Alma Mater Studiorum · Università di Bologna

facoltà di scienze matematiche, fisiche e naturali

Corso di Laurea in Matematica

FUNZIONI POLINOMIALI NEGLI ANELLI $\mathbb{Z}_{p^n}$

Tesi di Laurea in Algebra

Relatore:
Chiar.mo Prof.
Libero Verardi

Presentata da:
Giulia Tosetti

III Sessione
Anno Accademico 2013/2014

Alla Dariuccia

"Ci sono due posti al mondo dove possiamo vivere felici: a casa e a Parigi"
(Ernest Hemingway)

"Non preoccuparti delle difficoltà che incontri in matematica, ti posso assicurare che le mie sono ancora più grosse."
(Albert Einstein)

"Così si gioca solo in paradiso!"
(Fulvio Bernardini)

Indice

Iconografia: figure

Iconografia: tabelle

Introduzione

L'obiettivo primario che ci siamo posti durante la redazione di questa tesi è quello di effettuare una piccola ricerca sulle funzioni polinomiali negli anelli $\mathbb{Z}_{p^n}$, tentando di rispondere a questa domanda: *quante sono le funzioni polinomiali distinte da $\mathbb{Z}_{p^n}$ in sé?*
Lo spunto per iniziare quello che si è rivelato, per certi versi, un vero e proprio esperimento scientifico, ci è stato fornito dalla tesi di Laurea della dottoressa Marilena Accogli che, in anni passati, aveva introdotto questo argomento soffermandosi in particolare sulle funzioni polinomiali negli anelli $\mathbb{Z}_m$ con m numero primo o prodotto di due numeri coprimi.
La ricerca che presentiamo non affronta però il tema solamente da un punto di vista algebrico: una delle mire che ci siamo posti è stata proprio quella di affrontare un argomento prettamente tecnico in maniera interdisciplinare, affiancando al rassicurante ed elegante ordine dell'algebra, la programmazione informatica e l'analisi statistica.
Nel primo capitolo abbiamo esaminato approfonditamente il caso particolare delle funzioni polinomiali definite dal campo $\mathbb{Z}_p$ in sé, dimostrando che esse si possono tutte esprimere come un polinomio. Per farlo, abbiamo costruito l'applicazione $\rho : \mathbb{Z}_p[x] \longrightarrow \mathbb{Z}_p^{\mathbb{Z}_p}$, che associa al polinomio f la corrispondente funzione polinomiale. Abbiamo poi dimostrato che ρ è un epimorfismo e dunque, attraverso il teorema fondamentale di omomorfismo per anelli, abbiamo provato che il numero di funzioni polinomiali in $\mathbb{Z}_p$ è esattamente p^p.

Lo stesso non si può dire per gli interi modulo p^n, con $n > 1$: in quei casi le funzioni polinomiali sono un sottoinsieme proprio dell'anello di tutte le applicazioni da $\mathbb{Z}_{p^n}$ in sé.

Nel secondo capitolo dunque abbiamo calcolato il numero di funzioni polinomiali negli anelli $\mathbb{Z}_9$ e $\mathbb{Z}_{25}$ seguendo il procedimento qui descritto:

- abbiamo determinato i generatori del nucleo del morfismo ρ già descritto nel caso del campo $\mathbb{Z}_p$;
- abbiamo utilizzato le congruenze note per calcolare quali valori possono assumere i coefficienti dei monomi di ogni dato grado;
- abbiamo ottenuto il numero esatto p^k di funzioni polinomiali nell'anello $\mathbb{Z}_{p^2}$ per $p = 3$ e $p = 5$.

Il procedimento che abbiamo utilizzato è applicabile a qualsiasi n-esima potenza di un numero primo p. Già dai conti effettuati, appare però chiaro che i calcoli diventano rapidamente molto complessi. Per questo motivo ci siamo chiesti se è possibile congetturare una regola generale che consenta di calcolare direttamente il numero di funzioni polinomiali in un anello del tipo $\mathbb{Z}_{p^n}$. Così, nel terzo capitolo, abbiamo messo in cantiere un vero e proprio esperimento statistico. In primo luogo abbiamo osservato che, nei casi noti, valeva la seguente proprietà: *se p^k è il numero delle funzioni polinomiali nell'anello $\mathbb{Z}_{p^n}$ e p^h è il numero delle funzioni polinomiali nell'anello $\mathbb{Z}_{p^{n-1}}$, la differenza $k - h$ tra i due esponenti è pari al minimo intero m tale che p^n divida $m!$.*
Abbiamo così congetturato che questa proprietà sia vera per qualsiasi scelta di $n \in \mathbb{N}$ e di p numero primo. Sulla base di tale congettura abbiamo calcolato, con l'ausilio di un foglio di calcolo elettronico e del linguaggio di programmazione ad alto livello Python®, il numero di funzioni polinomiali negli anelli $\mathbb{Z}_{p^n}$, per $n = 1, ...1000$ e per $p = 2$ e $p = 3$. In entrambi i casi i valori trovati per gli esponenti di p si dispongono sul piano cartesiano seguendo un andamento che ricorda una parabola con la concavità rivolta verso l'alto e l'asse parallelo all'asse delle ordinate. Abbiamo quindi calcolato i coefficienti della parabola interpolante attraverso il metodo dei minimi quadrati e verificato l'attendibilità dei risultati ottenuti attraverso il test di Pearson. Il risultato è stato soddisfacente: in particolare abbiamo osservato che vi è una regolarità interessante tra i valori delle aperture delle parabole di interpolazione al variare di p.

Naturalmente quanto presentato non può essere esaustivo: si tratta, come detto in apertura, di un esperimento di tipo statistico, supportato da alcune considerazioni rigorose di tipo algebrico. Interessante sarebbe, in futuro, riuscire a dimostrare in maniera rigorosa le congetture da noi elaborate.

Abbiamo scelto di concludere la relazione con alcune appendici che, se necessario, possono chiarire eventuali dubbi e che costituiscono un prerequisito essenziale alla lettura.
Nella prima presentiamo il piccolo teorema di Fermat, insieme ad un breve cenno storico sul matematico che lo ideò.
Nella seconda ricordiamo una serie di nozioni di statistica descrittiva, in particolare il concetto di interpolazione e il test di Pearson.
Infine, dedichiamo un capitolo ad una breve panoramica operativa sulla programmazione in Python®.

Capitolo 1

Funzioni polinomiali in $\mathbb{Z}_p$

Abbiamo scelto di aprire la nostra ricerca studiando le funzioni polinomiali definite sul campo $\mathbb{Z}_p$. Si tratta della situazione particolare che si ottiene per $n = 1$ e che apre la strada alla trattazione del caso più generale sul quale è incentrata questa tesi.
Abbiamo voluto presentare in prima istanza l'anello degli interi modulo un numero primo, mostrando che esso è un campo finito e descrivendo i suoi elementi. In seguito abbiamo analizzato le funzioni definite su di esso e, in particolare, quelle di tipo polinomiale.

1.1 $\mathbb{Z}_p$ è un campo

$\mathbb{Z}_m$, con m intero positivo, in generale potrebbe avere zero divisori diversi da $[0]_m$ e quindi non essere né un dominio di integrità, né tantomeno un campo.

Esempio 1. In $\mathbb{Z}_6$, per esempio, $[2]_6$ e $[3]_6$ sono elementi diversi dalla classe di zero, ma il loro prodotto si annulla:

$$[2]_6 \cdot [3]_6 = [6]_6 = [0]_6$$

Esempio 2. Consideriamo l'anello $\mathbb{Z}_4$. L'elemento $[2]_4$ non è nullo, ma:

$$[2]_4 \cdot [2]_4 = [4]_4 = [0]_4$$

Gli zero divisori di $\mathbb{Z}_m$ sono gli elementi del tipo $[a]_m$, con $a \in \mathbb{Z}$ e $\text{MCD}(a, m) > 1$. Se m è un numero primo, $\forall a \in \mathbb{Z}$ t.c. $0 \leq a < m$ si ha però $\text{MCD}(a, m) = 1$. Questo prova che in $\mathbb{Z}_p$ con p primo, non esistono zero divisori non nulli.
Ma negli interi modulo m, se un elemento non è uno zero divisore, allora è invertibile.
Possiamo riassumere quanto detto nel seguente risultato, la cui dimostrazione sfrutta il piccolo teorema di Fermat.[1]

Teorema 1.1.1. *$\mathbb{Z}_p$, dove p è un numero primo, è un campo.*

Dimostrazione. Consideriamo l'elemento $[a]_p$, dove a è un intero tale che $0 < a < p$. Vogliamo dimostrare che questa classe di congruenza modulo p, non nulla e scelta arbitrariamente, ammette inverso moltiplicativo in $\mathbb{Z}_p$. Se resterà provato questo, allora avremo dimostrato che tutti gli elementi non nulli di $\mathbb{Z}_p$ sono invertibili e, di conseguenza, $\mathbb{Z}_p$ è un campo.
Per il piccolo teorema di Fermat, $[a]_p^{-1} = [a]_p^{p-2}$, poiché $a \equiv a^p \bmod p$. Osserviamo che:

$$[a]_p^{-1} \cdot [a]_p = [a]_p^{p-2} \cdot [a]_p = [a]_p^{p-1} = [1]_p.$$

Questo dimostra che $[a]_p^{-1}$ è proprio l'inverso moltiplicativo di $[a]_p$ e conclude la dimostrazione. □

$\mathbb{Z}_p$ è dunque un campo finito, anche se forse sarebbe più corretto definirlo "il" campo finito per eccellenza. Infatti sappiamo che ogni campo finito di caratteristica p contiene un sottocampo isomorfo a $\mathbb{Z}_p$.
Concludiamo il paragrafo con un teorema (del quale omettiamo la dimostrazione) che caratterizza i campi finiti.

Teorema 1.1.2. *Sia p un numero primo e sia q una potenza di p con esponente maggiore o uguale a 1. Allora:*

1. *esiste un campo F di ordine q;*

2. *due campi di ordine q sono isomorfi.*

[1] Per l'enunciato e la dimostrazione del piccolo teorema di Fermat si rimanda all'Appendice A.

1.2 Funzioni polinomiali nel campo $\mathbb{Z}_p$

Addentriamoci ora nell'argomento che più ci preme affrontare: le funzioni polinomiali, ossia quelle funzioni che si possono esprimere come polinomi.
L'abitudine a lavorare con i numeri reali porta uno studente medio delle scuole superiori a confondere o, se preferiamo, ad assimilare un polinomio alla corrispondente funzione polinomiale. L'errore nasce dal fatto che, se si lavora in $\mathbb{R}$, far corrispondere questi due oggetti diversi non ha gravi controindicazioni. Quando però l'ambiente nel quale si opera diventa l'aritmetica modulare, allora i problemi che possono sorgere sono di tutt'altra natura.

Esempio 3. Consideriamo il campo $\mathbb{Z}_3$ con le usuali operazioni di somma e prodotto. Il polinomio

$$f : x(x - [1]_3)(x - [2]_3) \in \mathbb{Z}_3[x]$$

è evidentemente diverso dal polinomio nullo. Eppure, la corrispondente funzione polinomiale è la funzione nulla: qualsiasi elemento di $\mathbb{Z}_3$ si vada a sostituire all'incognita x nel polinomio f, si otterrà sempre come risultato $[0]_3$.

Dall'esempio appena descritto è evidente che, nei campi finiti, polinomi e funzioni polinomiali non sono in corrispondenza biunivoca. Quindi, si può affermare che è sempre possibile associare ad un polinomio la corrispondente funzione polinomiale, ma l'applicazione che li lega non è, in generale, biiettiva.
Cerchiamo di approfondire questo ragionamento e, per farlo, diamo alcune definizioni.

Definizione 1.1. Se X è un insieme qualunque, con un numero finito $n \geq 2$ di elementi, indichiamo con X^X l'insieme di tutte le funzioni da X in sé.

Su X^X si può porre la struttura di monoide, con l'operazione di composizione e l'identità come elemento neutro. Infatti, la composizione è un'operazione binaria che gode della proprietà associativa. Il gruppo delle unità di X^X è il gruppo simmetrico S_X.
Se però come insieme X consideriamo l'anello $\mathbb{Z}_p$ con le usuali operazioni di somma e prodotto tra classi di congruenza, l'insieme X^X diventa l'anello $(\mathbb{Z}_p^{\mathbb{Z}_p}, +, \cdot)$, con la funzione costante $1 : a \longrightarrow [1]_p \ \forall a \in \mathbb{Z}_p$ come elemento neutro. Tale anello è commutativo e ha p^p elementi.

A questo punto possiamo costruire l'applicazione ρ che associa ad ogni polinomio a coefficienti in $\mathbb{Z}_p$ la corrispondente funzione polinomiale:

$$\rho : \mathbb{Z}_p[x] \longrightarrow \mathbb{Z}_p^{\mathbb{Z}_p} \tag{1.2.1}$$

dove

$$\rho(\sum_{i=1}^{k} a_i \cdot x^i) = f(x) = \sum_{i=1}^{k} a_i \cdot x^i \tag{1.2.2}$$

L'obiettivo che ci prefiggiamo in questa fase è quello di studiare l'applicazione ρ per classificare le funzioni polinomiali in $\mathbb{Z}_p$.
Consideriamo il morfismo di valutazione $\sigma_u : \mathbb{Z}_p[x] \longrightarrow \mathbb{Z}_p$. Fissando f e facendo variare u, descriviamo una funzione $u \longrightarrow \sigma_u(f)$, che indichiamo con $\widetilde{f}$ e che altro non è che la **funzione polinomiale** associata al polinomio f.
A questo punto abbiamo tutti gli strumenti necessari per dimostrare la prima e principale caratteristica dell'applicazione ρ come definita in (1.2.2).

Proposizione 1.2.1. *ρ è un morfismo di anelli.*

Dimostrazione. [2] Siano f e g due polinomi qualsiasi a coefficienti in $\mathbb{Z}_p$. Vogliamo dimostrare che:
$\rho(f + g) = \rho(f) + \rho(g)$;
$\rho(f \cdot g) = \rho(f) \cdot \rho(g)$;
$\rho(1) = 1$ (funzione costante).
Fissiamo arbitrariamente un elemento $u \in \mathbb{Z}_p$ e consideriamo il morfismo di valutazione relativo ad u:

$$\sigma_u : \mathbb{Z}_p[x] \longrightarrow \mathbb{Z}_p$$

dove $\sigma_u(f) = f(u)$.

