

# Volare in alto

*Matematica applicata al motore  
di un razzo modellino.*



# Volare in alto

Scaletta:

- Cos'era il cansat e nostra esperienza
- Il motore del razzo; com'è fatto
- Il motore del razzo, come funziona
- Il nostro caso
- Cos'è un integrale e funzione integrale
- Intehrazione in informatica
- Codice dei due metodi di integrazione

## CanSat

Il CanSat è una competizione tra scuole che si tiene ogni anno il cui scopo è creare un “satellite” delle dimensioni di una lattina.

Questo satellite viene portato a 1000 m di quota da un razzo vettore che all'apogeo lo rilascia. La discesa avviene a velocità controllata grazie ad un paracadute e nel mentre i sensori registrano temperatura e pressione(da cui ricavare l'altitudine), oltre ad altri dati differenti per ogni esperimento. Alla fine della competizione viene premiato il progetto più originale e ben svolto.

Con il Sabin partecipai sia nel 2014 che nel 2015. Il primo anno il razzo prestatoci da Acme Italia andò a fuoco e si sciolse, perciò non potemmo registrare alcun dato.

Vista l'esperienza negativa del primo tentativo, l'anno dopo decidemmo di auto costruirci anche il razzo che volò benissimo regalandoci tantissima soddisfazione.

L'unico componente che non è possibile prodursi è il motore, per via del tipo di carburante che utilizza e per la difficoltà ingegneristica per la sua progettazione e realizzazione.

Tralasciando il CanSat la tesina si concentrerà



sul motore, concludendo così il lungo percorso di questo progetto scolastico molto interessante e utile per la mia scelta universitaria.

## Il motore

Nel modellismo aerospaziale vengono utilizzati maggiormente motori a combustibile solido contenuto in un cilindro con uno sfoco su di una base modellata per ottimizzare la spinta data dalla combustione.

Nel nostro caso avevamo a disposizione il **Pro38 512I285-15A** della **Cesaroni tecnology**; un motore composito che utilizza un combustibile derivato da quello usato nei vettori spaziali, costituito essenzialmente da gomma con alcuni additivi. Questi combustibili sono estremamente sicuri in quanto particolarmente inerti.

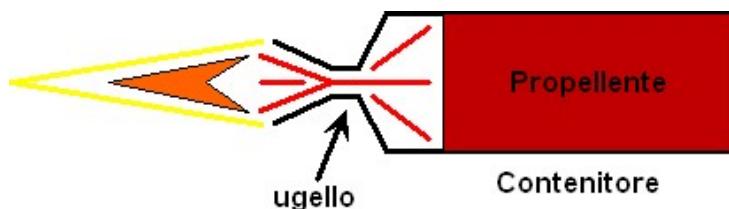
Dalla sigla è possibile ricavare moltissime specifiche:

- **38** mm di diametro;
- **512** Ns di Impulso totale;
- **285** N di spinta media;
- **15** s di delay prima della carica esplosiva che apre il razzo.



Il motore ad endoreazione, funziona sfruttando la terza legge di Newton: espellendo gas da una estremità si ottiene una spinta in senso opposto, di intensità proporzionale alla velocità ed alla massa di gas espulsi. I gas ovviamente sono prodotti dalla combustione rapida del combustibile solido contenuto nella camera cilindrica del motore.

Per ottenere una spinta utile i gas devono raggiungere velocità supersoniche passando attraverso una strozzatura (ugello) di forma opportuna e variabile a seconda delle caratteristiche del combustibile.



I motori utilizzati per il modellismo sono al 99% modelli commerciali per via dell'alta difficoltà nell'ingegnerizzarne la forma adeguata dell'ugello a seconda del carburante (che se prodotto artigianalmente ha quasi sempre caratteristiche differenti ad ogni prova).

## Parametri

Così come gli altri generi di motore, per definirne le caratteristiche e la potenza si utilizza una serie di parametri caratteristici, quelli riportati anche nella sigla del modello:

