

TERMA

Steen Lauridsen

stl@terma.com

Fulldautomatisk Optimering

DANSIS presentation

30 marts 2005



- ▶ Motivation
- ▶ Optimization loop
- ▶ Gambit/SolidWorks
 - ▶ Preparation of geometry in SolidWorks
 - ▶ Visual Basic routines
 - ▶ Generation of mesh files
- ▶ Fluent
- ▶ Matlab
 - ▶ Automatic scanning of Fluent output

- ▶ Closure



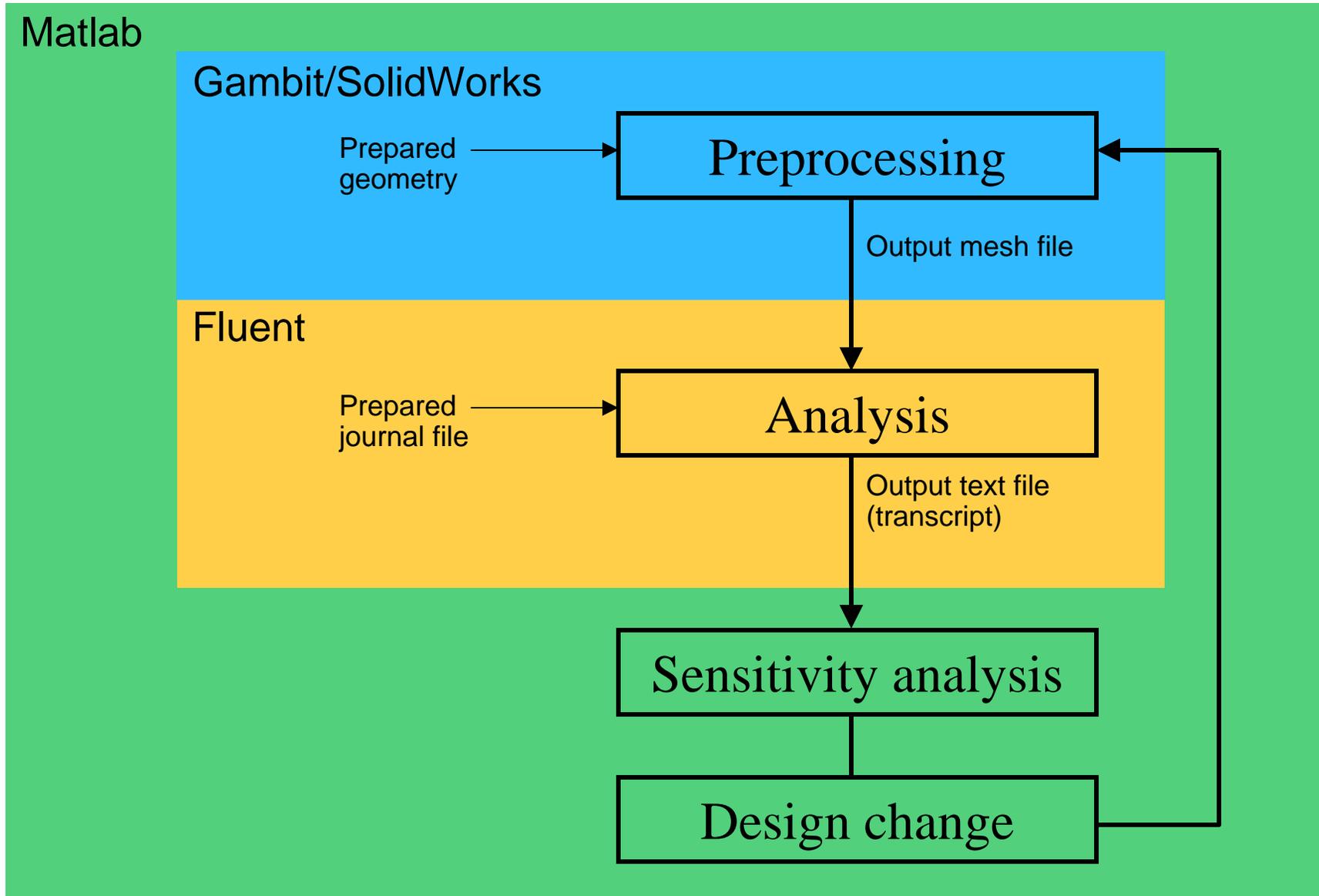
- ▶ No current major CFD vendor are providing sensitivity analysis as part of their product
- ▶ The required products e.g.
 - ▶ SolidWorks/Catia V5
 - ▶ Matlab

- are usually available
- ▶ Currently available packages for optimization are expensive!
- ▶ Estimated workload for this solution:
1-4 weeks depending on programming skills...



