

Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Case study 2 An Inventory system

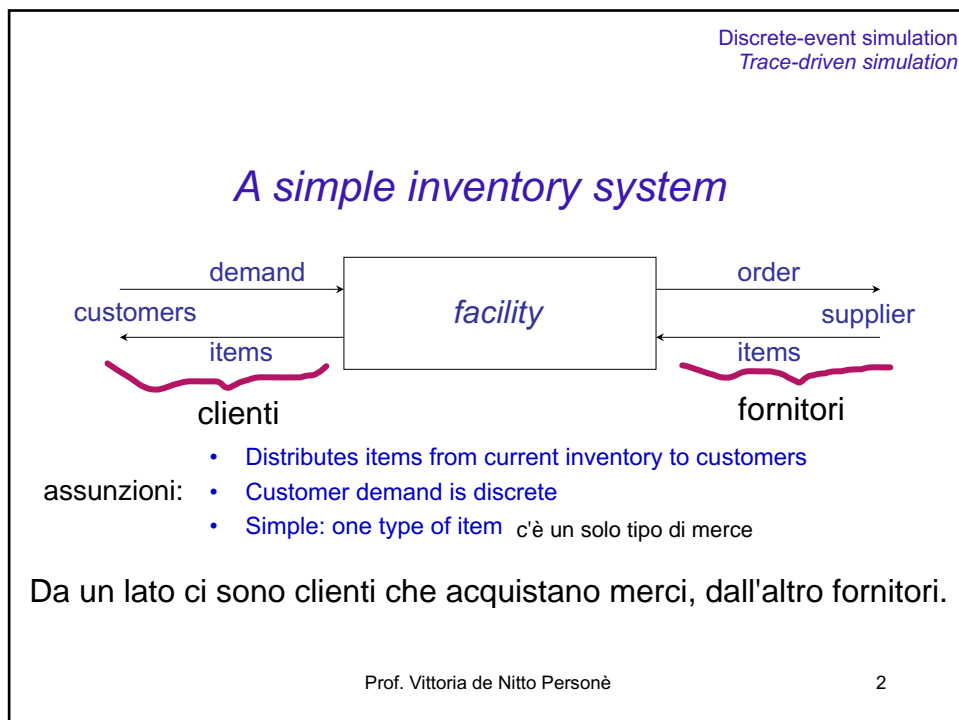
Università degli studi di Roma Tor Vergata
Department of Civil Engineering and Computer Science Engineering

Copyright © Vittoria de Nitto Personè, 2021
<https://creativecommons.org/licenses/by-nc-nd/4.0/>



1

Finora abbiamo sempre parlato di "code", che in questo caso di studi invece non sono presenti. Abbiamo un sistema "merci", diciamo un magazzino che compra e vende, periodicamente deve fare l'inventario, delle merci che sono presenti. Vediamolo ad alto livello:



2

1. **Goals**
2. Conceptual model
3. Specification model
4. Computational model
5. Verify
6. Validate

Discrete-event simulation
Trace-driven simulation

Inventory policy

- **Transaction reporting**
 - Inventory review after each transaction
 - Significant labor may be required
 - Less likely to experience “shortage” essendo più frequente, è meno probabile di avere carenze di prodotti.
- **Periodic inventory review**
 - Inventory review is periodic
 - Items are ordered, if necessary, only at review times
 - (s, S) are the min,max inventory levels, $0 \leq s < S$

L'ordine viene fatto solo se la giacenza è inferiore rispetto ad un parametro 's'.

Search for (s, S) that minimize cost

Prof. Vittoria de Nitto Personè

3

3

1. Goals
2. **Conceptual model**
3. Specification model
4. Computational model
5. Verify
6. Validate

Discrete-event simulation
Trace-driven simulation

Inventory System Costs

Abbiamo costi a diversi livelli.

- Holding cost: for items in inventory
- Shortage cost: for unmet demand
- Setup cost: fixed cost when order is placed
- Item cost: per-item order cost
- Ordering cost: sum of setup and item costs

Costo mantenimento singola merce.

"Penalità" nel non avere una merce.

Tipo spesa imballaggio

Costo ordine

Additional Assumptions

- Back ordering is possible
- No delivery lag
- Initial inventory level is S
- Terminal inventory level is S

è possibile che ci venga chiesto più di quello che c'è.

