# NEUTRAL FILE FORMAT

Version 19 June 1994

## 1) Objectives

The neutral file format was established with the following goals:

a) The file must contain all informations needed for the Finite Element pre, post and analysis programs of our group (PUC-Rio, CENPES/PETROBRAS, Alberta, etc...). That probably means that a given program may find information in the file that is not needed for its function. A simple way to skip this information must be provided.

b) The file must be easily readable from FORTRAN (or C) programs. That is, no special library should be provided to a program to interpreter the file. The maximum complexity allowable is the one that can be explained by a simple example of how a FORTRAN 77 program could read the file. As a corollary, the file must be easily produced from a FORTRAN or C program.

c) The file structure should be simple to understand. That is, no automatic generation or implicit rule should be associated with it. Moreover, the layout of the file must be so simple that a complete example should substitute its manual without no extra explanations. That is, a person familiar with Finite Element Analysis should be able to understand the file structure simply by looking it.

Obviously all three objectives are impossible to achieve simultaneously. So a compromise must be established.

## 2) General Guide Lines

a) The file is sequential, clear text format.

b) The file is line oriented but no colunm alignment is required.

c) The file is composed of SECTIONS, each one starting with a header (%LABEL). A section that is meaningless for a given program can be skipped by reading until the next section label.

d) The variable type convention adoted is:
[v] integer variable
<v> real variable
'v (n)' string of no more than n characters.

# 3) File Format

## 3.1) General File Information

**%REMARK**
The label REMARK can appear in any place (between sections) and its contents are ignored by the applications. It is used here to document the file.

**%HEADER**
This section is the first one in the file and should contain: (1) who created the file; (2) the creation date and any other interesting but not essential information.

**%HEADER.AUTHOR**
Define the author's name or the program's name that build the file.

**%HEADER.DATE**
Creation date of the file.

**%HEADER.VERSION**
<neutral_file_format_version_number>, eg. 1.01

**%HEADER.TITLE**
'File title(80)'

**%HEADER.ANALYSIS**
'analysis_type(12)', eg. 'plane_stress'

**%HEADER.ANALYSIS.MAXIMUN.STEPS**
<max_step>

**%HEADER.ANALYSIS.TIME.STEPS**
<time_steps>

**%HEADER.ANALYSIS.TOLERANCE.STEPS**
<tolerance>

**%HEADER.ANALYSIS.MAXIMUN.ITERATIONS**
<max_interation>

**%HEADER.ANALYSIS.PRINT.STEPS**
<print_steps>

**%HEADER.ANALYSIS.ALGORITHM**
'label_algorithm'

**%HEADER.ANALYSIS.ERROS**
<error>

**%HEADER.ANALYSIS.NLOOP**
<nloops>

**%HEADER.UNITS**
LENGTH FORCE ANGLE TEMPERATUTE TIME [5 x constant string]

Obs: These constant strings may be:
LENGTH: INCH, FEET, CM, M, MM
FORCE: LB, KP, KN, KG, N, MTONS
ANGLE: RAD, DEG, CYC
TEMPERATURE: DEGF, DEGC
TIME: SEC, MIN, HR

**%END**
This label can appear at any place of the file. It indicates the end of the file. All subsequent sections are ignored.

## 3.2) Nodal Information

Nodes are defined by their id and may have labels associated with them. To give node coordinates, node id must be consecutive integer. And, at support section, free nodes may be omited.

**%NODE**
[#_of_nodes]

**%NODE.COORD**
[#_of_nodes_coordinates]
[node_id] $< x >$ $< y >$ $< z >$
...

**%NODE.LABEL**
[#_of_node_labels]
[node_id] 'label (12)'
...

**%NODE.SUPPORT**
[#_of_constrained_node]
[node_id] [$d$_$x$] [$d_y$] [$d_z$] [$r_x$] [$r_y$] [$r_z$] (0=free, 1=fixed)
...

**%NODE.INCLINED.SUPPORT**
[#_of_inclined_supports]
[node_id] $< \Phi_{xx} >$ $< \Phi_{xy} >$ $< \Phi_{xz} >$ $< \Phi_{yx} >$ $< \Phi_{yy} >$ $< \Phi_{yz} >$
...

