

TESINA DI CALCOLO NUMERICO 1

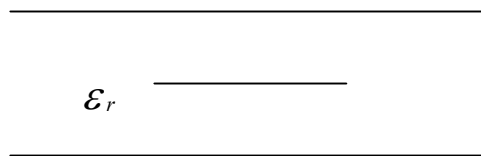
CALCOLO DELLA DISTRIBUZIONE DI POTENZIALE IN UNA STRIPLINE

INTRODUZIONE

In questa tesina sono stati utilizzati diversi algoritmi per ricavare la distribuzione del potenziale elettromagnetico all'interno di una stripline risolvendo l'equazione di Laplace:

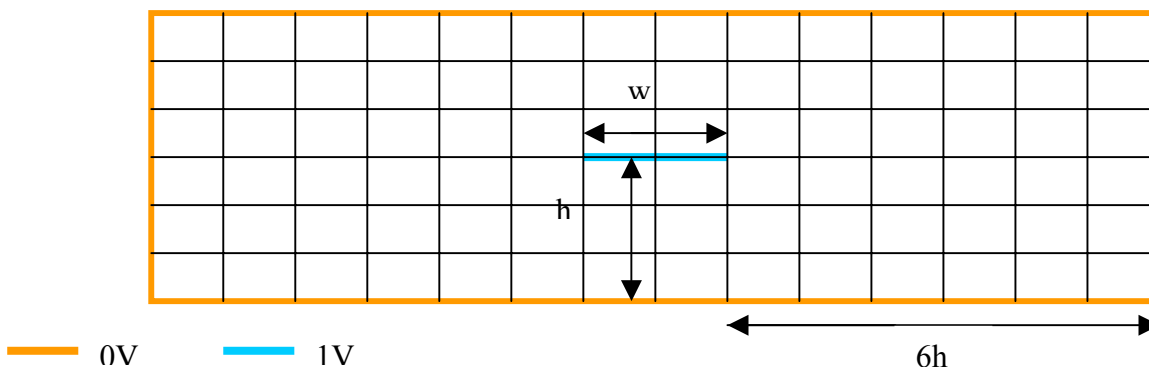
$$\nabla^2 \phi = 0$$

La stripline è una struttura costituita da due conduttori di massa esterni il cui spazio compreso è riempito da dielettrico omogeneo, con al centro della struttura una metallizzazione.



Per risolvere numericamente questo problema bisogna applicare una discretizzazione quindi scegliere un determinato passo reticolare. La discretizzazione permette di ottenere un numero discreto di punti sui quali risolvere l'equazione di Laplace.

Poiché consideriamo la microstrip infinitamente estesa lateralmente risulta ancora presente il problema di dover valutare il potenziale su un numero infinito di punti (anche se discreti), per ottenere un numero finito di punti consideriamo le proprietà elettrostatiche del problema. Le proprietà sono che se poniamo il potenziale delle metallizzazioni esterne a 0 e quello della metallizzazione centrale a 1 il potenziale interno alla stripline sarà compreso tra 0 e 1, inoltre il potenziale tenderà a 0 più ci allontaniamo dalla metallizzazione centrale.



Quindi si ricava sperimentalmente che se poniamo a zero il potenziale sui bordi laterali della stripline, posti ad una distanza di sei volte la spaziatura tra la metallizzazione centrale e quella di massa, il risultati sono accettabili.

Da notare che le misure in figura non sono in scala.

Quindi una volta discretizzato il laplaciano è stato ottenuto un sistema lineare che è stato risolto con l'algoritmo di Gauss (implementato da Matlab) e con tre differenti implementazioni

dell'algoritmo iterativo di Gauss-Siedel mettendo in evidenza le varie differenze in prestazioni ed occupazione di memoria.

DISCRETIZZAZIONE DEL LAPLACIANO

Il valore della derivata in un punto lo possiamo approssimare mediante il rapporto incrementale in avanti:

$$\frac{df(x_i)}{dx} = \frac{f(x_{i+1}) - f(x_i)}{h}$$

dove h è il passo di discretizzazione. Quindi se calcoliamo la derivata seconda di $f(x_i)$, sempre secondo la stessa approssimazione, otteniamo:

$$\frac{d^2 f(x_i)}{dx^2} = \frac{\frac{df(x_i)}{dx} - \frac{df(x_{i-1})}{dx}}{h} = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{h^2}$$

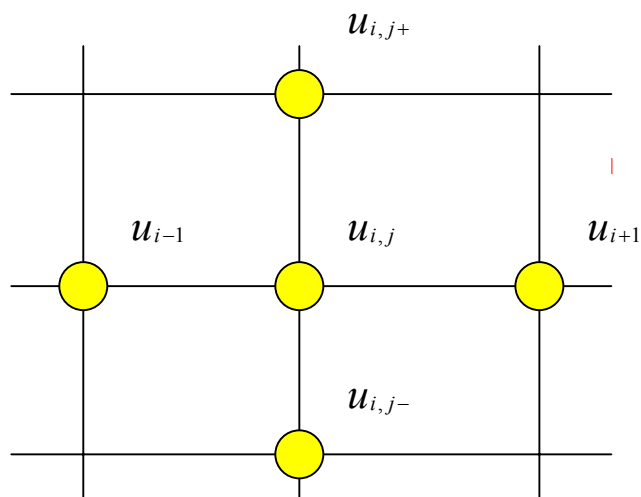
Tutto ciò che abbiamo visto vale in una dimensione ma in nostro problema è bidimensionale. Quindi in questo caso per calcolare il gradiente dobbiamo considerare sia la derivata parziale rispetto a x sia quella rispetto a y:

$$\Delta u = \frac{\partial^2}{\partial x^2} u(x_i, y_j) + \frac{\partial^2}{\partial y^2} u(x_i, y_j)$$

