

150 00 Prague Czech Republic Phone: +420 220 610 018 E-mail: <u>cervenka@cervenka.cz</u> Web: <u>http://www.cervenka.cz</u>

ATENA 2023 Program Documentation Part 2-4

User's Manual ATENA with CeSTaR 2 Module

Project result TM0100059-V4



Written by

Jan Červenka, Tomáš Altman, Zdeněk Janda, Pavel Pálek and Radomír Pukl

Prague, December 2022

Acknowledgements:

The software was developed with partial support of **TAČR DELTA 2 Programme**

T A Č R

("DELTA 2 Funding programme for applied research, experimental development and innovation", project No. TM01000059 **CeSTaR 2** – "Reducing material demands and enhancing structural capacity of multi-spiral reinforced concrete columns - advanced simulation and experimental validation")

Trademarks:

ATENA is registered trademark of Vladimir Cervenka.

GiD is registered trademark of CIMNE of Barcelona, Spain.

Microsoft and Microsoft Windows are registered trademarks of Microsoft Corporation.

Other names may be trademarks of their respective owners.

Copyright © 2022 Červenka Consulting, s.r.o.

CONTENTS

Con	ITENTS	. I
1		1
2	WORKFLOW IN ATENA-PRE	2
3	Installation and Registration	4
4	ANALYSIS TYPE	5
5	DEFINITION OF GEOMETRY	6
6	Materials	9
7	BUILDING ELEMENT TYPES	.21
8	REINFORCING TYPES	.25
9	SPRING AND INTERFACE TYPES	.26
10	FE MESH PARAMETERS AND MESH GENERATOR	.27
11	ANALYSIS	. 30
12	SCRIPTING IN ATENA-PRE	. 34
Ref	ERENCES	. 39

1 INTRODUCTION

Program **ATENA-PRE** is developed by Červenka Consulting s.r.o. as a direct preprocessor for **ATENA** that is used for nonlinear simulations of structures. In the preprocessor **ATENA-PRE** user can define finite element model with its geometry, boundary conditions and specific data for **ATENA** analysis. When the model definition is completed in **ATENA-PRE**, the input file for **ATENA** (file name extension: *.inp*) is generated and the simulation can be initiated. The following modules for different types of analyses are available:

• A	TENA/Static,	- static 2D and 3D analysis
-----	--------------	-----------------------------

- ATENA/Creep, creep 2D and 3D analysis
- ATENA/Transport, transport 2D and 3D analysis
- ATENA/Dynamic dynamic 2D and 3D analysis.

The postprocessing of the simulation is possible in ATENA Studio or ATENA Engineering.

2 WORKFLOW IN ATENA-PRE

This document represents the User's manual for the new module ATENA CeSTaR-2 software, which was developed during an international collaborative research project CeSTaR-2 "Reducing material demands and enhancing structural capacity of multi-spiral reinforced concrete columns - advanced simulation and experimental validation". The project was supported by the DELTA-2 research program of the Czech Technological Agency. The document focus is on the preparation of model data using the newly developed pre-processor, which is referred to as ATENA-PRE. For more detailed description of material input parameters, analysis execution, postprocessing the reader should consult the Example and Validation manual [2].

When working with ATENA-PRE, the workflow of model preparation is as follows:

- Select one of the problem types for **ATENA**.
- Create a geometrical model.
- Select material models, define parameters.
- Assign materials and element type to the geometry.
- Generate finite element mesh.
- Create Load cases and assign respective boundary conditions such as supports and load.
- Create loading history by defining in Tasks.
- Execute finite element analysis with ATENA Studio or AtenaConsole.

The following text has aim to explain the procedure above in more detail to help the user to define and execute the **ATENA** simulation.

Initial part of manual is dedicated to geometry definition while the geometry can be created directly in **ATENA-PRE**, standard export file types from other programs are supported for import. Also, **ATENA-PRE** supports the definition by scripting which is very helpful for various geometry shape that can be easily defined by functions along with direct definition of points etc.

The following part of the manual describes definition of ATENA model in ATENA-PRE.

It is also recommended to go through the **ATENA 2023 CeSTaR-2** Example and validation manual [2] before starting with one's own modelling.



Fig. 2-1: ATENA-PRE interface

3 INSTALLATION AND REGISTRATION

For installation download the installer package available at company website <u>www.cervenka.cz</u>. In section Download choose the ATENA 6.x.x package and when the downloading is finished run the installer. ATENA-PRE is inherent part of the installer therefore no special selection in needed.

TENA V6.0.0 pre-release	Setup – 🗆 🗙
	Choose Components Choose which features of ATENA V6.0.0 pre-release you want to install.
Check the components you install. Click Next to continu	want to install and uncheck the components you don't want to e.
Select components to instal	HASP Driver
	Description
Space required: 1.9 GB	Position your mouse over a component to see its description.
Nullsoft Install System v3.03 –	Next > Cancel

Fig. 3-2: ATENA installer

After installation open ATENA Center to view your curent licence (use the HASP HW Key or join network with NetHASP). Otherwise there is a link to trial licence information on company website. Some licence related questions are answered in [3].



Fig. 3-3: ATENA center – Your licence.

4 ANALYSIS TYPE

When user creates the new model in **ATENA-PRE**, the Analysis type of the simulation in **ATENA** has to be selected. The configuration of the Analysis type creates the compatible data such as units, materials, conditions etc. which are written in the input file for **ATENA**. To set the problem type, in main menu select **Settings** | **Project settings**. In the pop-up window user selects one of the **ATENA** Analysis type (Fig. 4-4):

- Static static analyses
- Dynamic analyses including dynamic loading
- Transport heat and humidity transfer
- Creep creep of concrete.

, toject	Project - General				
General AutoSave Modelling General Tolerances Snaps Viewports General Grid Undo/Redo Boundary conditions	Analysis type: Space type: Units base system: Project name: Project location: Static Dynamic Transport Project location: Crusers variovam AppData voor	aming\CervenkaConsulting\Ate	naStudio\Pre	eprocesso	or

Fig. 4-4 Project settings menu.

The analysis type definition must be done before starting input of any data. Executing this command later may result in the loss of some of the existing data.

Other general settings for the project can be specified in the Project settings such as 3D/2D/Axisymmetric space type, units, location of the project etc.

However, at the current state of development of **ATENA-PRE** the Static analysis is supported only.

5 DEFINITION OF GEOMETRY

There are more options how to create geometrical model in ATENA-PRE. The model can be created directly by defining points, lines, surfaces and volumes. Within the interface the definition can be done in two ways. First one utilizes the direct definition of geometry parts included in Input tree, see Fig. 5-5, and various options for geometry definition in commands panel. Or the second option is scripting of the geometry definition that enables to define complicated shapes with mathematical expressions, more info including the script sample is in chapter 12.



Fig. 5-5 Input data tree - Geometry.

Another way to define the geometry is to import it from other CAD system in IFC of IGES format. The import of ATENA project from older version ATENA 2D and 3D is supported as well along with the import for Gid project, see the Import in menu bar Fig. 5-6.

		Data Mark				
File	Edit View Geometry	Data Mesh	Settings Help lestPP lestIA lest			
	New Project	Ctrl+N	0 0 0 0 0 0			
B	Open Project	Ctrl+O	AIN SEL 🗢 🍊 🖊			
	Open Recent Project		등 🕾 🔚 🦯 🧔			
H	Save	Ctrl+S				
围	Save As	Ctrl+Alt+S				
	Import	•	Import From ATENA 2D			
	Export	Þ	Import From ATENA 3D			
	Exit	Alt+F4	Import From IFC			
	2D Interfaces		Import From IGES			
	3D interfaces		Import From GiD •			
Materials			Import From INP			
	Reinforcement					
	Interface		Import Printer 3d model			

Fig. 5-6 Import of data into ATENA-PRE.

Creating entities directly in ATENA-PRE has many possibilities. The new point can be

created with Create point button in commands panel or button Point in table of specifications then the Point create settings appear on the side where the coordinates can be specified directly or user can just add point with mouse click in the View window, see Fig. 5-7.

