



UNIVERSITA' DEGLI STUDI DI CAGLIARI

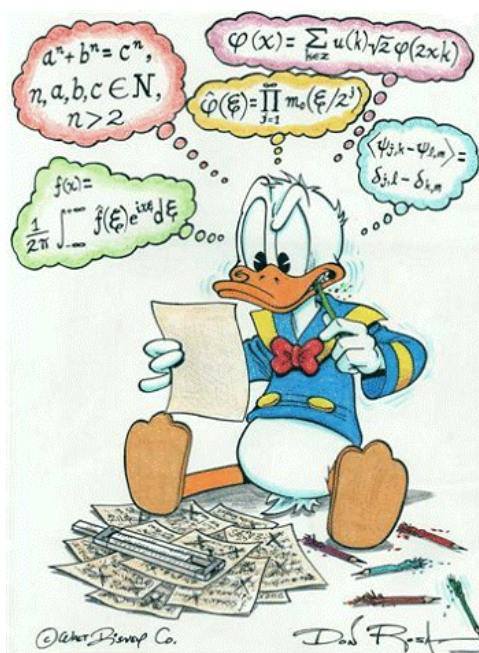
Facoltà di Ingegneria

Dipartimento di Ingegneria Elettrica ed Elettronica



CORSO DI CALCOLO NUMERICO II

ANALISI E CONFRONTO DI METODI PER LA RISOLUZIONE DI EQUAZIONI DIFFERENZIALI ORDINARIE IN MATLAB



Docente: Giuseppe Rodriguez

Studenti:

Mario Porru Matricola 40084

Nicola Usai Matricola 40093

Ingegneria Elettrica

A.A. 2009/2010

INDICE

0. Introduzione	3
1. Metodi monostep	5
2. Metodi multistep	12
3. Analisi del comportamento di <i>eulesp</i> con equazioni rigide	14
4. Analisi del metodo predictor corrector di ordine 2 basato su adams bashfort moulton	15

0. INTRODUZIONE

Le equazioni differenziali sono uno dei più importanti strumenti che l'analisi matematica mette a disposizione nello studio di modelli matematici nei più disparati settori della scienza, dalla fisica all'ingegneria fino ad arrivare all'economia e alla biologia.

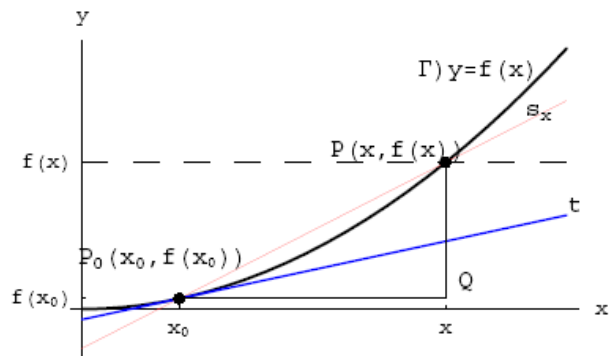
Le equazioni differenziali di ordine superiore al primo e i sistemi di equazioni differenziali possono essere sempre ricondotti, con particolari accorgimenti, in forma di equazioni differenziali ordinarie (ODE). È fondamentale disporre di metodi di risoluzione di queste equazioni.

Nelle scienze applicate le equazioni differenziali da risolvere sono poste sotto forma di problema di Cauchy:

$$\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0. \end{cases}$$

Molte di queste equazioni non possono essere risolte analiticamente, perciò si utilizzano metodi di risoluzione numerica, algoritmi che consentono di approssimare le funzioni soluzione del problema con una successione finita di punti.

Il metodo più semplice e più intuitivo è il metodo delle differenze finite. L'idea base del metodo è quella di sostituire alla derivata il rapporto incrementale.



$$y'(x_i) = f(x_i, y(x_i)) \cong \frac{y(x_{i+1}) - y(x_i)}{h}.$$

Per ottenere l'uguaglianza si sostituiscono le valutazioni della $y(x_i)$ con una loro stima η_i :

$$f(x_i, \eta_i) = \frac{\eta_{i+1} - \eta_i}{h},$$

da cui esplicitando η_{i+1} e inizializzando si ottiene il metodo di Eulero esplicito:

$$\begin{cases} \eta_{i+1} = \eta_i + h \cdot f(x_i, \eta_i) \\ \eta_0 = y_0. \end{cases}$$

Con altri procedimenti, analoghi e non, si possono ottenere anche altri metodi di risoluzione numerica, che genericamente vengono espressi:

$$\eta_{i+r} = - \sum_{j=0}^{r-1} a_j \cdot \eta_{i+j} + h \cdot \Phi(x_i, \eta_i, \dots, \eta_{i+r}).$$

I metodi sono classificati in:

- monostep: la valutazione di η_{i+1} ($r = 1$) coinvolge solo η_i e non le approssimazioni precedenti;

- multistep: la valutazione di η_{i+r} coinvolge non solo η_{i+r-1} ma anche le approssimazioni precedenti;
- espliciti: la valutazione della Φ non comprende il termine η_{i+r} che verrà calcolato;
- impliciti: la valutazione della Φ comprende il termine η_{i+r} .

Il fine di questa relazione è mostrare come sia possibile implementare metodi per la risoluzione di equazioni differenziali ordinarie in ambiente MATLAB, confrontarne le prestazioni e, dato che non esiste un metodo universalmente utilizzabile, capire i pregi e i difetti di ciascun metodo, in modo da poter scegliere il metodo più adatto ad ogni problema.

Il lavoro si suddivide in diverse sezioni:

1. Analisi di metodi monostep.
2. Analisi di metodi multistep.
3. Analisi di metodi Predictor-Corrector, che accoppiano un metodo esplicito (predictor) che fornisce una prima stima di η_{i+1} e uno implicito (corrector) che raffina tale stima.

1. METODI MONOSTEP

I metodi monostep analizzati sono:

- Eulero esplicito, esplicito, monostep, 1 stadio, (funzione **eulesp**);
- Heun, esplicito, monostep, 2 stadi, (funzione **heun**);
- Eulero implicito, implicito, monostep, 1 stadio, (funzione **eulimp** e **eulimpf**);
- Crank-Nicolson, implicito, monostep, 2 stadi, (funzione **gek**).

