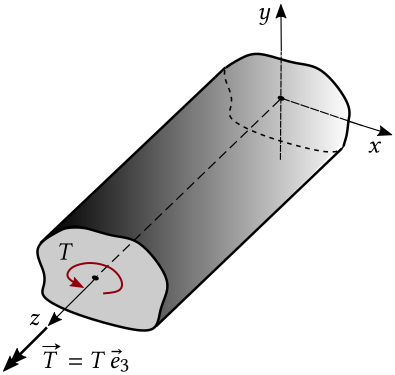
1st International Conference
Mathematical Modelling in Mechanics and Engineering
Mathematical Institute SANU, 8-10 September, 2022.

NUMERICAL COMPUTATIONS OF ELASTIC TORSION USING THE FINITE VOLUME METHOD

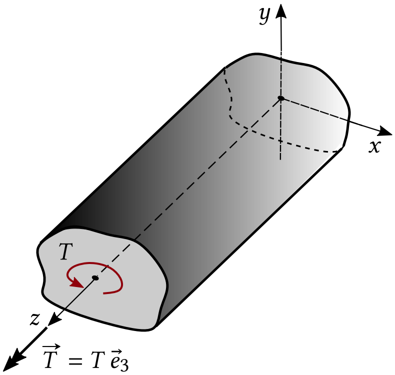
Aleksandar Ćoćić (acocic@mas.bg.ac.rs)

University of Belgrade, Faculty of Mechanical Engineering, Chair for Fluid Mechanics

Cylindrical bar with arbitrary cross-section subjected to end loading moment.

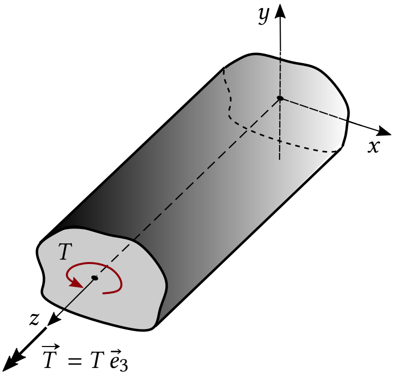


Cylindrical bar with arbitrary cross-section subjected to end loading moment.



Saint-Venant principle: seeking for the solution that satisfies the same resultant loading (not the pointwise traction conditions at the ends).

Cylindrical bar with arbitrary cross-section subjected to end loading moment.

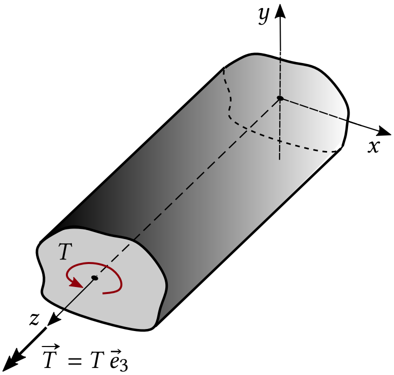


Reasonable assumptions:

Saint-Venant principle: seeking for the solution that satisfies the same resultant loading (not the pointwise traction conditions at the ends).

Physical Considerations: Formulation of the Problem

Cylindrical bar with arbitrary cross-section subjected to end loading moment.

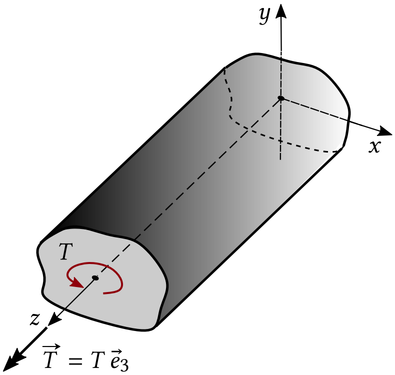


Reasonable assumptions:

- Each cross-section in (x, y)-plane rotates as rigid body around the central axis
- Amount of rotation is linear function of axial coordinate z (small deformation theory)

Saint-Venant principle: seeking for the solution that satisfies the same resultant loading (not the pointwise traction conditions at the ends).

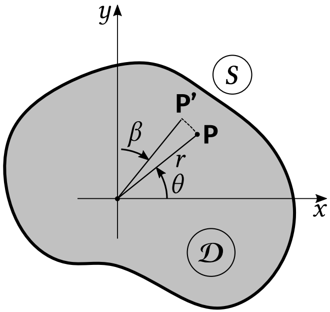
Cylindrical bar with arbitrary cross-section subjected to end loading moment.



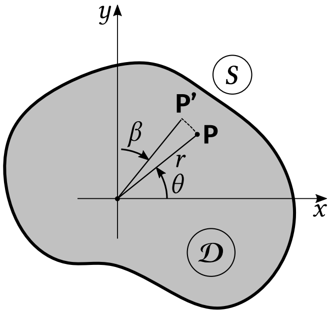
Reasonable assumptions:

- Each cross-section in (x, y)-plane rotates as rigid body around the central axis
- Amount of rotation is linear function of axial coordinate z (small deformation theory)
- Circular cross-sections remain plane after deformation
- Plane cross-sections do not remain plane after deformation - warping displacement

Saint-Venant principle: seeking for the solution that satisfies the same resultant loading (not the pointwise traction conditions at the ends).



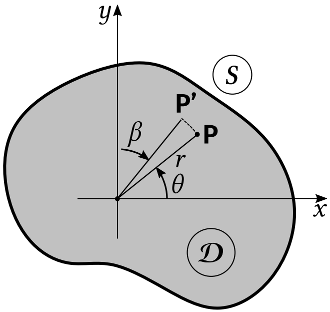
y
S
P'
β
P
r
θ
x
$\mathcal{D}$



- Starting point: "known" displacement field

$$u = -\alpha z y, \quad v = \alpha x z, \quad w = \alpha w^*(x, y)$$

$\alpha = \mathrm{const}$ - angle of twist per unit length

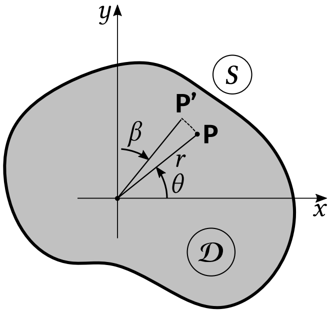


- Starting point: "known" displacement field

$$u = -\alpha z y, \quad v = \alpha x z, \quad w = \alpha w^*(x, y)$$

$\alpha = \mathrm{const}$ - angle of twist per unit length

- Displacement field must satisfy general equations of elasticity theory!

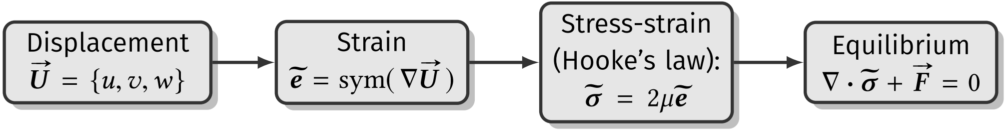


- Starting point: "known" displacement field

  $$u = -\alpha z y, \quad v = \alpha x z, \quad w = \alpha w^*(x, y)$$

  $\alpha = \text{const}$ - angle of twist per unit length

- Displacement field must satisfy general equations of elasticity theory!



Stress field:

$$\sigma_{ii} = 0, \ \tau_{yx} = \tau_{xy} = 0$$

$$\tau_{zx} = \tau_{xz} = \mu\alpha\left(\frac{\partial w^*}{\partial x} - y\right)$$

$$\tau_{zy} = \tau_{yz} = \mu\alpha\left(\frac{\partial w^*}{\partial y} + x\right)$$

Stress field:

$$\sigma_{ii} = 0, \ \tau_{yx} = \tau_{xy} = 0$$

$$\tau_{zx} = \tau_{xz} = \mu\alpha\left(\frac{\partial w^*}{\partial x} - y\right)$$

$$\tau_{zy} = \tau_{yz} = \mu\alpha\left(\frac{\partial w^*}{\partial y} + x\right)$$

Equilibrium ($\vec{F} = 0$)

$$\nabla \cdot \widetilde{\sigma} = 0$$

Stress field:

$$\sigma_{ii} = 0, \ \tau_{yx} = \tau_{xy} = 0$$

$$\tau_{zx} = \tau_{xz} = \mu\alpha\left(\frac{\partial w^*}{\partial x} - y\right)$$

$$\tau_{zy} = \tau_{yz} = \mu\alpha\left(\frac{\partial w^*}{\partial y} + x\right)$$

Equilibrium ($\vec{F} = 0$)

$\nabla \cdot \widetilde{\sigma} = 0$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} = 0$$