Within commands panel there are buttons to create line, circular arc, nurbs curve, polygon surface, surface between two curves, solid from boundary surfaces, solid from two opposite surfaces, 2D interface and 3D interface

Eile Edit View Geometry Data Mesh Settings Help TestPP TestTA TestMZ TestCC TestCC.GL
Global parameters
▲ Geometry (2) 88 to → ffT × X+1 X+1 Y+ Y-1 T I = 1 (8 (8) 1/2 + 1 (9) Name:
Points (2) Point coordinates
Curves
Sundes
2D interfaces Y: -0.1
3D interfaces Z
A Materials (5)
Concrete (1)
Reinforcement (1)
Interface (1)
Sping (i) OK Cancel Apply
Steel(1) Point N. V Shi X Y Z Pare La Imj BE/ Pc
Elastic Force to geometry 1 220 L0
A Building element types
Unspecified
Beam X Delete
Column
Input data Mesh generator Points table Mesh out Mesh msg Mesh err Script history Layers Point Create Settings Mesh settings
AutoSave was saved in "C:\Users\vaitovam\App[^
i Start creating points X: -1.90000 Y: -0.10000 Z: 0.00000 Snap: ☑ Grd _ Points _ Curves _ Surfaces _ Solids _ ₹(0)
Start creating points
Place point (write coordinates separated by \ or id of

Fig. 5-7 Creation of point entity.

Also the scripting can be used for entity definision. Enter Scripting history, write the script command to define point and press execute the script, the geometry will we created, Fig. 5-8.

Execute	Execute 11 material(name = "Interface", prototype = "Interface", generator = "Main") 12 12 materialGeneratorRun(material = "Interface") "Interface")		
Import	Import 13 point (location = [2.2,-5.3,0]) # id = 1		
Export Copy to clipboard		>	
Points table Mesh ou	ut Mesh msg Mesh err Script history		

Fig. 5-8 Script history.

There are of course tools to make changes to existing geometry where used can select particular part and change parameters in geometry settings or make groups of



Standard tools like mirror, translation, rotation or dividing of existing entities to cerate

6 MATERIALS

ATENA analyses offer various material models where user can define parameters based on standards or directly from experimental results on test samples.

Within the Input data tree (Fig. 6-9) the Materials marker sets out following material models:

- Concrete
- Reinforcement
- Interface
- Spring
- Soil-Rock
- Steel
- Elastic.

More details about respective material is summarized in Table 1.



Fig. 6-9 Input data tree - Materials.

User chooses material model from the Input data tree to define material that will be later assigned to geometrical entities. Generally ATENA is developed to simulate nonlinear behavior of reinforced concrete structures, therefore Concrete material is often assigned to 2D and 3D entities and reinforcement is mostly represented by 1D entities. Material is assigned to geometry in Building element types which comes after Materials in the Input tree.

ATENA-PRE name	ATENA name (INP command)	Description					
	Concrete						
Cementitious2	CC3DNonLinCementitious2	Materials suitable for rock or concrete like materials. This material is identical to 3DNONLINCEMENTITIOUS except that this model is fully incremental. Material properties can be generated according the EC2, ModelCode (FRC), SP63.					
	CC3DNonLinCementitious2 FRC	The material is suitable for fibre reinforced concrete.					
	CC3DNonLinCementitious2Fatigu e	This material is based on the CC3DNonLinCementitious2 material, extended for fatigue calculation.					
	CC3DNonLinCementitious2WithTe mpDepProperties	This model is to be used to simulate change of material properties due to current temperature. The temperature fields can be imported from a previously performed thermal analysis.					
Cementitiou2 RadWaste	CC3DNonLinCementitious2RadW aste	Material model is developed for concrete structures exposed to radiation. The effect of neutron and gamma radiation dose is reflected.					
Cementitious2 SHCC	CC3DNonLinCementitious2SHCC	Strain Hardening Cementitious Composite material. Material suitable for fiber reinforced concrete, such as SHCC and HPFRCC					
	CC3DNonLinCementitious2SHCC WithTempDepProperties	materials. Identical to CC3DNonLinCementitious2User except for the shear response definition.					
Cementitious2 User	CC3DNonLinCementitious2User	Materials suitable for rock or concrete like materials. This material is identical to CC3DNonLinCementitious2 except that selected material laws can be defined by user curves.					
Cementitious2 Variable	CC3DNonLinCementitious2Variabl e	In this material model the parameters of material behavior are developing in time.					
Cementitious3	CC3DNonLinCementitious3	Materials suitable for rock or concrete like materials. This material is an advanced version of CC3DNonLinCementitious2 material that can handle the increased deformation capacity of concrete under triaxial compression. Suitable for problems including confinement effects.					
Reinforced Concrete	CCCombinedMaterial	This material can be used to create a composite material consisting of various components, such as for instance concrete with smeared reinforcement in various directions. Unlimited number of components can be specified. Output data for each component are then indicated by the label #i. Where i indicates a value of the i-th component.					

Table 1: ATENA material collection in ATENA-PRE

Material From File	terial From File CCFromFile Material model defined by user directly material.inp file.						
Material Random Fields	CCMaterialWithRandomFields	Material definition to be used in connection with SARA software to model material with random spatial distribution of material parameters.					
Microplane	CCMicroplane7	Bazant Microplane material models for concrete					
SBETA Material CCSBETAMaterial Older version of the basic material concrete, only suitable for 2-D pl models		Older version of the basic material for concrete, only suitable for 2-D plane stress models					
	Reinforceme	ent					
Reinforcement	CCReinforcement	Material for discrete reinforcement – bars and cables. You can generate material properties according the EC2					
	CCCyclingReinforcement	This material is used for cycling loading					
	CCReinforcementWithTemp Dep Properties	Material suitable for analyses with temperature dependency					
	Interface						
Interface	CC2DInterface, CC3DInterface	Interface (GAP) material for 2D and 3D analysis.					
	Spring						
Spring Material	CCSpringMaterial	Material for spring type boundary condition elements, i.e. for truss element modeling a spring.					
	Soil-Rock						
Drucker Prager	CC3DDruckerPragerPlasticity	Plastic materials with Drucker-Prager yield condition.					
	Steel						
Steel Von Mises 3D	CC3DBiLinearSteelVonMises	Plastic materials with Von-Mises yield condition, e.g., suitable for steel.					
	CC3DBiLinearVonMisesWithTemp DepPropertiess	This model is to be used to simulate change of material properties due to current temperature. The temperature fields can be imported from a previously performed thermal analysis.					
	Elastic						
Elastic 3D	CC3DElastIsotropic	Linear elastic isotropic materials for 3D					

Concrete contains various version types of material model depending on the application of modelled structure, for more details about different versions see Table 1. In general, material Concrete is suitable for quazi-brittle materials such as concrete or rocks. When user creates specific Concrete material the dialog for material generation is used, see Fig. 6-10. The **Prototype** of material defines ATENA material model which parameters

can be adjusted in **Material properties** dialog. Default material is Cementitious2 that can be generated according to EuroCode 2, Model Code (FRC) or SP63, user can select one of the mentioned standards and based on the Strength Class, Safety Format, and in some cases Strength Type (sample shape) and specifics for FRC: Type of Fibres and Weight of Fibres. When user wants to use other Prototype, the checkbox **Change prototype** (for experts) has to be selected. Then the Prototype of Concrete material can be switched to differet version, see Fig. 6-11 and Fig. 6-11.

🔇 Generate materia	ı		_		×
Prototype	CC3DNonLinCementitious2 $~~$	Chang	e proto	type (for	expert)
Generate according	EuroCode2 ~				
StrengthClass SafetyFormat	12/15 ¥ Design ¥				
		C	K	Car	ncel

Fig. 6-10 Concrete – Generation of material.

Generate materia	al		-		×
Prototype	CC3DNonLinCementitious2 ~	🖌 Char	nge proto	otype (for	expert)
Generate according	CC3DNonLinCementitious2				
	Cementitious2_RadWaste				
StrengthClass	Cementitious2_SHCC Cementitious2_User Cementitious2_Variable CC3DNonLinCementitious3 CCCombinedMaterial		ОК	Ca	ncel
	MaterialFromFile MaterialRandomFields Microplane CCSBETAMaterial				

Fig. 6-11 Concrete – Selection of Concrete Prototypes.

