Multiphysics software for MDO at Numerola

Pasi Tarvainen
Numerola Oy (Inc.)
Jyväskylä, FINLAND

CONTENTS

-Numerola

 -Examples

-Requirements for MDO software

-Realization: Numerrin

-Future needs

NUMEROLA

- **Founded in 1998**
- **Employs 20 highly-educated professionals in numerical modelling, engineering sciences and software technology**
- **Implementation of hundreds of demanding customer projects**
- **Developed the concept *Computational Technology Services* that is offered together with Kuava Oy**

COMPUTATIONAL TECHNOLOGY SERVICES

- **Service concept offering comprehensive tools for**
  - More efficient experimental design
  - Computational product planning
  - Production optimization

- **Service categories:**

  -- Modeling and optimization

  -- Engineering analysis: CFD, acoustics, data-analysis,...

  -- Software solutions

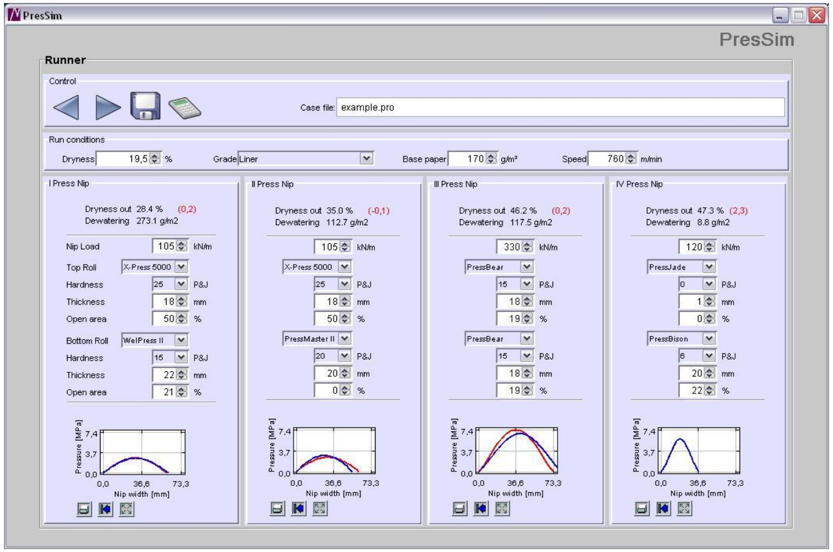
EXAMPLE: Press sections of a paper machine

