



UNIVERSITA' POLITECNICA DELLE MARCHE
FACOLTA' DI INGEGNERIA DELL'INFORMAZIONE

Corso di Laurea triennale Ingegneria Biomedica

**ABLAZIONE TERMICA DI NEOPLASIE MEDIANTE RF
PULSATA: SIMULAZIONE NUMERICA DELLO SCAMBIO
TERMICO CONDUTTIVO NEL FEGATO UMANO**

**THERMAL ABLATION OF TUMORS BY MEANS OF PULSING
RF: CONDUCTIVE HEAT TRANSFER NUMERICAL
SIMULATION IN HUMAN LIVER**

Relatore:
Prof. D'Alessandro Valerio

Tesi di Laurea di:
Fraticelli Nives

Correlatore:
Ing. Falone Matteo

A.A. 2021 / 2022

Indice

ABLAZIONE TERMICA DI NEOPLASIE MEDIANTE RF PULSATA: SIMULAZIONE NUMERICA DELLO SCAMBIO TERMICO CONDUTTIVO NEL FEGATO UMANO	1
Introduzione.....	3
1. Stato dell'arte.....	4
2. Complementi teorici	7
2.1 Equazioni di governo	7
2.1.1 Equazione generale della conduzione termica	7
2.1.2 Equazione di Pennes.....	11
2.1.3 Danneggiamento cellulare	13
2.1.4 Contributo elettrico.....	14
2.2 Metodo dei volumi finiti	18
2.3 OpenFOAM	19
3. Metodo Risolutivo.....	21
3.1 Grandezze termodinamiche e geometria	23
3.1.1 Discretizzazione Dominio	26
3.1.2 Condizioni al contorno	28
3.2 Processing	36
3.2.1 Risolutore.....	36
3.2.2 System	40
3.3 Post-Processing	42
4. Risultati	47
4.1 Verifiche correttezza procedura	47
4.2 Considerazioni finali	55
Conclusioni.....	56
Sitografia	57
Bibliografia.....	57

Introduzione

Nel presente studio si andranno ad esaminare gli effetti sul tessuto epatico quando quest'ultimo è soggetto all'ablazione termica a radiofrequenza. Viene posta l'attenzione sulla soluzione numerica ai volumi finiti dell'equazione di conduzione termica considerando perfusione sanguigna ed effetto joule. L'obiettivo è quello di definire un protocollo per l'esecuzione dell'ablazione a radiofrequenze. Dopo aver effettuato vari test con procedure differenti è possibile indicare il metodo migliore per effettuare ablazione e radiofrequenza. Il protocollo ottimale viene scelto in seguito al confronto dei risultati in termini di grandezze termodinamiche ottenuti mediante soluzioni numeriche. Un altro obiettivo è quello di implementare un sistema risolutivo utile a prevedere possibili effetti dannosi e limitarli. Gli esperimenti in vivo che utilizzano temperature così elevate possono essere pericolosi e avere effetti gravi per le cellule, in questo modo si possono analizzare effetti termici senza alcun pericolo.

Le motivazioni che hanno portato ad affrontare questo argomento risiedono nella maggiore frequenza di utilizzo dell'ablazione termica in campo tumorale, nella necessità di aumentare la sicurezza del suo impiego e di fungere da aiuto nella ricerca e nello sviluppo.

Lo studio viene effettuato utilizzando molteplici software con svariate funzionalità, si porrà l'attenzione principalmente su OpenFOAM il quale attua l'elaborazione effettiva dei dati. Dopo aver approfondito le nozioni teoriche, si esporranno dettagliatamente tutti gli step necessari per utilizzare OpenFOAM e raggiungere l'obiettivo sopra esposto. Al termine di essi sono presenti dei paragoni con altri studi per confrontare i dati ottenuti e contestualizzare le considerazioni con gli ultimi aggiornamenti del campo.

1. Stato dell'arte

Il trattamento delle neoplasie è una delle procedure cliniche più analizzate degli ultimi decenni a causa della sua complessità e della frequenza con la quale si propone questa malattia. Ogni giorno in Italia si diagnosticano più di 1.000 nuovi casi di cancro. Si stima, infatti, che nel nostro Paese vi siano nel corso dell'anno circa 377.000 nuove diagnosi di tumore, di cui 195.000 fra gli uomini e 182.000 fra le donne. [1]

Vengono proposti diversi approcci per curare o almeno fermare la propagazione delle cellule cancerose come la chirurgia, chemioterapia, immunoterapia e radioterapia, o la termoablazione a radiofrequenza.

La TermoAblazione a Radiofrequenza (RFA) è un trattamento mininvasivo che può essere utilizzato in alternativa alla chirurgia quando questa non risulta una buona opzione per il paziente. La termoablazione prevede l'uso del calore per la distruzione delle cellule tumorali. È una tecnica che viene utilizzata da circa trent'anni, attualmente è utilizzata per alcuni tumori e in alcune condizioni particolari. Il trattamento prevede l'introduzione di un ago-elettrodo nella lesione tumorale e l'erogazione di correnti elettriche a radiofrequenza (RF) (≈ 500 kHz) al fine di riscaldare selettivamente il tumore.

Gli studi inerenti agli effetti della corrente elettrica nei tessuti biologici risalgono a molto tempo fa. Si risale a Luigi Galvani (1737-1798) grazie al quale viene proposta la teoria dell'elettricità animale: dimostra la presenza dell'elettricità nei muscoli della rana. Nel 1791 pubblicando l'opera il *Commentarius* sostiene che le rane, pur decapitate e scuoiate, sono in grado di produrre ampi e duraturi movimenti delle zampe, se vengono collegati alcuni loro nervi e muscoli in cui si è immagazzinata elettricità. Successivamente con D'Arsonval si scoprì che una corrente elettrica alla frequenza di 10 kHz o più produce una sensazione di calore nell'attraversare i tessuti viventi, senza essere accompagnata dalla contrazione muscolare dolorosa che si avverte a più basse frequenze. Fu il 1892 quando si ebbe l'ingresso dell'elettromagnetismo nella Medicina. A partire dall'inizio del XX secolo vennero impiegate correnti a frequenza fino a 3Mhz per il trattamento di patologie dolorose dell'apparato muscolo-scheletrico. Nel 1907 venne coniato il termine *diatermia* da Nagelshmidt cioè il riscaldamento dei tessuti viventi generato dalla conversione di correnti ad alta frequenza in calore. Si hanno numerose scoperte riguardo la diatermia elettromagnetica principalmente in ambito militare finché nel 1930 diviene una metodologia standard in ambito terapeutico. Il vantaggio fondamentale di questa tipologia di riscaldamento rispetto a un riscaldamento che si basa sulla conduzione di calore risiede sul fatto che il riscaldamento generato per contatto nei tessuti biologici decade esponenzialmente fino al valore fisiologico di temperatura in pochi millimetri; invece, con l'utilizzo di correnti a radiofrequenze si riesce a entrare più in profondità. Oggigiorno la cura di malattie mediante l'uso di campi elettromagnetici è molto diffusa: Ricostruzione ossea, Stimolazione nervosa, Diatermia, Ipertermia, Ablazione... Diatermia, ipertermia e

ablazione sfruttano l'incremento di temperatura. Ipertermia e ablazione operano anche alle frequenze delle microonde.

In ambito oncologico la termoablazione non è una tecnica nuova, infatti le prime termoablazioni percutanee, ovvero eseguite "attraversando la pelle" senza far ricorso alla classica chirurgia, risalgono all'ultimo decennio del secolo scorso. I progressi in campo tecnologico e la maggiore comprensione dei meccanismi fisici e biologici alla base della termoablazione hanno reso la terapia sempre più precisa, efficace e sicura.

La RFA può essere utilizzata per il trattamento di noduli benigni della tiroide e per alcuni tumori del fegato, osso, rene e polmone distruggendo le cellule tumorali e preservando il tessuto sano. La maggior parte degli studi indica che l'ablazione ha oltre l'85% di successo nella distruzione di piccoli tumori epatici, e che oltre la metà dei tumori rimossi mediante ablazione con radiofrequenza non si ripresentano. L'ablazione con radiofrequenza di solito ha poche complicazioni e può essere ripetuta se necessario per lo stesso tumore o nel caso insorgano altri tumori.

Lo scopo della RFA è di produrre una zona di coagulazione di successo in termini di tumore completamente circondato da una zona in cui il tessuto è stato irreversibilmente distrutto dal calore. In particolare, con la termoablazione si causano al tumore danni diretti che distruggono l'integrità delle membrane delle cellule e degli organelli in essa presenti, e rallentano o bloccano del tutto la replicazione del DNA. Non mancano però anche i danni indiretti (anche questi voluti): dopo la termoablazione si osservano lesioni a cellule e tessuti che vanno dall'apoptosi (morte cellulare programmata), a danni che interessano i vasi sanguigni che nutrono il tumore. Negli ultimi anni è inoltre emerso un altro punto di forza della termoablazione, che consiste nella sua capacità di causare infiammazione e di stimolare quindi il sistema immunitario a reagire contro le cellule tumorali ancora presenti nell'area. Durante il riscaldamento, la carbonizzazione e la disidratazione del tessuto attorno all'elettrodo produce un aumento dell'impedenza elettrica (fenomeno noto come roll-off), che riduce significativamente la diffusione di energia e limita le dimensioni della zona di coagulazione. La modifica del protocollo di erogazione della potenza e del design degli elettrodi sono i modi principali per superare questa limitazione e ottenere zone di coagulazione più ampie.

La termoablazione nasce come terapia per il tumore del fegato e ancora oggi il fegato è uno degli organi nei quali il trattamento è più utilizzato, sia nel caso di malattia primaria (ovvero originaria del fegato) sia nel caso di metastasi. Altri tumori contro i quali la termoablazione ha un ruolo importante sono quelli del polmone, dei reni e delle ossa, e più recentemente anche quelli di seno, surrene e testa-collo. Ma anche fra questi tipi di tumore non tutti sono adatti a essere trattati con il calore, che viene in genere riservato ai tumori di dimensioni ridotte (in genere non superiori ai 5 cm) o che si trovano in aree non operabili chirurgicamente, oppure a pazienti che, per diverse ragioni, non possono essere sottoposti a chirurgia.

L'approccio clinicamente più utilizzato nella RFA tumorale è l'utilizzo di un elettrodo raffreddato simile ad un ago combinato con un protocollo di pulsazione controllata dall'impedenza. Il protocollo serve per aumentare le dimensioni della zona di coagulazione quindi ottimizzare il trattamento con un minor dispendio energetico, poiché periodi di bassa deposizione di corrente si alternano a una corrente di picco più elevata, che consente al tessuto vicino all'elettrodo di raffreddarsi mentre il tessuto più profondo viene riscaldato. L'alimentazione RF viene disattivata quando l'impedenza supera una soglia (solitamente 20 – 30 Ω superiore al valore iniziale) e viene riaccesa dopo alcuni secondi. Questo protocollo viene seguito fino alla fine del tempo di ablazione preimpostato. [3]

Numerosi studi di modellazione sono stati pubblicati su RFA con elettrodi raffreddati a forma di ago, sebbene raramente forniscano dettagli sufficienti sul protocollo di pulsazione utilizzato.

Inoltre, è anche necessario sottolineare la differenza che sussiste tra modelli basati su impulsi di tensione e quelli basati sugli impulsi di corrente. In questo studio verrà analizzato il caso con voltaggio pulsato. Il nostro obiettivo è quindi quello di provare alcuni modelli di RFA in cui applichiamo diversi protocolli con pulsazione di voltaggio deciso a priori seguendo in linea generale il protocollo a pulsazione di impedenza, implementato negli attuali generatori RF, al fine di studiare la correlazione tra la zona danneggiata del tessuto biologico cambiando leggermente il protocollo.

2. Complementi teorici

2.1 Equazioni di governo

Nel seguente studio bisognerà risolvere le seguenti equazioni differenziali nel dominio prescelto. Ciò che vogliamo studiare si basa sull'analisi della temperatura che dipende dai flussi di potenza termica collegati con trasmissione per conduzione, generazione calore da fonte elettrica, presenza di perfusione sanguigna.

2.1.1 Equazione generale della conduzione termica

Inizialmente è necessario definire il concetto di calore: particolare forma di energia (energia termica) che si trasmette, si trasferisce, si propaga da un punto a un altro di un corpo, da un corpo a un altro, da un sistema a un altro per il solo effetto di una differenza di temperatura. [5]. L'esperienza mostra che due corpi a contatto con diverse temperature interagiscono fra di loro fino a raggiungere la stessa temperatura, cioè vanno verso una situazione di equilibrio termico. Questo processo avviene per mezzo della trasmissione del calore in accordo con i due principi fondamentali della termodinamica. Dal Primo Principio della Termodinamica sappiamo che lungo tutto il processo si ha conservazione dell'energia totale in un sistema termodinamicamente isolato; invece, dal Secondo abbiamo la certezza che il calore fluisce spontaneamente dal corpo più caldo a quello più freddo. Il trasferimento di energia che si verifica per effetto della interazione delle particelle di una sostanza dotate di maggiore energia (punti a temperatura più alta) con quelle adiacenti dotate di minore energia (punti a temperatura più bassa) prende il nome di conduzione termica. Da un punto di vista macroscopico, la conduzione termica si manifesta come scambio di energia termica all'interno di corpi o tra corpi in contatto tra di loro, senza movimento macroscopico di materia. Lo scambio termico è dovuto alla cessione di energia cinetica molecolare (rotazionale e vibrazionale) da zone ad alta temperatura verso zone adiacenti a più bassa temperatura. Notiamo proporzionalità tra flusso termico che si genera e la differenza di temperatura tra i due corpi: il coefficiente di proporzionalità fra gradiente e flusso termico corrisponde alla conducibilità termica che misura l'affinità di un mezzo materiale a trasmettere calore per conduzione termica. Ciò significa che con conducibilità termica intendiamo la potenza termica che si trasmette attraverso uno spessore unitario di materiale per unità di superficie e per una differenza di temperatura unitaria.

Da queste considerazioni ne deriva il postulato di Fourier:

$$\mathbf{q} = -\lambda \nabla T \quad (1)$$

Dove:

- \mathbf{q} vettore flusso termico $[W]$,
- λ conducibilità termica $[\frac{W}{mK}]$,
- ∇T gradiente di temperatura $[\frac{K}{m}]$.

Osserviamo che il segno negativo risiede nel fatto che la direzione del flusso termico sarà discorde al vettore gradiente di temperatura; il gradiente della temperatura corrisponde al vettore con direzione perpendicolare a superfici isoterme e il cui verso è concorde a quello dell'aumento di temperatura.

Un'altra grandezza che occorre definire corrisponde alla capacità termica specifica, cioè l'attitudine di un materiale ad accumulare energia termica. Da questo riusciamo a definire la diffusività termica α che quantifica l'attitudine di un materiale a condurre calore rispetto a quanto è capace di accumularne. È data dal rapporto tra conducibilità termica e capacità termica specifica; quindi, la sua unità di misura è $[\frac{m^2}{s}]$.

Lo studio di un problema di conduzione termica prevede di ottenere il campo di temperatura nel mezzo di un oggetto, note le condizioni iniziali ed al contorno. Una volta noto il campo termico attraverso il postulato di Fourier riusciamo a calcolare il flusso termico conduttivo in qualsiasi punto del mezzo. Da questo deriva la necessità di voler ottenere la distribuzione di temperatura, la quale si ottiene risolvendo l'equazione della conduzione.

