Integrated Optimization Platform

Igor Grešovnik, Robert Vertnik & Božidar Šarler
Dec. 7, 2010

COBIK & Laboratorij za večfazne procese, Univerza v Novi Gorici
www.ung.si/si/raziskave/vecfazni-procesi/

This is presentation of plans for integrated optimization platform jointly developed in COBIK and UNG, which will also be used to support the UNG-StoreSteel project.

Why Optimization is Important?

Industrial use of simulations:

- **Improvement of Current Processes & Designs**
- **Virtual Prototyping**
  - **Optimization used in parameter identification**

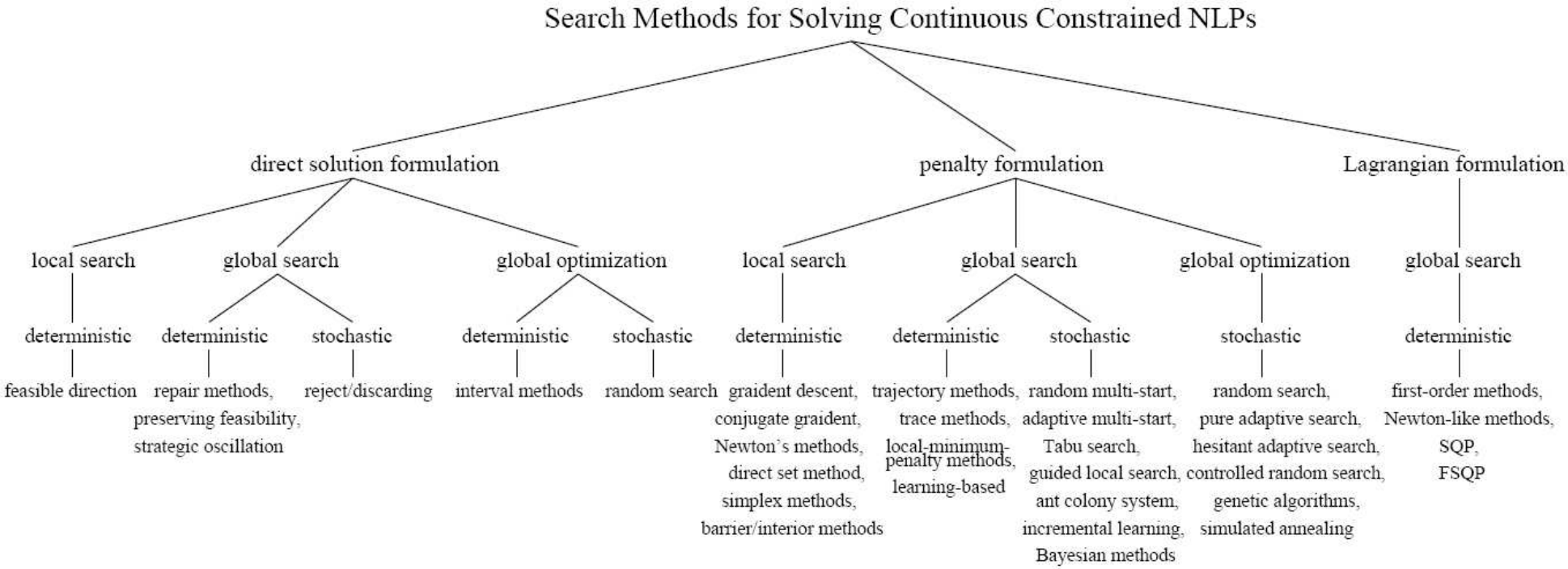
Development of numerical models:

- **Experimental Validation**
  - **Inverse identification of model parameters**

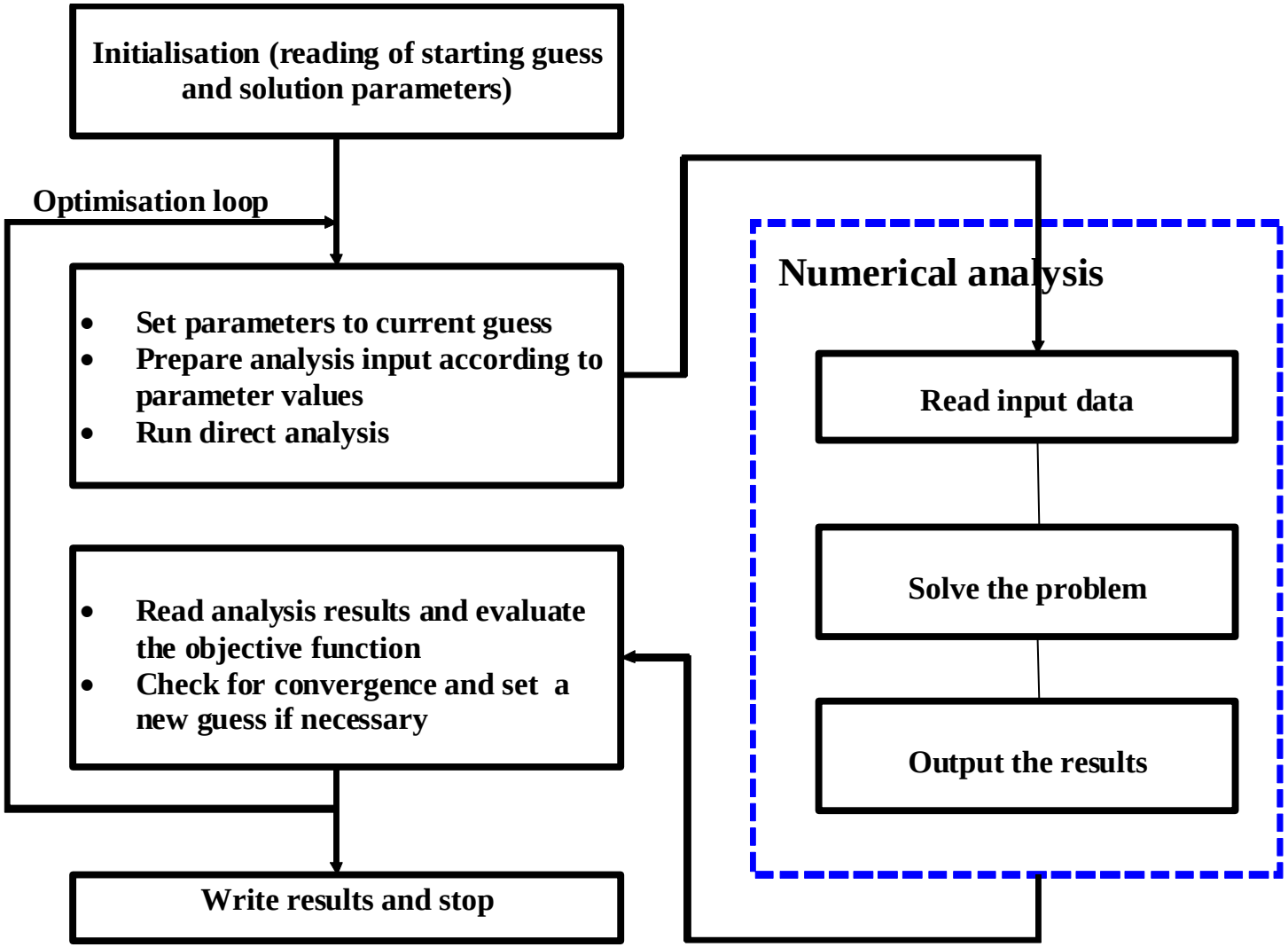
Optimization Problems – Formulation & Algorithms

minimise	$f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n$
subject to	$c_i(\mathbf{x}) \le 0, \ i \in I$
and	$c_j(\mathbf{x}) = 0, \ j \in E,$
where	$l_k \le x_k \le u_k, \ k = 1, 2, ..., n$.

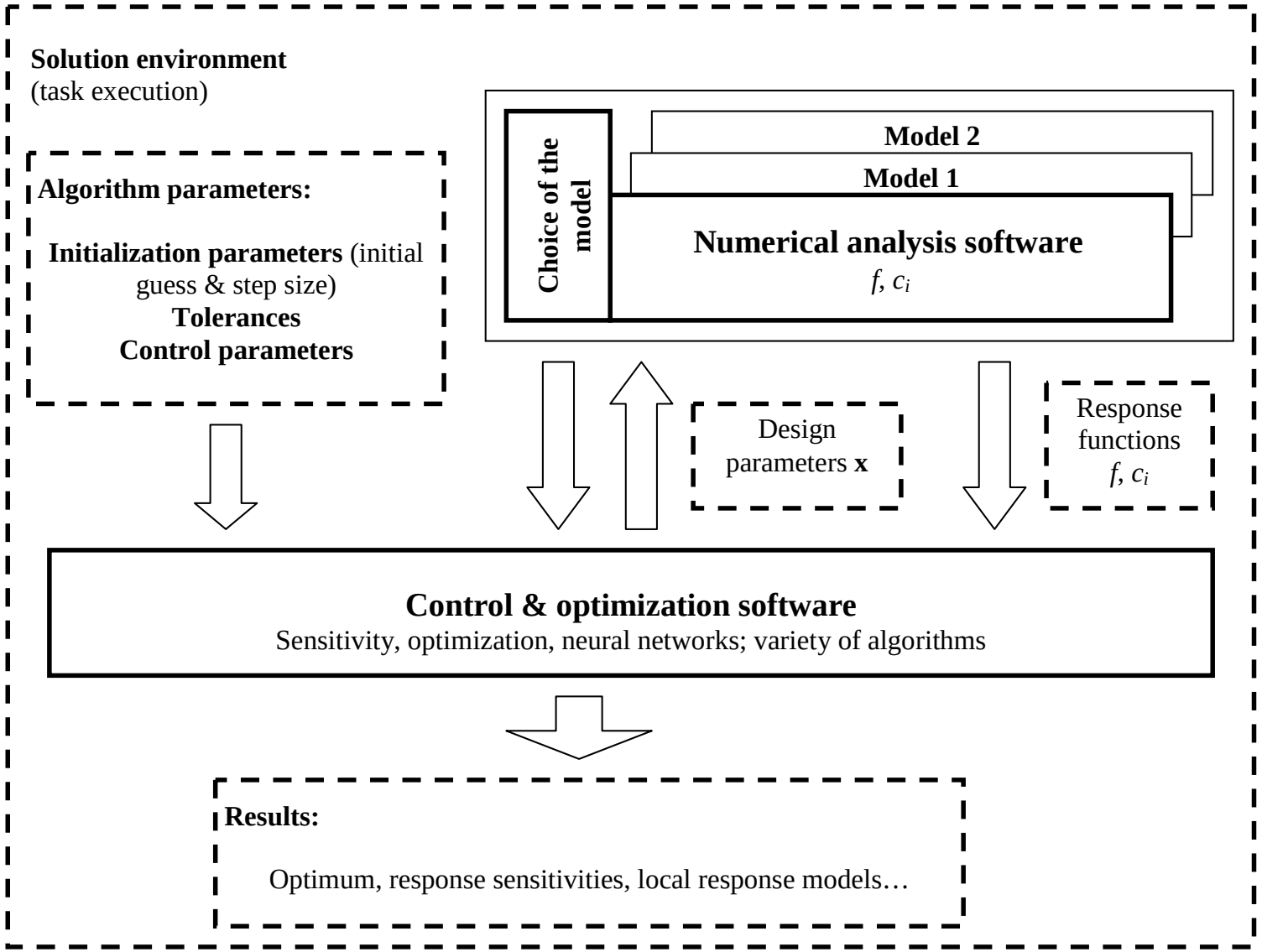
- **Classification of optimization algorithms:**



