol{M}_x = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & 0 & -1 & 2 & -1 & \dots & 0 \\ 0 & 0 & 0 & -1 & . & \dots & 0 \\ . & . & . & . & . & . & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$



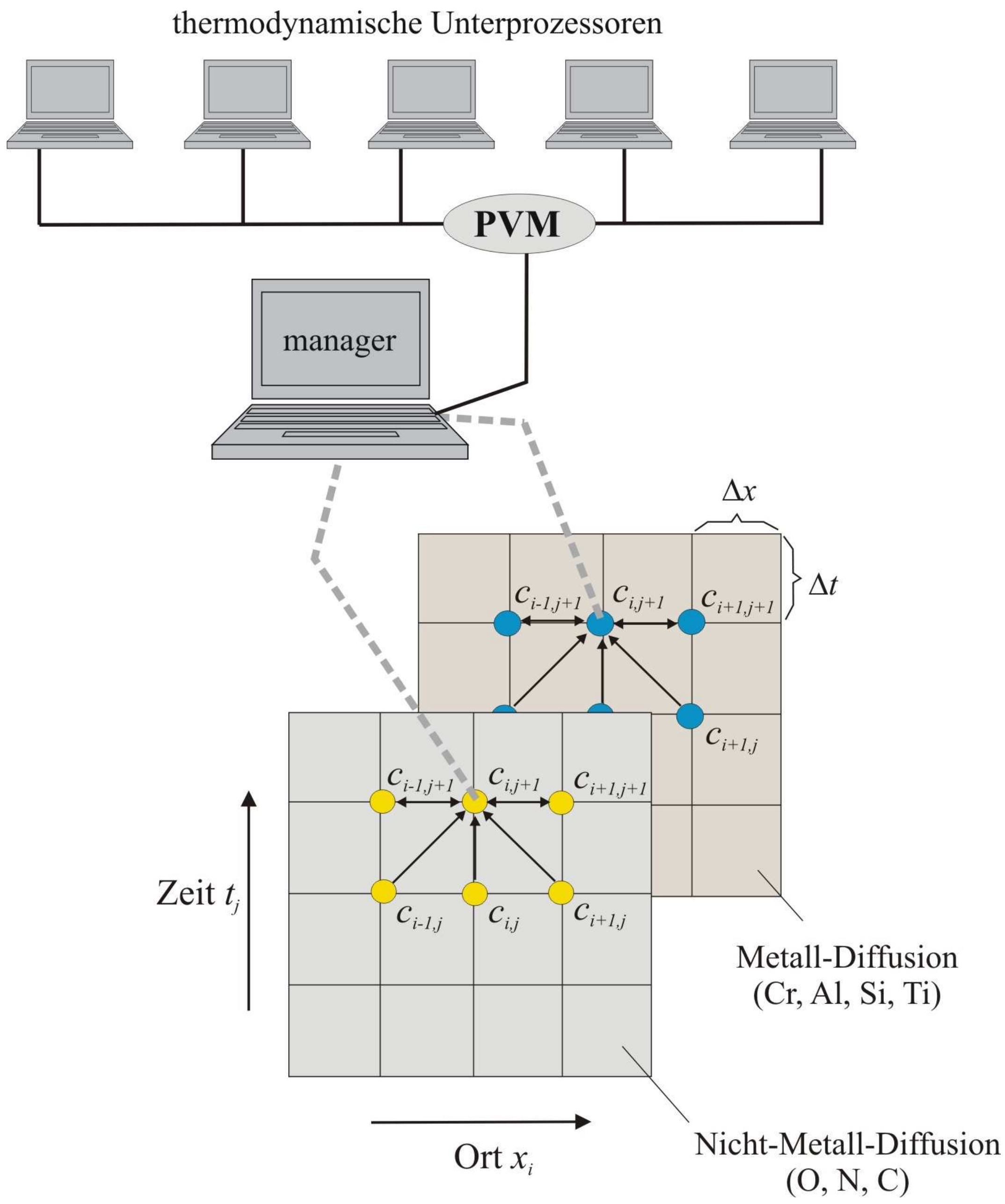
Two-Dimensional Problem – implicit finite-difference method

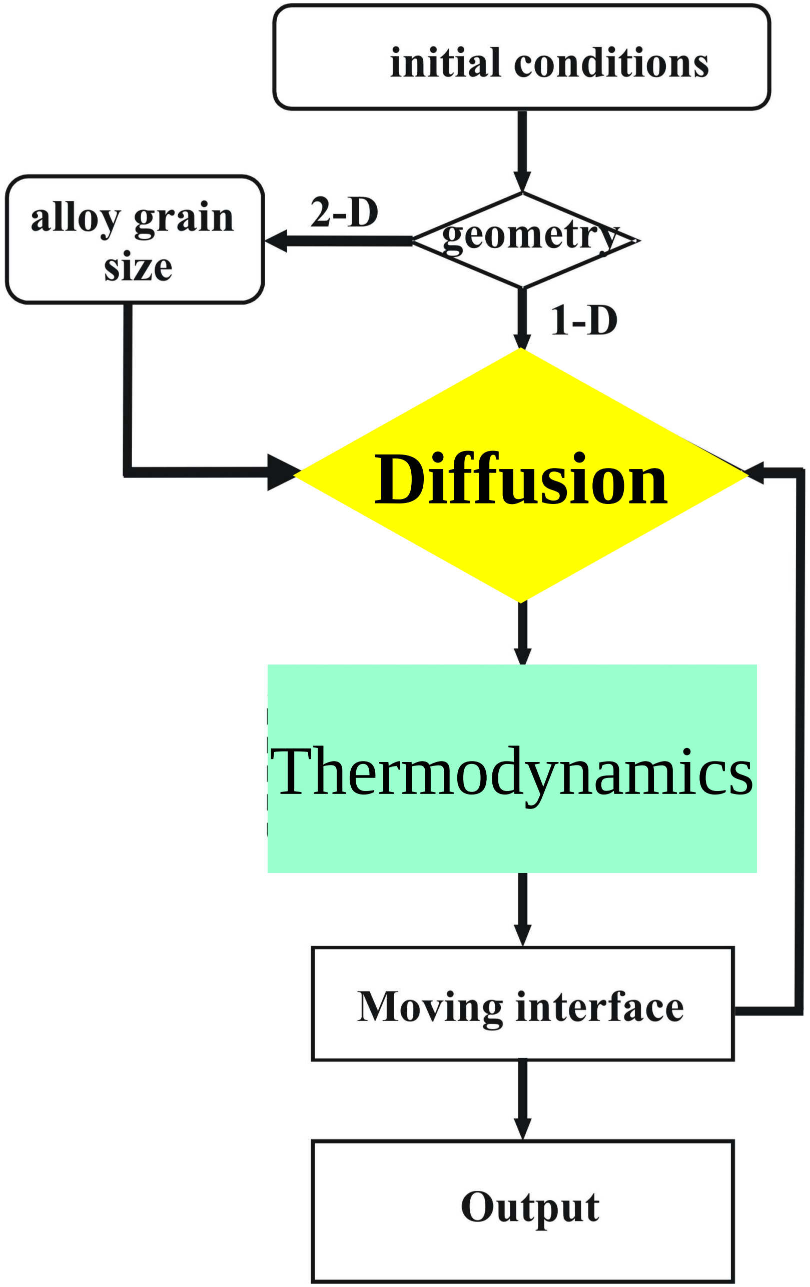
$$\frac{\partial c}{\partial t} = D_x \frac{\partial^2 c}{\partial x^2} + D_y \frac{\partial^2 c}{\partial y^2}$$