L'equazione deriva dall'applicazione del principio di conservazione dell'energia ad un oggetto sottoposto ad un fenomeno di scambio termico conduttivo. Si consideri un volume di controllo di estensione infinitesima centrato su di un punto O generico di un mezzo, espresso in Figura 1. All'interno non vi è alcun trasporto macroscopico di massa e la distribuzione della temperatura è espressa come: $T(x, y, z)$.

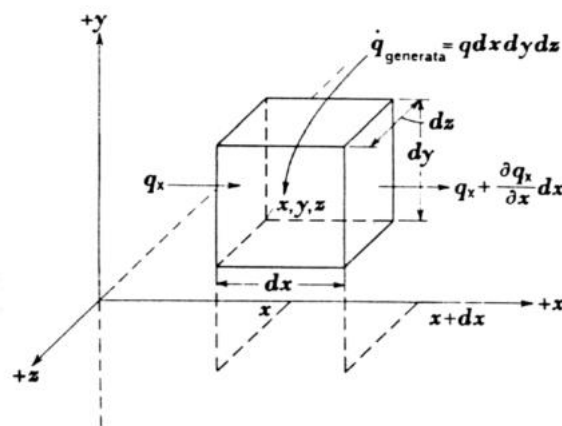


Figura 1: Volume di controllo di riferimento

La forma generale dell'equazione di conservazione dell'energia, limitatamente ai soli flussi termici, fornisce:

$$\dot{E}_{in} + \dot{E}_g - \dot{E}_{out} = \dot{E}_s \quad (2)$$

Dove:

- \dot{E}_{in} = potenza termica in ingresso [W];
- \dot{E}_{out} = potenza termica in uscita [W];
- \dot{E}_s = potenza termica immagazzinata [W];
- \dot{E}_g = potenza termica generata [W];

Il vettore flusso termico, \mathbf{q} è definito dalle sue tre componenti q_x , q_y e q_z sulle superfici individuate dalle coordinate x, y, z ; quindi i flussi termici in entrata. Su quelle poste a $x + dx, y + dy$ e $z + dz$ le componenti di \mathbf{q} sono valutate con uno sviluppo in serie di Taylor (troncato al primo ordine) a partire da quelle note in (x, y, z) ; quindi i flussi termici in uscita.

$$q_{x+dx} = q_x + \frac{\partial q_x}{\partial x} dx \quad (3.1)$$

$$q_{y+dy} = q_y + \frac{\partial q_y}{\partial y} dy \quad (3.2)$$

$$q_{z+dz} = q_z + \frac{\partial q_z}{\partial z} dz \quad (3.3)$$

Allora queste espressioni affermano che i flussi uscenti sono pari al flusso in entrata più la sua variazione lungo la direzione considerata, espressa dalla sua derivata prima. Infatti, se andiamo a fare il bilanciamento lungo le varie direzioni abbiamo:

$$q_x dydz - \left(q_x + \frac{\partial q_x}{\partial x} dx \right) dydz = -\frac{\partial q_x}{\partial x} dx dydz \quad (4.1)$$

$$q_y dx dz - \left(q_y + \frac{\partial q_y}{\partial y} dy \right) dx dz = -\frac{\partial q_y}{\partial y} dx dydz \quad (4.2)$$

$$q_z dx dy - \left(q_z + \frac{\partial q_z}{\partial z} dz \right) dx dy = -\frac{\partial q_z}{\partial z} dx dydz \quad (4.3)$$

Adesso si può considerare il termine di generazione interna $G \left[\frac{W}{m^3} \right]$ che rappresenta la potenza termica per unità di volume generata per fenomeni interni: reazioni elettrochimiche, effetto Joule, reazioni nucleari, perfusione sanguigna... Se ne può calcolare la potenza termica sul volume di controllo di riferimento:

$$\dot{E}_g = G dx dy dz \quad (5)$$

La potenza termica immagazzinata sarà determinata dalla variazione della temperatura nel tempo per il calore specifico c del materiale (se solido $c = c_p$), ovvero pari alla variazione temporale dell'energia interna U della sostanza. Quindi la variazione di energia interna associata al flusso termico conduttivo oggetto di studio si scrive come:

$$\dot{E}_{st} = \frac{\partial}{\partial t} (\rho c_p T dx dy dz) \quad (6)$$

Sostituendo le equazioni (4.*), (5), (6), all'interno del primo principio della termodinamica (2) rispettivamente per ogni direzione, attuando la sommatoria delle tre direzioni, dividendo tutto per $dx dy dz$ otteniamo l'equazione generale della conduzione:

$$-\frac{\partial q_x}{\partial x} - \frac{\partial q_y}{\partial y} - \frac{\partial q_z}{\partial z} + G = \rho c_p \frac{\partial T}{\partial t} \quad (7)$$

La quale può essere scritta in maniera compatta

$$-\nabla \cdot \mathbf{q} + G = \rho c_p \frac{\partial T}{\partial t} \quad (8)$$

In quest'ultima equazione è possibile sostituirci l'equazione del postulato di Fourier ed otteniamo:

$$\nabla \cdot (\lambda \nabla T) + G = \rho c_p \frac{\partial T}{\partial t} \quad (9)$$

Nel caso in cui il mezzo che andiamo a considerare fosse uniforme è possibile riscrivere l'equazione:

$$\nabla^2 T + \frac{G}{\lambda} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (10)$$

Da questa equazione generale della conduzione ne derivano altre tre le quali rappresentano dei casi specifici:

- Equazione di Poisson: conduzione termica stazionaria con generazione interna

$$\nabla^2 T + \frac{G}{\lambda} = 0 \quad (11)$$

- Equazione di Fourier: conduzione termica non-stazionaria in assenza di generazione interna

$$\nabla^2 T = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (12)$$

- Equazione di Laplace: conduzione termica stazionaria in assenza di generazione interna

$$\nabla^2 T = 0 \quad (13)$$

2.1.2 Equazione di Pennes

H.H. Pennes è stato un medico e ricercatore clinico americano che ha introdotto un modello matematico della velocità di produzione di calore da parte dei tessuti umani in relazione al flusso sanguigno locale. L'equazione di Pennes fu il primo modello scientificamente rilevante di scambio termico in tessuti biologici [4]. Fece un'analisi quantitativa dello scambio termico nell'avambraccio umano partendo da dati sperimentali. Nella sua analisi prese come un modello una geometria perfettamente cilindrica nonostante fosse più realistica una sezione ellittica. Un'altra ipotesi che fece la quale va in contrasto con le osservazioni sperimentali riguarda l'assunzione uniforme circa la rapidità del riscaldamento metabolico nel tessuto biologico, anche se la produzione di calore è minore in prossimità della superficie dell'avambraccio dove i gradienti di temperatura sono maggiori. Si trascura la presenza di pelle, strati di massa grassa e le due ossa dell'avambraccio. Si assume che una produzione di calore e conducibilità termica omogenea pari a quelle del tessuto muscolare. Pennes giustificò quest'ultima ipotesi attraverso dati sperimentali che non mostrarono una significativa perturbazione del campo di temperatura a causa della presenza delle ossa. Lo scambio termico fra il sangue ed il tessuto si ritiene sia governato dal Principio di Fick: la massa scambiata per unità di tempo, fra sangue e tessuto, è proporzionale alla differenza tra il livello del sangue e del tessuto moltiplicato per il tasso di flusso sanguigno. Utilizzando questo concetto per descrivere il tasso di trasferimento di calore tra sangue e tessuto, capiamo che il calore netto trasferito dal sangue al tessuto è semplicemente proporzionale alla differenza di temperatura tra il sangue arterioso che entra nel tessuto e il sangue venoso che lascia il tessuto stesso:

$$\dot{Q} = \dot{m}_b c_{pb} (T_a - T_v) \quad (14)$$

Le considerazioni fatte da Pennes si fondono sull'equazione generale della conduzione, la quale è stata precedentemente trattata. In questo caso specifico quindi possiamo definire il termine sorgente nell'equazione generale della conduzione, assumendo verificate le precedenti ipotesi.

Il termine di generazione interna precedentemente definito come $G \left[\frac{W}{m^3} \right]$ che rappresenta la potenza termica per unità di volume generata per fenomeni esterni possiamo così definirla:

$$G = \rho_b \frac{\dot{V}_b}{V} c_{pb} (T_a - T_v) \quad (15)$$

Si ha $\omega = \frac{\dot{V}_b}{V}$ la portata volumetrica di perfusione del sangue nel tessuto per unità di volume dello stesso e c_{pb} il calore specifico a pressione costante del sangue. A questo punto occorre trovare un legame tra temperatura arteriosa, venosa e del tessuto. Ricordiamo che nello studio poniamo l'attenzione sulla temperatura del tessuto. Partendo dal presupposto che la

temperatura del sangue venoso che abbandona il tessuto dipende dal grado di equilibrio termico che possiede con il tessuto ad esso circostante possiamo introdurre il parametro di equilibrio termico, k' :

$$T_v = T_t + k'(T_a - T_v) \quad (16)$$

Per $k' = 0$ si ha completo equilibrio termico cioè temperatura venosa e del tessuto si uguagliano. Invece nel caso in cui $k' = 1$ al contrario la temperatura del sangue venoso che abbandona il tessuto è esattamente T_t (la temperatura del tessuto); il sangue venoso abbandona il tessuto ad una temperatura eguale a quella del sangue arterioso entrante.

Nel modello di Pennes per ottenere come unica variabile T_t sono stati scelti i seguenti parametri:

1. la temperatura arteriosa, T_a , uniforme attraverso l'intero tessuto e pari a T_{a0} che è la temperatura media dell'arteria brachiale;
2. $k' \rightarrow 0$ i.e. $T_v = T_t$

Quindi, grazie a queste due ipotesi possiamo definire il termine di generazione interna, cioè il termine sorgente di perfusione nel modello di Pennes come:

$$G = \rho_b \omega_b c_{pb} (T_{a0} - T_t). \quad (17)$$

Ora abbiamo ricavato l'equazione di Pennes per lo scambio termico in tessuti in biologici:

$$\rho_t c_{pt} \frac{\partial T_t}{\partial t} = \nabla \cdot (\lambda_t \nabla T_t) + \rho_b \omega_b c_{pb} (T_{a0} - T_t) \quad (18)$$

Il termine sorgente di perfusione modella essenzialmente lo scambio termico fra il sangue ed i tessuti biologici che avviene nei più piccoli vasi della microcircolazione dove il sangue entra alla temperatura T_{a0} . La presenza del flusso sanguigno partecipa allo scambio termico agendo come termine sorgente o pozzo in funzione del segno algebrico della differenza di temperatura ($T_{a0} - T_t$): nel caso in cui $T_{a0} > T_t$ funge da sorgente, altrimenti da pozzo. Si ha però trascurato completamente qualsiasi scambio termico fra sangue e tessuto nelle regioni pre e post-capillari. Bisogna notare che alla base del modello di Pennes deve sussistere l'ipotesi per cui il sangue entra nel letto capillare e raggiunge il completo equilibrio termico con il tessuto biologico circostante entrando nella circolazione venosa alla temperatura del tessuto. Possiamo assumere come verificata l'ipotesi di completo equilibrio termico fra tessuto e sangue nel letto capillare poiché i diametri delle sezioni di efflusso sono molto piccoli.

Però, notiamo che nell'equazione generale del modello di Pennes abbiamo alcuni difetti, i quali sono stati causa di critiche. Innanzitutto, nell'equazione generale abbiamo tre temperature incognite, le quali vengono sostituite grazie ad ulteriori ipotesi grossolani. Ci sono tre differenti materiali che occupano lo stesso spazio ad un dato istante di tempo:

tessuto solido ed il flusso sanguigno arterioso e venoso. Sono presenti anche delle inconsistenze dal punto di vista fisico, come ad esempio che lo scambio termico dovrebbe avvenire in tutte le direzioni non solo in quella del gradiente di temperatura locale nel tessuto, e inoltre, lo scambio di calore dovrebbe venire in funzione della differenza di temperatura dei mezzi e non della temperatura arteriosa e venosa. Da queste criticità presenti nel modello di Pennes seguono altri modelli di scambio di calore nei tessuti biologici più raffinati come ad esempio il modello di Wulff. Un'ulteriore critica del modello riguarda l'aver trascurato l'effetto del ritardo fra il flusso termico innescato da un gradiente termico ed il gradiente stesso, da questa considerazione infatti deriva un'estensione del postulato di Fourier dove il ritardo è rappresentato da un tempo di rilassamento. Questo termine nei materiali biologici assume un valore compreso in un range tra $10 - 10^3$ s.

2.1.3 Danneggiamento cellulare

Nell'ipertermia e nei processi criogenici, la necrosi tissutale (danno permanente o morte del tessuto vivente) si verifica quando ci si imbatte in uno dei due seguenti fenomeni:

- viene assorbita un'eccessiva energia termica (processo ipertermico) o rilasciata (processo criogenico),
- è stata superata una temperatura critica alta (processo di ipertermia) o bassa (processo criogenico).

Le molecole e cellule che rimangono attive, in seguito a riscaldamenti significativi, possono essere valutate ad un certo livello di temperature utilizzando la legge di Arrhenius:

$$\log \frac{C(t)}{C_0} = -A \int_0^t \frac{\Delta E}{RT(t')} dt' = -\Omega \quad (19)$$

- C_0 è la concentrazione iniziale di molecole o cellule;
- $C(t)$ è la concentrazione di molecole o cellule attive al tempo t ;
- A è la costante di Arrhenius;
- R è la costante universale dei gas;
- ΔE l'energia di attivazione per il danno irreversibile;
- Ω il grado danneggiamento cellulare.

Quando $\Omega=1$ si ha un grado di danneggiamento pari al 63%, oppure, con $\Omega=4,6$, che comprende la zona con una probabilità di morte cellulare del 99%. In queste condizioni il tessuto è considerato danneggiato in maniera irreversibile. Quindi calcolando il grado di danneggiamento cellulare mediante l'equazione di Arrhenius riusciamo a definire il profilo del danno termico. [13]

2.1.4 Contributo elettrico

La carica elettrica è una proprietà fondamentale della materia rappresentabile come una grandezza scalare la cui unità di misura è il coulomb C. Le cariche elettriche presenti in natura si dividono in due diverse specie, che vengono contraddistinte con i termini di carica positiva e carica negativa. Le cariche di ugual segno si respingono e quelle di segno opposto si attraggono. Possiamo quindi sostenere che tra le cariche elettriche si esercita una forza. Per calcolare il modulo della forza esercitata possiamo far riferimento alla legge di Coulomb, la quale sostiene che la forza elettrostatica che si esercita tra due cariche puntiformi è direttamente proporzionale alle cariche e inversamente proporzionale al quadrato della loro distanza. Dunque, calcolando la seguente forza nel vuoto abbiamo:

$$F = \frac{1}{4\pi\epsilon_0} \frac{|Q_1| \cdot |Q_2|}{d^2} \quad (20)$$

Dove:

- ϵ_0 costante dielettrica del vuoto $\left[\frac{F}{m}\right]$
- d distanza relativa tra le cariche $[m]$
- $|Q_1||Q_2|$ i valori assoluti delle cariche elettriche $[C]$

Il campo elettrico è la regione di spazio in cui agiscono le forze elettriche su altre cariche eventualmente presenti. Una carica elettrica, o una distribuzione di cariche elettriche, genera attorno a sé un campo elettrico, nel senso che modifica le proprietà dello spazio circostante in modo che una qualunque altra carica posta nelle vicinanze viene sollecitata da una forza di natura elettrica. Per analizzare l'esistenza del campo elettrico generato da una carica Q e quantificarlo, si introduce una carica di prova q , di intensità molto più piccola, nella regione di spazio occupata da Q . Naturalmente il campo generato da Q esiste a prescindere dalla presenza della carica di prova q . Il campo elettrico corrisponde alla forza esercitata da Q su q , rispetto al valore di q stesso, ed è indipendente dalla carica di prova. Si ha quindi:

$$\mathbf{E} = \frac{\mathbf{F}}{q} \quad (21)$$

rappresenta la forza che agisce su una carica unitaria; nel Sistema Internazionale si misura in newton/coulomb $\left[\frac{N}{C}\right]$.

