

# Vademecum del corso A di Calcolo Numerico (laurea triennale in Informatica)

Felice Iavernaro\*

## Sommario

In questa breve nota, ho raccolto un insieme di informazioni riguardanti l'insegnamento di *Calcolo Numerico* (corso A) della laurea triennale di Informatica, di cui sono titolare. Accanto alle specifiche regolamentazioni relative al corso corrente, ho pensato di discutere alcune questioni più generali in merito alla preparazione e alla modalità di svolgimento dell'esame, alla preparazione dei programmi ed altro, con l'auspicio che tutto ciò possa essere di qualche utilità per i miei studenti :-)

## 1 Prologo

*“Qualsiasi sia il comportamento di un insegnante, esisteranno sempre degli studenti scontenti del suo operato.”*

La didattica e più in generale il confronto con gli studenti sono questioni che da sempre hanno destato il mio interesse e non è sorprendente il fatto che un docente ben predisposto, abbia spesso più da imparare piuttosto che da riversare durante la sua attività professionale. Il *Postulato di non esistenza del prof. perfetto* è anche sinonimo della distanza esistente tra due mondi che spesso parlano linguaggi differenti, poiché differenti sono i modi di percepire, elaborare, sintetizzare la realtà ed il vissuto quotidiani.

Per questioni di trasparenza e correttezza, ho pensato di puntualizzare i miei obiettivi didattici ed i principi su cui si basano i miei personali criteri

---

\*Dipartimento di Matematica, Università di Bari, Via Orabona 4, I-70125 Bari (Italy), felice.iavernaro@uniba.it, <http://www.dm.uniba.it/~iavernaro>

di valutazione (§2). Di seguito (§3) vi è una breve introduzione al *Calcolo Numerico*, ai contenuti ed all'organizzazione del corso. Infine (§4) trovate i dettagli sulla preparazione e le modalità di svolgimento dell'esame.

## 2 Obiettivi didattici e criteri di valutazione

**After explaining to a student through various lessons and examples that:**

$$\lim_{x \rightarrow 8} \frac{1}{x-8} = \infty$$

**I tried to check if she really understood that, so I gave her a different example. This was the result:**

$$\lim_{x \rightarrow 5} \frac{1}{x-5} = \infty$$

L'operato di un insegnante non deve avere necessariamente come obiettivo quello di massimizzare il livello di conoscenza dello studente medio<sup>1</sup>. Ciò che intendo dire, magari sottoponendomi per questo a critiche più o meno forti, è che non è del tutto corretto tarare la propria attività, magari rendendo i contenuti più semplici, in modo che quanti più studenti possano seguire più

agevolmente le spiegazioni o più facilmente preparare e sostenere l'esame. Sia ben inteso che in questi anni ho via via semplificato alcune questioni piuttosto teoriche preferendo dirottare l'attività dello studente maggiormente sulla programmazione. Ciononostante ho cercato di diversificare l'impalcatura dei contenuti del corso di modo che anche gli studenti più scaltri e volenterosi non venissero svantaggiati da questa semplificazione. L'obiettivo finale è determinare un onesto compromesso che abbia come prerogativa:

- la possibilità di superamento dell'esame da parte di coloro che abbiano raggiunto una preparazione sufficiente;
- la possibilità di differenziare i diversi livelli di preparazione.

Come motivazione, vi sottopongo la seguente riflessione:

un *compito difficile* farebbe molte vittime e soprattutto non sarebbe corretto assegnarlo poiché, per definizione, l'appellativo "difficile" significherebbe che, nei contenuti, andrebbe troppo al di là delle questioni affrontate durante le lezioni, penalizzando così lo studente che fa il proprio dovere di raggiungere una preparazione sufficiente;

un *compito facile* però penalizzerebbe lo studente più scaltro, visto che non fornirebbe una traccia dei diversi livelli di preparazione: tutti coloro che studiano a sufficienza avrebbero voti alti; nessun ulteriore criterio di differenziazione;

---

<sup>1</sup>L'immagine accanto e quelle seguenti sono prese dal web.

Una analogo parallelismo può riferirsi al livello di difficoltà dei contenuti delle lezioni.

Fissati questi principi, ho cercato di offrire la massima flessibilità in materia di “modalità di espletamento dell’esame”<sup>2</sup>, in modo da agevolare il più possibile il lavoro e l’organizzazione dello studente, compatibilmente con le direttive fissate dal Consiglio del Corso di Studi.