Ciò che viene acquistato è subito consegnato.

ad inizio e fine osservazione, l'inventario sarà pieno.

Le condizioni devono essere le stesse, all'inizio e alla fine.

Potrei anche considerare stato iniziale = stato finale Vuoto, l'importante è che sia significativo per il sistema.

I : inventory level

o : amount ordered

d : demand quantity

Le nostre variabili di stato.

Prof. Vittoria de Nitto Personè

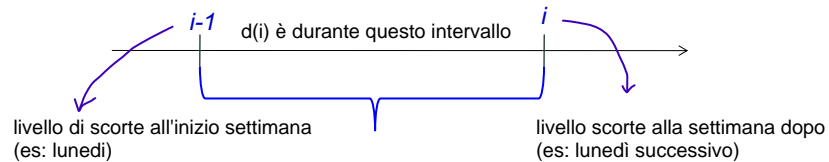
4

4

1. Goals
2. Conceptual model
3. **Specification model**
4. Computational model
5. Verify
6. Validate

Discrete-event simulation
Trace-driven simulation

- Time begins at $t=0$
- Review times are $t=0, 1, 2, \dots$ tempo discreto
- I_{i-1} : inventory level at beginning of i^{th} interval
- O_{i-1} : amount ordered at time $t=i-1$, ($O_{i-1} \geq 0$)
- d_i : demand quantity during i^{th} interval ($d_i \geq 0$)
- Inventory at end of interval **can be negative** (back ordering)



Prof. Vittoria de Nitto Personè

5

5

Discrete-event simulation
Trace-driven simulation

Inventory level considerations

- Inventory level is reviewed at $t=i-1$ (inizio settimana)
- If at least s , no order is placed se quantità $\geq s$, non faccio altri ordini.
if less than s , inventory is replenished to S :

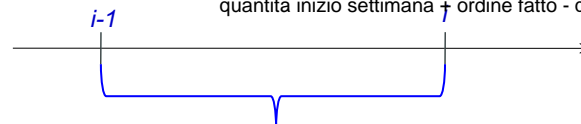
$$O_{i-1} = \begin{cases} 0 & I_{i-1} \geq s \\ S - I_{i-1} & I_{i-1} < s \end{cases}$$

se sto sopra la soglia minima, non faccio ordini.
se sto sotto soglia minima, faccio ordine tale da riportare la quantità a S (il massimo).

- Items are delivered immediately
- At end of i^{th} interval, inventory diminished by d_i

$$I_i = I_{i-1} + O_{i-1} - d_i$$

quantità inizio settimana + ordine fatto - quanto viene chiesto nella settimana



Prof. Vittoria de Nitto Personè

6

6

Time Evolution of Inventory Level

Discrete-event simulation
Trace-driven simulation

Algorithm 3: trace-driven simulation

```

 $l_0 = S;$                                 /* the initial inventory level is S */
 $i = 0;$ 
while ( more demand to process ) {
     $i++;$ 
    if ( $l_{i-1} < s$ )
         $o_{i-1} = S - l_{i-1};$     faccio ordine
    else
         $o_{i-1} = 0;$     else non ordino nulla
    prendo domanda     $d_i = \text{GetDemand}();$     ← Read data from file
    calcolo livello intervallo successivo     $l_i = l_{i-1} + o_{i-1} - d_i;$ 
    }
     $n = i;$     calcolo il numero di settimane che sto simulando
    ultimo ordine per riportarmi allo stato iniziale     $\begin{cases} o_n = S - l_n; \\ l_n = S; \end{cases}$     /* the terminal inventory level is S */

    return  $l_1, l_2, \dots, l_n$  and  $o_1, o_2, \dots, o_n$ ;

```

Prof. Vittoria de Nitto Personè

7

7

Discrete-event simulation
Trace-driven simulation

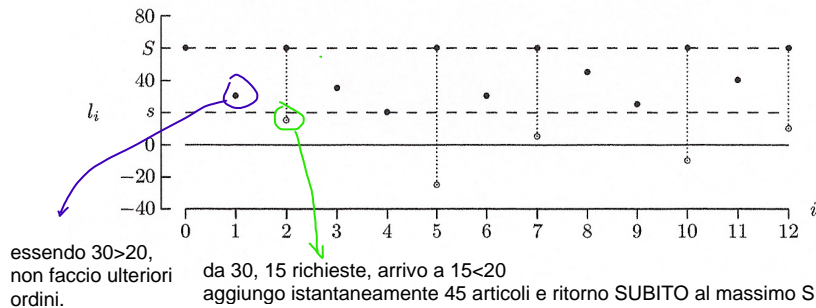
example

quindi massimo numero articoli 60, e ne dovrei avere almeno 20 per non rinnovare ordini.

