Manual of EBF3GLWingOpt

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

EBF3GLWingOpt Components

- **Currently, *EBF3GLWingOpt works on Unix/Linux and Windows***
- **For analysis and optimization, following software are required [1-24]**
  - **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***

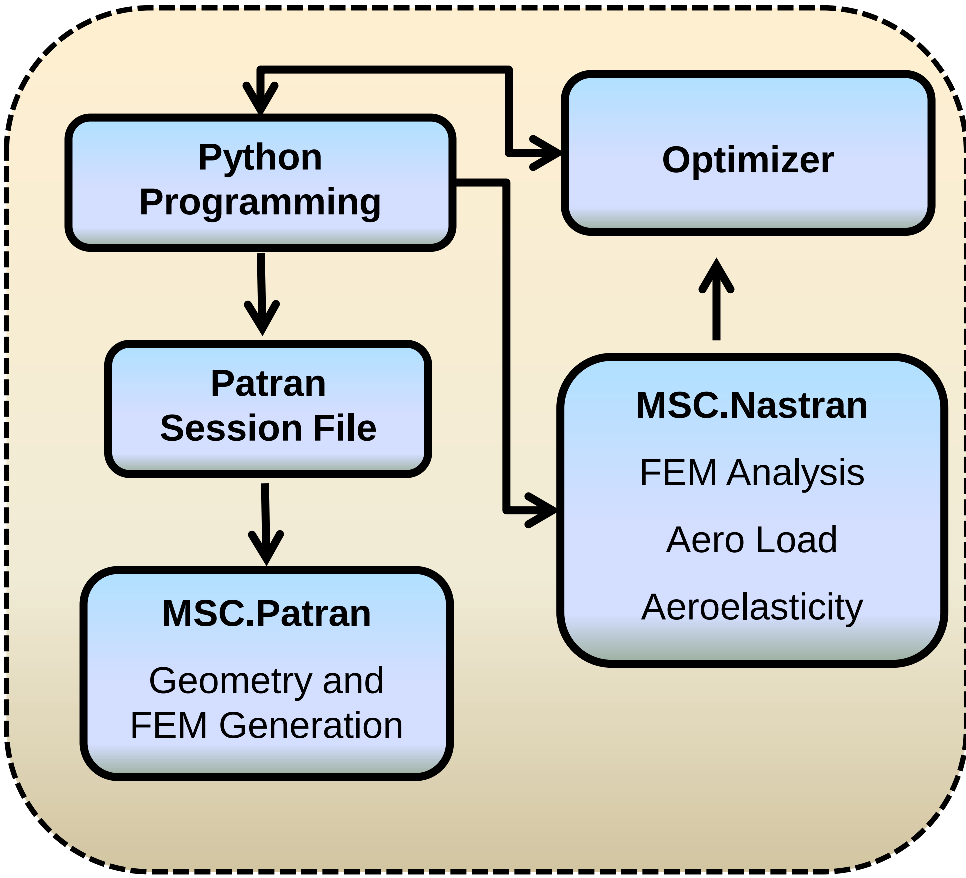
EBF3GLWingOpt : Global Wing Optimization

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

Geometry and FEM Generation

Import Wing Geometry

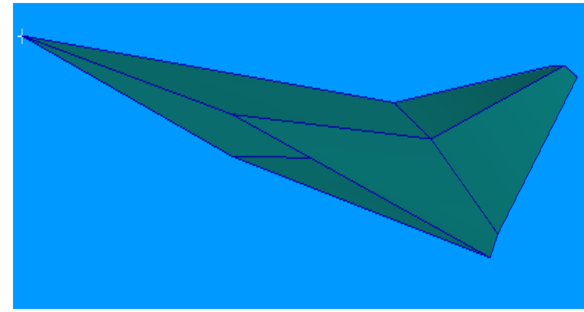
No *SpaRibs*

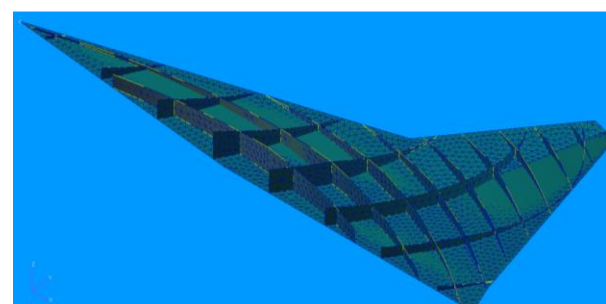
Define SpaRibs in Normalized Space

Project SpaRibs on Actual Wing Box

Create SpaRibs Curves Intersection

Meshing Surfaces and Application of Properties





Design Variables of EBF3GLWingOpt

CRM_Design.txt: Design Variables of EBF3GLWingOpt

Dvar1	Front Spar	Constant Parameters
Dvar2	Rear Spar	

Dvar3 (p_1)	Number of Spars	Inner Wing Spars
Dvar4 (p_2)	SpaRibs control Line start point	
Dvar5 (p_3)	SpaRibs control Line end point	
Dvar6 (p_4)	SpaRibs start point spacing	
Dvar7 (p_5)	SpaRibs control point spacing	
Dvar8 (p_6)	SpaRibs end point spacing	
Dvar9 (p_1)	Number of Ribs	Inner Wing Ribs
Dvar10 (p_2)	SpaRibs control Line start point	
Dvar11 (p_3)	SpaRibs control Line end point	
Dvar12 (p_4)	SpaRibs start point spacing	
Dvar13 (p_5)	SpaRibs control point spacing	
Dvar14 (p_6)	SpaRibs end point spacing	

Dvar15 (p_2)	SpaRibs control Line start point	Outer Wing Spars
Dvar16 (p_3)	SpaRibs control Line end point	
Dvar17 (p_5)	SpaRibs control point spacing	
Dvar18 (p_6)	SpaRibs end point spacing	
Dvar19 (p_1)	Number of Ribs	Outer Wing Ribs
Dvar20 (p_2)	SpaRibs control Line start point	
Dvar21 (p_3)	SpaRibs control Line end point	
Dvar22 (p_4)	SpaRibs start point spacing	
Dvar23 (p_5)	SpaRibs control point spacing	
Dvar24 (p_6)	SpaRibs end point spacing	

Size Variables of EBF3GLWingOpt

Thickness Variables of Nonlinear Thickness Fields

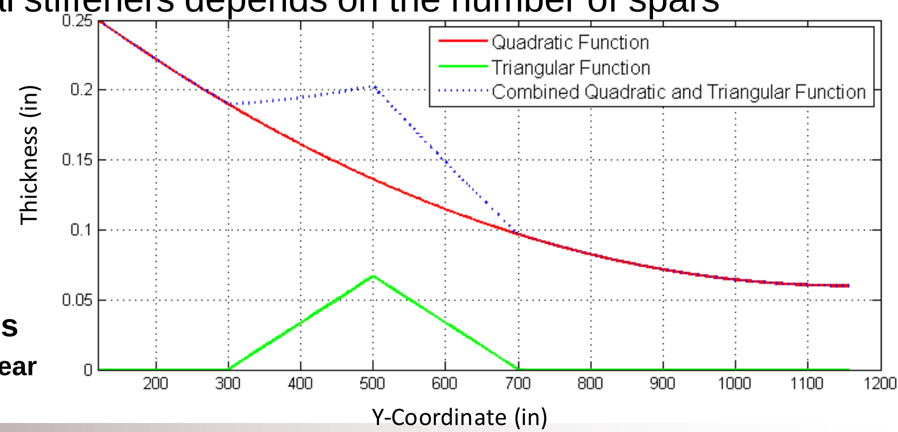
Dvar25	Thickness at Wing Root	Upper Skin
Dvar26	Thickness at Wing Tip	
Dvar27	Non-linear Thickness Parameter	
Dvar28	Thickness at Wing Root	Lower Skin
Dvar29	Thickness at Wing Tip	
Dvar30	Non-linear Thickness Parameter	

