

# **Applicazioni di metodi di estrapolazione vettoriale a problemi lineari e non lineari**

**Maria Rosaria Russo, Roberto Bertelle**

**Università degli Studi di Padova  
Dipartimento di Matematica Pura ed Applicata**



# Outline

- ▶ Motivazioni
- ▶ Teoria e metodi
- ▶ Applicazioni a sistemi lineari
- ▶ Applicazioni a problemi non lineari
- ▶ Risultati
- ▶ Conclusioni

# Motivazioni

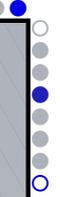
- ▶ Diversi metodi numerici generano successioni di vettori convergenti alla soluzione di un dato problema;

# Motivazioni



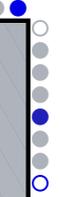
- ▶ Diversi metodi numerici generano successioni di vettori convergenti alla soluzione di un dato problema;
  - ▶ risoluzione di sistemi di grandi dimensioni provenienti dalla discretizzazione di EDP.

# Motivazioni



- ▶ Diversi metodi numerici generano successioni di vettori convergenti alla soluzione di un dato problema;
  - ▶ risoluzione di sistemi di grandi dimensioni provenienti dalla discretizzazione di EDP.
- ▶ Spesso il processo di convergenza è **lento**.

# Motivazioni



- ▶ Diversi metodi numerici generano successioni di vettori convergenti alla soluzione di un dato problema;
  - ▶ risoluzione di sistemi di grandi dimensioni provenienti dalla discretizzazione di EDP.
- ▶ Spesso il processo di convergenza è **lento**.
- ▶ In tutti i casi in cui è **rilevante** il tempo di calcolo si può ricorrere a tecniche di accelerazione della convergenza;

# Motivazioni



- ▶ Diversi metodi numerici generano successioni di vettori convergenti alla soluzione di un dato problema;
  - ▶ risoluzione di sistemi di grandi dimensioni provenienti dalla discretizzazione di EDP.
- ▶ Spesso il processo di convergenza è **lento**.
- ▶ In tutti i casi in cui è **rilevante** il tempo di calcolo si può ricorrere a tecniche di accelerazione della convergenza;
  - ▶ modellistica tridimensionale (**dispositivi elettronici**, elaborazione di immagini, ecc...);

# Motivazioni



- ▶ Diversi metodi numerici generano successioni di vettori convergenti alla soluzione di un dato problema;
  - ▶ risoluzione di sistemi di grandi dimensioni provenienti dalla discretizzazione di EDP.
- ▶ Spesso il processo di convergenza è **lento**.
- ▶ In tutti i casi in cui è **rilevante** il tempo di calcolo si può ricorrere a tecniche di accelerazione della convergenza;
  - ▶ modellistica tridimensionale (**dispositivi elettronici**, elaborazione di immagini, ecc...);
  - ▶ calcolo del vettore PageRank di Google.

# Teoria generale



Si considerano metodi di accelerazione della convergenza basati sull'idea della **estrapolazione vettoriale**



# Teoria generale



Si considerano metodi di accelerazione della convergenza basati sull'idea della **estrapolazione vettoriale**



Data una successione vettoriale convergente  $\mathbf{x}_n$ , con

$$\lim_{n \rightarrow \infty} \mathbf{x}_n = \mathbf{s}$$

# Teoria generale



Si considerano metodi di accelerazione della convergenza basati sull'idea della **estrapolazione vettoriale**



Data una successione vettoriale convergente  $\mathbf{x}_n$ , con

$$\lim_{n \rightarrow \infty} \mathbf{x}_n = \mathbf{s}$$

**un metodo di accelerazione** consiste nel costruire una trasformazione  $T$  che produce una nuova successione  $\mathbf{y}_n$  la quale converge più velocemente al medesimo limite:

$$\lim_{n \rightarrow \infty} \|\mathbf{x}_n - \mathbf{s}\| / \|\mathbf{y}_n - \mathbf{s}\| = 0$$

Le tecniche di accelerazione prese in analisi in questo lavoro appartengono a due categorie:

- ▶ metodi basati su **estrapolazione polinomiale**;
- ▶ metodi di tipo  **$\varepsilon$ -algorithms**;

si prendono in considerazione applicazioni di tali tecniche per risolvere

- ▶ sistemi **lineari**
- ▶ sistemi di EDP **non lineari**

a partire da successioni ottenute con metodi iterativi.

---

*K. Jbilou, H. Sadok, "Some results about vector extrapolation methods and related fixed-point iterations", Journal of Computational and Applied Mathematics - 1991*

*W.H.A. Schilders, P.A. Gough, K. Whight, "Extrapolation techniques for improved convergence in semiconductor device simulation", NASECODE VIII - 1992*

# Metodi polinomiali

Sia  $\{\mathbf{x}_n\}$ ,  $n = 0, 1, 2, \dots$  successione di vettori in  $\mathbb{R}^N$  con limite  $s$ . I **metodi polinomiali** forniscono una approssimazione di  $s$  come **combinazione lineare** dei termini della successione.