Various Formulations of Mathematical Model

Stress field:

$$\sigma_{ii} = 0, \ \tau_{yx} = \tau_{xy} = 0$$

$$\tau_{zx} = \tau_{xz} = \mu\alpha \left(\frac{\partial w^*}{\partial x} - y \right)$$

$$\tau_{zy} = \tau_{yz} = \mu\alpha \left(\frac{\partial w^*}{\partial y} + x \right)$$

Equilibrium ($\vec{F} = 0$)

$$\nabla \cdot \widetilde{\sigma} = 0$$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} = 0$$

Stress field:

$$\sigma_{ii} = 0, \; \tau_{yx} = \tau_{xy} = 0$$

$$\tau_{zx} = \tau_{xz} = \mu\alpha \left(\frac{\partial w^*}{\partial x} - y \right)$$

$$\tau_{zy} = \tau_{yz} = \mu\alpha \left(\frac{\partial w^*}{\partial y} + x \right)$$

Equilibrium ($\vec{F} = 0$)

$\nabla \cdot \widetilde{\sigma} = 0$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} = 0$$

$$\frac{\partial^2 w^*}{\partial x^2} + \frac{\partial^2 w^*}{\partial y^2} = 0$$

Various Formulations of Mathematical Model

Mathematical manipulation

Stress field:

$$\sigma_{ii} = 0, \;\; \tau_{yx} = \tau_{xy} = 0$$

$$\tau_{zx} = \tau_{xz} = \mu\alpha\left(\frac{\partial w^*}{\partial x} - y\right)$$

$$\tau_{zy} = \tau_{yz} = \mu\alpha\left(\frac{\partial w^*}{\partial y} + x\right)$$

Equilibrium ($\vec{F} = 0$)

$$\nabla \cdot \widetilde{\sigma} = 0$$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} = 0$$

$$\frac{\partial^2 w^*}{\partial x^2} + \frac{\partial^2 w^*}{\partial y^2} = 0$$

Various Formulations of Mathematical Model

Stress field:

$$\sigma_{ii} = 0, \quad \tau_{yx} = \tau_{xy} = 0$$

$$\tau_{zx} = \tau_{xz} = \mu\alpha\left(\frac{\partial w^*}{\partial x} - y\right)$$

$$\tau_{zy} = \tau_{yz} = \mu\alpha\left(\frac{\partial w^*}{\partial y} + x\right)$$

Mathematical manipulation

$$\frac{\partial \tau_{zy}}{\partial x} - \frac{\partial \tau_{zx}}{\partial y} = 2\mu\alpha$$

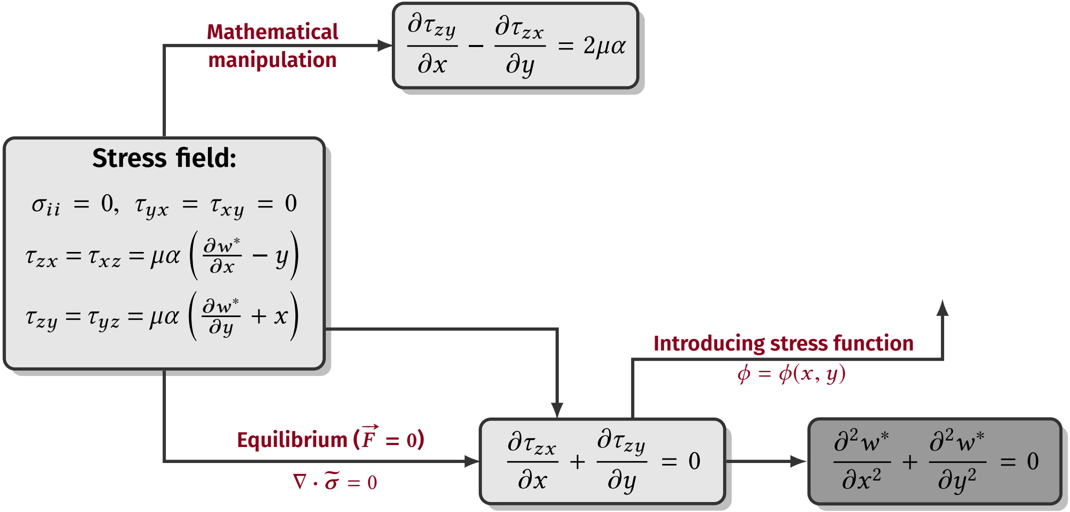
Equilibrium ($\vec{F} = 0$)

$$\nabla \cdot \widetilde{\sigma} = 0$$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} = 0$$

$$\frac{\partial^2 w^*}{\partial x^2} + \frac{\partial^2 w^*}{\partial y^2} = 0$$

Various Formulations of Mathematical Model



Various Formulations of Mathematical Model

Stress field:

$$\sigma_{ii} = 0, \ \tau_{yx} = \tau_{xy} = 0$$

$$\tau_{zx} = \tau_{xz} = \mu\alpha\left(\frac{\partial w^*}{\partial x} - y\right)$$

$$\tau_{zy} = \tau_{yz} = \mu\alpha\left(\frac{\partial w^*}{\partial y} + x\right)$$

Mathematical manipulation

$$\frac{\partial \tau_{zy}}{\partial x} - \frac{\partial \tau_{zx}}{\partial y} = 2\mu\alpha$$

Equilibrium ($\vec{F} = 0$)

$$\nabla \cdot \widetilde{\sigma} = 0$$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} = 0$$

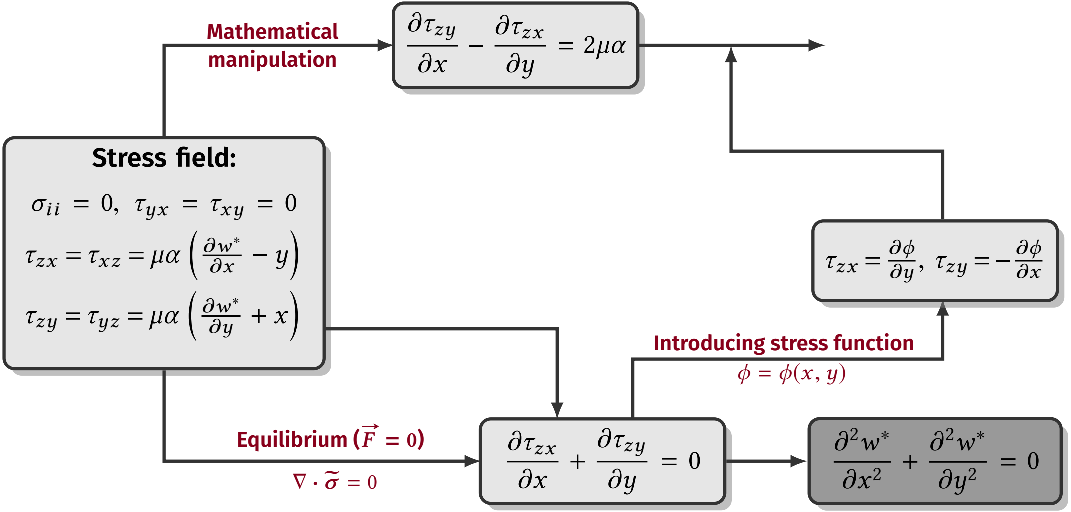
$$\frac{\partial^2 w^*}{\partial x^2} + \frac{\partial^2 w^*}{\partial y^2} = 0$$

Introducing stress function

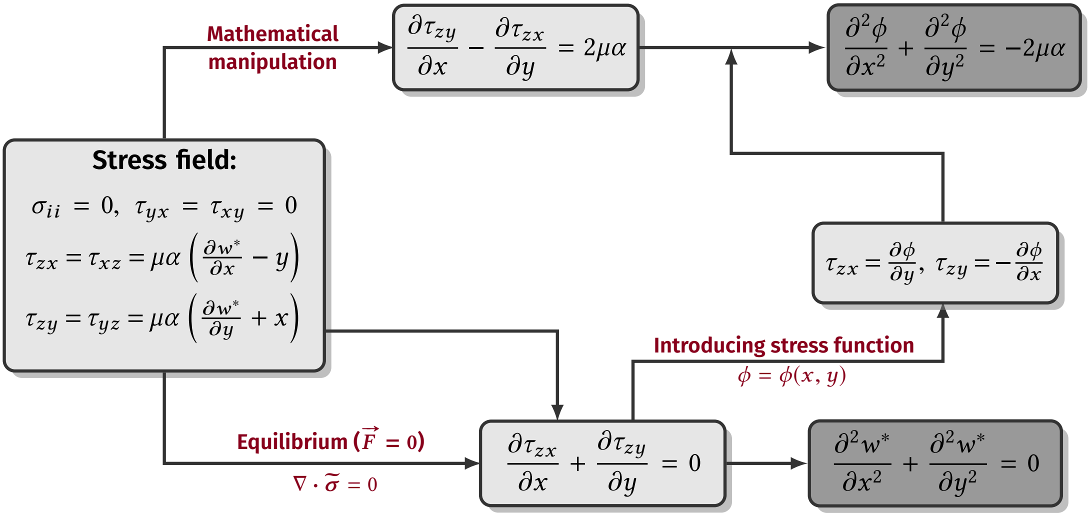
$$\phi = \phi(x, y)$$

$$\tau_{zx} = \frac{\partial \phi}{\partial y}, \ \tau_{zy} = -\frac{\partial \phi}{\partial x}$$