La loro implementazione in MATLAB è abbastanza semplice ed immediata, e con qualche accorgimento è possibile rendere i metodi applicabili a casi multidimensionali:

- Eulero esplicito
$$\begin{cases} \eta_{i+1} = \eta_i + h \cdot f(x_i, \eta_i) \\ \eta_0 = y_0. \end{cases}$$
- Heun
$$\begin{cases} \eta_{i+1} = \eta_i + \frac{h}{2} \cdot [f(x_i, \eta_i) + f(x_{i+1}, \eta_i + h \cdot f(x_i, \eta_i))] \\ \eta_0 = y_0. \end{cases}$$
- Eulero implicito
$$\begin{cases} \eta_{i+1} = \eta_i + h \cdot f(x_{i+1}, \eta_{i+1}) \\ \eta_0 = y_0. \end{cases}$$
- Crank-Nicolson
$$\begin{cases} \eta_{i+1} = \eta_i + \frac{h}{2} \cdot [f(x_i, \eta_i) + f(x_{i+1}, \eta_{i+1})] \\ \eta_0 = y_0. \end{cases}$$

1.1. Confronto tra i vari metodi monostep espliciti

L'obiettivo di questa analisi è confrontare le prestazioni dei vari metodi monostep espliciti (**eulesp** e **heun**) nella risoluzione dello stesso problema di Cauchy al variare del passo.

Le prove sono state fatte considerando le equazioni I_{ac} , che descrive il transitorio di corrente in un circuito RL con condizioni iniziali di circuito a riposo; e prepre, che descrive il modello preda predatore di Lotka-Volterra. Della prima è nota anche la soluzione analitica, quindi esatta, che può essere utilizzata come confronto con i risultati numerici; mentre per la seconda si confronteranno solo i risultati numerici forniti dai due metodi.

Il problema da risolvere è:

$$\begin{cases} \frac{di_{ac}}{dt} + \frac{R}{L} \cdot i_{ac} = \frac{1}{L} \cdot V_m \cdot \cos(\omega \cdot t + \varphi) \\ i_{ac}(0) = 0; \end{cases}$$

la soluzione analitica è:

$$i_{ac} = N \cdot \cos(\omega \cdot t + \varphi) + M \cdot \sin(\omega \cdot t + \varphi) + P \cdot e^{-\frac{R}{L}t};$$

con:

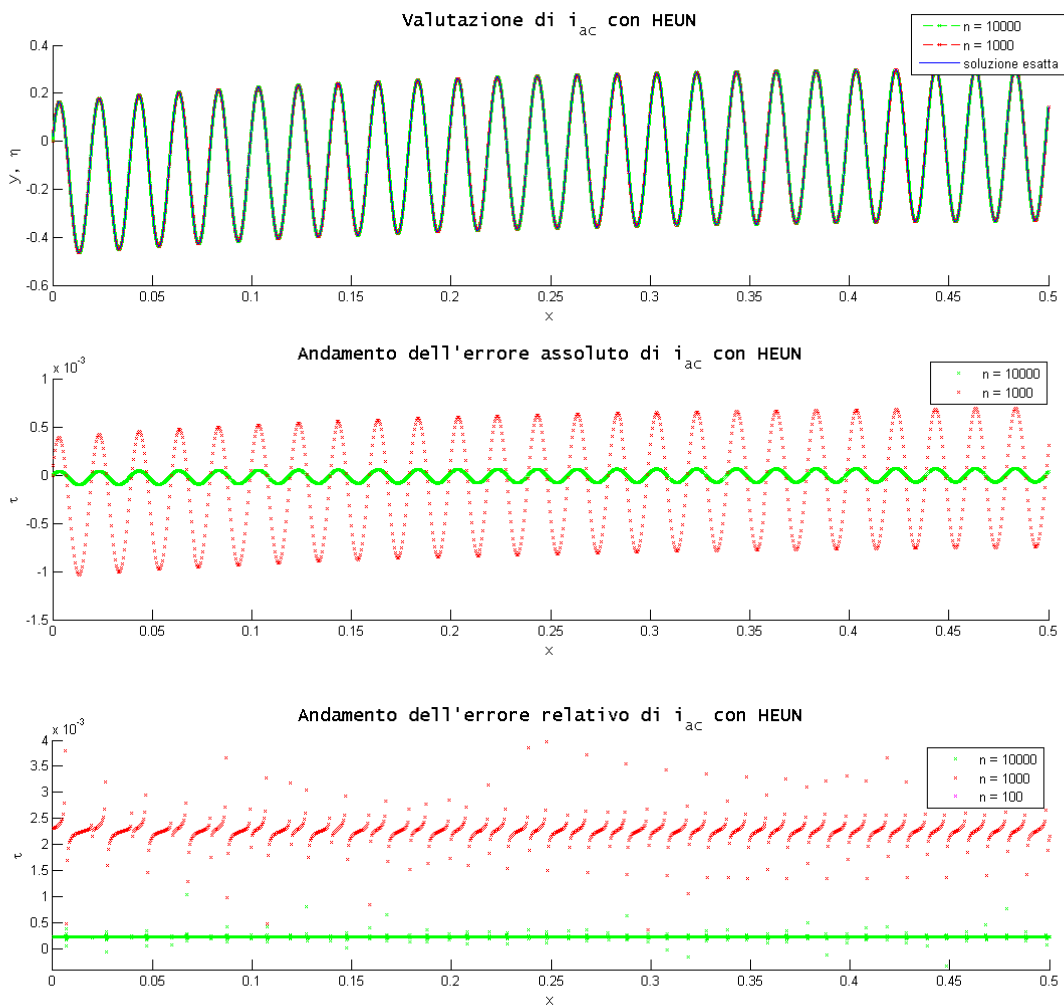
$$M = \frac{V_m \cdot \omega}{L \cdot \omega^2 + R}; N = \frac{V_m}{L \cdot \omega^2 + R} \cdot \frac{R}{L}; P = -N \cdot \cos(\varphi) - M \sin(\varphi).$$

Le prove sono state fatte integrando nell'intervallo $I_0 = [0; 0,5]$ utilizzando 100, 1000 e 10000 punti di valutazione, sia per **eulesp** che per **heun**. Il confronto tra i diversi metodi, al variare del passo, è riassunto nella seguente tabella:

Confronto tra i diversi metodi espliciti, al variare del passo, nella valutazione di i_{ac}

