

Inversione dell'equazione scalare delle onde

Mauro Ennas

Free Report
Creative Commons 2013

Creative Commons 2013
Estratto dalla tesi di laurea 1998

Indice

1 Migrazione dei dati sismici	1
1.1 Migrazione in profondità	4
1.1.1 Zero-offset e riflettori esplodenti	6
1.1.2 Stacking	8
1.1.3 Estrapolazione in profondità	9
1.2 Migrazione a velocità costante nel dominio di Fourier	10
1.2.1 Traslazione di fase	11
1.2.2 Migrazione f-k	13
1.3 Migrazione in mezzi non omogenei	21
1.3.1 Traslazione di fase più interpolazione	24
1.4 L'integrale di Kirchhoff	28
1.4.1 Migrazione mediante l'integrale di Kirchhoff	31
2 Migrazione Reverse Time dell'equazione d'onda completa	35
2.1 Implementazione numerica della migrazione	37
2.1.1 Metodi espliciti a più passi	38
2.1.2 Schema d'integrazione leap-frog a passo frazionario	41
2.1.3 Discretizzazione del laplaciano	43
2.1.4 Stabilità numerica	54

3 Implementazione parallela del migratore RTM	57
3.1 Il programma principale	58
3.2 Prestazioni del solutore tridiagonale parallelo	60
3.3 Esempi di simulazione	65
Bibliografia	

Capitolo 1

Migrazione dei dati sismici

L'obiettivo principale delle tecniche di prospezione geofisica, fondate sulla soluzione dell'equazione scalare delle onde, è quello di costruire una mappa acustica della crosta terrestre.

Questo obiettivo non è facilmente raggiungibile.

Possiamo introdurre delle ipotesi semplificative che permettano un primo approccio al problema: la crosta terrestre può essere immaginata come un mezzo stratificato e costituito da formazioni di diversa natura ciascuna delle quali con specifiche proprietà meccaniche.

Per ricostruire un'immagine acustica della crosta terrestre è necessario essere in grado di controllare il complesso gioco di riflessioni e rifrazioni delle onde emesse da una sorgente artificiale collocata in superficie che, propagandosi in direzione verticale e distorcendosi attraverso i vari strati litologici, vengono retropropagate verso la superficie stessa.

Il processo di migrazione, attraverso l'equazione scalare delle onde, utilizza i dati registrati in superficie e ripropagandoli in profondità, indietro nel tempo, li focalizza sulle discontinuità dell'impedenza acustica del mezzo fornendo una mappa dei coefficienti di riflettività del sottosuolo.

Per ridurre la complessità del problema, vengono ignorate le variazioni spaziali della densità; viene assunto noto il campo di velocità nel mezzo con valori, compresi tra $0.4 \text{ Km/sec. (sabbia)}$ e $7.0 \text{ Km/sec. (granito)}$ ¹.

¹Per uno stesso materiale la velocità aumenta al crescere della profondità.

Sotto queste ipotesi l'intero processo di migrazione può considerarsi basato su una stima *a priori* del campo di velocità. In particolari situazioni il modello del campo di velocità non è noto con la dovuta accuratezza e solo dall'interpretazione dei dati migrati si può effettuare la verifica del modello. Un modo di impostare la soluzione di un simile paradosso, per il quale la velocità è contemporaneamente dato d'ingresso ed incognita del problema, è quello di considerarlo come un problema di ottimizzazione: la soluzione è quel campo di velocità, le cui discontinuità approssimano meglio la geometria delle superfici riflettenti. A causa del costo computazionale del problema legato alla inversione matematica dei dati geofisici, la ricerca di una soluzione, in ambito industriale, viene realizzata in modo semi-automatico attraverso cicli di migrazione e fasi di correzione del campo di velocità, determinate da esperti interpretatori. Data la mole di dati ² che devono essere elaborati nel contesto delle applicazioni reali, sono necessarie metodiche e macchine in grado di trattare processi di tale vastità in tempi ragionevoli. Queste problematiche possono essere affrontate adeguatamente solo nell'ambito del calcolo parallelo. Nel seguito verranno indicate varie tecniche di soluzione del problema di migrazione dei dati sismici operanti su dati preventivamente *compressi* che soddisfano i modelli *zero-offset* e dei *riflettori esplodenti*. Verranno esposte le caratteristiche generali di alcuni algoritmi di migrazione dei dati sismici per la migrazione in profondità (*depth migration*) tramite l'equazione parassiale e tramite l'equazione d'onda completa. Verrà introdotta la migrazione di dati *post-stack* nel semplice caso teorico di velocità costante su tutto il dominio di calcolo e verrà indicata l'estensione a velocità variabili operante su un modello di Terra stratificato e nel caso più generale di velocità variabile *tout court*.

I metodi trattati non costituiscono la totalità di quelli proposti in letteratura, ma solo una parte, seppure rilevante, di quelli che hanno avuto riscontri positivi nell'ambito delle applicazioni industriali, prevalentemente

²Ciascuno dei ricevitori, un sismografo o un geofono, produce una traccia composta tipicamente di 1500 intervalli di tempo di 4 *msec.* ciascuno. Il comportamento temporale di ogni singola traccia dipende in modo complesso dalla disuniformità della crosta terrestre. La raccolta, nei casi reali è costituita da oltre 2000 di queste tracce (100 Km^2 di area producono oltre 10^5 campioni per un esperimento).

nell'industria petrolifera. Nella loro selezione si sono privilegiate quelle caratteristiche, come l'accuratezza, il costo computazionale e la possibilità di realizzare delle implementazioni, sia bidimensionali che tridimensionali, di non elevata complessità che permettono efficienti applicazioni su *dataset* reali. Tuttavia esistono trattazioni di grande interesse, che si discostano da questo modo di pensare e che affrontano problematiche teoriche utili per una più approfondita e completa comprensione dei fenomeni legati alla propagazione delle onde elastiche, che non verranno trattati nell'ambito di questo lavoro.

1.1 Migrazione in profondità

Consideriamo la crosta terrestre come uno spazio tridimensionale, con l'asse z in direzione verticale e crescente verso il basso. L'onda di pressione si assume viaggiante a partire dalla sorgente (figura (1.1)) posizionata nel punto $\mathbf{r}_s = (x_s, y_s, z_s) \in Volume$ con velocità pari a $v(\mathbf{r})$. Se il mezzo è omogeneo allora non vi sono riflessioni e l'onda viene trasmessa completamente, mentre in mezzi non omogenei, nei quali $v(\mathbf{r})$ varia spazialmente, le onde vengono in parte trasmesse in parte riflesse. Ogni punto può essere caratterizzato da un coefficiente di riflessione $\Gamma(\mathbf{r})$ ³, del quale si trascurano le variazioni angolari. Consideriamo la pressione $P(\mathbf{r}_g, \mathbf{r}_s; t)$ misurata

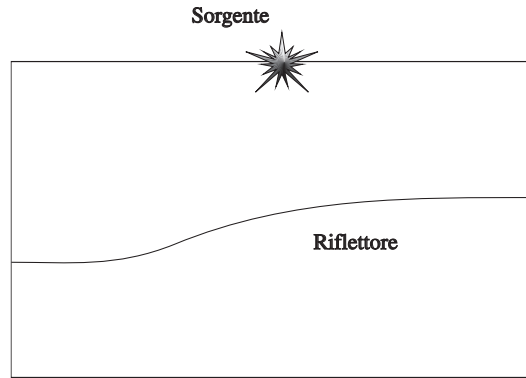


Figura 1.1: Sorgente e riflettore.

al tempo t da un ricevitore (geofono) in posizione $\mathbf{r}_g = (x_g, y_g, z_g)$ dopo che, all'istante $t = 0$, la sorgente posta in $\mathbf{r}_s = (x_s, y_s, z_s)$ è stata inizializzata. Fissata una sorgente, il campo di pressione misurato nel punto \mathbf{r}_g è soluzione dell'equazione:

$$\frac{\partial^2 P}{\partial x_g^2} + \frac{\partial^2 P}{\partial y_g^2} + \frac{\partial^2 P}{\partial z_g^2} - \frac{1}{v(x_g, y_g, z_g)^2} \cdot \frac{\partial^2 P}{\partial t^2} = 0 \quad (1.1)$$

³Nel caso particolare in cui il volume considerato sia diviso in due sottospazi con velocità $V_1 \gg V_2$, le onde viaggianti (a partire dal primo semispazio) alla velocità V_1 verranno per lo più riflesse dalla discontinuità e solo in piccola parte trasmesse.

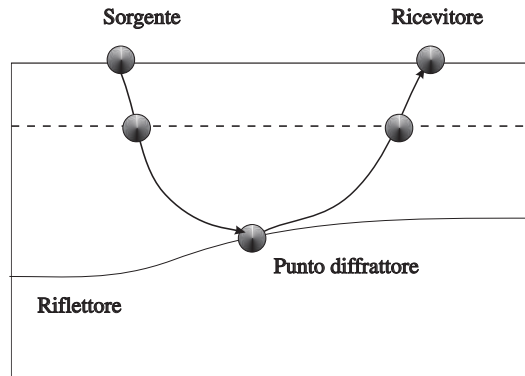


Figura 1.2: Sorgente e riflettore collasano su un punto diffrattore.

Fissato un ricevitore, come conseguenza del *principio di reciprocità*⁴, il campo di pressione prodotto spostando la sorgente dal punto r_s al punto $r_g = (x_g, y_g, z_g)$ e viceversa (il ricevitore dal punto r_g al punto di coordinate r_s) soddisfa l'equazione scalare delle onde:

$$\frac{\partial^2 P}{\partial x_s^2} + \frac{\partial^2 P}{\partial y_s^2} + \frac{\partial^2 P}{\partial z_s^2} - \frac{1}{v(x_s, y_s, z_s)^2} \cdot \frac{\partial^2 P}{\partial t^2} = 0 \quad (1.2)$$

Si tratta di trovare il campo d'onda $P(r_g, r_s; t)$ che soddisfa contemporaneamente le equazioni (1.1) e (1.2). Nel caso di più sorgenti e più ricevitori si ricorre al *principio di sovrapposizione degli effetti* e le due equazioni precedenti dovranno essere soddisfatte per ogni coppia sorgente ricevitore.

La migrazione, in questo contesto, può essere vista come il processo di correzione del campo di velocità attraverso il quale può ricostruirsi una mappa della riflettività locale $\Gamma(x, y, z)$ a partire dall'unica informazione disponibile rappresentata dalla *sezione sismica* $P(x_s, y_s, z_s = 0, x_g, y_g, z_g = 0; t)$ e dal campo di velocità $v(x, y, z)$ iniziale.

Tramite opportune condizioni al contorno possono risolversi le equazioni (1.1) e (1.2) per $P(r_s, r_g, t)$; i dati grezzi (precedenti lo *stacking*) che costituiscono la sezione sismica, possono essere migrati, avanzando lungo l'asse z e fornendo per ciascuna coppia sorgente-ricevitore il valore della pressione in funzione del tempo t di attraversamento del mezzo. Spostando sorgente e ricevitore verso il basso lungo il percorso del raggio (figura (1.2)), avremo

⁴Scambiando sorgente e ricevitore il campo d'onda osservato deve rimanere invariato.

che sorgente e ricevitore collassano in un punto $\mathbf{r}=\mathbf{r}_s=\mathbf{r}_g$ che rappresenta la posizione del riflettore a $t = 0$; la pressione così misurata viene considerata proporzionale al coefficiente di riflettività:

$$\Gamma(\mathbf{r}) \sim P(\mathbf{r}_s = \mathbf{r}, \mathbf{r}_g = \mathbf{r}, t = 0) \quad (1.3)$$

Il campo di pressione $P(\mathbf{r}_s = \mathbf{r}, \mathbf{r}_g = \mathbf{r}, t = 0)$ è la *sezione sismica migrata*. In questo modo la migrazione dei dati grezzi (*migrazione pre-stack*) potenzialmente è in grado di fornire una realistica ricostruzione acustica della crosta terrestre.

1.1.1 Zero-offset e riflettori esplodenti

Semplifichiamo la trattazione ponendoci su una sezione $y = y_0$ riducendo il problema, e quindi l'esperimento sismico, al caso bidimensionale $\mathbf{r} = (x, z)$. Il *modello zero-offset* introduce un nuovo sistema di riferimento che permette di semplificare notevolmente il modello fisico-matematico adottato per la soluzione dell'equazione d'onda (in particolare delle equazioni (1.1) e (1.2)).

Definiamo il punto di mezzo tra ricevitore e sorgente:

$$X = \frac{(x_g + x_s)}{2}, \quad Z = \frac{(z_g + z_s)}{2}; \quad (1.4)$$

e le coordinate relative o di *offset*:

$$x = \frac{(x_g - x_s)}{2}, \quad z = \frac{(z_g - z_s)}{2}. \quad (1.5)$$

Nel sistema di coordinate relativo al punto di mezzo e all'*offset*, avremo

$$P(X, x, Z, z, t) = \mathcal{P}(x_s, z_s, x_g, z_g, t), \quad (1.6)$$

e le equazioni (1.1) e (1.2) verranno trasformate nelle seguenti:

$$\frac{\partial^2 P}{\partial X^2} + \frac{\partial^2 P}{\partial Z^2} + \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial z^2} - \frac{4}{v(X+x, Z+z)^2} \cdot \frac{\partial^2 P}{\partial t^2} + 2 \cdot \left(\frac{\partial^2 P}{\partial X \partial x} + \frac{\partial^2 P}{\partial Z \partial z} \right) = 0 \quad (1.7)$$

$$\frac{\partial^2 P}{\partial X^2} + \frac{\partial^2 P}{\partial Z^2} + \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial z^2} - \frac{4}{v(X-x, Z-z)^2} \cdot \frac{\partial^2 P}{\partial t^2} - 2 \cdot \left(\frac{\partial^2 P}{\partial X \partial x} + \frac{\partial^2 P}{\partial Z \partial z} \right) = 0. \quad (1.8)$$

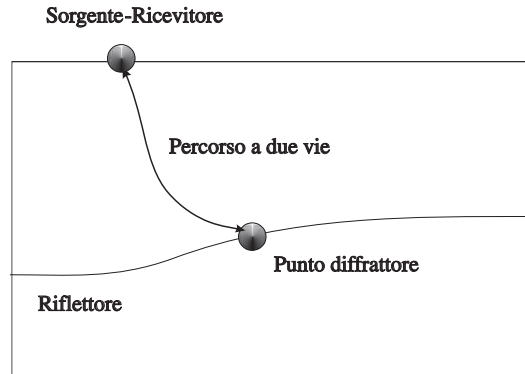


Figura 1.3: Punto di diffrazione in profondità.

Come conseguenza del *principio di reciprocità*, osserviamo che $P(x_s, z_s, x_g, z_g) = P(x_g, z_g, x_s, z_s)$, quindi nel nuovo sistema di coordinate il campo P è una funzione pari dell'*offset*:

$$P(X, x, Z, z, t) = P(X, -x, Z, -z, t). \quad (1.9)$$

Questo risultato ha immediate implicazioni sulle derivate prime rispetto all'*offset*:

$$\frac{\partial P}{\partial x}(X, 0, Z, 0, t) = \frac{\partial P}{\partial z}(X, 0, Z, 0, t) = 0; \quad (1.10)$$

e sulle derivate miste

$$\frac{\partial^2 P}{\partial X \partial x}(X, 0, Z, 0, t) = \frac{\partial^2 P}{\partial Z \partial z}(X, 0, Z, 0, t) = 0. \quad (1.11)$$

L'equazione (1.11) suggerisce come modificare le condizioni iniziali del modello a costo di qualche approssimazione. Mediante l'approssimazione derivante dal *modello zero-offset* dei dati sismici, possiamo scrivere $P(X, Z, t) = P(X, 0, Z, 0, t)$, e trascurando le derivate seconde rispetto a x e z , le equazioni (1.7) e (1.8) si riducono ad un'unica equazione scalare delle onde:

$$\frac{\partial^2 P}{\partial X^2} + \frac{\partial^2 P}{\partial Z^2} - \frac{4}{v(X, Z)^2} \cdot \frac{\partial^2 P}{\partial t^2} = 0. \quad (1.12)$$

Questa semplificazione deriva dall'aver supposto che, al tendere a zero dell'*offset*, le derivate seconde lungo x e z inducono variazioni trascurabili nell'ampiezza di P . È interessante notare che nella rappresentazione *zero-offset*, sorgente e ricevitore coincidono (figura (1.3)); in questo modo il tempo $t/2$ impiegato dal raggio, a partire dalla sorgente inizializzata al

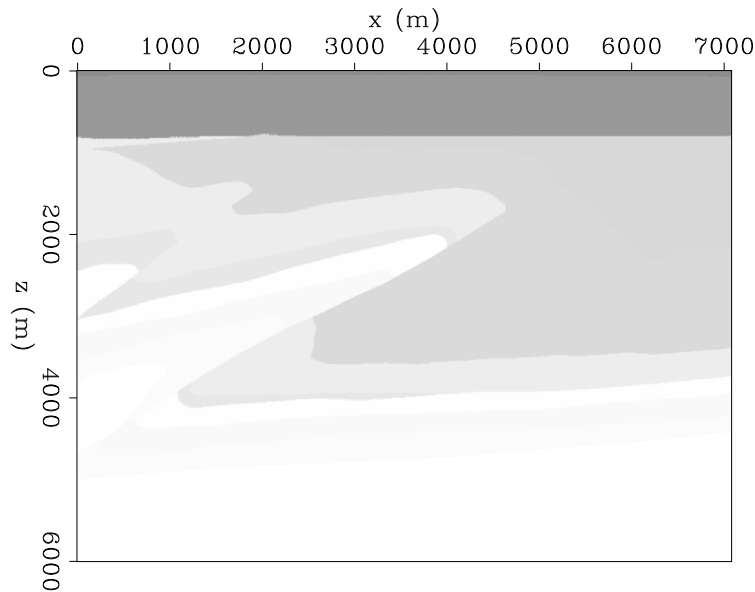


Figura 1.4: Esempio di campo di velocità bidimensionale (AGIP).

tempo $t = 0$, a scendere in profondità è esattamente uguale al tempo $t/2$ impiegato a salire in superficie.

Il campo di pressione sul riflettore sarà proporzionale al coefficiente di riflessione $\Gamma(\mathbf{r})$ (1.3). Come conseguenza, inizializzando una sorgente al tempo t sul riflettore, con una sorgente di intensità pari al coefficiente di riflessione, risulterà che il segnale compie lo stesso percorso in salita, producendo un'identica traccia in superficie. Tale propagazione è descritta dall'equazione (1.12) considerando il tempo di propagazione dimezzato o in modo equivalente la velocità di propagazione dimezzata. Questo è, in sintesi, l'effetto del *modello dei riflettori esplodenti*.

In definitiva, utilizzando come dati d'ingresso i dati registrati secondo il modello *zero-offset* $P(X, 0, t)$ e risolvendo l'equazione (1.12) all'indietro nei tempi (*Reverse Time Migration*), al tempo t otterremo un'immagine acustica dei riflettori in profondità.

1.1.2 Stacking

Nei casi pratici sono presenti chilometri di *offset* tra le coppie sorgente-ricevitore; tipicamente vengono utilizzate tecniche di pre-elaborazione, che

permettono lo *stacking* dei dati, per approssimare adeguatamente l'ipotesi di *zero-offset*. Fissato un modello di velocità (figura ⁵ (1.4)), le tracce non ancora approssimate *zero-offset* $P(x_s, y_s, z_s = 0, x_g, y_g, z_g = 0, t)$, aventi stesso punto di mezzo X , verranno modificate mediante una trasformazione iperbolica di coordinate con le cosiddette correzioni *Normal Move-Out* o *Dip Move-Out*. Quindi verranno mediate attorno al punto di mezzo per dare un'approssimazione *zero-offset* $P(X, 0, t)$ adeguata.

Il modello di velocità influenza notevolmente la qualità dei risultati della migrazione e l'ottimizzazione della sua valutazione è l'obiettivo principale nel miglioramento degli algoritmi di migrazione di dati sismici. Inoltre, è importante sottolineare che la sezione sismica i cui dati sono stati compressi attraverso il processo di *stacking* non coincide con i dati grezzi (*raw data*) osservati e quindi il processo di migrazione che li utilizza non ricostruirà il campo d'onda osservato ed i riflettori corrispondenti, se non in modo approssimato.

1.1.3 Estrapolazione in profondità

L'equazione scalare delle onde (1.12) può risciversi nel modo usuale:

$$\frac{\partial^2 P}{\partial X^2} + \frac{\partial^2 P}{\partial Z^2} - \frac{1}{u(X, Z)^2} \cdot \frac{\partial^2 P}{\partial t^2} = 0, \quad (1.13)$$

avendo cura di ricordare che $u = \frac{v}{2}$ è la mezza velocità dell'onda, X è la variabile orizzontale indicante il punto di mezzo e t il tempo del percorso andata-ritorno. Si può scegliere di estrapolare in profondità, in tale caso la variabile Z , la profondità, cresce con l'avanzare dell'estrapolazione verso il basso a partire dalla sezione sismica $P(X, 0, t)$. L'obiettivo finale è la determinazione della sezione sismica migrata $P(X, Z, 0)$.

Per risolvere l'equazione (1.12) è necessario disporre di condizioni al contorno; un modo è quello di porre condizioni periodiche al contorno lungo X e t che possono essere sfruttate efficacemente nel dominio di *Fourier*.

