Max-Planck-Institut für Eisenforschung, Düsseldorf

FEM: A Basic Overview of the Method & Outlook on Applications

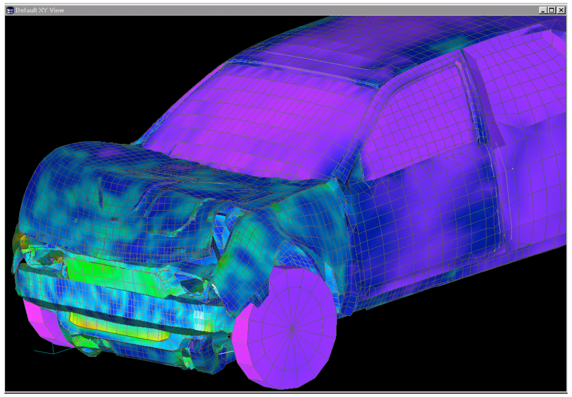
William Counts

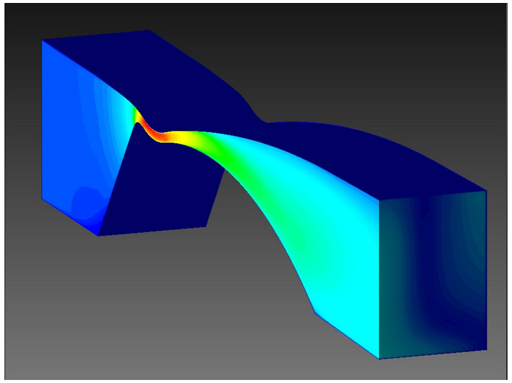
, August 25, 2008, Aachen

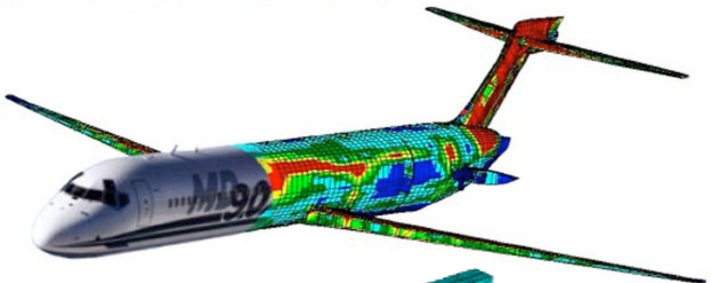
Introduction

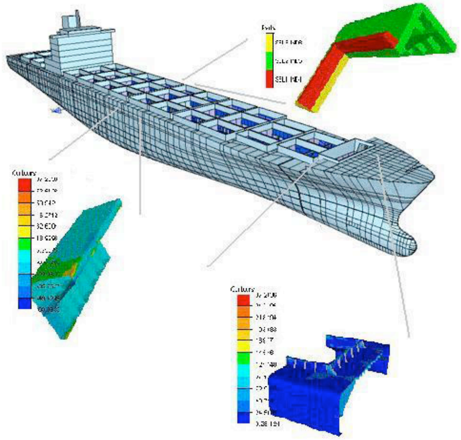
FEM: Finite Element Method

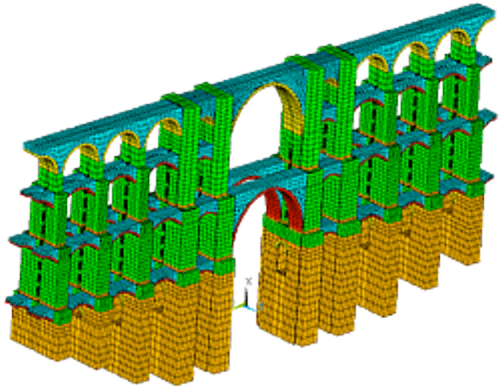
A great picture generating tool!!!!

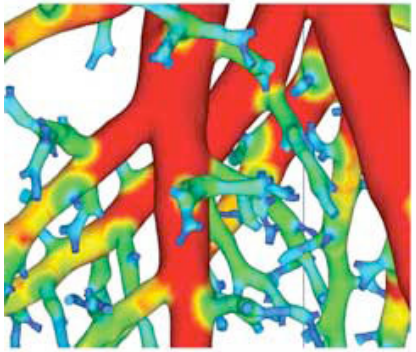










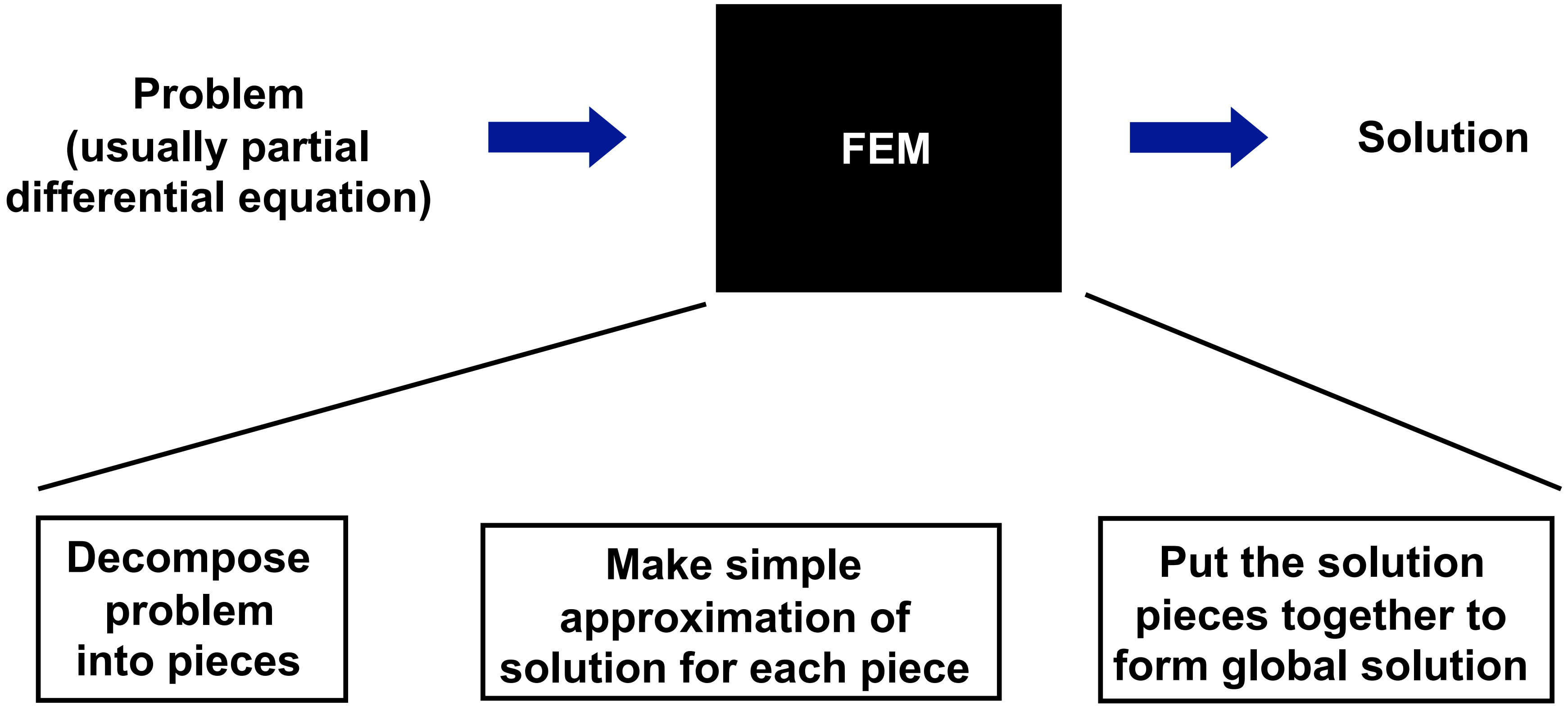


MORE THAN JUST COMPUTER GENERATED ART

Introduction

My view: FEM is a mathematical tool.

Any physically meaningful output MUST result from physical input provided by you to FEM



Problem (usually partial differential equation)

FEM

Solution

Decompose problem into pieces

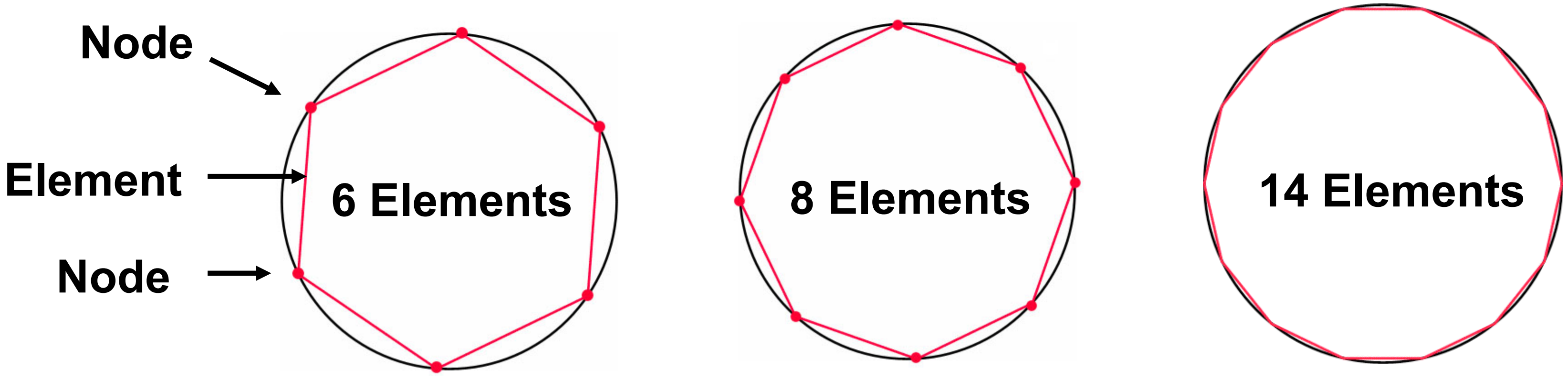
Make simple approximation of solution for each piece

Put the solution pieces together to form global solution

Introduction

Example: Computing the circumference of a circle with radius r

- Decompose (or Discretize) the problem



- Approximate solution for each piece (or element)

$$L_e = 2r\sin\frac{\theta}{2}$$

- Assemble the Element Equations and Solve

$$P_c = \sum_{N_e} L_e = \sum_{N_e} 2r\sin\frac{\theta}{2}$$

$N_e = 6$ $P_c = 6.00$ r
$N_e = 8$ $P_c = 6.12$ r
$N_e = 14$ $P_c = 6.23$ r

⟹ **2πr**

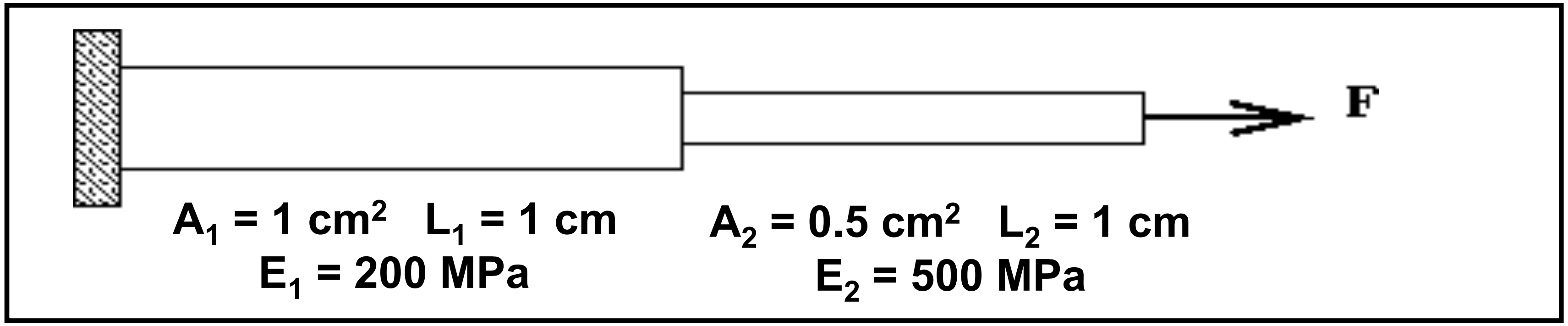
FEM

- FEM can be applied to many different problems

Problem Type	DOF #1	DOF #2
Structures and solid mechanics	Displacement	Mechanical force
Heat conduction	Temperature	Heat flux
Acoustic fluid	Displacement potential	Particle velocity
Potential flows	Pressure	Particle velocity
General flows	Velocity	Fluxes
Electrostatics	Electric potential	Charge density
Magnetistatics	Magnetic potential	Magnetic intensity

- Going to use “Structures and solid mechanics” examples
- Easily apply other problem types by substituting appropriate variables and equations

FEM – 1D Rod Example


F
A_1 = 1 cm^2 L_1 = 1 cm
E_1 = 200 MPa
A_2 = 0.5 cm^2 L_2 = 1 cm
E_2 = 500 MPa

FEM

1) Enforce Equilibrium ➡ $\sum F = 0$

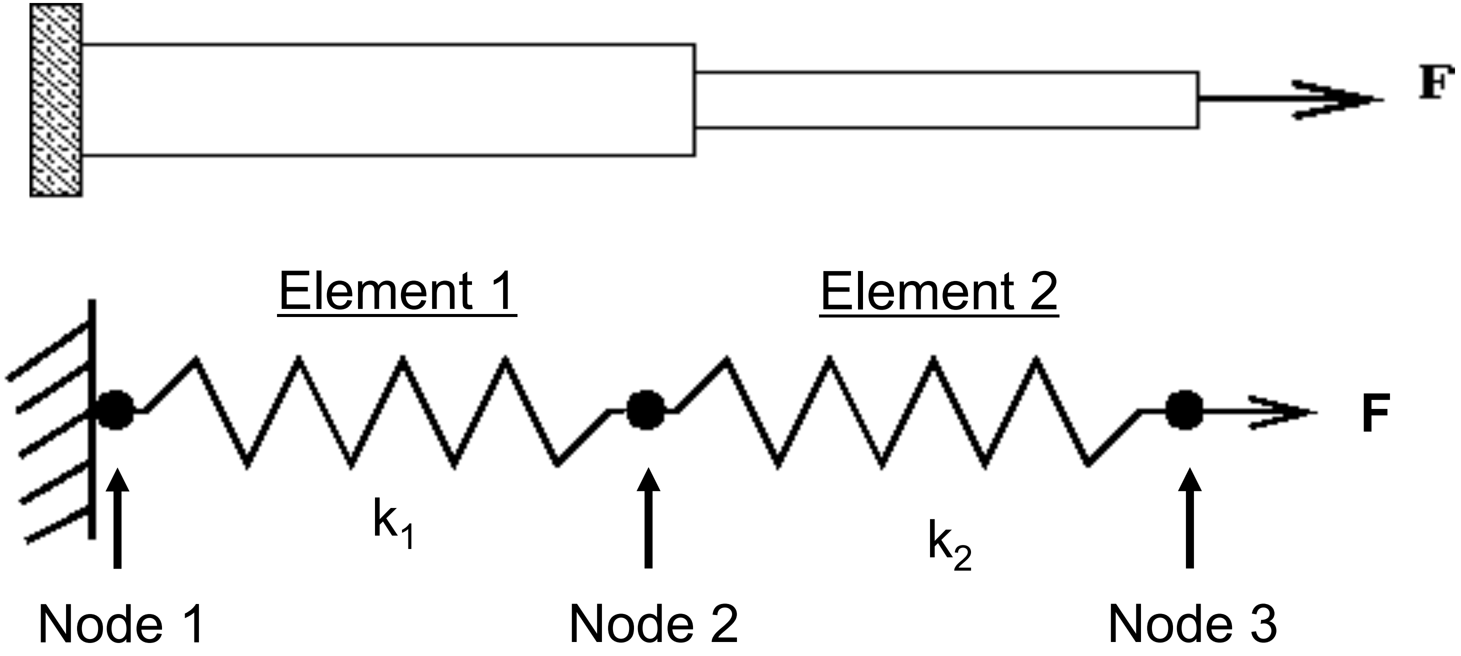
2) Ensure compatibility (no formation of holes/voids)

Modeler

Provide constitutive equation for each rod
(i.e. what is the relationship between force and displacement)

Model rods as elastic springs ➡ F = kd $k = \frac{AE}{L}$

FEM – 1D Rod Example



For each element, write a force balance at each node

Element 1

$$f_{E1}^{Node-1} + k_{E1}\left(u^{Node-1} - u^{Node-2}\right) = 0$$

$$f_{E1}^{Node-2} + k_{E1}\left(u^{Node-2} - u^{Node-1}\right) = 0$$

Element 2

$$f_{E2}^{Node-2} + k_{E2}\left(u^{Node-2} - u^{Node-3}\right) = 0$$

$$f_{E2}^{Node-3} + k_{E2}\left(u^{Node-3} - u^{Node-2}\right) = 0$$

FEM – 1D Rod Example

- **Assemble the force balance equations for all 3 nodes**

$$\begin{bmatrix} k_{E1} & -k_{E1} & 0 \\ k_{E1} & k_{E1}+k_{E2} & -k_{E2} \\ 0 & -k_{E2} & k_{E2} \end{bmatrix} \begin{bmatrix} u^{Node-1} \\ u^{Node-2} \\ u^{Node-3} \end{bmatrix} = \begin{bmatrix} -f_{E1}^{Node\text{-}1} \\ -f_{E1}^{Node\text{-}2} - f_{E2}^{Node\text{-}2} \\ -f_{E2}^{Node\text{-}3} \end{bmatrix}$$

$$[K][u] = [f]$$

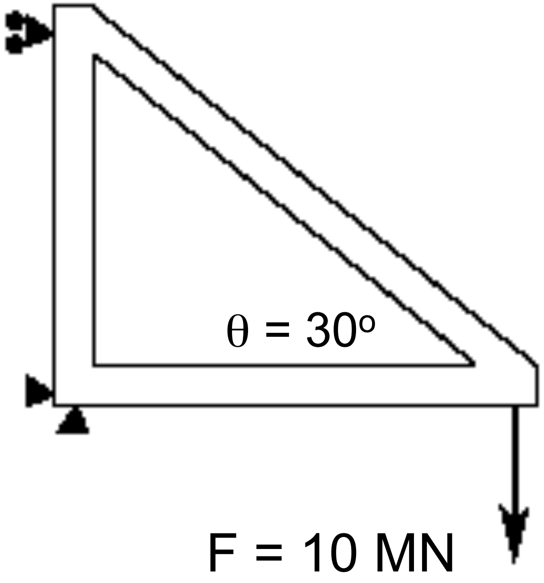
K = Stiffness Matrix u = displacement vector f = force vector

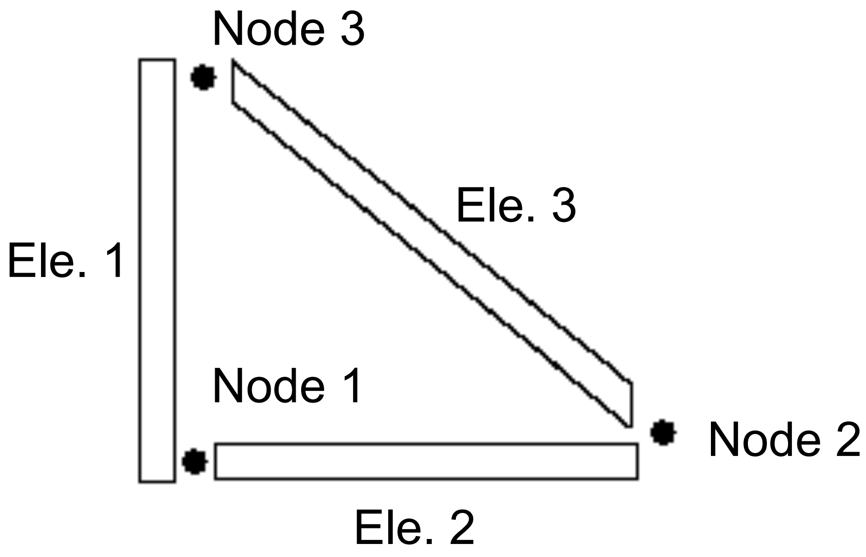
- **Apply Boundary Conditions**

$$\begin{bmatrix} k_{E1} & -k_{E1} & 0 \\ k_{E1} & k_{E1}+k_{E2} & -k_{E2} \\ 0 & -k_{E2} & k_{E2} \end{bmatrix} \begin{bmatrix} 0 \\ u^{Node-2} \\ u^{Node-3} \end{bmatrix} = \begin{bmatrix} -f_{E1}^{Node-1} \\ 0 \\ F \end{bmatrix}$$

3 equations and 3 unknowns **Easily solve with a matrix inversion**

FEM – 2D Truss Example





- **FEM** #1) Equilibrium $\sum F_x = 0$ $\sum F_y = 0$ #2) Compatibility

- **Modeler** Elastic spring model to define the F vs. d relationship

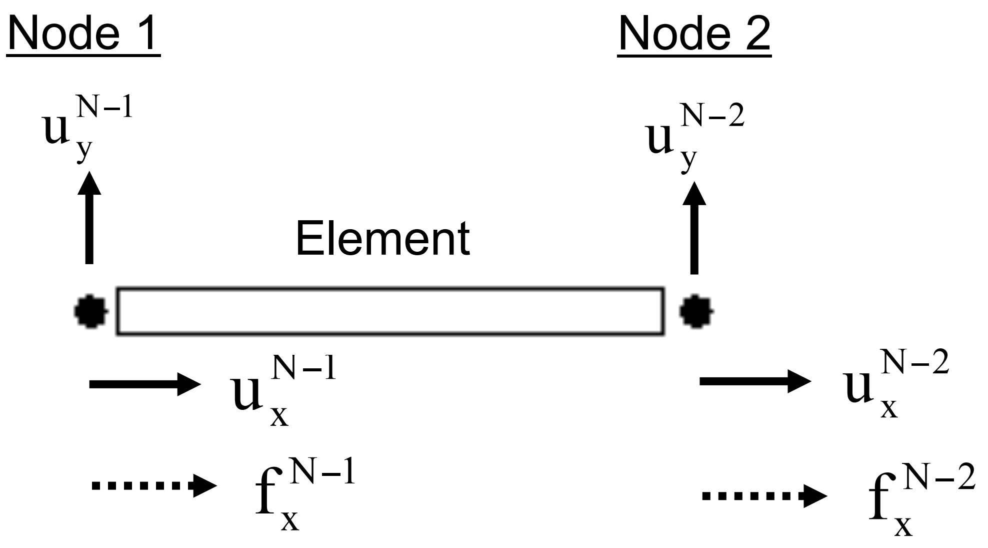
  - 2D ➡ Account for orientation
  - Rotate u and f

$$[k][T][u] = [T][f]$$

$$[k'][u] = [f] \qquad [k'] = [T]^T[k][T]$$

$$[T] = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

FEM – 2D Truss Example



- Element can be displaced in 2 directions
- Element supports along axial direction

Element force balance equations

$$f_x^{N-1} + k_E\left(u_x^{N-1} - u_x^{N-2}\right) = 0$$

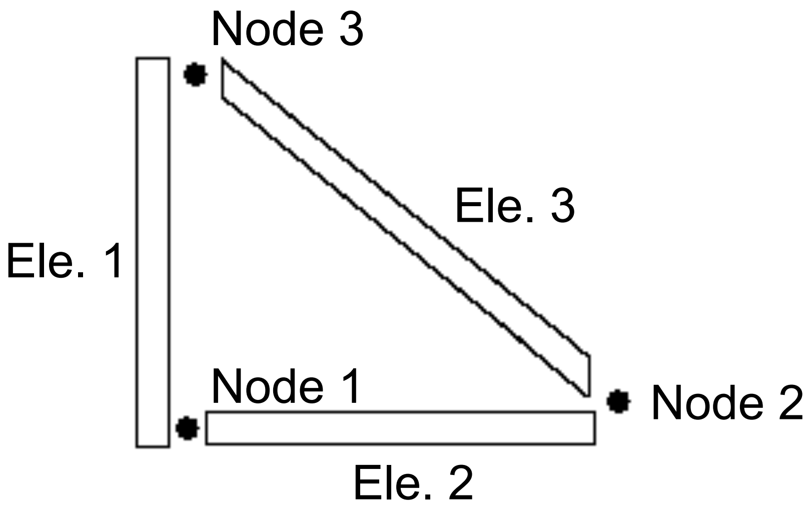
$$f_y^{N-1} + k_E\left(u_y^{N-1} - u_y^{N-2}\right) = 0$$

$$f_x^{N-2} + k_E\left(u_x^{N-2} - u_x^{N-1}\right) = 0$$

$$f_y^{N-2} + k_E\left(u_y^{N-2} - u_y^{N-1}\right) = 0$$

$$\Rightarrow \quad [T]^T \begin{bmatrix} k_E & 0 & -k_E & 0 \\ 0 & 0 & 0 & 0 \\ -k_E & 0 & k_E & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} [T] \begin{bmatrix} u_x^{N-1} \\ u_y^{N-1} \\ u_x^{N-2} \\ u_y^{N-2} \end{bmatrix} = \begin{bmatrix} f_x^{N-1} \\ 0 \\ f_x^{N-2} \\ 0 \end{bmatrix}$$

FEM – 2D Truss Example



	$A(m^2)$	E(MPa)	L(m)
Ele 1	0.01	10	5
Ele 2	0.01	10	8.66
Ele 3	0.01	10	10

$$k = \frac{AE}{L}$$

$k_1 = 2000$ $k_2 = 1154.73$ $k_3 = 1000$

$$[T_1] = \begin{bmatrix} c90 & s90 & 0 & 0 \\ -s90 & c90 & 0 & 0 \\ 0 & 0 & c90 & s90 \\ 0 & 0 & -s90 & c90 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$[T_2] = \begin{bmatrix} c0 & s0 & 0 & 0 \\ -s0 & c0 & 0 & 0 \\ 0 & 0 & c0 & s0 \\ 0 & 0 & -s0 & c0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[T_3] = \begin{bmatrix} c150 & s150 & 0 & 0 \\ -s150 & c150 & 0 & 0 \\ 0 & 0 & c150 & s150 \\ 0 & 0 & -s150 & c150 \end{bmatrix} = \begin{bmatrix} -0.866 & -0.5 & 0 & 0 \\ 0.5 & -0.866 & 0 & 0 \\ 0 & 0 & -0.866 & -0.5 \\ 0 & 0 & 0.5 & -0.866 \end{bmatrix}$$

FEM – 2D Truss Example

- Calculate the element stiffness matrixes ➡ $[T]^T \begin{bmatrix} k_E & 0 & -k_E & 0 \\ 0 & 0 & 0 & 0 \\ -k_E & 0 & k_E & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} [T]$

For example Element 3:

$$\begin{bmatrix} -0.866 & -0.5 & 0 & 0 \\ 0.5 & -0.866 & 0 & 0 \\ 0 & 0 & -0.866 & -0.5 \\ 0 & 0 & 0.5 & -0.866 \end{bmatrix} \begin{bmatrix} 1000 & 0 & -1000 & 0 \\ 0 & 0 & 0 & 0 \\ -1000 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.866 & -0.5 & 0 & 0 \\ 0.5 & -0.866 & 0 & 0 \\ 0 & 0 & -0.866 & -0.5 \\ 0 & 0 & 0.5 & -0.866 \end{bmatrix}$$

$$\boxed{[k'_{E3}] = \begin{bmatrix} 750 & -433.2 & -750 & 433.2 \\ -433.2 & 250 & 433.2 & -250 \\ -750 & 433.2 & 750 & -433.2 \\ 433.2 & -250 & -433.2 & 250 \end{bmatrix}}$$

$$\boxed{[k'_{E1}] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2000 & 0 & -2000 \\ 0 & 0 & 0 & 0 \\ 0 & -2000 & 0 & 2000 \end{bmatrix}}$$

$$\boxed{[k'_{E2}] = \begin{bmatrix} 1154.73 & 0 & -1154.73 & 0 \\ 0 & 0 & 0 & 0 \\ -1154.73 & 0 & 1154.73 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}$$

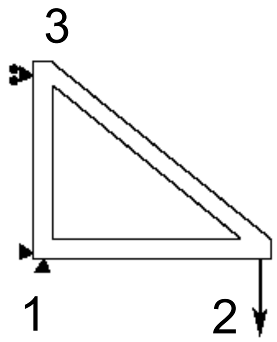
FEM – 2D Truss Example

- Assemble the element stiffness matrixes

$$\begin{bmatrix} 1154.73+750 & -433.2 & -750 & 433.2 & -1154.73 & 0 \\ -433.2 & 250 & 433.2 & -250 & 0 & 0 \\ -750 & 433.2 & 750 & -433.2 & 0 & 0 \\ 433.2 & -250 & -433.2 & 250+2000 & 0 & -2000 \\ -1154.73 & 0 & 0 & 0 & 1154.73 & 0 \\ 0 & 0 & 0 & -2000 & 0 & 2000 \end{bmatrix} \begin{bmatrix} u_x^{N-2} \\ u_y^{N-2} \\ u_x^{N-3} \\ u_y^{N-3} \\ u_x^{N-1} \\ u_y^{N-1} \end{bmatrix} = \begin{bmatrix} f_x^{N-2} \\ f_y^{N-2} \\ f_x^{N-3} \\ f_y^{N-3} \\ f_x^{N-1} \\ f_y^{N-1} \end{bmatrix}$$

- Apply boundary conditions

$$\begin{bmatrix} 1154.73+750 & -433.2 & -750 & 433.2 & -1154.73 & 0 \\ -433.2 & 250 & 433.2 & -250 & 0 & 0 \\ -750 & 433.2 & 750 & -433.2 & 0 & 0 \\ 433.2 & -250 & -433.2 & 250+2000 & 0 & -2000 \\ -1154.73 & 0 & 0 & 0 & 1154.73 & 0 \\ 0 & 0 & 0 & -2000 & 0 & 2000 \end{bmatrix} \begin{bmatrix} u_x^{N-2} \\ u_y^{N-2} \\ 0 \\ u_y^{N-3} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -10 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

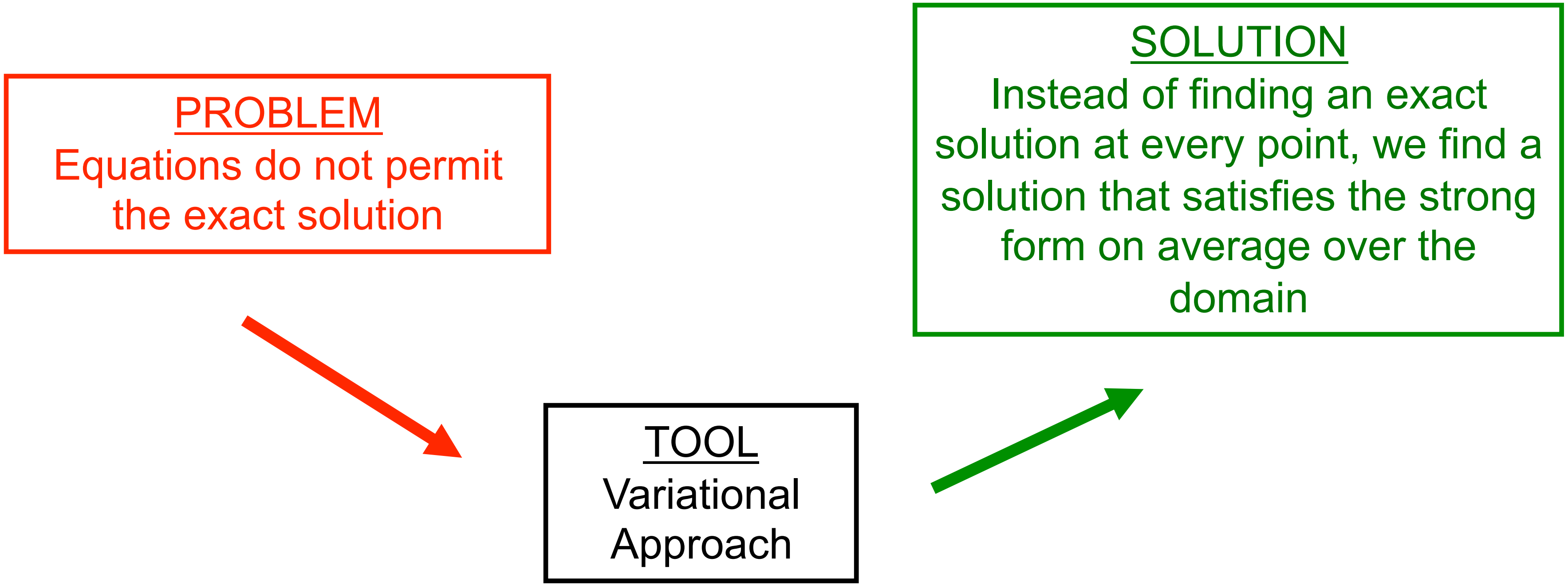


- Solve 3 equations with 3 unknowns

$$u_x^{N-2} = -0.015 \qquad u_y^{N-2} = -0.071 \qquad u_y^{N-3} = -0.005$$

FEM: Strong Form vs. Weak Form

- Governing equations are usually complex differential equations
- Often these equations cannot be solved over an element



