



Università  
Ca' Foscari  
Venezia

Università degli Studi di Venezia  
Dipartimento di Scienza Ambientali, Informatiche e  
Statistiche

---

Corso di Laurea in Informatica

Tesi di laurea

**Titolo**

Sottotitolo

Candidato:  
Matteo Scarpa  
Matricola 845087

Relatore:  
Claudio Silvestri

Anno Accademico 2015–2016

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Dedicata a tutti quelli che mi sono stati vicini nelle giornate no.



# Indice

<b>1</b>	<b>Descrizione dell'argomento</b>	<b>3</b>
1.1	Problema da risolvere . . . . .	3
1.1.1	R o erosività delle precipitazioni . . . . .	4
1.1.2	K o erodibilità del suolo . . . . .	4
1.1.3	LS rapporto tra lunghezza e pendenza . . . . .	5
1.2	Possibili risoluzioni . . . . .	5
1.2.1	Software di graphic design . . . . .	5
1.2.2	Strumenti di visualizzazione 3D . . . . .	6
1.2.3	Programmazione . . . . .	6
1.2.4	Desktop QGis . . . . .	6
1.3	QGis . . . . .	7
1.3.1	Descrizione dei tipi di dati supportati . . . . .	7
1.3.2	La struttura di QGis . . . . .	8
1.3.3	Sistema dei plugin . . . . .	9
<b>2</b>	<b>Realizzazione</b>	<b>11</b>
2.1	Divisione in sottoproblemi . . . . .	11
2.1.1	Interfaccia grafica . . . . .	12
2.1.2	Leggere e scrivere i dati . . . . .	13
2.1.3	Elaborazione dati . . . . .	15
	<b>Bibliografia</b>	<b>17</b>

Elenco delle figure

Elenco delle tabelle

# Introduzione

In questa tesi viene spiegato come è stato realizzato un plugin per QGis per il calcolo del terreno eroso avendo dei dati geomorfici legati al territorio scelto. Questi dati comprendono la conformazione del terreno, le precipitazioni registrate nell'area, il tipo di copertura della vegetazione e il numero di anni per cui questa previsione viene eseguita.

Questo plugin è stato scritto in Python e implementa il plugin ipotizzato nella tesi [Zen13] che risulta essere indipendente e utilizzabile per il calcolo su una qualunque area geografica di cui si hanno i dati richiesti. In oltre il sistema risulta indipendente dalle dimensioni dell'area geografica selezionata. E' infatti possibile inserire un'area geografica di una qualsiasi dimensione per procedere con l'elaborazione. Da questo punto di vista il programma può procedere con aree geografiche grandi quanto continenti ma non è detto che il modello matematico si comporti correttamente con aree sufficientemente grandi o piccole rispetto alla precisione dei dati acquisiti.





# Capitolo 1

## Descrizione dell'argomento

### 1.1 Problema da risolvere

Lo scopo di questa tesi è quello di calcolare e visualizzare in modo corretto l'erosione del territorio nell'area geografica scelta. Come già accennato nell'introduzione, si può scegliere qualunque area geografica superiore a un metro quadro<sup>1</sup>.

Il modello viene definito col nome RUSLE

Il problema può essere riassunto nell'equazione

$$A = R * K * LS * C * P \quad (1.1)$$

in cui

- **A** corrispondente alla perdita di suolo annua
- **R** erosività delle precipitazioni
- **K** erodibilità del suolo
- **LS** rapporto tra la lunghezza e la pendenza
- **C** fattore di copertura del suolo
- **P** i parametri delle misure di prevenzione dell'erosione espressi sotto forma di matrice

---

<sup>1</sup>Dimensione minima sotto la quale i file formato raster non possono andare

Tutti questi dati sono forniti sotto forma di file che vengono poi elaborati per ottenere il raster corrispondente alla perdita di suolo annuo. Fa eccezione il parametro P, che può venire omissso e rimosso dall'equazione, visto che in caso non ci siano misure di prevenzione dell'erosione il valore corrispondente al punto della matrice corrisponde a 1.