Quindi sostituendo le derivate seconde calcolate precedentemente otteniamo:

$$\Delta u(x_i, y_j) = \frac{u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i,j+1} + u_{i+1,j}}{h^2} = 0$$

Quindi possiamo scrivere un'equazione per ogni punto interno alla stripline in funzione dei valori del potenziale sui punti attorno quello di interesse:



Per quanto riguarda il range in cui devono variare i e j si ha che se la dimensione orizzontale della stripline è discretizzata con N+1 punti allora dovremo scartare i punti alle estremità perché in essi abbiamo già imposto che il potenziale sia pari a zero, analogamente per la dimensione verticale:

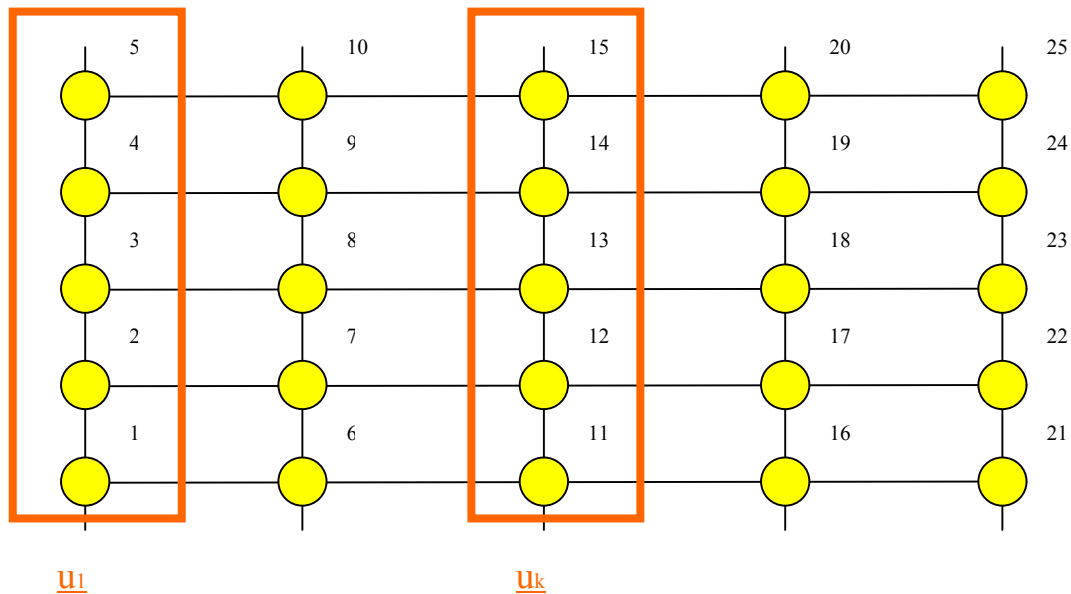
$$i = 1, 2, 3, \dots, N - 1$$

$$j = 1, 2, 3, \dots, M - 1$$

Da ciò segue che abbiamo ottenuto un sistema lineare di dimensioni al massimo pari a (N-1)x(M-1) (diciamo al massimo poiché come vedremo le equazioni relative ai punti della metallizzazione centrale possiamo scartarle perché imponiamo su di essi che il potenziale sia pari a 1).

A questo punto ci serve un modo intelligente per numerare le equazioni in modo tale da ottenere come matrice associata al sistema una strutturata.

Per ottenere questo vengono ordinate le equazioni in ordine **lessicografico** che prevede un ordinamento come in figura:



Quindi il vettore delle incognite del sistema ottenuto possiamo scriverlo nella seguente forma:

$$U = (\underline{u}_{i,j}) = \begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \vdots \\ \vdots \\ \vdots \\ \underline{u}_{M-1} \end{pmatrix} \quad \text{dove} \quad \underline{u}_k = \begin{pmatrix} u_{k,1} \\ u_{k,2} \\ \vdots \\ \vdots \\ \vdots \\ u_{k,N-1} \end{pmatrix} \in R^{N-1}$$

A questo punto notiamo che per ogni i possiamo scrivere il seguente sottosistema:

$$I \underline{u}_{k-1} + A_0 \underline{u}_k + I \underline{u}_{k+1} = 0$$

ovvero:

$$\begin{pmatrix} \mathbf{u}_{k-1,1} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{u}_{k-1,N-1} \end{pmatrix} + \begin{pmatrix} -4 & 1 & 0 & : & : & : \\ 1 & -4 & 1 & : & : & : \\ 0 & 1 & : & : & : & : \\ : & : & : & : & 1 & 0 \\ : & : & : & 1 & -4 & 1 \\ : & : & : & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{k,1} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{u}_{k,N-1} \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{k+1,1} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{u}_{k+1,N-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{u}_{k,0} \\ 0 \\ \vdots \\ \vdots \\ 0 \\ -\mathbf{u}_{k,N} \end{pmatrix}$$

con $k=1,2,\dots,M-1$.

Da notare che per le condizioni al contorno imposte, cioè che il potenziale alla frontiera del nostro problema è pari a zero, $\mathbf{u}_{k,0}$ e $\mathbf{u}_{k,N}$ risulteranno nulli. Inoltre risulteranno diversi i sottosistemi per $k=1$ e $k=M-1$, infatti per essi si avrà rispettivamente:

$$\begin{pmatrix} -4 & 1 & 0 & : & : & : \\ 1 & -4 & 1 & : & : & : \\ 0 & 1 & : & : & : & : \\ : & : & : & : & 1 & 0 \\ : & : & : & 1 & -4 & 1 \\ : & : & : & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{1,1} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{u}_{1,N-1} \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{2,1} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{u}_{2,N-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{u}_{0,1}-\mathbf{u}_{1,0} \\ -\mathbf{u}_{0,2} \\ \vdots \\ \vdots \\ -\mathbf{u}_{0,N-2} \\ -\mathbf{u}_{0,N-1}-\mathbf{u}_{1,N} \end{pmatrix}$$

$$\begin{pmatrix} \mathbf{u}_{M-2,1} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{u}_{M-2,N-1} \end{pmatrix} + \begin{pmatrix} -4 & 1 & 0 & : & : & : \\ 1 & -4 & 1 & : & : & : \\ 0 & 1 & : & : & : & : \\ : & : & : & : & 1 & 0 \\ : & : & : & 1 & -4 & 1 \\ : & : & : & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{M-1,1} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{u}_{M-1,N-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{u}_{M,1}-\mathbf{u}_{M-1,0} \\ -\mathbf{u}_{M,2} \\ \vdots \\ \vdots \\ -\mathbf{u}_{M,N-2} \\ -\mathbf{u}_{M,N-1}-\mathbf{u}_{M-1,N} \end{pmatrix}$$

