Updating and optimizing a CFcase file for the simulation of a hypersonic flow through a 2D axisymmetric cone with LTE assumption

Sana Amri[1]

[1]Master's Student in Applied Mathematics and Scientific Computing Methods
University Paris XIII - Engineering School Sup Galilée

Internship: April - September 2018

Outline

Physical Modeling

- 2D axisymmetric blunt cone: radius 7 mm
- Chemical perspective:
  - Flow: a Neutral 5-species air $\rightarrow O, N, O_2, N_2, NO$
  - Velocity: Hypersonic (Mach:$\approx$ 6.3)

Assumptions

1- The flow is a continuum medium

- The Knusden number must be strictly inferior than 0.2 (To apply the Navier Stokes equations)

2- The dissociating gas is in chemical equilibrium

- Damkohler number : $Da^c = \frac{\tau_f}{\tau_c} \rightarrow \infty$

3- The gas mixture is in Local Thermodynamic Equilibrium

- The chimical equilibrium composition, can be computed directly for given values of:
    - pressure,
    - temperature
    - finite (fixe) or variable (by adding continuity equations to the model) elemental fractions.

Navier-Stokes Equations

By integrating on a control volume ϖ_i and after applying the divergence formula, we read:

$$\underbrace{\frac{d}{dt}\int_{\varpi_i} \boldsymbol{U} dw}_{\substack{transient\\ term}} + \underbrace{\int_{\Sigma_i} \boldsymbol{F}^c \,\boldsymbol{.}\, \boldsymbol{n} ds}_{\substack{convective\\ term}} = \underbrace{\int_{\Sigma_i} \boldsymbol{F}^d \,\boldsymbol{.}\, \boldsymbol{n} ds}_{\substack{diffusif\\ term}} + \underbrace{\int_{\varpi_i} \boldsymbol{S} dw}_{\substack{source\\ term}} \tag{1}$$

with $\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}$ conservative variables

Gouverning equations for the chemical mixture (LTE)

- **L**ocal **T**hermodynamics **E**quilibrium with **F**ixed **E**lemental **F**ractions
  We assume the elemental composition constant throughout the flow and equal to the free stream values:

$$Y_O = 0.21 \qquad Y_N = 0.79$$

  Therefore, $y_s := y_s(p, T)$, $s \in \{N_2, O_2, NO, N, O\}$

- **L**ocal **T**hermodynamics **E**quilibrium with **V**ariable **E**lemental **F**ractions

$$\begin{cases} y_s := y_s(p, T, Y_O, Y_N) \\ \dfrac{\partial Y_O}{\partial t} + \nabla \cdot (\rho Y_O \boldsymbol{u}) = -\nabla \cdot \boldsymbol{J}_O \\ \dfrac{\partial Y_N}{\partial t} + \nabla \cdot (\rho Y_N \boldsymbol{u}) = -\nabla \cdot \boldsymbol{J}_N \end{cases}$$

  J_e: mass diffusion flux of species e $\quad$ $\boldsymbol{u}$: mass-averaged mixture velocity
  We solve simultaneously elemental continuity equations for the oxygen and the nitrogen. The missing source term in both equations translate the fact that no new elements are generated in the mixture.

[.CFcase] File

We received an old CFcase file from Fernando MiroMiro that we updated and adapted for our case.

For more detail about what has been done in this executable file so far:

`https://github.com/SanaAmri/LTE_NS_Cone/blob/master/README`

Parameters

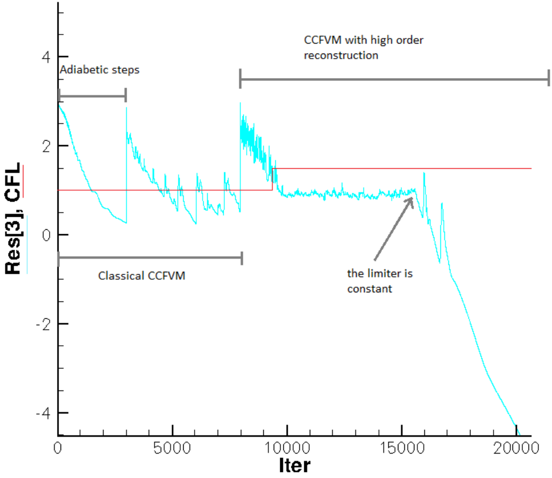
Parameters	values
Temperature	1192K
Pressure	6880 Pa
Speed of sound computed with the above parameters	674.32 (mutationpp)
Minimum Velocity	4244m/s
Mach	6.29
Isothermal wall	293K
Air mixture	N_2, O_2, NO, N, O