### 3 Il corso: motivazioni, contenuti ed organizzazione.

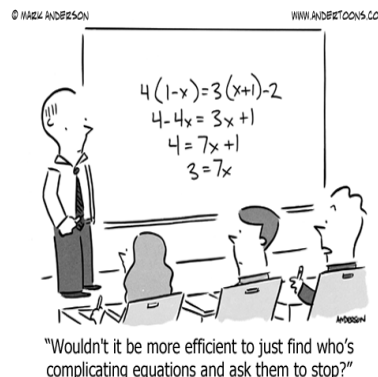
Il corso propone una panoramica sulle tecniche numeriche (algoritmi) per la risoluzione di problemi nell’ambito dell’algebra e dell’analisi elementare. Esso prevede anche un’ introduzione all’ algebra lineare.

#### 3.1 Perché il Calcolo Numerico?

Le principali motivazioni per un approccio numerico ai problemi della matematica sono:

- il fatto che la maggior parte dei problemi matematici (anche relativamente semplici) non sono risolvibili analiticamente<sup>3</sup>, ovvero non esistono tecniche che permettano di esprimerne la soluzione come combinazione finita di funzioni elementari note (soluzione non esprimibile in forma chiusa);<sup>4</sup>

- gran parte dei problemi matematici che provengono dalle diverse branche applicative delle discipline scientifiche e dal mondo dell’industria hanno *grandi dimensioni*, cioè coinvolgono un numero elevato di incognite. Anche qualora detti problemi fossero risolvibili elementarmente,



<sup>2</sup>Si veda il §4.

<sup>3</sup>o elementarmente.

<sup>4</sup>Alcuni esempi:

- risolvere l’equazione  $x = \cos(x)$ ;
- calcolare l’integrale  $\int_0^1 e^{-x^2} dx$ .

il tempo richiesto per determinarne la soluzione sarebbe troppo elevato.

- in molte applicazioni è frequente la necessità di risolvere problemi in tempo reale (ovvero in tempi ridottissimi).

La maggior parte dei problemi<sup>5</sup>, soprattutto quelli che provengono dalle applicazioni *real-life*, ricadono in una o più delle classi sopra elencate. Il *Calcolo Numerico*, e più in generale l'*Analisi Numerica*, studia la possibilità di poter approssimare le soluzioni di detti problemi, in maniera *efficiente* sul calcolatore. La potenza di calcolo fornita dai moderni calcolatori<sup>6</sup> permette, in effetti, di risolvere problemi di grandi dimensioni in tempi relativamente ridotti.

Tuttavia, a fronte di una potenza di calcolo molto elevata, il calcolatore, disponendo di memoria finita, implementa un'aritmetica differente da quella reale. L'*aritmetica di macchina* ha, per natura, due svantaggi:

- (a) può rappresentare solo un numero finito di numeri reali, i cosiddetti *numeri di macchina*; tutti i restanti numeri, se non troppo piccoli nè troppo grandi, verranno appropriatamente approssimati mediante i numeri di macchina. L'errore di rappresentazione che ne deriva, dipende dalla macchina e dal software utilizzati: ad esempio in Python e in Matlab è circa  $2 \cdot 10^{-16}$ ;
- (b) le quattro operazioni elementari,  $+$ ,  $-$ ,  $*$ ,  $/$ , non sono (in generale) eseguite correttamente, anche se l'errore che esse introducono è dello stesso ordine dell'errore di rappresentazione.

Il calcolatore può essere quindi inteso come uno strumento di calcolo dotato di una sua propria precisione che non può essere superata. Come conseguenza, in alcuni casi il risultato che esso ci restituisce, in seguito all'esecuzione di un programma che implementa un dato algoritmo, potrebbe essere del tutto erroneo, pur essendo il problema e l'algoritmo risolutivo impostati correttamente. Ciò accade in una delle seguenti situazioni:

- (1) quando il problema matematico è *mal condizionato*, cioè quando le soluzioni del problema (dati di input) dipendono in modo sensibile dai dati iniziali (dati di output): ciò significa che a piccole perturbazioni dei dati di input possono corrispondere grosse variazioni sulla soluzione. In effetti il calcolatore introduce piccole perturbazioni dei dati di input proprio quan-

---

<sup>5</sup>praticamente tutti quelli che non si trovano nei testi scolastici.

<sup>6</sup>gli attuali processori presenti in commercio hanno una potenza di calcolo superiore a 1Gflops/s ( $10^9$  operazioni elementari al secondo).

do questi vengono rappresentati mediante numeri di macchina (appunto gli errori di rappresentazione).

(2) quando l'algoritmo costruito per la risoluzione del problema risulta *instabile*: ciò si verifica quando la sequenza delle singole operazioni elementari che definiscono l'algoritmo è tale da amplificare notevolmente i piccoli errori dovuti al fatto che dette operazioni non sono effettuate correttamente.



