

การวิเคราะห์การแข่งตัวในการหล่อแบบต่อเนื่อง



นายปิยะ แก้วนพรัตน์

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโลหการ ภาควิชาวิศวกรรมโลหการ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2545

ISBN 974-17-2712-7

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

SOLIDIFICATION ANALYSIS IN CONTINUOUS CASTING



Mr. Piya Kaewnopparat

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Metallurgical Engineering

Department of Metallurgical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2002

ISBN 974-17-2712-7

หัวข้อวิทยานิพนธ์	การวิเคราะห์การแข็งตัวในการหล่อแบบต่อเนื่อง
โดย	นาย ปิยะ แก้วนพรัตน์
สาขาวิชา	วิศวกรรมโลหการ
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. อธิธิพล เตียวณิชย์
อาจารย์ที่ปรึกษาร่วม	Professor Takateru Umeda, D.Eng.

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร. ประสงค์ ศรีเจริญชัย)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร. อธิธิพล เตียวณิชย์)

..... อาจารย์ที่ปรึกษาร่วม
(Professor Takateru Umeda, D.Eng.)

..... กรรมการ
(อาจารย์ สุวันชัย พงษ์สุกิจวัฒน์)

ปิยะ แก้วนพรัตน์ : การวิเคราะห์การแข็งตัวในการหล่อแบบต่อเนื่อง. (Solidification Analysis in Continuous Casting) อ. ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร. อธิธิพล เดี่ยววณิชย์, อ.ที่ปรึกษาร่วม : Professor Takateru Umeda, D.Eng. ; 95 หน้า. ISBN 974-17-2712-7.

งานวิจัยนี้เป็นการพัฒนาโปรแกรมคอมพิวเตอร์เพื่อคำนวณการถ่ายเทความร้อนในการหล่อแบบต่อเนื่องสำหรับโลหะต่างๆที่มีรูปร่างแตกต่างกันโดยใช้วิธีการผลต่างสี่เหลี่ยม (Finite difference method) เพื่อศึกษาการแข็งตัวในกระบวนการและผลของพารามิเตอร์ต่างๆในกระบวนการ ได้มีการพัฒนาโปรแกรมสามโปรแกรม สองโปรแกรมแรกเป็นโปรแกรมสำหรับการหล่อแบบต่อเนื่องของเหล็กกล้าและอีกหนึ่งโปรแกรมสำหรับการหล่อแบบต่อเนื่องสำหรับโลหะนอกกลุ่มเหล็ก การนำความร้อนที่ต่ำของเหล็กกล้าและความเร็วการหล่อที่สูงจึงสามารถตัดพจน์การนำความร้อนในทิศทางการดึงออกได้ในกรณีสถานะอยู่ตัว ในกรณีของเหล็กแท่งแบนและเหล็กแท่ง จึงสามารถลดสมการให้อยู่ในรูปสมการการถ่ายเทความร้อนภายใต้สถานะชั่วคราวแบบหนึ่งมิติและสองมิติได้ สำหรับโปรแกรมที่สามใช้กับโลหะนอกกลุ่มเหล็กทั่วไปและสำหรับช่วงระยะเริ่มต้นของกระบวนการในเหล็กกล้า จากนั้นทำการตรวจสอบความถูกต้องของโปรแกรมโดยเปรียบเทียบกับงานวิจัยอื่นที่ได้ยืนยันผลกับการทดลอง ซึ่งผลจากโปรแกรมและผลที่ได้จากการวัดจริงสอดคล้องกันดี จากนั้นจึงใช้โปรแกรมคำนวณลำดับการแข็งตัวในกระบวนการ และตรวจสอบความไวของพารามิเตอร์ของกระบวนการ ซึ่งผลการคำนวณแสดงให้เห็นว่าพารามิเตอร์ที่ไวที่สุดได้แก่ ความเร็วในการหล่อและการเย็นตัวทุติยภูมิ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา...วิศวกรรมโลหการ.....ลายมือชื่อนิสิต.....
สาขาวิชา...วิศวกรรมโลหการ..... ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา.....2545..... ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

4270421321 : MAJOR METALLURGICAL ENGINEERING

KEY WORD: CONTINUOUS CASTING / SOLIDIFICATION / HEAT TRANSFER /
NUMERICAL MODEL / SIMULATION

PIYA KAEWNOPPARAT : SOLIDIFICATION ANALYSIS IN CONTINUOUS
CASTING. THESIS ADVISOR : ASST. PROF. ITTIPON DIEWWANIT, Sc.D.,
THESIS CO-ADVISOR : PROFESSOR TAKATERU UMEDA, D.Eng., 95 pp. ISBN
974-17-2712 -7.

Computer programs for calculating the heat transfer in continuous casting for various metals with different ingot shapes were developed based on finite difference method. Three types of algorithms were constructed; two for the continuous casting process for steels and one for generic. Low thermal conductivity of steels and relatively rapid casting rate allow, in the case of steady state, exclusion of the conduction term in the pulling direction. For slab and billet, this can reduce the problem to one and two-dimensional transient heat transfer, respectively. The third one is generic for non-ferrous metals and for the initial stage for steel. Then validation for programs was performed by comparing to other literatures which had been confirmed by experiments. Agreement between results from the program and actual measurement was well. After that the programs were used to calculate solidification progress in the process. By checking the sensibility of operating parameters it was shown that the most sensitive parameters were casting speed and secondary cooling.

สถาบันทศยบรกร
จุฬาลงกรณ์มหาวิทยาลัย

Department...Metallurgical Engineering..... Student's signature.....

Field of study...Metallurgical Engineering... Advisor's signature.....

Academic year ...2002....

Co-advisor's signature.....

กิตติกรรมประกาศ

งานวิจัยชิ้นนี้สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากความช่วยเหลือจากหลายๆ ท่าน ข้าพเจ้าขอขอบพระคุณ Professor Takateru Umeda ผู้ช่วยศาสตราจารย์ ดร. อธิพิณ เดี่ยว วณิชย์ และ อาจารย์ สุวันชัย พงษ์สุกิจวัฒน์ เป็นอย่างสูงสำหรับการอุทิศตนและสละเวลาอันมีค่าของพวกท่านในการถ่ายทอดความรู้ อบรมสั่งสอน ตลอดจนให้กำลังใจและคำแนะนำต่างๆ ทั้งในด้านการเรียนและการทำงานด้วยความเมตตา นายสุตสาคร อินธิเดช , Mr. Huy Tran Duc และเพื่อนๆ กลุ่มวิจัยการแข็งตัวของน้ำโลหะที่สละเวลาและแรงงานช่วยเหลือให้การทำงานสำเร็จลุล่วงไปได้ด้วยดี ตลอดจนผู้ที่มีส่วนเกี่ยวข้องที่ข้าพเจ้ามิได้เอ่ยนามในที่นี้ทุกท่าน

สุดท้ายนี้ข้าพเจ้าขอขอบพระคุณบิดามารดาและบุคคลอันเป็นเคารพรักสำหรับกำลังใจและการสนับสนุนทางการศึกษาตลอดมา



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญภาพ.....	ฌ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	1
1.3 ขอบเขตของงานวิจัย.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
2 การศึกษาข้อมูลเบื้องต้น.....	3
2.1 แนวคิดและทฤษฎี.....	3
2.2 พื้นฐานของการถ่ายเทความร้อน.....	4
2.3 การถ่ายเทความร้อนในการหล่อแบบต่อเนื่อง.....	5
2.4 ระเบียบวิธีการในการแก้ปัญหาทางความร้อนในกระบวนการหล่อแบบต่อเนื่อง.....	6
3 วิธีดำเนินการวิจัย.....	8
3.1 สมการพื้นฐานของการหล่อแบบต่อเนื่อง.....	8
3.2 โปรแกรมแสดงอุณหภูมิที่ตำแหน่งต่างๆแบบสองมิติในสภาวะอยู่ตัวที่พัฒนาจากแบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ เพื่อใช้ในการหล่อแบบต่อเนื่องแบบ แท่ง สำหรับเหล็กกล้า.....	9
3.2.1 สมการเบื้องต้น.....	9
3.2.2 แบบจำลองการดุลความร้อน (Heat balance).....	10
3.3 โปรแกรมแสดงอุณหภูมิที่ตำแหน่งต่างๆแบบสามมิติในสภาวะอยู่ตัวที่พัฒนาจากแบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ เพื่อใช้ในการหล่อแบบต่อเนื่องแบบ แท่ง สำหรับเหล็กกล้า.....	12
3.3.1 สมการเบื้องต้น.....	12
3.3.2 การดุลความร้อน	15

