

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Scuola di Scienze

Corso di Laurea triennale in Fisica

**Studio della correlazione tra il drogaggio
del silicio e l'energia superficiale del suo
ossido nativo**

Relatore:

Prof. Cristiano Albonetti

Presentata da:

Michele Della Ciana

Correlatore:

Prof. Beatrice Fraboni

Sessione II

Anno Accademico 2015

Indice

Introduzione	3
1 Ossido di silicio	6
1.1 Ossido Nativo	8
1.2 Ricostruzione dell'ossido nativo	9
2 Interpretazione dell'angolo di contatto	11
2.1 Teoria dell'angolo di contatto statico	11
2.2 Teoria dell'isteresi	15
3 Metodi Sperimentali	17
3.1 Preparazione dei campioni	17
3.2 Liquidi utilizzati nelle misure di angolo di contatto statico	19
3.3 Pulizia dei campioni	20
3.4 Strumentazione	21
3.5 Software Digidrop	24
4 Elaborazione dati	27
4.1 Angolo Statico	27
4.1.1 Calcolo dell'energia superficiale	28
4.2 Angolo Dinamico/Isteresi	30
5 Misure ed analisi dati	34
5.1 Misure sui substrati <i>as received</i>	34
5.2 Misure sui substrati dopo 7 giorni dalla rimozione dello strato di ossido .	37
5.3 Misure sui substrati dopo 21 giorni dalla rimozione dello strato di ossido	40
5.4 Discussione	44
6 Simulazione numerica di una goccia	48
6.1 Legge di Young	48
6.2 Simulazione	50
6.3 Stima dell'energia di Gibbs al variare dell'angolo di contatto	53

Conclusioni	56
A Codice dei programmi Utilizzati	58
A.1 Analisi angolo statico	58
A.2 <i>Fit</i> Owen-Wendt	62
A.3 Analisi angoli di isteresi	65
A.4 Simulazione numerica di una goccia su una superficie	69
A.5 Programma per la realizzazione dei grafici	81
Bibliografia	86
Ringraziamenti	88

Introduzione

Lo studio delle superfici é stato affrontato in maniera approfondita solo negli ultimi decenni, cioè soltanto quando ha iniziato a ricoprire un ruolo importante sia nella fisica sia nella chimica.

Il motivo é da ricercare nell'importanza pratica che lo studio delle superfici ha in quanto disciplina trasversale in molti campi quali, ad esempio, l'elettronica e le scienze applicate. I materiali piú studiati nella fisica delle superfici sono il silicio e i suoi composti, perché sono alla base di tutti gli apparecchi elettronici attuali.

In questa tesi verrà analizzata la relazione esistente fra la concentrazione di drogante di un substrato di silicio con l'energia superficiale del suo ossido nativo.

Con il termine drogaggio, si intende l'aggiunta al semiconduttore puro di piccole percentuali di atomi non facenti parte del semiconduttore stesso, allo scopo di modificare le proprietà elettroniche del materiale, come ad esempio la resistività .

Il drogaggio può essere di due tipi:

- **Tipo n** , dove l'atomo di impurità ha un elettrone in piú di quelli che servono per soddisfare i legami del reticolo cristallino.
- **Tipo p** , dove l'atomo di impurità ha un elettrone in meno di quelli che servono per soddisfare i legami del reticolo cristallino

Un atomo drogante di tipo n aggiunge al cristallo un elettrone libero di muoversi, mentre un atomo drogante di tipo p forma una lacuna, che é assimilabile a una carica positiva libera di muoversi.

Prima di procedere, é opportuno precisare che nella meccanica dei materiali la tensione superficiale, o energia superficiale (indicata nel testo con la lettera γ), é la densità superficiale di energia di legame sull'interfaccia tra una superficie solida e un materiale di un'altra natura come, ad esempio, un liquido o un gas. Non coincide, quindi, con la tensione definita come densità superficiale di forza.

Nel Sistema internazionale si misura in Newton su Metri (N/m), ma solitamente si utilizza il *dyne/cm*. Dal punto di vista termodinamico può essere definita come il lavoro necessario per aumentare la superficie del continuo di una quantità unitaria.

In questa tesi, per la misura della energia superficiale, si userà il metodo di Owen-Wendt che sfrutta l'angolo di contatto statico.

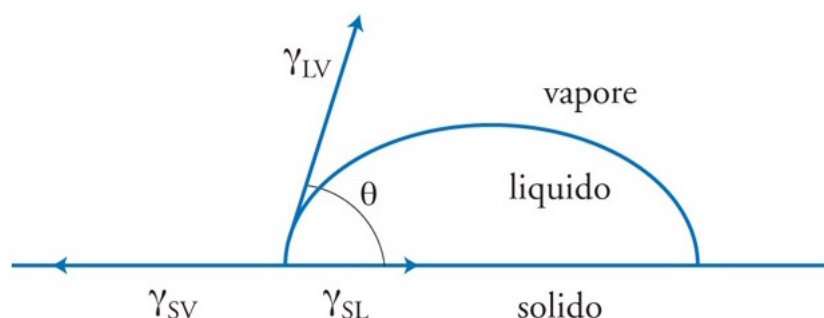


Figura 1: Angolo di contatto formato da una goccia liquida su una superficie solida

L'angolo di contatto statico (Figura 1) è una grandezza termodinamica descritta dall'angolo formato dall'incontro di un'interfaccia liquido-vapore con un'interfaccia liquido-solido in equilibrio e definisce la bagnabilità della superficie. Per convenzione si definiscono idrofobiche le superfici aventi un angolo di contatto con l'acqua maggiore di 90 gradi e idrofiliche le superfici con angoli minori di 90 gradi.

La relazione fondamentale che lega l'energia superficiale con l'angolo di contatto θ è l'equazione di Young [29] :

$$\gamma_{sv} - \gamma_{sl} = \gamma_{lv} \cos \theta$$

dove γ_{sv} , γ_{sl} , γ_{lv} sono rispettivamente le energie superficiali fra solido-vapore, solido-liquido, liquido-vapore.

Il metodo più semplice e utilizzato per la misura dell'angolo di contatto statico è il metodo goniometrico o della goccia sessile.

Le misure di angoli riportate in questa tesi sono eseguite utilizzando questo metodo.

Un altro valore che sarà usato nella classificazione dei substrati è la misura dell'isteresi dell'angolo di contatto dinamico; essa è definita come la differenza fra l'angolo di contatto di avanzamento e quello di recessione. Gli angoli di avanzamento e recessione sono rispettivamente gli angoli formati da una goccia quando avanza e indietreggia su una superficie solida. [27][30]

La misura dell'isteresi è stata effettuata utilizzando il metodo della goccia sessile dinamica, che consiste nel misurare la variazione dell'angolo di contatto mentre si aumenta

e si diminuisce il volume di una goccia.

I campioni con differente drogaggio saranno analizzati in 3 modi:

1. con l'ossido nativo originale, cioè come ricevuti (*as received*).
2. con l'ossido riformatosi in 7 giorni dopo un lavaggio in acido fluoridrico.
3. con l'ossido riformatosi in 21 giorni dopo un lavaggio in acido fluoridrico.

In questo modo si vede se la storia dell'ossido influisce sulle misure.

Capitolo 1

Ossido di silicio

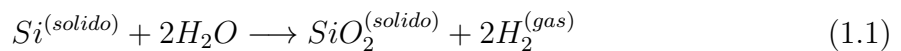
Quando un wafer di silicio entra in contatto con un agente ossidante come l'ossigeno o l'acqua, sulla superficie si forma un sottile strato di ossido di silicio amorfo. Il principale tipo di ossido che si forma é il biossido di silicio (SiO_2), che é un composto molto stabile a pressione atmosferica e temperatura ambiente.

La formazione dell'ossido di silicio é largamente influenzata sia dall'ambiente sia dal tipo di substrato su cui si forma.

Le condizioni ambientali che principalmente influiscono sulla formazione dell'ossido sono la temperatura e l'umidità relativa. Al crescere della temperatura e dell'umidità aumentano sia la velocità di formazione sia lo spessore dell'ossido [23].

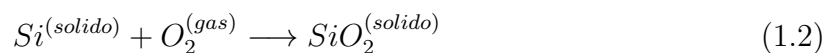
L'ossido che si forma ad alta temperatura ($600^\circ - 1000^\circ$) prende il nome di ossido termico e può essere o *wet* o *dry*, a seconda che l'agente ossidante sia H_2O o O_2 [18].

La reazione con l'acqua:



é considerata di qualità inferiore. Questo perché la reazione 1.1 porta alla liberazione di idrogeno gassoso, che, quando attraversa lo spessore di ossido già formato, reagendo con ioni O_2^- , forma gruppi idrossilici OH .

Questo rende la composizione dell'ossido meno omogenea. É quindi da preferirsi la reazione con l'ossigeno:



nonostante sia piú lenta.

Le caratteristiche del substrato che principalmente influiscono sulla crescita dell'ossido sono la concentrazione e il tipo di atomo drogante.

Sperimentalmente si vede che grandi concentrazioni di drogante ($\sim 10^{20} \text{ atomi/cm}^3$) aumentano la velocità di crescita e lo spessore dell'ossido [17]. Ciò é dovuto a due motivi:

- le impurità creano difetti nel reticolo del cristallo che é quindi piú facilmente ossidabile [17].

- parte del drogante diffonde nell'ossido, riducendone la densità e favorendo quindi la diffusione di ossidante attraverso l'ossido [17].

Il modo in cui diffonde il drogante nell'ossido di silicio dipende, oltre che dalla concentrazione, anche dal tipo di drogante. Si definisce il coefficiente di segregazione m , come il rapporto fra la solubilità nel Si con la solubilità nel SiO_2 del materiale drogante.

$$m = \frac{\text{solubilità in silicio}}{\text{solubilità in } SiO_2} \quad (1.3)$$

Se m è minore di 1, il drogante è più solubile nell'ossido.

Se m è maggiore di 1, il drogante è più solubile nel silicio.

In entrambi i casi si ha una redistribuzione della concentrazione di drogante nell'interfaccia che divide l'ossido dal silicio, redistribuzione dovuta alla diffusione di impurità da e verso l'ossido [17].

Boro	Fosforo	Antimonio	Arsenico	Gallio
0.1 – 0.3	10	10	10	20

Tabella 1.1: In tabella è riportato il valore di m per le principali specie droganti del silicio

Quando m è minore di 1 (es. Boro), le impurità nel silicio vicino all'interfaccia silicio-ossido tendono a diffondere all'interno dell'ossido, quindi la concentrazione di drogante nel silicio diminuisce a favore della concentrazione di drogante nell'ossido (Figura 1.1 A). Quando m è maggiore di 1 (es. Fosforo), si verifica il fenomeno contrario, le impurità presenti nell'ossido diffondono nel cristallo di silicio, aumentando la concentrazione di drogante nel silicio vicino all'interfaccia silicio-ossido (Figura 1.1 B).

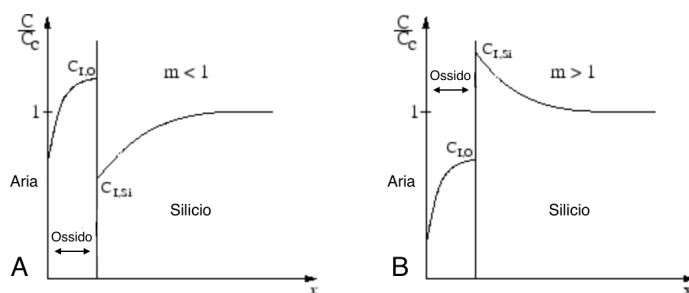
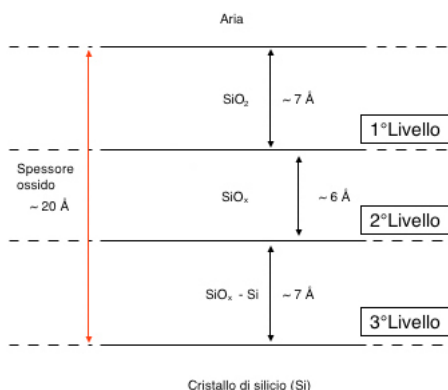


Figura 1.1: In figura è mostrata la redistribuzione relativa della concentrazione di drogante nell'interfaccia $SiO_2 - Si$. C_c indica la concentrazione di drogante prima della formazione dell'ossido, cioè all'interno del cristallo di silicio.

Dalla figura 1.1 si vede che, in entrambi i casi, nelle vicinanze dell'interfaccia aria-ossido la concentrazione di drogante cala drasticamente fino a un valore non nullo. Dai grafici in figura 1.1, si vede che questo valore é maggiore per $m < 1$ (es. Boro) rispetto a $m > 1$ (es Fosforo). Ci si aspetta, quindi, che esistano atomi di drogante in superficie che influiscano sull'angolo di contatto e quindi sull'energia superficiale. Inoltre, é ragionevole pensare che la percentuale di atomi superficiali possa dipendere anche dalla concentrazione di partenza C_c del silicio e non solo dal tipo di atomo drogante.

1.1 Ossido Nativo

Quando un wafer pulito é esposto all'atmosfera a temperatura ambiente, parte del silicio all'interfaccia si consuma, formando con l'ossigeno uno strato di ossido. Questo ossido prende il nome di ossido nativo e ha uno spessore compreso fra 12 e 30Å [8]. La composizione dell'ossido nativo é varia, infatti non é presente solo biossido di silicio SiO_2 , ma anche altri tipi di ossido SiO_x ($x < 2$). La sua struttura si può schematizzare in 3 livelli [8].



Il primo livello, il piú esterno, é sostanzialmente uno strato di biossido di silicio amorfo dello spessore di $\sim 7 \text{ \AA}$.

Il secondo livello é una miscela di vari ossidi SiO_x ($x < 2$) di $\sim 6 \text{ \AA}$ di spessore.

Il terzo livello é prevalentemente composto da atomi di Si e di SiO_x ($x < 2$) e separa l'ossido dal silicio cristallino sottostante. Lo spessore di questo strato é di $\sim 7 \text{ \AA}$.

In questa tesi tutte le misure sono state condotte su wafer con ossido di silicio nativo.

1.2 Ricostruzione dell'ossido nativo

Lo spessore dello strato di ossido nativo dipende molto dalle condizioni in cui esso si é formato, condizioni che determinano uno spessore variabile fra i 12 e i 30Å[8][23].

Per eliminare questa variabilit  si é misurata l'energia superficiale sia sui campioni *as received*¹, sia sui campioni con l'ossido ricostruito in 7 e 21 giorni.

Per ricostruire l'ossido, prima si é dovuto rimuovere lo strato gi  presente.

Per fare questa operazione si é utilizzata la seguente procedura [19][4], che prevede due differenti passaggi.

Il primo consiste nel rimuovere ogni traccia organica.

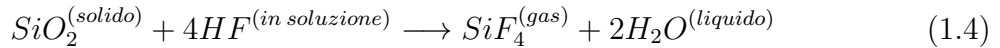
Per fare ci  si é utilizzata una soluzione composta da:

- 5 parti di acqua purificata (H_2O)
- 1 parte di idrossido di ammonio 27% (NH_4OH)
- 1 parte di perossido di idrogeno 30% (H_2O_2)

In questa fase il campione viene immerso nella soluzione per 15 minuti, poi viene lavato con 3 passaggi in acqua ultrapurificata.

Il secondo passaggio consiste nel rimuovere l'ossido con una soluzione di acido fluoridrico 2%.

La reazione é :



In questo stadio il campione viene immerso nella soluzione per 2 minuti, poi viene lavato con 3 passaggi in acqua ultrapura ($R \sim 18M\Omega$).

I campioni lavati vengono asciugati con un getto di azoto tecnico per ~ 30 secondi, poi sono riposti all'interno di un porta campioni in plastica in camera pulita (Figura 3.1).

Per formare un nuovo strato di ossido nativo, un campione di silicio trattato con HF impiega molto pi  tempo di un campione di silicio levigato [31]. Questo é dovuto al fatto che durante la procedura si forma uno strato molecolare di SiH , dovuto ai *dangling bonds*, ossia a delle valenze non soddisfatte negli atomi superficiali del cristallo. Affinch  l'ossido possa riformarsi, questi legami devono spezzarsi; a temperatura ambiente e pressione atmosferica il processo spontaneo attraverso cui questi legami si spezzano pu  richiederere da un'ora a una settimana e dipende dall'umidit  relativa [15] [22].

¹*As received* indica i wafer di silicio come sono stati ricevuti, ossia con l'ossido formatosi al momento della fabbricazione e quindi esposti all'aria per anni.

Per questo motivo i campioni di wafer trattati con l'HF vengono riposti in camera pulita per circa $\sim 10^4$ min (7 giorni). In questo periodo di tempo secondo [8] si riforma uno strato di ossido con uno spessore compreso fra i $16 - 20\text{\AA}$, di cui ~ 6.7 Å di SiO_2 . [8][31]

Capitolo 2

Interpretazione dell'angolo di contatto

2.1 Teoria dell'angolo di contatto statico

Il calcolo dell'energia superficiale tramite l'angolo di contatto inizia con l'equazione di Young [29].

$$\gamma_{sv} - \gamma_{sl} = \gamma_{lv} \cos \theta \quad (2.1)$$

In questa equazione sono presenti quattro quantità: l'angolo di contatto θ , l'energia superficiale solido-vapore γ_{sv} , l'energia superficiale solido-liquido γ_{sl} e l'energia superficiale liquido-vapore γ_{lv} .

Di questi valori solo θ e γ_{lv} sono misurabili direttamente, mentre per trovare i valori di γ_{sv} e γ_{sl} servono altre informazioni.

L'approccio piú ovvio é quello di cercare un'altra relazione che leghi i 4 parametri dell'equazione di Young. Il sistema fra queste 2 equazioni risolverebbe dunque il problema.

Il primo metodo usato per la misura dell'energia superficiale di un solido con l'angolo di contatto fu proposto da Zisman e dai suoi collaboratori [12].

L'osservazione sperimentale fu la seguente: per un dato solido, la misura dell'angolo di contatto non varia casualmente con la variazione del liquido, piuttosto $\cos \theta$ cambia leggermente con la tensione superficiale del liquido γ_{lv} , seguendo un andamento che suggerisce una relazione lineare fra $\cos \theta$ e γ_{lv} .

Questo andamento lineare prosegue fino a $\cos \theta = 1$, a quel punto al decrescere di γ_{lv} il valore del coseno rimane costante uno.

Da queste osservazioni Zisman postuló che l'energia superficiale del solido fosse uguale alla tensione superficiale del liquido che per primo rendeva l'angolo nullo ($\cos \theta = 1$)

ovvero il punto in cui il grafico cambiava bruscamente regime [12].

Quando l'angolo di contatto é nullo significa che la goccia bagna completamente il substrato. Quindi, energeticamente, l'apporto dell'interfaccia solido-liquido all'energia totale é nullo (o negativo).

Andando a sostituire i valori $\cos \theta = 1$ e $\gamma_{ls} = 0$ all'interno dell'equazione di Young (eq. 2.1) si trova $\gamma_{lv} = \gamma_{sv}$, che é il valore cercato.

Per molti tipi di substrati però o non c'è, o non si ha a disposizione una varietà sufficientemente estesa di liquidi che permettano di vedere l'andamento osservato da Zisman; in particolare non si ha quasi mai un liquido con un $\gamma_{critico}$ che si pone nel punto in cui la curva cambia bruscamente regime.

Si può ovviare a questo limite e trovare l'energia superficiale del substrato, graficando il coseno in funzione di γ_{lv} e trovando poi la retta che meglio approssima l'andamento dei punti misurati (Figura 2.1) [12].

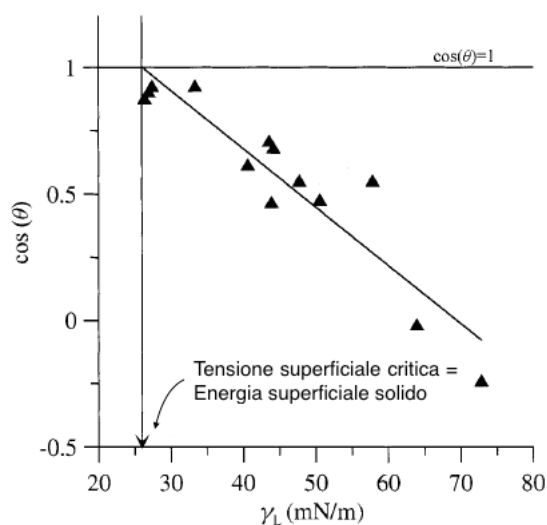


Figura 2.1: In figura é mostrato un esempio del grafico che si ottiene secondo il metodo di Zisman. Ogni punto rappresenta un liquido con γ_l decrescente da destra verso sinistra.

L'ascissa dove la retta assume valore 1 rappresenta l'ipotetica γ_{lv} di un liquido che bagnerebbe completamente il substrato, quindi la γ_{sv} del solido.

Una variante piú precisa, basata sempre sulle osservazioni di Zisman o piú propriamente sulla teoria a una componente dell'energia superficiale, é quella di approssimare i punti misurati non con una retta ma con la funzione (polinomiale) che meglio approssima il grafico e poi trovare l'intercetta con la retta $\cos \theta = 1$ [7].

Il metodo di Zisman é un metodo efficace per ricavare l'energia superficiale di molte superfici, tuttavia non é affidabile nel caso di liquidi e solidi polari. In questo caso bisogna utilizzare altri metodi.

In particolare in questa tesi viene utilizzato quello sviluppato da Owen e Wendt che prende il nome di metodo Owen-Wendt [24]. Sostanzialmente l'energia superficiale viene divisa in due componenti tali che :

$$\gamma = \gamma^d + \gamma^p$$

dove γ^d e γ^p prendono rispettivamente i nomi di componente dispersiva e componente polare.

La componente dispersiva rappresenta l'energia superficiale data dall'interazione molecolare dovuta alle forze di London¹, mentre la componente polare rappresenta l'energia superficiale data dal legame dipolo-dipolo² e dal legame a idrogeno³.

Viene poi postulata la seguente relazione:

$$\gamma_{sl} = \gamma_s + \gamma_l - 2\sqrt{\gamma_s^d \gamma_l^d} - 2\sqrt{\gamma_s^p \gamma_l^p} \quad (2.2)$$

Combinando questa equazione con l'equazione di Young eq.2.1 otteniamo:

$$\gamma_l(\cos \theta + 1) = 2\sqrt{\gamma_s^d \gamma_l^d} + 2\sqrt{\gamma_s^p \gamma_l^p} \quad (2.3)$$

Nell'equazione ottenuta sono presenti sei variabili, quattro delle quali misurabili direttamente, ossia $\cos \theta$, γ_l^d , γ_l^p , γ_l .

Per trovare i valori delle altre due sono necessarie almeno due equazioni, che si ottengono usando due liquidi diversi per lo stesso substrato.

Nella pratica però il metodo di Owen-Wendt non prevede l'utilizzo di due soli liquidi, ma di molti; infatti tutti i valori sono soggetti ad un margine di errore e piú liquidi permettono di avere una stima migliore di γ_s^p e γ_s^d . [24]

¹Le forze di London identificano tutte quelle forze che si presentano a livello atomico e molecolare e sono dovute a multipoli istantanei come risultato di effetti quantistici. Sono sempre presenti e spesso rappresentano una parte significativa della forza di interazione totale.

