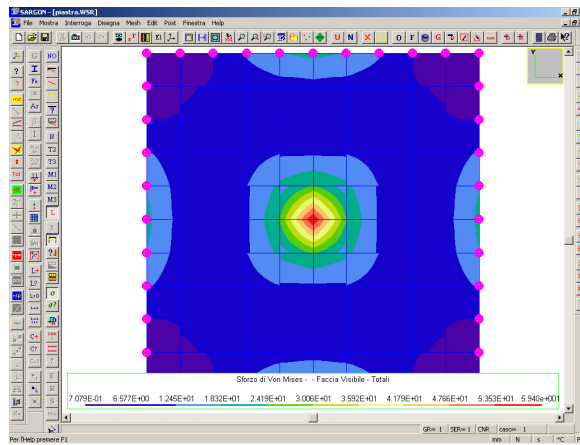




# WHO DOES WHAT



<http://www.castaliaweb.com>

Via Pinturicchio, 24

20133 Milano

[staff@castaliaweb.com](mailto:staff@castaliaweb.com)

Copyright © 1991-2008 – Castalia srl

Rev. 2.0 September 2008

*Added to documentation in January 2008*

## **1. Foreword**

This document is intended as a guide for third parties analyzing Sargon models and Sargon results, so as to check the work done by Sargon users. Sargon users should be completely aware of all the important concepts mentioned in this document, because they are fully explained in the Sargon documentation.

This document is deliberately kept short: much more information can be found in the Sargon documentation, which can be obtained along with the “Sargon Reader” from the user himself and can be downloaded from the company website [www.castaliaweb.com](http://www.castaliaweb.com).

## **2. What the program does**

### **2.1. Design lines**

Sargon is a finite element program using the displacement method, and it is purposely conceived to explicate finite element concepts, not conceal them. This means that there is no automatic algorithm that generates a model suitable for any design purpose. It is up to the user to build the finite element model in such a way that the mesh, loads, constraints and more are appropriate for the design purpose at hand.

The Sargon producer does not believe that automatic generation of models is an appropriate way of working when dealing with general purpose software. On the other hand, Sargon can be used, if properly managed, to study a very broad class of structures, ranging from industrial to civil, naval, mechanical and more.

Actually, since its conception in 1990, it has been used in each of these fields.

Given a FEM model, prepared in Sargon, it mainly computes stresses, displacements and exploitation indices according to different standards. Sargon is able to present results in a wide variety of ways and is a powerful tool for analysis which can greatly speed up computations and checks, performing very complex tasks in a reasonable amount of computational time.



Nothing in the program is intended to imply that the software could in any case substitute the skill and experience of a designer, even though the program is the result of years of careful, highly specialized programming by experts in the field of structural analysis.

## 2.2. General info

Sargon is a software package made up of four main groups of programs: graphic user interface (GUI), solvers, checkers, and helpers.

The model is necessarily built up using the GUI. Some commands make it possible to import meshes prepared elsewhere (not drawings: using drawings that automatically lead to FEM models is considered an unsafe practice by the producer), or to export finite element models already done.

The solvers are: CLEVER (linear static analysis); LEDA (modal analysis); SPECTRUM (response spectrum analysis); FREQUENCY (frequency response analysis via modal analysis); SOCLEVER (nonlinear static analysis in small displacements, geometric effects, not material nonlinearities).

The checkers are: CNR (allowable stress and limit states); AISC-ASD; BS 5950 (1985); EN 1993-1-1 (2005).

Helpers external modules are: Checksolvers (validation purposes); Mergedbase (load cases overwriting); Samba (cross sections management); and several programs to export and import the model into various formats.

## 2.3. Short reference main program (GUI)

The program manages the following finite elements:

1. truss (2-noded);
2. beam (hermitian two nodes, *id est* cubic shape functions);
3. membrane (in plane stiffness only, 3- and 4-noded elements, several formulations: CST, bilinear quad, SRI, Wilson Ibrahimbegovic);

4. thin plate (Kirchhoff plate theory, 3- and 4-noded elements, DKT, 4DKT);
5. thick plate (Mindlin-Reissner plate theory, 3- and 4-four noded elements, Hughes);
6. solid (4-, 6- and 8-noded elements, several formulations);
7. spring.

The program manages the following loads:

1. Nodal loads (forces and moments);
2. Temperature loads (applied to elements);
3. Distributed loads applied to beam elements (constant, concentrated, linear to partial length, forces and moments);
4. Settling of constraints.

Distributed loads over surface and solid elements are added in GUI and are always immediately converted into nodal loads (by lumping). This means that the mesh should be sufficiently fine to permit such modelling. No graphic deception regarding the real nature of load modelling is thus possible (coarse meshes with finely positioned internal loads are meaningless, no matter how they may appear).

Constraints are applied to nodes.

## **2.4. Short reference: solvers**

**CLEVER:**

Linear static solver using two possible solution algorithms: skyline solver and sparse matrix solver. Both use the Choleskij linear system solution method. The program is very strict about not allowing non-positive definite stiffness matrices.

**LEDA:**

Modal analysis solver using subspace iteration method. Matrix triangulation is done using skyline techniques. An check for orthogonality is done at each run.

**SPECTRUM:**

Response spectrum combination that also includes an “error factor” feature. Key points



regard mode combination rules (SRSS, CQC, correction methods, modal amplification method).