Numerical methods

Physical Model type	NavierStokes2D
Mutation version	Mutation2OLD
Space Method	**CellCenteredFVM**
Space Method accuracy	2^{nd} order reconstruction
Convective flux	AUSMPlus2D flux splitter
Diffusive flux	NavierStokes - Nodal extrapolation
SourceTerm	NavierStokes2DAxiST
limiter	Venkatakrishnan's limiter
Mesh	quads, unrefined, nb_elt : 10K

Convergence with high order reconstruction

F I R S T A T T E M P T: Patterns in the convergence for the temperature residue.



Current improvement

Improving the speed of the convergence in 2^{nd} order of accuracy (of the temperature residue) by testing different kind of algorithms (functions) for:

- the CFL parameter
- the transition in 1rst to 2nd order accuracy
- the limiter (when to freeze it during the convergence)

RESULTS: After the different tests (next slides), the residue of the temperature converge after only 9000 iterations (instead of 20K) in second order accuracy.

Current improvement

Two strategies for increasing the speed of the convergence of the temperature residue :

1. Compute from scratch the case with the classical CCFVM ($\Rightarrow O(h)$) and try to tend to the criteria of convergence (10^{-4} for the temperature residue) without reaching it. Then , make the transition in second order accuracy $O(h^2)$ by activating the high order reconstruction (gradient=1.0). That way, we expect that the convergence in 2^{nd} order accuracy will go faster because the residue will be very close to zero during the transition. The configurations for the CFL and the limiter are based on the results obained. **[See test functions 1A,1B,1C and 1D - next slides]**
2. Start from scratch to run with the classical CCFVM, and make the transition in second order accuracy right after the adiabatic steps. Then adapt the CFL and the limiter parameters based on the results obained. **[See test functions 2A and 2B - next slides]**

Function test 1A: Algorithm

Function test 1A

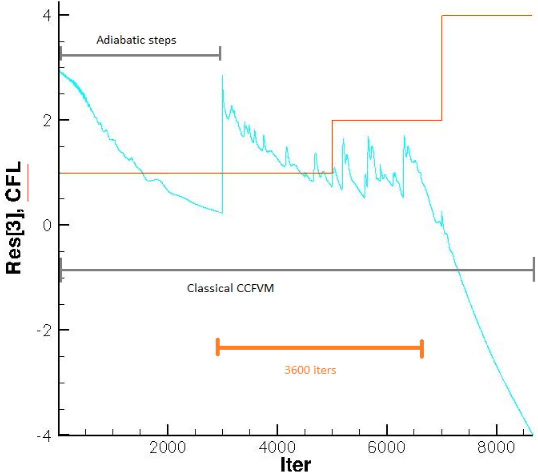
```
Gradient= 0.  → Classic CCFVM, 1rst order accuracy
WHILE temperature residue > 10^-4
      IF(iter<5000)
           CFL=1.0
      ELSE IF(5000≤iter<7000)
           CFL=2.0
      ELSE IF(7000≤iter<10000)
           CFL=4.0
      ELSE IF(10000≤iter<12000)
           CFL=1.0
      ELSE IF(12000≤iter<16000)
           CFL=2.0
           IF(iter=12000)
                Gradient=1.  → Transition to second order accuracy
           END IF
      ELSE (16000≤iter)
          CFL=4.0
           IF(iter=18000)
                → Freeze the factor limiter from the piecewise
                  polynomials approximation of each cells
           END IF
     END IF
END WHILE
```

Function test 1A: Expectations

G O A L S:

- Reduce the oscillating part after the end of the 3000 adiabatic iterations (These steps are usefull to detache the shock from the boundaries).
- Increase the order of accuracy of the scheme when the residue is close to 10^{-4} (convergence value) without reaching it. That way, we expect that the convergence in 2^{nd} order accuracy will go faster because the residue will be very close to zero during the transition.