²Il legame dipolo-dipolo é l'interazione fra due molecole con dipoli permanenti. Un dipolo permanente si forma quando due atomi in una molecola hanno una elettronegativitá differente. L'atomo piú elettronegativo attrae gli elettroni di legame piú esterni dell'altro, acquisendo una parziale carica negativa e inducendo una parziale carica positiva.

³Il legame a idrogeno é un caso particolare di forza intermolecolare, in cui é implicato un atomo di idrogeno coinvolto in un legame covalente con elementi molto elettronegativi. Questi elementi attraggono a sé gli elettroni di valenza, acquisendo una parziale carica negativa e lasciando l'idrogeno con una parziale carica positiva. Il legame a idrogeno si forma quando la parziale carica positiva dell'idrogeno viene in contatto con un doppietto elettronico di un elemento fortemente elettronegativo, il quale lega l'H.

Consideriamo dunque l'eq. 2.3 e dividiamo entrambi i lati per $2\sqrt{\gamma_l^d}$:

$$\frac{\gamma_l(\cos \theta + 1)}{2\sqrt{\gamma_l^d}} = \frac{\sqrt{\gamma_s^p \gamma_l^p}}{\sqrt{\gamma_l^d}} + \sqrt{\gamma_s^d} \quad (2.4)$$

Sostituendo y a $\frac{\gamma_l(\cos \theta + 1)}{2\sqrt{\gamma_l^d}}$ e x a $\frac{\sqrt{\gamma_l^p}}{\sqrt{\gamma_l^d}}$ otteniamo l'equazione di una retta:

$$y = \sqrt{\gamma_s^p} x + \sqrt{\gamma_s^d} \quad (2.5)$$

La y e la x sono indipendenti dalle proprietà del substrato, quindi ogni liquido utilizzato individua un punto sulla retta.

Facendo la regressione lineare di questi punti otteniamo che il coefficiente angolare corrisponde alla radice quadrata della componente polare del solido, mentre l'intercetta con l'asse delle y corrisponde alla radice quadrata della componente dispersiva del solido.

Per ottenere i valori di γ_s^d e γ_s^p è quindi sufficiente elevare al quadrato i due parametri della retta.

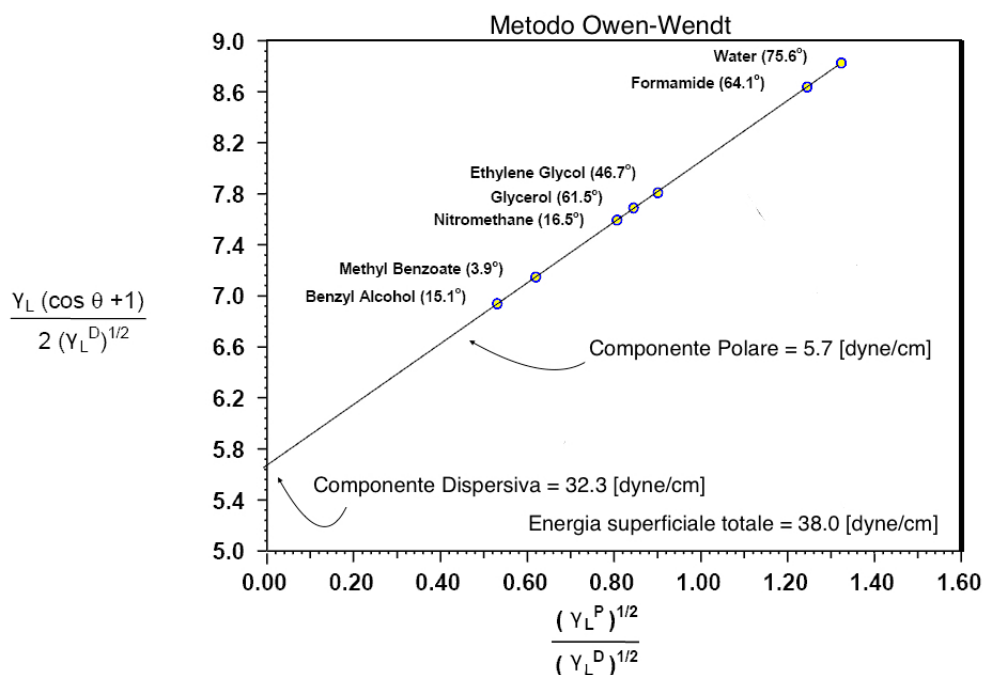


Figura 2.2: In figura é mostrato un esempio del grafico che si ottiene secondo il metodo di Owen-Wendt

2.2 Teoria dell'isteresi

In teoria una goccia su una superficie forma un solo angolo di contatto stabile (eq.2.1), ma nella pratica si osservano anche altri angoli meta-stabili. Il metodo piú semplice per osservare l'intervallo di variazione dell'angolo di contatto é aggiungere e sottrarre liquido alla goccia. [30]

L'angolo piú grande che si ottiene prima che l'interfaccia solido-liquido inizi ad avanzare é chiamato angolo di avanzamento θ_a , mentre l'angolo piú piccolo che si ottiene prima che l'interfaccia solido liquido inizi a retrocedere é chiamato angolo di recessione θ_r (Figura 2.2).

Si definisce l'isteresi H come la differenza fra θ_a e θ_r [30].

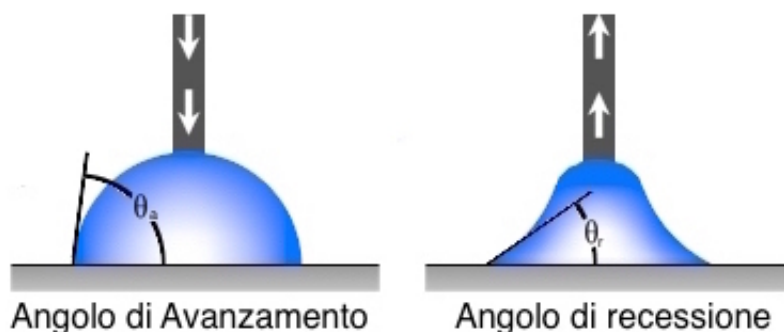


Figura 2.3: In figura sono rappresentati gli angoli di avanzamento θ_a e recessione θ_r

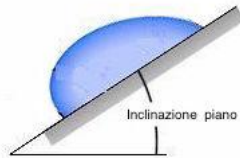
Si osserva anche che, durante l'avanzamento e la recessione, gli angoli θ_a e θ_r si mantengono costanti. Sono chiamati angoli di contatto dinamici, perché si misurano quando la goccia é in movimento.

La causa dell'isteresi viene attribuita a due fattori:

1. la rugositá della superficie, ossia la morfologia superficiale
2. la non omogeneitá della superficie, ossia la chimica della superficie

Secondo questo modello, quindi, una superficie idealmente piana e omogenea mostrerá isteresi nulla. L'origine dell'isteresi viene vista nella presenza di micro-domini con energia superficiale diversa all'interno della superficie. Una goccia quindi non puó avanzare o retrocedere liberamente. Per potersi muovere la goccia deve raggiungere un angolo di contatto tale da superare la barriera di potenziale creata dai micro-domini. Si puó quindi interpretare l'isteresi come un'energia di attivazione.

Consideriamo il seguente esempio esplicativo [27]. Una goccia posta su un piano inclinato, a seconda dell'inclinazione del piano, sarà piú o meno deformata dall'effetto della gravitá.



Questa deformazione puó essere vista come una variazione dell'energia della goccia rispetto allo stato non deformato:

$$\Delta E = \Delta A_{lv} \gamma_{lv} \quad (2.6)$$

dove ΔA_{lv} é la variazione dell'area dell'interfaccia liquido-vapore. Aumentando l'inclinazione del piano, la goccia puó iniziare a scivolare. Se consideriamo ΔE in questo istante, otteniamo:

$$E_a = \Delta A_{lv} \gamma_{lv} \quad (2.7)$$

E_a rappresenta l'energia necessaria alla goccia per iniziare a scivolare, ossia l'energia di attivazione che la goccia deve avere per superare le barriere di potenziale create dai micro-domini con γ_{sv} diverse.

Non potendo misurare direttamente ΔA_{lv} , si misura l'isteresi, quale indice della deformazione della goccia. I campioni che saranno analizzati hanno una rugositá quadratica media superficiale di $2.3 \pm 0.6 \text{ \AA}$, quindi, a paritá di morfologia superficiale, l'isteresi misura qualitativamente quanto il drogaggio del substrato sottostante influenza l'ossido nativo in termini di omogeneitá chimica/fisica superficiale.

Capitolo 3

Metodi Sperimentali

3.1 Preparazione dei campioni

Per trovare la dipendenza dell'energia superficiale dell'ossido di silicio dalla concentrazione di drogaggio del substrato sottostante, si sono utilizzati 10 wafer di silicio differenti. La resistività, il tipo e la concentrazione¹ dei wafer sono riportate in Tabella 3.1:

<i>Concentrazione atomi/cm³</i>	<i>Resistività $\Omega \cdot cm$</i>	<i>Tipo</i>
$(2.0 \pm 0.7) \cdot 10^{13}$	500 – 1000	<i>p</i>
$6.48 \cdot 10^{14}$	7	<i>n</i>
$(8 \pm 2) \cdot 10^{14}$	14 – 22	<i>p</i>
$(10 \pm 3) \cdot 10^{14}$	10 – 20	<i>p</i>
$(2.1 \pm 0.7) \cdot 10^{15}$	5 – 10	<i>p</i>
$(3 \pm 2) \cdot 10^{15}$	1 – 10	<i>n</i>
$(4 \pm 3) \cdot 10^{15}$	2 – 10	<i>p</i>
$(1.6 \pm 0.3) \cdot 10^{16}$	0.8 – 1.2	<i>p</i>
$(7 \pm 7) \cdot 10^{16}$	0.07 – 1	<i>n</i>
$1.497 \cdot 10^{19}$	0.0015	<i>n</i>

Tabella 3.1: In tabella sono riportati i tipi di wafer utilizzati

I wafer di silicio su é cresciuto l'ossido nativo sono dischi di mezzo millimetro di spessore e 10 *cm* di diametro, con un lato levigato a specchio.

I substrati di tipo *p* hanno come atomo drogante il boro, quelli di tipo *n* hanno come atomo drogante il fosforo.

Per condurre le misure, di ogni wafer si sono utilizzati dei frammenti delle dimensioni di circa $\frac{1}{2} cm^2$.

¹Per trovare l'intervallo di concentrazione relativo all'intervallo di resistività indicato, si é utilizzato un programma *online* [1].

Per ottenere tali frammenti si sfrutta la struttura cristallina del silicio. Questa struttura permette, premendo con una punta diamantata sul bordo, di spezzare lungo un piano cristallino il wafer. Tutta la procedura é stata condotta dentro a una cappa a flusso laminare per mantenere l'ambiente privo di polvere.

I frammenti vicino al bordo sono stati scartati perché, in genere, rimangono la parte piú sporca dell'intera superficie. I campioni sono stati poi puliti con la procedura di pulizia illustrata nel paragrafo 3.3 e riposti dentro una scatola porta campioni di plastica (Figura 3.1).

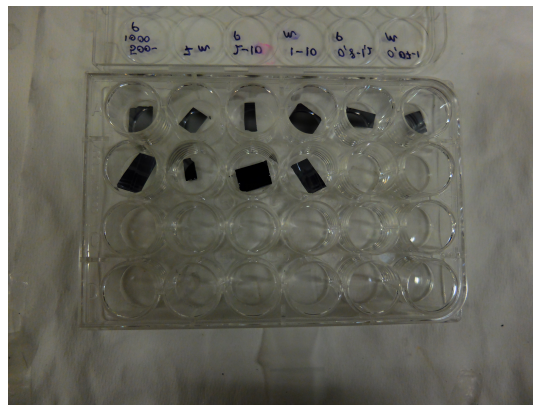


Figura 3.1: In figura é mostrata la foto della scatola porta campioni dove sono stati conservati i campioni

3.2 Liquidi utilizzati nelle misure di angolo di contatto statico

Per la misura dell'energia superficiale di ogni superficie si sono utilizzati i quattro liquidi riportati nella tabella 3.2.

	γ_{totale} dyne/cm	γ_{polare} dyne/cm	$\gamma_{dispersiva}$ dyne/cm
<i>Acqua</i>	72.8 ± 0.1	50.8 ± 0.4	22.0 ± 0.3
<i>Glicerolo</i>	63.6 ± 0.3	28 ± 2	35.8 ± 1.6
<i>Glicole – etilenico</i>	48.2 ± 0.4	17.9 ± 1.4	30.4 ± 1.8
<i>N – dimetilformamida</i>	37 ± 1	7.7 ± 0.4	29.1 ± 1.4

Tabella 3.2: In tabella sono riportati i liquidi utilizzati e le rispettive componenti della tensione superficiale

Poiché non si disponeva della strumentazione per la misurazione della tensione superficiale dei liquidi utilizzati, i valori in tabella sono la media \pm la deviazione standard dei valori dei liquidi puri trovati su vari articoli accademici [13][28][16][10][26][21][20][2][6]. Considerando ad esempio il glicerolo, si trovano i seguenti valori:

γ_l	γ_l^d	γ_l^p
64	34	30
63.4	37	26.4
63.4	37	26.4
63.4	37	26.4
64	34	30
63.7	–	–

Facendo la media aritmetica e la deviazione standard dei valori in tabella 3.2, otteniamo: $\gamma_{l(glicerolo)} = 63.6 \pm 0.3$ [dyne/cm], $\gamma_{l(glicerolo)}^d = 35.8 \pm 1.6$ [dyne/cm], $\gamma_{l(glicerolo)}^p = 27.8 \pm 2$ [dyne/cm].

I liquidi utilizzati sono stati scelti secondo i seguenti criteri:

1. ottenere il massimo intervallo di valori di tensione superficiale
2. aver solventi puri che non reagiscano con le superfici

3.3 Pulizia dei campioni

Di alcuni tipi di wafer si avevano a disposizione solo pochi frammenti, questo ha reso necessario trovare una tecnica di pulizia efficace per poterli riutilizzare senza compromettere le misure successive.

Il metodo di pulizia utilizzato prevede di pulire la superficie in 4 fasi:

1. risciacquo con acqua deionizzata
2. risciacquo con acetone
3. risciacquo con isopropanolo
4. asciugatura con azoto tecnico

Essendo tutti i liquidi utilizzati (Tabella 3.2) solubili in acqua, il risciacquo con acqua deionizzata sarebbe di per sé sufficiente a pulire i campioni, tuttavia il risciacquo con acetone e isopropanolo serve ad eliminare eventuali tracce oleose sulla superficie. Come ultimo solvente viene utilizzato l'isopropanolo perché evaporando non lascia praticamente nessun residuo. [3]

I tre risciacqui vengono fatti sostenendo il campione in un angolo con una pinzetta e facendo scorrere il solvente sulla superficie per ~ 3 secondi.

Per ridurre al minimo la quantità di solvente assorbita, si asciuga il campione con l'azoto tecnico, avendo l'accortezza di puntare il getto dalla superficie verso la pinzetta.

Per essere certi che le misure sul campione non siano influenzate dal metodo di pulizia, si è effettuata la seguente verifica. Si è preso un frammento di silicio pulito solo con l'azoto tecnico e se ne è misurata l'energia superficiale, poi è stata ripetuta la misura dopo la pulizia². I risultati ottenuti sono i seguenti:

	<i>Campione pulito</i>	<i>Campione originale</i>
<i>Angolo di contatto acqua</i>	$53^\circ \pm 1^\circ$	$52.2^\circ \pm 0.7^\circ$
<i>Angolo di contatto glicerolo</i>	$44.3^\circ \pm 0.9^\circ$	$45.2^\circ \pm 0.9^\circ$
<i>Angolo di contatto glicole – etilenico</i>	$27.8^\circ \pm 0.6^\circ$	$27.7^\circ \pm 0.8^\circ$
<i>Angolo di contatto N – dimetilformamida</i>	$10.0^\circ \pm 0.6^\circ$	$11^\circ \pm 1^\circ$
<i>Componente dispersiva γ_s^d</i>	$15 \pm 3 \text{ dyne/cm}$	$15 \pm 3 \text{ dyne/cm}$
<i>Componente polare γ_s^p</i>	$31 \pm 3 \text{ dyne/cm}$	$32 \pm 3 \text{ dyne/cm}$
<i>Energia superficiale totale γ_s</i>	$47 \pm 4 \text{ dyne/cm}$	$47 \pm 4 \text{ dyne/cm}$

Tabella 3.3: Nella tabella sono confrontate le misure ottenute coi campioni originali con quelle ottenute col campione pulito. Si nota che le due misure sono perfettamente compatibili fra loro

²Si intende il trattamento di pulizia appena descritto

Come si vede dalle misure nella tabella 3.3, il metodo di pulizia non influisce sulla misura dell'energia superficiale.

3.4 Strumentazione

Per le misure di questa tesi si é utilizzato un tensiometro ottico fornito dalla Digidrop [11](Figura 3.2), basato sulla tecnica della goccia sessile. Questa tecnica consiste nel fotografare il profilo di una goccia appoggiata su una superficie, per poi ricavare l'angolo di contatto tramite un software dedicato.

Il tensiometro é composto da vari parti:

1. una macchina fotografica digitale dotata di zoom
2. una siringa da 3 millilitri
3. un porta siringa capace di controllare il volume della goccia nell'ordine del microlitro³.
4. un piano motorizzato che puó muoversi in verticale
5. un porta campione
6. una siringa motorizzata per le misure di isteresi

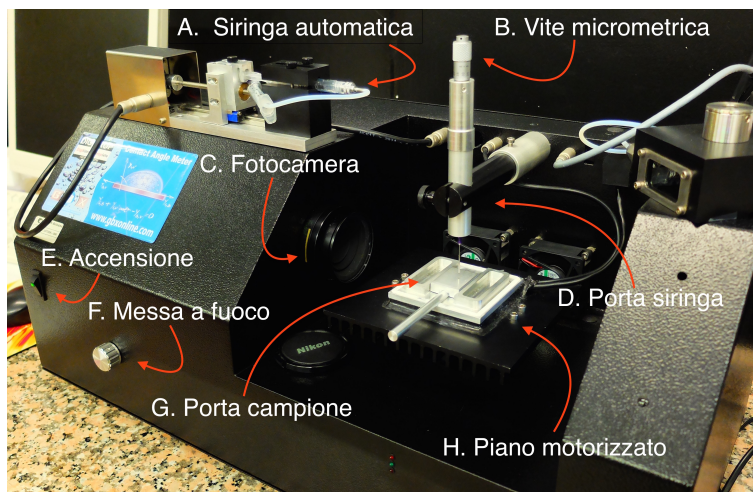


Figura 3.2: In figura é mostrato il tensiometro ottico utilizzata

³Una vite micrometrica preme sulla siringa contenente il liquido, controllando accuratamente la pressione dello stantuffo e quindi il volume

Tutta la strumentazione é collegata ad un computer con un software capace sia di raccogliere ed analizzare le immagini delle gocce, sia di manovrare le parti motorizzate della strumentazione.

Per la misurazione dell'angolo di contatto statico si procede in questo modo. Si riempie una siringa da 3 ml con il liquido da utilizzare e la si inserisce nel porta siringa (Figura 3.2 B). Poi si fissa il porta siringa perpendicolare al piano motorizzato e con la macchina fotografica si mette a fuoco la punta della siringa (Figura 3.3 A) che viene visualizzata in tempo reale sullo schermo del computer. A questo punto il frammento di silicio da analizzare viene appoggiato sopra il porta campione (Figura 3.2 G) che a sua volta viene appoggiato sopra il piano motorizzato.

Girando la vite micrometrica, si forma la goccia da depositare (Figura 3.3 B). Per regolare con precisione il volume della goccia si usa il software. Nel nostro caso si é utilizzato un volume di $\sim 1\mu l$. Ottenuto il volume desiderato, per depositare la goccia sulla superficie si avvicina il campione fino al distacco di quest'ultima (Figura 3.3 C). Questa procedura viene condotta direttamente dallo schermo manovrando il piano motorizzato.

Quando la goccia si stabilizza (Figura 3.3 D), premendo l'apposita icona (Figura 3.5), si scatta una foto, che verrá poi analizzata.

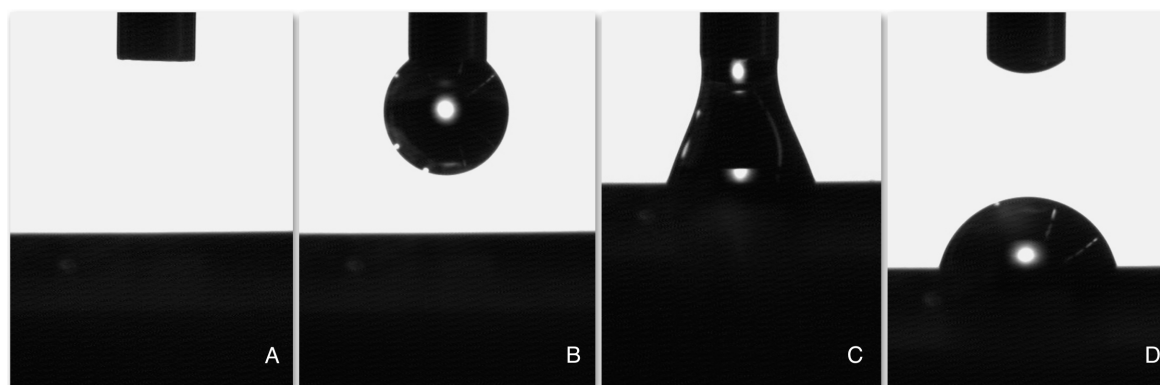


Figura 3.3: In figura é mostrata la procedura di deposizione di una goccia su una superficie

In generale la misura dell'angolo di contatto é una misura difficilmente riproducibile e soggetta a grosse fluttuazioni anche su campioni perfettamente omogenei. Per ottenere una precisione sufficientemente alta tutte le misure vengono prese su almeno 3 punti del campione. In seguito verrá illustrato lo studio statistico utilizzato per ottenere il giusto valore dell'angolo di contatto.

L'altra misura utilizzata per analizzare le superfici di ossido é l'isteresi dell'angolo di contatto.

Come descritto precedentemente, l'isteresi é definita come la differenza fra gli angoli di avanzamento e di recessione, che sono rispettivamente gli angoli formati dalla goccia con la superficie quando l'interfaccia liquido-solido si allarga e si restringe. Questa misura viene effettuata utilizzando il metodo della goccia sessile dinamica, che consiste nel visionare, fotogramma per fotogramma, un video del profilo della goccia che si allarga e si restringe.

Per la misura dell'isteresi si procede in questo modo.

Come si é fatto per l'angolo statico, si riempie la siringa⁴ (Figura 3.2 A) con il liquido da utilizzare e, grazie alla fotocamera, se ne mette a fuoco la punta che viene visualizzata sullo schermo del computer. Si pone poi il frammento di silicio sopra il porta campioni posizionandolo sopra il piano motorizzato.

Seguendo la stessa procedura illustrata per l'angolo statico, si deposita una goccia sulla superficie, ma si lascia la punta della siringa a contatto con l'interfaccia liquida in modo da poter aumentare o ridurre il volume della goccia (Figura 3.4 A). Essendo la siringa collegata al computer, si fa partire una procedura automatica che registra il profilo della goccia mentre il liquido viene iniettato e poi aspirato (Figura 3.4 B-C).