Nome metodo	Passo	Tempo risoluzione	Errore assoluto	Errore relativo
EULESP	5.05051e-003	3.55688e-003	2.37520e-001	5.08157e+001
EULESP	5.00501e-004	1.40095e-002	2.31965e-002	2.15016e+001
EULESP	5.00050e-005	1.33377e-001	2.31473e-003	1.00800e+001
HEUN	5.05051e-003	3.09027e-003	6.46447e-002	8.18336e-001
HEUN	5.00501e-004	3.03167e-002	6.91039e-004	5.21790e-002
HEUN	5.00050e-005	2.99094e-001	6.79295e-005	1.03476e-003

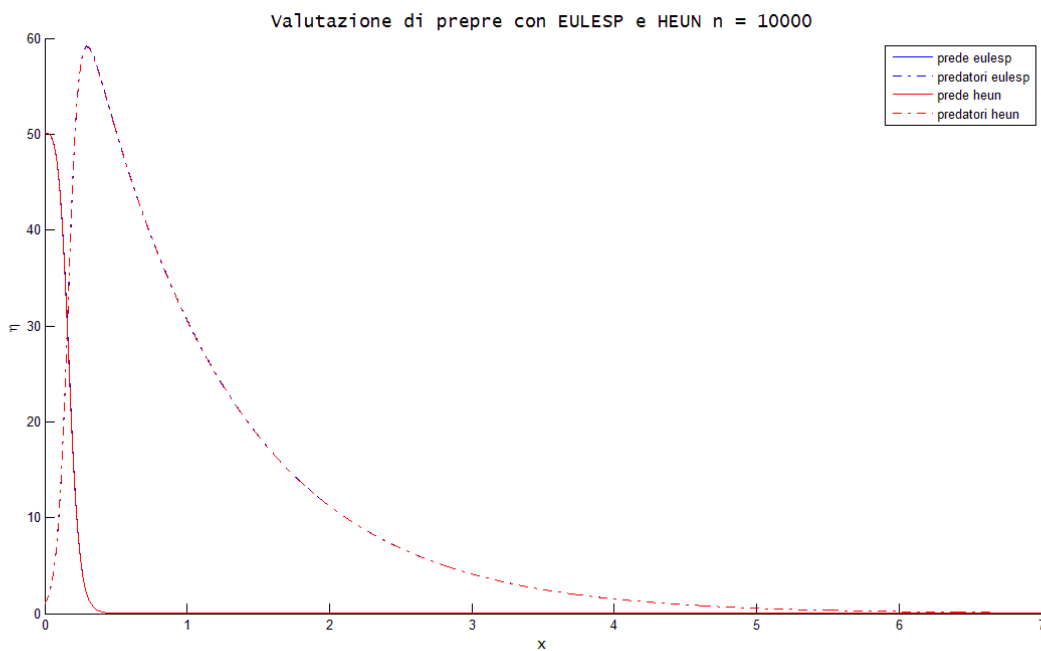
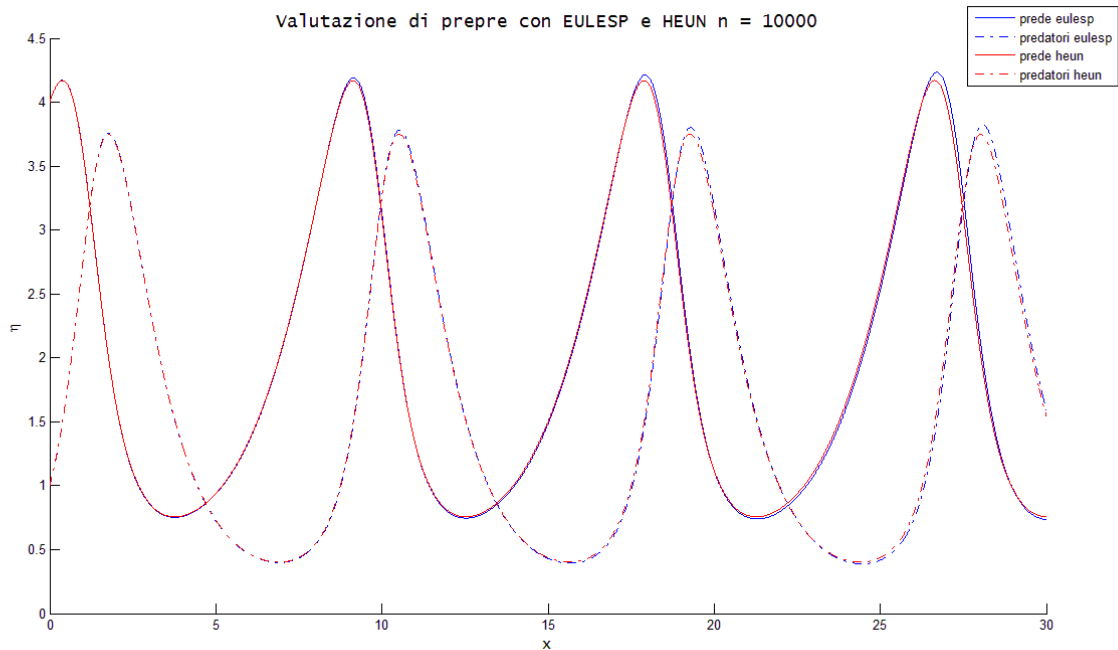
Come si può vedere, i risultati confermano quel che era già noto dalla teoria: il metodo di Heun impiega circa il doppio del tempo, rispetto a *eulesp*, per la risoluzione dello stesso problema a parità di passo, in quanto è a due stadi. Il beneficio si riscontra nella precisione, dato che l'errore assoluto è di un ordine di grandezza inferiore.



Nel grafico precedente sono riportati l'andamento della soluzione esatta e delle sue approssimazioni e dei rispettivi errori assoluti e relativi con $n = 1000$ e $n = 10000$. Si può vedere come l'errore assuma un andamento periodico che ricalca quello della funzione studiata e come il suo valore diminuisca al diminuire del passo. Questo è confermato dalla pressoché costanza dell'errore relativo, come si nota dall'ultimo dei grafici soprastanti. Come si può vedere la valutazione dell'errore relativo può presentare problemi in alcuni punti per via dell'annullarsi della funzione che lo fa esplodere.