3.4	โปรแกรมแสดงอุณหภูมิที่ตำแหน่งต่างๆแบบสามมิติในสภาวะชั่วคราว ที่พัฒนาจากแบบจำลองที่คิดการถ่ายเทความร้อนในสามมิติ เพื่อใช้ในการหล่อแบบต่อเนื่อง สำหรับโลหะนอกกลุ่มเหล็กและเหล็กกล้าในช่วงเริ่มต้นของกระบวนการ.....	17
3.4.1	สมการเบื้องต้น.....	17
3.4.2	การคูณความร้อน	18
3.5	การคิดความร้อนแฝงจากการแข็งตัว.....	19
3.6	แผนภาพแสดงลำดับขั้นตอนการคำนวณโดยใช้โปรแกรมทางคอมพิวเตอร์.....	20
4	ผลการวิเคราะห์ข้อมูล.....	24
4.1	แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติเพื่อใช้ในการหล่อแบบต่อเนื่องแบบแท่งแบนสำหรับเหล็กกล้า.....	24
4.2	แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติเพื่อใช้ในการหล่อแบบต่อเนื่องแบบแท่งสำหรับเหล็กกล้า.....	41
4.3	ตัวอย่างผลการคำนวณการเย็นตัวสำหรับการหล่อแท่งอะลูมิเนียม.....	47
5	สรุปผลการวิจัย.....	53
	รายการอ้างอิง.....	54
	ภาคผนวก.....	55
	ประวัติผู้เขียนวิทยานิพนธ์.....	95

สารบัญภาพ

	หน้า
ภาพที่ 2.1 การหล่อแบบต่อเนื่องประเภทต่างๆ.....	3
ภาพที่ 2.2 แสดงบริเวณต่างๆของการหล่อแบบต่อเนื่อง.....	6
ภาพที่ 3.1 แสดงการแบ่งชั้นของน้ำโลหะออกเป็นเอลิเมนต์ย่อย.....	10
ภาพที่ 3.2 แสดงทิศทางและรูปร่างของกระบวนการ.....	11
ภาพที่ 3.3 แสดงการถ่ายเทความร้อนแบบหนึ่งมิติ.....	12
ภาพที่ 3.4 แสดงการแบ่งชั้นของน้ำโลหะออกเป็นเอลิเมนต์ย่อย.....	14
ภาพที่ 3.5 แสดงทิศทางและรูปร่างของกระบวนการ.....	14
ภาพที่ 3.6 แสดงการถ่ายเทความร้อนแบบสองมิติ.....	16
ภาพที่ 3.7 แสดงเงื่อนไขขอบเขต.....	16
ภาพที่ 3.8 แสดงแนวการคิดรูปที่โดยใช้ระเบียบวิธีความแตกต่างและทิศทางการถ่ายเท ความร้อนในระนาบสามมิติ.....	17
ภาพที่ 3.9 แสดงการแบ่งรูปโดยใช้ระเบียบวิธีความแตกต่างและทิศทางการถ่ายเท ความร้อนในระนาบสามมิติ.....	18
ภาพที่ 3.10 แสดงการคิดค่าความร้อนจำเพาะที่อุณหภูมิต่างๆ โดยใช้ระเบียบวิธีเทียบเท่า ความร้อนจำเพาะเพื่อชดเชยความร้อนแฝงจากการแข็งตัว.....	20
ภาพที่ 3.11 แสดงการคิดการชดเชยความผิดพลาดที่เกิดจากลักษณะการแข็งตัวแบบต่างๆ ในแต่ละเอลิเมนต์ของระเบียบวิธีการกึ่งอนุกรม.....	21
ภาพที่ 3.12 แสดงลำดับขั้นตอนในการหาการกระจายตัวของอุณหภูมิในการถ่ายเท ความร้อนในระนาบหนึ่งมิติและสองมิติ.....	22
ภาพที่ 3.13 แสดงลำดับขั้นตอนในการหาการกระจายตัวของอุณหภูมิในการถ่ายเท ความร้อนในระนาบสามมิติ.....	23
ภาพที่ 4.1 แสดงผลการคำนวณการเย็นตัวที่ตำแหน่งต่างๆ โดยใช้แบบจำลองที่คิด การถ่ายเทความร้อนในหนึ่งมิติ.....	25
ภาพที่ 4.2 แสดงผลการคำนวณการเย็นตัวที่ตำแหน่งต่าง (แสดงด้วยเส้นกราฟ) โดยมี ผลเปรียบเทียบกับข้อมูลที่ได้จากการวัดจริงในโรงงาน (แสดงเป็นจุด ในภาพ).....	26
ภาพที่ 4.3 รูปร่างของบ่อน้ำโลหะที่คำนวณจากแบบจำลองที่คิดการถ่ายเท ความร้อนในหนึ่งมิติ.....	27

สารบัญญภาพ (ต่อ)

	หน้า
ภาพที่ 4.4 ผลของความเร็วในการหล่อต่อการเย็นตัวของบริเวณผิวและตรงกลาง ชิ้นงานที่คำนวณจากแบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ.....	29
ภาพที่ 4.5 ผลของความเร็วในการหล่อต่อช่วงการแข็งตัวของน้ำโลหะในการถ่ายเท ความร้อนในหนึ่งมิติ.....	30
ภาพที่ 4.6 ผลของความเร็วในการหล่อต่อช่วงการแข็งตัวของน้ำโลหะที่ตำแหน่งแกน กลางที่คำนวณ โดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ.....	32
ภาพที่ 4.7 ผลของอุณหภูมิในการเทต่อช่วงการแข็งตัวของน้ำโลหะในการถ่ายเท ความร้อนในหนึ่งมิติ.....	33
ภาพที่ 4.8 ผลของอุณหภูมิในการเทต่อช่วงการแข็งตัวของน้ำโลหะจากแบบจำลอง ที่คิดการถ่ายเทความร้อนในหนึ่งมิติ.....	34
ภาพที่ 4.9 ผลของอุณหภูมิในการเทต่อช่วงการแข็งตัวของน้ำโลหะ ที่ตำแหน่งแกนกลาง ที่คำนวณ โดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ.....	35
ภาพที่ 4.10 ผลของความเร็วในการหล่อต่อความหนาของเปลือกที่แข็งตัวที่ตำแหน่งต่างๆ ในแบบหล่อที่คำนวณ โดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ....	37
ภาพที่ 4.11 ผลของความเร็วในการหล่อในการเทต่อความหนาของเปลือกที่แข็งตัว ที่ตำแหน่งต่างๆในแบบหล่อคำนวณ โดยใช้แบบจำลองที่คิดการถ่ายเท ความร้อนในหนึ่งมิติ.....	38
ภาพที่ 4.12 ผลของอุณหภูมิในการตั้งในการเทต่อความหนาของเปลือกที่แข็งตัว ที่ระยะต่างๆ คำนวณ โดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ.....	39
ภาพที่ 4.13 ผลของอุณหภูมิในการเทต่อความหนาของเปลือกที่แข็งตัวที่ระยะต่างๆ คำนวณ โดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ.....	40
ภาพที่ 4.14 แสดงการกระจายตัวของอุณหภูมิที่ระยะต่างๆจากเมนิสคัส คำนวณ โดย ใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ.....	42
ภาพที่ 4.15 แสดงการกระจายตัวของอุณหภูมิในภาคตัดขวาง ที่ความเร็วในการหล่อ 0.65 m/min.....	43
ภาพที่ 4.16 แสดงการกระจายตัวของอุณหภูมิในภาคตัดขวาง ที่ความเร็วในการหล่อ 0.55 m/min.....	44
ภาพที่ 4.17 การกระจายตัวของอุณหภูมิในภาคตัดขวาง ที่ความเร็วในการหล่อ 0.75 m/min.....	45

สารบัญญภาพ (ต่อ)