[2] Nella dimostrazione che abbiamo scelto di proporre, non si fa uso delle proprietà tipiche degli interi modulo p, con p primo. Quindi, la Proposizione (1.2.1) e la relativa dimostrazione sono valide per un generico omomorfismo $\rho' : A[x] \longrightarrow A^A$, con A anello commutativo.

Siccome σ_u è un omomorfismo di anelli, allora $\forall u \in \mathbb{Z}_p$ valgono le seguenti uguaglianze:
$\sigma_u(f+g) = \sigma_u(f) + \sigma_u(g)$;
$\sigma_u(f \cdot g) = \sigma_u(f) \cdot \sigma_u(g)$;
$\sigma_u(1) = [1]_p$.
Poiché ciò è valido per ogni scelta di $u \in \mathbb{Z}_p$, allora si può concludere che:
$\widetilde{f+g} = \widetilde{f} + \widetilde{g}$;
$\widetilde{f \cdot g} = \widetilde{f} \cdot \widetilde{g}$;
$\widetilde{1} : u \longrightarrow [1]_p$.
Grazie a queste osservazioni, si riesce infine a provare che anche per ρ valgono le proprietà caratteristiche degli omomorfismi di anelli, concludendo così la dimostrazione:
$\rho(f+g) = \widetilde{f+g} = \widetilde{f} + \widetilde{g} = \rho(f) + \rho(g) \qquad \forall f, g \in \mathbb{Z}_p[x]$;
$\rho(f \cdot g) = \widetilde{f \cdot g} = \widetilde{f} \cdot \widetilde{g} = \rho(f) \cdot \rho(g) \qquad \forall f, g \in \mathbb{Z}_p[x]$.
$\rho(1) = 1$. □

1.3 Quante sono le funzioni polinomiali distinte?

Apriamo questo paragrafo ponendoci una domanda: *quante sono le funzioni polinomiali distinte da $\mathbb{Z}_p$ in sé?*
Innanzitutto teniamo presente che, in generale, non tutte le funzioni sono polinomiali. Infatti, non occorre addentrarsi in astrusi anelli dalle forme bizzarre per trovare esempi di funzioni che con i polinomi non hanno nulla a che fare.

Esempio 4. Consideriamo il campo $\mathbb{R}$ dei numeri reali. Le funzioni polinomiali in $\mathbb{R}$ sono della forma

$$f(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n \tag{1.3.1}$$

dove $n \in \mathbb{N}$, $n < \infty$, $a_i \in \mathbb{R} \ \ \forall i = 1, ...n$.
Esempi "celebri" di funzioni non polinomiali definite su $\mathbb{R}$ a valori reali sono le funzioni trascendenti, come $f(x) = sen(x)$, $f(x) = \mathrm{e}^x$. Infatti, ad esempio, $f(x) = sen(x)$ ha infinite radici, mentre un polinomio ne ha un numero finito, al più pari al suo grado.

Ciò che ci prefiggiamo di scoprire è, nel caso di $\mathbb{Z}_p$, "quante sono" le funzioni polinomiali, rispetto alla totalità delle applicazioni in $\mathbb{Z}_p^{\mathbb{Z}_p}$. Per farlo studiamo l'immagine del morfismo ρ definito in (1.2.2). $\mathrm{Im}(\rho) = \rho(\mathbb{Z}_p[x])$ è un sottoanello di $\mathbb{Z}_p^{\mathbb{Z}_p}$: lo denotiamo

con $PF_p[x]$ e lo chiamiamo **anello delle funzioni polinomiali** su $\mathbb{Z}_p$.
Quello che vogliamo dimostrare è che, nel caso particolare in cui si stia lavorando sul campo finito $\mathbb{Z}_p$, l'omomorfismo ρ è in realtà un epimorfismo[3]. Dimostrato ciò, dedurremo che l'immagine di ρ coincide con l'insieme di arrivo dell'applicazione ($\mathrm{Im}(\rho) = \mathbb{Z}_p^{\mathbb{Z}_p}$) e dunque l'anello delle funzioni polinomiali $PF_p[x]$ coincide con $\mathbb{Z}_p^{\mathbb{Z}_p}$. Questo ci consentirà di dire che, tutte le funzioni $f : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$ sono polinomiali.

Per dimostrare la suriettività di ρ, dobbiamo effettuare una breve trattazione che ha come protagonista il nucleo del morfismo ρ.
Innanzitutto ricordiamo che $\ker(\rho)$ è un ideale di $\mathbb{Z}_p[x]$ e che è quindi possibile costruire l'anello quoziente $\dfrac{\mathbb{Z}_p[x]}{\ker(\rho)}$. Infatti se H è un ideale di A, allora $\dfrac{A}{H}$ è un anello con somma e prodotto indotti da A. Consideriamo quindi la proiezione canonica sul quoziente π, che è un morfismo suriettivo di anelli:

$$\pi : \mathbb{Z}_p[x] \longrightarrow \frac{\mathbb{Z}_p[x]}{\ker(\rho)} \tag{1.3.2}$$

La proiezione canonica π associa ad ogni polinomio a coefficienti in $\mathbb{Z}_p$, la classe di equivalenza corrispondente, attraverso la relazione indotta dal nucleo del morfismo ρ:

$$f \equiv_{\ker(\rho)} g \iff f - g \in \ker(\rho). \tag{1.3.3}$$

Se riusciremo a calcolare la cardinalità del quoziente $\dfrac{\mathbb{Z}_p[x]}{\ker(\rho)}$, saremo in grado di rispondere alla domanda che ha aperto il paragrafo.
Il primo passo per caratterizzare il quoziente è senz'altro quello di determinare i generatori del nucleo $\ker(\rho)$.
Siccome $\mathbb{Z}_p$ è un campo, sappiamo che $\mathbb{Z}_p[x]$ è un dominio euclideo e dunque un PID: tutti i suoi ideali sono principali, ossia generati da un solo elemento. Questo prova, in particolare, che l'ideale $\ker(\rho)$ è principale.
Si tratta ora di trovare il polinomio che genera $\ker(\rho)$, attraverso il piccolo teorema di Fermat. Ciò che noi sappiamo è che $f \equiv_{\ker(\rho)} g \iff f - g \in \ker(\rho)$, ossia se e solamente se $f - g$ è congruo a 0 modulo p. Il piccolo teorema di Fermat, ci assicura che $x^p \equiv x$,

[3]Si dice **epimorfismo** un morfismo suriettivo.

modulo p e quindi $x^p - x$ è il generatore che cercavamo:

$$\ker(\rho) = (x^p - x) \tag{1.3.4}$$

Infatti, poiché $\mathbb{Z}_p$ è un campo, nessun polinomio di grado minore di p può avere come radici tutti gli elementi di $\mathbb{Z}_p$: il numero delle radici di un polinomio non può mai superare il suo grado.

A questo punto vogliamo studiare il quoziente:

$$\frac{\mathbb{Z}_p[x]}{\ker(\rho)} = \frac{\mathbb{Z}_p[x]}{(x^p - x)} \tag{1.3.5}$$

Per farlo, utilizziamo il Teorema Fondamentale di Omomorfismo per Anelli.

Costruiamo innanzitutto il seguente diagramma che riassume tutte le funzioni "in gioco":

$$\begin{array}{ccc} \mathbb{Z}_p[x] & \xrightarrow{\rho} & \mathbb{Z}_p^{\mathbb{Z}_p} \\ {\scriptstyle\pi}\downarrow & & \uparrow{\scriptstyle i} \\ \dfrac{\mathbb{Z}_p[x]}{(x^p - x)} & & \mathrm{Im}(\rho) \end{array}$$

- π è la proiezione canonica sul quoziente;
- ρ è il morfismo dal quale è iniziata la trattazione;
- i è il morfismo di inclusione.

Per il Teorema Fondamentale di Omomorfismo per Anelli, sappiamo che esiste un isomorfismo ϕ che lega l'anello quoziente $\dfrac{\mathbb{Z}_p[x]}{(x^p - x)}$ e $\mathrm{Im}(\rho)$:

$$\begin{aligned} \phi : \ \frac{\mathbb{Z}_p[x]}{(x^p - x)} &\longrightarrow \mathrm{Im}(\rho) \\ \bar{f} \quad &\longrightarrow \rho(f) \end{aligned} \tag{1.3.6}$$

Dunque, ϕ associa alla classe di equivalenza rappresentata dal polinomio f, l'unica funzione polinomiale associata a tutti gli elementi della classe.

Il diagramma commutativo diventa quindi:

$$\begin{array}{ccc} \mathbb{Z}_p[x] & \xrightarrow{\rho} & \mathbb{Z}_p^{\mathbb{Z}_p} \\ {\scriptstyle\pi}\downarrow & & \uparrow{\scriptstyle i} \\ \dfrac{\mathbb{Z}_p[x]}{(x^p - x)} & \xrightarrow{\phi} & \mathrm{Im}(\rho) \end{array} \tag{1.3.7}$$

L'esistenza dell'isomorfismo ϕ è il punto cardine di questo discorso. Infatti, è noto che la cardinalità dell'insieme quoziente è p^p e allora anche $\mathrm{Im}(\rho)$ avrà un numero di elementi pari a p^p. $\mathrm{Im}(\rho)$ è però un sottoanello di $\mathbb{Z}_p^{\mathbb{Z}_p}$, che ha anch'esso cardinalità p^p, e dunque si ha necessariamente che l'immagine del morfismo ρ coincide con $\mathbb{Z}_p^{\mathbb{Z}_p}$. Ciò equivale a dire che ρ è un epimorfismo.
Questo ci garantisce, come già accennato sopra, che tutte le p^p funzioni definite da $\mathbb{Z}_p$ in sé sono polinomiali:

$$PF_p[x] = \mathrm{Im}(\rho) = \mathbb{Z}_p^{\mathbb{Z}_p} \tag{1.3.8}$$

1.4 Dalla funzione polinomiale al polinomio

Data una funzione $F : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$, abbiamo appena dimostrato che questa è polinomiale. Utilizzando una tabella, è possibile individuare il polinomio f corrispondente, con $\deg(f) \leq p-1$.

$\boldsymbol{x}$	$\boldsymbol{F(x)}$
$[0]_p$	y_0
$[1]_p$	y_1
$[2]_p$	y_2
...	...
$[p-1]_p$	y_{p-1}

Tabella 1.1: Tabella relativa ai valori assunti dalla funzione $F : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$.

Abbiamo indicato con y_j i valori assunti dalla funzione F in corrispondenza della j-esima classe di equivalenza modulo p. Dunque possiamo affermare che:

$$\sum_{i=0}^{p-1} a_i x_j^i = y_j \qquad \forall j \ \text{ t.c. } \ 0 \leq j \leq p-1 \tag{1.4.1}$$

Al variare di j, questo rappresenta un sistema $p \times p$ di equazioni nelle incognite a_i, per $i = 0, ...p-1$.
La matrice dei coefficienti sarà

$$A = [x_j^i]_{0 \leq i,j \leq p-1} \tag{1.4.2}$$

A è una matrice di Vandermonde, ossia gli elementi delle sue righe sono in progressione geometrica. Sappiamo che il determinante di una matrice di Vandermonde è definito, a meno del segno, come segue:

$$\det(A) = \prod_{0 \le i < j \le p-1} (i - j) \tag{1.4.3}$$

Osservazione 1. $\det(A)$ non è nullo per come sono definiti gli estremi della produttoria.

Osservazione 2. $\det(A)$ non è multiplo di p poiché $i \neq j$ e $|i - j| < p$.

Questo dimostra che la matrice dei coefficienti A è invertibile e quindi la soluzione del sistema di equazioni definito in (1.4.1) esiste e il vettore delle soluzioni ha come componenti dei numeri interi modulo p.

Questo procedimento consente quindi di determinare il polinomio f associato alla funzione polinomiale F.

Esempio 5. Per $p = 3$ consideriamo la seguente tabella relativa alla funzione F calcolata sugli elementi di $\mathbb{Z}_3$:

$\boldsymbol{x}$	$\boldsymbol{F(x)}$
$[0]_3$	$[1]_3$
$[1]_3$	$[1]_3$
$[2]_3$	$[0]_3$

Tabella 1.2

La tabella può essere tradotta nel seguente sistema lineare:

$$\begin{cases} a_0 + [0]_3 \cdot a_1 + [0]_3 \cdot a_2 = [1]_3 \\ a_0 + a_1 + a_2 = [1]_3 \\ a_0 + [2]_3 \cdot a_1 + a_2 = [0]_3 \end{cases} \tag{1.4.4}$$

Calcoliamo il determinante della matrice dei coefficienti A, eliminando per semplicità di notazione le parentesi quadre:

$$\det A = \det \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} = 1 - 2 = -1 \equiv 2 \mod 3 \tag{1.4.5}$$

Risolvendo il sistema lineare con il metodo di Cramer, si ottengono i seguenti risultati:

- $a_0 = 1$;
- $a_1 = \frac{1}{2} \cdot \det \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = 2 \cdot (+1) = 2 \mod 3$;
- $a_2 = \frac{1}{2} \cdot \det \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 0 \end{bmatrix} = 2 \cdot (-1) = -2 \equiv 1 \mod 3$.

Ne viene che il polinomio f associato alla funzione polinomiale F è:

$$f(x) = x^2 + 2x + 1 = (x+1)^2. \tag{1.4.6}$$

Vedremo nel prossimo capitolo che, per m non primo, esistono invece funzioni da $\mathbb{Z}_m$ in sé che non sono polinomiali.
Consideriamo ora $m > 1$, numero intero non primo, un insieme X con m elementi e un isomorfismo qualsiasi siffatto:

$$\phi : X \longrightarrow \mathbb{Z}_m.$$

Per qualsiasi scelta di ϕ esistono funzioni $f : X \longrightarrow X$ t.c. $\phi \circ f \circ \phi^{-1} : \mathbb{Z}_m \longrightarrow \mathbb{Z}_m$ che non sono polinomiali.

Esempio 6. Consideriamo ancora l'anello $\mathbb{Z}_4$ e la seguente funzione definita su di esso:

$$f(x) = \begin{cases} [0]_4, & \text{se } x = [0]_4, [1]_4 \\ [1]_4, & \text{se } x = [2]_4, [3]_4 \end{cases}$$

Se tale funzione fosse rappresentabile tramite un polinomio, si avrebbe:

$$f(x) = \sum_{k=0}^{N} a_k x^k$$

e dunque

$$a_0 = f([0]_4) = [0]_4 \text{ e } \ f([2]_4) = [2]_4 a_1 = [1]_4$$

Ma, $\forall k \geq 2$, $a_k [2]_4^k = [0]_4$ e $[2]_4$ non è invertibile in $\mathbb{Z}_4$. Quindi tale funzione non è polinomiale, pur essendo definita da $\mathbb{Z}_m$ in sé.