Osserviamo che andando a sostituire la forza esercitata tra Q e q calcolata con la legge di Coulomb otteniamo che l'intensità del campo elettrico è data da:

$$E = \frac{1}{4\pi\epsilon_0} \frac{Q}{d^2} \quad (22)$$

Quindi si può evincere l'indipendenza dell'intensità del campo elettrico dalla carica di prova q , e la dipendenza dalla distanza alla quale il campo viene misurato.

Il campo elettrico è un campo conservativo, ciò significa che, il lavoro necessario per portare una carica da un punto a un altro all'interno del campo è indipendente dal percorso scelto e dipende unicamente dai punti iniziale e finale. Perciò, è possibile definire il potenziale elettrico V come il lavoro che occorre compiere per portare una carica unitaria da un punto qualsiasi del campo elettrico ad un punto infinitamente lontano esterno al campo. Questa grandezza varia da punto a punto in un campo: a punti diversi dello spazio corrispondono potenziali diversi. Il lavoro da compiere per portare una carica unitaria q da un punto A (potenziale elettrico V_A) a un punto B (potenziale elettrico V_B), entrambi interni al campo è proporzionale alla carica stessa. Infatti, si deve applicare una forza eguale ed opposta alla forza elettrostatica.

$$\mathbf{F} = -q\mathbf{E} \quad (23)$$

Quindi per la definizione di lavoro necessario per portare la carica dal punto A al punto B sarà dato da:

$$L_{AB} = \int_A^B \mathbf{F} \cdot d\mathbf{l} \quad (24)$$

Da questa considerazione possiamo definire la differenza di potenziale in funzione del lavoro, ovvero, la differenza di potenziale elettrico tra due punti corrisponde al lavoro per portare la carica q dal punto A al punto B diviso per il valore della carica q .

$$V_{AB} = \frac{L_{AB}}{q} \quad (25)$$

Ora, sostituendo l'eq. (23) nella eq. (24), e successivamente nella eq. (25) otteniamo la relazione che lega potenziale elettrico e campo elettrico.

$$V_{AB} = - \int_A^B \mathbf{E} \cdot d\mathbf{l} \quad (24)$$

La quale può essere riscritta in termini di gradiente nel seguente modo:

$$\mathbf{E} = -\nabla V \quad (25)$$

Quindi il vettore campo elettrico è funzione dello scalare V . Il segno meno indica che il campo elettrico va dalla regione a potenziale maggiore verso quella a potenziale minore, mentre il vettore gradiente ∇V è definito in modo che il suo verso coincida con quello delle V crescenti.

Un'altra grandezza che necessariamente bisogna definire riguarda il flusso $\Phi(\mathbf{E})$ del campo elettrico di una superficie. Si prenda in considerazione un volume di controllo Ω la cui superficie corrisponde $\partial\Omega$. Il flusso di campo elettrico su questa superficie corrisponde alla

quantità di E che fluisce attraverso la superficie $\partial\Omega$ del volume di controllo Ω . Si consideri $\partial\Omega$ come una superficie orientata in un campo vettoriale e dS un suo elemento con \mathbf{n} il versore normale corrispondente, dove per ogni elemento si ha il relativo vettore campo elettrico \mathbf{E}_i come in figura 2.

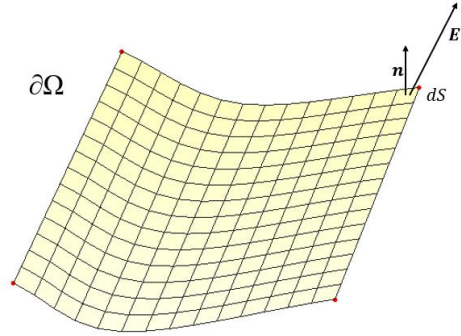


Figura 2: rappresentazione superficie calcolo flusso elettrico

Il flusso $\Phi(E)$ del vettore campo elettrico sulla superficie S sarà calcolato con il seguente integrale di superficie:

$$\Phi(E) = \oint_S (\mathbf{E}_i \cdot \mathbf{n}) dS \quad (26)$$

Uno dei teoremi fondamentali nello studio del campo elettrico è il *teorema di Gauss*, il quale sostiene che il flusso di un campo elettrico attraverso una superficie chiusa è dato dal rapporto tra la carica elettrica totale interna q_t alla superficie e la costante dielettrica assoluta del mezzo ϵ_0 .

$$\Phi(E) = \frac{q_t}{\epsilon_0} \quad (27)$$

Ciò significa che se la carica è esterna alla superficie chiusa il flusso è nullo.

Dato che nella maggior parte dei casi pratici non abbiamo una sola carica ma una distribuzione continua di cariche che occupano un determinato volume, possiamo definire la carica q_t nel modo seguente.

Dato un volume V , una densità di carica per unità di volume pari a $\rho_t(x, y, z)$, la carica totale sarà:

$$q_t = \iiint_V \rho_v(x, y, z) dV \quad (28)$$

Quindi, il *Teorema della divergenza* che ritiene il flusso del campo elettrico E attraverso una superficie chiusa S è pari all'integrale della sua divergenza nel volume contenuto dalla superficie chiusa S , ovvero:

$$\oint_S \mathbf{E} \cdot d\mathbf{S} = \iiint_V \nabla \cdot \mathbf{E} dV \quad (29)$$

A questo punto, alla precedente equazione se applichiamo il teorema di Gauss e la distribuzione di carica otteniamo

$$\varepsilon_0 \iiint_V \nabla \cdot \mathbf{E} dV = \iiint_V \rho_v(x, y, z) dV \quad (30)$$

Quindi, abbiamo ricavato il Teorema di Gauss scritto in forma differenziale:

$$\nabla \cdot \mathbf{E} = \frac{\rho_v}{\varepsilon_0} \quad (31)$$

Avendo approfondito questi concetti, è possibile ora concentrarsi sulla corrente elettrica. La corrente elettrica indica lo spostamento complessivo delle cariche elettriche. Naturalmente una corrente elettrica per fluire necessita di un conduttore elettrico, ossia un mezzo in cui è possibile il movimento delle cariche. Si hanno due grandezze fondamentali: l'intensità di corrente I [A] e la densità di corrente \mathbf{J} $\left[\frac{A}{m^2}\right]$. L'intensità di corrente è la quantità di carica che attraversa una superficie in un certo periodo di tempo ed è uno scalare, mentre la densità di corrente è il vettore che rappresenta la corrente elettrica che attraversa perpendicolarmente una superficie. Il vettore densità di corrente è direttamente proporzionale al campo elettrico mediante un fattore chiamato conduttività elettrica σ .

$$\mathbf{J} = \sigma \mathbf{E} \quad (32)$$

Tanto maggiore è il valore di conduttività elettrica, tanto più facilmente gli elettroni possono muoversi attraverso il conduttore che li ospita. Si può dimostrare che la conducibilità è l'inverso della resistività. Dunque, la conducibilità elettrica è la proprietà fisica di un materiale che esprime la capacità di trasporto della carica elettrica. L'unità di misura SI è il Siemens al metro [S/m].

Attraverso analisi sperimentali vengono definite le due leggi di Ohm:

- La prima legge di Ohm stabilisce che la resistenza elettrica è data dal rapporto tra la differenza di potenziale ai capi del conduttore e l'intensità di corrente;
- la seconda legge di Ohm mette in relazione la resistenza elettrica di un conduttore con la resistività.

In natura un conduttore metallico attraversato dalla corrente elettrica si riscalda. Uno dei fenomeni che accompagnano il passaggio della corrente elettrica in un conduttore è la produzione di calore, che si manifesta con un aumento di temperatura del conduttore stesso, tale fenomeno prende il nome di effetto Joule. In un generico elemento circuitale in cui scorre una corrente I e ai cui capi vi è una differenza di potenziale pari a V il valore della potenza elettrica P [Watt] fornita dal prodotto delle due grandezze precedenti, tale potenza viene trasformata in energia termica. Ricordando la prima legge di Ohm che lega la tensione V alla corrente I attraverso il fattore proporzionale resistenza R è possibile stimare la potenza generata nel seguente modo:

$$P = RI^2 = \frac{V^2}{R} \quad (33)$$

2.2 Metodo dei volumi finiti

Un modello numerico è una simulazione matematica di un processo fisico reale. I metodi di modellazione numerica sono strumenti per rappresentare, con un'adeguata precisione, la geometria di un sistema tenendo conto del suo comportamento ed effettuando i calcoli in tempi ragionevoli. Esistono diversi metodi comunemente utilizzati come ad esempio le Differenze Finite (FD), Volumi Finiti (FV), Elementi Finiti (FEM), Elementi al Contorno (BEM) ecc.

Il metodo dei Volumi Finiti (FV - Finite Volume) fa uso della forma integrale delle equazioni di conservazione. Il dominio viene suddiviso in volumi di controllo (CV - Control Volume), detti anche celle, adiacenti, e le equazioni di conservazione vengono applicate per ciascun volume, e la variabile giace tipicamente al centro della cella. Come illustrato nella seguente figura 3.

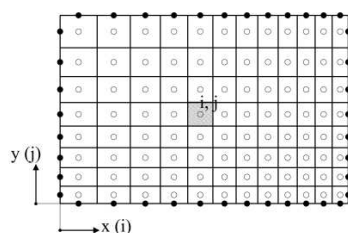


Figura 3: Esempio dominio discretizzato 2D

Una volta ottenuta la suddivisione del dominio nei vari volumi, utilizzando la forma integrale delle equazioni del problema specifico vengono definite le relazioni presenti tra le varie celle adiacenti così da poter essere risolte in modo numerico utilizzando un calcolatore. Il metodo può essere usato inserendo obbligatoriamente le condizioni al contorno. Queste celle hanno dimensione finita e non infinitesima, ciò significa che le soluzioni ottenute derivano dall'approssimazione in una funzione discreta (cioè per la quale vengono forniti i valori solamente in determinati punti) piuttosto che continua. Questi punti prendono il nome di “nodi della griglia di calcolo”. La soluzione in tutti i volumi in cui si è suddiviso il dominio, dopo essere state specificare le condizioni di congruenza sugli elementi adiacenti, costituisce il risultato del modello numerico.

È di fondamentale importanza come la griglia di calcolo venga discretizzata; infatti, occorre attuare la discretizzazione nel modo migliore in base al problema che si pone: ad esempio, raffinando la mesh nei punti di maggior interesse della trattazione.

Il seguente metodo è abbondantemente utilizzato in diversi campi dell'ingegneria, come la meccanica dei fluidi, lo studio del trasferimento di massa e calore attraverso un mezzo, come nel nostro caso.

2.3 OpenFOAM

OpenFOAM (Open Field Operation And Manipulation) è un software CFD (Computational Fluid Dynamics), gratuito e open source sviluppato originariamente da OpenCFD Ltd nel 2004. È un software che sviluppa un metodo per l'analisi numerica e algoritmi per risolvere e analizzare i problemi di fluidodinamica e termodinamica. Inizialmente è stato progettato come piattaforma per la meccanica dei continui ma è adatto per creare simulazioni in vari ambiti della fisica. [6]

OpenFOAM è costituito da un insieme di solutori scritti in C++ che implementano il metodo ai volumi finiti. È un solutore monolitico, ovvero che al suo interno vi sono più librerie e solutori interdipendenti.

I passi da seguire per la risoluzione di un'analisi CFD con OpenFOAM sono:

1. Costruzione della griglia
2. Scelta del solutore opportuno
3. Impostazioni delle condizioni iniziali ed al contorno necessarie
4. Impostazioni dei dictionaries
5. Lancio del calcolo
6. Controllo dei residui della simulazione
7. Post-processing dei dati

Il livello di utilizzo di OpenFOAM con cui sono ottenuti i dati vede il caso memorizzato in una cartella con struttura fissa, contenente: *0*, *constant*, *system*. Un esempio di solver ampiamente utilizzato è *laplacianFoam*. Quest'ultimo analizza stato stazionario o il transitorio. Il risolutore evolve uno scalare, T , risolvendo l'equazione (9). [7]

In *0* troviamo dei file in cui sono specificate le condizioni iniziali, ovvero i valori delle grandezze all'istante iniziale. Si hanno tanti file quante sono le grandezze che prendono parte all'equazione che devono essere calcolate. Nel caso di *laplacianFoam* abbiamo il file *T* in cui sono definiti i valori iniziali e le condizioni al contorno del dominio scelto. Il file sarà costituito da diverse righe dove inizialmente si specifica l'unità di misura della grandezza espressa nel Sistema Internazionale di unità di misura mediante *dimensions*, il valore effettivo numerico della condizione iniziale mediante *internalField*, e infine, con *boundaryField* specificando dettagliatamente il type si determina la condizione al contorno della specifica zona del dominio.

Per immettere dei dati all'interno con la relativa unità di misura si utilizza: *dimensions [massa(kg), lunghezza(m) , time(s) , temperatura(K) , moli(n) , corrente(A) , intensità_luminosa(cad)]*. Corrisponde ad un array in cui si deve inserire l'esponente dell'unità di misura specifica tenendo conto dell'ordine seguente.

In *constant* si hanno informazioni riguardo quelle grandezze fisse che non variano nel corso del calcolo. In questo livello troviamo *polyMesh* nella quale è presente la Mesh del dominio discretizzata e importata all'interno di OpenFOAM sottoforma di directory e, altri possibili file. All'interno di *polyMesh* si hanno molteplici file:

1. *points* per identificare tramite vettori i vertici delle celle;
2. *boundary* in cui sono denotati i contorni del dominio con le rispettive facce e il tipo di facce;
3. *faces* in cui sono riportate in una lista tutte le facce del dominio;
4. *owners* una lista di ugual numero di facce;
5. *neighbours* in cui, fondamentalmente, sono riportate le facce interne come etichette di celle vicine ad altre.

Nel caso dell'utilizzo di *laplacianFoam* possiamo avere *transportProperties* come esempio di un file presente dentro *constant*, in cui si specifica il valore fisso della costante di diffusività termica (se nel caso studiato sia fissa).

In *system* si osserva la presenza di tre file differenti: *controlDict*, *fvSchemes*, *fvSolution*. Attraverso questi sono definite le impostazioni del solutore (schemi di discretizzazione, tolleranze di convergenza, ecc.). [14]

1. *controlDict* definisce i controlli generali per l'esecuzione del caso: criterio di inizio simulazione, tempo iniziale della simulazione, criterio di fine simulazione, tempo finale della simulazione, passo temporale, criterio per la scrittura dei risultati, intervallo di scrittura dei risultati ecc.;
2. *fvSchemes* definisce gli schemi di discretizzazione da usare per i diversi termini delle equazioni;
3. *fvSolution* definisce i metodi risolutivi per i sistemi lineari associati alle equazioni discrete.