### 1.1.1 R o erosività delle precipitazioni

Il parametro R rappresenta quanto le precipitazioni in una determinata area geografica. Viene fatto attraverso dati pluviometrici. Questi ultimi non sono dati con risoluzione temporale pari a un singolo evento meteorico ma da equazioni semplificate per ottenere il parametro da i dati di precipitazioni annue.

$$R = \sum_{i=1}^n \frac{p_i}{n} \quad (1.2)$$

In questa equazione

- $n$  corrisponde al numero di anni in cui si ha lo storico pluviometrico
- $p_i$  corrisponde alle precipitazioni dell' $i$ -esimo anno.

Questo permette una approssimazione utile per l'applicazione del modello RUSLE.

### 1.1.2 K o erodibilità del suolo

Il parametro K rappresenta quanto il terreno può essere eroso. Questo dipende molto da come strutturato il terreno, dalla sua composizione geologica e da alcune caratteristiche fisiche e chimiche dello stesso.

Questo tipo di dato non è disponibile per aree geografiche di grosse dimensioni per cui è stata fatta una regressione matematica su dati sulla tessitura del terreno dell'area interessata. Questo ha permesso la creazione di un modello matematico descritto dall'equazione di Romkens che permette la creazione di una stima precisa (per quanto riguarda i nostri scopi) dell'erodibilità del suolo.

$$K = 0,0034 + 0,0405 * \exp -0,5 \left( \frac{(\log(D_g) + 1,659)^2}{0,7101} \right) \quad (1.3)$$

In cui  $D_g$  è il diametro medio pesato rispetto alla classi tessiturali dominanti in mm

### 1.1.3 LS rapporto tra lunghezza e pendenza

LS rappresenta un rapporto tra lunghezza e pendenza a meno di due parametri  $m = 0, 4$  e  $n = 1, 3$  definiti per convenzione.

$$LS = \left( \frac{A}{22,13} \right)^m \left( \frac{\sin(\alpha)}{0,0896} \right)^n \quad (1.4)$$

Che viene modificata nella 1.5 per ottenere una equazione che meglio rappresenti superfici con elevata complessità topografica e/o alta variabilità di pendenze.

$$LS = \left( \frac{flowacc * cellsize}{22,13} \right)^{0,4} \left( -1,5 + \frac{17}{1 + e^{2,3-6,1*\sin \alpha}} \right) \quad (1.5)$$

In cui

- *flowacc* corrisponde all' UCA o numero di pixel tramite i quali l'acqua defluisce nell'area precisa di pixel
- *cellsize* dimensione dei pixel
- $\alpha$  pendenza dell'area

### 1.1.4 C o fattore di copertura del suolo

Il fattore C è un rapporto tra la perdita di suolo in precise condizioni di copertura e la perdita di suolo nel caso il terreno sia privo di qualunque protezione. Attualmente sono state fornite quarantaquattro classi di copertura di terreno con altrettanti valori del fattore C.

E' possibile anche utilizzare i dati provenienti da sensori di tipo AVHRR<sup>2</sup> con cui si calcola l'indice NDVI<sup>3</sup> che varia tra 1 e -1

## 1.2 Possibili risoluzioni

La tesi [Zen13] propone quattro soluzioni per il problema dell'erosione annua del terreno. Sono state tutte prese in esame prima di iniziare lo sviluppo del plugin e qui riportiamo vantaggi, svantaggi e problemi legati a queste soluzioni.

<sup>2</sup>Advance Very High Resolution Radiometer

<sup>3</sup>Normalized Difference Vegetation Index

### 1.2.1 Software di grafic design

Attraverso i classici programmi di creazione di ambienti virtuali è possibile visualizzare le modifiche dell'ambiente in base ai dati raccolti. Questo permette di creare una rappresentazione realistica e molto dettagliata dell'ambiente preso in esame ma richiede una grossa quantità di tempo e notevoli sforzi, senza contare che necessita di un esperto di modellazione 3D che si occupi del lavoro. In oltre questo metodo non permette l'automatizzazione ma necessita comunque di una componente umana che realizzi il modello 3D.

