

**TRƯỜNG ĐẠI HỌC NÔNG LÂM TP.HỒ CHÍ MINH
KHOA MÔI TRƯỜNG VÀ TÀI NGUYÊN**



TIỂU LUẬN TỐT NGHIỆP

**ỨNG DỤNG ARC ENGINE XÂY DỰNG PHẦN MỀM
HỖ TRỢ PHÂN TÍCH DÒNG CHẢY MẶT TẠI TRƯỜNG
ĐẠI HỌC NÔNG LÂM**

Họ và tên sinh viên: NGUYỄN VĂN ANH TUẤN

Ngành: Hệ thống thông tin môi trường

Niên khóa: 2010 – 2014

Tp. Hồ Chí Minh, tháng 6 năm 2014

ỨNG DỤNG ARC ENGINE XÂY DỰNG PHẦN MỀM HỖ TRỢ PHÂN TÍCH DÒNG CHẢY MẶT TẠI TRƯỜNG ĐẠI HỌC NÔNG LÂM

Tác giả

NGUYỄN VĂN ANH TUẤN

Giáo viên hướng dẫn

ThS. KHUU MINH CẢNH

Tp. Hồ Chí Minh, 6/ 2014

LỜI CẢM ƠN

Trước tiên, tôi xin chân thành cảm ơn thầy ThS. Khuru Minh Cảnh và những người đã hướng dẫn, chỉ bảo tận tình và động viên tôi trong suốt thời gian qua, giúp tôi hoàn thành bài báo cáo tốt nghiệp này. Trân trọng cảm ơn Ban lãnh đạo Sở Khoa Học Công Nghệ Tp. Hồ Chí Minh đã tạo điều kiện cho tôi được thực tập tại cơ quan.

Tôi xin gửi lời tri ân sâu sắc đến quý thầy cô cùng KS. Nguyễn Duy Liêm và KS. Lê Hoàng Tú, trường Đại học Nông Lâm Tp. Hồ Chí Minh đã tận tình giảng dạy, truyền đạt những kiến thức quý giá dành cho tôi trong bốn năm học tập tại trường.

Tôi cũng cảm ơn những người bạn đồng hành cùng tôi trong quãng đời sinh viên, những người đã luôn giúp đỡ tôi khi tôi gặp khó khăn, sẵn sàng chia sẻ cho tôi những điều hay, lẽ phải và cũng là nguồn động lực để tôi phấn đấu vươn lên.

Cuối cùng, để có được thành quả như ngày hôm nay, con xin nói lời biết ơn chân thành đối với cha mẹ, những người đã sinh thành nên con, chăm sóc, nuôi dạy con thành người và tạo điều kiện cho con được học tập.

Sinh viên thực hiện
Nguyễn Văn Anh Tuấn
Bộ môn Tài nguyên và GIS
Khoa Môi trường và Tài nguyên
Trường Đại học Nông Lâm Tp. Hồ Chí Minh

MỤC LỤC

TRANG TỰA	i
LỜI CẢM ƠN.....	ii
MỤC LỤC	iii
DANH MỤC VIẾT TẮT.....	iv
DANH MỤC HÌNH ẢNH	v
CHƯƠNG 1. MỞ ĐẦU	1
1.1. Tính cấp thiết của đề tài	1
1.2. Mục tiêu nghiên cứu.....	2
1.3. Phạm vi nghiên cứu.....	2
CHƯƠNG 2. TỔNG QUAN	3
2.1. Thông tin về địa bàn nghiên cứu.....	3
2.2. Tổng quan các thuật toán về dòng chảy trên thế giới và quy trình chung cho các thuật toán định dòng.....	4
2.2.1. Thuật toán D8	5
2.2.2. Thông tin về một số công cụ phân tích dòng chảy	7
2.3. Tổng quan về thuật toán tìm đường đi ngắn nhất (Floyd)	9
2.4. Độ quy và tính toán đệ quy	11
2.5. Giới thiệu ngôn ngữ lập trình Visual Basic	14
2.6. Giới thiệu ngôn ngữ lập trình Python	16
2.7. Mô hình địa hình và cấu trúc cơ sở dữ liệu.....	19
CHƯƠNG 3. PHƯƠNG PHÁP THỰC HIỆN	22
3.1. Lược đồ phương pháp thực hiện	22
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU.....	24
4.1. Kết quả thực thi chương trình	24
4.1.1. Các module trong phần mềm	24
4.1.2. Cài đặt thuật toán D8	27
4.1.3. Thuật toán xác định dòng chảy tích lũy.....	28
4.1.4. Ứng dụng Arc Engine để thể hiện bản đồ dòng chảy	30
CHƯƠNG 5. KẾT LUẬN VÀ KIẾN NGHỊ.....	35
5.1. Kết luận	35
5.2. Đề xuất hướng phát triển.....	35
TÀI LIỆU THAM KHẢO	36
PHỤ LỤC	38

DANH MỤC VIẾT TẮT

DEM	Mô hình độ cao số (Digital Elevation Model)
GIS	Hệ thống thông tin địa lý (Geographic Information System)
VB	Ngôn ngữ lập trình (Visual Basic)
TauDEM	Terrain Analysis Using Digital Elevation Models
DTM	Mô hình địa hình số (Digital Terrain Model)
TIN	Mô hình lưới tam giác (Triangulated Irregular Network)
HĐH	Hệ điều hành
PEP	Python Enhancement Proposal
GPL	Giấy phép công cộng (General Public License)

DANH MỤC HÌNH ẢNH

Hình 2-1. Phương pháp xác định bậc Straler.....	4
Hình 2-2. Mô tả hướng dòng chảy trong D8	6
Hình 2-3. Công cụ Hydrology trong Arcmap.....	7
Hình 2-4. Các công cụ có trong TauDEM.....	9
Hình 2-5. Hình minh họa về bài toán thác Hà Nội.....	13
Hình 2-6. DEM cấu trúc dạng phân tử lưới đều Grid.....	19
Hình 2-7. DEM cấu trúc dạng TIN.....	20
Hình 2-8. Những đường đồng mức độ cao	20
Hình 2-9. DEM thể hiện dạng Grid	21
Hình 2-10. DEM thể hiện dạng XYZ	21
Hình 3-1. Sơ đồ thực hiện phần mềm.....	23
Hình 4-1. Giao diện người dùng.....	24
Hình 4-2. Dữ liệu DEM trong file Text.....	25
Hình 4-3. Dữ liệu về hướng dòng chảy được lưu ở dạng Text	27
Hình 4-4. Dữ liệu về dòng chảy tích lũy được tính toán bằng phần mềm và lưu ở dạng Text.	29
Hình 4-5. Bộ thư viện Arc Engine của ESRI	30
Hình 4-6. Sử dụng các ArcGis Window Form trong Toolbox lên form hiển thị	31
Hình 4-7. Hiển thị dữ liệu DEM khu vực Nông Lâm trên form.....	33
Hình 4-8. Dữ liệu dòng chảy của khu vực nghiên cứu sau khi thực hiện tính toán trên phần mềm.....	34

CHƯƠNG 1. MỞ ĐẦU

1.1. Tính cấp thiết của đề tài

Trường Đại học Nông Lâm với diện tích rộng khoảng 137 ha với khuôn viên nhiều cây xanh là nơi lý tưởng cho việc học tập và nghiên cứu của sinh viên. Do đó, một trong những điều kiện cần để trường nâng cao chất lượng đào tạo là cơ sở hạ tầng nói chung và hệ thống thoát nước nói riêng phải tiên tiến, đặc biệt trong điều kiện biến đổi khí hậu với những cơn mưa lớn và thất thường. Bên cạnh đó, với địa hình có độ cao tương đối như khu vực Thủ Đức tạo nên các dòng nước lớn gây sủi mòn đất và có thể gây ra nhiều nguy hiểm đối với nhiều sinh viên. Theo thông tin trên website vnexpress.net (An Nhơn, 2013), vào đầu tháng 7/2013, một sinh viên của trường Đại học Kinh Tế Luật TP. Hồ Chí Minh bị nước cuốn trôi trong một trận mưa lớn. Từ đây, chúng ta phải nhìn nhận và nên xem xét lại hệ thống thoát nước tại trường. Bước đầu trong công tác này là xác định dòng chảy trên bề mặt để địa hình và khả năng thoát nước của hệ thống thoát nước từ đó làm nền tảng tham khảo khi trường có đầu tư mở rộng hoặc sửa chữa nâng cấp.

Phân tích dòng chảy mặt là xác định các đặc trưng địa hình như hướng dòng chảy, dòng chảy tích lũy, độ dốc, xác định lưu vực,... trên dữ liệu đầu vào là mô hình số độ cao (DTM, DEM). Phân tích dòng chảy mặt được sử dụng nhiều trong các ứng dụng, nghiên cứu về lưu vực, các phân tích về địa hình để xây dựng hệ thống cấp thoát nước, xây dựng hệ thống đường giao thông, công trình nhà ở phù hợp với điều kiện khí hậu và địa hình của khu vực. Để thực hiện phân tích dòng chảy mặt cần ứng dụng nhiều thuật toán xác định hướng dòng chảy như D8, D16,... khi đã xác định được hướng dòng chảy tại mỗi điểm trên bề mặt địa hình có thể xác định được dòng chảy trên bề mặt (dòng chảy có thể là sông, suối,...). Qua đó, có thể xác định được các khu vực chịu tác động của dòng chảy trong khu vực địa hình.

Hiện nay, GIS đang phát triển nhanh chóng trên thế giới và du nhập vào Việt Nam nhiều năm qua đã tạo điều kiện cho các nghiên cứu cũng như các ứng dụng của GIS được đi vào thực tiễn cuộc sống thường ngày. Việc tích hợp GIS và phân tích

dòng chảy mặt đã góp phần xây dựng các ứng dụng như phân tích lưu vực, hỗ trợ ra quyết định xây dựng hệ thống đường giao thông, hệ thống cấp thoát nước...

Do đó tiểu luận “ *Ứng dụng Arc Engine xây dựng phần mềm hỗ trợ phân tích dòng chảy bề mặt tại trường Đại Học Nông Lâm*” được thực hiện nhằm xây dựng, thiết kế phần mềm để hỗ trợ việc thành lập bản đồ dòng chảy từ đó xác định được các khu vực chịu tác động của dòng chảy làm cơ sở đưa ra các quyết định nâng cấp cơ sở hạ tầng của trường.

1.2. Mục tiêu nghiên cứu

Xây dựng phần mềm hỗ trợ phân tích dòng chảy mặt với thuật toán D8. Cài đặt tính toán để phân tích dòng chảy theo địa hình tại khu vực trường Đại học Nông Lâm. Từ đó, làm cơ sở cho việc cải tạo, bổ sung, sửa chữa hệ thống thoát nước tại trường. Các mục tiêu cụ thể của đề tài:

- Xây dựng giao diện người dùng cho phần mềm.
- Cài đặt thuật toán D8 bằng ngôn ngữ lập trình Visual Basic.
- Ứng dụng Arc Engine để thể hiện bản đồ dòng chảy mặt.
- Xác định khu vực ảnh hưởng của dòng chảy.

1.3. Phạm vi nghiên cứu

- Về khu vực địa lý: Được thực hiện trong giới hạn ranh giới của trường Đại học Nông Lâm.
- Thuật toán: Giới hạn sử dụng các thuật toán phân tích dòng chảy đơn (D8), các công cụ thao tác với dữ liệu DEM.

CHƯƠNG 2. TỔNG QUAN

2.1. Thông tin về địa bàn nghiên cứu

Trường Đại học Nông Lâm TP. Hồ Chí Minh là trường đại học công lập đa ngành, đa lĩnh vực có nhiệm vụ đào tạo bậc đại học và sau đại học, nghiên cứu khoa học và chuyển giao công nghệ trong lĩnh vực Nông Lâm Ngư Nghiệp và các lĩnh vực liên quan. Từ năm 2000 trường mở rộng đào tạo sang các lĩnh vực khác như: Công nghệ thông tin, Công nghệ môi trường, Công nghệ sinh học, Ngoại ngữ và Sư phạm Kỹ thuật nông nghiệp, Công nghệ ô tô, Công nghệ nhiệt lạnh, Cơ điện tử, Điều khiển tự động, Công nghệ địa chính, Kế toán, Quản trị kinh doanh, Quản lý thị trường bất động sản, Hệ thống tin địa lý.

Từ khi thành lập vào năm 1955 cho đến hiện nay, Trường Đại học Nông Lâm TP. Hồ Chí Minh trong quá trình phát triển đã nhiều lần thay đổi về cơ cấu tổ chức, mục tiêu đào tạo, chương trình đào tạo, nội dung đào tạo cho phù hợp với yêu cầu qua từng thời kỳ phát triển kinh tế xã hội của đất nước. Hiện nay Trường Đại học Nông Lâm TP. Hồ Chí Minh có 12 khoa, 03 bộ môn trực thuộc trường, 01 viện nghiên cứu Công nghệ sinh học và môi trường, 13 trung tâm, 12 phòng ban chức năng, 01 Phân hiệu tại tỉnh Gia Lai và 01 Phân hiệu tại tỉnh Ninh Thuận.

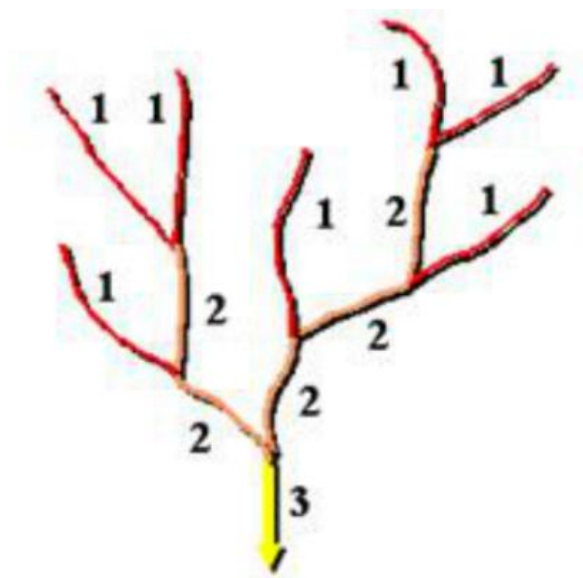
Đại Học Nông Lâm Thành Phố Hồ Chí Minh là một trường đại học đa ngành, trực thuộc Bộ Giáo dục và Đào tạo, tọa lạc trên khu đất rộng 118 ha, thuộc khu phố phường Linh Trung, Quận Thủ Đức, Thành phố Hồ Chí Minh và Huyện Dĩ An (Tỉnh Bình Dương). Hiện nay, tổng diện tích đất nhà trường đang quản lý sử dụng là 137.0156 ha, với 5 khu đất trong đó có: 5 nhà học (nhà Phụng Vỹ, nhà Cẩm Tú, Nhà Hướng Dương, nhà Rạn Đông và nhà Tường Vy), 127 nhà thí nghiệm, thực hành, 1 nhà tập và thi đấu thể dục thể thao, 6 khu kí túc xá cho trên 3750 sinh viên, 1 thư viện. Trường thuộc khu vực có địa hình tương đối cao trong TP. Hồ Chí Minh, dựa trên dữ liệu DEM thu thập được thì độ cao địa hình của trường cao nhất là 44m, thấp nhất là 7m và độ cao trung bình là 20.9m (Phòng quản trị vật tư, 2014).

2.2. Tổng quan các thuật toán về dòng chảy trên thế giới và quy trình chung cho các thuật toán định dòng

Các lưu vực sông có liên quan mật thiết đến đời sống của con người. Chính vì thế, các hoạt động nghiên cứu về lưu vực sông đã được con người thực hiện từ rất sớm với nhiều hướng tiếp cận khác nhau.

Ngày nay với sự phát triển của công nghệ thông tin đã mở ra một hướng đi mới trong nghiên cứu lưu vực: nghiên cứu lưu vực sông với sự hỗ trợ của Hệ Thống Thông Tin Địa Lý (GIS).

