

Resolución de la ecuación Kuramoto–Sivashinsky utilizando Spectral Splitting Methods.

Métodos Numéricos Avanzados - Trabajo Práctico 2

Grupo 8

Antonielli Julián
Capparelli Martín
Olivera Ramiro
Paganini Nicolás
Tay Lucía

Resumen

En el presente informe, buscaremos resolver la ecuación Kuramoto–Sivashinsky (KS) a través del uso de *Spectral Splitting Methods* (SSM). En particular, profundizaremos el análisis sobre la familia de métodos embebidos conocidos como *Métodos Afines*, tal como se describen en [1], y los compararemos con SSMs más conocidos como los propuestos por Lie-Trotter [2], Strang [3], Ruth [4] y Neri [5].

Introducción

La ecuación Kuramoto–Sivashinsky (KS) data de mediados de los años 70s, y se utiliza en el estudio de una variedad de sistemas de reacción-difusión [6]. La ecuación en una dimensión espacial en su forma derivada, puede expresarse de siguiente forma:

$$u_t = -uu_x - u_{xx} - u_{xxxx}, \quad x \in [0, 32\pi]. \quad (1)$$

La ecuación fue independientemente derivada en el contexto de diferentes sistemas físicos. Entre ellos, por ejemplo, es el modelado de las inestabilidades en frentes de llamas laminares [7].

Por contener tanto derivadas de segundo y cuarto orden, la ecuación produce resultados complejos. Se pueden separar sus términos en lineales, como el término de segundo orden que actúa como fuente de energía con un efecto desestabilizador, y el de cuarto orden con un efecto estabilizador; Y no-lineales, en que transfiere la energía desde valores bajos de la frecuencia espacial de las ondas, a valores más altos. [8]

Desarrollo matemático

En principio vamos a definir la familia de SSMs que utilizaremos en este trabajo, tal como se definen en [1]. Los mismos consisten en obtener los flujos asociados a los problemas

parciales de la ecuación diferencial que vamos a resolver, y aplicar composiciones de integradores de *LieTrotter* como se muestra a continuación.

$$\Phi(h) = \sum_{m=1}^s \gamma_m \Phi_m^{\pm}(h/m) \quad \text{Asimétrico (2)}$$

$$\Phi(h) = \sum_{m=1}^s \gamma_m (\Phi_m^{+}(h/m) + \Phi_m^{-}(h/m)) \quad \text{Simétrico (3)}$$

Donde se definen los integradores de la siguiente manera.

$$\Phi^{+}(h) = \phi_1(h) \circ \phi_0(h) \quad (4)$$

$$\Phi^{-}(h) = \phi_0(h) \circ \phi_1(h) \quad (5)$$

$$\Phi_m^{\pm}(h) = \Phi^{\pm}(h) \circ \Phi_{m-1}^{\pm}(h) \quad (6)$$

$$\Phi_1^{\pm} = \Phi^{\pm} \quad (7)$$

Con las constantes gammas definidas de la siguiente forma, para los métodos (2) y (3) respectivamente.

$$\begin{aligned} 1 &= \gamma_1 + \gamma_2 + \cdots + \gamma_s, \\ 0 &= \gamma_1 + 2^{-k} \gamma_2 + \cdots + s^{-k} \gamma_s, \quad 1 \leq k \leq q-1, \end{aligned} \quad \text{Asimétrico (8)}$$

$$\begin{aligned} \frac{1}{2} &= \gamma_1 + \gamma_2 + \cdots + \gamma_s, \\ 0 &= \gamma_1 + 2^{-2k} \gamma_2 + \cdots + s^{-2k} \gamma_s, \quad 1 \leq k \leq n-1 \end{aligned} \quad \text{Simétrico (9)}$$

Con $n = q/2$ en el caso de los integradores simétricos, donde el orden q solo puede ser par.