**%NODE.RIGID.LINK**
[#_of_node_rigid_links]
[slave_node_id] [master_node_id]
...

**%NODE.CONSTRAINT**
[#_of_constraint_nodes]
[constrained_node] [free_node] [$d_x$] [$d_y$] [$d_z$] [$r_x$] [$r_y$] [$r_z$]
...
Obs. Constrained_node have the same d.o.f. of the free node if code is 1. They have different d.o.f. if the code is 0.

**%NODE.FLOOR**
[#_of_nodes_connected_to_floors]
[node_id] [node_id_of_the_floor_node]
...

**%NODE.MASS**
[#_of_nodal_masses]
[node_id] $< m_x >$ $< m_y >$ $< m_z >$ $< i_x >$ $< i_y >$ $< i_z >$
...

**%NODAL.SPRING**
[#_of_nodal_springs]
[node_id] $< k_x >$ $< k_y >$ $< k_z >$ $< m_x >$ $< m_y >$ $< m_z >$
...

## 3.3) Material Information

Materials are defined by their id and may have labels associated with them. Basic material properties are given as a function of their type: ISOTROPIC, ORTHOTROPIC, TRESCA, ... .

Additional properties, such as density and thermal expansion factor, can be added to a given material using separeted sections.

**%MATERIAL**
[#_of_materials]

**%MATERIAL.LABEL**
[#_of_materials_labels]
[material_id] 'material_label (12)'
...

**%MATERIAL.ISOTROPIC**
[#_of_isotropic_materials]
[material_id] $< E > \; < \nu >$
...

**%MATERIAL.ORTHOTROPIC**
[#_of_orthotropic_materials]
[material_id] $< E_x > \; < E_y > \; < E_z > \; < \nu_x > \; < \nu_y > \; < \nu_z >$
...

**%MATERIAL.TRESCA**
[#_of_Tresca_materials]
[material_id] $< E > \; < \nu > \; < \tau_{max} >$
...

**%MATERIAL.MISES**
[#_of_Mises_materials]
[material_id] $< E > \; < \nu > \; < \tau_{max} >$
...

**%MATERIAL.MOHR.COULOMB**
[#_of_Mohr_Coulomb_materials]
[material_id] $< E > \; < \nu > \; < C > \; < \Phi >$
...

**%MATERIAL.BRESLER.PISTER**
[#_of_Breler_Pister_materials]
[material_id] $< E > \; < \nu > \; < K > \; < \alpha >$
...

**%MATERIAL.HSIEH.TING.CHENG**
[#_of_Hsieh.Ting_Cheng_materials]
[material_id] $< E > \; < \nu > \; < a > \; < b > \; < c > \; < d >$
...

**%MATERIAL.GENERIC**
[#_of_Generic_materials]
[material_id] $< \#\_of\_parameters >< parameter\_1 >< parameter\_2 > \ldots < parameter\_n >$
...

**%MATERIAL.INTERFACE**
[#_of_Interface_materials]
[material_id] $< K_n > \; < K_t > \; < C > \; < \Phi >$
...

**%MATERIAL.PROPERTY.DENSITY**
[#_of_material_density]
[material_id] <specific_weight>
...

**%MATERIAL.PROPERTY.THERMAL**
[#_of_Thermal_materials]
[material_id] $< conductivity > \; < capacity > \; < porosity > \; < decay >$
...

**%MATERIAL.PROPERTY.THERMAL.EXPANSION**
[#_of_material_thermal_expansion_factors]
[material_id] <thermal_expansion_factor>
...

**%MATERIAL.PROPERTY.PLASTIC**
[#_of_plastic_properties_sets]
[material_id] $< F_y > \; < F_u >$

## 3.4) Bar Section Information

This section is used only by programs dealing with frames(trusses, beams, columns and beam-columns). Bar cross sections are defined by treir id and may have an associate label. Cross section properties can be given directly (SECTION.PROPERTY, SECTION.PROPERTY.PLASTIC) or through the geometric definition of its cross section (TUBULAR, CIRCULAR, BOX,...). Analysis programs will prefer the former and design programs the latter.

Three extras titles are included in this section to deal with the details of bar elements: (1) orientation vector; (2) beta angle; and (3) end releases. They are used to define the bar orientation in space and possible joint releases.

Programmers should be aware of the redundacy to define cross section properties and minor axis orientation. Neutral file authors suggests that they should use the section properties and the orientation vectors in case they are given. That is, if a section is a box but have the properties explicity defined, an analysis program should use the latter. The same apply for orientation vectors an beta angles. Programs should also be prepared to convert this informations.