PROBLEM

Equations do not permit the exact solution

TOOL

Variational Approach

SOLUTION

Instead of finding an exact solution at every point, we find a solution that satisfies the strong form on average over the domain

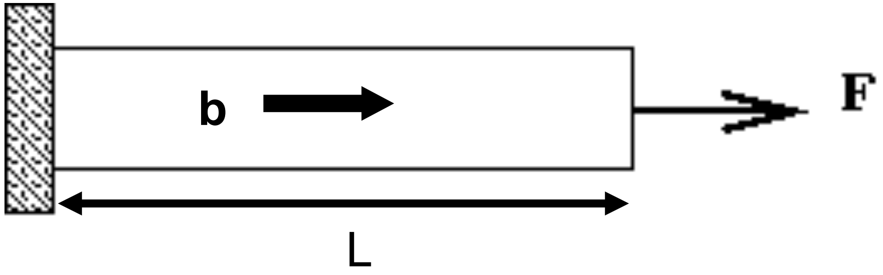
FEM: Strong Form vs. Weak Form

Weak Form is based on Potential Energy (Π)

$$\Pi = \text{Strain Energy} - \text{Energy fro m applied loads}$$

$$\Pi = U - W$$

Example System:



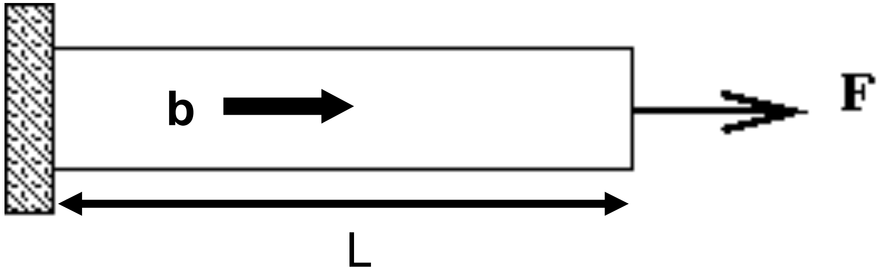
Assume Linear Elastic material: $dU = \frac{1}{2}\sigma\varepsilon = \frac{1}{2}E\left(\frac{du}{dx}\right)^2$ (eng. density)

$$\boxed{U = \int_0^L \frac{1}{2}E\left(\frac{du}{dx}\right)^2 dV = \int_0^L \frac{1}{2}E\left(\frac{du}{dx}\right)^2 A\,dx}$$

$$\boxed{W = \int_0^L bu\,dx + Fu(x = L)}$$

$$\boxed{\Pi(u) = \frac{1}{2}\int_0^L EA\left(\frac{du}{dx}\right)^2 dx - \int_0^L bu\,dx - Fu(x = L)}$$

FEM: Strong Form vs. Weak Form



Strong Form

- Governing PDE's with boundary conditions

PDE: $$EA\frac{d^2u}{dx^2}+b=F$$

BC: $$u(0)=0$$

$$EA\left.\frac{du}{dx}\right|_{x=L}=F$$

FEM →

Weak Form

- *Variational statement* of the problem

$$\Pi(u)=\frac{1}{2}\int_0^L EA\left(\frac{du}{dx}\right)^2 dx$$

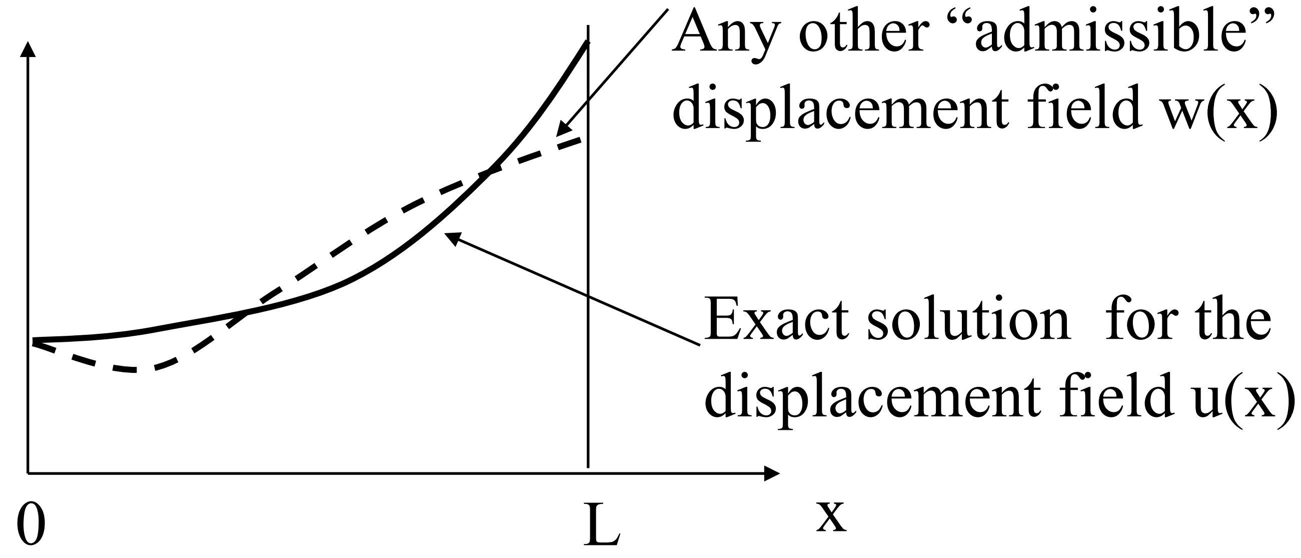
$$-\int_0^L bu\,dx-Fu(x=L)$$

FEM: Strong Form vs. Weak Form

Exact solution, u(x) is unknown

FEM “guesses” an admissible displacement field, w(x)

Admissible:
1) 1st derivative must be real
2) w(0) = 0
(Force BC automatically satisfied)


Any other “admissible” displacement field w(x)
Exact solution for the displacement field u(x)
0
L
x

FEM works w(x) ➡ $\Pi(w) = \frac{1}{2}\int_0^L EA\left(\frac{dw}{dx}\right)^2 dx - \int_0^L bw\,dx - Fw(x = L)$

FEM: Strong Form vs. Weak Form

How are w(x) and u(x) related?

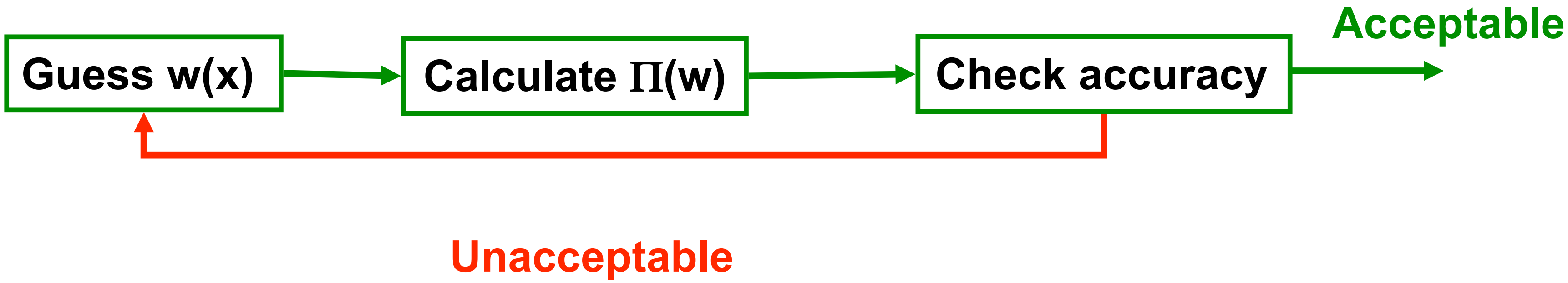
Principle of Minimum Potential Energy

**Among all admissible displacements (w(x)'s),
the one that MINIMIZES the total potential energy (Π) is u(x)**

$$\boxed{\Pi(u) < \Pi(w)}$$

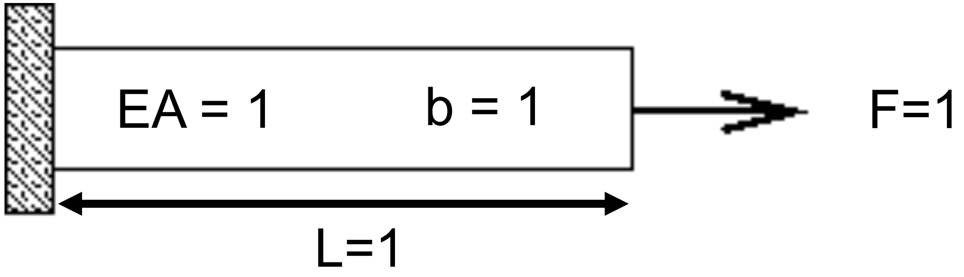
FEM

Given the Principle of Minimum Potential Energy → **Iterative scheme to find a w(x) that closely approximates u(x)**



FEM: Strong Form vs. Weak Form

Example:



Boundary Conditions:

$$EA\left.\frac{du}{dx}\right|_{x=L} = F \qquad u(0) = 0$$

Strong Form:

$$EA\frac{d^2u}{dx^2} + b = F \xrightarrow[\text{Solution}]{\text{Exact}} \boxed{u = 2x - \frac{x^2}{2}}$$

$$\Pi(u) = \frac{1}{2}\int_0^L EA\left(\frac{du}{dx}\right)^2 dx - \int_0^L bu\,dx - Fu(x = L) \longrightarrow \Pi(u) = -\frac{7}{6}$$

Weak Form:

$$\Pi(w) = \frac{1}{2}\int_0^L EA\left(\frac{dw}{dx}\right)^2 dx - \int_0^L bw\,dx - Fw(x = L) \xrightarrow{w(x) = x} \Pi(w) = -1$$

FEM: Strong Form vs. Weak Form

Strong Form vs. Weak Form

$$EA\frac{d^2u}{dx^2} = F$$

$$\Pi(w) = \frac{1}{2}\int_0^L EA\left(\frac{dw}{dx}\right)^2 dx - \int_0^L bw\,dx - Fw(x = L)$$

- Weak Form ➡ weaker statement of the problem
  - Based on potential energy
  - Has the effect of relaxing the problem
  - “Average” solution over the domain
  - A solution of the strong form will also satisfy the weak form, but not vice versa.
  - Principle of Minimum Potential Energy
    - w(x) that minimizes Π, equals u(x)

Galerkin's Method / Rayleigh-Ritz Principle

How does FEM determine w(x) function?

w(x) must be 1st order continuous and satisfy BC

Generally, polynomials and sine/cosine functions are simple enough to be practical.

$$w(x) = w_0 + w_1 x$$

$$w(x) = w_0 + w_1 x + w_2 x^2$$

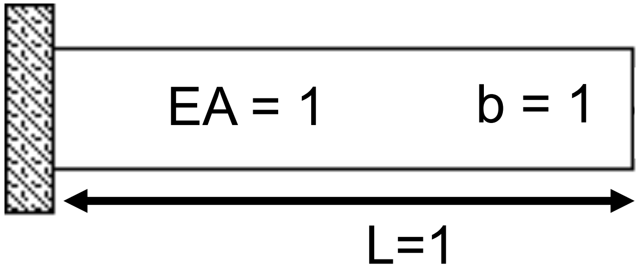
w_i's are to be determined

$$\frac{\partial \Pi}{\partial w_i} = 0$$

Numerical Methodology used to determine w_i's → **Galerkin's Method/ Rayleigh-Ritz**

Galerkin's Method / Rayleigh-Ritz Principle

Example:



$$\Pi(w) = \frac{1}{2}\int_0^L EA\left(\frac{dw}{dx}\right)^2 dx - \int_0^L bw\, dx$$

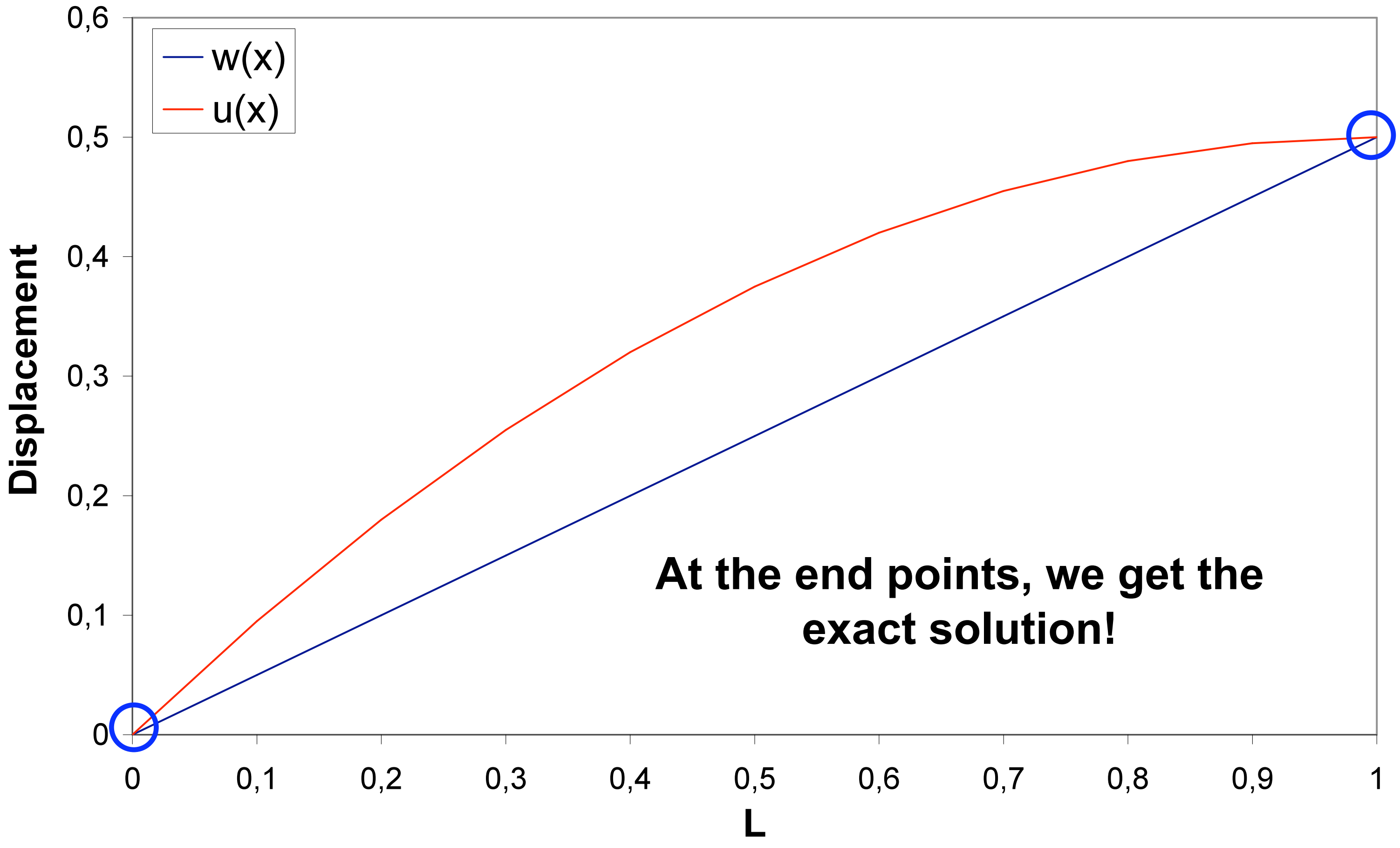
$$w(x) = w_0 + w_1 x \qquad w(0) = 0 \Rightarrow w_0 = 0$$

$$\Pi(w) = \frac{1}{2}\int_0^1 \left(\frac{d}{dx}(w_1 x)\right)^2 dx - \int_0^1 w_1 x\, dx \qquad \Pi(w) = w_1{}^2 - w_1$$

$$\frac{\partial \Pi}{\partial w_1} = 0 = 2w_1 - 1 \qquad w_1 = \frac{1}{2}$$

Galerkin’s Method / Rayleigh-Ritz Principle

$$w(x) = \frac{1}{2}x \qquad u(x) = bLx - \frac{bx^2}{2} = x - \frac{x^2}{2}$$



At the end points, we get the exact solution!

Galerkin's Method / Rayleigh-Ritz Principle

Reformulate equations in a more general form

$$w(x) = \sum_{j=1}^{N} c_j \varphi_j(x)$$

solve for c_j

$$\boxed{\sum_{j=1}^{N} c_j \int_0^L \frac{d\varphi_j}{dx}\frac{d\varphi_i}{dx}dx = \int_0^L b\varphi_i dx}$$

$$K_{ij} = \int_0^L \frac{d\varphi_j}{dx}\frac{d\varphi_i}{dx}dx$$

Stiffness Matrix

$$f_i = \int_0^L p_0 \varphi_i dx$$

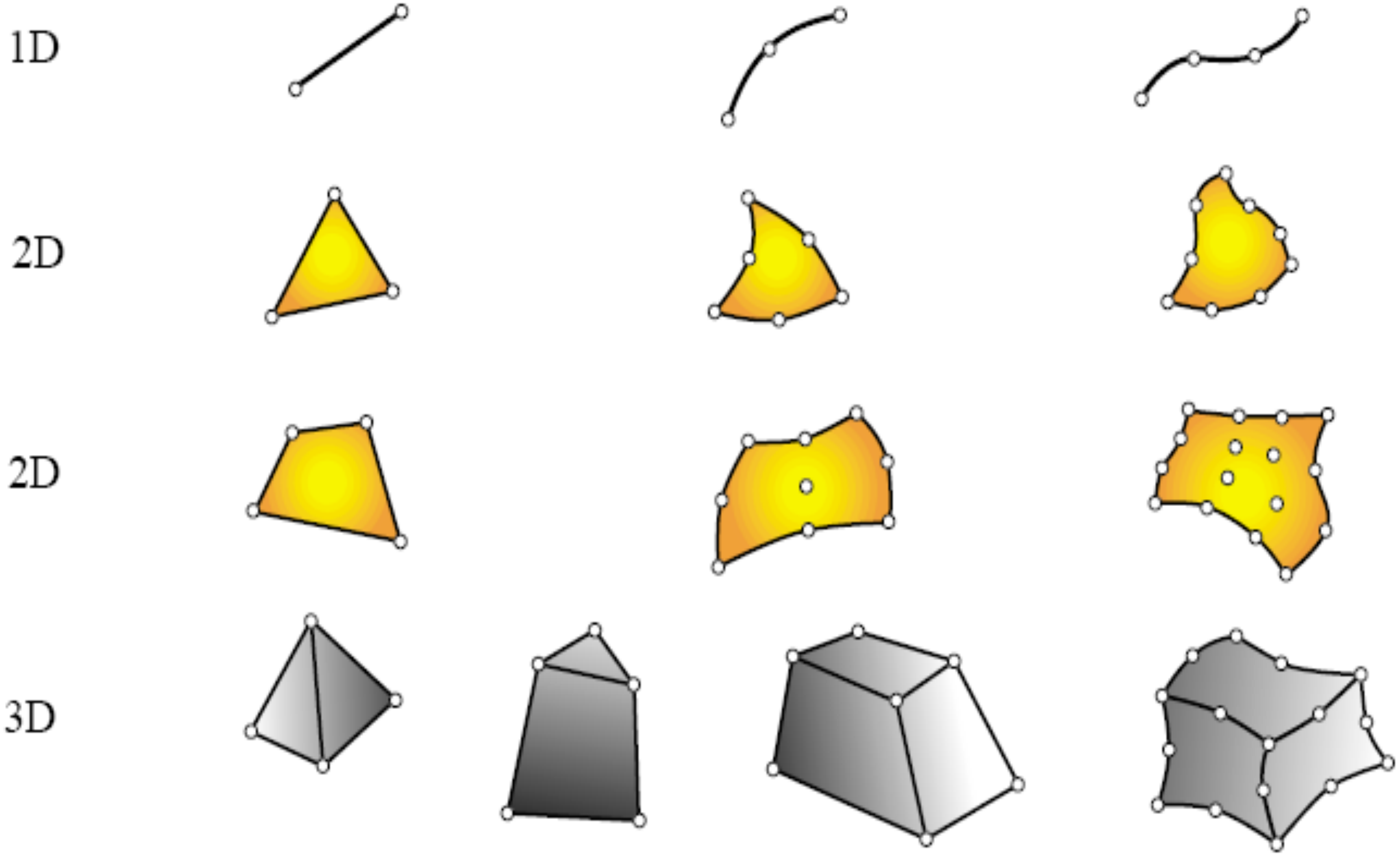
Force vector

Recover: Kc=f

FEM: Elements

- Wide range of element types and shapes

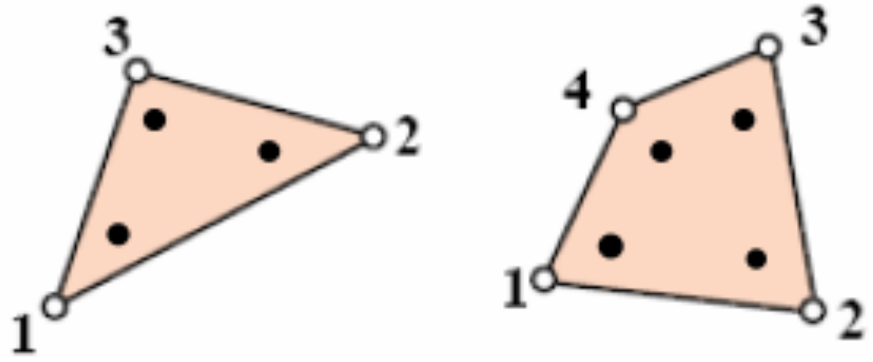
Sample Elements



- Main difference → Solution form within the element

FEM: Elements

- Three important parts that make up an element
  - Nodes
  - Integration Points
  - Shape Function (Internal interpolation function)



○ Node

● Integration Point

$\boxed{\#\text{Nodes} \neq \#\text{Integration Points}}$

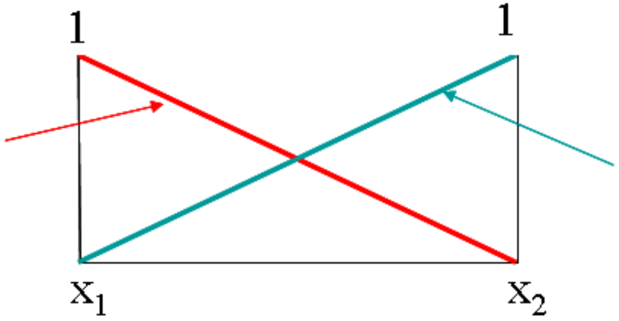
- Forces & displacements are defined at Nodes
- Stresses & Strains are defined at the Integration Points
- Nodes and Integration Points are linked via the shape functions

FEM: Elements

- Shape function describes how much each node affects the rest of the element
  - Internal interpolation functions
- 1 shape function per node
- Shape function can have various forms
  - Most common are linear & quadratic

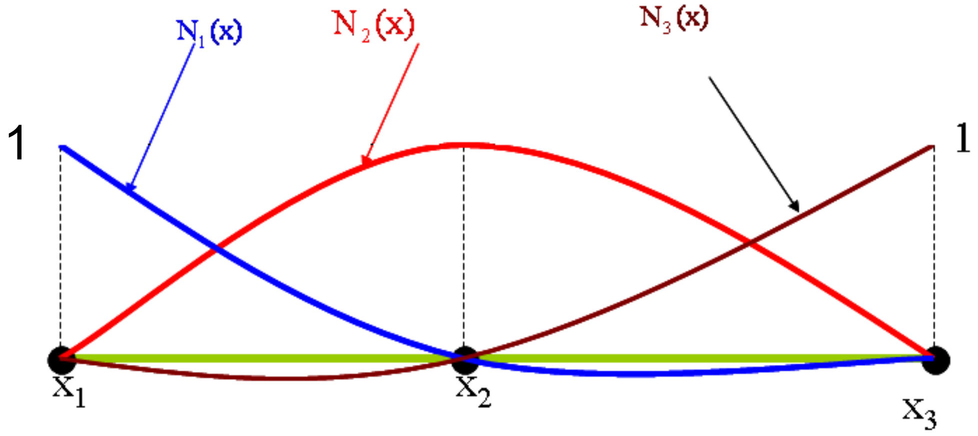
1D – Linear:

$$N_1(x) = \frac{x_2 - x}{x_2 - x_1}$$



$$N_2(x) = \frac{x - x_1}{x_2 - x_1}$$

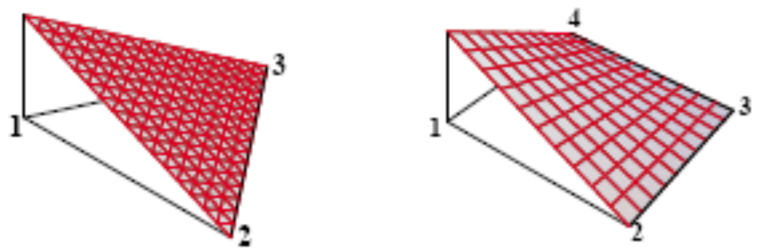
1D – Quad:



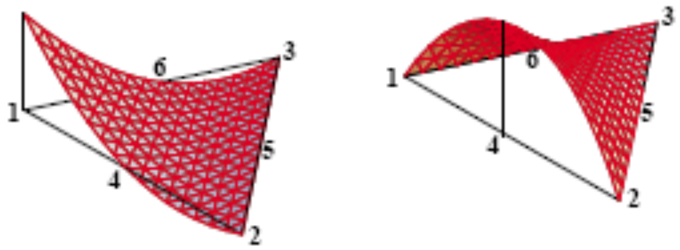
FEM: Elements

- 2D Shape functions are planes rather than lines

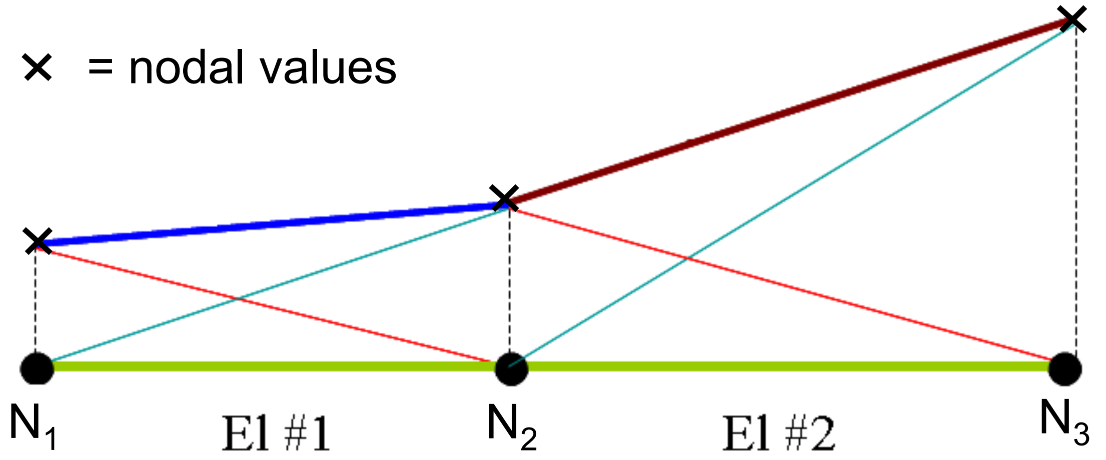
Linear Shape functions



Quad. Shape Functions

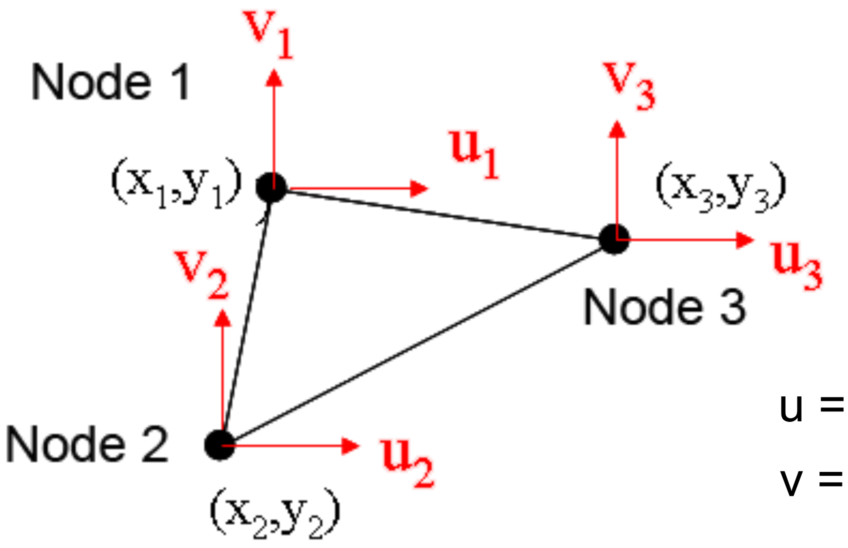


- Shape functions guarantee nodal based quantities (like force and displacement) are CONTINUOUS across element boundaries
- Shape function derivatives are NOT CONTINUOUS across element boundaries



FEM: Elements

2D Linear Element:



u = displacement in x

v = displacement in y

A displacement *approximation*

$$u(x,y) \approx N_1 u_1 + N_2 u_2 + N_3 u_3 \qquad v(x,y) \approx N_1 v_1 + N_2 v_2 + N_3 v_3$$

$$\begin{Bmatrix} u\,(x,y) \\ v\,(x,y) \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix}$$

FEM: Elements

FEM calculates strain from the nodal displacements

Definition of Strain

$$\varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{bmatrix}$$

FEM Strain Calculation

Based on the derivative of the shape functions

$$\varepsilon_x \approx \frac{\partial N_1(x,y)}{\partial x} u(x,y) + \frac{\partial N_2(x,y)}{\partial x} u(x,y) + \frac{\partial N_3(x,y)}{\partial x} u(x,y)$$

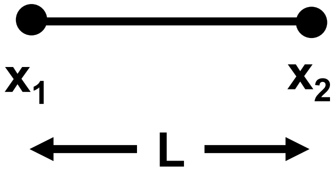
$$\varepsilon_y \approx \frac{\partial N_1(x,y)}{\partial y} v(x,y) + \frac{\partial N_2(x,y)}{\partial y} v(x,y) + \frac{\partial N_3(x,y)}{\partial y} v(x,y)$$

$$\varepsilon \approx \begin{bmatrix} \dfrac{\partial N_1(x,y)}{\partial x} & 0 & \dfrac{\partial N_2(x,y)}{\partial x} & 0 & \dfrac{\partial N_3(x,y)}{\partial x} & 0 \\ 0 & \dfrac{\partial N_1(x,y)}{\partial y} & 0 & \dfrac{\partial N_2(x,y)}{\partial y} & 0 & \dfrac{\partial N_3(x,y)}{\partial y} \\ \dfrac{\partial N_1(x,y)}{\partial y} & \dfrac{\partial N_1(x,y)}{\partial x} & \dfrac{\partial N_2(x,y)}{\partial y} & \dfrac{\partial N_2(x,y)}{\partial x} & \dfrac{\partial N_3(x,y)}{\partial y} & \dfrac{\partial N_3(x,y)}{\partial x} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ v_2 \\ u_2 \\ v_3 \\ u_3 \end{bmatrix}$$