Definidos los integradores que utilizaremos, comenzamos con la resolución de la ecuación (1) definiendo la condición inicial. Con este fin, y para el análisis que se realizará en este trabajo, utilizaremos la misma utilizada en [8].

$$u(x, 0) = \cos(x/16)(1 + \sin(x/16)). \quad (10)$$

Para resolver la ecuación vamos a trabajar en espacio de Fourier. Para ello transformamos la ecuación (1) aplicando la transformada de Fourier.

$$\widehat{u}_t = -\frac{ik}{2}\widehat{u}^2 + (k^2 - k^4)\widehat{u} \quad (11)$$

Para comenzar a definir los flujos necesarios para el cálculo de los integradores definidos en (2) y en (3), debemos separar la ecuación en su parte lineal y su parte no-lineal.

$$\partial_t \widehat{u} = (k^2 - k^4)\widehat{u} \quad \text{Lineal} \quad (12)$$

$$\partial_t \widehat{u} = \frac{-ik}{2}\widehat{u}^2 \quad \text{No-Lineal} \quad (13)$$

Para obtener los flujos resolvemos ambas ecuaciones. La ecuación (12) puede resolverse de forma analítica, obteniéndose la solución (14) donde $h = t - t_0$.

$$\widehat{u} = e^{(k^2 - k^4)h} \quad (14)$$

En cambio, la ecuación (13) no puede resolverse de forma analítica, por lo que se resuelve de forma aproximada utilizando Fast Fourier Transform (FFT) y Runge-Kutta de orden 4 (RK4) [9], aplicando las siguientes operaciones para obtener la solución aproximada (20). Nuevamente vale $h = t - t_0$.

$$g = \frac{-ikh}{2} \quad (15)$$

$$a = g * F\{F^{-1}\{\widehat{u}\}^2\} \quad (16)$$

$$b = g * F\{F^{-1}\{\widehat{u} + \frac{a}{2}\}^2\} \quad (17)$$

$$c = g * F\{F^{-1}\{\widehat{u} + \frac{b}{2}\}^2\} \quad (18)$$

$$d = g * F\{F^{-1}\{\widehat{u} + c\}^2\} \quad (19)$$

(20)

Habiendo obtenido las soluciones a los subproblemas, solo queda aplicarlas como lo definen los métodos definidos previamente en (2) y (3).

Desarrollo técnico

El desarrollo del presente trabajo se realizó en su totalidad en MATLAB [10] de manera modular, definiendo las estructuras básicas y utilizando módulos de más alto nivel para integrarlas. Además de los SSMs, se desarrollaron nuestras propias implementaciones de FFT y su inversa para su aplicación en el trabajo.

La efectividad de los métodos propuestos por [1] no solo está en que pueden alcanzar órdenes arbitrarios, como los métodos conocidos como simpléticos [4]. Pero a diferencia de éstos últimos, son altamente paralelizables. Por lo que aunque sea necesaria la ejecución de más pasos, la paralelización permite que se obtengan resultados a mayor velocidad (Siempre acotados por la ley de Amdahl [11]).

En el presente trabajo se implementaron los SSMs utilizando tanto programación secuencial como computación paralela, a través del uso de MPI en MATLAB.

Análisis

Para entender la efectividad de los métodos propuestos por [1] es necesario definir la forma en la que la mediremos. Para ello, se define la función *Error* de la siguiente manera.

$$Error(y_1, y_2) = \|y_1 - y_2\|_{\infty} \quad (21)$$

Donde y_1 y y_2 son funciones, o en nuestro caso las muestras de las funciones. En el presente trabajo se evalúan los integradores de dos maneras distintas: Utilizando el *error global* y el *error local*.

$$Error(f(h), f(\frac{h}{2})) \quad \text{Error Global (22)}$$

$$Error(f_n(h), f_{n+1}(h)) \quad \text{Error Local (23)}$$

Donde h es el paso, y n el orden del integrador. Definidas ya las funciones para calcular los errores de los distintos métodos, comenzaremos comparando los errores globales de los SSMs comunes de distintos órdenes (Figura 1). Cabe destacar que para todas las pruebas, utilizamos 256 muestras de la ecuación a analizar.

