แบบจำลองทางคณิตศาสตร์การไหลเวียนเลือดของมนุษย์

COMPUTATION MODELLING OF BLOOD FLOW IN THE HUMAN ARTERIAL SYSTEM

นางสาวธันย์ชนก ทองมาก
นางสาวปราณปริยา สันทรายเงิน
นางสาวสิรีธร สมสะอื้น

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

แบบจำลองทางคณิตศาสตร์การไหลเวียนเลือดของมนุษย์

COMPUTATION MODELLING OF BLOOD FLOW IN THE HUMAN ARTERIAL SYSTEM

นางสาวธันย์ชนก ทองมาก
นางสาวปราณปริยา สันทรายเงิน
นางสาวสิรีธร สมสะอื้น

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

COMPUTATION MODELLING OF BLOOD FLOW IN THE HUMAN ARTERIAL SYSTEM

Thanchanok Thongmak
Pranpriya Sunsignngen
Sireetorn Somsaauen

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHANICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2018

ปริญญานิพนธ์ปีการศึกษา 2561
สาขาวิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แบบจำลองทางคณิตศาสตร์การไหลเวียนเลือดของมนุษย์
COMPUTATION MODELLING OF BLOOD FLOW IN THE HUMAN ARTERIAL SYSTEM

ผู้จัดทำ
1. นางสาวธันย์ชนก ทองมาก รหัสประจำตัว 58010579
2. นางสาวปราณปริยา สันทรายเงิน รหัสประจำตัว 58010741
3. นางสาวสิรีธร สมสะอื้น รหัสประจำตัว 58011318

อาจารย์ที่ปรึกษา
(ดร.บำรุง พ่วงเกิด)

แบบจำลองทางคณิตศาสตร์การไหลเวียนเลือดของมนุษย์

นางสาวธันย์ชนก ทองมาก 58010579
นางสาวปราณปริยา สันทรายเงิน 58010741
นางสาวสิรีธร สมสะอื้น 58011318
ดร.บำรุง พ่วงเกิด อาจารย์ที่ปรึกษา

ปีการศึกษา 2561

บทคัดย่อ

ในปัจจุบันมีประชากรจำนวนไม่น้อยที่เสียชีวิตเพราะโรคที่เกี่ยวกับหลอดเลือด ซึ่งการรักษาโดยส่วนใหญ่เป็นการผ่าตัด ถ้าต้องการให้การผ่าตัดเป็นไปได้อย่างราบรื่น การมีแบบจำลองการไหลของเลือดจึงมีประโยชน์ต่อการรักษา วิทยานิพนธ์เล่มนี้ใช้กฎการอนุรักษ์มวล กฎการอนุรักษ์โมเมนตัม สมการความสัมพันธ์ระหว่างความดันกับพื้นที่เพื่อหาสมการที่จำเป็นในการแก้ปัญหาเชิงคณิตศาสตร์ ซึ่งสามารถแก้ปัญหาโดยใช้ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ และเขียนโค้ดในโปรแกรมไพธอนเพื่อจำลองการไหลของเลือดในร่างกายมนุษย์ที่มีสุขภาพแข็งแรงและอยู่ในสภาวะพัก โดยกำหนดให้เลือดเป็นของไหลที่อัดตัวไม่ได้ มีการไหลแบบนิวโตเนียน มีความหนาแน่นคงที่ หลอดเลือดมีความยืดหยุ่นแบบอิลาสติกและผนังภายในหลอดเลือดมีลักษณะเรียบ การจำลองการไหลนั้นทำเพียงหนึ่งมิติเท่านั้น และทำแค่ในหลอดเลือดแดงใหญ่กับหลอดเลือดแดงกลาง ผลลัพธ์ของการจำลองโดยการเขียนโค้ดคือความดัน ณ จุดๆหนึ่ง การทราบความดันนี้เป็นประโยชน์อย่างมากต่อการวินิจฉัยและการวางแผนการรักษา วิทยานิพนธ์เล่มนี้มีวัตถุประสงค์เพื่อศึกษาและวิเคราะห์ลักษณะการไหลของเลือดเพื่อทำนายพฤติกรรมการไหลของเลือดโดยใช้โปรแกรมคอมพิวเตอร์เพื่อสร้างแบบจำลองการไหลของเลือด โดยผู้ที่สนใจสามารถนำวิทยานิพนธ์นี้ไปศึกษาและพัฒนาต่อเพื่อให้มีความแม่นยำมากขึ้นและกว้างขวางมากขึ้น กราฟผลลัพธ์จากการจำลองหลอดเลือดแดงเส้นแรกแสดงค่าความดันมีลักษณะลู่ออก ดังนั้นจึงไม่สามารถดำเนินการคำนวณต่อได้ เนื่องจากต้องนำค่าความดันที่จุดสุดท้ายของหลอดเลือดแดงใหญ่เออร์ต้าไปเป็นค่าเริ่มต้นความดันของหลอดเลือดแดงใหญ่ต่อไป การลู่ออกเป็นผลเนื่องมาจากระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ เนื่องจากสมการนาเวียร์สโตกมีผลลัพธ์สัมพันธ์กับเวลาอยู่ในรูปไฮเพอร์โบลิก และเมื่อทำการขยายไปที่เส้นๆหนึ่ง เส้นนั้นไม่ได้แสดงเป็นเส้นตรงแต่เป็นเส้นที่อยู่ในรูปฟังก์ชันตรีโกณมิติ หากกำหนดขั้นเวลาได้พอดีกับคาบของฟังก์ชันตรีโกณมิติ ผลที่ได้จึงจะเป็นเส้นตรง แต่หากกำหนดขั้นเวลาไม่เหมาะสม ผลที่ได้จะมีค่าลู่ออก ซึ่ง [illegible]

COMPUTATION MODELLING OF BLOOD FLOW IN THE HUMAN ARTERIAL SYSTEM

Thanchanok Thongmak 58010579
Pranpriya Sunsignngen 58010741
Sireetorn Somsaauen 58011318
Dr.Bumroong Puangkird Advisor

Year 2018

ABSTRACT

Nowadays, a large number of popular death are from vascular diseases. Surgery is one of treatment choices. The blood flow model is useful for surgery. This thesis uses conservation of mass, balance of momentum, pressure-area relation to find the suitable equation to solve the problem. The problem can be solved by using finite different method. Use python for simulation. Simulating blood flow in the human body that is healthy and in a state of rest. Assume human blood is incompressible, Newtonian, and have constant density. Defining the blood vessels as elastic, flexible, and smooth wall. In 1D-simulation, aorta and arteries are concerned. The results of simulation by Python are pressure at cross section area which are uniform. For diagnosis and treatment planning, blood pressure is necessary. This thesis aims to study and analyze blood flow characteristics in order to predict blood flow behavior using computer program to generate blood flow model. This thesis can be studied and developed in order to be more accurate and more extensive. The result graph of the first artery simulation shows the divergence pressure. Therefore, the next artery cannot be calculated because the next artery uses the final pressure of the first artery to calculate as a initial pressure. Divergence is a consequence of the finite difference method. Because the Navier Stokes equation has a relation with time in the hyperbolic form. When expanding to a graph line, it is not shown as a straight line but it is trigonometric form. The time step have to be precise with the period of the trigonometric function for straight line result. Otherwise the result will be divergence. Which the time step determination has a very high chance of error.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากผู้วิจัยได้รับความเมตตากรุณา การดูแลเอาใจใส่อย่างดี และความช่วยเหลือโดยตรงจากอาจารย์ที่ปรึกษา ดร.บำรุง พ่วงเกิด ซึ่งท่านคอยให้คำแนะนำและให้แนวทางในการเขียนวิทยานิพนธ์ อีกทั้งยังช่วยแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการดำเนินงานอีกด้วย ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาและขอขอบพระคุณอย่างสูงไว้ ณ โอกาสนี้

สุดท้ายนี้ขอขอบพระคุณบิดา มารดา และครอบครัวของผู้วิจัยที่สนับสนุนด้านการเรียนและคอยให้กำลังใจในยามยากลำบากให้ผ่านพ้นไปด้วยดี และยังเอาใจใส่ในทุกด้านๆ ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้

นางสาวธันย์ชนก ทองมาก
นางสาวปราณปริยา สันทรายเงิน
นางสาวสิรีธร สมสะอื้น

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันโรคที่เกี่ยวข้องกับโรคหลอดเลือดเป็นหนึ่งในสาเหตุหลักที่ทำให้ประชากรในประเทศไทยเสียชีวิต โดยส่วนมากนั้นเป็นภาวะความดันโลหิตสูง โรคหลอดเลือดแดงแข็ง โรคเบาหวาน ซึ่งโรคเหล่านี้ส่งผลให้เกิดหลอดเลือดแดงโป่งพอง และภาวะที่ไขมันจับที่ผนังหลอดเลือดแดงจนก่อให้เกิดโรคหลอดเลือดตีบ เป็นต้น โดยการรักษาโรคที่เกี่ยวข้องกับโรคหลอดเลือดเหล่านี้ได้ จำเป็นต้องทราบพฤติกรรมการไหลของเลือดในหลอดเลือดว่ามีลักษณะอย่างไร เพื่อให้การรักษาเป็นไปอย่างสะดวกยิ่งขึ้น

งานวิจัยนี้ใช้ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ 1 ทิศทางในการจำลองการไหลของเลือดในหลอดเลือดแดงขนาดใหญ่เอออร์ตาและหลอดเลือดแดงขนาดกลางอาร์เทอรีทั่วร่างกาย เพื่อศึกษาพฤติกรรมของเลือด โดยงานวิจัยนี้ได้นำกฎการอนุรักษ์มวล กฎการอนุรักษ์โมเมนตัมและความสัมพันธ์ระหว่างความดันและพื้นที่หน้าตัด มาใช้พิจารณาจากสมมุติฐานที่ตั้งไว้ รวมถึงการเขียนโปรแกรมไพธอน (Python) เพื่อให้เห็นภาพของพฤติกรรมหลอดเลือดชัดเจนยิ่งขึ้น ผลของงานวิจัยนี้สามารถเป็นแนวทางในการศึกษาพฤติกรรมการไหลของเลือดในหลอดเลือดทั่วร่างกายได้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อทำนายพฤติกรรมของเลือด โดยใช้โปรแกรมคอมพิวเตอร์สร้างแบบจำลองการไหลของเลือดในหลอดเลือดแดงใหญ่เอออร์ตาและหลอดเลือดแดงกลางอาร์เทอรี

1.2.2 เพื่อจดบันทึกค่าความดันในหลอดเลือดแดงใหญ่เอออร์ตาและหลอดเลือดแดงกลางอาร์เทอรี

1.3 ค่าคงที่ของงานวิจัย

1.3.1 กำหนดให้ร่างกายแข็งแรง และอยู่ในสภาวะพักซึ่งมีลักษณะการไหลแบบราบเรียบ

1.3.2 เลือดเป็นของไหลที่อัดตัวไม่ได้

1.3.3 ลักษณะการไหลเป็นแบบนิวโตเนียน

1.3.4 ความหนาแน่นของเลือดมีค่า 1060 กิโลกรัมต่อลูกบาศก์เมตร

1.3.5 เลือดมีค่าความหนืดสัมบูรณ์ 2.78 มิลลิปาสคาล วินาที ที่อุณหภูมิร่างกาย 37 องศาเซลเซียส

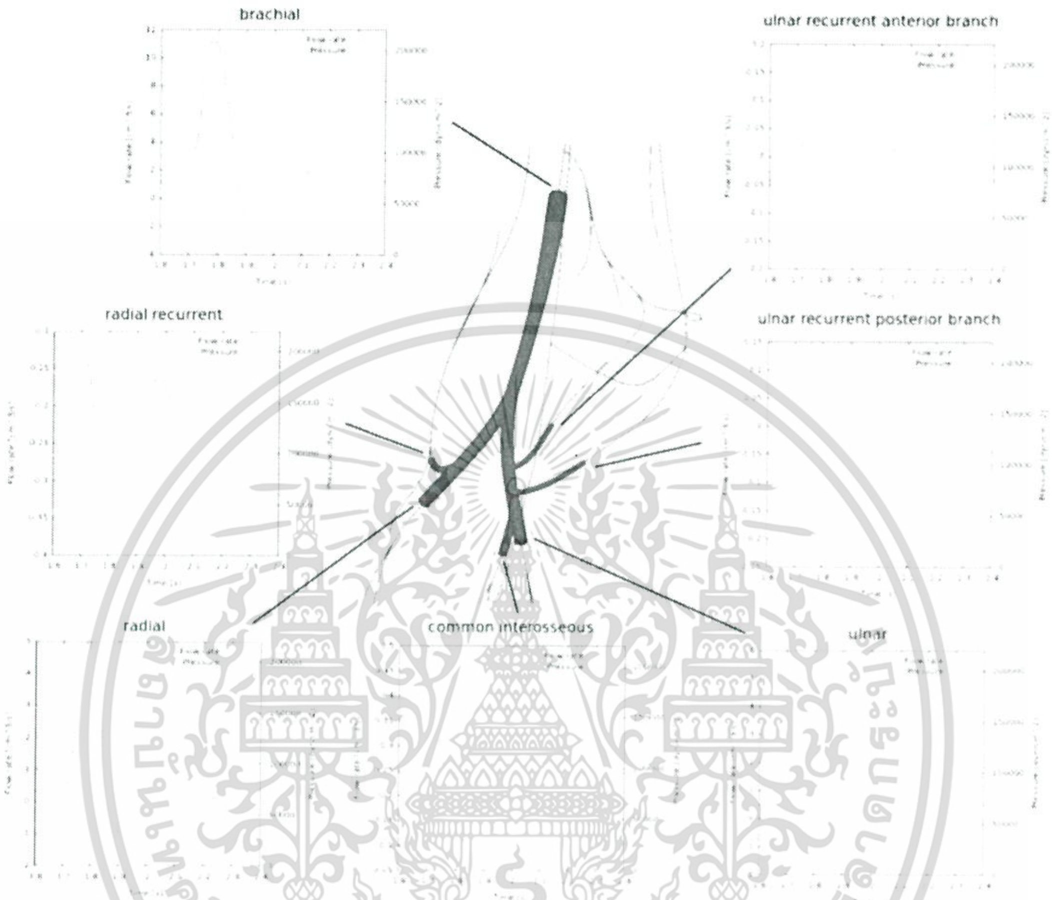
1.3.6 ปริมาตรเลือดที่ออกจากหัวใจมีค่า 5 ลิตรต่อนาที

1.3.7 หลอดเลือดมีความยืดหยุ่นแบบอีลาสติก

1.4 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1.4.1 งานวิจัยเรื่อง Blood flow modeling in a detailed arterial network of the arm โดย Sansuke M. Watanabe, Pablo J. Blanco และ Raúl A. Feijóo งานวิจัยนี้ได้สร้างแบบจำลองของโครงข่ายหลอดเลือดแดงที่แขน โดยมีการพิจารณาทางกายวิภาคและสรีรวิทยาอย่างเคร่งครัดเพื่อติดตามโครงสร้างของหลอดเลือดแดง งานวิจัยนี้ได้นำเอาสมการอนุรักษ์มวลและสมการอนุรักษ์โมเมนตัมมาใช้

พิจารณา ผลลัพธ์ของงานวิจัยนี้คือค่าความดันและอัตราการไหลที่สอดคล้องกับข้อมูลของผู้ป่วยที่ถูกตีพิมพ์ไว้



รูปที่ 1.1 ความดันและอัตราการไหลของหลอดเลือดที่แขน

1.4.2 งานวิจัยเรื่อง Revisiting the pressure-area relation for the flow in elastic tubes:Application to arterial vessels โดย Rafik Absi งานวิจัยนี้อธิบายถึงพฤติกรรมการไหลในหลอดเลือดแดงที่ยืดหยุ่น โดยต้องการศึกษาความสัมพันธ์ระหว่างความดันและพื้นที่หน้าตัด โดยงายวิจัยนี้มีพื้นฐานของทฤษฎีเส้นตรงของความยืดหยุ่น

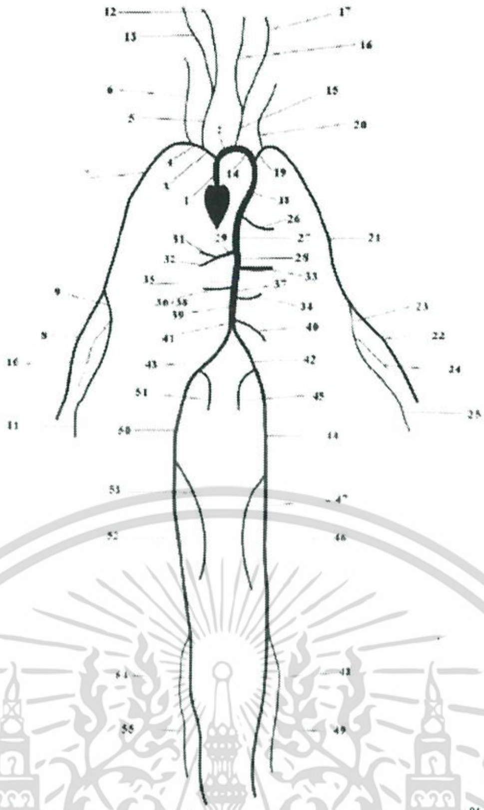
1.4.3 งานวิจัยเรื่อง Computational modelling of 1D blood flow with variable mechanical properties and its application to the simulation of wave propagation in the human arterial system โดย S.J. Sherwin, L. Formaggia, J. Peió และ V. Franke งานวิจัยนี้ได้ทำการศึกษารูปแบบของหลอดเลือดแดงในร่างกายมนุษย์แบบ 1 มิติ โดยการใช้ทั้ง ความไม่ต่อเนื่องของ Galerkin และสมการของ Tayloy-Galerkin โดยงานวิจัยนี้ศึกษาหลอดเลือดแดงทั้งหมด 55 เส้น ดูผลความดันและความเร็วผ่านการจำลองทั้งหมด และในงานวิจัยนี้มีค่าทางกายภาพของหลอดเลือดแนบมาด้วย

ตารางที่ 1.1 คุณสมบัติของหลอดเลือดในโครงข่ายหลอดเลือดแดงในร่างกายมนุษย์

	Artery	Length(cm)	Area(cm^2)	β (kgs^{-2} cm^{-2})
1	Ascending Aorta	4.0	5.983	97
2	Aortic Arch I	2.0	5.147	87
3	Brachiocephalic	3.4	1.219	233
4	R. Subclavian I	3.4	0.562	423
5	R. Carotid	17.7	0.432	516
6	R. Vertebral	14.8	0.123	2590
7	R. Subclavian II	42.2	0.51	466
8	R. Radial	23.5	0.106	2866
9	R. Ulnar I	6.7	0.145	2246
10	R. Interosseous	7.9	0.031	12894
11	R. Ulnar II	17.1	0.133	2446
12	R. Internal Carotid	17.6	0.121	2644
13	R. External Carolid	17.7	0.121	2467
14	Aortic Arch II	3.9	3.142	130
15	L. Carolid	20.8	0.43	519
16	L. Internal Carotid	17.6	0.121	2644
17	L. External Carotid	17.7	0.121	2467
18	Thoracic Aorta I	5.2	3.142	124
19	L. Subclavian I	3.4	0.562	416
20	Vertabral	14.8	0.123	2590
21	L. Subclavian II	42.2	0.51	466
22	L.Radial	23.5	0.106	2866
23	L. Ulnar I	6.7	0.145	2246
24	L. Interosseous	7.9	0.031	12894
25	L. Ulnar II	17.1	0.133	2446
26	Intercostals	8.0	0.196	885
27	Thoracic Aorta II	10.4	3.017	117

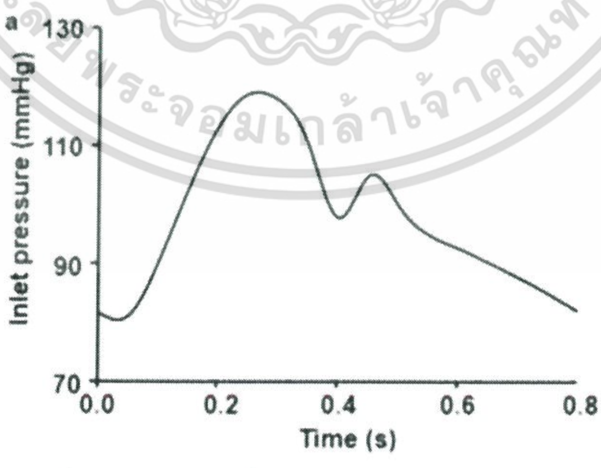
ตารางที่ 1.1 คุณสมบัติของหลอดเลือดในโครงข่ายหลอดเลือดแดงในร่างกายมนุษย์ (ต่อ)

28	Abdominal I	5.3	1.911	167
29	Celiac I	2.0	0.478	475
30	Celiac II	1.0	0.126	1805
31	Hepatic	6.6	0.152	1142
32	Gastric	7.1	0.102	1567
33	Splenic	6.3	0.238	806
34	Superior Mesenteric	5.9	0.43	569
35	Abdominal II	1.0	1.247	227
36	L. Renal	3.2	0.332	566
37	Abdominal III	1.0	1.021	278
38	R. Renal	3.2	0.159	1181
39	Abdominal IV	10.6	0.697	381
40	Inferior Mesenteric	5.0	0.08	1895
41	Abdominal V	1.0	0.578	399
42	R. Common Iliac	5.9	0.328	649
43	L. Common Iliac	5.8	0.328	649
44	L. External Iliac	14.4	0.252	1493
45	L. Internal Iliac	5.0	0.181	3134
46	L. Femoral	44.3	0.139	2559
47	L. Deep Femoral	12.6	0.126	2652
48	L. Posterior Tibial	32.1	0.11	5808
49	L. Anterior Tibial	34.3	0.06	9243
50	R. External Iliac	14.5	0.252	1493
51	R. Internal Iliac	5.1	0.181	3134
52	R. Femoral	44.4	0.139	2559
53	R. Deep Femoral	12.7	0.126	2652
54	L. Posterior Tibial	32.2	0.11	5808
55	R. Anterior Tibial	34.4	0.06	9243

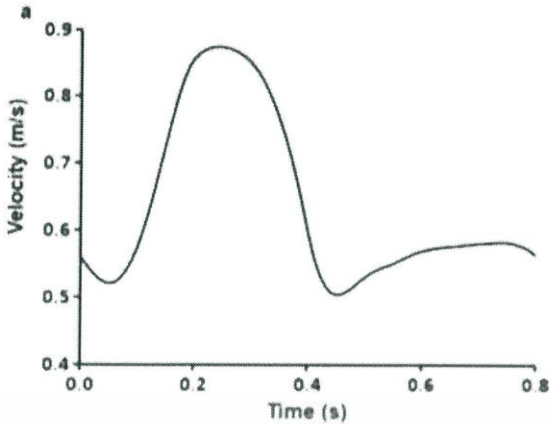


รูปที่ 1.2 โครงข่ายหลอดเลือดแดงในร่างกายมนุษย์ทั้ง 55 เส้น

1.4.4 งานวิจัย Hemodynamics in diabetic human aorta using computational fluid dynamics โดย Eunji Shin, Jung Joo Kim, Seonjoong Lee, Kyung Soo ko, Byoung Doo Rhee, Jin HanและNari Kim งานวิจัยนี้ศึกษาแบบจำลองทางคณิตศาสตร์ของหลอดเลือดแดงใหญ่ (Aorta) โดยในงานวิจัยมีค่าเงื่อนไขที่ขอบของหลอดเลือดแดงใหญ่ ที่เป็นค่าของความดันและค่าของความเร็ว



รูปที่ 1.3 ความดันที่หลอดเลือดแดงใหญ่ (Aorta)



รูปที่ 1.4 ความเร็วที่หลอดเลือดแดงใหญ่ (Aorta)

1.5 ขอบเขตการวิจัย

1.5.1 ใช้โปรแกรมคอมพิวเตอร์จำลองการไหลของเลือดโดยระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ 1 ทิศทาง

1.5.2 ผนังภายในหลอดเลือดมีลักษณะเรียบ

1.5.3 การไหลของเลือดในหลอดเลือดแดงใหญ่เอออร์ตาและหลอดเลือดแดงกลางอาร์เทอรีเป็นการไหลแบบนิวโตเนียน

1.6 ขั้นตอนของการศึกษา

1.6.1 ศึกษาหาข้อมูลเกี่ยวกับทฤษฎีและแนวคิดที่ใช้ในกาวิจัย

1.6.2 กำหนดพิกัดของหลอดเลือดแดงใหญ่และหลอดเลือดแดงกลางในร่างกายมนุษย์

1.6.3 ใช้โปรแกรมคอมพิวเตอร์สร้างแบบจำลองการไหลของเลือด และทำนายความดันของเลือดโดยระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์

1.6.4 วิเคราะห์พฤติกรรมของการไหลของเลือดว่าเงื่อนไขใดมีผลต่อความดัน

1.7 แผนการดำเนินงานวิจัย

ตารางที่ 1.2 แผนการดำเนินงาน

ขั้นตอนสำหรับงานวิจัย	ภาคเรียนที่ 1					ภาคเรียนที่ 2				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย	พ.ค.
1. ศึกษางานวิจัยที่เกี่ยวข้อง										
2. ศึกษาเกี่ยวกับเลือดและหลอดเลือดในร่างกายมนุษย์										

ตารางที่ 1.2 แผนการดำเนินงาน (ต่อ)

3. ศึกษาเกี่ยวกับกลศาสตร์ของไหล										
4. ศึกษาเกี่ยวกับระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์										
5. ศึกษาเกี่ยวกับโปรแกรม python										
6. จัดทำรูปเล่มนำเสนอภาคเรียนที่ 1										
7. เขียนไพธอน										
8. วิเคราะห์ผล										
9. จัดทำรูปเล่มนำเสนอภาคเรียนที่ 2										
10. พิสูจน์อักษร										

1.8 สมมติฐานของงานวิจัย

1.8.1 ค่าความดันที่ได้จากการทดลองเป็นฟังก์ชันคาบ

1.8.2 ค่าความดันที่ได้จากการทดลองอยู่ในช่วง 80 ถึง 120 มิลลิเมตรปรอท

1.8.3 ค่าความดันเฉลี่ยมีค่าลดลงเมื่อมีระยะห่างจากหัวใจมากขึ้น

1.9 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1.9.1 ทำให้ทราบลักษณะการไหลของเลือดตามเงื่อนไขที่ได้ตั้งไว้

1.9.2 สามารถทำนายความดันในหลอดเลือดแดงใหญ่เอออร์ตาและหลอดเลือดแดงกลางอาร์เทอรีได้ทั่วทั้งร่างกาย

1.9.3 เป็นองค์ความรู้และแนวทางในการศึกษาต่อสำหรับผู้ที่สนใจ

1.9.4 สามารถนำไปพัฒนาเป็นแบบจำลองของหลอดเลือดใน 3 มิติ เฉพาะจุด

บทที่ 2
เลือดและหลอดเลือดในร่างกายมนุษย์

สิ่งมีชีวิตต่างกับสิ่งไม่มีชีวิตตรงที่สิ่งมีชีวิตมีความสามารถในการใช้สสารและพลังงาน คือต้องการพลังงานในการดำรงชีวิต สามารถตอบสนองต่อสิ่งเร้า สามารถเคลื่อนที่ได้ มีการเปลี่ยนแปลงได้ เช่น การเติบโต เป็นต้น สามารถสืบพันธุ์ได้ และขับของเสียออกจากตัวเองได้ อย่างที่รู้กันว่าสิ่งมีชีวิตเริ่มเกิดหลังจากที่โลกเริ่มเย็นตัวลงหลังจากเกิดปรากฏการณ์บิกแบง (Big Bang) ทฤษฎีที่ยอมรับเกี่ยวกับการเกิดสิ่งมีชีวิตเริ่มแรกคือ สิ่งมีชีวิตเริ่มเกิดจากการทำปฏิกิริยาของสารเคมีซึ่งเกิดขึ้นในทะเล แล้วจึงเกิดเป็นสารประกอบพวกโปรตีน กรดอะมีโน และเอนไซม์ สะสมอยู่ในทะเลเป็นจำนวนมาก โดยมีสมมุติฐานที่ได้รับการสนับสนุนโดยการทดลองของ สแตนลีย์ มิลเลอร์ โดยมิลเลอร์ได้จำลองสภาวะที่เป็นระบบปิดหลังจากนั้นใส่ก๊าซมีเทน แอมโมเนีย ไฮโดรเจน และน้ำ ซึ่งเชื่อว่าสภาวะดังกล่าวเคยเกิดขึ้นในบรรยากาศของโลกในอดีต หลังจากนั้นให้ความร้อนและทำให้เกิดประกายไฟขึ้น แล้วรอหนึ่งสัปดาห์ มิลเลอร์พบว่าในชุดทดลองพบกรดอะมิโนและกรดอินทรีย์เกิดขึ้น สารประกอบอินทรีย์รวมตัวกันเป็นโมเลกุลอินทรีย์สารขนาดใหญ่ และวิวัฒนาการต่อไปจนเกิดเป็นโปรโตเซลล์ ซึ่งเป็นจุดเริ่มต้นของเซลล์ มีโครงสร้างของผนังเป็นไขมันและโปรตีน และเกิดการสันดาปภายในเซลล์ได้ หลังจากนั้นโปรโตเซลล์ ซึ่งเชื่อว่ามีอาร์เอ็นเอทำหน้าที่เป็นทั้งสารพันธุกรรม และเอนไซม์นั้นวิวัฒนาการเป็นเซลล์เริ่มแรกของสิ่งมีชีวิต โดยเมื่อเวลาผ่านไปสิ่งมีชีวิตก็ทำการปรับตัวให้เข้ากับสิ่งแวดล้อม เรียกว่าการมีวิวัฒนาการของสิ่งมีชีวิต วิวัฒนาการนี้ก่อให้เกิดความหลากหลายทางชีวภาพของสิ่งมีชีวิตเป็นจำนวนมากดังที่ปรากฏในปัจจุบัน ในปีค.ศ. 1735 คาโรลัส ลินเนียส ได้แบ่งสิ่งมีชีวิตออกเป็น 3 พวกคือ อาณาจักรพืช อาณาจักรสัตว์ และอาณาจักรแร่ธาตุ ต่อมานักวิทยาศาสตร์รุ่นหลังได้ศึกษาวิจัยและแบ่งจำแนกสิ่งมีชีวิตอย่างละเอียดขึ้น โดยจัดลำดับขั้นของหมวดหมู่สิ่งมีชีวิตออกเป็น 7 ลำดับ มีการจัดลำดับจากใหญ่ที่สุดถึงเล็กที่สุดดังนี้ อาณาจักร (kingdom) ไฟลัม (phylum) หรือดิวิชัน (division) ชั้น (class) อันดับ (order) วงศ์ (family) สกุล (genus) และสุดท้าย สปีชีส์ (species) ในปัจจุบันแบ่งสิ่งมีชีวิตออกเป็น 7 อาณาจักร ใน 3 โดเมน ได้แก่ อาณาจักรพืช อาณาจักรสัตว์ อาณาจักรฟังไจ อาณาจักรโพรทิสตา อาณาจักรโครมาลวีโอลาตา สังกัดโดเมนยูแคริโอต อาณาจักรอาร์คีแบคทีเรีย สังกัดโดเมนอาร์เคีย และอาณาจักรยูแบคทีเรีย สังกัดโดเมนโพราริโอต ชาลส์ ดาร์วิน และ แอลเฟรดรัสเซล วอลแลนซ์ นักธรรมชาติวิทยาชาวอังกฤษ ได้เสนอทฤษฎีวิวัฒนาการของสิ่งมีชีวิตบนโลก โดยกล่าวว่าวิวัฒนาการของสิ่งมีชีวิตเกิดจากการคัดเลือกตามธรรมชาติ คือสิ่งมีชีวิตภายในสปีชีส์เดียวกันนั้นมีความแตกต่างกันอยู่บ้าง ซึ่งความแตกต่างภายในสปีชีส์นี้ถูกเรียกว่าความผันแปร โดยความผันแปรนี้เป็นผลให้สิ่งมีชีวิตสามารถอยู่รอดได้ในสภาวะแวดล้อม คือเมื่อเกิดสภาวะที่ไม่ปกติ เช่น ภาวะแห้งแล้ง น้ำท่วม ฯ สิ่งมีชีวิตที่มีข้อได้เปรียบในสภาวะนั้นๆ มีความสามารถในการอยู่รอดมากกว่า เมื่อสิ่งมีชีวิตสายพันธุ์หนึ่งมีชีวิตได้นานก็ทำให้สามารถมีลูกหลายได้มากกว่าสิ่งมีชีวิตสายพันธุ์อื่นที่มีอายุสั้นกว่า และเมื่อเวลาผ่านไปสิ่งมีชีวิตสายพันธุ์ก็ยิ่งมีจำนวนมากขึ้นและเกิดเป็นสปีชีส์ใหม่ ในแต่ละอาณาจักรก็มีความแตกต่างกัน ทั้งระบบประสาท ระบบน้ำเหลือง การสืบพันธุ์ การขับถ่าย การหายใจ การย่อยอาหาร การไหลเวียนโลหิต และอีกมากมาย

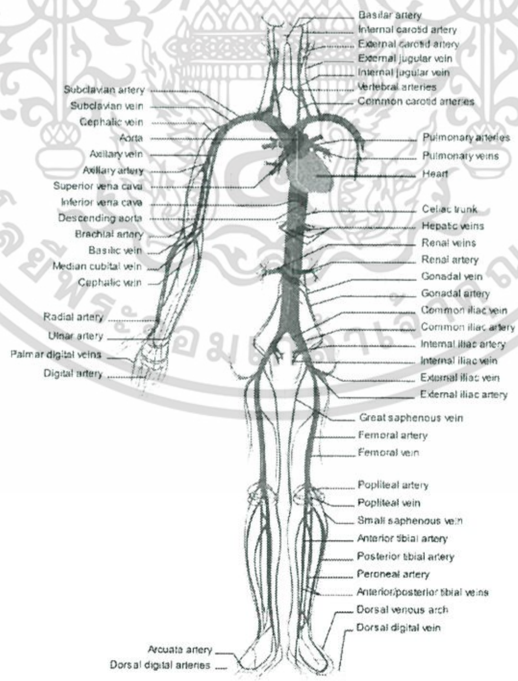
2.1 ระบบไหลเวียนโลหิต

ระบบไหลเวียนโลหิตของสิ่งมีชีวิตมี 2 แบบ คือ ระบบไหลเวียนโลหิตแบบปิด ระบบนี้พบได้ในสิ่งมีชีวิตที่มีประดูกสันหลังทุกชนิด รวมถึงหมึก ไส้เดือน และทากดูดเลือด โดยระบบไหลเวียนโลหิตนี้เลือดนั้นไหลแค่ในหลอดเลือด ไม่ไหลออกนอกหลอดเลือด ซึ่งต่างจากระบบไหลเวียนโลหิตแบบเปิด ระบบนี้พบได้ในสิ่งมีชีวิตที่ไม่มีกระดูกสันหลัง โดยระบบไหลเวียนโลหิตแบบเปิดนั้น เลือดไม่ไหลในหลอดเลือดตลอดเวลา แต่เป็นการไหลในหลอดเลือดช่วงแรก เป็นเลือดที่ออกจากหัวใจ แล้วหลังจากนั้นเลือดก็ไหลออกจากหลอดเลือดผ่านช่องว่างภายในลำตัว รวมถึงช่องว่างในแต่ละอวัยวะต่างๆ ซึ่งการไหลออกจากหลอดเลือดนั้นทำให้เลือดปนกับน้ำเหลือง เรียกว่า ฮีโมลิทพ์ (hemolymph) และช่องว่างภายในลำตัวเรียกว่า ฮีโมซีล (hemocoel)

ระบบไหลเวียนโลหิตในมนุษย์เป็นแบบปิด โดยระบบไหลเวียนโลหิตมีหน้าที่นำสารต่างๆ เช่น สารอาหาร เกลือแร่ ฮอร์โมน แอนติบอดี้ ก๊าซต่างๆ ไปส่งให้เซลล์ภายในอวัยวะต่างๆในร่างกาย และรับของเสียส่งออกนอกร่างกาย โดยลำเลียงสารที่ต้องการขนส่งไปตามหลอดเลือด ซึ่งเลือดประกอบด้วยส่วนที่เป็นของเหลวเรียกว่าพลาสม่า ทำหน้าที่ลำเลียงสารอาหาร เกลือแร่ ฮอร์โมน แอนติบอดี้ ก๊าซต่างๆ ไปยังเซลล์ต่างๆในร่างกาย และรับของเสียจากเซลล์เพื่อส่งกลับไปยังหัวใจ โดยมีการกำจัดของเสียทิ้งระหว่างทางโดยอวัยวะต่างๆ

ระบบไหลเวียนโลหิตเกิดขึ้นจากการบีบตัวของหัวใจทำให้มีแรงดันพอสำหรับการส่งเลือดไปที่ปอดเพื่อแลกเปลี่ยนแก๊สออกซิเจนกับแก๊สคาร์บอนไดออกไซด์ แล้วกลับเข้าสู่หัวใจเพื่อส่งไปหล่อเลี้ยงยังส่วนต่างๆของร่างกาย และสุดท้ายเลือดก็ทำการไหลเวียนกลับเข้าสู่หัวใจอีกครั้ง เป็นเช่นนี้เรื่อยไป

ระบบไหลเวียนโลหิต มีอวัยวะสำคัญที่เกี่ยวข้องได้แก่ เลือด หลอดเลือดและหัวใจ



รูปที่ 2.1 ระบบไหลเวียน

2.1.1 เลือด

เลือด คือ ของเหลวที่ไหลเวียนอยู่ภายในหลอดเลือดทั่วร่างกาย โดยแต่ละคนมีเลือดในร่างกายไม่เท่ากัน โดยมีปริมาณเลือดอยู่ 70 ซีซี ต่อน้ำหนัก 1 กิโลกรัม ส่วนประกอบของเลือดมี 2 ส่วน คือ พลาสมาและเซลล์เม็ดเลือด โดยพลาสมาเป็นส่วนที่เป็นของเหลว มีอยู่ประมาณร้อยละ 55 ของปริมาณโลหิตทั้งหมด และอีกร้อยละ 45 ของปริมาณเลือดทั้งหมด เป็นส่วนที่เป็นของแข็ง ซึ่งก็คือเซลล์เม็ดเลือด

หน้าที่ของเลือด คือ การขนส่งสารอาหารต่างๆที่ได้จากการดูดซึมสารอาหารจากกระเพาะอาหารและลำไส้ เพื่อส่งไปยังเซลล์ต่างๆทั่วร่างกาย

2.1.2 หลอดเลือด

ในระบบไหลเวียนโลหิตของมนุษย์ต้องอาศัยหลอดเลือดเป็นตัวลำเลียงเลือดที่ได้จากหัวใจไปยังอวัยวะต่างๆ และลำเลียงเลือดจากอวัยวะต่างๆกลับไปยังหัวใจ โดยหลอดเลือดในร่างกายมนุษย์มี 3 ชนิด

2.1.2.1 หลอดเลือดแดง (Artery)