Dvar37	Height of Stringers	Stringers
Dvar38	Thickness of Stringers	

Dvar31	Thickness at Wing Root	Spars
Dvar32	Thickness at Wing Tip	
Dvar33	Non-linear Thickness Parameter	
Dvar34	Thickness at Wing Root	Ribs
Dvar35	Thickness at Wing Tip	
Dvar36	Non-linear Thickness Parameter	
Dvar37	Imposed number of longitudinal stiffeners between adjacent spars. Dvar37, Dvar38, etc.	Stiffeners

Note:

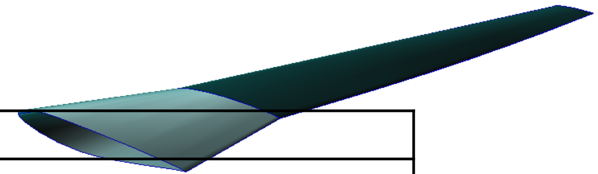
- The number of thickness design variables depends on the input parameter use_field.
- The number of design variables for imposed longitudinal stiffeners depends on the number of spars and on the input parameter ImposeNumbStiff
- The thickness variables are used to create the initial wing design. They 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

Inputs of EBF3GLWingOpt (1)



- Folder ‘**EBF3GLWingOpt**’
  - ‘**CRM_InputFiles**’ and ‘**RUN**’: Input Files of MSC.Nastran

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

Inputs of EBF3GLWingOpt (2)

- ‘**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

Inputs of EBF3GLWingOpt (3)

- 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

Inputs of EBF3GLWingOpt (4)

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.

Inputs of Particle Swarm Optimizer (1)

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.

Inputs of Particle Swarm Optimizer (2)