Function test 1A: Results



Function test 1A: Observations

O B S E R V A T I O N S:

- The residue converged (iter=8674, $\simeq$13h) before the attempt to reach a second order accuracy (gradient=1. at iter 12000 in the previous algorithm)
- We reduced (relatively to the **first try**) the oscillating part after the adiabatic steps (5000 $\rightarrow$ 3600)

Function test 1B: Algorithm

Function test 1B

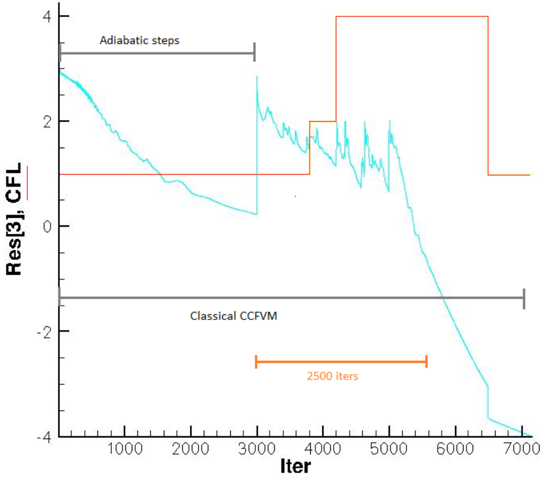
Gradient= 0. $\rightarrow$ Classic CCFVM, 1^{rst} order accuracy
WHILE temperature residue $> 10^{-4}$
    IF(iter<3800)
        CFL=1.0
    ELSE IF(3800$\leq$iter<4200)
        CFL=2.0
    ELSE IF(4200$\leq$iter<6500)
        CFL=4.0
    ELSE IF(6500$\leq$iter<7000)
        CFL=1.0
    ELSE IF(7000$\leq$iter<9000)
        CFL=2.0
        IF(iter=8500)
            Gradient=1. $\rightarrow$ Transition to second order accuracy
        END IF
    ELSE (9000$\leq$iter)
        CFL=4.0
        IF(iter=12000)
            $\rightarrow$ Freeze the factor limiter from the piecewise
            polynomials approximation of each cells
        END IF
    END IF
END WHILE

Function test 1B: Expectations

G O A L S:

- Reduce (relatively to function 1A) the oscillating part after the end of the 3000 adiabatic iterations.
- Change to second order accuracy sooner than the function 1A to avoid the complete convergence in first order accuracy.

Function test 1B: Results



function test 1B: Observations

O B S E R V A T I O N S:

- Again, the residue converged (iter=7176, $\simeq$11,5h) before the attempt to reach a second order accuracy (gradient=1. at iter 8000 in the algorithm 1B).
- As expected, we reduced (relatively to the function 1A) the oscillating part after the adiabatic steps (3600iters $\rightarrow$ 2500iters).

function test 1C: Algorithm

Function test 1C

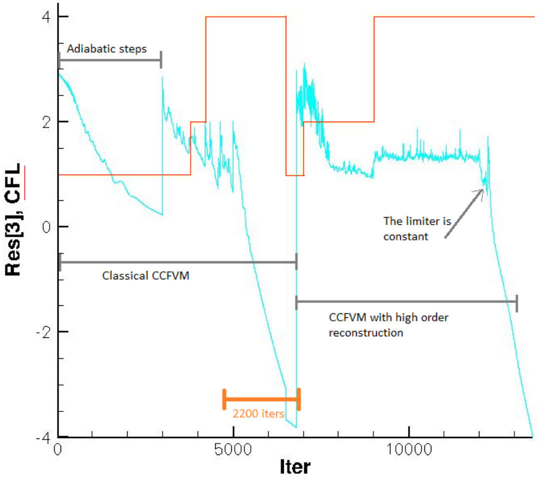
```
Gradient= 0. → Classic CCFVM, 1^rst order accuracy
WHILE temperature residue > 10^-4
      IF(iter<3800)
          CFL=1.0
      ELSE IF(3800≤iter<4200)
          CFL=2.0
      ELSE IF(4200≤iter<6500)
          CFL=4.0
      ELSE IF(6500≤iter<7000)
          CFL=1.0
          IF(iter=6800)
              Gradient=1. → Transition to second order accuracy
          END IF
      ELSE IF(7000≤iter<9000)
          CFL=2.0
      ELSE (9000≤iter)
         CFL=4.0
          IF(iter=12000)
              → Freeze the factor limiter from the piecewise
                polynomials approximation of each cells
          END IF
     END IF
END WHILE
```