⁵Il campo di velocità mostrato in figura è quello che è stato utilizzato per migrare la sezione zero-offset mostrata in figura (??) ottenendo come risultato l'immagine del sottosuolo mostrata nella figura (??).

1.2 Migrazione a velocità costante nel dominio di Fourier

Dal punto di vista concettuale la migrazione di dati *poststack* risulta più semplice dell'*imaging prestack* dei dati. Le ragioni principali sono essenzialmente due:

- si deve tenere conto di un unico sistema di coordinate;
- il volume di dati da elaborare risulta fortemente ridotto e può essere gestito con maggiore facilità.

Come abbiamo visto la compressione dei dati è ottenuta rispetto ad un riferimento *zero-offset*, ed in accordo al *modello a riflettori esplodenti*⁶.

I dati registrati in superficie $STACK(x, y, t)$ a $z = 0$ non sono altro che il risultato del campo che si propaga verso l'alto:

$$STACK(x, y, t) = P(x, y, z = 0, t); \quad (1.14)$$

L'ampiezza del campo di pressione $P(\mathbf{r}, t)$ varia localmente, aumentando e diminuendo al variare della velocità $V(x, y, z)$ dei punti del sottosuolo. La migrazione risulta essere una semplice mappatura dei dati rispetto alla riflettività del sottosuolo:

$$STACK(x, y, t) \longrightarrow \Gamma(x, y, z). \quad (1.15)$$

Più in particolare possiamo distinguere una mappatura che da $P(x, y, z, t)$ deriva il campo in profondità $P(x, y, z + \Delta z, t)$, che chiameremo *estrapolazione*, ed una mappatura che ricava il valore $P(x, y, z, t = 0)$ del campo di pressione ricollocando l'eco dei riflettori in profondità), che chiameremo *imaging*. L'estrapolazione può essere realizzata perchè regolata dall'equazione (1.12). Tale equazione, unita alla conoscenza del campo $P(\mathbf{r}, t)$ propagato in superficie, permette di ricavare tutte le informazioni necessarie per attuare l'estrapolazione di $P(\mathbf{r}, t)$ su tutto il volume. Sia nel caso del problema diretto che nel caso del problema inverso la trattazione può essere

⁶Lowental et al., 1976.

condotta nel dominio delle frequenze. Se la velocità è costante, il problema si semplifica notevolmente ed è caratterizzato dall'aver come soluzione una sovrapposizione di onde piane del tipo:

$$e^{j(k_x x + k_y y + k_z z \pm \omega t)}$$

La direzione del piano d'onda è determinata dal segno del versore $\frac{\mathbf{k}}{|\mathbf{k}|}$; l'estrapolazione rispetto a z del piano d'onda risulta essere costituita unicamente da uno sfasamento.

1.2.1 Traslazione di fase

In forma discreta possiamo riferirci alla forma seguente:

$$P(x, y, z, t) = \sum_{k_x} \sum_{k_y} \sum_{\omega} \hat{P}(k_x, k_y, z, \omega) \cdot e^{j(k_x x + k_y y + \omega t)} \quad (1.16)$$

cioè alla anti-trasformata discreta bidimensionale di *Fourier* nei numeri d'onda. Effettuando le opportune sostituzioni nell'equazione d'onda (1.12) otteniamo:

$$\frac{\partial^2 \hat{P}(k_x, z, k_y, \omega)}{\partial z^2} = - \left(\frac{2\omega}{v} \right)^2 \cdot \left[1 - \left(\frac{v}{2\omega} \right)^2 (k_x^2 + k_y^2) \right] \cdot \hat{P}(k_x, k_y, z, \omega) \quad (1.17)$$

Quest'equazione ha una soluzione generale nella forma:

$$\hat{P}(k_x, k_y, z, \omega) = A^+ e^{jk_z z} + A^- e^{-jk_z z} \quad (1.18)$$

con

$$k_z = \frac{\omega}{v} \sqrt{1 - \left(\frac{v\kappa}{\omega} \right)^2}, \quad \kappa = \sqrt{k_x^2 + k_y^2}. \quad (1.19)$$

In genere le due costanti A^+ e A^- vengono ricavate dalle condizioni iniziali. Fissando il segno di ω possiamo distinguere la componente in salita (z decrescente) da quella in discesa (z crescente), rispetto all'asse z di riferimento:

$$\begin{aligned} \hat{P}^+(\kappa, z, \omega) &= A^+ e^{jk_z z} \\ \hat{P}^-(\kappa, z, \omega) &= A^- e^{-jk_z z} \end{aligned} \quad (1.20)$$

Ponendo $A^- = 0$, cioè considerando la sola componente discendente, ci poniamo nelle condizioni di *estrapolazione in profondità*, all'indietro nei tempi:

$$\widehat{P}(k_x, k_y, z, \omega) = \widehat{P}(k_x, k_y, z = 0, \omega) \cdot e^{jk_z z}. \quad (1.21)$$

Il campo $\widehat{P}(k_x, k_y, z = 0, \omega)$ rappresenta la trasformata della sezione sismica nel dominio (k_x, ω) . Ponendo A^- a zero nella soluzione generale, la derivata prima della sezione sismica rispetto a zeta è definita univocamente e vale:

$$\frac{\partial \widehat{P}}{\partial z}(k_x, k_y, z, \omega) = jk_z \cdot \widehat{P}(k_x, k_y, z = 0, \omega) \cdot e^{jk_z z}. \quad (1.22)$$

Sostituendo l'equazione (1.22) nella (1.16) si ottiene:

$$P(x, y, z, t = 0) = \sum_{k_x} \sum_{k_y} \sum_{\omega \geq 0} \widehat{P}(k_x, k_y, z = 0, \omega) \cdot e^{j(k_z z + k_x x + k_y y)} \quad (1.23)$$

che rappresenta la soluzione discreta del problema per $v = \text{costante}$. Poiché siamo interessati all'eliminazione delle onde evanescenti, k_x , k_y ed ω devono essere ristretti al *range* di valori che permette di avere $k_z \in \mathcal{R}$:

$$\frac{|k|}{|\omega|} > v. \quad (1.24)$$

Riscrivendo la (1.23) in forma ricorsiva, legando la soluzione alla profondità z con quella alla profondità $z + \Delta z$, avremo:

$$P(x, y, z + \Delta z, 0) = \sum_{k_x} \sum_{k_y} \sum_{\omega} P(k_x, k_y, z, \omega) \cdot e^{j(k_z \Delta z + k_x x + k_y y)} \quad (1.25)$$

il cui significato può essere meglio compreso raggruppando in termini della forma:

$$P(x, y, z + \Delta z, 0) = \sum_{k_x} \sum_{k_y} \left[\sum_{\omega} P(k_x, k_y, z, \omega) \cdot e^{jk_z \Delta z} \right] e^{j(k_x x + k_y y)}, \quad (1.26)$$

l'espressione tra parentesi è la sommatoria su ω , con $|\omega| > v \cdot \sqrt{k_x^2 + k_y^2}$ per ciascun incremento Δz della profondità. In questo modo dobbiamo fare una sola anti-trasformata di *Fourier* per ricostruire la sezione sismica migrata. L'equazione (1.26) sta alla base del metodo di migrazione numerica di *Phase Shift*; essa permette l'estrapolazione in profondità dei dati sismici in mezzi stratificati, nei quali il campo di velocità è una funzione

di zeta, costante a tratti e variabile da strato a strato. Essendo un metodo spettrale, risulta chiaro che la formula del *Phase Shift* non è valida per velocità variabili, seppure lentamente. La difficoltà di questo approccio sta nel fatto che conduce alla soluzione di un sistema denso ⁷ di equazioni differenziali ordinarie per i coefficienti di *Fourier* $\hat{P}(k_z, k_y, z, \omega)$ per ogni z . Come mostreremo nel seguito, per risolvere questo problema si preferisce semplificare il modello di migrazione riducendo la sua complessità computazionale attraverso l'equazione *one-way* nel dominio di *Fourier* tramite *frazioni continue*.

1.2.2 Migrazione f-k

A partire dall'equazione (1.16), attraverso un cambiamento di variabile di integrazione tra ω e k_z , può ottenersi un metodo efficace di migrazione confrontabile in prestazioni con la migrazione mediante traslazione di fase. Utilizzando le equazioni (1.16) e (1.21), riscritte nel continuo, abbiamo l'equazione:

$$P(x, y, z, t) = \frac{1}{(2\pi)^3} \int dk_x \int dk_y \int_0^\infty d\omega \cdot \hat{P}(k_x, k_y, z = 0, \omega) \cdot e^{j \cdot (k_z z + \omega t + k_x x + k_y y)}, \quad (1.27)$$

che rappresenta una sovrapposizione di onde piane che si propagano verso z crescenti, indietro nel tempo. Tale equazione include onde evanescenti ⁸, in quanto k_x , k_y ed ω possono variare con continuità sul dominio d'interesse. Per $t = 0$ otteniamo la condizione di *imaging*:

$$P(x, y, z, t = 0) = \frac{1}{(2\pi)^3} \cdot \int dk_x \int dk_y \int_0^\infty d\omega \cdot \hat{P}(k_x, k_y, t = 0, \omega) \cdot e^{j(k_z z + k_x x + k_y y)}. \quad (1.28)$$

⁷Le cui equazioni hanno coefficienti per lo più non nulli.

⁸Per le onde evanescenti vale $k_x^2 + k_y^2 > \frac{4\omega^2}{v^2}$ e k_z diviene immaginario puro. Fisicamente, le onde evanescenti sono delle variazioni spaziali (in x ed y) troppo rapide per propagarsi alla frequenza ω ; tali variazioni decadono esponenzialmente nel loro percorso. Nelle applicazioni pratiche, le onde evanescenti risultano fastidiose e perciò ci si sforza di eliminarle perchè influenzano la stabilità numerica della soluzione.

dove $\widehat{P}(k_x, k_y, t = 0, \omega) = STACK(k_x, k_y, \omega)$ rappresenta la sezione sismica da migrare. Ricordando che $\Gamma(x, y, z) = P(x, y, z, t = 0)$, avremo:

$$\Gamma(x, y, z) = \frac{1}{(2\pi)^3} \cdot \int dk_x \int dk_y \int_0^\infty d\omega \cdot STACK(k_x, k_y, \omega) \cdot e^{j(k_z z + k_x x + k_y y)}, \quad (1.29)$$

che consiste di una anti-trasformata di *Fourier* bidimensionale nelle frequenze spaziali ($\kappa = (k_x, k_y)$) e in una somma sulle frequenze temporali.

Essa rappresenta la forma continua della migrazione mediante traslazione di fase⁹. Risolvendo l'equazione (1.19) rispetto ad ω , avremo:

$$\omega = -\frac{k_z \cdot v}{2} \cdot \sqrt{1 + \frac{(k_x^2 + k_y^2)}{k_z^2}}. \quad (1.30)$$

Per effettuare il cambiamento di variabile abbiamo bisogno della derivata

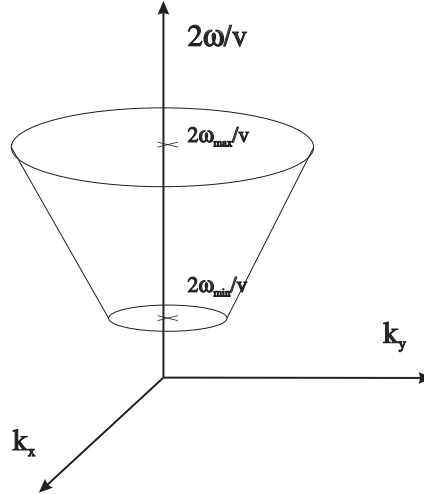


Figura 1.5: Tronco di cono dei vettori d'onda.

di ω rispetto a $k_z > 0$:

$$\frac{d\omega}{dk_z} = -\frac{k_z \cdot v^2}{4 \cdot \omega}. \quad (1.31)$$

Utilizzando queste relazioni l'equazione (1.29) diviene

$$\Gamma(x, y, z) = -\frac{v^2}{4 \cdot (2\pi)^3} \int dk_x \int dk_y \int dk_z \cdot \left(\frac{k_z}{\omega}\right) STACK(k_x, k_y, \omega) \cdot e^{j \cdot (k_x \cdot x + k_y \cdot y + k_z \cdot z)}. \quad (1.32)$$

Quest'ultima può essere indicata come la trasformata inversa di *Fourier* di $STACK(k_x, k_y, \omega) \cdot \left(\frac{k_z}{\omega}\right)$ con ω sostituito dall'espressione (1.30). L'intero processo di migrazione può essere sintetizzato nel modo seguente:

⁹Gazdag, 1978.

- trasformata di *Fourier* dei dati *STACK* nel dominio della frequenza e dei numeri d'onda $f-k$;
- cambiamento di variabile da ω a k_z , e moltiplicazione per il fattore $-\frac{v^2 \cdot k_z}{4 \cdot \omega}$;
- trasformata inversa di *Fourier* indietro nel dominio spazio-temporale $t-r$.

Un metodo di migrazione di questo tipo viene indicata come migrazione $f-k$ [14]. Nel dominio $f-k$:

$$\hat{\Gamma}(k_x, k_y, k_z) = -v^2 \cdot \frac{k_z}{4 \cdot \omega} \cdot STACK(k_x, k_y, \omega). \quad (1.33)$$

Le tre trasformate dirette di *Fourier* attuate nella (1.33) permettono di espandere i dati appartenenti all'onda piana in componenti, ciascuna delle quali con specifiche frequenze k_x , k_y e ω (figura (1.5)). Per ogni valore della terna (k_x, k_y, ω) , k_z risulta determinato.

Interpretazione geometrica della migrazione f-k

Il cambiamento di variabile che è stato attuato tra ω e k_z produce l'effetto di incrementare la pendenza apparente delle strutture riflettenti in profondità dopo la migrazione rispetto a quelle relative alla sezione sismica. Prima della migrazione, la pendenza apparente viene definita attraverso l'angolo θ_{app} nel modo seguente (con riferimento alla figura (1.6)):

$$tg(\theta_{app}) = \frac{v}{2|\omega|} \cdot \sqrt{k_x^2 + k_y^2}. \quad (1.34)$$

Dopo la migrazione $f-k$ dei dati, a seguito del cambiamento di variabile, la pendenza apparente diviene:

$$tg(\theta_{mig}) = \frac{\sqrt{k_x^2 + k_y^2}}{|k_z|}, \quad (1.35)$$

come mostrato in figura (1.7).

Possiamo mettere in relazione gli angoli θ_{app} e θ_{mig} rispettivamente prima e dopo la migrazione, con considerazioni che coinvolgono la direzione del

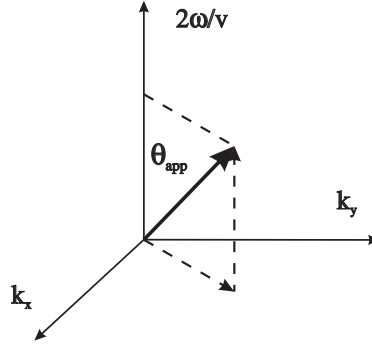


Figura 1.6: Pendenza apparente, prima della migrazione.

vettore d'onda \mathbf{k} . Con riferimento alla figura (1.7), ricordando che $|\mathbf{k}| = \sqrt{k_x^2 + k_y^2 + k_z^2}$, scriveremo:

$$\sqrt{k_x^2 + k_y^2} = |\mathbf{k}| \cdot \sin(\theta_{mig}). \quad (1.36)$$

col vincolo $\kappa^2 = k_x^2 + k_y^2 = \frac{2\omega}{v} - k_z^2$, avremo

$$\sin(\theta_{mig}) = \frac{v}{2|\omega|} \sqrt{\left(\frac{2\omega}{v}\right)^2 - k_z^2} = \text{tg}(\theta_{app}) \quad (1.37)$$

Ciò equivale a dire che la migrazione $f-k$ inasprisce la “pendenza dei dati” (riflettori) in profondità. L'angolo θ_{mig} mette in relazione ω e k_z nel modo

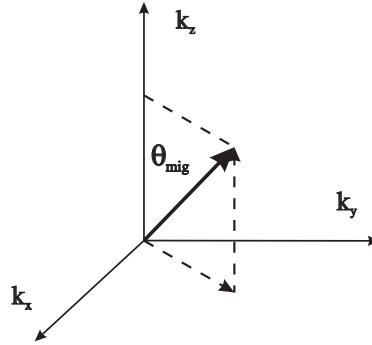


Figura 1.7: Pendenza apparente, dopo la migrazione.

seguinte:

$$\omega = -\frac{k_z v}{2} \cdot \sec(\theta_{mig}); \quad (1.38)$$

tramite quest'equazione possiamo interpretare la migrazione come una conversione *tempo-profondità* utilizzando una velocità apparente lungo z

dipendente dalla profondità attraverso l'angolo θ_{mig} :

$$V_z = \frac{v}{\cos(\theta_{mig})}. \quad (1.39)$$

Esigenze computazionali per la migrazione f-k e PS

Sia la migrazione *PS* (*Phase Shift*) che quella *f-k* utilizzano la trasformata (diretta) di *Fourier*. Per trasformate discrete di dimensione N_x , N_y ed N_t rispettivamente, il carico computazionale risulta proporzionale a:

$$N_1 = N_x N_y N_t \cdot \text{Log}(N_x N_y N_t). \quad (1.40)$$

In particolare l'operatore di *Phase Shift* $e^{j(k_z \cdot z)}$ deve essere costruito per ogni k_x , k_y ed ω ; ciò richiede un costo computazionale proporzionalità rispetto a $N_x N_y N_t$. Per ogni valore di z , dovranno esserci moltiplicazioni e somme complesse rispetto ad ω e questo necessita di proporzionalità rispetto a:

$$N_2 = N_x N_y N_t N_z. \quad (1.41)$$

Generalmente questo è il contributo dominante. L'antitrasformata che deve essere attuata dopo la somma sugli ω contribuisce proporzionalmente a $N_x N_y N_z \cdot \text{Log}(N_x N_y N_z)$. Normalmente tra questi contributi quello proporzionale a N_1 risulta dominante. Per la migrazione *f-k* il cambiamento di variabile e la moltiplicazione per $\frac{k_z}{\omega}$ richiede un numero di operazioni in virgola mobile proporzionale a $N_x N_y N_z$. La trasformata inversa di *Fourier* può considerarsi computazionalmente equivalente alla trasformata diretta. Il peso delle trasformate oppure del cambiamento di variabile può essere equivalente o essere a vantaggio di un metodo rispetto all'altro in dipendenza dei valori N_x , N_y , N_z , ed N_t .

L'angolo di migrazione

Il numero d'onda k_z può esprimersi nella forma seguente (nel semplice caso bidimensionale (k_x, k_z)):

$$k_z = \pm \sqrt{\frac{\omega^2}{v^2} - k_x^2} = \frac{\omega}{v} \sqrt{1 - \frac{v^2 k_x^2}{\omega^2}} = \frac{\omega}{v} \cdot \cos(\theta) \quad (1.42)$$

avendo posto:

$$\begin{cases} \sin(\theta) = \frac{v \cdot k_x}{\omega} \\ \cos(\theta) = \frac{v \cdot k_z}{\omega} \end{cases} \quad (1.43)$$

con θ angolo di migrazione. Le approssimazioni della radice, che rappresenta l'angolo di migrazione attraverso la funzione coseno, determinano angoli di propagazione minori dell'angolo piatto in relazione al numero di termini considerato.

In un campo di velocità costante un raggio, individuante la propagazione in una certa direzione inclinata di un angolo θ rispetto alla verticale, risulterà perpendicolare al fronte d'onda piano (figura (1.8)); un campo di velocità variabile determina la curvatura del raggio (figura (1.9)) che conseguentemente non risulterà normale al fronte d'onda. Le approssimazioni del termine cosinusoidale producono un effetto di curvatura del fronte d'onda, immerso in un campo di velocità costante, che da circolare (o sferico nel caso tridimensionale) diviene ellittico o cardioideo, dipendentemente dall'entità dell'approssimazione attuata (figura (1.10)).

La correttezza dell'approssimazione attuata dipenderà da quanto l'operatore approssimante si avvicinerà alla forma esatta, ossia da quanto il fronte d'onda approssimato riesce a sovrapporsi a quello reale; l'angolo, rispetto alla verticale, entro il quale i fronti d'onda si sovrappongono è l'angolo di propagazione (o di migrazione). In altri termini, ciò significa che l'approssimazione del termine $\cos\theta = \sqrt{1 - v^2 k_x^2 / \omega^2}$ determina l'entità dell'angolo massimo entro il quale la propagazione viene approssimata correttamente.

