FUN3D v14.0 Training:
A Few Attractive Additional Features for Internal Flow Simulations*

Jan Carlson

Spring 2023

*Features applicable to external flow simulations as well.

Today's Features

- Typical boundary conditions for internal flows
- Individual component performance tracking
- Back pressure controller
- Porous media modeling
- Three simple “turbofan” engine models

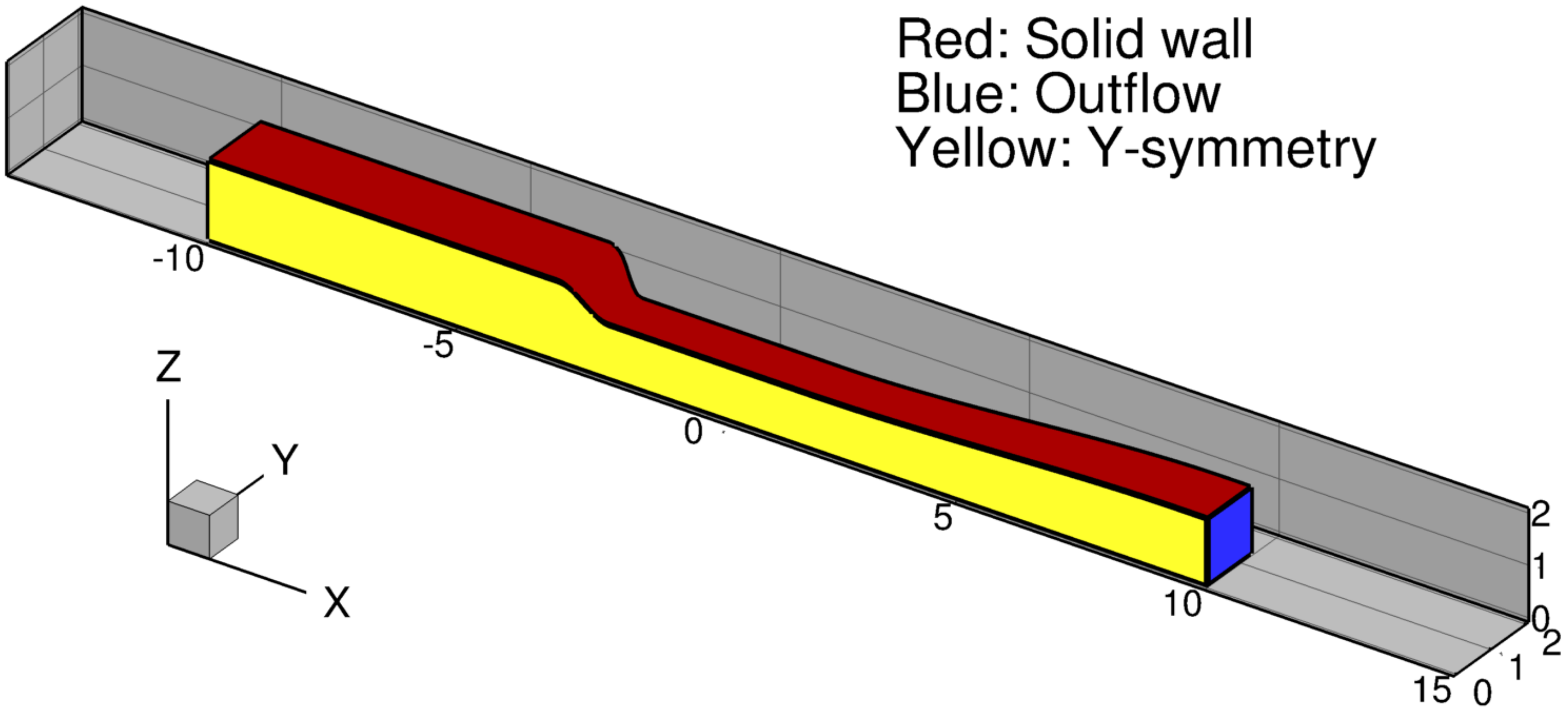
Sequence of Events

- Simple internal flow
  - Initial solution using fixed boundary conditions
  - Assessing solution development via component tracking
  - Adjusting flow conditions using a back pressure controller
  - Adding honeycomb/heat exchanger modeling using a porous media method
  - Readjusted flow conditions using the controller
- Simple engine model
  - Coupling boundary conditions

Defining Flows

- External and mixed external/internal flows
  - Far-field freestream exists, but can be different from the reference
  - $\mathrm{M}_\infty = \mathrm{M}_{\mathrm{ref}}$ (notable exception being many rotorcraft simulations)
  - $p_\infty = p_{ref} = 1/\gamma$
  - $p_{t\infty} = p_\infty \left(1 + \frac{1}{2}(\gamma - 1)\mathrm{M}_\infty^2\right)^{\gamma/\gamma-1}$
- Internal flows
  - No far-field freestream exists
  - Reference conditions *can* be different than “freestream”, but…
  - The pressure field may or may not be uniquely defined (depends upon the choice of boundary conditions)

Perspective View of Converging-Diverging Duct




```
mapbc file

6
1    6663 z-symmetry
2    xxxx wall
3    xxxx inflow
4    xxxx outflow
5    6662 y-symmetry
6    xxxx wall
```

- ¼-plane symmetric model
- Flow from left to right
- Boundary faces: 1, bottom; 2, top; 3, left; 4, right; 5: this side; 6: that side.

Boundary Condition Types

```
mapbc file*

6
1    6663 z-symmetry
2    4000 wall
3    7011 inflow
4    5051 outflow
5    6662 y-symmetry
6    4000 no-slip wall
```

No additional input required

3000: Tangency
4000: Strong no slip
4100: Wall function
4110: Weak no slip

7011: Subsonic inflow, total conditions
7036: Mass inflow
7100: Fixed (supersonic)
Velocity-Temperature (not yet released)

5051: Static pressure, supersonic extrapolate
5052: Mach number
5061: Static pressure, non-reflecting
7012: Static pressure, subsonic, no reverse flow
7031: Mass outflow
7200: Solution functional

No additional input required

3000: Tangency
4000: Strong no slip
4100: Wall function
4110: Weak no slip

*Not all combinations work or work well.

Terms and Conditions May Apply

- With no far-field boundaries, there is no “freestream”
- For most simulations, it is useful to define “freestream/infinity” as the condition existing in the area of interest, e.g., a test section or the area minimum, at any particular moment in time
- When using the total conditions—static pressure inflow-outflow combination for boundary types:
  - The values of the total pressure and temperature at the inflow are determined using the isentropic relations with the reference Mach number (thinking as an external flow), i.e.,
    - $p_t/p_{ref} = f(M_{ref})$
    - $T_t/T_{ref} = f(M_{ref})$
  - The freestream conditions can be modulated to compensate for viscous losses in the flow by using the back pressure at the outflow boundary to achieve $M_\infty = M_{ref}$

Next

A complete listing of the fun3d.nml file

fun3d.nml

```
&project
project_rootname = 'baseline'
/
 &governing_equations
   eqn_type         = 'compressible'
   viscous_terms    = 'turbulent'
 /
 &turbulent_diffusion_models
  turbulence_model      = 'sa-neg’
 /

&raw_grid
 grid_format = 'aflr3'
/

&inviscid_flux_method
 flux_limiter         = 'none’ ! default
 flux_construction     = 'roe’  ! default
 flux_construction_lhs = 'consistent’ ! default is ‘vanleer’
/

&nonlinear_solver_parameters
 schedule_iteration = 1 1000
 schedule_cfl       = 1. 50.
 schedule_cflturb   = 1. 20.
/

&code_run_control
 steps             = 25000
 restart_write_freq = 100
 stopping_tolerance = 1.0e-15
restart_read        = 'off’ ! ‘on’
/
```

fun3d.nml

```
&global
  boundary_animation_freq = -1. ! output solution on boundaries at job end
/

&boundary_output_variables
  number_of_boundaries = -1    ! determine number of boundaries out from boundary_list
  boundary_list       = '1-6' ! list of boundaries to be included
  ptot               = T     ! add these parameters
  htot               = T
  ttot               = T
  q_criterion        = T
  yplus              = T
/

&sampling_parameters
  number_of_geometries = 2
   type_of_geometry(1) = 'line'   ! sample the solution between points p1_line and p2_line
            label(1) = 'centerline'
        p1_line(1:3,1) = -1000.0 0.01 0.01
        p2_line(1:3,1) = 1000.0 0.01 0.01
      variable_list(1) = 'x,y,z,rho,u,v,w,p,ptot,ttot,htot,mach' ! extract these parameters
 sampling_frequency(1) = -1 ! write file at end of run

   type_of_geometry(2) = 'volume_points' ! extract single points of data from the volume
   number_of_points(2) = 2
 sampling_frequency(2) = 1 ! sample every iteration/timestep
          points(:,2,1) =  -0.5,0.01, 0.01 ! point 1 location
          points(:,2,2) =   0.5,0.01, 0.01 ! point 2 location
      variable_list(2) = 'p,ptot,u,mach'    ! extract these parameters
             plot(2) = 'serial_history'   ! format list as ASCII output list of data
/
```