Durante questa operazione si puó quantificare il volume di liquido da iniettare e aspirare inserendo direttamente i valori desiderati nel software (Figura 3.5). É possibile anche decidere se fare passare del tempo fra l'iniezione e l'aspirazione del liquido, per lasciare stabilizzare la goccia fra avanzamento e recessione.

Per le misure in questa tesi il volume della goccia si é fatto variare da 1 e 7 μl e si é utilizzato un intervallo di 3 secondi fra l'iniezione e la suzione.

Terminata questa operazione, si passa alla misura dell'angolo di contatto dinamico analizzando il video ottenuto fotogramma per fotogramma.

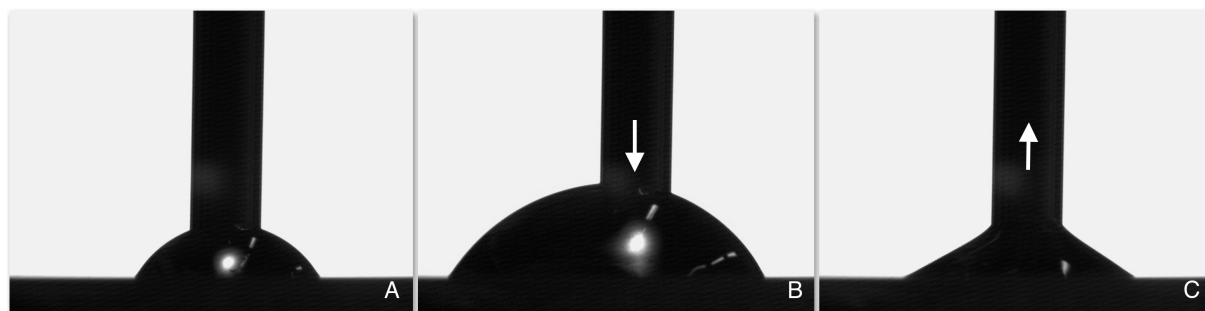


Figura 3.4: In figura é mostrata la procedura per la misura dell'isteresi

⁴In questo contesto si intende sempre la siringa motorizzata

3.5 Software Digidrop

Insieme alla strumentazione hardware la Digidrop fornisce anche un software, che può essere aperto in due modalità: *Contact Angle* o *Wetting Hysteresis*. I video si possono processare solo in modalità isteresi, per il resto le due modalità sono equivalenti.

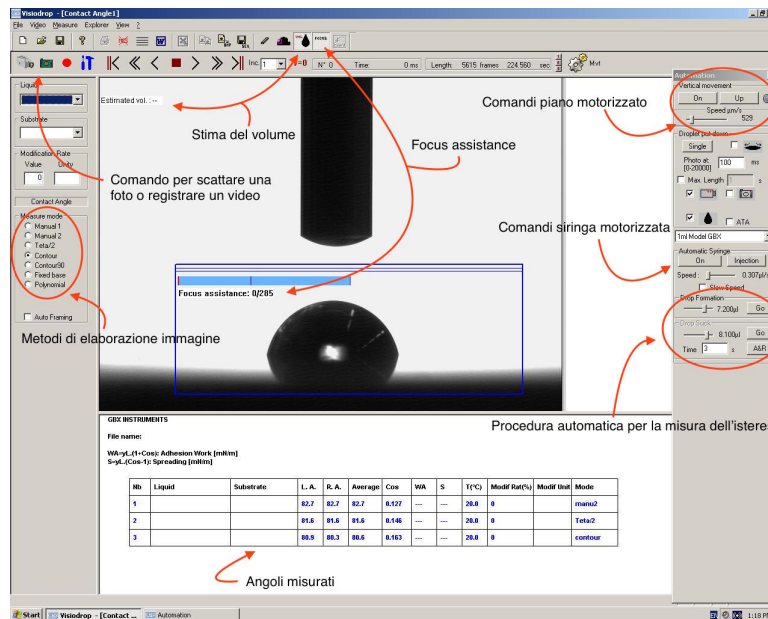


Figura 3.5: In figura é mostrata la schermata del software in modalità *contact angle*.

Il programma permette sia di manovrare tutte le parti motorizzate dello strumento sia di analizzare le immagini del profilo della goccia. Per mezzo della parte collegata alla strumentazione é possibile:

1. visualizzare in tempo reale il profilo della goccia
2. muovere in verticale e a varie velocità il piano motorizzato
3. comandare la siringa motorizzata, decidendo la velocità con cui iniettare o aspirare il liquido
4. svolgere l'operazione automatica per la misura dell'isteresi

Inoltre il programma offre un'opzione che aiuta l'operatore nella messa a fuoco manuale. Per fare funzionare correttamente il programma, una volta messa a fuoco la punta della siringa, bisogna calibrare l'immagine: questa operazione si fa inserendo lo spessore della

siringa nel comando *calibrate width*.

La parte del software dedicata all'analisi delle immagini permette di ricavare da una foto l'angolo di contatto di una goccia.

Per fare questa operazione il programma offre una serie di metodi piú e meno sofisticati, di cui proponiamo una breve descrizione:

1. Il primo metodo si chiama *Manual1*. Questo metodo chiede all'operatore di individuare i tre punti che delimitano la base e l'altezza della goccia. Dopodiché restituisce automaticamente il valore dell'angolo.
2. Il secondo metodo proposto si chiama *Manual2*. Sostanzialmente questo metodo é uguale a *Manual1*, l'unica differenza é che il punto che individua l'altezza della goccia é vincolato a giacere lungo l'asse della base.
3. Il terzo metodo si chiama $\theta/2$. Questo metodo trova automaticamente i tre punti richiesti dai primi 2 metodi analizzando i pixel dell'immagine.
4. Il quarto metodo si chiama *Contorn*. Questo metodo trova automaticamente il contorno della goccia analizzando i pixel dell'immagine e ne calcola l'angolo di contatto.
5. Il quinto metodo si chiama *Contorn90*. Il funzionamento di questo metodo é sostanzialmente uguale al precedente.
6. Il sesto metodo si chiama *Fixed base*. Questo metodo richiede all'operatore di individuare la retta dove giace la base della goccia.
7. Il settimo e ultimo metodo si chiama *Polynomial*. Questo metodo é automatico e approssima il profilo della goccia a una curva. Non é adatto a misurare angoli maggiori di 90 gradi.

L'algoritmo alla base dei primi tre metodi approssima la goccia a una sezione di sfera. In questo modo l'angolo di contatto é semplicemente il doppio degli angoli alla base che si formano congiungendo i tre punti individuati. Gli altri metodi invece utilizzano algoritmi piú sofisticati, che però non sempre funzionano correttamente⁵. Una buona tecnica per evitare malfunzionamenti é quella di ritagliare dall'immagine solo il profilo della goccia, escludendo altri oggetti (come la punta della siringa) che possono confondere il programma.

⁵Se l'immagine é leggermente sfocata o presenta bordi frastagliati vengono restituiti angoli falsati

Tutti i metodi elencati hanno una riproducibilità di circa 4-5 gradi, quindi, per ottenere la stima migliore degli angoli restituiti, abbiamo deciso di processare ogni immagine con almeno 3 tecniche diverse. In generale vengono utilizzati tutti i metodi che restituiscono valori attendibili, ovvero i metodi 2, 3, 4, 6 e 7. Questo, unito al fatto che ogni misura viene fatta su almeno 3 punti della superficie, porta ad avere per ogni angolo dalle 10 alle 15 misure. Per ottenere quindi la misura definitiva, é necessario un breve studio statistico, che sará descritto nel capitolo successivo.

Per analizzare i video degli angoli di avanzamento e recessione il programma va aperto in modalitá *Wetting Hysteresis*. In questo caso l'analisi avviene automaticamente fotogramma per fotogramma, utilizzando però solo i metodi *fixed base* e *polynomial*. Questa operazione richiede molto tempo e nella maggior parte dei casi trova molti piú dati di quelli necessari. Il programma offre quindi l'opzione di poter scegliere l'intervallo di fotogrammi da analizzare e il passo con cui analizzarli (per esempio uno ogni due). Tutti i dati che vengono raccolti (sia in modalitá angolo statico sia in modalitá isteresi) possono essere esportati come file di testo (.txt).

Capitolo 4

Elaborazione dati

4.1 Angolo Statico

Come si é visto nel capitolo precedente, dal software dell'angolo di contatto si ricavano molte misure dello stesso angolo.

Quello che vogliamo ottenere però é un solo valore. É quindi necessario un breve studio statistico delle misure.

Per velocizzare il procedimento é stato scritto un programma in C++.

Questo programma in ingresso riceve il file¹ di testo prodotto dal software di analisi dell'angolo di contatto e in uscita restituisce i seguenti valori²:

1. Media aritmetica angolo destro
2. Media aritmetica angolo sinistro
3. Media aritmetica fra tutti gli angoli (senza distinguere destro e sinistro)
4. Media pesata fra angolo destro e sinistro
5. Coseno della media aritmetica fra tutti gli angoli
6. Coseno della media pesata fra angolo destro e sinistro

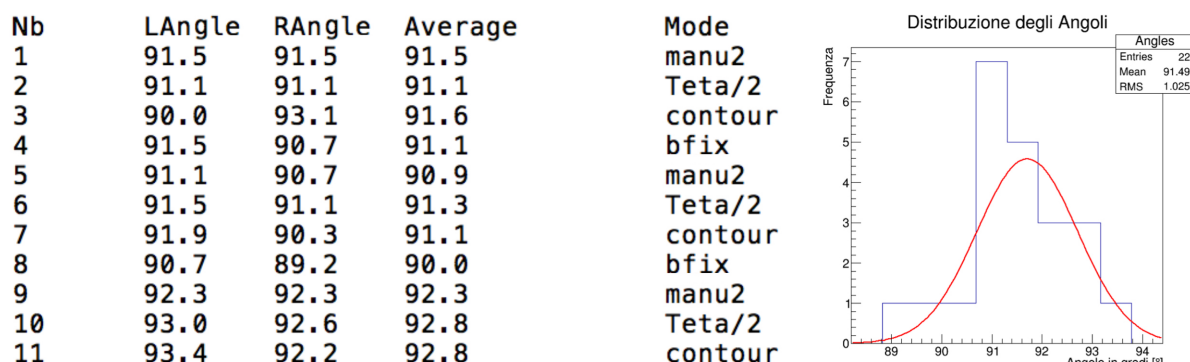
A tutte le misure é associato un errore assoluto pari alla deviazione standard. [25]

Per verificare visivamente che le misure di uno stesso angolo si dispongano come una gaussiana e che quindi l'errore sia casuale, il programma traccia un istogramma delle frequenze (Figura 4.1). Utilizzando la libreria *minuit* di ROOT data analysis [9], l'istogramma viene *fittato* con una gaussiana.

¹Questo file consiste in un elenco di angoli destri e sinistri, suddivisi per metodologia di elaborazione (*Manual1*, *Manual2*, ecc. . .).

²Affinché il programma lavori correttamente le misure devono riferirsi a un solo angolo

A titolo di esempio consideriamo il file in Figura 4.1, che contiene gli angoli statici dell'acqua sul campione di wafer *as received* tipo *n* a resistività $0.07 - 1 \Omega \cdot \text{cm}$:



Da questi dati il programma ricava le seguenti quantità:

1. Media aritmetica angolo destro = $91.3^\circ \pm 1.1^\circ$
2. Media aritmetica angolo sinistro = $91.6^\circ \pm 1.0^\circ$
3. Media aritmetica fra tutti gli angoli = $91.5^\circ \pm 1.0^\circ$
4. Media pesata fra angolo destro e sinistro = $91.5^\circ \pm 0.7^\circ$
5. Coseno della media aritmetica fra tutti gli angoli = $-0.026^\circ \pm 0.018^\circ$
6. Coseno della media pesata fra angolo destro e sinistro = $-0.026^\circ \pm 0.013^\circ$

I parametri della gaussiana restituiti da ROOT sono:

- Media = $91.7^\circ \pm 0.3^\circ$
- Deviazione standard = $1^\circ \pm 0.3^\circ$

che sono perfettamente compatibili con i valori trovati calcolando la media aritmetica e la deviazione standard dei valori.

4.1.1 Calcolo dell'energia superficiale

Trovata la migliore stima degli angoli statici dei quattro liquidi³, si passa al calcolo dell'energia superficiale con il Metodo di Owen-Wendt. Come spiegato in 2.1, per trovare l'energia superficiale é necessario fare la regressione lineare dei punti:

³Come migliore stima si é considerata la media aritmetica fra tutti gli angoli, senza distinguere fra destro e sinistro

$$\begin{aligned}
y_i &= \frac{\gamma(\cos\theta_i+1)}{2\sqrt{\gamma_d}} \quad [\text{dyne/cm}]^{1/2} \\
\delta_{y_i} &= \sqrt{\left(\frac{\delta_\gamma(\cos\theta_i+1)}{\sqrt{\gamma_d}}\right)^2 + \left(\frac{\gamma\delta_{\cos\theta_i}}{2}\right)^2 + \left(\frac{\delta_{\gamma_d}\gamma(\cos\theta_i+1)}{4\gamma_d\sqrt{\gamma_d}}\right)^2} \quad [\text{dyne/cm}]^{1/2} \\
x_i &= \frac{\sqrt{\gamma_p}}{\sqrt{\gamma_d}} \\
\delta_{x_i} &= \sqrt{\left(\sqrt{\frac{\gamma_p}{\gamma_d^3}}\frac{\delta_{\gamma_d}}{2}\right)^2 + \left(\frac{\delta_{\gamma_p}}{\sqrt{\gamma_d\gamma_p}}\right)^2}
\end{aligned}$$

In queste espressioni $\cos\theta_i$ é il coseno dell'angolo di contatto statico e γ_p , γ_d , γ sono la componente polare, la componente dispersiva e l'energia superficiale totale del liquido. $\delta_{\cos\theta_i}$, δ_{γ_p} , δ_{γ_d} e δ_γ sono le incertezze associate ai rispettivi pedici. Avendo utilizzato quattro liquidi si ottengo quattro punti.

La regressione lineare si ottiene con un programma scritto in C++ che sfrutta la libreria minuit di ROOT data analysis [9]. In ingresso il programma riceve un file di testo contenente $\cos\theta$, γ_i^d , γ_i^p e γ_l di ogni liquido, dai quali calcola y_i , δ_{y_i} , x_i , δ_{x_i} . Si ricava poi la retta di regressione, dai cui parametri si ottengono le componenti della energia superficiale del solido secondo le seguenti espressioni:

$$\begin{aligned}
\gamma_s^p &= m^2 \quad [\text{dyne/cm}] \\
\delta_{\gamma_s^p} &= 2m \cdot \delta_m \quad [\text{dyne/cm}] \\
\gamma_s^d &= q^2 \quad [\text{dyne/cm}] \\
\delta_{\gamma_s^d} &= 2q \cdot \delta_q \quad [\text{dyne/cm}]
\end{aligned}$$

dove m indica il coefficiente angolare della retta e q l'intersezione con l'asse delle y . L'energia superficiale totale γ_s é data da :

$$\begin{aligned}
\gamma_s &= \gamma_s^p + \gamma_s^d \quad [\text{dyne/cm}] \\
\delta_{\gamma_s} &= \delta_{\gamma_s^p}^2 + \delta_{\gamma_s^d}^2 \quad [\text{dyne/cm}]
\end{aligned}$$

Tali valori sono restituiti dal programma insieme al grafico della regressione per avere una verifica visiva dei risultati(Figura 4.1).

A titolo di esempio consideriamo gli angoli misurati per il wafer a resistività $0.07 - 1\Omega \cdot \text{cm}$ *as received* riportati in Figura 4.1. Da questi angoli, utilizzando le tensioni superficiali dei liquidi riportate in Tabella 3.2, otteniamo i punti y e x riportati in Figura 4.1. Da questi punti ROOT trova la retta di regressione con i seguenti parametri:

- Coefficiente angolare $m = 1.91 \pm 0.24 \quad [\text{dyne/cm}]^{1/2}$
- Intercetta Asse y $q = 4.6 \pm 0.3 \quad [\text{dyne/cm}]^{1/2}$

Dai questi valori si ottengono:

- Componente Polare $\gamma_s^p = 3.6 \pm 0.9 \quad [\text{dyne/cm}]$

- Componente Dispersiva $\gamma_s^d = 21 \pm 2$ [dyne/cm]
- Energia Superficiale $\gamma_s = 25 \pm 3$ [dyne/cm]

	Coseno Angolo	ErroreCoseno
Acqua	-0.0260183	0.0182983
Glicerolo	0.110255	0.0242777
Glicole etilenico	0.369679	0.0255271
DMF	0.743908	0.025094

y [dyne/cm] ^{1/2}	X
7.55859+-0.151423	1.51957+-0.0119639
5.90077+-0.186577	0.881213+-0.0373169
5.98189+-0.215082	0.766271+-0.0376514
5.94186+-0.232093	0.514398+-0.0182106

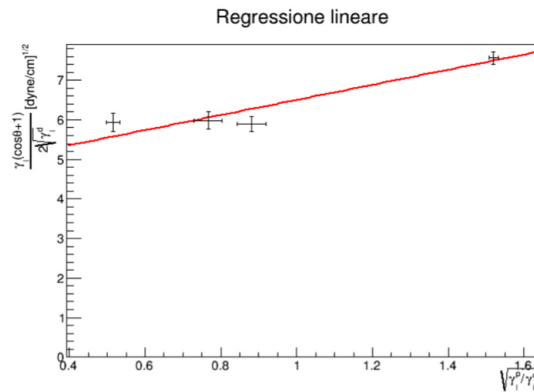


Figura 4.1: In figura é mostrato il fit lineare del metodo Owen-Wendt per il campione *as received* di resistività $0.07 - 1 \Omega \cdot \text{cm}$

4.2 Angolo Dinamico/Isteresi

L'analisi dati dell'angolo di contatto dinamico é leggermente piú complicata dell'analisi dell'angolo statico. Infatti il software del tensiometro produce un file.txt che contiene i valori di tutti gli angoli misurati in un determinato intervallo di tempo, senza distinguere le misure dell'angolo di avanzamento da quelle dell'angolo di recessione.

Per velocizzare il processo e per avere la stima piú corretta possibile degli angoli dinamici, tutti i dati vengono analizzati da un programma scritto in C++, il cui scopo principale é capire quali valori del file rappresentano l'angolo di avanzamento e quali quello di recessione.

Analizziamo ora il funzionamento del programma utilizzando come esempio il file i2.txt riportato in Figura 4.2.

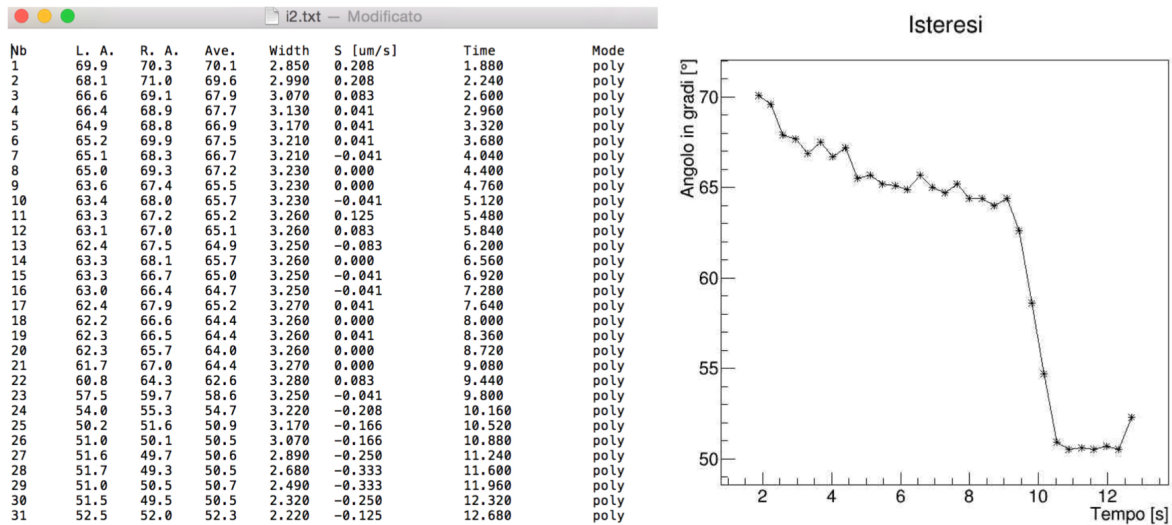


Figura 4.2: In figura é mostrato un esempio del contenuto di un file prodotto dal software della Digidrop in modalitá *Wetting Hysteresis*

Per prima cosa il programma richiede il nome del file (i2.txt), poi organizza i valori degli angoli riportati nella colonna *Ave.*⁴ in un array che chiamiamo *AngoloTotale*. A questo punto viene dichiarata una variabile intera che chiamiamo *Centro*. Questa variabile rappresenta l'indice dell'angolo nell'array che separa le misure dell'angolo di avanzamento [0 – *Centro*] da quelle dell'angolo di recessione [*Centro* – *fine*]. Per trovare il valore corretto di *Centro*, si calcola σ^2 per tutti i valori che può assumere *Centro* dentro *AngoloTotale*, ossia per tutti gli interi compresi tra 0 e *fine*⁵:

$$\sigma^2 = \sigma_{[0-Centro]}^2 + \sigma_{[Centro-fine]}^2$$

dove σ rappresenta la deviazione standard dei rispettivi intervalli. Se si osserva l'espressione di σ^2 , si vede che essa rappresenta il quadrato dell'errore dell'isteresi in funzione *Centro* (equazione 4.2).

L'intero per cui é minimo σ^2 rappresenta l'indice che meglio delimita i due intervalli e viene definitivamente assegnato alla variabile *Centro*.

Nel file in Figura 4.2, *Centro* può assumere i valori fra 2 e 28 compresi. Dall'equazione 4.2 si ottengono quindi 27 valori di σ^2 .

Per trovare il valore corretto di *Centro* dobbiamo, come spiegato sopra, prendere l'intero per cui é minima σ^2 .

⁴*Ave.* é l'abbreviazione per *Averaged Angle* che corrisponde al valore medio fra l'angolo destro (R.A) e l'angolo sinistro (L.A)

⁵in realtà da 2 a (*fine* – 3), perché σ necessita di almeno 3 valori per avere senso

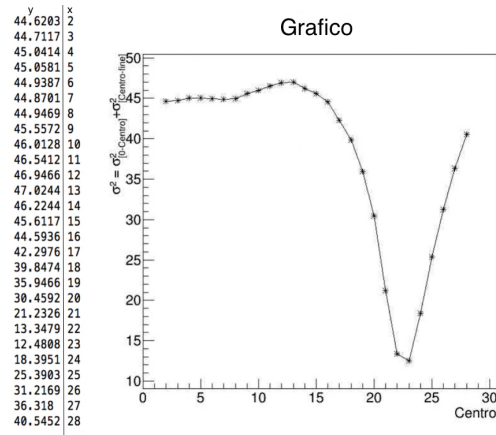


Figura 4.3: In figura sono riportati i valori e il grafico di σ^2 in funzione di $Centro$

Come si vede in Figura 4.3, il minimo di σ^2 corrisponde a $Centro = 23$.