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

Outputs of EBF3GLWingOpt

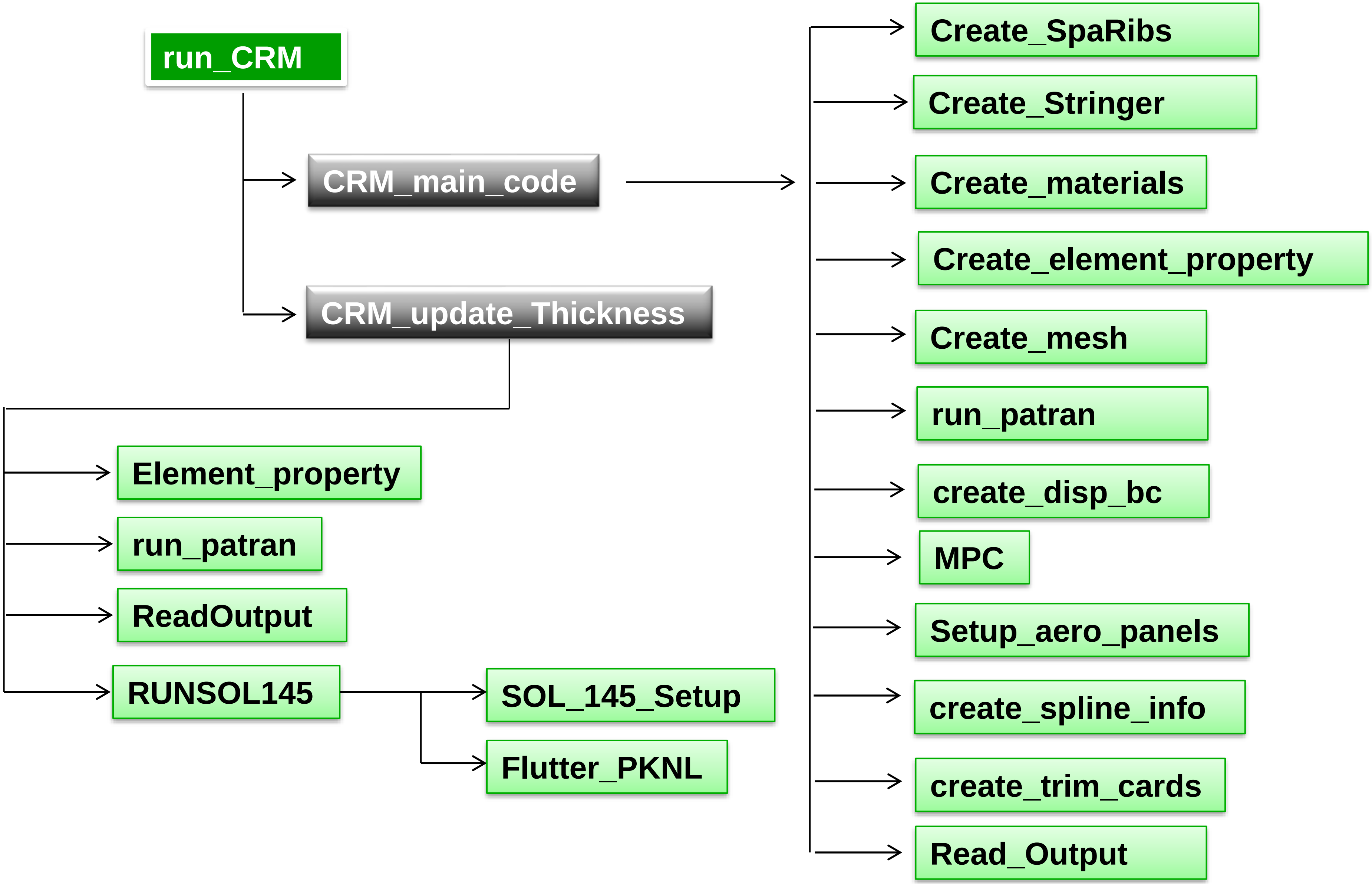
Output Results

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

Analysis Process of EBF3GLWingOpt

- 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

Code Structure of EBF3WingOpt



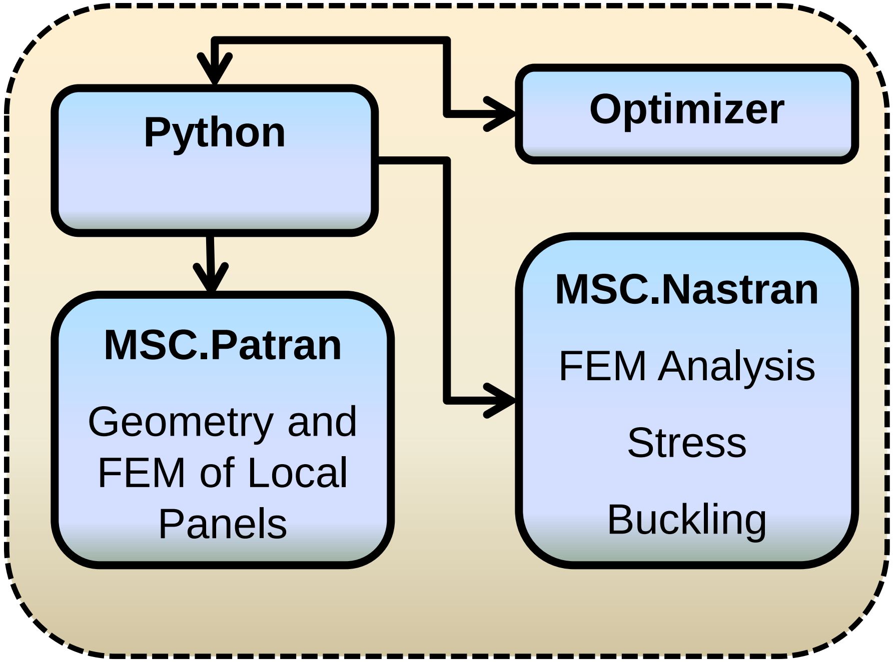
Description of Functions

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

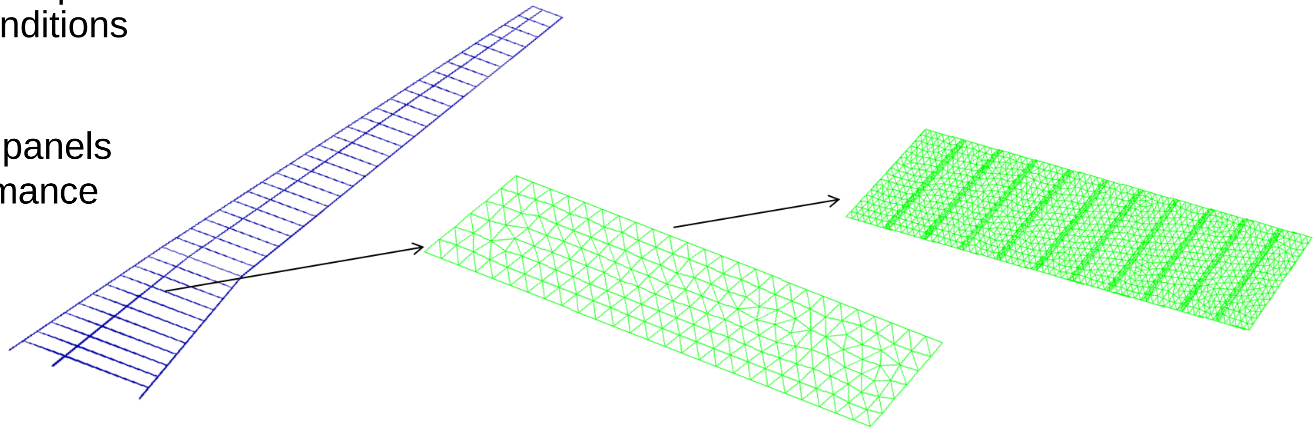
EBF3PanelOpt : Local Panel Optimization

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



Instruction of EBF3PanelOpt

- 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 of Stiffened Panel

➢ 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

<table>
<tr><td>Constant Parameter</td><td>Number of Stiffeners</td><td>Determined by Panel Size</td></tr>
<tr><td>Dvar1</td><td>Panel Thickness</td><td>Uniform Thickness for Each Panel</td></tr>
<tr><td>Dvar2</td><td>Stiffener Height</td><td>Uniform Height for All the Stiffeners on Each Panel</td></tr>
<tr><td>Dvar3</td><td>Stiffener Thickness</td><td>Uniform Thickness for All the Stiffeners on Each Panel</td></tr>
<tr><td>Dvar4 (P_4)</td><td>Stiffeners start point spacing</td><td rowspan="3">Option 1: DV4 ~ DV6 are Constant Parameters of Straight Stiffeners

Option 2: DV4 ~ DV6 are Design Variables of Curvilinear Stiffeners</td></tr>
<tr><td>Dvar5 (P_5)</td><td>Stiffeners control point spacing</td></tr>
<tr><td>Dvar6 (P_6)</td><td>Stiffeners end point spacing</td></tr>
</table>

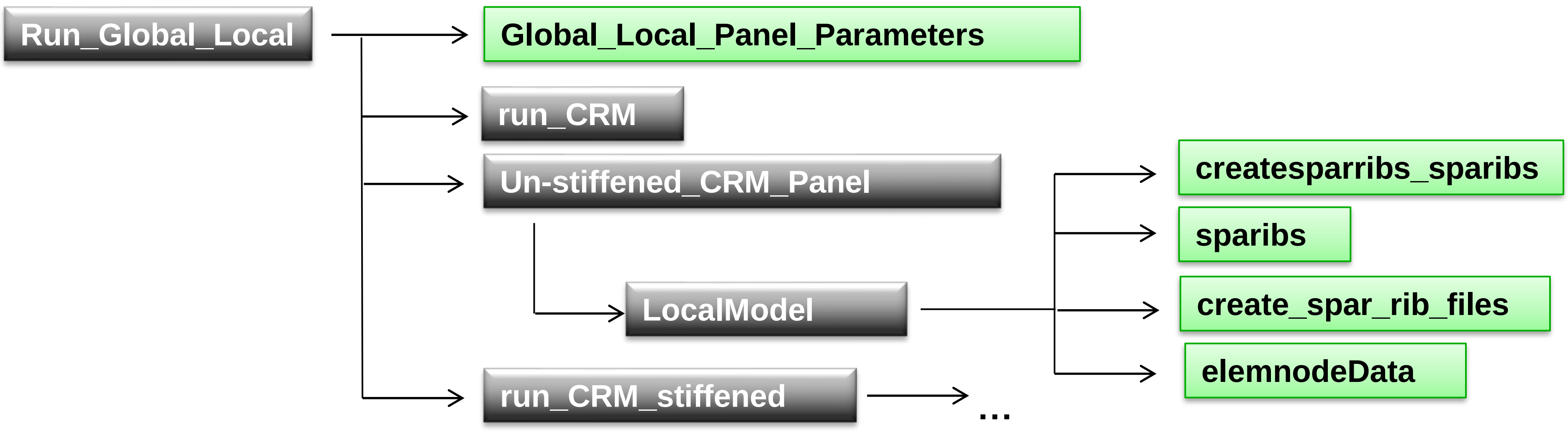
The Geometry of Wing Skins, Spars and Ribs Remain Constant in the Panel Optimization

Input Files of EBF3PanelOpt

- 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

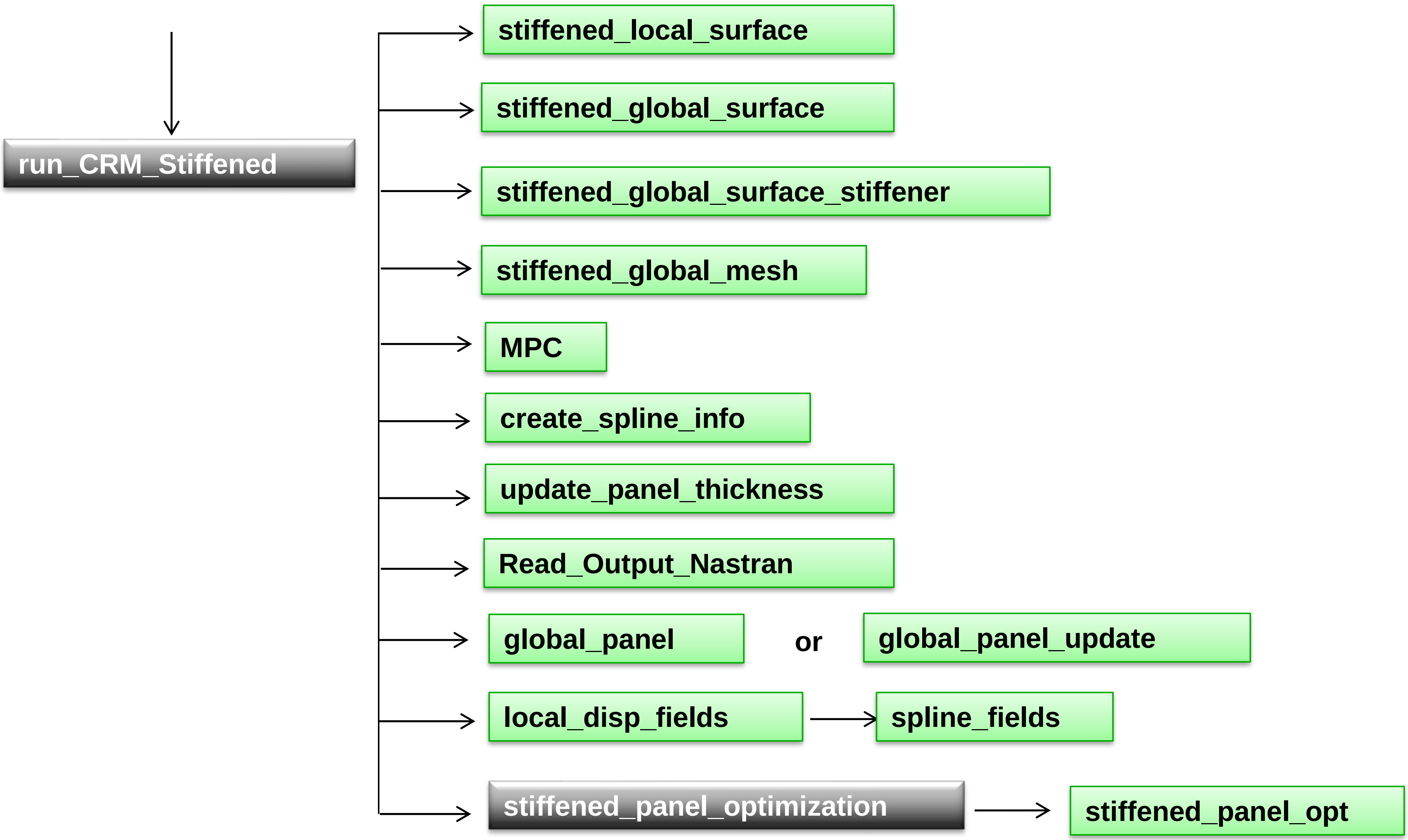
- Folder ‘Local_Panel_ini’:
  - ‘EBF3_Python’ Folder: Input Files of Single Stiffened Panel Analysis
  - ‘RUN’ Folder: Nastran Input Files ‘aero_cards.bdf’, ‘trim_cards.bdf’, ‘flutter_cards’, ‘sol*.bdf’
  - ‘panelinfo.mat’: XYZ Coordinates of the Vertices of Aerodynamic Panels

