

Manual of EBF3GLWingOpt [1-17]

EBF3GLWingOpt: A Global/Local Wing Optimization Design Environment

Rakesh K. Kapania

(Mitchell Professor of Aerospace and Ocean Engineering)

Mohamed Jrad, PhD

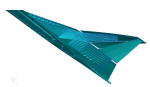
Shuvodeep De

**Multidisciplinary Analysis and Design Center for Advanced Vehicles
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061-0203**

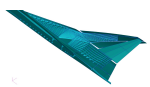


- ❖ Currently, *EBF3GLWingOpt* works on *Unix/Linux* and *Windows*
- ❖ For analysis and optimization, following software are required
 - ❑ Python 2.7
 - ❑ MSC Patran (2014 onwards)
 - ❑ MSC Nastran (2014 onwards)

- ❖ *EBF3GLWingOpt* supports parallel calculations (validated under Linux only. Not tested under Windows)
- ❖ Slide 3 shows the relationship between different components of *EBF3GLWingOpt*
- ❖ Slide 4-12 describes the general process of analysis in *EBF3WingOpt*
- ❖ Slide 13-21 describes the general process of analysis in *EBF3PanelOpt*



EBF3GLWingOpt : Global Wing Optimization



➤ EBF3WingOpt Framework

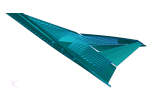
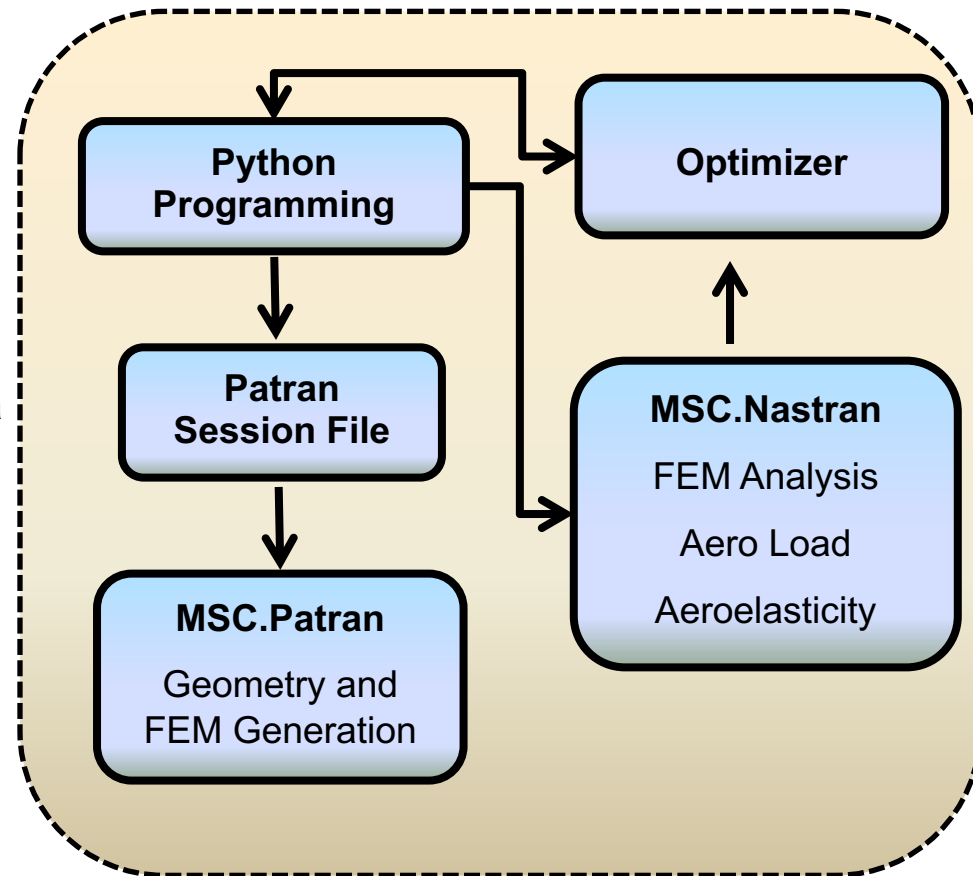
- ❑ Objective: Wing Weight Minimization and Flutter Speed Maximization
- ❑ Outer Mold Line is Not Optimized

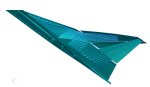
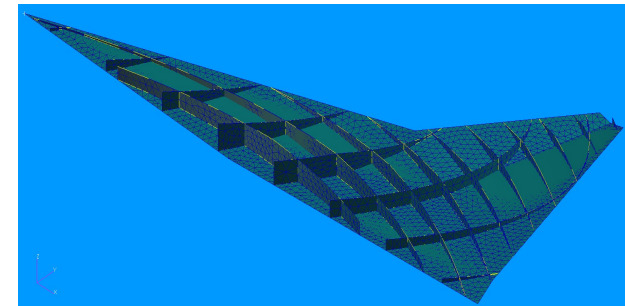
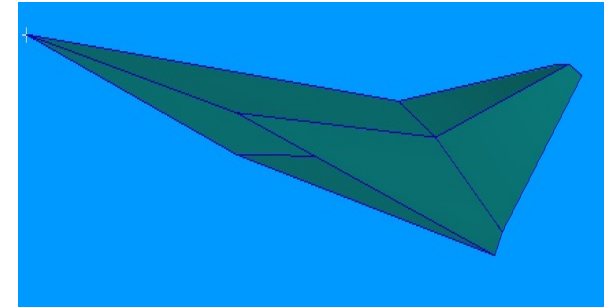
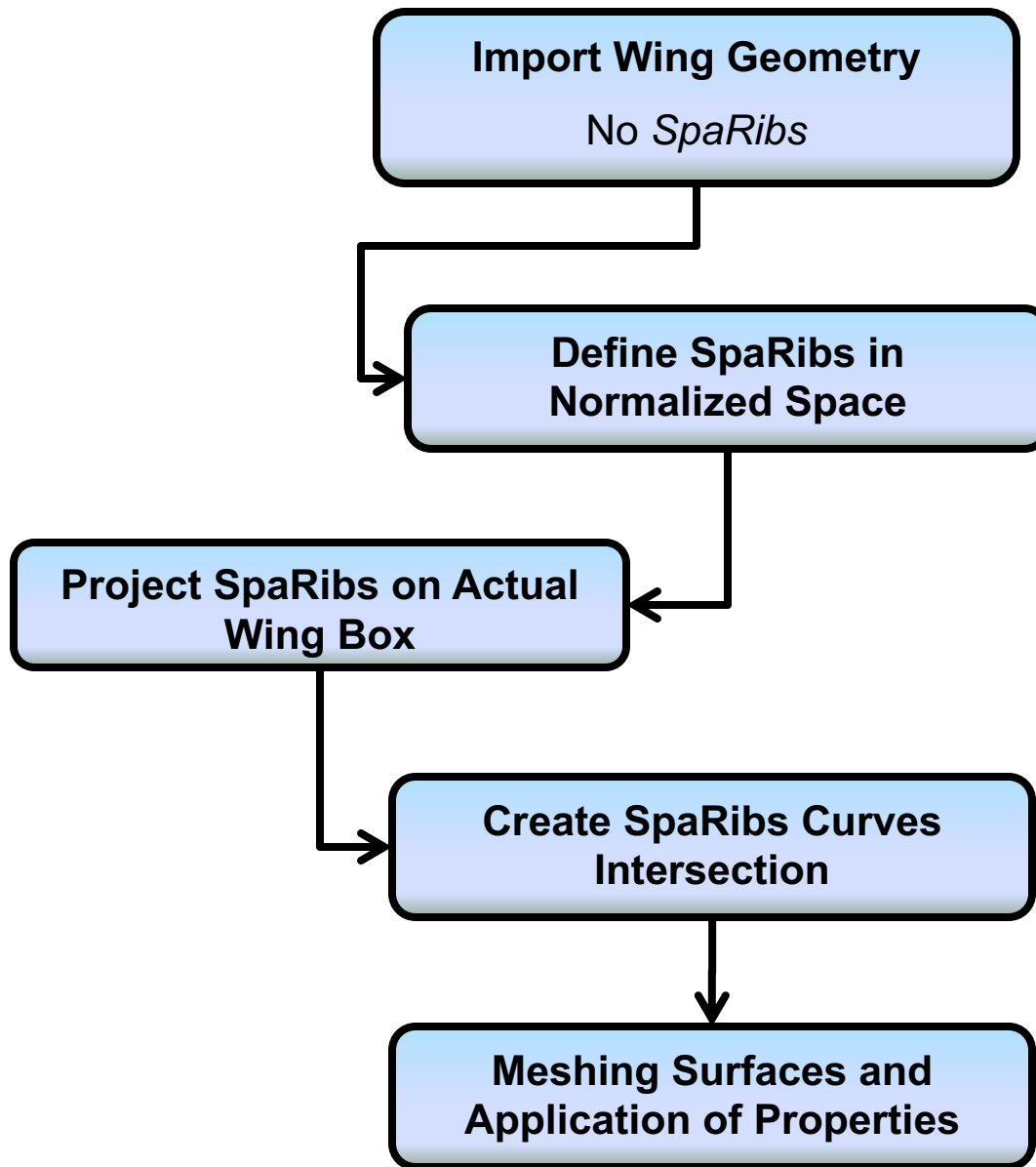
➤ Constraints:

- ❑ Maximum von Mises Stress in Kreisselmeier–Steinhauser (KS) Criteria
- ❑ Maximum Displacement
- ❑ Buckling

➤ Design Variables:

- ❑ Shape Variables of *SpaRibs*
 - ❑ Number of Spars or Ribs in Wing Box
 - ❑ Control Points of Base Curves (*SpaRibs*)
 - ❑ Translation or Offset Spacing Parameters
 - ❑ Number of stiffeners
- ❑ Thickness of Wing Skin, and Thickness of Spars and Ribs

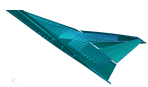




CRM_Design.txt: Design Variables of EBF3GLWingOpt

<i>Dvar1</i>	Front Spar	Constant Parameters
<i>Dvar2</i>	Rear Spar	

<i>Dvar3</i> (p_1)	Number of Spars	Inner Wing Spars	<i>Dvar15</i> (p_2)	SpaRibs control Line start point	Outer Wing Spars
<i>Dvar4</i> (p_2)	SpaRibs control Line start point		<i>Dvar16</i> (p_3)	SpaRibs control Line end point	
<i>Dvar5</i> (p_3)	SpaRibs control Line end point		<i>Dvar17</i> (p_5)	SpaRibs control point spacing	
<i>Dvar6</i> (p_4)	SpaRibs start point spacing		<i>Dvar18</i> (p_6)	SpaRibs end point spacing	
<i>Dvar7</i> (p_5)	SpaRibs control point spacing		<i>Dvar19</i> (p_1)	Number of Ribs	Outer Wing Ribs
<i>Dvar8</i> (p_6)	SpaRibs end point spacing		<i>Dvar20</i> (p_2)	SpaRibs control Line start point	
<i>Dvar9</i> (p_1)	Number of Ribs	Inner Wing Ribs	<i>Dvar21</i> (p_3)	SpaRibs control Line end point	
<i>Dvar10</i> (p_2)	SpaRibs control Line start point		<i>Dvar22</i> (p_4)	SpaRibs start point spacing	
<i>Dvar11</i> (p_3)	SpaRibs control Line end point		<i>Dvar23</i> (p_5)	SpaRibs control point spacing	
<i>Dvar12</i> (p_4)	SpaRibs start point spacing		<i>Dvar24</i> (p_6)	SpaRibs end point spacing	
<i>Dvar13</i> (p_5)	SpaRibs control point spacing				
<i>Dvar14</i> (p_6)	SpaRibs end point spacing				

