Elementi di Informatica e Applicazioni Numeriche T

I File in Octave

Octave ed il File System

Come ogni altro programma:

- Quando Octave viene eseguito…
- …Viene associato ad una directory del File System

La directory corrente:

- È quella visibile nel File Browser
- Può essere cambiata, sempre utilizzando il file browser

Un consiglio:

- Create una nuova cartella per ogni esercitazione
- Prima di scrivere funzioni/script, impostatela come cartella corrente

Octave ed il File System

È possibile interagire con il File System mediante comandi

Per visualizzare il percorso della directory corrente si usa:

```
pwd  % sta per "Print Working Directory"
```

Per visualizzare il contenuto della directory corrente:

```
ls  % sta per "list"
```

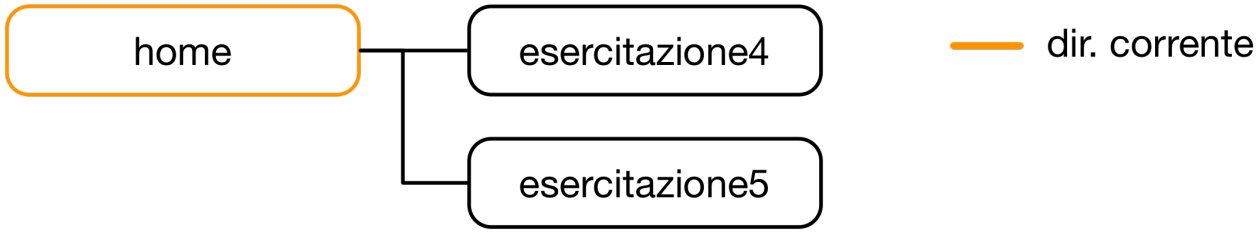
Per cambiare la directory corrente:

```
cd <percorso> % sta per Change Directory
```

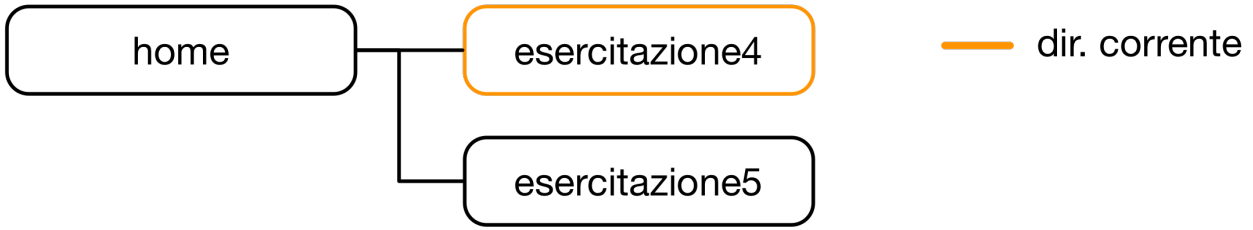
Octave ed il File System

Il percorso passato a `cd` può essere assoluto o relativo

Per esempio, partendo da questo stato:



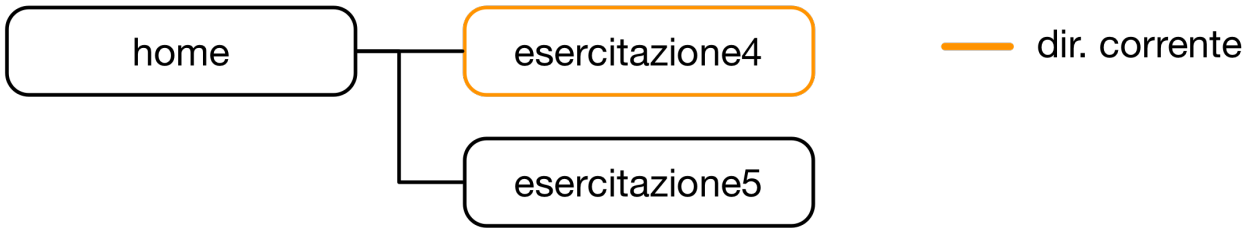
Con "`cd esercitazione4`", otteniamo:



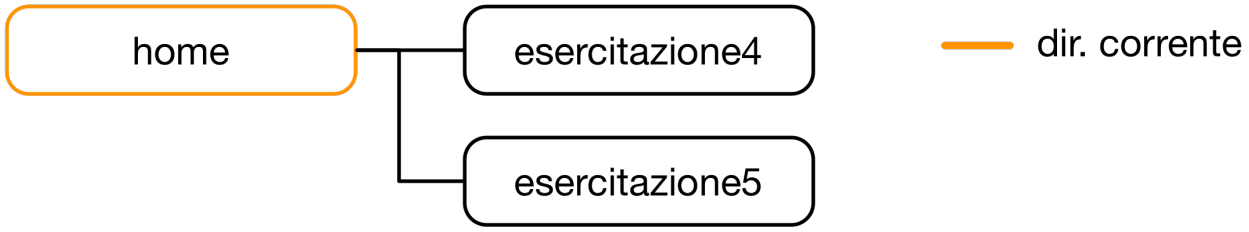
Octave ed il File System

Il percorso passato a `cd` può essere assoluto o relativo

Invece, partendo da questo stato:



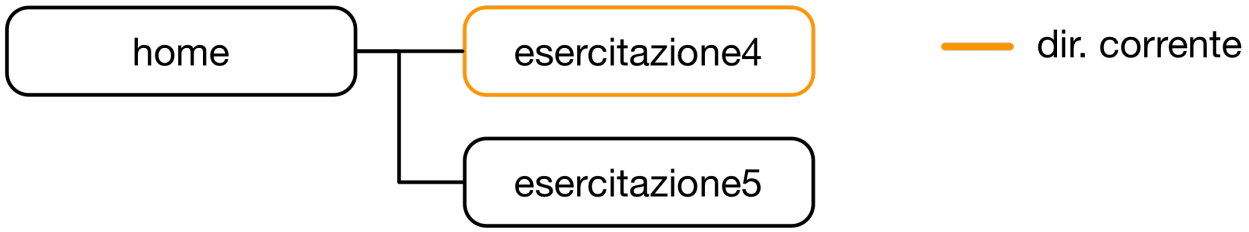
Con "`cd ..`", otteniamo:



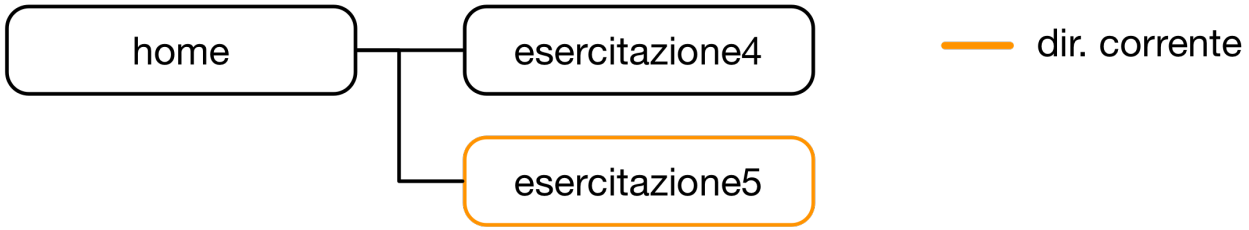
Octave ed il File System

Il percorso passato a `cd` può essere assoluto o relativo

Ancora, partendo da questo stato:



Con "`cd ../esercitazione5`", otteniamo:



I/O da File in Octave

Octave permette di leggere/scrivere dati su file

È un modo comodo per:

- Salvare informazioni importanti
- Passare dati tra programmi (e.g. Octave/Matlab/Excel)

Ci focalizziamo sui **file CSV (Comma Separated Values):**

- Sono file di testo
- Ogni riga corrisponde ad un vettore
- Gli elementi su ogni riga sono distinti da un separatore
  - Per esempio la virgola "`,`" (tipicamente Linux e Mac)
  - Oppure "`;`" (tipicamente Win e MS Excel)

CSV: Un Esempio

Un esempio di file CSV:

```
1; 2; 4; 8; 16; 32
1; 3; 9; 27; 81; 243
1; 4; 16; 64; 256; 1024
```