Generalizziamo quanto osservato nel seguente risultato.[4]

Teorema 1.4.1. *Sia A un anello commutativo unitario di cardinalità finita:*

- *se A non è un campo, allora esistono funzioni $A \longrightarrow A$ non polinomiali;*
- *se A è un campo, allora tutte le funzioni $A \longrightarrow A$ sono polinomiali.*

[4]Per la dimostrazione del teorema si rimanda alla tesi di Laurea della dottoressa Accogli [5].

Capitolo 2

Funzioni polinomiali in $\mathbb{Z}_{p^2}$ per p=3 e p=5

In questo capitolo vogliamo utilizzare gli strumenti dell'algebra per calcolare il numero di funzioni polinomiali negli anelli del tipo $\mathbb{Z}_{p^n}$, con $n \geq 2$. Ci interessa in particolare trattare le potenze di 3 e di 5.[1]

2.1 Funzioni polinomiali nel generico anello $\mathbb{Z}_{p^n}$

Ricordiamo che ogni polinomio $\sum_{i=0}^{k} a_i x^i \in \mathbb{Z}_{p^n}[x]$ con $k \geq 0$ è associato ad una funzione polinomiale f tale che $f(x) = \sum_{i=0}^{k} a_i x^i$.
Riproponiamo inoltre la definizione del morfismo ρ, già descritto nel capitolo precedente:

$$\rho : \mathbb{Z}_{p^n}[x] \longrightarrow (\mathbb{Z}_{p^n})^{\mathbb{Z}_{p^n}} \tag{2.1.1}$$

L'immagine di questo morfismo, esattamente come accadeva nel caso di $\mathbb{Z}_p$, è l'anello delle funzioni polinomiali su $\mathbb{Z}_{p^n}$ ed è isomorfo al quoziente $\frac{\mathbb{Z}_{p^n}[x]}{\ker(\rho)}$. Il nostro obiettivo

[1] Per quanto riguarda le potenze di 2, abbiamo deciso di non riportare il procedimento (che è comunque analogo agli altri che vengono presentati) poiché esso è già stato esaurientemente descritto nella tesi di Laurea della dottoressa Roberta Mazzone [6].

quindi è ancora quello di determinare il nucleo del morfismo ρ e l'ordine dell'anello quoziente $\dfrac{\mathbb{Z}_{p^n}[x]}{\ker(\rho)}$.
Vediamo ora un metodo generale per calcolare esattamente il numero di funzioni polinomiali in un generico anello $\mathbb{Z}_n$.
In primo luogo occorre determinare il minimo intero k tale che $x^k : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n$ sia combinazione lineare di monomi di grado inferiore. Chiamiamo m tale minimo: questa osservazione ci garantisce che il grado delle funzioni polinomiali da $\mathbb{Z}_n$ in sé sia al più $m-1$.
A questo punto, per tutti gli interi $h \leq m-1$, si cerca di esprimere il polinomio $a \cdot x^h$, con $a \neq 0$, ancora come combinazione lineare di monomi di grado più basso.
Effettuato questo calcolo, è sufficiente determinare in pratica il numero di funzioni polinomiali distinte.
Per determinare i generatori di $\ker(\rho)$ si può usare il seguente risultato.

Lemma 2.1.1. *Sia $\rho : \mathbb{Z}_n[x] \longrightarrow \mathbb{Z}_n^{\mathbb{Z}_n}$ definita come in (2.1.1) e sia h un divisore di n. Si ha dunque $n = h \cdot q$. Allora:*

1. *il polinomio $f(x) = h \cdot \prod_{r=0}^{q-1} (x-r) \in \ker \rho$;*
2. *il polinomio $g(x) = h \cdot \prod_{i=0}^{m-1} (x+i) \in \ker \rho$, dove $m = \min\{k \in \mathbb{Z} \ , \ m > 0 \ t.c. \ q | m!\}$;*
3. *il polinomio $p(x) = \prod_{j=0}^{r} (x+j) \in \ker \rho$, dove $r = \min\{h \in \mathbb{Z} \ , \ m > 0 \ t.c. \ n | r!\}$.*

Dimostrazione.

1. Considero $x = q \cdot s + r$ con $0 \leq r < q$ e studio il prodotto $h \cdot (x-r)$:

$$x = q \cdot s + r \Longrightarrow x - r = q \cdot s \Longrightarrow h \cdot (x-r) = h \cdot q \cdot s \equiv 0 \mod n$$

   Allora, il polinomio $f(x)$ è congruo a zero modulo n e quindi appartiene al nucleo del morfismo ρ.

2. Consideriamo ora il polinomio $g(x)$. Questo polinomio è ancora identicamente nullo in $\mathbb{Z}_n$, grazie alla seguente proprietà aritmetica: *comunque presi k, n e m interi positivi tali che $k|m!$, allora k divide anche* $\prod_{i=0}^{m-1}(n+i)$.

3. La dimostrazione di questo ultimo punto deriva da quella precedente considerando $h = 1$ e $q = n$.

□

Il lemma appena citato può essere riformulato considerando come n non un intero qualsiasi, ma la potenza di un numero primo p. In questo modo ci avviciniamo all'ambito di nostro interesse.

Lemma 2.1.2. *Considero il numero primo p e la sua potenza p^h, con $h \geq 1$. Allora:*

1. *il polinomio* $p(x) = \prod_{i=0}^{p-1}(x+i)^h \in \ker\rho$ *ed è monico di grado* $p \cdot h$;

2. *il polinomio* $q(x) = \prod_{j=0}^{m-1}(x+j) \in \ker\rho$ *ed è monico di grado m, con* $m = \min\{k \in \mathbb{Z}\ ,\ m > 0\ t.c.\ p^h|k!\}$.

Questi lemmi ci consentono di determinare i generatori del $\ker\rho$ e di calcolare, di conseguenza, il numero di funzioni polinomiali in un generico anello $\mathbb{Z}_{p^n}$.

2.2 Numero delle funzioni polinomiali in $\mathbb{Z}_9$

Iniziamo calcolando il numero delle funzioni polinomiali in $\mathbb{Z}_{p^n}$, per $p = 3$ e $n = 2$, ossia nell'anello degli interi modulo 9.
Per aiutarci con le prime catene di congruenze consideriamo una tabella che riguarda le potenze modulo 9.

	$3x$	$3x^2$	$3x^3$	$3x^4$	$3x^5$	$3x^6$	$3x^7$	$3x^8$	$3x^9$
$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$
$[1]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$
$[2]_9$	$[6]_9$	$[3]_9$	$[6]_9$	$[3]_9$	$[6]_9$	$[3]_9$	$[6]_9$	$[3]_9$	$[6]_9$
$[3]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$
$[4]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$
$[5]_9$	$[6]_9$	$[3]_9$	$[6]_9$	$[3]_9$	$[6]_9$	$[3]_9$	$[6]_9$	$[3]_9$	$[6]_9$
$[6]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$	$[0]_9$
$[7]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$	$[3]_9$
$[8]_9$	$[6]_9$	$[3]_9$	$[6]_9$	$[3]_9$	$[6]_9$	$[3]_9$	$[6]_9$	$[3]_9$	$[6]_9$

Tabella 2.1: Le potenze del tipo $3x^k$ nell'anello $\mathbb{Z}_9$, per $k = 0, ...9$.

Da questi calcoli evinciamo che $3x \equiv 3x^3 \mod 9$ e $3x^2 \equiv 3x^4 \mod 9$. Per trovare un altro generatore utilizziamo i lemmi descritti in precedenza e osserviamo che il polinomio

$$f(x) = x(x + [1]_9)(x + [2]_9)(x + [3]_9)(x + [4]_9)(x + [5]_9)$$

è identicamente nullo in $\mathbb{Z}_9$. Infatti, se consideriamo sei numeri consecutivi, vi saranno senza dubbio almeno due multipli di 3 e dunque il loro prodotto sarà multiplo di 9.
Calcolando esplicitamente $f(x)$ otteniamo:

$$\begin{aligned} f(x) &= x^6 + [15]_9x^5 + [85]_9x^4 + [225]_9x^3 + [274]_9x^2 + [120]_9x \\ &= x^6 + [6]_9x^5 + [4]_9x^4 + [4]_9x^2 + [3]_9x \end{aligned}$$

Ricordando le congruenze determinate sopra osserviamo che:

- $[6]_9x^5 = [3]_9x^5 + [3]_9x^5 = [3]_9x + [3]_9x = [6]_9x$;
- $[4]_9x^4 = [3]_9x^4 + x^4 = [3]_9x^2 + x^4$.

Allora, possiamo riscrivere il polinomio come segue:

$$f(x) = x^6 + x^4 + [7]_9x^2 \tag{2.2.1}$$

Questo significa, in particolare, che $x^6 \equiv 2x^4 + 8x^2 \mod 9$ e quindi che le funzioni polinomiali in $\mathbb{Z}_9$ hanno al massimo grado 5, dunque i polinomi che le rappresentano sono della forma

$$p(x) = a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0, \quad \text{con } a_i \in \mathbb{Z}_9 \ \ \forall i = 0, ...5 \tag{2.2.2}$$

Studiamo ora quanti e quali valori possono assumere i coefficienti a_i: il loro computo darà il numero di funzioni polinomiali da $\mathbb{Z}_9$ in sé.

- $a_5 = [0]_9,\ [1]_9,\ [2]_9$.
  Infatti, tutti gli altri valori si possono ricondurre a questi attraverso la congruenza $3x^5 \equiv 3x \mod 9$:
  - $[3]_9x^5 = [3]_9x$, allora $a_5 \neq [3]_9$;
  - $[4]_9x^5 = x^5 + [3]_9x$, allora ci si riconduce a $a_5 = [1]_9$;
  - $[5]_9x^5 = [2]_9x^5 + [3]_9x$, allora ci si riconduce a $a_5 = [2]_9$;
  - $[6]_9x^5 = [6]_9x$, allora $a_5 \neq [6]_9$;
  - $[7]_9x^5 = x^5 + [6]_9x$, allora ci si riconduce a $a_5 = [1]_9$;
  - $[8]_9x^5 = [2]_9x^5 + [6]_9x$, allora ci si riconduce a $a_5 = [2]_9$.
- $a_4 = [0]_9,\ [1]_9,\ [2]_9$.
  Si prova con lo stesso criterio di a_5, utilizzando la congruenza $3x^4 \equiv 3x^2 \mod 9$.
- $a_3 = [0]_9,\ [1]_9,\ [2]_9$.
  Si prova in maniera esattamente identica a a_5.

I coefficienti della seconda e prima potenza e il termine noto possono invece assumere qualsiasi valore nell'anello $\mathbb{Z}_9$.

- $a_2 \in \mathbb{Z}_9$;
- $a_1 \in \mathbb{Z}_9$;
- $a_0 \in \mathbb{Z}_9$.

Abbiamo così provato che le possibili combinazioni dei coefficienti del generico polinomio $f(x)$ associato ad una funzione lineare in $\mathbb{Z}_9$ sono:

$$3 \cdot 3 \cdot 3 \cdot 3^2 \cdot 3^2 \cdot 3^2 = 3^{1+1+1+2+2+2} = 3^9 = 19683 \tag{2.2.3}$$

Quindi, in conclusione, vi sono 19683 funzioni polinomiali definite da $\mathbb{Z}_9$ in sé.

2.3 Numero delle funzioni polinomiali in $\mathbb{Z}_{25}$

Vogliamo ora ripercorrere il procedimento appena illustrato per calcolare il numero di funzioni polinomiali definite sull'anello $\mathbb{Z}_{25}$ e a valori in esso.
Per prima cosa osserviamo che $[5]_{25}x^5 - [5]_{25}x$ è un polinomio identicamente nullo su $\mathbb{Z}_{25}$ e dunque vale la congruenza

$$5x^5 \equiv 5x \mod 25 \tag{2.3.1}$$

Appurato questo consideriamo il polinomio seguente, ottenuto attraverso i lemmi descritti nel caso generale:

$$f : x(x+[1]_{25})(x+[2]_{25})(x+[3]_{25})(x+[4]_{25})(x+[5]_{25})(x+[6]_{25})(x+[7]_{25})(x+[8]_{25})(x+[9]_{25})$$

Il polinomio f è identicamente nullo modulo 25 (osserviamo anche che il prodotto di 10 numeri consecutivi contiene due fattori multipli di 5 ed è quindi congruo a 0 modulo 25). Esplicitando i conti (eventualmente anche con l'ausilio di un software) e calcolando modulo 25 si ottiene:

$$f(x) = x^{10} + [20]_{25}x^9 + [20]_{25}x^8 + [23]_{25}x^6 + [5]_{25}x^4 + x^2 + [5]_{25}x \tag{2.3.2}$$

Utilizziamo la congruenza $5x^5 \equiv 5x \mod 25$ e osserviamo che:

- $[20]_{25}x^9 = [20]_{25}x$;
- $[20]_{25}x^8 = [20]_{25}x^4$;
- $[23]_{25}x^6 = [20]_{25}x^2 + [3]_{25}x^6$.

Questo ci permette di riscrivere il polinomio f come segue:

$$f(x) = x^{10} + [3]_{25}x^6 + [21]_{25}x^2 \tag{2.3.3}$$

Di conseguenza, ne deduciamo che $x^{10} \equiv 2x^6 - x^2 \mod 25$ e quindi che i polinomi associati alle funzioni polinomiali in $\mathbb{Z}_{25}$ sono al più di grado 9. Quindi, la generica funzione polinomiale in $\mathbb{Z}_{25}$ sarà associata al polinomio:

$$p(x) = a_9x^9 + a_8x^8 + a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0, \tag{2.3.4}$$

con $a_i \in \mathbb{Z}_{25} \ \ \forall i = 0, ...9$.
Cerchiamo ora di determinare i valori che possono assumere i coefficienti di questo polinomio e calcolare così il numero di funzioni polinomiali.

- $a_9 = [0]_{25},\ [1]_{25},\ [2]_{25},\ [3]_{25},\ [4]_{25}$.
  Infatti, tutti gli altri valori si possono ricondurre a questi attraverso la congruenza $5x^5 \equiv 5x \mod 25$.

Stesso ragionamento si può applicare ai coefficienti di x fino alla quinta potenza.

