

Crystal Growth: Physics, Technology and Modeling

Stanisław Krukowski & Michał Leszczyński
Institute of High Pressure Physics PAS
01-142 Warsaw, Sokołowska 29/37
e-mail: stach@unipress.waw.pl, mike@unipress.waw.pl

Zbigniew Żytkiewicz
Institute of Physics PAS
02-668 Warsaw, Al. Lotników 32/46
E-mail: zytkie@ifpan.edu.pl

Lecture 12. Growth modeling in macroscale

<http://w3.unipress.waw.pl/~stach/cg-2022-23/>

Growth modeling – two approaches

- **Modeling in macroscale**
 - transport processes during growth (mass, energy, momentum)
 - strain in nonuniform structures
 - electric properties of electronic structures and devices
 - optical properties of optoelectronic structures and devices
 - crystal morphology
- **Modeling in atomic scale**
 - crystal structure
 - energetic properties
 - kinetic properties
 - optical transitions

Growth modeling - methods

- **Modeling in macroscale**
 - finite difference
 - finite volume
 - finite element

- **Modeling in atomic scale**
 - Monte Carlo
 - molecular dynamics
 - ab initio – density functional theory (DFT)

Conservation laws – compressible fluid

$$\frac{\partial[\rho(\vec{r}, t)]}{\partial t} + \text{div}(\rho(\vec{r}, t)\vec{v}(\vec{r}, t)) = 0$$

$$\rho(\vec{r}, t) \left[\frac{\partial \vec{v}(\vec{r}, t)}{\partial t} + (\vec{v}(\vec{r}, t) \cdot \nabla) \vec{v}(\vec{r}, t) \right] = -\nabla p(\vec{r}, t) + \mu \Delta \vec{v}(\vec{r}, t) + \rho(\vec{r}, t) \vec{f}(\vec{r}, t)$$

$$\frac{\partial [C_p(T)T(\vec{r}, t)\rho(\vec{r}, t)]}{\partial t} + \rho(\vec{r}, t)C_p(T)(\vec{v}(\vec{r}, t) \cdot \nabla)T(\vec{r}, t) = \text{div}(\kappa \nabla T(\vec{r}, t)) + r_\varepsilon$$

- **6 variables: 3 velocity components, density, pressure, temperature**
- **5 equations of motion + equation of state**

$$p = p(\rho, T)$$

Conservation laws – incompressible fluids

$$\text{div}(\vec{v}(\vec{r}, t)) = 0$$

$$\rho_o \left[\frac{\partial \vec{v}(\vec{r}, t)}{\partial t} + (\vec{v}(\vec{r}, t) \cdot \nabla) \vec{v}(\vec{r}, t) \right] = \mu \Delta \vec{v}(\vec{r}, t) + \rho_o [\beta_T (T - T_o) + \beta_c c] \vec{f}(\vec{r}, t)$$

$$\rho_o C_p(T) \left[\frac{\partial [T(\vec{r}, t)]}{\partial t} + (\vec{v}(\vec{r}, t) \cdot \nabla) T(\vec{r}, t) \right] = \text{div}(\kappa \nabla T(\vec{r}, t)) + r_\varepsilon$$

- **5 variables: 3 velocity components, pressure, temperature**
- **5 equations**
- **Additional equation:**

$$\rho = \rho_o$$

Boundary conditions - velocity

- **Solid surfaces – no-slip condition:**

$$\vec{v}(\vec{r}, t) = 0$$

- **Solid surface – crystal growth (surface is nonmaterial, no-slip):**

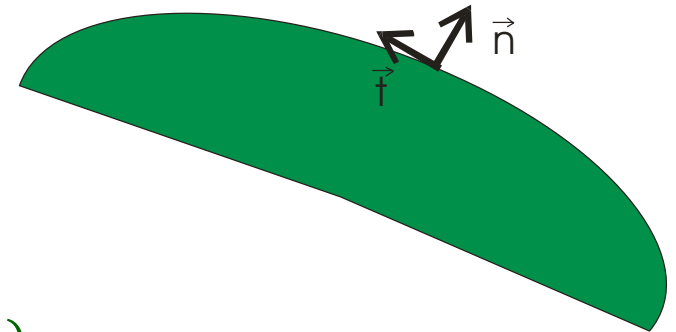
$$\vec{v}(\vec{r}, t) \cdot \vec{t}(\vec{r}, t) = 0$$

$$\rho_l(\vec{r}, t) [\vec{v}(\vec{r}, t) c_l(\vec{r}, t) - D_l \nabla c_l(\vec{r}, t)] \cdot \vec{n}(\vec{r}, t) = \rho_s(\vec{r}, t) [\vec{u}(\vec{r}, t) c_s(\vec{r}, t) - D_s \nabla c_s(\vec{r}, t)] \cdot \vec{n}(\vec{r}, t)$$

$\vec{t}(\vec{r}, t)$ - vector tangential to the surface

$\vec{n}(\vec{r}, t)$ - vector normal to the surface

$\vec{u}(\vec{r}, t)$ - crystallization velocity



Boundary conditions - temperature

- **Solid –vapor/liquid interface – perfect thermal contact:**

$$T_l(\vec{r}, t) = T_s(\vec{r}, t)$$

- **Solid surface – crystal growth (surface is nonmaterial, no-slip):**

$$[C_l \rho_l(\vec{r}, t) \vec{v}_l(\vec{r}, t) - C_s \rho_s(\vec{r}, t) \vec{v}_s(\vec{r}, t)] \cdot \vec{n}(\vec{r}, t) = \\ [\kappa_l \nabla T_l(\vec{r}, t) - \kappa_s \nabla T_s(\vec{r}, t) + \rho_s(\vec{r}, t) \vec{u}(\vec{r}, t) H] \cdot \vec{n}(\vec{r}, t) + Q$$

H – latent heat

Q – radiation flux

Boundary conditions – mathematical categories

- **Dirichlet condition**

$$\varphi_l(\vec{r}, t) = \varphi_s(\vec{r}, t)$$

- **Neumann condition**

$$\nabla\varphi_s(\vec{r}, t) \cdot \vec{n}(\vec{r}, t) = f_s(\vec{r}, t)$$

- **Mixed condition**

$$F[\nabla\varphi_s(\vec{r}, t) \cdot \vec{n}(\vec{r}, t)] + G[\varphi_s(\vec{r}, t)] = f_s(\vec{r}, t)$$

Boundary conditions – physical interpretation

- **Dirichlet boundary condition: temperature, concentration – local equilibrium between phases**

$$T_l(\vec{r}, t) = T_s(\vec{r}, t) \qquad C_l(\vec{r}, t) = kC_s(\vec{r}, t)$$

- **Dirichlet boundary condition – tangential component of the velocity disappears**

$$\vec{v}(\vec{r}, t) \cdot \vec{t}(\vec{r}, t) = 0$$

- **Neumann boundary condition – predetermined flows such as crystallization velocity, dissolution, flux**

$$D\nabla C(\vec{r}, t) \cdot \vec{n}(\vec{r}, t) = R(\vec{r}, t)$$

- **Mixed condition – crystallization velocity in function of the supersaturation**

$$D\nabla C(\vec{r}, t) \cdot \vec{n}(\vec{r}, t) = k\sigma = k \left(\frac{C - C_{eq}}{C_{eq}} \right)$$

Solution methods – approximation

- Conservation law: classical field ϕ – mathematical structure

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial(v_i\rho\phi)}{\partial r_i} = \frac{\partial}{\partial r_j} \left[\Gamma \frac{\partial\phi}{\partial r_j} \right] + q_\phi$$

Approximate algorithm

- Scalar field is replaced by its representation in the mesh sites - **pre-processing**
- Boundary condition are enforced – **pre-processing**
- Solution - **processing**
- Result – represented by continuous functions - **post-processing**

Mesh generation - preprocessing

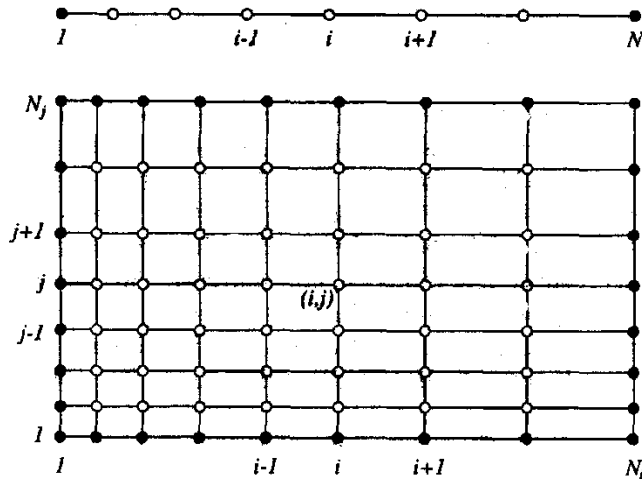
Mesh types:

- **Regular (structural)**
- **Block-regular (block-structural)**
- **Complex**
- **Irregular**

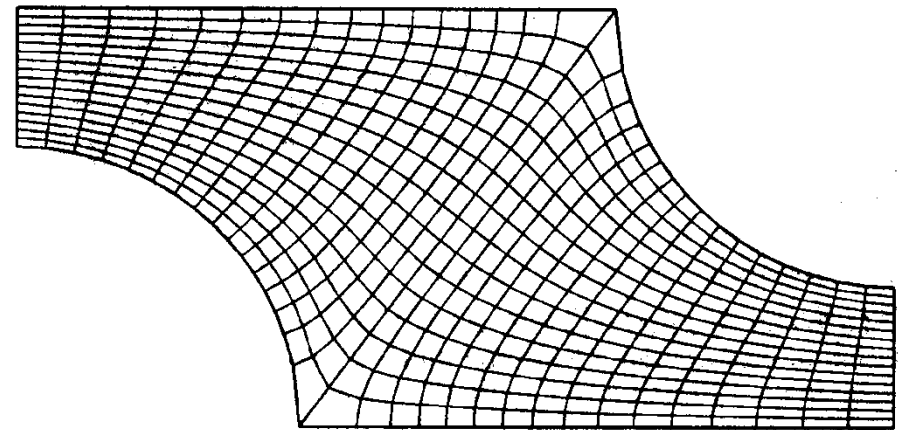
Regular mesh (structural)

Regular mesh – coordinate system

- Lines belonging to the same family do not cross
- Lines belonging to different families cross once only