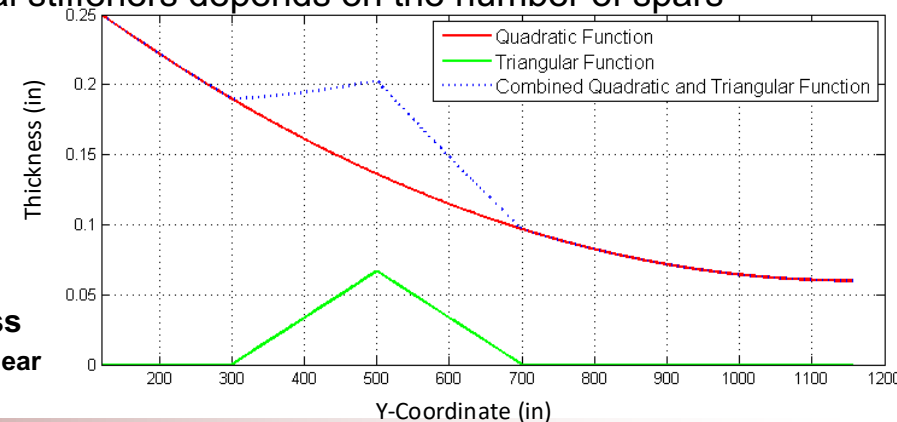


Thickness Variables of Nonlinear Thickness Fields

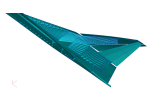
<i>Dvar25</i>	Thickness at Wing Root	Upper Skin	<i>Dvar31</i>	Thickness at Wing Root	Spars
<i>Dvar26</i>	Thickness at Wing Tip		<i>Dvar32</i>	Thickness at Wing Tip	
<i>Dvar27</i>	Non-linear Thickness Parameter		<i>Dvar33</i>	Non-linear Thickness Parameter	
<i>Dvar28</i>	Thickness at Wing Root	Lower Skin	<i>Dvar34</i>	Thickness at Wing Root	Ribs
<i>Dvar29</i>	Thickness at Wing Tip		<i>Dvar35</i>	Thickness at Wing Tip	
<i>Dvar30</i>	Non-linear Thickness Parameter		<i>Dvar36</i>	Non-linear Thickness Parameter	
<i>Dvar37</i>	Height of Stringers	Stringers	<i>Dvar37</i>	Imposed number of longitudinal stiffeners between adjacent spars. Dvar37, Dvar38, etc.	Stiffeners
<i>Dvar38</i>	Thickness of Stringers				

Note:

- The number of thickness design variables depends on the input parameter `use_field`.
- The number of design variables for imposed longitudinal stiffeners depends on the number of spars and on the input parameter `ImposeNumbStiff`
- The thickness variables are used to create the initial wing design. They will be optimized later in the Global/Local process in order to find the thickness of each local panel.



Non-Uniform Thickness
The Red Line Shows the Nonlinear Thickness vs Y-coordinates



❑ Folder 'EBF3GLWingOpt'

➤ 'CRM_InputFiles' and 'RUN': Input Files of MSC.Nastran

<i>CRM_Input.db</i>	Initial Wing Configuration
<i>aero_cards.bdf</i>	Aerodynamic Panels
<i>trim_cards.bdf</i>	Define the Parameters for Static Aeroelastic Analysis
<i>flutter_cards</i>	Define the Parameters for Flutter Analysis
<i>MPC_Grids.bdf</i>	Concentrated Masses Which Present the Control Surfaces
<i>dmicards.bdf</i>	Define DMI cards by interpolating midsurface definitions to pressure locations
<i>std_data.dat</i>	Standard Atmosphere Data
<i>sol*.bdf</i>	Nastran Input Files

- 'CRM_Global_PythonV5': Functions of for initial global wing design creation
- 'EBF3GLWingOpt_PythonV5': Functions of integrated Global/Local process
- 'Program_EBF3_PythonV5': Functions of local panels analysis

Note: Helpful instructions (but not necessary)

- After finishing simulations, one might want to check the folder
...**EBF3GLWingOptV5p3p8\EBF3GLWingOpt_PythonV5\parallel** and delete its content to save some space, if needed.
- After finishing a simulation (or before starting it), one could check the folder
...**EBF3GLWingOptV5p3p8\EBF3GLWingOpt_PythonV5\TempFiles** and delete its content, except the file '**Please do not modify any file or folder while running the code.txt**'



- **'CRM_Design.txt'**: Design Variables file
- **'config.dat'**: Path of EBF3GLWingOpt folder, Patran/Nastran and Python executable files
Format: *Parameter=value

Parameters	Value (example)	Description
Python27	python27	Python2.7 command
mpirun	mpirun	Mpirun command
EBF3PanelOptLinux	.../EBF3GLWingOpt/Program_EBF3_PythonV5/EBF3PanelOpt4.0	Path of EBF3PanelOpt code in Linux
EBF3PanelOptWindows	.../EBF3GLWingOptV5p3p8/Program_EBF3_PythonV5/EBF3PanelOpt4.0	Path of EBF3PanelOpt code in Windows
Work_DirectoryLinux		Work directory in Linux
Work_DirectoryWindows		Work directory in Windows
Work_DirectoryLinuxOtherHosts		Work directory for parallel computing in the other used hosts (if the main drive is not nfs mounted, i.e. not accessible to all of them)
CRMLinux	.../EBF3GLWingOpt/CRM_Global_PythonV5/CRM	Path of EBF3WingOpt code in Linux
CRMWindows	.../EBF3GLWingOptV5p3p8/CRM_Global_PythonV5/CRM	Path of EBF3WingOpt code in Windows
patran_root_dir		
patran_executable_pathWin	C:\MSC.Software\Patran_x64\20121\bin\patran.exe	Patran executable in Windows
patran_executable_pathLin	patran	Patran executable in Linux
patran_template_file		
patran_arguments	-ifile init.pcl -sfr -b -sfp	Patran arguments
nastran_executable_pathWin	C:\MSC.Software\MSC_Nastran\20122\bin\nastran.exe	Nastran executable in Windows
nastran_executable_pathLin	nastran	Nastran executable in Linux
nastran_arguments	batch=no scratch=yes	Nastran arguments

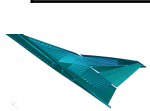
➤ Input Parameters in 'input_data.txt'

Format: Parameter=value1,value2,...

Please keep a copy of the provided example input file 'input_data.txt' for your reference. It contains also a description of each parameter.

There are around 68 input parameters. They include the following parameters.

Material Properties: Elastic modulus(psi); Poisson's ratio; Density(lb/in ³); Yield Stress(psi)	
Angle of attack in trim cards	
Mach number in trim cards	
Thickness fields option 'use_field': linear thickness or nonlinear thickness fields	
Y coordinate of wing root or wing tip	
Flutter method option: calculate Flutter Velocity or Flutter Dynamic Pressure	
Element size	
parallel_computing	Number of processes used in parallel computing
convergence	Convergence criteria: change of wing weight < convergence
selected_panel	Optimize these selected panels in local panel optimization; if it is empty, optimize all the local panels
elm_size_stiffened	Element size of wing with stiffeners (used in Patran mesh)
stiff_space	Average distance between adjacent stiffeners
subcase_num	Number of load cases in buckling analysis
stiffener_option	if stiffener_option = 1, create stiffners between two opposite panel edges else, curvilinear stiffeners can be created between adjacent panel edges



Hosts file **hosts.txt**

Format

Host1	solts=value1	max_slots=value2
Host2	solts=value3	max_slots=value4
Host3	solts=value5	max_slots=value6

Example:

Host1	slots=40	max_slots=64
Host2	slots=6	max_slots=8
Host3	slots=36	max_slots=48

Note:

Value1, value3, and value5 are used to decide how many processes are spawn in each machine.

Max_slots is not used in our case. One can fill it with any positive integer. They can be used if we change the method used in mpi.



PSO input file **input_dataPSO.txt**

Format: Parameter=value1,value2,...

There are around 19 input parameters. They include the following parameters.

Dimension (number of design variables considered in PSO)
Type of design variables
Upper and lower bounds
Swarm size
Number of parallel processes
Weight parameters of the evaluation function
Different options to use or not existing data, and keep or not the analyzed particles (to save storage space)

Please keep a copy of the provided example input file '**input_dataPSO.txt**' for your reference.

It contains also a description of each parameter.



Host file **hostsPSO.txt** format:

```
mainHost1 slotsPSO  nProcessesGlobLoc slotsGlobLoc host2 slotsGlobLoc host3 slotsGlobLoc...
mainHost2 slotsPSO  nProcessesGlobLoc slotsGlobLoc host2 slotsGlobLoc host3 slotsGlobLoc...
```

Parallel processing in the framework EBF3GLWingOpt can be used in Particle Swarm Optimization (PSO) as well as in the global/local process of each particle. The following format for the file hostsPSO.txt is used to contain all the needed information for this double-level parallel processing optimization. Assume we want to run PSO with 10 particles (this number 10 shows up in the input file and not here), using 6 parallel processes split as 4 in mainHostA and 2 in mainHostB. Assume also we want to run each PSO particle with 8 processes split into 5 in the main machine and 3 in a secondary machine. Then, the following format should be used:

```
mainHostA  4  8 5 secondaryHost 3
mainHostB  2  8 5 secondaryHost 3
```

In case we will use only parallel processing in the PSO, we can use the following format:

```
mainHostA  4  1 1
mainHostB  2  1 1
```