- Di fatto è un formato molto semplice
- Il nome deve terminare con l'estensione "`.csv`"

Excel può:

- Aprire (doppio click) i csv in dui i dati sono separati con "`;`"
- Salvare csv in dui i dati sono separati con "`;`"
- Importare qualsiasi tipo di file csv

CSV in Octave

In Octave, i file CSV si leggono con:

```
csvread(<percorso file>);
dlmread(<percorso file>, <separatore>)
```

- `csvread` assume che il separatore sia "`,`"
- `dlmread` funziona qualunque sia il separatore

`<percorso file>` è il percorso (relativo o assoluto) del file

- Caso importante: se il file è nella cartella corrente...
- ...il suo percorso relativo coincide con il suo nome

CSV in Octave

In Octave, i file CSV si leggono con:

```
csvwrite(<percorso file>, <dato>);
dlmwrite(<percorso file>, <dato>, <separatore>)
```

- `csvwrite` assume che il separatore sia "`,`"
- `dlmwrite` funziona qualunque sia il separatore

Un consiglio: utilizzare `dlmread`/`dlmwrite`

- Poter passare dati facilmente da/verso Excel è molto utile

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Fattoriale degli Elementi di un Vettore

Esercizio: Fattoriale di un Vettore

Octave fornisce la funzione:

```
factorial(V)
```

Che restituisce un vettore con il fattoriale di ogni numero in `v`

- Gli elementi di `v` devono essere numeri interi
- Il fattoriale $n!$ di un numero n è definito come:

$$n! = \begin{cases} 1 & \text{se } n = 0 \\ \prod_{i=1}^{n} i & \text{se } n > 0 \end{cases}$$

Esercizio: Fattoriale di un Vettore

Si definisca una funzione:

```
XXX_factorial(V)
```

Che replichi tale comportamento

- Si verifichi il funzionamento sui dati in **`ch5_data2.csv`**
- Potrebbe essere utile scrivere un file di script
- Ricordate che un file di script:
  - È un file di testo con estensione **`.m`**
  - Contiene una sequenza di comandi
  - Può essere eseguito con **`<nome file> + [invio]`**

Soluzione

Una possibile soluzione:

```
function z = ch5_factorial(V)
    z = [];
    for ii = 1:length(V)
        z(ii) = 1;
        for jj = 1:V(ii)
            z(ii) = z(ii) * jj;
        end
    end
end
```

Soluzione

Una possibile alternativa:

```
function z = ch5_factorial(V)
    z = zeros(1, length(V));
    for ii = 1:length(V)
        z(ii) = prod(1:V(ii));
    end
end
```

- Un range `a:b` equivale a `[]` se `a > b`
- `prod([])` restituisce `1`

Soluzione

Un possibile script di test:

```
% Carico i dati
V = dlmread('ch5_data2.csv');

% Calcolo i fattoriali
z1 = ch5_factorial(V);

% Ripeto l'operazione con la funzione di Octave
z2 = factorial(V);

% Verifico che il risultato sia corretto
all(z1 == z2)
```

Elementi di Informatica e Applicazioni Numeriche T

Plot in Octave

Plot in Octave

Molte delle funzioni di Octave operano su vettori:

- Per esempio `factorial` prende in ingresso un vettore
- Altro esempio: gli operatori elemento per elemento (es. `.^`)

Questo comportamento **risulta comodo per disegnare**

- Per esempio, utilizzando:

```
x = -5:0.1:5;
y = x.^2;
```

- Otteniamo velocemente un vettore di coordinate `x`...
- ...ed i valore di x_i^2 per ogni elemento x_i

Plot in Octave

A questo punto, si può disegnare la funzione con:

```
plot(x, y);
```

La funzione `plot` produce un grafico a linee

Si può decidere il formato della linea con:

```
plot(x, y, fmt);
```

Dove `fmt` è una stringa che decide il formato:

- Se `fmt` è `'b'` la linea è blu, se `fmt` è `'r'` è rossa
- Vedremo altri esempi più avanti (se siete curiosi: `help` o `doc`)