- $a_8 = [0]_{25},\ [1]_{25},\ [2]_{25},\ [3]_{25},\ [4]_{25}$;
- $a_7 = [0]_{25},\ [1]_{25},\ [2]_{25},\ [3]_{25},\ [4]_{25}$;
- $a_6 = [0]_{25},\ [1]_{25},\ [2]_{25},\ [3]_{25},\ [4]_{25}$;
- $a_5 = [0]_{25},\ [1]_{25},\ [2]_{25},\ [3]_{25},\ [4]_{25}$.

I coefficienti di x^4, x^3, x^2, x e il termine noto possono invece assumere qualsiasi valore nell'anello $\mathbb{Z}_{25}$.

- $a_4 \in \mathbb{Z}_{25}$;
- $a_3 \in \mathbb{Z}_{25}$;
- $a_2 \in \mathbb{Z}_{25}$;
- $a_1 \in \mathbb{Z}_{25}$;
- $a_0 \in \mathbb{Z}_{25}$.

Questo calcolo ci permette di concludere che il numero totale delle possibili combinazioni di tali coefficienti è

$$5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5^2 \cdot 5^2 \cdot 5^2 \cdot 5^2 \cdot 5^2 = 5^{1+1+1+1+1+2+2+2+2+2} = 5^{15} = 30\,517\,578\,125 \quad (2.3.5)$$

Resta così calcolato in maniera esatta anche il numero delle funzioni polinomiali in $\mathbb{Z}_{25}$.

Capitolo 3

Funzioni polinomiali in $\mathbb{Z}_{p^n}$: un approccio sperimentale

Osserviamo che già con potenze relativamente basse come 3^2 e 5^2, i calcoli con i polinomi non sono stati banali e i risultati ottenuti si sono rivelati numeri molto alti. Per questo motivo risulterebbe pratico scoprire una regolarità che ci permetta di calcolare efficacemente il numero p^t di funzioni polinomiali nell'anello $\mathbb{Z}_{p^n}$. Per congetturare un andamento generale, utilizziamo gli strumenti della statistica descrittiva, procedendo con un vero e proprio esperimento.

3.1 Congettura delle differenze prime

Riassumiamo i risultati ottenuti nel capitolo precedente e quelli noti sulle potenze di 2.

n	Anello	Numero di funzioni polinomiali
1	$\mathbb{Z}_2$	2^2
2	$\mathbb{Z}_4$	2^6
3	$\mathbb{Z}_8$	2^{10}

Tabella 3.1: Numero di funzioni polinomiali in $\mathbb{Z}_{2^n}$.

n	Anello	Numero di funzioni polinomiali
1	$\mathbb{Z}_3$	3^3
2	$\mathbb{Z}_9$	3^9

Tabella 3.2: Numero di funzioni polinomiali in $\mathbb{Z}_{3^n}$.

n	Anello	Numero di funzioni polinomiali
1	$\mathbb{Z}_5$	5^5
2	$\mathbb{Z}_{25}$	5^{15}

Tabella 3.3: Numero di funzioni polinomiali in $\mathbb{Z}_{5^n}$.

Concentriamoci per il momento sulle sole potenze di 2 tenendo presente che:

- n rappresenta l'esponente di 2 nell'anello degli interi modulo 2^n considerato;
- f_2 è la funzione che associa a n l'esponente k, dove 2^k è il numero di funzioni polinomiali nell'anello $\mathbb{Z}_{2^n}$;
- $\Delta(f_2)$ rappresenta la differenza prima[1], ossia $\Delta(f_2) = f_2(n) - f_2(n-1)$, con $n > 1$;
- m è il minimo intero tale che $2^n | m!$.

Anello	n	$f_2(n)$	$\Delta(f_2)$	m
$\mathbb{Z}_2$	1	2		2
$\mathbb{Z}_4$	2	6	4	4
$\mathbb{Z}_8$	3	10	4	4

Tabella 3.4: Calcolo dei valori assunti da $f_2(n)$ per $n = 1, 2, 3$.

[1]Il calcolo delle differenze prime è alla base dell'elaborazione dei dati derivanti da osservazioni sperimentali e spesso, come in questo caso, svela inaspettati retroscena.

Da questi dati, è immediato osservare che la colonna $\Delta(f_2)$ e la colonna m coincidono per $n \geq 2$.
Questa naturalmente potrebbe essere una coincidenza. Osserviamo però che si verifica anche negli altri due casi considerati, ossia $\mathbb{Z}_9$ (in rapporto a $\mathbb{Z}_3$) e $\mathbb{Z}_{25}$ (in rapporto a $\mathbb{Z}_5$).

Anello	n	$f_3(n)$	$\Delta(f_3)$	m
$\mathbb{Z}_3$	1	3		3
$\mathbb{Z}_9$	2	9	6	6

Tabella 3.5: Calcolo dei valori assunti da $f_3(n)$ per $n = 1, 2$.

Anello	n	$f_5(n)$	$\Delta(f_5)$	m
$\mathbb{Z}_5$	1	5		5
$\mathbb{Z}_{25}$	2	15	10	10

Tabella 3.6: Calcolo dei valori assunti da $f_5(n)$ per $n = 1, 2$.

In conclusione, congetturiamo che, fissato un numero primo p:

$$\Delta(f_p) = f_p(n) - f_p(n-1) = m \quad \forall n \in \mathbb{N}, \text{ dove } m = \min\{k \in \mathbb{Z} \text{ t.c. } p^n | k!\} \qquad (3.1.1)$$

A questo punto, vogliamo studiare l'andamento della funzione $f_p(n)$, definendola ricorsivamente sulla base della congettura appena elaborata.

Definizione 3.1. Fissato il numero primo p, sia $f_p : \mathbb{N} \longrightarrow \mathbb{N}$ la funzione definita come segue:

$$f_p(n+1) = f_p(n) + m$$

dove $m = \min\{k \in \mathbb{Z} \text{ t.c. } p^{n+1} | k!\}$ e $f_p(1) = p$.

Il nostro obiettivo è quello di calcolare l'esponente $f_p(n)$, per tutti gli n appartenenti ad un range il più ampio possibile, per scoprirne l'andamento e magari congetturare una regola per il calcolo del numero delle funzioni polinomiali in un generico anello $\mathbb{Z}_{p^n}$.

3.2 Diamo un procedimento generale

Ricostruiamo una tabella, sul tipo della (3.4) in un foglio di calcolo elettronico. Certamente Excel, prodotto dalla Microsoft Corporation® e facente parte del pacchetto Office, è il più conosciuto, ma per la nostra ricerca è più che sufficiente Apache® OpenOffice Calc, un software free per l'elaborazione di fogli elettronici (tra l'altro compatibile con Microsoft Excel). Per ricostruire la tabella, scriviamo sul foglio elettronico la legenda iniziale, inseriamo il numero 1 nella casella `A2` e il numero 2 nella casella `A3`. A questo punto, per trascinamento, completiamo la colonna `A` (che identifica il valore n).

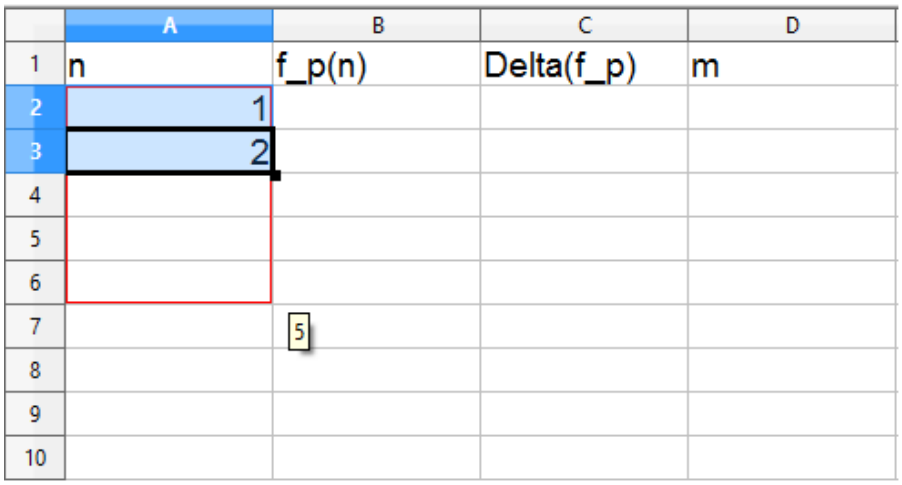

Figura 3.1: Foglio di calcolo: completamento per trascinamento della colonna `A`.

Per il momento è irrilevante fino a quale numero trasciniamo. Occorre senz'altro predisporre un range sufficientemente grande da comprendere l'andamento generale della funzione, ma tale scelta verrà fatta in maniera più consapevole una volta che esamineremo i dati veri e propri. Arrivati a questo punto, per ogni n all'interno del range che abbiamo stabilito, dobbiamo calcolare il minimo m tra i valori di k tali che 2^n divida $k!$. Effettuare questo calcolo manualmente aumenta il rischio di errori, senza contare che limita la possibilità di ampiezza del range. Noi invece vogliamo proporre una soluzione che sia adattabile senza nessuna fatica a range ampi anche decine di migliaia di numeri. Utilizziamo quindi il linguaggio di programmazione ad alto livello Python® per scrivere un programma che effettui al nostro posto questo lungo calcolo. Tale scelta, supponendo

di scrivere un programma corretto ed efficiente, azzera le possibilità di errori di calcolo e accorcia notevolmente i tempi di lavorazione.

Evidenziamo i vari passaggi che portano ad una corretta elaborazione di un programma:[2]

1. descrivere le varie tappe del ragionamento attraverso un algoritmo in linguaggio naturale;
2. descrivere eventuali funzioni necessarie per implementare l'algoritmo;
3. tradurre in linguaggio macchina le funzioni e il programma principale, utilizzando i costrutti del linguaggio di programmazione scelto (nel nostro caso Python®);
4. testare la correttezza e l'efficienza dell'elaborato su alcuni dati noti.

Applichiamo ognuno degli step appena descritti al nostro caso particolare.

1. Vogliamo creare una funzione denominata `calc(p,x0,x)` che, preso in input il numero primo p sul quale ci stiamo concentrando e il range nel quale vogliamo far variare n (ossia gli estremi dell'intervallo), restituisca l'elenco (al variare di n nel range) dei minimi valori m tali che $p^n|m!$.
2. Descriviamo la funzione `f(p,n)`, che dovremo richiamare nel programma principale: essa, presi in input il numero primo p ed il naturale n, restituisce il più piccolo numero intero positivo m t.c. p^n divida $m!$.

[2]Per alcune informazioni sulla programmazione in Python®, si rimanda all'Appendice C.

3. Scriviamo le funzioni descritte sopra utilizzando il linguaggio Python®:

```
def f(p,n):
    stop=p*n+1
    fact=1
    pn=p**n
    for i in range(1, stop):
        fact=fact*i
        if fact%pn==0:
            return i
    return -1

def calc(p,x0,x):
    for k in range (x0,x+1):
        print f(p,k)
```

4. Per verificare la correttezza e l'efficienza delle funzioni stilate, le facciamo girare nella shell scegliendo come input i dati noti calcolati nel capitolo 2.

Abbiamo dunque scritto in linguaggio Python® una funzione che calcola i valori di m che ci occorrono. Basterà copiare i risultati sul foglio di lavoro di partenza per aver completato anche la colonna `D`.

Completiamo ora la casella `B2` con il valore noto di $f_p(1)$ (che deriva da quanto abbiamo dimostrato sui campi finiti).
Per calcolare le differenze finite invece, utilizziamo direttamente le formule previste dal foglio di calcolo. Ricordando che la differenza finita si calcola a partire da $n = 2$, selezioniamo la casella `C3` e scriviamo `=D3`, ed estendiamo per trascinamento: in questo modo, la colonna `C` e la colonna `D` coincideranno, eccetto per $n = 1$.
In conclusione, ci posizioniamo sulla casella `B3`, scriviamo `=B2+D3` ed estendiamo per

trascinamento, fino a completare anche la colonna `B` (questo significa applicare la congettura che abbiamo elaborato sulla ricorsività di f_p).

Abbiamo così tutti gli strumenti (teorici ed informatici) che ci occorrono per costruire la tabella (3.4) per qualsiasi valore di n.

3.3 In pratica: le funzioni polinomiali in $\mathbb{Z}_{2^n}$

In questo paragrafo ci occuperemo di costruire in maniera concreta la tabella relativa alle potenze di 2. Per farlo, seguiremo i passaggi descritti nel paragrafo precedente e faremo uso degli strumenti informatici in nostro possesso.

Riportiamo qui di seguito, su due colonne, la tabella che si ottiene utilizzando il foglio di calcolo OpenOffice Calc per n che va da 1 a 50.

	A	B	C	D
1	n	f_2(n)	delta(f_2)	m
2	1	2		2
3	2	6	4	4
4	3	10	4	4
5	4	16	6	6
6	5	24	8	8
7	6	32	8	8
8	7	40	8	8
9	8	50	10	10
10	9	62	12	12
11	10	74	12	12
12	11	88	14	14
13	12	104	16	16
14	13	120	16	16
15	14	136	16	16
16	15	152	16	16
17	16	170	18	18
18	17	190	20	20
19	18	210	20	20
20	19	232	22	22
21	20	256	24	24
22	21	280	24	24
23	22	304	24	24
24	23	330	26	26
25	24	358	28	28
26	25	386	28	28

	A	B	C	D
1	n	f_2(n)	delta(f_2)	m
27	26	416	30	30
28	27	448	32	32
29	28	480	32	32
30	29	512	32	32
31	30	544	32	32
32	31	576	32	32
33	32	610	34	34
34	33	646	36	36
35	34	682	36	36
36	35	720	38	38
37	36	760	40	40
38	37	800	40	40
39	38	840	40	40
40	39	882	42	42
41	40	926	44	44
42	41	970	44	44
43	42	1016	46	46
44	43	1064	48	48
45	44	1112	48	48
46	45	1160	48	48
47	46	1208	48	48
48	47	1258	50	50
49	48	1310	52	52
50	49	1362	52	52
51	50	1416	54	54

Figura 3.2: Tabella sul numero delle funzioni polinomiali da $\mathbb{Z}_{2^n}$ in sé, per $n = 1, ...50$.

La tabella che abbiamo riportato costituisce il dato imprescindibile della nostra indagine. Noti gli esponenti, individuati da $f_2(n)$, il numero delle funzioni polinomiali definite da $\mathbb{Z}_{2^n}$ in sé si calcola come $2^{f_2(n)}$. Risulta però palese che questa tabella sia di non facile lettura e di ancor meno immediata interpretazione. Per questo riteniamo che sia opportuno tracciare un grafico che riassuma i dati sopra raccolti.