Quindi il sistema che dovremo risolvere avrà la seguente forma:

$$AU = F$$

dove la matrice A è una matrice quadrata di dimensione $(N-1)(M-1)$, U ed F sono dei vettori di dimensione $(N-1)(M-1)$.

Inoltre tale sistema assumerà la seguente forma:

$$\begin{pmatrix} A_0 & I & 0 & : & : & : \\ I & A_0 & I & : & : & : \\ 0 & I & : & : & : & : \\ : & : & : & : & I & 0 \\ : & : & : & I & A_0 & I \\ : & : & : & 0 & I & A_0 \end{pmatrix} \begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \\ : \\ : \\ : \\ \underline{u}_{M-1} \end{pmatrix} = F$$

A questo punto possiamo notare che in effetti a causa del fatto che le equazioni sui punti della metallizzazione centrale sono imposte pari a 1 il nostro sottosistema assumerà una forma diversa dipendentemente dal valore di k.

Si possono verificare sei possibili situazioni:

1)k=1

$$\begin{pmatrix} -4 & 1 & 0 & : & : & : \\ 1 & -4 & 1 & : & : & : \\ 0 & 1 & : & : & : & : \\ : & : & : & : & 1 & 0 \\ : & : & : & 1 & -4 & 1 \\ : & : & : & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} \underline{u}_{1,1} \\ : \\ : \\ : \\ : \\ \underline{u}_{1,N-1} \end{pmatrix} + \begin{pmatrix} \underline{u}_{2,1} \\ : \\ : \\ : \\ : \\ \underline{u}_{2,N-1} \end{pmatrix} = \begin{pmatrix} -\underline{u}_{0,1}-\underline{u}_{1,0} \\ -\underline{u}_{0,2} \\ : \\ : \\ -\underline{u}_{0,N-2} \\ -\underline{u}_{0,N-1}-\underline{u}_{1,N} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ : \\ : \\ 0 \\ 0 \end{pmatrix}$$

2)k=M-1

$$\begin{pmatrix} \underline{u}_{M-2,1} \\ : \\ : \\ : \\ : \\ \underline{u}_{M-2,N-1} \end{pmatrix} + \begin{pmatrix} -4 & 1 & 0 & : & : & : \\ 1 & -4 & 1 & : & : & : \\ 0 & 1 & : & : & : & : \\ : & : & : & : & 1 & 0 \\ : & : & : & 1 & -4 & 1 \\ : & : & : & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} \underline{u}_{M-1,1} \\ : \\ : \\ : \\ : \\ \underline{u}_{M-1,N-1} \end{pmatrix} = \begin{pmatrix} -\underline{u}_{M,1}-\underline{u}_{M-1,0} \\ -\underline{u}_{M,2} \\ : \\ : \\ -\underline{u}_{M,N-2} \\ -\underline{u}_{M,N-1}-\underline{u}_{M-1,N} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ : \\ : \\ 0 \\ 0 \end{pmatrix}$$

3)k+1 non coincide ancora alla colonna in cui vi sono punti della metallizzazione quindi il sottosistema avrà la forma vista precedentemente, e se teniamo conto delle condizioni al contorno avremo:

$$\begin{pmatrix} \mathbf{u}_{k-1,1} \\ \vdots \\ \mathbf{u}_{k-1,N/2} \\ \vdots \\ \mathbf{u}_{k-1,N-1} \end{pmatrix} + \begin{pmatrix} -4 & 1 & 0 & \vdots & \vdots & \vdots & \vdots \\ 1 & -4 & 1 & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & 1 & -4 & 1 \\ \vdots & \vdots & \vdots & \vdots & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{k,1} \\ \vdots \\ \mathbf{u}_{k,N/2} \\ \vdots \\ \mathbf{u}_{k,N-1} \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{k+1,1} \\ \vdots \\ \mathbf{u}_{k+1,N/2} \\ \vdots \\ \mathbf{u}_{k+1,N-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{u}_{k,0} \\ \vdots \\ 0 \\ \vdots \\ -\mathbf{u}_{k,N} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

4) Solo k+1 coincide con una colonna in cui vi sono punti della metallizzazione:

$$\begin{pmatrix} \mathbf{u}_{k-1,1} \\ \vdots \\ \mathbf{u}_{k-1,N/2} \\ \vdots \\ \mathbf{u}_{k-1,N-1} \end{pmatrix} + A_0 \begin{pmatrix} \mathbf{u}_{k,1} \\ \vdots \\ \mathbf{u}_{k,N/2} \\ \vdots \\ \mathbf{u}_{k,N-1} \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{k+1,1} \\ \vdots \\ \mathbf{u}_{k+1,N/2-1} \\ 0 \\ \mathbf{u}_{k+1,N/2+1} \\ \vdots \\ \mathbf{u}_{k+1,N-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{u}_{k,0} \\ 0 \\ \vdots \\ 0 \\ -\mathbf{u}_{k-1,N/2} \\ 0 \\ \vdots \\ 0 \\ -\mathbf{u}_{k,N} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