Arbitrariamente è possibile utilizzare un file log in cui salvare le indicazioni sui numeri calcolati e i residui ad ogni time step. Al termine dell'esecuzione si genera una directory *postProcessing* in cui sono contenuti i dati finali desiderati, cartella necessaria per visualizzare ed esportare i dati in software e applicazioni diverso da OpenFOAM.

3. Metodo Risolutivo

In questo studio si pone l'attenzione sullo scambio termico conduttivo nel fegato umano quando viene messa in atto la terapia di ablazione termica per curare la presenza di un tumore con durata di 720s. Per riuscire a definire un protocollo con prestazioni ottimali si analizzano i risultati in termini di temperatura dello scambio termico conduttivo dopo aver simulato numericamente il processo fisico. Matematicamente andiamo a risolvere l'equazione (33) che genera la mappa di temperatura sul dominio stabilito. L'equazione che vogliamo risolvere deve tenere conto delle scoperte di Pennes riguardo la perfusione sanguigna, la possibilità che il tessuto abbia avuto un danneggiamento e la fonte di calore generata per effetto Joule. Effettuando diverse prove riusciamo ad affermare quale tipo di voltaggio genera le migliori prestazioni. Lo studio riguarda un problema CFD quindi si utilizza OpenFOAM. I passi effettuati sono:

1. Pre-processing:
 - scelta del dominio
 - discretizzazione di esso
2. Processing:
 - scelta delle condizioni iniziali desiderate
 - inserimento delle caratteristiche del caso
 - definizione e utilizzo del risolutore
3. Post-Processing:
 - analisi i dati ottenuti.

Vogliamo trovare quindi: la mappa di temperatura sull'area del dominio scelto e il danneggiamento ottenuto rispetto all'area totale.

L'equazione che andiamo a risolvere è la seguente:

$$(\rho c)_t \frac{\partial T}{\partial t} = \nabla \cdot (k_t \nabla T_t) + Q_{perf} + Q_{met} + Q_{ext} \quad (33)$$

Dove:

- pedice t indica riferimento al tessuto
- T temperatura tessuto biologico
- ρ densità
- c calore specifico
- k_t conduttività termica
- t time
- Q_{perf} termine di perfusione sanguigna
- Q_{met} fonte di calore metabolico

- Q_{ext} la densità di potenza RF esterna imposta al tessuto durante l'applicazione di potenza RF

Si prende in considerazione la vaporizzazione utilizzando il metodo dell'entalpia discusso e presentato in [8]. Si modella il termine $(\rho c)_t$ presente nell'equazione (33) con la seguente funzione a tratti:

$$(\rho c)_t = \begin{cases} (\rho_l c_l)_t & 0^\circ C < T_t \leq 99^\circ C \\ \frac{h_{fg} C_{w,t}}{\Delta T_{b,t}} & 99^\circ C < T_t \leq 100^\circ C \\ (\rho_g c_g)_t & T_t > 100^\circ C \end{cases}$$

Dove

- ρ_l, c_l densità e calore specifico del tessuto a temperatura sotto $100^\circ C$, quindi in fase liquida
- ρ_g, c_g densità e calore specifico del tessuto a temperatura sopra $100^\circ C$, quindi in fase gassosa
- h_{fg} prodotto del calore latente dell'acqua di vaporizzazione e densità dell'acqua a $100^\circ C$ pari a $2.17 \cdot 10^9 \frac{J}{m^3}$
- $C_{w,t}$ contenuto di acqua all'interno del tessuto epatico (68%) [12]

Nel dettaglio abbiamo che:

$$Q_{perf} = \beta \rho_b c_b \omega_b (T_b - T_t) \quad (34)$$

Dove:

- ρ_b densità del sangue
- c_b calore specifico del sangue
- ω_b coefficiente di perfusione sanguigno
- T_b temperatura del sangue
- T_t temperatura del tessuto
- β un coefficiente il cui valore può essere 0 o 1 in base al danno causato dal danneggiamento termico. Grandezza adimensionale. Assume come valore 0 quando il termine Ω grado di danneggiamento del tessuto risulta maggiore di 4.6 altrimenti vale 1. Il grado di danneggiamento Ω è calcolato dall'eq. (19) nel capito 2.1.3. Osserviamo che si va a spegnere la generazione di calore derivante dalla perfusione sanguigna quando notiamo che il tessuto biologico è circa completamente danneggiato.

$$\beta = \begin{cases} 0 & \Omega \geq 4.6 \\ 1 & \Omega < 4.6 \end{cases} \quad (35)$$

Nel caso seguente studio Q_{met} , cioè il calore generato dalle attività metaboliche delle cellule viene trascurato.

Infine, analizziamo l'ultimo termine dell'equazione precedente, ovvero come viene considerato il problema elettrico. La generazione di calore deriva dal contributo elettrico grazie all'effetto Joule.

$$Q_{ext} = \sigma |\mathbf{E}|^2 \quad (36)$$

Dove:

- \mathbf{E} campo elettrico, per cui $\mathbf{E} = -\nabla V$ la quale viene ricavata dall'equazione di governo del problema elettrico: $\nabla \cdot \sigma(T) \nabla V = 0$. (vedi capitolo 2.1.4)
- V voltaggio
- $\sigma(T)$ conducibilità elettrica

Osserviamo che la conducibilità è un parametro che dipende dal materiale ma il suo valore è influenzato anche dalla temperatura.

3.1 Grandezze termodinamiche e geometria

Lo studio viene condotto su una porzione di fegato umano contenente un tumore. Scegliamo una porzione cilindrica, la quale per simmetria può essere trattata come un dominio bidimensionale di forma rettangolare. Lungo l'asse della porzione di tessuto biologico cilindrico selezionato viene inserito l'applicatore RF, il quale è composto da un manico in plastica che termina con un ago metallico, che funge da elettrodo. L'applicatore è completamente inserito all'interno del tessuto epatico. La situazione può essere idealizzata come nella figura 4.

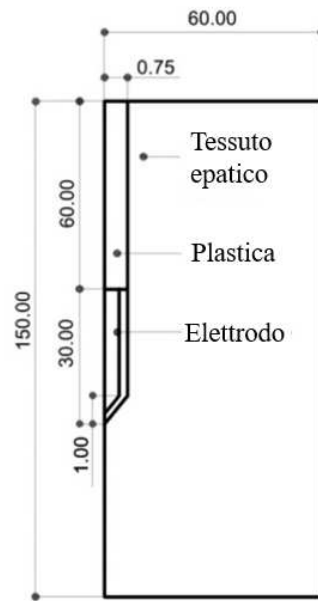


Figura 4: Geometria del modello usato

Nella figura sovrastante sono inserite le dimensioni effettive in millimetri del modello che abbiamo preso in considerazione, la figura ovviamente non è in scala. L'equazione differenziale presentata nel capitolo precedente deve essere risolta nel dominio costituito dal tessuto biologico: il rettangolo sovraesposto in figura 4 privato della componente elettrodo e plastica. Ci limitiamo ad analizzare la mappa della temperatura nel Tessuto epatico.

Bisogna tener conto che le grandezze termodinamiche di questo caso si dividono in grandezze fisse e costanti per tutta la durata del processo fisico e grandezze che dipendono dalla temperatura. È necessario definire a priori quali grandezze variano, quale legge descrive la variazione e quali sono fisse.

Le proprietà dei materiali quando la temperatura si trova in determinati range possono essere schematizzate nel seguente modo.

	$\rho \left[\frac{kg}{m^3} \right]$	$c \left[\frac{Jkg}{K} \right]$	$k \left[\frac{W}{m^2K} \right]$	$\sigma \left[\frac{s}{m} \right]$
Fegato	1080^a	3455^a	0.502	Vedi eq. (37)
	370^b	2156^b		
Sangue	1000^a	3639^a	0.502	
	370^b	2156^b		
Elettrodo	8000	480	15	7.4×10^6

Plastica	70	1045	0.026	10^{-5}
----------	----	------	-------	-----------

Rispettivamente con l'apice *a* si fa riferimento al valore quando la temperatura è minore di 100 °C e l'apice *b* quando è maggiore di 100 °C.

Nello specifico questi valori contenuti all'interno della tabella, quindi le proprietà dei vari materiali, devono essere inseriti all'interno della directory *constant* e salvati all'interno del file *transportProperties* nel seguente modo.

```

1  FoamFile
2  {
3      version      2.0;
4      format        ascii;
5      class         dictionary;
6      location      "constant";
7      object        transportProperties;
8  }
9  // *****

10
11  DT          DT [0 2 -1 0 0 0 0] 0.502;
12
13  lambda      0.502; //W/mK
14  rhob        1000;  // kg/m3
15  cpb         3639;  // J/kgK
16
17  rhot        1080;  // kg/m3
18  cpt         3455;  // J/kgK
19
20  omega       0.019; // 1/s
21  Ta0         310.15; // K
22
23  A           7.39e+39; // 1/s
24  dE          257700; // J/mol
25  R           8.314; //J/molK
26
27
28  // *****

```

Codice 1:

Directory: Constant File: transportProperties

La conduttività elettrica nei tessuti e nel sangue è stata definita come una funzione a tratti dipendente dalla temperatura così da considerare la perdita di contenuto d'acqua; infatti, si ha una riduzione drastica all'aumentare della temperatura e al verificarsi della vaporizzazione. Infatti vediamo dall'eq (37)

$$\sigma(T) = \begin{cases} 0.19 \cdot e^{0.015(T-37)} & 0^\circ\text{C} < T < 99^\circ\text{C} \\ 0.19 \cdot 2.5345 & 99^\circ\text{C} \leq T \leq 100^\circ\text{C} \\ 0.19 \cdot (2.5345 - 0.50183(T - 100)) & 100^\circ\text{C} \leq T \leq 105^\circ\text{C} \\ 0.19 \cdot 2.5345 \cdot 10^{-2} & T > 105^\circ\text{C} \end{cases} \quad (37)$$

Con T si intende la Temperatura.

3.1.1 Discretizzazione Dominio

Dopo aver scomposto nel dettaglio il caso che si deve studiare ed approfondito il dominio scelto, bisogna discretizzarlo. La discretizzazione è stata elaborata con un software esterno da OpenFOAM. Grazie a questo si ottiene la mesh specifica del dominio che può essere importata all'interno di OpenFOAM sotto forma della directory *polyMesh* internamente a *constant*. La mesh è tale per cui può essere applicato il metodo dei volumi finiti.

L'esempio con un numero di celle fittizio del nostro caso specifico è rappresentato in figura 5. È evidente che la risoluzione dell'esempio riportato è troppo piccola. I test sono stati effettuati con una risoluzione maggiore della mesh.

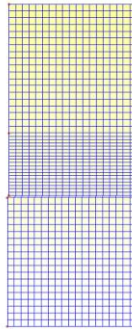


Figura 5: esempio discretizzazione dominio scelto

In questa fase con il software esterno devono essere definiti i luoghi geometrici del dominio dove successivamente saranno allocate le condizioni al contorno specifiche mediante OpenFOAM.

Come accennato nei capitoli precedenti la mesh viene inserita all'interno di OpenFOAM sottoforma di una directory contenente diversi file. Uno di questi è *boundary* nel quale sono salvate le seguenti informazioni:

```

1  FoamFile
2  {
3      version      2.0;
4      format        ascii;
5      class         polyBoundaryMesh;
6      location      "constant/polyMesh";
7      object        boundary;
8  }
9
10 7
11 (
12     bottom
13     {
14         type patch;
15         nFaces 59;
16         startFace 20767;
17     }
18     dx
19     {
20         type patch;
21         nFaces 178;
22         startFace 20826;
23     }
24     frontandbackplanes
25     {
26         type empty;
27         nFaces 21004;
28         startFace 21004;
29     }
30
31     q_in
32     {
33         type patch;
34         nFaces 60;
35         startFace 42008;
36     }
37     sx_p
38     {
39         type patch;
40         nFaces 59;
41         startFace 42068;
42     }
43     sx_t
44     {
45         type patch;
46         nFaces 59;
47         startFace 42127;
48     }
49     top
50     {
51         type patch;
52         nFaces 59;
53         startFace 42186;
54     }
55 )

```

Codice 2:

Directory: Constant/polyMesh File: boundary.

All'interno possiamo notare i luoghi geometrici del dominio in cui dobbiamo andare a settare le condizioni al contorno; infatti, vengono specificati mediante *type* i tipi di condizioni al contorno che il sistema pretenderà. Viene specificato il nome del luogo

geometrico, il tipo di condizione, il numero delle facce contenute nel raggruppamento, e il numero della faccia iniziale.

- *bottom*, *dx*, *sx_t*, *top* corrispondono agli estremi del dominio in cui è presente tessuto biologico del fegato. Definendo la variabile di tipo *patch*.
- Con *sx_p* si fa riferimento all'estremo sinistro del dominio in cui si ha la componente plastica dell'applicatore RF. La variabile è definita di tipo *patch*.
- Con *q_in* viene identificato l'estremo sinistro del dominio dove è presente l'elettrodo, zona che funge da sorgente di calore grazie all'effetto Joule.
- Con *frontandbackplanes* vengono inseriti tutti i bordi che sono necessari per avere una mesh tridimensionale, dove però non è presente alcun valore. Viene infatti definito di tipo *empty*: nessun valore, così da avere effettivamente una situazione di lavoro bidimensionale. La necessità deriva dal fatto che OpenFOAM lavora solamente con mesh 3D, ma nel nostro caso ci limitiamo al 2D.

3.1.2 Condizioni al contorno

In questa fase del lavoro dobbiamo assegnare la specifica condizione al contorno, ovvero il valore della soluzione di un'equazione differenziale ai margini dell'insieme di definizione dell'equazione. I margini dove devono essere inseriti sono stati appena descritti nel paragrafo precedente.

Essendo due le equazioni differenziali che dobbiamo risolvere dobbiamo settare queste condizioni per entrambe le incognite: Temperatura, Voltaggio. Sono presenti altre due variabili, e quindi altri due file, in quanto sono state utilizzate in maniera ausiliaria per la risoluzione.

Inizialmente ci soffermiamo sulle condizioni al contorno della temperatura. Di seguito il codice con cui sono state definite.

```

1  dimensions      [0 0 0 1 0 0 0];
2
3  internalField    uniform 310.15;
4
5  boundaryField
6  {
7      bottom
8      {
9          type      zeroGradient;
10     }
11
12     dx
13     {
14         type      zeroGradient;
15     }
16
17     q_in
18     {
19         type      zeroGradient;
20     }
21
22     sx_t
23     {
24         type      zeroGradient;
25     }
26
27     sx_p
28     {
29         type      zeroGradient;
30     }
31
32     top
33     {
34         type      zeroGradient;
35     }
36 }
37 }

```

Codice 3:
directory:0 File:T

Osserviamo che l'unità di misura corrisponde al Kelvin K. Abbiamo inserito la temperatura iniziale pari a 310.15 K, ovvero 37 °C. Le condizioni in tutti i margini del dominio sono settate con il *type zeroGradient*, ovvero abbiamo imposto in ognuno la condizione di adiabaticità.

Si ricordi lo scopo di questo studio: trovare il modo migliore in cui applicare il voltaggio per ottenere danno maggiore possibile e temperatura relativamente bassa. Il procedimento con cui vengono fatte le varie prove consiste nell'andare ad impostare condizioni di contorno

differenti della variabile V (cioè il voltaggio). Sono stati svolti 4 test diversi, quindi abbiamo 4 diverse modalità di applicazione della differenza di potenziale. Osserviamo interamente il *file* V contenuto nella *directory* 0 del primo test, in seguito, solo la condizione del voltaggio dell'elettrodo nei restanti 3 test.