Il secondo test dei metodi espliciti è stato eseguito sul modello Lotka-Volterra, preda predatore, per il quale non esiste una soluzione esatta. In compenso si può fare un confronto tra i due metodi espliciti. I casi analizzati sono due: un caso dove si raggiunge l'equilibrio biologico tra le specie e un altro caso dove si ha l'estinzione di entrambe. Questo fatto è dovuto al numero iniziale di prede e di predatori.

Il primo test è quello di valutare la funzione al variare del numero di punti. Si è riscontrato che con $n = 100$ la soluzione è inattendibile perché troppo differente da quelle per $n = 1000$ e $n = 10000$ che sono, invece, molto simili.

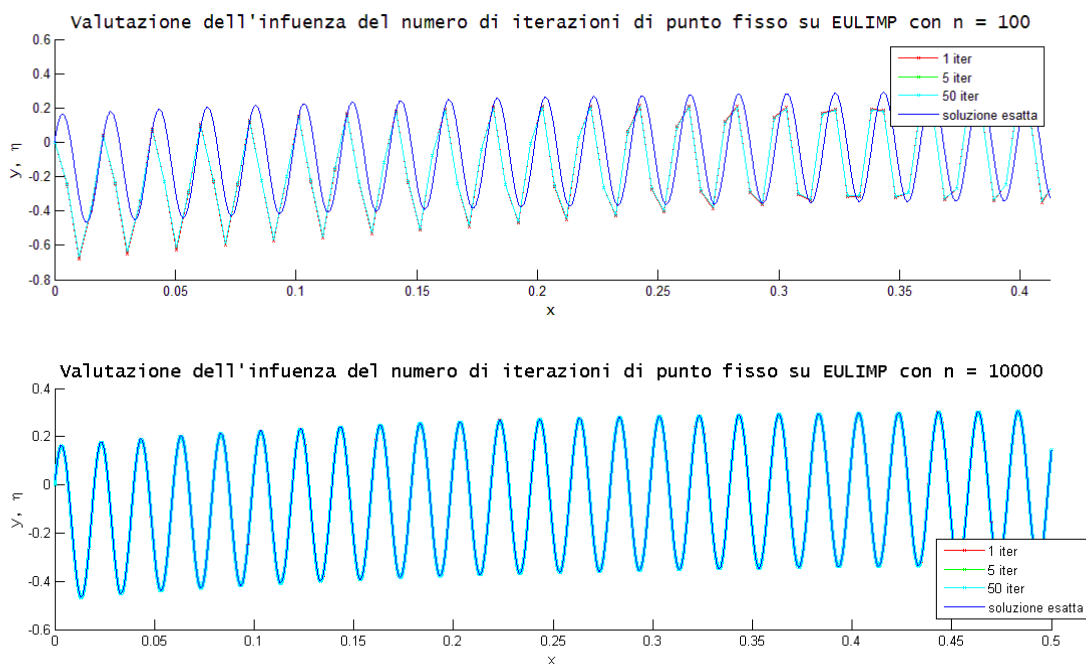


Un'ulteriore considerazione si può trarre dal primo grafico: i due metodi forniscono due soluzioni molto vicine, a parità di passo, il metodo **eulesp** tende a divergere e dato che il metodo di Heun è del second'ordine risulta più preciso.

1.2. Metodi monostep impliciti

I metodi impliciti di risoluzione delle equazioni differenziali implementati e analizzati sono: il metodo di Eulero implicito e il metodo di Crank-Nicolson. Anche in questo caso l'analisi consiste nel confrontare i risultati ottenuti con questi metodi con la soluzione, nota, del problema di Cauchy.

L'implementazione dei metodi espliciti non presenta la difficoltà che invece si incontra nell'implementazione dei metodi impliciti che prevedono, al loro interno, la risoluzione di un'equazione non lineare $\Psi(\eta_{i+1}) = 0$ per la determinazione di η_{i+1} . Il modo più intuitivo sarebbe quello di utilizzare la apposita funzione MATLAB *fsolve*, adatta per la risoluzione di sistemi di equazioni non lineari. In alternativa si può sfruttare la contrattività delle funzioni e risolverla con iterazioni di punto fisso. Nelle analisi il metodo di Eulero implicito è stato implementato con entrambe le soluzioni. Il metodo di Crank-Nicolson è stato realizzato solo utilizzando le iterazioni di punto fisso. Per questa seconda soluzione è necessario capire quante iterazioni deve effettuare il metodo per convergere alla soluzione. Il numero di iterazioni implementate per lo studio sono 1, 5 e 50. Il confronto con la soluzione analitica del problema dimostra che non esiste una apprezzabile differenza nei risultati se il risolutore fa 1, 5 o 50 iterazioni, i risultati numerici sono molto simili, la sostanziale e notevole differenza sta nel tempo di calcolo della funzione infatti per eseguire 50 iterazioni rispetto a 1 o 5 il calcolatore richiede un tempo di calcolo che è circa 10 volte superiore. Si può vedere come all'aumentare n , quindi al diminuire di h , la soluzione numerica approssima sempre meglio quella esatta indipendentemente dal numero di iterazioni di punto fisso. Questo lo si nota dai grafici sottostanti ottenuti per $n = 100$ e per $n = 10000$: il parametro fondamentale che influisce sulla precisione della soluzione numerica non è il numero di iterazioni, ma il passo h . Per questo i metodi di default eseguiranno solo 5 iterazioni di punto fisso.

