

UNIVERSITÀ DEGLI STUDI ROMA TRE



Scuola Dottorale

Dipartimento Ingegneria Meccanica ed Industriale

Ciclo XXV

**Schema innovativo per la gestione
multilivello di modelli
approssimati in ottimizzazione
globale.**

Candidato

Simone Menicucci

Tutor

Prof. **Umberto Iemma**

Coordinatore

Prof. **Edoardo Bemporad**

Aprile 2013

Vi sono cervelli atti più alla loquacità e ostentazione che alla specolazione e alla investigazione delle opere della natura, li quali prima di ridursi a proferir quella savia, ingenua e modesta parola “non lo so” scorrono a lasciarsi uscir di bocca ed anco dalla penna qualsivoglia grande esorbitanza.

(Galileo Galilei, per bocca di F. Salviati)

Indice

Introduzione	1
1 L'ottimo nella progettazione multidisciplinare	4
1.1 L'ottimizzazione multidisciplinare	4
1.1.1 Storia	5
1.1.2 L'ottimizzazione strutturale	7
1.2 Il problema di minimo vincolato	8
1.3 La scelta della funzione obiettivo	9
1.4 La definizione dei vincoli	9
1.5 Metodi di minimizzazione	9
2 Metodi di Ottimizzazione	11
2.1 Linear Programming	11
2.2 Ottimizzazione Numerica Non Lineare	12
2.2.1 Ottimizzazione Numerica Non Lineare Locale	12
2.2.2 Ottimizzazione Numerica Non Lineare Globale	14
3 Stato dell'arte AMMO	19
3.1 Idea di base	20
3.1.1 Condizioni di consistenza	21
3.2 β correlation	21

INDICE

3.3	Parametro di merito	22
3.4	Aggiornamento della <i>trust region</i> e criterio di convergenza . . .	22
3.5	Diagramma di flusso	23
4	L'idea Multilivello	24
4.1	Primo modello approssimato	25
4.2	Surrogato del modello approssimato	25
4.3	Aggiornamento di Δ_{int}	26
4.4	Aggiornamento di Δ	26
4.5	L'agoritmo MAMMO	27
5	Risultati	29
5.1	Numero chiamate funzione ad alta fedeltà	
	MAMMO <i>versus</i> AMMO	31
5.1.1	Funzione monodimensionale monomodale	32
5.1.2	Funzioni bidimensionali monomodali	36
5.1.3	Funzioni bidimensionali multimodali	43
5.2	Dove AMMO fallisce	47
6	Considerazioni sui tempi	54
6.1	Considerazioni su ulteriori potenzialità del metodo MAMMO .	61
7	Conclusioni	63
	Bibliografia	65
	Ringraziamenti	66

Elenco delle figure

1.1	Progettazione tradizionale <i>versus</i> MDO	5
1.2	BWB sviluppato da Roma Tre nel progetto OpenAir, sulla base del progetto ACFA2020	6
3.1	AMMO - Diagramma di flusso.	23
4.1	MAMMO - diagramma di flusso	28
5.1	$f(x) = x3^x$	32
5.2	$f(x, y) = x3^x + 5y^2$	36
5.3	Iterazioni metodo AMMO	37
5.4	Iterazioni metodo MAMMO	39
5.5	$f(x, y) = -\frac{1}{\cosh(x)\cosh(y)}$	40
5.6	AMMO e MAMMO	41
5.7	$f(x, y) = \frac{x^4}{4} - \frac{x^2}{2} + \frac{x}{10} + \frac{y^2}{2}$	43
5.8	AMMO e MAMMO	44
5.9	Funzione di Rosenbrock: $f(x, y) = (1 - x)^2 + 100(y - x^2)^2$	45
5.10	AMMO e MAMMO per Rosenbrock	46
5.11	$f(x, y) = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2$	47
5.12	AMMO al variare del punto iniziale	48
5.13	Iterazioni metodo MAMMO	50

ELENCO DELLE FIGURE

5.14	$f(x, y) = \sum_{i=1}^5 (i \times \cos((i+1)x + i) + \sum_{i=1}^5 (i \times \cos((i+1)y + i) + 0.5((x + 1.42513)^2 + (y + 0.8)^2))$	51
5.15	AMMO - minimo locale	52
5.16	MAMMO: minimo globale	53
6.1	Rapporto tempi di esecuzione per $f(x, y) = x3^x + 5y^2$	57
6.2	Rapporto tempi in ottimizzazione diretta, AMMO e MAMMO	59
6.3	Zoom figura 6.2a	60
6.4	Rapporto tempi tra modelli alta, media e bassa fedeltà	61

Introduzione

Il lavoro presentato è incentrato sull'utilizzo di metodi per la gestione di modelli multifedeltà per la progettazione ottima.

L'interesse per tali metodi è giustificato dal fatto che molti fenomeni fisici, nella progettazione ingegneristica, possono essere simulati da modelli computazionali tanto accurati quanto costosi; tali modelli possono essere chiamati anche centinaia di volte in un processo di ottimizzazione.

Un metodo di gestione dell'ottimizzazione che riduce il costo computazionale associato all'utilizzo di modelli ad *alta fedeltà* risulta di estrema importanza quando è significativa la riduzione della capacità computazionale richiesta.

Il presente lavoro verrà introdotto con una panoramica sull'ottimizzazione multidisciplinare (*Multidisciplinary Design Optimization o MDO*), cioè quella branca dell'ingegneria in cui vengono utilizzati i metodi di ottimizzazione per risolvere problemi di progettazione che coinvolgono varie discipline, quali ad esempio la fluidodinamica, l'analisi strutturale e/o la propulsione. Sarà mostrato come un problema di ottimizzazione altro non è che un problema di minimo vincolato in cui riconoscere la *funzione obiettivo*, i *vincoli* di uguaglianza e quelli di disuguaglianza.

Nel capitolo successivo verrà presentata una carrelata dei metodi maggiormente utilizzati in tale disciplina, sottolineando la differenza tra metodi di ottimizzazione locale e metodi di ottimizzazione globale e verrà indicato il

metodo scelto per portare avanti il presente lavoro.

Nel terzo capitolo sarà illustrato lo stato dell'arte per quanto riguarda l'ottimizzazione multi-fedeltà, in cui gli "attori" che entrano in gioco sono una funzione f_{HI} che descrive in modo accurato un certo fenomeno fisico e che verrà denominata ad "alta fedeltà", un'altra funzione f_{LO} di accuratezza minore della precedente e che, quindi, verrà chiamata a "bassa fedeltà", una funzione approssimata a_c combinazione delle precedenti, una regione Δ , detta "di confidenza", in cui le funzioni f_{HI} e a_c hanno lo stesso andamento di salita-discesa, e un parametro ρ , denominato "di merito", utile per quantificare l'errore che viene commesso utilizzando la funzione a_c anziché la f_{HI} .

Concluso il capitolo con il diagramma di flusso indicante l'algoritmo utilizzato, si porrà l'attenzione sulla nuova idea di ottimizzazione multi-fedeltà, definita "multi-livello", perchè non si avrà a che fare con soli due livelli di accuratezza, *alta* e *bassa*, bensì con tre, *alta*, *bassa* e *media*. Questo aumento di modelli porterà alla creazione di due funzioni approssimate a_{hm} e a_{sl} , oltre alla gestione di due *trust region*, Δ e Δ_{int} e al calcolo di due parametri di merito ρ_{int} e ρ_{ext} . Anche questo capitolo terminerà con un diagramma di flusso riassuntivo il nuovo algoritmo.

Verrà mostrato quali sono i risultati ottenuti e come tale approccio comporti un netto miglioramento per quanto riguarda il numero di chiamate al modello ad alta fedeltà; questo perchè, classicamente, nella letteratura disponibile su tale argomento, quando si parla di tempi di calcolo si fa sempre riferimento al numero di chiamate al modello a più alta fedeltà, dato che è lecito assumere che sia quello i cui tempi di soluzione siano di gran lunga superiori agli altri. Per tutti i *test case* analizzati verranno indicate non solo le chiamate al modello più accurato, ma anche quelle agli altri modelli, per analizzare quando sia opportuno utilizzare una metodologia, piuttosto che un'altra, piuttosto

che nessuna delle due. Verranno, inoltre, illustrati due *test case* in cui la metodologia oggetto del presente lavoro sembra migliorare quella classica non solo dal punto di vista delle chiamate, ma anche rispetto al risultato trovato; infatti in caso di funzioni multimodali, la metodologia classica riesce a trovare il minimo locale più vicino al punto di partenza, mentre il metodo presentato converge sul minimo globale.

Prima di concludere con i possibili sviluppi futuri dello studio in oggetto, un capitolo verrà dedicato ad alcune osservazioni sui tempi di calcolo e a considerazioni su quando, nei vari casi analizzati, sia consigliabile l'utilizzo di un metodo multi-fedeltà con approccio classico anziché una ottimizzazione diretta, oppure un metodo multi-fedeltà multi-livello anziché una ottimizzazione diretta, oppure un metodo multi-fedeltà multi-livello piuttosto che un metodo multi-fedeltà classico. Un paragrafo verrà infine dedicato a mostrare quali possano essere altre motivazioni, oltre a quella del guadagno in termini di tempi di calcolo, che possano far preferire la metodologia in esame a quella classica.

Capitolo 1

L'ottimo nella progettazione multidisciplinare

In questo capitolo sarà affrontato il problema della progettazione ottima multidisciplinare in campo aeronautico. Vedremo come tale problema possa essere ridotto alla formulazione di un problema matematico di minimo vincolato, cosa intenderemo per progettazione ottima multidisciplinare e le possibili scelte del progettista legate alla ricerca dell'ottimo.

Sarà altresì presentato un breve riassunto storico di tale disciplina.

1.1 L'ottimizzazione multidisciplinare

L'ottimizzazione multidisciplinare (*MDO*) è un campo dell'ingegneria in cui vengono utilizzati metodi di ottimizzazione per risolvere problemi di progettazione che coinvolgono più discipline. L'ottimo trovato considerando tutte le discipline simultaneamente è migliore di quello trovato ottimizzando ogni singola disciplina singolarmente in quanto viene tenuto conto dell'interazione tra le stesse. Tuttavia includere tutte le discipline simultaneamente incrementa

significativamente la complessità del problema.

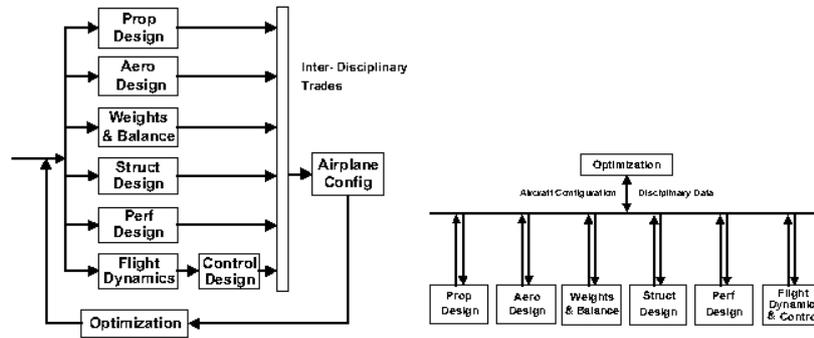


Figura 1.1: Progettazione tradizionale *versus* MDO

1.1.1 Storia

L'ottimizzazione viene utilizzata in numerosi campi, dalla progettazione di automobili all'architettura navale, dall'elettronica a problemi di distribuzione dell'elettricità, ma è nel campo dell'ingegneria aerospaziale che si è affermata maggiormente. Tutti gli studi connessi con tale disciplina hanno avuto un incremento notevole quando è aumentato l'interesse commerciale verso velivoli dalla geometria non convenzionale, quale il *Blended Wing Body (BWB)* indicato in figura 1.2. Nella fase di progettazione concettuale di tale velivolo l'ottimizzazione rappresenta senz'altro una delle fasi più rilevanti. Le discipline considerate in tale progettazione sono l'aerodinamica, l'analisi strutturale, la propulsione, la teoria del controllo, l'economia e l'aeroacustica.

Tradizionalmente la progettazione ingegneristica viene svolta da gruppi di lavoro, ognuno esperto in discipline specifiche, come ad esempio analisi strutturale e aerodinamica. Ogni gruppo di lavoro eredita sequenzialmente delle

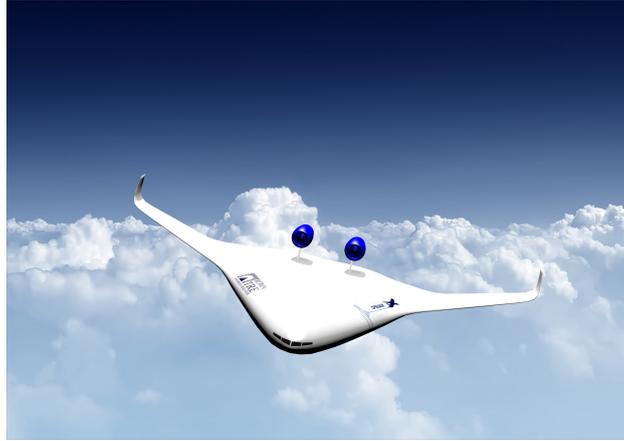


Figura 1.2: BWB sviluppato da Roma Tre nel progetto OpenAir, sulla base del progetto ACFA2020

specifiche che vengono imposte dagli altri; per esempio il gruppo aerodinamico potrebbe imporre al gruppo di strutturisti il rispetto della forma del corpo oggetto della progettazione.

Tra gli anni '70 e '90, due maggiori sviluppi dell'industria aeronautica cambiarono l'approccio dei progettisti verso i loro problemi di progettazione. Il primo fu il *computer-aided design* (CAD), che consentì ai progettisti rapide modifiche ed analisi dei loro progetti. Il secondo fu il cambiamento nella politica di progettazione delle compagnie aeree, delle organizzazioni militari (particolarmente delle organizzazioni militari degli Stati Uniti d'America), che si incentrò sul costo totale di vita di un progetto (*Total life cycle cost*). Questo ha portato ad una maggiore attenzione sui fattori economici che rientrano nei campi noti come: producibilità, realizzabilità, manutenibilità ecc. Dal 1990, la tecnica si è espansa alle altre industrie, i personal computer ad alte prestazioni sono stati sostituiti da supercomputer, internet e le reti locali hanno facilitato la condivisione di informazioni progettuali. Software di progettazione (come ad esempio NASTRAN, *NA*sa *STR*uctural *AN*alysis) sono

divenuti più maturi, come sono diventati più efficienti alcuni algoritmi di ottimizzazione (soprattutto quelli non basati sul gradiente, detti “*euristici*”).

1.1.2 L'ottimizzazione strutturale

I metodi di ottimizzazione sono vecchi quasi quanto il calcolo, difatti Isaac Newton, Leonhard Euler, Daniel Bernoulli, Joseph Lous Lagrange, usarono tali metodi per risolvere problemi come per esempio quello della forma della curva catenaria; l'ottimizzazione numerica, intesa come disciplina a sè stante, è divenuta importante nell'era digitale. La sua sistematica applicazione per la progettazine strutturale inizia con Schmit nel 1960. Il successo dell'ottimizzazione strutturale negli anni '70 motivò l'emergere dell'ottimizzazione multidisciplinare nel 1980. Jaroslaw Sobieski campionò metodi di decomposizione specifici per applicazione MDO. Nell'ottimizzazione strutturale si possono distinguere due classi di metodologie: quelle basate sul gradiente (*e.g.* metodo di Newton, metodo del Gradiente Coniugato) e quelle non basate sul gradiente (*e.g.* algoritmi genetici, algoritmi a sciame *Particle Swarme Optimization*). La combinazione di criteri di ottimo, storicamente utilizzati in problemi strutturali di minimizzazine vincolata¹ del peso della struttura, con criteri di programmazione matematica, i quali impiegano metodi classici basati sul gradiente (come tecniche di minimizzazione non vincolata, di proiezione del gradiente di Rosen, di programmazione sequenziale lineare ed eventualmente quadratica), hanno reso altamente efficienti i metodi basati sul gradiente. Recentemente, invece, sono stati sviluppati metodi di decomposizione, metodi di approssimazione, algoritmi evolutivi, metodologie delle

¹I vincoli venivano in genere posti sugli sforzi massimi, buckling, frequenze proprie di vibrazione

superfici di risposta, che rientrano nella classe dei cosiddetti metodi non basati sul gradiente.