Nel seguito indicheremo il termine sotto radice come $\mathcal{R} = \sqrt{1 - K^2}$ con $K = \frac{v \cdot k_x}{\omega}$. Esistono svariati metodi di espansione per approssimare il termine cosinusoidale, in sismica si fa riferimento all'*espansione in frazioni*

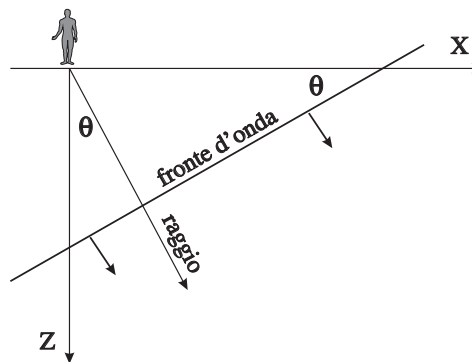


Figura 1.8: Fronte d'onda piano e raggio in un campo di velocità costante.

continue (o espansione di Muir). Con riferimento alla Tabella (1.1) viene indicato l'angolo entro il quale l'approssimazione risulta valida. È chiaro che

θ	$R = \sqrt{1 - K^2} = \cos(\theta)$
5°	$R_0 = 1$
15°	$R_1 = 1 - \frac{K^2}{2}$
45°	$R_2 = 1 - \frac{K^2}{2 - \frac{K^2}{2}}$
60°	$R_3 = 1 - \frac{K^2}{2 - \frac{K^2}{2 - \frac{K^2}{2}}}$

Tabella 1.1: Espansione in frazioni continue.

la perdita d'informazione è una caratteristica intrinseca dell'approssimazione, ciò limita l'angolo di migrazione e l'informazione dei riflettori molto inclinati viene perduta.

Dispersione per anisotropia

La *dispersione in frequenza* si manifesta quando onde di differente frequenza si propagano a differenti velocità nello stesso mezzo. L'effetto che si riscontra, mediante ispezione visuale di una sezione sismica migrata, è lo sparpagliamento del pacchetto d'onda con evidenti disuniformità nell'immagine. Migliorando l'accuratezza dell'approssimazione dell'equazione differenziale delle onde è possibile ridurre gli effetti della dispersione in fre-

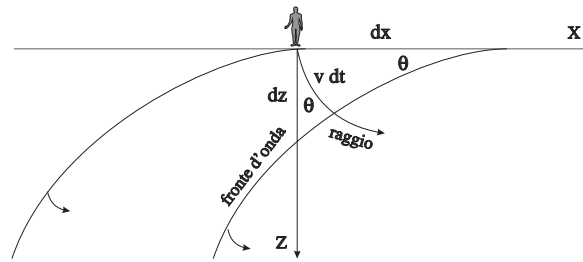


Figura 1.9: Fronte d'onda e raggio in un campo di velocità variabile verticalmente.

quenza [11].

La dispersione in frequenza è una delle più importanti cause d'errore riscontrabili nei processi di migrazione di dati sismici. Un'altro tipo di dispersione è la cosiddetta *dispersione per anisotropia*. Onde propagantisi in diverse direzioni si muovono con velocità differenti. Le cause della disper-

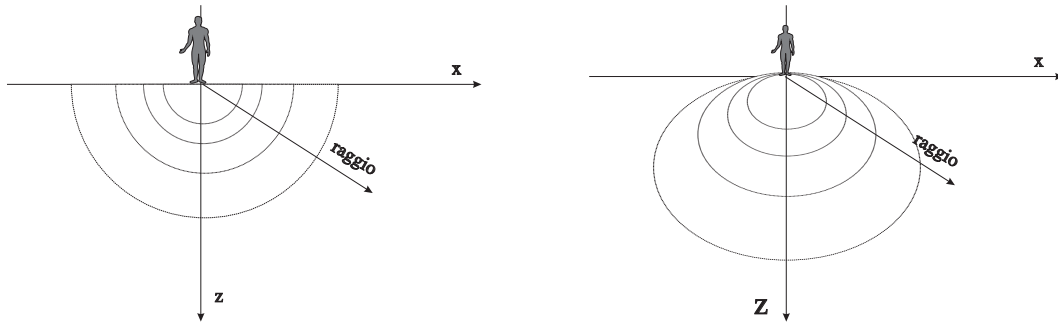


Figura 1.10: Fronti d'onda circolari, ellittici.

sione anisotropica sono legate al troncamento nell'approssimazione della relazione di dispersione. Tali espansioni in genere agiscono su un numero finito di termini, a causa del costo computazionale dell'intero processo di elaborazione. La peculiarità dell'anisotropia sta nel fatto che i raggi non sono perpendicolari al fronte d'onda (come mostrato in figura (1.10) nel confronto tra caso esatto e approssimazione a 15°), e i fronti d'onda in mezzi omogenei non sono più fronti sferici.

La forma del fronte d'onda delle cosiddette *sorgenti secondarie di Huygens* è un semicerchio nel caso esatto, un'ellisse nell'approssimazione a 15° e una cardioide nell'approssimazione a 45° .

1.3 Migrazione in mezzi non omogenei

Per semplicità consideriamo un campo costante a tratti corrispondente ad un mezzo stratificato. In questo contesto per un'onda viaggiante verso il basso, per z crescenti e all'indietro nel tempo, vale l'espressione seguente espressa nel dominio di *Fourier* nel caso bidimensionale:

$$\frac{\partial P}{\partial z}(k_x, z, \omega) = jk_z \cdot P(k_x, z, \omega), \quad (1.44)$$

che rappresenta l'equazione esatta di estrapolazione del campo nello strato $[z; z + \Delta z]$ con velocità $v = v_z$. Sappiamo che k_z deve assumere valori reali, e possiamo scrivere:

$$k_z = \frac{\omega}{v} \cdot \sqrt{1 - K^2}, \quad (1.45)$$

con $K = \frac{vk_x}{\omega}$, ed il vincolo $\frac{|\omega|}{|k_x|} > v$.

Un modo per impostare il problema [9] relativo ad un mezzo stratificato con variazioni laterali di velocità $v = v_z(x)$ può essere quello di operare nel dominio di *Fourier*. A questo scopo possiamo preventivamente approssimare la radice $\mathcal{R} = \sqrt{1 - K^2}$ espandendo in serie, ad esempio tramite serie di *Taylor* o frazioni continue (figura (1.11)). Un'approssimazione molto utilizzata è quella in frazioni continue,

$$\mathcal{R}_{n+1}^{cf} = 1 - \frac{K^2}{1 + \mathcal{R}_n^{cf}}, \quad \mathcal{R}_0^{cf} = 1. \quad (1.46)$$

Un'espansione accurata può ottenersi anche utilizzando l'algoritmo di *Newton*, espresso attraverso la seguente relazione ricorsiva:

$$\mathcal{R}_{n+1}^{nw} = \frac{1}{2} \cdot \left(\mathcal{R}_n^{nw} + \frac{1 - K^2}{\mathcal{R}_n^{nw}} \right), \quad \mathcal{R}_0^{nw} = 1. \quad (1.47)$$

Per verificare se una tale espansione converge all'equazione (1.45), è sufficiente fare tendere $n \rightarrow \infty$ nelle equazioni (1.46) ed (1.47).

Iterando entrambe le espressioni precedenti, otteniamo due approssimazioni della radice quadrata (troncando l'espansione al secondo passo):

$$\mathcal{R}_2^{cf} = 1 - \frac{K^2}{2 - \frac{1}{2}K^2}, \quad (1.48)$$

$$\mathcal{R}_2^{nw} = 1 - \frac{K^2}{2} \cdot \frac{1 - \frac{1}{4}K^2}{1 - \frac{1}{2}K^2}. \quad (1.49)$$

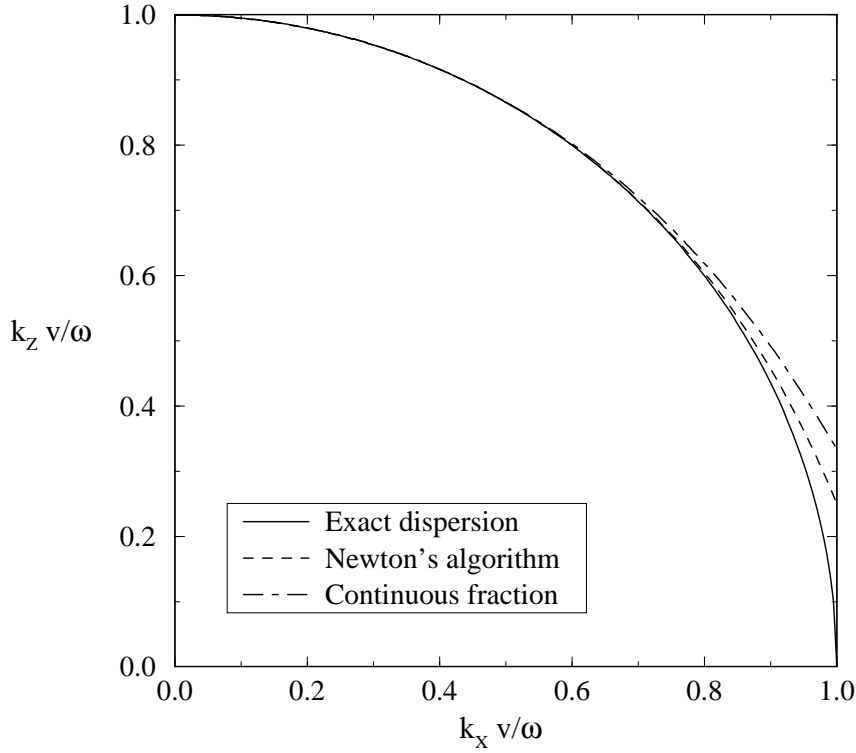


Figura 1.11: Approssimazioni della radice quadrata.

Sostituendo l'approssimazione \mathcal{R}^{nw} (o \mathcal{R}^{cf}) nell'equazione (1.48), si ottiene:

$$k_z = \frac{\omega}{v} + \mathcal{G}\left(\frac{v}{2\omega}, k_x\right) \quad (1.50)$$

In questo modo possiamo approssimare l'estrapolazione in profondità espressa nell'equazione (1.44), nel modo seguente:

$$\frac{dP}{dz}(k_z, z, \omega) = j \left[\frac{\omega}{v} + \mathcal{G}\left(\frac{v}{2\omega}, k_x\right) \right] \cdot P(k_x, z, \omega). \quad (1.51)$$

L'equazione (1.51) può essere risolta utilizzando un metodo a passi frazionari basato sulla fattorizzazione di *Cranck-Nicholson*. Il metodo numerico che ne risulta è consistente con un sistema di equazioni del tipo:

$$\frac{dP_0}{dz}(k_z, z, \omega) = \frac{j\omega}{v} \cdot P_0(k_z, z, \omega), \quad P_0(k_z, z, \omega) = P(k_z, z, \omega) \quad (1.52)$$

$$\frac{dP}{dz}(k_z, z, \omega) = j \cdot \mathcal{G} \cdot P(k_z, z, \omega), \quad P(k_z, z, \omega) = P_0(k_z, z + \Delta z, \omega) \quad (1.53)$$

che risulta essere la forma decomposta dell'equazione (1.51).

Per ogni passo Δz , la soluzione dell'equazione (1.52) viene utilizzata come condizione iniziale per l'equazione (1.53). L'equazione (1.52) considera

l'onda viaggiante verticalmente con lunghezza d'onda orizzontale infinita ($k_x = 0$), mentre l'equazione (1.53) tiene conto della correzione orizzontale. Sotto l'ipotesi di omogeneità dello strato $[z, z + \Delta z]$ nel quale il campo d'onda corrispondente è costante $v = v_z$, risolvere l'equazione (1.53) è semplicemente una moltiplicazione di $P(k_x, z, \omega)$ per un'operatore di traslazione di fase. Questa moltiplicazione può essere svolta nel dominio (x, ω) come segue:

$$P_1(x, z, \omega) = P(x, z, \omega) \cdot e^{j \cdot \left(\frac{\omega}{v}\right) \cdot \Delta z}. \quad (1.54)$$

Per esprimere l'equazione (1.53) nel dominio (x, ω) , occorre moltiplicare per il denominatore di \mathcal{G} e quindi calcolare la trasformata inversa di *Fourier* di tale equazione rispetto a k_x .

In particolare, utilizzando l'approssimazione (1.48), otteniamo

$$\mathcal{G}^{cf} = \frac{-\left(\frac{v}{2\omega}\right) k_x^2}{1 - \left(\frac{v}{2\omega}\right)^2 k_x^2} \quad (1.55)$$

mentre con l'approssimazione (1.44) abbiamo

$$\mathcal{G}^{nw} = -\left(\frac{v}{2\omega}\right) \cdot k_x^2 \cdot \left[\frac{1 - \left(\frac{v}{2\omega}\right)^2 \cdot k_x^2}{1 - 2 \left(\frac{v}{2\omega}\right)^2 \cdot k_x^2} \right]. \quad (1.56)$$

L'equazione d'onda può essere scritta, nel dominio (x, ω) , nella forma:

$$\left[1 + \left(\frac{v}{2\omega}\right)^2 \frac{\partial^2}{\partial x^2} \right] \frac{\partial P^{cf}}{\partial z} = \frac{jv}{2\omega} \frac{\partial^2 P^{cf}}{\partial x^2} \quad (1.57)$$

$$\left[1 + 2 \left(\frac{v}{2\omega}\right)^2 \frac{\partial^2}{\partial x^2} \right] \frac{\partial P^{nw}}{\partial z} = \frac{jv}{2\omega} \frac{\partial^2}{\partial x^2} \left[1 + \left(\frac{v}{2\omega}\right)^2 \frac{\partial^2}{\partial x^2} \right] P^{nw}. \quad (1.58)$$

L'*imaging* a $z + \Delta z$ si ottiene attraverso l'espressione:

$$P(x, z + \Delta z, 0) = \sum_{\omega} P(x, z + \Delta z, \omega), \quad (1.59)$$

nella quale la sommatoria viene effettuata su tutti i possibili ω .

Esprimendo il problema nel dominio (x, ω) , il controllo sulle onde evanescenti nel dominio (k_x, ω) è stato perso. Un approccio pratico è quello di introdurre una frequenza di taglio di ω attorno allo zero.

L'espressione (1.59) rappresenta la fase di *imaging* del processo di migrazione.

Le due equazioni (1.57) e (1.58) sono due differenti approssimazioni dell'estrapolazione in profondità in un mezzo stratificato. Il vantaggio sostanziale dell'approccio nello spazio (x, ω) consiste nella possibilità di considerare minori restrizioni sul campo di velocità e di includere nella trattazione le variazioni laterali all'interno di uno strato $[z, z + \Delta z]$.

Discretizzando le equazioni differenziali il problema viene trasformato in un sistema algebrico di equazioni lineari che deve essere risolto per tutti i possibili valori di ω :

$$Q(z) = P(z) \cdot e^{j\frac{\omega}{v}\Delta z}$$

$$AP(z + dz) = A^c Q(z), \quad A = \mathcal{A} \left(\frac{v}{2\omega} \right). \quad (1.60)$$

La matrice complessa \mathcal{A} è non singolare tridiagonale o pentadiagonale e A^c è la sua coniugata.

Il vettore P è l'approssimazione alle differenze finite della soluzione dell'equazione (1.60). Se il raggio spettrale del prodotto tra le matrici $(A^{-1}A^c)$ è uguale ad uno, lo schema di avanzamento (1.60) ha le proprietà di stabilità richieste per descrivere correttamente la propagazione dell'onda.

1.3.1 Traslazione di fase più interpolazione

Ciò che ci interessa ottenere è un'estrapolazione dei dati in un mezzo non omogeneo, utilizzando un metodo spettrale, operante nel dominio (k_x, k_y) - ω . Poniamoci, ancora una volta, per semplicità nel caso bidimensionale e riferiamoci all'equazione:

$$\frac{\partial P}{\partial z}(k_x, z, \omega) = j \left[\frac{\omega}{v} + \mathcal{G} \left(\frac{v}{2\omega}, k_x \right) \right] \cdot P(k_x, z, \omega). \quad (1.61)$$

È immediato notare la presenza di due termini, dei quali il primo

$$j\frac{\omega}{v}P(k_x, z, \omega)$$

è un termine esatto, mentre il secondo deve essere approssimato in qualche modo (*splitting approach*), come espresso nelle equazioni (1.52) e (1.53)¹⁰. La soluzione dell'equazione (1.53) può esprimersi nel dominio (x, ω)

¹⁰Sotto le ipotesi di mezzo omogeneo stratificato $[z, z + \Delta z]$ e con velocità costante $v = v_z$ su ogni strato.

come segue:

$$P_0(x, z, \omega) = P(x, z, \omega) \cdot e^{j\frac{\omega}{v}\Delta z}; \quad (1.62)$$

posto $\mathcal{G} = k_z - \frac{\omega}{v}$, la soluzione dell'equazione (1.53) nel dominio (k_x, ω) sarà:

$$P(k_x, z + \Delta z, \omega) = P_0(k_x, z, \omega) \cdot e^{j[(k_z - \frac{\omega}{v})\Delta z]}. \quad (1.63)$$

I due estrapolatori (1.61) e (1.63) rappresentano la forma decomposta (*splitting form*) dell'operatore di *Phase Shift* [8] [7]. In particolare l'equazione (1.53) viene espressa nel dominio (x, ω) e può essere calcolata in pratica per $v = v_z(x)$. Invece l'equazione (1.63) non è formalmente valida per variazioni laterali della velocità, ma supponendo la sua parziale validità, la possiamo valutare a differenti velocità per costruire la soluzione nel dominio (x, ω) . Ad esempio, consideriamo un campionamento di $v = v_z(x)$, includendo il valore minimo e quello massimo assunti appartenenti all'insieme di valori seguente:

$$v_{min} = v_0 < v_1 < v_2 < \dots < v_{M-1} = v_{Max}.$$

Utilizzando valori di questo insieme nel calcolo dell'equazione (1.63), possiamo estrapolare il campo di pressione $P_0(k_x, z, \omega)$, velocità dopo velocità e rappresentare ciascuno dei risultati ottenuti nel dominio (x, ω) :

$$P^{(n)}(x, z + \Delta z, \omega) = \sum_{k_x} P_0(k_x, z, \omega) \cdot e^{j[(k_z^{(n)} - \frac{\omega}{v_n})\Delta z + k_x x]} \quad (1.64)$$

con $n = 0, 1, 2, \dots, (M - 1)$ ¹¹. A partire dai campi di pressione trasformati $P^{(n)}(x, z + \Delta z, \omega)$ viene dedotto il campo finale per interpolazione (figura (1.12)):

$$P(x, z + \Delta z, \omega) = \sum_{n=0}^{M-1} P^{(n)}(x, z, \omega) \cdot \psi^n(x, z) \quad (1.65)$$

nella quale ψ^n rappresenta la funzione di interpolazione (*shape function*), la cui base viene definita in riferimento alle velocità v_{n-1} e v_{n+1} .

Nel semplice caso d'interpolazione lineare, avremo:

$$\psi^n(x, z) = \begin{cases} \frac{v_z(x) - v_{n-1}}{v_n - v_{n-1}} & \text{dove } v_{n-1} \leq v_z(x) \leq v_n \\ \frac{v_{n+1} - v_z(x)}{v_{n+1} - v_n} & \text{dove } v_{n-1} \leq v_z(x) \leq v_n \\ 0 & \text{altrove} \end{cases} \quad (1.66)$$

¹¹Tenendo conto della relazione $\frac{|\omega|}{|k_x|} > v_n$ per evitare le componenti spurie.

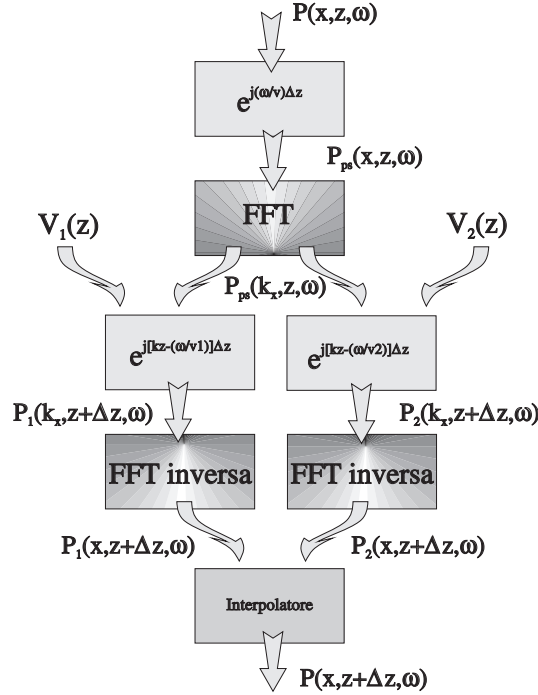


Figura 1.12: Algoritmo PSPI. A partire dalla valutazione del phase shift alle varie velocità si determina, per interpolazione, il campo di pressione ad una certa profondità per ogni ω d'interesse.

Sostituendo l'equazione (1.66) nella (1.65), la soluzione interpolata diviene:

$$P(x, z + \Delta z, \omega) = P^{(n)}(x, z + \Delta z, \omega) \cdot \left[\frac{v_{n+1} - v_z(x)}{v_{n+1} - v_n} \right] + P^{(n+1)}(x, z + \Delta z, \omega) \cdot \left[\frac{v_z(x) - v_{n-1}}{v_n - v_{n-1}} \right], \quad (1.67)$$

per tutti i valori di x tali che $v_n \leq v_z(x) \leq v_{n+1}$. Per quanto riguarda l'*imaging* dei riflettori alla profondità $z + \Delta z$, può essere realizzata attraverso l'espressione seguente:

$$P(x, z + \Delta z, t = 0) = \sum_{\omega} P(x, z + \Delta z, \omega). \quad (1.68)$$

Il metodo prende il nome di *Phase Shift Plus Interpolation* (PSPI) e costituisce un'alternativa valida alla migrazione mediante metodiche alle differenze finite ¹².