# Metodi polinomiali

Sia  $\{\mathbf{x}_n\}$ ,  $n = 0, 1, 2, \dots$  successione di vettori in  $\mathbb{R}^N$  con limite  $s$ . I **metodi polinomiali** forniscono una approssimazione di  $s$  come **combinazione lineare** dei termini della successione.

Fissato un numero opportuno  $k$  di elementi della successione, con  $k < N$ , definiti:

$$\mathbf{u}_i = \Delta \mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i, \quad \mathbf{w}_i = \Delta \mathbf{u}_i = \Delta^2 \mathbf{x}_i, \quad i = 0, 1, 2, \dots,$$

si risolve il sistema lineare dove  $\Lambda \zeta = \alpha$

$\Lambda$  matrice ( $N \times k$ )

$\alpha$  vettore colonna di dimensione  $N$ ,

$\zeta = (\zeta_0, \zeta_1, \dots, \zeta_{k-1})^T$  vettore dei coefficienti.

La matrice  $\Lambda$  e il vettore  $\alpha$  dipendono dal metodo.

# Metodi polinomiali

- ▶ Reduced Rank Extrapolation (**RRE**):

$$\Lambda = [w_n, w_{n+1}, \dots, w_{n+k-1}], \quad \alpha = -u_n$$

risolvendo, nel senso dei minimi quadrati, il sistema sovradeterminato, si ha

$$\mathbf{s}_k^{(n)} = \mathbf{x}_n + \sum_{j=0}^{k-1} \zeta_j \mathbf{u}_{n+j}.$$

# Metodi polinomiali

- ▶ Reduced Rank Extrapolation (**RRE**):

$$\Lambda = [w_n, w_{n+1}, \dots, w_{n+k-1}], \quad \alpha = -u_n$$

risolvendo, nel senso dei minimi quadrati, il sistema sovradeterminato, si ha

$$\mathbf{s}_k^{(n)} = \mathbf{x}_n + \sum_{j=0}^{k-1} \zeta_j \mathbf{u}_{n+j}.$$

- ▶ Minimal Polynomial Extrapolation (**MPE**):

$$\Lambda = [u_n, u_{n+1}, \dots, u_{n+k-1}], \quad \alpha = -u_{n+k}$$

determinati i coefficienti risolvendo ai minimi quadrati, posto  $\zeta_k = 1$ ,

$$\gamma_j = \frac{\zeta_j}{\sum_{i=0}^k \zeta_i}, \quad j = 0, 1, \dots, k \text{ si ha l'approssimazione}$$

$$\mathbf{s}_k^{(n)} = \sum_{j=0}^k \gamma_j \mathbf{x}_{n+j}$$

# Metodi $\varepsilon$ -algorithms

Sia  $\{\mathbf{x}_n\}$ ,  $n = 0, 1, 2, \dots$  successione **vettoriale** con limite  $s$ .  
I metodi di tipo  **$\varepsilon$ -algorithms** definiscono una trasformazione vettoriale per accelerare la convergenza di tale successione, basata sul **rapporto di determinanti**.

Sono algoritmi che si possono implementare con **formule ricorsive**; si considerano

- ▶ Vectorial Epsilon Algorithm (**VEA**)
- ▶ Topological Epsilon Algorithm (**TEA**)

# Metodi $\varepsilon$ -algorithms

## ► Vectorial $\varepsilon$ -algorithms

$$\varepsilon_{-1}^{(n)} = \mathbf{0}, \quad \varepsilon_0^{(n)} = \mathbf{x}_n, \quad n = 0, 1, \dots$$

$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + [\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}]^{-1} \quad k, n = 0, 1, \dots$$

$$\mathbf{z}^{-1} = \mathbf{z} / \|\mathbf{z}\|^2$$

# Metodi $\varepsilon$ -algorithms

## ► Vectorial $\varepsilon$ -algorithms

$$\varepsilon_{-1}^{(n)} = \mathbf{0}, \quad \varepsilon_0^{(n)} = \mathbf{x}_n, \quad n = 0, 1, \dots$$

$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + [\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}]^{-1} \quad k, n = 0, 1, \dots$$

$$\mathbf{z}^{-1} = \mathbf{z} / \|\mathbf{z}\|^2$$

## ► Topological $\varepsilon$ -algorithms

$$\varepsilon_{-1}^{(n)} = \mathbf{0}, \quad \varepsilon_0^{(n)} = \mathbf{x}_n, \quad n = 0, 1, \dots$$

$$\varepsilon_{2k+1}^{(n)} = \varepsilon_{2k-1}^{(n+1)} + \mathbf{y} / \left( \mathbf{y}, \Delta \varepsilon_{2k}^{(n)} \right)$$

$$\varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n+1)} + \Delta \varepsilon_{2k}^{(n)} / \left( \Delta \varepsilon_{2k+1}^{(n)}, \Delta \varepsilon_{2k}^{(n)} \right)$$

con  $\mathbf{y}$  vettore arbitrario.