- ▶ Parametric studies
- ▶ Optimization
- ▶ Mathematical modeling
 - ▶ Creation of Response Surfaces for mathematical modeling





- ▶ Automated generation of geometry
(SolidWorks or Catia V5 etc.)
- ▶ Automated generation of mesh
(Gambit or other automated meshing tool)



- ▶ Automated generation of geometry (Parasolid, Step etc.) through Visual basic programming
- ▶ Downloadable functions (www.solidworks.com) are 95% ready for use!

Minor correction: reading and handling input values...



```
Sub ParametricSub(ByVal Base1_Passed As Double, ByVal Base2_Passed As Double, ByVal  
Extrude_Passed As Double, ByVal Draft_Passed As Double)  
    Dim swApp As Object          ' Define variable used to hold the application object  
    Dim Part As Object          ' Define variable used to hold the part object  
    Dim Pi As Double  
    Pi = 3.14159265358979  
    MyPath = CurDir             ' Determine current directory  
  
    Set swApp = CreateObject("SldWorks.Application")  
  
' Uncomment this if you wish to make the new SolidWorks session visible  
    Set swApp.Visible (True)  
  
'Load file from current directory. This is currently hardcoded to c:\temp  
    Set Part = swApp.OpenDoc(MyPath + "\test.SLDPRT", swDocPART)
```

```
If Part Is Nothing Then  
    Exit Sub  
Else  
    Set Part = swApp.ActivateDoc("test.SLDPRT")  
End If  
  
Part.Parameter("D2@Sketch1").SystemValue = Base2_Passed / 1000  
Part.Parameter("D1@Extrude2").SystemValue = Extrude_Passed / 1000  
Part.Parameter("D1@Draft1").SystemValue = Draft_Passed / 180 * Pi  
  
' Regenerate the partfile since changes were made  
Part.EditRebuild  
  
' Save the changes made to the partfile  
Part.Save  
  
' Close the partfile  
swApp.CloseDoc "test.SLDPRT"  
  
swApp.ExitApp
```



```
Sub ParametricSub(ByVal Base1_Passed As Double, ByVal Base2_Passed As Double, ByVal  
Extrude_Passed As Double, ByVal Draft_Passed As Double)  
    Dim swApp As Object          ' Define variable used to hold the application object  
    Dim Part As Object          ' Define variable used to hold the part object  
    Dim Pi As Double  
    Pi = 3.14159265358979  
    MyPath = CurDir              ' Determine current directory  
  
    Set swApp = CreateObject("SldWorks.Application")  
  
' Uncomment this if you wish to make the new SolidWorks session visible  
    Set swApp.Visible (True)  
  
' Load file from current directory. This is currently hardcoded to c:\temp  
    Set Part = swApp.OpenDoc(MyPath + "\test.SLDPRT", swDocPART)
```

