

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Estudi tèrmic i fluidodinàmic del regenerador d'un motor Stirling

MEMÒRIA

Autor: Miquel Solé Bosch
Director: Elisabet Mas de Les Valls Ortiz
Convocatòria: Juny 2018



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

El present projecte s'entén com el primer d'una sèrie d'estudis destinats a la caracterització i optimització d'un motor Stirling. Aquest motor de combustió externa té un gran potencial d'aprofitament eficient de l'energia. De fet, el cicle Stirling ideal presenta el mateix rendiment que una màquina de Carnot. Per aconseguir-ho, però, cal afegir i optimitzar un regenerador al motor. El regenerador és un bescanviador de calor on s'aprofita la inèrcia tèrmica del propi fluid de treball.

Per a optimitzar un motor Stirling cal una simulació detallada del regenerador, típicament mitjançant eines de fluidodinàmica computacional (CFD). Per a fer-ho es requereix tenir ben caracteritzat el cicle complet i poder definir adequadament les condicions de treball i de contorn de la simulació CFD. Per tant, és imprescindible tenir un model simplificat del cicle complet. El present projecte té com a objectiu principal la definició, implementació i verificació d'aquest modelatge del cicle complet.

En aquest document, després d'una introducció definint la motivació, objectius i abast del projecte, s'inclou la descripció del cicle Stirling i les seves diferents configuracions. Seguidament, es comenten i justifiquen les hipòtesis emprades per la modelització matemàtica del cicle complet. Després de la descripció de la obtenció de les equacions de conservació es presenta l'algoritme implementat emprant l'eina MATLAB. Es dedica un capítol a mostrar el procés de verificació de l'eina desenvolupada, incloent-hi les diferents estratègies provades i les fortaleeses i les debilitats del codi. Finalment, s'inclou una primera proposta de l'estratègia a seguir en les següents fases del projecte global de caracterització i optimització del motor Stirling.

L'eina resultant del present estudi es facilitarà als docents de l'assignatura d'Ampliació de Màquines Tèrmiques (assignatura optativa del Màster Universitari en Enginyeria de l'Automoció i del Màster Universitari en Enginyeria Industrial) per ser emprada com a material docent.

Sumari

RESUM	1
SUMARI	3
1. PREFACI	5
1.1. Motivació i Origen del projecte	5
1.2. Requeriments previs	6
2. INTRODUCCIÓ	7
2.1. Objectius.....	7
2.2. Abast del projecte.....	7
3. EL MOTOR STIRLING	9
3.1. El cicle d'Stirling	10
3.2. Configuracions del motor Stirling.....	13
3.2.1. Configuració alpha	14
3.2.2. Configuració beta	15
3.2.3. Configuració gamma.....	16
3.3. Regenerador	17
3.4. Usos	18
4. CODI DE SIMULACIÓ	20
4.1. Procés d'elaboració del codi.....	21
4.1.1. Fase I: Pressió i temperatura homogènies.....	21
4.1.2. Fase II: Pressió homogènia i perfil de temperatura 1D al regenerador	22
4.1.3. Fase III: cicle real	23
4.1.4. Fase IV: cicle real amb balanç de potència total i conservació de la massa ...	23
4.2. Hipòtesis i simplificacions.....	24
4.3. Equacions.....	25
4.3.1. Geometria i volum	25
4.3.2. Balanços d'energia.....	27
4.3.3. Conservació de la massa i càlcul dels cabals	31
4.4. Algoritme	31
5. VERIFICACIÓ	36
5.1. Definició de les condicions de treball	36
5.2. Procés de verificació	37
5.3. Procés de millora.....	43

6. ACCIONS FUTURES	47
6.1. Estabilització del sistema	47
6.2. Millores al regenerador	47
6.3. Estudi CFD.....	48
7. PRESSUPOST	49
CONCLUSIONS	50
AGRAÏMENTS	51
BIBLIOGRAFIA	52
ANNEXOS	54
Annex A. Codi de simulació principal	54
Annex B. Codi amb el segon mecanisme d'acoblament.....	66

1. Prefaci

1.1. Motivació i Origen del projecte

En un inici, la motivació personal i a la vegada l'objectiu principal d'aquest projecte era posar en pràctica de forma directa els coneixements adquirits en l'assignatura de Termodinàmica (del Grau en Enginyeria en Tecnologies Industrials) referents a la transferència de calor. Una part interessant de l'assignatura, és la modelització de la transferència de calor a través de la discretització de sòlids, per a posteriorment poder-los estudiar amb mètodes numèrics. Així doncs, la idea inicial era treballar en aquesta línia.

A través del departament de Màquines i Motors Tèrmics de l'ETSEIB, va sorgir la idea d'estudiar el regenerador d'un motor Stirling, per a veure com aquest podia afectar al rendiment del cicle. La motivació per a dur a terme l'estudi és doble. Per una banda l'estudi i optimització del regenerador d'un motor Stirling té un gran interès tecnològic degut a la seva potencial aplicació en el camp de l'energia solar. Per l'altra, el desenvolupament d'una eina simplificada que caracteritzi el cicle Stirling és de gran utilitat com a material docent dins l'assignatura d'Ampliació de Màquines Tèrmiques (assignatura optativa del Màster Universitari en Enginyeria de l'Automoció i del Màster Universitari en Enginyeria Industrial).

L'estudi del regenerador implica un estudi numèric detallat mitjançant eines de fluidodinàmica computacional (CFD). Per a poder realitzar aquest estudi però, es requereix un coneixement profund del cicle i una caracterització detallada de les condicions de treball del regenerador. Ambdós aspectes només es poden aconseguir mitjançant un estudi global del cicle amb eines més senzilles i més conceptuals.

Donada la complexitat que implica el desenvolupament d'una eina simplificada per caracteritzar el cicle Stirling i coneixent els problemes de capacitat de càlcul que poden sorgir de l'estudi CFD, l'estudi complet del regenerador del motor Stirling s'ha definit com una sèrie d'estudis seqüencials. Així, el present treball mostra el avenços del primer d'aquesta sèrie d'estudis.

1.2. Requeriments previs

Per a la realització d'aquest projecte han estat imprescindibles els coneixements adquirits a les assignatures Termodinàmica Fonamental, Termodinàmica i Termotècnia del Grau en Enginyeria en Tecnologies Industrials de la UPC.

Les dues primeres han permès aprendre sobre la caracterització dels sistemes termodinàmics a partir de les seves variables d'estat, així com el funcionament dels cicles i els canvis que hi experimenta un sistema. L'altra, ha permès conèixer com es modelitza la transferència de calor en els diferents mecanismes en els quals es pot donar i quins són els balanços adequats per a cada sistema en cada situació.

Aquests coneixements han permès tenir una idea més completa de com enfocar un estudi sobre els sistemes que es troben en el nostre entorn.

A més, cal destacar també la importància dels coneixements obtinguts a les assignatures d'Informàtica sobre com funciona el món de la programació i la introducció a MATLAB rebuda, sense la qual utilitzar aquest software hagués estat més complicat.

2. Introducció

2.1. Objectius

L'objectiu del present estudi és el disseny, implementació i verificació d'una eina simplificada per la simulació numèrica del cicle Stirling complet. A aquests efectes hi ha dos condicionants:

- 1) que el sistema d'equacions ha de ser prou simple com per modelar el cicle de forma ràpida i permetre així estudis paramètrics
- 2) que s'ha de poder estudiar la zona del regenerador amb detall ja que és on tenen lloc els fenòmens claus per l'eficiència del cicle

Tradicionalment, els models simplificats de cicles termodinàmics es realitzen mitjançant balanços de potència 0D, és a dir, considerant una única pressió i temperatura en cada instant del cicle. Malauradament aquesta aproximació no és aplicable a la configuració del motor Stirling estudiada ja que es perd informació dels fenòmens que tenen lloc al regenerador que són, justament, els que donen eficiència al cicle. Per tant, com a objectiu secundari i gran repte del present treball, es planteja la construcció d'un algoritme que combini un anàlisi 0D de pressió del sistema amb una anàlisi 1D d'entalpia –i temperatura- en el regenerador.

L'eina resultant es facilitarà als docents de l'assignatura d'Ampliació de Màquines Tèrmiques (assignatura optativa del Màster Universitari en Enginyeria de l'Automoció i del Màster Universitari en Enginyeria Industrial) per ser emprada com a material docent.

2.2. Abast del projecte

El present projecte és el primer d'una sèrie d'estudis per caracteritzar el motor Stirling i, especialment, el funcionament del regenerador. En aquesta primera fase, l'estudi es centra en un modelat del cicle que combina una anàlisi 0D amb una d'1D. Donada la gran complexitat d'aquest modelat, en el present projecte no s'inclou cap estudi de parametrització ni optimització del cicle. Tampoc s'inclou l'estudi CFD, considerat com una acció futura a aquest estudi.

El modelat del cicle Stirling implicarà el desenvolupament de les equacions de conservació adients, la seva seqüenciació i acoblament en forma d'algoritme i la seva implementació

emprant l'eina MATLAB.

Per tal de connectar el present estudi amb els posteriors, es dedica un capítol a proposar una estratègia per la realització de les següents fases.

3. El motor Stirling

El motor Stirling va ser inventat per l'escocès Robert Stirling (Figura 3.1) l'any 1816. El seu objectiu era proporcionar un motor que pogués servir d'alternativa a la màquina de vapor, molt estesa en aquella època.

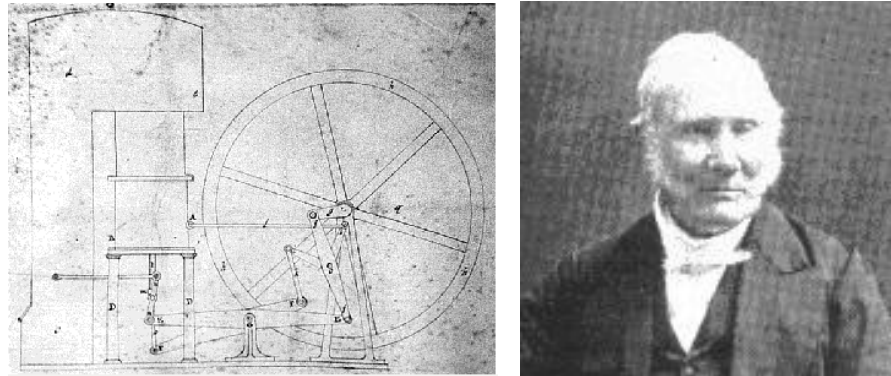


Figura 3.1: a l'esquerra una imatge de la patent de Robert Stirling¹. A la dreta, el seu inventor²

El motor Stirling, a diferència dels majorment utilitzats Otto o Diesel al llarg de la història, és un motor de combustió externa. El seu funcionament es basa en l'expansió i la contracció que experimenta un gas en fer variar la seva temperatura. Per a aconseguir aquesta variació de temperatura, s'obliga el gas a desplaçar-se d'un focus fred, en el qual es contrau, a un focus calent on s'expandeix.

A la següent figura (3.2) es pot veure un esquema d'un motor Stirling, on el focus fred correspondria a la part on el gas està representat de color blau i el calent on es veu de color vermell.

¹ Font: <https://www.nms.ac.uk/stirlingengine>

² Font: <http://www.robertstirlingengine.com>

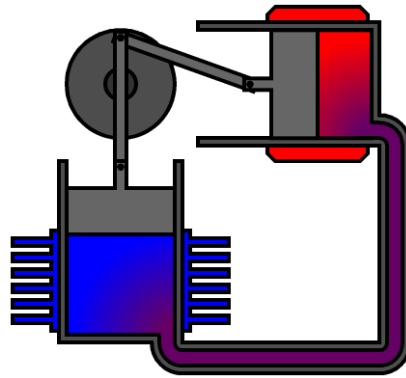


Figura 3.2: esquematització d'un motor Stirling³

El motor Stirling té dos principals avantatges sobre la majoria de motors. El primer és que, en ser de combustió externa, pot utilitzar com a combustible qualsevol font de calor. Així, no obliga a utilitzar combustibles fòssils i pot arribar a funcionar a partir d'energia solar o fins i tot nuclear.

Un altre avantatge important del motor Stirling és el seu gran rendiment. De fet, en el cicle d'Stirling ideal, el rendiment és el mateix que el d'una màquina de Carnot. Aquest rendiment de Carnot és el màxim possible que es pot obtenir amb una màquina tèrmica.

A més, té un llarg llistat d'avantatges implícits a la seva construcció i funcionament: el poc soroll que genera en funcionar, la resiliència i el baix cost de manteniment i la seva polivalència en quant als diferents usos que se li poden donar.

Com a contrapartida, també té alguns desavantatges considerables, la qual cosa fa que avui en dia no sigui un motor gaire popular. Aquests, es posen especialment de manifest de cara a la utilització d'aquest motor en automoció. Els principals són: el temps que triga en engegar-se i la seva baixa capacitat de resposta quan se li requereixen canvis de velocitat.

3.1. El cicle d'Stirling

El cicle d'Stirling és la representació teòrica de l'evolució termodinàmica del gas comprès en el motor del mateix nom. El cicle d'Stirling ideal està format per 4 processos, tots ells

³ Font: https://commons.wikimedia.org/wiki/File:Alpha_Stirling.gif

reversibles (Figura 3.3):

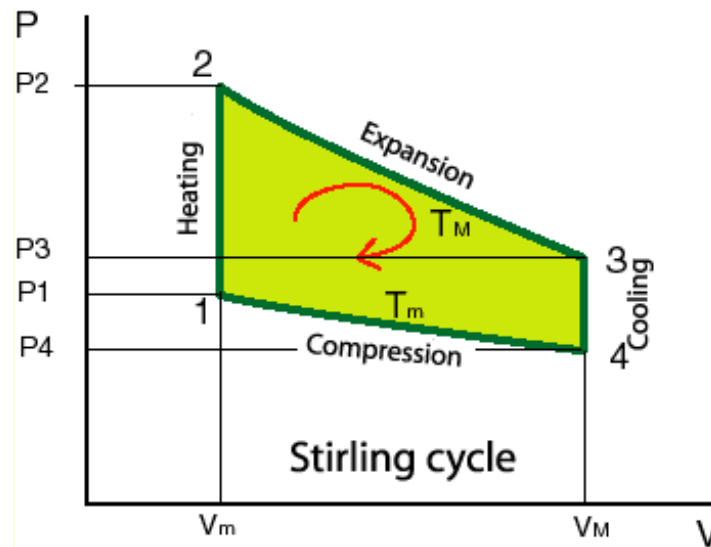


Figura 3.3: representació P-V del cicle d'Stirling [Robertstirlingengine]

- **1-2:** procés d'escalfament a volum constant, fins arribar a la temperatura màxima de treball, en el qual també augmenta la pressió
- **2-3:** procés d'expansió a temperatura constant, del volum mínim al volum màxim, disminuint la pressió
- **3-4:** refredament a volum constant, arribant a la temperatura i pressió mínimes
- **4-1:** procés de compressió a temperatura constant.

En primer lloc trobem el procés d'1 a 2, que és en el qual el gas es troba a la font calenta i rep calor d'aquesta, de forma que augmenta la seva temperatura. En el segon, el gas s'expansiona, generant així el treball que es pot obtenir d'aquest motor. Com que aquesta expansió es produeix principalment al focus calent, el gas segueix rebent calor, fent així que la seva temperatura no disminueixi (Figura 3.4).

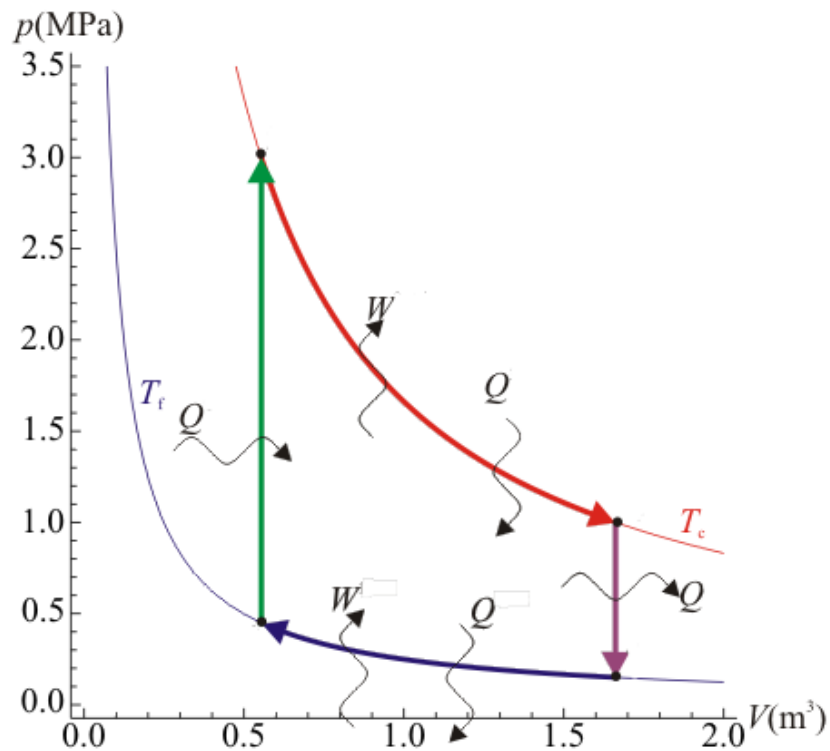


Figura 3.4: representació P-V del cicle Stirling amb la representació de les calors i el treball⁴

Els altres dos processos en canvi, tenen lloc principalment al focus fred. En el de 3 a 4, el gas allibera calor disminuint així la seva temperatura. En l'últim procés del cicle també s'allibera calor i a la vegada, es necessita una aportació de treball per a dur a terme el procés, que afecta negativament al treball net que es pot aconseguir del motor.

Una bona manera d'augmentar el rendiment del motor, és aprofitar la calor que allibera en alguns processos per ajudar en els que s'ha d'aportar calor al gas. Aquest procés, només és possible entre els processos de refredament (3-4) i d'escalfament (1-2) ja que del contrari es contradiria la segona llei de la termodinàmica.

La transferència de la calor generada en el procés de refredament al d'escalfament, rep el nom de regeneració i es pot dur a terme a través de diferents tipus d'intercanviadors de calor. Aquests intercanviadors, quan són utilitzats per a la tasca descrita, reben el nom de regeneradors.

⁴ Font: http://laplace.us.es/wiki/index.php/Caso_pr%C3%A1ctico_de_ciclo_de_Stirling

3.2. Configuracions del motor Stirling

El motor Stirling es pot construir de formes molt diverses i amb materials i objectes de tot tipus: des de llaunes de refrescs (Figura 3.5) a acers utilitzats a la indústria.



Figura 3.5: motor Stirling construït amb una llauna de refresc⁵

Aquesta llibertat de construcció, esdevinguda gràcies a la relativa facilitat de funcionament del motor Stirling, fa que aquest es pugui implementar de formes molt diverses. Tot i això, la majoria d'ells es poden classificar dins les 3 configuracions principals que pot tenir un Stirling: la alpha, la beta o la gamma.

Tot i que les diferents configuracions tenen diferències en la seva construcció, totes tenen en comú unes certes parts presents en la majoria de motors. Aquestes són:

- Cilindres
- Pistons
- Bieles
- Volant d'inèrcia

⁵ Font: <http://www.stirlingbuilder.com/new-pop-can-engine>

3.2.1. Configuració alpha

La configuració alpha es caracteritza per estar formada per dos cilindres, generalment situats a 90°. Cadascun dels cilindres conté el seu corresponent pistó, que estan fixats al volant d'inèrcia en el mateix punt. Aquesta configuració és l'única que utilitza dos pistons de treball per a implementar el motor.

Cadascun dels cilindres correspon a un dels focus: en un cilindre hi ha el focus calent i a l'altre el fred. Els dos cilindres estan units a través d'un conducte pel qual circula el gas d'un a l'altre i viceversa. L'existència del conducte, permet col·locar-hi un regenerador.

Tal com s'ha explicat en el punt 3.1, el regenerador té l'objectiu d'aprofitar la calor extreta del refredament per aportar-la al moment de l'escalfament. En la figura 3.6 es pot apreciar una representació del muntatge d'un motor Stirling tipus alpha amb regenerador.

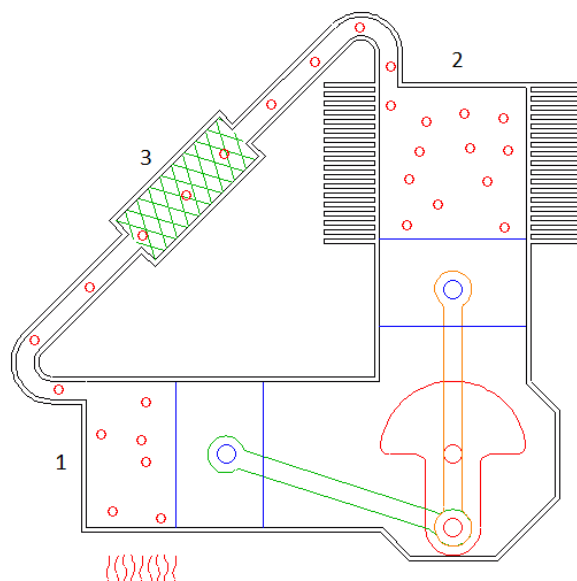


Figura 3.6: representació d'un motor Stirling tipus alpha⁶

En la mateixa figura 3.6, el número 1 indica el cilindre que actua com a focus calent i el 2 el que actua com a focus fred. El número 3 correspon al regenerador, situat en el tram central del conducte que uneix tots dos pistons.

⁶ Font: <http://www.animatedengines.com/vstirling.html>

Aquesta configuració, que és popular per una bona obtenció de potència amb un volum relativament reduït, és en la qual es basa el codi dissenyat.

