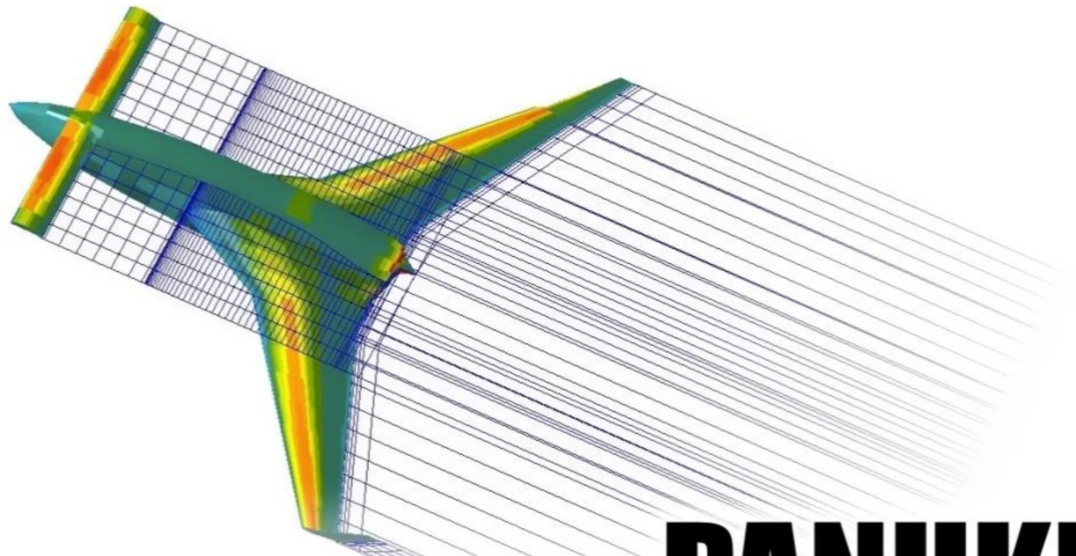
Warsaw, 01-04-2010



Users manual for PANUKL

Version **ENG** V1

Index

1. Foreword

PANUKL 2002 can be used for aerodynamic computation of an aircraft, using low cost panel methods. This is the continuation of set of programs (**PAN**eli **UKL**ad **96**) being developed in the middle 90ties of past century. The most important changes were made in 2001-2002. Program is still being developed.

Below one can find theoretical basis and description of main program functionalities. User manual will guide through program installation and usage.

1.1.1. Index of variables

a_∞ - sound speed of free stream flow

b – wing span

Cm – pitching moment coefficient with respect to ¼ of MAC

CD – drag coefficient

CL – lift coefficient

p – total pressure

Q – pitch rate

S – reference area

V_∞ - free stream velocity

x, y, z – Cartesian coordinates for geometry definition, usually defined as follows:
origin in fuselage nose or root wing section
x axis along root chord $c_{R,}$
z axis perpendicularly to root chord directed "up"
y axis perpendicularly to right wing.

α - angle of attack

Φ - full velocity potential

Φ_∞ - velocity potential in infinity

Φ_i - velocity potential inside the body

φ - potential of velocity disturbances

κ - isentropic exponent

Λ - geometric aspect ratio (b^2/S)

μ - doublet strength

ρ - air density

σ - source strength

1.1.2. Introduction

The develop of CFD methods and big increase of the computers power caused, that *Euler* or *Navier-Stockes* models are used more often and potential methods could be seen as obsolete. However potential methods, despite many simplification are still the attractive tool [1,2,3]. Low cost and fact, that they are relatively easy to apply compensate their disadvantages and lower accuracy.

1.1.3. Physical and mathematical model

The most important assumptions made for physical model definition are that fluid is inviscid and irrotational (except vortex wake). The viscidity effect is simulated by *Kutta-Joukowski* boundary condition, what could be interpreted that circulation on the trailing edge is equal to zero.

The mathematical model is as follows:

- continuity equation:

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho V) = 0 \qquad (1)$$

- *Eulera* equation:

$$\frac{\partial V}{\partial t} + (V\ \mathrm{grad})\ V = \frac{1}{\rho}\mathrm{grad}\ p \qquad (2)$$

- state equation:

$$p = p_\infty \left(\frac{\rho}{\rho_\infty}\right)^\kappa \qquad (3)$$

Because fluid is irrotational (rot **V** = 0) the scalar function, called velocity potential can be defined and the following condition is satisfied:

$$\mathrm{grad}\ \Phi(x, y, z, t) = V \qquad (4)$$

If we assume, that $\mathbf{\Phi = \Phi_\infty + \varphi}$ and: $\mathbf{mod\nabla\varphi << U_\infty}$, $\mathbf{mod\nabla\varphi << a_\infty}$ and $\mathbf{mod\nabla\varphi << (U_\infty\text{-}a_\infty)}$ then we obtain:

$$\frac{1}{a_\infty}\left(\frac{\partial}{\partial t} + V_\infty \frac{\partial}{\partial x}\right)^2 \varphi = \nabla\varphi \qquad (5)$$

assuming additionally, that flow is steady and incompressible, we have:

$$\nabla\varphi = 0 \qquad (6)$$

1.1.4. Computational method

Computational method strongly depends on the way of aircraft body modeling. The model defined in chapter 1.1.3 concerns only flow and doesn't define the object. Generally two methods are in use. In the first method the body of aircraft is modeled using thin surfaces. The second method uses three dimensional model of the aircraft body. *PANUKL 2002* package bases on the low order panel method, where the *Dirichlet* problem is solved (*Hess* method [7]). The quadrangle panels are used. The flat vortex wake, parallel to the free stream velocity or parallel to chord is assumed.

The base of the method is solution of the *Laplace* equation for the full velocity potential.

$$\nabla^2\Phi = 0; \tag{7}$$

The velocity potential can be written in form [6]:

$$\Phi(x,y,z) = \frac{1}{4\pi}\cdot\int\limits_{BODY+WAKE}\mu\frac{\partial}{\partial n}\cdot\left(\frac{1}{r}\right)dS - \frac{1}{4\pi}\cdot\int\limits_{BODY}\sigma\left(\frac{1}{r}\right)dS + \Phi_\infty \tag{8}$$

Assuming the following boundary conditions:

- internal *Dirichlet* boundary condition on the body surface:

$$\frac{1}{4\pi}\cdot\int\limits_{BODY+WAKE}\mu\frac{\partial}{\partial n}\left(\frac{1}{r}\right)dS - \frac{1}{4\pi}\cdot\int\limits_{BODY}\sigma\frac{1}{r}dS = 0 \tag{9}$$

where:

doublet strength: $\mu = -(\Phi - \Phi_i)$, (10)

source strength: $\sigma = \partial\mu/\partial \boldsymbol{n}$. (11)

- *Kutta-Joukowsky* conditions on the trailing edge:

$$\Delta p(x,y)_{TE} = 0 \tag{12}$$

- on the vortex wake:

$$\frac{\partial\varphi(x,y)}{\partial x} = 0; \tag{13}$$

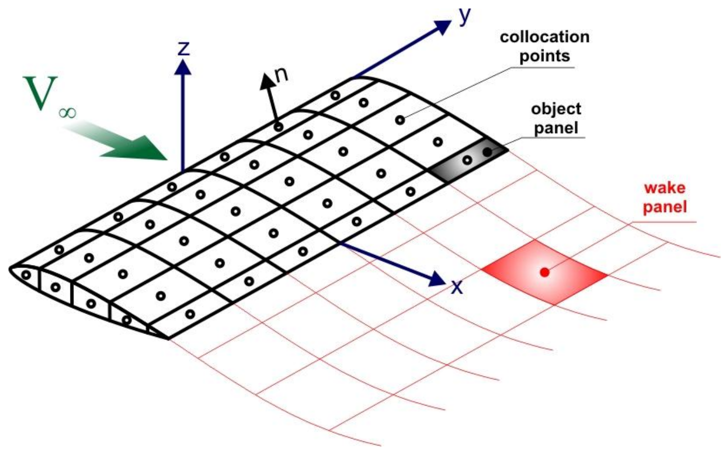


Fig. 1 – Approximation of the body surface by panel elements

and assuming, that the velocity potential inside the body Φ_i is equal to velocity potential in infinity Φ_∞ the integral equation is derived in form (9). The approximation of the aircraft body surface by flat panels allows to approximate the equation (9) by system of linear algebraic equations with unknown doublet strength (constant for panel):

$$\sum_{k=1}^{N} C_k \mu_k + \sum_{l=1}^{N_w} C_l \mu_l + \sum_{k=1}^{N} B_k \sigma_k = 0 \quad (14)$$

where C_k , C_l and B_k denote influence coefficients:

$$C_k = \frac{1}{4\pi} \cdot \int_{S_{1234}} \frac{\partial}{\partial n}\left(\frac{1}{r_k}\right) dS_k \, ; \quad B_k = -\frac{1}{4\pi} \cdot \int_{S_{1234}} \frac{1}{r_k} dS_k \quad (15)$$

N – numbers of panels on the aircraft surface;
N_W – number of panels on the wake;
S_{1234} – area of the k-th panel;

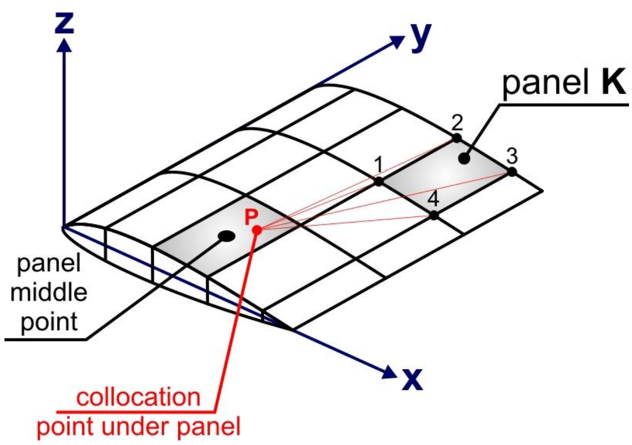


Fig. 2 – Influence of K-th panel on point P

The source strength σ (constant for panel), can be defined (using definitions (10) , (11) and boundary condition of the closed body $\partial\Phi_i/\partial\boldsymbol{n}=0$) as follows:

$$\sigma = -\boldsymbol{n} \cdot \mathrm{V}_{\infty} \qquad (16)$$

It will result in a set of equation with the doublet strength as the unknown. To determine the doublet strength on the vortex wake, the *Kutta-Joukowsky* condition is used:

$$\mu_{TE} = \mu_W = \mathbf{const} \qquad (17)$$

The doublet strength on the wake is equal to difference between doublet strength on the upper and lower surface close to the trailing edge. Using (17), the doublet strength on the wake can be obtained from formula:

$$\mu_W = \mu_U - \mu_L \qquad (18)$$

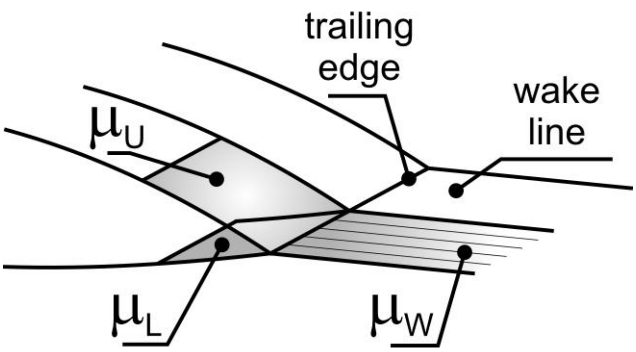


Fig. 3 – Relation between doublet strength on trailing edge and wake

The formula (18) completes the set of equations (14). Only integrals (15) have to be determined. Effective method of determination of these integrals is shown in [6] and [7].

The solution of set (14) gives the potential distribution on the body surface. To obtain the pressure distribution, necessary to obtain the global aerodynamic coefficients, the velocity distribution must be found. It can be made by differentiation of the potential with respect to defined tangential coordinates. Next using Bernoulli's theorem the pressure can be computed. The numerical differentiation in general case is not easy and can be the source of errors, especially in places, where the grid is not regular.

The aerodynamic loads can be obtained as follows:

- lift force

$$P_z = -\sum_{i=1}^{N} p_i S_i \boldsymbol{n}_i \cdot \boldsymbol{z} \tag{19}$$

- drag force

$$P_x = -\sum_{i=1}^{N} p_i S_i \boldsymbol{n}_i \cdot \boldsymbol{x} \tag{20}$$

- pitching moment

$$M_y = \sum_{i=1}^{N} p_i S_i x_i \boldsymbol{n}_i \cdot \boldsymbol{z} + \sum_{i=1}^{N} p_i S_i z_i \boldsymbol{n}_i \cdot \boldsymbol{x} \tag{21}$$

The lateral load components (P_y, M_x, M_z) can be computed in similar way if we define lateral components of airspeed. It must be underlined, that drag force obtained from (20) can be very inaccurate. The potential methods cannot give reliable results of aerodynamic drag. Package *PANUKL 2002* computes the induced drag coefficient by use of *Trefz* method.

1.2. Brief PANUKL software description – main subprograms and their functionalities

PANUKL 2002 application is composed of three main subprogram groups. In first group we can find data preparation programs. In second group we can find programs to process and compute data. The last group is the managing program where we can watch obtained results and make appropriate changes and modifications.

PANUKL works in Windows and Linux:

- MS Windows (2000/XP/Vista),
- Linux (extra information can be found in installation package).

Both program versions need **OpenGL** libraries.

1.2.1. Managing subprogram – GRIDVIEW

GRIDVIEW

All of **PANUKL 2002** subprograms can be executed from **GRIDVIEW** – **PANUKL's** managing application (for detailed description go to chapter 3). From **CREATE** menu (one of the managing application menus) we can access to subprograms. We can also run subprograms from command line. Each subprogram needs a configuration file (see below). The correct order is necessary during computation process.

1. **MESH** – grid generator (run: **Mesh.exe** *name*.**ms2**),
2. **NEIGH** – vortex wake and neighbor generator (**Neigh.exe** *name*.**ngh**),
3. **PANUKL** – velocity potential distribution solver (**Panukl.exe** *name*.**par**),
4. **PRESS** – pressure distribution and global aerodynamic results solver (**Press.exe** *name*.**prs**).

1.2.2. Data preparation subprogram – MESH

MESH

MESH subprogram is being used to create grid (made from quadrangle panels) describing aircraft body. To create grid file **[***name***.INP]** user must prepare correct input files:

- main aircraft geometry description file **[***name***.MS2]** (contains aircraft reference data, information about wing, tail , fuselage overall geometry),
- wing airfoil geometry file **[***name***.PRF]**,
- fuselage geometry file **[***name***.F]**.

For more information go to chapter 1.3.1.

1.2.3. Main computational subprograms – NEIGH, PANUKL and PRESS

NEIGH

NEIGH subprogram is being used to calculate neighboring panel numbers. Additionally it extends grid with wake grid panels. The input file for **NEIGH** is **[***name***.INP]** - grid geometry file. Configuration file is **[***name***.NGH]**. The output file with grid and wake panels is **[***name***.DAT]**.

PANUKL

PANUKL subprogram computes influence factors, missing geometrical data and solves system of equations. As a result we get velocity potential distribution. **PANUKL** subprogram input parameters like angle of attack or angular velocities are read from **[***name***.PAR]** configuration file.

The results are saved to **[***name***.PAN]** output file which is an input for **PRESS** subprogram. Creating **[***name***.PAN]** output file can last long and it is the most CPU consuming process.

PRESS

PRESS subprogram computes pressure distribution over the aircraft body by differentiating the velocity potential distribution. Additionally we can obtain global aerodynamic coefficient values, downwash distribution and inducted drag in *Trefz* plane. **PRESS** subprogram input parameters are read from **[***name***.PRS]** configuration file. The results are saved to three output files (for more information go to chapter 1.3.2).

- **[***name***.OUT]** - global aerodynamic results,
- **[***name***.CZY]** - aerodynamic coefficient distribution over the wing,
- **[***name***.TXT]** - the results for pressure coefficient, velocity, source or doublet distribution etc. (for each panel of aircraft body),
- **[***name***.EPS]** - downwash results (created as an option),
- **[***name***.BLN]** - object geometry outline for current downwash computational plane (created as an option).

1.3. Input data

1.3.1. Input data file description

File [*name*.**prf**] – wing airfoil geometry – file description

- comment line (not necessary)

WING AIRFOIL FILE EXAMPLE

24 #n – number of points defining curvature lines for current airfoil (both top and bottom curvature line), Fig. 4

# top curvature line definition		#bottom curvature line definition	
# **X** coordinates	# **Y** coordinates	# **X** coordinates	# **Y** coordinates
0.000	**0.000**	**0.000**	**0.000**
0.006	**0.093**	**0.006**	**-0.093**
0.622	**0.905**	**0.622**	**-0.905**
2.233	**1.655**	**2.233**	**-1.655**
4.806	**2.330**	**4.806**	**-2.330**
8.290	**2.911**	**8.290**	**-2.911**
12.615	**3.380**	**12.615**	**-3.380**
17.693	**3.722**	**17.693**	**-3.722**
23.422	**3.929**	**23.422**	**-3.929**
29.687	**4.001**	**29.687**	**-4.001**
36.361	**3.945**	**36.361**	**-3.945**
43.311	**3.776**	**43.311**	**-3.776**
...	**...**	**...**	**...**

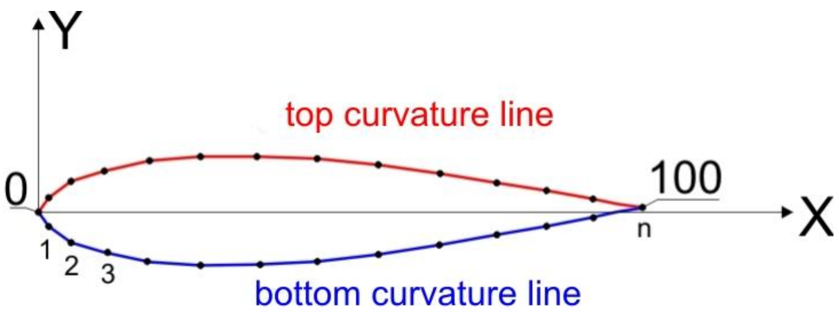


Fig. 4 – Wing airfoil *.prf file definition - example

File [*name*.**f**] – fuselage geometry – file description

- comment line (not necessary)

FUSELAGE GEOMETRY FILE EXAMPLE

Number of points in one section**15**

Number of sections **10** #n number of defined fuselage frames/ sections

Section **0** #0 first fuselage frame/ section

#section def. point number	# **Y** coordinate	# **Z** coordinate
0.000	**0.000**	**0.000**

Section **1** #0 second fuselage frame/ section

#section def. point number	# **Y** coordinate	# **Z** coordinate
-3.3	**0.000**	**-0.400**
-3.3	**0.100**	**-0.390**
-3.3	**0.200**	**-0.350**
-3.3	**0.280**	**-0.280**
-3.3	**0.350**	**-0.200**
-3.3	**0.390**	**-0.100**
-3.3	**0.400**	**0.000**
-3.3	**0.400**	**0.000**
-3.3	**0.400**	**0.000**
-3.3	**0.390**	**0.100**
-3.3	**0.350**	**0.200**
-3.3	**0.280**	**0.280**
-3.3	**0.200**	**0.350**
-3.3	**0.100**	**0.390**
-3.3	**0.000**	**0.400**

Section **2**

#section def. point number	# **Y** coordinate	# **Z** coordinate
-2.3	**0.000**	**-0.610**
-2.3	**0.160**	**-0.590**
-2.3	**0.300**	**-0.530**
-2.3	**0.430**	**-0.430**
...	**...**	**...**

Section **9** #(n-1) – number of the last fuselage Frome/ section

#section def. point number	# **Y** coordinate	# **Z** coordinate
1.3	**0.000**	**0.000**

Yellow marked section def. points belong to 3 independent stringers (Fig. 5), their coordinates are the same outside the area where wing or horizontal tail penetrates fuselage.

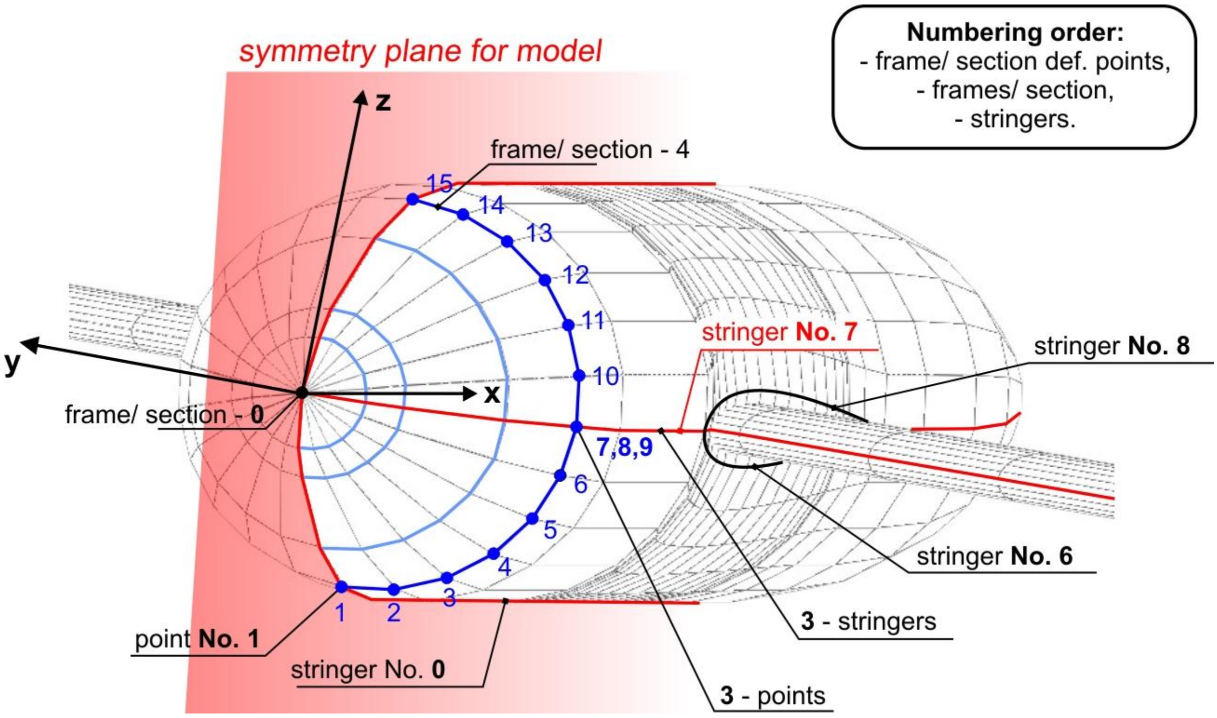


Fig. 5 – Fuselage geometry definition

IMPORTANT NOTES:

- Fuselage geometry is described with fuselage frames/ sections.
- The stringer which intersects the first leading edge or wing/ horizontal tail point, separates into two extra stringers which pass round the wing. Those extra stringers are defined in fuselage geometry file and they are count to the max. stringer number for current fuselage.
- Point numbering order is not the same for frames and stringers, **Fig. 5**
- First and last fuselage frame/ section reduce to a point.
- **Y** coordinate for points defining fuselage frames/ sections can't be negative.
- **Y** coordinate for the first and last point on a single frame/ section must be the same.

File [*name*.**ms2**] – complete aircraft model – file description

- comment line (not necessary)

AIRCRAFT GEOMETRY FILE EXAMPLE

# MAIN FILE SECTION	
begin	# *key word*
26.6	# *wing area (REAL)*
1.91	# *wing MAC (REAL)*
8.56	# *wing span (REAL)*
6.43	# ***x** coordinate for 0.25 MAC (REAL)*
0.00	# ***z** coordinate for 0.25 MAC (REAL)*
1	# *model scale factor (INTEGER)*
*****	# *key separation signs*
test_01	# *output file name „*.inp"*
2	# *number of independent wings, this wing has got defined distribution along chord and it forces its distribution over the fuselage-wing penetrate area (e.g.: fuselage gets the section/ frame distribution from wing).*
1	# *number of dependent wings, this wing hasn't got defined distribution along chord. It gets its distribution from the neighboring geometry (e.g.: wing gets the chord distribution from fuselage sections/ frames).*
0	# *object symmetry flag, „**0**" – object is symmetrical, „**1**" – only right half is taken into account, „**-1**" – only left half is taken into account during analysis.*
end	# *key word*

(Bracket beside the lines from "26.6" to "*****":) This data can be omitted. Program will set it automatically. This procedure is not correct when first wing section is placed not on the models symmetry plane.