**%SECTION**
[#_of_bars(trusses, beams and colunms) cross sections]

**%SECTION.LABEL**
[#_of_sections_labels]
[section_id] 'section_label (12)'
...

**%SECTION.PROPERTY**
[#_of_beam_column_sections]
[section_id] $< A_x >$ $< A_y >$ $< A_z >$ $< I_x >$ $< I_y >$ $< I_z >$ $< C_w >$ $< H_y >$ $< H_z >$
...

**%SECTION.PROPERTIES.PLASTIC**
[#_of_plastic_section_properties]
[section_id] $< Z_y >$ $< Z_z >$
...

**%SECTION.TUBULAR**
[#_of_tubular_beam_column_sections]
[section_id] <diameter> <thickness>
...

**%SECTION.CIRCULAR**
[#_of_circular_beam_column_sections]
[section_id] <diameter>
...

**%SECTION.ELLIPTIC**
[#_of_elliptic_beam_column_sections]
[section_id] $< a >$ $< b >$
...

**%SECTION.BOX**
[#_of_box_beam_column_sections]
[section_id] $< b >$ $< h >$
...

**%SECTION.HOLLOW.BOX**
[#_of_box_beam_column_sections]
[section_id] $< b > \; < h > \; < tb > \; < th >$
...

**%SECTION.I**
[#_of_I_beam_column_sections]
[section_id] $< b > \; < h > \; < tb > \; < tw >$
...

**%SECTION.U**
[#_of_U_beam_column_sections]
[section_id] $< b > \; < d > \; < tb > \; < td >$
...

**%SECTION.C**
[#_of_C_beam_column_sections]
[section_id] $< b > \; < h > \; < tb > \; < th >$
...

**%SECTION.L**
[#_of_L_beam_column_sections]
[section_id] $< b > \; < h > \; < tb > \; < th >$
...

**%SECTION.T**
[#_of_T_beam_column_sections]
[section_id] $< b > \; < h > \; < tb > \; < th >$
...

**%SECTION.VARIABLE**
[#_of_variable_beam_column_sections]
[section_id] [number_of_segments]
<segment_lenght> [ref_section_id]
...
Obs. This section refers other beam sections. It defines is as each segment was a uniform beam.

**%BEAM.ORIENTATION.VECTOR**
[#_of_beam_column_orientation_vectors]
[orientation_id] $< V_x > \; < V_y > \; < V_z >$
...

**%BEAM.ORIENTATION.BETHA**
[#_of_beam_column_orientation_betha]
[orientation_id] <betha>
...

**%BEAM.END.LIBERATION**
[#_of_beam_column_end_liberation_codes]
[lib_id] $[d_{xi}]$ $[d_{yi}]$ $[d_{zi}]$ $[r_{xi}]$ $[r_{yi}]$ $[r_{zi}]$ $[d_{xj}]$ $[d_{yj}]$ $[d_{zj}]$ $[r_{xj}]$ $[r_{yj}]$ $[r_{zj}]$
...
Obs - Liberation code 0 means free and 1 means no liberation.

## 3.5) Elements Attributes Information

Elements thickness and element integration order tables are given in these sections.

**%THICKNESS**
[#_of_element_thickness]
[section_id] <thickness>
...

**%INTEGRATION.ORDER**
[#_of_element_integration_orders]
[section_id] [stiffness_r] [stiffness_s] [stiffness_t] [stress_r] [stress_s] [stress_t]
...

## 3.6) Element Information

Elements are defined by their id and may have labels associated with them.

**%ELEMENT**
[#_of_elements]

**%ELEMENT.LABEL**
[#_of_elements_labels]
[element_id] 'element_label (12)'
...

**%ELEMENT.TRUSS**
[#_of_truss_elements]
[element_id] [material_id] [section_id] [node_i_id] [node_j_id]
...

**%ELEMENT.BEAM**
[#_of_beam.column_elements]
[element_id] [material_id] [section_id] [orientation_id] [lib_id] [node_i_id] [node_j_id]
...

**%ELEMENT.T3**
[#_of_t3_elements]
[element_id] [material_id] [thickness_id] [integration_id] [node_i_id] [node_j_id] [node_k_id]
...

**%ELEMENT.T6**
[#_of_t6_elements]
[element_id] [material_id] [thickness_id] [integration_id] [node_i_id] [node_j_id] ... [node_n_id]
...

**%ELEMENT.Q4**
[#_of_q4_elements]
[element_id] [material_id] [thickness_id] [integration_id] [node_i_id] [node_j_id] ... [node_l_id]
...