Tra le molteplici tipologie di diagrammi che la statistica descrittiva mette a nostra disposizione si è scelto di utilizzare il **diagramma a dispersione**. Si tratta di un piano cartesiano con il parametro di controllo indicato sull'asse x e la variabile misurata invece posta sull'asse y.

Nel nostro caso, il *parametro di controllo*, ossia la variabile indipendente, è n, mentre $f_2(n)$ è il valore che abbiamo misurato attraverso le nostre analisi. Le coppie del tipo $(n, f_2(n))$ sono rappresentate sul piano da punti (non a caso il diagramma a dispersione viene detto "confidenzialmente" nuvola di punti).

Il foglio di calcolo che abbiamo utilizzato sino ad ora consente di disegnare grafici a dispersione, quindi procediamo. Evidenziamo le colonne `A` e `B` (che rappresentano input e output della funzione f_2 che intendiamo studiare) e clicchiamo sull'icona "Grafico". Scegliamo poi la tipologia *a dispersione solo linee* e diamo l'ok. Il grafico "solo linee" evidenzia l'andamento della spezzata che collega tra loro i punti: esso costituisce quindi, nel nostro caso, la scelta più utile.

Presentiamo due grafici: il primo, monometrico, ci mostra quanto i punti si dispongano in prossimità dell'asse delle ordinate, ma difficilmente ci suggerisce una regolarità; il secondo invece, pur non essendo monometrico, descrive in maniera più immediata l'andamento della funzione f_2.

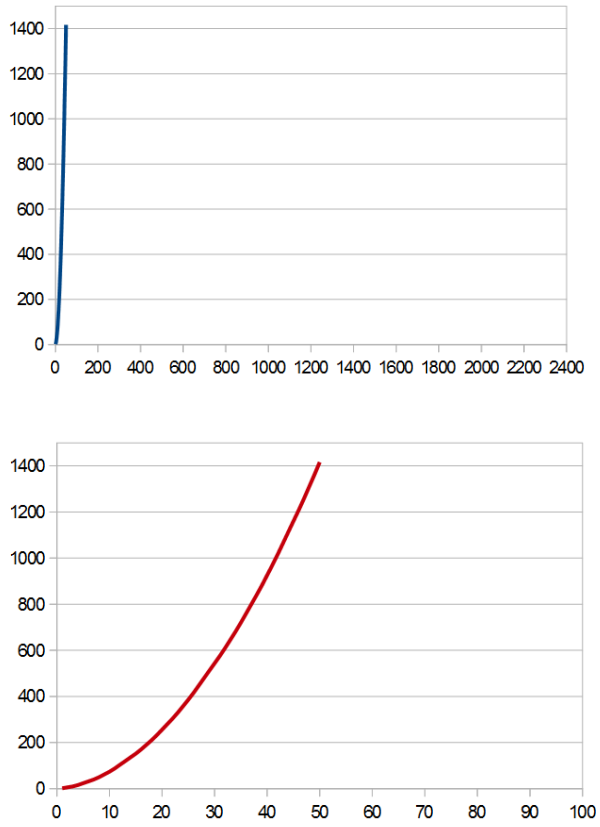

Figura 3.3: Confronto tra grafico monometrico e non monometrico di $f_2(n)$ per $n = 1, ...50$.

É ragionevole ipotizzare che la funzione f_2 presenti un andamento di tipo parabolico. Sarebbe interessante cercare di determinare, con la massima precisione possibile, i coefficienti della parabola interpolante. Per farlo utilizziamo il **metodo dei minimi quadrati** e le nostre conoscenze su **interpolazione** e **regressione polinomiale**.[3]
Ricordiamo che, data la generica parabola $y = ax^2 + bx + c$, i coefficienti a, b e c si

[3]Per una introduzione operativa al metodo dei minimi quadrati e per un breve ripasso di questi concetti statistici e probabilistici si veda l'appendice B.

calcolano attraverso il seguente sistema di equazioni:

$$\begin{cases} c \cdot N + b \cdot \sum x_i + a \cdot \sum x_i^2 = \sum y_i \\ c \cdot \sum x_i + b \cdot \sum x_i^2 + a \cdot \sum x_i^3 = \sum x_i y_i \\ c \cdot \sum x_i^2 + b \cdot \sum x_i^3 + a \cdot \sum x_i^4 = \sum x_i^2 y_i \end{cases} \tag{3.3.1}$$

dove N è il numero di misurazioni e (x_i, y_i) rappresentano le coppie ordinate che individuano i punti noti ossia, nelle notazioni precedenti, le coppie del tipo $(n, f_2(n))$. Le sommatorie, delle quali abbiamo omesso gli indici, sono tutte definite per i che va da 1 a N. Tale sistema è naturalmente complesso da risolvere manualmente, a causa dei numeri molto elevati che devono essere maneggiati. Ricorriamo quindi ancora al software di calcolo OpenOffice Calc il quale, pur non possedendo una funzione predefinita per il calcolo di questi coefficienti, ci permette di risolvere i sistemi lineari in forma matriciale. Andiamo quindi a riscrivere il sistema come segue:

$$\begin{bmatrix} N & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix} \tag{3.3.2}$$

Riscriviamo questo prodotto matriciale sostituendo a x_i e y_i le notazioni che abbiamo utilizzato anche in precedenza, ossia n e $f_2(n)$.

$$\begin{bmatrix} N & \sum_{n=1}^{N} n & \sum_{n=1}^{N} n^2 \\ \sum_{n=1}^{N} n & \sum_{n=1}^{N} n^2 & \sum_{n=1}^{N} n^3 \\ \sum_{n=1}^{N} n^2 & \sum_{n=1}^{N} n^3 & \sum_{n=1}^{N} n^4 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^{N} f_2(n) \\ \sum_{n=1}^{N} n \cdot f_2(n) \\ \sum_{n=1}^{N} n^2 \cdot f_2(n) \end{bmatrix} \tag{3.3.3}$$

Torniamo poi nel nostro file OpenOffice Calc e, accanto alle colonne `n` e `f(n)`, calcoliamo gli altri valori che ci occorrono, inserendo le opportune funzioni predefinite. Estendiamo poi le formule per trascinamento.

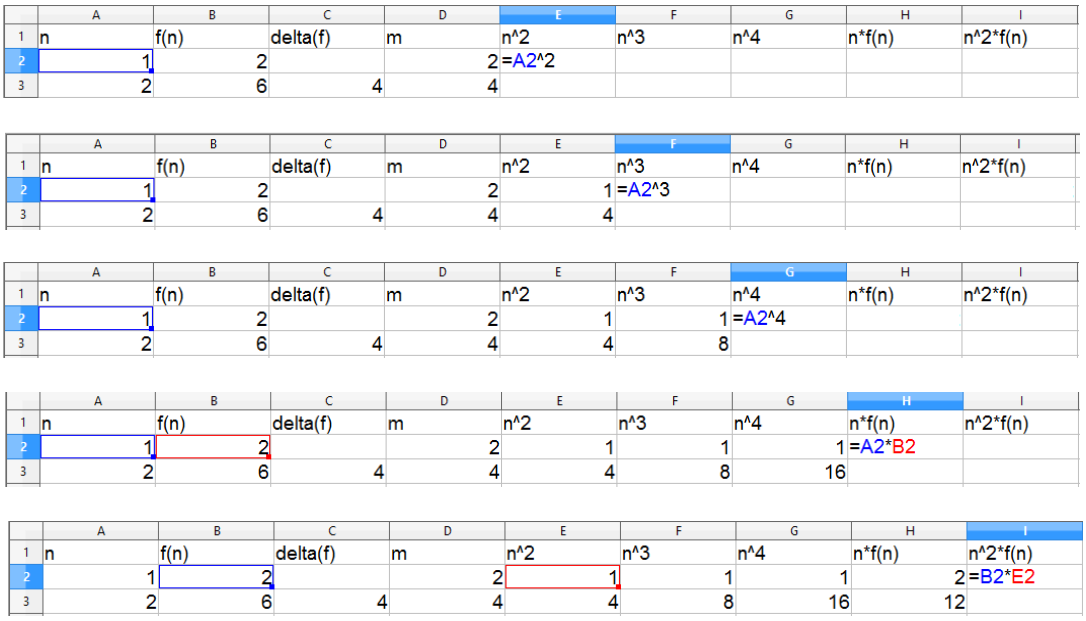

Figura 3.4: Foglio elettronico: funzioni predefinite necessarie per il calcolo del sistema lineare in forma matriciale.

Per risolvere il sistema lineare, ricorriamo alle nozioni note di algebra lineare: se la matrice A è invertibile, allora la soluzione del sistema lineare $AX = b$ è la seguente:

$$X = A^{-1}b \tag{3.3.4}$$

Questo ci consente di calcolare il vettore delle soluzioni selezionando tre caselle verticalmente e dando la funzione:

```
=MATR.PRODOTTO(MATR.INVERSA('matrice A');'vettore b')
```

dove per inserire la matrice A e il vettore dei termini noti b occorre selezionarli manualmente. Per confermare la funzione su tutte e tre le caselle selezionate, si devono premere simultaneamente il tasto INVIO e la barra spaziatrice.
A questo punto, i tre valori ottenuti saranno, nell'ordine dall'alto verso il basso, i parametri c, b e a.

Applicando questo procedimento alle 50 misure che abbiamo rilevato in precedenza, i valori di a, b e c che vengono rilevati sono i seguenti:

- $c_{50} = -3,3244897959$
- $b_{50} = 2,7464724351$
- $a_{50} = 0,5128589897$

La parabola $\mathscr{P}_{50} : y = a_{50}x^2 + b_{50}x + c_{50}$ ha il seguente grafico cartesiano:

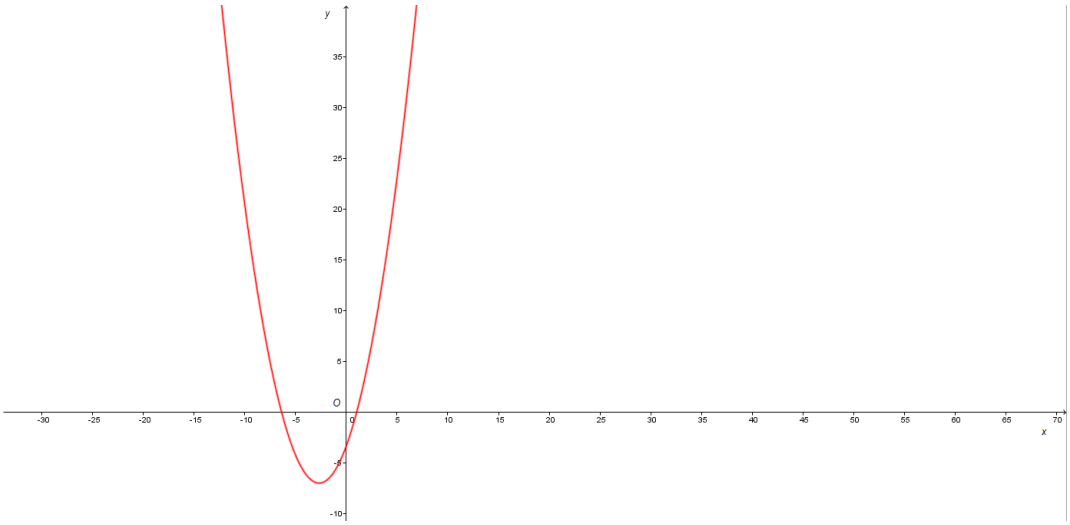

Figura 3.5: Grafico della parabola $\mathscr{P} : y = a_{50}x^2 + b_{50}x + c_{50}$.

Noi in realtà siamo interessati solo al ramo di parabola che occupa il primo quadrante, poiché la funzione f_2 è definita sull'insieme (peraltro discreto) dei numeri naturali.
Per cercare di capire se, all'aumentare del numero di prove, i parametri tendono a qualche terna notevole, effettuiamo lo stesso calcolo (del quale evitiamo di ripetere i passaggi) per n da 1 a 200.

Otteniamo dunque:

- $c_{200} = -16,1169636059$
- $b_{200} = 3,5689928045$
- $a_{200} = 0,5037178090$

Osserviamo che il parametro a continua ad "avvicinarsi" al numero razionale $\frac{1}{2}$ e si ha:

$$a_{50} - a_{200} = 0,5128589897 - 0,5037178090 = 0,0091411807 \tag{3.3.5}$$

Aumentiamo ancora il numero delle prove, per arrivare ad un'approssimazione sempre migliore e dunque ripetiamo il calcolo per n che va da 1 a 1000. I valori che otteniamo confermano le nostre ipotesi:

- $c_{1000} = -68,2154803076$
- $b_{1000} = 4.4730448255$
- $a_{1000} = 0,5010097685$

Osservazione 3. É curioso osservare che, se il coefficiente a sembra tendere al valore $\frac{1}{2}$, b e c invece continuano a divergere. In particolare, b si avvicina a $-\infty$, mentre c tende a $+\infty$, seppur meno rapidamente di b. Quindi, all'apparenza, la convessità della parabola può essere determinata, mentre non si può dire lo stesso della sua posizione all'interno del piano cartesiano: infatti, come sappiamo, i coefficienti b e c determinano rispettivamente la posizione dell'asse di simmetria (quindi la longitudine della parabola) e l'intersezione con l'asse delle ordinate.

3.3.1 Verifichiamo l'adeguatezza dell'interpolazione

Costruiamo una tabella che mostra alcuni valori notevoli di confronto: affianchiamo il valore di $f_2(n)$ misurato attraverso la congettura sulle differenze prime con quello ottenuto attraverso la parabola di regressione $\mathscr{P}_2^{(1000)} : y = a_{1000}n^2 + b_{1000}n + c_{1000}$.

n	$f_2(n)$	$y_2^{(1000)}(n)$ (arrotondato all'intero più vicino)
1	2	-63
10	74	27
50	1416	1408
100	5376	5389
200	20842	20867
500	127494	127421

Tabella 3.7: Confronto tra $f_2(n)$ e $y_2^{(1000)}(n)$, per alcuni valori di n.

I valori di b_{1000} e c_{1000} evidentemente non risultano adeguati a descrivere, per esempio, $f_2(n)$ per $n = 0, ...50$ mentre, all'aumentare di n, aumenta anche il livello di precisione. Consideriamo per esempio il valore di $f_2(10)$ calcolato attraverso la congettura delle differenze prime:

$$f_2(10) = 74$$

Se sostituiamo $x = 10$ nell'equazione della parabola $\mathscr{P}_2^{(1000)}$ otteniamo invece:

$$y_2^{(1000)}(10) = 26,6159447949$$

É chiaro che il valore trovato è ben lontano da quello auspicato. Viceversa, se effettuiamo lo stesso calcolo andando a sostituire $x = 10$ nell'equazione della parabola $\mathscr{P}_2^{(50)}$ ciò che otteniamo è qualcosa di più rassicurante:

$$y_2^{(50)}(10) = 75,4261335303$$

Tale differenza così accentuata fa supporre che il modello parabolico non sia così adeguato come sembrava di primo impatto.

