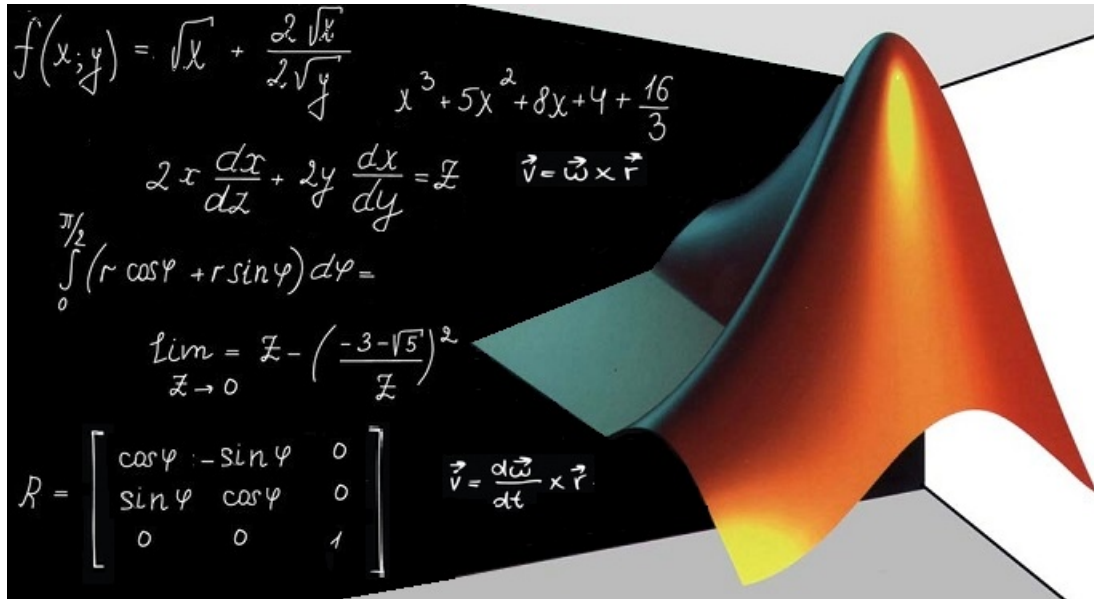


Fabio Nelli (meccanismocomplesso)

CALCOLO SIMBOLICO CON MATLAB (PARTE 1)

22 March 2014



Matlab è un'applicazione che tutti ben conosciamo, ma non sempre si ha consapevolezza di tutte le sue potenzialità. Infatti Matlab oltre che effettuare il calcolo numerico diretto a cui tutti siamo abituati, ci permette anche di valutare analiticamente (cioè mantenendo le espressioni parametriche) molti di questi calcoli. Infatti, grazie al **Symbolic Math Toolbox**, Matlab ci fornisce un set di istruzioni per il calcolo simbolico (o letterale). Quindi, piuttosto che effettuare i calcoli utilizzando dei numeri noti, come siamo soliti fare con Matlab, si utilizzeranno dei caratteri (lettere alfabetiche o lettere greche) in modo da formare delle espressioni simboliche. Tutto ciò è molto utile quando non si vuole conoscere immediatamente il risultato di un calcolo, ma mantenere la formula matematica durante tutta una serie di calcoli. Solo al termine, assegneremo dei valori numerici a questi parametri per conoscere i nostri risultati. Lavorando in questo modo sarà possibile impostare diversi set di valori ai vari parametri, analizzarne il risultato, eventualmente modificarli, e tutto ciò senza dover più effettuare alcun calcolo. Riassumendo, con Matlab è possibile gestire le operazioni matematiche analiticamente. Ecco un elenco delle possibili operazioni che Matlab ci mette a disposizione:

- derivate e integrali
- risoluzione delle equazioni
- manipolazione e semplificazione delle formule matematiche

- serie numeriche e sommatorie (Taylor,..)
- limiti
- trasformate (Fourier, Laplace, ecc.)
- calcolo con le matrici
- analisi vettoriale (jacobiano, laplaciano,..)