1.2 Il problema di minimo vincolato

La formulazione matematica di un problema di ottimizzazione altro non è che un problema di ricerca vincolata del minimo di una funzione, esprimibile come:

$$\begin{array}{ll} \text{trova minimo di} & f(x), x \in D \\ \text{soggetto a} & g_n(x) < 0, n = 1, \dots, N \\ \text{ed a} & h_m(x) = 0, m = 1, \dots, M \end{array}$$

in cui x rappresenta il vettore contenente le variabili di progetto, f è la funzione che deve essere minimizzata (**funzione obiettivo**), g_n sono i vincoli di disuguaglianza e h_m sono i vincoli di uguaglianza.

Diremo che f ha un minimo nel punto $\mathbf{x} = \mathbf{x}_0$ in un dominio R (dove f è definita) se $f(\mathbf{x}) > f(\mathbf{x}_0)$ per ogni \mathbf{x} in R , o che f ha un punto di minimo relativo in \mathbf{x}_0 se $f(\mathbf{x}) > f(\mathbf{x}_0)$ per ogni \mathbf{x} appartenente ad un intorno ammissibile di \mathbf{x}_0 tale che

$$|\mathbf{x} - \mathbf{x}_0| < r \tag{1.1}$$

con $r > 0$ sufficientemente piccolo.

Se f ha un estremo in \mathbf{x}_0 allora

$$\nabla f(\mathbf{x}_0) = \mathbf{0} \tag{1.2}$$

ed \mathbf{x}_0 è un punto stazionario.

1.3 La scelta della funzione obiettivo

Come detto in precedenza, la funzione f è l'oggetto o obiettivo del processo di ottimizzazione del progetto. Questa può essere definita come una grandezza di interesse per il velivolo (per es. peso a vuoto, consumo carburante) o come una combinazione lineare di più grandezze di interesse:

$$f(\mathbf{x}) := \sum_{k=1}^K \eta_k f_k(\mathbf{x}) \quad (1.3)$$

I coefficienti η_k sono arbitrariamente scelti dal progettista per pesare le funzioni di interesse, all'interno della funzione obiettivo.

1.4 La definizione dei vincoli

I vincoli in campo aeronautico che la configurazione finale deve rispettare riguardano le variabili significative (per esempio lo sforzo normale σ , la velocità di flutter u_f , la derivata del coefficiente di momento² rispetto all'angolo di attacco di fusoliera C_{M_α} , ecc.) delle varie discipline (costruzione aeronautiche, aeroelasticità, meccanica del volo, ecc.) prese in considerazione nel problema di ottimizzazione.

1.5 Metodi di minimizzazione

Come già accennato gli algoritmi di ottimizzazione si dividono in due grandi famiglie, quelli basati sul gradiente e quelli euristici o “*gradient-free*”. Nel prossimo capitolo saranno descritti alcuni dei metodi maggiormente utilizzati

²Usando come polo il centro di gravità G

a seconda del problema da risolvere. Intanto è utile sottolineare che la principale differenza tra le due famiglie è che i metodi basati sul gradiente vengono utilizzati per ottimizzazione locale, mentre i metodi euristici vengono utilizzati nell'ambito di una ottimizzazione globale. Dato che, come vedremo, il presente lavoro si basa sul fatto che all'interno di una determinata regione dello spazio deve essere trovato il minimo globale di una determinata funzione, in tutti i casi che verranno presentati viene utilizzato un metodo euristico e per l'esattezza il "metodo Nelder-Mead" illustrato nel paragrafo 2.2.2.

Capitolo 2

Metodi di Ottimizzazione

I metodi utilizzati per risolvere problemi di ottimizzazione locali e globali dipendono fortemente dal tipo di problema. I problemi di ottimizzazione possono essere classificati in base a molti criteri. Sulla base del tipo di funzioni coinvolte ci possono essere problemi di ottimizzazione lineari o non lineari (polinomiali, algebrici, trascendentali ecc. ecc.).

2.1 Linear Programming

I problemi di *linear programming* sono problemi di ottimizzazione in cui la funzione obiettivo e i vincoli sono tutti lineari. Un esempio di problema che può essere risolto con *linear programming* è:

$$\begin{array}{ll} \text{trova minimo di} & x + 2y \\ \text{soggetto a} & -5x + y = 7 \\ & x + y \geq 26 \\ & x \geq 3, y \geq 4 \end{array}$$

o, in forma più generale:

$$\begin{array}{ll} \text{trova minimo di} & c^T x \\ \text{soggetto a} & A_1 x = b_1 \\ & A_2 x = b_2 \\ & l \leq x \leq u \end{array}$$

Possibili algoritmi di risoluzione della programmazione lineare sono:

- metodo del simplesso;
- metodo del simplesso rivisitato;
- metodo del punto interno.

Come facilmente intuibile, in un problema reale di ottimizzazione avere a che fare con un caso di programmazione lineare è alquanto improbabile, quindi tali metodi non verranno approfonditi. Lo scopo del presente capitolo è quello di fornire una panoramica più completa possibile e non di entrare nello specifico di tutti i metodi di ottimizzazione.

2.2 Ottimizzazione Numerica Non Lineare

Gli algoritmi numerici per l'ottimizzazione vincolata non lineare possono essere ampiamente suddivisi in due categorie:

- metodi basati sul gradiente;
- metodi di ricerca diretta.

2.2.1 Ottimizzazione Numerica Non Lineare Locale

I metodi basati sul gradiente utilizzano informazioni sulle derivate prime (gradiente) e sulle derivate seconde (hessiana) e riescono a trovare un minimo locale della funzione obiettivo. Esempi di questo tipo di algoritmi sono:

- metodo SQP (*Sequential Quadratic Programming*);
- metodo di Lagrange esteso;
- metodo del punto interno non lineare.

Metodo SQP

Utilizzando il metodo SQP si ottiene la direzione secondo la quale la funzione obiettivo sarà minimizzata, mediante la soluzione di una sequenza di sottoproblemi QP (*Quadratic Programming*).

Ogni sottoproblema QP minimizza un modello quadratico di una certa funzione Lagrangiana soggetta a linearizzazione dei suoi vincoli. Una funzione di merito è introdotta per assicurare la convergenza a partire da un qualsivoglia punto iniziale. Il primo passo che viene effettuato è quello di convertire tutti i vincoli presenti in vincoli di uguaglianza introducendo una variabile s (*slack*) in modo che:

$$\mathbf{A}_L \mathbf{x} - \mathbf{s}_L = \mathbf{0} \quad (2.1)$$

$$\mathbf{f}(\mathbf{x}) - \mathbf{s}_N = \mathbf{0} \quad (2.2)$$

dove $\mathbf{A}_L \mathbf{x}$ sono vincoli lineari, $\mathbf{f}(\mathbf{x})$ sono i vincoli non lineari e \mathbf{s}_L , \mathbf{s}_N le variabili (*slack*) rispettivamente lineari e non lineari.

A questo punto, il metodo SQP coinvolge due tipi di iterazioni:

- *Major*
- *Minor*

che opportunamente combinati, garantiscono la convergenza della funzione obiettivo alla voluta condizione di ottimo.

Metodo di Lagrange esteso

Al fine di trovare il minimo della funzione obiettivo soggetta ai vincoli di uguaglianza e disuguaglianza, viene utilizzata una funzione di merito, combinazione della funzione obiettivo e dei vincoli stessi. Nell'utilizzare questo metodo, viene definita una funzione di merito:

$$\phi(x, \beta) = f(x) - \mu \sum_i \ln(x_i) - h(x)^T \lambda + \beta \|h(x)\|^2 \quad (2.3)$$

in cui $f(x)$ rappresenta la funzione obiettivo, $h(x)$ i vincoli (trasformati tutti in vincoli di uguaglianza), $\mu > 0$ è un parametro detto di “barriera” e $\beta > 0$ è un parametro di “penalità”.

Metodo del punto interno

Tale algoritmo risolve un problema di ottimizzazione vincolata combinando la funzione obiettivo ed i vincoli attraverso l'uso della funzione “barriera”. Esattamente, il problema viene convertito nella forma standard:

$$\begin{aligned} & \text{Minimo di } f(x) \\ & \text{soggetto a } h(x) = 0, x \geq 0 \end{aligned}$$

dopodichè i vincoli non negativi vengono sostituiti aggiungendo alla funzione obiettivo un termine, chiamato *barriera*:

$$\begin{aligned} & \text{Minimo di } f(x) - \mu \sum_i \ln(x_i) \\ & \text{soggetto a } h(x) = 0 \end{aligned}$$

in cui μ è il parametro barriera.

2.2.2 Ottimizzazione Numerica Non Lineare Globale

I metodi di ricerca diretta non utilizzano informazioni sulle derivate ed hanno l'enorme vantaggio rispetto ai precedenti di convergere al minimo globale della funzione obiettivo. Alcuni dei metodi in questione sono:

- metodo Nelder-Mead;
- algoritmi genetici ed evolutivi;
- *simulated annealing*.

Questi metodi hanno lo svantaggio di convergere molto più lentamente rispetto a quelli basati sul gradiente e quindi di avere dei tempi computazionali maggiori.

Metodo Nelder-Mead

Il metodo Nelder-Mead è un metodo di ricerca diretta. Per una funzione di n variabili, l'algoritmo consiste nello scegliere $n + 1$ punti rappresentanti un politopo nello spazio n -dimensionale (un politopo n -dimensionale è il corrispondente di un poligono nel piano e di un poliedro nello spazio generalizzato ad uno spazio euclideo reale \mathbb{R}^n).

Ad ogni iterazione, gli $n + 1$ punti formano un politopo. I punti sono ordinati in modo che $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$. Viene generato un nuovo punto che sostituisca il peggiore dei precedenti x_{n+1} . Sia c il centroide del politopo creato dagli n punti migliori:

$$c = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.4)$$

Utilizzando c , viene generato un punto di prova x_t che sostituisca il punto peggiore:

$$x_t = c + \alpha(c - x_{n+1}) \quad (2.5)$$

in cui $\alpha > 0$ è un parametro.

Se il nuovo punto x_t non è nè il punto migliore, nè il peggiore, $f(x_1) \leq f(x_t) \leq f(x_n)$, allora x_t sostituisce x_{n+1} .

Se il nuovo punto x_t è il punto migliore, $f(x_t) \leq f(x_1)$, viene calcolato un

altro punto $x_e = c + \beta(c - x_{n+1})$, in cui $\beta > 1$ è un parametro per espandere il politopo. Se $f(x_e) \leq f(x_t)$, allora $f(x_{n+1})$ viene sostituita da x_e , altrimenti da x_t . Se il nuovo punto x_t è il secondo punto peggiore, $f(x_t) \geq f(x_n)$ il politopo viene considerato troppo grande e necessita di essere contratto. E' definito un nuovo punto di prova:

$$x_c = \begin{cases} c + \gamma(x_{n+1} - c), & \text{se } f(x_t) \geq f(x_{n+1}), \\ c + \gamma(x_t - c), & \text{se } f(x_t) < f(x_{n+1}), \end{cases}$$

in cui $0 < \gamma < 1$ rappresenta un parametro. Se $f(x_c) < \text{Min}(f(x_{n+1}), f(x_t))$, allora x_c sostituisce x_{n+1} , altrimenti viene effettuata una nuova contrazione. Si assume che il processo giunga a convergenza quando la differenza tra il miglior valore nel nuovo e vecchio politopo, così come la distanza tra il nuovo miglior punto ed il vecchio, sono minori di due parametri di tolleranza precedentemente fissati.

Algoritmo evolutivo

L'algoritmo consiste nel creare una popolazione di m punti, $\{x_1, x_2, \dots, x_j, \dots, x_m\}$, con $m \gg n$, in cui n rappresenta il numero di variabili. Durante ogni iterazione dell'algoritmo, viene generata una nuova popolazione di m punti. Il j -esimo nuovo punto è generato scegliendo tre punti casuali della vecchia popolazione, x_u, x_v e x_w e creando un punto $x_s = x_w + s(x_u - x_v)$, in cui s rappresenta un fattore di scala reale scelto dal progettista. A questo punto viene costruito un nuovo punto x_{new} utilizzando x_j e x_s scegliendo la i -esima coordinata da x_s con una certa probabilità ϱ o da x_j altrimenti. La probabilità ϱ è ovviamente scelta dal progettista.

Se $f(x_{new}) < f(x_j)$, allora x_{new} sostituisce x_j nella nuova popolazione. Come nel caso precedente il processo si considera giunto a convergenza se la

differenza tra il valore minore della vecchia popolazione e quello della nuova è inferiore ad un certo parametro fissato inizialmente.

Il presente metodo è costoso dal punto di vista computazionale, ma è molto robusto e fornisce risultati affidabili quando il problema oggetto di studio ha molti minimi.

Simulated Annealing

L'idea di questo metodo si basa sul processo fisico della *ricottura*, in cui un oggetto di metallo viene portato ad una temperatura elevatissima e poi fatto raffreddare lentamente. Questo processo permette alla struttura atomica del metallo di ricomporsi divenendo un metallo più resistente. In termini di ottimizzazione si potrebbe affermare che la *ricottura* permette alla struttura di "uscire" da un minimo locale e di trovare un minimo globale.

Ad ogni iterazione è generato un nuovo punto x_{new} nell'intorno del punto corrente x . Il raggio dell'intorno decresce ad ogni iterazione. Trovato il punto x_{best} , se $f(x_{new}) \leq f(x_{best})$, allora x_{new} sostituisce x_{best} e x . Altrimenti x_{new} sostituisce x con una probabilità pari a $e^{b(i, \Delta f, f_0)}$ in cui b rappresenta una funzione definita come *Esponente di Boltzmann*, i è l'iterazione corrente, Δf è la variazione del valore della funzione obiettivo e f_0 è il valore della funzione obiettivo all'iterazione precedente. Solitamente il valore di b è:

$$b = \frac{-\Delta f \log(i + 1)}{10}$$

Il metodo *Simulated Annealing* utilizza molteplici punti di partenza e trova un minimo partendo da ognuno di essi. Il numero di punti di partenza è dato da $\min(2d, 50)$ in cui d rappresenta il numero di variabili. Per ogni punto di partenza, l'algoritmo viene ripetuto fino a quando non si determina una delle situazioni seguenti:

- viene raggiunto il numero massimo di iterazioni;
- il metodo converge ad un punto;
- il metodo trova lo stesso punto per un numero di iterazioni fissato a priori.

Capitolo 3

Stato dell'arte AMMO

Nella progettazione ottima in aeronautica un concetto di forte interesse è quello sull'utilizzo dei metodi di gestione di modelli multi-fedeltà. Questo perché molti fenomeni fisici, nella progettazione ingegneristica, possono essere simulati da modelli computazionali tanto accurati, quanto costosi dal punto di vista dei tempi di calcolo. Tali modelli possono essere richiamati centinaia di volte in un processo di ottimizzazione. Un metodo di gestione dell'ottimizzazione che riduce il costo computazionale associato all'utilizzo di modelli ad alta fedeltà risulta di estrema importanza quando è significativa la riduzione della capacità computazionale richiesta.