- Because shape function derivatives are NOT CONTINUOUS across element boundaries, calculating ε at nodes could be a problem.
- ε is always calculated at <u>integration points</u> (inside the element)

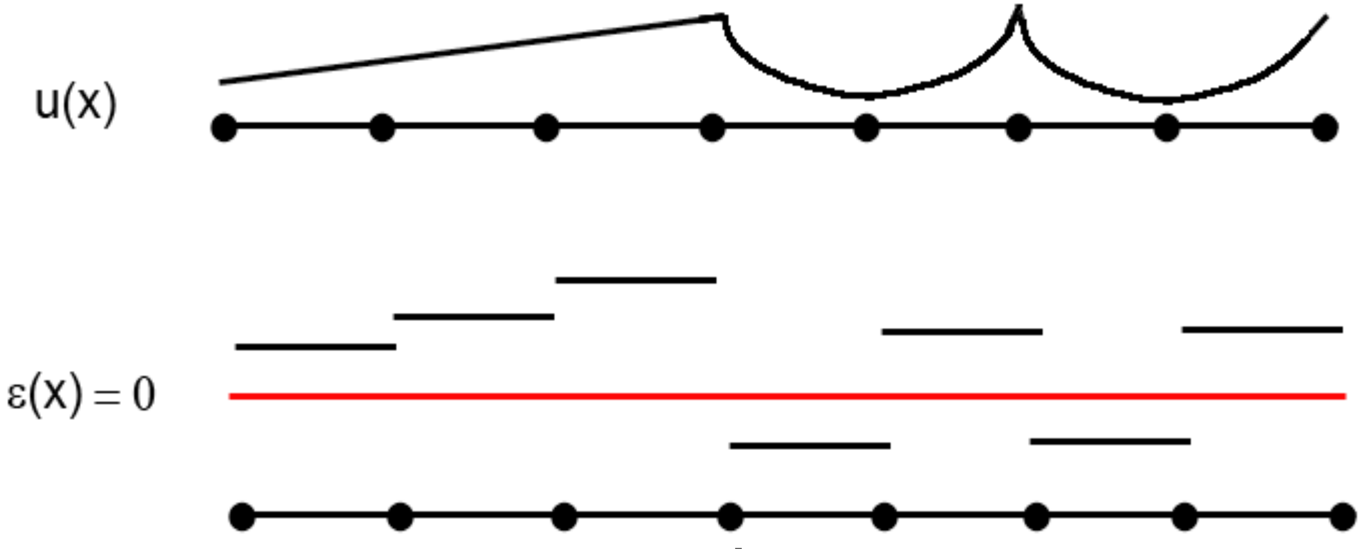
FEM: Elements

1D Element with Linear Shape Functions



Linear Shape Functions

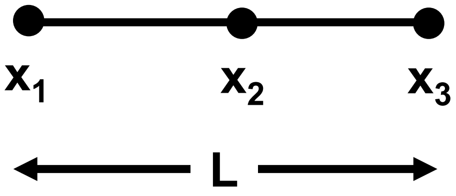
$$N_1(x) = \frac{x_2 - x}{x_2 - x_1} = \frac{x_2 - x}{L} \qquad N_2(x) = \frac{x - x_1}{x_2 - x_1} = \frac{x - x_1}{L}$$



Linear shape functions lead to elements that have a constant strain profile

FEM: Elements

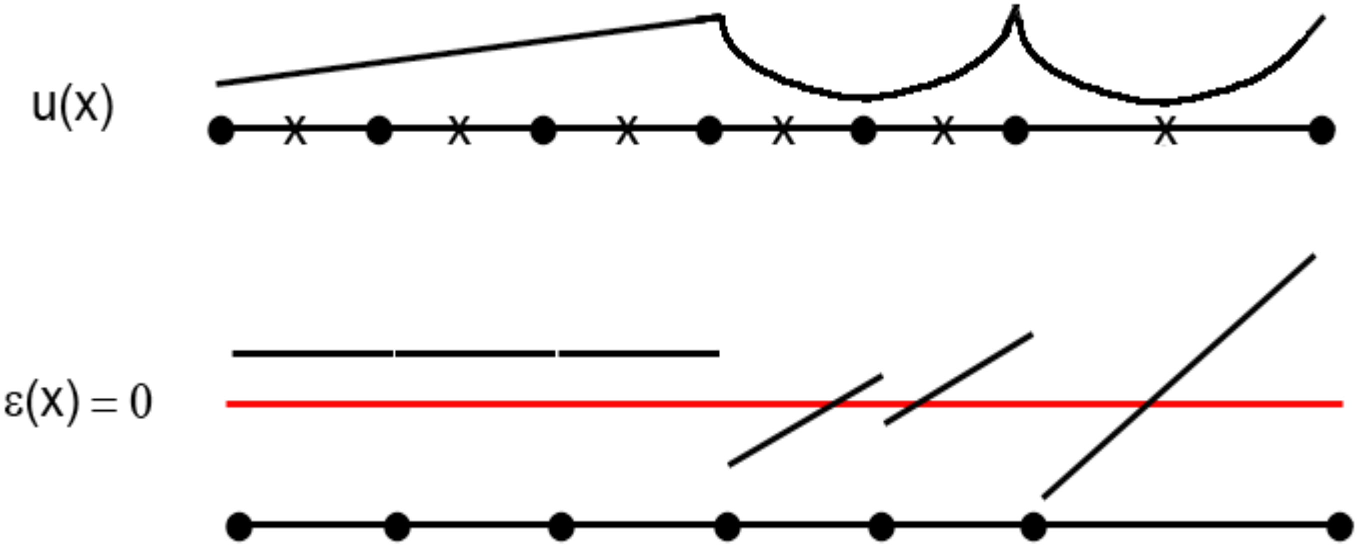
1D Element with Quadratic Shape Functions



Quad. Shape Functions

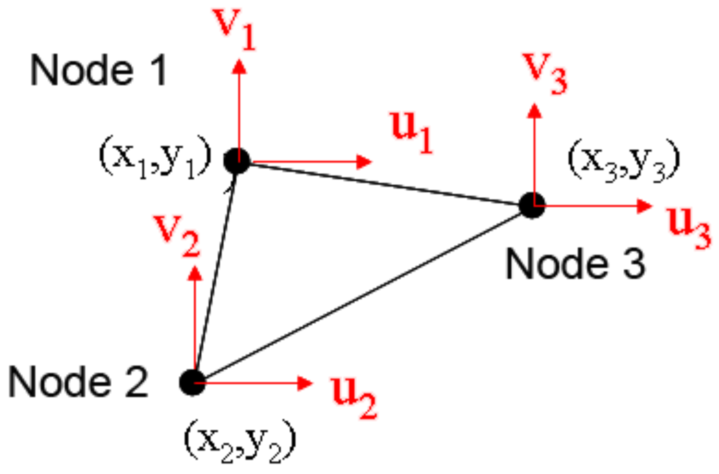
$$N_1(x) = \frac{(x_2 - x)(x_3 - x)}{(x_2 - x_1)(x_3 - x_1)} \qquad N_2(x) = \frac{(x_1 - x)(x_3 - x)}{(x_1 - x_2)(x_3 - x_2)}$$

$$N_3(x) = \frac{(x_1 - x)(x_2 - x)}{(x_1 - x_3)(x_2 - x_3)}$$



Quadratic shape functions lead to elements that have a linear strain profile

FEM: Elements



Linear Shape Functions

$$N_1 = \frac{a_1 + b_1 x + c_1 y}{2A}$$

$$N_2 = \frac{a_2 + b_2 x + c_2 y}{2A}$$

$$N_3 = \frac{a_3 + b_3 x + c_3 y}{2A}$$

$$[B] = \frac{1}{2A}\begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ b_1 & c_1 & b_2 & c_2 & b_3 & c_3 \end{bmatrix} \Rightarrow \begin{aligned} \varepsilon_x &= u_1 b_1 + u_2 b_2 + u_3 b_3 \\ \varepsilon_y &= u_1 c_1 + u_2 c_2 + u_3 c_3 \end{aligned}$$

Note: No dependence on x and y

Linear shape functions lead to elements that have a constant strain profile

FEM: Elements

Stress at each integration point

$$\sigma = [D][\varepsilon] = [D][B][u]$$

D = appropriate elasticity matrix (user input)

B = element dependent

Summary: For a 1D linear element

- **Approximate displacement**

$$\boxed{u(x) = \frac{1}{L}\begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}$$

- **Approximate Strain**

$$\boxed{\varepsilon = \frac{1}{L}\begin{bmatrix} -1 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}$$

- **Approximate Stress**

$$\boxed{\sigma = \frac{E}{L}\begin{bmatrix} -1 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}$$

FEM: Elements

FEM Method with Elements

- **Assemble individual element stiffness matrix**

$$[K_e] = \int_{V^e} [B_e]^T [D_e][B_e] dV$$

- **Assemble global stiffness matrix**

$$[K] = \sum_{\# Ele} [K_e]$$

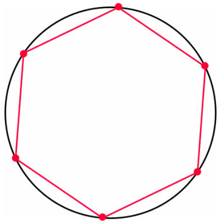
- **Apply Boundary Conditions**

$$[K][u] = [f]$$

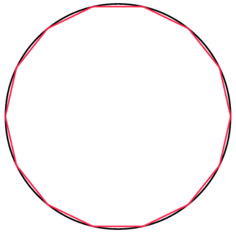
FEM ERROR

There are three general sources of error in a finite-element solution

#1: Errors due to the approximation of the domain



vs.



#2: Errors due to the approximation of the solution

$$EA\frac{d^2u}{dx^2} = F \qquad \text{vs.} \qquad \Pi(w) = \frac{1}{2}\int_0^L EA\left(\frac{dw}{dx}\right)^2 dx - \int_0^L bw\,dx - Fw(x=L)$$

#3: Errors due to numerical computation
(like numerical integration and round-off errors in a computer)

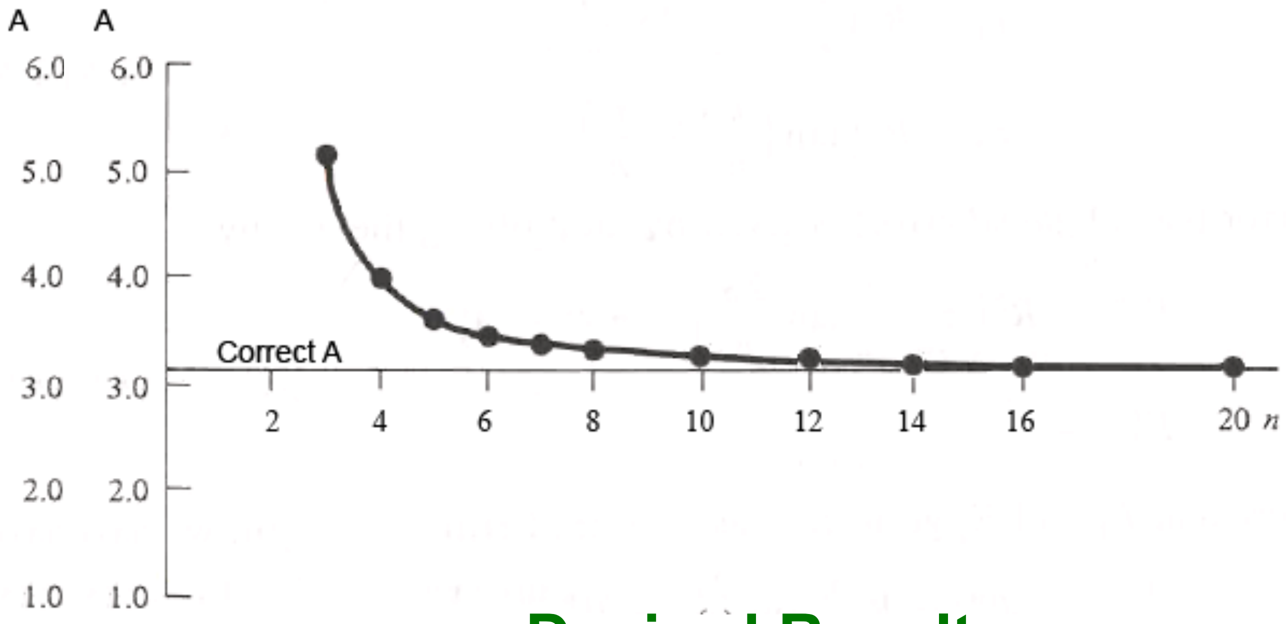
The estimation of these errors, in general, is not a simple matter.

FEM: Accuracy vs. Convergence

- The accuracy and convergence of the finite-element solution depends on a number of factors
  - Like the differential equation solved (or the variational form used) and the type of element.

- Accuracy
  - Difference between the exact solution and the finite-element solution

- Convergence
  - The accuracy of the solution as the number of elements in the mesh is increased



Desired Result

Converged solution is NOT necessarily an accurate solution

FEM: Overview

- FEM Partial Differential Equation Solver
  - Solves PDE by
    1) Breaking solution space into pieces (elements)
    2) Approximating the solution on each element
    3) Solving a force balance (equilibrium) equation

$$[K][u]=[f]$$

K = stiffness matrix (composed of element and material properties)

- FEM solves the weak form of the equilibrium equation
  - "Average" solution over the domain NOT the exact solution
- Three primary parts of an element
  - Nodes
  - Integration Points
  - Shape Functions

FEM: Overview

- Large number of element types and shapes
  - Element choice will affect your results

Linear elements → **Constant strain**

Quadratic elements → **Linear strain**

- FEM Error
  - Domain Approximation
  - Solution Approximation
  - Numerical Computations

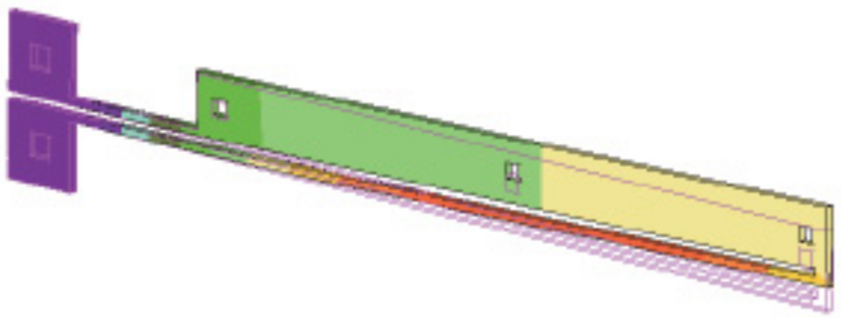
FEM error is difficult to quantify

- Convergence vs. Accuracy
  - A converged result is NOT necessarily an accurate result

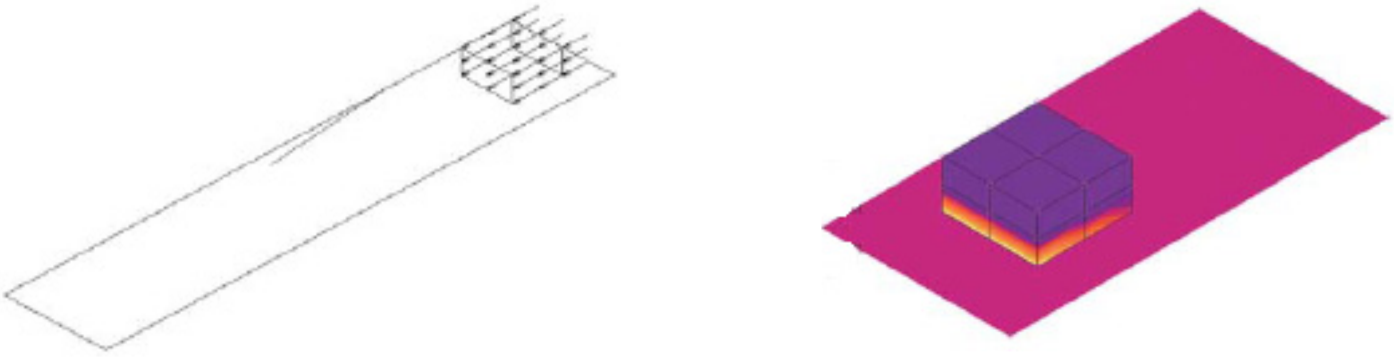
Any physically meaningful output MUST result from physical input provided by you to FEM

FEM: Cases where it works well

Examples:

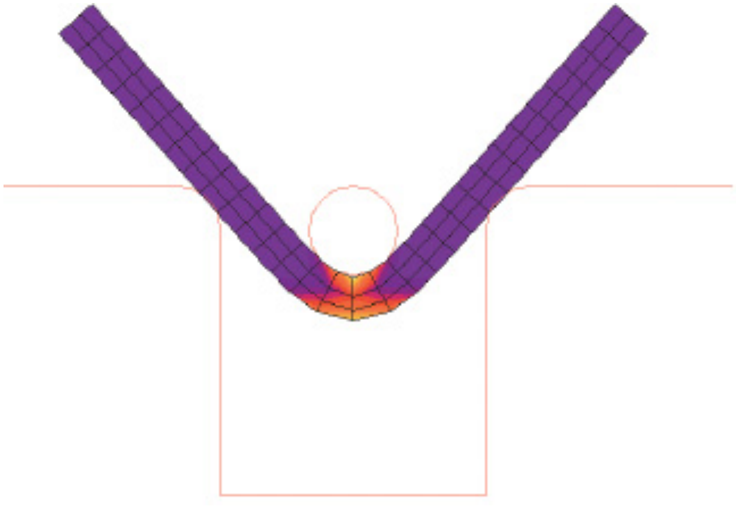


Coupled temperature and deformation problems

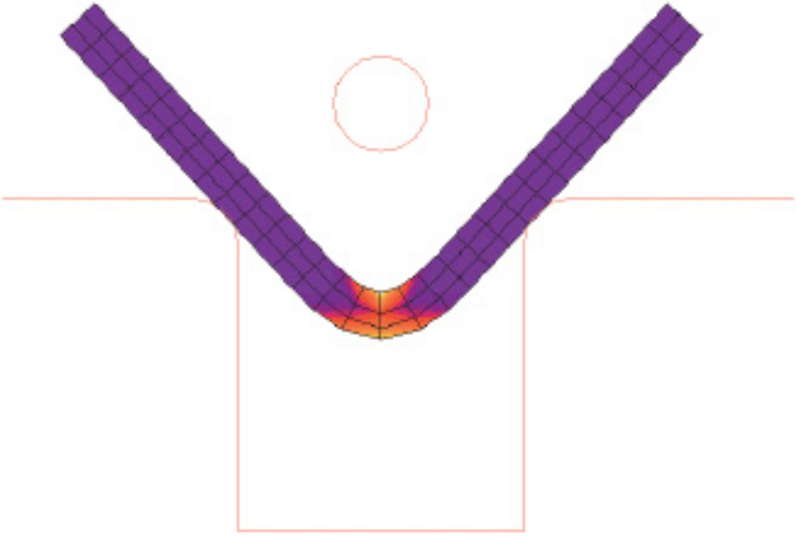


Time dependent temperature analysis due to friction

FEM: Cases where it works well



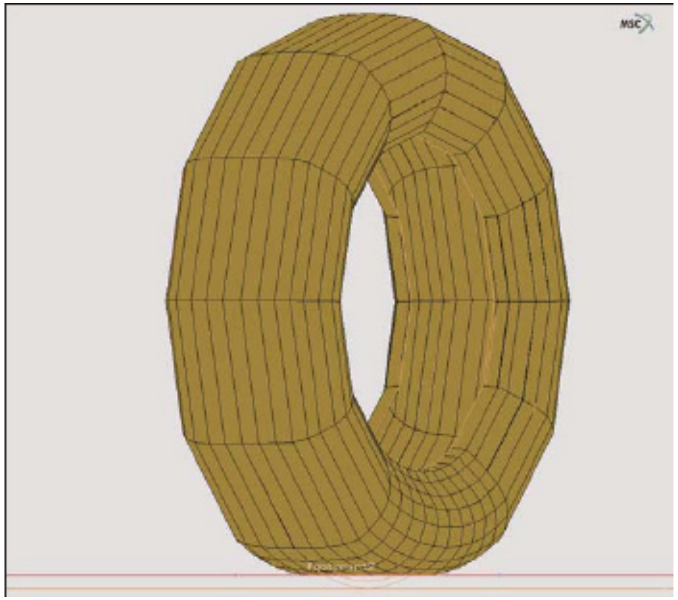
Complex deformations

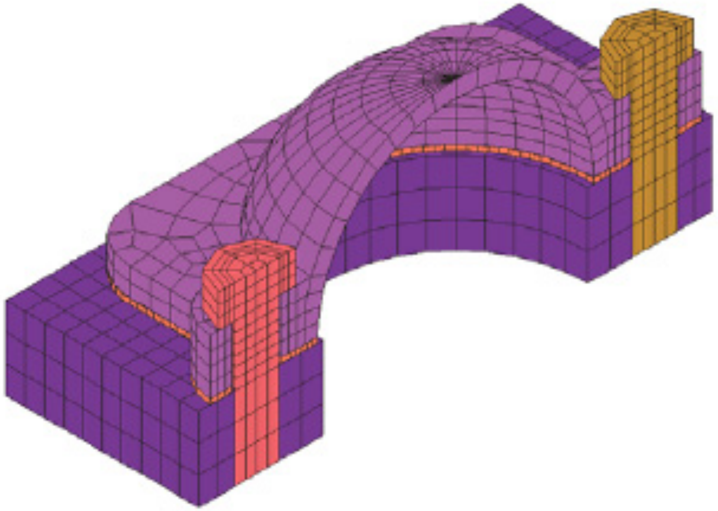


Elastic spring-back

FEM: Cases where it works well

Multiple components/complex geometries



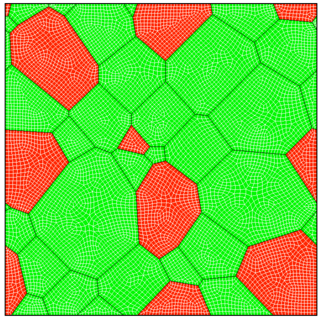


FEM: Cases where it does not work as well

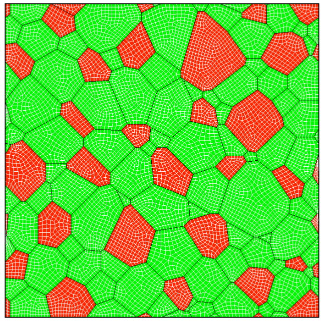
- 2 Material Science examples
  - FEM “works”
    - ⇨ Converges to a solution
    - ⇨ Solution is not correct
- Length scale issues
- Inhomogeneous Materials

FEM Example: Length Scales

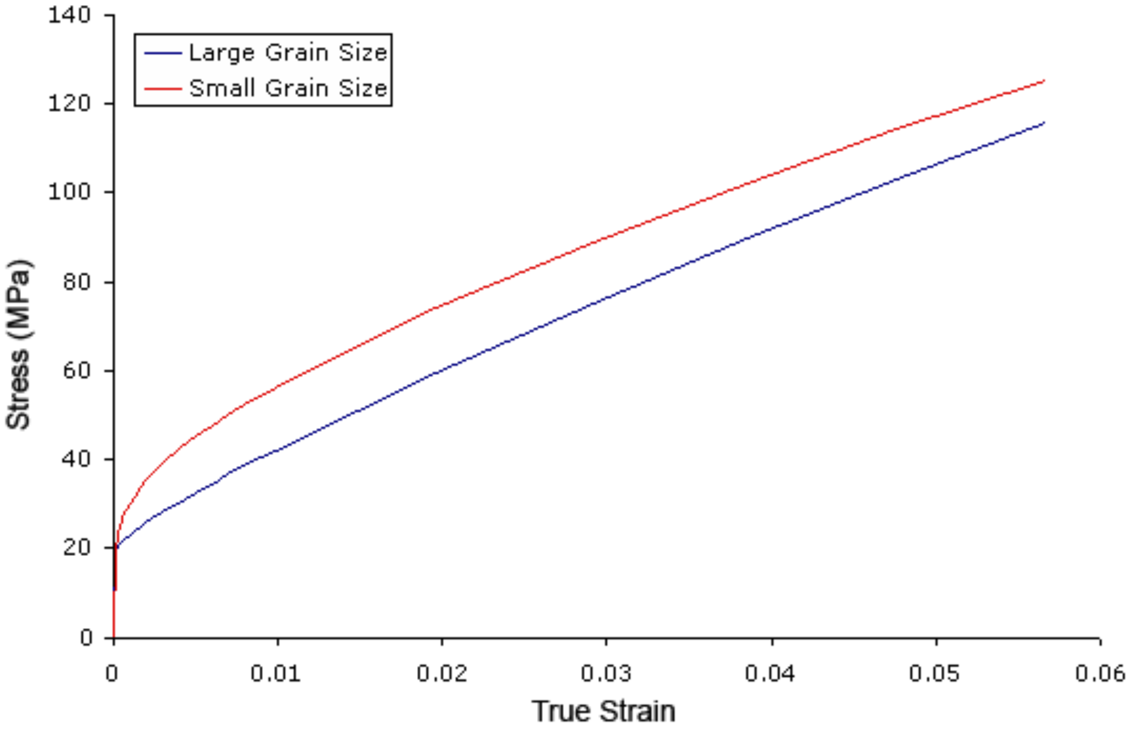
- Structural finite element codes are continuum based
- Finite element mesh has dimensions
- Local models: dimensions do not effect the σ−ε result
- Predicted σ−ε results from two different grain sizes are similar



150 um



3.4 um



FEM Example: Overview

Motivation

- Accounting for microstructure heterogeneity within a material computationally expensive.
- At the microscopic scale, strain and stress path are complex.
- Generally, it is not possible to exactly mesh and simulate the material’s microstructure

Problem Statement

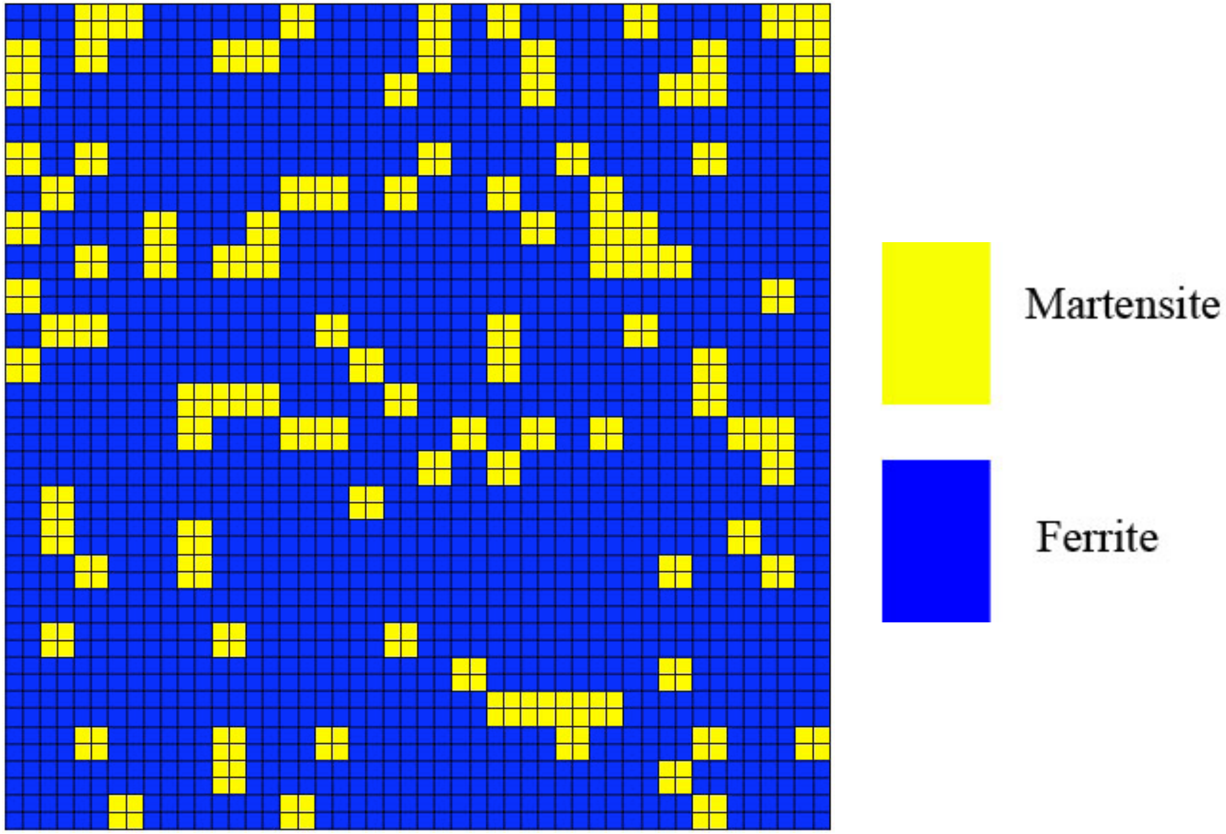
To investigate the effect of element type and mesh resolution on the σ-ε response of a two phase material.