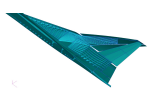
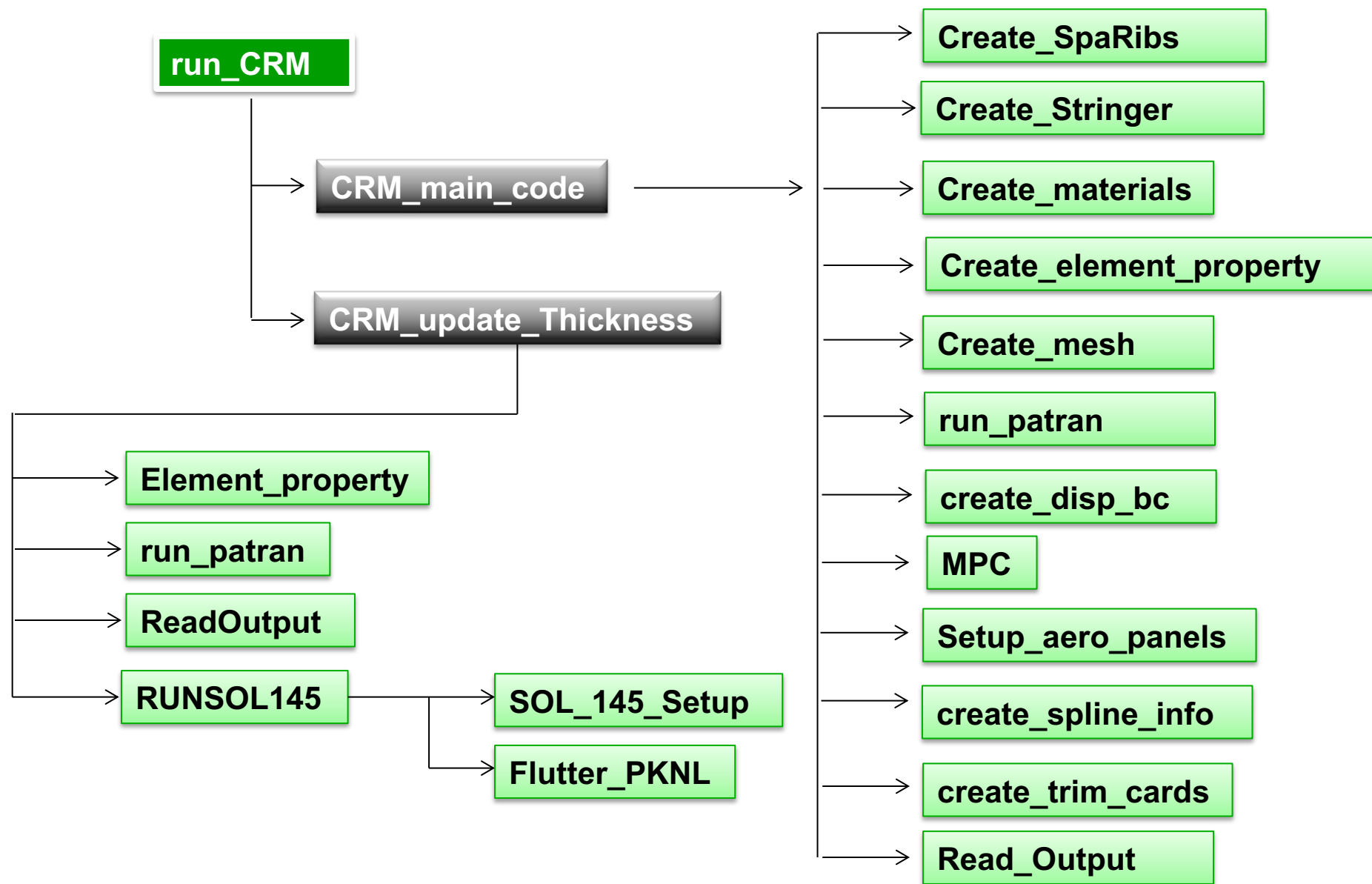
➤ Output Results

<i>Wing_Weight.txt</i>	<i>Optimal Wing Weight</i>
<i>Thickness_opt.mat</i>	<i>Optimal Thicknesses of Panels</i>
<i>Dvar_total.obj</i>	<i>Optimal Design Variables of Panels</i>
<i>Response.txt</i>	<i>Responses of Panel Analysis</i>
<i>Result_record.txt</i>	<i>PSO Optimization results</i>
<i>Result_temp.txt</i>	<i>Analyzed particles during PSO</i>



- Patran session files are created to place SpaRibs in 'Active Geometry' zone
 - ❑ Import Initial .db file: Initial .db is a Patran database which provides the OML of the Aircraft Wing
 - ❑ Create the Front/Rear Spars
 - ❑ During *EBF3GLWingOpt* operation SpaRibs will be placed in the 'Active Geometry' Zone Between Front and Rear Spars
 - ❑ Create the Mesh
 - ❑ Material properties and thicknesses of the members are created in *EBF3GLWingOpt* analysis
- Patran session files are executed to update the .db file and output the Nastran Input Files
- After Nastran successful execution, result file *.f06 is created
- Responses are calculated from .f06 files

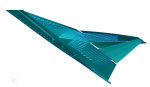




<i>Function Name</i>	Description of Functions in EBF3WingOpt
<i>CRM_main_code</i>	Main function of EBF3WingOpt (Create the geometry, FEM, run analysis, read the results etc.)
<i>CRM_update_Thickness</i>	Update the thickness of shell elements in sizing optimization
<i>Create_SpaRibs</i>	Create the geometry of SpaRibs in the wing boxes
<i>Create_Stringer</i>	Create the stringers at the spar caps
<i>Create_materials</i>	Define the material properties
<i>Create_element_property</i>	Define the element properties like thickness
<i>Create_mesh</i>	Create mesh of the wing structure
<i>Run_patran</i>	Run the Patran job to generate the Nastran input file
<i>Create_disp_bc</i>	Create the boundary condition at the wing root. The grid nodes on the wing root are fixed.
<i>MPC</i>	The control surfaces are represented by concentrated masses. Define the multi-points constraints using rigid body elements to connect the concentrated masses and wing box.
<i>Setup_aero_panels</i>	Write the aerodynamic elements into the 'aero_card.bdf' file
<i>create_trim_cards</i>	Write the flight condition into the 'trim_card.bdf' file
<i>Read_output</i>	Read the Nastran result file of SOL 144, then output the responses such as maximum von Mises stress
<i>Element_property</i>	
<i>RUNSOL145</i>	Write the parameters of flutter analysis into 'flutter_card.bdf'
<i>SOL_145_Setup</i>	Calculate the parameters of flutter analysis
<i>Flutter_PKNL</i>	Read the Nastran result file of SOL 145 (flutter analysis), then output the responses such as flutter velocity



EBF3PanelOpt : Local Panel Optimization



➤ EBF3PanelOpt Framework

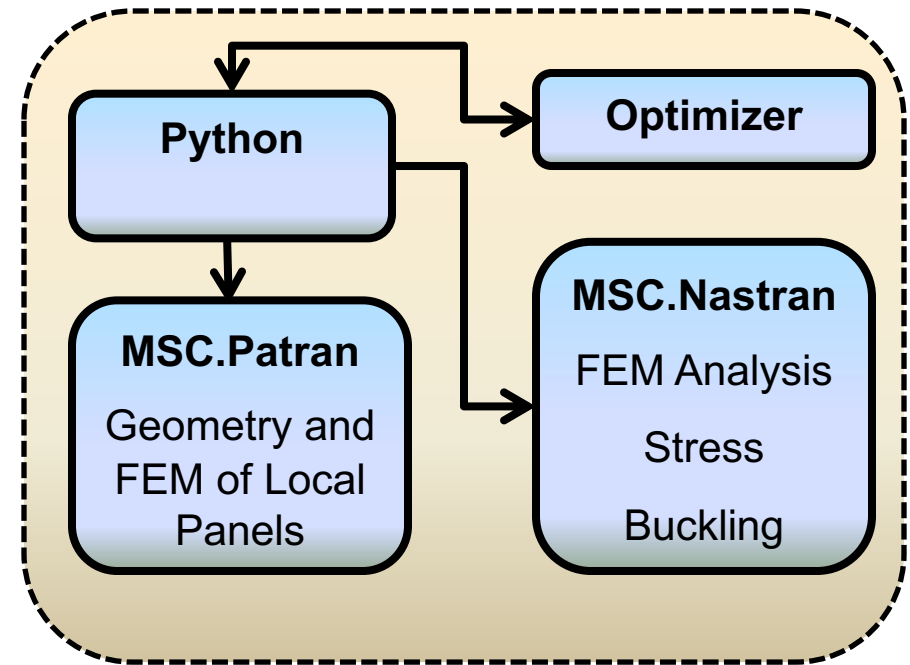
- ❑ Objective: Wing Weight Minimization
- ❑ Topology of SpaRibs Remains Constant

➤ Optimization Constraints:

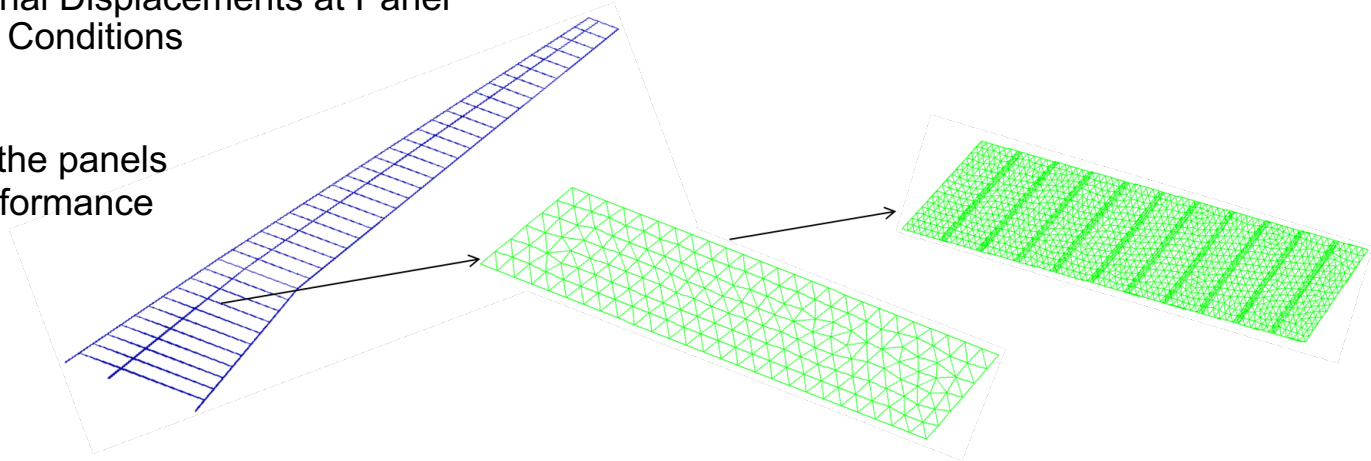
- ❑ Maximum von Mises Stress
- ❑ Buckling

➤ Boundary Condition:

- ❑ Global Displacements are extracted from .f06 file of the global wing analysis
- ❑ Displacements at the Panel Edge Grids are Calculated by Interpolation of the Global Displacements
- ❑ Apply the Translational Displacements at Panel Edges as Boundary Conditions



Stiffeners are applied onto the panels to improve the buckling performance



- Import the CRM.db and FEM_only.bdf of the global wing (without stiffeners)
- Create the Un-stiffened Local Panels from the Global Wing Model
- Add Stiffeners on the Un-stiffened Local Panels
 - ❑ Number of Stiffeners are Determined by Panel Size
- Mesh the Stiffened Panel Using Fine Grids
- Boundary Conditions:
 - Translational Displacements at Panel Edges Are Enforced as Boundary Conditions
 - ❑ Displacements at Panel Edges are Calculated by Interpolation of the Displacement Results of the Global Wing Model with Coarse Mesh
- Stiffened Panel Optimization:
 - ❑ Option 1: Sizing Optimization with Evenly Distributed Straight Stiffeners
 - ❑ Three Design Variables: Panel Thickness, Stiffener Height and Thickness
 - ❑ Option 2: Stiffened Panel Optimization with Curvilinear Stiffeners
 - ❑ Six Design Variables Including Starting, Control and Ending Points of Stiffeners



➤ Design Variables in EBF3PanelOpt:

- ☐ Thickness of Local Panels
- ☐ Height and Thickness of Stiffeners
- ☐ Shape Variables of Curvilinear Stiffeners (Optional)
 - Start-points, End-points and Control Points of Base Curves