### Pro38 512I285-15A

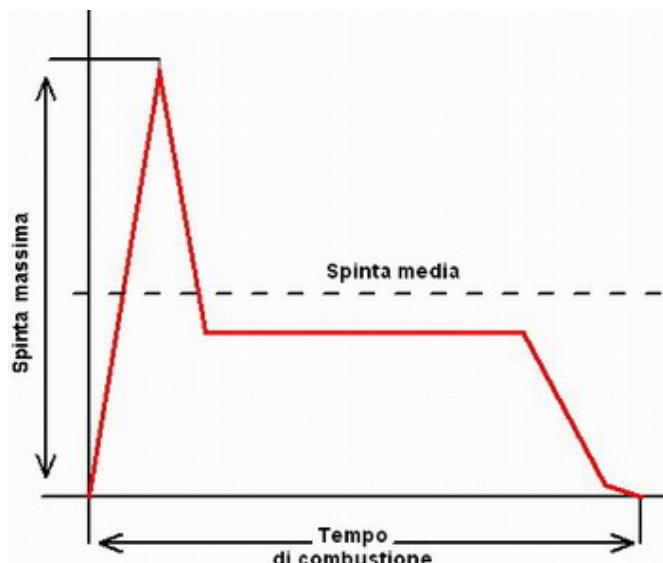
#### Motor Data

<b>Brandname</b>	Pro38 512I285-15A	<b>Manufacturer</b>	Cesaroni Technology
<b>Man. Designation</b>	512I285-15A	<b>CAR Designation</b>	512-I285-15A
<b>Test Date</b>	9/8/2001		
<b>Single-Use/ Reload/Hybrid</b>	Reloadable	<b>Motor Dimensions mm</b>	38.00 x 303.00 mm (1.50 x 11.93 in)
<b>Loaded Weight</b>		<b>Total Impulse</b>	510.1 Ns (114.7 lb.s)
<b>Propellant Weight</b>	272.4 g (9.61 oz)	<b>Maximum Thrust</b>	350.6 N (78.8 lb)
<b>Burnout Weight</b>		<b>Avg Thrust</b>	267.0 N (60.0 lb)
<b>Delays Tested</b>	"15,12,10,8,6"	<b>ISP</b>	191 s
<b>Samples per second</b>		<b>Burntime</b>	1.91 s
<b>Notes</b>	Classic™		

**Spinta media (average thrust)** - fornisce l'idea più immediata delle capacità di un motore, anche se deve essere valutato con molta attenzione. La spinta media viene misurata in Newton ed è il valore medio di spinta del motore lungo tutta la durata della sua combustione. Ciò è calcolabile tramite la media integrale della funzione che descrive la spinta.

Tipicamente la spinta raggiunge un picco iniziale che nel nostro caso è circa **350 N**, scende ad un valore medio che resta più o meno costante fino quasi al termine della combustione e scende ancora più o meno gradualmente poco prima di cessare.

**Tempo di combustione (burn time)** - è la durata della combustione del motore. Il tempo di combustione si ricava dalla curva di spinta fornita nelle istruzioni di ogni motore, che per noi corrisponde a **2 s**



**Impulso totale (Total impulse)** - L'impulso totale indica la "quantità di energia" che un dato motore è in grado di sviluppare. E' il prodotto della spinta media per il tempo di combustione:

$$\mathbf{I}_{\text{tot}} = \mathbf{S}_m * t$$

Più precisamente l'**impulso** è una grandezza vettoriale, misurata in Newton per secondo, definita in meccanica classica come l'integrale di una forza nel tempo.

$$\Delta \bar{p} = \int_{t_0}^{t_1} \bar{F} dt$$

Nel caso particolare di una forza costante avremo  $\Delta \bar{p} = \bar{F} \Delta t$ .

Il Pro38 512I285-15A possiede un Impulso totale di **510.1 Ns**

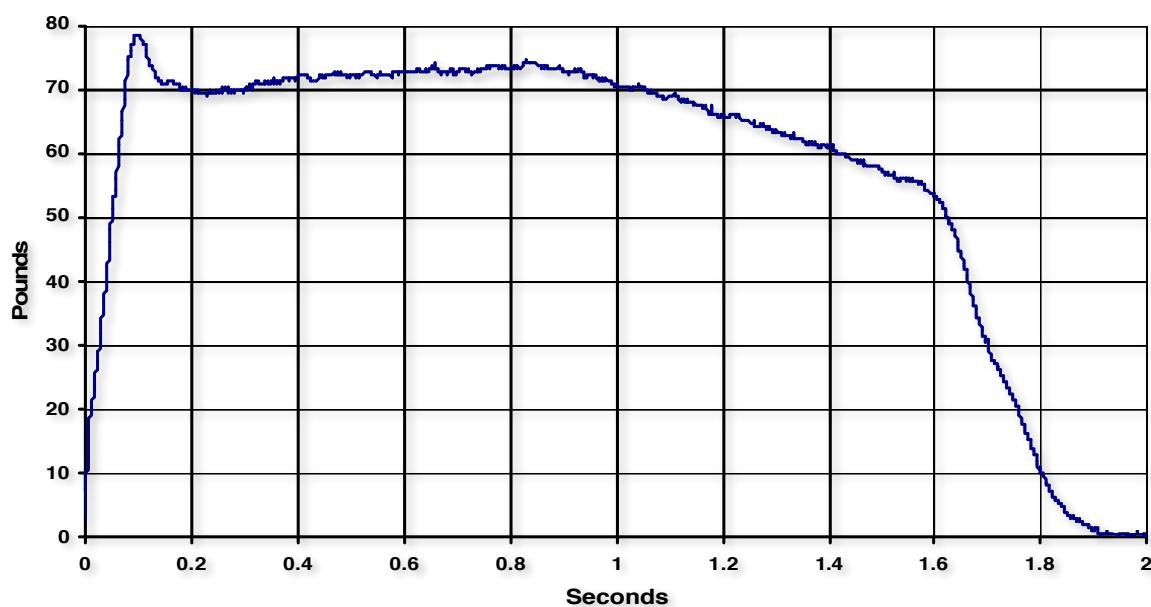
E' un parametro di grande importanza perché indica la classe di potenza del motore, un po' come fosse la "cilindrata". La classificazione standard dei motori prevede infatti proprio una suddivisione in classi di impulso totale.

**Curva di spinta (Thrust curve)** - La curva di spinta, o curva spinta-tempo, descrive l'andamento della spinta durante il tempo di funzionamento del motore. E' un grafico che si ottiene dalle prove statiche e che riporta la spinta in Newton o libbre sull'asse Y e il tempo in secondi sull'asse X.