Per questo motivo non è stato preso in esame questa soluzione. Infatti la possibilità di calcolare e visualizzare l'erosione del terreno deve essere facilmente automatizzabile e i risultati forniti in tempi brevi.

### 1.2.2 Strumenti di visualizzazione 3D

E' possibile utilizzare strumenti di visualizzazione di modelli 3D specifici per la visualizzazione di ambienti reali. Questo permette la visualizzazione dell'area presa in esame confrontandola anche con il terreno circostante ma prevede che sia già stato generato un file nel formato leggibile dal programma scelto.

Questo ripropone i problemi dati dalla scelta di un software di grafic design in quanto prevede che ci sia un software o un "tecnico" che generi il file nel formato richiesto rendendo l'applicativo non automatizzabile.

### 1.2.3 Programmazione

Attraverso alcuni linguaggi di programmazione è possibile calcolare il raster di output. Per esempio è possibile farlo con Mathlab, R o Python con tempi di esecuzione diverso in base al linguaggio e a quanto è stato ottimizzato il codice che esegue queste operazioni.

Questo risolve il problema dell'automatizzazione del processo, permettendo così di creare un servizio di elaborazione dei dati autonomo e indipendente dall'essere umano, ma non realizza l'altro punto importante del progetto: non permette di visualizzare i dati in input ne in output.

In oltre richiede un notevole lavoro in quanto questa soluzione necessita di scrivere un programma per intero, gestire le GUI e tutta la gestione dei formati di file solo per avere l'espressione [1.1](#) eseguita.

### 1.2.4 Desktop QGis

Software che permette di visualizzare e elaborare raster e altri formati di file geomorfici. Molto usato e supporta, nativamente o attraverso plugin installabili, la quasi totalità dei formati di file esistenti. In oltre permette, attraverso codice python, c e c++, di estendere le funzionalità del programma stesso.

Il vantaggio fondamentale di questo sistema è l'estremo supporto di formati diversi e la presenza di un motore grafico che permette la visualizzazione corretta dei dati, con supporto con i formati di file per mappe e con i servizi online come google maps. La sua installazione, in oltre, contiene alcune delle librerie più complete per quanto riguarda le operazioni sui raster e array.

Invece comporta anche dei difetti. Limita infatti sia i linguaggi<sup>4</sup> in cui è possibile sviluppare un plugin o estensione e ha solo alcune delle maggiori librerie disponibili.

## 1.3 QGis

QGis si definisce come "Un Sistema di Informazione Geografica Libero e Open Source"[\[Sita\]](#) ovvero è un software che gestisce, elabora e visualizza dati geomorfici e georeferenziati, completamente gratuito e modificabile in ogni sua parte(è limitato dalla licenza utilizzata nello sviluppo del software [\[Sitb\]](#)).

In particolare sono state utilizzate le componenti di gestione input e output dei raster, il motore di calcolo e il sistema di rendering video. In oltre il programma supporta anche un sistema di plugin e scripting che aumenta le funzionalità del programma utilizzando codice esterno sviluppato dalla comunità.

Altro dettaglio molto utile di QGis è il fatto che un programma multi piattaforma. Questo vuol dire che il programma è stato distribuito per più sistemi operativi<sup>5</sup> e obbliga gli sviluppatori a scrivere plugin e script che siano anch'essi multiplatforma. Questo permette l'utilizzo dei plugin e degli script per la creazione di funzioni automatizzate utilizzando le componenti già presenti in QGis o aggiungerne di nuove.

In oltre, QGis supporta la quasi totalità dei formati open source utilizzati per la codifica per dati geomorfici e georeferenziati in modo nativo mentre i formati di file che non supporta sono utilizzabili attraverso plugin gratuiti disponibili online.

---

<sup>4</sup>E' possibile sviluppare solo per Python, C e C++ i plugin anche se consigliano di usare Python per una questione di velocità di sviluppo

<sup>5</sup>In questo caso per Android, Mac OS, Windows e Linux

### 1.3.1 Descrizione dei tipi di dati supportati

Per risolvere il problema presentato è necessario passare al programma i dati. Sono devono essere già presenti all'interno dell'elaboratore o possono essere letti da un flusso in ingresso proveniente da sensori o da altri elaboratori.