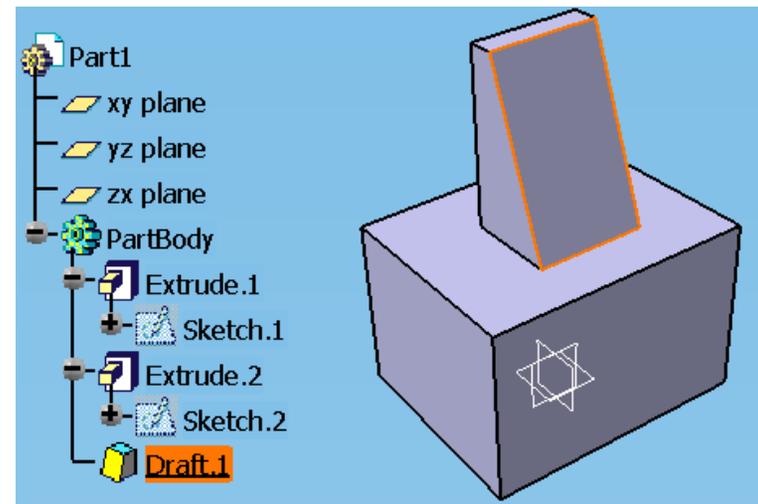
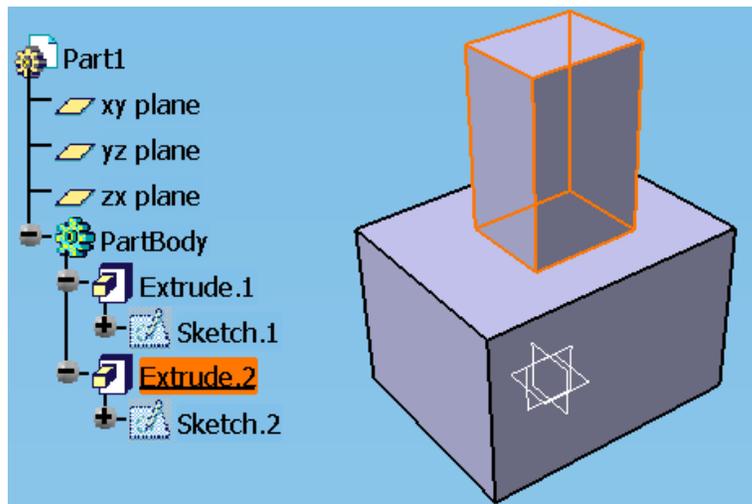
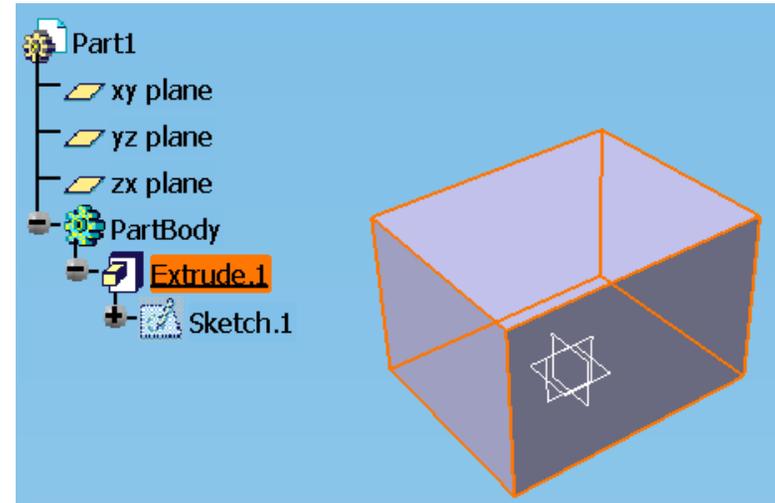
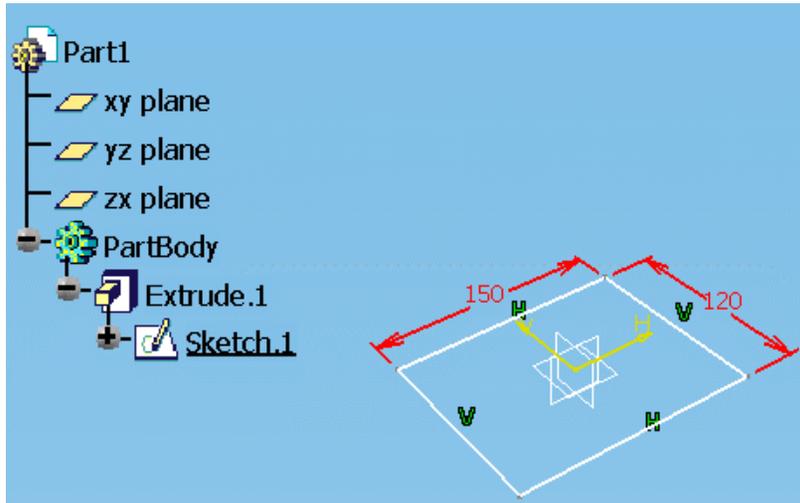
```
If Part Is Nothing Then  
    Exit Sub  
Else  
    Set Part = swApp.ActivateDoc("test.SLDPRT")  
End If  
  
Part.Parameter("D2@Sketch1").SystemValue = Base2_Passed / 1000  
Part.Parameter("D1@Extrude2").SystemValue = Extrude_Passed / 1000  
Part.Parameter("D1@Draft1").SystemValue = Draft_Passed / 180 * Pi  
  
' Regenerate the partfile since changes were made  
Part.EditRebuild  
  
' Save the changes made to the partfile  
Part.Save  
  
' Close the partfile  
swApp.CloseDoc "test.SLDPRT"  
  
swApp.ExitApp
```



```
Sub ParametricSub(ByVal Base1_Passed As Double, ByVal Base2_Passed As Double, ByVal  
Extrude_Passed As Double, ByVal Draft_Passed As Double)  
    Dim swApp As Object          ' Define variable used to hold the application object  
    Dim Part As Object           ' Define variable used to hold the part object  
    Dim Pi As Double  
    Pi = 3.14159265358979  
    MyPath = CurDir              ' Determine current directory  
  
    Set swApp = CreateObject("SldWorks.Application")  
  
' Uncomment this if you wish to make the new SolidWorks session visible  
    Set swApp.Visible (True)  
  
' Load file from current directory. This is currently hardcoded to c:\temp  
    Set Part = swApp.OpenDoc(MyPath + "\test.SLDPRT", swDocPART)  
  
If Part Is Nothing Then  
    Exit Sub  
Else  
    Set Part = swApp.ActivateDoc("test.SLDPRT")  
End If  
  
    Part.Parameter("D2@Sketch1").SystemValue = Base2_Passed / 1000  
    Part.Parameter("D1@Extrude2").SystemValue = Extrude_Passed / 1000  
    Part.Parameter("D1@Draft1").SystemValue = Draft_Passed / 180 * Pi  
  
' Regenerate the partfile since changes were made  
    Part.EditRebuild  
  
' Save the changes made to the partfile  
    Part.Save  
  
' Close the partfile  
    swApp.CloseDoc "test.SLDPRT"  
  
    swApp.ExitApp
```



```
Part.Parameter("D2@Sketch1").SystemValue = Base2_Passed / 1000  
Part.Parameter("D1@Extrude2").SystemValue = Extrude_Passed / 1000  
Part.Parameter("D1@Draft1").SystemValue = Draft_Passed / 180 * Pi
```