Code Structure of EBF3PanelOpt (1)



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

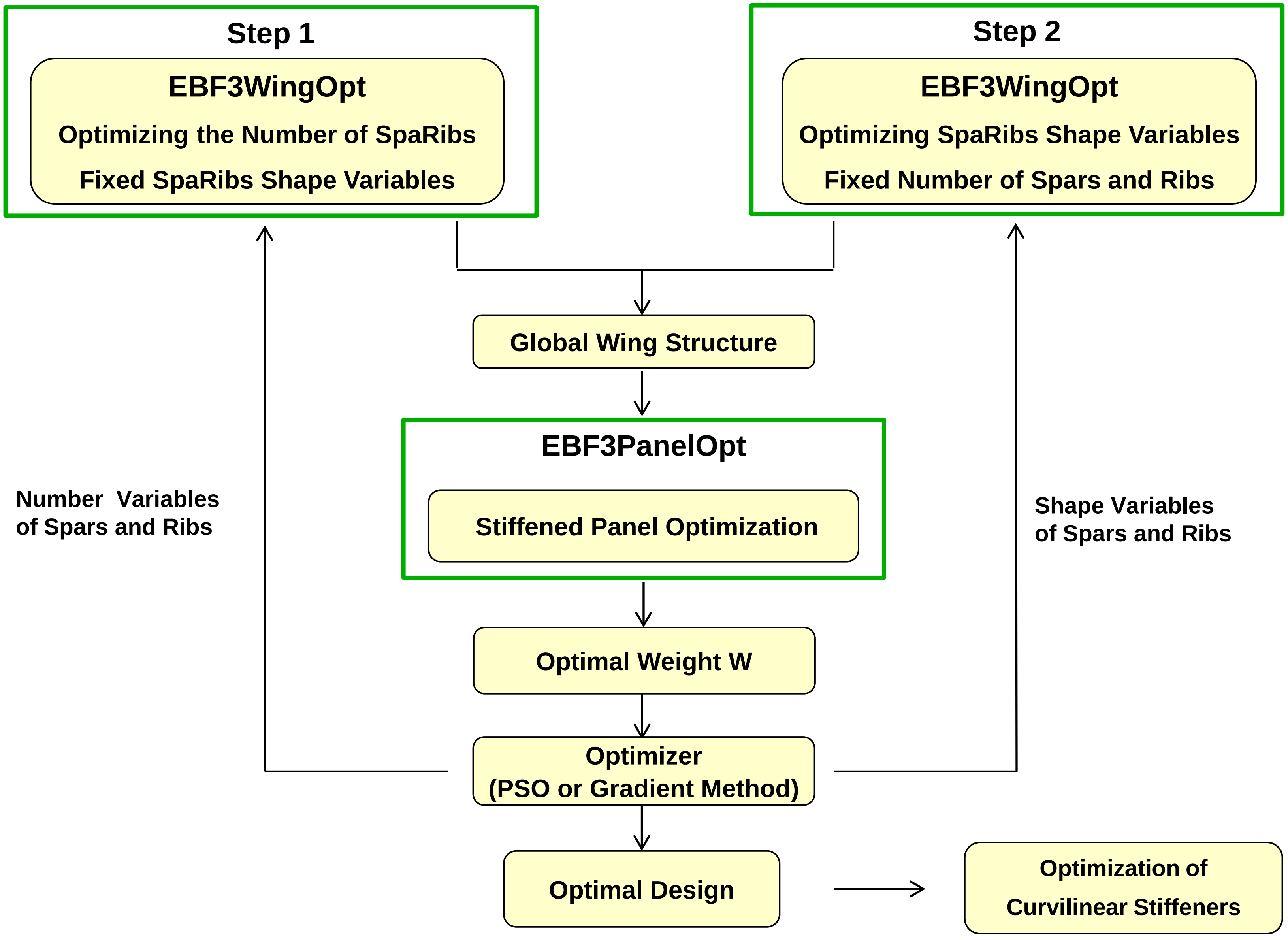
Code Structure of EBF3PanelOpt (2)