Per il nostro problema i dati vengono forniti sotto forma di file o di database in quanto possono necessitare di una pre elaborazione e una formattazione, e di conseguenza non è conveniente l'utilizzo di un flusso. In oltre è possibile che i dati non vengano da un unico sensore in quanto sono dati legati a composizione geologica e a piogge in determinate aree e, di conseguenza, che necessitano di multipli sensori.

I formati di dati possono essere divisi in tre macrocategorie in base a come vengono forniti:

- Vettoriali
- Raster
- DBRMS con estensione spaziale

Questi tre formati permettono solo alcune operazioni possibili legate alla composizione dei dati nel formato di input. Quindi, per alcune operazioni, QGis converte i dati nel formato appropriato in modo che sia sempre possibile fare tutte le operazioni disponibili, anche se queste può portare ad approssimazioni dei dati nella conversione.

### 1.3.2 La struttura di QGis

QGis è un programma avanzato con molte funzionalità che possono essere suddivise in alcune macro-categorie che sono indipendenti dal tipo di dato o file utilizzato. Le funzionalità standard di QGis sono gestite attraverso dei moduli che costituiscono l'installazione di base del programma e le funzionalità aggiuntive vengono implementate attraverso plugin dipendenti da questi moduli di base per poter funzionare correttamente. Solitamente i moduli sono scritti in C o C++ mentre i plugin e gli script sono in Python.

#### Rendering grafico dei dati

QGis ha la capacità di realizzare svariate visualizzazioni grafiche dei dati inseriti o calcolati. Questo permette la realizzazioni di immagini ad alta definizione dei dati inseriti e ne permetta anche l'esportazione nei formati più comuni di immagini.

## Realizzazione mappe

In presenza di dati georeferenziati QGIS permette di sovrapporre la rappresentazione grafica dei dati alla mappa corrispondente alle coordinate georeferenziali. Questo permette di creare mappe fisiche contenenti la visualizzazione di aree di erosione attraverso aree colorate o altri effetti applicabili alla mappa sottostante.

## Creazione, elaborazione e conversione dati

QGIS presenta due moduli per l'elaborazioni di dati:

- **Class** modulo standard di QGIS per i calcoli aritmetici elementari e alcuni calcoli statistici di base
- **Grass** modulo aggiuntivo di QGIS per i calcoli avanzati e regressioni. Contiene anche modelli previsionali e probabilistici.

## Analisi dati

Quando si lavora con un insieme di dati non è detto che questi siano tutti corretti o non abbiano un livello di approssimazione. All'interno di QGIS sono disponibili, di base, un insieme di funzionalità atte a calcolare la bontà delle approssimazioni o dei dati stessi presi in esame dal programma. Questo viene fatto di base con algoritmi generici per la maggior parte dei formati ma se si usa solo un tipo specifico o si vuole una approssimazione o un modello preciso è possibile ottenere il risultato desiderato attraverso plugin di terze parti.

### 1.3.3 Sistema dei plugin

Per aumentare la funzionalità di QGIS e scriptare alcune operazioni sono stati creati i plugin. Sono dei programmi dipendenti da QGIS che implementano funzionalità non presenti nel programma di base o estendono il supporto delle funzioni esistenti o i tipi di file supportati.

#### Plugin per formati di file

Vengono utilizzati per elaborare dati grezzi altrimenti difficilmente manipolabili. Un esempio possono essere i dati grezzi ottenuti da degli strumenti di misurazione con dati in output non standard. Normalmente questi file sono leggibili solo col programma

abilitato alle modifiche ma, con l'utilizzo del plugin corretto, è possibile leggerlo e elaborarlo direttamente in QGIS.

### **Plugin che alterano o aumentano le funzionalità**

La reale forza di QGIS. Permettono di ampliare le funzioni del programma semplificando o aumentando le funzionalità pre esistenti. Alcuni esempi sono i plugin della famiglia GRASS che aggiungono operazioni matematiche o gli strumenti di creazione timelapse partendo da una serie di raster temporizzati.