- Nel *test0* abbiamo inserito i seguenti dati:

```

1  dimensions      [1 2 -3 0 0 -1 0];
2
3  internalField    uniform 0.0;
4
5  boundaryField
6  {
7      bottom
8      {
9          type      fixedValue;
10         value      uniform 0;
11     }
12
13     dx
14     {
15         type      zeroGradient;
16     }
17
18     q_in
19     {
20         type      groovyBC;
21         variables  "Tg=10;Tn=8;step=((time()/(Tg+Tn))-0.5)%1+0.5;";
22         valueExpression  " (step<0.555) ? 90 : 0 ";
23         value      uniform 0;
24     }
25
26     sx_p
27     {
28         type      zeroGradient;
29     }
30
31     sx_t
32     {
33         type      zeroGradient;
34     }
35
36     top
37     {
38         type      fixedValue;
39         value      uniform 0;
40     }
41 }

```

Codice 4

Test0 Directory:0 File: V

Come prima cosa capiamo subito che si tratta del file del voltaggio dalle unità di misure definite. Immettiamo come valore iniziale 0V e poniamo a massa i margini di *top* e *bottom*. Utilizzando *type fixedValue* andiamo a definire che il margine specifico deve avere un valore fisso e lo poniamo a 0 con la dicitura *value uniform 0*. Analogamente al caso della temperatura, sui margini del tessuto rimanenti e la componente plastica, abbiamo usato *zeroGradient* per definire l'isolamento elettrico, adiabaticità.

Vogliamo riproporre un andamento del voltaggio pulsato: intervallo di tempo in cui l'ampiezza è pari a 90V seguito da un intervallo di tempo di 0V. Vogliamo modellizzare un segnale periodico. Ciò significa che vogliamo definire una condizione al contorno in q_in che corrisponda ad un'onda quadra. Scegliamo come tempo di erogazione $T_g = 10s$ seguito da un tempo di segnale nullo pari a $T_n = 8s$. Il periodo del segnale che si ripete è dunque 18s. L'andamento della grandezza V viene rappresentato nel seguente grafico 1.

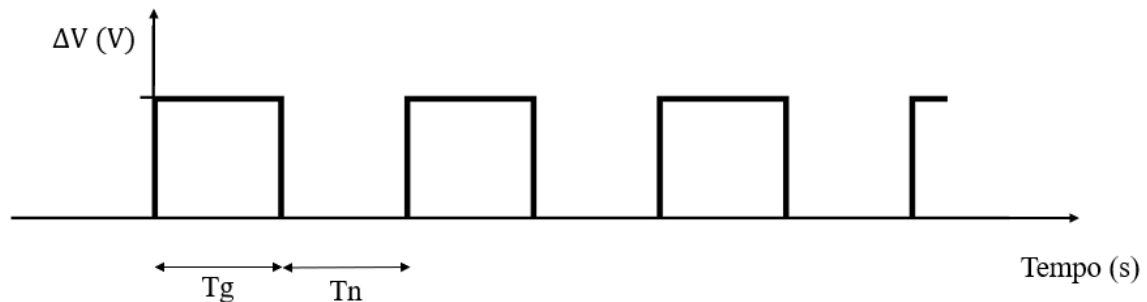


Grafico 1: Rappresentazione quantitativa andamento voltaggio test0

Per risolvere questa problematica della variabile q_in è stata imposta una nuova tipologia di margine che fa parte della libreria *libgroovyBC*; infatti, per usare *type groovyBC* è necessario aggiungere la libreria nel *controlDict*. Mediante questa libreria riusciamo a definire un andamento pulsato: creiamo una successione di gradini.

- *Variables*: riusciamo a definire delle variabili temporanee.
- *valueExpression*: corrisponde alla stringa con il valore da utilizzare quando è necessaria una condizione di Dirichlet. In questo caso la condizione viene utilizzata.
- *Value uniform 0*: necessario per puro formalismo

Nelle *Variables* vengono settate tre grandezze. T_n e T_g corrispondono all'intervallo di tempo per ogni gradino in cui la funzione rispettivamente assume valore massimo e minimo. La somma tra T_n e T_g corrisponde al periodo del segnale. Viene introdotta anche la variabile *step* attraverso la quale si ottiene il tempo in funzione del periodo del segnale. Nella sua definizione viene utilizzato l'operatore $\%$ ovvero l'operatore modulo: un'operazione di divisione con un resto come risultato. La variabile *step* assume il valore del resto della divisione tra la variabile tempo *time()* rapportata al periodo del segnale aumentato di un certo valore e una grandezza fissa, i valori aggiunti sono un artificio matematico per avere come resto un valore compreso tra 0 ed 1.

La problematica risiede nell'implementazione dell'operatore $\%$ in *groovyBC* leggermente diversa rispetto a quello standard. Nel caso standard l'operazione $a\%b$ mappa in $[0,b)$, ovvero risultati ottenibili sono contenuti all'interno dell'intervallo tra 0 e b incluso 0.

Invece, in questo caso l'implementazione differente fa sì che l'operazione mappi in $[-b/2, b/2)$ incluso $-b/2$. Dato che vogliamo tornare all'operatore standard e avere un intervallo pari a $[0,1)$, occorre aggiungere e sottrarre per il valore $b/2$, che nel nostro caso corrisponde a 0,5.

All'inizio di ogni ciclo *step* assume il valore effettivo 0, aumenta fino ad arrivare ad 1 che rappresenta la fine del ciclo stesso, il processo si ripropone per ogni ciclo. Esso funge quindi da discriminante per comprendere se la funzione deve assumere valore massimo o nullo. Questo è spiegato all'interno di *valueExpression*.

In *valueExpression* è stato usato l'operatore condizionale la cui sintassi è *(test ? val1 : val2)*: quando la condizione espressa in *test* è vera, come valore si assume *val1*, altrimenti *val2*. Parallelamente quando nel nostro caso *step* è minore di un valore definito come T_g normalizzato con il periodo totale (in questo test 0.555) la funzione deve essere massima altrimenti nulla. Il valore cambia per ogni test perché sono stati settati rispettivamente differenti periodi.

Date le caratteristiche scelte dei periodi di on ed off dei gradini e che il tempo totale della simulazione è pari a 720s, sono presenti 40 cicli dove in ogni ciclo abbiamo periodo di on con erogazione massima 90V seguita da periodo di off quindi 0 V. Possiamo affermare che durante tutti i 720s il tempo totale di erogazione è 400s mentre il tempo totale di off corrisponde a 320s.

- Nel *test1* abbiamo inserito i seguenti dati:

Come già detto identiche caratteristiche con il precedente, tranne la condizione al contorno della variabile q_{in} . È descritta nel seguente codice:

```

18 |   q_in
19 |   {
20 |       type                groovyBC;
21 |       variables           "Tg=10;Tn=8;step=((time()/(Tg+Tn))-0.5)%1+0.5;a=-0.000277*time()+1;";
22 |       valueExpression     " (step<0.555) ? 90*a : 0 ";
23 |       value               uniform 0;
24 |   }
25 |

```

Codice 5

Test1 Directory:0 File: V

La modifica fatta risiede nell'andamento di q_{in} , ovvero, la funzione è composta da treni di gradini, i quali sono via via attenuati. L'ampiezza di gradino in gradino decresce a mano a mano che il tempo aumenta. La decrescita dall'istante iniziale ($t=0s$) all'istante finale ($t=720s$) è del 20%. L'andamento in maniera qualitativa viene rappresentato nel seguente grafico.

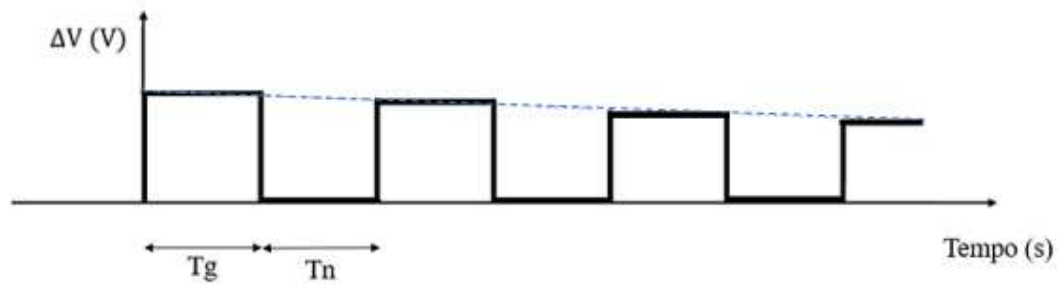


Grafico 2: Rappresentazione quantitativa andamento voltaggio test1

Per attuare la decrescita dei picchi viene semplicemente immessa in *variables* una nuova variabile a che rappresenta una retta, la quale provoca l'attenuazione del segnale poiché è moltiplicata con il valore massimo del segnale stesso in *valueExpression*. In questo modo i picchi decrescono dipendentemente dalla retta. Quest'ultima è stata definita selezionando semplicemente il coefficiente angolare e il termine noto dopo aver scelto la percentuale di attenuazione che vogliamo dare al segnale onda quadra. Dato che in questo caso vogliamo una diminuzione del valore di picco del 20% la retta necessaria passerà per due punti: punto A (0,1) e punto B (720, 0.8). Il coefficiente angolare m di una retta è calcolato mediante la formula per la sua determinazione sapendo due punti contenuti nella retta eq. (38)

$$m = \frac{y_B - y_A}{x_B - x_A} \quad (39)$$

La determinazione del termine noto è banale in quanto nell'istante $t=0s$ non vogliamo alcuna attenuazione, dunque, il valore della retta deve essere pari ad 1.

Da questo deriva la retta inserita all'interno del codice pari ad $a = -0.000277x + 1$. Ovviamente la variabile delle ascisse non è altro che la variabile tempo espressa nel codice con la dicitura *time()*. Essendo una retta decrescente il suo coefficiente angolare è negativo. Il valore a che assume la retta nel corso della simulazione, che moltiplica il voltaggio massimo 90V, è compreso quindi tra 1 e 0,8.

Dato che le caratteristiche scelte dei periodi di on ed off dei gradini sono identiche al test precedente abbiamo stesso numero di cicli e stessi tempi di erogazione con valore decresce durante la simulazione.

- Nel test2 abbiamo inserito i seguenti dati:

```

18 | q_in
19 | {
20 |     type                groovyBC;
21 |     variables            "Tg=35;Tn=10;step=((time()/(Tg+Tn))-0.5)%1+0.5;a=-0.000277*time()+1;";
22 |     valueExpression      " (step<0.777) ? 90*a : 0 "
23 |     value                uniform 0;
24 | }

```

Codice 6

Test1 Directory:0 File: V

Sono state fatte le stesse considerazioni dei test precedenti ma cambiando i dati dei Tg e Tn. È stato imposto periodo on di erogazione Tg di 35s seguito da periodo di off di 10s. Il periodo totale del segnale che verrà ripetuto ciclicamente è 45s. Viene mantenuta la pendenza del 20%, ciò significa che la retta è analoga al test1. Abbiamo una situazione con la stessa pendenza ma il periodo del segnale maggiore del precedente, ne segue che sono presenti meno pulsazioni di voltaggio. Durante tutta la simulazione di 720s si manifestano solo 16 gradini, il che comporta un tempo totale di erogazione pari a 560s e un tempo totale senza alcuna erogazione di 160s.

- Nel test3 abbiamo inserito i seguenti dati:

```

18 | q_in
19 | {
20 |     type                groovyBC;
21 |     variables            "Tg=15;Tn=5;step=((time()/(Tg+Tn))-0.5)%1+0.5;a=-0.00055*time()+1;";
22 |     valueExpression      " (step<0.75) ? 90*a : 0 "
23 |     value                uniform 0;
24 | }

```

Codice 7

Test1 Directory:0 File: V

L'ultimo test effettuato è stato compiuto con un'ulteriore variazione dei periodi dei gradini e attenuando maggiormente il segnale in quando è stata scelta una decrescita del 40%. È stato imposto periodo on di erogazione Tg di 15s seguito da periodo di off di 5s. Il periodo totale del segnale che verrà ripetuto ciclicamente è 20s. Avendo cambiato la decrescita rispetto ai precedenti si può notare la retta di attenuazione diversa: diverso coefficiente angolare ma stesso termine noto. I punti attraverso cui è stato calcolato il coefficiente angolare m sono A (0,1) e B (720, 0.6). Di conseguenza in modulo il coefficiente angolare è maggiore rispetto ai precedenti test. Il valore a che assume la retta nel corso della simulazione, che moltiplica il voltaggio massimo 90V, è dunque compreso quindi tra 1 e 0,6.

Per come sono stati settati i periodi di on ed off di ogni ciclo, ne segue che il periodo del segnale durante i 720s produce 36 cicli. Perciò, il tempo totale di erogazione equivale a 540 s, quello di assenza di erogazione, quindi off, pari a 180s.

Ricordiamo che sono presenti ulteriori due file: beta, sigma.

Beta corrisponde alla variabile presente nell'equazione (34) che determina la presenza o l'assenza dell'effetto di perfusione sanguigna.

```

1  dimensions      [0 0 0 0 0 0 0];
2
3  internalField    uniform 1;
4
5  boundaryField
6  {
7      bottom
8      {
9          type      calculated;
10         value      uniform 1;
11     }
12
13     dx
14     {
15         type      calculated;
16         value      uniform 1;
17     }
18
19     frontandbackplanes
20     {
21         type      empty;
22     }
23
24     q_in
25     {
26         type      calculated;
27         value      uniform 1;
28     }
29
30     sx_p
31     {
32         type      calculated;
33         value      uniform 1;
34     }
35
36     sx_t
37     {
38         type      calculated;
39         value      uniform 1;
40     }
41
42     top
43     {
44         type      calculated;
45         value      uniform 1;
46     }
47 }

```

Codice 8

Test1 Directory:0 File: V

È evidente sia un termine adimensionale. Valore iniziale pari a 1, tutti i margini definiti come *type calculated* e valore 1.

L'ultimo file presente internamente a 0 è sigma. Questo corrisponde alla conducibilità elettrica è stato settato con condizioni al contorno in ogni margine con *zeroGradient* ed inserita la condizione iniziale parti a $0.19 \left[\frac{S}{m} \right]$.

3.2 Processing

Una volta terminata la fase di pre-processing in cui è stato analizzato il caso da trattare, discretizzato il dominio, inserite le costanti e la mesh all'interno di OpenFOAM si passa alla fase di elaborazione effettiva in cui si ha la progettazione del risolutore, e il settaggio delle impostazioni dei vari test.

3.2.1 Risolutore

Per la risoluzione è stato implementato un primo risolutore chiamato *bioheatFoamV_beta3*, mediante il quale si riesce a calcolare la mappa di temperatura nel dominio scelto. Successivamente si utilizza un secondo risolutore chiamato *damage* per trarre considerazioni inerenti al danno termico generato al fine di valutare quale protocollo per l'erogazione del voltaggio è migliore in termini di danno e temperatura.