5) k e k+1 coincidono con una colonna in cui vi sono punti della metallizzazione:

$$\begin{pmatrix} \mathbf{u}_{k-1,1} \\ \vdots \\ \mathbf{u}_{k-1,N/2} \\ \vdots \\ \mathbf{u}_{k-1,N-1} \end{pmatrix} + A_1 \begin{pmatrix} \mathbf{u}_{k,1} \\ \vdots \\ \mathbf{u}_{k,N/2} \\ \vdots \\ \mathbf{u}_{k,N-1} \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{k+1,1} \\ \vdots \\ \mathbf{u}_{k+1,N/2-1} \\ 0 \\ \mathbf{u}_{k+1,N/2+1} \\ \vdots \\ \mathbf{u}_{k+1,N-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{u}_{k,0} \\ 0 \\ \vdots \\ 0 \\ -\mathbf{u}_{k,N/2-1} \\ 4\mathbf{u}_{k,N/2} - \mathbf{u}_{k+1,N/2} \\ -\mathbf{u}_{k,N/2+1} \\ 0 \\ \vdots \\ 0 \\ -\mathbf{u}_{k,N} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 3 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

6) $k-1$, k e $k+1$ coincidono con una colonna in cui vi sono punti della metallizzazione:

$$\begin{pmatrix} \mathbf{u}_{k-1,1} \\ \vdots \\ \mathbf{u}_{k-1,N/2-1} \\ \mathbf{0} \\ \mathbf{u}_{k-1,N/2+1} \\ \vdots \\ \mathbf{u}_{k-1,N-1} \end{pmatrix} + A_1 \begin{pmatrix} \mathbf{u}_{k,1} \\ \vdots \\ \mathbf{u}_{k,N/2} \\ \vdots \\ \mathbf{u}_{k,N-1} \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{k+1,1} \\ \vdots \\ \mathbf{u}_{k+1,N/2-1} \\ \mathbf{0} \\ \mathbf{u}_{k+1,N/2+1} \\ \vdots \\ \mathbf{u}_{k+1,N-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{u}_{k,0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_{k,N/2-1} \\ 4\mathbf{u}_{k,N/2} - \mathbf{u}_{k-1,N/2} - \mathbf{u}_{k+1,N/2} \\ -\mathbf{u}_{k,N/2+1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_{k,N} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -1 \\ 2 \\ -1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$$

7) $k-1$ e k coincidono con una colonna in cui vi sono punti della metallizzazione:

$$\begin{pmatrix} \mathbf{u}_{k-1,1} \\ \vdots \\ \mathbf{u}_{k-1,N/2-1} \\ \mathbf{0} \\ \mathbf{u}_{k-1,N/2+1} \\ \vdots \\ \mathbf{u}_{k-1,N-1} \end{pmatrix} + A_1 \begin{pmatrix} \mathbf{u}_{k,1} \\ \vdots \\ \mathbf{u}_{k,N/2} \\ \vdots \\ \mathbf{u}_{k,N-1} \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{k+1,1} \\ \vdots \\ \mathbf{u}_{k+1,N/2} \\ \vdots \\ \mathbf{u}_{k+1,N-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{u}_{k,0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_{k,N/2-1} \\ 4\mathbf{u}_{k,N/2} - \mathbf{u}_{k-1,N/2} \\ -\mathbf{u}_{k,N/2+1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_{k,N} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -1 \\ 3 \\ -1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$$

8) Solo $k-1$ coincide con una colonna in cui vi sono punti della metallizzazione:

$$\begin{pmatrix} \mathbf{u}_{k-1,1} \\ \vdots \\ \mathbf{u}_{k-1,N/2-1} \\ \mathbf{0} \\ \mathbf{u}_{k-1,N/2+1} \\ \vdots \\ \mathbf{u}_{k-1,N-1} \end{pmatrix} + A_0 \begin{pmatrix} \mathbf{u}_{k,1} \\ \vdots \\ \mathbf{u}_{k,N/2} \\ \vdots \\ \mathbf{u}_{k,N-1} \end{pmatrix} + \begin{pmatrix} \mathbf{u}_{k+1,1} \\ \vdots \\ \mathbf{u}_{k+1,N/2} \\ \vdots \\ \mathbf{u}_{k+1,N-1} \end{pmatrix} = \begin{pmatrix} -\mathbf{u}_{k,0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_{k-1,N/2} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{u}_{k,N} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ -1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$$

Quindi il sistema avrà la seguente forma:

$$\begin{pmatrix} A_0 & I & : & : & : & : & : & : & : & : & : \\ I & A_0 & : & 0 & : & : & : & : & : & : & : \\ 0 & I & : & I & 0 & : & : & : & : & : & : \\ : & 0 & : & A_0 & B & : & : & : & : & : & : \\ : & : & : & I & A_1 & : & 0 & : & : & : & : \\ : & : & : & 0 & B & : & B & 0 & : & : & : \\ : & : & : & : & 0 & : & A_1 & I & : & : & : \\ : & : & : & : & : & : & B & A_0 & : & 0 & : \\ : & : & : & : & : & : & 0 & I & : & I & 0 \\ : & : & : & : & : & : & : & 0 & : & A_0 & I \\ : & : & : & : & : & : & : & : & : & I & A_0 \end{pmatrix} \begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \vdots \\ \vdots \\ \vdots \\ \underline{u}_{M-1} \end{pmatrix} = F$$

Dove naturalmente F è il termine noto ottenuto concatenando tutti i termini noti associati ai sottosistemi, B è la matrice identità di ordine N-1 in cui la colonna N/2 ha tutti gli elementi nulli, I è la matrice identità di ordine N-1. La matrice A_1 è la matrice A_0 con tutti gli elementi nulli nella colonna N/2.

Come si nota vi saranno colonne con soli zeri quindi possiamo ridurre il sistema eliminando tali colonne e le righe con indice pari alle colonne eliminate.