Various Formulations of Mathematical Model



Various Formulations of Mathematical Model



Various Formulations of Mathematical Model

Stress field:

$$\sigma_{ii} = 0, \quad \tau_{yx} = \tau_{xy} = 0$$

$$\tau_{zx} = \tau_{xz} = \mu\alpha\left(\frac{\partial w^*}{\partial x} - y\right)$$

$$\tau_{zy} = \tau_{yz} = \mu\alpha\left(\frac{\partial w^*}{\partial y} + x\right)$$

Mathematical manipulation

$$\frac{\partial \tau_{zy}}{\partial x} - \frac{\partial \tau_{zx}}{\partial y} = 2\mu\alpha$$

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = -2\mu\alpha$$

Traction

$$\vec{t}_n = \vec{n} \cdot \tilde{\sigma}$$

Equilibrium ($\vec{F} = 0$)

$$\nabla \cdot \tilde{\sigma} = 0$$

$$\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} = 0$$

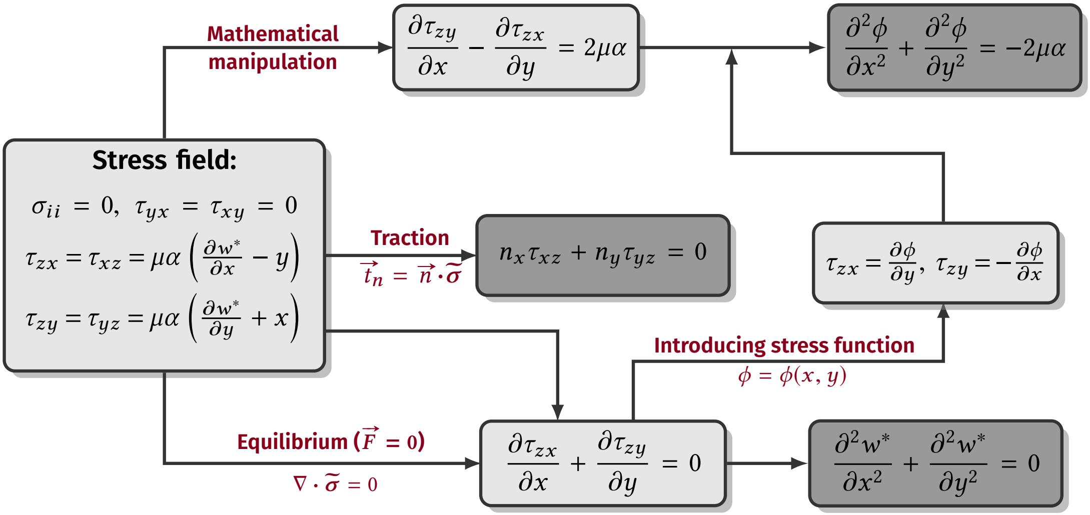
$$\frac{\partial^2 w^*}{\partial x^2} + \frac{\partial^2 w^*}{\partial y^2} = 0$$

Introducing stress function

$$\phi = \phi(x, y)$$

$$\tau_{zx} = \frac{\partial \phi}{\partial y}, \quad \tau_{zy} = -\frac{\partial \phi}{\partial x}$$

Various Formulations of Mathematical Model



Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x$, $\tau_{zy}/(\mu\alpha) = \tau_y$

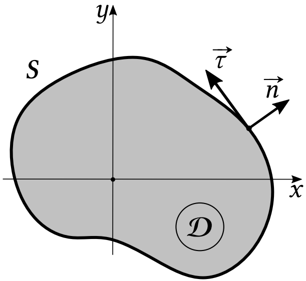
Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$



Simple-connected domain

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x$, $\tau_{zy}/(\mu\alpha) = \tau_y$

Elliptic type

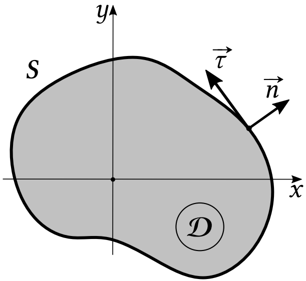
Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$

$w^* = w^*(x, y)$



Simple-connected domain

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x$, $\tau_{zy}/(\mu\alpha) = \tau_y$

Elliptic type

Laplace's PDE:

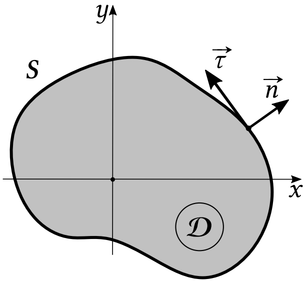
$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$

$w^* = w^*(x, y)$ →

$$\tau_x = \frac{\partial w^*}{\partial x} - y, \quad \tau_y = \frac{\partial w^*}{\partial y} + x$$



Simple-connected domain

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x$, $\tau_{zy}/(\mu\alpha) = \tau_y$

Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

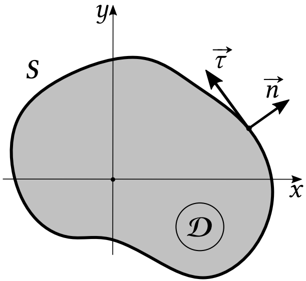
$w^* = w^*(x, y)$ →

$$\tau_x = \frac{\partial w^*}{\partial x} - y, \quad \tau_y = \frac{\partial w^*}{\partial y} + x$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$

$\phi = \phi(x, y)$ →



Simple-connected domain

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x$, $\tau_{zy}/(\mu\alpha) = \tau_y$

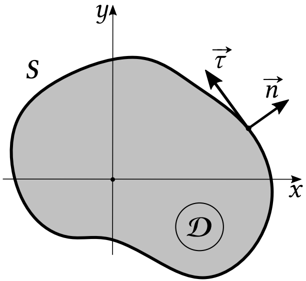
Elliptic type

Laplace's PDE:
$$\nabla^2 w^* = 0$$

$w^* = w^*(x, y)$ →
$$\tau_x = \frac{\partial w^*}{\partial x} - y, \quad \tau_y = \frac{\partial w^*}{\partial y} + x$$

Poisson's PDE:
$$\nabla^2 \phi^* = -2$$

$\phi = \phi(x, y)$ →
$$\tau_x = \frac{\partial \phi}{\partial y}, \quad \tau_y = -\frac{\partial \phi}{\partial x}$$



Simple-connected domain

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x$, $\tau_{zy}/(\mu\alpha) = \tau_y$

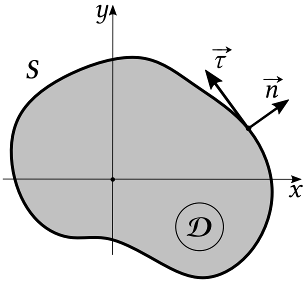
Elliptic type

Laplace's PDE:
$$\nabla^2 w^* = 0$$
$w^* = w^*(x, y)$
$$\tau_x = \frac{\partial w^*}{\partial x} - y, \quad \tau_y = \frac{\partial w^*}{\partial y} + x$$

Poisson's PDE:
$$\nabla^2 \phi^* = -2$$
$\phi = \phi(x, y)$
$$\tau_x = \frac{\partial \phi}{\partial y}, \quad \tau_y = -\frac{\partial \phi}{\partial x}$$

$$\vec{\tau} = \tau_x \vec{i} + \tau_y \vec{j}$$

$$\vec{T} = \iint_{\mathcal{D}} \vec{r} \times \vec{\tau}\, dxdy$$



Simple-connected domain

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x,\ \tau_{zy}/(\mu\alpha) = \tau_y$

Elliptic type

Laplace's PDE:
$$\nabla^2 w^* = 0$$

Poisson's PDE:
$$\nabla^2 \phi^* = -2$$

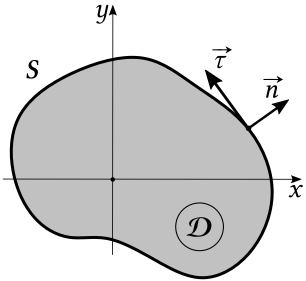
We need BC's on S!