fun3d.nml

```
&component_parameters          ! track F & M, as well as,
  allow_flow_through_forces = T ! parameters such as mass flow, Mach number,
  list_forces              = T ! pressure, and temperature

! track six components in this example, some are boundaries,
! some are slices through the mesh
  number_of_components = 6

     component_name(1) = 'inflow' ! component label
    component_input(1) = '3'      ! inflow boundary boundary number(s)
    component_count(1) = -1       ! extract count from component_input list
 component_symmetry(1) = 4        ! quarter-plane symmetric grid

     component_name(2) = 'outflow' ! component label
    component_input(2) = '4'       ! outflow boundary number(s)
    component_count(2) = 1         ! here the count is explicitly set
 component_symmetry(2) = 4         ! quarter-plane symmetric grid

     component_name(3) = 'total'   ! combine inflow and outflow for mass flow
    component_input(3) = '3,4'     ! imbalance as a convergence indicator
    component_count(3) = -1

    component_count(4) = 1         ! sample the middle of the test
    component_input(4) = '0'       ! section for conditions
     component_type(4) = 'circle'  ! geometry cut will be a circle
  circle_center(1:3,4) = 0.0, 0.0, 0.0  ! center of circle (x,y,z)
  circle_normal(1:3,4) = 1.0, 0.0, 0.0  ! direction normal to plane of circle
       circle_radius(4) = 3.0.          ! radius of circle
     component_name(4) = 'throat'       ! component label
 component_symmetry(4) = 4              ! quarter-plane symmetry

    component_count(5) = 1  ! sample at Sta. -9.
    component_input(5) = '0' ! component not a boundary
     component_type(5) = 'circle'
  circle_center(1:3,5) = -9.0, 0.0, 0.0
  circle_normal(1:3,5) = 1.0, 0.0, 0.0
      circle_radius(5) = 10.0
     component_name(5) = 'Sta.1' ! component label
 component_symmetry(5) = 4

    component_count(6) = 1  ! sample at Sta. -7.
    component_input(6) = '0' ! component not a boundary
     component_type(6) = 'circle'
  circle_center(1:3,6) = -7.0, 0.0, 0.0
  circle_normal(1:3,6) = 1.0, 0.0, 0.0
      circle_radius(6) = 10.0
     component_name(6) = 'Sta.2' ! component label
 component_symmetry(6) = 4
/
```

fun3d.nml

```
&reference_physical_properties
 mach_number      = 0.2        ! Reference condition
 reynolds_number  = 5.0e+6     ! 1/[m]
 temperature      = 288.15     ! Reference static temperature
 temperature_units = 'Kelvin'
/

&boundary_conditions
          grid_units = 'meters'
   wall_temperature(2) = -1. ! pseudo-adiabatic condition
    wall_temp_flag(2) = T
   wall_temperature(6) = -1. ! pseudo-adiabatic condition
    wall_temp_flag(6) = T

  total_pressure_ratio(3) = 1.02828 ! M = 0.2 using (p_t/p) isentropic equation
total_temperature_ratio(3) = 1.00800 ! M = 0.2 using (T_t/T) isentropic equation
  static_pressure_ratio(4) = 1.010   ! first guess
! solution functional
  dynamic_boundary_conditions = .false. ! Set to true to engage the boundary condition controller
 boundary_functional_name(4)   = 'back_pressure' ! name of boundary condition used by the controller
         field_point1(4,:) =  0.0,  0.01, 0.01 ! survey point
    update_frequency(4)   = 500
/

&flow_initialization
  number_of_volumes = 1
  type_of_volume(1) = 'sphere'    ! V-Parr type initialization
    center(1:3,1) = 0.0 0.0 0.0
      radius(1) = 50.0
        rho(1) = 1.00
         u(1) = 0.10
         c(1) = 1.0
/
```

Useful Files Output From This Simulation

Useful diagnostic files

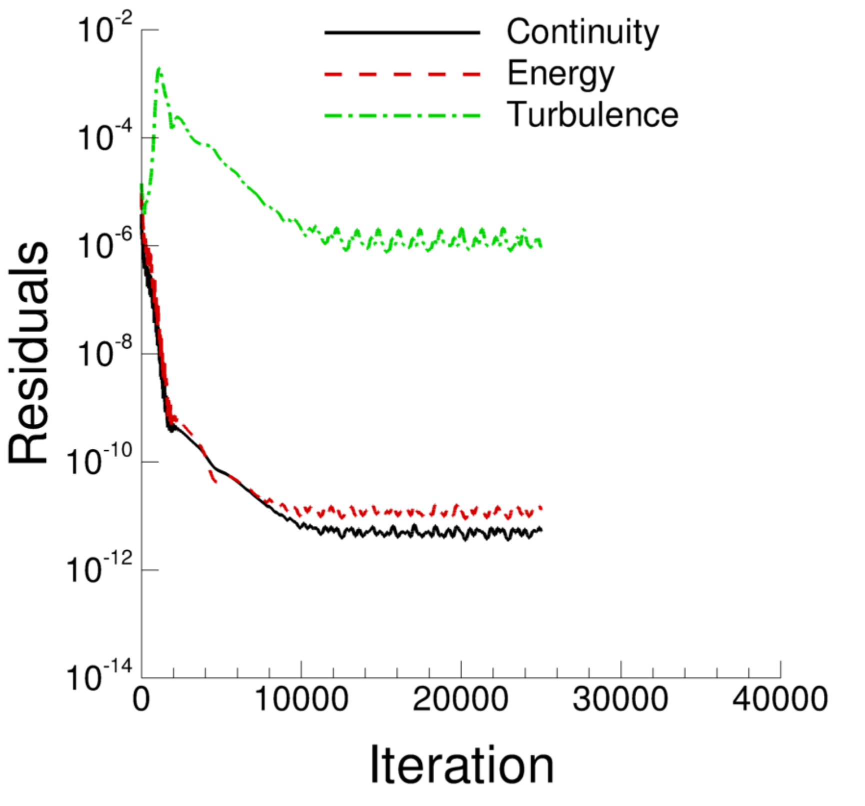
- [project_name] _hist.dat
  - Solution residuals
  - Simulation total, viscous, and pressure components of forces and moments
- [project_name]_fm_[component_name].dat
  - User defined component force and moment histories
  - “Flow through” area weighted statistics
    - Mass flow, density, velocity, Mach number,
    - Temperature, Total temperature
    - p/p_{ref}, p_{total}/p_{ref},
    - RMS temperature (generic gas path only)
    - X-momentum flux, Y-momentum flux, Z-momentum flux
- [project_name]_[boundary number]_controller.dat
  - History file of the PID-controller when the back pressure controller is engaged

Analysis

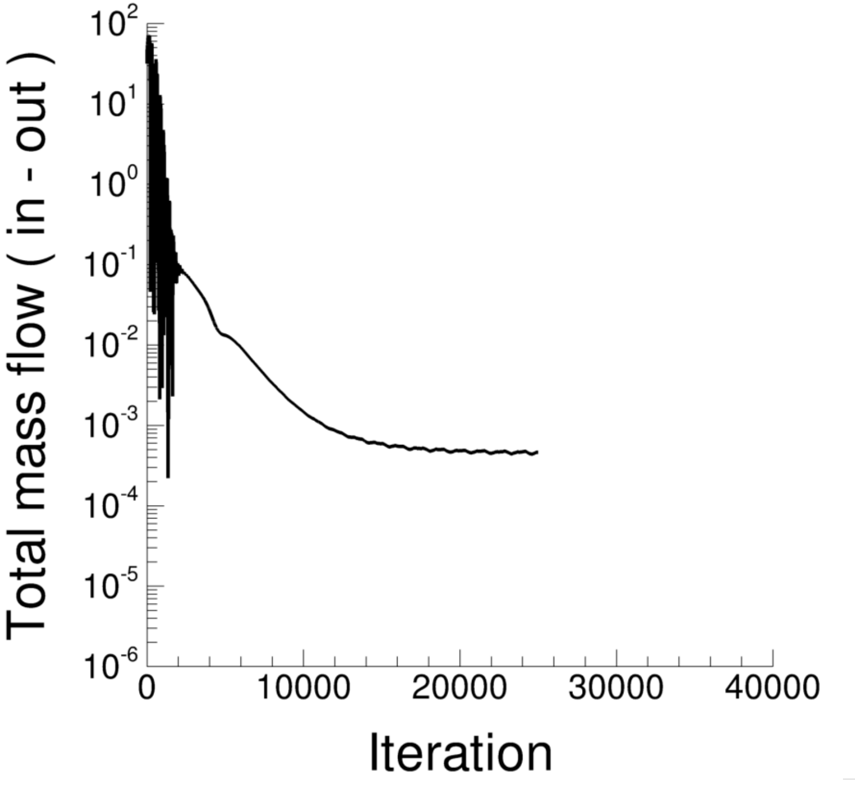
- Plots
  - Residual history
  - Mass flow conservation
  - Centerline Mach number
- Components output