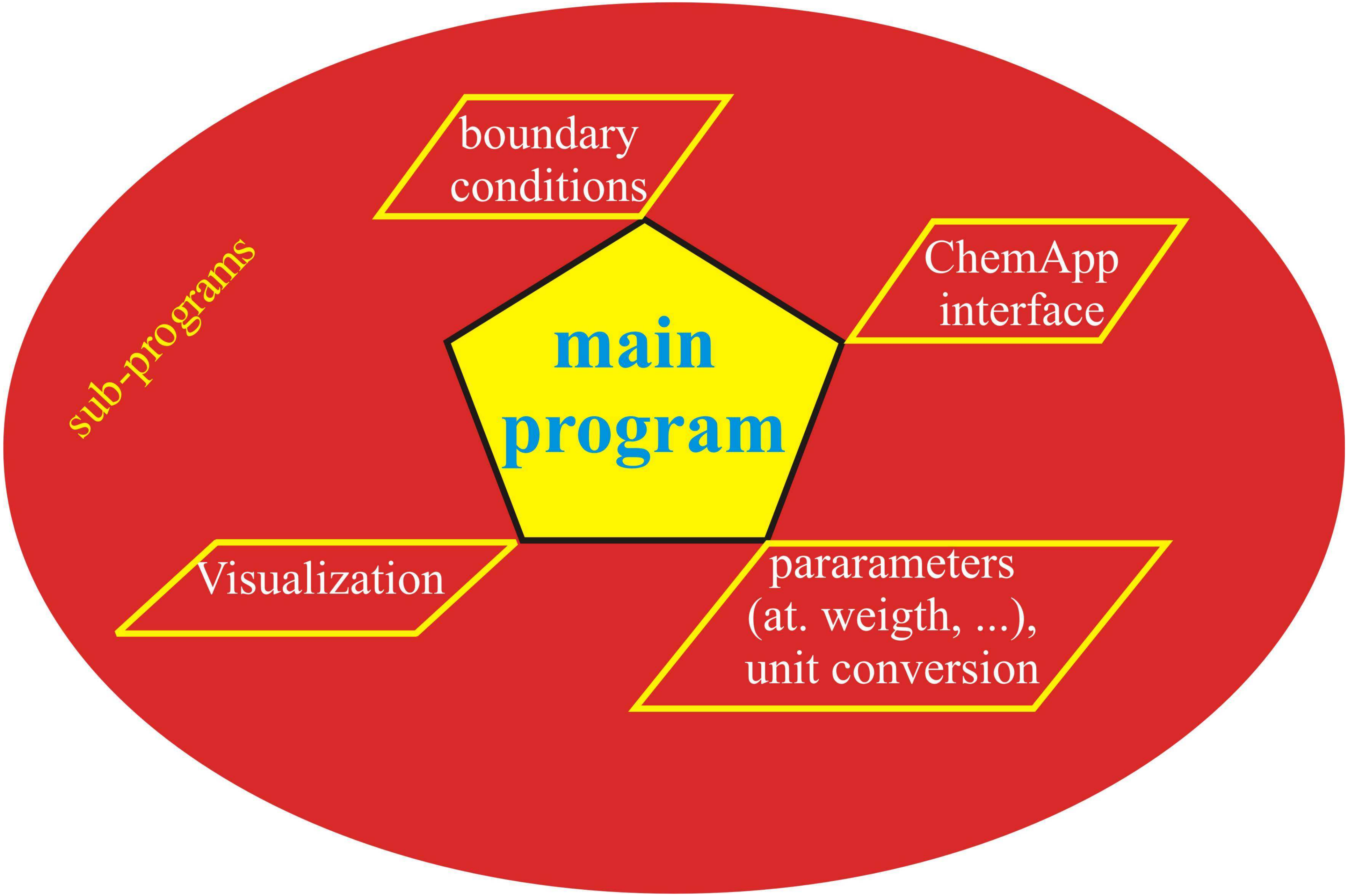
$$r_x = \frac{D_x \Delta t}{(\Delta x)^2} \qquad r_y = \frac{D_y \Delta t}{(\Delta y)^2}$$

$$r_x \cdot \left(c_{ix+1,iy}^{j+1} + c_{ix,iy-1}^{j+1}\right) - 2 \cdot \left(1 + r_x + r_y\right) \cdot c_{ix,iy}^{j+1} + r_y \cdot \left(c_{ix,iy+1}^{j+1} + c_{ix,iy-1}^{j+1}\right) =$$
$$- r_y \cdot \left(c_{ix,iy-1}^{j} + c_{ix,iy+1}^{j}\right) - 2 \cdot \left(1 - r_x - r_y\right) \cdot c_{ix,iy}^{j} - r_x \cdot \left(c_{ix-1,iy}^{j} + c_{ix+1,iy}^{j}\right)$$