$w^* = w^*(x, y)$ →
$$\tau_x = \frac{\partial w^*}{\partial x} - y, \quad \tau_y = \frac{\partial w^*}{\partial y} + x$$

$\phi = \phi(x, y)$ →
$$\tau_x = \frac{\partial \phi}{\partial y}, \quad \tau_y = -\frac{\partial \phi}{\partial x}$$

$$\vec{\tau} = \tau_x \vec{\imath} + \tau_y \vec{\jmath}$$

$$\vec{T} = \iint_{\mathcal{D}} \vec{r} \times \vec{\tau}\, dxdy$$



Simple-connected domain

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x,\ \tau_{zy}/(\mu\alpha) = \tau_y$

Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$

We need BC's on S!

$w^* = w^*(x, y)$ →

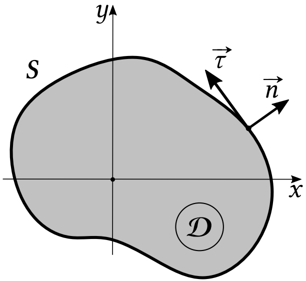
$$\tau_x = \frac{\partial w^*}{\partial x} - y, \quad \tau_y = \frac{\partial w^*}{\partial y} + x$$

$\phi = \phi(x, y)$ →

$$\tau_x = \frac{\partial \phi}{\partial y}, \quad \tau_y = -\frac{\partial \phi}{\partial x}$$

$$\vec{\tau} = \tau_x \vec{\imath} + \tau_y \vec{\jmath}$$

$$\vec{T} = \iint_{\mathcal{D}} \vec{r} \times \vec{\tau}\, dxdy$$



Simple-connected domain

Physical condition on S: zero traction

On S:

$$\vec{\tau} \cdot \vec{n} = 0$$

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x$, $\tau_{zy}/(\mu\alpha) = \tau_y$

Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$

We need BC's on S!

$w^* = w^*(x, y)$ →

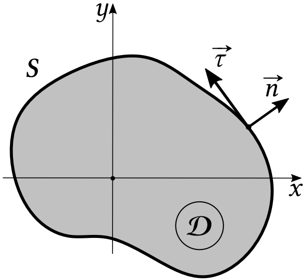
$$\tau_x = \frac{\partial w^*}{\partial x} - y, \quad \tau_y = \frac{\partial w^*}{\partial y} + x$$

$\phi = \phi(x, y)$ →

$$\tau_x = \frac{\partial \phi}{\partial y}, \quad \tau_y = -\frac{\partial \phi}{\partial x}$$

$$\vec{\tau} = \tau_x \vec{\imath} + \tau_y \vec{\jmath}$$

$$\vec{T} = \iint_{\mathcal{D}} \vec{r} \times \vec{\tau} \, dxdy$$



Simple-connected domain

Physical condition on S: zero traction

On S: $\vec{\tau} \cdot \vec{n} = 0$ →

$$\frac{d\phi}{ds} = 0$$

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x,\ \tau_{zy}/(\mu\alpha) = \tau_y$

Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$

We need BC's on S!

$w^* = w^*(x, y)$ →

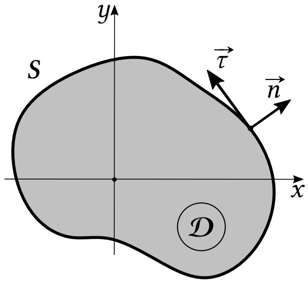
$$\tau_x = \frac{\partial w^*}{\partial x} - y,\quad \tau_y = \frac{\partial w^*}{\partial y} + x$$

$\phi = \phi(x, y)$ →

$$\tau_x = \frac{\partial \phi}{\partial y},\quad \tau_y = -\frac{\partial \phi}{\partial x}$$

$$\vec{\tau} = \tau_x\,\vec{\imath} + \tau_y\,\vec{\jmath}$$

$$\vec{T} = \iint_{\mathcal{D}} \vec{r} \times \vec{\tau}\, dxdy$$



Simple-connected domain

Physical condition on S: zero traction

On S: $\vec{\tau} \cdot \vec{n} = 0$ → $\dfrac{d\phi}{ds} = 0$ → **ϕ = const** (Dirichlet BC)

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x,\ \tau_{zy}/(\mu\alpha) = \tau_y$

Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$

We need BC's on S!

$w^* = w^*(x, y)$ →

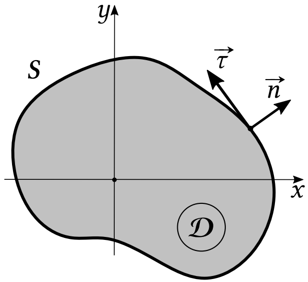
$$\tau_x = \frac{\partial w^*}{\partial x} - y, \quad \tau_y = \frac{\partial w^*}{\partial y} + x$$

$\phi = \phi(x, y)$ →

$$\tau_x = \frac{\partial \phi}{\partial y}, \quad \tau_y = -\frac{\partial \phi}{\partial x}$$

$$\vec{\tau} = \tau_x \vec{\imath} + \tau_y \vec{\jmath}$$

$$\vec{T} = \iint_{\mathcal{D}} \vec{r} \times \vec{\tau}\, dxdy$$



Simple-connected domain

Physical condition on S: zero traction

On S: $\vec{\tau} \cdot \vec{n} = 0$ → $\dfrac{d\phi}{ds} = 0$ → $\phi = \text{const}$ (Dirichlet BC)

On S: $\vec{\tau} \cdot \vec{n} = 0$ ↓

$$\frac{dw^*}{dn} = \frac{1}{2}\frac{d}{ds}(x^2 + y^2)$$

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x,\ \tau_{zy}/(\mu\alpha) = \tau_y$

Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

$w^* = w^*(x, y)$ →

$$\tau_x = \frac{\partial w^*}{\partial x} - y,\quad \tau_y = \frac{\partial w^*}{\partial y} + x$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$

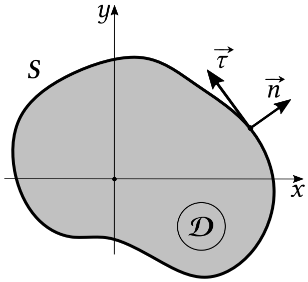
$\phi = \phi(x, y)$ →

$$\tau_x = \frac{\partial \phi}{\partial y},\quad \tau_y = -\frac{\partial \phi}{\partial x}$$

$$\vec{\tau} = \tau_x \vec{\imath} + \tau_y \vec{\jmath}$$

$$\vec{T} = \iint_{\mathcal{D}} \vec{r} \times \vec{\tau}\, dxdy$$

We need BC's on S!



Simple-connected domain

Physical condition on S: zero traction

On S: $\vec{\tau} \cdot \vec{n} = 0$ → $\frac{d\phi}{ds} = 0$ → $\boldsymbol{\phi = \text{const}}$ (Dirichlet BC)

On S: $\vec{\tau} \cdot \vec{n} = 0$ → $\frac{dw^*}{dn} = \frac{1}{2}\frac{d}{ds}(x^2 + y^2)$ → $\boldsymbol{\frac{dw^*}{dn} = y n_x - x n_y}$ (Neumann BC)

Final Formulation, With Simple Scaling: $\tau_{zx}/(\mu\alpha) = \tau_x$, $\tau_{zy}/(\mu\alpha) = \tau_y$

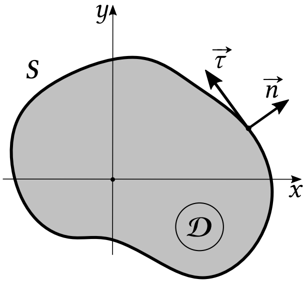
Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$



Simple-connected domain

Physical condition on S: zero traction

On S: $\vec{\tau} \cdot \vec{n} = 0$

$\phi = \text{const}$ (Dirichlet BC)

$\frac{dw^*}{dn} = yn_x - xn_y$ (Neumann BC)

Additional Generalization: What if We Have Hollow Cylinder

Additional Generalization: What if We Have Hollow Cylinder

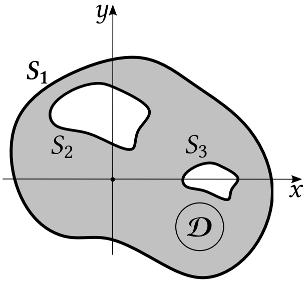
Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$



Multiply-connected domain

Additional Generalization: What if We Have Hollow Cylinder

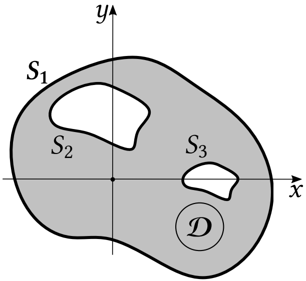
Elliptic type

Laplace's PDE:
$$\nabla^2 w^* = 0$$

Poisson's PDE:
$$\nabla^2 \phi^* = -2$$

The same physical condition: no traction on each contour

$\rightarrow$ boundary conditions on S_i:



Multiply-connected domain

Additional Generalization: What if We Have Hollow Cylinder

Elliptic type

Laplace's PDE:
$$\nabla^2 w^* = 0$$

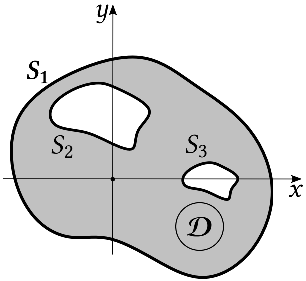
Poisson's PDE:
$$\nabla^2 \phi^* = -2$$

The same physical condition: no traction on each contour

$\rightarrow$ boundary conditions on S_i:

$\phi = C_i \quad \rightarrow \quad$ different values of constant C_i!