Plot in Octave

Si può disegnare più di una funzione con la sintassi:

```
plot(x1, y1, fmt1, x2, y2, fmt2, ...);
```

Per esempio, potete disegnare sia x^2 che x^3 con:

```
x = -5:.1:5;
plot(x, x.^2, 'b', x, x.^3, 'r')
```

- La linea per x^2 sarà blu
- La linea per x^3 sarà rossa

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Valutazione di Polinomi

Esercizio: Valutazione di Polinomi

Octave permette di valutare un polinomio con:

```
polyval(p, X)
```

Dove:

- `X` contiene i valori per cui fare la valutazione
- `p` è un vettore di n coefficienti che definiscono il polinomio:

$$p_1 x^{n-1} + p_2 x^{n-2} + \ldots + p_n$$

Per esempio `polyval([3, 0, 4], 2)` restituisce:

$$3 \times 2^2 + 0 \times 2^1 + 4 = 16$$

Esercizio: Valutazione di Polinomi

Si definisca una funzione:

```
XXX_polyval(X)
```

Che replichi tale comportamento

- Suggerimento: prima si assuma che `x` sia uno scalare...
- ...poi si estenda la funzione per gestire un vettore

Per verificare che tutto funzioni:

- Valutate un polinomio con `XXX_polyval` e con `polyval`
- Disegnate le funzioni ottenute (sullo stesso plot)
- Se le due curve sono sovrapposte, va tutto bene

Soluzione

Una possibile soluzione:

```
function z = ch5_polyval(p, X)
    z = 0;
    for ii = 1:length(p)
        z = z + p(ii) .* X.^(length(p)-ii);
    end
end
```

Idea generale: elevo ad esponente tutte le `X` e sommo

- Funziona sia per `X` scalare che per `X` vettore...
- ...Perché `+`, `.*` e `.^` operano elemento per elemento

Soluzione

Una soluzione alternativa:

```
function z = ch5_polyval(p, X)
    n = length(p)-1:-1:0; % esponenti
    z = [];
    for ii = 1:length(X)
        z(ii) = dot(p, X(ii).^n); % prodotto scalare
    end
end
```

Idea generale: elevo ciascuna `X` a tutti gli esponenti e sommo

- Uso il prodotto scalare per applicare i coefficienti e sommare

Soluzione

Un possibile script di test:

```
% Definisco un polinomio
p = [1, 3, -2, -1, 7];
% Definisco i valori per cui va calcolato
X = -5:.1:5;

% Valuto il polinomio utilizzando i due metodi
Z0 = ch5_polyval(p, X);
Z1 = polyval(p, X);

% Disegno le due curve
plot(X, Z0, 'b', X, Z1, 'r')
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Espansione in Serie di Taylor

Esercizio: Espansione in Serie di Taylor

Data una funzione f ed un punto x_0, la serie di Taylor:

- Permette di rappresentare la funzione come un polinomio...
- ...avente un numero infinito di termini

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots$$

In forma compatta:

$$f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}}{i!}(x - x_0)^i$$

- Requisito: f deve essere infinitamente differenziabile in x_0
- Caveat: per alcune funzioni la rappresentazione non è esatta

Esercizio: Espansione in Serie di Taylor

Considerando solo n termini:

- Otteniamo una approssimazione di f in ogni punto
- Calcolata conoscendo, per un solo punto x_0 :
  - Il valore di $f(x_0)$
  - Il valore di tutte le derivate, sempre in x_0

Un caso facile: e^x

- Per $x_0 = 0$, la funzione e^x vale 1...
- ...e tutte le derivate di e^x valgono 1

Esercizio: Espansione in Serie di Taylor

Si definisca una funzione:

```
XXX_taylor_exp(X, n)
```

Che approssimi e^x utilizzando la serie di Taylor per $x_0 = 0$:

- Si considerino solo i primi $\mathbf{n}$ termini della serie
- Si valuti l'approssimazione per tutti i valori x_i nel vettore $\mathbf{X}$

Per verificare se funziona:

- Calcolate il valore di e^x per un certo range
- Calcolate il valore dell'approssimazione
- Disegnate le due curve (provate a cambiare $\mathbf{n}$)

Soluzione

Una possibile soluzione:

```
function z = ch5_taylor_exp(x, n)
    z = 1; % termine di ordine 0
    for ii = 1:n
        % calcolo il coefficiente del polinomio
        % di Taylor di ordine ii
        c = 1 / factorial(ii);
        % sommo un termine del polinimio
        z = z + c .* x.^ii;
    end
end
```

Soluzione

Una alternativa:

```
function z = ch5_taylor_exp(x, n)
    % Calcolo i coefficienti del polinomio di Taylor
    c = 1./factorial(n:-1:0);
    % Valuto il polinomio
    z = polyval(c, x);
end
```

Soluzione

Un possibile script di test:

```
n = 5; % Decido il valore di n
x = 0:.2:4; % Il range per il disegno
z1 = e.^x; % I valori di e^x

% Calcolo l'approssimazione
z2 = ch5_taylor_exp(x, n);

% Disegno le due curve
plot(x, z1, 'b', x, z2, 'r');
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Divisioni Ripetute

Esercizio: Divisioni Ripetute

Per ottenere la rappresentazione binaria di un intero n si usa:

```
dec2bin(n)
```

- Funziona solo se n non ha parte frazionaria
- Restituisce una stringa

Si definisca una funzione:

```
XXX_dec2bin(n)
```

- Che faccia lo stesso, utilizzando il metodo delle divisioni ripetute
- Per semplicità, si restituisca un vettore di 0/1
- Si verifichi la correttezza confrontandosi con la funzione di Octave

Esercizio: Divisioni Ripetute

Alcune informazioni utili:

- La divisione intera si fa con:

```
idivide(<dividendo>, <divisore>)
```

- Il resto della divisione intera si ottiene con:

```
mod(<dividendo>, <divisore>)
```

Soluzione

Una possibile soluzione:

```
function z = ch5_dec2bin(n)
    z = [];
    ii = 1;
    while n > 0
        z(ii) = mod(n, 2);
        n = idivide(n, 2);
        ii = ii + 1;
    end
    % Inverto la sequenza dei bit
    z = z(end:-1:1);
end
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Moltiplicazioni Ripetute

Esercizio: Moltiplicazioni Ripetute

Si definisca una funzione:

```
XXX_dec2bin_f(x)
```

Che, dato un numero reale x in $]0, 1[$

- Restituisca la sua rappresentazione binaria (vettore di 0/1)…
- …Ottenuta mediante il metodo delle moltiplicazioni ripetute
- Si interrompa il calcolo alla 10^a cifra binaria

Si noti che non è necessario rappresentare il separatore "`.`"

- La parte intera si può ottenere con

```
fix(<dato>)
```

Soluzione

Una possibile soluzione:

```
function z = ch5_dec2bin_f(n)
    z = [];
    for ii = 1:10
        y = n * 2;
        z(ii) = fix(y);
        n = y - z(ii);
    end
end
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Somma di Matrici

Esercizio: Somma di Matrici

Si definisca una funzione:

```
XXX_msum(A, B)
```

Che calcoli a somma di due matrici

- Si eviti di utilizzare l'operatore + applicato a vettori
  - In altre parole: sommate gli elementi uno ad uno
- Si verifichi la correttezza usando i metodi di Octave

Soluzione

Una possibile soluzione:

```
function Z = ch4_msum(A, B)
    m = rows(A);
    n = columns(A);
    Z = zeros(m, n);
    for ii = 1:m
        for jj = 1:n
            Z(ii, jj) = A(ii,jj) + B(ii,jj);
        end
    end
end
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Trasposizione di Matrice

Esercizio: Trasposizione di Matrice

Si definisca una funzione:

```
XXX_transpose(A)
```

Che calcoli la trasposizione della matrice `A`

- Si verifichi la correttezza usando i metodi di Octave

Soluzione

Una possibile soluzione:

```
function Z = ch4_msum(A)
    m = rows(A);
    n = columns(A);
    Z = zeros(n, m);
    for ii = 1:m
        for jj = 1:n
            Z(jj, ii) = A(ii,jj);
        end
    end
end
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Prodotto Matriciale

Esercizio: Prodotto Matriciale

Si definisca una funzione:

```
XXX_mprod(A, B)
```

Che calcoli il prodotto matriciale di `A` e `B`

- Si verifichi la correttezza usando i metodi di Octave

Soluzione

Una possibile soluzione:

```
function Z = ch4_mprod(A, B)
    m = rows(A);
    n = columns(B);
    Z = zeros(m, n);
    for ii = 1:m
        for jj = 1:n
            % Qui potrei usare anche "dot"
            Z(ii,jj) = A(ii,:) * B(:, jj);
        end
    end
end
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Conteggio di Elementi

Esercizio: Conteggio di Elementi

Si definisca una funzione:

```
[U, C] = XXX_count(V)
```

Che, dato un vettore di ingresso `V`, restituisca `U` e `C` tali che:

- `U` contenga gli elementi distinti di `V`
- Per ogni `v` in `U`, `C` contenga il numero di occorrenze di `v` in `U`

In altre parole, la funzione deve contare le occorrenze di ogni elemento.

- Si sperimenti la funzione sui dati nel file **`ch5_data1.csv`**
- I risultati sono in **`ch5_count_u.csv`** e **`ch5_count_c.csv`**
- I file sono scaricabili dal sito del corso

Soluzione

Una possibile soluzione:

```
function [U, C] = ch5_count(V)
    U = unique(V);
    C = zeros(1, length(U));
    for ii = 1:length(U)
        for jj = 1:length(V)
            if V(jj) == U(ii)
                C(ii) = C(ii) + 1;
            end
        end
    end
end
```

Soluzione

Una alternativa:

```
function [U, C] = ch5_count(V)
    U = unique(V);
    C = zeros(1, length(U));
    for ii = 1:length(U)
        C(ii) = sum(V == U(ii));
    end
end
```

Soluzione

Un possibile script di test:

```
% Carico i dati
V = dlmread('ch5_data1.csv');
% Ottengo gli elementi unici ed i conteggi
[U, C] = ch5_count(V);
% Carico le soluzioni
U2 = dlmread('ch5_count_u.csv');
C2 = dlmread('ch5_count_c.csv');
% Controllo che i risultati siano corretti
all(U == U2)
all(C == C2)
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Somma di Fattoriali

Esercizio: Somma di Fattoriali

Si definisca una funzione:

```
XXX_serie_fatt(n)
```

Che calcoli per un ordine n il valore della serie:

$$\sum_{i=1}^{n} i!$$

Si provi a definire la funzione utilizzando:

- Due cicli innestati
- Un solo ciclo
- Nessun ciclo

Soluzione

Una possibile soluzione con cicli innestati:

```
function z = ch5_serie_fatt(n)
    z = 0;
    for ii = 1:n
        % calcolo il fattoriale di ii
        fatt = 1;
        for jj = 1:ii
            fatt = fatt * jj;
        end
        % sommo il fattoriale
        z = z + fatt;
    end
end
```

Soluzione

Una possibile soluzione con un solo ciclo:

```
function z = ch5_serie_fatt(n)
    z = 0;
    for ii = 1:n
        % calcolo il fattoriale di ii
        fatt = prod(1:ii); % oppure factorial(ii);
        % sommo il fattoriale
        z = z + fatt;
    end
end
```

- L'idea è di rimpiazzare un ciclo con una chiamata a funzione

Soluzione

Un'altra possibile soluzione con un solo ciclo:

```
function z = ch5_serie_fatt(n)
    z = 0;
    fatt = 1;
    for ii = 1:n
        % aggiorno il fattoriale di ii
        fatt = fatt * ii;
        % sommo il fattoriale
        z = z + fatt;
    end
end
```

- Idea: calcolo del fattoriale sulla base dei risultati precedenti

Soluzione

Una soluzione senza cicli

```
function z = ch5_serie_fatt(n)
 z = sum(factorial(1:n));
end
```