Let $(s, S)=(20, 60)$ and consider $n=12$ time intervals,
and the following input demands:

i	1	2	3	4	5	6	7	8	9	10	11	12
d_i	30	15	25	15	45	30	25	15	20	35	20	30

$d(i)$ arrivano DURANTE la settimana, non in un singolo tempo t .



Prof. Vittoria de Nitto Personè

8

8

- What statistics to compute?
- *average demand and average order*

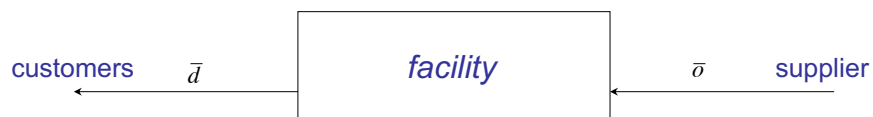
$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i \quad \bar{o} = \frac{1}{n} \sum_{i=1}^n o_i$$

- For our example data:

$$\bar{d} = \bar{o} = 305/12 \approx 25.42 \quad \text{items per time interval}$$

domanda media = ordini medi, perchè parto da S e ritorno in S.

- Average demand and order *must* be equal
- Ending inventory level is S
- Over the simulated period, all demand is satisfied
- Average “flow” of items in equals average “flow” of items out



Flow balance

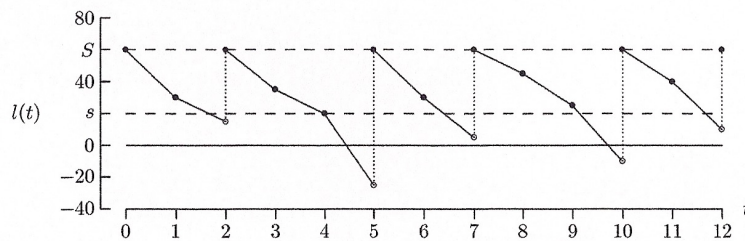
Ho bilanciamento perchè ordini effettuati = domande posti, sto bilanciando flussi di arrivo e di partenza.

Noi dobbiamo calcolare i costi (mantenimento, mancanza merci) in funzione del tempo, perchè una mancanza di materiale di 1 settimana è peggiore di una di 3 settimane. La funzione $l(t)$ deve essere dipendente dal tempo, non basta vederla all'inizio e alla fine.

Discrete-event simulation

- Holding and shortage costs are proportional to time-averaged inventory levels
- Must know inventory level for all t (not only at review times)
- Assume the demand rate is constant between review times

Se nel primo intervallo ho avuto richiesta globale di 30, queste 30 le spalmo in maniera uniforme in questo intervallo.



Prof. Vittoria de Nitto Personè

11

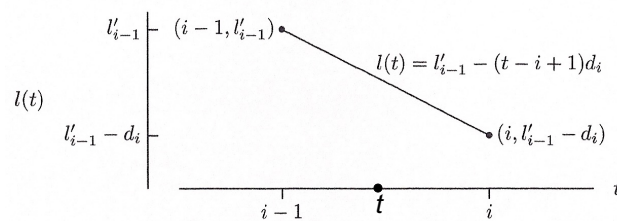
11

Discrete-event simulation

The inventory level at any time t in i^{th} interval is

$$l(t) = l'_{i-1} - (t - i + 1)d_i$$

If demand rate is constant between review times



$l'_{i-1} = l_{i-1} + o_{i-1}$ represents inventory level *after* review
dopo l'incremento dell'ordine ("o" può essere 0 se inventario sopra s).