**%ELEMENT.Q8**
[#_of_q8_elements]
[element_id] [material_id] [thickness_id] [integration_id] [node_i_id] [node_j_id] ... [node_p_id]
...

**%ELEMENT.DKT**
[#_of_dkt_elements]
[element_id] [material_id] [thickness_id] [integration_id] [node_i_id] [node_j_id] [node_k_id]
...

**%ELEMENT.DKQ**
[#_of_dkq_elements]
[element_id] [material_id] [thickness_id] [integration_id] [node_i_id] [node_j_id] ... [node_l_id]
...

**%ELEMENT.QUAD**
[#_of_quad_elements]
[element_id] [material_id] [thickness_id] [integration_id] [node_i_id] [node_j_id] ... [node_p_id]
...

**%ELEMENT.INTERFACE**
[#_of_interface_elements]
[element_id] [material_id] [thickness_id] [integration_id] [node_i_id] [node_j_id] ... [node_n_id]
...

**%ELEMENT.INFINITE**
[#_of_infinite_elements]
[element_id] [material_id] [thickness_id] [integration_id] [node_i_id] [node_j_id] ...[node_m_id]
...

**%ELEMENT.BRICK8**
[#_of_brick8_elements]
[element_id] [material_id] [integration_id] [node_1_id] [node_2_id] [node_3_id] ... [node_8_id]
...

**%ELEMENT.BRICK20**
[#_of_brick20_elements]
[element_id] [material_id] [integration_id] [node_1_id] [node_2_id] [node_3_id] ... [node_20_id]
...

## 3.7) Load Information

Load information is organized in load cases. Each load case information follow a load case section. And, if no load combination is given, programs should consider each load case as a load combination.

**%LOAD**
[#_of_load_cases]
[load_case_id] 'load_case_label (12)'
...

**%LOAD.CASE**
[current_load_case_id]
Obs. This section only sets the current load case to be used in the load definition that follows.

**%LOAD.CASE.NODAL.FORCE**
[#_of_nodal_forces]
[node_id] $< F_x >$ $< F_y >$ $< F_z >$ $< M_x >$ $< M_y >$ $< M_z >$
...

**%LOAD.CASE.NODAL.DISPLACEMENT**
[#_of_nodal_displacements]
[node_id] $< d_x >$ $< d_y >$ $< d_z >$ $< r_x >$ $< r_y >$ $< r_z >$
...

**%LOAD.CASE.NODAL.PRESCRIBED.TEMPERATURE**
[#_of_nodal_prescribed_temperatures]
[node_id] <temperature>
...

**%LOAD.CASE.NODAL.INITIAL.TEMPERATURE**
[#_of_nodal_initial_temperatures]
[node_id] <temperature>
...

**%LOAD.CASE.LINE.FORCE.UNIFORM**
[#_of_line_forces]
[element_id] [initial_node_id] [final_node_id] [local/global_key] (1=local/0=global)
$< q_x >$ $< q_y >$ $< q_z >$
....

**%LOAD.CASE.LINE.FORCE.VARIABLE**
[#_of_line_forces]
[element_id] [initial_node_id] [final_node_id] [local/global_key] (1=local/0=global)
$< q_{x1} >$ $< q_{y1} >$ $< q_{z1} >$
$< q_{x2} >$ $< q_{y2} >$ $< q_{z2} >$
... (#_of_incidence_nodes)
...

**%LOAD.CASE.LINE.MOMENT.UNIFORM**
[#_of_line_moments]
[element_id] [initial_node_id] [final_node_id] [local/global_key] (1=local/0=global)
$< m_x >$ $< m_y >$ $< m_z >$
...

### %LOAD.CASE.LINE.MOMENT.VARIABLE

[#_of_line_moments]
[element_id] [initial_node_id] [final_node_id] [local/global_key] (1=local/0=global)
$< m_{x1} > \; < m_{y1} > \; < m_{z1} >$
$< m_{x2} > \; < m_{y2} > \; < m_{z2} >$
... (#_of_incidence_nodes)
...

### %LOAD.CASE.DOMAIN.FORCE.UNIFORM

[#_of_domain_forces]
[element_id] [node_i_id] [node_j_id] [node_k_id] [local/global_key] (1=local/0=global)
$< p_x > \; < p_y > \; < p_z >$
...