Tot i aquest gran avantatge, els motors tipus alpha presenten també un problema important. En estar un dels pistons de treball en contacte amb el gas quan aquest està calent, les juntes d'estanqueïtat tendeixen a tenir una curta vida útil, produint problemes al funcionament del motor.

3.2.2. Configuració beta

Els motors Stirling tipus beta, només compten amb un cilindre en el qual s'hi troben, a la vegada, el focus fred i el calent, un a cada punta.

Donat que tot el gas del cicle està comprès en el mateix cilindre, fa falta un element dins del cilindre que el desplaci d'un focus a l'altre, per a tal que es realitzi el cicle: el pistó de desplaçament. De la mateixa manera que el pistó de treball, el de desplaçament també està fixat al volant d'inèrcia, però amb un cert desfasament (figura 3.7). El desfasament fa que els dos pistons es moguin de forma diferent al llarg del cicle, produint aquest moviment del gas d'un focus a un altre.

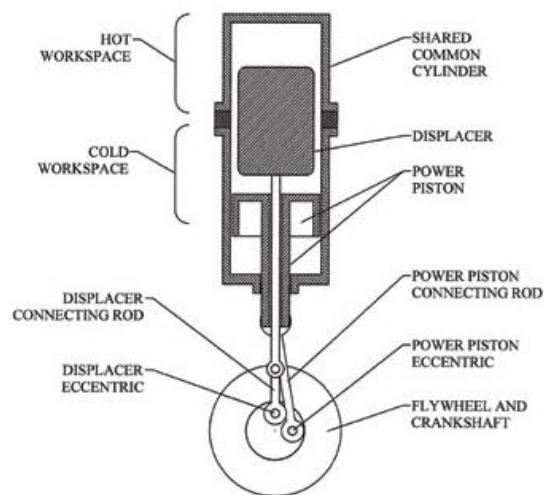


Figura 3.7: representació d'un motor Stirling tipus beta⁷

⁷ Font: <https://scholar.lib.vt.edu/ejournals/JOTS/v37/v37n2/foster.html>

En aquest tipus de motors, es soluciona el problema que hi ha en els de tipus alpha, ja que el pistó de treball no està en cap moment en contacte amb el gas a altes temperatures.

3.2.3. Configuració gamma

La configuració gamma, de la mateixa manera que l'alpha, està formada per dos cilindres. Tot i això, només té un pistó de treball.

Com es pot observar a la figura 3.8, en un dels dos cilindres, el de més volum, es on hi trobem els dos focus, tant el fred com el calent.

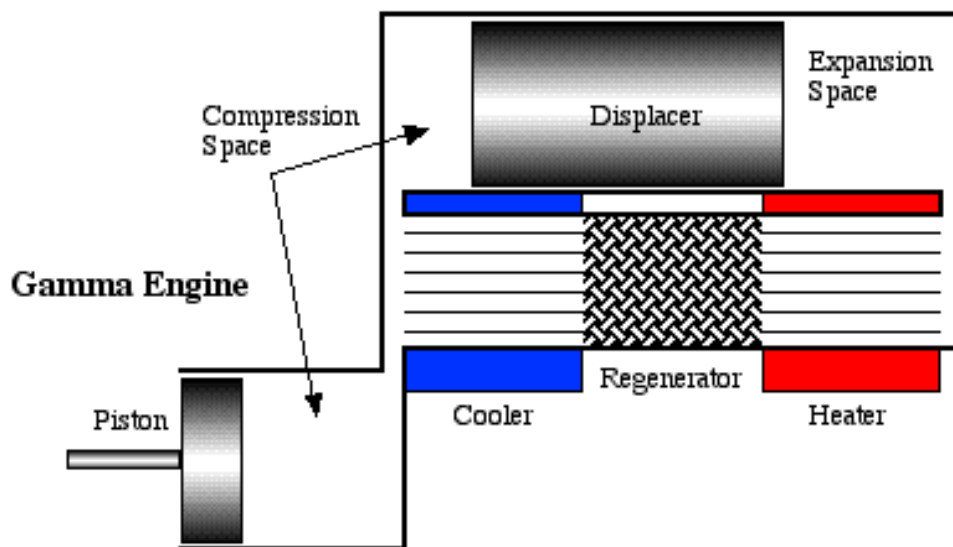


Figura 3.8: cilindres d'un motor Stirling tipus gamma⁸

De la mateixa manera que en la configuració tipus beta, en aquest cilindre s'hi troba un pistó de desplaçament, que mou el gas d'un focus a un altre. En l'altre cilindre, de menys volum, s'hi troba solament el pistó de treball.

⁸ Font: <https://www.ohio.edu/mechanical/stirling/engines/gamma.html>

Igual també que en la configuració beta, ambdós pistons estan fixats al volant d'inèrcia amb un cert desfasament, tal com s'observa a la figura 3.9

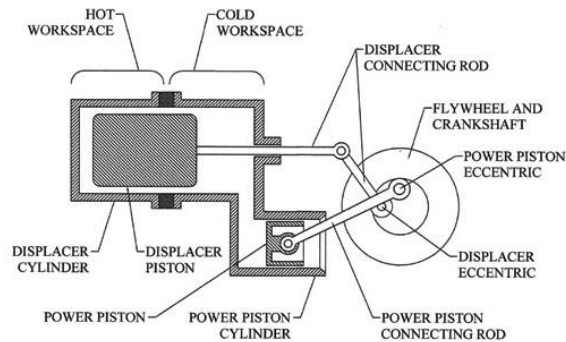


Figura 3.9: representació d'un motor Stirling tipus gamma [Virginia Tech]

Tots dos cilindres estan connectats per un petit conducte. Així, quan el gas està al focus calent, les altes pressions empenyen el cilindre de treball. En canvi, quan el gas està al focus fred, hi ha depressions que fan que el pistó de treball sigui succionat. Aquests moviments són els que permeten extreure treball del cicle.

3.3. Regenerador

Una de les aportacions més interessants de Robert Stirling en el moment de presentar la seva patent, és la figura del regenerador.

Com s'ha comentat, en el cicle del motor Stirling hi ha un procés on el fluid allibera calor i un altre en que necessita una aportació d'aquesta. L'objectiu del regenerador és aprofitar aquest alliberament de calor per a posteriorment escalfar el gas.

Amb aquest sistema, es pot augmentar el rendiment del motor, no millorant el treball obtingut però si requerint consumir menys energia per a funcionar.

La modelització del comportament del fluid en un regenerador no és trivial. És per això que al llarg de la història hi ha hagut molta recerca sobre ells i se n'ha dissenyat de molts tipus, amb formes i materials diversos.

El tipus de regenerador utilitzat en aquest estudi és l'anomenat "honeycomb", o bresca d'abella en català, forma que es pot veure a la figura 3.10. S'ha escollit aquest tipus perquè

és el que es podia simular de forma més fidedigne en el codi articulat, podent aconseguir així uns resultats més propers a la realitat.



Figura 3.10: diferents formes de regeneradors honeycomb⁹

3.4. Usos

Al llarg de la història, el motor Stirling no ha tingut una gran popularitat. Les seves limitacions i unes primeres implementacions en diverses tasques amb un resultat no gaire exitós i sense exportació al mercat (figura 3.11), han estat els responsables que hagi quedat en l'oblit.



Figura 3.11: prototip de cotxe desenvolupat per MTI i la NASA, propulsat per un motor Stirling¹⁰

⁹ Font: <http://www.pxjfhg.com/en/shownews.asp?id=541>

¹⁰ Font: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19880002196.pdf>

Actualment però, la importància guanyada per les energies renovables ha fet que el motor Stirling guanyi interès una altra vegada. En ser un motor de combustió externa, que pot funcionar amb varies fonts de calor de diferents tipus, és un potencial motor a tenir en compte per a utilitzar amb energies renovables.

Tot i que no ha tingut gaire rellevància i mai s'ha arribat a tenir gaire en compte a nivell industrial, hi ha moltes possibles aplicacions del motor Stirling. Algunes de les més significatives poden ser:

- propulsió mecànica
- escalfament o refredament, funcionant de manera inversa
- obtenció d'energia elèctrica

En el context d'aquesta última aplicació, és interessant l'obtenció d'energia elèctrica a partir d'energia solar. En aquesta línia, s'estan duent a terme actualment estudis, com per exemple el projecte EuroDISH, a la Plataforma Solar d'Almeria.

La implementació, que es pot veure a la figura 3.12, consisteix a instal·lar discs parabòlics de reflectors solars, que concentren aquesta energia en el focus calent del motor Stirling. Així, no és necessària cap tipus de combustió ja que la calor aportada per a fer funcionar el motor ve donada per l'energia solar reflectida.



Figura 3.12: instal·lació del projecte EuroDISH¹¹

¹¹ Font: <https://www.psa.es/es/instalaciones/discos/eurodish.php>

4. Codi de simulació

El codi que s'ha dissenyat tracta de representar el cicle termodinàmic que segueix el gas d'un motor Stirling tipus alpha en funcionament. El gas en qüestió s'ha considerat que és aire, estudiat com a gas ideal.

Concretament, l'objecte d'estudi és el comportament de les seves 3 variables termodinàmiques: pressió, volum i temperatura al llarg del cicle. Per a fer-ho, s'han utilitzat diferents conceptes i equacions provinents tant de la termodinàmica com de la termodinàmica.

La implementació del codi s'ha fet en llenguatge de programació anomenat "llenguatge M", propi del software MATLAB, ja que és una eina de programació coneguda i apropiada per a realitzar càlculs numèrics.

Tot el codi està construït en funció de l'angle girat pel volant d'inèrcia, alpha, en el qual estan connectats els dos pistons. Això determina la posició d'aquests en els cilindres i per tant, el volum del gas en cada instant (figures 4.1 i 4.2). Per tant, si es fixen les revolucions per minut, es pot conèixer alpha i el volum i, de les 3 variables termodinàmiques, faltarà determinar només la pressió i la temperatura.

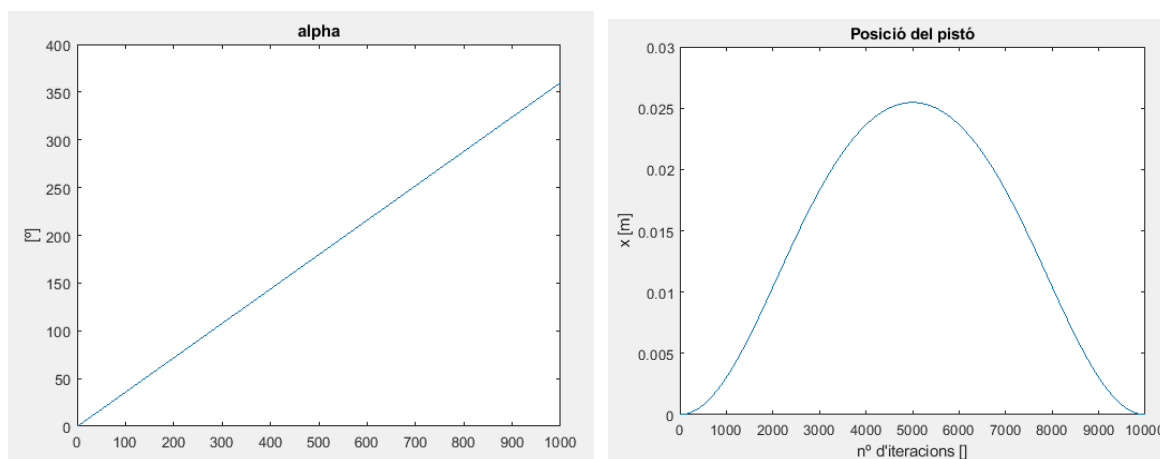


Figura 4.1: a l'esquerra, l'evolució de l'angle al llarg del cicle. A la dreta, l'evolució de la posició d'un dels pistons

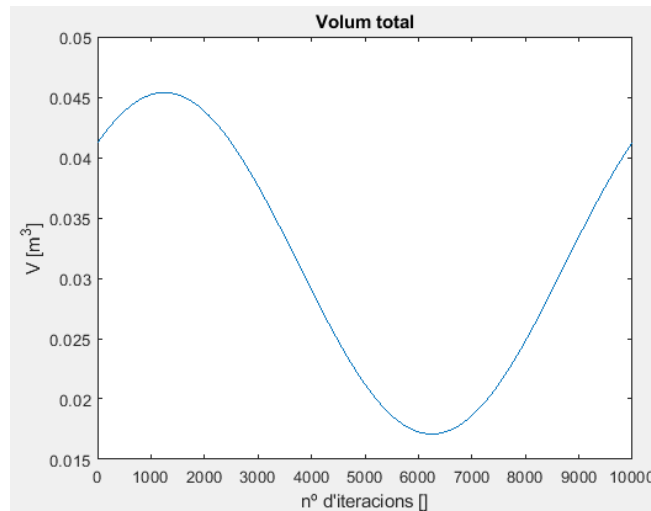


Figura 4.2: evolució del volum total al llarg del cicle

4.1. Procés d'elaboració del codi

El model matemàtic definit per descriure el cicle ha anat evolucionant al llarg del projecte, incrementant-ne la complexitat però també el seu rigor. A continuació es descriu aquest procés per donar una major visió global de la complexitat final del model matemàtic.

4.1.1. Fase I: Pressió i temperatura homogènies

En la primera fase, s'ha plantejat un model amb una pressió i temperatura homogènies per a tot el gas. El cicle es defineix a partir dels processos que segueix el cicle ideal i les equacions a implementar són les pròpies de la termodinàmica.

Així doncs, donat un increment d'alpha i unes condicions inicials de pressió i temperatura, es calcula primer la variació de volum donada per aquest increment d'alpha i, consegüentment, es calcula la pressió i la temperatura de l'instant següent.

Per als processos a temperatura constant el càlcul de la nova pressió es podia realitzar directament amb l'equació dels gasos ideals (Equació 4.1)

$$P \cdot V = n \cdot R \cdot T \quad (\text{Equació 4.1})$$

On P [atm] és la pressió del gas, V [l] el seu volum, n el nombre de mols, R [J / (K · mol)] és la constant dels gasos ideals i T [K] és la temperatura del gas.

Els processos a volum constant, però, no quedaven definits utilitzant l'equació dels gasos ideals. Calia conèixer l'evolució de la temperatura en ells. Per tant, tot i que el gràfic Pressió - Volum del cicle complet ideal queda perfectament definit, no es poden captar els fenòmens implicats al regenerador impossibilitant modelar el cicle real.

4.1.2. Fase II: Pressió homogènia i perfil de temperatura 1D al regenerador

Com a solució, sabent que en aquests processos és en els quals el gas passa a través del regenerador, es procedeix a realitzar un estudi més detallat.

A diferència dels cilindres, en els quals es considera una temperatura homogènia per a tot el gas, en el regenerador s'imposa un perfil de temperatures unidimensional (1D). Per tant, es discretitza el regenerador, dividint-lo en varis nodes i estudiant-lo mitjançant mètodes numèrics. A la figura 4.3 es pot veure un esquema de la discretització realitzada.

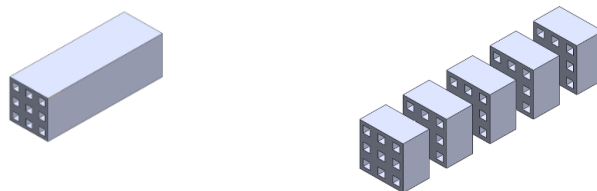
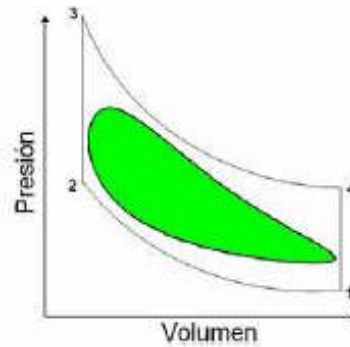


Figura 4.3: representació de la discretització del regenerador [Elaboració pròpia]

La resolució de l'evolució de temperatures al regenerador i als cilindres es realitza mitjançant balanços de potència tèrmica.

Amb aquesta configuració es reproduïx l'evolució tèrmica del fluid a través del regenerador en aquells processos a volum constant, podent així definir totes les variables del gas. Tot i això, en el cicle real del motor Stirling –que correspon a l'àrea verda de la figura 4.4-, el límit entre els processos isotèrmics i isocors no és definit i els processos coexisteixen. Per tant, la simulació 1D del regenerador s'ha d'integrar de forma natural al càlcul de tots els processos.

Figura 4.4: cicle Stirling real¹²

4.1.3. Fase III: cicle real

En aquesta fase es deixa de treballar amb una seqüenciació temporal de les 4 etapes del cicle ideal i es passa a calcular els balanços de potència de forma continuada al llarg de tot el cicle. El model resultant ha presentat dos grans inconvenients:

- 1) El balanç de potència tèrmic emprat pel càlcul de les temperatures als cilindres i al regenerador no inclou l'efecte del treball mecànic.
- 2) La massa total del sistema no es conserva.

4.1.4. Fase IV: cicle real amb balanç de potència total i conservació de la massa

En aquesta darrera fase del model matemàtic del cicle Stirling total, es planteja resoldre un balanç de potència global, incloent la potència tèrmica i el treball mecànic, en tots els punts del sistema. A més, es té especial cura en la discretització per tal de conservar la massa total del sistema.

El model resultant és el model definitiu del present estudi i és el que es descriu de forma detallada a continuació.

Com es pot comprovar, la complexitat ha augmentat de forma considerable en el procés: inicialment es pretenia representar uns processos amb unes equacions simplificades i,

¹² Font: <https://proyecto-motorstirling.webnode.cl/fisica-del-motor/ciclo-real/>

finalment, s'ha elaborat un codi que representa el cicle amb prou detall com per captar els fenòmens implicats en el regenerador i mantenir una coherència global en tot el sistema.

4.2. Hipòtesis i simplificacions

Com ja s'ha comentat, l'objectiu principal del programa és mostrar com evolucionen les variables termodinàmiques del gas durant el cicle d'Stirling, concretament la pressió i la temperatura, que són les variables que queden lliures havent fixat el volum segons l'angle alpha. Així doncs, les primeres hipòtesis realitzades refereixen a aquestes dues variables.

En primer lloc, la pressió: s'ha considerat en tot moment una pressió genèrica i uniforme per a tot el gas, indistintament de la posició d'aquest. És a dir, es considera que els gradients interns de pressió (per exemple, entre els cilindres) són negligibles. D'aquesta manera, s'ha pogut estudiar tot el volum de gas com un sistema tancat, en el qual ni hi entra ni hi surt massa, per determinar-ne la pressió.

De cara a la representació gràfica del cicle també es necessita una temperatura genèrica per a tot el gas en cada instant, però a diferència de la pressió, si que s'ha tingut en compte la localització per al seu càlcul. En concret, s'ha treballat de la següent manera:

- en ambdós cilindres s'ha suposat una temperatura instantània uniforme pel volum de gas que hi ha contingut
- la resta del gas es troba contingut al regenerador i, com ja s'ha explicat, aquest està dividit en diferents nodes amb propietats instantànies uniformes en cadascun. Així, cada node té també la seva pròpia temperatura, obtenint una distribució de temperatures variable en la direcció longitudinal del regenerador.

Per al càlcul de les temperatures als cilindres, s'han utilitzat 4 balanços d'energia, un pel gas de cada cilindre i un per les parets de cada cilindre, que s'expliquen detalladament a la Secció

5.3.2. Les hipòtesis principals d'aquests balanços són:

1. Sistema obert: hi ha un flux de massa que entra o surt del sistema
2. Temperatura instantània uniforme en cada material (gas/sòlid)
3. Superfície de transferència de calor entre el cilindre i l'exterior constant
4. Superfície de transferència de calor entre el cilindre i el gas interior variable, en funció del moviment del pistó

5. Inèrcia tèrmica al cilindre, que fa que les parets acumulin energia
6. Propietats tèrmiques del cilindre constants al llarg del cicle
7. Pistó adiabàtic
8. No es considera fricció en el moviment dels pistons

Al regenerador, per la seva banda, en estar discretitzat s'ha aplicat un balanç d'energia en cada volum discret, tractant-los també com sistemes oberts. En cadascun d'aquests volums, es suposa una temperatura i unes propietats del gas uniformes. Degut a la complexitat de l'acoblament entre els diferents punts del sistema total, en aquest estudi s'ha considerat que el sòlid regenerador té una temperatura constant i invariant durant tot el cicle. Aquesta és una simplificació molt rellevant en el funcionament del cicle i la seva eficiència, però necessària en aquesta primera fase de l'estudi del cicle complet.

Les potències tèrmiques bescanviades entre el gas i les parets dels cilindres s'han calculat a partir de coeficients globals de transferència de calor que poden imposar-se directament o calcular-se a partir de correlacions tipus Dittus-Boelter. En canvi, la potència transferida entre el gas i les parets del regenerador s'ha considerat que segueix un fenomen purament difusiu. Aquesta darrera simplificació s'ha de re-definir en següents etapes del projecte global.

En cap moment es tenen en compte pèrdues energètiques per dissipació viscosa.

4.3. Equacions

En el següent apartat es comentaran les diferents equacions utilitzades, tant les referents als balanços dels que s'ha parlat com a les que contribueixen a la implementació en general del codi.

4.3.1. Geometria i volum

Com s'ha comentat abans, tot el codi va a remolc de l'angle α girat pel volant d'inèrcia. A partir d'aquest angle, aplicant geometria es poden determinar les posicions dels pistons, coneixent així al seu torn el volum que hi ha en cada cilindre.