# Metodi $\varepsilon$ -algorithms



- ▶ In entrambi viene generata ricorsivamente una tavola di valori a partire da un numero  $k$  di elementi della successione di partenza e si approssima il limite con

$$s = \varepsilon_{2k}^{(n)}$$

- ▶ Per TEA l'efficacia del metodo è legata alla scelta del vettore  $y$ .

# Applicazioni: sistemi lineari

- ▶ Risoluzione di un sistema lineare di **grandi dimensioni**:  $A\mathbf{x} = \mathbf{b} \Rightarrow$  è un classico ed importante problema per cui si adoperano algoritmi di **tipo iterativo**.

- ▶ Sia  $\{\mathbf{x}_i\}, i = 0, 1, 2, \dots$  successione di vettori di  $\mathbb{R}^N$ , generata linearmente a partire da un vettore iniziale  $\mathbf{x}_0$ ,

$$\mathbf{x}_{i+1} = G\mathbf{x}_i + \mathbf{d},$$

dove la matrice  $G$  e il vettore  $\mathbf{d}$  dipendono dal metodo scelto; sotto opportune ipotesi

$$\lim_{i \rightarrow \infty} \mathbf{x}_i = \mathbf{s} = A^{-1}\mathbf{b}.$$

- ▶ Il processo di convergenza può essere **computazionalmente oneroso**: è di interesse indagare l'efficacia e l'applicabilità dei metodi di accelerazione.

# Algoritmo di estrapolazione

## Initializations

guess  $\mathbf{x}_0^{(0)}, k$

$j \leftarrow 0, \mathbf{s}_0 \leftarrow \mathbf{x}_0^{(0)}$

## repeat

1. **Collection step**, si generano e collezionano  $k$  vettori:

$$\mathbf{x}_{i+1}^{(j)} = G\mathbf{x}_i^{(j)} + \mathbf{d}, \quad i = 0, \dots, k-1$$

2. **Extrapolation step**, si estrapola con uno dei metodi:

$$\mathbf{x}_0^{(j)}, \dots, \mathbf{x}_k^{(j)} \implies \mathbf{x}_e^{(j)} \rightarrow \mathbf{s}_{j+1}$$

3. **Restarting step**, si utilizza il vettore estrapolato come

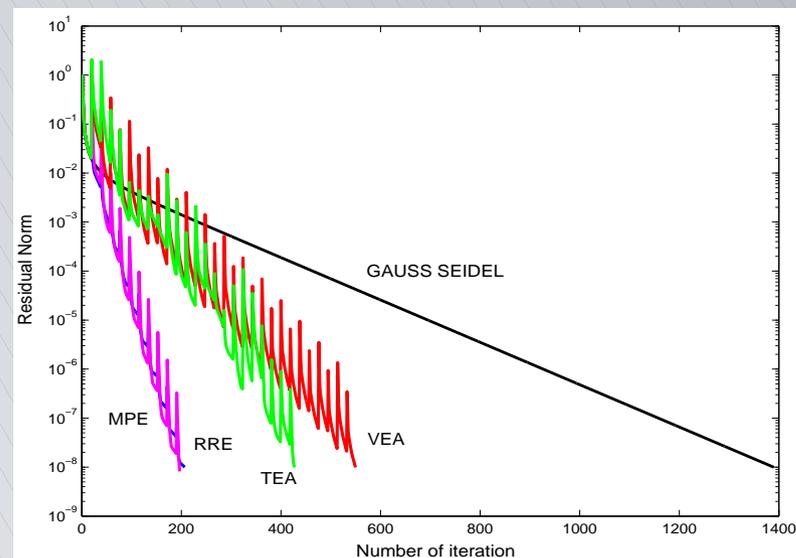
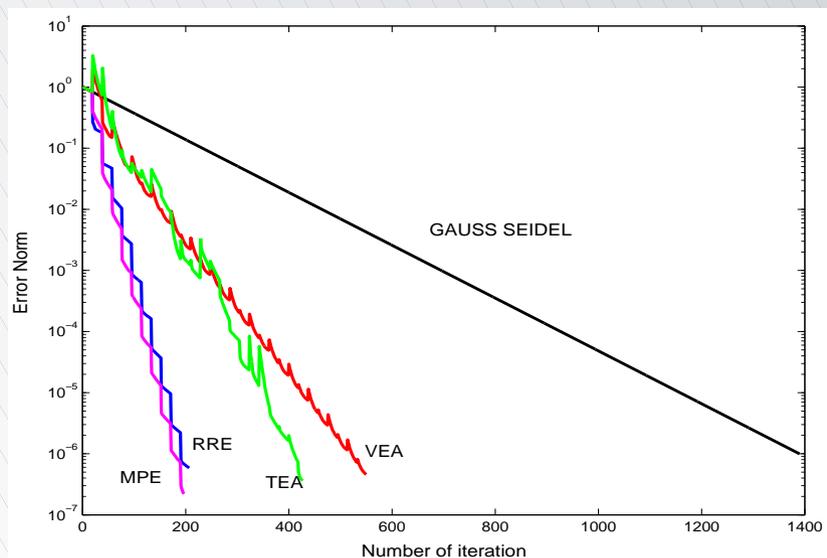
vettore iniziale:  $j \leftarrow j + 1, \mathbf{x}_0^{(j)} \leftarrow \mathbf{s}_{j+1}$