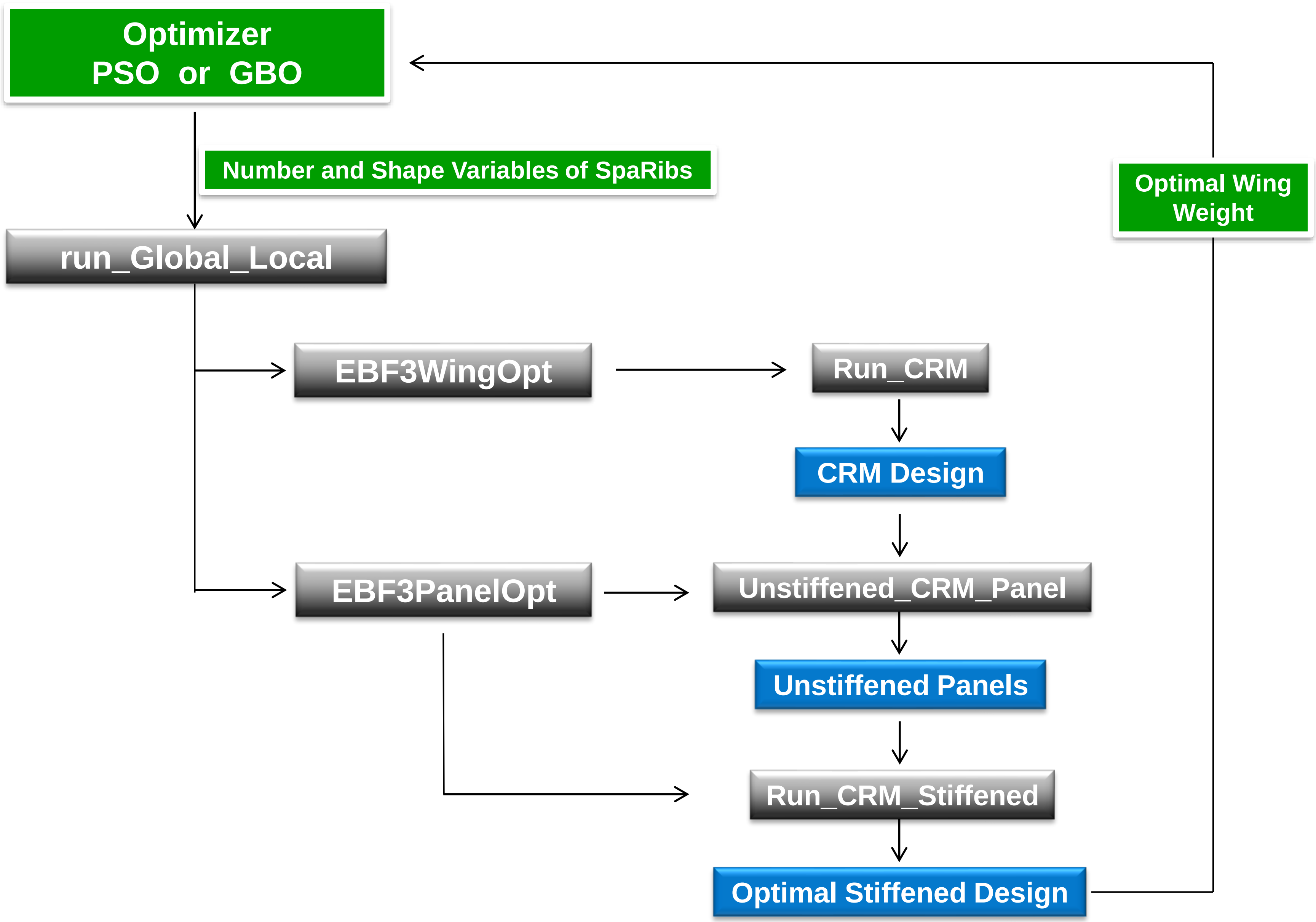
Description of Functions

Function Name	Description of Functions in *EBF3PanelOpt*
run_CRM_stiffened	Optimizing the CRM wing with stiffened panels
stiffened_local_surface	Create the Patran database for each local panel
stiffened_local_surface_parallel	Parallel processing of *stiffened_local_surface*
stiffened_local_surface_single	Single process computing of *stiffened_local_surface*
stiffened_global_surface	Create a surface for each local panel
stiffened_global_surface_stiffener	Add the stiffeners on the local panels
stiffened_global_mesh	Mesh the stiffened wing using fine grids
MPC	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.
create_spline_info	Write the spline nodes into Nastran input file
update_panel_thickness	Update the panel thickness in optimization process
Read_Output_Nastran	Read the displacement data from the Nastran result file of SOL 101
create_panel_bdf	Create the Nastran input files of local panels
create_panel_bdf_parallel	Parallel processing of *create_panel_bdf*
create_panel_bdf_single	Single process computing of *stiffened_panel_optimization*
global_panel	Create the Nastran input files of local panels
local_disp_fields	Write the displacement data of local panels into Nastran input files
spline_fields	Read the displacement data from local panel results
stiffened_panel_optimization	Optimize the stiffened or unstiffened local panels in the aircraft wing
stiffened_panel_optimization_parallel	Parallel processing of *stiffened_panel_optimization*
stiffened_panel_optimization_single	Single process computing of *stiffened_panel_optimization*
stiffened_panel_opt	Optimize the panel thickness of a single stiffened or unstiffened panel

EBF3GLWingOpt Framework



EBF3GLWingOpt Framework



Description of Functions

Function Name	**Description of Functions in EBF3GLWingOpt**
pso	Particle swarm optimization with continuous and integer design variables
	Inertia_weight: Inertia weight to prevent velocities becoming too large
	C1: Scaling co-efficient on the social component
	C2: Scaling co-efficient on the cognitive component
	Dimension: number of the design variables
	Type_DV: type of design variable: integer or real. For example Type_DV = [‘real’, ‘real’, ‘int’]
	Ub_DV: upper boundary of design variables
	Lb_DV: lower boundary of design variables
	Iterations: maximum number of iterations
	swarmSize: population size of particle swarm in each iteration
	Convergence_iter: check the convergence in consecutive iterations
	Convergence_relative: use the relative change of objective function as convergence criteria
gradient	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

Design Variables of EBF3GLWingOpt v6.0.1

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:

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

Dvar54	Stiffener thickness for panel thickness 0.15 -0.2 in
Dvar55	Stiffener thickness for panel thickness 0.2-0.3 in
Dvar56	Stiffener thickness for panel thickness 0.3 in

- **Offset Parameters:**

O11, O12, O21, O22, O31, O32

- **Thickness Increment Function (TIF)**

$Dmax_G$= Maximum displacement in eigenvector

$Dmax_P$ = Maximum displacement in a panel

$$m = \frac{Dmax_P}{Dmax_G}$$

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

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

Inputs of EBF3GLWingOpt v6.0.1 (1)

- '**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

Inputs of EBF3GLWingOpt v6.0.1 (2)

- Input Parameters in ‘**input_data.txt**’
  Format: Parameter=value1,value2,…

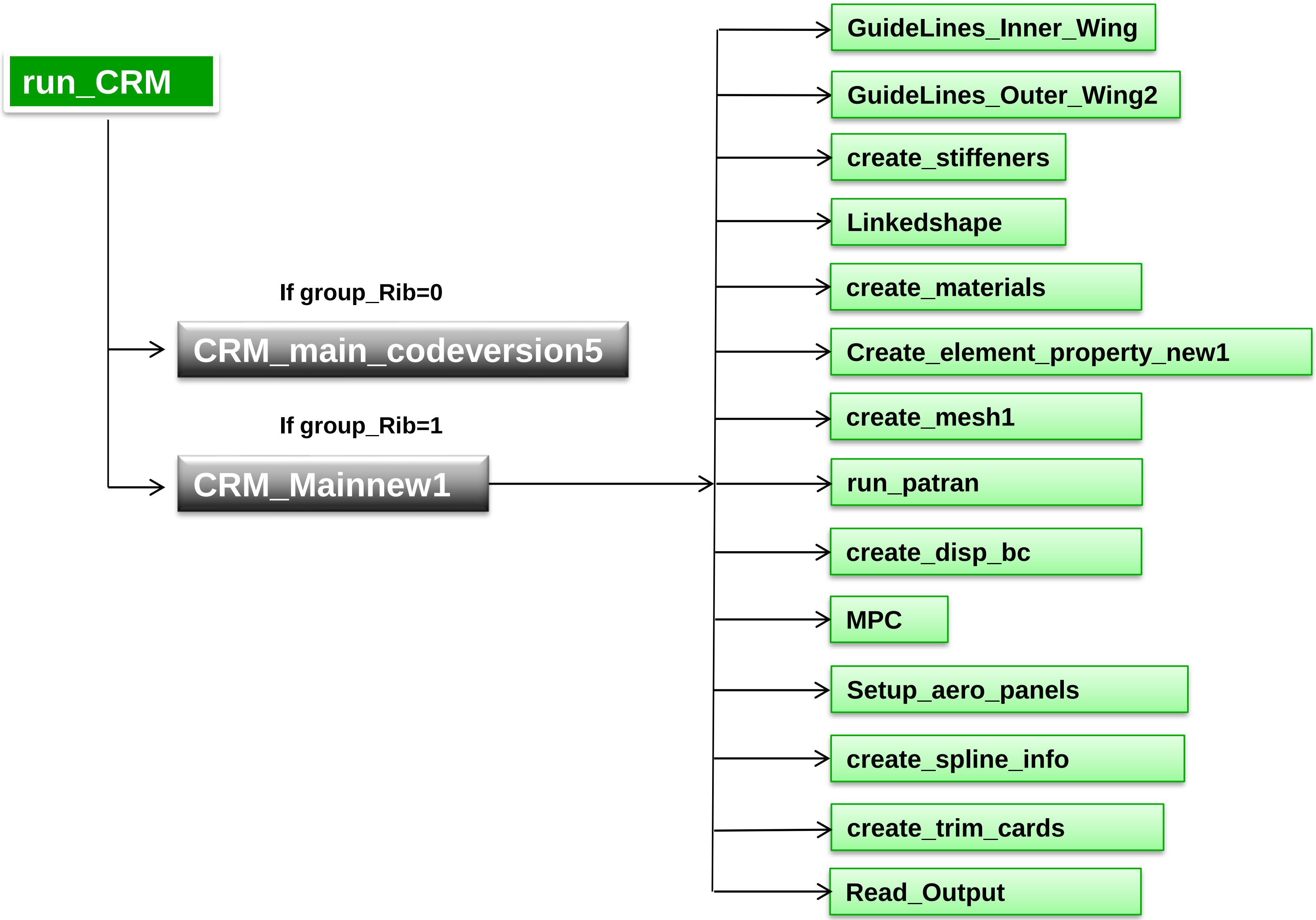
Most parameters are common as those for **EBF3GLWingOpt v5.3.8**
The list of additional parameters for **EBF3GLWingOpt v6.0.1** given below