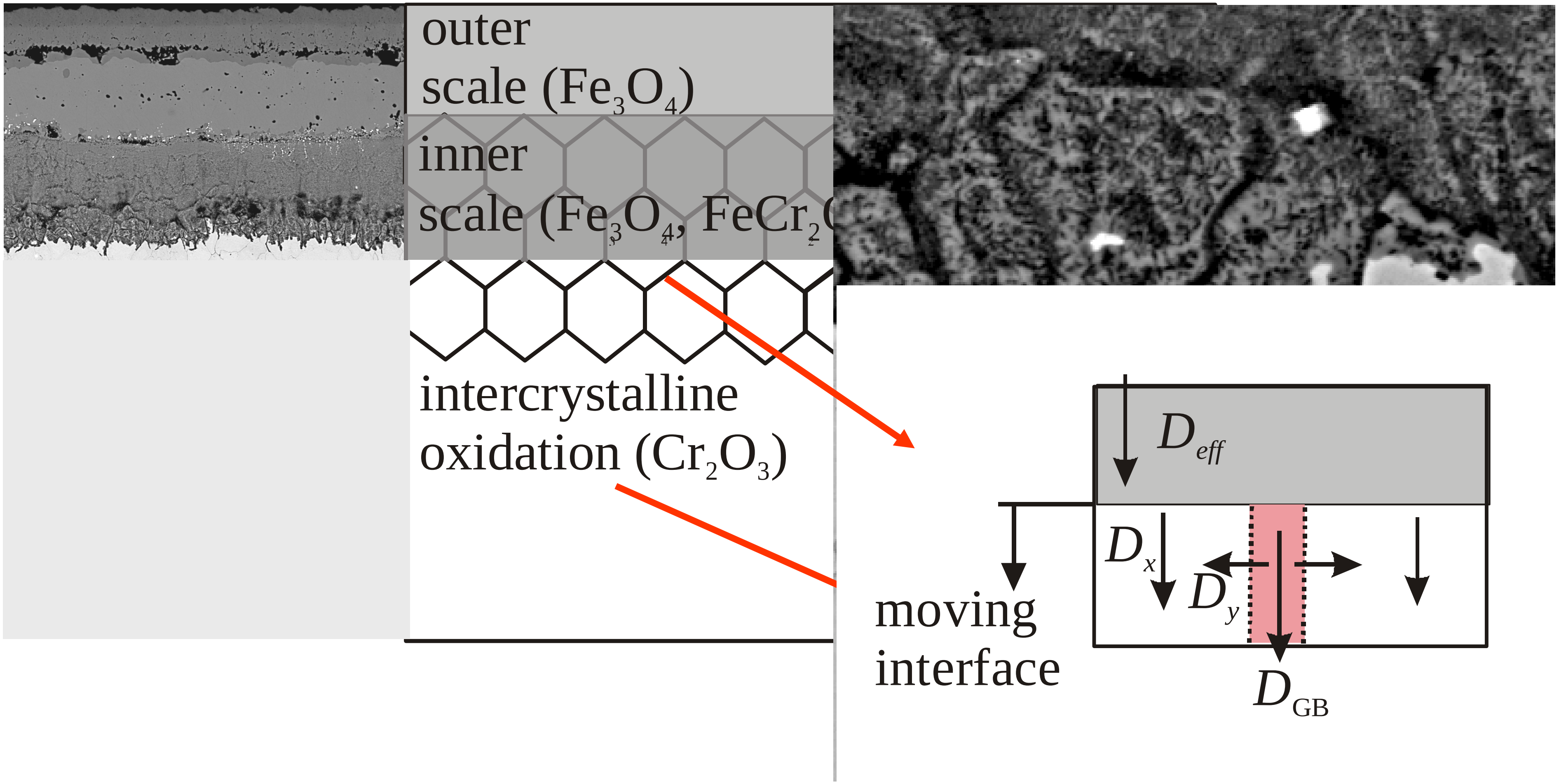
$$\frac{c(x,y,t+\Delta t) - c(x,y,t)}{\Delta t} = \frac{D_x(x,y)}{2(\Delta x(x,y))^2}[c(x-\Delta x,y,t) - 2c(x,y,t) + c(x+\Delta x,y,t)]$$
$$+ \frac{D_x(x,y)}{2(\Delta x(x,y))^2}[c(x-\Delta x,y,t+\Delta t) - 2c(x,y,t+\Delta t) + c(x+\Delta x,y,t+\Delta t)]$$
$$+ \frac{D_y(x,y)}{2(\Delta y(x,y))^2}[c(x,y-\Delta y,t) - 2c(x,y,t) + c(x,y+\Delta y,t)]$$
$$+ \frac{D_y(x,y)}{2(\Delta y(x,y))^2}[c(x,y-\Delta y,t+\Delta t) - 2c(x,y,t+\Delta t) + c(x,y+\Delta y,t+\Delta t)]$$

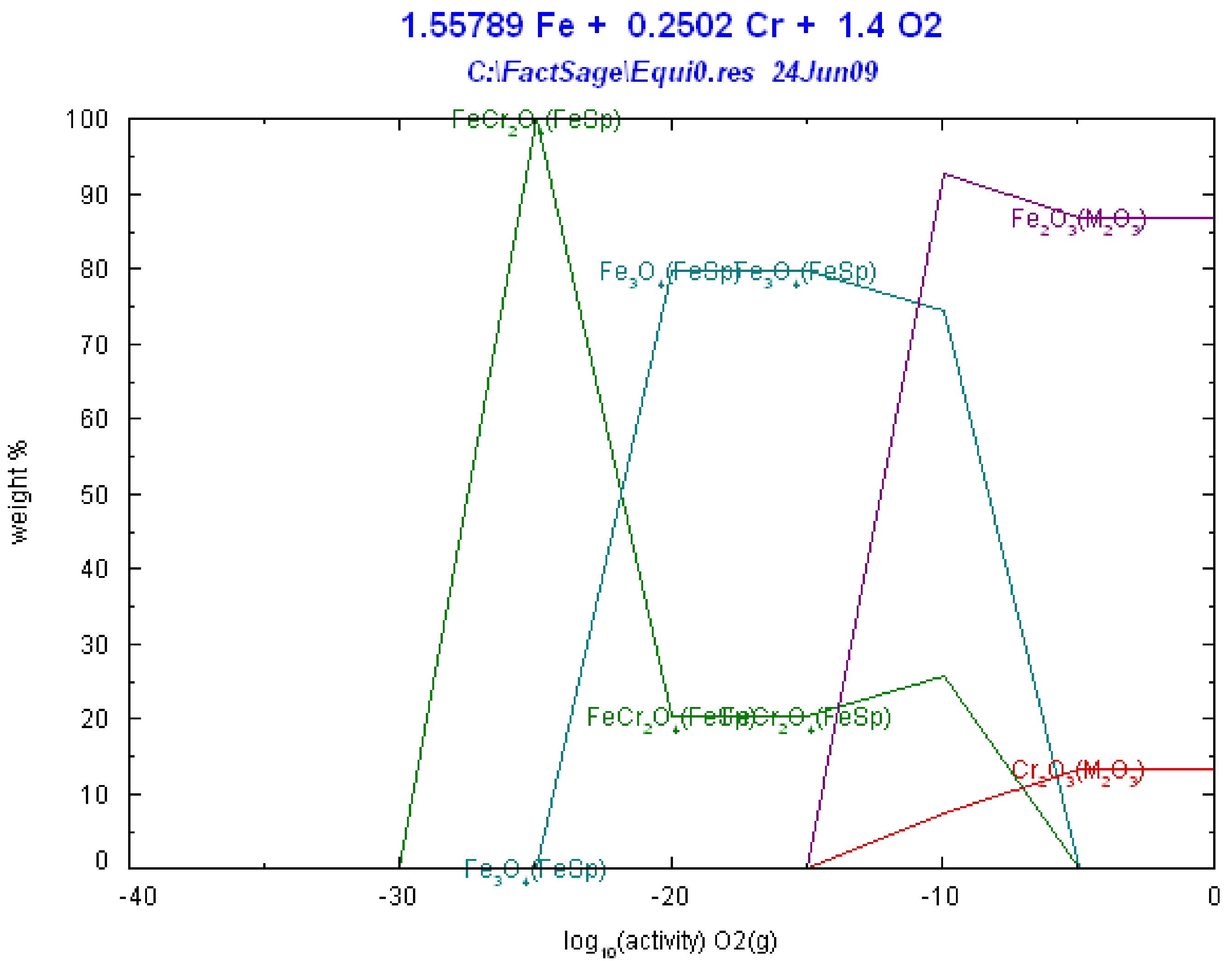

thermodynamische Unterprozessoren
PVM
manager
Δx
Δt
$c_{i-1,j+1}$
$c_{i,j+1}$
$c_{i+1,j+1}$
$c_{i+1,j}$
$c_{i-1,j}$
$c_{i,j}$
Zeit t_j
Ort x_i
Metall-Diffusion
(Cr, Al, Si, Ti)
Nicht-Metall-Diffusion
(O, N, C)


initial conditions
geometry
2-D
alloy grain size
1-D
Diffusion
Thermodynamics
Moving interface
Output

