



DOTTORATO DI RICERCA

in

SCIENZE COMPUTAZIONALI E INFORMATICHE

XXIV ciclo

Consorzio tra Università di Catania, Università di Napoli Federico II,
Seconda Università di Napoli, Università di Palermo, Università di Salerno
SEDE AMMINISTRATIVA: UNIVERSITÀ DI NAPOLI FEDERICO II

Tecniche numeriche per la risoluzione
in ambiente parallelo del problema
“Data Assimilation”:
un problema inverso mal posto

Rossella Arcucci

TESI DI DOTTORATO DI RICERCA

alla piccola *Aurora*...

*“Chiaror di luce dopo l'alba e prima
che spunti completamente il sole”.*

E come sempre alla fine di un percorso, mi viene quasi naturale tirare le somme e voltare un attimo lo sguardo a com'ero e com'era la mia vita prima di intraprendere questa strada.

Beh... Questa tesi per me racconta una storia... la storia di una involontaria iniziazione!!!

Ho conosciuto il mio mentore, il Prof.Murli per caso! ... pur avendo fatto tutti gli esami che mi erano possibili di analisi numerica, calcolo scientifico, etc., avevo cominciato la tesi di laurea in un'altra materia e con un altro Professore.

Contattata da un'azienda per un'opportunità di stage, vengo "spinta" dal mio allora relatore verso il Prof.Murli. Pochi giorni dopo, già lavoravo ad una nuova tesi in analisi numerica sui metodi di discretizzazione di operatori differenziali... relatore: Prof. Almerico Murli insieme con la Prof.ssa Luisa D'Amore.

Lí compare nella mia vita Livia... giovane ricercatrice piena di grinta, piena di vita e dai cui occhi traspariva letteralmente il fuoco della passione per la ricerca. Lei e la sua inseparabile "penna rossa" mi hanno aiutato ad assaporare il primo gusto per questo lavoro!

Ma... come un bambino che si svezza, ne odiavo quasi i primi sapori... mi facevano stare male!

"Non é matematica quella che scrivi tu!" mi diceva la Prof.ssa D'Amore

e io pensavo: “ma come? ho scritto tutto in formule... é matematica!”... “In questo lavoro non basta capire! ... é necessario capire VERA-MENTE!” mi ripeteva il Prof. Murli ... “che significa capire veramente???” ... “io ho capito!!!” ... ripetevo tra me i primi mesi...

Beh... non avevo capito proprio nulla dico adesso!

Solo dopo un anno di tesi e con l'aiuto di Livia, della Prof.ssa D'Amore e del Prof. Murli finalmente presi coscienza di cosa significasse capire veramente... e... da quel momento cominciai a capire veramente anch'io!!!

Una giovane laureata in Matematica piena di buoni propositi e un entusiasmo che mi ha sempre contraddistinto. In quel periodo, scrutavo “quel gruppo” di giovani ricercatori, post-doc e dottorandi che lavoravano per il Prof. Murli. Tutti in un unico laboratorio, stile diverso dagli altri gruppi del mio dipartimento che erano “piú sparsi” tra vari studi. Li vedevo andare sempre a pranzo tutti insieme, pausa caffè tutti insieme, praticamente facevano tutto... tutti insieme!

Il Prof. Murli mi propone di rimanere... di entrare a far parte di “quel gruppo”... ACCETTAI!

... Molto presto capii che quello era molto piú di un gruppo, era una grande famiglia.

Rosanna... con la sua bionda dolcezza e la sua disponibilità mi fece capire subito che quella poteva essere casa mia!

Valeria... beh... avevo la sensazione che pensasse “c'é un'altra che vuole immolarsi per la causa?... e quindi?... chi se ne frega!”... adesso che la conosco bene so che molto probabilmente pensava proprio questo!

Oggi sono diventate la mia quotidianità e mi diverte ricordare come le vedevo allora.

Diego... un maestro per me! ... ho impressa nella mente l'espressione sorridente che aveva quando gli facevo una domanda "banale" a cui non mi rispondeva subito per spingermi ad arrivarci da sola! ... e... "che soddisfazione quando riesco a rispondermi da sola!!!"

... un amico di pranzo... un confidente che ha saputo ascoltare quando ne ho avuto bisogno.

Oggi Livia, Diego, Rosanna e Valeria sono ormai davvero la mia seconda famiglia.

Ma il gruppo non finisce... Il Prof. Laccetti, il Prof. Lapegna, Vania, Luisina... altre persone ho incontrato in questo percorso e tutti mi hanno lasciato qualcosa.

... Maria... oggi compagna di pausa caffè, grande ascoltatrice e preziosa fonte di tutto per l'intero gruppo.

Voglio ringraziare tutti per quello che so e sono oggi... siete preziosi!!!

Naturalmente il mio ringraziamento più grande vá al Prof. Murli che ha saputo vedere in me quello che io non riesco a percepire.

Alla Prof.ssa D'Amore perché la sua infinita pazienza, la sua disponibilità ad ascoltarmi sempre, lo sprono che riesce a darmi e soprattutto l'entusiasmo che lascia trasparire quando dai nostri colloqui "produco qualcosa di buono" sono fondamentali per me.

e...

... Mentre la mia vita cominciava a svolgersi sempre più in università

che a casa in maniera “direttamente proporzionale” rispetto al tempo, la mia famiglia cominciava a percepire che “era finita”!

Come un’innamorata nei primi mesi di relazione, passavo ormai piú tempo al pc che a parlare con loro. L’ora di cena poteva bastare per una chiacchierata! ... se non avevo finito qualcosa in universitá, subito dopo cena mi rimettevo al pc... non c’ero per nessuno!

... che pazienza hanno avuto tutti!

Mamy e Papy scusate ma l’amore é amore!!!

Manu... meno male che esisti! ... quante volte ti ho trascinato in laboratorio con me, una piccola futura farmacista in un gruppo di ”folli”!
... grazie per le mille volte in cui mi hai supportato e sopportato.

Sister... credi in me piú di me! e questa cosa mi manda tutt’ora al manicomio! ... ”un cervello che va messo a servizio della scienza!” mi dici ... beh... “io non so se mi rimane un cervello... mi sa che me lo sono bruciato con la Data Assimilation!” rispondo scherzando.

Sasi... si dice che una persona é molto importante per te se quando sei “in pericolo” la cerchi! ... io al minimo segno di stress o tristezza, in ogni momento di felicitá e gioia,...
... io, in ogni momento e qualunque cosa accada, bella o brutta, chiamo te! ... per la serie “se non ci fossi bisognerebbe inventarti!!!”

Faby... sai piú tu di me che io stessa! negli ultimi anni ho condiviso con te ogni respiro! ... difficile spiegarla meglio di cosí!
... la tua capacitá di leggere in me mi rende ormai impossibile nasconderti qualcosa! ... scontato dire quanto sei importante... tu sei tu! ... tu e Cla mi meritate tutta! ... ci siete sempre... sempre!!!

Ste... quanto mi diverto con te! ... una nostra chiacchiera-caffé riesce a farmi staccare completamente il cervello che in quel momento é tutto impegnato ad osservare il tuo infinito gesticolare.

Sto pensando a te, Gigi e il baretto... eh... non dico altro!

Ego... non so come tu faccia a sopportare i miei sbalzi d'umore dovuti allo stress da dottorato con l'infinita calma che emani. "Non ti preoccupare!" é la tua parola d'ordine e "Io credo in te!" é la cosa piú bella che tu possa dirmi. Trovi sempre le parole giuste... Alter-Ego... mettila come vuoi... non esiste di piú!

Lele... un fratello acquisito!!! ... in alcuni momenti sei riuscito ad essere esattamente la voce della mia coscienza... con te il mio sguardo vá sempre al futuro positivo!

Lollo, Gianky, Renato, Checco, Marco... quante ne abbiamo vissute nell'ultimo paio d'anni... sono sopravvissuta alla "follia" grazie ai week end trascorsi assieme!

Si dice che chiunque entri nella tua vita lo faccia per un motivo... voi, siete spesso stati per me "la fonte di sfogo"!... che gruppo!!!

Tra qualche anno, se rileggendo queste righe rivivró le emozioni che sto provando in questo momento, allora ne sará valsa la pena perdere qualche ora a buttar giú di getto, nero su bianco, quello che ho dentro.

Grazie davvero a tutti! ...
... se saró riuscita a trasmettervi la metà di quello che avete trasmesso voi a me, vi avró dato grandi emozioni!

Rossella

Indice

Introduzione	1
Abbreviazioni	5
Elenco delle Variabili	6
1 Contesto Applicativo	7
1.1 Relazione tra i dati	9
1.1.1 Tipi di dati osservati	14
1.2 Metodi di risoluzione del Problema Data Assimilation (Stato dell'Arte)	15
1.3 Ulteriori Applicazioni	19
2 Data Assimilation: un problema inverso mal posto	21
2.1 Regolarizzazione del problema di Data Assimilation . . .	27
2.2 Problema di Minimi Quadrati con vincoli (Equazioni normali)	30
2.2.1 Optimal Interpolation e Filtro di Kalman (KF): un solo passo temporale	30
2.2.2 Filtro di Kalman Ensemble (EnKF): m passi temporali	35
2.3 Problema dei moltiplicatori di Lagrange (Approccio variazionale)	39
2.3.1 3DVar: un solo passo temporale	40

2.3.2	4DVar: m passi temporali	41
3	Il Modello OceanVar	44
3.1	Operatore lineare V	46
3.1.1	Variazione degli errori verticali: V_V	47
3.1.2	Variazione degli errori orizzontali. V_H	48
3.1.3	Modello barotropico: V_η	51
3.1.4	Operatore della velocità: V_{uv}	53
3.1.5	Operatore della divergenza: V_D	53
4	Il Software OceanVar	55
4.1	Analisi dei tempi di esecuzione del codice	63
5	Parallelizzazione del Software OcenVar	75
5.1	Il codice multicores	76
5.2	Il codice parallelo multiprocessore	79
A	Alcuni tra i piú noti Software per <i>data assimilation</i>	88
A.1	Schede tecniche software DA	88
B	Sui Problemi Inversi Mal Posti	94
B.1	Problema Inverso Mal Posto	95
B.2	Decomposizione SVD (<i>Singular Value Decomposition</i>)	106
B.3	Regolarizzazione Numerica	111
B.3.1	Metodo di Regolarizzazione di Tikhonov	112
B.3.2	Decomposizione in Valori Singolari Troncata	114
B.3.3	Approccio variazionale	116
	Bibliografia	122

Introduzione alla Data Assimilation

Negli ultimi 20 anni le metodologie di *data assimilation*, utilizzate inizialmente esclusivamente nei modelli atmosferici, sono diventate parte integrante per lo sviluppo e la validazione dei modelli oceanografici e, piú in generale, dei modelli matematici alla base della meteorologia, climatologia, geofisica, geologia e idrologia (spesso ci si riferisce a tali modelli con il termine di *modelli previsionali* per intendere che si tratta di problemi evolutivi nel tempo).

L'obiettivo che accomuna la comunitá scientifica che si occupa di data assimilation é, in sintesi, la possibilitá di integrare i dati acquisiti sperimentalmente (in vivo) con quelli forniti dai modelli matematici (in vitro) al fine di migliorare la conoscenza dell'ecosistema dell'ambiente circostante. É evidente che il beneficio che deriva da questa sinergia ha effetto sia sui modelli che sulle metodologie di osservazione dei dati, stimolando, lo sviluppo di metodi, algoritmi e software per la data assimilation piú adeguati, ovvero altamente efficienti e affidabili, in grado di far fronte sia ad una quantitá di dati sempre crescente, sia ad una sensibilitá agli errori che inevitabilmente perturbano i dati e i modelli. D'altra parte, le necessitá di rendere le metodologie proprie della data assimilation effettivamente disponibili ed utilizzabili in applicazioni di natura diversa, rende fondamentale la realizzazione e messa a punto di

strumenti software efficaci, ovvero robusti e flessibili. In questo contesto si inserisce il presente lavoro di tesi.

I modelli previsionali sono descritti da equazioni differenziali accoppiate a condizioni iniziali: data una stima dello stato del sistema ad un fissato istante temporale, il modello simula la sua evoluzione in un intervallo di tempo. A tal proposito, é opportuno notare che in effetti il termine “previsione” é usato impropriamente. Talvolta, infatti si parla di *predizioni* piuttosto che di previsioni. Tali predizioni sono ottenute da “simulazioni computazionali” per cui la presenza di errori di varia natura (da quelli noti come errori di semplificazione introdotti dal modello che non può mai essere perfettamente aderente alla reale natura fisica, perché non completamente descrivibile, a quelli introdotti dall’approccio computazionale: discretizzazione, roundoff,...) rende particolarmente importante la scelta delle condizioni iniziali. Infatti, una seppure piccola perturbazione sui dati iniziali può propagarsi sulla soluzione del modello in maniera tale da rendere l’errore sulla soluzione significativamente non trascurabile. Se si tiene conto che tale soluzione viene utilizzata, a sua volta, come condizione iniziale per il calcolo di una nuova soluzione, si innesca una reazione a catena che evidentemente può condurre a simulazioni completamente inaffidabili. Edward Lorenz nel 1960 definí un modello matematico della circolazione dell’aria nell’atmosfera terrestre basata su dodici equazioni. Risolvendo numericamente tale modello Lorenz si convinse che i modelli utilizzati per descrivere i cambiamenti potevano fornire soluzioni imprevedibili. Piccole variazioni dei parametri iniziali del modello sulla terza o quarta cifra significativa producevano enormi variazioni nella soluzione. La dipendenza così marcata dai parametri iniziali prese il nome di *effetto farfalla*: “Puó il batter d’ali di una farfalla in Brasile provocare un tornado in Texas?” fu il titolo di una conferenza all’American Association for the Advancement of Sciences tenuta da Lorenz nel 1979.

In questo contesto di inseriscono le metodologie della data assimilation: correggere periodocamente la condizione iniziale di un modello previsionale utilizzando le informazioni fornite dalle misure sperimentali, ovvero dalle osservazioni dello stato del sistema acquisite nello stesso istante temporale. Così facendo, infatti, si riesce a ridurre l'errore sulla soluzione del modello e sulle successive.

Assimilare i dati (dal latino “assimilare = rendere simile”) significa appunto, in questo contesto, rendere quanto più simile possibile i dati osservati (le misure) con quelli forniti dai modelli di simulazione. Il problema di data assimilation rientra infatti tra quelli di approssimazione di dati (fitting di dati) essendo riconducibile al problema del calcolo della migliore approssimazione.

Nelle prime tecniche di data assimilation (Richardson (1922) e Charney (1950)) l'approssimazione veniva determinata manualmente e chiaramente il procedimento era estremamente lungo e dispendioso. Cressman, nel 1959, fu tra i primi a proporre tecniche di analisi automatica, basate sull'uso dell'interpolazione lineare. Oggi, soprattutto perché le misure sono in una quantità dell'ordine di almeno 10^7 dati, è fondamentale l'uso di metodi, algoritmi e strumenti software ad alte prestazioni altamente efficienti ed affidabili. Attualmente presso i centri di ricerca che si occupano di data assimilation [CMCC (Centro Euro-Mediterraneo per i Cambiamenti Climatici) in Italia, l'NCAR (National Center for Atmospheric Research) e l'NCEP (National Centers for Environmental Prediction) negli Stati Uniti, il DWD (Deutscher Wetterdienst) in Germania, l'UK Met Office del Regno Unito, il JMA (Japan Meteorological Agency) del Giappone, il CMC (Canadian Association of Management Consultants) del Canada...] si sta lavorando alla messa a punto di strumenti software qualificati, tra cui DART (Data Assimilation Research

Testbed), 3DVAR (3D Variational Data Assimilation), OpenDA (Open-source data-assimilation toolbox). In particolare, al CMCC si usa il software OceanVar¹ (Ocean Variational Data Assimilation) sul quale si é rivolta l'attenzione in questo lavoro di tesi, identificando e analizzando i nuclei computazionali piú onerosi per una loro efficiente ottimizzazione, anche in ambiente di calcolo parallelo.

Il lavoro di tesi é articolato come segue.

Nel **Capitolo 1** si descrive il contesto applicativo del problema della data assimilation e si fornisce una breve sintesi delle tecniche utilizzate per la risoluzione di tale problema.

Nel **Capitolo 2** si definisce il problema data assimilation da un punto di vista matematico-numerico fornendone una formulazione come problema inverso mal posto.

Nel **Capitolo 3** si descrive l'applicazione al caso di dati di tipo oceanografico mediante il modello Oceanvar sviluppato per l'assimilazione di dati nello studio dell'evoluzione del Mar Mediterraneo.

Nel **Capitolo 4** si descrivono alcuni software sviluppati presso i piú importanti centri di ricerca che si occupano del problema di data assimilation e, in particolare, si descrive il software OceanVar.

Nel **Capitolo 5** si analizzano i nuclei computazionali piú onerosi del software Oceanvar e i risultati ottenuti dall'ottimizzazione del codice in ambiente di calcolo parallelo.

¹Software sviluppato dal CMCC e messo a disposizione del Prof. A.Murli in qualità di consulente della divisione di Calcolo Scientifico dello stesso centro che é responsabile scientifico della mia tesi insieme alla Prof.ssa L.D'Amore.

Abbreviazioni

DA Data Assimilation

3DVar 3D Variational Data Assimilation

4DVar 4D Variational Data Assimilation

KF Kalman Filter

OI Optimal Interpolation

EnKF Ensemble Kalman Filter

CMCC Centro Euro-Mediterraneo per i Cambiamenti Climatici

NCAR National Center for Atmospheric Research

NCEP National Centers for Environmental Prediction

DWD German Weather Service

UK Met Office del Regno Unito

JMA Japan Meteorological Agency

CMC Canadian Association of Management Consultants

DART Data Assimilation Research Testbed

OpenDA Open-source data-assimilation toolbox

OceanVar Ocean Variational Data Assimilation

Elenco delle Variabili

$\mathbf{M}(\mathbf{P})$ modello previsionale che descrive il problema evolutivo

$\mathbf{M}_n(\mathbf{P})$ modello previsionale discreto

\mathbf{L} operatore che descrive il modello previsionale

$\mathbf{L}_{i,i+1}$ operatore discreto

t_i variabile temporale discreta

\mathbf{x}_{Mi} soluzione del modello previsionale all'istante temporale t_i

y_i dati osservati all'istante temporale t_i

\mathbf{x}_i soluzione del problema di data assimilation all'istante temporale t_i

\mathbf{H} funzionale che lega dati osservati e variabile del modello

e errore sui dati

\mathbf{B} matrice dei pesi dell'errore sulla soluzione del modello previsionale

\mathbf{R} matrice dei pesi dell'errore sui dati osservati

\mathbf{K} matrice di Kalman

\mathbf{J} funzionale del modello 3DVar e 4DVar

Capitolo 1

Contesto Applicativo

Le metodologie di data assimilation si applicano laddove si é interessati allo studio, alla modellazione e quindi alla predizione del comportamento di sistemi evolutivi. Storicamente, la data assimilation si é sviluppata per l'analisi dei due principali ecosistemi terrestri, cioé quello generato dagli oceani e quello generato dall'atmosfera. In altre parole, le applicazioni di data assimilation si trovano prevalentemente in meteorologia e oceanografia. Pertanto, gli strumenti software piú avanzati sono in uso o sono stati sviluppati presso quei centri di ricerca orientati a questo tipo di applicazioni come ad esempio il CMCC (Centro Euro-Mediterraneo per i Cambiamenti Climatici) in Italia, l'NCAR (National Center for Atmospheric Research) e l'NCEP (National Centers for Environmental Prediction) negli Stati Uniti (rispettivamente a Boulder nel Colorado e nel Maryland). Ci sono inoltre il DWD (Deutscher Wetterdienst) che si trova a Offenbach in Germania, l'UK Met Office del Regno Unito, il JMA (Japan Meteorological Agency) del Giappone, il CMC (Canadian Association of Management Consultants) del Canada e altri centri dislocati in tutto il Globo.

Piú in generale, la data assimilation si applica in geofisica, ovvero nello studio di sistemi fluidi geofisici. É piuttosto recente e tuttavia molto

interessante l'estensione a contesti applicativi di biologia e medicina per l'analisi di processi evolutivi nel sangue o di genomi [47].

Indipendentemente dalla specifica applicazione, l'intero processo di analisi e di predizione consta essenzialmente di tre fasi:

1. raccolta di dati osservati

I dati osservati possono essere di vario tipo: rilevazioni da boe o navi, misure da satelliti, etc.

Tali dati sono gestiti in un quadro di cooperazione stabilito e mantenuto dalle Organizzazioni Meteorologiche Mondiali che hanno il comune scopo di regolamentare e garantire osservazioni dello stato dell'atmosfera e dell'oceano. Ogni centro è tenuto a produrre osservazioni proprie e a trasmetterle poi sulle reti di telecomunicazione. Un esempio di rete di telecomunicazione per dati osservati è il GTS (Global Telecommunication System) utilizzata per lo scambio di osservazioni a livello mondiale.

2. determinazione della condizione iniziale (stato iniziale)

È necessario trovare lo stato iniziale più accurato possibile considerando tutti i dati osservati raccolti e a disposizione in una fissata finestra temporale.

3. determinazione della soluzione all'istante corrente (stato "attuale") mediante il Modello Previsionale

risolvendo le equazioni del modello fisico-matematico, noto lo stato iniziale ed eventuali condizioni al contorno, se si tratta di un modello ad area limitata.

Una stima accurata delle condizioni iniziali e un buon modello numerico sono gli elementi fondamentali di un accurato sistema di previsione numerica.

La **Data Assimilation** si utilizza per **produrre il miglior stato iniziale** combinando dati osservati e dati prodotti da previsioni a breve

termine dai modelli previsionali, vedi figura 1.1.

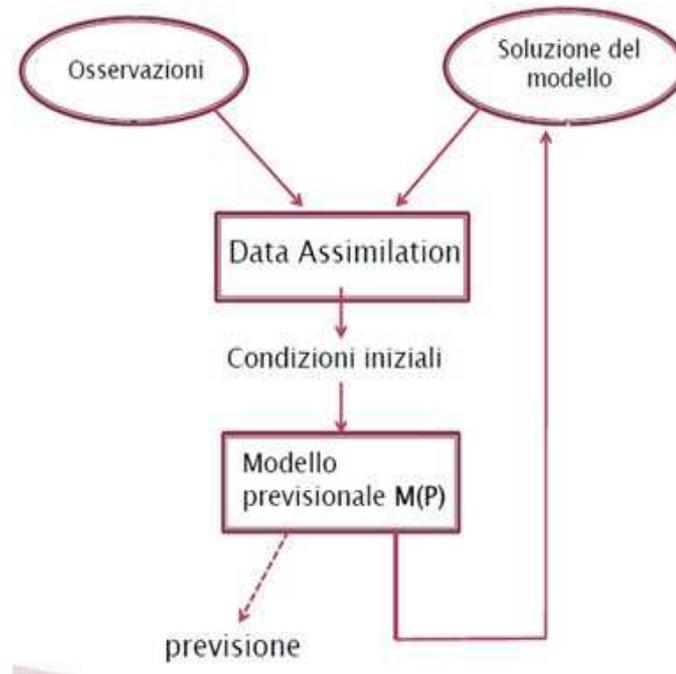


Figura 1.1: problema Data Assimilation

1.1 Relazione tra i dati

Come accennato nel paragrafo precedente, il problema *Data Assimilation* consiste nel produrre lo stato iniziale per il *Modello Previsionale* combinando dati osservati e dati prodotti da previsioni. Per fare ciò é necessario confrontare i due tipi di dati (dati osservati e dati prodotti dai modelli).

Tuttavia, questi sono sostanzialmente caratterizzati da:

- **differenti dimensioni;**
- **differenti grandezze.**

La **differenza di dimensione** si riferisce all'insufficienza dei dati osservati per l'inizializzazione di un modello (Figura 1.2).

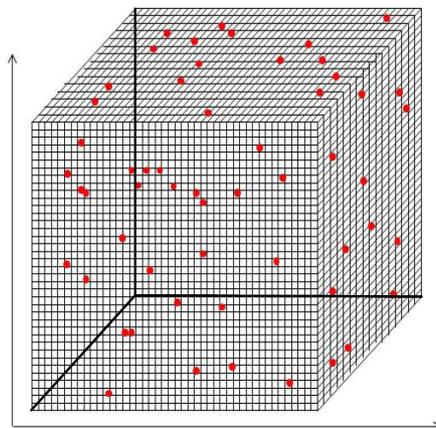


Figura 1.2: i dati osservati (punti rossi) sono sempre puntuali e ne abbiamo una conoscenza in un numero finito di punti nello spazio. Tali punti quasi mai coincidono con i punti del dominio discreto (cubo), sono presenti “lacune”.

Esempio 1.1.1. Consideriamo un modello previsionale espresso in coordinate sferiche (latitudine e longitudine).

Scegliamo una risoluzione del tipo:

- 1° = passo di discretizzazione orizzontale;
- 20 = livelli verticali.

Otteniamo in questo modo un dominio discreto costituito da una griglia di $360 \times 180 \times 20 = 1.3 \times 10^6$ punti.

Per ogni punto del dominio occorre in genere inizializzare almeno quattro variabili (ad esempio due componenti orizzontali del vento, temperatura e umidità), e per ogni colonna, la pressione superficiale. Quindi si hanno circa 5 milioni di variabili da inizializzare. Per ogni passo temporale di circa 6 ore ci sono di solito da 10 mila a 100 mila osservazioni dell'atmosfera, ovvero due ordini di grandezza in meno dei gradi di libertà del

modello. Le informazioni date dalle osservazioni non sono sufficienti per ottenere una stima della condizione iniziale per il modello previsionale.



In genere il numero di condizioni iniziali da specificare in un modello globale é dato dal prodotto tra il numero di variabili e il numero di punti della griglia (che coincide con la dimensione del dominio discreto) ed é tipicamente dell'ordine di $10^6 - 10^7$, mentre il numero di dati osservati disponibili in una finestra temporale di ± 3 ore é di $10^4 - 10^5$, con una distribuzione spaziale molto disuniforme [13].

In generale, poiché le misure date dalle osservazioni sono sempre puntuali abbiamo una conoscenza dei dati osservati in punti discreti nello spazio. Tali punti non sempre coincidono con i punti del dominio discreto, ovvero sono presenti "lacune".

Per ottenere una stima in tutti i punti del dominio discreto si può ricorrere a un'interpolazione di valori osservati sui punti del dominio.

Per confrontare i valori della soluzione del modello con i dati osservati occorre che tali valori siano misure relative al medesimo punto. Occorre quindi definire un funzionale che interpoli i valori della soluzione del modello nei punti in cui vi sono le osservazioni.

Esempio 1.1.2.

Supponiamo di avere un modello previsionale discreto definito su una griglia di punti (e, f, g) come in Fig.1.3

Sia $T = [T_e, T_f, T_g]^T$ l'incognita del nostro modello, dove T_e, T_f, T_g sono, ad esempio, valori della temperatura nei punti della griglia.

Siano inoltre T_1^{oss} e T_2^{oss} due misure nei punti $(1, 2)$ della griglia. Per confrontare i valori delle temperature (calcolati e osservati) occorre che

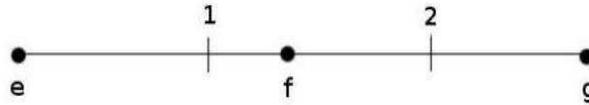


Figura 1.3: Griglia monodimensionale.

tali valori siano misure relative al medesimo punto. A tal fine é necessario introdurre un funzionale $\mathbf{H}(T)$ interpolante i valori (T_e, T_f, T_g) nei tre punti (e, f, g) (Figura 1.4), e valutare tale funzione nei punti $(1, 2)$:

$$\mathbf{H} \begin{bmatrix} T_e \\ T_f \\ T_g \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

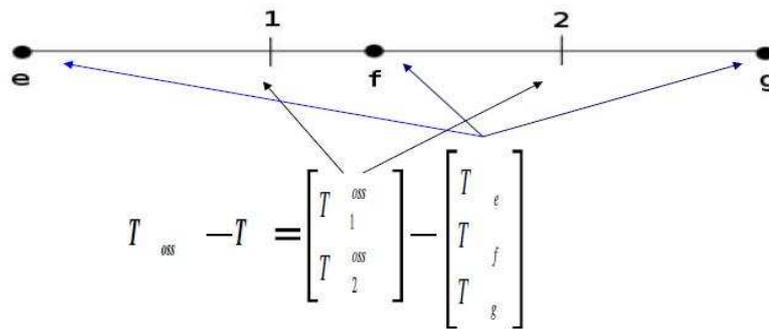


Figura 1.4: valori definiti sui punti della griglia

Se si sceglie un'iterpolazione lineare a tratti si ottiene la matrice:

$$\mathbf{H} = \begin{bmatrix} \frac{T_f - T_1}{T_f - T_e} & \frac{T_1 - T_e}{T_f - T_e} & 0 \\ 0 & \frac{T_g - T_2}{T_g - T_f} & \frac{T_2 - T_f}{T_g - T_f} \end{bmatrix}$$

dove i coefficienti sono ottenuti dall'interpolazione dei valori della temperatura nei punti della griglia (e, f, g) sui punti in cui si hanno le osservazioni $(1, 2)$.



La **differenza di tipologia di grandezza** si riferisce al fatto che un modello previsionale ha molte variabili alcune delle quali spesso non sono direttamente osservabili.

I dati osservati infatti possono essere classificati come

- **Osservazioni dirette o "Convenzionali":**

sono misure delle grandezze incognite del *Modello Previsionale* e sono quantità esplicitamente rappresentate dal vettore di stato. Tipiche osservazioni dirette sono le misurazioni del vento, della temperatura e dell'umidità.