หน้า

ภาพที่ 4.18	ผลของความเร็วในการหล่อต่อช่วงการแข็งตัวของน้ำโลหะที่ระยะต่างๆ จากเมนิสคัส คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ.....	48
ภาพที่ 4.19	ผลของความเร็วในการหล่อต่อช่วงการเย็นตัวของชิ้นงานที่ระยะต่างๆ จากเมนิสคัส คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ.....	49
ภาพที่ 4.20	ผลของอุณหภูมิเทต่อช่วงการแข็งตัวของน้ำโลหะที่ระยะต่างๆจากเมนิสคัส คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ.....	50
ภาพที่ 4.21	แสดงการกระจายตัวของอุณหภูมิของชิ้นงานหล่อแท่งเล็กที่ตำแหน่งต่างๆ จากเมนิสคัส.....	51
ภาพที่ 4.22	แสดงการกระจายตัวของอุณหภูมิของชิ้นงานหล่อแท่งอะลูมิเนียมที่ คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสามมิติ โดยใช้ โปรแกรม Micro AVS แสดงผล.....	52

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การหล่อแบบต่อเนื่องเป็นกระบวนการผลิตได้มีบทบาทสำคัญต่อภาคอุตสาหกรรมการผลิตในประเทศในช่วงทศวรรษที่ผ่านมา และมีแนวโน้มที่จะเพิ่มขึ้นเนื่องจากกระบวนการนี้เป็นกระบวนการที่ใช้พลังงานและวัตถุดิบได้คุ้มค่าที่สุด มีกำลังการผลิตที่สูง และสามารถผลิตผลิตภัณฑ์ที่มีคุณภาพและมีความสม่ำเสมอ แต่การควบคุมคุณภาพของผลิตภัณฑ์ในการหล่อแบบต่อเนื่องนั้นทำได้ยากเนื่องจากการถ่ายเทความร้อนที่แตกต่างกันในแต่ละส่วนของ กระบวนการ ซึ่งมีผลต่อการแข็งตัวของน้ำโลหะอันเป็นผลโดยตรงต่อกระบวนการและคุณภาพของผลิตภัณฑ์ ปัญหาเรื่องการถ่ายเทความร้อนส่งผลโดยตรงต่อกระบวนการผลิตและคุณภาพของผลิตภัณฑ์ เช่น การเกิดการร้าวของน้ำโลหะทำให้ต้องหยุดกระบวนการ การผิดรูปของผลิตภัณฑ์ การเกิดรอยแตกบริเวณผิว เป็นต้น ดังนั้นเพื่อที่จะปรับปรุงคุณภาพของกระบวนการ จึงมีความจำเป็นอย่างยิ่งที่จะต้องเข้าใจกระบวนการแข็งตัวของน้ำโลหะในกระบวนการ ซึ่งเกี่ยวข้องโดยตรงกับกระบวนการถ่ายเทความร้อนออกจากน้ำโลหะ

กลไกการถ่ายเทความร้อนเป็นกลไกที่ถือว่าสำคัญที่สุดในการหล่อแบบต่อเนื่อง เพราะส่งผลต่อการแข็งตัวของน้ำโลหะ ซึ่งเป็นตัวกำหนดโครงสร้างทางจุลภาคและคุณสมบัติทางกลของผลิตภัณฑ์ แต่การทดลองและศึกษาการถ่ายเทความร้อนโดยตรงในขณะที่แข็งตัวทำได้ยากเนื่องจากความซับซ้อนของกระบวนการ¹ ด้วยข้อจำกัดนี้จึงได้มีการนำแบบจำลองทางคณิตศาสตร์และระเบียบวิธีเชิงตัวเลขมาใช้ เพื่อช่วยทำนายปรากฏการณ์ต่างๆในกระบวนการ ประกอบกับในปัจจุบันความสามารถของคอมพิวเตอร์เพิ่มสูงขึ้นมากทำให้การคำนวณที่ซับซ้อนสามารถทำได้อย่างรวดเร็ว และสามารถแสดงผลการคำนวณในรูปแบบการนำเสนอที่เข้าใจได้ง่ายขึ้น ซึ่งสามารถนำข้อมูลเหล่านี้มาใช้ปรับปรุงกระบวนการผลิตให้ดียิ่งขึ้น อันยังประโยชน์ให้ได้ผลิตภัณฑ์ที่มีคุณภาพตามต้องการ

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อประดิษฐ์โปรแกรมจำลองการถ่ายเทความร้อนสำหรับการหล่อแบบต่อเนื่องโดยใช้วิธีการผลต่างสลับเนื่อง เพื่อศึกษาการแข็งตัวของน้ำโลหะในกระบวนการ

1.2.2 เพื่อศึกษาผลของพารามิเตอร์ต่างๆที่มีต่อการหล่อแบบต่อเนื่อง ได้แก่ ความเร็วในการหล่อ อุณหภูมิเท และการเย็นตัวทุติยภูมิ (secondary cooling)

1.3 ขอบเขตของการวิจัย

1.3.1 ประดิษฐ์โปรแกรมคอมพิวเตอร์โดยใช้ระเบียบวิธีผลต่างสี่เหลี่ยม เพื่อทำนายลักษณะการถ่ายเทความร้อนในกระบวนการหล่อแบบต่อเนื่อง

1.3.2 ปรับพารามิเตอร์ต่างๆ ของกระบวนการ เพื่อศึกษาผลกระทบจากพารามิเตอร์นั้นๆ ซึ่งได้แก่ ลักษณะการออกแบบกระบวนการ ลักษณะของแบบหล่อ ความเร็วในการหล่อ รูปแบบของการฉีดน้ำ เป็นต้น

1.3.3 เปรียบเทียบผลการคำนวณที่ได้กับการศึกษาอื่นๆก่อนหน้า เพื่อตรวจสอบความถูกต้องของแบบจำลอง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถนำโปรแกรมที่ประดิษฐ์ไปทำนายการถ่ายเทความร้อน ที่เกิดในกระบวนการเพื่อที่ได้ปรับปรุงกระบวนการให้ได้ผลิตภัณฑ์ตามที่ต้องการ

1.4.2 ทราบถึงผลของพารามิเตอร์ต่างๆ ของกระบวนการที่มีต่อผลิตภัณฑ์ เพื่อที่จะได้ปรับปรุงกระบวนการให้มีความเหมาะสม เพื่อให้ได้ผลิตภัณฑ์ตามที่ต้องการโดยกระบวนการให้ประสิทธิภาพสูงสุด

1.4.3 เป็นข้อมูลในการศึกษาและพัฒนาความรู้ด้านการจำลองปรากฏการณ์ในกระบวนการผลิตต่างๆ