หลอดเลือดแดงเป็นหลอดเลือดที่ลำเลียงเลือดที่มีปริมาณแก๊สออกซิเจนสูง ที่ออกจากหัวใจไปยังอวัยวะต่างๆ โดยเลือดที่มีปริมาณแก๊สออกซิเจนสูง เรียกว่า เลือดแดง ยกเว้นหลอดเลือดแดงที่ลำเลียงเลือดที่มีปริมาณแก๊สคาร์บอนไดออกไซด์สูง ออกจากหัวใจไปยังปอด โดยเลือดที่มีปริมาณแก๊สคาร์บอนไดออกไซด์สูง เรียกว่า เลือดดำ ลักษณะของหลอดเลือดแดง คือ มีผนังหนา ยืดหยุ่นได้ดี และไม่มีลิ้นกั้นภายในหลอดเลือด หลอดเลือดแดงเป็นหลอดเลือดที่มีความแข็งแรงมากที่สุดจาก 3 ชนิดของหลอดเลือด เนื่องจากต้องทนแรงดันเลือดที่สูบฉีดจากหัวใจให้ได้ หลอดเลือดแดงมีทั้งหมด 3 ขนาด ได้แก่

2.1.2.1.1 หลอดเลือดแดงใหญ่เอออร์ตา (Aorta)

หลอดเลือดแดงใหญ่เอออร์ตา(Aorta) คือ หลอดเลือดแดงขนาดใหญ่ที่สุดในร่างกายมนุษย์ ทำหน้าที่ลำเลียงเลือดแดงที่ถูกสูบฉีดจากหัวใจห้องล่างซ้ายไปยังหลอดเลือดแดงขนาดเล็กลงไป

2.1.2.1.2 หลอดเลือดแดงกลางอาร์เทอรี (Artery)

หลอดเลือดแดงกลางอาร์เทอรี(Artery) คือ หลอดเลือดแดงที่ทำหน้าที่รับเลือดแดงต่อจากหลอดเลือดแดงใหญ่เออร์ตา(Aorta) เพื่อไปหล่อเลี้ยงส่วนต่างๆของร่างกาย

2.1.2.1.3 หลอดเลือดแดงเล็กอาร์เทอริโอล (Arteriole)

หลอดเลือดแดงเล็กอาร์เทอริโอล (Arteriole) คือ หลอดเลือดที่ทำหน้าที่บังคับการไหลของเลือดเพื่อให้ไปเลี้ยงอวัยวะได้ทั่วถึง

2.1.2.2 หลอดเลือดดำ (Vein)

หลอดเลือดดำเป็นหลอดเลือดที่ลำเลียงเลือดที่มีปริมาณแก๊สคาร์บอนไดออกไซด์สูง เรียกว่า เลือดดำ จากส่วนต่างๆของร่างกายกลับเข้าสู่หัวใจ ยกเว้นหลอดเลือดดำที่ลำเลียงเลือดที่มีปริมาณแก๊สออกซิเจนสูง จากการฟอกที่ปอดกลับไปยังหัวใจ ลักษณะของหลอดเลือดดำ มีผนังบางกว่าหลอดเลือดแดงและมีลิ้นกั้นเป็นระยะๆภายในหลอดเลือด เพื่อป้องกันการไหลย้อนกลับของเลือด หลอดเลือดดำมีทั้งหมด 3 ขนาด ได้แก่

2.1.2.2.1 หลอดเลือดดำใหญ่เวนาคาวา (Vena cava)

หลอดเลือดดำใหญ่เวนาคาวา(Vena cava) คือ หลอดเลือดดำที่มีหน้าที่รับเลือดดำต่อจากหลอดเลือดดำกลางเวน(Vein) เพื่อส่งยังหัวใจห้องบนขวา

2.1.2.2.2 หลอดเลือดดำกลางเวน (Vein)

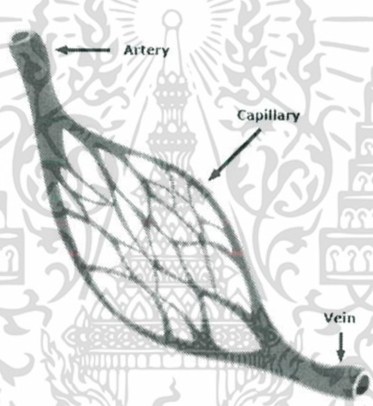
หลอดเลือดดำกลางเวน(Vein) คือ หลอดเลือดดำที่มีหน้าที่ขนส่งเลือดที่มีออกซิเจนต่ำจากเนื้อเยื่อกลับเข้าสู่หัวใจ

2.1.2.2.3 หลอดเลือดดำเล็กเวนูล (Venule)

หลอดเลือดดำเล็กเวนูล(Venule) คือ หลอดเลือดดำที่รับเลือดดำจากเนื้อเยื่อต่างๆในร่างกาย

2.1.2.3 หลอดเลือดฝอย (Capillary)

หลอดเลือดฝอยเป็นหลอดเลือดที่เชื่อมต่อระหว่างหลอดเลือดแดงขนาดเล็กอาร์เทอริโอล (Arteriole) และหลอดเลือดดำขนาดเล็กเวนูล (Venule) โดยมีการแลกเปลี่ยนสารอาหาร แก๊สออกซิเจน น้ำและของเสียต่างๆเมื่อเลือดไหลผ่านหลอดเลือดฝอย ลักษณะของหลอดเลือดฝอย เป็นการสานกันเป็นโครงตาข่าย มีผนังบางที่สุดในหลอดเลือดทั้ง 3 ชนิด เพื่อสะดวกในการแลกเปลี่ยนสารอาหาร แก๊สออกซิเจน น้ำและของเสียต่างๆ



รูปที่ 2.2 หลอดเลือดแดง หลอดเลือดดำ หลอดเลือดฝอย

2.1.3 หัวใจและลิ้นหัวใจ

ในระบบไหลเวียนโลหิตของมนุษย์ อวัยวะหัวใจถือว่าเป็นอวัยวะหลักของระบบ เนื่องจากหน้าที่สำคัญของหัวใจ คือ การ สูบฉีดเลือด เพื่อไปหล่อเลี้ยงส่วนต่างๆของร่างกาย ภายในหัวใจมีลักษณะเป็นโพรง แบ่งออกเป็น 4 ห้อง ได้ดัง หัวใจห้องบนขวา หัวใจห้องล่างขวา หัวใจห้องบนซ้ายและหัวใจห้องล่างซ้าย ส่วนที่เป็นห้องบน เรียกว่า เอเทรียม (Atrium) ห้องล่าง เรียกว่า เวนตริเคิล (Ventricle) โดยหัวใจแต่ละห้องนั้นทำหน้าที่แตกต่างกันไป

2.1.3.1 หัวใจห้องบนขวา

หัวใจห้องบนขวามีหน้าที่รับเลือดดำ ซึ่งเป็นเลือดที่มีปริมาณแก๊สคาร์บอนไดออกไซด์สูง จากหลอดเลือดดำใหญ่ 2 เส้น คือ หลอดเลือดเวนาคาวาด้านบน (Superior vena cava) หลอดเลือดเวนาคาวาด้านล่าง (Interior vena cava)

2.1.3.2 หัวใจห้องล่างขวา

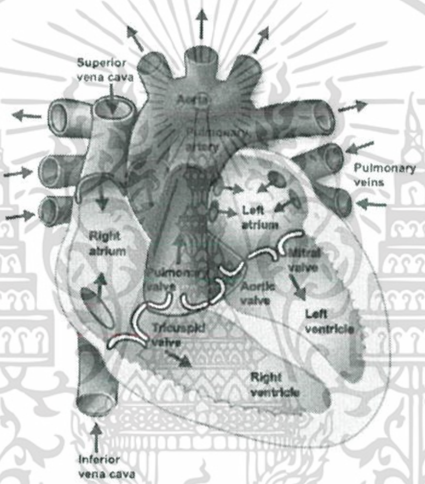
หัวใจห้องล่างขวามีหน้าที่รับเลือดดำจากหัวใจห้องบนขวา ผ่านทางลิ้นหัวใจและหลังจากนั้นในช่วงจังหวะการบีบตัวของหัวใจ เลือดดำถูกส่งไปยังปอด โดยผ่านลิ้นหัวใจอีกชิ้น และผ่านพัลโมนารีอาร์เทอรี (Pulmonary artery)

2.1.3.3 หัวใจห้องบนซ้าย

หัวใจห้องบนซ้ายมีหน้าที่รับเลือดแดงที่ผ่านการฟอกจากปอด ซึ่งเป็นเลือดที่มีปริมาณแก๊สออกซิเจนสูง

2.1.3.4 หัวใจห้องล่างซ้าย

หัวใจห้องล่างซ้ายมีหน้าที่รับเลือดแดงจากหัวใจห้องบนซ้าย ผ่านลิ้นหัวใจและหลังจากนั้นในช่วงจังหวะการบีบตัวของหัวใจ เลือดแดงถูกลำเลียงผ่านลิ้นหัวใจอีกชิ้น แล้วเข้าสู่หลอดเลือดแดงใหญ่เออร์ตา (Aorta) เพื่อไปหล่อเลี้ยงส่วนต่างๆของร่างกาย



รูปที่ 2.3 การทำงานของหัวใจ

2.1.3.5 ลิ้นหัวใจ

ลิ้นหัวใจ คือ แผ่นพังผืดที่สามารถทนแรงดันเมื่อหัวใจบีบตัวได้ ลิ้นหัวใจทำหน้าที่ป้องกันการไหลย้อนกลับของเลือด เมื่อเลือดออกจากหัวใจห้องต่างๆ โดยลิ้นหัวใจนั้นแบ่งเป็น 2 ประเภทและมี 4 ตำแหน่งภายในหัวใจ

2.1.3.5.1 ลิ้นเอตริโอเวนตริคูลา (Atrioventricular valve)

ลิ้นหัวใจประเภทนี้อยู่ระหว่างหัวใจห้องบนและหัวใจห้องล่าง มีอยู่ 2 แห่ง คือ ลิ้นหัวใจที่กั้นระหว่างหัวใจห้องบนซ้ายและหัวใจห้องล่างซ้าย เรียกว่า ลิ้นไบคัสปิด (Bicuspid valve)หรือลิ้นไมทรัล (Mitral valve) เปิดเมื่อถึงช่วงจังหวะหัวใจคลายตัว อีกตำแหน่ง คือ ลิ้นกั้นหัวใจที่กั้นระหว่างหัวใจห้องบนขวาและหัวใจห้องล่างขวา เรียกว่า ลิ้นไตรคัสปิด (Tricuspid valve) มีกลไลการทำงานเหมือนลิ้นไบคัสปิด (Bicuspid valve) แต่แตกต่างกันที่ลิ้นไบคัสปิด (Bicuspid valve) เป็นลิ้นหัวใจที่มี 2 แฉก ส่วนลิ้นไตรคัสปิด (Tricuspid valve) เป็นลิ้นหัวใจที่มี 3 แฉก

2.1.3.5.2 ลิ้นเซมิลูนนาร์ (Semilunar valve)

ลิ้นหัวใจประเภทนี้นั้นอยู่ระหว่างหัวใจและหลอดเลือด มีอยู่ 2 แห่ง คือ ลิ้นหัวใจที่กั้นระหว่างหัวใจห้องล่างซ้ายและหลอดเลือดแดงใหญ่เอออร์ตา (Aorta) เรียกว่า ลิ้นหัวใจเอออร์ติก (Aortic) และอีกตำแหน่ง คือ ลิ้นหัวใจที่กั้นอยู่ระหว่างหัวใจห้องขวาล่างและหลอดเลือดพัลโมนารี อาร์เทอรี (Pulmonary artery) เรียกว่า ลิ้นพัลโมนารี (Pulmonary valve)

2.2 คุณสมบัติของเลือด

2.2.1 การอัดตัวได้ของเลือด

เลือดเป็นของไหลที่มีการเปลี่ยนแปลงของค่าความหนาแน่นน้อยมากเมื่อมีการเปลี่ยนแปลงความดัน อาจเรียกได้ว่าเป็นของไหลที่มีค่าความหนาแน่นคงตัว ซึ่งเป็นของไหลที่อัดตัวไม่ได้(Incompressible flow) โดยค่าความหนาแน่นของเลือด มีค่าเท่ากับ 1060 กิโลกรัมต่อลูกบาศก์เมตร

2.2.2 ลักษณะการไหลแบบนอนนิวโตเนียน (Non-newtonian)และการไหลแบบนิวโตเนียน (Newtonian)

เลือดยังมีคุณสมบัติเป็นของไหลแบบนอนนิวโตเนียน (Non-newtonian fluid) ซึ่งก็คือของไหลที่มีค่าความหนืดไม่คงที่ มีการเปลี่ยนแปลงที่ขึ้นอยู่กับอัตราเฉือน หรือเรียกได้ว่าเลือดมีสมบัติเป็นเชียร์ตินนิง (shear-thinning) คือ ของไหลที่มีค่าความหนืดลดลงเมื่อเพิ่มอัตราเฉือน แต่เลือดที่มีคุณสมบัติเป็นของไหลแบบนอนนิวโตเนียน (Non-newtonian fluid) นั้น มีแค่ในหลอดเลือดแดงเล็กอาร์เทอริโอล (Arteriole) หลอดเลือดดำเล็กเวนูล (Venule) และหลอดเลือดฝอย (Capillary)เท่านั้น แต่ในหลอดเลือดแดงใหญ่เอออร์ตา (Aorta) หลอดเลือดแดงกลางอาร์เทอรี (Artery) หลอดเลือดดำใหญ่เวนาคาวา (Vena cava) และหลอดเลือดดำกลางเวน (Vein) สามารถมองเป็นของไหลแบบนิวโตเนียนได้ เนื่องจากของไหลในหลอดเลือดที่กล่าวไปข้างต้นมีค่าการเปลี่ยนแปลงของความหนืดน้อยมาก เมื่อเพิ่มอัตราเฉือน จึงสามารถมองเป็นของไหลแบบนิวโตเนียนได้ และของไหลที่เป็นของไหลแบบนิวโตเนียนสามารถเพิกเฉยต่อความเป็นของไหลประเภทยืดหยุ่นตัวได้ ที่อุณหภูมิร่างกาย 37 องศาเซลซียส มีค่าความหนืดสัมบูรณ์หรือความหนืดไดนามิก (Dynamic viscosity) เท่ากับ 2.78 มิลลิปาสคาล วินาที และมีค่าความหนืดจลน์หรือค่าความหนืดไคนีมาติก (Kinematic viscosity) เท่ากับ 2.65 ตารางมิลลิเมตรต่อวินาที

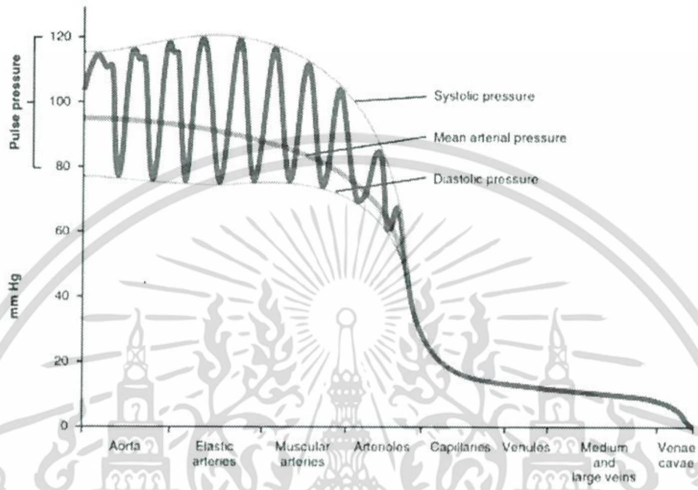
2.3 พลศาสตร์ของเลือด

บทนี้กล่าวถึงลักษณะทางกายภาพภายในระบบไหลเวียนโลหิต โดยเน้นความสัมพันธ์ระหว่างคุณลักษณะต่างๆที่ก่อให้เกิดการรักษาสมดุลของสิ่งต่างๆในระบบ ได้แก่ ความดัน ความต้านทาน ปริมาตรการไหล ลักษณะการไหลของเลือดและความยืดหยุ่นตัวได้ของของไหล

2.3.1 ความดันเลือด

ความดันเลือดวัดที่หลอดเลือดแดง เป็นความดันที่เกิดจากแรงที่เลือดกระทำต่อผนังเลือด ถ้าเรียกว่า “ความดันเลือด” แบบปกติ ไม่เจาะจง นั้นหมายถึงความดันเลือดแดงที่ไหลทั่วร่างกายระหว่างหัวใจเต้นแต่ละครั้ง ความดันเลือดแปรผันตามช่วงการบีบของหัวใจ คือ ความดันสูงสุด(ช่วงการบีบตัวของหัวใจ) และความดันต่ำสุด(ช่วงการคลายตัวของหัวใจ) ดังนั้นค่าความดันที่เป็นมาตรฐานจึงมีอยู่ 2 ค่า คือ 120/80 หมายความว่า 120 คือ ค่าความดันในช่วงการบีบตัวของหัวใจ และ 80 คือ ค่าความดันในช่วงการคลายตัวของหัวใจ โดยค่าความดันในช่วงการบีบตัวของหัวใจมีอีกชื่อเรียกว่า ความดันซีสโตลิค

(Systolic pressure) และค่าความดันในช่วงการคลายตัวของหัวใจมีอีกชื่อเรียกว่า ความดันไดแอสโตลิค (Diastolic pressure) ผลต่างของความดันเลือดเฉลี่ยนี้เป็นผลให้เกิดการไหลของเลือด โดยอัตราการไหลของเลือดเฉลี่ยขึ้นอยู่กับทั้งความดันเลือดและความต้านทานการไหล โดยเมื่อเลือดไหลเวียนเคลื่อนที่ไกลหัวใจผ่านหลอดเลือด ความดันเลือดมีค่าลดลง เนื่องจากการสูญเสียพลังงานกับความหนืด ความดันเลือดเฉลี่ยจึงลดลงตลอดทั้งการไหลเวียนเลือด โดยส่วนมากนั้นเป็นการลดลงในหลอดเลือดแดงกลางอาร์เทอรี (artery) และหลอดเลือดแดงเล็กอาร์เทอริโอล (arteriole)



รูปที่ 2.4 กราฟแสดงค่าความดัน

2.3.2 ความต้านทานการไหลของเลือด

ความต้านทานการไหลของเลือดเกิดขึ้นภายในหลอดเลือด โดยปัจจัยความต้านทานของไหลมีอยู่ 3 ประการ ได้แก่ ขนาดของหลอดเลือด ความหนืดของเลือดและอุณหภูมิของเลือด

2.3.2.1 ขนาดของหลอดเลือด

ความต้านทานที่เกิดขึ้นส่วนใหญ่ในร่างกายเกิดขึ้นที่หลอดเลือดแดง(Artery) เนื่องจากการทำงานของกล้ามเนื้อในชั้นผนังของหลอดเลือดแดงสามารถหดตัวเล็กลงได้มากถึง 4 เท่า ซึ่งทำให้ส่งผลถึงความต้านทานที่เพิ่มขึ้นและทำให้ปริมาตรการไหลลดลง

2.3.2.2 ความหนืดของเลือด

ความหนืดในหลอดเลือดมีผลต่อปริมาตรการไหลของเลือด ซึ่งโดยปกติแล้วความหนืดของเลือดนั้นอยู่ที่ประมาณ 3.6 เท่าของความหนืดของน้ำ แต่ความหนืดของพลาสมามีค่าประมาณ 1.6 เท่าของความหนืดของน้ำ ดังนั้นพลาสมาสามารถไหลไปตามท่อได้ง่ายกว่าเนื่องจากมีค่าความหนืดน้อยกว่า

ตัวการหลักที่ทำให้เกิดแรงเสียดทานของเลือด คือ เม็ดเลือดต่างๆที่อยู่ในเลือด ทำให้เกิดทั้งแรงเสียดทานระหว่างเม็ดเลือดด้วยกันเอง และแรงเสียดทานระหว่างเม็ดเลือดกับผนังเลือด ดังนั้นปริมาณเม็ดเลือดแดงซึ่งเป็นเม็ดเลือดที่มีมากที่สุดในร่างกายจึงเป็นปัจจัยหลักที่มีผลต่อความหนืดของเลือด ค่าฮีมาโทคริต(Hematocrit) คือ ค่าสัดส่วนของเม็ดเลือดต่อปริมาตรเลือด ซึ่งค่าฮีมาโทคริตคือค่าที่สามารถบ่งชี้ค่าความหนืดของเลือดได้ ถ้าค่าฮีมาโทคริตต่ำนั้นทำให้มีค่าความหนืดลดลงและความต้านทานการไหลก็

ลดลงด้วยเช่นกัน ในสภาวะนี้พบการเพิ่มขึ้นของค่าปริมาตรการไหลของเลือด ถ้าค่าฮีมาโทคริตสูงนั้นทำให้มีค่าความหนืดเพิ่มขึ้นแล้วความต้านทานการไหลก็เพิ่มขึ้นด้วยเช่นกัน ในสภาวะนี้สามารถพบการเพิ่มขึ้นของความดันเลือด

2.3.2.3 อุณหภูมิของเลือด

อุณหภูมิของเลือดสามารถส่งผลต่อการต้านทานการไหลได้ โดยทุกๆการลดลงของอุณหภูมิ 1 องศาเซลเซียสนั้นส่งผลให้ค่าความหนืดเพิ่มขึ้นได้ถึงร้อยละ 2 ถ้ามนุษย์ไปอยู่ในที่ที่มีอากาศหนาวเย็นเป็นระยะเวลานานนั้นส่งผลทำให้ร่างกายส่วนปลายเช่นปลายมือ ปลายเท้าอาจได้รับอันตราย เนื่องจากมีการหดตัวของหลอดเลือดและการเพิ่มขึ้นของค่าความหนืดของเลือด ทำให้มีการเพิ่มขึ้นของค่าความต้านทานการไหล ทั้งหมดที่กล่าวมาส่งผลให้ปริมาตรการไหลของเลือดลดลง ทำให้เลือดไม่สามารถไปเลี้ยงได้ถึงส่วนปลายของร่างกาย

2.3.3 ปริมาตรการไหลของเลือด

คาร์ดิแอก เอาท์พุท (cardiac output ; CO) หรือปริมาตรเลือดที่ออกจากหัวใจต่อนาที โดยส่วนมากใช้หน่วยเป็น ลิตรต่อนาที โดยปัจจัยที่สามารถกำหนดปริมาตรเลือดที่ออกจากหัวใจต่อนาทีได้ ได้แก่ ปริมาตรเลือดที่ถูกสูบฉีดออกจากหัวใจในการบีบตัวหนึ่งครั้งหรือเรียกว่าสโตรก โวลูม (stroke volume ; SV) และ อัตราการเต้นของหัวใจในหนึ่งนาที (Heart rate ; HR) โดยทั้งสามอย่างที่กล่าวมาข้างต้นมีความสัมพันธ์กันดังนี้

$$CO = SV \cdot HR \tag{2.1}$$

โดยปริมาตรเลือดที่ถูกสูบฉีดออกจากหัวใจในการบีบตัวหนึ่งครั้งหรือเรียกว่าสโตรก โวลูม (stroke volume ; SV)ในมนุษย์มีค่าปกติอยู่ที่ 60-70 ลูกบาศก์เซนติเมตร และอัตราการเต้นของหัวใจในหนึ่งนาที (Heart rate ; HR)ในผู้ใหญ่ขณะพักมีค่าอยู่ที่ 70-100 ครั้งต่อนาที สมมติว่าปริมาตรเลือดที่ถูกสูบฉีดออกจากหัวใจในการบีบตัวหนึ่งครั้งหรือเรียกว่าสโตรก โวลูม (stroke volume ; SV) มีค่า 70 ลูกบาศก์เซนติเมตรและอัตราการเต้นของหัวใจ(Heart rate ; HR) มีค่า 72 ครั้งต่อนาที ทำให้ได้ค่าคาร์ดิแอก เอาท์พุท (cardiac output ; CO) หรือปริมาตรเลือดที่ออกจากหัวใจต่อนาที เท่ากับ 5,040 ลูกบาศก์เซนติเมตรต่อนาที หรือประมาณ 5 ลิตรต่อนาที

ค่าคาร์ดิแอก เอาท์พุท (cardiac output ; CO) หรือปริมาตรเลือดที่ออกจากหัวใจต่อนาทีและปริมาตรเลือดที่ถูกสูบฉีดออกจากหัวใจในการบีบตัวหนึ่งครั้งหรือเรียกว่าสโตรก โวลูม (stroke volume ; SV) นั้นเปลี่ยนไปตามการทำงานของหัวใจ นอกจากนี้ปริมาตรเลือดที่ถูกสูบฉีดออกจากหัวใจในการบีบตัวหนึ่งครั้งหรือเรียกว่าสโตรก โวลูม (stroke volume ; SV)ยังขึ้นอยู่กับความแรงในการบีบตัวของหัวใจล่างหรือความสามารถในการส่งเลือดของหัวใจห้องล่างและยังขึ้นกับการบีบตัวเพื่อไล่เลือดจากห้องบนเข้าสู่ห้องล่างด้วยเช่นกัน

2.3.4 ลักษณะการไหลของเลือด

ลักษณะการไหลของเลือดภายในหลอดเลือดมีอยู่ 2 ลักษณะ ได้แก่ การไหลของเลือดแบบราบเรียบและการไหลของเลือดแบบปั่นป่วน

2.3.4.1 การไหลของเลือดแบบราบเรียบ

การไหลแบบราบเรียบหรือการไหลแบบสม่ำเสมอ คือ รูปแบบการไหลที่อนุภาคของของไหลที่เคลื่อนที่อย่างมีระเบียบ ไม่มีการผสมกันระหว่างชั้นของไหล การไหลของเลือดแบบราบเรียบ นั้นเกิดขึ้นเมื่อเลือดไหลผ่านหลอดเลือดเป็นแนวตรงไปตามความยาวของท่อ โดยมองเลือดเป็นโมเลกุลที่เคลื่อนที่ในแนวขนานกันเป็นชั้นๆ โมเลกุลที่เคลื่อนที่ติดกับผนังของหลอดเลือดนั้นมีความเร็วในการเคลื่อนที่ช้าที่สุด ความเร็วมีค่าเพิ่มขึ้นเมื่อโมเลกุลอยู่ในชั้นถัดเข้ามาจนถึงศูนย์กลางของหลอดเลือด และมีค่าลดลงอีกทีเมื่อโมเลกุลอยู่ในชั้นที่ถัดออกไป

2.3.4.2 การไหลของเลือดแบบปั่นป่วน

การไหลแบบปั่นป่วนหรือการไหลแบบไม่เป็นระเบียบนั้นเกิดขึ้นเมื่อเลือดไหลไปตามหลอดเลือดด้วยความเร็วสูงมากหรือไหลด้วยความเร็วที่สูงเกินความเร็ววิกฤต สามารถเกิดของเมื่อเลือดไหลผ่านตำแหน่งที่มีการตีบแคบของหลอดเลือดหรือตำแหน่งที่มีการอุดตัน การไหลของเลือดแบบปั่นป่วนนี้สามารถทำให้มีความต้านทานการไหลเพิ่มขึ้นอย่างมาก

รูปที่ 2.5 ลักษณะการไหลของเลือด

2.3.5 ความสามารถในการยืดหยุ่นตัว

ความสามารถในการยืดหยุ่นตัวมี 2 ลักษณะ ได้แก่ วิสโคอีลาสติค (viscoelastic) และอีลาสติค (elastic)

2.3.5.1 วิสโคอีลาสติค (viscoelastic)

วิสโคอีลาสติค (viscoelastic) คือ สมบัติของวัตถุที่มีทั้งการยืดหยุ่นและมีความหนืดในตัว

2.3.5.2 อีลาสติค (elastic)

อีลาสติค (elastic) คือ สมบัติของวัตถุที่ยืดหยุ่นได้โดยไม่คำนึงถึงความหนืด

จากข้อมูลข้างต้นที่กล่าวมา ในวิทยานิพนธ์นี้นั้นทำการศึกษาเพียงหลอดเลือดแดงใหญ่เอออร์ตาและหลอดเลือดแดงกลางอาร์เทอรีเท่านั้น ได้มีการกำหนดให้เลือดมีสมบัติเป็นของไหลที่อัดตัวไม่ได้ (Incompressible flow) โดยค่าความหนาแน่นของเลือด มีค่าเท่ากับ 1060 กิโลกรัมต่อลูกบาศก์เมตร มีการไหลแบบนิวโตเนียน (Newtonian) โดยมีค่าความหนืดสัมบูรณ์หรือความหนืดไดนามิก (Dynamic viscosity) เท่ากับ 2.78 มิลลิปาสคาล วินาที และมีค่าความหนืดจลน์หรือค่าความหนืดไคนีมาติก (Kinematic viscosity) เท่ากับ 2.65 ตารางมิลลิเมตรต่อวินาที ที่อุณหภูมิร่างกาย 37 องศาเซลซียส ได้มีการกำหนดให้ค่าคาร์ดิแอก เอาท์พุท (cardiac output ; CO) หรือปริมาตรเลือดที่ออกจากหัวใจต่อนาทีมีค่าเท่ากับ 5 ลิตรต่อนาทีและมีความสามารถในการยืดหยุ่นตัวแบบอีลาสติค (elastic)

บทที่ 3
กลศาสตร์ของไหล

จากที่ได้ทราบคุณสมบัติของเลือดและหลอดเลือดจากบทที่ 2 แล้ว ในบทนี้กล่าวถึงคุณสมบัติเหล่านั้นว่ามีลักษณะและมีผลต่อพฤติกรรมการไหลอย่างไรโดยใช้ความรู้จากวิชากลศาสตร์ของไหล

3.1 คุณสมบัติของของไหล

3.1.1 ความสามารถในการบีบอัดตัวของของไหล

สสารทุกชนิดมีความยืดหยุ่น นั่นคือสสารสามารถขยายตัวหรือหดตัวภายใต้สภาวะที่ต่างกัน โดยความหนาแน่นเปลี่ยนแปลงไปเมื่อมีปัจจัยภายนอกมากระทำ เช่น อุณหภูมิ ความดัน เป็นต้น ความสามารถในการบีบอัดตัวนี้เรียกอีกอย่างหนึ่งว่าความสามารถในการเปลี่ยนแปลงปริมาตร ซึ่งเป็นคุณสมบัติเฉพาะตัวของของไหลแต่ละชนิด นอกจากการจำแนกของไหลด้วยความหนืดแล้วของไหลมีคุณสมบัติอีกอย่างหนึ่งที่ไว้แยกชนิดของของไหลคือความสามารถในการบีบอัดตัวของของไหล โดยทั่วไปสามารถแบ่งได้ 2 แบบคือ

3.1.1.1 ของไหลแบบอัดตัวได้

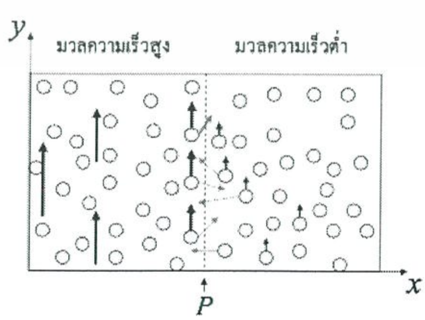
ของไหลแบบอัดตัวได้คือของไหลที่ความหนาแน่นไม่คงที่ ความหนาแน่นเปลี่ยนไปได้ในการไหลโดยขึ้นกับความดันที่มากระทำ ยิ่งมีความดันมากระทำมากความหนาแน่นก็ยิ่งมากขึ้นตามไปด้วย

3.1.1.2 ของไหลแบบอัดตัวไม่ได้

ของไหลแบบอัดตัวไม่ได้คือของไหลที่มีความหนาแน่นคงที่ตลอดการไหล โดยของเหลวเป็นของไหลที่ไม่สามารถอัดตัวได้ โดยสามารถอัดตัวได้ถ้าใช้ความดันที่สูงมากๆ จึงพิจารณาว่าของเหลวเป็นของไหลที่อัดตัวไม่ได้

3.1.2 ความหนืด

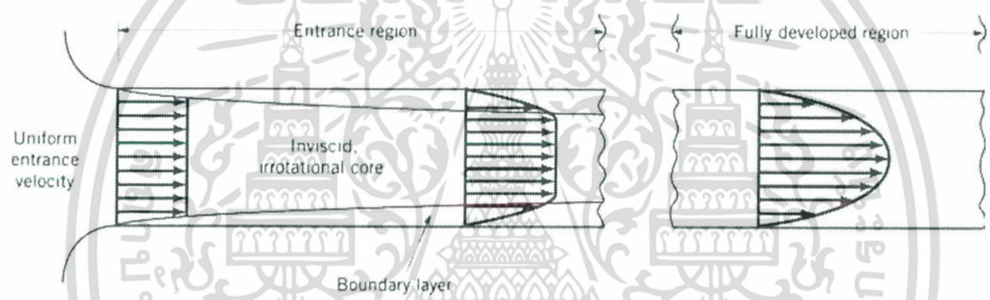
ความหนืดเป็นคุณสมบัติความต้านการไหลของของไหล ซึ่งทำให้เกิดการเปลี่ยนรูปจากการกระทำของความเค้นเฉือน



รูปที่ 3.1 แสดงการเปรียบเทียบการไหลของของไหลที่ความเร็วต่างกันและมีการเคลื่อนที่แบบกระจัดกระจาย

จากรูปที่ 3.1 แสดงให้เห็นว่าของไหลมีการเสียพลังงานให้กับการเคลื่อนที่แบบกระจัดกระจายและของไหลฝั่งที่มีความเร็วสูงมีการเสียพลังงานให้กับฝั่งที่มีความเร็วต่ำตรงส่วนหน้าตัด P ที่ของไหลความเร็วสูงสัมผัสกับของไหลความเร็วต่ำ และในกรณีที่ของไหลสัมผัสกับผนังก็ทำให้เสียพลังงานให้กับผนังด้วย ทำให้ความเร็วลดลง คือเกิดความเค้นเฉือนระหว่างมวลความเร็วสูงและมวลความเร็วต่ำ แรงเสียดทานที่ทำให้เกิดความเค้นเฉือนนี้เรียกว่าแรงหนืด โดยเกิดขึ้นทั่วพื้นที่หน้าตัดที่ของไหลสัมผัสกับสิ่งที่มีความเร็วต่างกัน ยิ่งของไหลมีความหนืดมากขึ้น ความสามารถในการเปลี่ยนรูปก็ยิ่งน้อยลง ความหนืดมีหน่วยเป็น N/m^2 หรือ $Pa \cdot s$

วิทยานิพนธ์นี้ศึกษาการไหลขณะที่ของไหลมีการเคลื่อนที่อยู่ในท่อกลม โดยความต้านทานการเคลื่อนที่ของของไหลเกิดขึ้นเมื่อของไหลไหลผ่านผิวเรียบผิวเรียบและมีส่วนที่สัมผัสกับผิวเรียบนั้น โดยที่ผิวเรียบมีความเร็วน้อยกว่าของไหลมากๆ ในกรณีศึกษานี้คือให้ความเร็วเป็นศูนย์ และมีการเปลี่ยนแปลงรูปร่างของการไหลเนื่องจากการเคลื่อนที่ของของไหลและการเคลื่อนที่สัมพัทธ์ระหว่างชั้นที่แตกต่างกัน แรงหนืดนั้นกระทำกับแต่ละชั้นไม่เท่ากันตามแนวชั้นของชั้นนั้นๆ ซึ่งแรงหนืดนั้นขึ้นกับพื้นที่ของชั้น ความเร็วของชั้น และระยะทางของชั้นจากผิว



รูปที่ 3.2 แสดงการเปลี่ยนรูปร่างของการไหล

3.1.3 ลักษณะการไหลตามความหนืด

ลักษณะการไหลของของไหลโดยทั่วไปสามารถแบ่งได้ 2 แบบตามความหนืดของของไหลเมื่อพิจารณาที่อุณหภูมิๆหนึ่ง คือ ของไหลแบบนิวโตเนียนและของไหลแบบนอนนิวโตเนียน

3.1.3.1 ของไหลแบบนิวโตเนียน (Newtonian fluid)

ของไหลแบบนิวโตเนียนคือของไหลที่ลักษณะของการไหลเป็นไปตามสันนิษฐานของนิวตัน คือค่าความหนืดของของไหลเป็นค่าคงที่ที่อุณหภูมิๆหนึ่ง ไม่เปลี่ยนแปลงตามอัตราเฉือนหรือความเร็วในการกวน โดยเป็นของไหลที่ความเค้นเฉือนแปรผันแปรผันเป็นเส้นตรงกับความเครียดเฉือน

3.1.3.2 ของไหลแบบนอนนิวโตเนียน (Non-newtonian fluid)

ของไหลแบบนอนนิวโตเนียนคือของไหลที่ลักษณะของการไหลไม่เป็นไปตามสันนิษฐานของนิวตัน คือค่าความหนืดของของไหลเป็นค่าไม่คงที่ที่อุณหภูมิๆหนึ่ง การเปลี่ยนแปลงขึ้นอยู่กับอัตราเฉือนหรือความเร็วในการกวน โดยเป็นของไหลที่ความเค้นเฉือนไม่แปรผันแปรผันเป็นเส้นตรงกับความเครียดเฉือน ลักษณะการไหลแบบนี้แบ่งเป็น 4 แบบ ได้แก่

1 สูโดพลาสติก (Pseudoplastic)

สูโดพลาสติกคือของไหลที่มีค่าความหนืดลดลงเมื่อเพิ่มอัตราเฉือน หรือยิ่งกวนเร็วยิ่งไหลงาย พฤติกรรมแบบนี้แสดงสมบัติเป็นเชียร์ตินนิง (shearthinning) เชน น้ำผลไม้เขมขน กาวน้ำใส สารชวยแขวนตะกอน สารละลายพอลิเมอร์ที่ได้จากธรรมชาติ สารละลายพอลิเมอร์สังเคราะห์ เปนต้น

2 ไดลาแตนท์ (Dilatant)

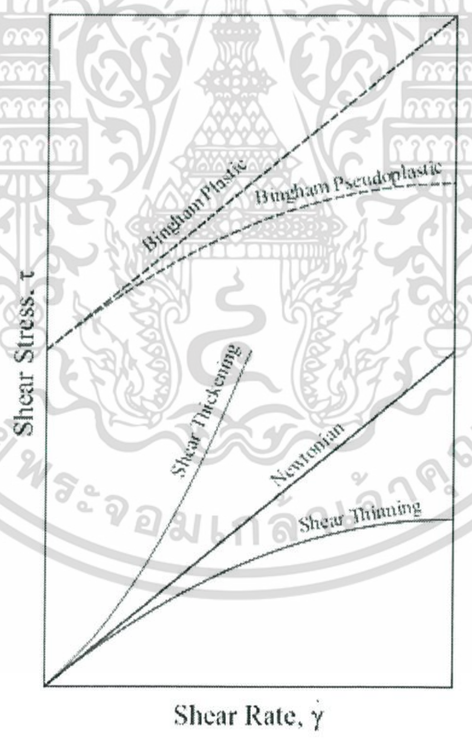
ไดลาแตนท์คือของไหลที่มีค่าความหนืดเพิ่มขึ้นเมื่อเพิ่มอัตราเฉือน หรือยิ่งกวนเร็วยิ่งหนืด พฤติกรรมแบบนี้แสดงสมบัติเปนเชียร์ติกเคนนิง (shearthickening) เช่น น้ำดิน น้ำแปง เปนตน