$\dfrac{dw^*}{dn} = yn_x - xn_y \quad \rightarrow \quad$ geometry of the contour S_i



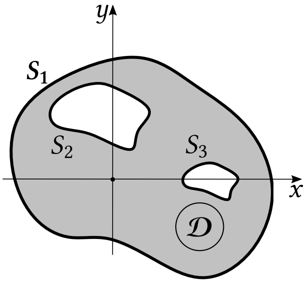
Multiply-connected domain

Additional Generalization: What if We Have Hollow Cylinder

Elliptic type

Laplace's PDE:
$$\nabla^2 w^* = 0$$

Poisson's PDE:
$$\nabla^2 \phi^* = -2$$



Multiply-connected domain

The same physical condition: no traction on each contour

$\rightarrow$ boundary conditions on S_i:

$\phi = C_i \quad \rightarrow \quad$ different values of constant C_i!

$\dfrac{dw^*}{dn} = yn_x - xn_y \quad \rightarrow \quad$ geometry of the contour S_i

Problem: values of C_i are not known in advance!

Additional Generalization: What if We Have Hollow Cylinder

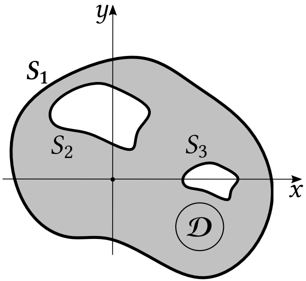
Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$



Multiply-connected domain

The same physical condition: no traction on each contour

$\rightarrow$ boundary conditions on S_i:

$\phi = C_i$	$\rightarrow$	different values of constant C_i!
$\dfrac{dw^*}{dn} = yn_x - xn_y$	$\rightarrow$	geometry of the contour S_i

Problem: values of C_i are not known in advance!

- Green's theorem can't help much:

$$\oint_{S_i} \vec{\tau} \cdot d\vec{s} = \iint_{A_i} \underbrace{\nabla \times \vec{\tau}}_{=2} \, dA \quad \rightarrow \quad \oint_{S_i} \vec{\tau} \cdot d\vec{s} = 2A_i$$

Additional Generalization: What if We Have Hollow Cylinder

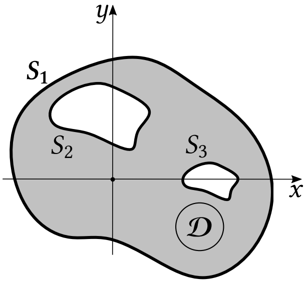
Elliptic type

Laplace's PDE:

$$\nabla^2 w^* = 0$$

Poisson's PDE:

$$\nabla^2 \phi^* = -2$$



Multiply-connected domain

The same physical condition: no traction on each contour

$\rightarrow$ boundary conditions on S_i:

$\phi = C_i \quad \rightarrow \quad$ different values of constant C_i!

$\dfrac{dw^*}{dn} = y n_x - x n_y \quad \rightarrow \quad$ geometry of the contour S_i

Problem: values of C_i are not known in advance!

- Green's theorem can't help much:

$$\oint_{S_i} \vec{\tau} \cdot d\vec{s} = \iint_{A_i} \underbrace{\nabla \times \vec{\tau}}_{=2} \, dA \quad \rightarrow \quad \oint_{S_i} \vec{\tau} \cdot d\vec{s} = 2A_i$$

$$\vec{\tau} \,||\, d\vec{s} \quad (\vec{\tau} = \tau \vec{e}_s) \quad \rightarrow \quad \oint_{S_i} \tau \, ds = \oint_{S_i} \frac{d\phi}{dn} \, ds = 2A_i$$

Finite Volume Method (FVM): Numerical Method for Solving of PDE's

- Natural method of discretization for PDE's written in strong conservative form (fluid mechanics usually, but it can be also used for solid mechanics!)

Finite Volume Method (FVM): Numerical Method for Solving of PDE's

- Natural method of discretization for PDE's written in strong conservative form (fluid mechanics usually, but it can be also used for solid mechanics!)

- General transport equation for quantity φ in fluid flow field

$$\frac{\partial(\rho\varphi)}{\partial t} + \nabla \cdot (\rho\vec{U}\varphi) = \nabla \cdot (\Gamma\nabla\varphi) + S_\varphi$$

Finite Volume Method (FVM): Numerical Method for Solving of PDE's

- Natural method of discretization for PDE's written in strong conservative form (fluid mechanics usually, but it can be also used for solid mechanics!)

- General transport equation for quantity φ in fluid flow field

$$\frac{\partial(\rho\varphi)}{\partial t} + \nabla \cdot (\rho\vec{U}\varphi) = \nabla \cdot (\Gamma\nabla\varphi) + S_\varphi$$

- Integration over volume V and application of Ostrogradsky-Gauss theorem

$$\iiint_V \frac{\partial(\rho\varphi)}{\partial t}\,\mathrm{d}V + \oiint_A \vec{n} \cdot (\rho\varphi\vec{U})\,\mathrm{d}A = \oiint_A \vec{n} \cdot (\Gamma\,\nabla\varphi)\,\mathrm{d}A + \iiint_V S_T\,\mathrm{d}V$$

Finite Volume Method (FVM): Numerical Method for Solving of PDE's

- Natural method of discretization for PDE's written in strong conservative form (fluid mechanics usually, but it can be also used for solid mechanics!)

- General transport equation for quantity φ in fluid flow field

$$\frac{\partial(\rho\varphi)}{\partial t} + \nabla \cdot (\rho\vec{U}\varphi) = \nabla \cdot (\Gamma\nabla\varphi) + S_\varphi$$

- Integration over volume V and application of Ostrogradsky-Gauss theorem

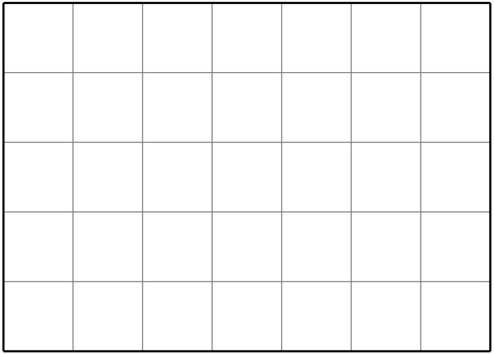
$$\iiint_V \frac{\partial(\rho\varphi)}{\partial t}\,\mathrm{d}V + \oiint_A \vec{n} \cdot (\rho\varphi\vec{U})\,\mathrm{d}A = \oiint_A \vec{n} \cdot (\Gamma\,\nabla\varphi)\,\mathrm{d}A + \iiint_V S_T\,\mathrm{d}V$$

- Pure steady-state diffusion, without source term

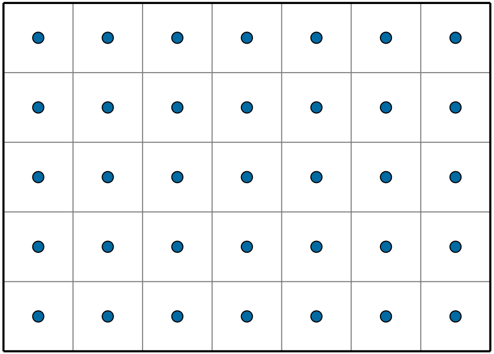
$$\oiint_A \vec{n} \cdot (\Gamma\,\nabla\varphi)\,\mathrm{d}A = 0 \quad \longrightarrow \quad \text{for } \Gamma = \text{const} \quad \longrightarrow \quad \oiint_A \vec{n} \cdot \nabla\varphi\,\mathrm{d}A = 0 \qquad (\text{PDE: } \nabla^2\varphi = 0)$$