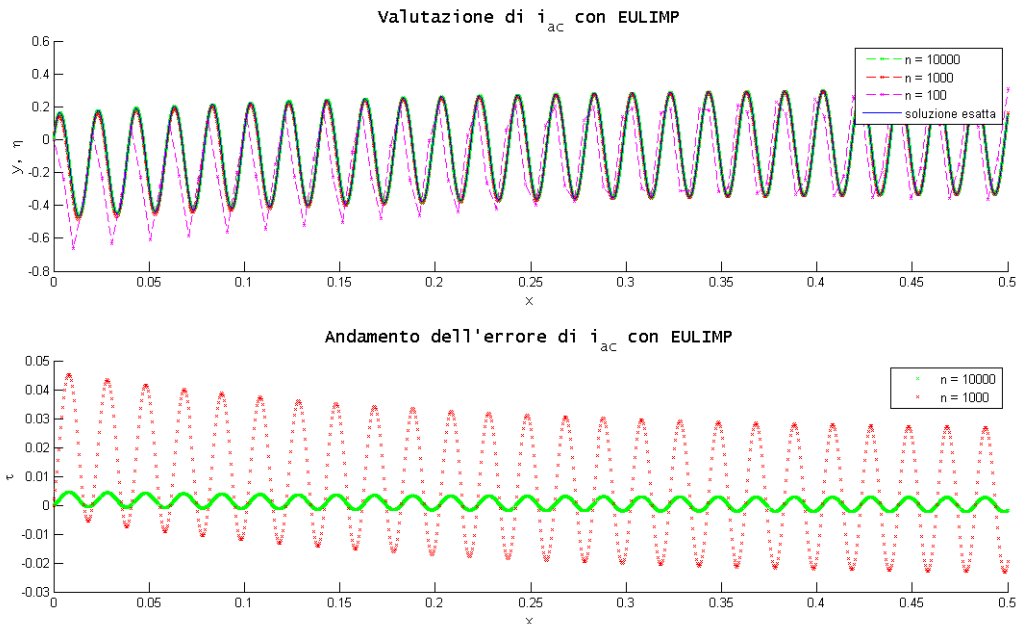


Una prima prova è stata fatta con la stessa funzione del transitorio di corrente i_{ac} , per poter confrontare i risultati con i metodi espliciti analizzati in precedenza. I risultati ottenuti utilizzando il metodo di Eulero implicito (nelle sue due implementazioni) e il metodo di Crank-Nicolson sono:

Confronto tra i diversi metodi impliciti, al variare del passo, nella valutazione di i_{ac}

Nome metodo	Passo	Tempo risoluzione	Errore massimo
EULIMP	5.05051e-003	1.06208e-002	3.64246e-001
EULIMP	5.00501e-004	8.69348e-002	4.52961e-002
EULIMP	5.00050e-005	8.24093e-001	4.53593e-003
EULIMPF	5.05051e-003	6.79362e-001	3.64246e-001
EULIMPF	5.00501e-004	4.07394e+000	4.52961e-002
EULIMPF	5.00050e-005	4.00782e+001	4.53593e-003
GEK	5.05051e-003	8.87943e-002	6.37478e-002
GEK	5.00501e-004	1.14094e-001	6.86844e-004
GEK	5.00050e-005	8.31640e-001	6.79009e-005

Il metodo di Eulero fornisce gli stessi risultati sia con l'implementazione del punto fisso sia con *fsolve*, che sono molto vicini alla soluzione analitica. L'unica differenza è il tempo di risoluzione, il tempo richiesto da *eulimpf* è fino a due ordini di grandezza superiore. Questo porta a concludere che i metodi impliciti possono essere efficacemente e efficientemente implementati utilizzando l'iterazione di punto fisso e non pesanti algoritmi per la risoluzione di sistemi non lineari. Tuttavia in entrambi i casi per $n = 100$ si ha un'approssimazione errata. All'aumentare del numero di passi l'errore sulla soluzione diminuisce.



Il metodo di Crank-Nicolson restituisce risultati sostanzialmente paragonabili ma consente di risolvere lo stesso problema commettendo errori, a parità di passo, alcuni ordini di grandezza inferiori; il tutto senza un aumento eccessivo del tempo di calcolo.

L'altro test eseguito è sul problema di Cauchy, che descrive il transitorio della tensione quando viene chiusa una linea a vuoto:

$$\begin{cases} \frac{d^2 v_c}{dt^2} + \mu_0^2 \cdot v_c = \mu_0^2 \cdot E_m \cdot \cos(\omega \cdot t) \\ v_c(0) = 0; \quad \left. \frac{dv_c}{dt} \right|_{t=0} = 0; \end{cases}$$

la soluzione analitica è:

$$v_c = V_m \cdot \cos(\omega \cdot t) - V_m \cdot \cos(\mu_0 \cdot t);$$

con:

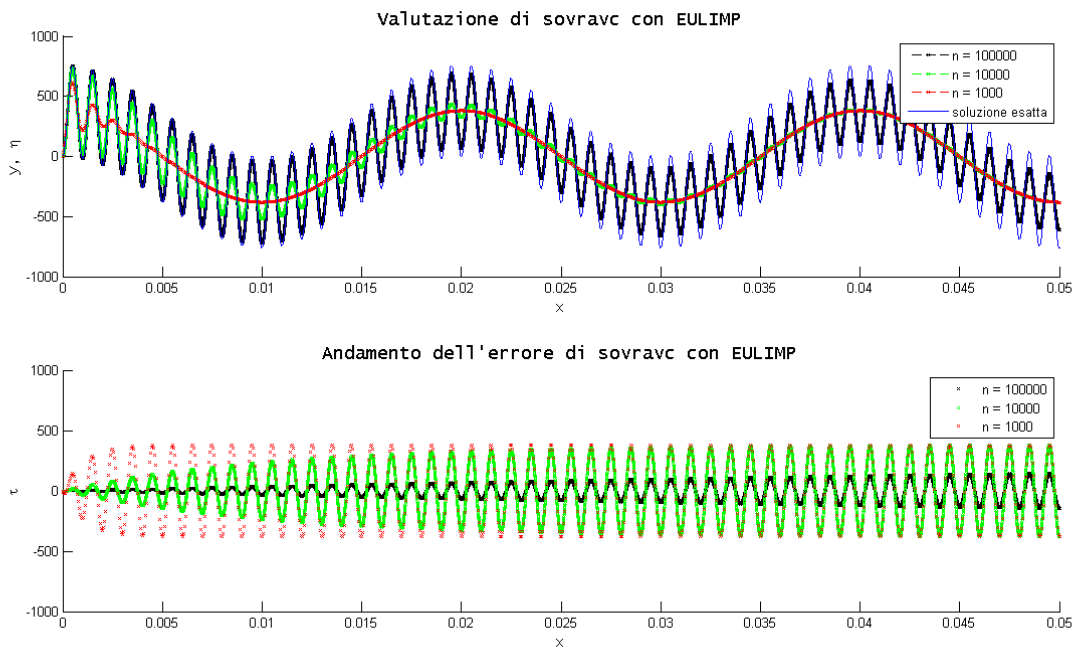
$$V_m = \frac{E_m}{1 - \frac{\omega^2}{\mu_0^2}}$$

Le prove sono state fatte integrando nell'intervallo $I_0 = [0; 0,05]$ utilizzando 100, 1000 e 10000 punti di valutazione, sia per *eulimp* che per *eulimpf* che per *gek*. Per *eulimp* e *gek* vengono utilizzati anche 100000. Il confronto tra i diversi metodi, al variare del passo, è riassunto nella seguente tabella:

Confronto tra i diversi metodi, al variare del passo, nella valutazione di sovracc

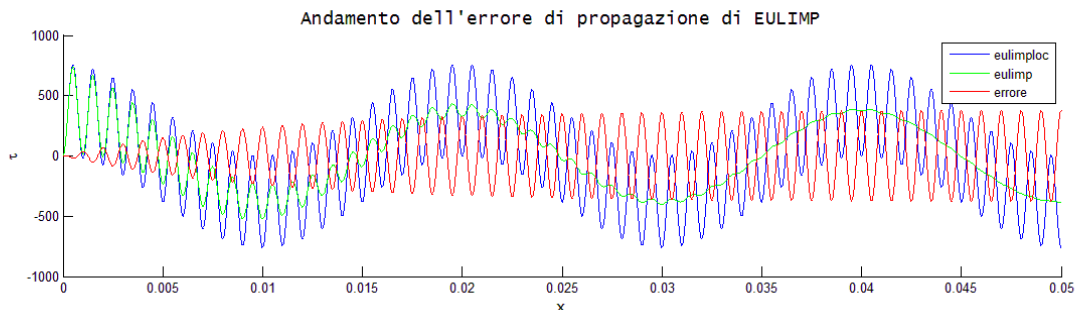
Nome metodo	Passo	Tempo risoluzione	Errore massimo
HEUN	5.00050e-006	2.29346e-001	1.96011e+001
EULIMP	5.05051e-004	6.84032e-003	6.29709e+248
EULIMP	5.00501e-005	4.59122e-002	3.80959e+002
EULIMP	5.00050e-006	4.63359e-001	3.78083e+002
EULIMP	5.00005e-007	4.66864e+000	1.47232e+002
EULIMPF	5.05051e-004	2.06150e+000	4.15597e+002
EULIMPF	5.00501e-005	1.57312e+001	3.80948e+002
EULIMPF	5.00050e-006	1.59778e+002	3.78083e+002
GEK	5.05051e-004	1.11951e-002	7.27265e+121
GEK	5.00501e-005	5.80385e-002	7.37036e+002
GEK	5.00050e-006	5.90351e-001	9.69522e+000
GEK	5.00005e-007	5.63739e+000	9.69585e-002

Il metodo di Eulero (*eulimp*) al passare del tempo smorza la componente ad alta frequenza della soluzione.



Questo comportamento è tanto più evidente quanto più grande è il passo e fa pensare che sia dovuto agli errori di propagazione. Infatti anche lo stesso metodo implementato con *fsolve* al posto delle iterazioni di punto fisso dà gli stessi risultati con l'inconveniente dell'elevato tempo di risoluzione.

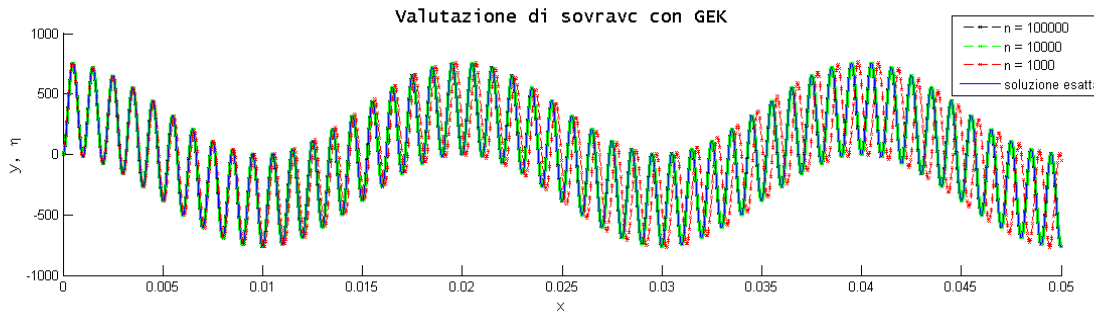
La conferma di questa affermazione è data dal grafico seguente, ottenuto annullando tutti gli errori di propagazione inizializzando ogni passo con il valore esatto della funzione.



Si nota come la soluzione affetta dal solo errore di discretizzazione locali ricalchi la soluzione analitica.

Il problema riscontrato con *gek* è differente: per passi troppo piccoli il metodo non converge, e l'errore tende ad aumentare sempre più. La non convergenza si manifesta con un progressivo aumento dello sfasamento tra la soluzione esatta e quella numerica, come se vi sia un piccolo errore nella determinazione della componente oscillante ad alta frequenza. I test sono stati fatti con $n = 100$, $n = 1000$, $n = 10000$,

$n = 100000$. Il problema si verifica solo nei primi due casi, e non sembra essere dovuto alla non contrattività della funzione: implementato con *fsolve* invece che con la funzione di iterazione si verifica lo stesso fenomeno. Con $n = 10000$ entrambe le realizzazioni di **gek** convergono commettendo errori paragonabili, ma il calcolo con *fsolve* impiega 186 secondi contro gli 0,590 del metodo che sfrutta le iterazioni di punto fisso! Facendo riferimento a quest'ultimo, anche in questo caso **gek** consente di ottenere risultati con errori di diversi ordini di grandezza inferiori rispetto a **eulimp**, a parità di passo, e senza alcuna spesa ulteriore nel tempo di calcolo.



Facendo un confronto a parità di passo, con $n = 10000$, il metodo migliore risulta essere uno proprio quello di Crank-Nicolson, seguito da quello di **heun** che non presenta né smorzamento né differenze di frequenza. Anche **eulesp** garantisce una buona approssimazione, ma solo se il passo è alcuni ordini di grandezza inferiore rispetto a quello impiegato negli altri metodi.

Definito l'errore globale di discretizzazione come:

$$E(h) = \max_{i=1, \dots, n} |y(x_i) - \eta_i|,$$

un metodo è convergente quando $E(h) \rightarrow 0$ per $h \rightarrow 0$ e lo è di ordine p se $E(h) = O(h^p)$. Condizioni necessarie perché il metodo sia convergente sono la consistenza (al tendere del passo a zero l'errore locale di discretizzazione tende a zero) e la stabilità (gli errori non si propagano).