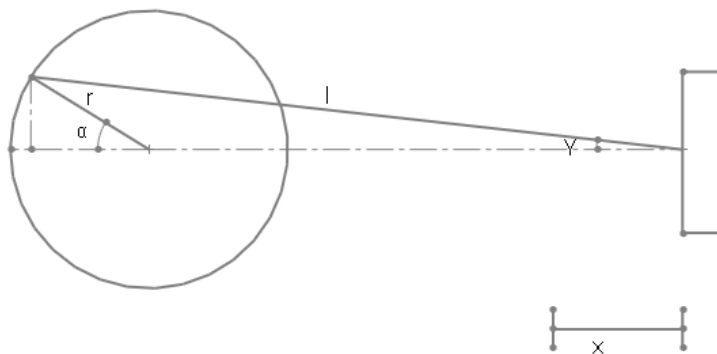


Figura 4.5: representació de la geometria considerada pels pistons [Elaboració pròpia]

A la figura 4.5 es poden veure els paràmetres considerats per al càlcul del moviment dels pistons a partir de alpha. Les equacions usades són les següents:

$$x = r - r \cdot \cos(\alpha) + l - l \cdot \cos(\gamma) \quad (\text{Equació 4.2})$$

$$\cos(\gamma) = \sqrt{1 - \sin^2(\gamma)} \quad (\text{Equació 4.3})$$

$$\sin(\gamma) = \frac{r \cdot \sin(\alpha)}{l} \quad (\text{Equació 4.4})$$

$$\lambda = \frac{r}{l} \quad (\text{Equació 4.5})$$

On x [m] és la distància a la que està el pistó del seu punt mort inferior, r [m] és el radi del volent d'inèrcia, l [m] és la llargada de la biela que uneix el volant amb el pistó, α [rad] és l'angle girat i γ [rad] és l'angle entre la biela i la referència horitzontal. Així, ajuntant les equacions 4.2, 4.3, 4.4 i 4.5 s'obté l'equació 4.6:

$$x = r \cdot (1 - \cos(\alpha)) + l \cdot (1 - \sqrt{1 - \lambda^2 \cdot \sin^2(\alpha)}) \quad (\text{Equació 4.6})$$

En la qual la variable d'estudi x només depèn de l'angle girat. A partir d'aquesta equació, es pot definir el volum que hi ha en el cilindre de la següent forma:

$$V_{cil} = V_{cc} + (CA - x) * S \quad (\text{Equació 4.7})$$

On V_{cil} [m³] és el volum de gas contingut al cilindre, V_{cc} [m³] és un cert volum que sempre

queda buit en el cilindre, CA [m] és el recorregut del pistó i S [m²] és la secció del cilindre. Així, el volum total es calcula com la suma dels volums dels cilindres amb el volum del regenerador, que té un valor constant.

4.3.2. Balanços d'energia

El següent grup d'equacions necessari per a la implementació del codi és el dels balanços d'energia realitzats tant a les parets dels cilindres com al volum de gas dins d'aquests. En els volums de fluid, s'ha realitzat un balanç de potència tèrmica i mecànica considerant sistema obert. En els volums sòlids (les parets dels cilindres) s'ha fet únicament un balanç de potència tèrmica. A continuació s'expliquen aquests balanços de forma més detallada:

1) Balanços de les parets

El balanç de potència tèrmica aplicat és:

$$q_{acumulada} = q_{entra} - q_{surt} + q_{generada} \quad (\text{Equació 4.8})$$

On q [W] representa potència i el subíndex de cadascuna indica l'origen d'aquesta. En cap dels casos es considera que hi hagi generació, així que el terme corresponent mai apareixerà en expressions posteriors.

Desenvolupant el balanç per a les parets del cilindre s'obté:

$$\frac{d}{dt} \int_{V_w} \rho_w \cdot C_{p_w} \cdot T \cdot dV_w + q_{w-f} = q_E \quad (\text{Equació 4.9})$$

On V_w [m³] és el volum de la paret, ρ_w [kg/m³] n'és la densitat, C_{p_w} [J/(K·kg)] és la calor específica a pressió constant d'aquesta, T [K] és la seva temperatura, q_{w-f} [W] és la potència que la paret transfereix al fluid i q_E [W] és la potència que intercanvia la paret amb el que l'envolta per la part exterior. Contextualitzant l'equació 4.9 segons l'establert en la 4.8, es pot definir que: $\frac{d}{dt} \int_{V_w} \rho_w \cdot C_{p_w} \cdot T \cdot dV_w$ correspon a la potència que acumula la paret, q_{w-f} és la potència que perd el sistema i q_E és la que entra o la que guanya.

Per a la correcta implementació del codi, s'ha discretitzat el temps en petits intervals. Tenint en compte que les propietats i el volum del sòlid es mantenen en cadascun d'ells, s'obté:

$$\rho_w \cdot C_{p_w} \cdot V_w \frac{(T_w^{n+1} - T_w^n)}{\Delta t} + q_{w-f} = q_E \quad (\text{Equació 4.10})$$

On T_w^n [K] és la temperatura de la paret en un instant determinat, T_w^{n+1} [K] és la mateixa temperatura a l'instant següent i Δt [s] representa la quantitat de temps que passa entre un instant d'estudi i un altre.

2) Balanços al gas

Els balanços realitzats al gas tenen el seu origen en el primer principi de la termodinàmica (equació 4.11) combinat amb la relació entre energia interna i entalpia (equació 4.12):

$$\Delta U = Q - P \cdot \Delta V + H_{in} - H_{out} \quad (\text{Equació 4.11})$$

$$\Delta U = \Delta H - \Delta(P \cdot V) \quad (\text{Equació 4.12})$$

En les quals, ΔU [J] correspon a la variació d'energia interna del gas en un instant de temps, Q [J] a la calor rebuda o extreta, P [Pa] a la pressió global del gas, ΔV [l] a la variació de volum del sistema considerat, H_{in} i H_{out} [J] l'entalpia absoluta que entra i surt del sistema respectivament, ΔH [J] la variació d'entalpia del sistema i $\Delta(P \cdot V)$ [J] la variació de la multiplicació entre la pressió i el volum entre 2 instants.

Un cop arribat aquí, és convenient tornar a dividir la casuística en 2. S'ha comentat ja que per a trobar la pressió, es considera el total del gas com un sistema tancat, que es tractarà com el cas d'estudi A. En canvi, tant en els cilindres com en els nodes del regenerador, s'ha considerat que són sistemes oberts, ens els quals hi ha fluxos màssics que entren i surten. Aquesta serà la casuística B.

A) Sistema global: sistema tancat

Si no hi ha cabal d'entrada i sortida al sistema, a l'equació del primer principi (Equació 4.11) desapareixen els termes que fan referència a l'entalpia d'entrada i sortida. Coneixent el càlcul de la variació d'energia interna per a un gas ideal:

$$\Delta U = m \cdot C_v \cdot \Delta T \quad (\text{Equació 4.13})$$

-on m [kg] és la massa del sistema, C_v [J/(K·kg)] és la calor específica a volum constant i ΔT [K] és la variació de temperatura del sistema-, i combinant-la amb l'equació d'estat dels gasos

ideals (Equació 4.1), es pot arribar a la següent equació pel càlcul global de la pressió:

$$\Delta P = \frac{R}{C_v \cdot V} \left[Q - \left(1 + \frac{C_v}{R} \right) \cdot P \cdot \Delta V \right] \quad (\text{Equació 4.14})$$

Que, convenientment discretitzada temporalment, queda:

$$P^{n+1} = P^n + \Delta t \cdot \frac{R}{C_v \cdot V^n} \left[q^n - \left(1 + \frac{C_v}{R} \right) \cdot P^n \cdot \left(\frac{V^{n+1} - V^n}{\Delta t} \right) \right] \quad (\text{Equació 4.15})$$

On P^{n+1} [Pa] serà la pressió a l'instant següent, en funció de la pressió actual (P^n [Pa]), de la potència calorífica que entra al sistema en aquell instant (q^n [W]) i de la variació de volum que experimentada entre l'instant estudiat i el següent.

B) Sistema local: sistema obert

Com ja s'ha explicat, per al càlcul de les temperatures en les diferents parts del sistema, s'han realitzat els balanços considerant-les com a sistemes oberts. Com a pas intermediari per arribar a la temperatura, s'ha treballat amb l'entalpia per a facilitar els càlculs. Per a la seva determinació, s'han combinat les equacions 4.11 i 4.12 obtenint:

$$\Delta H = Q + V \cdot \Delta P + H_{in} - H_{out} \quad (\text{Equació 4.16})$$

Igual que amb la resta de variables, s'ha discretitzat l'equació i adaptat per a la seva integració en el codi, resultant-ne:

$$H^{n+1} = H^n + \Delta t \cdot \left[q^n + V^n \cdot \left(\frac{P^{n+1} - P^n}{\Delta t} \right) + \dot{m}_{in}^n \cdot C_p \cdot T_{in}^n - \dot{m}_{out}^n \cdot C_p \cdot T_{out}^n \right] \quad (\text{Equació 4.17})$$

On es pot apreciar que, per al càlcul de l'entalpia instantània d'entrada i de sortida s'ha utilitzat l'expressió:

$$\dot{H}_i = \dot{m}_i^n \cdot C_p \cdot T_i^n \quad (\text{Equació 4.18})$$

En la qual, \dot{m}_i^n [kg/s] correspon al cabal màssic d'entrada o sortida respectivament, C_p [J/kg·K] és la calor específica a pressió constant i T_i^n [K] indica la temperatura a la qual es troba aquest cabal màssic.

Degut a aquests cabals màssics d'entrada i sortida i a la compressibilitat del gas, la massa a les diferents parts del motor considerades varia en cada instant. A partir d'un balanç de massa es pot conèixer la quantitat que se n'haurà de considerar en cada punt al següent interval:

$$m^{n+1} = m^n + \dot{m}_{in}^n \cdot \Delta t - \dot{m}_{out}^n \cdot \Delta t \quad (\text{Equació 4.19})$$

A tot això, l'entalpia només és un pas intermedi en el camí a calcular la variable que es vol conèixer, que és la temperatura. Aquesta, coneixent l'entalpia i la massa es pot deduir fàcilment:

$$T^{n+1} = \frac{H^{n+1}}{m^{n+1} \cdot Cp} \quad (\text{Equació 4.20})$$

Tant a l'equació 4.15 per al càlcul de la pressió com a l'equació 4.17 per al d'entalpies, hi apareix el terme q^n , definit com la potència calorífica que entra al volum considerat, en un instant de temps. Aquest terme s'ha calculat de forma diferent en els cilindres –on s'ha modelitzat com a convecció, seguint la llei de refredament de Newton (Equació 4.21)-, i en els nodes del regenerador, on s'ha modelitzat segons el mecanisme de difusió, seguint la primera llei de Fourier: (Equació 4.22):

$$q_{cilindres}^n = h \cdot A_{conv} \cdot (T_{paret} - T_{fluid}) \quad (\text{Equació 4.21})$$

$$q_{nodes}^n = \frac{\lambda_{fluid}^n \cdot A_{conv} \cdot (T_{paret} - T_{fluid})}{0,5 \cdot y} \quad (\text{Equació 4.22})$$

En les quals $h[W/(K \cdot m^2)]$ és el coeficient superficial de transferència de calor de l'aire en els cilindres, $A_{conv}[m^2]$ correspon a l'àrea del regenerador que està en contacte directe amb el fluid aire –i s'aplica a ambdues equacions de la mateixa manera-, $(0,5 \cdot y)[m]$ correspon a la distància que hi ha entre la paret del regenerador i el centre dels elements de la discretització, on es considera que hi ha el node i $\lambda_{fluid}^n [W/K \cdot m]$ és la conductivitat tèrmica de l'aire, que depèn de la temperatura segons:

$$\lambda_{fluid}^n = (3.807 + 0.074 \cdot T^n) \cdot 10^{(-3)} \quad (\text{Equació 4.23})$$

4.3.3. Conservació de la massa i càlcul dels cabals

Per a calcular els cabals màssics d'entrada o sortida tant als cilindres com als diferents nodes del regenerador, s'ha aplicat una equació de continuïtat (conservació de la massa) als diferents volums, referida a les superfícies d'entrada o sortida de la massa:

$$\int_{V_i} \frac{d\rho}{dt} dV_i + \int_{S_i} \rho \cdot v \cdot dS_i = 0 \quad (\text{Equació 4.24})$$

En la qual ρ [kg/m³] és densitat, v [m/s] és velocitat del fluid i la seva multiplicació amb la secció és equivalent al cabal màssic de fluid:

$$\dot{m} = \rho \cdot v \cdot S \quad (\text{Equació 4.25})$$

Discretitzant l'equació de continuïtat aplicada (equació 4.24), es pot conèixer el flux màssic que entra en un volum o en surt:

$$\dot{m}_{in}^{n+1} - \dot{m}_{out}^{n+1} = \frac{\rho^{n+1} \cdot V^{n+1} - \rho^n \cdot V^n}{\Delta t} \quad (\text{Equació 4.26})$$

Aquest cabal màssic és el que posteriorment s'utilitzarà en els càlculs de les altres variables. Això permet que la massa total de gas considerada inicialment es mantingui constant al llarg del temps.

4.4. Algoritme

Les equacions descrites en l'anterior apartat, s'han encadenat en un cert ordre, seguint la seva implementació òptima. D'aquesta manera, s'ha definit l'algoritme que segueix el codi per a fer els càlculs necessaris, que es representa a la següent figura:

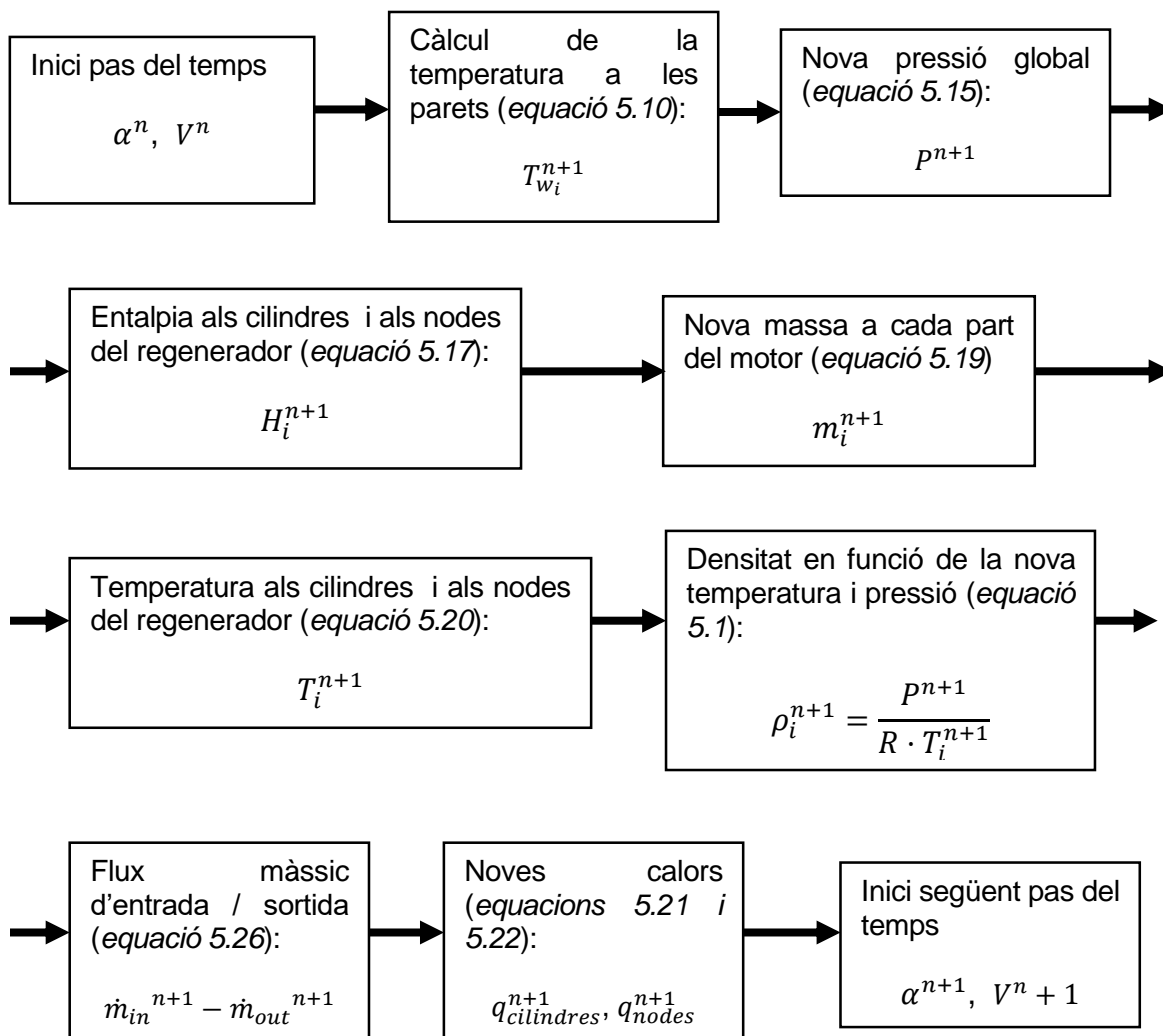


Figura 4.6: esquema de l'algoritme seguit [Elaboració pròpia]

En primer lloc, es defineix el nombre d'interval que es vol que tingui el cicle complet. Això divideix l'angle α en els n intervals que ha de tenir cada cicle definint, al seu torn, el volum que hi haurà en cada instant.

A continuació, s'entra al bucle principal. El primer càlcul que es realitza és el de les noves temperatures de la paret dels cilindres, que s'utilitzaran posteriorment per al càlcul de les calors als cilindres. La següent variable que es calcula és la pressió genèrica. En ser un dels primers càlculs, es fa de forma explícita, és a dir, agafant les variables de les que depèn de l'interval anterior.

El càlcul de les entalpies, que ve a continuació, canvia lleugerament segons el signe dels cabals màssics. Com que aquest s'avalua en superfícies i , el que realment es calcula és la

diferència entre el d'entrada i el de sortida, s'ha treballat amb aquesta diferència com a variable. Així:

- Per als dos cilindres, la variable cabal màssic és: $\dot{m}_{out}^{n+1} - \dot{m}_{in}^{n+1}$, definida positiva quan surt del cilindre
- Per a cada node del regenerador, el cabal màssic és el que surt d'aquell node, en sentit del cilindre 1 al cilindre 2.

La figura 4.7 exemplifica aquesta metodologia:

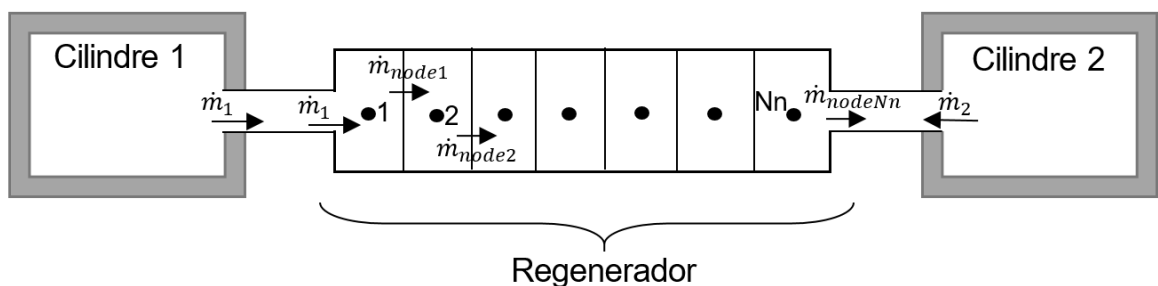


Figura 4.7: esquema del motor estudiat, amb els sentits dels cabals màssics definits

[Elaboració pròpia]

Tal com es veu, en el primer node del regenerador hi entra directament el cabal màssic que surt del cilindre 1. Per a conciliar els cabals màssics al final del regenerador, el cabal màssic de sortida de l'últim node és el mateix que el de sortida del cilindre 2, amb el signe canviat.

Tornant al càlcul de les entalpies, per als cilindres s'han definit 2 casos:

- 1) El cabal màssic de cilindre (sortint) és positiu: el cilindre perd energia i, la quantitat de massa que marxa, marxa a la temperatura del propi cilindre.
- 2) El cabal màssic del cilindre és negatiu, és a dir entra. En aquest cas, el cilindre guanya energia i, el cabal màssic que entra, ho fa a la temperatura del node del regenerador més proper al cilindre.

De la mateixa manera que en els cilindres, en els diferents nodes del regenerador s'han definit 4 casos:

- 1) Si el cabal màssic del node anterior és positiu (entra) i el del propi node és positiu també (surt). El cabal que entra, ho fa amb la temperatura del node anterior i el que

surt, amb la del mateix node que s'està considerant.

- 2) Si el cabal màssic del node anterior és negatiu (surt) i el del propi node és positiu (surt). En aquest cas, el node perd potència i, ambdós cabals màssics surten a la temperatura del propi node.
- 3) Si el cabal màssic del node anterior és negatiu (surt) i el del propi node és negatiu (entra). Si això es dona, el cabal del node anterior surt amb la temperatura del propi node i l'altre, entra amb la temperatura del node següent.
- 4) Si el cabal màssic del node anterior és positiu (entra) i el del propi node és negatiu (entra). En aquest cas, en el qual el node guanya potència, el cabal del node anterior entra amb la temperatura d'aquell node i el cabal del propi, entra també però amb la temperatura del node següent.

Quan les entalpies ja són conegudes, es calcula la massa que hi haurà als cilindres i a cada node. Aquest càlcul, com ja s'ha comentat, es fa mitjançant un balanç de massa que defineix que la nova serà la que ja hi havia en l'interval anterior restant-li o sumant-li la quantitat de massa que hagi entrat en l'interval.

Una vegada es té la massa de cada cel·la del motor, amb ella i la nova entalpia, es calcula la temperatura que es considerarà pel següent interval. Immediatament després, amb la temperatura i la pressió s'obté la nova densitat, necessària per al càlcul dels nous cabals màssics.

A la figura 4.7, s'ha mostrat com estaven definits aquests cabals màssics i s'ha explicat la diferència entre com estaven definits els que eren dels cilindres dels que corresponien als nodes del regenerador. En els cilindres, es calcula el balanç entrada – sortida de cabal màssic, tal com es mostra a l'equació 4.26.

