



PROJET INTÉGRATEUR III

AER3900 - HIVER 2022

Guide d'utilisation des scripts calculant le taux de déformation au point d'extinction de flammes.

Par

Rani Naaman

Rani.naaman@polymtl.ca

18 avril 2022

Table des matières

1	Détails des scripts de la solution retenue	1
1.1	Lecture du fichier <i>data.json</i>	1
1.2	Établissement des simulations en parallèle	3
1.3	Pré-simulation : Établissement d'une flamme libre	3
1.4	Simulation : Trouver le point de convergence et calculer le taux de défor- mation au point d'extinction	5
1.5	Lecture des résultats	5
1.5.1	Lecture du fichier excel	6
1.5.2	Lecture du fichier hdf5	7
2	Validation et analyse des scripts	7
3	Informations additionnelles	8
3.1	Librairies utilisées	8
3.2	Lecture des graphiques	9

Liste des figures

1	Exemple de plusieurs simulation dans <i>data.json</i> [4].	2
2	Exemple de bar de progression du fichier <i>main_PI3.py</i> [4].	3
3	Exemple de résultats dans un fichier excel [4].	6
4	Exemple de résultats dans un fichier hdf5 [4].	7
5	Tableau de comparaison entre les résultats du document DNS of hydrogen and methane laminar flames et les résultats obtenus avec les nouveaux scripts [4]	8
6	Exemple de graphique enregistré pour une simulation donnée[4]	9

1 Détails des scripts de la solution retenue

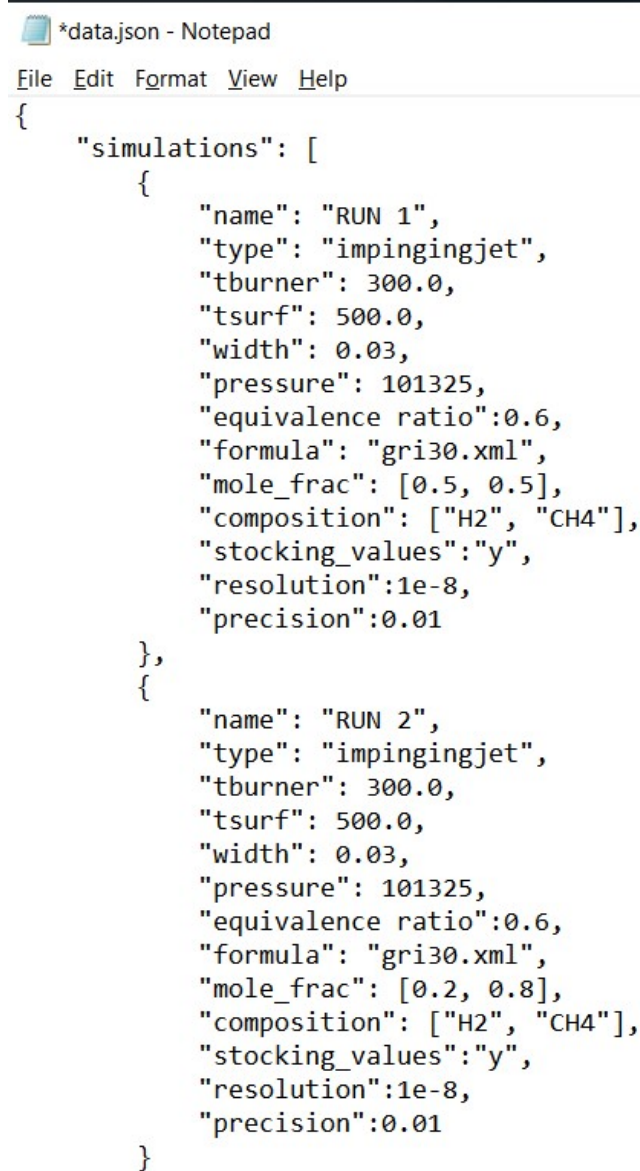
Alors, à l'aide d'un exemple d'algorithme [6], la méthode de la bisection a été implémentée par recommandation du client (voir la prochaine section pour les détails de son fonctionnement). De plus, la fonction `ThreadPoolExecutor(max_workers=n_cores)` a été choisie, car cette méthode de parallélisme fonctionnait bien avec les scripts. Il est important de noter que les scripts fonctionnent avec la version 2.5.1 de *Cantera*.