Per uno stesso problema possono esistere più tecniche risolutive le quali definiscono altrettanti algoritmi suscettibili di implementazione al calcolatore. Un confronto tra due o più metodi dovrà, in generale, tener conto dei seguenti fattori:

- *classe di problemi a cui il metodo è applicabile*: alcuni metodi possono, in effetti, richiedere ipotesi di lavoro più restrittive rispetto ad altri;
- *costo computazionale*: indica la quantità di operazioni che l'algoritmo richiede mediamente per fornire il risultato; esso è chiaramente proporzionale al tempo di esecuzione ma, al contrario di quest'ultimo, è indipendente dalla macchina utilizzata;
- *stabilità*: un algoritmo non sufficientemente stabile può generare errori significativi che ne rendono l'impiego inaccettabile.

## 3.2 I contenuti

Approssimativamente, il programma coprirà i seguenti argomenti:<sup>7</sup>

1. *Analisi dell'errore.*

Rappresentazione dei numeri. IEEE singola e doppia precisione. Troncamento e Arrotondamento. Precisione di macchina. Errore assoluto e relativo. Operazioni con i numeri di macchina. Cancellazione di cifre significative. Condizionamento di un problema. Stabilità degli algoritmi. Propagazione degli errori.

---

<sup>7</sup>ai fini della preparazione dell'esame, si faccia riferimento al programma ufficiale che sarà pubblicato alla fine del corso.

2. *Calcolo degli zeri di funzione.*

Metodo delle bisezioni. Convergenza. Criteri di arresto e stime dell'errore. Ordine di convergenza. Iterazione funzionale. Teorema di contrazione e corollari. Il metodo di Newton. Metodo della direzione costante. Metodo delle secanti. Criteri di stop.

3. *Elementi di Algebra Lineare.*

Definizione di spazi vettoriali. Operazioni tra vettori. Matrici. Operazioni tra matrici. Determinante e rango di una matrice. Inversa di una matrice. Matrici particolari. Operazioni sulle matrici. Sistemi lineari. Teorema di Rouché-Capelli. Norme su vettori e su matrici.

4. *Metodi numerici per l'algebra lineare.*

Algoritmi per la soluzione di sistemi lineari. Sistemi triangolari inferiori e superiori. Matrici di permutazione e proprietà. Algoritmo di eliminazione di Gauss. Problematiche di stabilità. Teorema di esistenza della fattorizzazione LU con pivot. Riduzione a scalini di una matrice. Calcolo del rango di una matrice. Studio del condizionamento di un sistema lineare.

5. *Interpolazione e approssimazione.*

Base delle potenze. Interpolazione con la base di Lagrange. Errore nell'interpolazione polinomiale. Cenni sul fenomeno di Runge. Studio del condizionamento. Fitting di dati: metodo dei minimi quadrati.

6. *Programmazione.* Il linguaggio Python e le sue applicazioni al Calcolo Numerico. Esempi di codici relativi ad alcuni algoritmi teorici.

I testi adottati sono <sup>8</sup>

L. Brugnano, C. Magherini, A. Sestini, *Calcolo Numerico*, Master Università & Professioni, 2005.

D. Bini, M. Capovani, O. Menchi, *Metodi numerici per l'algebra lineare*, Zanichelli, Bologna, 1988.

R. Bevilacqua, D. Bini, M. Capovani, O. Menchi, *Metodi numerici*, Zanichelli, Bologna, 1992.

Si ricorda che l'insegnamento di Analisi Matematica è propedeutico all'insegnamento di Calcolo Numerico.

---

<sup>8</sup>il primo testo in elenco sarà quello di riferimento. Per ulteriori dettagli si rimanda al link: <http://web.math.unifi.it/users/brugnano/Libri/MISL/index.html>

### 3.3 Ricevimento

L'orario ufficiale per il ricevimento è il lunedì dalle 14:00 alle 16:00. Tuttavia lo studente può chiedere spiegazioni alla fine della lezione (o durante la pausa, quando prevista), oppure direttamente per e-mail (specie se si tratta di dubbi sui programmi). Ulteriormente, è possibile concordare il ricevimento per e-mail (felice.iavernaro@uniba.it) o per telefono (080 5442703).

## 4 L'esame: manuale di istruzioni

In questa sezione ho raccolto gran parte delle informazioni che riguardano le modalità di svolgimento dell'esame.