Default Concrete material that is generated according to selected standard has Material properties divided in dialog window into tabs: Basic, Tensile, Compressive, Miscellaneous and Generated from, shown in Fig. 6-12.

The material prototype list box from the **Basic** tab allows to select the basic CC3DNonLinCementitious2, or CC3DNonLinCementitious2WithTempDepProperties, where some of the material values can depend on temperature, or CC3DNonLinCementitious2Fatigue for modelling high-cycle tensile fatigue.

The basic material parameters are defined in the **Basic** dialog – the Young's modulus of elasticity E, the Poisson's coefficient of lateral expansion, the strength in direct tension Ft, and the cylinder compressive strength Fc.

Material Properties		- 🗆 X
Name Concrete	G	
Basic Tensile Compressive	Miscellaneous <generated from=""></generated>	
Material_Prototype	CC3DNonLinCementitious2 ~	Crack opening law
Young's modulus	28.38013 GPa ¥	or totated fixed model
Poisson's ratio	0.2	a(w) for
Tension strength Ft	0.8515083 MPa 🖌 🔛	$f_1 < \frac{f_1}{f_1} = f_1$
Compression strength Fc	-10 MPa ~	tc
		we w
		OK Cancel

Fig. 6-12 Concrete – Basic material properties of Cementitious2.

The advanced parameters related to tension are defined at the **Tensile** tab: Fracture energy Gf, Fixed Crack coefficient (0 = rotated, 1 = fixed, more details you can find in ATENA Theory in section "2.1.6 Two Models of Smeared Cracks"), Crack Spacing, Tension Stiffening, Aggregate Interlock, manual definition of Shear Factor, and Unloading Factor (0 = the default unloading to the origin, 1 = unloading parallel to the initial elastic stiffness). The meaning of the parameters should be clear from the figures in the dialog and the help texts. For details on these (and also other) parameters, see the ATENA Theory Manual [1].

Crack Spacing option should be used when the element size is larger than the expected crack width. Typically, it should be used in reinforced concrete elements, and is equal to the expected crack spacing. In the simplest case, the spacing of ties or stirrups can be used to estimate its value.

Tension Stiffening - should be used only if reinforcement is present in the model. It defines a relative tensile stress minimal limit for cracked concrete. This means the tensile stress in the cracked concrete cannot drop below this relative level (i.e., ft times tension_stiffening).

Aggregate size for the calculation of **aggregate interlock** based on the modified compression field theory by Collins. When this parameter is set, the shear strength of the cracked concrete is calculated using the modified compression field theory by Collins. The input parameter represents the maximal size of aggregates used in the concrete material.

Shear factor that is used for the calculation of cracking shear stiffness. It is calculated as a multiple of the corresponding minimal normal crack stiffness that is based on the tensile softening law.

Unloading factor, which controls crack closure stiffness.

Material Properties	- 🗆 X
Name Concrete G	
Basic Tensile Compressive Miscellaneous <generated from=""></generated>	
Fracture energy Gf 0.00011049 MN/m Y Fixed crack 1 Image: Comparison of the second seco	Crack opening law σ_{f_1} $w_{f_2} = 5.14 \frac{G_f}{f_1}$
Activate aggregate interlock Agg size Activate shear factor Activate unloading factor	We we
	OK Cancel

Fig. 6-13 Concrete – Tensile material properties of Cementitious2.

The advanced parameters influencing the compressive response are defined at the **Compressive** tab:

Plastic Strain at peak load eps_cp (ε_c^p) – this corresponds to the total strain at Fc from which the elastic part should be subtracted.

Onset of Crushing Fc0 represents the linearity limit.

Critical Compressive Displacement wd is the crushing compressive displacement after which the compressive stress drops to zero.

The relative limit for reduction of compressive strength due to cracking **Fc Reduction** controls the limit of Fc reduction due to cracking.

The checkbox **Activate Crush Band Min** is used to specify the minimal value of the crush band size for compressive concrete crushing. Normally crush band size is calculated as a projection of the finite element size in the direction of the highest compressive stress. If this value is defined, the crush band size is limited to the provided value, i.e., if the calculated crush band size is smaller than the limiting value, the minimal values are used. It is recommended to define it as the smallest dimension of the compressive element. See ATENA Theory manual [1] chapter on Cementitious2 material for more details.

Katerial Properties				- 🗆 X
Name Concrete	G			
Basic Tensile Compressive	Miscellaneous	<generated from=""></generated>		
Plastic strain EPS CP Onset of crushing Fc0 Critical comp disp Wd Fc reduction Activate crush band min	0.001497641 -1.788167 -0.0005 0.5	MPa × m × int	Peak compressive strain $\sigma_{,\varepsilon}$ $\sigma_{,\varepsilon}$ f_{c0} f_{c0}	Compressive ductility $\int_{1}^{\infty} \int_{1}^{1} w_{0} \int_{1}^{\infty} f_{0}$
				OK Cancel

Fig. 6-14 Concrete – Compressive material properties of Cementitious2.

The **Miscellaneous** tab contains two additional plasticity-related parameters, the **Eccentricity Exc** defining the shape of the failure surface, and the **Direction of Plastic Flow Beta**, determining volume compaction (Beta<0) or expansion (Beta>0) during crushing, i.e., plasticization, and two general parameters: **Density Rho** (only used in dynamic analysis) and the coefficient of **Thermal Expansion Alpha** (only used when the thermal load is applied).



Fig. 6-15 Concrete – Miscellaneous material properties of Cementitious2.

🙆 Ma	terial Pro	perties				_		×
Name	Concrete	è		G				
Basic	Tensile	Compres	sive	Miscel	laneous	<ge< th=""><th>enerat</th><th>ed from></th></ge<>	enerat	ed from>
Mater	rial proto	type	CC3	DNonLi	nCemer	ntitiou	si 🗸	
Gener	rated acc	ording	Mo	delCode	•		\sim	
Streng	gthType		Cyli	nder-Ch	aracteri	istic	1	
Streng	gthValue		15			MPa	\sim	
Safety	/Format		Des	ign \vee				
					OK		(Cancel

Fig. 6-16 Concrete – Generated from info of Cementitious2.

The basic material parameters for one-dimensional reinforcement bars is defined in material **Reinforcement EC2**. The dialog Generate material creates stress-strain function for bars or tendons based on the reinforcement steel strength class, a few basic parameters (elastic modulus, characteristic yield strength, ...) and safety format, in Fig. 6-17.

Generate material		- D X
Prototype CCReinforcer Generate according EC2	nent ~	
Type_of_reinforcement Young's modulus Characteristic_Yield_Strength_f_xk Class_of_Reinforcement Epsilon_u_k Parameter_k SafetyFormat	Bar	$f_{m} = f_{k} \cdot 1.1$ $f_{k} = f_{k} / 1.15$ Mean Characteristic Design
		$f_{x} I E = e_{ud} = 0.9 \cdot e_{uk} E_{bk} \gtrsim E$ $f_{m} = f_{pm} = f_{pm} \qquad f_{vd} = f_{pd} = f_{pd} \qquad f_{sk} = f_{pk} = f_{p \ 0.1k}$ OK Cancel

Fig. 6-17 Reinforcement – Generate material.

When material is generated, the stress-strain function can still be edited by user in dialog window in Fig. 6-18 and Fig. 6-19. Also following material prototypes developed for cycling loading or temperature dependent properties can be selected.

CCCyclingReinforcement - Material for cyclic reinforcement. There is a tab Menegotto-Pinto where special parameters can be defined. Detailed information about these parameters can be find in ATENA Theory Manual [1], section 2.7.5.

CCReinforcementWithTempDepProperties - This model is used to simulate change of material properties due to current temperature. The temperature fields can be imported from a previously performed thermal analysis. Reinforcement parameters can be generated according to the production method.

Katerial Properties	– 🗆 X
Name Reinforcement	G
Basic Miscellaneous <g< td=""><td>enerated from></td></g<>	enerated from>
Type_of_reinforcement	Bar ~
Material_Prototype	CCReinforcement ~
Young's modulus	200 GPa ~
Reinf_function	1_Reinf_function ~ Edit
	OK Cancel

Fig. 6-18 Reinforcement – Material properties edit dialogue.



Fig. 6-19 Reinforcement – Edit function – stress-strain diagram.