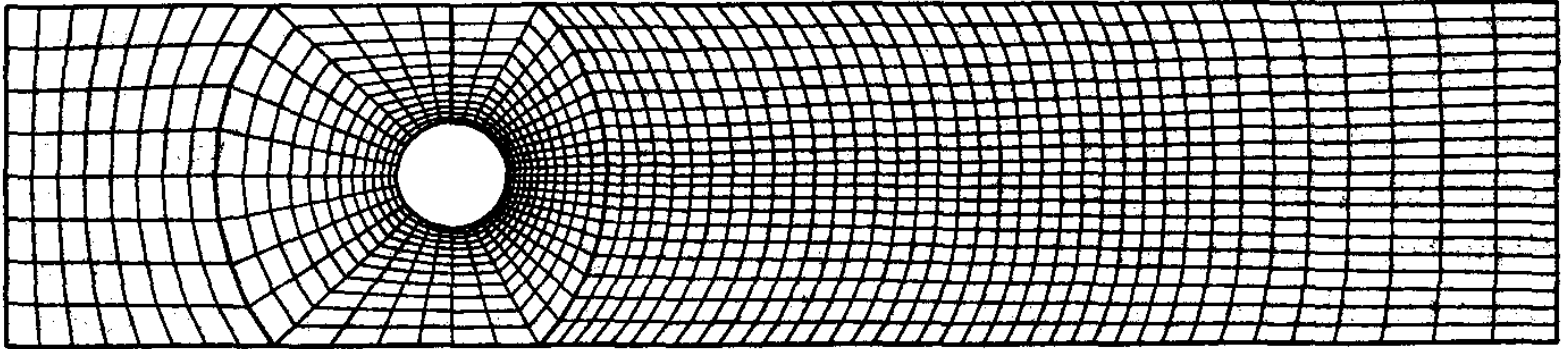
*Examples: 1-d and 2-d
regular orthogonal mesh*



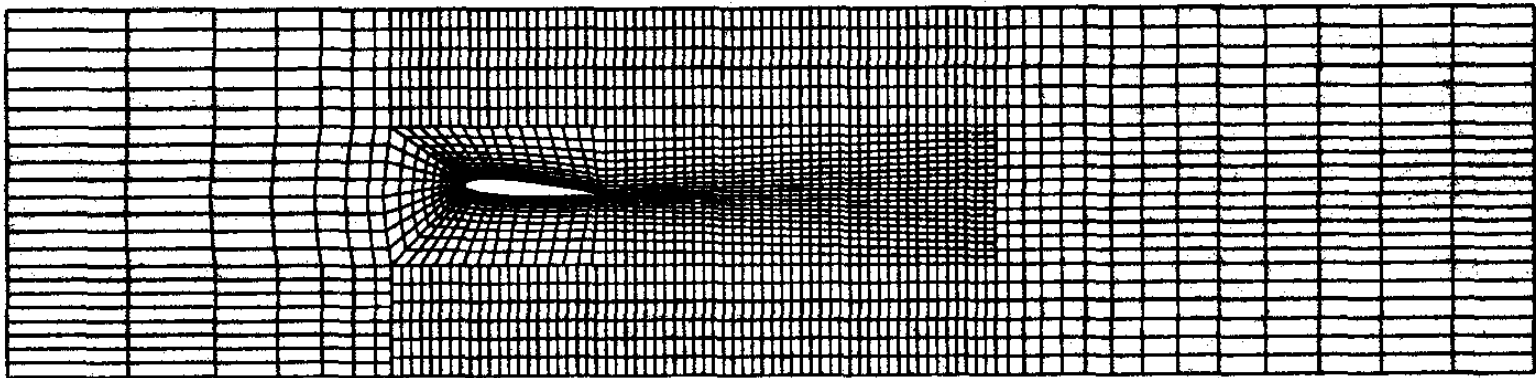
*Example: 2-d regular nonorthogonal mesh
designed for calculation of the flow
between two parallel pipes*

Block-regular mesh

- Segments are regular – the sum is not



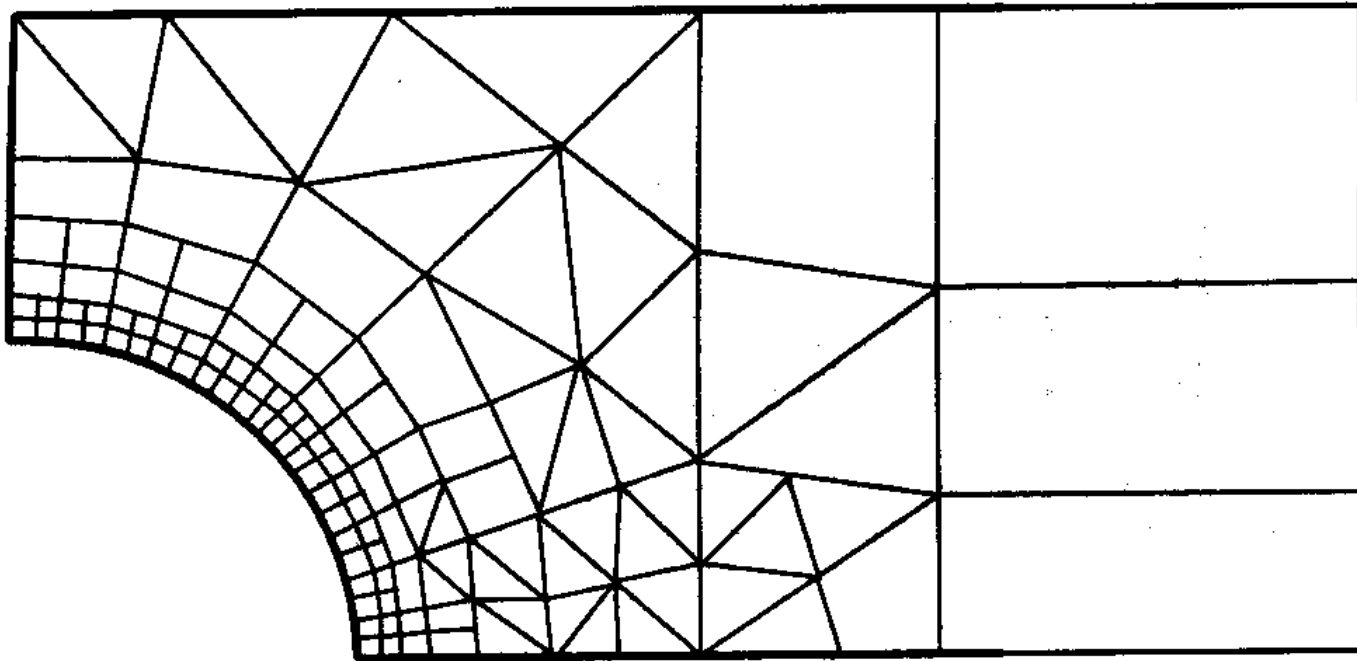
Example: block-regular nonorthogonal mesh. The sites are compatible.



Example: block-regular nonorthogonal mesh. The sites are not compatible.

Non-structural mesh

- Mesh filling any volume (area), e.g. triangular mesh



Example: 2-d mesh composed of triangular and tetra-angular elements

- Extension – adaptive mesh

Solution methods – algorithms

- **Conservation law: classical field ϕ**

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial(v_i\rho\phi)}{\partial r_i} = \frac{\partial}{\partial r_j} \left[\Gamma \frac{\partial\phi}{\partial r_j} \right] + q_\phi$$

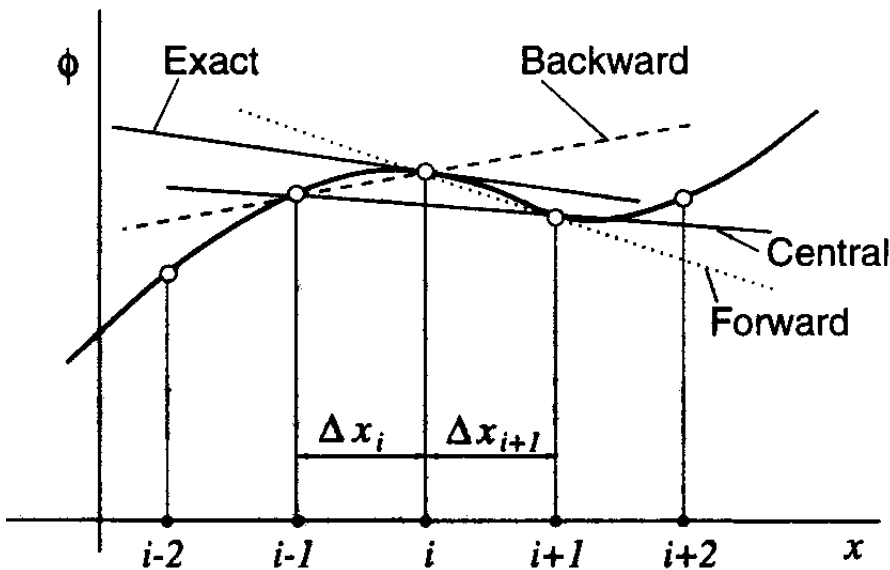
- **Finite difference method**
- **Finite volume method**
- **Finite element method**

Finite difference method

- L. Euler (XVIII century) – derivative are replaced by finite difference between sites

$$\frac{\partial \phi}{\partial r_i} = \lim_{\Delta r_i \rightarrow 0} \frac{\phi(r_i + \Delta r_i) - \phi(r_i)}{\Delta r_i}$$

Approximations



- **Backward difference scheme(BDS)**

$$\left(\frac{\partial \phi}{\partial r_j(i)} \right) = \frac{\phi(i) - \phi(i-1)}{r_j(i) - r_j(i-1)}$$

- **Central difference scheme (CDS)**

$$\left(\frac{\partial \phi}{\partial r_j(i)} \right) = \frac{\phi(i+1) - \phi(i-1)}{r_j(i+1) - r_j(i-1)}$$

- **Forward difference scheme (FDS)**

$$\left(\frac{\partial \phi}{\partial r_j(i)} \right) = \frac{\phi(i+1) - \phi(i)}{r_j(i+1) - r_j(i)}$$

Finite difference method – higher derivatives

- 2nd order - CDS between $r_j(i-1/2)$, $r_j(i+1/2)$:

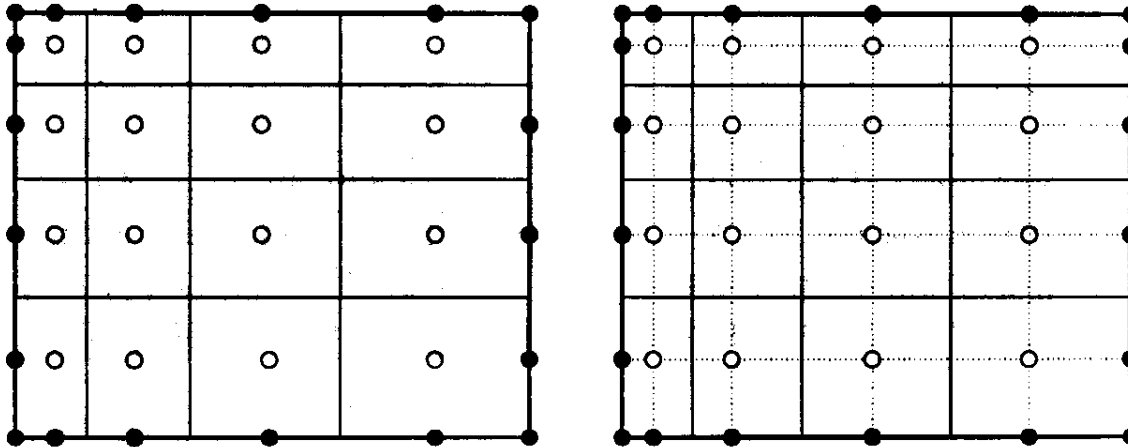
$$\left(\frac{\partial \phi}{\partial r_j(i+1/2)} \right) = \frac{\phi(i+1) - \phi(i)}{r_j(i+1) - r_j(i)}$$

$$\left(\frac{\partial \phi}{\partial r_j(i-1/2)} \right) = \frac{\phi(i) - \phi(i-1)}{r_j(i) - r_j(i-1)}$$

$$\begin{aligned} \left(\frac{\partial \phi}{\partial r_j(i)} \right) &= \frac{\left(\frac{\partial \phi}{\partial r_j(i+1/2)} \right) - \left(\frac{\partial \phi}{\partial r_j(i-1/2)} \right)}{[r_j(i+1) - r_j(i-1)]/2} \\ &= \frac{\phi(i+1)[r_j(i) - r_j(i-1)] + \phi(i-1)[r_j(i+1) - r_j(i)] - 2\phi(i)[r_j(i+1) - 2r_j(i) + r_j(i-1)]}{[r_j(i+1) - r_j(i-1)][r_j(i+1) - r_j(i)][r_j(i) - r_j(i-1)]/2} \end{aligned}$$

Finite volume method

- Full system is divided into control volumes (CV)



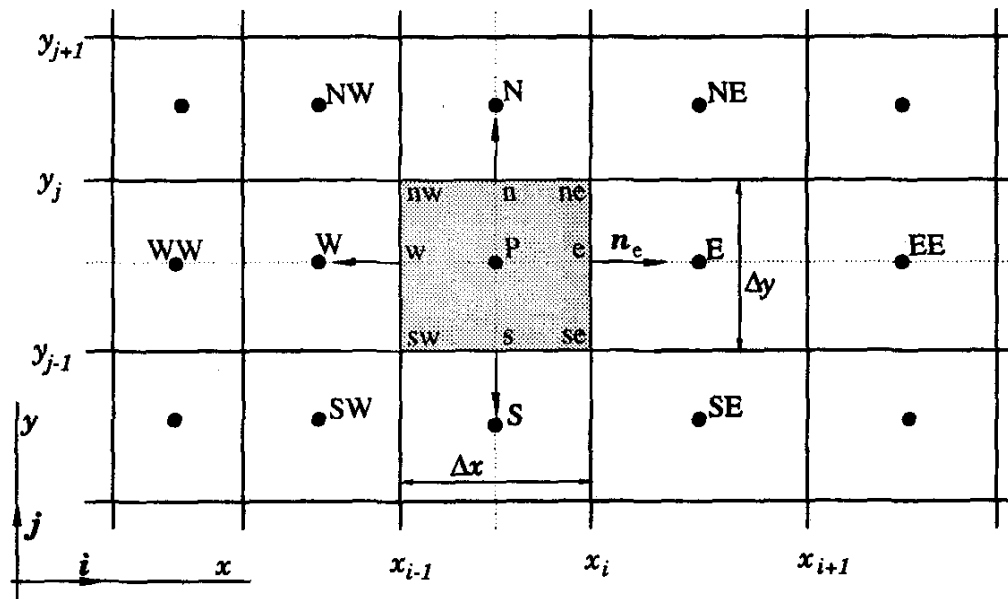
Different relations between mesh sites and CV:

Left – on the center of CV

Right – walls centered between the sites

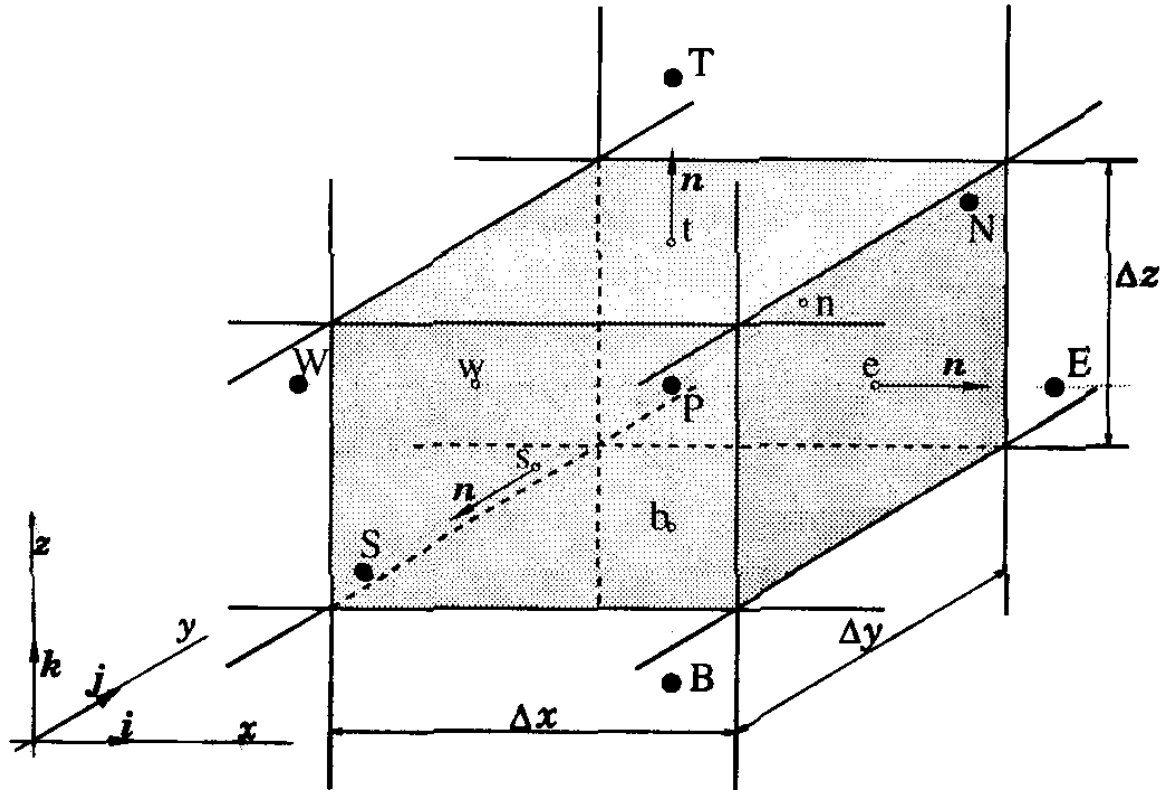
Finite volume method – implementation 2D

- Field values are determined in the center only



Definition of CV and notation in 2-d Cartesian mesh

Finite volume method– implementation 3D



Definition of CV and notation in 3-d Cartesian mesh

Finite volume method - integration

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial(v_i\rho\phi)}{\partial r_i} = \frac{\partial}{\partial r_j} \left[\Gamma \frac{\partial\phi}{\partial r_j} \right] + q_\phi$$

$$\frac{\partial(v_i\rho\phi)}{\partial r_i} = \frac{\partial}{\partial r_j} \left[\Gamma \frac{\partial\phi}{\partial r_j} \right] + q_\phi$$

- Equation of motion are integrated over CV

$$\int_S \rho\phi(\vec{v} \cdot \vec{n}) d^2r = \int_S \phi(\nabla\phi \cdot \vec{n}) d^2r + \int_V s_\phi d^3r$$

- Surface integrals

$$\int_{S_e} f d^2r = f_e S_e$$

- Volume integrals

$$\int_\Omega q d^3r = q_p \Omega$$

Finite element method (FEM) - properties

- **Strong differential & weak integral equations are equivalent**
- **Weak form – any function $w(\mathbf{r},t)$**

Solution – sum (combination) of functions defined on the element (any shape)

- **Interpolation function – any field value**
- **Interpolation function – any gradient value**
- **Field – continuous function of the coordinates**

Finite element method (FEM)

- **Equation of motion - differential form**

$$\frac{\partial}{\partial r_j} \left[\Gamma \frac{\partial \phi}{\partial r_j} \right] + q_\phi = 0$$

- **Equation of motion – integral (weak) form (w – any function)**

$$\int_V w \left\{ \frac{\partial}{\partial r_j} \left[\Gamma \frac{\partial \phi}{\partial r_j} \right] + q_\phi \right\} d^3r = 0$$

Green theorem gives

$$\int_S \left\{ \left[w \Gamma n_j \frac{\partial \phi}{\partial r_j} \right] \right\} d^2r + \int_V \left\{ w q_\phi - \frac{\partial w}{\partial r_j} \left[\Gamma \frac{\partial \phi}{\partial r_j} \right] \right\} d^3r = 0$$

Interpolation functions – linear (1-d & 2-d)

$$\phi(i_1) = \phi[x(i_1)]$$

$$\phi(i_2) = \phi[x(i_2)]$$

- **Field in linear interpolation**

$$\phi(x) = \phi(i_1) \frac{x - x(i_2)}{x(i_1) - x(i_2)} + \phi(i_2) \frac{x - x(i_1)}{x(i_2) - x(i_1)} = \phi(i_1)u_{i_1}(x) + \phi(i_2)u_{i_2}(x)$$