- ▶ Automated through the use of journal files
- ▶ Reads geometry in native CAD

- ▶ Special:
 - ▶ Boundary conditions (different from walls) must have fixed names. When entities in Gambit are prone to **name changes** use:
 - ▶ `$name = LOC2ENT(t_fa, $X, $Y, $Z)`
 - ▶ Locally refined mesh
 - ▶ **Dummy geometry** can be used as **source** in **size functions** when solving cases with:
 - ▶ moving geometry
 - ▶ Inconsistent topology

- ▶ Automated through the use of journal files.
 - ▶ Journal files are manually prepared in advance
 - ▶ Fluent cannot handle string variables
 - ▶ Boundary conditions are assigned with names consistent with the ones specified in the Gambit journal
 - ▶ When mesh files have been generated, fluent is run via a system call:
 - ▶ `Fluent 2ddp -g -i <journalfile.name>`
 - ▶ Windows OS does not have the '&' flag for app. threading – Use e.g:

```
struct stat buf;
char FluentCall[256];

sprintf(FluentCall,"Fluent 3ddp -g -i %s",name_variable);
remove("done.tmp");
system(FluentCall);

// looping while no "done.tmp" file exists in directory.
while(stat("done.tmp",&buf) !=0 ) {
    sleep(10000);
}
```

- ▶ Objectives and constraints are output in text files (transcripts) using basic functions within Fluent e.g.
 - ▶ Integral or min/max values on surfaces/volumes etc.
 - ▶ Pre-made user define function (UDF)
- ▶ Some functions that may not be available in Fluent are handled later in Matlab, e.g. pressure difference across domain:
 - ▶ Transcript contains both inlet pressure and outlet pressure
 - ▶ Matlab scans the transcript for each output value and calculates the pressure difference