- **Osservazioni non dirette:**

sono misure delle grandezze che devono essere convertite nelle grandezze incognite del *Modello Previsionale*. Un esempio di osservazioni non dirette sono le radiazioni solari misurate dai satelliti.

Per le osservazioni dirette l'operatore osservazione è un operatore interpolante. Per le osservazioni indirette invece l'operatore osservazione può essere molto complesso poiché non solo interpola i valori delle variabili del modello sui punti in cui si hanno le osservazioni, ma converte le variabili del modello in variabili osservate.

Un esempio di operatore per osservazioni indirette è costituito dalle equazioni di trasferimento radiativo che portano i profili verticali di temperatura e umidità in radiazioni osservabili dai satelliti [9].

A seconda del tipo di osservazione che si considera, occorre "costruire"

opportunamente un funzionale H necessario a combinare dati osservati e previsione a breve termine del modello.

La differenza sostanziale, determinata dalla scelta di un tipo di osservazioni piuttosto che altre, é nella dipendenza o meno dalla variabile temporale t del funzionale H .

É noto che tale funzionale può dipendere o meno dalla variabile tempo. In particolare si ha che per le osservazioni dirette, il funzionale H non dipende dalla variabile tempo e che per le osservazioni non dirette, invece, c'è dipendenza del funzionale H dalla variabile tempo.

1.1.1 Tipi di dati osservati

I dati osservati possono essere di vario tipo: stazioni al suolo (SYNOP), radiosondaggi (TEMP), boe (BUOYS), navi, rilevazioni da aerei commerciali e svariati tipi di misure da satellite.

Nel caso delle previsioni oceanografiche le osservazioni possono comprendere:

- dati di anomalia del livello del mare (SLA- Sea Level Anomaly),
- profili verticali di temperatura ottenuti da sonde chiamate XBT (Expandable BathyTermograph),
- acquisizione e elaborazione di profili di temperatura e salinitá forniti dal sistema di boe (ARGO),
- dati da satellite di temperatura superficiale del mare (SST- Sea Surface Temperature).

1.2 Metodi di risoluzione del Problema Data Assimilation (Stato dell'Arte)

Un primo approccio alla risoluzione del problema Data Assimilation é stato suggerito da George P. Cressman (1959) detto Cressman analysis. Il metodo fornisce come approssimazione della condizione iniziale il punto ottenuto effettuando un'interpolazione lineare di punti $\{(a_i, b_i)\}_{i=1, \dots, n}$ definiti in modo tale che le ascisse a_i rappresentano le coordinate della locazione del dominio discreto in cui é definita la soluzione del modello previsionale e le ordinate b_i sono calcolate come segue. Si definisce una sfera di centro uguale al dato osservato e di raggio R detta "sfera di influenza" del dato osservato sulla soluzione del modello previsionale (se la distanza tra i punti in cui ci sono osservazioni e i punti in cui é definita la soluzione del modello é minore di R , si assume che esista un'influenza del dato osservato sul modello) e si definiscono i valori b_i , $i = 1, \dots, n$, tali che:

- b_i coincide con il valore del dato osservato se in a_i questo é presente;
- b_i coincide con il valore della soluzione del modello previsionale se a_i dista dai punti in un cui sono presenti dati osservati di un segmento maggiore di R ;
- b_i coincide con la differenza tra la soluzione del modello previsionale e il dato osservato se a_i dista dai punti in un cui sono presenti dati osservati di un segmento minore di R ;

Nella figura 1.5, la linea nera rappresenta la previsione a breve termine del modello previsionale e i pallini rossi le osservazioni.

Il metodo Cressman Analysis produce la linea a pallini blu (figura 1.5), che coincide con la previsione del modello in punti distanti dalle osservazioni, e si rilassa verso i dati osservati dove sono presenti.

In tale metodo manca il controllo di qualità dei dati e soprattutto non

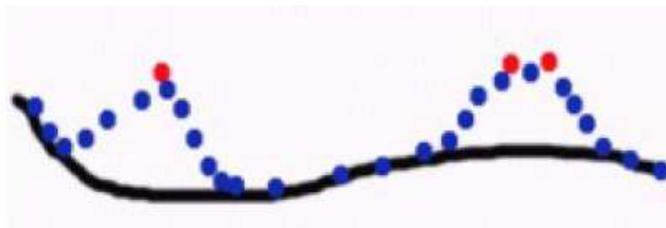


Figura 1.5: Cressman analysis

vengono presi in considerazione i pesi degli errori dei dati osservati e del modello.

Tra i metodi utilizzati vi é l'Optimal Interpolation.

Esempio 1.2.1. *Supponiamo di avere due misure di temperatura T_1 e T_2 che rappresentano rispettivamente un'osservazione e una previsione a breve termine del modello previsionale (ad esempio l'equazione del calore). Indichiamo con σ_1 e σ_2 le stime dell'errore sui dati T_1 e T_2 . Cerchiamo un valore della temperatura T che non si discosti il minimo possibile dai due dati. Pertanto scelgo una valore T che soddisfi la relazione:*

$$T = \min \{a_1^2(T - T_1)^2 + a_2^2(T - T_2)^2\} \quad (1.1)$$

con $a_1 = \frac{1}{\sigma_1}$ e $a_2 = \frac{1}{\sigma_2}$.

Quindi se la misura T_1 ha un grosso errore (σ_1 grande), il peso a_1 é piccolo e quindi nel determinare T ha maggiore importanza la misura T_2 . In pratica il contributo maggiore viene dato dalla componente piú precisa, come é logico che sia.



Le variabili nella (1.1) sono scalari. Se consideriamo tali grandezze come vettori di dimensione uno, la (1.1) puó essere riscritta come:

$$T = \min \{ \|(T - T_1)\|_{a_1}^2 + \|(T - T_2)\|_{a_2}^2 \} \quad (1.2)$$

dove i pesi a_1 e a_2 sono matrici di dimensione (1×1) .

Come mostreremo in dettaglio nel Capitolo 2, il metodo Optimal Interpolation é un metodo di approssimazione nel senso dei minimi quadrati. Tale metodo consente di effettuare prime considerazioni sul peso che la stima dell'errore sui dati deve avere nel calcolo della soluzione fornita dalla data assimilation. Infatti nella (1.1) e nella (1.2), i pesi a_i , $i = 1, 2$, sono calcolati tenendo conto dell'errore sui dati.

L'errore sul dato si stima calcolando la varianza dell'insieme di valori che descrivono il dato. Infatti, assunto che i dati siano affetti da errori distribuiti secondo una gaussiana (Figura 1.6), la varianza σ , per definizione é la distanza tra il valore medio dell'insieme di dati μ e il punto di flesso della curva.

Se assumiamo come dato "piú attendibile" il valore medio, si ha che la varianza, che rappresenta lo "scostamento" dei restanti dati dal valore medio, ci fornisce una stima dello "scostamento" dei dati dal dato "piú attendibile" che é per definizione l'errore.

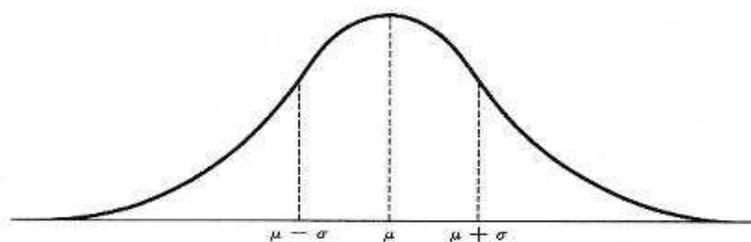


Figura 1.6: Gaussiana

A partire da tale metodo sono stati condotti studi su come selezionare i dati osservati cercando di considerare solo quelli con alta attendibilità cominciando a cercare un modo per coinvolgere il modello previsionale per produrre una stima piú accurata possibile dell'errore sul dato prodotto dalla previsione a breve termine.

I principali metodi di risoluzione del problema data assimilation che includono le problematiche sui pesi dell'errore nei dati sono essenzialmente riconducibili a quattro modelli (modello variazionale 3D **3DVar** e 4D **4DVar**, Filtro di Kalman **KF** e Filtro di Kalman Ensemble **EnKF**) che si differenziano per la presenza o meno della variabile tempo e per il tipo di problema matematico che risolvono.

Nel Capitolo 2 descriviamo tali metodi da un punto di vista matematico numerico e mostriamo i nuclei computazionali che implementano. In questa fase, vogliamo sottolineare aspetti che riguardano maggiormente il contesto applicativo. Abbiamo osservato, ad esempio, che la presenza della variabile tempo nei modelli Data Assimilation e quindi la tendenza ad utilizzare modelli di tipo 4D compare solo negli ultimi anni dettata dalla disponibilità sempre maggiore di dati provenienti da satelliti e quindi dall'introduzione di un funzionale "piú complesso" rispetto al funzionale intepolante utilizzato per le osservazioni dirette.

Ad oggi, non esiste una tendenza che spinga all'utilizzo di un metodo piuttosto che un altro, si é ancora in fase di confronto tra eventuali vantaggi o svantaggi che la scelta di un metodo puó portare.

Non esistono in letteratura criteri che dettano la scelta del metodo di risoluzione.

Abbiamo osservato, a tal proposito, che la scelta del metodo di risoluzione dipende essenzialmente da due fattori sintetizzati nella figura 1.7:

1. Tipo di dato osservato (disponibile e/o scelto) che determina il funzionale;
2. disponibilità di un software che implementa il modello previsionale. Tale possibilità consente una stima piú accurata dell'errore sul dato prodotto dalla previsione a breve termine del modello.

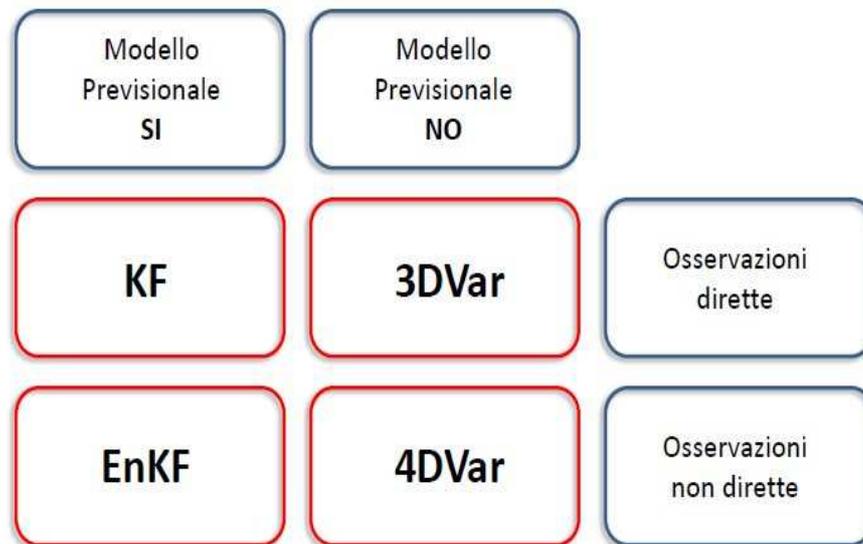


Figura 1.7: criterio di scelta del metodo DA

1.3 Ulteriori Applicazioni

Il problema Data Assimilation trova molte applicazioni anche in ambiti differenti dalle previsioni metereologiche, climatologiche e oceanografiche. Sono un esempio:

- gli studi sulla *Qualità dell'aria sulla superficie del mare* sviluppati per il Mar del Nord.
Regolamenti richiedono che le navi utilizzino combustibile a basso

tenore di zolfo. Per controllare che le navi realmente aderiscano ai regolamenti esiste un sistema che controlla la qualità dell'aria sulla superficie del Mar del Nord. Il sistema è basato su un modello di qualità dell'aria open Source di nome Chimere in cui si utilizzano per la fase di data assimilation sia osservazioni da satellite che osservazioni dalla superficie terrestre.

- il problema delle *Previsioni della qualità dell'aria per pazienti con problemi respiratori*

Nei Paesi Bassi in cui c'è un elevato numero di pazienti che avvertono problemi respiratori, è necessario un continuo controllo e previsione della qualità dell'aria. Tali previsioni sono calcolate mediante il modello dal nome LOTOS_EUROS che si basa su metodo di data assimilation.

- lo *studio delle maree*.

Un esempio è lo studio che riguarda il miglioramento del modello di marea di Singapore. Il flusso di marea intorno a Singapore è piuttosto complesso. La topografia dell'area, il particolare le isole e l'estremità dello stretto di Malacca, causa forti fluttuazioni nella marea. Per simulare correttamente questa fluttuazione mediante un modello numerico occorrono stime delle variabili del modello che siano molto accurate.

Capitolo 2

Data Assimilation: un problema inverso mal posto

La simulazione computazionale basata sui modelli matematici impiegati per descrivere l'evoluzione di un ecosistema é spesso affetta da semplificazioni e approssimazioni inevitabilmente dovuti alla complessità del sistema fisico e quindi alla non disponibilità di una completa conoscenza del sistema stesso. D'altra parte, anche le osservazioni sperimentali dello stato del sistema contengono errori dovuti allo strumento di misura e all'osservazione stessa. Di conseguenza, la predizione dello stato del sistema può scostarsi, nel tempo, in maniera significativa dallo stato del sistema rendendo vana, di fatto, la predizione. Lo scopo della data assimilation é combinare le misure/osservazioni con il modello matematico al fine di derivare stime più affidabili dello stato del sistema insieme con una stima dell'errore introdotto.

Introduciamo il problema dando la seguente definizione di data assimilation per ora in modo semplificato:

Definizione 2.0.1. *Considerato un sistema il cui stato evolve nel tempo, una stima (affetta da errore) dello stato del sistema ad un certo istante di tempo e una misura sperimentale (affetta da errore) dello sta-*

to del sistema in un insieme di istanti temporali, calcolare la migliore approssimazione dello stato del sistema corrente e futuro.

Indichiamo dunque con $M(P)$ il modello matematico che descrive l'evoluzione di un fissato sistema fisico (con il termine fisico ci riferiamo in questo contesto per indicare un qualunque sistema di natura fisica, biologica, economica). $M(P)$ é generalmente un sistema di equazioni differenziali di tipo evolutivo (ovvero dipendente dal tempo) accoppiate a condizioni iniziali ed eventualmente al contorno:

$$\begin{cases} L[x(t)] = f(t), & t \in \Omega \\ x(0) = x_0, & t = 0 \end{cases} \quad (2.1)$$

dove $L : X \rightarrow X'$ é un opportuno operatore definito su un dominio Ω , X e X' sono spazi di Banach, $x(t) : \Omega \rightarrow X$ é la funzione incognita del problema e f é il termine noto.

Per discretizzare il modello $M(P)$ rispetto al tempo, é necessario fissare un passo di discretizzazione h e definire i valori $\{t_i\}_{i=1,2,3,\dots}$ con $t_i - t_{i-1} = h$ con $i = 1, 2, 3, \dots$

Otteniamo, per ogni istante temporale $\{t_i\}_{i=1,2,3,\dots}$, l'operatore discreto $L_{i-1,i}$ e il valore della soluzione numerica del problema che indichiamo con la notazione:

$$x_{Mi}, \quad i = 1, 2, 3, \dots \quad (2.2)$$

a valori nello spazio discreto X_n , dove il pedice n indica la dimensione dello spazio stesso.

In particolare, indichiamo con

$$x_{M(i+1)} = L_{i,i+1}(x_{Mi}), \quad i = 1, 2, 3, \dots \quad (2.3)$$

la soluzione di $M(P)$ ottenuta all'istante temporale $i + 1$ utilizzando come condizione iniziale la soluzione all'istante temporale i , figura 2.1.

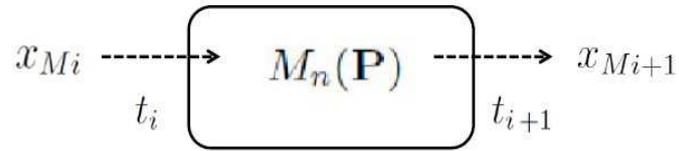


Figura 2.1: modello discreto di un problema di tipo evolutivo

Il valore della soluzione numerica x_{Mi} é affetto da vari tipi di errore¹ che indichiamo con un unico simbolo δ

$$x_{Mi} = x_i + \delta_i \quad (2.4)$$

Tale errore si propaga nella soluzione x_{Mi+1} mediante un fattore di amplificazione $c\mu$ che dipende dall'Algoritmo (mediante c) e dall'indice di condizionamento del problema (mediante μ):

$$x_{Mi+1} = x_{i+1} + \sigma_i \quad (2.5)$$

con

$$\sigma_i = c\mu\delta_i \quad (2.6)$$

Fissato il problema (fissato μ) e fissato l'algoritmo (fissato c), al fine di ridurre l'errore σ_i sulla soluzione é necessario ridurre l'errore δ_i sul dato iniziale x_{Mi} . Ciò é possibile se si utilizzano le informazioni fornite dalle osservazioni sperimentali effettuate allo stesso istante temporale t_i . Lo scopo della Data Assimilation é ridurre δ_i utilizzando una combinazione della soluzione \mathbf{x}_{Mi} e di osservazioni y_i ottenute all'istante t_i , al fine di produrre un dato iniziale \mathbf{x}_i che sia piú accurato (Figura 2.2).

¹errore che si commette nel calcolo della soluzione numerica ed é dovuto alla sostituzione del problema continuo con un problema discreto (*errore di discretizzazione*) e alla risoluzione del problema discreto in un sistema aritmetico floating point a precisione finita (*errore di round-off*), etc.

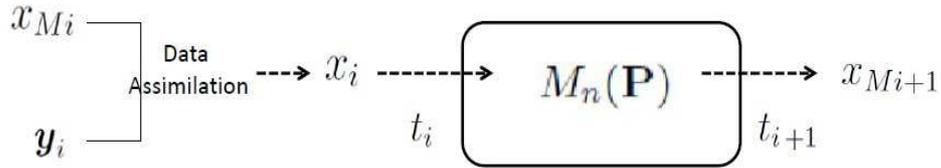


Figura 2.2: modello discreto di un problema di tipo evolutivo con fase di Data Assimilation

L'idea della data assimilation é quella di rendere minima la distanza tra la soluzione x_{Mi} e le osservazioni stesse. Tale problema si caratterizza quindi come un problema di approssimazione di dati (fitting o rappresentazione).

Problema 2.0.1. Formulazione 1

Dati:

1) due vettori (distinti tra loro) x_M e y ;
 calcolare sotto quali condizioni (su x_M e/o su y) la loro distanza euclidea é la minima possibile.

◇

Il problema 2.0.1 é un **Problema di Migliore Approssimazione**. In particolare, la data assimilation si occupa di trovare, a partire dai dati \mathbf{x}_{Mi} e \mathbf{y}_i , un dato \mathbf{x}_i , definito nello spazio \mathbf{X}_n , che sia la migliore approssimazione di \mathbf{x}_{Mi} e \mathbf{y}_i nel senso dei minimi quadrati:

$$x_i = \operatorname{argmin}\{(x_i - x_{Mi})^2, (x_i - y_i)^2\} \quad (2.7)$$

Un problema che si presenta subito, però, é il *problema di confrontabilità dei dati*. Infatti, i valori y_i che rappresentano i dati osservati sono definiti in uno spazio Y_p di dimensione p generalmente diverso da X_n .

La relazione che intercorre tra *soluzione "ideale" di un modello previsionale* x_i e *dati osservati* y_i in un assegnato istante t_i é esprimibile mediante un modello

$$\mathbf{y}_i = \mathbf{H}\mathbf{x}_i \quad i = 1, 2, 3, \dots \quad (2.8)$$

- \mathbf{y}_i é il vettore di *dimensione* p di osservazioni effettuate ad un istante $t = t_i$,
- \mathbf{x}_i é il vettore di *dimensione* $n > p$
- $\mathbf{H} : X_n \rightarrow Y_p$ é un opportuno operatore di *dimensione* $(p \times n)$.

pertanto il problema si caratterizza come segue:

Problema 2.0.2. Formulazione 2

Dati:

- 1) due vettori (distinti tra loro) x_M e y ;
 - 2) supponiamo che y sia funzione di x : $y = H(x)$
- calcolare sotto quali condizioni (su x_M e/o y), la loro distanza euclidea é la minima possibile.

◇

Allora il **Problema di Migliore Approssimazione** 2.0.1 diventa un **Problema Inverso con vincolo** in 2.0.2 e la (2.7) diventa:

$$x_i = \operatorname{argmin}\{(x_i - x_{Mi})^2, (H(x_i) - y_i)^2\} \quad (2.9)$$

La secondo parte della (2.9):

$$x_i = \operatorname{argmin}\{\dots, (H(x_i) - y_i)^2\} \quad (2.10)$$

é un problema inverso e la prima parte:

$$x_i = \operatorname{argmin}\{(x_i - x_{Mi})^2, \dots\} \quad (2.11)$$

rappresenta un vincolo.

Assunta la (2.8), il problema é calcolare \mathbf{x}_i , noti \mathbf{y}_i e \mathbf{H} :

$$\mathbf{x}_i = \mathbf{H}^* \mathbf{y}_i \quad i = 1, 2, 3, \dots \quad (2.12)$$

dove \mathbf{H}^* é l'inversa generalizzata di \mathbf{H} .

Il problema é pertanto un **problema inverso** e poiché $n \neq p$ e $\text{rango}(\mathbf{H}) = p < n$, il problema ammette infinite soluzioni ed é quindi **mal posto**.

Inoltre, l'operatore H é in generale noto solo attraverso sue approssimazioni, sia perché spesso le osservazioni non sono dirette, per cui H non é calcolabile, sia perché é generalmente non lineare e viene linearizzato. Queste approssimazioni introducono errori nella descrizione di H , che evidentemente si propagano nella soluzione x_i . In altre parole, qualora H fosse invertibile, non é possibile invertire H dal momento che nella sua rappresentazione si sono introdotti errori non trascurabili.

Rifacendosi ai minimi quadrati introdotti da Gauss, ci troviamo davanti a un problema sottodeterminato. Infatti, se pure fosse $n = p$ e $\text{rango}(\mathbf{H}) = n$, il problema é del tipo:

$$(\mathbf{H} + \epsilon_1) \mathbf{x}_i = \mathbf{y}_i + \epsilon_2 \quad (2.13)$$

dove ϵ_1 ed ϵ_2 sono gli errori rispettivamente su \mathbf{H} e su \mathbf{y}_i che non essendo "noti", costituiscono incognite che vanno a sommarsi ad \mathbf{x}_i rendendo il problema sottodeterminato.

La formulazione del problema della data assimilation come problema inverso mal posto ha consentito di inquadrare tutte le metodologie di risoluzione, attualmente utilizzate per la data assimilation, nell'ambito dei metodi di regolarizzazione.

2.1 Regolarizzazione del problema di Data Assimilation

Il piú diffuso metodo di regolarizzazione é il *metodo dei minimi quadrati di Gauss* che consiste nel calcolare una soluzione regolarizzata \mathbf{x}_i come soluzione del problema

$$\min \left\{ \|\mathbf{H}\mathbf{x}_i - \mathbf{y}_i\|_2^2 \right\} \quad (2.14)$$

dove la notazione $\|\cdot\|_2^2$ indica il quadrato della norma L^2 . In forma estesa:

$$\min \left\{ \sum_{h=1}^p \left(\mathbf{H}(h)x_i(h) - y_i(h) \right)^2 \right\}. \quad (2.15)$$

Ipotizzando che le componenti $y_i(h)$, ($h = 1, \dots, p$) del vettore \mathbf{y}_i siano dati ottenuti da misure, eventualmente soggetti ad errore, possiamo considerare il metodo dei minimi quadrati che tiene conto delle misure degli errori sui dati. Precisamente, é possibile calcolare una soluzione regolarizzata \mathbf{x}_i risolvendo il problema

$$\min \left\{ \sum_{h=1}^p e_{hh} \left(\mathbf{H}(h)x_i(h) - y_i(h) \right)^2 \right\} \quad (2.16)$$

dove e_{hh} é un "peso" dell'errore sulla componente $y_i(h)$ del vettore \mathbf{y}_i , per ogni $h = 1, \dots, p$.

Indicata con:

$$\mathbf{R} = \begin{bmatrix} e_{11} & 0 & 0 & \cdots & 0 \\ 0 & e_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & e_{pp} \end{bmatrix} \quad (2.17)$$

una matrice diagonale i cui elementi sono i valori e_{hh} , riscriviamo la (2.16) in forma compatta:

$$\min \left\{ \|\mathbf{H}\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{R}}^2 \right\} \quad (2.18)$$

dove la notazione $\|\cdot\|_{\mathbf{R}}^2$ indica il quadrato della norma L^2 pesata² rispetto alla matrice \mathbf{R} .

Il problema predetto non tiene conto delle informazioni che ci sono fornite dal modello previsionale $M(P)$, o meglio, non tiene conto della propagazione dell'errore che avviene nel modello. Per ovviare a questa mancanza, si utilizza il metodo di regolarizzazione di Tikhonov (vedi Appendice) che consiste nel calcolare la soluzione regolarizzata \mathbf{x}_i come soluzione del problema

$$\min \left\{ \|\mathbf{H}\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{R}}^2 + \|\mathbf{x}_i - \mathbf{x}_{M_i}\|_{\mathbf{B}}^2 \right\} \quad (2.19)$$

dove \mathbf{R} e \mathbf{B} sono rispettivamente la matrice dei pesi \mathbf{R} (definita in (2.17)) e una matrice \mathbf{B} che fornisce una stima dell'errore nei dati $\{x_i(k)\}_{k=1, \dots, n}$. Esplicitando:

$$\min \left\{ \sum_{h=1}^p e_{hh} \left(\mathbf{H}(h)x_i(h) - y_i(h) \right)^2 + \sum_{k, k^*=1}^n e'_{kk^*} \left(x_i(k) - x_{M_i}(k^*) \right)^2 \right\}. \quad (2.20)$$

dove:

$$\mathbf{B} = \{e'_{kk^*}\}_{k, k^*=1, \dots, n} = \begin{bmatrix} e'_{11} & e'_{12} & e'_{13} & \cdots & e'_{1n} \\ e'_{21} & e'_{22} & e'_{23} & \cdots & e'_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e'_{n1} & e'_{n2} & e'_{n3} & \cdots & e'_{nn} \end{bmatrix} \quad (2.21)$$

Il problema (2.19) può essere risolto mediante *equazioni normali* o mediante *metodo dei moltiplicatori di Lagrange*.

²Sia \mathbf{z} un vettore di dimensione n , $\mathbf{z} = \{z(k)\}_{k=1, \dots, n}$, si ha: $\|\mathbf{z}\|_{\mathbf{R}}^2 = \frac{1}{2} \sum_{k=1}^n e_{kk} z(k)^2$, oppure in forma vettoriale: $\|\cdot\|_{\mathbf{R}}^2 = \mathbf{z}^T \mathbf{R} \mathbf{z}$.

Mostreremo come risolvendo il problema (2.19) mediante i due approcci si ottengono alcuni tra i principali metodi di risoluzione del problema *Data Assimilation*:

1. Optimal Interpolation e Filtro di Kalman che otteniamo risolvendo la (2.19) mediante equazioni normali;
2. Approccio Variazionale 3D che otteniamo risolvendo la (2.19) mediante il metodo dei moltiplicatori di Lagrange.

Inoltre, come mostrato in appendice, risolvere un problema mediante le *equazioni normali* oppure mediante il *metodo dei moltiplicatori di Lagrange* conduce alla stessa soluzione numerica. La scelta di un approccio piuttosto che un altro spesso é dettata soltanto da considerazioni di tipo qualitativo di cui si é discusso nel Capitolo 1 che sono legate al problema reale di cui ci si occupa.

Nell'equazione (2.19) la variabile tempo é fissa ad un istante t_i . Talvolta, per quanto detto nel Capitolo 1, può risultare necessario introdurre la variabile tempo nel modello di regolarizzazione utilizzando dati osservati ottenuti il piú istanti temporali $\{t_j\}_{j=i,\dots,i+m}$ e risolvendo pertanto il problema:

$$\min \left\{ \sum_{j=i}^{i+m} \|\mathbf{H}\mathbf{x}_j - \mathbf{y}_j\|_{\mathbf{R}_j}^2 + \|\mathbf{x}_i - \mathbf{x}_{M_i}\|_{\mathbf{B}}^2 \right\} \quad (2.22)$$

In questo caso i metodi risolutivi possono essere gli stessi, ovvero la 2.22 può essere risolta mediante *equazioni normali* o mediante *metodo dei moltiplicatori di Lagrange*. In questo caso si ottengono i metodi di risoluzione del problema *Data Assimilation*:

1. Filtro di Kalman Ensemble che otteniamo risolvendo la (2.22) mediante equazioni normali;
2. Approccio Variazionale 4D che otteniamo risolvendo la (2.22) mediante il metodo dei moltiplicatori di Lagrange.

2.2 Problema di Minimi Quadrati con vincoli (Equazioni normali)

2.2.1 Optimal Interpolation e Filtro di Kalman (KF): un solo passo temporale

Fissato un istante temporale t_i , sia:

$$S = \|\mathbf{H}\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{R}}^2 + \|\mathbf{x}_i - \mathbf{x}_{\mathbf{M}i}\|_{\mathbf{B}}^2 \quad (2.23)$$

dove

- $\mathbf{x}_{\mathbf{M}i}$ é la soluzione del modello previsionale all'istante t_i
- \mathbf{y}_i é il vettore di dati osservati all'istante t_i
- \mathbf{x}_i é l'incognita.