In realtà però non è un solo punto a dividere i due intervalli, infatti, come si vede nel grafico di Figura 4.2, gli angoli non passano da avanzamento a recessione istantaneamente, ma c'è una serie di angoli intermedi, misurati mentre il moto della goccia si inverte. Per scartare questi valori, si dichiarano due nuove variabili A e R , che servono a rifinire gli intervalli delle misure dell'angolo di avanzamento $[0 - (Centro - A)]$ e recessione, $[(Centro + R) - fine]$.

Partendo dal valore 0, si aumenta di 1 il valore di A e R e si calcola ogni volta il σ dei rispettivi intervalli.

Questa operazione prosegue finché la deviazione standard non inizia a crescere. A quel punto ci si ferma, perché significa che i valori degli angoli si sono stabilizzati.

Vediamo ora come varia la deviazione standard degli intervalli $[0 - (Centro - A)]$, $[(Centro + R) - fine]$ in funzione di A e R nel file i2.txt (Figura 4.4).

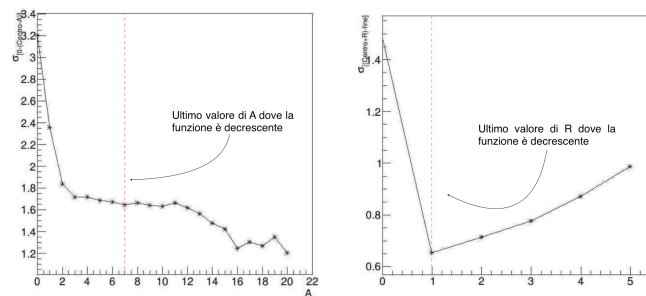


Figura 4.4: In figura sono riportati i grafici di $\sigma_{avanzamento}$ e $\sigma_{recessione}$ in funzione dei valori che possono assumere A e B

Nel primo grafico si vede che $\sigma_{avanzamento}$ decresce fino ad $A = 7$.
 Nel secondo grafico si vede che $\sigma_{recessione}$ decresce fino a $R = 1$.
 Si scartano in totale 7 angoli.

A questo punto si sono ottenuti gli intervalli esatti per θ_a [1 – 17] e per θ_r [24 – 30].
 Ai due angoli viene assegnata la media aritmetica dei rispettivi intervalli.
 L'isteresi é data da:

$$Isteresi = H = \theta_a - \theta_r [^\circ] \quad (4.1)$$

$$\delta_{Isteresi} = \sqrt{\delta_{\theta_a}^2 + \delta_{\theta_r}^2} [^\circ] \quad (4.2)$$

dove δ_{θ_a} e δ_{θ_r} sono gli errori di θ_a e θ_r che corrispondono a una deviazione standard. Per il file esempio i2.txt si trovano:

$\theta_a = 66.5 \pm 1.6$, $\theta_r = 50.9 \pm 0.7$, $Isteresi = 15.6 \pm 1.8$.

Per poter avere un controllo visivo dei risultati restituiti dal programma, tutti i dati vengono contemporaneamente graficati in funzione del tempo, utilizzando le librerie grafiche di ROOT data analysis (Figura4.5) [9].

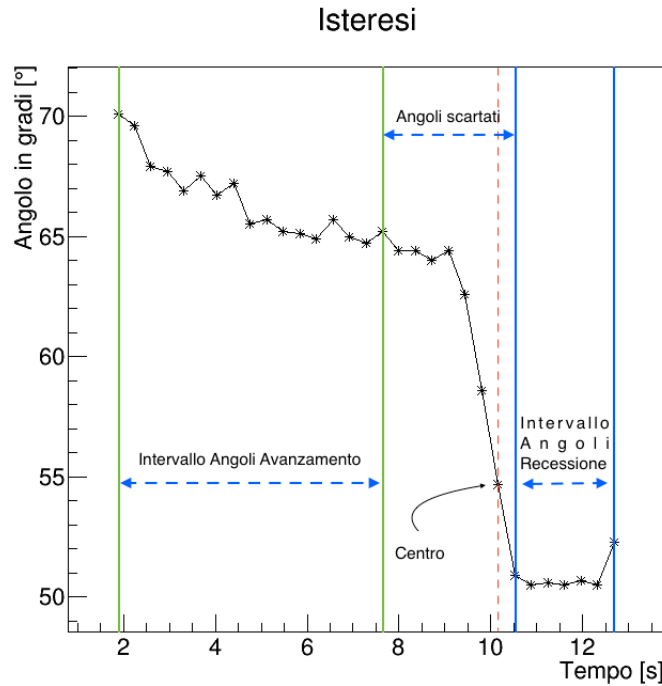


Figura 4.5: In figura é riportato il grafico di Figura 4.2, mettendo in evidenza i punti che rappresentano l'angolo di avanzamento (verde) e l'angolo di recessione (blu). La linea rossa tratteggiata segna il punto che in $AngoloTotale$ rappresenta $Centro$

Per le misure in questa tesi si sono utilizzati molti piú dati di quelli presenti nel file di esempio in Figura 4.2.

Capitolo 5

Misure ed analisi dati

5.1 Misure sui substrati *as received*

Si riportano di seguito i valori della componente dispersiva, della componente polare, dell'energia superficiale e dell'isteresi per i campioni *as received*.

<i>Conc.</i> [atomi/cm ³]	<i>Tipo</i>	γ_s^d [dyne/cm]	γ_s^p [dyne/cm]	γ_s^{tot} [dyne/cm]	<i>Isteresi</i>
$(2.0 \pm 0.7) \cdot 10^{13}$	<i>p</i>	19 ± 3	2.0 ± 0.8	21 ± 3	$16^\circ \pm 5^\circ$
$6.48 \cdot 10^{14}$	<i>n</i>	19 ± 3	11 ± 2	30 ± 4	$29^\circ \pm 3^\circ$
$(8 \pm 2) \cdot 10^{14}$	<i>p</i>	17 ± 3	18 ± 2	35 ± 4	$14^\circ \pm 3^\circ$
$(10 \pm 3) \cdot 10^{14}$	<i>p</i>	16 ± 3	23 ± 4	39 ± 5	$18.4^\circ \pm 1.4^\circ$
$(2.1 \pm 0.7) \cdot 10^{15}$	<i>p</i>	16 ± 3	31 ± 3	47 ± 4	$15^\circ \pm 3^\circ$
$(3 \pm 2) \cdot 10^{15}$	<i>n</i>	20 ± 2	2.2 ± 0.8	22 ± 3	$24^\circ \pm 5^\circ$
$(4 \pm 3) \cdot 10^{15}$	<i>p</i>	20 ± 3	8.6 ± 1.9	29 ± 4	$30^\circ \pm 3^\circ$
$(1.6 \pm 0.3) \cdot 10^{16}$	<i>p</i>	16 ± 3	2.6 ± 1.3	19 ± 3	$23^\circ \pm 6^\circ$
$(7 \pm 7) \cdot 10^{16}$	<i>n</i>	21 ± 2	3.6 ± 0.9	25 ± 3	$24^\circ \pm 5^\circ$
$1.497 \cdot 10^{19}$	<i>n</i>	21 ± 3	5.3 ± 1.2	27 ± 3	$16^\circ \pm 4^\circ$

Si riportano di seguito i grafici delle misure in tabella in funzione della concentrazione di drogante (in scala semi-logaritmica):

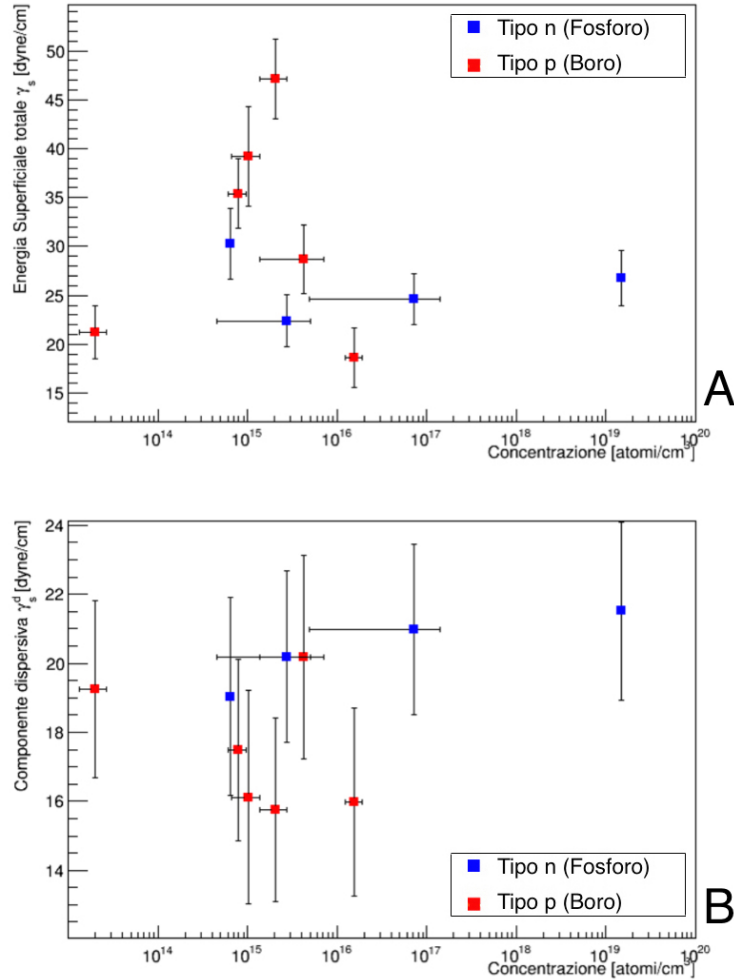


Figura 5.1: I grafici A e B rappresentano rispettivamente l'energia superficiale γ_s e la componente dispersiva γ_s^d in funzione della concentrazione di drogante per i campioni *as received*

Nel grafico in figura 5.1 A, che rappresenta γ_s in funzione della concentrazione di drogante, si osserva un evidente picco compreso fra 10^{15} e 10^{16} *atomi/cm³* composto esclusivamente da campioni di tipo *p* (rosso).

Questo picco raggiunge il suo massimo (47 ± 4 *dyne/cm*) con il campione $(2.1 \pm 0.7) \cdot 10^{15}$ *atomi/cm³*.

I campioni di tipo *n*, all'aumentare della concentrazione, si dispongono con un andamento lievemente crescente, ad eccezione del campione $6.48 \cdot 10^{14}$ *atomi/cm³*.

Nel grafico in figura 5.1 B, che rappresenta γ_s^d in funzione della concentrazione di dro-

gante, si osserva un lieve andamento decrescente per i campioni di tipo p (rosso) e un andamento lievemente crescente per i campioni di tipo n . Tuttavia né per i campioni p , né per i campioni n , γ_s^d presenta variazioni significative. Si può quindi approssimare con una costante ($\sim 20 \text{ dyne/cm}$).

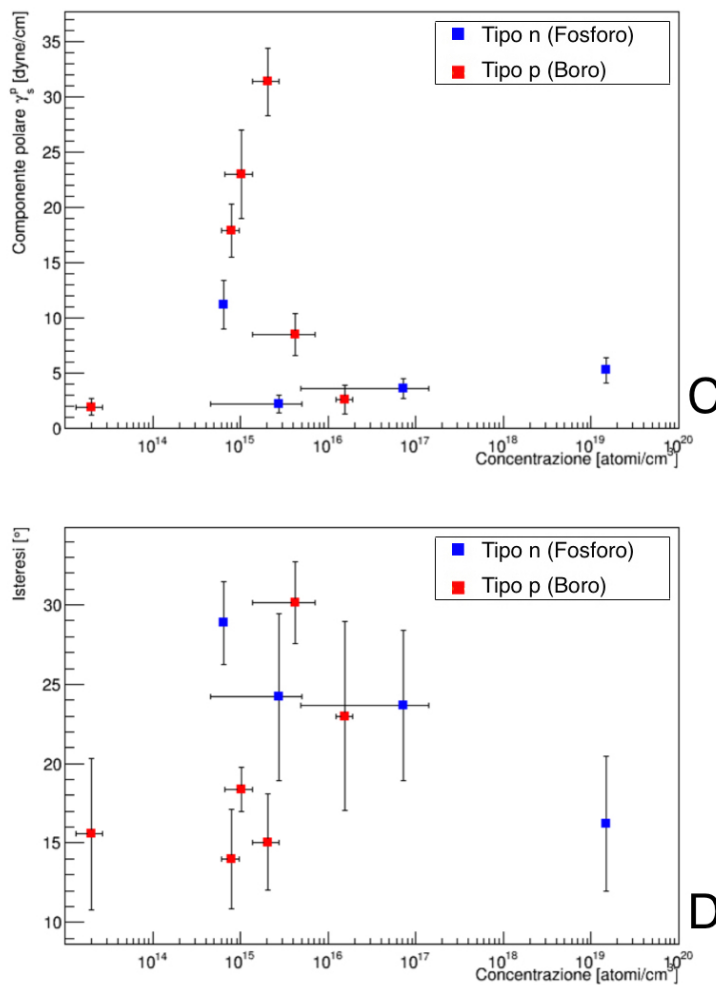


Figura 5.2: I grafici C e D rappresentano rispettivamente la componente polare γ_s^p e l'isteresi delle superfici in funzione della concentrazione di drogante per i campioni *as received*

Nel grafico in figura 5.2 C, che rappresenta γ_s^p in funzione della concentrazione di drogante, si osserva un evidente picco compreso fra 10^{15} e $10^{16} \text{ atomi/cm}^3$ composto esclusivamente da campioni di tipo p (rosso).

Questo picco raggiunge il suo massimo ($31 \pm 3 \text{ dyne/cm}$) con il campione $(2.1 \pm 0.7) \cdot 10^{15} \text{ atomi/cm}^3$.

I campioni di tipo n si dispongono con un andamento lievemente crescente all'aumentare della concentrazione, ad eccezione del campione $(2.0 \pm 0.7) \cdot 10^{13} \text{ atomi/cm}^3$.

Nel grafico in figura 5.2 A, che rappresenta l'isteresi in funzione della concentrazione di drogante, si osserva un andamento crescente per i campioni di tipo p (rosso) e un andamento decrescente per i campioni di tipo n .

Se consideriamo la media dei valori dell'isteresi otteniamo:

- Tipo $p = 19^\circ \pm 6^\circ$
- Tipo $n = 23^\circ \pm 5^\circ$
- Media pesata fra p e $n = 22^\circ \pm 4^\circ$

Dal momento che $\gamma_s = \gamma_s^p + \gamma_s^d$, notiamo che l'andamento dell'energia superficiale segue quello della componente polare. La componente dispersiva si comporta come un *offset* di $\sim 30 \text{ dyne/cm}$.

5.2 Misure sui substrati dopo 7 giorni dalla rimozione dello strato di ossido

Dopo la procedura illustrata in 1.2 si é lasciato crescere un nuovo strato di ossido in camera pulita per 7 giorni.

Nei campioni con l'ossido ricostruito si sono ottenuti i seguenti valori della componente dispersiva, della componente polare, dell'energia superficiale e dell'isteresi:

<i>Conc.</i> [atomi/cm^3]	<i>Tipo</i>	γ_s^d [dyne/cm]	γ_s^p [dyne/cm]	γ_s^{tot} [dyne/cm]	<i>Isteresi</i>
$(2.0 \pm 0.7) \cdot 10^{13}$	p	27 ± 4	6 ± 2	33 ± 5	$30^\circ \pm 3^\circ$
$6.48 \cdot 10^{14}$	n	27 ± 4	8 ± 2	35 ± 4	$34^\circ \pm 5^\circ$
$(8 \pm 2) \cdot 10^{14}$	p	30 ± 4	6 ± 2	36 ± 4	$22^\circ \pm 6^\circ$
$(10 \pm 3) \cdot 10^{14}$	p	25 ± 3	11.5 ± 1.6	37 ± 3	$24.9^\circ \pm 1.4^\circ$
$(2.1 \pm 0.7) \cdot 10^{15}$	p	29 ± 4	8 ± 2	37 ± 4	$24^\circ \pm 3^\circ$
$(3 \pm 2) \cdot 10^{15}$	n	26 ± 4	5.7 ± 1.8	32 ± 4	$31^\circ \pm 2^\circ$
$(4 \pm 3) \cdot 10^{15}$	p	23 ± 4	10 ± 3	34 ± 5	$24.8^\circ \pm 1.6^\circ$
$(1.6 \pm 0.3) \cdot 10^{16}$	p	29 ± 3	3.8 ± 1.1	33 ± 3	$32^\circ \pm 4^\circ$
$(7 \pm 7) \cdot 10^{16}$	n	28 ± 4	4.8 ± 1.4	33 ± 4	$24^\circ \pm 4^\circ$
$1.497 \cdot 10^{19}$	n	25 ± 3	11 ± 2	36 ± 4	$23^\circ \pm 3^\circ$

Si riportano di seguito i grafici delle misure in tabella in funzione della concentrazione di drogante (in scala semi-logaritmica):

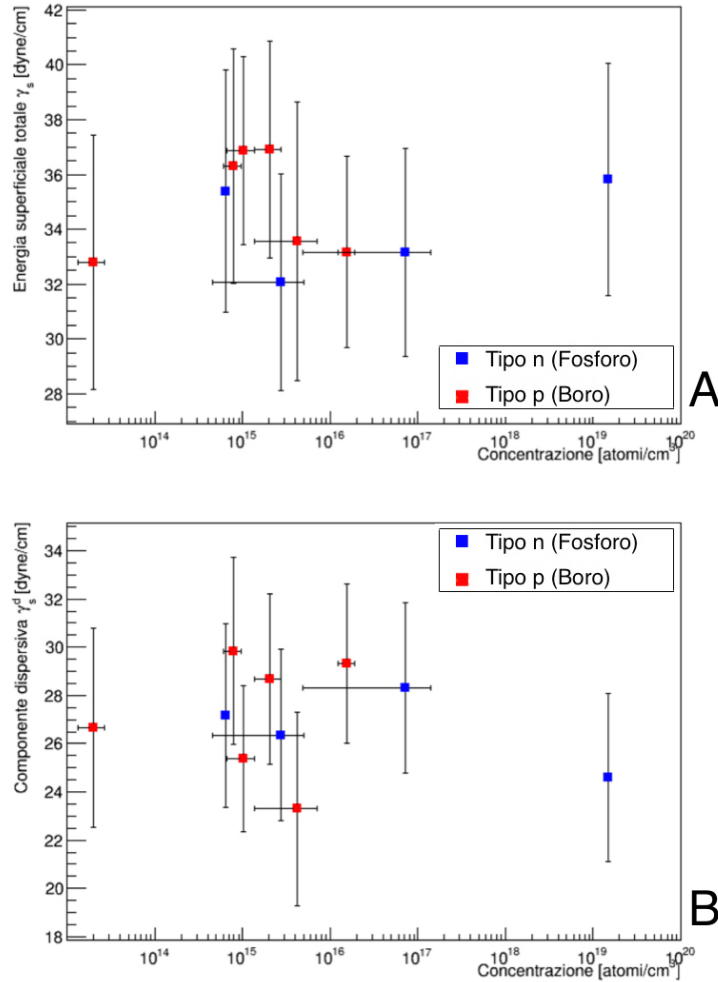


Figura 5.3: I grafici A e B mostrano rispettivamente l'energia superficiale γ_s e la componente dispersiva γ_s^d in funzione della concentrazione di drogante per i campioni con l'ossido ricostruito in 7 giorni.

Nel grafico in figura 5.3 A, che rappresenta γ_s in funzione della concentrazione di drogante, si osserva un lieve picco compreso fra 10^{15} e 10^{16} *atomi/cm³* composto esclusivamente da campioni di tipo *p* (rosso).

Questo picco raggiunge il suo massimo (37 ± 4 *dyne/cm*) con il campione $(2.1 \pm 0.7) \cdot 10^{15}$ *atomi/cm³*. I campioni di tipo *n*, all'aumentare della concentrazione, si dispongono con un andamento lievemente crescente, ad eccezione del campione $6.48 \cdot 10^{14}$ *atomi/cm³*.

Tuttavia, sia per i campioni *p* che *n*, γ_s non presenta variazioni significative e può essere approssimata da una costante (~ 35 *dyne/cm*).

Nel grafico in figura 5.3 B, che rappresenta γ_s^d in funzione della concentrazione di drogante, non si osserva né per i campioni di tipo p (rosso) né per i campioni di tipo n un andamento preferenziale, ma tutte le misure sono compatibili entro l'errore a ~ 27 *dyne/cm*.

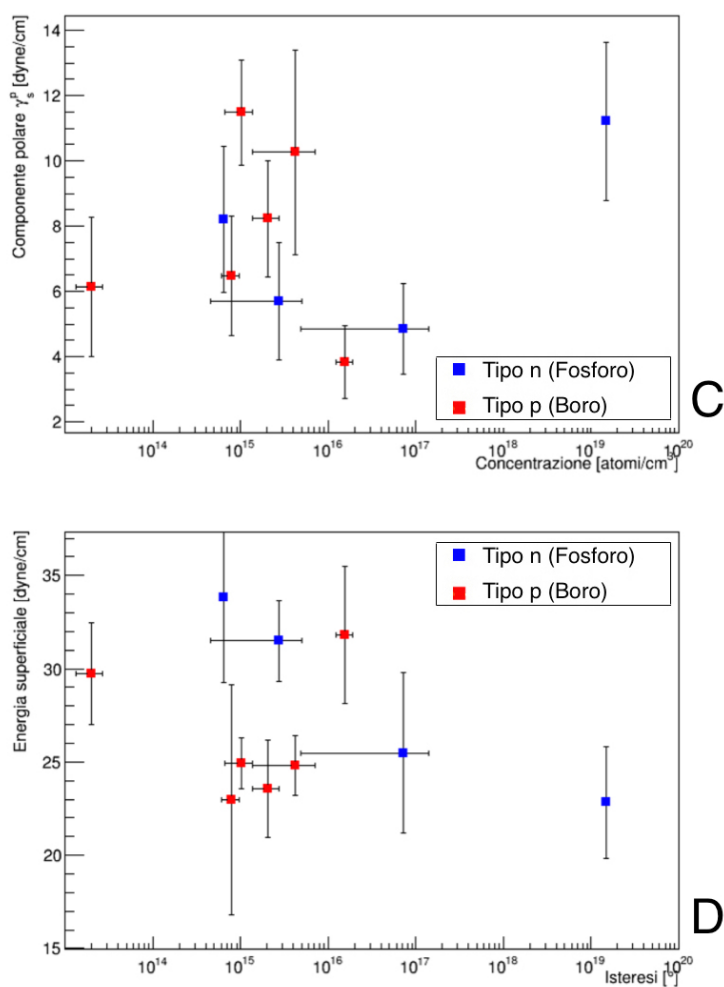


Figura 5.4: I grafici C e D mostrano rispettivamente la componente polare γ_s^p e l'isteresi delle superfici in funzione della concentrazione di drogante per i campioni con l'ossido ricostruito in 7 giorni.

Nel grafico in figura 5.4 C, che rappresenta γ_s^p in funzione della concentrazione di drogante, si osserva un picco compreso fra 10^{15} e 10^{16} *atomi/cm³* composto esclusivamente

da campioni di tipo p (rosso).