Le curve di spinta si trovano nelle confezioni di ogni motore.

Nel caso del nostro motore la curva è la seguente, con le libbre piuttosto che i Newton.

**Pro38 512I285**



E' molto importante conoscere la curva di spinta di un motore perché fornisce una indicazione più precisa sul suo funzionamento, indica la **spinta massima**, e la **distribuzione della spinta nel tempo**.

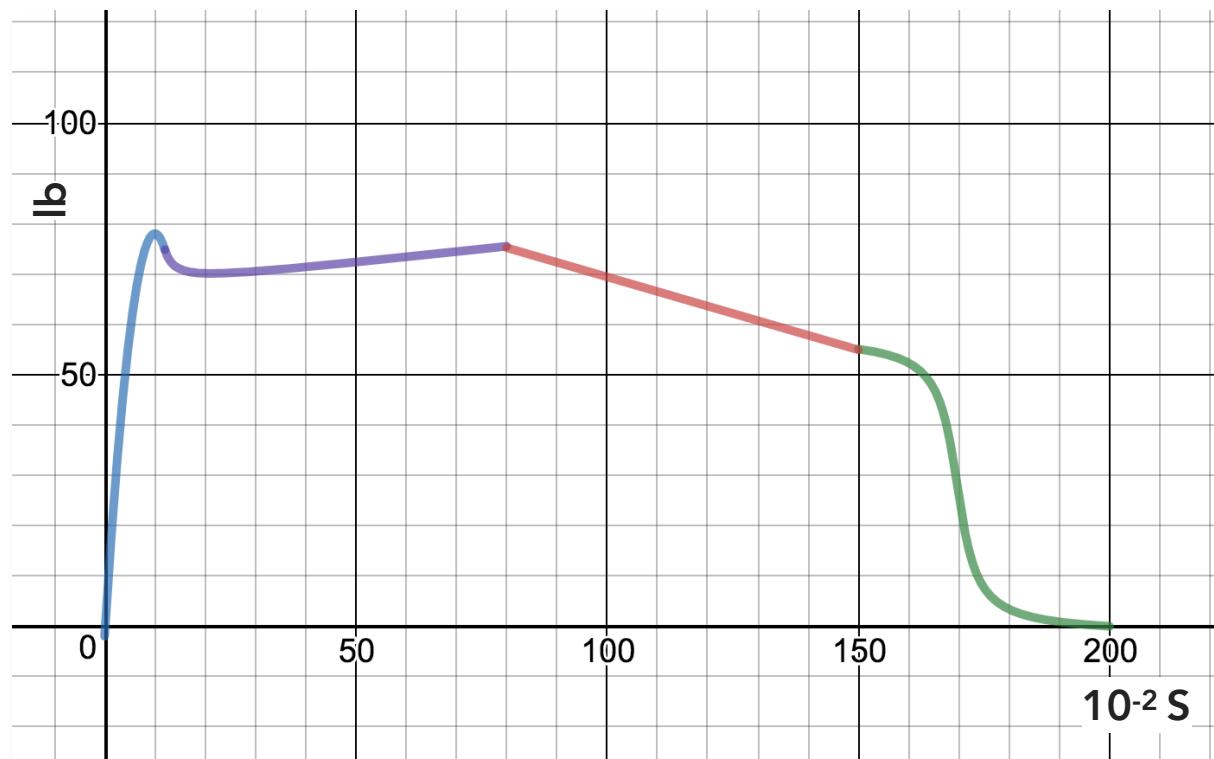
**Ritardo (delay)** - questo parametro non influenza la spinta. Il ritardo è espresso in secondi ed indica il tempo che intercorre tra il termine della combustione (quindi della spinta) e l'espulsione del paracadute. E' realizzato con una sostanza combustibile che brucia lentamente per il tempo indicato sul motore stesso e va scelto con attenzione in base alle caratteristiche (dimensioni, peso) del modello; nel caso del CanSat va calibrato ad ogni lancio per via del peso dei differenti razzi ed esperimenti. Bisogna calcolare il delay così da far esplodere la carica che pressurizza e fa aprire il razzo esattamente all'apogeo.

**Impulso specifico (Isp)** - E' un parametro che al combustibile. Indica la quantità di energia sviluppata per unità di massa. In sostanza indica quanti Newton-secondo vengono sviluppati da ogni kg di combustibile. Si misura in Newton-secondi/Kg, o in secondi.

Per noi l'Isp vale **191 s**.

Questi sono comunque i dati forniti dal produttore.

Per verificarli ho creato delle funzioni che andassero per quanto possibile a sovrapporsi al grafico originario usando l'applicativo web **Desmos**.