The **Interface** material (also called GAP) has been developed to model behavior of contacts between volumes, e.g., concrete - steel or thin layers of, e.g., mortar. This material should only be assigned to *contact volumes* (in 3D) or *contact surfaces* (in 2D and axisymmetry), in Fig. 6-20 and Fig. 6-21.

Material Properties	G		– U
Basic Miscellaneous <genera< th=""><th>ated from></th><th></th><th></th></genera<>	ated from>		
Material_Prototype	CCInterface V	2	Failure Criterion Stress-Displacement Law
Normal stiffness K NN	2E+08	MN/m²/m ~	+ t T, T,
Tangential stiffness K TT	2E+08	MN/m²/m ~	C-00 12
Cohesion	1	MPa 🖌 🔛	
Friction coeficient	0.1		
Tension strength Ft	0.3	MPa 🖌 🔛	
 Elipse in tension User defined softening hard 	dening		

Fig. 6-20 Interface – Basic parameters.



Fig. 6-21 Interface – Micsellaneous.

Spring material can be used for point, surface and line springs. Spring can be linear and also nonlinear. The linear spring are dfined with stiffness while nonlinear ones are defines by force-deformation diagram, Fig. 6-22.

Katerial Properties		- 🗆 X
Name Spring	G	
Basic <generated from<="" td=""><td>></td><td></td></generated>	>	
Material_Prototype	CCSpringMaterial ×	Spring stiffness
Spring non-linearity	NONLINEAR Y	Spring seminess
F of Eps diagram	1_F of Eps diagram 👻 Edit	F.
		u
		OK Cancel

Fig. 6-22 Spring – Basic material properties.

Soil-Rock material is useful for defining subsoil. It utilizes Drucker-Prager plasticity model that can be generated by combination of friction angle and cohesion or by compressive and tensile strength, see Fig. 6-23. Generated parameters of model can be adjusted, Fig. 6-24.

🔇 Generate material		_		×
Prototype		Soil-Rock		\sim
Generate according		From_Fi_and_C		~
Generate_DP_Cone	In	From_Ft_and_F	с	
Input_Friction	30)		
Input_Cohesion	2		MPa	~
		OK	Ca	ancel

Fig. 6-23 Soil-Rock – Generate material dialog.

🔇 Material Propertie	es			_		×
Name SoilRock	G]				
Basic Compressive	Miscellaneous	<generated from=""></generated>				
Young's modulus Poisson's ratio Prager alpha DP Prager param K	30320 0.2 0.1649572 1.714286	MPa ~	Failure function in hydrostatic plane $\rho = 12 J_2$ $f_0 \circ f_2 k$ $\xi = (a_1 + a_2 + a_3)/3$	Biaxial F	ailure I	aw ⊈ ft
				OK	Can	cel

Fig. 6-24 Soil-Rock – Material parameters.

Material Steel includes two material prototypes, the "normal" **CC3DBilinearSteelVonMises** and a variant supporting thermal degradation **CC3DBilinearVonMisesWithTempDepProperties**. It is also possible to define cyclic properties through enabling the option Activate Cyclic Params.

This material model is targeted for volume members made from steel and other similar (ductile) materials, i.e., elements which can not be well represented with the 1D reinforcement (truss) elements. Note the hardening is not limited, which means it has to be checked in post-processing if the ultimate strain has been exceeded.

Katerial Properties				-		×
Name Steel	G					
Basic Miscellaneous <ge< td=""><td>enerated from></td><td></td><td></td><td></td><td></td><td></td></ge<>	enerated from>					
Material_Prototype	CC3DBiLinearS	iteelVonMises 👻	Stress-Strain Law	Biaxial H	ailure L	aw
Young's modulus	200	GPa 👻	+0	σ,	1 ⁰ 2	
Poisson's ratio	0.3		oy J H	/		~
Yield strength Ys	200	MPa 🖌 🔛		-G.		1
Hardening modulus Hm	1000	MPa 🗠 🔛		y (У
Activate cyclic params				****	-f-σ _y	
				OK	Can	cel

Fig. 6-25 Steel – Material parameters.

Last of material choices in ATENA-PRE is **Elastic**. The material is defines by Young's modulus and Poisson's ratio, there is selection of various common materials like concrete, masonry, wood etc. that are available with their respective properties, Fig. 6-26 and Fig. 6-27.

🙆 Generat	e material			×
Prototype Generate according		Elastic	~	
		Maîn	Ŷ	
Material	Concrete 💙			
	Concrete	1		
	Masonry Wood	ОК	Car	ncel
	Steel			
	110-040300-01			
	Glass			

Fig. 6-26 Elastic – Generate material dialog.

🙆 Ma	terial Properties	_		×
Name	Elastic	G		
Basic	Miscellaneous	<generated from=""></generated>		
Young	g's modulus	30		GPa ~
Poisso	on's ratio	0.2		
		ОК	Ca	ancel

Fig. 6-27 Elastic – Material parameters.

7 BUILDING ELEMENT TYPES

Building element types interconnects model features including geometry, material and finite elemet types. The main Building element types are directly in the Input data tree (Fig. 7-28) while other types are available to choose in Unspecified Building element type, see complete list of options in Fig. 7-29. Each Building element type is assigned to geometrical entity along with specific material that has been defined by user in Material section. Based on the type of entity (point/surface/volume) the finite element properties are offered to adjust. The properties mainly include type of finite element, type of FE mesh and other idealization specifics when the user works with beam or shell approximation, or e.g. plane-stress or plain-strain etc.



Fig. 7-28 Input data tree – Building element types.



Fig. 7-29 Building element types options.

BE type properties differ for line, surface or volume, Fig. 7-30. 1D entites (lines) use beam static model with quadratic mesh. Beam cross-section is defined by one material and the reinforcing bars can be added to it.

For 2D entities (surface), user can choose between static model with 2D elements with thickness or layered 2D shell elements. The plane-stress or plain-strain idealization, and linear or quadratic mesh are combined with 2D elements with thickness. Thell elements are quadratic only and the shell layers can combine different materials.

Finally 3D entities (volumes) offer three static models: 3D solid, Beam or Layered shell. Solid offers both linear and quadratic mesh while beam and shell use just quadratic mesh. Beam and shell can combine base material with reinforcement in various cross-section shapes.

🔇 Create	new buildir	ig element	type		—		×
Name	Unspecified				✓ /	Automatio	: name
Basic Be	eam FE Mo	del IFC					
Geometr	ic entity			20	A		
Static mo	odel	Beam 1D)	ركا ي:			~
Mesh typ	e	O Linear				Qua	adratic
BE type		Unspecif	ied				~
				<u>O</u> K		Cano	el
	Create	new buildin	ıg elemei	n —	() ×	<
	Name	Inspecified		~] Autor	natic nam	ne
	Basic FE	Model IF	С				
	Geometri	c entity		/] Ø		
	Static mo	del	2D (su	rface with	n thickn	ess) ~	
	Idealisatio	on	Plan	e stress	🔿 Pla	ne strain	
	Mesh typ	e	⊖ Line	ar	Qu	adratic	
	Assigned	material	Concre	ete1		~	
	BE type		Unspe	cified] [~ ~	
	THICKNESS		I			in ·	
				<u>О</u> К		Cancel	
	🔇 Creat	e new build	ding el	_		×	
	Name	Unspecifie	ed	✓	Automa	atic name	2
	Basic (E Model	IFC				
	Geomet	ric entity		/	8	1	
	Static m	odel	3D	(solid)		~	
	Mesh ty	pe	ΟL	inear	•	Quadratic	
	Assigne	d material	Stee	el		~	
	BE type		Uns	pecified		~	
			<u>C</u>	<u>)</u> K	Ca	ancel	

Fig. 7-30 Building element types – 1D/2D/3D properties.

FE Model options offer choice og geometrical linearity of the model, more information can be find in input file manual [4] also in Fig. 7-31. Along with excetions that can be triggered during analysis and terminate the calculation, the choice of ignoring them leads to disputable results but might be used in specific cases e.g. to get results when the structure is already heavily damaged.