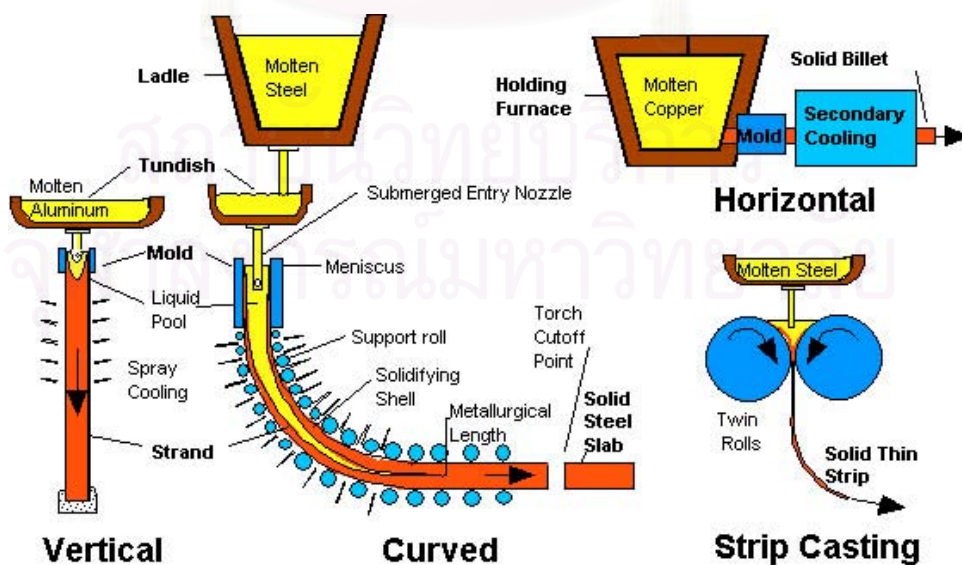
บทที่ 2

การศึกษาข้อมูลเบื้องต้น

2.1 แนวคิดและทฤษฎี

การหล่อแบบต่อเนื่องเป็นกระบวนการที่มีความสำคัญในอุตสาหกรรม โดยได้เข้ามาแทนที่กรรมวิธีการหล่อแบบแท่งขนาดใหญ่ (ingot) โดยเฉพาะการผลิตวัตถุบดที่สำเร็จรูป ข้อดีของการหล่อแบบต่อเนื่อง คือ สามารถผลิตผลิตภัณฑ์ได้โดยมีการสูญเสียโลหะ และพลังงานในการผลิตน้อยมากเมื่อเทียบกับการหล่อแบบแท่งขนาดใหญ่ โดยการผลิตแบบต่อเนื่องสามารถให้วัตถุดิบให้เกิดประโยชน์สูงสุดได้ถึง 97% เทียบกับการผลิตแบบแท่งขนาดใหญ่ที่ทำได้เพียงประมาณ 70% จากข้อมูลปริมาณการผลิตเหล็กในประเทศไทยมีการผลิตเหล็กประมาณ 2.1 ล้านตันในปี 2544 ซึ่งจะเห็นได้ว่าการหล่อแบบต่อเนื่องสามารถลดปริมาณของเสียได้เป็นจำนวนมาก ทั้งยังให้ประสิทธิภาพสูง ถึงแม้ว่าจะใช้การลงทุนสูง แต่ก็สามารถคืนทุนได้ในเวลาอันสั้น นอกจากนี้การหล่อแบบต่อเนื่องให้ผลิตภัณฑ์คุณภาพสูงและมีความสม่ำเสมอ

การหล่อแบบต่อเนื่องแตกต่างจากการหล่อประเภทอื่นตรงที่ การหล่อแบบต่อเนื่องดำเนินการในสถานะอยู่ตัว (steady-state) โดยโลหะหลอมเหลวเริ่มแข็งตัวเมื่อสัมผัสกับผนังแบบหล่อ (mold) ขณะที่ส่วนล่างที่แข็งตัวแล้วบางส่วนจะถูกดึงออกไปในอัตราที่รักษารอยต่อระหว่างของแข็งและของเหลวให้คงที่ที่ระยะต่างๆ โดยกระบวนการจะได้ผลดีที่สุดเมื่อภาวะต่างๆของกระบวนการดำเนินไปภายใต้สถานะอยู่ตัว



รูปที่ 2.1 การหล่อแบบต่อเนื่องประเภทต่างๆ

การหล่อแบบต่อเนื่องมีหลายประเภท ดังแสดงในรูปที่ 2.1 เครื่องหล่อในแนวตั้ง (vertical machine) มักจะใช้ในการหล่ออะลูมิเนียม และโลหะบางตัวในบางกรณี เครื่องหล่อแบบโค้ง (curved machine) ใช้เป็นหลักในการผลิตเหล็กกล้า เครื่องหล่อในแนวนอน (horizontal machine) ใช้ในการหล่อโลหะผสมนอกกลุ่มเหล็กและเหล็กกล้า โดยเครื่องหล่อประเภทนี้มีข้อเด่นตรงที่ไม่ต้องการพื้นที่ในแนวสูงมากนักจึงสามารถติดตั้งในอาคารเดี่ยวๆได้ และการหล่อแบบเป็นเส้น (strip casting) ซึ่งเริ่มนำมาใช้กับเหล็กกล้าและโลหะอื่นๆ กระบวนการนี้มีข้อดีคือ ผลิตภัณฑ์ที่ได้บาง ซึ่งทำให้สามารถลดปริมาณการรีดลงไปได้

2.2 พื้นฐานของการถ่ายเทความร้อน

การถ่ายเทความร้อนสามารถแบ่งออกได้เป็น 3 กลไกด้วยกันคือ การนำความร้อน การแผ่รังสีความร้อน และ การพาความร้อน

1) การนำความร้อน (Conduction)

เป็นปรากฏการณ์การส่งผ่านความร้อนจากการเคลื่อนที่ของตัวกลางที่อยู่นิ่ง หรือตัวกลางที่มีผิวสัมผัสกัน ซึ่งแสดงโดยสมการของฟูเรียร์

$$q_x = -k \left(\frac{dT}{dx} \right) \quad (2.1)$$

เมื่อ q_x = ฟลักซ์ความร้อนในทิศทาง x

$\left(\frac{dT}{dx} \right)$ = ความแตกต่างระหว่างอุณหภูมิ ของ 2 บริเวณในทิศทาง x

k = ค่าการนำความร้อนของวัสดุ

2) การแผ่รังสีความร้อนของวัสดุ (Radiation)

เป็นการปลดปล่อยพลังงานในรูปของคลื่นแม่เหล็กไฟฟ้าผ่านตัวกลางหรือบรรยากาศ รูปแบบของสมการคือ

$$e = \varepsilon \sigma T^4 \quad (2.2)$$

เมื่อ e = ฟลักซ์ของการแผ่รังสี

σ = ค่าคงที่มีค่าเท่ากับ $5.67 \times 10^{-8} \text{ W m}^{-2} \text{ K}^{-4}$

ε = ค่าสัมประสิทธิ์การแผ่รังสีความร้อน

T = อุณหภูมิสัมบูรณ์ของวัสดุที่แผ่รังสี (เป็นตัวบอกประสิทธิภาพของพื้นผิว ใดๆ ที่จะปล่อยพลังงานออกมาเทียบกับวัสดุในอุดมคติ มีค่าระหว่าง 0 ถึง 1)

3) การพาความร้อน(Convection)

เป็นการถ่ายเทความร้อนจากพื้นผิวของวัสดุผ่านไปยังของไหลหรือตัวกลางที่มีการเคลื่อนที่รอบพื้นที่ผิวดังกล่าว เนื่องจากการวัดสภาพการพาความร้อนไม่สามารถวัดได้โดยตรง จึงแสดงอยู่ในรูปของพลังงานความร้อนที่ของแข็งส่งผ่านไปยังของไหลต่อหน่วยพื้นที่ ดังแสดงโดยสมการ

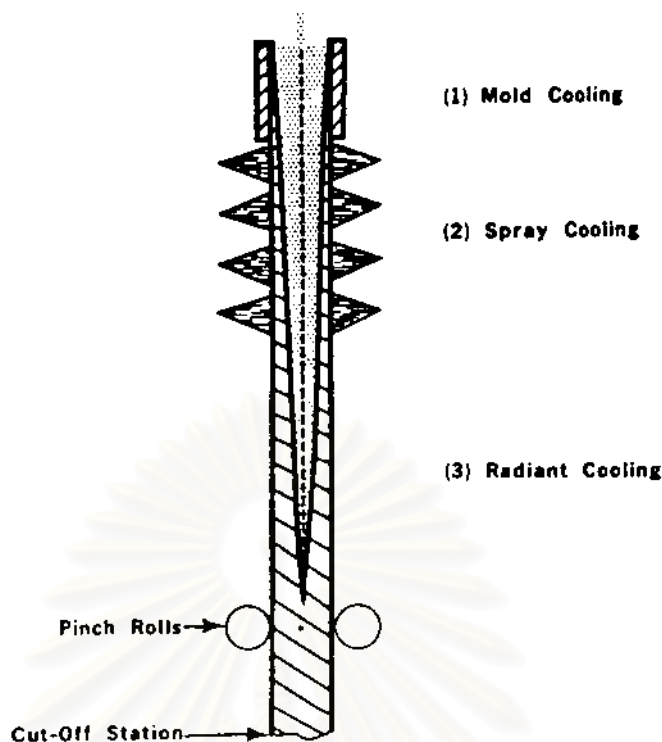
$$q_s = h(T_s - T_f) \quad (2.3)$$

เมื่อ q_s = พลักซ์ความร้อนต่อผืนผิว
 h = ค่าสัมประสิทธิ์ของการถ่ายเทความร้อน
 T_s, T_f = อุณหภูมิของผิววัสดุและของไหล ตามลำดับ