FVM: General Principle of Discretization of Computational Domain

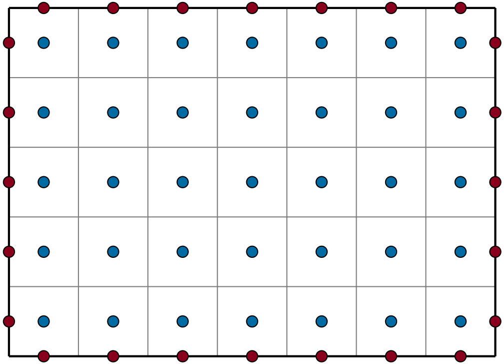
FVM: General Principle of Discretization of Computational Domain



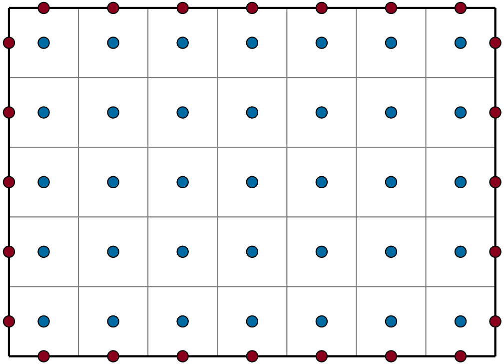
FVM: General Principle of Discretization of Computational Domain



FVM: General Principle of Discretization of Computational Domain

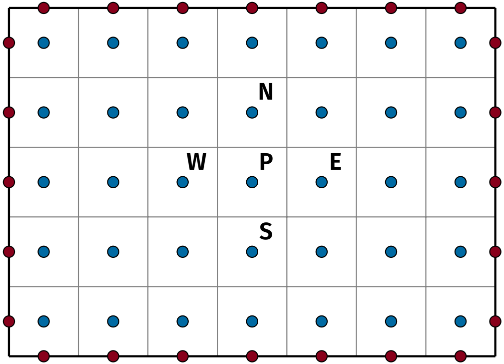


FVM: General Principle of Discretization of Computational Domain



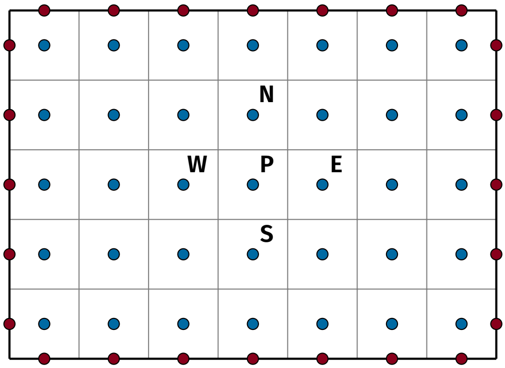
- Boundary conditions
- Computational points

FVM: General Principle of Discretization of Computational Domain



- Boundary conditions
- Computational points

FVM: General Principle of Discretization of Computational Domain

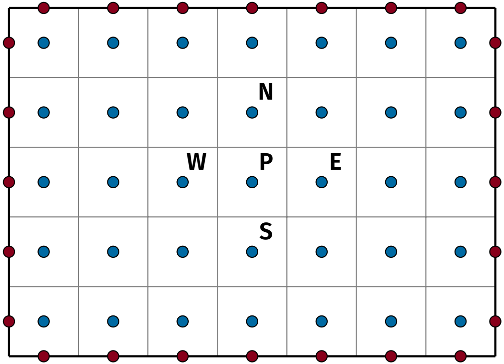


- Boundary conditions
- Computational points

P - centroid of computational cell (point)

W, E, N, S - centroids of neigbouring cells

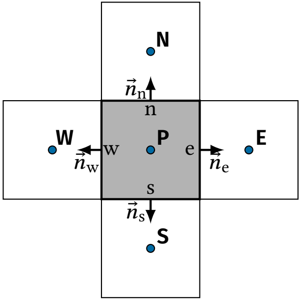
FVM: General Principle of Discretization of Computational Domain

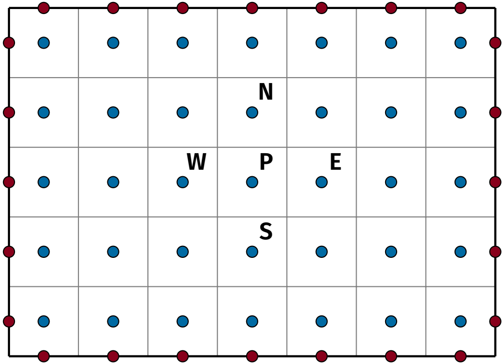


- Boundary conditions
- Computational points

P - centroid of computational cell (point)

W, E, N, S - centroids of neigbouring cells

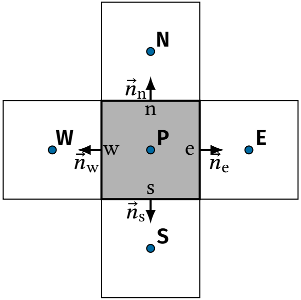




- Boundary conditions
- Computational points

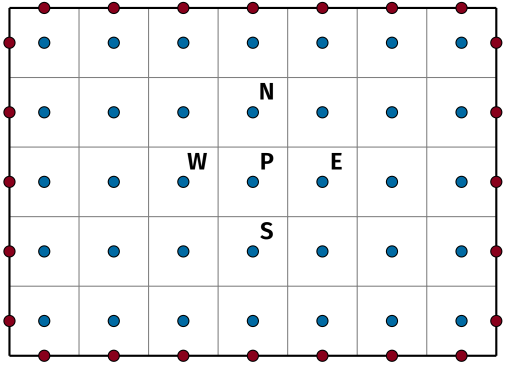
P - centroid of computational cell (point)

W, E, N, S - centroids of neigbouring cells



$$\oiint_A \vec{n} \cdot \nabla\varphi \, \mathrm{d}A = \sum_k \iint_{A_k} \vec{n} \cdot \nabla\varphi \, \mathrm{d}A, \quad k = \mathrm{e, w, n, s}$$

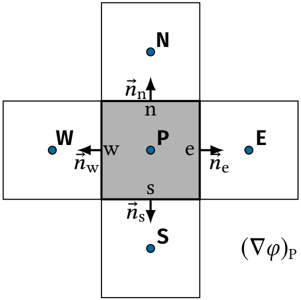
FVM: General Principle of Discretization of Computational Domain



- Boundary conditions
- Computational points

P - centroid of computational cell (point)

W, E, N, S - centroids of neigbouring cells



$$(\nabla\varphi)_P = \frac{1}{V_p}\sum_f \vec{n}_f \varphi_f A_f$$

$$\oiint_A \vec{n}\cdot\nabla\varphi\,\mathrm{d}A = \sum_k \iint_{A_k} \vec{n}\cdot\nabla\varphi\,\mathrm{d}A, \quad k = \mathrm{e, w, n, s}$$

Analysis of the Problem and Numerical Methods: Done! Imlementation Follows

Analysis of the Problem and Numerical Methods: Done! Imlementation Follows

- **Open-source philosophy:** use existing codes and suite them for your needs

```
const volVectorField& C = mesh.C();
const surfaceVectorField& Sf = mesh.Sf();
volVectorField coord = -C.component(vector::Y)*ei + C.
     component(vector::X)*ej;

fvScalarMatrix WEqn
(
  fvm::ddt(W) - fvm::laplacian(k, W)
);
WEqn.relax();
WEqn.solve();

tau = fvc::grad(W) + coord;
phi = fvc::interpolate(tau) & Sf;
magTau =mag(tau);
```

```
const fvPatch& boundaryPatch = patch();
const vectorField& Cf = boundaryPatch.Cf();
// Vektori normala na granicnoj povrsi
const vectorField& Sf = boundaryPatch.Sf();
// Povrsine pojedinacnih povrsi
const scalarField& magSf = boundaryPatch.magSf();
scalarField& field = this->refGrad();
forAll(Cf, faceI)
{
  const scalar x = Cf[faceI].x();
  const scalar y = Cf[faceI].y();
  const scalar nx = Sf[faceI].x()/magSf[faceI];
  const scalar ny = Sf[faceI].y()/magSf[faceI];
  field[faceI] = y*nx - x*ny;
}
```