(Hard Phase: Martensite)   (Soft Phase: Ferrite)

FEM Example: Mesh Refinement

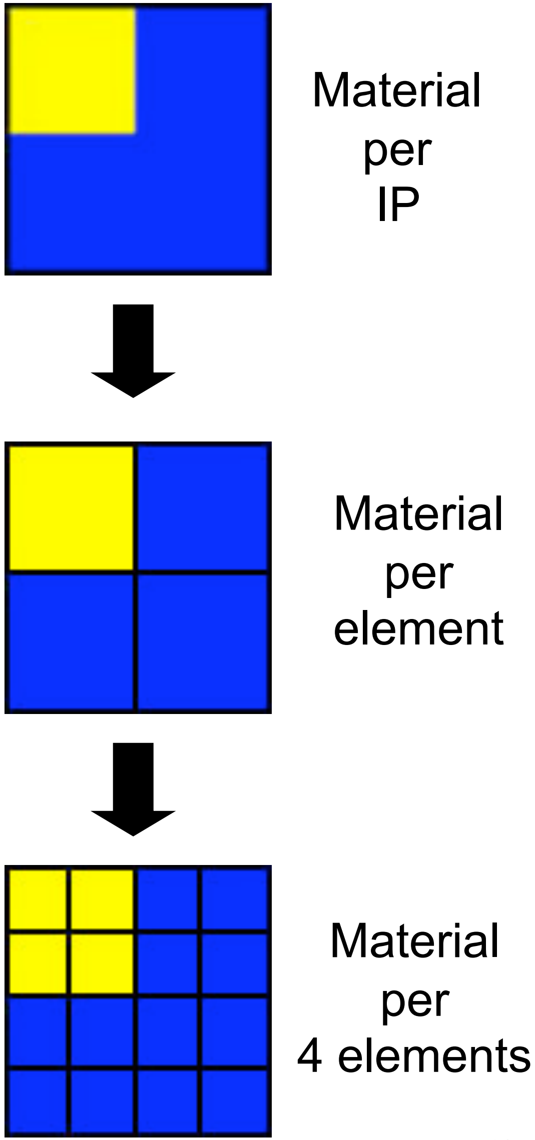
Coarsest Mesh: 1 Material per Int. Point

Finest Mesh: 1 Material per 4 elements



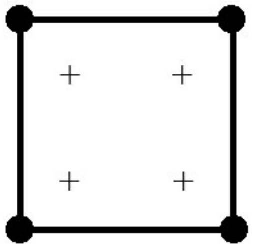
12x12 elements (144 total elements)

48x48 elements (2304 total elements)

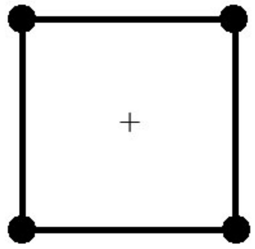


FEM Example: Element Types

- Element Types
  - 2D Linear (4 nodes)

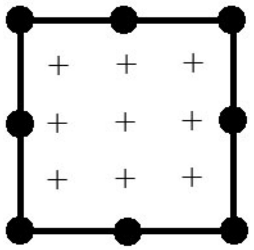


Full Integration

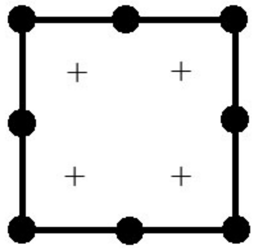


Reduced Integration

  - 2D Quadratic (9 nodes)



Full Integration



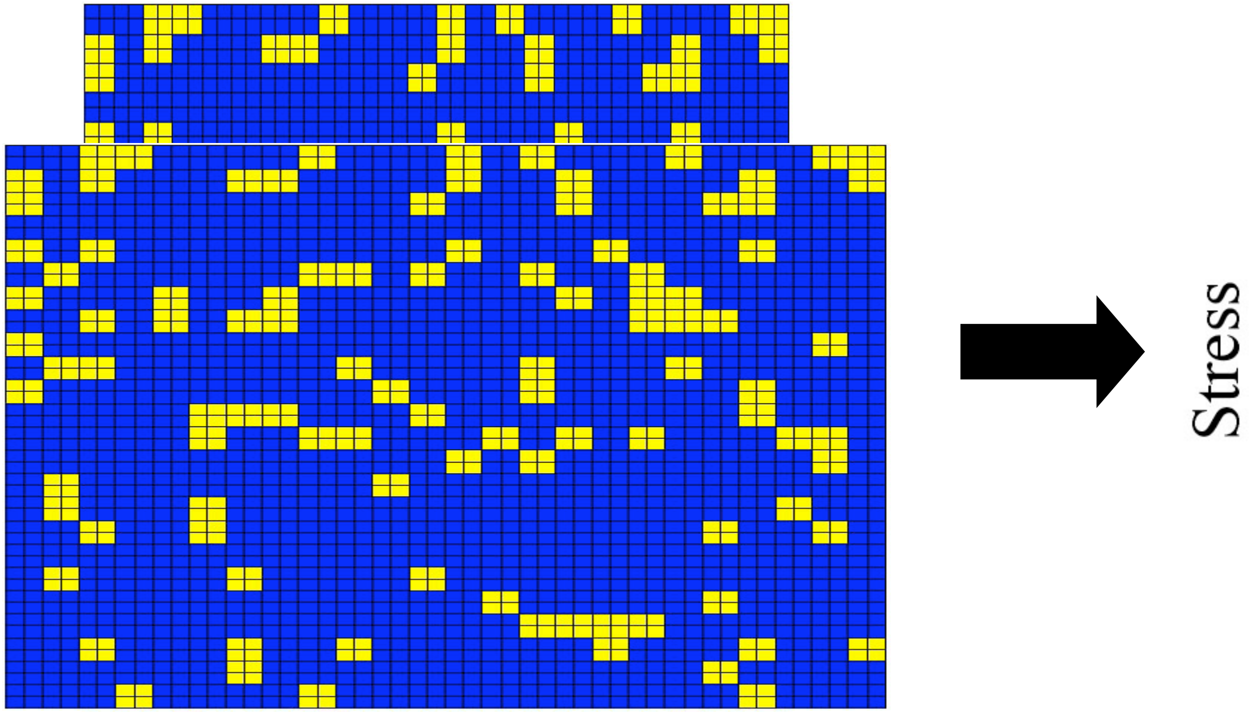
Reduced Integration

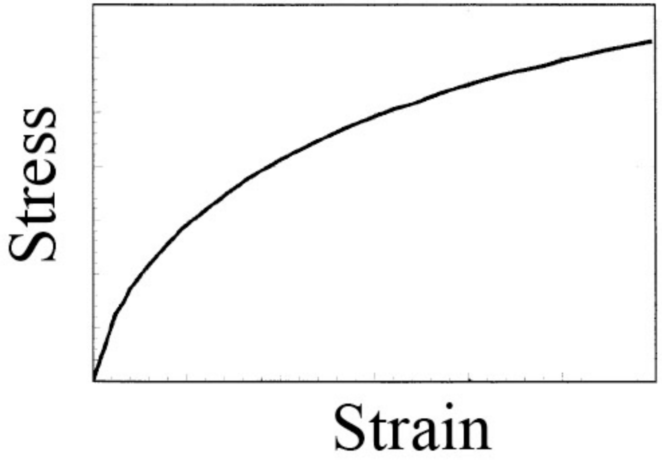
FEM Example: Results

2D Plane Strain Rolling

- Prescribe displacements
- Volume conserving ($\varepsilon_x = \varepsilon_y$)

- 30 % thickness reduction
- Elastic-plastic constitutive equations

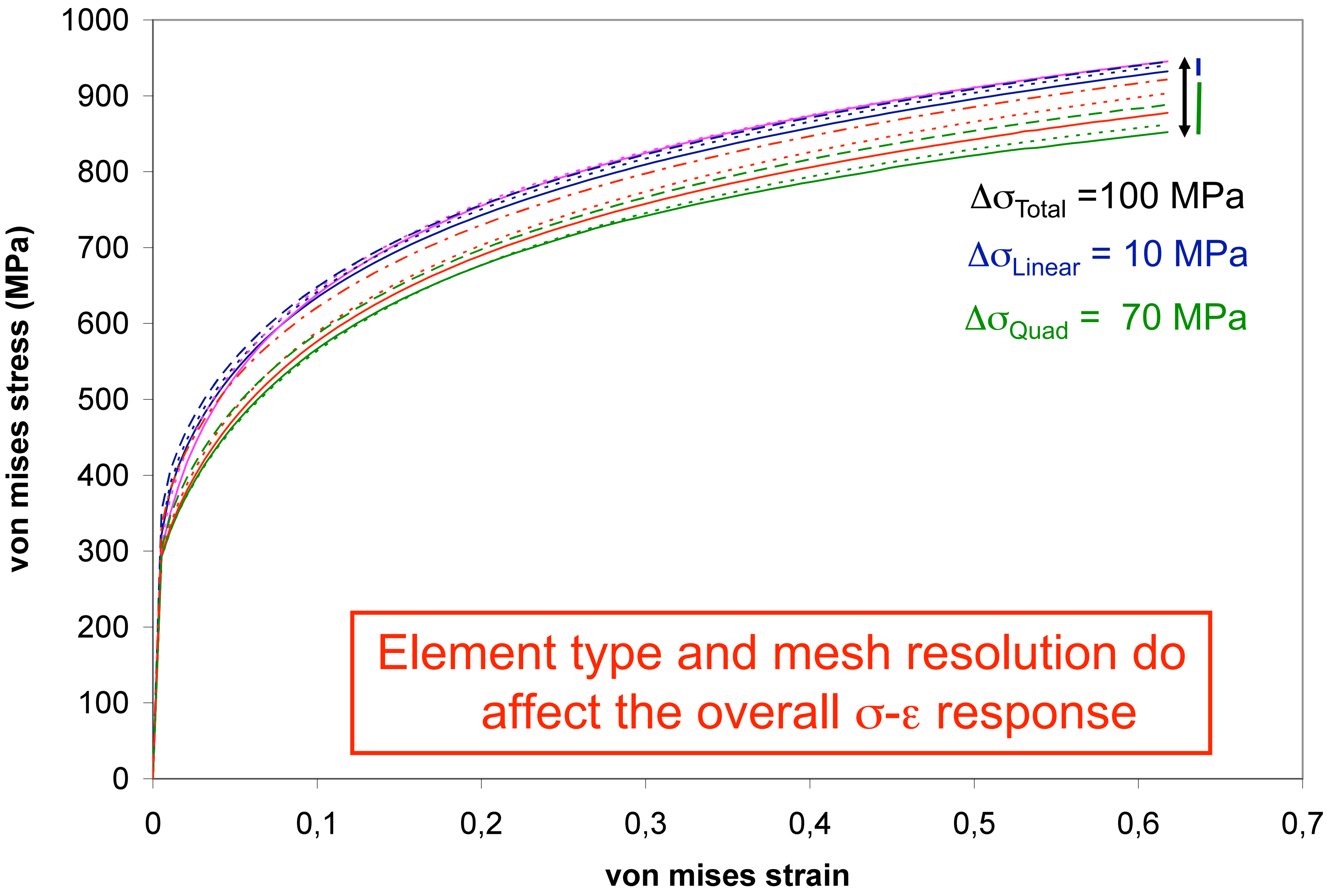




Study the effect of element type & mesh resolution on the σ-ε response

FEM Example: σ-ε Results

σ-ε curves of 4 different element types with 3 different mesh resolutions



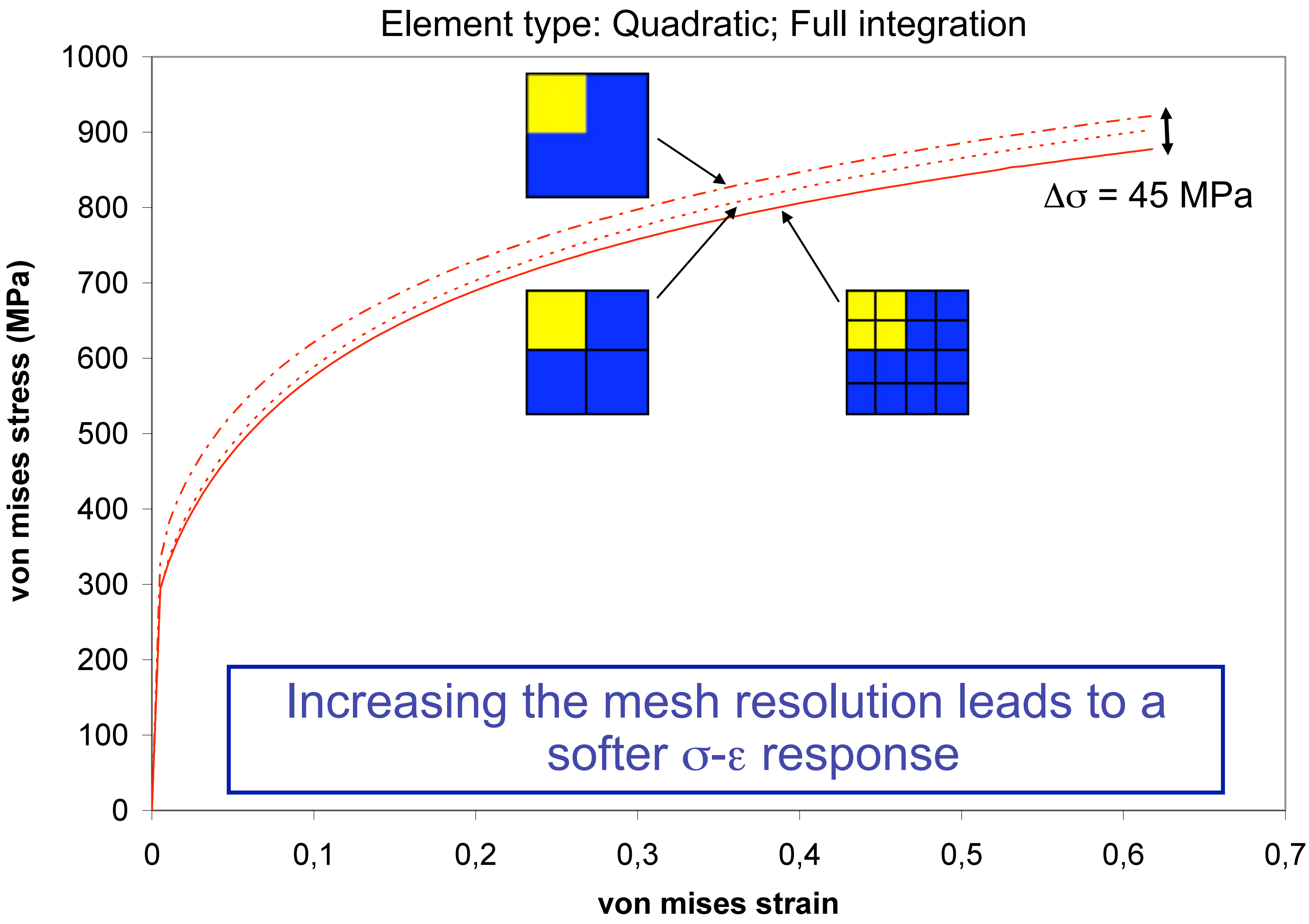
$\Delta\sigma_{Total}$ =100 MPa

$\Delta\sigma_{Linear}$ = 10 MPa

$\Delta\sigma_{Quad}$ = 70 MPa

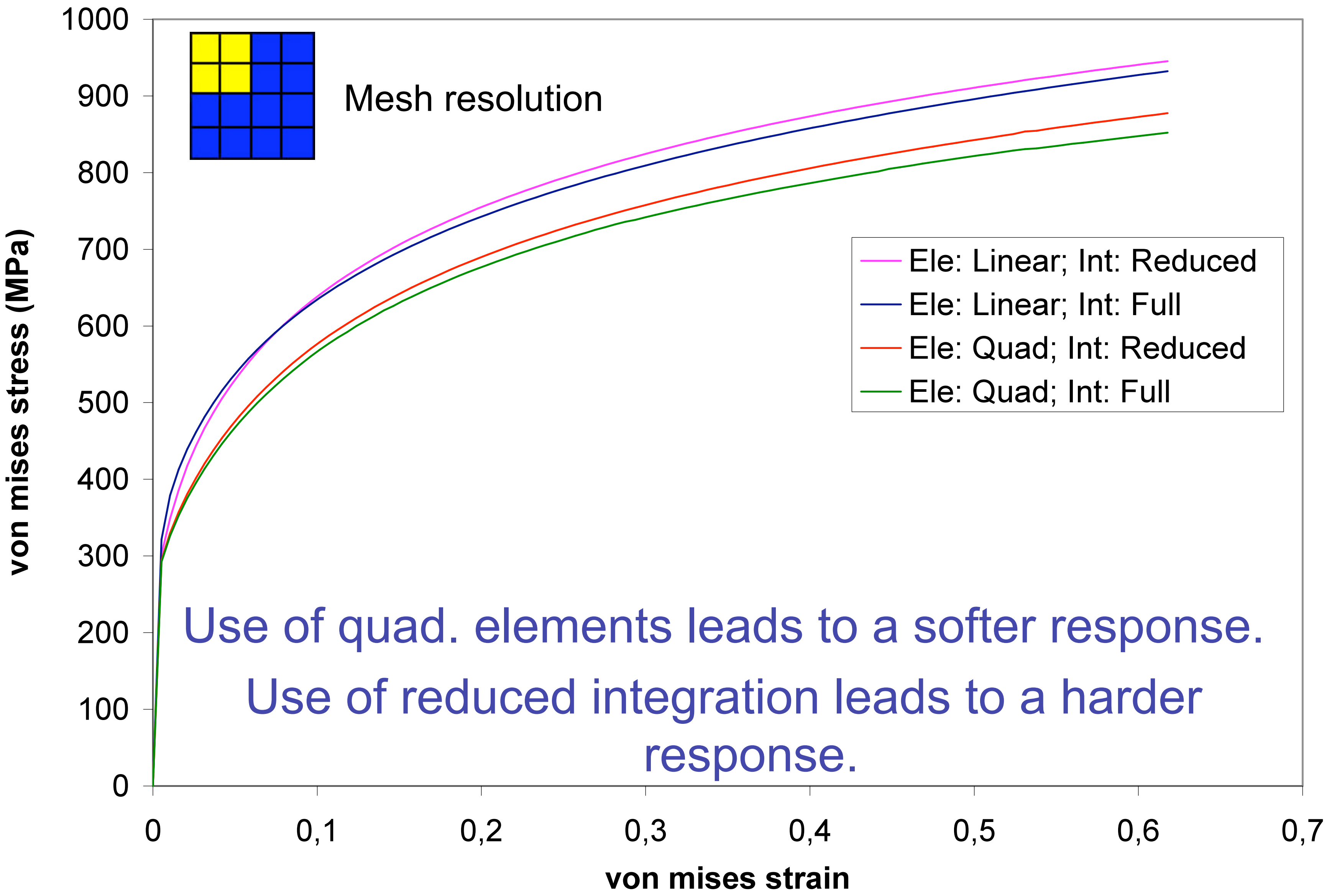
Element type and mesh resolution do affect the overall σ-ε response

FEM Example: Effect of Mesh Refinement



Increasing the mesh resolution leads to a softer σ-ε response

FEM Example: Effect of Element Type

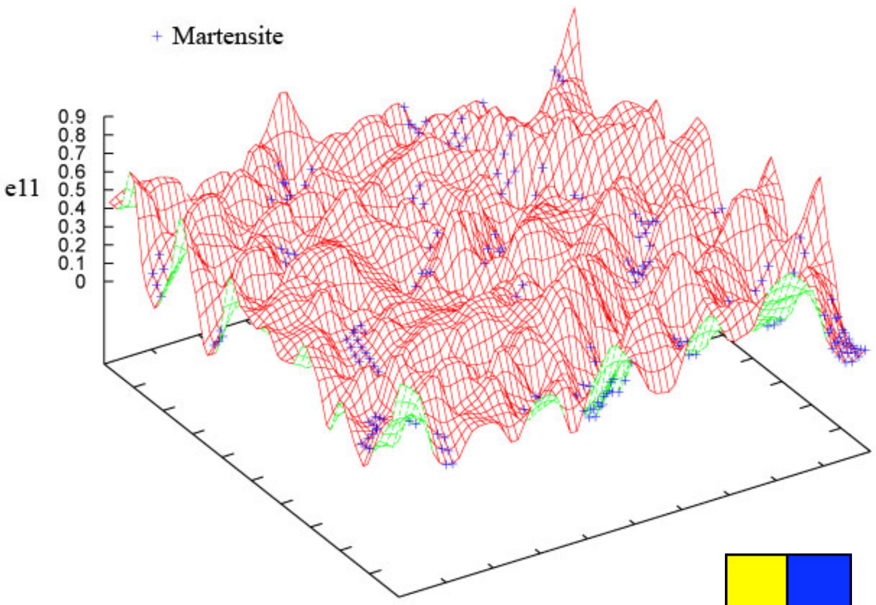


Use of quad. elements leads to a softer response.

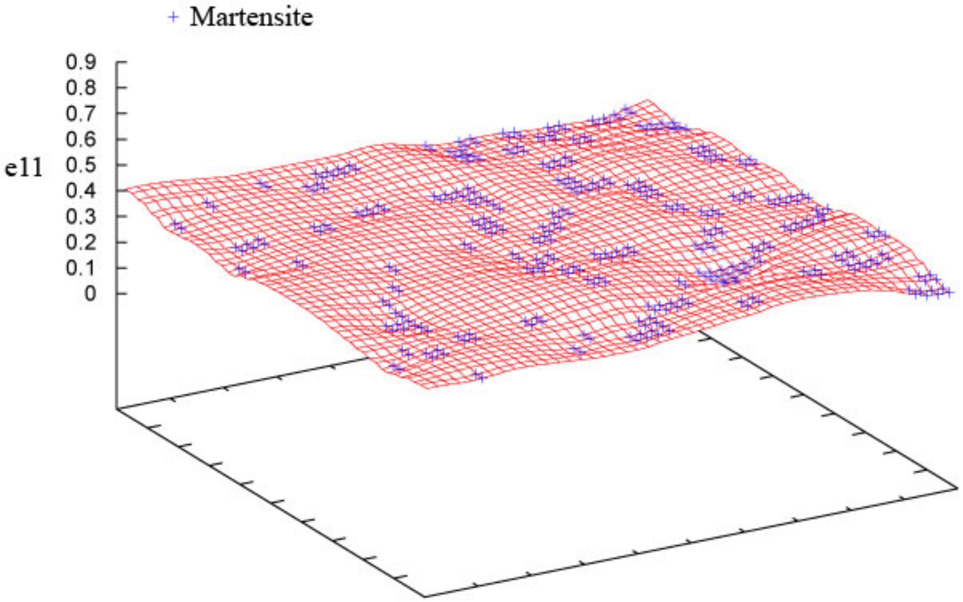
Use of reduced integration leads to a harder response.

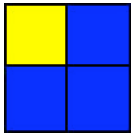
FEM Example: Strain Maps

Quad Element; Full Integration



Linear Element; Full Integration



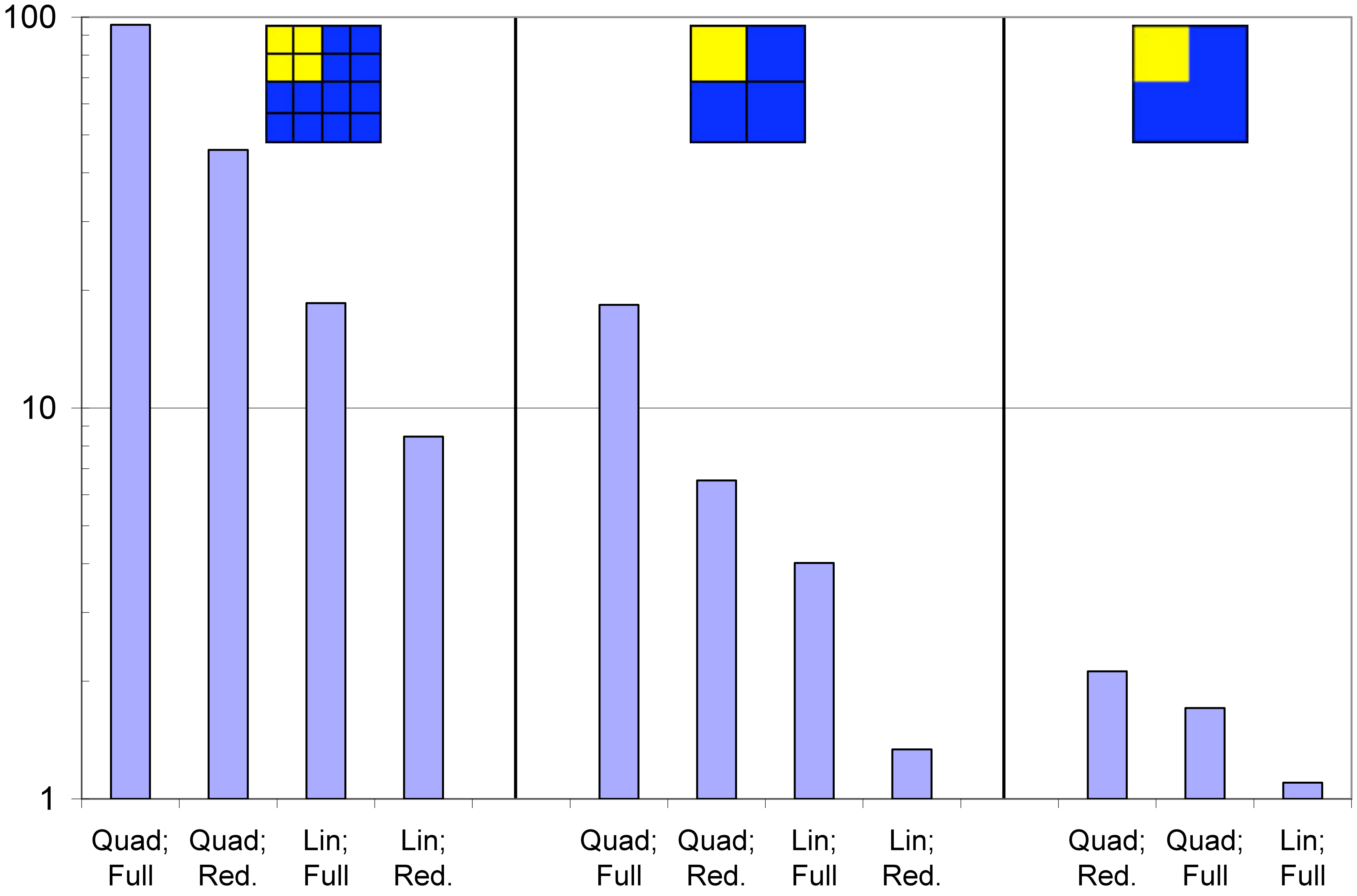


Mesh resolution

Strain state appears reasonable

- Linear elements ➡ ε is relatively flat
- Quad elements ➡ ε flows around the martensite

FEM Example: Performance



Use of linear elements GREATLY reduces computation time

FEM Example: Conclusions

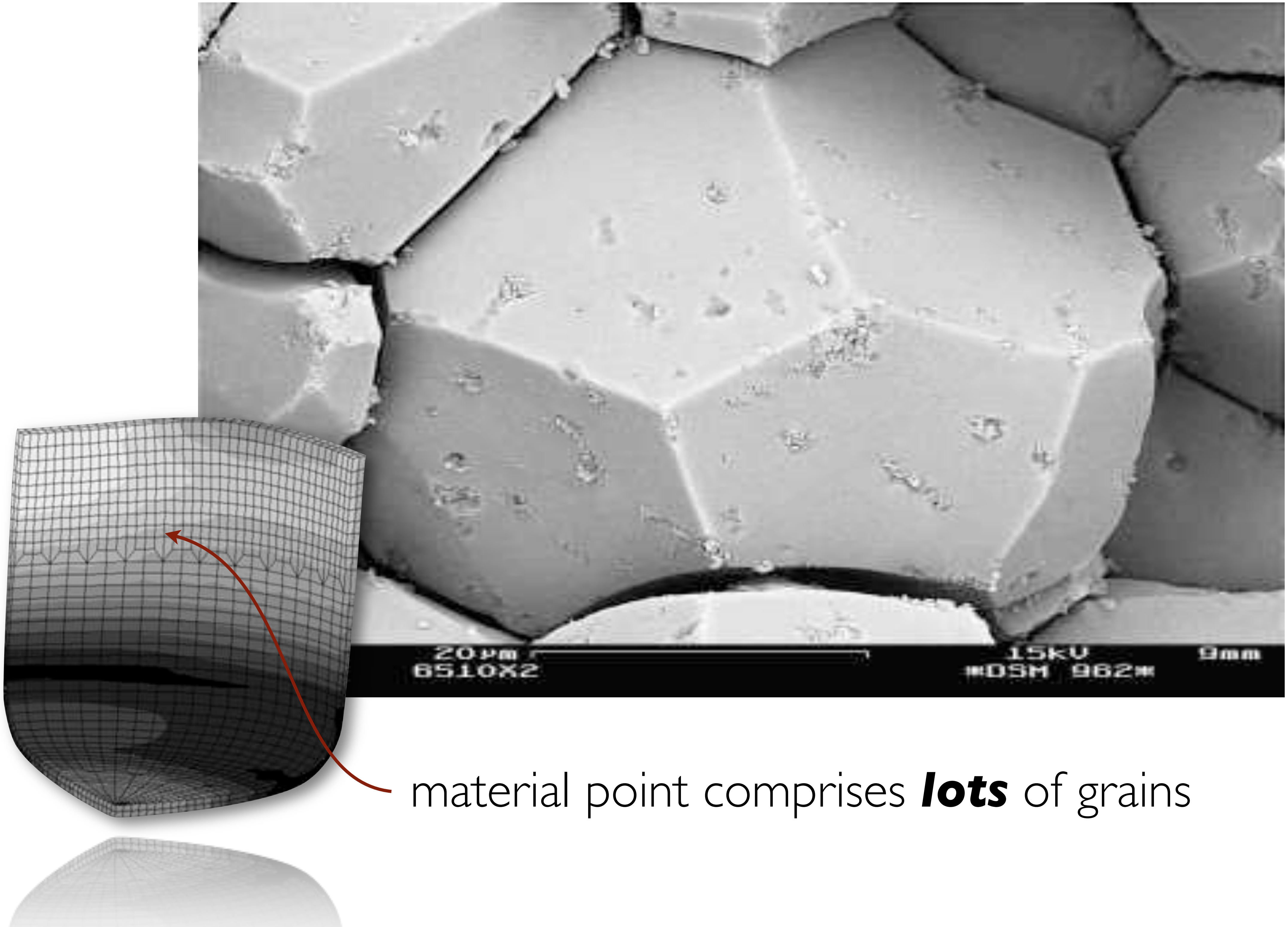
- Mesh dimensions do not enter into σ-ε constitutive relationship
- Element type and mesh resolution do affect the overall σ-ε response
  - Δσ = 100 MPa
- Increasing the mesh resolution leads to a softer σ-ε response
- Use of quadratic elements leads to a softer σ-ε response
- Linear elements predict a relatively flat e profile
- Quadratic elements predict a much more contoured e profile
- Use of linear elements GREATLY reduces computation time

content

PART IV

Polycrystal Plasticity Models

motivation



material point comprises ***lots*** of grains

representation of texture

- orientation, $\mathbf{g}$, of crystallite can be specified by a set of three Euler angles

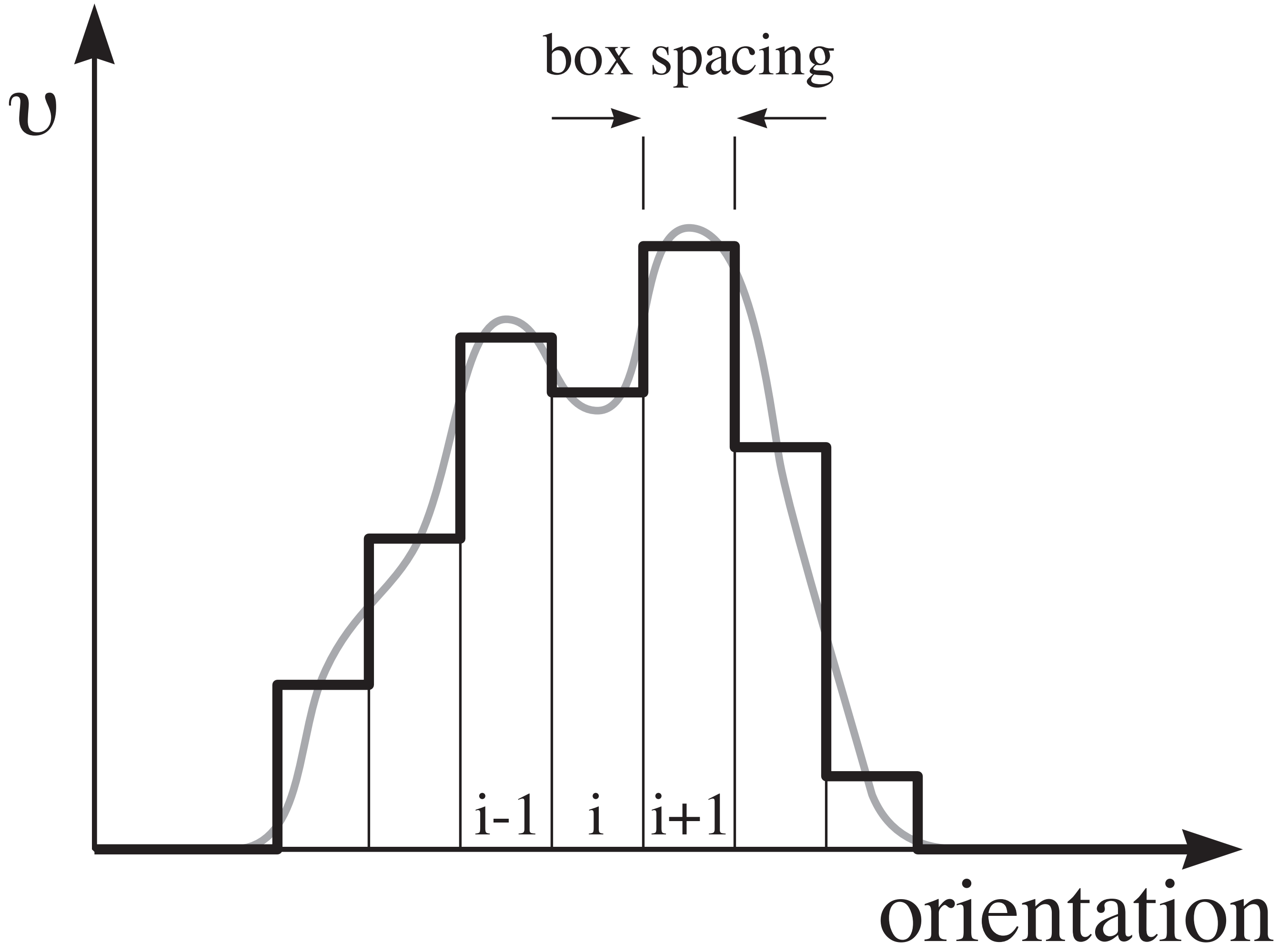
$$\mathbf{g} = \{\varphi_1\ , \phi, \varphi_2\}$$