Quindi il sistema diventerà:

$$DU = e$$

dove D è la matrice ridotta ed e è il vettore dei termini noti ridotto. Le dimensioni di questi ultimi due dipende dal numero di punti della metallizzazione centrale.

Infatti il numero di equazioni finale sarà:

$$\text{numero equazioni} = (N - 1) \cdot (M - 1) - \text{numero punti metallizzazione}$$

IMPLEMENTAZIONE GRADIENTE

E' stato implementato la funzione gradiente in Matlab nel file gradiente2_1. Tale funzione opera esattamente seguendo i passi visti cioè prima costruendo il sistema completo e poi eliminando le equazioni e quindi le incognite che non ci interessano.

La dichiarazione della funzione è la seguente:

$$[D,e,c,a,b,X,Y,N]=\text{gradiente2_1}(h,w,nx,ny)$$

Dove:

h è la distanza tra la metallizzazione interna e il piano di massa;

w è larghezza della metallizzazione interna;

nx è il numeri di punti per la semi larghezza della metallizzazione interna(escluso il punto centrale).

ny è il numero di punti per la distanza metallizzazione interna piano di massa(escluso il punto sul piano di massa e quello sulla metallizzazione centrale);

D è la matrice ridotta del sistema (tiene conto che il conduttore interno ha un potenziale pari a 1, quelli di massa hanno potenziale pari a zero e viene imposta una limitazione laterale al problema);
e è il termine noto ridotto;

c tiene traccia delle variabili (in effetti termini noti) trascurati nella matrice **D** (cioè la matrice **A** ridotta);

X è il numero totale di punti lungo l'asse x;

Y è il numero totale di punti lungo l'asse y;

N è la dimensione del sistema non ridotto;

a è lo spessore della stripline;

Risulta interessante mettere in evidenza il calcolo dei punti totali lungo le due direzioni.

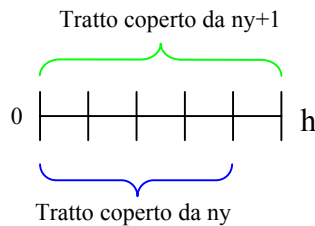
Per quanto riguarda **X** si ha che il numero totale dei punti sarà:

$$X = N_{totmet} + 12 \cdot N_{toty} - 2$$

E' dovuto al fatto che dobbiamo togliere i punti laterali su cui abbiamo imposto il potenziale pari a zero

$2 \cdot nx + 1$
 dove 1 è dovuto al punto centrale che non è compreso in nx

$ny + 1$
 Poiché a noi interessa lo spessore h che si ottiene considerando i punti ny ed aggiungendo un punto altrimenti avremo un valore minore di h



Quindi il valore totale di punti lungo **X** sarà:

$$X = 2nx + 12 \cdot (ny + 1) - 1$$

Invece per quanto riguarda **Y** si ha:

$$Y = 2ny + 1$$

RISOLUZIONE DEL SISTEMA LINEARE CON GAUSS

E' stato sfruttato l'algorithmo implementato nelle librerie di Matlab.

Infatti se in matlab digitiamo:

$$x = A \setminus b$$

se la matrice **A** non ha una struttura particolare utilizza il metodo di Gauss.

Una volta ottenuto il vettore **x** (che è la soluzione ridotta), tramite il vettore **c** fornito dalla funzione `gradiente2_1` (che tiene memorizzate le posizioni delle incognite eliminate dal problema perché imposte in partenza) costruiamo la soluzione totale **U**.

Una volta fatto questo passiamo dal vettore **U** alla matrice rappresentativa della distribuzione di potenziale all'interno della stripline, che è di dimensioni $(Y-1) \times (X-1)$. Come passo finale dobbiamo orlare tale matrice con degli zeri per tenere conto delle condizioni al contorno sulle metallizzazioni interne e sui bordi laterali.

IMPLEMENTAZIONE DELL'ALGORITMO DI GAUSS

Tale Algoritmo è stato implementato con la funzione:

`[x,L,t]=striplineG(h,w,nx,ny)`

Dove:

x è la soluzione finale del sistema ;

L è la soluzione in forma matriciale;

t è il tempo impiegato per risolvere il sistema;

RISOLUZIONE DEL SISTEMA LINEARE CON GAUSS-SIEDEL

Gauss-Siedel è un metodo iterativo lineare, stazionario del prim'ordine che assume la seguente forma:

$$x^{(k+1)} = Bx^{(k+1)} + f$$

dove B e k sono rispettivamente la matrice e l'indice di iterazione.

Possiamo pensare a questo punto la matrice A formata nel seguente modo:

$$A = D - E - F$$

cioè:

$$A = \begin{pmatrix} a_{1,1} & 0 & \cdot & \cdot & 0 \\ 0 & a_{2,2} & 0 & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & a_{n-1,n-1} & 0 \\ 0 & \cdot & \cdot & 0 & a_{n,n} \end{pmatrix} - \begin{pmatrix} 0 & \cdot & \cdot & \cdot & 0 \\ -a_{2,1} & 0 & \cdot & \cdot & \cdot \\ \cdot & -a_{3,2} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & \cdot \\ -a_{n,1} & \cdot & \cdot & -a_{n,n-1} & 0 \end{pmatrix} - \begin{pmatrix} 0 & -a_{1,2} & \cdot & \cdot & -a_{1,n} \\ \cdot & 0 & -a_{2,3} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & -a_{n-1,n} \\ \cdot & \cdot & \cdot & \cdot & 0 \end{pmatrix}$$

Quindi possiamo scrivere il nostro sistema nel seguente modo:

$$Ax = b \Rightarrow (D-E)x^{(k+1)} = b + Fx^{(k)}$$

che in termini di coordinate diventa:

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left[b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k+1)} - \sum_{j=i+1}^n a_{i,j} x_j^{(k+1)} \right] \quad i = 1, \dots, n$$

IMPLEMENTAZIONE DELL'ALGORITMO DI GAUSS SIEDEL

Nel caso di Gauss-Siedel sono state eseguite tre implementazioni una che utilizza cicli for e le altre che utilizzano funzioni già definite da Matlab per ricondurci alla risoluzione di un sistema triangolare.

