MultiFEBE: an open-source multi-domain integrated finite element and boundary element software

J.D.R. Bordón A.G. Vega-Artiles G.M. Álamo
L.A. Padrón J.J. Aznárez O. Maeso

ULPGC
Universidad de Las Palmas de Gran Canaria

Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería

SIANI

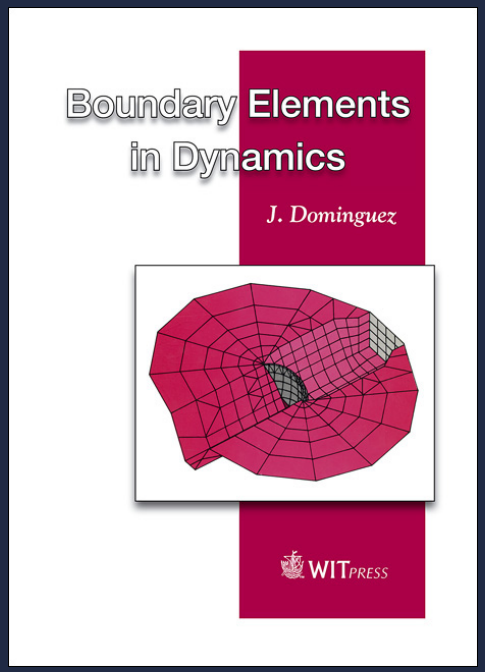
Congress on Numerical Methods in Engineering
CMN 2022
Las Palmas de Gran Canaria, Spain, 13 September, 2022

Outline

Outline

A little bit of history: origins

J. Domínguez, Boundary Elements in Dynamics, 1993.



- Time harmonic 2D anti-plane elastod.: `CONDAPEH`, `QUADAPEH`
- Time harmonic 2D plane elastod.: `CONDPLEH`, `QUADPLEH`
- Time harmonic 3D scalar wave prop.: `CONDTHSH`, `QUADTHSH`
- Time harmonic 3D elastod.: `CONDTHEH`, `QUADTHEH`
- and others ...

A little bit of history: in-house codes (dams)

3D BEM–BEM multi-domain multi-material models:

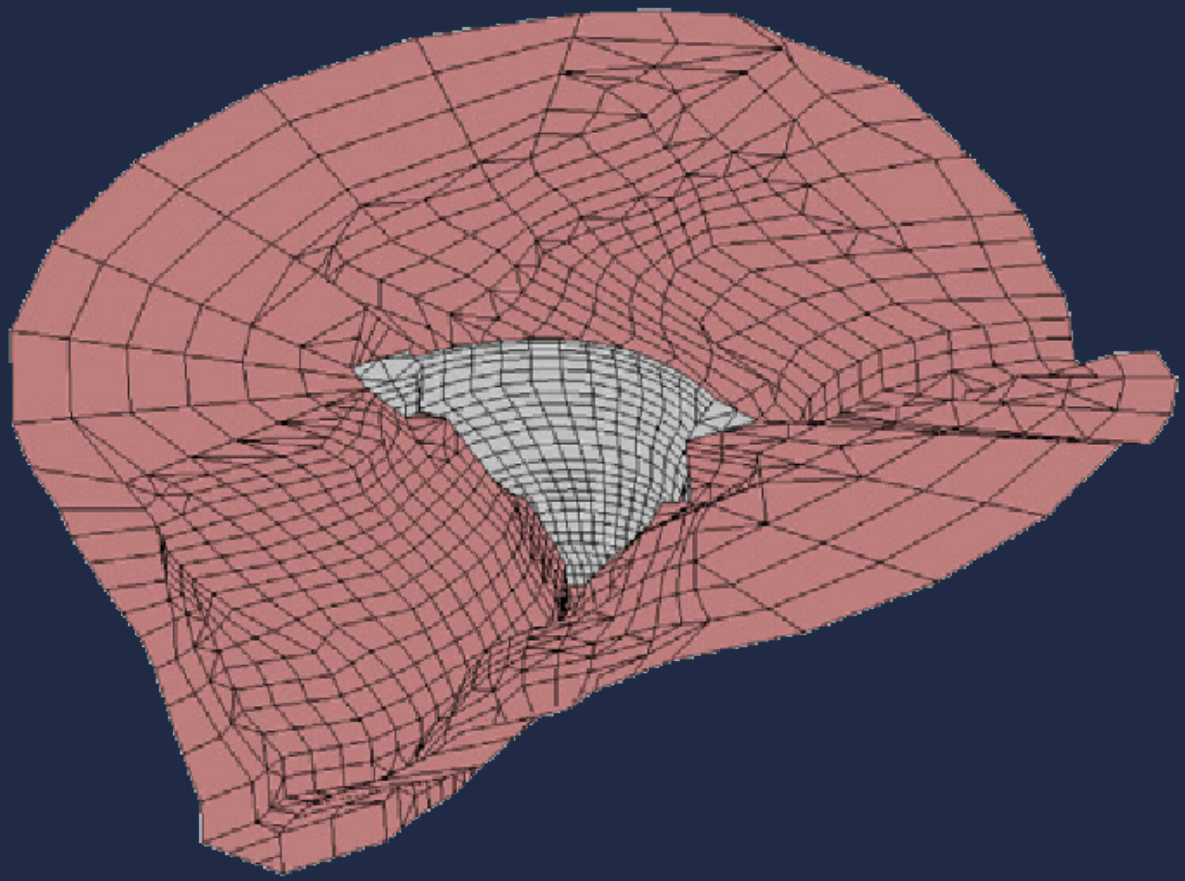


Figure: J.C. Galván et al., Eng Anal Bound Elem 144, 2022.

A little bit of history: in-house codes (cracks)

2D BEM (singular & hypersingular) models for crack analysis:

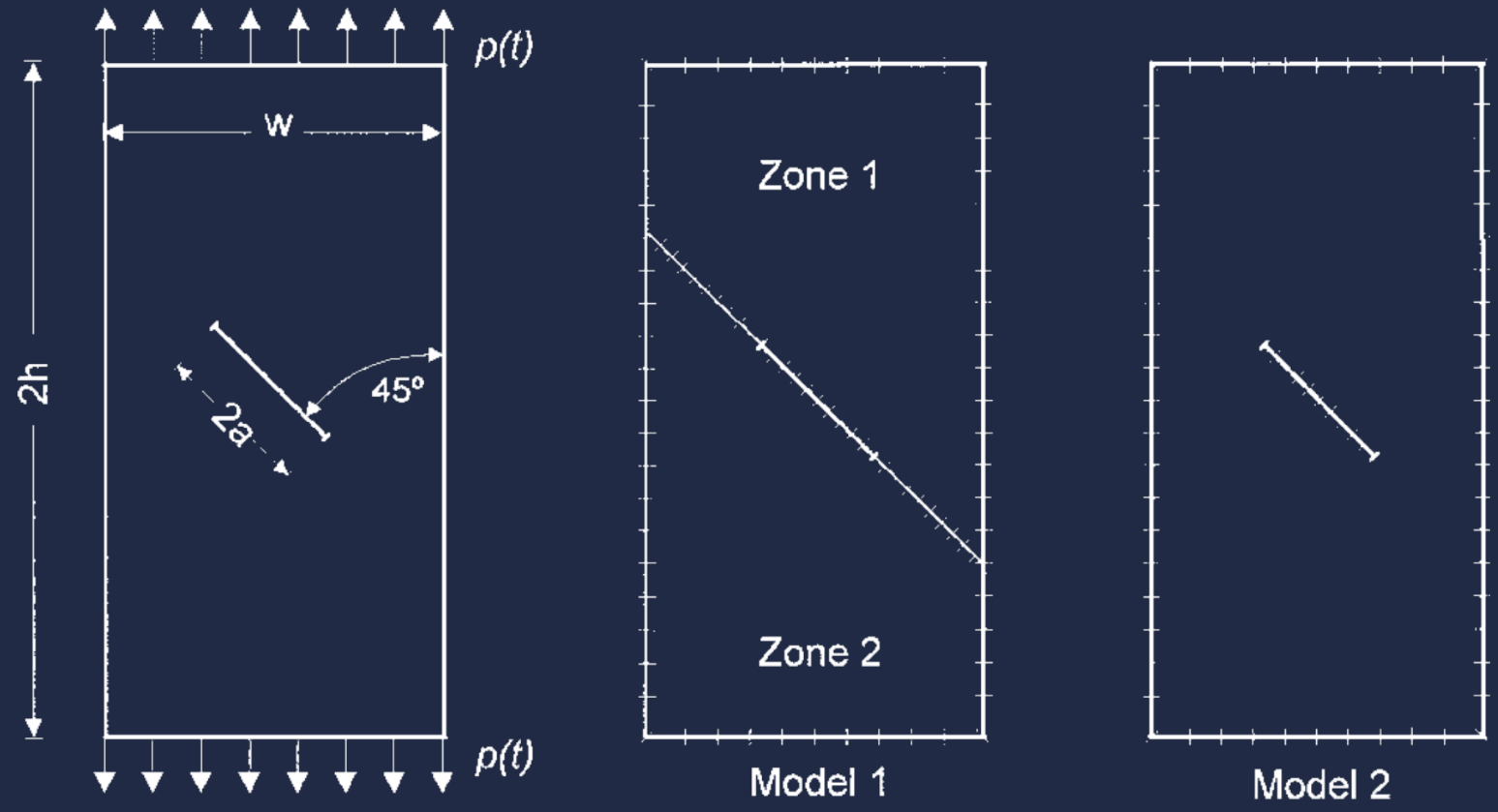


Figure: F. Chirino & R. Abascal, IJNME 43, 1998.

A little bit of history: in-house codes (acoustics)

2D BEM (singular & hypersingular) models for acoustics:

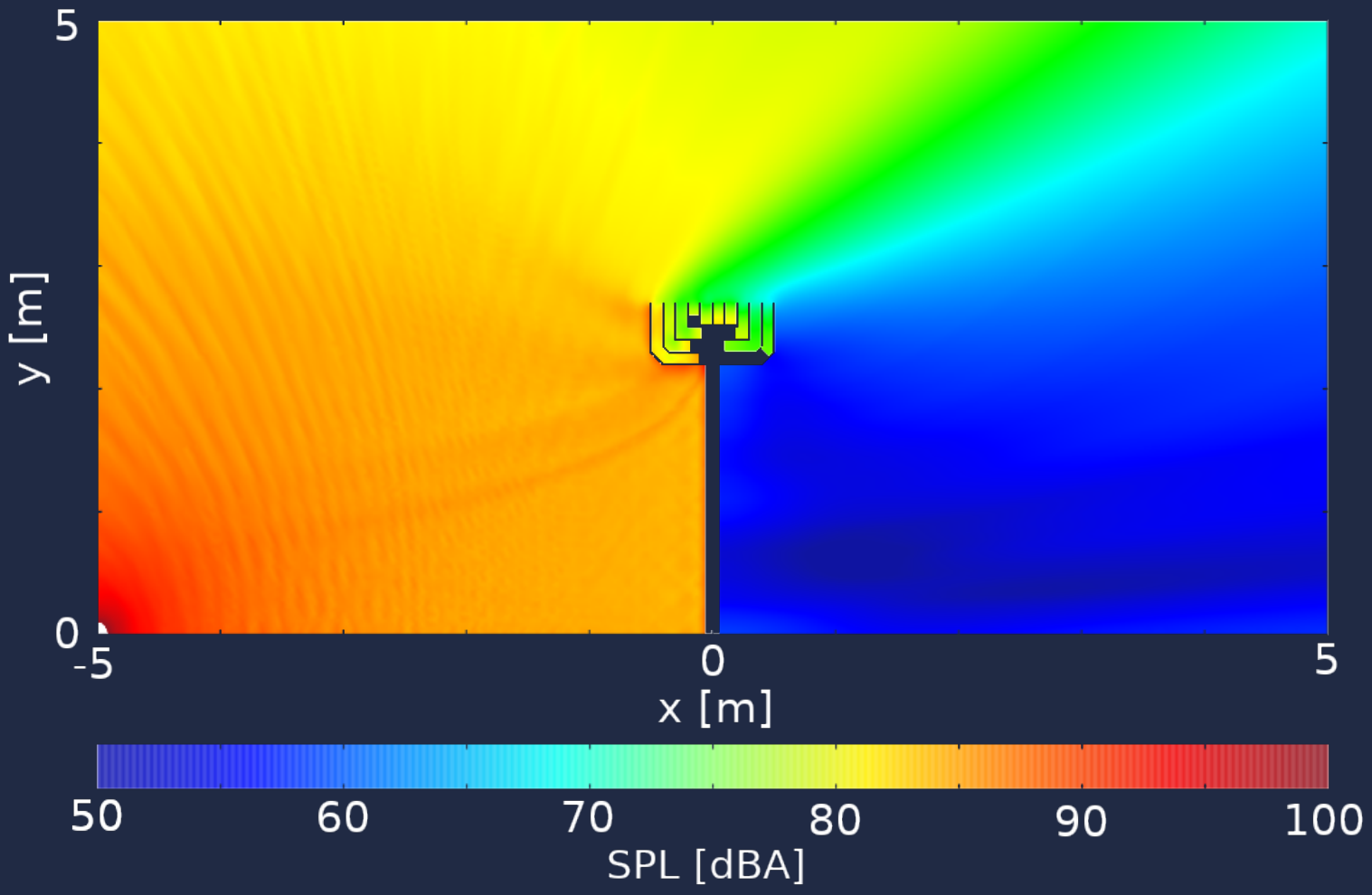


Figure: R. Toledo et al., Eng Anal Bound Elem 63, 2016.

A little bit of history: in-house codes (SSI)

3D BEM–FEM models for analysis of pile and pile groups:

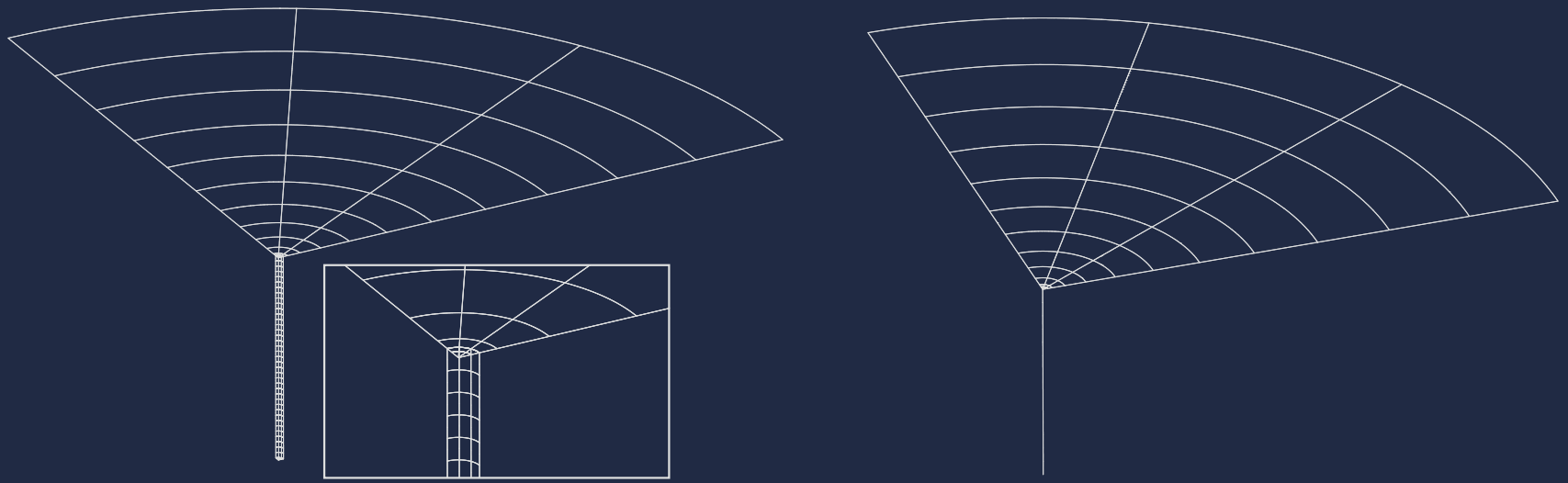


Figure: L.A. Padrón et al., Eng Anal Bound Elem 31, 2007.