Risolvere il problema $\min(S)$ dato dalla (2.19), equivale a calcolare la soluzione dell'equazione:

$$\frac{\partial S}{\partial \mathbf{x}_i} = 0. \quad (2.24)$$

Proposizione 2.2.1. *Sia*

$$S = (\mathbf{H}\mathbf{x}_i - \mathbf{y}_i)^T \mathbf{R}(\mathbf{H}\mathbf{x}_i - \mathbf{y}_i) + (\mathbf{x}_i - \mathbf{x}_{\mathbf{M}i})^T \mathbf{B}(\mathbf{x}_i - \mathbf{x}_{\mathbf{M}i}), \quad (2.25)$$

l'equazione (2.24) ammette una soluzione:

$$\mathbf{x}_i = \mathbf{x}_{\mathbf{M}i} + \mathbf{K}(\mathbf{y}_i - \mathbf{H}\mathbf{x}_{\mathbf{M}i}). \quad (2.26)$$

dove

$$\mathbf{K} = \mathbf{R}\mathbf{H}^T(\mathbf{H}^T\mathbf{R}\mathbf{H} + \mathbf{B})^{-1} \quad (2.27)$$

dim:

Effettuando i prodotti nella (2.25):

$$S = (\mathbf{H}\mathbf{x}_i)^T \mathbf{R}(\mathbf{H}\mathbf{x}_i) - \mathbf{y}_i^T \mathbf{R}\mathbf{H}\mathbf{x}_i - (\mathbf{H}\mathbf{x}_i)^T \mathbf{R}\mathbf{y}_i + \mathbf{y}_i^T \mathbf{R}\mathbf{y}_i + \mathbf{x}_i^T \mathbf{B}\mathbf{x}_i + \\ - \mathbf{x}_i^T \mathbf{B}\mathbf{x}_i - \mathbf{x}_i^T \mathbf{B}\mathbf{x}_{M_i} + \mathbf{x}_i^T \mathbf{B}\mathbf{x}_{M_i} \quad (2.28)$$

Derivando la (2.28) rispetto a \mathbf{x}_i ; si ottiene:

$$2\mathbf{H}^T \mathbf{R}\mathbf{H}\mathbf{x}_i - \mathbf{y}_i^T \mathbf{R}\mathbf{H} - \mathbf{H}^T \mathbf{R}\mathbf{y}_i + 2\mathbf{B}\mathbf{x}_i - \mathbf{x}_{M_i}^T \mathbf{B} - \mathbf{B}\mathbf{x}_{M_i} = 0 \quad (2.29)$$

da cui:

$$2(\mathbf{H}^T \mathbf{R}\mathbf{H} + \mathbf{B})\mathbf{x}_i = \mathbf{y}_i^T \mathbf{R}\mathbf{H} + \mathbf{H}^T \mathbf{R}\mathbf{y}_i + \mathbf{x}_{M_i}^T \mathbf{B} + \mathbf{B}\mathbf{x}_{M_i} \quad (2.30)$$

Poiché le matrici \mathbf{R} e \mathbf{B} sono simmetriche e definite positive, i termini al secondo membro sono a due a due coincidenti. Per cui la (2.30) può essere riscritta come:

$$2(\mathbf{H}^T \mathbf{R}\mathbf{H} + \mathbf{B})\mathbf{x}_{M_i} = 2\mathbf{H}^T \mathbf{R}\mathbf{y}_i + 2\mathbf{B}\mathbf{x}_{M_i}. \quad (2.31)$$

Aggiungendo e sottraendo il termine $\mathbf{H}^T \mathbf{R}\mathbf{H}\mathbf{x}_{M_i}$ si ottiene:

$$(\mathbf{H}^T \mathbf{R}\mathbf{H} + \mathbf{B})\mathbf{x}_i = \mathbf{H}^T \mathbf{R}\mathbf{y}_i + \mathbf{H}^T \mathbf{R}\mathbf{H}\mathbf{x}_{M_i} - \mathbf{H}^T \mathbf{R}\mathbf{H}\mathbf{x}_{M_i} + \mathbf{B}\mathbf{x}_{M_i} \quad (2.32)$$

da cui:

$$(\mathbf{H}^T \mathbf{R}\mathbf{H} + \mathbf{B})\mathbf{x}_i = (\mathbf{H}^T \mathbf{R}\mathbf{H} + \mathbf{B})\mathbf{x}_{M_i} - \mathbf{H}^T \mathbf{R}\mathbf{H}\mathbf{x}_{M_i} + \mathbf{H}^T \mathbf{R}\mathbf{y}_i \quad (2.33)$$

Esplicitando l'espressione della soluzione \mathbf{x}_i si ha:

$$\mathbf{x}_i = \mathbf{x}_{M_i} + (\mathbf{H}^T \mathbf{R}\mathbf{H} + \mathbf{B})^{-1}(-\mathbf{H}^T \mathbf{R}\mathbf{H}\mathbf{x}_{M_i} + \mathbf{H}^T \mathbf{R}\mathbf{y}_i). \quad (2.34)$$

Mettendo in evidenza il termine $\mathbf{H}^T \mathbf{R}$ si ottiene l'espressione:

$$\mathbf{x}_i = \mathbf{x}_{M_i} + (\mathbf{H}^T \mathbf{R} \mathbf{H} + \mathbf{B}^{-1} \mathbf{H}^T \mathbf{R})(\mathbf{y}_i - \mathbf{H} \mathbf{x}_{M_i}) \quad (2.35)$$

o anche:

$$\mathbf{x}_i = \mathbf{x}_{M_i} + \mathbf{R} \mathbf{H}^T (\mathbf{H}^T \mathbf{R} \mathbf{H} + \mathbf{B})^{-1} (\mathbf{y}_i - \mathbf{H} \mathbf{x}_{M_i}) \quad (2.36)$$

Posto:

$$\mathbf{K} = \mathbf{R} \mathbf{H}^T (\mathbf{H}^T \mathbf{R} \mathbf{H} + \mathbf{B})^{-1}$$

si ha

$$\mathbf{x}_i = \mathbf{x}_{M_i} + \mathbf{K}(\mathbf{y}_i - \mathbf{H} \mathbf{x}_{M_i}).$$



Le equazioni (2.26) e (2.27) sono alla base dei Metodi Optimal Interpolation e Filtro di Kalman.

In questa fase della trattazione del problema abbiamo parlato della presenza di matrici di pesi \mathbf{R} e \mathbf{B} , ma non abbiamo specificato come gli elementi di tali matrici sono calcolati.

In particolare, la differenza tra i metodi Optimal Interpolation e Filtro di Kalman risiede appunto nel modo in cui le matrici \mathbf{R} e \mathbf{B} sono costruite. Più precisamente nell'Optimal interpolation si calcola l'errore nei dati \mathbf{x}_M mediante elaborazioni di tipo statistico sui campioni di dati. In tale fase non si tiene conto della natura dei dati, ovvero non si tiene conto del *Modello Previsionale* e di come, applicando il modello, ci sia una propagazione dell'errore.

Invece il Filtro di Kalman calcola l'errore nei dati \mathbf{x}_M mediante elaborazioni di tipo statistico sui campioni di dati ma tenendo conto del

modello previsionale, ovvero, tenendo conto dell'amplificazione dell'errore nel *Modello Previsionale*.

Si ipotizza che l'errore nei dati $\{x_i(k)\}_{k=1,\dots,n}$ dipenda concorrentemente dall'errore sui dati $\{x_{i-1}(k)\}_{k=1,\dots,n}$ calcolati nell'istante temporale precedente e da come quest'errore si propaga applicando il modello evolutivo. Quindi, si assume che gli elementi della matrice \mathbf{B}_i , che indichiamo con σ_i , si ottengono applicando il modello evolutivo agli elementi σ_{i-1} calcolati al passo precedente, che esprimono i pesi per gli errori sui dati $\{x_{i-1}(k)\}_{k=1,\dots,n}$ (vedi Figura 2.3)

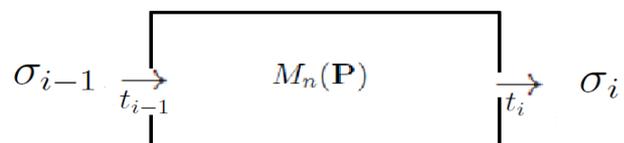


Figura 2.3: elementi della matrice \mathbf{B}_i

Il processo che si genera può essere sintetizzato nell'immagine 2.4.

Sostituendo nell'immagine 2.4 le espressioni delle equazioni della soluzione, si ottiene l'immagine 2.5.

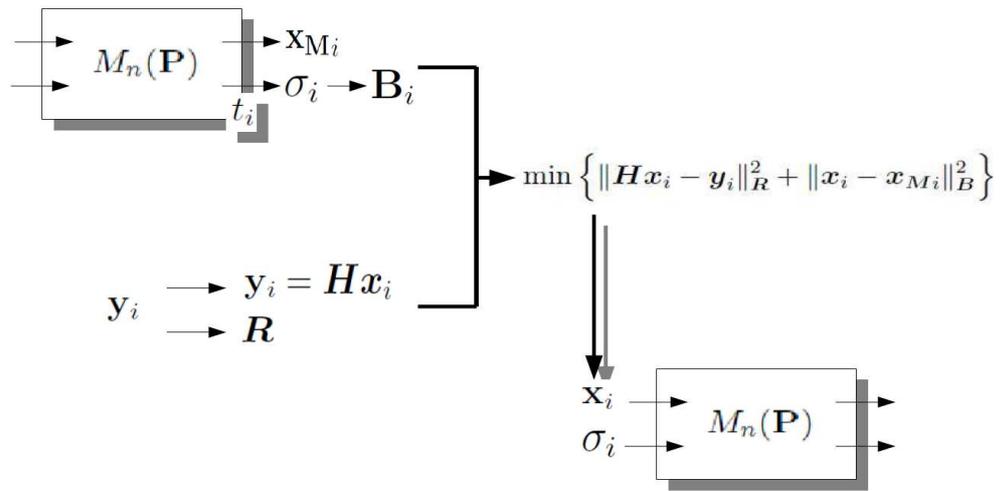


Figura 2.4:

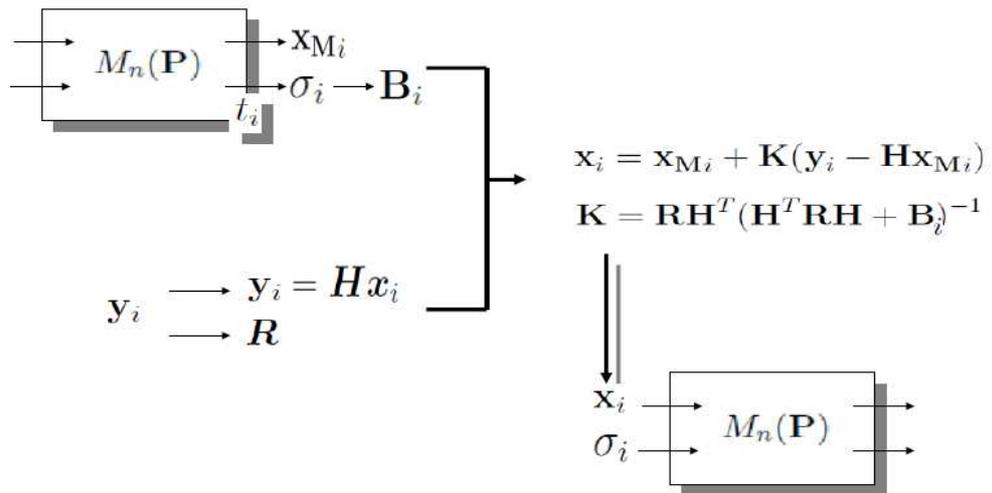


Figura 2.5: filtro di Kalman

2.2.2 Filtro di Kalman Ensemble (EnKF): m passi temporali

Fissato un istante temporale t_i , sia:

$$S = \sum_{j=i}^{i+m} \|\mathbf{H}\mathbf{x}_j - \mathbf{y}_j\|_{\mathbf{R}_j}^2 + \|\mathbf{x}_i - \mathbf{x}_{\mathbf{M}_i}\|_{\mathbf{B}}^2 \quad (2.37)$$

dove ancora

- $\mathbf{x}_{\mathbf{M}_i}$ é la soluzione del modello previsionale all'istante t_i
- \mathbf{y}_j , ($j=i, \dots, i+m$) sono vettori di dati osservati agli istanti t_i, \dots, t_{i+m}
- \mathbf{x}_i é l'incognita.

Risolvere il problema $\min(S)$ con S definita nella (2.37), equivale a calcolare la soluzione dell'equazione:

$$\frac{\partial S}{\partial \mathbf{x}} = 0. \quad (2.38)$$

Proposizione 2.2.2. *Sia*

$$S = \sum_{j=i}^{i+m} \{(\mathbf{H}\mathbf{x}_j - \mathbf{y}_j)^T \mathbf{R}_j (\mathbf{H}\mathbf{x}_j - \mathbf{y}_j)\} + (\mathbf{x}_i - \mathbf{x}_{\mathbf{M}_i})^T \mathbf{B} (\mathbf{x}_i - \mathbf{x}_{\mathbf{M}_i}), \quad (2.39)$$

l'equazione (2.38) ammette una soluzione:

$$\mathbf{x}_i = \mathbf{x}_{\mathbf{M}_i} + \sum_{j=i}^{i+m} \mathbf{K}_j (\mathbf{y}_j - \mathbf{H}\mathbf{x}_{\mathbf{M}_i}). \quad (2.40)$$

dove

$$\mathbf{K}_j = \mathbf{R}_j \mathbf{H}^T (\mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B})^{-1} \quad (2.41)$$

dim.:

Effettuando i prodotti nella (2.39):

$$S = \sum_{j=i}^{i+m} \{(\mathbf{H}\mathbf{x}_j)^T \mathbf{R}_j (\mathbf{H}\mathbf{x}_j) - \mathbf{y}_j^T \mathbf{R}_j \mathbf{H}\mathbf{x}_j - (\mathbf{H}\mathbf{x}_j)^T \mathbf{R}_j \mathbf{y}_j + \mathbf{y}_j^T \mathbf{R}_j \mathbf{y}_j\} + \mathbf{x}_i^T \mathbf{B}\mathbf{x}_i + \mathbf{x}_i^T \mathbf{B}\mathbf{x}_{M_i} - \mathbf{x}_i^T \mathbf{B}\mathbf{x}_{M_i} + \mathbf{x}_i^T \mathbf{B}\mathbf{x}_{M_i} \quad (2.42)$$

Deriviamo la (2.42) rispetto a \mathbf{x} ; si ottiene:

$$\sum_{j=i}^{i+m} \{2\mathbf{H}^T \mathbf{R}_j \mathbf{H}\mathbf{x}_j - \mathbf{y}_j^T \mathbf{R}_j \mathbf{H} - \mathbf{H}^T \mathbf{R}_j \mathbf{y}_j\} + 2\mathbf{B}\mathbf{x}_i - \mathbf{x}_{M_i}^T \mathbf{B} - \mathbf{B}\mathbf{x}_{M_i} = 0 \quad (2.43)$$

da cui:

$$2\left(\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B}\right)\mathbf{x}_i = \sum_{j=i}^{i+m} \{\mathbf{y}_j^T \mathbf{R}_j \mathbf{H} + \mathbf{H}^T \mathbf{R}_j \mathbf{y}_j\} + \mathbf{x}_{M_i}^T \mathbf{B} + \mathbf{B}\mathbf{x}_{M_i} \quad (2.44)$$

Poiché le matrici \mathbf{R}_j e \mathbf{B} sono simmetriche e definite positive, i termini al secondo membro sono a due a due coincidenti. Per cui la (2.44) può essere riscritta come:

$$2\left(\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B}\right)\mathbf{x}_{M_i} = 2\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{y}_j + 2\mathbf{B}\mathbf{x}_{M_i}. \quad (2.45)$$

Aggiungendo e sottraendo il termine $\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H}\mathbf{x}_{M_i}$ si ottiene:

$$\left(\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B}\right)\mathbf{x}_i = \sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{y}_j + \sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H}\mathbf{x}_{M_i} - \sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H}\mathbf{x}_{M_i} + \mathbf{B}\mathbf{x}_{M_i} \quad (2.46)$$

da cui:

$$\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B}\mathbf{x}_i = \left(\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B}\right)\mathbf{x}_{M_i} - \sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H}\mathbf{x}_{M_i} + \sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{y}_j \quad (2.47)$$

Esplicitando quindi l'espressione della soluzione \mathbf{x}_i si ha:

$$\mathbf{x}_i = \mathbf{x}_{M_i} + \left(\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B} \right)^{-1} \left(- \sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H} \mathbf{x}_{M_i} + \sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{y}_j \right). \quad (2.48)$$

Mettendo in evidenza il termine $\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j$ si ottiene l'espressione:

$$\mathbf{x}_i = \mathbf{x}_{M_i} + \left(\sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B} \right)^{-1} \sum_{j=i}^{i+m} \mathbf{H}^T \mathbf{R}_j (\mathbf{y}_j - \mathbf{H} \mathbf{x}_{M_i}) \quad (2.49)$$

o anche:

$$\mathbf{x}_i = \mathbf{x}_{M_i} + \sum_{j=i}^{i+m} \mathbf{R}_j \mathbf{H}^T (\mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B})^{-1} (\mathbf{y}_j - \mathbf{H} \mathbf{x}_{M_i}) \quad (2.50)$$

Posto:

$$\mathbf{K}_j = \mathbf{R}_j \mathbf{H}^T (\mathbf{H}^T \mathbf{R}_j \mathbf{H} + \mathbf{B})^{-1}$$

si ha

$$\mathbf{x}_i = \mathbf{x}_{M_i} + \sum_{j=i}^{i+m} \mathbf{K}_j (\mathbf{y}_j - \mathbf{H} \mathbf{x}_{M_i}).$$



Le equazioni (2.40) e (2.41) sono alla base del Filtro di Kalman Ensemble.

Il processo che si genera può essere sintetizzato nell'immagine 2.6.

Sostituendo nell'immagine 2.4 le espressioni delle equazioni della soluzione, si ottiene l'immagine 2.7.

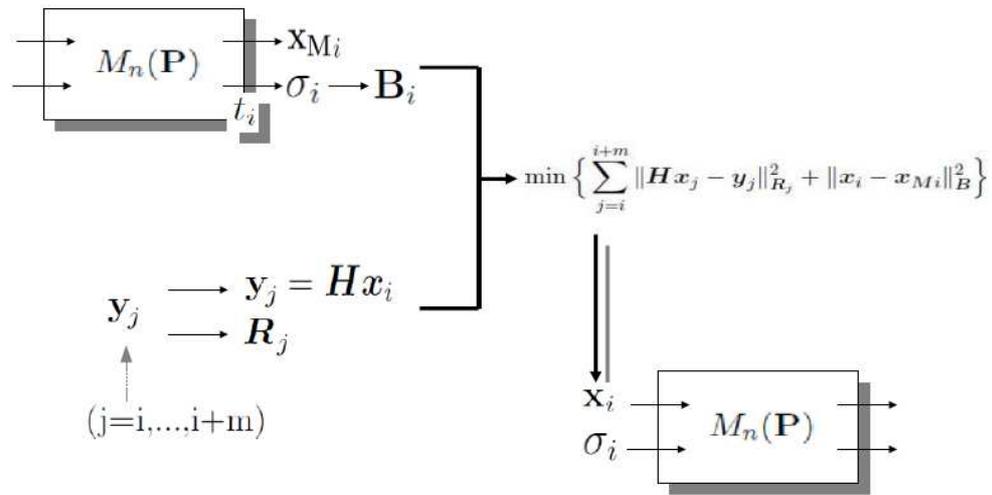


Figura 2.6: ciclo del filtro di Kalman Ensemble

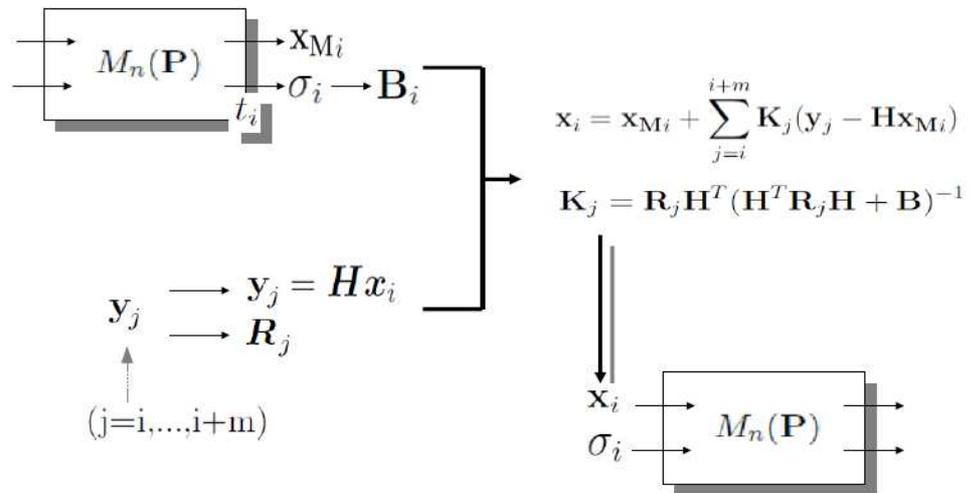


Figura 2.7: equazioni del filtro di Kalman Ensemble

2.3 Problema dei moltiplicatori di Lagrange (Approccio variazionale)

Sia

$$J = \|\mathbf{H}\mathbf{x}_i - \mathbf{y}_i\|_{\mathbf{R}}^2 + \|\mathbf{x}_i - \mathbf{x}_{\mathbf{M}i}\|_{\mathbf{B}}^2$$

nel caso di un solo passo temporale, oppure

$$J = \sum_{j=i}^{i+m} \|\mathbf{H}\mathbf{x}_j - \mathbf{y}_j\|_{\mathbf{R}_j}^2 + \|\mathbf{x}_i - \mathbf{x}_{\mathbf{M}i}\|_{\mathbf{B}}^2$$

nel caso di piú passi temporali. L'Approccio Variazionale consiste nel risolvere il problema

$$\min_{x_i} J(x_i). \quad (2.51)$$

Il problema (2.51) é in genere risolto utilizzando il Metodo Quasi Newton.

I metodi di risoluzione del problema Data Assimilation che si basano su un approccio di tipo variazionale sono noti come 3DVar e 4DVar e si differenziano per l'assunzione o meno della variabile tempo.

Osservazione 2.3.1. *Le matrici che intervengono nel problema hanno dimensioni molto grandi. Spesso, per ridurre la dimensione del problema, si utilizzano metodi dal nome "Low order Incremental"[36] che consistono nella definizione di una nuova variabile \mathbf{v}_i ottenuta da \mathbf{x}_i mediante un'opportuna trasformazione che dipende dal caso specifico.*

Indichiamo con il termine **spazio fisico**, lo spazio in cui é definita la variabile \mathbf{x}_i del modello e con il termine **spazio di controllo**, lo spazio in cui é definta la variabile \mathbf{v}_i .

2.3.1 3DVar: un solo passo temporale

Fissato un istante temporale t_i , lo schema 3D-Var calcola il minimo della funzione

$$J(\mathbf{x}_i) = \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_{M_i})^T \mathbf{B}^{-1}(\mathbf{x}_i - \mathbf{x}_{M_i}) + \frac{1}{2}(H[\mathbf{x}_i] - \mathbf{y}_i)^T \mathbf{R}^{-1}(H[\mathbf{x}_i] - \mathbf{y}_i) \quad (2.52)$$

dove ancora

- \mathbf{x}_{M_i} é la soluzione del modello previsionale all'istante t_i
- \mathbf{y}_i é il vettore di dati osservati all'istante t_i
- \mathbf{x}_i é l'incognita.

Proposizione 2.3.1. *Risolvere il problema 2.51 con $J(x_i)$ dato dalla 2.52 definito nello spazio fisico, equivale a risolvere il problema*

$$\min_{v_i} J(v_i) \quad (2.53)$$

con

$$J(\mathbf{v}_i) = \frac{1}{2}\mathbf{v}_i^T \mathbf{v}_i + \frac{1}{2}(\mathbf{H}\mathbf{V}\mathbf{v}_i - \mathbf{d}_i)^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{V}\mathbf{v}_i - \mathbf{d}_i) \quad (2.54)$$

definito nello spazio di controllo.

dim:

Posto

$$\delta \mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{M_i} \quad (2.55)$$

si ha che:

$$H[\mathbf{x}_i] = H[\mathbf{x}_{M_i} - \mathbf{x}_{M_i} + \mathbf{x}_i] = H[\mathbf{x}_{M_i} + \delta\mathbf{x}_i] \cong H[\mathbf{x}_{M_i}] + \mathbf{H}\delta\mathbf{x}_i. \quad (2.56)$$

Sostituendo le espressioni (2.55) e (2.56) nell'equazione (2.52), il funzionale $J(\mathbf{x}_i)$ assume la seguente forma:

$$\begin{aligned} J(\mathbf{x}_i) &= \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_{M_i})^T \mathbf{B}^{-1}(\mathbf{x}_i - \mathbf{x}_{M_i}) + \frac{1}{2}(H[\mathbf{x}_i] - \mathbf{y}_i)^T \mathbf{R}^{-1}(H[\mathbf{x}_i] - \mathbf{y}_i) = \\ &= \frac{1}{2}\delta\mathbf{x}_i^T \mathbf{B}^{-1}\delta\mathbf{x}_i + \frac{1}{2}(H[\mathbf{x}_{M_i} + \delta\mathbf{x}_i] - \mathbf{y}_i)^T \mathbf{R}^{-1}(H[\mathbf{x}_{M_i} + \delta\mathbf{x}_i] - \mathbf{y}_i) \\ &\cong \frac{1}{2}\delta\mathbf{x}_i^T \mathbf{B}^{-1}\delta\mathbf{x}_i + \frac{1}{2}(H[\mathbf{x}_{M_i}] + \mathbf{H}\delta\mathbf{x}_i - \mathbf{y}_i)^T \mathbf{R}^{-1}(H[\mathbf{x}_{M_i}] + \mathbf{H}\delta\mathbf{x}_i - \mathbf{y}_i) \\ &\cong \frac{1}{2}\delta\mathbf{x}_i^T \mathbf{B}^{-1}\delta\mathbf{x}_i + \frac{1}{2}(\mathbf{H}\delta\mathbf{x}_i - \mathbf{d}_i)^T \mathbf{R}^{-1}(\mathbf{H}\delta\mathbf{x}_i - \mathbf{d}_i) \end{aligned} \quad (2.57)$$

dove $\mathbf{d}_i = [\mathbf{y}_i - H[\mathbf{x}_{M_i}]]$.

Poiché \mathbf{B} é una matrice simmetrica definita positiva, possiamo effettuare la fattorizzazione di Cholesky $\mathbf{B} = \mathbf{V}\mathbf{V}^T$.

Posto $\mathbf{v}_i = \mathbf{V}^+\delta\mathbf{x}_i$, con \mathbf{V}^+ inversa generalizzata di \mathbf{V} la funzione può essere riscritta come segue:

$$J(\mathbf{v}_i) = \frac{1}{2}\mathbf{v}_i^T \mathbf{v}_i + \frac{1}{2}(\mathbf{H}\mathbf{V}\mathbf{v}_i - \mathbf{d}_i)^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{V}\mathbf{v}_i - \mathbf{d}_i) \quad (2.58)$$

♠

2.3.2 4DVar: m passi temporali

Fissato un istante temporale t_i , lo schema 4D-Var calcola il minimo della funzione

$$J(\mathbf{x}_i) = \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_{M_i})^T \mathbf{B}^{-1}(\mathbf{x}_i - \mathbf{x}_{M_i}) + \frac{1}{2} \sum_{j=0}^m (H^j[\mathbf{x}_i] - \mathbf{y}_i^j)^T \mathbf{R}^{j-1} (H^j[\mathbf{x}_i] - \mathbf{y}_i^j) \quad (2.59)$$

dove ancora

- \mathbf{x}_{M_i} é la soluzione del modello previsionale all'istante t_i
- \mathbf{y}_j , ($j=i, \dots, i+m$) sono vettori di dati osservati agli istanti t_i, \dots, t_{i+m}
- \mathbf{x}_i é l'incognita.

Proposizione 2.3.2. *Sia $J(\mathbf{x}_i)$ dato dalla 2.59, risolvere il problema 2.51 definito nello spazio fisico, equivale a risolvere il problema*

$$\min_{v_i} J(v_i) \quad (2.60)$$

con

$$J(\mathbf{v}_i) = \frac{1}{2} \mathbf{v}_i^T \mathbf{v}_i + \frac{1}{2} \sum_{j=i}^{i+m} (\mathbf{H}\mathbf{V}\mathbf{v}_i - \mathbf{d}_j)^T \mathbf{R}_j^{-1} (\mathbf{H}\mathbf{V}\mathbf{v}_i - \mathbf{d}_j) \quad (2.61)$$

definito nello spazio di controllo.

dim:

Posto

$$\delta \mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{M_i} \quad (2.62)$$

e assunta la (2.56), si ha che il funzionale $J(\mathbf{x}_i)$ assume la seguente forma:

$$\begin{aligned} J(\mathbf{x}_i) &= \frac{1}{2} (\mathbf{x}_i - \mathbf{x}_{M_i})^T \mathbf{B}^{-1} (\mathbf{x}_i - \mathbf{x}_{M_i}) + \frac{1}{2} \sum_{j=i}^{i+m} (H[\mathbf{x}_i] - \mathbf{y}_j)^T \mathbf{R}_j^{-1} (H[\mathbf{x}_i] - \mathbf{y}_j) = \\ &= \frac{1}{2} \delta \mathbf{x}_i^T \mathbf{B}^{-1} \delta \mathbf{x}_i + \frac{1}{2} \sum_{j=i}^{i+m} (H[\mathbf{x}_{M_i} + \delta \mathbf{x}_i] - \mathbf{y}_j)^T \mathbf{R}_j^{-1} (H[\mathbf{x}_{M_i} + \delta \mathbf{x}_i] - \mathbf{y}_j) \\ &\cong \frac{1}{2} \delta \mathbf{x}_i^T \mathbf{B}^{-1} \delta \mathbf{x}_i + \frac{1}{2} \sum_{j=i}^{i+m} (H[\mathbf{x}_{M_i}] + \mathbf{H} \delta \mathbf{x}_i - \mathbf{y}_j)^T \mathbf{R}_j^{-1} (H[\mathbf{x}_{M_i}] + \mathbf{H} \delta \mathbf{x}_i - \mathbf{y}_j) \\ &\cong \frac{1}{2} \delta \mathbf{x}_i^T \mathbf{B}^{-1} \delta \mathbf{x}_i + \frac{1}{2} \sum_{j=i}^{i+m} (\mathbf{H} \delta \mathbf{x}_i - \mathbf{d}_j)^T \mathbf{R}_j^{-1} (\mathbf{H} \delta \mathbf{x}_i - \mathbf{d}_j) \end{aligned} \quad (2.63)$$

dove $\mathbf{d}_j = [\mathbf{y}_j - H[\mathbf{x}_{M_j}]]$.