Analysis of the Problem and Numerical Methods: Done! Imlementation Follows

- **Open-source philosophy:** use existing codes and suite them for your needs
- **OpenFOAM:** huge collection of C++ libraries for CCM (based on FVM)

```
const volVectorField& C = mesh.C();
const surfaceVectorField& Sf = mesh.Sf();
volVectorField coord = -C.component(vector::Y)*ei + C.
     component(vector::X)*ej;

fvScalarMatrix WEqn
(
  fvm::ddt(W) - fvm::laplacian(k, W)
);
WEqn.relax();
WEqn.solve();

tau = fvc::grad(W) + coord;
phi = fvc::interpolate(tau) & Sf;
magTau =mag(tau);
```

```
const fvPatch& boundaryPatch = patch();
const vectorField& Cf = boundaryPatch.Cf();
// Vektori normala na granicnoj povrsi
const vectorField& Sf = boundaryPatch.Sf();
// Povrsine pojedinacnih povrsi
const scalarField& magSf = boundaryPatch.magSf();
scalarField& field = this->refGrad();
forAll(Cf, faceI)
{
  const scalar x = Cf[faceI].x();
  const scalar y = Cf[faceI].y();
  const scalar nx = Sf[faceI].x()/magSf[faceI];
  const scalar ny = Sf[faceI].y()/magSf[faceI];
  field[faceI] = y*nx - x*ny;
}
```

Analysis of the Problem and Numerical Methods: Done! Imlementation Follows

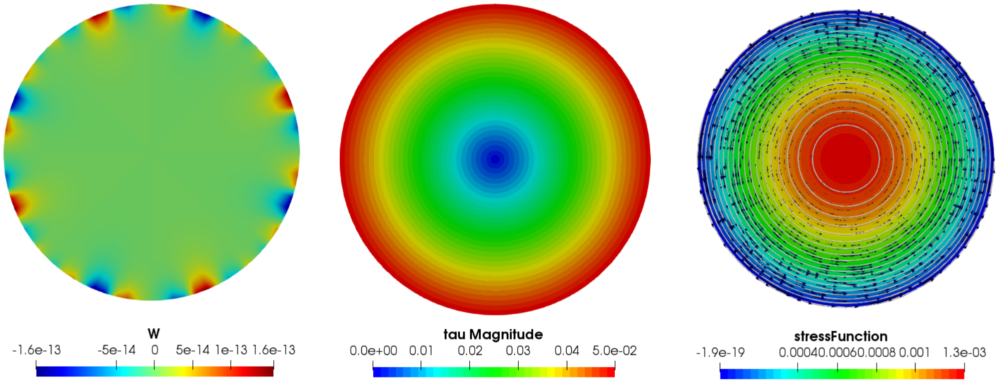
- **Open-source philosophy:** use existing codes and suite them for your needs
- **OpenFOAM:** huge collection of C++ libraries for CCM (based on FVM)
- Modification of the solver `laplacianFoam`, create `torsionWarpingFoam`, with additional implementation for boundary condition

```
const volVectorField& C = mesh.C();
const surfaceVectorField& Sf = mesh.Sf();
volVectorField coord = -C.component(vector::Y)*ei + C.
      component(vector::X)*ej;

fvScalarMatrix WEqn
(
  fvm::ddt(W) - fvm::laplacian(k, W)
);
WEqn.relax();
WEqn.solve();

tau = fvc::grad(W) + coord;
phi = fvc::interpolate(tau) & Sf;
magTau =mag(tau);
```

```
const fvPatch& boundaryPatch = patch();
const vectorField& Cf = boundaryPatch.Cf();
// Vektori normala na granicnoj povrsi
const vectorField& Sf = boundaryPatch.Sf();
// Povrsine pojedinacnih povrsi
const scalarField& magSf = boundaryPatch.magSf();
scalarField& field = this->refGrad();
forAll(Cf, faceI)
{
  const scalar x = Cf[faceI].x();
  const scalar y = Cf[faceI].y();
  const scalar nx = Sf[faceI].x()/magSf[faceI];
  const scalar ny = Sf[faceI].y()/magSf[faceI];
  field[faceI] = y*nx - x*ny;
}
```

Simple Cases, Known Analytical Solution - Circle And Square Domain

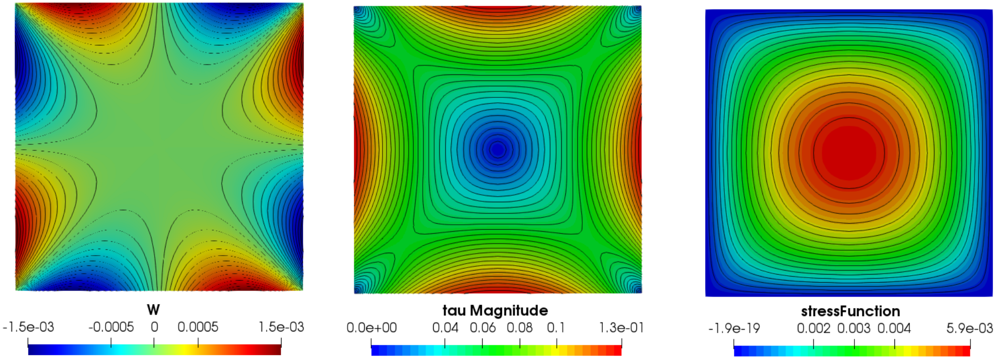
- **Circle:** distribution of warping function, stress vectors, and stress function



- Perfect agreement with analytical solution

Simple Cases, Known Analytical Solution - Circle And Square Domain

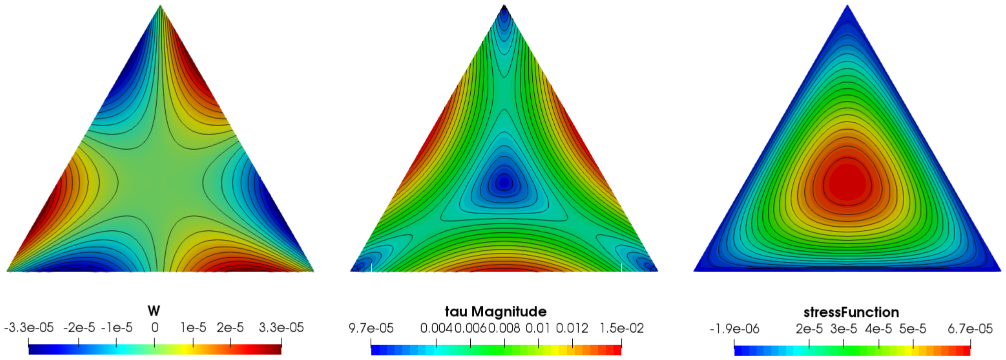
- **Square:** distribution of warping function, stress vectors, and stress function



- Perfect agreement with analytical solution

Simple Cases, Known Analytical Solution - Equiliteral Triangle

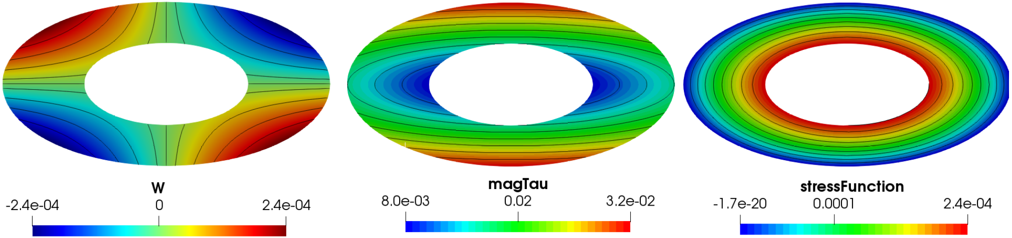
- **Equiliteral triangle:** distribution of warping function, stress vectors, and stress function



- Perfect agreement with analytical solution

Multiply-connected Domains: Hollow Ellipse (Known Analytical Solution)

- Geometrical characteristics: $a/b = 2$, $k = 0.5$ (outer ellipse $a = 4\,\mathrm{cm}$, $b = 2\,\mathrm{cm}$)



