SU² Design Exploration
NACA 0012

SU² Release Version 2.0 Workshop
Stanford University
Tuesday, January 15th, 2013

Trent Lukaczyk
Department of Aeronautics & Astronautics
Stanford University
Indirect Design Approaches

- Example: Response Surface Modeling
- Two Hicks-Henne Bump Functions
- 10x10 grid of simulations
Indirect Design Approaches

- Example: Response Surface Modeling
- Two Hicks-Henne Bump Functions
- 10x10 grid of simulations
Task-Based Design Exploration

Project

Root folder of data
Baseline config and mesh

Config Change

Job 0
Contains analysis tasks
Output meshes, restarts

Job 1

Job 2

Config Change

Config Change

* Only design variables for now

Task
Deformation

Task
Direct Solution

Task
Drag Adjoint

Task
Lift Adjoint

Jan 15th, 2013

SU2
Release
Version 2.0
Workshop
NACA0012 Test Case

- NACA 0012 Test Case
- One Hicks-Henne Bump Function
- 11 Evaluations in $X \in [-0.02, 0.02]$
% ------------------------- EVALUATE PROJECT DEFINITION -------------------------%

% List of tasks to complete
TASKS= DEFORM, DIRECT, CONT_ADJOINT

% Number of partitions (0 for Serial)
NUMBER_PART= 2

% List of design variables (Design variables are separated by semicolons)
% - HICKS_HENNE: ( 1, Scale | Mark. List | Lower(0)/Upper(1) side, x_Loc )
DEFINITION_DV= ( 1, 1.0 | airfoil | 0, 0.05 ); ( 1, 1.0 | airfoil | 0, 0.10 ); (...)

% Gradients to calculate
GRADIENTS= LIFT, DRAG, MOMENT_Z

% Console output (VERBOSE, CONCISE, QUIET)
CONSOLE= CONCISE
% ------------------------- EVALUATE PROJECT DEFINITION -------------------------%

% List of tasks to complete
TASKS= DEFORM, DIRECT, CONT_ADJOINT

% Number of partitions (0 for Serial)
NUMBER_PART= 2

% List of design variables (Design variables are separated by semicolons)
% - HICKS_HENNE: ( 1, Scale | Mark. List | Lower(0)/Upper(1) side, x_Loc )
DEFINITION_DV= ( 1, 1.0 | airfoil | 0, 0.05 ); ( 1, 1.0 | airfoil | 0, 0.10 ); (...)

% Gradients to calculate
GRADIENTS= LIFT, DRAG, MOMENT_Z

% Console output (VERBOSE, CONCISE, QUIET)
CONSOLE= CONCISE
config_NACA0012.cfg

% ------------------------- EVALUATE PROJECT DEFINITION ------------------------%

% List of tasks to complete
TASKS= DEFORM, DIRECT, CONT_ADJOINT

% Number of partitions (0 for Serial)
NUMBER_PART= 2

% List of design variables (Design variables are separated by semicolons)
% - HICKS_HENNE: ( 1, Scale | Mark. List | Lower(0)/Upper(1) side, x_Loc )
DEFINITION_DV= ( 1, 1.0 | airfoil | 0, 0.05 ); ( 1, 1.0 | airfoil | 0, 0.10 ); (...)

% Gradients to calculate
GRADIENTS= LIFT, DRAG, MOMENT_Z

% Console output (VERBOSE, CONCISE, QUIET)
CONSOLE= CONCISE
% ------------------------- EVALUATE PROJECT DEFINITION ---------------------------%

% List of tasks to complete
TASKS= DEFORM, DIRECT, CONT_ADJOINT

% Number of partitions (0 for Serial)
NUMBER_PART= 2

% List of design variables (Design variables are separated by semicolons)
% - HICKS_HENNE: ( 1, Scale | Mark. List | Lower(0)/Upper(1) side, x_Loc )
DEFINITION_DV= ( 1, 1.0 | airfoil | 0, 0.05 ); ( 1, 1.0 | airfoil | 0, 0.10 ); (...)