until  $\|A\mathbf{s}_{j+1} - \mathbf{b}\| < \varepsilon$

# Sistemi lineari: alcuni risultati

## Esempio 1

$$A = \text{tridiag}([-1, 2.1, -1]), N = 500, \mathbf{b} = A\mathbf{x}, \mathbf{x} = [1, \dots, 1]^T$$
$$\mathbf{x}_{i+1} = G\mathbf{x}_i + \mathbf{d} \rightarrow \text{Gauss Seidel}$$

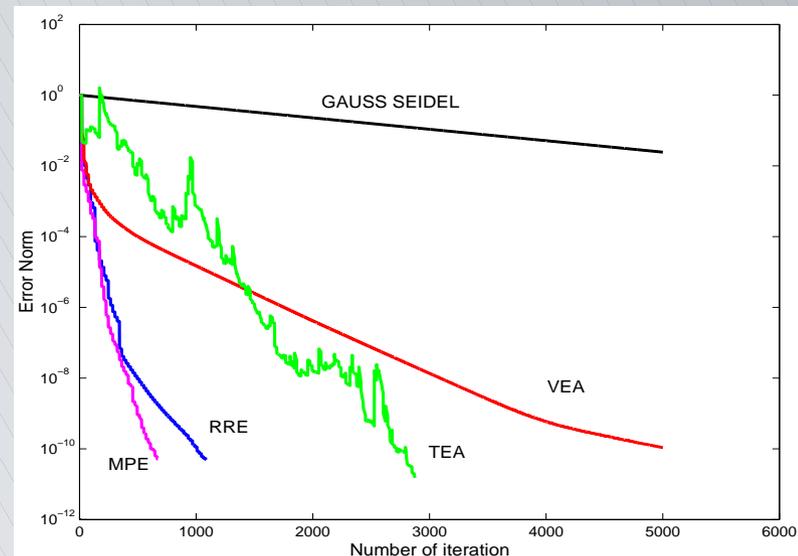
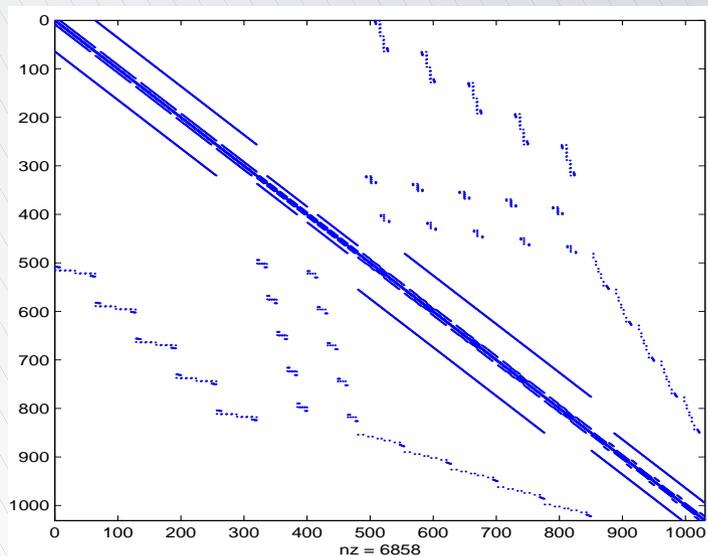


$k = 20$ ,  $\mathbf{y} = \text{randn}(N, 1)$  a media nulla e varianza unitaria.

# Sistemi lineari: alcuni risultati

## Esempio 2

ORSIRR 1- Oil reservoir simulation 21x21x5 irregular grid  
Harwell-Boeing Collection, (1030 x 1030), 6858 entries



$k = 20$ ,  $y = \text{randn}(N,1)$  a media nulla e varianza unitaria.