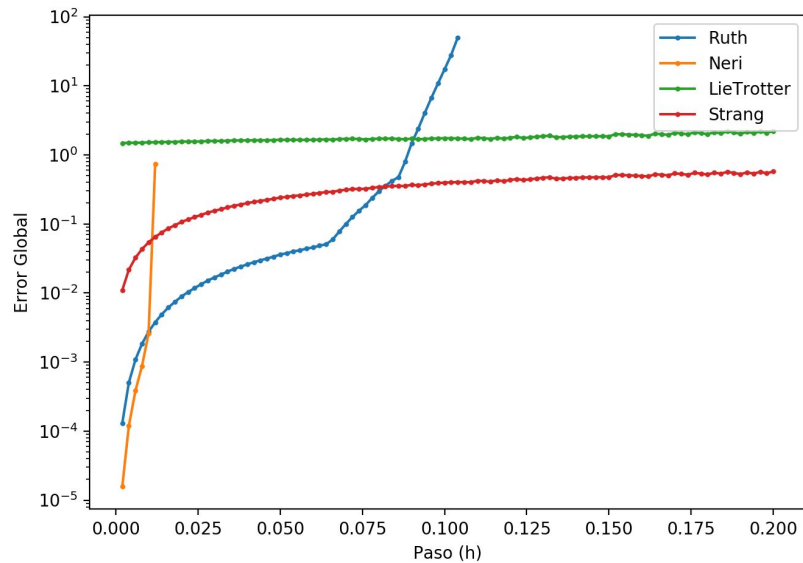


Figura 1 - Comparación de los Errores globales de SSMs comunes con $T = 10$, evaluados con pasos desde 0.002 hasta 0.2, a intervalos de 0.002.

Cómo puede observarse, a mayor orden del integrador, menor el error global para el mismo paso. No obstante, se puede detectar un problema con los integradores *simpléticos*: Si el paso no es suficientemente pequeño, el error se dispara, implicando que el integrador diverge y no puede ser utilizado a esa escala. El ejemplo de Neri es el que más resalta, con un error menor a 10^{-5} para pasos suficientemente pequeños, pero a partir de $h = 0.01$ diverge y ya no puede utilizarse.

Habiendo analizado los SSMs comunes, es de interés analizar la familia de integradores definidos por los métodos afines (Figura 2).

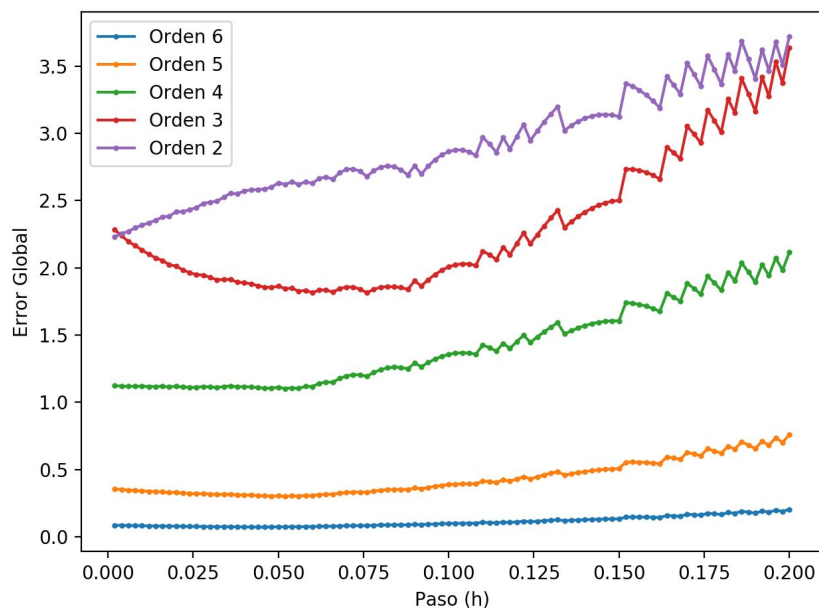


Figura 2 - Comparación de los Errores globales de métodos afines asimétricos de distintos órdenes, con $T = 10$, evaluados con pasos desde 0.002 hasta 0.2, a intervalos de 0.002.