Năm 1957, sự ra đời của khái niệm Strahler và định luật Horton – Strahler đã tạo cơ sở cho việc hình thành các thuật toán xác định dòng chảy và phân chia ranh giới lưu vực sông. Cấp Strahler là một thuộc tính được gán cho mọi đoạn sông trong một mạng lưới dòng chảy. Một chuỗi các đoạn sông có cùng cấp Strahler tạo thành một dòng Strahler. Cấp của dòng Strahler được xem như một thước đo để đánh giá kích cỡ và sự phức tạp về cấu trúc của một mạng lưới dòng chảy. Theo định luật Horton – Strahler, giá trị cơ sở của cấp Strahler là cấp 1, khi 2 dòng Strahler cùng cấp gặp nhau sẽ tạo ra một dòng Strahler với cấp lớn hơn (Strahler, A. N, 1957).



Hình 2-1. Phương pháp xác định bậc Straler.

Từ những khái niệm cơ sở ban đầu các tác giả khác nhau đã phát triển nhiều thuật toán xác định dòng chảy khác nhau mà cơ bản nhất là thuật toán D8. Các thuật toán ra đời sau này được đề xuất để bổ sung, bù đắp những thiếu sót của các thuật toán

ra đời trước đó. Các thuật toán phân tích dòng chảy như: D8, FD8, Rho8, FRho8, DEMON, và $D \infty$... (Bradley Miller 2010)

O'Callaghan & Mark (1984) giới thiệu thuật toán dòng chảy đầu tiên, mô hình dòng chảy với 8 hướng (D8). Phương pháp D8 được sử dụng rộng rãi do tính toán đơn giản và nó được đưa vào trong các phần mềm GIS phổ biến.

Freeman (1991) và Quinn et al. (1991) cùng đưa ra thuật toán dòng chảy FD8, thuật toán cho phép xác định nhiều hướng dòng chảy từ ô trung tâm đến các ô lân cận. Tỷ lệ của mỗi hướng dòng chảy được chỉ định dựa trên Slope - weight. FD8 được thiết lập để hạn chế thiếu sót của D8, nhưng có một vấn đề với FD8 là nó hạn chế việc hình thành dòng chảy tích lũy trong lưu vực, vì tại mỗi điểm có nhiều hướng dòng chảy và sẽ có một vài hướng không góp phần vào việc hình thành dòng chảy tích lũy. Chính vì điều này, để thực hiện tính dòng chảy tích lũy người ta vẫn sử dụng D8.

Cả hai thuật toán D8 và FD8 mặc dù có khái niệm khác nhau nhưng các chức năng xử lý dữ liệu độ cao tương tự nhau. Ngoài ra, các thuật toán D8, FD8 không thể hiện dòng chảy tích lũy của lưu vực. Với những thiếu sót của D8, FD8 thì Fairfield và Leymarie (1991) nhằm sửa đổi các thiếu sót của D8, FD8 đã đưa ra các thuật toán Rho8 và FRho8 tương ứng. Rho8 và FRho8 sử dụng mức độ ngẫu nhiên để tính toán dòng chảy.

Sau này có nhiều thuật toán dòng chảy ra đời như: Costa-Cabral & Burgess (1994) với DEMON; Desmet và Govers (1996) với Flux decomposition; Tarboton (1997) với Dinf; Gallant và Wilson (2000) với TAPES-G cải tiến từ DEMON; Qin et al (2007) với MFD-md; Wan MuhdHairi bin Wan AbKarim (sau 2012) với D16.

Quy trình chung cho các thuật toán xác định hướng dòng chảy:

- **Bước 1:** Tiền xử lý DEM: Để lấp các điểm lồi, lõm không mong muốn (Fill), loại bỏ những khu vực bằng phẳng.
- **Bước 2:** Tính toán hướng dòng chảy trên mỗi ô (pixel).
- **Bước 3:** Tính toán dòng chảy tích lũy.

2.2.1. Thuật toán D8

D8 là thuật toán xác định hướng dòng chảy phổ biến nhất hiện nay, nó sử dụng 1 cửa sổ các điểm ảnh với kích thước 3x3 như hình minh họa. Hướng dòng chảy của một điểm bất kỳ trong cửa sổ được xác định trên cơ sở so sánh độ chênh cao của điểm đó

với 8 điểm xung quanh. Hướng dòng chảy được xác định là hướng tới điểm có độ chênh cao chia cho khoảng cách lớn nhất. Quá trình tính toán được lặp lại để xác định hướng dòng chảy cho toàn bộ các điểm trong lưu vực. Góc giữa các hướng là 45^0 và chỉ được lựa chọn 1 trong 8 hướng.

Cách tính thuật toán D8:

- Dòng chảy từ 1 điểm sang 1 trong 8 điểm lân cận.
- Nguyên tắc là chọn hướng có Slope lớn nhất.



Hình 2-2. Mô tả hướng dòng chảy trong D8

$$\text{Slope} = \arctan\left(\frac{\Delta Z}{d}\right)$$

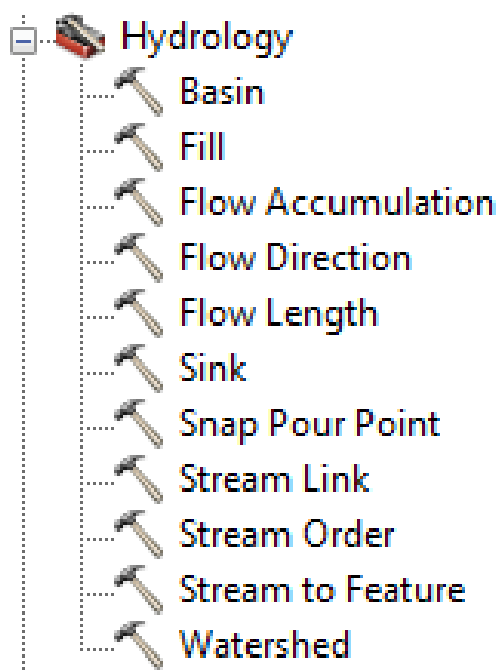
- Với:
 - ΔZ là độ chênh cao.
 - d là khoảng cách giữa các ô: các ô đông, tây, nam, bắc có giá trị bằng 1. Các ô đường chéo có giá trị là $\sqrt{2}$.

Việc D8 chỉ cho phép lựa chọn ra một hướng trong tám hướng là hạn chế trong thuật toán này, trong thực tế nước có thể chảy về rất nhiều hướng vì vậy thuật toán chưa mô tả chính xác được hướng di chuyển của dòng chảy trong lưu vực hay một khu vực địa hình làm ảnh hưởng đến việc tính toán dòng chảy tích lũy. Ngoài ra, trong trường hợp Slope của điểm trung tâm đến một vài điểm (có thể là tám điểm) lân cận bằng nhau thì thuật toán sẽ lựa chọn ngẫu nhiên một hướng trong những hướng có Slope bằng nhau đó. Với những hạn chế nêu trên thì D8 không phù hợp để áp dụng cho các khu vực có địa hình bằng phẳng như đồng bằng, thung lũng.

2.2.2. Thông tin về một số công cụ phân tích dòng chảy

2.2.2.1. Tool Hydrology

Hydrology là một công cụ trong Arcmap được sử dụng để thành lập mạng lưới sông hoặc phân định lưu vực dựa trên mô hình số độ cao DEM. Trong công cụ này tính hướng dòng chảy dựa trên thuật toán D8 (Arcgis Resource Center. 2011).



Hình 2-3. Công cụ Hydrology trong Arcmap

Trong đó:

- Basin được sử dụng để tạo ra một raster phân chia lưu vực.
- Fill được dùng để hiệu chỉnh các điểm lồi, lõm không mong muốn trên dữ liệu.
- Flow accumulation thực hiện tính toán để xác định dòng chảy tích lũy. Sự tích lũy dòng chảy cho một ô nào đó trong khu vực trên mô hình DEM được xác định bằng cách tính tổng số ô lưới tập trung nước về ô đó theo hướng dòng chảy. Bản đồ tích lũy dòng chảy cũng chính là bản đồ sông suối của lưu vực trong điều kiện tự nhiên.
- Flow direction xây dựng dữ liệu raster về hướng dòng chảy. Hướng dòng chảy cho một điểm bất kỳ trong lưu vực được xác định trên cơ sở so sánh độ chênh cao của điểm đó với 8 điểm xung quanh. Hướng dòng chảy được xác định là hướng tới điểm có độ chênh cao chia cho khoảng cách

lớn nhất. Quá trình tính toán được lặp lại để xác định hướng dòng chảy cho toàn bộ các điểm trong lưu vực.

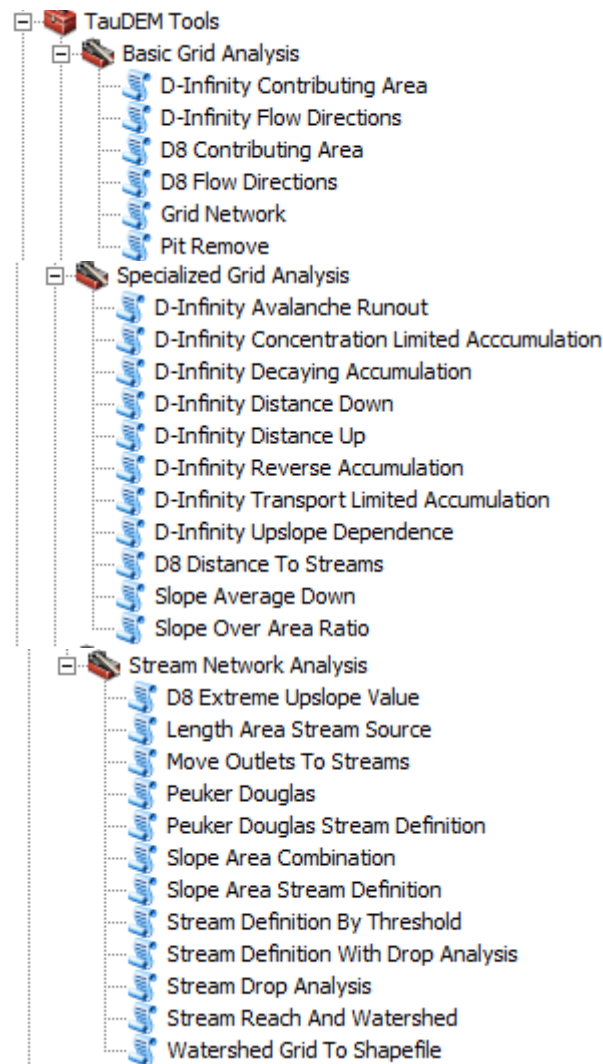
- Flow length tính toán khoảng cách giữa thượng lưu và hạ lưu, hoặc độ dài dòng chảy trong lưu vực.
- Sink được sử dụng để xác định tất cả các điểm trũng hoặc những vùng không xác định được hướng dòng chảy. Sink sẽ gán giá trị nhất định cho vùng này để phù hợp với độ cao của các ô lân cận.
- Snap Pour Point sẽ xác định ranh giới lưu vực ứng với cửa xả xác định.
- Stream Link liên kết các dòng chảy với nhau tạo thành một hệ thống dòng chảy trong lưu vực.
- Stream Order xác định bậc của các dòng chảy trong lưu vực.
- Stream to Feature là chuyển dữ liệu dòng chảy sang dạng vector.
- Watershed dùng để phân chia lưu vực.

2.2.2.2. Công cụ TauDEM

TauDEM viết tắt của Terrain Analysis Using Digital Elevation Models là một bộ các công cụ được sử dụng để khai thác, phân tích thông tin thủy văn từ mô hình số độ cao (DEM). TauDEM cung cấp các chức năng như:

- Tính toán hướng dòng chảy và độ dốc.
- Sử dụng nhiều phương pháp tính hướng dòng chảy (hướng dòng chảy đơn và dòng chảy đa).
- Sử dụng nhiều phương pháp để mô tả dòng chảy bao gồm các phương pháp dựa trên địa hình có mật độ dòng chảy dễ thay đổi.
- Phân định lưu vực sông và xác định tiểu lưu vực của một hệ thống sông và mối liên hệ của các thuộc tính trong lưu vực để thiết lập mô hình thủy văn.
- Các chức năng chuyên để phân tích địa hình:
 - o Tính tỉ lệ giữa độ dốc với diện tích khu vực là cơ sở để xác định khả năng giữ nước của địa hình.
 - o Tính khoảng cách từ đỉnh núi đến dòng chảy theo mặt phẳng ngang, mặt phẳng đứng.

- Ước lượng dòng chảy tích lũy tại khu vực.



Hình 2-4. Các công cụ có trong TauDEM

2.3. Tổng quan về thuật toán tìm đường đi ngắn nhất (Floyd)

Thuật toán Floyd – Warshall còn được gọi là thuật toán Floyd được Robert Floyd tìm ra năm 1962. Thuật toán Floyd là một thuật toán giải quyết bài toán đường đi ngắn nhất trong một đồ thị có hướng, có cạnh mang trọng số dương dựa trên khái niệm các đỉnh trung gian (Lê Minh Hoàng, 2003).

Cho một đồ thị có hướng, có trọng số $G = (V, E)$ với n đỉnh và m cạnh. Bài toán đặt ra là hãy tính tất cả các $d(u, v)$ là khoảng cách từ u tới v . Rõ ràng là có thể áp dụng thuật toán tìm đường đi ngắn nhất xuất phát từ một đỉnh với n khả năng chọn đỉnh xuất phát. Nhưng còn có cách làm gọn hơn nhiều, cách làm này rất giống với thuật toán

Wardhall nó được hiểu như sau: Từ ma trận trọng số c , thuật toán Floyd tính lại các $c[u,v]$ thành độ dài đường đi ngắn nhất từ u tới v :

Với mọi đỉnh k của đồ thị được xét theo thứ tự từ 1 tới n , xét mọi cặp đỉnh u, v . Cực tiểu hóa $c[u,v]$ theo công thức:

$$c[u,v] = \min(c[u,v], c[u,k] + c[k,v])$$

Tức là nếu như đường đi từ u tới v đang có dài hơn đường đi từ u tới k cộng với đường đi từ k tới v thì ta hủy bỏ đường đi từ u tới v .

Gọi $c^k[u,v]$ là độ dài đường đi ngắn nhất từ u tới v mà chỉ đi qua đỉnh trung gian thuộc tập $\{1, 2, \dots, k\}$. Rõ ràng khi $k=0$ thì $c^0[u,v]$ (đường đi ngắn nhất là đường đi trực tiếp).

Giả sử đã tính được các $c^{k-1}[u,v]$ thì $c^k[u,v]$ sẽ được xây dựng như sau:

Nếu đường đi ngắn nhất từ u tới v mà chỉ qua các đỉnh trung gian thuộc $\{1, 2, \dots, k\}$ mà lại:

- Không đi qua đỉnh k thì tức là chỉ qua đỉnh trung gian thuộc tập $\{1, 2, \dots, k\}$ thì: $c^k[u,v] = c^{k-1}[u,v]$
- Có đi qua đỉnh k thì đường đi đó sẽ là nối của một đường đi từ u tới k và một đường đi từ k tới v , hai đường đi này chỉ đi qua các đỉnh trung gian thuộc tập $\{1, 2, \dots, k\}$.

$$c^k[u,v] = c^{k-1}[u,k] + c^{k-1}[k,v]$$

Vì muốn $c^k[u,v]$ là cực tiểu nên suy ra: $c^k[u,v] = \min(c^{k-1}[u,v], c^{k-1}[u,k] + c^{k-1}[k,v])$.

Và cuối cùng, $c^n[u,v]$: Độ dài đường đi ngắn nhất từ u tới v mà chỉ đi qua các đỉnh trung gian thuộc tập $\{1, 2, \dots, k\}$.

Khi cài đặt, thì sẽ không có các khái niệm $c^k[u,v]$ mà sẽ thao tác trực tiếp trên các trọng số $c[u,v]$. $c[u,v]$ tại từng bước tối ưu thứ k sẽ được tính toán để tối ưu qua các giá trị $c[u,v]$; $c[u,k]$; $c[k,v]$ tại bước thứ $k-1$. Tính chính xác của cách cài đặt dưới dạng ba vòng lặp For lồng vào nhau có thể thấy được sự tối ưu bậc câu chỉ làm tăng tốc độ tối ưu các $c[u,v]$ trong mỗi bước.