Il vantaggio essenziale sta nel fatto che tutte le velocità di riferimento fanno parte (in dipendenza dal tipo di interpolazione utilizzata) della soluzione

¹²In pratica si dimostra che questo approccio fornisce prestazioni confrontabili con gli schemi alle differenze finite nel dominio (x, ω) .

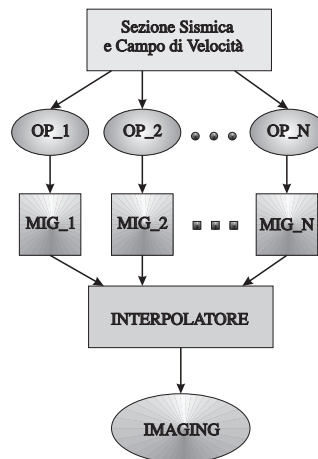


Figura 1.13: Algoritmo PSPI. Viene evidenziato un parallelismo nei task (determinazione del Phase Shift alle diverse velocità); i dati relativi ad un singolo task possono essere allocati su un unico processore (uno per ogni task), oppure distribuiti su svariati processori dipendentemente dall'implementazione che s'intende realizzare.

analitica (1.68). Ogni soluzione può essere calcolata in parallelo utilizzando l'algoritmo di *Phase Shift* (figura 1.13)). È importante sottolineare che il problema della determinazione delle velocità di riferimento ¹³ e dello schema d'interpolazione, in modo ottimizzato, è a tutt'oggi un problema aperto ed in via di rapido sviluppo [6]. Esiste un unico modello che attualmente fornisce buoni risultati anche per campi di velocità variabili. Si tratta di un modello statistico dovuto a *Bonomi* (1995), utilizzato per la selezione delle velocità di riferimento [3].

¹³Un modo di scegliere le velocità di riferimento potrebbe essere quello in forma di progressione geometrica del tipo $\frac{v_{n+1}}{v_n} = R$, con $n = 0, 1, 2, \dots, M - 1$, con $(M - 1)$ il più piccolo intero tale che $\frac{v_{Max}}{v_{min}} \geq R^M$ [9].

1.4 L'integrale di Kirchhoff

Consideriamo la derivazione originale dell'algoritmo noto come integrale di *Kirchhoff*, partendo dal caso in cui la velocità sia costante. La trattazione seguente [1] si adatta bene anche al caso generale di onde elastiche così come alle onde elettromagnetiche (potenziale scalare).

Supponiamo di risolvere un problema al contorno per l'equazione delle onde in un dominio $V \subset \mathcal{R}^3$, con condizioni al contorno specificate alla frontiera ∂V . Riferiamoci all'equazione d'onda $\nabla^2 P(\mathbf{r}, t) - \frac{1}{v^2} \frac{\partial^2 P(\mathbf{r}, t)}{\partial t^2} = 0$ per ogni $\mathbf{r} \in V$.

Il problema è completamente specificato se sono indicati i valori iniziali di P e della sua derivata normale $\frac{\partial P}{\partial n}$ su tutto il dominio V (esclusa la frontiera) per $t = 0$, inoltre P (oppure $\frac{\partial P}{\partial n}$) fissato sulla frontiera ∂V .

Abbiamo due tipi di condizioni al contorno:

- specificare la funzione al contorno significa risolvere il problema di *Dirichlet*:

$$P|_{\partial V} = f, \forall t$$

- specificare la derivata direzionale

$$\left. \frac{\partial P}{\partial n} \right|_{\partial V} = h, \forall t$$

significa risolvere il problema di *Neumann*.

Nella soluzione delle equazioni differenziali è spesso possibile specificare entrambe le condizioni simultaneamente (problema di *Cauchy*), questo non è il caso dell'equazione scalare delle onde [2].

Con riferimento all'esperimento di acquisizione di dati sismici, consideriamo le sorgenti del campo di pressione localizzate sulla superficie terrestre che fa parte della frontiera del dominio ∂V , in questo modo utilizzeremo l'equazione omogenea delle onde all'interno del volume V .

Il *principio di Huygens* può essere sintetizzato in termini di sovrapposizione degli effetti prodotti da un numero infinito di punti sorgente. Ciò può essere espresso definendo una risposta all'impulso, ossia una funzione di *Green*, come soluzione dell'equazione d'onda generata da una sorgente

te puntiforme. Definiamo una risposta impulsiva che dipende dalle coordinate spazio-temporali della sorgente e dell'osservatore $G(\mathbf{r}_s, t_s; \mathbf{r}, t)$ come soluzione dell'equazione:

$$\nabla^2 G - \frac{1}{v^2} \frac{\partial^2 G}{\partial t^2} = -4\pi \cdot \delta(\mathbf{r} - \mathbf{r}_s) \delta(t - t_s) \quad (1.69)$$

con (\mathbf{r}_s, t_s) riferito alla sorgente del campo.

Se moltiplichiamo l'equazione omogenea per la funzione G e l'equazione (1.69) per la funzione P , sottraendo membro a membro si ottiene:

$$G\nabla^2 P - P\nabla^2 G - \frac{1}{v^2} \left[G \frac{\partial^2 P}{\partial t^2} \right] = 4\pi \delta(\mathbf{r} - \mathbf{r}_s) \delta(t - t_s) \quad (1.70)$$

che può essere riscritta nella forma:

$$\nabla \cdot [G\nabla P - P\nabla G] - \frac{1}{v^2} \frac{\partial}{\partial t} \left[G \frac{\partial P}{\partial t} - P \frac{\partial G}{\partial t} \right] = 4\pi \delta(\mathbf{r} - \mathbf{r}_s) \delta(t - t_s). \quad (1.71)$$

Integrando quest'ultima equazione per ogni t e su tutto il volume V otteniamo:

$$\int_{-\infty}^{+\infty} \int_V \nabla \cdot [G\nabla P - P\nabla G] dV_s dt_s - \frac{1}{v^2} \int_{-\infty}^{+\infty} \int_V \frac{\partial}{\partial t} \left[G \frac{\partial P}{\partial t} - P \frac{\partial G}{\partial t} \right] dV_s dt_s = P(\mathbf{r}, t) \eta(\mathbf{r}) \quad (1.72)$$

per ogni $\mathbf{r} \in \mathcal{R}^3$, con $dV_s = dx_s dy_s dz_s$ e:

$$\eta(\mathbf{r}) = \begin{cases} 4\pi & \text{se } \mathbf{r} \in V - \partial V \\ 2\pi & \text{se } \mathbf{r} \in \partial V \\ 0 & \text{altrove} \end{cases} \quad (1.73)$$

Il primo integrale nell'equazione (1.72) può essere convertito in un integrale di superficie utilizzando il *teorema di Gauss-Ostrogradskij*:

$$\int \int \int_V \nabla p dV = \int \int_{\partial V} p n dA \quad (1.74)$$

Per la causalità della risposta impulsiva G , il limite inferiore dell'integrale temporale al primo membro dovrà porsi a zero, inoltre il secondo membro della (1.72), integrando nel tempo può essere espresso nella forma seguente:

$$\int_{V_s} \left[\frac{1}{v^2} \left(G \frac{\partial P}{\partial t} - P \frac{\partial G}{\partial t} \right) \right]_{-\infty}^{+\infty} dV_s \quad (1.75)$$

e considerando che per $t = -\infty$ la funzione G vale zero (causalità) mentre per $t = \infty$ è P ad essere nulla, avremo che il secondo membro dell'equazione (1.72) risulta nullo. Il risultato è il *teorema integrale di Kirchhoff* esprimibile come segue:

$$P(\mathbf{r}, t)\eta(\mathbf{r}) = \int_{-\infty}^{+\infty} \int_{\partial V} [G\nabla P - P\nabla G] \mathbf{n} dA_s dt_s \quad (1.76)$$

$\forall \mathbf{r} \in \mathcal{R}^3$; \mathbf{n} rappresenta il versore normale, esterno alla frontiera ∂V . L'integrale di *Kirchhoff* fornisce la soluzione dell'equazione d'onda P ovunque in V , quando i valori di P e di $\frac{\partial P}{\partial n}$ sono noti su tutta la frontiera ∂V . In generale le due condizioni al contorno non possono essere specificate contemporaneamente in modo consistente per l'equazione d'onda, ciò significa che dovremo risolvere l'equazione per una condizione o per l'altra. In pratica, supponendo di risolvere il problema con la *condizione di Neumann*, avremo:

$$f(\mathbf{r}, t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{\partial V} \left[Gh - f \frac{\partial G}{\partial n} \right] dA_s dt_s \quad (1.77)$$

$\forall \mathbf{r} \in \partial V$; in questo modo specifichiamo la derivata $\partial P/\partial n|_{\partial V} = h$ e risolviamo rispetto a $P_{\partial V} = f$ (equazione integrale in $f(\mathbf{r}, t)$)¹⁴. Non abbiamo specificato niente riguardo alla funzione di *Green* se non che deve essere causale (ossia nulla per $t < 0$). Se la funzione impulsiva G si annulla alla frontiera ∂V allora possiamo risolvere il problema di *Dirichelet* per il campo d'onda, direttamente dall'integrale di *Kirchhoff* per il quale il termine contenente la derivata $\partial P/\partial n|_{\partial V}$ può essere eliminato. In altri termini, se la funzione di *Green* G si annulla alla frontiera, l'integrale di *Kirchhoff* si riduce a:

$$P(\mathbf{r}, t) = \frac{1}{4\pi} \int_{-\infty}^{+\infty} \int_{\partial V} f(\mathbf{r}_s, t_s) \frac{\partial G}{\partial n}(\mathbf{r}, t; \mathbf{r}_s, t_s) dA_s dt_s \quad (1.78)$$

che risulta essere una relazione integrale esplicita, una volta nota l'espressione f .

Allo stesso modo, se la derivata $\frac{\partial G}{\partial n}|_{\partial V} = 0$, allora possiamo risolvere il problema di *Neumann* per il campo d'onda, dal momento in cui il termine

¹⁴Analogamente se utilizziamo la condizione $f = P|_{\partial V}$ risolveremo l'equazione integrale rispetto ad h .

incognito viene eliminato. Ciò significa che se la funzione di *Green* soddisfa particolari condizioni al contorno, l'equazione da risolvere non è nella forma completa.

1.4.1 Migrazione mediante l'integrale di Kirchhoff

L'adattamento dell'integrale di *Kirchhoff* alla migrazione dei dati sismici è dovuta a *Schneider* nel 1978. Consideriamo i dati registrati in superficie a $z = 0$. Il volume di nostro interesse (figura (1.14)) è il semispazio limitato

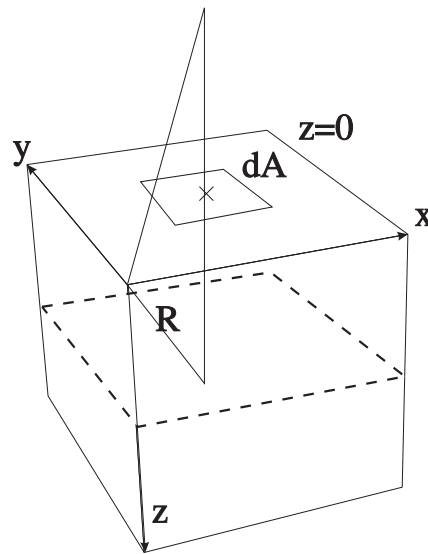


Figura 1.14: Sistema di riferimento.

dal piano $\pi(x, y)$ a $z = 0$, illimitato nelle altre direzioni. Consideriamo il caso particolare di velocità costante su tutto il dominio $V \cup \partial V$; vi sono due funzioni di *Green* che soddisfano la condizione di *Sommerfeld*, ossia possono essere considerate trascurabili per $z = 0$; esse vengono ricavate tramite il metodo delle immagini e pensate avere origine in profondità per $R_s = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2}$ molto grande:

$$\begin{aligned} G_r(\mathbf{r}, t; \mathbf{r}_s, t_s) &= \frac{\delta(t - t_s - \frac{R}{v})}{R} - \frac{\delta(t - t_s - \frac{R_s}{v})}{R_s} \\ G_a(\mathbf{r}, t; \mathbf{r}_s, t_s) &= \frac{\delta(t - t_s + \frac{R}{v})}{R} - \frac{\delta(t - t_s + \frac{R_s}{v})}{R_s} \end{aligned} \quad (1.79)$$

con $R = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z + z_s)^2}$.

Tali funzioni sono a simmetria sferica e rappresentano rispettivamente una

propagazione causale (G_a), che si allontana dalla sua origine per tempi crescenti (risposta impulsiva ritardata) e una propagazione anti-causale con fronte d'onda sferico propagantesi all'indietro nei tempi (G_a).

Per rappresentare una propagazione all'indietro utilizzeremo la risposta G_a sostituendola nell'integrale di *Kirchhoff* espresso nella forma (1.78). Osservando che $R_s(z_s) = R(-z_s)$ scriveremo:

$$P(\mathbf{r}, t) = \frac{1}{2\pi} \int_0^\infty \int_{\partial V} f(\mathbf{r}_s, t_s) \frac{\partial}{\partial z_s} \frac{\delta(t - t_s + \frac{R}{v})}{R} dA_s dt_s. \quad (1.80)$$

La variabile zeta appare solo nella forma $z - z_s$ nell'espressione R , ciò ci permette di scrivere:

$$P(\mathbf{r}, t) = -\frac{1}{2\pi} \frac{\partial}{\partial z} \int_0^\infty \int_{z_s=0} f(\mathbf{r}_s, t_s) \frac{\delta(t - t_s + \frac{R}{v})}{R} dA_s dt_s. \quad (1.81)$$

e sfruttando la proprietà funzionale della delta di *Dirac* di collassare il tempo d'integrazione, scriveremo:

$$P(\mathbf{r}, t) = -\frac{1}{2\pi} \frac{\partial}{\partial z} \int \int_{\pi(x,y)} \frac{f(\mathbf{r}_s, t + \frac{R}{v})}{R} dx_s dy_s. \quad (1.82)$$

che rappresenta la formula di migrazione di *Schneider*, con

$$f(\mathbf{r}_s, t_s) = P(x_s, y_s, z_s = 0, t + R/v)$$

Siamo partiti dall'equazione d'onda convertendola in un'equazione integrale introducendo un'arbitraria funzione di *Green*, quindi scegliendo una funzione G casuale e fissando le condizioni al contorno in maniera opportuna, abbiamo trasformato l'equazione integrale in una relazione integrale che mappa il campo d'onda al contorno (che deve essere noto) in una soluzione valida in tutti i punti del dominio. Per la migrazione ciò equivale a dire che a partire dai dati registrati in superficie a $z = 0$ si può riuscire a dedurre il campo P nel volume, a patto di essere in grado di fornire una adeguata funzione di *Green* ¹⁵.

Il problema è ora quello di determinare le adeguate condizioni al contorno. Lo *stacking* CMP dei dati produce una funzione $P_s(x, y, z = 0, t)$, ogni punto della quale approssima un esperimento *zero offset* indipendente. Ciò significa che i dati utilizzati come ingresso del processo di migrazione non

¹⁵Nel semplice caso esaminato si è considerata $v=costante$.

corrispondono alle effettive osservazioni fisiche. In questo senso la formula di migrazione di *Kirchhoff* può vedersi come un *mapping* a partire dallo spazio delle possibili condizioni al contorno nella soluzione dell'equazione d'onda.

Per effettuare una migrazione *post-stack* utilizzando la formula di *Kirchhoff*, dobbiamo identificare la funzione f nell'equazione 1.82) con la sezione sismica P_s (*CMP stack*); inoltre dobbiamo porre $v \rightarrow v/2$ e $t = 0$ in accordo col *modello dei riflettori esplodenti*.

A questo punto possiamo integrare sulla superficie di registrazione dei dati nel piano $\pi(x_s, y_s)$, ottenendo come risultato la sezione migrata in profondità:

$$P(\mathbf{r}) = \frac{1}{2\pi} \frac{\partial}{\partial z} \int_{z_s=0} \frac{P_s(\mathbf{r}_s, 2R/v)}{R} dA_s. \quad (1.83)$$

R rappresenta la distanza tra un punto sulla superficie in profondità ed il ricevitore posto a $z_s = 0$; integrare lungo la superficie $z_s = 0$ significa sommare i contributi provenienti dai dati *zero offset* misurati in superficie che dalle iperboli portano ai punti diffrattori in profondità.

L'equazione (1.83) può essere vista sotto forma di convoluzione spaziale che propaga i dati registrati tra strati adiacenti in profondità. In un mezzo stratificato, nel quale il campo di velocità sia costante su ciascuno degli strati, nel dominio di *Fourier* scriveremo:

$$P(k_x, k_y, z = \Delta z, \omega) = P(k_x, k_y, z, \omega) \mathcal{H}(k_x, k_y, \Delta z, \omega) \quad (1.84)$$

con

$$\mathcal{H}(k_x, k_y, \Delta z, \omega) = e^{\pm j \Delta z \sqrt{\left(\frac{\omega}{v}\right)^2 - k_x^2 - k_y^2}} \quad (1.85)$$

Ciò significa che, per un mezzo a velocità costante, la propagazione in profondità corrisponde ad una pura traslazione di fase. Nel caso di velocità variabili occorre valutare i tempi di transito dei raggi in qualche modo (ad esempio tramite l'equazione iconale), il problema principale diviene la risoluzione ottenibile.

Capitolo 2

Migrazione Reverse Time dell'equazione d'onda completa

La possibilità di utilizzare l'equazione d'onda completa per la migrazione dei dati sismici non è un'idea nuova [15] [4], ma la difficoltà di avere velocità di calcolo elevate e memoria sufficiente, per potere ottenere risultati accurati in tempi ragionevoli, ha permesso solo di recente un suo impiego in modo soddisfacente. I metodi di migrazione che utilizzano l'equazione d'onda completa si basano sulla soluzione dell'equazione

$$\left[\frac{\partial^2}{\partial t^2} - v^2(\mathbf{r})\nabla^2 \right] P(t; \mathbf{r}) = 0, \quad (2.1)$$

per $t > 0$, con condizioni iniziali fissate a $t = 0$, ed il campo di velocità $v(\mathbf{r})$ definito in un dominio finito \mathcal{V} .

I metodi numerici convenzionali per la migrazione di dati sismici, basati sulle differenze finite [12], utilizzano un solo termine propagatorio della soluzione dell'equazione d'onda [13], permettendo all'energia di propagarsi, nel caso specifico, esclusivamente verso il basso [11] ed indietro nei tempi. Il processo di migrazione permette, a partire dai dati registrati in superficie (sezione sismica *zero-offset*), di ricavare la posizione dei riflettori in profondità mediante un procedimento a ritroso nel tempo. Il segnale registrato

collassa in profondità nella posizione di *imaging* dei riflettori.

Procedere a ritroso nel tempo significa partire dal tempo finale di acquisizione (T_f) sino a raggiungere la condizione relativa al tempo iniziale ($t_0 = 0$); in altri termini, la condizione iniziale del processo di migrazione sarà:

$$\begin{cases} P(t = T_f; z, x, y) = 0 \\ \dot{P}(t = T_f; z, x, y) = 0 \end{cases} \quad (2.2)$$

mentre il risultato finale sarà la determinazione della “*superficie riflettente*”, rappresentata dalla relazione:

$$R(z, x, y) = P(t = 0; z, x, y) \quad (2.3)$$

che soddisfa il principio di *imaging* adottato (condizione di *imaging*).

Per ogni passo del processo di collasso in profondità viene imposta una condizione al contorno dinamica in superficie:

$$P(t; z = 0, x, y) = STACK(t, x, y) \quad (2.4)$$

dove $STACK(t, x, y)$ rappresenta la sezione sismica acquisita in superficie ($z = 0$). Le esigenze computazionali richiedono un volume di dati limitato e ciò comporta, dal punto di vista numerico, la definizione di condizioni al contorno assorbenti sulla frontiera del dominio considerato.

L'implementazione vera e propria è basata sulla selezione di un integratore temporale e di uno schema di calcolo del *laplaciano* per l'equazione acustica delle onde, oltre alle opportune condizioni al contorno assorbenti (*Absorbing Boundary Conditions*).

Dal punto di vista geofisico, il grande vantaggio di utilizzare una tale tecnica deriva dal fatto che essa permette di determinare strutture prodotte dalla propagazione di onde in corrispondenza di elevati gradienti di velocità (*turning waves*) e quindi di “*illuminare*” zone del sottosuolo che normalmente verrebbero ignorate dalle procedure *standard* di migrazione *one way*. In definitiva questo metodo rappresenta un utile approccio alla soluzione del problema dell'*imaging* di dettagli strutturali nascosti da grossi gradienti di velocità.

2.1 Implementazione numerica della migrazione

L'equazione d'onda può essere discretizzata nel tempo e nello spazio, ad esempio mediante metodi alle differenze finite (FDM). Nel caso in esame tali schemi sono stati selezionati bilanciando opportunamente le esigenze di accuratezza e costo computazionale. Le esigenze di stabilità e dispersione dei metodi adottati impongono vincoli stringenti nel dimensionamento degli intervalli di discretizzazione spaziale e temporale. L'obiettivo fondamentale per un utilizzo ottimale del codice è proprio quello di stabilire il massimo valore di discretizzazione spaziale e temporale compatibile con le esigenze di stabilità e dispersione minima dello schema. La soluzione di un tale problema permette di ridurre il costo computazionale della migrazione, compatibilmente con l'accuratezza che s'intende raggiungere, e l'occupazione di memoria ¹.