First Pass

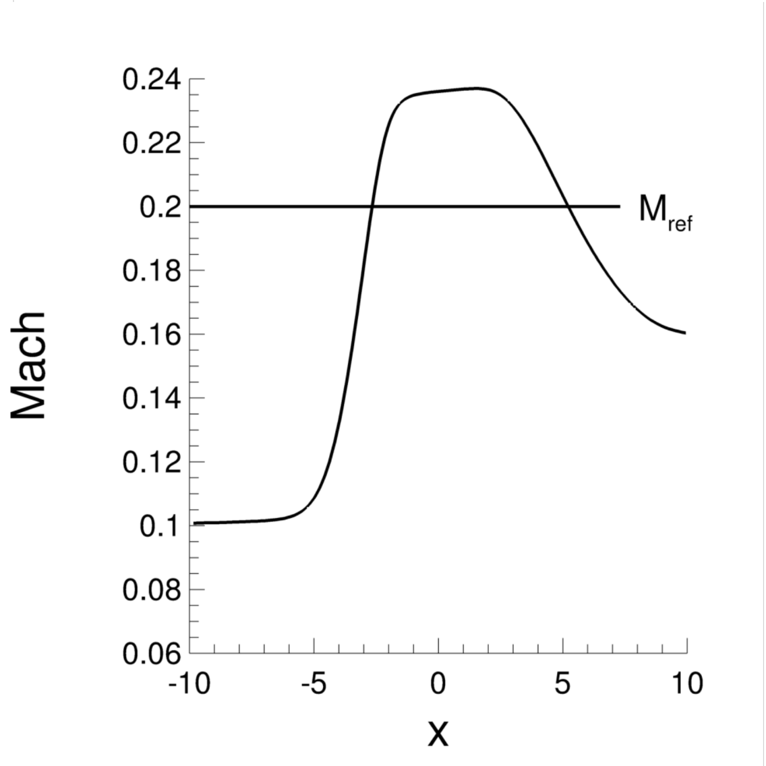
- First guess on back pressure, $p_{back} = p_{outflow}/p_{ref} = 1.010$, is a little low, pushing the centerline Mach number at X = 0.0 to close to $M_\infty = 0.24$. The desired condition is $M_\infty = M_{ref} = 0.20$.



Plot using component #3 output



Plot using sample #1 output



[project_name]_stream_info.dat

Component performance summaries

FUN3D 14.0-8c4c24db0e Flow started 02/10/2023 at 09:25:56 with 400 processors.

Component/stream information for project: baseline

Iteration = 25000
Grid units = meters
l_ref = 1.0000

Free stream:
Reynolds # = 5.0000E+06 per meters
Mach (reference) = 0.200
(farfield) = 0.2000 (Mach_reference * vinf_ratio)
cv = 717.633 [J/kg-k]
cp = 1004.686 [J/kg-k]
gamma = 1.400
Molecular weight = 28.9645 [kg/kmol]
R (specific) = 287.0530 [J/kg-K]
Equation type: cal_perf_compress
Viscous terms: turbulent
Chemical kinetics: not applicable
Thermal energy: not applicable
T_infinity = 288.1 [K]
= 518.7 [R]
= 59.0 [F]
temperature_ref = 288.1 [K]
r_infinity = 1.3146 [kg/m^3]
= 0.82068E-01 [lbm/ft^3]
= 0.25507E-02 [slug/ft^3]
rho_ref = 0.0000 [kg/m^3]
p_infinity = 108736.107 [Pa]
= 15.770839 [lbf/sq.in.]
= 2271.00081 [lbf/sq.ft.]

a_infinity = 340.29 [m/s]
= 1116.45 [ft/s]
= 761.22 [miles/hour]
u_infinity = 68.059 [m/s]
= 223.290 [ft/s]
= 152.243 [miles/hour]
v_ref = 0.00 [m/s]
q_infinity = 3044.611 [Pa]
= 0.442 [lbf/sq.in.]
= 63.588 [lbf/sq.ft.]
rho*a^2 infinity = 152230.550 [Pa]
= 22.079175 [lbf/sq.in.]
= 3179.40114 [lbf/sq.ft.]
p,t_infinity = 111811.3 [Pa]
= 16.217 [lbf/sq.in.]
= 2335.227 [lbf/sq.ft.]
= 1.0283 [Fun3d]
T,t_infinity = 290.5 [K]
= 522.8 [R]
= 63.1 [F]
bulk viscosity = 1.78940E-05 [kg/m-sec]
= 1.0000 [non-dimensional]

Time step (dt) = 0.0000 [non-dimensional]
(dt/a_inf) = 0.0000 [s]
Reference areas :
aerodynamic = 1.0000 mesh^2, 1.0000 [m^2]
propulsion = 1.0000 mesh^2, 1.0000 [m^2]
....
Summary of plenum gas properties
Boundary T_total Avg. Mol. wt R_specific gamma species
....

[project_name]_stream_info.dat

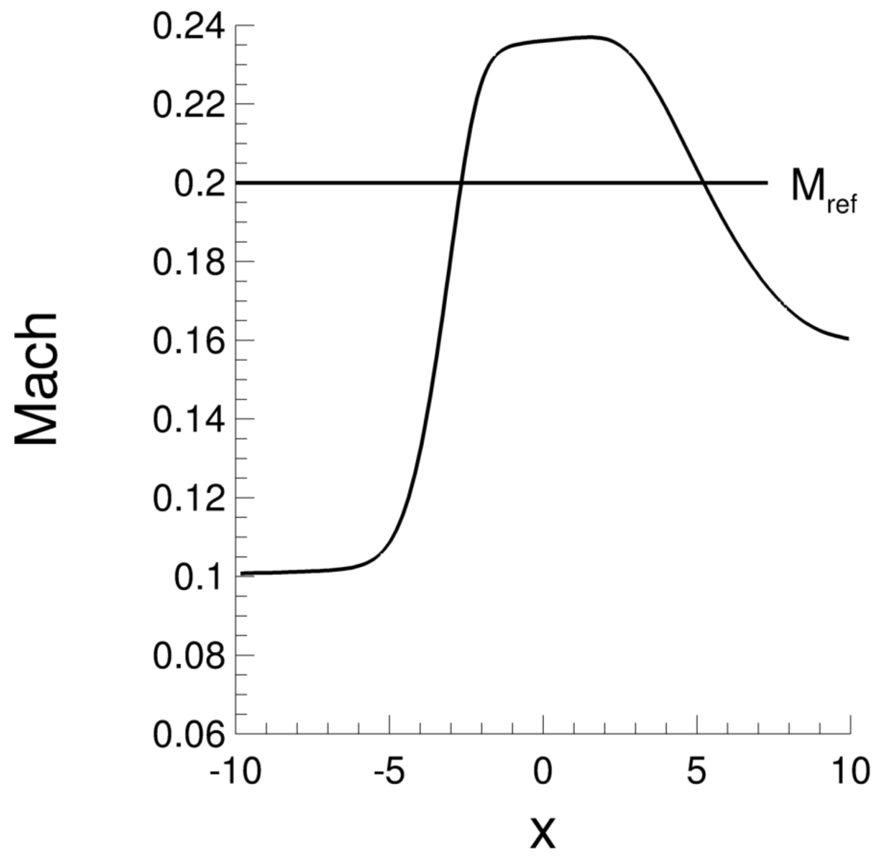
```
......................................................................
            inflow

component_type     ( 1)   = boundary
component_symmetry ( 1)   =    4.0
shape direction    ( 1)   =    0.000    0.000    0.000
shape area                =  2.25000E+00 mesh^2,   2.25000E+00 [m^2],   2.42188E+01 [ft^2]
shape area * symmetry     =  9.00000E+00 mesh^2,   9.00000E+00 [m^2],   9.68752E+01 [ft^2]
average density           =      1.015 [Fun3d],    1.3343 [kg/m^3],    0.08330 [lbm/ft^3]
average u velocity        =      0.101 [Fun3d],    34.3469 [m/s],       112.687 [ft/s]
average v velocity        =     -0.000 [Fun3d],    -0.0149 [m/s],        -0.049 [ft/s]
average w velocity        =     -0.000 [Fun3d],    -0.0149 [m/s],        -0.049 [ft/s]
average static pressure   =      0.729 [Fun3d],  111021.870 [Pa],        16.102 [psi]
velocity magnitude        =      0.101 [Fun3d],    34.3469 [m/s],       112.687 [ft/s]
local speed of sound      =      1.003 [Fun3d],   341.3070 [m/s],      1119.774 [ft/s]
local Mach number         =      0.101
average total pressure    =      0.734 [Fun3d],  111811.079 [Pa],        16.217 [psi]
minimum total pressure    =      0.729 [Fun3d],  111026.497 [Pa],        16.103 [psi]
maximum total pressure    =      0.735 [Fun3d],  111931.415 [Pa],        16.234 [psi]
average cp * Tt           =    2.5200    [Fun3d],   2.91816E+05 [J/kg]
average cp * T            =    2.5149    [Fun3d],   2.91226E+05 [J/kg]
average entropy           =  0.34074E-06 [Fun3d],   0.13693E-03 [J/K]
average static temperature =     1.006 [Fun3d],    289.868 [K],         521.762 [R]
average total temperature  =     1.008 [Fun3d],    290.455 [K],         522.819 [R]
 --------------------------------------------
mass flow                 =    0.92200 [Fun3d],   412.45502 [kg/s],   909.30767 [lbm/s]
mass flow ( corrected )   =    0.83886 [Fun3d],   375.26552 [kg/s],   827.31886 [lbm/s]
 : sqrt(T_total / T_std.) =      1.004
 : p_total / p_std.       =      1.103
 --------------------------------------------
```