#### FREQUENCY:

Linear frequency response. The user inputs forces, frequencies and damping, and chooses the modes to include in computation.

#### SOCLEVER

Nonlinear static solver (Newton Raphson) capable of performing second-order analysis of models made up of beams and trusses. If the loads applied are higher than buckling loads, the program stops and predicts the load level at which the matrix is no longer positive definite.

### **2.5. Short reference: checkers**

All the checkers check resistance and stability of members. Global stability checks should be done using SOCLEVER or by displacement checks. Connection checking is not within the program application range (use CSE instead). Also, local checks, such as web buckling under concentrated loads, are beyond the range of the checkers.

Checks are performed for every load combination. No envelope concept is used in checking because it is neither possible nor appropriate in steel structures.

When a member is made up of several finite elements, stability checks are performed using “superelements”, which have to be added to the model by the user.

The internal forces used by the checkers are those stored into SDB file. This file can be managed by the solvers directly, or by Mergedbase, if the user has chosen to use it.

### **3. What the user should do**

#### GUI

The user must conceive the finite element model in such a way that the model

properly fits the design purpose at hand. Specifying mesh size, finite element kind and formulation, setting up several possible models, and studying the sensitivity of the solution are all up to the user and are strongly recommended.

The user must apply loads properly and, by means of internal commands, must check global and partial resultants against expected values. Spatial distribution of loads must be carefully checked. Load combinations are absolutely of the greatest importance with regard to the checks. The user can use automatic rules and/or add combinations manually, and must be sure of the proper definition of the combinations. When there are hundreds or thousands of combinations, it is suggested as a normal safety procedure that sample combinations be checked independently in a separate model.

When using surface and solid elements the user must specify an appropriate mesh size.

Mesh should be checked for double nodes, unreferenced nodes and unconnected elements by carefully using internal commands and by carefully checking displacement field and stress fields.

Constraints must reflect the real design situation.

## **Solvers**

### **CLEVER**

The user must be aware that singularities in the stiffness matrix could be due to inappropriate design and must fully understand the reasons why a model does not run, avoiding the ad hoc addition of constraints to bypass the block. LEDA with shift can be used to outline rigid or almost rigid body motions, if any.

### **LEDA**

The user must properly specify the number of modes, the tolerance and the target total mass percentage.

### **SPECTRUM**

The user must fully understand all the rules governing mode combinations and understand that some of them lose signs while others use “numerical” sign permutations to get results. The mode amplification method, when possible, should be used together with SRSS and

CQC.

## FREQUENCY

Mode choice and damping tuning are key points in using this solver. They must be fully explained and documented.

## SOCLEVER

Here the number of steps and the tolerance play important roles. If final displacements are not within the range of small displacement theory the analysis becomes meaningless.

## Checkers

To properly fine-tune the stability checks, the user must carefully set the three slenderness coefficients of all elements and superelements. The program cannot set these coefficients because there are no reliable ways to do that automatically.

The user must have completely understood the way superelements work in Sargon and should have added as many superelements as needed in the model to assure that the checks are properly done. Changing superelements can dramatically change model results.

The user must have added the correct number of combinations.

## 4. Validation

Here is a summary of what the user should do to better assess model reliability. The Sargon producer's validation policy is explained in another document (Sargon: validation, also downloadable from the web site).

### 4.1. Solvers

The user should use the module "checksolvers.exe" and carefully examine the output file of this program, which is (optionally) run at each solving. This program controls the output independently of the solver, and so is able to point out possible problems in the

solving phase. The “Checksolvers.exe” output file should be an integral part of the output documentation available to third parties.

Additionally, the user should manually check some key features of the stress and displacement fields:

1. global simplified equilibrium checks to once more verify load resultants and constraint reactions;
2. order of magnitude of stresses and displacements;
3. displacement fields, deformed views must be compliant with the expected behaviour;
4. internal action diagrams should be sampled in an appropriate number of points and diagrams should match those expected in theory;
5. constraints and end releases should be compliant with the initial settings (e.g., free displacement vs possible obstacles);
6. displacement amplitude must be compliant with small displacement theory.

#### **4.2. Checkers**

The user must carefully review resistance and stability checks and verify them independently in a sample made up of a number of checks which necessarily increases with the number of load combinations and the size of the finite element model.

These independent checks must be done in a simplified way, if possible, completely independently of the model results. Possible tests regard axial force estimate, simplified buckling checks, maximum bending moment evaluation and simplified checks against stress limits. Key members should be checked independently and results must be discussed and explained in view of the program results.

If available, checking the model with more than one Sargon checker (using allowable stress and/or partial factors method) is highly recommended.

Automatic checks do not completely free designers from the burden of responsibility for this task, which is of the utmost importance. Automatic checks are a design instrument, *not the design itself*.



### 4.3. Third party management

In order to simplify the checks performed by third parties, the user should follow these possible steps:

1. Prepare an input file for other solvers by using Sargon internal commands, and give it to the third party in charge of independent checks;
2. Inform the third party of the existence of the free Sargon Reader, which can be downloaded from Castalia web site <http://www.castaliaweb.com>. The third party can watch the model and check it by using the same program. The Program version should coincide with model version, for obvious reasons.
3. A higher level of safety is obtained if the model is run in a different solver or, perhaps, recreated independently using a different program (see Italian norms, NTS, chapt. 10).

Milan, 16 January 2008