

Regione Toscana  
Accordo di Programma Quadro Ricerca e trasferimento tecnologico per il sistema produttivo  
III Accordo integrativo



## Rapporto finale WP5

*Pacco di lavoro:* WP.5 – Validazione dei modelli

*A cura di:* CINIGeo – Consorzio Interuniversitario Nazionale per l'Ingegneria delle  
Georisorse – Università di Bologna; Responsabile scientifico Prof. Ing.  
Paolo Berry  
DSI – Dipartimento di Sistemi e Informatica – Università di Firenze;  
Responsabile scientifico Prof. Fabio Schoen

*Data compilazione:* Mese

## Indice

<b>1. Introduzione .....</b>	<b>1</b>
<b>2. Simulazione inversa di bacini geotermici .....</b>	<b>1</b>
2.1. Simulazione inversa: procedura .....	3
<b>3. Codice iTOUGH2.....</b>	<b>4</b>
3.1. Utilizzo di iTOUGH2 .....	5
3.2. File di input per iTOUGH2 .....	5
3.3. File di output di iTOUGH2 .....	6
<b>4. Modelli simulati .....</b>	<b>6</b>
4.1. Stato Naturale del Bacino geotermico di Larderello-Travale .....	7
4.1.1 Modello a colonna verticale .....	7
4.1.2 Stato Naturale del Bacino di Larderello-Travale - griglia regolare ....	9
4.1.3 Stato Naturale del Bacino di Larderello-Travale - griglia irregolare	15
4.2 Modello 3x3x5 .....	21
<b>5. Parallel Virtual Machine (PVM).....</b>	<b>11</b>
5.1. Definizioni .....	11
5.2. Caratteristiche .....	11
5.3. Primitive .....	12
5.4. Architettura e sincronizzazione .....	12
5.5. Cluster installato da CINIGeo .....	13
<b>6. iTOUGH2-PVM .....</b>	<b>14</b>
6.1. Architettura di iTOUGH2-PVM.....	15
6.2. Utilizzo di iTOUGH2-PVM.....	15
6.3. File di input per iTOUGH2-PVM.....	15
6.4. File di output di iTOUGH2-PVM.....	15
<b>7. Algoritmo di Inversione implementato in RBFOPT .....</b>	<b>16</b>
7.1. Sviluppo Software .....	17
7.2. Test numerici effettuati.....	19
7.3. Sviluppi Futuri .....	25
<b>8. Riferimenti Bibliografici .....</b>	<b>26</b>

## 1. Introduzione

Questo documento espone le attività svolte nell'ambito del WP5.

Verranno introdotti alcuni concetti relativi alla simulazione diretta e inversa dei bacini geotermici, una breve descrizione del codice di simulazione utilizzato, le implementazioni proposte per ottimizzare il codice e i modelli numerici sviluppati per testare il software implementato.

Hanno partecipato ai lavori di ricerca del WP5 ed alla redazione di questo documento:

Prof. Ing. Paolo Berry CINIGeo

Prof. Ing. Guido Gottardi CINIGeo

Ing. Alfredo Battistelli CINIGeo

Prof. Dott. Villiam Bortolotti CINIGeo

Ing. Carlo Cormio CINIGeo

Ing. Alberto Diolaiti CINIGeo

Ing. Stefano Bonduà CINIGeo

Dott. Geol. Elena Lognoli CINIGeo

Ing. Nicola Frignani CINIGeo

Prof. Fabio Schoen DSI

Ing. Andrea Cassioli DSI

## 2. Simulazione inversa di bacini geotermici

La valutazione della risposta di un sistema fisico, come ad esempio un bacino geotermico, ad una perturbazione esterna (in questo caso l'attività estrattiva operata dall'uomo) attraverso la simulazione del suo modello, è definito come problema "*diretto*" ("*forward*"). Le proprietà del modello indipendenti dalla tipologia di perturbazione simulata sono definite *parametri* del sistema, mentre le proprietà che variano nel tempo vengono chiamate "*variabili*".

L'obiettivo di uno studio di bacino é la realizzazione di un modello numerico, ovvero una rappresentazione numerica del sistema fisico, che sia in grado di prevedere nel tempo il comportamento del sistema fisico reale che rappresenta, in funzione delle diverse ipotesi di sviluppo.

Il primo passo della procedura consiste nella creazione del modello numerico a partire da tutte le informazioni, di tipo geologico, geofisico, idrogeologico, petrofisico, ingegneristico, disponibili. Questo insieme di conoscenze permette di simulare uno stato iniziale del

serbatoio anteriore alla messa in produzione, di simulare la storia produttiva passata e considerare i futuri scenari di sviluppo.

Nel caso di un bacino geotermico, situato a qualche migliaio di metri di profondità, le informazioni a disposizione consentono di conoscere con un ragionevole dettaglio solo una parte estremamente limitata del dominio indagato, in genere nelle vicinanze dei pozzi. Le inevitabili approssimazioni nella definizione del modello numerico dovuta ai valori che si assegnano ai parametri, ha un impatto notevole sull'attendibilità delle previsioni che derivano dall'uso modello.

In presenza di una storia produttiva, la capacità del modello di simulare correttamente il passato noto rappresenta un requisito fondamentale affinché le simulazioni di scenari futuri possano essere considerate attendibili. Il processo che porta alla convalida del modello attraverso il confronto tra i risultati della simulazione numerica e i dati di produzione noti è detto *"History Matching"*. In tale fase il modello numerico viene utilizzato confrontando il risultato delle simulazioni con dati misurati in posizione e coordinate note dette *punti di calibrazione*. L'obiettivo è di ottenere, attraverso opportune modifiche dei parametri, un modello numerico del bacino che riproduca i profili di produzione noti. Se il problema della previsione del comportamento del reservoir nei confronti di una perturbazione esterna è stato indicato con il termine di problema diretto, la determinazione di parametri che consenta di riprodurre il comportamento reale, fissate nel tempo le condizioni operative, è definito problema *"inverso"*.

Il confronto tra dati simulati e dati misurati è condotto attraverso la definizione di una opportuna *"funzione obiettivo"*, che sintetizzi in un unico numero lo scarto fra i dati calcolati e osservati. Nel caso in cui l'accordo fra il calcolo e l'osservazione non sia soddisfacente, il modello viene modificato agendo su un opportuno insieme di parametri. La scelta di questi parametri e l'entità delle relative modifiche sono definite attraverso un procedimento per tentativi in cui il fattore umano, ovvero l'esperienza di chi conduce lo studio, gioca un ruolo determinante.

La procedura prosegue iterativamente fino alla determinazione di un set di parametri che minimizza la funzione obiettivo.

L'History Matching, ovvero la soluzione del problema inverso così formulato, rappresenta quindi un momento chiave dello studio di giacimento poiché, come già detto precedentemente, condiziona pesantemente le previsioni. Le maggiori difficoltà di questo procedimento nascono dalla natura stessa del problema inverso, ovvero dalla dipendenza della risposta del sistema dai parametri di modello. In generale, la relazione risposta – parametri, definita attraverso la simulazione numerica, ovvero la soluzione di un

problema diretto, è non lineare. Conseguentemente il problema inverso è a priori mal condizionato, ovvero più configurazioni dei parametri sono in grado di fornire risposte molto simili, di fatto equivalenti; nonostante si scelga accuratamente un set di parametri rappresentativo per l'intero giacimento, l'History Matching rappresenta un problema matematico di ottimizzazione multidimensionale, caratterizzato da una funzione obiettivo estremamente complessa, non lineare e, nella stragrande maggioranza dei casi, non convessa. L'insieme di tutti questi fattori fa sì che l'unicità della soluzione al problema venga meno: si ottiene cioè una funzione obiettivo multimodale, così definita poiché presenta un numero elevato, in genere ignoto, di minimi locali isolati. In altre parole, il modello validato ottenuto attraverso il processo di History Matching, non è necessariamente unico e possono esistere diverse distribuzioni spaziali di parametri che riproducono, entro i limiti degli errori da cui sono affette le misure di campo, la storia passata del serbatoio. È dunque buona pratica utilizzare dei metodi di ottimizzazione matematica che permettano di localizzare più minimi della funzione oggetto.

L'incertezza su alcuni parametri del serbatoio, la presenza e ubicazione di barriere impermeabili, comportano inoltre la necessità di valutare, mediante simulazioni, l'influenza che ipotesi diverse su tali parametri hanno sulla distribuzione delle grandezze simulate (*“studi di sensibilità”* – *“sensitivity studies”*). Questi studi di sensibilità sull'influenza dei parametri dotati di minor precisione conoscitiva, forniscono un quadro più completo del comportamento del serbatoio.

## **2.1. Simulazione inversa: procedura**

Riassumendo, la procedura di simulazione inversa prevede le seguenti operazioni:

- 1) creazione del modello che simula la fase di coltivazione del bacino oggetto di studio;
- 2) scelta dei parametri da stimare;
- 3) scelta dei valori iniziali dei parametri da stimare;
- 4) identificazioni dei valori misurati nei punti di calibrazione;
- 5) simulazione del sistema;
- 6) confronto tra i valori simulati e i valori misurati mediante la definizione della funzione obiettivo;
- 7) modifica dei parametri da stimare;
- 8) ripetizione iterativa dei passaggi da 5) a 7) fino ad ottenere il set di valori dei parametri che minimizza la funzione obiettivo;
- 9) analisi di sensitività.

### 3. Codice iTOUGH2

In bacini geotermici caratterizzati da dimensioni medio grandi, quali quelli di interesse per il presente progetto, la corretta caratterizzazione dei parametri petrofisici (porosità, permeabilità, ecc.) comporta onerose (in termini di tempo di calcolo) e complesse operazioni di sintonizzazione del modello mediante tecniche inverse (history matching). Questa considerazione comporta che il simulatore sia dotato di strumenti per gestire in modo automatico il problema inverso e suggerisce l'utilizzazione di un codice che sia predisposto a lavorare in modo parallelo.

Si è scelto di utilizzare il codice iTOUGH2 in considerazione delle seguenti caratteristiche:

- è il software di simulazione geotermica maggiormente impiegato dalla comunità scientifica negli ultimi 20 anni (oltre il 50 % delle simulazioni);
- ha un'ampia comunità di ricercatori presso l'Università di Berkeley, dove il sw è stato sviluppato, che garantisce la manutenzione e ne cura lo sviluppo, e di ricercatori italiani che hanno collaborato all'implementazione di alcuni suoi moduli;
- è un sw open source ben documentato, per cui le modifiche dedicate risultano, contestualmente con la difficoltà intrinseca della materia, relativamente facili da realizzare;
- esiste una versione specifica, iTOUGH2-PVM, che implementa diversi algoritmi di inversione ed è già predisposta per il calcolo parallelo mediante l'utilizzo della libreria PVM, libreria che è open source e free;
- è scritto in Fortran e quindi può essere utilizzato con più sistemi operativi;
- adotta una struttura modulare molto flessibile che permette di adattarlo a nuovi sistemi/scenari modificando lo specifico modulo EOS che si occupa del bilancio termodinamico senza dover intervenire sul codice restante;
- ha un costo di licenza assolutamente competitivo se paragonato ad equivalenti programmi in commercio.

iTOUGH2 integra come subroutine TOUGH2 e questo permette di eseguire, sullo stesso modello, sia le simulazioni in forward mode, che le simulazioni inverse. La struttura a moduli "EOS" di iTOUGH2 è la medesima di TOUGH2.

Per lo studio dei bacini geotermici, sono stati individuati e utilizzati i moduli "EOS1" ed "EWASG":

- EOS1 permette di gestire sistemi monocomponente bifasici;

- EWASG (Water, Salt, Gas) descrive il comportamento di bacini geotermici caratterizzati da fluidi salini e gas non condensabili.

Si è utilizzato inoltre il modulo EOS3 (acqua e aria) per testare il software attraverso dei modelli numerici di esempio forniti assieme al codice sorgente.