# WING SECTION – WING „0"	
begin_wing0	# *wing No. „**0**" start section – main wing*
1	# *both wing ends closed with ribs (INTEGER)* **0** – *wing not closed with rib* **1** – *wing closed with tip rib* **2** – *wing closed with tip and root rib*
1	# *wing – fuselage intersection type (INTEGER)* **1** – *intersection (rib No. „**0**" – inside fuselage geometry rib No. „**1**" - outside fuselage geometry);* **0** – *no intersection (fuselage is fixed to the first rib of the wing, rib No. „**0**" rib is outside the fuselage geometry)*
7	# *number of the stringer which intersects the with wing leading edge point (this number is consistent with number in fuselage geometry file ,Fig. 5)*
3	# *number of sections defining wing (INTEGER)* *see description below:*
nac65006.prf	# *airfoil type for current wing section ("***.prf**" file name)*
5.20	# *rib chord (REAL)*
6.27 1.09 1.01	# *rib leading edge coordinates (REAL)*
0.00 1.00 0.00	# *rib rotation angles X, Y, Z [deg] (REAL)*

(Bracket beside the last four lines:) section - **1**

```
0                # current section number (INTEGER)
nac65006.prf     }
4.09             }
8.96  2.21  0.94 }  section - 2
0.00  0.00  0.00 }
1                }
nac65004.prf
1.60                                              }
13.90  4.31  0.73                                 }  section - 3
 0.00  0.00 -2.03                                 }
8                                                 }
0 0.5 1.25 2.5 5 7.5 10 15 20 30 40 50 60 74 90 100
                 # wing chord distribution [%MAC] there are additional option available:
                   linear 15 – equal length distribution
                   cosine 15 – cosine distribution
end_wing         # wing No. „0" end section – main wing
# WING SECTION – WING „1"
begin_wing1      # wing No. „1" – start section, in this example it is horizontal tail
1                  The description is the same as in the wing No. "0"
1
…
end_wing         # wing No. „1" – end section – horizontal tail
# WING SECTION – WING „2"
begin_wing2      # wing No. „2" – start section, in this example it is vertical tail
                   The description is the same as in the wing No. "0"
1
0                # 0 – no intersection (fuselage is fixed to the first rib of the vertical tail,
                   V-tail rib No. „0" is outside the fuselage geometry)
16
3
nac65006.prf
5.00
10.58  0.00  1.52
90.00  0.00  0.00
1
nac65006.prf
3.54
12.57  0.00  2.25
90.00  0.00  0.00
3
nac65004.prf
1.93
15.69  0.00  4.35
90.00  0.00  0.00
7
unknown          # undefined V-tail chord distribution
                   (program will set it automatically)
```

bottom	#	*key word „**bottom**" or „**top**" means that the current wing has got only bottom or top surface (it applies only to vertical wings – vertical tail, etc., the other half of the wing is created as a mirror copy*
end_wing	#	*wing No. „**2**" – end section – vertical tail*
# WING SECTION – WING „3"		
begin_wing3	#	*wing No. „**3**" – start section, in this example it is wing end plate*
2		*The description is the same as in the wing No. "**0**"*
0		
0		
3		
…		
end_wing	#	*wing No. „**3**" – end section – wing end plate*
# FUSELAGE SECTION		
begin_fuselage	#	*beginning of the fuselage section*
1	#	**1** – *fuselage is defined,* **0** – *fuselage does not exist (the rest of this section can be omitted)*
test_01.f	#	*fuselage geometry file name*
7	#	*fuselage extra sections/ frames (set automatically)*
9.9 10.5 11.3 11.6 11.9 12.3 12.6	#	*extra section/frame X coordinate*
end_fuselage	#	*end of fuselage section*
# CONNECTIONS SECTION		
begin_connections	#	*the beginning of horizontal connections section*
1	#	*number of connections (when 0 the rest of this section can be omitted)*
2 1 0 0 0 15 1	#	*connecting V-tail (**wing no. 2**) with H-tail (**wing No. 1**), values description:* **2** – *wing number – V-tail (begin_wing**2**)* **1** – *wing number – H-tail (begin_wing**1**)* **0** – *V-tail rib number which is connected with H-tail:* **0** – *first,* **1** – *last* **0** – *H-tail rib number which is connected with V-tail:* **0** – *first,* **1** – *last* **0** – *H-tail stringer number which is connected with V-tails leading edge* **15** – *H-tail stringer number which is connected with V-tails trailing edge* **1** – *H-tail surface which will be modified:* **0** – *bottom surface,* **1** – *top surface*
end	#	*the end of horizontal connections section*
begin_connections_V	#	*the beginning of vertical connections section*
1	#	*number of vertical connections, e.g.: wing end plate + wing (when 0 the rest of this section can be omitted)*
0 3 1 3 0 16 1	#	*wing end plate(**wing no. 3**) (vertical) with wing (**wing no. 0**) (horizontal):* **0** – *horizontal wing number – wing (**wing no. 0**)* **3** – *vertical wing number – wing end plate (**wing no. 3**)* **1** – *horizontal wing rib number which will be connected with vertical wing:* **0** – *first,* **1** – *last rib* **3** – *vertical wing rib number which will be connected with horizontal wing*

	0 – *vertical wing stringer number which will be connected with horizontal wing leading edge* **16** – *vertical wing stringer number which will be connected with horizontal wing trailing edge* **1** – *vertical wing surface which will be modified:* **0** – *outside surface,* **1** – *inside surface*
end	# *the end of vertical connections section*

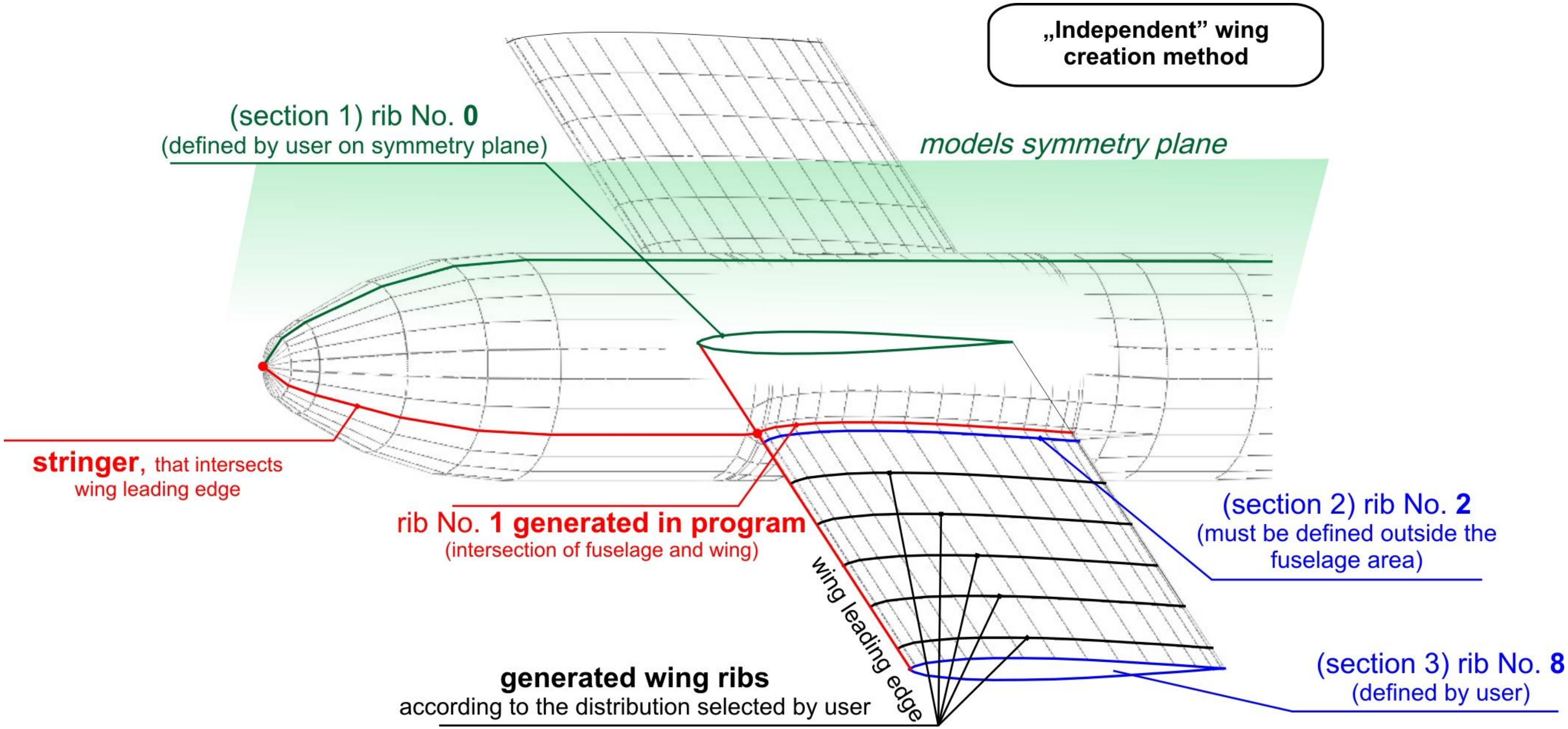


Fig. 6 – The example of „independent" wing creation method (see **wing No. 0** definition above in text)

1.3.2. Output data file description

File [*name*.**OUT**]

Global aerodynamic results:

- comment line

OUTPUT FILE STRUCTURE EXAMPLE

```
Data from file:
C:/Users/Lucas/Panukl/dat/panukl/predator.pan  # file path

Geometry data: # geometry reference data
S   =  10.00
MAC =  0.74
B   =  14.70
Coordinates of reference point for moments calculation:
X   =  3.31 Y   =  0.00

Angle of attack, sideslip angle and Mach number:
Alfa =   5.0
Beta =   0.0
Mach =   0.0
angular velocities:
P =   0.0
Q =   0.0
R =   0.0

Global results : # global results for current object

in body axis system: # global results for current object in body axis system
Cx =  -0.0488533498
Cy =  -0.000400980979
Cz =   0.757529054
Cl =   0.000270848351
Cm =  -0.303237661
Cn =   0.000253265641

in stability axis system: # # global results for current object in stability axis system (related to ¼ MAC)
Cz =   0.758904277
Cx =   0.0173555587

Induced drag and corresponding lift coefficient:
Cxi=   0.00832755596
Czi=   0.722843174
```

File [*name*.**TXT**]

The results for pressure coefficient, velocity, source or doublet distribution etc. (for each panel of aircraft body) are placed in a single TXT file (easy to use file in most graph software):

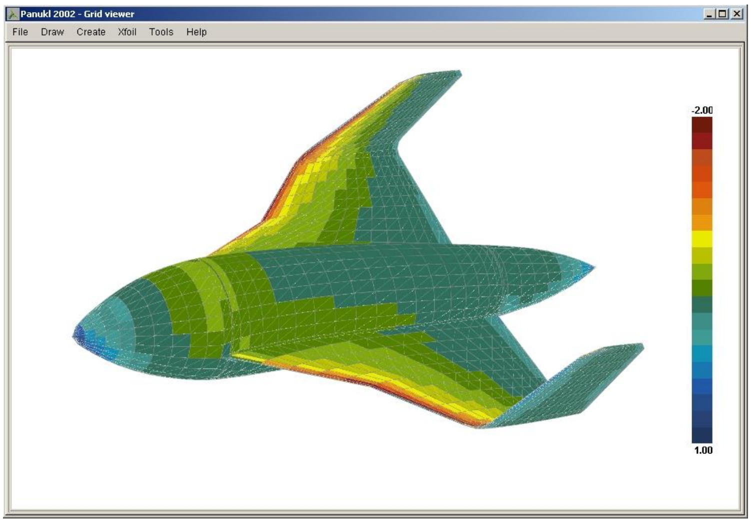


Fig. 7 – Pressure coefficient distribution for current aircraft body – example results.
Data saved in *.TXT file

File [*name*.**EPS**]

Wing downwash/ angle of deviation results (see chapter 3.1.4):

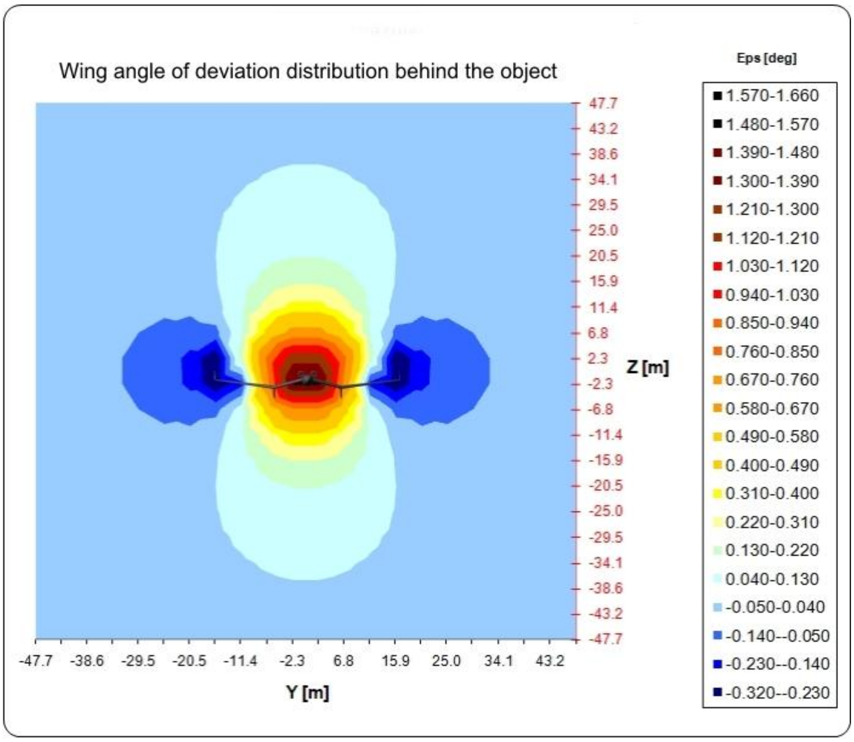


Fig.8 – Example wing angle of deviation results for analyzed object
(graph made in MS Excel with *.EPS result file)

File [*name*.**BLN** and *name*.**EPS]**

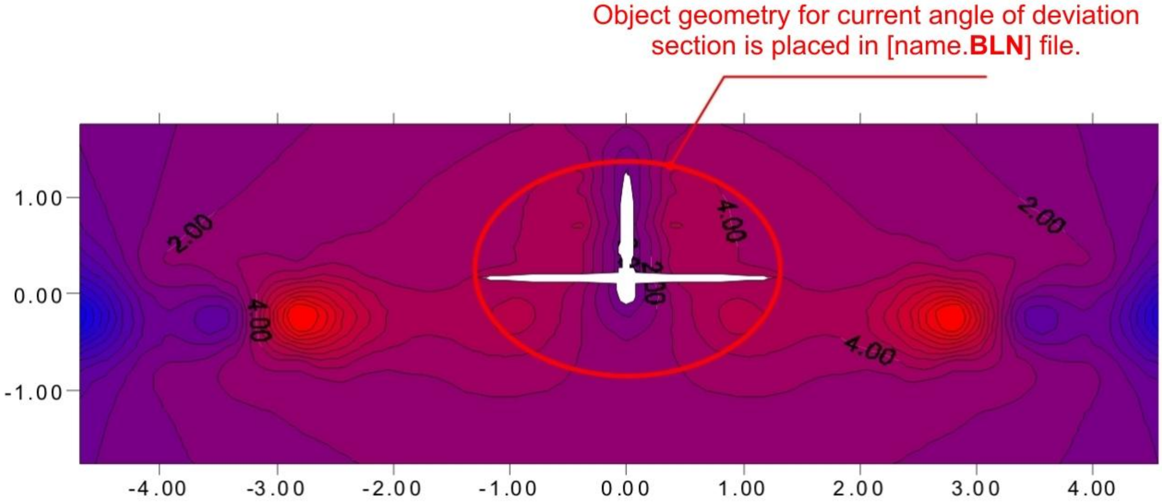


Fig. 9 – Example angle of deviation results near horizontal tail area
(graph made in GRAPHER with *.BLN & *.EPS result files)

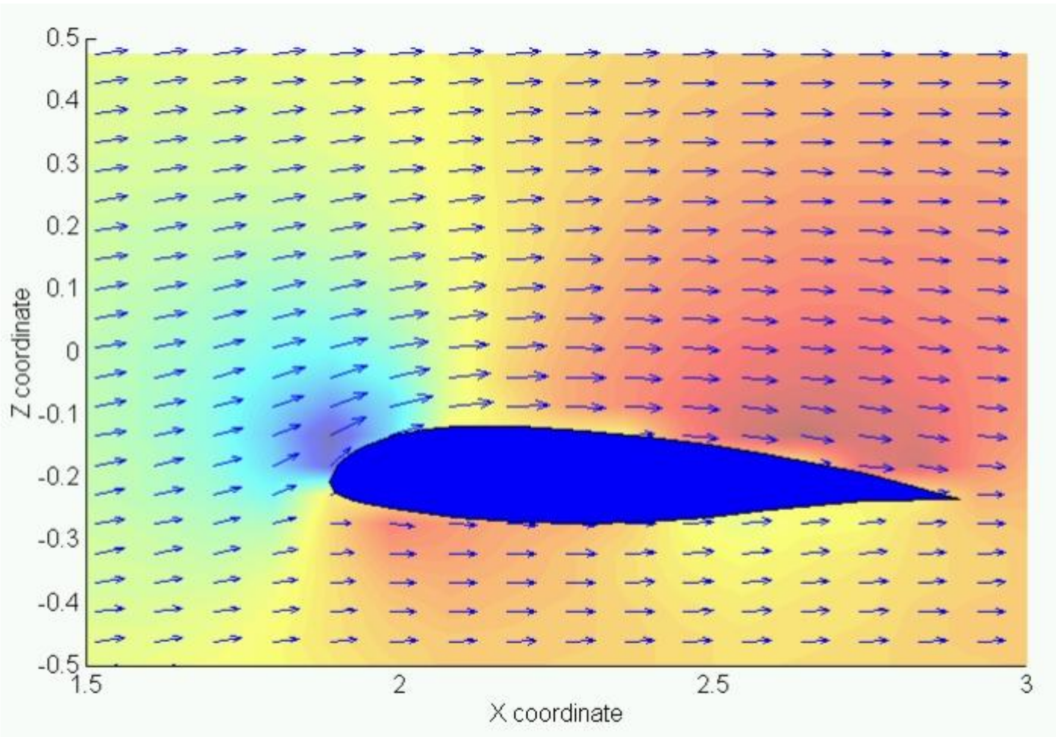


Fig. 10 – Example angle of deviation results near wing section area
(graph made in MATHLAB with *.BLN & *.EPS result files)

File [*name*.**CZY]**

Aerodynamic coefficients distribution over the wing – results, (Y, Cz, Cm, Cxi, Si, Ci):

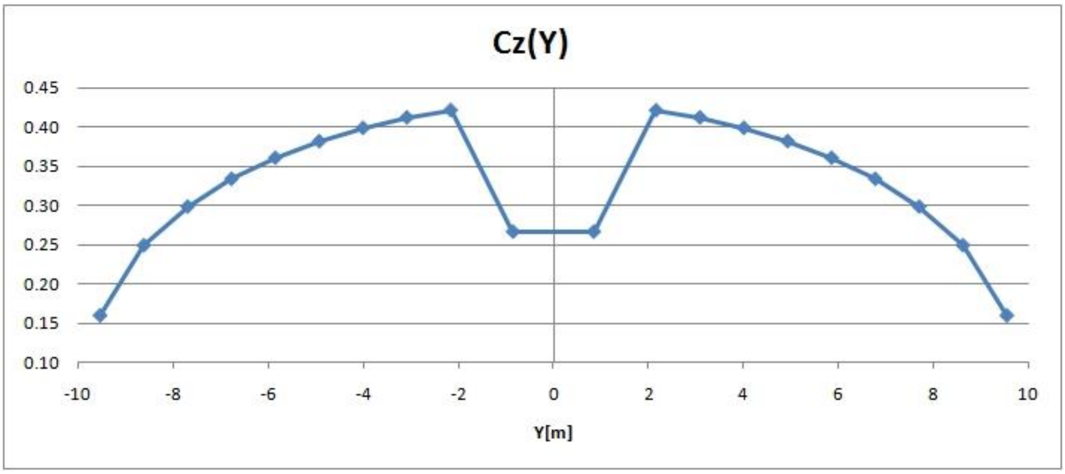


Fig. 11 – Example results – Lift coefficient distribution vs. wing span (results from *.CZY file)

2. Installation process

PANUKL software is made for PCs with **Microsoft – Windows 2000/ Windows XP/ Windows Vista** software. Additionally it will work on **Linux** based platforms.

Before installation process user must download the latest version of **PANUKL** software suitable for current operating system. For latest version of program go to: http://itlims.meil.pw.edu.pl/zsis/pomoce/PANUKL/panukl.htm - **Files to download** – card.

2.1. PANUKL installation guide In MS WINDOWS

Step 1) Download the: **Panukl_Setup.zip** archive file and unpack its content to a free folder on your hard drive.

Step 2) Run: **Panukl_Setup.exe** – The installation window appears Fig. 12.

Fig. 12 **–** Installation Welcome window

Step 3) Click **NEXT** button and choose the destination folder for **PANUKL** software to install to. The default setting is **C:\Program Files\Panukl**. Click **BROWSE** button to change the default installation folder Fig. 13.

Click **CANCEL** button to stop installation. To go to previous installation window click **BACK**.

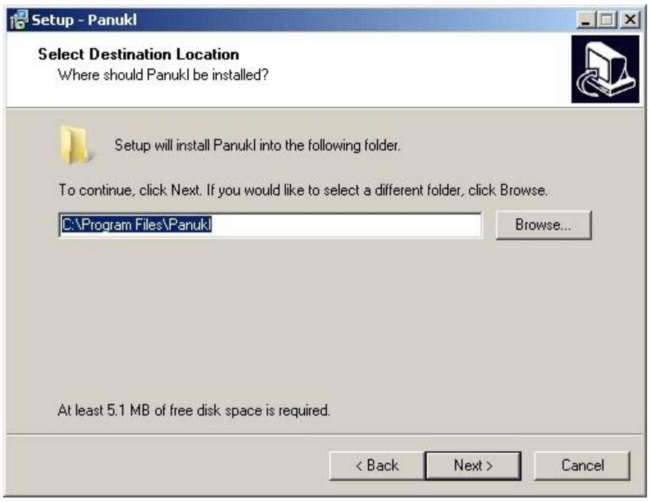


Fig. 13 – Destination Folder selection

Step 4) In next installation setup window, user can select the **PANUKL** software components to install (or not) Fig. 14. **XFOIL** and **FUSELAGE** components are not essential to run **PANUKL**.

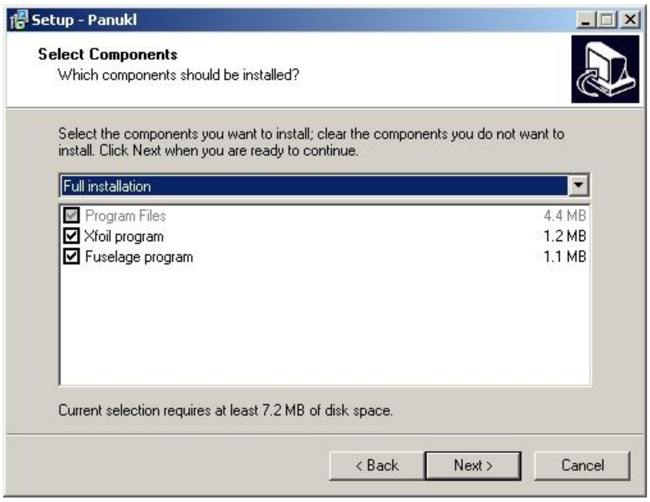
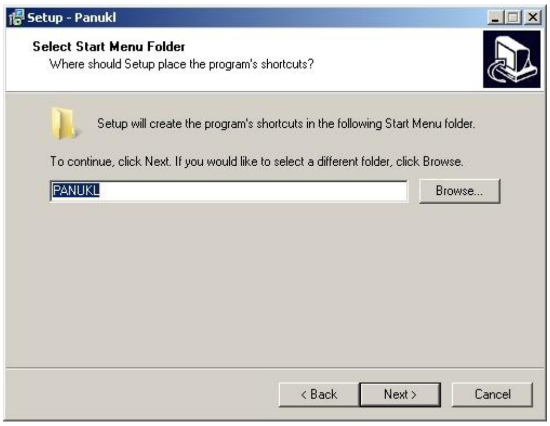


Fig. 14 – Select components window

Step 5) In next installation setup window user is asked to select **Start Menu** folder for program's shortcuts. Additionally one can choose program's start icon to create on **Desktop** and in **Quick Launch Bar** Fig. 15.



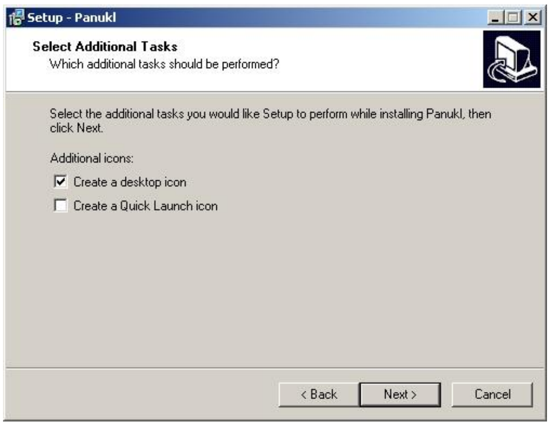


Fig. 15 Fig. 16

Step 6) After selecting available options, setup displays review window. Click **INSTALL** button to proceed with the installation. To end installation process click **FINISH** button Fig. 17**.**

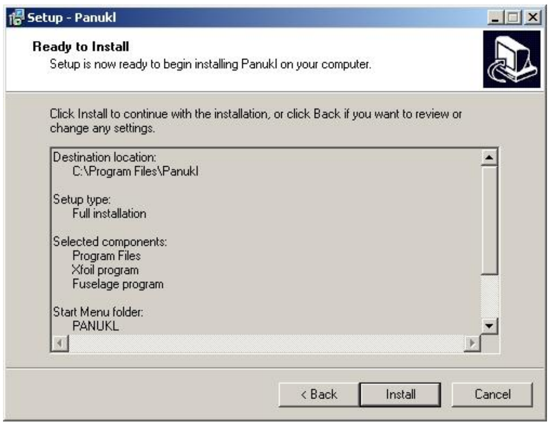


Fig. 17 Fig. 18

The end of installation process)

PANUKL first start) To start program click **GridView** [3] **PANUKL** icon.

During the first start **/panukl** subfolder will be created In users home folder. The ***.ini** and *.**log** files will be stored in there. Additionally user will be asked to create **DATA** subfolders: **/DAT** & **/OUT**.

The proper **DATA** subfolder structure is essential to work with **PANUKL**. User can create subfolder structure also in **GridView** managing subprogram [3.1.6].

2.2. PANUKL installation guide in LINUX

Step 1) Download the: **Panukl_Setup.zip** archive file and unpack its content to a free folder on your hard drive.

Step 2) When **PanuklLinux.zip** archive is unpacked, run installation script with: „**sh PanuklSetup.sh"** command.

When script is run from **user** level, **PANUKL** will install to **/home/username/panukl**. Additionally the desktop icon will be created for **GridView** – managing application.

When script is run from **root** level, **PANUKL** will install to **/usr/local/panukl.** Additionally the symbolic **panukl** start link will be created in **/usr/local/bin.**

The end of installation process)

PANUKL first start) To start **PANUKL** type **panukl** in terminal window – the desktop icon will be created during the first start.

During the first start **/panukl** subfolder will be created In users home folder. The ***.ini** and *.**log** files will be stored in there. Additionally user will be asked to create **DATA** subfolders: **/DAT** & **/OUT**.

PANUKL Requirements) **GLIBC** ≥ 2.3, **libXft.so.2**, **libXext.so.6,** additionally **XFOIL** program requires **Fortran 77** libraries. Some of the new **LINUX** distributions need **compat-libf2c**. To run **XFOIL** program from **GridView** managing application, interface install **xterm** software.

3. Working with PANUKL

3.1. PANUKL GUI description

To run **PANUKL's GUI**, click **GridView.exe** managing subprogram icon, 

on the **DESKTOP** or in the **START MENU**. After few seconds the main application window displays Fig. 19. **PANUKL** is ready to operate.