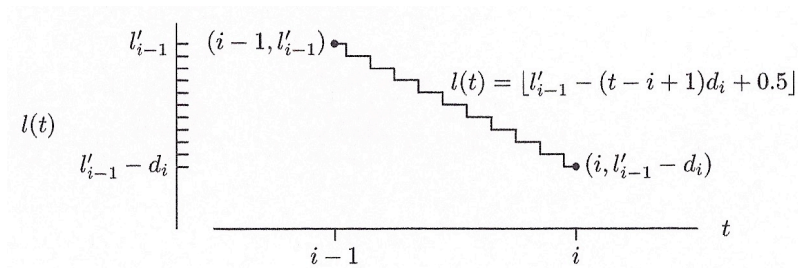
Poichè costante, se voglio vedere al tempo t , prendo come tempo " $t - (i-1)$ " dove " $i-1$ " è l'inizio.

Prof. Vittoria de Nitto Personè

12

12

- Inventory level at any time t is an integer
- $l(t)$ should be rounded to an integer value
- $l(t)$ is a stair-step function



ho 10 scalini, il numero scende di 1 ogni volta, perchè abbiamo detto che l'ordine è di 0 o 1 merci "spalmate". 0.5 serve per avere numero intero, non ci soffermiamo su.

Prof. Vittoria de Nitto Personè

13

13

Time-Averaged Inventory Level

Case 1

$l(t)$ remains non-negative over i^{th} interval

$$\bar{l}_i^+ = \int_{i-1}^i l(t) dt \quad \text{costo "holding"}$$

Case 2

$l(t)$ becomes negative at some time τ

$$\bar{l}_i^+ = \int_{i-1}^{\tau} l(t) dt \quad \bar{l}_i^- = -\int_{\tau}^i l(t) dt \quad \text{costo "shortage"/mancanza}$$

\bar{l}_i^+ is the time-averaged holding level

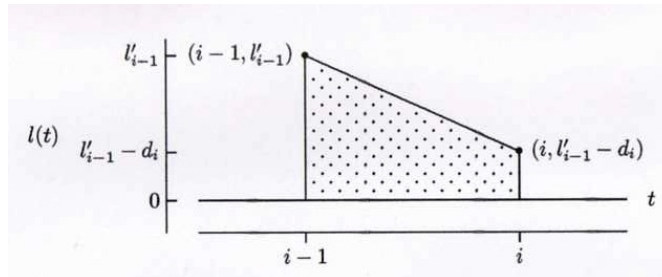
\bar{l}_i^- is the time-averaged shortage level

Prof. Vittoria de Nitto Personè

14

14

Case 1 no back-ordering

No shortage during i^{th} time interval iff $d_i \leq l'_{i-1}$ 

Time-averaged holding level is computed as area of a trapezoid:

$$\bar{l}_i^+ = \int_{i-1}^i l(t) dt = \frac{l'_{i-1} + (l'_{i-1} - d_i)}{2} = l'_{i-1} - \frac{1}{2} d_i$$

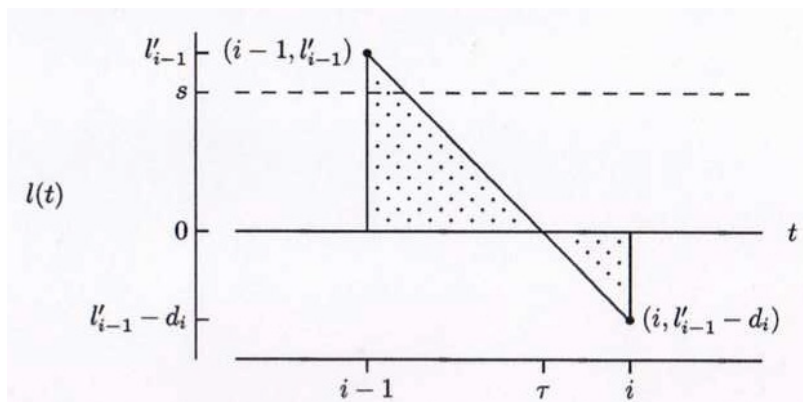
caso 1: non scendo sotto 0, sto sopra, allora mi calcolo l'area, sto nel caso "positivo".
Ho solo costo "holding" no "shortage", posso calcolarmi l'integrale.

Prof. Vittoria de Nitto Personè

15

15

Case 2 back-ordering

Inventory becomes negative iff $d_i > l'_{i-1}$ 

Prof. Vittoria de Nitto Personè

16

16

Qui ho "shortage", vado sotto 0, e continuo ad avere ordini. Quindi devo calcolarmi sia l'integrale di "holding" (come sopra) a cui sottrarre l'integrale dello "shortage".