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)

Inputs of EBF3GLWingOpt v6.0.1 (3)

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

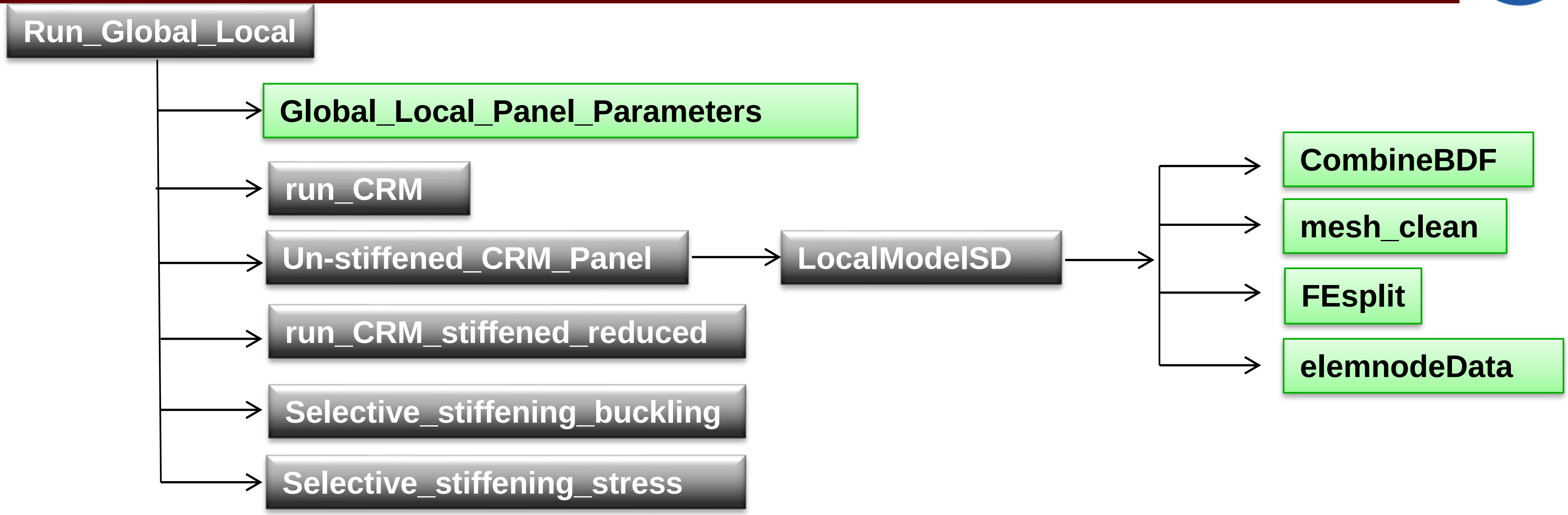
Code Structure of EBF3GLWingOpt v6.0.1



Description of Functions

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

Code Structure of EBF3GLWingOpt v6.0.1



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

Parallel Computing and Installation

Parallel Framework: Setup Requirements

- 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 openmipi-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 openmipi-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

Instructions for Running the Code (1)

- **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

Instructions for Running the Code (2)

- **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

Instructions for Running the Code (3)

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

PSO Parameters

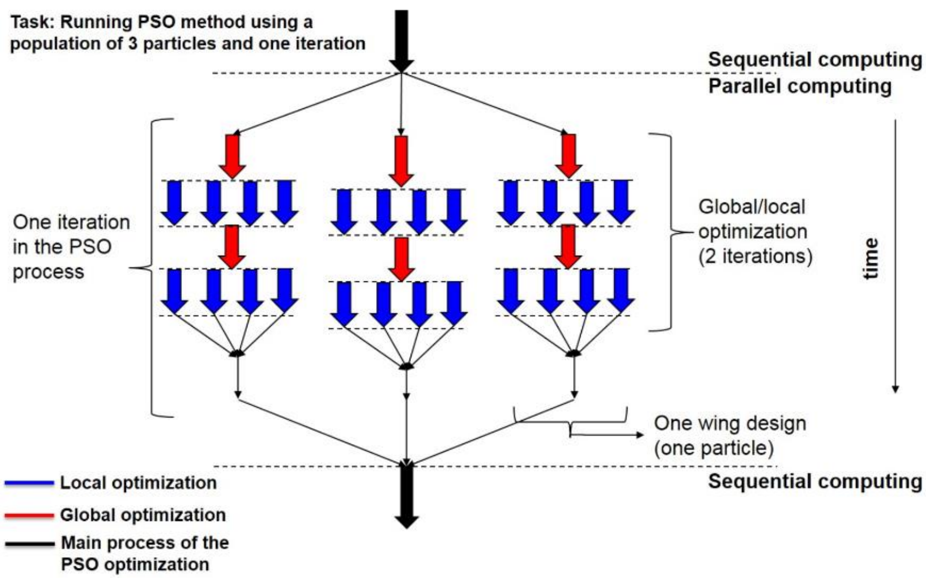
- NASA Common Research Model
  - Number of the design variables: 13
  - Population size of particle swarm: 28
  - 3 spars

W: Weight (lbs)
V_f: Flutter Speed (knots)
β_w: Mass penalty (1/lbs)
β_V: Flutter velocity penalty (knots)