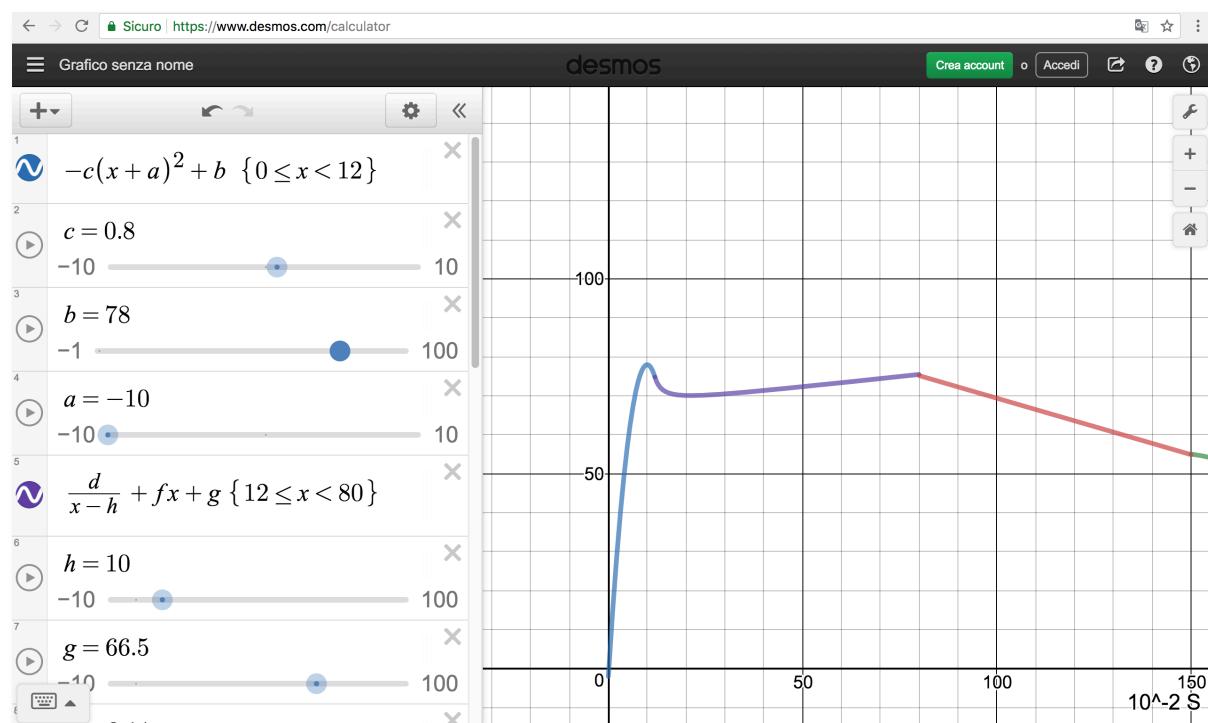
## Costruzione della funzione

Per costruire la funzione è necessario ovviamente conoscere il comportamento delle varie curve, e grazie agli slider di Desmos si è facilitati nella ricerca dei parametri più adatti per ricalcare fedelmente il grafico reale.

Ho scelto di replicare il grafico in libbre piuttosto che Newton sull'asse delle ordinate per poterlo confrontare al meglio col grafico reale. Per le ascisse vi sono centesimi di secondo.

La prima parte è descritta da una parabola con estremi 0 e 12 di equazione:

$$y = \frac{-4(x - 10)^2}{5} + 78 \quad \{0 \leq x < 12\}$$



Il secondo tratto è costruito a partire da un iperbole quadratica. La funzione finale è:

$$y = \frac{14}{x - 10} + \frac{11}{100}x + \frac{133}{2} \quad \{12 \leq x < 80\}.$$

L'  $\frac{11}{100}$  è il coefficiente angolare dell'asintoto obliqua dell'iperbole di equazione completa:

$$y = \frac{11}{100}x + \frac{133}{2}.$$

Di seguito vi è una semplice retta di equazione  $y = \frac{-29}{100}x + \frac{492}{5}$   $\{80 \leq x < 150\}$ .

Per finire un'arcotangente descrive l'andamento decrescente dell'ultima fase della spinta.

L'equazione risulta:  $y = -19 \arctan\left(\frac{10}{29}x - \frac{117}{2}\right) + 28$   $\{150 \leq x < 191\}$

Il coefficiente iniziale decreta l'ampiezza della curva;  $\frac{10}{29}$  influenza la derivata nel flesso e di conseguenza l'ampiezza in X della curva;  $-\frac{117}{2}$  ci trasla la funzione su X ed infine 28 è la coordinata in Y del flesso obliquo.

Di queste finzioni non ho verificato continuità e derivabilità in quanto devono descrivere approssimativamente un fenomeno fisico campionato ogni millesimo di secondo con tanto di errori strumentali e fluttuazioni, inoltre la non derivabilità è causata anche della composizione stessa del motore che in quanto composto da differenti stadi di carburante non offre una spinta continua, e lo si nota dal grafico fornito dal produttore.

Calcolando la funzione integrale  $F(x) = \int_0^x F(t)dt$  utilizzando le curve appena trovate si ottiene che in  $X=191$  l'area totale è circa **112.38 lbs (499.89 Ns)**, e che rappresenta la spinta totale.

Da ciò si capisce che il modello è abbastanza fedele al fenomeno fisico essendoci una discrepanza di appena **10.21 Ns**.

La **Average thrust** è calcolabile mediante la media integrale la cui formula corrisponde a:

$$f(c) = \frac{1}{b-a} \int_a^b f(x)dx$$

dove  $a = 0$ ,  $b = 191$  e  $f(x)$  rappresenta le nostre funzioni.

Dal momento che il risultato dell'integrale lo abbiamo già, per ottenere la media integrale ci basta calcolare  $f(c) = \frac{499.89 \text{Ns}}{1.91 \text{s}} = 261.72 \text{N}$ .