3 บิงแฮมพลาสติก (Bingham Plastic)

บิงแฮมพลาสติกคือของไหลที่ต้องมีแรงมากระทำสูงพอจึงสามารถเกิดลักษณะการไหลแบบนิวโตเนียน เชน ยาสีฟัน นมช็อกโกแลต เปนต้น

4 พลาสติก (Plastic)

พลาสติกคือของไหลที่ต้องมีแรงกระทำสูงพอที่สามารถเอาชนะคาความเคนที่จุดครากได้ จึงสามารถเริ่มไหลได และการไหลเป็นแบบสูโดพลาสติก เช่น ซอสมะเขือเทศ สี ดินเหนียว มายองเนส ยาน้ำ แขวนตะกอน เป็นตน



รูปที่ 3.3 กราฟแสดงความสัมพันธ์ระหว่างความเค้นเฉือนกับอัตราเฉือนของของไหลรูปแบบต่างๆ

จากรูปที่ 3.3 ค่าความขันของกราฟระหว่างความเค้นเฉือนกับอัตราความเครียดเฉือนคือความหนืดปรากฏ μ_{ab}

3.1.4 ลักษณะการไหลพิจารณาจากเลขเรย์โนลด์

ลักษณะของการไหลยังสามารถแบ่งได้เป็น 3 แบบตามช่วงของค่าของเลขเรย์โนลด์ ดังนี้

3.1.4.1 การไหลแบบราบเรียบ

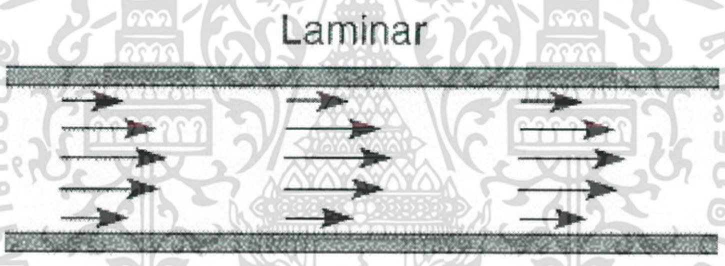
การไหลแบบราบเรียบหรือการไหลแบบลามินาร์ (laminar flow) เป็นการไหลที่โมเลกุลของไหลไหลแบบราบเรียบ การไหลแบบนี้เกิดขึ้นเมื่อของเส้นทางการไหลไม่มีสิ่งกีดขวาง ไหลมีความหนืดสูงๆและไหลด้วยความเร็วต่ำๆ มีค่าเลขของเรย์โนลด์อยู่ในช่วงน้อยกว่า 2300

3.1.4.2 การไหลแบบผสม

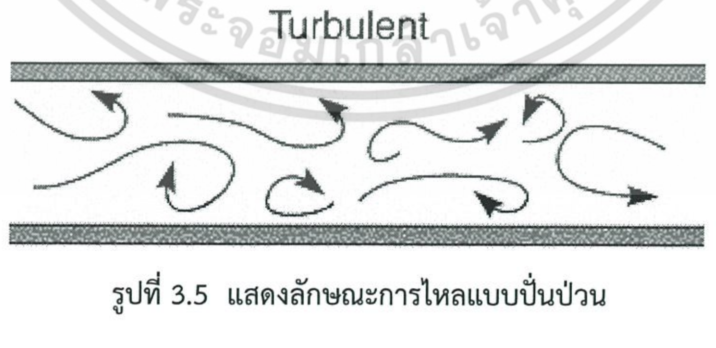
การไหลแบบผสมหรือการไหลแบบทรานซิชัน (transition flow) เป็นการไหลที่โมเลกุลของไหลที่ไหลแบบราบเรียบปนกับแบบปั่นป่วน มีค่าเลขของเรย์โนลด์อยู่ในช่วง 2300-4000

3.1.4.3 การไหลแบบปั่นป่วน

การไหลแบบปั่นป่วนหรือการไหลแบบเทอร์บูเลนท์ (turbulent flow) เป็นการไหลที่โมเลกุลของไหล ไหลแบบไม่เป็นระเบียบ ความเร็วของอนุภาคแตกต่างกันทั้งขนาดและทิศทาง การไหลแบบนี้เกิดขึ้นเมื่อเส้นทางการไหลมีสิ่งกีดขวาง ของไหลมีความหนืดต่ำและไหลด้วยความเร็วสูงๆ มีค่าเลขของเรย์โนลด์อยู่ในช่วงมากกว่า 4000



รูปที่ 3.4 แสดงลักษณะการไหลแบบราบเรียบ



รูปที่ 3.5 แสดงลักษณะการไหลแบบปั่นป่วน

ค่าเลขเรย์โนลด์เป็นเลขที่บอกลักษณะของการไหล สามารถคำนวณได้จากสูตร

$$\mathrm{Re} = \frac{\rho \bar{v} D}{\mu} = \frac{\bar{v} D}{\upsilon} = \frac{QD}{\upsilon A} \tag{3.1}$$

โดยที่

Re คือ เลขของเรย์โนลด์

ρ คือ ความหนาแน่นของของเหลว $\left(kg/m^3\right)$

$\bar{v}$ คือ ความเร็วเฉลี่ยของของไหล (m/s)

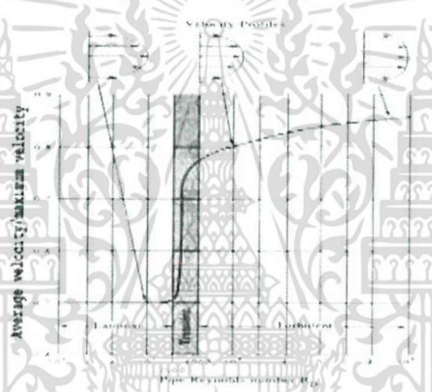
D คือ ขนาดเส้นผ่านศูนย์กลางของท่อ (m)

μ คือ ความหนืดสัมบูรณ์ หรือความหนืดแบบไดนามิก(dynamic viscosity) $(Pa \cdot s)$

υ คือ ความหนืดจลน์ หรือความหนืดแบบไคนีมาติก(kinematic viscosity) $\left(m^2/s\right)$

Q คือ อัตราหารไหลของของไหล $\left(m^3/s\right)$

A คือ พื้นที่หน้าตัดของท่อ $\left(m^2\right)$



รูปที่ 3.6 แสดงความสัมพันธ์ระหว่างช่วงของเลขเรย์โนลด์และลักษณะการไหล

3.2 สมการควบคุมการไหลของของไหล

3.2.1 ทฤษฎีการถ่ายเทของเรย์โนลด์(Reynolds' transport theorem)

พิจารณาท่อซึ่งแสดงในรูปที่ 3.7 เป็นหลอดเลือด โดยกำหนดให้หลอดเลือดนั้นเป็นเส้นตรงตามแนวแกน x ในการหาสมการควบคุมการไหลของของไหลในหนึ่งมิติเริ่มจากทฤษฎีการถ่ายเทของเรย์โนลด์ โดยพิจารณาปริมาตรควบคุม V_t ที่มีขอบเขต $\partial \mathrm{V}_t$ และเวกเตอร์หนึ่งหน่วย $\bar{n}$ ซึ่งตั้งฉากกับพื้นที่หน้าตัด เมื่อ $f = f(t, \bar{x})$ เป็นฟังก์ชันที่มีความต่อเนื่อง

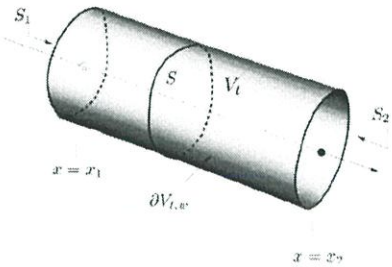
$$\frac{d}{dt}\int_{\mathrm{V}_t} f\, d\mathrm{V} = \int_{\mathrm{V}_t} \frac{\partial f}{\partial t} d\mathrm{V} + \int_{\partial \mathrm{V}_t} f\, \bar{u}_b \cdot \bar{n}\, d\sigma \tag{3.2}$$

โดยที่

$\bar{x}$ คือ เวกเตอร์ตำแหน่งที่พิกัด (x, y, z)

f คือ คุณสมบัติภายใน หรือ คุณสมบัติภายนอกต่อมวล

$\bar{u}_b$ คือ ความเร็วของขอบเขตของปริมาตร V_t



รูปที่ 3.7 แสดงหลอดเลือดที่เป็นเส้นตรงตามแนวแกน x

ขอบเขตของปริมาตร V_t ประกอบด้วยผนังหลอดเลือดแดง $\partial \mathrm{V}_{t,\,wall}$ และหน้าตัดที่ส่วนปลายทั้งสองข้าง S_1 และ S_2 โดยสมมติให้หน้าตัดทั้งสองนี้ตั้งฉากกับแกน ที่บนหน้าตัดทั้งสองนี้เวกเตอร์ความเร็ว $\bar{u}_b$ เป็นศูนย์ ขณะที่บน $\partial \mathrm{V}_{t,\,wall}$ ความเร็ว $\bar{u}_b$ เท่ากับกับความเร็วของผนังหลอดเลือดแดง $\bar{u}_{wall}$ ได้สมการคือ

$$\int_{\partial \mathrm{V}_t} f\,\bar{u}_b \cdot \bar{n}\, d\sigma = \int_{\partial \mathrm{V}_{t,wall}} f\,\bar{u}_{wall} \cdot \bar{n}\, d\sigma \tag{3.3}$$

สำหรับการหาสมการของกฎการอนุรักษ์ในหนึ่งมิติ ได้ทำการพิจารณาค่าเฉลี่ยของฟังก์ชัน f เขียนเป็น $\bar{f}$ ได้

$$\bar{f} = \frac{1}{\mathrm{A}} \int_{\mathrm{S}} f\, d\sigma \tag{3.4}$$

โดยที่

$\mathrm{A} = \mathrm{A}(x, t) = \int_{\mathrm{S}} d\sigma$ คือ พื้นที่ของหน้าตัด S

จากข้อมูลดังกล่าว เมื่อ x_1 และ x_2 เป็นพิกัดในแนวแกน x ของหน้าตัด S_1 และ S_2 โดย $x_2 > x_1$ ทำให้ได้ปริพันธ์ของปริมาตร

$$\frac{d}{dt} \int_{\mathrm{V}_t} f\, d\mathrm{V} = \int_{\mathrm{V}_t} \frac{\partial f}{\partial t} d\mathrm{V} + \int_{\partial \mathrm{V}_t} f\,\bar{u}_b \cdot \bar{n}\, d\sigma \tag{3.5}$$

กำหนดให้ x_1 และ x_2 เป็นอิสระต่อเวลา พจน์ฝั่งซ้ายของสมการ (3.1) เขียนได้ว่า

$$\frac{d}{dt} \int_{\mathrm{V}_t} f\, d\mathrm{V} = \int_{x_1}^{x_2} \frac{\partial}{\partial t} \left(\mathrm{A}\, \bar{f} \right) dx \tag{3.6}$$

ความเร็วสัมพัทธ์ระหว่างผนังหลอดเลือดกับของไหลคือ

$$\bar{w} = \bar{u}_{wall} - \bar{u}$$

หรือได้

$$\bar{u}_{wall} = \bar{w} + \bar{u}$$

โดยที่

$\bar{u} = u_1\bar{\mathrm{i}} + u_2\bar{\mathrm{j}} + u_3\bar{\mathrm{k}}$ คือ ความเร็วของของไหล

$\bar{u}_{wall}$ คือ ความเร็วของผนังหลอดเลือด

$\bar{w}$ คือ ความเร็วสัมพัทธ์ระหว่างผนังหลอดเลือดกับของไหล

เมื่อนำคุณสมบัติของความเร็วสัมพัทธ์จากด้านบนมาแทนค่าในสมการที่ (3.2) ได้สมการคือ

$$\int_{\partial \mathrm{V}_{t,wall}} f\,\bar{u}_{wall} \cdot \bar{n}\, d\sigma = \int_{\partial \mathrm{V}_{t,wall}} f\,\bar{w} \cdot \bar{n}\, d\sigma + \int_{\partial \mathrm{V}_{t,wall}} f\,\bar{u} \cdot \bar{n}\, d\sigma \qquad (3.7)$$

จากที่ทราบแล้วว่าขอบเขตของปริมาตร V_t ประกอบด้วยผนังหลอดเลือดแดง $\partial \mathrm{V}_{t,\,wall}$ และหน้าตัดที่ส่วนปลายทั้งของข้าง S_1 และ S_2 ทำให้ได้

$$\begin{aligned}\int_{\partial \mathrm{V}_{t,wall}} f\,\bar{u} \cdot \bar{n}\, d\sigma &= \int_{\partial \mathrm{V}_t} f\,\bar{u} \cdot \bar{n}\, d\sigma - \int_{\mathrm{S}_1} f\,\bar{u} \cdot \bar{n}\, d\sigma - \int_{\mathrm{S}_2} f\,\bar{u} \cdot \bar{n}\, d\sigma \\ &= \int_{\partial \mathrm{V}_t} f\,\bar{u} \cdot \bar{n}\, d\sigma + \int_{\mathrm{S}_1} f\,u_1\, d\sigma - \int_{\mathrm{S}_2} f\,u_1\, d\sigma\end{aligned}$$

จากทฤษฎีบทไดเวอร์เจนซ์ ถ้า W เป็นย่านทรงตันที่มีขอบเป็นพื้นผิวเรียบ และถ้า $\bar{F}$ เป็นสนามเวกเตอร์เรียบบนย่านเปิดซึ่งบรรจุ W และ S เมื่อ S ถูกกำหนดทิศพุ่งออก แล้ว

$$\int_{\mathrm{S}} \bar{F} \cdot d\bar{A} = \int_{\mathrm{W}} \nabla \cdot \bar{F} dV$$

เมื่อ u_1 คือตัวประกอบที่มีทิศตามแนวแกน x ของความเร็ว $\bar{u}$ ทำให้ได้ว่า

$$\int_{\partial \mathrm{V}_{t,wall}} f\,\bar{u} \cdot \bar{n}\, d\sigma = \int_{\mathrm{V}_t} \nabla \cdot (f\,\bar{u}) dV + \int_{\mathrm{S}_1} f\,u_1\, d\sigma - \int_{\mathrm{S}_2} f\,u_1\, d\sigma$$

เมื่อนำสมการที่ได้ไปแทนในสมการที่ (2.6)

$$\int_{\partial \mathrm{V}_{t,wall}} f\,\bar{u}_{wall} \cdot \bar{n}\, d\sigma = \int_{\partial \mathrm{V}_{t,wall}} f\,\bar{w} \cdot \bar{n}\, d\sigma + \int_{\mathrm{V}_t} \nabla \cdot (f\,\bar{u}) dV + \int_{\mathrm{S}_1} f\,u_1\, d\sigma - \int_{\mathrm{S}_2} f\,u_1\, d\sigma$$

เมื่อพิจารณาเป็นค่าเฉลี่ย

$$\int_{\partial V_{t,wall}} f\,\bar{u}_{wall}\cdot\bar{n}\,d\sigma = \int_{\partial V_{t,wall}} f\,\bar{w}\cdot\bar{n}\,d\sigma + \int_{V_t}\nabla\cdot(f\,\bar{u})dV + A(\overline{fu_1})\Big|_{S_1} - A(\overline{fu_1})\Big|_{S_2}$$

เมื่อค่า $x = x_1$ สำหรับย่าน S_1 และ $x = x_2$ สำหรับย่านS_2 แล้ว

$$\int_{\partial V_{t,wall}} f\,\bar{u}_{wall}\cdot\bar{n}\,d\sigma = \int_{\partial V_{t,wall}} f\,\bar{w}\cdot\bar{n}\,d\sigma + \int_{V_t}\nabla\cdot(f\,\bar{u})dV - \int_{x_1}^{x_2}\frac{\partial}{\partial x}\left[A(\overline{fu_1})\right]dx \tag{3.8}$$

เมื่อแทนสมการ (2.5) และ (2.7) ลงในสมการ (2.1) ทำให้ได้

$$\int_{x_1}^{x_2}\frac{\partial}{\partial t}\left(A\,\bar{f}\right)dx = \int_{V_t}\frac{\partial f}{\partial t}dV + \int_{\partial V_{t,wall}} f\,\bar{w}\cdot\bar{n}\,d\sigma + \int_{V_t}\nabla\cdot(f\,\bar{u})dV - \int_{x_1}^{x_2}\frac{\partial}{\partial x}\left[A(\overline{fu_1})\right]dx$$

ได้

$$\int_{x_1}^{x_2}\frac{\partial}{\partial t}\left(A\,\bar{f}\right)dx = \int_{x_1}^{x_2}\left(\int_S\frac{\partial f}{\partial t}d\sigma\right)dx + \int_{x_1}^{x_2}\left(\int_{\partial S} f\,\bar{w}\cdot\bar{n}\,d\gamma\right)dx + \int_{x_1}^{x_2}\left(\int_S\nabla\cdot(f\,\bar{u})d\sigma\right)dx - \int_{x_1}^{x_2}\frac{\partial}{\partial x}\left[A(\overline{fu_1})\right]dx$$

สมการเป็นจริงสำหรับพิกัด x_1, x_2 ใดๆของส่วนปลายหน้าตัดของหลอดเลือด รูปสุดท้ายของทฤษฎีการถ่ายเทในหนึ่งมิติสำหรับตัวแปรทั่วไป f คือ

$$\frac{\partial}{\partial t}\left(A\,\bar{f}\right) + \frac{\partial}{\partial x}\left[A(\overline{fu_1})\right] = \int_S\left[\frac{\partial f}{\partial t} + \nabla\cdot(f\,\bar{u})\right]d\sigma + \int_{\partial S} f\,\bar{w}\cdot\bar{n}\,d\gamma \tag{3.9}$$

สูตรดังกล่าวนี้ใช้วิเคราะห์ได้ทั้งในกรณีของของไหลอัดตัวได้และของไหลอัดตัวไม่ได้

ต่อไปเป็นการดำเนินการหาสมการควบคุมการไหลของของไหลโดยใช้กฎการอนุรักษ์มวลและโมเมนตัมมาช่วยในการดำเนินการ

3.2.2 สมการอนุรักษ์มวล

สำหรับสมการอนุรักษ์มวลในท่อที่มีความยืดหยุ่น แทน $f = \frac{m}{m} = 1$ ในสมการ (3.9) ทำให้ได้ว่า

$$\frac{\partial A}{\partial t} + \frac{\partial}{\partial x}\left(A\bar{u}_1\right) = \int_S\nabla\cdot\bar{u}\,d\sigma + \int_{\partial S}\bar{w}\cdot\bar{n}\,d\gamma$$

ในกรณีที่ของไหลเป็นของไหลแบบอัดตัวไม่ได้มีสมบัติ $\nabla\cdot\bar{u} = 0$ ทำให้ได้ว่า

$$\frac{\partial A}{\partial t} + \frac{\partial}{\partial x}\left(A\bar{u}_1\right) = \int_{\partial S}\bar{w}\cdot\bar{n}\,d\gamma \tag{3.10}$$

พจน์ทางขวาคือพจน์ที่แสดงถึงปริมาตรที่ไหลออกจากผนังของหลอดเลือดต่อหนึ่งหน่วยความยาวและหนึ่งหน่วยเวลา เมื่อกำหนดให้ไม่มีของไหลไหลผ่านผนังของหลอดเลือด($\bar{w}\cdot\bar{n} = 0$) แล้ว

$$\frac{\partial A}{\partial t}+\frac{\partial}{\partial x}\left(A\bar{u}_1\right)=0 \tag{3.11}$$

3.2.3 สมการอนุรักษ์โมเมนตัม

สำหรับสมการอนุรักษ์โมเมนตัม แทน $f=\frac{mu_1}{m}=u_1$ ในสมการ (3.9) และของไหลมีคุณสมบัติเป็นของไหลที่อัดตัวไม่ได้ ทำให้ได้

$$\frac{\partial}{\partial t}\left(\mathrm{A}\bar{u}_1\right)+\frac{\partial}{\partial x}A\overline{u_1^{\,2}}=\int_{\mathrm{S}}\left[\frac{\partial u_1}{\partial t}+u_1\nabla\cdot\bar{u}\right]d\sigma+\int_{\partial\mathrm{S}}u_1\,\bar{w}\cdot\bar{n}\,d\gamma \tag{3.12}$$

สำหรับการคิดคำนวณพจน์แรกทางด้านขวาในสมการที่ (2.11) ได้มีการนำสมดุลของโมเมนตัมสำหรับปริมาตรควบคุม V_t ซึ่งอยู่ในรูปมาใช้

$$\int_{\mathrm{V}_t}\left[\frac{\partial\rho\bar{u}}{\partial t}+u_1\nabla\cdot\rho\bar{u}\right]d\mathrm{V}=\int_{\mathrm{V}_t}\rho\bar{F}_{body}\,d\mathrm{V}+\int_{\partial\mathrm{V}_t}\bar{T}\cdot\bar{n}\,d\sigma \tag{3.13}$$

โดยที่

$\bar{F}_{body}$ คือ body force ต่อหนึ่งหน่วยปริมาตร

$\bar{T}$ คือ เทนเซอร์ความเค้นของโคชี (Cauchy stress tensor)

ρ คือ ความหนาแน่นของของไหลมีค่าคงที่

จากทฤษฎีบทไดเวอร์เจนซ์ นั้นได้

$$\int_{\mathrm{V}_t}\left[\frac{\partial\bar{u}}{\partial t}+u_1\nabla\cdot\bar{u}\right]d\mathrm{V}=\int_{\mathrm{V}_t}\bar{F}_{body}\,d\mathrm{V}+\frac{1}{\rho}\int_{\mathrm{V}_t}\nabla\cdot\bar{T}\,d\mathrm{V} \tag{3.14}$$

สำหรับพจน์เทนเซอร์ความเค้นของโคชี $\bar{T}$ สามารถเขียนในรูป

$$\bar{T}=-p\bar{I}+\bar{D} \tag{3.15}$$

โดยที่

p คือ ความดัน

$\bar{I}$ คือ เทนเซอร์เอกลักษณ์

$\bar{D}$ คือ ความเค้นเบี่ยงเบนที่เกิดจากความหนืดของของไหล

กำหนดให้ $\nabla\cdot\bar{D}=\bar{d}$ ทำให้ได้ว่า

$$\nabla \cdot \vec{T} = -\nabla p + \nabla \cdot \vec{D} = -\nabla p + \vec{d}$$

เพราะฉะนั้น สามารถเขียนสมการ (3.13) ได้

$$\int_{x_1}^{x_2} \int_S \left[\frac{\partial \bar{u}}{\partial t} + u_1 \nabla \cdot \bar{u} \right] d\sigma dx = \int_{x_1}^{x_2} \int_S \left[\bar{F}_{body} + \frac{1}{\rho} \left(-\nabla p + \vec{d} \right) \right] d\sigma dx \tag{3.16}$$

สมการ (2.15) เป็นจริง ไม่ว่า x_1 และ x_2 เป็นตำแหน่งใดก็ตาม และเขียนตัวประกอบในแนวแกน x ของสมการนี้ได้ว่า

$$\int_S \left[\frac{\partial \bar{u}}{\partial t} + u_1 \nabla \cdot \bar{u} \right] d\sigma = \int_S \left[F_{body} + \frac{1}{\rho} \left(\frac{\partial p}{\partial x} + d_1 \right) \right] d\sigma \tag{3.17}$$

โดยที่ d_1 เป็นตัวประกอบตามแนวแกน x ของ $\vec{d}$

แทนสมการ (3.16) ลงในสมการ (3.11) ทำให้ได้

$$\frac{\partial}{\partial t}\left(\mathrm{A} \bar{u}_1 \right) + \frac{\partial}{\partial x} A\overline{u_1^{\,2}} = \int_S \left[F_{body} + \frac{1}{\rho} \left(-\frac{\partial p}{\partial x} + d_1 \right) \right] d\sigma + \int_{\partial S} u_1 \,\vec{w} \cdot \vec{n} \, d\gamma \tag{3.18}$$

เมื่อพิจารณาเป็นค่าเฉลี่ยแล้วได้

$$\frac{\partial}{\partial t}\left(\mathrm{A} \bar{u}_1 \right) + \frac{\partial}{\partial x} A\overline{u_1^{\,2}} = \frac{A}{\rho} \left(\rho \bar{F}_{body} - \frac{\partial \bar{p}}{\partial x} + \bar{d}_1 \right) + \int_{\partial S} u_1 \,\vec{w} \cdot \vec{n} \, d\gamma \tag{3.19}$$

พจน์ $\overline{u_1^{\,2}}$ ในสมการนี้ คือฟังก์ชันของแนวเส้นความเร็ว (velocity profile) เขียนในรูปสัมประสิทธิ์คอริออลิส (Coriolis coefficient) ได้ว่า

$$\overline{u_1^{\,2}} = \frac{1}{A} \int_S u_1^{\,2} d\sigma = \alpha \bar{u}_1^{\,2} \tag{3.20}$$

โดยที่

α คือ สัมประสิทธิ์คอริออลิส (Coriolis coefficient) ซึ่ง $\alpha = 1$ สำหรับการไหลที่มีรูปร่างความเร็วแบบราบเรียบ และ $\alpha = 4/3$ สำหรับการไหลที่มีรูปร่างความเร็วแบบพาราโบลิก

พจน์ที่เป็นพจน์ของแรงหนืด $\bar{d}_1$ เขียนในรูปฟังก์ชันเส้นตรงของค่าเฉลี่ยความเร็ว $\bar{u}_1$ ได้ว่า

$$\frac{A}{\rho} \bar{d}_1 = -K_R \bar{u}_1 \tag{3.21}$$

โดย K_R มีค่าเป็นบวก ซึ่งแสดงถึงค่าความต้านทานความหนืดของการไหลต่อหนึ่งหน่วยความยาวของท่อ และพิจารณาได้จากกฎการไหลของปัวซอยย์ ซึ่งได้อธิบายค่าความต้านทานความหนืดสำหรับการไหลแบบราบเรียบไว้ คือ

$$K_R = \frac{8\mu L}{Ar^2} \tag{3.22}$$

และได้สมการรูปสุดท้ายของการอนุรักษ์โมเมนตัม คือ

$$\frac{\partial}{\partial t}\left(\mathrm{A}\bar{u}_1\right) + \frac{\partial}{\partial x}\left(\alpha \bar{u}_1^{\,2}\right) = A\bar{F}_{body} - \frac{A}{\rho}\left(\frac{\partial \bar{p}}{\partial x}\right) - K_R\bar{u}_1 + \int_{\partial \mathrm{S}} u_1\, \bar{w}\cdot\bar{n}\, d\gamma \tag{3.23}$$

เมื่อ กำหนดให้ไม่มีของไหลไหลผ่านผนังของหลอดเลือด ($\bar{w}\cdot\bar{n} = 0$) และ $\bar{F}_{body} = 0$ แล้ว

$$\frac{\partial}{\partial t}\left(\mathrm{A}\bar{u}_1\right) + \frac{\partial}{\partial x}\left(\alpha \bar{u}_1^{\,2}\right) + \frac{A}{\rho}\left(\frac{\partial \bar{p}}{\partial x}\right) + K_R\bar{u}_1 = 0 \tag{3.24}$$

ตัวแปรที่ไม่ทราบค่าในสมการ (3.11) และ (3.24) คือค่าความดัน p พื้นที่หน้าตัด A และความเร็วเฉลี่ย $\bar{u}_1$ ซึ่งเห็นว่าจำนวนตัวแปรเกินจำนวนสมการ ดังนั้นวิธีการแก้ปัญหาที่ทำได้คือการกำหนดความสัมพันธ์ระหว่างความดันของหลอดเลือด p กับพื้นที่หน้าตัดของหลอดเลือด A

3.2.4 ความสัมพันธ์ระหว่างความดันและพื้นที่หน้าตัดของท่อ

กำหนดให้ความสัมพันธ์ระหว่างความดันท่อและพื้นที่หน้าตัดของท่อเป็นแบบยืดหยุ่นหรือแบบอีลาสติก เมื่อพิจารณาแรงในแนวรัศมีของท่อทรงกระบอกในสภาวะสมดุลสถิต ได้ความสัมพันธ์ทางพีชคณิตในหนึ่งมิติดังนี้

$$p = p_{ext} + \beta\left(\sqrt{A} - \sqrt{A_0}\right) \tag{3.25}$$

เมื่อ

$$\beta = \frac{Eh_0}{D_0 A_0} \tag{3.26}$$

โดยที่

h_0 คือความหนาของหลอดเลือดที่สภาวะสมดุล

$A_0 = A_0(x)$ คือพื้นที่หน้าตัดที่สภาวะสมดุล

D_0 คือรัศมีของหลอดเลือดที่สภาวะสมดุล

$(p,Q) = (P_{ext},0)$ ที่สภาวะสมดุล

$E = E(x)$ คือค่าโมดูลัสของยัง

P_{ext} คือ ความดันภายนอก (กำหนดให้มีค่าคงที่)

บทที่ 4
ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์

เมื่อทราบคุณสมบัติของเลือดและหลอดเลือดจากบทที่ 2 และทราบว่าคุณสมบัติเหล่านั้นมีลักษณะและมีผลต่อพฤติกรรมการไหลอย่างไรจากบทที่ 3 แล้ว ในบทนี้กล่าวถึงระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ซึ่งใช้ในการหาคำตอบจากสมการควบคุม และกล่าวถึงการสร้างแบบจำลองจากสมการนาเวียร์-สโตกส์ด้วยการเขียนไพธอน

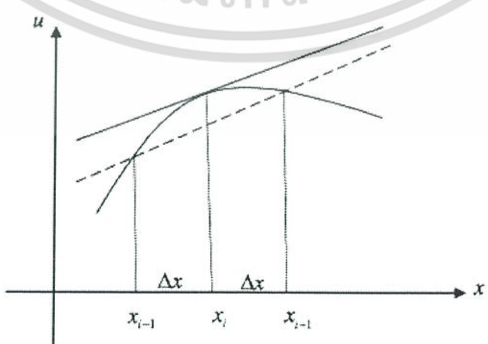
4.1 ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์

การวิเคราะห์เชิงตัวเลข (numerical analysis) คือ การวิเคราะห์สมการทางคณิตศาสตร์โดยใช้เทคนิควิธีการวิเคราะห์เชิงตัวเลข (numerical method) ซึ่งเป็นเทคนิควิธีการหาคำตอบที่ถูกต้องโดยการประมาณค่า สามารถใช้ได้กับการแก้ปัญหาสมการที่มีความซับซ้อน เช่น สมการเชิงอนุพันธ์ย่อย และสมการปริพันธ์.โดยการวิเคราะห์เชิงตัวเลขนี้สามารถทำอะไรได้มากในปัจจุบัน เช่น การพยากรณ์อากาศ การคำนวณเส้นทางการเคลื่อนที่ของยานอวกาศ การจำลองการชนของรถยนต์ การคิดราคาค่าตั๋วของเครื่องบินโดยใช้ข้อมูลงบประมาณต่างๆ เช่น จำนวนลูกเรือ น้ำมัน เป็นต้น การวิเคราะห์การประกันภัย การคาดการการขึ้น-ลงของหุ้น เป็นต้น โดยในการแก้ปัญหาเชิงอนุพันธ์มีหลายวิธี ได้แก่

1. ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ (Finite difference method)
2. ระเบียบวิธีไฟไนต์เอลิเมนต์ (Finite element method)
3. ระเบียบวิธีไฟไนต์วอลุม (Finite volume method)

ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์

เป็นการประมาณค่าอนุพันธ์ วิธีนี้สามารถใช้ประมาณค่าได้ทั้งอนุพันธ์อันดับหนึ่งและอนุพันธ์อันดับสูงๆ โดยการประมาณค่าของอนุพันธ์อันดับสูงนั้นต้องประยุกต์ใช้การประมาณค่าอนุพันธ์อันดับหนึ่ง การประมาณค่าอนุพันธ์อันดับหนึ่งของฟังก์ชันตัวแปรตาม u เทียบกับตัวแปรอิสระ x ซึ่งเขียนแทนด้วย du/dx การวิเคราะห์โดยวิธีนี้ สามารถประมาณค่าของ du/dx ซึ่งคือความชันของเส้นสัมผัสเส้นโค้ง u(x) ที่จุด x ได้ 3 วิธี โดยพิจารณาจากรูปต่อไปนี้



รูปที่ 4.1 ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์

1. ผลต่างข้างหน้า

รูปแบบของผลต่างข้างหน้า คือ

$$\left(\frac{du}{dx}\right)_i \approx \frac{u_{i+1} - u_i}{h} \tag{4.1}$$

2. ผลต่างข้างหลัง

รูปแบบของผลต่างข้างหลัง คือ

$$\left(\frac{du}{dx}\right)_i \approx \frac{u_i - u_{i-1}}{h} \tag{4.2}$$

3. ผลต่างตรงกลาง

รูปแบบของผลต่างตรงกลาง คือ

$$\left(\frac{du}{dx}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2h} \tag{4.3}$$

4.2 ค่าคงที่ของงานวิจัย

1. กำหนดให้ร่างกายแข็งแรง และอยู่ในสภาวะพักซึ่งมีลักษณะการไหลแบบราบเรียบ
2. เลือดเป็นของไหลที่อัดตัวไม่ได้
3. ลักษณะการไหลเป็นแบบนิวโตเนียน
4. ความหนาแน่นของเลือดมีค่า 1060 กิโลกรัมต่อลูกบาศก์เมตร
5. เลือดมีค่าความหนืดจลน์ 2.65 ตารางมิลลิเมตร · วินาที ที่อุณหภูมิร่างกาย 37 องศาเซลเซียส
6. ปริมาตรเลือดที่ออกจากหัวใจมีค่า 5 ลิตรต่อนาที
7. หลอดเลือดมีความยืดหยุ่นแบบอีลาสติก

4.3 การสร้างแบบจำลองจากสมการนาเวียร์-สโตกส์ด้วยการเขียนไพธอน

12 ขั้นตอนสู่นาเวียร์-สโตกส์เป็นส่วนหนึ่งของการสอนในหลักสูตรการจำลองทางคณิตศาสตร์ของของไหลของศาสตราจารย์โลรีนา บาร์บบา มหาวิทยาลัยบอสตันโดยใช้ไพธอนในการหาคำตอบของสมการ มี 12 ขั้นตอนดังนี้

4.3.1 ขั้นที่1 การพาความร้อนแบบเส้นตรงในหนึ่งมิติ

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{4.4}$$

สมการที่ (4.4) คือสมการของการพาความร้อนแบบเส้นตรงในหนึ่งมิติซึ่งเป็นสมการพื้นฐานที่สุด เมื่อให้เงื่อนไขเริ่มต้นของคลื่นมีความเร็ว c คงที่โดยที่รูปร่างของคลื่นไม่เปลี่ยนแปลง นั่นคือให้ให้เงื่อนไขเริ่มต้นคือ $u(x,0)=u_0(x)$ ดังนั้นคำตอบแม่นตรงของสมการนี้คือ $u(x,t)=u_0(x-ct)$

ในการประมาณค่าคำตอบของสมการนี้ ใช้ผลต่างข้างหน้าในการหาอนุพันธ์เมื่อเทียบกับเวลา และใช้ผลต่างข้างหลังในการหาอนุพันธ์เมื่อเทียบกับระยะ โดยกำหนดให้ดัชนีพิกัดของระยะ x มีค่าตั้งตั้ง $i=0$ ถึง N และมีระยะขั้นของเวลาเป็น Δt

จากนิยามของอนุพันธ์

$$\frac{\partial u}{\partial x} \approx \frac{u(x+\Delta x)-u(x)}{\Delta x} \tag{4.5}$$

ได้

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+c\frac{u_i^n-u_{i-1}^n}{\Delta x}=0 \tag{4.6}$$

เมื่อ n และ $n+1$ เป็นค่าดัชนีที่แสดงถึงเวลาสองค่าที่เป็นลำดับที่ติดกัน และขณะที่ $i-1$ และ i เป็นค่าดัชนีที่แสดงถึงพิกัดในแกน x ที่ติดกัน ถ้ากำหนดค่าเงื่อนไขเริ่มต้นแล้ว ค่าที่ไม่ทราบค่าเกิดขึ้นเพียงค่าเดียวคือ u_i^{n+1} สามารถหาค่าที่ไม่ทราบค่าได้จากสมการที่ (4.7)

$$u_i^{n+1}=u_i^n-c\frac{\Delta t}{\Delta x}(u_i^n-u_{i-1}^n) \tag{4.7}$$

ต่อไปเป็นขั้นตอนการดำเนินการในไพธอน โดยใช้โปรแกรมจูปิเตอร์โน้ตบุ๊ก เริ่มต้นโดยการนำเข้าคลังโปรแกรมหรือเรียกว่าไลบรารี (library) ที่จำเป็น เช่น

- numpy คือ ไลบรารีที่จัดเตรียมเมตริกซ์ที่มีประโยชน์
- matplotlib คือ ไลบรารีที่ใช้ในการวาดกราฟสองมิติ ซึ่งใช้ในการวาดกราฟของผลลัพธ์

In[1]:

```
import numpy
from matplotlib import pyplot
%matplotlib inline
```

ต่อไปเป็นการกำหนดตัวแปร โดยกำหนดให้โดเมนคือ $x_i \in (0,2)$ และให้ nx และ nt เป็นจำนวนจุด

ของกริดและจำนวนของขั้นเวลา และกำหนด dx และ dt ให้เป็นระยะห่างระหว่างจุดของกริดที่ติดกันและช่วงเวลาของแต่ละขั้นเวลา และให้ความเร็วคลื่น $c=1$ และเลขเคอร์เรนท์ $\sigma=0.5$ ซึ่งมีความสัมพันธ์กับขั้นเวลาดังสมการที่ (4.8)

$$\Delta t = \frac{\sigma \Delta x}{c} \tag{4.8}$$

In[2]:

```
nx = 41
dx = 2 / (nx-1)
nt = 25
sigma = .5
c = 1
dt = sigma * dx / c
```

ในการกำหนดความเร็วเริ่มต้น(u_0) เมื่อกำหนดให้ $u_0=2$ เมื่อ $0.5 \le x \le 1$ และ $u_0=1$ เมื่อ x อยู่นอกช่วง $0.5 \le x \le 1$ ในโดเมน จากนั้นเป็นการใช้ฟังก์ชัน ones() กำหนดอาร์เรย์(array)ของความเร็วเริ่มต้นที่ $t=0$ ของเอลิเมนต์จำนวน nx เอลิเมนต์ทุกค่ามีค่าเท่ากับ 1

In[3]:

```
u = numpy.ones(nx)        #กำหนดอาร์เรย์ของความเร็วเริ่มต้นที่ t = 0 ของเอลิเมนต์จำนวน nx เอลิเมนต์ทุกค่ามีค่าเท่ากับ 1

u[int(.5 / dx):int(1 / dx + 1)] = 2        #กำหนดเอลิเมนต์ที่ตำแหน่ง 0.5 ≤ x ≤ 1 ในอาร์เรย์ของความเร็วเริ่มต้นที่ t = 0 มีค่าเท่ากับ 2
print(u)
```

Out[3]:

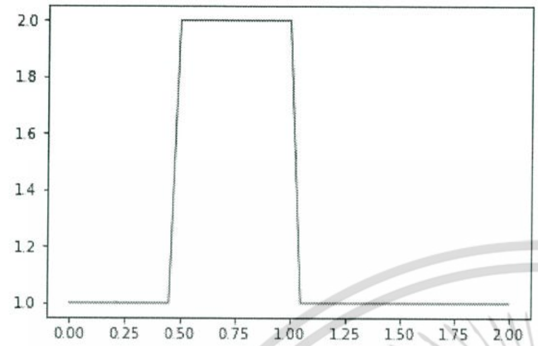
```
[ 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

จากนั้นสามารถวาดกราฟของผลลัพธ์ที่ได้โดยใช้ฟังก์ชัน plot ในไลบรารี pyplot ได้โดยใช้คำสั่ง pyplot.plot(x,y) โดยให้ค่า x คืออาร์เรย์ของตำแหน่งจุดของกริด และให้ค่า y คืออาร์เรย์ของความเร็วเริ่มต้น u_0

In[4]:

```
pyplot.plot(numpy.linspace(0, 2, nx), u);
```

Out[4]:



จากสมการที่ (4.7)

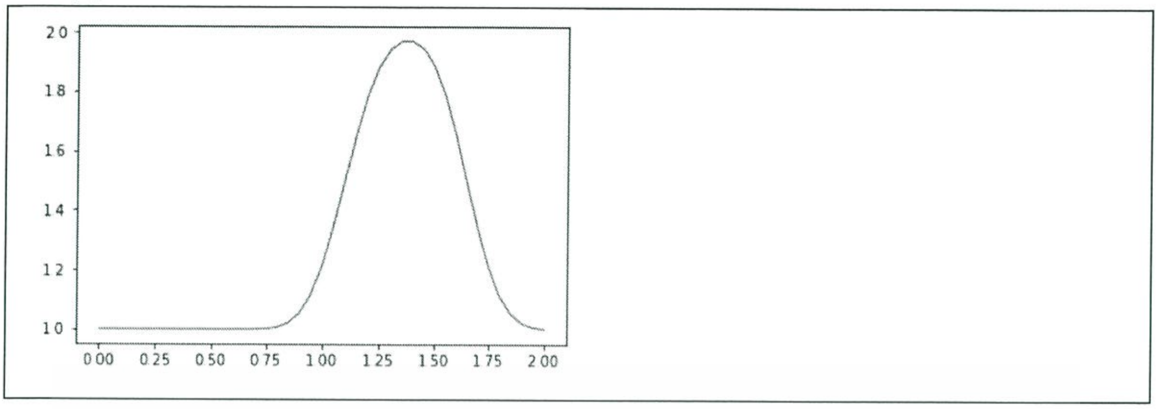
$$u_i^{n+1} = u_i^n - c\frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) \tag{4.7}$$

ดังนั้นสามารถหาค่า u_i^{n+1} ได้ดังนี้

In[5]:

```
for n in range(nt):        #กำหนดลูปสำหรับค่า n จาก 0 ถึง nt ดังนั้นลูปนี้ต้องดำเนินการ nt ครั้ง
    un = u.copy()        #คัดลอกอาร์เรย์ u
    for i in range(1, nx):        #กำหนดลูปสำหรับค่า i จาก 1 ถึง nx-1
        u[i] = un[i] - c * dt / dx * (un[i] - un[i-1])        #สมการที่ (4.7)

pyplot.plot(numpy.linspace(0, 2, nx), u);
```

Out[5]:



4.3.2 ขั้นที่2 การพาความร้อนแบบไม่เป็นเส้นตรงในหนึ่งมิติ

สมการการพาความร้อนแบบไม่เป็นเส้นตรงในหนึ่งมิติคือ

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{4.9}$$

ในสมการที่ (4.9) เป็นการเปลี่ยนค่าคงที่ c ที่คูณในพจน์ที่สอง เป็น u แทน ดังนั้นพจน์ที่สองไม่เป็นเส้นตรง เมื่อใช้ผลต่างข้างหน้าในการหาอนุพันธ์เมื่อเทียบกับเวลา ใช้ผลต่างข้างหลังในการหาอนุพันธ์เมื่อเทียบกับระยะ

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \tag{4.10}$$

ตัวแปรที่ไม่ทราบค่า u_i^{n+1} สามารถหาได้จากสมการที่ (4.11)

$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) \tag{4.11}$$

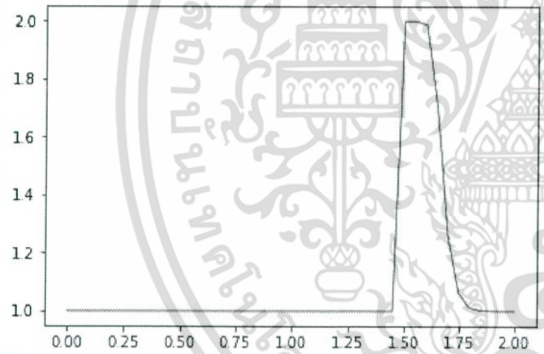
ดังนั้นสามารถแก้สมการได้ดังนี้

In[1]:

```
import numpy
from matplotlib import pyplot
%matplotlib inline


nx = 41
dx = 2 / (nx - 1)
```

```
nt = 20
sigma = .5
dt = sigma * dx


u = numpy.ones(nx)
u[int(.5 / dx) : int(1 / dx + 1)] = 2
un = numpy.ones(nx)

for n in range(nt):
        un = u.copy()
        for i in range(1, nx):
                u[i] = un[i] - un[i] * dt / dx * (un[i] - un[i-1])

pyplot.plot(numpy.linspace(0, 2, nx), u)
```

Out[1]:



4.3.3 ขั้นที่3 การแพร่ในหนึ่งมิติ

สมการการแพร่ในหนึ่งมิติ คือ

$$\frac{\partial u}{\partial t} = \upsilon \frac{\partial^2 u}{\partial x^2} \tag{4.12}$$

ในการพิจารณาพจน์อนุพันธ์ลำดับที่ 2 จากพหุนามเทเลอร์อันดับสองซึ่งใช้ประมาณค่า $u(x)$ สำหรับจุด x ใกล้จุด a คือ

$$u(x) \approx Q(x) = u(a) + \frac{\partial u(a)}{\partial x}(x-a) + \left[\left(\frac{1}{2}\right)\left(\frac{\partial u(a)}{\partial x}\right)(x-a)^2\right] \tag{4.13}$$

ดังนั้น ประมาณค่าของ u_{i+1} และ u_{i-1} รอบ u_i ได้ว่า

$$u_{i+1} = u_i + \Delta x \left.\frac{\partial u}{\partial x}\right|_i + \frac{(\Delta x)^2}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_i \tag{4.14}$$

$$u_{i-1} = u_i - \Delta x \left.\frac{\partial u}{\partial x}\right|_i + \frac{(\Delta x)^2}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_i \tag{4.15}$$

เมื่อนำ สมการ(4.14) + สมการ(4.15) ทำให้ได้

$$u_{i+1} + u_{i-1} = 2u_i + (\Delta x)^2 \left.\frac{\partial^2 u}{\partial x^2}\right|_i \tag{4.16}$$

จัดรูปเพื่อหาค่า $\left.\frac{\partial^2 u}{\partial x^2}\right|_i$ ได้ดังนี้

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} \tag{4.17}$$

นำค่าประมาณอนุพันธ์อันดับสองที่ได้มาเขียนสมการของการแพรในหนึ่งมิติได้

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \upsilon \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \tag{4.18}$$

ตัวแปรเดียวที่ไม่ทราบค่าคือ u_i^{n+1} สามารถจัดรูปใหม่ได้

$$u_i^{n+1} = u_i^n + \frac{\upsilon \Delta t}{(\Delta x)^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \tag{4.19}$$

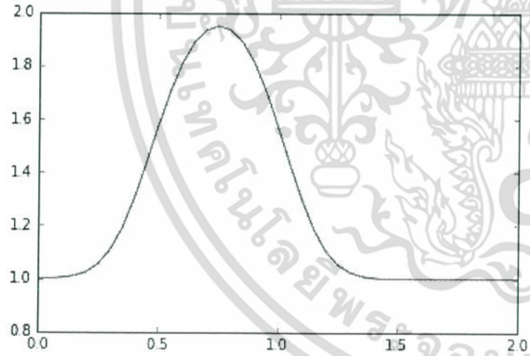
ในการแก้สมการข้างต้น จำเป็นต้องมีเงื่อนไขเริ่มต้น ดังนั้นในโดเมน $0 \le x \le 2$ กำหนดให้ที่ $t = 0$ ความเร็วเริ่มต้น $u_0 = 2$ เมื่อ $0.5 \le x \le 1$ และ $u_0 = 1$ เมื่อ x อยู่นอกช่วง $0.5 \le x \le 1$ ในโดเมน

In[1]:

```
import numpy
from matplotlib import pyplot
%matplotlib inline
```

```
nx = 41
dx = 2 / (nx - 1)
nt = 20
nu = 0.3      #กำหนดความหนืดเป็น0.3
sigma = .2
dt = sigma * dx**2 / nu

u = numpy.ones(nx)
u[int(.5 / dx):int(1 / dx + 1)] = 2

    un = numpy.ones(nx)
    for n in range(nt):
        un = u.copy() ##copy the existing values of u into un
        for i in range(1, nx - 1):
            u[i] = un[i] + nu * dt / dx**2 * (un[i+1] - 2 * un[i] + un[i-1])

pyplot.plot(numpy.linspace(0, 2, nx), u);
```

Out[1]:



4.3.4 ขั้นที่4 สมการของเบอร์เจอร์ (Burgers' Equation)

สมการของเบอร์เจอร์ในหนึ่งมิติ คือ

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=\upsilon\frac{\partial^2 u}{\partial x^2} \tag{4.20}$$

สมการที่ (4.20) เป็นการรวมกันของการพาความความร้อนและการแพร่ ซึ่งได้กล่าวถึงไว้ในขั้นตอนที่ 2 และ 3 โดยใช้ผลต่างข้างหน้าในการหาอนุพันธ์เมื่อเทียบกับเวลา ใช้ผลต่างข้างหลังในการหาอนุพันธ์เมื่อเทียบกับระยะ และใช้พหุนามเทเลอร์อันดับสองสำหรับการประมาณค่าอนุพันธ์อันดับที่ 2 ดังนั้นได้ว่า

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+u_i^n\frac{u_i^n-u_{i-1}^n}{\partial x}=\upsilon\frac{u_{i+1}^n-2u_i^n+u_{i-1}^n}{\left(\Delta x\right)^2} \tag{4.21}$$

จากสมการที่ (4.21) ถ้ามีเงื่อนไขเริ่มต้น ตัวแปรตัวเดียวที่ไม่ทราบค่าคือ u_i^{n+1} เขียนสมการใหม่ได้ว่า

$$u_i^{n+1}=u_i^n-u_i^n\frac{\Delta t}{\Delta x}\left(u_i^n-u_{i-1}^n\right)+\frac{\upsilon\Delta t}{\left(\Delta x\right)^2}\left(u_{i+1}^n-2u_i^n+u_{i-1}^n\right) \tag{4.22}$$

กำหนดให้เงื่อนไขเริ่มต้นคือ

$$u=-\frac{2\upsilon}{\phi}\frac{\partial\phi}{\partial x}+4 \tag{4.23}$$

$$\phi=\exp\left(\frac{-\left(x-4t\right)^2}{4\upsilon\left(t+1\right)}\right)+\exp\left(\frac{-\left(x-4t-2\pi\right)^2}{4\upsilon\left(t+1\right)}\right) \tag{4.24}$$

และกำหนดให้เงื่อนไขขอบเป็นแบบคาบคือ

$$u\left(0\right)=u\left(2\pi\right) \tag{4.25}$$

จากเงื่อนไขข้างต้น พจน์ $\frac{\partial\phi}{\partial x}$ มีความซับซ้อนและยุ่งยากในการเขียนพจน์นี้โดยตรง ดังนั้นจึงเรียกใช้ไลบรารี sympy ซึ่งเป็นไลบรารีสัญลักษณ์ทางคณิตศาสตร์สำหรับไพธอน ได้ดังนี้

In[1]:

```
import numpy
import sympy

from sympy import init_printing
init_printing(use_latex=True)      #กำหนดการแสดงผลเพื่อให้มองง่ายขึ้น

x, nu, t = sympy.symbols('x nu t')      #กำหนดสัญลักษณ์ให้ x nu t
phi = (sympy.exp(-(x - 4 * t)**2 / (4 * nu * (t + 1))) +sympy.exp(-(x - 4 * t - 2 * sympy.pi)**
      2 / (4 * nu * (t + 1))))
      #ใช้สัญลักษณ์ทางคณิตศาสตร์ของ exp และ pi ในไลบรารี sympy
phiprime = phi.diff(x)      #กำหนดให้ phiprime มีค่าเป็นอนุพันธ์ของ phi เมื่อเทียบกับ x
phiprime
```

Out[1]:

$$-\frac{(-8t+2x)e^{-\frac{(-4t+x)^2}{4\upsilon(t+1)}}}{4\upsilon(t+1)}-\frac{(-8t+2x-4\pi)e^{-\frac{(-4t+x-2\pi)^2}{4\upsilon(t+1)}}}{4\upsilon(t+1)}$$

ต่อไปเป็นการใช้ประโยชน์จากฟังก์ชันแลมดิไฟ(lambdify) เพื่อแปลสัญลักษณ์ในไลบรารี sympy และแสดงออกในรูปของฟังก์ชันไพธอน ดังแสดงในตัวอย่างต่อไปนี้

In[2]:

```
from sympy.utilities.lambdify import lambdify

u = -2 * nu * (phiprime / phi) + 4

ufunc = lambdify((t, x, nu), u)       #สร้างฟังก์ชัน ufunc(t, x, nu) คือค่าของ u ที่ t, x, nu ใดๆ

print(ufunc(1, 4, 3))       #แสดงผลค่าของ u เมื่อ t=1 x=4 nu=3
```

Out[2]:

```
3.49170664206445
```

ต่อไปคือการกำหนดเงื่อนไขเบื้องต้นที่ $t=0$ และใช้ฟังก์ชัน ufunc(t, x, nu) ที่กำหนดไว้ข้างต้นในการกำหนดค่าของเงื่อนไขเบื้องต้น (u_0)

In[3]:

```
from matplotlib import pyplot
%matplotlib inline

nx = 101
nt = 100
dx = 2 * numpy.pi / (nx - 1)
nu = .07
dt = dx * nu

x = numpy.linspace(0, 2 * numpy.pi, nx)
un = numpy.empty(nx)            #กำหนดให้อาร์เรย์ un มีสมาชิกเป็น 0 หรือเข้าใกล้ 0 และมีสมาชิก
                                nx ตัว
t = 0
```

```
u = numpy.asarray([ufunc(t, x0, nu) for x0 in x])      #กำหนดให้ u เป็นอาร์เรย์ของฟังก์ชัน
                                                         ufunc(t, x0, nu) เมื่อ t=0, x0 คือสมาชิก
                                                       ในอาร์เรย์ x, nu=.07
u
```

Out[3]:

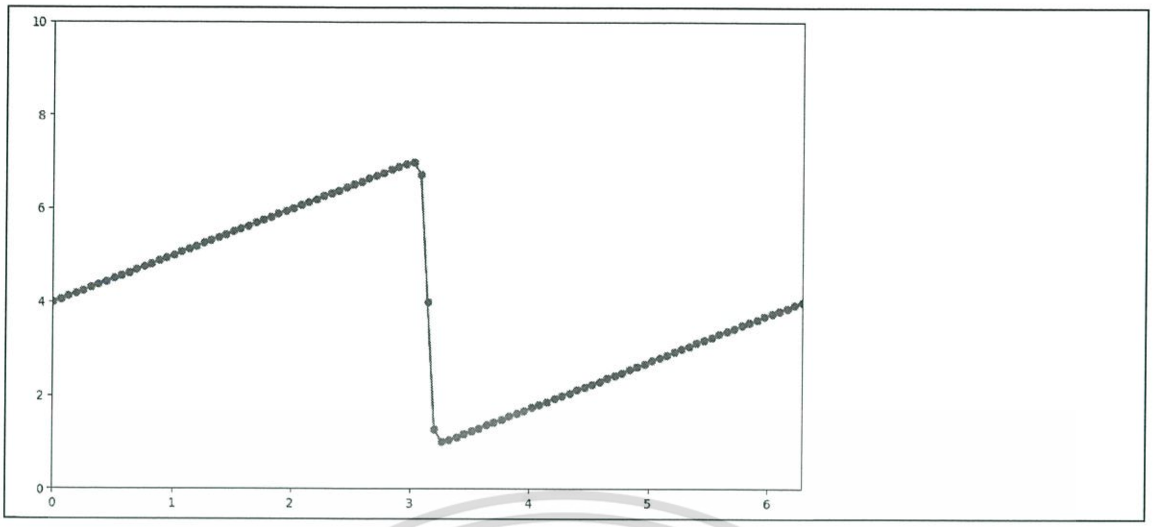
```
array([ 4.        , 4.06283185, 4.12566371, 4.18849556, 4.25132741,
        4.31415927, 4.37699112, 4.43982297, 4.50265482, 4.56548668,
        4.62831853, 4.69115038, 4.75398224, 4.81681409, 4.87964594,
        4.9424778 , 5.00530965, 5.0681415 , 5.13097336, 5.19380521,
        5.25663706, 5.31946891, 5.38230077, 5.44513262, 5.50796447,
        5.57079633, 5.63362818, 5.69646003, 5.75929189, 5.82212374,
        5.88495559, 5.94778745, 6.0106193 , 6.07345115, 6.136283  ,
        6.19911486, 6.26194671, 6.32477856, 6.38761042, 6.45044227,
        6.51327412, 6.57610598, 6.63893783, 6.70176967, 6.76460125,
        6.82742866, 6.89018589, 6.95176632, 6.99367964, 6.72527549,
        4.        , 1.27472451, 1.00632036, 1.04823368, 1.10981411,
        1.17257134, 1.23539875, 1.29823033, 1.36106217, 1.42389402,
        1.48672588, 1.54955773, 1.61238958, 1.67522144, 1.73805329,
        1.80088514, 1.863717  , 1.92654885, 1.9893807 , 2.05221255,
        2.11504441, 2.17787626, 2.24070811, 2.30353997, 2.36637182,
        2.42920367, 2.49203553, 2.55486738, 2.61769923, 2.68053109,
        2.74336294, 2.80619479, 2.86902664, 2.9318585 , 2.99469035,
        3.0575222 , 3.12035406, 3.18318591, 3.24601776, 3.30884962,
        3.37168147, 3.43451332, 3.49734518, 3.56017703, 3.62300888,
        3.68584073, 3.74867259, 3.81150444, 3.87433629, 3.93716815, 4.        ])
```

แสดงกราฟความสัมพันธ์ระหว่าง x และ u_0

In[4]:

```
pyplot.figure(figsize=(11, 7), dpi=100)     #กำหนดขนาดรูปกราฟและขนาดพิกเซล
pyplot.plot(x, u, marker='o', lw=2)         #กำหนดให้แสดงกราฟระหว่าง x และ u โดยให้แต่ละ
                                              จุดทำเครื่องหมายวงกลม และกำหนดความหนาของ
                                              เส้นกราฟ=2
pyplot.xlim([0, 2 * numpy.pi])              #กำหนดให้แสดงค่าแกนของ x ที่ 0 ถึง 2
pyplot.ylim([0, 10]);                       #กำหนดให้แสดงค่าแกนของ y ที่ 0 ถึง 10
```

Out[4]:



จากสมการที่ (4.22)

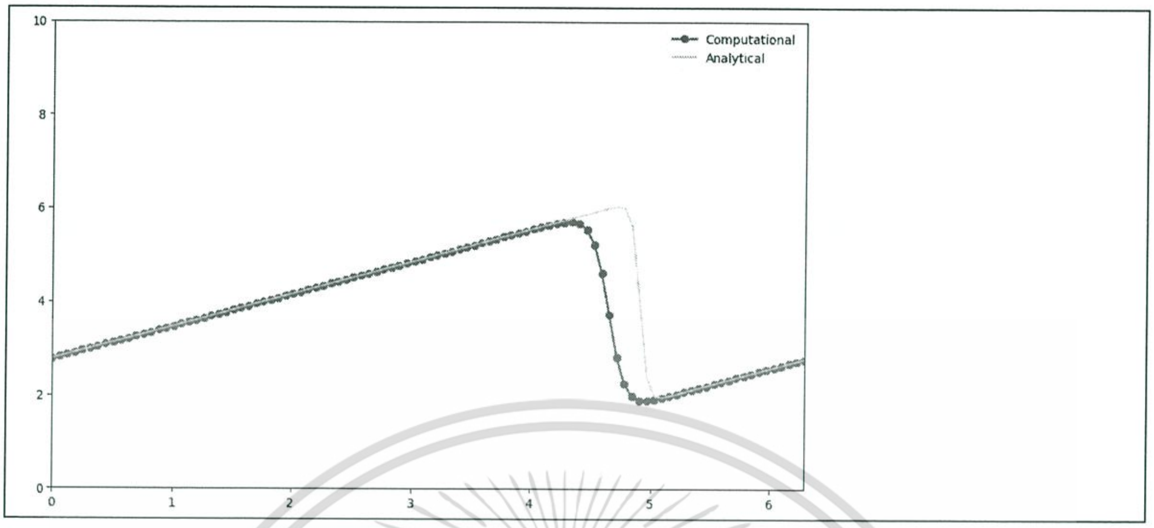
$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right) + \frac{\upsilon \Delta t}{\left(\Delta x\right)^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \qquad (4.22)$$

สามารถหาค่าของ u_i^{n+1} และกำหนดเงื่อนไขขอบแบบคาบได้ดังนี้

In[5]:

```
for n in range(nt):
    un = u.copy()
    for i in range(1, nx-1):
        u[i] = un[i] - un[i] * dt / dx *(un[i] - un[i-1]) + nu * dt / dx**2 *\ (un[i+1] - 2 *
            un[i] + un[i-1])
    u[0] = un[0] - un[0] * dt / dx * (un[0] - un[-2]) + nu * dt / dx**2 *\ (un[1] - 2 * un[0] +
        un[-2])      #กำหนดค่าเอลิเมนต์ที่ 0 ของอาร์เรย์ u ในลูป n
    u[-1] = u[0]      #กำหนดเงื่อนไขขอบแบบคาบ

u_analytical = numpy.asarray([ufunc(nt * dt, xi, nu) for xi in x])

pyplot.figure(figsize=(11, 7), dpi=100)
pyplot.plot(x,u, marker='o', lw=2, label='Computational')
pyplot.plot(x, u_analytical, label='Analytical')
pyplot.xlim([0, 2 * numpy.pi])
pyplot.ylim([0, 10])
pyplot.legend();
```

Out[5]:



4.3.5 ขั้นที่5 การพาความร้อนแบบเส้นตรงในสองมิติ

สำหรับการวิเคราะห์ในสองมิติ เมื่อกริดมีรูปร่างเป็นสี่เหลี่ยมผืนผ้า กำหนดจุดของกริด $u_{i,j} = u\left(x_i, y_j\right)$ ได้ดังนี้

$$x_i = x_0 + i\Delta x$$
$$y_j = y_0 + j\Delta y \tag{4.26}$$

และอนุพันธ์ย่อยสำหรับฟังก์ชันสองตัวแปร เมื่อใช้ผลต่างข้างหลัง คือ

$$\left.\frac{\partial u}{\partial x}\right|_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}$$
$$\left.\frac{\partial u}{\partial y}\right|_{i,j} = \frac{u_{i,j} - u_{i,j-1}}{\Delta y} \tag{4.27}$$

สมการควบคุมสำหรับการพาความร้อนแบบเส้นตรงในสองมิติคือ

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} + c\frac{\partial u}{\partial y} = 0 \tag{4.28}$$

เมื่อใช้ผลต่างข้างหน้าในการหาอนุพันธ์เมื่อเทียบกับเวลา และใช้ผลต่างข้างหลังในการหาอนุพันธ์เมื่อเทียบกับระยะ

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}+c\frac{u_{i,j}^{n}-u_{i-1,j}^{n}}{\Delta x}+c\frac{u_{i,j}^{n}-u_{i,j-1}^{n}}{\Delta y}=0 \tag{4.29}$$

จัดสมการใหม่เพื่อหาตัวที่ไม่ทราบค่า ได้

$$u_{i,j}^{n+1}=u_{i,j}^{n}-c\frac{\Delta t}{\Delta x}(u_{i,j}^{n}-u_{i-1,j}^{n})-c\frac{\Delta t}{\Delta y}(u_{i,j}^{n}-u_{i,j-1}^{n}) \tag{4.30}$$

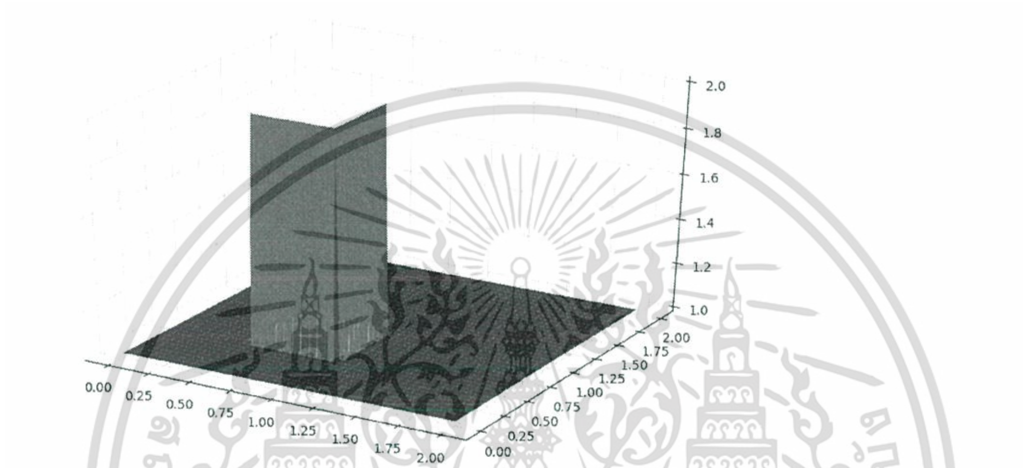
เมื่อกำหนดให้เงื่อนไขเริ่มต้น $u_0(x,y)=2$ เมื่อ $0.5\le x,y\le 1$ และ $u_0(x,y)=1$ เมื่ออยู่นอกช่วง $0.5\le x,y\le 1$ และกำหนดให้เงื่อนไขขอบ $u(x,y)=1$ เมื่อ $x=0$ หรือ 2 และ $y=0$ หรือ 2 และใช้ไลบรารี Axes3D ในการวาดกราฟในสามมิติ

In[1]:

```
from mpl_toolkits.mplot3d import Axes3D        #นำเข้าไลบรารี Axes3D
import numpy
from matplotlib import pyplot, cm
%matplotlib inline

nx = 81
ny = 81
nt = 100
c = 1
dx = 2 / (nx - 1)
dy = 2 / (ny - 1)
sigma = .2
dt = sigma * dx

x = numpy.linspace(0, 2, nx)
y = numpy.linspace(0, 2, ny)

##กำหนดเงื่อนไขเริ่มต้น

u = numpy.ones((ny, nx))
un = numpy.ones((ny, nx))

#กำหนดให้อาร์เรย์มีเอลิเมนต์ 0.5<=x<=1 และ 0.5<=y<=1 มีค่าเท่ากับ 2
u[int(.5 / dy):int(1 / dy + 1),int(.5 / dx):int(1 / dx + 1)] = 2
```

```
#กำหนดกราฟความสัมพันธ์ระหว่าง x, y, u
fig = pyplot.figure(figsize=(11, 7), dpi=100)
ax = fig.gca(projection='3d')
X, Y = numpy.meshgrid(x, y)
surf = ax.plot_surface(X, Y, u[:], cmap=cm.viridis)
```

Out[1]:



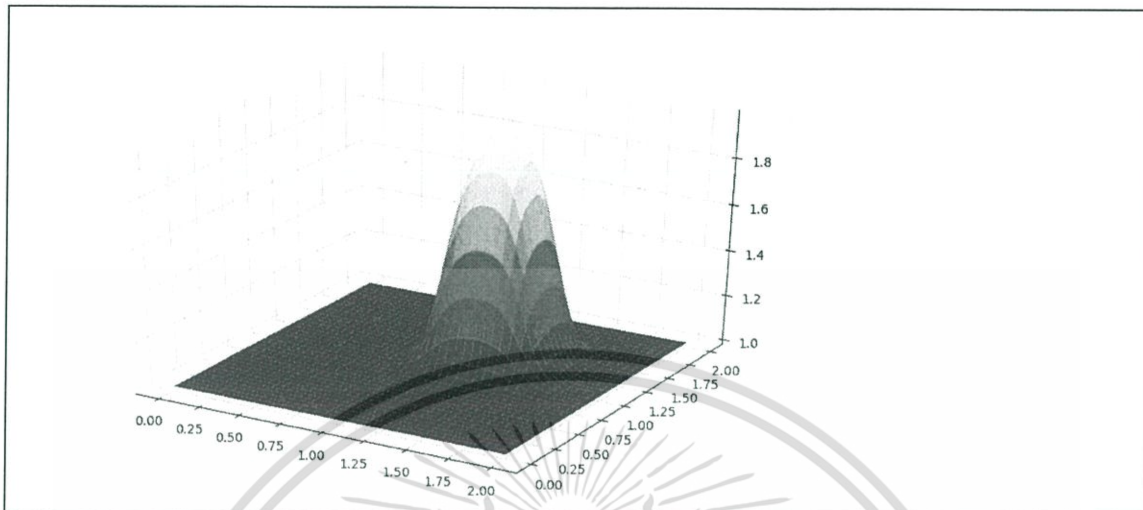
ใช้การทำงานของอาร์เรย์ในการวิเคราะห์ข้อมูลของการพาความร้อนแบบเส้นตรงในสองมิติได้ดังนี้

In[2]:

```
u = numpy.ones((ny, nx))
u[int(.5 / dy):int(1 / dy + 1), int(.5 / dx):int(1 / dx + 1)] = 2

for n in range(nt + 1):
    un = u.copy()
    u[1:, 1:] = (un[1:, 1:] - (c * dt / dx * (un[1:, 1:] - un[1:, :-1])) - (c * dt / dy * (un[1:, 1:] -
                 un[:-1, 1:]))
    u[0, :] = 1
    u[-1, :] = 1
    u[:, 0] = 1
    u[:, -1] = 1

fig = pyplot.figure(figsize=(11, 7), dpi=100)
ax = fig.gca(projection='3d')
X, Y = numpy.meshgrid(x, y)
```

```
surf2 = ax.plot_surface(X, Y, u[:], cmap=cm.viridis)
```

Out[2]:



4.3.6 ขั้นที่6 การพาความร้อนแบบไม่เป็นเส้นตรงในสองมิติ

สมการการพาแบบไม่เป็นเส้นตรงในสองมิติคือ

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=0$$
$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=0 \tag{4.31}$$

เมื่อใช้ผลต่างข้างหน้าในการหาอนุพันธ์เมื่อเทียบกับเวลา และใช้ผลต่างข้างหลังในการหาอนุพันธ์เมื่อเทียบกับระยะ

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{u_{i,j}^{n}-u_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\frac{u_{i,j}^{n}-u_{i,j-1}^{n}}{\Delta y}=0$$
$$\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{v_{i,j}^{n}-v_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\frac{v_{i,j}^{n}-v_{i,j-1}^{n}}{\Delta y}=0 \tag{4.32}$$

จัดสมการใหม่เพื่อหาค่าตัวแปรที่ไม่ทราบค่า ได้ดังนี้

$$u_{i,j}^{n+1}=u_{i,j}^{n}-u_{i,j}\frac{\Delta t}{\Delta x}(u_{i,j}^{n}-u_{i-1,j}^{n})-v_{i,j}^{n}\frac{\Delta t}{\Delta y}(u_{i,j}^{n}-u_{i,j-1}^{n})$$
$$v_{i,j}^{n+1}=v_{i,j}^{n}-u_{i,j}\frac{\Delta t}{\Delta x}(v_{i,j}^{n}-v_{i-1,j}^{n})-v_{i,j}^{n}\frac{\Delta t}{\Delta y}(v_{i,j}^{n}-v_{i,j-1}^{n}) \tag{4.33}$$

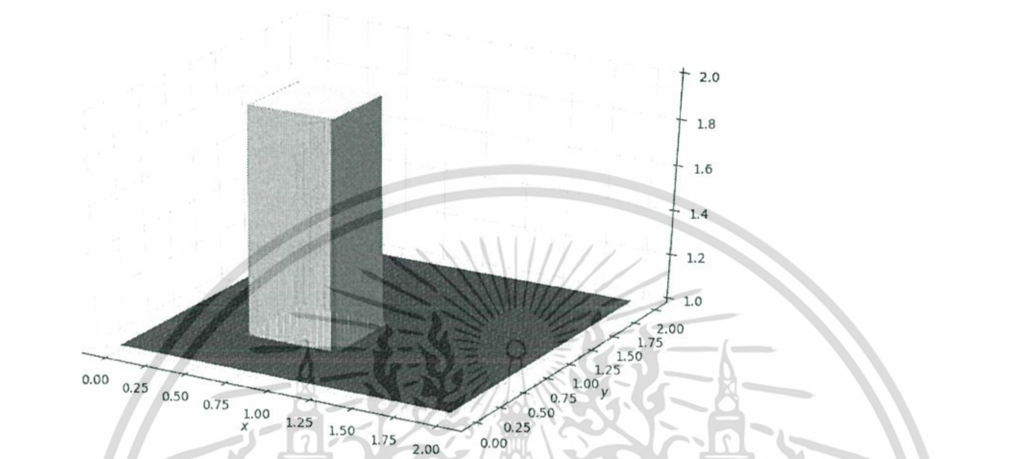
ให้เงื่อนไขเริ่มต้นเหมือนกันกับเงื่อนไขเริ่มต้นสำหรับการพาความร้อนในหนึ่งมิติ ทั้งในแกน x และ y ดังนั้น $v_0(x,y)=u_0(x,y)=2$ เมื่อ $0.5 \le x,y \le 1$ และ $v_0(x,y)=u_0(x,y)=1$ เมื่ออยู่นอกช่วง $0.5 \le x,y \le 1$ และกำหนดให้เงื่อนไขขอบ $u(x,y)=1$ และ $v(x,y)=1$ เมื่อ $x=0$ หรือ 2 และ $y=0$ หรือ 2

In[1]:

```
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import pyplot, cm
import numpy
%matplotlib inline

nx = 101
ny = 101
nt = 80
c = 1
dx = 2 / (nx - 1)
dy = 2 / (ny - 1)
sigma = .2
dt = sigma * dx / c

x = numpy.linspace(0, 2, nx)
y = numpy.linspace(0, 2, ny)

u = numpy.ones((ny, nx))
v = numpy.ones((ny, nx))
un = numpy.ones((ny, nx))
vn = numpy.ones((ny, nx))

##กำหนดเงื่อนไขเริ่มต้น

#กำหนดให้อาร์เรย์ u มีเอลิเมนต์ 0.5<=x<=1 และ 0.5<=y<=1 มีค่าเท่ากับ 2
u[int(.5 / dy):int(1 / dy + 1), int(.5 / dx):int(1 / dx + 1)] = 2
#กำหนดให้อาร์เรย์ v มีเอลิเมนต์ 0.5<=x<=1 และ 0.5<=y<=1 มีค่าเท่ากับ 2
v[int(.5 / dy):int(1 / dy + 1), int(.5 / dx):int(1 / dx + 1)] = 2

fig = pyplot.figure(figsize=(11, 7), dpi=100)
ax = fig.gca(projection='3d')
```

```
X, Y = numpy.meshgrid(x, y)
ax.plot_surface(X, Y, u, cmap=cm.viridis, rstride=2, cstride=2)
ax.set_xlabel('$x$')
ax.set_ylabel('$y$');
```

Out[1]:



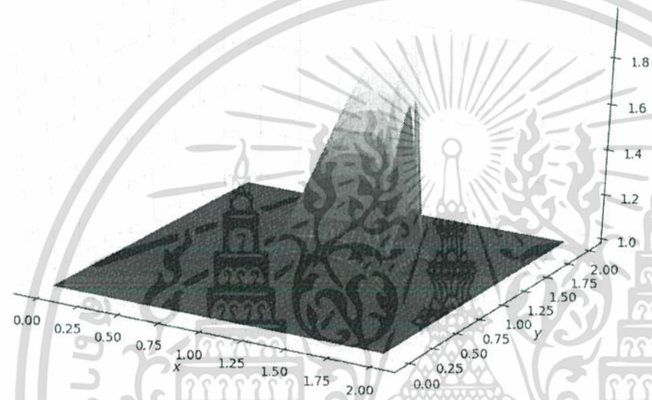
หาค่า $u_{i,j}^{n+1}$ จากสมการที่ (4.33) ได้ดังนี้

In[2]:

```
for n in range(nt + 1): ##loop across number of time steps
    un = u.copy()
    vn = v.copy()
    u[1:, 1:] = (un[1:, 1:] -  (un[1:, 1:] * c * dt / dx * (un[1:, 1:] - un[1:, :-1])) -vn[1:, 1:] * c *
                dt / dy * (un[1:, 1:] - un[:-1, 1:]))
    v[1:, 1:] = (vn[1:, 1:] - (un[1:, 1:] * c * dt / dx * (vn[1:, 1:] - vn[1:, :-1])) -vn[1:, 1:] * c *
                dt / dy * (vn[1:, 1:] - vn[:-1, 1:]))

    u[0, :] = 1
    u[-1, :] = 1
    u[:, 0] = 1
    u[:, -1] = 1

    v[0, :] = 1
    v[-1, :] = 1
    v[:, 0] = 1
    v[:, -1] = 1
```

```
fig = pyplot.figure(figsize=(11, 7), dpi=100)
ax = fig.gca(projection='3d')
X, Y = numpy.meshgrid(x, y)
ax.plot_surface(X, Y, u, cmap=cm.viridis, rstride=2, cstride=2)
ax.set_xlabel('$x$')
ax.set_ylabel('$y$');
```

Out[2]:

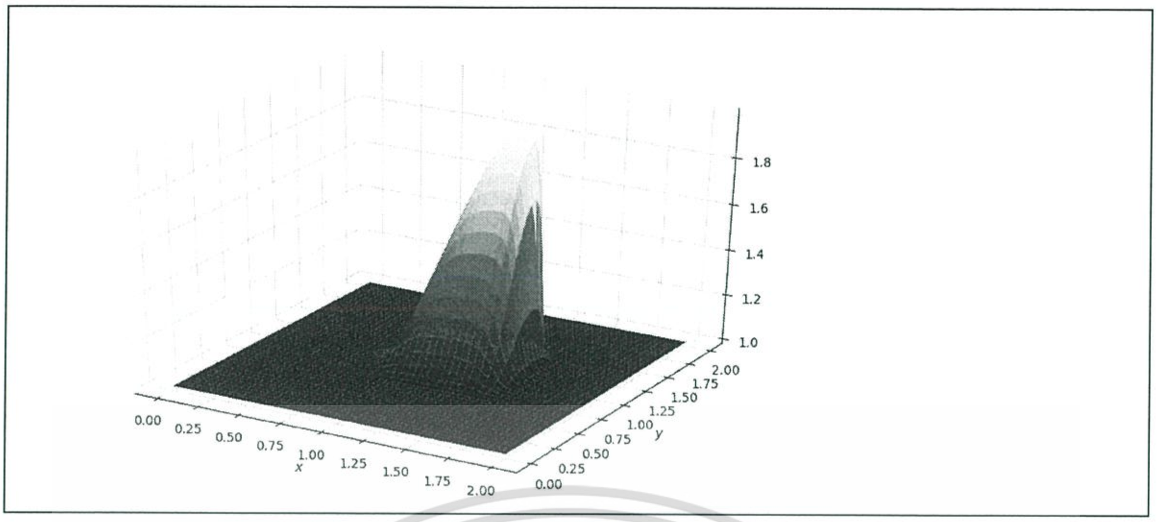


และหาค่า $v_{i,j}^{n+1}$ จากสมการที่ (4.33) ได้ดังนี้

In[3]:

```
fig = pyplot.figure(figsize=(11, 7), dpi=100)
ax = fig.gca(projection='3d')
X, Y = numpy.meshgrid(x, y)
ax.plot_surface(X, Y, v, cmap=cm.viridis, rstride=2, cstride=2)
ax.set_xlabel('$x$')
ax.set_ylabel('$y$');
```

Out[3]:



4.3.7 ขั้นที่7 การแพร่ในสองมิติ

สมการการแพร่ในสองมิติ คือ

$$\frac{\partial u}{\partial t} = \upsilon \frac{\partial^2 u}{\partial x^2} + \upsilon \frac{\partial^2 u}{\partial y^2} \tag{4.34}$$

เมื่อใช้ผลต่างข้างหน้าในการหาอนุพันธ์เมื่อเทียบกับเวลา และใช้ผลต่างข้างหลังในการหาอนุพันธ์เมื่อเทียบกับระยะ

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \upsilon \frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \upsilon \frac{u_{i,j+1}^{n} - 2u_{i}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2} \tag{4.35}$$

ตัวแปรเดียวที่ไม่ทราบค่าคือ u_i^{n+1} จัดรูปใหม่เพื่อหาค่าที่ไม่ทราบค่าได้ดังนี้

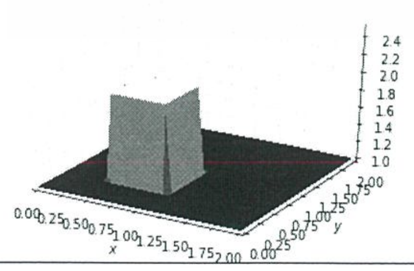
$$u_{i,j}^{n+1} = u_{i,j}^{n} + \frac{\upsilon \Delta t}{(\Delta x)^2}\left(u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}\right) + \frac{\upsilon \Delta t}{(\Delta y)^2}\left(u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}\right) \tag{4.36}$$

In[1]:

```
import numpy
from matplotlib import pyplot, cm
from mpl_toolkits.mplot3d import Axes3D ##library for 3d projection plots
%matplotlib inline


nx = 31
ny = 31
nt = 17
```

```
nu = .05
dx = 2 / (nx - 1)
dy = 2 / (ny - 1)
sigma = .25
dt = sigma * dx * dy / nu

x = numpy.linspace(0, 2, nx)
y = numpy.linspace(0, 2, ny)

u = numpy.ones((ny, nx))
un = numpy.ones((ny, nx))

#กำหนดเงื่อนไขเริ่มต้นให้อาร์เรย์ u มีเอลิเมนต์ 0.5<=x<=1 และ 0.5<=y<=1 มีค่าเท่ากับ 2
u[int(.5 / dy):int(1 / dy + 1),int(.5 / dx):int(1 / dx + 1)] = 2

fig = pyplot.figure()
ax = fig.gca(projection='3d')
X, Y = numpy.meshgrid(x, y)
surf = ax.plot_surface(X, Y, u, rstride=1, cstride=1, cmap=cm.viridis,
        linewidth=0, antialiased=False)

ax.set_xlim(0, 2)
ax.set_ylim(0, 2)
ax.set_zlim(1, 2.5)

ax.set_xlabel('$x$')
ax.set_ylabel('$y$');
```

Out[1]:



สร้างฟังก์ชัน diffuse(nt) จากสมการที่ (4.31) ดังนี้

In[2]:

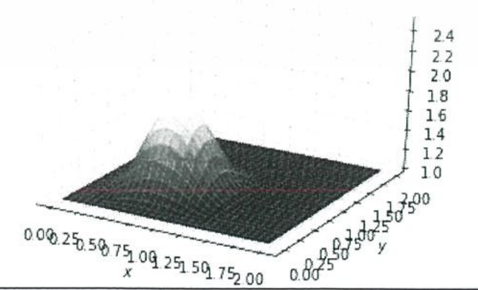
```
def diffuse(nt):
    u[int(.5 / dy):int(1 / dy + 1),int(.5 / dx):int(1 / dx + 1)] = 2

    for n in range(nt + 1):
        un = u.copy()
        u[1:-1, 1:-1] = (un[1:-1,1:-1] + nu * dt / dx**2 *  (un[1:-1, 2:] - 2 * un[1:-1, 1:-1] +
                        un[1:-1, 0:-2]) +nu * dt / dy**2 * (un[2:,1: -1] - 2 * un[1:-1, 1:-1] +
                        un[0:-2, 1:-1]))
        u[0, :] = 1
        u[-1, :] = 1
        u[:, 0] = 1
        u[:, -1] = 1

    fig = pyplot.figure()
    ax = fig.gca(projection='3d')
    surf = ax.plot_surface(X, Y, u[:], rstride=1, cstride=1, cmap=cm.viridis,linewidth=0,
        antialiased=True)
    ax.set_zlim(1, 2.5)
    ax.set_xlabel('$x$')
    ax.set_ylabel('$y$');
```

กำหนด 10 ขั้นเวลา

In[3]:

```
diffuse(10)
```

Out[3]:

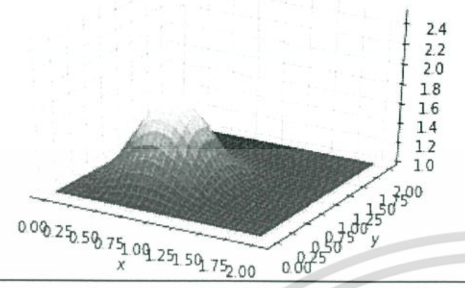


กำหนด 14 ขั้นเวลา

In[4]:

```
diffuse(14)
```

Out[4]:

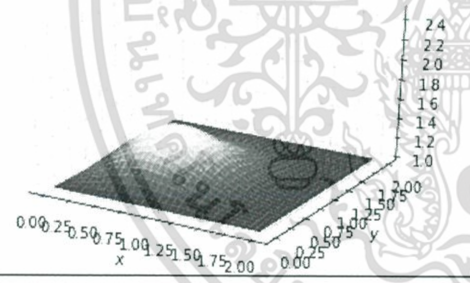


กำหนด 50 ขั้นเวลา

In[5]:

```
diffuse(50)
```

Out[5]:



4.3.8 ขั้นที่8 สมการของเบอร์เจอร์ในสองมิติ

สมการของเบอร์เจอร์ในสองมิติ คือ

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\upsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\upsilon\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right) \tag{4.37}$$

เมื่อใช้ผลต่างข้างหน้าในการหาอนุพันธ์เมื่อเทียบกับเวลา ใช้ผลต่างข้างหลังในการหาอนุพันธ์เมื่อเทียบกับระยะ และใช้พหุนามเทเลอร์อันดับสองสำหรับการประมาณค่าอนุพันธ์อันดับที่ 2 ดังนั้นได้ว่า

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{u_{i,j}^{n}-u_{i-1,j}^{n}}{\partial x}+v_{i}^{n}\frac{u_{i,j}^{n}-u_{i,j-1}^{n}}{\partial y}=\upsilon\left(\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{(\Delta x)^{2}}+\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{(\Delta y)^{2}}\right)$$

(4.38)

$$\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{v_{i,j}^{n}-v_{i-1,j}^{n}}{\partial x}+v_{i}^{n}\frac{v_{i,j}^{n}-v_{i,j-1}^{n}}{\partial y}=\upsilon\left(\frac{v_{i+1,j}^{n}-2v_{i,j}^{n}+v_{i-1,j}^{n}}{(\Delta x)^{2}}+\frac{v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n}}{(\Delta y)^{2}}\right)$$

จากสมการที่ (4.38) เมื่อทราบเงื่อนไขเริ่มต้น ตัวแปรที่ไม่ทราบค่าคือ u_i^{n+1} และ v_i^{n+1} สามารถเขียนสมการใหม่ได้ว่า

$$u_{i,j}^{n+1}=u_{i,j}^{n}-u_{i,j}^{n}\frac{\Delta t}{\Delta x}\left(u_{i,j}^{n}-u_{i-1,j}^{n}\right)-v_{i,j}^{n}\frac{\Delta t}{\Delta y}\left(u_{i,j}^{n}-u_{i-1,j}^{n}\right)+\frac{\upsilon\Delta t}{(\Delta x)^{2}}\left(u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}\right)$$
$$+\frac{\upsilon\Delta t}{(\Delta y)^{2}}\left(u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}\right)$$

(4.39)

$$v_{i,j}^{n+1}=v_{i,j}^{n}-u_{i,j}^{n}\frac{\Delta t}{\Delta x}\left(v_{i,j}^{n}-v_{i-1,j}^{n}\right)-v_{i,j}^{n}\frac{\Delta t}{\Delta y}\left(v_{i,j}^{n}-v_{i-1,j}^{n}\right)+\frac{\upsilon\Delta t}{(\Delta x)^{2}}\left(v_{i+1,j}^{n}-2v_{i,j}^{n}+v_{i-1,j}^{n}\right)$$
$$+\frac{\upsilon\Delta t}{(\Delta y)^{2}}\left(v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n}\right)$$

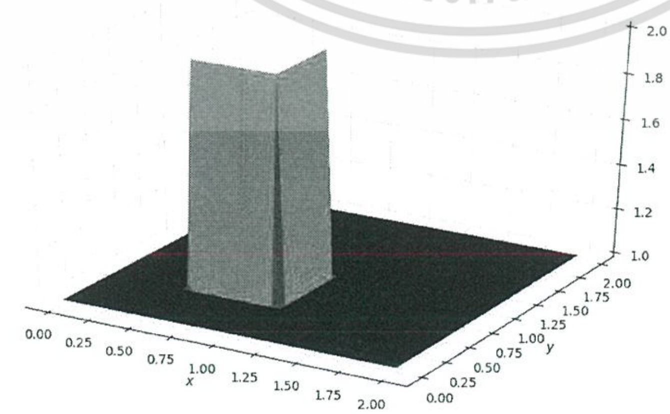
กำหนดเงื่อนไขเริ่มต้นได้ดังนี้

In[1]:

```
import numpy
from matplotlib import pyplot, cm
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline

nx = 41
ny = 41
nt = 120
c = 1
dx = 2 / (nx - 1)
dy = 2 / (ny - 1)
sigma = .0009
nu = 0.01
dt = sigma * dx * dy / nu
```

```
x = numpy.linspace(0, 2, nx)
y = numpy.linspace(0, 2, ny)

u = numpy.ones((ny, nx))
v = numpy.ones((ny, nx))
un = numpy.ones((ny, nx))
vn = numpy.ones((ny, nx))
comb = numpy.ones((ny, nx))

##กำหนดเงื่อนไขเริ่มต้น

#กำหนดให้อาร์เรย์ u มีเอลิเมนต์ที่ 0.5<=x<=1 และ 0.5<=y<=1 มีค่าเท่ากับ 2
u[int(.5 / dy):int(1 / dy + 1),int(.5 / dx):int(1 / dx + 1)] = 2
#กำหนดให้อาร์เรย์ v มีเอลิเมนต์ที่ 0.5<=x<=1 และ 0.5<=y<=1 มีค่าเท่ากับ 2
v[int(.5 / dy):int(1 / dy + 1),int(.5 / dx):int(1 / dx + 1)] = 2

#แสดงกราฟของเงื่อนไขเริ่มต้น
fig = pyplot.figure(figsize=(11, 7), dpi=100)
ax = fig.gca(projection='3d')
X, Y = numpy.meshgrid(x, y)
ax.plot_surface(X, Y, u[:], cmap=cm.viridis, rstride=1, cstride=1)
ax.plot_surface(X, Y, v[:], cmap=cm.viridis, rstride=1, cstride=1)
ax.set_xlabel('$x$')
ax.set_ylabel('$y$');
```

Out[1]:

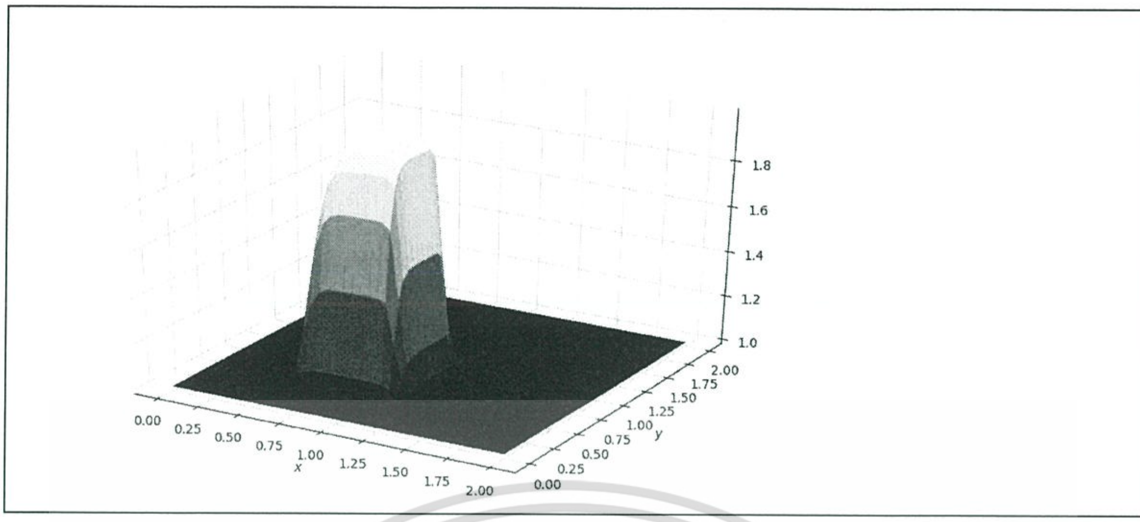


ดังนั้นสามารถหาคำตอบของสมการ (4.39) ได้ดังนี้

In[2]:

```
for n in range(nt + 1):
    un = u.copy()
    vn = v.copy()

    u[1:-1, 1:-1] = (un[1:-1, 1:-1] -dt / dx * un[1:-1, 1:-1] * (un[1:-1, 1:-1] - un[1:-1, 0:-2]) –
                    dt / dy * vn[1:-1, 1:-1] * (un[1:-1, 1:-1] - un[0:-2, 1:-1]) + nu * dt / dx**2
                    * (un[1:-1,2:] - 2 * un[1:-1, 1:-1] + un[1:-1, 0:-2]) + nu * dt / dy**2 *
                    (un[2:, 1:-1] - 2 * un[1:-1, 1:-1] + un[0:-2, 1:-1]))

    v[1:-1, 1:-1] = (vn[1:-1, 1:-1] - dt / dx * un[1:-1, 1:-1] * (vn[1:-1, 1:-1] - vn[1:-1, 0:-2]) –
                    dt / dy * vn[1:-1, 1:-1] * (vn[1:-1, 1:-1] - vn[0:-2, 1:-1]) + nu * dt / dx**2
                    * (vn[1:-1, 2:] - 2 * vn[1:-1, 1:-1] + vn[1:-1, 0:-2]) +nu * dt / dy**2 *
                    (vn[2:, 1:-1] - 2 * vn[1:-1, 1:-1] + vn[0:-2, 1:-1]))

    u[0, :] = 1
    u[-1, :] = 1
    u[:, -1] = 1

    v[-1, :] = 1
    v[:, 0] = 1
    v[:, -1] = 1

fig = pyplot.figure(figsize=(11, 7), dpi=100)
ax = fig.gca(projection='3d')
X, Y = numpy.meshgrid(x, y)
ax.plot_surface(X, Y, u, cmap=cm.viridis, rstride=1, cstride=1)
ax.plot_surface(X, Y, v, cmap=cm.viridis, rstride=1, cstride=1)
ax.set_xlabel('$x$')
ax.set_ylabel('$y$');
```

Out[2]:



4.3.9 ขั้นที่9 สมการลาปลาสสองมิติ

สมการลาปลาสในสองมิติ คือ

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=0 \tag{4.40}$$

เมื่อใช้ผลต่างกลางสำหรับการประมาณค่าอนุพันธ์อันดับที่ 2

$$\frac{p^n_{i+1,j}-2p^n_{i,j}+p^n_{i-1,j}}{\Delta x^2}+\frac{p^n_{i,j+1}-2p^n_{i,j}+p^n_{i,j-1}}{\Delta y^2}=0 \tag{4.41}$$

เนื่องจากสมการลาปลาสเป็นอิสระจากเวลา สมการนี้จึงไม่มีพจน์ p^{n+1} เหมือนในขั้นตอนก่อนหน้าที่คลื่นเปลี่ยนแปลงตามเวลา โดยสมการลาปลาสคำนวณที่สภาวะสมดุลของระบบภายใต้เงื่อนไขขอบเขตที่ให้มา โดยคำนวณระบบที่เวลาหนึ่ง แล้วทำซ้ำหาค่าของพจน์ $p^n_{i,j}$ จนกว่าจะตรงตามเงื่อนไขที่ได้กำหนดไว้ ระบบจะสมดุลก็ต่อเมื่อจำนวนการทำซ้ำนั้นเข้าสู่อนันต์ แต่สามารถประมาณสถานะสมดุลได้โดยการทำซ้ำจนกว่าความแตกต่างระหว่างการทำซ้ำครั้งแรกและครั้งถัดไปมีค่าน้อยมาก

จากสมการที่ (4.41) จัดรูปสมการใหม่เพื่อหาค่าของพจน์ $p^n_{i,j}$

$$p^n_{i,j}=\frac{\Delta y^2\left(p^n_{i+1,j}+p^n_{i-1,j}\right)+\Delta x^2\left(p^n_{i,j+1}+p^n_{i,j-1}\right)}{2\left(\Delta x^2+\Delta y^2\right)} \tag{4.42}$$

กำหนดที่สภาวะเริ่มต้น $p=0$ ทุกที่ และมีเงื่อนไขที่ขอบ $p=0$ ที่ $x=0$, $p=y$ ที่ $x=2$ และ $\frac{\partial p}{\partial y}=0$ ที่ $y=0,1$

ต่อไปเป็นการสร้างฟังก์ชัน plot2D(x, y, p) เพื่อสั่งให้วาดกราฟสามมิติ

In[1]:

```
import numpy
from matplotlib import pyplot, cm
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline

def plot2D(x, y, p):        #สร้างฟังก์ชัน plot2D(x, y, p)
    fig = pyplot.figure(figsize=(11, 7), dpi=100)
    ax = fig.gca(projection='3d')
    X, Y = numpy.meshgrid(x, y)
    surf = ax.plot_surface(X, Y, p[:], rstride=1, cstride=1, cmap=cm.viridis, linewidth=0,
            antialiased=False)
    ax.set_xlim(0, 2)
    ax.set_ylim(0, 1)
    ax.view_init(30, 225)
    ax.set_xlabel('$x$')
    ax.set_ylabel('$y$')
```

จากสมการที่ (4.42)

$$p_{i,j}^{n} = \frac{\Delta y^2\left(p_{i+1,j}^{n} + p_{i-1,j}^{n}\right) + \Delta x^2\left(p_{i,j+1}^{n} + p_{i,j-1}^{n}\right)}{2\left(\Delta x^2 + \Delta y^2\right)} \tag{4.42}$$

ต่อไปเป็นการสร้างฟังก์ชัน laplace2d(p, y, dx, dy, l1norm_target) โดยสั่งให้ทำซ้ำในการค่า p จนกระทั่งค่า p อยู่ในเงื่อนไขที่กำหนดไว้

In[2]:

```
def laplace2d(p, y, dx, dy, l1norm_target):
    l1norm = 1
    pn = numpy.empty_like(p)

    while l1norm > l1norm_target:
        pn = p.copy()
        p[1:-1, 1:-1] = ((dy**2 * (pn[1:-1, 2:] + pn[1:-1, 0:-2]) + dx**2 * (pn[2:, 1:-1] +
                        pn[0:-2, 1:-1])) / (2 * (dx**2 + dy**2)))
                        #กำหนดค่าเอลิเมนต์ที่ไม่ใช่ที่ขอบจากสมการ (4.42)
```

```
        #กำหนดค่าเอลิเมนต์ที่ขอบ
        p[:, 0] = 0  # p = 0 @ x = 0
        p[:, -1] = y  # p = y @ x = 2
        p[0, :] = p[1, :]  # dp/dy = 0 @ y = 0
        p[-1, :] = p[-2, :]  # dp/dy = 0 @ y = 1
        l1norm = (numpy.sum(numpy.abs(p[:]) - numpy.abs(pn[:])) /numpy.sum
                (numpy.abs(pn[:])))

    return p
```

กำหนดค่าเงื่อนไขขอบและเงื่อนไขเริ่มต้น ดังต่อไปนี้

In[3]:

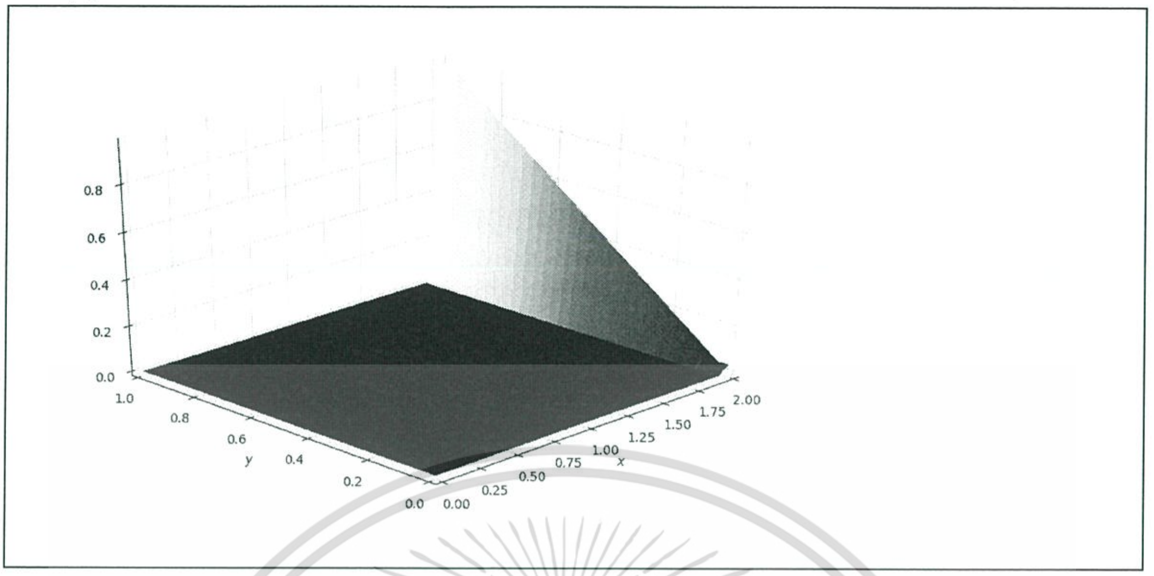
```
nx = 31
ny = 31
c = 1
dx = 2 / (nx - 1)
dy = 2 / (ny - 1)

x = numpy.linspace(0, 2, nx)
y = numpy.linspace(0, 1, ny)

#กำหนดเงื่อนไขเริ่มต้น
p = numpy.zeros((ny, nx))

#กำหนดเงื่อนไขขอบ
p[:, 0] = 0  # p = 0 @ x = 0
p[:, -1] = y  # p = y @ x = 2
p[0, :] = p[1, :]  # dp/dy = 0 @ y = 0
p[-1, :] = p[-2, :]  # dp/dy = 0 @ y = 1
```

ใช้ฟังก์ชัน plot2D(x, y, p) วาดกราฟของเงื่อนไขเริ่มต้นที่ค่า $p=0$ ทุกที่ ยกเว้นที่ $x=2$ ที่มี $p=y$

In[4]:

```
plot2D(x, y, p)
```

Out[4]:

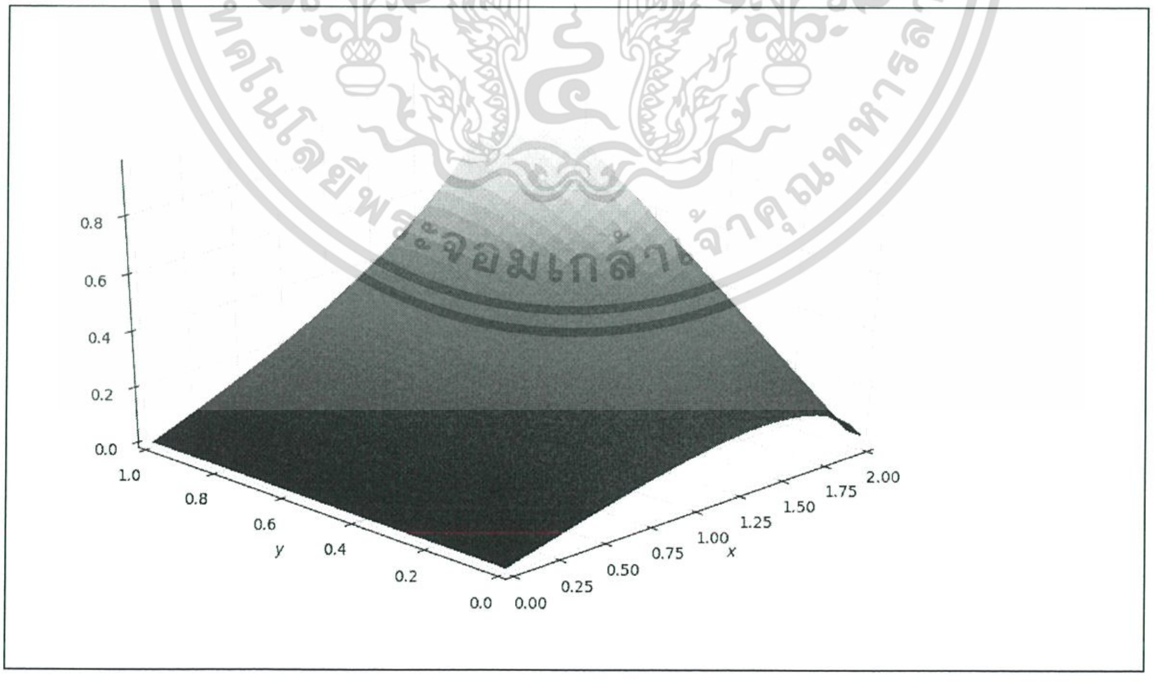


ต่อไปเป็นการดำเนินการฟังก์ชัน laplace2d(p, y, dx, dy, l1norm_target) โดยกำหนดค่าของ l1norm_target ที่มีค่าเท่ากับ 0.0001

In[5]:

```
p = laplace2d(p, y, dx, dy, 1e-4)
plot2D(x, y, p)
```

Out[5]:



4.3.10 ขั้นที่10 สมการปัวซองสองมิติ

ในสมการของปัวซองมีการเพิ่มพจน์แหล่งกำเนิดไปยังข้างขวาของสมการลาปลาส

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=b \tag{4.43}$$

เมื่อใช้ผลต่างกลางสำหรับการประมาณค่าอนุพันธ์อันดับที่ 2

$$\frac{p_{i+1,j}^n-2p_{i,j}^n+p_{i-1,j}^n}{\Delta x^2}+\frac{p_{i,j+1}^n-2p_{i,j}^n+p_{i,j-1}^n}{\Delta y^2}=b_{i,j}^n \tag{4.44}$$

จัดรูปใหม่เพื่อหาค่าของพจน์ $p_{i,j}^n$

$$p_{i,j}^n=\frac{\left(p_{i+1,j}^n+p_{i-1,j}^n\right)\Delta y^2+\left(p_{i,j+1}^n+p_{i,j-1}^n\right)\Delta x^2-b_{i,j}^n\Delta x^2\Delta y^2}{2\left(\Delta x^2+\Delta y^2\right)} \tag{4.45}$$

กำหนดค่าเริ่มต้น $p=0$ ทุกที่ และมีเงื่อนไขขอบ $p=0$ ที่ $x=0,2$ และ $y=0,1$ และพจน์ของแหล่งกำเนิดประกอบด้วย 2 เงื่อนไขในโดเมน ดังนี้

$b_{i,j}=100$ ที่ $i=\frac{1}{4}nx, j=\frac{1}{4}ny$

$b_{i,j}=-100$ ที่ $i=\frac{3}{4}nx, j=\frac{3}{4}ny$

$b_{i,j}=0$ ในที่อื่นๆ

In[1]:

```
import numpy
from matplotlib import pyplot, cm
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline


nx = 50
ny = 50
nt  = 100
xmin = 0
xmax = 2
ymin = 0
ymax = 1
```

```
dx = (xmax - xmin) / (nx - (1
dy = (ymax - ymin) / (ny - (1

#กำหนดเงื่อนไขขอบและเงื่อนไขเริ่มต้น
p  = numpy.zeros((ny, nx))
pd = numpy.zeros((ny, nx))
b  = numpy.zeros((ny, nx))
x  = numpy.linspace(xmin, xmax, nx)
y  = numpy.linspace(xmin, xmax, ny)

#กำหนดเงื่อนไขแหล่งกำเนิด
b[int(ny / (4, int(nx / 100 =  [(4
b[int( * 3ny / (4, int( * 3nx /  100- = [(4
```

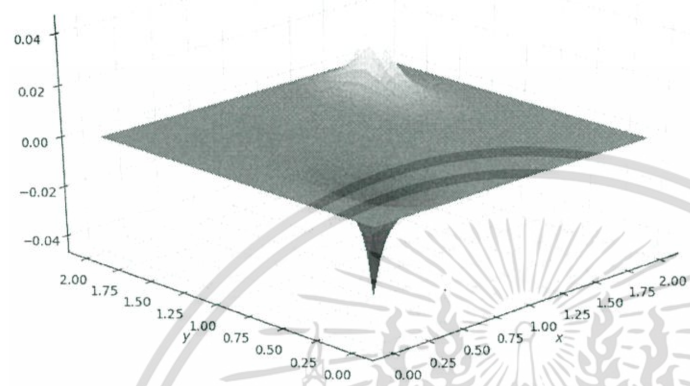
เหมือนเดิม

In[2]:

```
for it in range(nt):

        pd = p.copy()

        p[1:-1,1:-1] = (((pd[1:-1, 2:] + pd[1:-1, :-2]) * dy**2 + (pd[2:, 1:-1] + pd[:-2, 1:-1]) *
                        dx**2 -b[1:-1, 1:-1] * dx**2 * dy**2) / (2 * (dx**2 + dy**2)))

    p[0, :] = 0
    p[ny-1, :] = 0
    p[:, 0] = 0
    p[:, nx-1] = 0

#กำหนดฟังก์ชัน plot2D(x, y, p) เหมือนขั้นตอนที่ 9
def plot2D(x, y, p):
    fig = pyplot.figure(figsize=(11, 7), dpi=100)
    ax = fig.gca(projection='3d')
    X, Y = numpy.meshgrid(x, y)
    surf = ax.plot_surface(X, Y, p[:], rstride=1, cstride=1, cmap=cm.viridis,
            linewidth=0, antialiased=False)
    ax.view_init(30, 225)
    ax.set_xlabel('$x$')
```

```
    ax.set_ylabel('$y$')


plot2D(x, y, p)
```

Out[2]:



4.3.11 ขั้นที่11 การไหลผ่านช่องด้วยสมการนาเวียร์-สโตกส์

สมการควบคุมสำหรับการไหลในช่อง ประกอบด้วยสมการการอนุรักษ์โมเมนตัมในทิศองค์ประกอบของความเร็ว u, v และสมการความดันของปัวซอง ดังนี้

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\upsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\upsilon\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right) \quad (4.46)$$

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=-\rho\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x}+2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}+\frac{\partial v}{\partial y}\frac{\partial v}{\partial y}\right)$$

สมการโมเมนตัมในทิศ u คือ

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x}+\upsilon_{i,j}^{n}\frac{u_{i,j}^{n}-u_{i,j-1}^{n}}{\Delta y}=-\frac{1}{\rho}\frac{p_{i+1,j}^{n}-p_{i-1,j}^{n}}{2\Delta x}+\nu\left(\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^2}+\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\Delta y^2}\right) \quad (4.47)$$

สมการโมเมนตัมในทิศ v คือ

$$\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{v_{i,j}^{n}+v_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\frac{v_{i,j}^{n}-v_{i,j-1}^{n}}{\Delta y}=-\frac{1}{\rho}\frac{p_{i,j+1}^{n}-p_{i,j-1}^{n}}{2\Delta y}+\upsilon\left(\frac{v_{i+1,j}^{n}-2v_{i,j}^{n}+v_{i-1,j}^{n}}{\Delta x^{2}}+\frac{v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n}}{\Delta y^{2}}\right) \tag{4.48}$$

และ สมการความดันของปัวซอง คือ

$$\frac{p_{i+1,j}^{n}-2p_{i,j}^{n}+p_{i-1,j}^{n}}{\Delta x^{2}}+\frac{p_{i,j+1}^{n}-2p_{i,j}^{n}+p_{i,j-1}^{n}}{\Delta y^{2}}=\rho\left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}+\frac{v_{i,j+1}-v_{i,j-1}}{2\Delta y}\right)-\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}-2\frac{u_{i,j+1}-u_{i,j-1}}{2\Delta y}\frac{v_{i+1,j}-v_{i-1,j}}{2\Delta x}-\frac{v_{i,j+1}-v_{i,j-1}}{2\Delta y}\frac{v_{i,j+1}-v_{i,j-1}}{2\Delta y}\right] \tag{4.49}$$

จัดรูปสมการเหล่านี้ใหม่เพื่อหาค่าของพจน์ของ $u_{i,j}^{n+1}$, $v_{i,j}^{n+1}$, $p_{i,j}^{n+1}$ ได้ดังนี้

สมการโมเมนตัมในทิศ u

$$u_{i,j}^{n+1}=u_{i,j}^{n}-u_{i,j}^{n}\frac{\Delta t}{\Delta x}\left(u_{i,j}^{n}-u_{i-1,j}^{n}\right)-v_{i,j}^{n}\frac{\Delta t}{\Delta y}\left(u_{i,j}^{n}-u_{i,j-1}^{n}\right)-\frac{\Delta t}{\rho 2\Delta x}\left(p_{i+1,j}^{n}-p_{i-1,j}^{n}\right)+\upsilon\left[\frac{\Delta t}{\Delta x^{2}}(u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n})+\frac{\Delta t}{\Delta y^{2}}(u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n})\right] \tag{4.50}$$

สมการโมเมนตัมในทิศ v

$$v_{i,j}^{n+1}=v_{i,j}^{n}-u_{i,j}^{n}\frac{\Delta t}{\Delta x}\left(v_{i,j}^{n}-v_{i-1,j}^{n}\right)-v_{i,j}^{n}\frac{\Delta t}{\Delta y}\left(v_{i,j}^{n}-v_{i,j-1}^{n}\right)-\frac{\Delta t}{\rho 2\Delta y}\left(p_{i,j+1}^{n}-p_{i,j-1}^{n}\right)+\upsilon\left[\frac{\Delta t}{\Delta x^{2}}(v_{i+1,j}^{n}-v_{i,j}^{n}+v_{i-1,j}^{n})+\frac{\Delta t}{\Delta y^{2}}(v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n})\right] \tag{4.51}$$

สมการความดันปัวซอง

$$p_{i,j}^n = \frac{\left(p_{i+1,j}^n + p_{i-1,j}^n\right)\Delta y^2 + \left(p_{i,j+1}^n + p_{i,j-1}^n\right)\Delta x^2}{2\left(\Delta x^2 + \Delta y^2\right)} - \frac{\rho \Delta x^2 \Delta y^2}{2\left(\Delta x^2 + \Delta y^2\right)} \times$$
$$\left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right) - \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}\right. \qquad (4.52)$$
$$\left. -2\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} - \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right]$$

กำหนดเงื่อนไขเริ่มต้นให้ทุกตำแหน่งมีค่า $u, v, p = 0$ และเงื่อนไขขอบได้แก่

$u = 1$ ที่ $y = 2$

$u, v = 0$ ที่ $y \neq 2$

$\frac{\partial p}{\partial y} = 0$ ที่ $y = 0$

$p = 0$ ที่ $y = 2$

$\frac{\partial p}{\partial x} = 0$ ที่ $x = 0, 2$

การดำเนินการของการไหลในท่อปิด เริ่มจากกำหนดค่าตัวแปร ความเร็วคลื่น $1\, m/s$, ความหนาแน่นของของไหล $1\, kg/m^3$, ความหนืด $0.1\, m^2/s$ และกำหนดโดเมน $0 \leq x \leq 2$ และ $0 \leq y \leq 2$ จำนวนจุดกริดของ x และ y จำนวน 41 จุด จำนวนขั้นเวลา 500 ขั้น และขั้นเวลาเทียมสำหรับคำนวณความดันในสมการของปัวซองจำนวน 50 ขั้น

In[1]:

```
import numpy
from matplotlib import pyplot, cm
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline

nx = 41
ny = 41
nt = 500
nit = 50
c = 1
dx = 2 / (nx - 1)
dy = 2 / (ny - 1)
x = numpy.linspace(0, 2, nx)
y = numpy.linspace(0, 2, ny)
X, Y = numpy.meshgrid(x, y)
rho = 1
```

```
nu = .1
dt = .001

u = numpy.zeros((ny, nx))
v = numpy.zeros((ny, nx))
p = numpy.zeros((ny, nx))
b = numpy.zeros((ny, nx))
```

ในการคำนวณความดันในสมการของปัวซอง เริ่มจากสร้างฟังก์ชัน build_up_b(b, rho, dt, u, v, dx, dy) สำหรับค่าของพจน์ฝั่งขวาของสมการที่ (4.49) ที่ตำแหน่งที่ไม่ใช่ขอบ

In[2]:

```
def build_up_b(b, rho, dt, u, v, dx, dy):

    b[1:-1, 1:-1] = (rho * (1 / dt * ((u[1:-1, 2:] - u[1:-1, 0:-2]) / (2 * dx) + (v[2:, 1:-1] - v[0:-2,
                    1:-1]) / (2 * dy)) - ((u[1:-1, 2:] - u[1:-1, 0:-2]) / (2 * dx))**2 - 2 * ((u[2:,
                    1:-1] - u[0:-2, 1:-1]) / (2 * dy) * (v[1:-1, 2:] - v[1:-1, 0:-2]) / (2 * dx))
                    - ((v[2:, 1:-1] - v[0:-2, 1:-1]) / (2 * dy))**2))

    return b
```

สร้างฟังก์ชัน pressure_poisson(p, dx, dy, b) เพื่อหาค่าความดันตามสมการที่ (4.52) ที่ตำแหน่งที่ไม่ใช่ขอบ และความดันที่ขอบจากเงื่อนไขขอบ

In[3]:

```
def pressure_poisson(p, dx, dy, b):
    pn = numpy.empty_like(p)
    pn = p.copy()

    for q in range(nit):
        pn = p.copy()
        p[1:-1, 1:-1] = (((pn[1:-1, 2:] + pn[1:-1, 0:-2]) * dy**2 + (pn[2:, 1:-1] + pn[0:-2,
                        1:-1]) * dx**2) / (2 * (dx**2 + dy**2)) - dx**2 * dy**2 / (2 * (dx**2 +
                        dy**2)) * b[1:-1,1:-1])

        p[:, -1] = p[:, -2]     #จาก dp/dy = 0 ที่ x = 2
        p[0, :] = p[1, :]       #จาก dp/dy = 0 ที่ y = 0
        p[:, 0] = p[:, 1]       #จาก dp/dx = 0 ที่ x = 0
```

```
        p[-1, :] = 0            #จาก p = 0 ที่ y = 2

    return p
```

เมื่อหาคำตอบของสมการความดันของปัวซองได้แล้ว สร้างฟังก์ชัน cavity_flow(nt, u, v, dt, dx, dy, p, rho, nu) เพื่อหาคำตอบของสมการการอนุรักษ์โมเมนตัมในสมการที่ (4.50) และ (4.51)

In[4]:

```
def cavity_flow(nt, u, v, dt, dx, dy, p, rho, nu):
    un = numpy.empty_like(u)
    vn = numpy.empty_like(v)
    b = numpy.zeros((ny, nx))

    for n in range(nt):
        un = u.copy()
        vn = v.copy()

        b = build_up_b(b, rho, dt, u, v, dx, dy)
        p = pressure_poisson(p, dx, dy, b)

        u[1:-1, 1:-1] = (un[1:-1, 1:-1]-un[1:-1, 1:-1] * dt / dx * (un[1:-1, 1:-1] - un[1:-1,
                        0:-2]) -vn[1:-1, 1:-1] * dt / dy * (un[1:-1, 1:-1] - un[0:-2, 1:-1]) -dt /
                        (2 * rho * dx) * (p[1:-1, 2:] - p[1:-1, 0:-2]) +nu * (dt / dx**2 *(un[1:
                        -1, 2:] - 2 * un[1:-1, 1:-1] + un[1:-1, 0:-2]) +dt / dy**2 * (un[2:, 1:
                        -1] - 2 * un[1:-1, 1:-1] + un[0:-2, 1:-1])))

        v[1:-1,1:-1] = (vn[1:-1, 1:-1] -un[1:-1, 1:-1] * dt / dx * (vn[1:-1, 1:-1] - vn[1:-1, 0:-2])
                       -vn[1:-1, 1:-1] * dt / dy * (vn[1:-1, 1:-1] - vn[0:-2, 1:-1]) -dt / (2 * rho
                       * dy) * (p[2:, 1:-1] - p[0:-2, 1:-1]) +nu * (dt / dx**2 * (vn[1:-1, 2:] - 2
                        * vn[1:-1, 1:-1] + vn[1:-1, 0:-2]) +dt / dy**2 * (vn[2:, 1:-1] - 2 * vn[1:
                     -1, 1:-1] + vn[0:-2, 1:-1])))

        u[0, :] = 0
        u[:, 0] = 0
        u[:, -1] = 0
        u[-1, :] = 1
        v[0, :] = 0
        v[-1, :]=0
```

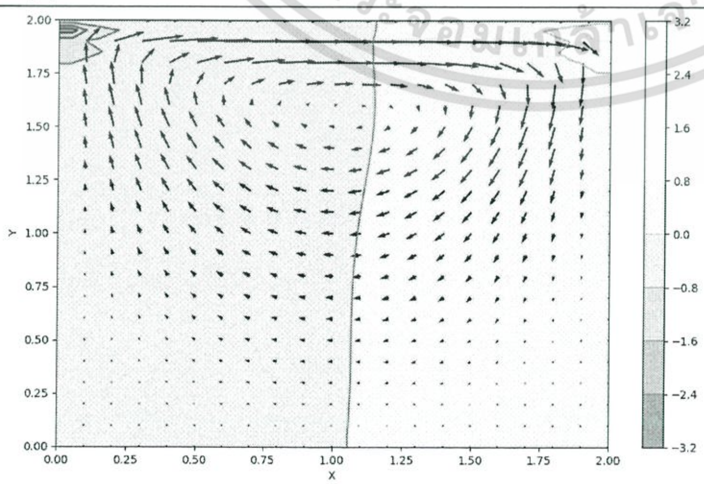
```
        v[:, 0] = 0
        v[:, -1] = 0

    return u, v, p
```

พิจารณาผลเมื่อระบบเสถียร กำหนดจำนวนขั้นเวลา nt มี 700 ขั้น

In[5]:

```
u = numpy.zeros((ny, nx))
v = numpy.zeros((ny, nx))
p = numpy.zeros((ny, nx))
b = numpy.zeros((ny, nx))
nt = 700
u, v, p = cavity_flow(nt, u, v, dt, dx, dy, p, rho, nu)

fig = pyplot.figure(figsize=(11,7), dpi=100)
#วาดแผนภาพสีคอนทัวร์ของสนามความดัน
pyplot.contourf(X, Y, p, alpha=0.5, cmap=cm.viridis)
pyplot.colorbar()
#วาดเส้นคอนทัวร์ของสนามความดัน
pyplot.contour(X, Y, p, cmap=cm.viridis)
#วาดสนามเวกเตอร์ความเร็ว
pyplot.quiver(X[::2, ::2], Y[::2, ::2], u[::2, ::2], v[::2, ::2])
pyplot.xlabel('X')
pyplot.ylabel('Y');
```

Out[5]:

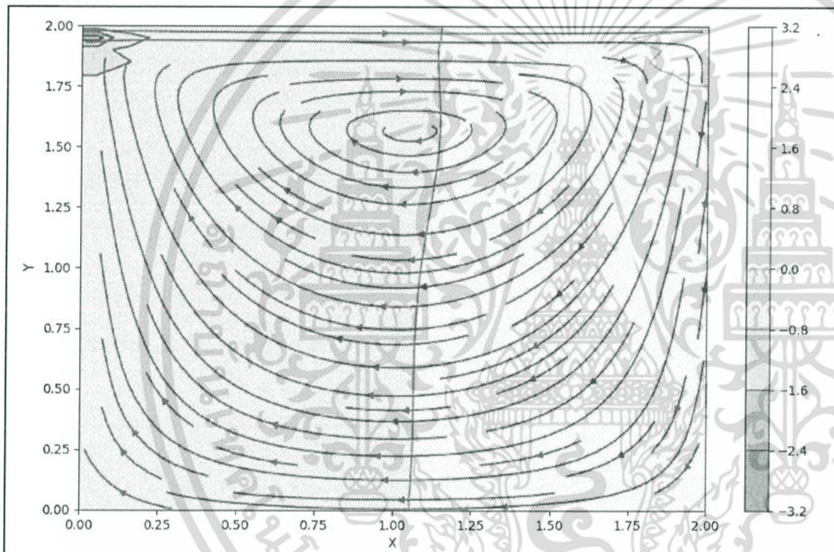


อีกวิธีหนึ่งในการมองเห็นการไหลคือการใช้การสร้างเส้นการไหล

In[7]:

```
fig = pyplot.figure(figsize=(11, 7), dpi=100)
pyplot.contourf(X, Y, p, alpha=0.5, cmap=cm.viridis)
pyplot.colorbar()
pyplot.contour(X, Y, p, cmap=cm.viridis)
pyplot.streamplot(X, Y, u, v)       #สร้างเส้นการไหล
pyplot.xlabel('X')
pyplot.ylabel('Y');
```

Out[7]:



4.3.12 ขั้นที่12 การไหลผ่านท่อด้วยสมการนาเวียร์-สโตกส์

ขั้นตอนที่ 12 ต่างจากขั้นตอนที่ 11 ที่มีพจน์ต้นกำเนิดเพิ่มขึ้นในสมการการอนุรักษ์โมเมนตัมในทิศ u เพื่อจำลองการไหลในช่องซึ่งเป็นผลของความดัน สมการนาเวียร์-สโตกส์สำหรับการไหลในช่องคือ

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\upsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+F$$

(4.53)

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\upsilon\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)$$

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=-\rho\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x}+2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}+\frac{\partial v}{\partial y}\frac{\partial v}{\partial y}\right)$$

สมการการอนุรักษ์โมเมนตัมในทิศ u

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\frac{u_{i,j}^{n}-u_{i,j-1}^{n}}{\Delta y}=-\frac{1}{\rho}\frac{p_{i+1,j}^{n}-p_{i-1,j}^{n}}{2\Delta x}+\nu\left(\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^2}+\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\Delta y^2}\right)+F_{i,j} \quad (4.54)$$

สมการการอนุรักษ์โมเมนตัมในทิศ v

$$\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{v_{i,j}^{n}+v_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\frac{v_{i,j}^{n}-v_{i,j-1}^{n}}{\Delta y}=-\frac{1}{\rho}\frac{p_{i,j+1}^{n}-p_{i,j-1}^{n}}{2\Delta y}+\nu\left(\frac{v_{i+1,j}^{n}-2v_{i,j}^{n}+v_{i-1,j}^{n}}{\Delta x^2}+\frac{v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n}}{\Delta y^2}\right) \quad (4.55)$$

สมการความดันของปัวซอง

$$\frac{p_{i+1,j}^{n}-2p_{i,j}^{n}+p_{i-1,j}^{n}}{\Delta x^2}+\frac{p_{i,j+1}^{n}-2p_{i,j}^{n}+p_{i,j-1}^{n}}{\Delta y^2}=\rho\left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}+\frac{v_{i,j+1}-v_{i,j-1}}{2\Delta y}\right)-\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}-2\frac{u_{i,j+1}-u_{i,j-1}}{2\Delta y}\frac{v_{i+1,j}-v_{i-1,j}}{2\Delta x}-\frac{v_{i,j+1}-v_{i,j-1}}{2\Delta y}\frac{v_{i,j+1}-v_{i,j-1}}{2\Delta y}\right] \quad (4.56)$$

จัดรูปสมการเหล่านี้ใหม่เพื่อหาค่าของพจน์ของ $u_{i,j}^{n+1}$, $v_{i,j}^{n+1}$, $p_{i,j}^{n+1}$ ได้ดังนี้

สมการโมเมนตัมในทิศ u

$$u_{i,j}^{n+1}=u_{i,j}^{n}-u_{i,j}^{n}\frac{\Delta t}{\Delta x}\left(u_{i,j}^{n}-u_{i-1,j}^{n}\right)-v_{i,j}^{n}\frac{\Delta t}{\Delta y}\left(u_{i,j}^{n}-u_{i,j-1}^{n}\right)-\frac{\Delta t}{\rho 2\Delta x}\left(p_{i+1,j}^{n}-p_{i-1,j}^{n}\right)+\nu\left[\frac{\Delta t}{\Delta x^2}(u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n})+\frac{\Delta t}{\Delta y^2}(u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n})\right]+\Delta tF \quad (4.57)$$

สมการโมเมนตัมในทิศ v

$$v_{i,j}^{n+1} = v_{i,j}^{n} - u_{i,j}^{n}\frac{\Delta t}{\Delta x}\left(v_{i,j}^{n} - v_{i-1,j}^{n}\right) - v_{i,j}^{n}\frac{\Delta t}{\Delta y}\left(v_{i,j}^{n} - v_{i,j-1}^{n}\right) - \frac{\Delta t}{\rho 2\Delta y}\left(p_{i,j+1}^{n} - p_{i,j-1}^{n}\right) + \upsilon\left[\frac{\Delta t}{\Delta x^2}(v_{i+1,j}^{n} - 2v_{i,j}^{n} + v_{i-1,j}^{n}) + \frac{\Delta t}{\Delta y^2}(v_{i,j+1}^{n} - 2v_{i,j}^{n} + v_{i,j-1}^{n})\right] \tag{4.58}$$

สมการความดันของปัวซอง

$$p_{i,j}^{n} = \frac{\left(p_{i+1,j}^{n} + p_{i-1,j}^{n}\right)\Delta y^2 + \left(p_{i,j+1}^{n} + p_{i,j-1}^{n}\right)\Delta x^2}{2\left(\Delta x^2 + \Delta y^2\right)} - \frac{\rho\Delta x^2\Delta y^2}{2\left(\Delta x^2 + \Delta y^2\right)} \times \left[\frac{1}{\Delta t}\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right) - \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} - 2\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} - \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}\right] \tag{4.59}$$

โดยกำหนดเงื่อนไขเริ่มต้น $u, \upsilon, p = 0$ ในทุกตำแหน่ง และเงื่อนไขขอบได้แก่
u, v, p มีค่าเท่ากันที่ตำแหน่ง x เท่ากับ 0 และ 2
$u, v = 0$ ที่ y เท่ากับ 0 และ 2
$\frac{\partial p}{\partial y} = 0$ ที่ y เท่ากับ 0 และ 2
$F = 1$ ตลอดทั้งช่วง

เข้าสู่ขั้นตอนการเขียนโปรแกรม เริ่มจากการนำเข้าไลบรารีที่จำเป็นต้องใช้

In[1]:

```
import numpy
from matplotlib import pyplot, cm
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline
```

ในการคำนวณความดันในสมการของปัวซอง เริ่มจากสร้างฟังก์ชัน build_up_b(rho, dt, dx, dy, u, v) สำหรับค่าของพจน์ฝั่งขวาของสมการที่ (4.56) ที่ตำแหน่งที่ไม่ใช่ขอบ และเงื่อนไขขอบเป็นคาบ

In [2]:

```
def build_up_b(rho, dt, dx, dy, u, v):
    b = numpy.zeros_like(u)
    b[1:-1, 1:-1] = (rho * (1 / dt * ((u[1:-1, 2:] - u[1:-1, 0:-2]) / (2 * dx) + (v[2:, 1:-1] - v[0:-2,
                    1:-1]) / (2 * dy)) - ((u[1:-1, 2:] - u[1:-1, 0:-2]) / (2 * dx))**2 -2 * ((u[2:, 1:-1]
                    - u[0:-2, 1:-1]) / (2 * dy) * (v[1:-1, 2:] - v[1:-1, 0:-2]) / (2 * dx))- ((v[2:, 1:-1]
```

```
                - v[0:-2, 1:-1]) / (2 * dy))**2))

    #เงื่อนไขขอบแบบคาบ ที่ x = 2
    b[1:-1, -1] = (rho * (1 / dt * ((u[1:-1, 0] - u[1:-1,-2]) / (2 * dx) + (v[2:, -1] - v[0:-2, -1]) /
                (2 * dy)) - ((u[1:-1, 0] - u[1:-1, -2]) / (2 * dx))**2 -2 * ((u[2:, -1] - u[0:-2, -1])
                / (2 * dy) * (v[1:-1, 0] - v[1:-1, -2]) / (2 * dx)) -((v[2:, -1] - v[0:-2, -1]) / (2 *
                dy))**2))

    #เงื่อนไขขอบแบบคาบ ที่ x = 0
    b[1:-1, 0] = (rho * (1 / dt * ((u[1:-1, 1] - u[1:-1, -1]) / (2 * dx) + (v[2:, 0] - v[0:-2, 0]) / (2
                * dy)) - ((u[1:-1, 1] - u[1:-1, -1]) / (2 * dx))**2 -2 * ((u[2:, 0] - u[0:-2, 0]) / (2 *
                dy) * (v[1:-1, 1] - v[1:-1, -1]) / (2 * dx))- ((v[2:, 0] - v[0:-2, 0]) / (2 * dy))**2))

    return b
```

สร้างฟังก์ชัน pressure_poisson(p, dx, dy) เพื่อหาค่าความดันตามสมการที่ (4.59) ที่ตำแหน่งที่ไม่ใช่ขอบ และความดันที่ขอบจากเงื่อนไขขอบ

In [3]:

```
def pressure_poisson_periodic(p, dx, dy):
    pn = numpy.empty_like(p)

    for q in range(nit):
        pn = p.copy()
        p[1:-1, 1:-1] = (((pn[1:-1, 2:] + pn[1:-1, 0:-2]) * dy**2 + (pn[2:, 1:-1] + pn[0:-2, 1:
                        -1]) * dx**2) / (2 * (dx**2 + dy**2)) - dx**2 * dy**2 / (2 * (dx**2 +
                        dy**2)) * b[1:-1, 1:-1])

        #เงื่อนไขขอบแบบคาบของความดัน ที่ x = 2
        p[1:-1, -1] = (((pn[1:-1, 0] + pn[1:-1, -2])* dy**2 + (pn[2:, -1] + pn[0:-2, -1]) *
                        dx**2) / (2 * (dx**2 + dy**2)) -dx**2 * dy**2 / (2 * (dx**2 + dy**2))
                        * b[1:-1, -1])

        #เงื่อนไขขอบแบบคาบของความดัน ที่ x = 0
        p[1:-1, 0] = (((pn[1:-1, 1] + pn[1:-1, -1])* dy**2 + (pn[2:, 0] + pn[0:-2, 0]) * dx**2)
                        / (2 * (dx**2 + dy**2)) -dx**2 * dy**2 / (2 * (dx**2 + dy**2)) * b[1:
                        -1, 0])
```

```
        #เงื่อนไขขอบของความดันที่ติดกับผนัง
        p[-1, :] =p[-2, :]      #dp/dy = 0 ที่ y = 2
        p[0, :] = p[1, :]       #dp/dy = 0 ที่ y = 0

    return p
```

กำหนดค่าตัวแปร ความเร็วคลื่น $1\,m/s$, ความหนาแน่นของของไหล $1\,kg/m^3$, ความหนืด $0.1\,m^2/s$ และกำหนดโดเมน $0 \le x \le 2$ และ $0 \le y \le 2$ จำนวนจุดกริดของ x และ y จำนวน 41 จุด จำนวนขั้นเวลา 10 ขั้น และขั้นเวลาเทียมสำหรับคำนวณความดันในสมการของปัวซองจำนวน 50 ขั้น

In [4]:

```
nx = 41
ny = 41
nt = 10
nit = 50
c = 1
dx = 2 / (nx - 1)
dy = 2 / (ny - 1)
x = numpy.linspace(0, 2, nx)
y = numpy.linspace(0, 2, ny)
X, Y = numpy.meshgrid(x, y)
rho = 1
nu = .1
F = 1
dt = .01

#กำหนดเงื่อนไขเริ่มต้น
u = numpy.zeros((ny, nx))
un = numpy.zeros((ny, nx))

v = numpy.zeros((ny, nx))
vn = numpy.zeros((ny, nx))

p = numpy.ones((ny, nx))
pn = numpy.ones((ny, nx))

b = numpy.zeros((ny, nx))
```

จากขั้นตอนที่ 9 สำหรับสมการลาปลาซ กำหนดเงื่อนไข udiff ไว้ และกำหนดให้ทำซ้ำจนกว่าจะตรงตามเงื่อนไขที่กำหนดไว้ โดยกำหนดให้จนกว่า udiff ≤ 0.001

In [5]:

```
udiff = 1
stepcount = 0

while udiff > .001:
    un = u.copy()
    vn = v.copy()

    b = build_up_b(rho, dt, dx, dy, u, v)
    p = pressure_poisson_periodic(p, dx, dy)

    #หาคำตอบของสมการที่ (4.59)
    u[1:-1, 1:-1] = (un[1:-1, 1:-1] -un[1:-1, 1:-1] * dt / dx *  (un[1:-1, 1:-1] - un[1:-1, 0:-2]) -
                    vn[1:-1, 1:-1] * dt / dy *  (un[1:-1, 1:-1] - un[0:-2, 1:-1]) -dt / (2 * rho *
                    dx) *  (p[1:-1, 2:] - p[1:-1, 0:-2]) +nu * (dt / dx**2 * (un[1:-1, 2:] - 2 *
                     un[1:-1, 1:-1] + un[1:-1, 0:-2]) +dt / dy**2 * (un[2:, 1:-1] - 2 * un[1:-1, 1:
                    -1] + un[0:-2, 1:-1])) + F * dt)

    #หาคำตอบของสมการที่ (4.54)
    v[1:-1, 1:-1] = (vn[1:-1, 1:-1] -un[1:-1, 1:-1] * dt / dx * (vn[1:-1, 1:-1] - vn[1:-1, 0:-2]) –
                    vn[1:-1, 1:-1] * dt / dy *  (vn[1:-1, 1:-1] - vn[0:-2, 1:-1]) -dt / (2 * rho *
                    dy) * (p[2:, 1:-1] - p[0:-2, 1:-1]) + nu * (dt / dx**2 * (vn[1:-1, 2:] - 2 *
                     vn[1:-1, 1:-1] + vn[1:-1, 0:-2]) + dt / dy**2 * (vn[2:, 1:-1] - 2 * vn[1:-1,
                    1:-1] + vn[0:-2, 1:-1])))

    #เงื่อนไขขอบแบบคาบของ u ที่ x = 2
    u[1:-1, -1] = (un[1:-1, -1] - un[1:-1, -1] * dt / dx * (un[1:-1, -1] - un[1:-1, -2]) -vn[1:-1,
                  -1] * dt / dy * (un[1:-1, -1] - un[0:-2, -1]) -dt / (2 * rho * dx) * (p[1:-1, 0]
                  - p[1:-1, -2]) + nu * (dt / dx**2 * (un[1:-1, 0] - 2 * un[1:-1,-1] + un[1:-1,
                  -2]) + dt / dy**2 * (un[2:, -1] - 2 * un[1:-1, -1] + un[0:-2, -1])) + F * dt)

    #เงื่อนไขขอบแบบคาบของ u ที่ x = 0
    u[1:-1, 0] = (un[1:-1, 0] - un[1:-1, 0] * dt / dx * (un[1:-1, 0] - un[1:-1, -1]) - vn[1:-1, 0] *
                 dt / dy *  (un[1:-1, 0] - un[0:-2, 0]) - dt / (2 * rho * dx) * (p[1:-1, 1] - p[1:-1,
               -1]) + nu * (dt / dx**2 * (un[1:-1, 1] - 2 * un[1:-1, 0] + un[1:-1, -1]) + dt /
```

```
                    dy**2 * (un[2:, 0] - 2 * un[1:-1, 0] + un[0:-2, 0])) + F * dt)

    #เงื่อนไขขอบแบบคาบของ v ที่ x = 2
    v[1:-1, -1] = (vn[1:-1, -1] - un[1:-1, -1] * dt / dx * (vn[1:-1, -1] - vn[1:-1, -2]) - vn[1:-1,
                -1] * dt / dy * (vn[1:-1, -1] - vn[0:-2, -1]) -dt / (2 * rho * dy) * (p[2:, -1]
                 - p[0:-2, -1]) +nu * (dt / dx**2 *(vn[1:-1, 0] - 2 * vn[1:-1, -1] + vn[1:-1, -2])
                +dt / dy**2 * (vn[2:, -1] - 2 * vn[1:-1, -1] + vn[0:-2, -1])))

    #เงื่อนไขขอบแบบคาบของ v ที่ x = 0
    v[1:-1, 0] = (vn[1:-1, 0] - un[1:-1, 0] * dt / dx * (vn[1:-1, 0] - vn[1:-1, -1]) - vn[1:-1, 0] *
                dt / dy * (vn[1:-1, 0] - vn[0:-2, 0]) - dt / (2 * rho * dy) *  (p[2:, 0] - p[0:-2,
                0]) +nu * (dt / dx**2 * (vn[1:-1, 1] - 2 * vn[1:-1, 0] + vn[1:-1, -1]) + dt /
                dy**2 * (vn[2:, 0] - 2 * vn[1:-1, 0] + vn[0:-2, 0])))

    #ความเร็วที่ผนังเป็นศูนย์ หรือ u,v = 0 ที่ y = 0,2
    u[0, :] = 0
    u[-1, :] = 0
    v[0, :] = 0
    v[-1, :]=0

    udiff = (numpy.sum(u) - numpy.sum(un)) / numpy.sum(u)
    stepcount += 1
```

ค่าของ stepcount คือ จำนวนรอบที่วนซ้ำจนตรงตามเงื่อนไขที่กำหนดไว้

In [6]:

```
print(stepcount)
```

Out[6]:

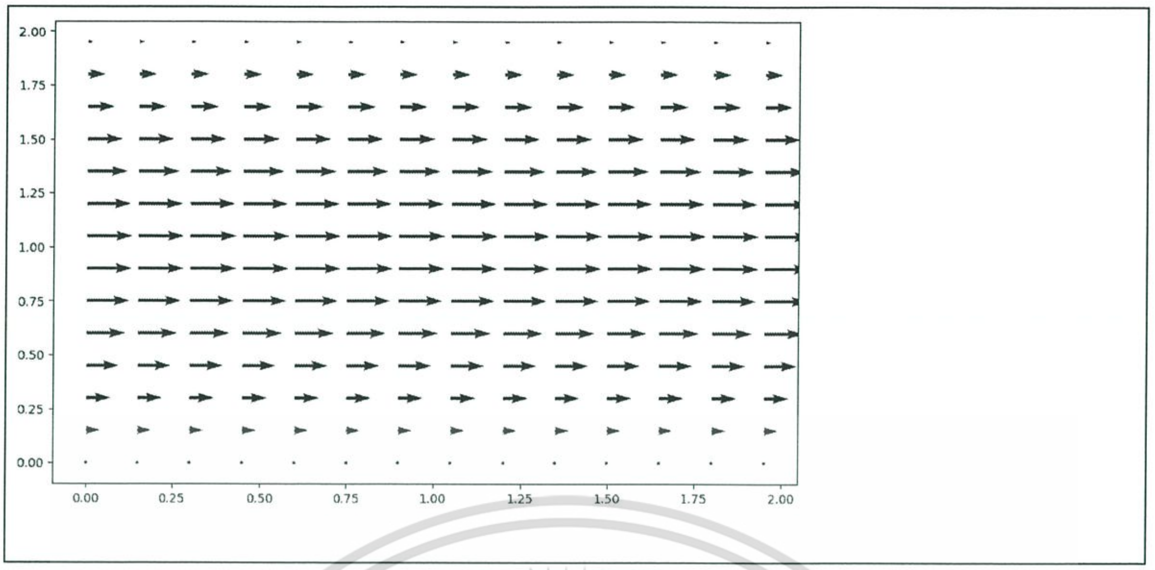
```
499
```

นั่นคือ nt=499 จึงประมาณได้ว่าระบบเสถียร จากนั้นแสดงการไหลโดยสนามความเร็วดังนี้

In[7]:

```
fig = pyplot.figure(figsize = (11,7), dpi=100)
pyplot.quiver(X[::3, ::3], Y[::3, ::3], u[::3, ::3], v[::3, ::3]);
```

Out[7]:


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

บทที่ 5
ขั้นตอนการดำเนินงานวิจัย

ในบทนี้กล่าวถึงขั้นตอนการดำเนินงานวิจัย โดยในงานวิจัยนี้ทำการจำลองการไหลของเลือดโดยการเขียนโค้ดในโปรแกรมคอมพิวเตอร์ ซึ่งใช้ภาษาไพธอนในการแก้สมการควบคุมทั้ง 3 สมการ

5.1 ไพธอน

ภาษาโปรแกรม เป็นภาษาที่ถูกออกแบบขึ้นมาเพื่อใช้สื่อสารกับเครื่องจักร เช่นคอมพิวเตอร์ โดยภาษาโปรแกรมนี้สามารถใช้สร้างโปรแกรมที่ใช้สำหรับควบคุมพฤติกรรมของเครื่องจักรได้ และยังสามารถแสดงออกด้วยขั้นตอนวิธี (algorithm) ได้ ภาษาโปรแกรมในยุคแรกนั้นเกิดขึ้นก่อนที่คอมพิวเตอร์ถูกประดิษฐ์ โดยวัตถุประสงค์ของการเขียนโปรแกรมนั้นใช้เพื่อควบคุมการทำงานของเครื่องทอผ้าและเครื่องเล่นเปียโน หลังจากที่มีผู้คิดค้นคอมพิวเตอร์แล้ว ภาษาโปรแกรมส่วนมากจึงถูกสร้างขึ้นมาเพื่อใช้ในการป้อนชุดคำสั่งให้กับคอมพิวเตอร์ ซึ่งสามารถแบ่งได้ดังนี้

5.1.1 ภาษาระดับต่ำ

ภาษาระดับต่ำสามารถแบ่งได้เป็น ภาษาเครื่อง (Machine Language) และภาษาแอสแซมบลี (Assembly Language) โดยภาษาระดับต่ำนี้เป็นภาษาที่มนุษย์เข้าใจได้ยาก ผู้ที่ใช้ภาษานี้ได้ดีต้องมีความเข้าใจในเรื่องสถาปัตยกรรมคอมพิวเตอร์และฮาร์ดแวร์เป็นอย่างดี ข้อด้อยของภาษาระดับนี้คือไม่เหมาะกับการนำไปพัฒนาต่อ ในส่วนของข้อดีคือสามารถควบคุมฮาร์ดแวร์ได้โดยตรง

5.1.2 ภาษาระดับกลาง

ภาษาระดับกลางมีลักษณะเป็นภาษาแบบโครงสร้าง สามารถทำความเข้าใจได้ง่ายเหมือนภาษาระดับสูง โดยภาษาระดับกลางนำเอาข้อดีของทั้งภาษาระดับต่ำและระดับสูงมาใช้ จึงเป็นที่นิยมอย่างแพร่หลาย เช่น ภาษาซี เป็นต้น

5.1.3 ภาษาระดับสูง

ภาษาระดับสูงใช้คำสั่งเป็นภาษาอังกฤษซึ่งใกล้เคียงกับภาษามนุษย์มาก จึงเป็นภาษาที่เข้าใจได้ง่าย การทำงานของภาษาระดับนี้ทำโดยการสั่งให้คอมพิวเตอร์แปลความหมายของคำสั่ง ซึ่งใช้ตัวแปลภาษาทีละชุดคำสั่ง เรียกว่า อินเทอร์พรีเตอร์ (Interpreter) หรือแปลภาษาครั้งเดียวทั้งโปรแกรม เรียกว่า คอมไพเลอร์ (Compiler)

ไพธอน คือ ภาษาที่ใช้ในการเขียนโปรแกรมเพื่อสื่อสารกับคอมพิวเตอร์ภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม (platform) คือสามารถใช้ภาษาไพธอนได้ทั้งบนระบบ Unix, Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่ง ภาษาตัวนี้เป็นซอฟต์แวร์ที่เปิดเผยทำให้ทุกคนสามารถใช้ภาษาไพธอนเพื่อพัฒนาโปรแกรมของได้โดยไม่ต้องเสียค่าใช้จ่าย ภาษาไพธอนนั้นถูกพัฒนาขึ้นมาโดยทำให้เป็นภาษาที่อ่านง่าย มีโครงสร้างที่ไม่ซับซ้อน และมีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่าภาษาอื่นๆ

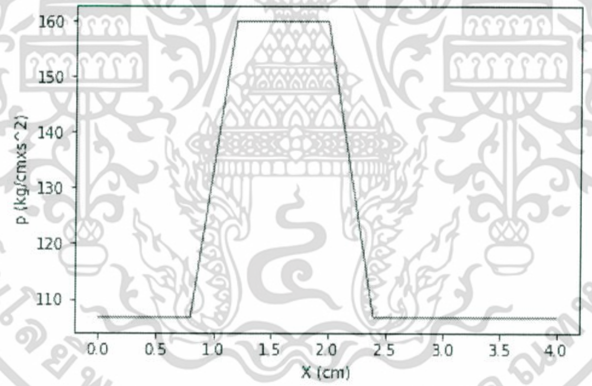
โดยภาษาไพธอนนั้นใช้อินเทอร์พรีเตอร์(interpreter) ในการแปล เพื่อให้สามารถประมวลผลโค้ดไพธอนได้ ซึ่งอินเทอร์พรีเตอร์นั้นทำงานโดยประมวลผลไปทีละบรรทัด ข้อดีคือทราบทันทีว่าบรรทัดไหนมีการทำงานผิดพลาด(error)

5.2 การกำหนดค่าเริ่มต้นและค่าขอบ

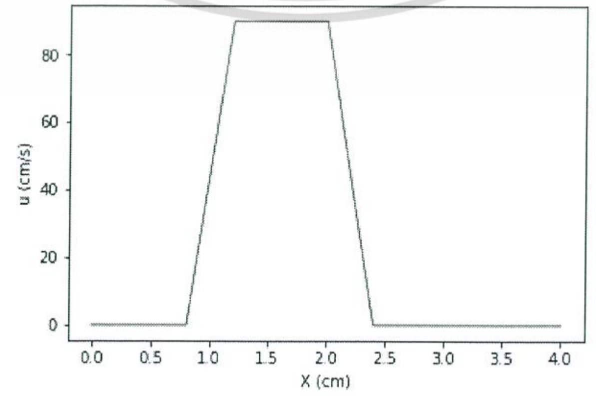
การเขียนโปรแกรมเริ่มต้นจากข้อมูลของหลอดเลือดแดงใหญ่เอออร์ตา (Aorta) โดยนำเอาสมบัติของหลอดเลือดแดงใหญ่เอออร์ต้า (Aorta) ที่เป็นค่าความยาว, ความหนาและรัศมี มาจากงานวิจัย Computational modelling of 1D blood flow with variable mechanical properties and its application to the simulation of wave propagation in the human arterial system โดย S.J. Sherwin, L. Formaggia, J. Peió และ V. Franke และค่าทางกายภาพของหลอดเลือด ซึ่งได้แก่ค่าความหนาแน่นของเลือด, ค่าความหนืดสัมบูรณ์ของเลือดและปริมาตรเลือดที่ออกจากหัวใจนำมาจากค่าคงที่ของงานวิจัยที่ได้กำหนดไว้ในบทที่ 4

ค่าเงื่อนไขที่ขอบนำมาจากค่าของงานวิจัย Hemodynamics in diabetic human aorta using computational fluid dynamics โดย Eunji Shin, Jung Joo Kim, Seonjoong Lee, Kyung Soo ko, Byoung Doo Rhee, Jin Hanและ Nari Kim โดยมีทั้งค่าของความดันและความเร็วที่หลอดเลือดแดงใหญ่เอออร์ต้า (Aorta)

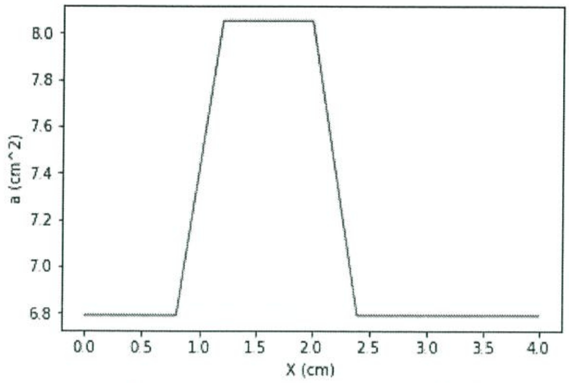
ค่าเงื่อนไขเริ่มต้นกำกนดเป็นฟังก์ชันขั้นบันได ดังนี้



รูปที่ 5.1 ฟังก์ชันขั้นบันไดของค่าความดัน

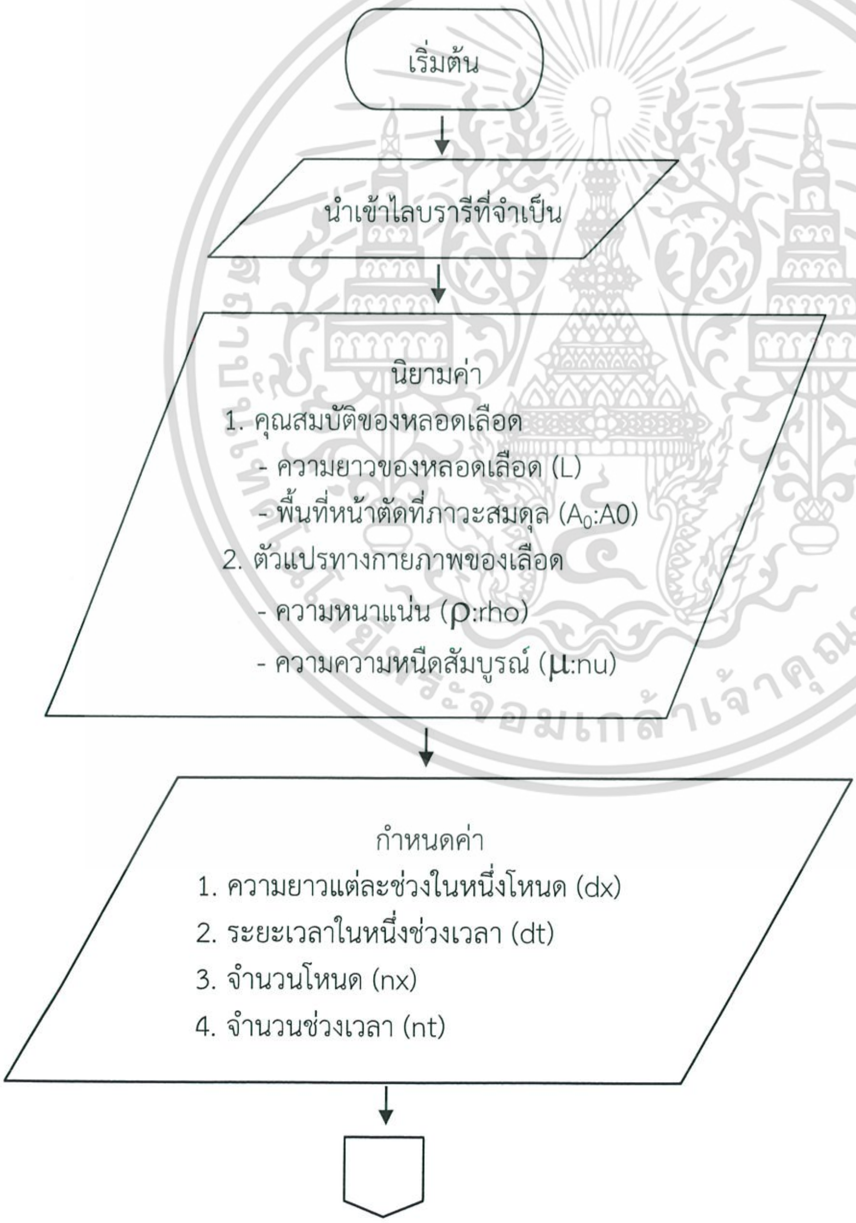


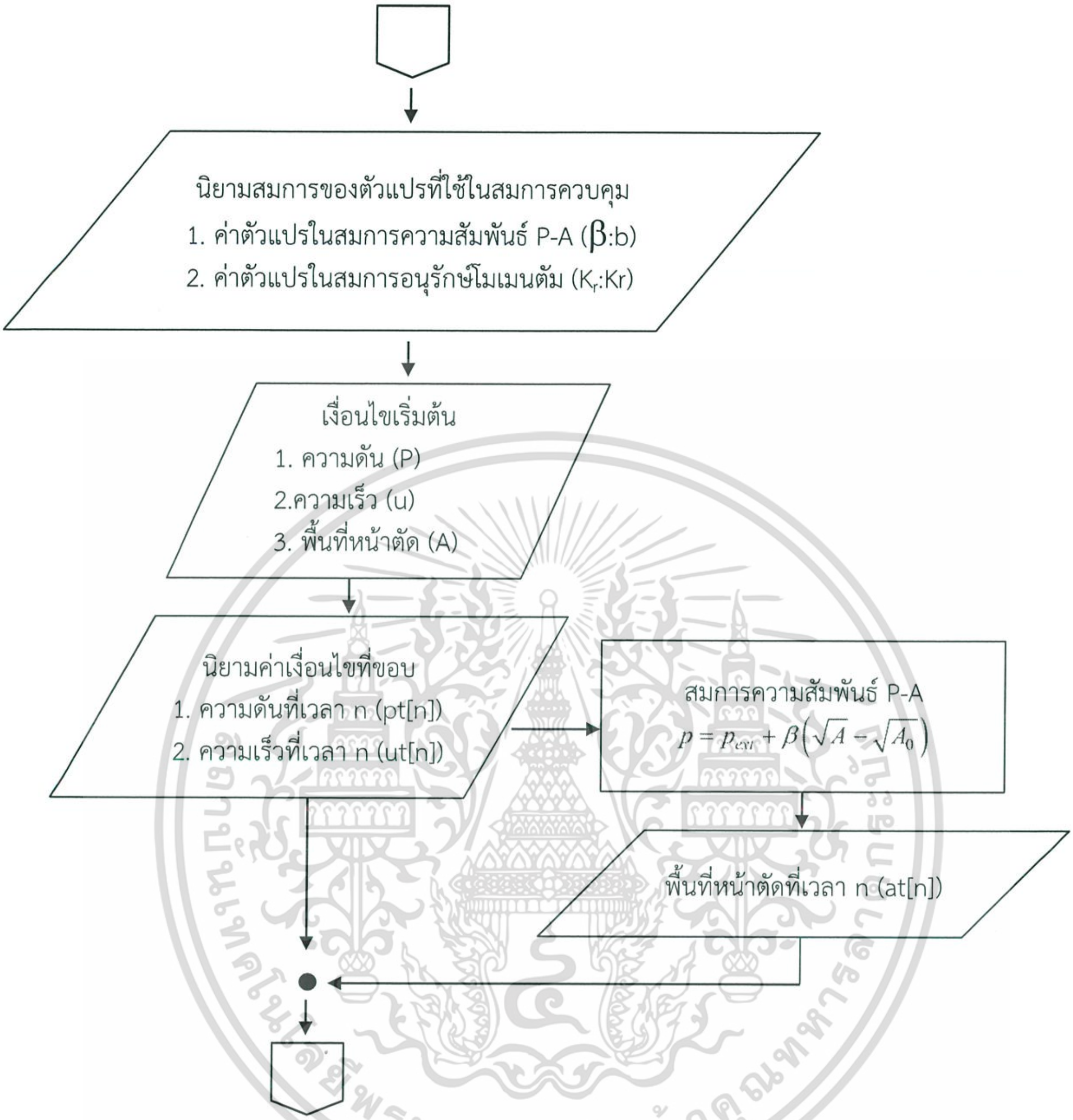
รูปที่ 5.2 ฟังก์ชันขั้นบันไดของค่าความเร็ว

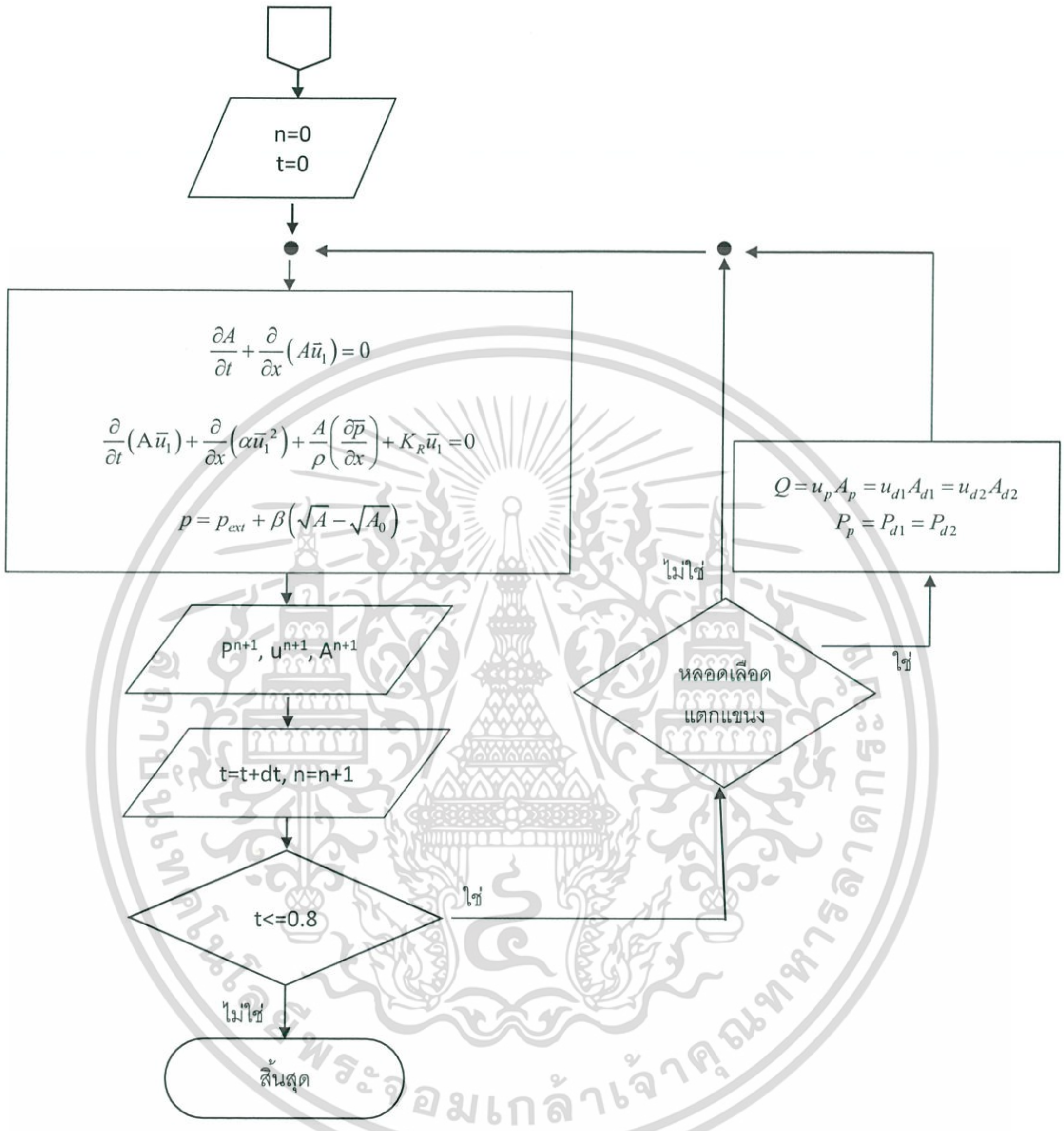


รูปที่ 5.3 ฟังก์ชันขั้นบันไดของพื้นที่

5.3 แผนผังขั้นตอนการทำงานของไพธอน




นิยามสมการของตัวแปรที่ใช้ในสมการควบคุม
1. ค่าตัวแปรในสมการความสัมพันธ์ P-A (β:b)
2. ค่าตัวแปรในสมการอนุรักษ์โมเมนตัม (K_r:Kr)
เงื่อนไขเริ่มต้น
1. ความดัน (P)
2.ความเร็ว (u)
3. พื้นที่หน้าตัด (A)
นิยามค่าเงื่อนไขที่ขอบ
1. ความดันที่เวลา n (pt[n])
2. ความเร็วที่เวลา n (ut[n])
สมการความสัมพันธ์ P-A
$p = p_{ext} + \beta\left(\sqrt{A} - \sqrt{A_0}\right)$
พื้นที่หน้าตัดที่เวลา n (at[n])



5.4 โค้ดที่ใช้ในไพธอน

โค้ดของหลอดเลือดแดงใหญ่เอออร์ต้า (Aorta) และคุณสมบัติของหลอดเลือดแดงหลัก 55 เส้น