Al igual que con la Figura 1, se observa una reducción del error global a medida que se utilizan integradores de mayor orden. Puede observarse también que el error de los métodos de mayor orden, tienen errores superiores a sus pares *simpléticos*. Pero a diferencia de ellos, no tienen la limitación del paso, siendo superiores a los mismos para pasos de mayor magnitud. Este fenómeno es más notorio en la Figura 3, donde se compara un método afín con su par simplético.

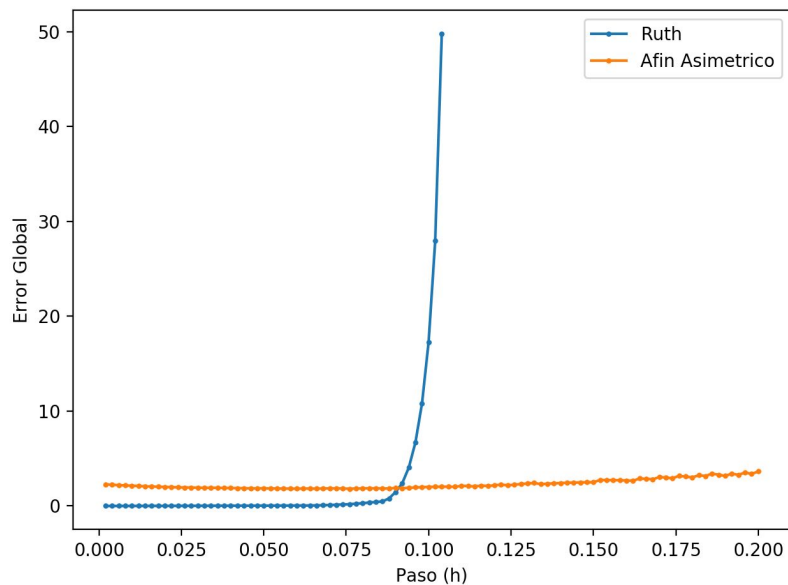


Figura 3 - Comparación de los Errores globales del método afín asimétrico de orden 3 con el integrador simplético de Ruth, con $T = 10$, evaluados con pasos desde 0.002 hasta 0.2, a intervalos de 0.002.

Para comparar la diferencia entre los órdenes de los métodos afines, es mejor idea utilizar la definición de Error local (23). Como se observa en la Figura 4, y como venimos observando, el error decrece de manera significativa a medida que el orden del integrador aumenta.

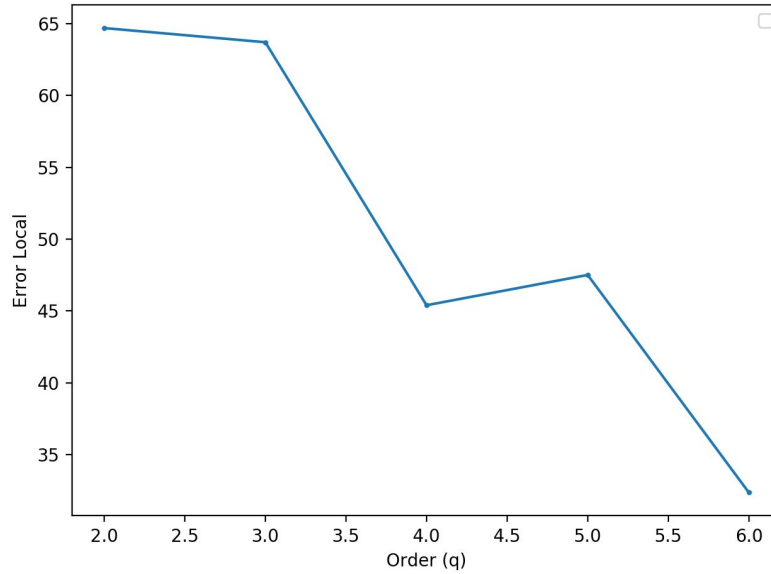


Figura 4 - Errores locales de métodos afines asimétricos de acuerdo a sus órdenes, con $T = 150$, evaluados con paso $h = 0.001$.