È stato dimostrato che tutte le formule monostep sono stabili e l'ordine di convergenza globale risulta pari all'ordine di convergenza dell'errore locale. Il teorema della barriera di Butcher afferma che per le formule monostep esplicite fino a 4 stadi l'ordine di convergenza coincide con il numero degli stadi. Al crescere dell'ordine di convergenza solo necessari sempre più stadi.

Dato che la complessità computazionale è fortemente influenzata dal numero degli stadi sembra vantaggioso cercare di trovare nuovi metodi numerici che possano aumentare la velocità di convergenza senza però aumentare il numero di stadi. La soluzione a questo problema è l'utilizzo di metodi multistep.

2. METODI MULTISTEP

I metodi multistep analizzati sono:

- Punto medio, esplicito, multistep, 1 stadio, (funzione **pume**);
- Simpson, implicito, multistep, 1 stadio, (funzione **homer**).

La loro implementazione in MATLAB risulta essere:

- Punto medio
$$\begin{cases} \eta_{i+1} = \eta_{i-1} + 2 \cdot h \cdot f(x_i, \eta_i) \\ \eta_0 = y_0. \end{cases}$$
- Simpson
$$\begin{cases} \eta_{i+1} = \eta_{i-1} + \frac{h}{3} \cdot [f(x_{i-1}, \eta_{i-1}) + 4 \cdot f(x_i, \eta_i) + f(x_{i+1}, \eta_{i+1})] \\ \eta_0 = y_0. \end{cases}$$

L'analisi dei risultati dimostrerà che questa seconda soluzione garantisce la stessa precisione con un notevole risparmio di tempo.

2.1. Confronto metodi multistep

Risolvendo entrambi i problemi di Cauchy precedenti con i metodi del punto medio e col metodo di Simpson si vede che le soluzioni sono molto prossime alla soluzione analitica.

Confronto tra i diversi metodi multistep, al variare del passo, nella valutazione di i_{ac}

Nome metodo	Passo	Tempo risoluzione	Errore massimo
PUME	5.05051e-003	3.42145e-003	2.63622e+000
PUME	5.00501e-004	1.65567e-002	1.52868e-002
PUME	5.00050e-005	1.63518e-001	1.55525e-004
HOMER	5.05051e-003	9.94973e-003	3.64788e-001
HOMER	5.00501e-004	9.76147e-002	2.36745e-003
HOMER	5.00050e-005	9.81985e-001	8.26675e-005

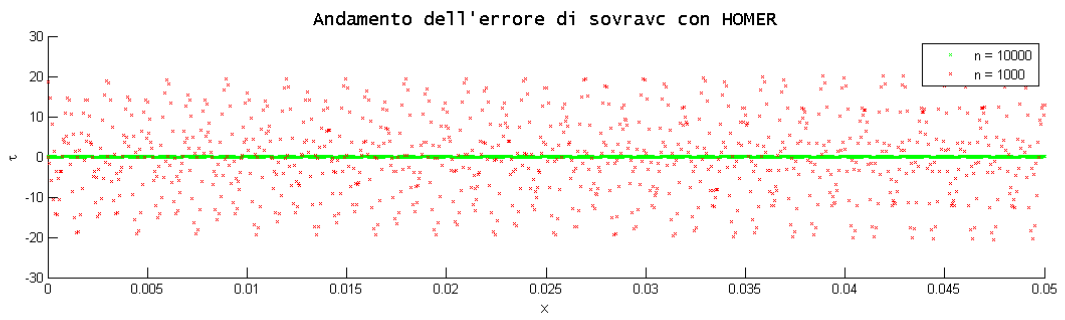
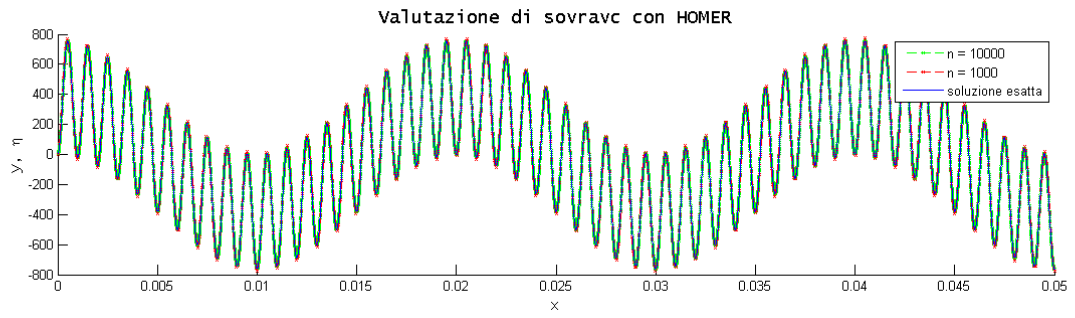
Da questa tabella si vede che il metodo migliore di risoluzione è quello di Simpson (**homer**), l'errore è minore rispetto al metodo del punto medio (**pume**).

Per l'equazione I_{ac} si erano raggiunti dei buoni risultati già con i metodi precedenti, ma con questi metodi multistep l'errore è di 2 ordini di grandezza inferiore a parità di passo.

Confronto tra i diversi metodi multistep, al variare del passo, nella valutazione di $sovravc$

Nome metodo	Passo	Tempo risoluzione	Errore massimo
PUME	5.05051e-004	4.89091e-003	3.59155e+078
PUME	5.00501e-005	9.71534e-003	7.79005e+002
PUME	5.00050e-006	9.31739e-002	1.96018e+001
HOMER	5.05051e-004	1.39205e-002	1.84902e+078
HOMER	5.00501e-005	6.33365e-002	2.07551e+001
HOMER	5.00050e-006	5.83147e-001	1.87603e-001