Come prima cosa analizziamo i vari blocchi del file *bioheat.cpp* del risolutore *bioheatFoamV_beta3*.

```
1  #include "fvCFD.H"
2  #include "simpleControl.H"
3
4  // * * * * *
5
6  int main(int argc, char *argv[])
7  {
8      #include "setRootCase.H"
9
10     #include "createTime.H"
11     #include "createMesh.H"
12     #include "createFields.H"
13
14     simpleControl simple(mesh);
15
16     // * * * * *
17
18     Info<< "\nCalculating temperature distribution\n" << endl;
19     //-----
```

```

20
21 while (simple.loop())
22 {
23     Info<< "Time = " << runTime.timeName() << nl << endl;
24
25     forAll(mesh.C(), celli)
26     {
27         if (T[celli] <= 372.15)
28         {
29             rho[celli] = rhot;
30             cp[celli] = cpt;
31
32             //
33             rhobf[celli] = rhob;
34             cpbf[celli] = cpb;
35             alpha[celli] = lambda/(rho[celli]*cp[celli]);
36         }
37
38         if (T[celli] > 372.15 && T[celli] <= 373.15 )
39         {
40             rho[celli] = 0.68*2.17e9;
41             cp[celli] = 1.0;
42             //
43
44             rhobf[celli] = rhob;
45             cpbf[celli] = cpb;
46             alpha[celli] = lambda/(rho[celli]*cp[celli]);
47         }
48         if(T[celli] > 373.15)
49         {
50             rho[celli] = 370;
51             cp[celli] = 2156;
52             //
53             rhobf[celli] = 370;
54             cpbf[celli] = 2156;
55             alpha[celli] = lambda/(rho[celli]*cp[celli]);
56         }
57     }
58

```

Codice 9

Parte 1:Calcolo proprietà rispetto alla temperatura: densità, calore specifico e diffusività termica

```

59
60     forAll(mesh.C(), celli)
61     {
62         if (T[celli] < 372.15)
63         {
64             sigma_[celli] = 0.19*Foam::exp(0.015*(T[celli]-310.15)) ;
65         }
66         if (T[celli] >= 372.15 && T[celli] <= 373.15 )
67         {
68             sigma_[celli] = 0.481555 ;
69         }
70         if (T[celli] > 373.15 && T[celli] <= 378.15 )
71         {
72             sigma_[celli] = 0.19*( 2.5345 - 0.50183*(T[celli]-373.15)) ;
73         }
74         if (T[celli] > 378.15 )
75         {
76             sigma_[celli] = 0.19*( 2.5345*0.01*T[celli]) ;
77         }
78     }
79
80     solve
81     (
82         fvm::laplacian(sigma_, V)
83     );
84 //
85     Esq = magSqr(-fvc::grad(V));
86     forAll(mesh.C(), celli)
87     {
88         //ohmic_[celli] = (sigma_[celli]* Esq[celli]) ;
89         ohmic_[celli] = (sigma_[celli]* Esq[celli])/( rho[celli]*cp[celli] ) ;
90     }
91

```

Codice 10

Parte 2: Risoluzione problema elettrico

```

92 //
93 while (simple.correctNonOrthogonal())
94 {
95     forAll(mesh.C(), celli)
96     {
97         om[celli] += A*Foam::exp(-dE/(R*T[celli]))*runTime.deltaTValue();
98
99         if (om[celli] < 4.6)
100         {
101             beta[celli] = 1;
102         }
103         else
104         {
105             beta[celli] = 0;
106         }
107         pennes_[celli] = beta[celli]*rhobf[celli]*cpbf[celli]*omega*(Ta0 - T[celli])/(rho[celli]*cp[celli]) ;
108     }
109     solve
110     (
111         fvm::ddt(T) - fvm::laplacian(alpha, T) - pennes_ - ohmic_
112     );
113 }
114
115 #include "write.H"

```

```

116
117     Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
118         << "   ClockTime = " << runTime.elapsedClockTime() << " s"
119         << nl << endl;
120 }
121
122 Info<< "End\n" << endl;
123
124 return 0;
125 }

```

Codice 11

Parte 3: Risoluzione problema termico

Mediante la prima parte vengono aggiornati i valori delle grandezze fisiche tenendo conto del valore della temperatura in quel punto. Attraverso la seconda parte viene risolto il problema elettrico ovvero calcola l'equazione (36) tenendo conto come varia la conducibilità elettrica in base alla temperatura. Infine, con la terza parte si risolve il problema termico tenendo conto della possibilità di avere un danneggiamento della cellula, calcolato mediante Arrhenius.

Al termine del processo viene utilizzato un altro risolutore chiamato *damage* il quale ha il compito di calcolare il rapporto delle cellule danneggiate rispetto alla porzione di fegato considerata nel domino. Di seguito viene esposto il file *damage.cpp* del risolutore.

```

1  #include "fvCFD.H"
2
3  // * * * * *
4  int main(int argc, char *argv[])
5  {
6      timeSelector::addOptions();
7
8      #include "addRegionOption.H"
9      #include "setRootCase.H"
10     #include "createTime.H"
11     instantList timeDirs = timeSelector::select0(runTime, args);
12     #include "createNamedMesh.H"
13     #include "staticFields.H"
14
15     // *****
16     scalar areat = 0.0;
17     forAll(mesh.C(), celli)
18     {
19         areat += mesh.V().field()[celli]/0.07 ;
20     }
21
22
23     // *****
24
25     forAll(timeDirs, timeI)
26     {
27

```

```

28     runTime.setTime(timeDirs[timeI], timeI);
29     Info<< "Time = " << runTime.timeName() << endl;
30     #include "createFields.H"
31
32     scalar aread = 0.0;
33     forAll(mesh.C(), celli)
34     {
35         if (beta[celli] == 0 )
36         {
37             aread += mesh.V().field()[celli]/0.07 ;
38         }
39     }
40
41
42     Info << "    Damage ratio: " << aread/areat << " []\n" << endl;
43
44 }
45
46 Info << "End\n" << endl;
47 return 0;
48 }

```

Codice 12

Parte 3:Risolutore damage

Si noti che con questo riusciamo a trovare il rapporto tra l'area in cui presente danno termico e quella totale del dominio. Nelle varie prove effettuate, per come è stata costruita la griglia della mesh, si passa dal volume fittizio della griglia all'area del dominio effettivo dividendo per il valore 0.07.

3.2.2 System

Nella fase di processing rientra anche la scelta di come deve essere effettuato il test. Ricordiamo che la cartella *system* sono presenti i file *controlDict*, *fvSchemes* e *fvSolution*. Analizziamo nel dettaglio *controlDict*.

```

1  FoamFile
2  {
3      version      2.0;
4      format       ascii;
5      class        dictionary;
6      location     "system";
7      object       controlDict;
8  }
9  // *****

```



```

10
11 application      laplacianFoam;
12
13 startFrom        latestTime;
14
15 startTime        0;
16
17 stopAt           endTime;
18
19 endTime          720;
20
21 deltaT           0.0025;
22
23 writeControl      runtime;
24
25 writeInterval     10;
26
27 purgeWrite       0;
28
29 writeFormat       ascii;
30
31 writePrecision    12;
32
33 writeCompression  off;
34
35 timeFormat        general;
36
37 timePrecision     12;
38
39 runtimeModifiable true;
40
41 libs ( "libgroovyBC.so" );
42
43 functions
44 {
45
46     probes
47     {
48         functionObjectLibs ( "libsampling.so" );
49
50         type      probes;
51
52         name      probes;
53
54         fields ( T );
55
56         probeLocations
57         (
58             (0.008 0.09 0.0) // at top of the helmet
59         );
60     }
61
62 }

```

Codice 13

Directory: system File: ControlDict

In questo file vengono inserite le impostazioni con cui deve essere effettuato il test. È stata inserita la durata di 720 s, il criterio per la scrittura dei risultati, l'intervallo di scrittura dei

risultati. È stata introdotta anche una sonda mediante *function* per tenere traccia dell'andamento della temperatura in un punto preciso. Questo è stato un passo fondamentale per verificare la veridicità dei risultati e il corretto funzionamento del risolutore creato poiché in questo modo è stato possibile confrontare i dati ottenuti con articoli del settore.

3.3 Post-Processing

Al termine della risoluzione è necessario esportare i dati ottenuti per poter essere visionati graficamente rendendo più facile le considerazioni finali. Per ogni test analizzato abbiamo come risultato la mappa della temperatura T , il termine di danneggiamento β e l'andamento della percentuale del danno nel tempo.

- Test0: Danneggiamento a 720s pari a 10,6% rispetto al volume totale.

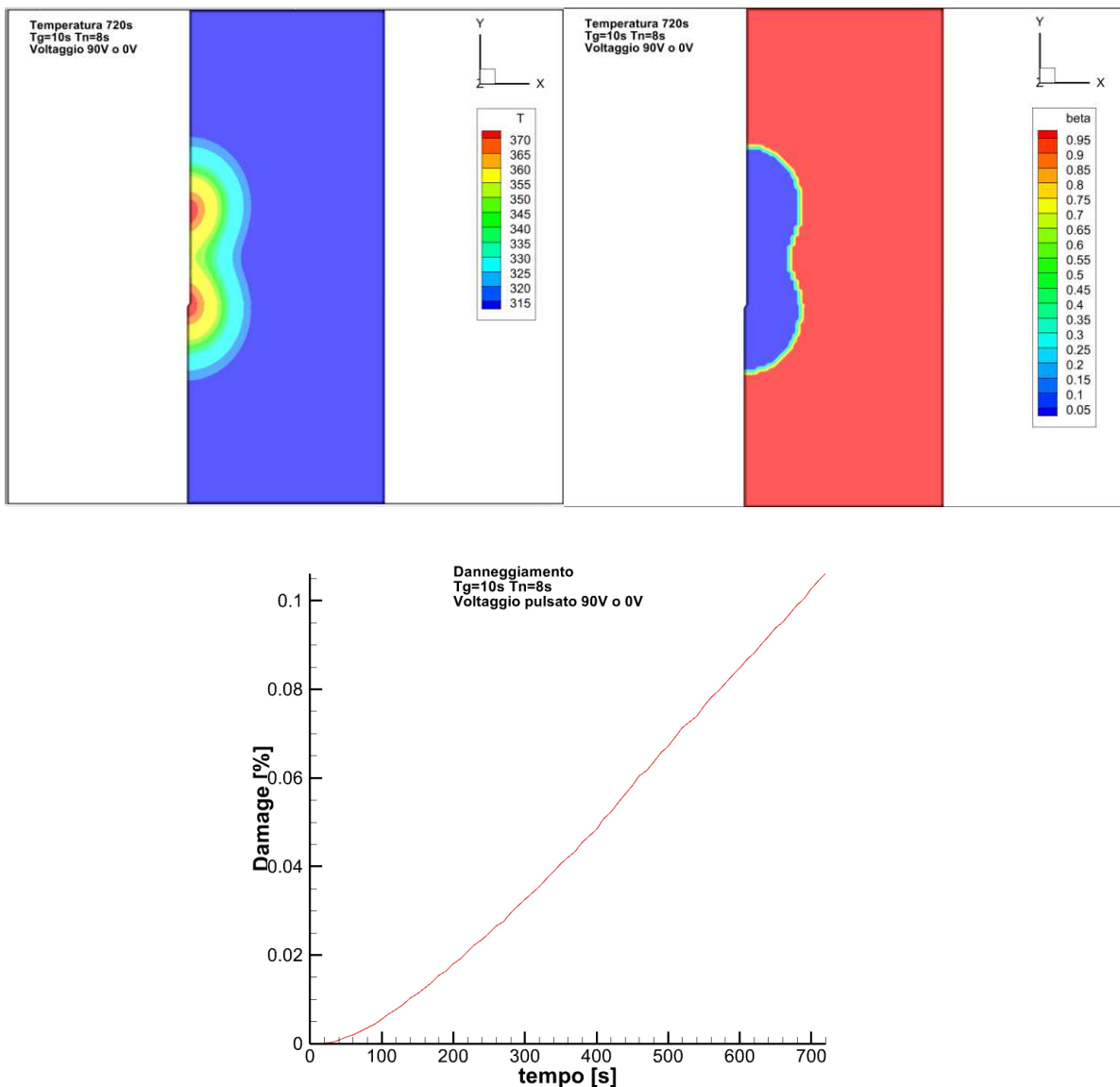


Figura 6: mappa di temperatura, mappa di danneggiamento e l'andamento della percentuale del danno nel tempo del test0.

- Test1: Danneggiamento a 720s pari a 8,3% rispetto al volume totale

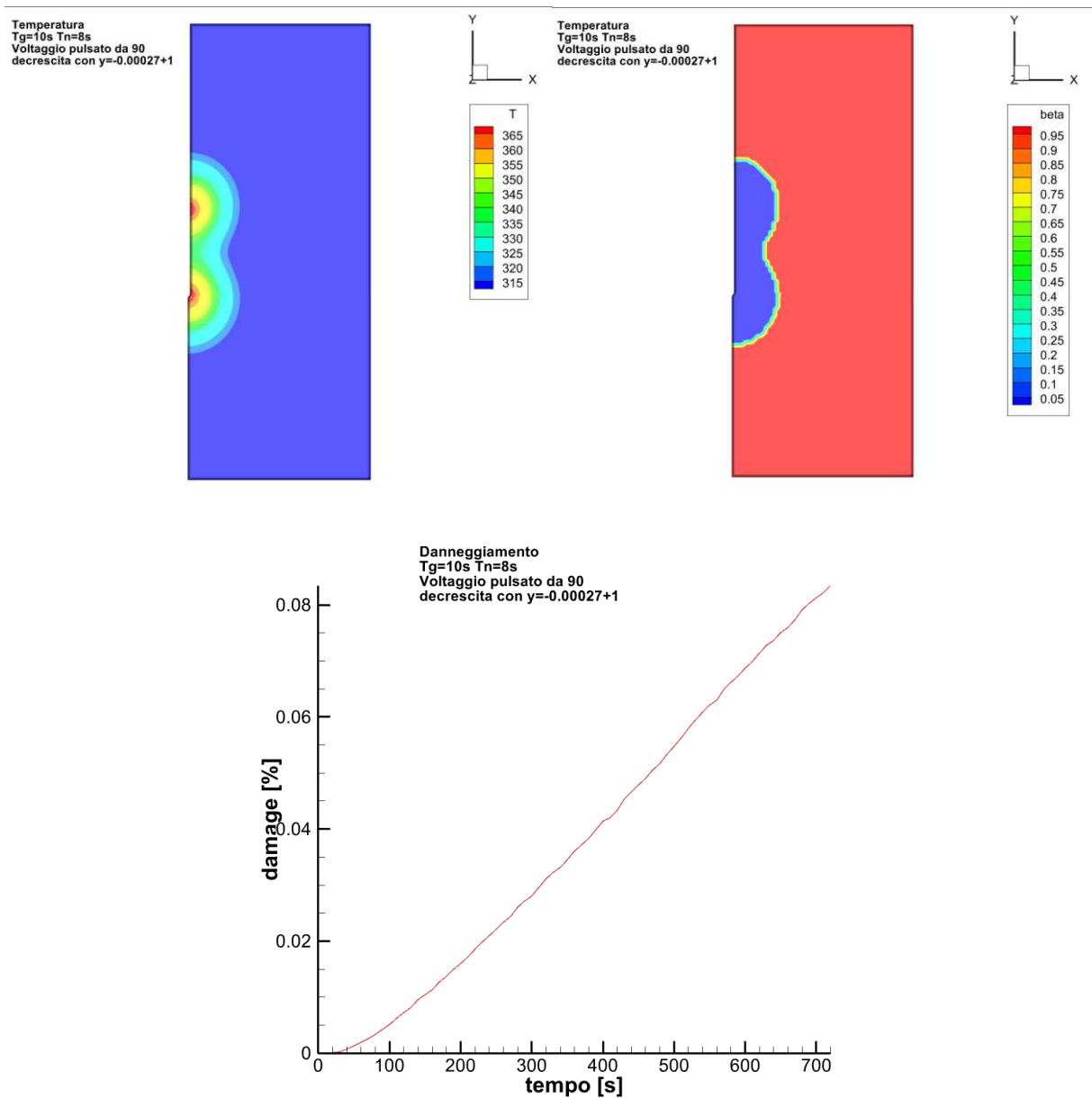


Figura 7: mappa di temperatura, mappa di danneggiamento e andamento della percentuale del danno nel tempo del test1.