Les sous-sections ci-dessous représentent comment les scripts permettent de calculer le taux de déformation au point d'extinction de flammes.

1.1 Lecture du fichier *data.json*

Le premier fichier important, soit *data.json*, est représenté dans l'illustration ci-dessous. Afin de commencer une simulation, l'utilisateur doit réaliser les étapes suivantes :

1. Définir le nom de la simulation dans la section "name".
2. Définir le type de la simulation dans "type". Pour l'instant, les scripts fonctionnent seulement pour les simulations «ImpingingJet». Soit une flamme qui se repend vers une surface non réactive.
3. Définir la température du brûleur (K) dans la section "tburner". À titre d'exemple, une plage entre [300 ; 1500] K serait attendue pour cette variable.
4. Définir la température de la surface non réactive (K) dans la section "tsurf". À titre d'exemple, une plage entre [300 ; 1500] K serait attendue pour cette variable.
5. Définir la distance entre le brûleur et la paroi (m) dans la section "width". À titre d'exemple, une plage entre [0.01 ; 0.1] m serait attendue pour cette variable.
6. Définir la pression de la simulation (pa) dans la section "pressure". À titre d'exemple, une plage entre [1 ; 10] atm (1 atm= 101325 pa) serait attendue pour cette variable.
7. Définir le ratio d'équivalence de la simulation dans la section "equivalence ratio". À titre d'exemple, une plage entre [0,4 ; 1,4] serait attendue pour cette variable.
8. Définir le mécanisme de calcul de la simulation dans la section "formula". À titre d'exemple, les mécanismes *Sd_mec.cti*, *gri30.xml* et *Reaction.cti*.



```

{
  "simulations": [
    {
      "name": "RUN 1",
      "type": "impingingjet",
      "tburner": 300.0,
      "tsurf": 500.0,
      "width": 0.03,
      "pressure": 101325,
      "equivalence ratio": 0.6,
      "formula": "gri30.xml",
      "mole_frac": [0.5, 0.5],
      "composition": ["H2", "CH4"],
      "stocking_values": "y",
      "resolution": 1e-8,
      "precision": 0.01
    },
    {
      "name": "RUN 2",
      "type": "impingingjet",
      "tburner": 300.0,
      "tsurf": 500.0,
      "width": 0.03,
      "pressure": 101325,
      "equivalence ratio": 0.6,
      "formula": "gri30.xml",
      "mole_frac": [0.2, 0.8],
      "composition": ["H2", "CH4"],
      "stocking_values": "y",
      "resolution": 1e-8,
      "precision": 0.01
    }
  ]
}

```

FIGURE 1 – Exemple de plusieurs simulation dans *data.json* [4].

9. Définir les fractions molaires des composantes de la flamme dans la section "mole_frac".
Il est important à noter que la somme des fractions molaires doit être égal à 1. Sinon, la simulation sera erronée.
10. Définir les composantes de la flamme dans la section "composition". Il est important à noter que l'ordre des composantes est la même que l'ordre des fraction molaire. Pour l'instant, les scripts fonctionnent seulement pour les compositions d'ammoniac (NH_3), d'hydrogène (H_2) et de méthane (CH_4).
11. Choisir si les scripts enregistrent les conditions de simulation dans la section "sto-

cking_values". Si l'utilisateur souhaite enregistrer les valeurs, il faut écrire "y" dans cette section. Sinon, il faut écrire "n".

12. Définir la résolution de la simulation dans la section "resolution". À titre d'exemple, une plage entre $[1e-4; 1e-8]$ serait attendue pour cette variable.
13. Définir la précision de la convergence de la simulation dans la section "precision". À titre d'exemple, une plage entre $[0.1; 0.001]$ serait attendue pour cette variable.
14. Si l'utilisateur le souhaite, celui-ci peut rajouter de nouvelles simulations dans la section "simulations" en rajoutant une nouvelle section après la fermeture des accolades et répétant les étapes ci-dessus.

1.2 Établissement des simulations en parallèle

Le fichier *main_PI3.py* représente le code principal à exécuter. Tout d'abord, ce fichier va lire les conditions de simulations définies dans *data.json* et vérifier le contenu. Ensuite, le script va lancer au maximum n simulations en parallèle, où n est le nombre de coeurs de l'ordinateur. Cette limite est introduite pour ne pas excéder les limites de l'ordinateur.