- **Universal interpolation functions**

$$u_{i_1}(x) = \frac{x - x(i_2)}{x(i_1) - x(i_2)} \quad \longrightarrow \quad u_{i_1}(x_{i_1}) = 1 \quad u_{i_1}(x_{i_2}) = 0$$

$$u_{i_2}(x) = \frac{x - x(i_1)}{x(i_2) - x(i_1)} \quad \longrightarrow \quad u_{i_2}(x_{i_1}) = 0 \quad u_{i_2}(x_{i_2}) = 1$$

Interpolation functions – linear (2-d)

$$\phi(i_1) = \phi[x(i_1)]$$

$$\phi(i_2) = \phi[x(i_2)]$$

- **Field in linear interpolation**

$$\phi(x) = \phi(i_1) \frac{x - x(i_2)}{x(i_1) - x(i_2)} + \phi(i_2) \frac{x - x(i_1)}{x(i_2) - x(i_1)} = \phi(i_1)u_{i_1}(x) + \phi(i_2)u_{i_2}(x)$$

- **Universal interpolation functions**

$$u_{i_1}(x) = \frac{x - x(i_2)}{x(i_1) - x(i_2)} \quad \longrightarrow \quad u_{i_1}(x_{i_1}) = 1 \quad u_{i_1}(x_{i_2}) = 0$$

$$u_{i_2}(x) = \frac{x - x(i_1)}{x(i_2) - x(i_1)} \quad \longrightarrow \quad u_{i_2}(x_{i_1}) = 0 \quad u_{i_2}(x_{i_2}) = 1$$

Interpolation functions – linear (2-d)

$$\phi(i_1, j_1) = \phi[x(i_1), y(j_1)]$$

$$\phi(i_2, j_1) = \phi[x(i_2), y(j_1)]$$

$$\phi(i_1, j_2) = \phi[x(i_1), y(j_2)]$$

$$\phi(i_2, j_2) = \phi[x(i_2), y(j_2)]$$

- **Field in linear interpolation**

$$\phi(x, y) = \phi(i_1, j_1) \frac{x - x(i_2)}{x(i_1) - x(i_2)} \frac{y - y(i_2)}{y(i_1) - y(i_2)} + \dots = \phi(i_1, j_1) u_{i_1}(x) u_{i_1}(y) + \dots$$

- **Universal interpolation functions**

$$u_{i_1}(x) = \frac{x - x(i_2)}{x(i_1) - x(i_2)}$$

$$u_{j_1}(y) = \frac{y - y(j_2)}{y(j_1) - y(j_2)}$$

$$u_{i_2}(x) = \frac{x - x(i_1)}{x(i_1) - x(i_2)}$$

$$u_{j_2}(y) = \frac{y - y(j_1)}{y(j_1) - y(j_2)}$$

Weight function $w(\mathbf{r}, t)$

- Approximate solution gives residue R

$$\int_S \left\{ \left[w \Gamma n_j \frac{\partial \phi}{\partial r_j} \right] \right\} d^2 r + \int_V \left\{ w q_\phi - \frac{\partial w}{\partial r_j} \left[\Gamma \frac{\partial \phi}{\partial r_j} \right] \right\} d^3 r = \int_V R w d^3 r$$

- Galerkin method – weight function is solution

$$w(\vec{r}, t) = \phi(\vec{r}, t) = \sum c_i u_i(\vec{r}, t)$$

- Additional condition – Residue is orthogonal to weight function

$$\int_V R w d^3 r = 0$$

- Final equation – array equation

$$K\phi = f$$

Array equation

$$K\phi = f$$

$$\phi \sum_j \int_V d^3r [\Gamma(\nabla u_i \cdot \nabla u_j)] \phi(j) = \int_V d^3r [u_i q_\phi] + \int_S d^2r \{\vec{n} \cdot [u_i \Gamma \nabla(\phi)]\}$$

- **Rigidity matrix K:**

$$K_{ij} = \int_V d^3r [\Gamma(\nabla u_i \cdot \nabla u_j)]$$

- **Force vector f:**

$$f = f^s + f^b$$

- **Source vector f^s:**

$$f^s = \int_V d^3r [u_i \cdot q_\phi]$$

- **Boundary condition vector f^b:**

$$f^b = \int_S d^2r \{\vec{n} \cdot [u_i \nabla(\phi)]\}$$

Solution of nonlinear array equation – linear algebra

$$K(\phi)\phi = f$$

- **Initial solution - linear**

$$K_o = K(\phi = 0) \quad \Rightarrow \quad K_o\phi = f \quad \Rightarrow \quad \phi = K_o^{-1}f$$

- **SS –successive substitutions**

$$K(\phi_0)\phi_1 = f \quad \Rightarrow \quad \phi_1 = K^{-1}(\phi_0)f \Rightarrow \quad \phi_2 = K^{-1}(\phi_1)f$$

- **Global methods: Newton-Raphson, Quasi-Newton**
- **Partial methods: Segregated solver**

- **Solution strategy (small system):**

SS – first (slow convergence, extensive convergence radius)

QN, NR – finish (fast convergence, small convergence radius)

- **Solution strategy (large system):**

Segregated solver

Convergence conditions

$$K(\phi)\phi = f$$

- At any iteration (i), the following equation is fulfilled :

$$K(\phi)\phi = f + r$$

r_i - residuum – of the norm:

$$R_i = \|r_i\| = \left(\sum_{\alpha} r_{\alpha,i}^2 \right)^{1/2} \quad \frac{\|R_i(\phi)\|}{\|R_o\|} \leq \varepsilon$$

- Convergence criteria:

Relative residuum measure

$$\frac{R_i}{R_o} \leq \varepsilon$$

Relative solution change in subsequent iterations

$$\frac{\|u_{i+1} - u_i\|}{\|u_i\|} \leq \varepsilon$$

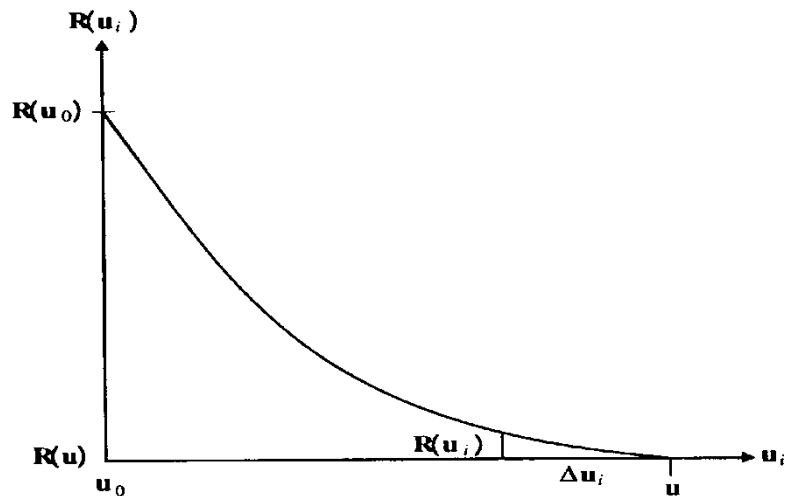
Types of convergence

- Asymptotic convergence is defined as:

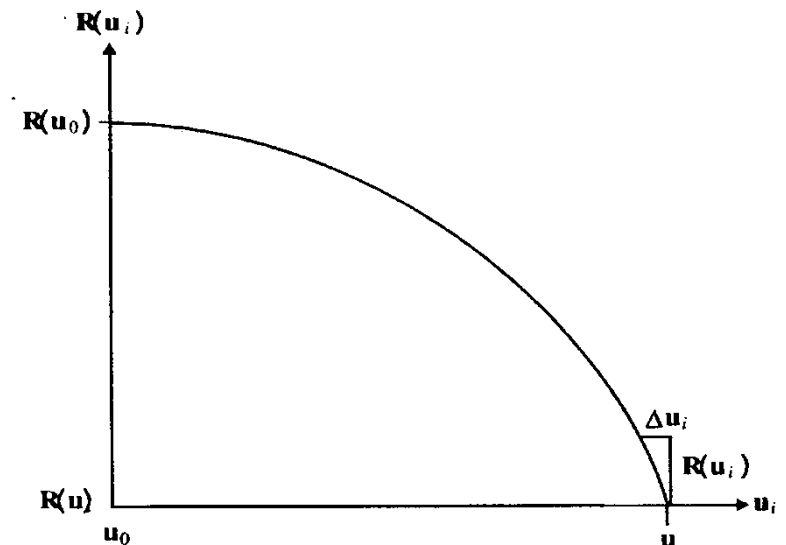
$$\|\phi_{i+1} - \phi_i\| \leq V \|\phi_i - \phi_{i-1}\|^k$$

k – convergence exponent: $k=1 \rightarrow$ linear (SS), $k=2 \rightarrow$ parabolic (NR, QN)

Residuum criterion



Relative change criterion



Example 1 : isothermal forced flow – gas mixer

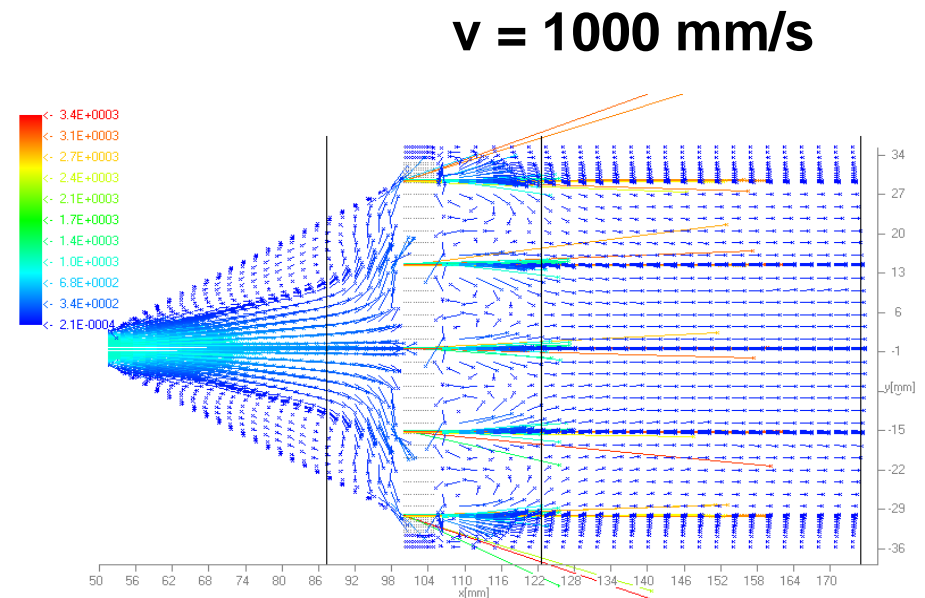
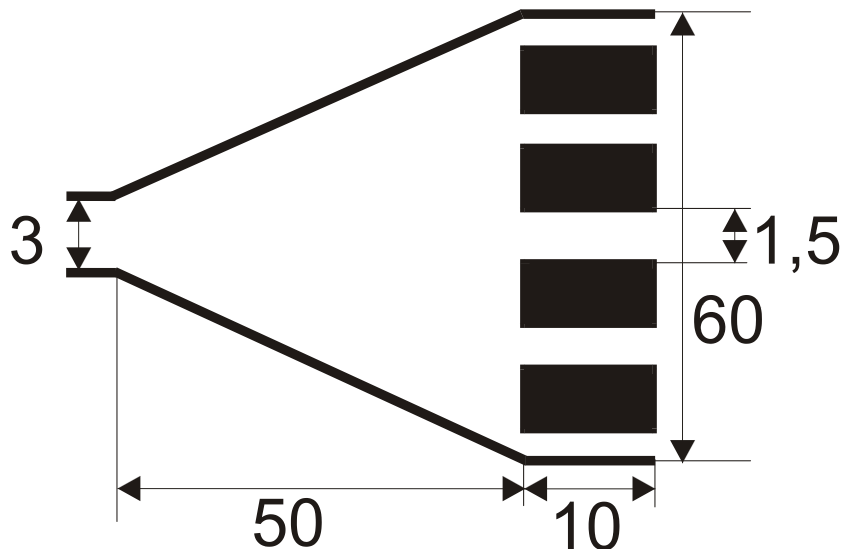
- Equation motion isothermal incompressible fluid:

$$\text{div}(\vec{v}(\vec{r}, t)) = 0$$

$$\rho_o \left[\frac{\partial \vec{v}(\vec{r}, t)}{\partial t} + (\vec{v}(\vec{r}, t) \cdot \nabla) \vec{v}(\vec{r}, t) \right] = \mu \Delta \vec{v}(\vec{r}, t)$$

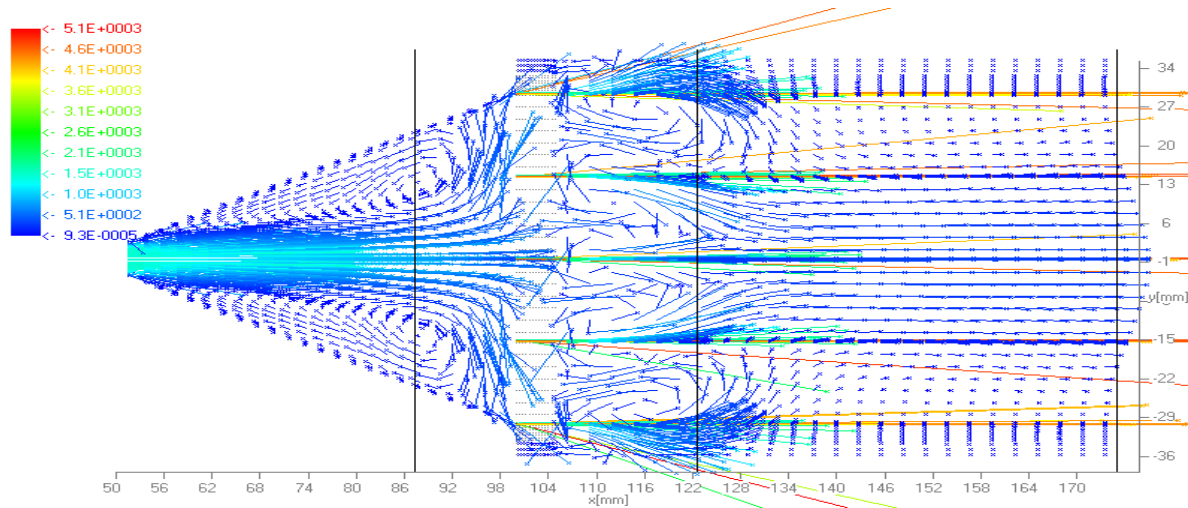
- Solid surfaces – no-slip condition:

$$\vec{v}(\vec{r}, t) = 0$$

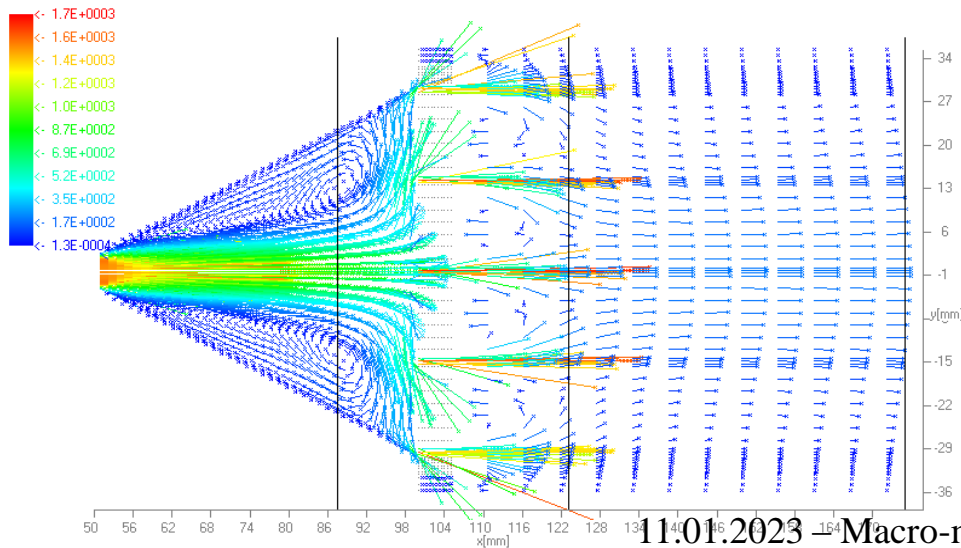


Isothermal forced flow – error

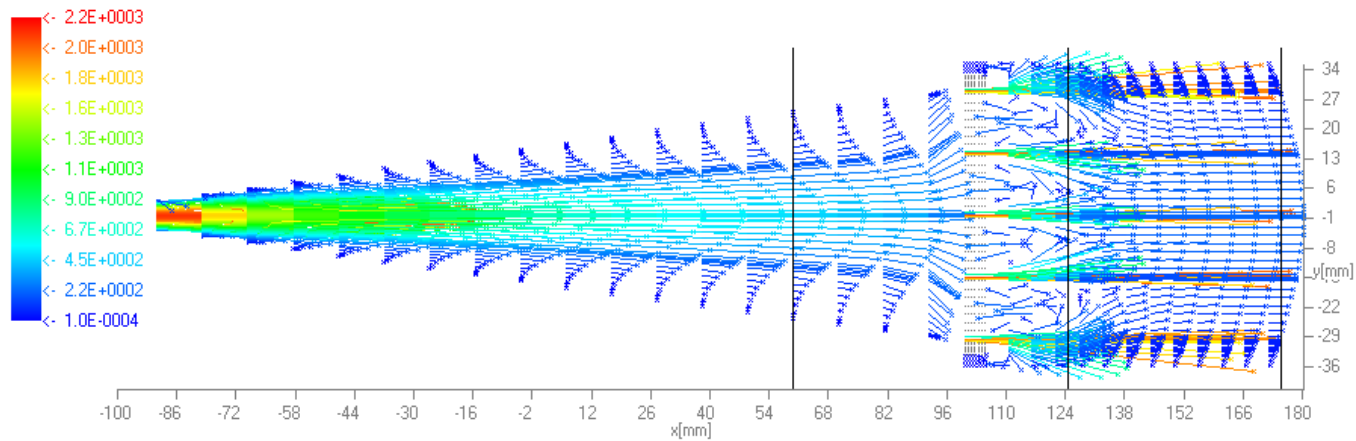
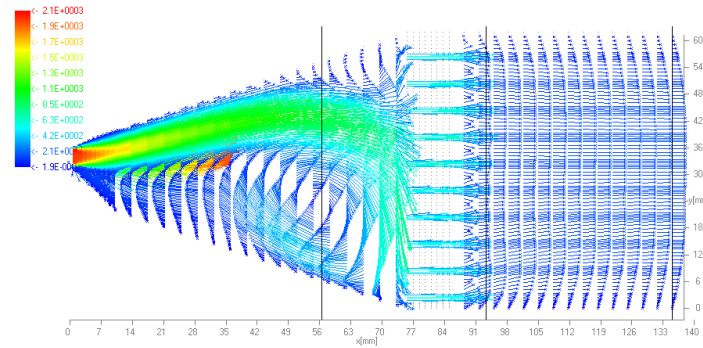
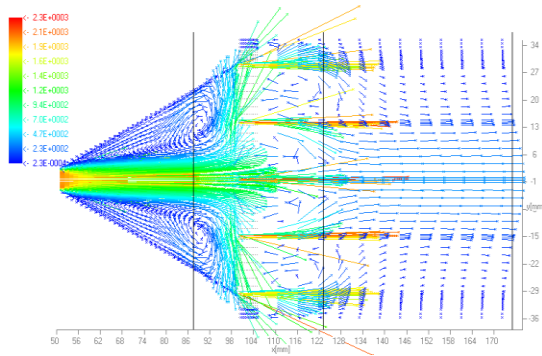
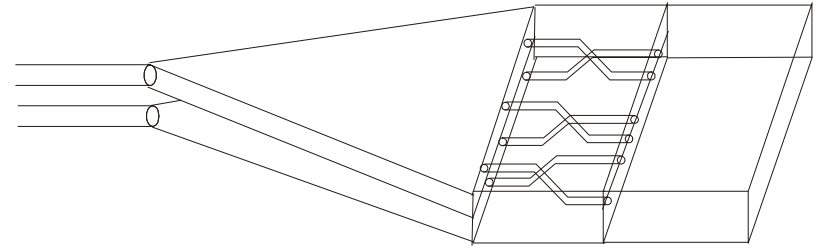
- **High flow – high error**