```
import numpy
import math
from matplotlib import pyplot, cm
```

```
def build_up_kr(an):
    kr = numpy.zeros_like(an)
    kr[:] = (8*nu*L[m])/(an[:]*(an[:]/math.pi))
    return kr



def pressure_area(BCpressure, a):
    p[0]=BCpressure[n]
    p[1:] =po+b[m]*((a[1:]**0.5)-(ao[m]**0.5))
    return p



def mass_conserv(BCarea, an, un):
    a[0]=BCarea[n]
    a[1:-1] = an[1:-1]-(dt/dx[m])*(an[1:-1]*un[1:-1]-an[0:-2]*un[0:-2])
    a[-1] = an[-1]-(dt/dx[m])*(an[-1]*un[-1]-an[-2]*un[-2])
    return a



def mom_conserv(BCu, an, un, pn, a):
    u[0]=BCu[n]
    u[1:-1] = (an[1:-1]*un[1:-1]-dt*((un[1:-1]**2-un[0:-2]**2)/dx[m]+(an[1:-
1]/(rho*dx[m]))*(pn[1:-1]-pn[0:-2])+kr[1:-1]*un[1:-1]))/a[1:-1]
    u[-1] = (an[-1]*un[-1]-dt*((un[-1]**2-un[-2]**2)/dx[m]+(an[-1]/(rho*dx[m]))*(pn[-1]-pn[-
2])+kr[-1]*un[-1]))/a[-1]
    return u

def ini_p(p, m):
    p = numpy.zeros(int(nx[m]), dtype=numpy.longdouble)
    p[:]=po
    p[int(1.5/dx[m]):int(2/dx[m]+1)]=160
    pn=p.copy()
    return p

def ini_a(a, p, m):
    a = numpy.zeros(int(nx[m]), dtype=numpy.longdouble)
    a[:]=((p[:]-po)/b[m]+(ao[m])**0.5)**2
    an=a.copy()
```

```
    return a

def ini_u(u, m):
    u = numpy.zeros(int(nx[m]), dtype=numpy.longdouble)
    u[:]=0
    u[int(1.5/dx[m]):int(2/dx[m]+1)]=90
    un=u.copy()
    return u


##สมบัติหลอดเลือด
#ความยาวหลอดเลือด55เส้น
L=numpy.zeros(56)
L[1]=4
L[2]=2
L[3]=3.4
L[4]=3.4
L[5]=17.7
L[6]=14.8
L[7]=42.2
L[8]=23.5
L[9]=6.7
L[10]=7.9
L[11]=17.1
L[12]=17.6
L[13]=17.7
L[14]=3.9
L[15]=20.8
L[16]=17.6
L[17]=17.7
L[18]=5.2
L[19]=3.4
L[20]=14.8
L[21]=42.2
L[22]=23.5
L[23]=6.7
L[24]=7.9
L[25]=17.1
```

```
L[26]=8
L[27]=10.4
L[28]=5.3
L[29]=2
L[30]=1
L[31]=6.6
L[32]=7.1
L[33]=6.3
L[34]=5.9
L[35]=1
L[36]=3.2
L[37]=1
L[38]=3.2
L[39]=10.6
L[40]=5
L[41]=1
L[42]=5.9
L[43]=5.8
L[44]=14.4
L[45]=5
L[46]=44.3
L[47]=12.6
L[48]=32.1
L[49]=34.3
L[50]=14.5
L[51]=5
L[52]=44.4
L[53]=12.7
L[54]=32.2
L[55]=34.4

#bของหลอดเลือด55เส้น
b=numpy.zeros(56)
b[1]=230
b[2]=240
b[3]=490
b[4]=690
b[5]=850
```

```
b[6]=4700
b[7]=760
b[8]=1920
b[9]=1340
b[10]=8950
b[11]=1480
b[12]=1860
b[13]=1730
b[14]=240
b[15]=1110
b[16]=2430
b[17]=2270
b[18]=260
b[19]=880
b[20]=6570
b[21]=970
b[22]=2470
b[23]=1720
b[24]=11390
b[25]=1890
b[26]=1470
b[27]=260
b[28]=320
b[29]=560
b[30]=4810
b[31]=700
b[32]=960
b[33]=1090
b[34]=830
b[35]=340
b[36]=1300
b[37]=380
b[38]=1300
b[39]=510
b[40]=3440
b[41]=490
b[42]=820
b[43]=820
```

```
b[44]=1370
b[45]=5310
b[46]=2310
b[47]=2230
b[48]=3830
b[49]=11970
b[50]=1370
b[51]=5310
b[52]=2310
b[53]=2230
b[54]=3850
b[55]=12100

#aoของหลอดเลือด55เส้น
ao=numpy.zeros(56)
ao[1]=6.789
ao[2]=5.011
ao[3]=1.535
ao[4]=0.919
ao[5]=0.703
ao[6]=0.181
ao[7]=.833
ao[8]=.423
ao[9]=.648
ao[10]=.118
ao[11]=.589
ao[12]=.458
ao[13]=.458
ao[14]=4.486
ao[15]=.536
ao[16]=.35
ao[17]=.35
ao[18]=3.941
ao[19]=.706
ao[20]=.129
ao[21]=.65
ao[22]=.33
ao[23]=.505
```

```
ao[24]=.093
ao[25]=.461
ao[26]=.316
ao[27]=.3604
ao[28]=2.659
ao[29]=1.086
ao[30]=.126
ao[31]=.659
ao[32]=.442
ao[33]=.468
ao[34]=.782
ao[35]=2.233
ao[36]=.385
ao[37]=1.981
ao[38]=.385
ao[39]=1.389
ao[40]=.118
ao[41]=1.251
ao[42]=.694
ao[43]=.694
ao[44]=.73
ao[45]=.285
ao[46]=.409
ao[47]=.398
ao[48]=.444
ao[49]=.123
ao[50]=.73
ao[51]=.285
ao[52]=.409
ao[53]=.398
ao[54]=.442
ao[55]=.522

##physical variables of blood
rho = 1060*(10**-6)
nu = 2.78*(10**-5)


##variable declarations
```

```
nx=numpy.array([ 0, 11, 7, 9, 9, 45, 39, 107, 59, 17, 21, 43, 45, 45, 11, 53, 45, 45, 15, 9, 39,
107, 59, 17, 21, 43, 21, 27, 15, 7, 3, 17, 19, 17, 15, 3, 9, 3, 9, 27, 13, 3, 15, 15, 37, 13, 111,
33, 81, 87, 37, 13, 113,  33,  81,  87])
dx=numpy.zeros(56)
dx[1:] =  L[1:]/(nx[1:]-1)
nt = 10001
dt = 0.8/(nt-1) #หัวใจเต้น 0.8 วินาที/ครั้ง

#po=numpy.zeros(nt)
#po=rho*ut[]
po=106.66
pmax=109.312
#qo=83.333

#initial conditions
p1 = numpy.zeros(int(nx[1]), dtype=numpy.longdouble)
p1[:]=po
p1[int(1.5/dx[1]):int(2/dx[1]+1)]=160
#p = numpy.linspace(pt[0], po, nx)
pn1=p1.copy()


a1 = numpy.zeros(int(nx[1]), dtype=numpy.longdouble)
a1[:]=((p1[:]-po)/b[1]+(ao[1])**0.5)**2
#a[:]=ao
an1=a1.copy()

u1 = numpy.zeros(int(nx[1]), dtype=numpy.longdouble)
u1[:]=0
u1[int(1.5/dx[1]):int(2/dx[1]+1)]=90
#u = numpy.linspace(56, 90, nx)
#u[:]=ut[0]*a[0]/a[:]
un1 = u1.copy()


p1end=numpy.zeros(nt)
u1end=numpy.zeros(nt)
p1mid=numpy.zeros(nt)
```

```
#boundary condition

pt = numpy.zeros((nt), dtype=numpy.longdouble)

i=0
while (i <=int(.035/dt)):
    pt[i]=(82-1.5/.035*dt*i)*1.33325
    i += 1
while (i <=int(.06/dt)):
    pt[i]=(80.5+(1.5)/(.06-.035)*dt*(i-.035/dt))*1.33325
    i += 1
while (i <=int(.08/dt)):
    pt[i]=(82+(86-82)/(.08-.06)*dt*(i-.06/dt))*1.33325
    i += 1
while (i <=int(.2/dt)):
    pt[i]=(86+(113-86)/(.2-.08)*dt*(i-.08/dt))*1.33325
    i += 1
while (i <=int(.235/dt)):
    pt[i]=(113+(117.5-113)/(.235-.2)*dt*(i-.2/dt))*1.33325
    i += 1
while (i <=int(.25/dt)):
    pt[i]=(117.5+(119-117.5)/(.25-.235)*dt*(i-.235/dt))*1.33325
    i += 1
while (i <=int(.265/dt)):
    pt[i]=(119+(119.5-119)/(.265-.25)*dt*(i-.25/dt))*1.33325
    i += 1
while (i <=int(.28/dt)):
    pt[i]=(119.5+(118.5-119.5)/(.28-.265)*dt*(i-.265/dt))*1.33325
    i += 1
while (i <=int(.305/dt)):
    pt[i]=(118.5+(117.5-118.5)/(.305-.28)*dt*(i-.28/dt))*1.33325
    i += 1
while (i <=int(.33/dt)):
    pt[i]=(117.5+(115.5-117.5)/(.33-.305)*dt*(i-.305/dt))*1.33325
    i += 1
while (i <=int(.35/dt)):
    pt[i]=(115.5+(112.5-115.5)/(.35-.33)*dt*(i-.33/dt))*1.33325
```

```
    i += 1
while (i <=int(.38/dt)):
    pt[i]=(112.5+(101-112.5)/(.38-.35)*dt*(i-.35/dt))*1.33325
    i += 1
while (i <=int(.4/dt)):
    pt[i]=(101+(98-101)/(.4-.38)*dt*(i-.38/dt))*1.33325
    i += 1
while (i <=int(.415/dt)):
    pt[i]=(98+(98-98)/(.415-.4)*dt*(i-.4/dt))*1.33325
    i += 1
while (i <=int(.43/dt)):
    pt[i]=(98+(100-98)/(.43-.415)*dt*(i-.415/dt))*1.33325
    i += 1
while (i <=int(.45/dt)):
    pt[i]=(100+(105-100)/(.45-.43)*dt*(i-.43/dt))*1.33325
    i += 1
while (i <=int(.46/dt)):
    pt[i]=(105+(106-105)/(.46-.45)*dt*(i-.45/dt))*1.33325
    i += 1
while (i <=int(.47/dt)):
    pt[i]=(106+(105-106)/(.47-.46)*dt*(i-.46/dt))*1.33325
    i += 1
while (i <=int(.495/dt)):
    pt[i]=(105+(102-105)/(.495-.47)*dt*(i-.47/dt))*1.33325
    i += 1
while (i <=int(.52/dt)):
    pt[i]=(102+(97-102)/(.52-.495)*dt*(i-.495/dt))*1.33325
    i += 1
while (i <=int(.55/dt)):
    pt[i]=(97+(95-97)/(.55-.52)*dt*(i-.52/dt))*1.33325
    i += 1
while (i <=int(.62/dt)):
    pt[i]=(95+(93-95)/(.62-.55)*dt*(i-.55/dt))*1.33325
    i += 1
while (i <=int(.67/dt)):
    pt[i]=(93+(89-93)/(.67-.62)*dt*(i-.62/dt))*1.33325
    i += 1
while (i <=int(.74/dt)):
```

```
    pt[i]=(89+(85-89)/(.74-.67)*dt*(i-.67/dt))*1.33325
    i += 1
while (i <=int(.8/dt)):
    pt[i]=(85+(82-85)/(.8-.74)*dt*(i-.74/dt))*1.33325
    i += 1

at=numpy.zeros(nt)
at[:]=((pt[:]-po)/b[1]+(ao[1])**0.5)**2

ut=numpy.zeros(nt)
i=0
while (i<=int(.03/dt)):
    ut[i]=(.56+(.53-.56)/(.03-.00)*dt*(i))*100
    i+=1
while (i<=int(.04/dt)):
    ut[i]=(.53+(.52-.53)/(.04-.03)*dt*(i-(.03/dt)))*100
    i += 1
while (i<=int(.06/dt)):
    ut[i]=(.52+(.52-.52)/(.06-.04)*dt*(i-(.04/dt)))*100
    i+=1
while (i<=int(.07/dt)):
    ut[i]=(.52+(.525-.52)/(.07-.06)*dt*(i-(.06/dt)))*100
    i+=1
while (i<=int(.08/dt)):
    ut[i]=(.525+(.54-.525)/(.08-.07)*dt*(i-(.07/dt)))*100
    i+=1
while (i<=int(.10/dt)):
    ut[i]=(.54+(.565-.54)/(.10-.08)*dt*(i-(.08/dt)))*100
    i+=1
while (i<=int(.11/dt)):
    ut[i]=(.565+(.59-.565)/(.11-.10)*dt*(i-(.10/dt)))*100
    i+=1
while (i<=int(.175/dt)):
    ut[i]=(.59+(.81-.59)/(.175-.11)*dt*(i-(.11/dt)))*100
    i+=1
while (i<=int(.19/dt)):
    ut[i]=(.81+(.85-.81)/(.19-.175)*dt*(i-(.175/dt)))*100
    i+=1
```

```
while (i<=int(.21/dt)):
    ut[i]=(.85+(.865-.85)/(.21-.19)*dt*(i-(.19/dt)))*100
    i+=1
while (i<=int(.23/dt)):
    ut[i]=(.865+(.875-.865)/(.23-.21)*dt*(i-(.21/dt)))*100
    i+=1
while (i<=int(.265/dt)):
    ut[i]=(.875+(.87-.875)/(.265-.23)*dt*(i-(.23/dt)))*100
    i+=1
while (i<=int(.29/dt)):
    ut[i]=(.87+(.86-.87)/(.29-.265)*dt*(i-(.265/dt)))*100
    i+=1
while (i<=int(.31/dt)):
    ut[i]=(.86+(.84-.86)/(.31-.29)*dt*(i-(.29/dt)))*100
    i+=1
while (i<=int(.33/dt)):
    ut[i]=(.84+(.80-.84)/(.33-.31)*dt*(i-(.31/dt)))*100
    i+=1
while (i<=int(.36/dt)):
    ut[i]=(.80+(.74-.80)/(.36-.33)*dt*(i-(.33/dt)))*100
    i+=1
while (i<=int(.39/dt)):
    ut[i]=(.74+(.65-.74)/(.39-.36)*dt*(i-(.36/dt)))*100
    i+=1
while (i<=int(.41/dt)):
    ut[i]=(.65+(.55-.65)/(.41-.39)*dt*(i-(.39/dt)))*100
    i+=1
while (i<=int(.42/dt)):
    ut[i]=(.55+(.525-.55)/(.42-.41)*dt*(i-(.41/dt)))*100
    i+=1
while (i<=int(.44/dt)):
    ut[i]=(.525+(.51-.525)/(.44-.42)*dt*(i-(.42/dt)))*100
    i+=1
while (i<=int(.45/dt)):
    ut[i]=(.51+(.50-.51)/(.45-.44)*dt*(i-(.44/dt)))*100
    i+=1
while (i<=int(.47/dt)):
    ut[i]=(.50+(.51-.50)/(.47-.45)*dt*(i-(.45/dt)))*100
```

```
    i+=1
while (i<=int(.51/dt)):
    ut[i]=(.51+(.53-.51)/(.51-.47)*dt*(i-(.47/dt)))*100
    i+=1
while (i<=int(.54/dt)):
    ut[i]=(.53+(.54-.53)/(.54-.51)*dt*(i-(.51/dt)))*100
    i+=1
while (i<=int(.59/dt)):
    ut[i]=(.54+(.565-.54)/(.59-.54)*dt*(i-(.54/dt)))*100
    i+=1
while (i<=int(.65/dt)):
    ut[i]=(.565+(.58-.565)/(.65-.59)*dt*(i-(.59/dt)))*100
    i+=1
while (i<=int(.74/dt)):
    ut[i]=(.58+(.585-.58)/(.74-.65)*dt*(i-(.65/dt)))*100
    i+=1
while (i<=int(.78/dt)):
    ut[i]=(.585+(.58-.585)/(.78-.74)*dt*(i-(.74/dt)))*100
    i+=1
while (i<=int(.79/dt)):
    ut[i]=(.58+(.57-.58)/(.79-.78)*dt*(i-(.78/dt)))*100
    i+=1
while (i<=int(.80/dt)):
    ut[i]=(.57+(.56-.57)/(.80-.79)*dt*(i-(.79/dt)))*100
    i+=1

t = 0
n = 0

while t <= 0.8:

    #หลอดเลือดที่1
    m=1
    an1 = a1.copy()
    pn1 = p1.copy()
    un1 = u1.copy()

    ##mass_eq
```

```
    a1=mass_conserv(at, an1, un1)

    ##pressure_area
    p1=pressure_area(pt, a1)

    ##mom_eq
    kr = build_up_kr(an1)
    u1=mom_conserv(ut, an1, un1, pn1, a1)

    p1mid[n]=p1[5]
    p1end[n]=p1[10]
    a1end=numpy.zeros(nt)
    a1end[:]=((p1end[:]-po)/b[m]+(ao[m])**0.5)**2

    t+=dt
    n+=1

pyplot.figure(figsize=(11, 7), dpi=100)
pyplot.plot(numpy.linspace(0, 0.008, n-1),p1mid[:n-1], marker='o', lw=1, label='Mid of
vessel')
pyplot.plot(numpy.linspace(0, 0.008, n-1),p1end[:n-1], marker='^', lw=1, label='end of
vessel')
pyplot.xlabel('t (s)')
pyplot.ylabel('p (kg/cmxs^2)')
pyplot.legend();
```

โค้ดของหลอดเลือดแดงหลัก 55 เส้น

```
mport numpy
import math
from matplotlib import pyplot, cm


def build_up_kr(an):
    kr = numpy.zeros_like(an)
    kr[:] = (8*nu*L[m])/(an[:]*(an[:]/math.pi))
    return kr
```

```
def pressure_area(BCpressure, a):
    p[0]=BCpressure[n]
    p[1:] =po+b[m]*((a[1:]**0.5)-(ao[m]**0.5))
    return p


def mass_conserv(BCarea, an, un):
    a[0]=BCarea[n]
    a[1:-1] = an[1:-1]-(dt/dx[m])*(an[1:-1]*un[1:-1]-an[0:-2]*un[0:-2])
    a[-1] = an[-1]-(dt/dx[m])*(an[-1]*un[-1]-an[-2]*un[-2])
    return a


def mom_conserv(BCu, an, un, pn, a):
    u[0]=BCu[n]
    u[1:-1] = (an[1:-1]*un[1:-1]-dt*((un[1:-1]**2-un[0:-2]**2)/dx[m]+(an[1:-
1]/(rho*dx[m]))*(pn[1:-1]-pn[0:-2])+kr[1:-1]*un[1:-1]))/a[1:-1]
    u[-1] = (an[-1]*un[-1]-dt*((un[-1]**2-un[-2]**2)/dx[m]+(an[-1]/(rho*dx[m]))*(pn[-1]-pn[-
2])+kr[-1]*un[-1]))/a[-1]
    return u

def ini_p(p, m):
    p = numpy.zeros(int(nx[m]), dtype=numpy.longdouble)
    p[:]=po
    p[int(1.5/dx[m]):int(2/dx[m]+1)]=160
    pn=p.copy()
    return p

def ini_a(a, p, m):
    a = numpy.zeros(int(nx[m]), dtype=numpy.longdouble)
    a[:]=((p[:]-po)/b[m]+(ao[m])**0.5)**2
    an=a.copy()
    return a

def ini_u(u, m):
    u = numpy.zeros(int(nx[m]), dtype=numpy.longdouble)
    u[:]=0
    u[int(1.5/dx[m]):int(2/dx[m]+1)]=90
```

```
    un=u.copy()
    return u

##สมบัติหลอดเลือด
#ความยาวหลอดเลือด55เส้น
L=numpy.zeros(56)
L[1]=4
L[2]=2
L[3]=3.4
L[4]=3.4
L[5]=17.7
L[6]=14.8
L[7]=42.2
L[8]=23.5
L[9]=6.7
L[10]=7.9
L[11]=17.1
L[12]=17.6
L[13]=17.7
L[14]=3.9
L[15]=20.8
L[16]=17.6
L[17]=17.7
L[18]=5.2
L[19]=3.4
L[20]=14.8
L[21]=42.2
L[22]=23.5
L[23]=6.7
L[24]=7.9
L[25]=17.1
L[26]=8
L[27]=10.4
L[28]=5.3
L[29]=2
L[30]=1
L[31]=6.6
L[32]=7.1
```

```
L[33]=6.3
L[34]=5.9
L[35]=1
L[36]=3.2
L[37]=1
L[38]=3.2
L[39]=10.6
L[40]=5
L[41]=1
L[42]=5.9
L[43]=5.8
L[44]=14.4
L[45]=5
L[46]=44.3
L[47]=12.6
L[48]=32.1
L[49]=34.3
L[50]=14.5
L[51]=5
L[52]=44.4
L[53]=12.7
L[54]=32.2
L[55]=34.4

#bของหลอดเลือด55เส้น
b=numpy.zeros(56)
b[1]=230
b[2]=240
b[3]=490
b[4]=690
b[5]=850
b[6]=4700
b[7]=760
b[8]=1920
b[9]=1340
b[10]=8950
b[11]=1480
b[12]=1860
```

```
b[13]=1730
b[14]=240
b[15]=1110
b[16]=2430
b[17]=2270
b[18]=260
b[19]=880
b[20]=6570
b[21]=970
b[22]=2470
b[23]=1720
b[24]=11390
b[25]=1890
b[26]=1470
b[27]=260
b[28]=320
b[29]=560
b[30]=4810
b[31]=700
b[32]=960
b[33]=1090
b[34]=830
b[35]=340
b[36]=1300
b[37]=380
b[38]=1300
b[39]=510
b[40]=3440
b[41]=490
b[42]=820
b[43]=820
b[44]=1370
b[45]=5310
b[46]=2310
b[47]=2230
b[48]=3830
b[49]=11970
b[50]=1370
```

```
b[51]=5310
b[52]=2310
b[53]=2230
b[54]=3850
b[55]=12100

#aoของหลอดเลือด55เส้น
ao=numpy.zeros(56)
ao[1]=6.789
ao[2]=5.011
ao[3]=1.535
ao[4]=0.919
ao[5]=0.703
ao[6]=0.181
ao[7]=.833
ao[8]=.423
ao[9]=.648
ao[10]=.118
ao[11]=.589
ao[12]=.458
ao[13]=.458
ao[14]=4.486
ao[15]=.536
ao[16]=.35
ao[17]=.35
ao[18]=3.941
ao[19]=.706
ao[20]=.129
ao[21]=.65
ao[22]=.33
ao[23]=.505
ao[24]=.093
ao[25]=.461
ao[26]=.316
ao[27]=.3604
ao[28]=2.659
ao[29]=1.086
ao[30]=.126
```

```
ao[31]=.659
ao[32]=.442
ao[33]=.468
ao[34]=.782
ao[35]=2.233
ao[36]=.385
ao[37]=1.981
ao[38]=.385
ao[39]=1.389
ao[40]=.118
ao[41]=1.251
ao[42]=.694
ao[43]=.694
ao[44]=.73
ao[45]=.285
ao[46]=.409
ao[47]=.398
ao[48]=.444
ao[49]=.123
ao[50]=.73
ao[51]=.285
ao[52]=.409
ao[53]=.398
ao[54]=.442
ao[55]=.522

##physical variables of blood
rho = 1060*(10**-6)
nu = 2.78*(10**-5)

##variable declarations
nx=numpy.array([ 0, 11, 7, 9, 9, 45, 39, 107, 59, 17, 21, 43, 45, 45, 11, 53, 45, 45, 15, 9, 39, 107, 59, 17, 21, 43, 21, 27, 15, 7, 3, 17, 19, 17, 15, 3, 9, 3, 9, 27, 13, 3, 15, 15, 37, 13, 111, 33, 81, 87, 37, 13, 113,  33,  81,  87])
dx=numpy.zeros(56)
dx[1:] =  L[1:]/(nx[1:]-1)
nt = 10001
dt = 0.8/(nt-1) #หัวใจเต้น 0.8 วินาที/ครั้ง
```

```
#po=numpy.zeros(nt)
#po=rho*ut[]
po=106.66
pmax=109.312
#qo=83.333

#initial conditions
p1 = numpy.zeros(int(nx[1]), dtype=numpy.longdouble)
p1[:]=po
p1[int(1.5/dx[1]):int(2/dx[1]+1)]=160
#p = numpy.linspace(pt[0], po, nx)
pn1=p1.copy()


a1 = numpy.zeros(int(nx[1]), dtype=numpy.longdouble)
a1[:]=((p1[:]-po)/b[1]+(ao[1])**0.5)**2
#a[:]=ao
an1=a1.copy()

u1 = numpy.zeros(int(nx[1]), dtype=numpy.longdouble)
u1[:]=0
u1[int(1.5/dx[1]):int(2/dx[1]+1)]=90
#u = numpy.linspace(56, 90, nx)
#u[:]=ut[0]*a[0]/a[:]
un1 = u1.copy()


p1end=numpy.zeros(nt)
u1end=numpy.zeros(nt)
p1mid=numpy.zeros(nt)

#IC เส้น2
p2=numpy.zeros(nt)
a2=numpy.zeros(nt)
u2=numpy.zeros(nt)
p2=ini_p(p2, 2)
a2=ini_a(a2, p2, 2)
```

```
u2=ini_u(u2, 2)
an2 = a2.copy()
pn2 = p2.copy()
un2 = u2.copy()

#IC เส้น3
p3=numpy.zeros(nt)
a3=numpy.zeros(nt)
u3=numpy.zeros(nt)
p3=ini_p(p3, 3)
a3=ini_a(a3, p3, 3)
u3=ini_u(u3, 3)
an3 = a3.copy()
pn3 = p3.copy()
un3 = u3.copy()
.
.
.
.
#boundary condition

pt = numpy.zeros((nt), dtype=numpy.longdouble)

i=0
while (i <=int(.035/dt)):
    pt[i]=(82-1.5/.035*dt*i)*1.33325
    i += 1
while (i <=int(.06/dt)):
    pt[i]=(80.5+(1.5)/(.06-.035)*dt*(i-.035/dt))*1.33325
    i += 1
while (i <=int(.08/dt)):
    pt[i]=(82+(86-82)/(.08-.06)*dt*(i-.06/dt))*1.33325
    i += 1
while (i <=int(.2/dt)):
    pt[i]=(86+(113-86)/(.2-.08)*dt*(i-.08/dt))*1.33325
    i += 1
while (i <=int(.235/dt)):
    pt[i]=(113+(117.5-113)/(.235-.2)*dt*(i-.2/dt))*1.33325
```

```
    i += 1
while (i <=int(.25/dt)):
    pt[i]=(117.5+(119-117.5)/(.25-.235)*dt*(i-.235/dt))*1.33325
    i += 1
while (i <=int(.265/dt)):
    pt[i]=(119+(119.5-119)/(.265-.25)*dt*(i-.25/dt))*1.33325
    i += 1
while (i <=int(.28/dt)):
    pt[i]=(119.5+(118.5-119.5)/(.28-.265)*dt*(i-.265/dt))*1.33325
    i += 1
while (i <=int(.305/dt)):
    pt[i]=(118.5+(117.5-118.5)/(.305-.28)*dt*(i-.28/dt))*1.33325
    i += 1
while (i <=int(.33/dt)):
    pt[i]=(117.5+(115.5-117.5)/(.33-.305)*dt*(i-.305/dt))*1.33325
    i += 1
while (i <=int(.35/dt)):
    pt[i]=(115.5+(112.5-115.5)/(.35-.33)*dt*(i-.33/dt))*1.33325
    i += 1
while (i <=int(.38/dt)):
    pt[i]=(112.5+(101-112.5)/(.38-.35)*dt*(i-.35/dt))*1.33325
    i += 1
while (i <=int(.4/dt)):
    pt[i]=(101+(98-101)/(.4-.38)*dt*(i-.38/dt))*1.33325
    i += 1
while (i <=int(.415/dt)):
    pt[i]=(98+(98-98)/(.415-.4)*dt*(i-.4/dt))*1.33325
    i += 1
while (i <=int(.43/dt)):
    pt[i]=(98+(100-98)/(.43-.415)*dt*(i-.415/dt))*1.33325
    i += 1
while (i <=int(.45/dt)):
    pt[i]=(100+(105-100)/(.45-.43)*dt*(i-.43/dt))*1.33325
    i += 1
while (i <=int(.46/dt)):
    pt[i]=(105+(106-105)/(.46-.45)*dt*(i-.45/dt))*1.33325
    i += 1
while (i <=int(.47/dt)):
```

```
    pt[i]=(106+(105-106)/(.47-.46)*dt*(i-.46/dt))*1.33325
    i += 1
while (i <=int(.495/dt)):
    pt[i]=(105+(102-105)/(.495-.47)*dt*(i-.47/dt))*1.33325
    i += 1
while (i <=int(.52/dt)):
    pt[i]=(102+(97-102)/(.52-.495)*dt*(i-.495/dt))*1.33325
    i += 1
while (i <=int(.55/dt)):
    pt[i]=(97+(95-97)/(.55-.52)*dt*(i-.52/dt))*1.33325
    i += 1
while (i <=int(.62/dt)):
    pt[i]=(95+(93-95)/(.62-.55)*dt*(i-.55/dt))*1.33325
    i += 1
while (i <=int(.67/dt)):
    pt[i]=(93+(89-93)/(.67-.62)*dt*(i-.62/dt))*1.33325
    i += 1
while (i <=int(.74/dt)):
    pt[i]=(89+(85-89)/(.74-.67)*dt*(i-.67/dt))*1.33325
    i += 1
while (i <=int(.8/dt)):
    pt[i]=(85+(82-85)/(.8-.74)*dt*(i-.74/dt))*1.33325
    i += 1

at=numpy.zeros(nt)
at[:]=((pt[:]-po)/b[1]+(ao[1])**0.5)**2
ut=numpy.zeros(nt)
i=0
while (i<=int(.03/dt)):
    ut[i]=(.56+(.53-.56)/(.03-.00)*dt*(i))*100
    i+=1
while (i<=int(.04/dt)):
    ut[i]=(.53+(.52-.53)/(.04-.03)*dt*(i-(.03/dt)))*100
    i += 1
while (i<=int(.06/dt)):
    ut[i]=(.52+(.52-.52)/(.06-.04)*dt*(i-(.04/dt)))*100
    i+=1
while (i<=int(.07/dt)):
```