Tất cả các cài đặt thuật toán Floyd đều sử dụng ma trận trọng số chứ không sử dụng danh sách cạnh hay danh sách kề có trọng số, nên đều đưa về đồ thị đầy đủ và đem trọng số $+\infty$ gán cho những cạnh không có trong đồ thị ban đầu. Trên máy tính thì không có khái niệm trừu tượng $+\infty$ nên ta phải chọn một số dương đủ lớn, số lớn hơn tất cả các trọng số của các đường đi cơ bản để cho dù đường đi thật có tồi tệ tới đâu

vẫn tốt hơn đường đi trực tiếp theo cạnh tương đương ra đó. Vậy nên nếu đồ thị cho số đỉnh cũng như trọng số các cạnh vào khoảng 300 chẳng hạn thì giá trị đó không thể chọn trong phạm vi Integer hay Word. Ma trận c sẽ phải là khai báo là ma trận LongInt và giá trị hằng số $\max C$ trong các chương trình trên phải đổi lại là $300 * 299 + 1$ điều đó có thể gây ra nhiều phiền toái, chẳng hạn như vấn đề lãng phí bộ nhớ. Để khắc phục, có thể cài đặt bằng danh sách kề kèm trọng số hoặc sử dụng những kỹ thuật đánh dấu khéo léo trong từng trường hợp cụ thể. Tuy nhiên, có một điều chắc chắn là khi đồ thị cho số đỉnh cũng như trọng số các cạnh vào khoảng 300 thì các trọng số $c[u,v]$ trong thuật toán Floyd và các nhãn $d[v]$ trong ba thuật toán còn lại chắc chắn không thể khai báo là Integer được.

Khác với các bài toán đại số hay hình học có nhiều cách giải thì chỉ cần nắm vững một cách cũng có thể coi là đạt yêu cầu, nhưng thuật toán tìm đường đi ngắn nhất bộc lộ rất rõ ưu, nhược điểm trong từng trường hợp cụ thể, ví dụ như số đỉnh của đồ thị quá lớn làm cho không thể biểu diễn bằng ma trận trọng số thì thuật toán Floyd gặp khó khăn. Vì vậy, yêu cầu trước tiên là phải hiểu bản chất và thành thạo trong việc cài đặt thuật toán trên để có thể sử dụng chúng một cách uyển chuyển trong từng trường hợp cụ thể.

2.4. Độ quy và tính toán đệ quy

2.4.1.1. Khái niệm về đệ quy

Trong toán học và khoa học máy tính, một đối tượng là đệ quy nếu nó được định nghĩa qua chính nó hoặc một đối tượng khác cùng dạng với chính nó bằng quy nạp (Lê Minh Hoàng, 2003).

Ví dụ: Đặt hai chiếc gương cầu đối diện nhau. Trong chiếc gương thứ nhất chứa hình chiếc gương thứ hai. Chiếc gương thứ hai lại chứa hình chiếc gương thứ nhất nên tất nhiên nó chứa lại hình ảnh của chính nó trong chiếc gương thứ nhất... Ở góc nhìn hợp lý, ta có thể thấy một dãy ảnh vô hạn của cả hai chiếc gương.

2.4.1.2. Giải thuật về đệ quy

Nếu lời giải của một bài toán P được thực hiện bằng lời giải của bài toán P' có dạng giống như P thì đó là một lời giải đệ quy. Giải thuật tương ứng với lời giải như vậy gọi là giải thuật đệ quy. Trong giải thuật đệ quy điểm mấu chốt cần lưu ý là: P' tuy

có dạng giống như P, nhưng theo một nghĩa nào đó, nó phải “nhỏ” hơn P, dễ giải hơn P và việc giải nó không cần dùng đến P.

Định nghĩa một hàm đệ quy hay thủ tục đệ quy gồm hai phần:

- Phần neo (anchor): Phần này được thực hiện khi mà công việc quá đơn giản, có thể giải trực tiếp chứ không cần phải nhờ đến một bài toán con nào cả.
- Phần đệ quy: Trong trường hợp bài toán chưa thể giải được bằng phần neo, xác định những bài toán con và gọi đệ quy giải những bài toán con đó. Khi đã có lời giải của những bài toán con rồi thì phối hợp chúng lại để giải bài toán đang quan tâm.

Phần đệ quy thể hiện tính “quy nạp” của lời giải. Phần neo cũng rất quan trọng bởi nó quyết định tới tính hữu hạn dừng của lời giải.

2.4.1.3. Một số bài toán giải thuật đệ quy

Bài toán 1: Bài toán tính giai thừa:

Cho n là một số tự nhiên ($n \geq 0$). Hãy tính giai thừa của n ($n!$) biết rằng $0! = 1$ và $n! = (n-1)! \times n$.

Phân tích : Theo giả thiết, có : $n! = (n-1)! \times n$. Như vậy :

- Để tính $n!$ cần phải tính $(n-1)!$
- Để tính $(n-1)!$ phải tính $(n-2)!$
- ...

Cứ như vậy, cho tới khi gặp trường hợp $0!$. Khi đó, lập tức có được kết quả là 1, không cần phải tính thông qua một kết quả trung gian khác.

Bài toán 2: Dãy Fibonacci:

Dãy Fibonacci là dãy vô hạn các số tự nhiên. Số Fibonacci thứ n , ký hiệu $F(n)$, được định nghĩa như sau :

- $F(n) = 1$, nếu $n=1$ hoặc $n=2$
- $F(n) = F(n-1) + F(n-2)$, nếu $n \geq 3$

Yêu cầu : Tính số Fibonacci thứ n với n cho trước.

Phân tích : Theo giả thiết:

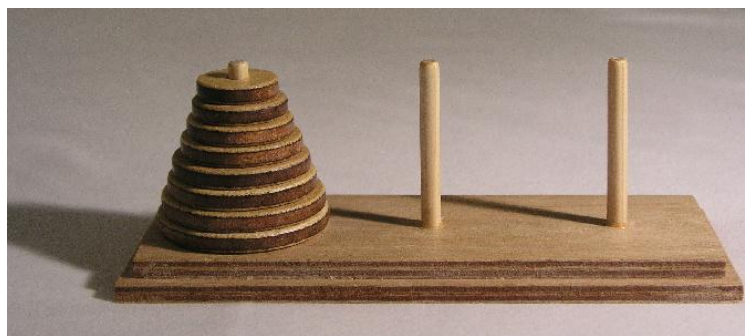
- Với $n < 3$: Suy ra ngay $F(n)=1$

- Với $n \geq 3$:
 - Để tính $F(n)$ phải tính $F(n-1)$ và $F(n-2)$.
 - Để tính $F(n-1)$ lại phải tính $F(n-2)$ và $F(n-3)$, và để tính $F(n-2)$ phải tính $F(n-3)$ và $F(n-4)$.
 - ...

Bài toán 3: Tháp Hà Nội

Đây là một bài toán rất nổi tiếng và kinh điển, rất thích hợp để minh họa cho thuật toán đệ quy. Sau đây là nội dung bài toán :

Có 3 chiếc cọc được đánh dấu lần lượt là A, B, C và n chiếc đĩa. Các đĩa này có kích thước khác nhau và mỗi đĩa đều có một lỗ ở giữa để cắm vào cọc. Ban đầu, các đĩa đều nằm ở cọc A, trong đó, đĩa nhỏ luôn nằm trên đĩa lớn hơn.



Hình 2-5. Hình minh họa về bài toán tháp Hà Nội

Yêu cầu : Chuyển n đĩa từ cọc A sang cọc đích C với các điều kiện sau :

- Mỗi lần chỉ chuyển được 1 đĩa.
- Trong quá trình chuyển, đĩa nhỏ phải luôn nằm trên đĩa lớn hơn.
- Cho phép sử dụng cọc B làm cọc trung gian.

Phân tích : Xét các trường hợp của n :

- Trường hợp đơn giản nhất, $n=1$, chỉ cần chuyển 1 đĩa từ cọc A sang cọc C.
- Nhiều hơn một chút, $n=2$, chuyển đĩa nhỏ nhất sang cọc B, chuyển đĩa còn lại sang cọc C, và cuối cùng chuyển đĩa nhỏ ở cọc B sang cọc C.
- Bây giờ ta xét n đĩa ($n > 2$). Giả sử đã có cách chuyển $n-1$ đĩa từ một cọc sang một cọc khác. Như vậy, để chuyển n đĩa từ cọc nguồn sang cọc đích, cần chuyển $n-1$ đĩa từ cọc nguồn sang cọc trung gian. Sau đó

chuyển đĩa lớn nhất từ cọc nguồn sang cọc đích. Cuối cùng, chuyển $n-1$ từ cọc trung gian về cọc đích.

2.4.1.4. **Hiệu lực của đệ quy**

Qua các bài toán trên, có thể thấy đệ quy là một công cụ mạnh mẽ để giải các bài toán. Có những bài toán mà bên cạnh giải thuật đệ quy vẫn có những giải thuật lặp khá đơn giản và hữu hiệu. Chẳng hạn bài toán tính giai thừa hay tính số Fibonacci. Tuy vậy, đệ quy vẫn có vai trò xứng đáng của nó, có nhiều bài toán mà thiết kế giải thuật đệ quy đơn giản hơn nhiều so với lời giải lặp và trong một số trường hợp chương trình đệ quy hoạt động nhanh hơn chương trình viết không đệ quy. Giải thuật cho bài toán Tháp Hà Nội là một ví dụ.

Có một mối quan hệ khăng khít giữa đệ quy và quy nạp toán học. Cách giải đệ quy cho một bài toán dựa trên việc định rõ lời giải cho trường hợp suy biến (neo) rồi thiết kế làm sao để lời giải của bài toán được suy ra từ lời giải của bài toán nhỏ hơn cùng thể loại như thế. Tương tự như vậy, quy nạp toán học chứng minh một tính chất nào đó ứng với số tự nhiên cũng bằng cách chứng minh tính chất đó đúng với một số trường hợp cơ sở (thường người ta chứng minh nó đúng với 0 hay đúng với 1) và sau đó chứng minh tính chất đó sẽ đúng với n bất kỳ nếu nó đã đúng với mọi trường hợp nhỏ hơn n . Do đó, quy nạp toán học được dùng để chứng minh các tính chất có liên quan tới giải thuật đệ quy.

Mọi giải thuật đệ quy đều có các thay thế bằng một giải thuật không đệ quy (khử đệ quy), có thể nói được như vậy bởi tất cả các chương trình con đệ quy sẽ đều được trình dịch chuyển thành những mã lệnh không đệ quy trước khi giao cho máy tính thực hiện.

2.5. **Giới thiệu ngôn ngữ lập trình Visual Basic**

Trong giới hạn đề tài sử dụng ngôn ngữ lập trình Visual Basic (VB) và ngôn ngữ Python để xây dựng các modul của phần mềm.

Visual Basic được giới thiệu lần đầu tiên vào năm 1991, tiền thân là ngôn ngữ lập trình Basic trên HĐH DOS. Tuy nhiên, lúc bấy giờ Visual Basic chưa được nhiều người người tiếp nhận. Mãi cho đến năm 1992, khi phiên bản 3.0 ra đời với rất nhiều

cải tiến so với các phiên bản trước đó, VB mới thật sự trở thành một trong những công cụ chính để phát triển các ứng dụng trên Windows.

Các phiên bản sau đó của VB, như phiên bản 4.0 ra đời năm 1995, phiên bản 5.0 ra đời năm 1996 và gần đây nhất là phiên bản 6.0 ra đời năm 1998 với các tính năng ngày càng được nâng cao đã khiến mọi người công nhận VB hiện là một trong những công cụ chính để phát triển các ứng dụng trên Windows.

Visual Basic 6.0 cho phép người dùng tiếp cận nhanh cách thức lập trình trên môi trường Windows.

- Ưu điểm:
 - Tiết kiệm được thời gian và công sức so với một số ngôn ngữ lập trình có cấu trúc khác vì bạn có thể thiết lập các hoạt động trên từng đối tượng được VB cung cấp.
 - Khi thiết kế chương trình có thể thấy ngay kết quả qua từng thao tác và giao diện khi thi hành chương trình.
 - Cho phép chỉnh sửa dễ dàng, đơn giản.
 - Làm việc với các điều khiển mới (ngày tháng với điều khiển MonthView và DateTimePicker, các thanh công cụ có thể di chuyển được CoolBar, sử dụng đồ họa với ImageCombo, thanh cuộn FlatScrollBar,...).
 - Làm việc với cơ sở dữ liệu.
 - Các bổ sung về lập trình hướng đối tượng.
 - Khả năng kết hợp với các thư viện liên kết động DLL.
- Nhược điểm:
 - Yêu cầu cấu hình máy khá cao.
 - Chỉ chạy được trên môi trường Win95 trở lên.
- Một số hàm, phương thức trong VBA
 - Hàm Split: Được sử dụng để tách một chuỗi thành các chuỗi con bằng một kí tự nhất định.
 - OpenFileDialog: Mở ra một Form cho phép lấy đường dẫn đến một File nhất định.
 - SaveFileDialog: Mở ra một Form cho phép lưu một file với một đường dẫn được chọn.

- Sử dụng cấu trúc vòng lặp For, If.
- Thủ tục đọc File với IO.
- Replace: Thay thế một chuỗi có sẵn bằng một chuỗi mới.
- Length: Trả về giá trị độ dài của một chuỗi.
- ReferenceEquals: So sánh hai giá trị, nếu giống nhau thì trả về True nếu khác nhau thì trả về False.

2.6. Giới thiệu ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình thông dịch do Guido Van Rossum tạo ra năm 1990, Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian nó đã phát triển nhanh chóng sang mọi hệ điều hành từ MS-DOS đến Mac OS, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido Van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

• Các phiên bản của Python

Python 1: Bao gồm các bản phát hành 1.x. Giai đoạn này, kéo dài từ đầu đến cuối thập niên 1990. Từ năm 1990 đến 1995, Guido làm việc tại CWI (Centrum voor Wiskunde en Informatica - Trung tâm Toán-Tin học tại Amsterdam, Hà Lan). Vì vậy, các phiên bản Python đầu tiên đều do CWI phát hành. Phiên bản cuối cùng phát hành tại CWI là 1.2.

Vào năm 1995, Guido chuyển sang CNRI (Corporation for National Research Initiatives) ở Reston, Virginia. Tại đây, ông phát hành một số phiên bản khác. Python 1.6 là phiên bản cuối cùng phát hành tại CNRI.

Sau bản phát hành 1.6, Guido rời bỏ CNRI để làm việc với các lập trình viên chuyên viết phần mềm thương mại. Tại đây, ông có ý tưởng sử dụng Python với các phần mềm tuân theo chuẩn GPL. Sau đó, CNRI và FSF (Free Software Foundation - Tổ chức phần mềm tự do) đã cùng nhau hợp tác để làm bản quyền Python phù hợp với

GPL. Cùng năm đó, Guido được nhận Giải thưởng FSF vì sự phát triển phần mềm tự do (Award for the Advancement of Free Software).

Phiên bản 1.6.1 ra đời sau đó là phiên bản đầu tiên tuân theo bản quyền GPL. Tuy nhiên, bản này hoàn toàn giống bản 1.6, trừ một số sửa lỗi cần thiết.

Python 2: vào năm 2000, Guido Van Rossum và nhóm phát triển Python dời đến BeOpen.com và thành lập BeOpen PythonLabs. Phiên bản Python 2.0 được phát hành tại đây. Sau khi phát hành Python 2.0, Guido và các thành viên PythonLabs gia nhập Digital Creations.

Python 2.1 ra đời kế thừa từ Python 1.6.1 và Python 2.0. Bản quyền của phiên bản này được đổi thành Python Software Foundation License. Từ thời điểm này trở đi, Python thuộc sở hữu của Python Software Foundation (PSF), một tổ chức phi lợi nhuận được thành lập theo mẫu Apache Software Foundation.

Python 3: Còn gọi là Python 3000 hoặc Py3K: Dòng 3.x sẽ không hoàn toàn tương thích với dòng 2.x, tuy vậy có công cụ hỗ trợ chuyển đổi từ các phiên bản 2.x sang 3.x. Nguyên tắc chủ đạo để phát triển Python 3.x là "bỏ cách làm việc cũ nhằm hạn chế trùng lặp về mặt chức năng của Python". Trong PEP có mô tả chi tiết các thay đổi trong Python.