- crystallite orientation distribution function (codf) defines probability, $f(\mathbf{g})$, that a volume fraction, $\mathbf{dV/V}$, of the polycrystaline aggregate is taken up by crystals of orientation between $\mathbf{g}$ and $\mathbf{g}+\mathbf{dg}$

$$v \equiv \frac{\mathrm{d}V}{V} = f(\mathbf{g})\mathrm{d}\mathbf{g}$$

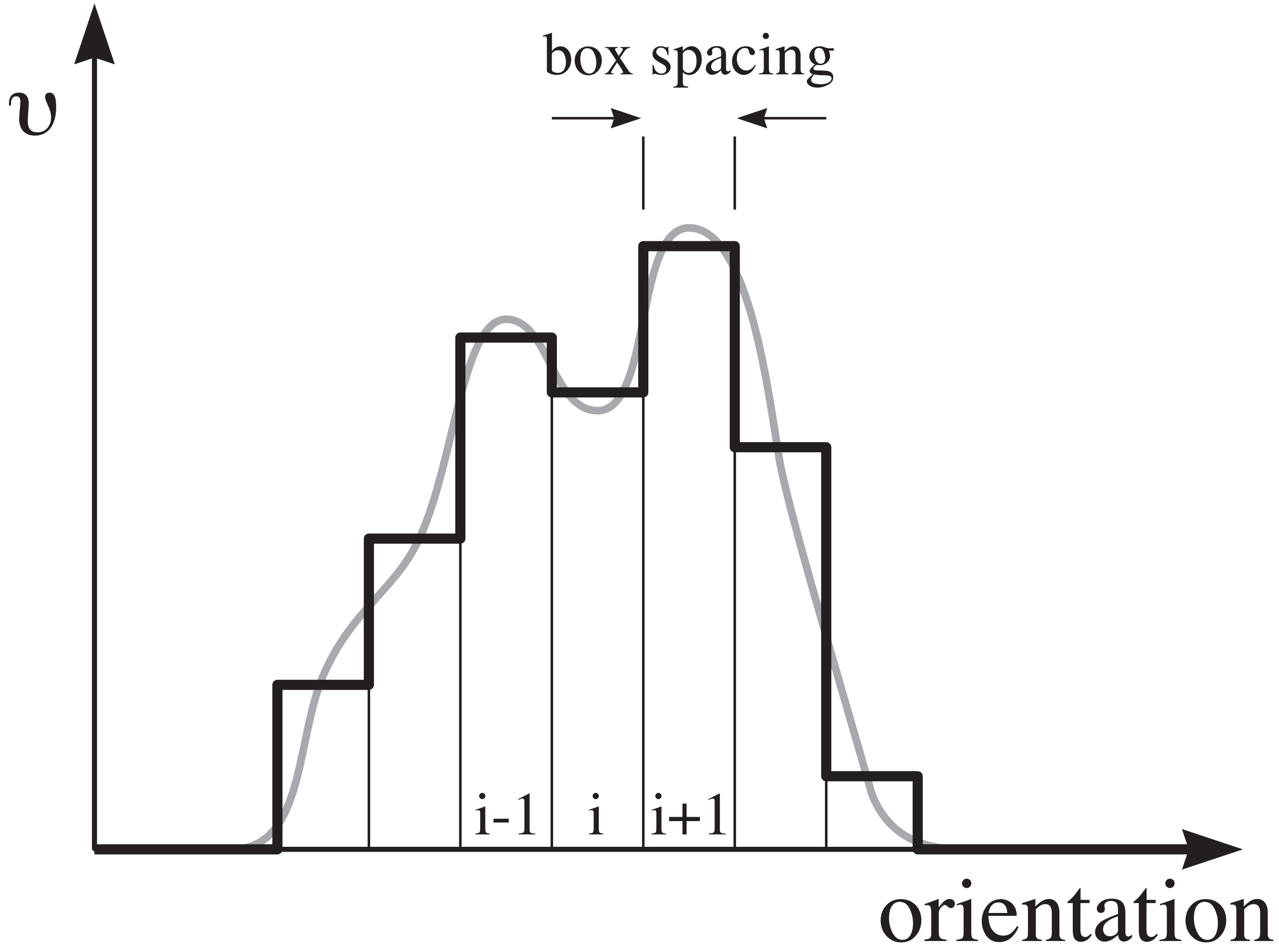
- codf values typically available on a discrete grid in orientation space, or continuously from coefficients of a harmonic series expansion

goal

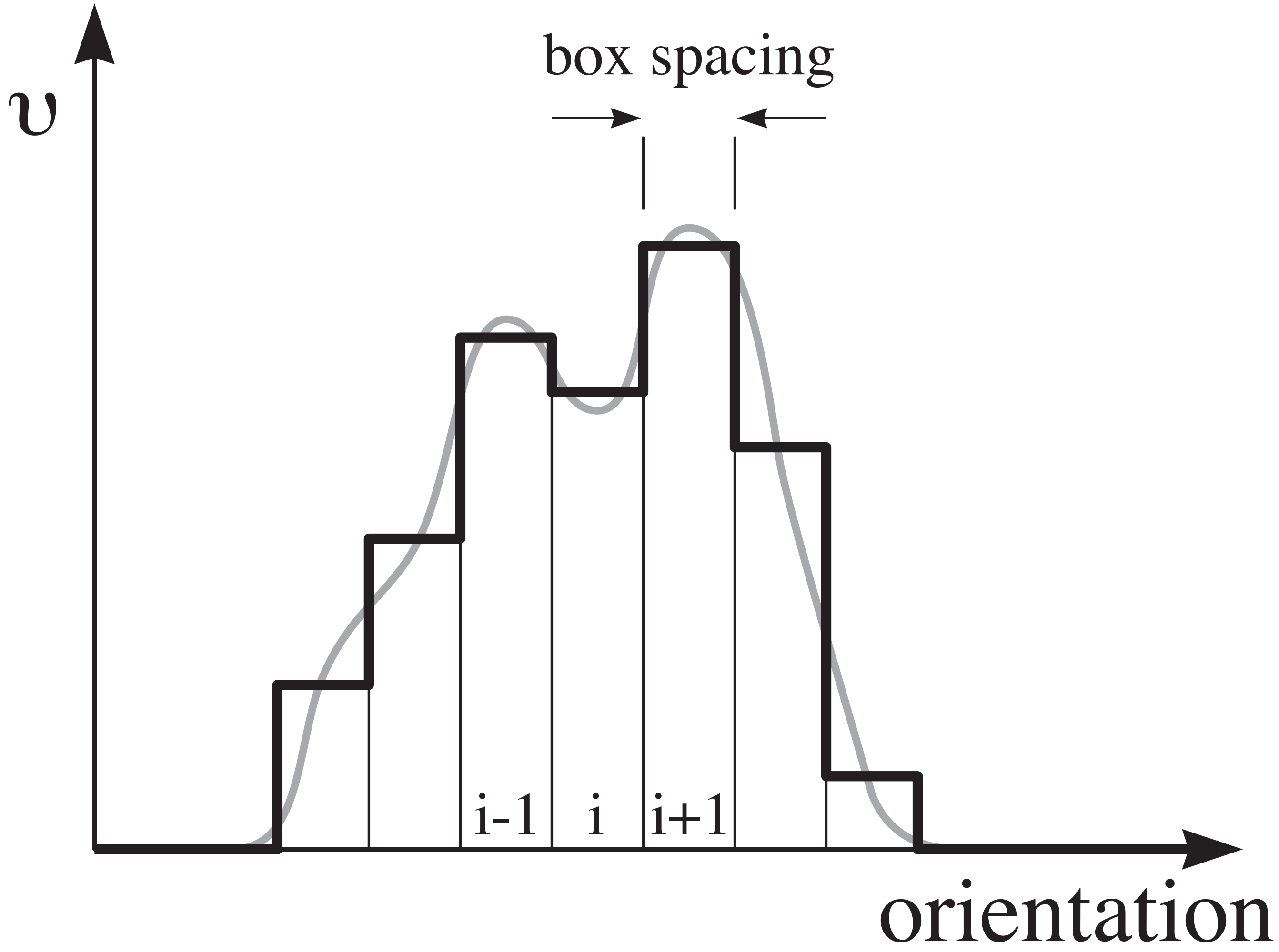


goal

- select N^* from all those N orientations having non-zero $\boldsymbol{\upsilon}_i$ in discrete codf representation



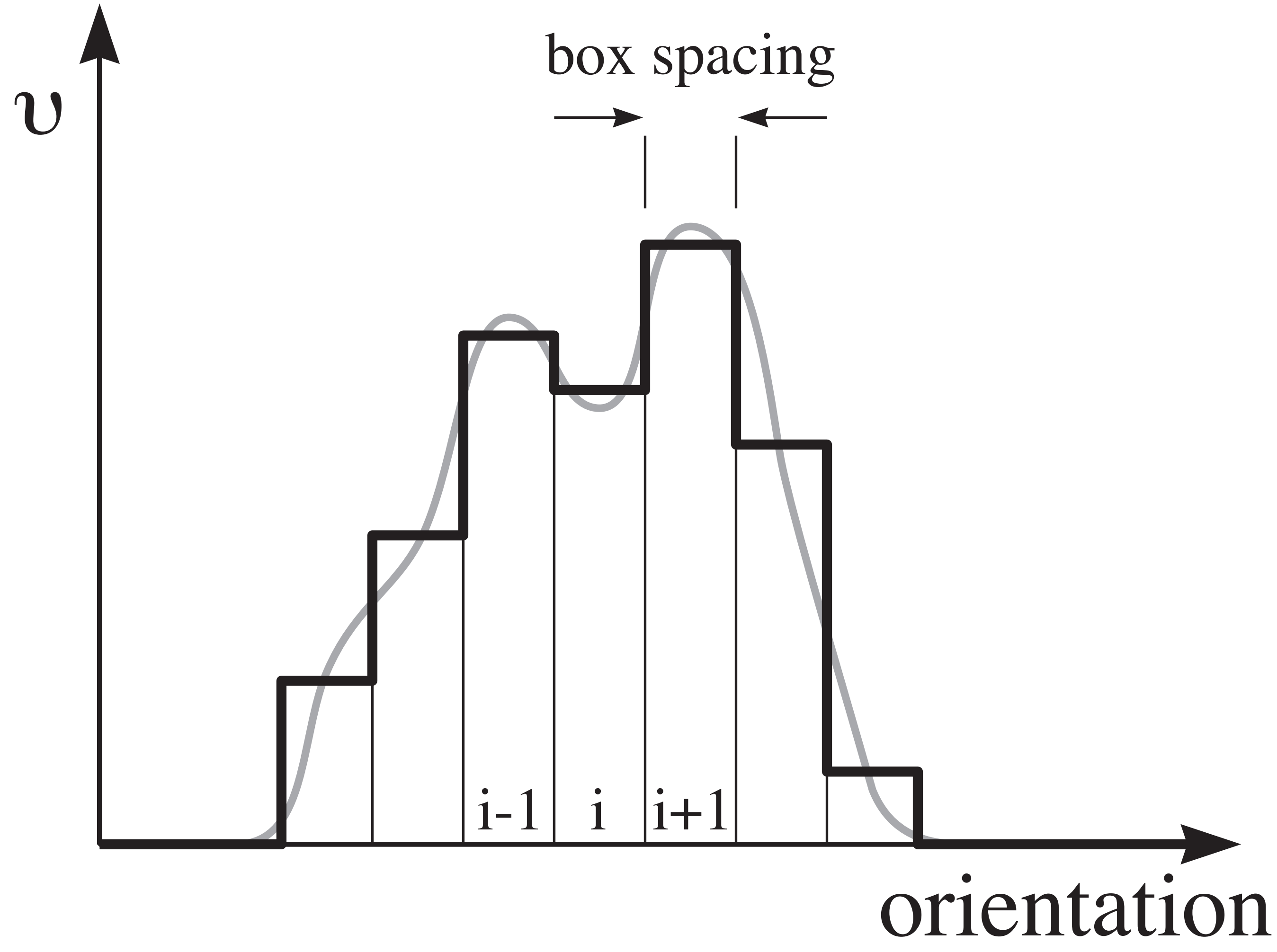
probabilistic reconstruction



probabilistic reconstruction

- add randomly selected orientation if random number $r \in [0, 1] \leq v_i$

- continue until N^* orientations collected

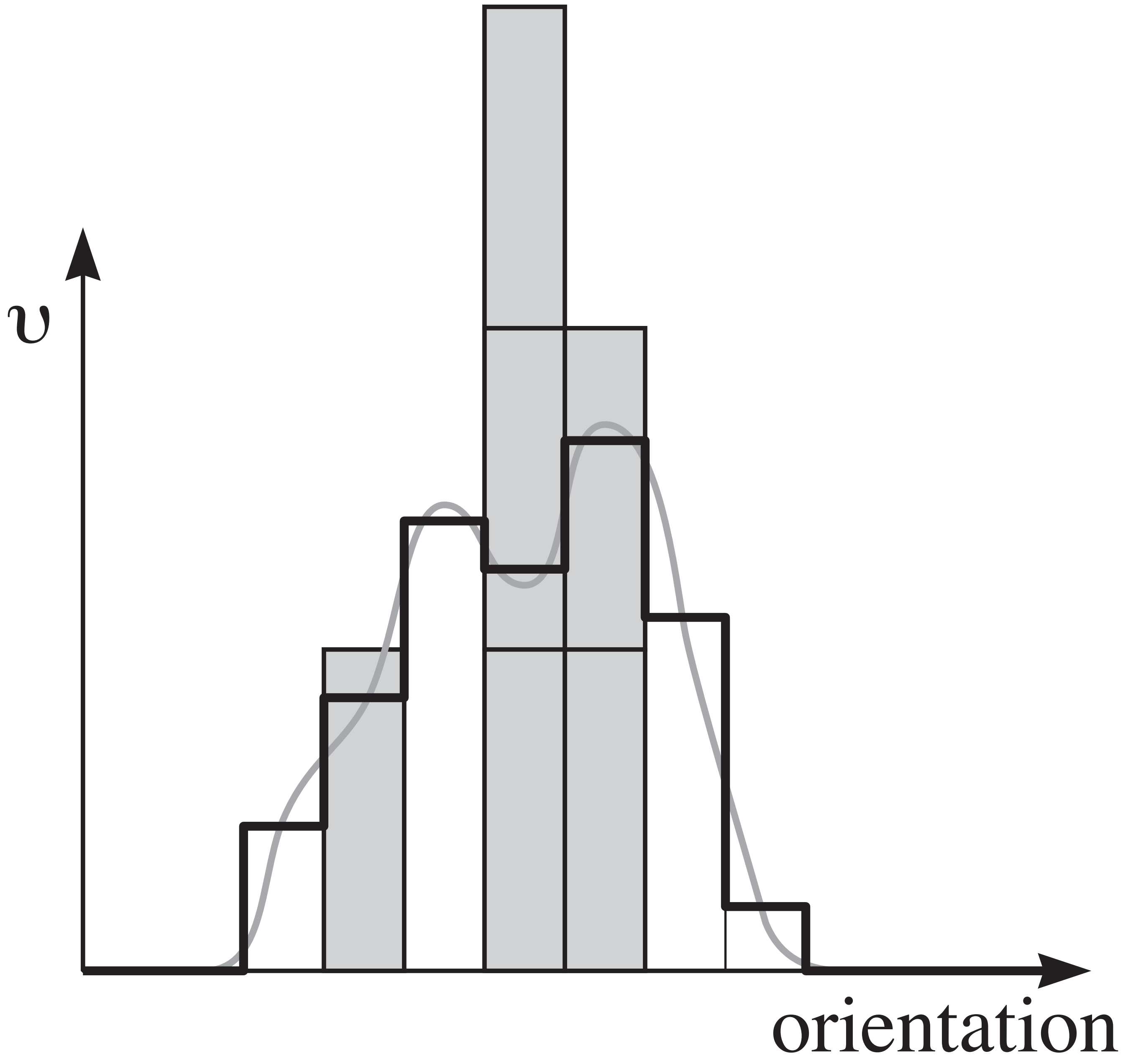


L.S. Tóth, P. Van Houtte, Textures Microstruct. 19 (1992) 229–244.

probabilistic reconstruction

- add randomly selected orientation if random number
  $r \in [0, 1] \leq v_i$

- continue until N^* orientations collected

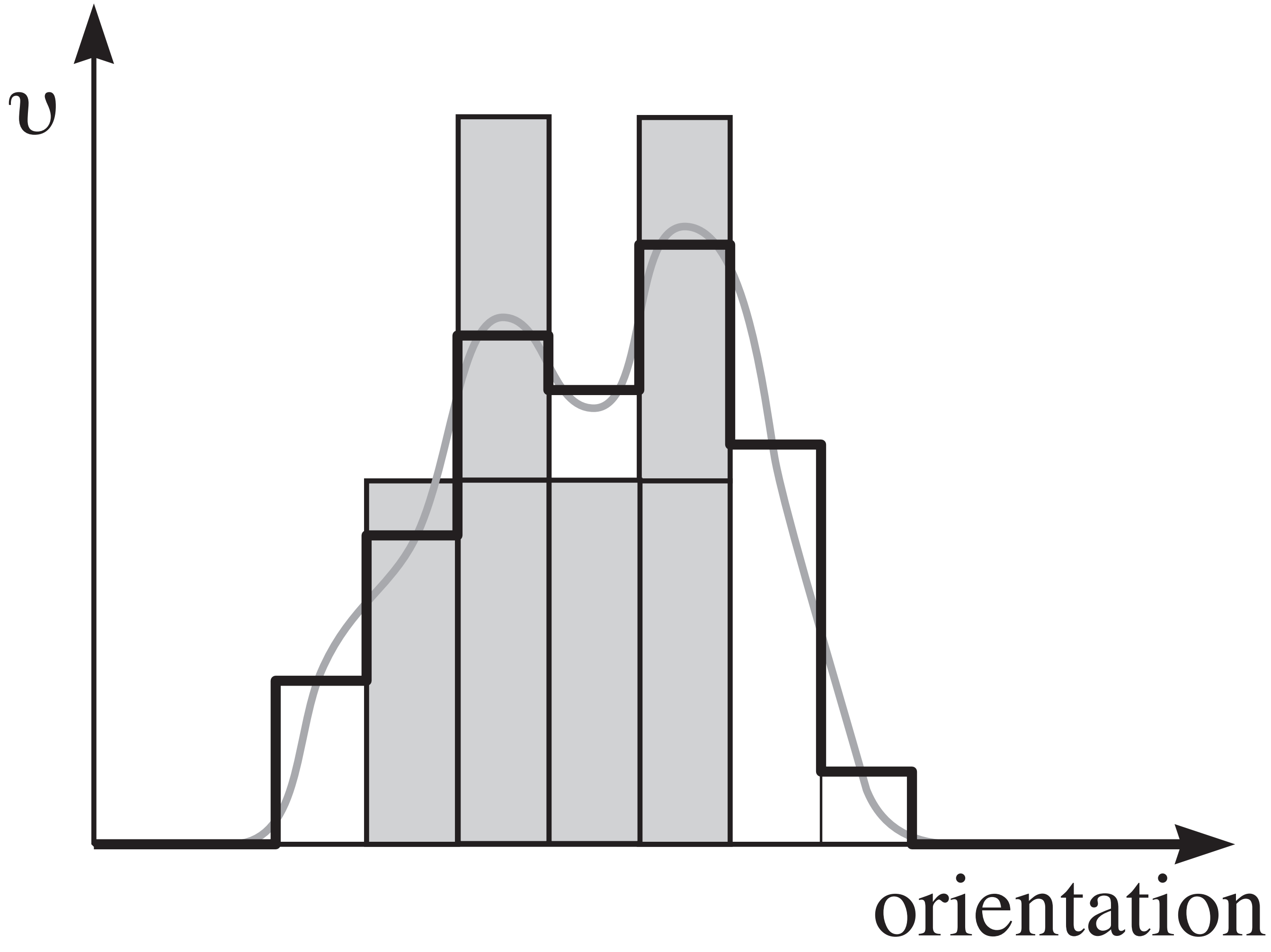


L.S. Tóth, P. Van Houtte, Textures Microstruct. 19 (1992) 229–244.

probabilistic reconstruction

- add randomly selected orientation if random number $r \in [0,1] \leq v_i$

- continue until N^* orientations collected

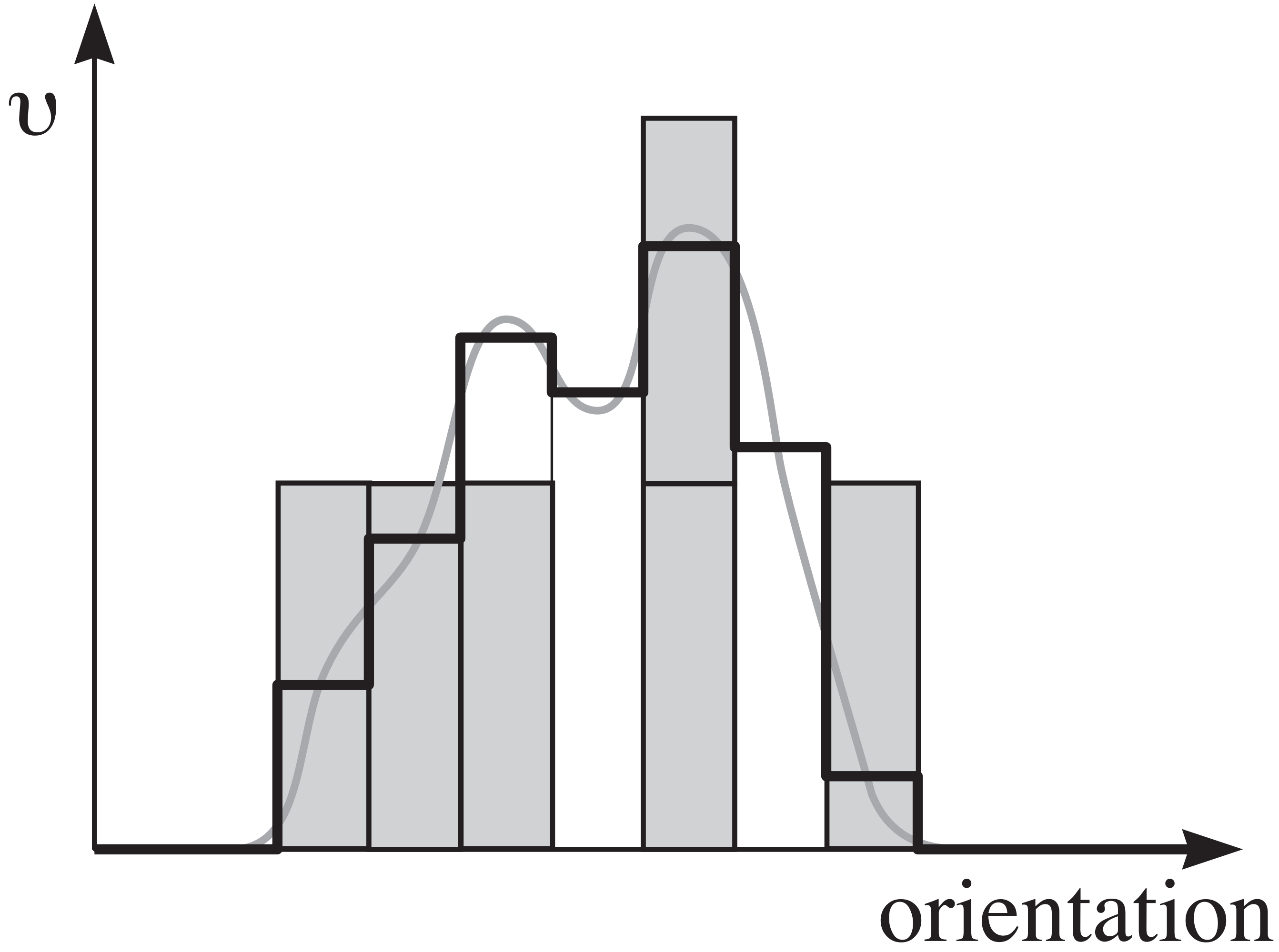


L.S. Tóth, P. Van Houtte, Textures Microstruct. 19 (1992) 229–244.

probabilistic reconstruction

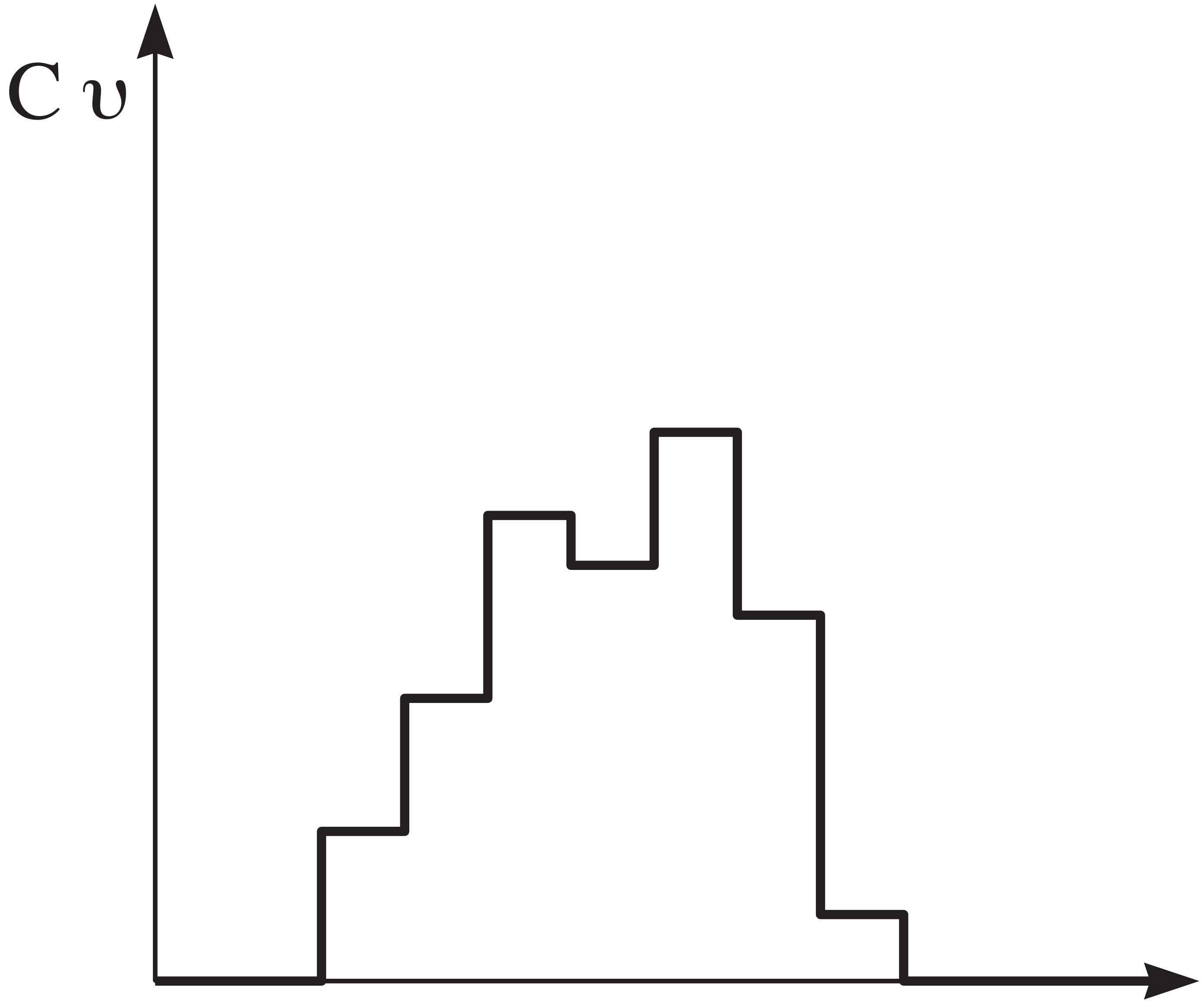
- add randomly selected orientation if random number $r \in [0,1] \leq v_i$

- continue until N^* orientations collected



L.S. Tóth, P. Van Houtte, Textures Microstruct. 19 (1992) 229–244.