```
    ut[i]=(.52+(.525-.52)/(.07-.06)*dt*(i-(.06/dt)))*100
    i+=1
 while (i<=int(.08/dt)):
    ut[i]=(.525+(.54-.525)/(.08-.07)*dt*(i-(.07/dt)))*100
    i+=1
 while (i<=int(.10/dt)):
    ut[i]=(.54+(.565-.54)/(.10-.08)*dt*(i-(.08/dt)))*100
    i+=1
 while (i<=int(.11/dt)):
    ut[i]=(.565+(.59-.565)/(.11-.10)*dt*(i-(.10/dt)))*100
    i+=1
 while (i<=int(.175/dt)):
    ut[i]=(.59+(.81-.59)/(.175-.11)*dt*(i-(.11/dt)))*100
    i+=1
 while (i<=int(.19/dt)):
    ut[i]=(.81+(.85-.81)/(.19-.175)*dt*(i-(.175/dt)))*100
    i+=1
 while (i<=int(.21/dt)):
    ut[i]=(.85+(.865-.85)/(.21-.19)*dt*(i-(.19/dt)))*100
    i+=1
 while (i<=int(.23/dt)):
    ut[i]=(.865+(.875-.865)/(.23-.21)*dt*(i-(.21/dt)))*100
    i+=1
 while (i<=int(.265/dt)):
    ut[i]=(.875+(.87-.875)/(.265-.23)*dt*(i-(.23/dt)))*100
    i+=1
 while (i<=int(.29/dt)):
    ut[i]=(.87+(.86-.87)/(.29-.265)*dt*(i-(.265/dt)))*100
    i+=1
 while (i<=int(.31/dt)):
    ut[i]=(.86+(.84-.86)/(.31-.29)*dt*(i-(.29/dt)))*100
    i+=1
 while (i<=int(.33/dt)):
    ut[i]=(.84+(.80-.84)/(.33-.31)*dt*(i-(.31/dt)))*100
    i+=1
 while (i<=int(.36/dt)):
    ut[i]=(.80+(.74-.80)/(.36-.33)*dt*(i-(.33/dt)))*100
    i+=1
```

```
while (i<=int(.39/dt)):
    ut[i]=(.74+(.65-.74)/(.39-.36)*dt*(i-(.36/dt)))*100
    i+=1
while (i<=int(.41/dt)):
    ut[i]=(.65+(.55-.65)/(.41-.39)*dt*(i-(.39/dt)))*100
    i+=1
while (i<=int(.42/dt)):
    ut[i]=(.55+(.525-.55)/(.42-.41)*dt*(i-(.41/dt)))*100
    i+=1
while (i<=int(.44/dt)):
    ut[i]=(.525+(.51-.525)/(.44-.42)*dt*(i-(.42/dt)))*100
    i+=1
while (i<=int(.45/dt)):
    ut[i]=(.51+(.50-.51)/(.45-.44)*dt*(i-(.44/dt)))*100
    i+=1
while (i<=int(.47/dt)):
    ut[i]=(.50+(.51-.50)/(.47-.45)*dt*(i-(.45/dt)))*100
    i+=1
while (i<=int(.51/dt)):
    ut[i]=(.51+(.53-.51)/(.51-.47)*dt*(i-(.47/dt)))*100
    i+=1
while (i<=int(.54/dt)):
    ut[i]=(.53+(.54-.53)/(.54-.51)*dt*(i-(.51/dt)))*100
    i+=1
while (i<=int(.59/dt)):
    ut[i]=(.54+(.565-.54)/(.59-.54)*dt*(i-(.54/dt)))*100
    i+=1
while (i<=int(.65/dt)):
    ut[i]=(.565+(.58-.565)/(.65-.59)*dt*(i-(.59/dt)))*100
    i+=1
while (i<=int(.74/dt)):
    ut[i]=(.58+(.585-.58)/(.74-.65)*dt*(i-(.65/dt)))*100
    i+=1
while (i<=int(.78/dt)):
    ut[i]=(.585+(.58-.585)/(.78-.74)*dt*(i-(.74/dt)))*100
    i+=1
while (i<=int(.79/dt)):
    ut[i]=(.58+(.57-.58)/(.79-.78)*dt*(i-(.78/dt)))*100
```

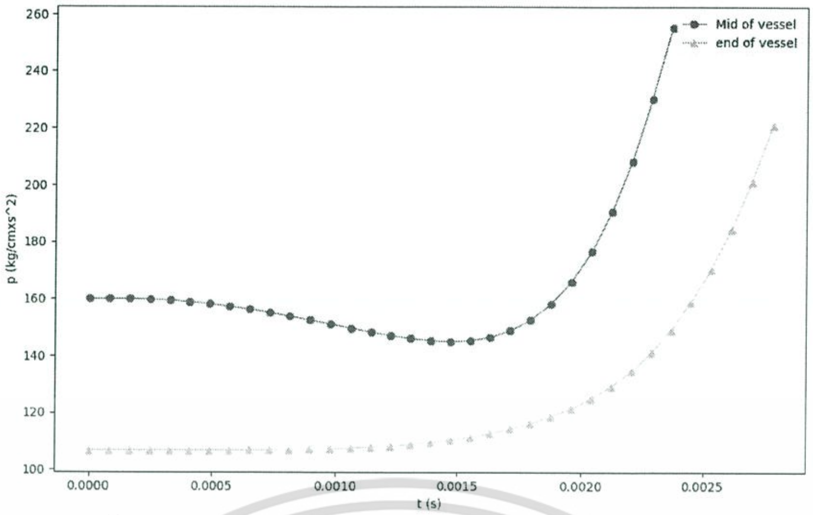
```
    i+=1
while (i<=int(.80/dt)):
    ut[i]=(.57+(.56-.57)/(.80-.79)*dt*(i-(.79/dt)))*100
    i+=1

t = 0
n = 0

while t <= 0.8:

    #หลอดเลือดที่1
    m=1
    an1 = a1.copy()
    pn1 = p1.copy()
    un1 = u1.copy()

    ##mass_eq
    a1=mass_conserv(at, an1, un1)

    ##pressure_area
    p1=pressure_area(pt, a1)

    ##mom_eq
    kr = build_up_kr(an1)
    u1=mom_conserv(ut, an1, un1, pn1, a1)

    p1mid[n]=p1[5]
    p1end[n]=p1[10]
    a1end=numpy.zeros(nt)
    a1end[:]=((p1end[:]-po)/b[m]+(ao[m])**0.5)**2

    #หลอดเลือดที่2
    m=2
    p2end=numpy.zeros(nt)
    u2end=numpy.zeros(nt)
    p2mid=numpy.zeros(nt)

    ##mass_eq
```

```
a2=mass_conserv(a1end, an2, un2)

##pressure_area
p2=pressure_area(p1end, a2)

##mom_eq
kr = build_up_kr(an2)
u2=mom_conserv(u1end, an2, un2, pn2, a2)

p2mid[n]=p2[3]
p2end[n]=p2[6]
a2end=numpy.zero(nt)
a2end[:]=((p2end[:]-po)/b[m]+(ao[m])**0.5)**2


#หลอดเลือดที่3
m=3
p3end=numpy.zeros(nt)
u3end=numpy.zeros(nt)
p3mid=numpy.zeros(nt)

##mass_eq
a3=mass_conserv(a1end, an3, un3)

##pressure_area
p3=pressure_area(p1end, a3)

##mom_eq
kr = build_up_kr(an3)
u3=mom_conserv(u1end, an3, un3, pn3, a3)

p3mid[n]=p3[(nx[m]-1)/2]
p3end[n]=p3[nx[m]-1]
a3end=numpy.zero(nt)
a3end[:]=((p3end[:]-po)/b[m]+(ao[m])**0.5)**2
.
.
.
```

```
n+=1
t+=dt
```

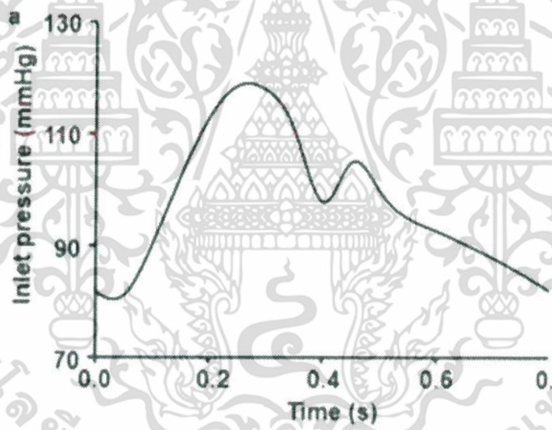
บทที่ 6
ผลการจำลอง

ในบทนี้กล่าวถึงผลที่ได้จากการเขียนโปรแกรมคอมพิวเตอร์ โดยใช้ภาษาไพธอนในการเขียน เพื่อการจำลองการไหลของเลือดเมื่อพิจารณาการไหลที่หลอดเลือดแดงใหญ่เอออร์ต้าที่ออกจากหัวใจโดยตรง เมื่อกำหนดขอบเขตและเงื่อนไขดังนี้ กำหนดให้ร่างกายแข็งแรง และอยู่ในสภาวะพักซึ่งมีลักษณะการไหลแบบราบเรียบ หลอดเลือดมีความยืดหยุ่นแบบอีลาสติก เลือดเป็นของไหลที่อัดตัวไม่ได้ ลักษณะการไหลเป็นแบบนิวโตเนียน ความหนาแน่นของเลือดมีค่า 1060 กิโลกรัมต่อลูกบาศก์เมตร เลือดมีค่าความหนืดสัมบูรณ์ 2.78 มิลลิปาสคาลวินาที เลือดมีค่าความหนืดจลน์ 2.65 ตารางมิลลิเมตร·วินาที ที่อุณหภูมิร่างกาย 37 องศาเซลเซียส ปริมาตรเลือดที่ออกจากหัวใจมีค่า 5 ลิตรต่อนาที หลอดเลือดมีความยืดหยุ่นแบบอีลาสติก ผนังภายในหลอดเลือดมีลักษณะเรียบ จึงจะได้ผลการทดลองจากการเขียนโค้ดทางคอมพิวเตอร์เพื่อจำลองการไหลของเลือดในหลอดเลือดอย่างง่ายโดยใช้ภาษาไพธอน ซึ่งใช้ระเบียบวิธีไฟไนท์ดิฟเฟอเรนซ์ 1 ทิศทาง เพื่อคำนวณหาค่าประมาณทางคณิตศาสตร์ที่กำหนดโดยสมการ (3.11) ซึ่งเป็นสมการของกฎของการอนุรักษ์มวล, สมการ (3.24) ที่เป็นสมการของกฎของการอนุรักษ์โมเมนตัม และสมการ (3.25) เป็นสมการความสัมพันธ์ระหว่างความดัน (P) และพื้นที่หน้าตัด (A) กำหนดค่าทางกายภาพของหลอดเลือด ซึ่งได้แก่ค่าความ หนาแน่นของเลือด ค่าความหนืดสัมบูรณ์ของเลือดและปริมาตรเลือดที่ออกจากหัวใจนำมาจากค่าคงที่ ของงานวิจัยที่ได้กำหนดไว้ในบทที่ 4 เมื่อกำหนดคุณสมบัติของเส้นเลือด พื้นที่หน้าตัดเริ่มต้น ความยาวของเส้นเลือด ความหนาและรัศมีของเส้นเลือดโดยอ้างอิงจาก [13] โดยในหลอดเลือดแต่ละเส้นมีพื้นที่หน้าตัดเริ่มต้นและความยาวของเส้นเลือดต่างกัน โดยความยาวแต่ละช่วงในหนึ่งโหนด คือ Δx = ความยาวรวม/(จำนวนโหนด-1) โดยให้จุดอ้างอิงในเส้นเลือดนั้นๆคือจุดเริ่มต้นหรือคือจุดทางเข้าของเลือด และระยะเวลาในหนึ่งช่วงเวลา (time step) คือ Δt = 0.8/(จำนวนช่วงเวลา-1) เมื่อหัวใจเต้น 0.8 วินาที/ครั้ง รอบการเต้นของหัวใจรอบแรกเพียงรอบเดียว เมื่อทราบค่าทั้งหมด กล่าวคือค่าของความดัน พื้นที่หน้าตัด และความเร็วที่เวลา n ต่อมาหาค่าของความดัน พื้นที่หน้าตัด และความเร็วของช่วงเวลาที่ $n-1$ ที่ไม่ทราบค่าได้โดยคำนวณแบบวิธีผลต่างข้างหลังเพื่อหาค่าของความดัน พื้นที่หน้าตัด และความเร็วที่วินาทีที่ $n-1$ ในสมการ เพื่อหาค่าของความดัน พื้นที่หน้าตัด และความเร็วย้อนหลังที่วินาทีที่ $n+1$ โดยใช้วิธีผลต่างข้างหน้า เมื่อ $n=0,1,2,3,\ldots$ ในการจำลองสถานการณ์นี้จะพิจารณาขอบเขตของการไหลทีละ 1 เส้นเลือด โดยเส้นเลือดแรกคือเส้นเลือดใหญ่เส้นแรกที่เลือดไหลผ่านเมื่อออกจากหัวใจหรือคือเส้นเลือดแดงใหญ่เอออร์ต้า เมื่อสมมติฐานของงานวิจัย คือค่าความดันที่ได้จากการทดลองเป็นฟังก์ชันคาบ ค่าความดันที่ได้จากการทดลองอยู่ในช่วง 80 ถึง 120 มิลลิเมตรปรอท และค่าความดันเฉลี่ยมีค่าลดลงเมื่อมีระยะห่างจากหัวใจมากขึ้น จากการเขียนโค้ดจะได้ผลการจำลองดังนี้

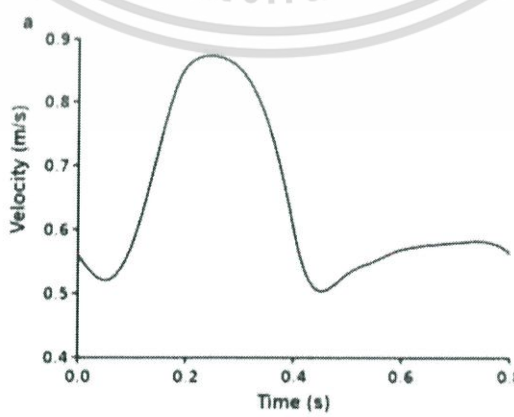


รูปที่ 6.1 กราฟความสัมพันธ์ระหว่างความดัน(p) กับเวลา(t)

ผลลัพธ์ที่ได้มาจากการจำลองโดยกำหนดค่าเงื่อนไขที่ขอบที่นำมาจากค่าของงานวิจัย [8] โดยมีทั้งค่าของความดันแสดงดังรูป 1.3 และความเร็วแสดงดังรูป 1.4 ที่หลอดเลือดแดงใหญ่เอออร์ต้า (Aorta)

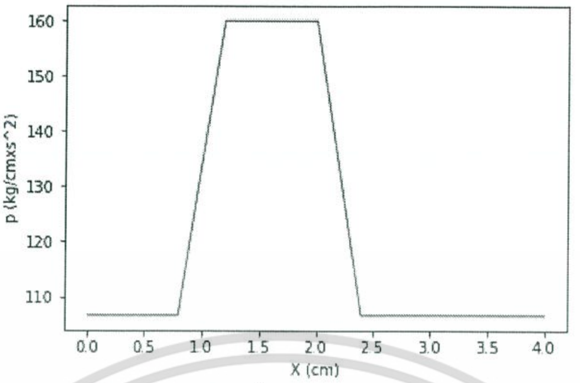


รูปที่ 1.3 ความดันที่หลอดเลือดแดงใหญ่ (Aorta)

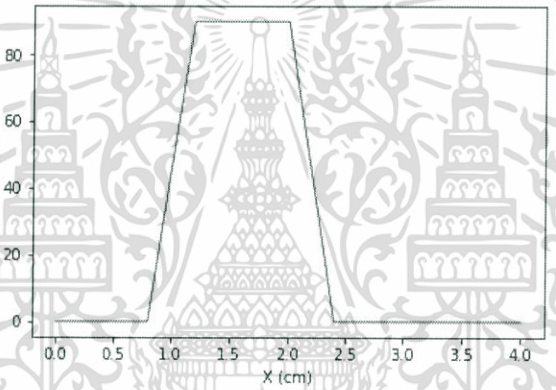


รูปที่ 1.4 ความเร็วที่หลอดเลือดแดงใหญ่ (Aorta)

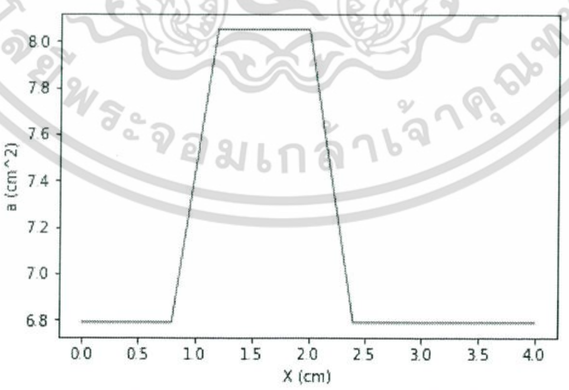
และกำหนดค่าเงื่อนไขเริ่มต้นกำหนดเป็นฟังก์ชันขั้นบันได โดยกำหนดค่าความดันเริ่มต้นเป็นดังรูปที่ 5.1 กำหนดค่าความเร็วเริ่มต้นเป็นดังรูป 5.2 และกำหนดค่าพื้นที่หน้าตัดเริ่มต้นเป็นดังรูป 5.3



รูปที่ 5.1 ฟังก์ชันขั้นบันไดของค่าความดัน



รูปที่ 5.2 ฟังก์ชันขั้นบันไดของค่าความเร็ว



รูปที่ 5.3 ฟังก์ชันขั้นบันไดของพื้นที่

จากผลลัพธ์ที่ได้ในรูปของกราฟความดันกับเวลาที่ได้นั้นไม่สอดคล้องกับผลที่ควรจะเป็นเพราะเห็นได้ว่ากราฟความดันกับเวลาที่ได้นั้นมีลักษณะลู่ออก

บทที่ 7
สรุปและข้อเสนอแนะ

7.1 อภิปรายและสรุปแบบจำลอง

งานวิจัยนี้ทำการจำลองการไหลของเลือดในหลอดเลือดแดงใหญ่และหลอดเลือดแดงกลางโดยใช้ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ 1 ทิศทางในการจำลอง เพื่อให้เป็นประโยชน์ต่อการรักษาคนไข้ที่มีปัญหาเกี่ยวกับโรคหลอดเลือด โดยเฉพาะโรคหลอดลือดหัวใจที่มีอัตราการเสียชีวิตของคนไทยเพิ่มขึ้นทุกๆปี และในงานวิจัยนี้สามารถพัฒนาต่อได้อีก อาทิเช่น โรคไขมันอุดตันในเส้นเลือดที่ต้นขา ถ้าทางการแพทย์สามารถทราบความดันที่เปลี่ยนไปจากปกติอย่างชัดเจนจะยิ่งช่วยให้การรักษาเป็นไปได้ง่าย โดยวิธีการจำลองการไหลของเลือดในงานวิจัยนี้ทำโดยการเขียนโค้ดในโปรแกรมคอมพิวเตอร์ ซึ่งได้นำกฎการอนุรักษ์มวล กฎการอนุรักษ์โมเมนตัม สมการความสัมพันธ์ระหว่างความดันกับพื้นที่มาช่วยในการเขียนโค้ดการแก้ปัญหาเชิงอนุพันธ์นั้นมีหลายวิธี อาทิเช่น ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ (Finite difference method) ระเบียบไฟไนต์เอลิเมนต์ (Finite element method) และระเบียบฟไนต์วอลุม (Finite volume method) เป็นต้น วิธีการแก้ปัญหาที่เลือกใช้ในงานวิจัยนี้คือ ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ (Finite difference method) เนื่องจากระเบียบวิธีนี้สามารถประมาณค่าได้ทั้งอนุพันธ์อันดับหนึ่งและอนุพันธ์อันดับสูงๆ โดยเลือกใช้วิธีแก้ปัญหาที่เฉพาะเจาะจงขึ้น คือ ใช้ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ 1 ทิศทาง จากบทที่ 3.2 ได้ทำการอธิบายเหตุผลของการเลือกทำการจำลองการไหลใน 1 ทิศทาง เนื่องจากในงานวิจัยนี้ต้องการวัดค่าความดันเป็นหลัก และเปรียบเสมือนหลอดเลือดแดงเป็นท่อทรงกระบอกท่อหนึ่ง เมื่อทำการตัดหน้าตัดจุดใดจุดหนึ่งของท่อทรงกระบอกจะได้ค่าความดันที่เท่ากันทุกจุด

วิธีการดำเนินงานที่สำคัญของงานวิจัยนี้อีกอย่างหนึ่งคือการพิสูจน์สมการควบคุมของการจำลอง ได้แก่ สมการอนุรักษ์มวลและสมการอนุรักษ์โมเมนตัม ซึ่งมีข้อมูลปรากฎอยู่ในบทที่ 3 กลศาสตร์ของไหล โดยของไหลสามารถแบ่งประเภทได้หลายแบบ อย่างแรกคือ ความสามารถในการบีบอัดตัวของของไหล แบ่งได้เป็นของไหลแบบอัดตัวได้และของไหลแบบอัดตัวไม่ได้ ซึ่งดูได้จากการที่ของไหลนั้นมีความหนาแน่นคงที่หรือไม่ ถ้าของไหลมีความหนาแน่นไม่คงที่นั่นคือของไหลแบบอัดตัวได้ อย่างที่สองคือ ความหนืด โดยจะขึ้นอยู่กับการไหลของของไหลนั้นๆ ในงานวิจัยนี้ศึกษาการไหลขณะที่ของไหลมีการเคลื่อนที่อยู่ในท่อกลม ความต้านทานที่เกิดขึ้นนั้นเกิดเมื่อของไหลเคลื่อนที่ผ่านผิวเรียบและมีส่วนที่ของไหลสัมผัสกับผิวเรียบนั้น อย่างที่สามคือ ลักษณะการไหลตามความหนืด โดยสามารถแบ่งย่อยลงไปได้อีกคือ ของไหลแบบนิวโตเนียนและของไหลแบบนอนนิวโตเนียน ของไหลในงานวิจัยนี้เป็นของไหลแบบนิวโตเนียน มีนิยามว่าความหนืดของของไหลเป็นค่าคงที่ที่อุณหภูมิๆหนึ่ง และเป็นของไหลที่ความเค้นเฉือนแปรผันเป็นเส้นตรงกับความเครียดเฉือน อย่างที่สี่คือ ลักษณะการไหลพิจารณาจากเลขเรย์โนลด์ แบ่งได้เป็นการไหลแบบราบเรียบ การไหลแบบผสม และการไหลแบบปั่นป่วน โดยทั้งสามประเภทจำแนกได้จากเลขเรย์โนลด์ คือ ค่าเลขของเรย์โนลด์อยู่ในช่วงน้อยกว่า 2300 นั่นคือการไหลแบบราบเรียบ ค่าเลขของเรย์โนลด์อยู่ในช่วง 2300-4000 นั่นคือการไหลแบบผสม ค่าเลขของเรย์โนลด์อยู่ในช่วงมากกว่า 4000 นั่นคือการไหลแบบปั่นป่วน โดยค่าที่กล่าวไปข้างต้นสามารถคำนวณได้จากสมการที่ (3.1) สำหรับลักษณะการไหลของเลือดในร่างกายที่แข็งแรงในสภาวะพักเป็นการไหลแบบราบเรียบ และจะกลายเป็นการไหลแบบปั่นป่วนเมื่อเลือดไหลด้วยความเร็วที่มากขึ้น เช่น ขณะออกกำลังกาย โดยในงานวิจัยนี้กำหนดการจำลองของเลือดในสภาวะพัก ในการพิสูจน์สมการอนุรักษ์มวลและสมการอนุรักษ์โมเมนตัมนั้นเริ่มมาจากการหาสมการ

ควบคุมการไหลของของไหลในหนึ่งมิติจากทฤษฎีการถ่ายเทของเรย์โนลด์โดยพิจารณาท่อซึ่งแสดงในรูปที่ 3.7 เป็นหลอดเลือด โดยกำหนดให้หลอดเลือดนั้นเป็นเส้นตรงตามแนวแกน x ในการหาสมการควบคุมการไหลของของไหลในหนึ่งมิติเริ่มจากทฤษฎีการถ่ายเทของเรย์โนลด์ โดยพิจารณาปริมาตรควบคุม V_t ที่มีขอบเขต $\partial \mathrm{V}_t$ และเวกเตอร์หนึ่งหน่วย $\bar{n}$ ซึ่งตั้งฉากกับพื้นที่หน้าตัด เมื่อ $f = f(t, \bar{x})$ เป็นฟังก์ชันที่มีความต่อเนื่อง

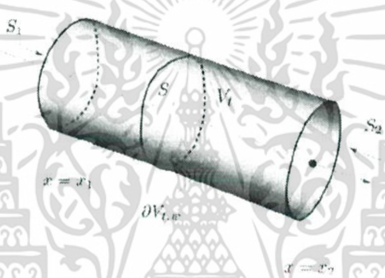
$$\frac{d}{dt}\int_{\mathrm{V}_t} f\, d\mathrm{V} = \int_{\mathrm{V}_t} \frac{\partial f}{\partial t} d\mathrm{V} + \int_{\partial \mathrm{V}_t} f\, \bar{u}_b \cdot \bar{n}\, d\sigma \tag{3.2}$$

โดยที่

$\bar{x}$ คือ เวกเตอร์ตำแหน่งที่พิกัด (x, y, z)

f คือ คุณสมบัติภายใน หรือ คุณสมบัติภายนอกต่อมวล

$\bar{u}_b$ คือ ความเร็วของขอบเขตของปริมาตร V_t



รูปที่ 3.7 แสดงหลอดเลือดที่เป็นเส้นตรงตามแนวแกน x

ผู้จำลองได้พิสูจน์ทฤษฎีการถ่ายเทของเรย์โนลด์ดังที่ปรากฏในหัวข้อที่ 3.2.1 ทฤษฎีการถ่ายเทของเรย์โนลด์(Reynolds' transport theorem) ได้รูปสุดท้ายในหนึ่งมิติดังสมการที่ (3.9)

$$\frac{\partial}{\partial t}\left(\mathrm{A}\bar{f}\right) + \frac{\partial}{\partial x}\left[A(\overline{fu_1})\right] = \int_{\mathrm{S}}\left[\frac{\partial f}{\partial t} + \nabla \cdot (f\bar{u})\right] d\sigma + \int_{\partial \mathrm{S}} f\,\bar{w} \cdot \bar{n}\, d\gamma \tag{3.9}$$

พิจารณาสมการอนุรักษ์มวลในท่อที่มีความยืดหยุ่นโดยแทน $f = \frac{m}{m} = 1$ ในสมการ (3.9) และทราบว่าเลือดเป็นของไหลแบบอัดตัวไม่ได้มีสมบัติ $\nabla \cdot \bar{u} = 0$ ทำให้ได้ว่ากำหนดให้ไม่มีของไหลไหลผ่านผนังของหลอดเลือด($\bar{w} \cdot \bar{n} = 0$) ได้รูปสุดท้ายของสมการอนุรักษ์มวลดังสมการที่ (3.11)

$$\frac{\partial A}{\partial t} + \frac{\partial}{\partial x}\left(A\bar{u}_1\right) = 0 \tag{3.11}$$

พิจารณาสมการอนุรักษ์โมเมนตัมโดยแทน $f = \frac{mu_1}{m} = u_1$ ในสมการ (3.9) และผู้จำลองได้พิสูจน์สมการอนุรักษ์โฒเมนตัมดังที่ปรากฏในหัวข้อที่ 3.2.3 สมการอนุรักษ์โมเมนตัม ได้รูปสุดท้ายดังสมการที่ (3.24)

$$\frac{\partial}{\partial t}\left(\mathrm{A}\bar{u}_1\right)+\frac{\partial}{\partial x}\left(\alpha\bar{u}_1^{\ 2}\right)+\frac{A}{\rho}\left(\frac{\partial\overline{p}}{\partial x}\right)+K_R\bar{u}_1=0 \tag{3.24}$$

ตัวแปรที่ไม่ทราบค่าในสมการ (3.11) และ (3.24) คือค่าความดัน p พื้นที่หน้าตัด A และความเร็วเฉลี่ย $\bar{u}_1$ ซึ่งเห็นว่าจำนวนตัวแปรเกินจำนวนสมการ ดังนั้นวิธีการแก้ปัญหาที่ทำได้คือการกำหนดความสัมพันธ์ระหว่างความดันของหลอดเลือด p กับพื้นที่หน้าตัดของหลอดเลือด A ดังที่ปรากฏในหัวข้อที่ 3.2.4 ความสัมพันธ์ระหว่างความดันและพื้นที่หน้าตัดของท่อ

$$p=p_{ext}+\beta\left(\sqrt{A}-\sqrt{A_0}\right) \tag{3.25}$$

เมื่อ

$$\beta=\frac{Eh_0}{D_0A_0} \tag{3.26}$$

โดยที่

h_0 คือความหนาของหลอดเลือดที่สภาวะสมดุล

$A_0=A_0\left(x\right)$ คือพื้นที่หน้าตัดที่สภาวะสมดุล

D_0 คือรัศมีของหลอดเลือดที่สภาวะสมดุล

$\left(p,Q\right)=\left(P_{ext},0\right)$ ที่สภาวะสมดุล

$E=E\left(x\right)$ คือค่าโมดูลัสของยัง

P_{ext} คือ ความดันภายนอก (กำหนดให้มีค่าคงที่)

งานวิจัยนี้ทำการเขียนโค้ดโดยใช้ภาษาไพธอน (Python) ซึ่งนอกจากนำกฎการอนุรักษ์มวล กฎการอนุรักษ์โมเมนตัม สมการความสัมพันธ์ระหว่างความดันกับพื้นที่มาช่วยในการเขียนโค้ดแล้ว ยังต้องทำการกำหนดเงื่อนไขเริ่มต้น คุณสมบัติต่างๆของเลือดและหลอดเลือด และเงื่อนไขขอบ ลงในการเขียนโค้ดของภาษาไพธอน (Python) อีกด้วย นั่นก็คือกำหนดให้เลือดเป็นของไหลที่อัดตัวไม่ได้ มีค่าความหนาแน่นคงที่คือ 1060 กิโลกรัมต่อลูกบาศก์เมตร ความหนืดจลน์คงที่ คือ 2.78 มิลลิปาสคาลวินาที ที่อุณหภูมิร่างกาย 37 องศาเซลเซียสเป็นการไหลแบบนิวโตเนียน หลอดเลือดมีความยืดหยุ่นแบบอิลาสติกและผนังภายในหลอดเลือดมีลักษณะเรียบ และปริมาตรเลือดที่ออกจากหัวใจมีค่า 5 ลิตรต่อนาที โดยข้อกำหนดทั้งหมดนี้ทดลองกับร่างกายที่มีสุขภาพแข็งแรงและอยู่ในสภาวะพัก

ในงานวิจัยนี้ได้เริ่มกำหนดข้อมูลในการเขียนโค้ดจากหลอดเลือดแดงใหญ่เอออร์ตา (Aorta) ซึ่งเป็นหลอดเลือดแดงใหญ่ที่มีขนาดใหญ่ที่สุดและมีหน้าที่ลำเลียงเลือดที่ถูกสูบฉีดจากหัวใจห้องซ้ายล่างส่งไปยังหลอดเลือดแดงอาร์เทอรี (Artery) และลำเลียงเลือดต่อไปเพื่อไปเลี้ยงส่วนต่างๆของร่างกาย โดยข้อมูลของหลอดเลือดแดงใหญ่เอออร์ต้า (Aorta) มีดังนี้ ความยาวมีค่า 4 เซนติเมตร และ พื้นที่หน้าตัดมีค่า 5.983 ตารางเซนติเมตร โดยการเขียนโค้ดในงานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาลักษณะการไหลของเลือดและเพื่อให้ได้ค่าความดันที่จุดต่างๆในเส้นเลือดแดงใหญ่เอออร์ตา (Aorta) และเส้นเลือดแดงกลางอาร์เทอรี (Artery) เพื่อนำไปใช้ประโยชน์ทางการแพทย์ต่อไป

เมื่อทำการดำเนินการเขียนโค้ดแล้ว ผลลัพธ์ที่ได้คือเป็นค่าละเอียดที่จุดต่างๆของหลอดเลือดแดงใหญ่เอออร์ต้า (Aorta) ในแต่ละวินาที โดยที่สามารถกำหนดคำสั่งให้ค่าทั้งหมดที่ได้นี้ออกมาเป็นกราฟเพื่อให้เห็นค่าความดันที่เพิ่มขึ้นและลดลงอย่างชัดเจน และค่าความดันที่ได้จากการทดลองนั้นมีค่ามากกว่าค่าความดันปกติของมนุษย์ที่ร่างกายมีสุขภาพแข็งแรงและอยู่ในสภาวะพัก ค่าดังกล่าวคือ 120/80 มิลลิเมตรปรอท โดยที่ค่า 120 คือ ค่าความดันในช่วงการบีบตัวของหัวใจ และ 80 คือ ค่าความดันในช่วงการคลายตัวของหัวใจ ค่าความดัน 120/80 นี้คือค่ามาตรฐานของหลอดเลือดแดงกลางบริเวณต้นแขน ซึ่งสังเกตุได้จากเวลาที่ผู้ป่วยไปตรวจวัดความดัน แต่ค่าที่ได้จากการทดลองอาจสันนิษฐานได้ว่าเป็นการคำนวณจากหลอดเลือดแดงใหญ่เอออร์ตา (Aorta) โดยตรง จึงมีค่ามากกว่า 120/80 มิลลิเมตรปรอท

เมื่อนำค่าผลลัพธ์จากการทดลองมาพลอต (Plot) เป็นกราฟแล้วทำให้เห็นได้ว่าค่าความดันมีลักษณะลู่ออก ดังนั้นจึงไม่สามารถดำเนินการคำนวณต่อได้ เนื่องจากต้องนำค่าความดันที่จุดสุดท้ายของหลอดเลือดแดงใหญ่เออร์ต้า (Aorta) ไปเป็นค่าเริ่มต้นความดันของหลอดเลือดแดงใหญ่ต่อไป

การลู่ออกเป็นผลเนื่องมาจากระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์ เนื่องจากสมการนาเวียร์สโตกมีผลลัพธ์สัมพันธ์กับเวลาอยู่ในรูปไฮเพอร์โบลิก และเมื่อทำการขยายไปที่เส้นๆหนึ่ง เส้นนั้นไม่ได้แสดงเป็นเส้นตรง แต่เป็นเส้นที่อยู่ในรูปฟังก์ชันตรีโกณมิติ หากกำหนดขั้นเวลาได้พอดีกับคาบของฟังก์ชันตรีโกณมิติ ผลที่ได้จึงจะเป็นเส้นตรง แต่หากกำหนดขั้นเวลาไม่เหมาะสม ผลที่ได้จะมีค่าลู่ออก ซึ่งการกำหนดขั้นเวลามีโอกาสผิดพลาดสูงมาก

7.2 ข้อเสนอแนะ

ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์และระเบียบวิธีไฟไนต์เอลิเมนต์ มีโอกาสทำให้ผลลัพธ์มีค่าลู่ออกสูงมาก เนื่องจากใช้ความชันแก้สมการนาเวียร์สโตก ซึ่งหากใช้ระเบียบวิธีไฟไนต์ดิฟเฟอเรนซ์และระเบียบวิธีไฟไนต์เอลิเมนต์ต้องทำการฉายผลลัพธ์ให้อยู่ในทิศที่ควรจะเป็นของเส้นการไหล โดยปรับเกรเดียนต์ด้วยวิธีเกรเดียนต์รีคัฟเวอรี (gradient recovery) หรือใช้ระเบียบวิธีไฟไนต์วอลุม เนื่องจากสมการอนุรักษ์ฟลักซ์บังคับให้ทิศทางการไหลเป็นแนวเส้นตรง เมื่อปรับปรุงวิธีการดำเนินงานได้แล้วจะทำให้งานวิจัยสามารถพัฒนาและต่อยอดไปได้ โดยถ้าค่าความดันของหลอดเลือดแดงเส้นแรกที่ออกจาหัวใจมีลักษณะลู่เข้า ก็สามารถดำเนินการเขียนโค้ดและทำให้ทราบค่าความดันของหลอดเลือดแดงเส้นต่อๆไปได้ เพื่อให้เป็นประโยชน์สูงสุดต่อทางการแพทย์และมีประโยชน์กับผู้ที่กำลังศึกษาวิธีการจำลองการไหลของของไหลต่อไป

บรรณานุกรม

[1]งานธนาคารเลือด โรงพยาบาลมหาราชนครเชียงใหม่.*หน้าที่ของโลหิตและส่วนประกอบของโลหิต*, [ระบบออนไลน์], แหล่งที่มา http://www.med.cmu.ac.th, เข้าดูเมื่อวันที่ 28/10/2561.

[2]นายแพทย์ธีรวัฒน์ สุวรรณี. *การทำงานของระบบหัวใจ*, [ระบบออนไลน์], แหล่งที่มา http://www.idoctorhouse.com, เข้าดูเมื่อวันที่ 02/11/2561.

[3]พญ.อรอนงค์ กุละพัฒน์. *พลศาสตร์ของเลือด*, [ระบบออนไลน์], แหล่งที่มา http://www.cai.md.chula.ac.th, เข้าดูเมื่อวันที่ 26/10/2561.

[4]ศาสตราจารย์คลินิกเกียรติคุณนายแพทย์พนัส เฉลิมแสนยากร. *หัวใจ: กายวิภาคหัวใจ(Heart anatomy)/ สรีรวิทยาของหัว(Heart physiology)*
, [ระบบออนไลน์], แหล่งที่มา http://haamor.com, เข้าดูเมื่อวันที่ 02/11/2561.

[5]สารานุกรมไทยสำหรับเยาวชน.*ระบบการไหลเวียนเลือด*, [ระบบออนไลน์], แหล่งที่มา http://kanchanapisek.or.th/kp 6/index.php, เข้าดูเมื่อวันที่ 26/10/2561.

[6]A.P. Avolio. 1980. **Multi-branched model of the human arterial system**. Doctor in Philosophy. Department of Biomedical Sciences. Faculty of Medicine and Health Scinces

[7]Cristóbal Rodero Gómez. 2017. **Analysis of blood flow in one dimensional elastic artery using Navier-Stokes conservation laws.** Facultat de Matemàtiques. Universitat de València

[8]Eunji Shin , Jung Joo Kim , Seonjoong Lee , et al. 2018. **Hemodynamics in diabetic human aorta using computational fluid dynamics.** College of medicine. Inje University

[9]Luca Formaggia , Alfio Quarteroni , Alessandro Veneziani. 2009. **Cardiovascular Mathematics Modeling and simulation of the circulatory system**. Italy : Springer

[10]Luca Formaggia , Daniele Lamponi , Alfio Quarteroni . 2003. **One-dimensional models for blood flow in arteries.** Institut de Mathématiques. École Polytechnique Fédérale de Lausanne

[11]Rafik Absi. 2018. **Revisting the pressure-area relation for the flow in elastic tubes:Application to arterial vessels.** Doctor in Philosophy. Ecole de Biologie Industrielle France

[12]Sansuke M. Watanabe , Pablo J. Blanco and Raúl A. Feijóo. 2011. **Blood flow modeling in a detailed arterial network of the arm.** Instituto Nacional de Ciênciae Tecnologia. Medicina Assistida por Computac¸ ˜ao Cient´ıfica, Brazil

[13]S.J. Sherwin , L. Formaggia , J. Peiro and V. Franke. 2003. **Computational modelling of 1D blood flow with variable mechanical properties and its application to the simulation of wave propagation in the human arterial system.** Department of Aeronautics. Imperial College London