En el regenerador en canvi, la informació viatja d'un cilindre cap a l'altre. Per tant, pot aparèixer una incompatibilitat entre l'últim node del regenerador i el cilindre veí.

Per a intentar mitigar aquest efecte i no anar acumulant error en la simulació, s'ha decidit utilitzar la següent metodologia:

- En les iteracions imparelles, els cabals màssics es calculen a partir del de sortida del cilindre 2 i el del cilindre 1, es calcula a partir del cabal màssic del primer node.
- En les iteracions parelles, es calculen a partir del cabal de sortida del cilindre 1 i el cabal del cilindre 2 es calcula a partir del de l'últim node del regenerador.

D'aquesta manera, el càlcul dels nous cabals màssics al regenerador es fa adaptant l'equació 4.26 a:

$$\dot{m}_{out}^{n+1} = \dot{m}_{in}^{n+1} - \frac{\rho^{n+1} \cdot V^{n+1} - \rho^n \cdot V^n}{\Delta t} \quad (\text{Equació 4.27})$$

On \dot{m}_{in}^{n+1} serà, depenent del sentit de càlcul que s'estigui considerant, el cabal del node de l'esquerra o de la dreta.

A continuació, abans del càlcul de les potències, tan sols a nivell informatiu per a poder fer la seva posterior representació es calcula la temperatura global del sistema. La forma de calcular-la és una mitja ponderada de les temperatures dels cilindres i del regenerador segons el volum:

$$T^{n+1} = \frac{T_{cil1}^{n+1} \cdot V_{cil1}^{n+1} + T_{cil2}^{n+1} \cdot V_{cil2}^{n+1} + T_{reg}^{n+1} \cdot V_{reg}^{n+1}}{V_{total}^{n+1}} \quad (\text{Equació 4.28})$$

Finalment, es calculen les noves calors del sistema que, en la següent iteració, permetran calcular les temperatures a les parets dels cilindres i la nova pressió global, començant així la nova iteració.

5. Verificació

Una vegada acabat el procés de disseny i implementació del codi, s'ha dut a terme un procés de verificació per a comprovar la correcta implementació i detectar errors conceptuals que s'haguessin pogut cometre. Aquesta verificació ha constatat de dues parts. En la primera, s'han fet comprovacions en varies situacions simplificades per a veure si el comportament en aquestes condicions era l'esperat. En la segona, s'han contemplat possibles millores que es podrien fer al model dissenyat.

5.1. Definició de les condicions de treball

A continuació s'inclou una taula en la qual es mostren els valors de les dimensions i propietats utilitzats en la realització de la simulació:

Variable	Valor [unitats]
Constant dels gasos ideals (R)	287,54 [J / (K·kg)]
Calor específica a pressió constant (Cp)	1006,4 [J / (K·kg)]
Calor específica a volum constant (Cv)	718,86 [J / (K·kg)]
Massa molar de l'aire (M)	0,0289 [kg / mol]
Volum dels cilindres (Vcu)	0,02 [m ³]
Volum d'espai lliure (Vcc)	0,002 [m ³]
Revolucions per minut (RPM)	5000 [min ⁻¹] (523,599 [rad / s])
Pas del temps (Δt)	0,0000012 [s]
Nombre de nodes al regenerador (Nn)	50 nodes
Longitud del regenerador (Lr)	0,6 [m]
Distància entre nodes (deltax)	0,012 [m]
Longitud del costat d'un conducte del	0,003 [m]

regenerador (costat)	
Nombre de conductes al regenerador (Numcond)	1600
Pressió inicial	1 [atm] (101325 [Pa])
Volum inicial	0,04122 [m ³]
Temperatura inicial	15 [°C] (288,15 [K])
Calor proporcionada pel combustible (qcomb)	800 [kW]
Temperatura a l'exterior del cilindre 2 (TE2)	10 [°C] (283,15 [K])
Coeficient de convecció (U)	50 [W / (K·m ²)]
Densitat de les parets dels cilindres (row)	785 [kg / m ³]
Calor específica de les parets dels cilindres (Cpw)	49 [J / (kg·K)]

Taula 5.1: valors de les diferents variables utilitzades

5.2. Procés de verificació

El primer cas d'estudi per a la comprovació ha estat suprimir la convecció als cilindres i no considerar tampoc transferència de calor al regenerador. Així, s'està verificant la part de potència mecànica del sistema global.

A la figura 5.1 es mostra l'evolució de la temperatura i pressions globals al llarg del cicle. S'observa que tant la temperatura com la pressió del sistema acaben el cicle aproximadament al mateix valor al qual l'havien començat.

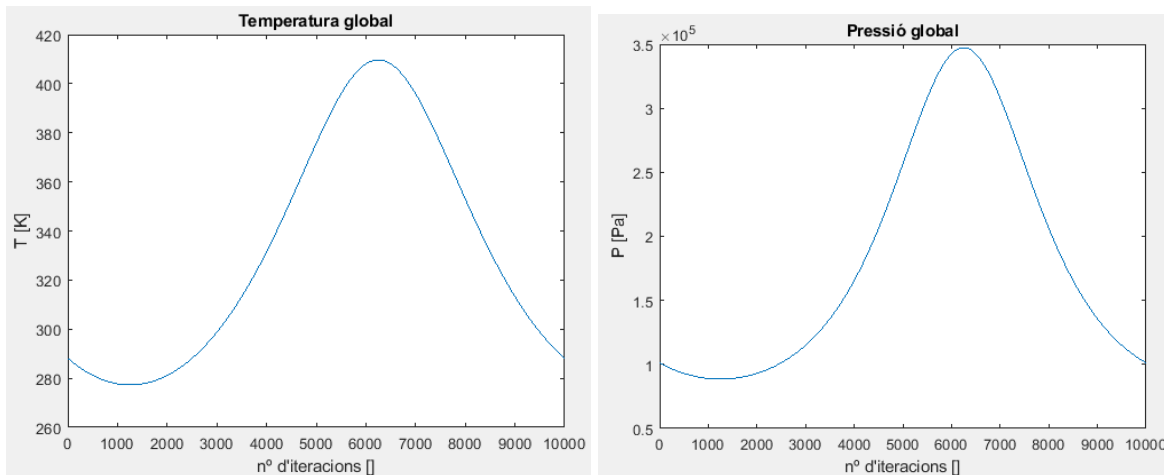


Figura 5.1: d'esquerra a dreta: Temperatura i Pressió globals del sistema respectivament
[Elaboració pròpia]

Per a més seguretat, es realitza la comprovació de l'evolució de la temperatura en uns quants cicles, obtenint:

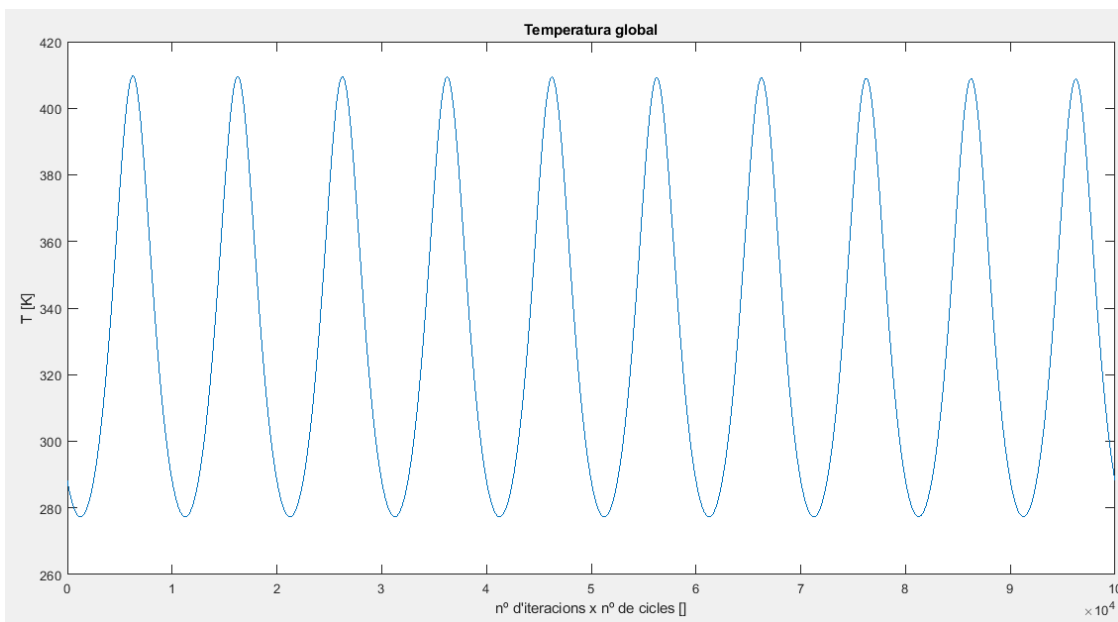


Figura 5.2: evolució de la temperatura global en 10 cicles [Elaboració pròpia]

Com es pot observar a la figura 5.2, sembla que l'evolució de la temperatura al llarg dels cicles es manté constant tal i com s'espera.

Havent comprovat aquest cas, s'incorpora la transferència de calor al regenerador, sense

tenir en compte encara la convecció als cilindres. L'evolució de la temperatura global en aquest cas és:

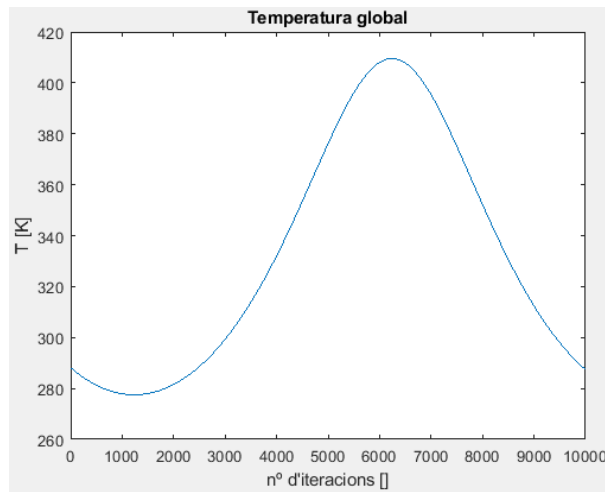


Figura 5.3: evolució de la temperatura global en 1 cicle [Elaboració pròpia]

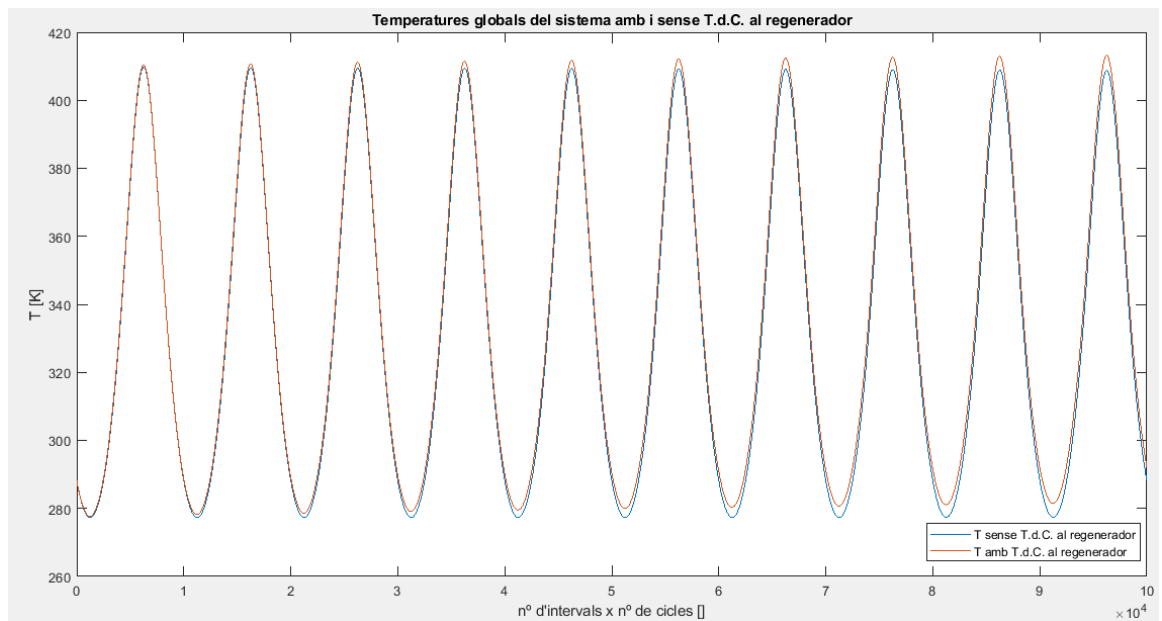


Figura 5.4: comparativa de l'evolució de la temperatura global en 10 cicles, segons si es considera la transferència de calor al regenerador o no [Elaboració pròpia]

Com es veu en les figures 5.3 i 5.4, la incorporació de la transferència de calor del regenerador al fluid, fa que la temperatura global del segon augmenti lleugerament, tal i com s'esperava en els resultats.

Finalment, es procedeix a incorporar també la convecció als cilindres i simular així el motor amb totes les condicions.

Els resultats obtinguts en aquesta prova definitiva han estat semblants als obtinguts sense considerar la convecció als cilindres. Aquesta situació és lògica ja que, tot i que en el primer cilindre hi entra calor, en el segon en surt, per tant el balanç global de calor del sistema no hauria de diferir molt respecte el de la situació anterior.

Els resultats presentats anteriorment corresponen a l'encesa del motor. És a dir, com a usuari s'estan fixant les condicions inicial (més fredes) del sistema i aquest evoluciona consegüentment. Perseguint arribar a un estat estacionari, es realitza una simulació de més cicles en aquesta última situació. Així, en una simulació de 20 cicles (figura 5.5) s'observa una certa tendència creixent:

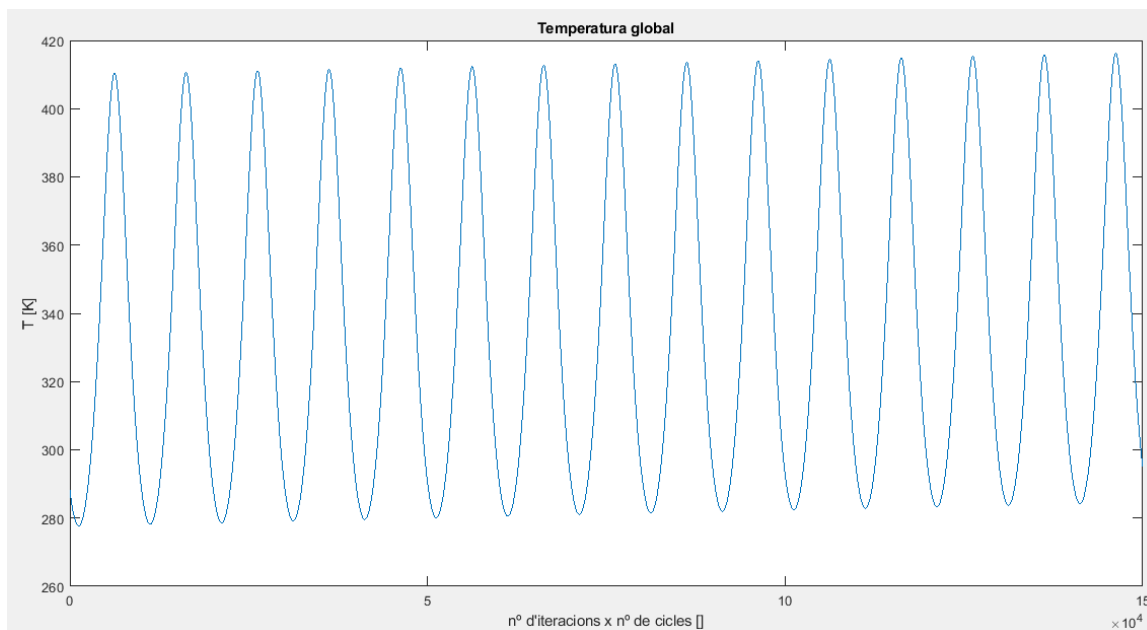


Figura 5.5: evolució de la temperatura global en 15 cicles, considerant transferència de calor al regenerador i als cilindres [Elaboració pròpia]

Prenent un cicle qualsevol de la figura 5.5 i analitzant l'evolució del volum i la pressió globals del cicle en el temps, s'obté una única línia d'anada i tornada en comptes d'un cicle tancat amb àrea finita (figura 5.6). Per tant, el cicle no està realitzant treball. Aquesta representació del cicle com a una única línia és coherent amb el fet de no modelar la inèrcia tèrmica del regenerador. Per tant, és satisfactòria tenint en compte l'estat actual del model.

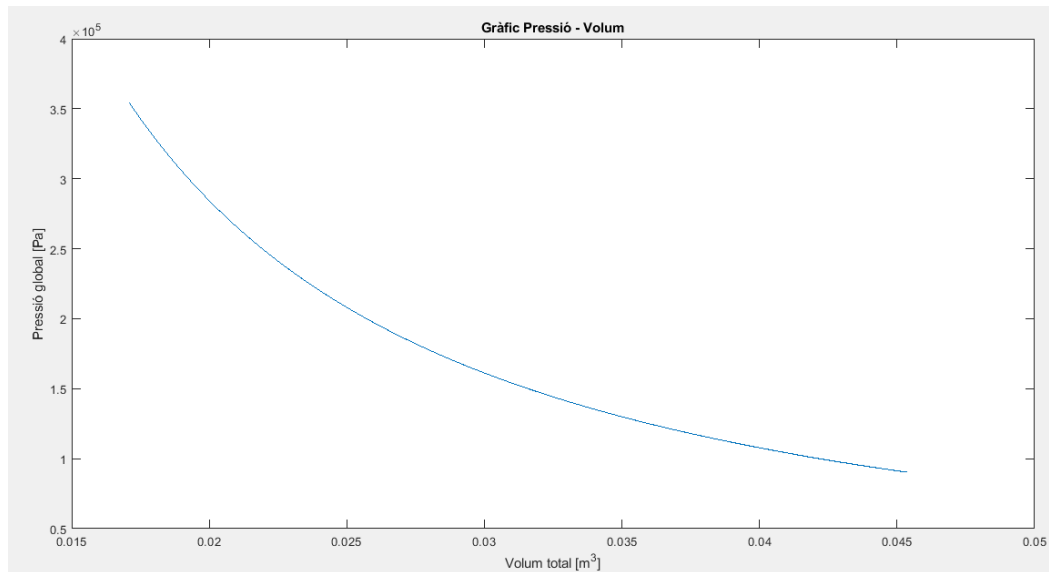


Figura 5.6: gràfic Pressió - Volum [Elaboració pròpia]

Tornant a la petita tendència creixent de la temperatura global del sistema, s'ha incrementat el nombre de cicles estudiats per tal d'intentar aconseguir un cicle desenvolupat i estable al llarg del temps. Els resultats (figura 5.7) mostren com, entre els cicles 80 i 100, el sistema comença a oscil·lar d'una forma més marcada, fet que el porta finalment a desestabilitzar-se, tal com es mostra a la figura 5.8.

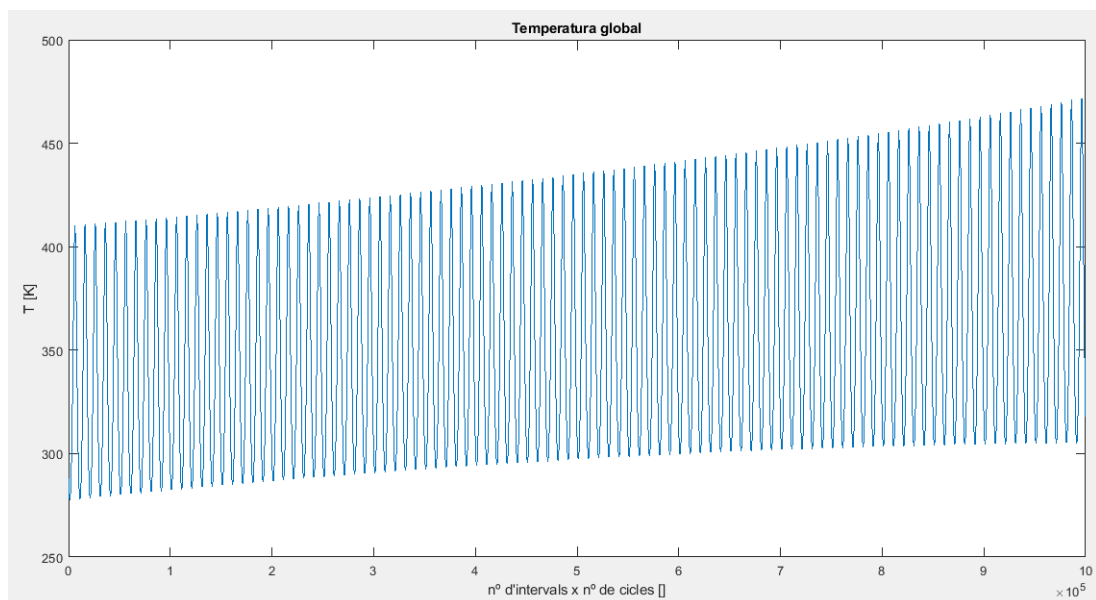


Figura 5.7: evolució de la temperatura global en 100 cicles [Elaboració pròpia]

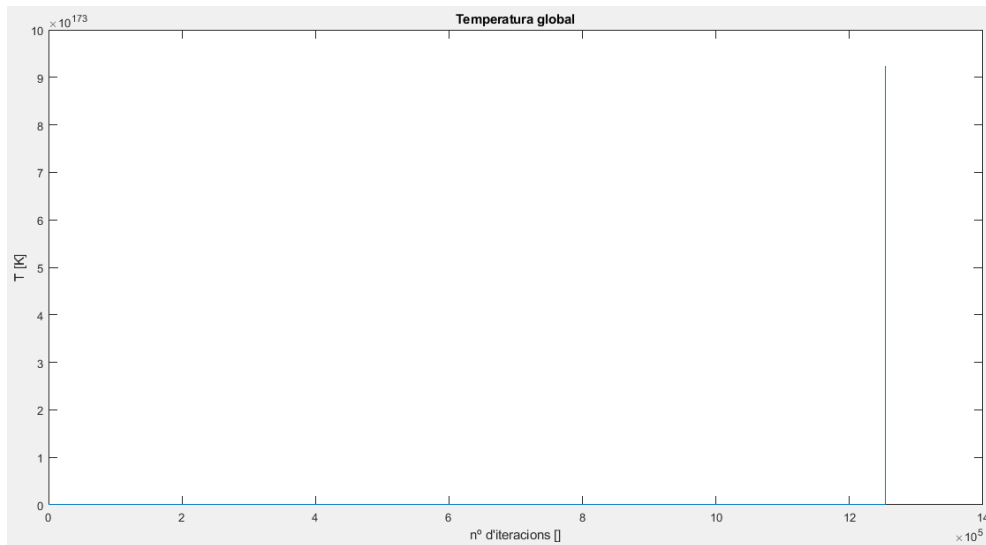


Figura 5.8: evolució de la temperatura global en 140 cicles [Elaboració pròpia]

Per tant, s'assumeix que el codi presenta algun error que es va ampliant amb les simulacions, desestabilitza el sistema i no permet arribar a simular un estat estacionari, és a dir, un cicle desenvolupat independentment de les condicions inicials.