% Gradients to calculate
GRADIENTS= LIFT, DRAG, MOMENT_Z

% Console output (VERBOSE, CONCISE, QUIET)
CONSOLE= CONCISE
% EVALUATE PROJECT DEFINITION

**List of tasks to complete**

```
TASKS= DEFORM, DIRECT, CONT_ADJOINT
```

**Number of partitions (0 for Serial)**

```
NUMBER_PART= 2
```

**List of design variables (Design variables are separated by semicolons)**

```
% - HICKS_HENNE: ( 1, Scale | Mark. List | Lower(0)/Upper(1) side, x_Loc )
DEFINITION_DV= ( 1, 1.0 | airfoil | 0, 0.05 ); ( 1, 1.0 | airfoil | 0, 0.10 ); (...)
```

**Gradients to calculate**

```
GRADIENTS= LIFT, DRAG, MOMENT_Z
```

**Console output (VERBOSE, CONCISE, QUIET)**

```
CONSOLE= CONCISE
```
% EVALUATE PROJECT DEFINITION

% List of tasks to complete
TASKS= DEFORM, DIRECT, CONT_ADJOINT

% Number of partitions (0 for Serial)
NUMBER_PART= 2

% List of design variables (Design variables are separated by semicolons)
% - HICKS_HENNE: ( 1, Scale | Mark. List | Lower(0)/Upper(1) side, x_Loc )
DEFINITION_DV= ( 1, 1.0 | airfoil | 0, 0.05 ); ( 1, 1.0 | airfoil | 0, 0.10 ); (...)

% Gradients to calculate
GRADIENTS= LIFT, DRAG, MOMENT_Z

% Console output (VERBOSE, CONCISE, QUIET)
CONSOLE= CONCISE
# Setup

(...)  

# design variable values

DV_vals = numpy.linspace(-0.02, 0.02, 11)

# setup config changes

config_delta = []

for X in DV_vals:
    DV_X = numpy.zeros(n_DV)
    DV_X[i_DV] = X
    config_delta.append( {'VARIABLES':DV_X} )

# initialize project

The_Project = Project( config_name = config_filename, 
                        design_name = design_filename  )
# Setup

(…)

# design variable values

```python
DV_vals = numpy.linspace(-0.02, 0.02, 11)
```

# setup config changes

```python
config_delta = []
for X in DV_vals:
    DV_X = numpy.zeros(n_DV)
    DV_X[i_DV] = X
    config_delta.append( {'VARIABLES':DV_X} )
```

# initialize project

```python
The_Project = Project( config_name = config_filename, design_name = design_filename )
```
# Setup

(…)

# design variable values

\[ DV\_vals = \text{numpy.linspace}(-0.02, 0.02, 11) \]

# setup config changes

\[ \text{config\_delta} = [] \]

for \( X \) in \( DV\_vals \):

\[ DV\_X = \text{numpy.zeros}(n\_DV) \]

\[ DV\_X[i\_DV] = X \]

\[ \text{config\_delta}.\text{append}( \{'\text{VARIABLES'}:DV\_X\} ) \]

# initialize project

\[ \text{The} \_\text{Project} = \text{Project}( \text{config\_name} = \text{config\_filename}, \]

\[ \text{design\_name} = \text{design\_filename} ) \]
# Setup

(...) 

# design variable values
DV_vals = numpy.linspace(-0.02, 0.02, 11)

# setup config changes
config_delta = []
for X in DV_vals:
    DV_X = numpy.zeros(n_DV)
    DV_X[i_DV] = X
    config_delta.append( {'VARIABLES':DV_X} )

# initialize project
The_Project = Project( config_name = config_filename,
                       design_name = design_filename  )

run_project.py
# Run Project

# evaluate project
design_new,_,_ = The_Project.evaluate(config_delta)

# save project
libSU2.save_data(project_filename,The_Project)

# save data
# already done by The_Project.evaluate()