Effettuando la fattorizzazione di Cholesky della matrice \mathbf{B} : $\mathbf{B} = \mathbf{V}\mathbf{V}^T$ e posto $\mathbf{v}_i = \mathbf{V}^+\delta\mathbf{x}_i$, con \mathbf{V}^+ inversa generalizzata di \mathbf{V} , la funzione può essere riscritta come:

$$J(\mathbf{v}_i) = \frac{1}{2}\mathbf{v}_i^T\mathbf{v}_i + \frac{1}{2}\sum_{j=i}^{i+m}(\mathbf{H}\mathbf{V}\mathbf{v}_i - \mathbf{d}_j)^T R_j^{-1}(\mathbf{H}\mathbf{V}\mathbf{v}_i - \mathbf{d}_j) \quad (2.64)$$



Capitolo 3

Il Modello OceanVar

In questo lavoro di tesi, particolare attenzione é rivolta all'utilizzo della data assimilation nell'ambito dei problemi oceanografici.



Figura 3.1: Mar Mediterraneo

Di seguito descriviamo il modello implementato dal software Oceanvar sviluppato e utilizzato dal CMCC (Centro Euro-Mediterraneo per i Cambiamenti Climatici) per l'assimilazione di dati nello studio dell'evoluzione del Mar Mediterraneo e messo a disposizione del Prof. A.Murli¹ che é

¹Università degli Studi di Napoli Federico II e consorzio SPACI, Napoli, Italia.

responsabile scientifico della mia tesi assieme alla Prof.ssa L.D'Amore².

Lo schema su cui si basa il Software OceanVar é di tipo 3DVar, ovvero calcola il minimo della funzione

$$J(x) = \frac{1}{2}\delta\mathbf{x}^T \mathbf{B}^{-1}\delta\mathbf{x} + \frac{1}{2}(\mathbf{H}\delta\mathbf{x} - \mathbf{d})^T \mathbf{R}^{-1}(\mathbf{H}\delta\mathbf{x} - \mathbf{d}) \quad (3.1)$$

dove il vettore $\mathbf{x} = [\mathbf{T}, \mathbf{S}, \boldsymbol{\eta}, \mathbf{u}, \mathbf{v}]^T$, raccoglie la temperatura \mathbf{T} nello spazio tridimensionale, la salinità \mathbf{S} nello spazio tridimensionale, l'altitudine della superficie libera $\boldsymbol{\eta}$ (dato bidimensionale), ed i vettori \mathbf{u} e \mathbf{v} delle componenti della velocità orizzontale, $\delta\mathbf{x}$ é definita come:

$$\delta\mathbf{x} = \mathbf{x} - \mathbf{x}_M$$

e \mathbf{d} é ancora lo "spostamento":

$$\mathbf{d} = [\mathbf{y} - H(\mathbf{x}_M)].$$

Assunto lo spazio fisico in cui é definita la variabile \mathbf{x} del modello occorre passare allo spazio di controllo in cui é definita la variabile \mathbf{v} e quindi definire il funzionale:

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{v}^T \mathbf{v} + \frac{1}{2}(\mathbf{H}\mathbf{V}\mathbf{v} - \mathbf{d})^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{V}\mathbf{v} - \mathbf{d}) \quad (3.2)$$

La funzione J é minimizzata numericamente usando il metodo quasi-Newton L-BFGS[42]. Le iterazioni del minimizzatore si fermano quando il valore assoluto del gradiente della funzione

$$\nabla J = \mathbf{v} + \mathbf{V}^T \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{V}\mathbf{v} - \mathbf{d}) \quad (3.3)$$

diventa relativamente piccolo rispetto al suo valore iniziale.

²Università degli Studi di Napoli Federico II.

3.1 Operatore lineare \mathbf{V}

A causa della sua grande dimensione, la matrice di trasformazione \mathbf{V} , ad ogni iterazione del minimizzatore, é modellata come una sequenza di operatori lineari. In questo modo, \mathbf{V} trasforma successivamente incrementi nello spazio di controllo verso incrementi nello spazio fisico. Considerando i requisiti specifici per i problemi oceanografici, la matrice \mathbf{V} é definita nel modo seguente:

$$\mathbf{V} = \mathbf{V}_D \mathbf{V}_{uv} \mathbf{V}_\eta \mathbf{V}_H \mathbf{V}_V \quad (3.4)$$

dove l'operatore lineare \mathbf{V}_V trasforma i coefficienti che moltiplicano le EOFs³ verticali nei profili verticali di temperatura e salinitá definiti dal modello dei livelli verticali, l'operatore \mathbf{V}_H applica le variazioni dell'errore orizzontale sui campi della temperatura e della salinitá, l'operatore \mathbf{V}_η calcola la variazione dell'errore dell'altezza della superficie del mare dai campi tridimensionali di temperatura e salinitá, l'operatore \mathbf{V}_{uv} calcola la velocità dall'altezza della superficie del mare, temperatura e salinitá e l'operatore \mathbf{V}_D applica una divergenza, che smorza il filtro sul campo della velocità.

Molto del tempo di calcolo necessario per minimizzare la funzione J viene speso nella trasformazione degli incrementi dallo spazio di controllo allo spazio fisico da parte dell'operatore lineare \mathbf{V} e la sua trasposta. Pertanto, il costo computazionale dello schema 3D-Var é approssimativamente proporzionale alla dimensione del vettore di controllo \mathbf{v} e dipende dalla complessitá degli operatori lineari utilizzati in (3.4). L'utilizzo delle EOFs verticali riduce sia il numero di operazioni sul vettore di stato all'interno del minimizzatore che la dimensione stessa del vettore di controllo \mathbf{v} e, pertanto, puó ridurre il numero di iterazioni del minimizzatore.

³le EOFs (Empirical Orthogonal Functions) consistono in un'analisi degli autovalori della matrice dell'errore. Il loro numero coincide con il numero di autovalori numericamente significativi.

Inoltre, si é applicato un approccio *multigrid* [10] (Figura 3.2). Cosí come il modello é definito sui punti della griglia, allo stesso modo l'approccio multigrid calcola la prima elaborazione su una griglia con una bassa risoluzione spaziale, poi interpola questa soluzione a una griglia piú fine per un'ulteriore minimizzazione, se necessaria. Questo é implementato nello schema 3D-Var mediante un ciclo esterno sulle diverse griglie, ed un ciclo interno che minimizza la funzione di costo per ogni griglia.

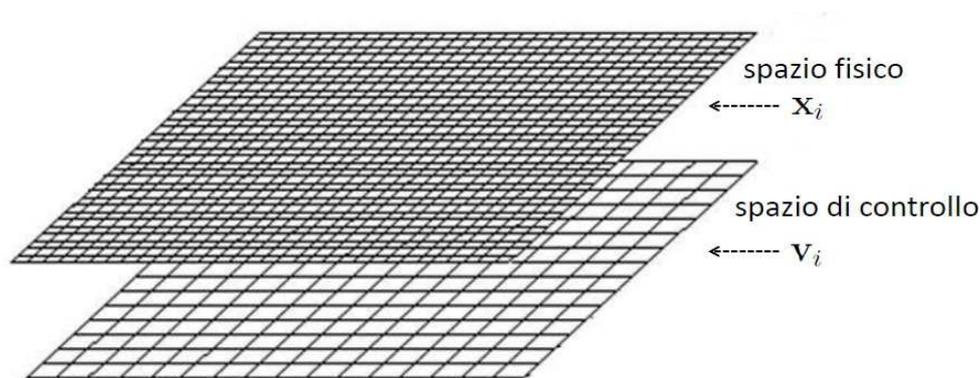


Figura 3.2: spazio fisico e spazio di controllo

Un altro modo per ridurre il carico computazionale é quello di applicare una versione semplificata di operatori lineari in (3.4) durante la prima iterazione del ciclo esterno e quindi utilizzare tale soluzione in luogo della soluzione numerica \mathbf{x}_M per la minimizzazione della successiva iterazione con tutti gli operatori lineari.

3.1.1 Variazione degli errori verticali: V_V

Le variazioni degli errori verticali sono calcolate mediante le EOFs multivariate applicate all'altezza della superficie, ai profili verticali di temperatura e salinitá. Pertanto, in ogni valutazione della funzione $J(x)$

il vettore di controllo \mathbf{v} , contenente i dati relativi alla temperatura, alla salinitá, alla velocitá ed all'altezza della superficie, viene opportunamente modificato mediante prodotti matriciali. Infatti, l'operatore di trasformazione verticale \mathbf{V}_V ha la forma:

$$\mathbf{V}_V = \mathbf{S}\mathbf{\Lambda}^{\frac{1}{2}} \quad (3.5)$$

dove le colonne di \mathbf{S} contengono gli autovettori multivariati e $\mathbf{\Lambda}$ é una matrice diagonale con gli autovalori delle EOFs, assegnati come dati iniziali.

3.1.2 Variazione degli errori orizzontali. \mathbf{V}_H

Si suppone che le variazioni degli errori orizzontali siano funzioni gaussiane con il raggio di correlazione costante. Nei modelli oceanografici, le correlazioni spaziali isotropiche e gaussiane possono essere modellate in modo relativamente efficiente utilizzando un filtro ricorsivo. Esso richiede in genere solo alcune iterazioni, al fine di approssimare la funzione Gaussiana. Inoltre, la sua applicazione su una griglia orizzontale può essere suddivisa in due direzioni indipendenti. Tuttavia, l'applicazione del filtro ricorsivo deve necessariamente tener conto della presenza delle coste.

Le variazioni degli errori orizzontali possono essere calcolate dal filtro ricorsivo come un prodotto dei seguenti operatori:

$$\mathbf{V}_H = \mathbf{W}_y \mathbf{G}_y \mathbf{W}_x \mathbf{G}_x \quad (3.6)$$

dove \mathbf{G}_x e \mathbf{G}_y rappresentano gli operatori del filtro ricorsivo nelle direzioni x e y , mentre \mathbf{W}_x e \mathbf{W}_y sono le matrici diagonali contenenti i coefficienti di normalizzazione che tengono conto della risoluzione variabile della griglia nelle direzioni x e y .

Siccome si utilizzano griglie con coordinate sferiche, la distanza tra i punti in senso longitudinale é una funzione della latitudine. Per la

risoluzione orizzontale variabile é necessario calcolare i coefficienti del filtro ricorsivo per ogni punto del modello sulla griglia orizzontale (costruzione di \mathbf{G}_x e \mathbf{G}_y) ed é necessario quindi ridimensionare il campo filtrato mediante un fattore di normalizzazione spazialmente variabile (costruzione di \mathbf{W}_x e \mathbf{W}_y). L'operatore di normalizzazione \mathbf{W} puó essere applicato indipendentemente nelle due direzioni x e y , subito dopo l'applicazione del corrispondente operatore \mathbf{G} .

Nella Figura 3.3 é riportato un esempio del risultato avuto con e senza l'applicazione della normalizzazione al filtro ricorsivo considerando la risoluzione variabile orizzontale della griglia del modello.

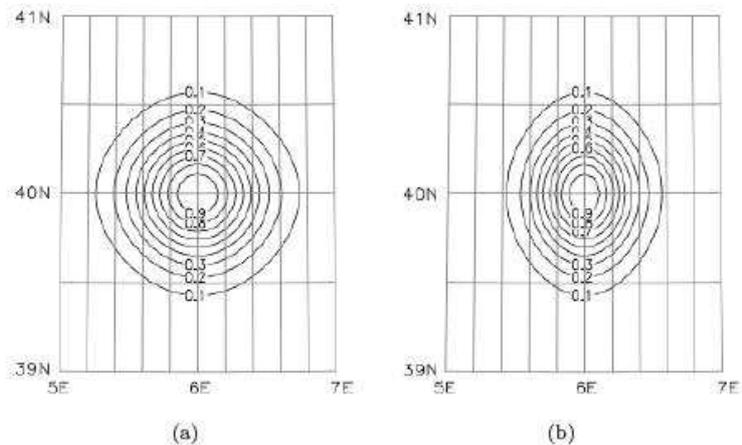


Figura 3.3: Esempio di applicazione (a) e non applicazione (b) della normalizzazione. Le variazioni degli errori corrispondono alla perturbazione di un'unit  dell'altezza del livello del mare centrata a (6 E, 40 N). Le linee sono tracciate con un intervallo di 0,1 unit . Il raggio orizzontale di correlazione é fissato a 30 km, e ci sono otto iterazioni dovute al filtro (quattro per l'operatore \mathbf{V}_H e quattro per la sua trasposta).

Nello schema 3D-Var il fattore di normalizzazione, inizialmente, non viene calcolato sulla griglia di partenza ma utilizzando una tabella di

informazioni ("look-up table"). Tale tabella contiene un numero relativamente piccolo di distanze, comprese tra il valore minimo e il valore massimo fissati sulla griglia di partenza. Sui punti indicati dalla tabella vengono effettuati i calcoli e, solo successivamente, il fattore di normalizzazione viene calcolato per ogni punto della griglia, interpolando, linearmente o quadraticamente, i valori discreti della tabella.

La tabella non può stimare i fattori di normalizzazione vicino alle coste, perché questi rappresentano discontinuità e covarianze orizzontali non omogenee. Per tener conto dei confini costieri in modo computazionalmente efficiente, in primo luogo si applica una condizione precisa di confine all'interno degli operatori \mathbf{G}_x e \mathbf{G}_y . In ogni applicazione monodimensionale del filtro la griglia si estende sulla terra con l'aggiunta di alcuni punti immaginari di confine in prossimità delle coste, come illustrato in Figura 2.23.4.

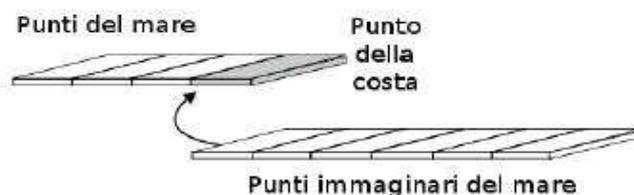


Figura 3.4: Inserimento dei punti immaginari nelle vicinanze delle coste.

Tali punti permettono agli operatori \mathbf{G} di rilevare la presenza della costa e vengono rimossi subito dopo l'applicazione del filtro stesso. In questo modo \mathbf{V}_H non fornisce una soluzione simmetrica, indipendente dall'ordine di applicazione dei filtri monodimensionali. Pertanto, è necessario ridefinire la (3.6) in termini di un operatore simmetrico. Nello schema 3D-Var si utilizza la seguente forma simmetrica:

$$\mathbf{V}_H = \frac{1}{2}(\mathbf{W}_y \mathbf{G}_y \mathbf{W}_x \mathbf{G}_x + \mathbf{W}_x \mathbf{G}_x \mathbf{W}_y \mathbf{G}_y) \quad (3.7)$$

Questa soluzione raddoppia il numero di calcoli in ogni applicazione del filtro orizzontale, ma ci permette ancora di applicare il filtro ricorsivo separatamente in ogni direzione.

Nella prima direzione fissata il filtro ricorsivo é definito dalla seguente formula:

$$A'_i = \alpha A'_{i-1} + (1 - \alpha)A_i, \quad i = 1, \dots, n \quad (3.8)$$

dove A_i rappresenta il valore iniziale del punto i della griglia, A'_i il nuovo valore calcolato, e α il coefficiente di filtro con $0 < \alpha < 1$. Nella seconda direzione fissata, invece, si ha la seguente formula:

$$A''_i = \alpha A''_{i-1} + (1 - \alpha)A'_i, \quad i = 1, \dots, n \quad (3.9)$$

dove A''_i rappresenta il valore ottenuto dopo l'applicazione del secondo filtro.

Il coefficiente di filtro α é dato dalla seguente espressione:

$$\alpha = 1 + E - \sqrt{E(E + 2)} \quad (3.10)$$

ottenuto dall'equazione

$$\frac{\alpha}{(1 - \alpha)^2} = \frac{4R^2}{2N\delta^2} \quad \text{con } E = \frac{N\delta^2}{4R^2} \quad (3.11)$$

dove N rappresenta il numero delle iterazioni del filtro, δ rappresenta la risoluzione della griglia e R il raggio di correlazione.

3.1.3 Modello barotropico: V_η

L'equilibrio dinamico tra le perturbazioni a un livello fissato della superficie del mare e le corrispondenti perturbazioni della temperatura e della salinitá possono essere stimati statisticamente, o applicando una sorta di vincolo geostrofico ⁴. E' stato dimostrato che le EOFs verticali, calcolate esclusivamente tramite le covarianze tra la temperatura, la salinitá

⁴Il bilancio geostrofico richiede che la forza di Coriolis bilanci completamente i gradienti di pressione orizzontali.

e il livello del mare, possono produrre rettifiche sul livello del mare che non sono geostroficamente equilibrate con le rettifiche sulla densit . Se si assimilano solo le osservazioni in situ della temperatura e della salinit , tutte le correzioni non bilanciate saranno rapidamente rimosse dalla regolazione barotropica.

Nel sistema di assimilazione 3D-Var si utilizza l'operatore V_η che fornisce lo stato stazionario di un modello barotropico forzato alle perturbazioni di densit  indotte con l'inserimento delle perturbazioni della temperatura e della salinit . La discretizzazione temporale secondo uno schema semi-implicito del modello barotropico   la seguente:

$$\frac{U^{n+1} - U^{n-1}}{\Delta t} - fV^n = -gH \frac{\partial \eta^*}{\partial x} - \int_{-H}^0 \left[\int_{-z}^0 \frac{\partial(\delta b)}{\partial x} dz' \right] dz + \gamma \nabla^2 U^{n-1} \quad (3.12)$$

$$\frac{V^{n+1} - V^{n-1}}{\Delta t} - fU^n = -gH \frac{\partial \eta^*}{\partial y} - \int_{-H}^0 \left[\int_{-z}^0 \frac{\partial(\delta b)}{\partial y} dz' \right] dz + \gamma \nabla^2 V^{n-1} \quad (3.13)$$

$$\frac{\eta^{n+1} - \eta^{n-1}}{\Delta t} + \left(\frac{\partial U^*}{\partial x} \frac{\partial V^*}{\partial y} \right) = 0 \quad (3.14)$$

dove U e V sono le componenti della velocit  integrate verticalmente, f   il parametro di Coriolis, g l'accelerazione dovuta alla gravit , H la profondit , η l'altezza della superficie, $\partial b = g \frac{\partial \rho}{\rho_0}$ l'anomalia di buoyancy e c   il coefficiente della viscosit  orizzontale.

3.1.4 Operatore della velocità: V_{uv}

L'operatore V_{uv} calcola le correzioni della velocità dividendola nelle componenti barotropiche e barocline ⁵. Le correzioni della velocità barotropica sono ottenute risolvendo le equazioni (3.12-3.14), mentre le parti barocline delle componenti della velocità u_{bc} e v_{bc} sono stimate mediante le formule della dinamica, assumendo il bilancio geostrofico:

$$f_{v_{bc}} = \int_{-z}^0 \frac{\partial(\delta b)}{\partial x} - \frac{1}{H} \int_{-H}^0 \left[\int_{-z}^0 \frac{\partial(\delta b)}{\partial x} dz' \right] dz \quad (3.15)$$

$$f_{u_{bc}} = - \int_{-z}^0 \frac{\partial(\delta b)}{\partial y} + \frac{1}{H} \int_{-H}^0 \left[\int_{-z}^0 \frac{\partial(\delta b)}{\partial y} dz' \right] dz \quad (3.16)$$

3.1.5 Operatore della divergenza: V_D

In assenza di coste l'operatore della velocità V_{uv} impone un equilibrio dinamico tra le correzioni della velocità, l'altezza della superficie del mare e i campi di massa. Anche la velocità baroclina deve tener conto della presenza della costa, ma se il passaggio dal suo valore a zero é troppo brusco, la componente della divergenza del campo della velocità può aumentare irrealisticamente. Per tale motivo si utilizza l'operatore della divergenza V_D . Esso, infatti, serve a ottenere una stima piú precisa delle perturbazioni della velocità in prossimitá delle coste, mantenendo inalterata la vorticitá.

In Figura 3.5 é riportato un esempio del risultato avuto con e senza l'applicazione della divergenza al campo della velocità.

⁵Le variazioni verticali del flusso possono essere decomposte in una componente barotropica, che é indipendente dalla profonditá, e una componente baroclina che varia con la profonditá.

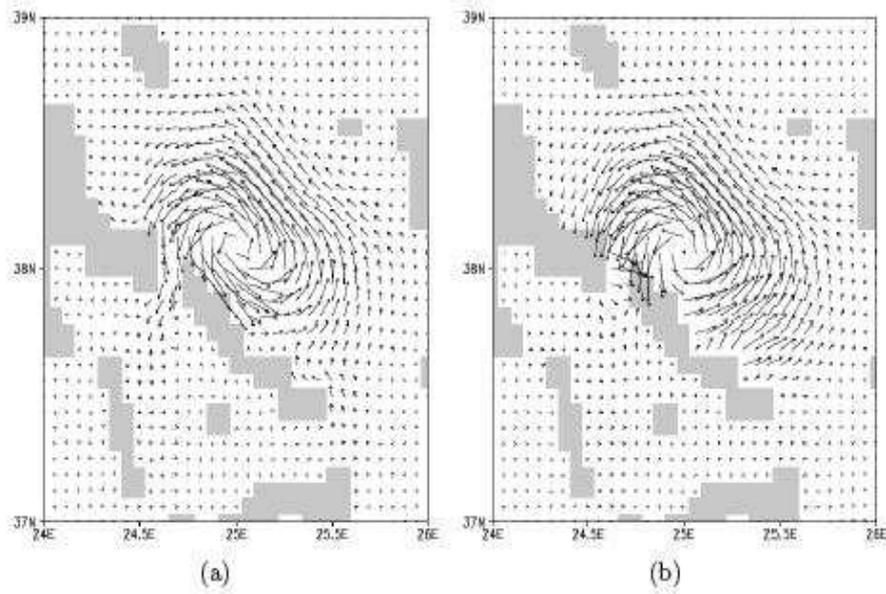


Figura 3.5: Esempio delle correzioni delle velocità ottenute dall'applicazione (a) e dalla non applicazione (b) della divergenza. Le osservazioni sono relative ai profili di temperatura e salinità ottenuti dalle navi Argo. I punti della griglia corrispondenti alla terra sono colorati di grigio.

La divergenza di smorzamento si ottiene mediante l'applicazione successiva di un filtro nella forma:

$$\partial v^{n+1} = \partial v^n + k \nabla \mathbf{D}^n \quad (3.17)$$

dove $\partial \mathbf{v} = (\partial u, \partial v)^T$ è l'incremento della velocità totale ad ogni livello verticale, gli indici indicano il passo di iterazione, k è il coefficiente del filtro, e \mathbf{D} è la divergenza orizzontale degli incrementi della velocità.

Capitolo 4

Il Software OceanVar

Per comprendere a fondo le problematiche che occorre affrontare nell'implementazione del problema data assimilation, é stato necessario installare e studiare alcuni tra i software piú utilizzati. In particolare é stato necessario analizzare e studiare nel dettaglio il software **Oceanvar**, sviluppato dall'INGV (Istituto di Geofisica e Vuolcanologia) di Bologna e utilizzato dal CMCC (Centro Euro-Mediterraneo per i Calmbiamenti Climatici) al fine di un contributo dal punto di vista della parallelizzazione del codice e i Software open-source **3DVAR**, **OpenDA** e **DART** descritti in Appendice A.

Ocenvar (Ocean Variational Data Assimilation) é un software che implementa un modello di tipo 3DVar descritto con dettaglio nel Capitolo3.

Il software é costituito da una libreria suddivisa in due moduli principali, `calc_msft` e `oceanvar`.

Il modulo `calc_msft` é interamente implementata in linguaggio Fortran 90. Per `oceanvar`, invece, il linguaggio utilizzato é il Fortran 90 ad eccezione della routine `setulb` e delle routine da esse sotto richiamate, implementate nel linguaggio Fortran 77.

Il modulo `Calc_msft` legge da file i valori relativi ai dati osservati SLA, ARGO, XBT e quelle effettuate mediante i gliders, memorizzandoli poi in file dati; calcola quindi i residui tra il valore osservato e il valore del background per ogni tipo di osservazione, salvando tali valori in file dati.

La struttura di `Calc_msft` é rappresentata dallo schema ad albero illustrato in Figura 4.1, dove la routine preceduta dalla freccia viene richiamata due volte consecutivamente.

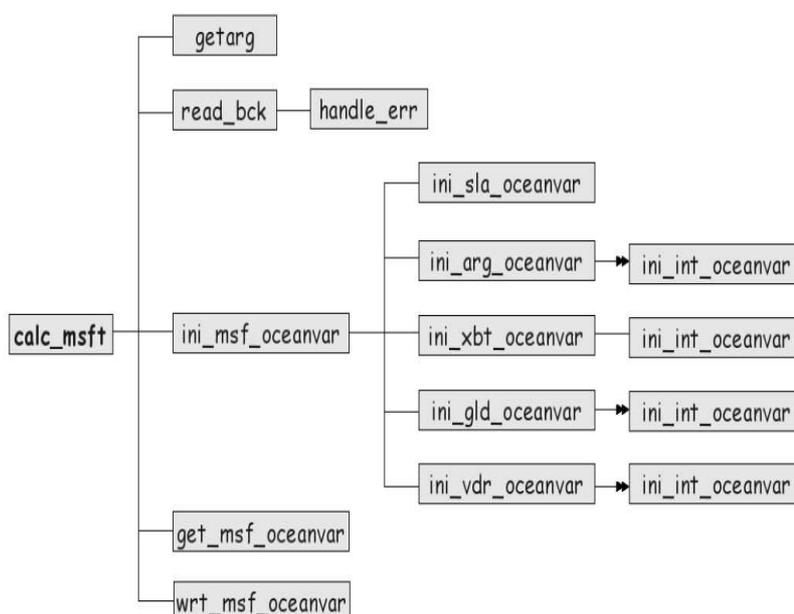


Figura 4.1: Schema ad albero della routine `calc_msft`

Il modulo `Oceanvar`, invece, dopo aver letto tutti i dati in input relativi alle osservazioni effettuate e, dopo aver creato la griglia su cui lavorare, minimizza la funzione:

$$J(\mathbf{x}) = \frac{1}{2} \mathbf{v}^T \mathbf{v} + \frac{1}{2} (\mathbf{H} \mathbf{V} \mathbf{v} - \mathbf{d})^T \mathbf{R}^{-1} (\mathbf{H} \mathbf{V} \mathbf{v} - \mathbf{d})$$

che misura la differenza tra lo stato del sistema, stimato dal modello, e lo stato osservato.