## Capitolo 2

# Realizzazione

Il progetto è stato realizzato attraverso la divisione in subproblemi. Questo permette la sostituzione di una componente con un'altra in modo rapido ed sicuro in quanto viene a essere modificata solo la parte del codice interessata dall'operazione e non tutte le altre componenti, che rimangono invariate.

Questo ha portato anche alla ricerca di alcune librerie per eseguire le operazioni necessarie alle funzioni scelte.

### 2.1 Divisione in sottoproblemi

Per semplificare lo sviluppo di questa applicazione e il suo sviluppo è stato diviso il progetto in quattro componenti indipendenti che dialogano tra di loro, in modo da rendere più isolate le varie componenti, rendendole meno soggette a errori non previsti. Queste si dividono in:

- Interfaccia grafica
- Lettura dei file in ingresso
- Gestione ed elaborazione dei dati
- Scrittura dei file in uscita

Questa strutturazione permette anche il riciclo del codice utilizzato per risolvere i sottoproblemi in altri progetti con la stessa necessità o la facile estensione del plugin in un secondo momento

### 2.1.1 Interfaccia grafica

Al momento della progettazione dell'applicazione è stato necessario pensare come realizzare l'inserimento dei dati da elaborare da parte dell'utente. Questo, solitamente, viene fatto in due modi:

- Attraverso l'uso di un **terminale** in cui vengono passati dei comandi che eseguono le operazioni richieste e mostra a schermo tutti i dati dei comandi eseguiti. Questo però obbliga l'utente a sapere i comandi necessari alla esecuzione dei task e ritorna a schermo dati di poco interesse per l'utente medio, molto utile invece se sei un programmatore e vuoi controllare come procedono i vari passi dei comandi eseguiti e permette di vedere subito ove avvengono gli errori e cosa li generano.
- Attraverso l'uso di una **interfaccia grafica o GUI** in cui viene selezionato, attraverso l'uso di bottoni, menu e interruttori le funzioni che si vogliono eseguire e si danno i file in input con un selettore grafico. Permette all'utente di non interessarsi a come è stata realizzata l'operazione richiesta, di non consultare guide e manuali con gli elenchi dei comandi disponibili e di ricevere a schermo solo le informazioni di cui è realmente interessato. Oltre a essere la scelta più pratica per un nuovo utente è anche quella che dà al programmatore che sviluppa questa applicazione la maggior libertà e indipendenza. Infatti la presenza di una grafica comune tra aggiornamenti diversi permette al programmatore di riscrivere interamente i comandi che vengono eseguiti al di sotto dell'interfaccia senza dover avvisare l'utente del cambiamento dei comandi all'interno del terminale.

Per come è strutturato QGIS è possibile avere entrambi implementati nel plugin ma è stato preferito dare maggiore attenzione alla componente grafica anche se è comunque possibile aprire il terminale di QGIS ed eseguire dei comandi.

In particolare in questa applicazione è stata implementata l'interfaccia grafica in quanto risulta molto più adatta all'uso effettivo dell'applicazione.

Infatti come si vede nell'immagine [link immagine] per il progetto in esame è necessario soltanto una GUI di inserimento dati in cui, in base al tipo di file inserito e al tipo di output, li elabora correttamente. Questo, su terminale, risulta invece più difficile in quanto bisognerebbe avere una lista dei driver necessari per la corretta lettura e inserirli assieme al file da leggere<sup>1</sup>.

---

<sup>1</sup>Questo viene fatto automaticamente dall'interfaccia grafica selezionando il file desiderato

## Qt4

Le librerie grafiche per Python<sup>2</sup> sono molte e varie ma la scelta è stata abbastanza facile: QT.