-simulator to examine press constructions of the press section

-multiphysics modeling coupled with model-based optimization

-easy-to-use end-user application

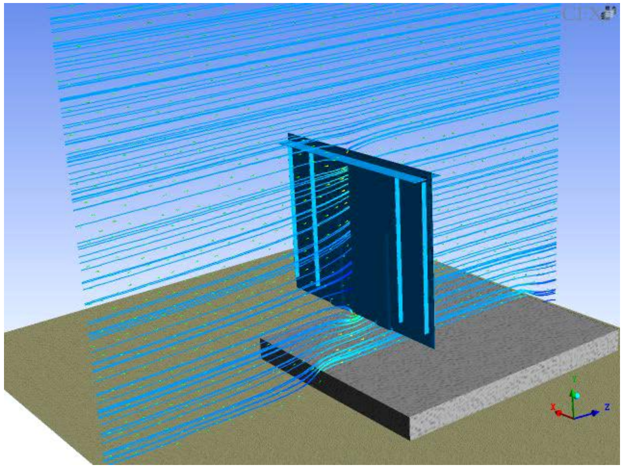
-for the business-unit of Metso Paper



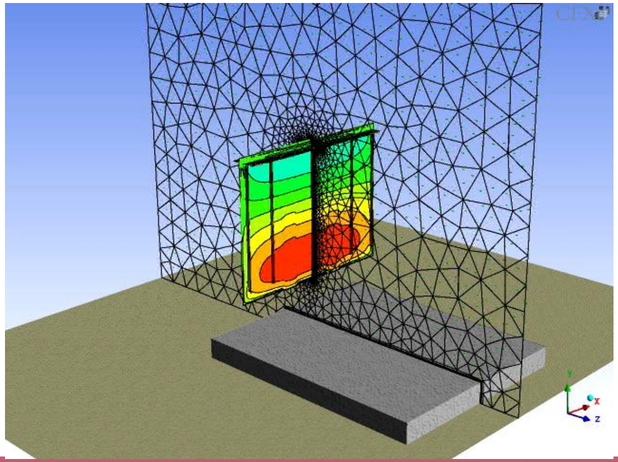
EXAMPLE: Waveroller power plant

- A power plant concept developed by a Finnish company AW-Energy, where the ocean bottom wave motion is captured to produce energy
- First prototypes were installed on the coast line of Portugal during April 2007
- The device consists of a bottom wave capturing “wing” and hydraulic generator connected via hydraulic cylinder

EXAMPLE: Coupled model for Waveroller power plant



- **CFD model**
  - Deforming mesh
  - Turbulent, time-dependent, 3D
  - Ocean velocity (wave height) is the only boundary condition
  - Water flow induces torque on the wing
- **Numerrin model**
  - Hydraulic circuit
  - Coupling with the CFD model via torque balance



REQUIREMENTS FOR MDO SOFTWARE

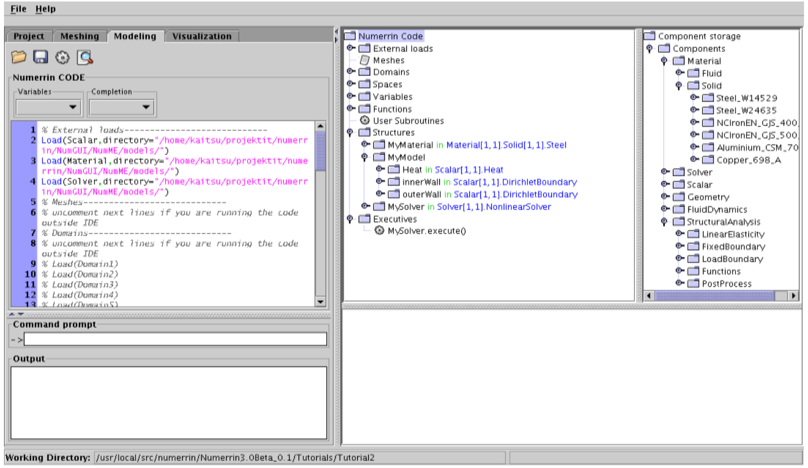
- *Fast modeling response*
- *From unit process studies to the process line studies*
- *Coupling of physical, statistical, and expert models*
- *Multiobjective optimisation*
- *Usability: flexibility to analyse different set-up's and constructions*