Le funzioni implementate, rispettivamente, sono:

$[x,L,t]=\text{striplineGS1}(h,w,nx,ny,Nmax,Tau)$
 $[x,L,t]=\text{striplineGS2}(h,w,nx,ny,Nmax,Tau)$
 $[x,L,t,\text{dim1},\text{dim2}]=\text{striplineGS}(h,w,nx,ny,Nmax,Tau)$

Dove:

x è la soluzione finale del sistema ;

L è la soluzione in forma matriciale;

t è il tempo impiegato per risolvere il sistema;

$Nmax$ è il numero massimo di iterazioni;

Tau è la precisione massima richiesta;

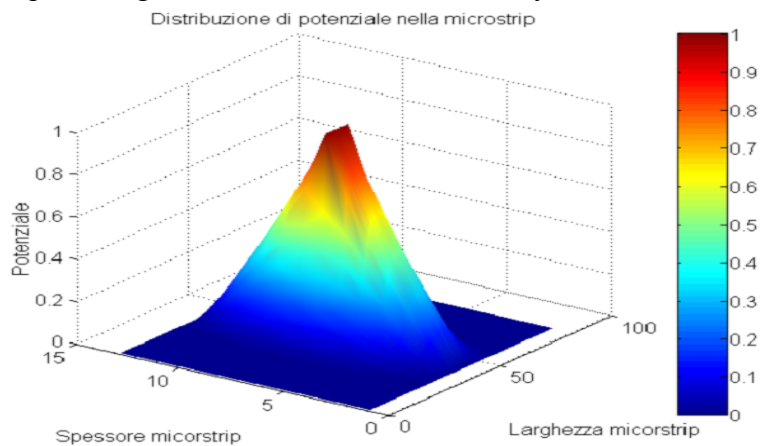
dim1 è la dimensione e l'occupazione in memoria della matrice del sistema prima di essere memorizzata come sparsa;

dim2 è la dimensione e l'occupazione in memoria della matrice del sistema dopo essere stata memorizzata come sparsa;

Una volta risolto il sistema i passi successivi per ottenere la distribuzione totale del potenziale nella stripline sono gli stessi che per Gauss.

CONFRONTO DELLE PRESTAZIONI

Per quanto riguarda Gauss utilizzando $n_x=n_y=5$ si ottiene la seguente soluzione:



Che è esattamente quello che ci aspettiamo poiché il potenziale sarà pari a 1 sulla metallizzazione centrale , pari a zero sui bordi e sarà compreso tra 1 e 0 negli altri punti.

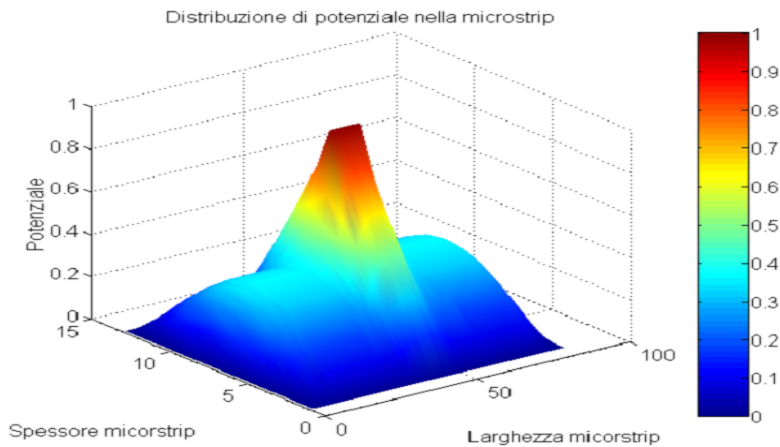
Inoltre si nota come all'allontanarci dalla metallizzazione centrale il potenziale tende a 0.

Il tempo impiegato per risolvere il sistema è:

$t=0.491\text{sec}$

Consideriamo ora l'utilizzo di `striplineGS1` che utilizza dei cicli for per implementare l'algoritmo di Gauss-Siedel.

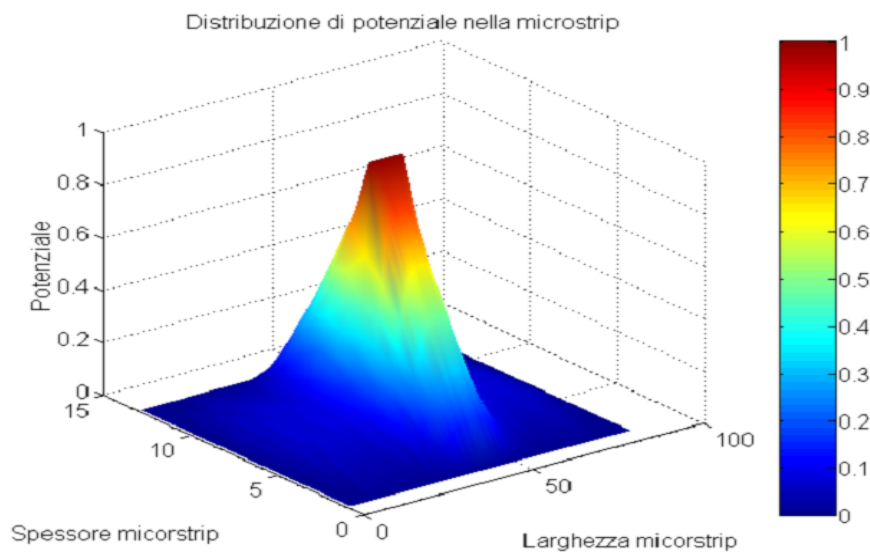
Si nota che se n_x ed n_y uguali a quelli utilizzati precedentemente utilizzando un $Nmax=15$ e $Tau=0.001$ si ottiene:



Come si nota la soluzione è ancora troppo approssimata (basta vedere la soluzione di Gauss). Inoltre si ha che il tempo impiegato per risolvere il sistema è pari a:

$t = 128.5750 \text{sec}$

Proviamo ora ad imporre $N_{\text{max}} = 60$:



Si nota come all'aumentare delle iterazioni la soluzione tenda a convergere con la soluzione di Gauss.