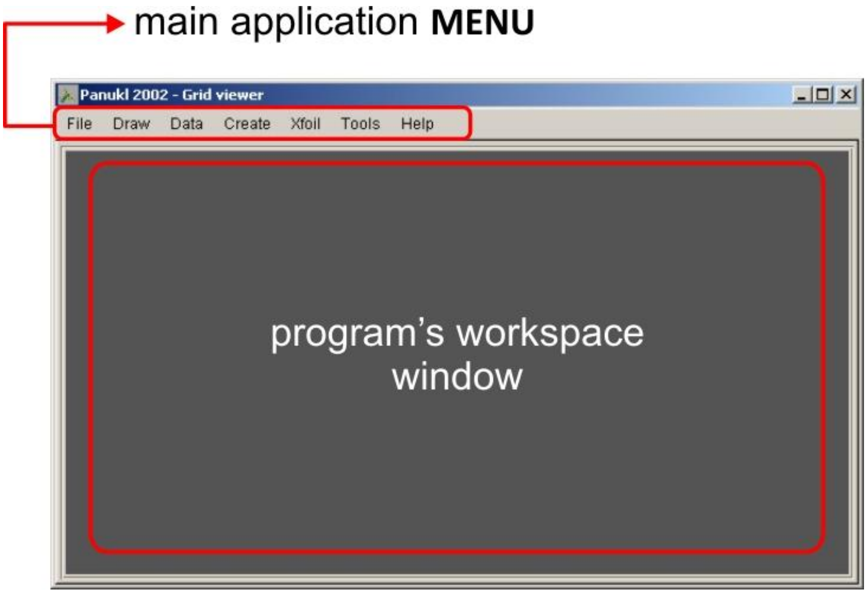


Fig. 19 – Main application window

To change program window size use standard **WINDOWS** buttons. User can access the particular program functions from drop down managing application **MENU.**

3.1.1. FILE menu description

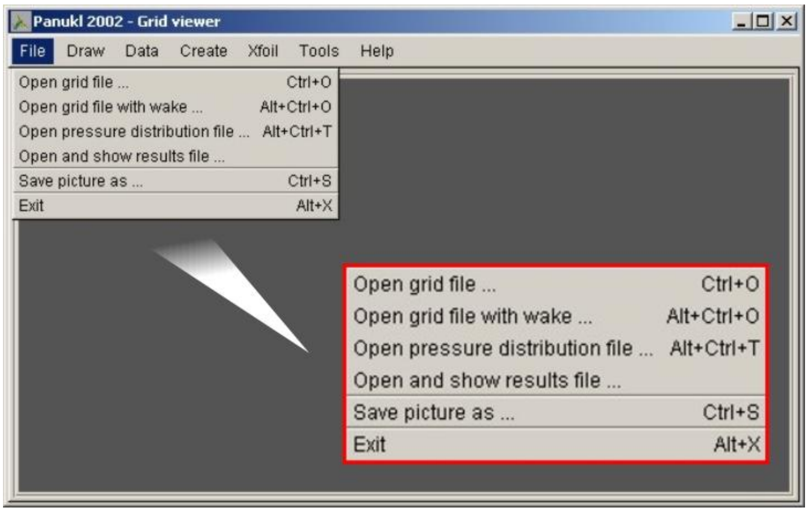


Fig. 20 – FILE menu

Available options in FILE menu)

Function	Description
Open grid file [Ctrl+O]	Open grid file ***.inp**, from user selected disc location, Fig. 21.
Open grid file with Wake [Alt+Ctrl+O]	Open grid file with wake ***.dat**, from user selected disc location.
Open pressure distribution file [Alt+Ctrl+T]	Open output ***.txt** file with pressure distribution for current analyzed body (for each grid panel).
Open and show results file	Open and show global results ***.out** file in external window, Fig. 23.
Save picture as [Ctrl+S]	Save current graphical window to **JPEG** , **PNG** or **BMP** file , Fig. 22.
Exit [Alt+X]	End program, exit application.

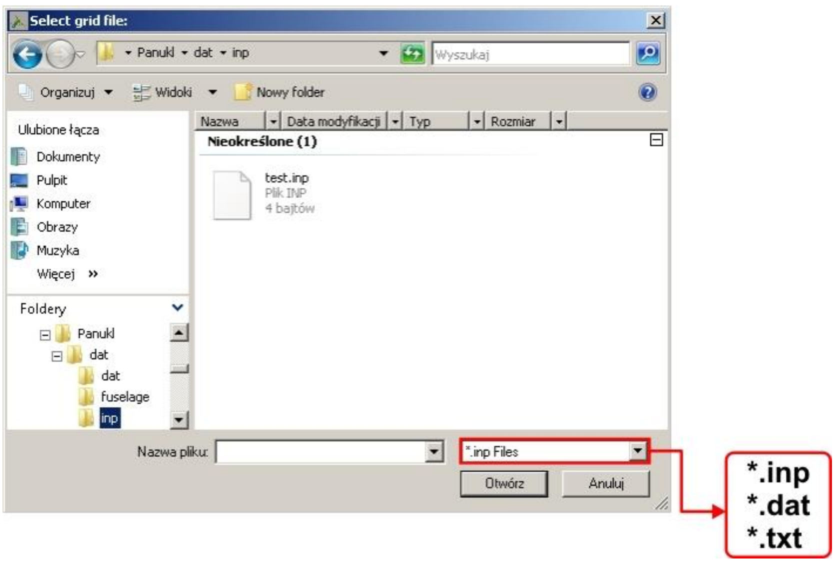


Fig. 21 – File selection window example

File selection Windows can be different dependent on the current operating software version, window looks does not influence PANUKL's functionality.

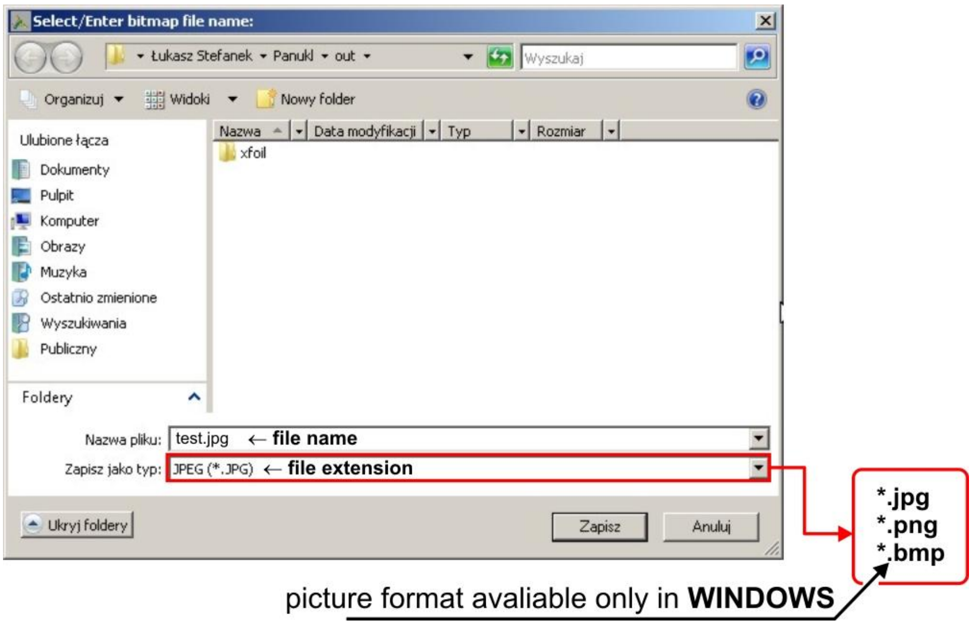


Fig. 22 – Save to graphic file current PANUKL window

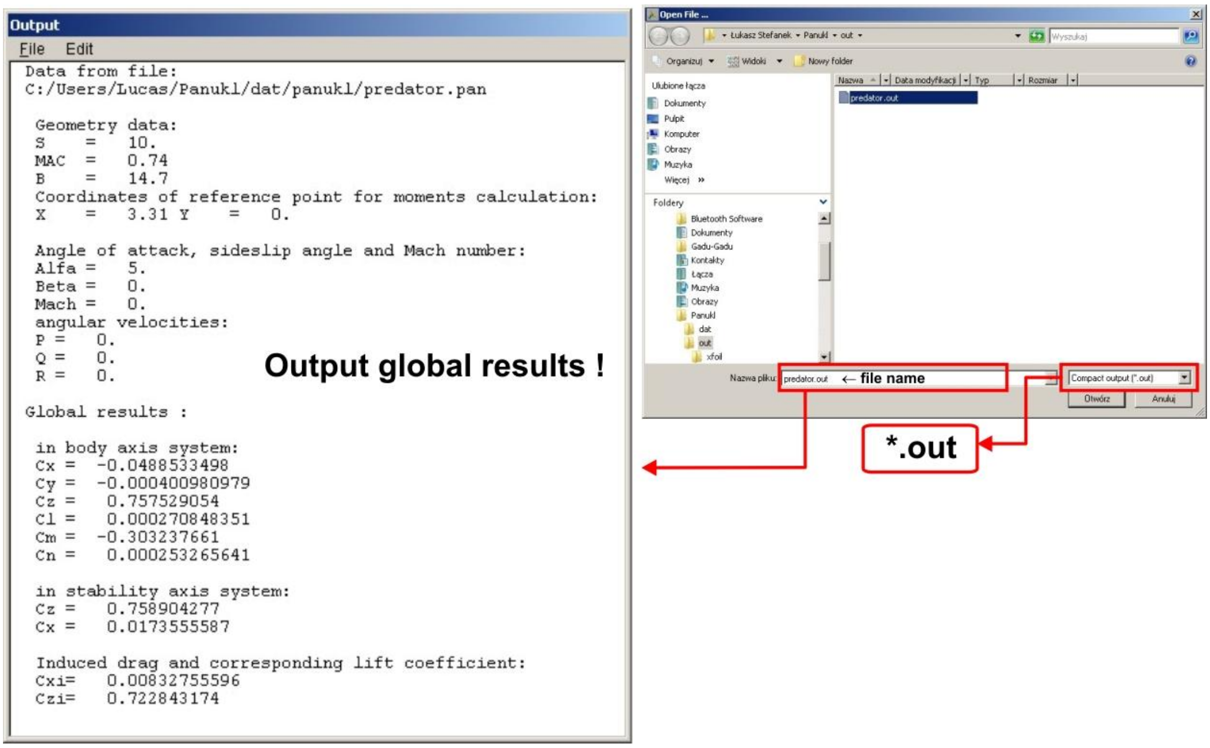


Fig. 23 – Output results window example

3.1.2. DRAW menu description

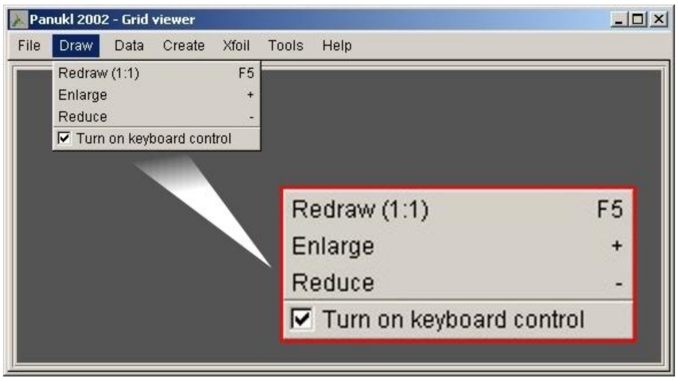


Fig. 24 – Draw menu

Available options in DRAW menu)

Function	Description
Redraw (1:1) [F5]	Redraw current object in main application window.
Enlarge [+]	Zoom in viewport (in main application window).
Reduce [-]	Zoom out viewport (in main application window).
Turn on keyboard control [check box]	Turn on keyboard control for analyzed object , Fig. 25.

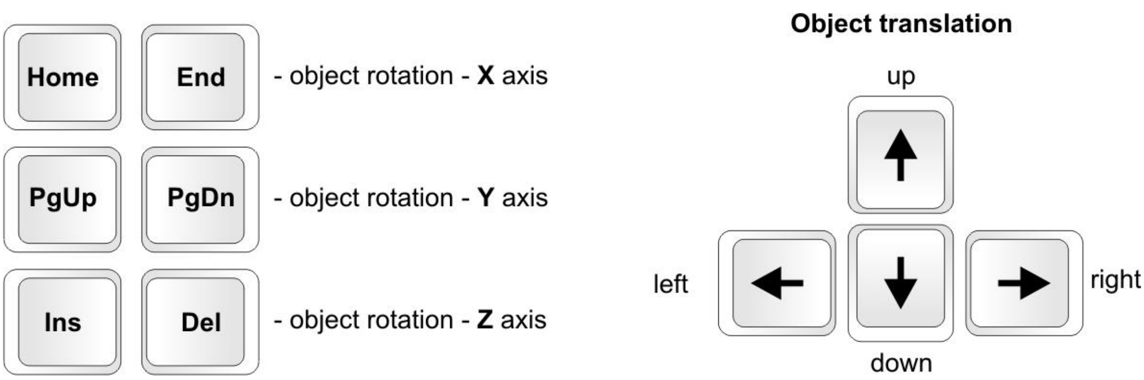


Fig. 25 – Keyboard controls explanation

3.1.3. DATA menu description

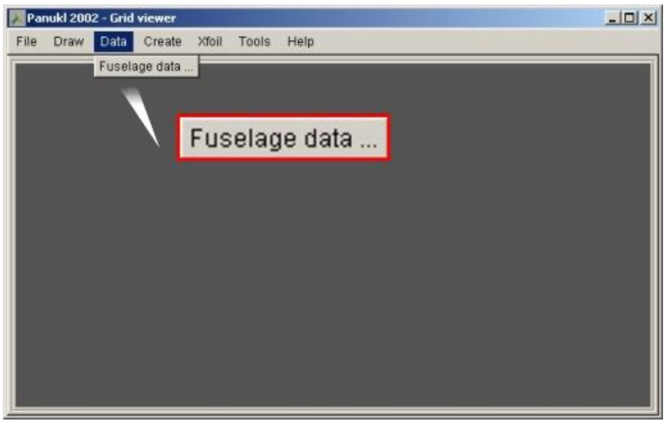


Fig. 26 – DATA menu

In **DATA** menu user can find subprograms that are part of the **PANUKL** application. Now there is only one subprogram **FUSELAGE DATA** which can help to create fuselage geometry file **[*name*.f]**. For more information on **FUSELAGE DATA** go to chapter 4.2.

3.1.4. CREATE menu description

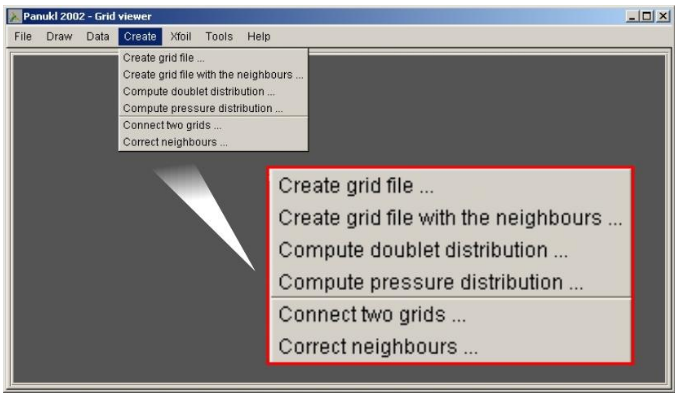


Fig. 27 – CREATE menu

CREATE menu is the most important menu in **PANUKL**. User can perform the complete computational session with functions from CREATE menu for current analyzed object (aircraft).

Available options in CREATE menu)

Function	Description
Create grid file	This command will run **Mesh.exe**, **PANUKL** application component. User will be asked to point input file (complete aircraft geometry file***.ms2**). Based on input file the output ***.inp,** geometry gird file will be created, Fig. 28.

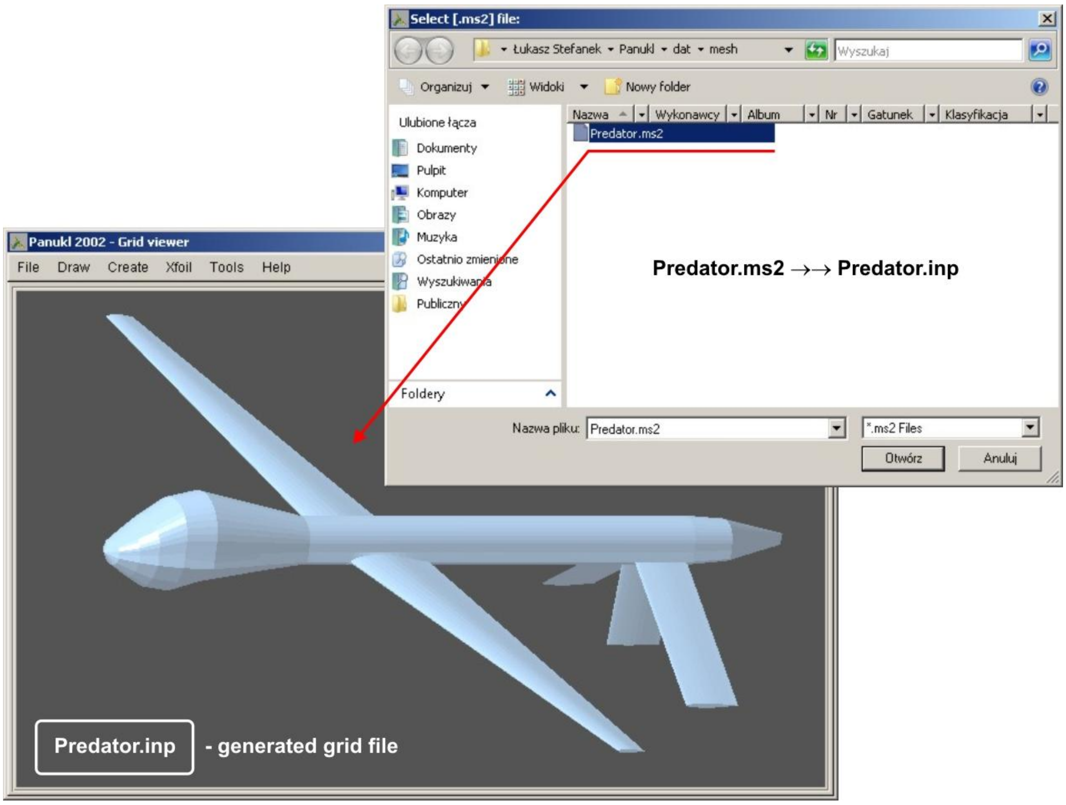


Fig. 28 – Creating grid file for current input geometry data

Function	Description
Create grid file with the neighbours	This command will run **Neigh.exe**, **PANUKL** application component. User will be asked to point input grid file (***.inp**). Based on input geometry file the output ***.dat,** gird with wake file will be created, Fig. 29. The ***.dat** file contains information about grid, wake and numbers of „neighbours" for current grid panels.

Option No. 1 – <u>we do have</u> saved on disk configuration file ***.ngh**, Fig. 29

Run **Create grid file with the neighbours** and select saved configuration file ***.ngh** – file contains all necessary information to create ***.dat** file. To open selected ***.ngh** file click **OPEN** button. Configuration window will appear (Fig. 30) where one can see saved ***.dat** file creation options. To generate ***.dat** file click **Save and Compute (ok)** button.

Option No. 2 – <u>we do not have</u> saved on disk configuration file ***.ngh**, Fig. 29

Run **Create grid file with the neighbours** and click **CANCEL** button when prompted for saved configuration file ***.ngh**. The configuration window will appear (Fig. 30) where user can select options to create ***.dat** file. To save current ***.dat** options to ***.ngh** file click **Save [*.ngh] file as**, to create ***.dat** file click **Save and Compute (ok)** button.

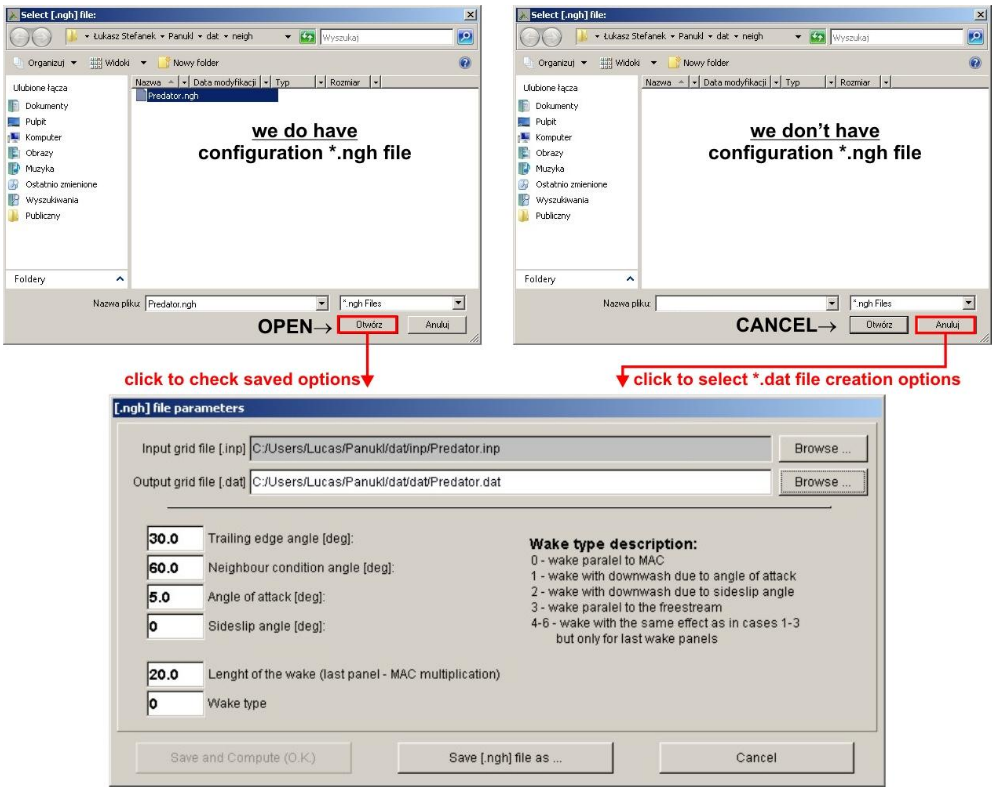


Fig. 29 – Creating *.dat file (grid with wake)

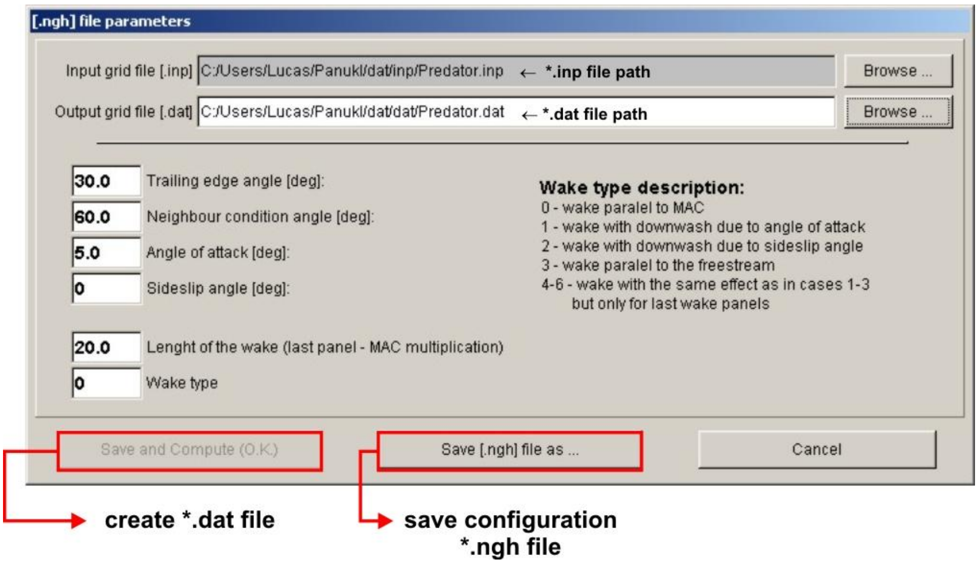


Fig. 30 – Configuration options – creating *.dat file (grid with wake)

Setting	Description
Length of the wake (MAC multiplication)	Length of the wake (MAC multiplication)
Wake type description	**<u>Wake type creation methods:</u>** **0** – Wake parallel to MAC **1** – Wake with downwash due to angle of attack

	2 – Wake with downwash due to sideslip of attack **3** – Wake parallel to the free stream **4 , 5, 6 –** Wake with the same effect as in cases 1-3 but only for last wake panels
Trailing edge angle [deg]	Trailing edge angle. If the angle between two trailing edge grid panels is lower or equal to defined value, than wake line will be created from such trailing edge.
Neighbour condition angle [deg]	Neighbor condition angle **.** If the angle between two neighboring grid panels is higher than defined value, both panels are not treated as neighbours.
Angle of attack [deg]	Angle of attack (taken into account during wake creation).
Sideslip angle [deg]	Sideslip angle (taken into account during wake creation).

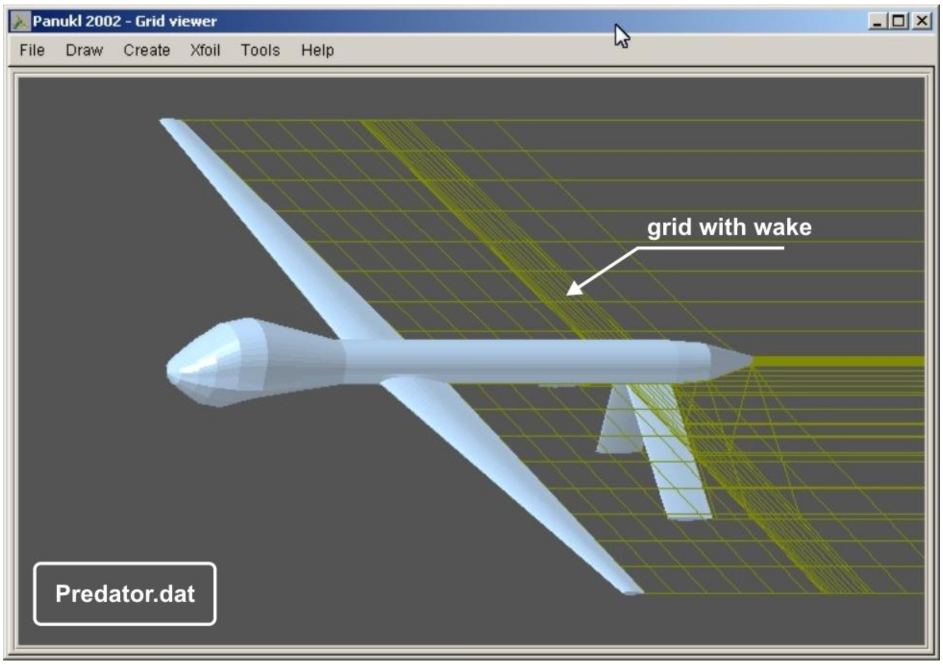


Fig. 31 – *.dat file example: grid & wake „Predator.dat”

Function	Description
Compute doublet distribution	This command will run **Panukl.exe**, **PANUKL** application component. User will be asked to point input grid with wake file (***.dat**). Based on input file the output ***.pan,** file will be created. The ***.pan** file contains computed results for velocity potential distribution for analyzed body.

Option No. 1 – <u>we do have</u> saved on disk configuration file ***.par**, Fig. 32

Run **Compute doublet distribution** and select saved configuration file ***.par** – file contains all necessary information to create ***.pan** file. To open selected ***.par** file click **OPEN** button. Configuration window will appear (Fig. 33) where one can see saved ***.pan** file creation options. To generate ***.pan** file click **Save and Compute (ok)** button.

Option No. 2 – <u>we do not have</u> saved on disk configuration file ***.par,** Fig. 32

Run **Compute doublet distribution** and click **CANCEL** button when prompted for saved configuration file ***.par**. The configuration window will appear (Fig. 33) where user can select options to create ***.pan** file. To save current ***.pan** options to ***.par** file click **Save [*.par] file as**, to create ***.pan** file click **Save and Compute (ok)** button.

Panukl.exe computes influence factors and solves system of equations. As a result we get velocity potential distribution. It is the most time and CPU consuming process. The computations can last long . Computation time rises to the third power with generated grid panels.