ในทางปฏิบัติการถ่ายเทความร้อนทั้ง 3 รูปแบบมักเกิดขึ้นพร้อมๆกัน การแก้ปัญหาจะใช้สมการสมดุลทางความร้อน (Heat balance) จากกฎข้อที่หนึ่งของเทอร์โมไดนามิกส์ โดยพิจารณาระบบที่มีปริมาตรที่แน่นอน และความร้อนสามารถไหลผ่านเข้ามาได้ ที่เวลาขณะใดๆ “อัตราการถ่ายเทพลังงานทางความร้อนและ/หรือทางกลที่เข้าสู่ระบบรวมกับอัตราการถ่ายเทพลังงานที่เกิด ขึ้นภายในระบบลดด้วยอัตราการถ่ายเทพลังงานทางความร้อนและ/หรือทางกลที่ไหลออกจากระบบนั้นจะมีค่าเท่ากับอัตราการเพิ่มขึ้นของพลังงานที่สะสมอยู่ในระบบนั้น”²

2.3 การถ่ายเทความร้อนในการหล่อแบบต่อเนื่อง

การหล่อแบบต่อเนื่องเป็นกระบวนการขึ้นรูปชิ้นงานโดยการดึงเอาความร้อนออกจากโลหะหลอมเหลวจนกระทั่งกลายเป็นของแข็ง ซึ่งความร้อนที่ดึงออกมาประกอบด้วยความร้อนส่วนที่สูงกว่าจุดหลอมเหลวที่เรียกว่า super heat ความร้อนที่คายออกจากการแข็งตัวของน้ำโลหะที่เรียกว่า latent heat และความร้อนที่ดึงออกตั้งแต่จากจุดแข็งตัวถึงอุณหภูมิห้องที่เรียกว่า sensible heat โดยอาศัยกระบวนการถ่ายเทความร้อนแบบต่างๆ คือ การพาความร้อน (convection) ที่เกิดในส่วนของน้ำโลหะ ที่มาจากการไหลเนื่องจากโมเมนตัมที่ไหลลงมาจากทูนดิช (tundish) และที่เกิดจากการไหลโดยแรงลอยตัวของน้ำโลหะเอง การนำความร้อนที่เกิดในโลหะที่แข็งตัวจากด้านที่ร้อนไปยังผิวด้านนอกที่เย็นกว่า และการถ่ายเทความร้อนออกไปสู่ภายนอกโดยอาศัยทั้งการนำความร้อน การพาความร้อนและการแผ่รังสีความร้อน ซึ่งจะเกิดขึ้นแตกต่างกันไปในแต่ละบริเวณของกระบวนการ ได้แก่ บริเวณแบบหล่อ (mold) บริเวณน้ำฉีด (spray) บริเวณการแผ่รังสี (radiant cooling) ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แสดงบริเวณต่างๆ ของการหล่อแบบต่อเนื่อง

การถ่ายเทความร้อนเป็นปรากฏการณ์ที่สำคัญในการหล่อแบบต่อเนื่อง โดยจะเป็นตัวกำหนดความสามารถในการผลิต ถ้าการถ่ายเทความร้อนออกมาทำได้มาก ความลึกของบ่อน้ำโลหะจะต่ำทำให้สามารถตัดชิ้นงานได้เร็วขึ้นทำให้ผลิตได้มาก นอกจากนี้ที่สำคัญการถ่ายเทความร้อนยังส่งผลต่อคุณภาพของผลิตภัณฑ์ด้วย โดยเฉพาะผลต่อการเกิดรอยแตกที่บริเวณผิวและด้านในผลิตภัณฑ์ซึ่งเกิดจากการขยายตัวเนื่องจากความร้อนที่ต่างกันในแต่ละบริเวณที่ทำให้เกิดความเครียด และยังส่งผลต่อการโก่งของผลิตภัณฑ์ด้วย จากที่กล่าวมาแล้วจึงสรุปได้ว่าการหล่อแบบต่อเนื่องนั้นต้องไม่พิจารณาที่อัตราการดึงความร้อนออกแต่เพียงอย่างเดียว แต่ต้องคำนึงถึงโครงสร้างที่เกิดจากการแข็งตัวด้วย

2.4 ระเบียบวิธีการในการแก้ปัญหาทางความร้อนในกระบวนการหล่อแบบต่อเนื่อง

ในช่วงหลายปีที่ผ่านมาคอมพิวเตอร์ได้มีบทบาทที่สำคัญในการศึกษาเชิงตัวของน้ำโลหะโดยการสร้างแบบจำลองทางคณิตศาสตร์เพื่อทำนายปรากฏการณ์ต่างๆที่เกิดขึ้นในกระบวนการทำให้สามารถเข้าใจในกระบวนการ สามารถผลิตผลิตภัณฑ์ที่มีคุณภาพสูงและทั้งยังประหยัดเวลาและลดต้นทุนในการผลิต คอมพิวเตอร์ในปัจจุบันมีความเร็วในการประมวลผลสูงมาก มีหน่วย

ความจำสูงและราคาถูก ทั้งยังมีความหลากหลายในการใช้งาน ซึ่งเหนืออำนาจต่อการแก้ปัญหาเรื่องการถ่ายเทความร้อนในกระบวนการได้

มีผู้เสนอระเบียบวิธีการในการแก้ปัญหาทางความร้อนในกระบวนการหล่อแบบต่อเนื่องไว้หลายวิธี เช่น J.K. Brimacombe^{3,4} ใช้วิธีแก้ปัญหาโดยการใส่ระบบสมการการถ่ายเทความร้อนใน 2 มิติและระเบียบวิธีผลต่างสี่เหลี่ยม (finite difference) และแก้สมการโดยใช้ระเบียบวิธีแบบชัดแจ้ง (explicit method), E.A. Mizikar⁵ ใช้ระบบสมการการถ่ายเทความร้อนในและระเบียบวิธีผลต่างสี่เหลี่ยม 1 มิติ แก้สมการโดยใช้ระเบียบวิธีแบบชัดแจ้ง และได้เสนอวิธีการคิดความร้อนแฝงที่เกิดจากการแข็งตัว (latent heat of freezing) เข้ามาคำนวณด้วยและได้ทำการเปรียบเทียบผลกับการแก้ปัญหาโดยใช้วิธีการวิเคราะห์ (analytical method) ที่ทำโดย J. Savage และ W.H. Pritchard⁶ ซึ่งระเบียบวิธีการต่างๆเหล่านี้ใช้หลักการไม่คิดความร้อนในแนวการตั้ง ซึ่งให้ผลสอดคล้องกับการหล่อแบบต่อเนื่องพวกเหล็กกล้า เนื่องจากการนำความร้อนของเปลือกของแข็งต่ำและมีปริมาณความร้อนสูง ทำให้ขอบของเหลวมีความยาวมากจนสามารถไม่คิดผลความร้อนในแนวการตั้งได้ แต่ในกรณีการหล่อแบบต่อเนื่องของโลหะนอกกลุ่มเหล็กที่มีคุณสมบัติการนำความร้อนที่ดี เช่น ทองแดง และอลูมิเนียมไม่สามารถนำมาใช้ได้ จึงมีผู้เสนอวิธีการต่างๆเพื่อใช้กับการหล่อโลหะชนิดอื่นๆ เช่น Chingping Tong และ Kenji Yokomizo⁷ ได้ใช้วิธีการคิดการถ่ายเทความร้อนใน 3 มิติและแก้สมการโดยใช้ระเบียบวิธีผลต่างสี่เหลี่ยมและระเบียบวิธีแบบปริยาย (implicit method) ซึ่งให้ผลที่สอดคล้องกับการทดลองในระดับที่น่าพอใจ แต่การแก้ปัญหาโดยใช้วิธีแบบปริยายนั้นมีข้อจำกัด ตรงความละเอียดของการแบ่งช่วง เพราะถ้าแบ่งละเอียดมากจะใช้เวลาในการคำนวณนานและใช้หน่วยความจำมาก⁷ ซึ่งในช่วงนั้นคอมพิวเตอร์ยังไม่มีประสิทธิภาพใกล้เคียงกับปัจจุบัน จึงไม่สามารถจำลองปรากฏการณ์ที่ยาวมากนักได้ดีในทางปฏิบัติ