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

β_w=0.1/lbs
β_V=25.10^3 $\sqrt{\text{knots}}$

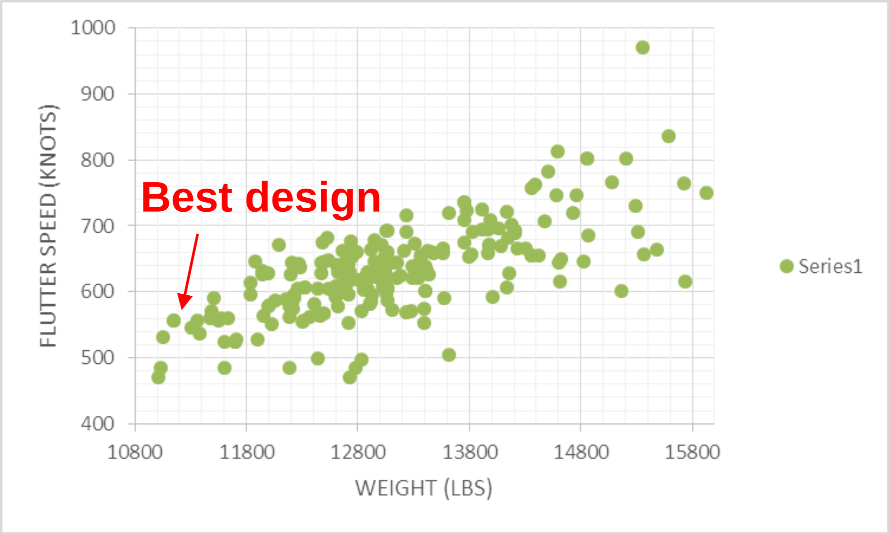
Double Level Parallel computing:
PSO level: **28 processes**
Each particle: **10 processes**

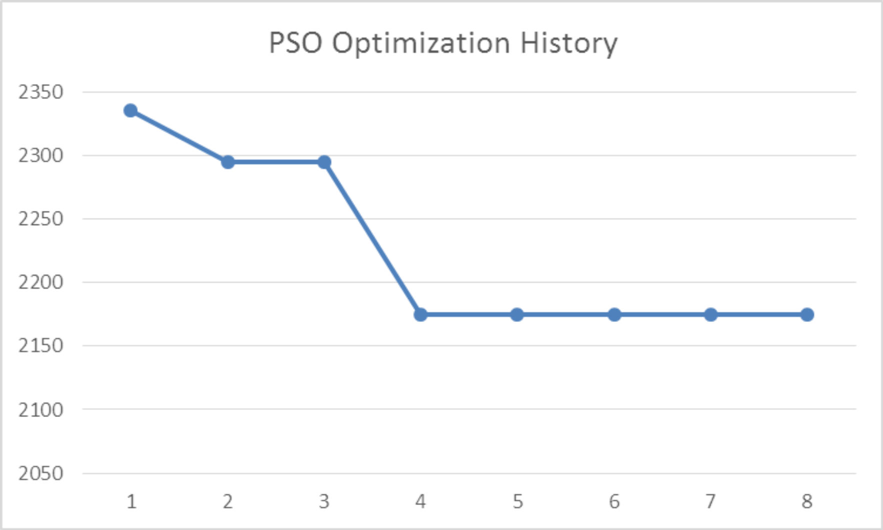


		X_{min}	X_{max}
DV6	IW spars: Start point spacing	0.2	10
DV7	IW spars: Control point spacing	0.2	10
DV8	IW spars: End point spacing	0.2	10
DV9	IW number of ribs	3	20
DV12	IW ribs: Start point spacing	0.2	10
DV13	IW ribs: Control point spacing	0.2	10
DV14	IW ribs: End point spacing	0.2	10
DV17	OW spars: 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 Results

- 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.



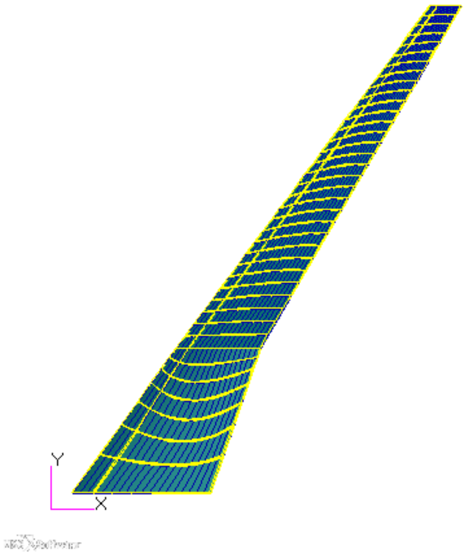


- The corresponding input files are provided with the code.
- The input files of the best obtained design are also provided.

PSO Best Design

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

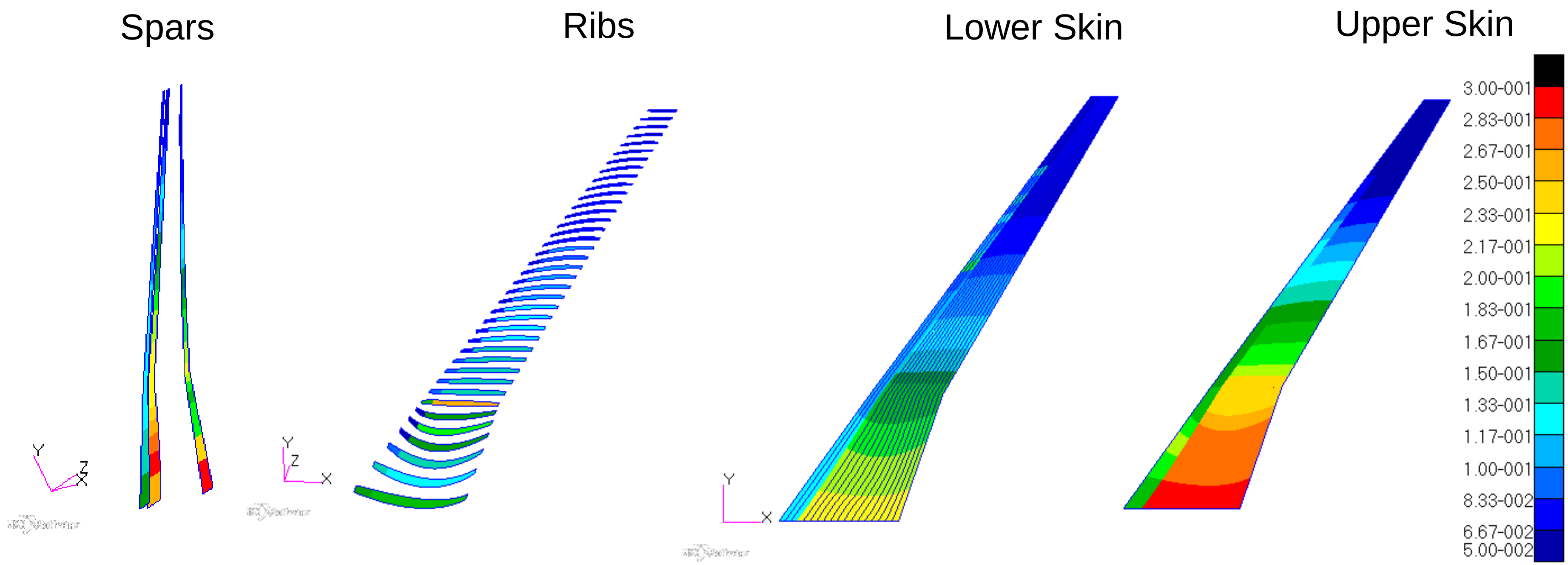
Best design: 3 Spars, 6 inner ribs, 30 outer ribs
Wing weight: **11152.61 lbs (42% improvement)**
Flutter speed: **556.53 knots**
f=2174.99