- **Khả năng mở rộng**

Python có thể được mở rộng: Nếu ta biết sử dụng ngôn ngữ C, ta có thể dễ dàng viết và tích hợp vào Python nhiều hàm tùy theo nhu cầu. Các hàm này sẽ trở thành hàm xây dựng sẵn (*built-in*) của Python. Ta cũng có thể mở rộng chức năng của trình thông dịch, hoặc liên kết các chương trình Python với các thư viện chỉ ở dạng nhị phân (như các thư viện đồ họa do nhà sản xuất thiết bị cung cấp). Hơn thế nữa, cũng có thể liên kết trình thông dịch của Python với các ứng dụng viết từ C và sử dụng nó như là một mở rộng hoặc một ngôn ngữ dòng lệnh phụ trợ cho ứng dụng đó.

Arcpy là một gói (phần mở rộng) được xây dựng dựa trên mô đun *arcgisscripting* của ESRI trước đây. Với Arcpy sẽ mang lại sự tiện lợi và hiệu quả trong việc phân tích dữ liệu địa lý, chuyển đổi dữ liệu, quản lý dữ liệu và tự động hóa bản đồ với Python.

Lợi thế của việc sử dụng Arcpy trong Python là vì Python là một ngôn ngữ lập trình mục đích. Nó dễ hiểu, rõ ràng, tương tác tốt và xây dựng nhanh chóng các Scripts

cho một công cụ hay một ứng dụng lớn trong ArcGis. Vì vậy, Arcpy mang lại tiện lợi cho việc phát triển mô đun giải quyết các chuỗi xử lý phức tạp trong Gis.

- **Trình thông dịch**

Python là một ngôn ngữ lập trình dạng thông dịch, do đó có ưu điểm tiết kiệm thời gian phát triển ứng dụng vì không cần phải thực hiện biên dịch và liên kết. Trình thông dịch có thể được sử dụng để chạy file script, hoặc cũng có thể được sử dụng theo cách tương tác. Ở chế độ tương tác, trình thông dịch Python tương tự shell của các hệ điều hành họ Unix. Tại đó, ta có thể nhập vào từng biểu thức rồi gõ Enter, và kết quả thực thi sẽ được hiển thị ngay lập tức. Đặc điểm này rất hữu ích cho người mới học, giúp họ nghiên cứu tính năng của ngôn ngữ; hoặc để các lập trình viên chạy thử mã lệnh trong suốt quá trình phát triển phần mềm. Ngoài ra, cũng có thể tận dụng đặc điểm này để thực hiện các phép tính như với máy tính bỏ túi.

- **Module**

Python cho phép chia chương trình thành các module để có thể sử dụng lại trong các chương trình khác. Nó cũng cung cấp sẵn một tập hợp các modules chuẩn mà lập trình viên có thể sử dụng lại trong chương trình của họ. Các module này cung cấp nhiều chức năng hữu ích, như các hàm truy xuất tập tin, các lời gọi hệ thống, trợ giúp lập trình mạng (socket),...

- **Lệnh và cấu trúc điều khiển**

Mỗi câu lệnh trong Python nằm trên một dòng mã nguồn. Ta không cần phải kết thúc câu lệnh bằng bất kì kí tự gì. Cũng như các ngôn ngữ khác, Python cũng có các cấu trúc điều khiển. Chúng bao gồm:

- Cấu trúc rẽ nhánh: Cấu trúc `if` (có thể sử dụng thêm `elif` hoặc `else`), dùng để thực thi có điều kiện một khối mã cụ thể.
- Cấu trúc lặp, bao gồm:
 - Lệnh `while`: chạy một khối mã cụ thể cho đến khi điều kiện lặp có giá trị `false`.
 - Vòng lặp `for`: lặp qua từng phần tử của một dãy, mỗi phần tử sẽ được đưa vào biến cục bộ để sử dụng với khối mã trong vòng lặp.

- Python cũng có từ khóa `class` dùng để khai báo lớp (sử dụng trong lập trình hướng đối tượng) và lệnh `def` dùng để định nghĩa hàm.

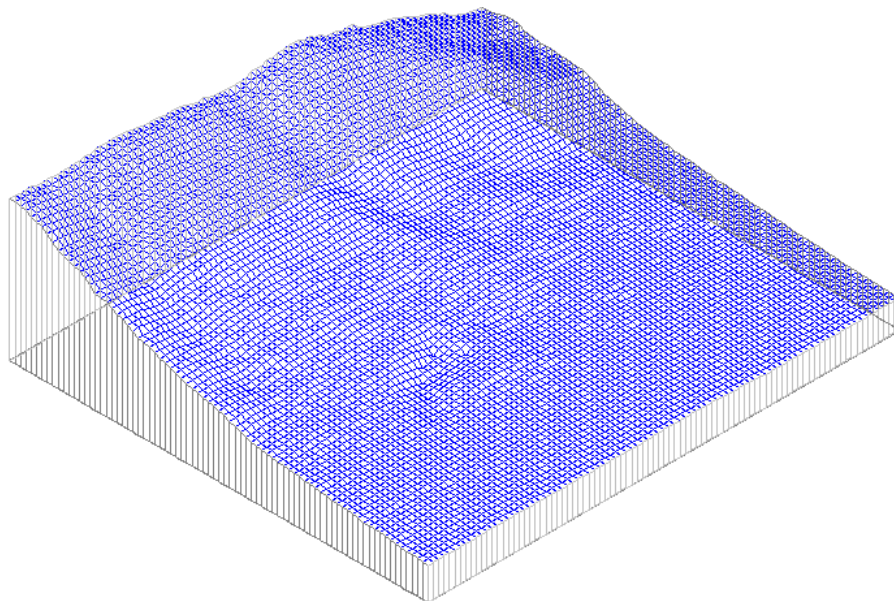
Với những ưu điểm của mình Python đã được tích hợp trong ArcGis để hỗ trợ việc xử lý, lưu trữ, khai thác và hiển thị dữ liệu, phiên bản ArcGis 9 được tích hợp Python 2.5 đến phiên bản ArcGis 10 được tích hợp Python 2.6.

2.7. Mô hình địa hình và cấu trúc cơ sở dữ liệu

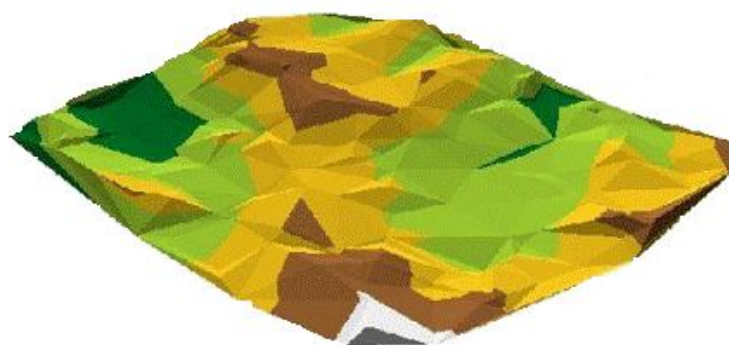
DTM là sự thể hiện dưới cấu trúc dạng số của những đặc trưng địa hình bao gồm cao độ, độ dốc, mặt dốc, dòng chảy, những đặc tính địa hình khác và bao gồm cả các thực thể nhân tạo. DEM được đề cập đến tới như là mô hình với dữ liệu cao độ dạng số phân bố vị trí một cách ngẫu nhiên trong một khu vực có thể dùng để nội suy, trong khi đó DTM đề cập tới những đặc trưng địa hình ở dạng số được tạo ra từ dữ liệu cao độ (TS. Nguyễn Kim Lợi và ctv, 2009).

Mô hình số độ cao DEM:

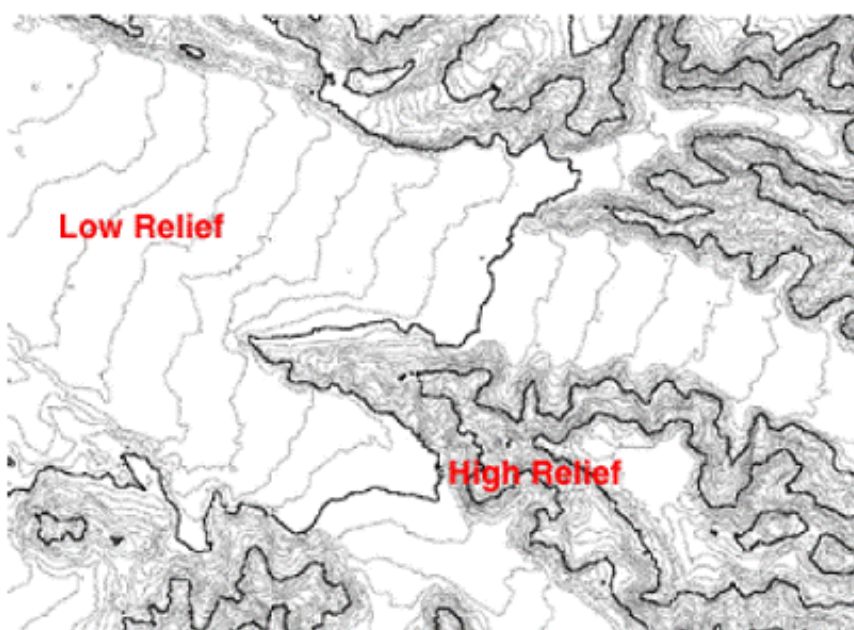
DEM là từ viết tắt của cụm từ tiếng anh Digital Elevation Model và có nghĩa là mô hình số độ cao. DEM chứa các giá trị độ cao mặt đất dựa trên hệ qui chiếu đo đạc. Có nhiều cấu trúc lưu trữ DEM trên máy tính trong đó có ba dạng chính là:



Hình 2-6. DEM cấu trúc dạng phân tử lưới đều Grid



Hình 2-7. DEM cấu trúc dạng TIN



Hình 2-8. Những đường đồng mức độ cao

- **Mạng lưới tam giác không đều TIN**

Mạng lưới tam giác không đều là một dạng của DEM với sự hình thành từ một mạng lưới các tam giác không đều đặt tại những điểm địa hình một cách ngẫu nhiên. Do vị trí của các điểm địa hình là ngẫu nhiên, không theo quy luật được đo từ khu vực địa hình bên ngoài thực tế nên bề mặt địa hình sẽ ít trơn hơn. Các tam giác trong mạng lưới TIN được tạo thành từ sự kết hợp của những điểm độ cao theo tiêu chuẩn Delaunay. Tiêu chuẩn Delaunay phát biểu như sau: Qua ba điểm bất kỳ đường tròn ngoại tiếp của chúng không chứa bất cứ một điểm nào khác. Các tam giác được hình thành dựa trên tiêu chuẩn Delaunay gọi là tam giác Delaunay.

- **Sự tạo thành đường đồng mức**

Những đường đồng mức là một trong những đặc trưng địa hình thể hiện đáng đất và địa hình của khu vực như là một dạng độ cao. Có 2 dạng đường đồng mức trong việc thể hiện dữ liệu trực quan GIS: dạng vector với các đường và dạng ảnh raster.

Các cách thể hiện dữ liệu DEM dạng ASCII là: Dạng Grid là một mạng lưới các giá trị độ cao, ngoài ra còn có các giá trị số hàng, số cột, kích thước của ô pixel tọa độ điểm góc trái bên dưới của DEM và nó được sử dụng nhiều trong phần mềm ArcGis (như Hình 2-9). Dạng XYZ chứa một dãy các giá trị độ cao với tọa độ xác định ở cột thứ nhất và cột thứ hai, ở cột thứ 3 là giá trị độ cao của điểm (như Hình 2-10).

```
ncols          43
nrows          38
xllcorner      106.78291972546
yllcorner      10.866678949909
cellsize       0.00027766029143888
NODATA_value   -9999
78 72 69 71 58 49
74 67 56 49 46 50
69 53 44 37 38 48
64 58 55 22 31 24
68 61 47 21 16 19
74 53 34 12 11 12|
```

Hình 2-9. DEM thể hiện dạng Grid

```
head -n 4 VTL2733.XYZ
617000.0 148000.0 157.92
617025.0 148000.0 157.82
617050.0 148000.0 157.70
617075.0 148000.0 157.60|
```

Hình 2-10. DEM thể hiện dạng XYZ

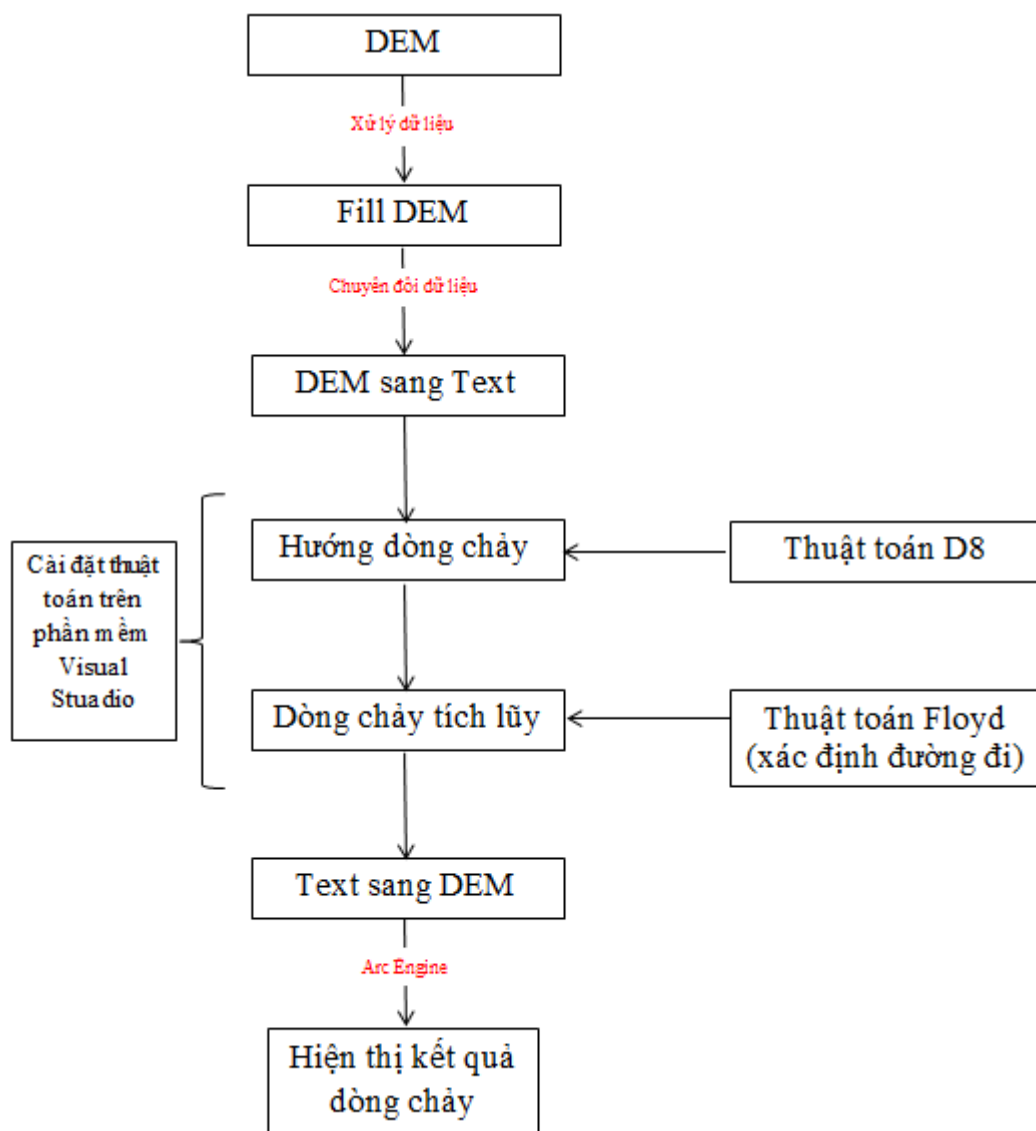
CHƯƠNG 3. PHƯƠNG PHÁP THỰC HIỆN

3.1. Lược đồ phương pháp thực hiện

Các phương pháp tính toán dòng chảy tích lũy hay xác định dòng chảy trên địa hình đều dựa trên cơ sở hướng dòng chảy tại mỗi điểm từ đó liên kết các hướng này với nhau để tạo thành dòng chảy. Hiện tại có rất nhiều thuật toán để xác định hướng dòng chảy, các thuật toán về sau đều có những cải tiến để bổ sung những thiếu sót của thuật toán trước đây. Tất cả các thuật toán xác định hướng dòng chảy đều sử dụng dữ liệu đầu vào là DEM để cho ra các kết quả tùy thuộc vào mục đích sử dụng mà cơ bản nhất là để xác định dòng chảy.