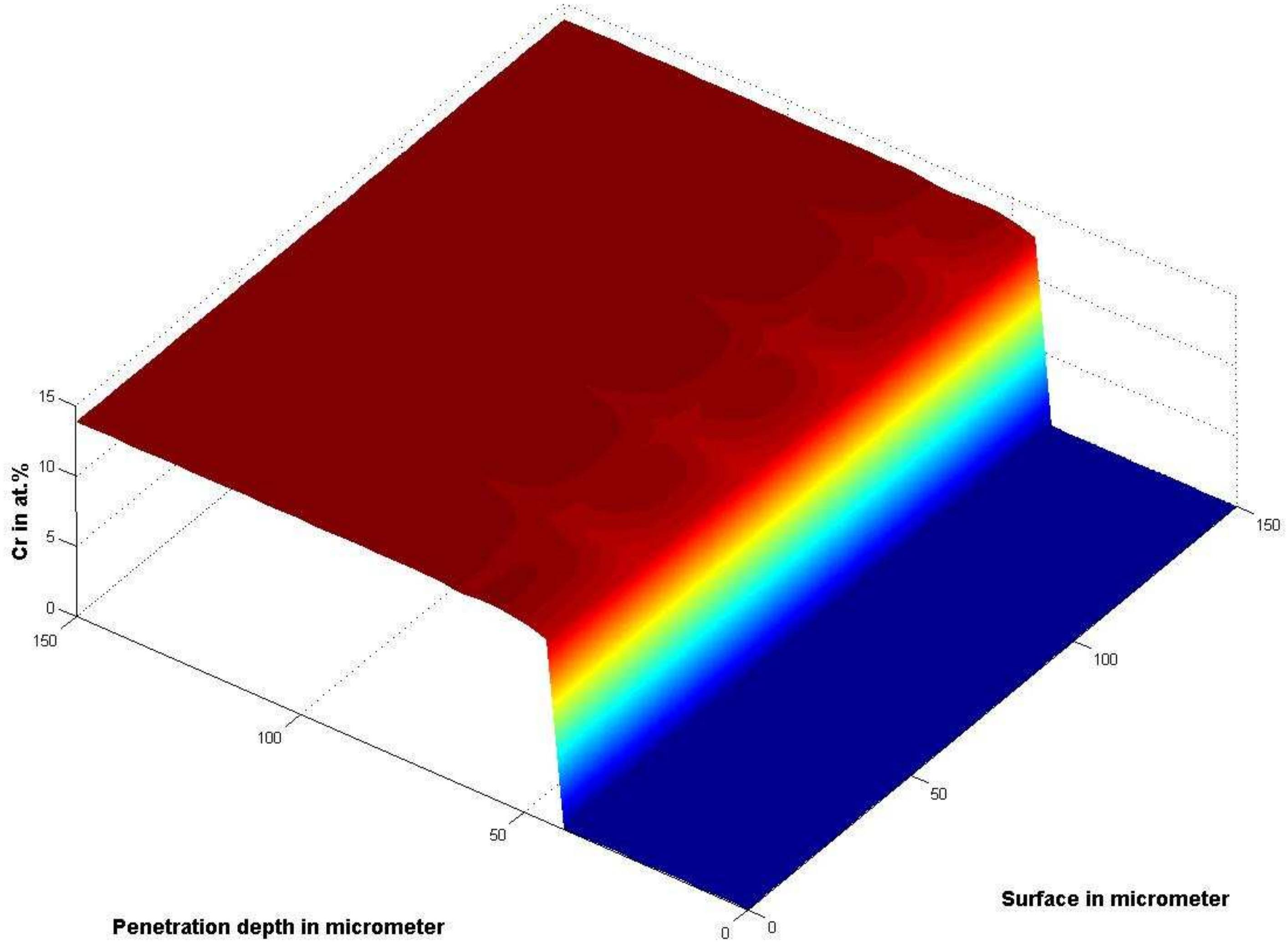

sub-programs
boundary conditions
ChemApp interface
main program
Visualization
pararameters (at. weigth, ...), unit conversion