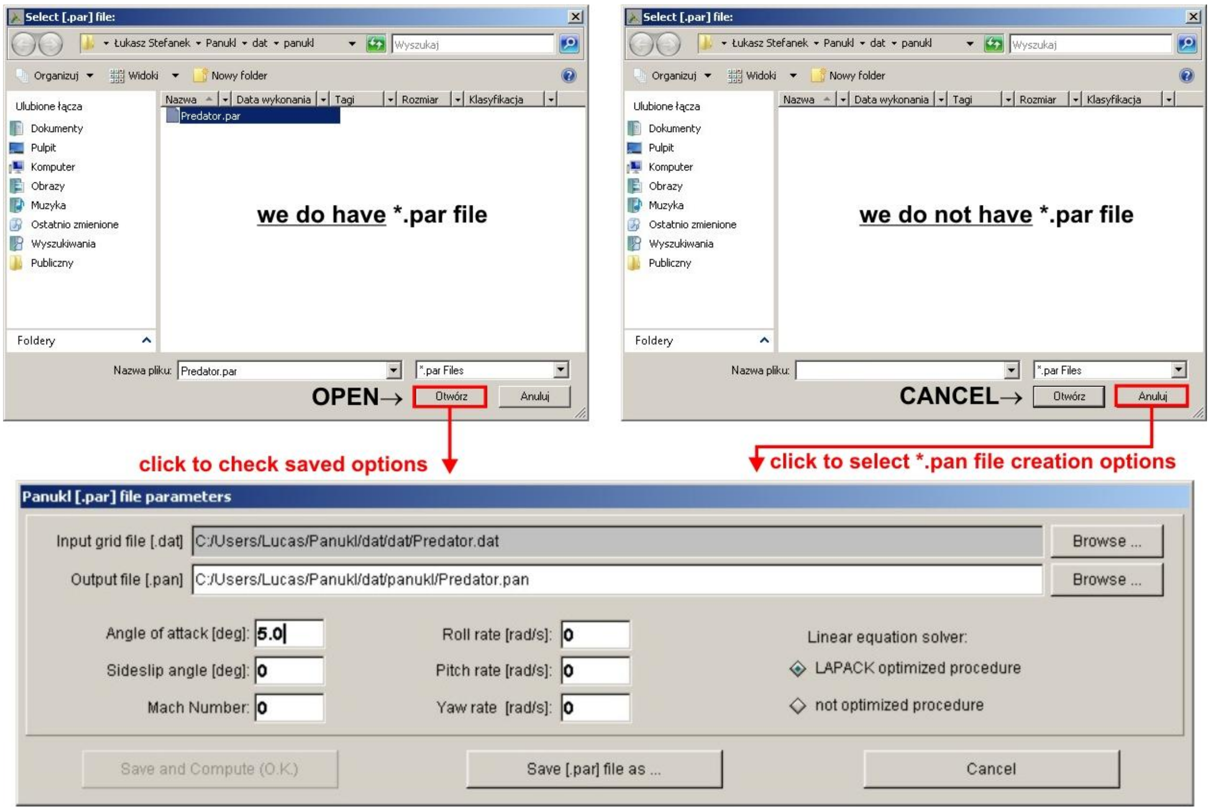


Fig. 32 – Creating *.pan file

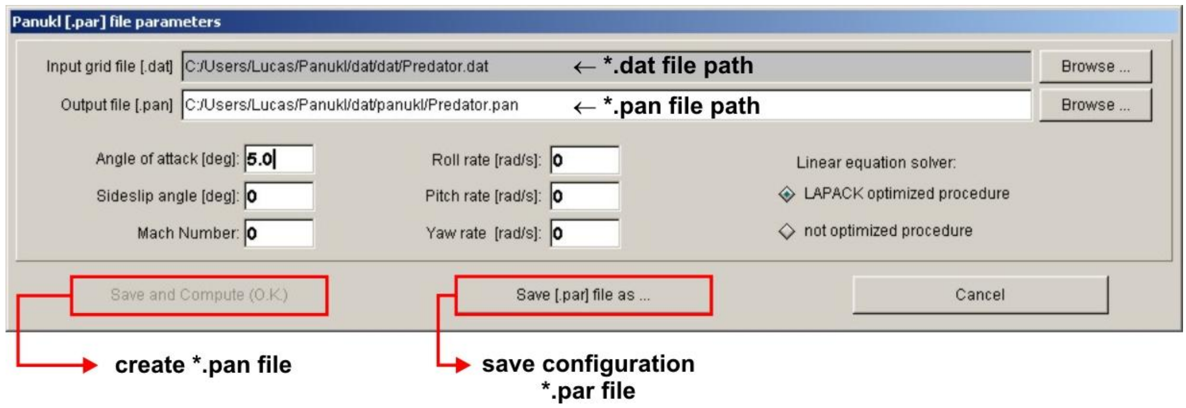


Fig. 33 – Configuration options – creating *.pan file (velocity potential distribution)

Setting	Description
Linear equation solver	Linear equation solver selection: - **LAPACK optimized procedure** (default) - **not optimized procedure** (more time consuming procedure but more accurate)
Angle of attack [deg]	Angle of attack [deg], measured from free stream velocity direction and **OX** axis**.**
Sideslip angle [deg]	Sideslip angle [deg].

Mach Number	Mach number
Roll rate [rad/s]	**P** – roll rate [rad/s]
Pitch rate [rad/s]	**Q** – pitch rate [rad/s]
Yaw rate [rad/s]	**R** – yaw rate [rad/s]

Function	Description
Compute pressure distribution	This command will run **Press.exe**, **PANUKL** application component. User will be asked to point input velocity potential distribution file (***.pan**). Based on input file the output ***.out, *.txt & *.eps,** files will be created. The ***.out** file contains computed global aerodynamic coefficients results for analyzed body. The ***.txt** file contains computed results for pressure coefficient, velocity, source or doublet distribution etc. for each aircraft body panel. The ***.eps** file contains wing angle of deviation results.

Option No. 1 – we do have saved on disk configuration file ***.prs**, Fig. 34

Run **Compute pressure distribution** and select saved configuration file ***.prs** – file contains all necessary information to create output result files – ***.out**, ***.txt**, ***.eps**. To open selected ***.prs** file click **OPEN** button. Configuration window will appear (Fig. 35) where one can see saved ***.prs** file creation options. To generate **output** files click **Save and Compute (ok)** button.

Option No. 2 – we do not have saved on disk configuration file ***.prs,** Fig. 34

Run **Compute pressure distribution** and click **CANCEL** button when prompted for saved configuration file ***.prs**. The configuration window will appear (Fig. 35) where user can select options to create **output** files – ***.out**, ***.txt**, ***.eps**. To save current **output** files options to ***.prs** file click **Save [*.prs] file as**, to create **output** files click **Save and Compute (ok)** button.

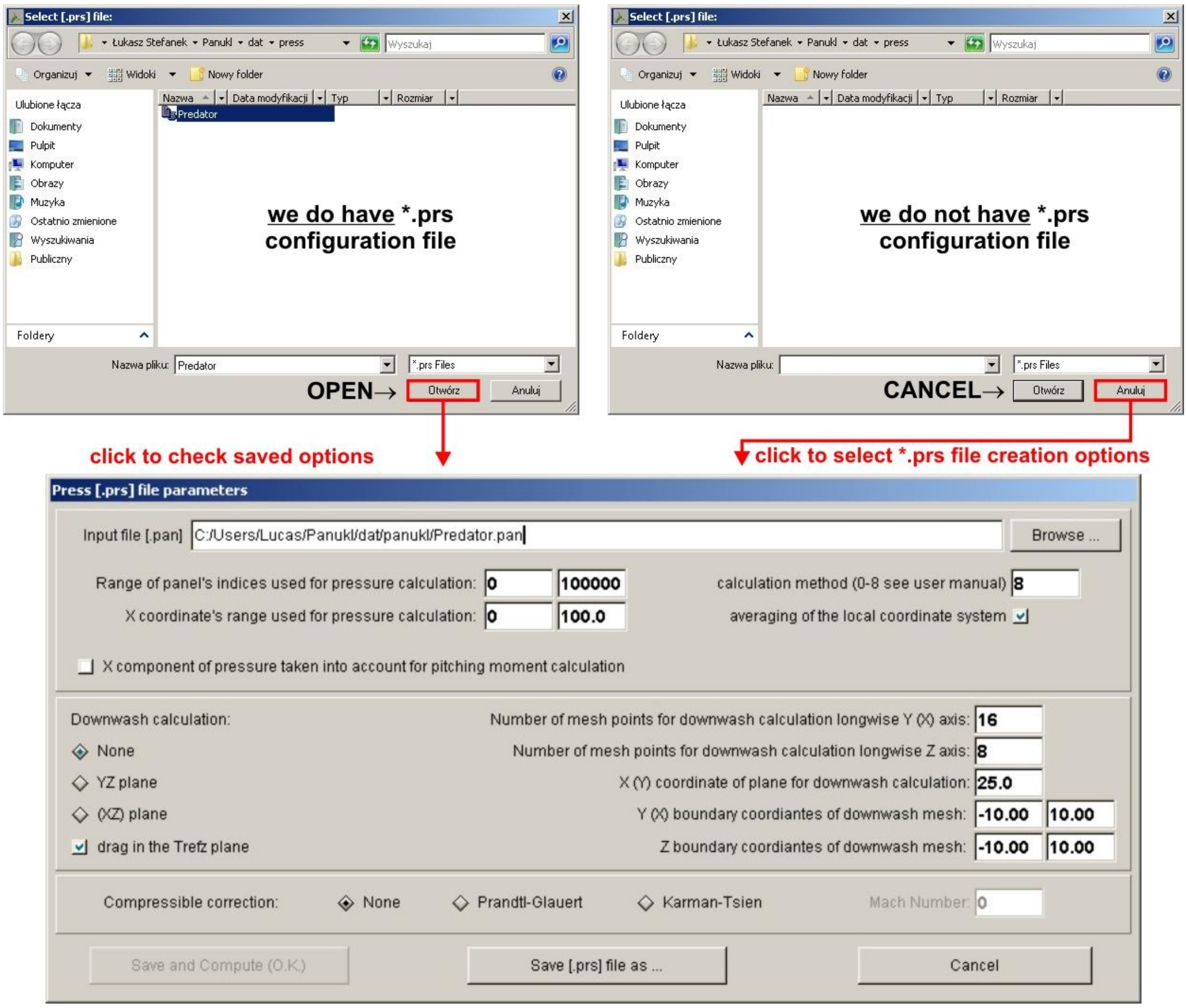


Fig. 34 – Creating output result files: *.out, *.txt, *.eps

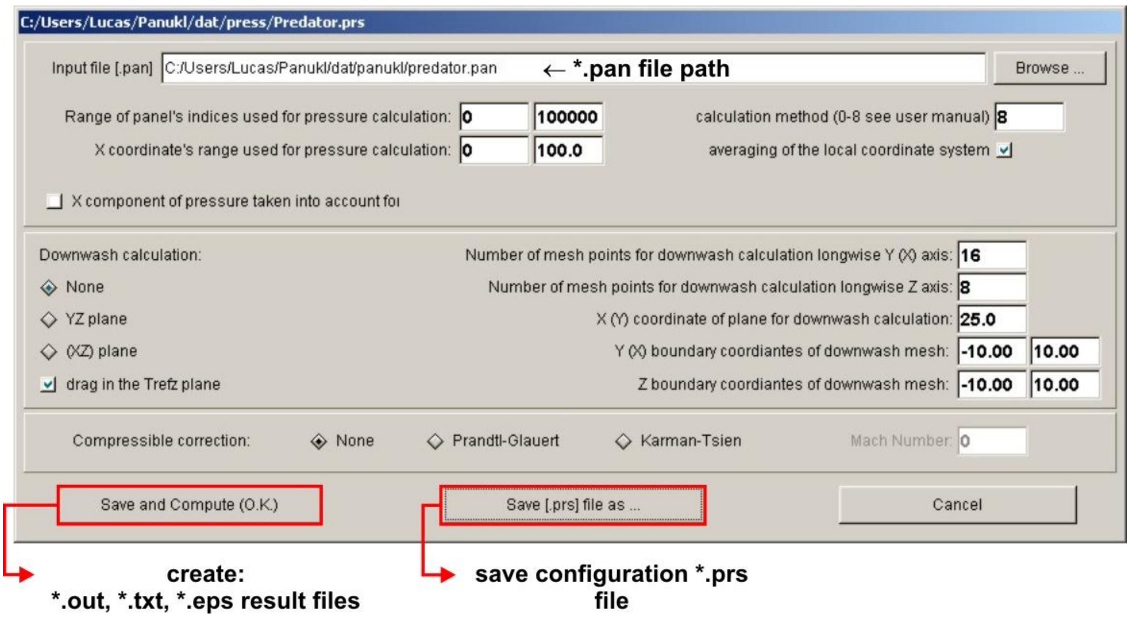


Fig. 35 – Configuration options – creating result files

Setting	Description
Range of panel's indices used for pressure calculation	The values are the numbers of the first and the last panel which will be taken into account for pressure computations.
X coordinate's range used for pressure calculation	X coordinate's range used for pressure calculation (global aerodynamic coefficients).
X component of pressure taken into account for pitching moment calculation	X component of pressure taken into account for pitching moment calculation.
Calculation metod (0-8)	Computation method selection: **0** – average from two out four of described below methods, **1** – collocation method – with polynomial: **$\varphi(x,y)=Ax^2y^2+Bx^2y+Cxy^2+Dxy+Ex^2+Fy^2+Gx+Hy+I$,** **2** – collocation method (omitting point on current panel) – with polynomial: **$\varphi(x,y)=Bx^2y+Cxy^2+Dxy+Ex^2+Fy^2+Gx+Hy+I$,** **3** – approximation with polynomial: **$\varphi(x,y)=Bx^2y+Cxy^2+Dxy+Ex^2+Fy^2+Gx+Hy+I$,** **4** – approximation with polynomial: **$\varphi(x,y)=Dxy+Ex^2+Fy^2+Gx+Hy+I$,** **5** – method 1, 2 i 3, **6** – method 1, 2 i 4, **7** – method 1, 3 i 4, **8** – (default) method 2, 3 i 4.
Averaging of local coordinate system	Averaging of local coordinate system to eliminate errors during potential differentiation.

Downwash calculation:	Downwash (angle of deviation) computations: **None** – downwash is not computed (result ***.eps** file is not created), **YZ plane** – downwash results are computed in **OYZ** plane**,** **XZ plane** – downwash results are computed in **OXZ** plane**,**
Number of mesh points for downwash calculation longwise Y (X) axis:	Number of mesh points for downwash calculation longwise **Y (X)** axis**.**
Number of mesh points for downwash calculation longwise Z axis:	Number of mesh points for downwash calculation longwise **Z** axis.
X (Y) coordinate of plane for downwash calculation:	**X (Y)** coordinate of plane for downwash calculation.
Y (X) boundary coordinates of downwash mesh:	**Y (X)** boundary coordinates of downwash mesh.
Z boundary coordinates of downwash mesh:	**Z** boundary coordinates of downwash mesh.
Drag In the Trefz plane	Drag in the **Trefz** plane computation

Compressible correction:	Compressible correction method for set **Mach** number: **None** – no correction, **Prandtl-Glauert** – correction method **Karmana-Tsiena** – correction method

Function	Description
Connect two grids	Click to run **PANUKL's** application feature which enables user to connect input grids, [4.1].
Correct neighbours	Click to run program function for correcting neighbours (Fig. 36) for current ***.dat** (grid & wake) file [4.2].

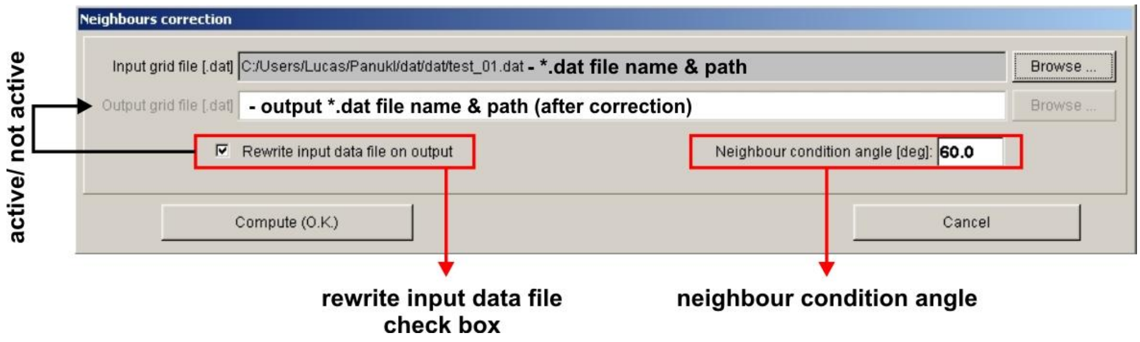


Fig. 36 – Configuration window – Correct Neighbours

3.1.5. XFOIL menu description

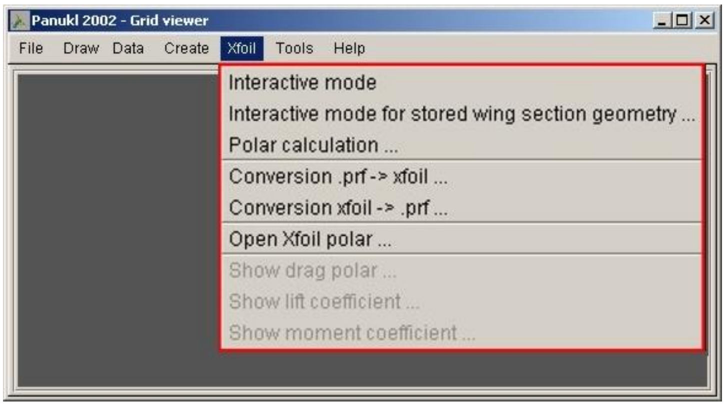


Fig. 37 – XFOIL menu

In **XFOIL** menu user can find functions which can help to analyze aerodynamic airfoils (used in wing geometry definition) using **XFOIL** program. To use it wisely user must have basic knowledge about **XFOIL** program.

Available options in XFOIL menu)

Function	Description
Interactive mode	Click to run external **XFOIL** program (must be installed [2.1]). Standard program window will appear (Fig. 38), **XFOIL** is ready to work.
Interactive mode for stored wing section geometry	Click to run **XFOIL** for specified ***.dat** file (Fig. 38). ***.dat** file contains coordinates of an airfoil to be analyzed.

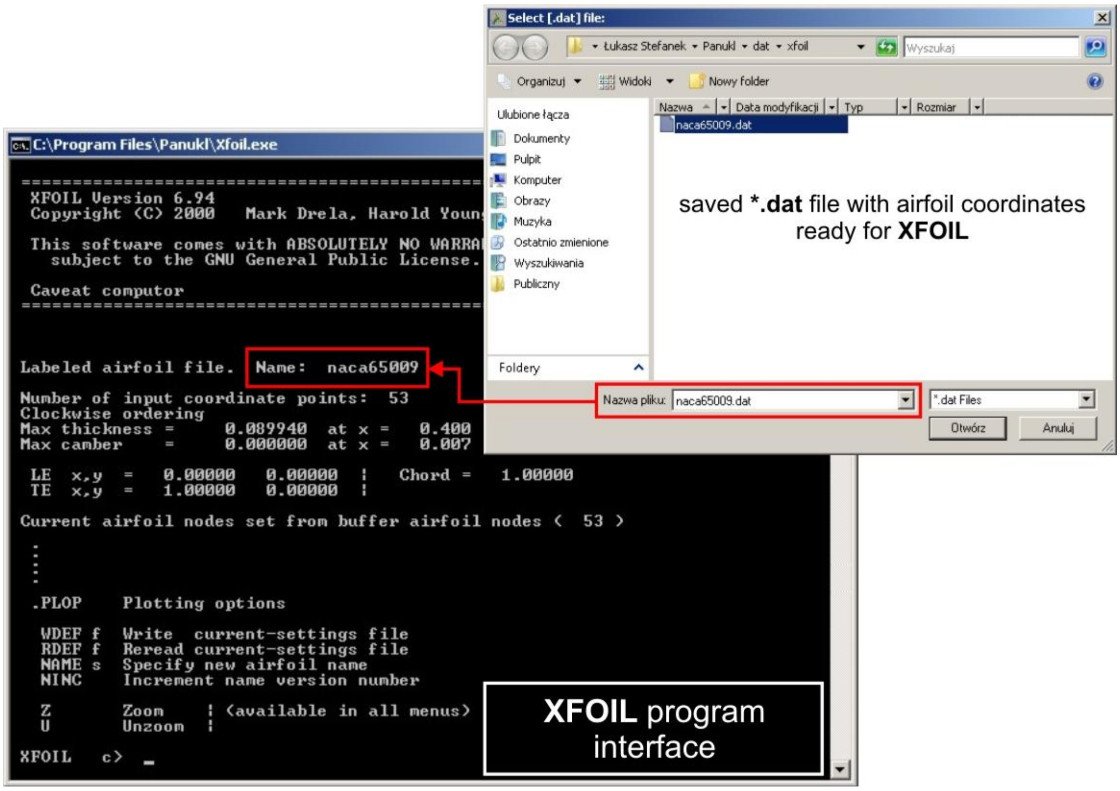


Fig. 38 – External XFOIL program window & *.dat file selection window

Function	Description
Polar calculation	Computing basic aerodynamic characteristics for an airfoil: **CL-lift**, **CD-drag**, **CM-moment**, versus **angle of attack** and **Reynolds** & **Mach** number. (airfoil geometry saved to ***.dat** file). Aerodynamic characteristics are computed in **XFOIL** external program. Results saved to ***.txt** file (Fig. 41).
Open XFOIL polar	Load file from disk with saved aerodynamic characteristics (Fig. 41). When *.txt file is loaded, functions below turn active:
Show Drag polar	Show drag **CD** polar, Fig. 42.
Show Lift coefficient	Show lift coefficient **CL** graph, Fig. 42.
Show Moment coefficient	Show moment coefficient **CM** graph, Fig. 42.

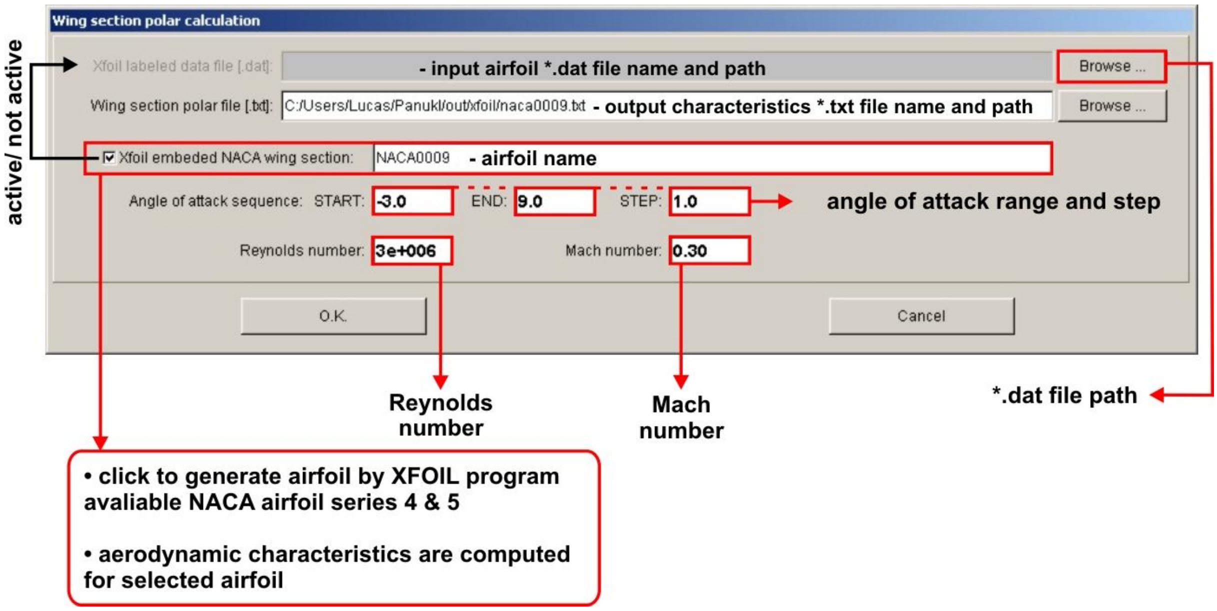


Fig. 39 – Polar calculations setup window

During **XFOIL** aerodynamic computations user must check if results converge.
Otherwise obtained results can have no physical sense. For more information go to **XFOIL** manual.

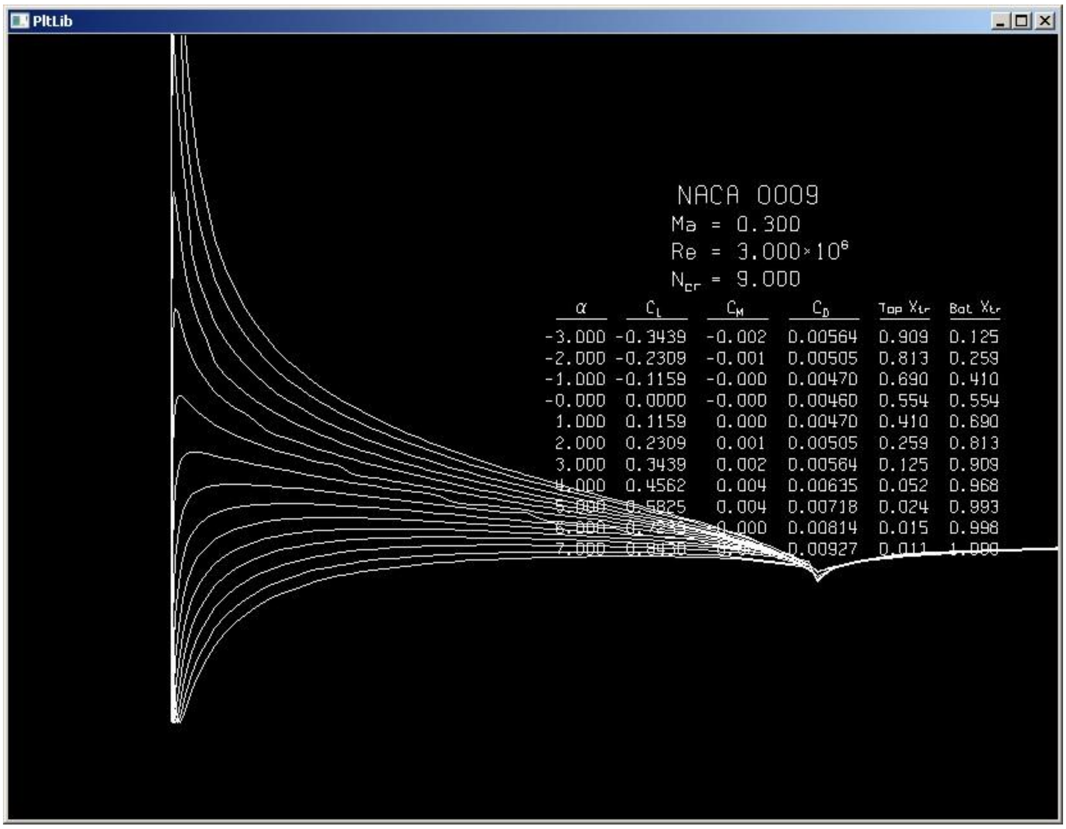


Fig. 40 – XFOIL window – aerodynamic coefficients computations for an airfoil

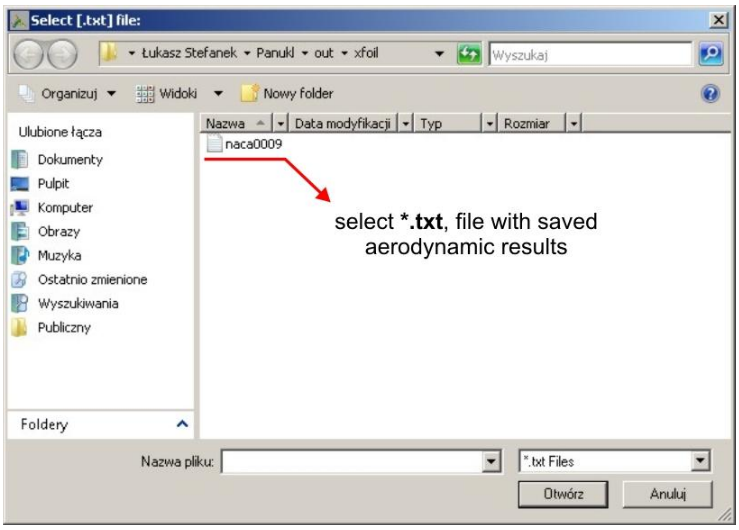


Fig. 41 – Selecting result file with aerodynamic characteristics for an airfoil

When ***.txt** file is loaded: **Show Drag polar**, **Show Lift coefficient**, **Show Moment coefficient** functions turn active.