บทที่ 3

วิธีดำเนินการวิจัย

การวิจัยนี้ได้ประดิษฐ์โปรแกรมคอมพิวเตอร์ โดยใช้แบบจำลองการถ่ายเทความร้อนในการหล่อแบบต่อเนื่องแบบต่างๆ โดยใช้วิธีการผลต่างสี่เหลี่ยม (finite difference method) เพื่อให้ทราบถึงอุณหภูมิที่บริเวณต่างๆ ในกระบวนการ และทำการกำหนดค่าต่างๆ ให้กับตัวแปรและพารามิเตอร์ของกระบวนการ ให้สอดคล้องกับงานวิจัยอื่นที่ได้ยืนยันกับผลการทดลองจริง จากนั้นจึงนำผลที่ได้มาเปรียบเทียบกัน เพื่อตรวจสอบความถูกต้องของโปรแกรมที่ประดิษฐ์ เมื่อปรับแก้ไขโปรแกรมให้มีความถูกต้องแม่นยำแล้ว จึงนำโปรแกรมที่ประดิษฐ์มาทำการคำนวณโดยปรับพารามิเตอร์ต่างๆ ของกระบวนการ ได้แก่ ความเร็วในการหล่อ, อุณหภูมิเท, ชนิดของวัสดุที่นำมาหล่อ, การเย็นตัวในช่วงทุติยภูมิ (secondary cooling)

3.1 สมการพื้นฐานของการหล่อแบบต่อเนื่อง

สมการอนุพันธ์แสดงความสัมพันธ์ของอุณหภูมิที่ตำแหน่งต่างๆ ในกระบวนการ สามารถเขียนในรูปทั่วไปได้ดังนี้

$$\rho C_p \left\{ \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right\} = k \nabla^2 T - v \rho C_p \frac{\partial T}{\partial z} \quad (3.1)$$

โดยที่ \mathbf{u} คือ ความเร็วในการไหลของน้ำโลหะ

v คือ ความเร็วในการดึงขึ้นงาน

ซึ่งถ้าไม่คิดพจน์ของการไหลของน้ำโลหะ จะได้สมการเป็น

$$\rho C_p \frac{\partial T}{\partial t} = k \nabla^2 T - v \rho C_p \frac{\partial T}{\partial z} \quad (3.2)$$

3.2 โปรแกรมแสดงอุณหภูมิที่ตำแหน่งต่างๆ แบบสองมิติในสถานะอยู่ตัวที่พัฒนาจากแบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ เพื่อใช้ในการหล่อแบบต่อเนื่องแบบแท่งแบนสำหรับเหล็กกล้า

3.2.1 สมการเบื้องต้น

เมื่อพิจารณากระบวนการโดยให้ผู้สังเกตอยู่กับที่ จะเห็นได้ว่ากระบวนการอยู่ในสถานะอยู่ตัว สมการที่ 3.2 จึงจัดใหม่ได้ดังสมการที่ 3.3

$$0 = k\nabla^2 T - \rho C_p \frac{\partial T}{\partial z} \quad (3.3)$$

และจากรูปที่ 3.1 หากพิจารณาโดยแบ่งน้ำโลหะในกระบวนการออกเป็นชั้นเล็กๆ ให้ผู้สังเกตเคลื่อนที่ไปกับชั้นของน้ำโลหะ โดยที่เวลาที่ $t = 0$ ชั้นของน้ำโลหะที่พิจารณาจะอยู่ที่ระยะ $z = 0$ ความสัมพันธ์ระหว่าง z และ t สามารถเขียนในรูป

$$z = vt \quad (3.4)$$

ซึ่งจะได้ว่า

$$dz = vdt \quad (3.5)$$

โดยที่ v คือ ความเร็วในการดึง

ซึ่งเมื่อนำสมการที่ 3.5 ลงไปแทนในสมการที่ 3.3 จะได้

$$\rho C_p \frac{\partial T}{\partial t} = k\nabla^2 T = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right) \quad (3.6)$$

ซึ่งในกรณีของเหล็กกล้า ผลจากการนำความร้อนที่ต่ำและบ่อน้ำโลหะที่ลึกมาก ทำให้สามารถไม่คิดพจน์การนำความร้อนในทิศทางกรตั้งได้ และนอกจากนี้เนื่องจากรูปร่างของงานที่หล่อเป็นแท่งแบนซึ่งมีด้านยาวยาวกว่าด้านกว้างมาก ผลของการนำความร้อนที่ต่ำทำให้สามารถไม่คิดพจน์การนำความร้อนในแนวตามยาวได้อีกพจน์หนึ่งด้วย สมการที่ 3.5 จึงลดรูปได้ดังสมการที่ 3.6 ซึ่งมีรูปของสมการเหมือนกับ สมการการถ่ายเทความร้อนภายใต้สถานะชั่วคราวในหนึ่งมิติ (one-dimensional transient heat transfer equation)

$$\rho C_p \frac{\partial T}{\partial t} = k\left(\frac{\partial^2 T}{\partial x^2}\right) \quad (3.7)$$

ในการที่จะแก้สมการที่ 3.7 นี้ซึ่งอยู่ในรูปของการกระจายตัวของอุณหภูมิที่เปลี่ยนแปลงตามเวลา จะต้องมีการกำหนดเงื่อนไขตอนเริ่มต้น (initial condition) และเงื่อนไขขอบเขต (boundary condition) โดยสำหรับเงื่อนไขตอนเริ่มต้น ที่เวลา $t = 0$ กำหนดให้อุณหภูมิของชั้นของน้ำโลหะที่บริเวณเมนัสคัส (meniscus) มีค่าเท่ากับอุณหภูมิเท (T_p)

$$t = 0, \quad 0 < x < X, \quad T(x) = T_p \quad (3.8)$$

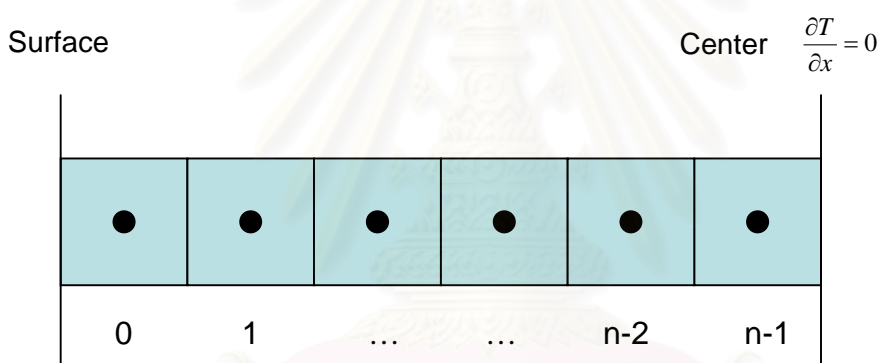
สำหรับการกำหนดเงื่อนไขขอบเขตที่บริเวณตรงกลางชิ้นงาน ถือว่าไม่มีการถ่ายเทความร้อน

$$x = X/2, \quad \frac{\partial T}{\partial x} = 0 \quad (3.9)$$

สำหรับการกำหนดเงื่อนไขขอบเขตที่ผิวด้านนอก กำหนดโดยสมการ

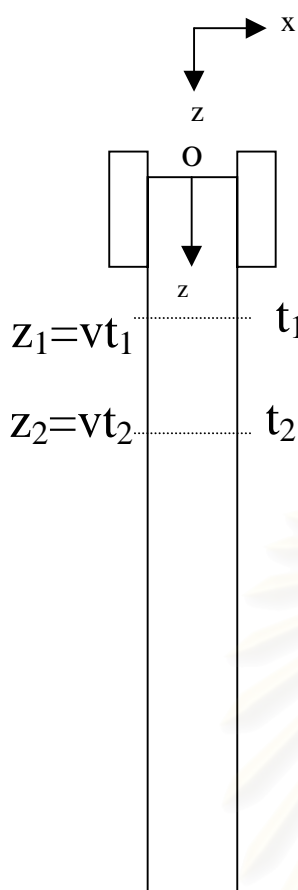
$$x = 0, \quad -k \left(\frac{dT}{dx} \right) \Big|_{x=0} = q_0 \quad (3.10)$$