Fino a pochi anni il processo di gestione di modelli a fedeltà variabile era portato avanti in maniera euristica (*trial and error*), mentre nell'ultimo decennio si è focalizzata l'attenzione su come poter utilizzare una metodologia rigorosa di ricorso ai modelli a bassa fedeltà, durante il processo di ottimizzazione di un modello ad alta fedeltà. La metodologia maggiormente sviluppata in letteratura è l'Approximation Model Management Optimization (AMMO) che permette la soluzione di problemi di ottimizzazione di modelli ad alta fedeltà riducendo il numero di chiamate al modello stesso nel processo di

ottimizzazione. Tale approccio combina l'utilizzo di modelli approssimati, con metodi numerici basati sul metodo della “*trust region*”, in quanto l'uso di modelli approssimati permette di ridurre il costo computazionale di una ottimizzazione, mentre l'utilizzo di una “*trust region*” consente di gestire il risultato dell'ottimizzazione di un modello approssimato senza il bisogno di usare il più dettagliato modello ad alta fedeltà. L'utilizzo di modelli particolarmente accurati può rendere onerosa ogni iterazione volta alla soluzione del problema di ottimizzazione; da qui prende piede la ricerca di una metodologia rivolta alla riduzione del costo computazionale associato all'utilizzo del modello ad alta fedeltà. Nel seguito saranno indicati con \mathbf{x}_c il valore iniziale assunto dalle variabili di progetto, con $a_c(\mathbf{x})$ il modello approssimato, con Δ_c le dimensioni della *trust region*, con $f_{HI}(\mathbf{x})$ il modello ad alta fedeltà, con $f_{LO}(\mathbf{x})$ il modello a bassa fedeltà, con \mathbf{s} la variabile locale definita nella *trust region* ed infine con ρ_c la cifra di merito.

3.1 Idea di base

L'approccio multi-fedeltà è uno dei modi di ridurre lo sforzo di calcolo necessario in un processo di ottimizzazione. *Approximation Model Management Optimization* (AMMO) è una metodologia rigorosa che consente l'utilizzo di modelli meno accurati (**modelli a bassa fedeltà**) per raggiungere le stesse soluzioni ottimali che potrebbero essere ottenute utilizzando modelli con una migliore corrispondenza alla realtà (**modelli ad alta fedeltà**). Questo approccio si basa sulla generazione di modelli “surrogati”, che combina le proprietà di modelli più o meno accurati attraverso l'uso appropriato del concetto di “*trust region*”. In particolare, il processo di ottimizzazione viene

effettuato in una regione di fiducia in cui la funzione ad alta fedeltà ed il modello surrogato hanno le stesse caratteristiche di salita e discesa. La funzione surrogata, quindi, mima l'andamento del modello ad alta fedeltà non solo in un punto, ma anche in un intorno di esso. Per far sì che il modello surrogato risponda a queste caratteristiche, si fa ricorso alle “condizioni di consistenza” spiegate nel paragrafo seguente.

3.1.1 Condizioni di consistenza

L'idea è quella di utilizzare un modello approssimato $a_c(\mathbf{x})$ che soddisfi le cosiddette “condizioni di consistenza” con $f_{HI}(\mathbf{x})$, nel punto iniziale \mathbf{x}_c :

$$a_c(\mathbf{x}_c) = f_{HI}(\mathbf{x}_c) \quad (3.1)$$

$$\nabla a_c(\mathbf{x}_c) = \nabla f_{HI}(\mathbf{x}_c) \quad (3.2)$$

Queste condizioni garantiscono che $a_c(\mathbf{x})$ simuli l'andamento di $f_{HI}(\mathbf{x})$ nell'intorno di \mathbf{x}_c .

3.2 β correlation

Il modo di costruire una funzione $a_c(\mathbf{x})$ che soddisfi le condizioni di congruenza non è univoco. Il metodo maggiormente utilizzato in letteratura è quello della β – correlation, in cui si definisce una funzione rapporto $\beta(\mathbf{x})$ come:

$$\beta(\mathbf{x}) = \frac{f_{HI}(\mathbf{x})}{f_{LO}(\mathbf{x})} \quad (3.3)$$

Dopodichè si effettua l'espansione in serie di Taylor di $\beta(\mathbf{x})$ intorno ad \mathbf{x}_c , troncata al primo ordine:

$$\beta_c(\mathbf{x}) = \beta(\mathbf{x}_c) + \nabla^T \beta(\mathbf{x}_c) \cdot (\mathbf{x} - \mathbf{x}_c) \quad (3.4)$$

e si costruisce il modello approssimato $a_c(\mathbf{x})$ come:

$$a_c(\mathbf{x}) = \beta_c(\mathbf{x})f_{LO}(\mathbf{x}) \quad (3.5)$$

essenzialmente $\beta_c(\mathbf{x})$ può essere visto come un fattore di scala lineare, attorno ad \mathbf{x}_c , cosicché $a_c(\mathbf{x})$ mima $f_{HI}(\mathbf{x})$ non solo in \mathbf{x}_c , ma anche “sufficientemente” lontano da esso.

3.3 Parametro di merito

Per confrontare il risultato ottenuto in termini di minimo, utilizzando $a_c(\mathbf{x})$ anziché $f_{HI}(\mathbf{x})$, viene utilizzato un parametro, detto “di merito” ρ_c definito:

$$\rho_c = \frac{f_{HI}(\mathbf{x}_c) - f_{HI}(\mathbf{x}_{min})}{f_{HI}(\mathbf{x}_c) - a_c(\mathbf{x}_{min})} \quad (3.6)$$

quindi ρ_c indica il rapporto tra il miglioramento ottenuto minimizzando $a_c(\mathbf{x})$ invece di $f_{HI}(\mathbf{x})$.

3.4 Aggiornamento della *trust region* e criterio di convergenza

Sul valore di ρ_c è basata la gestione della *trust region* e del nuovo valore di \mathbf{x}_c ad ogni iterazione. Infatti la minimizzazione di $a_c(\mathbf{x})$ è effettuata in una regione con lunghezza caratteristica Δ_c e centrata in \mathbf{x}_c . Il modo di aggiornare tale *trust region* non è unico, ma l'idea di base è molto semplice: aumentare la dimensione se nella iterazione corrente vi è un miglioramento, diminuirla altrimenti. Un semplice esempio può essere:

$$\begin{array}{ll} \text{Se} & \rho_c \leq 0.5 \implies \Delta_c^{i+1} = 0.8\Delta_c^i \\ \text{Altrimenti se} & \rho_c > 0.5 \implies \Delta_c^{i+1} = 1.2\Delta_c^i \end{array}$$

Come ogni metodo iterativo, vi deve essere una condizione di convergenza, e.g.:

$$\begin{aligned} &\text{Scelto} \quad \epsilon > 0 \\ &\text{Fermati se} \quad \| \mathbf{x}_c - \mathbf{x}_{min} \| \leq \epsilon \end{aligned}$$

3.5 Diagramma di flusso

L'algoritmo utilizzato è mostrato nel diagramma di flusso indicato nella Figura 3.1.

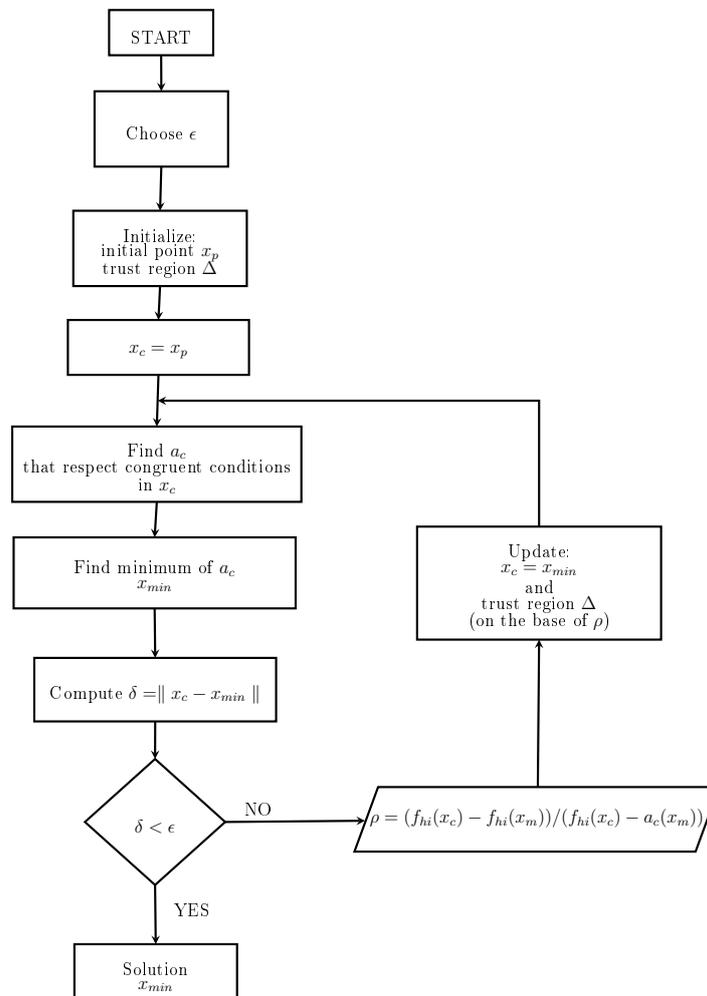


Figura 3.1: AMMO - Diagramma di flusso.

Capitolo 4

L'idea Multilivello

L'idea sviluppata nel presente lavoro si basa su un innovativo schema di gestione della *trust region* derivante dall'introduzione di un modello di precisione intermedia f_{MID} , nonché un ulteriore modello surrogato. Tale gestione è progettata in modo da ridurre le chiamate al modello ad alta fedeltà al costo dell'introduzione delle chiamate ad un modello a media fedeltà f_{MID} . Inutile dire che, il miglioramento dell'efficienza operativa del calcolo ottenuto con l'approccio proposto dipende fortemente dall'onere computazionale richiesto dai modelli ad alta, media e bassa fedeltà.

Vale la pena sottolineare che l'analisi sviluppata nella presente tesi è incentrata sulla diminuzione dei tempi di calcolo a parità di accuratezza della soluzione ottenuta. Tuttavia, l'utilizzo di tre modelli potrebbe, potenzialmente, permettere di ampliare l'applicabilità dei metodi multi-fedeltà a problemi che presentino una fisica particolarmente articolata.

4.1 Primo modello approssimato

Fissata una prima *trust region* Δ , viene creata una funzione $a_{hm}(\mathbf{x})$ utilizzando il metodo della β – *correlation* (par. 3.2) tra $f_{HI}(\mathbf{x})$ e $f_{MID}(\mathbf{x})$:

$$\beta_{MID}(\mathbf{x}) = \frac{f_{HI}(\mathbf{x})}{f_{MID}(\mathbf{x})} \quad (4.1)$$

$$\beta_{MID}^c(\mathbf{x}) = \beta_{MID}(\mathbf{x}_c) + \nabla^T \beta_{MID}(\mathbf{x}_c) \cdot (\mathbf{x} - \mathbf{x}_c) \quad (4.2)$$

$$a_{hm}(\mathbf{x}) = \beta_{MID}^c(\mathbf{x}) f_{MID}(\mathbf{x}) \quad (4.3)$$

Ovviamente, deve essere trovato il minimo di a_{hm} all'interno della *trust region* Δ e a questo punto si ha il carattere completamente innovativo dell'approccio proposto rispetto al "classico" AMMO. La funzione a_{hm} non viene minimizzata direttamente, ma avvalendosi di un modello surrogato di quest'ultima funzione.

4.2 Surrogato del modello approssimato

Per trovare il minimo della funzione $a_{hm}(\mathbf{x})$ indicata nel paragrafo precedente viene creata una funzione $a_{sl}(\mathbf{x})$ utilizzando nuovamente il metodo della β – *correlation*, questa volta tra $a_{hm}(\mathbf{x})$ e $f_{LO}(\mathbf{x})$

$$\beta_{sur}(\mathbf{x}) = \frac{a_{hm}(\mathbf{x})}{f_{LO}(\mathbf{x})} \quad (4.4)$$

$$\beta_{sur}^c(\mathbf{x}) = \beta_{sur}(\mathbf{x}_c) + \nabla^T \beta_{sur}(\mathbf{x}_c) \cdot (\mathbf{x} - \mathbf{x}_c) \quad (4.5)$$

$$a_{sl}(\mathbf{x}) = \beta_{sur}^c(\mathbf{x}) f_{LO}(\mathbf{x}) \quad (4.6)$$

La funzione $a_{sl}(\mathbf{x})$ verrà minimizzata all'interno di una *trust region* Δ_{int} **inizialmente**¹ interna alla *trust region* Δ .

¹Il perchè *inizialmente* verrà chiarito nel capitolo dedicato ai risultati.

4.3 Aggiornamento di Δ_{int}

Ad ogni iterazione e cioè ogni volta che verrà trovato il minimo globale di $a_{sl}(\mathbf{x})$ all'interno di Δ_{int} quest'ultima verrà aggiornata, con lo stesso criterio mostrato nel paragrafo 3.4, sulla base del valore assunto dal parametro di merito:

$$\rho_{int} = \frac{a_{hm}(\mathbf{x}_c) - a_{hm}(\mathbf{x}_{min})}{a_{hm}(\mathbf{x}_c) - a_{sl}(\mathbf{x}_{min})} \quad (4.7)$$

La ricerca del minimo di $a_{hm}(\mathbf{x})$ all'interno di Δ utilizzando $a_{sl}(\mathbf{x})$ termina quando è verificata una delle due condizioni seguenti:

- o la differenza tra il minimo trovato in due iterazioni successive è minore di un ϵ_{int} scelto a priori dal progettista;
- o il minimo trovato ricade all'esterno di Δ e quindi come nuovo punto da cui far ripartire la ricerca del minimo si sceglie il punto sulla frontiera di Δ più vicino a tale minimo.

Questa doppia opportunità di stop si è rilevata di importanza fondamentale per ciò che viene presentato. Nel capitolo 5 verrà infatti mostrato come, mentre nelle prime iterazioni la condizione che porta allo stop delle iterazioni è sempre la seconda, da un certo punto in poi le iterazioni si fermeranno per il verificarsi della prima condizione e quindi la *trust region* Δ_{int} , che inizialmente era interna a Δ , si svincolerà da quest'ultima.

4.4 Aggiornamento di Δ

L'ultimo aspetto che resta da illustrare è quello riguardante l'aggiornamento della *trust region* Δ . Ogni volta che viene trovato il minimo globale di $a_{hm}(\mathbf{x})$ all'interno di Δ , quest'ultima viene aggiornata, insieme al punto iniziale corrente, sempre secondo il criterio descritto nel paragrafo 3.4 sulla base del

valore di ρ_{ext} :

$$\rho_{ext} = \frac{f_{HI}(\mathbf{x}_c) - f_{HI}(\mathbf{x}_{min})}{f_{HI}(\mathbf{x}_c) - a_{hm}(\mathbf{x}_{min})} \quad (4.8)$$

4.5 L'agoritmo MAMMO

L'agoritmo dell'intero processo è mostrato nel diagramma di figura 4.1. E' importante notare che nel processo di ottimizzazione interno, il costo computazionale è dato da f_{LO} , visto che la funzione che viene minimizzata è $a_{sl}(\mathbf{x})$. L'onere computazionale del ciclo esterno è invece quello di f_{MID} dato che viene utilizzata la funzione:

$$a_{hm}(\mathbf{x}) = \beta_{MID}^c(\mathbf{x})f_{MID}(\mathbf{x}) \quad (4.9)$$

Quindi, il modello ad alta fedeltà f_{HI} viene richiamato solo per calcolare $\beta_{MID}^c(\mathbf{x})$ ad ogni iterazione del ciclo esterno.

Un altro aspetto da sottolineare è come sia possibile minimizzare $a_{sl}(\mathbf{x})$ per trovare il minimo di $f_{HI}(\mathbf{x})$ dato che $a_{hm}(\mathbf{x})$ soddisfa le condizioni di consistenza con $f_{HI}(\mathbf{x})$ e, a sua volta, $a_{sl}(\mathbf{x})$ soddisfa le condizioni di consistenza con $a_{hm}(\mathbf{x})$ e quindi, in definitiva, $a_{sl}(\mathbf{x})$ soddisfa le condizioni di consistenza con $f_{HI}(\mathbf{x})$.