Il nostro valore si discosta di appena 6.28N da quello effettivo, confermando la validità del modello.

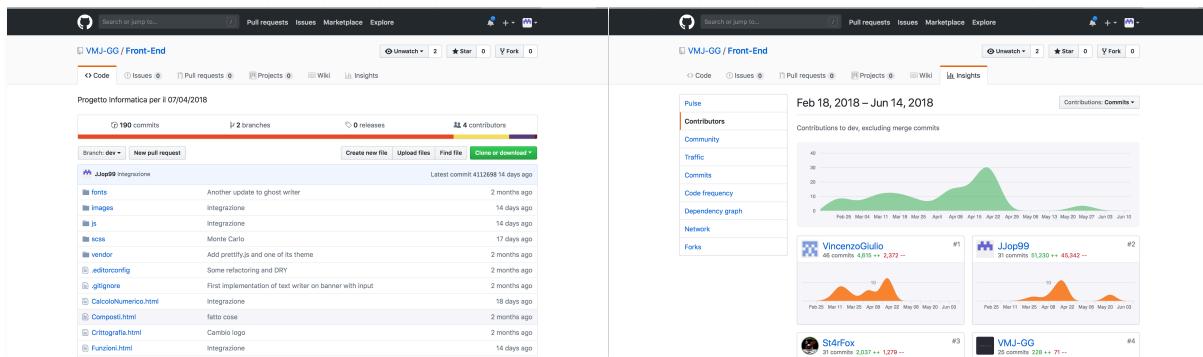
## Il sito

Per svolgere questi integrali è bastato fare affidamento al sito web in fase di sviluppo creato durante l'anno scolastico 2017/18 da Bosco Vincenzo, Favale Stefano, Guazzaloca Mattia e il sottoscritto Jop Jacopo.

Il sito, che prende il nome di Web-tips, è nato in seguito a un lavoro scolastico di cui ci siamo appassionati e che abbiamo deciso di portare avanti simulando un **team di sviluppo** e con la missione di caricare nel web un portale in cui gli studenti di scienze applicate potessero ritrovare gli argomenti di Informatica svolti a scuola con spiegazione e dimostrazione di ognuno, il tutto incorniciato da una grafica moderna e, per quanto possibile, accattivante, pensando molto all'esperienza utente.

Alla fine lo studente che naviga su questo sito si trova di fronte a tutti i contenuti fondamentali di ogni argomento, avendo la possibilità di ripassare agevolmente.

Per gestire il lavoro su diversi terminali è stato utilizzato **Git**, dopo aver creato un repository sul portale GitHub.



Inizialmente è stato necessari prendere confidenza col terminale e i comandi Git, cosa mai affrontata a scuola e che ci ha stimolato da subito ad imparare, soprattutto perché si è rivelato realmente uno strumento comodissimo per lavorare visto che consente di apportare modifiche contemporaneamente da più devices sullo stesso progetto, fondendo poi il lavoro di tutti in maniera intelligente. Inoltre ci ha permesso di monitorare la nostra attività.

Oltre a ciò è stato necessari approfondire notevolmente le nostre conoscenze di **HTML**, **Javascript**, **CSS**.

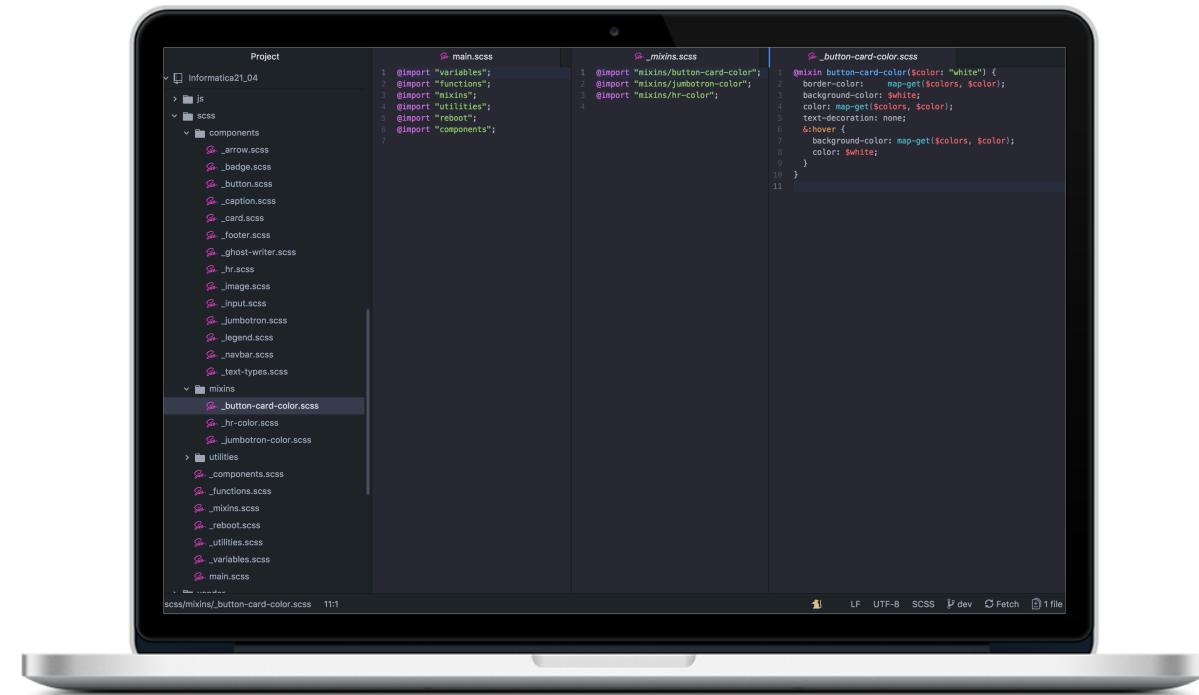
In seguito alla divisione dei compiti ho lavorato principalmente al Front-End(interfaccia) del sito assieme a Bosco. La grafica è basata sul toolkit **Bootstrap** che permette di sfruttare un gran numero di classi di stile già scritte, velocizzando così il lavoro.