- **Small flow – small error**



New design



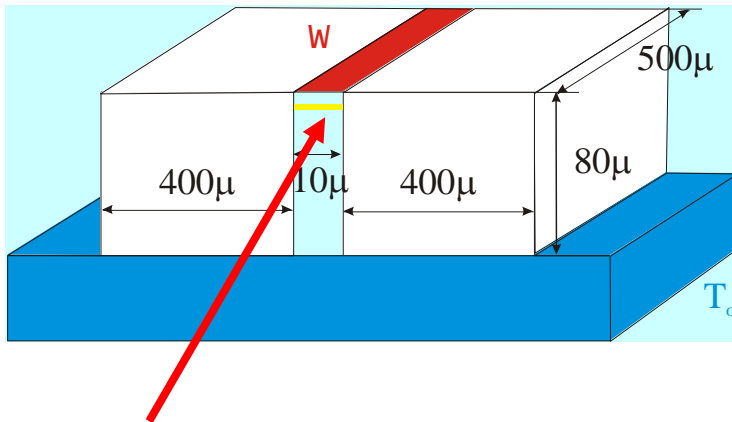
Blue laser temperature increase – heat conductivity

- Outside heat generation zone

$$C_p \frac{\partial [T(\vec{r}, t)\rho(\vec{r}, t)]}{\partial t} + \kappa \Delta T(\vec{r}, t) = 0$$

- Heat generation zone

$$C_p \frac{\partial [T(\vec{r}, t)\rho(\vec{r}, t)]}{\partial t} + \kappa \Delta T(\vec{r}, t) = j\rho$$



p-n junction

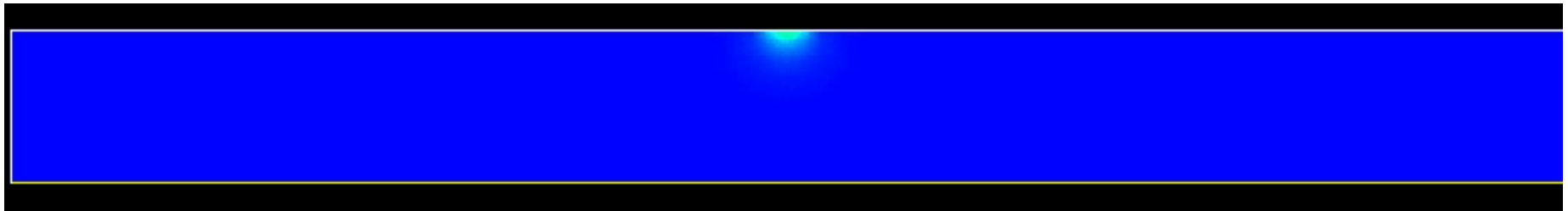
- Boundary conditions: diamond chip

$$T = T_0$$

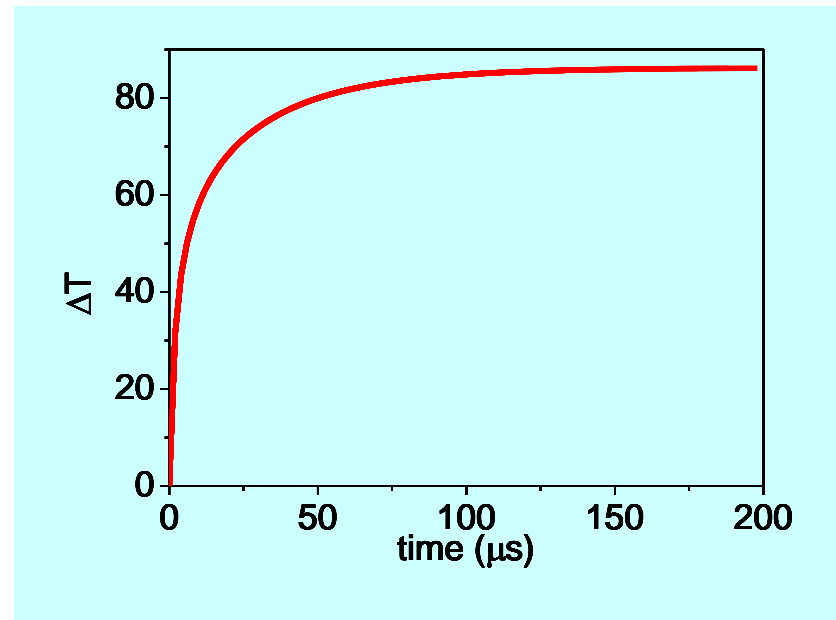
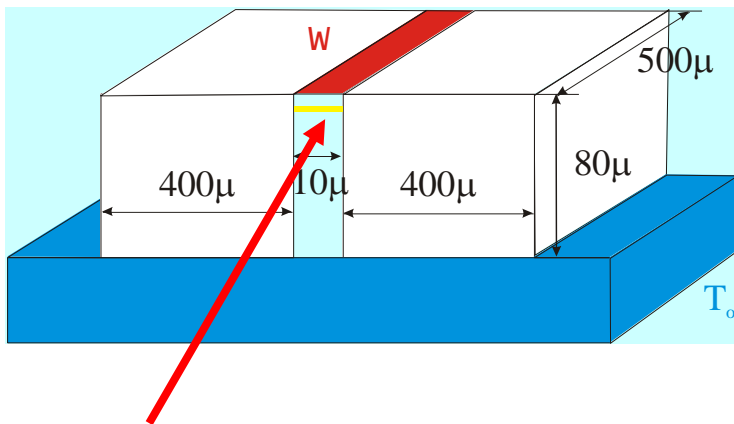
- Boundary conditions: other

$$\vec{J}_q = -\kappa \nabla T = 0$$

Time evolution in IHPP laser diode



Power $W = 12\text{ W}$



p-n junction

Natural convection in liquid Ga – GaN growth

- **Equation of motion**

$$\text{div}(\vec{v}(\vec{r}, t)) = 0$$

$$\rho_o \left[\frac{\partial \vec{v}(\vec{r}, t)}{\partial t} + (\vec{v}(\vec{r}, t) \cdot \nabla) \vec{v}(\vec{r}, t) \right] = \mu \Delta \vec{v}(\vec{r}, t) + \rho_o [\beta_T (T - T_o) + \beta_c c] \vec{f}(\vec{r}, t)$$

$$\rho_o C_p(T) \left[\frac{\partial [T(\vec{r}, t)]}{\partial t} + (\vec{v}(\vec{r}, t) \cdot \nabla) T(\vec{r}, t) \right] = \text{div}(\kappa \nabla T(\vec{r}, t)) + r_\varepsilon$$

$$\vec{f}(\vec{r}, t) = \vec{g}$$

- **5 variables: 3 velocity components, pressure, temperature**
- **5 equations**
- **Additional equation:**

$$\rho = \rho_o$$

Natural convection – GaN crystal growth



- **Solid surfaces – no-slip condition velocity:**

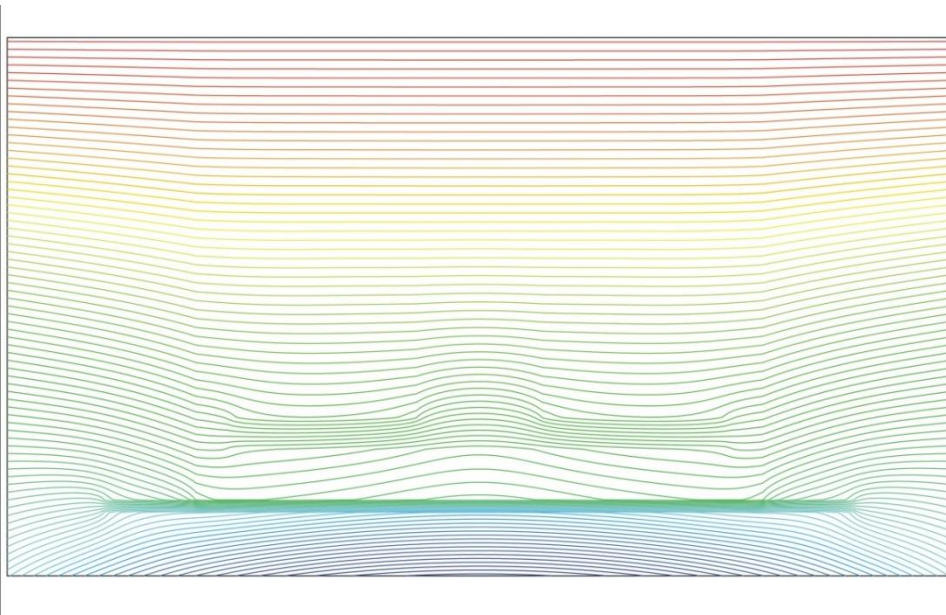
$$\vec{v}(\vec{r}, t) = 0$$

- **Solid surfaces – temperature – from measurements:**

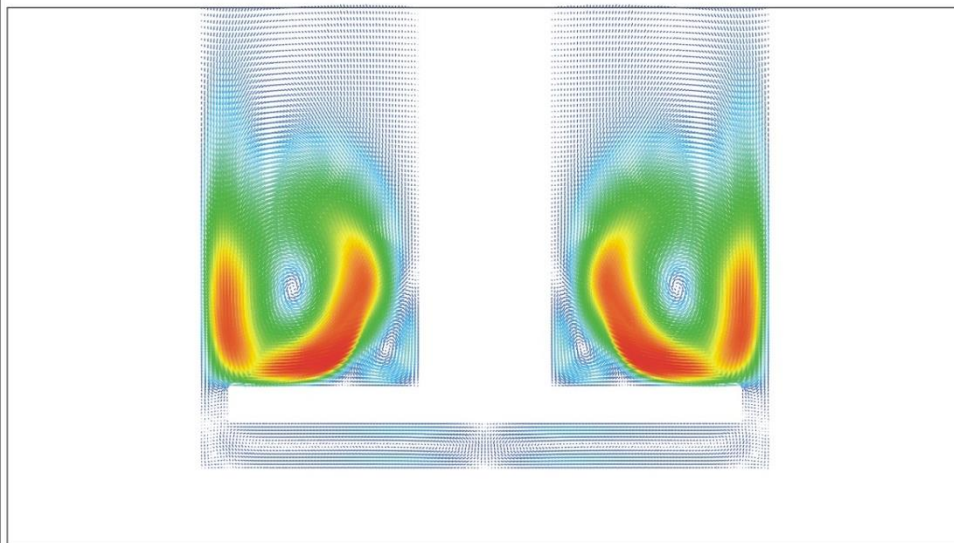
$$T(\vec{r}, t) = T(\vec{r})$$

Paweł Strąk (Fidap)