- Test2: Danneggiamento a 720s pari a 13,2% rispetto al volume totale

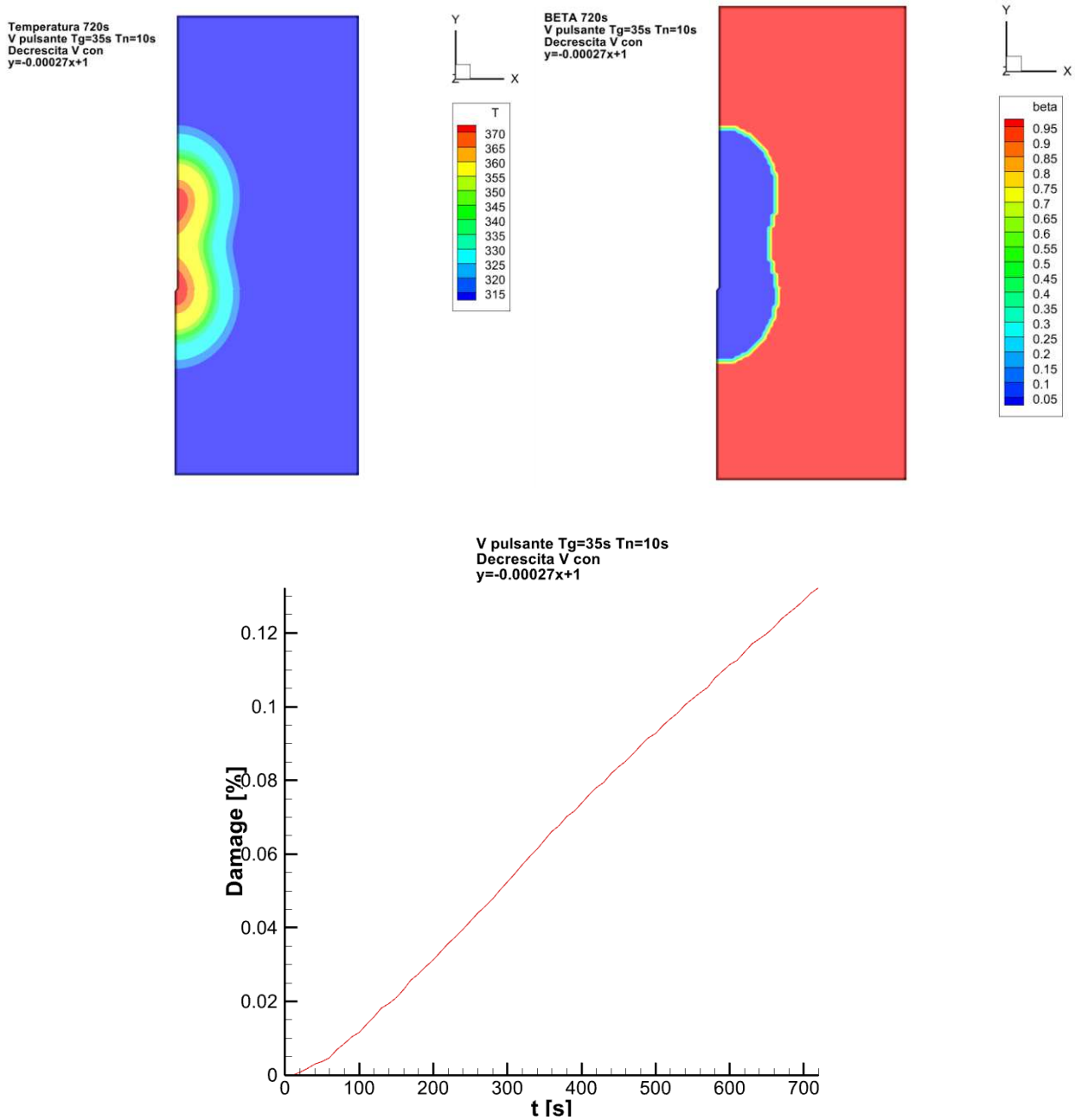


Figura 8: mappa di temperatura, mappa di danneggiamento e andamento della percentuale del danno nel tempo del test2

- Test3: Danneggiamento a 720s pari a 10,1 % rispetto al volume totale

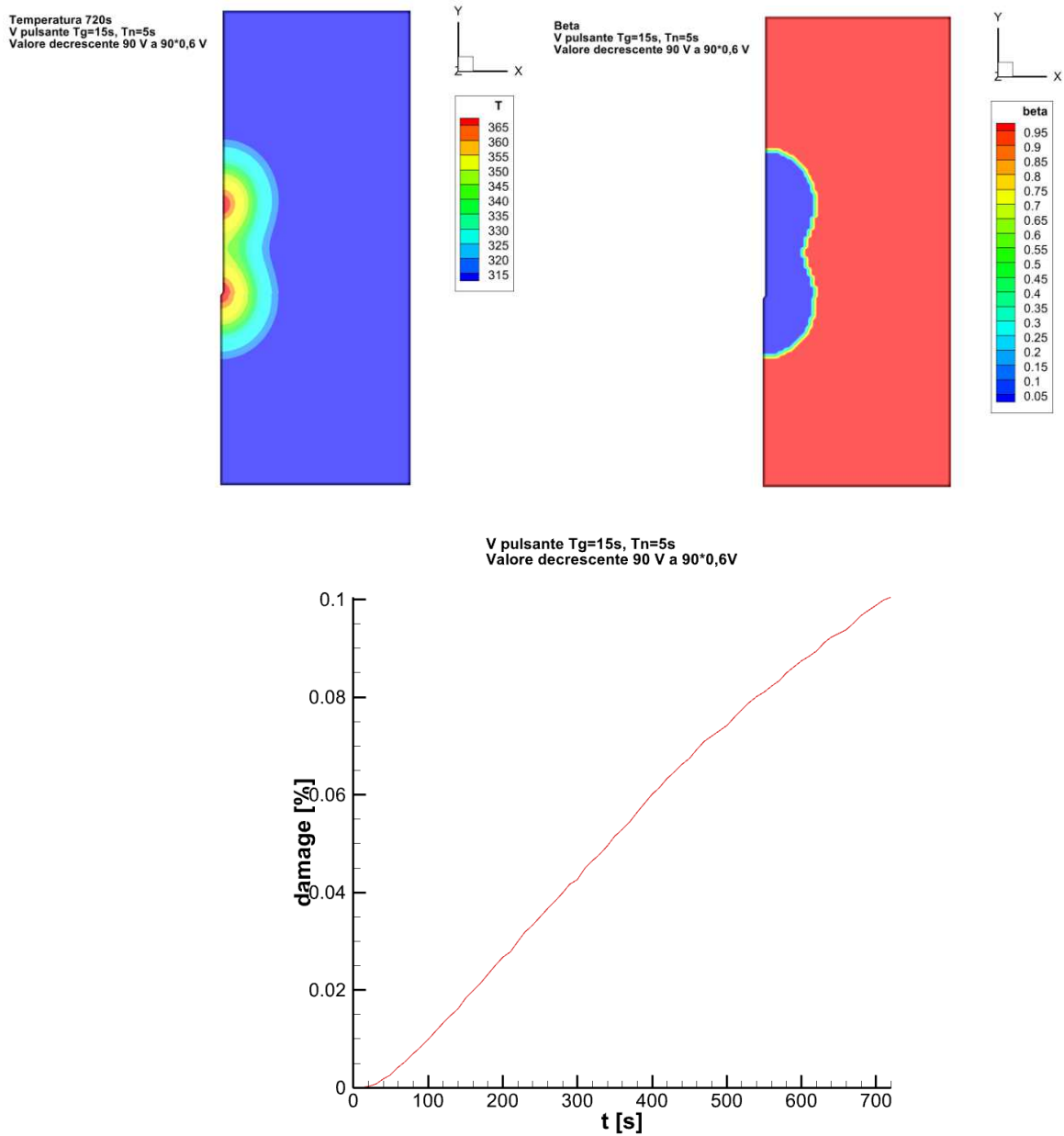


Figura 9: mappa di temperatura, mappa di danneggiamento e andamento della percentuale del danno nel tempo del test3.

Infine, se vogliamo visionare l'andamento del danneggiamento dei vari test effettuati in un grafico unico otteniamo la seguente situazione rappresentata nel grafico 3.

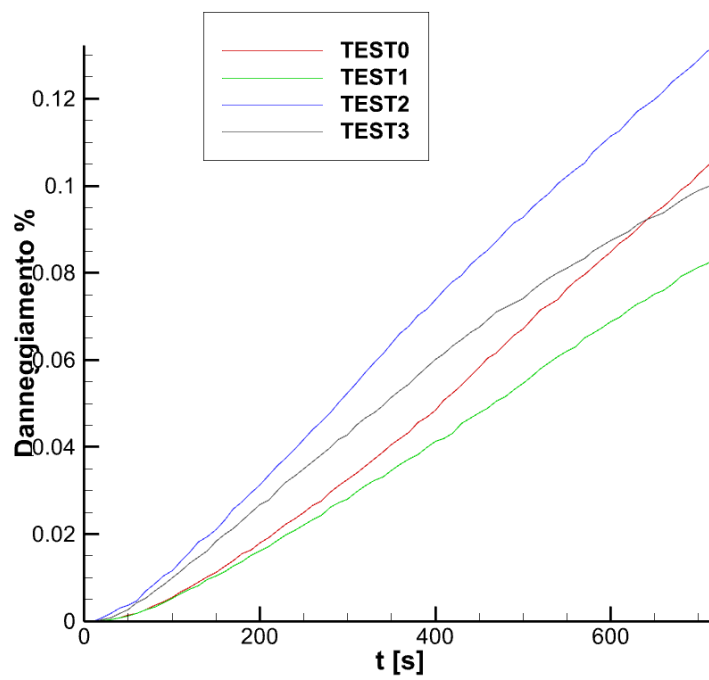


Grafico 3: rappresentazione andamento temporale della % di danneggiamento rispetto dominio totale mediante l'utilizzo dei vari test

4. Risultati

Al termine di queste simulazioni è possibile definire quale protocollo riguardo la modalità di erogazione del voltaggio produce nel modo ottimale l'effetto desiderato. Si ricorda che la necessità di avere un andamento pulsato del voltaggio risiede nella presenza di roll-off. Per la scelta della tipologia di pulsazione sono stati studiati e confrontati differenti articoli a riguardo.

4.1 Verifiche correttezza procedura

Per dimostrare la veridicità e correttezza della nostra procedura sono state compiute due tipologie di verifica: comparando i dati e le procedure effettuate con quelli presenti in letteratura.

Come prima cosa è stato effettuato uno studio per valutare la correttezza dell'implementazione su OpenFOAM del risolutore *bioheatFoam_wb* equiparando la soluzione ottenuta con le soluzioni dell'articolo [9]. Successivamente, il risolutore è stato ampliato considerando anche il problema elettrico, così da ottenere *bioheatFoamV_beta3*, già discusso.

L'articolo [9] propone la risoluzione analitica dell'equazione di scambio del biocalore di Pennes con condizione del flusso di calore sinusoidale sulla superficie della pelle. Il procedimento adoperato è analogo a quello descritto precedentemente. Per la creazione del risolutore *bioheatFoam_wb* ci si è appoggiati al *laplacianFoam* convalidato e ampiamente utilizzato nella piattaforma OpenFOAM, al quale è stato aggiunto il termine di generazione di calore collegato con la perfusione sanguigna. Nell'articolo si esamina la situazione di flusso di calore sinusoidale, la quale è stata tradotta con condizioni al contorno esplicitate con la libreria *broovyBC*. Attraverso vari tentativi e correzioni si è giunti ad una sovrapposizione dei risultati dell'articolo e quelli derivati da *bioheatFoam_wb*, al termine per processo si è potuto affermare la validità del risolutore.

Nell'articolo si vuole risolvere la seguente equazione:

$$\rho_t c_t \frac{\partial T}{\partial t} + W_b c_b (T - T_a) = k \frac{\partial^2 T}{\partial x^2}$$

Dove: ρ_t è la densità del tessuto, c_t il calore specifico del tessuto, T la temperatura del tessuto, t tempo, W_b tasso di perfusione sanguigna, c_b il calore specifico del sangue, T_a temperatura del sangue arterioso, k la conduttività termica del tessuto e x distanza dalla superficie della pelle. Altro non è che l'equazione di Pennes unidimensionale.

Si ha un flusso di calore sinusoidale indotto, il che implica una condizione al contorno così espressa:

$$-k \frac{\partial T}{\partial x} \Big|_{x=0} = q_0 e^{i\omega t}$$

Dove: q_0 flusso di calore sulla superficie della pelle, ω frequenza, i unità numero complesso.

Ora, analizziamo come è stato affrontato il problema con OpenFOAM. Il dominio scelto corrisponde ad un rettangolo, il quale è stato discretizzato ed inserito all'interno della directory *constant* sottoforma della directory *polyMesh*. Attraverso il codice 13 vengono illustrati i margini in cui occorre definire le condizioni al contorno.

```
1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        polyBoundaryMesh;
6     location      "constant/polyMesh";
7     object        boundary;
8 }
9
10 6
11 (
12     bottom
13     {
14         type patch;
15         nFaces 39;
16         startFace 5890;
17     }
18     dx
19     {
20         type patch;
21         nFaces 77;
22         startFace 5929;
23     }
```



```

24     empty
25     {
26         type empty;
27         nFaces 6006;
28         startFace 6006;
29     }
30     q_in
31     {
32         type patch;
33         nFaces 39;
34         startFace 12012;
35     }
36     sx
37     {
38         type patch;
39         nFaces 38;
40         startFace 12051;
41     }
42     top
43     {
44         type patch;
45         nFaces 39;
46         startFace 12089;
47     }
48 )
49

```

Codice 14

Directory: constant/polyMesh File: boundary

Le grandezze termodinamiche presenti nell'equazione che deve essere risolta sono le seguenti. La densità del tessuto è $1050 \left[\frac{kg}{m^3} \right]$, sia la capacità termica specifica del tessuto che del sangue sono $3770 \left[\frac{J \cdot kg}{K} \right]$ e la conducibilità termica del tessuto è $0,5 \left[\frac{W}{m^2 K} \right]$. Inoltre, la temperatura arteriosa è di $37 \text{ }^\circ\text{C}$ e il flusso di calore sulla superficie della pelle vale a $5000 \left[\frac{W}{m^3} \right]$. Questi valori sono inseriti all'interno del file *transportProperties* presente dentro la directory *constant*.