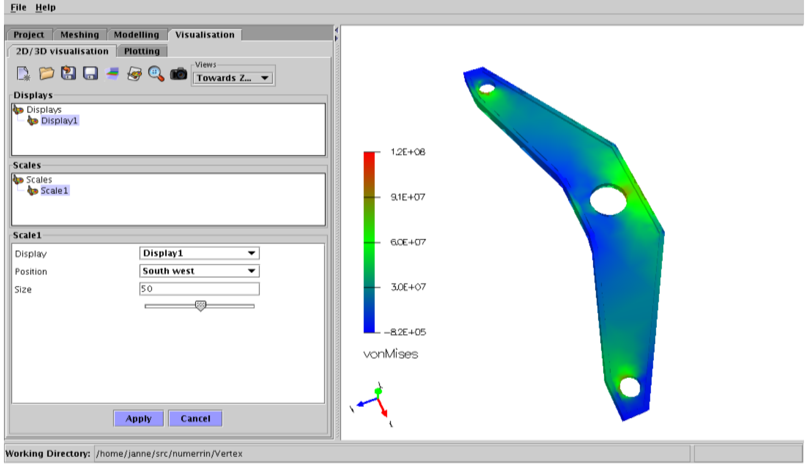
numerola

www.numerola.fi

REALIZATION: MODELLING

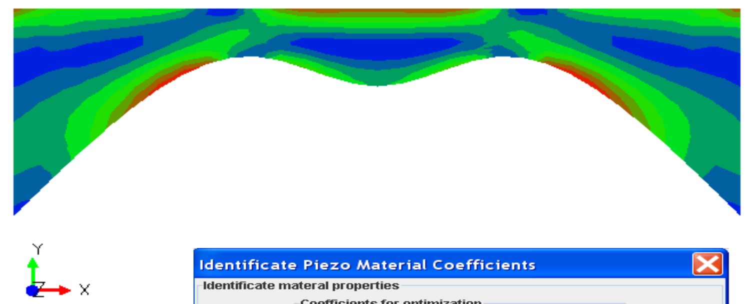
- ***The Numerrin software*** **is a platform for development and utilisation of numerical models**
- **The Numerrin models**
  - are customisable
  - can be combined
- ***Automatic differentiation*****: nonlinear problems do not require any additional steps by the user**
- **Increasing library of prepared models and components**
- **Graphical tool for pre- and post-processing as well as for using the existing components**

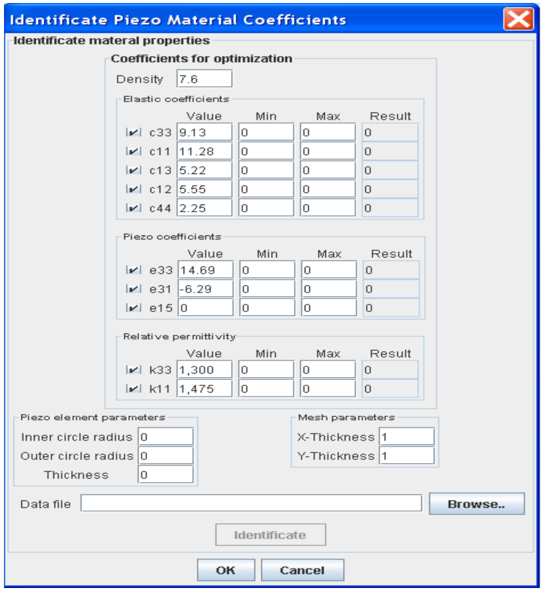




REALIZATION: OPTIMISATION

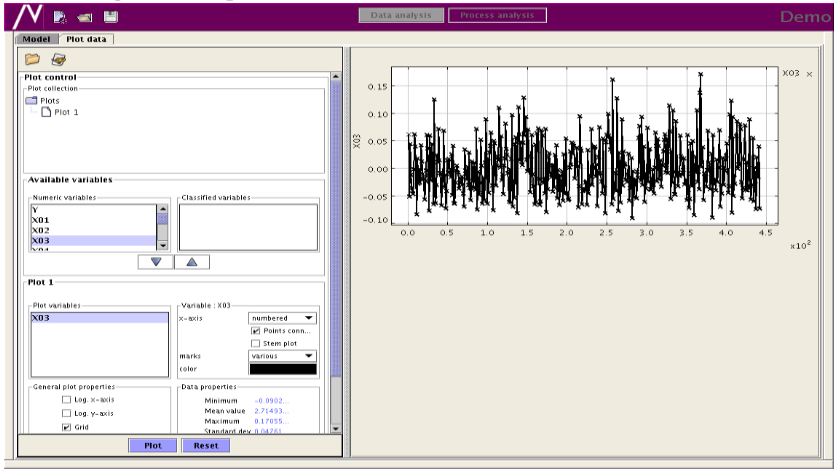
- Numerrin contains in-built model-based optimisation
- Provides e.g. dimensioning, shape and control optimisation, identification of parameters
- Efficient gradient-based optimisation methods may be utilised by means of automatic differentiation
- Shape sensitivities will be calculated for shape optimisation when needed