Il processo di stima dei parametri può richiedere tempi di calcolo molto lunghi a causa del numero, anche elevato, di simulazioni necessarie per risolvere il problema diretto per ciascuna differente combinazione dei valori dei parametri. Queste simulazioni possono essere eseguite in parallelo, ovvero contemporaneamente, per il fatto che risultano indipendenti una dall'altra, riducendo così i tempi di calcolo. Tra le varie versioni disponibili di iTOUGH2, CINIGeo ha acquistato la versione iTOUGH2-PVM che implementa il codice con il sistema Parallel Virtual Machine (PVM) e permette la simulazione inversa, in parallelo, su un cluster eterogeneo di workstation Unix.

La versione sequenziale del software, a differenza della versione parallela che funziona solamente con S.O. Linux, può essere utilizzata sia in Linux che in Windows.

### 3.1. Utilizzo di iTOUGH2

Per poter utilizzare iTOUGH2 é necessario innanzitutto settare correttamente i parametri contenuti nel file *maxsize.inc*, fornito con il codice sorgente, in maniera da rendere usufruibile il codice per le applicazioni di nostro interesse. Una volta modificato tale file, è necessario compilare il codice, con le dovute opzioni per Linux e per Windows, per ciascun modulo (EOS1, EOS3, EWASG).

### 3.2. File di input per iTOUGH2

Una simulazione mediante iTOUGH2 necessita di file di input specifici di TOUGH2 e file di input specifici di iTOUGH2. Le informazioni contenute nei file di input di TOUGH2 sono indispensabili per la definizione del modello numerico su cui eseguire la calibrazione.

Il pacchetto di informazioni necessarie per eseguire una simulazione è costituito da diversi file in formato ASCII:

- **inomefile.inp**: file di input specifico di iTOUGH2; contiene varie informazioni, tra cui i parametri da stimare, l'algoritmo di minimizzazione scelto;
- **nomefile.dat**: file di input specifico di iTOUGH2; contiene i valori delle grandezze misurate nei punti di calibrazione;
- **nomefile.inp**: file di input specifico di TOUGH2; contiene le informazioni sui materiali (fluidi e rocce) e sulle loro condizioni iniziali, i parametri di simulazione

(intervallo di tempo simulato, criteri di convergenza, etc) per le simulazioni in avanti;

- **nomefile.gen**: contiene le informazioni relative ai blocchi tra i quali avvengono scambi di energia e/o di massa;
- **nomefile.mes**: contiene le informazioni relative alla geometria dei blocchi con i quali è suddiviso il dominio indagato;
- **nomefile.ini**: contiene le condizioni iniziali per determinati blocchi.

I nomi dei file sopracitati si riferiscono alla versione di iTOUGH2/TOUGH2 per Linux. Il simulatore funziona alla stessa maniera anche nel S.O. Windows: alcuni file, pur avendo la medesima struttura e il medesimo formato, hanno nomi differenti (Linux→Windows):

- **nomefile.mes**→**MESH**
- **nomefile.gen**→**GENER**
- **nomefile.ini**→**INCON**

### 3.3. File di output di iTOUGH2

Al termine della simulazione, che può avere durata anche dell'ordine di qualche ora, iTOUGH2 crea vari file in formato ASCII (per una descrizione completa si rimanda al manuale di iTOUGH2) tra cui: un file denominato **inomefile.out** (stesso nome sia per Linux che per Windows) che contiene un riassunto del processo di minimizzazione della funzione obiettivo e dell'analisi di sensitività.

## 4. Modelli simulati

Le conoscenze acquisite durante l'ultimo secolo di studio, esplorazione e sfruttamento del bacino di Larderello-Travale hanno dimostrato che si tratta di un sistema geotermico a vapore dominante nella maggior parte delle aree e profondità esplorate.

I sistemi geotermici a vapore dominante o altrimenti detti “ a vapore secco”, come The Geysers (California) e Larderello presentano temperature e pressioni del reservoir piuttosto uniformi, e le relative distribuzioni verticali, insieme ai profili di saturazioni e composizione delle fasi, mostrano trend di variazione caratteristici.

Nello stato naturale, cui non sono forniti disturbi esterni dovuti all'attività antropica di estrazione e sfruttamento, il trasferimento di calore avviene prevalentemente in



verticale, mentre le componenti orizzontali di trasporto assumono rilevanza solo nelle aree marginali del bacino in associazione ad elevati gradienti orizzontali di temperatura. Il trasferimento di calore lungo la verticale avviene globalmente sia per convezione che per conduzione; in particolare, il trasferimento di calore per convezione è associato ad un contro-flusso di liquido e vapore.

Fin dai primi anni Settanta, gli studi inerenti i bacini al alta entalpia concordano nel considerare che lo stato naturale dei sistemi geotermici a vapore dominante è costituito infatti da flussi ascendenti di vapore e discendenti di condensati che costituiscono una “*heat pipe*”. Con questo termine si indica un meccanismo che porta un elevato trasferimento di calore dovuto ad un sistema accoppiato di correnti di flusso: il calore di condensazione che risale dai serbatoi verso la copertura viene rilasciato dal sistema dopo aver attraversato la copertura di rocce impermeabili, che sigillano il bacino; il calore viene trasferito al caprock per conduzione, mentre i fluidi condensati ridiscendono verso il basso in fase liquida, andando ad alimentare il serbatoio. Nello stato naturale, inoltre, il sistema è in uno stato semi-stazionario per il flusso di fluido e di calore: il bilancio globale di flusso del sistema è considerato essere quasi nullo perché il contro-flusso di liquido e vapore è quasi all’equilibrio.

La calibrazione del modello di stato naturale è stata quindi elaborata per riprodurre le distribuzioni di parametri tipiche dei sistemi geotermici a vapore dominante.

## **4.1. Stato Naturale del bacino geotermico di Larderello-Travale**

La simulazione dello stato naturale del bacino geotermico di Larderello-Travale è stata sviluppata con un procedimento di simulazione articolato in due fasi successive.

Nella prima fase di elaborazione è stato sviluppato un modello a colonna verticale in cui si sono ottenute le condizioni tipiche di un sistema geotermico a vapore dominante in mezzo poroso. Le distribuzioni verticali di pressione e temperatura così ottenute sono state quindi associate al modello 3D, sul quale è stata condotta una simulazione di circa 20650 anni.

### **4.1.1. Modello a colonna verticale**

Nel primo passaggio della realizzazione dello stato naturale di Larderello-Travale è stato realizzato un modello a colonna verticale, con dominio come da relazione WP4.

Descrizione della Griglia di discretizzazione spaziale

La discretizzazione del dominio, date le sue caratteristiche geometriche, è stata operata con griglia regolare costituita da 36 blocchi in direzione z (profondità) di 100 x 100 x 100 m ciascuno come riassunto nel seguente schema:

- area interessata: 100x100 m
- strato di copertura: da +500 a 0 m s.l.m.
- reservoir shallow: da 0 a -1500 m s.l.m.
- reservoir deep: da -1500 a -3000 m s.l.m

Al modello sono stati assegnati i seguenti materiali, dall'alto verso il basso:

- ATMOS: atmosfera; layers 1
- CAPRO: strato di copertura; layers 2 – 6
- SHALL: serbatoio superficiale (reservoir shallow); layers 7 – 21
- DEEP: serbatoio profondo (reservoir deep); layers 22 – 36

Il file di input di TOUGH2 “nomefile.mes” relativo alla griglia di simulazione sopra illustrata è stato quindi modificato dall'operatore esperto per introdurre specifiche condizioni al contorno da associare al layer rappresentativo dell'atmosfera (layer 1; materiale identificativo: ATMOS).

Infatti, per rappresentare correttamente i parametri specifici dell'atmosfera, è necessario associare ai blocchi relativi le condizioni al contorno di tipo Dirichlet, che impongono un valore costante alle variabili dipendenti, quindi condizioni termodinamiche fisse, al bordo del dominio di simulazione.

Operativamente, è necessario modificare il file “nomefile.mes”, che contiene le caratteristiche geometriche della griglia di simulazione, nel modo seguente:

- Nella keyword ELEM (prima parte del file .mes) viene modificato il volume dei blocchi associati al materiale ATMOS con un valore maggiore od uguale a  $10^{50} \text{ m}^3$ . L'assegnazione di volumi enormi ai blocchi che si trovano ai bordi del dominio è necessaria per rendere invariabili le loro condizioni termodinamiche in risposta al cambiamento dei flussi all'interno del sistema.
- Nella keyword CONN (seconda parte del file .mes) si assegna un valore significativamente piccolo alla distanza del nodo dei blocchi esterni rispetto a quella del blocco più interno, in modo che tali condizioni si localizzino in prossimità dell'interfaccia tra il modello fisico e l'atmosfera.

Descrizione del modello numerico: parametri di simulazione

I diversi materiali sono associati al modello numerico tramite la keyword ROCKS del file di input del software TOUGH2, e per la realizzazione dello stato naturale sono stati utilizzati valori concordati con Saipem e riferiti a studi pregressi (Pruess *et al.*, 1987).

Le condizioni iniziali del modello sono state introdotte con la keyword INDOM, associate ad ogni materiale. La simulazione è stata elaborata associando ai livelli rappresentativi di litotipi relativamente impermeabili (materiale identificativo: CAPRO) condizioni di fluido monofase, mentre i livelli che costituiscono il vero e proprio “serbatoio” (materiali identificativi: SHALL e DEEP) sono stati introdotti con condizioni iniziali di fluido bifasico, con CO<sub>2</sub> e “brine”. I due serbatoi si differenziano sia nei parametri petrofisici introdotti nel blocco ROCKS, che per condizioni iniziali (blocco INDOM) distinte nei valori di temperatura e saturazione gas.

La simulazione è stata impostata per un tempo di circa 20650 anni.

#### **4.1.2. Stato Naturale del bacino di Larderello-Travale – griglia regolare**

La simulazione sopra descritta, verificata la corretta risoluzione ed attendibilità dei risultati, è stata estesa ad un modello tridimensionale.

Descrizione della Griglia di discretizzazione spaziale

Estremi del dominio di simulazione:

- area interessata: 5 \* 5 km
- strato di copertura: da +500 a 0 m s.l.m.
- reservoir shallow: da 0 a -1500 m s.l.m.
- reservoir deep: da -1500 a -3000 m s.l.m

Il modello, essendo l'estensione del precedente, ha la stessa distribuzione di materiali in profondità:

- ATMOS: atmosfera; layers 1
- CAPRO: strato di copertura; layers 2 – 6
- SHALL: serbatoio superficiale (reservoir shallow); layers 7 – 21
- DEEP: serbatoio profondo (reservoir deep); layers 22 – 36

Il dominio è stato discretizzato mediante griglia regolare di 10x10x36 blocchi per un totale di 3600 blocchi (**Figura 4.1. Griglia regolare per il bacino di Larderello**) realizzata sia processando la keyword MESHMAKER, strumento dedicato di TOUGH2,

che utilizzando la procedura GRASS+AMESH, come illustrato nella precedente relazione WP4..

01	02	03	04	05	06	07	08	09	100
081	082	083	084	085	086	087	088	089	090
071	072	073	074	075	076	077	078	079	080
061	062	063	064	065	066	067	068	069	070
051	052	053	054	055	056	057	058	059	060
041	042	043	044	045	046	047	048	049	050
031	032	033	034	035	036	037	038	039	040
021	022	023	024	025	026	027	028	029	030
011	012	013	014	015	016	017	018	019	020
01	02	03	04	05	06	07	08	09	010

*Figura 4.1. Griglia regolare per il bacino di Larderello-Travale*

#### Descrizione del modello numerico

Il modello geometrico così ottenuto è stato popolato con i parametri utilizzati nel modello a colonna verticale 1D.

Al livello 36, attribuito al materiale DEEP, è stata associata una produzione di calore costante in modo da simulare l'intrusione magmatica che ha originato l'anomalia termica della zona di studio, e che ha innescato la formazione del bacino geotermico.

I risultati della simulazione sono stati analizzati diagrammando i valori dei parametri caratteristici e valutandone i trend ad un tempo di simulazione di circa 20650.