Optimization Problems – Solution Scheme



Integrated Optimization Platform



File Format for Data Exchange

Analysis input:

```
{ { p1, p2, … }, { reqcalcobj, reqcalcconstr, reqcalcgradobj,
      reqcalcgradconstr }, cd }
```

Analysis output file:

```
{
  { p1, p2 ... },
  {
    calcobj, obj,
    calcconstr, { constr1, constr2, ... },
    calcgradobj, { dobjdp1, dobjdp2, ... },
    calcgradconstr,
    {
      { dconstr1dp1, dconstr1dp2, ... },
      { dconstr2dp1, dconstr2dp2, ... },
      ...
    },
    errorcode
  },
  { reqcalcobj, reqcalcconstr, reqcalcgradobj, reqcalcgradconstr }
  < , { ind1, ind2, ... }, { coef1, coef2, ... }, defdata >
}
```

THE IDEA OF UNIFIED SIMULATION FRAMEWORK

Interactions with Simulation Development

- Definition of response functions
- Data exchange
- Model control (e.g. coarse/fine models)
- Model adjustment
  - For smoothness of response
  - Differentiation of numerical models
- Investigation of problem characteristics

Conclusion:

- Optimization depends on fitness of simulation software

Simulation Software: Current State:

- Several independent development threads
- Each simulation code developed for specific purpose
- Modularity & extensibility usually not a primary issue
- Remarkable achievements made in narrow areas

Drawbacks

- Duplication of work
- Expensive to maintain
- Difficult to extend & increase complexity
- Weak development potential per software unit
- Short lifetime (overall consequence)

Alternative Way: Unified Simulation Framework

Impulses:

- **Industrial demands for solving complex problems**
  - Multiphysics
  - Multiphase
  - Multiscale
  - Multibody, with contact interactions
  - Complex 3D geometries
- **Academic work in the field becomes increasingly multidisciplinary**
- **Increased complexity is reflected in Ph.D. subjects**

Motivation:

- **To alleviate drawbacks of the current approach**

Great Advantage: Huge Human Potential

3 Ph.D. students can be actively involved within next 6 months

- **Large amount of useful work can be done by this potential**
- **We must ensure that students will benefit from their involvement**
  - Solid framework with extensive base libraries
  - Environment for testing how new methods work in practice
    - Added value to Ph.D. thesis

Current State

- **Decision is being made with regard to development platform**
- **Establishing working conditions**
  - Equipment & infrastructure (network, etc.)
  - Workspace
- **Organization is roughly planned but must be elaborated**
  - Depends on a number of agreements

Plans

- **Relevant decisions can be made in 2-3 months**
  - We need to discuss this within project teams
- **Proof of concept possible by summer**
- **In 1 year we should catch up with current dynamics of work**
  - Work should concentrate on the new framework as much as possible

Organization of Work

- **Basic framework is set up by a small team**
  - Simple example solved (heat conduction, then coupling of fluid flow & conduction)
  - Cleaning of code
  - Elaboration of design
  - Broader discussion of concepts, code review, etc.
- **Incorporation of others:**
  - Introductory tasks (solution of isolated problems under supervision)
  - Introduction to the code
  - Work under supervision
  - Independent work (but coordinated)