🔇 Crea	te new bui	ding elem	nent type	_		×
Name	Unspecifi	ed		✓ A	utomatic	name
Basic	FE Model	IFC				
Lineari	ty		Linear			~
Ignore	negative ja	cobian				
Ignore	element ty	pe except	ior 🗌			
Used F	E typesss		CCIsoQu	iad8x, C	ClsoTriar	ngle6x
			<u>O</u> K		Cano	el

Fig. 7-31 Building element types – FE model.

8 **REINFORCING TYPES**

Reinforcing types serves for definiton of properties of 1D reinforcement elements their bond properties, cross section area of bar/tendon or the whole region of reinforcement, and also properties for FE meshing, Fig. 8-32.

Claculator of cross-section area and its perimeter is included as well.

🙆 Crea	ate rein	forcing e	lement	type	—		×
Name	Bar				✓	Automat	ic name
Basic	Bond	FE Mod	el IFC				
RE typ	e		Bar				~
Bond	type		Bond				~
Assign	ied mat	erial	Reinforcement10				
Calcul	ator		✓				
Bar	diamete	er	12			ſ	mm ~
Nun	nber of	profiles	1				
Cross	Cross section area		113.1			m	nm² ~
Profile	Profile perimeter					ſ	mm Y
				<u>O</u> K		Car	ncel

Fig. 8-32 Reinforcing element types – Options.

There are three RE types bar, tendon and region of reinforcement. Bar and tendon allow to assign bond properties and in Bond tab specify maximum bond strength with bond slip function that can be also generated according to Bigaj or Ceb Fib Model Code, Fig. 8-33.

Create reinforcing element	type —				
Name Bar	✓ Aut	omatic name			
Basic Bond FE Model IFC					
Bar end	Fixed start	~			
Maximal bond strength	10.46 MPa ~				
Function for bond strength	1_Function_for_Bond_Strength	Edit			
Generate bond law					
Dependency functions					
	<u>O</u> K	Cancel			

Fig. 8-33 Reinforcing element types – Bond settings.

9 SPRING AND INTERFACE TYPES

As in previous chapters where material and finite element properties have been assigned to geometry, Spring and Interface types serve the same purpose just for spring and interface materials.

Spring types can be assigned to point, lines and surfaces, Fig. 9-34. In all cases the mesh of spring types is linear. Based on the geometric entity the spring is specified by cross-section area, thickness or just length. The orientation of the spring can be defined in the local or global system. More spring can be assigned by activating another spring that has different orientation or material.

Create new spring		- 0	×
Name Spring		✓ Autom	atic name
Basic FE Model			
Geometric entity	0		
Static model	Surface spring		
Mesh type	Linear	🔵 Qua	dratic
Activate Spring 1	\checkmark		
Assigned material	Spring1		~
Coordinate system	Global		al
Direction	X : 1 Y : 0	0 Z : 0	
Length	1		m ~
Activate Spring 2			
Activate Spring 3			
L	<u>O</u> K		Cancel

Fig. 9-34 Spring types.

Interface is applicable to surface and volume entities, Fig. 9-35. The mesh assigned to interface types can be linear or quadratic and for 2D entities the interface thickness has to be specified.

Create new interf	ace — 🗆 X				
Name Interface	✓ Automatic name				
Basic FE Model					
Geometric entity					
Static model	3D (solid)				
Mesh type	● Linear ○ Quadratic				
Assigned material	Interface1 ~				
[<u>O</u> K Cancel				

Fig. 9-35 Interface types.

10 FE MESH PARAMETERS AND **M**ESH GENERATOR

The finite element mesh quality has a very high influence on the quality of the analysis results, the speed, and memory requirements. Refining only the important parts can save a lot of processor time and disk space.

Partially the finite element mesh settings are described in previous chapters mainly the finite element types and its definition, however in this chapter the mesh size, shape of finite elements and mesh compatibility are presented, Fig. 10-36.

FE mesh parameters allows to define mesh settings in specific parts of the model while general settings that are applied on the whole structure are adjustable in Mesh generator. ATENA-PRE uses T3D mesh generator that is developed by Daniel Rypl [5].



Fig. 10-36 Input data tree – FE mesh parameters.

First of mesh settings in Fig. 10-37 is mesh size where size of mesh can be default, uniform or user defined. Then there is equidistant possibility to ensure the uniform mesh size within selected area. Finally the factor option is used to specify the vertex mesh size multiplication factor, the default value is 1 and it must not be negative.

🔇 FE me	K FE mesh parameters				×	
Name	FEMeshSizeS	✓ A	utoma	tic name		
Basic						
Mesh si	ze definition	Default (global)				
Size		0.05			m ~	
Equidis	tant					
E Fact	tor					
		<u>O</u> K		Ca	ncel	

Fig. 10-37 FE mesh parameters – Size of mesh.

Following mesh settings is mesh type for surfaces and solids Fig. 10-38 and Fig. 10-39. Basic option for surface mesh specifies quad-dominant mesh and whether the mesh is structured or not. Structured mesh is usually better for identification of elements and nodes and gives mostly better results than unstructured. However some complicated geometries do not allow to use structured mesh. I case of solids, the static model is specified once more to ensure compatibility with BE types.

🔇 FE me	esh parameters		—		\times		
Name	FEMeshTypeSu	rface	✓ /	Automatic	name		
Basic							
Quad-d	lominant mesh	✓					
FE mest	n structured	Not us	sed		~		
🗌 Diag	gonal						
Only for polygonal or planar surfaces							
		<u>O</u> K		Cano	el		

Fig. 10-38 FE mesh parameters – Mesh type for surface.

🔇 FE me	sh parameters	_		×			
Name	FEMeshTypeS	✓ A	utomatic	name			
Basic							
Static mo	odel	3D (so	olid)		~		
Hexa-do	minant mesh	✓					
FE mesh	structured	Not us	lot used 🗸 🗸				
🗌 Iso							
	L	<u>O</u> K		Cano	el		

Fig. 10-39 FE mesh parameters – Mesh type for solid.

Section of FE mesh parameters advanced is dedicated to connecting of model part that contain incompatible mesh that might be given by different element shape or type, basically mostly at places where finite element nodes are not matching on the connecting point/line/surface, Fig. 10-40.

FE m	esh parameters	ĺ.			×	
Name Basic	FEMeshAdvar	icedPoint	7 A	utomatio	name	
Slave / master		Not used	I		~	
0.		Not used	I			
	Ok & Assign g	Slave Master				
	202	No slave No master				
	XX	Slave, no	maste	r		
		Master, no slave				

Fig. 10-40 FE mesh parameters – Master-slave connections.

More detailed information on meshing and settings of T3D mesh generator are available in [5].

ile	Edit	View Geometry Data Mesh	Se
P		🗒 🛃 🛹 🍌 🍳 C	
	Silent m	node	^
	Suppres	ss warning messages	
	Include	virtual entites	
-	Enable	curvature violation	
-	Dicablo	default designation	
	Disable	nodal smoothing	
1	Disable	quad and hexa meshing	
	Enable	convexity check	
	0.05	Default mesh size [m]	
	0.5	Uniform mesh size [m]	
	1E-05	User defined epsilon [m]	
	1	Mesh size mult. factor	
	1	Default curve density	
	1	Default curvature rate	
	1	Curvature rate mult. factor	
	Linear	 Element degree 	
	Discor	nnect V Mid nodes conflict	
٣) Mesh (quality report	
\odot) Weigh	ting for smoothing	
0	Additio	anal output	
C	Auditio	Shar output	
_	Surface:	s as patches	
=	All face	s mesh as quads	
=	All face	s structured mesh	
1		one mach as hevas	
-	All regio	ons structured mesh	
	1		
Ge	ometry:		
	1	Preview T3D geometry	
- 1		Preview FE mesh	
-	S	how T3D geometry inp	
[
		Show FE mesh inp	

Fig. 10-41 FE mesh parameters – Mesh generator.

11 ANALYSIS

When geometry of the model has assigned material and all necessary parameters of FE mesh, analysis data can be created. Analysis data are divided into categories: Load cases, Boundary conditions, Monitors, Solution parameters, Tasks and Functions.

Load cases serve for creating the load types (forces, shrinkage, prestressing etc.) acting on the geometrical entity including supports, see Fig. 11-42 and Fig. 11-43. Load cases are then combined in Tasks to create the loading history of the whole analysis.