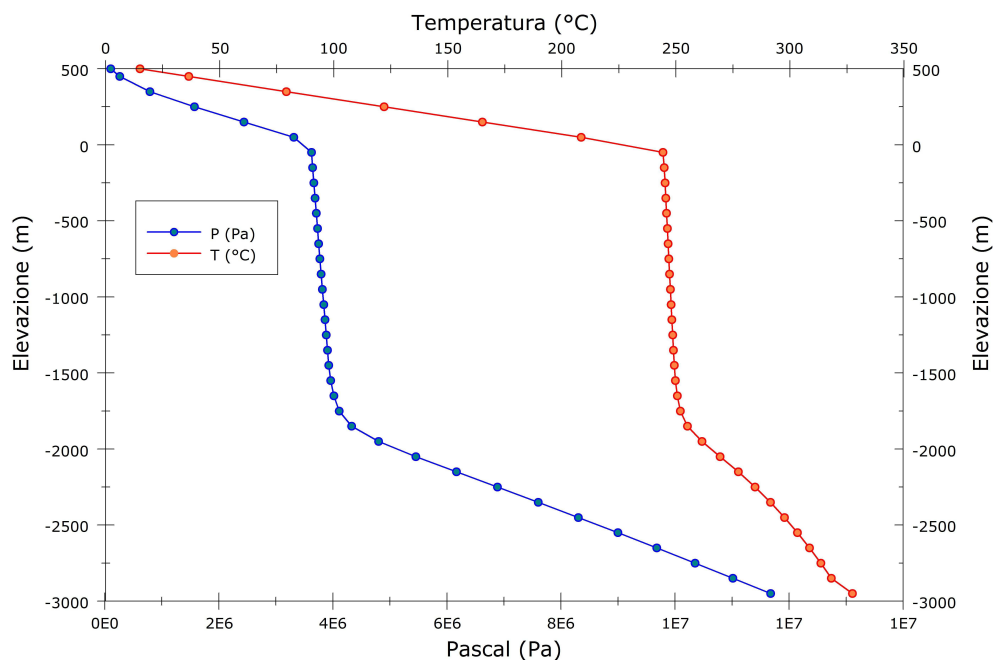


Figura 4.2. Profilo di temperatura e pressione – 20650 anni

In figura 4.2 sono riportati i profili caratteristici di temperatura e pressione per una generica colonna del modello simulato. Nel grafico si possono individuare gli andamenti lineari dello strato di copertura impermeabile, il trend tipico delle condizioni vapor statiche che caratterizzano il serbatoio superficiale e le condizioni liquido-statiche associate al serbatoio profondo.

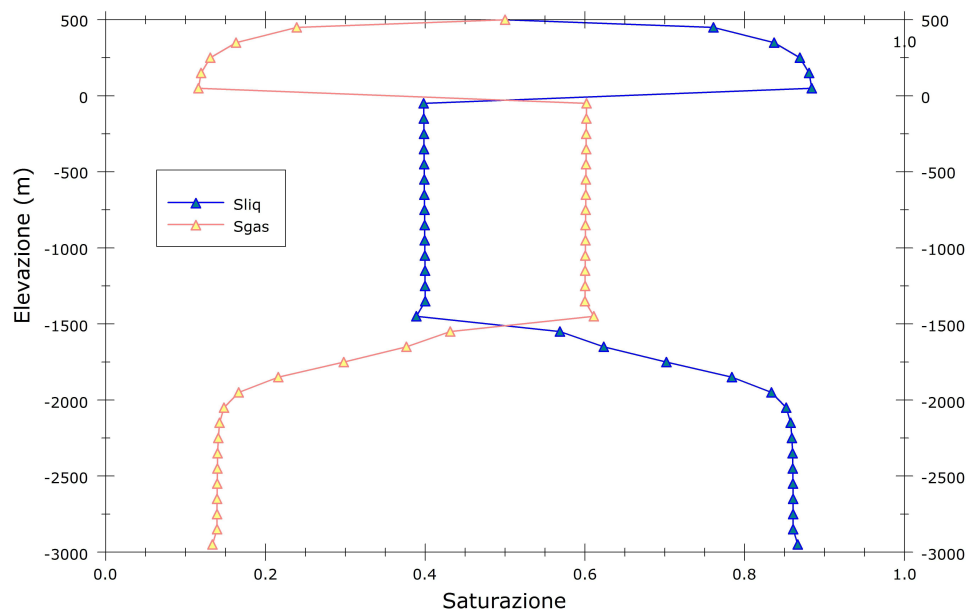


Figura 4.3 Saturazione gas e liquido – 20650 anni

I trend delle saturazioni liquida e gassosa mostrano come nello stato naturale si siano raggiunte e persistano le condizioni bifasiche lungo tutto il profilo. L'andamento è congruente con l'evoluzione dello stato naturale di un sistema di questo tipo: le condizioni monofasiche degli strati di copertura si evolvono in bifasiche e consentono il rilascio dei fluidi in atmosfera, contribuendo all'instaurarsi del meccanismo di *heat pipe*.

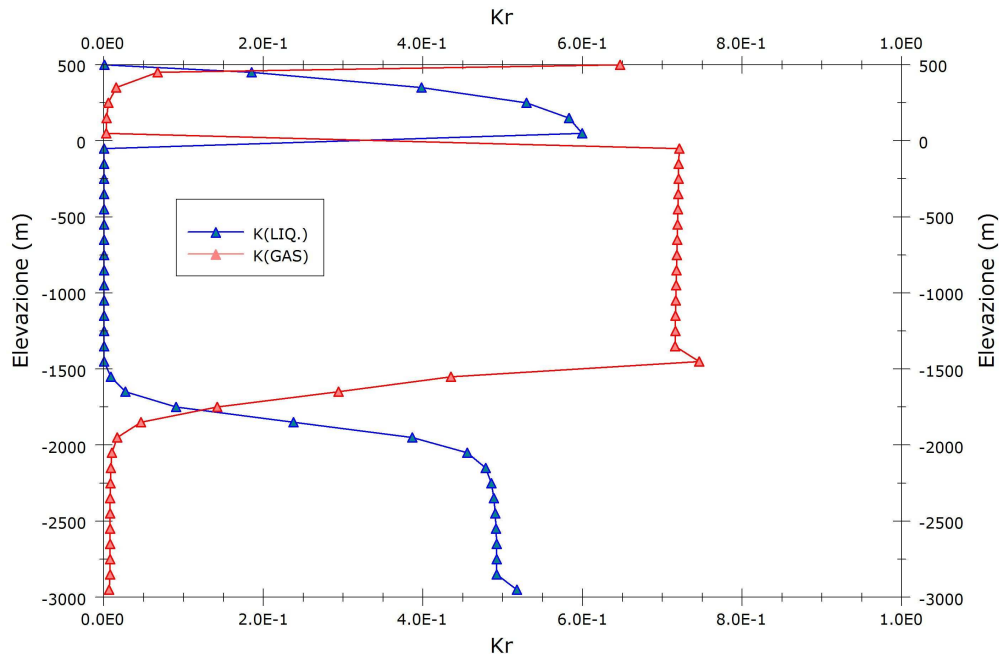


Figura 4.4 Permeabilità relativa gassosa e liquida – 20650 anni

Il diagramma in cui sono riportati i valori delle permeabilità relative, liquida e gassosa, (Fig. 4.3) evidenzia la bassissima permeabilità liquida nel serbatoio superficiale, in condizioni vapor-statiche. La permeabilità è comunque sufficiente ad alimentare il meccanismo della *heat-pipe* e consente quindi alla fase acquosa di fluire verso il basso. Gli andamenti riflettono il passaggio tra i serbatoi e gli strati di copertura; in particolare, il passaggio dal serbatoio superficiale agli stati impermeabili è marcato dalla marcata diminuzione della permeabilità relativa della fase gassosa, prossima allo zero, che risale gradualmente avvicinandosi alla superficie, interfaccia bacino/atmosfera.

I valori delle componenti del flusso indicano che nel sistema è stato raggiunto uno stato semi-stazionario: la componente del flusso di fase gassosa, verso l'alto, viene infatti bilanciato con quasi totale corrispondenza dal flusso della fase liquida verso il basso, alimentando il contro-flusso di liquido vapore con bilancio globale praticamente nullo (Fig. 4.5)

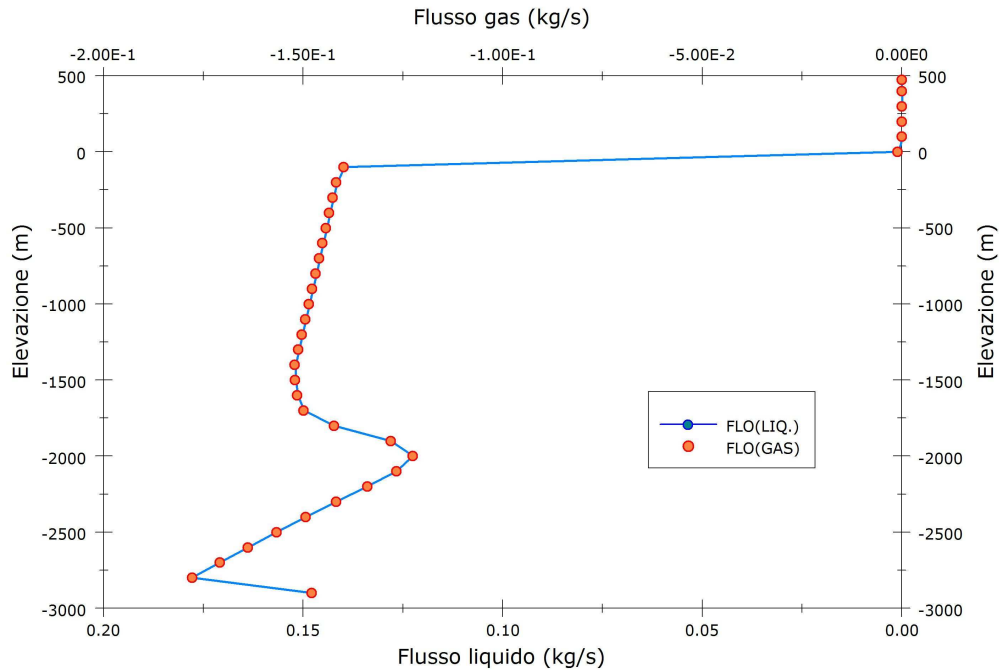


Figura 4.5 Profilo flusso gas e liquido – 20650 anni

In figura 4.6 è riportato il diagramma il profilo P-T di confronto con la curva di saturazione dell'acqua. E' evidente la corrispondenza tra le due curve per tutto lo sviluppo del grafico a parte le estremità. Nella parte superiore del diagramma, che corrisponde al dominio P-T degli strati di copertura, il modello si discosta dalla curva dell'acqua a causa delle alte concentrazioni in  $\text{CO}_2$  che corrispondono a tali livelli. Infatti, l'anidride carbonica tende ad accumularsi negli strati superficiali a bassa permeabilità poiché, portata verso la superficie dal meccanismo della heat pipe, la fase gassosa attraversa strati caratterizzati da valori di permeabilità relativa della fase gas prossima allo zero (McKibbin e Pruess, 1989).



Un'ulteriore deviazione dal trend della curva di saturazione dell'acqua, seppur di minore entità, si evidenzia invece nell'angolo opposto, individuato dai valori P-T relativi al fondo del bacino in simulazione; l'anomalia in questo caso è da ricondurre all'accumulo di sali di NaCl negli ultimi livelli del sistema. La precipitazione nel settore a comportamento liquido-statico porta infatti all'aumento della pressione di vapore della fase acquosa; l'accumulo di sali al fondo è da ricondurre all'azione di dilavamento della condensa su tutto lo spessore del bacino e al successivo trasporto negli strati inferiori grazie ai flussi discendenti della *heat-pipe* (Shoot, 1995).

