



COST 526

**“Automatic Process Optimization in Materials Technology”  
(APOMAT)**

**Half-Yearly Report**

<b>1. Reporting Period</b>	<b>1.1.2003-30.6.2003</b>
Project title	Optimization of Fatigue Resistance of Cold Forging Tools by Considering Damage Mechanisms at Micro Scale
Project leader	<b>Dr. Igor Gresovnik</b>
Organization	C3M
Main collaborators involved	Faculty of Natural Sciences and Technology, University of Ljubljana, Slovenia.

**2. Funding Situation**

Amount of money received specifically for COST	2.9 kEuros
Other resources partially used for the project	kEuros

**3. International Collaboration**

(mention group and type of work done in collaboration during the reporting period)

Participation in the Working Group Meeting + project progress report

YES

NO

UTC, France: discussio of direct shape transform methodolodies.  
Rockfield Software, Swansea; implementation of basic functions for program flow control and data exchange between the optimisation shell and the simulation programme “Elfen”.

**4. Industry participation**

(mention name of companies and work done in collaboration during the whole project)

Iskra-Avtoelektrika. Analysis of service life of tooling systems

<b>5. Meetings, visits, exchange of scientists, short-term scientific missions</b>	<b>Location, date</b>
Working Group Meeting Meeting with UTC researchers to exchange ideas on shape parameterisation	Brussels Universite de Technologie de Compiegne, France, July 8 2003

## **6. Progress within the reporting period**

(Not exceeding 3 pages, including tables and figures)

Within the reported period, work on the interface between the optimization shell Inverse and the simulation program Elfen has continued. Development of shape parameterization techniques has also started.

The direct interface with Elfen was defined and its beta version was implemented. As opposed to the file interface, this will enable the optimization shell to access simulation data directly through the database of the simulation program. An obvious advantage of this will be speed-up of those parts of the solution procedure where optimization algorithms must interact with the finite element analysis. Beside that, the control over the simulation flow and access to its data will be improved. The data that is not output to the result files can now be accessed and the data that affects the course of the simulation can be controlled at any stage of the incremental solution procedure, not only at the beginning and end of the analysis.

The direct interface requires optimization and simulation routines to be linked in the same executable, which enables direct data access through global variables and crosswise function calls. Since the optimization shell and the simulation program have been historically developed independently, some compatibility problems were solved first in order to successfully implement the interface. The structure of the simulation program was adjusted in order to allow successive running of several analyses within a single process. This involved resetting of the initial state after completion of the finite element analysis.

In order to minimize the development and maintenance effort, the interface was based on a basic layer containing a small number of low level functions, on top of which a more extensive functionality of the interface was built in the appropriate module of the optimization shell. The basic layer contains a single data access function, which takes as its input a specification (naming and indices) of the data within the analysis database, and returns the address of the data in memory. Due to the organization of Elfen database, this enables reading and changing any piece of the analysis data during its course.

The flow of control was organized in such a way that the structure of the finite element analysis program remained almost unchanged. A single analysis function was constructed through which the shell can invoke a finite element analysis, being more or less literal transcription of the existent code. Elfen holds the main function and invokes the shell after its initialization. After taking control, the shell can run the finite element simulation by calling the analysis function whenever this is required. The simulation is run as a whole. Intervention of the shell at any intermediate stage is enabled through an event protocol. The analysis triggers events at certain stages (e.g. after reading input data, after every time increment, etc.). Events are equipped by tags, which identify the places where they occurred. All events are intercepted by the shell code, which inspects their tags and undertakes given actions (e.g. reading or changing analysis data) when these are prescribed for given kinds of events. Development of direct shape parameterization techniques has been considered. The goal is to be able to explicitly relate the finite element mesh to the design parameters. In this way the derivative information necessary for sensitivity analysis would be easily obtained. The approach is required to work in conjunction with automatic mesh generation, in which case only object boundaries are related to design parameters and the finite element mesh is automatically generated. The approach must provide consistent derivatives of material point positions, which can be assigned to mesh nodes upon mesh generation. Until now an approach utilizing parameter dependent maps in 2D or 3D has been considered in detail. We tried to analyze possible drawbacks and identified some possible alternative approaches by the literature review. A prototype implementation of algorithms will be performed in the next reported period.

## **7. List of publications**

a) Published

I. Gresovnik and T. Rodic, "*An Integral Approach to Optimisation in Forming Technology*", 5<sup>th</sup> World Congress of Structural and Multidisciplinary Optimisation, May 19-23, 2003 Lido di Jesolo, Italy

b) Submitted for publications

c) In preparation