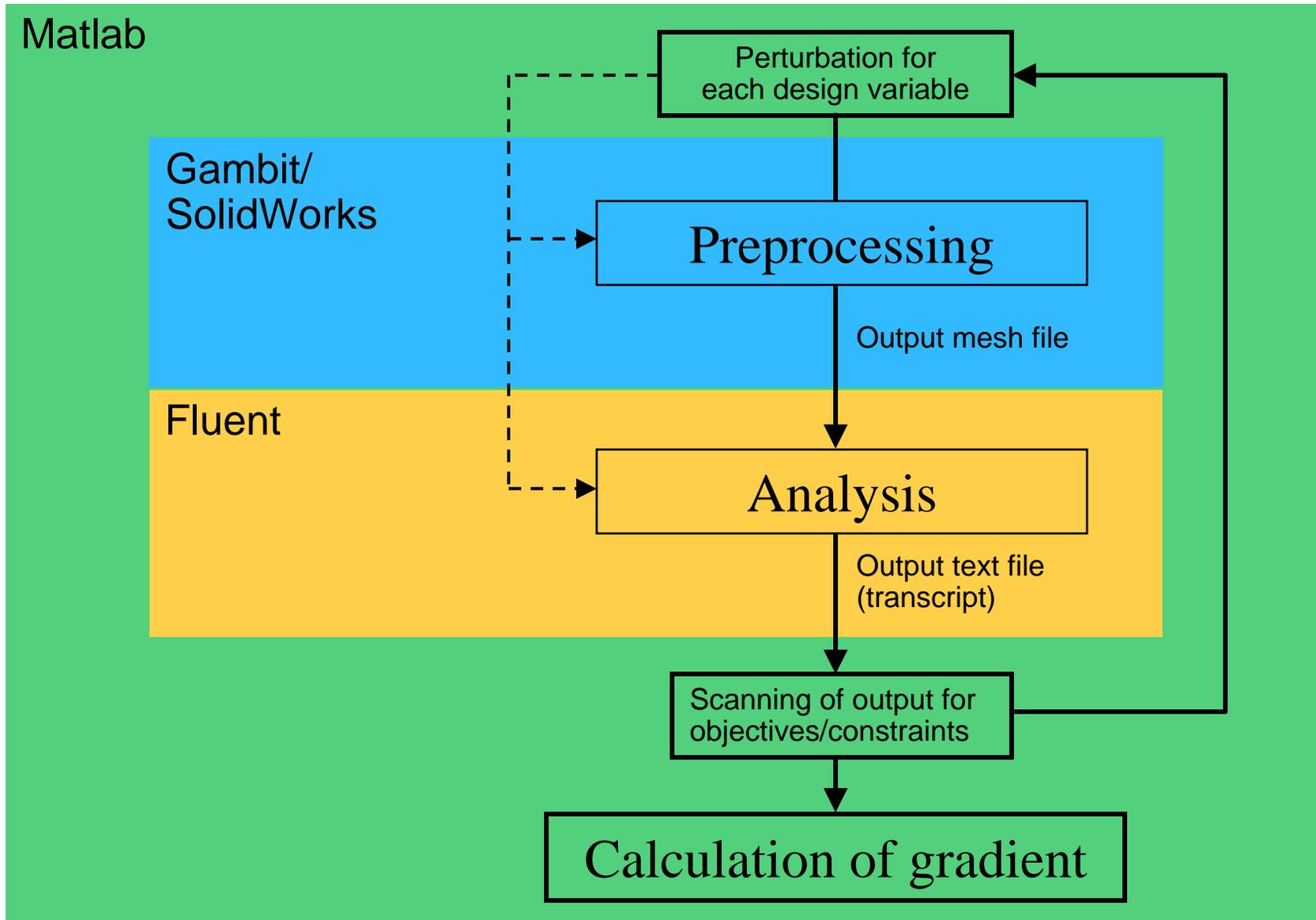
- ▶ Typical fluent transcript format

```

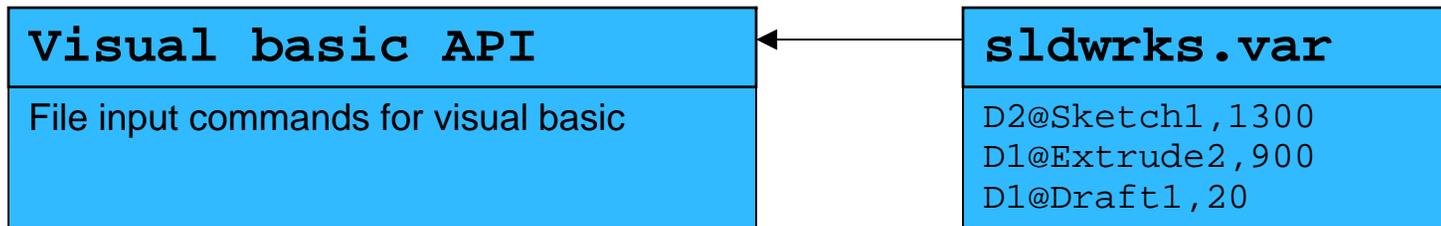
result.trn - Notepad
File Edit Format View Help
> report/wall-forces yes 1 0
Force vector: (1 0 0)
zone name          pressure          viscous          total
                   force            force            force
                   n              n              n
-----
<NAME>             817.52597        2.4565692       819.98254
-----
net                 817.52597        2.4565692       819.98254
> report/wall-forces yes 0 1
Force vector: (0 1 0)
zone name          pressure          viscous          total
                   force            force            force
                   n              n              n
-----
<NAME>             428.86211        0.6758506       429.53796
-----
net                 428.86211        0.6758506       429.53796
> file/stop-trans
    
```