เมื่อ q_0 คือ ปริมาณความร้อนที่ผิว ซึ่งแตกต่างกันไปในแต่ละส่วนของกระบวนการ



รูปที่ 3.1 แสดงการแบ่งชั้นของน้ำโลหะออกเป็นเอลิเมนต์ย่อย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.2 แสดงทิศทางและรูปร่างของกระบวนการ

3.2.2 การดุลความร้อน (heat balance)

ชั้นของน้ำโลหะจะถูกแบ่งย่อยออกเป็นส่วนเล็กๆ เรียกว่า เอลิเมนต์ (element) ดังในรูปที่ 3.1 โดยถือว่าภายในเอลิเมนต์เดียวกันจะมีอุณหภูมิและคุณสมบัติทางกายภาพของความร้อน (thermo-physical property) ที่เท่ากันทุกบริเวณ เมื่อพิจารณาแต่ละเอลิเมนต์ดังในรูปที่ 3.3 (b) การดุลความร้อนทำได้โดยพิจารณาปริมาณความร้อนที่ไหลเข้าเอลิเมนต์และออกจากเอลิเมนต์ในแต่ละทิศทาง ซึ่งในกรณีนี้มีเพียงทิศทางเดียวคือในแนวแกน x

ตัวอย่างการดุลความร้อน โดยพิจารณาว่าความร้อนที่ไหลเข้าเอลิเมนต์มาจากการนำความร้อนจากเอลิเมนต์ข้างเคียง จากรูปที่ 3.3 (b) พลังงานความร้อนที่ไหลเข้าสู่ระบบได้แก่

$$q_1 = kA \frac{(T_{i+1,j} - T_{i,j})}{\Delta x}$$

$$q_2 = kA \frac{(T_{i-1,j} - T_{i,j})}{\Delta x}$$

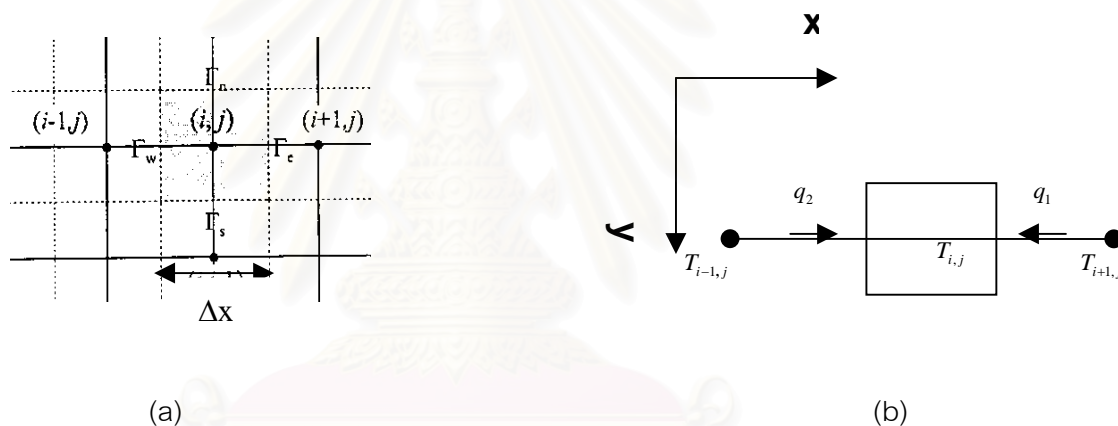
ดังนั้นจะได้ว่า $\rho C_p V \frac{\Delta T}{\Delta t} = q_1 + q_2$

และกำหนดให้ $\alpha = \frac{k}{\rho C_p}$, $\Delta T = T_{i,j}^{t+\Delta t} - T_{i,j}^t$

เมื่อจัดสมการจะได้ว่า

$$T_{i,j}^{t+\Delta t} = \left(1 - \frac{\alpha \Delta t}{(\Delta x)^2} - \frac{\alpha \Delta t}{(\Delta x)^2}\right) T_{i,j}^t + \frac{\alpha \Delta t}{(\Delta x)^2} T_{i+1,j}^t + \frac{\alpha \Delta t}{(\Delta x)^2} T_{i-1,j}^t \quad (3.11)$$

จากสมการนี้จะเห็นได้ว่า เราสามารถรู้ค่าของอุณหภูมิที่จุดต่างๆ ณ เวลา $t + \Delta t$ ได้ถ้าเรารู้อุณหภูมิที่จุดนั้นและจุดข้างเคียง ณ เวลา t ซึ่งจากเงื่อนไขตอนเริ่มต้นที่กำหนด ทำให้เราสามารถทราบอุณหภูมิที่จุดต่างๆ ของทั้งระบบได้



รูปที่ 3.3 แสดงการถ่ายเทความร้อนแบบหนึ่งมิติ

3.3 โปรแกรมแสดงอุณหภูมิที่ตำแหน่งต่างๆ แบบสามมิติในสภาวะอยู่ตัวที่พัฒนาจากแบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ เพื่อใช้ในการหล่อแบบต่อเนื่องแบบแท่งสำหรับเหล็กกล้า

3.3.1 สมการเบื้องต้น

เนื่องจากกระบวนการอยู่ในสภาวะอยู่ตัว และในกรณีของเหล็กกล้า ผลจากการนำความร้อนที่ต่ำและบ่อน้ำโลหะที่ลึกมาก ทำให้สามารถไม่คิดพจน์การนำความร้อนในทิศทางกรตั้งได้ จากสมการพื้นฐานที่ 3.2 และสมการที่ 3.5 สามารถจัดสมการใหม่ได้ดังสมการที่ 3.12 โดยใช้วิธีคิดดังที่ได้กล่าวไว้แล้วในหัวข้อที่ 3.2.1

$$\rho C_p \frac{\partial T}{\partial t} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \quad (3.12)$$

ซึ่งรูปของสมการเหมือนกับ สมการการถ่ายเทความร้อนภายใต้สถานะชั่วครู่ในสองมิติ (two-dimensional transient heat transfer equation)

สำหรับเงื่อนไขตอนเริ่มต้น ที่เวลา $t = 0$ กำหนดให้อุณหภูมิของชั้นของน้ำโลหะที่บริเวณเมนัสคัส (meniscus) มีค่าเท่ากับอุณหภูมิเท (T_p)

$$t = 0, \quad 0 < x < X, \quad 0 < y < Y, \quad T(x, y) = T_p \quad (3.13)$$

สำหรับการกำหนดเงื่อนไขขอบเขตที่บริเวณแกนสมมาตร ถือว่าไม่มีการถ่ายเทความร้อน

$$x = X/2, \quad \frac{\partial T}{\partial x} = 0 \quad (3.14)$$

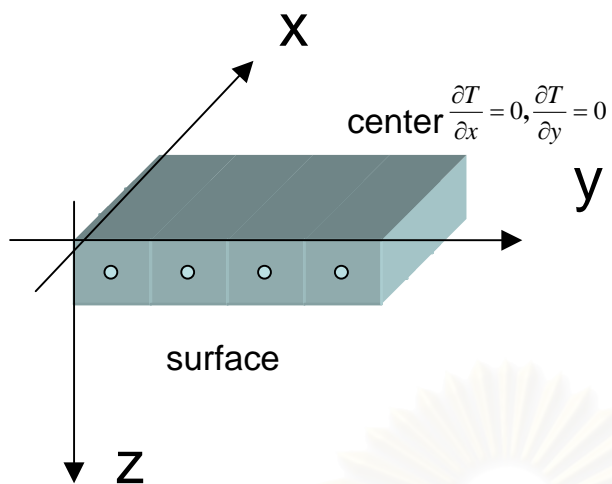
$$y = Y/2, \quad \frac{\partial T}{\partial y} = 0 \quad (3.15)$$