Per questo motivo è necessario effettuare dei test specifici che quantifichino la bontà dell'interpolazione. Abbiamo scelto di utilizzare il test di Pearson che ci consente di determinare il coefficiente di correlazione tra i dati teorici e i dati sperimentali.

Vista la quantità e qualità dei dati, per calcolare il coefficiente di correlazione abbiamo scelto di utilizzare la funzione predefinita

```
=PEARSON(dati1;dati2)
```

presente nel foglio di calcolo: al posto delle diciture `dati1` e `dati2` abbiamo inserito rispettivamente i valori di $f_2(n)$ e $y_2^{(1000)}(n)$, per $n = 1, ...1000$.
Il risultato che otteniamo è:

$$R = 0,999999988 \tag{3.3.6}$$

Il fatto che tale valore sia decisamente prossimo a 1, ci fa ragionevolmente supporre di aver costruito una buona funzione interpolante.
A supporto di questa ipotesi vi sono anche le osservazioni seguenti:

- il massimo errore assoluto che si ottiene è circa 88, ossia

  $$\max\{|f_2(n) - y_2^{(1000)}(n)| = 88 \text{ (arrotondato agli interi)}$$

  e si ottiene per $n = 507$;

- la somma di tutte le differenze $f_2(n) - y_2^{(1000)}(n)$, per $n = 1, ...1000$ vale appena:

  $$\sum_{n=1}^{1000}(f_2 - y_2^{(1000)})(n) = 0.00739991$$

Quanto descritto non ci garantisce naturalmente che la funzione di interpolazione che abbiamo costruito sia ottimale, ma ci fa supporre, se non altro, che i nostri calcoli non siano troppo lontani dall'essere esatti.

3.4 In pratica: le funzioni polinomiali in $\mathbb{Z}_{3^n}$

Il procedimento descritto per gli anelli $\mathbb{Z}_{2^n}$ può essere esteso, senza stravolgimenti, alle potenze di 3 e, in generale, di tutti i numeri primi.
Costruiamo il foglio elettronico relativo alla funzione f_3, sfruttando la congettura che $f_3(n+1) = f_3(n) + m$, con $m = \min\{k \in \mathbb{Z}, k > 0 \text{ t.c. } 3^{n+1}|k!\}$. Come già fatto in precedenza, includiamo nella tabella i valori di $f_3(n)$ per $n = 1, ...50$.

	A	B	C	D
1	n	f(n)	delta(f)	m
2	1	3		3
3	2	9	6	6
4	3	18	9	9
5	4	27	9	9
6	5	39	12	12
7	6	54	15	15
8	7	72	18	18
9	8	90	18	18
10	9	111	21	21
11	10	135	24	24
12	11	162	27	27
13	12	189	27	27
14	13	216	27	27
15	14	246	30	30
16	15	279	33	33
17	16	315	36	36
18	17	351	36	36
19	18	390	39	39
20	19	432	42	42
21	20	477	45	45
22	21	522	45	45
23	22	570	48	48
24	23	621	51	51
25	24	675	54	54
26	25	729	54	54

	A	B	C	D
1	n	f(n)	delta(f)	m
27	26	783	54	54
28	27	840	57	57
29	28	900	60	60
30	29	963	63	63
31	30	1026	63	63
32	31	1092	66	66
33	32	1161	69	69
34	33	1233	72	72
35	34	1305	72	72
36	35	1380	75	75
37	36	1458	78	78
38	37	1539	81	81
39	38	1620	81	81
40	39	1701	81	81
41	40	1782	81	81
42	41	1866	84	84
43	42	1953	87	87
44	43	2043	90	90
45	44	2133	90	90
46	45	2226	93	93
47	46	2322	96	96
48	47	2421	99	99
49	48	2520	99	99
50	49	2622	102	102
51	50	2727	105	105

Figura 3.6: Foglio elettronico: numero delle funzioni polinomiali da $\mathbb{Z}_{3^n}$ in sé, per $n = 1, ...50$.

Costruiamo ora il diagramma a dispersione e osserviamo che anche i valori assunti da f_3 si dispongono lungo una linea che ricorda il grafico di una parabola.

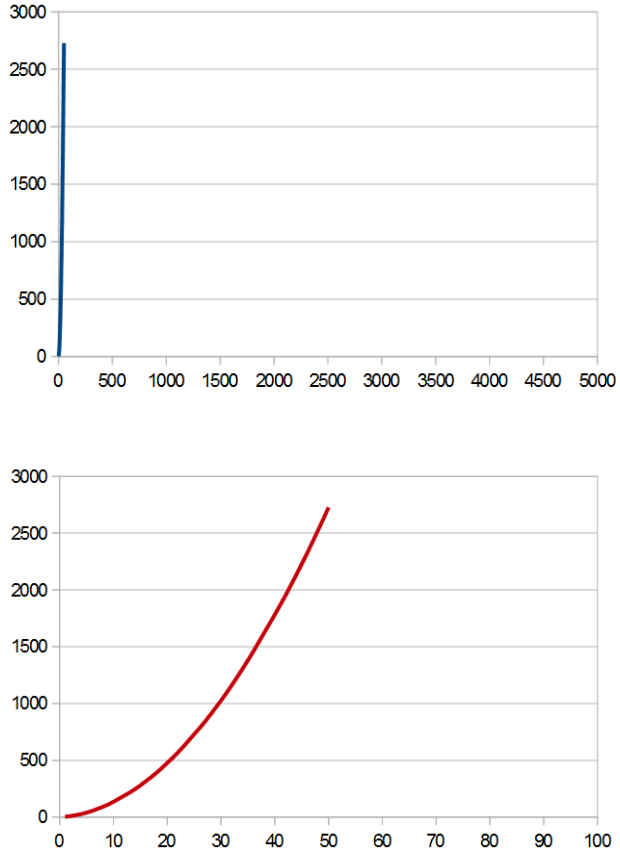

Figura 3.7: Confronto tra grafico monometrico e non monometrico di $f_3(n)$ per $n = 1, ...50$.

Abbiamo già osservato che, nel caso di $p = 2$ si evince una certa regolarità nel coefficiente a del termine di secondo grado. Applichiamo allora di nuovo il metodo dei minimi quadrati per calcolare i coefficienti della parabola di regressione e continuiamo ad aiutarci con il foglio di lavoro OpenOffice Calc.

I valori che otteniamo per $n = 1, ...50$ sono i seguenti:

- $c_{50} = -6,1230612245$
- $b_{50} = 3,9794685567$
- $a_{50} = 1,0152668760$

Ripetiamo il calcolo per $n = 1, ...200$ e per $n = 1, ...1000$:

- $c_{200} = -25,6401530380$
- $b_{200} = 5,0986029389$
- $a_{200} = 1,0036167460$
- $c_{1000} = -84,2984416075$
- $b_{1000} = 6,0357530554$
- $a_{1000} = 1,0013348140$

Osserviamo che l'andamento di questi coefficienti è paragonabile a quello che abbiamo rilevato nel caso delle potenze di 2: il coefficiente a sembra tendere ad un valore noto (in questo caso $a = 1$), b è positivo e cresce al crescere di n, c è negativo e decresce all'aumentare di n, in maniera all'apparenza "più veloce" rispetto a b.

3.4.1 Verifichiamo l'adeguatezza dell'interpolazione

Costruiamo nuovamente una tabella di confronto, come già effettuato nel paragrafo relativo agli anelli $\mathbb{Z}_{2^n}$.

n	$f_3(n)$	$y^3_{1000}(n)$ (arrotondato all'intero più vicino)
1	3	-77
10	135	76
50	2727	2720
100	10521	10533
200	41148	41176
500	253242	253267

Tabella 3.8: Confronto tra $f_3(n)$ e $y_3^{(1000)}(n)$, per alcuni valori di n.

Possiamo osservare che, al crescere di n, cala il divario tra le due misurazioni e la parabola $\mathscr{P}_3^{(1000)}$ rappresenta sempre meglio il fenomeno studiato.
Anche in questo caso utilizziamo il test di Pearson per studiare l'adeguatezza della funzione interpolante. Il calcolo, effettuato attraverso la funzione predefinita del foglio di calcolo elettronico, dà come risultato:

$$R = 0.9999999953 \tag{3.4.1}$$

Il valore ottenuto è ancora prossimo a 1 e quindi possiamo supporre coerente la costruzione della parabola $\mathscr{P}_3$. Osserviamo inoltre che:

- il massimo errore assoluto è circa 94, e si ottiene per $n = 725$;
- la somma di tutte le differenze $f_3(n) - y_3^{(1000)}(n)$, per $n = 1, ...1000$ vale appena:

$$\sum_{n=1}^{1000}(f_3 - y_3^{(1000)})(n) = 0.000000078$$

Come prima, i risultati che otteniamo ci fanno ben sperare: pare che la parabola di regressione che abbiamo calcolato sia, tutto sommato, una buona approssimazione di $f_3(n)$.

In conclusione, la convessità della parabola di regressione relativa alle funzioni polinomiali negli anelli $\mathbb{Z}_{3^n}$ è determinata dal coefficiente a del termine di secondo grado, che abbiamo visto tendere a 1.

3.5 Apertura della parabola di regressione al variare di p

Ora ci chiediamo se il fatto che l'andamento dei punti $(n, f_p(n))$ sia simile ad una parabola sia una casualità legata ai numeri primi 2 e 3, oppure una caratteristica permanente anche per $p > 3$, con p primo. Per rispondere a questo interrogativo riportiamo i diagrammi a dispersione delle funzioni $f_5(n)$, $f_7(n)$, $f_{11}(n)$ e $f_{13}(n)$ per $n = 1, ...50$.[4]

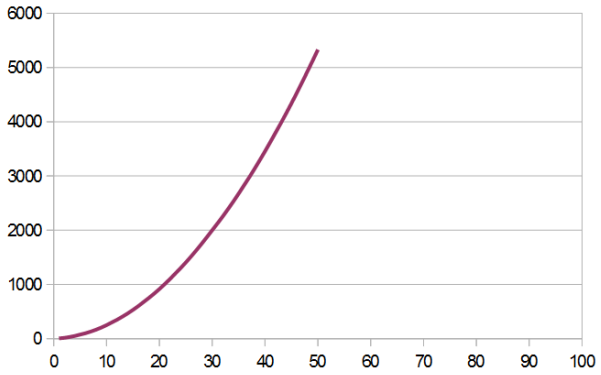

Figura 3.8: Grafico non monometrico di $f_5(n)$ per $n = 1, ...50$.

[4]I diagrammi a dispersione "solo linee" riportati sono stati costruiti con il software OpenOffice Calc con procedimenti del tutto identici a quelli utilizzati per f_2 e f_3. Abbiamo scelto di riportare in questa fase solo i grafici non monometrici, poiché si è visto che essi evidenziano meglio l'andamento parabolico delle funzioni $f_p(n)$.

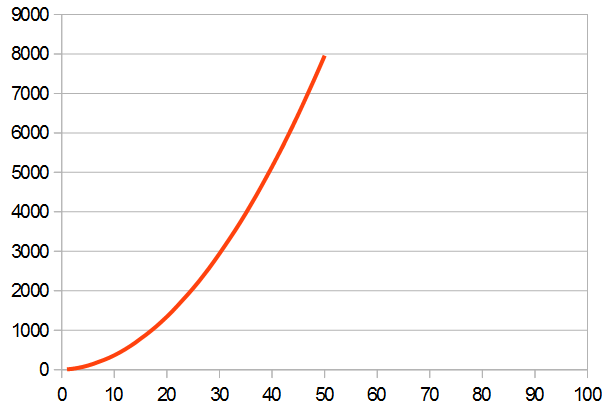

Figura 3.9: Grafico non monometrico di $f_7(n)$ per $n = 1, ...50$.

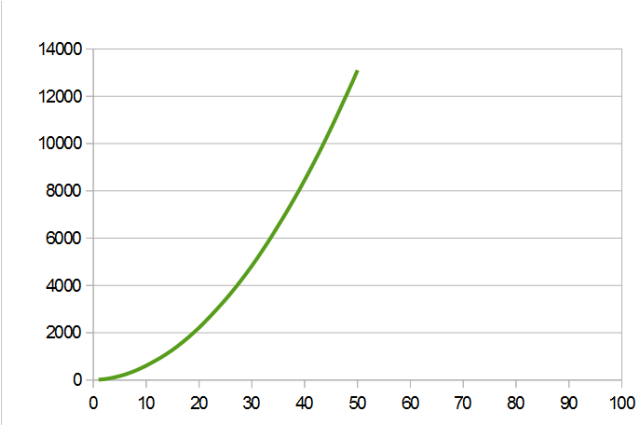

Figura 3.10: Grafico non monometrico di $f_{11}(n)$ per $n = 1, ...50$.

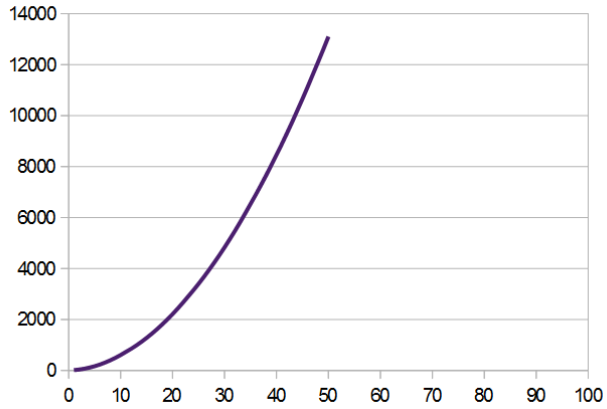

Figura 3.11: Grafico non monometrico di $f_{13}(n)$ per $n = 1, ...50$.

Dall'analisi dei grafici sembra ragionevole supporre che l'andamento sia parabolico per qualsiasi scelta di p numero primo.
Arrivati a questo punto, sarebbe interessante scoprire se esiste una relazione tra il valore (presunto) del coefficiente a della parabola di regressione e il numero primo p oggetto di studio. Per farlo calcoliamo, in maniera del tutto analoga a quanto fatto per le potenze di 2, i coefficienti delle parabole $\mathscr{P}_5$, $\mathscr{P}_7$, $\mathscr{P}_{11}$ e $\mathscr{P}_{13}$.
I valori presunti delle aperture delle parabole di regressione saranno:

- $a_{(5)} = 2$;
- $a_{(7)} = 3$;
- $a_{(11)} = 5$;
- $a_{(13)} = 6$.