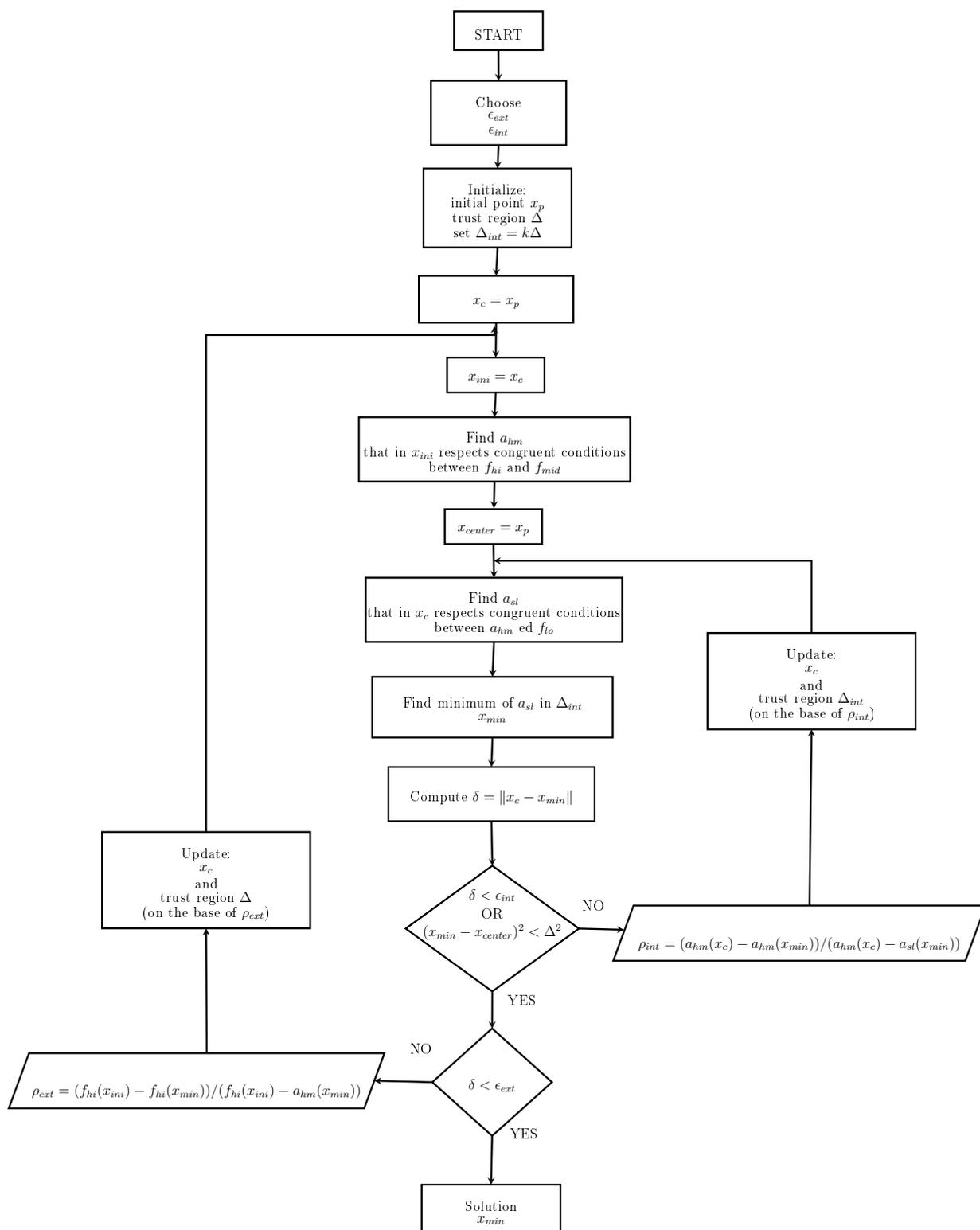


Figura 4.1: MAMMO - diagramma di flusso

Capitolo 5

Risultati

Il presente lavoro è stato interamente portato avanti con l'utilizzo del Software *Wolfram Mathematica*[®] che fornisce un ottimizzatore globale numerico tramite l'utilizzo del comando *NMinimize*. Tale ottimizzatore implementa tutti i metodi illustrati nella sezione 2.2.2. Tali metodi sono sufficientemente flessibili da poter ricoprire tutte le tipologie di problemi e funzioni, anche quelle non continue o non differenziabili e in cui il minimo globale non è di immediata individuazione. I vincoli possono essere una combinazione di vincoli di uguaglianza, disuguaglianza o specifiche di dominio. I primi due tipi possono essere non lineari. Ogni vincolo di disuguaglianza forte viene convertito in debole per evitare problemi dovuti al fatto che si sta lavorando con numeri approssimati. Le variabili possono essere numeri interi o reali e vengono assunti reali se non diversamente specificato. I vincoli generalmente vengono fatti rispettare applicando delle penalità ogni qualvolta i punti lasciano la regione accettabile.

Per funzionare, il comando ha bisogno di una regione rettangolare iniziale in cui far partire l'ottimizzazione, il che è l'equivalente di altri metodi numerici che hanno bisogno di uno o più punti di partenza. Tale regione iniziale è

specificata dando ad ogni variabile un limite inferiore ed uno superiore e cioè aggiungendo il vincolo $a \leq x \leq b$.

Prima di procedere con il mostrare i risultati ottenuti è importante sottolineare due aspetti:

il primo è che lo scopo del presente lavoro non è lo sviluppo di un nuovo tipo di ottimizzatore, quindi l'ottimizzatore viene utilizzato come una "black-box" in grado di trovare il minimo globale vincolato di una data funzione;

il secondo e più significativo è che, andando avanti con il lavoro, oltre ad un notevole miglioramento in termini di chiamate al modello ad alta fedeltà, è stata riscontrata una caratteristica veramente innovativa dell'attuale approccio, probabilmente più importante della riduzione del numero di chiamate: per alcune funzioni multimodali, anche utilizzando un ottimizzatore globale, la metodologia classica AMMO sembra fallire, nel senso che non è in grado di determinare il minimo globale, ma si ferma al minimo locale più vicino al punto iniziale. Contrariamente, il metodo MAMMO, anche in questi casi, è in grado di determinare il minimo globale.

Nell'illustrare i risultati, la prima sezione sarà dedicata ai casi per cui anche il metodo AMMO trova il minimo globale, ma utilizzando il metodo MAMMO il numero di chiamate al modello ad alta fedeltà è considerevolmente minore. Nella seconda parte verranno mostrate due funzioni per cui il metodo AMMO si ferma nel minimo locale più vicino al punto iniziale, mentre la metodologia MAMMO riesce a trovare il minimo globale.

Per tutti i test verrà indicato quali sono le funzioni f_{LO} e f_{MID} utilizzate. A questo proposito è opportuna una considerazione sulla scelta di tali funzioni. Esse devono avere un andamento che non si discosti troppo da quello della funzione ad alta fedeltà e, per garantire ciò, come funzioni a media e bassa fedeltà sono stati scelti sviluppi in serie di Taylor troncati ad ordini diversi

della funzione ad alta fedeltà stessa, con il grado di f_{MID} , ovviamente, maggiore di quello di f_{LO} . Nell'illustrare i risultati sarà quindi indicato l'ordine a cui si è scelto di troncare tali serie nei due casi di media e bassa fedeltà.

5.1 Numero chiamate funzione ad alta fedeltà MAMMO *versus* AMMO

I test sono stati portati avanti su funzioni analitiche di complessità man mano crescente. Si è iniziato con una funzione monodimensionale monomodale, per poi proseguire con alcune funzioni bidimensionali prima monomodali, dopodichè multimodali. In tutti i casi viene graficato l'andamento della funzione ad alta fedeltà e viene indicato il numero di chiamate ai vari modelli per i seguenti tre tipi di ottimizzazione:

- ottimizzazione diretta senza utilizzare nessuna funzione approssimante e quindi l'unico numero di chiamate è quello alla funzione ad alta fedeltà;
- ottimizzazione utilizzando la metodologia AMMO e quindi i contatori saranno due, quello relativo alla funzione ad alta fedeltà f_{HI} e quello relativo alla funzione a bassa fedeltà f_{LO} ;
- ottimizzazione utilizzando la metodologia MAMMO e quindi tre contatori, rispettivamente quelli relativi a f_{HI} , f_{MID} e f_{LO} .

Inoltre, verrà mostrata l'evoluzione delle *trust region* in entrambe le metodologie. Mostrare tutte le *trust region* ad ogni iterazione risulterebbe, ovviamente, estremamente pesante dal punto di vista della lettura nonchè poco chiaro. Sono state scelte, quindi, le iterazioni ritenute più significative e utili

per meglio comprendere i due metodi. Nel caso monodimensionale questa visualizzazione è stata sostituita da due tabelle.

Nei vari casi analizzati verrà, di volta in volta, sottolineato l'aspetto più rilevante riscontrato per il caso in esame.

5.1.1 Funzione monodimensionale monomodale

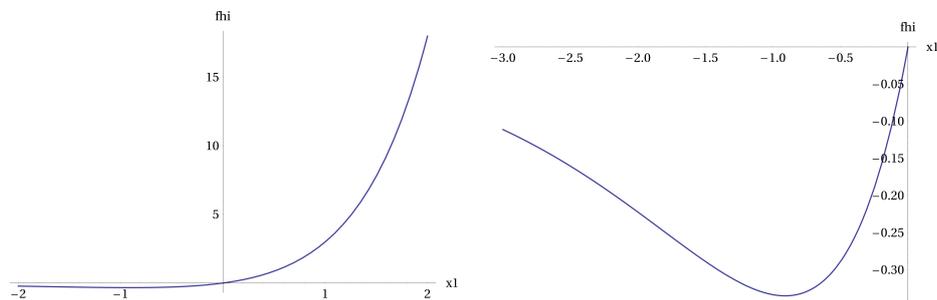


Figura 5.1: $f(x) = x3^x$

Minimo: $x = -0.91$, $f(x) = -0.335$.

Numero di chiamate:

- Ottimizzazione diretta:
 - funzione alta fedeltà: 519;
- Metodologia AMMO:
 - funzione alta fedeltà: 12 ;
 - funzione bassa fedeltà (serie di Taylor troncata al 2° ordine): 3152;

x_c	Trust region
-2	0.2
-1.8	0.24
-1.56	0.288
-1.2720	0.346
-0.9264	0.415
-0.9102	0.498
-0.9102	0.597

Tabella 5.1: Punto iniziale e ampiezza della *trust region* in AMMO

- Metodologia MAMMO:
 - funzione alta fedeltà: 6;
 - funzione media fedeltà (serie di Taylor troncata al 6° ordine): 46;
 - funzione bassa fedeltà: 12272.

5. Risultati

x_c	Trust region 1	Trust region 2
-2	0.2	0.02
-1.9878	0.2	0.0240
-1.9738	0.2	0.0288
-1.9578	0.2	0.0345
-1.9391	0.2	0.0415
-1.9171	0.2	0.0498
-1.8907	0.2	0.0597
-1.8586	0.2	0.0717
-1.8187	0.2	0.0859
-1.76785	0.2	0.10319
-1.70106	0.2	0.12383
-1.6103	0.2	0.14860
-1.48398	0.2	0.178322
-1.31204	0.2	0.21398
-1.1207	0.2	0.256784
-0.992423	0.2	0.30814
-0.951314	0.2	0.369769
-0.945119	0.2	0.443722
-0.944482	0.2	0.53246
-0.944421	0.24	0.63896
-0.912233	0.24	0.76675
-0.910287	0.24	0.920102
-0.91024	0.288	1.10412

Tabella 5.2: Punto iniziale e ampiezza delle due *trust region* in MAMMO

I metodi vengono paragonati scegliendo, ovviamente, stesso punto iniziale e stessa dimensione iniziale della trust region, nello specifico:

$$\begin{cases} x_c = -2 \\ \Delta = 2 \end{cases}$$

Nel caso MAMMO, la trust region inizialmente interna è stata scelta pari ad un decimo di quella esterna:

$$\Delta_{int} = 0.1\Delta \tag{5.1}$$

Analizzando il numero di chiamate alla funzione ad alta fedeltà utilizzando il metodo AMMO e la tabella 5.1, è immediato notare come il numero di chiamate sia dato da:

$$2 \times \text{numero di iterazioni}$$

infatti, ad ogni iterazione va calcolata la funzione approssimata da minimizzare a_c e quindi lo sviluppo in serie di Taylor troncato al primo ordine di

$$\beta(x) = \frac{f_{HI}(x)}{f_{LO}(x)}$$

E' necessario quindi conoscere il valore di f_{HI} e di $\frac{df_{HI}}{dx}$ in x_c e per calcolare la derivata, ad esempio, alle differenze finite (all'avanti o indietro) si devono conoscere $f_{HI}(x_c)$ e $f_{HI}(x_c \pm \Delta x)$, quindi ad ogni iterazione f_{HI} deve essere valutata due volte.

Per quanto riguarda il metodo MAMMO vale lo stesso ragionamento notando che ogni volta che viene aggiornata la *trust region* relativa al ciclo esterno del diagramma di flusso 4.1 si deve valutare due volte la funzione ad alta fedeltà f_{HI} , mentre ad ogni iterazione del ciclo interno la funzione da valutare due volte è f_{MID} .

5.1.2 Funzioni bidimensionali monomodali

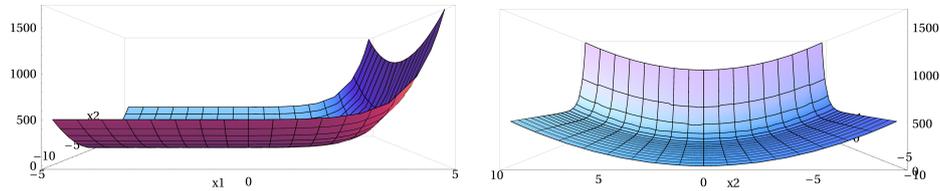


Figura 5.2: $f(x, y) = x3^x + 5y^2$

Minimo: $x = -0.91, y = 0, f(x, y) = -0.335$.

Numero di chiamate:

- Ottimizzazione diretta:
 - funzione alta fedeltà: 66;
- Metodologia AMMO:
 - funzione alta fedeltà: 36 ;
 - funzione bassa fedeltà (serie di Taylor troncata al 2° ordine): 4758;
- Metodologia MAMMO:
 - funzione alta fedeltà: 15;
 - funzione media fedeltà (serie di Taylor troncata al 4° ordine): 66;
 - funzione bassa fedeltà: 16632.

Nella figura 5.3 sono mostrate **tutte** le iterazioni del metodo AMMO, con la variazione della dimensione e del centro della *trust region* graficate sulle isolivello della funzione in oggetto. In rosso è indicata la posizione del minimo.