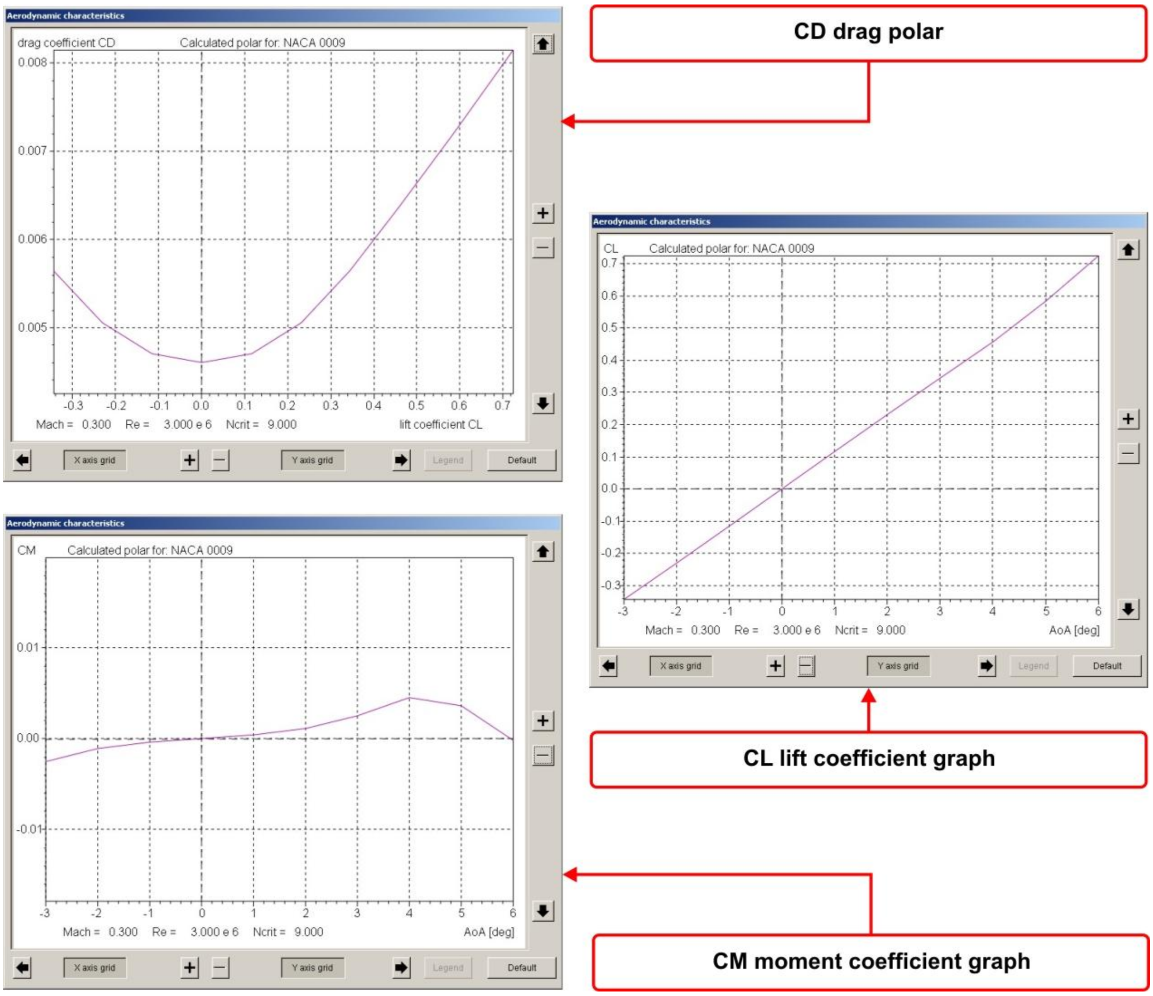


Fig. 42 – Example results CL, CD, CM versus Angle of Attack

Function	Description
Conversion *.prf to XFOIL	Click to convert ***.prf PANUKL** airfoil geometry file to ***.dat** airfoil **XFOIL** file (Fig. 43).
Conversion XFOIL to *.prf	Click to convert ***.dat XFOIL** airfoil file to ***.prf PANUKL** airfoil geometry file (Fig. 43).

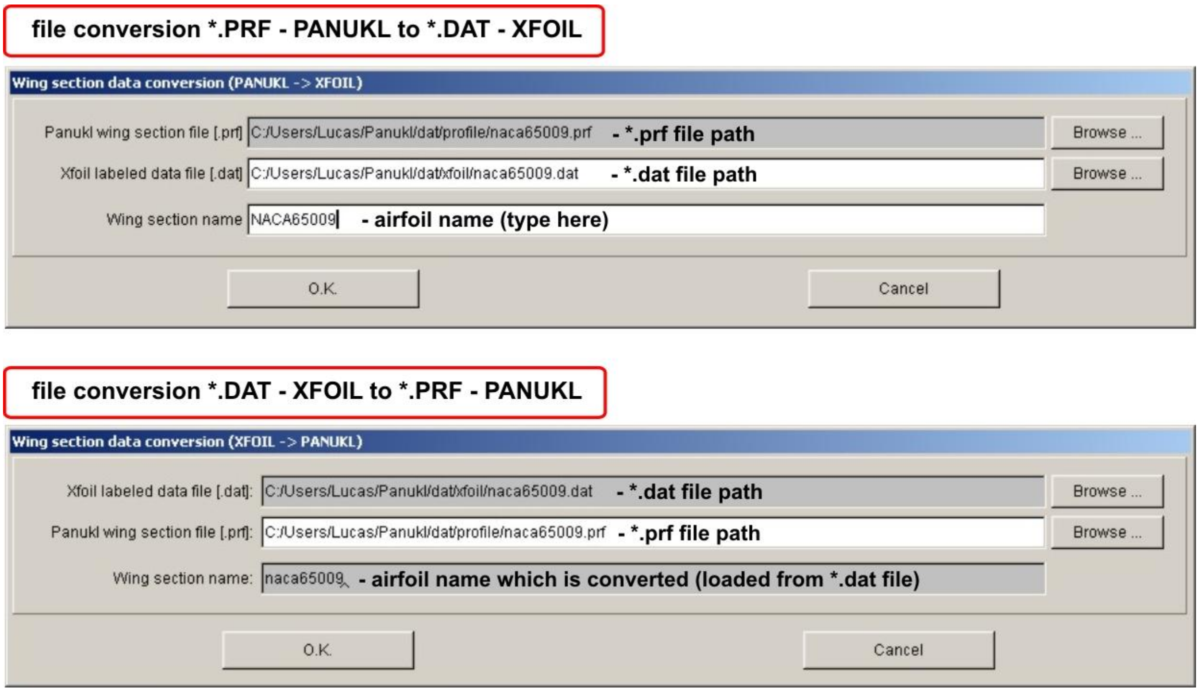


Fig. 43 – Airfoil file type conversion

3.1.6. TOOLS menu description

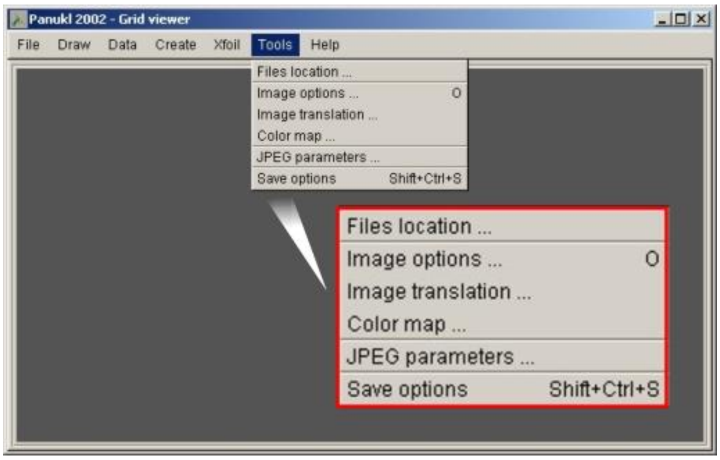


Fig. 44 – TOOLS menu

Available options in TOOLS menu)

Function	Description
Files location	Click to display window where user can select path for input and output files, (Fig. 45).

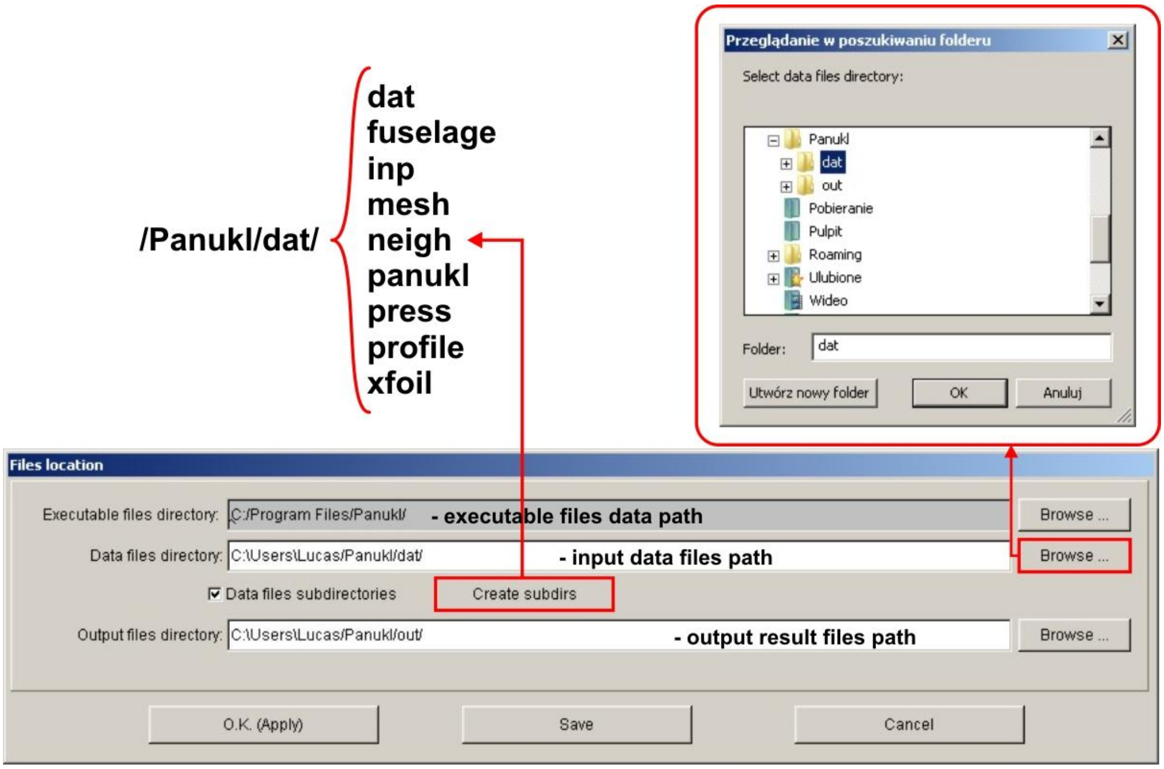


Fig. 45 – Files location selection window

Click **CREATE SUBDIRS** button to automatically create proper directory structure for **PANUKL**'s input and output data. . Click **SAVE** button to save options.

Function	Description
Image options [O]	Click to show window where user can change the way **PANUKL** displays geometry grid model in main **GUI** window, (Fig. 46).

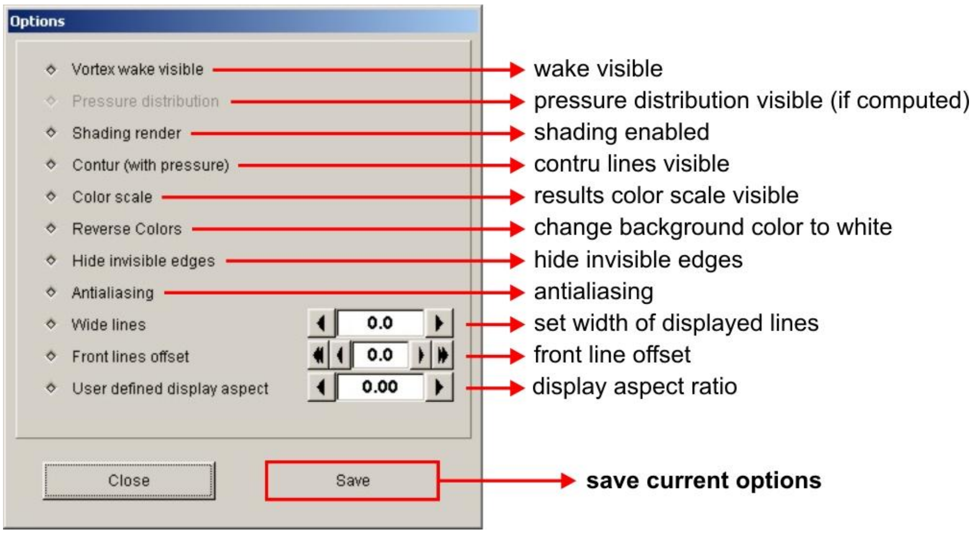


Fig. 46 – Grid model display options window

Function	Description
Image translation	Click to display window where user can adjust image translation options, (Fig. 47).

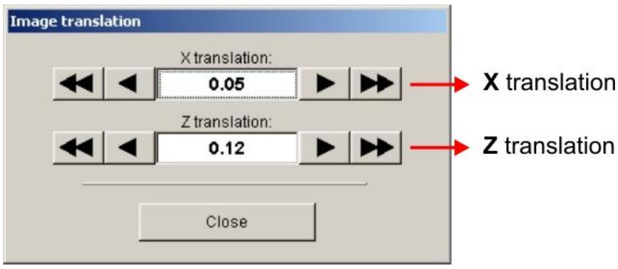


Fig. 47 – Image translation options

Function	Description
Color map	Click to display window where user can adjust options for graphic representation of results, (Fig. 48).

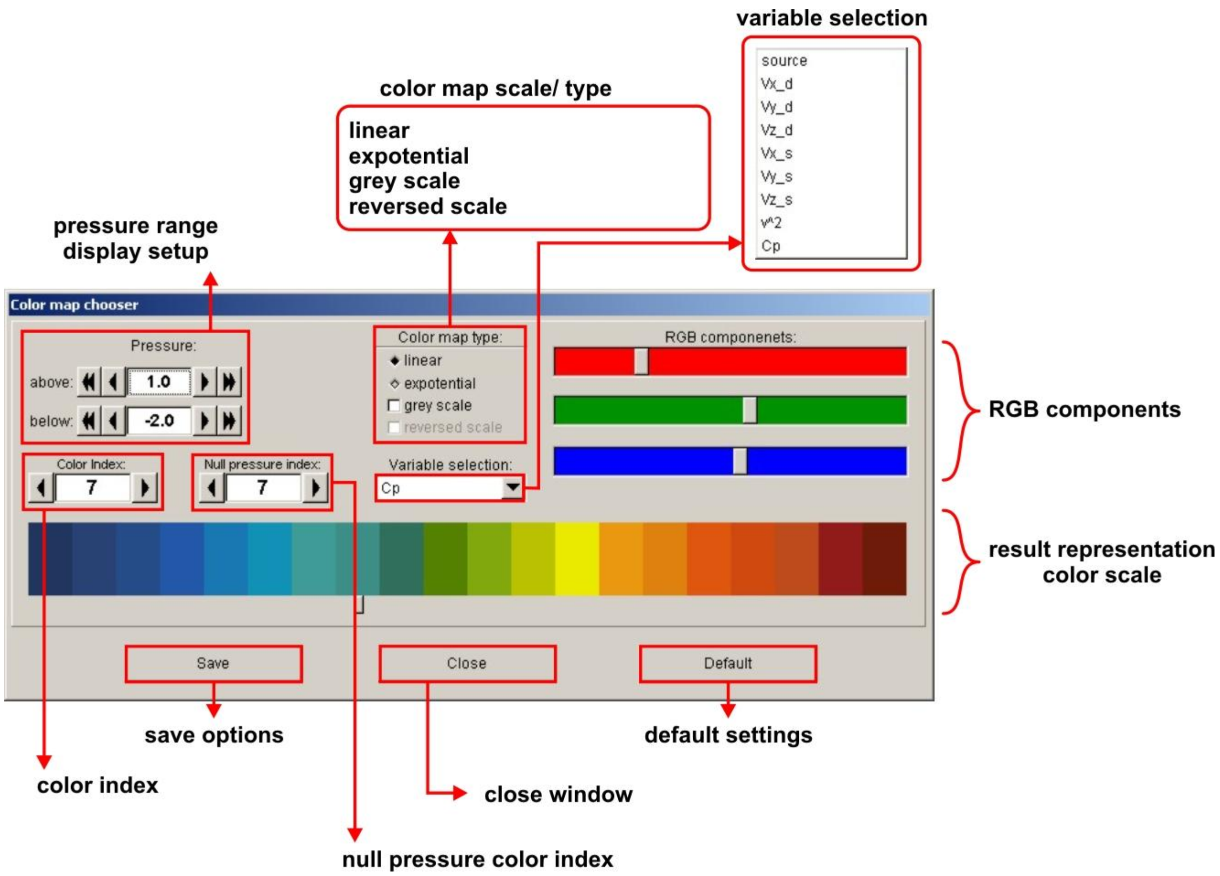


Fig. 48 – Graphic representation of results setup window

Function	Description
JPEG parameters	Click to display window where user can adjust **JPEG** picture capture quality and options, (Fig. 49).

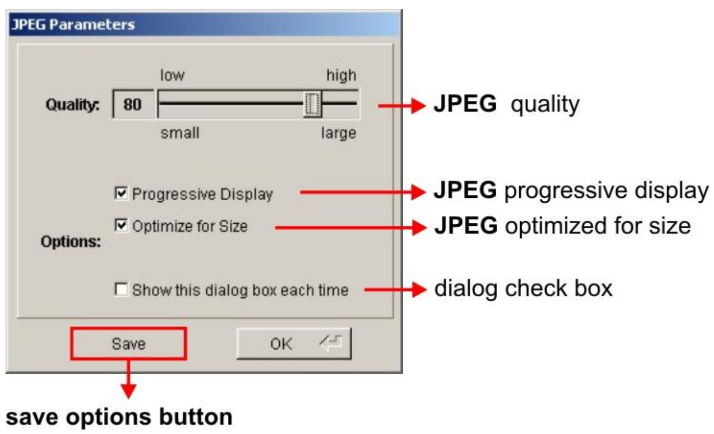


Fig. 49 – JPEG screen capture setup window

Function	Description
Save options [Shift+Ctrl+S]	Click to save current **JPEG** options.

3.1.7. HELP menu description

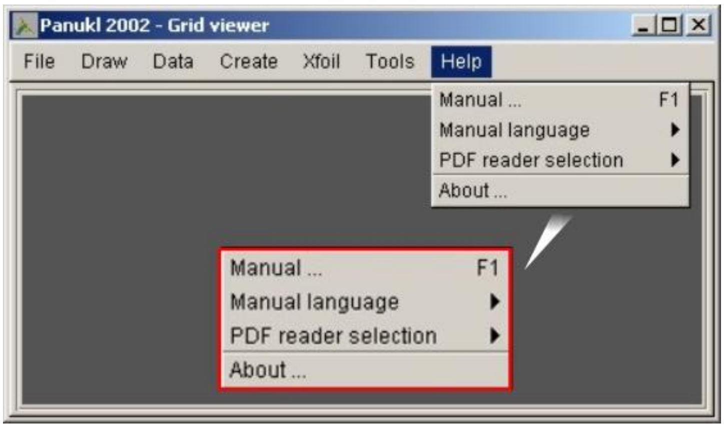


Fig. 50 – HELP menu

Available options in HELP menu)

Function	Description
Manual (F1)	Click to display the manual document for **PANUKL**.
Manual language	Manual language selection (available: **PL** & **ENG**)
PDF reader selection	PDF document reader application selection.
About	Click to display about **PANUKL** window, Fig. 51.

Fig. 51 – About information window

3.2. Computational procedure – diagram

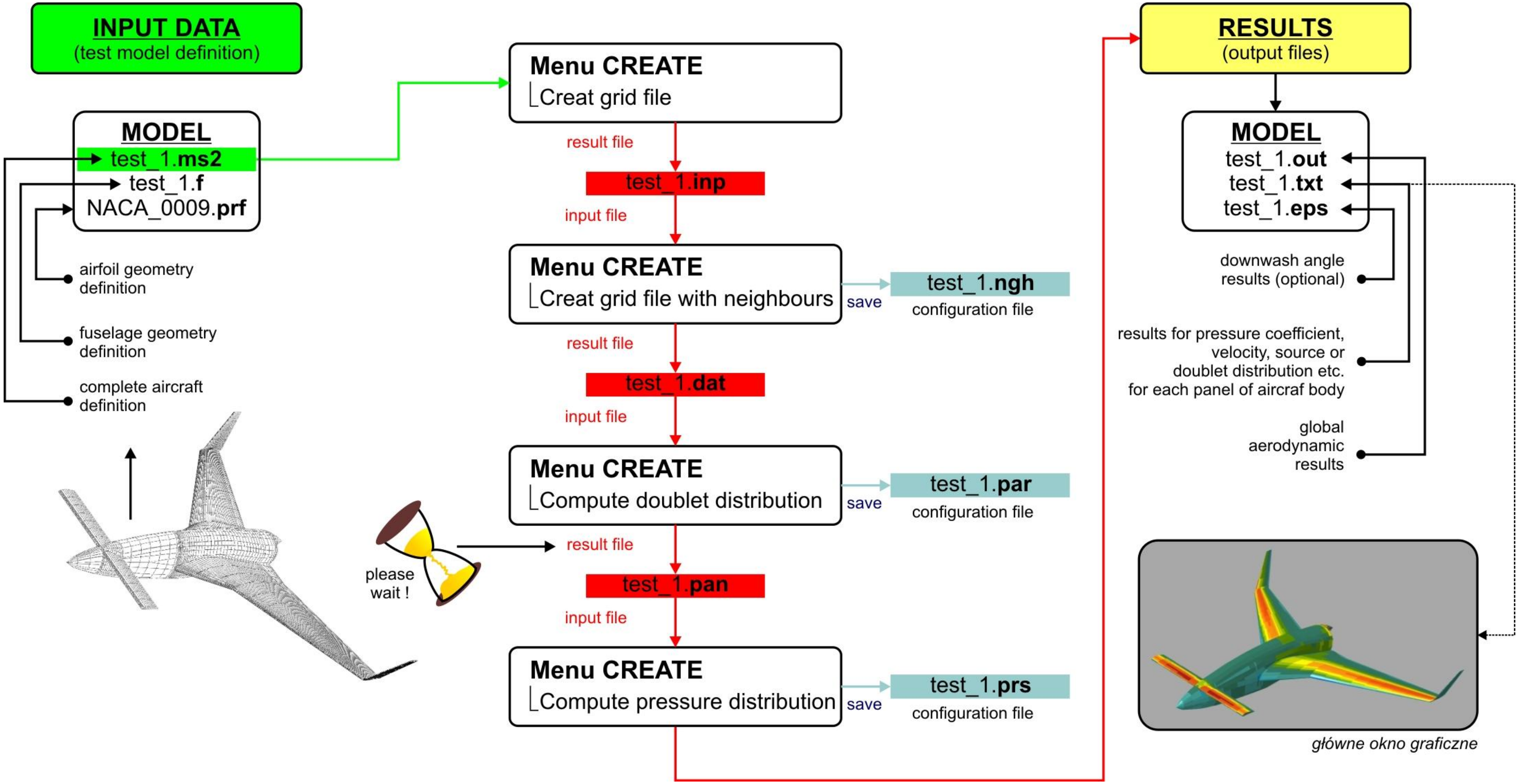


Fig. 52 – Option No. 1 – basic computational procedure

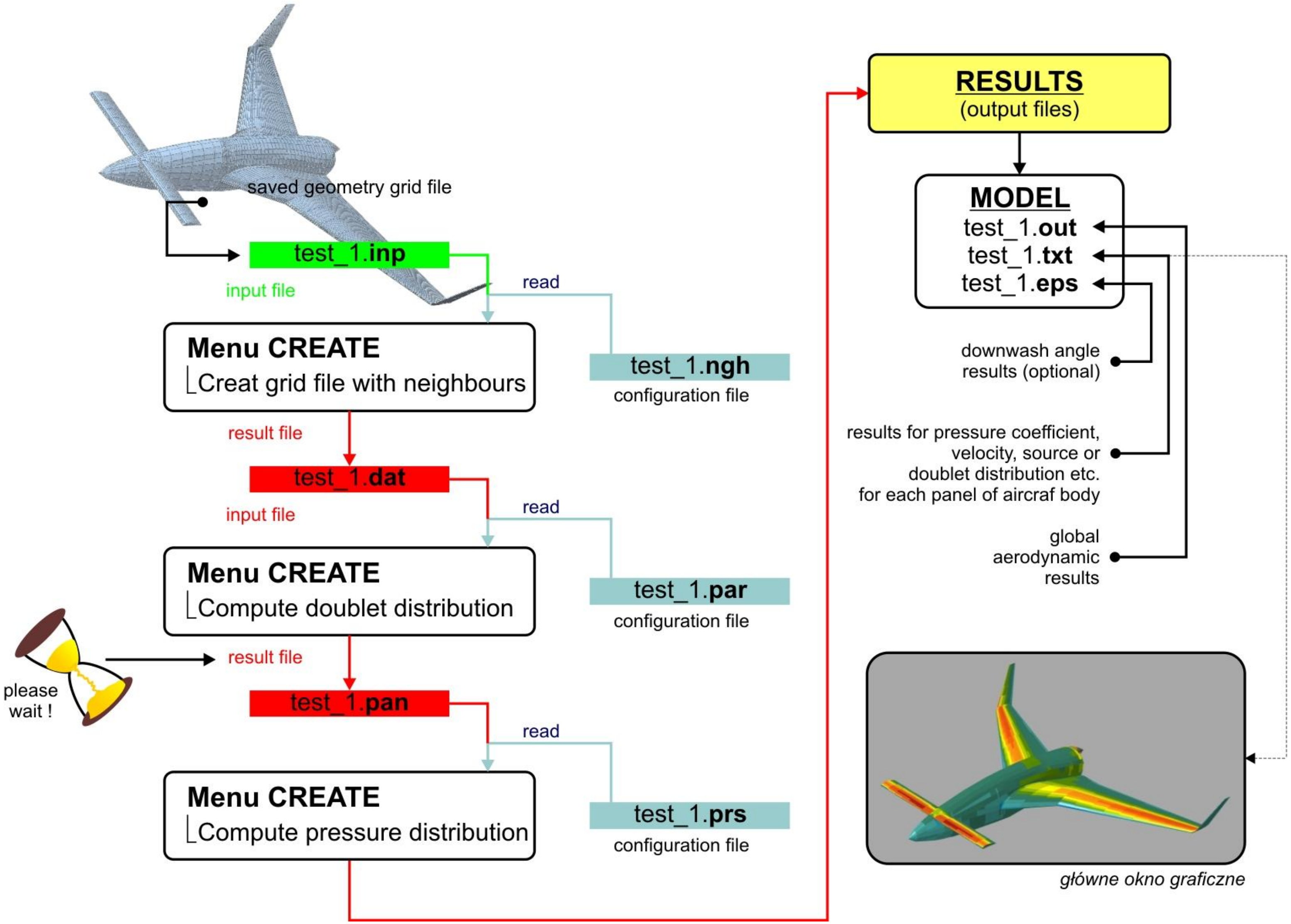


Fig. 53 – Option No. 2 – simplified computational procedure (user has got input grid and all necessary configuration files)