Però si nota anche che l'algoritmo implementato in questo modo è lentissimo infatti in questa seconda prova ha impiegato per risolvere il sistema (che ha dimensione minore, per via della riduzione, di 891) un tempo pari a:

$t = 487.9520 \text{sec}$

Questo è dovuto al fatto che l'algoritmo è stato implementato con cicli for che rallentano enormemente Matlab in quanto è un linguaggio interpretato.

Vediamo ora, utilizzando striplineGS2, come si velocizza l'esecuzione dell'algoritmo utilizzando delle funzioni definite nelle librerie di Matlab che ci permettono di evitare i cicli for. Mantenendo tutti i parametri uguali all'ultima prova otteniamo che il tempo per eseguire 60 iterazioni è:

t = 8.0510sec

Ora, utilizzando striplineGS3, vogliamo mostrare come, dichiarando la matrice del sistema sparsa, si risparmi sia in termini di spazio in memoria sia in termini di tempo di esecuzione. La funzione striplineGS3 fornirà in uscita anche lo spazio occupato in memoria prima e dopo la memorizzazione della matrice del sistema come sparsa. Mantenendo i parametri delle ultime due prove otteniamo:

t = 0.2510sec

```
dim1 = name: 'D'  
      size: [880 880]  
      bytes: 6195200  
      class: 'double'
```

```
dim2 = name: 'D'  
      size: [880 880]  
      bytes: 53828  
      class: 'sparse'
```

Come si nota il tempo di calcolo si riduce notevolmente (da 8.0510sec a 0.2510sec) come d'altronde lo spazio di memorizzazione che passa da 6195200 a 53828 bytes.

ULTERIORI PROVE

Sono state svolte ulteriori prove con i tre algoritmi i cui risultati (in termini di tempo per risolvere il sistema e memoria occupata) sono riportati nella seguente tabella.

Da notare che è stato imposto negli algoritmi iterativi Nmax=100 e Tau=0.001.

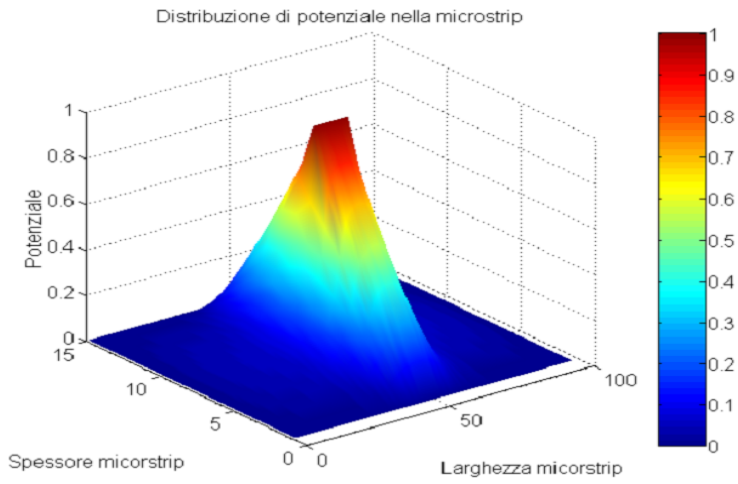
	Dimensioni del sistema	Tempo di risoluzione (sec)				Occupazione memoria (Kbyte)			
		Gauss	GS1	GS2	GS3	Gauss	GS1	GS2	GS3
nx=6 ny=6	1235	1.412	1579	25.797	0.351	11946.3	11946.3	11946.3	75.284
nx=8 ny=8	2091	3.925	n.e.	72.975	0.591	34411.8	34411.8	34411.8	128.95
nx=10 ny=8	2159	4.036	n.e.	78.443	0.741	36568.3	36568.3	36568.3	132.85
nx=10 ny=10	3171	33.738	n.e.	>>3000	1.282	79380	79380	79380	196.95
nx=12 ny=12	4475	288.67	n.e.	n.e.	5.228	158420	158420	158420	279.28

Dove n.e. sta per non eseguito poiché tali esecuzioni richiedevano troppo tempo.

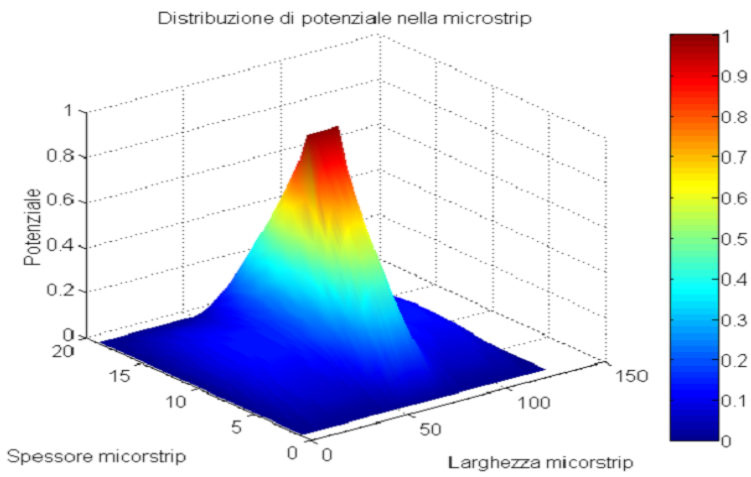
OSSERVAZIONI

Il fatto più importante da mettere in evidenza nelle prove precedenti è che l'algoritmo di Gauss-Siedel è stato eseguito con Nmax fissato a 100 però come mostrano le seguenti figure la soluzione risulta sempre peggiore (rispetto a Gauss) all'aumentare delle dimensioni:

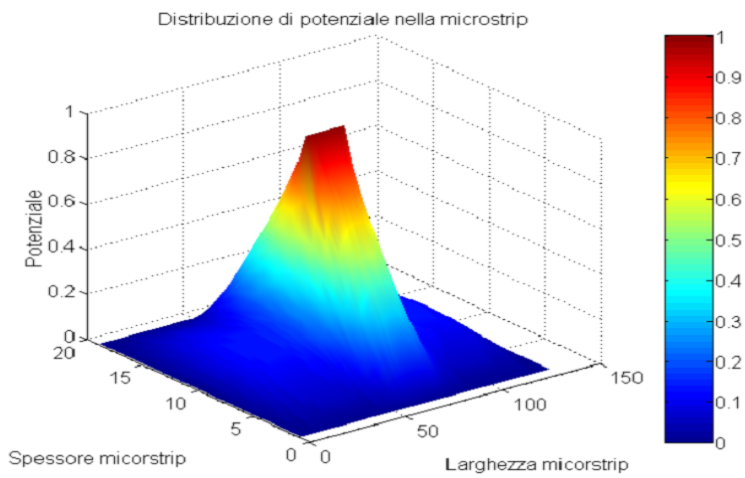
1) nx=6 ny=6



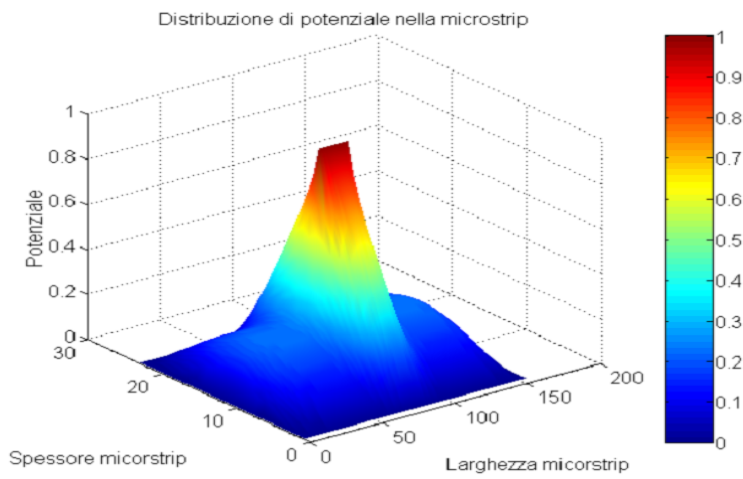
2) $n_x=8$ $n_y=8$



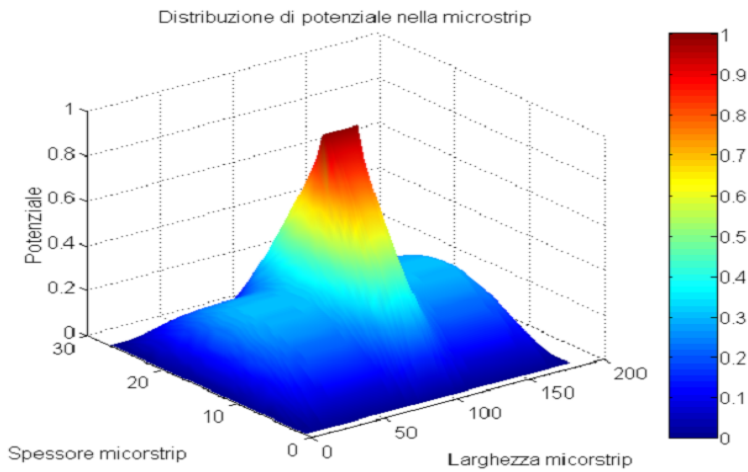
3) $n_x=10$ $n_y=8$



4) $n_x=10$ $n_y=10$

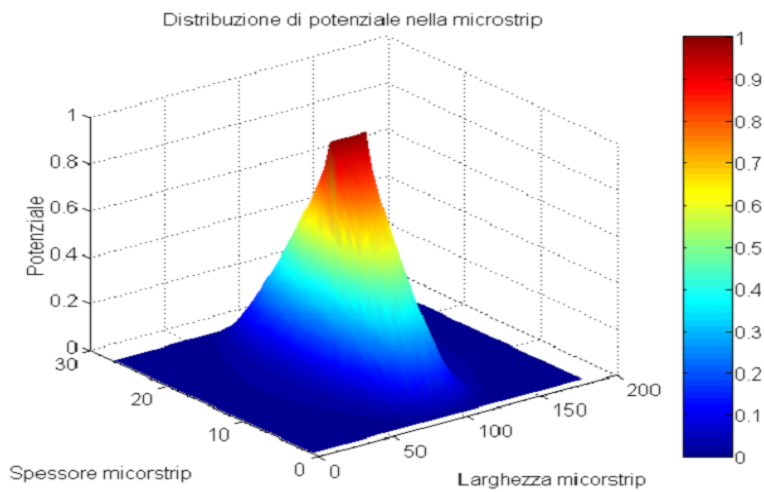


5) $n_x=12$ $n_y=12$



Quindi per ottenere una soluzione accettabile bisogna aumentare notevolmente il numero di iterazioni massime.

Per esempio con $n_x=12$, $n_y=12$ ed $N_{max}=500$ si ottiene:



Però il tempo impiegato per risolvere il sistema è aumentato:

$$t = 13.76\text{sec}$$

Nonostante l'elevato numero di iterazioni da eseguire tale implementazione è molto più veloce di Gauss, oltre naturalmente al grande risparmio di memoria.