<i>Constant Parameter</i>	Number of Stiffeners	Determined by Panel Size
<i>Dvar1</i>	Panel Thickness	Uniform Thickness for Each Panel
<i>Dvar2</i>	Stiffener Height	Uniform Height for All the Stiffeners on Each Panel
<i>Dvar3</i>	Stiffener Thickness	Uniform Thickness for All the Stiffeners on Each Panel
<i>Dvar4</i> (P_4)	Stiffeners start point spacing	Option 1: DV4 ~ DV6 are Constant Parameters of Straight Stiffeners Option 2: DV4 ~ DV6 are Design Variables of Curvilinear Stiffeners
<i>Dvar5</i> (P_5)	Stiffeners control point spacing	
<i>Dvar6</i> (P_6)	Stiffeners end point spacing	

The Geometry of Wing Skins, Spars and Ribs Remain Constant in the Panel Optimization



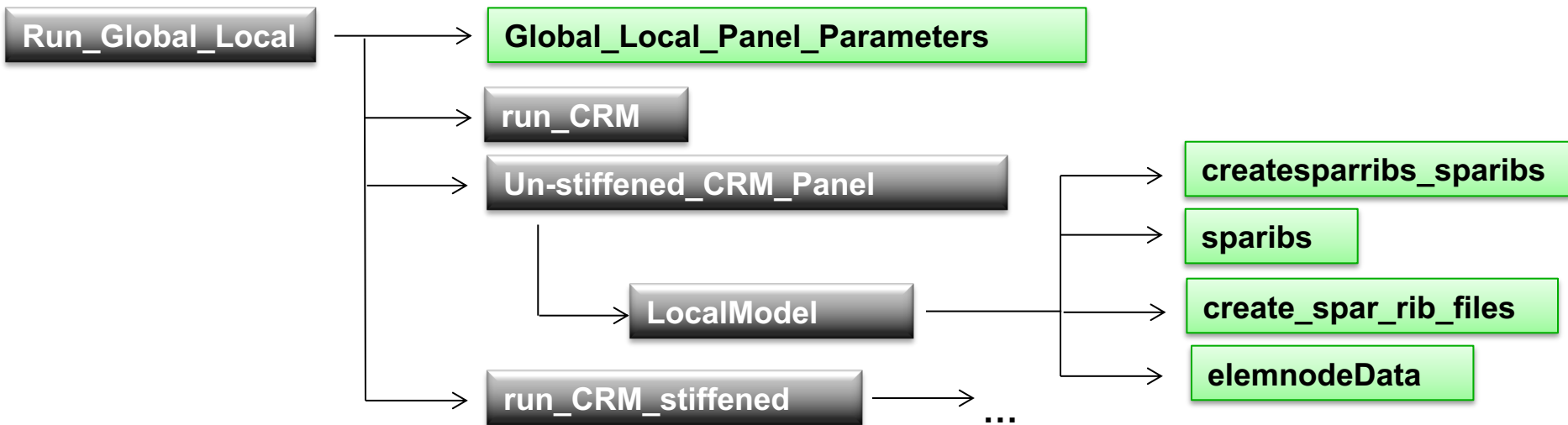
➤ Folder 'Design_1':

- ❑ Patran Database CRM_Design.db: Geometry and FEM of Wing with Spars and Ribs
- ❑ FEM_only.bdf: Finite Elements of CRM Wing

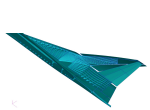
➤ Folder 'Local_Panel_ini':

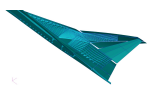
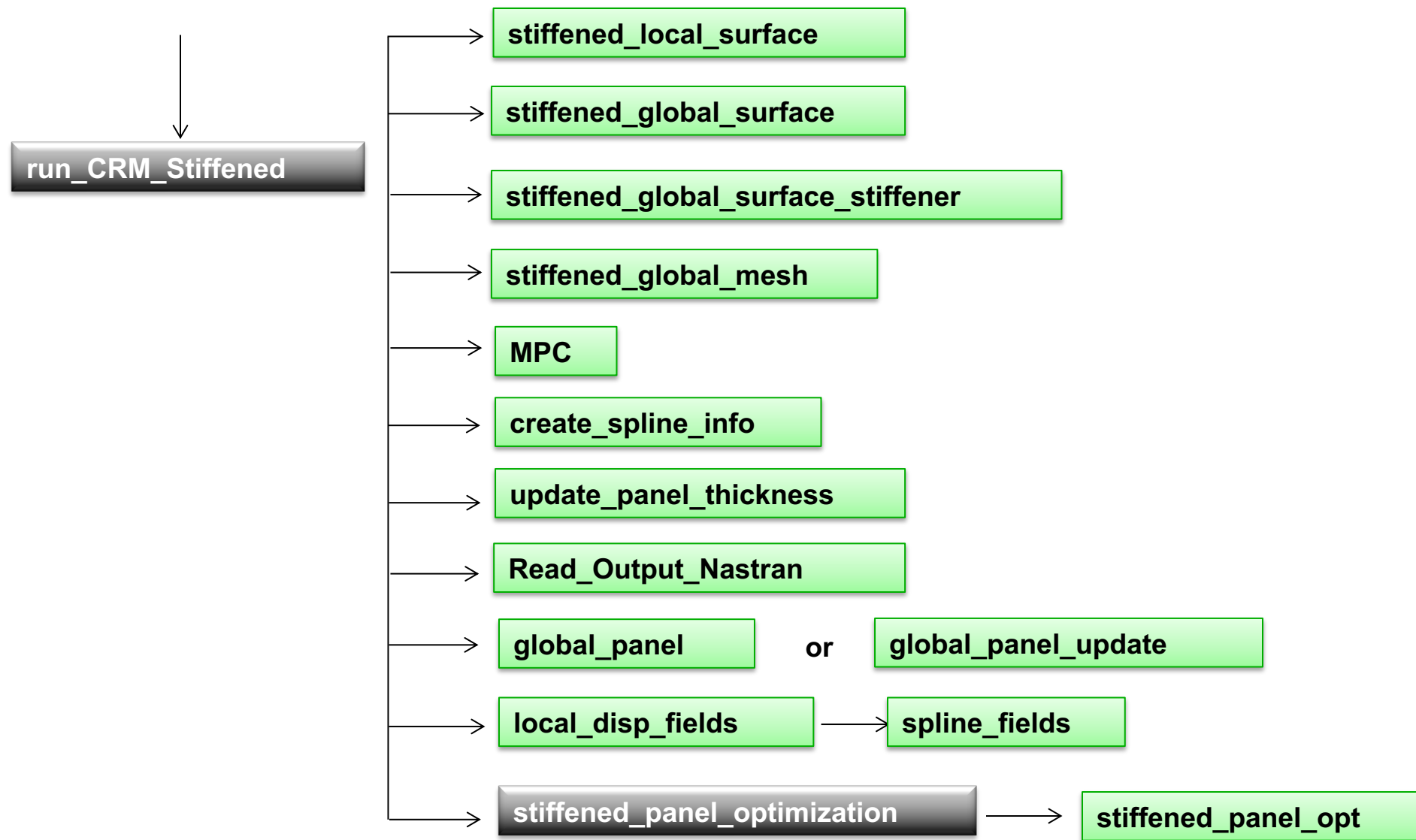
- ❑ 'EBF3_Python' Folder: Input Files of Single Stiffened Panel Analysis
- ❑ 'RUN' Folder: Nastran Input Files 'aero_cards.bdf', 'trim_cards.bdf', 'flutter_cards', 'sol*.bdf'
- ❑ 'panelinfo.mat': XYZ Coordinates of the Vertices of Aerodynamic Panels



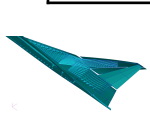


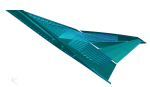
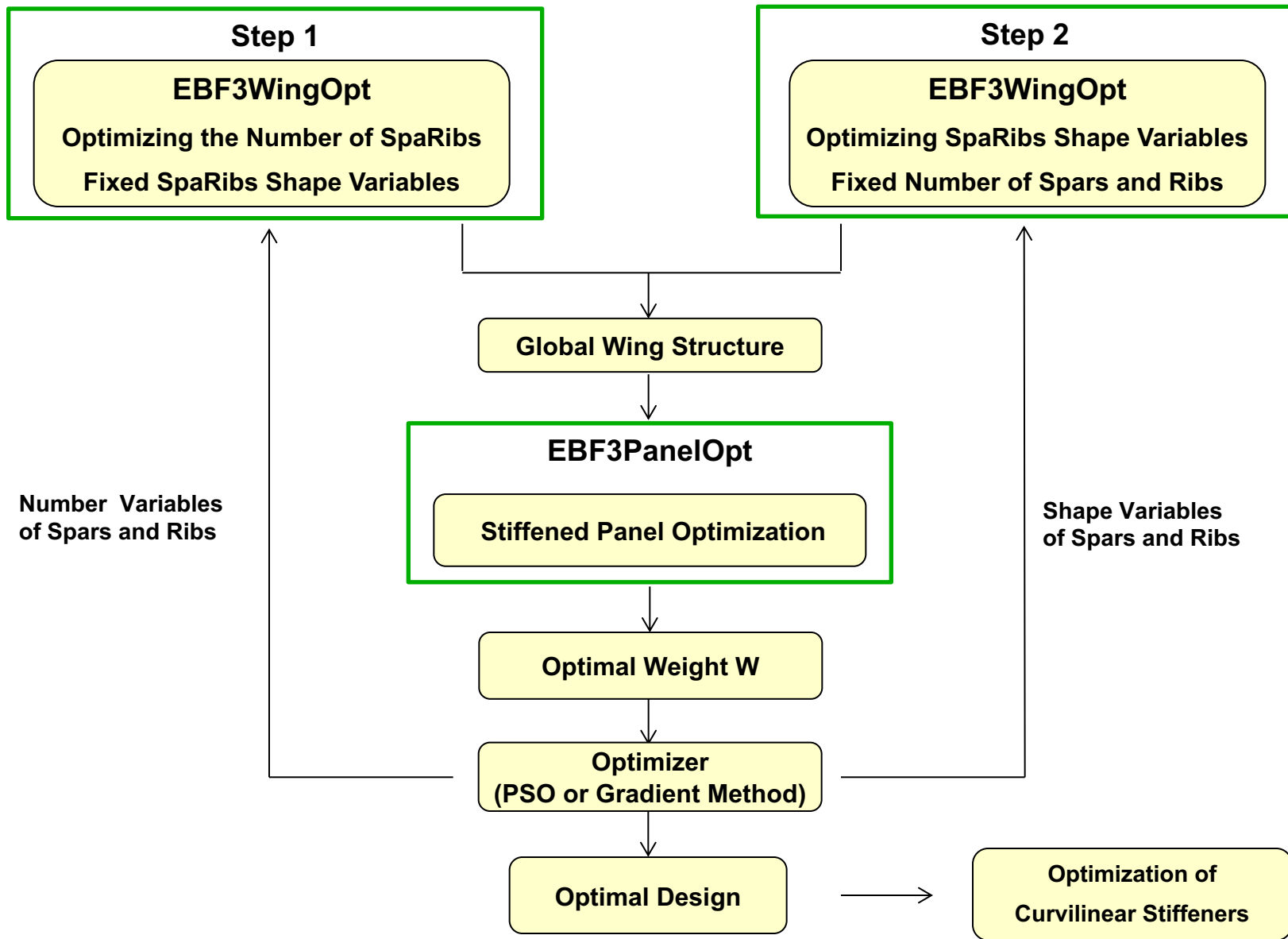
Function Name	Description of Functions in EBF3PanelOpt
<i>run_Global_Local</i>	The main function of EBF3PanelOpt
<i>Global_Local_Panel_Parameters</i>	Define the parameters of the CRM wing for the global-local optimization
<i>run_CRM</i>	Create the CRM design with <i>SpaRibs</i>
<i>un-stiffened_CRM_Panel</i>	Extract the local panel elements and nodes from the global CRM wing model
<i>LocalModel</i>	This function defines the local panel model
<i>createsparribs_sparibs</i>	Create nodes lists of spars or ribs in the Patran database
<i>sortdata</i>	This function defines the panel edges using the grid nodes at the intersections of spars and ribs. Then find the interior elements and grid nodes of the local panels
<i>create_spar_rib_files</i>	This function creates the Nastran input files for each spar or rib
<i>elemnodeData</i>	This function extracts nodes and elements data from the Nastran input files of skins, spars and ribs