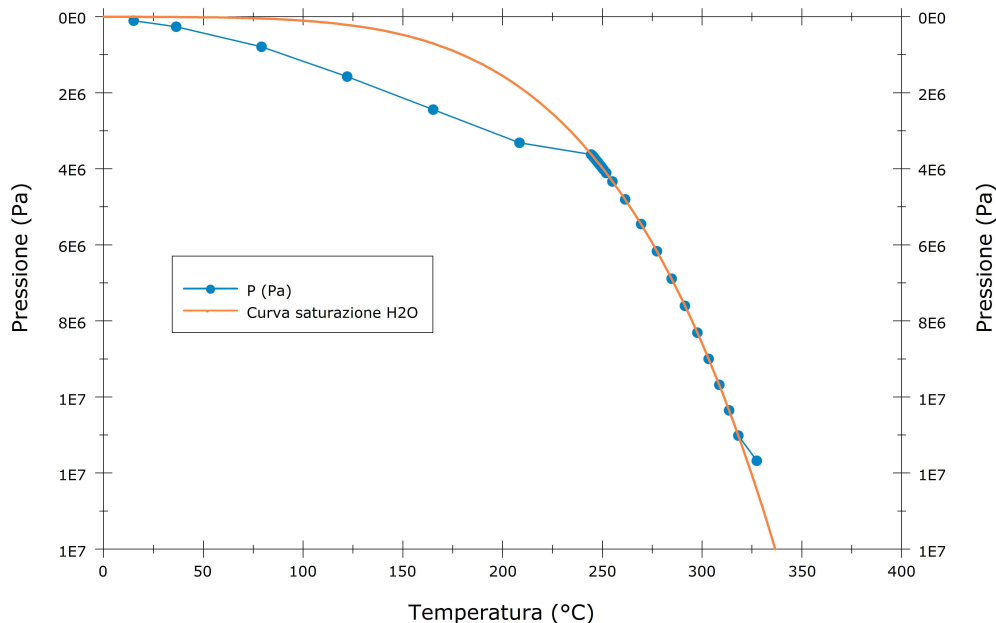


Figura 4.6 Diagramma P-T di confronto con la curva di saturazione dell'acqua – 20650 anni

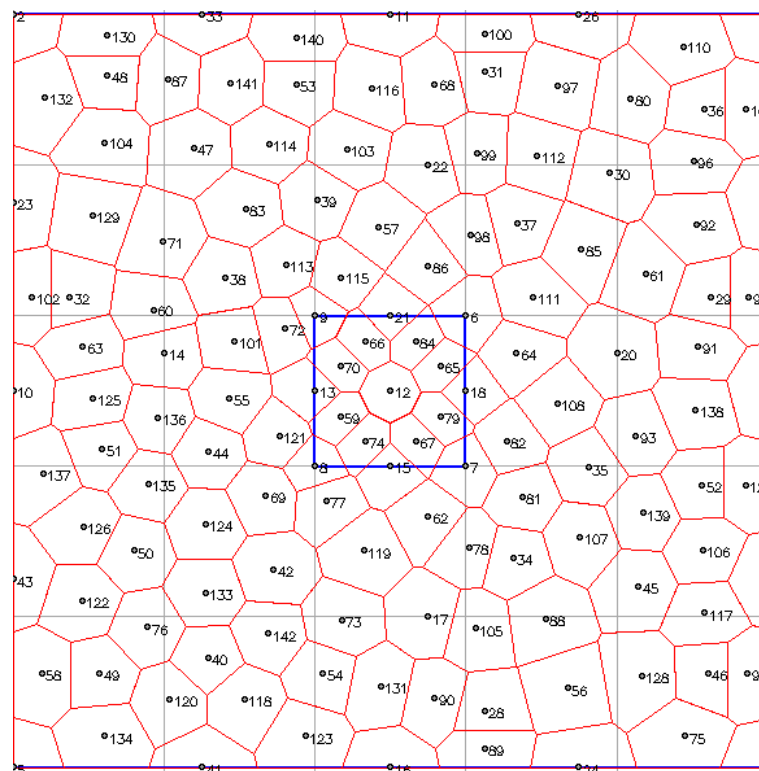
#### 4.1.3. Stato Naturale del bacino di Larderello-Travale – griglia irregolare

Per il modello sopra descritto è stata realizzata anche una discretizzazione spaziale con griglia irregolare, con la procedura sviluppata da C.I.N.I.Geo che combina le funzionalità

dei software GRASS e AMESH, che permette di suddividere le aree di interesse in domini spaziali che hanno le caratteristiche dei poligoni di Voronoi.

#### Descrizione della Griglia di discretizzazione spaziale

Il volume simulato risulta lo stesso; la griglia è stata creata considerando un maggior raffittimento nell'area centrale individuata dal bordo blu (Fig. 4.7), mentre si è scelto di discretizzare il dominio con blocchi di dimensioni maggiori nelle aree più esterne. In totale, si sono ottenuti 141x36 blocchi (totale 5072).



*Figura 4.7 Diagramma P-T di confronto con la curva di saturazione dell'acqua – 20650 anni*

#### Descrizione del modello numerico

La simulazione è stata condotta utilizzando la stessa distribuzione di materiali, condizioni iniziali e condizioni al contorno del modello di stato naturale a griglia regolare.

I risultati delle simulazioni sono congruenti. I profili di pressione, temperatura, saturazioni, permeabilità relative e flussi mostrano, a fronte di differenze in valore numerico, gli stessi andamenti in profondità (Figg. 4.8, 4.9, 4.10, 4.11, 4.12).

Il confronto mette in evidenza la validità della procedura di discretizzazione a maglia irregolare: infatti, nonostante il numero dei blocchi sia quantitativamente diverso ma soprattutto differiscono le aree di interfaccia e le distanze di connessione tra blocchi, le simulazioni dei due modelli mostrano gli stessi risultati, e cioè gli stessi profili di variazione delle variabili principali, poichè sono caratterizzati da uniformità spaziale per livelli. Ogni livello, che in entrambe le simulazioni mantiene lo stesso spessore, ha infatti una distribuzione di parametri legata al materiale e non al singolo blocco, e le variazioni nei valori dei parametri seguono la stessa evoluzione.

E' quindi evidente l'importanza di ricorrere alla discretizzazione a maglia irregolare, con raffittimenti localizzati, laddove sia presente un dettaglio di conoscenze maggiore, che rende necessaria la creazione di blocchi minori per poter opportunamente popolare il modello numerico.