[project_name]_stream_info.dat

```
Flow-through

X-momentum flux        =  0.93081E-01 [Fun3D],   0.142E+05 [N],     3185.5    [lbf]
Y-momentum flux        = -0.40299E-04 [Fun3D],   -6.13     [N],    -1.3791    [lbf]
Z-momentum flux        = -0.40299E-04 [Fun3D],   -6.13     [N],    -1.3791    [lbf]
Delta X-pressure flux  =  0.13514     [Fun3D],   0.206E+05 [N],     4624.7    [lbf]
Delta Y-pressure flux  =  0.0000      [Fun3D],   0.00      [N],     0.0000    [lbf]
Delta Z-pressure flux  =  0.0000      [Fun3D],   0.00      [N],     0.0000    [lbf]
-------------------------------------------
Totals

X-Viscous forces         =  0.93081E-01 [Fun3D],   0.142E+05 [N],     3185.5    [lbf]     4.654 [1/Q infinity]
Y-Viscous forces         = -0.40299E-04 [Fun3D],   -6.13     [N],    -1.3791    [lbf]    -0.002 [1/Q infinity]
Z-Viscous forces         = -0.40299E-04 [Fun3D],   -6.13     [N],    -1.3791    [lbf]    -0.002 [1/Q infinity]
Delta X-Pressure forces  =  0.13514     [Fun3D],   0.206E+05 [N],     4624.7    [lbf]     6.757 [1/Q infinity]
Delta Y-Pressure forces  =  0.0000      [Fun3D],   0.00      [N],     0.0000    [lbf]     0.000 [1/Q infinity]
Delta Z-Pressure forces  =  0.0000      [Fun3D],   0.00      [N],     0.0000    [lbf]     0.000 [1/Q infinity]
X-vacuum forces          =  0.33581E-04 [Fun3D],   5.11      [N],     1.1492    [lbf]
Y-vacuum forces          =  0.0000      [Fun3D],   0.00      [N],     0.0000    [lbf]
Z-vacuum forces          =  0.0000      [Fun3D],   0.00      [N],     0.0000    [lbf]

X-Total forces           =  0.22822     [Fun3D],   0.347E+05 [N],     7810.2    [lbf]    11.411 [1/Q infinity]
Y-Total forces           = -0.40299E-04 [Fun3D],   -6.13     [N],    -1.3791    [lbf]    -0.002 [1/Q infinity]
Z-Total forces           = -0.40299E-04 [Fun3D],   -6.13     [N],    -1.3791    [lbf]    -0.002 [1/Q infinity]
atan ( F_y / F_x )       = -0.10117E-01[deg.]
atan ( F_z / F_x )       = -0.10117E-01[deg.]

Rotated X-Total forces   =  0.22822     [Fun3D],   0.347E+05 [N],     7810.2    [lbf]    11.411 [1/Q infinity]
Rotated Y-Total forces   = -0.40299E-04 [Fun3D],   -6.13     [N],    -1.3791    [lbf]    -0.002 [1/Q infinity]
Rotated Z-Total forces   = -0.40299E-04 [Fun3D],   -6.13     [N],    -1.3791    [lbf]    -0.002 [1/Q infinity]
atan ( F_y,rot / F_x,rot ) = -0.10117E-01[deg.]
atan ( F_z,rot / F_x,rot ) = -0.10117E-01[deg.]

Total force              =  0.22822     [Fun3D],   0.347E+05 [N],     7810.2    [lbf]    11.411 [1/Q infinity]
Total vacuum force       =  0.93115E-01 [Fun3D],   0.142E+05 [N],     3186.6    [lbf]     4.656 [1/Q infinity]
```

Modulate The Back Pressure

- The freestream Mach number is high due to a low value of the back pressure, $p_{back} = p_{outflow}/p_{ref} = 1.010$, being a bit low
- Tactics to bring M_∞ down to M_{ref}
  - Manual trial and error
    - This often is the more expedient course of action for simulations with long or unknown response times
  - PID-control process
    - Takes some trial and error to determine the controller coefficients for new problems
    - Allows for automation of a series of simulations



Engage The Controller: update #1 fun3d.nml

Set dynamic_boundary_conditions to .true. in the fun3d.nml file

```
 &boundary_conditions
                  grid_units = 'meters'
        wall_temperature(2) = -1. ! psuedo-adiabatic condition
          wall_temp_flag(2) = T
        wall_temperature(6) = -1. ! psuedo-adiabatic condition
          wall_temp_flag(6) = T

   total_pressure_ratio(3) = 1.02828 ! M = 0.2 from (p/p_t) isentropic equation
total_temperature_ratio(3) = 1.00800 ! M = 0.2 from (T/T_t) isentropic equation
  static_pressure_ratio(4) = 1.010   ! first guess
! solution functional
  dynamic_boundary_conditions = .true. ! Set to true to engage the back pressure controller
boundary_functional_name(4)    = 'back_pressure'
            field_point1(4,:) =  0.0,  0.01, 0.01 ! survey point
        update_frequency(4)    = 500
 /
```

Edit the [project_name].mapbc file

```
mapbc file

6
1    6663 z-symmetry
2    4000 wall
3    7011 inflow
4    7200 outflow solution functional
5    6662 y-symmetry
6    4000 no-slip wall
```

Create a new, separate namelist file: controller.nml

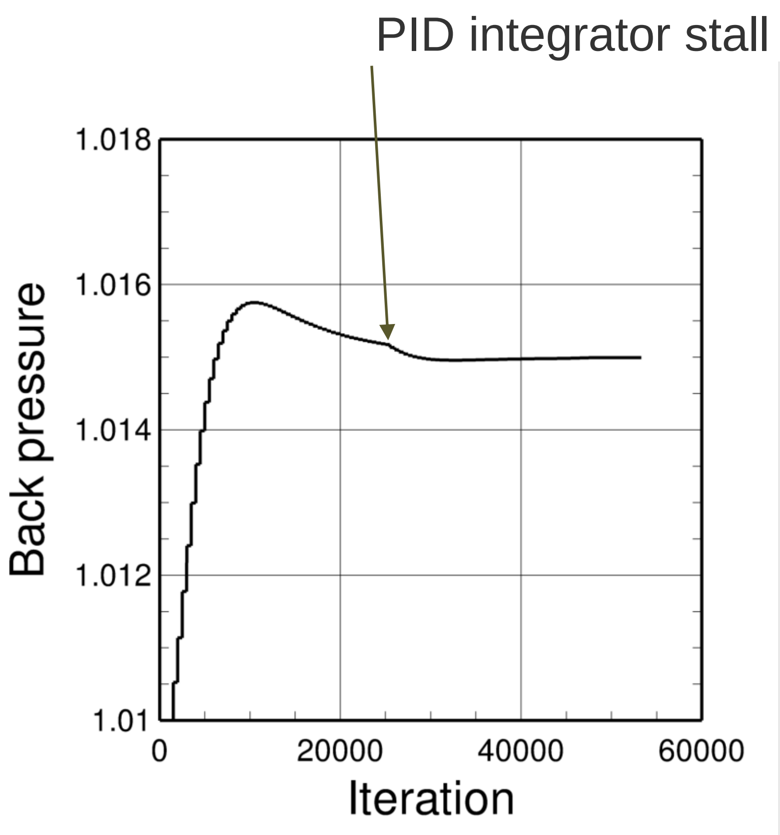
```
&tunnel_control
  number_of_controllers = 1 ! one controller to set up
  controller_points(4)  = 1 ! control boundary 4
          find_mach(4)  = .true. ! objective is Mach number
        target_mach(4)  = 0.20.  ! Mach number value
      initial_delay(4)  = 1000 ! to allow transients to die down
                 kp(4)  = 0.025  ! proportional
                 ki(4)  = 0.2e-5 ! integral
                 kd(4)  = 1.0e-6 ! derivative
/
```