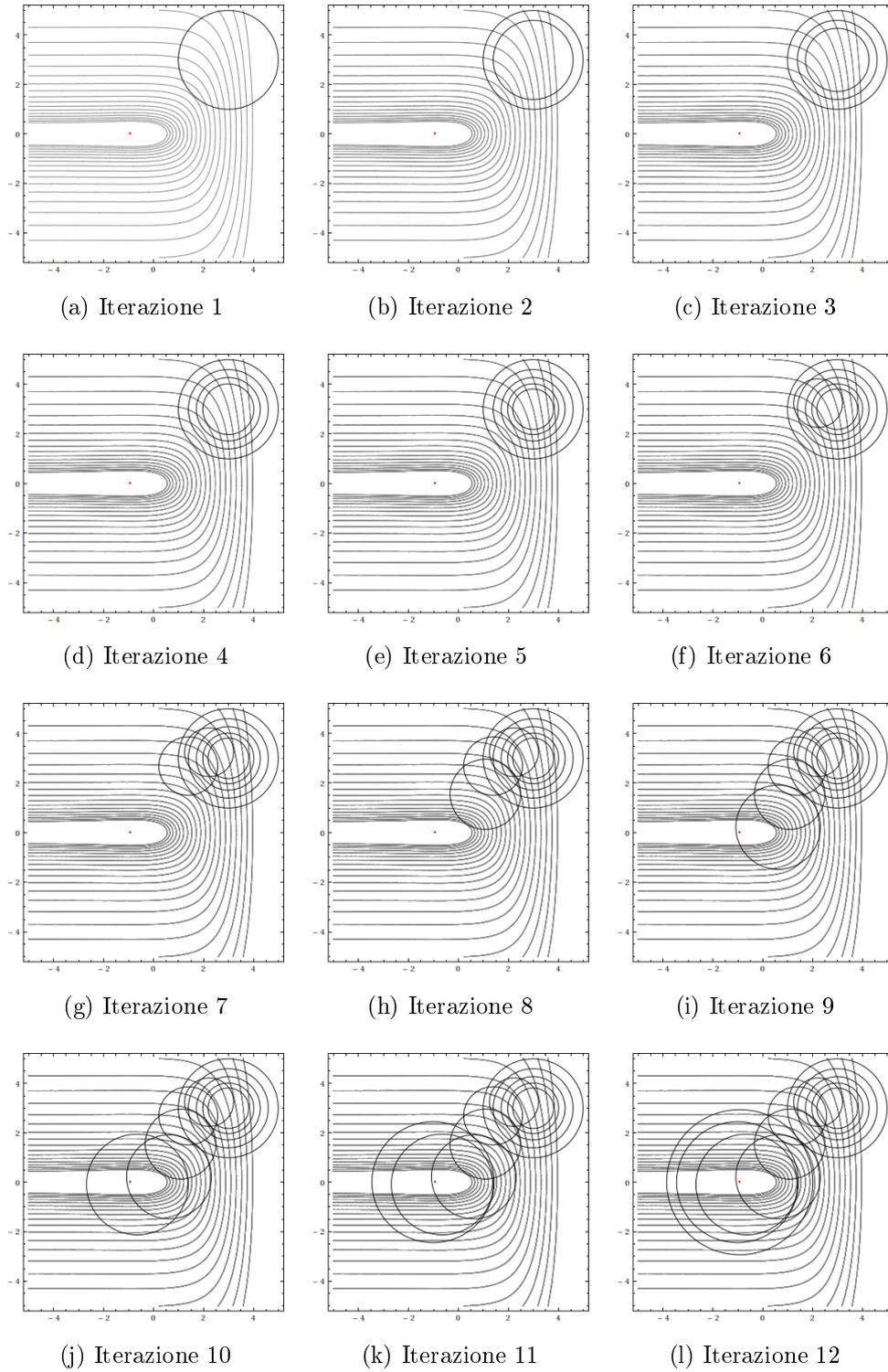


Figura 5.3: Iterazioni metodo AMMO

È interessante notare il comportamento della *trust region* nelle iterazioni iniziali. Vale la pena ricordare che il primo valore di Δ è assegnato “arbitrariamente” dal progettista. Dalle figure 5.3a-5.3e si nota come, qualunque sia il valore assegnato inizialmente a Δ , l’algoritmo adatta tale dimensione in modo che sia rispettata la condizione sul parametro di merito ρ_c , dopodichè, dalla iterazione 6 (figura 5.3f) inizia a cambiare il punto iniziale corrente \mathbf{x}_c e, così, a ricercare il minimo.

Nel caso di funzioni bidimensionali il numero di chiamate al modello ad alta fedeltà sarà dato da:

$$3 \times \text{numero di iterazioni}$$

in quanto, per la stessa motivazione fornita nel paragrafo 5.1.1, il metodo necessita di $f_{HI}(\mathbf{x}_c)$, $\frac{\partial f_{HI}(\mathbf{x}_c)}{\partial x}$ e $\frac{\partial f_{HI}(\mathbf{x}_c)}{\partial y}$ e quindi, per calcolarsi le derivate nelle due direzioni alle differenze finite, di:

$$f_{HI}(x_c, y_c) \quad , \quad f_{HI}(x_c \pm \Delta x, y_c) \quad \text{e} \quad f_{HI}(x_c, y_c \pm \Delta y)$$

In figura 5.4 sono mostrate alcune delle iterazioni ritenute maggiormente significative nell’evoluzione delle due *trust region* con cui si ha a che fare utilizzando il metodo MAMMO. Il minimo è indicato in blu, la *trust region* esterna in nero, mentre quella interna in rosso. Le prime cinque immagini mostrano l’evoluzione della *trust region* interna all’interno di quella esterna; nelle restanti quattro si è voluto porre l’attenzione sui momenti in cui la *trust region* esterna viene aggiornata.

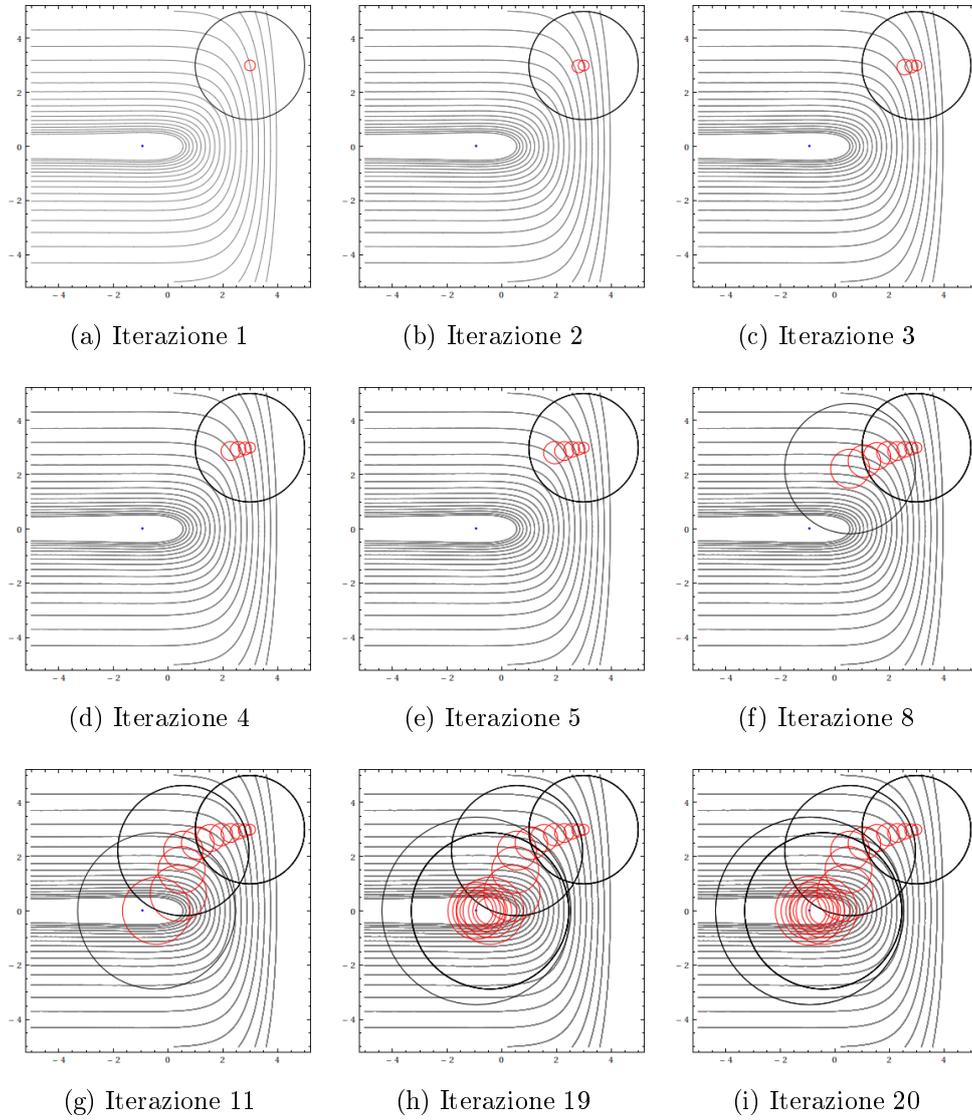


Figura 5.4: Iterazioni metodo MAMMO

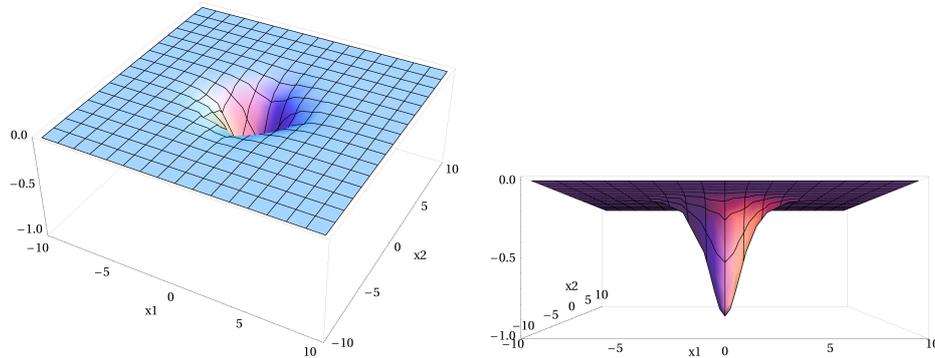


Figura 5.5: $f(x, y) = -\frac{1}{\cosh(x)\cosh(y)}$

Minimo: $x = 0, y = 0, f(x, y) = -1$.

Numero di chiamate:

- Ottimizzazione diretta:
 - funzione alta fedeltà: 129;
- Metodologia AMMO:
 - funzione alta fedeltà: 39 ;
 - funzione bassa fedeltà (serie di Taylor troncata al 4° ordine): 2007;
- Metodologia MAMMO:
 - funzione alta fedeltà: 15;
 - funzione media fedeltà (serie di Taylor troncata al 6° ordine): 51;
 - funzione bassa fedeltà: 3067.

In figura 5.6 sono mostrate le iterazioni iniziali e finali per entrambi i metodi.

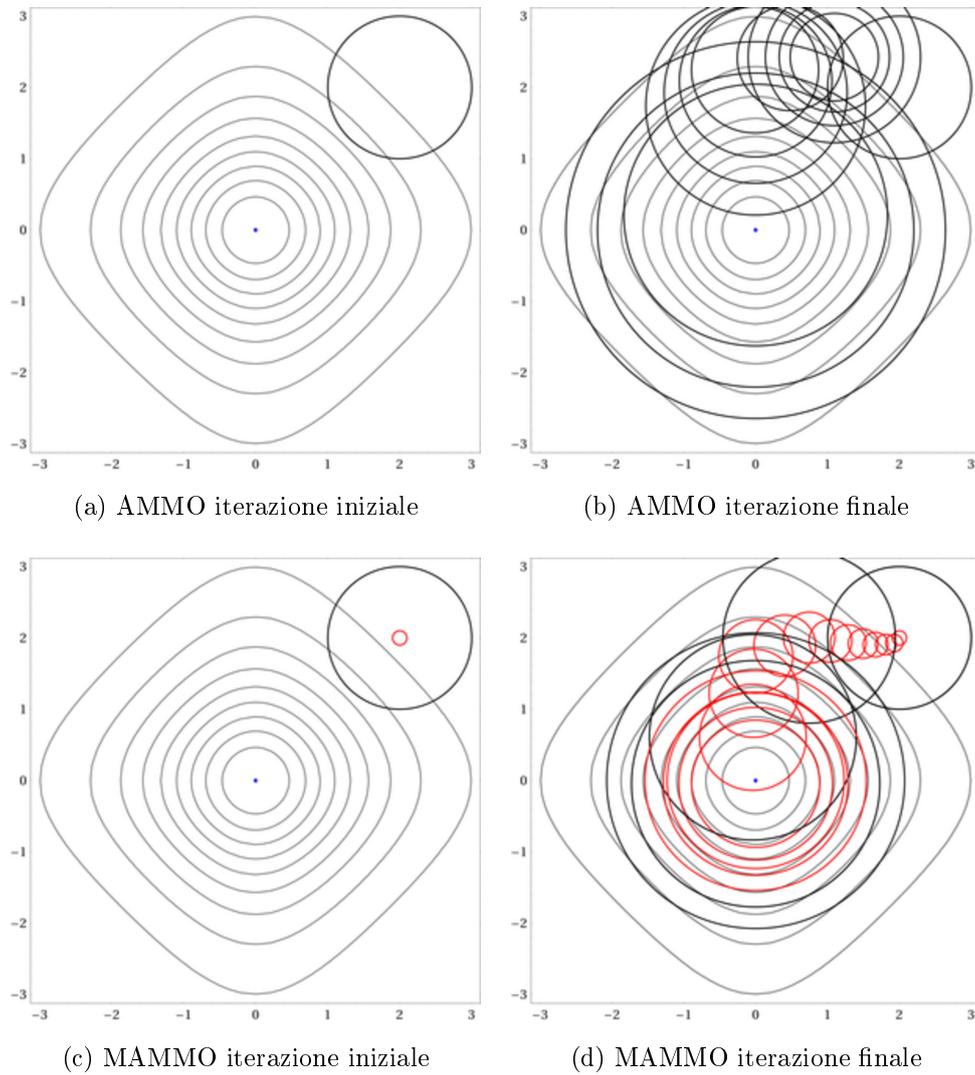


Figura 5.6: AMMO e MAMMO

Arrivati a questo esempio, una parentesi può essere fatta sulla scelta delle funzioni a media e bassa fedeltà. In tutti questi casi puramente matematici, come funzioni a fedeltà diversa dalla alta sono stati scelti dei polinomi, di grado variabile con f_{MID} di grado superiore di f_{LO} . Finchè la funzione f_{HI} è rappresentata anch'essa da un polinomio è evidente che f_{MID} (e di conseguenza anche f_{LO} per quanto appena detto) debba essere di grado inferiore,

quindi non si ha piena libertà sulla scelta del grado delle funzioni a media e bassa fedeltà. In questo caso, trattandosi di una funzione trascendente, il grado dei vari polinomi può essere qualunque, con l'unica limitazione che il grado di f_{LO} sia inferiore di quello di f_{MID} . Nel caso in esame come funzione a media e bassa fedeltà sono stati scelti polinomi di sesto e quarto grado rispettivamente. L'aumentare il grado di f_{MID} , f_{LO} o entrambi per valutare se il risultato in termini di chiamate al modello ad alta fedeltà è maggiormente migliorabile, sarà sicuramente oggetto di lavori futuri.

5.1.3 Funzioni bidimensionali multimodali

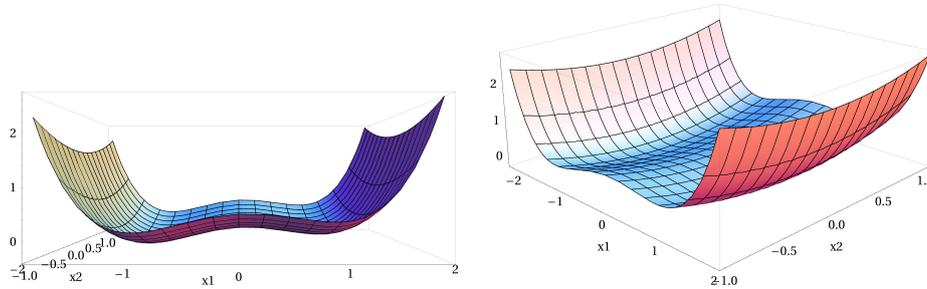


Figura 5.7: $f(x, y) = \frac{x^4}{4} - \frac{x^2}{2} + \frac{x}{10} + \frac{y^2}{2}$

Minimo: $x = -1.047$, $y = 0$, $f(x, y) = -0.352$.

Numero di chiamate:

- Ottimizzazione diretta:
 - funzione alta fedeltà: 156;
- Metodologia AMMO:
 - funzione alta fedeltà: 105;
 - funzione bassa fedeltà (serie di Taylor troncata al 2° ordine): 7206;
- Metodologia MAMMO:
 - funzione alta fedeltà: 57;
 - funzione media fedeltà (serie di Taylor troncata al 3° ordine): 111;
 - funzione bassa fedeltà: 7175.

Per motivi di completezza anche in questo caso vengono mostrate l'iterazione iniziale e finale per entrambi i metodi (figura 5.8), anche non avendo particolari aspetti da sottolineare e rendendosi conto della scarsa rilevanza che grafici di questo genere possono avere.