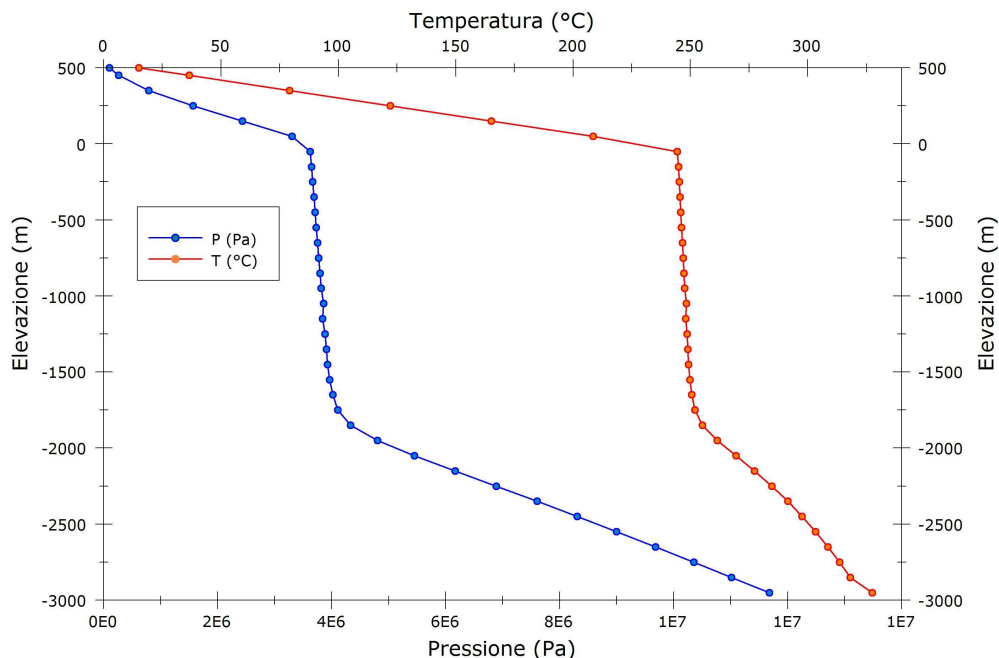


Figura 4.8 Profilo di temperatura e pressione – 20650 anni

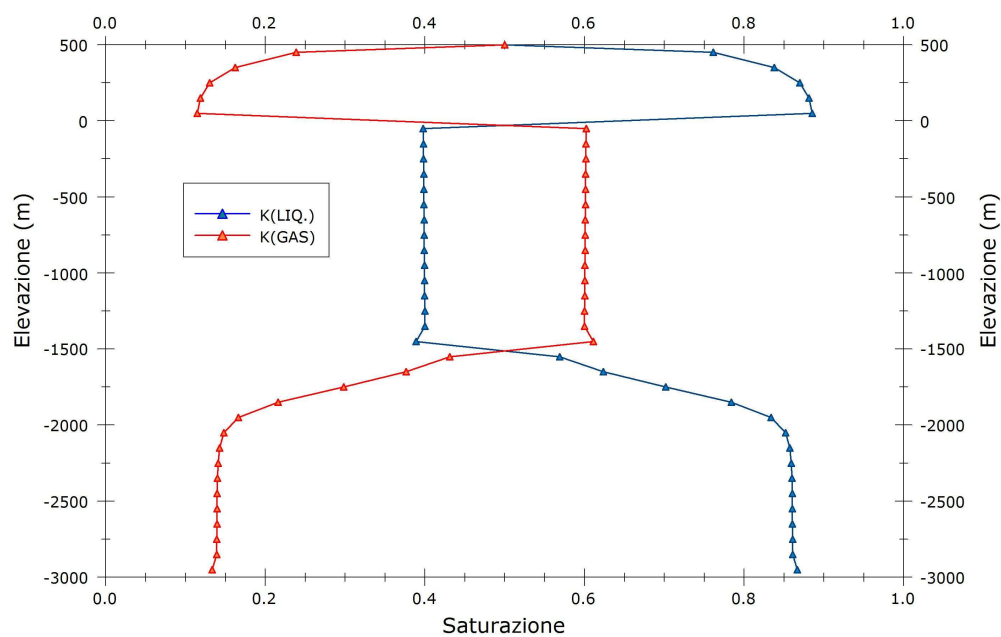


Figura 4.9 Saturazione gas e liquido – 20650 anni

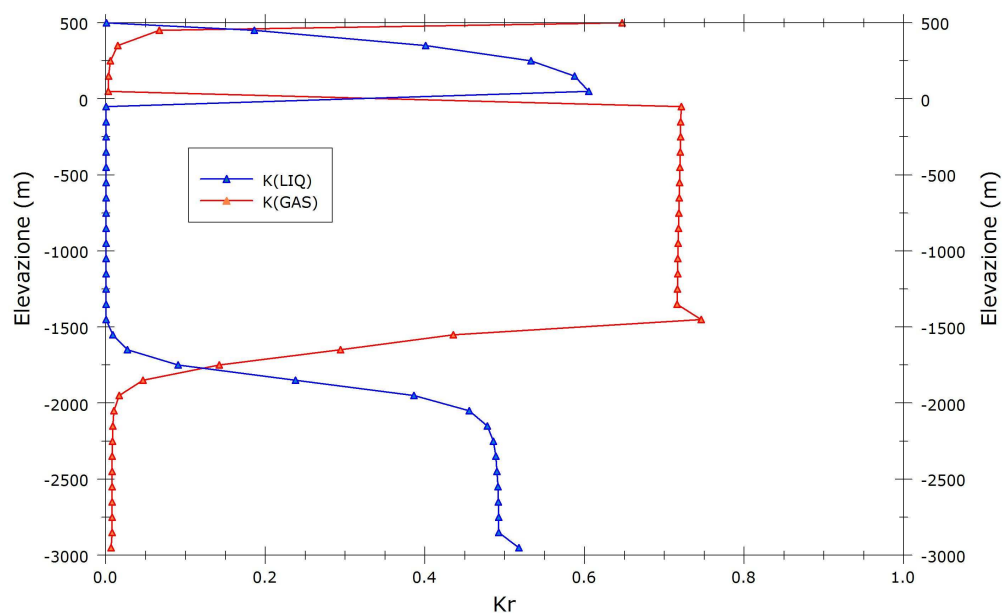


Figura 4.10 - Permeabilità relativa gas e liquido – 20650 anni

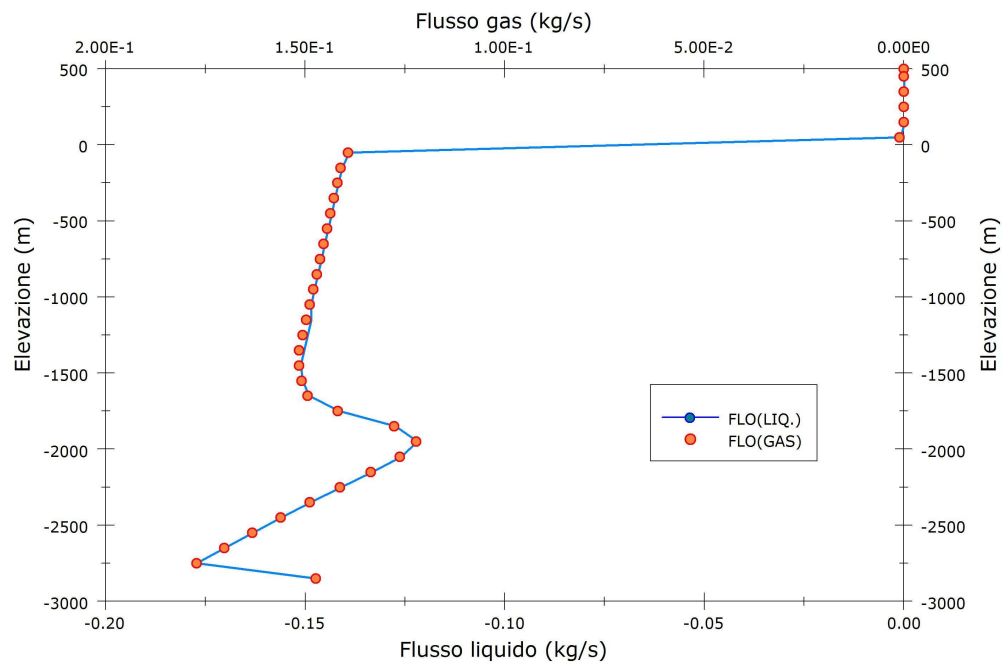


Figura 4.11- Profilo flusso gas e liquido – 20650 anni

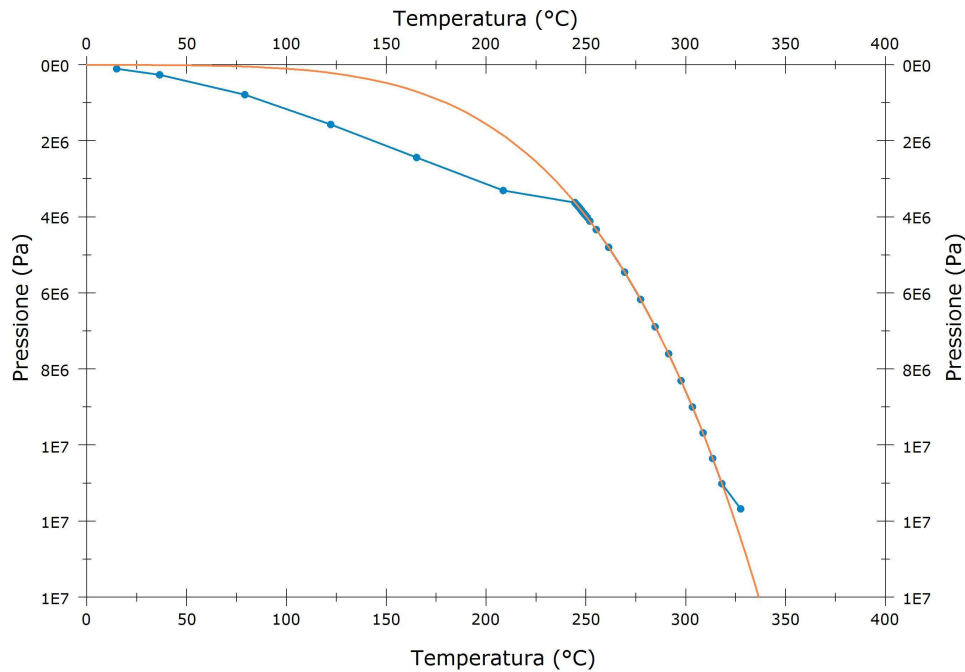


Figura 4.12 - Diagramma P-T di confronto con la curva di saturazione dell'acqua – 20650 anni

#### 4.1.4. Modello 3x3x5

La complessa architettura di operazioni necessarie per effettuare simulazioni corrette dal punto di vista concettuale comprende anche l'utilizzo del processo di simulazione inversa che, come illustrato nel capitolo 2, è necessario per le funzionalità di *History Matching* e *Sensitivity Studies*, analisi fondamentali per la calibrazione dei modelli simulati.

Le elaborazioni di *History Matching* e la verifica della corretta compilazione del software dedicato per le simulazioni inverse necessitano dell'utilizzo dei *punti di calibrazione*, spesso costituiti da parametri di produzione, o comunque da dati sperimentali o valori misurati.

Per sola finalità di verifica dell'accuratezza di una elaborazione, e quindi di affidabilità del codice, è possibile sopperire alla carenza di valori noti/misurati in ingresso fornendo al software dati *sintetici*; tale procedura prevede che sia preparata una simulazione in forward i cui risultati nel tempo costituiscano i *punti di calibrazione* (i dati sintetici) in ingresso per la simulazione inversa. Il modello numerico viene alterato nel valore di uno o più dei parametri associati e così fornito al software, in modo tale che l'utente esperto possa verificare le potenzialità e la corretta calibrazione della "*funzione obiettivo*" valutando con quale precisione il simulatore inverso è in grado di ricostruire i valori corretti dei parametri alterati partendo dai valori prodotti dal modello numerico originale.

La scelta di tale procedura di verifica si è motivata per la mancata accessibilità ad un set idoneo di parametri di produzione per i dati di ingresso, insieme a considerazioni in termini di disponibilità di risorse e tempi computazionali.

Per verifica delle funzionalità e applicabilità del nuovo codice di simulazione inversa messo a punto da DSI – Firenze e C.I.N.I.Geo, si è quindi utilizzato un modello di dimensioni ridotte, denominato 3x3x5, che è stato utilizzato per la simulazione di iTOUGH2 in forward da cui sono stati estrapolati i dati per creare un file di input per iTOUGH2.

#### Modello numerico

La discretizzazione spaziale è stata realizzata creando un modello semplice di 45 blocchi, suddivisi in 5 layer di 3x3 blocchi ciascuno.

I blocchi hanno dimensioni di 100 x 100 metri in orizzontale, e spessori (dall'alto verso il basso) di 100, 20, 100, 100 e 100 m.

Per introdurre le condizioni al contorno di tipo Dirichlet, il file MESH è stato modificato in modo da conferire ai blocchi appartenenti al dominio ATMOS un volume molto elevato dell'ordine di grandezza di  $10^{53}$  m<sup>3</sup> (keyword ELEME) e posizione del centroide del blocco prossima all'interfaccia (keyword CONNE).

#### TOUGH2 Input File

Il modello è stato costruito identificando con ogni layer un dominio (litotipo) diverso. Rispettivamente, dall'altro verso il basso, si individuano i domini ATMOS (atmosfera), CAPRO, ROC11, ROC22, ROC33 e sono assegnati nel blocco identificato dalla keyword "ROCK" del file di input i relativi parametri. In particolare, per quanto riguarda il parametro porosità, sono stati assegnati i seguenti valori:

- (ATMOS)=0.99,



- (CAPRO)=0.01,
- (ROC11)=0.10,
- (ROC22)=0.12,
- (ROC33)=0.15.

Con la keyword **INDOM** sono state inoltre introdotte le condizioni iniziali dei singoli domini assegnando valori delle variabili primarie richieste secondo il modello ricreato nella simulazione del modello Stato Naturale di Larderello-Travale.

Analogamente a tale modello sono state associate temperature elevate ai layer **ROC11**, **ROC22**, **ROC33**, che simulano il serbatoio geotermico, mentre si è deciso di non introdurre nella fase preliminare di simulazione la fonte di calore basale attraverso la keyword **GENER**.

La simulazione in oggetto è stata condotta per ottenere un set di dati per compilare il file di input per la relativa simulazione inversa: in particolare, si è deciso di fornire una serie di misure di pressioni (Pa) di un singolo blocco (**A42 2**) in 51 istanti temporali, separati dallo stesso intervallo di tempo.

A tal fine, si è operato sul file di input di **TOUGH2** in modo da ottenere nel file di output della simulazione una tale quantità di informazione senza però rendere ingestibile il documento per dimensioni.

Questo è stato possibile mediante la compilazione di specifici campi afferenti alle keywords **PARAM** e **TIMES**.

In **PARAM** le scelte principali sono state quelle di porre **KDATA=1**, in modo da avere una stampa per ogni blocco soltanto delle variabili principali per ogni istante di tempo, e **MOP(1)=0** in modo da evitare la stampa delle iterazioni di Newton-Raphson non convergenti (la convergenza della simulazione è stata verificata in una fase precedente con runs appositi), così da alleggerire il file di output.

La keyword **TIMES** è stata utilizzata per imporre la stampa delle variabili di tutti i blocchi a 51 istanti di tempo predefiniti, ad intervalli regolari di  $1.72E+10$  secondi.

#### Estrapolazione dati per file di input di iTOUGH2

Dal file di output della simulazione eseguita con le caratteristiche sopra descritte, sono stati estrapolati i dati da inserire nel file di input di **iTOUGH2** per il blocco “**A42 2**” (layer 4, dominio **ROC22**), scelto arbitrariamente con l’univo vincolo di non appartenere al dominio (**ROC11**) di cui abbiamo modificato la porosità per la simulazione inversa. La tabella risultante è quindi costituita da 51 misure di pressione del blocco prescelto ognuna riferita a 51 istanti temporali, equamente distribuiti nel tempo della simulazione (dall’istante 0 a circa 20.650 anni).

## 5. Parallel Virtual Machine (PVM)

La Parallel Virtual Machine (PVM), sviluppata presso l'Oak Ridge National Laboratory dell'University of Tennessee e dall'Emory University (1989), è un insieme integrato di strumenti software e di librerie che emulano un ambiente generale e flessibile per l'elaborazione concorrente di macchine interconnesse con architettura eterogenea.

### 5.1. Definizioni

- **Host:** ciascuna workstation facente parte l'insieme di computer che costituisce il cluster;
- **Demone:** [programma eseguito in background](#), senza che sia sotto il controllo diretto dell'utente;
- **Task:** porzioni di programma eseguite in modo asincrono (le operazioni sulla base dati possono essere eseguite simultaneamente in quanto non sono interdipendenti, viceversa possono esserlo i rispettivi risultati). Tutti i task sono identificati da un intero detto *task identifier (TID)*;
- **TID (task identifier):** codice che identifica univocamente ciascun task. I TID indirizzano univocamente i demoni, i task e i gruppi di task interni alla macchina virtuale. Essi devono essere unici all'interno della macchina virtuale e sono assegnati dal demone locale di pvm, non dall'utente. Il TID è assimilabile al concetto di indirizzo, infatti permette ai task di comunicare. Al suo interno sono codificate informazioni che permettono di localizzare il task riferito, come ad esempio su quale computer ("*host*") risiede.

### 5.2. Caratteristiche

L'obiettivo di PVM è quello di permettere di usare una rete di computer come un'unica macchina parallela.

Le principali caratteristiche sono:

- riconfigurazione dinamica, da parte dell'utente, della macchina virtuale;
- accesso trasparente all'hardware;
- il task è l'unità di parallelismo;
- modello di comunicazione basato sullo *scambio di messaggi*;

- supporto trasparente di molteplici architetture hardware.