Case 2 back-ordering (cont.)

- $l(t)$ becomes negative at time $t = \tau = i-1 + (l'_{i-1}/d_i)$
- time-averaged holding and shortage levels for i^{th} interval computed as the areas of triangles

$$\bar{l}_i^+ = \int_{i-1}^{\tau} l(t) dt = \dots = \frac{(l'_{i-1})^2}{2d_i}$$

$$\bar{l}_i^- = -\int_{\tau}^i l(t) dt = \dots = \frac{(d_i - l'_{i-1})^2}{2d_i}$$

Time-Averaged Inventory Level

- *Time-averaged holding level and time-averaged shortage level*

$$\bar{l}^+ = \frac{1}{n} \sum_{i=1}^n \bar{l}_i^+ \quad \bar{l}^- = \frac{1}{n} \sum_{i=1}^n \bar{l}_i^-$$

- *The time-averaged inventory level is*

$$\bar{l} = \frac{1}{n} \int_0^n l(t) dt = \bar{l}^+ - \bar{l}^-$$

qui ovviamente medio su TUTTE le settimane.

Goals
Conceptual model
Specification model
Computational model
Verify
Validate

Discrete-event simulation

- Computes the statistics

$$\bar{d}, \bar{o}, \bar{I}^+, \bar{I}^-$$

and the order frequency

$$\bar{u} = \frac{n. of orders}{n} \quad n^\circ \text{ ordini globali} / n^\circ \text{ settimane.}$$

- Consistency check:
compute \bar{o}, \bar{d} separately, then compare

Prof. Vittoria de Nitto Personè

19

19

sis1.c

```
#include <stdio.h>

#define FILENAME "sis1.dat"
#define MINIMUM 20
#define MAXIMUM 80
#define sqr(x) ((x) * (x))

long GetDemand(FILE *fp)
{
    long d;
    fscanf(fp, "%ld\n", &d);
    return (d);}

```

20

```

int main(void)
{
    FILE *fp;
    long index      = 0;
    long inventory = MAXIMUM;
    long demand;
    long order;
    struct {
        double setup;
        double holding; /* inventory held (+) */
        double shortage; /* inventory short (-) */
        double order;
        double demand;
    } sum = { 0.0, 0.0, 0.0, 0.0, 0.0 };

    fp = fopen(FILENAME, "r");
    if (fp == NULL) { fprintf(stderr, "Cannot open input file
%s\n", FILENAME);
        return (1);
    }
}

```

Prof. Vittoria de Nitto Personè

21

21

```

while (!feof(fp)) {
    index++;
    if (inventory < MINIMUM) {
        order = MAXIMUM - inventory;
        sum.setup++;
        sum.order += order;
    }
    else order = 0;
    inventory += order; /* there is no delivery lag */
    demand = GetDemand(fp);
    sum.demand += demand;
    if (inventory > demand)
        sum.holding += (inventory - 0.5 * demand);
    else {
        sum.holding += sqr(inventory) / (2.0 * demand);
        sum.shortage += sqr(demand - inventory) / (2.0 * demand);
    }
    inventory -= demand;
}

```

Prof. Vittoria de Nitto Personè

22

22

```

if (inventory < MAXIMUM) {
    order = MAXIMUM - inventory;
    sum.setup++;
    sum.order+= order;
    inventory+= order;
}

printf("\nfor %ld time intervals ", index);
printf("with an average demand of %.2f\n", sum.demand / index);
printf("and policy parameters (s, S) = (%d, %d)\n\n", MINIMUM, MAXIMUM);
printf("  average order ..... = %.2f\n", sum.order / index);
printf("  setup frequency ..... = %.2f\n", sum.setup / index);
printf("  average holding level .... = %.2f\n", sum.holding / index);
printf("  average shortage level ... = %.2f\n", sum.shortage / index);

fclose(fp);
return (0);
}

```

Prof. Vittoria de Nitto Personè

23

23

Discrete-event simulation

- trace file sis1.dat contains data for n=100 time intervals
- with (s, S)=(20, 80)

$$\bar{o} = \bar{d} = 29.29 \quad \bar{u} = 0.39 \quad \bar{l}^+ = 42.40 \quad \bar{l}^- = 0.25$$