Le condizioni al contorno inserite all'interno del file *T in 0* sono riportate nel Codice 15.

```

1  FoamFile
2  {
3      version      2.0;
4      format       ascii;
5      class        volScalarField;
6      object       T;
7  }
8  // * * * * *
9
10 dimensions      [0 0 0 1 0 0 0];
11
12 internalField    uniform 310.15;
13
14 boundaryField
15 {
16     bottom
17     {
18         type      zeroGradient;
19     }
20
21     dx
22     {
23         type      zeroGradient;
24     }
25
26     q_in
27     {
28         type      groovyBC;
29         variables  "Omega=0.05;lam=0.5;q=5000;gradT=q/(lam);";
30         gradientExpression  "(q/lam)*cos(Omega*time())";
31         fractionExpression  "0";
32         value      uniform 310.15;
33     }
34
35     sx
36     {
37         type      groovyBC;
38         variables  "Omega=0.05;lam=0.5;q=5000;gradT=q/(lam);";
39         gradientExpression  "(q/lam)*cos(Omega*time())";
40         fractionExpression  "0";
41         value      uniform 310.15;
42     }
43
44     top
45     {
46         type      zeroGradient;
47     }
48 }
49
50
51 // *****
52

```

Codice 15
Directory: 0 File: T

La condizione al contorno è definita mediante l'utilizzo della libreria *groovyBC*. In questo caso differentemente dalle condizioni già viste si utilizza *gradientExpression* e *fractionExpression*. Ricordiamo che questi valori sono uniti nel seguente modo:

$$T_f = f \cdot valueExp + (1 - f)(T_{centro} + gradExp \cdot \delta)$$

Quando $f=0$ e $\delta = 0$, si genera $gradientExpression = \frac{T_f - T_c}{\delta} = \frac{\partial T}{\partial x}$; ciò significa che con $f=0$ e $\delta = 0$ impongo il *gradientExpression* pari alla derivata della temperatura rispetto alla variabile x e quindi ho soddisfatto la condizione al contorno del secondo tipo richiesta.

Le direttive del test possono essere visionate all'interno del *controlDict*.

```

1  FoamFile
2  {
3      version      2.0;
4      format       ascii;
5      class        dictionary;
6      location     "system";
7      object       controlDict;
8  }
9  // * * * * *
10
11 application      laplacianFoam;
12
13 startFrom        latestTime;
14
15 startTime        0;
16
17 stopAt           endTime;
18
19 endTime          500;
20
21 deltaT           0.005;
22
23 writeControl      runtime;
24
25 writeInterval     10;

```

```

26
27   purgeWrite      0;
28
29   writeFormat      ascii;
30
31   writePrecision   6;
32
33   writeCompression off;
34
35   timeFormat       general;
36
37   timePrecision    6;
38
39   runTimeModifiable true;
40
41   libs ( "libgroovyBC.so" );
42
43   functions
44   {
45
46       probes
47       {
48           functionObjectLibs ( "libsampling.so" );
49
50           type      probes;
51
52           name      probes;
53
54           fields ( T );
55
56           probeLocations
57           (
58               (0.0  0.0  0.0)           // at top of the helmet
59           );
60       }
61
62   }

```

Codice 16
Directory: system File: controlDict

È evidente la presenza della sonda posta nel punto (0,0,0) della griglia, grazie alla quale riusciamo a ricostruire l'andamento della temperatura in quel punto e paragonarlo con quello generato dalla soluzione analitica dell'articolo. La sovrapposizione è rappresentata nel Grafico 3. Come già accennato questo test è stato risolto con l'utilizzo di un nuovo risolutore implementato appositamente per inserire all'interno del *laplacianFoam* anche il contributo di Pennes, chiamato *bioheatFoam_wb*.

Per capire il funzionamento di *bioheatFoam_wb* occorre esaminare il relativo file *bioheat.cpp*

```

1
2  #include "fvCFD.H"
3  #include "simpleControl.H"
4
5  // * * * * *
6
7  int main(int argc, char *argv[])
8  {
9      #include "setRootCase.H"
10
11      #include "createTime.H"
12      #include "createMesh.H"
13      #include "createFields.H"
14
15      simpleControl simple(mesh);
16
17      // * * * * *
18
19      Info<< "\nCalculating temperature distribution\n" << endl;
20      //-----
21
22      while (simple.loop())
23      {
24          Info<< "Time = " << runTime.timeName() << nl << endl;
25
26          while (simple.correctNonOrthogonal())
27          {
28              forAll(mesh.C(), celli)
29              {
30                  pennes_[celli] = cpb*omega*(Ta0 - T[celli])/(rhot*cpt) ;
31              }
32              solve
33              (
34                  fvm::ddt(T) - fvm::laplacian(DT, T) - pennes_
35              );
36
37              #include "write.H"
38
39              Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
40                  << "   ClockTime = " << runTime.elapsedClockTime() << " s"
41                  << nl << endl;
42          }
43
44          Info<< "End\n" << endl;
45
46          return 0;
47      }
48
49
50  // *****

```

Dal file è ben visibile come il risolutore utilizzato per l'esperimento generale si appoggi su questo appena descritto: *bioheatFoamV_beta3* si ottiene aggiungendo analogamente le considerazioni del contributo elettrico.

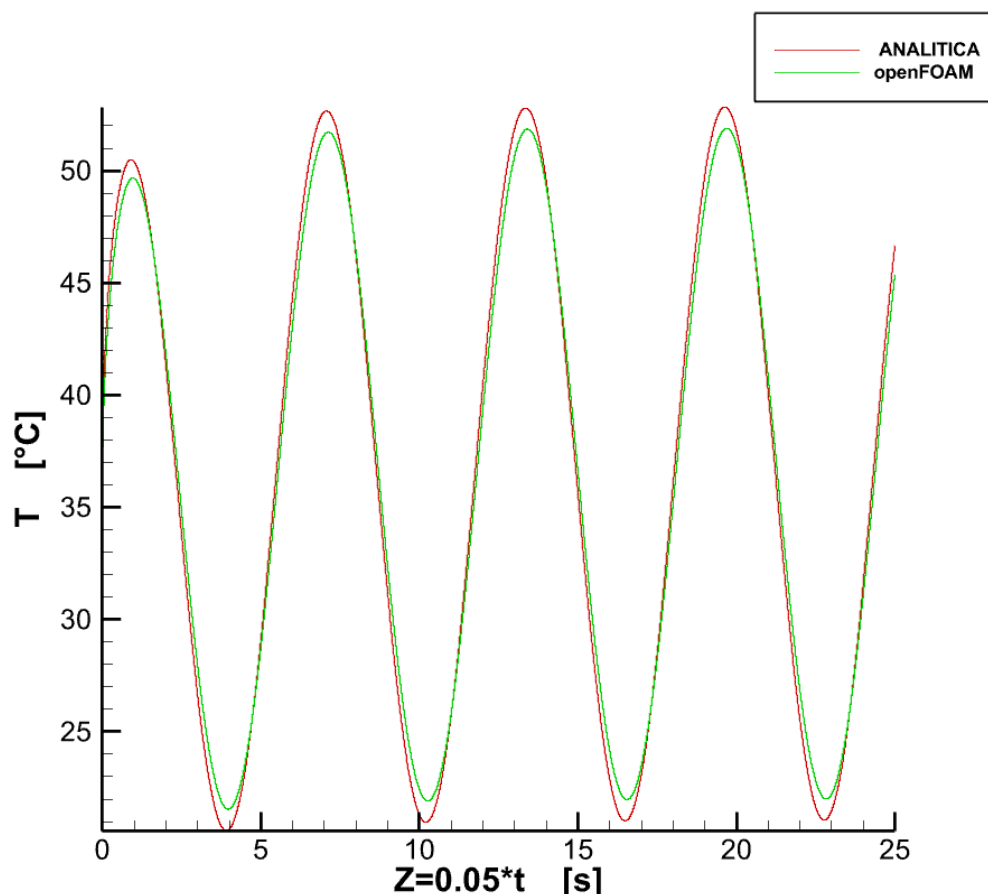


Grafico 4: Sovrapposizione soluzione analitica dell'articolo e risultato mediante OpenFOAM

Visivamente risulta evidente quanto i risultati ottenuti nei due metodi diversi siano approssimabili tra loro. Questo conferma la correttezza del risolutore.

Invece, per verificare la validità del caso appena trattato si compara l'andamento della temperatura in un punto specifico della griglia del domino con l'andamento della temperatura presentata in altri studi con le identiche caratteristiche. Gli studi in cui si hanno analoghe caratteristiche e il luogo della sonda coincide sono stati condotti da Golberg et. Al. e Tucci et. Al. rispettivamente negli articoli [10] e [11]. Vengono comparati gli andamenti della temperatura di una sonda posta a 10 mm dall'elettrodo utilizzando l'equazione (33), ciò è rappresentato nel Grafico 4. Da qui possiamo concludere che il metodo risolutivo è valido in quanto si ottengono valori simili a dei studi precedenti eseguiti in maniera sperimentale in vivo e studi effettuati con software diverso da OpenFOAM. Si nota però che le funzioni non sono completamente sovrapposte ma quella generata dal nostro studio è

caratterizzato da un transitorio che si estingue in maniera più lenta; infatti, il valore dello stato stazionario dopo un tempo maggiore si pone all'incirca sullo stesso valore dei precedenti.

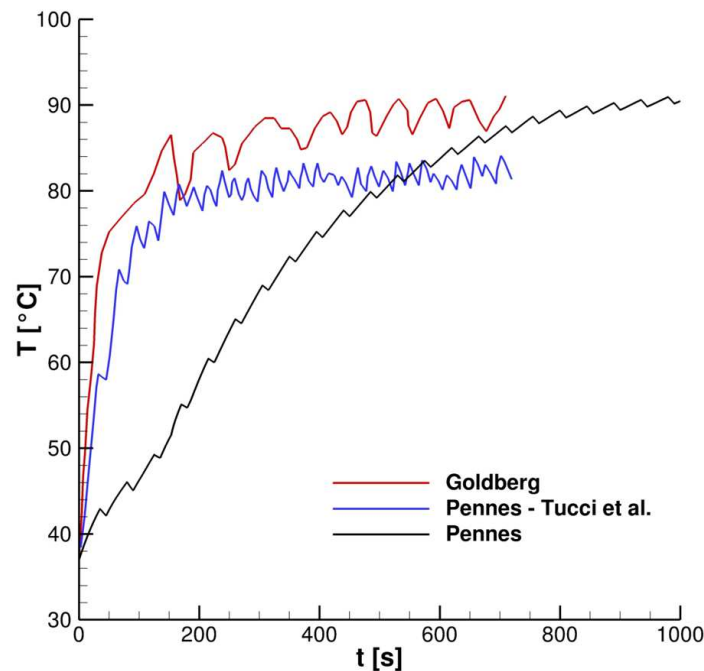


Grafico 5: confronto andamento temperatura

4.2 Considerazioni finali

Tra i nostri test effettuati notiamo che il test che genera una migliore mappa di temperatura con la percentuale di danneggiamento più alta è il *test2*. Questo corrisponde al test in cui abbiamo l'andamento del voltaggio costituito da 16 picchi e una attenuazione nel corso della simulazione del valore massimo del 20%. Un tempo totale di erogazione pari a 560s e un tempo totale senza alcuna erogazione di 160s sui 720s generali. Come è stato già esposto, necessariamente si ha la presenza dei roll off durante l'esecuzione di ablazione termica, grazie a questo lavoro è stato implementato un metodo per erogare la differenza di potenziale. Siamo riusciti a trovare un algoritmo che definisce il criterio con cui far variare il voltaggio che generi delle prestazioni ottimali della terapia.

Notiamo che il risultato da noi ottenuto è in buon accordo con l'andamento che era stato individuato da Tucci et. Al. [10]

Conclusioni

Lo scopo di questo lavoro era confrontare vari modelli di somministrazione di voltaggio per l'ablazione a radiofrequenza (RFA) di tessuto epatico in vivo utilizzando un elettrodo raffreddato. Il nostro obiettivo era quello di suggerire il protocollo migliore riuscendo a predire la zona di coagulazione nell'ablazione del tumore RF grazie al software OpenFOAM. Abbiamo raggiunto l'obiettivo descrivendo i periodi di erogazione e non del voltaggio insieme l'ampiezza del voltaggio stesso mediante formulazioni matematiche del caso che genera risultati migliori. Lo studio è stato condotto implementando un livello di alta tensione al fine di considerare i casi di roll-off. Inoltre, i risultati ottenuti al computer sono stati confrontati con i pochi lavori sperimentali in vivo disponibili in letteratura e si è riscontrato essere in buon accordo. Nonostante la scarsità di studi in vivo e la necessità di ulteriori studi sperimentali, il modello informatico sviluppato e l'implementazione di questo risolutore potrebbe essere preso in considerazione in futuro per continuare a migliorare i modelli medici protocolli e dispositivi. Il grande vantaggio di questo studio risiede nel fatto che è stato creato un modello risolutivo capace di predire in maniera per nulla invasiva un campo di temperatura in vivo.

Nel seguente studio sono presenti delle limitazioni che possono essere superate in approfondimenti futuri. Si possono effettuare miglioramenti focalizzati sullo sviluppo di nuovi modelli considerando anche le differenze di conducibilità elettrica tra tumore e tessuti sani. Si possono ideare diverse metodologie per l'erogazione del voltaggio dipendentemente dalla resistenza assunta dal tessuto biologico. Tutto ciò contribuirebbe allo sviluppo di modelli ancora più realistici riguardo specifici organi e, quindi, anche per altre malattie.

Sitografia

- [1] = <https://www.airc.it/>
- [5] = <https://www.treccani.it/enciclopedia/calore>
- [6] = <https://www.openfoam.com/>
- [7] = <https://conoscerelinux.org/wp-content/uploads/2016/06/openfoam-introduzione-corso-conoscerelinux-modena-2016.pdf>

Bibliografia

- [2]= Computer modelling of an impedance-controlled pulsing protocol for RF tumour ablation with a cooled electrode. Macarena Trujillo, Jose Bon,María José Rivera, Fernando Burdío & Enrique Berjano
- [3]= Numerical analysis of the relationship between the area of target tissue necrosis and the size of target tissue in liver tumours with pulsed radiofrequency ablation
- [4]= H.H. Pennes. Analysis of tissue and arterial blood temperatures in the resting forearm. J. Appl. Physiol., 93, 1948.
- [8] = Abraham, J. P. & Sparrow, E. M. A thermal-ablation bioheat model including liquid-to-vapor phase change, pressure- and necrosis dependent perfusion, and moisture-dependent properties. Int. J. Heat. Mass Transf. 50, 2537–2544 (2007).
- [9] = Analytical analysis of the Pennes bioheat transfer equation with sinusoidal heat flux condition on skin surface by Tzu-Ching Shih a, Ping Yuan b, Win-Li Lin c,d, Hong-Sen Kou
- [10] = Pennes'bioheat equation vs. porous media approach in computer modeling of radiofrequency tumor ablation by Tucci et al.
- [11] = Percutaneous Radiofrequency Tissue Ablation: Optimization of Pulsed Radiofrequency Technique to Increase Coagulation Necrosis by Golberg et al.
- [12]= Patz T, Kroger T, Preusser T, Simulation of radiofrequency ablation including water evaporation. IFMBE Proc 2009;25/IV:1287–90
- [13]= Chang, I. A. Considerations for thermal injury analysis for RF ablation devices. Biomed. Eng. Online 4, 3–12 (2010)
- [14]= F. Moukalled, L. Mangani e M. Darwish, The Finite Volume Method in Computational Fluid Dynamics, Springer, Cham, 2016.