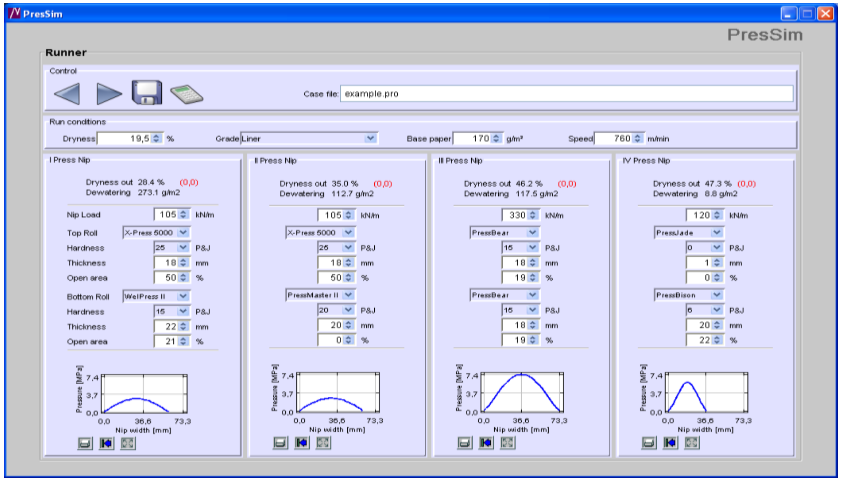




REALIZATION: SIMULATORS

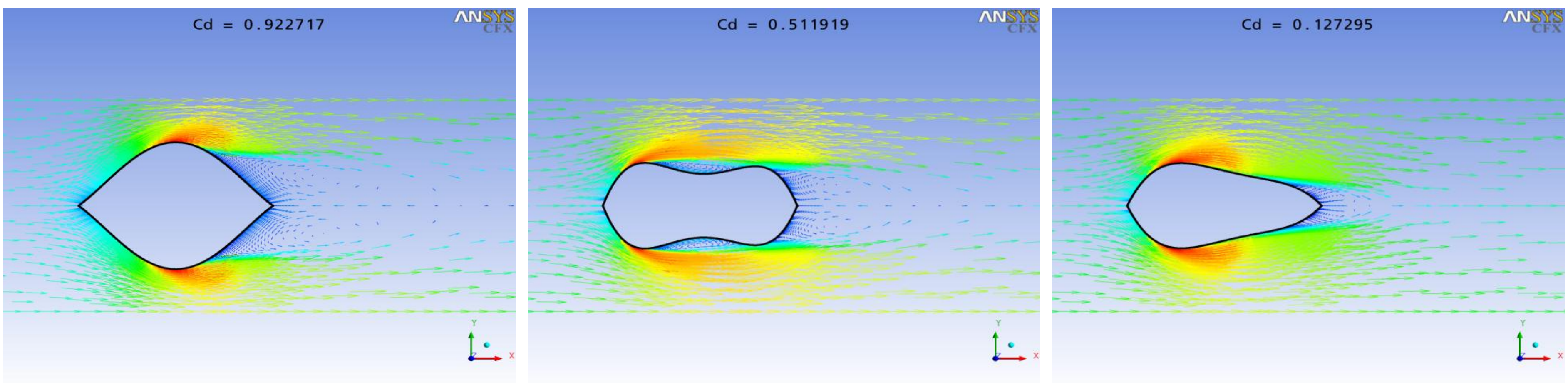
- Numerrin-based modelling systems can be equipped with a graphical user interface according to customers' needs
- The user interface may include graphical components of the Numerrin environment
- Numerrin-based simulators may contain customisable models





REALIZATION: SYSTEM AND SOFTWARE INTERFACES

- **Numerrin can communicate with other modelling and simulation software.**
- **Benefits:**
  - Expanding the modelling systems
  - Interconnecting programs
  - Adding model-based optimisation