function test 1C: Expectations

G O A L S:

- Make the transition to a second order accuracy before the residue reaches 10^{-4} .

function test 1C: Results



function test 1C: Observations

O B S E R V A T I O N S:

- We manage to make the transition in a higher order accuracy, but we have a big gap that we didn't expect.
- In second order, we can see that the limiter has a big impact on the convergence of the temperature residue.

```
Total Number Iter: 13537 Reached Residual: -4.00207 and took: 13 h 54 min 45.0284 sec
```

function test 1D: Algorithm

Function test 1D

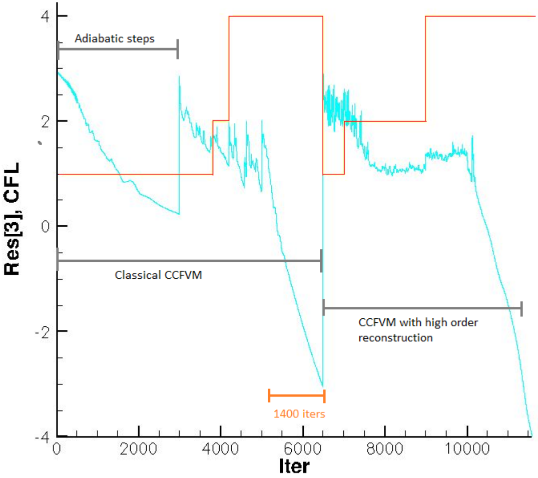
```
Gradient= 0. → Classic CCFVM, 1^rst order accuracy
WHILE temperature residue > 10^-4
      IF(iter<3800)
           CFL=1.0
      ELSE IF(3800≤iter<4200)
           CFL=2.0
      ELSE IF(4200≤iter<6500)
           CFL=4.0
      ELSE IF(6500≤iter<7000)
           CFL=1.0
           IF(iter=6500)
                Gradient=1. → Transition to second order accuracy
           END IF
      ELSE IF(7000≤iter<9000)
           CFL=2.0
      ELSE (9000≤iter)
          CFL=4.0
           IF(iter=10000)
                → Freeze the factor limiter from the piecewise
                  polynomials approximation of each cells
           END IF
      END IF
END WHILE
```

function test 1D: Expectations

G O A L S:

- We would like to minimize (relatively to the function 1C), the gap during the transition part. The trick is to change the gradient to 1. **and** the CFL to 1 at the same iteration.

function test 1D: Results



function test 1D: Observations

O B S E R V A T I O N S:

- We manage to reduce the transition to a higher order accuracy, but the gap is still too high and add many unnecessary steps of convergence.
- In the first attempt, where we configurated the parameter manually (interfile), we changed the gradient to 1. without waiting for the convergence in first order of accuracy. This strategy avoided the discontinuity that we obtained here. That is why I changed the strategy of convergence for the next test functions.

```
Total Number Iter: 11592 Reached Residual: -4.00108 and took: 12 h 28 min 38.6879 sec
```

Function test 2A: Algorithm

Function test 2A : (we don't wait to converge in first order accuracy)