Lorsque les simulations sont lancées, une bar de progression est présentée dans le terminal pour suivre le déroulement des simulations. Soit la figure ci-dessous représente un exemple de bar de progression du fichier *main_PI3.py*.

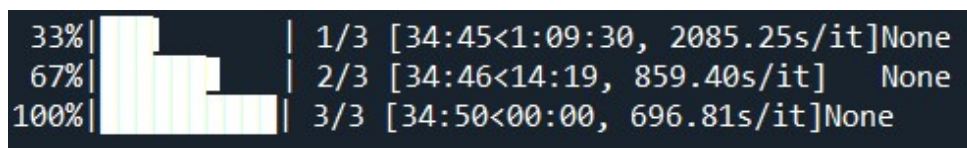


FIGURE 2 – Exemple de bar de progression du fichier *main_PI3.py* [4].

Sur la barre de progression, réalisée avec la librairie *tqdm*, il est possible de voir le nombre total de simulations, le pourcentage de complétion et le temps de compilation de chaque simulation.

1.3 Pré-simulation : Établissement d'une flamme libre

Dans le but de converger les simulations vers le point d'extinction, il faut tout d'abord trouver le point d'extinction. Pour se faire, il faut préalablement trouver une vitesse en

[kg/(m² * s)] qui sera it rer vers la vitesse maximal, soit celle au point d’extinction.

Le fichier *freeflame.py* permet de simuler une flamme libre avec les conditions  tablies par l’utilisateur dans *data.json* pour extraire une vitesse et l’inverse du temps caract ristique de de la flamme, soit une valeur de r f rence en [1/s] pour les simulations ayant des parois limitant les flammes.

D’abord, le script calcule la masse molaire totale de la composition avec l’ quation suivante :

$$M_{tot} = \frac{\sum_{i=1}^n (M_i * \frac{x_i}{\phi}) + M_{air}}{\sum_{i=1}^n (\frac{x_i}{\phi}) + n_{air}} \quad (1)$$

Ensuite, le script calcule la densit  de la flamme libre avec l’ quation suivante :

$$\rho = \frac{P * M_{tot}}{R * T} \quad (2)$$

Apr s, le script calcul l’ paisseur de la flamme avec l’ quation suivante, tir e d’un exemple de simulation du Centre Europ en de recherche et de formation avanc e en calcul scientifique [3] :

$$e = \frac{T_{max} - T_{min}}{\nabla T} \quad (3)$$

Ensuite, l’inverse du temps caract ristique de de la flamme est calcul e avec l’ quation suivante :

$$t_{inv} = \frac{v_{norm}}{e} \quad (4)$$

Par la suite, la vitesse de la flamme libre en [m/s] est calcul e avec l’ quation suivante :

$$v = \frac{a}{d} \quad (5)$$

Finalement, la vitesse en [kg/(m² * s)] de la flamme libre est calcul e avec l’ quation suivante :

$$\dot{m} = \rho \cdot v \quad (6)$$

La valeur de la vitesse en [kg/(m² * s)] (\dot{m}) et celle du taux de d formation (a) sont retenues et transf rer au fichier *main_P13.py*.

1.4 Simulation : Trouver le point de convergence et calculer le taux de déformation au point d'extinction

Le fichier *ImpingingJet.py* permet de simuler une flamme axisymétrique qui se propage vers une surface non réactive avec les conditions établies par l'utilisateur dans data.json pour extraire le taux de déformation au point d'extinction.