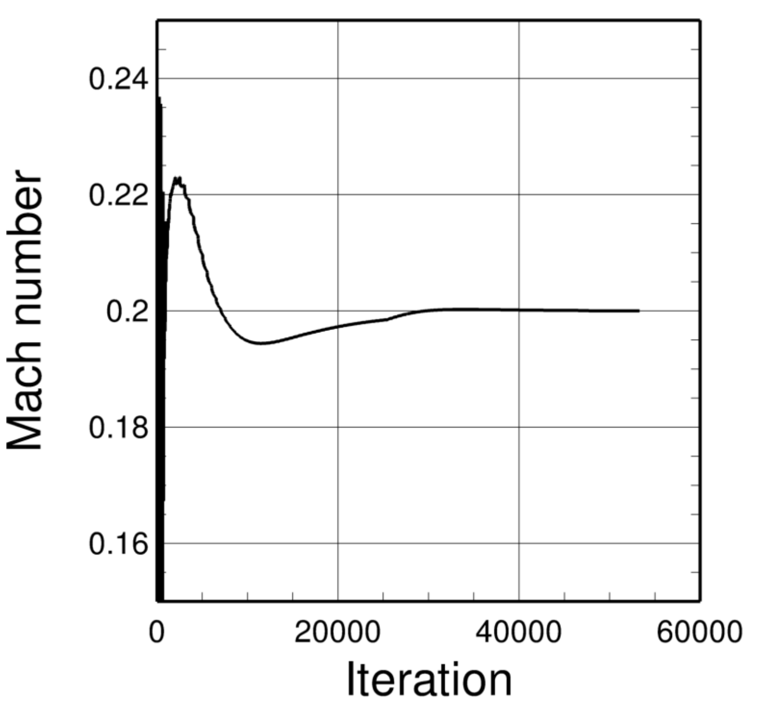
Analysis

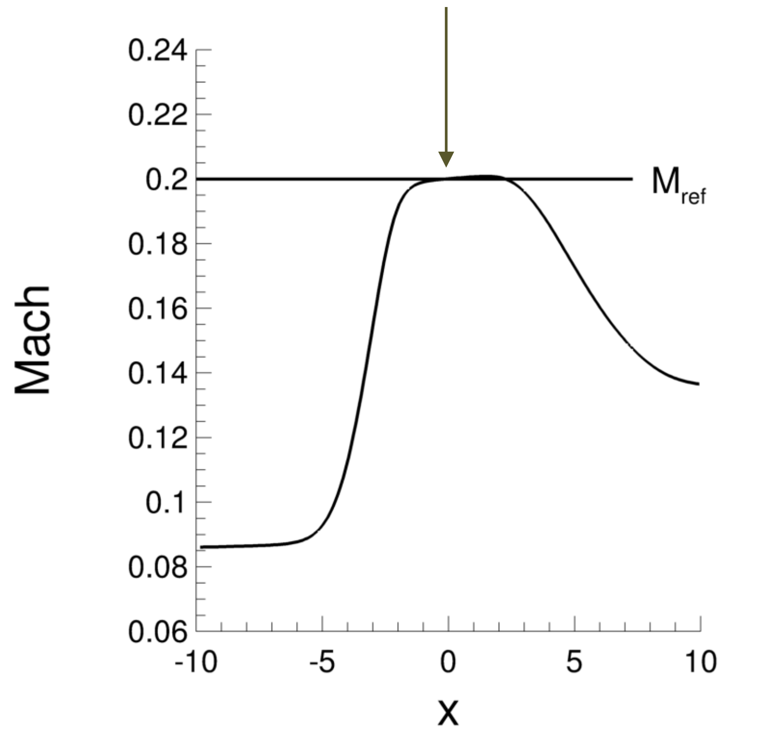
- Back pressure history
- Test section Mach number history
- Centerline Mach number

Back Pressure Controller

- Back pressure and Mach number output from [project_name]_[boundary]_controller.dat
- Centerline plot using the same sampling output file as previously discussed







Example of Porous Media Modeling

- Add a screen/honeycomb model in the upstream section
- Keep the back pressure controller active
- Examine back pressure history and centerline Mach number
- Plot centerline total pressure

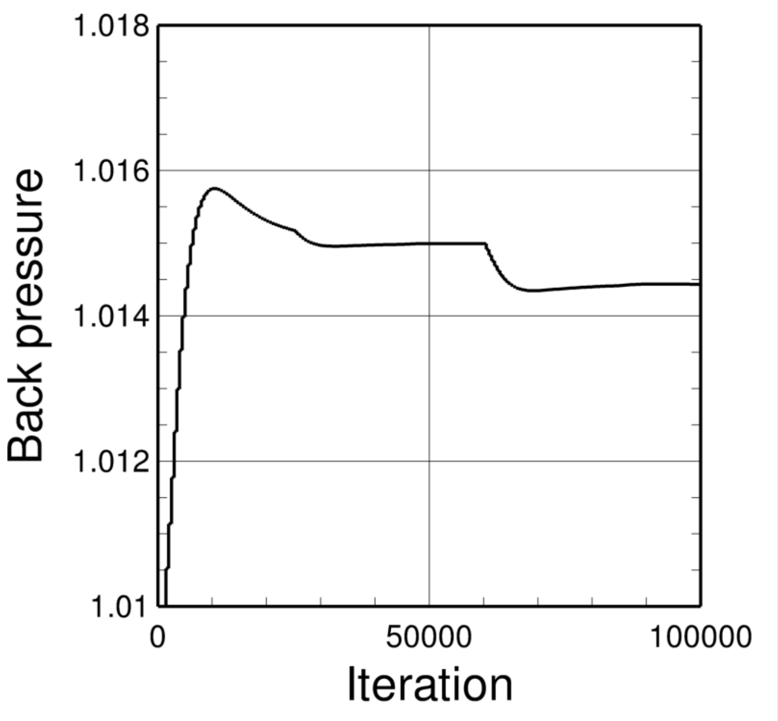
Honeycomb Model: update #2 fun3d.nml

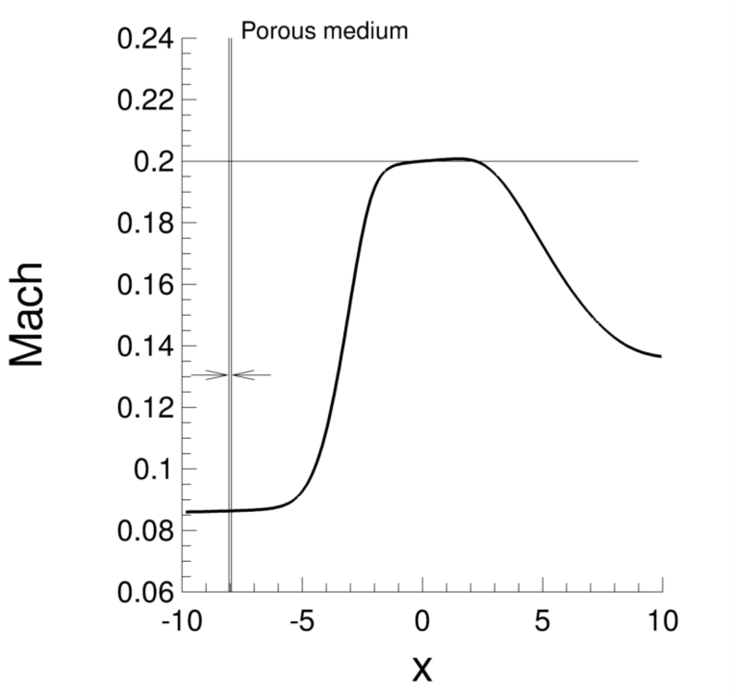
- Add a model of a honeycomb in the upstream portion of the tunnel
- &filters is part of the fun3d.nml file

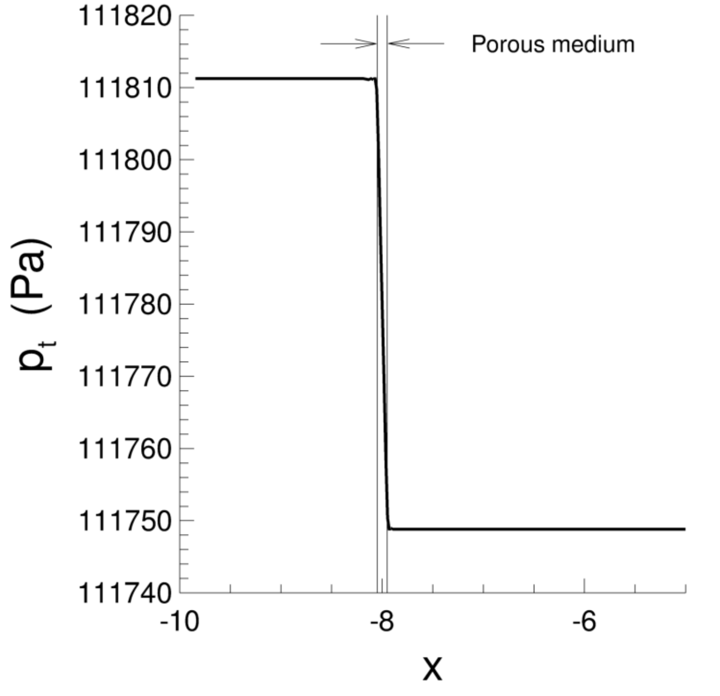
```
&filters
    passive_filter_flag = T  ! activate porous media model
      number_of_fences  = 1  ! list fences (volumetric) first
              gain(1)   = 1.0 ! default
      permeability(1)   = 1.0E+35 ! essentially no viscous part
   fence_thickness(1)   = 0.1 ! meters
 ! pressure loss factor (continuity, x-mom, y-mom, z-mom)
pressure_loss_factor(1,:) = 0.0 1.0  0. 0.  ! [1/m]
     fence_shape(1)     = 'hex' !
         corners(1,1,:) = -8.05    0.0    0.00
         corners(1,2,:) = -7.95    0.0    0.00
         corners(1,3,:) = -7.95    1.50   0.00
         corners(1,4,:) = -8.05    1.50   0.00
         corners(1,5,:) = -8.05    0.0    1.50
         corners(1,6,:) = -7.95    0.0    1.50
         corners(1,7,:) = -7.95    1.50   1.50
         corners(1,8,:) = -8.05    1.50   1.50
/
```

Restarting From Previous Solution

- To maintain the requested centerline Mach number, the back pressure must shift to compensate for the total pressure loss incurred by porous medium (honeycomb) model
- In this example, approximately 60 Pa total pressure drop simulated. The pressure loss is a function of the local dynamic pressure, $\Delta p_t / \Delta x = C\, q_{local}$