Tuttavia è stato necessario creare molte nuove classi e per fare ciò abbiamo deciso di usare l'**SCSS** strutturato in **SASS**.

Tutto ciò con ore di ricerca e tentativi, in quanto strumenti mai utilizzati a scuola, per ottenere un risultato quanto più simile possibile.

**SASS** (Syntactically Awesome StyleSheets) è un'estensione del linguaggio **CSS** che permette di utilizzare variabili, di creare funzioni e di organizzare il foglio di stile in più file.

Il linguaggio **SASS** si basa sul concetto di preprocessore **CSS**, il quale serve a definire fogli di stile con una forma più semplice, completa e potente rispetto ai **CSS** e a generare file **CSS** ottimizzati, aggregando le strutture definite anche in modo complesso.



**SASS** va installato tramite **Ruby** e di ciò si è occupato Guazzaloca in quanto già più esperto in materia, inoltre ha creato uno script per consentire la compilazione del foglio di stile.

```

Project
└─ Informatica21_04
    └─ .DS_Store
    └─ Rakefile
        1 task :default => :watch
        2
        3 desc "Compile stylesheets"
        4 namespace :compile do
        5   task :expanded do
        6     sh "sass --style=expanded scss/main.scss:css/main.css"
        7   end
        8
        9   task :minified do
        10    sh "sass --style=compressed scss/main.scss:css/main.min.css"
        11  end
        12
        13   task :singleline do
        14     sh "sass --style=compact scss/main.scss:css/main.compact.css"
        15   end
        16 end
        17
        18 desc "Watch stylesheets"
        19 task :watch do
        20   sh "sass --watch --style=expanded scss/main.scss:css/main.css"
        21 end
        22
        23 desc "Pushing changes to remote"
        24 namespace :push do
        25   task :all, [:message] do |t, args|
        26     sh "git add -A && git commit -am '#{args.message}' && git push"
        27   end
        28
        29   task :files, [:message] do |t, args|
        30     files = args.extras
        31     files.each do |fn|
        32       sh "git add #{fn}"
        33     end
        34     sh "git commit -am '#{args.message}' && git push"
        35   end
        36 end
        37

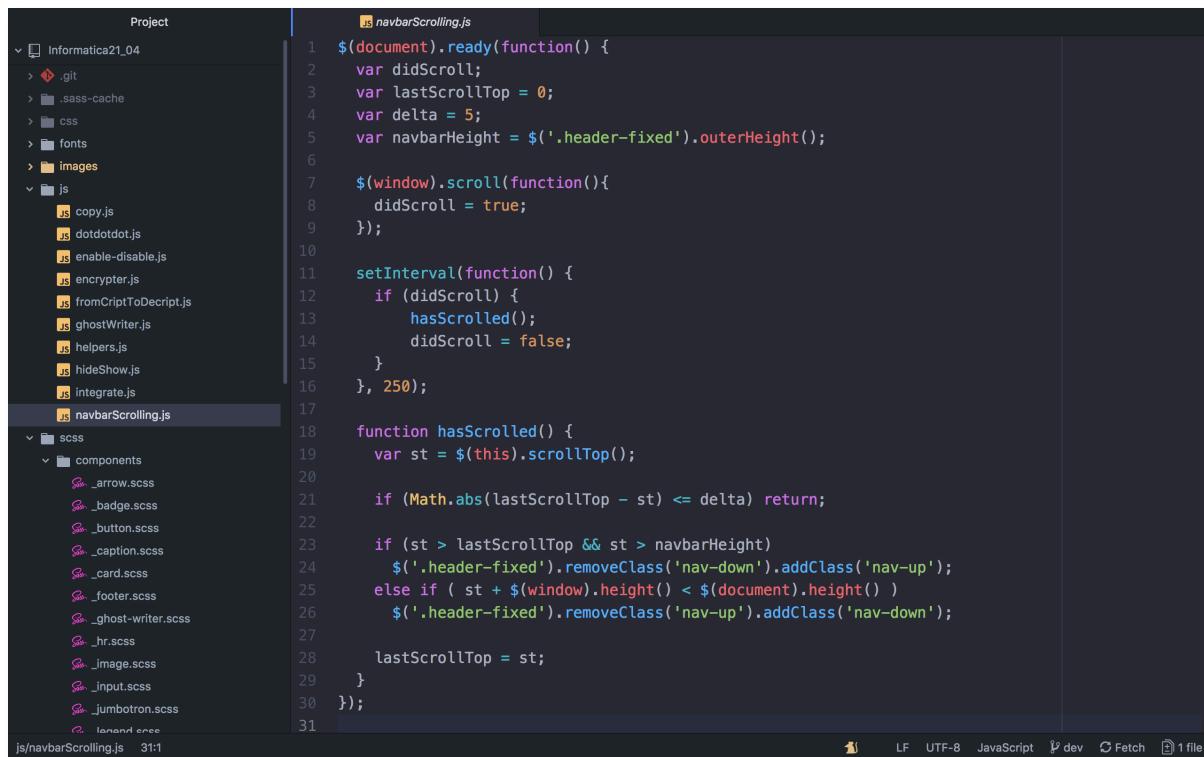
```

Al momento nel sito è presente solo l'area dedicati agli algoritmi svolti in quinta, quindi Crittografia e Calcolo Numerico.