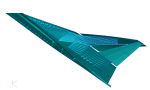
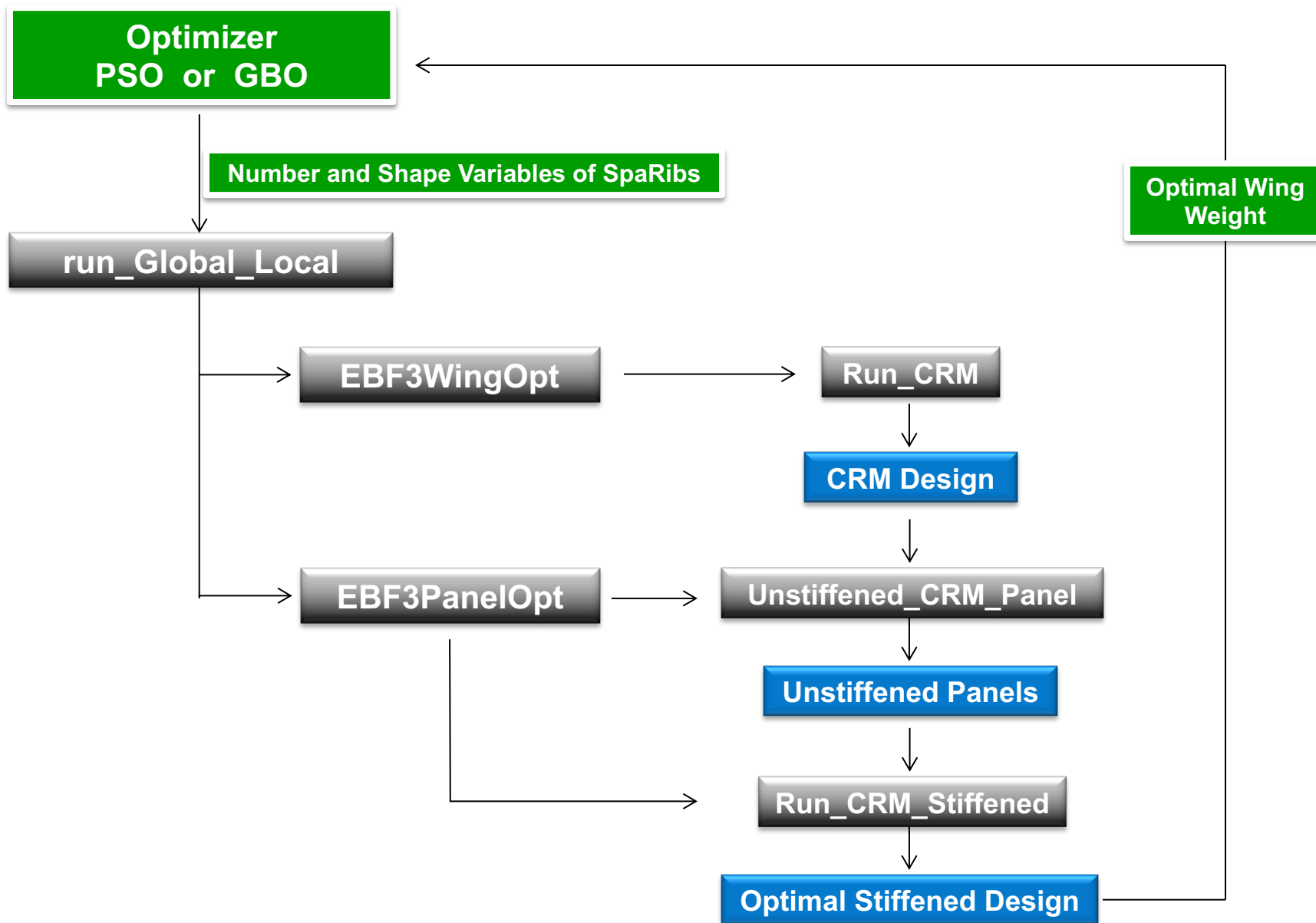




Function Name	Description of Functions in <i>EBF3PanelOpt</i>
<i>run_CRM_stiffened</i>	Optimizing the CRM wing with stiffened panels
<i>stiffened_local_surface</i>	Create the Patran database for each local panel
<i>stiffened_local_surface_parallel</i>	Parallel processing of <i>stiffened_local_surface</i>
<i>stiffened_local_surface_single</i>	Single process computing of <i>stiffened_local_surface</i>
<i>stiffened_global_surface</i>	Create a surface for each local panel
<i>stiffened_global_surface_stiffener</i>	Add the stiffeners on the local panels
<i>stiffened_global_mesh</i>	Mesh the stiffened wing using fine grids
<i>MPC</i>	The control surfaces are represented by concentrated masses. Define the multi-points constraints using rigid body elements to connect the concentrated masses and wing box.
<i>create_spline_info</i>	Write the spline nodes into Nastran input file
<i>update_panel_thickness</i>	Update the panel thickness in optimization process
<i>Read_Output_Nastran</i>	Read the displacement data from the Nastran result file of SOL 101
<i>create_panel_bdf</i>	Create the Nastran input files of local panels
<i>create_panel_bdf_parallel</i>	Parallel processing of <i>create_panel_bdf</i>
<i>create_panel_bdf_single</i>	Single process computing of <i>stiffened_panel_optimization</i>
<i>global_panel</i>	Create the Nastran input files of local panels
<i>local_disp_fields</i>	Write the displacement data of local panels into Nastran input files
<i>spline_fields</i>	Read the displacement data from local panel results
<i>stiffened_panel_optimization</i>	Optimize the stiffened or unstiffened local panels in the aircraft wing
<i>stiffened_panel_optimization_parallel</i>	Parallel processing of <i>stiffened_panel_optimization</i>
<i>stiffened_panel_optimization_single</i>	Single process computing of <i>stiffened_panel_optimization</i>
<i>stiffened_panel_opt</i>	Optimize the panel thickness of a single stiffened or unstiffened panel



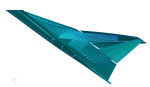




<i>Function Name</i>	Description of Functions in EBF3GLWingOpt
<i>pso</i>	Particle swarm optimization with continuous and integer design variables
	<i>Inertia_weight</i> : Inertia weight to prevent velocities becoming too large
	<i>C1</i> : Scaling co-efficient on the social component
	<i>C2</i> : Scaling co-efficient on the cognitive component
	<i>Dimension</i> : number of the design variables
	<i>Type_DV</i> : type of design variable: integer or real. For example <i>Type_DV</i> = ['real', 'real', 'int']
	<i>Ub_DV</i> : upper boundary of design variables
	<i>Lb_DV</i> : lower boundary of design variables
	<i>Iterations</i> : maximum number of iterations
	<i>swarmSize</i> : population size of particle swarm in each iteration
	<i>Convergence_iter</i> : check the convergence in consecutive iterations
	<i>Convergence_relative</i> : use the relative change of objective function as convergence criteria
<i>gradient</i>	Gradient based optimization with continuous design variables
	options.TolFun: relative convergence tolerance of objective function
	options.TolX: relative convergence tolerance of design variables
	options.MaxIter: maximum number of iterations
	options.FinDiffRelStep: finite difference step of design variables
	options.MaxFunEvals: maximum number of objective function evaluation



EBF3GLWingOpt v6.0.1 Upgradation



CRM_Design.txt: Design Variables of EBF3GLWingOpt v 6.0.1

- Design variables Dvar1 to Dvar41 same as v 5.8.3
- New design variables introduced in v 6.0.1 given below:

<i>Dvar42</i>	O11
<i>Dvar43</i>	O12
<i>Dvar44</i>	O21
<i>Dvar45</i>	O22
<i>Dvar46</i>	O31
<i>Dvar47</i>	O32
<i>Dvar49</i>	Number of division in displacement spectrum (for Selective Stiffening)
<i>Dvar50</i>	M
<i>Dvar51</i>	N
<i>Dvar52</i>	Stiffener height in inner wing-box
<i>Dvar53</i>	Stiffener height in outer wing-box

<i>Dvar54</i>	Stiffener thickness for panel thickness 0.15 -0.2 in
<i>Dvar55</i>	Stiffener thickness for panel thickness 0.2-0.3 in
<i>Dvar56</i>	Stiffener thickness for panel thickness 0.3 in

➤ Offset Parameters:

O11, O12, O21, O22, O31, O32

➤ Thickness Increment Function (TIF)

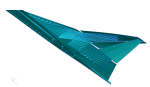
D_{maxG} = Maximum displacement in eigenvector

D_{maxP} = Maximum displacement in a panel

$$m = \frac{D_{maxP}}{D_{maxG}}$$

$$TIF = m \left(\left(\frac{BF}{1.05} \right)^{-\frac{N}{2}} - 1 \right) + 1 \text{ for Skin panels}$$

$$= m \left(\left(\frac{BF}{1.05} \right)^{-\frac{M}{2}} - 1 \right) + 1 \text{ for SpaRibs}$$



- **'CRM_Design.txt'**: Design Variables file
- **'config.dat'**: Path of EBF3GLWingOpt folder, Patran/Nastran and Python executable files
Format: *Parameter=value
(spaces left blank need not to be entered)

Parameters	Value (example)	Description
Python27	python27	Python2.7 command
mpirun	mpirun	Mpirun command
Work_DirectoryLinux		Work directory in Linux
Work_DirectoryWindows		Work directory in Windows
Work_DirectoryLinuxOtherHosts		Work directory for parallel computing in the other used hosts (if the main drive is not nfs mounted, i.e. not accessible to all of them)
CRMLinux	.../EBF3GLWingOptV6p0p1/CRM_Global_PythonV6/CRM	Path of EBF3WingOpt code in Linux
CRMWindows	.../EBF3GLWingOptV6p0p1\CRM_Global_PythonV6\CRM	Path of EBF3WingOpt code in Windows
patran_root_dir		
patran_executable_pathWin	C:\MSC.Software\Patran_x64\20121\bin\patran.exe	Patran executable in Windows
patran_executable_pathLin	patran	Patran executable in Linux
patran_template_file		
patran_arguments	-ifile init.pcl -sfr -b -sfp	Patran arguments
nastran_executable_pathWin	C:\MSC.Software\MSC_Nastran\20122\bin\nastran.exe	Nastran executable in Windows
nastran_executable_pathLin	nastran	Nastran executable in Linux
nastran_arguments	batch=no scratch=yes	Nastran arguments



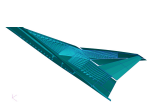
➤ Input Parameters in 'input_data.txt'

Format: Parameter=value1,value2,...