3.3. Data flow In PANUKL during the computation process

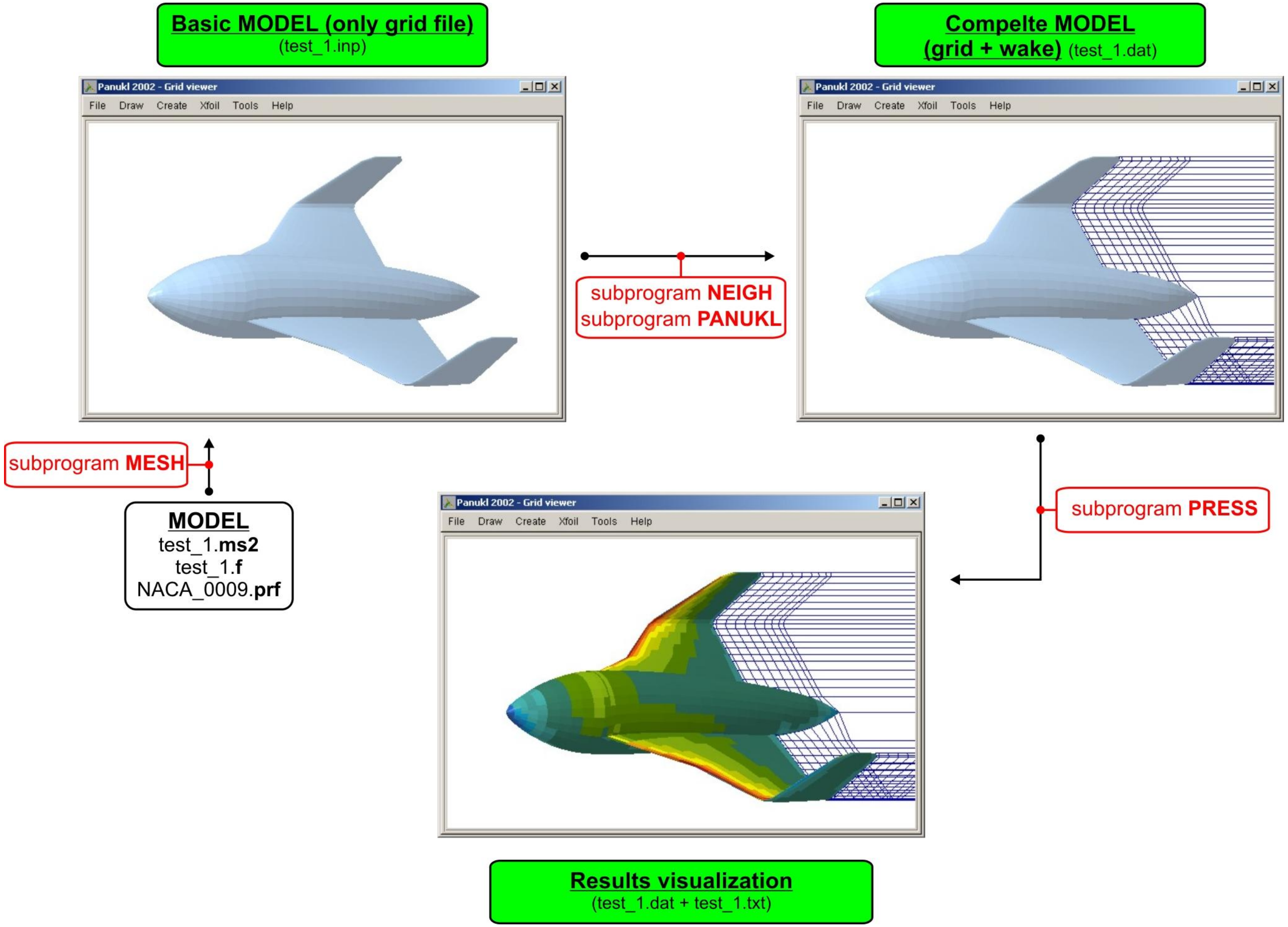


Fig. 54 – Data flow between MESH, NEIGH, PANUKL i PRESS subprograms

4. Suplement

4.1. How to connect grids - CONNECT TWO GRIDS option

From **CREATE** menu choose **CONNECT TWO GRIDS** option (Fig. 55). It will be used to connect saved on disc two grid files with wake**[***name*.**dat –** file**]**.

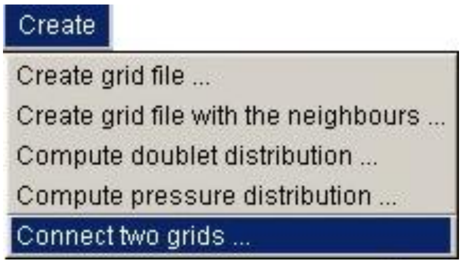


Fig. 55 – CREATE menu – CONNECT TWO GRIDS

This program function enables user to create complicated model grids assembled from more than one object. Additionally we can create non symmetrical model grids.

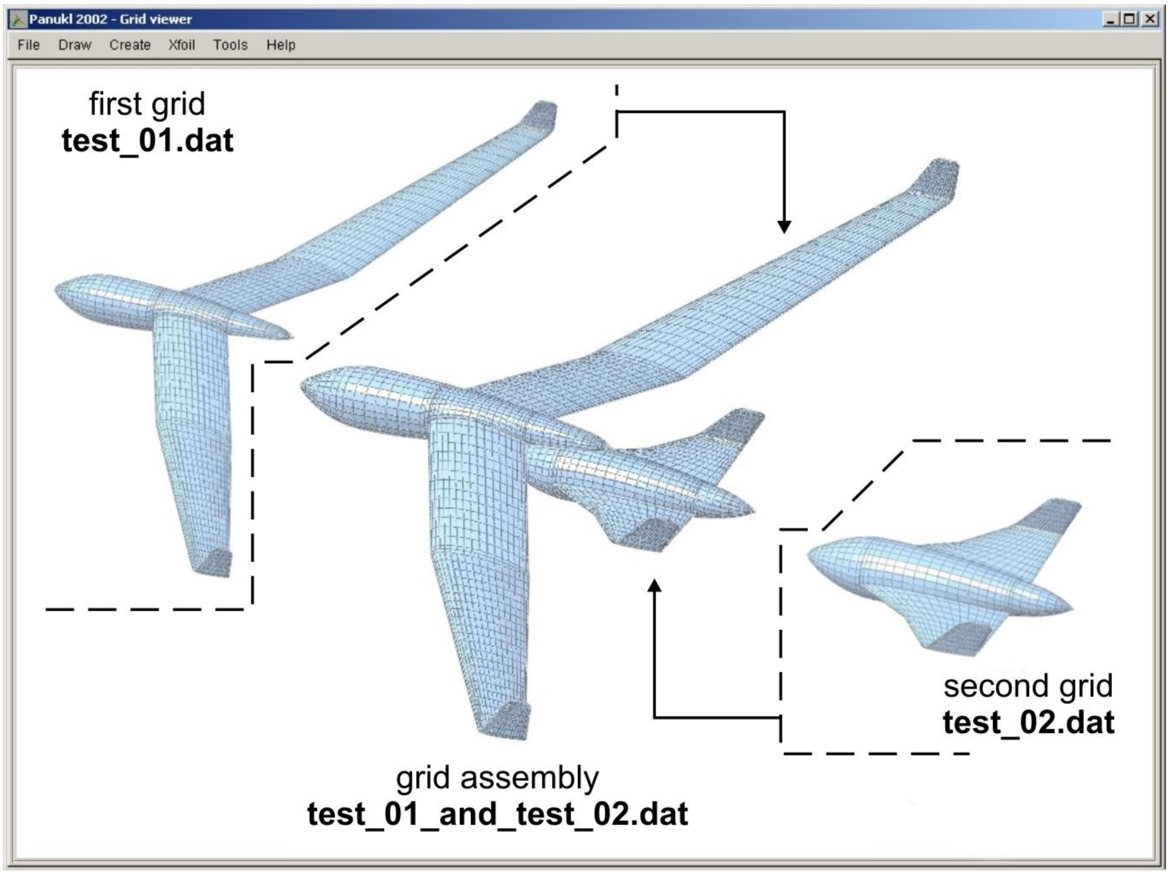


Fig. 56 – What does CONNECT TWO GRIDS function do ?

How it Works ?

Option 1 – <u>we do have</u> configuration ***.con** file, Fig. 57

Run **CONNECT TWO GRIDS** and select saved configuration file ***.con** – file contains all necessary information to create ***.dat** file (which will be an assembly of two existing grids). To open selected ***.con** file click **OPEN** button. Configuration window will appear (Fig. 58) where one can see saved ***.dat** file creation options. To generate ***.dat** file click **Save and Compute (ok)** button.

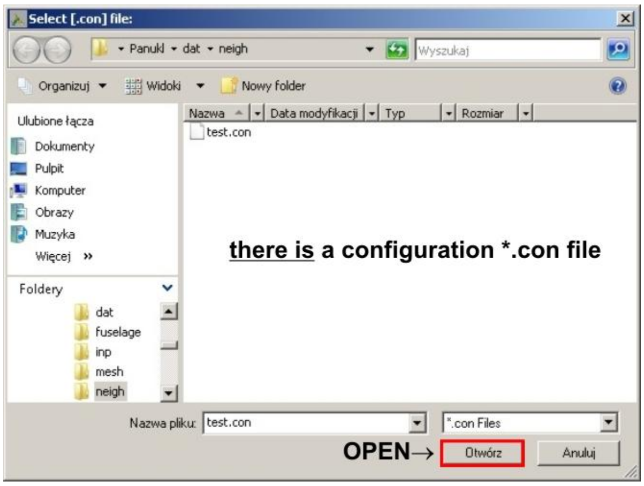


Fig. 57 – Open the connection configuration file *.con

Option 1 – <u>we do not have</u> configuration ***.con** file

Run **CONNECT TWO GRIDS** and click **CANCEL** button when prompted for saved configuration file ***.con**. The configuration window will appear (Fig. 59) where user can select options to create ***.dat** file. To save current ***.dat** creation options to ***.con** file click **Save [*.con] file as**, to create ***.dat** file click **Save and Compute (ok)** button.

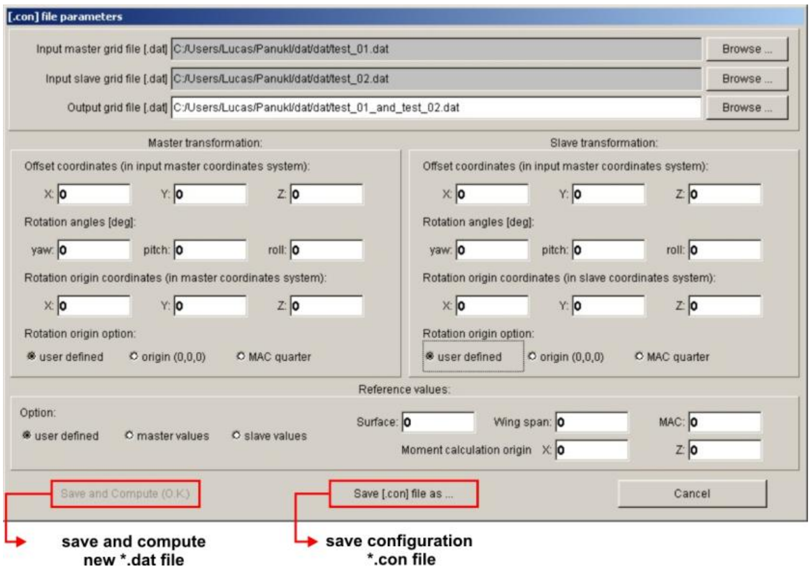


Fig. 58 – CONNECT TWO GRIDS setup window

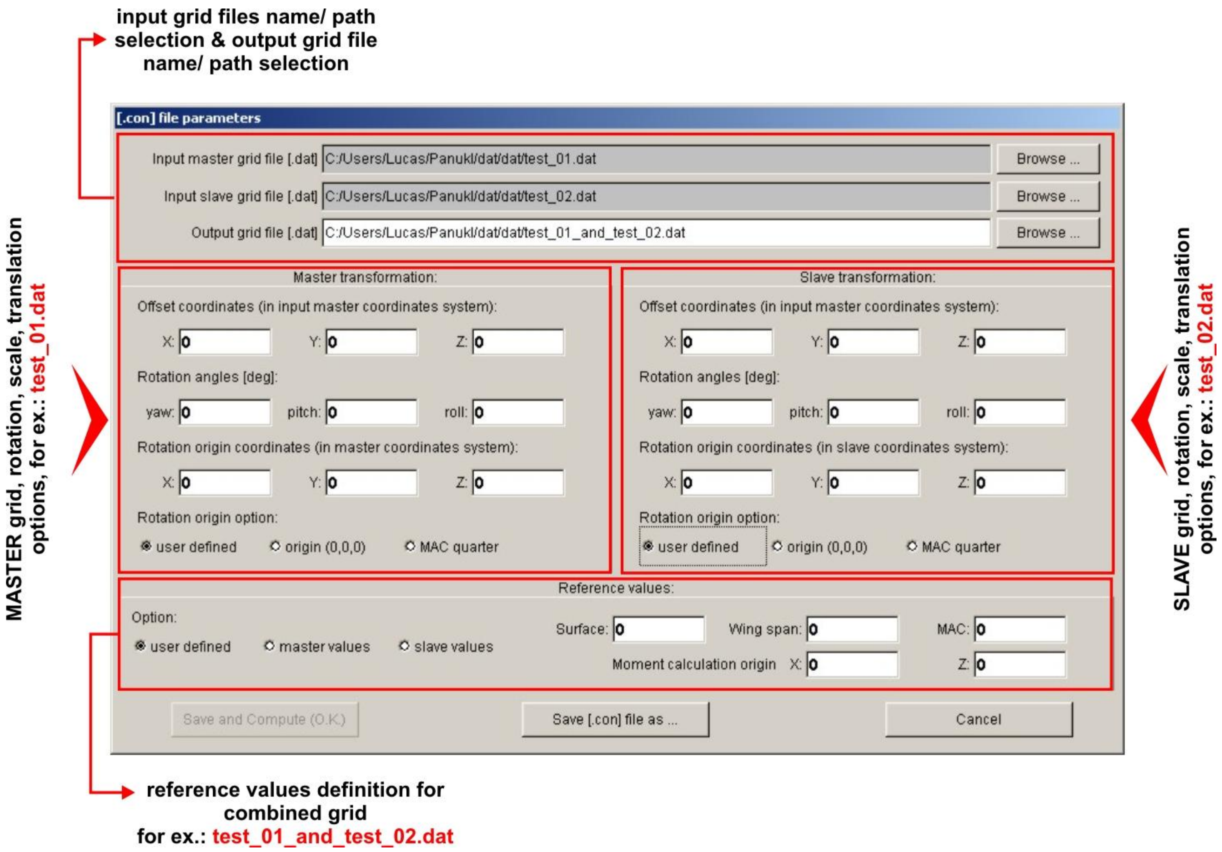


Fig. 59 – CONNECT TWO GRIDS main options

Function	Description
Offset coordinates (in input master coordinates system)	Offset coordinates (in input master coordinates system) – **X, Y, Z**
Rotation angles [deg]	To rotate component enter the necessary rotation angles **[deg]**. Rotation origin can be defined as: **User defined** – defined by user, **Origin (0,0,0)** – the origin of coordinate system for current object, **MAC quarter** – ¼ MAC for current aircraft.
Reference values	User must specify reference values for target output object: **User defined** – defined by user, **Master values** – reference values will be taken from master grid model **Slave values** – reference values will be taken from slave grid model

4.2. Creation of complex computational grids – CONNECT TWO GRIDS option

The program **PANUKL** in the newest version offers the possibility of creating complicated computational grids and asymmetrical grids. The whole procedure of creating is based on the function – **CONNECT TWO GRIDS** [4.1].

Below we can find an example procedure of creating complicated grid file with use of components – ***.dat** geometry files.

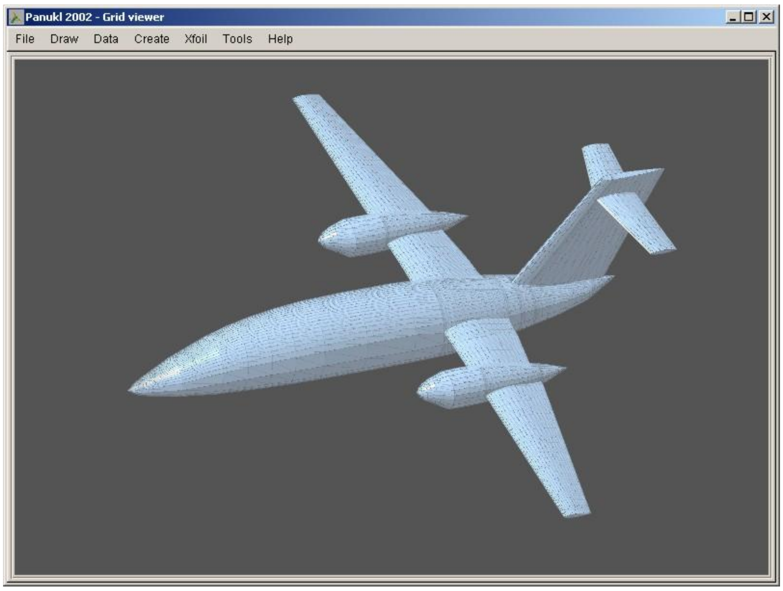


Fig. 60 – Complicated grid mesh example in PANUKL

How it is made ?

Step 1

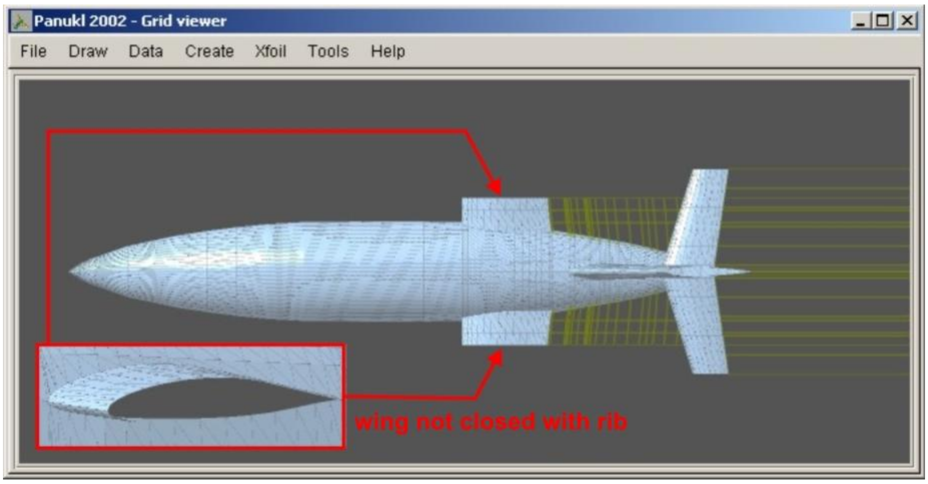


Fig. 61 – *.dat file – first part of the complicated grid file (wing part, fuselage, tail)

Create first geometry file „**01.dat**” . File will contain symmetrical fuselage with wing part and complete tail unit. The „**01.dat**” file must have vortex wake generated.

Important note: The wing part cannot be closed with rib because the next grid part will be connected to it. In aircraft ***.ms2** definition file choose: **0** – don't close wing with rib Fig. 62.

Step 2

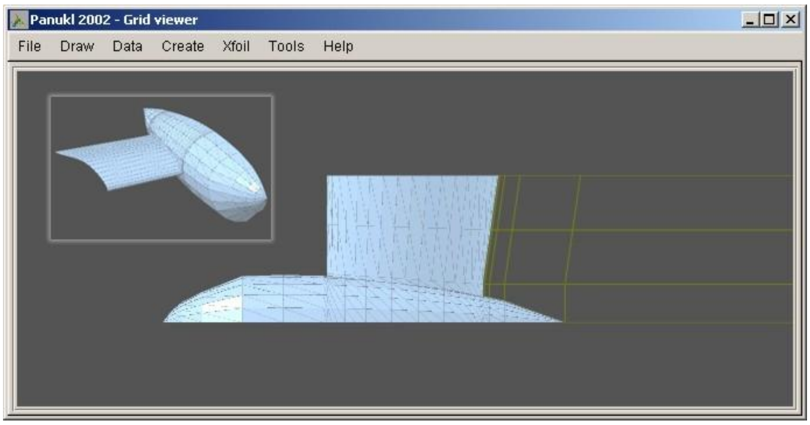


Fig. 62 – *.dat file – next part of the complicated grid file (wing & nacelle part – left side)

Next geometry „**02_L.dat**” file contains right nacelle part, part of wing (not closed with rib) and generated wake. Not symmetrical grids can be made by changing the proper flag in ***.ms2** file section. Nacelle is made similarly like fuselage.

Important note: Changing the proper flag in **MAIN FILE SCTION** in ***.ms2** file [1.3.1], we can easily create (not symmetrical) left or right side of model/ body.

Step 3

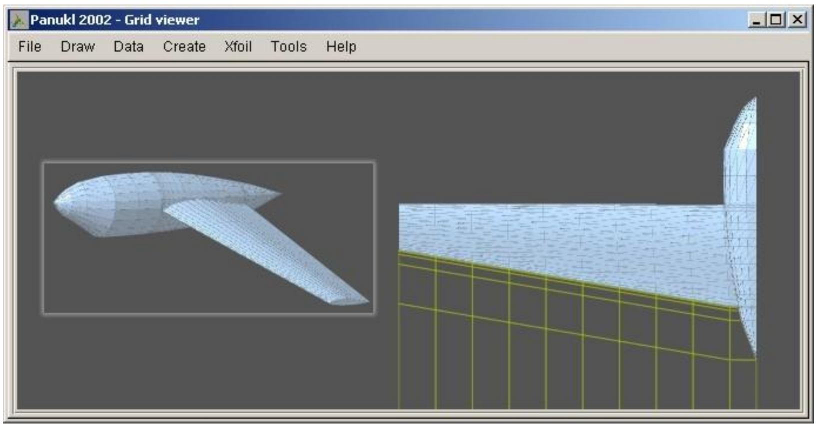


Fig. 63 – *.dat file – next part of the complicated grid file (wing end & nacelle part – right side)

Next geometry „**03_L.dat**” file contains left nacelle part and right wing ending. The „**03.dat**” file must have vortex wake generated.

Right side of the object is made similarly!

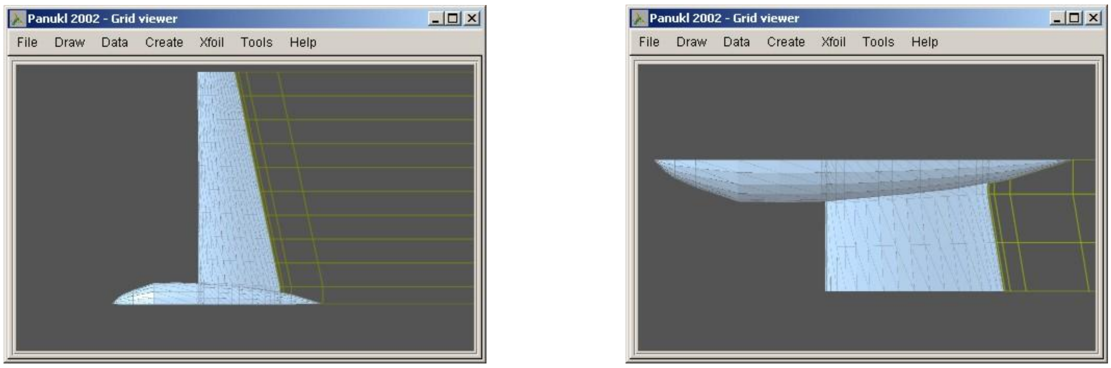


Fig. 64 – Complicated grid elements (right side)

Step 4

When we have all grid elements, we are ready to connect them into one complicated grid file. We will use **CONNECT TWO GRIDS** [4.1] function. To connect grids properly we need to know their exact position in global coordinate system.