La struttura del modulo Oceanvar può essere rappresentata dallo schema ad albero riportato in Figura 4.2, da interpretare con la seguente legenda:

(a) IF (`ktr .le. drv%ktr`)

Se la variabile `ktr`, contatore del ciclo DO, vale meno o uguale del massimo numero di iterazioni `drv%ktr`

(b) IF (`ktr.eq.1 .or. drv%ratio(ktr).ne.1.0`)

Se la variabile `ktr` vale 1, cioè ci si trova alla prima iterazione oppure se la variabile del rapporto tra le griglie successive `drv%ratio` al passo `ktr` è diversa da 1

(c) IF (`ktr .eq. 1`)

Se la variabile `ktr` vale 1, cioè ci si trova alla prima iterazione

(d) IF (`drv%bmd(drv%ktr) .eq. 1`)

Se la variabile del flag del modello barotropico `drv%bmd` al passo `drv%ktr` vale 1

(e) IF (`ktr.gt.1 .and. drv%ratio(ktr).ne.1.0`)

Se la variabile `ktr` è maggiore di 1, cioè se non ci si trova alla prima iterazione, e se la variabile del rapporto tra le griglie successive `drv%ratio` al passo `ktr` è diversa da 1

(f) IF (`ktr .eq. drv%ntr`)

Se la variabile `ktr` vale `drv%ntr`, cioè se ci si trova all'ultima iterazione `drv%ntr`

(g) IF (`ktr .ne. drv%ntr`) then

```
IF ( drv%ratio(ktr+1) .ne. 1.0 )
```

Se la variabile `ktr` non vale `drv%ntr`, cioè se non ci si trova all'ultima iterazione, e la variabile del rapporto tra le griglie successive `drv%ratio` al passo `ktr+1` è diversa da 1

```
(h) IF ( stat /= nf90 noerr )
```

Se la variabile `stat` è diversa da `nf90 noerr`, cioè se l'uscita di una chiamata a funzione della libreria `NetCDF` non ha dato errore

```
(i) IF ( k .le. rcf%ntb )
```

Se la variabile `k`, contatore del ciclo `DO`, vale meno o uguale della metà del numero delle iterazioni `rcf%ntr`

```
(l) IF ( drv%ktr .eq. 1 )
```

Se la variabile `drv%ktr` vale 1, cioè ci si trova alla prima iterazione

```
(m) WHILE ( ( ctl%task(1:5).eq.'NEW X') .or. ( ctl%task(1:2).eq.'FG'  
) .or.  
( Ctl%task(1:5).eq.'START'))
```

Finché i primi 5 caratteri della variabile `ctl%task` sono `'NEW X'` oppure finché i primi 2 caratteri della variabile `ctl%task` sono `'FG'` oppure finché i primi 5 caratteri della variabile `ctl%task` sono `'START'`

```
(n) IF ( ctl%task(1:2) .eq. 'FG' )
```

Se i primi 2 caratteri della variabile `ctl%task` sono `'FG'`

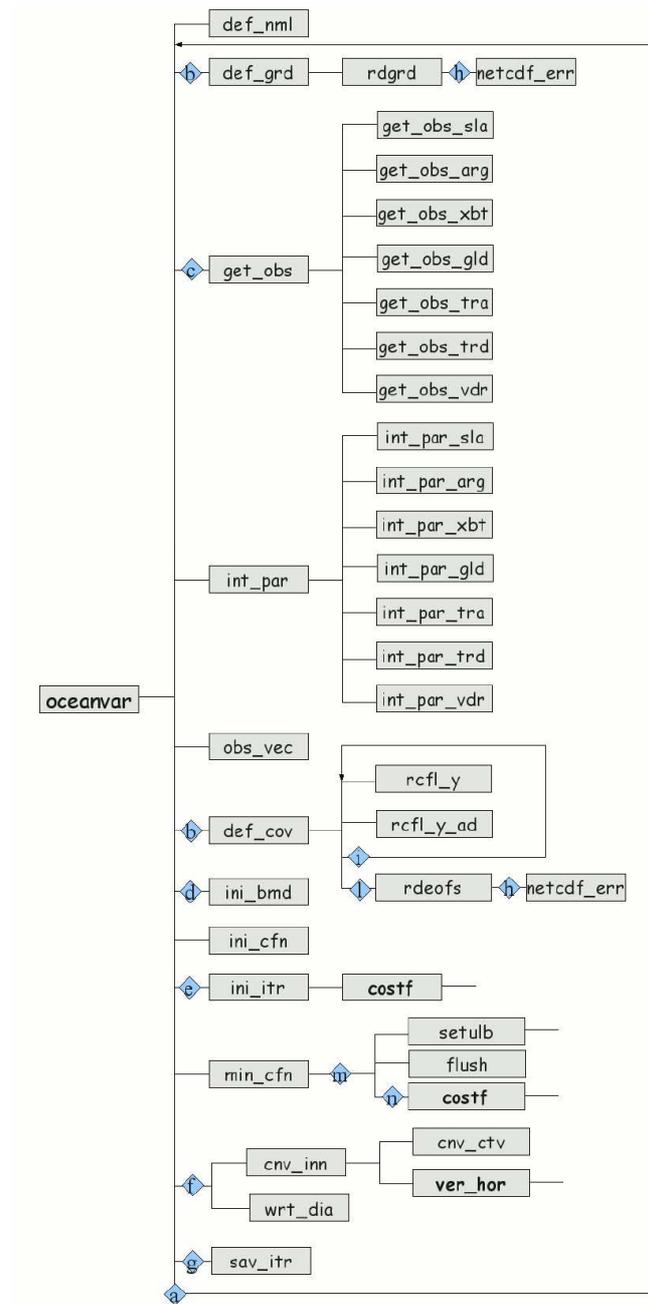


Figura 4.2: schema ad albero della routine **oceanvar**

Per la routine `setulb` si ha lo schema riportato in Figura 4.3

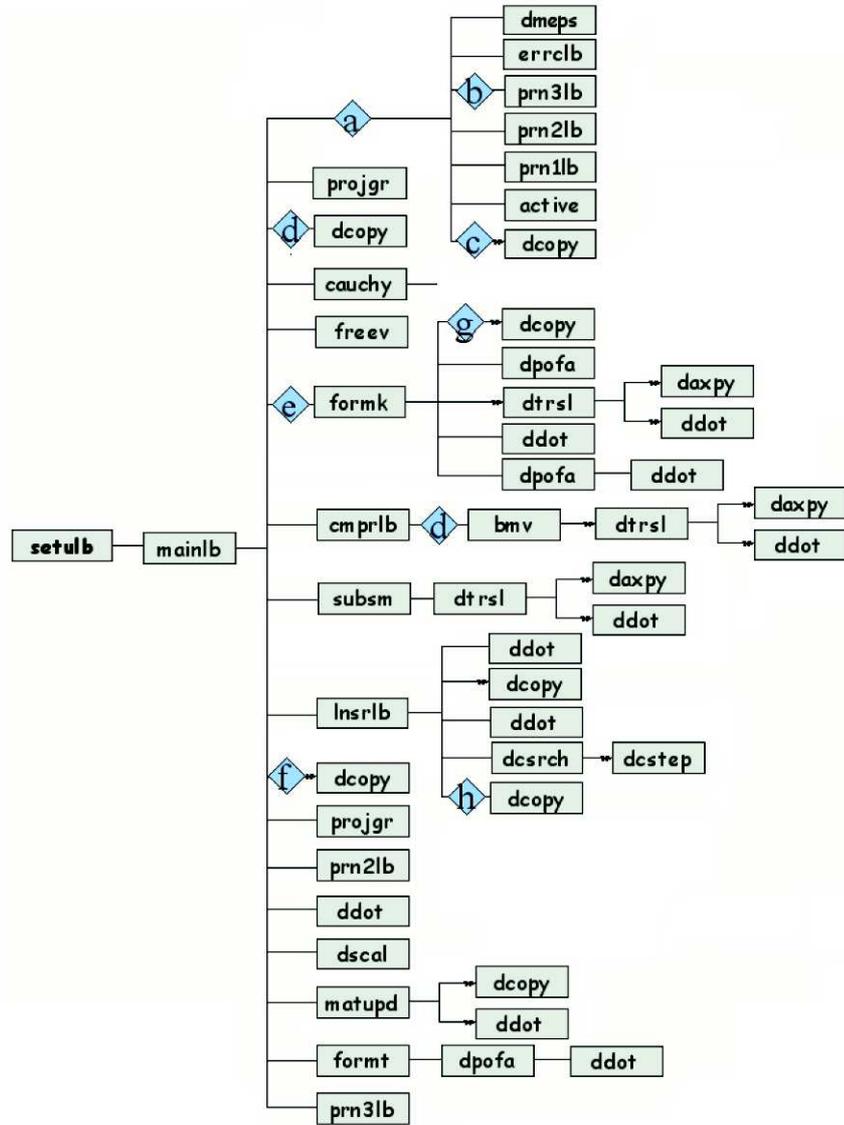


Figura 4.3: schema ad albero della routine `setulb`

Tale schema va interpretato con la seguente legenda:

(→) le routine precedute dalla freccia sono richiamate piú volte consecutivamente

(a) IF (task .eq. 'START')

Se la variabile `task` ha il valore 'START', cioè se é la prima volta che si utilizza `mainlb`

(b) IF (task(1:5) .eq. 'ERROR')

Se i primi 5 caratteri della variabile `task` hanno il valore 'ERROR', cioè se c'è qualche errore negli argomenti in input

(c) IF (task(1:4) .eq. 'STOP') then

IF (task(7:9) .eq. 'CPU')

Se i primi 4 caratteri della variabile `task` hanno il valore 'STOP' e se i caratteri dal settimo al nono hanno il valore 'CPU'

(d) IF (.not. cnstnd .and. col .gt. 0)

Se la variabile `cnstnd` é falsa, cioè il problema non é limitato, e se la variabile `col` é maggiore di zero

(e) IF (wrk)

Se la variabile `wrk` é vera

(f) IF (info .ne. 0 .or. iback .ge. 20)

Se la variabile `info` non é uguale a zero, cioè se non ci sono stati errori, o se la variabile `iback` é maggiore o uguale a 20

(g) IF (updatd) then

IF (iupdat .gt. m)

Se la variabile `updatd` é vera, cioè se la matrice L-BFGS é stata aggiornata, e se il valore della variabile `iupdat` é maggiore del valore della

variabile `m`, cioè se il numero totale degli aggiornamenti effettuati fino a quel momento é maggiore del numero massimo di correzioni

```
(h) if ( (csave(1:4) .ne. 'CONV') .and.
(csave(1:4) .ne. 'WARN') ) then
if (stp .eq. one)
```

Se le prime 4 cifre della variabile `csave` non hanno il valore 'CONV' né il valore 'WARN' e se la variabile `stp` é uguale a 1

La routine Cauchy a sua volta richiama altre funzioni mediante lo schema riportato in Figura 4.4

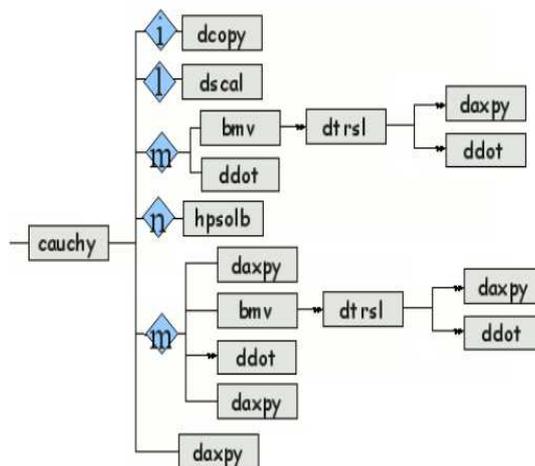


Figura 4.4: schema ad albero della routine `cauchy`

(→) le routine precedute dalla freccia sono richiamate piú volte consecutivamente

```
(i) IF ( sbgnrm .le. zero )
```

Se il valore della variabile `sbgnrm` é minore o uguale del valore della vari-

abile zero, cioè 0.0d0

(1) IF (theta .ne. one)

Se il valore della variabile `theta` non é uguale al valore della variabile `one`, cioè 1.0d0

(m) IF (col .gt. 0) Se il valore della variabile `col` é maggiore di zero

(n) IF (iter .ne. 1) Se il valore della variabile `iter` non é uguale a uno

4.1 Analisi dei tempi di esecuzione del codice

L'architettura utilizzata nella fase di testing e di ottimizzazione é Asama2, cluster vettoriale della NEC allocato nei laboratori del CMCC di Capua¹. Come mostrato in Figura 3.5, tale cluster vettoriale é costituito da 4 nodi di tipo SX-8R disomogenei: tre di essi montano 8 CPU, mentre uno ne monta 6, per un totale di 128 Gb di RAM ciascuno; tutti sono muniti di processori Itanium. La lettera R che compare nel nome sta per reinforced. Questa qualità permette di avere maggiori potenzialità e affidabilità della macchina nell'esecuzione delle diverse applicazioni. Esse vanno ad aggiungersi ai livelli di performance e affidabilità propri delle macchine vettoriali appartenenti alla famiglia NEC-SX.

Al fine di ridurre i tempi di calcolo, un primo tentativo é stato ottimizzare i moduli di calcolo a basso livello e utilizzare, lí dove possibile, librerie di base del calcolo Scientifico quali BLAS (Basic Linear Algebra Subprograms) e Lapack (Linear Algebra PACKage). A tal fine é stato

¹<http://www.cmcc.it/chi-siamo/sedi/cmcc-capua>

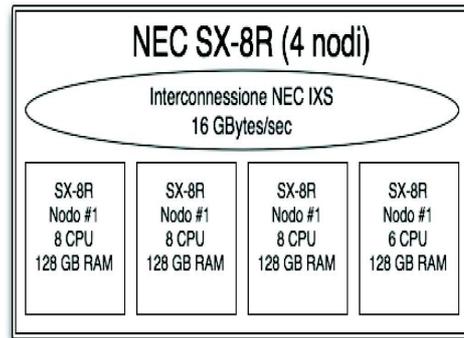


Figura 4.5: Architettura NEC SX-8R

monitorato il tempo dei due moduli principali (*Calc.msft* e *Oceanvar*).

In particolare é stata utilizzata la function *etime*, tramite la quale é possibile conoscere:

- il tempo di CPU impiegato dal processo (*user time*);
- il tempo di CPU impiegato dal sistema (*system time*);
- il tempo totale impiegato dal processo (*elapsed time*).

Inoltre, siccome non ci si puó attenero ai tempi presi in un'unica esecuzione, tutti i tempi riportati sono i valori medi dei tempi ottenuti in piú esecuzioni.

Per il modulo **Calc.msft** si sono ottenuti i seguenti valori:

- 0.51 secondi per l'*elapsed time*;
- 0.35 secondi per lo *user time*;
- 0.16 secondi per il *system time*.

Per quanto riguarda **Oceanvar**, invece, inizialmente sono stati rilevati i tempi dalla routine principale e quelli delle prime routine da essa richiamate, quelle cioè relative al primo ramo dell'albero e si sono ottenuti i seguenti valori:

- 75.30 secondi per l'elapsed time;
- 71.57 secondi per lo user time;
- 3.73 secondi per il system time.

E' utile analizzare nel dettaglio lo user time impiegato dalle prime routine richiamate da Oceanvar. Nonostante i tempi non siano uguali in tutte le prove, é possibile comunque notare differenze negli ordini di grandezza dei tempi delle routine. Nella Tabella 4.1 sono state riportate solo le tre routine computazionalmente piú onerose: def_cov, min_cfn e cnv_inn. Le altre routine, infatti, riportano tempi di ordine minore o uguale di 10^{-1} secondi.

ROUTINE	Media dei tempi
def_cov	6.48 sec.
min_cfn	57.69 sec.
cnv_inn	7.31 sec.

Tabella 4.1: Media degli user time delle routine def_cov, min_cfn e cnv_inn

E' chiaro quindi che la routine che impiega piú tempo nell'esecuzione é min_cfn. Per tale motivo, si é passati ad analizzare nel dettaglio il suo schema ad albero, riportato in Figura 4.6.

In particolare, i simboli (m) e (n) riportati in Figura 4.6 indicano:

```
(m) WHILE ( ( ct1%task(1:5).eq.'NEW X') .or. ( ct1%task(1:2).eq.'FG'
) .or. ( Ct1%task(1:5).eq.'START'))
```

Finché i primi 5 caratteri della variabile ct1%task sono 'NEW X'

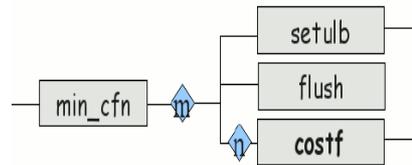


Figura 4.6: Schema ad albero della routine **min_cfn**

oppure finché i primi 2 caratteri della variabile `ctl%task` sono 'FG' oppure finché i primi 5 caratteri della variabile `ctl%task` sono 'START'

(n) IF (`ctl%task(1:2)` .eq. 'FG')

Se i primi 2 caratteri della variabile `ctl%task` sono 'FG'

Le routine `setulb`, `flush` e `costf`, possono essere richiamate anche piú di una volta, finché é soddisfatta la condizione del `while`. In particolare la routine `costf` viene richiamata solo se la condizione IF é soddisfatta. Per tale motivo, i tempi considerati per ogni sottoroutine richiamata sono i tempi totali raccolti nella routine `min_cfn`. Si sono ottenuti i valori riportati in Tabella 4.2.

ROUTINE	Media dei tempi
<code>setulb</code>	1.80 sec.
<code>costf</code>	55.89 sec.

Tabella 4.2: Media degli user time delle routine `setulb` e `costf`

In questo caso, é evidente che la routine che impiega piú tempo nell'esecuzione é **costf**. Si é passati quindi ad analizzare nel dettaglio tale routine, il cui schema ad albero é riportato in Figura 4.7, che va interpretato con la seguente legenda:

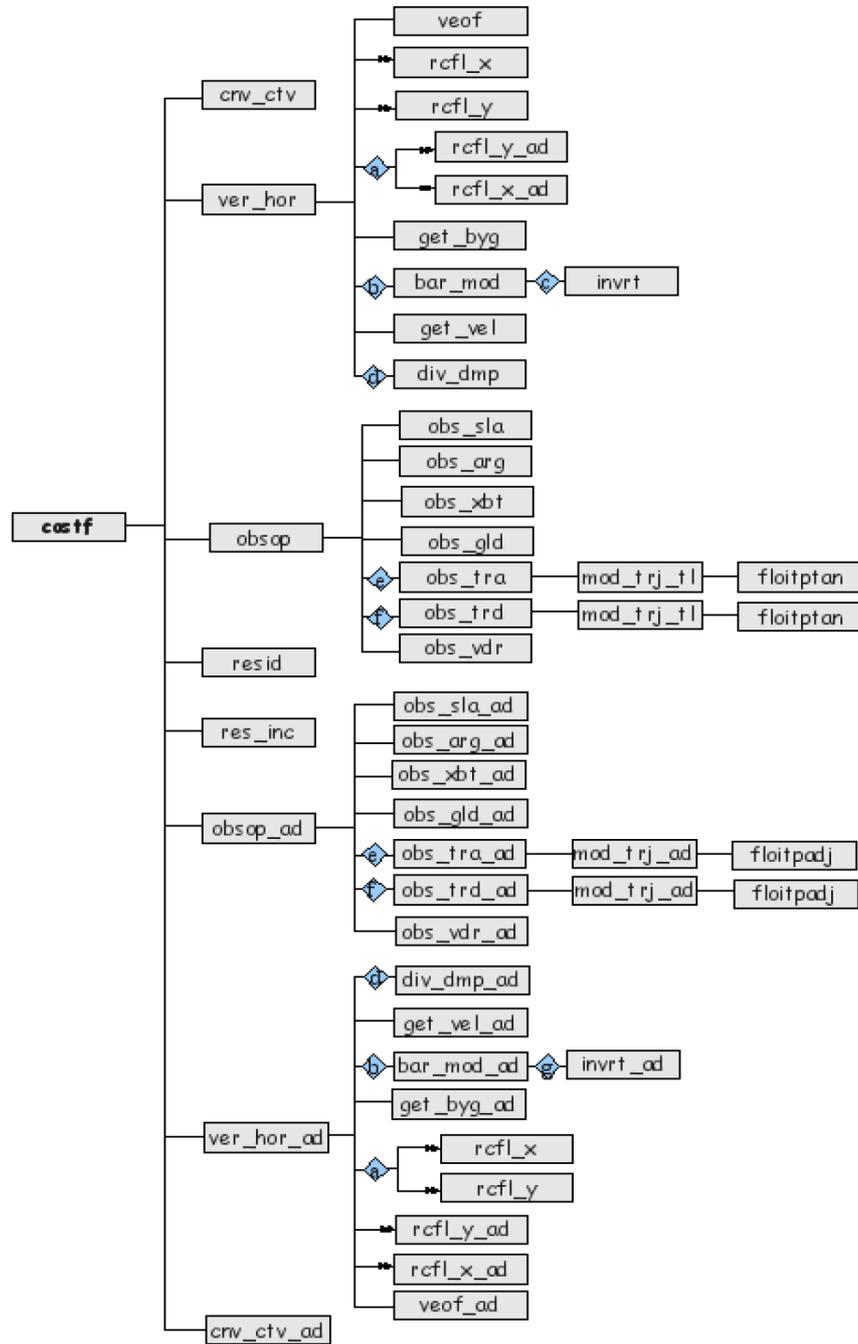


Figura 4.7: Schema ad albero della routine `costf`

- (→) le routine precedute dalla freccia sono richiamate piú volte consecutivamente se: (`drv%bmd(drv%ktr) .ne. 1`)
cioé se la variabile contenente il flag relativo al modello barotropico al passo `drv%ktr` non vale 1
- (a) IF (`drv%mask(drv%ktr) .gt. 1`)
Se la variabile `drv%mask` contenente la maschera usata per le covarianze orizzontali al passo `drv%ktr` é maggiore di 1, cioé se ci si trova nella presenza dei confini di costa
- (b) IF (`drv%bmd(drv%ktr) .eq. 1`)
Se la variabile del flag del modello barotropico `drv%bmd` al passo `drv%ktr` vale 1
- (c) IF (`kstp .le. bmd%nstps`)
Se il contatore `kstp` del ciclo D0 é minore o uguale del numero dei tempi step nel ciclo principale `bmd%nstps`
- (d) IF (`drv%dda(drv%ktr) .eq. 1`)
Se la variabile `drv%dda` contenente il flag per il filtro della divergenza in orizzontale al passo `drv%ktr` vale 1
- (e) IF (`tra%no .gt. 0`)
Se il numero delle osservazioni delle traiettorie di Argo é maggiore di 0
- (f) IF (`trd%no .gt. 0`)
Se il numero delle osservazioni delle traiettorie della superficie dei drifters é maggiore di 0
- (g) IF (`kstp .ge. 1`)
Se il contatore `kstp` del ciclo D0, che inizialmente é uguale al numero dei tempi step nel ciclo principale `bmd%nstps` e ad ogni passo viene decrementato di una unità, é maggiore o uguale a 1

Inizialmente sono stati rilevati i tempi per le prime routine richiamate da `costf`, quelle relative al primo ramo dell'albero. Osserviamo che la routine `costf` può essere richiamata anche più di una volta, finché è soddisfatta la condizione del `while` della funzione `min_cfn`, ma è richiamata solo se la condizione IF della funzione `min_cfn` è soddisfatta. Per tale motivo, i tempi considerati per ogni sotto-routine richiamata da `costf` sono i tempi totali raccolti nella routine `min_cfn`.

Anche in questo caso, le differenze tra i tempi rilevati nelle varie prove non ci impediscono di osservare le differenze negli ordini di grandezza dei tempi delle routine. Nella Tabella 4.3 sono state riportate solo le due routine computazionalmente più onerose: `ver_hor` e `ver_hor_ad`. Le altre routine, infatti, riportano tempi di ordine minore o uguale di 10^{-2} secondi.

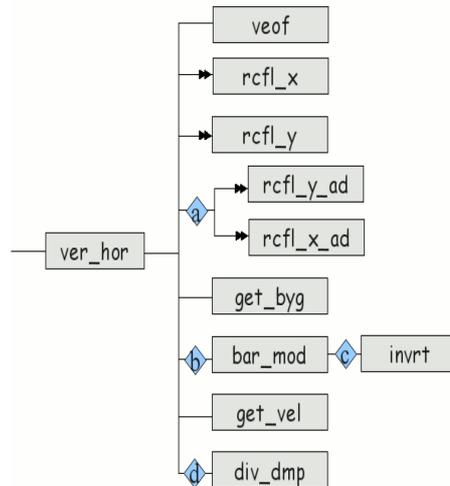
ROUTINE	Media dei tempi
<code>ver_hor</code>	24.68 sec.
<code>ver_hor_ad</code>	31.05 sec.

Tabella 4.3: Media degli user time delle routine `ver_hor` e `ver_hor_ad`

Si è passati quindi ad analizzare nel dettaglio le routine `ver_hor` e `ver_hor_ad`.

Lo schema ad albero della routine `ver_hor` è illustrato nella Figura 5.3.

- (→) Le routine precedute dalla freccia vengono richiamate due volte consecutivamente oppure tre volte consecutivamente se: (`drv%bmd(drv%ktr)`
`.ne. 1`)
 cioè se la variabile contenente il flag relativo al modello barotropico al passo `drv%ktr` non vale 1

Figura 4.8: Schema ad albero della routine **ver_hor**

- (a) IF (`drv%mask(drv%ktr) .gt. 1`)
 Se la variabile `drv%mask` contenente la maschera usata per le covarianze orizzontali al passo `drv%ktr` è maggiore di 1, cioè se ci si trova nella presenza dei confini di costa
- (b) IF (`drv%bmd(drv%ktr) .eq. 1`)
 Se la variabile del flag del modello barotropico `drv%bmd` al passo `drv%ktr` vale 1
- (c) IF (`kstp .le. bmd%nstps`)
 Se il contatore `kstp` del ciclo DO è minore o uguale del numero dei tempi step nel ciclo principale `bmd%nstps`
- (d) IF (`drv%dda(drv%ktr) .eq. 1`)
 Se la variabile `drv%dda` contenente il flag per il filtro della divergenza in orizzontale al passo `drv%ktr` vale 1

Nella routine `ver_hor` viene costruita la matrice di trasformazione V . In Figura 4.9 è riportato il flowchart che indica tutte le routine richia-

mate per la costruzione della matrice \mathbf{V} .

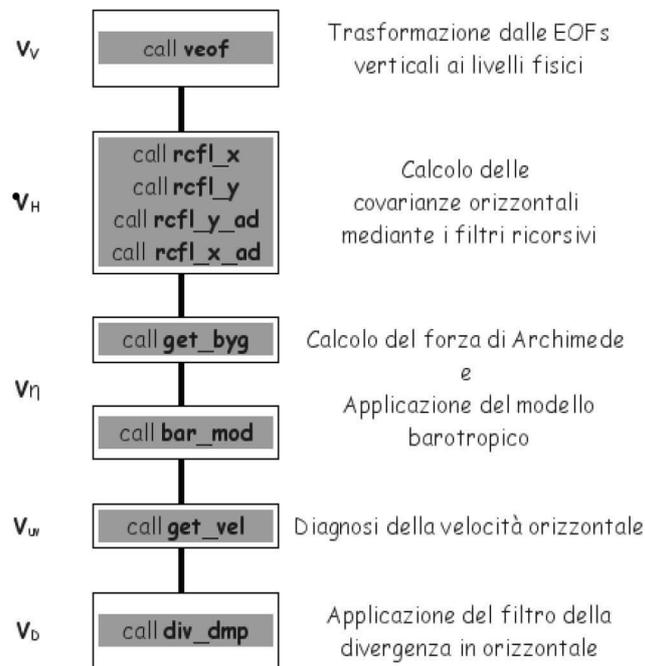


Figura 4.9: Flowchart della routine `ver_hor`

Considerando tale schema si è deciso di analizzare i tempi della routine `ver_hor` non secondo le varie sotto-routine richiamate ma secondo i calcoli effettuati per ogni operatore che compone la matrice \mathbf{V} . Poiché la routine `ver_hor` viene richiamata più volte nello schema del modulo Oceanvar, i tempi considerati per ogni operatore sono i tempi totali raccolti nella routine principale Oceanvar.

I tempi di esecuzione per ogni operatore sono riportati in Tabella 4.4

È importante osservare che la somma di tali tempi supera il tempo di `ver_hor` calcolato in `min_cfn`. Tale risultato è dovuto al fatto che la routine `ver_hor` non è richiamata solo dalla routine `costf` nella routine `min_cfn`, ma nello schema generale è richiamata più volte. I tempi presi

CALCOLI	Media dei tempi
EOFs	6.48 sec.
covarianze orizzontali	10.85 sec.
modello barotropico	3.62 sec.
velocità orizzontale	0.79 sec.
divergenza orizzontale	10.25 sec.

Tabella 4.4: Media degli user time dei calcoli principali effettuati in `ver_hor`

per i vari calcoli effettuati in `ver_hor` sono i tempi totali ottenuti durante tutta l'esecuzione del modulo Oceanvar.

Lo schema ad albero della routine `ver_hor_ad`, invece, è illustrato in Figura 4.10.

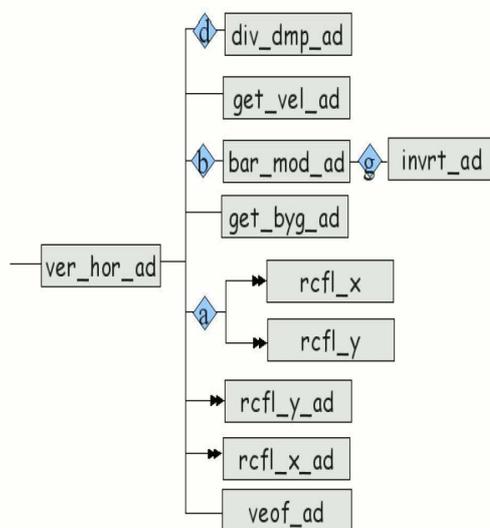


Figura 4.10: Schema ad albero della routine `ver_hor_ad`

(→) Le routine precedute dalla freccia vengono richiamate due volte

consecutivamente oppure tre volte consecutivamente se

(`drv%bmd(drv%ktr) .ne. 1`)

cioé se la variabile contenente il flag relativo al modello barotropico al passo `drv%ktr` non vale 1

(a) IF (`drv%mask(drv%ktr) .gt. 1`)

Se la variabile `drv` covarianze orizzontali al passo `drv%ktr` é maggiore di 1, cioé se ci si trova nella presenza dei confini di costa

(b) IF (`drv%bmd(drv%ktr) .eq. 1`)

Se la variabile del flag del modello barotropico `drv%bmd` al passo `drv%ktr` vale 1

(d) IF (`drv%dda(drv%ktr) .eq. 1`)

Se la variabile `drv%dda` contenente il flag per il filtro della divergenza in orizzontale al passo `drv%ktr` vale 1

(g) IF (`kstp .ge. 1`)

Se il contatore `kstp` del ciclo DO, che inizialmente é uguale al numero dei tempi step nel ciclo principale `bmd%nsteps` e ad ogni passo viene decrementato di una unità, é maggiore o uguale a 1

Anche per la routine `ver_hor_ad`, si é deciso di analizzare i tempi non secondo le varie sotto-routine richiamate ma secondo i calcoli principali effettuati, riportati nel flowchart in Figura 4.11.

Poiché la routine `ver_hor_ad` viene richiamata piú volte nello schema del modulo `Oceanvar`, i tempi considerati per ogni operatore sono i tempi totali raccolti nella routine principale `Oceanvar`.