Questa scelta è stata dettata dal desiderio di ridurre al minimo le dipendenze esterne dal plugin. Questo comporta l'utilizzo preferenziale delle librerie utilizzate da QGIS per semplificare l'installazione delle dipendenze. In particolare, con QGIS è stato necessario scegliere anche la versione di QT con cui sviluppare il plugin. La scelta è stata fatta su QT4 ovvero la funzione supportata dalla versione 2.0 in poi di QGIS. Questo permette un enorme supporto di versioni di QGIS da parte del plugin.

La versione QT5 che verrà impiegata per le future versioni di QGIS è già interamente supportata dall'applicazione. Questo è stato possibile attraverso un attento sviluppo attraverso funzioni non deprecate e controllo continuo delle nuove funzionalità di QT5 e delle modifiche che questa fa nel funzionamento complessivo in modo di essere già compatibile quando sarà necessario passare a QT5 in quanto QT4 non sarà più prerequisito di QGIS.

Ovviamente si poteva avere lo stesso risultato utilizzando altre librerie grafiche esterne o quelle di python ma è stato scelto QT anche per avere una coerenza grafica tra le varie finestre che vanno ad aprirsi. Infatti se venisse utilizzata un'altra libreria grafica sarebbe necessario una rielaborazione della grafica standard della libreria in modo che si comporti coerentemente rispetto a QT per ogni sistema operativo, il che porterebbe a una complicazione del codice non necessaria.

### 2.1.2 Leggere e scrivere i dati

Il plugin prende in ingresso i file contenenti i dati necessari per l'applicazione al modello matematico descritto dall'equazione 1.1. Quindi, una volta eseguite le operazioni produce l'output corrispondente all'erosione per il numero di anni dati in input.

Le operazioni di lettura e scrittura sono le operazioni più sensibili in questo applicativo in quanto non c'è modo di controllare la correttezza dei dati in input dal punto di vista fisico/geologico, solo dal punto di vista informatico di buona formattazione.

---

<sup>2</sup>Il linguaggio in cui è stato scritto il plugin

## Input

Per una questione di coerenza rispetto a QGis è stato deciso di passare i dati in input come file, in modo che possano essere facilmente condivisi e possano interagire più facilmente con QGis stesso. Per fare ciò è stato necessario fare una ricerca sui tipi di file utilizzati per i dati georeferenziati.

Per questo motivo è stata fatta una ricerca sui formati di file supportati nativamente da QGis che fossero di tipo raster<sup>3</sup> e quali fossero supportati attraverso plugin esterni.

E' stata quindi usata una delle funzioni del core di QGis per il riconoscimento automatico dei formati riconosciuti da QGis in modo da eseguire l'elaborazione del singolo file col driver corrispondente installato migliore.

Questo permette di avere un plugin più elastico, in quanto non necessita di aggiornamento se viene aggiunto il supporto a un tipo aggiuntivo di file per QGis ma recupera direttamente dalle configurazioni di QGis i dati necessari per la lettura, decodifica e rielaborazione del file in questione.

In oltre, il codice di gestione per i file in input è integrato nell'interfaccia grafica in modo da rendere semplice l'inserimento di dati all'interno del programma stesso. Supporta comunque l'utilizzo attraverso il terminale di QGis anche se ne è sconsigliato l'uso ai meno esperti e a chi si approccia per la prima volta a questo plugin.

Come vedete dall'immagine il plugin apre una finestra di sistema differente in base al sistema operativo utilizzato, in quanto la gestione della scelta del file è delegato a esso. Una volta avvenuta la scelta questa finestra si chiuderà passando la stringa corrispondente al path del file aperto al campo adiacente presente nella gui principale. Se l'utente lo desidera può editare o direttamente inserire il path al file scelto direttamente nel campo senza aprire la finestra per la scelta della cartella.

## Output

Come per l'input, l'output può essere gestito in due modi: attraverso il terminale python o via interfaccia grafica.