Computer Simulation of Oxidation Processes

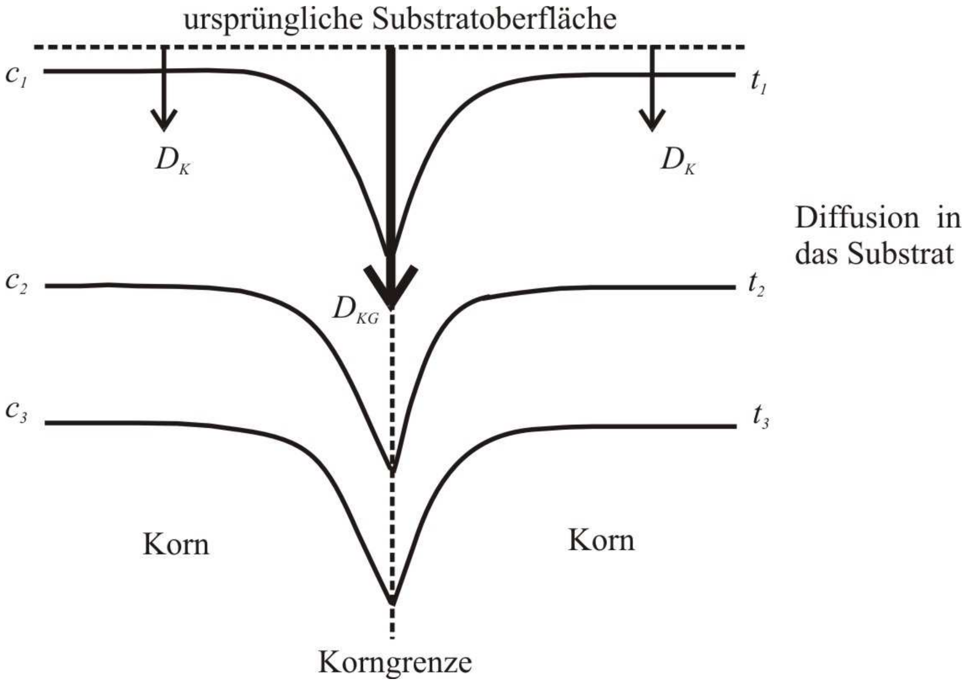


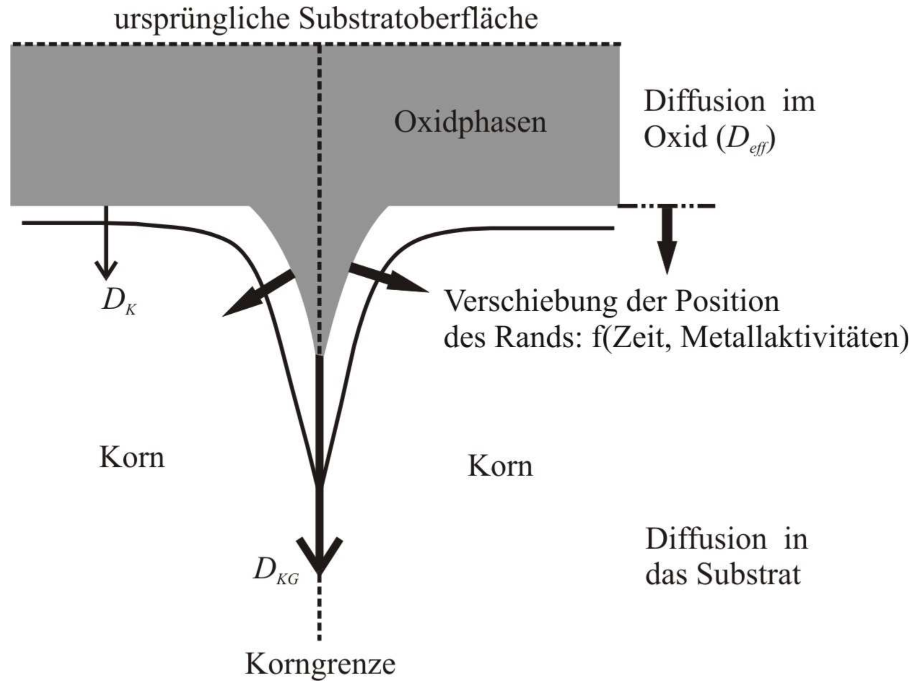

1.55789 Fe + 0.2502 Cr + 1.4 O2
C:\FactSage\Equi0.res 24Jun09
weight %
log10(activity) O2(g)
FeCr2O4(FeSp)
Fe2O3(M2O3)
Fe3O4(FeSp)
Cr2O3(M2O3)
100
90
80
70
60
50
40
30
20
10
0
-40
-30
-20
-10
0