REALIZATION: NUMERRIN MODELING LANGUAGE

***N*ovel programming language for development and simulation of numerical models:**

•Object-oriented

•Java-like implementation: precompilation and execution environment

•Includes concepts and numerical methods (e.g. PDE capabilities) needed in mathematical modelling

•Syntax resembles mathematics

•File system supports management of the modelling cases

EXAMPLE: NUMERRIN LANGUAGE

Equations

$$-\nabla\cdot(k\nabla T)=f(u),\, x\in\Omega$$
$$-\nabla\cdot(\sigma(T)\nabla u)=0,\, x\in\Omega$$
$$\sigma(T)=c/T$$
$$f(u)=\sigma\|\nabla u\|^2$$
$$T=T_0,\, u=u_0,\, x\in\Gamma_1$$
$$k\nabla T\cdot\mathbf{n}=0,\, \sigma(T)\nabla u\cdot\mathbf{n}=0,\, x\in\Gamma_2$$

Variational formulation

Find $T, u \in V = H^1(\Omega)$ such that

$$r_1(T,u,\phi_1)=0,$$
$$r_2(T,u,\phi_2)=0,$$

for all $\phi_1, \phi_2 \in \{v \in H^1(\Omega) : v_{\Gamma_1}=0\}$, where

$$r_1(T,u,\phi)=\begin{Bmatrix}\int_\Omega k\nabla T\cdot\nabla\phi - f(u)\phi\, dx\\ T_{\Gamma_1}-T_0\end{Bmatrix}$$

$$r_2(T,u,\phi)=\begin{Bmatrix}\int_\Omega \sigma(T)\nabla u\cdot\nabla\phi\, dx\\ u_{\Gamma_1}-u_0\end{Bmatrix}$$

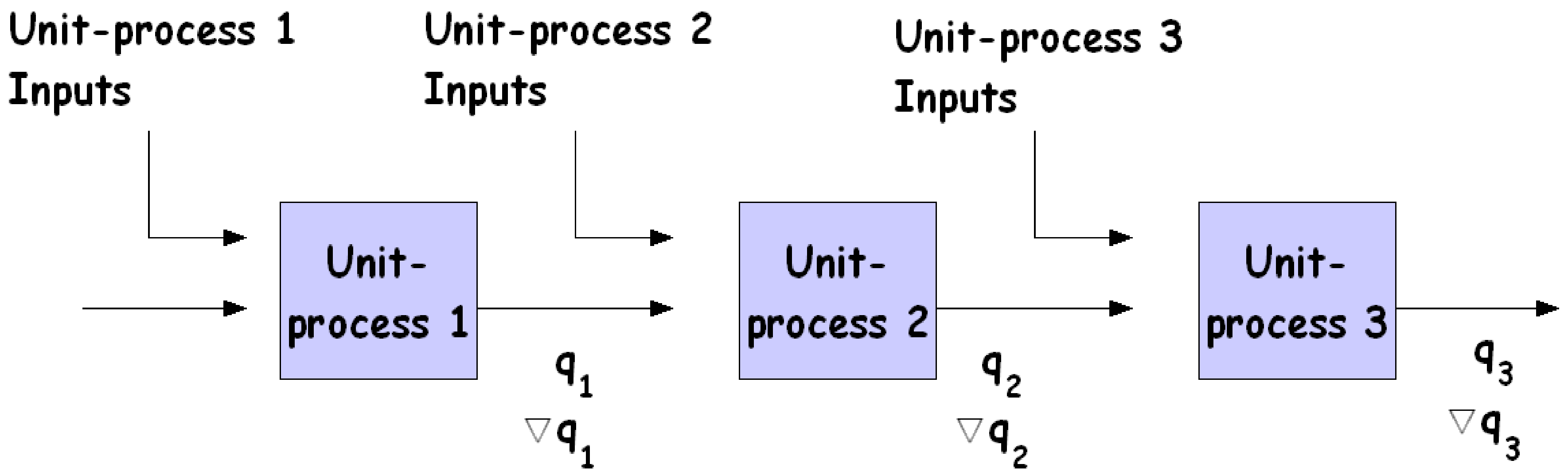
Numerrin

```
Load(Omega)
Load(Gamma1)
V = Space(Omega,"Lagrange",1)
q in V(2)
r in V(2)
drdq = Derivative(r, q)
T => q[1]
u => q[2]
c=1.5   k=0.5   T0=20   u0=100
T = T0
u = u0
for it=1:20
 Integral(Omega,"Gauss",2)
  phi => BasisFunction(V)
  sigma = c/T
  f = sigma * grad(u) dot grad(u)
  r[1] = k * grad(T) dot grad(phi) -f * phi
  r[2] = sigma * grad(u) dot grad(phi)
 EndIntegral()
 Constraint(Gamma1,V)
  r[1] = T - T0
  r[2] = u - u0
 EndConstraint()
 q = q - LU(drdq, r)
 if norm(r) < 1.e-6
   exit
 endif
endfor
```