A little bit of history: in-house codes (SSI)

2D/3D BEM-FEM models for buried shell structures:

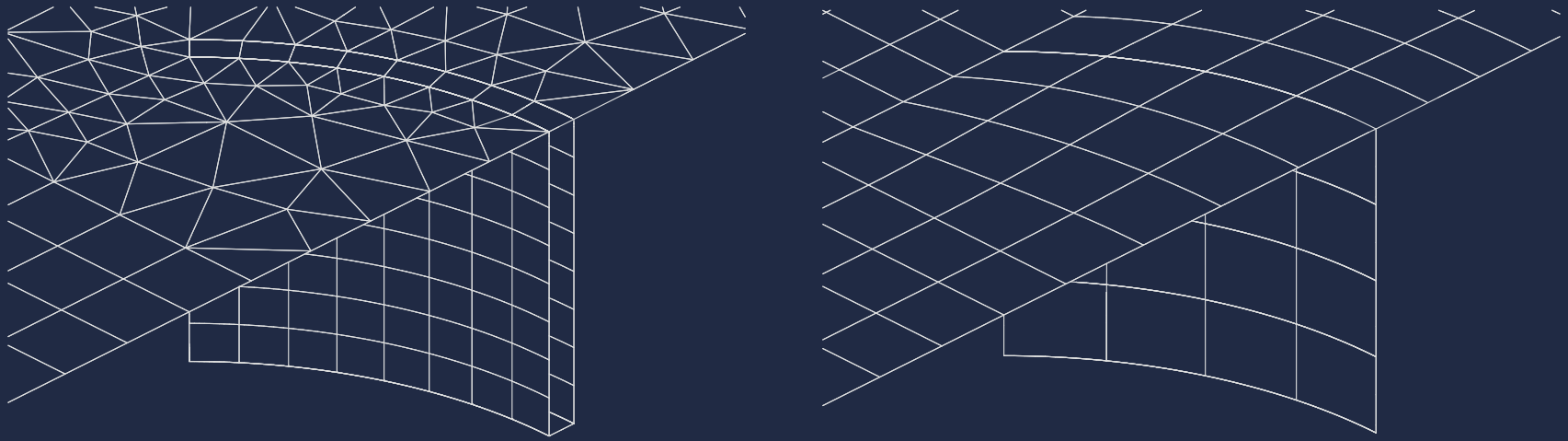
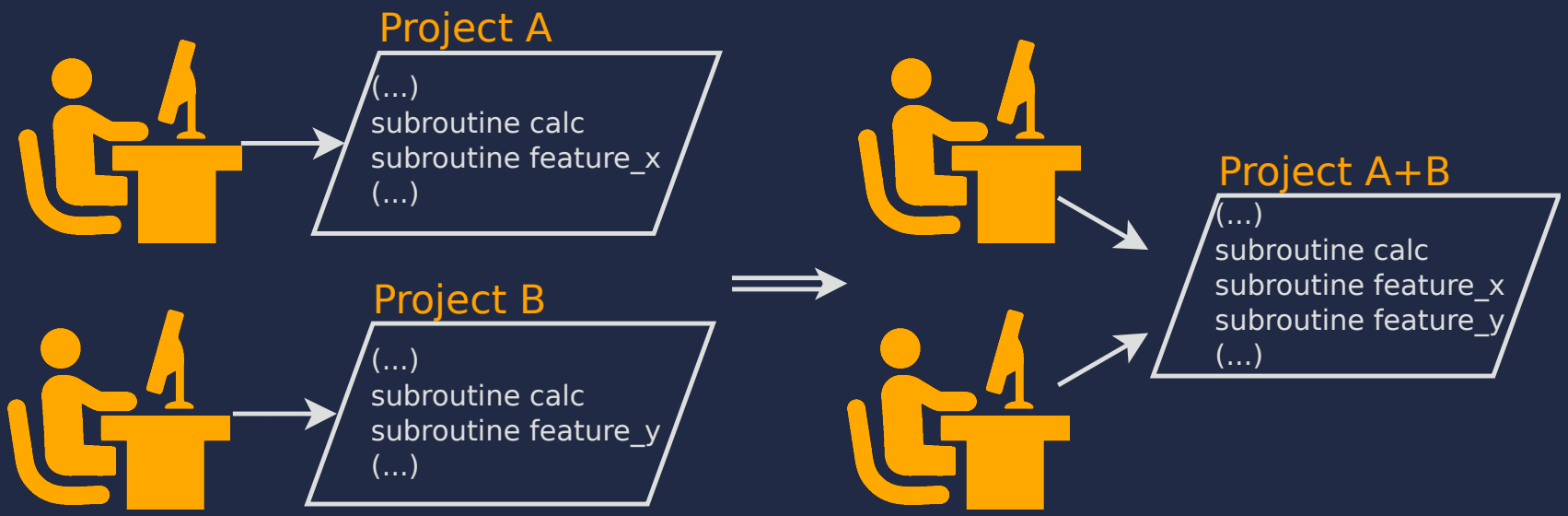


Figure: J.D.R. Bordón et al., Comput Mech 60, 2017.

Why developing MultiFEBE?

- ✗ Repeated work for maintaining several codes
- ✗ Features scattered in different codes
- ✓ Combine features and methodologies
- ✓ Share core algorithms



Why developing MultiFEBE?

Existing projects (not open source):

- BEASY: corrosion, cathodic protection, crack analysis
- BEASTS: elastodynamics, FEM via ANSYS
- ...

Existing projects (open source):

- Domínguez's book codes[1]
- Beer's book codes[2]: elasticity, limited BEM-FEM
- OpenBEM: acoustics
- Bempp: acoustics & EM
- ...

[1] J. Domínguez, Boundary Elements in Dynamics, WIT Press, 1993.

[2] G. Beer et al., The Boundary Element Method with Programming, Springer-Verlag, 2008.

Why developing MultiFEBE?

No previous software combining:

- BEM for acoustics, elastostatics, elastodynamics, poroelastodynamics
- Ordinary and crack boundary elements
- Multi-domain multi-material coupling
- Mixed-dimensional BEM-FEM coupling

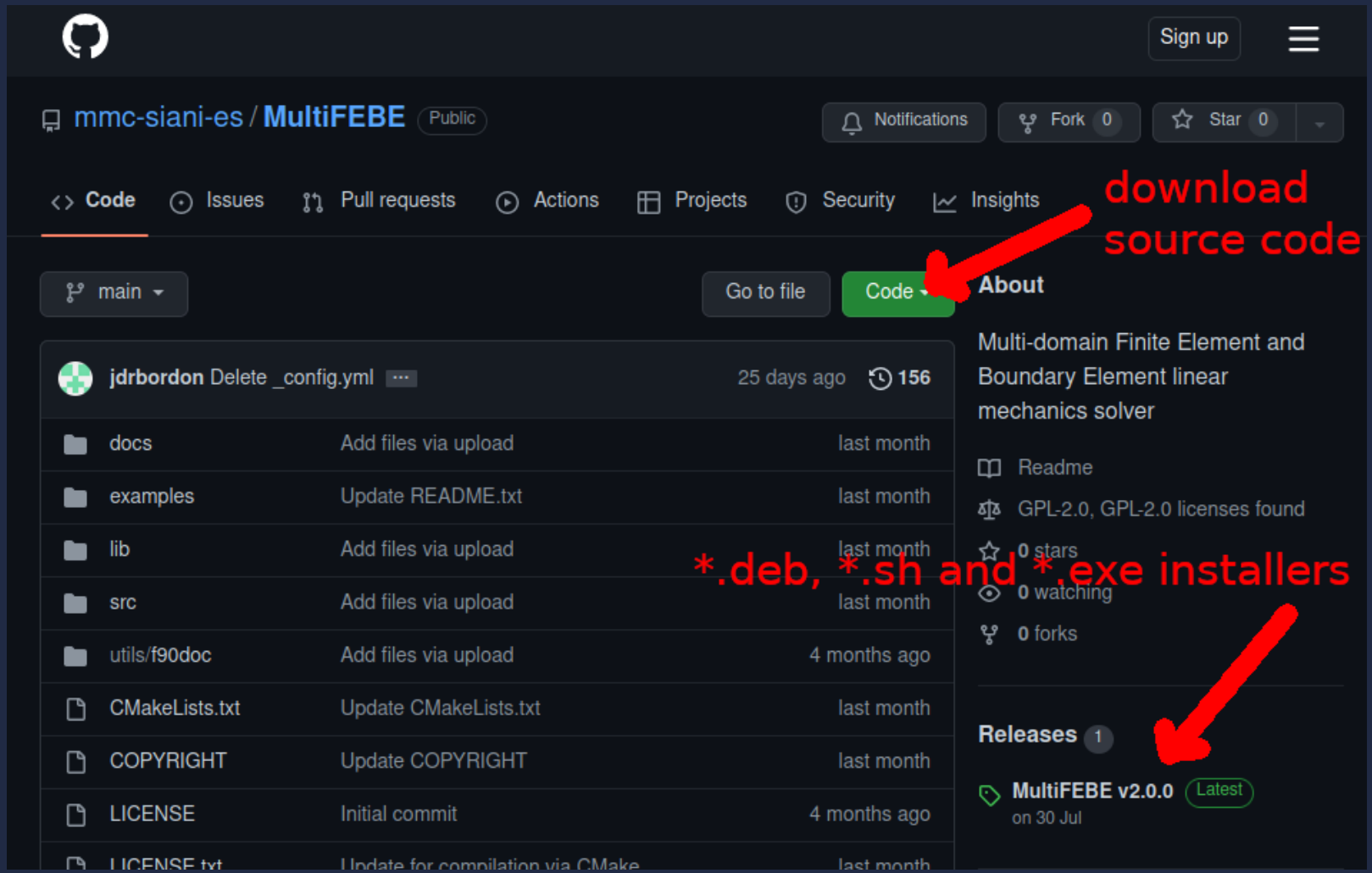
License

GNU GPLv2

Can	Cannot	Must
▶ Commercial Use	▶ Sublicense	▶ Include original
▶ Modify	▶ Hold Liable	▶ Disclose source
▶ Distribute		▶ Include Copyright
▶ Place Warranty		▶ State Changes
		▶ Include License

Repository at GitHub

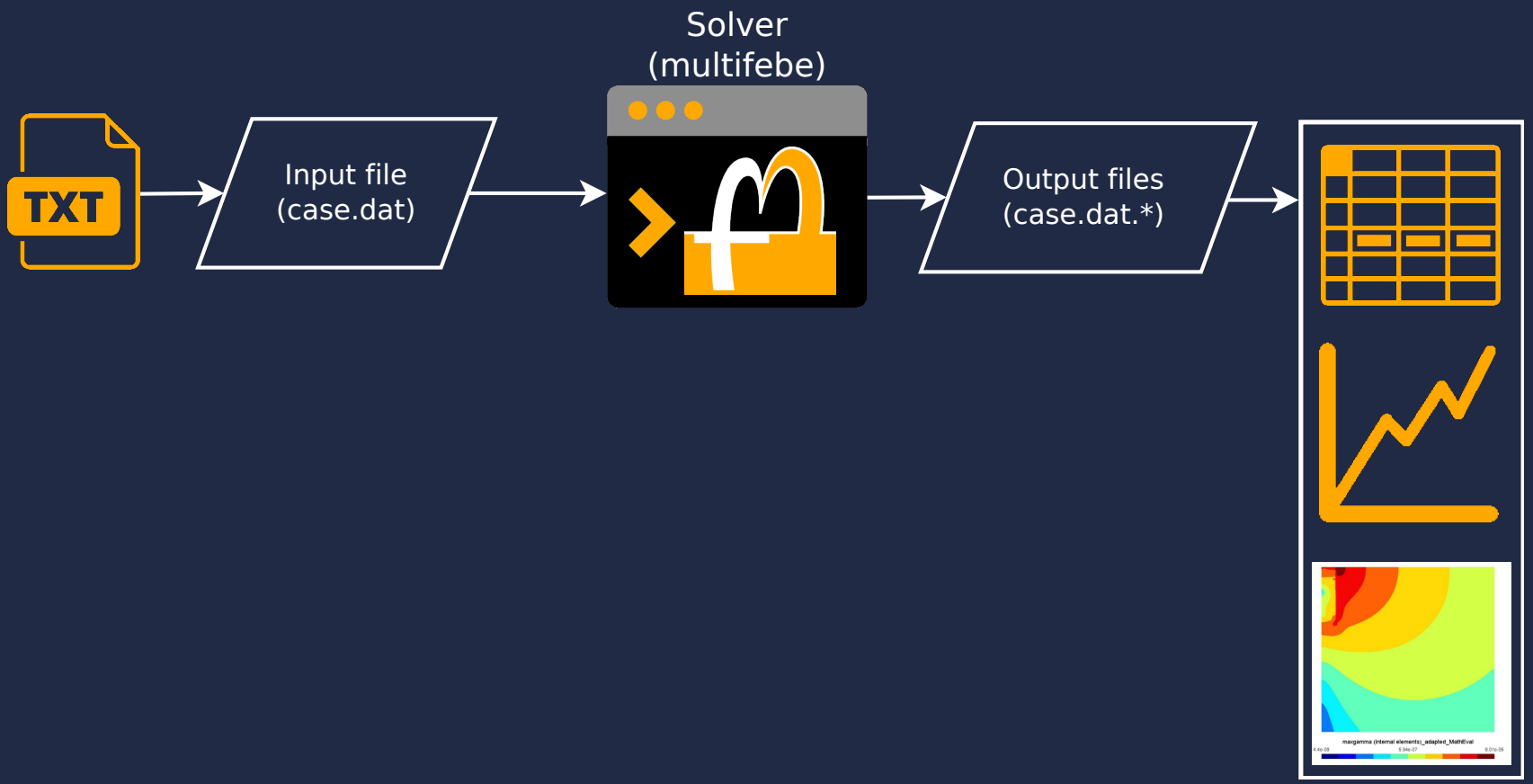
https://github.com/mmc-siani-es/MultiFEBE



Outline

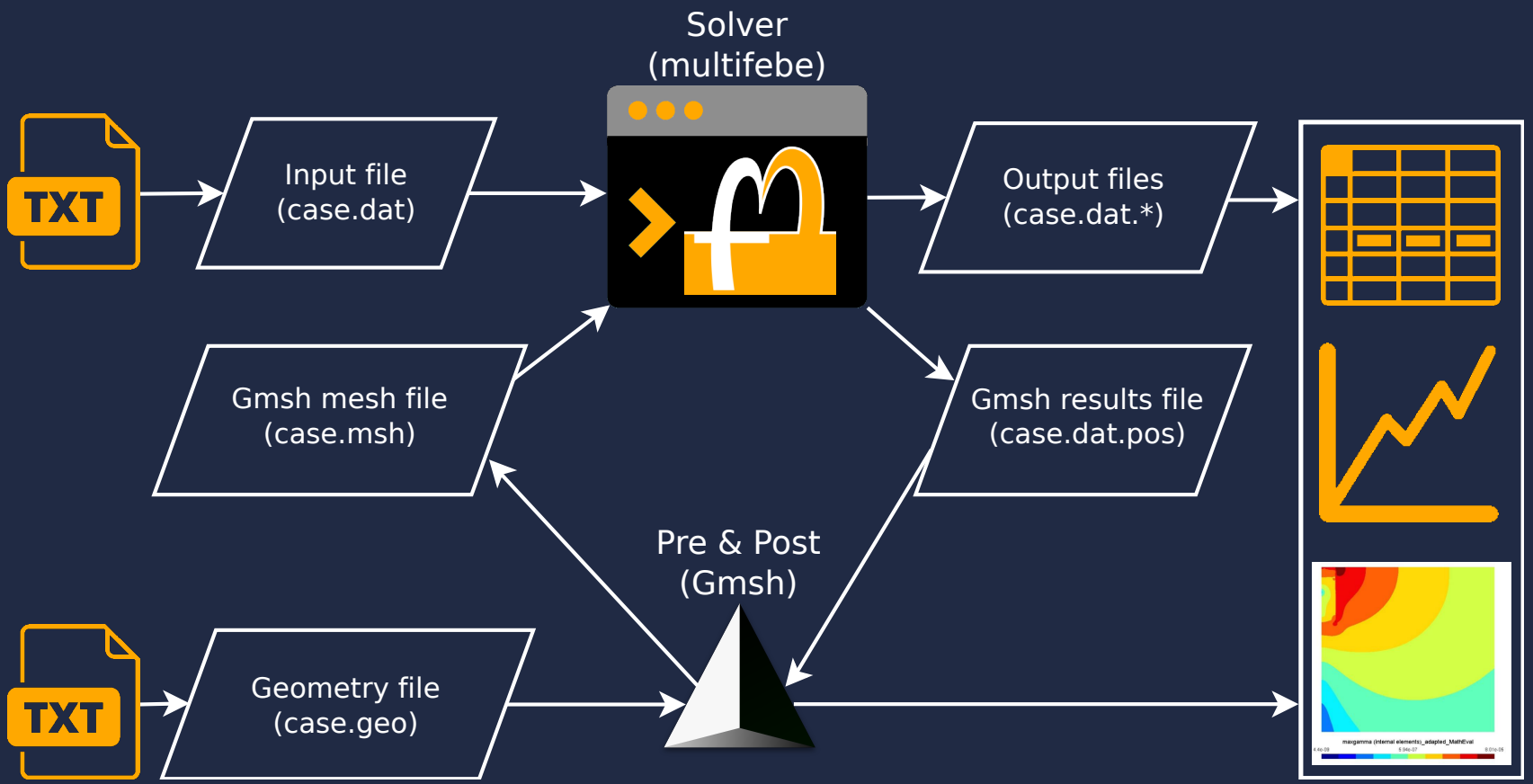
Design choices: USER perspective

MultiFEBE is a command-line solver:



Design choices: USER perspective

MultiFEBE is a command-line solver:



Design choices: DEVELOPER perspective

Programming languages:

- Fortran 2003 (use PP & some OOP)

```
22 program multifebe
23
24   use iso_fortran_env
25   use fbem_string_handling
26   use fbem_data_structures
27   use fbem_shape_functions
28   use problem_variables
29
30   implicit none
31   integer :: kf
32
33   call date_and_time(timestamp_date_start,timestamp_time_start)
34
35   call process_command_line_options
36
```

Design choices: DEVELOPER perspective

Programming languages:

- Fortran 2003 (use PP & some OOP)
- Fortran preprocessor (FPP)

```
4238          x1(1)=x1_i(1)+rho*costheta
4239          xi(2)=xi_i(2)+rho*sintheta
4240          ! XI->X transformation
4241          ! Geometrical shape functions and first derivatives at xi
4242 #        define etype type_g
4243 #        define delta 0.0d0
4244 #        define phi phi_g
4245 #        define dphidxi1 dphidxi1_g
4246 #        define dphidxi2 dphidxi2_g
4247 #        include <phi_and_dphidxik_2d.rc>
4248 #        undef etype
4249 #        undef delta
4250 #        undef phi
4251 #        undef dphidxi1
4252 #        undef dphidxi2
```

directives

Design choices: DEVELOPER perspective

Programming languages:

- Fortran 2003 (use PP & some OOP)
- Fortran preprocessor (FPP)
- OpenMP (shared-memory parallelism)

```
270  ! REGION BOUNDARIES
271  do kb_int=1,region(kr)%n_boundaries
272    sb_int=region(kr)%boundary(kb_int)
273    sb_int_reversion=region(kr)%boundary_reversion(kb_int)
274    sp_int=boundary(sb_int)%part
275    !$omp parallel do schedule (dynamic) default (shared) private (se_int,se_int_n_nodes
276    do ke_int=1,part(sp_int)%n_elements
277      se_int=part(sp_int)%element(ke_int)
278      se_int_n_nodes=element(se_int)%n_nodes
279      call build_lse_mechanics_bem_harela_element(omega,kr,sb_int,sb_int_reversion,se_ir
280    end do
281    !$omp end parallel do
282  end do
283  ! REGION BODY LOADS
284  do kb_int=1,region(kr)%n_be_bodyloads
```

Design choices: DEVELOPER perspective

Libraries:

- LAPACK (from OpenBLAS or ATLAS)

```
171  if (refine) then
172    allocate (bcopy(n_dof,n_rhs))
173    bcopy=b
174  end if
175  trans='N'
176  call zgetrs(trans,n_dof,n_rhs,A,n_dof,ipiv,b,n_dof,info)
177  if (info.eq.0) then
178    if (verbose_level.ge.2) then
179      call fbem_timestamp_w_message(output_unit,2,'Linear system of equations solved')
180    end if
181  else
182    call fbem_timestamp_message(output_unit,2)
183    write(output_unit,'(a13,i11)') 'zgetrs info: ', info
184    call fbem_error_message(error_unit,0,__FILE__,__LINE__,'solver has failed')
185  end if
```

Design choices: DEVELOPER perspective

Tools:

- 
  Compiler: GNU Fortran
- 
  Build/packaging automation tool: CMake
- 
  GNU/Linux envir. in Windows: MSYS2

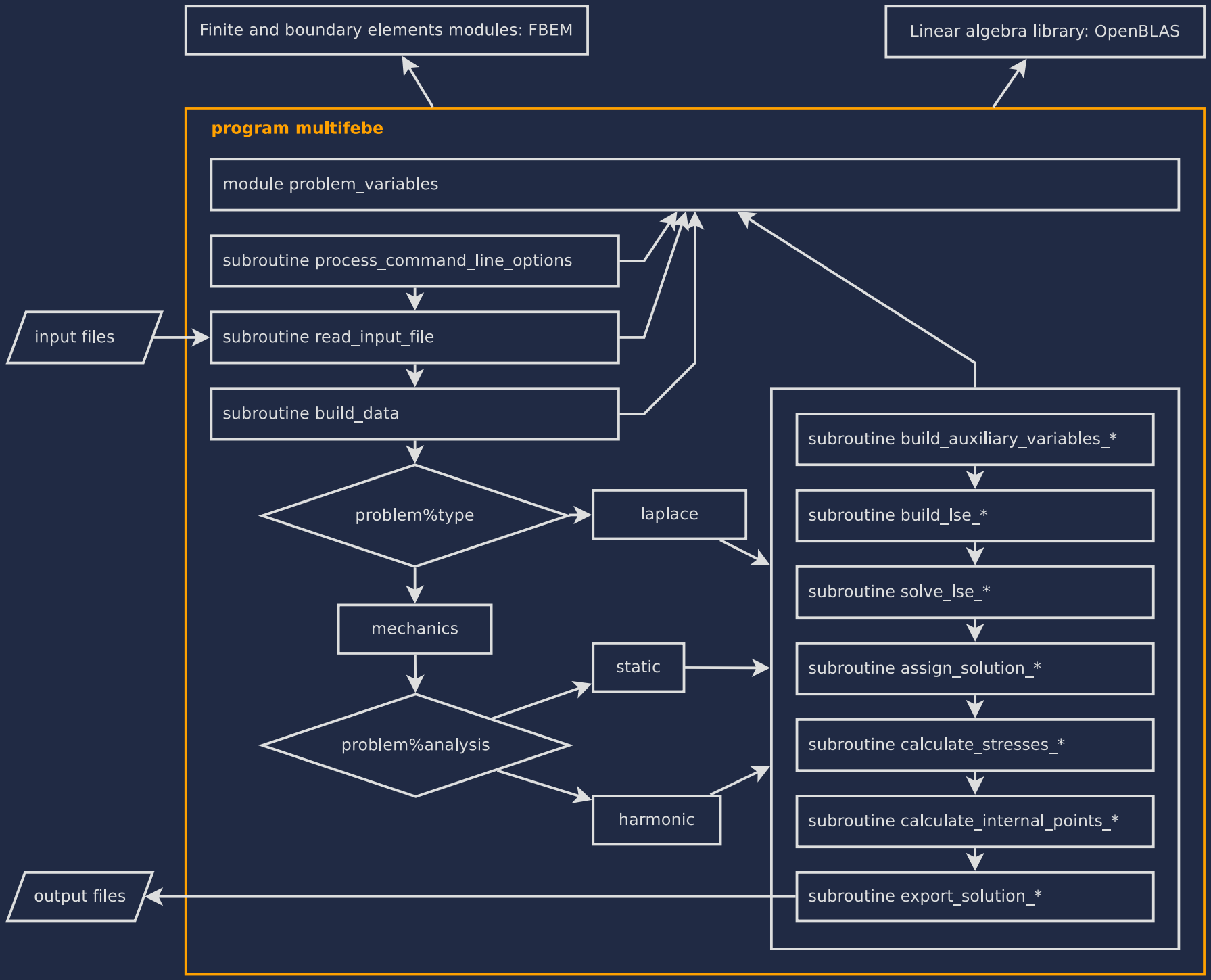
Operating systems (64-bits):

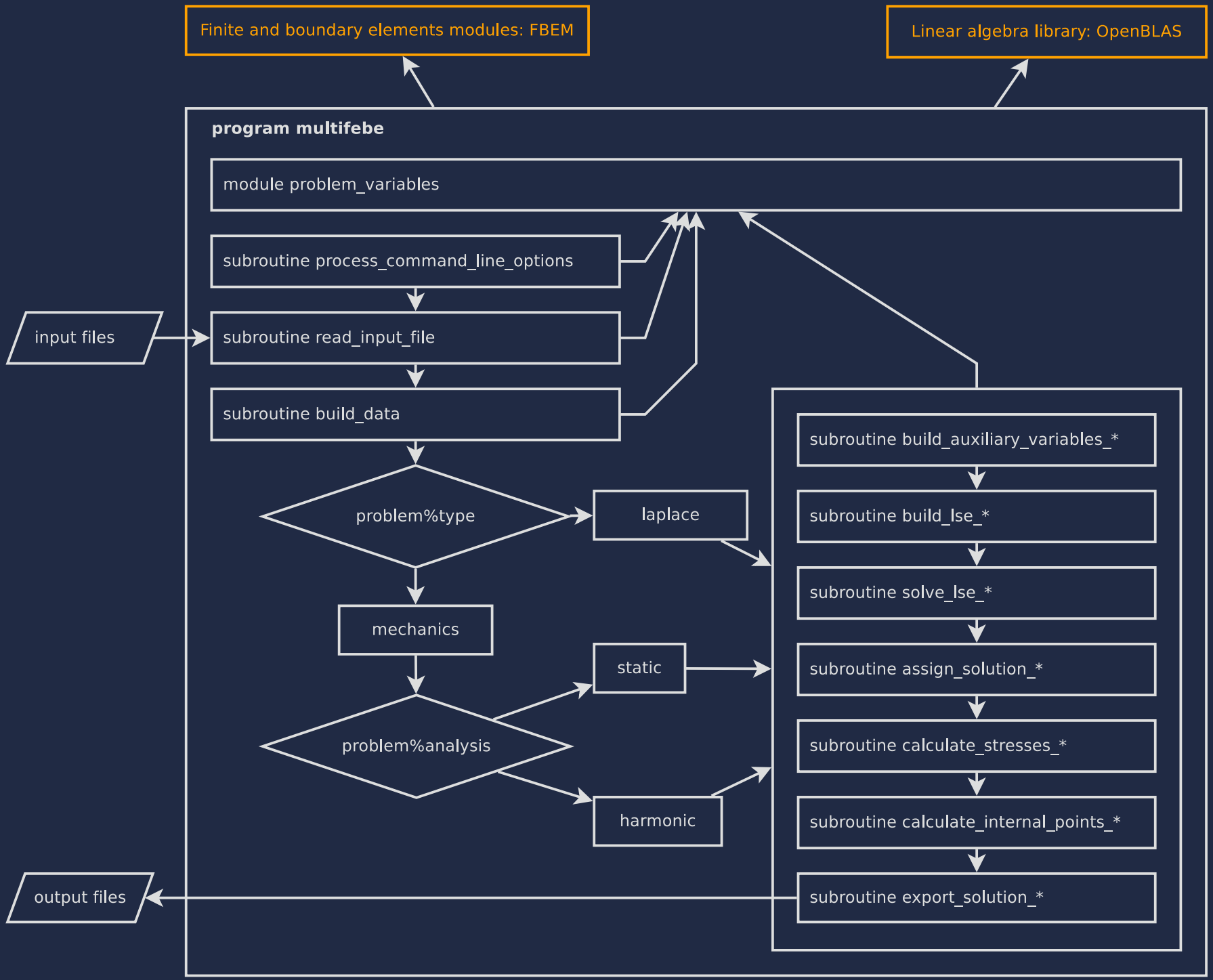
- 
  GNU/Linux
- 
  Windows

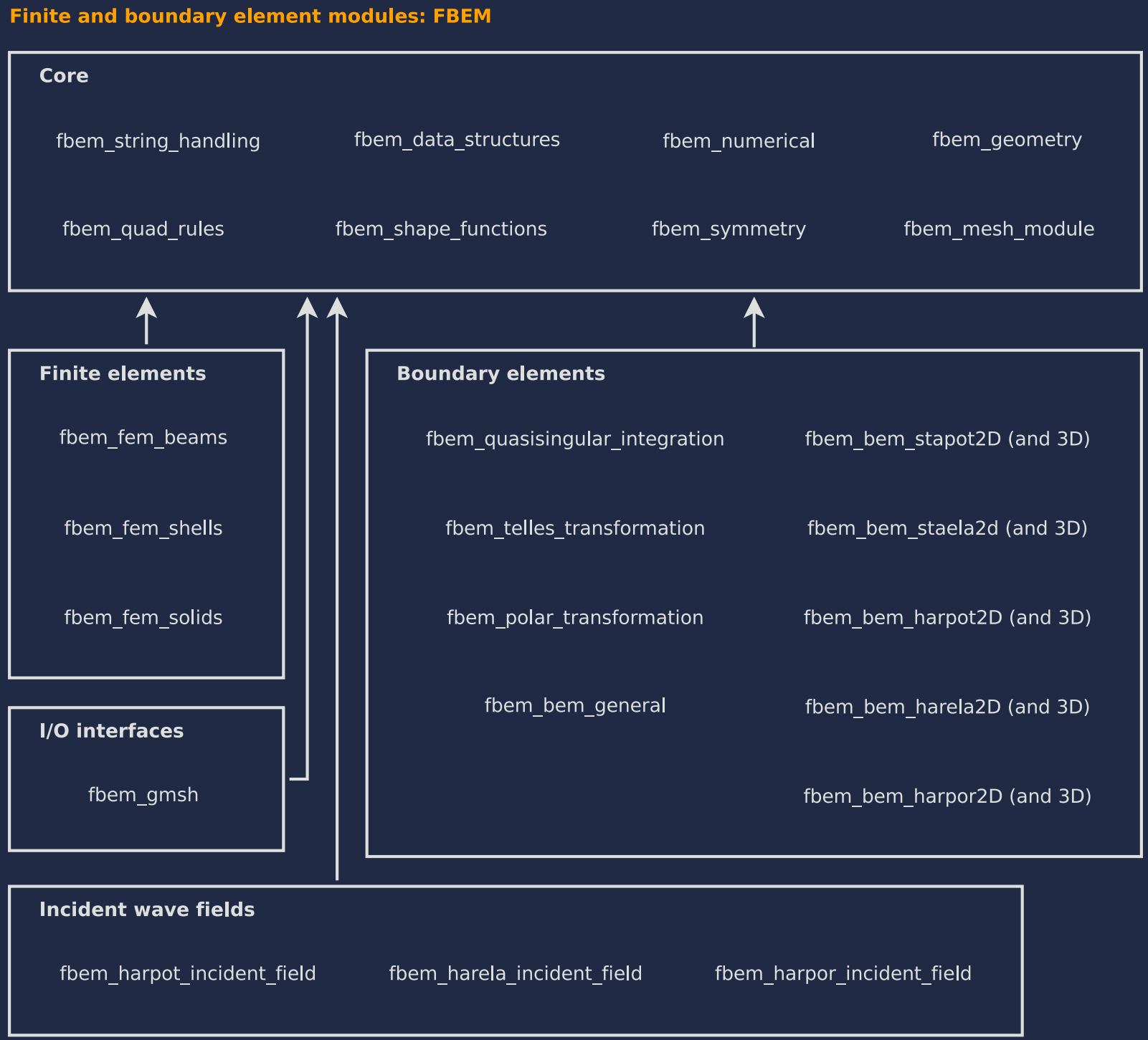
Design choices: DEVELOPER perspective

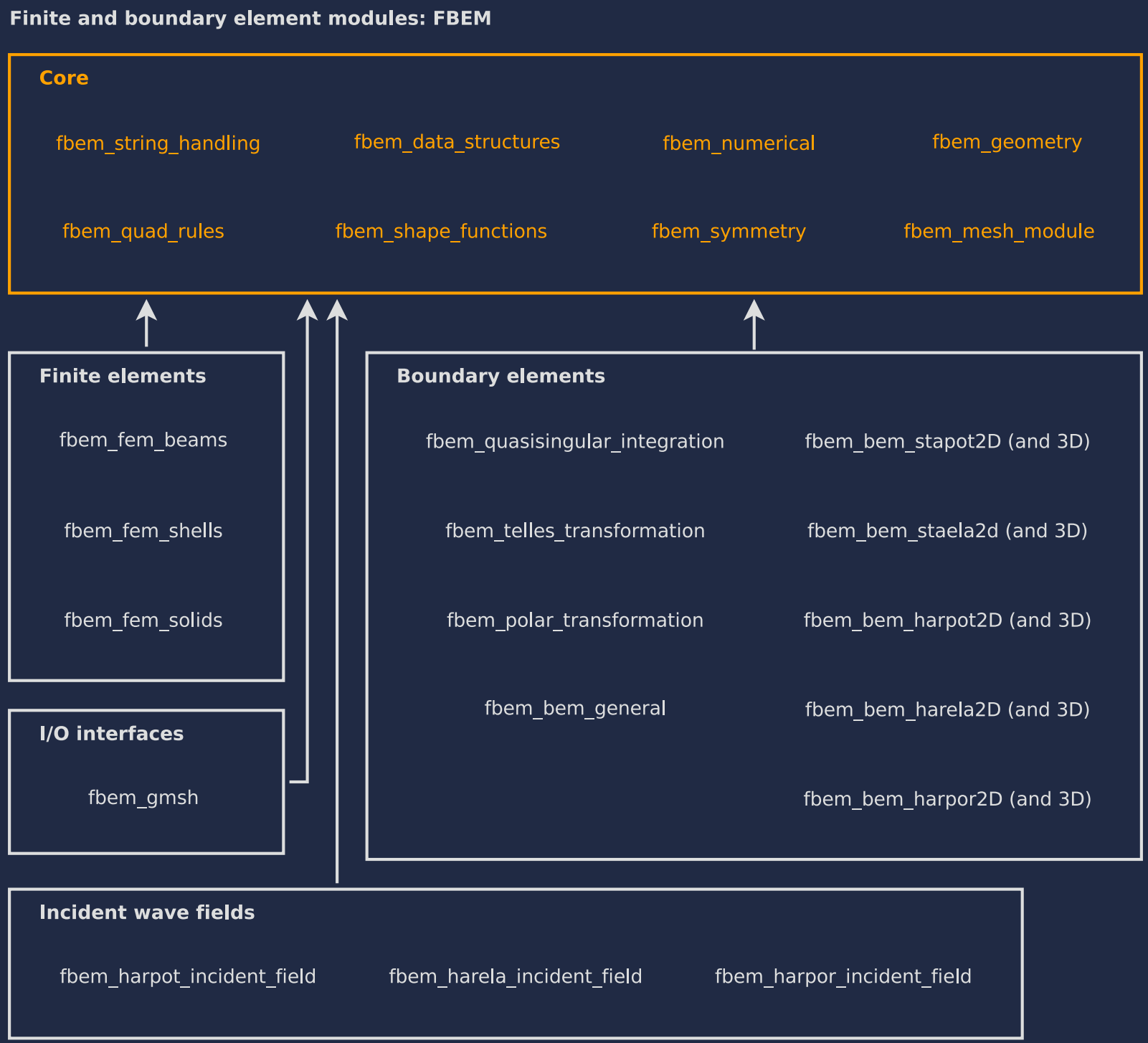
Arquitecture – a two-layer design:

- Application layer
- Computational layer

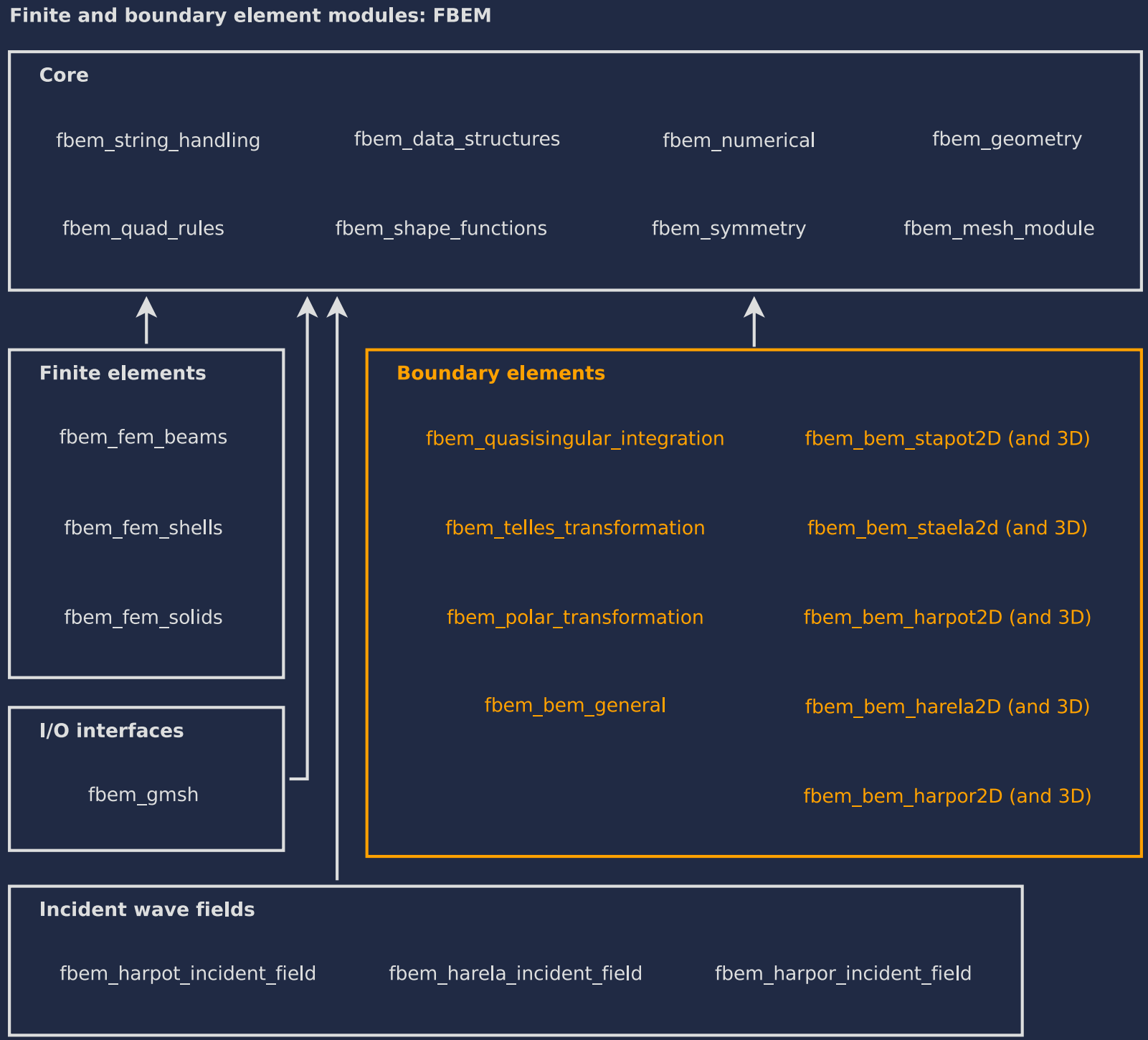

Finite and boundary elements modules: FBEM
Linear algebra library: OpenBLAS
program multifebe
module problem_variables
subroutine process_command_line_options
input files
subroutine read_input_file
subroutine build_data
problem%type
laplace
mechanics
static
problem%analysis
harmonic
output files
subroutine build_auxiliary_variables_*
subroutine build_lse_*
subroutine solve_lse_*
subroutine assign_solution_*
subroutine calculate_stresses_*
subroutine calculate_internal_points_*
subroutine export_solution_*

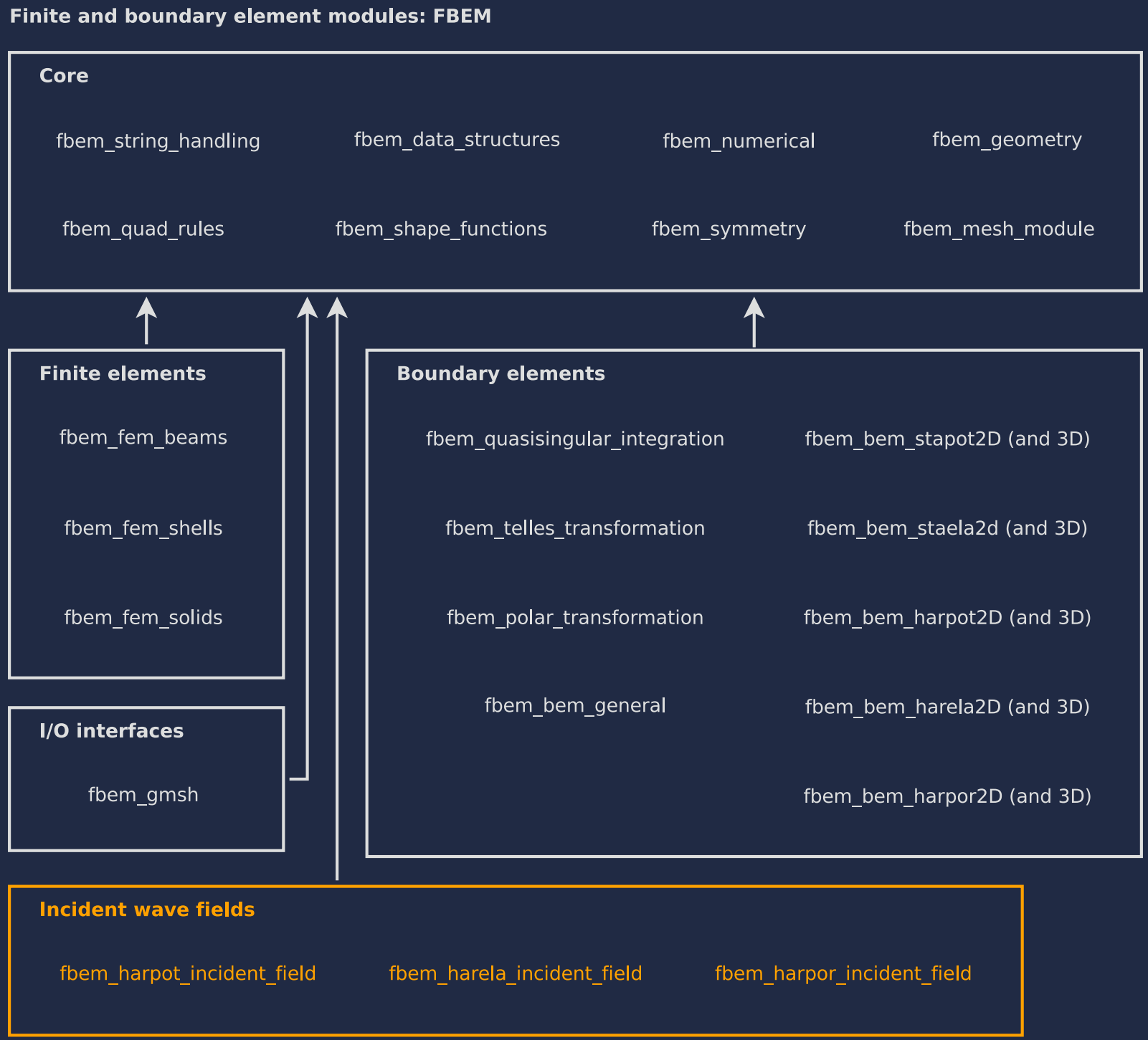

Finite and boundary elements modules: FBEM
Linear algebra library: OpenBLAS
program multifebe
module problem_variables
subroutine process_command_line_options
input files
subroutine read_input_file
subroutine build_data
problem%type
laplace
mechanics
static
problem%analysis
harmonic
output files
subroutine build_auxiliary_variables_*
subroutine build_lse_*
subroutine solve_lse_*
subroutine assign_solution_*
subroutine calculate_stresses_*
subroutine calculate_internal_points_*
subroutine export_solution_*

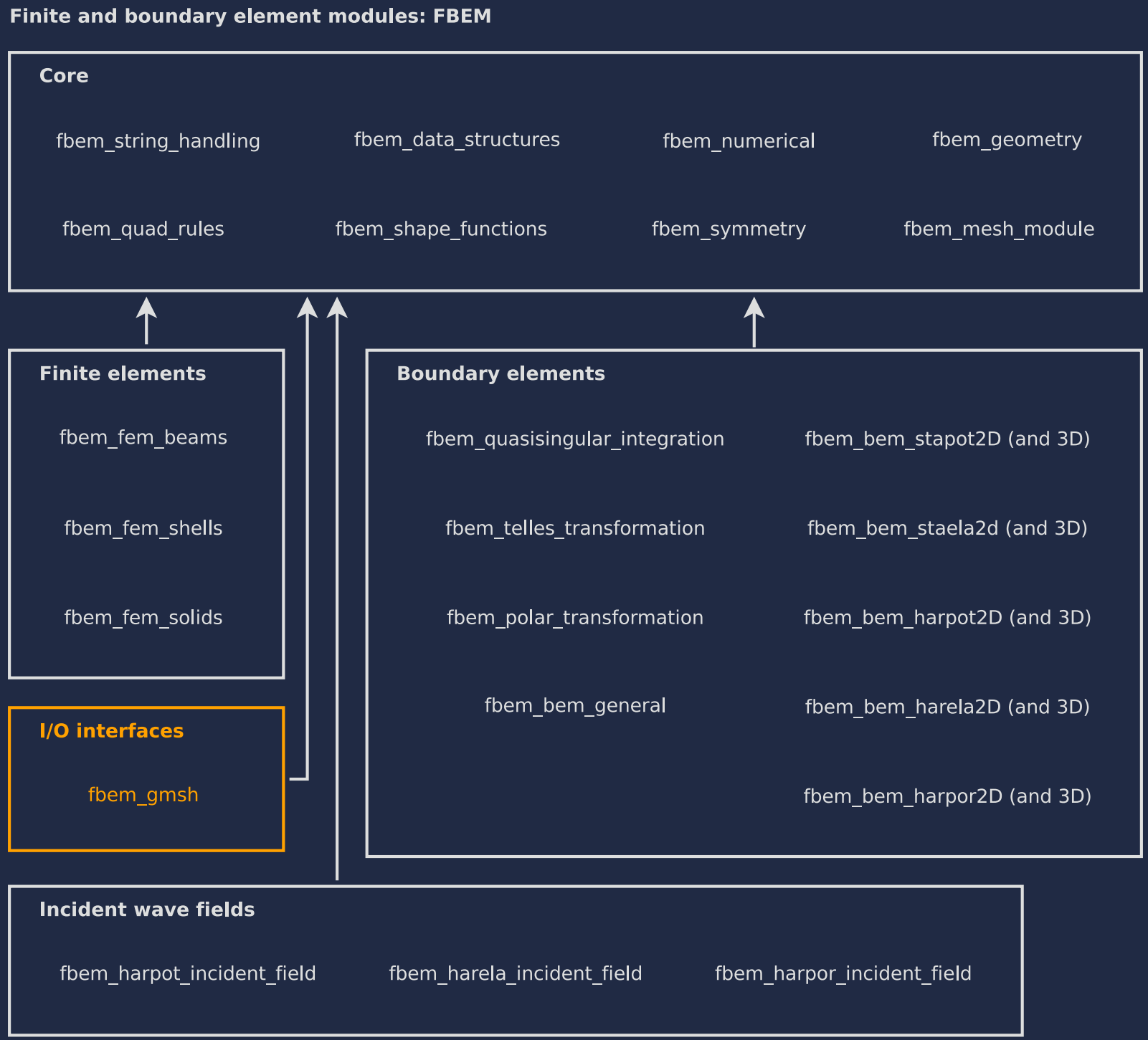

Finite and boundary element modules: FBEM
Core
fbem_string_handling
fbem_data_structures
fbem_numerical
fbem_geometry
fbem_quad_rules
fbem_shape_functions
fbem_symmetry
fbem_mesh_module
Finite elements
fbem_fem_beams
fbem_fem_shells
fbem_fem_solids
I/O interfaces
fbem_gmsh
Boundary elements
fbem_quasisingular_integration
fbem_bem_stapot2D (and 3D)
fbem_telles_transformation
fbem_bem_staela2d (and 3D)
fbem_polar_transformation
fbem_bem_harpot2D (and 3D)
fbem_bem_general
fbem_bem_harela2D (and 3D)
fbem_bem_harpor2D (and 3D)
Incident wave fields
fbem_harpot_incident_field
fbem_harela_incident_field
fbem_harpor_incident_field


Finite and boundary element modules: FBEM
Core
fbem_string_handling
fbem_data_structures
fbem_numerical
fbem_geometry
fbem_quad_rules
fbem_shape_functions
fbem_symmetry
fbem_mesh_module
Finite elements
fbem_fem_beams
fbem_fem_shells
fbem_fem_solids
I/O interfaces
fbem_gmsh
Boundary elements
fbem_quasisingular_integration
fbem_bem_stapot2D (and 3D)
fbem_telles_transformation
fbem_bem_staela2d (and 3D)
fbem_polar_transformation
fbem_bem_harpot2D (and 3D)
fbem_bem_general
fbem_bem_harela2D (and 3D)
fbem_bem_harpor2D (and 3D)
Incident wave fields
fbem_harpot_incident_field
fbem_harela_incident_field
fbem_harpor_incident_field


Finite and boundary element modules: FBEM
Core
fbem_string_handling
fbem_data_structures
fbem_numerical
fbem_geometry
fbem_quad_rules
fbem_shape_functions
fbem_symmetry
fbem_mesh_module
Finite elements
fbem_fem_beams
fbem_fem_shells
fbem_fem_solids
I/O interfaces
fbem_gmsh
Boundary elements
fbem_quasisingular_integration
fbem_bem_stapot2D (and 3D)
fbem_telles_transformation
fbem_bem_staela2d (and 3D)
fbem_polar_transformation
fbem_bem_harpot2D (and 3D)
fbem_bem_general
fbem_bem_harela2D (and 3D)
fbem_bem_harpor2D (and 3D)
Incident wave fields
fbem_harpot_incident_field
fbem_harela_incident_field
fbem_harpor_incident_field


Finite and boundary element modules: FBEM
Core
fbem_string_handling
fbem_data_structures
fbem_numerical
fbem_geometry
fbem_quad_rules
fbem_shape_functions
fbem_symmetry
fbem_mesh_module
Finite elements
fbem_fem_beams
fbem_fem_shells
fbem_fem_solids
I/O interfaces
fbem_gmsh
Boundary elements
fbem_quasisingular_integration
fbem_bem_stapot2D (and 3D)
fbem_telles_transformation
fbem_bem_staela2d (and 3D)
fbem_polar_transformation
fbem_bem_harpot2D (and 3D)
fbem_bem_general
fbem_bem_harela2D (and 3D)
fbem_bem_harpor2D (and 3D)
Incident wave fields
fbem_harpot_incident_field
fbem_harela_incident_field
fbem_harpor_incident_field

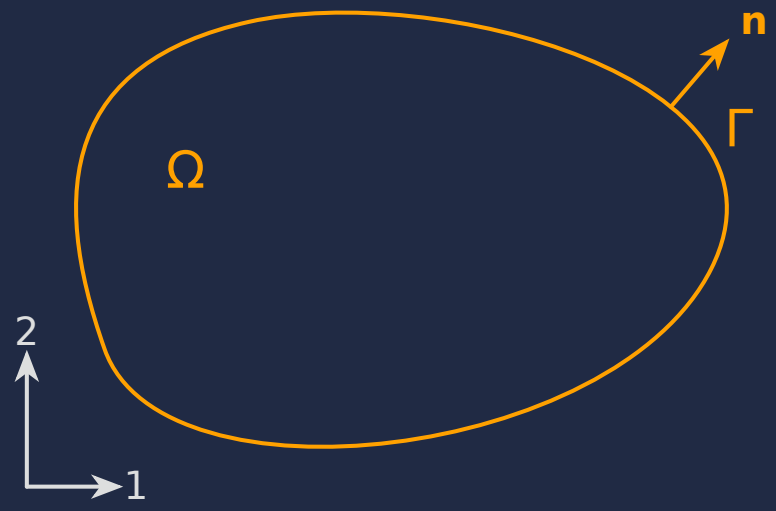

Finite and boundary element modules: FBEM
Core
fbem_string_handling
fbem_data_structures
fbem_numerical
fbem_geometry
fbem_quad_rules
fbem_shape_functions
fbem_symmetry
fbem_mesh_module
Finite elements
fbem_fem_beams
fbem_fem_shells
fbem_fem_solids
I/O interfaces
fbem_gmsh
Boundary elements
fbem_quasisingular_integration
fbem_bem_stapot2D (and 3D)
fbem_telles_transformation
fbem_bem_staela2d (and 3D)
fbem_polar_transformation
fbem_bem_harpot2D (and 3D)
fbem_bem_general
fbem_bem_harela2D (and 3D)
fbem_bem_harpor2D (and 3D)
Incident wave fields
fbem_harpot_incident_field
fbem_harela_incident_field
fbem_harpor_incident_field