Habiendo estudiado la efectividad de la familia de integradores definidos por los métodos afines, queda pendiente el estudio de su eficiencia. Como se mencionó previamente, una de las ventajas de estos métodos es que si bien, la cantidad de operaciones con respecto a sus pares *simplécticos* es mayor, al no ser todas aplicaciones de funciones en secuencia su cómputo pueden paralelizarse.

Para estudiar la eficiencia de la paralelización, vamos a utilizar un concepto conocido como *Speedup*. El *Speedup* es la mejora en la velocidad de ejecución de una tarea ejecutada en dos arquitecturas similares con diferentes recursos. Utilizando la formulación de la Ley de Amdahl [11] podemos definir el *Speedup* como

$$A = \frac{T_{serie}}{T_{paralelo}} \quad (24)$$

Donde A es el incremento de la velocidad dado por el cambio en los recursos. En otras palabras, qué tanto más rápido es el sistema por ejecutarlo en paralelo, en lugar de ejecutarlo en serie.

Es de interés entonces, evaluar cómo evoluciona el speedup a medida que se aumenta el orden de los integradores (Figura 5).

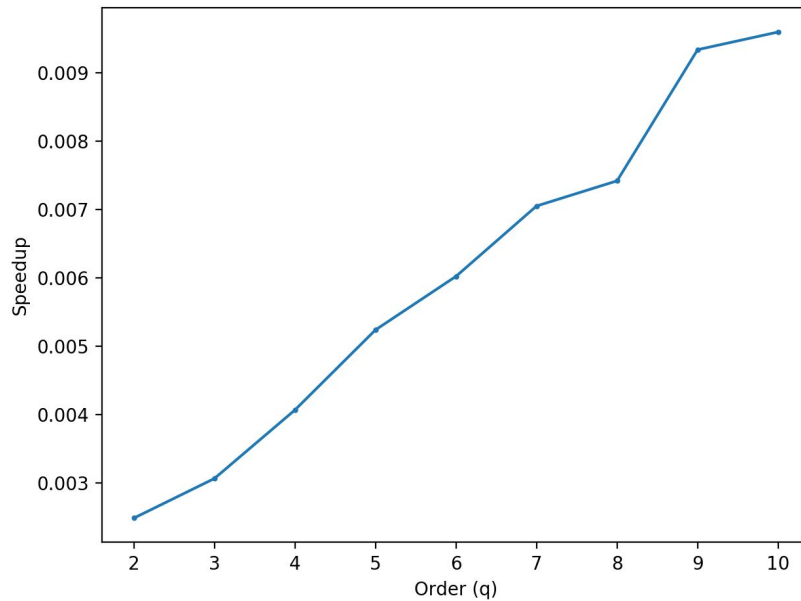


Figura 5 - Evolución del Speedup en función del orden de los métodos afines. Ejecutado con $T = 20$, $h = 0.2$, y promediado 100 corridas para los métodos en serie, y 10 para los métodos en paralelo.

Lo primero que llama la atención es que en lugar de una mejora al correr el cómputo de manera paralela, obtenemos un decrecimiento de la velocidad. Esto puede deberse a distintos motivos. En particular, los autores consideramos que se debe a que Matlab realiza optimizaciones sobre el código en serie, paralelizando aún sin ser pedido. Al forzar la paralelización de una forma distinta, removemos estas optimizaciones, haciendo parecer que la paralelización hace la ejecución más lenta [12].