Lo schema d'integrazione scelto è il *leap frog* a salti frazionati. Esso presenta l'importante vantaggio di non determinare un aumento delle richieste di memoria rispetto agli schemi alle differenze finite centrate, seppure con un deciso miglioramento dell'accuratezza ² (quarto ordine). Lo schema alle differenze finite compatte utilizzato per la discretizzazione del *laplaciano* definisce un operatore differenziale di quart'ordine, utilizzato per ciascuna delle direzioni spaziali. Schemi di questo tipo sono *schemi impliciti* che coinvolgono tutti i punti lungo una direzione per il calcolo della derivata (*spectral like*). Una delle condizioni essenziali per la soluzione del problema della migrazione sismica è la stabilità dell'algoritmo. Per assicurare la stabilità è necessario valutare il coefficiente di stabilità r_{CFL} che definisce il vincolo che la discretizzazione spaziale e temporale devono soddisfare

¹La scelta degli algoritmi che determinano il giusto *trade-off* tra accuratezza e costo computazionale caratterizza fortemente l'approccio pragmatico adottato, orientato alla possibilità di migrare *dataset* reali (di grosse dimensioni) con grande accuratezza ed in tempi ragionevoli (in relazione alle dimensioni spaziali del *dataset* ed al numero di processori a disposizione).

²Ciò può essere dimostrato espandendo in serie di *Taylor*, nello spazio diretto, il campo di pressione P e la sua derivata prima temporale $Q = \dot{P}$.

affinchè la soluzione non sia divergente.

2.1.1 Metodi espliciti a più passi

Per risolvere le equazioni differenziali ordinarie di primo e second'ordine a più passi esistono svariati metodi [10]. Un modo semplice è quello di utilizzare l'espressione simmetrica:

$$y'(x_n) = \frac{dy}{dx}(x_n) = \frac{y(x_{n+1}) - y(x_{n-1}))}{2 \cdot h} + O(h^2)$$

che realizza il cosiddetto *algoritmo di Verlet*. Tale metodo esplicito può esprimersi come segue:

$$y_{n+1} = y_{n-1} + 2 \cdot h \cdot f(y_n, x_n),$$

ovvero ad un metodo a due passi, in quanto sono richiesti i due valori passati y_{n-1} ed y_n per il calcolo del valore attuale y_{n+1} ; inoltre è necessario fissare la condizione iniziale y_1 . L'errore di troncamento locale è $\mathcal{O}(h^3)$ mentre quello globale è $\mathcal{O}(h^2)$ ³.

³Data un'equazione differenziale ordinaria del prim'ordine

$$\frac{dy}{dx} = f(x, y)$$

con condizione iniziale $y(a) = c$, si riesce a determinare la soluzione esatta y solo per determinate funzioni $f(x, y)$. La soluzione di un problema come questo può essere considerata come una funzione $y(x, c)$, nella quale c è la condizione iniziale e quindi $y(x, c)$ rappresenta una famiglia di curve di parametro c . Se il parametro c non è costante ma varia per effetto di un disturbo, la soluzione tenderà ad approssimare a tratti la curva corrispondente al valore attuale ($c \pm i \cdot \Delta c$) sino a discostarsi nettamente dalla soluzione esatta. Nell'applicazione di metodiche numeriche, un comportamento di questo tipo è frequente ed è necessario distinguere gli errori che ne derivano.

Diciamo **errore globale** di troncamento nel punto x_{n+1} la differenza $y_{n+1} - y(x_{n+1})$, nella quale $y(x)$ è la soluzione esatta al problema di valore iniziale dato. Diciamo, invece, **errore locale** di troncamento in x_{n+1} la differenza tra il valore calcolato y_{n+1} ed il valore in x_{n+1} della soluzione dell'equazione differenziale passante per il punto (x_n, y_n) . L'errore globale è più semplicemente la somma degli errori locali.

La ragione nella differenza dell'entità tra i due errori sta nel fatto che il numero di errori locali che devono essere addizionati per costruire l'errore globale è inversamente proporzionale ad h .

Un metodo tra i più comuni è quello di *Heun* [10]:

$$\begin{cases} k_1 = h \cdot f(x_n, y_n) \\ k_2 = h \cdot f(x_n + h, y_n + k_1) \\ y_{n+1} = y_n + \frac{1}{2} \cdot (k_1 + k_2) \end{cases} \quad (2.5)$$

Un'importante classe di metodi espliciti è quella dei metodi di *Runge-Kutta*. Supposto noto $y(x_n)$ si vuole determinare un'approssimazione y_{n+1} di $y(x_n + h)$. L'idea di base è quella di calcolare $f(x, y)$ in un insieme di punti selezionati in modo opportuno in prossimità della soluzione, nell'intervallo (x_n, x_{n+h}) , e combinare questi valori in modo da produrre una buona accuratezza nel calcolo dell'incremento $y_{n+1} - y_n$.

Il più noto tra i metodi di *Runge-Kutta* è definito come segue:

$$\begin{cases} k_1 = h \cdot f(x_n, y_n) \\ k_2 = h \cdot f(x_n + \frac{1}{2} \cdot h, y_n + \frac{1}{2} \cdot k_1) \\ k_3 = h \cdot f(x_n + \frac{1}{2} \cdot h, y_n + \frac{1}{2} \cdot k_2) \\ k_4 = h \cdot f(x_n + h, y_n + k_3) \\ y_{n+1} = y_n + \frac{1}{6} \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) \end{cases} \quad (2.6)$$

In questo caso l'errore di troncamento globale risulta essere di $\mathcal{O}(h^4)$. È anche possibile ottenere il quint'ordine di accuratezza con sei valutazioni per passo [10].

Algoritmo di Verlet

In dinamica molecolare, è noto da tempo un interessante algoritmo attribuito a *Störmer*⁴ ed utilizzato per la prima volta da *Verlet* (1967) per integrare l'equazione del moto. Si tratta semplicemente di un metodo alle differenze finite, come quelli già visti, che rappresenta la soluzione diretta all'equazione differenziale del secondo ordine:

$$m \cdot \ddot{\mathbf{r}} = \mathbf{F}(\mathbf{r}). \quad (2.7)$$

⁴Gear, 1971.

Applicato al problema della dinamica lo schema si basa sulla posizione del punto materiale ad un certo istante t e ad un istante precedente ⁵:

$$\mathbf{r}(t + \Delta t) = 2 \cdot \mathbf{r}(t) - \mathbf{r}(t - \Delta t) + \Delta t^2 \cdot \mathbf{a}(t). \quad (2.8)$$

Come si può notare, la derivata prima del tempo (velocità) non compare; essa viene eliminata sommando gli sviluppi di *Taylor* delle posizioni agli istanti $(t + \Delta t)$ e $(t - \Delta t)$:

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t \cdot \mathbf{v}(t) + \frac{1}{2} \cdot \Delta t^2 \cdot \mathbf{a}(t) + \dots \quad (2.9)$$

$$\mathbf{r}(t - \Delta t) = \mathbf{r}(t) - \Delta t \cdot \mathbf{v}(t) + \frac{1}{2} \cdot \Delta t^2 \cdot \mathbf{a}(t) - \dots \quad (2.10)$$

Da ciò risulta che le velocità non sono necessarie per calcolare le traiettorie. La derivata prima della posizione è utile nella stima dell'energia cinetica, e può ricavarsi nel modo seguente:

$$\mathbf{v} = \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t - \Delta t)}{2 \cdot \Delta t}.$$

Per la posizione, l'errore locale è dell'ordine di $\mathcal{O}(\Delta t^4)$, mentre per la velocità è chiaramente dell'ordine di $\mathcal{O}(\Delta t^2)$; ciò significa che, in questo modo, anche la velocità può essere stimata con buona accuratezza a patto di conoscere la posizione ai due istanti richiesti.

Inoltre, la simmetria dello schema rispetto alle posizioni permette un'utile reversibilità dell'algoritmo nel tempo. Risulta evidente che l'avanzamento consta di due passi come per i metodi *predictor-corrector* pur distinguendosi da essi per le modalità di avanzamento. L'equazione (2.7) evidenzia

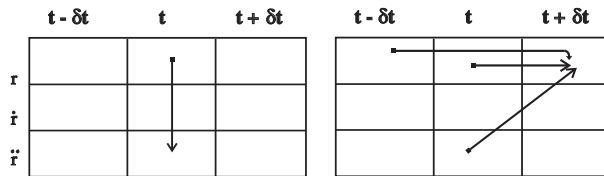


Figura 2.1: *Algoritmo di Verlet.*

la dipendenza dell'accelerazione dalla posizione allo stesso istante (a sinistra in figura 2.1), mentre l'equazione (2.8), attraverso le espressioni (2.9)

⁵Ricordando che $a(t) = \ddot{r}(t) = \frac{r(t+\Delta t) - 2 \cdot r(t) + r(t-\Delta t)}{\Delta t^2}$

e (2.10) indica la dipendenza della posizione all'istante $(t + \Delta t)$ dalla posizione e dall' accelerazione all' istante precedente oltre che dalla posizione all'istante $(t - \Delta t)$ (a destra in figura (2.1)). Nonostante i numerosi pregi di questo algoritmo, in particolari situazioni vengono evidenziate imprecisioni dovute principalmente alla somma di un piccolo termine $\mathcal{O}(\Delta t^2)$ con una grosso termine $\mathcal{O}(\Delta t^0)$, dato dalla differenza della posizione a due diversi istanti di tempo, come espresso nell'equazione (2.7). È questo il motivo principale che porta ad una modifica dell'algoritmo di *Verlet* in vari modi, dando vita a nuovi algoritmi di integrazione. Uno di questi è il cosiddetto schema di tipo *leap-frog* a passo frazionario.

2.1.2 Schema d'integrazione leap-frog a passo frazionario

Il nome ⁶ è intuibile dallo schema seguente:

$$\mathbf{v}(t + \frac{\Delta t}{2}) = \mathbf{v}(t - \frac{\Delta t}{2}) + \Delta t \cdot \mathbf{a}(t) \tag{2.11}$$

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t \cdot \mathbf{v}(t + \frac{\Delta t}{2}) \tag{2.12}$$

nel quale il termine dello spostamento viene ricavato da quello della velocità ritardata di $\frac{\Delta t}{2}$ e il termine di velocità dalla corrispondente accelerazione ritardata della stessa quantità; ne risulta uno schema a salti incrociati mostrato in figura (2.2). Vengono memorizzati i valori della funzione (po-

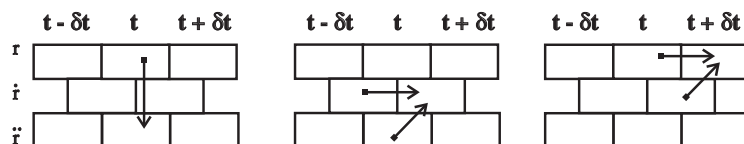


Figura 2.2: Schema Leap Frog.

sizione), della sua derivata seconda (accelerazione) calcolata all'istante t

⁶Letteralmente il *salto della rana*.

ed inoltre della derivata prima (velocità) calcolata in $(t - \frac{\Delta t}{2})$. La velocità corrente dovrà essere calcolata come una media geometrica, del tipo:

$$\mathbf{v}(t) = \frac{\mathbf{v}(t + \frac{\Delta t}{2}) + \mathbf{v}(t - \frac{\Delta t}{2})}{2}. \quad (2.13)$$

A partire da questo schema sono stati sviluppati altri schemi, di varia complessità, per soddisfare esigenze applicative tra le più svariate. Nel caso di nostro interesse utilizzeremo uno *schema esplicito del quarto ordine in tre passi temporali* [10] le cui variabili sono il campo di pressione $P(\mathbf{r}, t)$ e la sua derivata prima rispetto al tempo. Lo schema che viene utilizzato per risolvere l'equazione alle derivate parziali (2.15), ossia l'equazione:

$$\ddot{P} = v^2 \cdot \nabla^2 P = \Phi(P, t), \quad (2.14)$$

avendo cura di indicare ⁷

$$\begin{cases} \dot{P} = Q \\ \dot{Q} = \Phi(P, t), \end{cases} \quad (2.15)$$

è il seguente schema a passi frazionari:

$$\begin{cases} Q^{n+\frac{1}{3}} = Q^n - \alpha \cdot \Delta t \cdot \Phi(P^n, t^n) \\ P^{n+\frac{1}{2}} = P^n - \beta \cdot \Delta t \cdot Q^{n+\frac{1}{3}} \\ Q^{n+\frac{2}{3}} = Q^{n+\frac{1}{3}} - \gamma \cdot \Delta t \cdot \Phi(P^{n+\frac{1}{2}}, t_n + \frac{\Delta t}{2}) \\ P^{n+1} = P^{n+\frac{1}{2}} - \beta \cdot \Delta t \cdot Q^{n+\frac{2}{3}} \\ Q^{n+1} = Q^{n+\frac{2}{3}} - \alpha \cdot \Delta t \cdot \Phi(P^{n+1}, t_n + \Delta t) \end{cases} \quad (2.16)$$

con $t_n = n \cdot \Delta t$ e $P^n(\mathbf{r}) = P(t_n; \mathbf{r})$. I coefficienti α , β , γ e δ vengono ricavati in modo che l'errore locale sia dell'ordine di $\mathcal{O}(\Delta t^4)$ per il campo di pressione e di un ordine inferiore ($\mathcal{O}(\Delta t^3)$) per la sua derivata. Si ricavano i seguenti valori dei coefficienti:

$$\begin{cases} \alpha = \frac{1}{6} \\ \beta = \frac{1}{2} = \delta \\ \gamma = \frac{2}{3}. \end{cases} \quad (2.17)$$

Si può notare che è stata imposta la simmetria dei coefficienti, la quale conferisce all'algorithmo l'importante caratteristica di reversibilità temporale.

⁷Riferendoci alla formulazione *hamiltoniana*.

2.1.3 Discretizzazione del laplaciano

Esistono schemi alle differenze finite che permettono di migliorare l'efficienza nel calcolo delle derivate. Essi inoltre si prestano bene ad estensioni di scala necessarie per descrivere un'ampia classe di fenomeni fisici.

Le buone caratteristiche di risoluzione⁸ e di accuratezza⁹ hanno permesso l'applicazione di questi schemi di calcolo nei campi più disparati, a partire dalle prime applicazioni utilizzate nello studio dei flussi termodinamici turbolenti.

Nei paragrafi seguenti esaminiamo schemi compatti di discretizzazione della derivata prima e seconda, al fine di scegliere uno schema per la discretizzazione del laplaciano nell'equazione d'onda.

Approssimazione della derivata prima

Dati i valori di una funzione in un insieme di punti campionati uniformemente, lo schema di approssimazione di una funzione alle differenze finite viene espresso in termini di combinazione lineare dei punti di tale funzione. Consideriamo il semplice caso di reticolo uniforme. Per ogni nodo $1 \leq i \leq N$ del reticolo, avremo che $x_i = h(i - 1)$, con h passo della rappresentazione. Inoltre siano dati i valori della funzione f_i calcolata nei nodi, nella forma $f_i = f(x_i)$. L'approssimazione alle differenze finite della derivata prima di una tale funzione $f'(x_i) = \frac{df}{dx}(x_i)$ (calcolata nel nodo i) dipenderà dal valore della funzione nei nodi vicini.

Per differenze finite centrate di secondo e quarto ordine, l'approssimazione di f'_i dipende dagli insiemi (f_{i-1}, f_{i-2}) e $(f_{i-2}, f_{i-1}, f_{i+1}, f_{i+2})$ rispettivamente. Nei metodi spettrali esiste invece una dipendenza globale tra f'_i e tutti i nodi del reticolo. Indichiamo come schema alle differenze finite compatte uno schema che si propone di approssimare la derivata imponendo una dipendenza globale, intrinseca negli schemi spettrali. Una formulazione generale di tali schemi viene offerta da S. K. Lele (1991) [16].

⁸Per **risoluzione** intendiamo l'accuratezza alle diverse scale realizzabili su un dato reticolo.

⁹Nel senso dell'errore di troncamento.

Partendo dall'espressione implicita:

$$\sum_{n=-N}^N \alpha_n \cdot f'_{i+n} = \sum_{n=-N-1}^{N+1} \frac{\beta_n}{2 \cdot h \cdot n} \cdot f_{i+n} \quad (2.18)$$

con

$$\begin{cases} \alpha_0 = 1 \\ \beta_0 = 0 \\ \beta_{-n} = -\beta_n \\ \alpha_{-n} = \alpha_n \end{cases} \quad (2.19)$$

si esprime (per $N = 2$, per esempio) l'approssimazione della derivata prima, nella forma:

$$\begin{aligned} &\alpha_2 \cdot f'_{i-2} + \alpha_1 \cdot f'_{i-1} + f'_i + \alpha_1 \cdot f'_{i+1} + \alpha_2 \cdot f'_{i+2} = \\ &\beta_3 \cdot \left(\frac{f_{i+3} - f_{i-3}}{6 \cdot h}\right) + \beta_2 \cdot \left(\frac{f_{i+2} - f_{i-2}}{4 \cdot h}\right) + \beta_1 \cdot \left(\frac{f_{i+1} - f_{i-1}}{2 \cdot h}\right) \end{aligned} \quad (2.20)$$

Le relazioni che legano i coefficienti $\beta_1, \beta_2, \beta_3$ ed α_1 e α_2 vengono ricavate dal confronto fra i coefficienti di *Taylor* dell'espansione delle funzioni $f(x)$ e della sua derivata $f'(x)$ da approssimare. Il primo coefficiente diverso determina l'errore di troncamento e quindi l'ordine dell'approssimazione. I vincoli sui coefficienti, rispetto all'ordine dei termini dello sviluppo di *Taylor*, sono espressi nella tabella 2.1.

La tabella 2.2 rappresenta i valori e le relazioni tra i parametri che in-

Ordine	Vincolo
Secondo	$\beta_1 + \beta_2 + \beta_3 = 1 + 2 \cdot \alpha_1 + 2 \cdot \alpha_2$
Quarto	$\beta_1 + 2^2 \cdot \beta_2 + 3^2 \cdot \beta_3 = 2 \frac{3!}{2!} \cdot (\alpha_1 + 2^2 \cdot \alpha_2)$
Sesto	$\beta_1 + 2^4 \cdot \beta_2 + 3^4 \cdot \beta_3 = 2 \frac{5!}{4!} \cdot (\alpha_1 + 2^4 \cdot \alpha_2)$
Ottavo	$\beta_1 + 2^6 \cdot \beta_2 + 3^6 \cdot \beta_3 = 2 \frac{7!}{6!} \cdot (\alpha_1 + 2^6 \cdot \alpha_2)$
Decimo	$\beta_1 + 2^8 \cdot \beta_2 + 3^8 \cdot \beta_3 = 2 \frac{9!}{8!} \cdot (\alpha_1 + 2^8 \cdot \alpha_2)$

Tabella 2.1: *Legame tra i parametri per l'approssimazione della derivata prima.*

tercorrono fra i coefficienti dei vincoli, i quali concorrono ad individuare

LHS	RHS	α_2	α_1	β_3	β_2	β_1	Errore
3	5	$\alpha_2 = 0$	α	$\beta_3 = 0$	$\frac{1}{3}(4\alpha - 1)$	$\frac{2}{3}(\alpha + 2)$	$\frac{4}{5!}(3\alpha - 1)h^4 f^{(5)}$
3	5	0	$\alpha_1 \rightarrow 0$	0	$\beta_2 \rightarrow -\frac{1}{3}$	$\beta_1 \rightarrow \frac{4}{3}$	–
3	5	0	$\frac{1}{4}$	0	0	$\frac{3}{2}$	–
3	5	0	$\frac{1}{3}$	0	$\frac{1}{9}$	$\frac{14}{9}$	$\frac{4}{7!}h^6 f^{(7)}$
3	7	$\alpha_2 = 0$	α	$\frac{1}{10}(-3\alpha + 1) \neq 0$	$\frac{1}{15}(32\alpha - 9)$	$\frac{1}{6}(\alpha + 9)$	$\frac{12}{7!}(8\alpha + 3)h^6 f^{(7)}$
3	5	0	$\frac{1}{3}$	$\frac{1}{10}(-3\alpha + 1) = 0$	$\frac{1}{9}$	$\frac{14}{9}$	$\frac{4}{7!}h^6 f^{(7)}$
3	7	0	$\frac{3}{8}$	$-\frac{1}{80}$	$\frac{1}{5}$	$\frac{25}{16}$	$-\frac{36}{9!}h^8 f^{(9)}$

Tabella 2.2: Derivata prima tramite metodi alle differenze compatte.

le soluzioni dell'espressione (2.20) relativamente alle approssimazioni del secondo e quart'ordine indicati nella tabella 2.1. Nelle prime due colonne della tabella 2.2 viene indicato il numero massimo di termini, rispettivamente al primo e secondo membro dell'equazione (2.20); tali valori sono significativi per valutare i tempi di calcolo. Infatti le condizioni $\alpha_1 = 0$ e $\beta_3 = 0$ hanno l'effetto immediato di ridurre il numero dei termini utili alla soluzione dell'equazione (2.20), influenzando sui tempi di calcolo ma anche sull'accuratezza della soluzione. Nella stessa tabella viene riportato l'errore di troncamento relativo ad alcuni schemi di approssimazione. A partire da tali condizioni, ponendo $\alpha_1 \rightarrow 0$ viene ricavato uno schema di quart'ordine alle differenze finite centrate del tipo:

$$f'_i = \frac{1}{3} \cdot \left(\frac{f_{i+2} - f_{i-2}}{4h} \right) + \frac{4}{3} \cdot \left(\frac{f_{i+1} - f_{i-1}}{2h} \right) \tag{2.21}$$

media ponderata tra due schemi alle differenze finite centrate di second'ordine. Allo stesso modo, per $\alpha_1 = \frac{1}{3}$ si ottiene lo schema classico di Padè nella forma:

$$\frac{1}{3} \cdot f'_{i-1} + f'_i + \frac{1}{3} \cdot f'_{i+1} = \frac{3}{2} \cdot \left(\frac{f_{i+1} - f_{i-1}}{2h} \right) \tag{2.22}$$

Per $\alpha_1 = \frac{1}{3}$ l'errore di troncamento diventa del sest'ordine ($\epsilon_t = \frac{4}{7!}h^6 f^{(7)}$) come riportato nella tabella 2.2. Ponendo $\beta_3 \neq 0$ la famiglia di schemi estende la sua dipendenza a due parametri, relativamente agli schemi tridiagonali di quart'ordine. Con $\alpha_1 = \frac{1}{3}$ si trova lo schema di sest'ordine già visto:

$$\alpha_2 = 0, \quad \alpha_1 = \alpha$$

$$\beta_1 = \frac{1}{6}(\alpha + 9), \quad \beta_2 = \frac{1}{15}(32\alpha - 9), \quad \beta_3 = \frac{1}{10}(-3\alpha + 1), \tag{2.23}$$

che può essere specializzato in uno schema dell'ottavo ordine ponendo $\alpha = \frac{3}{8}$; in questo caso l'errore è pari a:

$$\epsilon_t = -\frac{36}{9!}h^8 f^{(9)} \tag{2.24}$$

che rappresenta la migliore accuratezza tra gli schemi tridiagonali ottenibili tramite l'equazione (2.20). Gli schemi pentadiagonali, generati per $\alpha_1 \neq 0$, raggiungono accuratezze maggiori ma hanno l'inconveniente di essere più costosi dal punto di vista computazionale ¹⁰. Se la funzione è periodica lungo x , allora l'espressione (2.20) può essere risolta per ciascun nodo come un sistema lineare di equazioni con incognite le derivate. La matrice associata al sistema in questione risulterà *pentadiagonale ciclico* [16] quando $\alpha_2 \neq 0$ mentre sarà *tridiagonale ciclico* per $\alpha_2 = 0$:

$$\begin{pmatrix} d_1 & e_1 & 0 & 0 & 0 & 0 & 0 & 0 & c_1 \\ c_2 & d_2 & e_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_3 & d_3 & e_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{n-1} & d_{n-1} & e_{n-1} \\ e_n & 0 & 0 & 0 & 0 & 0 & 0 & c_n & d_n \end{pmatrix} \tag{2.25}$$

Nel caso in cui le condizioni al bordo fossero non periodiche, sarebbero necessarie delle condizioni addizionali per i nodi posti sulla frontiera.