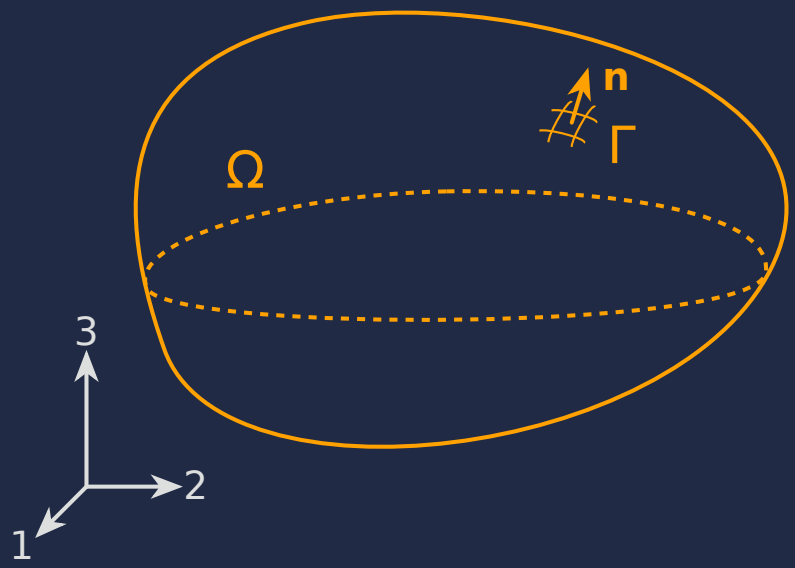

Finite and boundary element modules: FBEM
Core
fbem_string_handling
fbem_data_structures
fbem_numerical
fbem_geometry
fbem_quad_rules
fbem_shape_functions
fbem_symmetry
fbem_mesh_module
Finite elements
fbem_fem_beams
fbem_fem_shells
fbem_fem_solids
I/O interfaces
fbem_gmsh
Boundary elements
fbem_quasisingular_integration
fbem_bem_stapot2D (and 3D)
fbem_telles_transformation
fbem_bem_staela2d (and 3D)
fbem_polar_transformation
fbem_bem_harpot2D (and 3D)
fbem_bem_general
fbem_bem_harela2D (and 3D)
fbem_bem_harpor2D (and 3D)
Incident wave fields
fbem_harpot_incident_field
fbem_harela_incident_field
fbem_harpor_incident_field

Outline

Methodologies: types of analyses

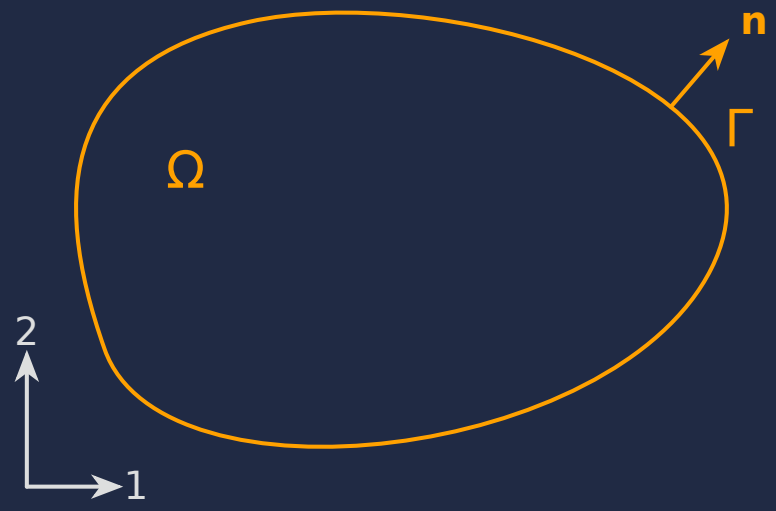
- 2D and 3D

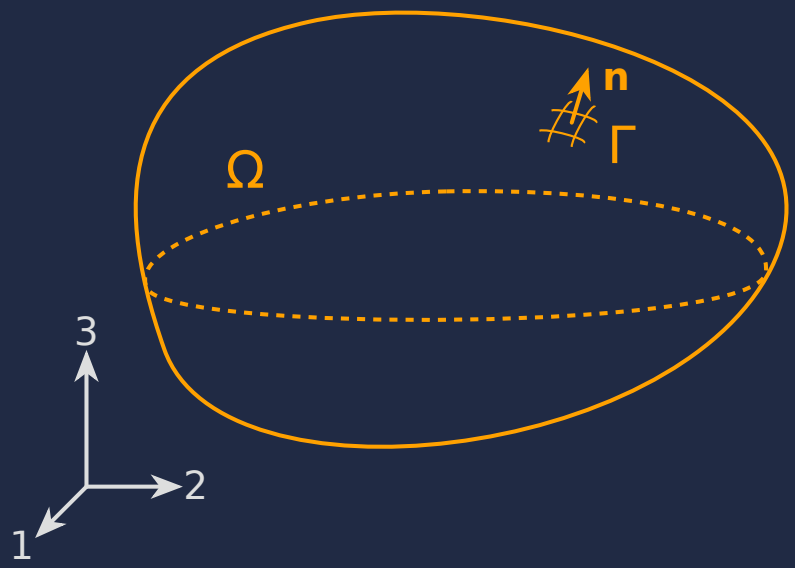




Methodologies: types of analyses

- 2D and 3D
- Static analysis and time harmonic analysis

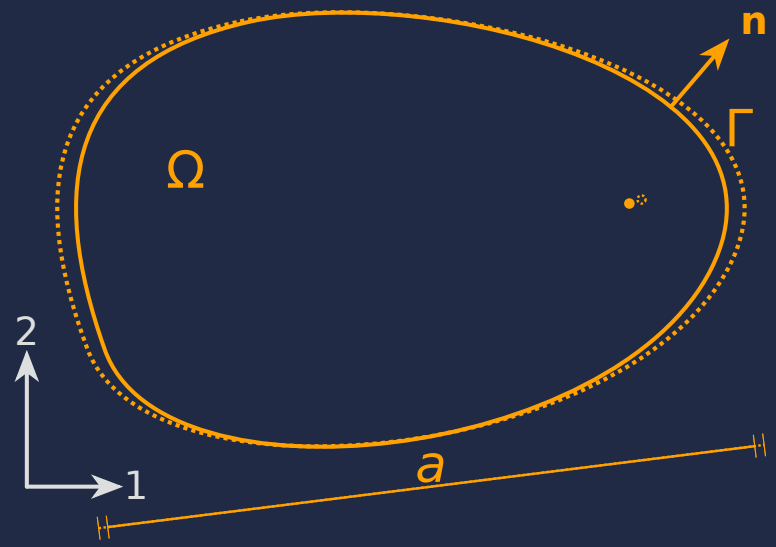


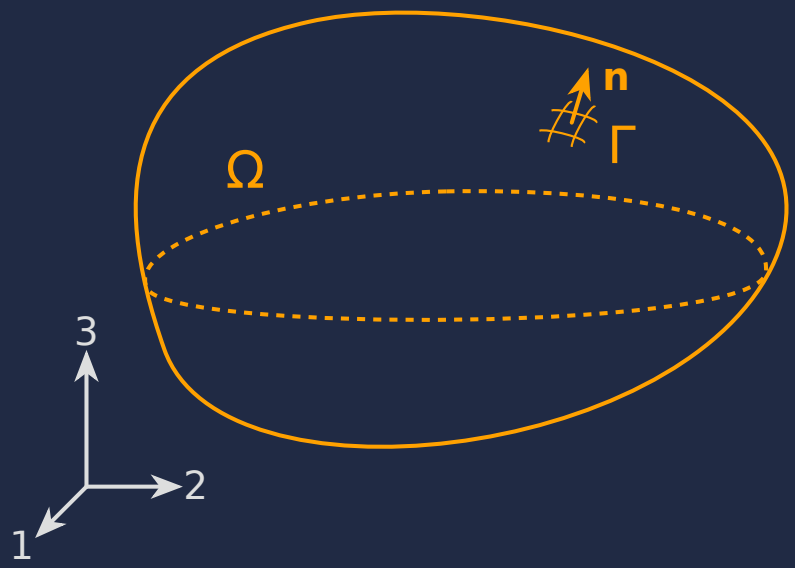


$$\mathbf{u}(\mathbf{x}) \quad \text{or} \quad \mathbf{u}(\mathbf{x}, t) = \tilde{\mathbf{u}}(\mathbf{x}, \omega) e^{i\omega t}$$

Methodologies: types of analyses

- 2D and 3D
- Static analysis and time harmonic analysis
- First-order geometric sensitivity analysis (limited)





$$\mathbf{u}(\mathbf{x}, a) = \mathbf{u}(\mathbf{x}, a_0) + \left(\frac{\partial \mathbf{u}}{\partial a}\right)_{a=a_0} (a - a_0) + \text{higher order terms}$$

Methodologies: Finite Element Method

- Discrete: mass, springs, dashpots.
- Beam elements:
  - Straight with 2 and 3 nodes (Euler-Bernoulli and Tim.[3])
  - Deg. from solid curved with 3 and 4 nodes (Tim.)
- Shell elements (Reissner-Mindlin):
  - Deg. from solid with 3, 4, 6, 8 and 9 nodes
  - MITC4[4] and MITC9[5] quadrilaterals (defect-free)
  - Automatic handling of 5 or 6 DOF nodes

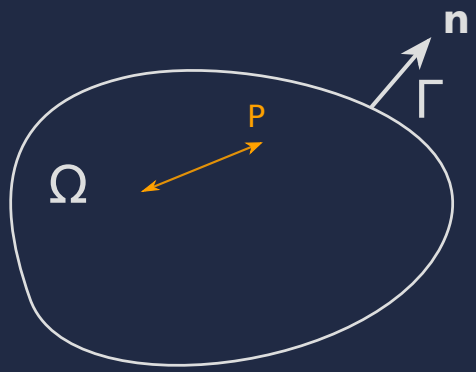
[3] Based on: Friedman & Kosmatka, Comp & Struct 47, 1993

[4] Dvorkin & Bathe, Eng Comput 1, 1984.

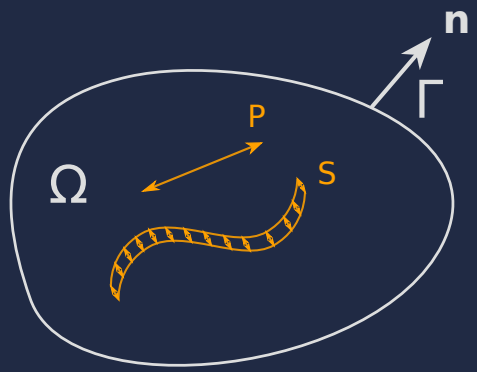
[5] Bucalem & Bathe, Int J Num Meth Eng 36, 1993.

Methodologies: Boundary Element Method

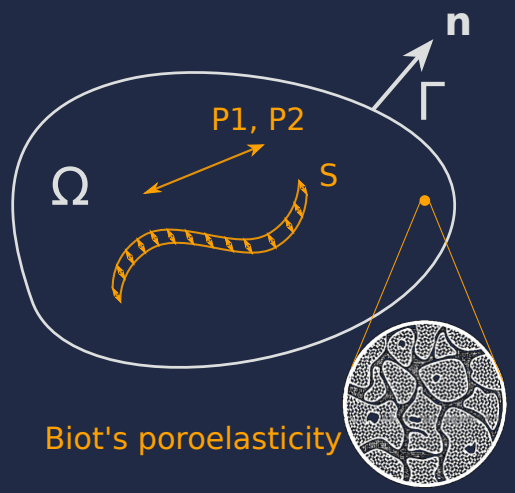
- Material models



Inviscid fluid



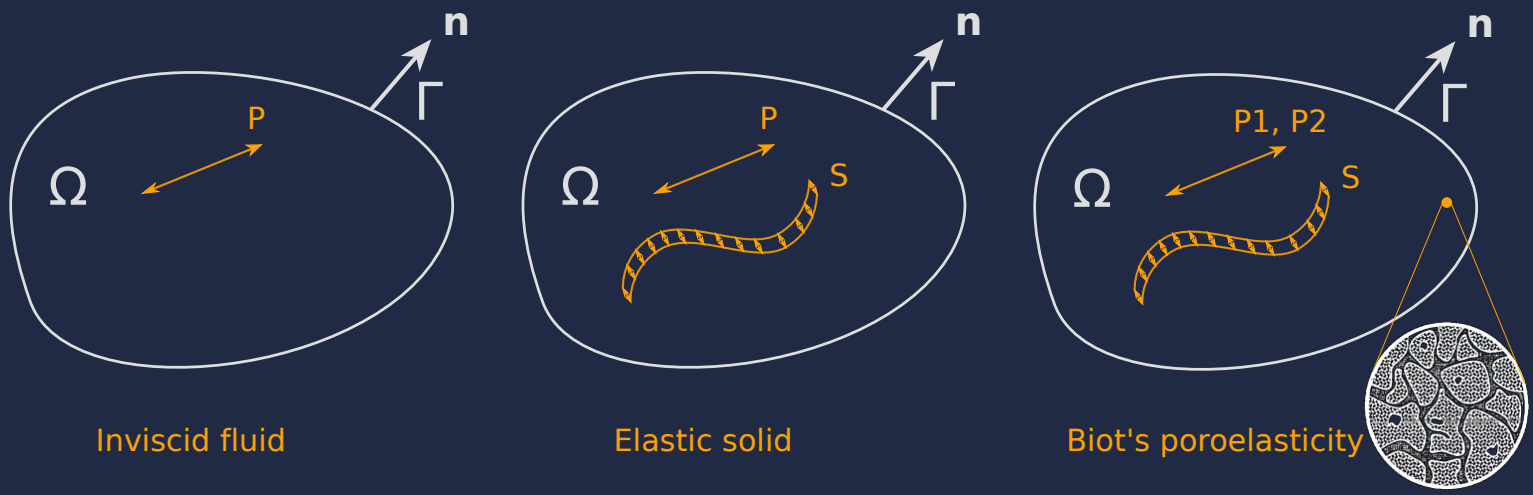
Elastic solid



Biot's poroelasticity

Methodologies: Boundary Element Method

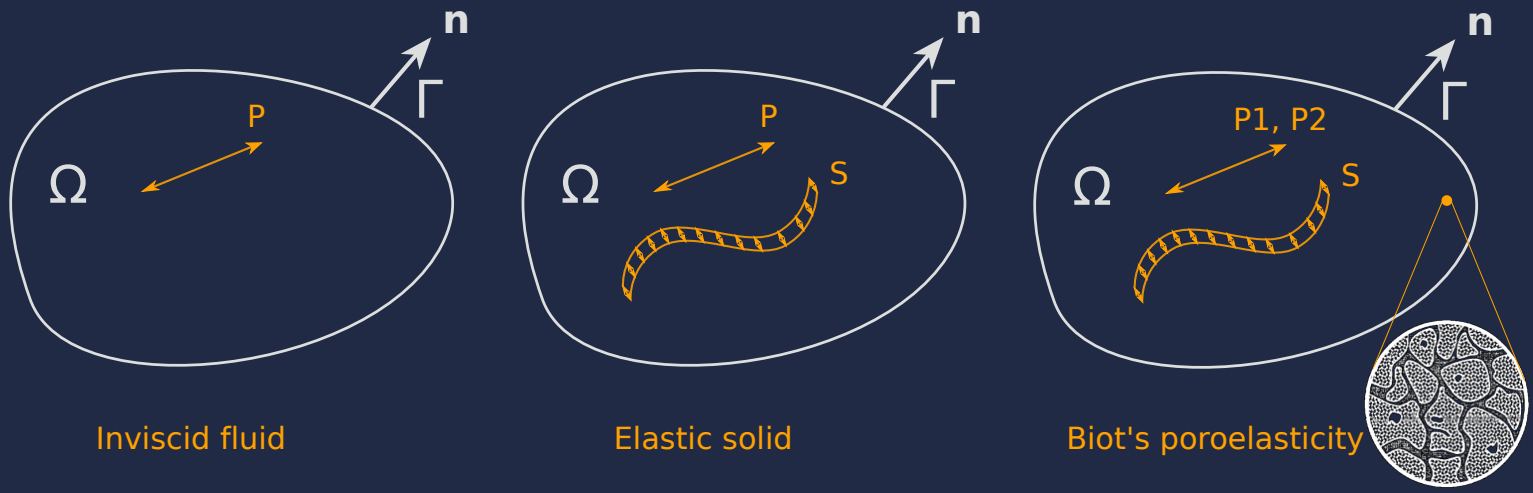
- Material models



- Available Green's functions (u^*_{lk}, t^*_{lk}, d^*_{lk}, s^*_{lk}):
  - Fundamental solutions for all material models
  - Half-space GF for inviscid fluid
  - Half-space GF for 2D/3D elastostatics

Methodologies: Boundary Element Method

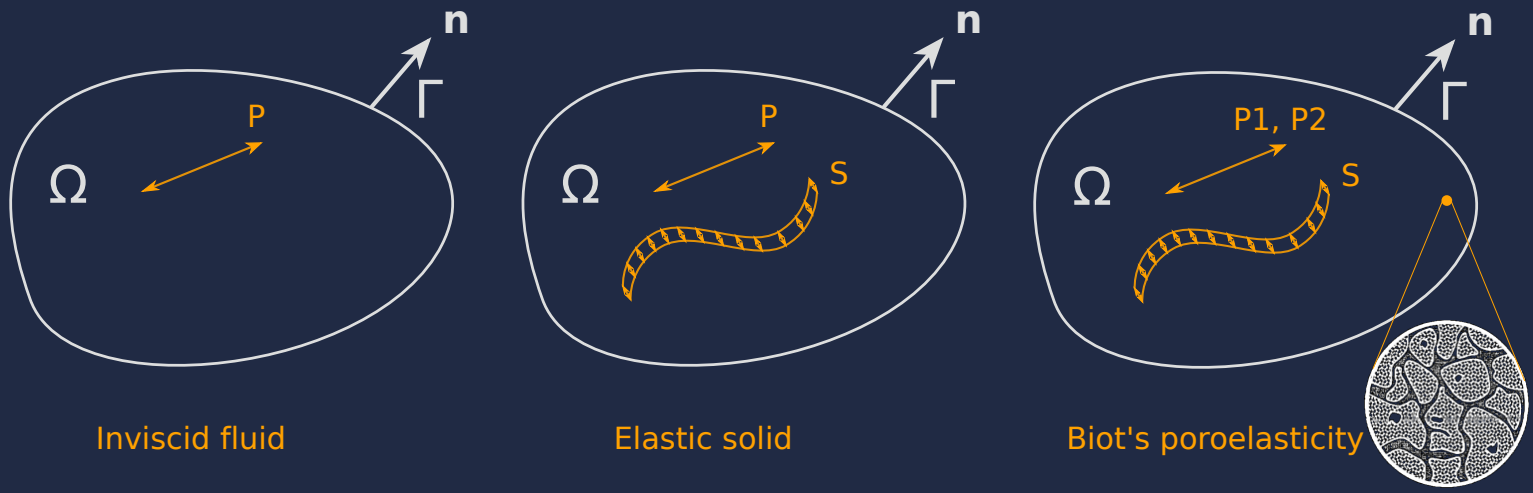
- Material models



- Available Green's functions (u^*_{lk}, t^*_{lk}, d^*_{lk}, s^*_{lk}):
  - Fundamental solutions for all material models
  - Half-space GF for inviscid fluid
  - Half-space GF for 2D/3D elastostatics
- Incident fields (point sources, P, S and R plane waves)