Questo picco raggiunge il suo massimo ($11.5 \pm 1.6 \text{ dyne/cm}$) con il campione $(2.1 \pm 0.7) \cdot 10^{15} \text{ atomi/cm}^3$.

Dei campioni di tipo n non si hanno abbastanza punti per individuare un andamento. Nel grafico in figura 5.4 D, che rappresenta l'isteresi in funzione della concentrazione di drogante, si osserva, per i campioni di tipo p (rosso), un minimo compreso fra 10^{15} e $10^{16} \text{ atomi/cm}^3$. I campioni di tipo n seguono un andamento decrescente.

Se consideriamo la media dei valori dell'isteresi otteniamo:

- Tipo $p = 26^\circ \pm 4^\circ$
- Tipo $n = 28^\circ \pm 5^\circ$
- Media pesata fra p e $n = 27^\circ \pm 3^\circ$

Rispetto ai campioni *as received*, l'andamento dell'energia superficiale totale non é sufficientemente influenzato dall'andamento della componente polare. Questo perché la componente polare assume valori molto piú bassi della componente dispersiva.

Prevale quindi l'andamento di γ_s^d , rendendo $\gamma_s \sim \text{costante}$.

5.3 Misure sui substrati dopo 21 giorni dalla rimozione dello strato di ossido

Dopo 21 giorni¹ dalla rimozione dello strato di ossido si sono ottenuti i seguenti valori della componente dispersiva, della componente polare, dell'energia superficiale e dell'isteresi:

$Conc. [\text{atomi/cm}^3]$	$Tipo$	$\gamma_s^d [\text{dyne/cm}]$	$\gamma_s^p [\text{dyne/cm}]$	$\gamma_s^{tot} [\text{dyne/cm}]$	$Isteresi$
$(2.0 \pm 0.7) \cdot 10^{13}$	p	16 ± 5	16 ± 2	32 ± 6	$34^\circ \pm 7^\circ$
$6.48 \cdot 10^{14}$	n	13 ± 2	30 ± 3	43 ± 4	$23^\circ \pm 2^\circ$
$(8 \pm 2) \cdot 10^{14}$	p	8 ± 2	56 ± 6	64 ± 6	$20^\circ \pm 3^\circ$
$(10 \pm 3) \cdot 10^{14}$	p	11 ± 2	44 ± 4	55 ± 5	$24^\circ \pm 2^\circ$
$(2.1 \pm 0.7) \cdot 10^{15}$	p	10 ± 2	49 ± 6	59 ± 6	$20^\circ \pm 2^\circ$
$(3 \pm 2) \cdot 10^{15}$	n	16 ± 3	16 ± 3	32 ± 5	$29^\circ \pm 4^\circ$
$(4 \pm 3) \cdot 10^{15}$	p	14 ± 3	29 ± 4	43 ± 5	$15^\circ \pm 7^\circ$
$(1.6 \pm 0.3) \cdot 10^{16}$	p	4 ± 1	32 ± 3	36 ± 4	$33^\circ \pm 6^\circ$
$(7 \pm 7) \cdot 10^{16}$	n	18 ± 4	19 ± 4	37 ± 6	$32^\circ \pm 5^\circ$
$1.497 \cdot 10^{19}$	n	15 ± 3	29 ± 4	44 ± 5	$28^\circ \pm 3^\circ$

¹Dopo i 7 giorni in camera pulita i campioni sono stati conservati per 14 giorni in laboratorio all'interno del porta campioni in figura 3.1

Si riportano di seguito i grafici delle misure in tabella in funzione della concentrazione di drogante (in scala semi-logaritmica):

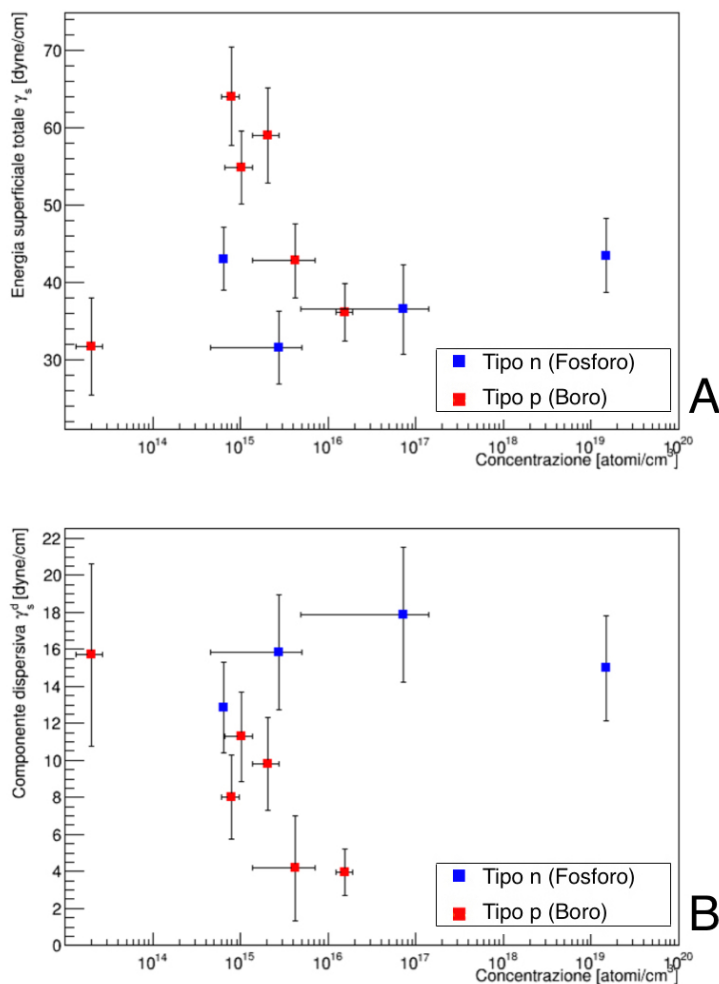


Figura 5.5: I grafici A e B mostrano rispettivamente l'energia superficiale γ_s e la componente dispersiva γ_s^d in funzione della concentrazione di drogante per i campioni con l'ossido ricostruito in 21 giorni

Nel grafico in figura 5.5 A, che rappresenta γ_s in funzione della concentrazione di drogante, si osserva un evidente picco nell'intorno di 10^{15} *atomi/cm³* composto esclusivamente da campioni di tipo *p* (rosso). Questo picco raggiunge il suo massimo (64 ± 6 *dyne/cm*) con il campione $(8 \pm 2) \cdot 10^{14}$ *atomi/cm³*.

I campioni di tipo n , all'aumentare della concentrazione, si dispongono con un andamento lievemente crescente, ad accezione del campione $6.48 \cdot 10^{14} \text{ atomi/cm}^3$.

Nel grafico in figura 5.5 B, che rappresenta γ_s^d in funzione della concentrazione di drogante, si osserva un andamento decrescente per i campioni di tipo p (rosso) e un andamento costante ($\sim 17 \text{ dyne/cm}$) per i campioni di tipo n .

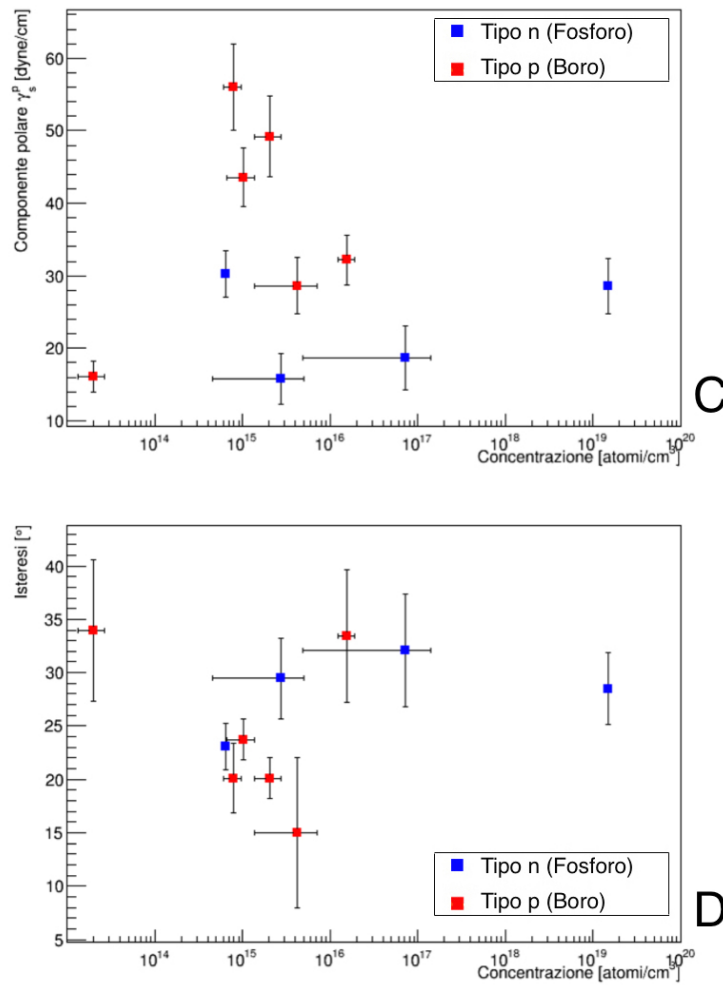


Figura 5.6: I grafici C e D mostrano rispettivamente la componente polare γ_s^p e l'isteresi delle superfici in funzione della concentrazione di drogante per i campioni con l'ossido ricostruito in 21 giorni

Nel grafico in figura 5.6 C, che rappresenta γ_s^p in funzione della concentrazione di

drogante, si osserva un evidente picco nell'intorno di 10^{15} *atomi/cm*³ composto esclusivamente da campioni di tipo *p* (rosso).

Questo picco raggiunge il suo massimo (56 ± 6 *dynes/cm*) con il campione $(8 \pm 2) \cdot 10^{14}$ *atomi/cm*³.

Nel grafico in figura 5.6 D, che rappresenta l'isteresi in funzione della concentrazione di drogante, si osserva un minimo per i campioni di tipo *p* (rosso) compreso fra 10^{15} e 10^{16} *atomi/cm*³. I campioni di tipo *n* seguono un andamento leggermente crescente.

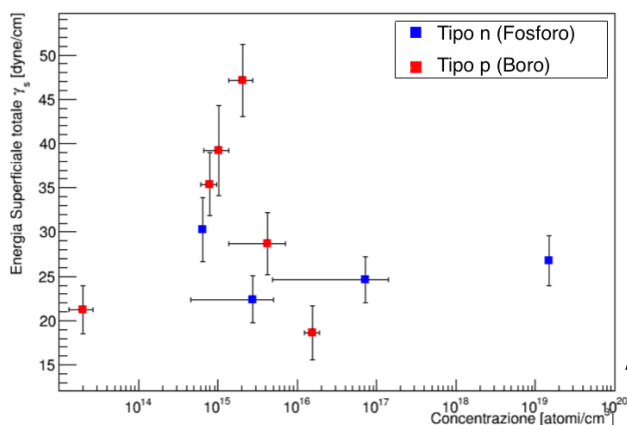
Se consideriamo la media dei valori dell'isteresi otteniamo:

- Tipo *p* = $24^\circ \pm 8^\circ$
- Tipo *n* = $28^\circ \pm 4^\circ$
- Media pesata fra *p* e *n* = $28^\circ \pm 3^\circ$

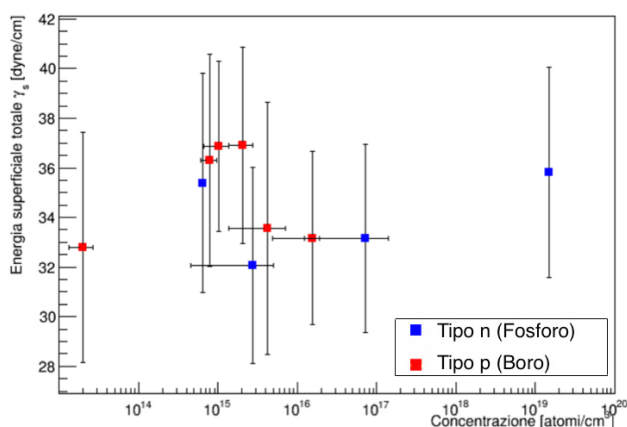
Si osserva che l'energia superficiale totale segue l'andamento della componente polare. Questo perché γ_s^p é molto maggiore di γ_s^d .

5.4 Discussione

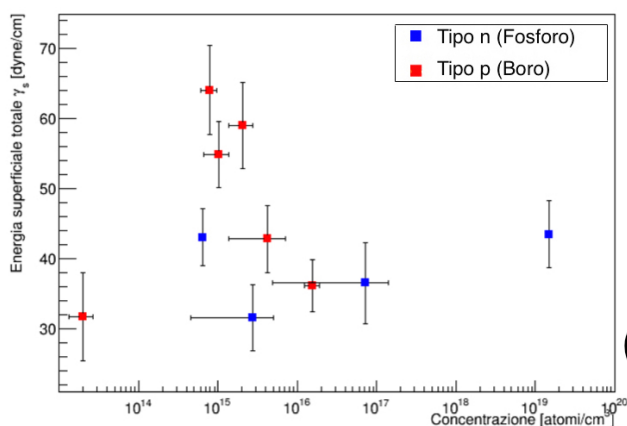
Confrontiamo i grafici dell'energia superficiale γ_s per i campioni *as received* e con l'ossido formato in 7 e 21 giorni:



A



B

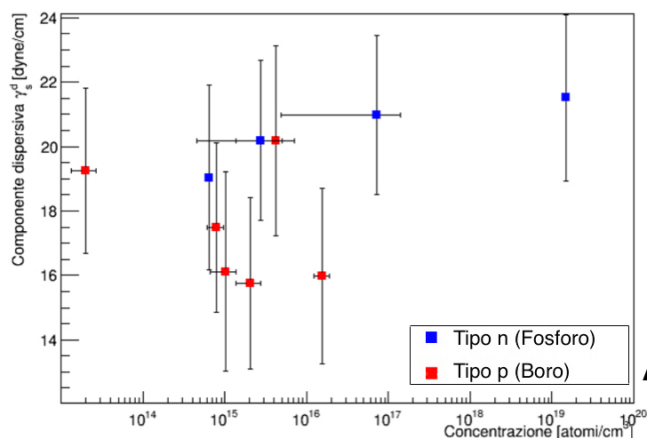


C

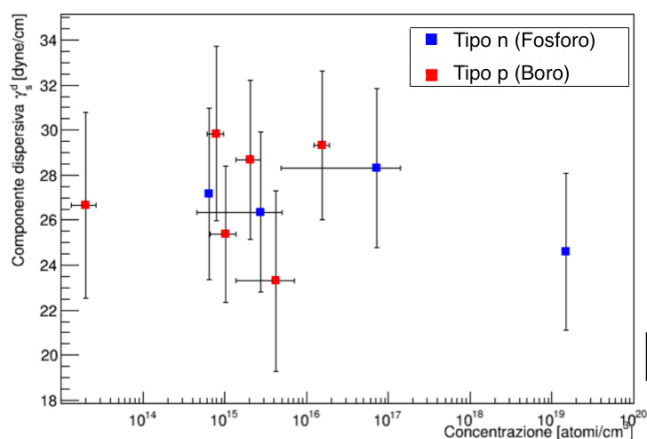
Nella figura a sinistra osserviamo che in tutti e tre i grafici di γ_s l'andamento rimane invariato, con un picco centrato fra 10¹⁵ e 10¹⁶ *atomi/cm*³ composto da campioni di tipo *p* (rosso). Cambia invece il valore che raggiunge tale picco.

Nei due campioni con l'ossido ricostruito notiamo che 7 giorni non sono sufficienti affinché l'energia superficiale nel picco si discosti sensibilmente dagli altri valori. Dopo 21 giorni invece si forma un picco evidente come per i campioni *as received*. I punti relativi ai campioni di tipo *n* (blu), pur mantenendo in tutti e tre i grafici un andamento leggermente crescente, non si discostano l'uno dall'altro abbastanza da poter individuare un andamento preferenziale.

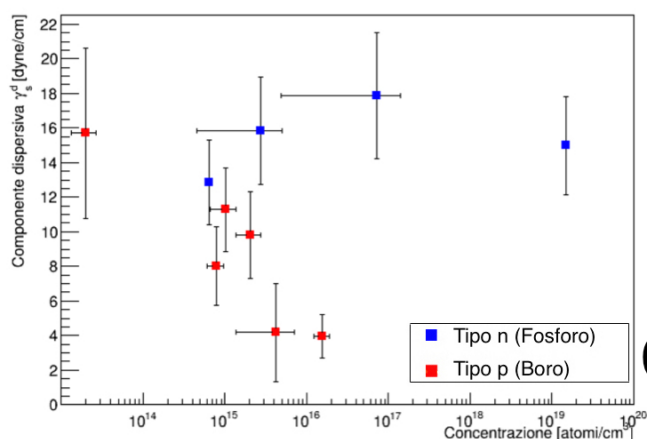
Confrontiamo i grafici della **componente dispersiva** per i campioni *as received* e con l'ossido formato in 7 e 21 giorni.



A



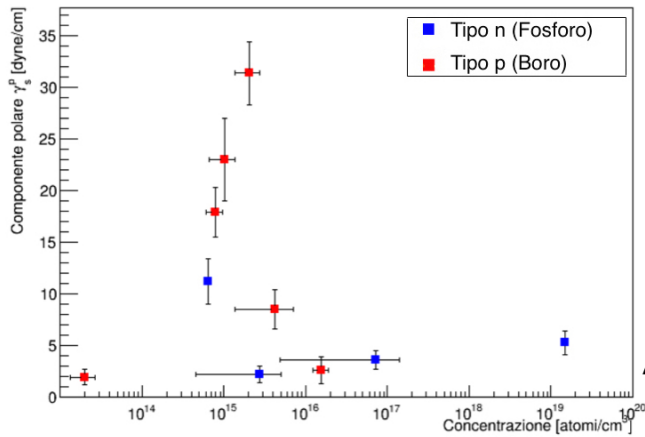
B



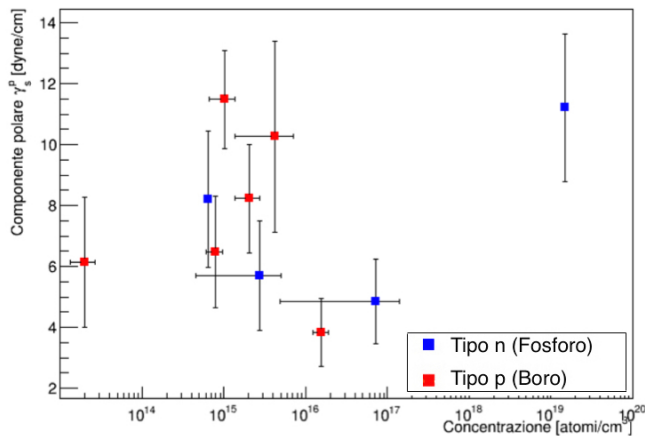
C

Nella figura a sinistra osserviamo che nei grafici A e B tutti i punti rimangono compatibili entro l'errore e quindi sono associabili a un valore costante, $\sim 20 \text{ dyne/cm}$ per il grafico A e $\sim 27 \text{ dyne/cm}$ per il grafico B. Nel grafico C invece, si osserva un andamento decrescente all'aumentare della concentrazione per i campioni di tipo *p*. In generale nei grafici A e B e per i punti *n* del grafico C, non si osservano variazioni sufficientemente grandi da poter individuare un andamento preferenziale di γ_s^d . Tuttavia nel grafico C si vede un andamento lineare decrescente per i campioni *p* al crescere della concentrazione, con una pendenza di $\sim -4 \text{ dyne/cm}$ per decade.

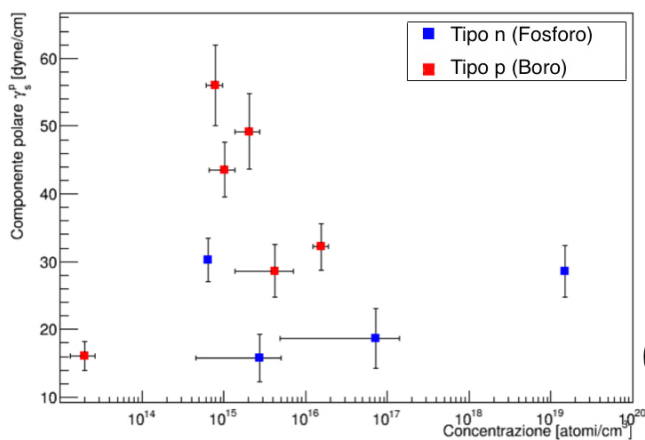
Confrontiamo i grafici della **componente polare** per i campioni *as received* e con l'ossido formato in 7 e 21 giorni:



A



B

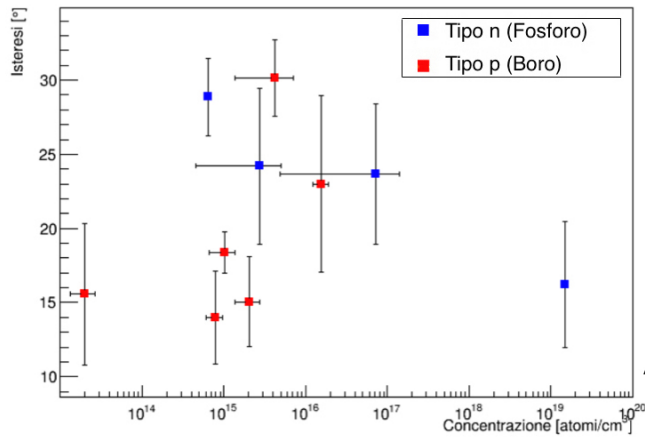


C

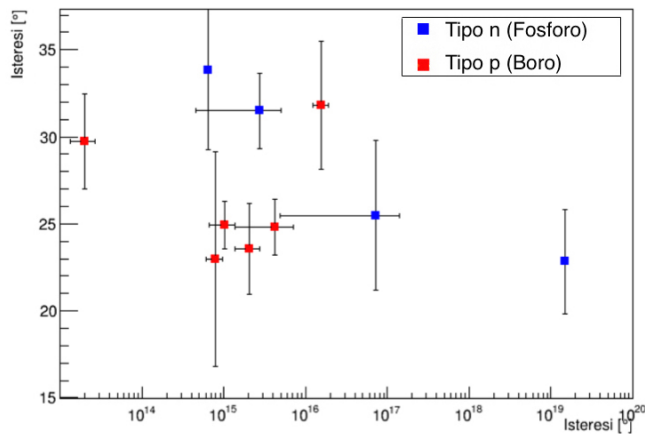
Nella figura a sinistra osserviamo che in tutti e tre i grafici di γ_s^p , l'andamento rimane invariato, con un picco centrato fra 10¹⁵ e 10¹⁶ *atomi/cm*³ composto da campioni di tipo *p* (rosso). Cambia invece il valore che raggiunge tale picco. Nei campioni con l'ossido ricostruito in 7 giorni, γ_s^p nel picco si discosta dagli altri punti molto meno di quanto non faccia per i campioni *as received* e con l'ossido dopo 21 giorni.

I punti relativi ai campioni di tipo *n* (blu), pur mantenendo in tutti e tre i grafici un andamento leggermente crescente, non si discostano l'uno dall'altro abbastanza da poter individuare un andamento preferenziale.