Queste due categorie sono poi divise per genere e per metodo.

Per ogni metodo l'utente trova una sezione della pagina i cui testare il metodo subito all'inizio; scorrendo in basso vengono visualizzate la descrizione e la teoria legate ad esso; il codice **HTML** e **JS** per permettere di testare da sé il codice con un semplice copia e incolla; la storia delle origini e dell'evoluzione di tale metodo.

La parte che permette di provare direttamente i metodi e confrontarli è in **JS** e **jQuery**, così come gli script che permettono di animare alcuni elementi dell'interfaccia come la navbar. Anche di questi linguaggi sono state imparate tante nuove funzioni.

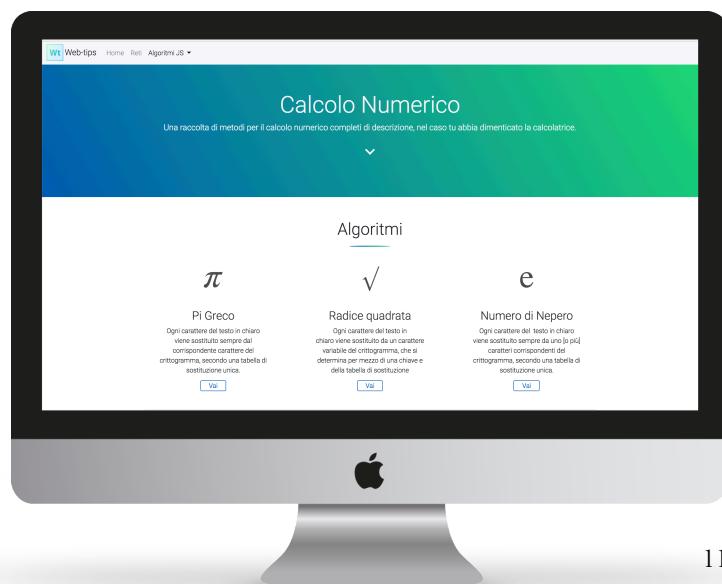


```

Project
└─ Informatica21.04
    ├ .git
    ├ .sass-cache
    ├ css
    ├ fonts
    └─ js
        └─ navbarScrolling.js
            1 $(document).ready(function() {
            2     var didScroll;
            3     var lastScrollTop = 0;
            4     var delta = 5;
            5     var navbarHeight = $('.header-fixed').outerHeight();
            6
            7     $(window).scroll(function(){
            8         didScroll = true;
            9     });
            10
            11     setInterval(function() {
            12         if (didScroll) {
            13             hasScrolled();
            14             didScroll = false;
            15         }
            16     }, 250);
            17
            18     function hasScrolled() {
            19         var st = $(this).scrollTop();
            20
            21         if (Math.abs(lastScrollTop - st) <= delta) return;
            22
            23         if (st > lastScrollTop && st > navbarHeight)
            24             $('.header-fixed').removeClass('nav-down').addClass('nav-up');
            25         else if ( st + $(window).height() < $(document).height() )
            26             $('.header-fixed').removeClass('nav-up').addClass('nav-down');
            27
            28         lastScrollTop = st;
            29     }
            30 });
            31
js/navbarScrolling.js 31:1

```

Il fatto di usare **Bootstrap** agevola anche il porting su dispositivi mobili in quanto gestisce autonomamente e in modo dinamico sul layout del sito a seconda della dimensione dello schermo su cui viene visualizzato dall'utente.



Nel sito è presente la categoria Calcolo Numerico che comprende anche l'integrazione.

Qui si trovano al momento due metodi di integrazione: Trapezi e Rettangoli.

Il metodo dei Rettangoli è il più semplice dei metodi di integrazione numerica per l'approssimazione di un integrale definito della forma:  $\int_a^b f(x) dx$