Natural convection – GaN crystal growth

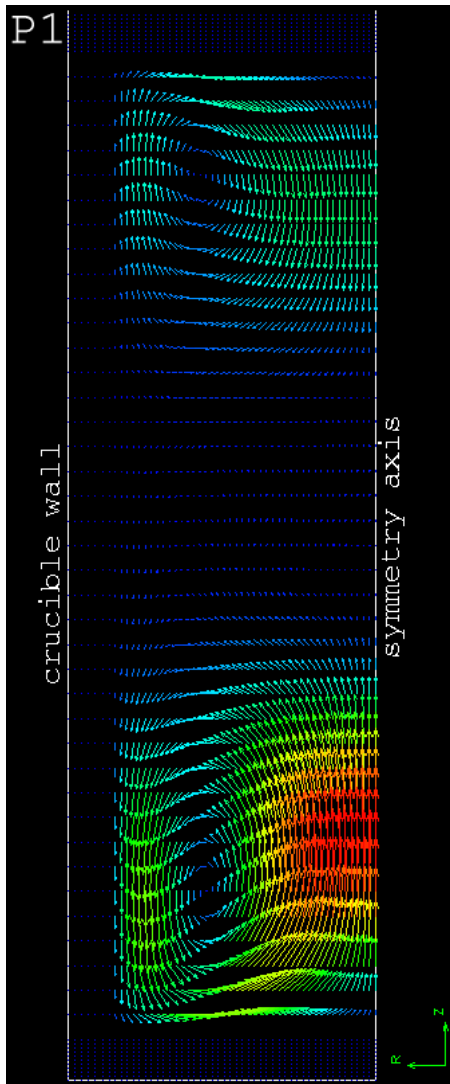


Temperature

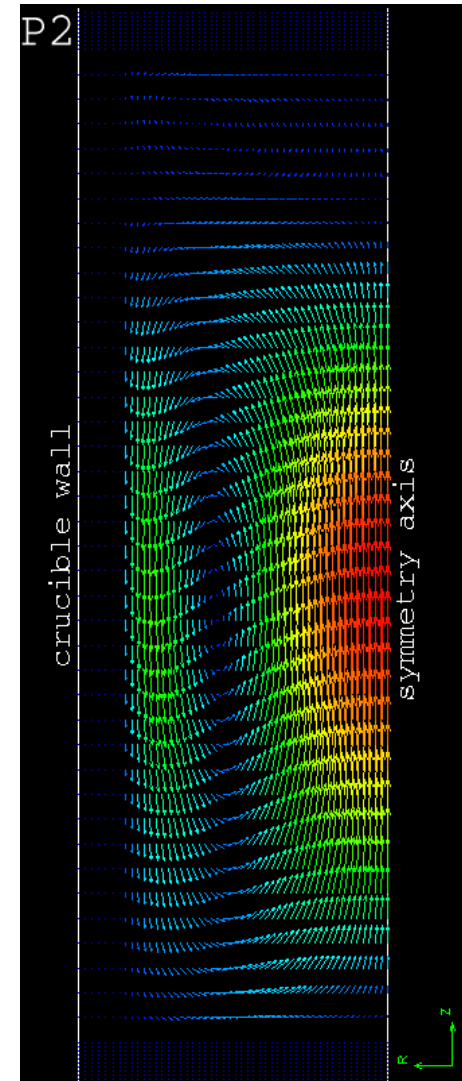
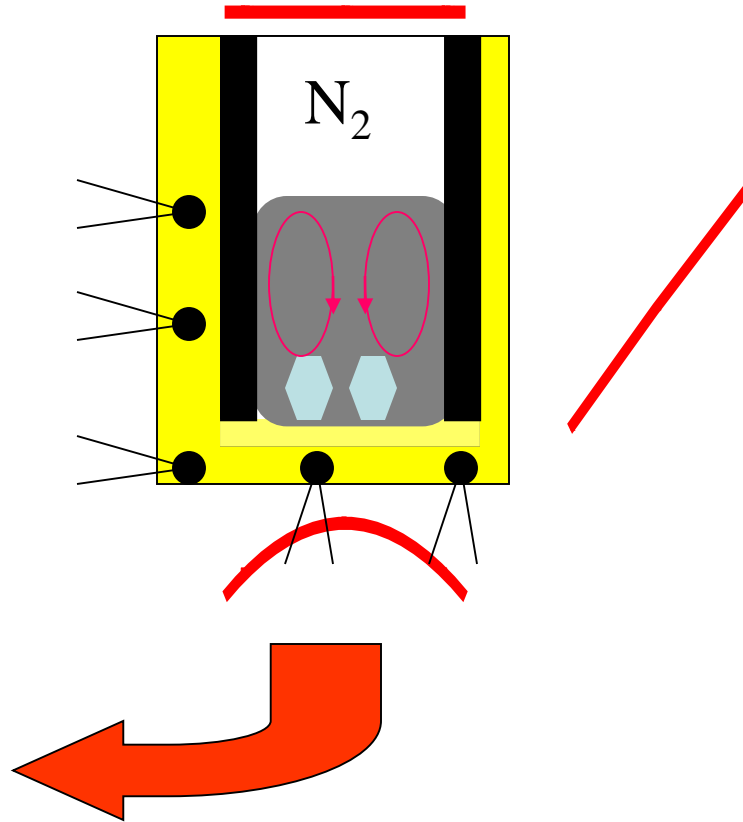


Velocity

Convection in gallium (Fidap)

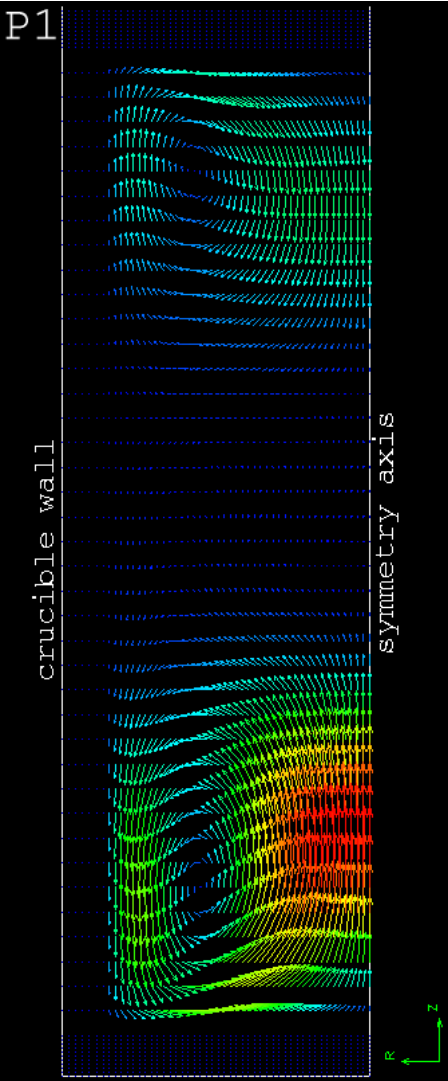


$$v_{\max} = 0.96 \text{ mm/s}$$

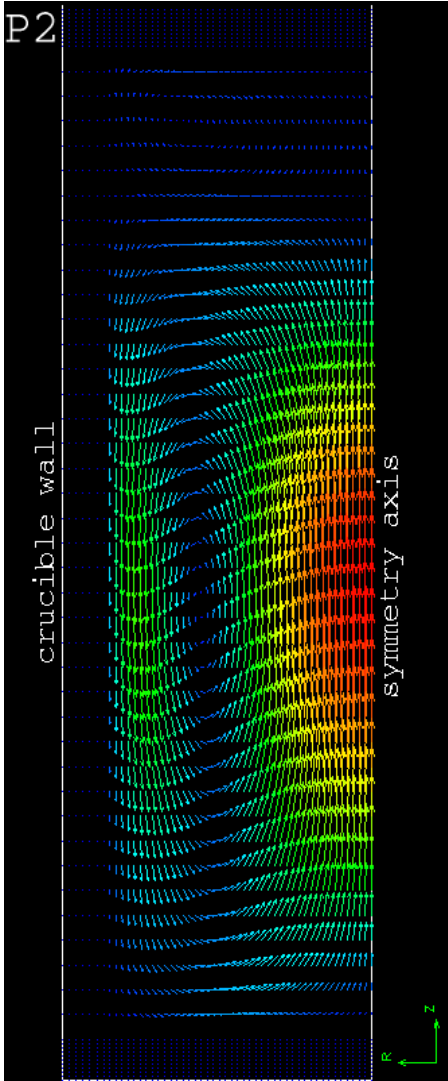
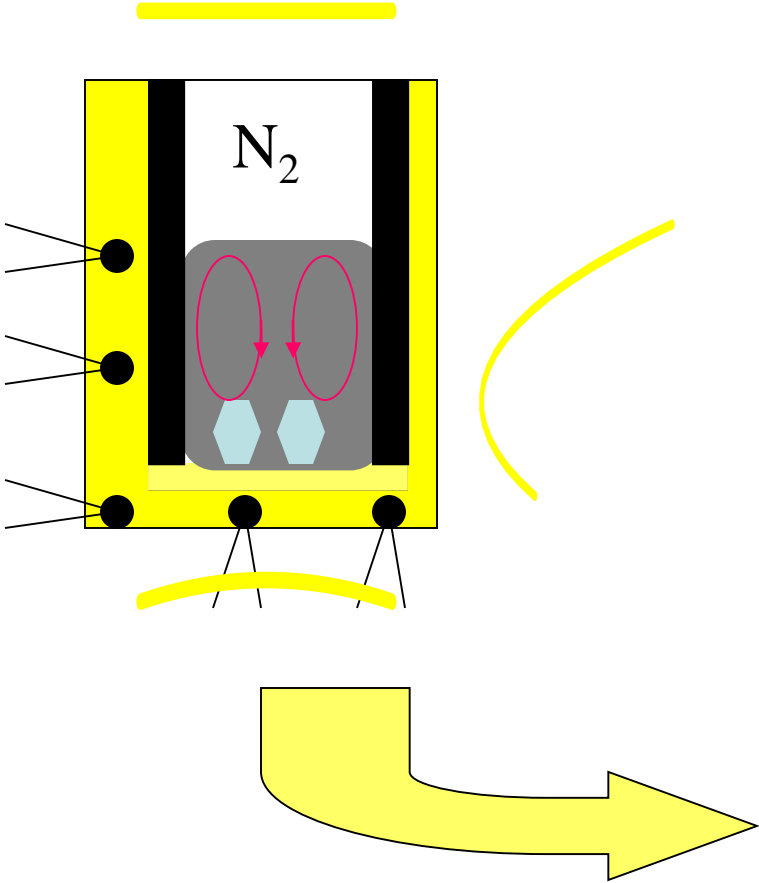


$$v_{\max} = 3.57 \text{ mm/s}$$

Convection in gallium (Fidap)