Back

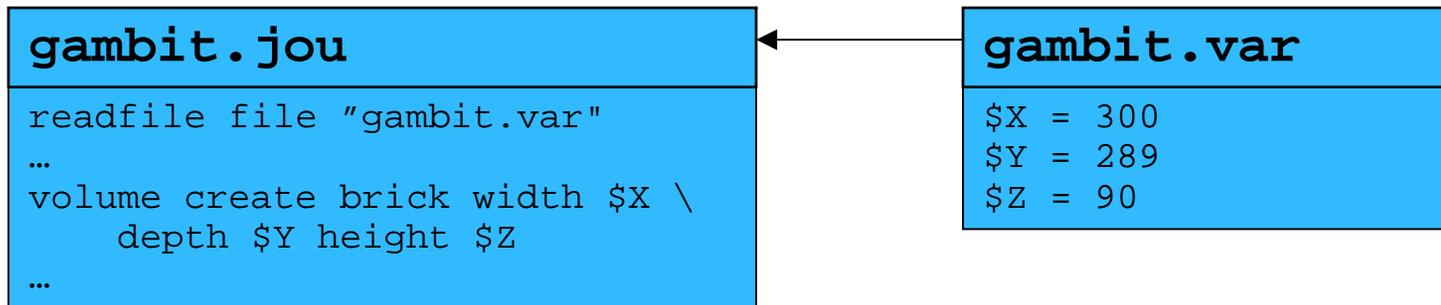




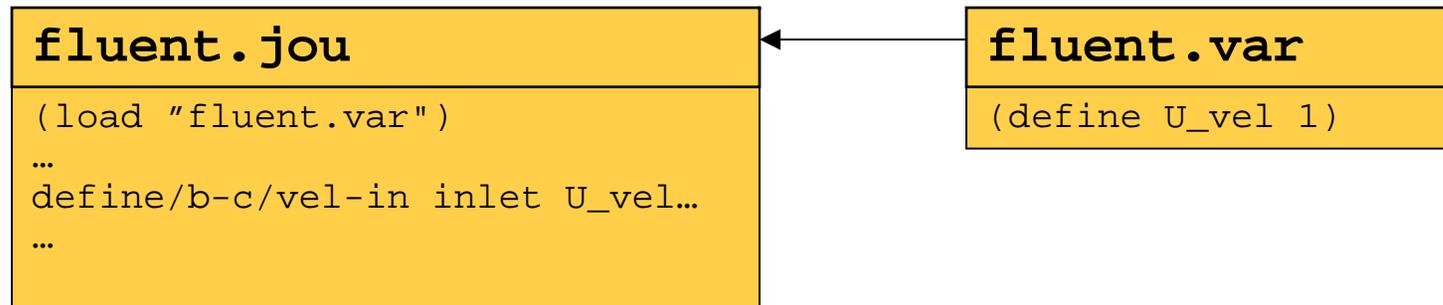
▶ Handling by SolidWorks



▶ Handling by Gambit



- ▶ Flow inlet velocity
- ▶ Inlet/outlet pressure
- ▶ Concentration of species



Matlab

Visual basic API

File input commands for visual basic

sldwrks.var

```
D2@Sketch1,1300  
D1@Extrude2,900  
D1@Draft1,20
```

gambit.jou

```
readfile file "gambit.var"  
...  
volume create brick width $X \  
    depth $Y height $Z  
...
```