Most parameters are common as those for **EBF3GLWingOpt v5.3.8**

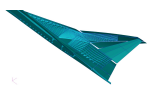
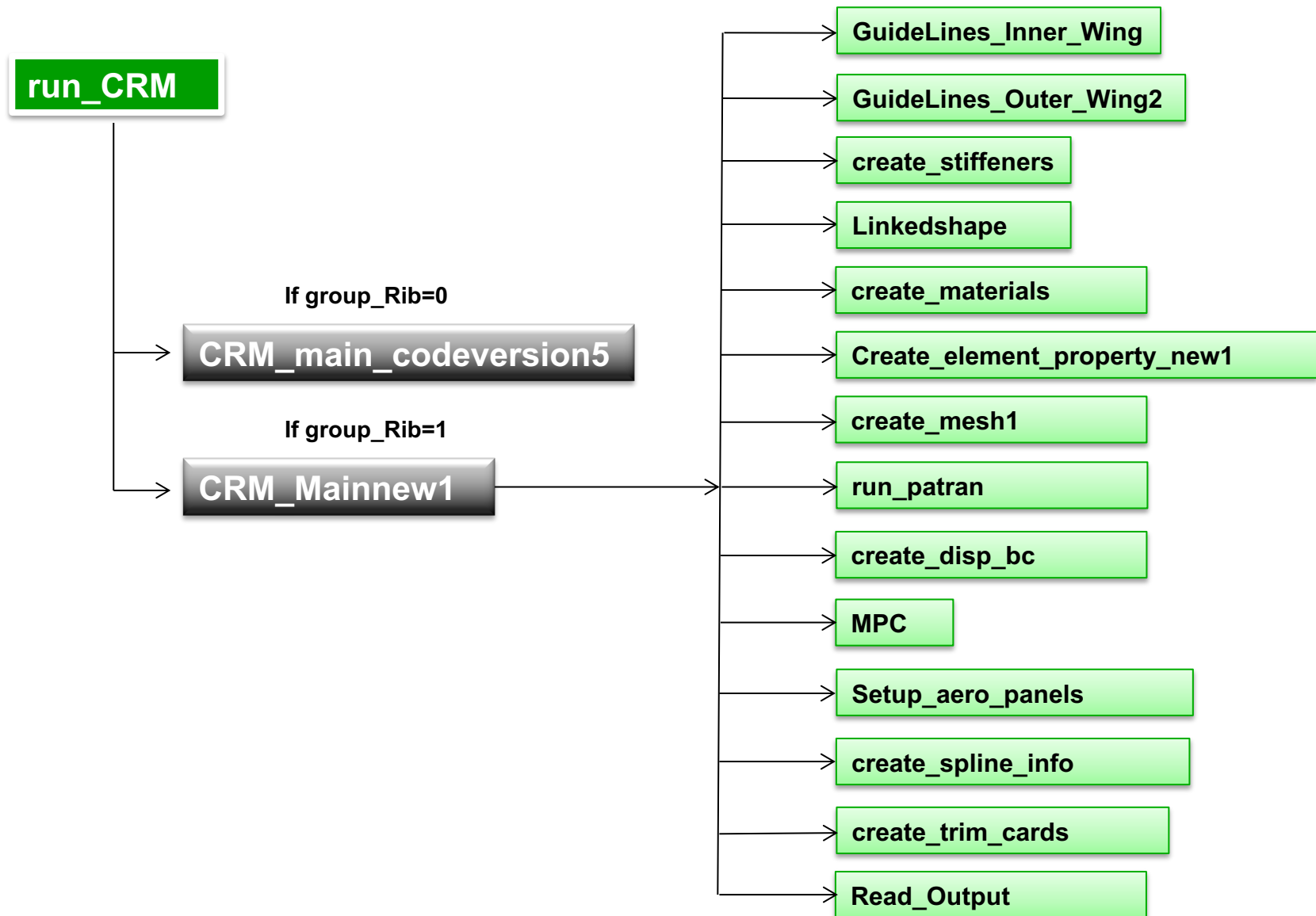
The list of additional parameters for **EBF3GLWingOpt v6.0.1** given below

Parameters	Values that can be assigned and corresponding output
group_Rib	group_Rib == 0, consider independent design space for inner and outer wing-box and define three Rib groups in MSC-PATRAN database: Rib_inboard, Rib_mid Rib_outboard group_Rib ==1, consider singe design space for the entire wing and define one Rib group in MSC-PATRAN database
SRTYPE	SRTYPE == 1, create straight spars and ribs perpendicular to as in conventional design SRTYPE == 2, create spars and ribs using second order B-Spines (considering only the starting and the ending points)
CreateHoles	CreateHoles == 0, create ribs without holes CreateHoles == 1, create ribs with holes (currently available only for multiple wing-boxes)
Bifurcation	Bifurcation == 0, do not create spar bifurcation in the inner wing-box Bifurcation == 1, create spar bifurcation in the inner wing-box (available only for Local Method == 1)
rib9000	rib9000 == 0, do not create rib at the trailing-edge break rib9000 == 1, create rib at the trailing-edge break
Extension	Percentage of extension of leading and trailing edges while using extended space method (for single wing-box)

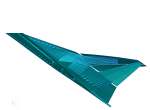


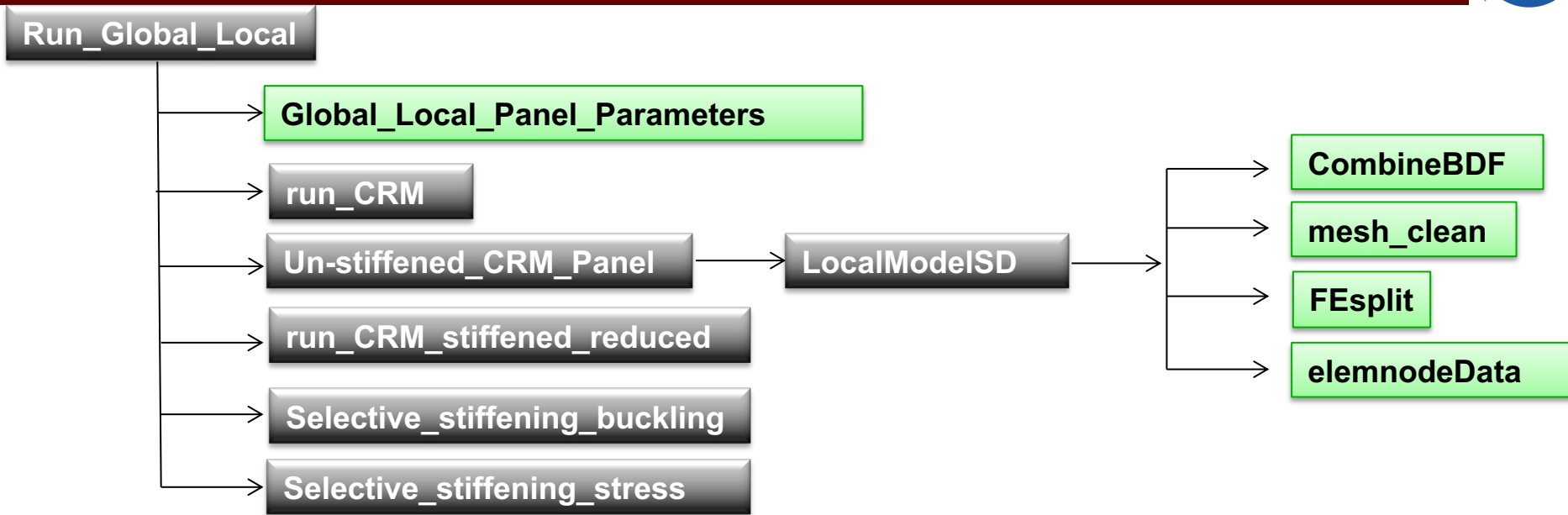
Input parameter	Values that can be assigned and Corresponding Output
iter_final	Maximum number of iteration in Global-local optimization using Nastran Solution 200. (Set to 0 for optimization only by Selective Stiffening)
iter_max	Maximum allowable number of iteration (Global-local + Selective Stiffening)
num_stringer	Number of stringer/stiffener in inner wing-box
num_stringers	Number of stringer/stiffener in outer wing-box
OptimizeSR	OptimizeSR=1 , optimize SpaRibs in Global-local optimization using Nastran Solution 200 OptimizeSR=0 , skip
OptimizePanel	OptimizeSR=1 , optimize skin panels in Global-local optimization using Nastran Solution 200 OptimizeSR=0 , skip
Local_Tolerance	Skip local optimization of panel if change in weight in consecutive global iteration less than this value



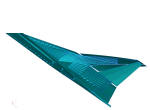


<i>Function Name</i>	<i>Description of Functions in EBF3WingOpt v6.0.1</i>
<i>CRM_main_codeversion5</i>	Main function of EBF3WingOpt v6.0.1 for group_Rib=0 (
<i>CRMMain1</i>	Main function of EBF3WingOpt v6.0.1 for group_Rib =1 (Parameterization involving Offset Parameters)
<i>GuideLines_Inner_Wing</i>	Creates curves in the inner wing-box for stiffener construction
<i>GuideLines_Outer_Wing2</i>	Creates curves in outer wing-box for stiffener construction
<i>Create_materials</i>	Define the material properties
<i>Create_element_property_new1</i>	Define the element properties like thickness
<i>Create_mesh1</i>	Create mesh of the wing structure (skin and SpaRibs meshed with triangular elements, stiffeners meshed with quad elements)
<i>Run_patran</i>	Run the Patran job to generate the Nastran input file
<i>Create_disp_bc</i>	Create the boundary condition at the wing root. The grid nodes on the wing root are fixed.
<i>MPC</i>	The control surfaces are represented by concentrated masses. Define the multi-points constraints using rigid body elements to connect the concentrated masses and wing box.
<i>Setup_aero_panels</i>	Write the aerodynamic elements into the 'aero_card.bdf' file
<i>create_trim_cards</i>	Write the flight condition into the 'trim_card.bdf' file
<i>Read_output</i>	Read the Nastran result file of SOL 144, then output the responses such as maximum von Mises stress

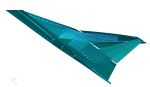




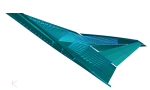
<i>Function Name</i>	<i>Description of Functions in EBF3PanelOpt 6.0.1</i>
<i>run_Global_Local</i>	The main function of EBF3PanelOpt
<i>Global_Local_Panel_Parameters</i>	Define the parameters of the CRM wing for the global-local optimization
<i>run_CRM</i>	Creates the CRM design with <i>SpaRibs</i>
<i>un-stiffened_CRM_Panel</i>	Extracts the local panel elements and nodes from the global CRM wing model and creates .bdf for each
<i>LocalModelSD</i>	Defines local panels (of any shape)
<i>CombineBDF</i>	Combines .bdf of part of SpaRib in inner wing-box and part in outer wing-box
<i>mesh_clean</i>	Modifies mesh of skin and Sparibs to removes highly skewed elements from Skin
<i>FEsplit</i>	Splits FEM of upper skin and lower skin using SpaRibs intersection
<i>elemnodeData</i>	Extracts nodes and elements data from the Nastran input files of skins, spars and ribs