Il sistema PVM è composto di due parti: la *macchina virtuale parallela* e una *libreria di primitive* per il supporto del calcolo parallelo.

La macchina virtuale parallela fornita da PVM è un sistema software che permette di virtualizzare un'architettura parallela astruendo l'architettura sottostante. PVM si basa sulla presenza di un demone (*pvm*) in esecuzione su ogni host reale nel quale si vuole suddividere la computazione. L'insieme di tali host è definito *macchina virtuale parallela*.

Il demone ha il compito di scambiare i messaggi e verificare lo stato dei task interni alla macchina virtuale. Esso fornisce quindi un punto di contatto su ogni host, sia per quanto riguarda le *intra-comunicazioni* (comunicazioni tra task che risiedono sullo stesso host), sia per le *inter-comunicazioni* (comunicazioni tra task residenti su host diversi).

Il demone residente sull'host che ha inizializzato il calcolo è detto *master* (o *parent*), mentre tutti gli altri sono detti *slave* (o *child*).

PVM supporta i linguaggi C, C++ e Fortran.

### 5.3. Primitive

Esistono due tipologie di primitive all'interno della libreria PVM:

- controllo dei processi: insieme di primitive che consentono di agire direttamente sui task, come ad esempio restituire il TID del task che la richiama, oppure eliminare il task identificato da un certo TID;
- scambio messaggi: insieme di primitive che consentono ai task di comunicare tra loro per conoscere lo stato di avanzamento dei vari processi.

### 5.4. Architettura e sincronizzazione

Le strutture dati principali di PVM sono:

- La tavola degli host
- La tabella dei task
- Le code dei messaggi

Alla macchina virtuale PVM è associata la tavola degli host. Ogni elemento di tale tavola rappresenta un host facente parte della macchina virtuale e ne descrive nome, indirizzo e stato.

Per ragioni di efficienza (accelerare le riconfigurazioni dinamiche della macchina virtuale) e di tolleranza ai guasti (per rilevare i partizionamenti) la tavola degli host è

replicata su ogni altro e le copie sono mantenute consistenti tramite particolari algoritmi di aggiornamento.

Ad ogni demone PVM è associata la tavola dei task. In questa tabella sono descritti i task che sono gestiti da quel demone, cioè quelli residenti sull'host.

Per ogni elemento della tabella dei task è mantenuta una coda (FIFO) di messaggi inevasi, cioè ricevuti dal demone, ma non evasi dal task destinatario, da inoltrare al corrispondente task.

Due processi che interagiscono hanno necessità di sincronizzarsi al fine di scambiarsi informazioni e accedere in modo esclusivo a determinate risorse.

### 5.5. Cluster installato da CINIGeo

CINIGeo ha installato un cluster composto, attualmente, da 4 workstation, tutte con Sistema Operativo Linux. Il sistema PVM si basa su un'architettura client-server, secondo la quale i vari host che compongono la macchina virtuale devono condividere una *home directory*, al fine di agire come una singola unità logica. Pertanto solo su un host fisico (il *server*) si trova fisicamente la cartella *home* che viene condivisa a tutte le altre workstation (*client*). Tale condivisione viene effettuata attraverso la rete ethernet ed il servizio Network File System (NFS). La home che risiede sul server viene condivisa esclusivamente alle macchine che compongono il sistema PVM. Le comunicazioni tra host avvengono attraverso il servizio Secure SHell (SSH), che consente l'invio di comandi tra macchine differenti, secondo un sistema di chiavi pubbliche e private. Per tali ragioni si è scelto di utilizzare come server la macchina con la maggior quantità di disco. Tale macchina può anche non partecipare alla parte di calcolo, ma esportare semplicemente la home; nel caso in cui anche questo host debba partecipare alla parte computazionale è bene dimensionare correttamente CPU e RAM.

Il cluster installato è composto dalle seguenti macchine con le seguenti caratteristiche:

- Server:
  - Sistema operativo Linux Ubuntu 8.04
  - Kernel 2.6.24-28-server
  - 4 processori Intel® Core™2 QUAD CPU Q8200 @ 2.33 GHz
  - RAM 3,9 GiB
- Client1:
  - Sistema operativo Linux Ubuntu 8.10
  - Kernel 2.6.27-17-generic
  - 2 processori Intel® Core™2 DUO CPU E4600 @2.40 GHz

- RAM 3,2 GiB
- Client2:
  - Sistema operativo Linux Ubuntu 8.10
  - Kernel 2.6.27-17-generic
  - Processore Intel® Pentium®4 CPU 2.20 GHz
  - RAM 2.0 GiB
- Client3:
  - Sistema operativo Linux Ubuntu 9.10
  - Kernel 2.6.31-22-generic
  - Processore Intel® Celeron® CPU 3.06 GHz
  - RAM 712 MiB

In questa struttura, il server coincide con il master della PVM, ovvero la workstation che fa da server per il cluster é la macchina da cui viene fatto partire il sistema virtuale.

## 6. iTOUGH2-PVM

iTOUGH2 esegue tante simulazioni dirette in TOUGH2, quanti sono i set di valori dei parametri da stimare e confronta, per ciascuna simulazione, l'output ottenuto con i valori misurati, determinando così il set di parametri che meglio approssima i valori misurati. Alcuni di queste simulazioni dirette sono indipendenti le una dalle altre e ciò rende possibile eseguirle contemporaneamente in parallelo su macchine differenti. iTOUGH2-PVM sfrutta questo concetto per eseguire diverse simulazioni dirette contemporaneamente su macchine diverse e ridurre così il tempo totale di calcolo.

Per una descrizione completa dei processi che é possibile eseguire in parallelo si rimanda al manuale *Parallelization of iTOUGH2 Using PVM*.

Non é possibile sapere a priori il risparmio di tempo di calcolo, poiché il guadagno che deriva dall'uso di iTOUGH2-PVM dipende dalla struttura del cluster che si utilizza, dal numero e dalla tipologia di workstation disponibili, dal modello oggetto di studio, dal numero dei parametri che occorre stimare e dal metodo scelto per la ricerca dei minimi della funzione obiettivo. Pertanto, é necessario eseguire alcune simulazioni, sia in sequenziale (su una sola macchina), sia in parallelo, per determinare la configurazione migliore in termini di risparmio di tempo di calcolo. In particolare, è la workstation con le prestazioni peggiori all'interno del cluster che influenza enormemente il tempo di calcolo totale poiché rallenta l'elaborazione dell'intero sistema.

## 6.1. Architettura di iTOUGH2-PVM

iTOUGH2-PVM é basato sul modello “*parent-child*” nel quale il processo *parent* inizializza il problema e invia informazioni specifiche a un determinato numero di processi *child*. Ciascun processo *child* esegue quindi un’intera simulazione in TOUGH2 e restituisce il risultato della simulazione al *parent* per le successive elaborazioni.

## 6.2. Utilizzo di iTOUGH2-PVM

Per poter utilizzare la versione per PVM di iTOUGH2, si esegue la stessa procedura descritta per iTOUGH2: occorre innanzitutto modificare il file *maxsize.inc* (fornito con il codice sorgente), settando i parametri in maniera funzionale per gli scopi di questo progetto. In seguito occorre compilare il codice per ciascun modulo (EOS1, EWASG ed EOS3) con le corrette opzioni per Linux e per la PVM. Una volta compilato il codice con tali opzioni, é indispensabile lanciare il programma mediante gli scripts file forniti assieme al codice sorgente. Per una descrizione completa della procedura, si vedano i manuali di *iTOUGH2 User’s Guide* e *Parallelization of iTOUGH2 Using PVM*.

## 6.3. File di input per iTOUGH2-PVM

- I file di input di iTOUGH2-PVM sono gli stessi di iTOUGH2. L’unica differenza si riscontra nel file **inomefile.inp**, nel quale vengono aggiunte le opzioni necessarie per il funzionamento del codice con il cluster di workstation, ovvero numero di host slave presenti e tipo di calcolo eseguito in parallelo.

## 6.4. File di output di iTOUGH2-PVM

iTOUGH2-PVM crea gli stessi file in formato ASCII di iTOUGH2. Con la versione parallela, il file di output **inomefile.out** contiene anche un report che descrive le prestazioni e l’utilizzo di ciascun host all’interno del processo di simulazione eseguito.

# 7. Algoritmo di Inversione implementato in RBFOPT

Il modulo RBFOPT implementa l'algoritmo di ottimizzazione globale basato su funzioni surrogate a base radiale proposto in (Gutmann 2010). L'algoritmo utilizza una funzione surrogata per interpolare le osservazioni disponibili della funzione di errore ed una funzione ausiliaria (detta di utilità) per decidere quale debba essere il nuovo set di parametri per cui eseguire la simulazione.

Lo schema generale dell'algoritmo é il seguente:

1. Genera un campione di  $n+1$  set di parametri opportunamente distribuiti negli intervalli di loro definizione;
2. Esegui le corrispondenti simulazioni;
3. Calcola l'interpolante iniziale;
4. Ripeti per  $n_{\max}$  iterazioni:
  1. Ottimizza l'interpolante attuale per decidere un nuovo valore target dell'errore;
  2. Ottimizza la funzione di utilità per decidere il nuovo set parametri candidato;
  3. Esegui la simulazione con i parametri scelti in 4.2;
  4. Torna a 4.

Le ottimizzazioni sia dell'interpolante sia della funzione di utilità vengono risolte con un semplice algoritmo di ottimizzazione globale di tipo Multistart che genera una configurazione casuale delle variabili ed una ricerca locale (che nel nostro caso utilizza l'algoritmo L-BFGS-B), tenendo traccia del migliore valore trovato.

L'algoritmo viene inizializzato con  $n+1$  valutazioni scelte mediante l'utilizzo di un campionamento basato su un ipercubo latino, oppure valutando alcuni dei vertici dell'ipercubo stesso. Si può inoltre includere una valutazione aggiuntiva del first guess fornito dall'utente, se viene ritenuto un buon valore.

L'algoritmo non utilizza quindi nessuna approssimazione delle informazioni del primo ordine (come differenze finite per approssimare il gradiente) ed è pensato per un'esplorazione di tutto l'intervallo di definizione fornito dall'utente.

Avendo rilevato che la funzione di errore può assumere valori molto grandi, si è previsto inoltre la possibilità di utilizzarne il logaritmo in modo da limitarne l'escursione. Quest'ultimo aspetto è noto in letteratura per essere un punto cruciale per l'efficacia dei metodi basati su base radiale (si veda Jakobsson S. et Al).

Si vuol inoltre rimarcare che l'algoritmo RBFOPT esegue un'ottimizzazione globale e che quindi è relativamente poco sensibile al first guess disponibile, contrariamente ad algoritmi locali come Levenberg-Marquardt (il default di iTOUGH2). La comparazione fra gli algoritmi disponibili deve quindi inevitabilmente tener conto della differente ottica secondo cui gli algoritmi sono progettati e del diverso obiettivo che si prefiggono.

## 7.1. Sviluppo Software

Un certo sforzo di sviluppo software si è reso necessario per arrivare ad una usabilità soddisfacente del modulo di stima dei parametri. In quest'ottica è stato ulteriormente migliorata l'integrazione con il simulatore iTOUGH2, al fine di ottenere un utilizzo

semplice ed analogo a quello che l'utente sperimenta nell'uso degli altri moduli già presenti nel simulatore stesso (si veda il manuale di iTOUGH2 disponibile in rete all'indirizzo <http://esd.lbl.gov/ITOUGH2/>).