Afin de converger la simulation vers la vitesse maximale, soit celle au point d'extinction, le script utilise ma méthode de la bisection pour trouver la solution, soit avec la logique suivante :

1. Une valeur minimale de vitesse est choisie à $(\dot{m}/1000)$ et une vitesse maximale égale à (\dot{m}) .
2. Le nombre d'itération est choisi pour avoir une précision donnée par l'utilisateur sur la vitesse finale avec l'équation suivante [1] : $N > \log_2(\dot{m}_{max} - \dot{m}_{min})/\epsilon$, où N est le nombre d'itération, \dot{m}_{max} et \dot{m}_{min} sont respectivement les valeurs maximales et minimales de la vitesse et ϵ est la précision de la convergence.
3. Pour la première simulation, le script vérifie si la vitesse de convergence est entre les bornes attendues en vérifiant que la vitesse maximale fait éteindre la flamme. Si la flamme n'est pas éteinte, la nouvelle vitesse maximale devient $(1.5 * \dot{m})$.
4. Une vitesse milieu est définie entre la valeur minimale et maximale de la vitesse et une simulation avec cette vitesse est lancée. Si la flamme s'éteint, alors la vitesse milieu devient la nouvelle vitesse maximale. Si la flamme ne s'éteint pas, alors la vitesse milieu la nouvelle vitesse minimale.
5. L'étape 4 est répétée N fois pour atteindre une précision donnée par l'utilisateur de la vitesse de convergence.

1.5 Lecture des résultats

Les résultats sont enregistrés dans 2 formats : un excel et un hdf5. Afin de comprendre les résultats, l'utilisateur doit opter pour les démarches suivantes.

1.5.1 Lecture du fichier excel

Pour chaque simulation, le nom du fichier excel sera nommé **VALUES_X.csv** où X est le nom de chaque simulation. La figure ci-dessous représente un exemple de résultats dans un fichier excel.

	A	B	C	D	E	F	G	H	I	J	K	L
1	T	maxHeat	strainRate	p	tburner	tsurf	mdot	width	Resolution	eq		
2	700	0.01253	291.5124		101325	300	700	8.423124	0.03	1.00E-08	0.6	
3	1627.714	1.27E+09	147.2138		101325	300	700	4.253678	0.03	1.00E-08	0.6	
4	700	8.69E-06	219.3631		101325	300	700	6.338401	0.03	1.00E-08	0.6	
5	700	9.88E-06	183.2884		101325	300	700	5.296039	0.03	1.00E-08	0.6	
6	1592.207	1.2E+09	165.2511		101325	300	700	4.774858	0.03	1.00E-08	0.6	
7	1557.405	1.1E+09	174.2698		101325	300	700	5.035449	0.03	1.00E-08	0.6	
8	700	0.023037	178.7791		101325	300	700	5.165744	0.03	1.00E-08	0.6	
9	700	0.144471	176.5244		101325	300	700	5.100596	0.03	1.00E-08	0.6	
10	700	0.052	175.3971		101325	300	700	5.068023	0.03	1.00E-08	0.6	
11	700	0.000176	174.8334		101325	300	700	5.051736	0.03	1.00E-08	0.6	
12												
13												
14												
15												

FIGURE 3 – Exemple de résultats dans un fichier excel [4].

Chaque itération est représentée sous le format d'une ligne dans le fichier. La variable **T** représente la température de l'itération en [K], **maxHeat** représente le taux de dégagement de la chaleur en [W/m^3], **strainRate** représente le taux de déformation en [1/s], **p** représente la pression en [pa], **tburner** et **tsurf** représentent respectivement la température au brûleur et à la paroi en [K], **mdot** représente la vitesse de la flamme en [$kg/(m^2 * s)$], **width** représente la distance entre le brûleur et la paroi en [m], **Resolution** représente la résolution choisi par l'utilisateur et **eq** représente le ratio d'équivalence.

Ainsi, la dernière itération qui a convergée est la dernière itération où la valeur de la température **T** n'est pas égale à **tsurf**. Dans l'exemple ci-dessus, la dernière itération convergeante est celle à la ligne 7 où **T**=1557.405 K et **strainRate**=174.2698 1/s.

1.5.2 Lecture du fichier hdf5

Pour chaque simulation, le nom du fichier excel sera nommé **X.h5** où X est le nom de chaque simulation. La figure ci-dessous représente un exemple de résultats dans un fichier hdf5 ouvert avec **HDFView**.