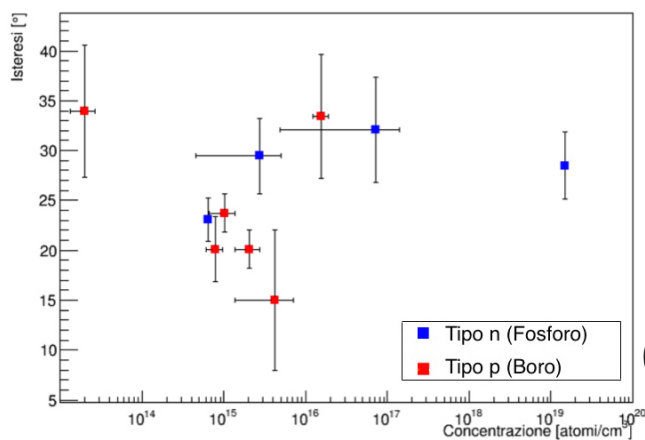
Confrontiamo infine i grafici dell'isteresi per i campioni *as received* e con l'ossido formato in 7 e 21 giorni:



A



B



C

Nella figura a sinistra osserviamo, in tutti e tre i grafici, che per i campioni *p* é presente di un minimo in corrispondenza di $10^{15} - 10^{16} \text{ atomi/cm}^3$. Per i campioni di tipo *n*, nei grafici A e B, osserviamo un andamento linearmente decrescente all'aumentare della concentrazione con una pendenza di $\sim -3^\circ$ per decade. Nel grafico C non si ritrova l'andamento decrescente dell'isteresi per i campioni *n*, ma i punti diventano compatibili fra loro entro l'errore. Confrontando il valore medio per i campioni *as received* e con l'ossido ricostruito in 7 e 21 giorni, osserviamo per questi ultimi un aumento dell'isteresi di $\sim 6^\circ$.

Capitolo 6

Simulazione numerica di una goccia

6.1 Legge di Young

La legge di Young [29]:

$$\gamma_{sv} - \gamma_{sl} = \gamma_{lv} \cos \theta \quad (6.1)$$

é la relazione fondamentale che lega l'energia superficiale all'angolo di contatto.

Per ricavarla si possono seguire due metodi equivalenti: si pone nulla la risultante delle forze nel punto triplo¹, oppure si calcola lo stato della goccia che minimizza l'energia libera di Gibbs.

Il primo metodo é il piú semplice e si basa sulla seguente proprietá della tensione superficiale:

$$\gamma = \frac{dF}{dL} \quad (6.2)$$

In questa equazione dF rappresenta la forza infinitesima che tiene uniti due lembi di superficie separati da un ipotetico taglio di lunghezza dL . Il versore di dF é tangente alla superficie e perpendicolare al taglio.

Se consideriamo γ costante su tutta la superficie, l'equazione precedente si semplifica in:

$$\gamma = \frac{F}{L} \quad (6.3)$$

Per verificare che con questa equazione si definisce l'energia superficiale in maniera equivalente alla definizione di energia superficiale come energia necessaria a creare un'interfaccia, si propone il seguente esempio [5]. Consideriamo un telaio rettangolare con un lato che può scorrere perpendicolare ai due adiacenti. Se dentro al telaio formiamo una superficie liquida, essa avrà una energia pari a due volte l'area del rettangolo² (Figura

¹Il punto triplo é la linea che divide le tre interfacce solido-vapore, liquido-vapore, liquido-solido. Si chiama triplo perché in esso si incontrano i tre stati della materia

²Le superfici sono due, una per ognuna delle due facce del rettangolo

6.1). Aumentando l'area del rettangolo, aumenteremo contemporaneamente l'energia della superficie di $\Delta E = 2\gamma \mathbf{dx} \cdot \mathbf{L}$.

L'energia aggiunta é data dalla forza³ sul lato mobile per il suo spostamento, ossia $\Delta E = 2\mathbf{F} \cdot \mathbf{dx}$.

Unendo queste due equazioni, otteniamo $\gamma = \frac{F}{L}$ che puntualmente diventa $\gamma = \frac{dE}{dL}$.

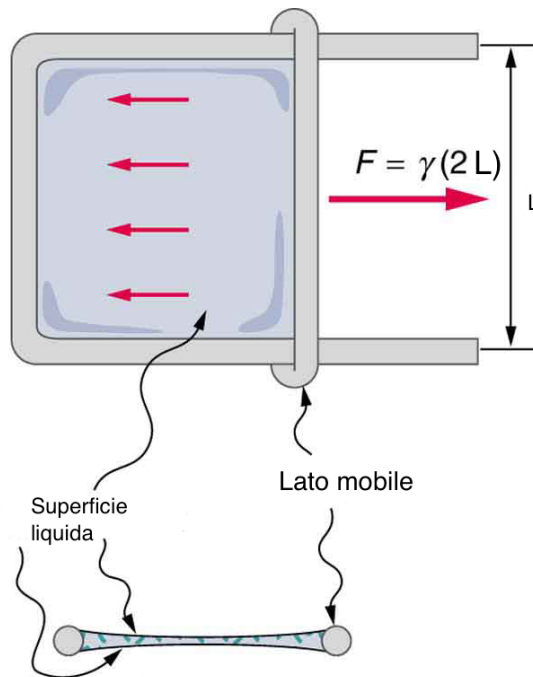


Figura 6.1:

Questo esempio si può applicare solo nel caso di superfici liquide, ma l'equazione 6.3 si può estendere anche alle interfacce solide.

Tornando alla legge di Young, sul punto triplo di una goccia poggiata su una superficie solida agiscono quattro forze date, secondo l'equazione 6.3 [29]:

1. La forza diretta verso l'interno dell'interfaccia solida $dF = \gamma_{sv} \cdot dL$
2. La forza diretta verso l'interno dell'interfaccia solido-liquido $dF = \gamma_{sl} \cdot dL$
3. La forza tangente dell'interfaccia liquida $dF = \gamma_{sv} \cdot dL$
4. La forza di adesione diretta verso l'interno del solido, forza che vincola la goccia sulla superficie.

³F si riferisce alla forza su un solo lembo

In queste equazioni dL rappresenta un tratto infinitesimo della linea tripla. Affinché la goccia sia in equilibrio, la risultante delle forze sulla linea tripla deve essere nulla.

Consideriamo la Figura 1. Lungo l'asse y agiscono 2 forze: la componente verticale di F_{lv} e la forza di adesione. La loro somma é sempre nulla, perché la goccia per ipotesi é vincolata sul piano. Lungo l'asse x invece agiscono 3 forze: F_{sl} , F_{sv} e la componente orizzontale di F_{lv} . Perché la goccia sia in equilibrio la risultante deve essere nulla, quindi :

$$F_{sl} - F_{sv} + \cos \theta F_{lv} = 0 \quad (6.4)$$

Andando a sostituire le rispettive espressioni (eq. 6.2) e dividendo per dl :

$$\gamma_{sl}dl - \gamma_{sv}dl + \cos \theta \gamma_{lv}dl = 0 \quad (6.5)$$

otteniamo esattamente la 2.1, che ha quindi una valenza puntuale.

Il secondo metodo per ricavare l'equazione di Young consiste nel calcolare la forma (a parità di volume) con cui la goccia minimizza l'energia libera di Gibbs [14]. In questo sistema l'energia di Gibbs, a meno di una costante, é data da:

$$E_{Gibbs} = Superficie_{lv}\gamma_{lv} + Superficie_{ls}\gamma_{ls} - Superficie_{sv}\gamma_{sv} \quad (6.6)$$

dove la superficie asciutta é posta a energia 0. Questo tipo di approccio permette di trovare l'intero profilo di una superficie e non solo l'angolo di contatto.

Si possono aggiungere anche altri termini all'energia di Gibbs, per esempio la *gravità*.

6.2 Simulazione

Per risolvere equazione 6.6 si é scelto un approccio numerico, implementando un programma C++ che ricostruisce il profilo di una goccia in 2 dimensioni ⁴.

Risolvendo il problema in due dimensioni, l'energia di Gibbs diventa:

$$E_{Gibbs} = Perimetro_{lv}\gamma_{lv} + Perimetro_{ls}\gamma_{ls} - Perimetro_{sv}\gamma_{sv} \quad (6.7)$$

ed é da minimizzare a parità di area. In questo contesto γ ha le dimensioni di una forza.

Il funzionamento del programma si sintetizza nel seguente modo.

Il profilo della goccia viene approssimato a un poligono a N vertici.

Ogni vertice-iesimo é descritto in coordinate polari⁵ in modo tale che l'intera goccia sia descritta da una raggiera di 180° e $\theta_{i+1} - \theta_i = d\theta = costante$.

⁴Il problema é analogo se ipotizziamo una simmetria sferica

⁵A ogni punto é associato un raggio $[0 - \infty]$ e un angolo $[0 - \pi]$

Il primo e l'ultimo vertice appartengono all'asse x e delimitano l'interfaccia solido-liquido, tutti gli altri punti invece delimitano l'interfaccia liquido-gas.

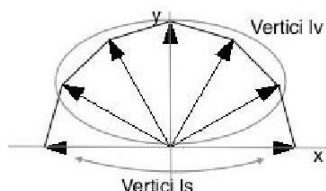


Figura 6.2: La figura mostra come viene approssimato il profilo di una goccia con una serie di punti disposti a raggiera

Inizialmente gli N punti vengono inizializzati a formare un poligono generico. Per ottenere la forma corretta della goccia, ogni punto va modificato in modo da minimizzare l'energia libera. L'unico modo in cui il programma può modificare il poligono è modificando il raggio dei vertici, a questo scopo ad ogni estremità è associato un $dRaggio$, che rappresenta la variazione infinitesima del raggio.

I $dRaggi$ però non sono assegnati casualmente, ogni vertice ha associato un $dRaggio$ tale che, se sottratto, modifica l'area totale di una costante $dArea$.

Di conseguenza, se si modifica qualsiasi raggio del proprio $dRaggio$, l'area finale del poligono sarà sempre la stessa, ossia l'area iniziale meno $dArea$. A ogni vertice è associato il $dRaggio$ secondo la seguente formula:

$$dRaggio_i = 2 \frac{dArea}{\sin d\theta (Raggio_{i-1} + Raggio_{i+1})} \quad (6.8)$$

dove $d\theta$ è l'angolo compreso fra due qualsiasi vertici.

Il funzionamento logico del programma si sviluppa all'interno di un doppio ciclo *for*, uno annidato dentro l'altro.

Il ciclo interno prosegue per tutti i vertici ed è organizzato in tre fasi: prima si sottrae $dRaggio_i$ al vertice-iesimo, poi si calcola l'energia totale del sistema modificato e se ne salva il valore in un array, infine si riassegnano le condizioni iniziali al sistema.

Terminato questo ciclo, nell'array sono raccolti tutti i valori dell'energia che avrebbe il sistema se si modificassero i vertici uno solo alla volta.

Il ciclo esterno serve a modificare il vertice per cui è minima l'energia del sistema e a ricalcolare le caratteristiche della goccia modificata. A questo scopo si trova il minimo dell'array dove sono raccolti i valori delle energie di ogni possibile cambiamento. Questo ciclo prosegue fin quando non si raggiunge l'area decisa.

Si scolpisce quindi da un poligono generico la forma corretta della goccia che minimizza l'energia libera.

Confrontando gli angoli delle gocce della simulazione numerica con gli angoli previsti dalla legge di Young, si vede che, maggiore é il numero di vertici utilizzati e minore é $dArea$, piú i risultati tendono a coincidere.

Con questo programma é possibile vedere anche l'effetto di un campo verticale costante sull'angolo di contatto.

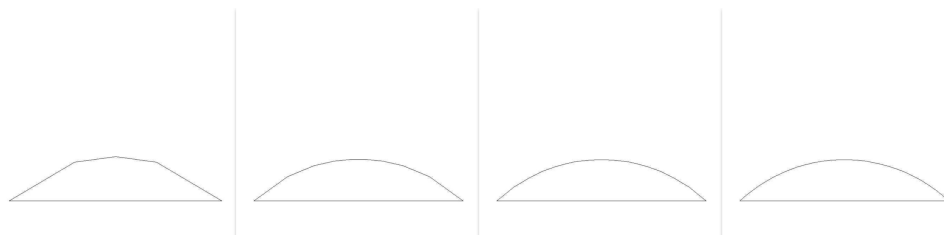


Figura 6.3: La Figura mostra una goccia 2D con angolo di contatto previsto 45° al crescere del numero di punti con cui viene approssimata (5-10-20-50 punti)

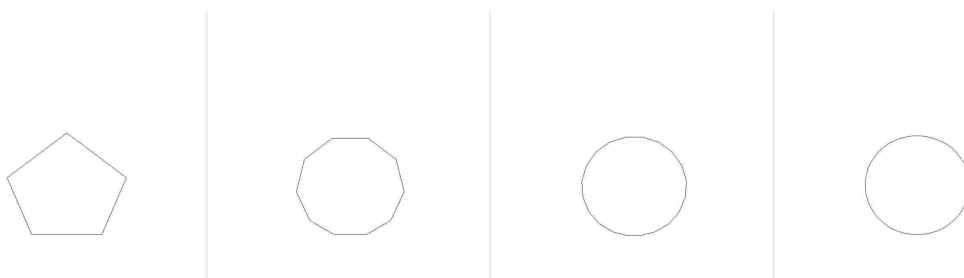


Figura 6.4: La Figura mostra una goccia 2D con angolo di contatto previsto 180° al crescere del numero di punti con cui viene approssimata (5-10-20-50 punti)

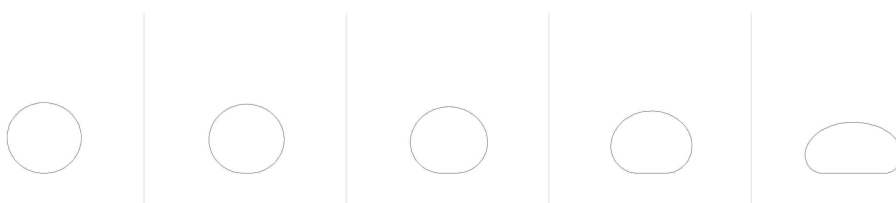


Figura 6.5: La Figura mostra come cambia il profilo di una goccia 2D al crescere del valore della gravitá. Si vede che l'angolo di contatto non cambia significativamente, é quindi accettabile trascurare la gravitá nella legge di Young

6.3 Stima dell'energia di Gibbs al variare dell'angolo di contatto

Con la simulazione numerica otteniamo la forma della goccia-2D, che a parità di volume minimizza l'energia libera di Gibbs. Vediamo inoltre che l'angolo di contatto che si ottiene coincide con l'angolo previsto dall'equazione di Young.

Facendo il ragionamento opposto possiamo quindi decidere i valori di γ_{lv} , γ_{sv} e γ_{ls} che formano una goccia con un angolo di contatto dato dall'equazione di Young.

Cerchiamo ora la stima dell'andamento dell'energia libera di Gibbs al variare di θ .

Per fare questo, a volume costante, poniamo $\gamma_{lv} = 1$ [dyne/cm], $\gamma_{sv} - \gamma_{ls} = \cos \theta$ [dyne/cm] .

Riportiamo ora il grafico dell'energia libera trovata con questi vincoli, utilizzando 18 angoli compresi fra 180° e 0° (Figura 6.6).

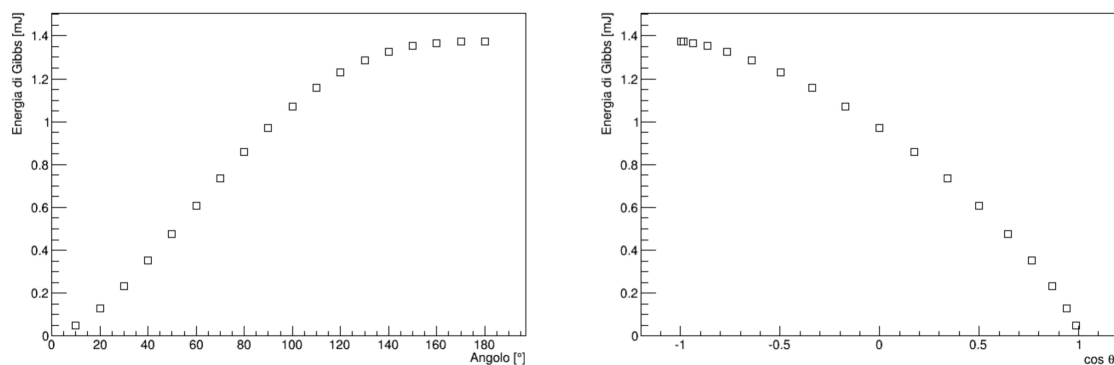


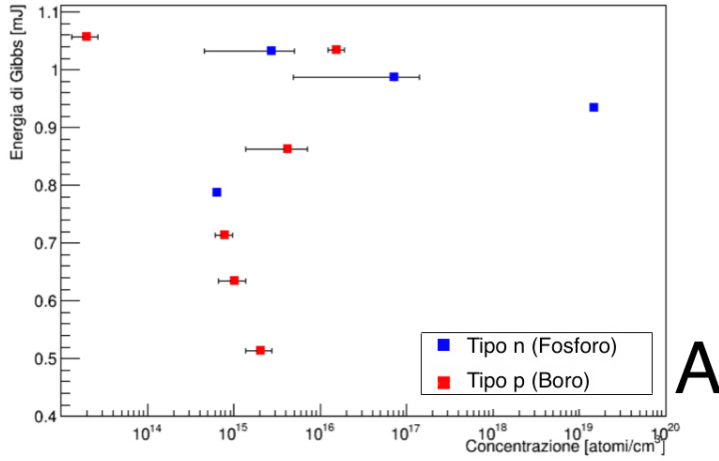
Figura 6.6: In figura é mostrata la stima dell'andamento dell'energia libera di Gibbs in funzione di θ e $\cos \theta$

Come possiamo vedere, l'energia di Gibbs tende a diminuire con un andamento quasi lineare al calare dell'angolo di contatto.

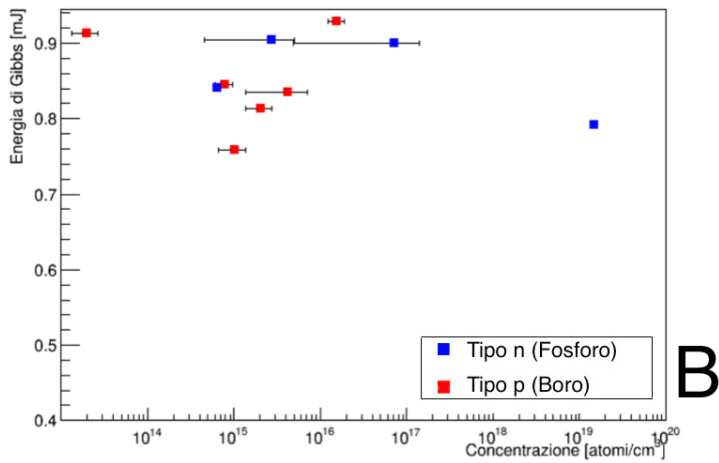
Tenendo in considerazione i grafici di Figura 6.6, si sono simulate le gocce sui vari substrati, utilizzando come angolo di riferimento quello misurato con l'acqua.

In questo modo si é potuto costruire il grafico dell'andamento dell'energia libera di una goccia di acqua, in funzione della concentrazione di drogante per i dieci campioni in tabella 3.1, sia *as received*, sia con l'ossido ricostruito dopo 7 e 21 giorni.

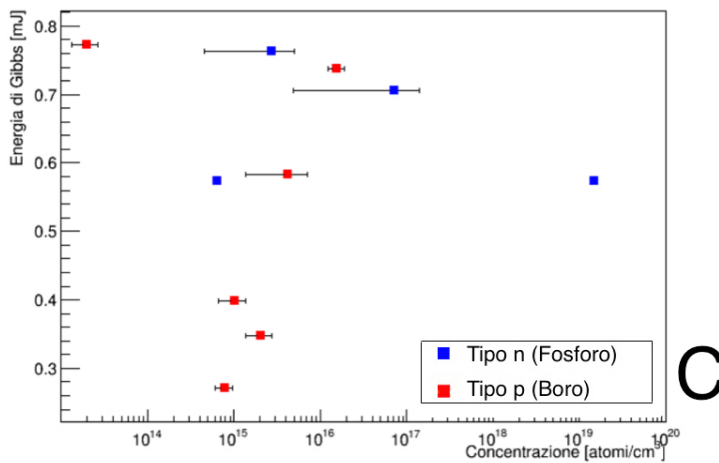
Si riportano di seguito i grafici dell'energia di Gibbs:



A



B



C

Si nota che i punti rossi, corrispondenti ai campioni drogati *p*, formano un minimo fra $10^{15} - 10^{16}$ *atomi/cm*³ in tutti e tre i grafici.

Tale minimo é accentuato per i campioni *as received* e con l'ossido ricostruito in 21 giorni, mentre é presente in maniera meno evidente nei campioni con l'ossido ricostruito in 7 giorni.

I punti blu, corrispondenti ai campioni di tipo *n*, si dispongono lungo una curva decrescente all'aumentare della concentrazione di drogante in tutti e tre i grafici ad, eccezione del campione meno drogato. Non si ha però un numero sufficiente di dati per affermare che il punto $6.48 \cdot 10^{14}$ *atomi/cm*³ appartiene al lato crescente di un minimo compreso fra $0 - 10^{15}$ *atomi/cm*³.

Si riportano in tabella 6.1 i coseni dell'angolo di contatto dell'acqua.

<i>Concentrazione atomi/cm³</i>	<i>Tipo</i>	<i>cos $\theta_{as\ received}$</i>	<i>cos $\theta_{dopo\ 7\ giorni}$</i>	<i>cos $\theta_{dopo\ 21\ giorni}$</i>
$(2.0 \pm 0.7) \cdot 10^{13}$	<i>p</i>	-0.15 ± 0.02	0.09 ± 0.05	0.29 ± 0.08
$6.48 \cdot 10^{14}$	<i>n</i>	-0.23 ± 0.03	0.20 ± 0.04	0.54 ± 0.05
$(8 \pm 2) \cdot 10^{14}$	<i>p</i>	0.37 ± 0.02	0.19 ± 0.03	0.84 ± 0.03
$(10 \pm 3) \cdot 10^{14}$	<i>p</i>	0.47 ± 0.04	0.311 ± 0.012	0.72 ± 0.02
$(2.1 \pm 0.7) \cdot 10^{15}$	<i>p</i>	0.606 ± 0.014	0.24 ± 0.03	0.77 ± 0.03
$(3 \pm 2) \cdot 10^{15}$	<i>n</i>	-0.10 ± 0.02	0.10 ± 0.04	0.31 ± 0.05
$(4 \pm 3) \cdot 10^{15}$	<i>p</i>	0.17 ± 0.03	0.21 ± 0.06	0.53 ± 0.03
$(1.6 \pm 0.3) \cdot 10^{16}$	<i>p</i>	-0.11 ± 0.04	0.07 ± 0.02	0.34 ± 0.03
$(7 \pm 7) \cdot 10^{16}$	<i>n</i>	-0.026 ± 0.018	0.11 ± 0.03	0.38 ± 0.06
$1.497 \cdot 10^{19}$	<i>n</i>	0.06 ± 0.02	0.27 ± 0.03	0.54 ± 0.03

Tabella 6.1: In tabella é riportato il coseno dell'angolo che forma l'acqua per i campioni *as received* e con l'ossido ricostruito dopo 7 e 21 giorni

Per comprendere il significato del minimo che si osserva nei grafici dell'energia di Gibbs consideriamo il seguente esperimento ipotetico ed esplicativo. Consideriamo un wafer di silicio tale che la concentrazione di drogante sia una funzione radiale decrescente all'aumentare del raggio che varia da un massimo di $\sim 10^{18}$ *atomi/cm³* a un minimo di $\sim 10^{12}$ *atomi/cm³*. Lasciamo poi crescere sulla superficie uno strato di ossido nativo. Se bagniamo completamente la superficie con acqua pura e lasciamo che si asciughi in un ambiente pulito, osserviamo il seguente fenomeno: man mano che l'acqua evapora, l'acqua rimasta si ritirerá verso le zone in cui la concentrazione di drogaggio é $\sim 10^{15}$ *atomi/cm³*, ossia formerá una corona circolare di acqua sulla superficie, a una distanza dal centro tale che la concentrazione sia $\sim 10^{15}$ *atomi/cm³*. Questo perché il sistema tende ad occupare lo stato che minimizza l'energia di Gibbs. Essendo presente un minimo nei grafici sopra riportati, tale stato si ottiene bagnando le zone dove la concentrazione é $\sim 10^{15}$ *atomi/cm³*.