Parallel Computing and Installation



- ❖ For Red Hat Based Systems, set the content of the files `.bash_profile` and `.bashrc` in the home directory as follows:
 - **cat .bash_profile** (for interactive shells)
module load openmpi-x86_64 Enables to use mpi (loads the module)
 - **cat .bashrc** (for non-interactive shells)
source /usr/share/Modules/init/bash Enables the module commands.
module load openmpi-x86_64 Enables to use mpi (loads the module)
export PATH="\${PATH}:/aoe/bin" Enables the system to find Patran and all installed software.
- ❖ Make firewall exceptions in order to allow communication between the different hosts
- ❖ Create an empty folder with the same name in all machines used for parallel computation and add its path in the file `config.dat` under:
`*Work_DirectoryLinuxOtherHosts,folder_path`



➤ **EBF3GLWingOpt v5.3.8**

1. Go to ...\\EBF3GLWingOptV5p3p8\\EBF3GLWingOpt_PythonV5
2. Prepare all the required input files described previously.
3. Run: `python2.7command run_Global_Local.py -i input_data.txt`

➤ **Particle Swarm Optimization for EBF3GLWingOpt v5.3.8**

1. Go to ...\\EBF3GLWingOptV5p3p8\\EBF3GLWingOpt_PythonV5
2. Prepare all the required input files described previously
3. Run: `python2.7command run_pso.py -i input_dataPSO.txt`



➤ **EBF3GLWingOpt v6.0.1**

1. Go to ...\\EBF3GLWingOptV6p0p1\\EBF3GLWingOpt_PythonV6
2. Prepare all the required input files described previously.
3. Run: `python2.7command run_Global_Local.py -i input_data.txt`

➤ **Particle Swarm Optimization for EBF3GLWingOpt v6.0.1**

1. Go to ...\\EBF3GLWingOptV6p0p1\\EBF3GLWingOpt_PythonV6
2. Prepare all the required input files described previously
3. Run: `python2.7command run_pso.py -i input_dataPSO.txt`



Alternative: Integrated function EBF3GLWingOpt v5.3.8 and v6.0.1

Parametrization methods:

-v 0 corresponds to v5.3.8

-v 1 corresponds to v6.0.1

➤ **EBF3GLWingOpt v5.3.8**

1. Go to ...\\EBF3GLWingOpt\\
2. Prepare all the required input files described previously and place them under ...\\EBF3GLWingOpt\\input_files_V5p3p8\\
3. For Global-Local: python2.7command run_Global_Local_integ.py -i input_data.txt -v 0
4. For PSO: python2.7command run_pso_integ.py -i input_dataPSO.txt -v 0

➤ **EBF3GLWingOpt v6.0.1**

1. Go to ...\\EBF3GLWingOpt\\
2. Prepare all the required input files described previously and place them under ...\\EBF3GLWingOpt\\input_files_V6p0p1\\
3. For Global-Local: python2.7command run_Global_Local_integ.py -i input_data.txt -v 1
4. For PSO: python2.7command run_pso_integ.py -i input_dataPSO.txt -v 1
5. => This function will copy the input files into the desired code directory and run it.



Examples of Analysis and Optimization Using EBF3GLWingOpt



➤ NASA Common Research Model

- Number of the design variables: 13
- Population size of particle swarm: 28
- 3 spars

$$f = \beta_w W + \frac{\beta_v}{\sqrt{V_f}}$$

Double Level Parallel computing:

PSO level: **28 processes**

Each particle: **10 processes**

W: Weight (lbs)

V_f : Flutter Speed (knots)

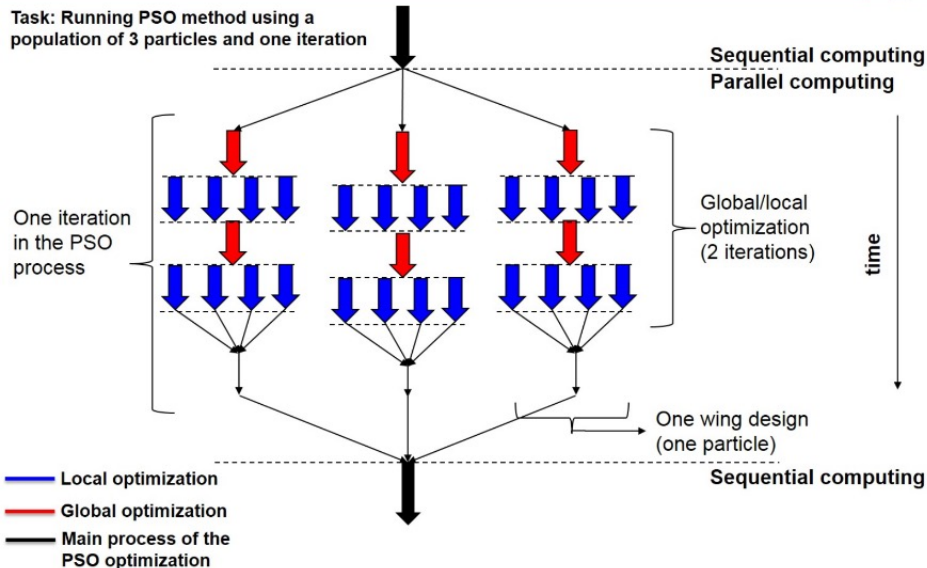
β_w : Mass penalty (1/lbs)

β_v : Flutter velocity penalty (knots)

$$\beta_w = 0.1/\text{lbs}$$

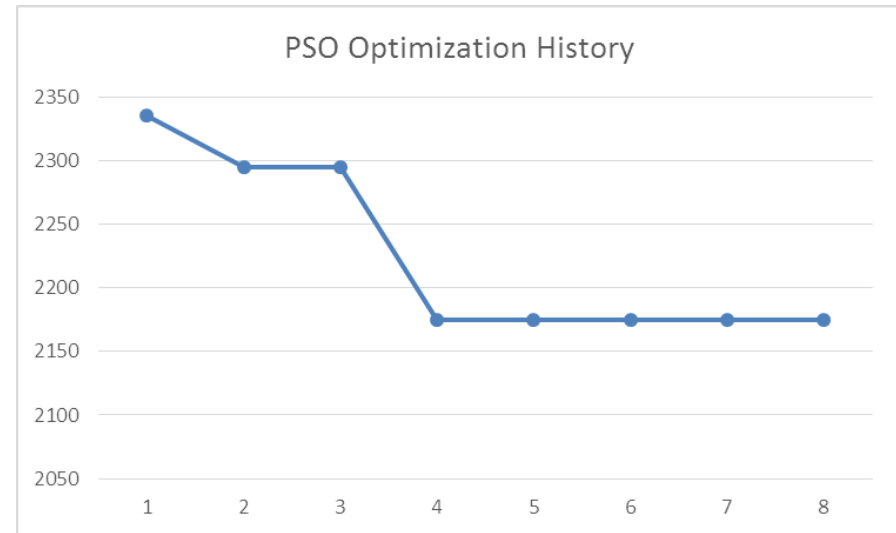
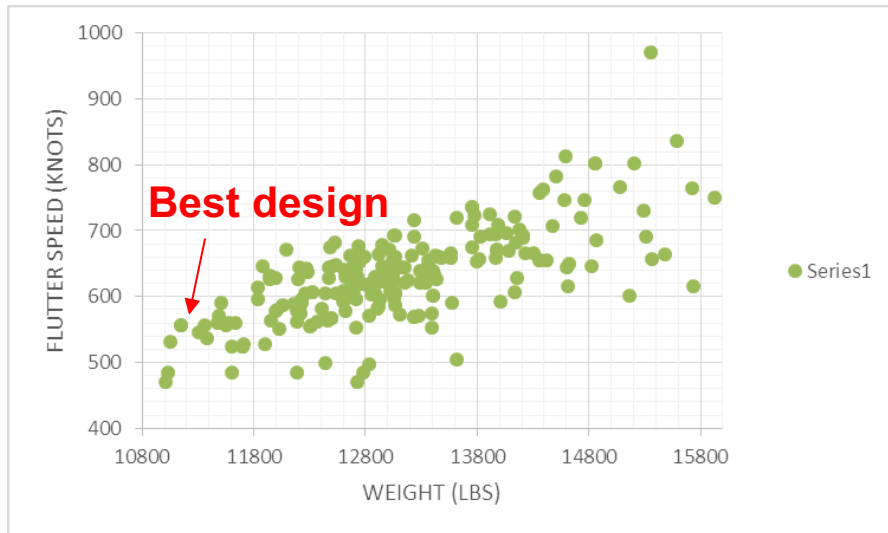
$$\beta_v = 25 \cdot 10^3 \sqrt{\text{knots}}$$

Task: Running PSO method using a population of 3 particles and one iteration



		$X_{\min}$	$X_{\max}$
DV6	IW spars: Start point spacing	0.2	10
DV7	IW spars: Control point spacing	0.2	10
DV8	IW spars: End point spacing	0.2	10
DV9	IW number of ribs	3	20
DV12	IW ribs: Start point spacing	0.2	10
DV13	IW ribs: Control point spacing	0.2	10
DV14	IW ribs: End point spacing	0.2	10
DV17	OW spars: Control point spacing	0.2	10
DV18	OW spars: End point spacing	0.2	10
DV19	OW number or ribs	5	40
DV22	OW spars: Start point spacing	0.2	10
DV23	OW spars: Control point spacing	0.2	10
DV24	OW spars: End point spacing	0.2	10

- PSO Run in Advanced Research Computing using **7 machines** with **24 cores** each.
- **Double Level Parallel computing: 28 processes at PSO level and 10 processes at each particle**
- Converged within 8 iterations in **53 hours**.
- **135 designs** analyzed.

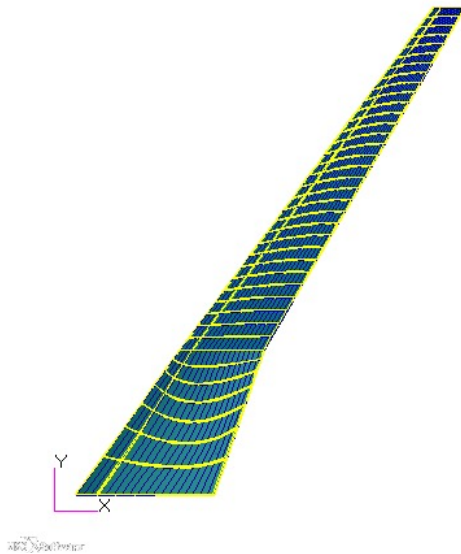


- The corresponding input files are provided with the code.
- The input files of the best obtained design are also provided.