Por otro lado, puede observarse que a mayor orden del método, mayor es el valor del *Speedup*. Ésto es, a mayor orden de integrador, más *conviene* paralelizarlo. Sin embargo, como prevé la ley de Amdahl, esta mejora no es eterna, sino que hacia los órdenes más altos la mejora tiende a estancarse.

Dejando de lado el análisis sobre los integradores, nos pasamos a enfocar en la solución a la ecuación KS. Dado que la ecuación puede interpretarse como llamas, y utilizando el estilo de visualización definido en [8], encontramos la solución como puede observarse en la Figura 6.

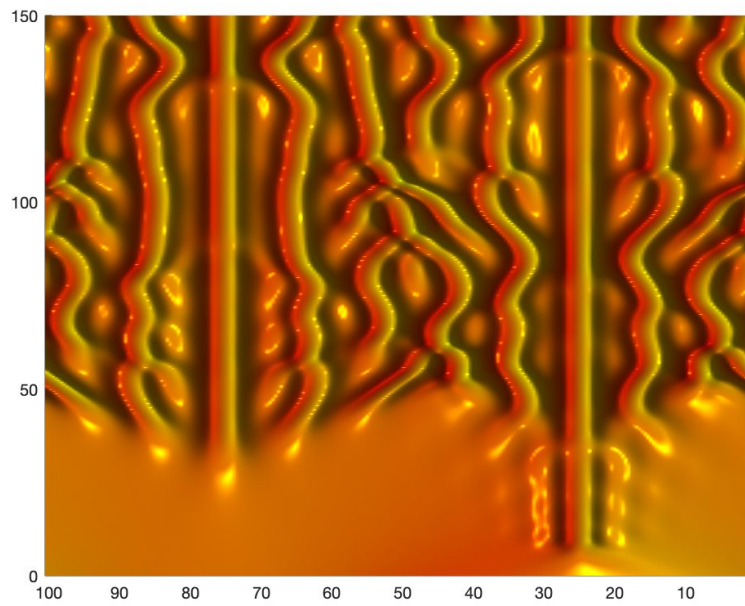


Figura 6 - Solución a la ecuación KS utilizando el método afín asimétrico de orden 4 con $T = 150$ y $h = 0.002$.

Un análisis interesante para realizar es el de las perturbaciones espaciales. Dada la interpretación a la ecuación KS definida antes, podemos simular una *brisa* de aire sobre la llama, introduciendo un ligero ruido a la condición inicial. De esta manera, como se observa en la Figura 7, obtenemos soluciones ligeramente distintas para los tiempos más altos.