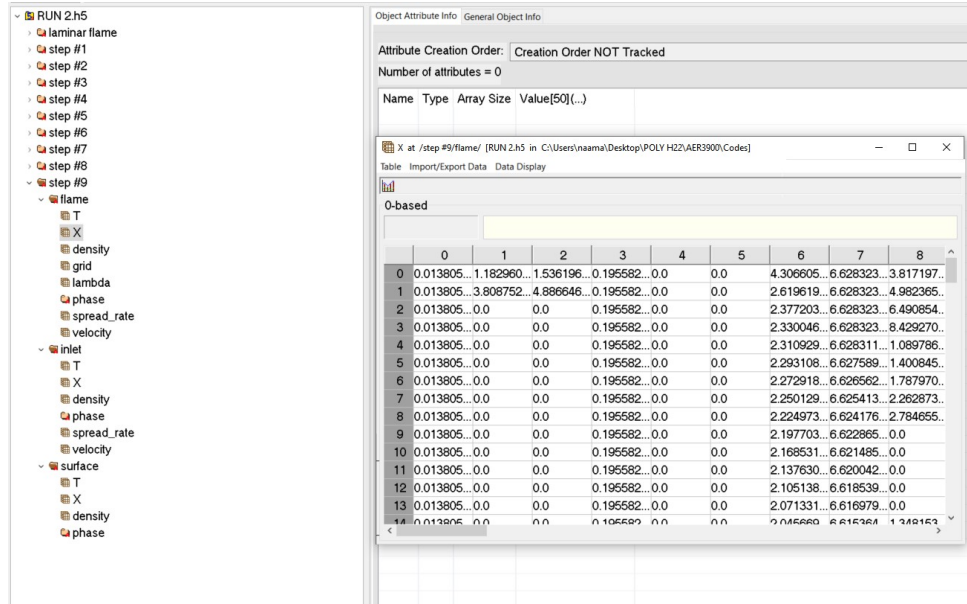


FIGURE 4 – Exemple de résultats dans un fichier hdf5 [4].

Chaque itération est représentée par une section dans le fichier en ordre chronologique ou l'étape n est "step #n" et les résultats de la flamme laminaire est représenté par "laminar flame". Pour chaque section, un objet de la flamme "flame", du brûleur "inlet" et de la surface "surface" sont sauves. Toutes les sous-sections sont définies par les fonctions *Cantera* et peuvent être ajustées selon la documentation [2].

2 Validation et analyse des scripts

Pour valider les scripts, une comparaison avec les scripts et un rapport du clients a été réalisée pour voir la différence entre les résultats.

En effet, la figure ci-dessous représente une comparaison entre les résultats du document **DNS of hydrogen and methane laminar flames** et les résultats obtenus avec les nouveaux scripts [5].

Conditions de simulation	Taux de déformation au point d'extinction des anciens scripts [1/s]	Taux de déformation au point d'extinction des nouveaux scripts [1/s]	Différence entre les valeurs [%]
<code>type = impingingjet, tburner = 300.0 K, tsurf = 500.0 K, width = 0.03 m, pressure = 101325 pa, equivalence ratio = 0.6, formula = gri30.xml, mole_frac: 0.5 et 0.5 , composition: H2 et CH4</code>	160.92	162.39	0.9%
<code>type = impingingjet, tburner = 300.0 K, tsurf = 500.0 K, width = 0.03 m, pressure = 101325 pa, equivalence ratio = 0.6, formula = gri30.xml, mole_frac: 0.2 et 0.8 , composition: H2 et CH4</code>	59.56	61.96	3.9%
<code>type = impingingjet, tburner = 300.0 K, tsurf = 500.0 K, width = 0.03 m, pressure = 101325 pa, equivalence ratio = 0.6, formula = gri30.xml, mole_frac: 0.8 et 0.2 , composition: H2 et CH4</code>	616.64	637.13	3.2%
<code>type = impingingjet, tburner = 300.0 K, tsurf = 300.0 K, width = 0.03 m, pressure = 101325 pa, equivalence ratio = 0.6, formula = gri30.xml, mole_frac: 0.5 et 0.5 , composition: H2 et CH4</code>	152.28	153.98	1.1%
<code>type = impingingjet, tburner = 300.0 K, tsurf = 700.0 K, width = 0.03 m, pressure = 101325 pa, equivalence ratio = 0.6, formula = gri30.xml, mole_frac: 0.5 et 0.5 , composition: H2 et CH4</code>	171.31	174.83	2.0%

FIGURE 5 – Tableau de comparaison entre les résultats du document **DNS of hydrogen and methane laminar flames** et les résultats obtenus avec les nouveaux scripts [4]

Or, les termes en **rouge** représentent les conditions qui ont été modifiées pour chaque simulation selon la première. Il est possible de voir que, pour toutes les différentes conditions de simulation, les valeurs obtenus pour les seconds scripts sont plus élevés d'un petit pourcentage. Ceci est normal, car les anciennes simulations ont été effectuée avec des itérations sur la vitesse choisies par l'utilisateur, tandis que les nouveaux scripts utilisent la méthode de la bisection pour effectuer les itérations automatiquement.