Trên cơ sở thuật toán D8 và kết hợp với thuật toán Floyd để cài đặt thuật toán D8 trên ngôn ngữ lập trình VB, kết quả là xây dựng được phần mềm với khả năng sử dụng dữ liệu đầu vào là DEM kèm với các gói hỗ trợ từ Arcpy để đưa ra kết quả là dòng chảy trên địa hình tại khu vực nghiên cứu.

Thuật toán Floyd được ứng dụng để cài đặt thuật toán D8 nhờ vào khả năng xác định đường đi từ một điểm đến một điểm bất kì trong đồ thị gồm các điểm và các cạnh, trong DEM thì đồ thị tương ứng là ma trận các giá trị độ cao, từ đó có thể xác định được dòng chảy xuất phát từ một điểm đến điểm tích lũy dòng chảy. Do đồ thị (ma trận độ cao) các đỉnh được sử dụng có những đặc điểm riêng nên mục đích của việc sử dụng thuật toán Floyd không phải là xác định con đường ngắn nhất mà là xác định có dòng chảy hay không và dòng chảy đó không phải dòng chảy ngắn nhất. Cụ thể như sơ đồ Hình 3-1.



Hình 3-1. Sơ đồ thực hiện phần mềm

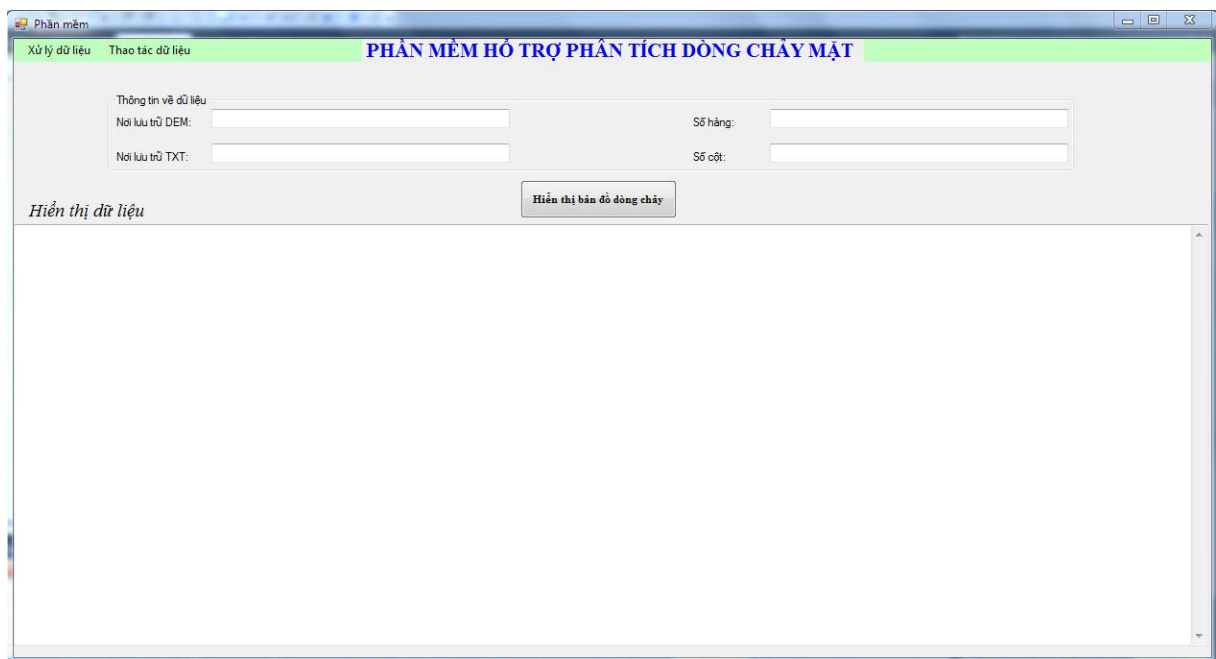
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1. Kết quả thực thi chương trình

4.1.1. Các module trong phần mềm

Giao diện người dùng được xây dựng bằng phần mềm Visual Studio với ngôn ngữ lập trình là Visual Basic. Phần mềm gồm 2 Form chính, Form1 là giao diện chính gồm Menu, Label, các nút Button, TextBox, Groupbox. Menu **Xử lý dữ liệu** gồm **FillDEM**, **DEM sang TXT**, **TXT sang DEM**. Menu **Thao tác dữ liệu** gồm **Đọc dữ liệu**, **Hướng dòng chảy**, **Dòng chảy tích lũy** và **Lưu dữ liệu**. Trong Form chính còn có 1 TextBox nhằm hiển thị các dữ liệu được đọc và tính toán ra (Hình 4-1).

Form thứ 2 là Form được tích hợp bộ thư viện Arc Engine để hiển thị kết quả sau khi thuật toán thực hiện xong.



Hình 4-1. Giao diện người dùng

Với dữ liệu DEM của khu vực trường Đại học Nông Lâm, sử dụng ngôn ngữ lập trình Python với gói Arcpy cho phép thực hiện được nhiều Tool trong ArcGis. Dữ liệu DEM ban đầu còn có nhiều điểm lỗi, lồi lõm không mong muốn nên cần phải chỉnh sửa lại các điểm này bằng công cụ Fill. Arcpy cung cấp một bộ thư viện các hàm cho Python. Trong Python, import gói Arcpy vào và gọi đến hàm Fill để truyền các tham

số cần thiết cho hàm Fill như: in_surface_raster là dữ liệu đầu vào DEM dạng Raster, out_surface_raster là nơi lưu trữ dữ liệu đầu ra. Click vào **Fill DEM** sẽ xuất hiện một cửa sổ openfiledialog để chọn đường dẫn đến DEM cần Fill tức là in_surface_raster, sau khi chọn xong sẽ xuất hiện thêm 1 cửa sổ savefiledialog để chọn nơi lưu trữ DEM sau khi Fill chạy xong tức là out_surface_raster trong hàm Fill (**Phụ lục 1**).

Sau khi chỉnh sửa các lỗi trên DEM bằng **Fill DEM**, thuật toán được cài đặt trong phần mềm sử dụng dữ liệu độ cao dạng Grid và định dạng là Text để có thể đọc được ma trận độ cao. Chính vì vậy, cần phải chuyển đổi dữ liệu dạng DEM đã được sửa lỗi bằng **Fill DEM** sang dạng Text. Để thực hiện điều này, trong Python sử dụng đến hàm rasterToASCII_conversion với các thông số là in_raster (dữ liệu dạng Raster), tham chiếu này được chuyển vào hàm thông qua một openfiledialog, out_ascii_file (nơi và định dạng file Text lưu trữ) được truyền vào hàm thông qua một savefiledialog (**Phụ lục 2**).

Định dạng Text của dữ liệu DEM thì sẽ cung cấp thông số: số cột (ncols là 43), số hàng (nrows là 38), tọa độ điểm ở góc dưới bên trái của DEM được hiện thị trên phần mềm ArcGis (xllcorner là 106.78291972546, yllcorner là 10.866678949909), kích thước của một ô pixel (cellsize là 0.00027766029143888), các ô không có giá trị độ cao (NODATA_value) được gán giá trị là -9999 trong file Text (như Hình 4-2).

```
ncols      43
nrows     38
xllcorner  106.78291972546
yllcorner  10.866678949909
cellsize   0.00027766029143888
NODATA_value -9999
12 8 10 13 18 17 16 19 23 21 16 16 19 21 25 26 23 22 22 19 24 29 27 24 17 17 14 15 19 21 27 25 24 21 22 27 24 20 20 23 26 24 22
13 8 9 11 16 17 16 19 18 15 14 15 16 16 18 21 21 20 20 26 27 27 24 22 21 19 17 16 18 17 22 22 22 23 24 26 25 21 25 30 29 27 22
13 10 10 10 15 14 12 12 13 15 12 12 14 15 15 16 18 18 19 23 27 26 22 20 21 20 17 17 19 17 18 20 20 23 25 24 26 28 29 27 24 23
14 12 11 10 9 13 12 11 11 14 11 12 15 16 15 14 15 14 14 18 22 24 22 21 21 22 23 22 20 20 18 17 20 22 24 24 26 29 31 33 27 24 25
11 11 9 8 9 13 12 11 11 11 11 14 15 15 18 15 14 15 14 18 20 21 20 21 21 20 20 19 18 20 19 20 22 22 19 21 21 23 29 33 32 27 26
12 12 8 13 12 12 12 11 11 11 13 17 20 15 14 14 14 14 16 16 17 18 23 20 21 19 18 17 18 22 22 21 19 19 22 26 27 25 27 27 30 25
13 11 9 8 11 11 11 11 12 11 13 16 16 13 18 18 16 16 19 16 15 15 15 18 19 17 17 17 17 18 21 23 19 17 17 21 22 27 20 22 24 27 27
13 11 8 11 12 12 13 14 13 11 12 15 12 14 15 16 15 18 18 15 15 15 16 17 17 17 17 17 17 19 19 17 17 19 22 20 20 23 26 26
12 9 8 8 9 14 15 16 14 13 11 13 11 12 13 13 16 15 15 15 15 15 16 17 17 17 17 17 17 17 17 18 19 20 20 20 20 22 24
9 10 8 8 10 14 17 17 16 11 11 15 14 15 16 15 15 15 15 15 15 17 17 17 17 17 17 17 17 18 19 20 20 20 20 21 22
8 8 8 8 12 15 13 15 18 18 13 12 14 13 13 14 16 15 15 15 15 15 17 18 19 18 17 17 18 21 21 20 19 20 22 25 23 27 20 21 20 26
9 9 8 12 14 15 16 16 19 18 14 12 13 16 13 16 15 15 15 15 16 15 15 17 21 25 22 19 20 22 25 27 29 26 25 20 22 25 29 24 20 20 23
10 10 9 11 11 12 13 15 15 14 13 12 12 13 14 15 18 15 15 16 15 15 16 24 29 35 26 19 19 19 21 27 27 30 26 30 29 27 32 28 23 20 21
8 8 8 12 11 11 11 16 12 12 12 13 12 17 14 16 15 18 17 16 19 23 30 34 35 29 20 19 19 21 20 27 32 27 31 32 30 32 32 25 24 22
7 8 9 10 15 11 12 12 13 12 12 12 13 14 15 14 14 15 19 21 21 22 20 32 32 35 25 23 19 19 19 20 24 27 28 31 29 32 32 28 30 28
12 11 15 18 17 16 12 13 12 16 12 12 12 14 14 14 14 15 19 17 17 20 28 31 29 23 20 20 21 20 21 23 26 31 31 33 27 31 34 33 32 30
12 9 14 15 17 14 17 15 15 12 12 13 15 17 14 16 15 17 22 23 24 26 24 22 21 20 23 25 22 23 24 25 30 32 28 27 27 34 35 30 28
9 12 10 12 10 10 14 13 14 13 15 12 16 15 16 15 16 16 20 18 17 27 24 24 21 20 20 20 25 30 22 22 23 25 31 31 26 24 24 29 30 27 28
8 9 11 10 10 10 10 10 11 11 12 16 19 15 15 16 19 16 20 24 20 28 27 23 20 21 20 22 25 23 25 23 26 29 32 34 29 26 25 24 24 23
9 10 12 10 10 10 10 10 11 12 13 15 17 16 19 16 16 22 20 21 24 28 28 28 31 37 33 33 31 27 26 27 32 33 39 31 25 24 24 24 27
9 10 10 10 12 11 11 10 12 11 11 15 17 19 23 23 27 24 25 28 28 28 28 32 34 36 34 36 35 30 26 26 30 33 36 26 24 24 26 31
12 11 11 14 16 15 13 13 16 14 11 13 15 20 21 24 25 29 28 33 30 28 28 28 29 32 34 35 32 40 36 33 28 29 35 37 38 29 26 24 25 28 29
12 11 14 15 14 15 13 13 13 12 11 14 14 16 19 16 23 26 34 28 28 28 28 28 33 33 32 32 36 36 36 32 34 33 34 30 28 25 24 26 29 30
10 12 11 12 13 14 13 13 13 14 15 13 15 16 16 17 21 27 30 28 28 28 28 31 40 35 35 32 32 38 40 34 30 30 33 35 29 26 26 24 29 27 26
7 11 13 13 12 14 14 17 17 19 22 15 15 20 18 18 18 24 31 30 28 28 30 36 38 44 40 37 32 32 37 30 28 30 34 34 26 25 25 28 28 30 26
15 13 19 17 14 13 14 17 17 28 21 23 28 22 22 21 19 26 39 34 36 31 29 35 41 41 36 38 36 32 31 26 25 29 30 29 26 30 32 31 29 29 25
19 15 15 15 15 16 17 20 22 23 27 24 28 30 21 19 23 32 37 35 29 37 41 37 31 32 33 30 27 29 22 24 27 26 26 28 34 35 33 29 29 28
19 17 17 15 17 16 16 21 24 24 25 25 30 28 32 19 19 27 35 38 35 35 39 31 28 22 20 17 24 22 20 24 27 28 32 27 29 36 35 28 26 23
20 15 15 15 20 19 20 25 23 23 24 25 31 29 26 26 27 33 34 39 41 34 30 20 16 16 16 16 19 20 21 24 29 34 30 28 23 26 28 26 23 25
13 17 15 18 26 22 20 20 21 21 21 25 25 25 25 28 30 35 34 33 21 16 16 16 16 16 21 24 28 29 30 29 24 23 24 24 24 25 24
13 15 19 20 21 23 20 21 22 24 22 21 21 21 25 28 25 27 20 28 30 33 29 22 17 16 16 16 16 18 21 21 24 25 30 31 24 22 22 22 22 21 22
15 19 18 20 18 19 18 19 21 23 26 24 22 22 25 23 22 25 26 26 28 28 29 22 20 16 16 16 17 16 19 22 21 23 26 23 22 22 22 20 20 20
14 16 17 17 18 18 18 19 29 28 32 27 24 24 23 20 18 22 22 20 21 23 21 19 18 16 16 16 16 16 18 23 23 23 22 25 26 28 24 22 21 21
15 15 14 16 18 16 18 23 31 32 33 29 25 21 18 17 17 22 18 17 20 19 17 16 16 16 16 16 16 18 22 21 23 22 26 30 35 35 33 28 23 22
17 14 16 17 18 16 16 20 28 31 31 29 24 25 18 18 17 17 16 16 16 16 16 16 17 18 18 16 16 16 17 20 21 21 27 32 36 39 35 30 26 27
15 14 16 18 18 15 22 18 22 31 27 25 25 22 19 20 18 17 20 18 18 18 19 17 16 17 20 22 19 16 16 16 16 20 19 23 26 29 29 32 35 30 27 28
9 9 10 10 12 12 14 17 15 18 19 18 21 23 21 20 19 18 21 19 20 21 18 16 16 21 21 22 16 17 18 19 22 22 21 22 27 29 32 31 28
10 11 11 11 12 13 13 14 15 18 18 17 17 18 17 15 16 15 19 20 20 21 17 15 16 18 21 22 22 20 16 17 21 16 23 23 19 28 27 30 32 28
```

Hình 4-2. Dữ liệu DEM trong file Text

Trong Visual Studio, sử dụng phương thức đọc file `IO.File.ReadAllLines()` với đường dẫn đến file Text vừa tạo. Với phương thức này sẽ đọc tất cả các dòng trong file Text, mỗi dòng sẽ được gán vào một mảng do mình khai báo. Để lấy được các giá trị số hàng, số cột sử dụng hàm Mid (cắt một chuỗi từ một vị trí bắt đầu xác định và cắt số ký tự xác định).

Riêng với ma trận độ cao cần phải được gán vào một mảng hai chiều, mảng này được khởi tạo từ đầu và tất cả các modul đều có thể gọi được nó. Chính vì vậy, trước khi đưa ma trận độ cao vào mảng thì cần phải dùng hàm Redim (định lại kích thước của một mảng) để định lại kích thước của mảng với số hàng và số cột. Việc lấy ra số hàng, số cột và ma trận độ cao được viết trong nút **Đọc dữ liệu**. **Đọc dữ liệu** sẽ cung cấp một openFileDialog để người sử dụng có thể chọn đường dẫn đến file Text cần tính toán, khi chọn đường dẫn xong các thông tin về số hàng và số cột được hiển thị trong TextBox trên GroupBox thông tin dữ liệu, ma trận độ cao sẽ được hiển thị tại TextBox **Hiển thị dữ liệu** (**Phụ lục 3**).