deterministic reconstruction

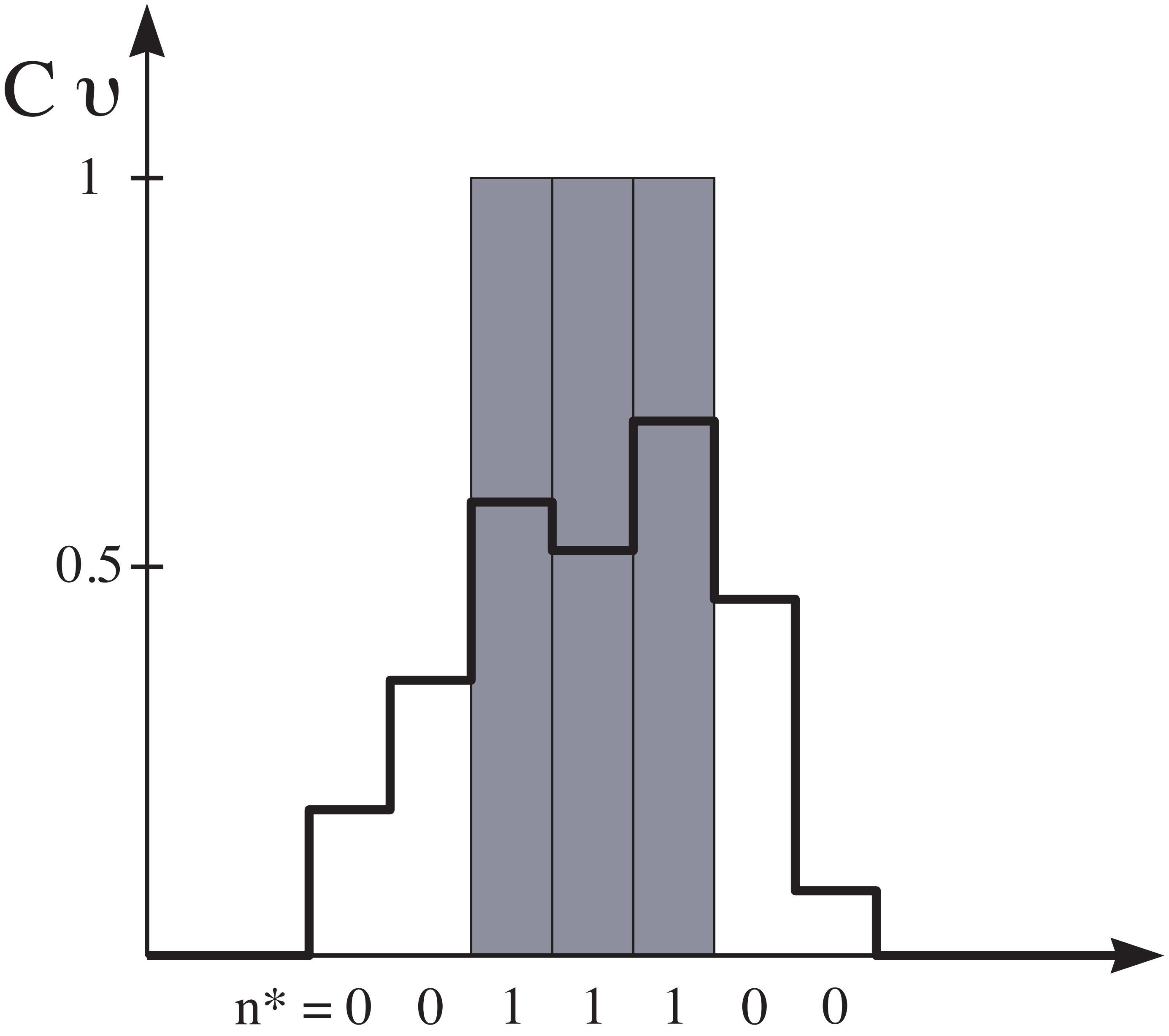


deterministic reconstruction

- select each orientation

$$n_i^* = \mathrm{round}\left(C\, v_i\right)$$

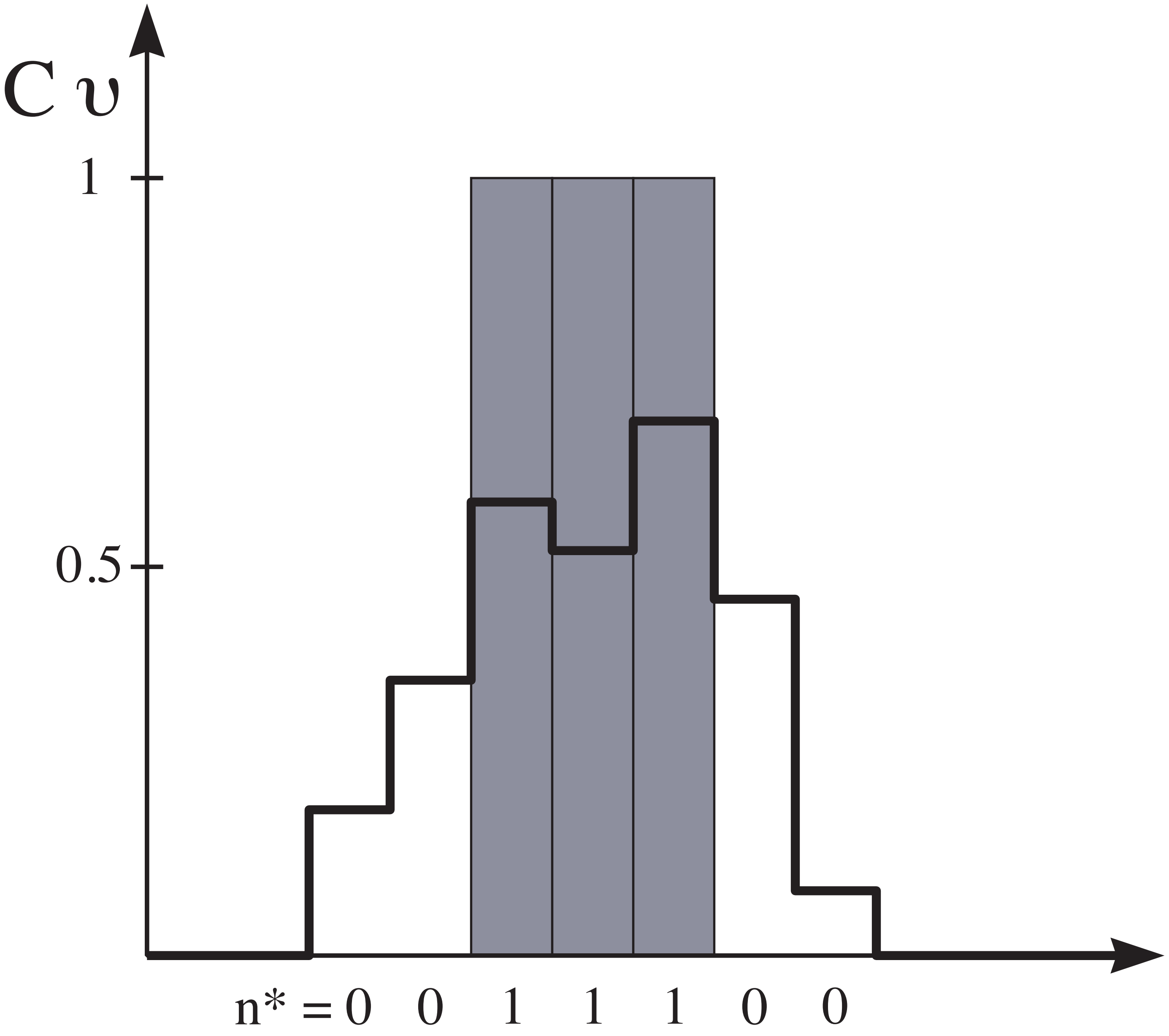
times



deterministic reconstruction

- select each orientation
  $$n_i^* = \mathrm{round}\left(C\, v_i\right)$$
  times

- vary C to ensure
  $$\sum_{i=1}^{N} n_i^* \overset{!}{=} N^*$$



deterministic reconstruction

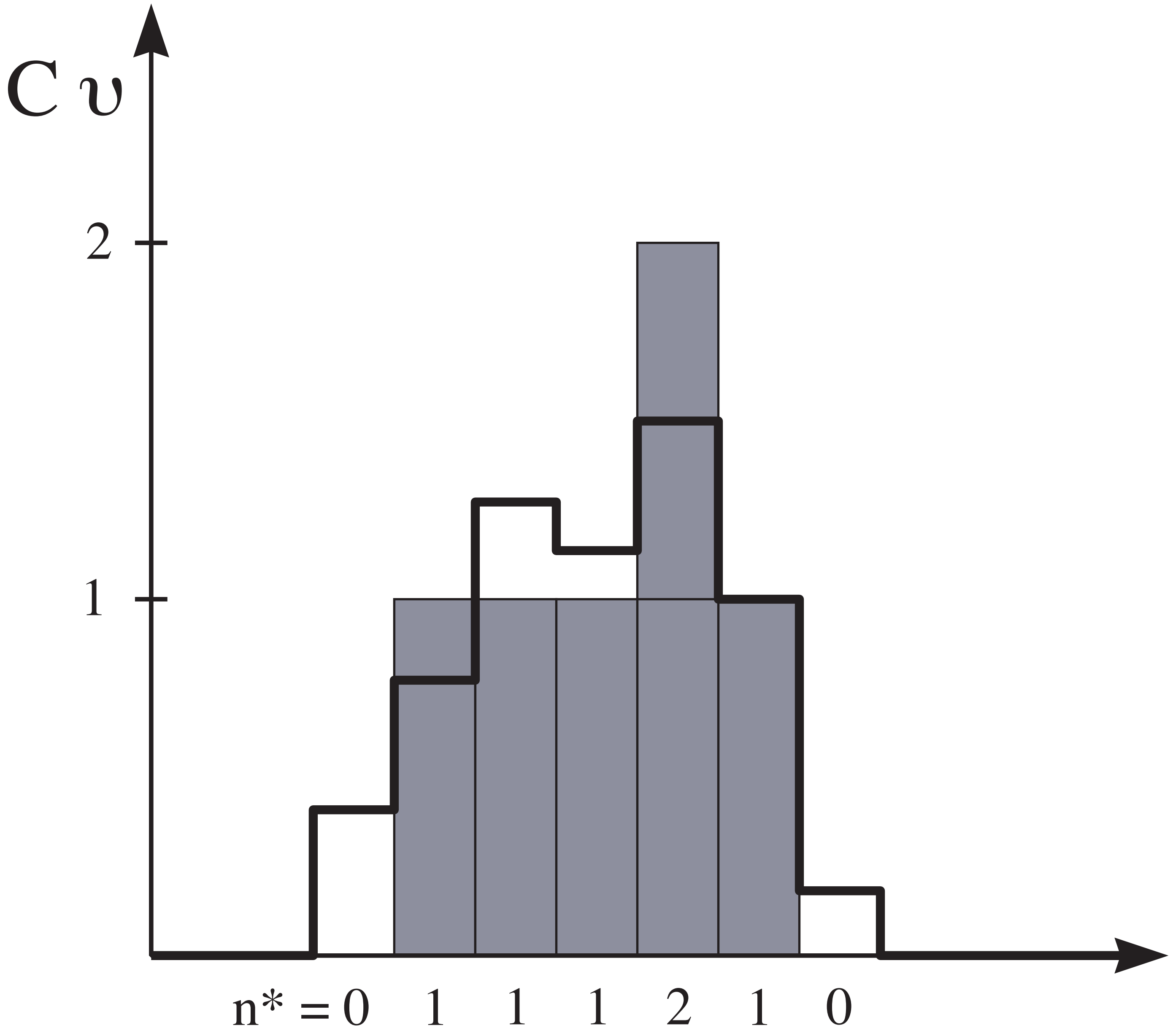
- select each orientation

  $$n_i^* = \mathrm{round}\left(C\, v_i\right)$$

  times

- vary C to ensure

  $$\sum_{i=1}^{N} n_i^* \overset{!}{=} N^*$$



deterministic reconstruction

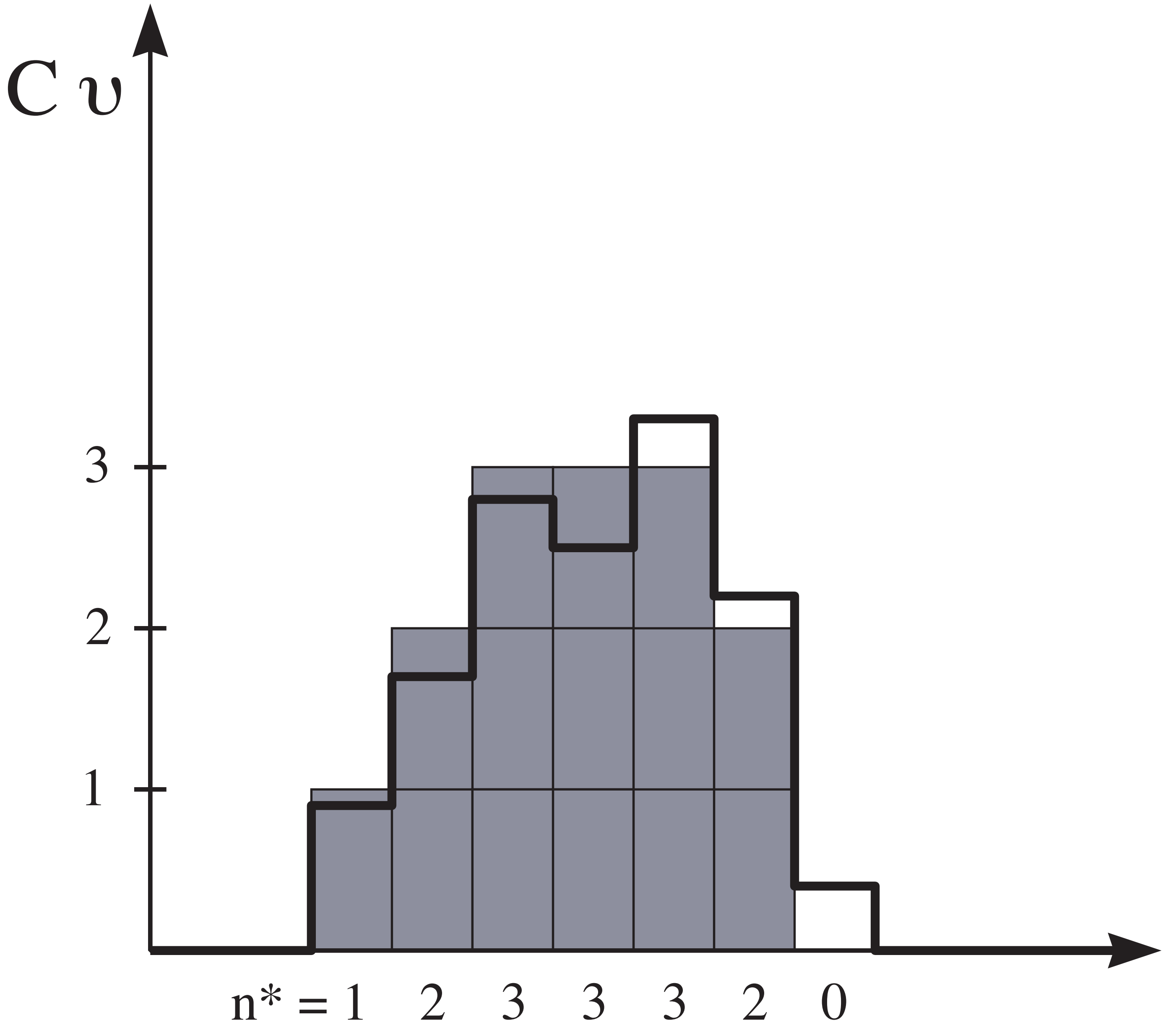
- select each orientation

  $$n_i^* = \mathrm{round}\,(C\,v_i)$$

  times

- vary C to ensure

  $$\sum_{i=1}^{N} n_i^* \overset{!}{=} N^*$$



T. Leffers, D. Juul Jensen, Textures Microstruct. 6 (1986) 231–263.

exemplary textures

strong

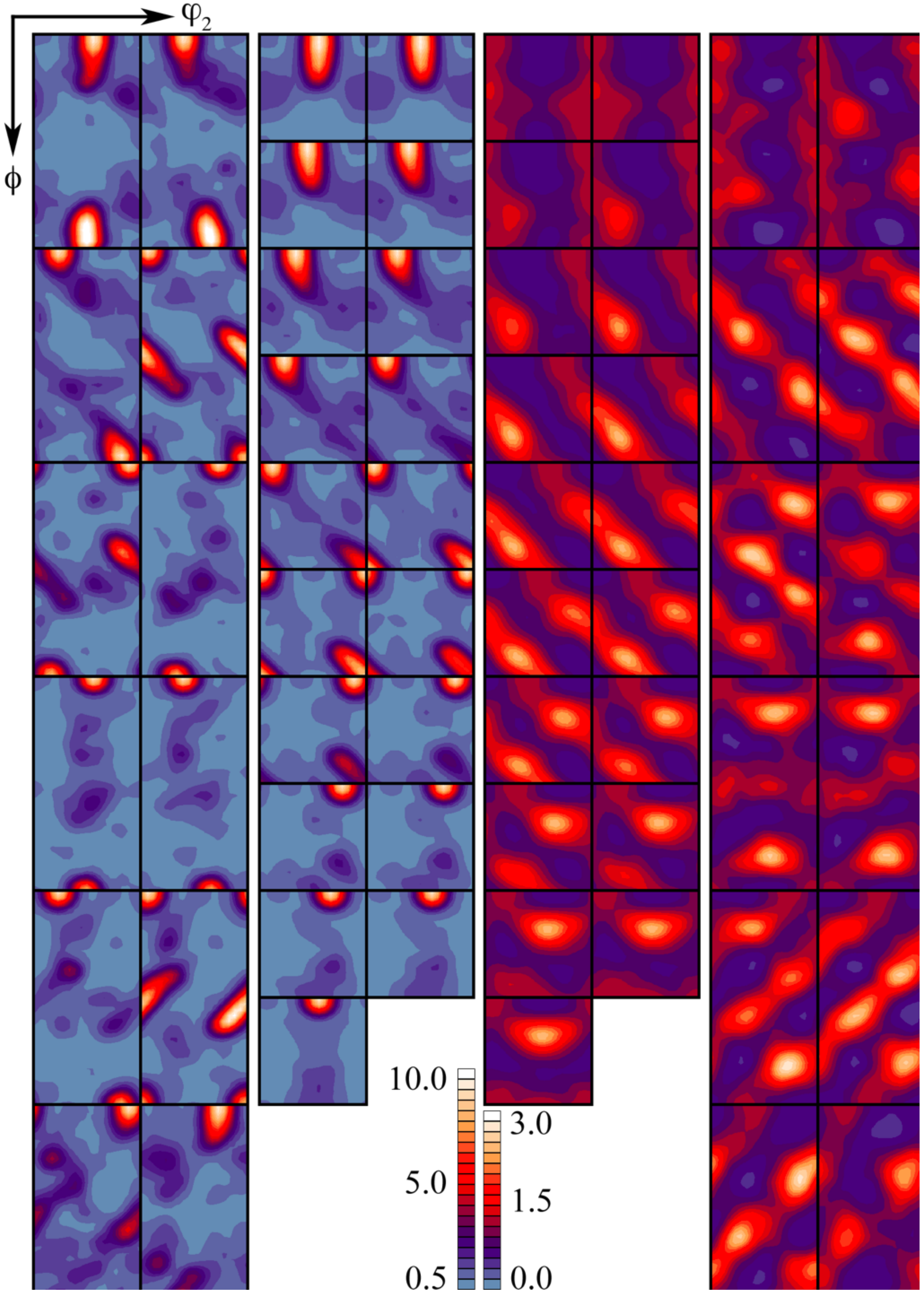
- $N = 6048$ (monoclinic)
- $N = 1512$ (orthotropic)

intermediate

- $N = 6048$
- $N = 1512$

random

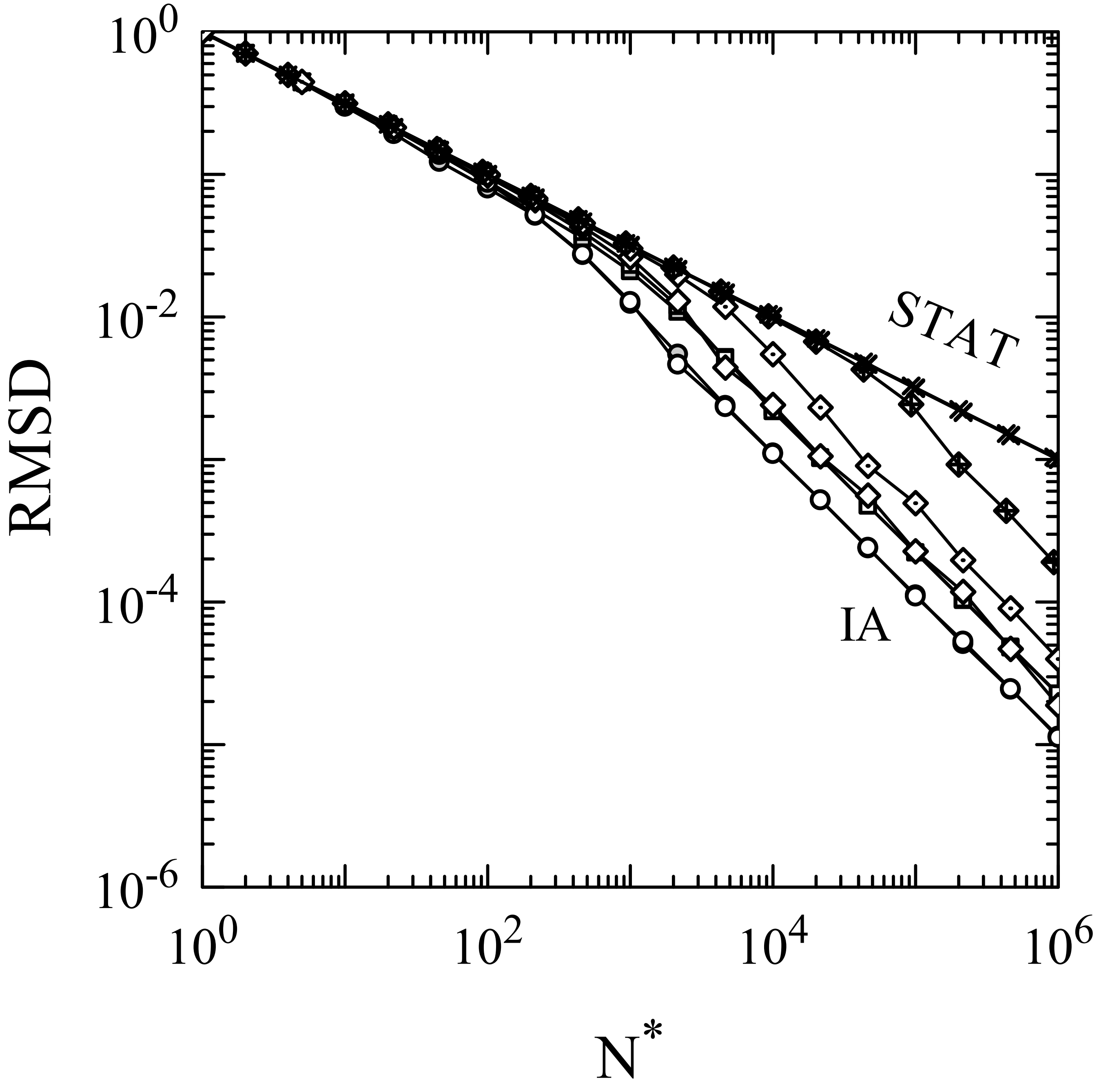
- $N = 186624$
- $N = 27000$
- $N = 5832$



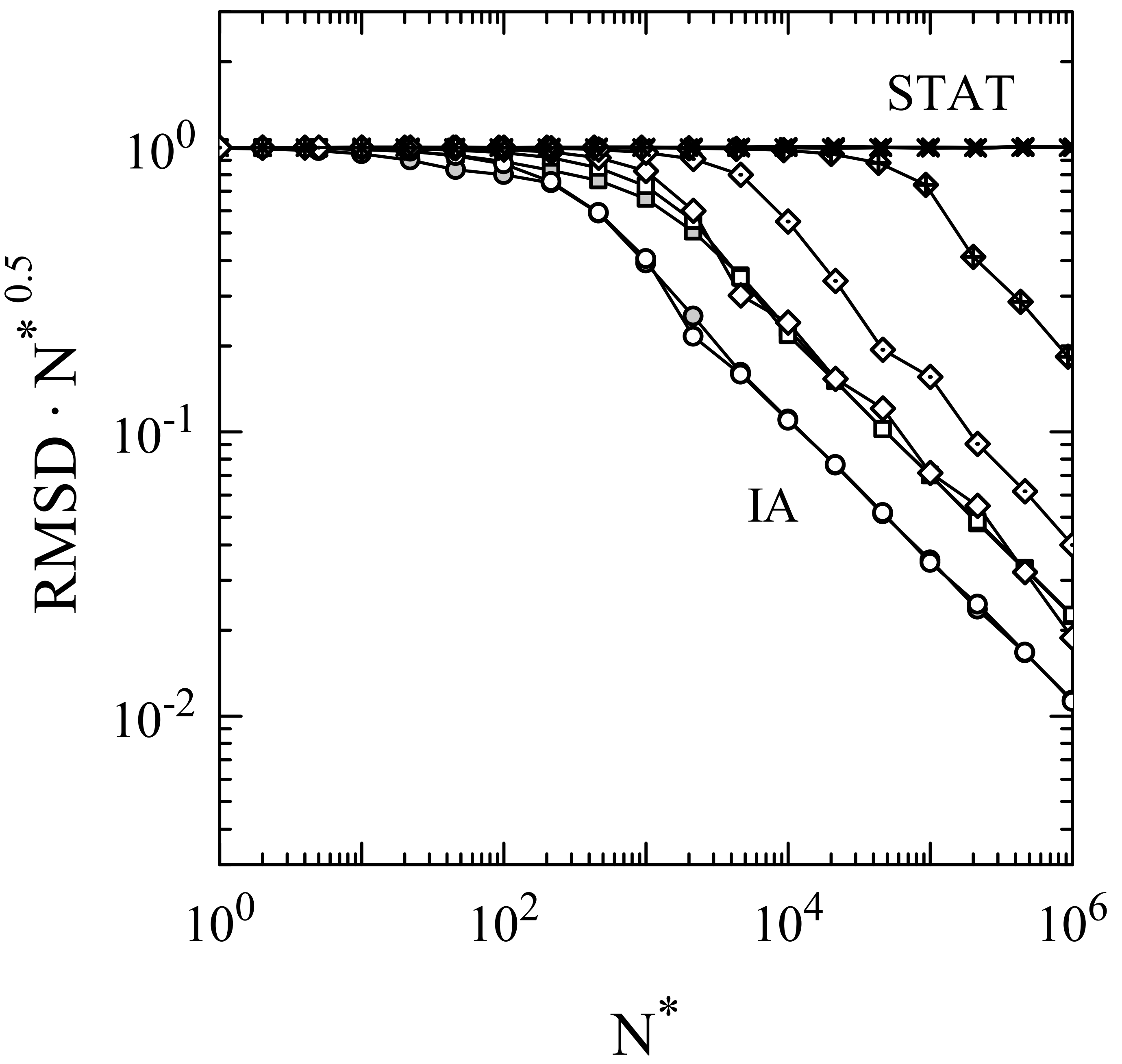
measure of approximation quality

$$\mathrm{RMSD} = \sqrt{\sum_{i=1}^{N} (v_i - v_i^*)^2}$$

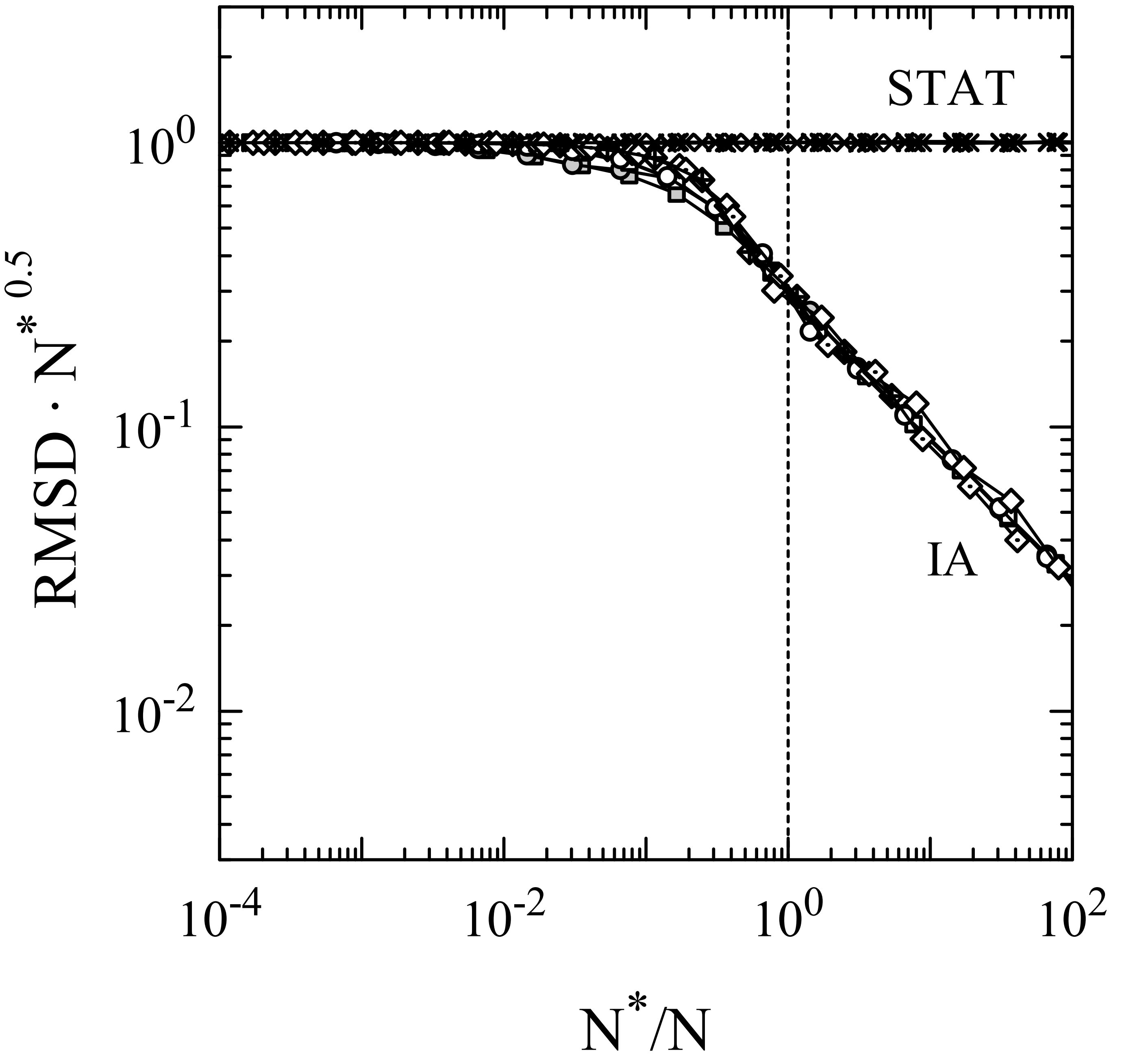
results *(root mean squared deviation)*



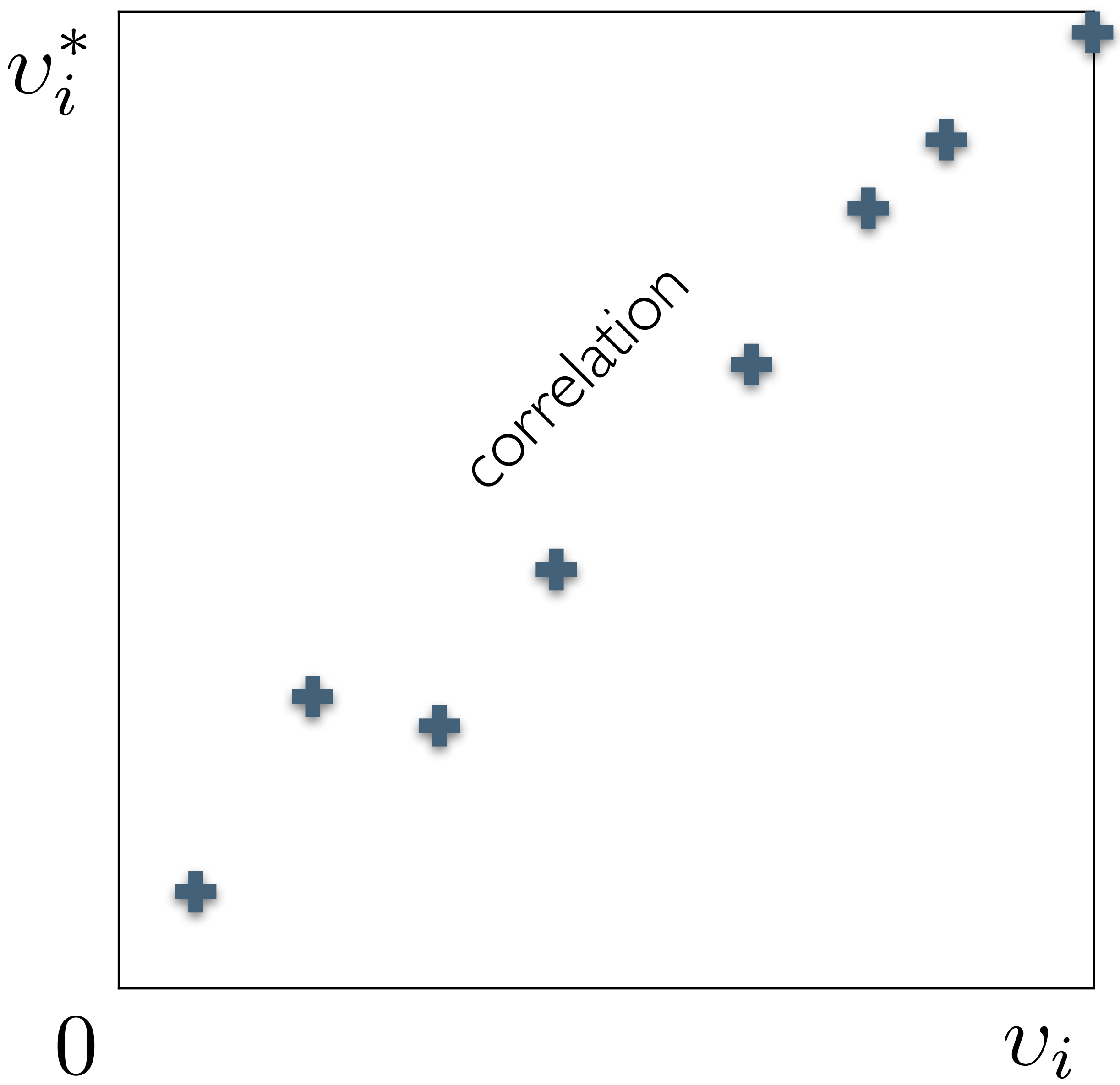
results (root mean squared deviation)