### %LOAD.CASE.DOMAIN.INITIAL.STRESS.UNIFORM

[#_of_dimain_initial_stress]
[element_id]
$< \sigma_{xx} > \; < \sigma_{yy} > \; < \sigma_{zz} > \; < \sigma_{xy} > \; < \sigma_{xz} > \; < \sigma_{yz} >$
...

### %LOAD.CASE.AREA.PRESSURE.UNIFORM

[#_of_area_pressures]
[element_id] [node_i_id] [node_j_id] [node_k_id] [local/global_key] (1=local/0=global)
$< p_x > \; < p_y > \; < p_z >$
...

### %LOAD.CASE.AREA.PRESSURE.VARIABLE

[#_of_area_pressures]
[element_id] [node_i_id] [node_j_id] [node_k_id] [local/global_key] (1=local/0=global)
$< p_{x1} > \; < p_{y1} > \; < p_{z1} >$
$< p_{x2} > \; < p_{y2} > \; < p_{z2} >$
... (#_of_incidence_nodes)
...

### %LOAD.CASE.BEAM.FORCE

[#_of_beam_forces]
[element_id] [local/global_key] (1=local/0=global) <point_distance>
$< F_x > \; < F_y > \; < F_z > \; < M_x > \; < M_y > \; < M_z >$
...

### %LOAD.CASE.BEAM.UNIFORM

[#_of_beam_uniform_loads]
[element_id] [local/global_key] (1=local/0=global) <start_dist> <end_dist>
$< q_x > \; < q_y > \; < q_z > \; < m_x > \; < m_y > \; < m_z >$
...

### %LOAD.CASE.BEAM.LINEAR

[#_of_beam_linear_loads]
[element_id] [local/global_key] (1=local/0=global) <start_dist> <end_dist>
$< q_{xi} > \; < q_{yi} > \; < q_{zi} > \; < m_{xi} > \; < m_{yi} > \; < m_{zi} >$
$< q_{xf} > \; < q_{yf} > \; < q_{zf} > \; < m_{xf} > \; < m_{yf} > \; < m_{zf} >$
...

### %LOAD.CASE.LINE.CONVECTION.UNIFORM

[#_of_line_convection]
[element_id] [initial_node_id] [final_node_id]
$< q >$
...

**%LOAD.CASE.LINE.HEAT.FLUX.UNIFORM**
[#_of_line_forces]
[element_id] [initial_node_id] [final_node_id]
$< q >$
...

**%LOAD.CASE.LINE.HEAT.FLUX.VARIABLE**
[#_of_line_forces]
[element_id] [initial_node_id] [final_node_id]
$< q_1 >$
$< q_2 >$
... (#_of_incidence_nodes)
...

**%LOAD.CASE.AREA.TEMPERATURE**
[#_of_area_temperature]
[element_id] <temp>
...

**%LOAD.CASE.DOMAIN.TEMPERATURE.UNIFORM**
[#_of_domain_temperature]
[element_id] <temp>
...

**%LOAD.CASE.AREA.HEAT.FLUX.UNIFORM**
[#_of_area_pressures]
[element_id] [node_i_id] [node_j_id] [node_k_id]
$< q >$
...

**%LOAD.CASE.AREA.HEAT.FLUX.VARIABLE**
[#_of_area_pressures]
[element_id] [node_i_id] [node_j_id] [node_k_id]
$< q_1 >$
$< q_2 >$
... (#_of_incidence_nodes)
...

**%LOAD.CASE.GRAVITY**
$< g_x > \; < g_y > \; < g_z >$
**%LOAD.COMBINATION**
[#_of_load_combination]
[combination_id] 'label (12)' <factor_1> ... <factor_#_of_load_cases>
...

## 3.8) Rezone Information

Rezone information is organized in rezone cases. Each rezone case information follow a rezone case section.

**%REZONE**
[#_of_rezone_cases]
[rezone_case_id]
...

**%REZONE.CASE**
[current_rezone_case_id]
Obs. This section only sets the current rezone case to be used in the rezone definition that follows.

**%REZONE.CASE.ELEMENT**
[#_of_elements_rezone]
[element_id]
...

## 3.9) Function Information

Function information is applied to nodes, lines and elements.