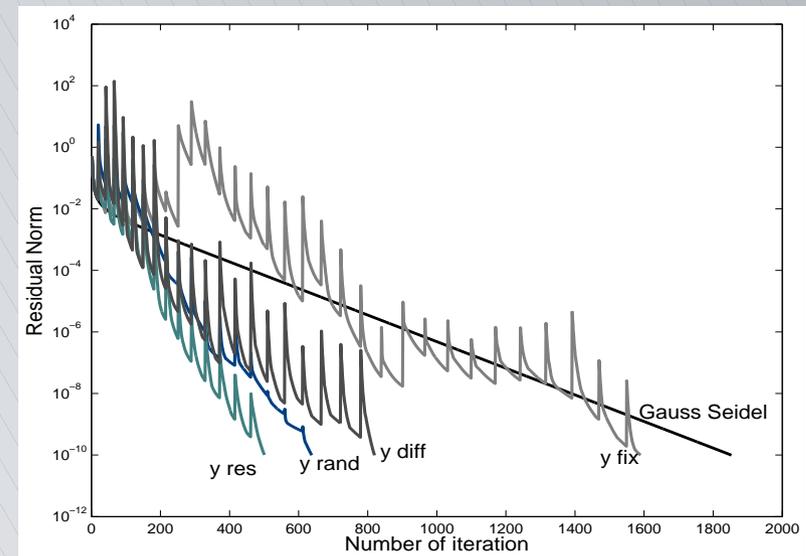
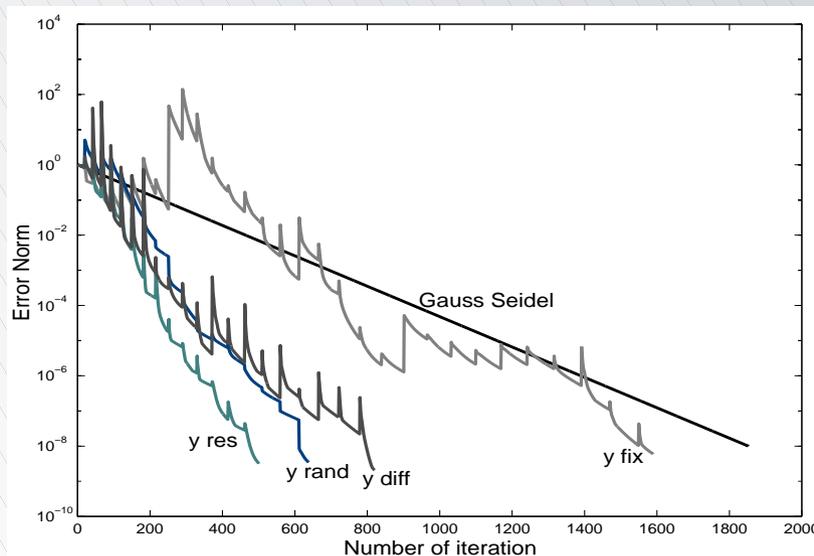
# TEA: scelta del vettore $y$

- ▶ Gli esempi numerici mostrano che l'efficacia del metodo è legata alla scelta di  $y$ ;
- ▶ sono in corso approfondimenti per trarre giustificazioni teoriche;
- ▶ alcune possibili scelte:
  - ▶  $y$  generato random e mantenuto fisso durante l'esecuzione ( $y$  fix);
  - ▶  $y$  generato random ad ogni restarting ( $y$  rand);
  - ▶  $y$  posto uguale al residuo  $(b - As_i)$  ad ogni restarting ( $y$  res);
  - ▶  $y$  posto uguale alla differenza  $(s_{i+1} - s_i)$  tra due iterate successive ( $y$  diff)

# TEA: scelta del vettore $y$

## Esempio 1

$$A = \text{tridiag}([-1, 2.1, -1]), N = 500, \mathbf{b} = A\mathbf{x}, \mathbf{x} = [1, \dots, 1]^T$$
$$\mathbf{x}_{i+1} = G\mathbf{x}_i + \mathbf{d} \rightarrow \text{Gauss Seidel}$$



- In seguito a diversi test numerici la scelta migliore risulta essere  $y$  res.

# Caso non lineare: mappa di Gummel

Tipico problema che nasce dalla fisica dei semiconduttori, descritto dal modello *Drift-Diffusion*

$$\begin{cases} \nabla \cdot (-\epsilon \nabla \psi) & = q(p - n + C) \\ \nabla \cdot \mathbf{J}_n & = R_n \\ \nabla \cdot \mathbf{J}_p & = R_p, \end{cases}$$

dove  $\mathbf{J}_n$  e  $\mathbf{J}_p$  sono le correnti associate ai portatori  $n$  e  $p$ .

$$\begin{cases} \mathbf{J}_n & = -q\mu_n n \nabla \psi + qD_n \nabla n \\ \mathbf{J}_p & = -q\mu_p p \nabla \psi - qD_p \nabla p, \end{cases}$$

e la funzione di ricombinazione di Shockley–Read–Hall è

$$R_n = -R_p = \frac{np - n_i^2}{\tau_p(n + n_i) + \tau_n(p + n_i)}.$$

# Modello Drift Diffusion

- ▶ Sistema composto da un'equazione di **diffusione** per il potenziale elettrostatico associata alle equazioni di **convezione-diffusione** per i portatori di carica;
- ▶ tre equazioni differenziali non lineari mutuamente accoppiate nelle tre variabili  $\psi$ ,  $n$  e  $p$ , in cui ciascuna dipende contemporaneamente da tutte e tre le incognite;
- ▶ si introducono i quasi-livelli di Fermi per elettroni e lacune,  $\phi_n$ ,  $\phi_p$  e le associate variabile di Slotboom  $u = \exp(-\phi_n)$ ,  $v = \exp(\phi_p)$ ;
- ▶ si usa un'algoritmo per **disaccoppiare** ogni equazione dalle altre due: **mappa di Gummel**