I valori ottenuti sono riportati nella Tabella 4.5

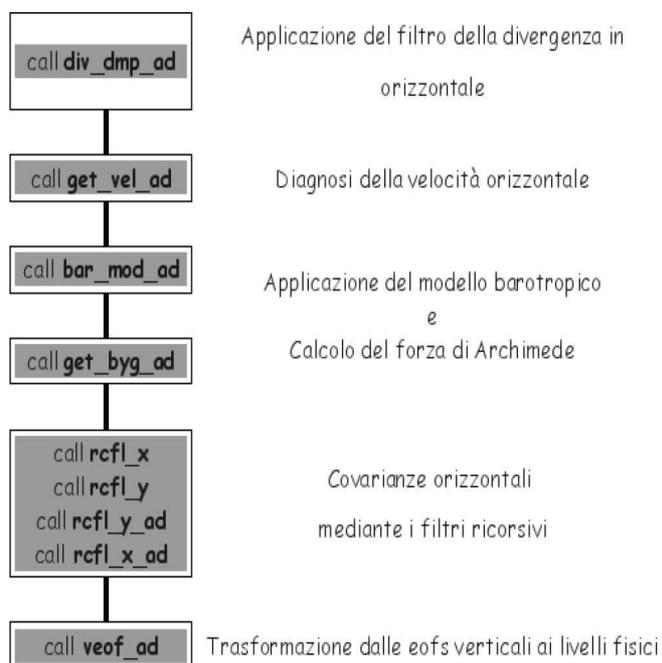


Figura 4.11: Flowchart della routine `ver_hor_ad`

CALCOLI	Media dei tempi
EOFs	4.80 sec.
covarianze orizzontali	9.77 sec.
modello barotropico	5.46 sec.
velocità orizzontale	1.70 sec.
divergenza orizzontale	9.32 sec.

Tabella 4.5: Media degli user time dei calcoli principali effettuati in `ver_hor_ad`

Capitolo 5

Parallelizzazione del Software OcenVar

Oggetto del lavoro svolto é stato rendere il software OceanVarpiú efficiente, tramite l'introduzione di una qualche forma di parallelismo e ottimizzazione. Il Calcolatore utilizzato é della NEC e consiste in 4 nodi SX-8R: tre montano 8 CPUs e uno soltanto monta 6 CPUs.

Il risultato é stato quello di introdurre, nel codice OceanVar, una forma di parallelismo su due livelli: multi-core e multiprocessore. L'introduzione del parallelismo multicore avviene mediante l'utilizzo delle librerie di base Blas e Lapack per l'esecuzione delle operazioni di algebra lineare che nelle loro implementazioni piú moderne, sono in grado di utilizzare efficientemente, grazie al multithreading, le architetture multicore. L'introduzione del parallelismo multiprocessore é avvenuto nelle due routine che hanno il maggiore costo computazionale in cui é stato implementato un parallelismo sulle operazioni in quanto ée stato possibile individuare sequenze di calcoli che vengono eseguiti, in maniera indipendente, sulle seguenti tre grandezze: altitudine della superficie libera η , temperatura T e salinitá S.

5.1 Il codice multicore

Dopo un'accurata analisi del codice del software OceanVAR, si è scelto di eseguire una prima parallelizzazione a livello di operazioni di base. L'idea è di lanciare un solo processo su un intero nodo (costituito da otto core) sfruttando la versione nativa multi-thread della macchina delle librerie BLAS e Lapack, al fine di rendere visibile ad ogni processo la possibilità di utilizzare più core per l'esecuzione di operazioni di base.

In Figura 5.1 è mostrato un esempio di parallelizzazione di primo livello in cui si fruttano 6 core.

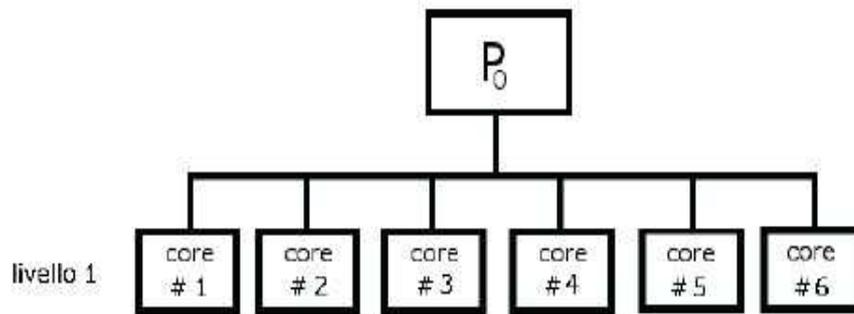


Figura 5.1: Esempio di parallelizzazione di primo livello in cui si fruttano 6 core.

Per attivare l'ambiente multicore di BLAS e Lapack, è stato inizializzato l'ambiente della libreria MPI (Message Passing Interface).

Per il nostro obiettivo, il software è stato eseguito utilizzando un solo processo che può gestire fino ad un tetto massimo di otto core.

Per eseguire il software, é stato modificato lo script per la sottomissione. Quello di cui ci si serve é il seguente:

```
# PBS -S / bin /ksh
# PBS -q s tandard
#PBS -T mpisx
#PBS -l e l ap s t im r e q=1000
#PBS -l cpunum job=8
#PBS -b 1
#PBS -j o
work=/home/sm502/OceanVar Lina /exp/MFS471/work
cd ${work}
expor t OMPNUMTHREADS=8
expor t MPIPROGINF=YES
expor t MPIMULTITASKMIX=YES
# $ Link files with input variables
rm SLA.dat ARGO.dat XBT.dat GLIDER.dat driftervel.dat
ln -fs exercise_input/SLA.dat SLA.dat

.....

rm var_3dp
ln -fs ../../bin/var_3dp var_3dp
mpirun -np 1 -nnp 1 ./var_3dp
```

Tabella 5.1: script utilizzato

In particolare:

- é stata aggiunta la riga `#PBS -T mpisx` che indica il tipo di job, in questo caso si specifica `mpisx` per l'uso dell'MPI su architettura SX;

- é stata sostituita la riga `#PBS -l cpunum job=8` alle righe:
`#PBS -l cpunum job=1`
`#PBS -l cpunum prc=1`
`#PBS -l tasknum prc=1`

Esse indicavano l'utilizzo di un solo processore e la possibilità di utilizzare un solo core, infatti:

- `# PBS -l cpunu_ job` indica il massimo numero di CPU che il job può utilizzare per nodo;
- `#PBS -l cpunum_prc` indica il massimo numero di CPU che ogni processo può utilizzare;
- `#PBS -l tasknum_prc` indica il massimo numero di tasks che il processo può avere.

Ora invece, con l'inserimento del comando

```
export OMP_NUM_THREADS=8
```

si richiede di poter utilizzare al più otto core per le operazioni multithread, in modo da non incorrere in comunicazioni intranodo. Infine mediante la riga

```
mpirun -np 1 -nnp 1 ./var_3dp
```

si richiede di eseguire un processo MPI utilizzando un unico processore.

Per il codice multicore, si sono ottenuti i seguenti valori dei tempi:

- 57.28 secondi per l'elapsed time;

- 7.29 secondi per il tempo totale impiegato dalla routine `wrt_dia` (un'approssimazione del `system time`);
- 49.99 secondi per il tempo di CPU impiegato dal processo (un'approssimazione dello `user time`).

Tempo	Codice iniziale	Codice multicore
<code>elapsed time</code>	75.30 sec.	57.28 sec.
<code>user time</code>	71.57 sec.	49.98 sec.
<code>system time</code>	3.73 sec.	7.29 sec.

Tabella 5.2: Confronto dei tempi iniziali e quelli del codice multicore

Confrontando i valori del codice multicore con quelli del sequenziale iniziale (Tabella 5.2), si può notare che si è ottenuto un miglioramento nel tempo di esecuzione del software Oceanvar. In questocaso infatti si è passati per il tempo totale da 75.30 secondi a 57.28 secondi, con un miglioramento del 23.93%, e per il tempo di CPU impiegato dal processo da 71.57 secondi a 49.98 secondi, con un miglioramento totale del 30.16%.

5.2 Il codice parallelo multiprocessore

Studiando con attenzione il codice, si è notato che molti dati non vengono calcolati utilizzando elementi corrispondenti o contigui. Per tale motivo non è conveniente utilizzare una distribuzione dei dati a blocchi, in quanto si avrebbe bisogno di un numero elevato di comunicazioni tra i processori, che farebbero aumentare il tempo di esecuzione e quindi decadere le prestazioni.

Inoltre, da quanto analizzato per il codice sequenziale, si evince che il carico maggiore di tempo è dato dalle routine `ver_hor` e `ver_hor_ad`. In tali routine entrano in gioco le matrici bidimensionali `grd%eta` e `grd%eta_ad` e le matrici tridimensionali `grd%tem`, `grd%tem_ad`, `grd%sal` e `grd%sal_ad`,

con le quali vengono anche eseguite operazioni diverse e indipendenti.

In Tabella 5.3 é riportato un esempio si codice tratto dalla routine `ver_hor`:

```

if ( drv%bmd( drv%kt r ) . ne . 1 ) &
  grd%eta ( : , : ) = ( grd%eta ( : , : ) + grd%eta_ad ( : , : ) ) * 0.5
  grd%tem ( : , : , : ) = ( grd%tem ( : , : , : ) + grd%tem_ad ( : , : , : ) ) * 0.5
  grd%sal ( : , : , : ) = ( grd%sal ( : , : , : ) + grd%sal_ad ( : , : , : ) ) * 0.5

```

Tabella 5.3: esempio si codice tratto dalla routine `ver_hor`

L'idea per il nostro software é di far eseguire il programma a tre processori.

In Figura 5.2 é mostrato un esempio di parallelizzazione multilivello in cui si utilizzano 3 processori ognuno dei quali sfrutta 4 core per le operazioni multi-thread.

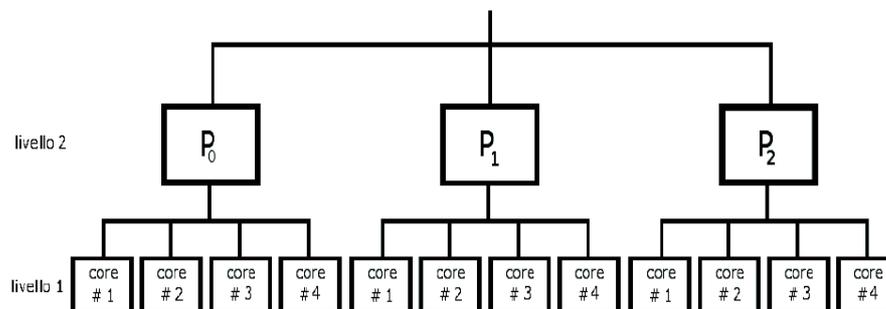


Figura 5.2: Esempio di parallelizzazione multilivello in cui si utilizzano 3 processori ognuno dei quali utilizza 4 core.

I tre processori eseguono tutto il programma contemporaneamente, ma il processore P_0 esegue solo i calcoli relativi alle matrici `grd%eta` e `grd%eta_ad`, il processore P_1 esegue solo i calcoli relativi alle matrici

$grd\%tem$ e $grd\%tem_ad$ mentre il processore P_2 esegue solo i calcoli relativi alle matrici $grd\%sal$ e $grd\%sal_ad$. Alla fine di ogni iterazione c'è poi bisogno della sincronizzazione e dello scambio dati tra i tre processori (Figura 5.3).

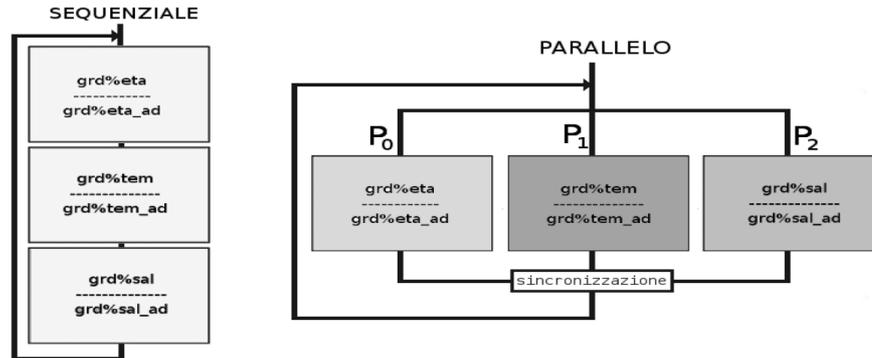


Figura 5.3: Differenza dell'esecuzione dei calcoli tra il codice sequenziale e il codice parallelo multilivello, divisione dei dati nel codice parallelo multilivello.

In tal modo, il codice 5.3 prima riportato assumerebbe la forma:

```

if (mycol . eq . 0 ) then
if ( drv%bmd( drv%ktr ) . ne . 1) &
grd%eta(:, :) = ( grd%eta(:, :) + grd%eta_ad(:, :) ) * 0.5
else if (mycol . eq . 1 ) then
grd%tem(:, :, :) = ( grd%tem(:, :, :) + grd%tem_ad(:, :, :) ) * 0.5 endif
else if (mycol . eq . 2 ) then
grd%sal(:, :, :)= ( grd%sal(:, :, :) + grd%sal_ad(:, :, :) ) * 0.5
endif

```

Tabella 5.4: esempio di codice tratto dalla routine `ver_hor`

Per tale motivo, al posto della routine `ver_hor`, si utilizza ora la routine `ver_horp` strutturata secondo lo schema ad albero illustrato in Figura 5.4.

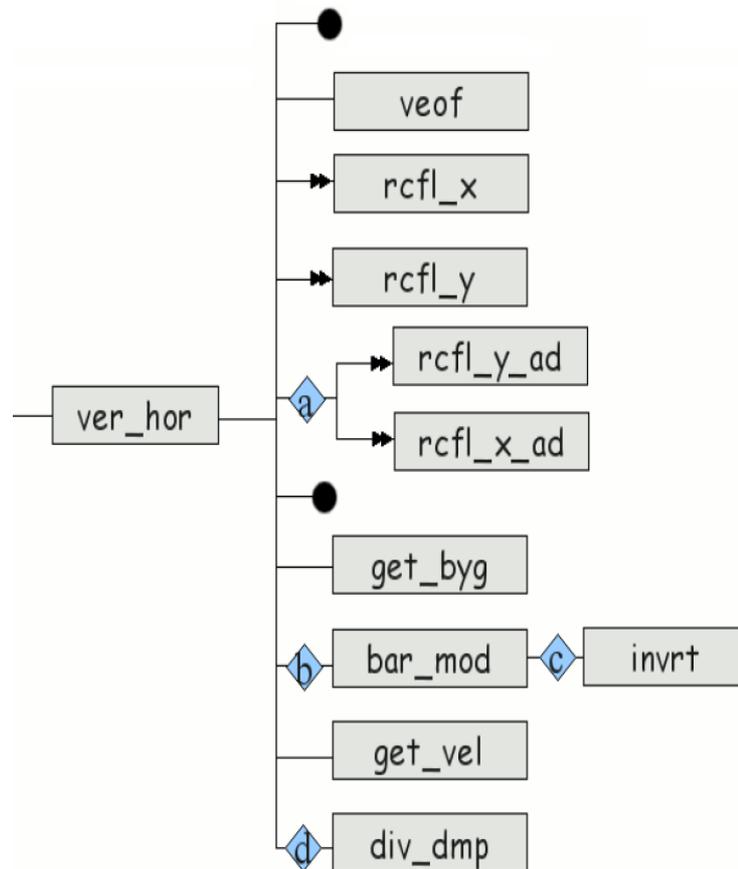


Figura 5.4: Schema ad albero della routine ver_horp.

Il primo pallino indica dove si iniziano a dividere i calcoli tra i tre processori, il secondo indica dove termina la parallelizzazione e si effettua lo scambio di dati tra i processori.

In particolare, ogni processore invia le proprie matrici risultanti agli altri due mediante una routine della libreria Blacs (Basic Linear Algebra Communication Subprograms), utilizzando una comunicazione ad anello (di tipo i-ring).

Al posto di ver_hor ad si utilizza ver_hor_adp, routine strutturata

secondo lo schema ad albero illustrato in Figura 5.5.

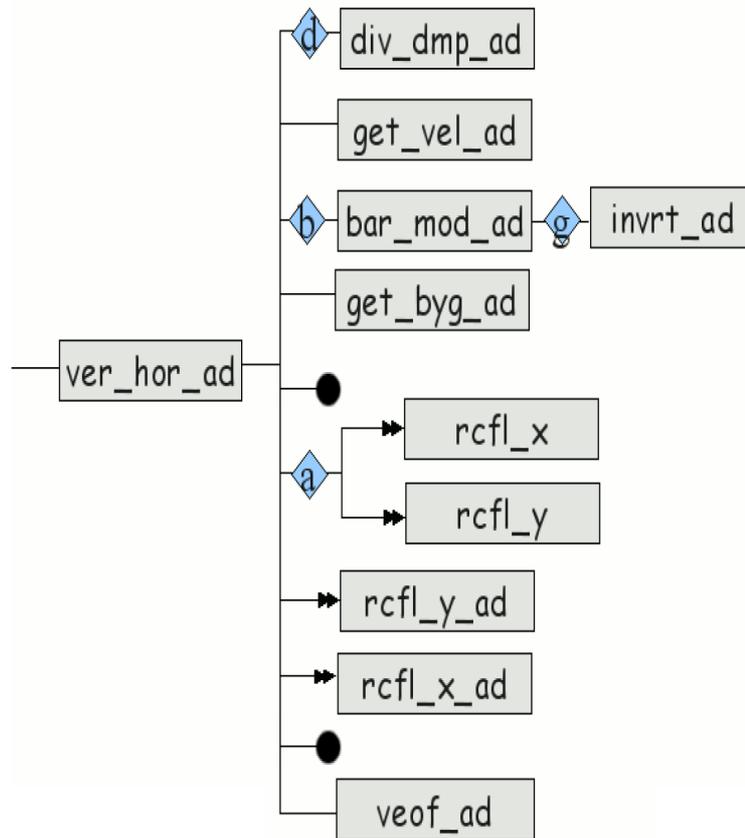


Figura 5.5: Schema ad albero della routine `ver_hor_adp`.

Anche qui, il primo pallino indica dove iniziano a i calcoli concorrenti tra i tre processori, il secondo indica dove termina la parallelizzazione e si effettua lo scambio di dati tra i processori.

In particolare, ogni processore invia la propria matrice risultante agli altri due mediante una routine della libreria Blacs, utilizzando una comunicazione ad anello (di tipo `i-ring`).

Per eseguire in tal modo il software, é stato modificato lo script per la sottomissione. Quello di cui ci si serve é il seguente:

```
.....  
#PBS -S /bin/ksh  
#PBS -q standard  
#PBS -T mpisx  
#PBS -l elapstim_req=1000  
#PBS -l cpunum_job=8  
#PBS -b 3  
#PBS -j o  
work=/home/sm502/OceanVar  
cd $work  
export OMPNUMTHREADS=8  
export MPIPROGINF=YES  
export MPIMULTITASKMIX=YES  
# $Link files with input variables  
  
.....  
  
mpirun -np 3 -nnp 1 ./var_3dp
```

Tabella 5.5:

In particolare:

- é stata sostituita la riga `#PBS -b 1` con la riga `#PBS -b 3` che indica quanti nodi utilizzare, in questo caso si richiedono tre nodi;
- é stata sostituita la riga `mpirun -np 1 -nnp 1 ./var_3dp` con la `mpirun -np 3 -nnp 1 ./var_3dp`, ora si richiede di eseguire il software su tre processori, ognuno in un proprio nodo.

Lasciando inalterata la riga
`export OMP_NUM_THREADS=8`
 si richiede che ogni processore possa utilizzare al piú otto core per le operazioni multi-thread.

É interessante analizzare i tempi di comunicazione in `ver_horp` e `ver_hor_adp`. Il tempo medio speso per le comunicazioni é di 1.11 secondi.

Tempo	Codice iniziale	Parallelo multilivello
elapsed time	75.30 sec.	48.82 sec.
user time	71.57 sec.	39.44 sec.
system time	3.73 sec.	9.38 sec.

Tabella 5.6: Confronto dei tempi iniziali e quelli del parallelo multilivello

Confrontando i valori del parallelo multilivello con quelli del sequenziale iniziale (Tabella 5.6), si osserva un ulteriore miglioramento nel tempo di esecuzione del modulo Oceanvar. In questo caso infatti si é passati per il tempo totale da 75.30 secondi a 48.82 secondi, con un miglioramento del 35.17%, e per il tempo di CPU impiegato dal processo da 71.57 secondi a 39.44 secondi, con un miglioramento totale del 44.89%.

Tempo	Codice multicore	Parallelo multilivello
elapsed time	57.28 sec.	48.82 sec.
user time	49.98 sec.	39.44 sec.
system time	7.29 sec.	9.38 sec.

Tabella 5.7: Confronto dei tempi: codice multicore e parallelo multilivello

Confrontando, infine, i valori del codice multicore con quelli del parallelo multilivello (Tabella 5.7), si osserva un miglioramento nel tempo di esecuzione. In questo caso infatti si é passati per il tempo totale da

57.28 secondi a 48.82 secondi, con un miglioramento del 14.75%, e per il tempo di CPU impiegato dal processo da 49.98 secondi a 39.44 secondi, con un miglioramento totale del 21.09% .

Per avere una visualizzazione totale dei tempi ottenuti, nella Tabella 5.8 sono stati riportati tutti i tempi ottenuti con il codice iniziale, quello multicore e quello parallelo multilivello.

Tempo	Codice iniziale	Codice multicore	Parallelo multilivello
elapsed time	75.30	57.28 sec.	48.82 sec.
user time	71.57	49.98 sec.	39.44 sec.
system time	3.73	7.29 sec.	9.38 sec.

Tabella 5.8: Confronto dei tempi: codice iniziale, codice multicore e parallelo multilivello

In particolare, l'elapsed time e lo user time di ogni codice analizzato sono stati riportati nella Figura 5.6.

Nella Tabella 5.9 sono riportati i guadagni in termini percentuali di ciascuno dei due software realizzati: il codice multicore e il parallelo multilivello.

Guadagno in percentuale	Codice multicore	Parallelo multilivello
Tempo totale	23.93%	35.17%
Tempo CPU	30.16%	44.89%

Tabella 5.9: Guadagno in percentuale del tempo totale e del tempo di CPU per ciascuno dei codice realizzati rispetto a quello iniziale

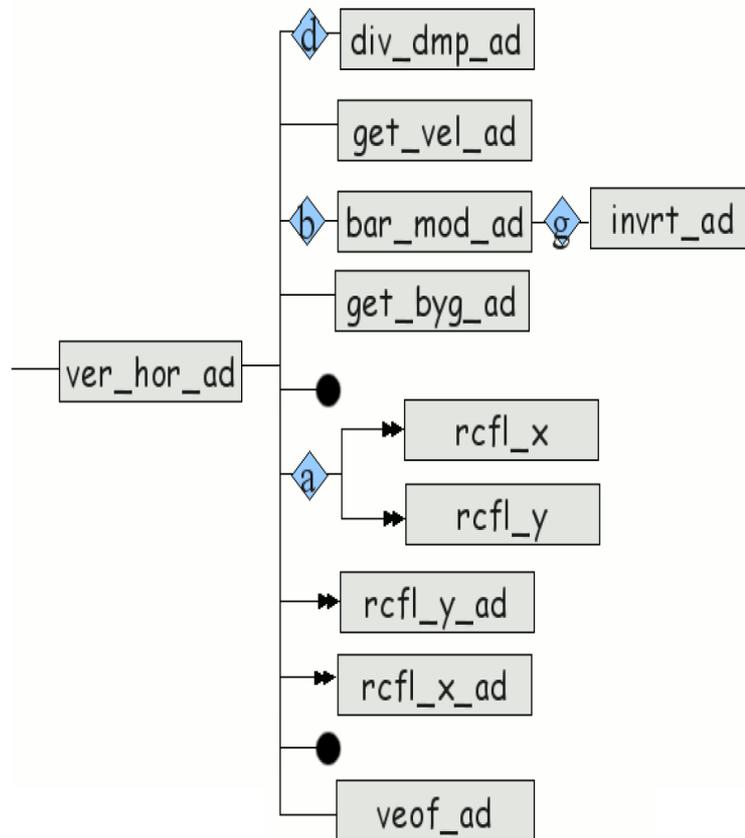


Figura 5.6: Confronto degli elapsed time e degli user time.

Appendice A

Alcuni tra i piú noti Software per *data assimilation*

Per comprendere a fondo le problematiche che occorre affrontare nell'implementazione del problema data assimilation, é stato necessario installare e studiare alcuni tra i software piú utilizzati. Tra questi i software **Oceanvar**, **3DVAR**, **OpenDA** e **DART**, sviluppati e mantenuti da importanti centri di ricerca del mondo che si occupano del problema Data Assimilation.

Di seguito sono riportate delle schede tecniche frutto dello studio e l'analisi di questi software.

A.1 Schede tecniche software DA

Nelle schede tecniche che seguono sono riportati i nomi dei software, i centri di ricerca che li sviluppano, il modello che implementano e dettagli su eventuali strategie di parallelizzazione e architetture hardware di riferimento.

I file dati utilizzati dai software che implementano il problema data assimilation sono di due tipi: i file ASCII e i file nel formato **NetCDF**

(Network Common Data Form).

Il formato NetCDF é stato messo a punto dalla statunitense University Corporation for Atmospheric Research (UCAR), partendo dal modello concettuale della NASA CDF. Tale formato é uno standard open source funzionale alla gestione di dati multidimensionali organizzati in forma matriciale (dove ad esempio uno scalare viene interpretato come un array 0-dimensionale).

Esso é alla base della struttura dati del pacchetto software NetCDF, costituito da un insieme di librerie e formati di dati, indipendenti dalla piattaforma utilizzata, che supportano la creazione, l'accesso e la condivisione di dati scientifici array-oriented. Il formato utilizzato consente un accesso diretto ai dati senza necessariamente conoscere i dettagli relativi alla loro memorizzazione. Inoltre, correlati ai dati ci sono informazioni ausiliari, quali ad esempio l'unita di misura relativa, che consentono a qualsiasi applicazione o utility di accedere ai datasets di tipo NetCDF. Uno degli obiettivi di questo formato é quello di consentire un accesso efficiente a piccoli sottoinsiemi di set di dati di grandi dimensioni. Per tale motivo, il formato NetCDF utilizza un accesso diretto, piuttosto che un accesso sequenziale. Tale caratteristica risulta particolarmente vantaggiosa quando l'ordine in cui i dati vengono letti é diverso dall'ordine in cui sono stati memorizzati, o quando i dati devono essere letti in diversi ordini per diverse applicazioni. L'implementazione corrente, però, limita l'accesso simultaneo a un dataset netCDF. Uno scrittore e piú lettori possono accedere ai dati in un singolo set di dati simultaneamente, ma non vi é alcun supporto per diversi scrittori concorrenti.

Software: **OceanVar** - Ocean Variational Data Assimilation

Institution: INGV- Istituto Nazionale Geofisica e Vulcanologia e CMCC - Centro EuroMediterraneo per i Cambiamenti Climatici.

Home page: non esiste una home page che documenti il software.

Accesso: Chiuso.

Release: ———.

Modello: Modello Variazionale 3D per dati di tipo oceanografico [14]

Metodi numerici: Metodo quasi Newton.

Parallelizzazione: nessuna strategia di parallelizzazione.

Architetture Hardware di riferimento: macchine linux

Tabella A.1: **Software OceanVar**

Software: **3DVAR** - 3D Variational Data Assimilation

Institution: National Center for Atmospheric Research (NCAR), in particolare il sottogruppo che fa capo al progetto Weather Research and Forecasting (WRF).

Home page: <http://www.mmm.ucar.edu/3dvar> .

Accesso: Aperto.

Release: Release per WRF - estate 2003.

Modello: Modello Variazionale 3D per dati di tipo atmosferico [37]

Metodi numerici: Metodo Quasi-Newton (QN) e Metodo del Gradiente Coniugato (CG).

Parallelizzazione: utilizza la strategia di parallelizzazione *domain decomposition*.

Architetture Hardware di riferimento: macchine linux

Tabella A.2: **Software 3DVAR**

Software:OpenDA - an open-source data-assimilation toolbox

Institution:Delft University of Technology.

Home page: <http://www.openda.org/joomla/index.php> .

Accesso: Aperto.

Release: versione 1.0, release 10 Maggio 2010.

Modello: Ensemble method, adattabile a qualunque tipo di dato

Metodi numerici: Metodo del Gradiente Coniugato (CG).

Parallelizzazione: nessuna strategia di parallelizzazione.

Architetture Hardware di riferimento: macchine linux e windows

Tabella A.3: **Software OpenDA**

Software: **DART** - Data Assimilation Research Testbed.

Institution: Data Assimilation Research Section (DAReS) at the National Center for Atmospheric Research (NCAR).

Home page: <http://www.image.ucar.edu/DAReS/DART> .

Accesso: Aperto con registrazione.

Release: jamaica - 12 Apr 2007.

Modello: Ensemble Kalman Filter per dati di tipo atmosferico, adattabile a qualunque tipo di dato [39]

In particolare :

- Modello utilizzato per "selezionare" i dati osservati - Perturbed Observation Ensemble Kalman Filter (EnKF);
- Modello che descrive il problema data assimilation - Ensemble Adjustment Kalman Filter (EAKF).

Metodi numerici:

- Perturbed Observation Ensemble Kalman Filter (EnKF) : Metodo di Monte Carlo;
- Ensemble Adjustment Kalman Filter (EAKF) : Metodo dei Minimi Quadrati.

Parallelizzazione: vedi [41]; utilizza lo standard MPI.