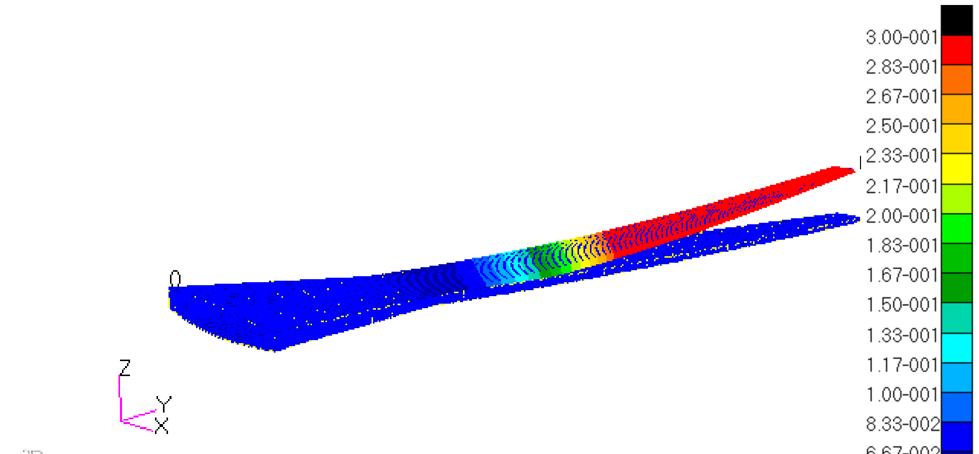


		X_{min}	X_{max}	X_{opt}
DV6	IW spars: Start point spacing	0.2	10	**5.52**
DV7	IW spars: Control point spacing	0.2	10	**3.19**
DV8	IW spars: End point spacing	0.2	10	**5.75**
DV9	IW number of ribs	3	20	**6**
DV12	IW ribs: Start point spacing	0.2	10	**4.33**
DV13	IW ribs: Control point spacing	0.2	10	**0.69**
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**

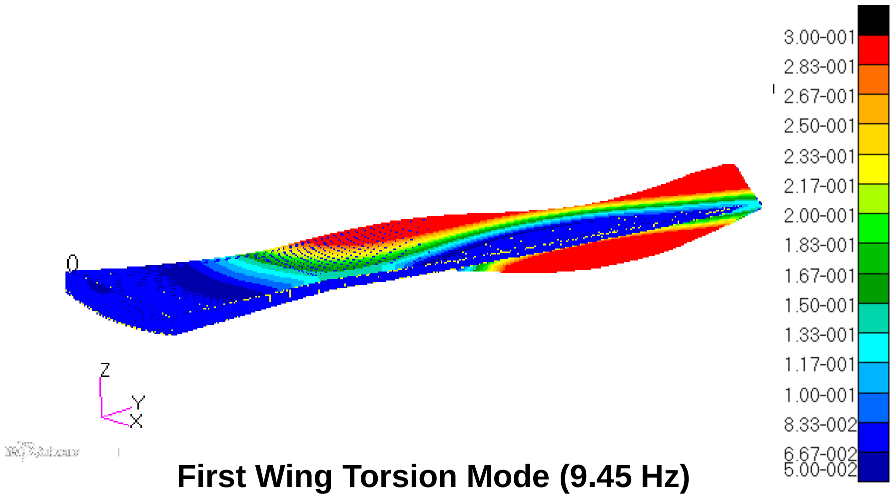
Q. Liu, M. Jrad, S. B. Mulani, and R. K. Kapania, "Global/Local Optimization of Aircraft Wing Using Parallel Processing, AIAA Journal, Vol. 54, No. 11, 2016, pp 3338-3348.

Thickness Distribution and Vibration Modes





First Wing Bending Mode (1.23 Hz)



First Wing Torsion Mode (9.45 Hz)

References

1. Jrad, M., De, S. and Kapania, R.K., 2017. Global-Local Aeroelastic Optimization of Internal Structure of Transport Aircraft Wing. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (p. 4321).
2. 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*(2), pp.707-718.
3. Robinson, J.H., Doyle, S., Ogawa, G., Baker, M., De, S., Jrad, M. and 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).
4. De, S., White, J., Brusuelas, T., Patton, C., Koh, A. and Huang, Q., 2020. Electrochemical behavior of protons and cupric ions in water in salt electrolytes with alkaline metal chloride. *Electrochimica Acta*, *338*, p.135852.
5. 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. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (p. 0570).
6. Biswas, S., Saha, D., De, S., Cobb, A.D., Das, S. and Jalaian, B.A., 2021, June. Improving differential evolution through bayesian hyperparameter optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC)* (pp. 832-840). IEEE.
7. De, S., Sides, W.D., Brusuelas, T. and Huang, Q., 2020. Electrodeposition of superconducting rhenium-cobalt alloys from water-in-salt electrolytes. *Journal of Electroanalytical Chemistry*, *860*, p.113889.
8. De, S., Singh, K., Alanbay, B., Kapania, R.K. and Aguero, R., 2018, November. Structural Optimization of Truck Front-Frame Under Multiple Load Cases. In *ASME International Mechanical Engineering Congress and Exposition* (Vol. 52187, p. V013T05A039). American Society of Mechanical Engineers.
9. De, S., Singh, K., Seo, J., Kapania, R., Aguero, R., Ostergaard, E. and Angelini, N., 2019. *Unconventional Truck Chassis Design with Multi-Functional Cross Members* (No. 2019-01-0839). SAE Technical Paper.
10. De, S. and Huang, Q., 2021, May. Mathematical Modeling of Cyclic Voltammogram Curves of Copper Deposition Involving Multiple Additives. In *ECS Meeting Abstracts* (No. 28, p. 977). IOP Publishing.
11. De, S., Singh, K., Seo, J., Kapania, R.K., Ostergaard, E., Angelini, N. and Aguero, R., 2021. Lightweight Chassis Design of Hybrid Trucks Considering Multiple Road Conditions and Constraints. *World Electric Vehicle Journal*, *12*(1), p.3.
12. De, S. and Huang, Q., 2020, November. Estimation of Kinetic Parameters of Additives By Semi-Analytical Solution. In *ECS Meeting Abstracts* (No. 17, p. 1483). IOP Publishing.
13. Huang, Q., Sides, W.D., De, S. and Hu, Y., 2019, September. Electrodeposition of Cu, Co and Re from Water in Salt Electrolytes. In *ECS Meeting Abstracts* (No. 46, p. 2116). IOP Publishing.
14. De, S., 2017. *Structural Modeling and Optimization of Aircraft Wings having Curvilinear Spars and Ribs (SpaRibs)* (Doctoral dissertation, Virginia Tech).
15. De, S., Jrad, M., Locatelli, D., Kapania, R.K., Baker, M. and Pak, C.G., 2017. *SpaRibs Geometry Parameterization for Wings with Multiple Sections using Single Design* (No. AFRC-DAA-TN37932).
16. 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.
17. 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).
18. Li, Q., Devarajan, B., Zhang, X., Burgos, R., Boroyevich, D. and Raj, P., 2016. *Conceptual design and weight optimization of aircraft power systems with high-peak pulsed power loads* (No. 2016-01-1986). SAE Technical Paper.
19. Devarajan, B. and Kapania, R.K., 2022. Analyzing thermal buckling in curvilinearly stiffened composite plates with arbitrary shaped cutouts using isogeometric level set method. *Aerospace Science and Technology*, p.107350.
20. Devarajan, B., 2019. *Thermomechanical and vibration analysis of stiffened unitized structures and threaded fasteners* (Doctoral dissertation, Virginia Tech).
21. Miglani, J., Devarajan, B. and Kapania, R.K., 2021. Isogeometric Thermal Buckling and Sensitivity Analysis of Periodically Supported Laminated Composite Beams. *AIAA Journal*, pp.1-10.
22. Devarajan, B., 2021. Vibration Analysis of Timoshenko Beams using Isogeometric Analysis. *arXiv preprint arXiv:2104.12860*.
23. Devarajan, B., 2021. Free Vibration analysis of Curvilinearly Stiffened Composite plates with an arbitrarily shaped cutout using Isogeometric Analysis. *arXiv preprint arXiv:2104.12856*.
24. Devarajan, B., Locatelli, D., Kapania, R.K. and Meritt, R.J., 2016. Thermo-Mechanical Analysis and Design of Threaded Fasteners. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (p. 0579)