Nel caso di condizioni fisse, avremo $f'_1 = b_1$ e $f'_n = b_n$:

¹⁰Per la trattazione degli schemi pentadiagonali di approssimazione della derivata prima, vedere [16].

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha_1 & 1 & \alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_1 & 1 & \alpha_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 & 0 \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_1 & 1 & \alpha_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f'_1 \\ f'_2 \\ f'_3 \\ . \\ . \\ . \\ . \\ f'_{n-1} \\ f'_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ . \\ . \\ . \\ . \\ b_{n-1} \\ b_n \end{bmatrix} \quad (2.26)$$

Il sistema così ottenuto può essere indicato come una efficiente soluzione numerica del problema dell'approssimazione della derivata prima.

Approssimazione della derivata seconda

L'approssimazione delle derivate seconde attraverso schemi alle differenze finite compatte è analoga a quella vista per le derivate prime.

Partiamo da un'espressione del tipo:

$$\begin{aligned} & \alpha_2 \cdot f''_{i-2} + \alpha_1 \cdot f''_{i-1} + f''_i + \alpha_1 \cdot f''_{i+1} + \alpha_2 \cdot f''_{i+2} = \\ & = \beta_3 \cdot \left(\frac{f_{i+3} - 2f_i + f_{i-3}}{9 \cdot h^2} \right) + \beta_2 \cdot \left(\frac{f_{i+2} - 2f_i + f_{i-2}}{4 \cdot h^2} \right) + \beta_1 \cdot \left(\frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} \right), \end{aligned} \quad (2.27)$$

nella quale la f''_i rappresenta l'approssimazione della derivata seconda nel nodo $i - mo$. Le relazioni tra i coefficienti $\beta_1, \beta_2, \beta_3$ ed α_1 e α_2 vengono ricavate dal confronto con i coefficienti di *Taylor* di vario ordine. La forma di queste condizioni è molto simile a quella ottenuta per la derivata prima, ma cambiano i coefficienti moltiplicativi del secondo membro.

Il sistema che deve essere risolto è pentadiagonale o tridiagonale ed è definito dall'equazione (2.27). Per il caso non periodico sono necessarie delle condizioni al contorno addizionali.

Scegliendo $\alpha_2 = 0$ e $\beta_3 = 0$ dallo schema del quart'ordine, viene generata una famiglia di funzioni di un parametro con:

$$\alpha_2 = 0, \quad \beta_3 = 0$$

Ordine	Vincolo
Secondo	$\beta_1 + \beta_2 + \beta_3 = 1 + 2 \cdot \alpha_1 + 2 \cdot \alpha_2$
Quarto	$\beta_1 + 2^2 \cdot \beta_2 + 3^2 \cdot \beta_3 = \frac{4!}{2!} \cdot (\alpha_1 + 2^2 \cdot \alpha_2)$
Sesto	$\beta_1 + 2^4 \cdot \beta_2 + 3^4 \cdot \beta_3 = \frac{6!}{4!} \cdot (\alpha_1 + 2^4 \cdot \alpha_2)$
Ottavo	$\beta_1 + 2^6 \cdot \beta_2 + 3^6 \cdot \beta_3 = \frac{8!}{6!} \cdot (\alpha_1 + 2^6 \cdot \alpha_2)$
Decimo	$\beta_1 + 2^8 \cdot \beta_2 + 3^8 \cdot \beta_3 = \frac{10!}{8!} \cdot (\alpha_1 + 2^8 \cdot \alpha_2)$

Tabella 2.3: *Legame tra i parametri per l'approssimazione della derivata seconda.*

$$\beta_1 = \frac{4}{3} \cdot (1 - \alpha_1), \quad \beta_2 = \frac{4}{3} \cdot (-1 + 10 \cdot \alpha_1) \quad (2.28)$$

L'errore di troncamento del secondo membro dell'equazione (2.27) di un tale schema può esprimersi:

$$\epsilon_t = -\frac{4}{6!} \cdot (11 \cdot \alpha_1 - 2) \cdot h^4 f^{(6)}. \quad (2.29)$$

È interessante notare che per $\alpha_1 \rightarrow 0$ l'espressione (2.27) coincide con lo

α_2	α_1	β_3	β_2	β_1
$\alpha_2 = 0$	α	$\beta_3 = 0$	$\frac{1}{3}(10\alpha - 1)$	$\frac{4}{3}(1 - \alpha)$
0	$\alpha_1 = \frac{2}{11}$	0	$\frac{3}{11}$	$\frac{12}{11}$
$\alpha_2 \neq 0$	α_1	$\beta_3 \neq 0$	$\frac{1}{3}(-1 + 10\alpha_1 + 46\alpha_2 - 8\beta_3)$	$\frac{1}{3}(4 - 4\alpha_1 - 4\alpha_2 + 5\beta_3)$
α_2	α_1	$\frac{2-11\alpha_1+124\alpha_2}{20}$	$\frac{-3+24\alpha_1-6\alpha_2}{5}$	$\frac{6-9\alpha_1-12\alpha_2}{4}$
$38\frac{\alpha-9}{214}$	α	$\frac{1179\alpha-344}{2140}$	$\frac{2454\alpha-294}{535}$	$\frac{696-1191\alpha}{428}$
$\frac{43}{1798}$	$\frac{334}{899}$	$\frac{79}{1798}$	$\frac{1038}{899}$	$\frac{1065}{1798}$

Tabella 2.4: *Derivata seconda tramite metodi alle differenze compatte.*

schema di quart'ordine alle differenze finite centrate nella forma:

$$f_i'' = -\frac{1}{3} \left(\frac{f_{i+2} - 2 \cdot f_i + f_{i-2}}{4 \cdot h^2} \right) + \frac{4}{3} \left(\frac{f_{i+1} - 2 \cdot f_i + f_{i-1}}{h^2} \right); \quad (2.30)$$

ottenuta dall'espressione generale (2.27) tramite i coefficienti:

$$\alpha_1 = 0, \quad \alpha_2 = 0$$

$$\beta_1 = \frac{4}{3}, \quad \beta_2 = -\frac{1}{3}, \quad \beta_3 = 0. \quad (2.31)$$

Per $\alpha_1 = \frac{1}{10}$ si ricava lo schema classico di Padè:

$$\alpha_1 = \frac{1}{10}, \quad \alpha_2 = 0$$

$$\beta_1 = \frac{6}{5}, \quad \beta_2 = 0, \quad \beta_3 = 0, \tag{2.32}$$

mentre per $\alpha_1 = \frac{2}{11}$ si ottiene uno schema di sest'ordine tridiagonale nel quale i coefficienti hanno i seguenti valori:

$$\alpha_1 = \frac{2}{11}, \quad \alpha_2 = 0$$

$$\beta_1 = \frac{12}{11}, \quad \beta_2 = \frac{3}{11}, \quad \beta_3 = 0. \tag{2.33}$$

L'errore di troncamento risulta, in questo caso:

$$\epsilon_t = -\frac{8 \cdot 23}{11 \cdot 8!} \cdot h^6 f^{(8)}. \tag{2.34}$$

Quest'ultimo schema possiede una buona precisione ed è quello utilizzato nell'implementazione del solutore laplaciano. Si noti che la scelta di $\alpha_1 = 2/11$ annulla l'errore del quart'ordine espresso in (2.29) producendo uno schema con accuratezza del sest'ordine.

LHS	RHS	Errore
3	5	$-\frac{4}{6!}(11\alpha - 2)h^4 f^{(6)}$
3	5	$-\frac{8 \cdot 23}{11 \cdot 8!}h^6 f^{(8)}$
5	7	$-\frac{4}{6!}(-2 + 11\alpha_1 - 124\alpha_2 + 20\beta_3)h^4 f^{(6)}$
5	7	$-\frac{8}{8!}(9 - 38\alpha_1 + 214\alpha_2)h^6 f^{(8)}$
5	7	$\frac{899\alpha - 334}{2696400}h^8 f^{(10)}$
5	7	$\frac{619}{299043360}h^{10} f^{(12)}$

Tabella 2.5: Errore di troncamento.

In definitiva è stato implementato uno schema alle differenze finite compatte tridiagonale per la discretizzazione spaziale del tipo:

$$\frac{2}{11}f''_{i+1} + f''_i + \frac{2}{11}f''_{i-1} = \frac{3}{11} \frac{1}{4h^2} \cdot (f_{i+2} - 2f_i + f_{i-2}) + \frac{12}{11} \frac{1}{h^2} \cdot (f_{i+1} - 2f_i + f_{i-1}) \tag{2.35}$$

ossia uno schema del sest'ordine di accuratezza.

In pratica, supponendo di dovere calcolare la derivata seconda nella direzione z costruiremo un sistema lineare di N_z equazioni, caratterizzato da una matrice compatta (nel nostro caso tridiagonale) della forma seguente:

$$\begin{aligned} & \frac{2}{11}(f'')_{i+1} + (f'')_i + \frac{2}{11}(f'')_{i-1} = \\ & = \frac{3}{44}\frac{1}{h^2} \cdot f_{i+2} + \frac{12}{11}\frac{1}{h^2} \cdot f_{i+1} - \frac{51}{22}\frac{1}{h^2} \cdot f_i + \frac{12}{11}\frac{1}{h^2} \cdot f_{i-1} + \frac{3}{44}\frac{1}{h^2} \cdot f_{i-2} \end{aligned} \quad (2.36)$$

per $k = 3, 4, N_z - 3$; in prossimità della la frontiera del dominio, per $k = 2$ e $k = N_z - 1$, si adotta la soluzione classica di *Padè*:

$$\frac{1}{10}(f'')_{i+1} + (f'')_i + \frac{1}{10}(f'')_{i-1} = \frac{12}{10}\frac{1}{h^2} \cdot f_{i+1} - \frac{24}{10}\frac{1}{h^2} \cdot f_i + \frac{12}{10}\frac{1}{h^2} \cdot f_{i-1} \quad (2.37)$$

mentre sulla frontiera, per $k = 1$ e $k = N_z$, si pone $(f'')_{ij(k=1)} = 0$ e $(f'')_{ij(k=N_z)} = 0$. In questo modo si ottiene un sistema di equazioni globalmente tridiagonale, che può essere scritto in forma compatta nel modo seguente:

$$\mathbf{T}\vec{f}''(t) = \mathbf{R}\vec{f}(t); \quad (2.38)$$

con \mathbf{T} si è indicata la matrice tridiagonale relativa al primo membro dello schema di approssimazione, mentre con \mathbf{R} si è indicata la matrice pentadiagonale relativa alla combinazione lineare approssimante. Il vettore incognito $\vec{f}''(t)$ viene ricavato, punto per punto, in funzione di valori locali di $f_{ijk}(t)$, risolvendo il sistema:

$$\vec{f}''(t) = \mathbf{T}^{-1}\mathbf{R}\vec{f}(t). \quad (2.39)$$

Questo schema, applicato nelle tre direzioni indipendenti x , y e z , permette di approssimare il laplaciano.

Riduzione pari-dispari per sistemi tridiagonali

Consideriamo un sistema di equazioni $\mathbf{A}x = b$, nel quale \mathbf{A} sia tridiagonale, cioè con $a_{i,j} = 0$ se $|i - j| > 1$. Un tale sistema è esprimibile nel modo seguente:

$$\begin{cases} g_1 \cdot x_1 + h_1 \cdot x_2 = b_1 \\ f_i \cdot x_{i-1} + g_i \cdot x_i + h_i \cdot x_{i+1} = b_i \\ f_n \cdot x_{n-1} + g_n \cdot x_n = b_n \end{cases} \quad (2.40)$$

con $i = 2, 3, \dots, n - 1$.

In uno schema di questo tipo i termini g_i stanno sulla diagonale di **A** mentre f_i ed h_i sono gli elementi di **A** rispettivamente sotto e sopra la diagonale.

$$\begin{bmatrix} g_1 & h_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ f_2 & g_2 & h_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & f_3 & g_3 & h_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & f_{n-1} & g_{n-1} & h_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & f_n & g_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ b_{n-1} \\ b_n \end{bmatrix} \quad (2.41)$$

Il metodo di risoluzione del sistema tridiagonale che descriveremo nel seguito è noto col nome di *riduzione pari-dispari*¹¹[5]. L'idea di base sta nel fatto che per $g_i \neq 0$, possiamo risolvere l'equazione precedente per x_i in termini di x_{i-1} e x_{i+1} , rispetto ad elementi con la stessa parità.

Se risolviamo per ogni intero dispari i , e quindi sostituiamo l'espressione così ottenuta in termini di x_i nella rimanente equazione, otteniamo un sistema di equazioni nella sola variabile x_i , con i pari. Il sistema di equazioni è ancora tridiagonale ed ha la metà delle variabili¹².

La stessa procedura può applicarsi ricorsivamente sino ad arrivare ad N sistemi con una o due variabili; nel caso in cui le variabili siano 2^N il sistema di partenza viene ridotto ad N sistemi di una variabile ciascuno.

Per semplificare le equazioni utilizziamo la convenzione $x_0 = x_{n+1} = 0$, che rende l'equazione (2.40) valida anche per $i = 1$ oltre che per $i = n$. Risolviamo tale equazione per x_i ed otteniamo:

$$x_i = \frac{1}{g_i} \cdot (b_i - f_i \cdot x_{i-1} - h_i \cdot x_{i+1}). \quad (2.42)$$

Utilizziamo quest'ultima espressione, sostituendo i con $i - 1$ e $i + 1$, per eliminare x_{i-1} ed x_{i+1} dall'equazione (2.40).

¹¹Odd-even reduction.

¹²Si procede analogamente per quanto riguarda i nodi pari.

Ciò porta all'espressione :

$$\frac{f_i}{g_{i-1}} \cdot (b_{i-1} - f_{i-1} \cdot x_{i-2} - h_{i-1} \cdot x_i) + g_i \cdot x_i + \frac{h_i}{g_{i+1}} \cdot (b_{i+1} - f_{i+1} \cdot x_i - h_{i+1} \cdot x_{i+2}) = b_i, \quad (2.43)$$

la quale può essere semplificata nella forma seguente:

$$-\left(\frac{f_i f_{i-1}}{g_{i-1}}\right) x_{i-2} + \left(g_i - \frac{h_{i-1} f_i}{g_{i-1}} - \frac{h_i f_{i+1}}{g_{i+1}}\right) x_i - \left(\frac{h_i h_{i+1}}{g_{i+1}}\right) x_{i+2} = b_i - \frac{f_i}{g_{i-1}} b_{i-1} - \frac{h_i}{g_{i+1}} b_{i+1} \quad (2.44)$$

Consideriamo questa equazione per ciascuno degli indici pari con $1 \leq i \leq n$. Essa rappresenta un sistema nelle variabili $x_2, \dots, x_{2 \cdot \lfloor \frac{n}{2} \rfloor}$, ed è chiaramente un sistema tridiagonale; possiamo utilizzare la stessa procedura ricorsivamente, ottenendo sistemi sempre più piccoli, sino a giungere ad un sistema in una sola variabile che è anche la soluzione diretta del sistema.

La corrispondenza tra il sistema tridiagonale originale e quello dimezzato è caratterizzata da una dipendenza del tipo:

$$\begin{cases} f_{i-2}^{(1)} = -\left(\frac{f_i \cdot f_{i-1}}{g_{i-1}}\right) \\ f_i^{(1)} = \left(g_i - \frac{h_{i-1} \cdot f_i}{g_{i-1}} - \frac{h_i \cdot f_{i+1}}{g_{i+1}}\right) \\ f_{i+2}^{(1)} = -\left(\frac{h_i \cdot h_{i+1}}{g_{i+1}}\right) \\ b_i^{(1)} = b_i - \frac{f_i}{g_{i-1}} \cdot b_{i-1} - \frac{h_i}{g_{i+1}} \cdot b_{i+1} \end{cases} \quad (2.45)$$

L'algoritmo termina se viene tentata una divisione per zero; ciò non avviene se la matrice A è non singolare ovvero se gli elementi della diagonale sono diversi da zero.

Nei casi pratici si ha a che fare con matrici non singolari e quindi un tale metodo può essere utilizzato favorevolmente nella maggioranza dei casi.

Per quanto concerne il peso computazionale, in termini di tempo di calcolo (escluse le comunicazioni), si può immediatamente notare che per ogni passo il numero di variabili viene ridotto approssimativamente alla metà. Dopo $\mathcal{O}(\log(n))$ passi tutte le variabili, esclusa una, sono state eliminate.

Ad ogni stadio, abbiamo bisogno di calcolare i coefficienti del sistema ridotto; ciò è immediato a partire dall'equazione (2.44).

Da ciò discende che l'intero algoritmo può essere implementato in $\mathcal{O}(\log(n))$ passi temporali; la struttura delle comunicazioni è del tipo mostrato nella figura 2.3 (i tempi sono crescenti dal basso verso l'alto).

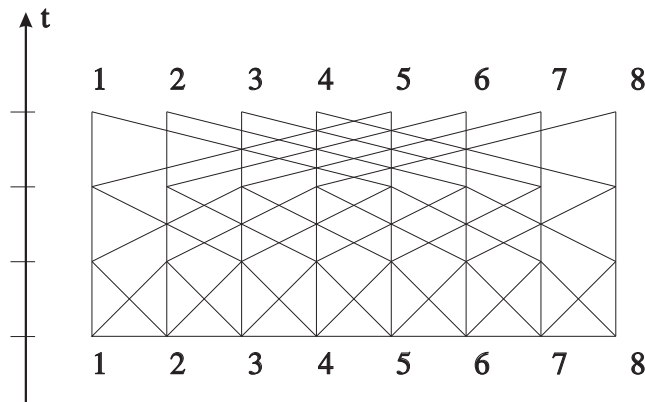


Figura 2.3: Schematizzazione butterfly per l'algoritmo di riduzione.

Il numero totale di operazioni è soltanto $\mathcal{O}(n)$, questo perchè il carico computazionale di ogni stadio è la metà di quello dello stadio precedente.

Ciò significa che esiste un algoritmo sequenziale in grado di impiegare $\mathcal{O}(n)$ passi. Inoltre, bisogna considerare che ogni algoritmo sequenziale ha bisogno di un certo tempo per leggere l'input.