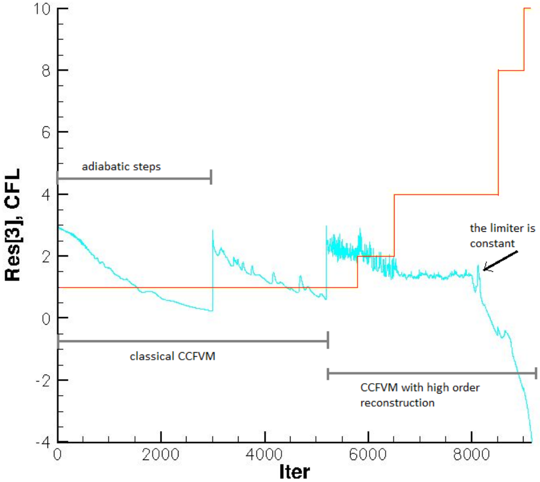
1\. Gradient= 0. $\rightarrow$ Classic CCFVM, 1^{rst} order accuracy
2\. **WHILE** temperature residue $> 10^{-4}$
3\.   **IF**(iter$<$5800)
4\.   CFL=1.0
5\.   **IF**(iter=5200)
6\.   Gradient=1. $\rightarrow$ Transition to second order accuracy
7\.   **END IF**
8\.   **ELSE IF**(5800$\leq$iter$<$6500)
9\.   CFL=2.0
10\.   **ELSE IF**(6500$\leq$iter$<$8500)
11\.   CFL=4.0
18\.   **IF**(iter=8000)
19\. $\rightarrow$ Freeze the factor limiter from the piecewise
20\.   polynomials approximation of each cells
21\.   **END IF**
12\.   **ELSE** (8500$\leq$iter$<$9000)
13\.   CFL=8.0
16\.   **ELSE** (9000$\leq$iter)
17\.   CFL=10.0
22\.   **END IF**
23\. **END WHILE**

Function test 2A: Expectations

G O A L S:

- Based on the observations made in the first attempt, we would like to reduce the oscillating part after the 3000 iteration.
- Also, when we reach the second order accuracy, we would like to freeze the limiter sooner to reduce the oscillating part and reach the downward slope faster.

Function test 2A: Results



Function test 2A: Observations

O B S E R V A T I O N S:

- The expectations are reached, but we can reduce even more the oscillating parts.

```
Total Number Iter: 9170 Reached Residual: -4.00844 and took: 10 h 22 min 30.3091
 sec
```

Function test 2B: Algorithm

Function test 2B :

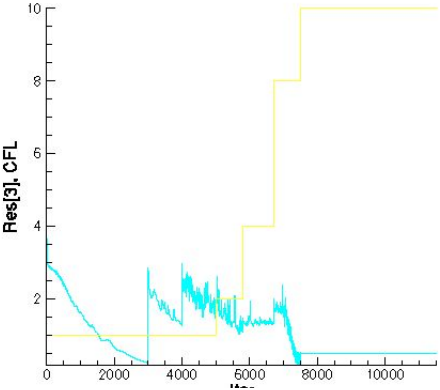
```
1.  Gradient= 0.  → Classic CCFVM, 1^rst order accuracy
2.  WHILE temperature residue > 10^-4
3.        IF(iter<5000)
4.            CFL=1.0
5.             IF(iter=4000)
6.                  Gradient=1.  → Transition to second order accuracy
7.             END IF
8.        ELSE IF(5000≤iter<5800)
9.            CFL=2.0
10.       ELSE IF(5800≤iter<6700)
11.           CFL=4.0
18.           IF(iter=7000)
19.                → Freeze the factor limiter from the piecewise
20.                  polynomials approximation of each cells
21.           END IF
12.      ELSE IF(6700≤iter<7500)
13.           CFL=8.0
16.      ELSE (7500≤iter)
17.          CFL=10.0
22.      END IF
23. END WHILE
```

Function test 2B: Expectations

G O A L S:

- Optimizing the previous results obtained using the function 2A.

Function test 2B: Results



Function test 2B: Observations

O B S E R V A T I O N S:

- When we tried to reduce the oscillating parts the residue tend to stay constant. So the best result that we have is the function 2A.

- Andrea Lani's thesis
- https://github.com/andrealani/COOLFluiD