Cài đặt thuật toán D8 và tính dòng chảy tích lũy được viết trong modul **Hướng dòng chảy** và **Dòng chảy tích lũy**, cách thực hiện sẽ được trình bày ở phần thuật toán D8 và thuật toán xác định dòng chảy tích lũy.

Sau khi tính toán dòng chảy tích lũy xong, cần lưu dữ liệu vào file Text để dễ dàng chuyển đổi định dạng dữ liệu sau này. Việc lưu dữ liệu được thực hiện trong **Lưu dữ liệu**. Click vào **Lưu dữ liệu** thì 1 cửa sổ savefiledialog hiện ra để chọn đường dẫn đến nơi lưu trữ file Text (**Phụ lục 4**).

Khi việc tính hướng dòng chảy và dòng chảy tích lũy được tính xong thì kết quả sẽ được trả về một file định dạng Text với các thông tin số hàng, số cột, tọa độ điểm góc trái bên dưới, kích thước của ô pixel và ma trận các giá trị dòng chảy tích lũy bên trong file Text này. Sau đó, được chuyển sang dạng DEM và được hiển thị trên formarcengine.

Import bộ thư viện arcpy vào Python, gọi đến hàm ASCIIToRaster_conversion và truyền các tham số inASCII, outRaster vào để xử lý. Với inASCII là dữ liệu đầu vào có định dạng Text (.txt), outRaster là nơi lưu dữ liệu sau khi chuyển đổi xong, khi click vào **TXT sang DEM** một cửa sổ openFileDialog hiện ra để có thể chọn đường dẫn

4.1.3. Thuật toán xác định dòng chảy tích lũy

Sự tích lũy dòng chảy cho một ô nào đó trong khu vực trên nền mô hình DEM được xác định bằng cách tính tổng số ô lưới tập trung nước về ô đó theo hướng dòng chảy. Bản đồ tích lũy dòng chảy cũng chính là bản đồ sông suối của khu vực trong điều kiện tự nhiên.

Ứng dụng thuật toán Floyd (tìm đường đi ngắn nhất) để cài đặt thuật toán tính dòng chảy tích lũy trong phần mềm. Thuật toán Floyd được sử dụng để xác định khoảng cách (độ dài) từ một điểm đến một điểm bất kì trong một đồ thị có hướng với n đỉnh và m cạnh bằng cách lấy giá trị nhỏ nhất trong các trường hợp độ dài đường đi giữa hai điểm đi qua các điểm trung gian.

Ma trận các giá trị hướng dòng chảy được tính bởi thuật toán D8 chính là đồ thị có tương ứng trong thuật toán Floyd (hướng được thể hiện qua các giá trị 1, 2, 4, 8, 16, 32, 64, 128), hướng dòng chảy tích lũy được xác định là hướng của một điểm (A) đến một điểm lân cận (B), nếu điểm B có hướng dòng chảy đến một điểm lân cận khác (C), thì có nghĩa là sẽ có một dòng chảy từ A đến C thông qua điểm trung gian là B, tức là C sẽ tích lũy dòng chảy của A đến B và vào C. Dựa trên cơ sở của thuật toán Floyd, sẽ xác định được liệu có đường đi nào từ A đến C không và thông qua điểm điểm trung gian nào. Trong thuật toán Floyd, nó sẽ xét hai trường hợp độ dài giữa hai điểm là độ dài đường đi trực tiếp và độ dài đường đi qua điểm trung gian và từ đó lấy ra giá trị nhỏ nhất, nhưng do đồ thị hướng dòng chảy có đặc trưng về hướng là chỉ chảy đến một điểm lân cận nên sẽ chỉ có một trường hợp là độ dài đường đi trung gian chảy từ A đến C là phải thông qua điểm trung gian B. Ứng dụng thuật toán Floyd trong tính dòng chảy tích lũy là xác định đường đi của dòng chảy tích lũy thông qua các điểm trung gian.

Với ma trận giá trị hướng dòng chảy (Q) có m hàng và n cột cần phải chuyển thành ma trận vuông (P) với kích thước là $\text{size} = m * n$. Ma trận P là ma trận mà mỗi phần tử thể hiện liệu có dòng chảy từ một điểm đến điểm lân cận không, nếu có thì gán giá trị là 1, nếu không có dòng chảy nào thì gán giá trị là $+\infty$ ($+\infty$ là giá trị lớn nhất nào đó lớn hơn tất cả các giá trị trong ma trận P), ví dụ như điểm có tọa độ (1, 1) có dòng chảy đến điểm (1, 2) tức là hướng của điểm (1, 1) là 1 thì trong ma trận (P) điểm có tọa độ là (1, 2) sẽ được gán bằng 1 và điểm (2, 1) được gán là 1. Để thành lập được ma

trận P cần phải xét tất cả các trường hợp dòng chảy từ một điểm đến tất cả các điểm còn lại trong ma trận Q dựa trên cơ sở hướng dòng chảy tại mỗi điểm đã được xác định (Phụ lục 7).

Sau khi đã có được ma trận P thì cần phải biết được dòng chảy từ điểm này đến điểm khác phải thông qua bao nhiêu điểm trung gian, bằng cách nhân hai ma trận P lại với nhau cho đến lúc nào đó ma trận mới này sẽ ổn định, tức là các giá trị trong ma trận không thay đổi. Khi đó, các giá trị trong ma trận mới chính là số điểm trung gian cần có để đi từ điểm này đến điểm kia (Phụ lục 8).

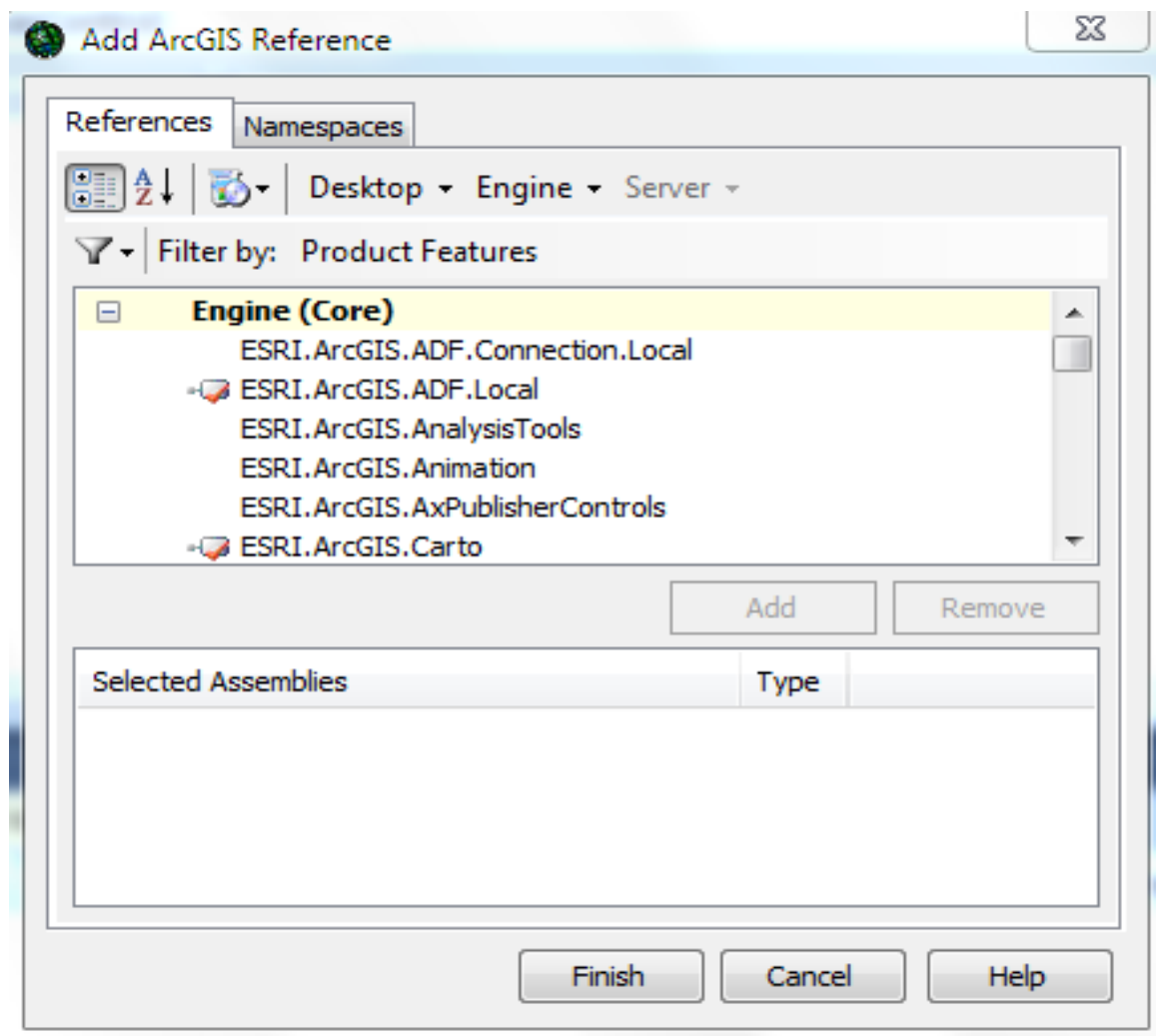
Bước cuối cùng trong tính toán dòng chảy tích lũy là đếm số dòng chảy qua các điểm trung gian và được tích lũy ở điểm cuối trong ma trận mới được lập ở trên (Phụ lục 9). Kết quả tính toán dòng chảy tích lũy được thể hiện như Hình 4-4.

ncols	43																																									
nrows	38																																									
xllcorner	106.78291972546																																									
yllcorner	10.866678949909																																									
cellsize	0.00027766029143888																																									
NODATA_value	-9999																																									
0 13 2 1 0 0 2 0 0 0 0 1 0 0 0 0 0 0 0 5 1 0 0 0 6 7 49 2 0 0 0 0 0 4 2 0 0 14 6 2 0 0 13																																										
0 8 3 1 0 0 3 0 0 2 3 0 0 1 2 0 0 2 0 0 0 0 1 3 0 4 2 29 0 23 0 1 0 0 0 0 5 1 0 0 0 11																																										
0 0 0 0 0 5 1 1 0 8 7 4 0 3 1 4 0 1 2 1 0 0 0 3 0 0 2 25 0 18 3 5 4 1 0 0 1 0 0 0 1 7																																										
0 0 0 0 3 0 0 19 19 0 48 3 0 0 0 14 0 38 15 1 0 0 1 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 1 5 0																																										
23 1 63 3 0 0 19 4 4 21 0 1 0 0 0 19 0 7 0 0 1 2 0 0 1 0 0 0 0 1 0 0 3 0 5 2 0 0 0 0 0 0																																										
0 0 51 0 0 0 0 3 3 17 0 0 0 0 2 1 1 1 3 0 1 5 1 0 1 0 2 1 134 1 0 0 0 4 6 1 0 0 1 1 0 0 2																																										
0 1 0 40 1 0 2 1 0 16 1 0 0 5 0 0 0 0 0 0 78 71 60 0 0 8 4 4 123 1 0 0 0 65 48 0 0 0 88 3 1 0 0																																										
0 0 36 0 0 0 0 0 1 13 1 0 7 0 0 0 2 0 0 3 3 4 50 11 9 0 1 5 4 109 0 0 9 1 35 0 2 0 0 78 0 0 0																																										
0 1 2 31 3 0 0 0 1 0 5 0 152 11 8 7 0 0 2 2 3 4 32 0 0 0 4 3 99 0 6 6 0 3 29 9 2 0 1 0 71 1 0																																										
1 0 7 1 16 9 1 0 0 0 3 128 0 0 0 0 0 1 2 3 6 1 21 0 3 2 97 0 5 1 2 2 3 14 0 2 0 0 0 66 1 0																																										
3 6 0 4 0 0 4 1 0 0 1 126 0 9 7 3 0 0 1 2 5 0 16 0 0 2 1 96 1 2 0 1 1 2 11 0 0 0 0 4 0 57 0																																										
0 0 1 0 0 0 0 0 0 0 0 125 10 0 2 0 1 0 1 3 0 4 10 2 0 1 0 95 0 0 0 0 0 1 0 9 6 0 0 1 2 51 0																																										
0 0 0 0 18 1 0 0 1 1 0 10 101 0 72 0 0 0 1 0 3 1 4 1 0 0 0 6 87 86 0 0 0 0 0 0 3 0 0 0 49 0																																										
1 30 28 25 0 15 14 7 0 2 9 21 0 76 0 71 0 0 0 0 1 0 0 0 0 0 0 4 0 83 0 3 0 0 1 0 0 0 0 4 1 3																																										
36 1 1 3 0 3 0 1 0 7 0 19 0 0 0 0 68 1 0 0 0 0 1 0 0 0 2 0 17 2 60 20 13 6 3 2 0 0 0 0 1 0 2																																										
0 0 0 0 0 0 1 0 2 0 2 0 13 11 0 1 0 61 2 0 12 10 3 1 0 0 0 1 13 1 38 1 2 1 0 0 4 0 0 0 0 0																																										
0 5 0 0 0 0 0 0 0 0 0 0 6 1 0 0 20 0 35 31 0 1 0 0 1 1 0 11 0 0 36 0 0 1 0 0 5 1 0 0 0 0																																										
7 0 2 0 93 90 0 0 0 0 0 2 0 1 0 4 1 12 0 1 17 0 0 0 3 51 1 8 0 0 6 28 5 2 0 0 3 27 11 0 0 3 0																																										
20 2 0 0 1 5 79 75 1 3 1 0 0 0 0 1 1 0 11 0 0 15 0 0 1 12 35 2 1 1 1 2 21 0 0 0 2 0 0 5 4 8 77																																										
7 0 0 0 4 2 2 1 66 52 0 0 0 0 0 0 2 6 0 2 10 2 0 0 4 34 1 0 0 0 1 1 18 1 0 0 0 1 2 3 44 0																																										
0 5 0 2 0 1 1 0 8 0 50 49 7 2 0 0 0 3 0 1 5 0 0 3 31 0 0 0 0 0 0 1 7 7 2 0 0 0 1 2 40 2 0																																										
0 0 1 0 0 0 0 5 0 0 34 4 0 0 0 0 0 0 1 0 0 3 2 27 2 1 0 0 14 0 0 1 4 2 0 0 0 2 3 37 1 0 0																																										
0 73 0 0 0 0 1 2 2 10 32 0 3 1 0 38 0 1 0 0 2 1 4 16 4 0 0 2 8 0 0 0 0 0 0 1 0 1 32 1 0 0																																										
75 0 72 67 0 0 1 1 6 1 1 15 0 0 4 29 2 0 0 1 0 3 13 1 0 1 0 3 2 0 0 0 0 1 0 0 0 2 0 27 0 0 2																																										
85 0 1 1 63 0 0 0 5 0 0 5 7 0 1 1 25 2 0 0 2 10 1 0 0 0 0 0 0 0 0 0 1 4 1 0 0 3 18 21 0 1 0 90																																										
0 5 0 0 1 58 2 0 4 0 3 0 0 4 0 0 20 1 0 0 0 0 8 0 0 0 0 0 0 0 1 3 8 0 0 9 0 0 0 0 0 92																																										
0 2 0 0 7 12 33 0 0 1 1 0 2 0 0 2 17 0 0 0 0 4 0 0 0 2 1 0 1 2 0 15 1 0 3 6 1 0 0 0 1 0 0																																										
0 0 0 5 0 11 3 32 1 0 0 0 1 0 0 0 3 8 3 0 0 0 0 0 0 1 5 1 6 0 1 25 1 0 1 0 0 1 0 0 0 2 9																																										
0 0 1 2 5 0 2 28 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 3 155 140 132 127 2 26 4 1 0 0 0 0 6 1 1 0 5 0																																										
8 0 1 2 1 0 0 0 27 25 23 19 17 0 4 2 1 0 0 0 3 0 0 2 11 0 1 3 11 100 0 0 0 0 0 0 2 7 0 2 0 0 0																																										
4 1 0 0 0 0 0 0 0 1 1 3 7 0 0 0 0 0 5 0 0 0 1 3 0 2 10 69 0 0 13 0 2 0 0 1 20 20 5 0 3 0																																										
0 0 0 0 2 1 9 6 4 3 0 0 2 0 0 1 2 0 1 7 0 1 0 0 0 1 9 10 0 56 15 0 11 8 2 0 0 20 1 0 39 25 4																																										
2 0 0 4 0 10 3 2 0 1 0 1 0 0 0 1 6 0 0 10 1 2 1 0 0 8 6 2 2 36 1 0 0 0 1 0 1 1 0 2 1 12 2																																										
0 0 14 2 0 17 1 0 0 0 0 0 1 5 8 31 13 4 0 0 16 0 0 2 7 5 1 1 0 32 0 0 0 0 1 0 0 0 0 0 11 1																																										
0 2 0 0 0 18 2 1 0 0 0 0 0 1 0 3 1 0 0 3 37 27 7 2 4 3 0 0 0 2 0 26 4 1 0 4 1 0 0 0 0 1 8 0																																										
0 0 0 0 0 23 0 0 1 0 0 1 0 0 1 0 0 1 0 3 1 2 0 0 2 1 0 0 0 2 0 16 0 7 0 0 0 2 0 0 0 0 6 0																																										
53 50 46 44 40 38 3 0 4 1 1 3 0 0 0 0 0 0 0 2 0 1 0 1 0 1 0 0 0 0 4 0 8 0 0 1 1 4 4 1 0 0 2																																										
0 0 0 0 0 0 8 7 1 0 0 11 3 1 0 9 1 3 0 0 0 0 0 1 12 4 3 1 0 0 0 1 0 0 6 0 0 14 0 1 0 0 1																																										