When the Load Cases and their load types are prepared, the specific Boundary conditions can be assigned to entites with buttons in table of specifications • 🖉 🗗 🖸 🕻 in Fig. 11-44.

Load case	Id 🔳	Name	Show	Color	Description	Load type	Load categ	Load case multiplier	Boundary conditions
Edit Id=4] 4	Interval 1	-	-		General	Undefined	1	1, 59, 60, 61, 70, 71, 72, 7
X Delete Id=4	5	Interval 2	v	~		General	Undefined	1	2, 62, 63, 64, 65, 66, 67, 6
Boundary conditions:	6	Interval 3	v	•		General	Undefined	1	3, 31, 86, 87, 88, 89, 90, 9
• / 0 0	7	Interval 4	~	~		General	Undefined	1	4, 32, 94, 95, 96, 97, 98,
	8	Interval 5	-			General	Undefined	1	5, 33, 102, 103, 104, 105
	~	h e	1 _		1	- ·		l.	· · · · · · · · · · · · · · · · · · ·

Load cases table Mesh out Mesh msg Mesh err Script history

Fig. 11-42 Load cases – Table of specifications.

\delta New	Load Case	-		×			
Name:	Forces (2)	✓ Automatic name					
Descrip	tion (optional)	:					
Load ty	pe:	Forces		,			
Load ca	teaorv:	Body force					
Loud category.	Supports						
LUdu Ca	se multiplier:	Forces					
		Prescribed deformation					
		Temperature					
	02402402	Shrinkage					
		Prestressing	9				
		ASR					
		Corrosion					
		Snow					
		Wind					
		NATM supp	port				
		General					

Fig. 11-43 Load cases – Load types.

New Solid Boundary Conditions ×								
Name:	Weight (4) 🖌 Automatic name							
Descript	ion (optional):							
Load cas	se:	4	Interval 1	~				
Conditio	on type:	Weight						
		Weight						
U We	ight in X:	Fixe	d contact	- 1				
We We	ight in Y:	Fixe	d 1d beam to solid	- 1				
We We	ight in 7:	Temperature						
	.g	Hum	nidity	- 1				
S	ave & Assign geom	Initial temperature						
		Initial humidity						
		Initial gap load						
98983		Initial strain						
223X	i si	Initia	al stress					
		ASR						
833333	*****	Selection nodes						

Fig. 11-44 Boundary conditions – Types of boundary conditions.

Then there are Monitors which are very helpful for evaluation of the whole analysis. Monitors for various types of load or reactions of the structure resulting from acting load are available in table of specifications for different geometrical entities with

Nonlinear analyses in ATENA generaly offer two different type of solution method which are Newton-Raphson and Arc-length method, in Fig. 11-46. Both method have their strength and limitation therefore the proper one needs to be set to find the correct solution, more information on methods and their detailed settings is available in ATENA Theory [1]. The settings of method and other solution parameters are within the Solution parameters.

The main part of Analysis section is Task, there all the previous parts come together and the loading history for the analysis can be created. Within Loading history separate intervals are defined and their specifics are adjustable in Interval parameters. The solution parameters and combination of load cases are defined for each interval separately including the number of steps, interval multipliet etc. Finally the diagram of load cases action is at the bottom of Tasks settings, see Fig. 11-47. Different loading history can be defined in another Task.

Once the loading history and solution of analysis is set the INP file can be created or just previewed, or the analysis can be executed with Run button, see Fig. 11-48.

New Solid Monitor		;				
Name: Monitor (62)	✓ Automatic nam	ne				
Description (optional):						
Output data type:	Displacements	•				
Direction X	Displacements	~				
Direction Y	External forces					
Direction Z	Internal Forces					
Component 4	Reactions					
Component 5	Stress Strain					
Component 6						
Draw each iteration						
Math operation:	Principal stress					
	Principal strain					
Save & Ass	Principal plastic strain					
	Stress (R1)					
	Stress (R2)					
	Stress (K3)					
3996833993	SUESS (N4)	1				

Fig. 11-45 Monitors – Data types.

Name	Solution paramet	ers	✓ Automa	tic nam				
General	Line Search Co	onditional l	ditional Break Criteria					
Descripti	on (optional)							
Solution	method	Newto	n-Raphson	Y				
Optimize	band width	Sloan	Sloan					
Solution	method subtype	Modifi	Modified N-R					
Stiffness	matrix update	Each st	Each step					
Stiffness	type	Elastic	Elastic					
Iteration	limit	60	60					
Linear so	lver	PARDIS	PARDISO					
Extend a	ccuracy factor	2	2					
PARDISO	required accurate	y 1E-08						
Negligib	le size type	Relativ	Relative					
Negligib	le size relative	1E-06						
Displace	ment error	0.01	0.01					
Residual	error	0.01	0.01					
Absolute	residual error	0.01						
Energy e	rror	0.0001	0.0001					

Fig. 11-46 Solution parameters – Settings of solution method and other criteria.

Edit Task	#1								
Name:	Task - Static analysis	Automatic name	Interval parameters						
Loading Task int	History Transport Analysis Restar ervals: Interval 1 - self weight Interval 2 - prestress Interval 3 - displ - 0.25 Interval 4 - displ - 0.25 Interval 4 - displ - 0.25	t	Name:	Interval 1 - self weight 3 From problem data 1 - 1 (1 steps) • •	· · ·	Interval load cases	Load type 1 General	Interval steps [AII]	LC function Multiplier
 ✓ 6 ✓ 7 ✓ 8 ✓ 9 	Interval 6 - displ -0.375 Interval 7 - displ +0.5 Interval 8 - displ +0.5 Interval 9 - displ +0.75 Interval 9 - displ +0.75	10 10 20 20 Copy Delete rval 3 Interval 4	Eigenvalue Construction process terval 5 Interval 6	Interval 7 Interval 8	nterval 9	Load cases:	1 Body force	tterval 13 <mark>ma</mark> interval 14	Add Add all as active Add all as active Add all as inactive Delete selected Interval 15 Interval 16 Interva
Final Multiplier		0.1 0.15	0.2	0.25 0.3	0.35	0.4	0.45	0.5 0.55	0.6 0.65 Time [sec]
. writedill	and and a sume a step L	, points							Save Cancel

Fig. 11-47 Tasks – Creating the loading history.

Task		✓ Id	Name	Description	Intervals	Steps	Start time	Duration	End time	
Edit Id=	:1	1	Task - Static anal	ysis	30	707	0	1.2	1.2	
🗙 Delete	ld=1									
Preview I	NP									
Save INP	as									
Run										
		<								
Tasks table	Mesh or	ut M	esh msg 🛛 Mesh	err Script histe	ory					

Fig. 11-48 Tasks – INP preview/save, Run of analysis.

Finally there is the section with defined and saved Functions. The can be find all the general functions that are utilized for the whole model. For example the function of material laws like strass-strain diagrams, material properties dependent on time/temperature/radiation, function that defines application of load during the loading history and many others, example of function dialog in Fig. 11-49.



Fig. 11-49 Functions – Definition of function.

12 SCRIPTING IN ATENA-PRE

Scripting of geometry is great advantage to create complicated geometry that can be expressed mathematically. Example of such scripts and its visual results are presented bellow.

The scripting method is especially useful for instance for the parameteric modelling of standard structural elements such as for instance the spiral reinforcement as indicated below.

Working with scripts is enabled in table of specification at the last tab Script history in Fig. 12-50. The script can be written and executed directly in Script history or user can import or export existing scripts.



Fig. 12-50 Script history.