Only Internal diffusion calculation

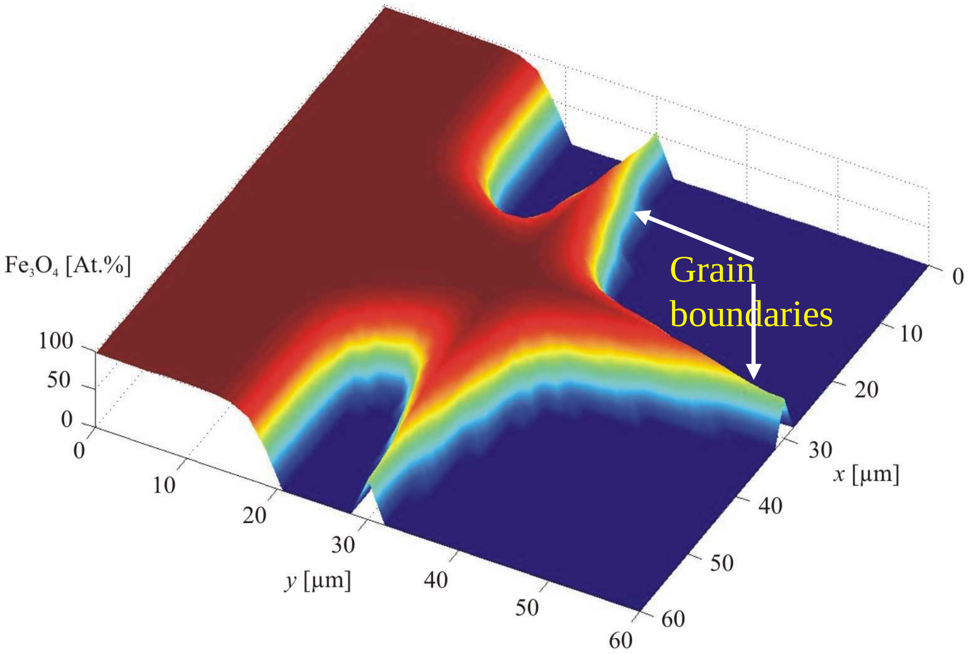


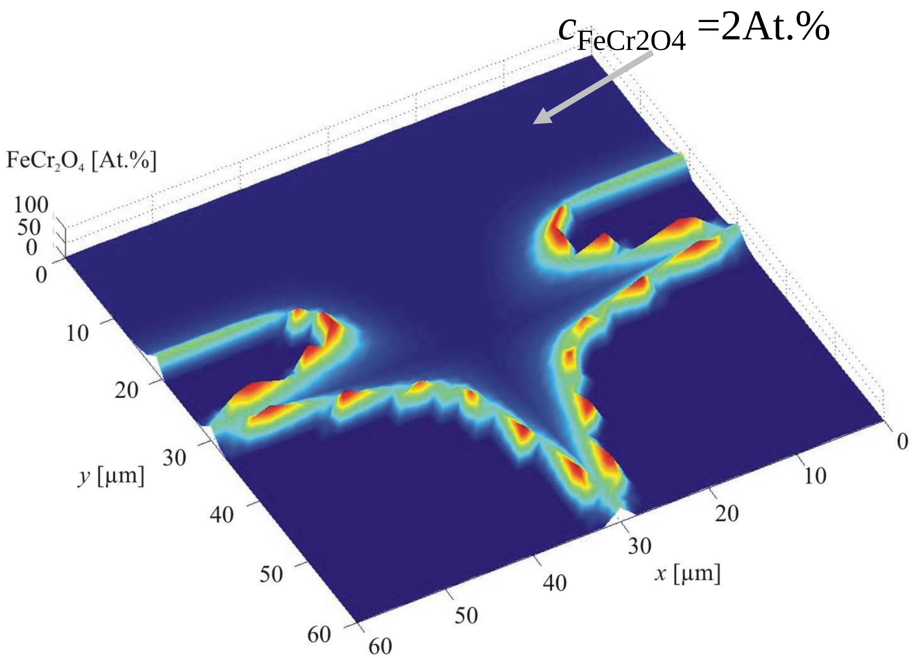
Inner layer growth





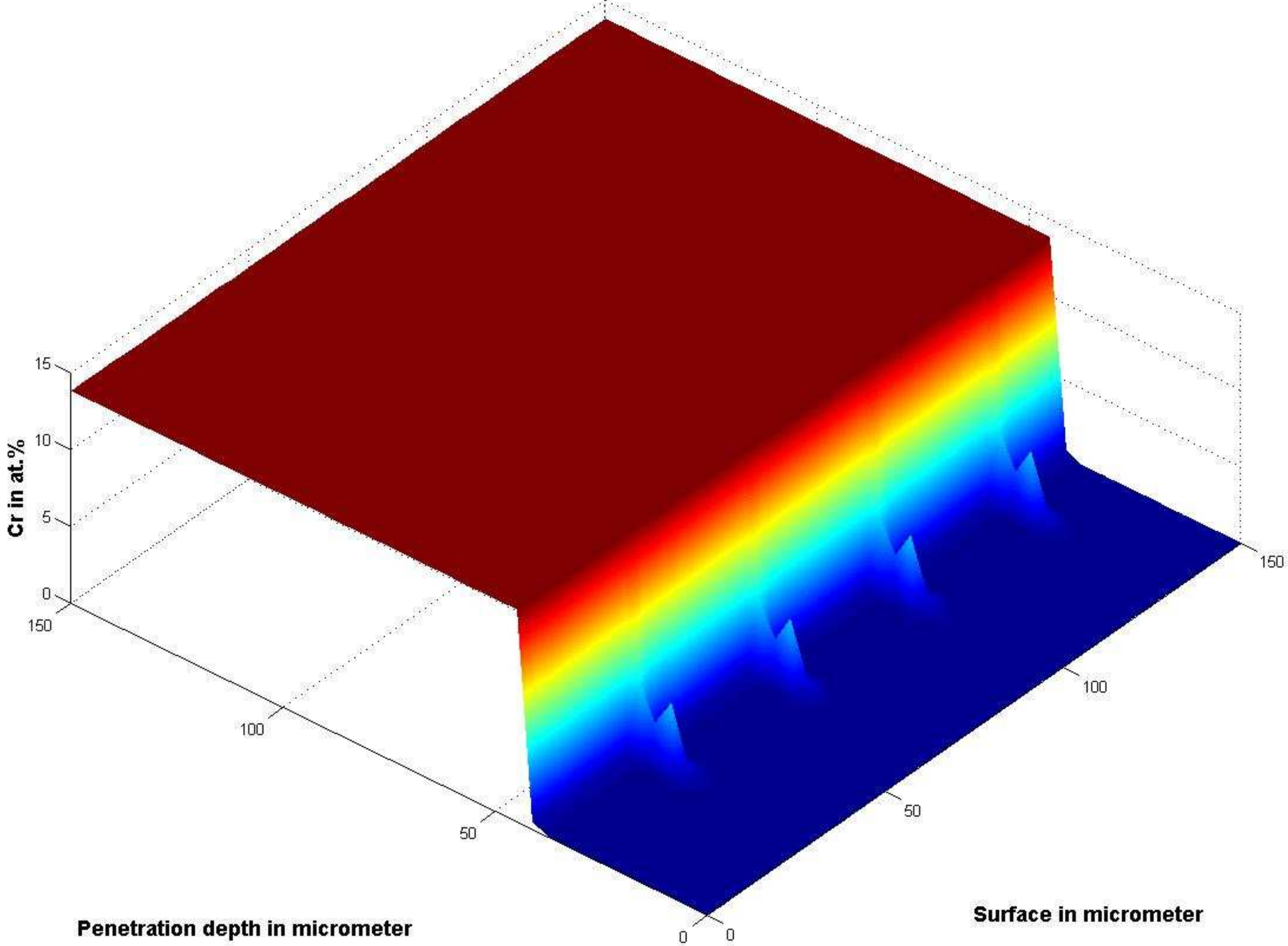
$$D_{eff} = \frac{\xi^2}{t\left[\ln\left(p(O_2)Fe_3O_4\right) - \ln\left(p(O_2)FeCr_2O_4\right)\right]}$$