Definizione delle espressioni simboliche: **sym** e **syms**

Il calcolo simbolico si basa sull'uso delle variabili della classe **sym** (definite attraverso il Symbolic Toolbox). Quindi se hai bisogno di definire un'espressione simbolica con un determinato insieme di variabili, la prima cosa che dovrai fare sarà quella di dichiarare queste variabili come simboli. Per fare ciò si usano i comandi **sym** e **syms**. Consideriamo il caso in cui vogliamo calcolarci il seno di un angolo θ , con $\theta = \pi/2$.

```
>> theta = pi/2;
ans =
1
```

Giustamente, otterrai un valore numerico. Ma qualche volta, avrai bisogno di mantenere l'espressione in forma simbolica (analitica). Quindi per prima cosa dovrai dichiarare l'angolo θ come simbolo utilizzando la funzione **sym()**.

```
>> x = sym('theta')
x =
theta
>> sin(x)
ans =
sin(theta)
```

Questa volta l'espressione mantiene l'angolo θ come parametro durante il calcolo. Generalmente, può essere necessario utilizzare più di una variabile in forma simbolica, per esempio, se vogliamo specificare la funzione seguente:

$$y = \sin^2(\alpha) + \cos(\beta) + \sin(\gamma)$$

dovrai utilizzare il comando **syms** seguito da tre parametri e poi (opzionalmente) specificare il tipo dei dati (per esempio di tipo real)

```
>> syms alpha beta gamma real
>> y = sin(alpha)^2 + cos(beta) + sin(gamma)
ans =
y = sin(alpha)^2 + cos(beta) + sin(gamma)
```

Quando il comando **syms** viene usato senza alcun argomento, tutti i valori simbolici

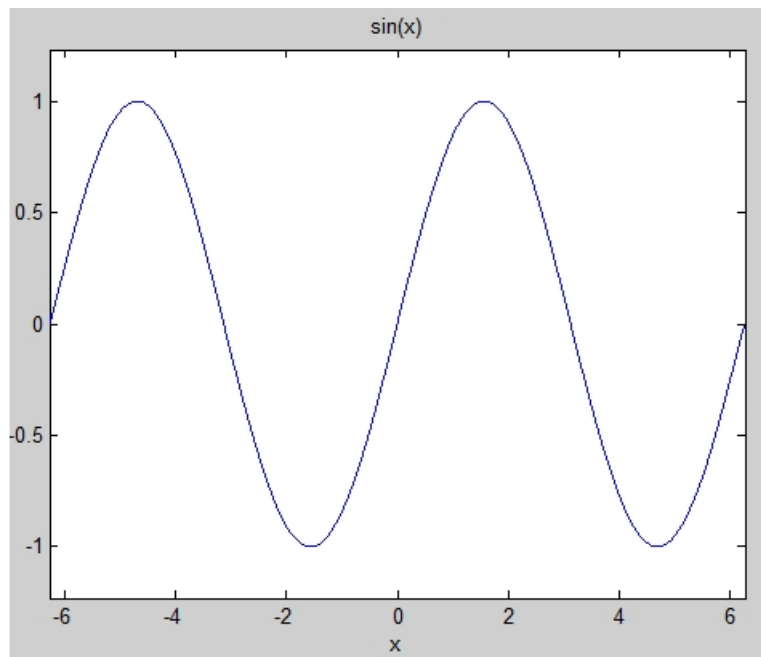
presenti nel workspace di Matlab verranno elencati. I comandi **sym** e **syms** sono parole riservate in Matlab . Personalmente, per semplicità e per chiarezza nella sintassi, preferisco usare sempre il comando **syms** anche nel caso di un singolo parametro.

```
>>> syms theta
>> sin(theta)
ans =
sin(theta)
```