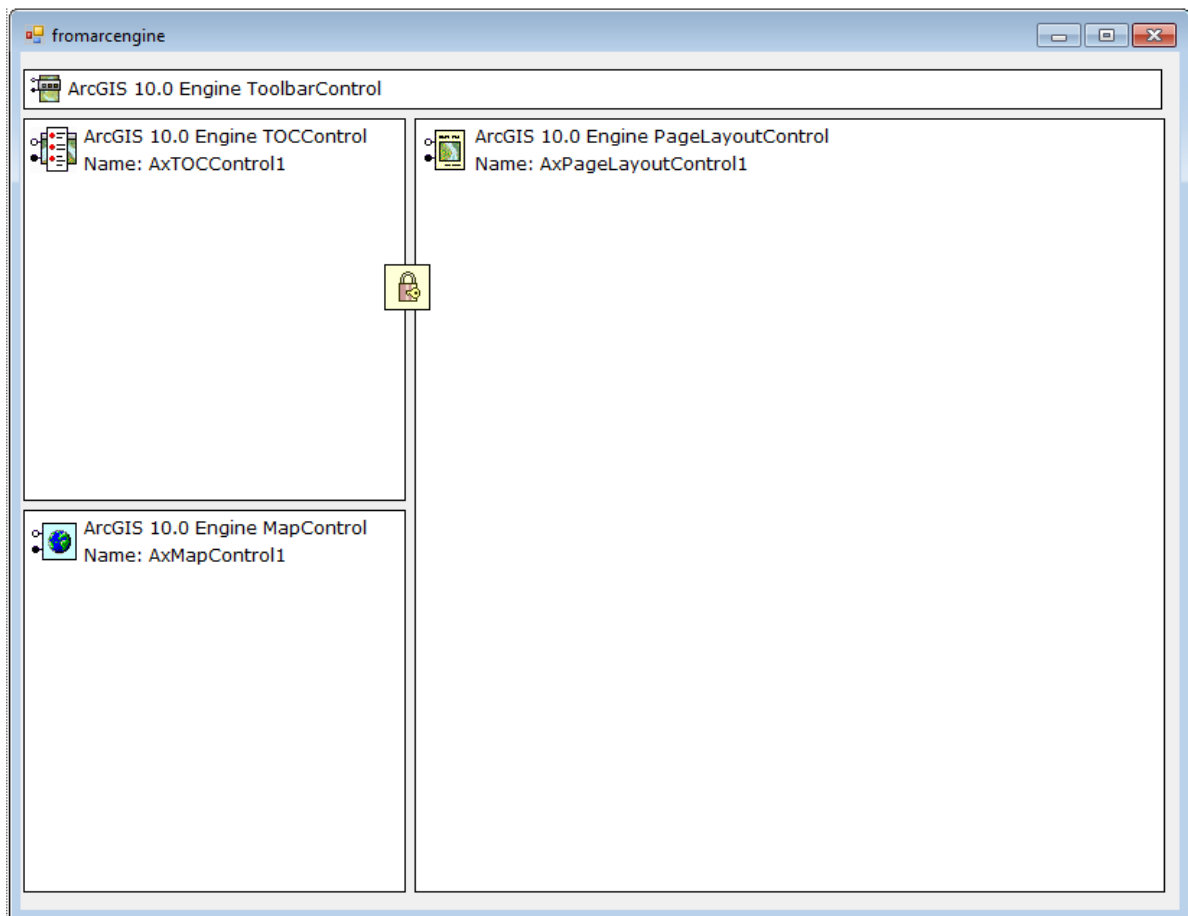
Hình 4-4. Dữ liệu về dòng chảy tích lũy được tính toán bằng phần mềm và lưu ở dạng Text.

4.1.4. Ứng dụng Arc Engine để thể hiện bản đồ dòng chảy

Để có thể hiển thị được dữ liệu dòng chảy tích lũy trong DEM cần sử dụng đến bộ thư viện Arc Engine của ESRI. Trước hết cần phải tạo mới một Form sau đó dùng Add reference ArcGis để thêm bộ thư viện Arc Engine vào form (như Hình 4-5). Trong Visual Studio chọn Toolbox kéo các tool trong ArcGis Windows Form vào form hiển thị, sẽ sử dụng các tool là ToolbarControl, MapControl, PageLayerControl, TOCControl và LincenseControl (Hình 4-6).



Hình 4-5. Bộ thư viện Arc Engine của ESRI



Hình 4-6. Sử dụng các ArcGis Window Form trong Toolbox lên form hiển thị

Chọn Properties của form hiển thị, đến Events là Load đặt tên là Map_viewer, Map_Viewer này dùng để load dữ liệu lên cho tất cả các Tool trên form, viết đoạn code sau vào Map_Viewer:

```
Private Sub MapViewer_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    'Load a pre-authored map document into the PageLayoutControl using
    relative paths.

    Dim sFileName As String = "C:\Users\Admin\Desktop\DU_LIEU_NL\DEM.mxd"

    If AxPageLayoutControl1.CheckMxFile(sFileName) Then

        AxPageLayoutControl1.LoadMxFile(sFileName)

    End If

End Sub
```

Với “sFileName” là đường dẫn đến dữ liệu muốn hiển thị lên PageLayerControl. Code trong đoạn If dùng để kiểm tra file dữ liệu và thực hiện load dữ liệu lên.

Vào Properties của PageLayerControl chọn đến Events OnPageLayoutReplaced, Events này được dùng để liên kết dữ liệu được hiển thị trên PageLayerControl với MapControl. Sau đây là code cho events này:

```
Private Sub AxPageLayoutControl1_OnPageLayoutReplaced(ByVal sender As
System.Object, ByVal e As
ESRI.ArcGIS.Controls.IPageLayoutControlEvents_OnPageLayoutReplacedEvent)
Handles AxPageLayoutControl1.OnPageLayoutReplaced

    'Load the same pre-authored map document into the MapControl.

    AxMapControl1.LoadMxFile(AxPageLayoutControl1.DocumentFilename)

    'Set the extent of the MapControl to the full extent of the data.

    AxMapControl1.Extent = AxMapControl1.FullExtent
End Sub
```

Sau cùng trở lại Map_Viewer thêm vào đoạn code sau:

```
AxTOCControl1.SetBuddyControl(AxPageLayoutControl1)
AxToolbarControl1.SetBuddyControl(AxPageLayoutControl1)
```

Đoạn code trên để hiển thị thông tin dữ liệu lên TOCControl.

Để thêm các tool trên ToolbarControl, có thể sử dụng các tool đã được cung cấp sẵn bằng việc sử dụng đến hàm AddItem và chọn các tool cần thiết. Thêm đoạn code sau vào Map_Viewer để hiển thị các tool lên ToolbarControl:

```
'Add generic commands.

AxToolbarControl1.AddItem("esriControls.ControlsOpenDocCommand", -1, -1,
False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolbarControl1.AddItem("esriControls.ControlsAddDataCommand", -1, -1,
False, 0, esriCommandStyles.esriCommandStyleIconOnly)

'Add page layout navigation commands.

AxToolbarControl1.AddItem("esriControls.ControlsPageZoomInTool", -1, -1,
True, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolbarControl1.AddItem("esriControls.ControlsPageZoomOutTool", -1, -1,
False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolbarControl1.AddItem("esriControls.ControlsPagePanTool", -1, -1,
False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolbarControl1.AddItem("esriControls.ControlsPageZoomWholePageCommand",
-1, -1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)

'Add map navigation commands.

AxToolbarControl1.AddItem("esriControls.ControlsMapZoomInTool", -1, -1,
True, 0, esriCommandStyles.esriCommandStyleIconOnly)
```

```

AxToolBarControl1.AddItem("esriControls.ControlsMapZoomOutTool", -1, -1,
False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapPanTool", -1, -1, False,
0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapFullExtentCommand", -1,
-1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapZoomToLastExtentBackCommand"
, -1, -1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapZoomToLastExtentForwardComma
nd", -1, -1, False, 0, esriCommandStyles.esriCommandStyleIconOnly)

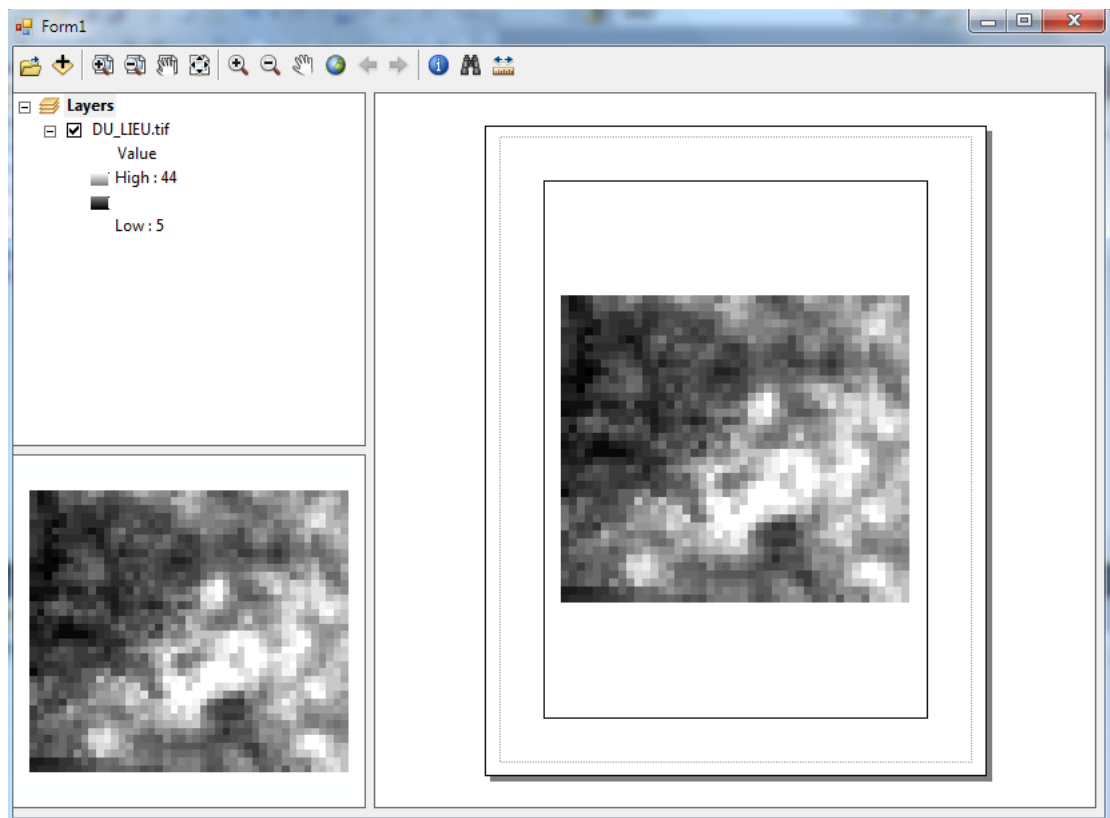
'Add map inquiry commands.

AxToolBarControl1.AddItem("esriControls.ControlsMapIdentifyTool", -1, -1,
True, 0, esriCommandStyles.esriCommandStyleIconOnly)

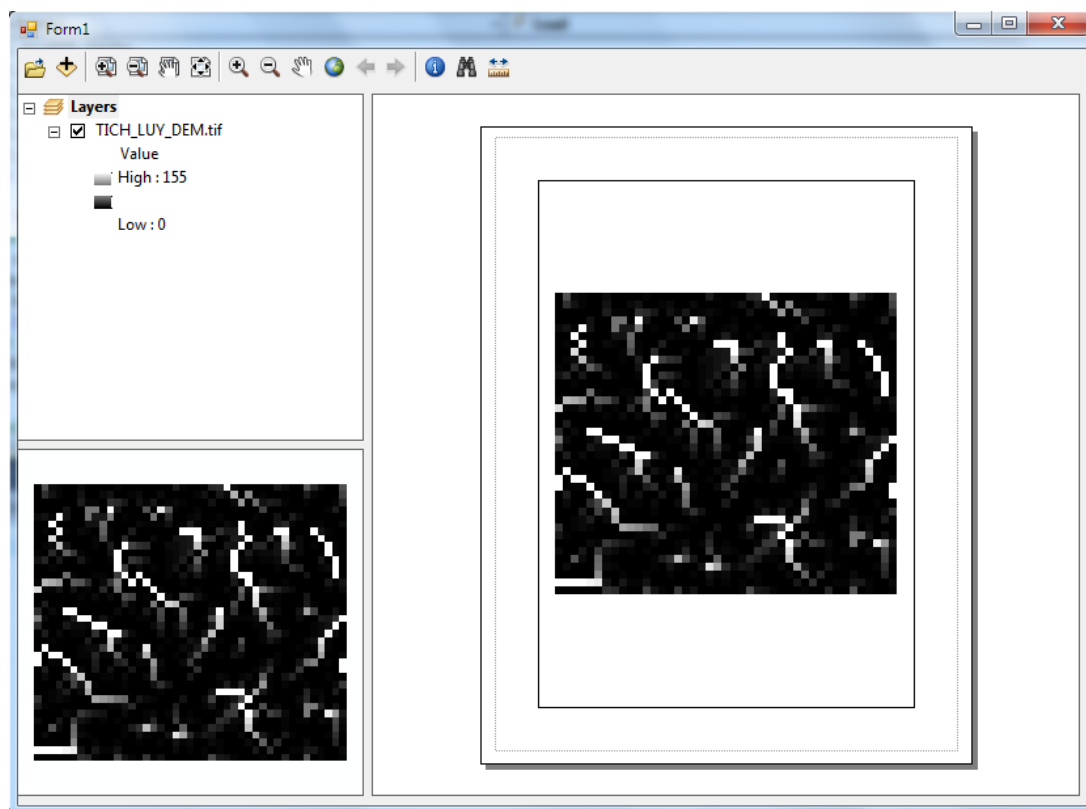
AxToolBarControl1.AddItem("esriControls.ControlsMapFindCommand", -1, -1,
False, 0, esriCommandStyles.esriCommandStyleIconOnly)

AxToolBarControl1.AddItem("esriControls.ControlsMapMeasureTool", -1, -1,
False, 0, esriCommandStyles.esriCommandStyleIconOnly)

```



Hình 4-7. Hiển thị dữ liệu DEM khu vực Nông Lâm trên form



Hình 4-8. Dữ liệu dòng chảy của khu vực nghiên cứu sau khi thực hiện tính toán trên phần mềm

CHƯƠNG 5. KẾT LUẬN VÀ KIẾN NGHỊ

5.1. Kết luận

Các kết quả mà đề tài đạt được:

- Xây dựng thành công giao diện người dùng và các modun khác trong phần mềm.
- Cài đặt được thuật toán tính hướng dòng chảy D8 và tính tích lũy dòng chảy trên Visual Studio bằng ngôn ngữ lập trình Visual Basic.
- Tích hợp Arc Engine vào phần mềm để hiển thị kết quả sau khi tính toán.
- Xây dựng các modun xử lý dữ liệu bằng Python và tích hợp được vào phần mềm.
- Hiển thị được kết quả trên form.