Architetture Hardware e Software di riferimento: macchine linux.

Tabella A.4: **Software DART**

Appendice B

Sui Problemi Inversi Mal Posti

Sia P un problema reale, e $M(P)$ la sua modellizzazione mediante un modello matematico:

$$M(P) : K(x(t)) = y(t), \quad t \in \Omega \quad (\text{B.1})$$

dove $K : X \rightarrow Y$ é un opportuno operatore definito su un dominio Ω , X e Y sono due spazi di Banach e, x e y sono definite su Ω e sono tali che $x \in X$ e $y \in Y$.

Tali problemi possono essere classificati nel modo seguente:

1. **Problema Diretto.** Dati K e x , determinare y .

Un problema di questo tipo é, ad esempio, il calcolo di un integrale definito.

2. **Problema Inverso.** Dati K e y , determinare x .

Esempi di problemi inversi sono la risoluzione di sistemi di equazioni lineari, equazioni differenziali ordinarie e parziali ed equazioni integrali.

3. **Problema di Identificazione.** Dati x e y , trovare K .

E', in sostanza, il problema della costruzione di un modello matematico.

Esempi di problemi di identificazione sono tutti i problemi di approssimazione ad esempio il metodo dei minimi quadrati.

B.1 Problema Inverso Mal Posto

Se il problema da risolvere é un problema inverso si ha che non sempre é possibile invertire l'operatore K poiché:

- i problemi inversi sono spesso mal posti;
- i problemi inversi sono sempre mal condizionati.

Sia P un problema reale, $M(P)$ la sua modellizzazione mediante un modello matematico e $M_h(P)$ il corrispondente modello discreto. Si definisce **indice di condizionamento** di un problema la misura della sensibilità della soluzione rispetto a piccole perturbazioni nei dati di input. In seguito mostreremo nel dettaglio come si possa ottenere una misura dell'indice di condizionamento di un problema.

Osservazione B.1.1. *Nella risoluzione numerica di $M_h(P)$ si riscontra che qualsiasi perturbazione sui dati si propaga sulla soluzione secondo un fattore di amplificazione. Ovvero detto δ l'errore sui dati e σ l'errore corrispondente nella soluzione si ha che*

$$\sigma = c\mu\delta \tag{B.2}$$

dove c é un parametro che dipende dall'algoritmo e μ é l'indice di condizionamento del problema. Si ha che se $\mu \leq 1$ (l'errore nel dato non si amplifica) il problema é ben condizionato e se $\mu > 1$ il problema é mal condizionato.

Si osservi che la soluzione numerica é inaccettabile se:

1. Il problema P é mal condizionato ($\mu > 1$). Dunque l'errore sui dati, δ , viene amplificato e l'errore sulla soluzione calcolata é "grande". Si ha quindi che l'errore sulla soluzione calcolata puó risultare inaccettabile anche se l'errore sui dati é piccolo, ovvero anche se l'algoritmo risulta essere stabile nel senso della b.e.a. (backward error analysis)¹.
2. L'errore sui dati é grande, cioè l'algoritmo é instabile nel senso della b.e.a., anche se il problema é ben condizionato.

Definizione B.1.1. (Problema Ben posto secondo Hadamard)

Un problema (K, X, Y) é ben posto se dati due spazi di Banach X e Y e l'applicazione $K : X \rightarrow Y$ sono verificate le seguenti condizioni:

1. Per ogni $y(t) \in Y$ esiste una soluzione $x(t) \in X$ di (B.1);
2. la soluzione $x(t)$ é unica in X ;
3. la dipendenza di $x(t)$ da $y(t)$ é continua.

Dunque il buon posizionamento di un problema é legato non solo all'operatore K , ma anche agli spazi topologici X e Y ed é quindi una proprietá della terna (K, X, Y) . La prima condizione equivale alla suriettivitá di K , la seconda all'iniettivitá di K e la terza alla continuitá di K^{-1} .

E' detto *mal posto* un problema per il quale non vale almeno una delle condizioni della Definizione (B.1.1).

¹L'analisi dell'errore basata sull'idea di considerare la soluzione, calcolata dall'algoritmo in un sistema aritmetico a precisione finita, come soluzione esatta (ovvero ottenuta in aritmetica a precisione infinita) di un problema dello stesso tipo, con dati perturbati, viene detta **backward error analysis (b.e.a.)** (analisi all'indietro).

Esempio B.1.1. (Risoluzione di un sistema di equazioni lineari)

Sia

$$Ax = b \tag{B.3}$$

un sistema di equazioni lineari con $A \in \mathbf{R}^{n \times m}$.

- *Se $m = n$ e A é una matrice singolare, ovvero non ammette matrice inversa, allora il sistema (B.3) non ammette soluzione (non é verificata la prima condizione della Definizione (B.1.1)).*
- *Se $m > n$ e $\text{rango}(A) < n$ allora il sistema (B.3) non ammette un' unica soluzione (non é verificata la seconda condizione della Definizione (B.1.1)).*

Dunque in entrambi i casi si puó affermare che il problema (B.3) é mal posto secondo Hadamard.



Si é osservato che il mal posizionamento di un problema coincide con la discontinuitá dell'operatore inverso K^{-1} . Notoriamente la continuitá é una proprietá topologica e quindi puó essere studiata ogni volta che il problema reale P é descritto mediante un modello matematico $M(P)$ che sia rappresentato da un'applicazione tra spazi topologici. Il mal posizionamento di un problema continuo $M(P)$ si traduce nel mal condizionamento del corrispondente problema discreto $M_h(P)$, ma anche un problema ben posto puó essere mal condizionato.

Il condizionamento di un problema $M_h(P)$ si misura mediante il suo indice di condizionamento.

Consideriamo il problema

$$Ax = b \tag{B.4}$$

per stimare il condizionamento di un problema é necessario valutare l'errore relativo nella soluzione del sistema

$$\frac{\|\delta x\|}{\|x\|}$$

in funzione degli errori nei dati

$$\frac{\|\delta A\|}{\|A\|} \quad e \quad \frac{\|\delta b\|}{\|b\|}.$$

Supponiamo che ci sia una perturbazione nei dati, e quindi anziché risolvere il problema (B.4), si sta risolvendo il problema perturbato

$$(A + \delta A)\hat{x} = b + \delta b \tag{B.5}$$

Si vuole limitare la norma di $\delta x \equiv \hat{x} - x$. Sottraendo a (B.5) la (B.4) abbiamo

$$(A + \delta A)\hat{x} - Ax = \delta b \tag{B.6}$$

dunque si ha che

$$A\delta x + \delta A\hat{x} = \delta b \tag{B.7}$$

e risolvendo rispetto a δx si ha:

$$\delta x = A^{-1}(-\delta A\hat{x} + \delta b) \tag{B.8}$$

Passando alla norma si ha:

$$\|\delta x\| = \|A^{-1}(-\delta A\hat{x} + \delta b)\| \leq \|A^{-1}\|(\|\delta A\| \cdot \|\hat{x}\| + \|\delta b\|) \tag{B.9}$$

inoltre

$$\frac{\|\delta x\|}{\|\hat{x}\|} \leq \|A^{-1}\| \left(\|\delta A\| + \frac{\|\delta b\|}{\|\hat{x}\|} \right) \tag{B.10}$$

e

$$\frac{\|\delta x\|}{\|\hat{x}\|} \leq \|A^{-1}\| \|A\| \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|\hat{x}\|} \right). \tag{B.11}$$

Dunque la quantità $\|A\| \cdot \|A^{-1}\|$ stima l'errore relativo nei dati sulla soluzione.

Definizione B.1.2. (Indice di condizionamento)

La quantità

$$\mu(A) = \|A^{-1}\| \cdot \|A\|. \quad (\text{B.12})$$

é detta indice di condizionamento del sistema che ha A come matrice dei coefficienti ad esso associata.

Dall'espressione di $\mu(A)$, si deduce che il condizionamento di un sistema lineare é una proprietá intrinseca della matrice dei coefficienti (e non dipende dagli errori di cui é affetto il vettore dei termini noti). Inoltre, é sempre $\mu(A) \geq 1$, in quanto:

$$1 = \mu(I) = \|AA^{-1}\| \leq \|A\| \|A^{-1}\| = \mu(A) \quad (\text{B.13})$$

L'indice di condizionamento misura il massimo fattore di amplificazione dell'errore relativo sulla soluzione, rispetto all'errore relativo sui dati. Un sistema di equazioni é, quindi, *mal condizionato* se $\mu(A) \gg 1$, mentre é *ben condizionato* se $\mu(A) \simeq 1$.

Esempio B.1.2. (Problema ben posto mal condizionato)

Consideriamo il sistema di equazioni

$$\begin{cases} x_1 + x_2 = 3 \\ x_1 + 1.0001x_2 = -2 \end{cases} \quad (\text{B.14})$$

La matrice A dei coefficienti associata al sistema (B.14)

$$A = \begin{pmatrix} 1. & 1. \\ 1. & 1.0001 \end{pmatrix} \quad (\text{B.15})$$

ha determinante non nullo e rango pari a 2. Il sistema (B.14) ammette l'unica soluzione (5.0003, 5.).

Dunque tale sistema di equazioni lineari é ben posto.

Poiché si ha

$$A^{-1} = \begin{pmatrix} 10001 & -10000 \\ -10000 & 10000 \end{pmatrix} \quad (\text{B.16})$$

L'indice di condizionamento di A é:

$$\mu(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = 2.0001 \cdot 20001 \simeq 4 \times 10^4$$

Dunque il sistema di equazioni lineari (B.14) é mal condizionato.

Lemma B.1.1. *Sia $\|\cdot\|$ una norma matriciale submoltiplicativa (cioé $\|A \cdot B\| \leq \|A\| \|B\|$) e compatibile con una norma vettoriale (cioé $\|Ax\| \leq \|A\| \|x\| \quad \forall x \neq 0$). Se una matrice A é tale che $\|A\| < 1$, si ha:*

1. $I + A$ é non singolare;
2. $(1 + \|A\|)^{-1} \leq \|(I + A)^{-1}\| \leq (1 - \|A\|)^{-1}$

Dimostrazione. Per dimostrare la 1. si consideri un vettore $x \neq 0$. Allora si ha:

$$\|(I+A)x\| = \|x+Ax\| \geq \|x\| - \|Ax\| \geq \|x\| - \|x\| \|A\| = \|x\| (1 - \|A\|) > 0$$

Ciò vuol dire che il vettore $(I + A)x \neq 0$, e quindi il sistema di equazioni $(I + A)x = 0$ ammette solo la soluzione banale e quindi la matrice $(I+A)$ é non singolare.

Per dimostrare la 2. si osservi che:

$$1 = \|I\| = \|(I+A)(I+A)^{-1}\| \leq \|I+A\| \|(I+A)^{-1}\| \leq (1+\|A\|) \cdot \|(I+A)^{-1}\|$$

da cui si ha:

$$\|(I + A)^{-1}\| \geq \frac{1}{1 + \|A\|}$$

e quindi la prima disuguaglianza é dimostrata. Inoltre si ha:

$$I = (I + A)(I + A)^{-1} = (I + A)^{-1} + A(I + A)^{-1}$$

da cui:

$$(I + A)^{-1} = I - A(I + A)^{-1}$$

Passando alle norme:

$$\|(I + A)^{-1}\| = \|I - A(I + A)^{-1}\| \leq (1 + \|A\|) \cdot \|(I + A)^{-1}\|$$

e quindi

$$\|(I + A)^{-1}\|(1 + \|A\|) \leq 1$$

Ma, poiché l'ipotesi $\|A\| \leq 1 \Rightarrow (1 - \|A\|) > 0$, si ha

$$\|(I + A)^{-1}\| \leq \frac{1}{1 - \|A\|}$$

e quindi anche la seconda disuguaglianza della 2. è dimostrata. □

Il teorema seguente fornisce una stima dell'indice di condizionamento per un sistema lineare.

Teorema B.1.1. (Teorema del condizionamento)

Sia $\|\cdot\|$ una norma matriciale submoltiplicativa (cioè $\|A \cdot B\| \leq \|A\|\|B\|$) e compatibile con una norma vettoriale (cioè $\|Ax\| \leq \|A\|\|x\| \quad \forall x \neq 0$). Sia inoltre il sistema $Ax = b$ (con A non singolare) e si consideri il sistema perturbato

$$(A + \delta A)(x + \delta x) = (b + \delta b) \tag{B.17}$$

Se

$$\|\delta A\| < \frac{1}{\|A^{-1}\|} \tag{B.18}$$

posto $\mu(A) = \|A^{-1}\| \cdot \|A\|$ si ha

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\mu(A)}{1 - \mu(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right). \tag{B.19}$$

Dimostrazione. Posto $C = A^{-1}\delta A$ si ha

$$\|C\| = \|A^{-1}\delta A\| \leq \|A^{-1}\| \cdot \|\delta A\| < 1$$

per cui applicando il Lemma alla matrice C si ha che la matrice $(I + C) = (I + A^{-1})\delta A$ è non singolare. Poiché inoltre $(A + \delta A) = A(I + A^{-1}\delta A)$ si ha che anche $(A + \delta A)$ è non singolare. Dal Lemma si ha anche che:

$$\|(I + C)^{-1}\| = \|(I + A^{-1}\delta A)^{-1}\| \leq \frac{1}{1 - \|A^{-1}\delta A\|} \leq \frac{1}{1 - \|A^{-1}\| \cdot \|\delta A\|} \tag{B.20}$$

Moltiplicando quindi per A^{-1} il sistema perturbato (B.17) si ha:

$$A^{-1}(A + \delta A)(x + \delta x) = (I + A^{-1}\delta A)(x + \delta x) = A^{-1}b + A^{-1}\delta b$$

da cui

$$(I + A^{-1}\delta A)x + (I + A^{-1}\delta A)\delta x = x + A^{-1}\delta b$$

e quindi

$$(I + A^{-1}\delta A)\delta x = x + A^{-1}\delta b - x - A^{-1}\delta Ax$$

Isolando a primo membro δx e passando alle norme si ha

$$\begin{aligned} \|\delta x\| &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \cdot \|\delta A\|} (\|\delta b\| + \|\delta A\| \cdot \|x\|) \\ \frac{\|\delta x\|}{\|x\|} &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \cdot \|\delta A\|} \left(\|\delta A\| + \frac{\|\delta b\|}{\|x\|} \right) \end{aligned}$$

Ma, poiché $Ax = b$ si ha $\|x\| \geq \frac{\|b\|}{\|A\|}$, dalla precedente espressione, moltiplicando e dividendo per $\|A\|$, si ottiene

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\|A^{-1}\| \cdot \|A\|}{1 - \|A^{-1}\| \cdot \|A\| \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|x\|} \right)$$

e quindi, posto $\mu(A) = \|A^{-1}\| \cdot \|A\|$, si ha

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\mu(A)}{1 - \mu(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right)$$

ovvero la tesi. □

Si osservi che da $\|A\|\|A^{-1}\| \geq 1$, l'ipotesi (B.18), implica che $\|\delta A\| < \|A\|$, che é un'ipotesi realistica, in quanto afferma che l'ordine di grandezza delle perturbazioni é inferiore all'ordine di grandezza degli elementi di A . Se cosí non fosse le perturbazioni altererebbero completamente i dati. Inoltre, la stessa ipotesi (B.18) garantisce che

$$0 < \mu(A) \frac{\|\delta A\|}{\|A\|} < 1$$

e quindi il denominatore al secondo membro della (B.19) sará sempre strettamente positivo.

Teorema B.1.2. *Sia A una matrice non singolare. Allora*

$$\min \left\{ \frac{\|\delta A\|_2}{\|A\|_2} : A + \delta A \text{ singolare} \right\} = \frac{1}{\|A^{-1}\|_2 \cdot \|A\|_2} = \frac{1}{\mu(A)}. \quad (\text{B.21})$$

Dunque, la distanza dalla piú vicina matrice singolare (problema mal posto) é il reciproco dell'indice di condizionamento.

Dimostrazione. Basta dimostrare che $\min\{\|\delta A\|_2 : A + \delta A \text{ singolare}\} = \frac{1}{\|A^{-1}\|_2}$.

Per mostrare che il minimo é minore di $\frac{1}{\|A^{-1}\|_2}$, notiamo che se $\|\delta A\|_2 < \frac{1}{\|A^{-1}\|_2}$, allora $1 > \|\delta A\|_2 \cdot \|A^{-1}\|_2 \geq \|A^{-1}\delta A\|_2$, dunque $I + \|A^{-1}\|_2 \delta A$ é invertibile, e quindi $A + \delta A$ é invertibile.

Per mostrare che il minimo é uguale a $\frac{1}{\|A^{-1}\|_2}$, costruiamo δA con norma $\frac{1}{\|A^{-1}\|_2}$ tale che $A + \delta A$ sia singolare. Notiamo che posto $\|A^{-1}\|_2 = \max_{x \neq 0} \frac{\|A^{-1}x\|_2}{\|x\|_2}$, esiste un x tale che $\|x\|_2 = 1$ e $\|A^{-1}\|_2 = \|A^{-1}x\|_2 > 0$. Sia $y = \frac{A^{-1}x}{\|A^{-1}x\|_2} = \frac{A^{-1}x}{\|A^{-1}\|_2}$ allora $\|y\|_2 = 1$. Sia $\delta A = \frac{-xy^T}{\|A^{-1}\|_2}$.

Allora,

$$\|\delta A\|_2 = \max_{z \neq 0} \frac{\|xy^T z\|_2}{\|A^{-1}\|_2 \|z\|_2} = \max_{z \neq 0} \frac{|y^T z|}{\|z\|_2} \frac{\|x\|_2}{\|A^{-1}\|_2} = \frac{1}{\|A^{-1}\|_2}, \quad (\text{B.22})$$

dove il massimo si ottiene quando z é un multiplo non nullo di y , e $A + \delta A$ é singolare perché

$$(A + \delta A)y = Ay - \frac{xy^T y}{\|A^{-1}\|_2} = \frac{x}{\|A^{-1}\|_2} - \frac{x}{\|A^{-1}\|_2} = 0. \quad (\text{B.23})$$

□

Tale teorema assicura che $A + \delta A$ é non singolare, cosa di cui abbiamo bisogno per l'esistenza di δx e che ci da una caratterizzazione geometrica dell'indice di condizionamento.

Definizione B.1.3. (Problema Mal Posto secondo Demmel)

Un problema é mal posto se il suo indice di condizionamento é infinito.

Esempio B.1.3. (Inversione di matrici)

Sia A una matrice perturbata in $A + \delta A$. La soluzione A^{-1} diventa $A^{-1} + \delta A^{-1}$ dove:

$$\frac{\|\delta A^{-1}\|}{\|A^{-1}\|} \leq \|A^{-1}\| \cdot \|\delta A\|. \quad (\text{B.24})$$

Dunque l'indice di condizionamento di questo problema é $\|A^{-1}\|$. La distanza di A dalla piú vicina matrice singolare (problema mal posto) é $\frac{1}{\|A^{-1}\|}$.

Se si applica la SVD al caso della risoluzione di un sistema di equazioni lineari (B.3) si ha che la soluzione puó essere espressa come

$$x = A^+b = (U\Sigma V^T)^{-1}b = (V\Sigma U^T)b = \sum_{i=1}^n \frac{u_i^T b}{\sigma_i} v_i \quad (\text{B.25})$$

dove la matrice A^+ é un'inversa generalizzata ² di A .

Osservazione B.1.2. Dalla (B.25) si ha che per i valori singolari piú prossimi allo zero, piccoli cambiamenti nella matrice A , e quindi in

2

Teorema B.1.3. Sia $A \in \mathbf{R}^{m \times n}$. Ognuna delle equazioni

$$AXA = A \quad (\text{B.26})$$

$$XAX = X \quad (\text{B.27})$$

$$(AX)^* = AX \quad (\text{B.28})$$

$$(XA)^* = XA \quad (\text{B.29})$$

ammette un'unica soluzione detta **Inversa Generalizzata** di A oppure **Inversa di Moore-Penrose** di A e si scrive $X = A^+$.

Lemma B.1.2. Se A é una matrice non singolare allora $A^+ = A^{-1}$.

$(u_i)_{i=1,\dots,n}$ e $(v_i)_{i=1,\dots,n}$, e in b provocano grandi cambiamenti nella soluzione effettivamente calcolata.

Consideriamo ora le relazioni

$$Av_i = \sigma_i u_i, \quad \|Av_i\|_2 = \sigma_i, \quad i = 1, \dots, n \quad (\text{B.30})$$

$$A^T u_i = \sigma_i v_i, \quad \|A^T u_i\|_2 = \sigma_i, \quad i = 1, \dots, n \quad (\text{B.31})$$

Dalla (B.30) possiamo osservare che se abbiamo un valore singolare σ_i piccolo, rispetto a $\sigma_1 = \|A\|_2$, esiste una combinazione lineare di colonne di A , caratterizzata dagli elementi del vettore singolare destro v_i , tale che $\|Av_i\|_2 = \sigma_i$ é piccola (analogamente vale per u_i e le righe di A). In altre parole se ci sono uno o piú valori singolari piccoli i vettori u_i e v_i associati a σ_i piccoli sono vettori numerici nulli, rispettivamente di A^T e di A . In tal caso la matrice A non ha rango pieno ³ (quindi A é numericamente singolare). Dunque il problema (B.3) é mal posto.

Utilizzando le proprietá della SVD si puó avere una **misura del condizionamento** di A . Infatti

$$\mu(A) = \|A\|_2 \cdot \|A^+\|_2 = \frac{\sigma_1}{\sigma_n} \quad (\text{B.32})$$

Dalla (B.32) si evince che in un problema discreto mal posto, ovvero quando i valori σ_i decadono a zero, $\mu(A) \gg 1$ e la matrice A é forte-

3

Definizione B.1.4. Sia A una matrice di dimensioni $n \times m$, si dice che:

- A ha rango pieno di riga se il rango di A é uguale a n ;
- A ha rango pieno di colonna se il rango di A é uguale a m .

Poiché in A il massimo numero di righe linearmente indipendenti é uguale al massimo numero di colonne linearmente indipendenti, si puó parlare di *rango pieno* sia che $rg(A) = n$ sia che $rg(A) = m$.

Proposizione B.1.1. A é non singolare se e soltanto se A ha rango pieno.

mente mal condizionata.

B.2 Decomposizione SVD (*Singular Value Decomposition*)

Assumiamo la seguente definizione:

Definizione B.2.1. (Valori Singolari)

Sia $A \in \mathbf{R}^{m \times n}$, se

$$\begin{cases} Av = \sigma u \\ A^T u = \sigma v \end{cases} \quad (\text{B.33})$$

per una coppia di vettori $v \in \mathbf{R}^m$ e $u \in \mathbf{R}^n$ non nulli, e σ numero reale positivo, allora σ è detto valore singolare di A e i due vettori v e u si dicono vettori singolari di A .

Teorema B.2.1. (Singular Value Decomposition)

Sia $A \in \mathbf{R}^{m \times n}$ con $m \geq n$. Esistono $U \in \mathbf{R}^{m \times m}$, $V \in \mathbf{R}^{n \times n}$ matrici ortogonali e $\Sigma \in \mathbf{R}^{n \times n}$ matrice diagonale, tali che

$$A = U\Sigma V^T \quad (\text{B.34})$$

Dimostrazione. Dimostriamo il teorema per induzione. Poniamo come base di induzione $n = 1$. Si ha che $A = U\Sigma V^T$ con $U = \frac{A}{\|A\|_2}$, $\Sigma = \|A\|_2$, e $V = 1$.

Assumiamo che la SVD esiste per matrici di dimensioni $(m-1) \times (n-1)$ e dimostriamo che esiste per matrici di dimensioni $m \times n$.

Poiché per definizione $\|A\|_2 = \max_{\|v\|_2=1} \|Av\|_2$, sia v tale che

$$\|v\|_2 = 1 \text{ e } \|A\|_2 = \|Av\|_2 > 0. \quad (\text{B.35})$$

Sia

$$u = \frac{Av}{\|Av\|_2} \quad (\text{B.36})$$

un vettore unitario. Scegliamo \tilde{U} e \tilde{V} tali che $U = [u, \tilde{U}]$ sia una matrice di dimensioni $m \times m$ ortogonale, e $V = [v, \tilde{V}]$ sia una matrice di dimensioni $n \times n$ ortogonale. Si ha dunque che

$$U^T AV = \begin{bmatrix} u^T \\ \tilde{U}^T \end{bmatrix} \cdot A \cdot [v \quad \tilde{V}] = \begin{bmatrix} u^T Av & u^T A\tilde{V} \\ \tilde{U}^T Av & \tilde{U}^T A\tilde{V} \end{bmatrix}.$$

Allora

$$u^T Av = \frac{(Av)^T(Av)}{\|Av\|_2} = \frac{\|Av\|_2^2}{\|Av\|_2} = \|Av\|_2 \stackrel{(B.35)}{=} \|A\|_2 \stackrel{def}{=} \sigma$$

e per la definizione di u si ha $\tilde{U}^T Av = \tilde{U}^T u \|Av\|_2 = 0$. Poiché U e V sono ortogonali $\sigma = \|A\|_2 = \|U^T AV\|_2$ inoltre,

$$\|U^T AV\|_2 \geq \|[1, 0, \dots, 0]U^T AV\|_2 = \|[1, 0, \dots, 0] \begin{bmatrix} u^T \\ \tilde{U}^T \end{bmatrix} A [v \quad \tilde{V}]\|_2 =$$

$$\| \begin{bmatrix} u^T Av & u^T A\tilde{V} \end{bmatrix} \|_2 = \| \begin{bmatrix} \sigma & u^T A\tilde{V} \end{bmatrix} \|_2 > \sigma,$$

allora $u^T A\tilde{V} = 0$.

Dunque, ponendo $\tilde{U}^T A\tilde{V} = \tilde{A}$ si ha

$$U^T AV = \begin{bmatrix} \sigma & 0 \\ 0 & \tilde{U}^T A\tilde{V} \end{bmatrix} = \begin{bmatrix} \sigma & 0 \\ 0 & \tilde{A} \end{bmatrix}.$$

Per l'ipotesi di induzione $\tilde{A} = U_1 \Sigma_1 V_1^T$, dove U_1 ha dimensioni $(m - 1) \times (n - 1)$, Σ_1 ha dimensioni $(n - 1) \times (n - 1)$ e V_1 ha dimensioni $(n - 1) \times (n - 1)$. Si ha

$$U^T AV = \begin{bmatrix} \sigma & 0 \\ 0 & U_1 \Sigma_1 V_1^T \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}^T$$

ovvero

$$A = \left(U \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix} \right) \begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_1 \end{bmatrix} \left(V \begin{bmatrix} 1 & 0 \\ 0 & V_1^T \end{bmatrix} \right)^T$$

da cui la tesi. □

Esempio B.2.1.

$$A = \begin{bmatrix} 0.96 & 1.72 \\ 2.28 & 0.96 \end{bmatrix} = U\Sigma V^T = \begin{bmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{bmatrix}.$$



Assumiamo la seguente definizione:

Definizione B.2.2. (Autovalori)

Data $A \in \mathbf{R}^{n \times n}$, se

$$Ax = \lambda x \tag{B.37}$$

con $x \in \mathbf{R}^n$, $x \neq 0$ e λ numero complesso, allora λ é detto autovalore di A e x autovettore di A .

Osserviamo dalla Definizione (B.2.2) che gli autovalori si definiscono solo per matrici quadrate.

Se $A \in \mathbf{R}^{n \times n}$ si dimostra che gli autovalori e i valori singolari di A sono "legati" dal seguente risultato:

Corollario B.2.1. Sia $A \in \mathbf{R}^{n \times n}$ una matrice simmetrica e sia $A = U\Sigma V^T$ la sua SVD. Se λ_i sono gli autovalori di A e u_i gli autovettori ortonormali, posto $\sigma_i = \text{diag}(\Sigma)$, $U = [u_1, \dots, u_n]$ e $V = [v_1, \dots, v_n]$, allora si ha: $\sigma_i = |\lambda_i|$ e $v_i = \text{sign}(\lambda_i)u_i$.