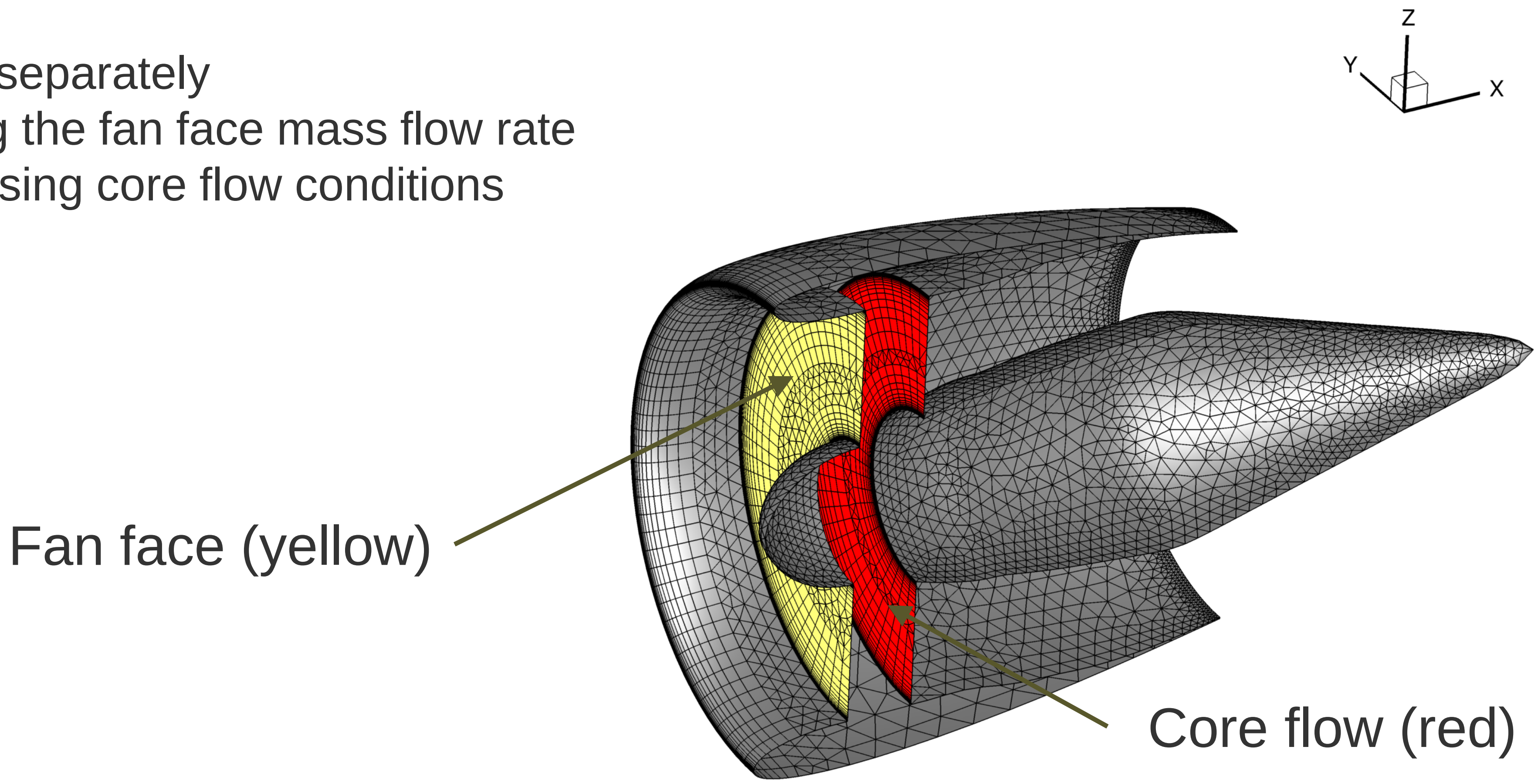




Three Simple Engine Simulations

Matching mass flow rates for a simple single stream engine

1. Set mass flow rates separately
2. Drive core flow using the fan face mass flow rate
3. Drive fan face flow using core flow conditions



Engine 1: Set mass flow rates separately

- Fan – boundary condition number 7031
- Core – boundary condition number 7036
- When using these boundary conditions, the input should be in units of grid units squared.
- From the reference conditions: $\rho_{\mathrm{ref}}\, c_{\mathrm{ref}}$ = 106.84 kg/s-m^2
- Suppose the target mass flow rate is $\dot{m}_{\mathrm{target}}$ = 2.183 kg/s
- To convert to FUN3D units: $\dot{m}_{\mathrm{target}}$ / $\rho_{\mathrm{ref}}\, c_{\mathrm{ref}}$ = 2.0435e-2 1/m^2
  = 31.675 1/inch^2
- Configure boundary condition namelist

```
&boundary_conditions
      grid_units = 'inches’
     massflow(1) = 31.675 ! core
     massflow(2) = 31.675 ! fan
  wall_temp_flag(3) = .true. ! no-slip
 wall_temperature(3) = -1.0
/
```

fun3d.nml for Engine 1

```
&global
 volume_animation_freq   = -1
 boundary_animation_freq = -1
/

&reference_physical_properties
  mach_number       = 0.72
  reynolds_number   = 136721 ! 1/[in]
  temperature       = 392.4
  temperature_units = 'Rankine'
/

&governing_equations
  viscous_terms = 'turbulent'
/

&turbulent_diffusion_models
  turbulence_model      = 'sa-neg'
  reynolds_stress_model = 'qcr2020'
  use_diff_element      = T
/

&spalart
  turbinf = 3.0
/

&debug
 weighted_lsq_diffusion_m = .true.
 weighted_lsq_diffusion_t = .true.
/

&code_run_control
  steps             = 50000
  restart_read      = 'off'
  stopping_tolerance = 1.0e-15
/

&linear_solver_parameters
  linear_projection = .false.
/

&nonlinear_solver_parameters
 time_accuracy      = 'steady'
 schedule_iteration = 1   250
 schedule_cfl       = 1.   50.
 schedule_cflturb   = 1.   20.
/

&inviscid_flux_method
   flux_limiter        = 'none'
   flux_construction    = 'roe'
/

&boundary_conditions
        grid_units = 'inches’
        massflow(1) = 31.675  ! core
        massflow(2) = 31.675  ! fan
  wall_temp_flag(3) = .true.  ! no-slip
 wall_temperature(3) = -1.0
/

&component_parameters

 allow_flow_through_forces = T
     number_of_components = 5

       component_count(1) = 1
       component_input(1) = '2'
        component_name(1) = 'inlet'
    component_symmetry(1) = 2.
```

fun3d.nml for Engine 1

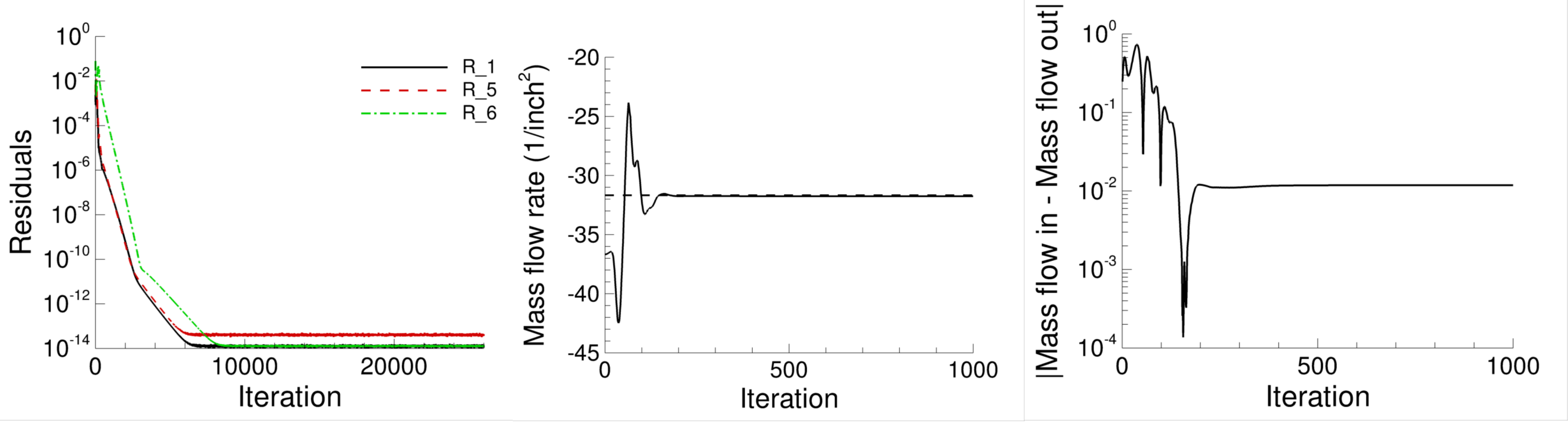
```
    component_count(2) = 1
    component_input(2) = '1'
     component_name(2) = 'exit'
  component_symmetry(2) = 2.

    component_count(3) = 2
    component_input(3) = '1,2'
     component_name(3) = 'balance'
  component_symmetry(3) = 2.

    component_count(4) = 3
    component_input(4) = '1,2,3'
     component_name(4) = 'total'
  component_symmetry(4) = 2.

    component_count(5) = 1
    component_input(5) = '3'
     component_name(5) = 'nacelle'
  component_symmetry(5) = 2.
/

&volume_output_variables
  export_to          = 'tec'
  x                  = .true.
  y                  = .true.
  z                  = .true.
  primitive_variables  = .true.
  mach               = .true.
  ptot               = .true.
  ttot               = .true.
/
```

```
&boundary_output_variables
  number_of_boundaries = 5
  boundary_list = '1-5'
  primitive_variables = .true.
  mach = .true.
  ttot = .true.
  ptot = .true.
  yplus = .true.
  uavg = .true.
  vavg = .true.
  wavg = .true.
/

&project
  project_rootname = 'engine_only'
/

&raw_grid
  grid_format   = 'aflr3'
  data_format   = 'stream'
  patch_lumping = 'family'
/

&force_moment_integ_properties
  area_reference  = 95.0
  x_moment_center = 0.0
  y_moment_center = 0.0
  z_moment_center = 0.0
/
```