# Mappa di Gummel: algoritmo

## Initializations

guess  $\psi^{(0)}, \phi_n^{(0)}, \phi_p^{(0)}$

$k \leftarrow 0$

## repeat

1. **solve**  $-\lambda^2 \Delta \psi^{(k+1)} = -n_i e^{\psi^{(k+1)} - \phi_n^{(k)}} + n_i e^{\phi_p^{(k)} - \psi^{(k+1)}} + C$   
acceleration procedure on  $\psi$

2. **solve**  $\nabla \cdot [n_i \mu_n e^{\psi^{(k+1)}} \nabla u^{(k+1)}] = R(\psi^{(k+1)}, \phi_n^{(k)}, \phi_p^{(k)})$   
acceleration procedure on  $\phi_n$

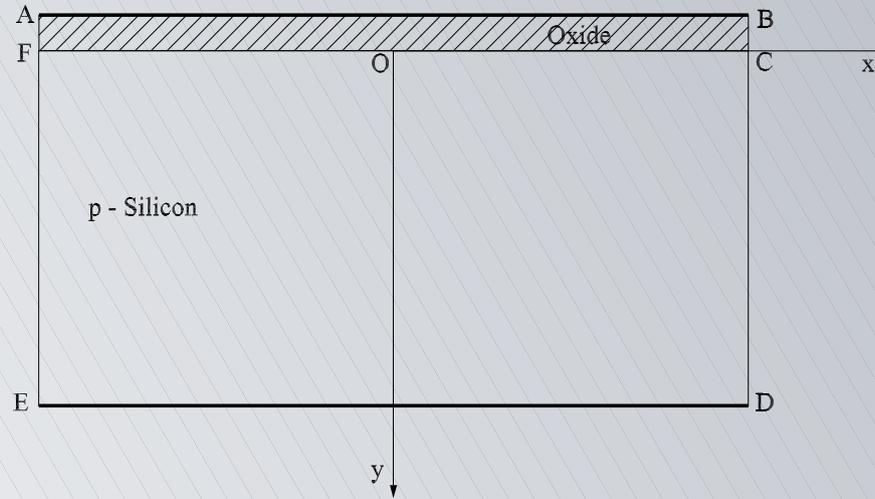
3. **solve**  $\nabla \cdot [n_i \mu_p e^{-\psi^{(k+1)}} \nabla v^{(k+1)}] = R(\psi^{(k+1)}, \phi_n^{(k)}, \phi_p^{(k)})$   
acceleration procedure on  $\phi_p$

4.  $k = k + 1$

**until**  $\max\{\varepsilon_\psi, \varepsilon_{\phi_n}, \varepsilon_{\phi_p}\} < \varepsilon$

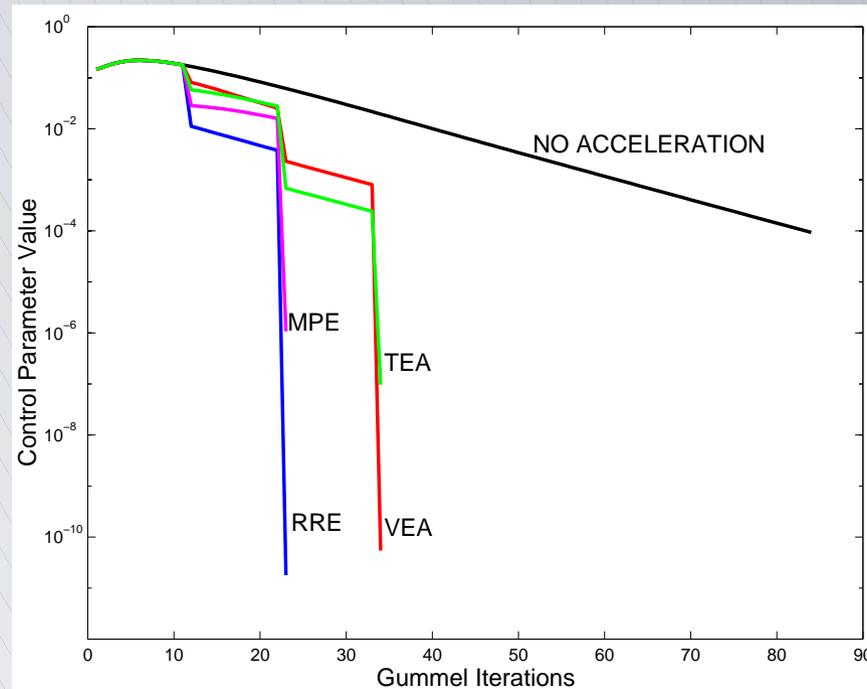
con  $\varepsilon_\psi, \varepsilon_{\phi_n}, \varepsilon_{\phi_p}$  errore relativo su iterate successive per  $\psi, \phi_n, \phi_p$ .