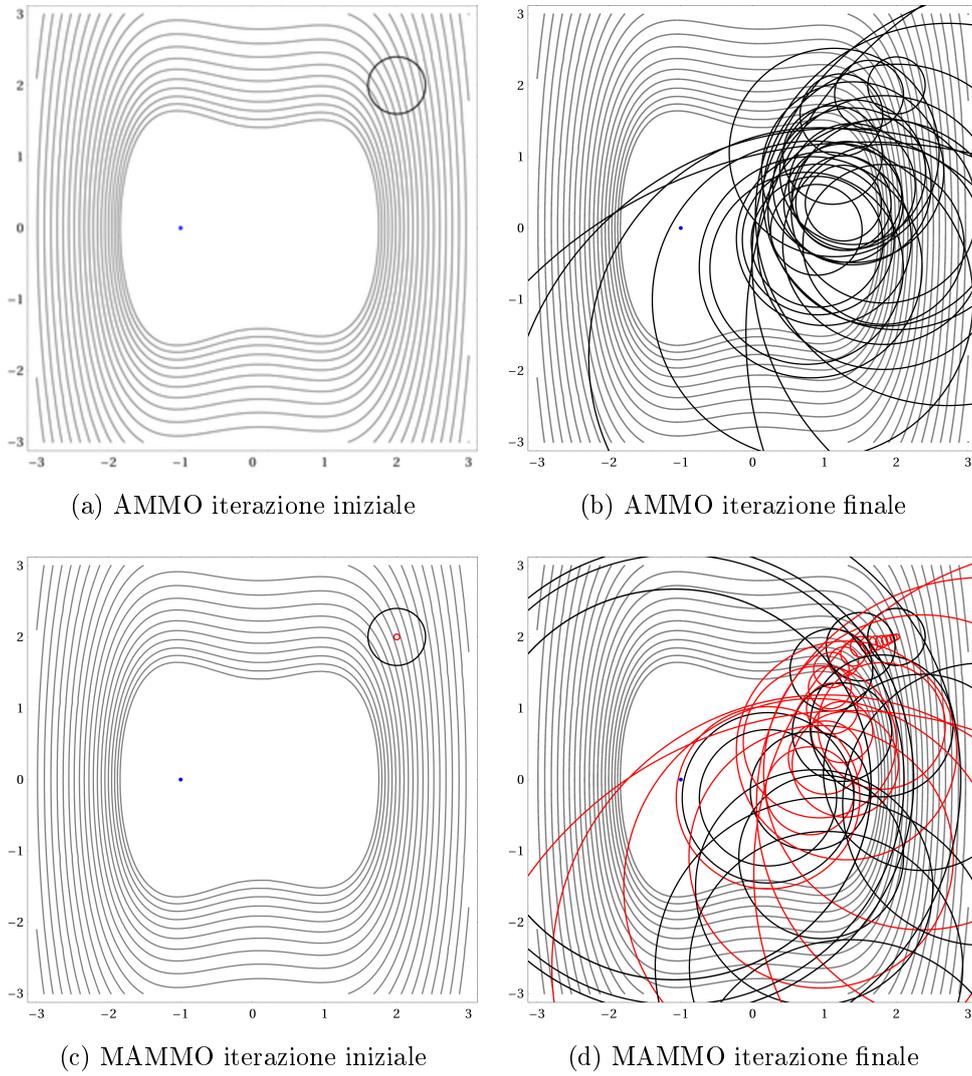


Figura 5.8: AMMO e MAMMO

Analizzata una prima funzione multimodale e confermati ancora una volta i miglioramenti in termini di riduzione di chiamate al modello ad alta fedeltà, è stata analizzata la funzione di Rosenbrock.

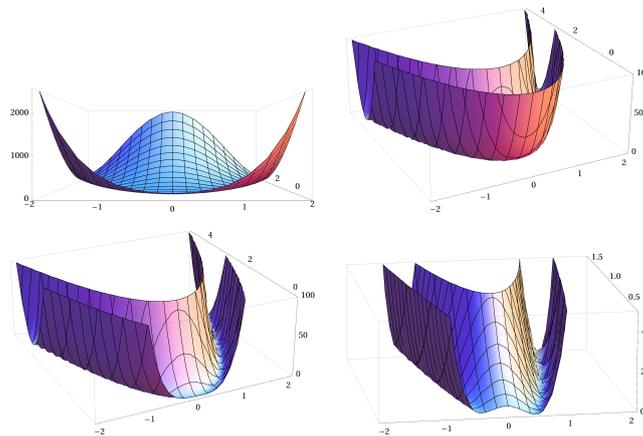


Figura 5.9: Funzione di Rosenbrock: $f(x, y) = (1 - x)^2 + 100(y - x^2)^2$

Tale funzione è largamente conosciuta nel mondo dell'ottimizzazione perchè viene utilizzata come *test case* per i vari ottimizzatori. La funzione è costituita da queste due “pareti” altissime formanti una valle in cui sono presenti diversi minimi di cui solo uno globale ed esattamente:

$$\text{Minimo: } x = 1, y = 1, f(x, y) = 0.$$

I risultati ottenuti in termini di **numero di chiamate** sono:

- Ottimizzazione diretta:
 - funzione alta fedeltà: 106;
- Metodologia AMMO:
 - funzione alta fedeltà: 66 ;
 - funzione bassa fedeltà (serie di Taylor troncata al 2° ordine): 22458;

- Metodologia MAMMO:
 - funzione alta fedeltà: 15;
 - funzione media fedeltà (serie di Taylor troncata al 3° ordine): 84;
 - funzione bassa fedeltà: 36729.

Per completezza di informazione, in figura 5.10, vengono mostrate le evoluzioni delle *trust region* nei due metodi.

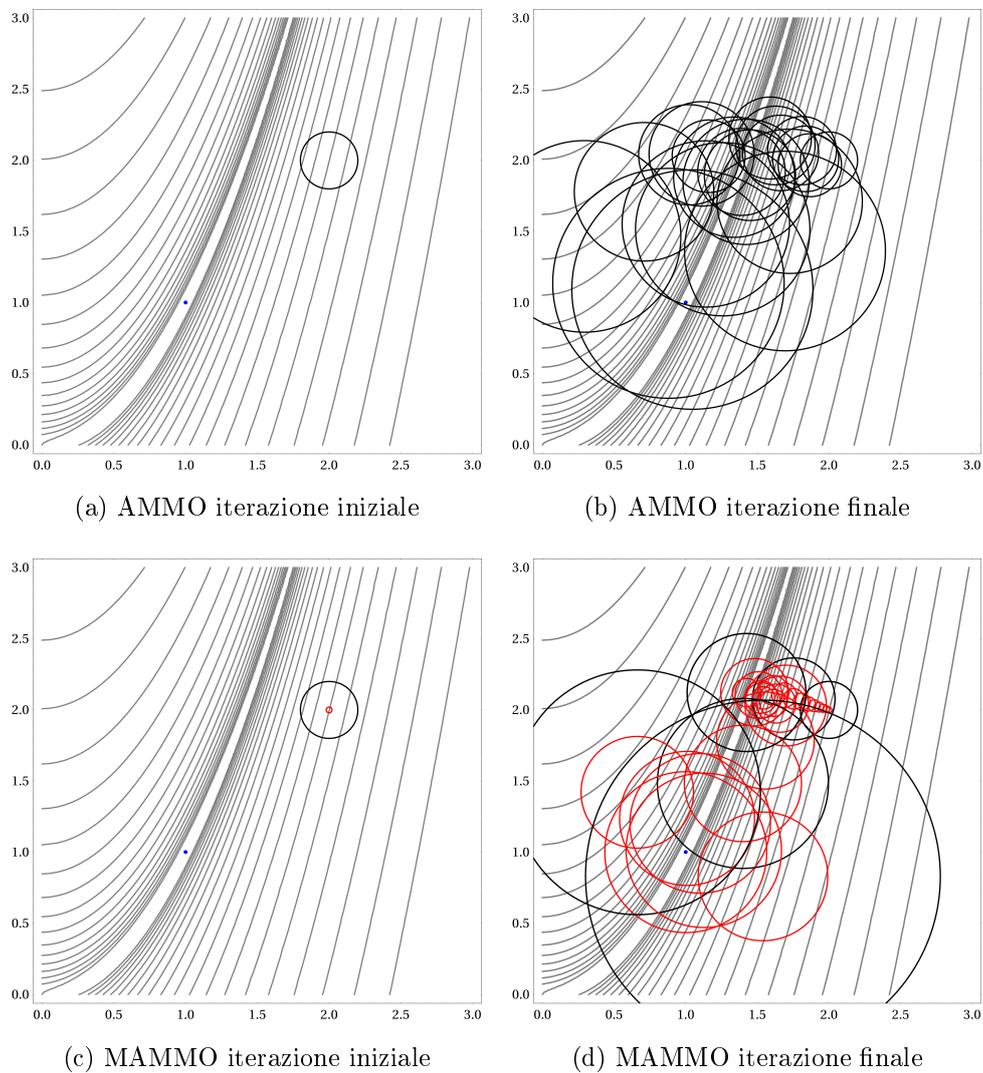


Figura 5.10: AMMO e MAMMO per Rosenbrock

5.2 Dove AMMO fallisce

Una volta illustrati diversi *test case* in cui la metodologia MAMMO risulta più performante di quella AMMO dal punto di vista della chiamate alla funzione ad alta fedeltà, passiamo adesso ad illustrare un altro aspetto per cui MAMMO si fa preferire all'approccio classico. I due metodi sono stati implementati per la funzione illustrata nella figura 5.11.

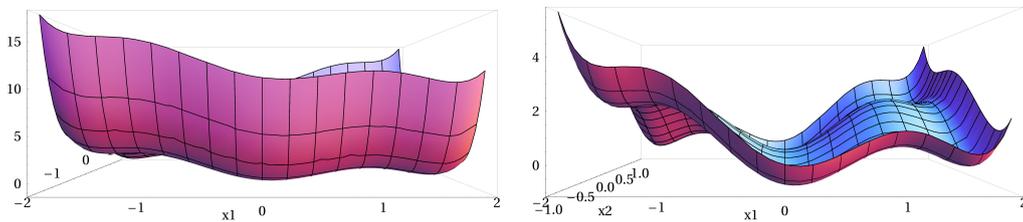


Figura 5.11: $f(x, y) = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2$

In figura 5.12 sono riportate le evoluzioni della *trust region* utilizzando la metodologia AMMO ed è evidente come, anche al variare del punto iniziale, tale algoritmo non riesca a trovare il minimo globale di f_{HI} (indicato in blu), ma si fermi nel punto di minimo globale più vicino al punto di partenza.

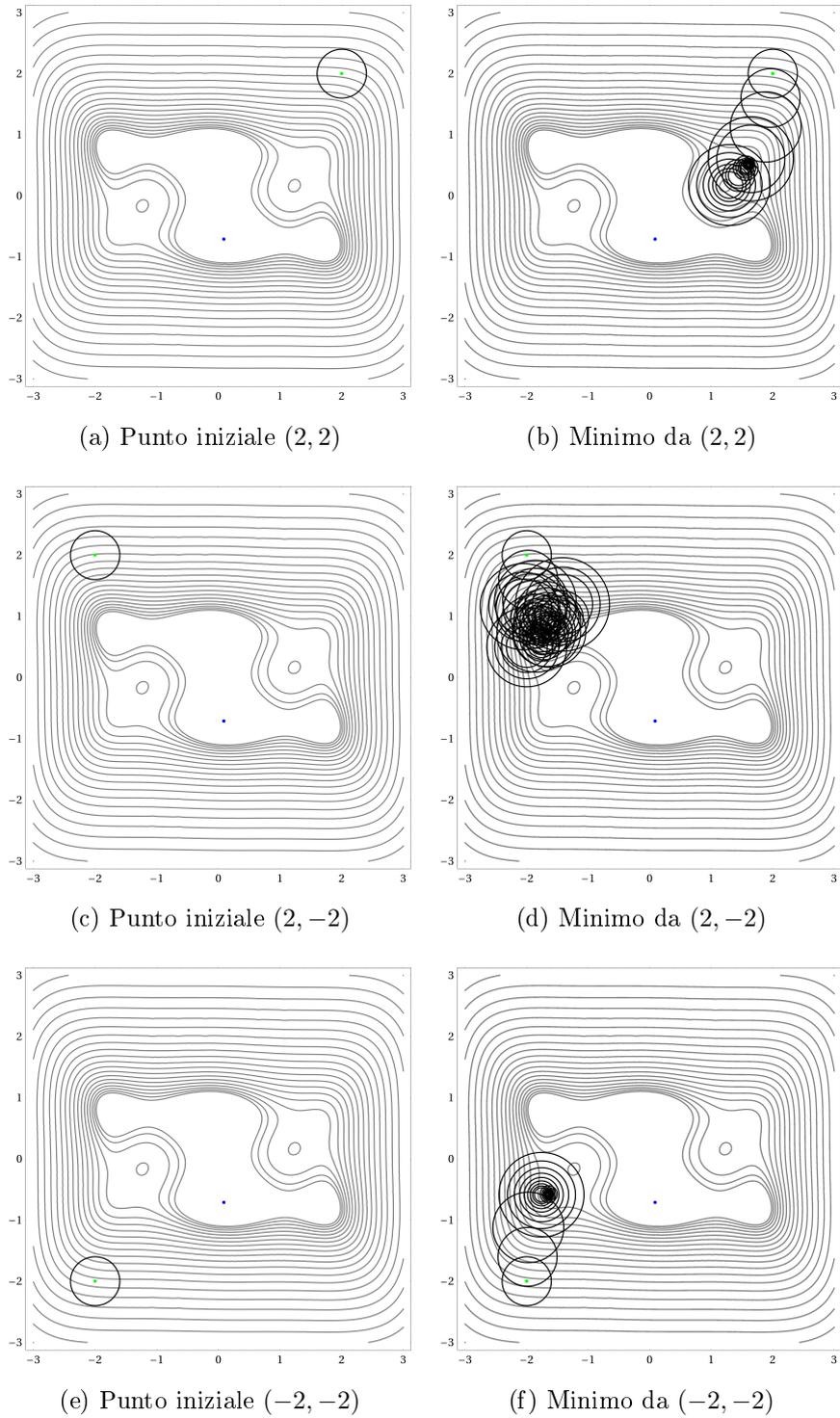


Figura 5.12: AMMO al variare del punto iniziale

Applicando, invece, il metodo in esame il risultato ottenuto è il minimo globale:

$$x_{min} = 0.0898, y_{min} = -0.7126 \longrightarrow f(x_{min}, y_{min}) = -1.0316$$

e i risultati ottenuti in termini di chiamate ai modelli sono:

- Ottimizzazione diretta:
 - funzione ad alta fedeltà: 108;
- Metodologia MAMMO:
 - funzione alta fedeltà: 33;
 - funzione media fedeltà (serie di Taylor troncata al 3° ordine): 72;
 - funzione bassa fedeltà (serie di Taylor troncata al 2° ordine): 32482.

Osservando il diagramma di flusso 4.1, la caratteristica di riuscire a catturare il minimo globale è senz'altro imputabile al fatto che ad ogni iterazione il valore trovato deve rispettare la condizione di convergenza sia del ciclo interno che di quello esterno; quindi si ha un valore di minimo ottenuto utilizzando a_{sl} con cui vengono calcolati sia il valore di ρ_{int} (legato ad a_{hm}) che quello di ρ_{ext} (legato a f_{HI}). Questo doppio controllo fa sì che le due *trust region* interagiscano tra di loro in modo da trovare il minimo globale. In figura 5.13 oltre alla prima ed ultima iterazione, sono mostrate le fasi in cui è rilevante il comportamento delle due *trust region*.