Rappresentare graficamente le funzioni simboliche

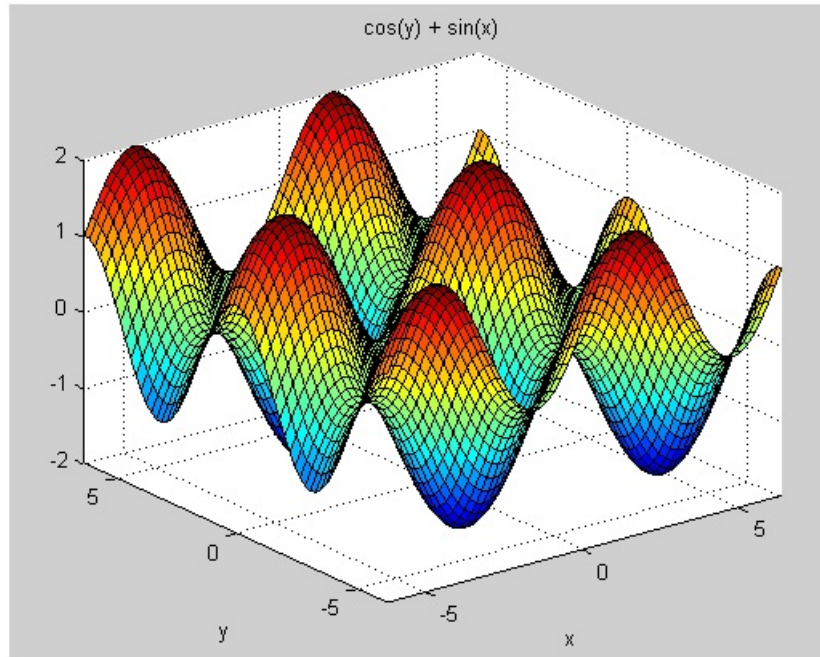
E' possibile rappresentare graficamente l'andamento di una funzione simbolica rispetto ad una variabile utilizzando la funzione **ezplot()**. Ecco un esempio:

```
>> syms x real
>> y = sin(x);
>> ezplot(y);
```



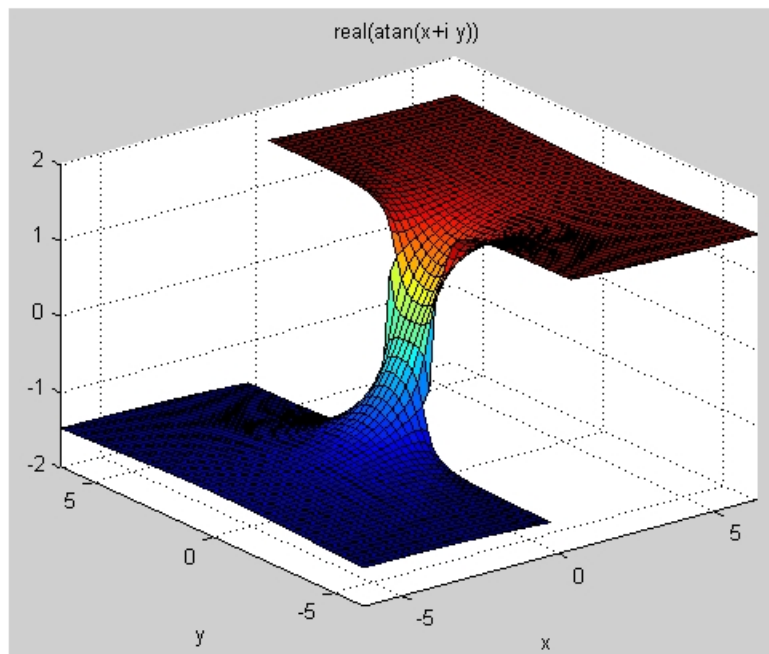
Se invece vuoi vedere qualcosa di più accattivante, prova la funzione **ezsurf()**.

```
>> syms x y real
>> f = sin(x);
>> g = cos(y);
>> ezsurf(f+g);
```



o anche questo...

```
>> ezsurf('real(atan(x+i*y))');
```



Risoluzione di equazioni

Un altro caso utile è la possibilità di trovare le radici di un polinomio. Consideriamo la seguente equazione simbolica:

$$ax^2 - 2bx + c = 0$$

Si possono ricavare le radici, cioè i valori x che risolvono l'equazione, usando la funzione **solve()**.

```
>> syms x a b x
>> y = a*x^2-2*b*x+c;
>> solve(y)
ans =
(b + (b^2 - a*c)^(1/2))/a
(b - (b^2 - a*c)^(1/2))/a
```

Manipolazione e Semplificazione delle espressioni simboliche

Spesso, quando si lavora con le espressioni simboliche (analitiche), e particolarmente con i polinomi, si ha bisogno di espanderle o semplificarle. La funzione **expand()** espande una formula dove sia possibile. Per esempio si può riscrivere il prodotto di somme come somme di prodotti.

```
>> syms x a b;
>> f = (x+a)*(x+b)
>> expand(f)
ans =
a*b + a*x + b*x + x^2
```

L'operazione opposta alla espansione è la semplificazione e quest'ultima si effettua con la funzione **simplify()**. Per vedere come sia possibile manipolare le espressioni polinomiali, consideriamo come esempio il cubo di un binomio.