- Utilizzo funzioni che operano su vettori
- In realtà i cicli ci sono, ma nelle funzioni **`sum`** e **`factorial`**

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Derivata di un Polinomio

Esercizio: Derivata di un Polinomio

La derivata di un polinomio:

$$p_1 x^{n-1} + p_2 x^{n-2} + \ldots + p_n$$

È un polinomio:

$$(n-1)p_1 x^{n-2} + (n-2)p_2 x^{n-3} + \ldots + p_{n-1}$$

Octave permette di calcolarlo utilizzando:

```
polyder(p)
```

- Si definisce una funzione `XXX_polyder(p)`...
- ...Che replichi tale comportamento

Soluzione

Una possibile soluzione:

```
function p2 = ch4_polyder(p)
    n = length(p)-1; % ordine del polinomio
    % Coefficienti dovuti alla derivazione dei monomi
    ce = n:-1:1;
    % Primi n coefficienti del polinomio originale
    cp = p(1:n);
    % Faccio il prodotto ed ottengo la derivata
    p2 = ce .* cp;
end
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Integrale di un Polinomio

Esercizio: Integrale di un Polinomio

L'integrale di un polinomio:

$$p_1 x^{n-1} + p_2 x^{n-2} + \ldots + p_n$$

È un polinomio:

$$\frac{1}{n} p_1 x^n + \frac{1}{n-1} p_2 x^{n-1} + \ldots + p_n x$$

Octave permette di calcolarlo utilizzando:

```
polyint(p)
```

- Si definisce una funzione `XXX_polyint(p)`...
- ...Che replichi tale comportamento

Soluzione

Una possibile soluzione:

```
function p2 = ch4_polyint(p)
    n = length(p)-1; % ordine del polinomio
    % Coefficienti dovuti all'integrazione dei monomi
    ce = 1 ./ (n+1:-1:1);
    % Integrale per i primi n termini del polinomio
    p2 = ce .* p;
    % Estendo il polinomio risultato con uno 0
    p2(end+1) = 0;
end
```

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Permutazione Inversa

Esercizio: Permutazione Inversa

Si può ottenere una permutazione dei numeri da 1 a n con:

```
randperm(n)
```

- La permutazione è casuale
- È un buon modo per "mescolare" un vettore...
- ...Una operazione importante nelle simulazioni

Esempio:

```
v = [2, 4, 6, 8];
p = randperm(length(v)) % denota (e.g.) [2, 4, 1, 3]
w = v(p) % [4, 8, 2, 6]
```

Esercizio: Permutazione Inversa

Si definisca una funzione:

```
XXX_invperm(v1, v2)
```

Che, dati due vettori contenenti gli stessi elementi in ordine diverso:

- Restituisce un vettore `w` tale che...
- ...`w` contiene le posizioni in `v2` degli elementi di `v1`

`v2` è una sorta di permuazione inversa:

- Se per un vettore `p` abbiamo che `v1(p) == v2`...
- ...e `p2 = XXX_invperm(v1, v2)`
- ...Allora `v2(p2) == v1`

Soluzione

Una possibile zSoluzione:

```
function p = ch4_invperm(v1, v2)
    n = length(v1);
    p = zeros(1, n);
    for ii = 1:n
        for jj = 1:n
            if v1(ii) == v2(jj)
                p(ii) = jj;
                break; % interrompe il ciclo su jj
            end
        end
    end
end
```

Soluzione

Una possibile alternativa:

```
function p = ch4_invperm(v1, v2)
    n = length(v1);
    p = zeros(1, n);
    for ii = 1:n
        p(ii) = find(v2 == v1(ii));
    end
end
```

Soluzione

Un possibile script di test:

```
v1 = [4, 7, 9, 1, 5] % costruisco un vettore
p = randperm(length(v1)) % permutazione casuale
v2 = v1(p) % "Mescolo" v1

% Calcolo la permutazione inversa
p2 = ch4_invperm(v1, v2)

% stampo il risultato dell'indicizzazione con p2.
% Deve essere uguale a v1
v2(p2)
```