This is an example grid connection procedure:

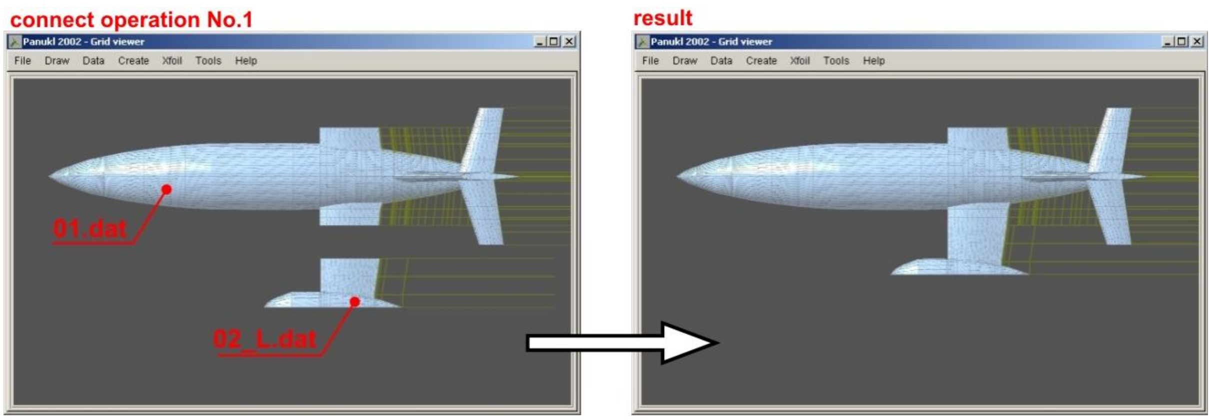


Fig. 65 – Connect operations number 1

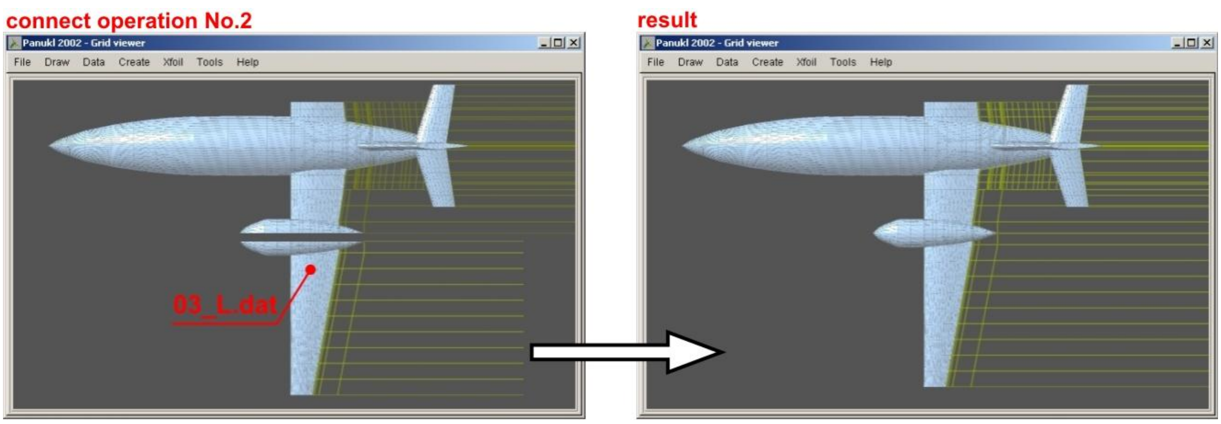


Fig. 66 – Connect operations number 2

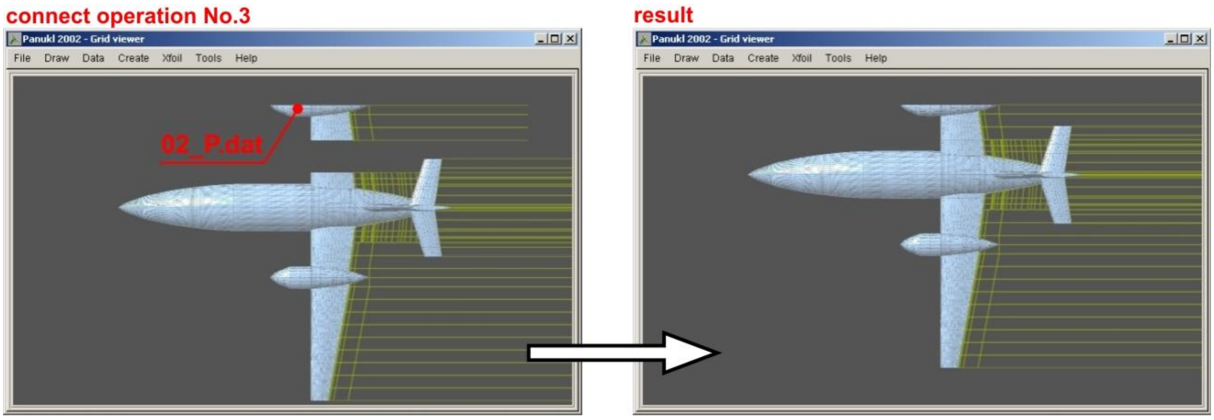


Fig. 67 – Connect operations number 3

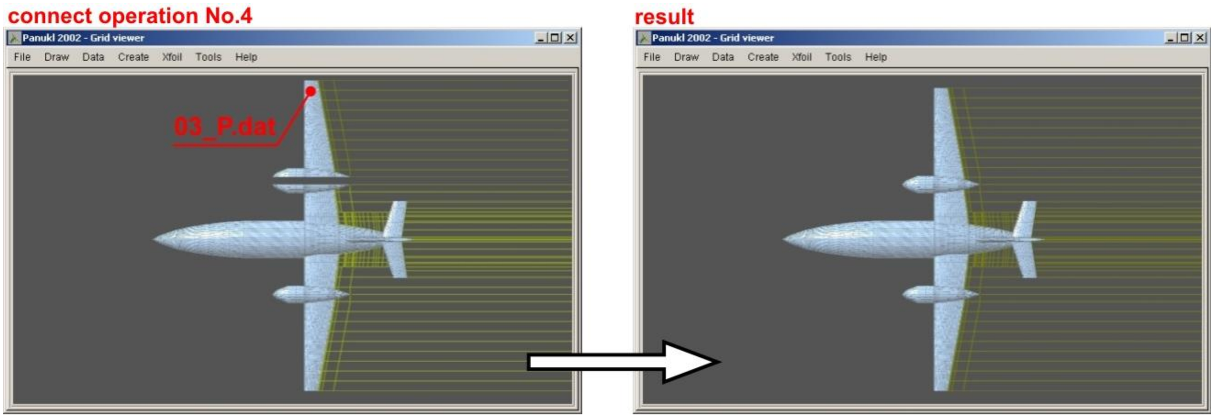


Fig. 68 – Connect operations number 4

Step 5

After the last connect operation we must check the new grid with – **CORRECT NEIGHBOURS** [3.1.4] function to avoid grid errors.

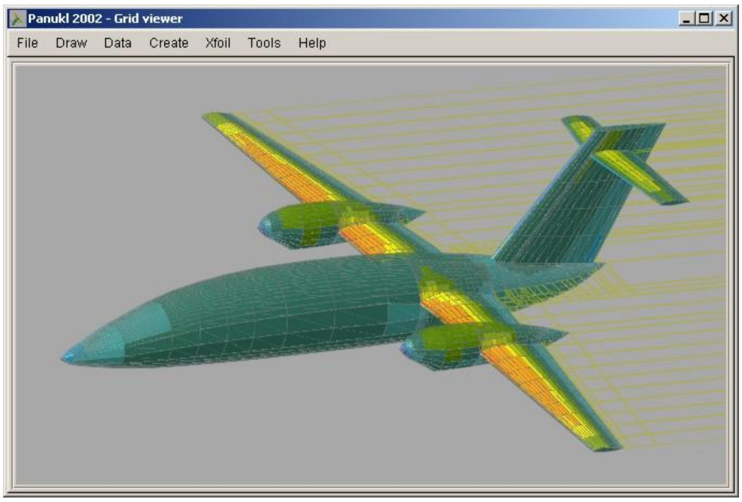


Fig. 69 – Generated complicated grid + example results

The complicated grid is ready to proceed with farther analysis.

Important notes:

What we should know before we will start to connect grids.

- Connected grid elements cannot be closed and must have the same division, e.g., two wing grid parts must have in connection area the same airfoil and its chord division and chord length.
- Connected grid elements must have one common plane.
- With this connection procedure we can create difficult grids that are not symmetrical.
- To make not symmetrical fuselage or nacelle we must remember to have the same number of sections/ frames for both sides. Section/ frame location must be also the same for both fuselage or nacelle parts.
- Fuselage or nacelle section/ frame number and location must be the same for left and right part.
- Sometimes we can simplify the geometry to connect two grids, e.g. wing body intersection area.

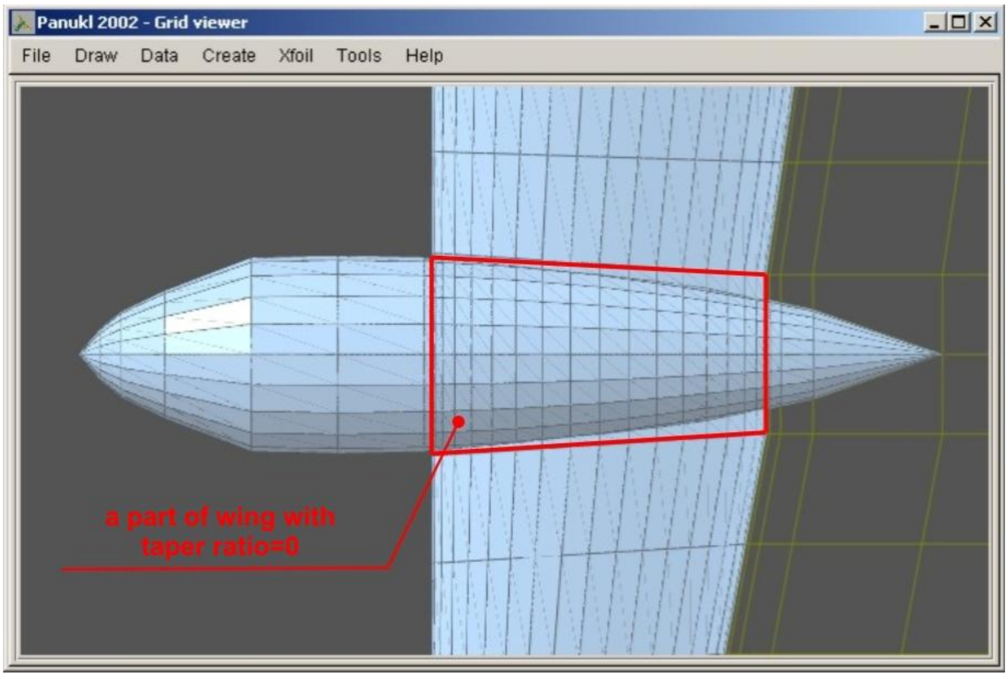


Fig. 70 – Grid trick – easy to create wing body intersection area

4.3. FUSELAGE DATA – external subprogram description

Program **FUSELAGE DATA**

FUSELAGE DATA program was made to help with creating fuselage/ nacelle geometry files **[*name*.f]** for **PANUKL** application**.** With **FUSELAGE DATA** we can create:

- Fuselage geometry file **[*name*.f]** from **txt [*name*.w]** files. Each file contains frame outline definition points described by free coordinates.
- Fuselage geometry file **[*name*.f]** from **txt [*name*.txt] UNIGRAPHICS** files. Each **UNIGRAPHICS** file contains information about point describing one frame**.**

Important note: Each fuselage or nacelle section/ frame needs one **[*name*.w]** or **[*name*.txt]** file with points describing its geometry. The section/frame point coordinates order in file is free.

- Modification of present fuselage files **[*name*.f]**

Main **FUSELAGE DATA application window**

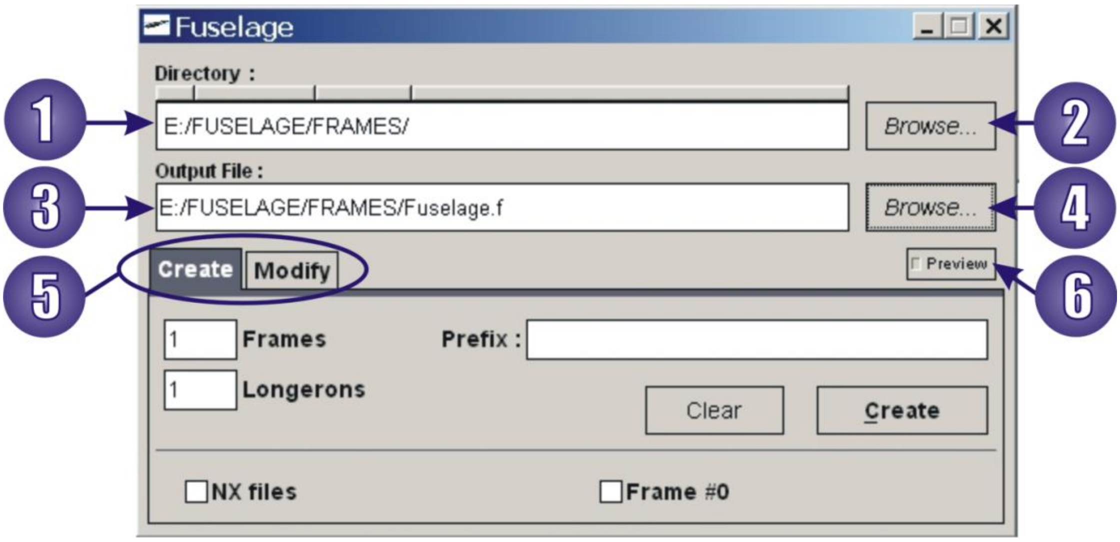


Fig. 71 – Main FUSELAGE application window

Main program options:

1. Folder path for input fuselage geometry frame files: **[***name***.w]** or **[***name***.txt]** files.
2. **Browse** button, for input files.
3. Output fuselage geometry file name and path **[***name***.f]**.
4. **Browse** button, for output file.
5. **CREATE** or **MODIFY** program modes.
6. Fuselage geometry file – **Preview** button.

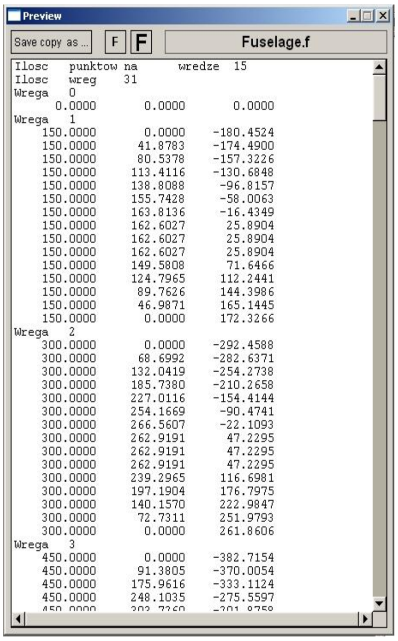


Fig. 72 – Fuselage geometry file – preview window

Create mode in **FUSELAGE DATA**

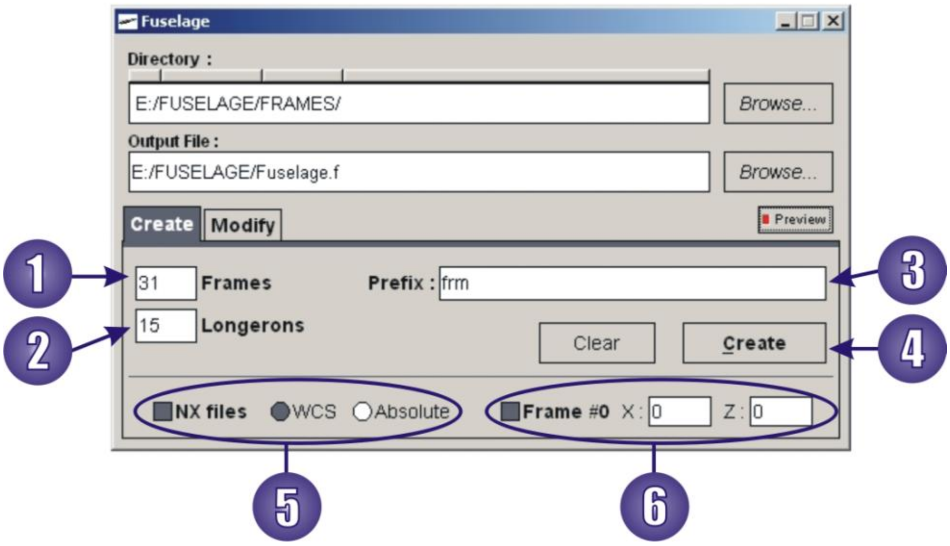


Fig. 73 – Create mode in FUSELAGE program

In this mode we can create fuselage geometry file for PANUKL:

(1) Max. frame number defining fuselage geometry (frame No.0 counts in).

(2) Number of stringers on a single frame.

(3) Frame name syntax must be: **[**<*prefix*>_<*No.*>**.w]** or [<*prefix*>_<*No.*>**.txt]**. Where ***<No.>*** is the number of next frame (numeration starts from frame No.0). Frame No.0 point coordinates can be defined in (6).

To use **[***name***.txt] UNIGRAPHICS** files in section (5) select **NX FILES** check box and specify coordinate system for defining frame points: **WCS** or **ABSOLUTE**.

(4) **Create** fuselage geometry file button**.**

Modification mode in **FUSELAGE DATA**

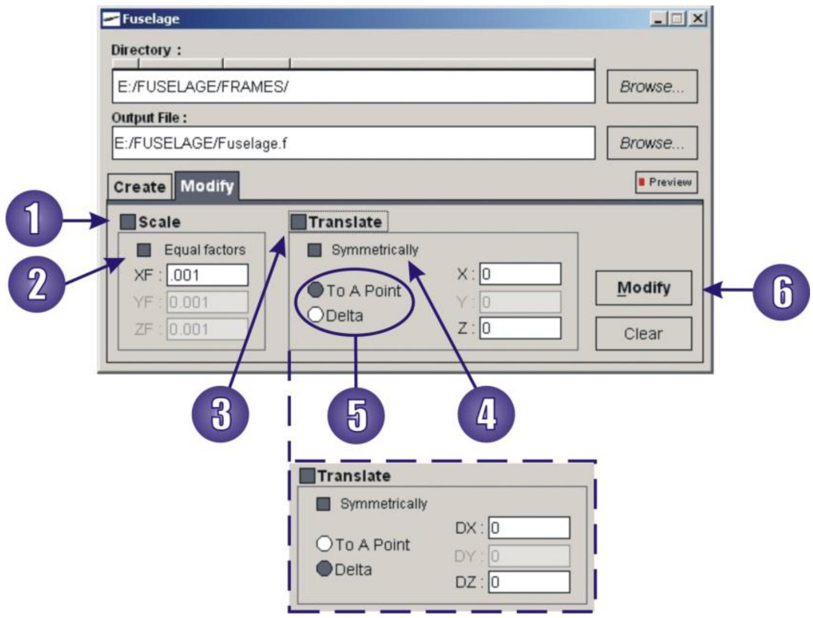


Fig. 74 – Modify mode in FUSELAGE program

In this mode we can modify existing fuselage geometry file. Things we can modify:

(1) Scale fuselage. **EQUAL** scale **FACTORS** (2) can be used in every scaling direction (**X**, **Y**, **Z**).

(3) Translate fuselage. Translate to a specified **X**, **Y**, **Z** point location
or delta translate: **DX**, **DY**, **DZ** coordinate increments must be defined.

(4) Translate fuselage only in **XZ** plane.

(5) Translation type selection.

(6) Click **Modify** button to accept changes and modify fuselage file.

4.4. How to export geometry from UG NX system to PANUKL software

Export geometry procedure from **UNIGAPHICS NX** to **PANUKL** is only an example what we can do with available engineering tools. The export procedure can be used for complicated grids. It enables to export in detail 3D model geometry to **PANUKL** grid file. It is not optimized procedure it takes long but it works well.

Below in document you will find a short description of procedure for preparing fuselage geometry for **PANUKL** in **UNIGRAPHICS NX** software. The example can show you the way to create your own **PANUKL's** geometry files: **[***name***.f]**, **[***name***.ms2]** & **[***name***.prf]** with a help of any available CAD tools.

Export geometry procedure assumptions:

- we do have 3D model geometry file and we can open it in **UNIGRAPHICS NX**;
- we have basic knowledge about modeling in **UNIGRAPHICS NX** system.

Example procedure

1. Load 3D model geometry file. Translate geometry if necessary to **(0,0,0)** point of global coordinate system (nose of the aircraft is in **(0,0,0)** point, **X** axis to back, **Y** axis to the right wing, **Z** axis up).
2. Trim right half of model with **ZX** plane. We will not use it.

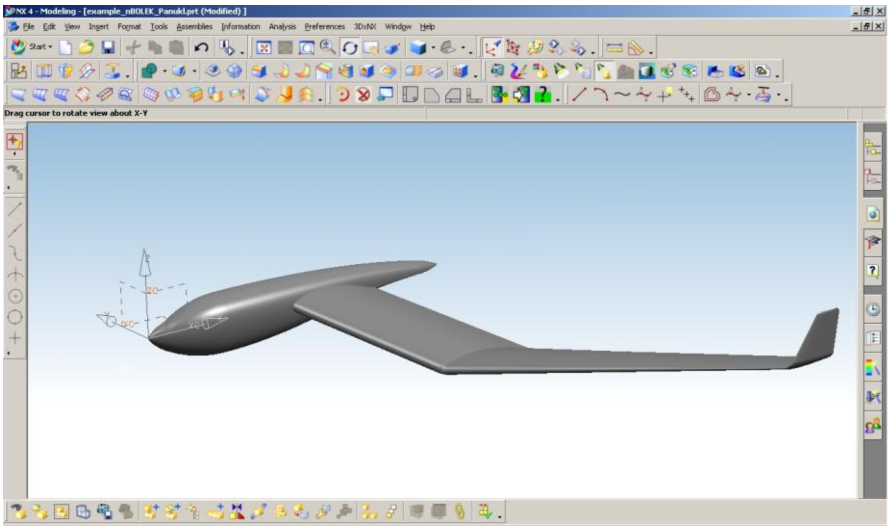


Fig. 75 – Left half of model in UG NX4, aircraft nose located in (0,0,0) of global coordinate system

3 Trim fuselage with scaled wing . (Scale wing geometry for about 30 percent). Hide wing we will not use it now.

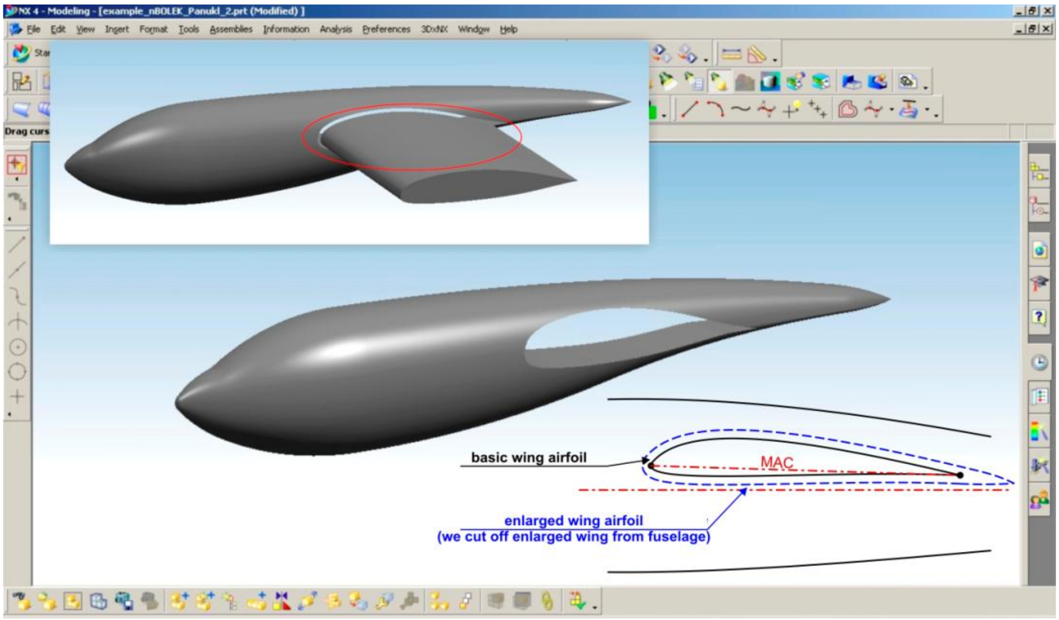


Fig. 76 – Trimmed fuselage geometry in UG NX4

4 Insert **ZY**, **DATUM PLANES** where we will have fuselage frames. It is important to remember that too many fuselage frames make geometry hard to export to **PANUKL**. If you have time and fast CPU insert as many as you want ☺.

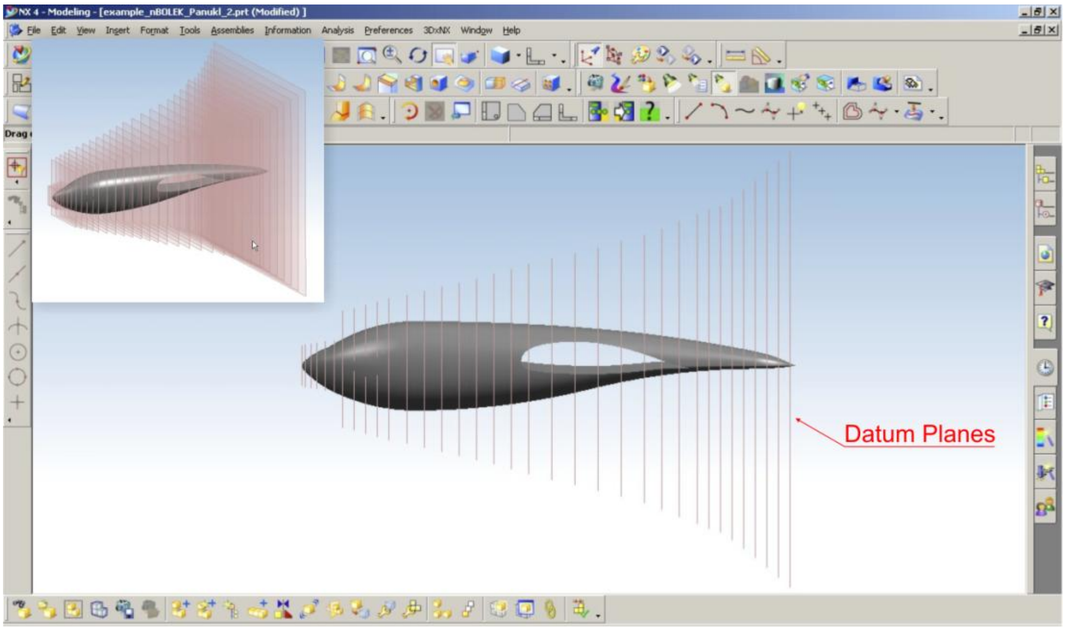


Fig. 77 – Fuselage half with DATUM PLANES in UG NX4

Remember: don't insert **DATUM PLANES** where wing trailing & leading edge intersect fuselage.

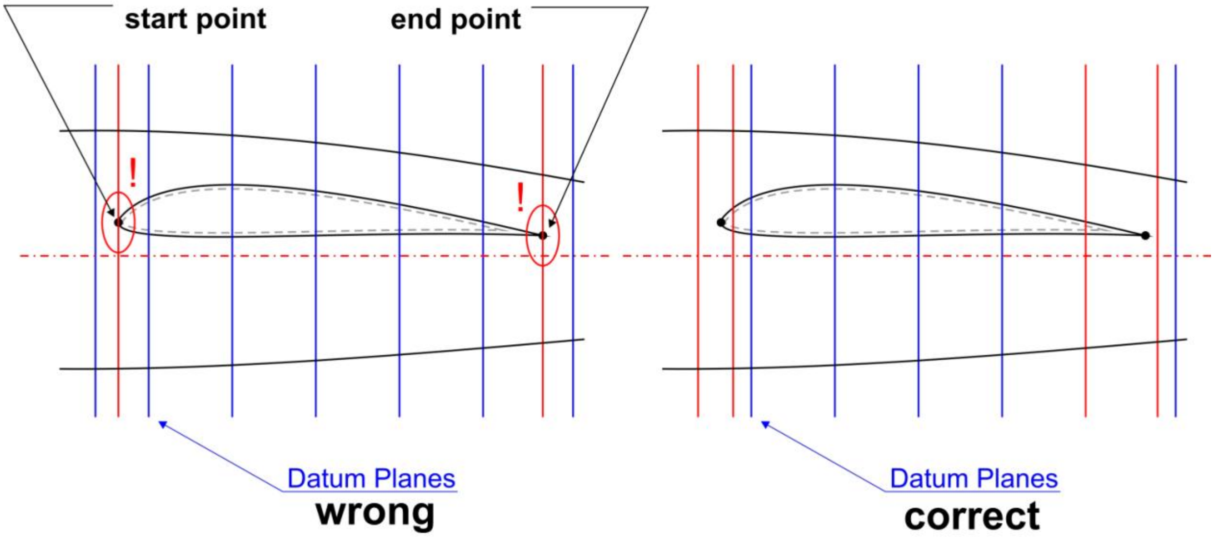


Fig. 78 – Important note about inserting DATUM PLANES for frames, UG NX4

5 Use **INTERSECTION CURVE** (**UG NX4)** function to create intersection curves where we have inserted **DATUMS**. Additionally add curves where the model was trimmed with wing.
We have made our fuselage frames.