Methodologies: Boundary Element Method

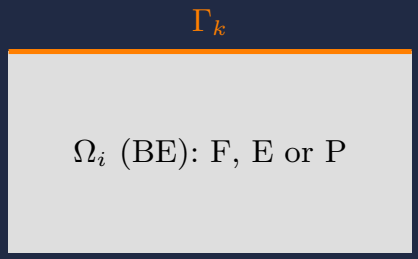
- Material models



- Available Green's functions (u^*_{lk}, t^*_{lk}, d^*_{lk}, s^*_{lk}):
  - Fundamental solutions for all material models
  - Half-space GF for inviscid fluid
  - Half-space GF for 2D/3D elastostatics
- Incident fields (point sources, P, S and R plane waves)
- Mirror symmetry

Methodologies: Boundary Element Method

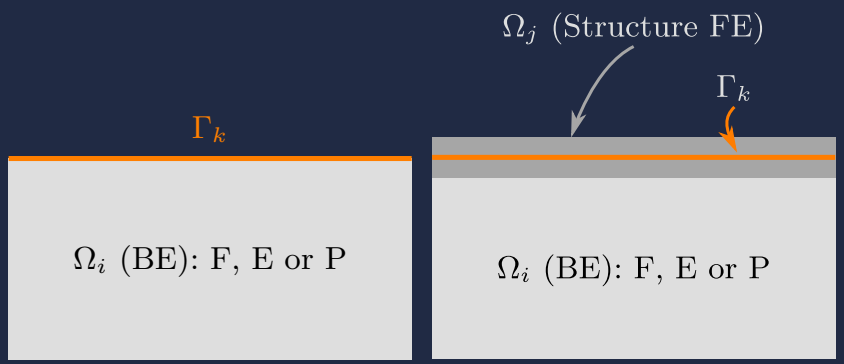
Figure: Inviscid fluid (F), elastic solid (E), poroelastic medium (P).



(a) Ordinary BE

Methodologies: Boundary Element Method

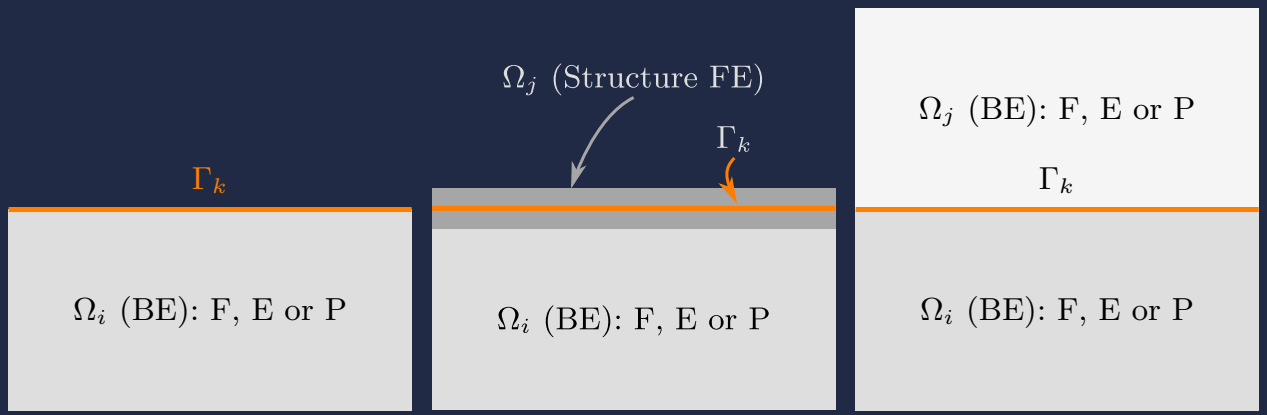
Figure: Inviscid fluid (F), elastic solid (E), poroelastic medium (P).



(a) Ordinary BE (b) Ordinary BE-FE

Methodologies: Boundary Element Method

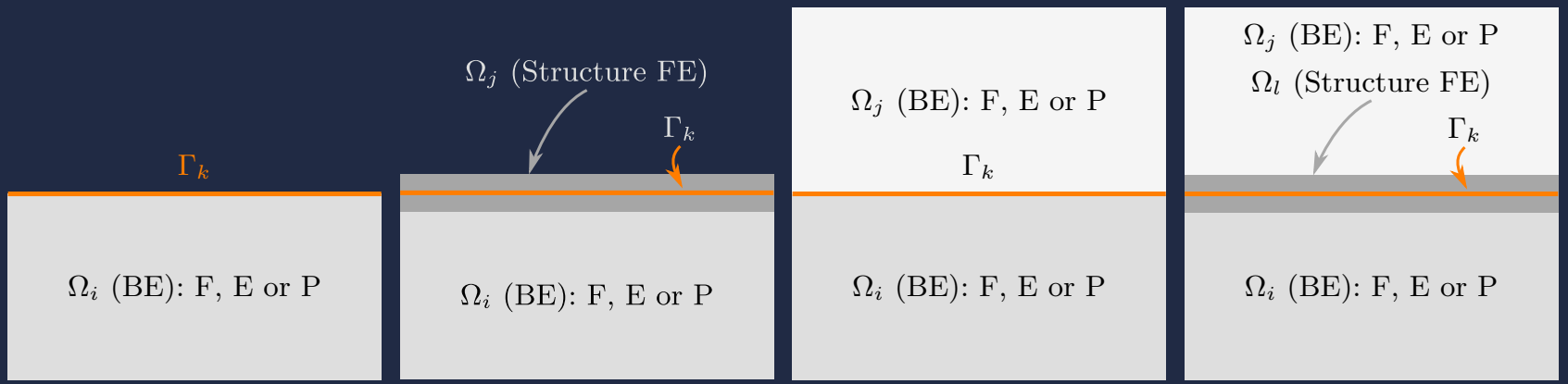
Figure: Inviscid fluid (F), elastic solid (E), poroelastic medium (P).



(a) Ordinary BE (b) Ordinary BE-FE (c) BE-BE interface

Methodologies: Boundary Element Method

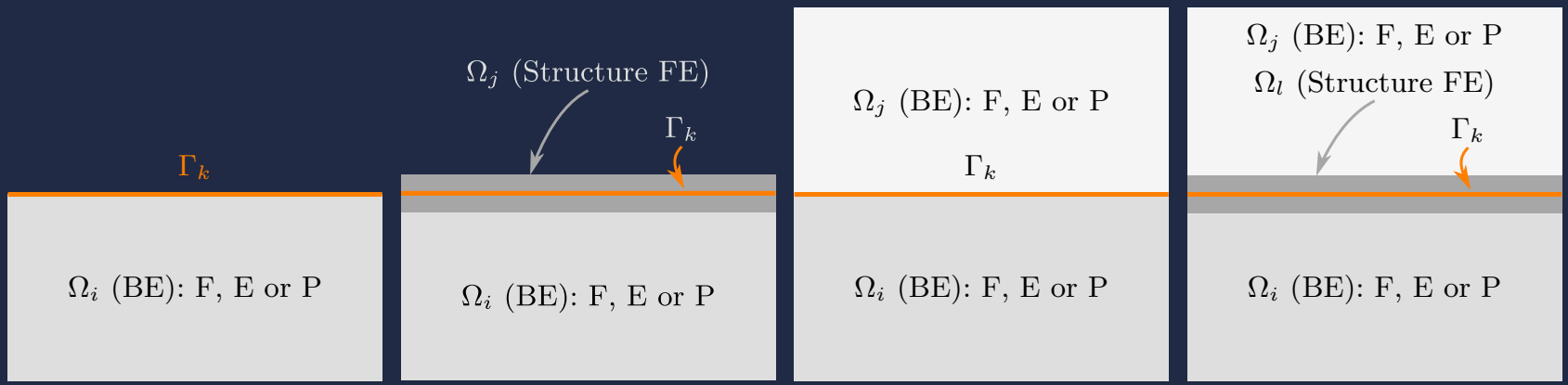
Figure: Inviscid fluid (F), elastic solid (E), poroelastic medium (P).



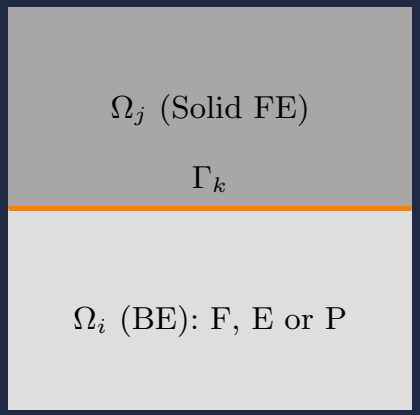
(a) Ordinary BE (b) Ordinary BE-FE (c) BE-BE interface (d) BE-FE-BE

Methodologies: Boundary Element Method

Figure: Inviscid fluid (F), elastic solid (E), poroelastic medium (P).



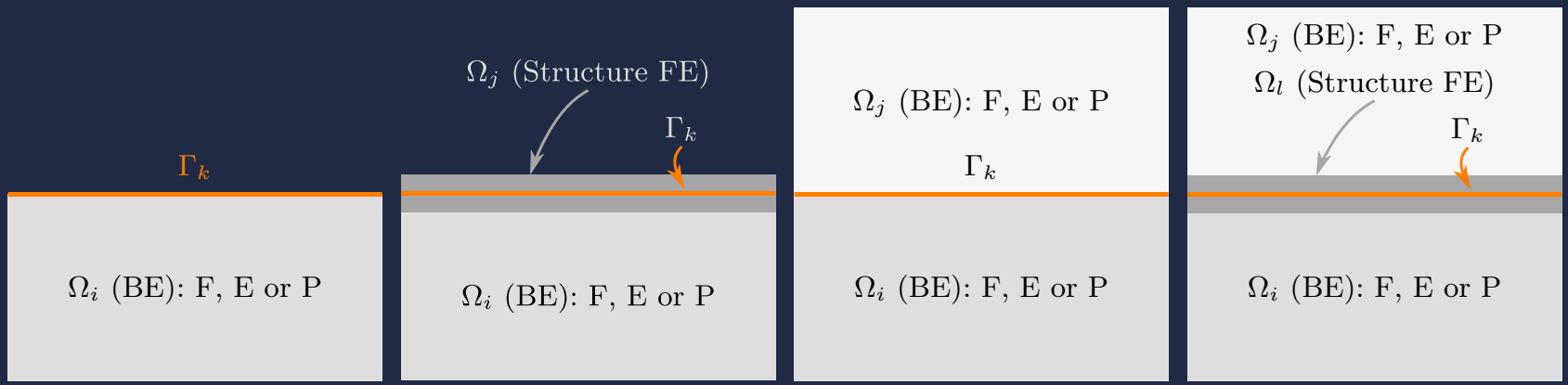
(a) Ordinary BE (b) Ordinary BE-FE (c) BE-BE interface (d) BE-FE-BE



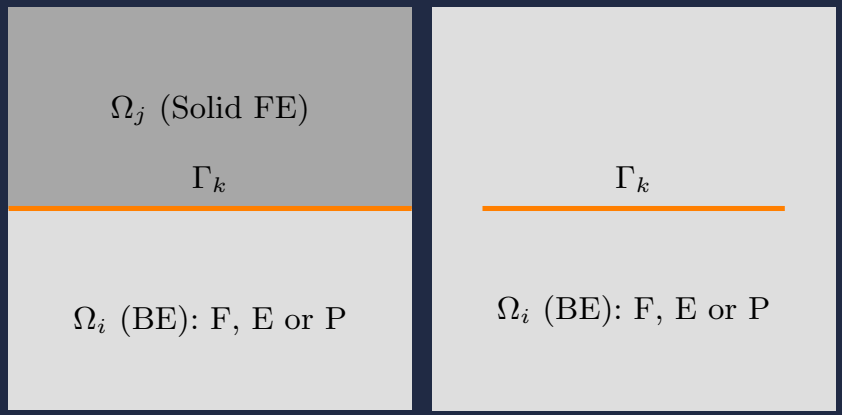
(e) Ordinary BE-FE

Methodologies: Boundary Element Method

Figure: Inviscid fluid (F), elastic solid (E), poroelastic medium (P).



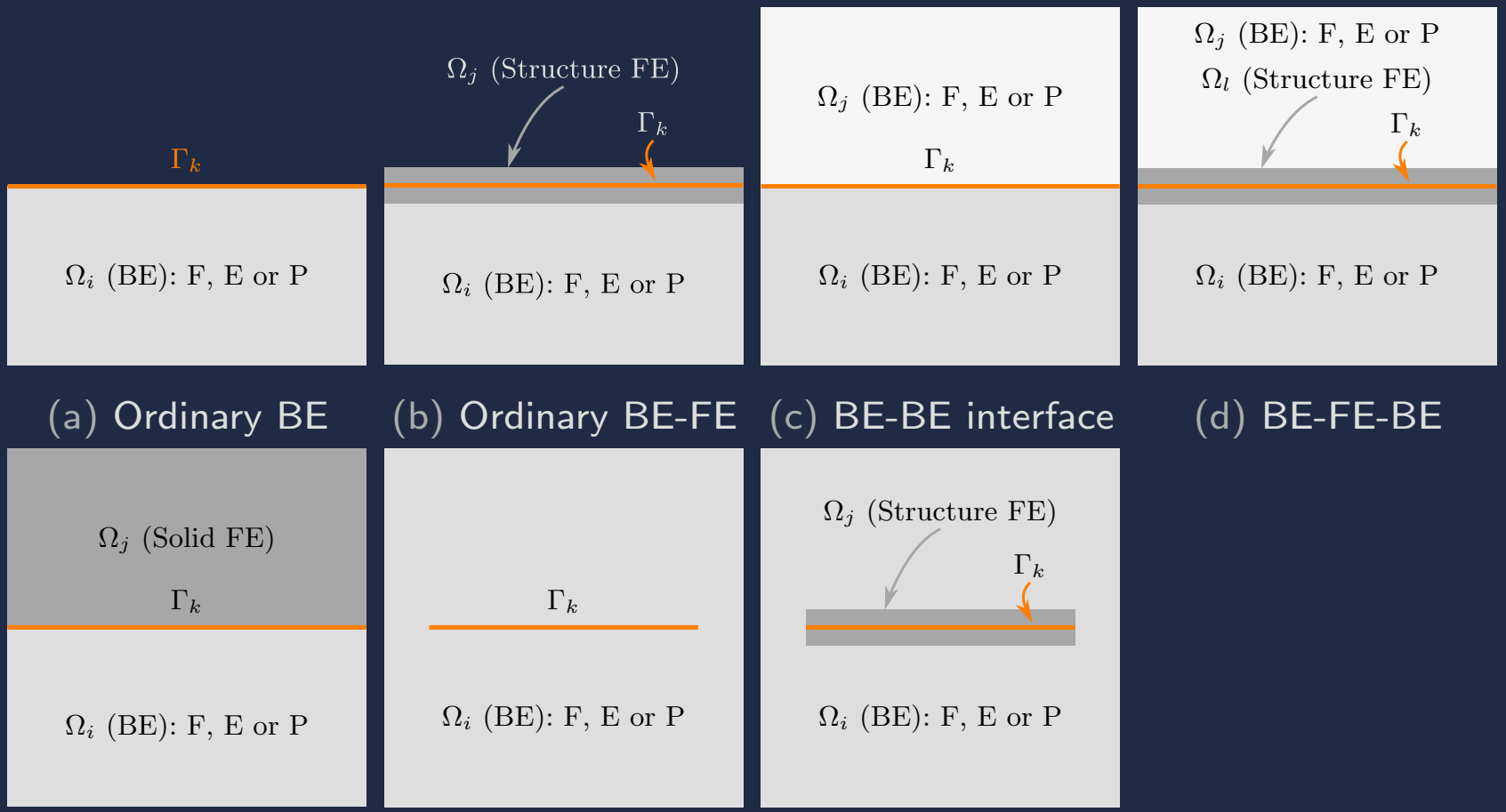
(a) Ordinary BE (b) Ordinary BE-FE (c) BE-BE interface (d) BE-FE-BE