EXAMPLE 1 – Spiral reinforcement, Fig. 12-51.

```
# First spiral
# _____
#define global parameters
globalParameter(equation = "diameter = 0.1") # id = 1
globalParameter(equation = "radius = diameter / 2") # id = 1
globalParameter(equation = "loopHeight = 0.01") # id = 2
globalParameter(equation = "loopsCount = 10") # id = 3
globalParameter(equation = "originX = 0") # id = 4
globalParameter(equation = "originY = 0") # id = 5
globalParameter(equation = "originZ = 0") # id = 6
```

#create first loop from four circular arcs

circularArc(startPoint = ["originX + radius", "originY", "originZ"], pointOnInterior = ["originX +
(math.cos(0.78539816339744828) * radius)", "originY + (math.sin(0.78539816339744828) * radius)",
"originZ + loopHeight / 8"], endPoint = ["originX", "originY + radius", "originZ + 2 * loopHeight / 8"], createCircle = "False", createCircularSurface = "False") # id = 2

circularArc(startPoint = 2, pointOnInterior = ["originX + (math.cos(2.3561944901923448) * radius)", "originY + (math.sin(2.3561944901923448) * radius)", "originZ + 3 * loopHeight / 8"], endPoint =
["originX - radius", "originY", "originZ + 4 * loopHeight / 8"], createCircle = "False",
createCircularSurface = "False") # id = 2

circularArc(startPoint = 3, pointOnInterior = ["originX + (math.cos(3.9269908169872414) * radius)", "originY + (math.sin(3.9269908169872414) * radius)", "originZ + 5 * loopHeight / 8"], endPoint
["originX", "originY - radius", "originZ + 6 * loopHeight / 8"], createCircle = "False"
createCircularSurface = "False") # id = 2 "False",

circularArc(startPoint = 4, pointOnInterior = ["originX + (math.cos(5.497787143782138) * radius)", "originY + (math.sin(5.497787143782138) * radius)", "originZ + 7 * loopHeight / 8"], endPoint =
["originX + radius", "originY", "originZ + 8 * loopHeight / 8"], createCircle = "False",
createCircularSurface = "False") # id = 2

#copy loop to get required loops count

translate(startPoint = [0,0.0,0.0], endPoint = [0,0.0,"loopHeight"], copies = "loopSCount - 1", collapse = "True", extrude = "False", curves = [1,2,3,4])

Second spiral

#define global parameters

globalParameter(equation = "diameter2 = 0.05") # id = 1

globalParameter(equation = "radius2 = diameter2 / 2") # id = 1

globalParameter(equation = "loopHeight2 = 0.015") # id = 2

globalParameter(equation = "loopsCount2 = 15") # id = 3

globalParameter(equation = "originX2 = 0.2") # id = 4

globalParameter(equation = "originY2 = 0") # id = 5

globalParameter(equation = "originZ2 = 0") # id = 6

#create first loop from four circular arcs

circularArc(startPoint = ["originX2 + radius2", "originY2", "originZ2"], pointOnInterior = ["originX2 + (math.cos(0.78539816339744828) * radius2)", "originY2 + (math.sin(0.78539816339744828) * radius2)", "originZ2 + loopHeight2 / 8"], endPoint = ["originX2", "originY2 + radius2", "originZ2 + 2 * loopHeight2 / 8"], createCircle = "False", createCircularSurface = "False") # id = 2

circularArc(startPoint = 52, pointOnInterior = ["originX2 + (math.cos(2.3561944901923448) *
radius2)", "originY2 + (math.sin(2.3561944901923448) * radius2)", "originZ2 + 3 * loopHeight2 / 8"],
endPoint = ["originX2 - radius2", "originY2", "originZ2 + 4 * loopHeight2 / 8"], createCircle =
"False", createCircularSurface = "False") # id = 2

circularArc(startPoint = 53, pointOnInterior = ["originX2 + (math.cos(3.9269908169872414) *
radius2)", "originY2 + (math.sin(3.9269908169872414) * radius2)", "originZ2 + 5 * loopHeight2 / 8"],
endPoint = ["originX2", "originY2 - radius2", "originZ2 + 6 * loopHeight2 / 8"], createCircle =
"False", createCircularSurface = "False") # id = 2

circularArc(startPoint = 54, pointOnInterior = ["originX2 + (math.cos(5.497787143782138) * radius2)",
"originX2 + (math.sin(5.497787143782138) * radius2)", "originZ2 + 7 * loopHeight2 / 8"], endPoint =
["originX2 + radius2", "originY2", "originZ2 + 8 * loopHeight2 / 8"], createCircle = "False",
createCircularSurface = "False") # id = 2

#copy loop to get required loops count

translate(startPoint = [0,0.0,0.0], endPoint = [0,0.0,"loopHeight2"], copies = "loopsCount2 - 1", collapse = "True", extrude = "False", curves = [41,42,43,44])

Third spiral

#define global parameters

globalParameter(equation = "diameter3 = 0.035") # id = 1

globalParameter(equation = "radius3 = diameter3 / 2") # id = 1

globalParameter(equation = "loopHeight3 = 0.04") # id = 2

globalParameter(equation = "loopsCount3 = 5") # id = 3

globalParameter(equation = "originX3 = 0.4") # id = 4

globalParameter(equation = "originY3 = 0") # id = 5

globalParameter(equation = "originZ3 = 0") # id = 6

#create first loop from four circular arcs

circularArc(startPoint = ["originX3 + radius3", "originY3", "originZ3"], pointOnInterior = ["originX3 + (math.cos(0.78539816339744828) * radius3)", "originY3 + (math.sin(0.78539816339744828) * radius3)", "originZ3 + loopHeight3 / 8"], endPoint = ["originX3", "originY3 + radius3", "originZ3 + 2 * loopHeight3 / 8"], createCircle = "False", createCircularSurface = "False") # id = 2

circularArc(startPoint = 127, pointOnInterior = ["originX3 + (math.cos(2.3561944901923448) *
radius3)", "originX3 + (math.sin(2.3561944901923448) * radius3)", "originZ3 + 3 * loopHeight3 / 8"],
endPoint = ["originX3 - radius3", "originY3", "originZ3 + 4 * loopHeight3 / 8"], createCircle =
"False", createCircularSurface = "False") # id = 2

circularArc(startPoint = 128, pointOnInterior = ["originX3 + (math.cos(3.9269908169872414) *
radius3)", "originY3 + (math.sin(3.9269908169872414) * radius3)", "originZ3 + 5 * loopHeight3 / 8"],
endPoint = ["originX3", "originY3 - radius3", "originZ3 + 6 * loopHeight3 / 8"], createCircle =
"False", createCircularSurface = "False") # id = 2

circularArc(startPoint = 129, pointOnInterior = ["originX3 + (math.cos(5.497787143782138) *
radius3)", "originY3 + (math.sin(5.497787143782138) * radius3)", "originZ3 + 7 * loopHeight3 / 8"],
endPoint = ["originX3 + radius3", "originY3", "originZ3 + 8 * loopHeight3 / 8"], createCircle =
"False", createCircularSurface = "False") # id = 2

#copy loop to get required loops count

translate(st collapse = "	artPoint True ", ex	= [0,0 xtrude =	.0,0.0], e "False",	endPoint = [0 curves = [101]	,0.0,"loc 102,103,	pHeight3"], 104])	copies =	"loopsCount	3 - 1",
COLLEADSE = " COLLEADSE COLLEADS	And Constant (10) - State() eth Settings Help Test Constant (10) - State() Constant (10) - State() Con	19 hetth feeds feed set						M\\$\$ K to € € d	Low C C C C C C C C C C C C C C C C C C C
	Script history	Forst spical Forst spical Forst spical Goldsulfar americinguit Go	fi on = "damedar" = 0.17 ± id = 1 on = "control = damedar(27 ± id = on = "control = damedar(27 ± id = on = "control = 0.17 ± id = 0. on = "control = 0.17 ± id = 0. on = "control = 0.17 ± id = 0. on = "control = 0.17 ± id = 0. Control = 1.00 ±	1. postocimitados - E posporo ciman constito 785 horazo 2019 de la constitución de la constitución de la constitución horazo 2019 de la constitución de la constitución de la constitución horazó 47787 14782 1187 validad de la constitución de la constitución horazó - A constitución de la constitució	855337746227va5ud1, org	nn sinentuum 7253314539746500 aastuur Yoogaa Pooperagina II aa aastuur Yoogaa Pooperagina II aa aastuur Yoogaa Pooperagina II aastu	nium noganz-neopenegrup (), ered Mari - Organz - Neopenegrup (), ered Mari - Organz - Neopenegrup - Securi - Org et - Congast - Reduct - Organi - Organi et - Congast - Reduct - Organi - Organi	est = torget, torget, establish grade = torget, torget, establish grade = torget, torget, torget, torget, torget, grade = torget, torget, torget, torget, torget, grade = torget, torget, torget, torget, torget, torget, grade = torget, torget, torget, torget, torget, torget, torget, grade = torget, torget, torget, torget, torget, torget, torget, grade = torget, torg	Lynn Commed String Meh string
Script generation finished s	Marcal 00.00.00 M See Inc.	wesh en	scopt motory					1	continiano secongs wesh secongs
 script execution trisined successfully AutoSave was saved in "C:\Atena_File 	enapsed 000000000000000000000000000000000000	kutoSave_2022-01-19-14-55	-46.pre"				Snap to: 🗹 Grid 🗌 Centers	X: 0.00000 Y: 0.00000 Z: 0.00000	Active layer 1