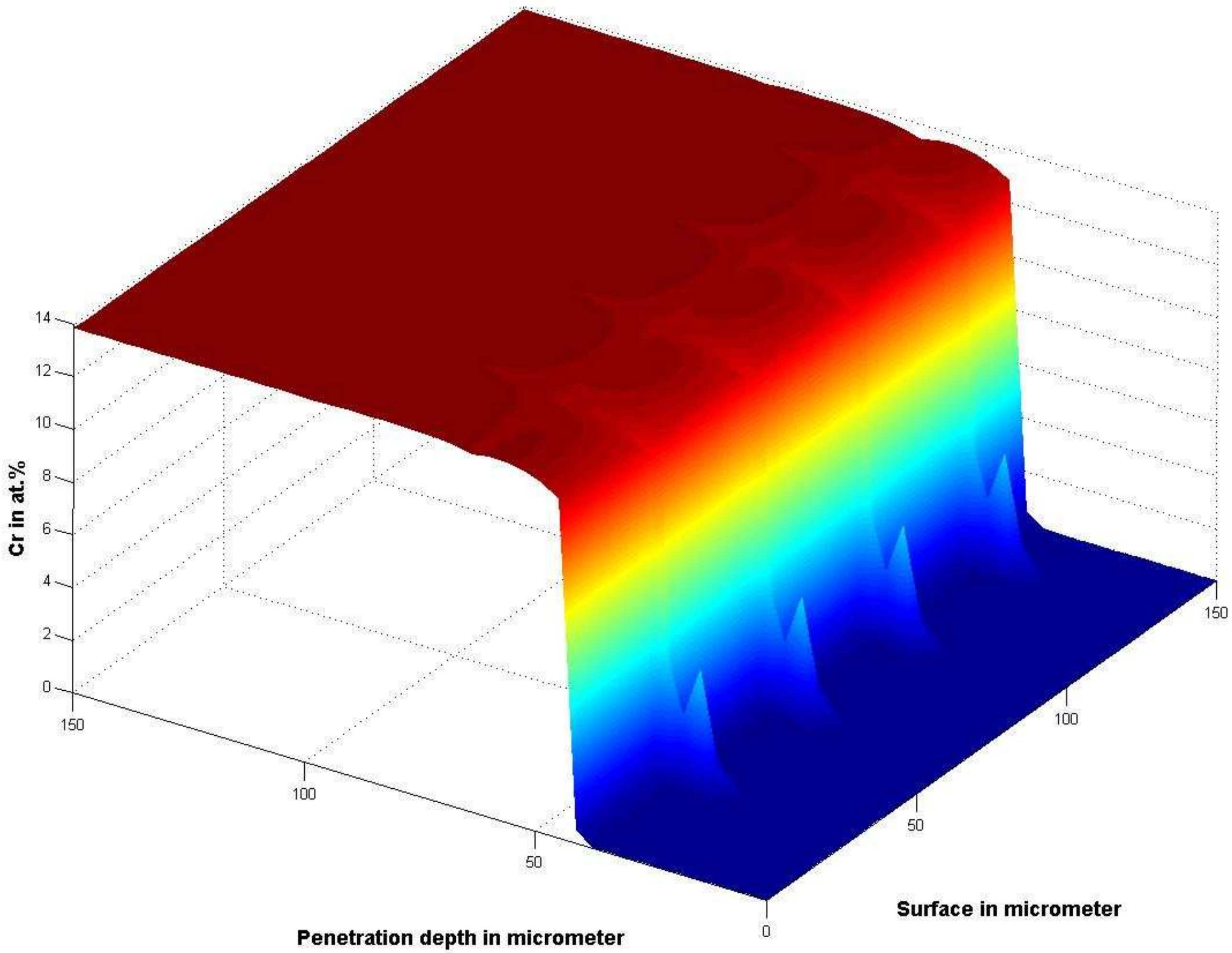


Duration = 270 hrs
T = 550^{o}C
d = 30 µm
Lab air

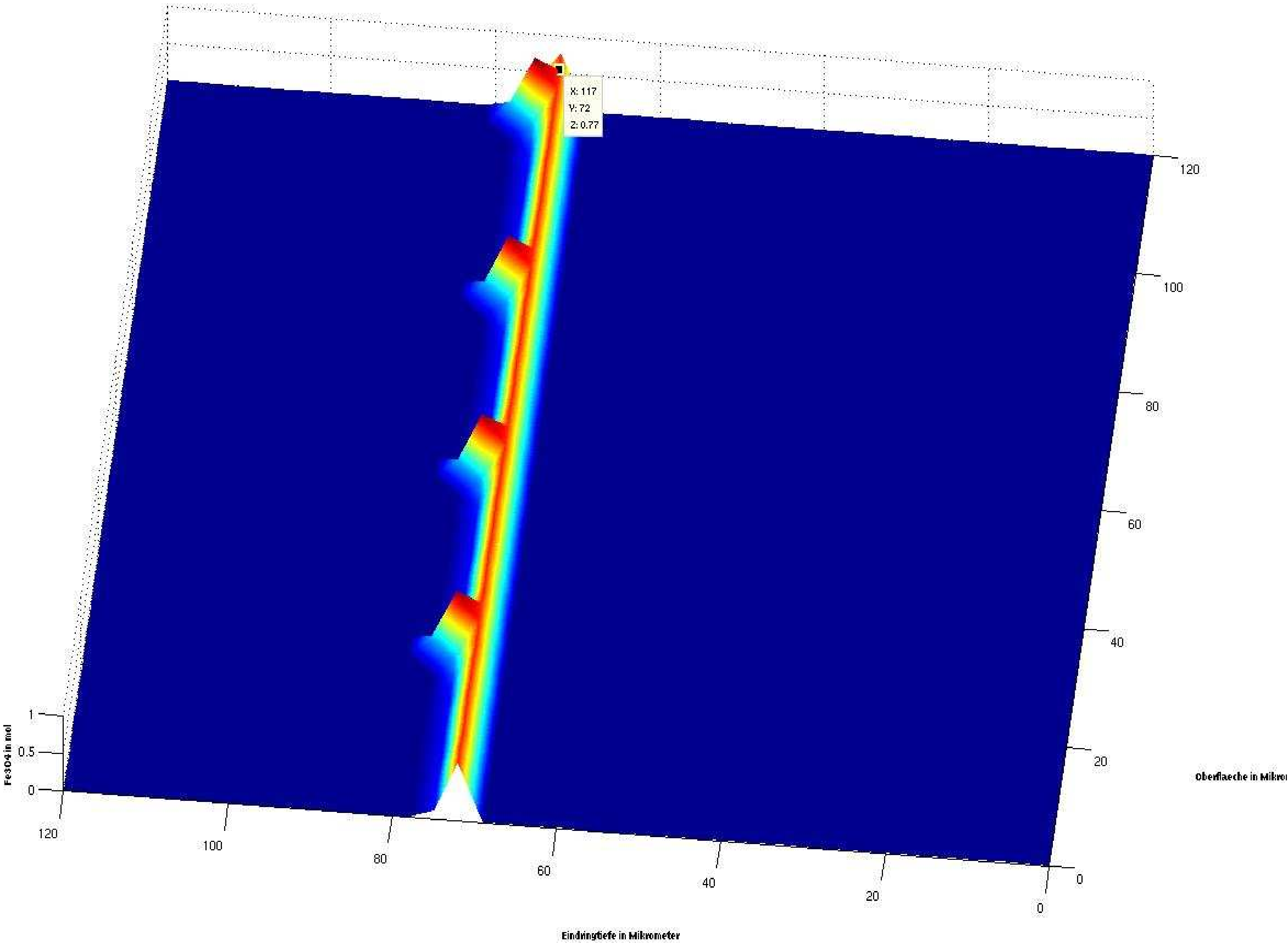
External diffusion calculation



Both internal and external diffusion



Lack of diffusion data in multi layered oxide scales



Recent work and future aspects

- Addition of outward scale growth
- Simulating the effect of shotpeening in KinCorr
- Effect of Water vapour on oxidation

Conclusions

oxidation and internal nitridation of steels

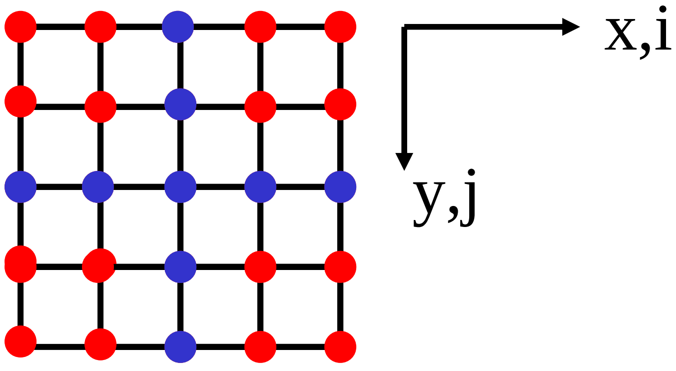
The developed software **KinCorr** is capable to account for: local thermodynamic equilibrium, solid state diffusion and alloy microstructure

carburization of steels

It can be calculated: concentration profiles of C, Cr, Fe, etc.
mass gain as a function of time

The high performance of the developed software ***KinCorr*** is sustained by a solid theoretical background.

Moving Interface Condition and Diffusion Matrix



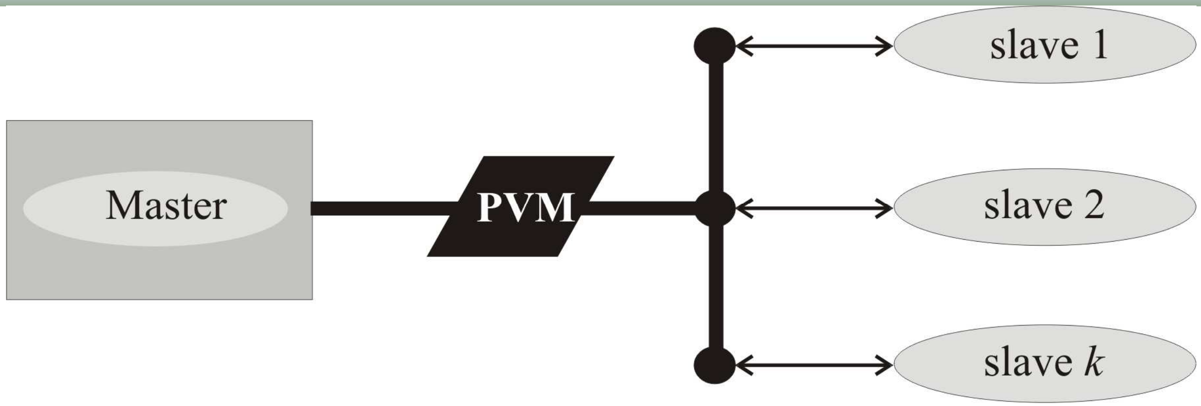
$$D_{i,j} = \begin{pmatrix} D11 & D12 & D13 & D14 & D15 \\ D21 & D22 & D23 & D24 & D25 \\ D31 & D32 & D33 & D34 & D35 \\ D41 & D42 & D43 & D44 & D45 \\ D51 & D52 & D53 & D54 & D55 \end{pmatrix}$$