Si s'analitza el cabal màssic en tots els volums del sistema al llarg del temps (figura 5.9):

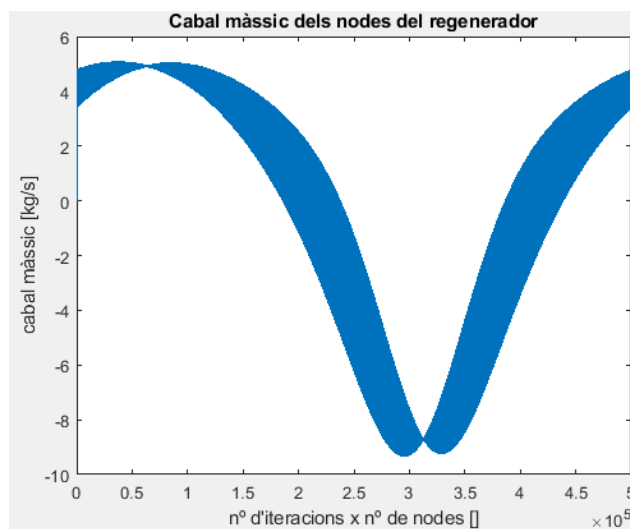


Figura 5.9: evolució del cabal màssic dels nodes del regenerador en un cicle [Elaboració pròpia]

es pot observar una important oscil·lació, que pot ser provocada per la metodologia seguida en el seu càlcul (alternant la direcció de propagació de la informació). Com es pot comprovar,

aquesta metodologia pot provocar que el cabal oscil·li de forma significativa entre iteracions successives, passant d'un valor a un altre sense continuïtat en cada iteració. Això esdevé una potencial causa de problemes.

Un altre aspecte a tenir en compte és el fort desacoblament entre el cabal màssic i l'entalpia. Aquest desacoblament també pot ser una causa d'instabilitat.

Amb aquesta informació, s'inicia un procés de millora del codi per evitar instabilitats i potencial divergència dels resultats numèrics.

5.3. Procés de millora

Després de la identificació de les possibles causes que poden estar portant a la desestabilització final, es defineixen 3 millores del codi. La primera ataca directament a la oscil·lació observada en el cabal màssic. Les altres dues intenten solucionar la deficiència d'acoblament entre el cabal màssic i l'entalpia.

1) Mitjana dels cabals màssics en les dues direccions

La primera estratègia implementada és la menys complexa de les 3. Consisteix en calcular els cabals màssics dels cilindres i de cada node del regenerador de forma doble (una per cada direcció de propagació de la informació). Aquesta informació s'emmagatzema en 2 variables diferents per a, posteriorment, fer-ne la mitjana i treballar amb aquest valor de cabal màssic.

Un cop aplicat, quan s'analitza l'evolució del cabal màssic dels nodes del regenerador (figura 5.10) i es compara amb la figura 5.9, s'observa que amb aquest mètode no varia gairebé gens el comportament dels fluxos màssics al regenerador. Així doncs, es descarta la seva utilitat i es procedeix a provar altres alternatives.

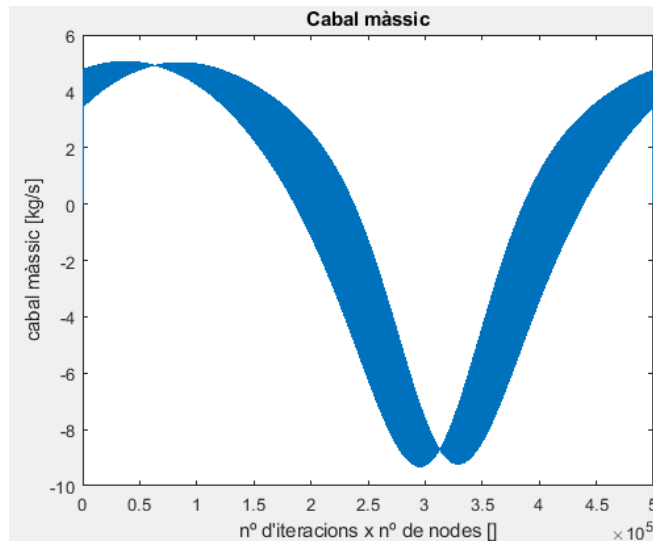


Figura 5.10: evolució del cabal màssic dels nodes del regenerador en un cicle, aplicant la mitjana dels cabals [Elaboració pròpia]

2) Primer mecanisme d'acoblament

Tal com s'ha comentat, aquesta segona solució està orientada al desacoblament existent entre els cabals màssics i les entalpies. El que es planteja és:

- Calcular els cabals màssics de la forma en que es feia inicialment.
- En el cilindre en que es defineix el cabal màssic d'entrada/sortida en funció del balanç màssic al regenerador, es recalcula la seva densitat a partir de l'equació de conservació de la massa.
- Amb la nova densitat i a partir de la pressió global del sistema, es torna a calcular la temperatura i l'entalpia del cilindre

Aquest mètode, tot i permetre realitzar una simulació de 140 cicles sense que el sistema es desestabilitzi, no mostra resultats lògics ja que, tot i entrar potència al sistema, la temperatura global disminueix:

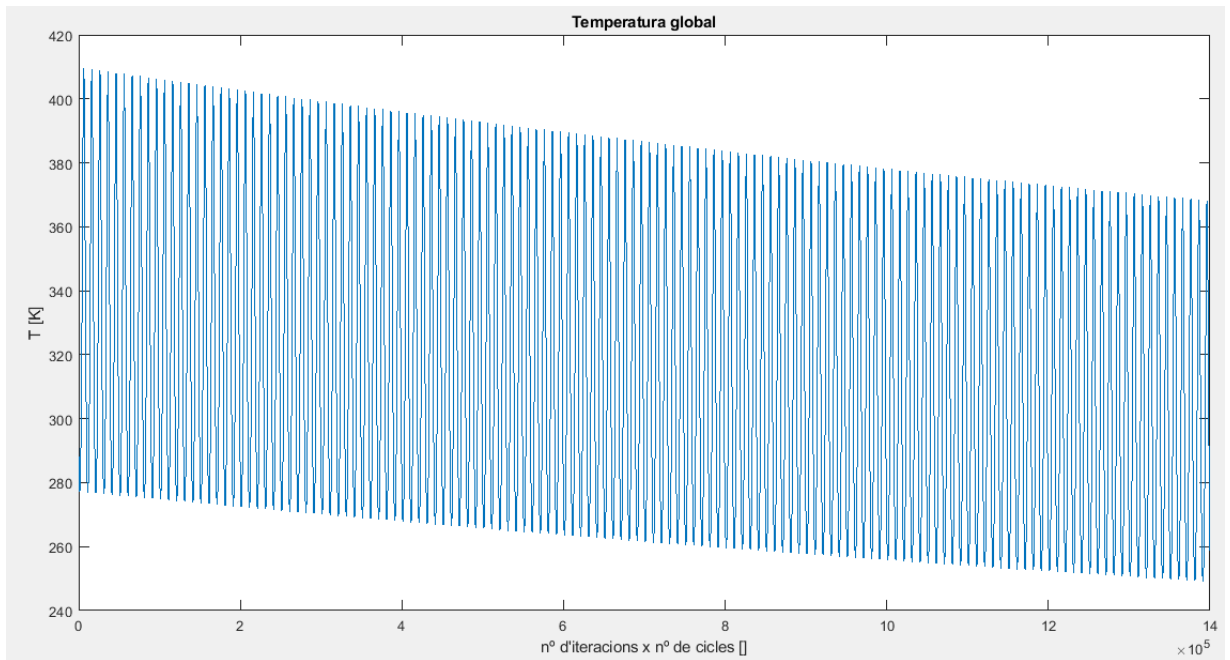


Figura 5.11: evolució de la temperatura global en 140 cicles, utilitzant el mecanisme d'acoblament [Elaboració pròpia]

3) Segon mecanisme d'acoblament

L'última solució que s'ha plantejat, és la més elaborada de les 3. Consisteix en afegir un bucle dins l'algoritme (un bucle intern) que, en una mateixa iteració i a partir de la pressió global ja calculada, calculi implícitament les noves entalpies, temperatures, densitats i cabals màssics. Aquest bucle intern segueix fins que la solució convergeixi o fins que es sobrepassi un número límit d'iteracions. D'aquesta forma, l'algoritme queda:

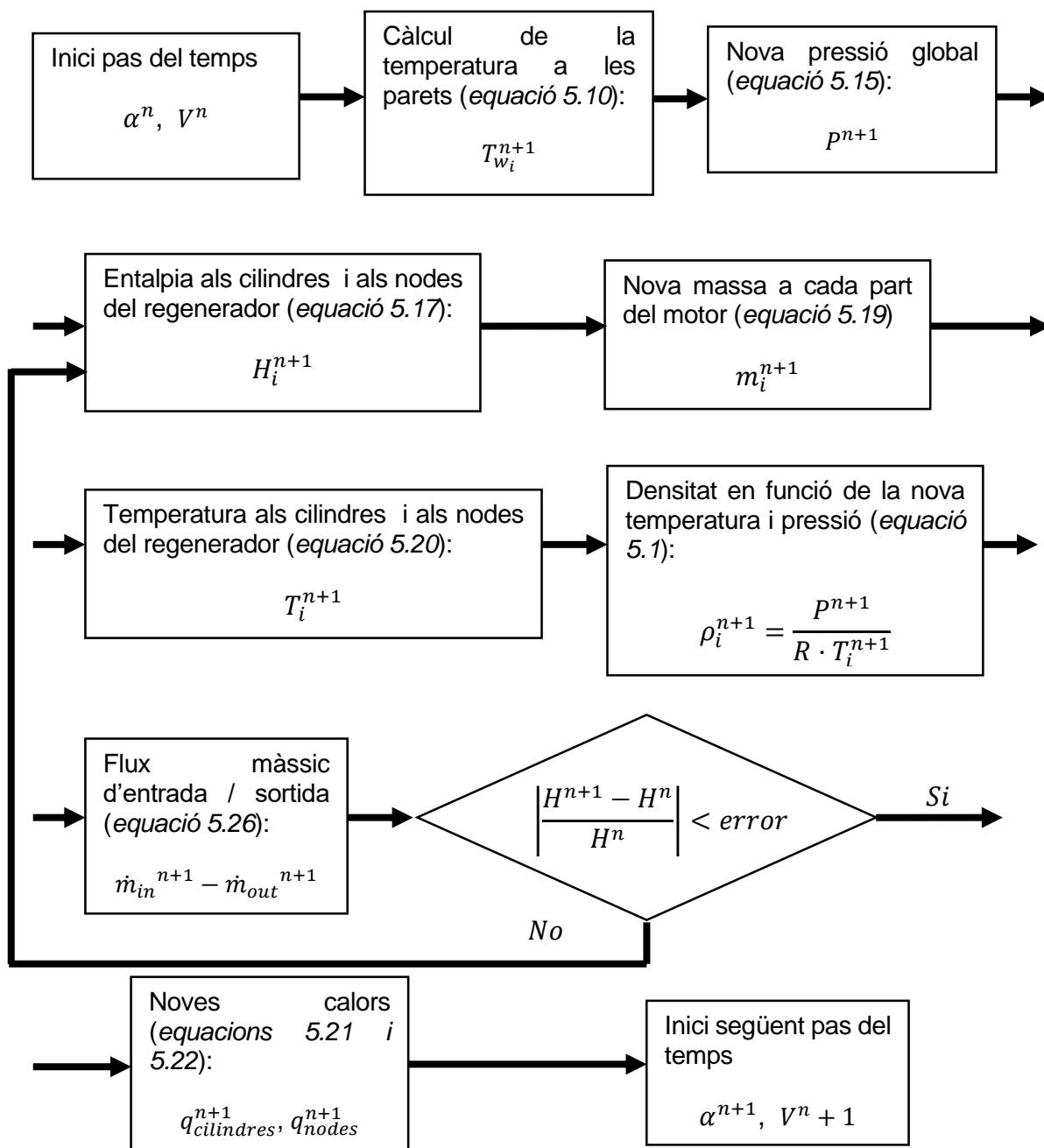


Figura 5.12: esquema de l'algoritme adaptat [Elaboració pròpia]

Malauradament no s'ha aconseguit que el bucle intern convergís. Degut a la limitació temporal que experimenta el present estudi, es procedeix a incloure la verificació d'aquesta tercera millora dins de les accions futures del projecte global d'estudi del cicle Stirling.

El codi amb aquest algoritme s'afegeix als annexos (Annex B), a més del codi principal, de manera que estigui sempre disponible per a la seva revisió.

6. Accions futures

La simulació numèrica de fenòmens físics és un camp molt complex i en el qual sempre es pot concretar, aprofundir o detallar més. Així doncs, un projecte d'aquestes característiques és normalment susceptible de deixar portes obertes al futur, per a seguir amb la recerca feta.

A continuació, es proposa l'estratègia a seguir en les properes fases del projecte global de caracterització i optimització d'un motor Stirling.

6.1. Estabilització del sistema

Com ja s'ha comentat, el codi actual presenta un molt bon funcionament quan no existeix potència tèrmica al sistema. En incloure la potència tèrmica però, les oscil·lacions dels cabals màssics s'incrementen. Els efectes corresponents no s'observen fins transcorregut un nombre elevat de cicles. Finalment, el sistema es desestabilitza.

S'han estudiat 3 estratègies per tal de minimitzar aquestes oscil·lacions sense aconseguir millores. Tot i així, queda pendent la recerca d'errors en l'últim dels mecanismes empleats, el que implica la resolució implícita del balanç de potència en els sistemes oberts.

A banda d'això, també es suggereix buscar noves formes d'intentar estabilitzar el sistema, que fins ara no s'hagin tingut en compte.

6.2. Millores al regenerador

En l'estudi realitzat en aquest treball, una de les hipòtesis considerades és que la temperatura del regenerador és constant. Això ha permès simplificar prou el sistema com per aconseguir una primera versió del codi. Tot i així, el cicle no dona potència ja que no es modela la inèrcia tèrmica del regenerador.

Per tant, és imprescindible que la següent fase del projecte global inclogui la modelització dels fenòmens tèrmics en les parets del regenerador. Per a fer-ho, es proposa seguir la mateixa estratègia que en les parets del cilindre.

6.3. Estudi CFD

Tot i que el codi implementat ha seguit un cert procés de verificació i presenta uns resultats que semblen ajustar-se correctament a la realitat, una bona pràctica seria fer una comprovació d'aquests amb un estudi CFD.

Els estudis CFD (“*Computational fluid Dynamics*”) consisteixen en utilitzar l'anàlisi numèrica per a resoldre problemes associats a fluxos de fluid. Es resolen, per tant, a més de l'equació de l'energia i la de continuïtat, les equacions de quantitat de moviment (Navier-Stokes), les quals no s'han considerat en el present estudi.

El model CFD necessari per a la simulació del motor ha de permetre treballar amb geometries mòbils (els pistons), fluid compressible, acoblaments entre materials i geometries complexes. S'estima que la potència computacional requerida pel model serà elevada.

Es planteja la possibilitat d'iniciar aquesta fase del projecte global estudiant únicament el regenerador imposant les condicions dels cilindres com condicions de contorn del sistema d'estudi. Això permetria reduir la complexitat del sistema i focalitzar els esforços en el regenerador, el seu funcionament i la optimització de la geometria (honeycomb).

7. Pressupost

En el càlcul del cost associat a aquest projecte, s'ha tingut en compte tan sols els costos directes derivats de la seva realització. Aquests implicarien les llicències dels programes que s'han utilitzat per a la realització del treball:

PROGRAMA	PREU LLICÈNCIA
MATLAB (VERSIÓ ESTUDIANTS)	35€ / any
MICROSOFT WORD	99€/any

Taula 7.1: preus de les llicències dels programes utilitzats

A més, no es pot deixar de banda el cost de les hores invertides, sobretot en la implementació i revisió del codi:

HORES	PREU PER HORA	COST TOTAL
350	15€ / hora	5250€

Taula 7.2: preu associat a les hores invertides

Ajuntant els preus d'ambdues taules, s'obté que el preu total de realització del projecte és d'uns 5384€.

Conclusions

Després de tot el procés de recerca, modelització, implementació i verificació del codi que ocupa el projecte, s'han obtingut uns resultats que compleixen de forma satisfactòria amb els objectius que inicialment s'havien plantejat.

S'ha aconseguit un codi que reproduïx correctament l'evolució de la pressió i la temperatura en un cicle, sense aportacions/extraccions de calor. A més, en incorporar aquestes aportacions i, com a mínim en els primers cicles de funcionament, el codi modelitza de forma coherent el funcionament del motor Stirling considerat.

Aquest és un gran pas dins el projecte global de l'estudi del motor Stirling ja que actualment es disposa d'un model simplificat que combina de forma innovadora una anàlisi 0D de pressions amb una anàlisi 1D de temperatures. Tot i així, s'han detectat algunes inestabilitats i es proposen estratègies per intentar compensar-les. També es perfilen els següents passos a realitzar dins el projecte global.

És important remarcar que l'estudi realitzat i el codi definit han ajudat a entendre i assimilar de forma molt clara el comportament del cicle i de les diferents variables implicades.

Com a conclusió extra del treball, cal comentar que durant la seva realització també s'ha obtingut un gran aprenentatge de resiliència i de la quantitat de problemes que poden sorgir en fer un estudi d'aquest tipus. Tot i això, ha quedat patent que sempre hi ha alguna solució, alguna sortida o alguna cosa més a provar, per tal de seguir avançant i millorar els resultats.

Agraïments

En primer lloc, vull agrair la passió i la gran dedicació de la tutora d'aquest treball, l'Elisabet Mas de Les Valls. Sense la seva vitalitat i empenta per a tirar endavant el projecte, hagués estat molt més difícil la seva realització. A més, a nivell tècnic ha estat una gran mentora i ha proporcionat els coneixements necessaris sempre que s'ha necessitat ajuda.

També m'agradaria fer una menció al professor Vicente Medina per la seva aportació de coneixements tècnics sobre l'estudi realitzat.

Finalment, voldria agrair el suport d'amics i família que en algun moment del projecte m'han ajudat o simplement s'han interessat per ell, propiciant que jo mateix m'hi impliqués encara més.

Bibliografia

- Cardona, S., & Clos, D. (2001). *Teoría de máquinas*. Barcelona: Edicions UPC.
- Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT). (s.f.). *Plataforma Solar de Almería*. Obtenido de <https://www.psa.es/es/instalaciones/discos/eurodish.php>
- Corey, J. A. (1984). *U.S.A. Patente nº US4532765A*.
- Foster, P. R. (2011). Innovative Rotary Displacer Stirling Engine: Sustainable Power Generation for Private and Fleet Vehicle Applications. *Journal Of Technology Studies*, 95 - 107.
- García, Á. F. (s.f.). *Física Estadística y Termodinámica*. Obtenido de http://www.sc.ehu.es/sbweb/fisica_/estadistica/estadistica.html
- Gras, P. (07 de 01 de 2009). *Robert Stirling Engine*. Obtenido de <http://www.robertstirlingengine.com/history.php>
- Homutescu, V. M. (2016). Variable displacement alpha-type stirling engine. *IOP Conf. Ser.: Mater. Sci. Eng.* 147 012143, (pág. 7).
- Induc ceramic. (2018). *Honeycomb ceramic regenerator*. Obtenido de <http://www.induc ceramic.com/ceramic-honeycomb/honeycomb-ceramic-regenerator>
- Kamen, D. L., Langenfeld, C. C., Owens, K., & Strimling, J. (1997). *U.S.A. Patente nº US6862883B2*.
- Keveney, M. (23 de 11 de 2011). *Animated Engines*. Obtenido de <http://www.animatedengines.com/vstirling.html>
- Larsen, J. (s.f.). *Stirling Builder*. Obtenido de <http://www.stirlingbuilder.com/new-pop-can-engine>
- National Museums Scotland. (07 de 2016). *Stirling Engine*. Obtenido de <https://www.nms.ac.uk/stirlingengine>
- Toral, R. (20 de 10 de 2015). Obtenido de <https://ifisc.uib-csic.es/users/raul/CURSOS/TERMO/Stirling%20engine.pdf>
- Universidad de los Andes. (25 de 08 de 2010). *Experimentos Demostrativos de Física*. Obtenido de <https://fisicaexpdemostrativos.uniandes.edu.co/MotorStirling.html>

Universidad de Santiago Licenciatura en Educación de Física y Matemática. (2015). *Proyecto motor Stirling: La unión entre lo simple y lo complejo*. Obtenido de <https://proyecto-motorstirling.webnode.cl/fisica-del-motor/ciclo-real/>

Universidad de Sevilla. (20 de 07 de 2010). *Wiki del Departamento de Física Aplicada III: Caso Práctico de Ciclo de Stirling*. Obtenido de http://laplace.us.es/wiki/index.php/Caso_pr%C3%A1ctico_de_ciclo_de_Stirling

Urieli, I. (29 de 01 de 2018). *Stirling Cycle Machine Analysis*. Obtenido de <https://www.ohio.edu/mechanical/stirling/>

Valvias. (2013). *Valvias*. Obtenido de <http://www.valvias.com/index-es.php>

Verschaeve, F., & Schildermans, I. (2009). *U.S.A. Patente nº US8782890B2*.