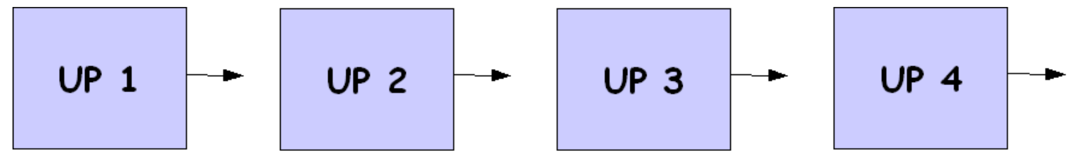
PROCESS LINE MODELLING

$$\underset{\boldsymbol{x}}{\text{Optimize}} \left(f_1(\boldsymbol{x}, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_{nm}), \ldots, f_{nf}(\boldsymbol{x}, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_{nm})\right)$$

$$\text{subject to} \begin{cases} A_1(\boldsymbol{x}, \boldsymbol{q}_1) = 0 \\ A_2(\boldsymbol{x}, \boldsymbol{q}_1, \boldsymbol{q}_2) = 0 \\ \vdots \\ A_{nm}(\boldsymbol{x}, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_{nm}) = 0 \\ \boldsymbol{x} \in S, \end{cases}$$



PROCESS LINE MODELLING



- Defining the process line:

[Process line]
title = Example_construction
sections = UP1, UP2, UP3, UP4

- Each unit process will be defined with the Numerrin language:

[UP1]
title = process_one
inputs = UPP11, UPP12
outputs = UPP13
model = UP1.num
    UPP13 = exp(UPP11) + 2.0*max(1.5, abs(2.34 - UPP12)) - 3.2

EXAMPLE: PROCESS LINE MODELLING

Model

[UP2]
title = process_two
inputs = UPP13, UPP14, UPP15, UPP16
outputs = UPP17
model = UP2.num

Equations

$$-\nabla\cdot(k\nu\nabla c)+\rho\,\mathbf{u}\cdot\nabla c=f$$

Variational formulation

$$\text{Find } c\in V=H^1(\Omega) \text{ such that}$$
$$r(c,\mathbf{u},\phi)=0,$$
$$\text{for all } \phi\in\{v\in H^1(\Omega)\},\ \text{where}$$
$$r(c,\mathbf{u},\phi)=\int_\Omega k\nu\nabla c\cdot\nabla\phi+\rho\mathbf{u}\cdot\nabla c\phi-f\phi\,dx$$

UP2.num

```
% Preprocessing
c_in = UPP13
... ...
% FEM definitions etc.
... ...
   Integral(Domain1,"Gauss",3)
    ... ...
     r = k*nu*( (grad(c,1)*grad(phi,1) +
     grad(c,2)*grad(phi,2) ) +
     rho*(u dot grad(c))*phi  -  f*phi
   EndIntegral
... ...
Constraint(Inlet boundary, V)
  r=c-c_in
Endconstraint
q = q - LU( A , r )
UPP17 = q        % Postprocessing
```

www.numerola.fi