Conclusioni

In questo lavoro di tesi abbiamo studiato la relazione esistente tra la concentrazione di drogante in un wafer di silicio e l'energia superficiale del suo ossido nativo.

La strumentazione utilizzata per la misura dell'energia superficiale é il tensiometro ottico, uno strumento semplice ma efficace per valutare le propriet  chimico-fisiche dell'interfaccia liquido-solido.

Il tensiometro ottico misura l'angolo di contatto statico e dinamico. La misura dell'angolo statico ci ha permesso di valutare l'energia superficiale dell'ossido nativo attraverso il metodo di Owen-Wendt; esso considera le componenti polare γ_s^p e dispersiva γ_s^d di una qualsivoglia superficie. L'energia superficiale dell'ossido nativo é stata misurata utilizzando quattro solventi puri a tensione superficiale decrescente: acqua ultrapura (72.8 dyne/cm), glicerolo (63.6 dyne/cm), glicole etilico (48.2 dyne/cm) e dimetilformamida (37 dyne/cm). Per valutare l'omogeneit  chimica/fisica dell'ossido abbiamo misurato l'isteresi dell'angolo di contatto in configurazione dinamica, utilizzando solamente acqua ultrapura.

Le misure di angolo statico e dinamico sono state realizzate su 10 frammenti di wafer di silicio a concentrazione crescente da 10^{13} a 10^{19} *atomi/cm*³ di entrambi i tipi di drogante (ossia di tipo *p* - boro - e di tipo *n* - fosforo -).

Nel capitolo 5 abbiamo visto che, per i substrati drogati con boro, l'energia superficiale presenta un picco corrispondente ad una concentrazione di $\sim 10^{15}$ *atomi/cm*³ nell'intervallo di concentrazione $2 \cdot 10^{13} - 1.6 \cdot 10^{16}$ *atomi/cm*³. Mentre i campioni drogati con fosforo presentano un andamento dell'energia superficiale leggermente crescente al crescere della concentrazione di drogaggio nell'intervallo di concentrazione $6.5 \cdot 10^{14} - 1.5 \cdot 10^{19}$ *atomi/cm*³.

Come spiegato nel capitolo 1, esiste un fattore di segregazione *m*, legato alla diffusione del drogante all'interno dell'ossido per una determinata concentrazione del silicio. Come suggerito dalla figura 1.1, una parte del drogante raggiunge l'interfaccia *SiO*₂ - *Aria*, modificandone l'energia superficiale. In particolare si é visto che, all'aumentare di *m*, cala la concentrazione di drogante che diffonde nell'ossido.

In tabella 1.1 vediamo che, per il boro, *m* é $\sim [0.1 - 0.3]$, mentre per il fosforo é ~ 10 . Quindi, a parit  di concentrazione, nei campioni di tipo *p* il boro diffonde nell'ossido in

quantità molto maggiore di quanto faccia il fosforo nei campioni di tipo n .

L'osservazione sperimentale che l'energia superficiale dell'ossido dipenda dalla concentrazione di drogante è avvalorata dal confronto fra la componente polare e dispersiva. Si osserva che la componente dispersiva γ_s^d , per tutti i campioni, non presenta variazioni significative, mentre la componente polare γ_s^p , presenta lo stesso picco osservato nell'energia superficiale.

Le impurità nell'ossido, determinate dagli atomi di drogante, conferiscono quindi polarità alla superficie aumentando l'energia superficiale totale.

Dal confronto fra le misure dei campioni *as received* con le misure dell'ossido ricostruito dopo 7 e 21 giorni di esposizione all'aria (Paragrafo 1.2), ricaviamo che gli atomi di drogante diffondono nel tempo e, in particolare, la polarità superficiale ritorna alle condizioni *as received* dopo 21 giorni dalla rimozione dell'ossido.

Dalle misure riportate nel paragrafo 5.2, dopo 7 giorni l'andamento dell'energia superficiale è sostanzialmente costante, mentre nei campioni dopo 21 giorni è compatibile con quello *as received*.

Dalla simulazione numerica di una goccia su una superficie, abbiamo compreso che il picco osservato nell'energia superficiale corrisponde ad un minimo dell'energia di Gibbs per i campioni di tipo p .

Infine, osserviamo che la misura dell'isteresi non evidenzia un andamento preferenziale al variare della concentrazione di drogante. Tuttavia essa aumenta il valore medio dell'isteresi per i campioni con ossido ricostruito rispetto ai campioni *as received*, ad indicare una possibile variazione dell'omogeneità chimico-fisica delle superfici.

Appendice A

Codice dei programmi Utilizzati

Si riporta il codice completo (in C++) dei programmi realizzati per l'analisi dei dati di questa tesi.

A.1 Analisi angolo statico

```
#include <iostream>
#include <cmath>
#include <fstream>
#include <time.h>
#include <TCanvas.h>
#include <TApplication.h>
#include <TUID.h>
#include <TGraphErrors.h>
#include <cstdlib>

using namespace std;
TApplication myapp("app",NULL,NULL);

double TrovaMassimo (double * ARRAY, int NumeroElementi){
double massimo = ARRAY[0];
for (int i=0; i<NumeroElementi;i++)
if (massimo<ARRAY[i]) massimo=ARRAY[i];
return massimo;
}
```

```

double TrovaMinimo(double * ARRAY, int NumeroElementi){
double minimo = ARRAY[0];
for(int i=0; i<NumeroElementi;i++)
if(minimo>ARRAY[i]) minimo=ARRAY[i];
return minimo;
}

```

```

double media(double * valori , int n){
double risultato=0;
for(int i=0;i<n;i++){
risultato = risultato + valori[i];
}
return double(risultato/n);
}

```

```

double deviazione_standard (double * valori , int n){
double risultato=0;
double m = media(valori , n);
for(int i=0;i<n;i++){
risultato = risultato + pow((valori[i]-m),2);
}
return sqrt(risultato/(n-1.));
}

```

```

double * media_pesata (double * valori , double * errori , int n)
{
double pesi[n];
double * risultato=new double [2];
risultato[0]=0;
risultato[1]=0;

for(int i=0;i<n;i++){
pesi[i]=1./pow(errori[i],2);
}

for(int i=0;i<n;i++){
risultato[0]=risultato[0]+valori[i]*pesi[i];
risultato[1]=risultato[1]+ pesi[i];
}
risultato[0]=risultato[0]/risultato[1];
}

```



```

risultato [1]=sqrt (1./( risultato [1]));
return risultato;
}

int main(){

char * NomeFile = new char [100];
cout<<" Write file 's name : "<<endl;
cin.getline(NomeFile, 100);

ifstream in(NomeFile, ios::in);

int NumeroDati;
double * AngoloSinistro = new double [1000];
double * AngoloDestro = new double [1000];

if (!in){
cout<<"ERROR : The file doesn't exist or it's impossible to
open"<<endl;
return 0;
}

in.ignore(5000, '\n');
in.ignore(5000, '\n');

for(int i=0; in>>NumeroDati>>AngoloSinistro [ i]>>AngoloDestro [ i
]; i++){
NumeroDati = i;
in.ignore(5000, '\n');
}
NumeroDati++;
cout<<" The Data in the file are: "<<NumeroDati<<endl;
cout<<"N$\circ \$"<<'\t'<<" LeftAngle"<<'\t'<<" RightAngle"<<endl;
for(int i=0; i<(NumeroDati); i++){
cout<< i+1 <<'\t'<<AngoloSinistro [ i]<<'\t'<<AngoloDestro [ i]<<
endl;}

double AngoloSinistroMedio = media(AngoloSinistro , NumeroDati);
double AngoloDestroMedio = media(AngoloDestro , NumeroDati);

```

```

double ErroreAngoloSinistro= deviazione_standard (AngoloSinistro
, NumeroDati);
double ErroreAngoloDestro= deviazione_standard (AngoloDestro ,
NumeroDati);
cout<<endl;
cout<<" Left angle : "<<AngoloSinistroMedio<<"+"<<
ErroreAngoloSinistro<<endl;
cout<<" cos Left angle : "<<cos ((AngoloSinistroMedio*2*M_PI)
/360.)<<"+"<<sqrt (pow (sin ((AngoloSinistroMedio*2*M_PI)
/360.)*(ErroreAngoloSinistro*2*M_PI)/360.,2))<<endl;
cout<<endl;

cout<<" Right angle : "<<AngoloDestroMedio<<"+"<<
ErroreAngoloDestro<<endl;
cout<<" cos Right angle : "<<cos ((AngoloDestroMedio*2*M_PI)
/360.)<<"+"<<sqrt (pow (sin ((AngoloDestroMedio*2*M_PI)/360.)
*(ErroreAngoloDestro*2*M_PI)/360.,2))<<endl;

double Angoli [] = { AngoloSinistroMedio , AngoloDestroMedio };
double Errori [] = { ErroreAngoloSinistro , ErroreAngoloDestro };

double * AngoloFinale = media_pesata (Angoli , Errori , 2);
cout<<endl;
cout<<" Weighted mean Angle " <<AngoloFinale[0]<<"+"<<
AngoloFinale[1]<<endl;
cout<<" Weighted mean cos Angle " <<cos ((AngoloFinale [0]*2*M_PI
)/360.)<<"+"<<sqrt (pow (sin ((AngoloFinale [0]*2*M_PI)/360.)*(
AngoloFinale [1]*2*M_PI)/360.,2))<<endl;

NumeroDati=NumeroDati*2;
double * TuttiGliAngoli = new double [NumeroDati];
for (int i=0; i<NumeroDati; i++){
if (i<NumeroDati/2){
TuttiGliAngoli [i] = AngoloSinistro [i];} else {TuttiGliAngoli [i] =
AngoloDestro [i-NumeroDati/2];}
}

double MediaTuttiGliAngoli = media (TuttiGliAngoli , NumeroDati)
;

```

```

double ErroreTuttiGliAngoli =deviazione_standard(TuttiGliAngoli
    , NumeroDati);
cout<<endl;
cout<<" Total angle : "<< MediaTuttiGliAngoli<<"+"<<
    ErroreTuttiGliAngoli<<endl;
cout<<" Total cos angle : "<<cos(( MediaTuttiGliAngoli*2*M_PI)
    /360.)<<"+"<<sqrt(pow(sin(( MediaTuttiGliAngoli*2*M_PI)
    /360.)*(ErroreTuttiGliAngoli*2*M_PI)/360.,2))<<endl;
cout<<endl;

    TCanvas Tela(" Angle 's Distribution", "Angle 's Distribution",
        700,700);
    TH1D histo(" Angles", "Distribuzione degli Angoli", 10,
        TrovaMinimo(TuttiGliAngoli , NumeroDati)-1, TrovaMassimo(
        TuttiGliAngoli , NumeroDati)+1);

    for (int i=0;i<NumeroDati;i++){
        histo.Fill(TuttiGliAngoli[i]); }

    histo.GetAxis()->SetTitle(" Angolo in gradi [#circ]");
    histo.GetAxis()->SetTitle(" Frequenza");
    histo.Fit(" gaus");
    histo.Draw();
    myapp.Run(true);
    Tela.Close();

return 0;
}

```

A.2 *Fit* Owen-Wendt

```

#include<iostream>
#include<fstream>
#include<cmath>
#include<time.h>
#include <TCanvas.h>
#include <TGraphErrors.h>
#include <TApplication.h>
#include <TF1.h>
#include<TAxis.h>

```

```

using namespace std;
TApplication * myapp = new TApplication ("app", NULL, NULL);

double Retta (double * x , double * par ){

    return x[0]*par[1]+par[0];
};

int main(){

    int NumeroElementi=0;
    double * cos = new double[40];
    double * cosError = new double[40];

    double * gammaP = new double[40];
    double * gammaPError = new double[40];
    double * gammaD = new double[40];
    double * gammaDError = new double[40];
    double * x = new double[40];
    double * xError = new double[40];
    double * y = new double[40];
    double * gammaTot= new double[40];
    double * gammaTotError = new double[40];
    double * yError = new double[40];
    double * Fattore = new double[10];

    ifstream in ("DatiOW", ios::binary);

    if (!in){
        cout<<"ERROR : The file doesn't exist or it's
            impossible to open"<<endl;
        return 0;
    }

    in.ignore(5000, '\n');

```

```

for (int i=0; i<<cos[i]>>cosError[i]>>gammaP[i]>>
    gammaPErr[i]>>gammaD[i]>>gammaDErr[i]>>gammaTot[
    i]>>gammaTotErr[i]; i++){
    NumeroElementi++;
    y[i] = ((gammaTot[i])*(cos[i]+1))/(2*sqrt(
        gammaD[i]));

    Fattore[0] = (gammaTotErr[i]*((cos[i]+1)*0.5)
        )/sqrt(gammaD[i]);
    Fattore[1] = (0.5*(gammaTot[i])*cosError[i])/
        sqrt(gammaD[i]);
    Fattore[2] = (gammaDErr[i]*gammaTot[i]*(cos
        [i]+1))/(4*gammaD[i]*sqrt(gammaD[i]));
    yError[i] = sqrt(pow(Fattore[0],2)+pow(Fattore
        [1],2) + pow(Fattore[2],2));

    x[i] = sqrt(gammaP[i]/gammaD[i]);
    Fattore[0] = 0.5*sqrt(gammaP[i]/pow(gammaD[i]
        ),3))*gammaDErr[i];
    Fattore[1] = (0.5*gammaPErr[i])/sqrt(gammaD[i]
        )*gammaP[i]);
    xError[i] = sqrt(pow(Fattore[0],2)+pow(Fattore
        [1],2));
    cout<<y[i]<<"+-"<<yError[i]<<'\t'<<x[i]<<"+-"<<
        xError[i]<<endl;
}

TCanvas * Tela = new TCanvas("OW", "owen-wendt");
TGraphErrors * grafico = new TGraphErrors(
    NumeroElementi, x, y, xError, yError);
TF1 * funz = new TF1("Retta", Retta, 0,2,2);

grafico->SetTitle("Regressione lineare");
grafico->Fit("Retta");
grafico->SetMinimum(0.);
cout<<endl;
cout<<" Componente Dispersiva : " << pow(funz->
    GetParameter(0),2)<<" + " <<2*funz->GetParError(0)*
    funz->GetParameter(0) <<endl;

```

```

cout<<" Componente polare : " << pow(funz->GetParameter
(1),2) <<" + " <<2*funz->GetParError(1)*funz->
GetParameter(1) <<endl;
cout<<"Tensione di Superficie Totale : " <<pow(funz->
GetParameter(0),2) + pow(funz->GetParameter(1),2)<<"
+ " <<sqrt(pow(2*funz->GetParError(0)*funz->
GetParameter(0),2) + pow(2*funz->GetParError(1)*funz->
->GetParameter(1),2))<<endl;

grafico->GetXaxis()->SetTitle("#sqrt{#gamma_{1}^{p}/#
gamma_{1}^{d}}");
grafico->GetYaxis()->SetTitle("#frac{#gamma_{1}(cos#
theta+1)}{2#sqrt{#gamma_{1}^{d}}} [dyne/cm]^{1/2}");

grafico->Draw("AP");
myapp->Run(true);
Tela->Close();
return 0;
}

```

A.3 Analisi angoli di isteresi

```

#include<iostream>
#include<cmath>
#include<fstream>
#include<time.h>
#include <TCanvas.h>
#include <TApplication.h>
#include <TID.h>
#include <TGraphErrors.h>
#include <cstdlib>

using namespace std;
TApplication myapp("app",NULL,NULL);

double Media(double * valori , int inizio , int fine){
double risultato=0;
for(int i=inizio; i<=fine; i++){

```

```

        risultato = risultato + valori[i];
    }
    return double(risultato/(fine-inizio+1));
}

double DeviazioneStandard(double * valori, int inizio, int fine
){
    double risultato=0;
    double m = Media(valori, inizio, fine);
    for(int i=inizio; i<=fine; i++){
        risultato = risultato + pow((valori[i]-m),2);
    }

    return sqrt(risultato/(fine-inizio));
}

int main(){

    char * NomeFile = new char[100];
    cout<<" Write file 's name : "<<endl;
    cin.getline(NomeFile, 100);

    ifstream in(NomeFile, ios::in);

    int NumeroDati;
    double * AngoloSinistro = new double[1000];
    double * AngoloDestro = new double[1000];
    double * AngoloTotale = new double[1000];
    double * Tempi = new double[1000];

    if (!in){
        cout<<"ERROR : The file doesn't exist or it's
            impossible to open"<<endl;
        return 0;
    }

    in.ignore(5000, '\n');
    in.ignore(5000, '\n');
}

```

```

in.ignore(5000, '\n');

for(int i=0; i<<NumeroDati>>AngoloSinistro[i]>>
    AngoloDestro[i]>>AngoloTotale[i]>>Tempi[i]>>Tempi[i
]>>Tempi[i]; i++){
    NumeroDati = i;
    in.ignore(5000, '\n');
}

NumeroDati++;

cout<<" The Data in the file are: "<<NumeroDati<<endl;

cout<<"N"<<'\t'<<"LeftAngle (degree)"<<'\t'<<"
    RightAngle (degree)"<<'\t'<<"Total Angle (degree)
"<<'\t'<<"Tempo sec"<<endl;
for(int i=0; i<NumeroDati; i++){
    cout<< i+1 <<'\t'<<'\t'<<AngoloSinistro[i]<<'\t'
        '<<'\t'<<AngoloDestro[i]<<'\t'<<'\t'<<'\t'<<
        AngoloTotale[i]<<'\t'<<'\t'<<'\t'<<Tempi[i
]<<endl;}

double TParagone = sqrt(pow( DeviazioneStandard(
    AngoloTotale, 0, 2),2)+pow(DeviazioneStandard(
    AngoloTotale, 2, NumeroDati-1),2));
double Centro = 2;
double IntervalloDecrescita[2];
double minimo;

for(int i=2; i<NumeroDati-2;i++){
    minimo = sqrt(pow( DeviazioneStandard(
        AngoloTotale, 0,i),2)+pow(DeviazioneStandard(
        AngoloTotale, i, NumeroDati-1),2));
    if(minimo<=TParagone) {TParagone = minimo;
        Centro = i;} //trovo centro
}

TParagone = DeviazioneStandard(AngoloTotale, 0,Centro);

for(int i=0; i<Centro-2; i++){

```



```

        minimo = DeviazioneStandard(AngoloTotale , 0,
        Centro-i);
        if(minimo<=TParagone) {TParagone = minimo;
        IntervalloDecrescita [0] = i;} else { break
        ;} //trovo intervallo angolo avanzamento
    }

TParagone = DeviazioneStandard(AngoloTotale , Centro ,
    NumeroDati-1);

for(int i=0; i<(NumeroDati-Centro)-2; i++){
    minimo = DeviazioneStandard(AngoloTotale ,
    Centro+i , NumeroDati-1);
    if(minimo<=TParagone) {TParagone = minimo;
    IntervalloDecrescita [1] = i;} else { break;}
    //trovo intervallo angolo di recessione
}

cout<<"Per trovare l'angolo di avanzamento utilizzo i
    punti da : " <<0<<" a "<<Centro-IntervalloDecrescita
    [0]<<endl;
cout<<"Per trovare l'angolo di Recessione utilizzo i
    punti da : " <<Centro+IntervalloDecrescita[1]<<" a
    "<< NumeroDati-1<<endl;

double AA [] = {Media(AngoloTotale ,0 ,Centro-
    IntervalloDecrescita [0]) , DeviazioneStandard(
    AngoloTotale , 0 ,Centro-IntervalloDecrescita [0]) };
double AR[] = {Media(AngoloTotale , Centro+
    IntervalloDecrescita [1] , NumeroDati-1) ,
    DeviazioneStandard(AngoloTotale , Centro+
    IntervalloDecrescita [1] , NumeroDati-1)};

cout<<"Angolo avanzamento : "<<AA[0]<<" + "<<AA[1]<<
    endl;
cout<<"Angolo Recessione : "<<AR[0]<<" + "<<AR[1]<<
    endl;
cout<<"Isteresi : " << AA[0]-AR[0] <<"+"<<sqrt(AA[1]*
    AA[1]+AR[1]*AR[1])<<endl;

```

```

TCanvas Tela("Isteresi","Isteresi", 700,700);
TGraph grafico(NumeroDati, Tempi, AngoloTotale);
grafico.SetTitle("Isteresi");
grafico.Draw("AL*");

grafico.GetAxis()->SetTitle("Tempo [s]");
grafico.GetAxis()->SetTitle("Angolo in gradi [#circ]");
;
//histo.Fit("gaus");
//histo.Draw();
myapp.Run(true);

return 0;
}

```

A.4 Simulazione numerica di una goccia su una superficie

vertice.h

```

#include<iostream>
#include<cmath>
#include<fstream>
#include<time.h>

using namespace std;

class Vertice{

    double Teta, Raggio, dRaggio;
    int Tipo;

public:
    Vertice() {}

    Vertice(double A, double R, int T){
        Teta = A;

```

```

Raggio = R;
Tipo = T;
}

void Inizializza(double angolo, double raggio, int T){
Teta = angolo;
Raggio=raggio;
Tipo = T;}

void SetTipo(int T){
Tipo=T;
}

void SetRaggio(double R){
Raggio = R;
}

void SetdRaggio (double dR){
dRaggio = dR;
}

double GetRaggio(){
return Raggio;
}

double GetTeta(){
return Teta;
}

double GetdRaggio(){
return dRaggio;
}

double GetTipo(){
return Tipo;
}

double GetX(){
return Raggio*cos(Teta);}

```

```

double GetY() {
return  Raggio*sin(Teta);}

void stampa() {
cout<<" Teta : "<<Teta<<'\t'<<"Raggio : "<< Raggio<< " dRaggio
: "<<dRaggio<<endl;
}
};