**%FUNCTION**
[#_of_functions]

**%FUNCTION.LABEL**
[#_of_function_label]
[function_id]'function_label(12)'

**%FUNCTION.LABEL**
[#_of_functions]
[function_id][#_of_function_points][genericpredefined-key] (0=generic $\neq 0$ predefined)
$< f_x > \; < f_y >$
...
obj: only it will have points if key = 0 (generic function)

**%FUNCTION.NODAL**
[#_of_nodal_functions]
[node_id][function_id]
...

**%FUNCTION.LINE**
[#_of_line_functions]
[element_id][initial_node_id][final_node_id][function_id]
...

**%FUNCTION.ELEMEMT**
[#_of_element_function]
[element_id][function_id]
...

## 4.0) Result Information

Result information is organized in result case. Each result case is organized in result case step. Each result case step may be associated with a time step or a load factor.

**%RESULT**
[#_of_result_cases]
[result_case_id] 'result_case_title (12)'
...

**%RESULT.CASE**
[current_result_case_id] [#_of_steps]
[step_id] 'label (12)'
...

**%RESULT.CASE.STEP**
[current_case_step_id]

**%RESULT.CASE.STEP.TIME**
<time_step>

**%RESULT.CASE.STEP.FACTOR**
<load_factor>

**%RESULT.CASE.STEP.NODAL.DISPLACEMENT**
[#_of_nodes] 'nodal_displacement_label (12)'
[node_id] $< d_x >$ $< d_y >$ $< d_z >$ $< r_x >$ $< r_y >$ $< r_z >$
...

**%RESULT.CASE.STEP.NODAL.SCALAR**
[#_of_scalar_fields]
'field_label_1 (12)' 'field_label_2 (12)' ...'field_label_#_of_scalar_fields (12)'

**%RESULT.CASE.STEP.NODAL.SCALAR.DATA**
[#_of_nodes]
[node_id] <field_1_dat> <field_2_dat> ... <field_#_of_scalar_fields_dat>
...

**%RESULT.CASE.STEP.ELEMENT.NODAL.SCALAR**
[#_of_scalar_fieds]
'field_label_1 (12)' 'field_label_2 (12)' ...'field_label_#_of_scalar_fields (12)'

**%RESULT.CASE.STEP.ELEMENT.NODAL.SCALAR.DATA**
[#_of_elements]
[element_id]
<field_1_node_1> <field_2_node_1> ... <field_#_of_scalar_fields_node_1]
<field_1_node_2> <field_2_node_1> ... <field_#_of_scalar_fields_node_1]
... (#_of_incidence_nodes)
...

**%RESULT.CASE.STEP.ELEMENT.GAUSS.SCALAR**
[#_of_scalar_fieds]
'field_label_1 (12)' 'field_label_2 (12)' ...'field_label_#_of_scalar_fields (12)'

**%RESULT.CASE.STEP.ELEMENT.GAUSS.SCALAR.DATA**
[#_of_elements]
[element_id] [#_of_gauss_points]
<field_1_gauss_1> <field_2_gauss_1> ... <field_#_of_scalar_fields_gauss_1>
<field_1_gauss_2> <field_2_gauss_2> ... <field_#_of_scalar_fields_gauss_2>
... (#_of_gauss_points)

**%RESULT.CASE.STEP.NODAL.VECTOR**
[#_of_vector_fields]
'field_label_1 (12)' 'field_label_2 (12)' ...'field_label_#_of_vector_fields (12)'

**%RESULT.CASE.STEP.NODAL.VECTOR.DATA**
[#_of_nodes]
[node_id]
<field_1_vx> <field_1_vy> <field_1_vz>
<field_2_vx> <field_2_vy> <field_2_vz>
...
<filed_#_vx> <field_#_vy> <field_#_vz>
... (#_of_gauss_points)

**%RESULT.CASE.STEP.ELEMENT.GAUSS.VECTOR**
[#_of_vector_fieds]
'field_label_1 (12)' 'field_label_2 (12)' ...'field_label_#_of_vector_fields (12)'

**%RESULT.CASE.STEP.ELEMENT.GAUSS.VECTOR.DATA**
[#_of_elements [integer>
[element_id] [#_of_gauss_points]
<field_1_gauss_1_vx> <field_1_gauss_1_vy> <field_1_gauss_1_vz>
<field_2_gauss_1_vx> <field_2_gauss_1_vy> <field_2_gauss_1_vz>
...
<field_#_gauss_1_vx> <field_#_gauss_1_vy> <field_#_gauss_1_vz>
<field_#_gauss_2_vy> <field_#_gauss_2_vy> <field_#_gauss_2_vz>
... (#_of_gauss_points)