(e) Ordinary BE-FE (f) Crack BE

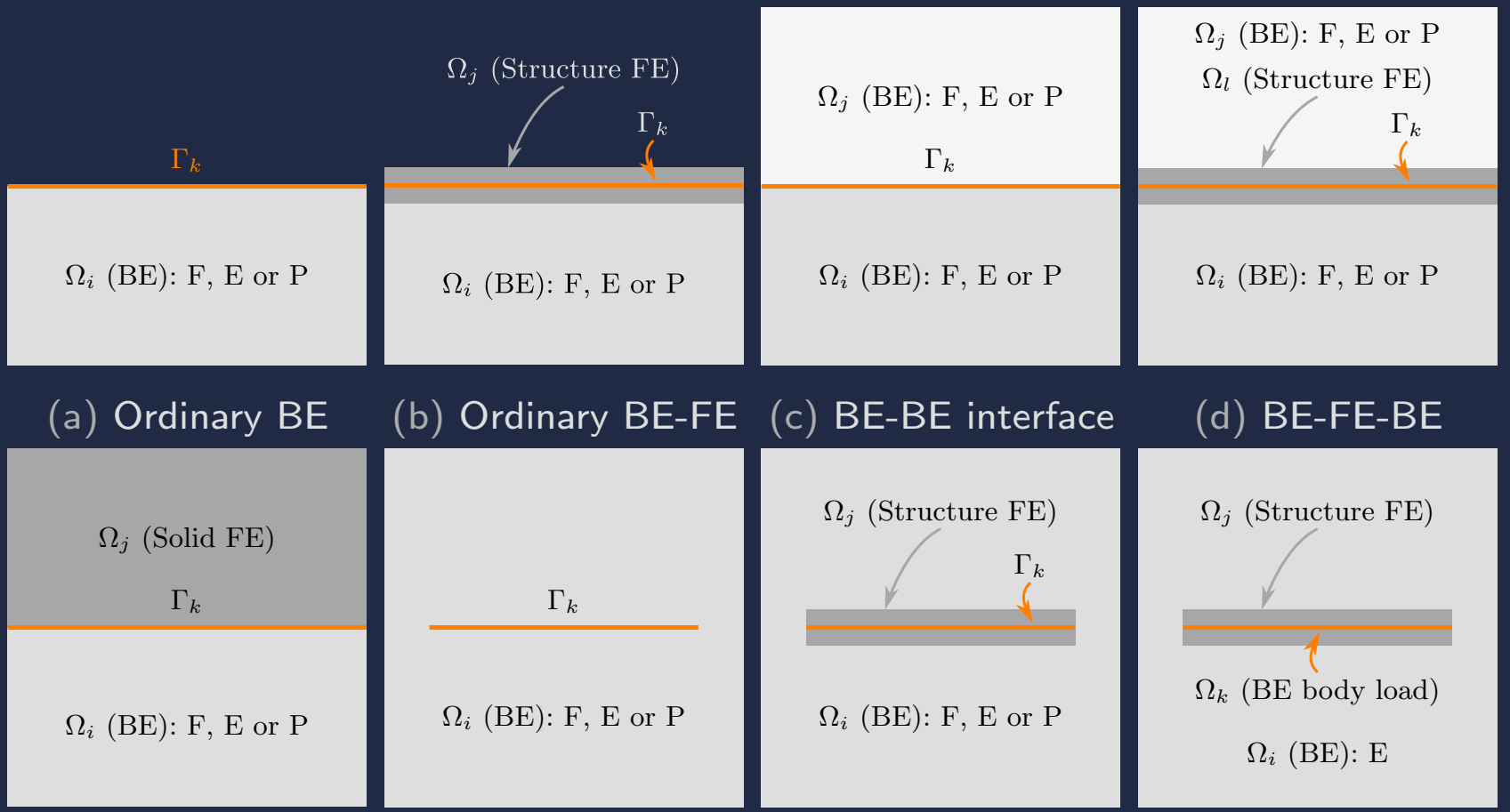
Methodologies: Boundary Element Method

Figure: Inviscid fluid (F), elastic solid (E), poroelastic medium (P).



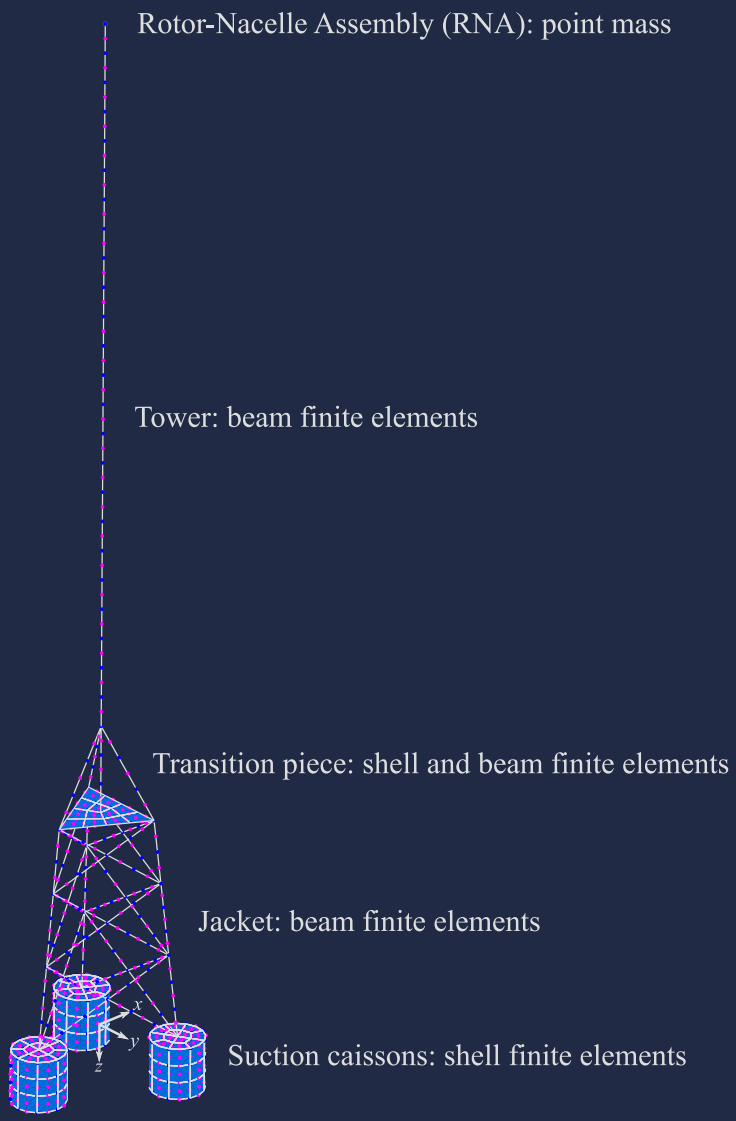
Methodologies: Boundary Element Method

Figure: Inviscid fluid (F), elastic solid (E), poroelastic medium (P).