Wikipedia. (23 de 04 de 2018). *Motor Stirling*. Obtenido de https://es.wikipedia.org/wiki/Motor_Stirling

Annexos

Annex A. Codi de simulació principal

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELITZACIÓ MOTOR STIRLING AMB REGENERADOR %%%%%%%%%%
%% INICI:

% 0) NUM CICLES A RECÓRRER

% Nombre de cicles:
numcic = 100;

% 1) PROPIETATS FLUID MOTOR: AIRE

R = 8.31; %[J/mol K]
Cp = 7/2*R; %[J/mol K]
Cv = Cp - R; %[J/mol K]
M = 28.9; %[grs/mol]
M = M/1000; %[kg/mol]
k = Cp/Cv; %[]
ENT = 0.179; %[J/ mol ·K]
Cp = Cp / M; %[J/ Kg K]
Cv = Cv / M; %[J/ Kg K]
R = R / M; %[J/ Kg K]

% 2) GEOMETRIA CILINDRES

%Cilindres
Vcu1 = 20000; %[cm3]
Vcu1 = Vcu1 / 1000000; %[m3]
Vcu2 = 20000; %[cm3]
Vcu2 = Vcu2 / 1000000; %[m3]
Vcc1 = 2000; %[cm3]
Vcc1 = Vcc1 / 1000000; %[m3]
Vcc2 = 2000; %[cm3]
Vcc2 = Vcc2 / 1000000; %[m3]
Vcc = Vcc1 + Vcc2;
Vcu = Vcu1 + Vcu2;
rel = (Vcc + Vcu) / Vcc; %[]
D = 100; %[cm]
D = D / 100; %[m]
S = pi*D^2/4; %[m2]
CA = Vcu1 / S; %[m]
Per = pi*D; %[m]
RPM = 5000; %[RPM]
RPM = RPM/60*2*pi; %[rad/s]
r = CA / 2; %[m]
l = 1.8 * CA; %[m]
lambda = r / l; %[]
Dout = 300; %[mm]
Dout = Dout / 1000; %[m]
Sout1 = pi*Dout^2/4; %[m2]
Sout2 = Sout1;

```



```

gruixcil = 0.01; %[m]
Vw1 = (pi*(D + 2*gruixcil)^2/4*((Vcc1+Vcu1)/S+gruixcil)) -
(Vcc1+Vcu1);%[m3]
Vw2 = Vw1;
Aw2 = (pi*(D + 2*gruixcil))*((Vcc1+Vcu1)/S+gruixcil); %[m2]

% 3) DEFINICIÓ D'ALPHA, DELTA T I ÍNDEX PROCESSOS

%ALPHA:
alphamin = 0; %[rad]
alphamax = 2*pi; %[rad]
N = 10000; %[]
alpha1 = linspace(alphamin, alphamax, N); %[rad]
alpha2 = alpha1 - pi/2; %[rad]
alpha_grad1 = alpha1*180/pi; %[°]
alpha_grad2 = alpha2*180/pi; %[°]

%DELTA T:
t = alpha1(N)/RPM; %[s]
deltat = t/N; %[s]

% 4) CINEMÀTICA: x, v, a PISTONS I V CILINDRES I TOTAL

x1 = r * (1 - cos(alpha1)) + l * (1 - sqrt(1 - lambda^2*sin(alpha1).^2));
x2 = r * (1 - cos(alpha2)) + l * (1 - sqrt(1 - lambda^2*sin(alpha2).^2));
ve1 = r * (sin(alpha1) + lambda * sin(alpha1).*cos(alpha1) / sqrt(1 -
lambda^2*sin(alpha1).^2)) * RPM;
ac1 = RPM^2 * r * (cos(alpha1) + lambda * cos(2*alpha1));
ve2 = r * (sin(alpha2) + lambda * sin(alpha2).*cos(alpha2) / sqrt(1 -
lambda^2*sin(alpha2).^2)) * RPM;
ac2 = RPM^2 * r * (cos(alpha2) + lambda * cos(2*alpha2));
velve2 = ve1 + ve2;
velve2t = ve1 - ve2;
V_1 = Vcc1 + (CA - x1) * S; %[m^3]
A1_conv = (Vcc1/S)*Per + (CA - x1)*Per; %[m2]
V_2 = Vcc2 + (CA - x2) * S; %[m^3]
A2_conv = (Vcc2/S)*Per + (CA - x2)*Per; %[m2]
Vcilindres = Vcc1 + (CA - x1) * S + Vcc2 + (CA - x2) * S; %[m^3]

%% GEOMETRIA REGENERADOR

%Dr, Lr, deltax

Nn = 50; %nombre nodes
Lr = 0.6 ; %[m]
deltax = Lr / Nn; %[m]
costat = 0.003; %[m]
Numcond = 1600; %nombre de conductes Honeycomb
Sr = costat * costat * Numcond; %[m^2]
Per_reg = 4 * costat * Numcond; %[m]

%Volum Total (cilindres + reg)
V = Vcilindres + Lr * Sr;

```

```

%% CONDICIONS INICIALS FLUID:

Pini = 1; %[atm]
Pini = Pini * 101325; %[Pa]
Vini = V(1); %[m^3]
Tini = 15; %[°]
Tini = Tini + 273.15; %[K]

%Massa
mass = Pini * Vini / (R * Tini); %[Kg]
mass1ini = (Pini * V_1(1) / (R * Tini));
mass2ini = (Pini * V_2(1) / (R * Tini));
mass1 = linspace(mass1ini, mass1ini, N);
mass2 = linspace(mass2ini, mass2ini, N);

%Propietats
roini = (Pini / (R * Tini));
ro1 = linspace(roini, roini, N);
ro2 = linspace(roini, roini, N);

%Entalpia
Hini = Cp * Tini; %[J/kg]

%% CONDICIONS INICIALS I DE CONTORN:

% Condicions de contorn:

qcomb = 800000; %[W] %Calor exterior paret cilindre 1
TE2 = 10 + 273.15; %[K] %T exterior al cilindre 2
hE2 = 50; %[W/K m^2] %Coef conv exterior cilindre 2

% Condicions inicials:

T1 = linspace(Tini, Tini, N); %[K]
H1 = linspace(Hini*mass1ini, Hini*mass1ini, N); %[J]
T2 = linspace(Tini, Tini, N); %[K]
H2 = linspace(Hini*mass2ini, Hini*mass2ini, N); %[J]
Tw1 = linspace(Tini, Tini, N); %[K]
Tw2 = linspace(Tini, Tini, N); %[K]
P = linspace(Pini, Pini, N); %[Pa]
T = linspace(Tini, Tini, N); %[K]

Tmass = linspace(Tini, Tini, N);

%% REGENERADOR:

%H's i T's
massRini = (Pini * Sr * deltax / (R * Tini));
massRi = linspace(massRini, massRini, Nn);
massRiacu = linspace(massRini, massRini, N*Nn);

```

```

massR = linspace(sum(massRi), sum(massRi), N);
roi = linspace(roini, roini, Nn);
roiacu = linspace(roini, roini, N*Nn);
roi_antic = roi;
Hi = linspace(Hini*massRini, Hini*massRini, Nn);
Hiacu = linspace(Hi(1), Hi(1), N*Nn);
Ti = linspace(Tini, Tini, Nn);
Tiacu = linspace(Ti(1), Ti(1), N*Nn);
qdifi = linspace(0, 0, Nn);
Tr = 75; %[°C]
Tr = Tr + 273.15; %[K]

%Lambda Inicial Fluid

lam = (3.807 + 0.074 * Tini) * 10^(-3); %[W/K m]
lambdaR = lam;

%U i Dh

Dh = 0.003; %[m]

U = 50; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROVA U=0
h1 = U;
h2 = U;

%% LOOP:

%Variables totals
Vgen = linspace(Vini, Vini, N*numcic);
Pgen = linspace(Pini, Pini, N*numcic);
Tgen = linspace(Tini, Tini, N*numcic);
Tmassgen = linspace(Tini, Tini, N*numcic);

Hgen =
linspace(Hini*mass1ini+Hini*mass2ini+sum(Hi), Hini*mass1ini+Hini*mass2ini+
sum(Hi), N*numcic);
massagen = linspace(mass, mass, N*numcic);

Wgen = linspace(Pini*(V(1)-V(2)), Pini*(V(1)-V(2)), N+numcic);

dPdtgen = linspace(0, 0, N*numcic);

Hiacugen = linspace(Hi(1), Hi(1), N*Nn*numcic);
Tiacugen = linspace(Tini, Tini, N*Nn*numcic);
mRpuntacugen = linspace(roini*(V_1(1)-V_1(2))/deltat/2, roini*(V_1(1)-
V_1(2))/deltat/2, N*Nn*numcic);

%q's
qE1gen = linspace(0, 0, N*numcic);
qE2gen = linspace(0, 0, N*numcic);
qI1gen = linspace(0, 0, N*numcic);
qI2gen = linspace(0, 0, N*numcic);
qreggen = linspace(0, 0, N*numcic);
qgen = linspace(0, 0, N*numcic);

```

```

aux = linspace(0,0,N);
auxgen = linspace(0,0,N*numcic);

%T's
Twtlgen = linspace(Tini,Tini,N*numcic);

%errors
epsW = linspace(1,1,numcic);
epsT = linspace(1,1,numcic);
epsTini = linspace(1,1,numcic);

%Variables necessàries

%Parets Cilindres
row1 = 785; %[kg/m3]
row2 = 785; %[kg/m3]
Cpw1 = 46; %[J/kg K]
Cpw2 = 46; %[J/kg K]

%Temperatura regenerador
Tfr = linspace(Tini,Tini,N);

%Variació temporal
dVdt = linspace(0,0,N);
dPdt = linspace(0,0,N);

%Calors
q = linspace(0,0,N);
qE1 = linspace(qcomb,qcomb,N);
qE2 = linspace(0,0,N);
qI1 = linspace(0,0,N);
qI2 = linspace(0,0,N);
qreg = linspace(0,0,N);

%Fluxos màssics
mlpuntini = roini*(V_1(1)-V_1(2))/deltat;
mlpuntpar = linspace(mlpuntini,mlpuntini,N);
mlpuntimpar = linspace(mlpuntini,mlpuntini,N);
mlpunt = linspace(mlpuntini,mlpuntini,N);
m2puntini = roini*(V_2(1)-V_2(2))/deltat;
m2puntpar = linspace(m2puntini,m2puntini,N);
m2puntimpar = linspace(m2puntini,m2puntini,N);
m2punt = linspace(m2puntini,m2puntini,N);

mRpuntpar = linspace(mlpuntini/2,mlpuntini/2,Nn);
mRpuntimpar = linspace(mlpuntini/2,mlpuntini/2,Nn);
mRpunt = linspace(mlpuntini/2,mlpuntini/2,Nn);
vR = linspace(mRpunt(1)/roini/Sr,mRpunt(1)/roini/Sr,Nn);
mRpuntacu = linspace(mRpunt(1),mRpunt(1),N*Nn);
mRpunt_nou = linspace(mRpunt(1),mRpunt(1),Nn);

m2punt(1) = -mRpunt(Nn);

```

```

CFL = linspace(0,0,Nn);
CFLacu = linspace(0,0,N*Nn);

%Massa total
massatotal = linspace(mass, mass, N); %[Kg]

%Entalpia total
Htotal =
linspace(Hini*mass1ini+Hini*mass2ini+sum(Hi),Hini*mass1ini+Hini*mass2ini+
sum(Hi),N);

%Trebball
W = linspace(Pini*(V(1)-V(2)),Pini*(V(1)-V(2)),N);

%mpunt factors relax
A1 = 0.5;
A2 = 0.5;
AR = 0.5;

%%Potències a l'inici%%
qI1(1) = h1*A1_conv(1)*(Tw1(1) - T1(1)); %[W]
qI2(1) = h2*A2_conv(1)*(Tw2(1) - T2(1)); %[W]
qE2(1) = hE2*Aw2*(TE2 - Tw2(1)); %[W]

q(1) = (qI1(1) + qI2(1) + qreg(1)); %[W]

% Loop:
for p = 1:numcic
    for i = 1:(N-1)

        %%Temperatures a la paret%%
        Tw1(i+1) = ((qE1(i) - qI1(i)) * deltat/(row1*Cpw1*Vw1)) + Tw1(i);
        Tw2(i+1) = ((qE2(i) - qI2(i)) * deltat/(row2*Cpw2*Vw2)) + Tw2(i);

        %%Variació temporal volum%%
        dVdt(i+1) = (V(i+1) - V(i)) / deltat;

        %%Nova P global%%
        P(i+1) = P(i) + (q(i) - (1 + Cv / R)*dVdt(i+1)*P(i)) * ((R *
deltat) / (Cv * V(i)));
        aux(i+1) = (1 + Cv / R)*dVdt(i+1)*P(i);
        %Variació temporal P
        dPdt(i+1) = (P(i+1) - P(i)) / deltat;

        %%Regenerador%% --> Entalpia, Temperatura, massa, densitat i
mpunt acumulades
        Hiacu((i-1)*Nn+1:i*Nn) = Hi;
        Tiacu((i-1)*Nn+1:i*Nn) = Ti;
        massRiacu((i-1)*Nn+1:i*Nn) = massRi;
        roiacu((i-1)*Nn+1:i*Nn) = roi;
        mRpuntacu((i-1)*Nn+1:i*Nn) = mRpunt;
        CFLacu((i-1)*Nn+1:i*Nn) = CFL;
    end
end

```

```

%%Cilindres%% --> Nova entalpia
    %Cilindre 1
    if m1punt(i) > 0
        H1(i+1) = H1(i) + deltat*(qI1(i)+V_1(i)*dPdt(i+1)-
m1punt(i)*Cp*T1(i));
    else
        H1(i+1) = H1(i) + deltat*(qI1(i)+V_1(i)*dPdt(i+1)-
m1punt(i)*Cp*Ti(1));
    end

    %Cilindre 2
    if m2punt(i) > 0
        H2(i+1) = H2(i) + deltat*(qI2(i)+V_2(i)*dPdt(i+1)-
m2punt(i)*Cp*T2(i));
    else
        H2(i+1) = H2(i) + deltat*(qI2(i)+V_2(i)*dPdt(i+1)-
m2punt(i)*Cp*Ti(Nn));
    end

%%Regenerador%% --> Entalpia de cada node

    %Node 1
    if and(m1punt(i)>0,mRpunt(1)>0)
        Hi(1) = Hi(1) + deltat*((lambdaR*Per_reg*deltax*(Tr - Ti(1))
/ (costat / 2)) + Sr*deltax*dPdt(i+1) + m1punt(i)*Cp*T1(i) -
mRpunt(1)*Cp*Ti(1));
    elseif and(m1punt(i)<0,mRpunt(1)>0)
        Hi(1) = Hi(1) + deltat*((lambdaR*Per_reg*deltax*(Tr - Ti(1))
/ (costat / 2)) + Sr*deltax*dPdt(i+1) + m1punt(i)*Cp*Ti(1) -
mRpunt(1)*Cp*Ti(1));
    elseif and(m1punt(i)<0,mRpunt(1)<0)
        Hi(1) = Hi(1) + deltat*((lambdaR*Per_reg*deltax*(Tr - Ti(1))
/ (costat / 2)) + Sr*deltax*dPdt(i+1) + m1punt(i)*Cp*Ti(1) -
mRpunt(1)*Cp*Ti(2));
    else
        Hi(1) = Hi(1) + deltat*((lambdaR*Per_reg*deltax*(Tr - Ti(1))
/ (costat / 2)) + Sr*deltax*dPdt(i+1) + m1punt(i)*Cp*T1(i) -
mRpunt(1)*Cp*Ti(2));
    end

    %Resta de nodes
    for node = 2:(Nn-1)
        if and(mRpunt(node-1)>0,mRpunt(node)>0)
            Hi(node) = Hi(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)) / (costat / 2))+Sr*deltax*dPdt(i+1)+mRpunt(node-1)*Cp*Ti(node-
1)-mRpunt(node)*Cp*Ti(node));
        elseif and(mRpunt(node-1)<0,mRpunt(node)>0)
            Hi(node) = Hi(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)) / (costat / 2))+Sr*deltax*dPdt(i+1)+mRpunt(node-1)*Cp*Ti(node)-
mRpunt(node)*Cp*Ti(node));
        elseif and(mRpunt(node-1)<0,mRpunt(node)<0)
            Hi(node) = Hi(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)) / (costat / 2))+Sr*deltax*dPdt(i+1)+mRpunt(node-1)*Cp*Ti(node)-
mRpunt(node)*Cp*Ti(node+1));
        end
    end

```

```

else
    Hi (node)= Hi (node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti (node)) / (costat / 2))+Sr*deltax*dPdt (i+1)+mRpunt (node-1)*Cp*Ti (node-
1)-mRpunt (node)*Cp*Ti (node+1));
end
end

%Node n
if and(mRpunt (Nn-1)>0,m2punt (i)>0)
    Hi (Nn) = Hi (Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti (Nn)) / (costat / 2)) + Sr*deltax*dPdt (i+1) + mRpunt (Nn-1)*Cp*Ti (Nn-1)
+ m2punt (i)*Cp*T2 (i));
elseif and(mRpunt (Nn-1)<0,m2punt (i)>0)
    Hi (Nn) = Hi (Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti (Nn)) / (costat / 2)) + Sr*deltax*dPdt (i+1) + mRpunt (Nn-1)*Cp*Ti (Nn) +
m2punt (i)*Cp*T2 (i));
elseif and(mRpunt (Nn-1)<0,m2punt (i)<0)
    Hi (Nn) = Hi (Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti (Nn)) / (costat / 2)) + Sr*deltax*dPdt (i+1) + mRpunt (Nn-1)*Cp*Ti (Nn) +
m2punt (i)*Cp*T2 (i));
else
    Hi (Nn) = Hi (Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti (Nn)) / (costat / 2)) + Sr*deltax*dPdt (i+1) + mRpunt (Nn-1)*Cp*T2 (i-1)
+ m2punt (i)*Cp*T2 (i));
end

%%Regenerador%% --> massa, Temperatura i densitat de cada node

%Node 1
massRi (1) = massRi (1) + deltat*(m1punt (i) - mRpunt (1));
Ti (1) = Hi (1)/(massRi (1)*Cp);
roi_antic (1) = roi (1);
roi (1) = (P (i+1) / (R * Ti (1)));
qdifi (1) = lambdaR*Per_reg*deltax*(Tr - Ti (1)) / (costat /
2); % ho posem a 0 pqe no hi ha conv en prova U=0

for node = 2:(Nn-1)

%Resta de nodes
massRi (node) = massRi (node) + deltat*(mRpunt (node-1) -
mRpunt (node));
Ti (node) = Hi (node)/(massRi (node)*Cp);
roi_antic (node) = roi (node);
roi (node) = (P (i+1) / (R * Ti (node)));
qdifi (node) = lambdaR*Per_reg*deltax*(Tr - Ti (node)) /
(costat / 2);

end

%Node n
massRi (Nn) = massRi (Nn) + deltat*(mRpunt (Nn-1) + m2punt (i));
Ti (Nn) = Hi (Nn)/(massRi (Nn)*Cp);
roi_antic (Nn) = roi (Nn);
roi (Nn) = (P (i+1) / (R * Ti (Nn)));

```

```

2);
    qdifi(Nn) = lambdaR*Per_reg*deltax*(Tr - Ti(Nn)) / (costat /

total
    %%%Regenerador%%% --> massa total, Temperatura global i calor
    massR(i+1) = sum(massRi);
    Tfr(i+1) = mean(Ti);
    qreg(i+1) = mean(qdifi);

    %%%Cilindres%%% --> Nova massa
    mass1(i+1) = mass1(i) - deltat * mlpunt(i);
    mass2(i+1) = mass2(i) - deltat * m2punt(i);

    %%%Cilindres%%% --> Nova Temperatura
    T1(i+1) = H1(i+1)/(mass1(i+1)*Cp);
    T2(i+1) = H2(i+1)/(mass2(i+1)*Cp);

    %%%Cilindres%%% --> Noves propietats
    ro1(i+1) = (P(i+1) / (R * T1(i+1)));
    ro2(i+1) = (P(i+1) / (R * T2(i+1)));

    %%%Nous cabals màssics%%%

    %RELAX Cabals màssics
    a1 = 0; %factor relax cill
    aR = 0; %factor relax reg
    a2 = 0; %factor relax cil2