All'apparenza vi è una certa regolarità nei risultati di questi calcoli. Si potrebbe ipotizzare che il valore dell'apertura della parabola a sia legato alla distanza tra un numero primo e il suo precedente.

Elaboriamo allora la seguente congettura:
Data la funzione f_p, dove p è un numero intero, definita secondo la congettura sulle differenze prime, l'apertura della parabola di regressione ottenuta per $n \longrightarrow +\infty$ è:

$$a_{(p)} = a_{(p')} + \frac{p - p'}{2} \tag{3.5.1}$$

$\forall p$ primo, dove $a_{(2)} = \frac{1}{2}$ e p' è il numero primo precedente a p.
Effettuiamo qualche altra prova a supporto della nostra teoria, calcolando l'apertura della parabola di regressione $\mathscr{P}_p$ per $p = 17, 19, 23, 29, 31$:[5]

- $a_{(17)} = 8$;
- $a_{(19)} = 9$;
- $a_{(23)} = 11$;
- $a_{(29)} = 14$;
- $a_{(31)} = 15$.

I test effettuati per definire la nostra congettura hanno dato esito positivo. Naturalmente produrre qualche esempio non equivale ad una prova rigorosa, ma questo esperimento si può considerare come il trampolino di lancio per approdare, forse, in futuro, alla dimostrazione vera e propria.

[5]Si può verificare attraverso i grafici a dispersione che anche le funzioni $f_{17}(n)$, $f_{19}(n)$, $f_{23}(n)$, $f_{29}(n)$ e $f_{31}(n)$ hanno un andamento simile a quello di una parabola.

Appendice A

Piccolo teorema di Fermat

A.1 Breve biografia di Pierre de Fermat

Pierre de Fermat nasce nel 1601 a Beaumont-de-Lomagne, nel sud della Francia. Suo padre è un ricco commerciante e un influente uomo politico. Non c'è da stupirsi dunque che il figlio venga indirizzato agli studi giuridici e diventi un brillante avvocato. Quello che invece può legittimamente stupire è che Fermat sia ricordato come uno dei più illustri matematici della storia.

Figura A.1: Pierre de Fermat (1601-1665).

I suoi primi contatti con la matematica risalgono alla seconda metà degli anni Venti, quando si trasferisce per lavoro a Bordeaux. Qui conosce il matematico Jean Beaugrand e inizia una fitta collaborazione epistolare con un gruppo di matematici parigini. Queste lettere sono, insieme alle note a margine dei libri che consultava, tutto ciò che ci rimane degli studi di Fermat, che infatti non aveva nessuna intenzione di descrivere le sue scoperte in maniera dettagliata. Se vogliamo, egli preferiva "sfidare" matematici celebri a raggiungere risultati che lui aveva già prodotto (confutò persino alcune affermazioni dell'intoccabile Cartesio). Il caso più noto è senz'altro l'ultimo teorema di Fermat che venne dimostrato solo nella seconda metà del Novecento da Andrew Wiles: 300 anni di tentativi falliti, che hanno però prodotto buona parte della teoria sugli anelli che oggi conosciamo.

A.2 Enunciato e dimostrazione

Gli studi di Fermat si concentrarono principalmente sulla teoria dei numeri. Ciò che a noi interessa in prevalenza è il **piccolo teorema di Fermat**, uno dei più importanti risultati sui numeri primi.
Vediamolo nella sua prima formulazione che corrisponde pressapoco a quanto enunciato da Fermat negli anni Trenta del Seicento, fatto salvo che in 300 anni la simbologia si è evoluta in maniera notevole e che all'epoca di Fermat i teoremi erano ancora espressi in linguaggio "naturale".

Teorema A.2.1. (Piccolo teorema di Fermat (1)). *Se p è un numero primo e a è un intero che non è multiplo di p, allora $a^{p-1} \equiv 1 \mod p$.*

Dimostrazione. Sia $k \in \mathbb{Z}$ t.c. p non divida k. Allora p non divide neanche il prodotto di $a \cdot k$. Dunque, per il lemma di divisione applicato alle congruenze, esiste univocamente determinato $r \in \mathbb{Z}$, $0 < r < p$ tale che $ak \equiv r \mod p$.
Sappiamo che r così definito è diverso da 0 poiché p non divide ak. Allora, possiamo definire una funzione

$$\delta : \{1, 2, 3, ..., p-1\} \longrightarrow \{1, 2, 3, ..., p-1\}$$

che associa a $k \in \{1, 2, 3, ..., p-1\}$, il resto r della divisione di ak per p.
Dimostriamo che δ è biiettiva.

- δ è iniettiva
  Considero k e l due interi nell'insieme $\{1, 2, 3, ..., p-1\}$ tali che $\delta(k) = \delta(l)$. Allora necessariamente k e l hanno lo stesso resto modulo p e quindi $k \equiv l \mod p$. Siccome $ak \equiv al \mod p$ e $k \equiv l \mod p$, $(p, a) = 1$ e $l, k \leq p-1$ e allora $k = l$.

- δ è suriettiva
  δ è suriettiva poiché, per il lemma di divisione, esiste sempre il resto.

Ora consideriamo:

$$\begin{aligned} a^{p-1} \cdot (p-1)! &= a^{p-1} \cdot (p-1) \cdot (p-2) \cdot ... \cdot 2 \cdot 1 = \\ &= (a \cdot 1) \cdot (a \cdot 2) \cdot ... \cdot (a \cdot (p-1)) \equiv \\ &\equiv \delta(1) \cdot \delta(2) \cdot ... \cdot \delta(p-1) = (p-1)! \end{aligned}$$

Allora ne verrà che:

$$\begin{aligned} & a^{p-1} \cdot (p-1)! \equiv (p-1)! \mod p \\ & \implies (\text{siccome } (p, p-1) = 1)\ a^{p-1} \cdot (p-1)! \equiv (p-1)! \mod p \\ & \implies a^{p-1} \equiv 1 \mod p. \end{aligned} \tag{A.2.1}$$

□

Osservazione 4. Spesso il piccolo teorema di Fermat viene enunciato in una maniera alternativa (2): *se p è primo e $a \in \mathbb{Z}$ allora $a^p \equiv a \mod p$.*

Notiamo che questa formulazione (2) non necessita dell'ipotesi che p non divida a.

Dimostriamo l'equivalenza delle due affermazioni.

Dimostrazione.
(1) $\Longrightarrow$ (2)
Consideriamo p primo.

- Se p non divide a, allora l'asserto viene banalmente moltiplicando per a sia a destra sia a sinistra del segno $\equiv$.
- Se p divide a, il teorema dice che $a^p \equiv 0 \mod p$. Inoltre, per definizione di congruenza, sappiamo che $a \equiv 0 \mod p$. Allora, per la proprietà transitiva, $a^p \equiv a \mod p$.

(2) $\Longrightarrow$ (1)
Considero p primo tale che $a^p \equiv a \mod p$. Se p non divide a, allora $(a, p) = 1$ e dunque $a^{p-1} \equiv a^0 \mod p$, ossia $a^{p-1} \equiv 1 \mod p$. □

Appendice B

Interpolazione e regressione

In questo capitolo riassumiamo alcune nozioni di statistica e probabilità: in particolare ci occupiamo di richiamare i concetti di interpolazione e regressione, di riprendere il metodo dei minimi quadrati e il test di Pearson.

B.1 Introduzione al concetto di interpolazione statistica

Quando cerchiamo di studiare un fenomeno che coinvolge due grandezze attraverso prove sperimentali, può rivelarsi utile rappresentare i dati in nostro possesso in un diagramma. Il grafico che meglio si addice a questo procedimento è detto nuvola di punti o diagramma a dispersione e rappresenta in un piano cartesiano le coppie di valori del tipo (x_i, y_i), rilevate dalla nostra indagine.
Non è da escludere che i punti (x_i, y_i) si dispongano nel piano in maniera regolare (per esempio linearmente, o lungo il grafico di una funzione nota). In tal caso, vogliamo cercare una funzione $y = f(x)$, detta funzione interpolante, che interpreti il più correttamente possibile i dati sperimentali.

Le strade percorribili sono due:

- **interpolazione per punti noti**: si cerca una funzione che assuma esattamente tutti i valori rilevati. In altre parole si cerca una funzione tale che TUTTI i valori rilevati appartengano al suo grafico;
- **interpolazione statistica**: si cerca una funzione il cui grafico assuma valori prossimi a quelli rilevati nell'indagine sperimentale.

Il primo modo di procedere è senza dubbio da preferire, qualora sia possibile attuarlo. Esso si rivela però quasi sempre impossibile da attuare negli esperimenti reali, nonostante sia corretto e facilmente utilizzabile nei modelli matematici.
Ci occupiamo quindi di descrivere l'interpolazione statistica che, nonostante sia soggetta a errori, è lo strumento ideale per studiare un esperimento statistico.

B.2 Determinare la funzione interpolante

Il problema di determinare la funzione interpolante non ha una soluzione univoca e può rivelarsi complesso. Nel caso di un esperimento statistico, il miglior punto di partenza è senza dubbio quello di osservare il grafico a dispersione e cercare di ipotizzare quale sia l'andamento dei punti: lineare, parabolico, sinusoidale, ecc.
A questo punto, stabilito quale funzione ci sembra essere adatta ad interpolare i punti, dovremo determinarne una forma canonica, dipendente da un certo numero di parametri. Per esempio, se supponiamo che la funzione interpolante sia una retta, scriveremo

$$f(x) = mx + q$$

e cercheremo di determinare i parametri m e q sulla base dei dati in nostro possesso. Analogamente se ipotizziamo che la funzione interpolante sia una parabola, scriveremo $y = ax^2 + bx + c$ e determineremo a, b e c. E così via qualsiasi sia la funzione interpolante scelta.
Per determinare tali coefficienti si può utilizzare il metodo dei minimi quadrati: esso

consiste nella scelta come parametri di quei valori che minimizzano la funzione S così definita:

$$S(a, b, c...) = \sum_{i=1}^{n} (f(x_i) - y_i)^2 \tag{B.2.1}$$

dove y_i sono i valori rilevati nell'esperimento, $f(x_i)$ sono i valori teorici determinati dalla funzione interpolante ed S è una funzione che dipende solamente dai parametri che occorrono per definire la funzione (a, b, c, m, q,... ecc.).
Esaminiamo il caso particolare di una funzione lineare. Se la funzione interpolante è del tipo $y = mx + q$, allora la funzione S da minimizzare sarà:

$$S(m, q) = \sum_{i=1}^{n} (f(x_i) - y_i)^2 = \sum_{i=1}^{n} (mx + q - y_i)^2. \tag{B.2.2}$$

Svolgendo i calcoli e ordinando rispetto a q si ottiene:

$$S(m, q) = nq^2 + 2q(mn\overline{x} - n\overline{y}) + m^2 \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2 - 2m \sum_{i=1}^{n} x_i y_i$$

dove $\overline{x}$ e $\overline{y}$ sono i valori medi rispettivamente di x_i e y_i.
Questa è una parabola con la concavità rivolta verso l'alto (infatti il coefficiente di q^2 è $n > 0$) che dunque ha minimo nel vertice:

$$q = \frac{-2(mn\overline{x} - n\overline{y})}{2n} = \overline{y} - m\overline{x}. \tag{B.2.3}$$

Allo stesso modo, ordinando rispetto a m si ottiene una parabola:

$$S(m, q) = m^2 \sum_{i=1}^{n} x_i^2 + 2m(qn\overline{x} - m \sum_{i=1}^{n} x_i y_i) + (nq^2 + \sum_{i=1}^{n} y_i^2 - 2qn\overline{y}). \tag{B.2.4}$$

Essa ha minimo in:

$$m = \frac{-2(qn\overline{x} - \sum_{i=1}^{n} x_i y_i)}{2 \sum_{i=1}^{n} x_i^2} = \frac{\sum_{i=1}^{n} x_i y_i - qn\overline{x}}{\sum_{i=1}^{n} x_i^2}. \tag{B.2.5}$$

Sostituendo l'espressione di q nell'espressione di m e sviluppando i calcoli, otteniamo le seguenti formule per calcolare i coefficienti della retta di regressione attraverso il metodo dei minimi quadrati:

$$q = \overline{y} - m\overline{x}, \qquad m = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \cdot \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}. \tag{B.2.6}$$

Il procedimento per calcolare i coefficienti della funzione parabolica $y = ax^2 + bx + c$ è analogo, seppur con maggiori difficoltà di calcolo. Occorre infatti calcolare e porre uguali a zero le derivate parziali rispetto alle variabili a, b e c.
Ciò che si ottiene è il sistema seguente, lineare nelle variabili a, b e c:

$$\begin{cases} c \cdot N + b \cdot \sum x_i + a \cdot \sum x_i^2 = \sum y_i \\ c \cdot \sum x_i + b \cdot \sum x_i^2 + a \cdot \sum x_i^3 = \sum x_i y_i \\ c \cdot \sum x_i^2 + b \cdot \sum x_i^3 + a \cdot \sum x_i^4 = \sum x_i^2 y_i \end{cases} . \tag{B.2.7}$$

B.3 Test di Pearson

Una volta determinata la funzione interpolante $y = f(x)$ occorre effettuare alcuni test per verificare di aver scelto un'applicazione che approssimi correttamente i dati sperimentali. Definiamo innanzitutto gli errori di accostamento.

Definizione B.1. Siano (x_i, y_i) le coppie di dati sperimentali e siano invece $(x_i, f(x_i))$ i valori teorici determinati dalla funzione interpolante $y = f(x)$, per $i = 1, ...n$. Definiamo:

- $e_i = y_i - f(x_i)$ l'i-esimo errore di accostamento o i-esima differenza tra il valore sperimentale e il valore teorico;
- $e = \sum_{i=1}^{n} e_i$ l'errore assoluto o errore totale.

Gli errori parziali, ossia le i-esime differenze, sono fondamentali da studiare: infatti, potrebbe accadere che l'errore totale valga 0, inducendo a pensare che l'interpolazione non sia soggetta ad errori, quando invece essi si sono semplicemente "compensati" tra loro.