Langley Research Center

Engine 1: Solution convergence

- Iterative convergence
- Mass flow rate achieved ≈ 0.2%
- Inlet and core flow balanced ≈ 0.6%



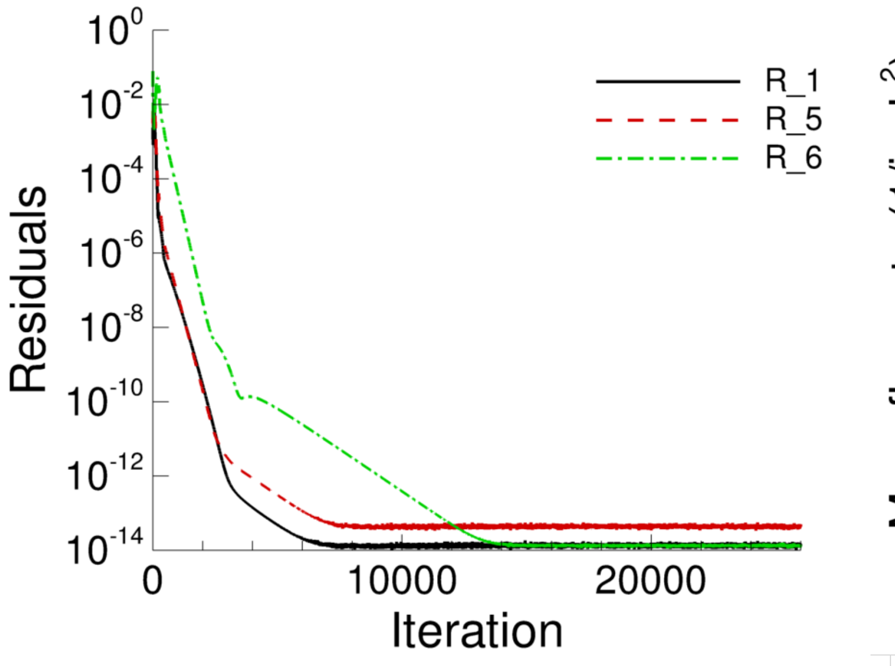
Engine 2a: Drive core flow via the fan face flow

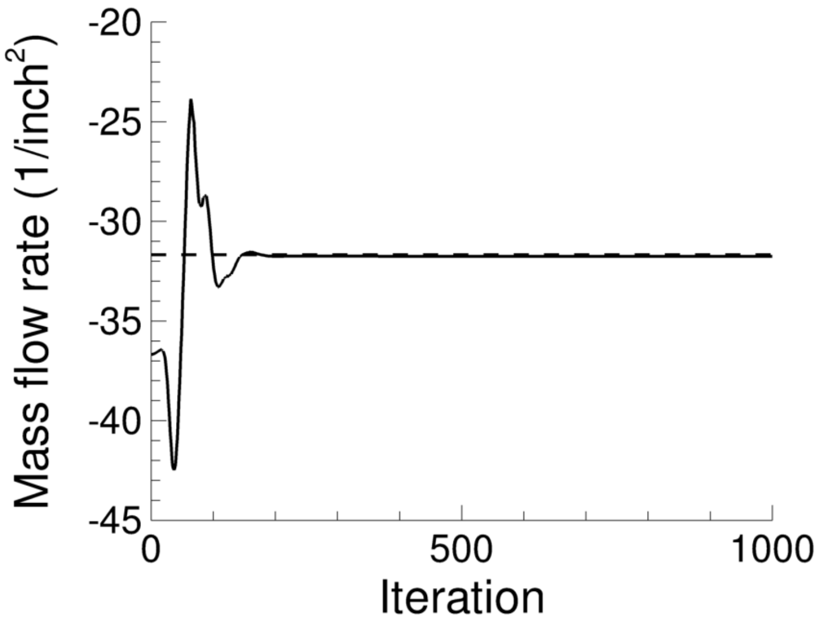
- Fan – boundary condition number 7031
- Core – boundary condition number 7036
- Fan boundary condition the same as Engine1
- No need to set core boundary condition value
- Configure mass flow rate matching

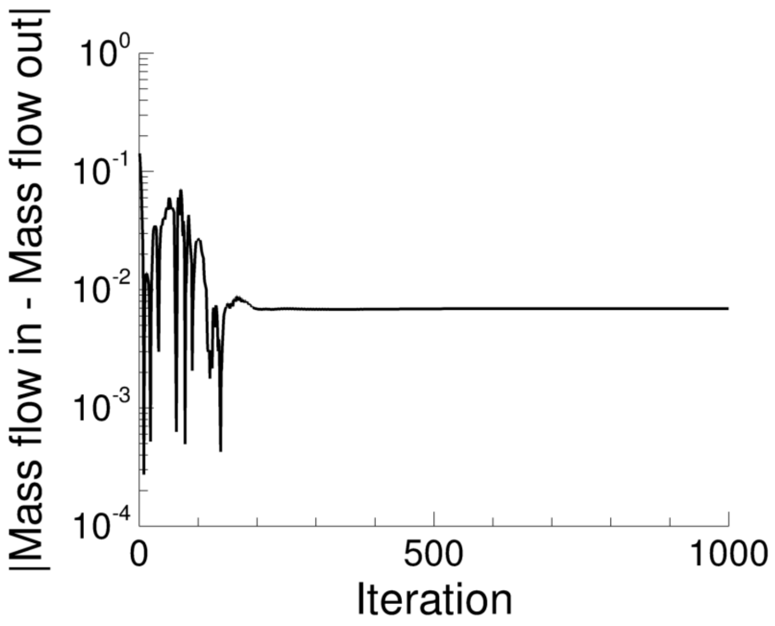
```
&boundary_conditions
    grid_units = 'inches'
    massflow(2) = 31.675 ! fan
  wall_temp_flag(3) = .true. ! no-slip
 wall_temperature(3) = -1.0

 number_of_engines    = 1   ! just one engine here
   engine_symmetry(1) = 2.0 ! grid is half plane symmetric
 number_of_streams(1) = 1   ! single stream engine
    inlet_bc(1) = 2   ! fan bc – driver
     core_bc(1) = 1   ! core bc - follower
   cycle_name(1) = 'massflow'  ! mass flow bc
/
```

Engine 2a: Solution convergence

- Iterative convergence
- Mass flow rate achieved ≈ 0.2%
- Inlet and core flow balanced ≈ 0.3%







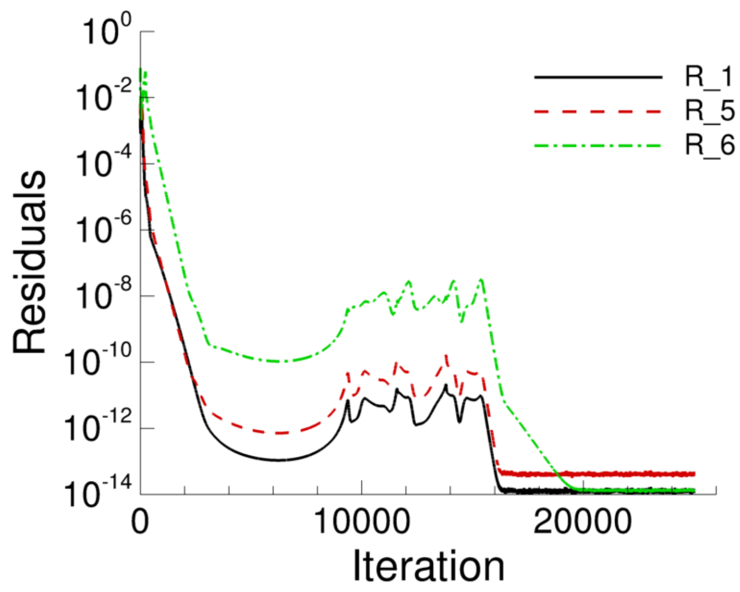
Engine 2b: Drive core flow via the fan face flow

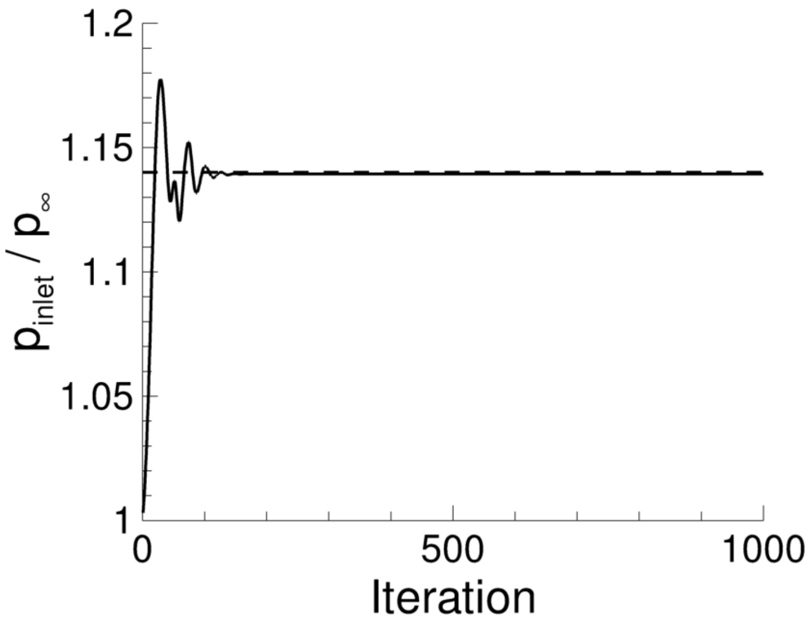
- Fan – boundary condition number 7012
- Core – boundary condition number 7036
- Fan boundary condition the same as Engine1
- No need to set core boundary condition value
- Configure mass flow rate matching