Consideriamo nel seguito due possibili configurazioni di processori. Se abbiamo a che fare con una configurazione lineare di p processori con $p < n$, su ciascuno di essi verranno disposte n/p variabili (supponendo n/p intero). Per i primi $N = \lfloor \log(n/p) \rfloor$ passi dell'algoritmo, i calcoli saranno eseguiti, per lo più, localmente ad ogni processore. Il numero totale di operazioni aritmetiche eseguite su ciascuno dei processori dopo N passi è dell'ordine $\mathcal{O}(n/p)$ e le comunicazioni tra processori vicini incidono debolmente sul costo computazionale totale. Dopo l' N -mo passo rimane un sistema di $\mathcal{O}(p)$ variabili. Tale sistema si risolve con un costo di comunicazione pari a $\mathcal{O}(p)$ e un costo di calcolo dello stesso ordine; possiamo concludere che il tempo totale di calcolo è $\mathcal{O}(n/p) + \mathcal{O}(p)$.

Ottimizzando rispetto al numero p di processori, deve aversi $p = \mathcal{O}(\sqrt{n})$ e quindi il tempo totale di esecuzione (χ) sarà dell'ordine $\mathcal{O}(\sqrt{n})$, infatti:

$$\chi = a \cdot \frac{n}{p} + b \cdot p$$

$$\frac{d\chi}{dp} = -a \cdot \frac{n}{p^2} + b = 0 \quad \Rightarrow p \propto \sqrt{n}$$

per cui il tempo totale di esecuzione risulta decisamente migliore del tempo seriale $\mathcal{O}(n)$ ma è peggiore del risultato ottenibile considerando le comunicazioni istantanee $\mathcal{O}(\log(n))$.

2.1.4 Stabilità numerica

Lo studio della stabilità di un metodo numerico può essere condotta nel dominio dei numeri d'onda attraverso l'uso della trasformata di *Fourier*. L'equazione d'onda discretizzata nel tempo e nei numeri d'onda può rappresentarsi come segue:

$$\hat{P}(n+1; \mathbf{k}) - 2\hat{P}(n; \mathbf{k}) + \hat{P}(n-1; \mathbf{k}) = -h(\mathbf{k})\hat{P}(n; \mathbf{k}) \quad (2.46)$$

che equivale alla forma:

$$\begin{bmatrix} \hat{P}(n; \mathbf{k}) \\ \hat{P}(n+1; \mathbf{k}) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2 - h(\mathbf{k}) \end{bmatrix} \begin{bmatrix} \hat{P}(n-1; \mathbf{k}) \\ \hat{P}(n; \mathbf{k}) \end{bmatrix} = \mathcal{A} \begin{bmatrix} \hat{P}(n-1; \mathbf{k}) \\ \hat{P}(n; \mathbf{k}) \end{bmatrix} \quad (2.47)$$

con $h(\mathbf{k})$ dipendente dallo schema di discretizzazione adottato.

La rappresentazione in forma discreta di un'equazione alle derivate parziali (PDE), continua e lineare, deve essere *consistente* e *stabile*; consistenza e stabilità implicano la convergenza, nel senso che la soluzione dell'equazione discretizzata deve sovrapporsi alla corrispondente soluzione nel continuo al tendere del passo di discretizzazione a zero. In presenza di piccoli

K	<i>spettrale</i>	<i>compatto</i>	<i>IV - FD</i>	<i>II - FD</i>
<i>leap - frog</i>	0.78	0.935	1.061	2.225
<i>II - FD</i>	0.637	0.763	0.867	1.0

Tabella 2.6: Valori del parametro K per differenti combinazioni tra schemi di discretizzazione spaziale e temporale.

errori di troncamento un metodo numerico stabile permette di ricavare la soluzione senza degradarla ulteriormente. Si può dimostrare che, nel caso tridimensionale, per un campo di velocità variabile deve essere soddisfatta la relazione:

$$C_{max} \Delta t \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}} \leq K, \quad (2.48)$$

dove K è un parametro dipendente dallo schema utilizzato sia nel tempo che nello spazio ed inoltre:

$$C_{max} = \max\{c(x, y, z)\}.$$

L'equazione (2.48) potrà scriversi nella forma più debole:

$$\frac{C_{max} \cdot \Delta t}{\min(\Delta x, \Delta y, \Delta z)} \leq \frac{K}{\sqrt{3}}. \quad (2.49)$$

Capitolo 3

Implementazione parallela del migratore RTM

In questo paragrafo verrà analizzata nel dettaglio la struttura del migratore tridimensionale (figura (3.1)), denominato RTM3D, basato sull'equazione acustica completa delle onde, con particolare riferimento alla struttura dei dati e alle prestazioni del solutore tridiagonale parallelo che rappresenta il nucleo di calcolo del migratore proposto.

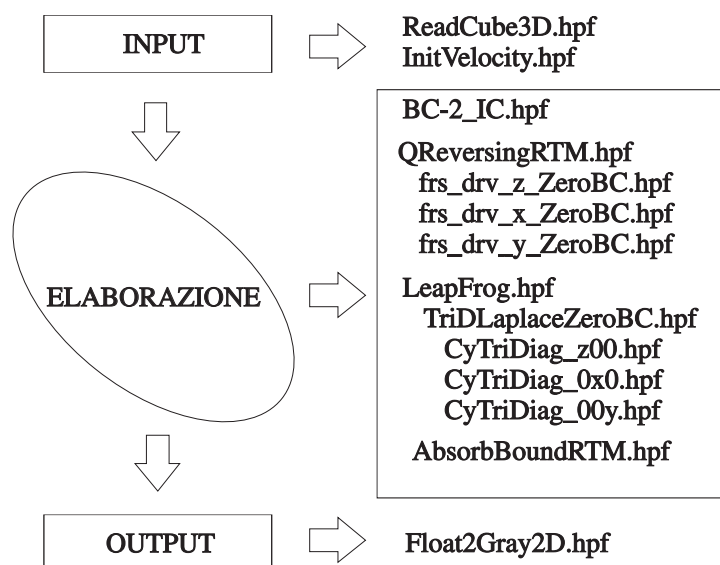


Figura 3.1: Il migratore RTM-3D: struttura del programma principale.

3.1 Il programma principale

Il flusso delle istruzioni del programma principale può essere schematizzato per semplicità nelle tre fasi di *input* dei dati, *elaborazione* ed *output*. I dati in ingresso sono rappresentati dal cubo tridimensionale (z, x, y) del campo di velocità e dalla sezione sismica $(t; x, y)$ che rappresenta le misure effettuate in superficie a $z = 0$ per $t_i = i \cdot \Delta t$ (con $t_{max} = N_T \cdot \Delta t$). La sezione sismica può avere una duplice origine, dai dati reali oppure dalla simulazione ottenuta tramite il modellatore (in questo caso diremo che si tratta di una *sezione sismica sintetica*). I parametri d'ingresso, attraverso il meccanismo NAMELIST, vengono letti da file. Le variabili principali sono *allocate* all'inizio del programma principale e *deallocate* alla fine in quanto vengono aggiornate durante il ciclo di elaborazione. La dimensione di queste matrici incide fortemente sulle richieste di memoria in esecuzione, si pensi che un cubo con 100 elementi reali per lato è costituito da $100^3 = 1.000.000$ elementi ovvero *quattro milioni di byte* (essendo un numero in virgola mobile codificato da 32 bit). Le altre variabili vengono perciò *allocate* dinamicamente laddove vengono utilizzate e quindi *deallocate* alla fine delle *subroutine* che le utilizzano.

Si è cercato, ove possibile, di rendere il flusso modulare in modo tale da avere sempre presente l'ordine logico funzionale e facilitare il *debugging*. Le chiamate a funzione incidono sui tempi di calcolo ma non in modo rilevante rispetto al *processing* e alla considerevole mole di dati.

Possiamo caricare la sezione sismica da file mediante la *subroutine*

```
ReadCube3d(NT, NX, NY, SEISMIC_SEC, String)
```

che prende in ingresso le dimensioni spaziali e quella temporale della sezione sismica, il nome del file contenente tale sezione e la matrice tridimensionale che la conterrà durante l'elaborazione. La chiamata alla funzione di lettura è preceduto da un meccanismo di *default*; tale meccanismo provvede a controllare se è presente un file di nome fissato per effettuare il caricamento dei parametri, il controllo viene effettuato nella lista dei nomi (file dei parametri), se tale nome è assente provvede a produrre una sezione

sismica sintetica a partire dai parametri contenuti in un file apposito (nel caso in esame si tratta di un'onda di *Ricker* opportunamente centrata). Dalla sezione sismica così ottenuta si preleva una sezione di controllo ($y_0 =$ costante) e se ne dà *output* su file, scegliendo i livelli di quantizzazione dell'ampiezza (nel nostro caso 256 livelli di grigio) mediante la funzione

```
Float2Gray2D(livelli_grigi,NT,NX,SEISMIC_SEC(:, :, y0),nome_file).
```

Viene quindi inizializzato il campo di velocità tramite

```
InitVelocity(NZ,NX,NY,Vfield)
```

costruendo *ad hoc* il modello geometrico di velocità oppure caricandone uno preesistente da file.

La fase successiva consiste nella valutazione del coefficiente di stabilità per il controllo della correttezza dei parametri d'ingresso; ciò comporta un *tuning* dal quale discende la determinazione dei parametri di lavoro. Se il coefficiente di stabilità è maggiore di un prefissato *target* (minore dell'unità) viene ricalcolato il Δt rispetto a quello indicato in ingresso ed il nuovo valore sostituito nel contatore dell'integratore.

La variazione del parametro temporale determina la necessità del ricalcolo, per interpolazione lineare del campo contenuto nella sezione sismica, su una griglia più fine; a ciò serve la *routine*:

```
BC_2_IC(NZ,NX,NY,NT,oldDT,DT,step,Ti,SEISMIC_SEC,P,Q)
```

che ha come variabili d'ingresso il vecchio ΔT e quello attuale, oltre che il passo temporale del contatore principale d'integrazione e la sezione sismica, mentre come uscite il campo P e la sua derivata temporale Q . Esse verranno utilizzate come condizioni iniziali nel processo di migrazione.

La *subroutine*

```
QReversing(NZ,NX,NY,SQRT(DTDZ),SQRT(DTDX),SQRT(DTDY),
.          xi,psi,eta,omega,Vfield,P,Q)
```

provvede all'inversione del segno della derivata temporale alla frontiera, laddove l'energia (valutata attraverso il segno del vettore $\mathbf{j} = \partial P \cdot \nabla P$) tenda a uscire dal volume. Tale *subroutine* al suo interno contiene le chiamate alle derivate prime secondo le tre direzioni spaziali.

Il valore del campo P e della derivata temporale $\dot{P} = Q$, oltre alla matrice d'appoggio denominata *Laplace* sono variabili d'ingresso e d'uscita per la *subroutine*

```
LeapFrog(step, NZ, NX, NY, DTDZ, DTDX, DTDY, Vfield, P, Q, Laplace)
```

che provvede all'integrazione dell'equazione d'onda attraverso un algoritmo a passi frazionari e al suo interno determina il *laplaciano* attraverso uno schema di discretizzazione spaziale basato sulla soluzione di un sistema tridiagonale. Il sistema tridiagonale deve essere risolto nelle tre direzioni spaziali attraverso il metodo di riduzione ciclica del sistema (implementato in parallelo), come descritto nel paragrafo §(2.1.3) ed utilizzando uno schema alle differenze finite compatte come descritto nel §(2.1.3).

Alla fine di ogni passo temporale di questo processo di calcolo vengono imposte le condizioni assorbenti (paragrafo §(??)) alla frontiera del dominio di calcolo, quindi viene copiata su file la sequenza temporale di una particolare sezione (ad esempio a $y_0 = \text{costante}$; tali sezioni possono essere visualizzate nella loro evoluzione mediante un *movie* MPEG) e alla fine dell'intero processo ricorsivo di migrazione, viene copiato su file l'intero volume rappresentante la sezione sismica tridimensionale migrata.

Il programma è stato eseguito su una serie di casi particolari, per effettuare dei *test* di funzionamento durante il *debugging*.

3.2 Prestazioni del solutore tridiagonale parallelo

Le prestazioni dei codici implementati secondo il modello *data parallel*, come il modellatore tridimensionale MOD3D ed il migratore all'indietro

nei tempi RTM3D, dipendono dalla combinazione di caratteristiche legate all'*hardware* ed al *software*. In sintesi, potremmo dire che, scelto un determinato algoritmo, le prestazioni del singolo nodo, le caratteristiche della rete di connessione tra i processori e le possibilità di ottimizzazione del codice offerte dal compilatore concorrono alla stima delle prestazioni su un'architettura parallela. L'analisi delle prestazioni viene condotta su

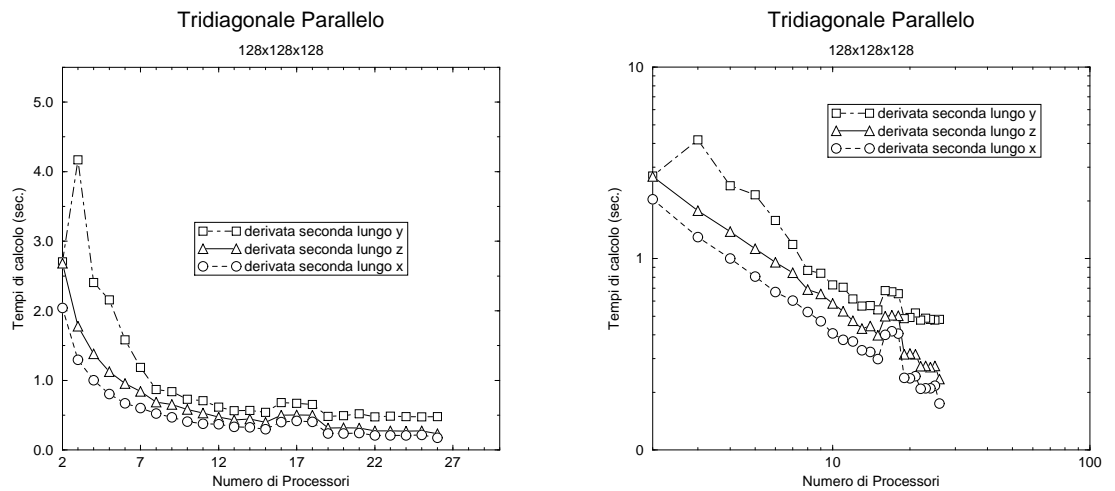


Figura 3.2: Run time.

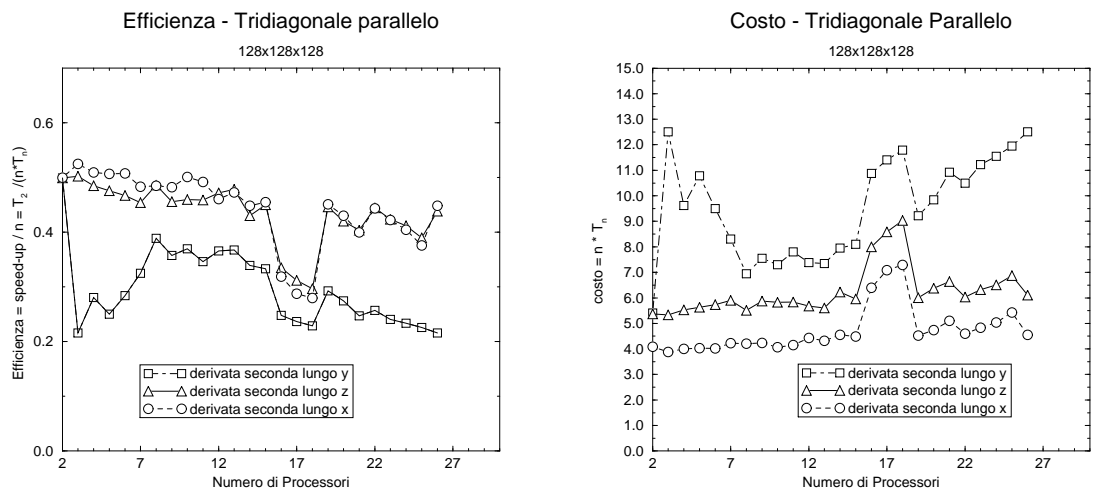


Figura 3.3: Efficienza e costo.

indicatori parametrizzati (paragrafo §(??)) rispetto al numero di operazioni

in virgola mobile per secondo (FLOPS). Tra questi indicatori vi è lo *speed up* che rende conto del guadagno tra implementazione sequenziale e parallela di uno stesso algoritmo. L'*efficienza* viene misurata in termini di percentuale di tempo impiegato da n processore per eseguire un algoritmo in modo concorrente (figura (3.3)).

Gli elementi delle matrici tridimensionali vengono assegnate alla memoria dei processori mediante le direttive di distribuzione e di allineamento dell'HPF. I dati *mappati* su un singolo nodo sono accessibili globalmente tramite la sintassi HPF ¹.

Le comunicazioni vengono rese possibili con diretto riferimento alla sintassi HPF (attraverso direttive e procedure intrinseche).

Nell'implementazione del modellatore e del migratore gli algoritmi sono stati pensati *collassando* (*) i dati relativi ai primi due assi (X e Z) su ciascun processore e distribuendo a blocchi (*BLOCK*) l'asse Y su tutti gli N_P processori:

```
!HPF$ DISTRIBUTE (*, *, BLOCK) :: P
```

Il parallelismo si esplicita consistentemente alla distribuzione dei dati. Per calcolare il *laplaciano* abbiamo bisogno di valutare le derivate seconde nelle tre direzioni spaziali. Il solutore tridiagonale è stato implementato secondo il metodo di riduzione ciclica considerando le tre direzioni (z , x , y) indipendenti. In pratica nella valutazione della derivata seconda ($\frac{\partial^2 P}{\partial z^2}$) rispetto all'asse z abbiamo N_z incognite e $N_x \times N_y$ chiamate al solutore tridiagonale:

$$T_{tot} = \frac{\chi(N_x \times N_y)}{N_P}.$$

Lo stesso discorso vale anche per $\frac{\partial^2 P}{\partial x^2}$.

Per quanto riguarda $\frac{\partial^2 P}{\partial y^2}$, le incognite sono N_y e le chiamate del solutore tridiagonale sono $N_z \times N_x$. In questo caso, poichè è necessario risolvere un sistema le cui incognite sono distribuite su tutti i processori, è presente un

¹Generalmente si può operare anche tramite *procedure locali estrinseche*, scritte in F90, compilate e collegate separatamente permettendo l'uso di librerie scientifiche ottimizzate (quali ad esempio ESSL, NAG, SGIMATH) oppure le librerie *message passing* (MPI, PVM, MPL).

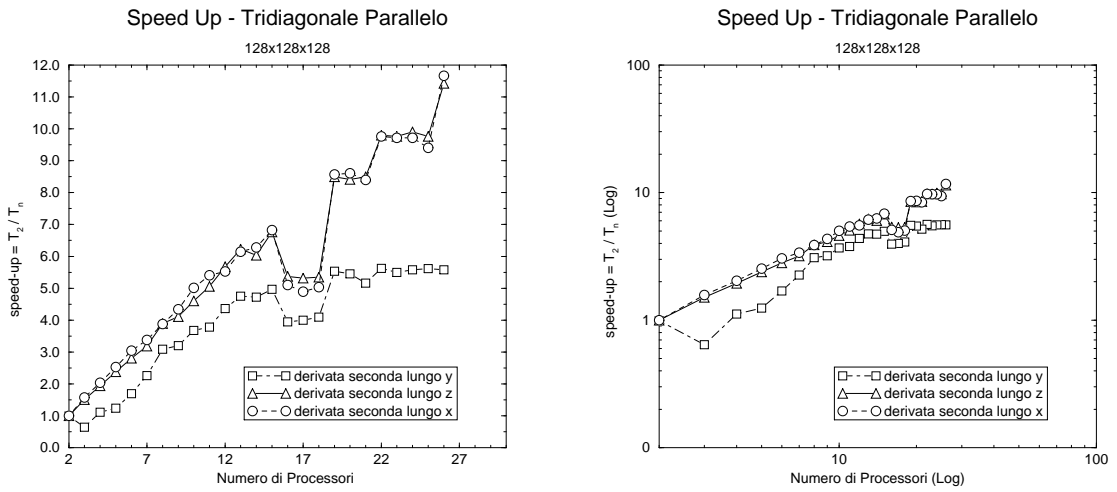
tempo addizionale dovuto alle comunicazioni inter processore. Allo scopo di minimizzare il costo computazionale sono possibili due strategie:

- si possono trasporre le matrici tridiagonali rispetto agli assi X e Y , ricordandosi di trasporre nuovamente dopo la fase elaborativa: in questo modo si concentrano le comunicazioni solo all'inizio ed alla fine del calcolo (ciò ha l'effetto di eliminare le comunicazioni all'interno del *loop* del solutore iterativo);
- si possono ricercare nuovi algoritmi che permettano un'ottimizzazione nella distribuzione dei dati, in funzione dell'algoritmo utilizzato.