    %i parella
    if i/2 == floor(i/2)
        %m1 i mR1
        mlpunt(i+1) = (1-a1)*(ro1(i)*V_1(i) -
ro1(i+1)*V_1(i+1))/deltat + a1*mlpunt(i);
        mRpunt(1) = (1-aR)*(mlpunt(i+1) - (roi(1)-
roi_antic(1))*Sr*deltax/deltat)+aR*mRpunt(1);
        vR(1) = mRpunt(1)/(roi_antic(1)*Sr);
        CFL(1) = vR(1)*deltat/deltax;
        %mRnodes
        for node = 2:(Nn-1)
            mRpunt(node) = (1-aR)*(mRpunt(node-1) - (roi(node)-
roi_antic(node))*Sr*deltax/deltat)+aR*mRpunt(node);
            vR(node) = mRpunt(node)/(roi_antic(node)*Sr);
            CFL(node) = vR(node)*deltat/deltax;
        end
        %mRn i m2
        mRpunt(Nn) = (1-aR)*(mRpunt(Nn-1) - (roi(Nn)-
roi_antic(Nn))*Sr*deltax/deltat)+aR*mRpunt(Nn);
        vR(Nn) = mRpunt(Nn)/(roi_antic(Nn)*Sr);
        CFL(Nn) = vR(Nn)*deltat/deltax;
        m2punt(i+1) = (1-a2)*(-mRpunt(Nn))+a2*m2punt(i); %-
(ro2(i+1)*V_2(i+1) - ro2(i)*V_2(i))/deltat;

    %i imparella
    else
        %mRn i m2

```



```

        m2punt(i+1) = (1-a2)*(ro2(i)*V_2(i) -
ro2(i+1)*V_2(i+1))/deltat + a2*m2punt(i);
        mRpunt(Nn) = (1-aR)*(-m2punt(i+1))+aR*mRpunt(Nn); %-
(ro2(Nn)-roi_antic(Nn))*Sr*deltax/deltat;
        vR(Nn) = mRpunt(Nn)/(roi_antic(Nn)*Sr);
        CFL(Nn) = vR(Nn)*deltat/deltax;
        %mRnodes
        for node = (Nn-1):-1:2
            mRpunt(node) = (1-aR)*(mRpunt(node+1) + (roi(node+1)-
roi_antic(node+1))*Sr*deltax/deltat)+aR*mRpunt(node);
            vR(node) = mRpunt(node)/(roi_antic(node)*Sr);
            CFL(node) = vR(node)*deltat/deltax;
        end
        %m1 i mR1
        mRpunt(1) = (1-aR)*(mRpunt(2) + (roi(2)-
roi_antic(2))*Sr*deltax/deltat)+aR*mRpunt(1);
        vR(1) = mRpunt(1)/(roi_antic(1)*Sr);
        CFL(1) = vR(1)*deltat/deltax;
        m1punt(i+1) = (1-a1)*(mRpunt(1) + (roi(1)-
roi_antic(1))*Sr*deltax/deltat)+a1*m1punt(i);

    end

    %%%Unificar par impar%%

    m1punt(i+1) = (1-A1)*m1puntpar(i+1) + A1*m1puntimpar(i+1);
    mRpunt(1) = (1-AR)*mRpuntpar(1)+AR*mRpuntimpar(1);
    %for node = 2:(Nn-1)
    %    mRpunt(node) = (1-AR)*mRpuntpar(node) +
AR*mRpuntimpar(node);
    %end
    mRpunt(Nn) = (1-AR)*mRpuntpar(Nn) + AR*mRpuntimpar(Nn);
    m2punt(i+1) = (1-A2)*m2puntpar(i+1) + A2*m2puntimpar(i+1);

    %%%Recalculer lambdaR%%
    lambdaR = (3.807 + 0.074 * Tfr(i+1)) * 10^(-3); %[W/K m]

    %%%Nova T global%%
    T(i+1) = (V_1(i+1) * T1(i+1) + Sr * Lr * Tfr(i+1) + V_2(i+1) *
T2(i+1)) / (V(i+1));
    Tmass(i+1) = (mass1(i+1)*T1(i+1) + massR(i+1)*Tfr(i+1) +
mass2(i+1)*T2(i+1))/(mass1(i+1) + mass2(i+1) + massR(i+1));

    %%%Noves Potències%%
    qI1(i+1) = h1*A1_conv(i+1)*(Tw1(i+1) - T1(i+1)); %[W]
    qI2(i+1) = h2*A2_conv(i+1)*(Tw2(i+1) - T2(i+1)); %[W]
    qE2(i+1) = hE2*Aw2*(TE2 - Tw2(i+1)); %[W]

    q(i+1) = (qI1(i+1) + qI2(i+1) + qreg(i+1)); %[W]

    %%%Massa total%%
    massatotal(i+1) = mass1(i+1) + mass2(i+1) + massR(i+1);

    %%%Entalpia total%%
    Htotal(i+1) = H1(i+1) + H2(i+1) + sum(Hi);

```

```

%%%Trebball%%%
W(i+1) = P(i+1)*dVdt(i+1)*deltat;

%if i==9
%   break
%end

end

%Variables totals
Vgen((p-1)*N+1:p*N) = V;
Pgen((p-1)*N+1:p*N) = P;
Tgen((p-1)*N+1:p*N) = T;
Tmassgen((p-1)*N+1:p*N) = Tmass;

Hgen((p-1)*N+1:p*N) = Htotal;
massagen((p-1)*N+1:p*N) = massatotal;

Wgen((p-1)*N+1:p*N) = W;

dPdtgen((p-1)*N+1:p*N) = dPdt;

Hiacugen((p-1)*N*Nn+1:p*N*Nn) = Hiacu;
Tiacugen((p-1)*N*Nn+1:p*N*Nn) = Tiacu;
mRpuntacugen((p-1)*N*Nn+1:p*N*Nn) = mRpuntacu;

qE1gen((p-1)*N+1:p*N) = qE1;
qE2gen((p-1)*N+1:p*N) = qE2;
qI1gen((p-1)*N+1:p*N) = qI1;
qI2gen((p-1)*N+1:p*N) = qI2;
qreggen((p-1)*N+1:p*N) = qreg;
qgen((p-1)*N+1:p*N) = q;
auxgen((p-1)*N+1:p*N) = aux;

Tw1gen((p-1)*N+1:p*N) = Tw1;

%Error
if p>1
    epsW(p) = (sum(Wgen((p-2)*N+1:(p-1)*N)) - sum(Wgen((p-1)*N+1:p*N)))/sum(Wgen((p-1)*N+1:p*N))*100;
    disp(p)
    disp('epsW')
    disp(epsW(p))

    epsT(p) = (sum(Tgen((p-2)*N+1:(p-1)*N)) - sum(Tgen((p-1)*N+1:p*N)))/sum(Tgen((p-1)*N+1:p*N))*100;
    disp('epsT')
    disp(epsT(p))

    epsTini(p) = (sum(Tgen(1:N)) - sum(Tgen((p-1)*N+1:p*N)))/sum(Tgen(1:N))*100;
    disp('epsTini')
    disp(epsTini(p))

```

```
end

%Reassignació de primers valors
if p < numcic
    V(1) = V(N);
    P(1) = P(N);
    T(1) = T(N);
    Tmass(1) = Tmass(N);

    H1(1) = H1(N);
    H2(1) = H2(N);
    Htotal(1) = Htotal(N);

    mass1(1) = mass1(N);
    mass2(1) = mass2(N);

    mlpunt(1) = mlpunt(N);
    m2punt(1) = m2punt(N);

    T1(1) = T1(N);
    T2(1) = T2(N);
    Tw1(1) = Tw1(N);
    Tw2(1) = Tw2(N);

    ro1(1) = ro1(N);
    ro2(1) = ro2(N);

    W(1) = W(N);

    dPdt(1) = dPdt(N);

    qI1(1) = qI1(N);
    qI2(1) = qI2(N);
    qreg(1) = qreg(N);
    q(1) = q(N);

end

end
```

Annex B. Codi amb el segon mecanisme d'acoblament

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELITZACIÓ MOTOR STIRLING AMB REGENERADOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%% INICI:
```

```
% 0) NUM CICLES A RECÓRRER
```

```
% Nombre de cicles:
```

```
numcic = 1;
```

```
% 1) PROPIETATS FLUID MOTOR: AIRE
```

```
R = 8.31; %[J/mol K]
```

```
Cp = 7/2*R; %[J/mol K]
```

```
Cv = Cp - R; %[J/mol K]
```

```
M = 28.9; %[grs/mol]
```

```
M = M/1000; %[kg/mol]
```

```
k = Cp/Cv; %[]
```

```
ENT = 0.179; %[J/ mol ·K]
```

```
Cp = Cp / M; %[J/ Kg K]
```

```
Cv = Cv / M; %[J/ Kg K]
```

```
R = R / M; %[J/ Kg K]
```

```
% 2) GEOMETRIA CILINDRES
```

```
%Cilindres
```

```
Vcu1 = 20000; %[cm3]
```

```
Vcu1 = Vcu1 / 1000000; %[m3]
```

```
Vcu2 = 20000; %[cm3]
```

```
Vcu2 = Vcu2 / 1000000; %[m3]
```

```
Vcc1 = 2000; %[cm3]
```

```
Vcc1 = Vcc1 / 1000000; %[m3]
```

```
Vcc2 = 2000; %[cm3]
```

```
Vcc2 = Vcc2 / 1000000; %[m3]
```

```
Vcc = Vcc1 + Vcc2;
```

```
Vcu = Vcu1 + Vcu2;
```

```
rel = (Vcc + Vcu) / Vcc; %[]
```

```
D = 100; %[cm]
```

```
D = D / 100; %[m]
```

```
S = pi*D^2/4; %[m2]
```

```
CA = Vcu1 / S; %[m]
```

```
Per = pi*D; %[m]
```

```
RPM = 5000; %[RPM]
```

```
RPM = RPM/60*2*pi; %[rad/s]
```

```
r = CA / 2; %[m]
```

```
l = 1.8 * CA; %[m]
```

```
lambda = r / l; %[]
```

```
Dout = 300; %[mm]
```

```
Dout = Dout / 1000; %[m]
```

```
Sout1 = pi*Dout^2/4; %[m2]
```

```
Sout2 = Sout1;
```

```
gruixcil = 0.01; %[m]
```

```
Vw1 = (pi*(D + 2*gruixcil)^2/4*((Vcc1+Vcu1)/S+gruixcil)) -  
(Vcc1+Vcu1); %[m3]
```

```
Vw2 = Vw1;
```



```

Aw2 = (pi*(D + 2*gruixcil))*((Vcc1+Vcu1)/S+gruixcil); %[m2]

% 3) DEFINICIÓ D'ALPHA, DELTA T I ÍNDEX PROCESSOS

%ALPHA:
alphamin = 0; %[rad]
alphamax = 2*pi; %[rad]
N = 10000; %[]
alpha1 = linspace(alphamin, alphamax, N); %[rad]
alpha2 = alpha1 - pi/2; %[rad]
alpha_grad1 = alpha1*180/pi; %[]
alpha_grad2 = alpha2*180/pi; %[]

%DELTA T:
t = alpha1(N)/RPM; %[s]
deltat = t/N; %[s]

% 4) CINEMÀTICA: x, v, a PISTONS I V CILINDRES I TOTAL

x1 = r * (1 - cos(alpha1)) + l * (1 - sqrt(1 - lambda^2*sin(alpha1).^2));
x2 = r * (1 - cos(alpha2)) + l * (1 - sqrt(1 - lambda^2*sin(alpha2).^2));
ve1 = r * (sin(alpha1) + lambda * sin(alpha1).*cos(alpha1) / sqrt(1 -
lambda^2*sin(alpha1).^2)) * RPM;
ac1 = RPM^2 * r * (cos(alpha1) + lambda * cos(2*alpha1));
ve2 = r * (sin(alpha2) + lambda * sin(alpha2).*cos(alpha2) / sqrt(1 -
lambda^2*sin(alpha2).^2)) * RPM;
ac2 = RPM^2 * r * (cos(alpha2) + lambda * cos(2*alpha2));
velve2 = ve1 + ve2;
velve2t = ve1 - ve2;
V_1 = Vcc1 + (CA - x1) * S; %[m^3]
A1_conv = (Vcc1/S)*Per + (CA - x1)*Per; %[m2]
V_2 = Vcc2 + (CA - x2) * S; %[m^3]
A2_conv = (Vcc2/S)*Per + (CA - x2)*Per; %[m2]
Vcilindres = Vcc1 + (CA - x1) * S + Vcc2 + (CA - x2) * S; %[m^3]

%% GEOMETRIA REGENERADOR

%Dr, Lr, deltax

Nn = 50; %nombre nodes
Lr = 0.6 ; %[m]
deltax = Lr / Nn; %[m]
costat = 0.003; %[m]
Numcond = 1600; %nombre de conductes Honeycomb
Sr = costat * costat * Numcond; %[m^2]
Per_reg = 4 * costat * Numcond; %[m]

%Volum Total (cilindres + reg)
V = Vcilindres + Lr * Sr;

%% CONDICIONS INICIALS FLUID:

Pini = 1; %[atm]

```

```

Pini = Pini * 101325; %[Pa]
Vini = V(1); %[m^3]
Tini = 15; %[°]
Tini = Tini + 273.15; %[K]

%Massa
mass = Pini * Vini / (R * Tini); %[Kg]
mass1ini = (Pini * V_1(1) / (R * Tini));
mass2ini = (Pini * V_2(1) / (R * Tini));
mass1 = linspace(mass1ini, mass1ini, N);
mass2 = linspace(mass2ini, mass2ini, N);

%Propietats
roini = (Pini / (R * Tini));
ro1 = linspace(roini, roini, N);
ro2 = linspace(roini, roini, N);

%Entalpia
Hini = Cp * Tini; %[J/kg]

%% CONDICIONS INICIALS I DE CONTORN:

% Condicions de contorn:

qcomb = 800000; %[W] %Calor exterior paret cilindre 1
TE2 = 10 + 273.15; %[K] %T exterior al cilindre 2
hE2 = 50; %[W/K m^2] %Coef conv exterior cilindre 2

% Condicions inicials:

T1 = linspace(Tini, Tini, N); %[K]
H1 = linspace(Hini*mass1ini, Hini*mass1ini, N); %[J]
T2 = linspace(Tini, Tini, N); %[K]
H2 = linspace(Hini*mass2ini, Hini*mass2ini, N); %[J]
Tw1 = linspace(Tini, Tini, N); %[K]
Tw2 = linspace(Tini, Tini, N); %[K]
P = linspace(Pini, Pini, N); %[Pa]
T = linspace(Tini, Tini, N); %[K]

Tmass = linspace(Tini, Tini, N);

%% REGENERADOR:

%H's i T's
massRini = (Pini * Sr * deltax / (R * Tini));
massRi = linspace(massRini, massRini, Nn);
massRiacu = linspace(massRini, massRini, N*Nn);
massR = linspace(sum(massRi), sum(massRi), N);
roi = linspace(roini, roini, Nn);
roiacu = linspace(roini, roini, N*Nn);
roi_antic = roi;
Hi = linspace(Hini*massRini, Hini*massRini, Nn);

```

```

Hiacu = linspace(Hi(1),Hi(1),N*Nn);
Ti = linspace(Tini, Tini, Nn);
Tiacu = linspace(Ti(1),Ti(1),N*Nn);
qdifi = linspace(0, 0, Nn);
Tr = 50; %[°C]
Tr = Tr + 273.15; %[K]

%Lambda Inicial Fluid

lam = (3.807 + 0.074 * Tini) * 10^(-3); %[W/K m]
lambdaR = lam;

%U i P

Dh = 0.003; %[m]

U = 50; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROVA U=0
h1 = U;
h2 = U;

%% LOOP:

%Variables totals
Vgen = linspace(Vini,Vini,N*numcic);
Pgen = linspace(Pini,Pini,N*numcic);
Tgen = linspace(Tini,Tini,N*numcic);

Hgen =
linspace(Hini*mass1ini+Hini*mass2ini+sum(Hi),Hini*mass1ini+Hini*mass2ini+
sum(Hi),N*numcic);
massagen = linspace(mass,mass,N*numcic);

Wgen = linspace(Pini*(V(1)-V(2)),Pini*(V(1)-V(2)),N+numcic);

dPdtgen = linspace(0,0,N*numcic);

Hiacugen = linspace(Hi(1),Hi(1),N*Nn*numcic);
Tiacugen = linspace(Tini,Tini,N*Nn*numcic);
mRpuntacugen = linspace(roini*(V_1(1)-V_1(2))/deltat/2,roini*(V_1(1)-
V_1(2))/deltat/2,N*Nn*numcic);

%q's
qE1gen = linspace(0,0,N*numcic);
qE2gen = linspace(0,0,N*numcic);
qI1gen = linspace(0,0,N*numcic);
qI2gen = linspace(0,0,N*numcic);
qgen = linspace(0,0,N*numcic);
aux = linspace(0,0,N);
auxgen = linspace(0,0,N*numcic);

%errors

```

```
epsW = linspace(1,1,numcic);
epsT = linspace(1,1,numcic);
epsTini = linspace(1,1,numcic);
```

%Variables necessàries

%Parets Cilindres

```
row1 = 7850; %[kg/m3]
row2 = 7850; %[kg/m3]
Cpw1 = 460; %[J/kg K]
Cpw2 = 460; %[J/kg K]
```

%Temperatura regenerador

```
Tfr = linspace(Tini,Tini,N);
```

%Variació temporal

```
dVdt = linspace(0,0,N);
dPdt = linspace(0,0,N);
```

%Calors

```
q = linspace(0,0,N);
qE1 = linspace(qcomb,qcomb,N);
qE2 = linspace(0,0,N);
qI1 = linspace(0,0,N);
qI2 = linspace(0,0,N);
qreg = linspace(0,0,N);
```

%Fluxos màssics

```
m1puntini = roini*(V_1(1)-V_1(2))/deltat;
m1puntpar = linspace(m1puntini,m1puntini,N);
m1puntimpar = linspace(m1puntini,m1puntini,N);
m1punt = linspace(m1puntini,m1puntini,N);
m2puntini = roini*(V_2(1)-V_2(2))/deltat;
m2puntpar = linspace(m2puntini,m2puntini,N);
m2puntimpar = linspace(m2puntini,m2puntini,N);
m2punt = linspace(m2puntini,m2puntini,N);
```

```
mRpuntpar = linspace(m1puntini/2,m1puntini/2,Nn);
mRpuntimpar = linspace(m1puntini/2,m1puntini/2,Nn);
mRpunt = linspace(m1puntini/2,m1puntini/2,Nn);
vR = linspace(mRpunt(1)/roini/Sr,mRpunt(1)/roini/Sr,Nn);
mRpuntacu = linspace(mRpunt(1),mRpunt(1),N*Nn);
mRpunt_nou = linspace(mRpunt(1),mRpunt(1),Nn);
```

```
m2punt(1) = -mRpunt(Nn);
```

```
CFL = linspace(0,0,Nn);
CFLacu = linspace(0,0,N*Nn);
```

%Massa total

```
massatotal = linspace(mass, mass, N); %[Kg]
```

%Entalpia total




```

Htotal =
linspace(Hini*mass1ini+Hini*mass2ini+sum(Hi),Hini*mass1ini+Hini*mass2ini+
sum(Hi),N);

%Trebball
W = linspace(Pini*(V(1)-V(2)),Pini*(V(1)-V(2)),N);

%Acoblament 2
iterMAX = 15;
error = 1;
epsilon = 0.0005;

% Loop:
for p = 1:numcic
    for i = 1:(N-1)

        %%%Potències a l'inici%%
        qI1(i) = h1*A1_conv(i)*(Tw1(i) - T1(i)); % [W]
        qI2(i) = h2*A2_conv(i)*(Tw2(i) - T2(i)); % [W]
        qE2(i) = hE2*Aw2*(TE2 - Tw2(i)); % [W]

        q(i) = (qI1(i) + qI2(i) + qreg(i)); % [J]

        %%%Temperatures a la paret%%
        Tw1(i+1) = ((qE1(i) - qI1(i)) * deltat/(row1*Cpw1*Vw1)) + Tw1(i);
        Tw2(i+1) = ((qE2(i) - qI2(i)) * deltat/(row2*Cpw2*Vw2)) + Tw2(i);

        %%%Variació temporal volum%%
        dVdt(i+1) = (V(i+1) - V(i)) / deltat;

        %%%Nova P global%%
        P(i+1) = P(i) + (q(i) - (1 + Cv / R)*dVdt(i+1)*P(i)) * ((R *
deltat) / (Cv * V(i)));
        aux(i+1) = (1 + Cv / R)*dVdt(i+1)*P(i);
        %Variació temporal P
        dPdt(i+1) = (P(i+1) - P(i)) / deltat;

        %%%Regenerador%% --> Entalpia, Temperatura, massa, densitat i
mpunt acumulades
        Hiacu((i-1)*Nn+1:i*Nn) = Hi;
        Tiacu((i-1)*Nn+1:i*Nn) = Ti;
        massRiacu((i-1)*Nn+1:i*Nn) = massRi;
        roiacu((i-1)*Nn+1:i*Nn) = roi;
        mRpuntacu((i-1)*Nn+1:i*Nn) = mRpunt;
        CFLacu((i-1)*Nn+1:i*Nn) = CFL;

        %%%Cilindres%% --> Nova entalpia
        %Cilindre 1
        if mlpunt(i) > 0
            H1(i+1) = H1(i) + deltat*(qI1(i)+V_1(i)*dPdt(i+1)-
mlpunt(i)*Cp*T1(i));
        else