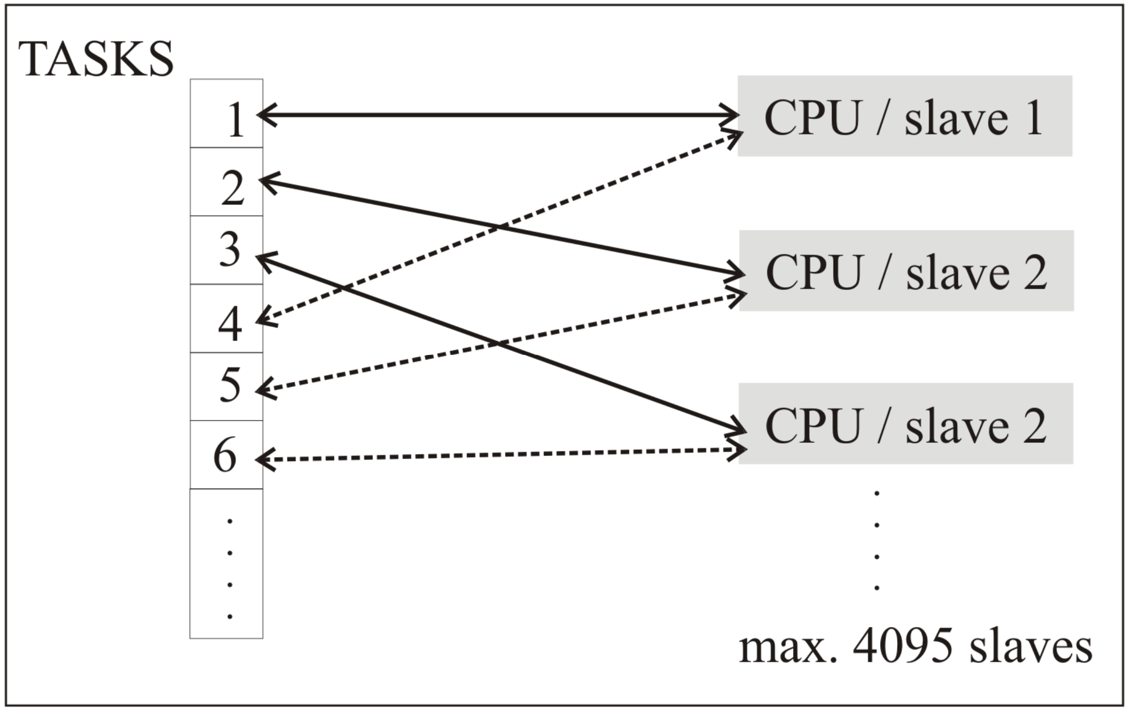
time = 100 h
dimension = 1 mm^2

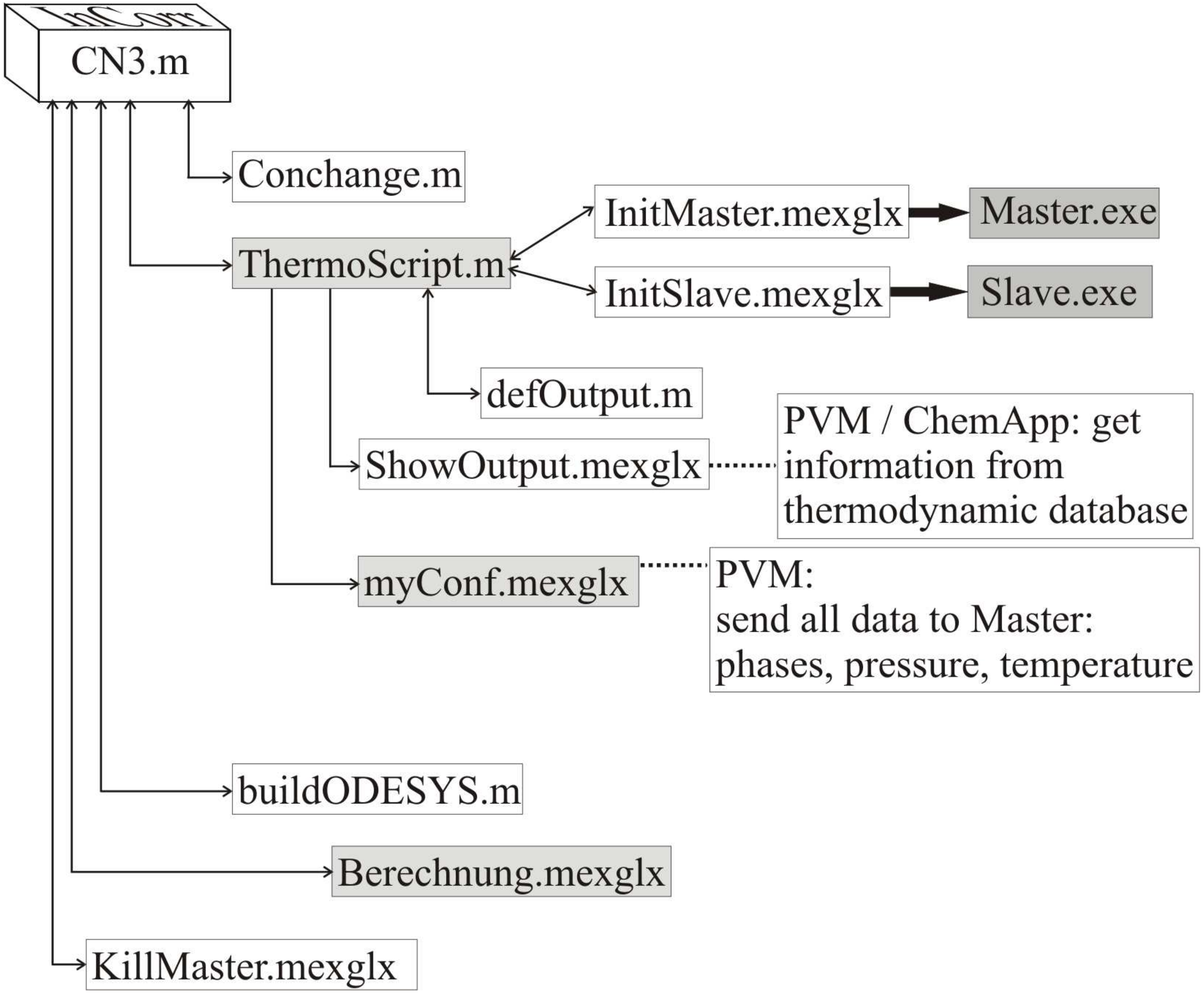
$\Delta x = \Delta y = 0.1\ \mu m$
$nx = ny = 10\ 000$
$\Delta t = 1\ s$
$nt = 360\ 000$

$$D_{i,j} = \begin{cases} Dx = Dy \text{ if } (i,j) \text{ is not a gb and } c_{Fe} > 0 \\ D_{gb} \text{ if } (i,j) \text{ is a gb and } c_{Fe} > 0 \\ D_{oxide} \text{ if in } (i,j)\ c_{Fe} = 0 \end{cases}$$

this must be checked 360 000 times at every (i,j)


Master
PVM
slave 1
slave 2
slave *k*


TASKS
1
2
3
4
5
6
CPU / slave 1
CPU / slave 2
CPU / slave 2
max. 4095 slaves


CN3.m
Conchange.m
ThermoScript.m
InitMaster.mexglx
Master.exe
InitSlave.mexglx
Slave.exe
defOutput.m
ShowOutput.mexglx
PVM / ChemApp: get information from thermodynamic database
myConf.mexglx
PVM: send all data to Master: phases, pressure, temperature
buildODESYS.m
Berechnung.mexglx
KillMaster.mexglx