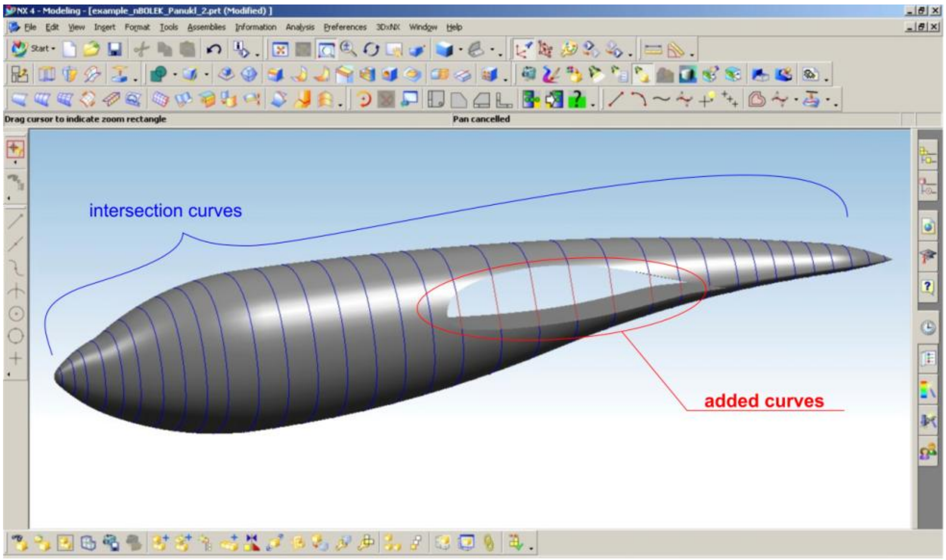


Fig. 79 – Fuselage intersection curves – fuselage frames, UG NX4

6 Now we need to create points on our frames – use **POINT SET/POINTS ON CURVE** function.

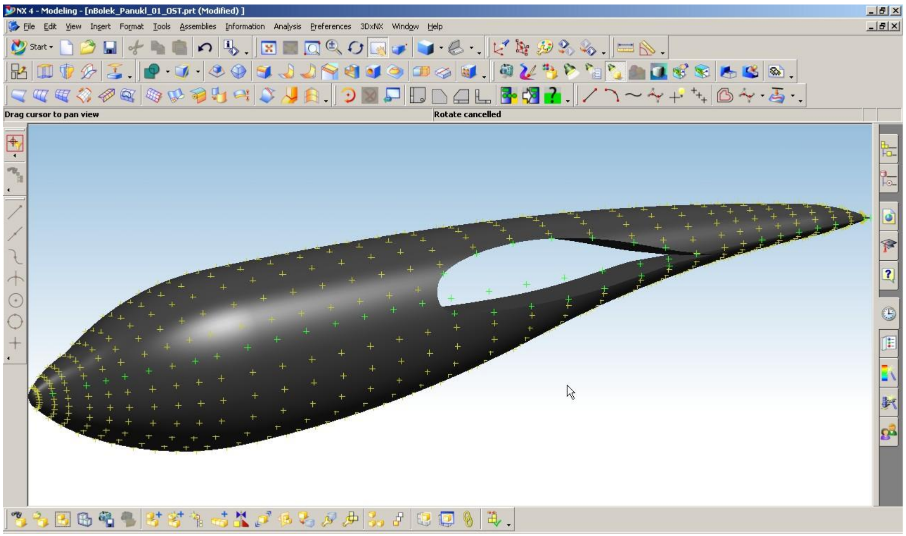


Fig. 80 – Model half with points placed on frames, UG NX4

- Connected points on frames will create fuselage stringers. If you want to have a very accurate grid model in **PANUKL** – create a large number of points.
- Number of points on a single frame must be the same for every fuselage frame. The only exception is fuselage start and end point.

Remember: always think about number of grid panels - if you create more frames more frame points than the grid in **PANUKL** will be more complicated and the computations will last longer.

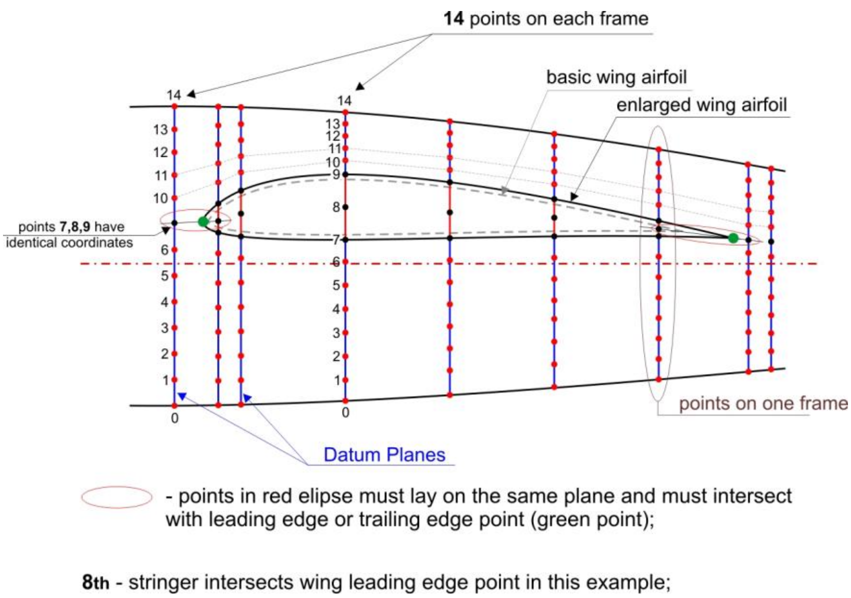


Fig. 81 – Frame point distribution near area where wing intersect fuselage

In some fuselage areas you will have to adjust the position of points on frame. There are also locations where you will have to insert more than one frame point in the same place (see Fig. 81). Well positioned points on frames will create smooth shaped fuselage stringers.

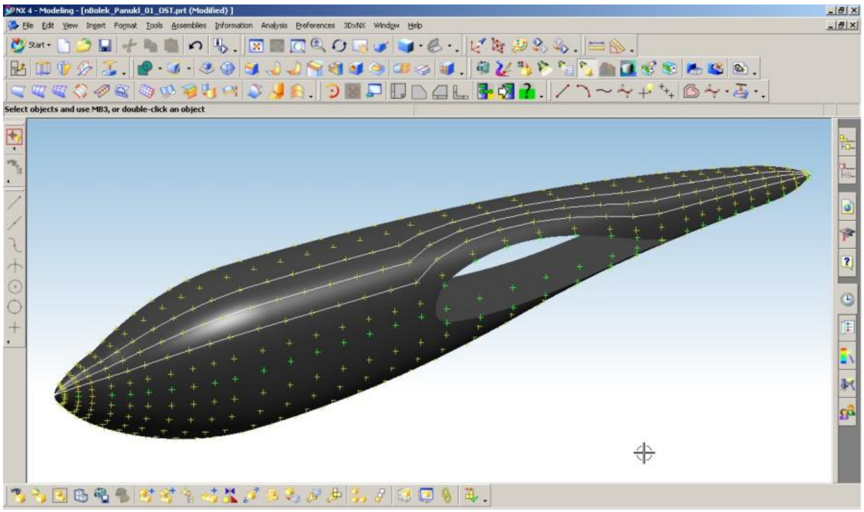


Fig. 82 – Fuselage points connected into fuselage stringers, UG NX4

7 Now we are ready to export all points from each frame to fuselage frame files. **INFORMATION/ OBJECT/ POINT**.

Remember: to set proper names to consecutive frames, e.g.: **w_1.txt** – frame No.1. than **w_2.txt** – frame No.2 etc. It will help to avoid mistakes.

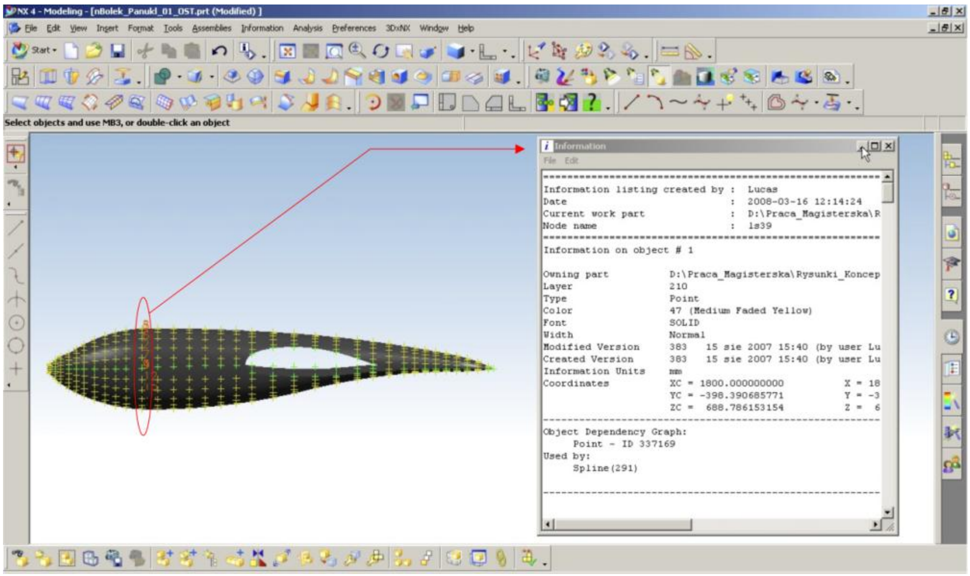


Fig. 83 – Export fuselage frame points from UG NX4

8 Now when we have all fuselage frame files we can create complete fuselage geometry file**[***name***.f]** (4.3).

Important notes:

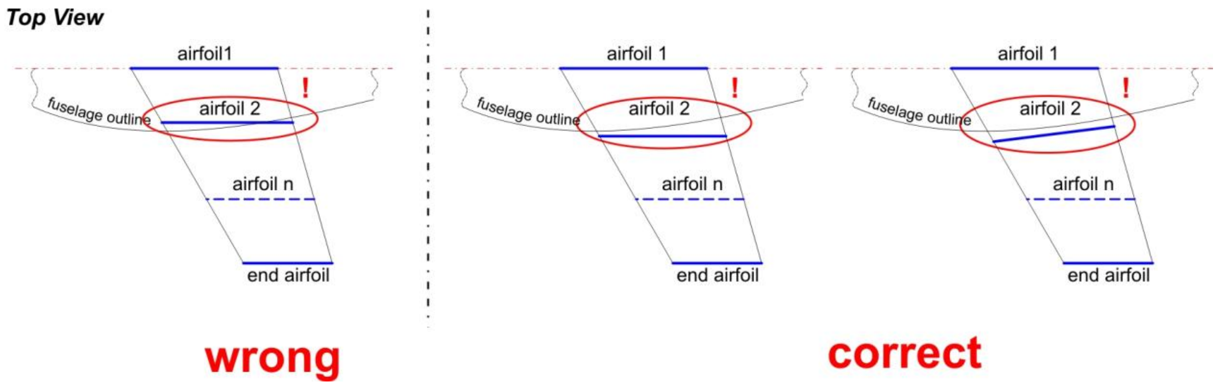


Fig. 84 – Definition of two first airfoils of a wing in PANUKL

Below one can find the results of export geometry procedure:

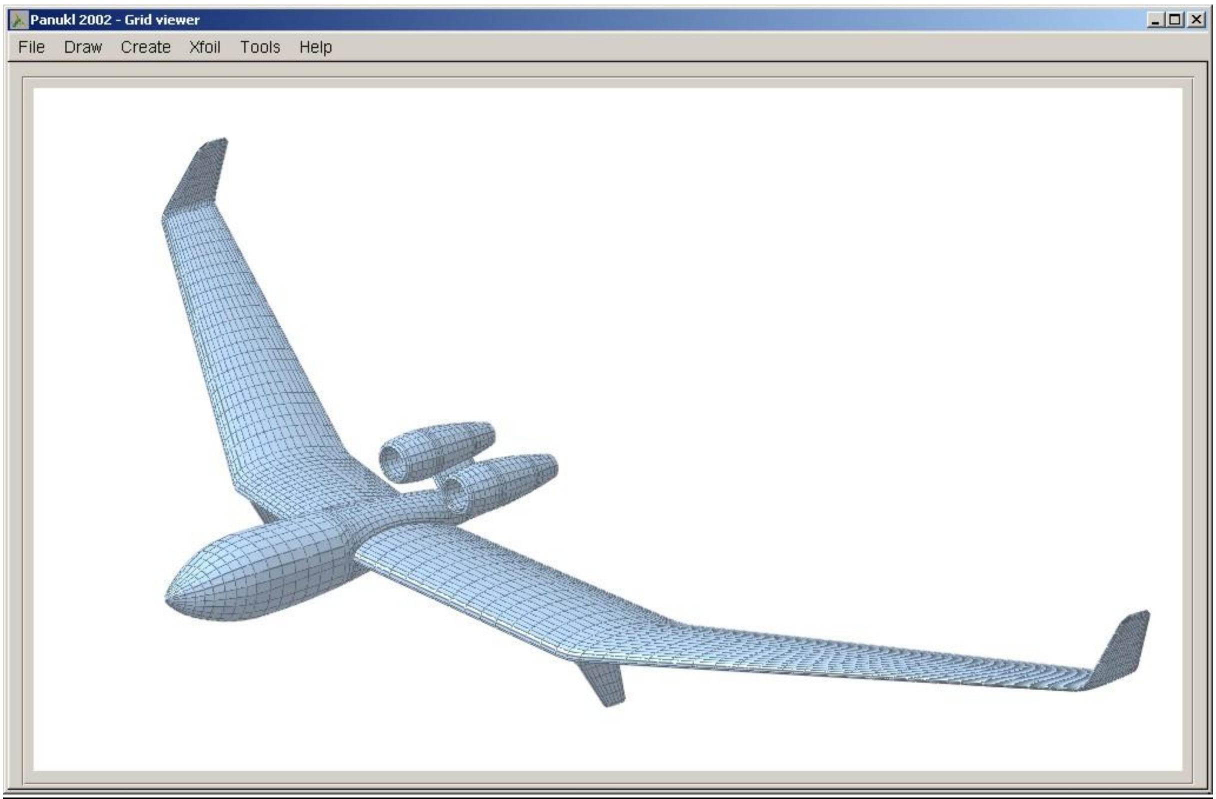


Fig. 85 – Models prepared with the use of the introduced procedure of the export of geometry for PANUKL

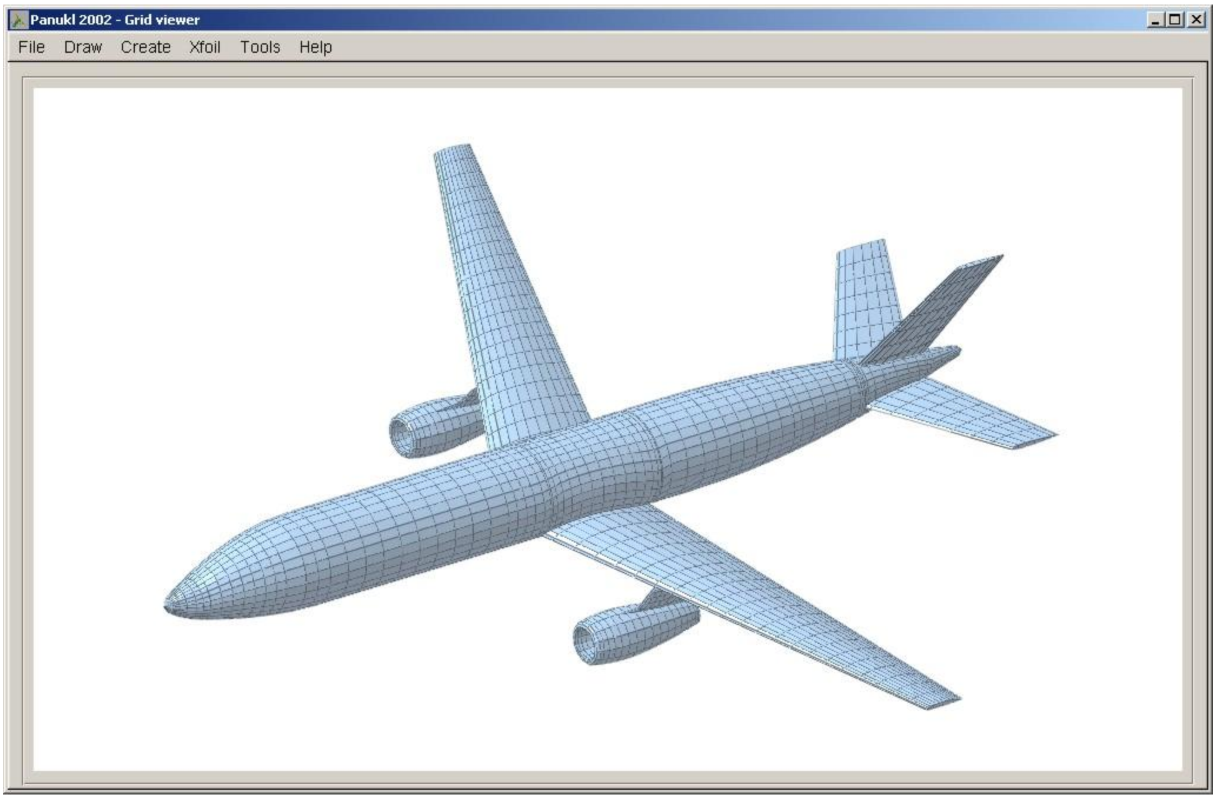


Fig. 86 – Models prepared with the use of the introduced procedure of the export of geometry for PANUKL

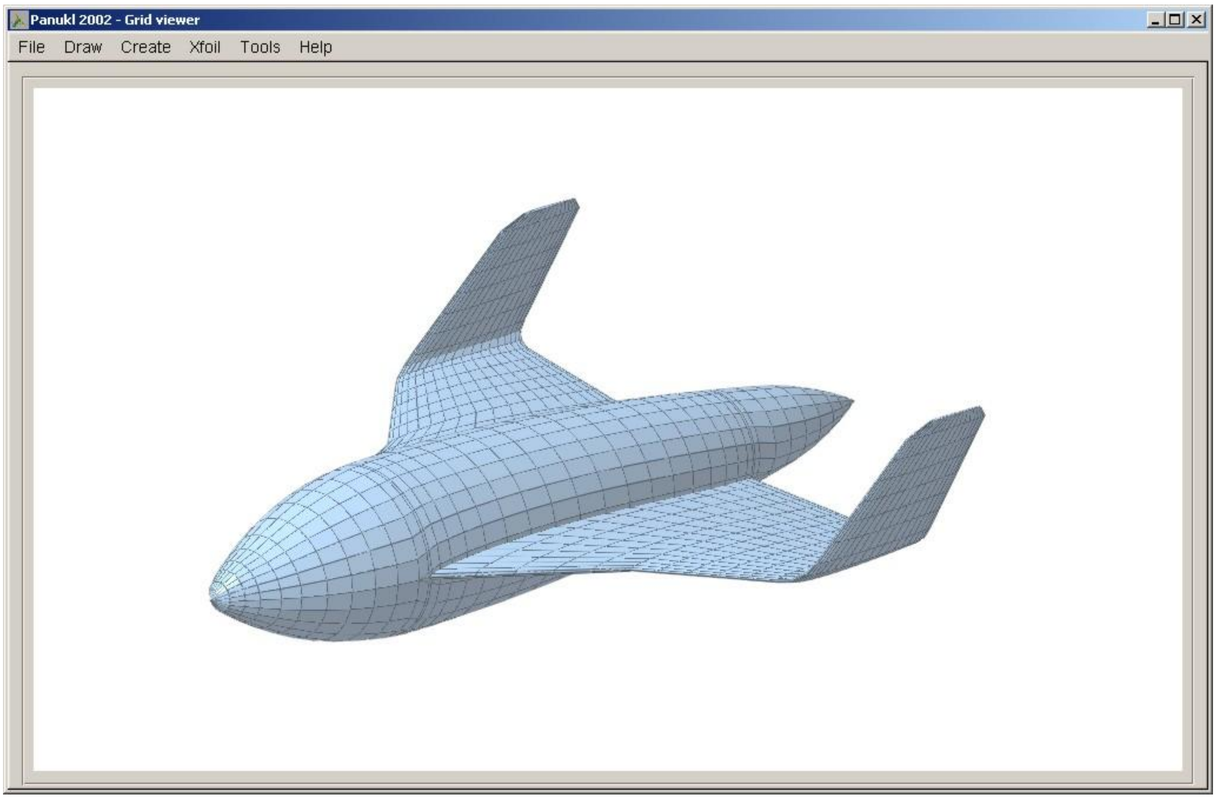


Fig. 87 – Models prepared with the use of the introduced procedure of the export of geometry for PANUKL

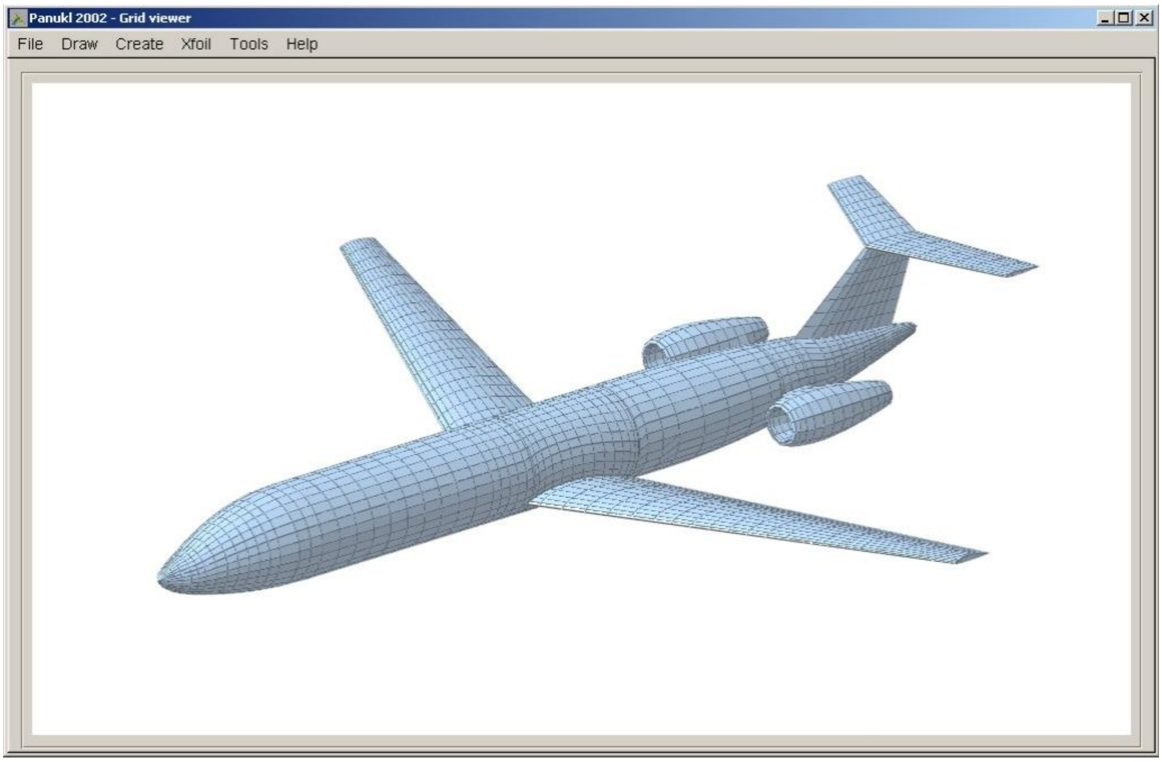


Fig. 88 – Models prepared with the use of the introduced procedure of the export of geometry for PANUKL

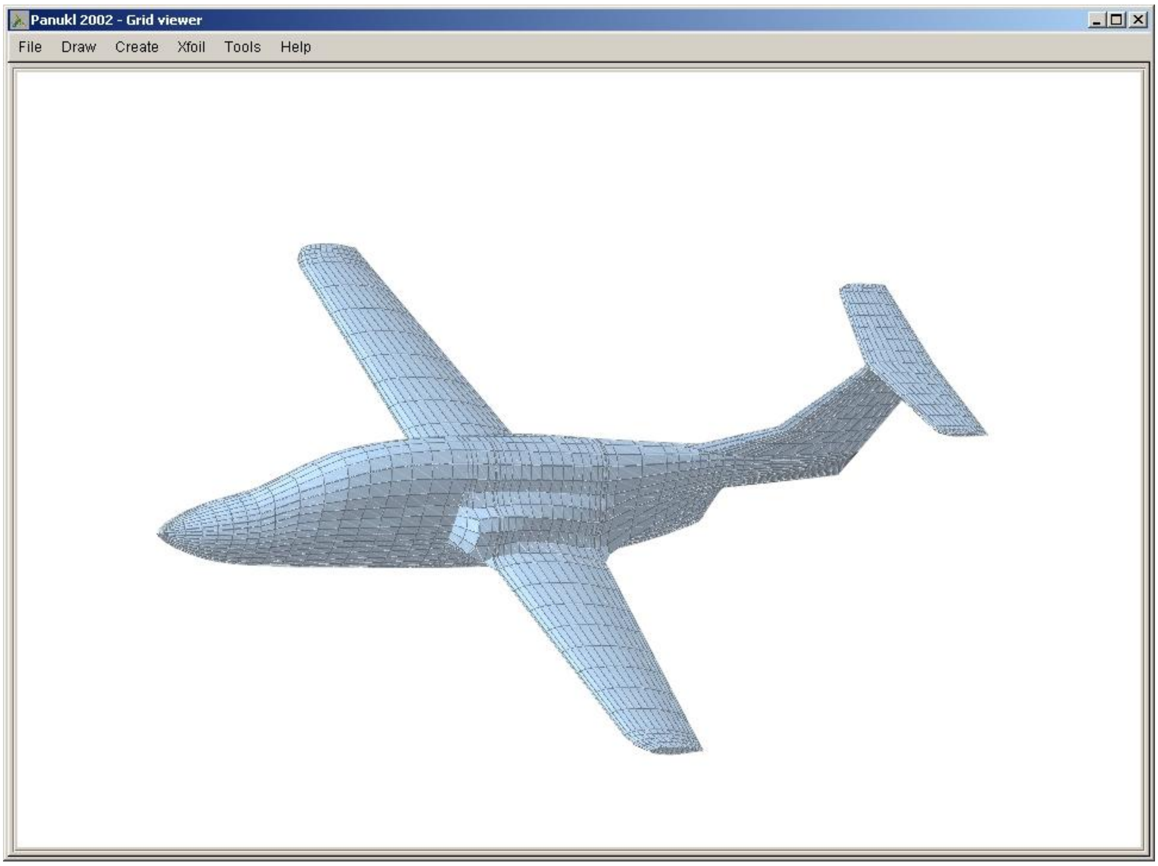


Fig. 89 – Models prepared with the use of the introduced procedure of the export of geometry for PANUKL