```

```

        H1(i+1) = H1(i) + deltat*(qI1(i)+V_1(i)*dPdt(i+1)-
m1punt(i)*Cp*Ti(1));
        end

        %Cilindre 2
        if m2punt(i) > 0
            H2(i+1) = H2(i) + deltat*(qI2(i)+V_2(i)*dPdt(i+1)-
m2punt(i)*Cp*T2(i));
        else
            H2(i+1) = H2(i) + deltat*(qI2(i)+V_2(i)*dPdt(i+1)-
m2punt(i)*Cp*Ti(Nn));
        end

        %%%Regenerador%% --> Entalpia de cada node

        %Node 1
        if and(m1punt(i)>0,mRpunt(1)>0)
            Hi(1) = Hi(1) + deltat*((lambdaR*Per_reg*deltax*(Tr - Ti(1)))
+ Sr*deltax*dPdt(i+1) + m1punt(i)*Cp*T1(i) - mRpunt(1)*Cp*Ti(1));
        elseif and(m1punt(i)<0,mRpunt(1)>0)
            Hi(1) = Hi(1) + deltat*((lambdaR*Per_reg*deltax*(Tr - Ti(1)))
+ Sr*deltax*dPdt(i+1) + m1punt(i)*Cp*Ti(1) - mRpunt(1)*Cp*Ti(1));
        elseif and(m1punt(i)<0,mRpunt(1)<0)
            Hi(1) = Hi(1) + deltat*((lambdaR*Per_reg*deltax*(Tr - Ti(1)))
+ Sr*deltax*dPdt(i+1) + m1punt(i)*Cp*Ti(1) - mRpunt(1)*Cp*Ti(2));
        else
            Hi(1) = Hi(1) + deltat*((lambdaR*Per_reg*deltax*(Tr - Ti(1)))
+ Sr*deltax*dPdt(i+1) + m1punt(i)*Cp*T1(i) - mRpunt(1)*Cp*Ti(2));
        end

        %Resta de nodes
        for node = 2:(Nn-1)
            if and(mRpunt(node-1)>0,mRpunt(node)>0)
                Hi(node) = Hi(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)))+Sr*deltax*dPdt(i+1)+mRpunt(node-1)*Cp*Ti(node-1)-
mRpunt(node)*Cp*Ti(node));
            elseif and(mRpunt(node-1)<0,mRpunt(node)>0)
                Hi(node) = Hi(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)))+Sr*deltax*dPdt(i+1)+mRpunt(node-1)*Cp*Ti(node)-
mRpunt(node)*Cp*Ti(node));
            elseif and(mRpunt(node-1)<0,mRpunt(node)<0)
                Hi(node) = Hi(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)))+Sr*deltax*dPdt(i+1)+mRpunt(node-1)*Cp*Ti(node)-
mRpunt(node)*Cp*Ti(node+1));
            else
                Hi(node) = Hi(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)))+Sr*deltax*dPdt(i+1)+mRpunt(node-1)*Cp*Ti(node-1)-
mRpunt(node)*Cp*Ti(node+1));
            end
        end

        %Node n
        if and(mRpunt(Nn-1)>0,m2punt(i)>0)
            Hi(Nn) = Hi(Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(Nn)))+ Sr*deltax*dPdt(i+1) + mRpunt(Nn-1)*Cp*Ti(Nn-1) +
m2punt(i)*Cp*T2(i));
        elseif and(mRpunt(Nn-1)<0,m2punt(i)>0)

```

```

        Hi(Nn) = Hi(Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(Nn))) + Sr*deltax*dPdt(i+1) + mRpunt(Nn-1)*Cp*Ti(Nn) +
m2punt(i)*Cp*T2(i));
        elseif and(mRpunt(Nn-1)<0,m2punt(i)<0)
            Hi(Nn) = Hi(Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(Nn))) + Sr*deltax*dPdt(i+1) + mRpunt(Nn-1)*Cp*Ti(Nn) +
m2punt(i)*Cp*Ti(Nn));
        else
            Hi(Nn) = Hi(Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(Nn))) + Sr*deltax*dPdt(i+1) + mRpunt(Nn-1)*Cp*Ti(Nn-1) +
m2punt(i)*Cp*Ti(Nn));
        end

%%%Regenerador%%% --> massa, Temperatura i densitat de cada node

        %Node 1
        massRi(1) = massRi(1) + deltat*(m1punt(i) - mRpunt(1));
        Ti(1) = Hi(1)/(massRi(1)*Cp);
        roi_antic(1) = roi(1);
        roi(1) = (P(i+1) / (R * Ti(1)));
        qdif1(1) = lambdaR*Per_reg*deltax*(Tr - Ti(1)) / (costat /
2); % ho posem a 0 pqe no hi ha conv en prova U=0

        for node = 2:(Nn-1)

            %Resta de nodes
            massRi(node) = massRi(node) + deltat*(mRpunt(node-1) -
mRpunt(node));
            Ti(node) = Hi(node)/(massRi(node)*Cp);
            roi_antic(node) = roi(node);
            roi(node) = (P(i+1) / (R * Ti(node)));
            qdif1(node) = lambdaR*Per_reg*deltax*(Tr - Ti(node)) /
(costat / 2);

        end

        %Node n
        massRi(Nn) = massRi(Nn) + deltat*(mRpunt(Nn-1) + m2punt(i));
        Ti(Nn) = Hi(Nn)/(massRi(Nn)*Cp);
        roi_antic(Nn) = roi(Nn);
        roi(Nn) = (P(i+1) / (R * Ti(Nn)));
        qdif1(Nn) = lambdaR*Per_reg*deltax*(Tr - Ti(Nn)) / (costat /
2);

%%%Cilindres%%% --> Nova massa
mass1(i+1) = mass1(i) - deltat * m1punt(i);
mass2(i+1) = mass2(i) - deltat * m2punt(i);

%%%Cilindres%%% --> Nova Temperatura
T1(i+1) = H1(i+1)/(mass1(i+1)*Cp);
T2(i+1) = H2(i+1)/(mass2(i+1)*Cp);

```

```

%%%Cilindres%%% --> Noves propietats
ro1(i+1) = (P(i+1) / (R * T1(i+1)));
ro2(i+1) = (P(i+1) / (R * T2(i+1)));
%disp('ro1_1')
%disp(ro1(i+1))
%disp('ro2_1')
%disp(ro2(i+1))

%%%Nous cabals màssics%%%

%i parella
if i/2 == floor(i/2)
    %m1 i mR1
    m1punt(i+1) = (ro1(i)*V_1(i) - ro1(i+1)*V_1(i+1))/deltat;
    mRpunt(1) = (m1punt(i+1) - (roi(1)-
roi_antic(1))*Sr*deltax/deltat);
    vR(1) = mRpunt(1)/(roi_antic(1)*Sr);
    CFL(1) = vR(1)*deltat/deltax;
    %mRnodes
    for node = 2:(Nn-1)
        mRpunt(node) = (mRpunt(node-1) - (roi(node)-
roi_antic(node))*Sr*deltax/deltat);
        vR(node) = mRpunt(node)/(roi_antic(node)*Sr);
        CFL(node) = vR(node)*deltat/deltax;
    end
    %mRn i m2
    mRpunt(Nn) = (mRpunt(Nn-1) - (roi(Nn)-
roi_antic(Nn))*Sr*deltax/deltat);
    vR(Nn) = mRpunt(Nn)/(roi_antic(Nn)*Sr);
    CFL(Nn) = vR(Nn)*deltat/deltax;
    m2punt(i+1) = (-mRpunt(Nn));

%i imparella
else
    %mRn i m2
    m2punt(i+1) = (ro2(i)*V_2(i) - ro2(i+1)*V_2(i+1))/deltat;
    mRpunt(Nn) = (-m2punt(i+1)); %- (roi(Nn)-
roi_antic(Nn))*Sr*deltax/deltat;
    vR(Nn) = mRpunt(Nn)/(roi_antic(Nn)*Sr);
    CFL(Nn) = vR(Nn)*deltat/deltax;
    %mRnodes
    for node = (Nn-1):-1:2
        mRpunt(node) = (mRpunt(node+1) - (roi(node+1)-
roi_antic(node+1))*Sr*deltax/deltat);
        vR(node) = mRpunt(node)/(roi_antic(node)*Sr);
        CFL(node) = vR(node)*deltat/deltax;
    end
    %m1 i mR1
    mRpunt(1) = (mRpunt(2) - (roi(2)-
roi_antic(2))*Sr*deltax/deltat);
    vR(1) = mRpunt(1)/(roi_antic(1)*Sr);
    CFL(1) = vR(1)*deltat/deltax;
    m1punt(i+1) = (mRpunt(1) - (roi(1)-
roi_antic(1))*Sr*deltax/deltat);

```

```

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Acoblament
2%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%Assignacions Previaes%%%
H1_ = H1(i+1);
H1_antic_ = H1_;
mass1_ = mass1(i+1);
m1punt_ = m1punt(i+1);
ro1_ = ro1(i+1);
ro1_antic_ = ro1_;
T1_ = T1(i+1);

H2_ = H2(i+1);
mass2_ = mass2(i+1);
m2punt_ = m2punt(i+1);
ro2_ = ro2(i+1);
ro2_antic_ = ro2_;
T2_ = T2(i+1);

Hi_ = Hi;
massRi_ = massRi;
mRpunt_ = mRpunt;
roi_antic_ = roi_antic;
roi_ = roi;
Ti_ = Ti;

iterations = 0;
error = 1;

while and(error>epsilon, iterations<iterMAX)

    iterations = iterations + 1;

    %%%Cilindres%%% --> Nova entalpia
    %Cilindre 1
    H1_antic_ = H1_;
    if m1punt_ > 0
        H1_ = H1_ + deltat*(qI1(i)+V_1(i)*dPdt(i+1)-
m1punt_*Cp*T1_);
    else
        H1_ = H1_ + deltat*(qI1(i)+V_1(i)*dPdt(i+1)-
m1punt_*Cp*Ti_(1));
    end

    %Cilindre 2
    if m2punt_ > 0
        H2_ = H2_ + deltat*(qI2(i)+V_2(i)*dPdt(i+1)-
m2punt_*Cp*T2_);
    else
        H2_ = H2_ + deltat*(qI2(i)+V_2(i)*dPdt(i+1)-
m2punt_*Cp*Ti_(Nn));
    end

```

```

end

%%%Regenerador%%% --> Entalpia de cada node

%Node 1
if and(mlpunt_>0,mRpunt_(1)>0)
    Hi_(1) = Hi_(1) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(1))) + Sr*deltax*dPdt(i+1) + mlpunt_*Cp*T1_ - mRpunt_(1)*Cp*Ti_(1));
elseif and(mlpunt_<0,mRpunt_(1)>0)
    Hi_(1) = Hi_(1) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(1))) + Sr*deltax*dPdt(i+1) + mlpunt_*Cp*Ti_(1) -
mRpunt_(1)*Cp*Ti_(1));
elseif and(mlpunt_<0,mRpunt_(1)<0)
    Hi_(1) = Hi_(1) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(1))) + Sr*deltax*dPdt(i+1) + mlpunt_*Cp*Ti_(1) -
mRpunt_(1)*Cp*Ti_(2));
else
    Hi_(1) = Hi_(1) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(1))) + Sr*deltax*dPdt(i+1) + mlpunt_*Cp*T1_ - mRpunt_(1)*Cp*Ti_(2));
end

%Resta de nodes
for node = 2:(Nn-1)
    if and(mRpunt_(node-1)>0,mRpunt_(node)>0)
        Hi_(node)=
Hi_(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)))+Sr*deltax*dPdt(i+1)+mRpunt_(node-1)*Cp*Ti_(node-1)-
mRpunt_(node)*Cp*Ti_(node));
elseif and(mRpunt_(node-1)<0,mRpunt_(node)>0)
        Hi_(node)=
Hi_(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)))+Sr*deltax*dPdt(i+1)+mRpunt_(node-1)*Cp*Ti_(node)-
mRpunt_(node)*Cp*Ti_(node));
elseif and(mRpunt_(node-1)<0,mRpunt_(node)<0)
        Hi_(node)=
Hi_(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)))+Sr*deltax*dPdt(i+1)+mRpunt_(node-1)*Cp*Ti_(node)-
mRpunt_(node)*Cp*Ti_(node+1));
else
        Hi_(node)=
Hi_(node)+deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(node)))+Sr*deltax*dPdt(i+1)+mRpunt_(node-1)*Cp*Ti_(node-1)-
mRpunt_(node)*Cp*Ti_(node+1));
    end
end

%Node n
if and(mRpunt_(Nn-1)>0,m2punt_>0)
    Hi_(Nn) = Hi_(Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(Nn))) + Sr*deltax*dPdt(i+1) + mRpunt_(Nn-1)*Cp*Ti_(Nn-1) +
m2punt_*Cp*T2_);
elseif and(mRpunt_(Nn-1)<0,m2punt_(i)>0)
    Hi_(Nn) = Hi_(Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(Nn))) + Sr*deltax*dPdt(i+1) + mRpunt_(Nn-1)*Cp*Ti_(Nn) +
m2punt_*Cp*T2_);
elseif and(mRpunt_(Nn-1)<0,m2punt_<0)

```

```

        Hi_(Nn) = Hi_(Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(Nn))) + Sr*deltax*dPdt(i+1) + mRpunt_(Nn-1)*Cp*Ti_(Nn) +
m2punt_*Cp*Ti_(Nn));
        else
        Hi_(Nn) = Hi_(Nn) + deltat*((lambdaR*Per_reg*deltax*(Tr -
Ti(Nn))) + Sr*deltax*dPdt(i+1) + mRpunt_(Nn-1)*Cp*Ti_(Nn-1) +
m2punt_*Cp*Ti_(Nn));
        end

    %%%Regenerador%%% --> massa, Temperatura i densitat de cada
node

    %Node 1
    massRi_(1) = massRi_(1) + deltat*(m1punt_ - mRpunt_(1));
    Ti_(1) = Hi_(1)/(massRi_(1)*Cp);
    roi_antic_(1) = roi_(1);
    roi_(1) = (P(i+1) / (R * Ti_(1)));
    qdifi(1) = lambdaR*Per_reg*deltax*(Tr - Ti(1)) / (costat
/ 2); % ho posem a 0 pqe no hi ha conv en prova U=0

    for node = 2:(Nn-1)

        %Resta de nodes
        massRi_(node) = massRi_(node) + deltat*(mRpunt_(node-1) -
mRpunt_(node));
        Ti_(node) = Hi_(node)/(massRi_(node)*Cp);
        roi_antic_(node) = roi_(node);
        roi_(node) = (P(i+1) / (R * Ti_(node)));
        qdifi(node) = lambdaR*Per_reg*deltax*(Tr - Ti(node)) /
(costat / 2);

        end

        %Node n
        massRi_(Nn) = massRi_(Nn) + deltat*(mRpunt_(Nn-1) +
m2punt_);
        Ti_(Nn) = Hi_(Nn)/(massRi_(Nn)*Cp);
        roi_antic_(Nn) = roi_(Nn);
        roi_(Nn) = (P(i+1) / (R * Ti_(Nn)));
        qdifi(Nn) = lambdaR*Per_reg*deltax*(Tr - Ti(Nn)) /
(costat / 2);

    %%%Cilindres%%% --> Nova massa
    mass1_ = mass1_ - deltat * m1punt_;
    mass2_ = mass2_ - deltat * m2punt_;

    %%%Cilindres%%% --> Nova Temperatura
    T1_ = H1_/(mass1_*Cp);
    T2_ = H2_/(mass2_*Cp);

    %%%Cilindres%%% --> Noves propietats
    roi_antic_ = roi_;

```

```

ro1_ = (P(i+1) / (R * T1_));
ro2_antic_ = ro2_;
ro2_ = (P(i+1) / (R * T2_));
%disp('ro1_1')
%disp(ro1(i+1))
%disp('ro2_1')
%disp(ro2(i+1))

%%%Nous cabals massics%%%

%i parella
if i/2 == floor(i/2)
    %m1 i mR1
    m1punt_ = (ro1_antic_*V_1(i) - ro1_*V_1(i+1))/deltat;
    mRpunt_(1) = (m1punt_ - (roi_(1)-
roi_antic_(1))*Sr*deltax/deltat);
    %vR(1) = mRpunt(1)/(roi_antic(1)*Sr);
    %CFL(1) = vR(1)*deltat/deltax;
    %mRnodes
    for node = 2:(Nn-1)
        mRpunt_(node) = (mRpunt_(node-1) - (roi_(node)-
roi_antic_(node))*Sr*deltax/deltat);
        %vR(node) = mRpunt(node)/(roi_antic(node)*Sr);
        %CFL(node) = vR(node)*deltat/deltax;
    end
    %mRn i m2
    mRpunt_(Nn) = (mRpunt_(Nn-1) - (roi_(Nn)-
roi_antic_(Nn))*Sr*deltax/deltat);
    %vR(Nn) = mRpunt(Nn)/(roi_antic(Nn)*Sr);
    %CFL(Nn) = vR(Nn)*deltat/deltax;
    m2punt_ = (-mRpunt_(Nn));

%i imparella
else
    %mRn i m2
    m2punt_ = (ro2_antic_*V_2(i) - ro2_*V_2(i+1))/deltat;
    mRpunt_(Nn) = (-m2punt_); %- (roi_(Nn)-
roi_antic_(Nn))*Sr*deltax/deltat;
    %vR(Nn) = mRpunt(Nn)/(roi_antic(Nn)*Sr);
    %CFL(Nn) = vR(Nn)*deltat/deltax;
    %mRnodes
    for node = (Nn-1):-1:2
        mRpunt_(node) = (mRpunt_(node+1) - (roi_(node+1)-
roi_antic_(node+1))*Sr*deltax/deltat);
        %vR(node) = mRpunt(node)/(roi_antic(node)*Sr);
        %CFL(node) = vR(node)*deltat/deltax;
    end
    %m1 i mR1
    mRpunt_(1) = (mRpunt_(2) - (roi_(2)-
roi_antic_(2))*Sr*deltax/deltat);
    %vR(1) = mRpunt(1)/(roi_antic(1)*Sr);
    %CFL(1) = vR(1)*deltat/deltax;
    m1punt_ = (mRpunt_(1) - (roi_(1)-
roi_antic_(1))*Sr*deltax/deltat);

```



```

end

T1_;
H1_;
m1punt_;
T2_;
H2_;
m2punt_;
mRpunt_;

error = abs((H1_ - H1_antic_)/H1_antic_);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Final
Acoblament%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%Assignacions Post%%
H1(i+1) = H1_;
mass1(i+1) = mass1_;
m1punt(i+1) = m1punt_;
ro1(i+1) = ro1_;
ro1_ = ro1_antic_;
T1(i+1) = T1_;

H2(i+1) = H2_;
mass2(i+1) = mass2_;
m2punt(i+1) = m2punt_;
ro2(i+1) = ro2_;
ro2_ = ro2_antic_;
T2(i+1) = T2_;

Hi = Hi_;
massRi = massRi_;
mRpunt = mRpunt_;
roi_antic = roi_antic_;
roi = roi_;
Ti = Ti_;

%%Regenerador%% --> massa total, Temperatura global i calor
total
massR(i+1) = sum(massRi);
Tfr(i+1) = mean(Ti);
qreg(i+1) = mean(qdififi);

%%Recalcular lambdaR%%
lambdaR = (3.807 + 0.074 * Tfr(i+1)) * 10^(-3); %[W/K m]

```

```

    %%%Nova T global%%%
    T(i+1) = (V_1(i+1) * T1(i+1) + Sr * Lr * Tfr(i+1) + V_2(i+1) *
T2(i+1)) / (V(i+1));

    %%%Massa total%%%
    massatotal(i+1) = mass1(i+1) + mass2(i+1) + massR(i+1);

    %%%Entalpia total%%%
    Htotal(i+1) = H1(i+1) + H2(i+1) + sum(Hi);

    %%%Trebball%%%
    W(i+1) = P(i+1)*dVdt(i+1)*deltat;

    %if i==3
    %   break
    %end
end

%%Variables totals
Vgen((p-1)*N+1:p*N) = V;
Pgen((p-1)*N+1:p*N) = P;
Tgen((p-1)*N+1:p*N) = T;

Hgen((p-1)*N+1:p*N) = Htotal;
massagen((p-1)*N+1:p*N) = massatotal;

Wgen((p-1)*N+1:p*N) = W;

dPdtgen((p-1)*N+1:p*N) = dPdt;

Hiacugen((p-1)*N*Nn+1:p*N*Nn) = Hiacu;
Tiacugen((p-1)*N*Nn+1:p*N*Nn) = Tiacu;
mRpuntacugen((p-1)*N*Nn+1:p*N*Nn) = mRpuntacu;

qE1gen((p-1)*N+1:p*N) = qE1;
qE2gen((p-1)*N+1:p*N) = qE2;
qI1gen((p-1)*N+1:p*N) = qI1;
qI2gen((p-1)*N+1:p*N) = qI2;
qgen((p-1)*N+1:p*N) = q;
auxgen((p-1)*N+1:p*N) = aux;

%%Error
if p>1
    epsW(p) = (sum(Wgen((p-2)*N+1:(p-1)*N)) - sum(Wgen((p-
1)*N+1:p*N)))/sum(Wgen((p-1)*N+1:p*N))*100;
    disp(p)
    disp('epsW')
    disp(epsW(p))

    epsT(p) = (sum(Tgen((p-2)*N+1:(p-1)*N)) - sum(Tgen((p-
1)*N+1:p*N)))/sum(Tgen((p-1)*N+1:p*N))*100;
    disp('epsT')
    disp(epsT(p))

```

```
        epsTini(p) = (sum(Tgen(1:N)) - sum(Tgen((p-
1)*N+1:p*N)))/sum(Tgen(1:N))*100;
        disp('epsTini')
        disp(epsTini(p))

    end

    %Reassignació de primers valors
    if p < numcic
        V(1) = V(N);
        P(1) = P(N);
        T(1) = T(N);

        H1(1) = H1(N);
        H2(1) = H2(N);

        mass1(1) = mass1(N);
        mass2(1) = mass2(N);

        mlpunt(1) = mlpunt(N);
        m2punt(1) = m2punt(N);

        T1(1) = T1(N);
        T2(1) = T2(N);
        Tw1(1) = Tw1(N);
        Tw2(1) = Tw2(N);

        ro1(1) = ro1(N);
        ro2(1) = ro2(N);

        W(1) = W(N);

        dPdt(1) = dPdt(N);

    end

end
```