```

confine.h

```
#include" Vertice.h"
```

```

class Confine {

    int NumeroVertici;
    Vertice ** Vertici;
    Vertice ** VerticiP;
    Vertice ** VerticiS;
    double Perimetro1;
    double Perimetro0;
    double Area;
    double dArea;
    double dTeta;

void InizializzaVerticiSP () {
    VerticiP [0] = Vertici [NumeroVertici -1];
    VerticiS [0] = Vertici [1];
    VerticiP [NumeroVertici -1] = Vertici [
        NumeroVertici -2];
    VerticiS [NumeroVertici -1] =
        Vertici [0];
    for (int i=1; i<NumeroVertici -1; i++){
        VerticiP [i] = Vertici [i -1];
        VerticiS [i] = Vertici [i +1];}

}

```

```

public :
Confine () {}

```

```

~Confine () {
    delete Vertici;
    delete VerticiP;
    delete VerticiS;
}

Confine (int NV){
    dTeta = (M_PI)/ (NV-1);
    NumeroVertici = NV;

    Vertici = new Vertice * [NV];
    VerticiP = new Vertice * [NV];
    VerticiS = new Vertice
        * [NV];
}

Vertice * GetVertice(int indice){
    return Vertici[indice];
}

int GetNumeroVertici () {
    return NumeroVertici;
}

double GetPerimetro1 () {
    return Perimetro1;
}

double GetPerimetro0 () {
    return Perimetro0;
}

double GetArea () {
    return Area;
}

double GetdArea () {
    return dArea;
}

```

```

}

double GetdTeta(){
    return dTeta;
}

void SetdArea(double dA){
    dArea = dA;
}

void SetArea(double A){
    Area = A;
}

void SetPerimetro0(double P0){
    Perimetro0=P0;
}

void SetPerimetro1(double P1){
    Perimetro1=P1;
}

void AggiungiVertice (double raggio , int indice){
    if(indice >= NumeroVertici) cout<<"Errore"<<endl;

    if(indice>0 && indice<(NumeroVertici-1)){
        Vertici[indice] = new Vertice(indice*dTeta ,
            raggio , 0);} else {
            Vertici[indice] = new Vertice(indice*
                dTeta , raggio , 1);}

        if(indice == NumeroVertici-1)
            InizializzaVerticiSP ();
}

void TrovadRaggio (int indice){
    if(indice == 0){

```

```

Vertici [ indice ]->SetdRaggio ( 2*dArea / ( sin ( dTeta )
* Vertici [ 1 ]->GetRaggio ( ) ) ); } else {
    if ( indice == ( NumeroVertici - 1 ) ) {
        Vertici [ indice ]->SetdRaggio ( 2*
        dArea / ( sin ( dTeta ) * Vertici [
        indice - 1 ]->GetRaggio ( ) ) ); }
        else {
            Vertici [ indice ]->
            SetdRaggio ( 2*dArea / (
            sin ( dTeta ) * ( VerticiP
            [ indice ]->GetRaggio
            ( ) + VerticiS [ indice
            ]->GetRaggio ( ) ) ) ); } }
}

void AggiornaIdRaggi ( int indice ) {

    if ( indice == 0 ) {
        TrovadRaggio ( 1 ) ;
        TrovadRaggio ( NumeroVertici - 1 ); } else {
        if ( indice == NumeroVertici - 1 ) {
            TrovadRaggio ( 0 ) ;
            TrovadRaggio ( NumeroVertici - 2 ); }
            else {
                TrovadRaggio ( indice + 1 )
                ;
                TrovadRaggio ( indice - 1 )
                ; } }
}

double Gravita ( int indice , double g ) {

    return g*dArea*( 2* Vertici [ indice ]->GetY ( ) +
    VerticiP [ indice ]->GetY ( ) + VerticiS [ indice ]->
    GetY ( ) - Vertici [ indice ]->GetdRaggio ( ) * sin (
    Vertici [ indice ]->GetTeta ( ) ) ) / 4. ;
}

void AggiornaPerimetro ( int indice ) {

```

```

if(Vertici[indice]->GetTipo() == 1){
Perimetro1 = Vertici[0]->GetX() - Vertici[
NumeroVertici-1]->GetX();
    if(indice == 0){
        Perimetro0 = Perimetro0 - sqrt(
            pow(Vertici[indice]->
                GetRaggio()-Vertici[indice
                ]->GetdRaggio(),2) + pow(
                VerticiS[indice]->GetRaggio
                (),2) - 2*cos(dTeta)*(
                Vertici[indice]->GetRaggio()
                -Vertici[indice]->GetdRaggio
                ())*VerticiS[indice]->
                GetRaggio());
        Perimetro0 = Perimetro0 + sqrt(
            pow(Vertici[indice]->
                GetRaggio(),2) + pow(
                VerticiS[indice]->GetRaggio
                (),2) - 2*cos(dTeta)*Vertici
                [indice]->GetRaggio()*
                VerticiS[indice]->GetRaggio
                ());}
    if(indice == NumeroVertici-1){
        Perimetro0 = Perimetro0 - sqrt(
            pow((Vertici[indice]->
                GetRaggio()-Vertici[indice
                ]->GetdRaggio()),2) + pow(
                VerticiP[indice]->GetRaggio
                (),2) - 2*cos(dTeta)*(
                Vertici[indice]->GetRaggio()
                -Vertici[indice]->GetdRaggio
                ())*VerticiP[indice]->
                GetRaggio());
        Perimetro0 = Perimetro0 + sqrt(
            pow(Vertici[indice]->
                GetRaggio(),2) + pow(
                VerticiP[indice]->GetRaggio
                (),2) - 2*cos(dTeta)*Vertici
                [indice]->GetRaggio()*
                VerticiP[indice]->GetRaggio

```



```

        ());}
    }else{
        Perimetro0 = Perimetro0 - sqrt(pow(Vertici[
            indice]->GetRaggio()-Vertici[indice]->
            GetdRaggio(),2) + pow(VerticiS[indice]->
            GetRaggio(),2) - 2*cos(dTeta)*(Vertici[
            indice]->GetRaggio()-Vertici[indice]->
            GetdRaggio())*VerticiS[indice]->GetRaggio())
        ;
        Perimetro0 = Perimetro0 + sqrt(pow(Vertici[
            indice]->GetRaggio(),2) + pow(VerticiS[
            indice]->GetRaggio(),2) - 2*cos(dTeta)*
            Vertici[indice]->GetRaggio()*VerticiS[indice
            ]->GetRaggio());

        Perimetro0 = Perimetro0 - sqrt(pow((Vertici[
            indice]->GetRaggio()-Vertici[indice]->
            GetdRaggio()),2) + pow(VerticiP[indice]->
            GetRaggio(),2) - 2*cos(dTeta)*(Vertici[
            indice]->GetRaggio()-Vertici[indice]->
            GetdRaggio())*VerticiP[indice]->GetRaggio())
        ;
        Perimetro0 = Perimetro0 + sqrt(pow(Vertici[
            indice]->GetRaggio(),2) + pow(VerticiP[
            indice]->GetRaggio(),2) - 2*cos(dTeta)*
            Vertici[indice]->GetRaggio()*VerticiP[indice
            ]->GetRaggio());}
    }

void TrovaPerimetro0(){
    Perimetro0=0;
    for(int i=0;i<NumeroVertici-1;i++)
        Perimetro0 = Perimetro0 + sqrt(pow(
            Vertici[i]->GetRaggio(),2) + pow(
            VerticiS[i]->GetRaggio(),2) - 2*cos(
            dTeta)*Vertici[i]->GetRaggio()*
            VerticiS[i]->GetRaggio());
    }
}

```

```

void TrovaPerimetro1 () {
    Perimetro1 = ( Vertici[0]->GetX() - Vertici[
        NumeroVertici-1]->GetX() );
}

```

```

void TrovaArea () {
    Area=0;
    for (int i=0;i<NumeroVertici-1;i++)
        Area = Area + sin(dTeta)*Vertici [ i]->GetRaggio
            ()*VerticiS [ i]->GetRaggio ();

        Area = Area / 2.;
    }

};

```

drop_shape.h

```

#include "Confine.h"
#include <iostream>
#include <TCanvas.h>
#include <TApplication.h>
#include <TH1D.h>
#include <TGraphErrors.h>
#include <cstdlib>
#include <Tpolyline.h>

void InizializzaVertici(Confine * Bordo){

    for (int i=0; i<Bordo->GetNumeroVertici(); i++)
        Bordo->AggiungiVertice (0.8, i);

}

int TrovaMinimo(double * ARRAY, int ElementiArray){
    double minimo=ARRAY[0];
    int coordinata_minimo=0;

```

```

        for (int i=0; i<ElementiArray; i++)
            if (ARRAY[ i]<minimo) {
                minimo=ARRAY[ i ];
                coordinata_minimo=i;
            }

        return coordinata_minimo;

void CostruisciGoccia (double PAdA, double Area, Confine *
    BORDO, double cos, double gravita=0){

double * ENERGIA = new double [BORDO->GetNumeroVertici() ];
    double gamma_LS = -cos;
    double gamma_LV = 1;

double RaggioPrecedente;
    double PerimetroPrecedente0;
    double PerimetroPrecedente1;

int K;

    BORDO->TrovaArea();
    BORDO->TrovaPerimetro0();
    BORDO->TrovaPerimetro1();
    BORDO->SetdArea(PAdA*Area*0.01);

    for (int i=0; i<BORDO->GetNumeroVertici(); i++)
        BORDO->TrovaRaggio(i);

while (BORDO->GetArea() > Area){

    for (int i=0; i<BORDO->GetNumeroVertici(); i++){
        PerimetroPrecedente0 = BORDO->GetPerimetro0();
        PerimetroPrecedente1 = BORDO->
            GetPerimetro1();
    }
}

```

```

        RaggioPrecedente = BORDO->
            GetVertice(i)->GetRaggio();

BORDO->GetVertice(i)->SetRaggio(BORDO->
    GetVertice(i)->GetRaggio() + BORDO->
    GetVertice(i)->GetdRaggio());

BORDO->AggiornaPerimetro(i);
ENERGIA[i] = gamma_LV*BORDO->GetPerimetro0()+
    gamma_LS*BORDO->GetPerimetro1() + BORDO->
    Gravita(i, gravita);

BORDO->SetPerimetro0(PerimetroPrecedente0);
    BORDO->SetPerimetro1(
        PerimetroPrecedente1);
        BORDO->GetVertice(i)->SetRaggio
            (RaggioPrecedente);
            }

K = TrovaMinimo (ENERGIA, BORDO->GetNumeroVertici());

BORDO->GetVertice(K)->SetRaggio(BORDO->GetVertice(K)->
    GetRaggio() + BORDO->GetVertice(K)->GetdRaggio());

BORDO->AggiornaPerimetro(K);
    BORDO->AggiornaIdRaggi(K);
        BORDO->SetArea(BORDO->GetArea()+BORDO->
            GetdArea());
BORDO->SetdArea(PAdA*Area*0.01);

}

delete [] ENERGIA;

```

```

double ENERGIA (Confine * BORDO, double cos){
    double gamma_LS = -cos;
    double gamma_LV = 1;

```

```

        return gamma_LV*BORDO->GetPerimetro0()+ gamma_LS*BORDO
            ->GetPerimetro1();
    }

int main(){

    int NV;
    cout<<"inserire numero Vertici : "<<endl;
    cin>> NV;

    Confine * BORDO = new Confine(NV);
    InizializzaVertici(BORDO);

    double cos;
    cout<<"inserire cos"<<endl;
    cin>>cos;

    CostruisciGoccia(-0.0005, 0.15, BORDO, cos);

    cout<<"angolo = "<<360*atan((BORDO->GetVertice(1)->GetY
        ()-BORDO->GetVertice(0)->GetY())/ (BORDO->GetVertice
        (1)->GetX()-BORDO->GetVertice(0)->GetX()))
        / (2*3.14159)<<endl;
    cout<<ENERGIA(BORDO, cos)<<endl;
    double Centro [2] = {0,0};

    TApplication myapp(" app",NULL,NULL);

    TCanvas * drop_ROOT = new TCanvas(" MyDrop", " MyDrop",
        700,700);
    gPad->Range(-0.6, -0.2, 0.6, 1);

    double * x = new double [BORDO->GetNumeroVertici()+1];
    double * y = new double [BORDO->GetNumeroVertici()+1];

    for(int i=0; i< BORDO->GetNumeroVertici(); i++) { x[i]
        = BORDO->GetVertice(i)->GetX()+Centro [0]; y[i] =
        BORDO->GetVertice(i)->GetY()+Centro [1];}

```

```

x[BORDO->GetNumeroVertici()]=x[0];
y[BORDO->GetNumeroVertici()]=y[0];

TPolyLine * profilo = new TPolyLine(BORDO->
    GetNumeroVertici()+1,x,y);
profilo -> Draw();
myapp.Run(true);
drop_ROOT->Close();

return 0;
}

```

A.5 Programma per la realizzazione dei grafici

```

#include<iostream>
#include<fstream>
#include<cmath>
#include<time.h>
#include <TCanvas.h>
#include <TGraphErrors.h>
#include <TApplication.h>
#include <TF1.h>
#include<TAxis.h>
#include<TStyle.h>
#include<TLatex.h>

using namespace std;
TApplication * myapp = new TApplication ("app", NULL, NULL);

double * media_pesata (double * valori , double * errori , int n)
{
    double pesi[n];
    double * risultato=new double[2];
    risultato[0]=0;
    risultato[1]=0;
}

```

```

    for (int i=0;i<n;i++){
        pesi[i]=1./pow( errori[i],2);
    }

    for (int i=0;i<n;i++){
        risultato[0]=risultato[0]+valori[i]*pesi[i];
        risultato[1]=risultato[1]+ pesi[i];
    }
    risultato[0]=risultato[0]/risultato[1];
    risultato[1]=sqrt(1./(risultato[1]));
    return risultato;
}

double media(double * valori , int n){
    double risultato=0;
    for (int i=0;i<n;i++){
        risultato = risultato + valori[i];
    }
    return double(risultato/n);
}

double deviazione_standard (double * valori , int n){
    double risultato=0;
    double m = media(valori , n);
    for (int i=0;i<n;i++){
        risultato = risultato + pow((valori[i]-m),2);
    }
    return sqrt(risultato/(n-1.));
}

int main(){

    char tipo;
    int NumeroElementi=0;
    int NumeroElementi1=0;

    double * y = new double[40];
    double * yError = new double[40];

    double * x = new double[40];
    double * xError = new double[40];

```

```

double * y1 = new double[40];
double * y1Error = new double[40];

double * x1 = new double[40];
double * x1Error = new double[40];
ifstream in ("DatiPerGrafico", ios::binary);

if (!in){
    cout<<"ERROR : The file doesn't exist or it's
        impossible to open"<<endl;
    return 0;
}

in.ignore(5000, '\n');

for(int i=0; in>>tipo; i++){
    if(tipo == 'p'){
        in>>x[NumeroElementi]>>xError[
            NumeroElementi]>>y[NumeroElementi]>>
            yError[NumeroElementi];
        NumeroElementi++;}
    if(tipo == 'n'){
        in>>x1[NumeroElementi1]>>x1Error[
            NumeroElementi1]>>y1[NumeroElementi1
            ]>>y1Error[NumeroElementi1];
        NumeroElementi1++;}
}

double * h = new double[2];
double * he = new double[2];

h[0] = media(y, NumeroElementi);
h[1] = media(y1, NumeroElementi1);
he[0] = deviazione_standard(y, NumeroElementi);
he[1] = deviazione_standard(y1, NumeroElementi1);

cout<<"media+devizione standard dei valori P "<<media(
    y, NumeroElementi)<<"+"<<deviazione_standard(y,
    NumeroElementi)<<endl;

```



```

cout<<"media+deviazione standard dei valori N "<<media(
    y1, NumeroElementi1)<<"+"<<deviazione_standard(y1,
    NumeroElementi1)<<endl;
double * hh = media_pesata(h,he,2);
cout<<"media pesata totale"<<hh[0]<<"+"<<hh[1]<<endl;
gStyle->SetOptLogx(1);

TCanvas * Tela = new TCanvas(" Grafico", " Grafico");

TGraphErrors * grafico1 = new TGraphErrors(
    NumeroElementi1, x1, y1, x1Error, y1Error);
TGraphErrors * grafico = new TGraphErrors (
    NumeroElementi, x, y, xError, yError );
//TF1 * funz = new TF1(" funzione", funzione, 0,2,4);

//grafico->Fit(" funzione");

grafico->SetTitle(" ");
grafico1->SetTitle(" ");

//grafico1->SetMinimum(0.4);
//grafico->SetMinimum(0.4);
grafico->GetXaxis()->SetLimits(pow(10,13), pow(10,20));

grafico->GetXaxis()->SetTitle(" Concentrazione [atomi/cm
    ^{3}]");
grafico->GetYaxis()->SetTitle(" Isteresi [#circ]"); //
    Componente polare #gamma_{s}^{p} [dyne/cm]
grafico->SetMarkerStyle(21);
grafico->SetMarkerColor(2);
grafico1->SetMarkerStyle(21);
grafico1->SetMarkerColor(4);

grafico->Draw("AP");
grafico1->Draw("SAMEP");

Tela->Update();
myapp->Run(true);
Tela->Close();
return 0;

```

}

Bibliografia

- [1] <http://www.cleanroom.byu.edu/ResistivityCal.phtml>.
- [2] <http://macro.lsu.edu/HowTo/solvents/Surface%20Tension.htm>.
- [3] <http://www.fisher.co.uk/1/1/42578-isopropanol-99-8-electronic-use-residue-free.html>.
- [4] Cleaning procedures for silicon wafers.
<https://www.inrf.uci.edu/wordpress/wp-content/uploads/sop-wet-silicon-solvent-clean.pdf>.
- [5] Mechanical definition of surface tension.
<http://web.mit.edu/nmf/education/wettability/definition.html>.
- [6] *SPE/ANTEC 1997 Proceedings*. Annual Technical Conference Papers. Taylor & Francis, 1997.
- [7] Contact angle goniometry as a tool for surface tension measurements of solids, using zisman plot method. a physical chemistry experiment. *Journal of Chemical Education*, 77(1):63, 2000.
- [8] J.A. van den Berg Amir H. Al-Bayati, Kevin G. Orrman-Rossiter and D.G. Armour. Composition and structure of the native si oxide by high depth resolution medium energy ion scattering. *Surface Science*, (241):91–102, 1991.
- [9] R Brun, F Rademakers, and S Panacek. ROOT, an object oriented data analysis framework. 2000.
- [10] J.R Dann. Forces involved in the adhesive process i. critical surface tensions of polymeric solids as determined with polar liquids. *Journal of Colloid and Interface Science*, (32), 02 1970.
- [11] GBX digidrop. http://www.gbxonline.com/Home_Page.html.

- [12] A.W. Neumann D.Y. Kwok. Contact angle measurement and contact angle interpretation. *Advances in Colloid and Interface Science*.
- [13] Accu dyne Test. Surface tension components and molecular weight of selected liquids. http://www.accudynetest.com/surface_tension_table.html.
- [14] J.W. Gibbs, H.A. Bumstead, and R.G. Van Name. *Scientific Papers of J. Willard Gibbs ...: Thermodynamics*. Scientific Papers of J. Willard Gibbs. Longmans, Green and Company, 1906.
- [15] D. Graf, M. Grundner, R. Schulz, and L. Muhlhoff. Oxidation of hf treated si wafer surfaces in air. *Journal of Applied Physics*, 68(10):5155–5161, 1990.
- [16] Allan S. Hoffman.
<http://www.uweb.engr.washington.edu/education/pdf/ashsurfscontact%20angles05.pdf>.
- [17] Christian Hollauer. Modeling of thermal oxidation and stress effects. 1 2007.
- [18] Richard C. Jaeger. Introduction to microelectronic fabrication:volume 5 of modular serieson solid state devices. 11 2001.
- [19] Werner Kern. The evolution of silicon wafer cleaning technology. *Journal of The Electrochemical Society*, 137(6):1887–1892, 1990.
- [20] D. M. Koenhen and C. A. Smolders. The determination of solubility parameters of solvents and polymers by means of correlations with other physical quantities. *Journal of Applied Polymer Science*, 19(4):1163–1179, 1975.
- [21] Viorica Stanescu Manuela Florea-Spiroiu, Mihaela Olteanu and G. Nechifor. Surface tension components of plasma treated polysulphone membranes. <http://gw-chimie.math.unibuc.ro/anunivch/2008-2/AUBCh2008XVII21318.pdf>.
- [22] G. Mende, J. Finster, D. Flamm, and D. Schulze. Oxidation of etched silicon in air at room temperature; measurements with ultrasoft x-ray photoelectron spectroscopy (esca) and neutron activation analysis. *Surface Science*, 128(1):169 – 175, 1983.
- [23] M. Morita, T. Ohmi, E. Hasegawa, M. Kawakami, and M. Ohwada. Growth of native oxide on a silicon surface. *Journal of Applied Physics*, 68(3), 1990.
- [24] D. K. Owens and R. C. Wendt. Estimation of the surface free energy of polymers. *Journal of Applied Polymer Science*, 13(8):1741–1747, 1969.
- [25] Taylor John R. Introduzione all’analisi degli errori. lo studio delle incertezze nelle misure fisiche. 1999.

- [26] Christopher Rulison.
<https://cmi.epfl.ch/metrology/files/Kruss-DSA30/kruss-tn306-en.pdf>.
- [27] Anthony M Schwartz. Contact angle hysteresis: A molecular interpretation. *Journal of Colloid and Interface Science*, 1980.
- [28] C.M.S Vicente, P.S. Andr  , and R.A.S. Ferreira. Simple measurement of surface free energy using a web cam. *Revista Brasileira de Ensino de Fisica*, 34:1 – 5, 09 2012.
- [29] Thomas Young. An essay on the cohesion of fluids. *Philosophical Transactions of the Royal Society of London*, 95:65–87, 1805.
- [30] Yuehua Yuan and T.Randall Lee. Contact angle and wetting properties. In Gianangelo Bracco and Bodil Holst, editors, *Surface Science Techniques*, volume 51 of *Springer Series in Surface Sciences*, pages 3–34. Springer Berlin Heidelberg, 2013.
- [31] Xiaoge Gregory Zhang. *Electrochemistry of Silicon and Its Oxide*. 2001.

Ringraziamenti

Ringrazio i miei genitori Manlio e Mirna e mia sorella Irene per avermi sostenuto in questi anni di università e in modo particolare durante la stesura di questa tesi.

Ringrazio i miei nonni per aver sempre creduto nelle mie possibilità

Ringrazio i miei amici per i bei momenti passati insieme durante questi tre anni di università

Ringrazio Serena per essermi sempre stata accanto.

Ringrazio la mia correlatrice Beatrice Fraboni per la disponibilità dimostrata.

Ringrazio tutti i ricercatori e dottorandi del CNR di Bologna per avermi aiutato ogni volta che ho avuto bisogno, in particolar modo ringrazio: Denis Gentili, Francesco Valle, Luca Bartolini, Marianna Barbalinardo, Francesca Leonardi, Stefano Chiodini, Adrica Kyndiah, Marco Brucale e Marco Alfonso Salvatore.

Un ringraziamento particolare va soprattutto al mio relatore, il professor Cristiano Albonetti. Grazie per la simpatia, la disponibilità, la competenza e per essere stato sempre presente con i consigli e l'incoraggiamento che mi servivano.