Dimostrazione. La tesi é vera per il Teorema B.2.1. □

Se $A \in \mathbf{R}^{m \times n}$ si dimostra che i valori singolari di A sono "legati" agli autovalori della matrice quadrata $A^T A \in \mathbf{R}^{n \times n}$:

Teorema B.2.2. *Sia $A \in \mathbf{R}^{m \times n}$, con $m \geq n$ e sia $A = U\Sigma V^T$ la sua SVD. Gli autovalori della matrice simmetrica $A^T A$ sono σ_i^2 , e i suoi vettori singolari destri v_i sono i corrispondenti autovettori ortonormali.*

Dimostrazione. Poiché $A = U\Sigma V^T$ si ha che

$$A^T A = V\Sigma U^T U\Sigma V^T = V\Sigma^2 V^T \quad (\text{B.38})$$

poiché la matrice V é ortogonale, la (B.38) é una trasformazione di similitudini e dunque gli autovalori di $A^T A$ sono gli autovalori di Σ^2 . \square

Teorema B.2.3. *Sia $A \in \mathbf{R}^{m \times n}$, con $m \geq n$ e sia $A = U\Sigma V^T$ la sua SVD. Se $\text{diag}(\Sigma) = (\sigma_1, \sigma_2, \dots, \sigma_n)$ con $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$, allora $\|A\|_2 = \sigma_1$. Inoltre se A é quadrata e non singolare si ha che $\|A^{-1}\|_2^{-1} = \sigma_n$.*

Dimostrazione. Poiché U e V sono matrici ortogonali si ha che

$$\|A\|_2 = \|U^T A V\|_2.$$

Da (B.34) e dal fatto che Σ é una matrice diagonale si ha che

$$\|U^T A V\|_2 = \|\Sigma\|_2 = \rho(\Sigma) = \sigma_1$$

dove $\rho(\Sigma)$ é il raggio spettrale di Σ . Dunque

$$\|A\|_2 = \sigma_1. \quad (\text{B.39})$$

Inoltre

$$\|A^{-1}\|_2 = \|V^T A^{-1} U\|_2 = \|\Sigma^{-1}\|_2 = \sigma_n^{-1}. \quad (\text{B.40})$$

\square

Consideriamo la SVD di una matrice $A \in \mathbf{R}^{m \times n}$ come nell'equazione (B.34), se r é tale che:

$$\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0,$$

allora

$$\text{rango}(A) = r \tag{B.41}$$

$$\text{range}(A) = \text{span}\{u_1, \dots, u_r\}^4 \tag{B.42}$$

Definiamo ora

$$\text{rango}(A, \varepsilon) = \min_{\|A-B\|_2 \leq \varepsilon} \text{rango}(B).$$

Si dimostra il seguente teorema:

Teorema B.2.4. *Sia $A \in \mathbf{R}^{m \times n}$. Se $k < r = \text{rango}(A)$ e*

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T \tag{B.43}$$

allora

$$\min_{\text{rango}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}. \tag{B.44}$$

⁴Dato un insieme di vettori $a_1, \dots, a_n \in \mathbf{R}^m$, l'insieme di tutte le combinazioni lineari di questi vettori costituiscono un sottospazio detto *span* di $\{a_1, \dots, a_n\}$:

$$\text{span}(a_1, \dots, a_n) := \left\{ \sum_{j=1}^n \beta_j a_j : \beta_j \in \mathbf{R} \right\}$$

B.3 Regolarizzazione Numerica

Si é osservato che se il problema

$$Ax = b \tag{B.45}$$

é mal posto, la soluzione x é instabile, ovvero é molto sensibile a perturbazioni nei dati.

Per ridurre tale instabilit  si applicano metodi di *regolarizzazione*. Tali metodi consistono nel definire un nuovo problema, vicino a quello di partenza, che per  risulti essere meglio condizionato, e quindi che abbia una soluzione meno sensibile a perturbazioni rispetto alla soluzione del problema iniziale.

Il metodo di regularizzazione pi  utilizzato é il *metodo dei minimi quadrati*: assegnato il problema

$$Ax = b$$

in forma matriciale

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

tale metodo consiste nel trovare un valore \hat{x} che minimizzi l'errore $e(x)$ tale che

$$e^2(x) = \|A\hat{x} - b\|_2^2 = \sum_{i=1}^m \left[\sum_{j=1}^n a_{ij}\hat{x}_j - b_j \right]^2 \tag{B.46}$$

ovvero calcolare

$$\min \|A\hat{x} - b\|_2^2 \tag{B.47}$$

Nel caso di problemi discreti fortemente mal condizionati i metodi di regularizzazione che si applicano sono:

- *Regolarizzazione di Tikhonov*

- *Decomposizione in Valori Singolari Troncata.*

B.3.1 Metodo di Regolarizzazione di Tikhonov

Il metodo di *Regolarizzazione di Tikhonov* consiste nel calcolare la soluzione regolarizzata x_λ come soluzione del problema

$$\min\{\|Ax - b\|_2^2 + \lambda^2\|L(x - x_0)\|_2^2\}. \quad (\text{B.48})$$

dove x_0 é una stima a priori della soluzione x del problema (B.45), la quantità λ é un *parametro di regolarizzazione* che controlla il peso dato alla minimizzazione della norma di x e L é una matrice scelta in modo appropriato. Tipicamente, L é la matrice identica I_n oppure un'approssimazione discreta di un operatore differenziale.

Ponendo $x_0 = 0$ il problema (B.48) diventa

$$\min\{\|Ax - b\|_2^2 + \lambda^2\|Lx\|_2^2\}. \quad (\text{B.49})$$

Per il calcolo della soluzione di (B.49) consideriamo che

$$\begin{aligned} \|Ax - b\|_2^2 &= (Ax - b)^T(Ax - b) \\ &= (Ax)^T(Ax) - (Ax)^Tb - b^T(Ax) + b^Tb \\ &= (Ax)^T(Ax) - (Ax)^Tb - (Ax)^Tb + b^Tb \\ &= (Ax)^T(Ax) - 2(Ax)^Tb + b^Tb. \end{aligned} \quad (\text{B.50})$$

e che

$$\|Lx\|_2^2 = (Lx)^T(Lx) \quad (\text{B.51})$$

allora derivando la (B.50) e la (B.51) e imponendo che la loro somma sia uguale a zero si ha

$$2A^T Ax - 2A^T b + 2\lambda^2 L^T Lx = 0 \quad (\text{B.52})$$

$$(A^T A - \lambda^2 L^T L)x = A^T b \quad (\text{B.53})$$

e dunque

$$x_\lambda = (A^T A - \lambda^2 L^T L)^{-1} A^T b. \quad (\text{B.54})$$

Posto

$$A_\lambda^+ = [A^T A + \lambda^2 L^T L]^{-1} A^T. \quad (\text{B.55})$$

si ha che

$$x_\lambda = A_\lambda^+ b \quad (\text{B.56})$$

dove $A_\lambda^+ \in \mathbf{R}^{m \times n}$ é un'inversa generalizzata di A .

Dunque si ha che la soluzione regolarizzata x_λ di (B.49) può essere espressa in termini di SVD di A come

$$x_\lambda = V \Sigma_\lambda^+ U^T b. \quad (\text{B.57})$$

dove $\Sigma_\lambda^+ \in \mathbf{R}^{n \times n}$ é una matrice diagonale,

$$\Sigma_\lambda^+ = \begin{pmatrix} \frac{\sigma_1^2}{\sigma_1^2 + \lambda^2} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\sigma_n^2}{\sigma_n^2 + \lambda^2} \end{pmatrix} = \begin{pmatrix} f_1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & f_n \end{pmatrix} \quad (\text{B.58})$$

i cui elementi sono i valori singolari della matrice A_λ^+ e sono detti *fattori di filtro* del metodo di regolarizzazione.

Dunque i fattori di filtro del metodo di regolarizzazione di Tikhonov sono:

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2}, \quad i = 1, \dots, n \quad (\text{B.59})$$

In generale questi fattori sono:

- vicini a 1 per σ_i grandi;
- molto piú piccoli di 1 per σ_i piccoli.

La soluzione generalizzata del problema (B.49) in termini di SVD é

$$x_\lambda = \sum_{i=1}^n f_i \frac{u_i^T b}{\sigma_i} v_i = \sum_{i=1}^n \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \frac{u_i^T b}{\sigma_i} v_i. \quad (\text{B.60})$$

Si osservi in corrispondenza dei piú piccoli valori singolari σ_i , un valore grande di λ fa si che la norma della soluzione sia piccola, mentre un valore piccolo di λ ha l'effetto opposto. Quindi, il parametro di regolarizzazione λ controlla le proprietá della soluzione regolarizzata. Si dice che.

- per λ grande ($\lambda > \sigma_1$) si ha una forte regolarizzazione;
- per λ piccolo si ha una debole regolarizzazione;

Inoltre dalla (B.59) e dalla formula della soluzione regolarizzata (B.60) si ha che l'indice di condizionamento di A_λ^+ é

$$\mu(A_\lambda^+) = \frac{\sigma_1}{2\lambda} \quad (\text{B.61})$$

Dunque piú grande é il valore di λ , piú piccolo é l'indice di condizionamento del metodo, e dunque la soluzione regolarizzata é meno sensibile alle perturbazioni.

B.3.2 Decomposizione in Valori Singolari Troncata

La TSVD (*Truncated Singular Value Decomposition*) sostituisce il problema $Ax = b$, la cui matrice dei coefficienti A é mal condizionata, con un problema $A_k x = b$ la cui matrice dei coefficienti A_k é ben conzionata.

Quindi, la TSVD risolve il problema

$$A_k x_k = b \quad (\text{B.62})$$

dove

$$A_k \equiv U \Sigma_k V^T \quad (\text{B.63})$$

é la matrice di rango $k < n$ e

$$\Sigma_k = \begin{pmatrix} \sigma_1 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_k & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix} \quad (\text{B.64})$$

é una matrice diagonale $m \times n$ uguale a Σ con gli $n - k$ valori singolari piú piccoli sostituiti da zeri. Si osservi che, poiché

$$\sigma_1 > \dots > \sigma_k > \dots > \sigma_n,$$

l'indice di condizionamento di A_k

$$\mu(A_k) = \frac{\sigma_1}{\sigma_k}$$

é piú piccolo di $\mu(A) = \frac{\sigma_1}{\sigma_n}$.

Sia A_k^+ l'inversa generalizzata di A_k ottenuta mediante una SVD:

$$A_k^+ = V \Sigma_k^+ U^T \quad (\text{B.65})$$

dove la matrice

$$\Sigma_k^+ = \begin{pmatrix} \frac{1}{\sigma_1} & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{\sigma_k} & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix} \quad (\text{B.66})$$

é una matrice diagonale di dimensioni $m \times n$ la soluzione di (B.3), ottenuta mediante la TSVD, é definita da:

$$x_k = A_k^+ b = \sum_{i=0}^k \frac{u_i^T b}{\sigma_i} v_i. \quad (\text{B.67})$$

Per l'Osservazione B.1.2, tale soluzione non é molto sensibile a perturbazioni in b e in A in quanto la TSVD tende a eliminare i contributi apportati dai valori singolari piú piccoli alla soluzione.

Il metodo TSVD é importante perché rende piú semplice l'analisi del metodo di regolarizzazione di Tikhonov. Infatti applicando il metodo di regolarizzazione di Tikhonov con la TSVD i fattori di filtro (B.59) diventano:

$$f_i = \begin{cases} 1, & \text{se } \sigma_i \geq \sigma_k \\ 0, & \text{se } \sigma_i < \sigma_k \end{cases} \quad i = 1, \dots, n \quad (\text{B.68})$$

i quali tagliano nettamente gli ultimi $n - k$ valori singolari.

B.3.3 Approccio variazionale

Sia $f(P)$ una funzione delle n variabili x_1, x_2, \dots, x_n tale che $f \in C^2(\Omega)$, con Ω aperto di \mathbf{R}^n .

Definizione B.3.1. *Un punto P_0 di Ω é un punto di **massimo (minimo) relativo** per la funzione $f(P)$ se esiste un suo intorno $I(P_0)$ tale che $\forall P \in I(P_0)$ si ha:*

$$f(P) \leq f(P_0) \quad (f(P) \geq f(P_0)) \quad (\text{B.69})$$

Teorema B.3.1. *Condizione necessaria affinché la funzione $f(P)$ abbia nel punto P_0 un punto di massimo (minimo) relativo é che in esso si annullino le derivate parziali prime di $f(P)$*

$$\frac{\partial}{\partial x_1} f(P_0) = \frac{\partial}{\partial x_2} f(P_0) = \dots = \frac{\partial}{\partial x_n} f(P_0) = 0 \quad (\text{B.70})$$

e che le derivate parziali seconde siano tali che la forma quadratica

$$\sum_{i,j} \frac{\partial}{\partial x_i \partial x_j} f(P_0) \lambda_i \lambda_j \quad (\text{B.71})$$

risulti semidefinita negativa ⁵ (positiva).

Definizione B.3.2. Un punto P_0 di Ω per cui vale (B.70) si dice **estremale**. Il punto P_0 si dice **estremante** se é un punto di massimo e di minimo relativo.

Consideriamo

$$f, \varphi_1, \varphi_2, \dots, \varphi_p, \quad (\text{B.73})$$

$p + 1$ funzioni di $n + p$ variabili $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_p$ definite in un insieme aperto Ω e tali che $f \in C^1(\Omega)$ e $\varphi_i \in C^1(\Omega), \forall i = 1, \dots, p$.

Indichiamo con Φ l'insieme dei punti di Ω le cui coordinate soddisfano alle equazioni:

$$\varphi_i(x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_p) = 0, \quad i = 1, \dots, p \quad (\text{B.74})$$

Sia $P_0 = (x_1^0, \dots, x_n^0; y_1^0, \dots, y_p^0)$ un punto di Φ tale che per ogni altro punto P di Φ , appartenente ad un opportuno intorno $I(P_0)$, si abbia:

$$f(P) \leq f(P_0) \quad \text{oppure} \quad f(P) \geq f(P_0). \quad (\text{B.75})$$

Il problema di determinare i punti di Ω soddisfacenti alle condizioni (B.74) e (B.75) si dice **problema di estremo condizionato**, in quanto le variabili da cui dipende la funzione f , della quale si cercano i massimi e i minimi relativi, non sono indipendenti, ma legate dalle condizioni

⁵Una forma quadratica

$$\sum_{h,k} a_{hk} \lambda_h \lambda_k \quad (\text{B.72})$$

si dice **semidefinita positiva (negativa)** se assume, per ogni insieme di valori $\lambda_1, \lambda_2, \dots, \lambda_n$, valori non negativi (non positivi).

(B.74).

Un metodo per la risoluzione di problemi di estremo condizionato consiste nell'assumere la (B.74) con la condizione che in ogni punto di Φ si abbia:

$$\frac{\partial(\varphi_1, \varphi_2, \dots, \varphi_p)}{\partial(y_1, y_2, \dots, y_p)} \neq 0 \quad (\text{B.76})$$

Tale condizione assicura che nell'intorno di ogni punto di Φ , in particolare in P_0 , é possibile esplicitare le y in funzione delle x ⁶. Queste funzioni, se note, sostituite alle y nella f , consentono di ricondurre il problema di estremo condizionato a un problema di estremo libero.

Un altro metodo per la risoluzione di problemi di estremo condizionato é costituito dal **Metodo dei moltiplicatori di Lagrange** che si basa sull'assunzione del seguente teorema:

Teorema B.3.3. *Siano $f, \varphi_1, \varphi_2, \dots, \varphi_p, p+1$ funzioni di $n+p$ variabili $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_p$ definite in un insieme aperto Ω e tali che $f \in C^1(\Omega)$ e $\varphi_i \in C^1(\Omega)$. Se P_0 é un punto di $\Phi \cap \overset{\circ}{\Omega}$ con $\overset{\circ}{\Omega}$ aperto di Ω dove la f assume un massimo o un minimo relativo rispetto ai valori che essa assume su Φ , allora esistono p costanti $\lambda_1, \lambda_2, \dots, \lambda_p$ tali che P_0 risulta un estremale per la funzione*

$$f + \lambda_1\varphi_1 + \lambda_2\varphi_2 + \dots + \lambda_p\varphi_p. \quad (\text{B.79})$$

Dimostrazione. Dimostriamo il teorema limitandoci, per semplicitá, al caso in cui: $n = 1$ e $p = 2$.

6

Teorema B.3.2. *Sia $f(x, y, z) \in C^1(\Omega)$. Se in un punto $P_0 = (x_0, y_0, z_0)$ di Ω si ha*

$$\begin{cases} f(x_0, y_0, z_0) = 0 \\ \frac{\partial}{\partial(y,z)} f(P_0) \neq 0 \end{cases} \quad (\text{B.77})$$

esiste un $\delta > 0$ tale che in un intorno $I(x_0)$ esiste una e una sola coppia di funzioni per cui in $(x_0 - \delta, x_0 + \delta)$ si ha

$$\begin{cases} f(x, y(x), z(x)) = 0 \\ y(x_0) = y_0, z(x_0) = z_0. \end{cases} \quad (\text{B.78})$$

Sia allora $P_0 \equiv (x_0, y_0, z_0)$ un punto dell'insieme Φ dei punti $P \equiv (x, y, z)$ le cui coordinate soddisfano alle relazioni:

$$\begin{cases} \varphi_1(x, y, z) = 0 \\ \varphi_2(x, y, z) = 0 \end{cases} \quad (\text{B.80})$$

e appartengono all'insieme aperto Ω dove la funzione $f(x, y, z)$ é definita, continua con le derivate parziali prime. Supponiamo per fissare le idee che P_0 sia un punto di massimo relativo per la funzione f su Φ . Ciò comporta che, per ogni punto $P \equiv (x, y, z)$ di Φ sufficientemente prossimo a P_0 , si ha:

$$f(x, y, z) \leq f(x_0, y_0, z_0) \quad (\text{B.81})$$

Nell'ulteriore ipotesi che in Φ sia:

$$\frac{\partial(\varphi_1, \varphi_2)}{\partial(y, z)} \neq 0 \quad (\text{B.82})$$

é possibile determinare un numero $\delta > 0$ tale che in un opportuno intorno di P_0 l'insieme Φ é suscettibile della rappresentazione locale:

$$y = y(x), \quad z = z(x), \quad x_0 - \delta < x < x_0 + \delta \quad (\text{B.83})$$

essendo inoltre:

$$y(x_0) = y_0, \quad z(x_0) = z_0 \quad (\text{B.84})$$

Segue dalla (B.81) che, posto:

$$F(x) = f(x, y(x), z(x)) \quad (\text{B.85})$$

deve aversi, in un opportuno intorno di x_0 :

$$F(x) \leq F(x_0) \quad (\text{B.86})$$

e ciò implica, essendo $F(x)$ derivabile: $F'(x_0) = 0$, ossia:

$$\frac{\partial}{\partial x} f(x_0, y_0, z_0) + \frac{\partial}{\partial y} f(x_0, y_0, z_0) y'(x_0) + \frac{\partial}{\partial z} f(x_0, y_0, z_0) z'(x_0) = 0 \quad (\text{B.87})$$

D'altra parte deve anche essere:

$$\begin{aligned} \frac{\partial}{\partial x}\varphi_1(x_0, y_0, z_0) + \frac{\partial}{\partial y}\varphi_1(x_0, y_0, z_0)y'(x_0) + \frac{\partial}{\partial z}\varphi_1(x_0, y_0, z_0)z'(x_0) &= 0 \\ \frac{\partial}{\partial x}\varphi_2(x_0, y_0, z_0) + \frac{\partial}{\partial y}\varphi_2(x_0, y_0, z_0)y'(x_0) + \frac{\partial}{\partial z}\varphi_2(x_0, y_0, z_0)z'(x_0) &= 0 \end{aligned}$$

Ma la coesistenza di queste ultime tre relazioni implica l'annularsi del determinante:

$$\left. \frac{\partial(f, \varphi_1, \varphi_2)}{\partial(x, y, z)} \right|_{P_0} = 0 \quad (\text{B.88})$$

e di qui segue l'esistenza di due costanti λ_1, λ_2 tali che in P_0 si abbia:

$$\begin{cases} \frac{\partial}{\partial x}f + \lambda_1 \frac{\partial}{\partial x}\varphi_1 + \lambda_2 \frac{\partial}{\partial x}\varphi_2 = 0 \\ \frac{\partial}{\partial y}f + \lambda_1 \frac{\partial}{\partial y}\varphi_1 + \lambda_2 \frac{\partial}{\partial y}\varphi_2 = 0 \\ \frac{\partial}{\partial z}f + \lambda_1 \frac{\partial}{\partial z}\varphi_1 + \lambda_2 \frac{\partial}{\partial z}\varphi_2 = 0 \end{cases} \quad (\text{B.89})$$

e quindi l'asserto é provato. □

In generale i massimi e i minimi relativi di f su Φ sono in punti le cui coordinate soddisfano le equazioni (B.74) e il sistema

$$\begin{cases} \frac{\partial}{\partial x_j}f + \sum_{k=1}^p \lambda_k \frac{\partial}{\partial x_j}\varphi_k = 0 \quad (j = 1, \dots, n) \\ \frac{\partial}{\partial y_i}f + \sum_{k=1}^p \lambda_k \frac{\partial}{\partial y_i}\varphi_k = 0 \quad (i = 1, \dots, p) \end{cases} \quad (\text{B.90})$$

Consideriamo ora il problema (B.1). Abbiamo visto che per risolvere il problema inverso, che consiste nel determinare x essendo noti K e y , si ha che non sempre si puó invertire l'operatore K , in quanto tali problemi sono spesso mal posti e sempre mal condizionati. Dunque per la risoluzione di tali problemi é necessario applicare dei metodi di regolarizzazione che, come abbiamo visto, consistono nel risolvere il problema

$$x = K^{-1}y \quad \text{con l'aggiunta di vincoli } C_1, C_2, \dots, C_n. \quad (\text{B.91})$$

ovvero, é necessario risolvere il problema

$$\min \|y - K(x)\|_2^2 \quad \text{con i vincoli } \|C_1x\|_2^2, \|C_2x\|_2^2, \dots, \|C_nx\|_2^2. \quad (\text{B.92})$$

Posto

$$f = \|y - K(x)\|_2^2 \quad \text{e} \quad \varphi_i = \|C_ix\|_2^2, \quad \forall i = 1, \dots, n \quad (\text{B.93})$$

il teorema (B.3.3) ci assicura che risolvere il problema con vincoli (B.91) equivale a risolvere il problema

$$\|y - K(x)\|_2^2 + \lambda_1 \|C_1x\|_2^2 + \lambda_2 \|C_2x\|_2^2 + \dots + \lambda_n \|C_nx\|_2^2. \quad (\text{B.94})$$

Posto

$$G[x(t)] = \|y - K(x)\|_2^2 + \lambda_1 \|C_1x\|_2^2 + \lambda_2 \|C_2x\|_2^2 + \dots + \lambda_n \|C_nx\|_2^2 \quad (\text{B.95})$$

si osserva che calcolare il minimo del problema (B.94) equivale a minimizzare il funzionale (B.95). Tale posizione conduce ad un approccio di risoluzione del problema (B.91) detto **Approccio Variazionale**, che consiste nel trovare una soluzione di (B.91) mediante il

$$\min_{x(t)} G[x(t)]. \quad (\text{B.96})$$

Tale minimo si calcola imponendo che la derivata prima di $G[x(t)]$ si annulli ovvero

$$\begin{cases} \frac{\partial}{\partial x_1} G = 0 \\ \frac{\partial}{\partial x_2} G = 0 \\ \dots \\ \frac{\partial}{\partial x_n} G = 0 \end{cases} \quad (\text{B.97})$$

da cui si ottiene il sistema

$$\begin{cases} \frac{\partial}{\partial x_1} \|y - K(x)\|_2^2 + \sum_{k=1}^n \lambda_k \frac{\partial}{\partial x_1} \|C_kx\|_2^2 = 0 \\ \frac{\partial}{\partial x_2} \|y - K(x)\|_2^2 + \sum_{k=1}^n \lambda_k \frac{\partial}{\partial x_2} \|C_kx\|_2^2 = 0 \\ \dots \\ \frac{\partial}{\partial x_n} \|y - K(x)\|_2^2 + \sum_{k=1}^n \lambda_k \frac{\partial}{\partial x_n} \|C_kx\|_2^2 = 0 \end{cases} \quad (\text{B.98})$$

che coincide con il sistema (B.90).

Bibliografia

- [1] La Data Assimilation come problema inverso mal posto - L.D'Amore - CMCC, Bologna - Marzo 2010
- [2] A Parallel Three-dimensional Variational Data Assimilation Scheme - L. D'Amore, R. Arcucci, L.Marcellino, A. Murli - Numerical Analysis and Applied Mathematics ICNAAM 2011, AIP Conf. Proc.1389, 1829-1831 (2011)
- [3] Metodi Numerici per la Data Assimilation. Il filtro di Kalman - L.D'Amore et al. - Research Papers_Issue 2011
- [4] Atmospheric modeling, data assimilation and predictability - E.Kalnay - 2003 Cambridge
- [5] Inverse Modeling of the Ocean and Atmosphere - A.F.Bennett - 2004 Cambridge
- [6] An oceanographic three-dimensional variational data assimilation scheme - S.Dobricic, N.Pinardi - 2008 Elsevier
- [7] Analysys methods for numerical weather prediction - A.C.Lorenc - The Quarterly Journal of the Royal Meteorological Society, Vol.112, Issue 474, 1986 pp.1177-1194
- [8] Global Oceanographic Variational Data Assimilation in-citu observation and space-bprn altimeter data for reanalysis applications - A.Storto , S.Dobricic , S.Masina , P.Di Pietro.

-
- [9] Observation Operator in Variational Data Assimilation - D.Tan ,
A.Benedetti
- [10] A Multigrid Tutorial - W.L.Briggs - SIAM 1987
- [11] Recursive filter objective analysis of meteorological fields: applications
to Nesdis operational processing - C.M.Hayden , R.J.Purses - Journal
of Applied Meteorology, Val.34, 1995, pp3-15
- [12] Introduction to MM5 3D-Var Data Assimilation System: Theoretical
basis - F.Vandenberghe , Y-H-Kuo
- [13] Fisica dell'Atmosfera - S.Serafin - 2008
- [14] Mediterranean Forecasting System: An improved assimilation
scheme for sea-level anomaly and its validation - S.Dobric, N.Pinardi
. M.Adani - A.Bonazzi- C.Fratianni - M. Tonani - 2006
- [15] Tesi di laurea magistrale: Su Alcuni Metodi di Discretizzazione per
la Risoluzione Numerica di Operatori Differenziali - Rossella Arcucci,
relatore: Prof.A. Murli - Università degli Studi di Napoli Federico II
, facoltà di MM.FF.NN. - A.A. 2006-2007
- [16] Matematica Numerica Parte I: metodi, algoritmi e software -
A.Murli -2007.
- [17] Matematica Numerica Parte II: metodi, algoritmi e software -
A.Murli -2008.
- [18] An Introduction to the Kalman Filter - Greg Welch and Gary Bishop
- UNC-Chapel Hill, 2006.
- [19] A new approach to linear filtering and prediction problems -
R.E.Kalman - J.Basic Emg. vol.82D, pp.35-45 - Mar. 1960
- [20] Least-square estimation: from Gauss to Kalman - H.W.Sorenson -
Cop. 2004, reprinted from IEEE Spectrum, vol7, pp. 63-68, 1970

-
- [21] THE DATA ASSIMILATION RESEARCH TESTBED - Jeffrey Anderson, Tim Hoar, Kevin R aeder, Hui Liu, Nancy Collins, Ryan Torn, Avelino Avellano - settembre 2009.
- [22] The Data Assimilation Research Section (DAReS) - Data Assimilation Research Testbed (DART) - <http://www.image.ucar.edu/DAReS/DART>
- [23] Data Assimilation Research Testbed Tutorial - Section22: Parallel Algorithm implementation - Version 2.0 - November 2006.
- [24] Appendix E of MM5 Tutorial, Pennsylvania State University / National Center for Atmospheric Research , 2008.
- [25] MM5 3DVAR Data Assimilation Home Page - <http://www.mmm.ucar.edu/3dvar>
- [26] Applied Numerical Linear Algebra, Demmel J.W., SIAM
- [27] On Condition Numbers and the Distance to the Nearest Ill-posed Problem, Demmel J.W., Numer.Math.,51, 251-289, 1987
- [28] Matrix Computations, Second Edition ,Golub G.H., Van Loan C.F.
- [29] The theory of Tikhonov regularization for Fredholm equations of the first kind, Groetsch C.W., Pitman Advanced Publishing Program
- [30] Rank-Deficient and Discrete Ill-Posed Problems, Numerical Aspects of Linear Inversion, Hansen P.C., SIAM
- [31] The Truncated SVD as a Method For Regularization, Hansen P.C., BIT 27 (1987), 534-553
- [32] A Generalized Inverse for Matrices, Penrose R., Proc. Cambridge Philos. Soc.51(1955), 406-413.
- [33] Stampacchia G., *Lezioni di Analisi Matematica*, Parte Seconda, Liguori Editore

-
- [34] R.Esposito - A.Russo , *Lezioni di Geometria*, Parte Prima, Liguori Editore
- [35] F.Vandenberghe - Y.H.Kuo, *Introduction to MM5 3D-Var Data Assimilation System: Theoretical basis*
- [36] N.Nichols - A.Lawless - C.Boess - A.Bunse-Gerstner , *Using Model Reduction in Data Assimilation*, The University of Reading.
- [37] Appendix E of MM5 Tutorial, Pennsylvania State University / National Center for Atmospheric Research , 2008.
- [38] MM5 3DVAR Data Assimilation Home Page - <http://www.mmm.ucar.edu/3dvar>
- [39] THE DATA ASSIMILATION RESEARCH TESTBED - Jeffrey Anderson, Tim Hoar, Kevin R aeder, Hui Liu, Nancy Collins, Ryan Torn, Avelino Avellano - settembre 2009.
- [40] The Data Assimilation Research Section (DAReS) - Data Assimilation Research Testbed (DART) - <http://www.image.ucar.edu/DAReS/DART>
- [41] Data Assimilation Research Testbed Tutorial - Section22: Parallel Algorithm implementation - Version 2.0 - November 2006.
- [42] L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization - J. Nocedal R.H. Byrd, P. Lu and C. Zhu - ACM Transactions on Mathematical Software, Vol. 23, No. 4, December 1997, Pages 550-560.
- [43] Blas - Blacs User Manual - Netlib <http://www.netlib.org/blas-blacs>.
- [44] Lapack - Linpack User Manual - Netlib <http://www.netlib.org/lapack-linpack>.

-
- [45] GLOBAL OCEAN DATA ASSIMILATION: PROSPECTS & STRATEGIES, USGODAE Workshop, April 23-25, 2001 University of Maryland.
- [46] NEC SUPERCOMPUTER, <http://www.nec.com.ar/pdf/1.2.5/sx8r.pdf>
- [47] Genomic Data Assimilation for Estimating Hybrid Functional Petri Net from Time-Course Gene Expression Data - M.Nagasaki, R.Yamaguchi, R.Yoshida, S.Imoto, A.Do, Y.Tamada, H.Matsuno, S.Miyano, T,Higuchi - *Genome Informatics* 17(1): 46-61 (2006)