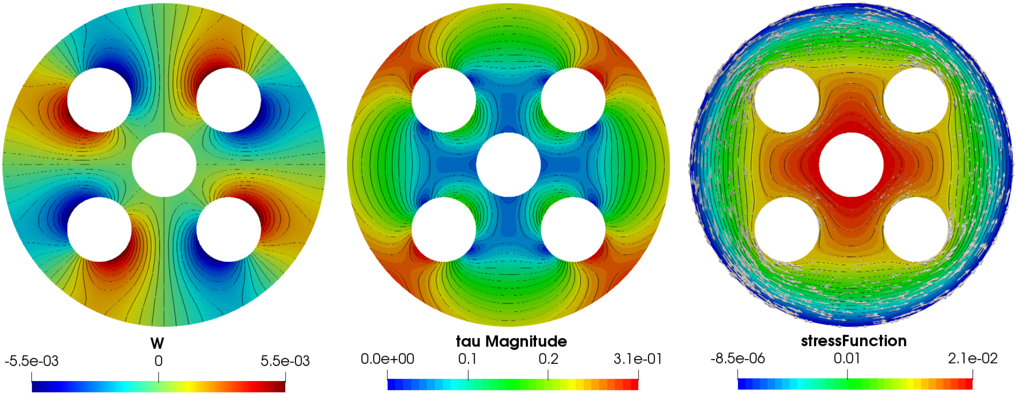
- Analytical solution for torque (torsional constant): $\dfrac{T}{\mu\alpha} = \dfrac{a^3 b^3 \pi}{a^2 + b^2}(1 - k^4) = 75.39822\,\mathrm{cm}^4$

- Numerical solution: based on solved distribution for w^* and numerical integration:

$$\frac{T}{\mu\alpha} = \iint_{\mathcal{D}} \underbrace{\left(x^2 + y^2 + x\frac{\partial w^*}{\partial y} - y\frac{\partial w^*}{\partial x}\right)}_{F} dxdy \;\longrightarrow\; [\text{fvc::domainIntegrate(F)}] \;\longrightarrow\; 75.38238\,\mathrm{cm}^4$$

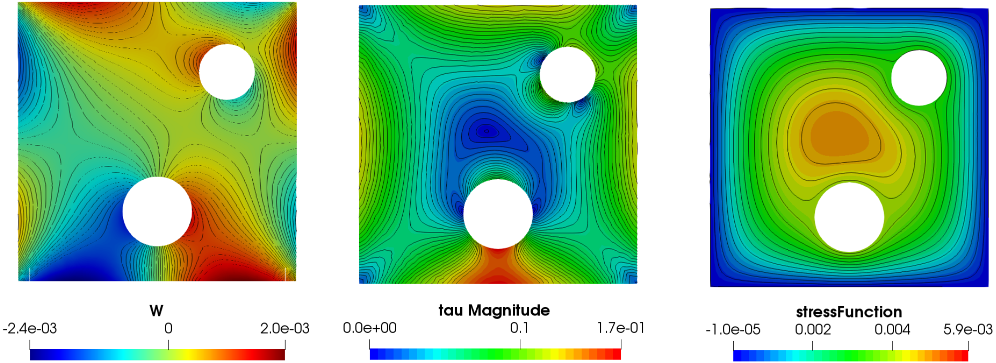
Multiply-connected Domains: Playing Around With Some Random Shapes

- Domain symmetry - symmetry in solution (well known and expected)

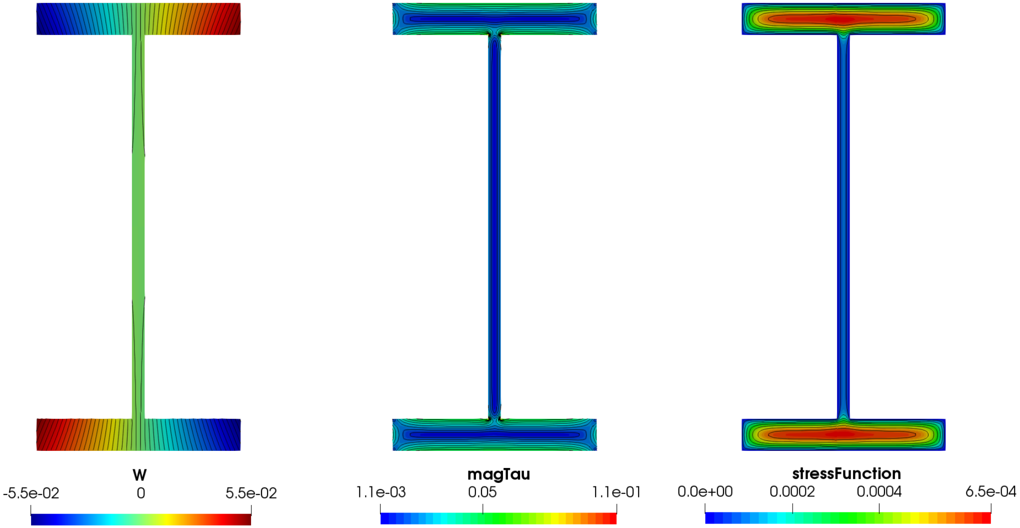


Multiply-connected Domains: Playing Around With Some Random Shapes

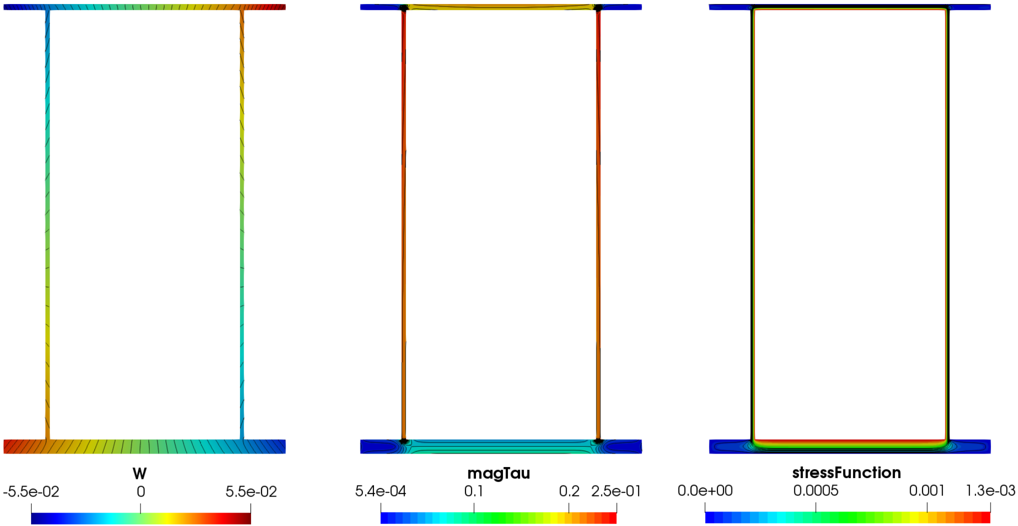
- Domain asymmetry - asymmetry in solution (well known and expected)



Practical, Engineering Applications (Thin-walled Cross-sections)



Practical, Engineering Applications (Thin-walled Cross-sections)



Summary

- Problem of elastic torsion is analyzed using Saint-Venant principle

Summary

- Problem of elastic torsion is analyzed using Saint-Venant principle
- Obtained Laplace's PDE with inhomogeneous Neumann BC is solved numerically using finite volume method

Summary

- Problem of elastic torsion is analyzed using Saint-Venant principle
- Obtained Laplace's PDE with inhomogeneous Neumann BC is solved numerically using finite volume method

Summary

- Problem of elastic torsion is analyzed using Saint-Venant principle
- Obtained Laplace's PDE with inhomogeneous Neumann BC is solved numerically using finite volume method
- OpenFOAM tools are used for that purpose, and new solver is created

Summary

- Problem of elastic torsion is analyzed using Saint-Venant principle
- Obtained Laplace's PDE with inhomogeneous Neumann BC is solved numerically using finite volume method
- OpenFOAM tools are used for that purpose, and new solver is created
- This is very simple and rather isolated approach for numericaly studies of solid mechanics (torsion problem), but (for those interested)

Summary

- Problem of elastic torsion is analyzed using Saint-Venant principle
- Obtained Laplace's PDE with inhomogeneous Neumann BC is solved numerically using finite volume method
- OpenFOAM tools are used for that purpose, and new solver is created
- This is very simple and rather isolated approach for numericaly studies of solid mechanics (torsion problem), but (for those interested)
- Check out the work of Aleksandar Karač, Željko Tuković, Hrvoje Jasak, Philip Cardiff, Declan Carolan, Michael Leonard and Valentine Kanyanta:

Summary

- Problem of elastic torsion is analyzed using Saint-Venant principle
- Obtained Laplace's PDE with inhomogeneous Neumann BC is solved numerically using finite volume method
- OpenFOAM tools are used for that purpose, and new solver is created
- This is very simple and rather isolated approach for numericaly studies of solid mechanics (torsion problem), but (for those interested)
- Check out the work of Aleksandar Karač, Željko Tuković, Hrvoje Jasak, Philip Cardiff, Declan Carolan, Michael Leonard and Valentine Kanyanta:

SOLID MECHANICS – FINITE VOLUME SOLVERS

https://github.com/wyldckat/solidMechanics