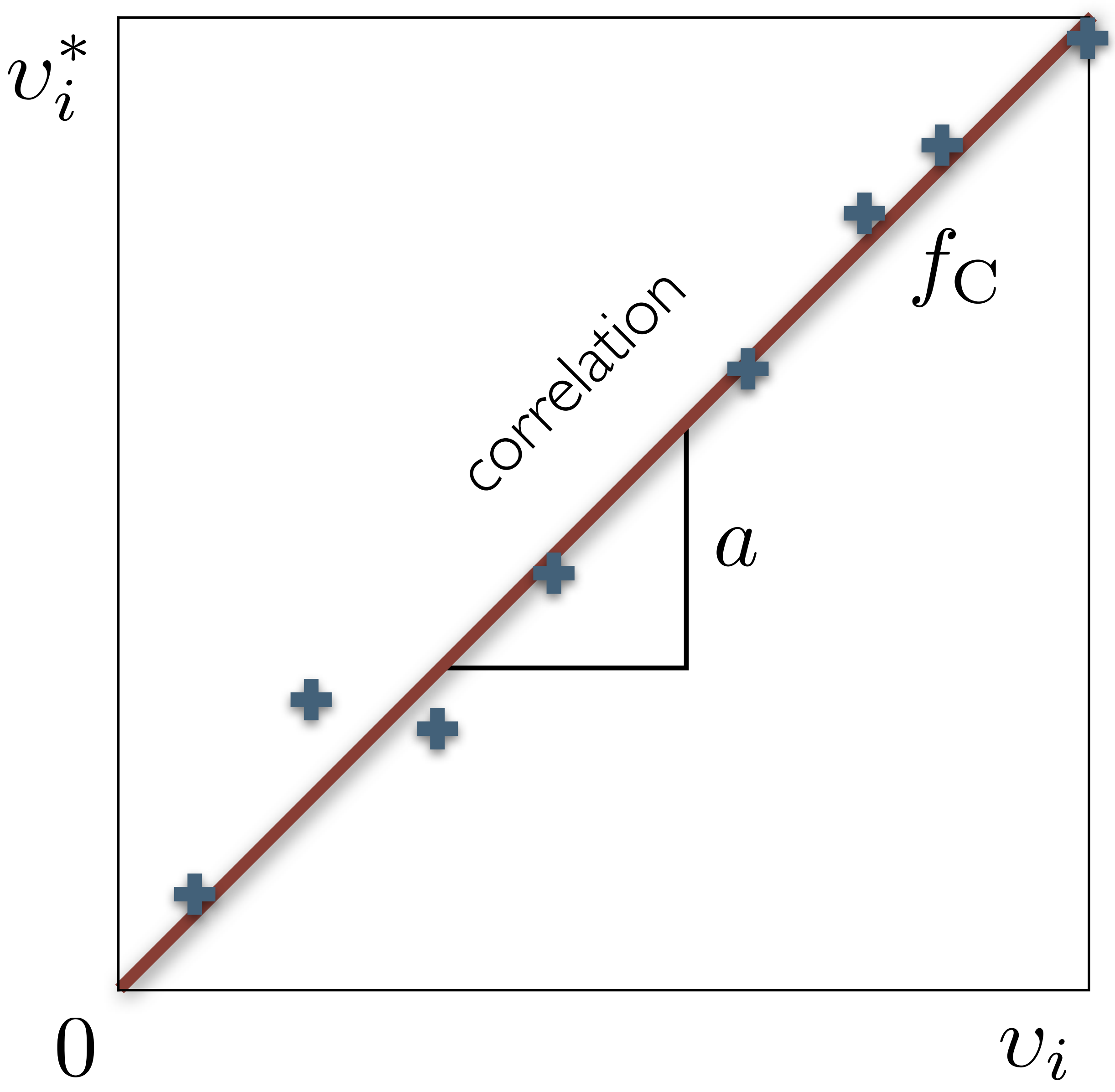
results (root mean squared deviation)



measure of approximation quality



measure of approximation quality

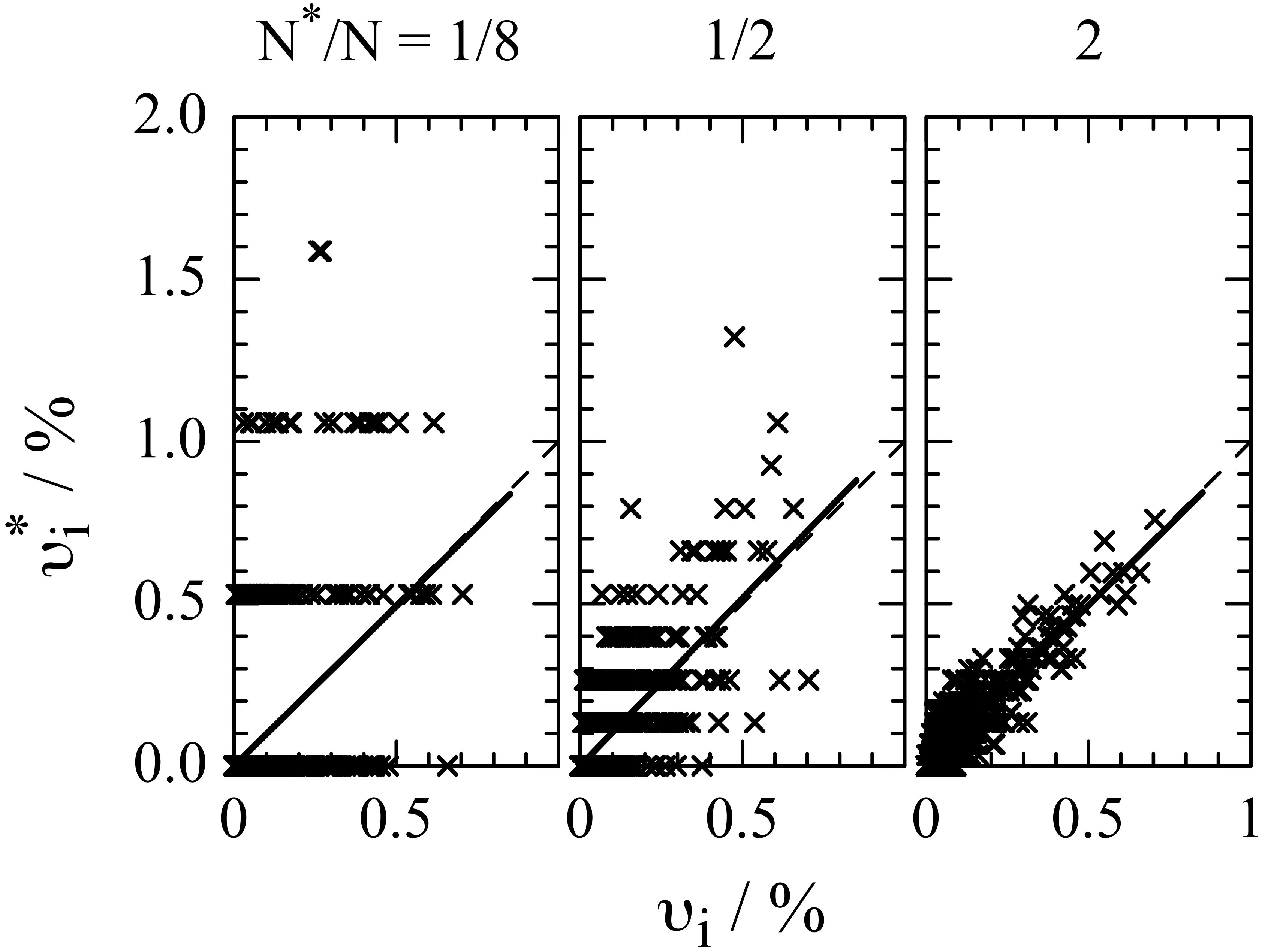


$$a = \frac{N \sum\limits_{i=1}^{N} v_i v_i^* - \sum\limits_{i=1}^{N} v_i \sum\limits_{i=1}^{N} v_i^*}{N \sum\limits_{i=1}^{N} v_i v_i - \sum\limits_{i=1}^{N} v_i \sum\limits_{i=1}^{N} v_i}$$

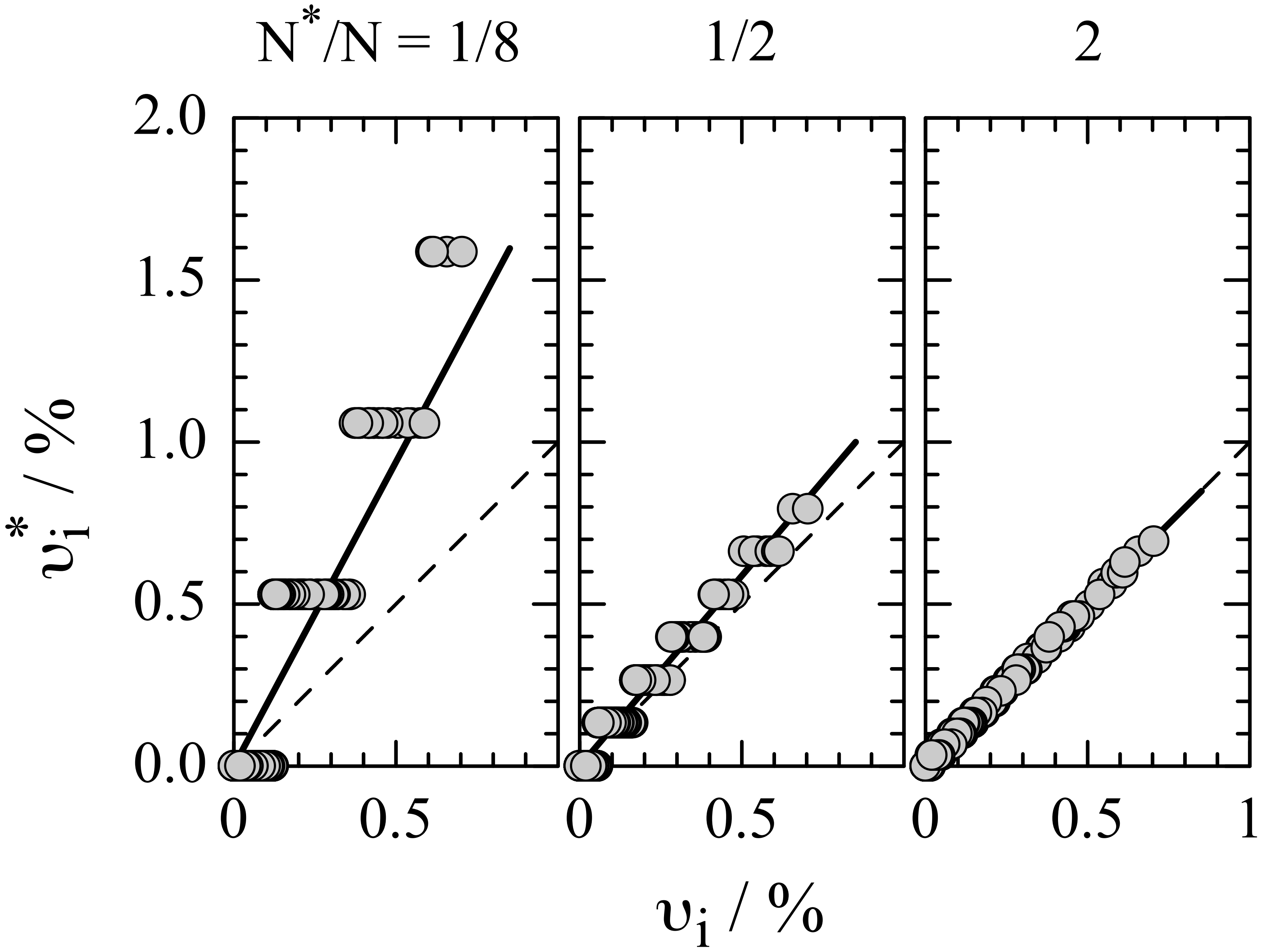
$$f_\mathrm{C} = \frac{\sum\limits_{i=1}^{N} v_i v_i^* / N - \left\langle v_i \right\rangle \left\langle v_i^* \right\rangle}{\sqrt{\left\langle {v_i}^2 \right\rangle - \left\langle v_i \right\rangle^2} \sqrt{\left\langle {v_i^*}^2 \right\rangle - \left\langle v_i^* \right\rangle^2}}$$

J. Tarasiuk and K. Wierzbanowski, Phil. Mag. A 73 (1996) 1083–1091

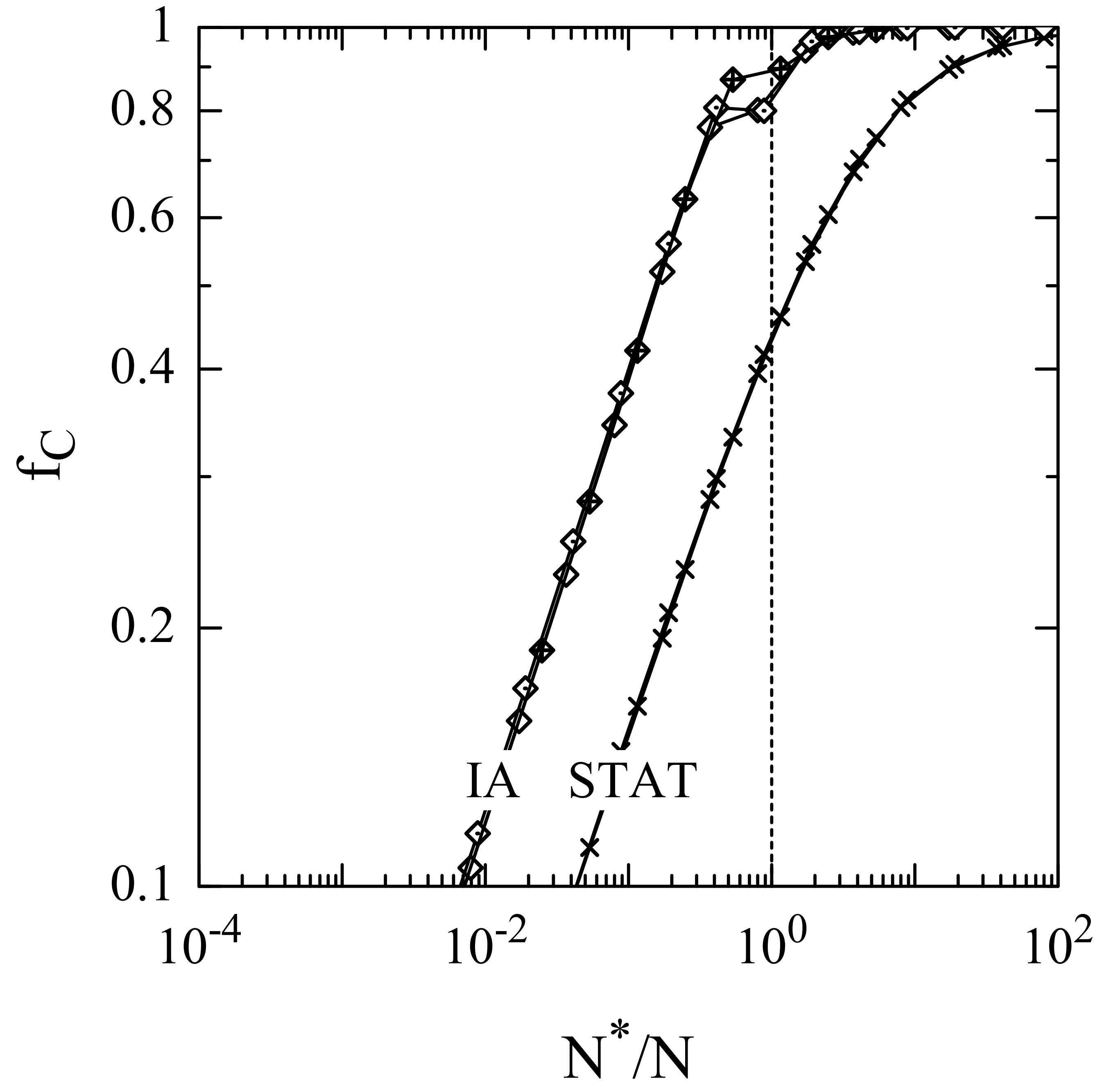
results (correlation for strong texture / probabilistic reconstruction)



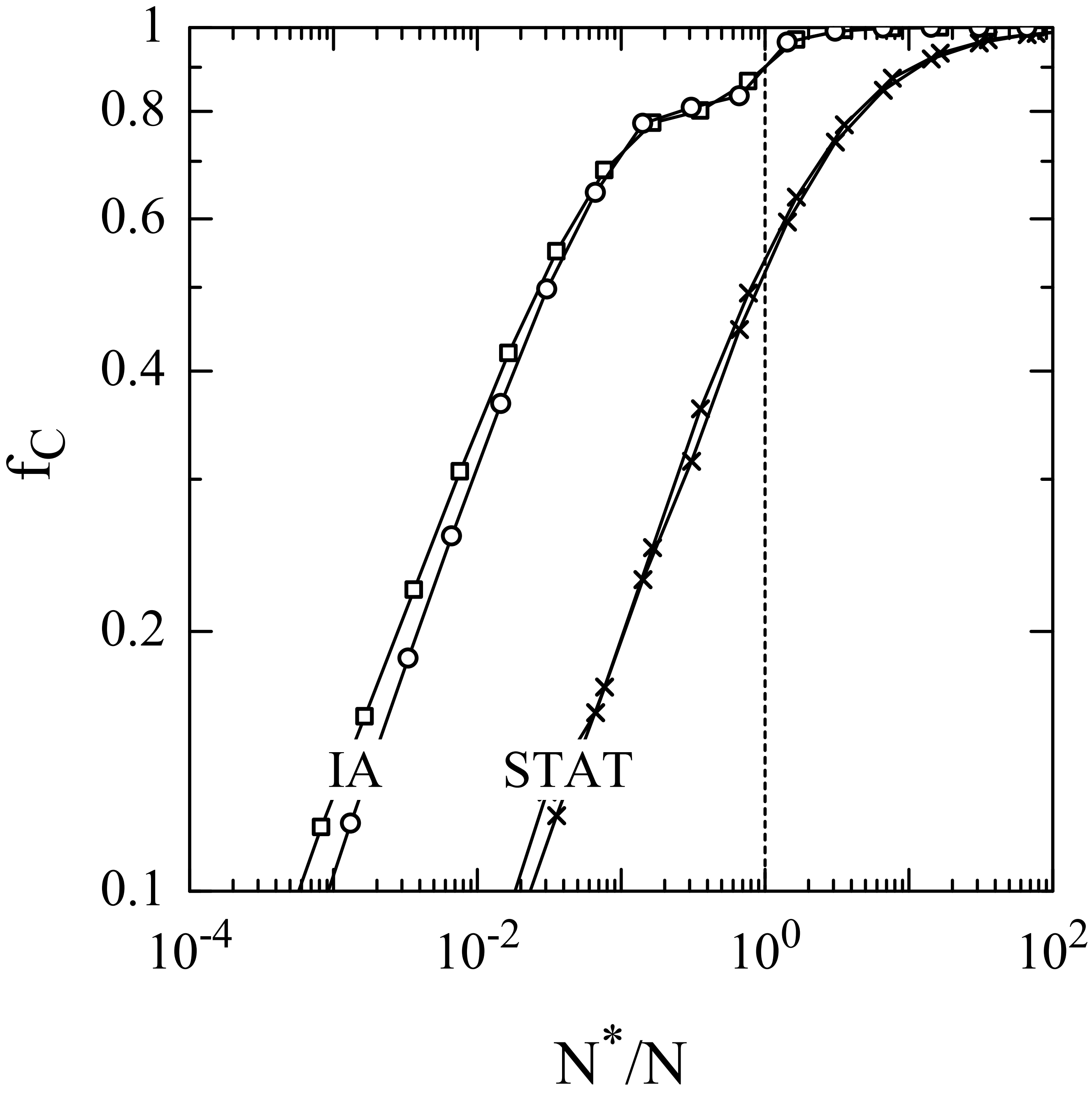
results *(correlation for strong texture / deterministic reconstruction)*



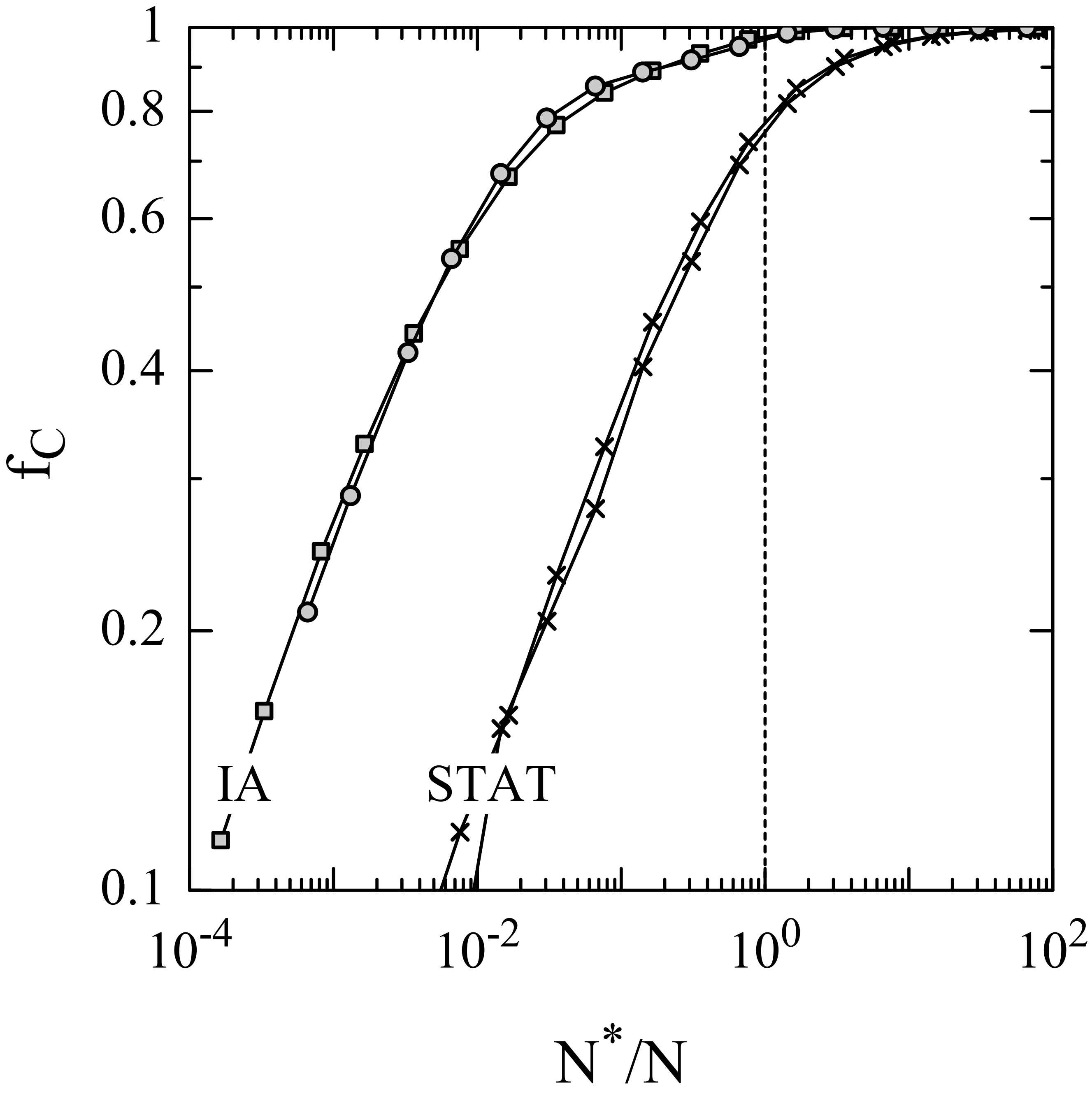
results (correlation factor)



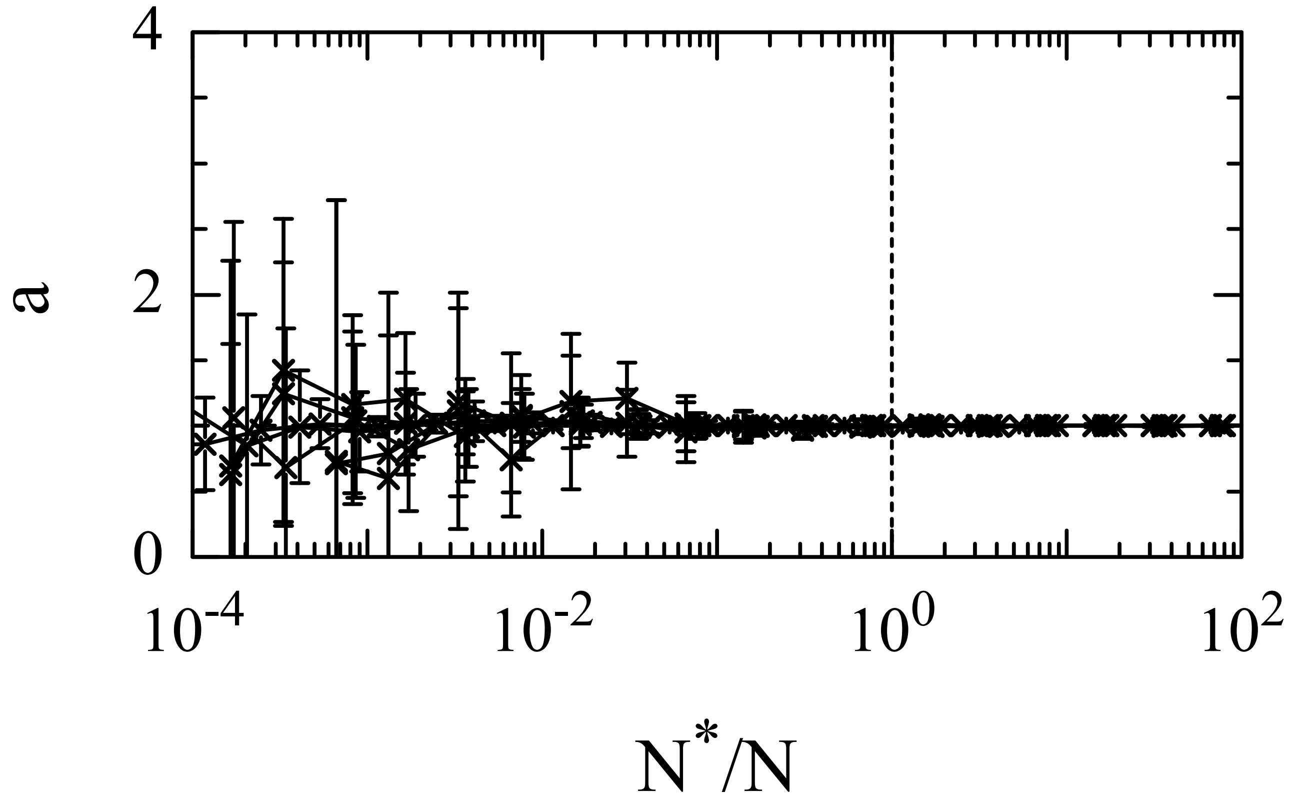
results (correlation factor)



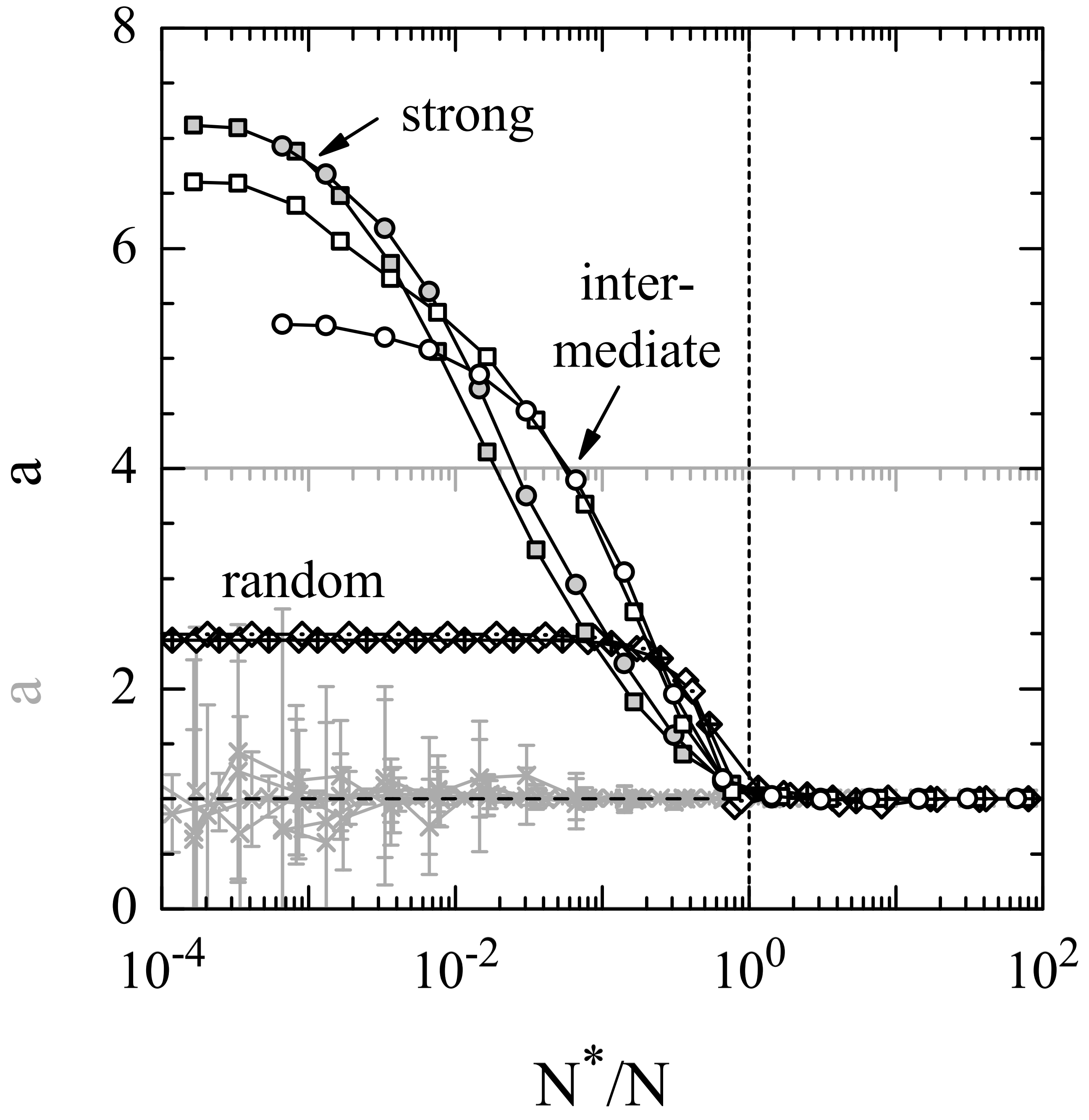
results *(correlation factor)*



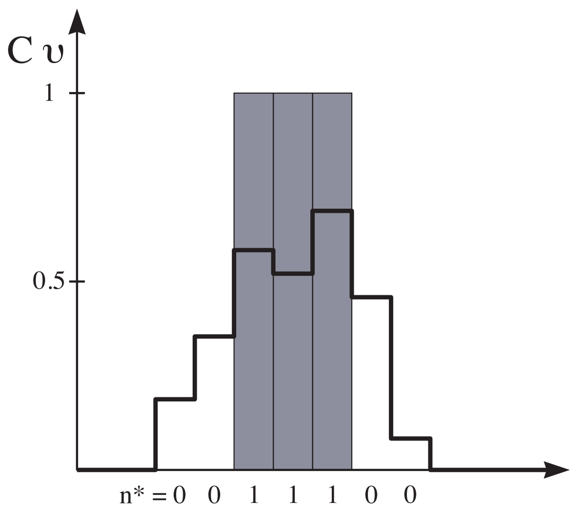
results (correlation slope)