### 4.1 Appelli

Il giorno dell'appello, all'ora stabilita, lo studente che intende sostenere l'esame, può presentarsi nel mio studio<sup>9</sup>. In generale, in quella occasione, viene redatto un calendario: le date, concordate tenendo conto delle esigenze di tutti, possono anche ricoprire l'intero mese a cui l'appello fa riferimento. Eventualmente è anche possibile sostenere l'esame lo stesso giorno dell'appello. L'esame è solo orale, tuttavia è anche previsto un esercizio in Python (per coloro che dovranno discutere i programmi). Lo studente che sia impossibilitato a presentarsi il giorno dell'appello, può contattarmi qualche giorno prima per e-mail o telefonicamente per indicare una preferenza circa la data: provvederò ad aggiungere il suo nominativo alla lista relativa a uno dei turni stabiliti.

### 4.2 Preparazione dei programmi in Scilab (o Matlab)

L'esame prevede, oltre la parte di teoria, la preparazione di alcuni programmi in Python.<sup>10</sup> Non vi è un numero fisso di programmi da presentare anche se si richiede di cercare di ricoprire tutti gli argomenti trattati durante le lezioni. Cura dello studente dovrà anche essere quella di riprodurre e presentare le esecuzioni relative ai programmi sviluppati. Ai fini della valutazione, si terrà conto di quanto segue:

---

<sup>9</sup>stanza n. 2, piano IV, del Dipartimento di Matematica, tel: 080 5442703.

<sup>10</sup>Il software per il calcolo scientifico *Python* è scaricabile gratuitamente dalla rete. Se ne consiglia la distribuzione "Anaconda" al link <https://www.continuum.io>

1. il livello di conoscenza tecnica acquisita. L'esercizio proposto in sede d'esame e le domande tecniche sui programmi presentati hanno in effetti proprio questo scopo.
2. Gli esempi riportati ed i confronti effettuati al fine di mettere in evidenza *pregi e difetti* di ciascun metodo (costo computazionale, stabilità, classe di problemi a cui il metodo può essere applicato, ecc.).
3. Infine, a patto che la valutazione al punto 1 abbia rivelato una corrispondenza oggettiva tra conoscenza acquisita e stato qualitativo dei programmi, inciderà anche il numero e la qualità stessa dei programmi.

### 4.3 Esoneri

Per l'anno corrente sono previste due prove di verifica che, per quanto riguarda Calcolo Numerico avranno validità esonerativa, ovvero potranno esonerare lo studente dall'esame finale

Ogni prova scritta prevede la risoluzione di un certo numero di tracce che richiedono la discussione di parti teoriche, la risoluzione di esercizi, la scrittura di programmi in Python<sup>11</sup>. Generalmente ad ogni traccia viene assegnato un dato punteggio, per un totale nominale di circa 35 punti<sup>12</sup>. Le date degli esoneri sono concordate insieme agli studenti durante ed alla fine del corso. Le tracce degli anni scorsi sono disponibili in rete. Inoltre:

1. Allo studente è lasciata la libertà di partecipare ad entrambi oppure ad uno solo degli esoneri: in quest'ultimo caso, previo superamento, lo studente potrà sostenere l'esame orale discutendo solo la parte mancante ed i programmi relativi.
2. Ciascun esonero è da ritenersi superato se il punteggio non è inferiore a 18/30, oppure se non è inferiore a 15/30 e la media complessiva tra i due esoneri è almeno 18/30. Se la media complessiva è invece inferiore a 18/30, è necessario recuperare, in sede d'esame orale, la prova risultata insufficiente.
3. Coloro che hanno conseguito una votazione inferiore a 15/30 in una delle due prove d'esonero, o non ne hanno sostenuta una delle due, dovranno recuperare in sede d'esame orale la relativa parte.

---

<sup>11</sup>Gli esercizi sul Python possono anche riguardare la scrittura di un algoritmo studiato teoricamente a lezione.

<sup>12</sup>Tuttavia, poiché la votazione è espressa in trentesimi, se il punteggio supera 30, verrà segnalato con 30+.



4. Il superamento di entrambe le prove esonera anche dalla discussione dei programmi in Python. Tuttavia, lo studente che desidera migliorare il proprio voto può, a sua discrezione, preparare e discutere ugualmente i programmi. In tal caso gli verrà chiesto di svolgere preventivamente un esercizio di programmazione in Python.
5. Infine, lo studente che abbia superato entrambe le prove d'esonero, avrà anche la possibilità di rinunciare ad una delle due parti ed a discuterla in sede d'esame. In tal caso egli resta comunque esonerato dalla parte sullo Python (si applica anche qui quanto specificato al punto precedente).

## **5 Conclusioni**

Augurando a tutti un proficuo lavoro, confermo la mia piena disponibilità a prestare attenzione ai vostri eventuali suggerimenti e proposte al fine di migliorare la gestione complessiva di questo insegnamento.