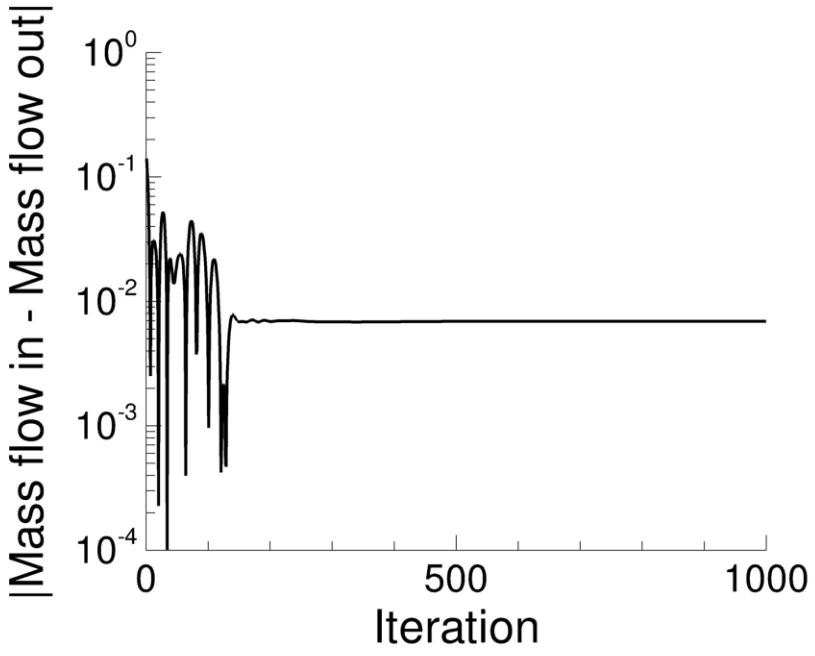
```
&boundary_conditions
            grid_units = 'inches’
  static_pressure_ratio(2) = 1.140 ! fan
       wall_temp_flag(3) = .true. ! no-slip
      wall_temperature(3) = -1.0

 number_of_engines    = 1   ! just one engine here
   engine_symmetry(1) = 2.0 ! grid is half plane symmetric
 number_of_streams(1) = 1   ! single stream engine
        inlet_bc(1) = 2   ! fan bc – driver
         core_bc(1) = 1   ! core bc - follower
       cycle_name(1) = 'massflow’  ! mass flow bc
/
```

Engine 2b: Solution convergence

- Iterative convergence
- Static pressure ratio achieved ≈ 0.05%
- Inlet and core flow balanced ≈ 0.3%







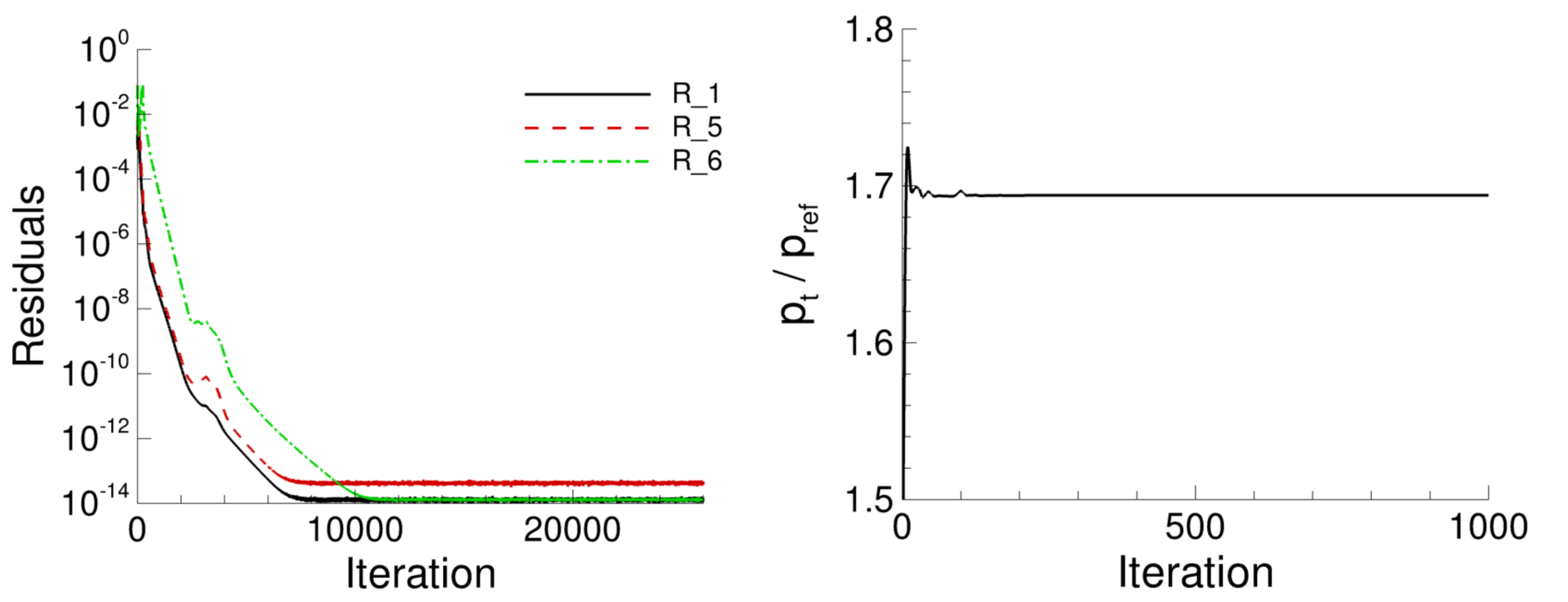
Engine 3: Drive fan flow via the core flow

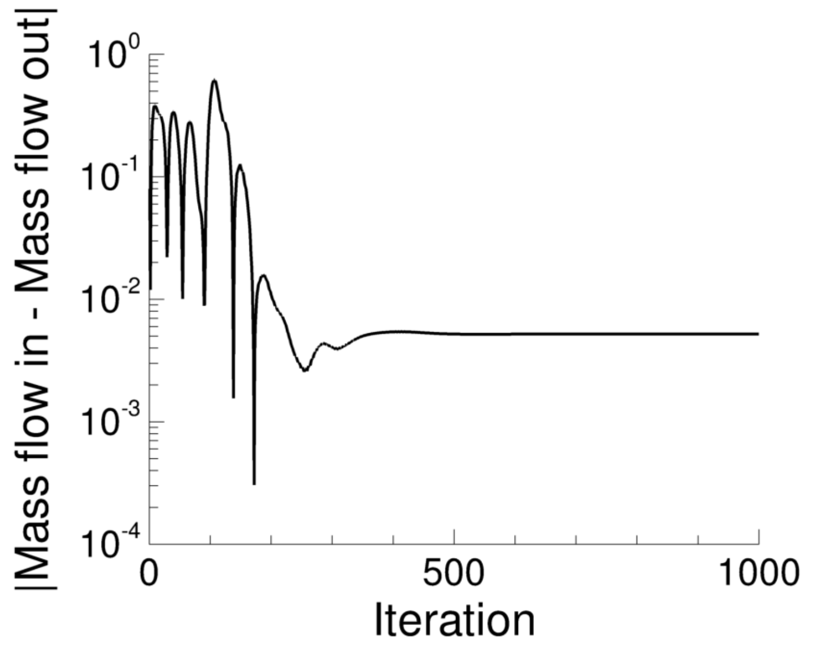
- Fan – boundary condition number 7031
- Core – boundary condition number 7011
- Set the core total pressure and temperature
- Configure mass flow rate matching

```
&boundary_conditions
      grid_units = 'inches’
   total_pressure_ratio(1) = 1.6949  ! core
  total_temperature_ratio(1) = 1.1627  ! core
        massflow(2) = 30.0 ! initial guess
      wall_temp_flag(3) = .true. ! no-slip
     wall_temperature(3) = -1.0

 number_of_engines    = 1   ! just one engine here
  engine_symmetry(1) = 2.0 ! grid is half plane symmetric
 number_of_streams(1) = 1   ! single stream engine
     inlet_bc(1) = 2   ! fan bc – driver
      core_bc(1) = 1   ! core bc – follower
    cycle_name(1) = 'total_conditions'
/
```

Engine 3: Solution convergence

- Iterative convergence
- Total pressure ratio achieved ≈ 0.05%
- Inlet and core flow balanced ≈ 0.3%





Today’s Features Review

- Sketched out of some boundary conditions that can be used for internal flows
- F & M and flow tracking using component parameters
- An example using the back pressure controller
- An example of using the porous media model
- Discussed three simple “turbofan” engine models

Short List of References

- “Inflow/Outflow Boundary Conditions with Application to FUN3D,” NASA-TM-2011-208444
- “Boundary Condition Study for the Juncture Flow Experiment in the NASA Langley 14x22-Foot Subsonic Wind Tunnel,” AIAA-2017-4126
- “FUN3D and USM3D Analysis of the Propulsion Aerodynamic Workshop 2018 S-duct Test Case,” AIAA-2019-3848
- “Setting Boundary Conditions For Slotted Throat Wind Tunnels Using Calorically Imperfect Gas Assumptions,” AIAA-2022-0807