EXAMPLE 2 – Nurbs reinforcement, Fig. 12-52.

```
# ------
# First spiral
# ------
#define global parameters
globalParameter(equation = "diameter = 0.1") # id = 1
globalParameter(equation = "radius = diameter / 2") # id = 1
globalParameter(equation = "loopHeight = 0.009") # id = 2
globalParameter(equation = "loopsCount = 30") # id = 3
globalParameter(equation = "originX = 0") # id = 4
globalParameter(equation = "originY = 0") # id = 5
globalParameter(equation = "originZ = 0") # id = 6
#define points through them nurbs curve will go
point(location = ["originX+radius","originY","originZ"]) # id = 1
point (location
["originX+(math.cos(0.78539816339744828)*radius)","originY+(math.sin(0.78539816339744828)*radius)","o
riginZ+loopHeight/8"]) # id = 2
point(location = ["originX", "originY+radius", "originZ+2*loopHeight/8"]) # id = 3
point (location
["originX+(math.cos(2.3561944901923448)*radius)","originY+(math.sin(2.3561944901923448)*radius)","ori
ginZ+3*loopHeight/8"]) # id = 4
point(location = ["originX-radius","originY","originZ+4*loopHeight/8"]) # id = 5
point (location
["originX+(math.cos(3.9269908169872414)*radius)","originY+(math.sin(3.9269908169872414)*radius)","ori
ginZ+5*loopHeight/8"]) # id = 6
point(location = ["originX","originY-radius","originZ+6*loopHeight/8"]) # id = 7
point (location
["originX+(math.cos(5.497787143782138)*radius)","originY+(math.sin(5.497787143782138)*radius)","origi
nZ+7*loopHeight/8"]) # id = 8
```

point(location = ["originX+radius", "originY", "originZ+8*loopHeight/8"]) # id = 9 translate(startPoint = [0,0,0], endPoint = [0,0,"loopHeight"], copies = "loopsCount - 1", collapse = "True", extrude = "False", points = [1, 2, 3, 4, 5, 6, 7, 8, 9]) nurbsCurveThroughPoints(degree = 3, fromPoint = 1, toPoint = 270)# ------# Second spiral # ------#define global parameters globalParameter(equation = "diameter2 = 0.3") # id = 1 globalParameter(equation = "radius2 = diameter2 / 2") # id = 1 globalParameter(equation = "loopHeight2 = 0.08") # id = 2 globalParameter(equation = "loopsCount2 = 10") # id = 3 globalParameter(equation = "originX2 = 0.5") # id = 4 globalParameter(equation = "originY2 = 0.0") # id = 5 globalParameter(equation = "originZ2 = 0") # id = 6 #define points through them nurbs curve will go point(location = ["originX2+radius2","originY2","originZ2"]) # id = 1 point (location ["originX2+(math.cos(0.78539816339744828)*radius2)","originY2+(math.sin(0.78539816339744828)*radius2) ","originZ2+loopHeight2/8"]) # id = 2 point(location = ["originX2", "originY2+radius2", "originZ2+2*loopHeight2/8"]) # id = 3 point (location ["originX2+(math.cos(2.3561944901923448)*radius2)","originY2+(math.sin(2.3561944901923448)*radius2)", "originZ2+3*loopHeight2/8"]) # id = 4 point(location = ["originX2-radius2","originY2","originZ2+4*loopHeight2/8"]) # id = 5 point (location "originX2+(math.cos(3.9269908169872414)*radius2)","originY2+(math.sin(3.9269908169872414)*radius2)", "originZ2+5*loopHeight2/8"]) # id = 6 point(location = ["originX2", "originY2-radius2", "originZ2+6*loopHeight2/8"]) # id = 7 point (location ["originX2+(math.cos(5.497787143782138)*radius2)","originY2+(math.sin(5.497787143782138)*radius2)","o riginZ2+7*loopHeight2/8"]) # id = 8 point(location = ["originX2+radius2","originY2","originZ2+8*loopHeight2/8"]) # id = 9 translate(startPoint = [0,0,0], endPoint = [0,0,"loopHeight2"], copies = "loopsCount2 - 1", collapse "True", extrude = "False", points = [271, 272, 273, 274, 275, 276, 277, 278, 279]) nurbsCurveThroughPoints(degree = 3, fromPoint = 271, toPoint = 360) # ______ # Third spiral - points # ------#define global parameters globalParameter(equation = "diameter3 = 0.2") # id = 1 globalParameter(equation = "radius3 = diameter3 / 2") # id = 1 globalParameter(equation = "loopHeight3 = 0.2") # id = 2 globalParameter(equation = "loopsCount3 = 5") # id = 3 globalParameter(equation = "originX3 = -0.5") # id = 4globalParameter(equation = "originY3 = 0.0") # id = 5globalParameter(equation = "originZ3 = 0") # id = 6 #define points through them nurbs curve will go point(location = ["originX3+radius3", "originY3", "originZ3"]) # id = 1 point (location ["originX3+(math.cos(0.78539816339744828)*radius3)","originY3+(math.sin(0.78539816339744828)*radius3) ", "originZ3+loopHeight3/8"]) # id = 2 point(location = ["originX3","originY3+radius3","originZ3+2*loopHeight3/8"]) # id = 3

point(location =
["originX3+(math.cos(2.3561944901923448)*radius3)","originY3+(math.sin(2.3561944901923448)*radius3)",
"originZ3+3*loopHeight3/8"]) # id = 4
point(location = ["originX3-radius3","originY3","originZ3+4*loopHeight3/8"]) # id = 5
point(location = ["originX3-radius3","originY3","originZ3+4*loopHeight3/8"]) # id = 5

["originX3+(math.cos(3.9269908169872414)*radius3)","originY3+(math.sin(3.9269908169872414)*radius3)", "originZ3+5*loopHeight3/8"]) # id = 6

point(location = ["originX3","originY3-radius3","originZ3+6*loopHeight3/8"]) # id = 7

point(location =
["originX3+(math.cos(5.497787143782138)*radius3)","originY3+(math.sin(5.497787143782138)*radius3)","o
riginZ3+7*loopHeight3/8"]) # id = 8

point(location = ["originX3+radius3","originY3","originZ3+8*loopHeight3/8"]) # id = 9

translate(startPoint = [0,0,0], endPoint = [0,0,"loopHeight3"], copies = "loopsCount3 - 1", collapse = "True", extrude = "False", points = [361, 362, 363, 364, 365, 366, 367, 368, 369])

nurbsCurveThroughPoints(degree = 3, fromPoint = 361, toPoint = 405)

#delete intervening points



delete(deleteLowerEntities = "False", deleteHigherEntities = "False", deleteElementTypes = "False", points = "all")

Fig. 12-52 Script – Parametric nurbs curves.

REFERENCES

- [1] Cervenka, V., Jendele, L, Cervenka, J., (2022), ATENA Program Documentation, Part 1, Theory, Cervenka Consulting, 2, 2022
- [2] Cervenka, J., Altman, T., Janda, Z., Palek, P., Pukl, R., ATENA 2023 Program Documentation Part 3-4, Example & Validation Manual, ATENA with CeSTaR 2 Module, Cervenka Consulting s.r.o. 2023.
- [3] Pryl, D. and Cervenka, J., (2022), ATENA Program Documentation Part 11, ATENA Troubleshooting, Cervenka Consulting, 2022
- [4] Cervenka, J., and Jendele, L., (2022), *ATENA Program Documentation, Part 6, ATENA Input File Format*, Cervenka Consulting, 2022
- [5] Rypl, D. (2016), *Triangulation of 3D domains User guide*, CTU in Prague, 2016