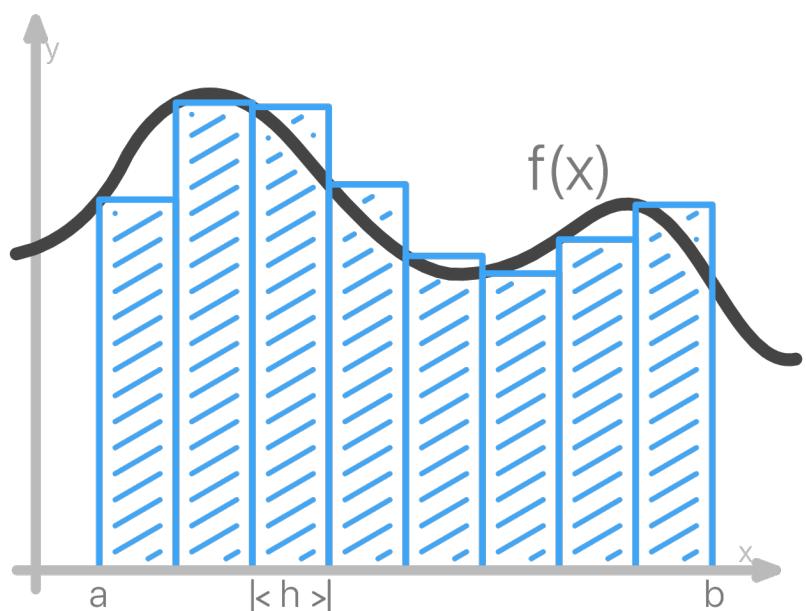
Questo metodo approssima l'integrazione (quindi l'area sottesa alla funzione) creando rettangoli di base  $h = \frac{b-a}{precision}$  e altezza  $f(x_n)$ .

Il valore di  $x_n$  lo si determina con  $x_0 = a$ ,  $x_{n+1} = x_n + h$

I rettangoli hanno quindi il vertice sinistro sulla funzione, impedendo così di rappresentare fedelmente l'integrale.

Per minimizzare l'errore si dovrà aumentare il numero di rettangoli riducendo la dimensione di  $h$  per imitare il ruolo del differenziale  $dx$ .

In informatica  $h$  non può tendere a 0, ma può assumere valori abbastanza bassi da causare un errore trascurabile.



Lo script del metodo dei Rettangoli è il seguente:

```

1.      <script>
2.          function f(x) {
3.              return eval(fn);
4.          }
5.
6.          function Rettangoli() {
7.              a = parseInt(document.f.a.value);
8.              b = parseInt(document.f.b.value);
9.              fn = document.f.fn.value;
10.             iters= parseInt(document.f.k.value);
11.
12.             h = (b - a) / iters;
13.             x = a;
14.             result = 0;
15.
16.             for(i = 0; i < iters; i++)
17.             {
18.                 result += f(x) * h;
19.                 x += h;
20.             }
21.             document.f.r.value = result;
22.         }
23.     </script>
```

Il metodo dei Trapezi è una evoluzione del metodo dei Rettangoli ed è uno dei metodi di integrazione numerica più preciso per l'approssimazione di un integrale definito.

Questo metodo è pressoché identico al quello dei rettangoli, ma approssima l'integrazione (quindi l'area sottesa alla funzione) creando trapezi rettangoli di altezza  $h = \frac{b-a}{precision}$  e due basi  $f(x_n)$  ed  $f(x_{n+1})$ , piuttosto che i rettangoli visti prima.

Di questi trapezi si ottiene l'area :  $A = \frac{(f(x_n) + f(x_{n+1}))}{2}h$ .

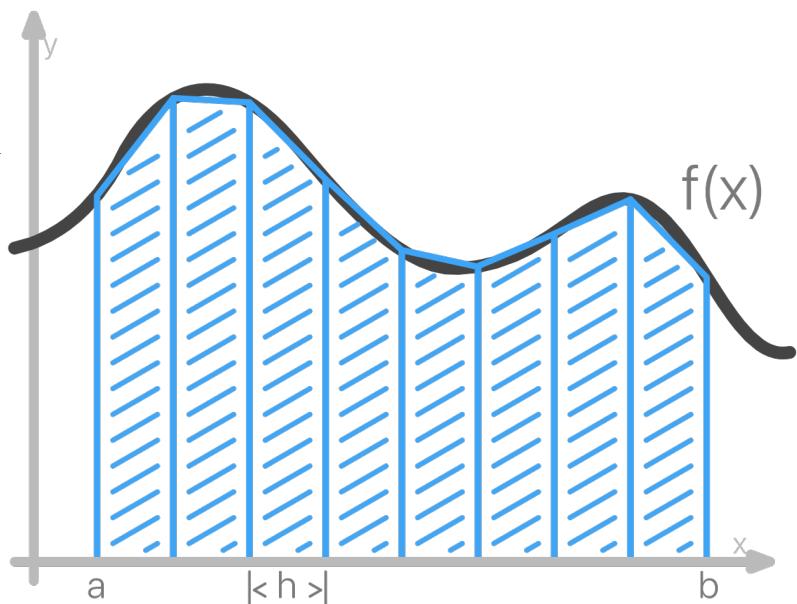
Anche qui il valore di  $x_n$  lo si

determina con  $x_0 = a$ ,

$x_{n+1} = x_n + h$

I trapezi hanno quindi i vertici adiacenti al lato obliquo sulla funzione, rappresentando quasi fedelmente l'integrale.

Sul sito si nota la differenza di precisione tra i due metodi, ma anche il loro avvicinarsi aumentando il numero di iterazioni che dovrà svolgere lo script.



## Conclusione

Grazie a due progetti scolastici abbiamo potuto:

- vedere l'applicazione fisica dell'integrazione per via dell'Impulso concludendo così il percorso studio di ogni aspetto del CanSat;
- analizzato l'integrazione dal punto di vista informatico;
- ampliare notevolmente conoscenze di informatica, elettronica, problem-solving, team-working.
- Comprendere cosa cercare di studiare nell'immediato futuro.