Ainsi, les taux de déformation au point d'extinction des nouveaux scripts sont plus élevés d'un petit pourcentage, car les simulations sont plus précises.

3 Informations additionnelles

3.1 Bibliothèques utilisées

Voici la liste des bibliothèques à installer pour faire fonctionner les scripts.

1. **json** : Utilisé pour préparer les simulations
2. **sys** : Utilisé pour arrêter une simulation si le json n'est pas adéquat
3. **time** : Utilisé pour connaître le temps des simulation
4. **concurrent.futures** : Utilisé pour le parallélisme
5. **tdqm** : Utilisé pour suivre les étapes du parallélisme
6. **psutil** : Utilisé pour connaître le nombre de coeur dans l'ordinateur pour le parallélisme

3.2 Lecture des graphiques

Pour chaque simulation, un graphique sera enregistré sous le nom *X.png* où X est le nom de la simulation. Soit la figure ci-dessous représente un exemple de graphique enregistré :

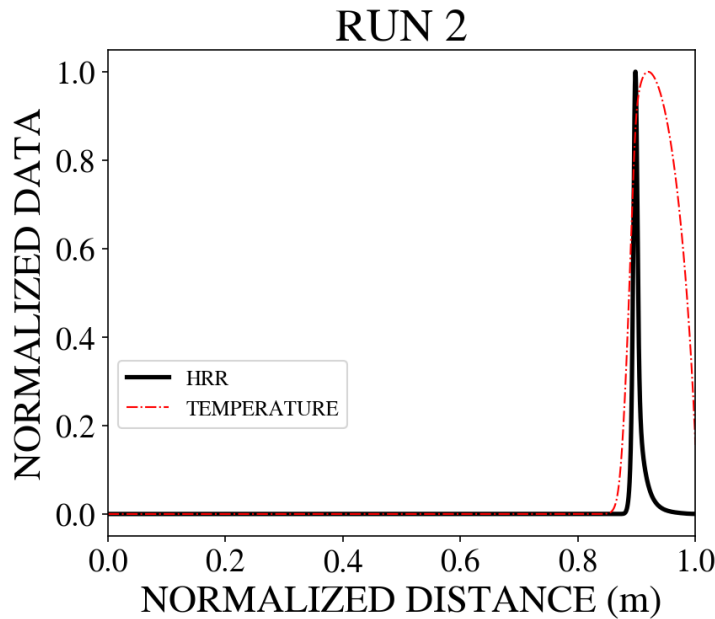


FIGURE 6 – Exemple de graphique enregistré pour une simulation donnée[4]

Il est possible de voir que la température et le taux de dégagement de la chaleur (HRR) sont près de la distance maximal normalisée de 1.0, car les réactions de la flamme sont proche de la paroi et près de l'extinction.

Références

- [1] AMSI. *Speed of the bisection algorithm*. URL : https://amsi.org.au/ESA_Senior_Years/SeniorTopic3/3j/3j_3links_1.html. (accessed : 04.10.2022).
- [2] CANTERA. *Objects Representing Phases*. URL : <https://cantera.org/documentation/dev/sphinx/html/cython/importing.html>. (accessed : 04.10.2022).
- [3] CERFACS. *ADIABATIC_FLAME—Afreely—propagating, premixed flat flame*. URL : https://www.cerfacs.fr/cantera/docs/mechanisms/hydrogen-oxygen/Boivin/flame/flamePropagation_RM.py. (accessed : 04.10.2022).
- [4] Rani NAAMAN. 4 nov. 2022.
- [5] Rani NAAMAN. “DNS of hydrogen and methane laminar flames”. In : (2021).
- [6] Mathematical PYTHON. *Bisection Method*. URL : <https://personal.math.ubc.ca/~pwalls/math-python/roots-optimization/bisection/>. (accessed : 04.12.2022).