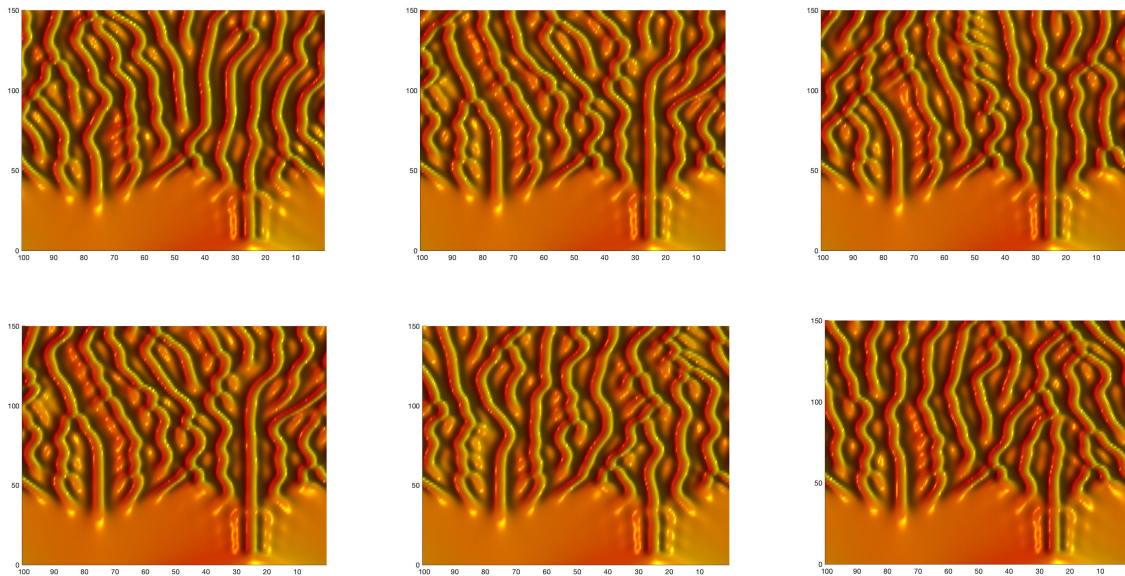


Figura 7 - Soluciones de la ecuación KS utilizando un paso de 0.02 en el método afín asimétrico de orden 4 y distintas perturbaciones espaciales aleatorias del 1% sobre los valores de x de la condición inicial.

Conclusiones

Por lo analizado en este trabajo, concluimos que la familia de integradores definida en [1] son efectivos desde varios puntos de vista. Por un lado, permiten la resolución de las ecuaciones diferenciales parciales en órdenes altos con pasos no tan chicos como los que requieren los integradores *simplécticos*, con un error aceptable.

Por el otro, para problemas que requieran mucho poder de cómputo, la posibilidad de paralelizar los cálculos podrían acelerar en gran medida la resolución, pero si los problemas se pueden computar rápidamente, el *overhead* asociado con la creación de threads y nuevos procesos, y el intercambio de mensajes entre estos, pueden hacer que la solución paralela tome más tiempo que la versión secuencial.

Referencias

- [1] Alvarez A. Rial D. Affine combination of splitting type integrators implemented with parallel computing methods. International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering, 9(2):146–149, 2015.
- [2] Trotter H.F. On the product of semigroups of operators. 10:545–551, 1959. Proc. Amer. Math. Soc.
- [3] Strang G. Accurate partial difference methods ii: Non linear problems. Numerische Math., 6:37–46, 1963.
- [4] Ruth R. D. A canonical integration technique. IEEE Transactions on Nuclear Science, NS 30(4):2669–2671, 1983.
- [5] Neri F. Lie algebras and canonical integration. Department of Physics report, University of Maryland, pages 1489–1500, 1987.
- [6] Y. Kuramoto and T. Tsuzuki, Persistent propagation of concentration waves in dissipative media far from thermal equilibrium, Prog. Theoret. Phys., 55 (1976), pp. 356–369.
- [7] G. Sivashinsky, "Nonlinear analysis of hydrodynamic instability in laminar flames I. Derivation of basic equations" Acta Astron. , 4 (1977) pp. 1177–1206
- [8] Trefethen L.N. Kassam A.K. Fourth order time stepping for stiff PDE's., volume 26 No 4 pages: 1214-1233. SIAM J. SCI Comput., 2005.
- [9]"Método de Runge-Kutta", Es.wikipedia.org, 2019. [Online]. Available: https://es.wikipedia.org/wiki/M%C3%A9todo_de_Runge-Kutta#M%C3%A9todos_de_Runge-Kutta_de_cuarto_orden. [Accessed: 20- Nov- 2019].
- [10]"MATLAB Online R2019b", Matlab.mathworks.com, 2019. [Online]. Available: <https://matlab.mathworks.com/>. [Accessed: 20- Nov- 2019].

[11] Gene Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities", AFIPS Conference Proceedings, (30), pp. 483-485, 1967.

[12]"parfor is slower than for - MATLAB Answers - MATLAB Central", La.mathworks.com, 2019. [Online]. Available:
<https://la.mathworks.com/matlabcentral/answers/389191-parfor-is-slower-than-for>.
[Accessed: 20- Nov- 2019].