Tali indici però non sono gli unici in grado di determinare "quanto è buona l'approssimazione". Vi sono moltissimi valori che, se calcolati, possono fornirci un'informazione di questo tipo: noi abbiamo scelto di utilizzare il test di Pearson.
Il test di Pearson, considerati due vettori di dati, ne misura la correlazione lineare calcolando il coefficiente di correlazione lineare di Bravais-Pearson, così definito:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{B.3.1}$$

dove $-1 \leq r \leq 1$. Per i nostri studi possiamo considerarne il valore assoluto:

$$R = |r|. \tag{B.3.2}$$

Se il valore di R fosse uguale a 1 si avrebbe correlazione perfetta e questo garantirebbe l'eccellenza dell'approssimazione. Possiamo considerare abbastanza efficace un'approssimazione che produce un coefficiente $R > 0,75$. Naturalmente, più R si avvicina al valore 1 e più la funzione interpolerà bene i dati sperimentali.

Appendice C

Programmare con Python®

In questa appendice intendiamo presentare rapidamente il funzionamento della macchina Python® e fornire al lettore i soli strumenti di programmazione necessari per comprendere questa ricerca. Questo capitolo non può e non vuole insegnare al lettore l'arte della programmazione in Python® e potrebbe non essere esaustivo. Per trattazioni più complete e sistematiche si rimanda alla rete che, come possiamo immaginare, si rivela essere la fonte di informazione più ricca, anche grazie alla collaborazione tra utenti e programmatori che avviene sui forum di discussione.[1]

C.1 Introduzione informale

Python® è un linguaggio di programmazione ad alto livello progettato all'inizio degli anni Novanta dall'informatico olandese Guido van Rossum. Python®, il cui nome deriva da quello del gruppo comico-intellettuale Monty Python, ha ottenuto in breve tempo un apprezzamento notevole dalla comunità informatica e in particolare dagli sviluppatori di software professionisti. Una delle ragioni di tale successo è l'avere molte caratteristiche in comune con linguaggi universalmente utilizzati come Java e C++. Ma c'è dell'altro:

[1]Sono disponibili due versioni di Python®: Python 2 e Python 3. Le due versioni presentano lievi differenze: noi abbiamo scelto di lavorare con la 2.7.8. Se il lettore utilizzerà versioni diverse di Python® potrebbe trovare qualche incongruenza.

Python® è semplice! O, quantomeno, più semplice della maggior parte dei linguaggi di programmazione a così alto livello. E questo non ha fatto che aumentarne i consensi. Emblematica è sicuramente una frase dell'insegnante e programmatore Jeff Elkner: "*I see a higher general level of success and a lower level of frustration than I experienced during the two years I taught C++*", ossia "Ho visto un livello generale di successo più elevato e un minor livello di frustrazione, rispetto all'esperienza maturata durante i due anni di insegnamento di C++".

Come per un matematico, il lavoro dell'informatico consiste nel risolvere problemi e utilizzare un linguaggio formale per presentare i risultati. Andando nel dettaglio, programmare (con Python® o con qualsiasi altro linguaggio ad alto livello) significa fornire al computer una serie di istruzioni, dette *codice sorgente*, che vengono poi elaborate da un "interprete intelligente" (il computer), il quale restituisce uno o più risultati.
Un programma è una sequenza di istruzioni, delle quali diamo un assaggio nella tabella seguente:

Istruzione	**Descrizione**	**Esempi**
input	ricezione dei dati dall'esterno (tipicamente da tastiera)	`raw.input('Dai un numero')`
output	scrittura dei dati su file	`print x`

istruzioni matematiche	esecuzione di operazioni matematiche elementari	`2+7`
		`3*8`
		`3**2`
		`4//2`

istruzioni condizionali	controllo di alcune condizioni ed esecuzione	`if(x>5): print x`
ripetizione	ripetizione di un'azione	`while(x>0): x=x-1`

Tabella C.1: Descrizione delle istruzioni fondamentali in Python®.

Le istruzioni elencate in questa tabella possono essere considerate i mattoncini con i quali sono costituiti tutti i programmi più complessi. Possiamo quindi affermare che la programmazione consiste proprio nel suddividere compiti complessi in altri più piccoli, che possono essere completamente ricondotti a queste istruzioni fondamentali (si tenga presente che sono univoche le istruzioni, NON gli esempi indicati nell'ultima colonna della tabella).

Per scrivere un programma non occorre nessun tipo di inizializzazione: le uniche cose che vanno segnalate nel preambolo sono le importazioni dei moduli, effettuate attraverso il comando `import` seguito dalla parola riservata che caratterizza il modulo (come per esempio `math`).

C.2 Variabili e istruzioni

Le variabili, in informatica come in matematica, possono essere considerate come scatole vuote alle quali viene associato un valore (un numero, una lettera, una parola, una matrice...).
Vi sono diversi **tipi** di valori (numeri interi, numeri in virgola mobile, stringhe di caratteri...), ma una delle grandi comodità del linguaggio Python® è quella di non dover definire a priori di che tipo sarà il valore che assoceremo ad una variabile. Quindi, alla variabile denominata `x`, possiamo associare tranquillamente un valore numerico prima e una stringa poi.
L'operazione di **assegnazione** avviene attraverso il simbolo `=` (uguale).
Dunque, le istruzioni

```
x=4
y=5
```

creano le variabili `x` ed `y` e vi associano rispettivamente il numero 4 e il numero 5.

Ad una variabile può essere associato anche il valore corrispondente ad un'espressione matematica:[2]

```
x=4+6*8-9//3
```

Alla variabile `x` viene associato il valore 49, risultato dell'operazione indicata (si tenga presente che Python® è in grado di risolvere le operazioni nell'ordine matematicamente corretto e non nell'ordine in cui le "legge").
Quando si crea un'assegnazione di questo tipo, nessun output viene però comunicato allo stato esterno, ma si crea un'associazione che rimane racchiusa nello stato interno della macchina. Per trasferire un'informazione allo stato esterno si utilizza il comando `print`:

```
x=4+6*8-9//3
print x
```

Sullo stato esterno (a video) comparirà il numero 49.
Come già accennato, esistono innumerevoli tipi, ma noi ci occuperemo solamente del tipo `int`, ossia dei numeri interi.

C.3 Funzioni in Python®

Python® possiede una serie di funzioni matematiche predefinite, come `sin`, `log`, `sqrt`, contenute nel modulo `math`, che va importato all'inizio della stesura del programma con il comando `import math`.
Per tutte le funzioni che non sono predefinite, occorre invece che il programmatore le definisca autonomamente, attraverso la seguente modalità:

```
def NOME_FUNZIONE(PARAMETRI_SEPARATI_DA_VIRGOLE):
    ISTRUZIONI
```

Il nome della funzione può essere una parola qualsiasi, ad esclusione di quelle "riservate", ossia già utilizzate da Python® per funzioni predefinite.

[2] Il simbolo `//` rappresenta la divisione tra numeri interi, che quindi dà come risultato il quoziente intero e non il risultato decimale della divisione.

La funzione deve restituire uno o più valori: per comunicare alla macchina qual è il valore di ritorno occorre usare il comando `return`. Quando l'esecutore incontra questo comando, termina l'esecuzione della funzione ed ignora qualsiasi eventuale istruzione successiva.
Così, se desideriamo creare una funzione che sommi due numeri interi presi come parametro, scriveremo:

```
def somma(x,y):
    s=x+y
    return s
```

Una volta definite le funzioni che ci occorrono per completare il nostro programma, possiamo richiamarle nel programma principale (a patto di averle precedentemente definite nello stesso file .py).
L'utilizzo delle funzioni permette una maggiore leggibilità del programma, agevola i test (si possono infatti testare separatamente le funzioni e verificare la presenza di eventuali errori) e accorcia sensibilmente il programma, eliminando le parti ripetitive.

C.4 Ciclo for e ciclo while

Durante la stesura di un programma, alcune istruzioni necessitano di essere ripetute per un certo numero di volte, tipicamente finché una condizione iniziale non viene meno. Per svolgere questo tipo di iterazione si può utilizzare l'istruzione `while`. Al `while` è associata una condizione e dunque esso si presenta in questo modo:

```
while (CONDIZIONE):
   ISTRUZIONI
```

Ciò che fa il comando `while` è ripetere le istruzioni finché non viene meno la condizione. Quest'ultima deve essere un'espressione booleana (che può assumere valore `True` oppure `False`): finchè essa risulta vera, le istruzioni vengono ripetute.
Le istruzioni sono poste rientrate rispetto alla scritta `while`. In questo modo non occorre

comunicare al programma la "fine" del ciclo, ma è sufficiente tornare a scrivere a margine sinistro una volta terminato l'elenco delle istruzioni che il programma deve ripetere.
Per evitare che il programma continui all'infinito, le istruzioni devono contenere dei passaggi che modifichino la condizione iniziale e che, dopo un numero finito di step, la rendano falsa.
Consideriamo, per esempio, il seguente programma:

```
x=1
t=9
while (x>0):
    print t
    t=t+1
```

Questo frammento di programma è corretto da un punto di vista sintattico, ma diverge, poiché la condizione `x>0` non viene mai meno. Vengono quindi stampati all'infinito tutti i numeri naturali da 9 in poi.
Vediamo come possiamo migliorarlo:

```
x=5
t=9
while (x>0):
    print t
    t=t+1
    x=x-1
```

Questo programma stampa, nell'ordine: 9, 10, 11, 12, 13. Infatti, la variabile `x` viene diminuita di 1 ad ogni ripetizione e quindi, dopo 5 ripetizioni, non soddisfa più la condizione booleana.

Oltre al ciclo `while`, per ripetere certe istruzioni si può utilizzare il comando `for`. In particolare ci interessa mostrare cosa accade quando scriviamo in un programma:

```
for i in range (PRIMO_ESTREMO,SECONDO_ESTREMO):
    ISTRUZIONI
```

La traduzione di quanto appena scritto in linguaggio naturale è: "svolgi le ISTRUZIONI dando ad `i`, nell'ordine, i valori interi compresi nell'intervallo". Il primo estremo è compreso nell'intervallo, mentre il secondo no.
Facciamo un esempio:

```
for i in range (0,5):
    print i
```

Questo programma stampa tutti i numeri interi da 0 a 4: 0, 1, 2, 3, 4.

Le informazioni date, anche se frammentarie, dovrebbero aver fornito al lettore tutte le nozioni fondamentali per comprendere la creazione del programma necessario in questa ricerca.

Epilogo

Molte informazioni di tipo informatico sono legate al sistema operativo utilizzato e alla potenza della macchina. Abbiamo lavorato su una macchina HP® sulla quale è installato il sistema operativo Windows 8.1®. Dunque, le informazioni che forniamo potrebbero non essere coerenti con sistemi operativi e macchine diverse.

Vista la grande quantità di materiale reperibile in rete, abbiamo preferito adottare, per la nostra ricerca, una ferrea politica "software free". Infatti, riteniamo che certe ricerche debbano essere fruibili da tutti e non subordinate all'acquisto di software di calcolo spesso molto dispendiosi. Questo è il motivo per cui abbiamo scelto di utilizzare Apache® OpenOffice Calc invece di Microsoft® Excel. OpenOffice Calc fa parte della suite per ufficio Apache® OpenOffice, distribuita con licenza libera e Open Source.
Le licenze Open Source indicano che gli autori del programma hanno reso pubblici i files sorgente, permettendo anche a programmatori indipendenti di apportarvi modifiche e di studiarne il funzionamento.
Anche il linguaggio di programmazione ad alto livello Python® e il programma che occorre per farlo funzionare sono disponibili gratuitamente in rete. Infatti, sul sito web ufficiale si leggono le testuali parole: *All Python® releases are Open Source.* Questo è uno dei motivi principali per cui abbiamo scelto, tra tanti linguaggi di programmazione, di utilizzare proprio Python®. Naturalmente vi sono anche altre ragioni: prima tra tutte il fatto che Python® è un linguaggio "relativamente semplice", che consente di costruire programmi funzionanti e ben strutturati anche dopo solo poche ore di studio. Abbiamo quindi ritenuto che questa scelta rendesse agevole la comprensione del testo anche per un informatico alle prime armi o addirittura per un profano.

Bibliografia

[1] M. Artin, *Algebra*, Bollati Boringheri, 1997.

[2] E. Bedocchi, *Esercizi di algebra*, Pitagora Editrice, 1995/96.

[3] I. N. Herstein, *Algebra*, Editori Riuniti, 1982.

[4] A.Vistoli, *Note di algebra*, Bologna 1993/94.

[5] M. Accogli, *Polinomi e funzioni polinomiali negli anelli* $\mathbb{Z}_m$, Tesi di Laurea Triennale in Matematica, Bologna, 2007.

[6] R. Mazzone, *Algebre monounarie polinomiali*, Tesi di Laurea Magistrale in Matematica, Bologna, 2012.

[7] [Open Book] A. Downey, J. Elkner, C. Meyers, *How to think like a computer scientist - Learning with Python*, Edited by Shannon Turlington and Lisa Cutler, 2002.
La traduzione italiana di questo libro è disponibile gratuitamente all'indirizzo web: *http://www.python.it/doc/Howtothink/Howtothink-html-it/index.htm*

[8] [Articolo] O. Muscato *Metodi Matematici e Statistici*
Dispensa reperibile gratuitamente sul sito internet dell'Università degli studi di Catania: *http://www.dmi.unict.it/ muscato/MMStat.pdf*

[9] M. Bergamini, A. Trifone, G. Barozzi, *Matematica.blu 2.0*, Zanichelli, 2011.

Ringraziamenti

Ringrazio prima di tutti il mio relatore, il professor Libero Verardi, per avermi seguito con grande precisione e pazienza durante la stesura della tesi.
Un pensiero grato va al professor Simone Martini, che ha corretto e commentato la parte strettamente informatica di questa tesi, e al professor Marco Lenci che mi ha suggerito l'uso del test di Pearson per determinare il coefficiente di correlazione.
Ringrazio la dottoressa Giulia Laffi e con lei tutta la redazione di matematica: in particolare Marinella Lombardi, Fabio Bettani e Roberta Maroni che hanno affrontato questa avventura insieme a me.
Ringrazio la Cate che, nonostante sia una viola, è stata da subito ben più che una collega per me, la Fra, compagna eccezionale e amica insostituibile, e la Corimbi che sono vent'anni che mescola la sua vita alla mia.
Un grazie speciale va a tutta la mia splendida famiglia, che non ha mai avuto dubbi anche quando io ne avevo tantissimi: papà, la bis e la mia mamma, che è stata il miglior correttore di bozze che potevo sperare di assumere.
E per finire ringrazio mia figlia perché con lei è sempre tutto più bello.