Le opzioni disponibili attualmente sono le seguenti:

- `USE_FIRST_GUESS`: include il first guess fra i punti iniziali con cui inizializzare l'algoritmo;
- `BUMPY_ITERATIONS`: numero di iterazioni dell'algoritmo multistart nell'ottimizzazione della funzione di utilità;
- `RBF_ITERATIONS`: numero di iterazioni dell'algoritmo multistart nell'ottimizzazione della funzione interpolante;
- `MAX_F_EVAL`: numero massimo di simulazioni effettuabili;
- `RND_SEED`: seme di inizializzazione per la generazione casuale di valori usati nell'ottimizzazione globale delle funzioni interpolanti e di utilità;
- `SAMPLING_TYPE`: seleziona il metodo di generazione delle prime osservazioni. Si può scegliere tra un campionamento dei vertici dell'ipercubo individuato dagli intervalli di definizione delle variabili oppure un campionamento basato su ipercubo latino.
- `USE_FIRST_GUESS`: include anche il first guess fornito dall'utente nelle osservazioni iniziali (disabilitato di default);
- `USE_LOG_OBJ_FUN`: usa il logaritmo della funzione di errore per ridurre problemi di scala (disabilitato di default);
- `USE_INPUT_FILE`: legge osservazioni disponibili da un file;
- `USE_OUTPUT_FILE`: salva le osservazioni fatte in un file;
- `ENABLE_DEBUG`: abilita il debug di RBFOPT (un output dettagliato dell'esecuzione viene generato nel file `rbf_opt.out`; disabilitato di default);
- `FUN_THRESHOLD`: rappresenta la soglia sotto la quale ci si ritiene soddisfatti dell'ottimizzazione (il default è 0);

Dal punto di vista dell'integrazione del codice vero e proprio, è stata semplificata l'interazione fra il modulo RBFOPT (che ricordiamo è sviluppato in C++) con il codice originale FORTRAN di iTOUGH2. In particolare, RBFOPT gestisce le sue opzioni adesso come ogni altro modulo di iTOUGH2 accedendo direttamente alle strutture apposite di iTOUGH2.



Risulta inoltre anche possibile includere o no RBFOPT in iTOUGH2 in fase di compilazione, rendendolo un modulo aggiuntivo, tramite l'opzione `-rbf_opt`.

Questo approccio rientra proprio nello spirito di iTOUGH2 cercare di mantenere i moduli di inversione che esplorano globalmente come dei pacchetti separati da compilare solo se necessario.

La modularizzazione ha richiesto la modifica degli script di compilazione forniti da iTOUGH2 e l'aggiunta di opportuni frammenti di codice FORTRAN (denominati stubs) che permettono al codice di includere o meno il modulo esterno.

Inoltre é stata prevista la possibilità di reinizializzare l'algoritmo con dati provenienti da esecuzioni precedenti e salvati con l'apposita opzione (si veda l'opzione `USE_OUTPUT_FILE`). In tal modo, a patto che non venga chiaramente modificato il modello fisico che si intende calibrare, si può proseguire una sessione di calibrazione precedentemente interrotta.

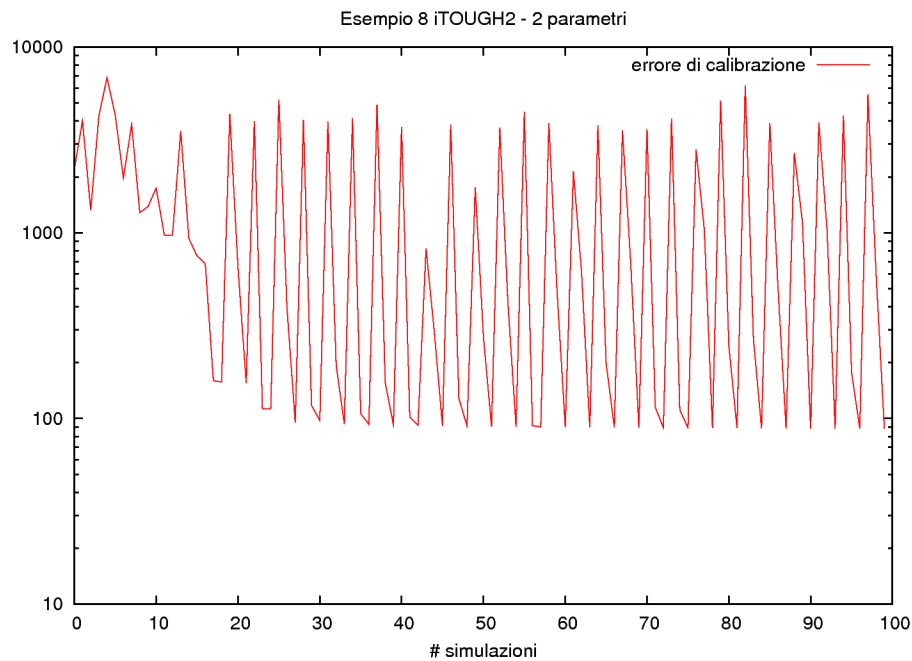
Quest'ultima funzionalità può essere particolarmente conveniente nel caso in cui una simulazione fallisca (per problemi numerici o per un'errata definizione del problema). L'algoritmo RBFOPT allo stato attuale termina la sua esecuzione: l'utente può se possibile modificare alcuni parametri del modello e/o simulatore e ripartire reinizializzando la procedura con le valutazioni già effettuate, sempre che queste siano consistenti con le modifiche effettuate. Un tipico esempio riguarda la definizione degli intervalli di definizione dei parametri da stimare: se l'intervallo scelto é troppo grande, il simulatore potrebbe fallire; anziché effettuare una serie di simulazioni preliminari per stimare questo parametro, si può utilizzare RBFOPT e solo in caso di fallimento ridurre gli intervalli di definizione delle variabili in modo da escludere la combinazione incriminata, per poi ripartire da dove il metodo si é arrestato.

## **7.2. Test numerici effettuati**

Il modulo é stato testato utilizzando essenzialmente due insiemi di test: uno composto da test facenti parte del pacchetto iTOUGH2 (Sample1, Sample3 e Sample8) come test per verificare la correttezza delle simulazioni), mentre l'altro riguarda alcuni modelli sviluppati dal gruppo CINIGeo.

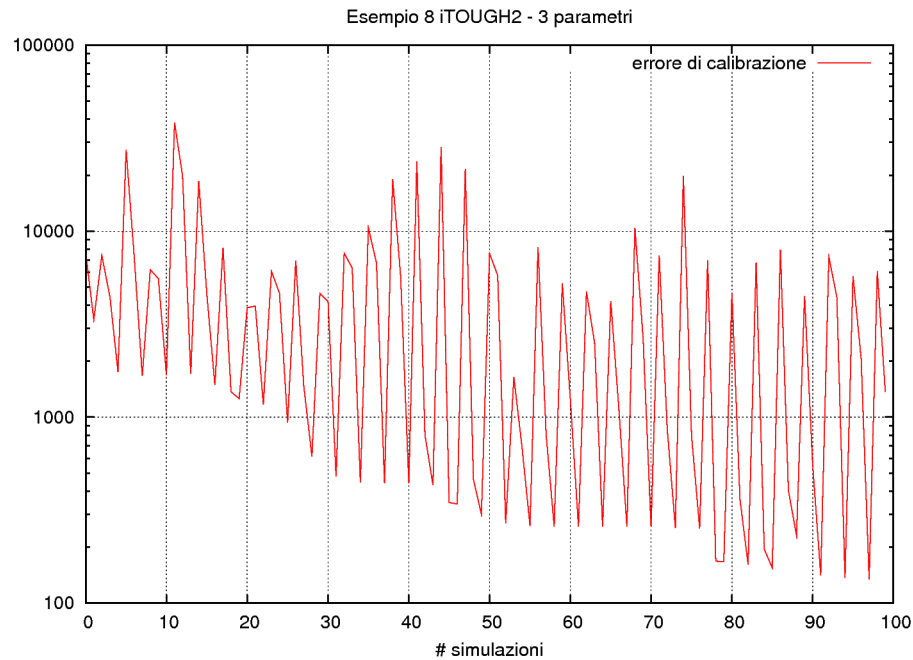
### **7.2.1. iTOUGH2-Sample8**

Per quanto riguarda gli esempi di iTOUGH2, riportiamo qui i risultati per l'esempio 8. Sono disponibili due casi che calibrano due o tre parametri. Per il caso a due parametri le osservazioni della funzione di errore durante l'esecuzione di RBFOPT sono riportate nella figura seguente:

**Figura 7.2**

L'esecuzione dell'algoritmo locale di Levenberg-Marquardt (con parametri standard) permette con 27 iterazioni di trovare un valore di errore pari a 87.3. La natura locale dell'algoritmo LM si conferma in quanto cambiando il first guess, da  $(-12.5, 0.2)$  del caso precedente a  $(-11.7, 0.4)$ , si ottiene un valore di 113 in 23 simulazioni.

Nel caso di stima di tre parametri otteniamo l'andamento delle osservazioni durante l'esecuzione di RBFOPT è riportato nella figura seguente:

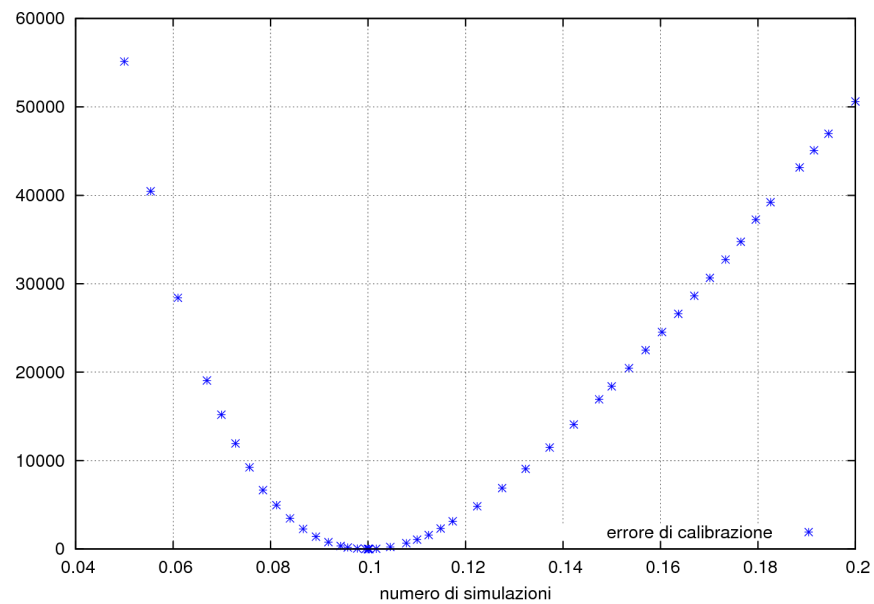
**Figura 7.3.**

L'andamento oscillante riflette la ciclicità delle fasi di esplorazione in ampiezza/profondità che l'algoritmo compie al fine di assicurare sia una buona qualità che una sufficiente esplorazione del dominio in cui i parametri sono definiti.

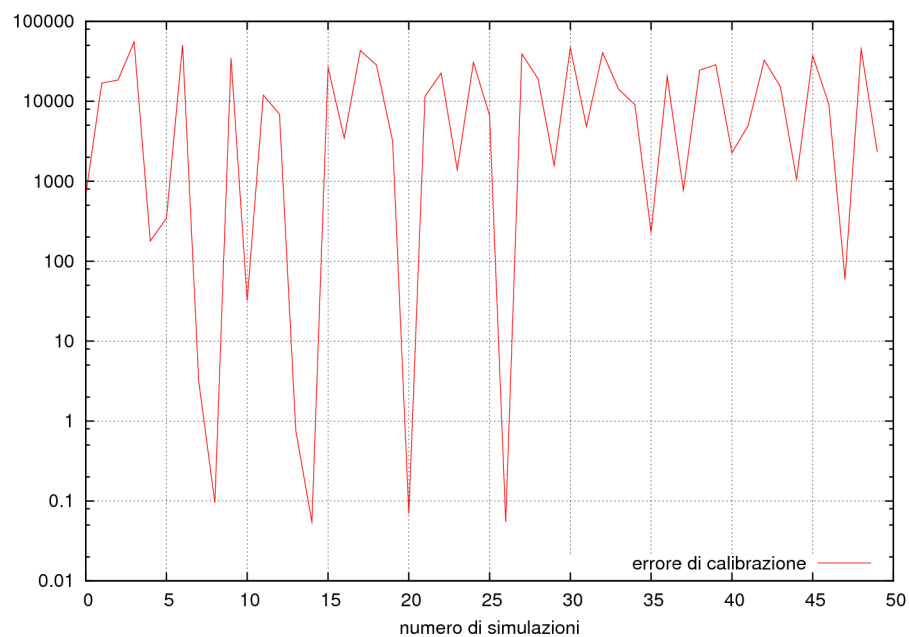
### **7.2.2. 3x3x5**

Questo modello composto da una griglia di 3x3x5 blocchi, in cui si sono ipotizzati rispettivamente 1 o 2 parametri da calibrare.