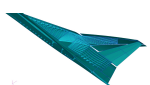
Baseline model: 3 Spars, 10 inner ribs, 25 outer ribs, unstiffened
Wing weight (after size optimization): 19269 lbs (*Qiang et al.*)
Flutter speed: NA

Best design: 3 Spars, 6 inner ribs, 30 outer ribs
Wing weight: 11152.61 lbs (42% improvement)
Flutter speed: 556.53 knots
 $f=2174.99$

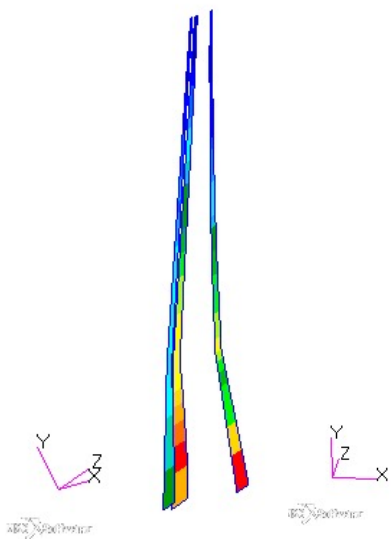


		$X_{\min}$	$X_{\max}$	X_{opt}
DV6	IW spars: Start point spacing	0.2	10	5.52
DV7	IW spars: Control point spacing	0.2	10	3.19
DV8	IW spars: End point spacing	0.2	10	5.75
DV9	IW number of ribs	3	20	6
DV12	IW ribs: Start point spacing	0.2	10	4.33
DV13	IW ribs: Control point spacing	0.2	10	0.69
DV14	IW ribs: End point spacing	0.2	10	4.13
DV17	OW spars: Control point spacing	0.2	10	3.16
DV18	OW spars: End point spacing	0.2	10	6.0
DV19	OW number or ribs	5	40	30
DV22	OW spars: Start point spacing	0.2	10	1.18
DV23	OW spars: Control point spacing	0.2	10	1.56
DV24	OW spars: End point spacing	0.2	10	1.5

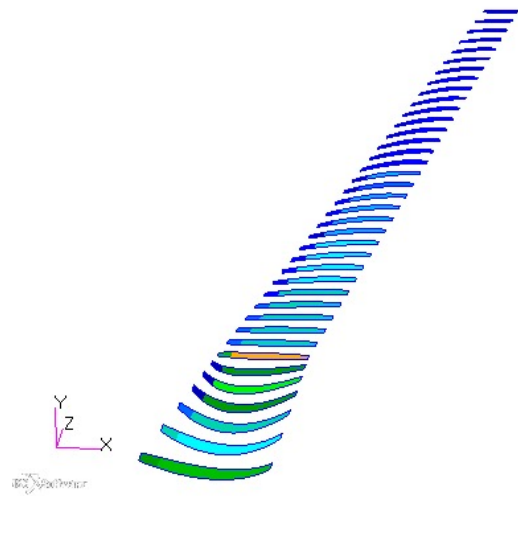
Q. Liu, M. Jrad, S. B. Mulani, and R. K. Kapania, "Global/Local Optimization of Aircraft Wing Using Parallel Processing, AIAA Journal, Vol. 54, No. 11, 2016, pp 3338-3348.



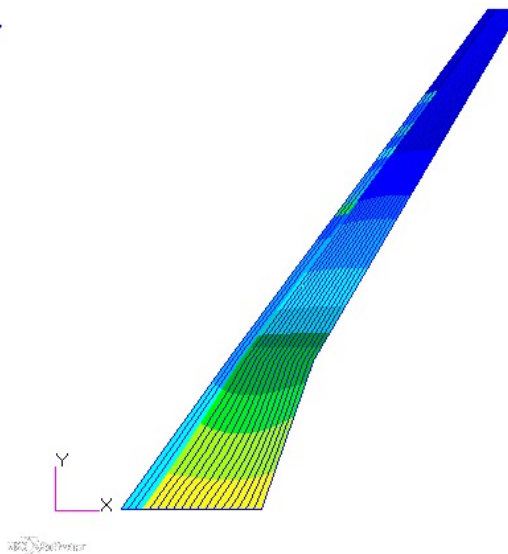
Spars



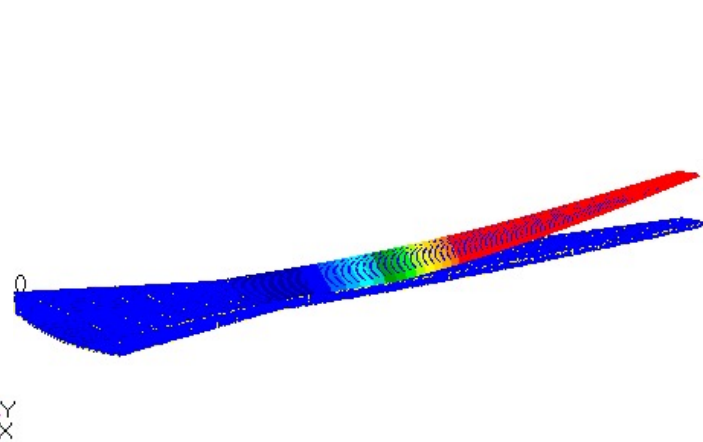
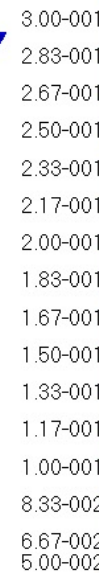
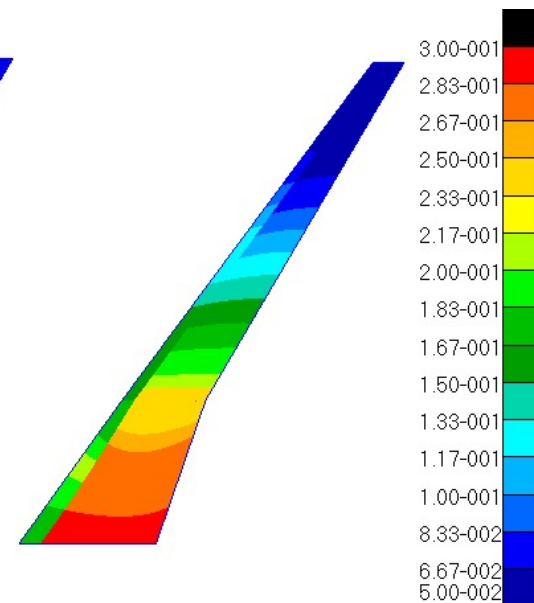
Ribs



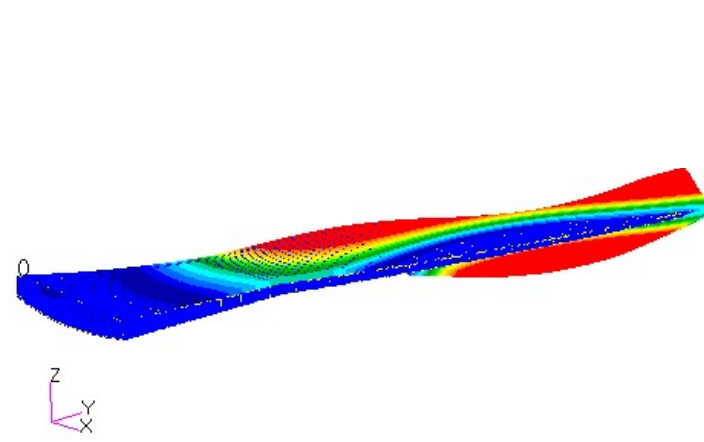
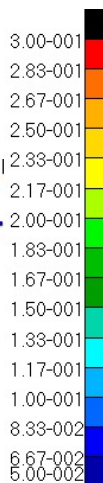
Lower Skin



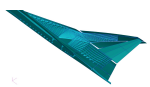
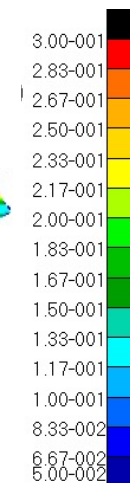
Upper Skin



First Wing Bending Mode (1.23 Hz)



First Wing Torsion Mode (9.45 Hz)



- [1] Taminger, K. M. and Hafley, R. A. 2006. "Electron Beam Freeform Fabrication for Cost Effective Near-Net Shape Manufacturing," in VA:NATO/RTOAVT-139 specialists' meeting on cost effective manufacture via net shape processing. Amsterdam (The Netherlands): NATO Hampton.
- [2] Locatelli, D., Mulani, S. B., and Kapania, R. K. 2011. "Wing-Box Weight Optimization Using Curvilinear Spars and Ribs (SpaRibs)." *Journal of Aircraft* 48:1671-1684.
- [3] Locatelli, D., Tamijani, A. Y., Mulani, S. B., and Kapania, R. K. 2013. "Multidisciplinary Optimization of Supersonic Wing Structures Using Curvilinear Spars and Ribs (SpaRibs)." 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials and Co-located Conferences, Boston, Massachusetts. AIAA 2013-1931.
- [4] Jrad, M., De, S., and Kapania, R. K. 2017. "Global-Local Aeroelastic Optimization of Internal Structure of Transport Aircraft Wing." 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Denver, Colorado.
- [5] Devarajan, B. and Kapania, R.K., 2020. Thermal buckling of curvilinearly stiffened laminated composite plates with cutouts using isogeometric analysis. *Composite Structures*, 238, p.111881.
- [6] Miglani, J., Devarajan, B. and Kapania, R.K., 2018. Thermal buckling analysis of periodically supported composite beams using Isogeometric analysis. In 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (p. 1224).
- [7] Miglani, J., Devarajan, B., & Kapania, R. K. (2021). Isogeometric Thermal Buckling and Sensitivity Analysis of Periodically Supported Laminated Composite Beams. *AIAA Journal*, 1-10.
- [8] Devarajan, B. (2021). Vibration Analysis of Timoshenko Beams using Isogeometric Analysis. arXiv preprint arXiv:2104.12860.
- [9] Devarajan, B. (2021). Free Vibration analysis of Curvilinearly Stiffened Composite plates with an arbitrarily shaped cutout using Isogeometric Analysis. arXiv preprint arXiv:2104.12856.
- [10] Devarajan, B. (2021). Analyzing Thermal Buckling in Curvilinearly Stiffened Composite Plates with Arbitrary Shaped Cutouts Using Isogeometric Level Set Method. arXiv preprint arXiv:2104.05132.
- [11] De, S., Jrad, M., and Kapania, R. K. 2019. "Structural Optimization of Internal Structure of Aircraft Wings with Curvilinear Spars and Ribs." *Journal of Aircraft* 56:707-718. 35. Robinson, J. H., Doyle, S., Ogawa, G., Baker, M., De, S., Jrad, M., & Kapania, R. K. (2016). Aeroelastic Optimization of Wing Structure Using Curvilinear Spars and Ribs (SpaRibs). In 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (p. 3994).



- [12] De, S., Jrad, M., Locatelli, D., Kapania, R. K., and Baker, M. 2017. "SpaRibs Geometry Parameterization for Wings with Multiple Sections using Single Design Space." 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials, Dallas.
- [13] De, S. (2017). Structural Modeling and Optimization of Aircraft Wings having Curvilinear Spars and Ribs (SpaRibs) (Doctoral dissertation, Virginia Tech).
- [14] De, S.; Singh, K.; Seo, J.; Kapania, R.K.; Ostergaard, E.; Angelini, N.; Aguero, R. Lightweight Chassis Design of Hybrid Trucks Considering Multiple Road Conditions and Constraints. World Electr. Veh. J. 2021, 12, 3.
- [15] De, S., Singh, K., Alanbay, B., Kapania, R. K., "Structural Optimization of Truck Front-Frame under Multiple Load Cases," in ASME International Mechanical Engineering Congress & Exposition, Pittsburgh, PA, November 9-15, 2018.
- [16] De, S., Singh, K., Seo, J., Kapania, R. K., "Structural Design and Optimization of Commercial Vehicles Chassis under Multiple Load Cases and Constraints," in 60th AIAA/ ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, San Diego, CA, 2018.
- [17] De, S., Singh, K., Seo, J., Kapania, R. et al., "Unconventional Truck Chassis Design with Multi-Functional Cross Members," SAE Technical Paper 2019-01-0839, 2019

