Braunschweig, 24 October 2017

The DLR strategy for MDO: Computationally and collaboration intensive MDO

...or, the cybermatrix

Caslav Ilic and the VicToria team
Motivation: Realistic multi-fidelity many-discipline MDO

- Experience with multi-fidelity many-discipline MDO in the Digital-X project
- Reductions in consistency and design space to have feasible run times
- Long development and integration times to a functioning process
- Lack of performance and robustness in across-network execution
- Leaps in availability and expected use of HPC for design
- DLR Strategy 2030, guiding concept 6 “Virtual Product”
- NASA CFD Vision 2030, grand challenge 3 “MDAO of a highly-flexible advanced aircraft configuration”
- Cooperation between increasing number of diverse actors
- Technical/organizational differences between disciplinary design teams
- Disciplinary-tuned design methods, tools, and process implementations

- Another approach needed to reach the goals of the VicToria project
Representation: Universal design equation

- An **equation** that describes **any** design process?

\[
\frac{df(w,p)}{dp} - \lambda \frac{dc(w,p)}{dp} = 0, \quad c(w,p) = 0, \quad r(w,p) = 0
\]

- f — objective  
- p — design parameters  
- r — consistencies  
- c — constraints  
- \( \lambda \) — constraint scales  
- w — states

- **Implicit approximate** KKT system of complex human-machine interaction

  - “Human in the equation”, **cybernetic** Jacobians

**Cybernetics:**

a transdisciplinary approach to modeling, analysis, and control in complex systems
Expand the design equation for multiple disciplines and a global objective

\[ i = 1 \ldots n \text{ disciplines}; \quad \min F(f_1, f_2, \ldots, f_n) \quad \text{subject to } c_i, p_i, \lambda_i, r_i, w_i \]

Top-/interdependencies in classic A/C design = 0, MDO \( \neq 0 \)

Every actor maintains full control of own row
Representation: Cybermatrix reconfigurations

- Reconfiguration types: extension, decomposition, continuation
- Applying **extension** to design phases:

```
CONCEPTUAL DESIGN

PRELIMINARY DESIGN

DETAILED DESIGN
```

MDO coupling across phases
Realization: Disciplinary design processes

- **Cybermatrix row:** a disciplinary *iterative* design process

  - Disciplinary process can be implemented in any language/framework
    - Input: initial state and parameters, external data (from other disciplines)
    - Output: final state and parameters, target data (to other disciplines)
    - Execution: top steering process visible as an OS process
Realization: Interleaved design processes

- **Cybermatrix**: design processes running **interleaved**, periodic data exchange

![Diagram showing interleaved design processes]

- *estimate* missing external data using **lower-fi**
- *exchange* data state/parameter **couplings** design **interdependencies**

- **Data exchange**: exclusively file I/O, over **file system** in general sense
disk FS, mem. FS, parallel-disk FS, parallel-mem. FS, area network FS...
(workst.) (workst.) (cluster) (cluster) (multi-cluster/-workst.)

- **“Central consistent model”?** None
- **“Central exchange database”?** None
Realization: Parallel distributed assembly

- **Input collectors**: scripts for collecting top/off-diagonal data
  - Each actor provides one collector per other discipline it depends on
  - Collectors can be implemented in any language/framework
  - Actors can define data exchange directly among themselves
  - Cybermatrix representation extracted from comments in collectors

- **Automatic assembly** of a working MDO process
  - MDO process definition is a directory of input collector scripts
  - Input collectors are associated to disciplinary processes
  - Actors can test and debug own input collectors in parallel
  - Maintenance by standard software engineering practices

“Workflow integrator”? None
Realization: MDO formulations

- **Semi-automatic** using actor **hints**, or runtime **adaptive**
  - Periods of data exchange and damping per disciplinary process
  - Based on computing resources and properties of disciplinary processes

“Classic converger”
= sequential design  
= Gauss-Seidel

“Outer one-shot”
= fully-interlvd. design  
= parallel Jacobi

Real-world
= case-tailed “hybrid”
Demonstration: DLR project VicToria

Resource
- Actors: 10 departments over 8 DLR institutes (20-30 people assigned)
- Computing: 4,000-12,000 cores (DLR CASE-2 cluster + new)

Disciplinary design processes (cybermarix rows)
- Overall aircraft (label for next slide: OAD)
- Multi-fidelity aircraft synthesis (A)
- Wing/tail aerodynamics (B)
- Wing/tail structure (C)
- Fuselage structure (D)
- Loads alleviation (E)
- Flight stability (F)
- Engine* (G) ← tentative
- Flutter* (H) ← evaluation only, no design
Demonstration: VicToria MDO formulation

OAD: custom cybermatrix-tailored derivative-free optimization algorithm

- **Component Design**
  - 200 cores
  - 2 days

- **Aircraft Design**
  - 2,000 cores
  - 14 days

- **Trade Studies**
  - 10,000 cores
  - 30 days

- Full convergence of non-OAD processes
- Parallel evaluation of multiple designs in single OAD step
Cybermatrix beyond state of the art (1)

- **Practicality**
  - Synthesis of experiences and interviews through 8 DLR institutes
  - Minimally sufficient way to reach the ambitious VicToria goals
  - Highly open to involving ad-hoc human inventivity

- **Fidelity**
  - Focus on top design fidelity of involved disciplines ("hi-fi")
  - Lower fidelities supported and expected in any practical process

- **Consistency**
  - Fully converged inter-disciplinary consistency*
  - Disciplines can formulate process-specific consistency needs*
Cybermatrix beyond state of the art (2)

- **Optimality**
  - Gradient-free, gradient-based, and mixed processes supported
  - Solutions nearer to optimum than classic A/C design possible*

- **Parallellism**
  - Enormously parallel execution on HPC resources included ground-up
  - Distributed parallel process assembly and debugging among actors

- **Tolerance**
  - Graceful deterioration of optimality in face of missing capability
  - All actors continue to use their own development environments

New MDO integration framework for HPC:
**DLR Institute of Software Methods for Product Virtualization**
est. 2017 (IT/math. branch)
Thank you for your attention!

Questions?
### Some research questions

<table>
<thead>
<tr>
<th>MATH</th>
<th>INFO</th>
<th>MATH, INFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed reformulations of optimization algorithms</td>
<td>Effective execution of cyberm\textsuperscript{x} row processes on HPC resources</td>
<td>Assisted discovery and quantification of design interdepend's</td>
</tr>
<tr>
<td>ENGN, INFO</td>
<td>MATH</td>
<td>INFO</td>
</tr>
<tr>
<td>Collaborative assembly and comprehension of cybermatrix represent's</td>
<td>Parallel extensions to optimization algorithms</td>
<td>Site/hardware-specific filesystems for cybermatrix execution</td>
</tr>
<tr>
<td>MATH, INFO</td>
<td>ENGN, INFO</td>
<td>MATH</td>
</tr>
<tr>
<td>Automatic and runtime-adaptive row interleaving</td>
<td>Relaxing and blending different fidelity external data</td>
<td>Inexact evaluation reformulations of optimization algorithms</td>
</tr>
<tr>
<td>INFO</td>
<td>MATH</td>
<td></td>
</tr>
<tr>
<td>Reduction of limits to HPC use in multi-site distributed frameworks</td>
<td>HPC-capable model problem derivation from actual problem</td>
<td></td>
</tr>
</tbody>
</table>