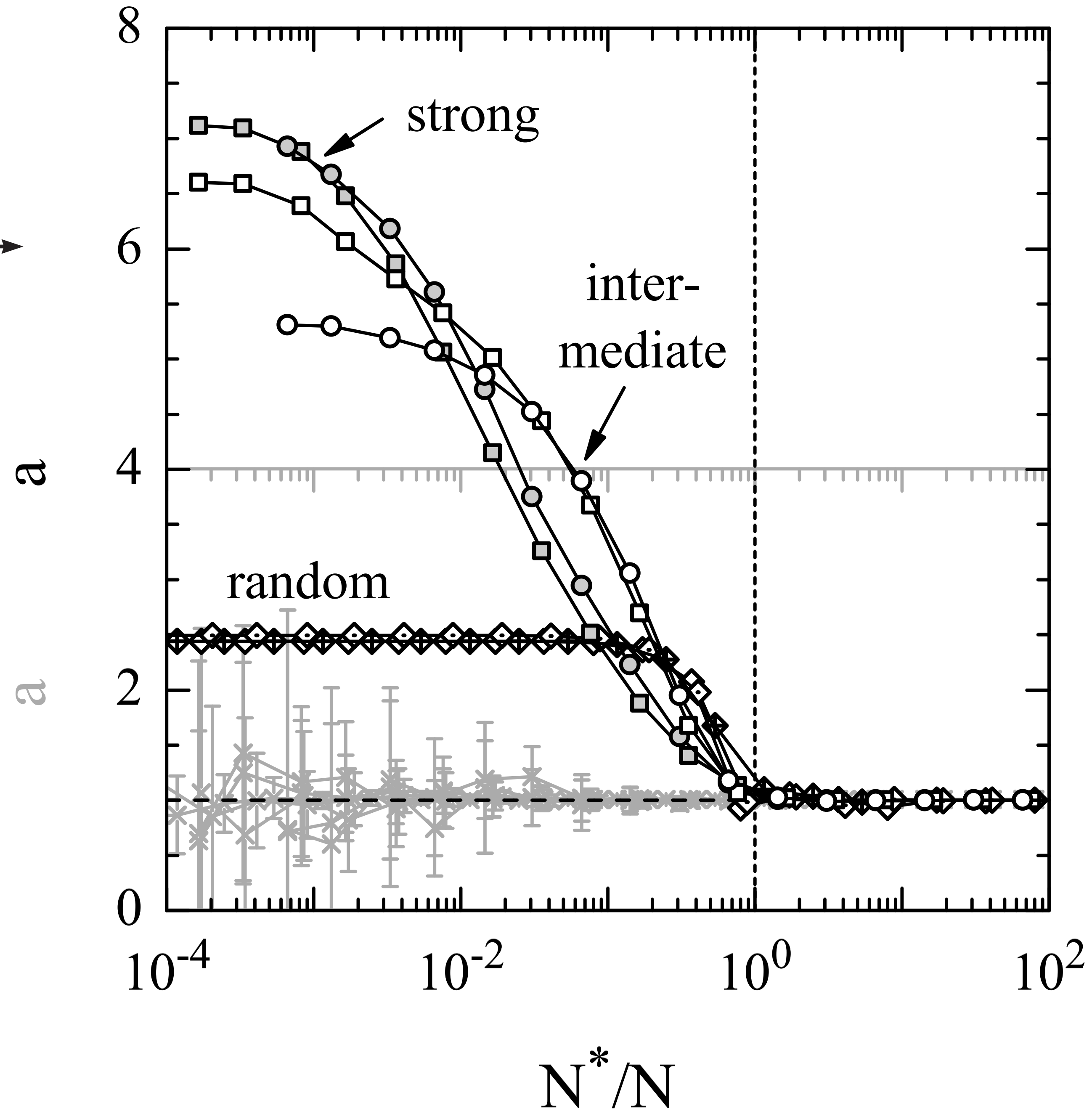


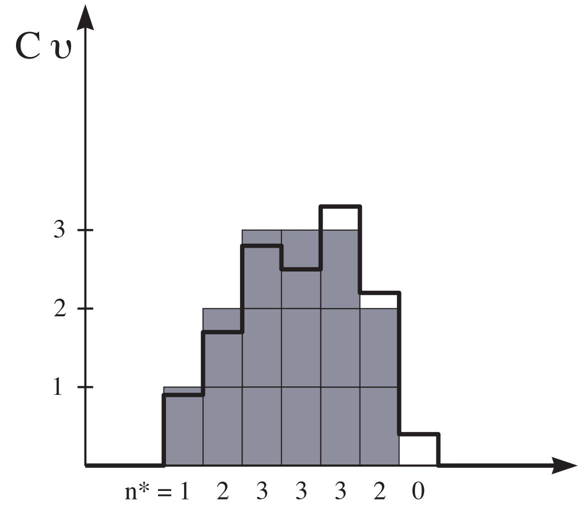
results (correlation slope)



results (correlation slope)







hybrid approach

- select each orientation

  $$n_i^* = \mathrm{round}\,(C\,v_i)$$

  times

- vary C to ensure population size

  $$\sum_{i=1}^{N} n_i^* \overset{!}{=} \max\,(N^*, N)$$

- select N^* orientations from this population at random

hybrid approach

- select each orientation

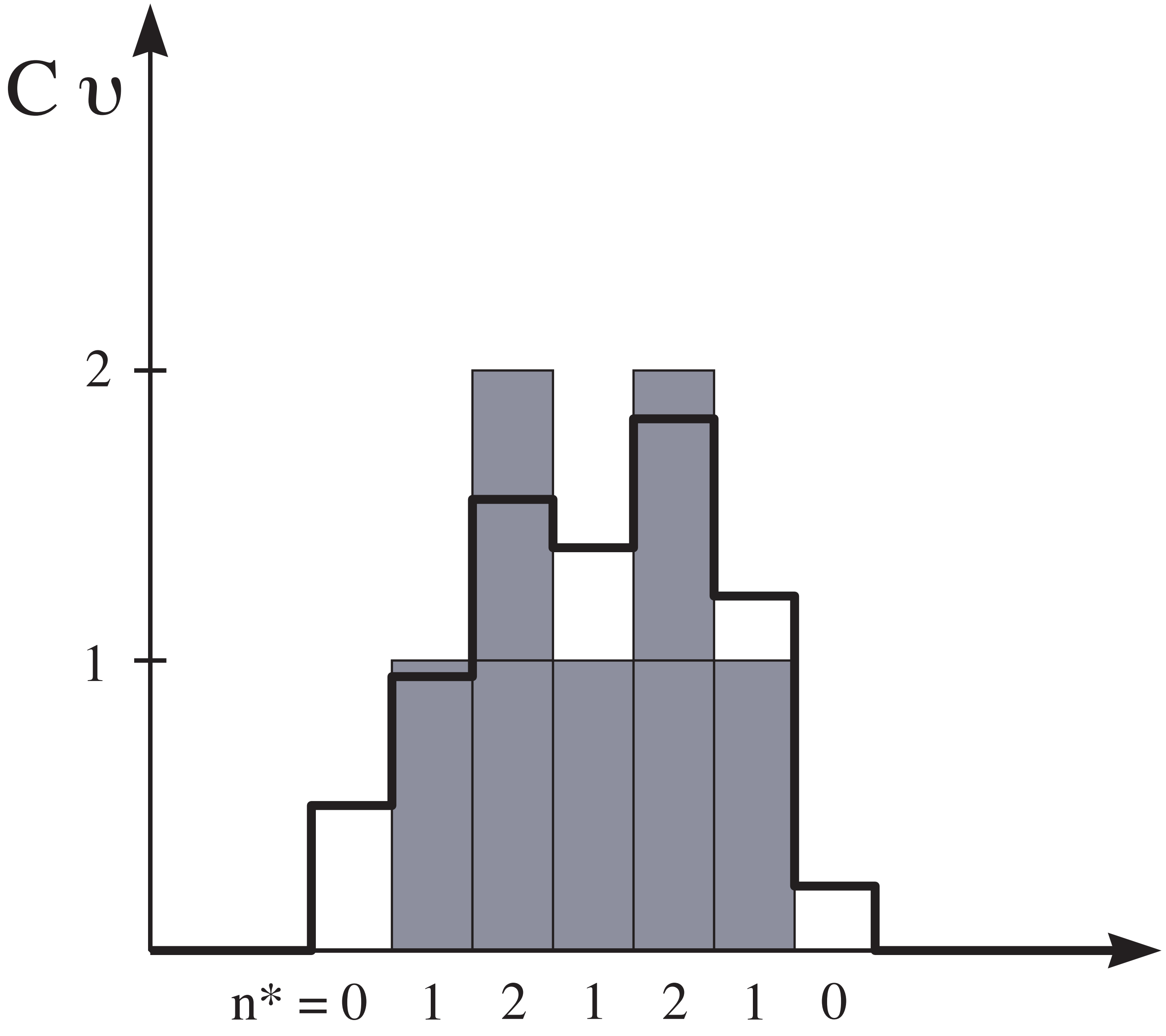
  $$n_i^* = \mathrm{round}\,(C\,v_i)$$

  times

- vary C to ensure population size

  $$\sum_{i=1}^{N} n_i^* \overset{!}{=} \max\,(N^*, N)$$

- select N^* orientations from this population at random



hybrid approach

- select each orientation

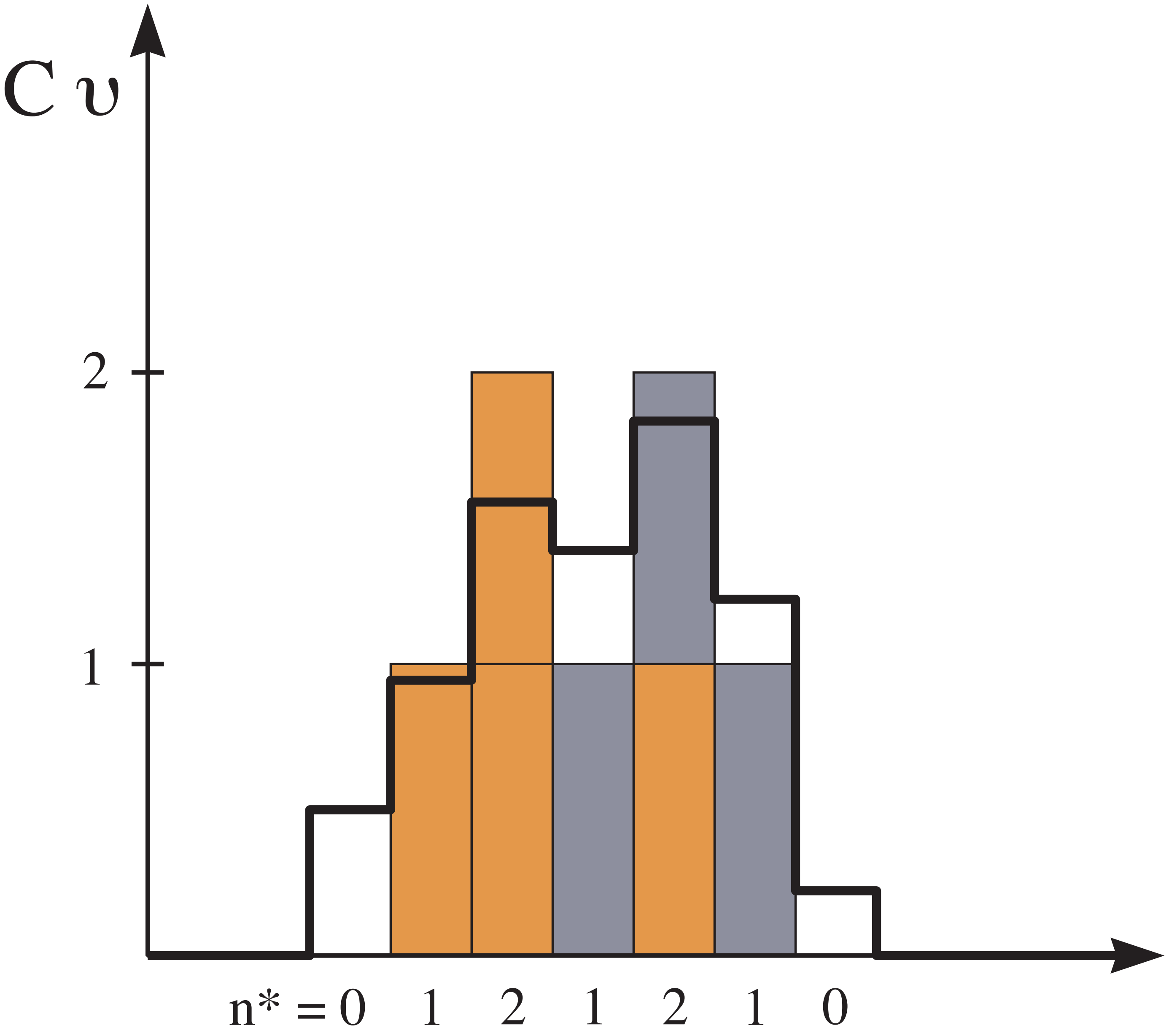
  $$n_i^* = \mathrm{round}\,(C\, v_i)$$

  times

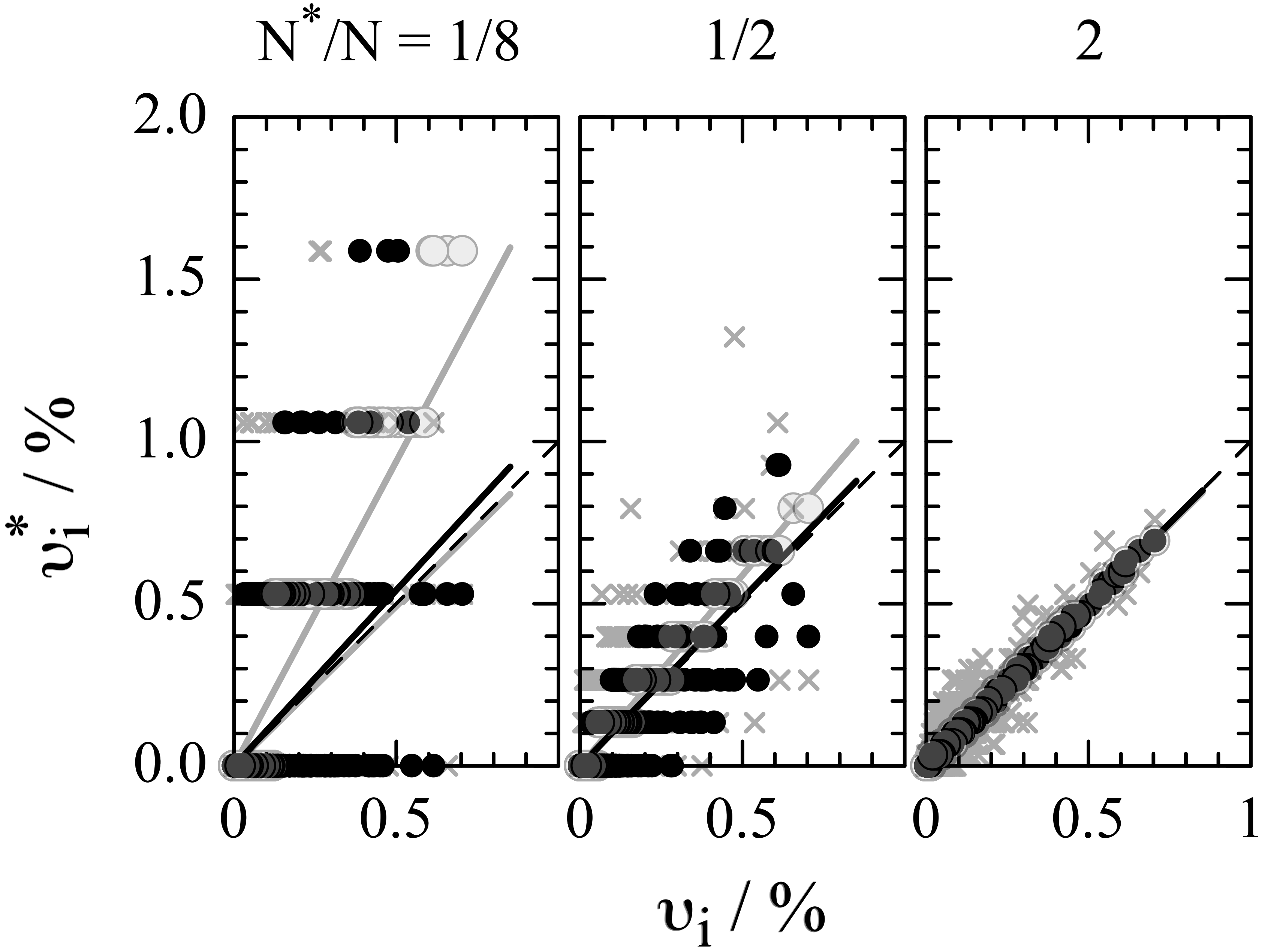
- vary C to ensure population size

  $$\sum_{i=1}^{N} n_i^* \overset{!}{=} \max\,(N^*, N)$$

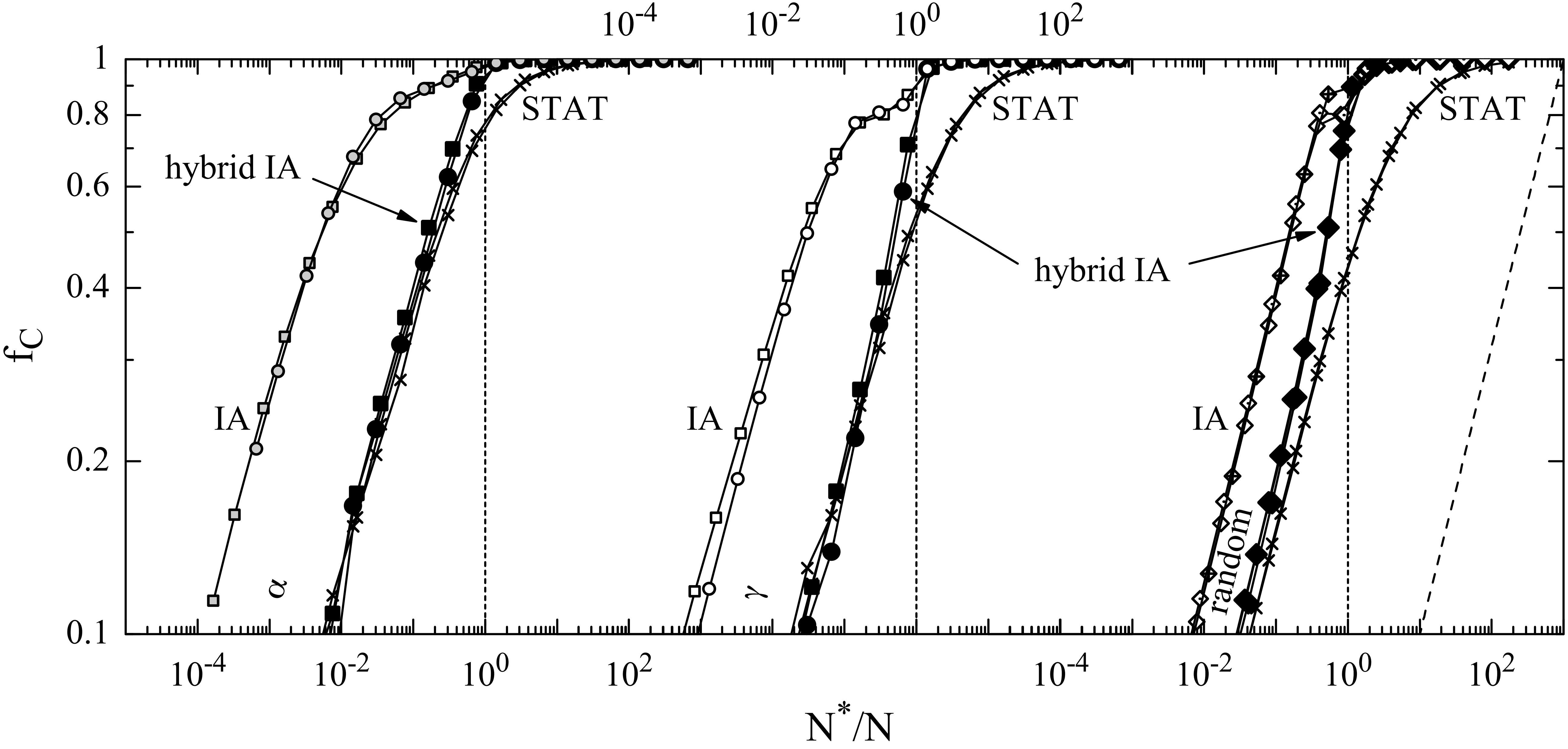
- select N^* orientations from this population at random



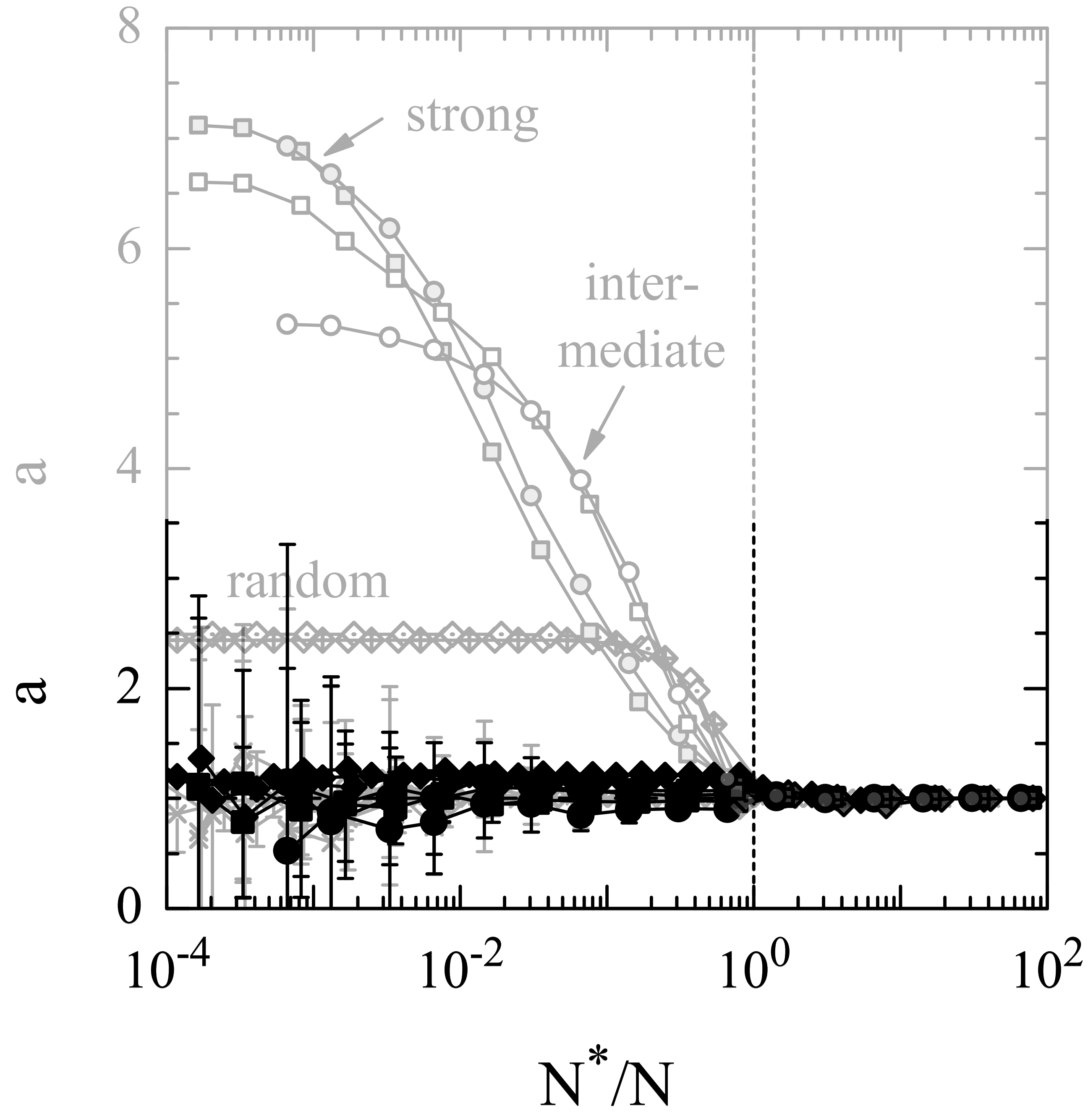
results (correlation for strong texture / hybrid reconstruction)



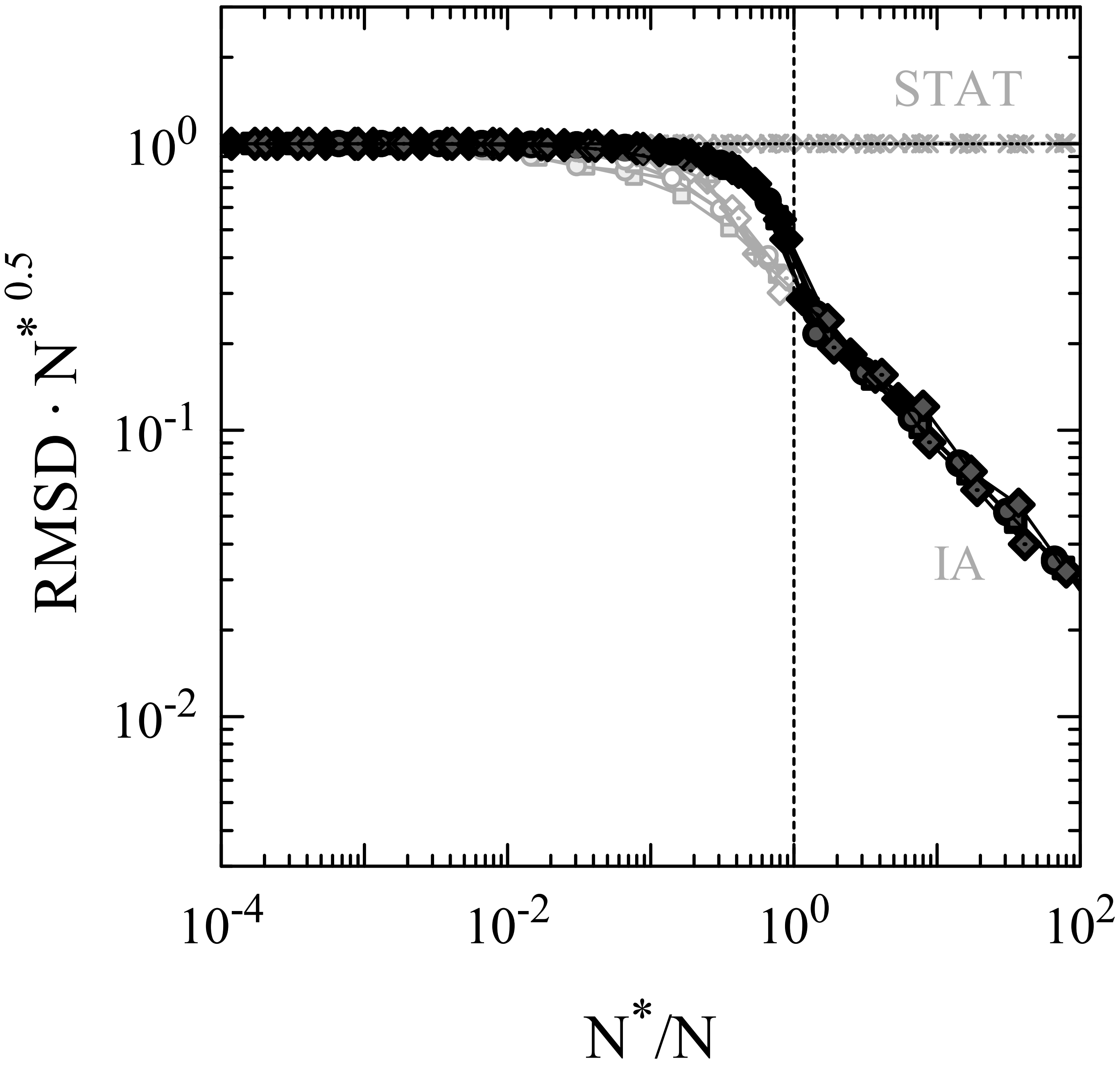
results (correlation factor)



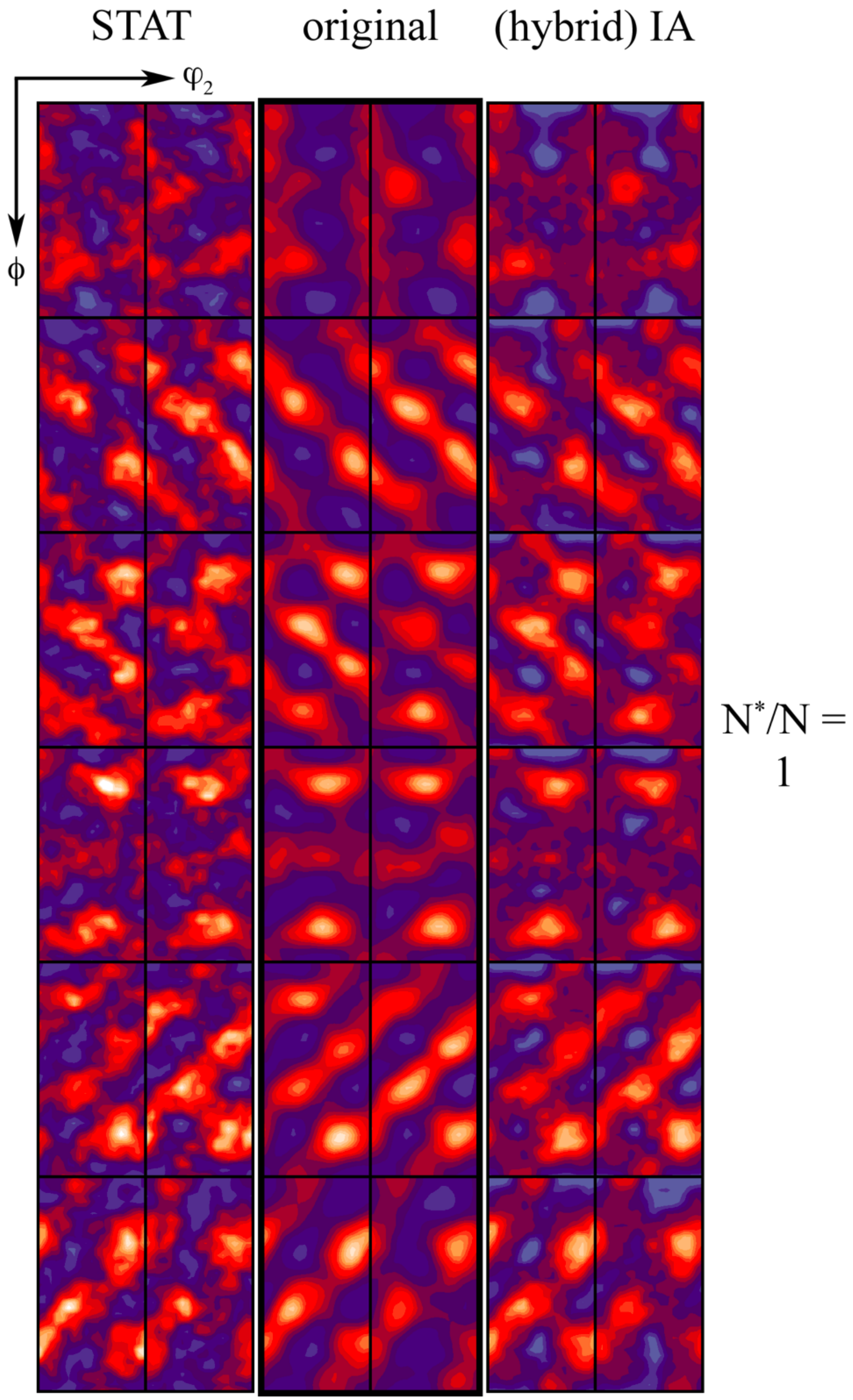
results *(correlation slope)*



results *(root mean squared deviation)*



results *(visual codf)*


STAT
original
(hybrid) IA
φ2
ϕ
N*/N =