In entrambi i casi è possibile indicare il path e il nome del file di output e il formato in cui lo si vuole salvare. Il formato è scelto dall'utente da un elenco di formati disponibili nativamente da QGis con l'aggiunta dei formati disponibili da i plugin

---

<sup>3</sup>Unico tipo di dati che il plugin accetta in input in quanto è l'unica struttura dati adatta a contenere i dati che ci servono per l'elaborazione

installati<sup>4</sup> con driver per la scrittura. Per come è costruito QGis è possibile che alcuni formati non siano disponibili per la scrittura in quanto i driver sono di solo lettura. In questo caso, indipendentemente dalla scelta di utilizzare il terminale o una gui il plugin lancerà una finestra con l'errore corrispondente all'errore lanciato dal driver. Questo viene fatto in quanto è possibile che i formati in input non possono essere correttamente codificati nel formato di output scelto.

In ogni caso si consiglia che il formato di input e output dei file sia lo stesso per avere il minor problemi dati dalla decodifica e codifica degli stessi.

Nel caso i driver non vengono registrati all'interno dell'elenco dei driver per l'output di QGis seguendo le api sarà necessario implementarle sequendole per poterle utilizzare all'interno del plugin.

### 2.1.3 Elaborazione dati

Una volta ricevuti i dati in input si pensa a elaborarli. Per farlo è stato necessario generare una struttura dati idonea ai formati in input. E' stato quindi deciso di implementare l'elaborazione attraverso la struttura dati raster. A sua volta i dati geomorfici del raster sono all'interno di una matrice, che è l'oggetto con cui il plugin esegue le operazioni.

In oltre è stato necessario trovare una o più librerie che implementassero le funzionalità richieste per l'elaborazione dei dati raster mantenendo al minimo le dipendenze esterne. In caso di assenza di librerie si sarebbe dovuto implementare tutto da zero. Per nostra fortuna sono disponibili alcune librerie che eseguono calcoli in raster.

#### Grass

Plugin standard che dà accesso a tutte le funzionalità avanzate di GRASS GIS. Dal punto di vista tecnico non è una libreria ma un insieme di librerie che gestiscono raster e elaborazioni avanzate dei dati. Essendo di facile installazione e, in alcune versioni, distribuito direttamente con QGis è un ottimo sistema per avere facilmente una libreria per l'elaborazione dei raster.

Il plugin descritto in questa tesi richiede l'installazione di Grass per il suo corretto funzionamento anche se non usa direttamente la libreria di Grass.

---

<sup>4</sup>Solo se questi registrano il formato nell'elenco nativamente presente in QGis come indicato dalle API per sviluppatori.

La libreria di Grass non è stata usata direttamente ma solo una delle sue componenti per avere maggiore libertà nei formati disponibili di file. Infatti, a meno di un ridotto numero di plugin, non è possibile aumentare i tipi di file supportati da Grass in quanto si limitano solo ai formati forniti con l'installazione standard di QGIS.

### **Osgeo**

Una delle librerie utilizzate da Grass. Formalmente è una libreria per il supporto di operazioni su raster, in pratica è un porting di Numpy per i formati di file contenenti dati georeferenziati.

Nel nostro caso è stato utilizzato per elaborare i dati dei file raster.

# Bibliografia

- [Lut11] Mark Lutz. *Programming Python*. A cura di O'Reilly. O'Reilly, 2011.
- [Pan09] Lorenzo Pantieri. *L'arte di gestire la bibliografia con L<sup>A</sup>T<sub>E</sub>X*. 2009. URL: [http://www.lorenzopantieri.net/LaTeX\\_files/Bibliografia.pdf](http://www.lorenzopantieri.net/LaTeX_files/Bibliografia.pdf).
- [PG11] Lorenzo Pantieri e Tommaso Gordini. *L'arte di scrivere con L<sup>A</sup>T<sub>E</sub>X*. 2011. URL: [http://www.lorenzopantieri.net/LaTeX\\_files/ArteLaTeX.pdf](http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf).
- [Sita] URL: <http://www.qgis.org/it/>.
- [Sitb] URL: <http://creativecommons.org/licenses/by-sa/3.0/>.
- [Zen13] Michele Zen. “Metodi e strumenti per la costruzione di territori virtuali per l'applicazione di modelli e di scenari ambientali”. Tesi di laurea mag. Università Ca Foscari Venezia, 2013/2014.