# Condensatore MOS - modello



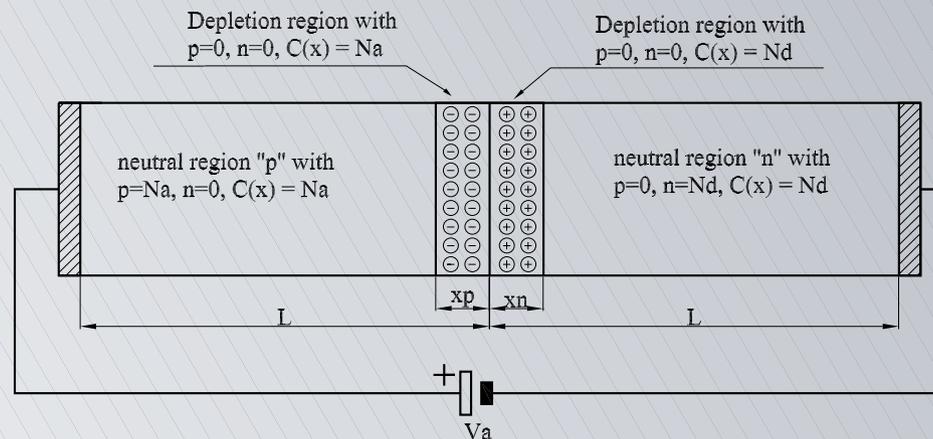
- ▶ Il modello è assimilabile ad un condensatore classico in cui
  - ▶ una delle due armature è metallica: il contatto  $AB$
  - ▶ l'altra è formata dal semiconduttore (p-Silicon): regione  $CDEF$
- ▶ Le tensioni sono applicate ai contatti  $AB$  ed  $ED$ .

# Condensatore MOS - risultati



- ▶ in condizioni stazionarie, nessuna corrente attraversa l'ossido,  $\mathbf{J}_n = \mathbf{J}_p = 0$  ed il modello si riduce all'equazione di Poisson (accelerazione su  $\psi$ );
- ▶ il tempo di calcolo è ridotto di circa 3 volte;
- ▶ l'efficienza è maggiore per i metodi polinomiali.

# Giunzione p-n



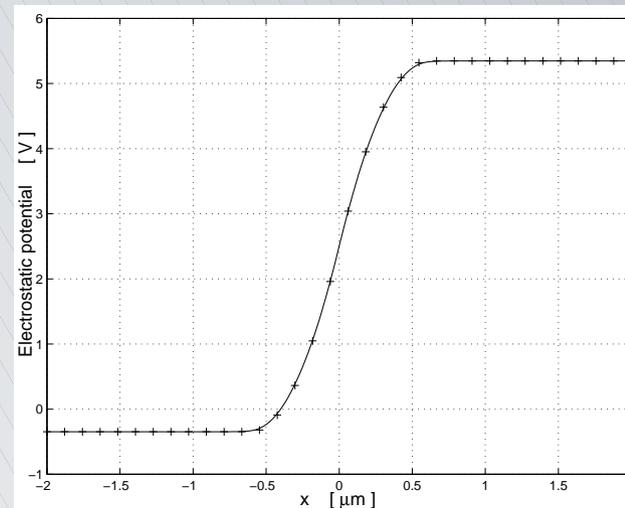
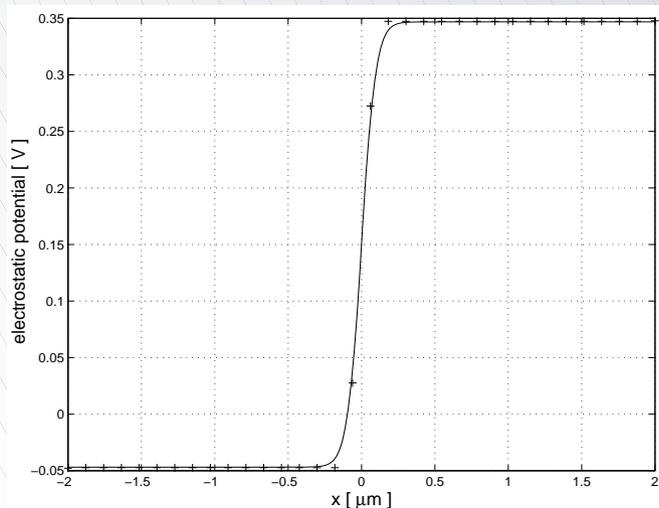
- ▶ Il diodo a giunzione è stato il primo dispositivo a semiconduttore reso disponibile commercialmente;
- ▶ è un rettificatore bipolare, la cui funzione ideale è quella di permettere il flusso di corrente elettrica in una direzione e di bloccarla nell'altra.