Hạn chế của đề tài:

- Thời gian tính toán dòng chảy tích lũy khá lâu khoảng 20 phút.
- Tích hợp các modun xử lý dữ liệu thông qua trình biên dịch Python nên khả năng liên kết với phần mềm chưa cao.
- Form hiển thị kết quả chỉ cung cấp một vài Tool cơ bản để thao tác với dữ liệu như phóng to thu nhỏ, thêm lớp dữ liệu,...
- Phần mềm chỉ sử dụng thuật toán D8, chưa cài đặt được các thuật toán khác.
- Chưa cung cấp các công cụ nhằm phân tích hỗ trợ ra quyết định dựa trên kết quả thu được.

5.2. Đề xuất hướng phát triển

Dựa trên các hạn chế của đề tài và thực tế khu vực, có một số đề xuất hướng phát triển và cải tiến phần mềm sau:

- Sử dụng thuật toán song song để tăng tốc độ xử lý dữ liệu.
- Phát triển thêm các tool mới nhằm hỗ trợ việc quản lý, khai thác, phân tích dữ liệu tốt hơn để đưa ra kết quả chính xác góp phần hỗ trợ ra quyết định xây dựng, nâng cấp cơ sở hạ tầng.
- Tích hợp thêm nhiều thuật toán tính hướng dòng chảy để có thể lựa chọn thuật toán phù hợp với yêu cầu và đặc trưng của dữ liệu.

TÀI LIỆU THAM KHẢO

Tiếng việt

- Nguyễn Kim Lợi, Lê Cảnh Đình và Trần Thống Nhất, 2009. *Hệ thống thông tin địa lý nâng cao*. NXB Nông Nghiệp, Tp. Hồ Chí Minh.
- Nguyễn Kim Lợi và Trần Thống Nhất, 2007. *Hệ thống thông tin địa lý*. NXB Nông Nghiệp, Tp. Hồ Chí Minh.
- Lê Minh Hoàng, 2003. *Giải thuật và lập trình*. Đại học Sư Phạm Hà Nội.
- Nguyễn Sơn Hải, 2006. *Giáo trình tin học: Lập trình với Microsoft Visual Basic 6.0*. Hà Nội.
- An Nhơn, 2013. *Hai sinh viên bị nước cuốn trôi trong cơn mưa lớn ở Sài Gòn*. Địa chỉ: <<http://vnexpress.net/tin-tuc/thoi-su/2-nu-sinh-bi-cuon-troi-trong-con-mua-lon-o-sai-gon-2846340.html>> [Truy cập ngày: 13/5/2014].
- Phòng quản trị vật tư, 2014. *Cơ sở vật chất trường Đại học Nông Lâm*. Địa chỉ: <<http://pqtvt.hcmuaf.edu.vn/pqtvt-10423-1/vn/co-so-vat-chat.html>> [Truy cập ngày: 15/5/2014].
- Trường Đại học Nông Lâm, 2011. *Dự thảo: Chiến lược phát triển trường Đại học Nông Lâm TP. HCM 2011 – 2020*, Tp. Hồ Chí Minh.

Tiếng anh

- Dave Kuhlman, 2009. *A Python Book: Beginning Python, Advanced Python, and Python Exercises*.
- Bradley Miller, 2010. *Comparison of Surface Hydrologic Algorithms in GIS*. Michigan state university.
- Chase Wallis, Dan Watson, David Tarboton, Robert Wallace, 2009. *Parallel Flow-Direction and Contributing Area Calculation for Hydrology Analysis in Digital Elevation Models*.
- Wan MuhdHairi bin Wan AbKarim, 2012. *New D16 Algorithm for surface water flow direction*. University Teknologi Malaysia (UTM) Skudai, Johor.
- Arcgis Resource Center, 2011. *Hydrologic analysis sample applications*.

Available at:

<<http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/009z0000005z000000.htm>> [Accessed May 5, 2014].

Hydrology Resource Group, 2014. Terrain Analysis using digital elevation models (TauDEM).

Available at: <<http://hydrology.usu.edu/taudem/taudem5/index.html>> [Accessed May 15, 2014].

USGS science for a changing world, 2013 . *Global Data Explorer*. Available at: <<http://gdex.cr.usgs.gov/gdex/>> [Accessed May 8, 2014].

Strahler, A.N, 1957. *Quantitative analysis of watershed geomorphology*, *Transactions of the American Geophysical Union* 38: 913 – 920.

PHỤ LỤC

Phụ lục 1: Code cho Fill DEM

```
import Tkinter,tkFileDialog
import arcpy
from arcpy import env
from arcpy.sa import *
root = Tkinter.Tk()
root.withdraw()
myFormats_DEM = [('Chon File','*.tif') ]
myFormats_TXT = [ ('Chon noi luu du lieu','*.*)']
file=tkFileDialog.askopenfile(parent=root,
mode='rb',filetypes=myFormats_DEM,title='Choose file DEM')
if file != None:
    duongdan_input=file.name
    inSurfaceRaster = duongdan_input
    print(inSurfaceRaster)
    arcpy.CheckOutExtension("Spatial")
    save=tkFileDialog.asksaveasfilename(parent=root,filetypes=myFormats_TXT
,title="Save file as...")
    if save!=None:
        duongdan_output=save
        print(duongdan_output)
        # arcpy.CheckOutExtension("Spatial")
        outFill= Fill(inSurfaceRaster)
        outFill.save(duongdan_output)
    else:
        file.close()
```

Phụ lục 2: Code chuyển từ dạng Raster sang dạng Text

```
import Tkinter,tkFileDialog
import arcpy
from arcpy import env
root = Tkinter.Tk()
root.withdraw()
myFormats_TXT = [ ('Chon File', '*.txt'),]
file = tkFileDialog.askdirectory(parent=root,initialdir="",title='Chon duong dan
den du lieu DEM')
if file !=None:
    duongdan_input=file
    inRaster = duongdan_input
    print(duongdan_input)
    save=tkFileDialog.asksaveasfilename(parent=root,filetypes=myFormats_TXT,
    title="Save file as...")
    if save!= None:
        duongdan_output=save
        outASCII = duongdan_output
        print(duongdan_output)
        arcpy.RasterToASCII_conversion(inRaster, outASCII)
    else:
        file.close()
```

Phụ lục 3: Code cho Đọc dữ liệu

'Đọc file txt matrix lấy giá trị để thực hiện tính

Dim duongdan As String

OpenFileDialog1.Title = "Chọn tập tin: "

OpenFileDialog1.Filter = "Dạng tập tin (*.txt)|*.txt"

If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK

Then

duongdan = OpenFileDialog1.FileName

txbnoiluutxt.Text = duongdan

```

Dim h, c As String
Dim ha, n As Integer
Dim na As String
Dim reco() As String = IO.File.ReadAllLines(duongdan) 'đọc tất cả các
    hàng
c = reco(0)
cot = CInt(Mid(c, 14)) 'lấy số cột
h = reco(1)
hang = CInt(Mid(h, 14)) 'lấy số hàng
size = hang * cot
For n = 0 To 5 'LẤY RA CÁC DÒNG ĐẦU TIÊN
    chuoi = chuoi + reco(n) + vbNewLine
Next
If hang > 0 And cot > 0 Then 'redim lại kích thước của ma trận
    ReDim docao(hang, cot)
    ReDim huong(hang, cot)
    ReDim tichluy(hang, cot)
    ReDim ketqua(hang, cot)
    ReDim matran1(size, size)
    ReDim matran2(size, size)
    ReDim matranmoi(size, size)
End If
'Lấy ra ma trận trong file text
For ha = 6 To hang + 5
    matran = matran + reco(ha) + vbNewLine
Next
Dim path As String = "D:\\Matrix_dem.txt" 'xuất ma trận ra text file
If File.Exists(path) = True Then
    Kill(path) 'Xóa file
    Dim sw As StreamWriter = File.CreateText(path)
    sw.WriteLine(matran)

```

```

        sw.Flush()
        sw.Close()
Else
    Dim sw As StreamWriter = File.CreateText(path)
    sw.WriteLine(matran)
    sw.Flush()
    sw.Close()
End If

Dim giatri() As String = IO.File.ReadAllLines("D:\\Matrix_dem.txt")
For i = 0 To hang - 1
    For j = 0 To cot - 1
        na = giatri(i)
        Dim pt() As String = na.Split(" ")
        docao(i, j) = pt(j)
    Next
Next

Next

For i = 0 To hang - 1
    For j = 0 To cot - 1
        hienthi = hienthi & docao(i, j) & " "
    Next
    hienthi = hienthi + vbNewLine
Next

txbhienthi.Text = chuoi & hienthi
txbsohang.Text = hang
txbsocot.Text = cot
FileClose()
End If

```

Phụ lục 4: Code cho Lưu dữ liệu

```
Dim path2 As String = txbhienthi.Text
If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK
Then
    SaveFileDialog1.Title = "Chọn nơi lưu tập tin kết quả"
    System.IO.File.WriteAllText(SaveFileDialog1.FileName, path2)
End If
```

Phụ lục 5: Code chuyển Text sang DEM

```
import Tkinter,tkFileDialog
import arcpy
root = Tkinter.Tk()
root.withdraw()
myFormats_TXT = [('Chon File','*.txt'),]
myFormats_DEM = [('Chon noi luu du lieu','*.tif'), ]
file = tkFileDialog.askopenfile(parent=root,
mode='rb',filetypes=myFormats_TXT ,title='Choose file DEM')
if file != None:
    duongdan_input=file.name
    inASCII = duongdan_input
    print(inASCII)
save=tkFileDialog.asksaveasfilename(parent=root,filetypes=myFormats_DEM
,title="Save file as...")
if save!=None:
    outRaster=save
    print(outRaster)
    arcpy.ASCIIToRaster_conversion(inASCII, outRaster, rasterType)
else:
    file.close()
```


Phụ lục 6: Code cho Hướng dòng chảy

Đọc file txt matrix lấy giá trị để thực hiện tính

TÍNH D8:

For i = 0 To hang - 1

For j = 0 To cot - 1

Dim max As Double = 0

Dim direct As Integer = 0

'XÉT 8 TRƯỜNG HỢP:

If j + 1 < cot Then ' 1

max = Math.Max(max, (docao(i, j) - docao(i, j + 1)))

If max = (docao(i, j) - docao(i, j + 1)) Then

direct = 1

End If

End If

If i + 1 < hang And j + 1 < cot Then '2

max = Math.Max(max, (docao(i, j) - docao(i + 1, j + 1)) / Math.Sqrt(2))

If max = (docao(i, j) - docao(i + 1, j + 1)) / Math.Sqrt(2) Then

direct = 2

End If

End If

If i + 1 < hang Then '4

max = Math.Max(max, (docao(i, j) - docao(i + 1, j)))

If max = (docao(i, j) - docao(i + 1, j)) Then

direct = 4

End If

End If

If i + 1 < hang And j - 1 >= 0 Then '8

max = Math.Max(max, (docao(i, j) - docao(i + 1, j - 1)) / Math.Sqrt(2))

If max = (docao(i, j) - docao(i + 1, j - 1)) / Math.Sqrt(2) Then

direct = 8

End If

```

End If
If j - 1 >= 0 Then '16
    max = Math.Max(max, (docao(i, j) - docao(i, j - 1)))
    If max = (docao(i, j) - docao(i, j - 1)) Then
        direct = 16
    End If
End If
If i - 1 >= 0 And j - 1 >= 0 Then '32
    max = Math.Max(max, (docao(i, j) - docao(i - 1, j - 1)) / Math.Sqrt(2))
    If max = (docao(i, j) - docao(i - 1, j - 1)) / Math.Sqrt(2) Then
        direct = 32
    End If
End If
If i - 1 >= 0 Then '64
    max = Math.Max(max, (docao(i, j) - docao(i - 1, j)))
    If max = (docao(i, j) - docao(i - 1, j)) Then
        direct = 64
    End If
End If
If i - 1 >= 0 And j + 1 < cot Then '128
    max = Math.Max(max, (docao(i, j) - docao(i - 1, j + 1)) / Math.Sqrt(2))
    If max = (docao(i, j) - docao(i - 1, j + 1)) / Math.Sqrt(2) Then
        direct = 128
    End If
End If
huong(i, j) = direct
Next
Next
Dim hienthi1 As String
For i = 0 To hang - 1
    For j = 0 To cot - 1

```

```

        hienthi1 = hienthi1 & huong(i, j) & " "
    Next
    hienthi1 = hienthi1 + vbNewLine
Next
txbhienthi.Text = chuoi & hienthi1

```

Phụ lục 7: Code lập ma trận mới

```

Dim size1 As Integer = hang * cot
For i = 0 To size1 - 1
    For j = 0 To size1 - 1
        If i = j Then
            matran1(i, j) = 0
        Else
            matran1(i, j) = Integer.MaxValue
        End If
    Next j
Next i
For i = 0 To hang - 1
    For j = 0 To cot - 1
        k = (i) * cot + j
        If (huong(i, j) = 1) Then
            l = k + 1
        End If
        If (huong(i, j) = 2) Then
            l = k + cot + 1
        End If
        If (huong(i, j) = 4) Then
            l = k + cot
        End If
        If (huong(i, j) = 8) Then
            l = k + cot - 1

```

```

End If
If (huong(i, j) = 16) Then
    l = k - 1
End If
If (huong(i, j) = 32) Then
    l = k - cot - 1
End If
If (huong(i, j) = 64) Then
    l = k - cot
End If
If (huong(i, j) = 128) Then
    l = k - cot + 1
End If
matran1(l, k) = 1
matran1(k, l) = -1
Next j
Next i

'Gán thêm một ma trận
Dim q, p, r, s As Integer
For i = 0 To size1 - 1
    For j = 0 To size1 - 1
        If i <> j Then
            p = i \ cot
            q = i Mod cot
            r = j \ cot
            s = j Mod cot
            If ((docao(p, q) > docao(r, s)) Or (docao(p, q) = -9999)) Then
                matran1(i, j) = -1
            End If
        End If
    Next j
End If

```

```

        matran2(i, j) = matran1(i, j)
        matranmoi(i, j) = matran1(i, j)
    Next j
Next i

```

Phụ lục 8: Code cộng hai ma trận

```

Dim giatri As Integer = Integer.MaxValue
Dim cothaydoi As Boolean = False
Do
    cothaydoi = False
    'Nhân hai ma trận
    For i = 0 To size1 - 1
        For j = 0 To size1 - 1
            If matranmoi(i, j) > 0 Then
                For k = 0 To size1 - 1
                    If (matran1(i, k) >= 0) And (matran2(k, j) >= 0) Then
                        If (matran1(i, k) <> Integer.MaxValue) And (matran2(k, j) <>
Integer.MaxValue) Then
                            giatri = matran1(i, k) + matran2(k, j)
                            If (matranmoi(i, j) > giatri) Then
                                matranmoi(i, j) = giatri
                                cothaydoi = True
                            End If
                        End If
                    End If
                Next k
            End If
        Next j
    Next i
    'copy lại ma trận cũ bằng ma trận mới
    For i = 0 To size1 - 1

```

```

For j = 1 To size1 - 1
    matran1(i, j) = matranmoi(i, j)
    matran2(i, j) = matranmoi(i, j)
Next j
Next i
Loop While cothaydoi = True

```

Phụ lục 9: Code xác định dòng chảy tích lũy

```

For i = 0 To size1 - 1
    Dim tl As Integer = 0
    For j = 0 To size1 - 1
        If (i <> j) And (matran1(i, j) < Integer.MaxValue) And
            (matran1(i, j) >= 0) Then
            tl += 1
        End If
    Next
    tichluy(i \ cot, i Mod cot) = CStr(tl)
Next
Dim hienthi2 As String
For i = 0 To hang - 1
    For j = 0 To cot - 1
        hienthi2 = hienthi2 & tichluy(i, j) & " "
    Next
    hienthi2 = hienthi2 + vbNewLine
Next
txbhienthi.Text = chuoi & hienthi2

```