$$(x + a)^3$$

Per prima cosa, il binomio può essere espanso nel modo seguente:

$$(x + a)^3 = x^3 + 3ax^2 + 3a^2x + a^3$$

e se poi si effettua una semplificazione sulla espansione precedente, si riotterrà nuovamente il cubo di un binomio.

$$x^3 + 3ax^2 + 3a^2x + a^3 = (x + a)^3$$

```
>> syms x a;
>> f = (x+a)^3
>> g = expand(f)
```

```
g = a^3 + 3*a^2*x + 3*a*x^2 + x^3
>> simplify(g)
(a + x)^3
```

Derivate e integrali

Matlab è in grado di calcolare anche molti integrali e derivate che si possono trovare in un corso di Calcolo numerico o in altri corsi avanzati di ingegneria. Le funzioni chiave sono **int()** per l'integrazione e **diff()** per le derivate.

Derivate

Consideriamo la funzione seguente:

$$f(x) = x^3 - \cos(x)$$

Supponiamo di essere interessati ad ottenere l'espressione analitica (simbolica) della sua derivata. Per prima cosa sarà necessario specificare la variabile x come simbolo e successivamente l'espressione della funzione che vogliamo derivare. Infine si dovrà passare la funzione appena definita come argomento di **diff()**. In questa maniera, Matlab restituisce come risultato l'espressione analitica a cui siamo interessati.

```
>> syms x>> f = x^3 - cos(x);>> g = diff(f)      g = 3*x^2 + sin(x)
```

Se abbiamo invece bisogno di calcolare la derivata seconda (o maggiore) della funzione, sarà necessario specificare un secondo argomento nella funzione **diff()**: **diff(x,n)**, dove n è l'ordine della derivata. Per esempio, se vogliamo conoscere la derivata seconda della funzione simbolica f definita in precedenza, dobbiamo immettere le seguenti righe:

```
>> g = diff(f,2)      g = 6*x + cos(x)
```

Derivata parziale

Finora abbiamo usato il comando **diff** con una funzione ad sola una variabile indipendente x . Quindi il comando **diff** effettua la derivata rispetto alla variabile x . Se invece abbiamo a che fare con funzioni con più variabili indipendenti, sarà necessario specificare la variabile di derivazione come secondo parametro.

```
diff(f,y)
```

dove y è la variabile di derivazione. Notate che in questo caso il comando **diff(f,y)** è equivalente alla formula delle derivate parziali:

$$\frac{\partial f(x, y)}{\partial y}$$

Per esempio, se desideriamo ottenere la derivata parziale della funzione seguente calcolata rispetto alla variabile y :

$$f(x, y) = 2x^2 + (y + 7)^3$$

```
>> syms x y>> f = 2*x^2 + (y+7)^3;>> diff(f,y)ans=      3*(y + 7)^2
```

Integrali

Allo stesso modo, se abbiamo bisogno di ottenere l'espressione analitica (simbolica) dell'integrale indefinito, utilizzeremo la funzione **int()** in un maniera simile a come abbiamo fatto con la funzione **diff()**. Consideriamo la stessa funzione che abbiamo utilizzato per il calcolo delle derivate.

```
>> syms x y
>> f = 2*x^2 + (y+7)^3;
```

Anche qui, se la funzione presenta più di una variabile indipendente, sarà necessario specificare la variabile di integrazione come secondo argomento, che nel nostro caso sarà la variabile x . Eseguendo l'integrazione Matlab restituisce il seguente risultato:

```
>> int(f,x)ans =      1/3*x^3+(y+5)^3*x
```

Come possiamo vedere, abbiamo ottenuto ancora un'espressione simbolica come risultato. Invece, se abbiamo bisogno di calcolare un integrale definito tra due valori (per esempio tra 0 e 10).

$$\int_0^{10} f(x, y) dy$$

si dovranno specificare altri due argomenti all'interno della funzione **int()**.

```
>> int(f,y,0,10)ans =      12500+10*x^2
```

In questo caso il parametro simbolico y è stato sostituito con dei valori numerici.

Fine prima parte.

Nella [seconda parte](#) vedremo il calcolo simbolico per:

- serie numeriche e sommatorie (Taylor,..)
- limiti
- trasformate (Fourier, Laplace, ecc.)

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Meccanismocomplesso:calcolo-simbolico-con-matlab-2>"