# Giunzione p-n brusca

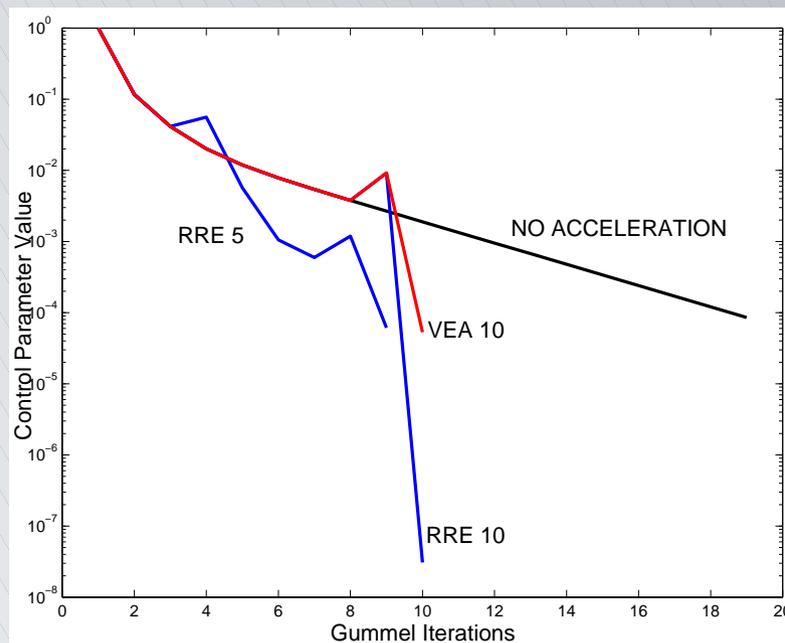
Giunzione brusca, simmetrica con i seguenti parametri

- ▶  $N_a = N_d = 10^{22} m^{-3}$  (drogaggio, funzione  $C(x)$ );
- ▶  $\tau_n = \tau_p = 10^{-5} s$  (generazione, funzione  $R$ );
- ▶  $\mu_n = \mu_p = 0.06 m^2/s$  (mobilità, funzioni  $\mu_n, \mu_p$ ).
- ▶  $V_a = 0.3V, V_a = -5V$

Forti gradienti si presentano in corrispondenza dell'unione dei due drogaggi differenti; la soluzione presenta strati limite.

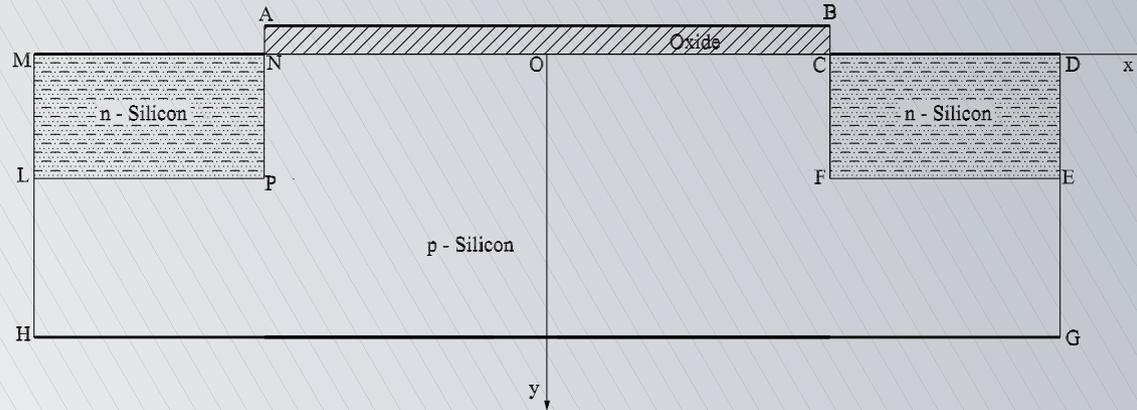


# Giunzione $p - n$ brusca - risultati



- ▶ L'accelerazione riduce di circa 2 volte il tempo di calcolo;
- ▶ l'efficienza è maggiore per i metodi polinomiali.

# MOSFET



- ▶ dispositivo fondamentale della moderna microelettronica;
- ▶ da notare la presenza del condensatore MOS e delle giunzioni p–n.

# Conclusioni

- ▶ Le tecniche di accelerazione hanno fornito buoni risultati per i sistemi lineari, in particolare
  - ▶ I metodi polinomiali sono più efficienti;
  - ▶ si sta indagando per migliorare le prestazioni degli  $\varepsilon$ -algorithms.
- ▶ L'applicazione alla mappa di Gummel fornisce risultati buoni, anche se preliminari.
  - ▶ Si intende estendere lo studio al caso bidimensionale per la giunzione p-n;
  - ▶ applicare l'accelerazione ad altri dispositivi (MOSFET,...).

R. Bertelle, M.R. Russo, "*An Approach to the Gummel Map by Vector Extrapolation Methods*", submitted.