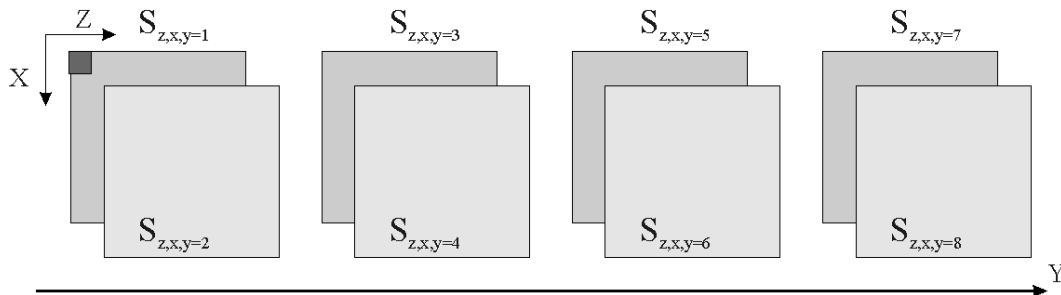
Nel calcolo della derivata seconda lungo l'asse Y , la distribuzione $(*, *, BLOCK)$ dei dati rispetto agli assi (z, x, y) comporta un *overhead* temporale dovuto alle comunicazioni; in particolare l'algoritmo tridiagonale richiede la contiguità di pochi dati su *pattern* variabili, ciò permette di ipotizzare comunicazioni frequenti dovute alla transizione di pochi dati per volta tra le memorie dei nodi, attraverso la rete di collegamento tra i processori.

Il programma è stato provato su un piccolo cubo di dati, 128 punti di lato (un volume di 16 Mbyte), su una SP2-IBM (27 nodi attivi).

La richiesta di memoria del prototipo è stata valutata attorno ai sei volumi spaziali $N_x N_y N_z$ (dovuta ai dati utili e alle matrici di appoggio) oltre ad un *surplus* del 200% (dovuto alle matrici temporanee create dal compilatore): il carico complessivo risulta pari a circa 18 volumi. Il carico può essere ottimizzato riducendo il *surplus* di memoria al solo 40%. Il grafici riportati si riferiscono allo *speed up*, al *run time*, all'*efficienza* e al *costo* relativo all'esecuzione del solutore tridiagonale parallelo in funzione del numero di processori impiegati. Si può notare che non è stato possibile effettuare la prova su singolo processore a causa dell'indisponibilità di memoria sufficiente su ogni nodo (ogni *Thin Node2* dell'SP2 del CRS4 dispone di soli 128 Mbyte di RAM), quindi le grandezze caratteristiche sono state riferite al caso di due processori ($N_P = 2$). Il tridiagonale parallelo viene calcolato sulle tre direzioni spaziali, come descritto in precedenza, considerando i dati distribuiti in modo tale da collassare su ogni processore i dati appartenenti alle superfici $S_{z,x}$. In altri termini, disponendo di 4 processori (figura (3.5)) e di

Figura 3.4: *Speed up.*

un cubo di dati di lato 8, ciò significa disporre sul processore P1 le sezioni $S_{z,x,y=1}$ e $S_{z,x,y=2}$ costituite da 64 punti ciascuna, su P2 le sezioni $S_{z,x,y=3}$ e $S_{z,x,y=4}$ della stessa dimensione e così via. Le derivate seconde calcolate

Figura 3.5: *Distribuzione dei dati sui processori: $(z, x, y) \rightarrow (*, *, BLOCK)$.*

nelle direzioni X e Z troveranno i dati utili a loro disposizione sullo stesso processore, mentre la derivata seconda nella direzione Y dovrà ottenere i dati da utilizzare nel calcolo attraverso le *comunicazioni inter-processore*. È dunque evidente come i tempi di calcolo lungo Y siano sensibilmente maggiori rispetto alle altre due direzioni (tanto maggiori quanto più piccolo è il numero di processori a causa della limitatezza della memoria su ogni processore), ma tendano ad un *overhead* stazionario all'aumentare del numero di processori (non ci sono più problemi di memoria e l'*overhead* è dovuto

alle sole comunicazioni inter-processore). Relativamente al grafico del *run time* può evidenziarsi una differenza (almeno finché non si giunge ad avere 12 processori) anche tra i tempi di calcolo relativi alle derivate seconde nelle direzioni *X* e *Z*. Ciò è dovuto ancora alla limitazione della *memoria cache dati* sul singolo nodo (*128 kbyte*) che non permette il caricamento completo dei dati utili, ma solo di blocchi di dimensione massima pari alla memoria disponibile ed inoltre alla disposizione dei dati cui si deve accedere; ciò comporta ripetute operazioni di rimpiazzamento memoria-memoria (RAM \Leftrightarrow CACHE) con conseguenti incrementi del tempo di calcolo.

3.3 Esempi di simulazione

In questo paragrafo verranno mostrati alcuni esempi di migrazione tramite il codice parallelo denominato *RTM3D*. Il primo esempio è il semplice caso di migrazione di una sezione sismica tridimensionale costituita da un'onda di *Ricker*. La migrazione di una tale sezione sismica produce una geometria semisferica dei riflettori (nel dominio tridimensionale, indietro nei tempi); in figura (3.6) vengono mostrate delle sezioni bidimensionali che fotografano la situazione a vari istanti. Supponendo di volere verificare la veridicità del modello fisico, in una sorta di esperimento concettuale, possiamo pensare di disporre delle cariche sul riflettore semisferico (ERM), ed osservando la propagazione del segnale per tempi crescenti constateremo la convergenza dei raggi di propagazione in un unico punto collocato nella posizione coincidente con la posizione spaziale dell'onda di *Ricker* utilizzata come sezione sismica iniziale.

Il secondo esempio è quello relativo alla migrazione della sezione sismica mostrata in figura (??) mediante *RTM3D*. Le quattro sezioni mostrate nelle figure (3.7) e (3.8) evidenziano le fasi finali del processo di migrazione che produce l'immagine dei riflettori in profondità. La figura (3.9) mostra l'immagine a falsi colori della sezione migrata, rappresentante l'imaging dei riflettori in profondità.

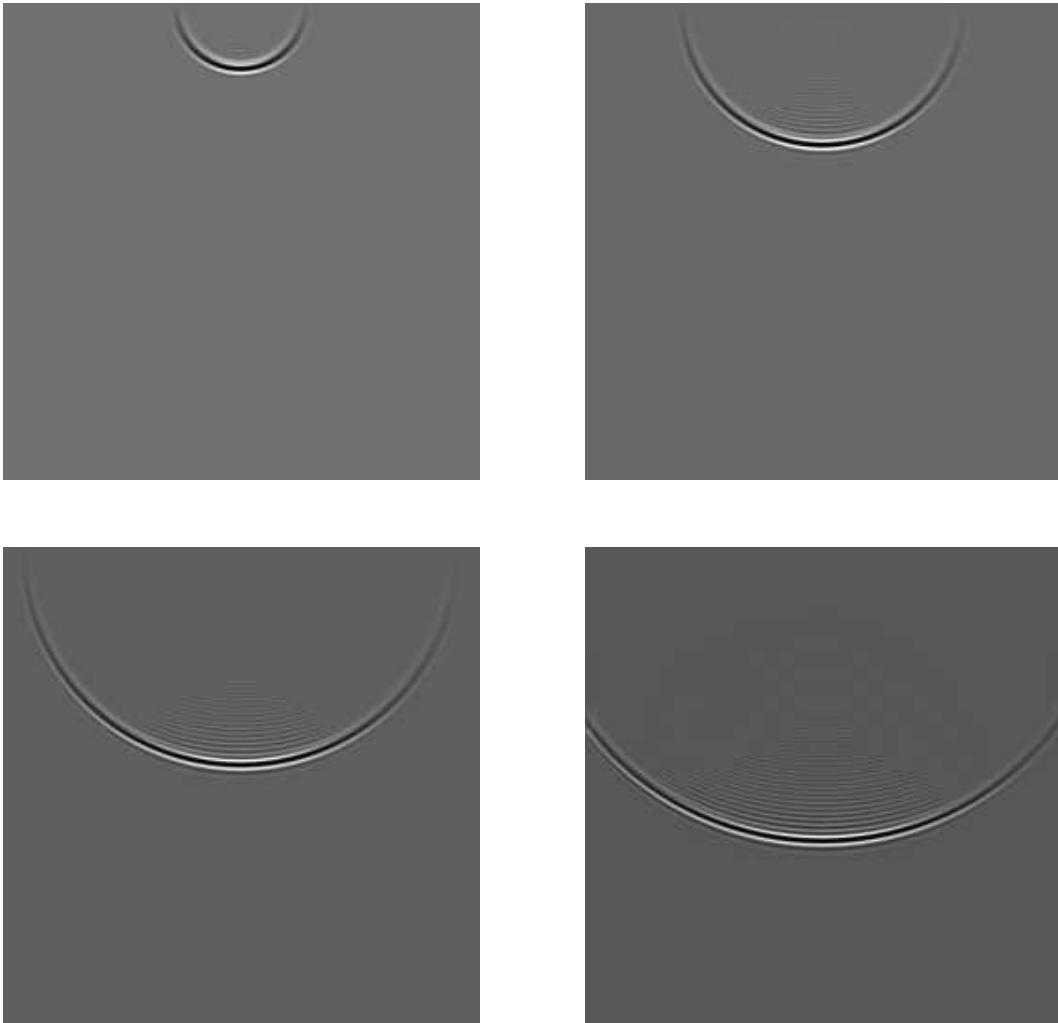


Figura 3.6: Migrazione di una sezione sismica costituita da una sorgente di Ricker.

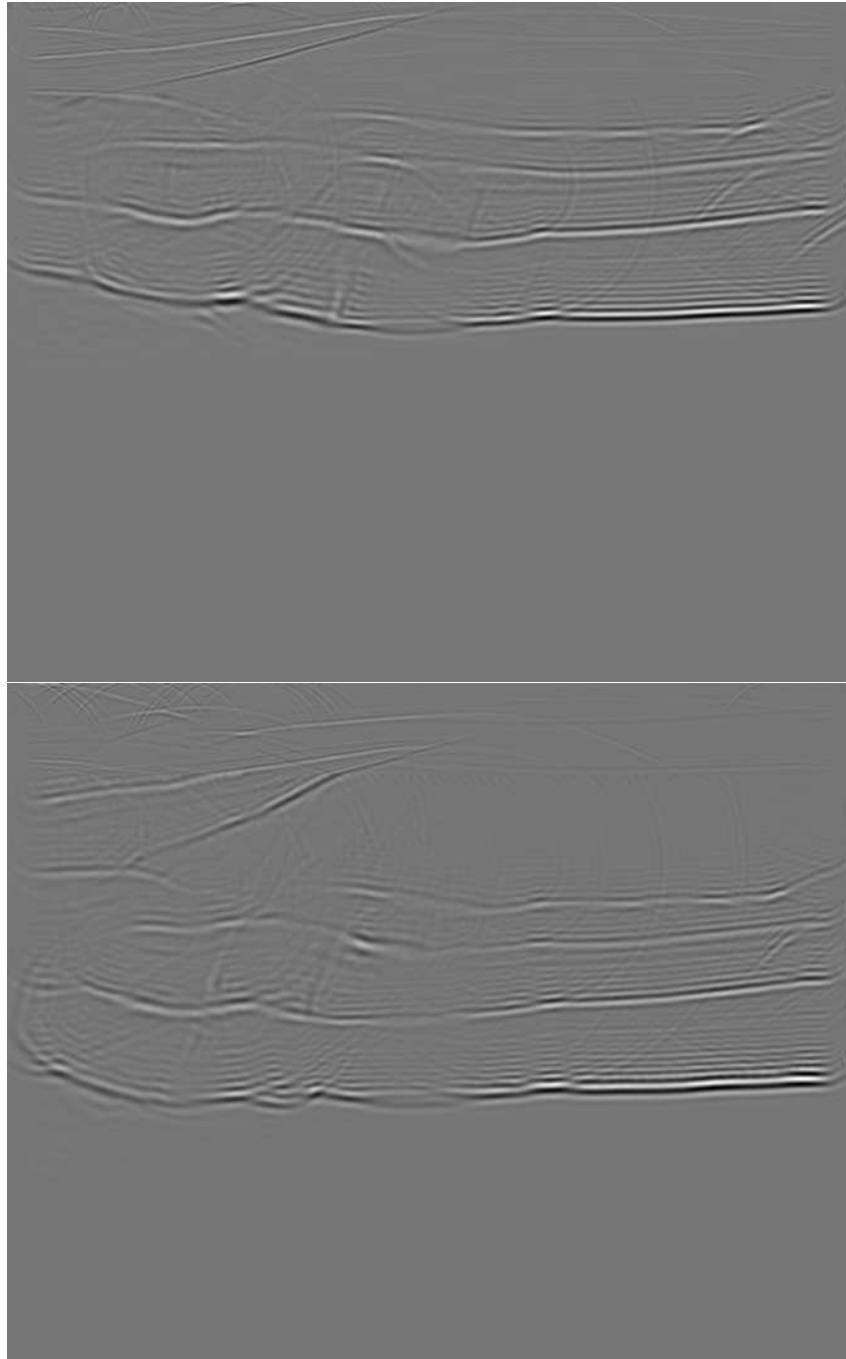


Figura 3.7: *Esempio di migrazione tramite RTM3D: le sezioni sono relative ad istanti precedenti l'imaging ($t = 0$).*

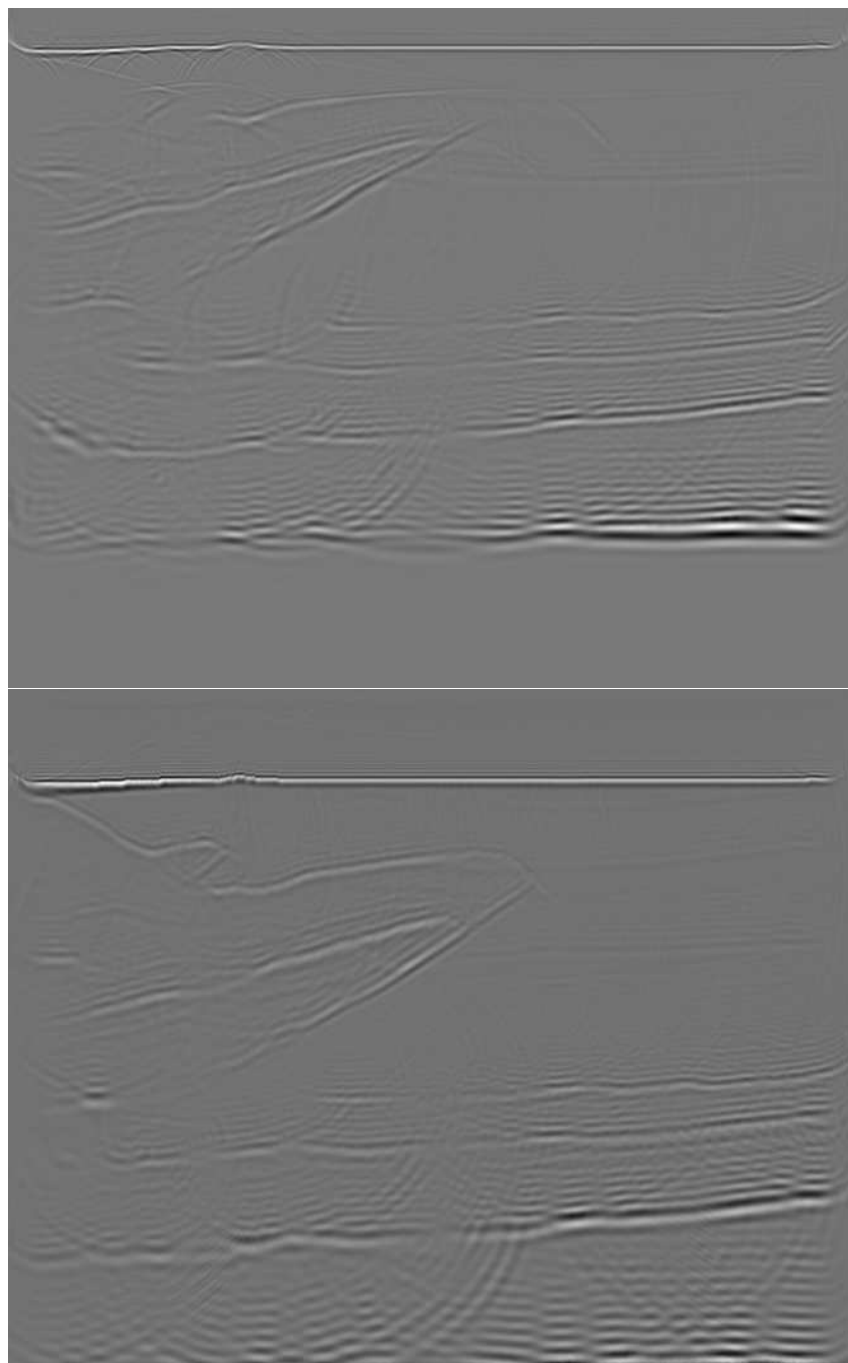


Figura 3.8: Esempio di migrazione tramite RTM3D: la sezione inferiore è quella relativa all'imaging della sezione zero-offset ($t = 0$).

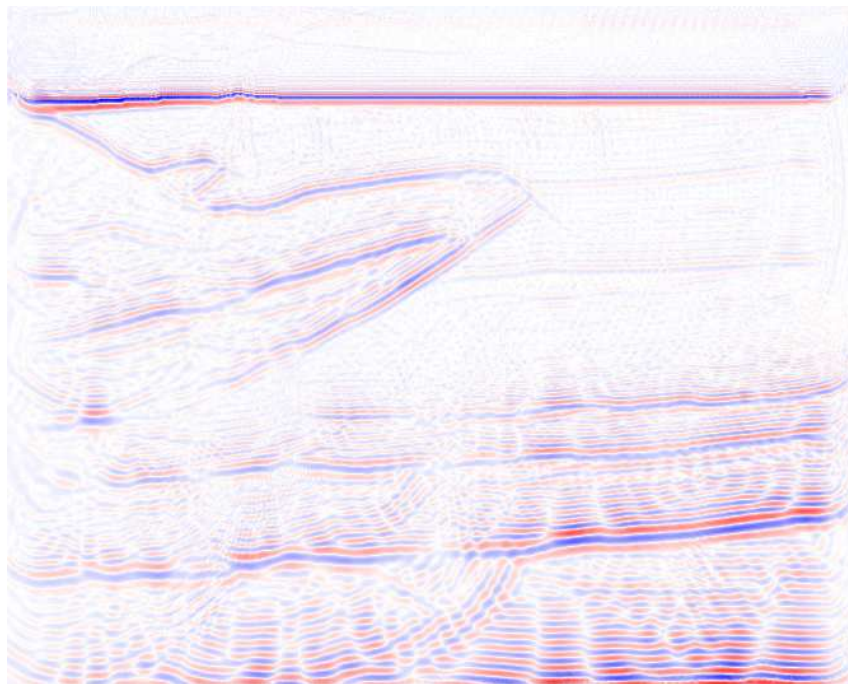


Figura 3.9: Sezione ottenuta tramite la migrazione reverse time (RTM3D) della sezione sismica indicata in figura (1.6) con campo di velocità (sezione 2D) rappresentato in figura (3.4) (AGIP).

Bibliografia

- [1] Scales J. A. *Note for an introductory course of Seismic Migration*. Center of Wave Phenomena - Colorado School of Mines, Samizdat Press, 1993.
- [2] Tikhonov A.N. Samarskij A.A. *Equazioni della fisica matematica*. MIR Publishers, Moscow, 1981.
- [3] Bagaini C. Bonomi E. Nardone C. and Pieroni E. *3D Split Convolutional Migration with Laterally Variable Velocity*. Centro di Ricerca Sviluppo e Studi Superiori in Sardegna, 1996.
- [4] Kosloff Dan D. and Baysal E. *Migration with the full acoustic wave equation*. Geophysicist, 48, 677-687, 1982.
- [5] Bertsekas D.P. and Tsitsiklis J. N. *Parallel and distributed computation, numerical methods*. Prentice Hall, 1989.
- [6] Bagaini C. Bonomi E. and Pieroni E. *3D Convolutional Migration with Laterally variable Velocity*. Centro di Ricerca Sviluppo e Studi Superiori in Sardegna, 1995.
- [7] Bagaini C. Bonomi E. and Pieroni E. *Data Parallel Implementation of 3D PSPI*. Centro di Ricerca Sviluppo e Studi Superiori in Sardegna, 1995.
- [8] Bagaini C. Bonomi E. and Pieroni E. *Split Convolutional Approach to 3D Depth Extrapolation*. Centro di Ricerca Sviluppo e Studi Superiori in Sardegna, 1995.

- [9] Bonomi E. and Cabitza G. *Migration of Seismic Data*. Centro di Ricerca Sviluppo e Studi Superiori in Sardegna Journal of Statistical Physics, 1994.
- [10] Dahlquist G. and Bjork A. *Numerical Methods*. Prentice-Hall Inc.
- [11] Claerbaut J. *Imaging the earth interior*. Blackwell Scientific Publications, 1985.
- [12] Mufti I.R. Pita J.A. and Huntley R.W. *Finite-difference depth migration of exploration-scale 3-D seismic data*. Soc. Explor. Geophysics, 1995.
- [13] Holberg O. *Towards Optimum One-Way Wave Propagation*. Geophysical Prospecting, 1988.
- [14] Stolt R.H. and Benson A.K. *Seismic Migration - Theory and Practice*. GEOPHISYCAL PRESS, 1986.
- [15] M.T.Taner F.Koehler R.E Sheriff. *Migration with the full acoustic wave equation*. Geophysicist, Vol. 44, No. 6, P. 1041-1063, June 1979.
- [16] Lele S.K. *Compact Finite Difference Schemes with Spectral-like Resolution*. Journal of Computational Physics, 103, 16-42, 1992.