Prof. Vittoria de Nitto Personè

24

24

Operating Costs

A facility's cost of operation is determined by:

- C_{item} : unit cost of new item
- C_{setup} : fixed cost for placing an order (setup cost)
- C_{hold} : cost to hold one item for one time interval
- C_{short} : cost of being short one item for one time interval

Case study

- Automobile dealership that uses weekly periodic inventory review
- The facility is the showroom and surrounding areas
- The items are new cars
- The supplier is the car manufacturer
- Simple inventory system: one type of car

Case study

- Limited to a maximum of $S=80$ cars
- Inventory reviewed every Monday
- If inventory falls below $s=20$, order cars sufficient to restore to $S=80$
- For now, ignore delivery lag
- Costs:
 - Item cost is $C_{item} = \$8000$ per item
 - Setup cost is $C_{setup} = \$1000$
 - Holding cost is $C_{hold} = \$25$ per week
 - Shortage cost is $C_{short} = \$700$ per week

Prof. Vittoria de Nitto Personè

27

27

Per-Interval Average Operating Costs

- The average operating costs *per time interval* are:
 - item cost: $c_{item} \cdot \bar{D}$
 - setup cost: $c_{setup} \cdot \bar{U}$
 - holding cost: $c_{hold} \cdot \bar{I}^+$
 - shortage cost: $c_{short} \cdot \bar{I}^-$
- The average *total* operating cost *per time interval* is their sum
- For the stats and costs of the hypothetical dealership:
 - item cost $\$8000 \cdot 29.29 = \234.320 per week
 - setup cost $\$1000 \cdot 0.39 = \390 per week
 - holding cost $\$25 \cdot 42.40 = \1.060 per week
 - shortage cost $\$700 \cdot 0.25 = \175 per week

Prof. Vittoria de Nitto Personè

28

28

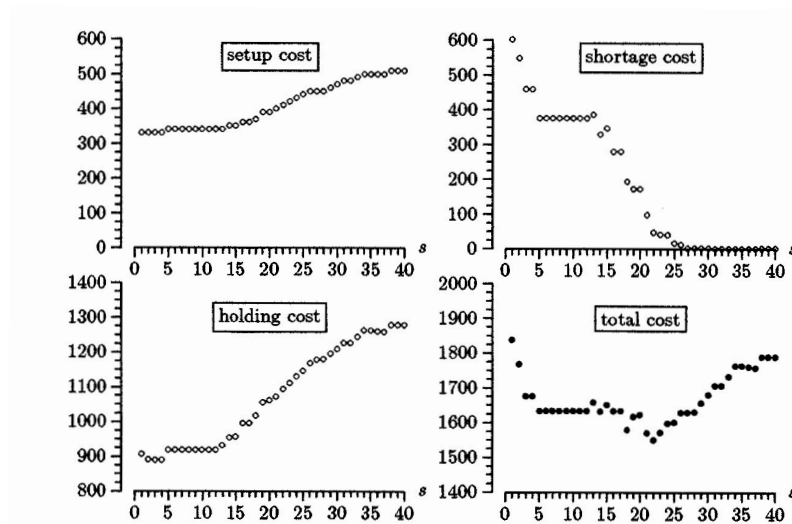
Cost Minimization

- By varying s (and possibly S), an optimal policy can be determined
- Optimal \Leftrightarrow minimum average cost
- Note that $\bar{o} = \bar{d}$ and \bar{d} depends only on the demands. Hence, item cost is independent of (s, S)
- Average dependent cost is:
avg setup cost + avg holding cost + avg shortage cost

7. Experiments design

8. Runs production
9. Output analysis
10. Decisional phase
11. Results documentation

- Let S and the demand sequence be fixed
- If s is systematically increased, we expect:
 - average setup cost and holding cost will increase
 - average shortage cost will decrease
- average dependent cost will have 'U' shape, yielding an optimum
avg setup cost +
avg holding cost +
avg shortage cost
- From results (next slide), minimum cost is \$1550 at $s=22$



Prof. Vittoria de Nitto Personè

31

31

Exercises

- Study program `sis1.c`
- Run with `sis1.dat`; analyze the results
- Generate a data file with the values on p.8
- Run with the new data, verify that the results confirm the expectations
- Ex 1.3.1 and Ex 1.3.8 on p.36 of the textbook

Prof. Vittoria de Nitto Personè

32

32