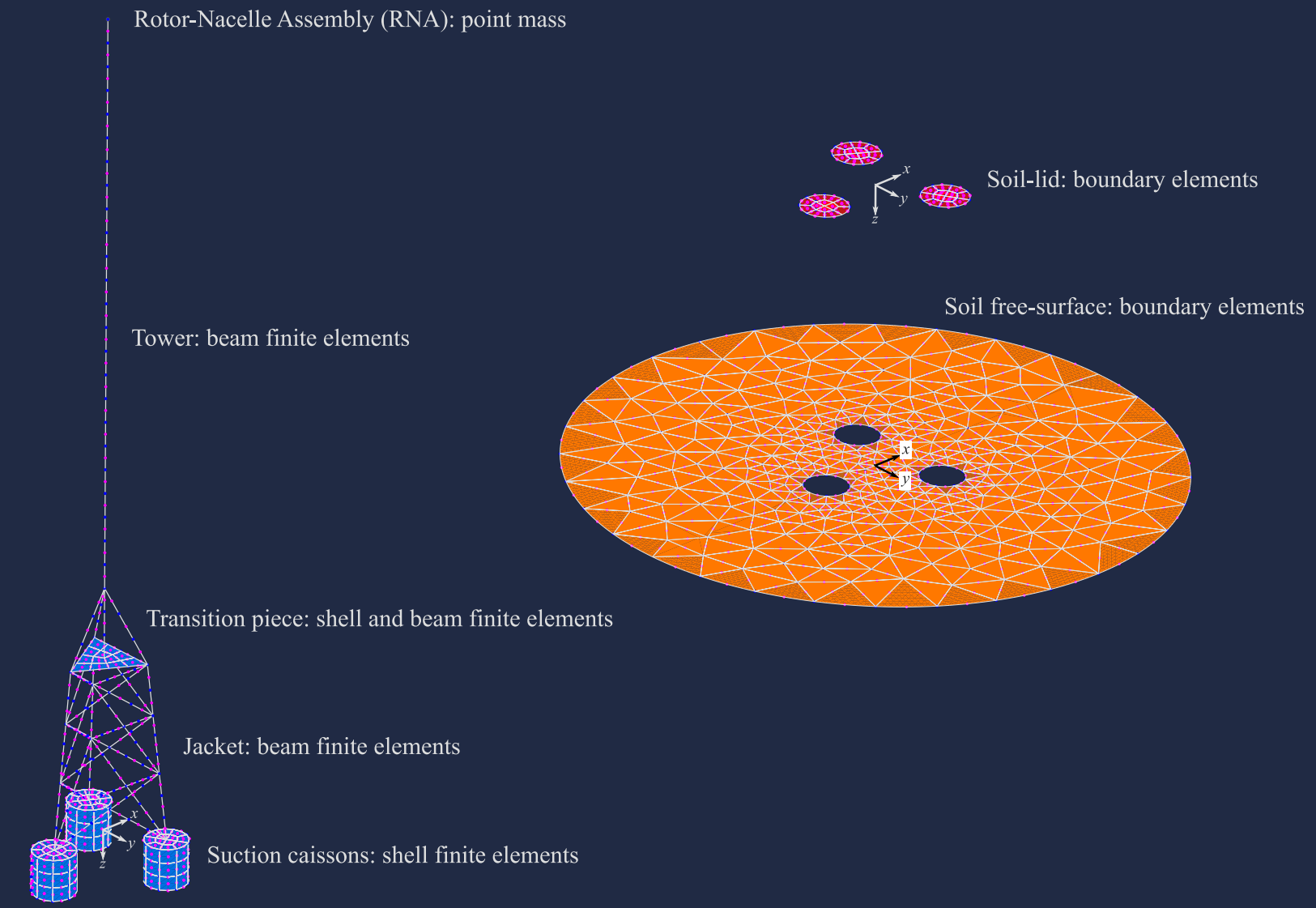
Outline

Offshore Wind Turbine



Finite Element Mesh

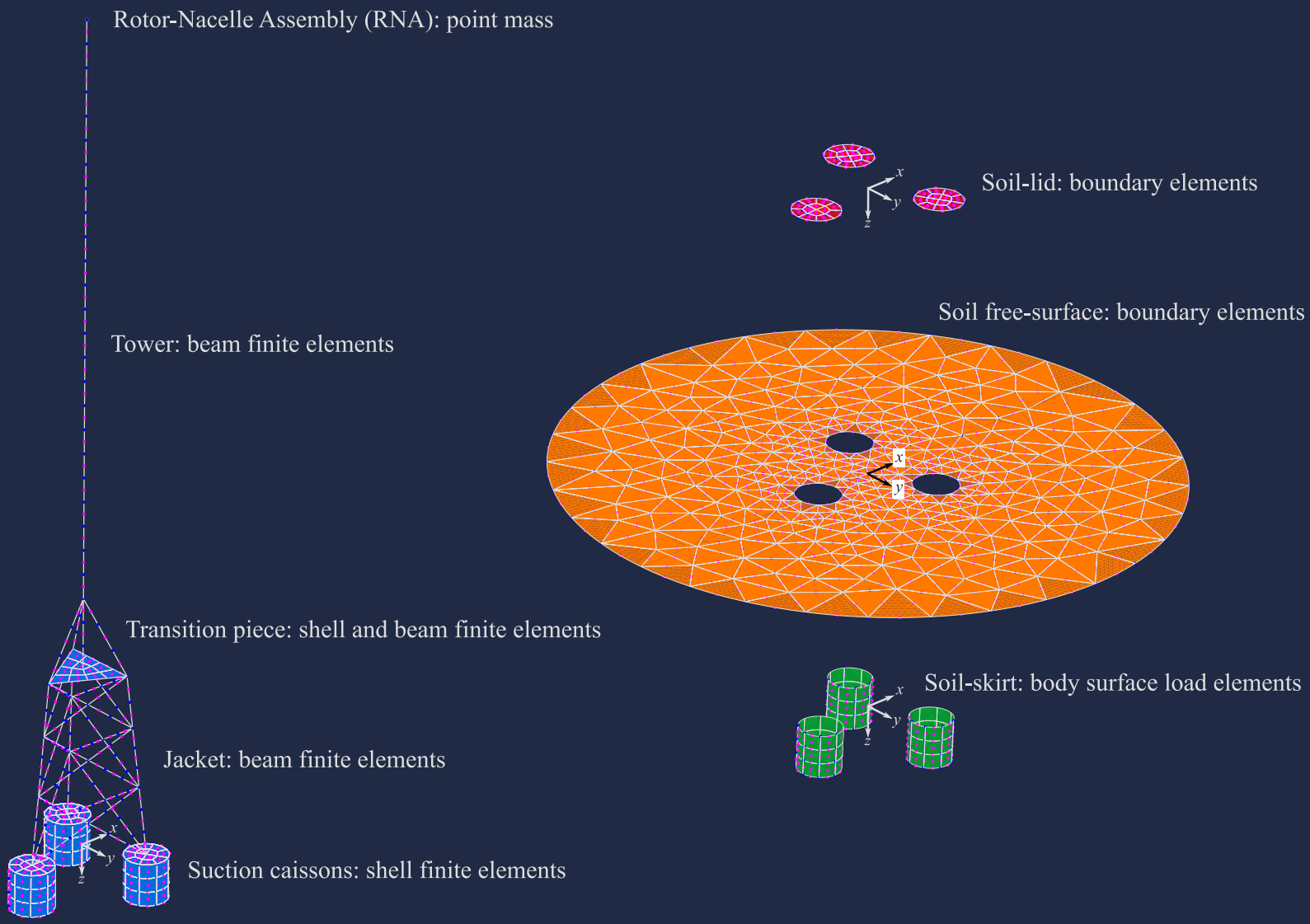
Offshore Wind Turbine



Finite Element Mesh

Boundary Element Mesh

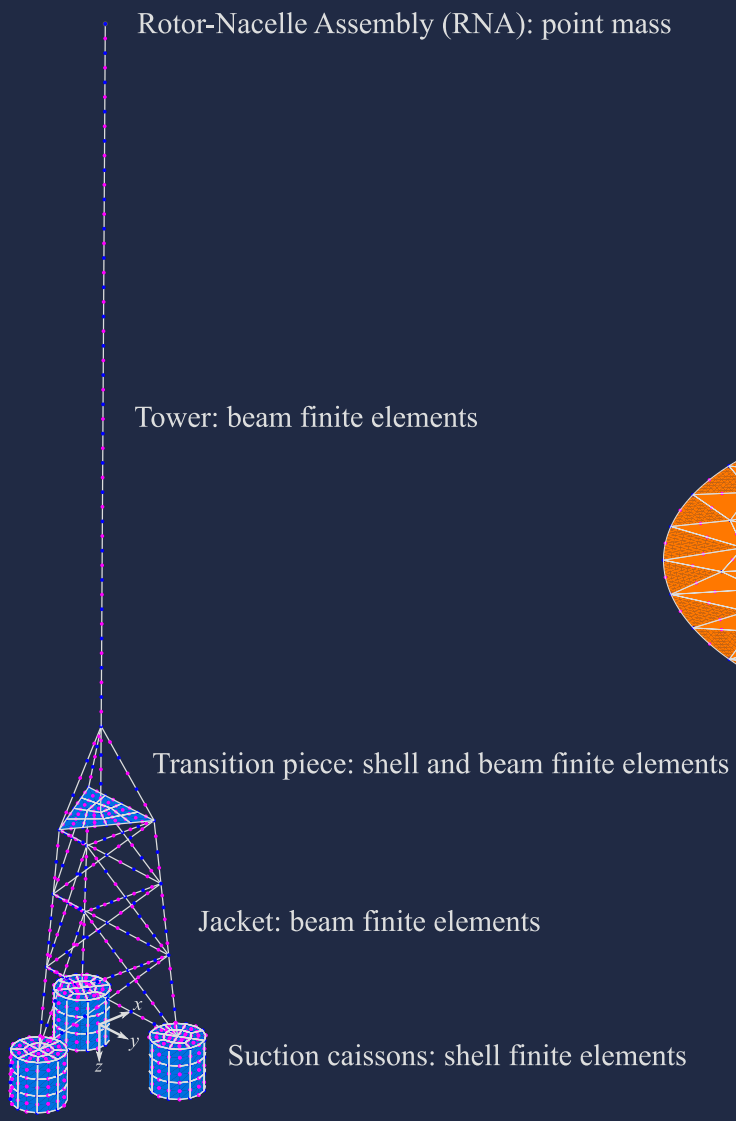
Offshore Wind Turbine



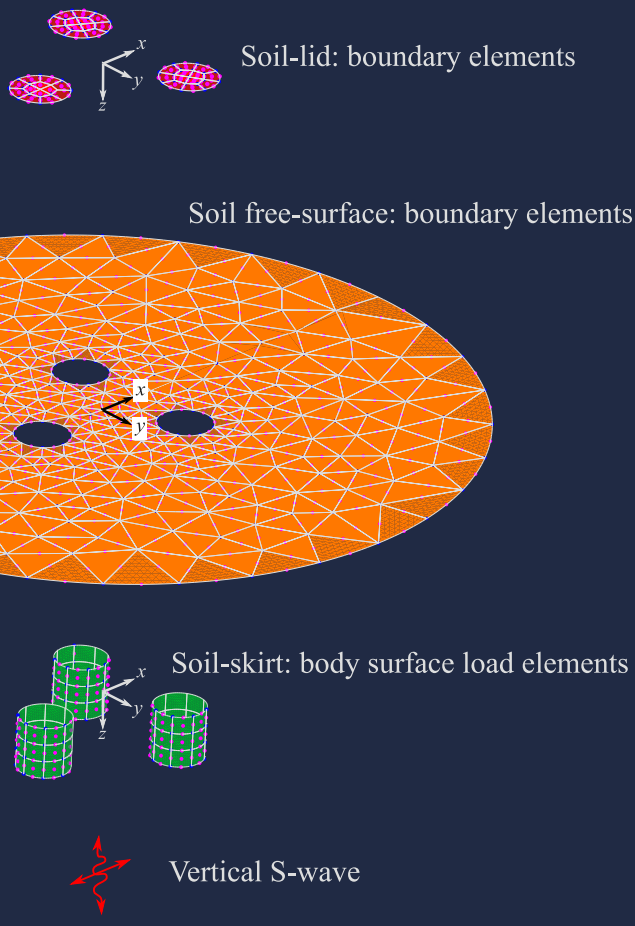
Finite Element Mesh

Boundary Element Mesh

Offshore Wind Turbine

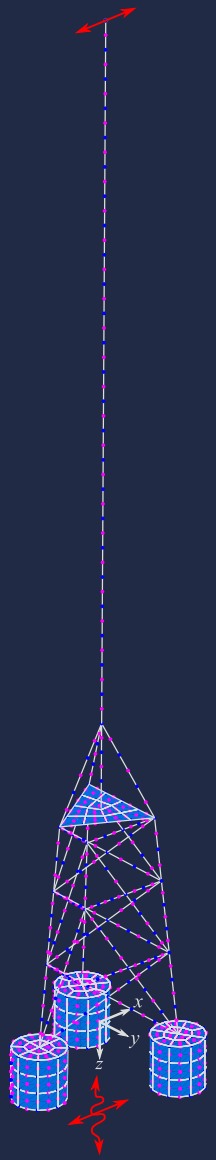


Finite Element Mesh



Boundary Element Mesh

Results: tower top displacement



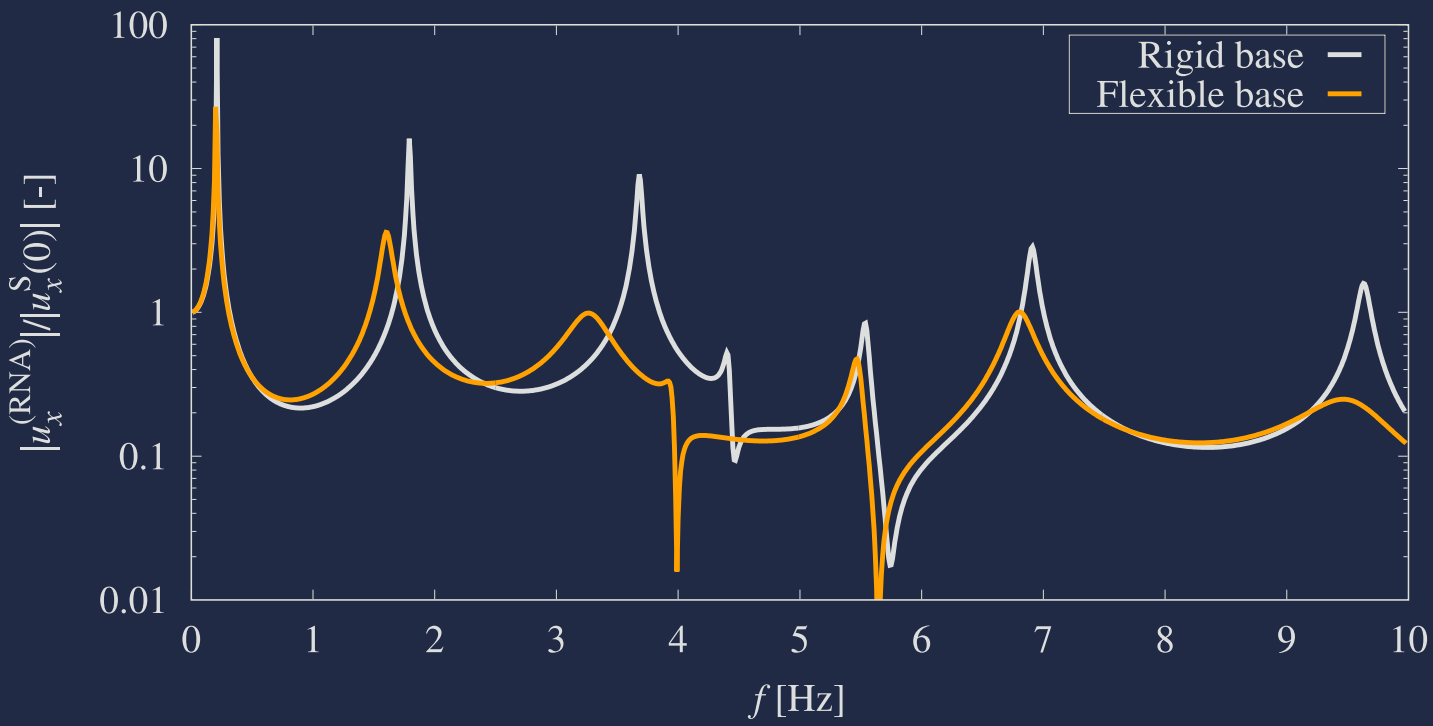
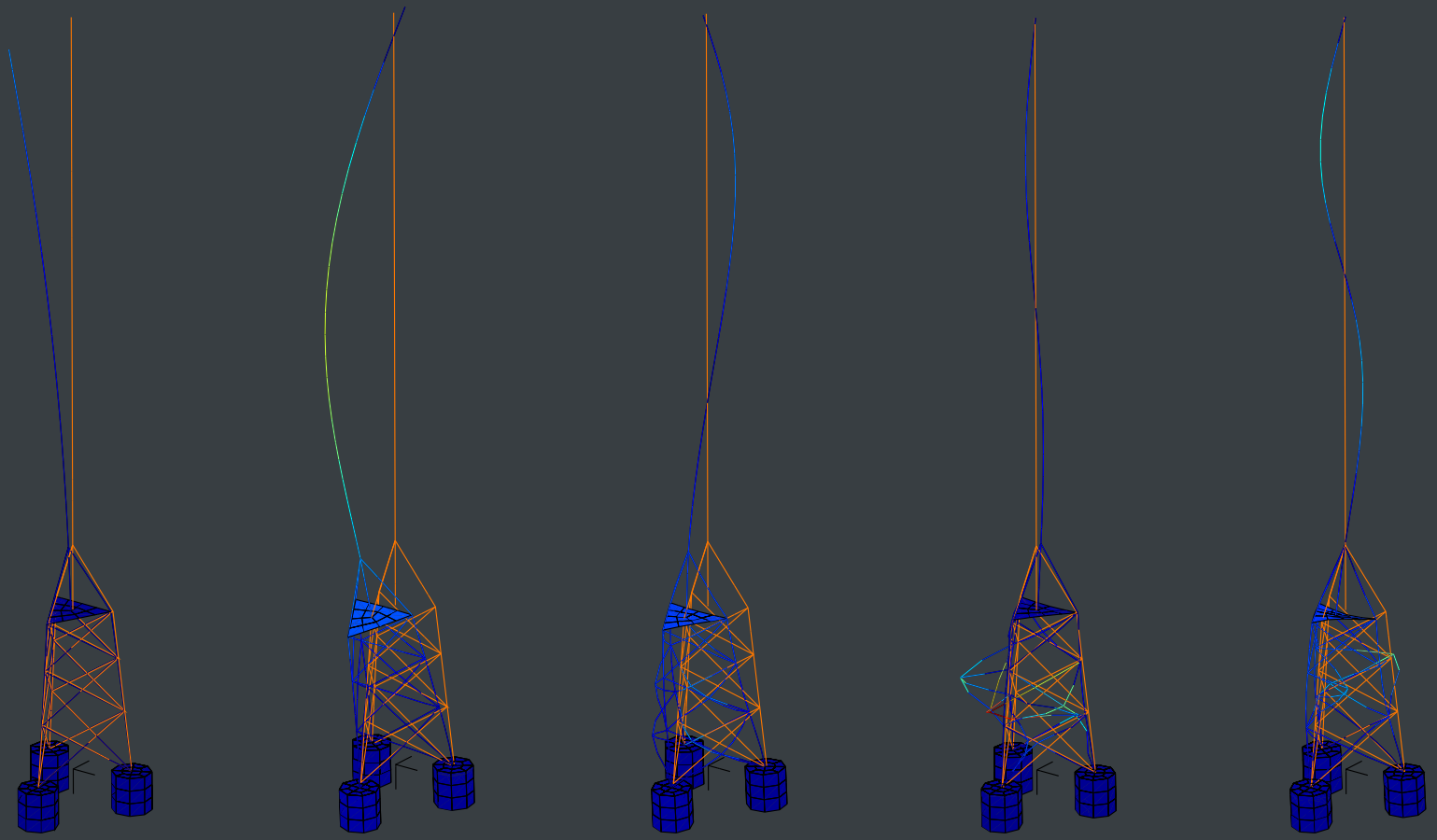


Figure: Normalized horizontal displacement response at the tower top

Results: deformed shapes



(a) $f = 0.2$ Hz (b) $f = 1.6$ Hz (c) $f = 3.26$ Hz (d) $f = 5.47$ Hz (e) $f = 6.81$ Hz

Outline

Conclusions

- Presentation of MultiFEBE project:
  - Why?
  - How?
  - What?

Conclusions

- Presentation of MultiFEBE project:
  - Why?
  - How?
  - What?
- Documentation (in progress):
  - User reference manual
  - Tutorials
  - Examples
  - Developer documentation

Conclusions

- Presentation of MultiFEBE project:
  - Why?
  - How?
  - What?
- Documentation (in progress):
  - User reference manual
  - Tutorials
  - Examples
  - Developer documentation
- Check it out!

Acknowledgments

Research Project PID2020-120102RB-I00, funded by the Agencia Estatal de Investigación of Spain, MCIN/AEI/10.13039/501100011033:

ULPGC
Universidad de
Las Palmas de
Gran Canaria

Instituto Universitario de
Sistemas Inteligentes y Aplicaciones
Numéricas en Ingenierías

SIANI

División de Mecánica de los Medios Continuos y Estructuras:
`www.mmc.siani.es`

Methodologies: Boundary Element Method

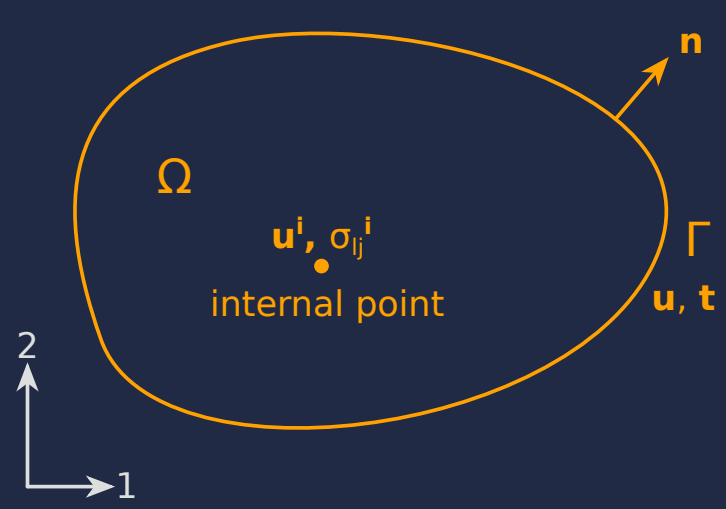
- Boundary elements. Boundary classes:
  - Ordinary:
    - Uncoupled: BC imposed
    - Coupled: BE-BE, BE-FE, BE-FE-BE
  - Crack:
    - Uncoupled: BC imposed
    - Coupled: BE-FE
- Body load elements (only coupled BE-FE):
  - Line load
  - Surface load

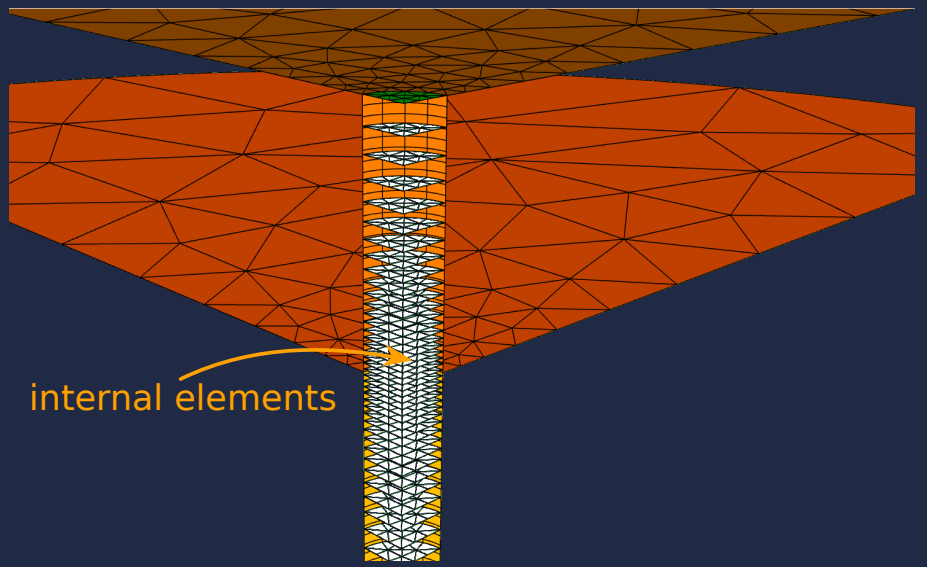
Methodologies: Boundary Element Method

- Internal points & internal elements (post-processing)

$$u_l^i = \int_\Gamma u_{lk}^* t_k \, \mathrm{d}\Gamma - \int_\Gamma t_{lk}^* u_k \, \mathrm{d}\Gamma + \int_\Omega u_{lk}^* b_k \, \mathrm{d}\Omega$$

$$\sigma_{lj}^i = \int_\Gamma d_{lkj}^* t_k \, \mathrm{d}\Gamma - \int_\Gamma s_{lkj}^* u_k \, \mathrm{d}\Gamma + \int_\Omega d_{lkj}^* b_k \, \mathrm{d}\Omega$$





Methodologies: Boundary Element Method

- Elements: linear, quadratic and cubic (only line)

[6] Domínguez, Ariza, Gallego. Int J Numer Meth Eng 48, 2000

Methodologies: Boundary Element Method

- Elements: linear, quadratic and cubic (only line)
- BEM formulations:
  - Regularization before discretization

[6] Domínguez, Ariza, Gallego. Int J Numer Meth Eng 48, 2000

Methodologies: Boundary Element Method

- Elements: linear, quadratic and cubic (only line)
- BEM formulations:
  - Regularization before discretization
  - Singular BEM
  - Hypersingular BEM
  - Dual BEM:
    - For crack elements: SBIE-HBIE
    - For avoiding spurious frequencies: SBIE$+\alpha\frac{i}{k}$HBIE

[6] Domínguez, Ariza, Gallego. Int J Numer Meth Eng 48, 2000

Methodologies: Boundary Element Method

- Elements: linear, quadratic and cubic (only line)
- BEM formulations:
  - Regularization before discretization
  - Singular BEM
  - Hypersingular BEM
  - Dual BEM:
    - For crack elements: SBIE-HBIE
    - For avoiding spurious frequencies: SBIE$+\alpha\frac{i}{k}$HBIE
  - Collocation strategies (automatic):
    - Nodal (SBIE)
    - Non-nodal multiple collocation (SBIE, HBIE) approach[6].

[6] Domínguez, Ariza, Gallego. Int J Numer Meth Eng 48, 2000

Methodologies: Boundary Element Method

Numerical algorithms:

- Closed-form 3D free-term c_{lk}^{i} calculation[7]

[7] Mantic, A new formula for the C-matrix in the Somigliana identity, J Elast 33, 1993

[8] Bordón, Aznárez, Maeso, Methodology for automatic integration (...), BETEQ, 2018

Methodologies: Boundary Element Method

Numerical algorithms:

- Closed-form 3D free-term c^i_{lk} calculation[7]
- Numerical integration:
  - Gaussian quadratures

[7] Mantic, A new formula for the C-matrix in the Somigliana identity, J Elast 33, 1993

[8] Bordón, Aznárez, Maeso, Methodology for automatic integration (...), BETEQ, 2018

Methodologies: Boundary Element Method

Numerical algorithms:

- Closed-form 3D free-term c^i_{lk} calculation[7]
- Numerical integration:
  - Gaussian quadratures
  - Adaptive[8]: $N = N(\text{error}, \text{int. pars.})$

[7] Mantic, A new formula for the C-matrix in the Somigliana identity, J Elast 33, 1993

[8] Bordón, Aznárez, Maeso, Methodology for automatic integration (...), BETEQ, 2018

Methodologies: Boundary Element Method

Numerical algorithms:

- Closed-form 3D free-term c^i_{lk} calculation[7]
- Numerical integration:
  - Gaussian quadratures
  - Adaptive[8]: $N = N(\text{error}, \text{int. pars.})$
  - Weakly singular integrals:
    - Line integrals: Telles' transformation
    - Surface integrals: polar transformation

[7] Mantic, A new formula for the C-matrix in the Somigliana identity, J Elast 33, 1993

[8] Bordón, Aznárez, Maeso, Methodology for automatic integration (...), BETEQ, 2018

Methodologies: Boundary Element Method

Numerical algorithms:

- Closed-form 3D free-term c^i_{lk} calculation[7]
- Numerical integration:
  - Gaussian quadratures
  - Adaptive[8]: $N = N(\text{error}, \text{int. pars.})$
  - Weakly singular integrals:
    - Line integrals: Telles' transformation
    - Surface integrals: polar transformation
  - Quasi-singular integrals: combination of Telles' transformation & recursive uniform subdivision.

[7] Mantic, A new formula for the C-matrix in the Somigliana identity, J Elast 33, 1993

[8] Bordón, Aznárez, Maeso, Methodology for automatic integration (...), BETEQ, 2018