Nel caso di un parametro da stimare otteniamo:

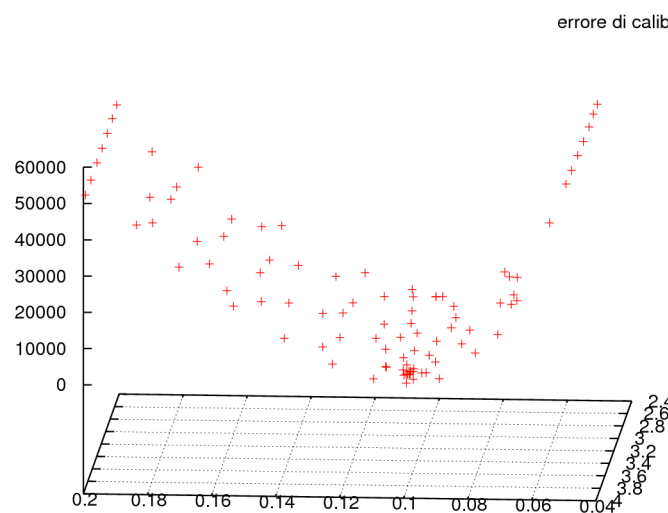
**Figura 7.4.**

L'algoritmo ha mostrato di poter velocemente calibrare il modello, come si evince dalla figura seguente, in cui sono riportati i valori della funzione di errore durante l'esecuzione di RBFOPT.

**Figura 7.5**

L'esecuzione di iTOUGH2 standard (algoritmo di Levenberg-Marquardt) permette di trovare un valore di circa 135 eseguendo 12 simulazioni.

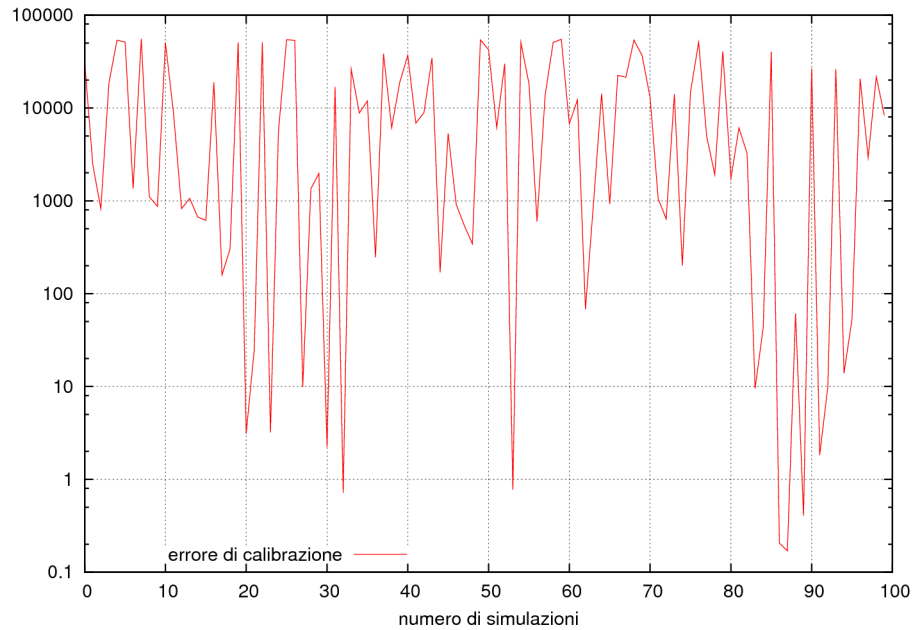
Nel caso di 2 parametri da stimare si ottiene:



**Figura 7.6**

Lo stesso modello dell'esempio precedente é stato arricchito di un secondo parametro. La funzione di errore prende la forma seguente:

Si tratta in sostanza di un paraboloide in cui solo uno dei parametri risulta fondamentale, mentre l'altro poco impatta sulla qualità della simulazione. L'algoritmo evolve come segue:

**Figura 7.7**

Buoni risultati, tenendo conto che si tratta di esplorazioni globali dell'insieme di definizione dei parametri, si ottengono già dopo poche iterazioni. L'esecuzione di iTOUGH2 standard (algoritmo di Levenberg-Marquardt) permette di trovare un valore di circa 7.6 eseguendo 16 simulazioni (valori dei parametri pari a 0.1 e 3.6). Come si vede non si tratta di un valore troppo diverso da quanto siamo in grado di trovare tramite una esplorazione globale con 20 simulazioni.

### **7.2.3. Stato naturale Larderello**

Un modello più complesso prevede l'utilizzo di una griglia di 3600 blocchi. Abbiamo testato RBFOPT nel caso della calibrazione di un singolo parametro. La forma della funzione di errore, tracciata sulla base dei valori in cui è stata valutata da RBFOPT, evidenzia la presenza di almeno due ottimi locali, come si evince dalla figura seguente.

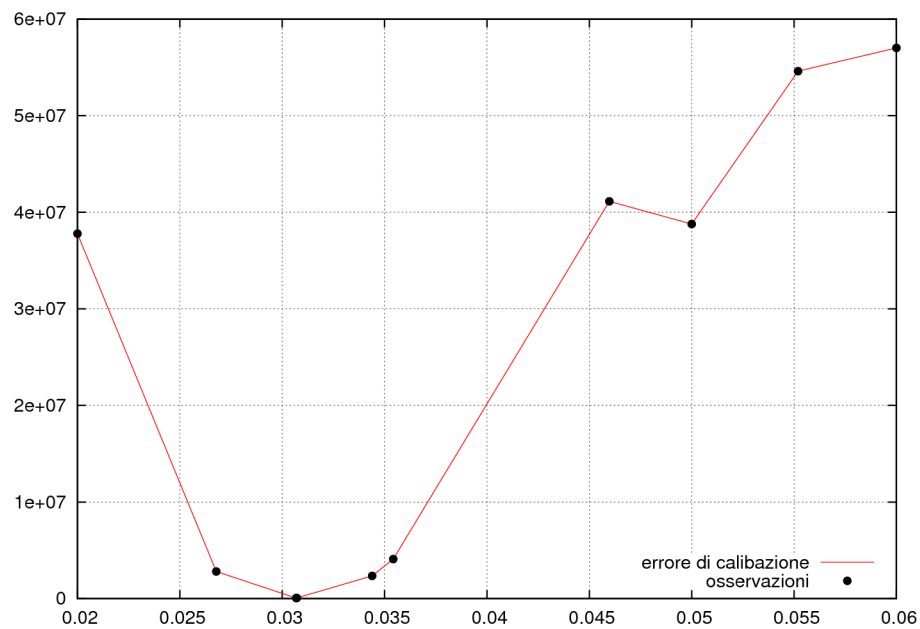


Figura 7.8.

La scelta del first guess sarebbe quindi cruciale per un algoritmo di ottimizzazione locale, che potrebbe non essere in grado di trovare l'ottimo globale. L'esecuzione di RBFOPT ha il seguente andamento:

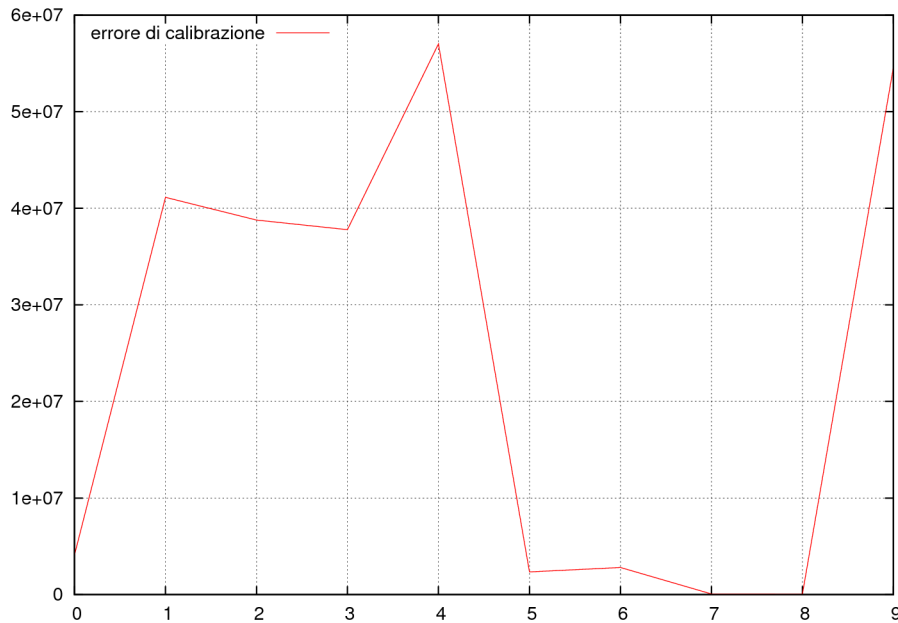


Figura 7.9

Il valore trovato per il parametro da calibrare e' stato 0.0306868, a fronte di un valore effettivo di 0.03.

### 7.3. Sviluppi Futuri

Gli sviluppi futuri saranno sicuramente rivolti al miglioramento dell'algoritmo implementato nel modulo RBFOPT, al fine di adattarlo al problema specifico. In particolare diversi parti dell'algoritmo possono essere modificate semplicemente utilizzando altre possibilità già note in letteratura, ma la cui efficacia deve essere verificata per il problema in esame:

- utilizzo di altre basi interpolanti;
- utilizzo di polinomi di grado superiore;
- altre regole di esplorazione;
- estensione parallela per sfruttare la possibilità data dalla presenza del supporto PVM.

Anche la gestione dei casi in cui la simulazione richiesta fallisca vanno indagati: la strada più semplice è probabilmente quella di arrestare il processo di inversione (come avviene attualmente), ma in alternativa si potrebbe semplicemente assegnare alla configurazione dei parametri che dà errore un valore molto alto dei residui (in linea con



iTOUGH2) così da scoraggiare l'algoritmo nell'avvicinarsi di nuovo a questa configurazione. Si può inoltre facilmente prevedere la possibilità, su richiesta dell'utente, di eseguire una ottimizzazione locale, partendo dal miglior valore trovato da RBFOPT, usando Levemberg-Marquardt come algoritmo di discesa locale.

## 8. Riferimenti Bibliografici

Al Geist, *PVM: Parallel Virtual Machine*, 1994

Battistelli A., Calore C., Pruess K., *Simulator TOUGH2/EWASG for modelling geothermal reservoir with brines and non condensable gas*, Geothermics 26, no4 (1997)

Cazzola W., *PVM: Parallel Virtual Machine*, 1997

Chierici G.L., *Principi di ingegneria dei giacimenti di idrocarburi*, 2004

Finsterle Stefan, *iTOUGH2 Sample Problems*, 2007

Finsterle Stefan, *iTOUGH2 User's Guide*, 2007

Finsterle Stefan, *Parallelization of iTOUGH2 Using PVM*, 1998

Gavana A., *Validation of a non-local optimization Algorithm using History Matching problems*, 2002

Gutmann, H. M., *A Radial Basis Function Method for Global Optimization*, 2001, Journal of Global Optimization

Jakobsson S. et Al. *A method for simulation based optimization using radial basis functions*, 2010 , Optimization and Engineering

R. McKibbin and K. Pruess, *Some effects of noncondensable gas in geothermal reservoirs with steamwater counterflow*, Geothermics 18, n. 3 (1989). ]

Pruess Karsten, *TOUGH2 User's Guide Version 2.0*, 1999

Shook G. M., *Development of a vapor-dominated reservoir with a “high-temperature” component*. Geothermics 24, n. 4 (1995)