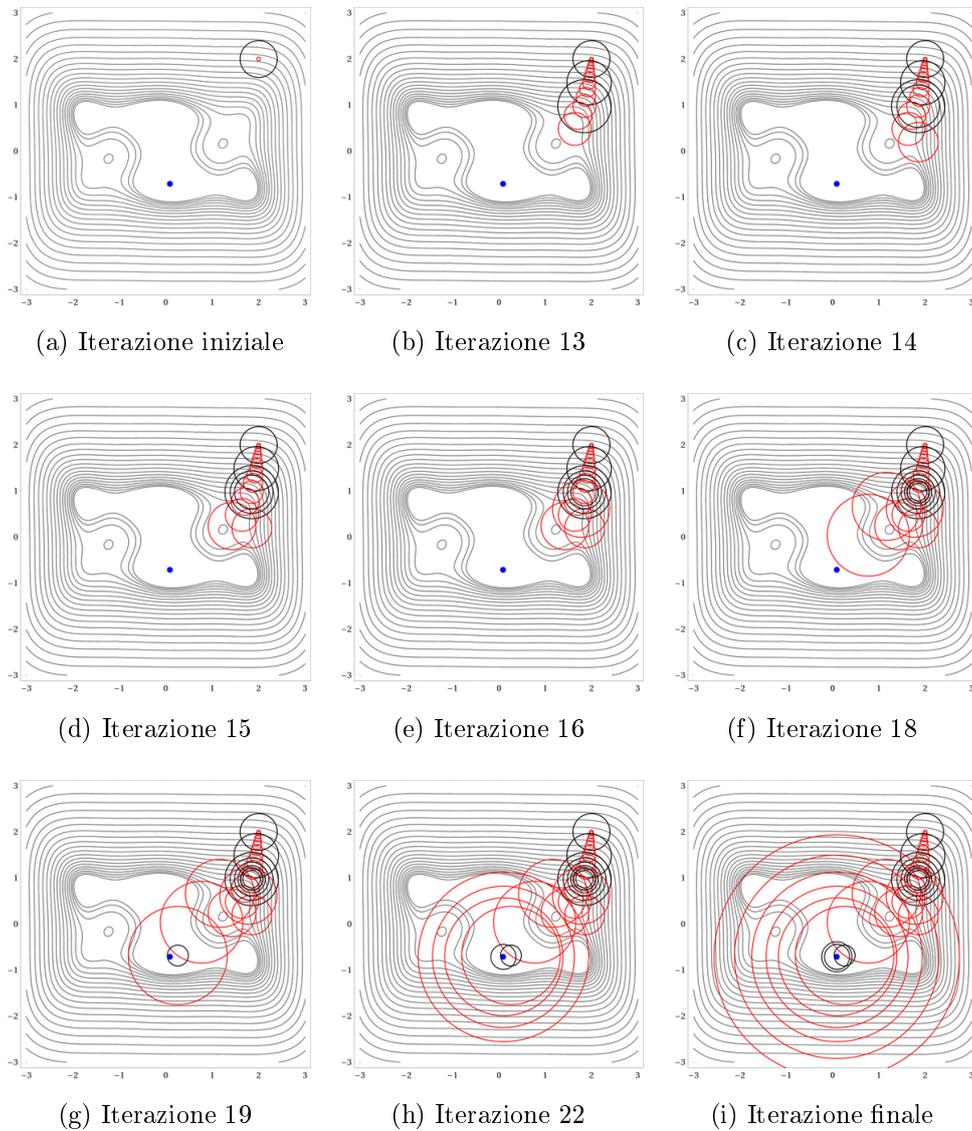


Figura 5.13: Iterazioni metodo MAMMO

Si nota come, nel passaggio dalla iterazione 13 alla 14 (figure 5.13b e 5.13c), la *trust region* inizialmente interna (rossa) diventi indipendente da quella denominata come esterna, la quale rimane nel punto in cui si trovava all'iterazione precedente e si aggiorna in base al valore assunto da ρ_{ext} nel punto di minimo trovato all'interno della *trust region* rossa (il fatto quindi che la

trust region nera sia centrata in tale punto è solo perchè nell'ultima iterazione era in quella posizione, ciò che va sottolineato è che il suo raggio si sta aggiornando in base al valore assunto da ρ_{ext} nel nuovo minimo trovato ad ogni iterazione interna). L'algoritmo continua a cercare il minimo all'interno della *trust region* rossa (figure 5.13d, 5.13e e 5.13f), finchè non trova un minimo per cui anche la condizione di convergenza all'interno della *trust region* nera è soddisfatta (figura 5.13g) e quindi tale regione viene nuovamente centrata nel minimo trovato. A questo punto (figure 5.13h e 5.13i) il processo di ottimizzazione continua, riuscendo a catturare il minimo globale.

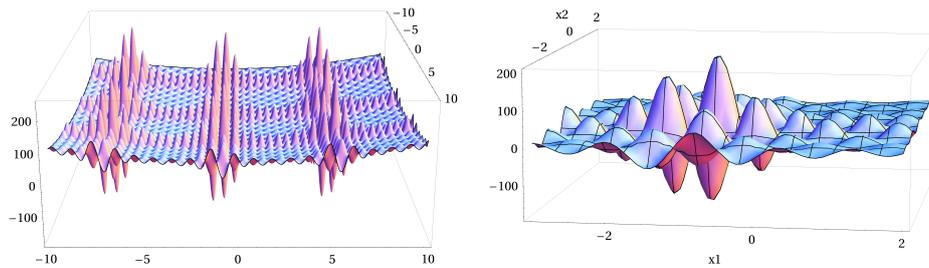


Figura 5.14: $f(x, y) = \sum_{i=1}^5 (i \times \cos((i + 1)x + i) + \sum_{i=1}^5 (i \times \cos((i + 1)y + i) + 0.5((x + 1.42513)^2 + (y + 0.8)^2)$

Rilevato, analizzato e motivato tale comportamento, per verificare se anche in altri casi AMMO fallisse al contrario di MAMMO, è stato pensato di testare i due metodi su di una funzione con moltissimi minimi, di cui uno solo globale, quella mostrata in figura 5.14 e cioè un “campo” di coseni a cui è stato sommato un paraboloide, in modo da avere un unico minimo globale. Tale minimo globale è:

$$x_{min} = -1.425, y_{min} = -0.8 \longrightarrow f(x_{min}, y_{min}) = -123.2$$

ed anche in questo caso, come mostrato in figura 5.15, AMMO si ferma sul minimo locale più vicino al punto iniziale.

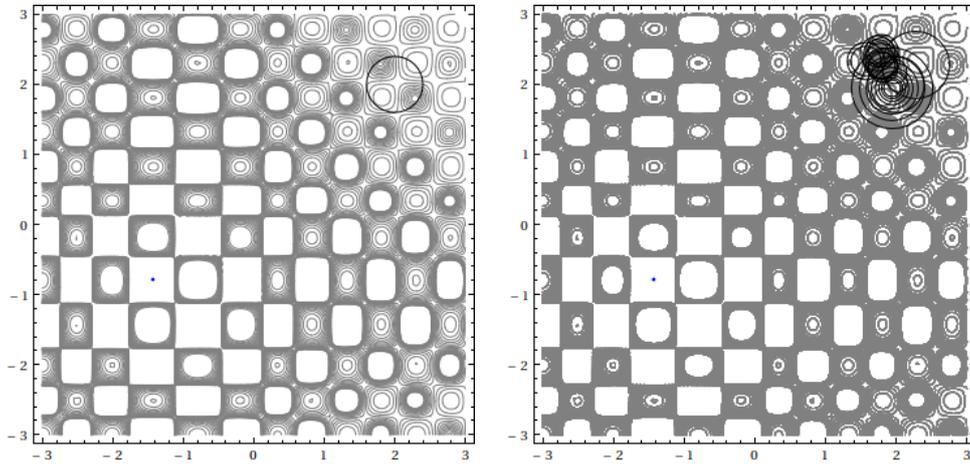


Figura 5.15: AMMO - minimo locale

Utilizzando il metodo MAMMO, anche in un *test case* complesso come questo, si riesce a trovare il minimo globale e l'evoluzione delle due *trust region* è quella presentata in figura 5.16 (la rappresentazione statica di tale evoluzione non è di facile interpretazione e viene presentata al solo fine di far capire quale sia il punto iniziale e di come nell'iterazione finale le due *trust region* circondino effettivamente il punto di minimo giusto).

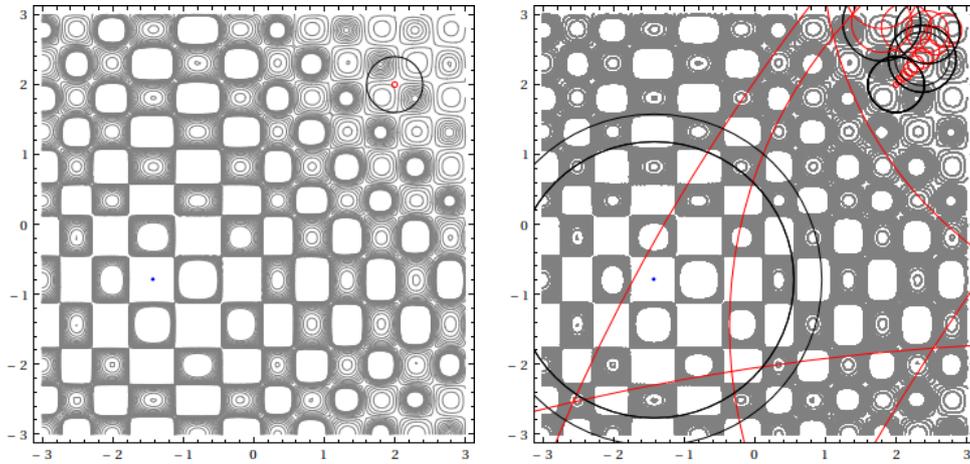


Figura 5.16: MAMMO: minimo globale

I risultati ottenuti in termini di chiamate ai modelli sono:

- Ottimizzazione diretta:
 - funzione alta fedeltà: 191;
- Metodologia MAMMO:
 - funzione alta fedeltà: 96;
 - funzione media fedeltà (serie di Taylor troncata al 6° ordine) : 132;
 - funzione bassa fedeltà (serie di Taylor troncata al 2° ordine): 46632.

Capitolo 6

Considerazioni sui tempi

I risultati presentati nel capitolo precedente mostrano come l'approccio multi-livello permetta di ridurre il numero di chiamate al modello ad alta fedeltà in tutti i *test case* esaminati. Questo è ottenuto al costo di un maggior numero di chiamate al modello a bassa fedeltà e l'introduzione di ulteriori chiamate ad un modello a fedeltà media. I test presentati sono stati eseguiti completamente su funzioni analitiche. In vista dell'uso di tale approccio nella progettazione reale, l'analisi del metodo in termini di effetti attesi sul tempo di simulazione globale è fondamentale.

Indicando con T^h , T^m and T^l il tempo computazionale necessario per una simulazione ai modelli, rispettivamente, ad alta, media e bassa fedeltà, si ha che i tempi necessari nei vari processi di ottimizzazione sono:

- ottimizzazione diretta

$$T = N^*T^h \tag{6.1}$$

- utilizzando metodologia AMMO

$$T_{AM} = N^hT^h + N^lT^l \tag{6.2}$$

- utilizzando metodologia MAMMO

$$T_{MA} = \hat{N}^h T^h + \hat{N}^m T^m + \hat{N}^l T^l \quad (6.3)$$

in cui N^* indica il numero di chiamate al modello ad alta fedeltà (l'unico disponibile) nell'ottimizzazione diretta, N^h e N^l quelle, utilizzando il metodo AMMO, ai modelli ad alta e bassa fedeltà rispettivamente e \hat{N}^h , \hat{N}^m , \hat{N}^l le chiamate ai tre tipi di modelli impiegati nel metodo MAMMO.

Allo scopo di stimare il beneficio in termini di tempi di calcolo, si deve fare qualche considerazione rispetto ai tempi di calcolo relativi ai vari modelli utilizzati e soprattutto rispetto al rapporto tra essi. Si cercherà di stabilire quando sia preferibile utilizzare un metodo anziché un altro. Per fare ciò verranno effettuati i confronti:

- AMMO *versus* ottimizzazione diretta;
- MAMMO *versus* ottimizzazione diretta;
- AMMO *versus* MAMMO;

le condizioni trovate verranno messe a sistema e verranno prodotti dei grafici con ascisse ed ordinate indicanti i diversi tempi di calcolo al fine di individuare in tale dominio la regione in cui un metodo è conveniente rispetto ad un altro.

AMMO *versus* ottimizzazione diretta

Utilizzare AMMO anziché l'ottimizzazione diretta risulta vantaggioso se $\frac{T_{AM}}{T} < 1$ e quindi se:

$$\frac{N^h T^h + N^l T^l}{N^* T^h} < 1 \quad (6.4)$$

che, svolgendo i calcoli, porta a:

$$T^{lh} < \frac{N^* - N^h}{N^l} \quad (6.5)$$

in cui con T^{lh} è indicato il rapporto $\frac{T^l}{T^h}$.

MAMMO *versus* ottimizzazione diretta

L'utilizzo del metodo MAMMO piuttosto che una ottimizzazione diretta è efficace se $\frac{T_{MA}}{T} < 1$ e quindi se:

$$\frac{\hat{N}^h T^h + \hat{N}^m T^h m + \hat{N}^l T^l}{N^* T^h} < 1 \quad (6.6)$$

Introducendo la variabile T^{mh} uguale al rapporto $\frac{T^m}{T^h}$ si ha:

$$T^{mh} < \frac{N^* - \hat{N}^h}{\hat{N}^m} - \frac{\hat{N}^l}{\hat{N}^m} T^{lh} \quad (6.7)$$

AMMO *versus* MAMMO

Infine, è conveniente utilizzare il metodo oggetto del presente lavoro piuttosto che l'approccio classico se $\frac{T_{MA}}{T_{AM}} < 1$ e cioè se:

$$\frac{\hat{N}^h T^h + \hat{N}^m T^h m + \hat{N}^l T^l}{N^h T^h + N^l T^l} < 1 \quad (6.8)$$

che porta a:

$$T^{mh} < \frac{N^h - \hat{N}^h}{\hat{N}^m} - \frac{\hat{N}^l - N^l}{\hat{N}^m} T^{lh} \quad (6.9)$$

Le tre equazioni 6.5, 6.7, 6.9 vengono messe a sistema:

$$\begin{cases} T^{lh} < \frac{N^* - N^h}{N^l} \\ T^{mh} < \frac{N^* - \hat{N}^h}{\hat{N}^m} - \frac{\hat{N}^l}{\hat{N}^m} T^{lh} \\ T^{mh} < \frac{N^h - \hat{N}^h}{\hat{N}^m} - \frac{\hat{N}^l - N^l}{\hat{N}^m} T^{lh} \end{cases} \quad (6.10)$$

e le tre rette rappresentanti le equazioni vengono graficate in un piano cartesiano che ha come ascisse T^{lh} e come ordinate T^{mh} .