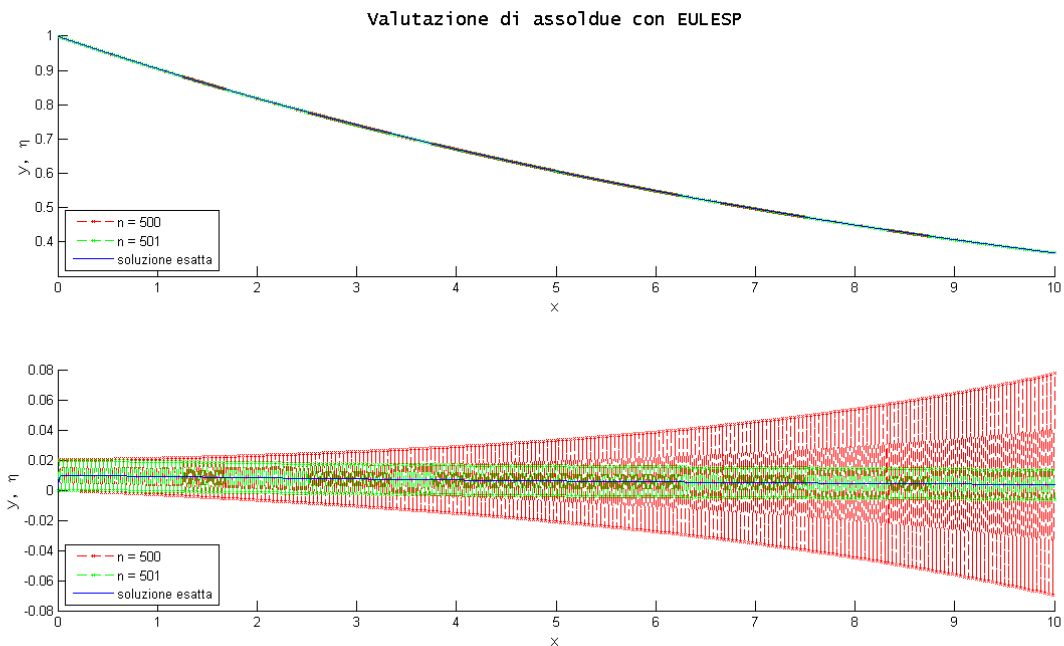
Questi metodi danno delle soluzioni molto prossime alla funzione analitica. e, come per Heun, non si hanno problemi né di smorzamento della soluzione, né di frequenza errata. In particolare il metodo di Simpson risulta essere quello più preciso in assoluto



3. ANALISI DEL COMPORTAMENTO DI EULESP CON EQUAZIONI RIGIDE

Questa analisi è orientata a evidenziare il problema che presentano i metodi espliciti nella risoluzione di equazioni rigide (stiff). La prova è stata fatta sull'equazione:

$$\begin{cases} y_1' = -100 \cdot y_1 + y_2 \\ y_2' = -\frac{1}{10} \cdot y_2 \\ y_1(0) = 0; y_2(0) = 1; \end{cases}$$



Lo studio da un punto di vista numerico della risoluzione del problema impone un limite superiore per l'ampiezza del passo per arrivare a convergenza, che risulta pari a $\frac{1}{50}$. Ciò vuol dire che il numero di punti minimi in quest'intervallo $[0; 10]$ deve essere maggiore di 500.

Infatti la figura soprastante mostra che per $n = 500$ il metodo non converge, mentre già per $n = 501$ la soluzione numerica si avvicina di molto a quella analitica. Chiaramente per una migliore risoluzione conviene aumentare ancora il numero di punti.

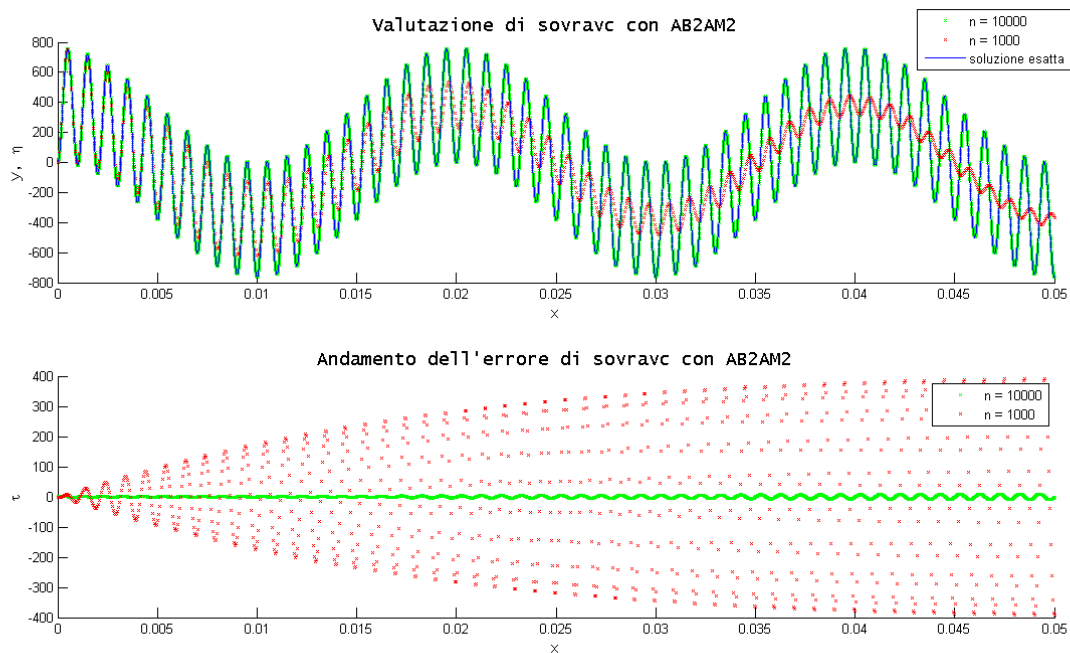
4. ANALISI DEL METODO PREDICTOR CORRECTOR DI ORDINE 2 BASATO SU ADAMS BASHFORT MOULTON

Il metodo di Adams Bashfort Moulton (**ab2am2**) è un Predictor-Corrector che accoppia un metodo esplicito (predictor) in questo caso del second'ordine e uno implicito (corrector) sempre del second'ordine.

- Predictor - Corrector

$$\begin{cases} \eta_{i+1}^* = \eta_i + \frac{h}{2} \cdot [3 \cdot f(x_i, \eta_i) - f(x_{i-1}, \eta_{i-1})] \\ \eta_{i+1} = \eta_i + \frac{h}{2} \cdot [f(x_{i+1}, \eta_{i+1}^*) + f(x_i, \eta_i)] \end{cases}$$

L'equazione utilizzata per l'analisi è sovravc. I risultati ottenuti sono riassunti nel grafico sottostante:



Anche con questo metodo la soluzione risulta essere smorzata e aumentando il numero di punti le condizioni migliorano, il tempo di risoluzione è dello stesso ordine di grandezza degli altri metodi analizzati, a parità di numero di passi.