$v_{\max} = 0.96 \text{ mm/s}$



$v_{\max} = 3.57 \text{ mm/s}$

Elasticity

- **Force equilibrium: f_i – bulk force**

$$\frac{\partial \sigma_{ij}}{\partial r_j} + f_i = 0$$

- **Hooke law: stress tensor σ_{ij} strain tensor u_{ij} :**

$$\sigma_{ij} = s_{ijkl} u_{kl}$$

- **Strain tensor and vector**

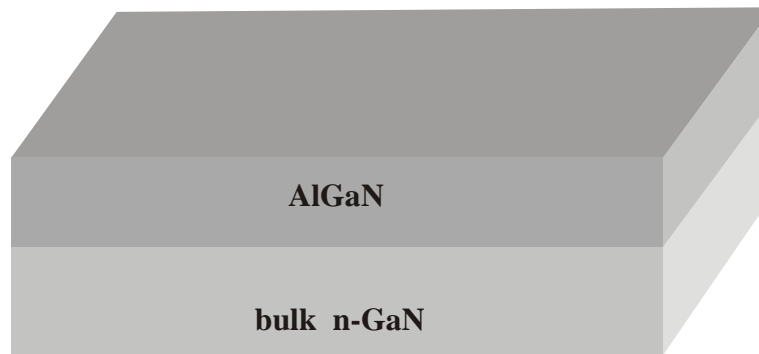
$$u_{kl} = \frac{1}{2} \left(\frac{\partial u_k}{\partial r_l} + \frac{\partial u_l}{\partial r_k} \right)$$

- **Linear elasticity equations**

$$s_{ijkl} \left(\frac{\partial^2 u_k}{\partial r_j \partial r_l} + \frac{\partial^2 u_l}{\partial r_j \partial r_k} \right) + 2f_i = 0$$

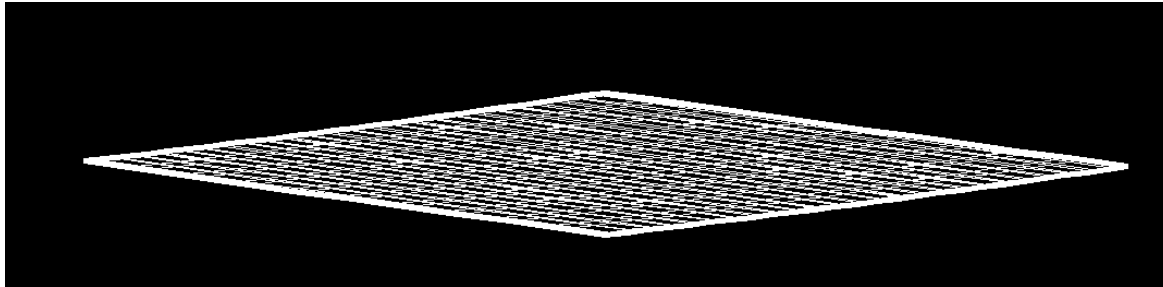
Strain in GaN laser - Abaqus

- **2-layer structure**
- **Fully strained**
- **Wegard law for lattice constants**
- **Elastic anisotropic approach**



<http://www.simulia.com/>

Bowing in GaN/AlGaN structure - FEM

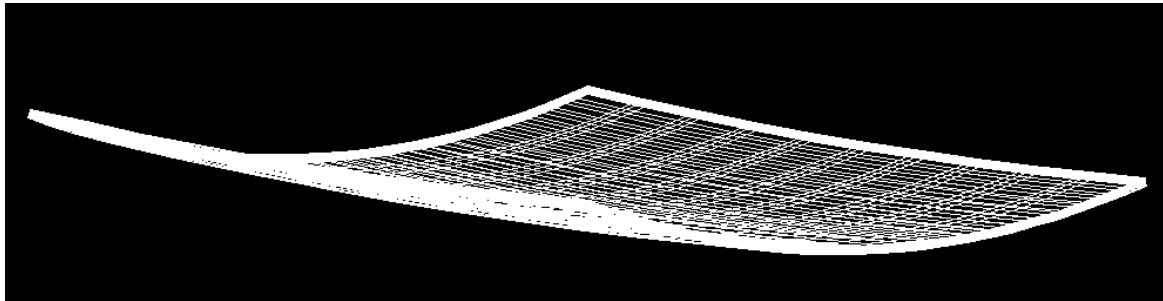


$h = 1 \mu\text{m}$

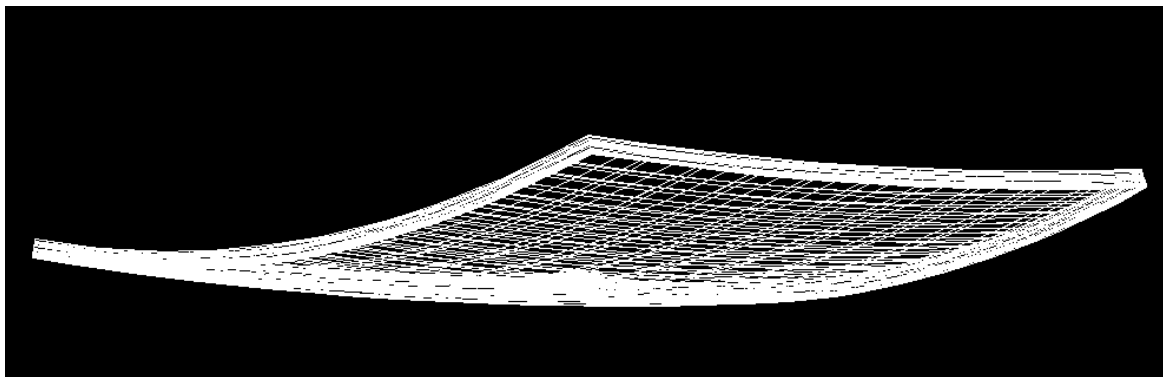
Substrate:

Thickness $H = 60 \mu\text{m}$

Size – 1cm x 1cm



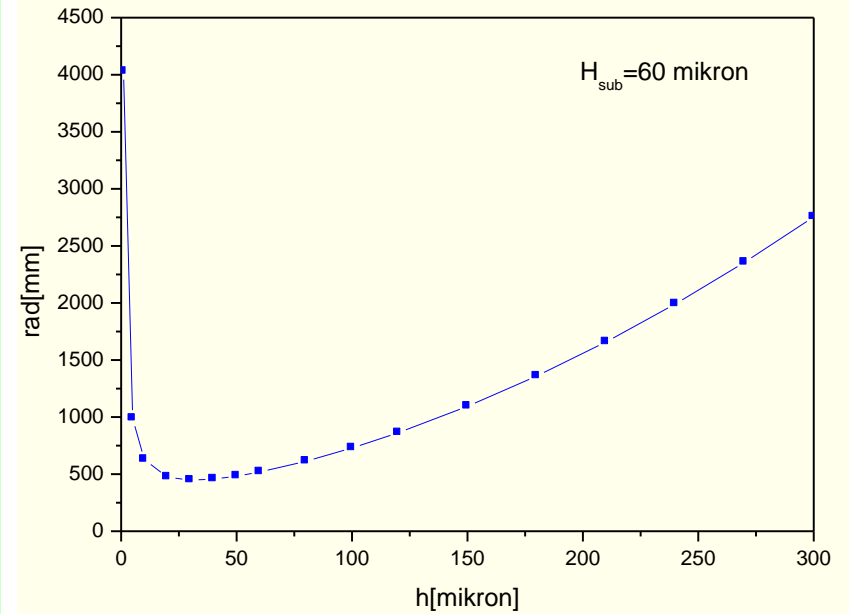
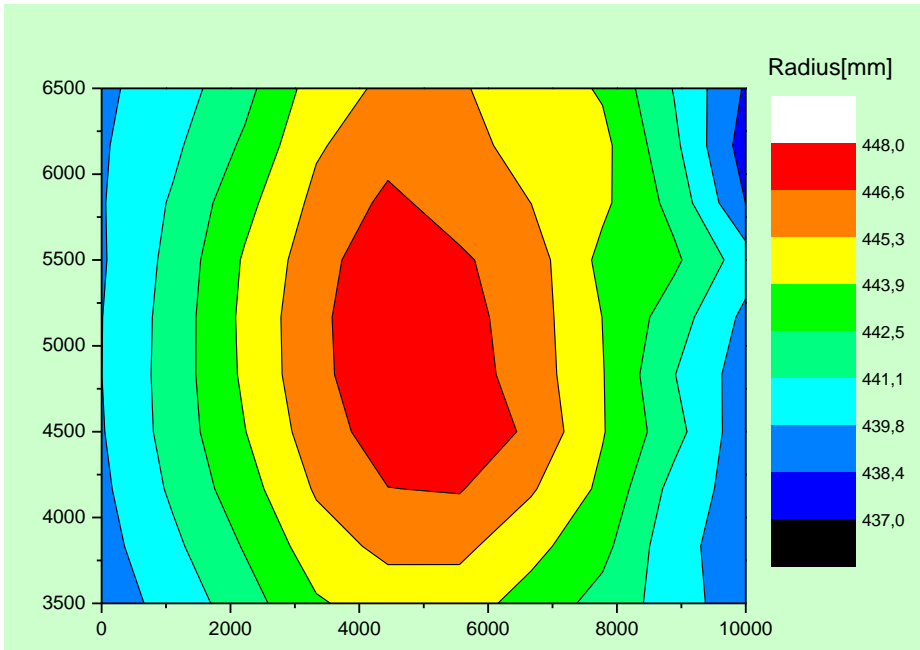
$h = 50 \mu\text{m}$



$h = 180 \mu\text{m}$

Magnification factor - 100

Curvature radii - surface



Clyne formula

$$\kappa = \frac{1}{R} = \frac{6 f H}{(h + H)^2}$$

Stoney formula (dla $h/H < 0.05$)

$$\kappa = \frac{1}{R} = \frac{6 f}{H}$$

$E_{\text{layer}} = E_{\text{substrate}}$

f – misfit

h – layer thickness

H – substrate thickness

Good agreement with Stoney –Clyne theory

Acknowledgements for ICM UW

- **Access to commercial software**
 - Fidap (ANSYS Inc.)**
 - Abaqus (Dassault Systèmes)**
- **Computing facilities within G15-9 project**