In figura 6.1 è mostrato tale tipo di grafico per la funzione presentata nella figura 5.2. Ovviamente, dato che stiamo parlando di tempi, viene analizzato solo il primo quadrante. In nero è rappresentata la prima equazione del sistema 6.10, in rosso la seconda e in blu la terza. La parte a sinistra della retta verticale nera identifica, quindi, la zona in cui, anzichè una ottimizzazione diretta, è conveniente utilizzare la metodologia AMMO, la regione colorata di rosa è quella in cui è conveniente utilizzare MAMMO (sempre rispetto ad una ottimizzazione diretta), mentre la regione in viola identifica la zona in cui il metodo MAMMO si fa preferire alla metodologia classica. Il fatto che si

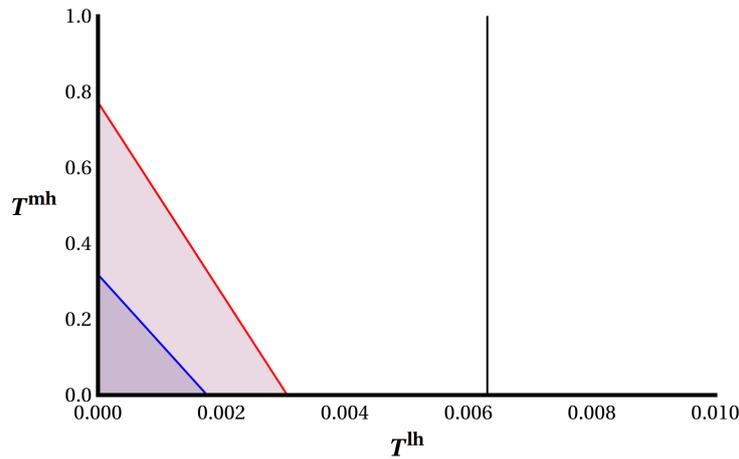


Figura 6.1: Rapporto tempi di esecuzione per $f(x, y) = x3^x + 5y^2$

abbia a che fare con numeri così piccoli non deve stupire, così come non deve far pensare che l'utilizzo di un approccio che si basi su modelli approssimati non sia poi così efficace. Innanzitutto, prendendo come esempio la disciplina *fluidodinamica*, basti pensare che se come modello ad alta fedeltà viene utilizzato un modello RANS, questo può impiegare anche più di un mese per trovare la soluzione, mentre se come modello approssimato viene scelto un modello di Eulero incomprimibile risolto ad esempio con un metodo BEM, i tempi di soluzione possono essere di pochi minuti, quindi per avere un'idea

qualitativa:

$$\frac{T^l}{T^h} \cong 10^{-5} \quad (6.11)$$

e quindi siamo abbondantemente nella zona viola. Inoltre, c'è da sottolineare che l'ottimizzatore che viene utilizzato è basato su metodi euristici e quindi il numero di chiamate al modello a bassa fedeltà indicato per tutti i *test case* non è assolutamente un valore deterministico. Tra gli sviluppi futuri del presente lavoro, c'è anche quello di intervenire sull'ottimizzatore variando, ad esempio, numero iniziale di individui, numero di generazioni massime, ϵ di convergenza, al fine di verificare se sia possibile diminuire il numero di chiamate al modello a bassa fedeltà ottenendo ugualmente il valore di minimo esatto. Questo porterebbe, di conseguenza, ad un valore più alto dei rapporti tra i tempi e quindi ad una maggior libertà di scelta del progettista sul metodo da utilizzare.

Tornando ad analizzare il rapporto tra i tempi, in figura 6.2 è mostrato il grafico $T^{lh} - T^{mh}$ per tutti i *test case* in cui è possibile analizzare anche il metodo AMMO, mentre in figura 6.4 sono presentati i due casi in cui l'unico confronto possibile è quello tra la metodologia presentata ed una ottimizzazione diretta. Tutti i grafici suggeriscono delle considerazioni.

6. Considerazioni sui tempi

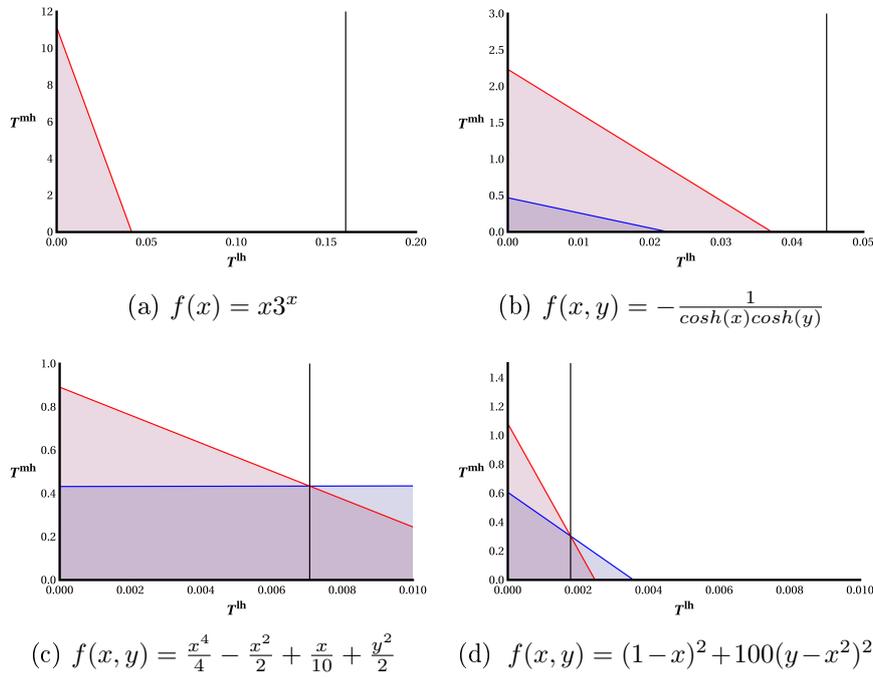


Figura 6.2: Rapporto tempi in ottimizzazione diretta, AMMO e MAMMO

Nel primo caso, la zona in cui è conveniente utilizzare MAMMO anziché AMMO è talmente piccola rispetto a quella in cui AMMO è conveniente rispetto all'ottimizzazione diretta che, utilizzando una scala delle ascisse che permetta di contenerle entrambe, la zona viola sembrerebbe risultare assente. In realtà effettuando uno zoom intorno all'origine (figura 6.3) si nota come tale regione esista, con valori comunque in linea con quanto motivato nell'eq. 6.11.

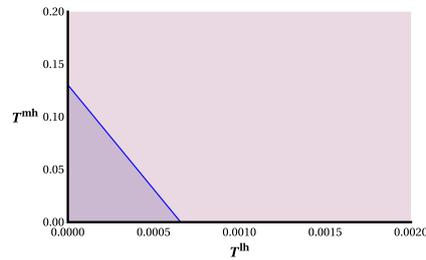


Figura 6.3: Zoom figura 6.2a

Dall'analisi della figura 6.2b risulta in primo luogo evidente come i rapporti tra i tempi aumentino, dando quindi maggior possibilità di scelta al progettista. Un altro aspetto da evidenziare è il fatto che la retta rossa incontra l'asse delle ordinate per un valore > 1 . Questo sta ad indicare che utilizzando il metodo MAMMO anziché un'ottimizzazione diretta, il numero di chiamate ai modelli diminuisce così sensibilmente da poter permettere di avere guadagni in termini di tempi di calcolo, anche se il tempo di soluzione del modello a fedeltà media risultasse maggiore di quello del modello ad alta fedeltà. Questa nuova osservazione potrebbe aprire tutto un'altro settore da indagare all'interno dell'ottimizzazione multi-fedeltà che al momento esula dal presente lavoro. Nelle figure 6.2c e 6.2d si può invece notare come la regione in cui è conveniente utilizzare la metodologia multi-livello aumenta sensibilmente.

Per completezza di informazione, in figura 6.4 sono indicate le regioni per cui è efficace utilizzare la metodologia MAMMO anziché l'ottimizzazione diretta, per i casi in cui la metodologia classica fallisce.

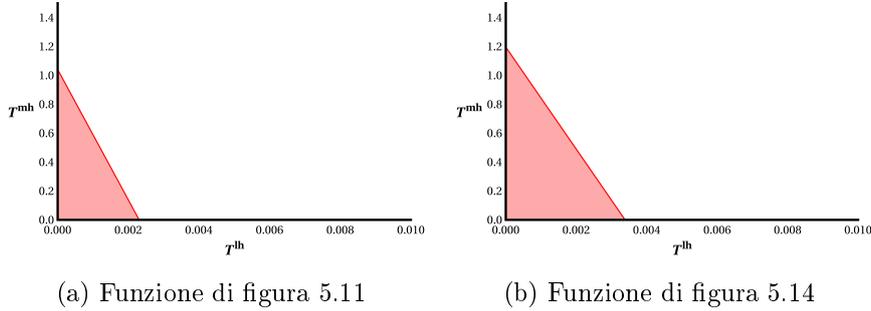


Figura 6.4: Rapporto tempi tra modelli alta, media e bassa fedeltà

Dalle analisi svolte nel presente capitolo non resta che concludere che, al momento, il miglioramento dei tempi di calcolo ottenibile utilizzando la metodologia corrente è fortemente *problem dependent*, fermo restando, però, una diminuzione del numero di chiamate al modello ad alta fedeltà per tutti i casi esaminati.

6.1 Considerazioni su ulteriori potenzialità del metodo MAMMO

Come accennato all'inizio del capitolo 4 l'idea dell'introduzione di un modello a media fedeltà è stata concepita non solo per la riduzione delle chiamate al modello ad alta fedeltà, ma anche perchè, in un problema di tipo progettistico, tale introduzione potrebbe permettere di alleviare una delle criticità del metodo AMMO, cioè l'enorme differenza che ci può essere tra il modello ad alta fedeltà e quello a bassa nel descrivere un fenomeno fisico, che potrebbe impedire al metodo multi-fedeltà di trovare il minimo esatto.

Per chiarire questo concetto è opportuno fare riferimento ad un esempio riguardante, ancora una volta, la disciplina *fluidodinamica*. Un velivolo vola in aria (quindi fluido comprimibile) ad una velocità di circa $Mach = 0.8$,

cioè in regime transonico. Al fine di ottimizzare una qualsiasi grandezza di interesse del velivolo stesso, per eseguire una analisi il più coerente possibile con la realtà, andrebbe utilizzato, come minimo, un metodo *RANS* compressibile che permette di risolvere le equazioni di Navier-Stokes assumendo che il moto turbolento sia composto da una parte media e da una fluttuante ed è in grado di catturare le interazioni tra strato limite e urti. Si ricorda che un'analisi di questo tipo, in un processo di ottimizzazione, potrebbe essere ripetuta anche centinaia di volte e i tempi di calcolo per ogni iterazione potrebbero essere alquanto onerosi. Quindi, nel processo di ottimizzazione multi-fedeltà, si potrebbe pensare di utilizzare un modello meno accurato, quale ad esempio un metodo BEM che permetta di risolvere l'equazione di Eulero. In questo caso si avrebbe a che fare con un flusso incompressibile inviscido, il che si allontana sensibilmente dal fenomeno in analisi. Ciò potrebbe impedire al metodo AMMO di convergere al minimo esatto o ridurre talmente la dimensione della *trust region* da rendere il metodo inapplicabile. Con la metodologia mostrata nel presente lavoro, questa difficoltà può essere superata utilizzando modelli di accuratezza intermedia tra i due precedenti, come per esempio quello di flusso incompressibile viscoso o quello di flusso non viscoso compressibile, entrambi risolti con un metodo BEM. Il primo consentirebbe di catturare tutti gli effetti legati alla viscosità, quale la resistenza di forma, mentre il secondo consentirebbe di tenere conto, ad esempio, degli urti. Inoltre, per quanto indicato a proposito della figura 6.2b, si potrebbe anche profilare l'eventualità di utilizzare questi ultimi due come modelli multipli ad alta fedeltà, nel senso che potrebbero essere utilizzati alternativamente in un processo MAMMO facendo le veci del modello ad alta e media fedeltà e mantenendo come modello a bassa fedeltà quello di flusso incompressibile non viscoso.

Capitolo 7

Conclusioni

È stata presentata una metodologia che può essere utilizzata in ogni problema di ottimizzazione per trovare il minimo globale e che consente di risparmiare tempo di calcolo. Risparmio di tempo, in questo contesto, significa ridurre le chiamate dell'ottimizzatore alla funzione da minimizzare.

Il metodo si basa sulla nota metodologia AMMO, introducendo un terzo modello con fedeltà intermedia e utilizzando quindi funzioni a tre livelli di fedeltà: alta, media e bassa. L'introduzione della nuova funzione implica l'uso di una nuova *trust region* e di due diversi parametri di merito invece di uno. La gestione e l'aggiornamento iterativo di queste entità permette di trovare il minimo del modello ad alta fedeltà facendo uso di funzioni meno accurate. Per convalidare la bontà del metodo, è stato testato su diverse funzioni matematiche di complessità crescente. Inizialmente su una funzione unidimensionale, poi su casi bidimensionali, prima monomodali, poi multimodali. In ogni caso i risultati ottenuti sono in linea con quanto atteso cioè vi è un considerevole risparmio in termini di chiamate al modello ad alta fedeltà. Inoltre, nella parte finale del capitolo riservato ai risultati sono presentati due casi in cui la metodologia classica sembra fallire, trovando un minimo

locale, mentre la nuova metodologia è in grado di trovare il minimo globale. Un capitolo è stato dedicato ad osservazioni riguardanti i tempi di calcolo dei vari processi di ottimizzazione e, soprattutto, i rapporti tra i vari tempi. E' stato concluso che in tutti i *test case* presentati esiste una zona nel dominio dei tempi in cui è preferibile utilizzare la metodologia presentata anziché quella classica. Tra gli sviluppi futuri sono previste analisi di sensibilità al variare dei vari parametri in gioco:

- accuratezza delle funzioni a media e bassa fedeltà;
- punto iniziale da cui far partire l'ottimizzazione;
- dimensioni iniziali delle *trust region*;
- criteri di aggiornamento delle *trust region*;
- ... e quanto altro possa essere interessante per ridurre ulteriormente il numero di chiamate al modello ad alta fedeltà.

Inoltre, seguendo l'approccio indicato nell'ultimo capitolo, sarebbe interessante riuscire a trovare una qualche regola generale per cui, conoscendo a priori i tempi di calcolo dei vari modelli presi singolarmente, il progettista sia in grado di valutare se sia conveniente utilizzare un'ottimizzazione diretta, anziché un metodo AMMO o MAMMO. E' altresì prevista l'applicazione della metodologia presentata in processi di ottimizzazione multidisciplinare di tipo ingegneristico, coinvolgendo quindi i problemi fisici riguardanti le varie discipline, per confermare eventualmente i risultati estremamente promettenti raggiunti nei *benchmark* analitici.

Bibliografia

- [1] N.M. ALEXANDROV, R.M. LEWIS, *An Overview of First-Order Model Management for Engineering Optimization*, Optimization and Engineering, 2001
- [2] R. HAFTKA, *Combining Global and Local Approximation*, AIAA Journal vol.29, 1991
- [3] D. PERI, E.F. CAMPANA, *Variable Fidelity and Surrogate Modelling in Simulation-Based Design*, 27th Symposium on Naval Hydrodynamics, 2008
- [4] N.M. ALEXANDROV, *Multilevel and multiobjective optimization in multidisciplinary design*, AIAA Paper 96-4122, 1996
- [5] N.M. ALEXANDROV, *On managing the use of surrogates in general nonlinear optimization and mdo*, AIAA Paper 98-4798, 1998
- [6] N.M. ALEXANDROV *Optimization with Variable-fidelity Models Applied to Wing Design*, AIAA Paper 2000-0841, 2000
- [7] N.M. ALEXANDROV ET AL. *Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models*, AIAA Journal of Aircraft, vol. 38, 2001

BIBLIOGRAFIA

- [8] N.M. ALEXANDROV ET AL. *First-Order Model Management with Variable-Fidelity Physics Applied to Multi-Element Airfoil Optimizations*, AIAA Paper 2000-4886, 2000
- [9] R.M. LEWIS *A trust region framework for managing approximation models in engineering optimization*, AIAA Papers 96-4101, 1996
- [10] N.M. ALEXANDROV *Robustness properties of a trust region framework for managing approximations in engineering optimization*, AIAA Paper 96-4102, 1996
- [11] J.J. MORÉ *Recent developments in algorithms and software for trust region methods*, Argonne National Lab, Technical Report, 1982

Ringraziamenti

Ringrazio innanzitutto i miei meravigliosi genitori che mi appoggiano in ogni scelta di vita.

Dopodichè voglio ringraziare tutti coloro che nei tre anni e più di questo viaggio hanno significato qualcosa, cominciando, naturalmente, da **mia moglie** Patrizia, compagna di viaggio in un cammino ben più lungo di questo.

Nel ringraziare tutti gli “abitanti” che sono passati dal GRACO nei vari periodi, un ringraziamento particolare va, ovviamente, ai *Titani*, che non sono stati e non sono semplici colleghi, ma amici con cui ho trascorso dei momenti che ricorderò per sempre! Quindi un ringraziamento particolare ad Alessandro, Emanuele, Eugenio ma, certo che loro non se la prenderanno, un grazie speciale va a Paolo, con cui questo viaggio è iniziato circa sette anni fa!!!

Stando più sul pezzo, grazie ad Umberto che mi ha aiutato a portare a termine un lavoro che mi rende immensamente orgoglioso.

Grazie a tutti, ricordando che ... *l'importante non è la meta, ma il viaggio per raggiungerla.*