gambit.var

```
$X = 300  
$Y = 289  
$Z = 90
```

fluent.jou

```
(load "fluent.var")  
...  
define/b-c/vel-in inlet U_vel...  
...
```

fluent.var

```
(define U_vel 1)
```

Back

- ▶ Fluent transcript is consistent

```

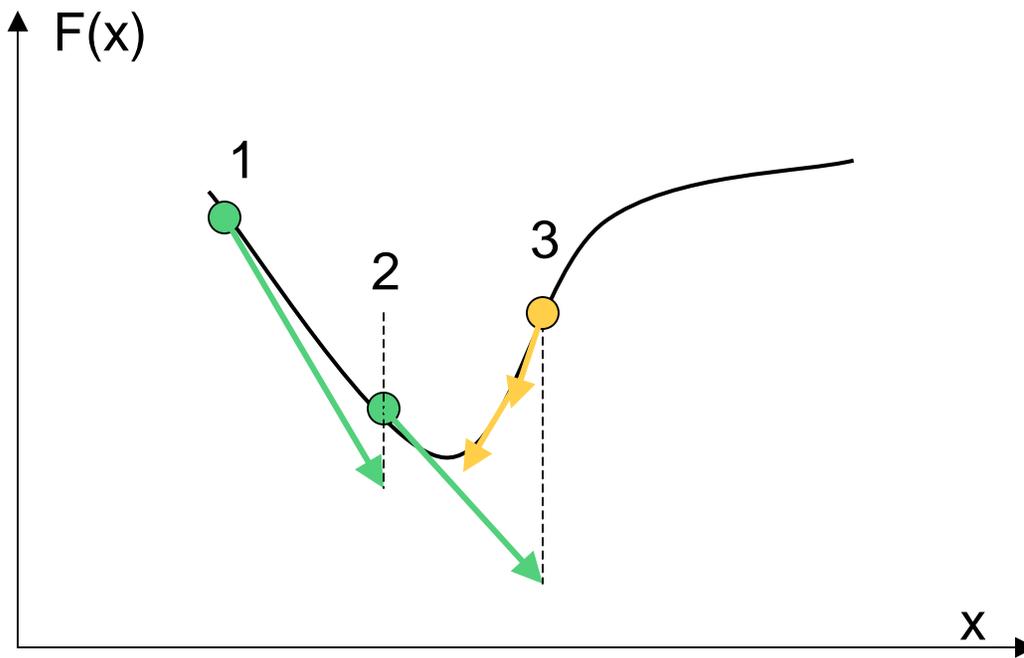
result.trn - Notepad
File Edit Format View Help
> report/wall-forces yes 1 0
Force vector: (1 0 0)
zone name          pressure      viscous      total
                   force         force        force
                   n             n            n
-----
<NAME>             817.52597    2.4565692    819.98254
-----
net                 817.52597    2.4565692    819.98254
> report/wall-forces yes 0 1
Force vector: (0 1 0)
zone name          pressure      viscous      total
                   force         force        force
                   n             n            n
-----
<NAME>             428.86211    0.6758506    ██████████
-----
net                 428.86211    0.6758506    429.53796
> file/stop-trans
    
```

- ▶ Matlab scans the file for e.g.
 - ▶ The second occurrence of the string <NAME>
 - ▶ The third number occurring hereafter

Back



- ▶ Gradient achieved using finite difference
- ▶ Step change set to e.g. 10% of domain diagonal length
 - ▶ Adaptive refinement of step size



- ▶ Estimated workload for this solution:
1-4 weeks depending on programming skills...
- ▶ Matlab supplies an optimization package that might substitute some of this work.
- ▶ With a computing cluster, sensitivity analysis is performed in parallel.
- ▶ A fully automated system enables “working” during weekends.