สำหรับการกำหนดเงื่อนไขขอบเขตที่ผิวด้านนอก กำหนดโดยสมการ

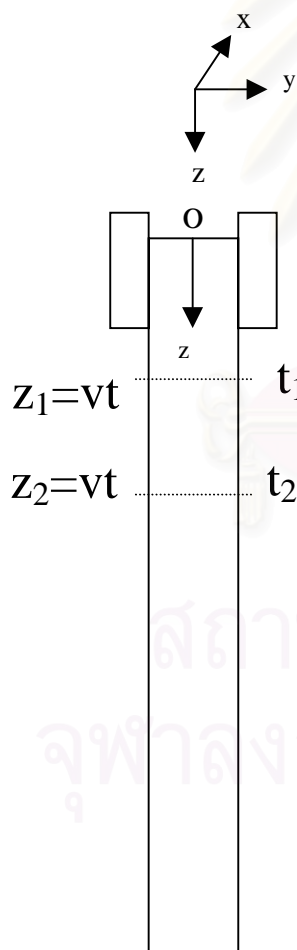
$$x = 0, \quad -k \left(\frac{dT}{dx} \right) \Big|_{x=0} = q_0 \quad (3.16)$$

$$y = 0, \quad -k \left(\frac{dT}{dy} \right) \Big|_{y=0} = q_0 \quad (3.17)$$

เมื่อ q_0 คือ ปริมาณความร้อนที่ผิว ซึ่งแตกต่างกันไปในแต่ละส่วนของกระบวนการ



รูปที่ 3.4 แสดงการแบ่งชั้นของน้ำโลหะออกเป็นเอลิเมนต์ย่อย



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.5 แสดงทิศทางและรูปร่างของกระบวนการ

3.3.2 การดูลความร้อน

การดูลความร้อนทำได้โดยพิจารณาปริมาณความร้อนที่ไหลเข้าเอลิเมนต์และออกจากเอลิเมนต์ในแต่ละทิศทาง ซึ่งในกรณีนี้มี 2 ทิศทางคือในแนวแกน x และแนวแกน y

ตัวอย่างการดูลความร้อน โดยพิจารณาว่าความร้อนที่ไหลเข้าเอลิเมนต์มาจากการนำความร้อนจากเอลิเมนต์ข้างเคียง จากรูปที่ 3.6 พลังงานความร้อนที่ไหลเข้าสู่ระบบได้แก่

$$q_1 = kA \frac{(T_{i+1,j} - T_{i,j})}{\Delta x} \quad q_2 = kA \frac{(T_{i-1,j} - T_{i,j})}{\Delta x}$$

$$q_3 = kA \frac{(T_{i,j+1} - T_{i,j})}{\Delta y} \quad q_4 = kA \frac{(T_{i,j-1} - T_{i,j})}{\Delta y}$$

ดังนั้นจะได้ว่า $\rho C_p V \frac{\Delta T}{\Delta t} = q_1 + q_2 + q_3 + q_4$

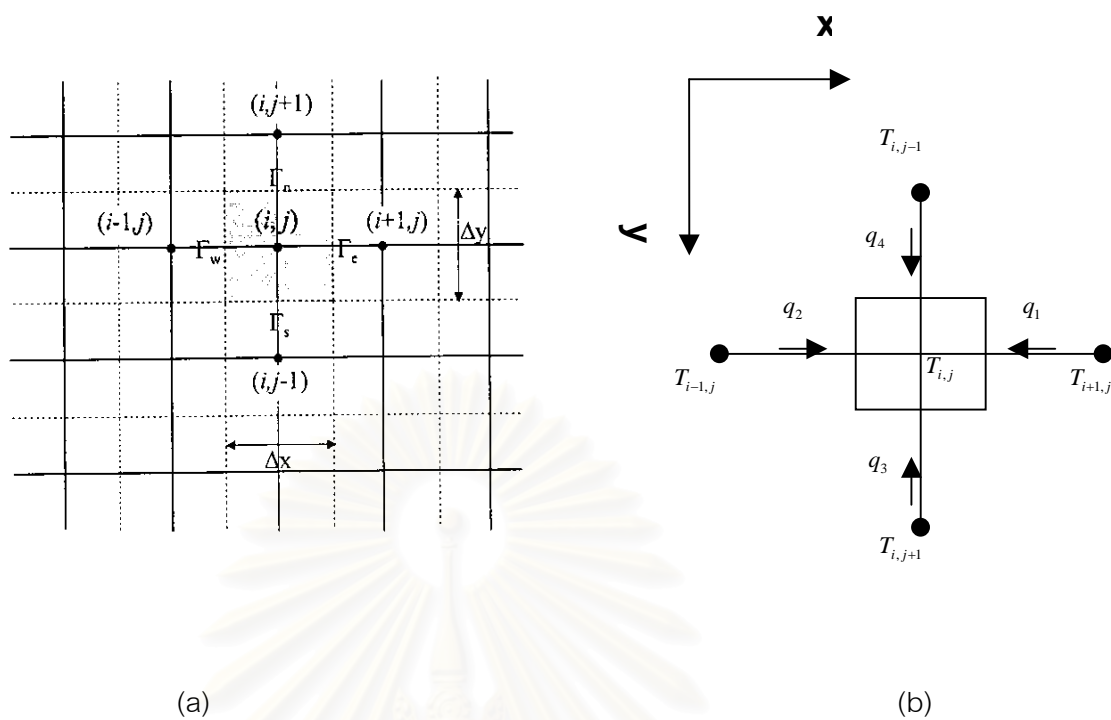
และกำหนดให้ $\alpha = \frac{k}{\rho C_p}$, $\Delta T = T_{i,j}^{t+\Delta t} - T_{i,j}^t$

เมื่อจัดสมการใหม่จะได้ว่า

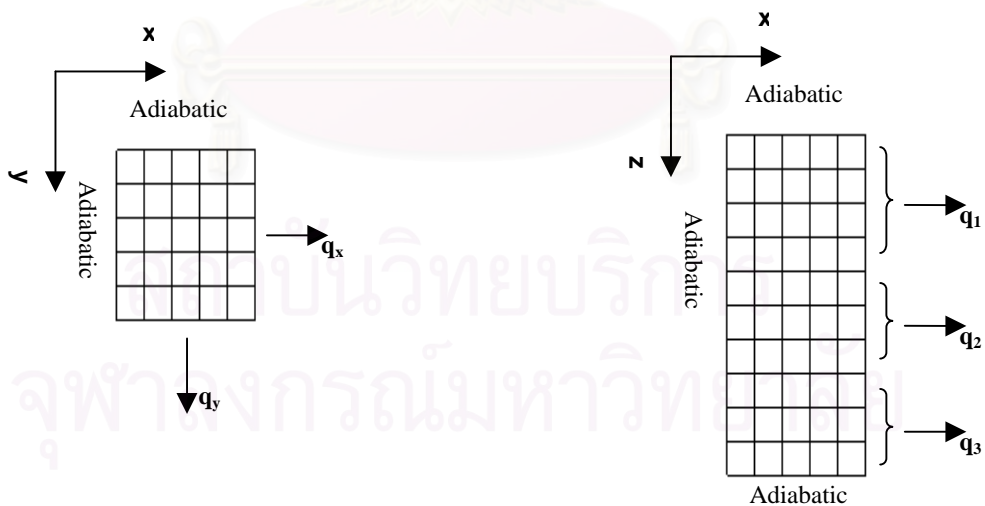
$$T_{i,j}^{t+\Delta t} = \left(1 - \frac{\alpha \Delta t}{(\Delta x)^2} - \frac{\alpha \Delta t}{(\Delta x)^2} - \frac{\alpha \Delta t}{(\Delta y)^2} - \frac{\alpha \Delta t}{(\Delta y)^2} \right) T_{i,j}^t \quad (3.18)$$

$$+ \frac{\alpha \Delta t}{(\Delta x)^2} T_{i+1,j}^t + \frac{\alpha \Delta t}{(\Delta x)^2} T_{i-1,j}^t + \frac{\alpha \Delta t}{(\Delta y)^2} T_{i,j+1}^t + \frac{\alpha \Delta t}{(\Delta y)^2} T_{i,j-1}^t$$

จากสมการนี้จะเห็นได้ว่า เราสามารถรู้ค่าของอุณหภูมิที่จุดต่างๆ ณ เวลา $t + \Delta t$ ได้ๆ ถ้าเรารู้อุณหภูมิที่จุดนั้นและจุดข้างเคียง ณ เวลา t ซึ่งจากเงื่อนไขตอนเริ่มต้นที่กำหนด ทำให้เราสามารถทราบอุณหภูมิที่จุดต่างๆของทั้งระบบได้



รูปที่ 3.6 แสดงการถ่ายเทความร้อนแบบสองมิติ

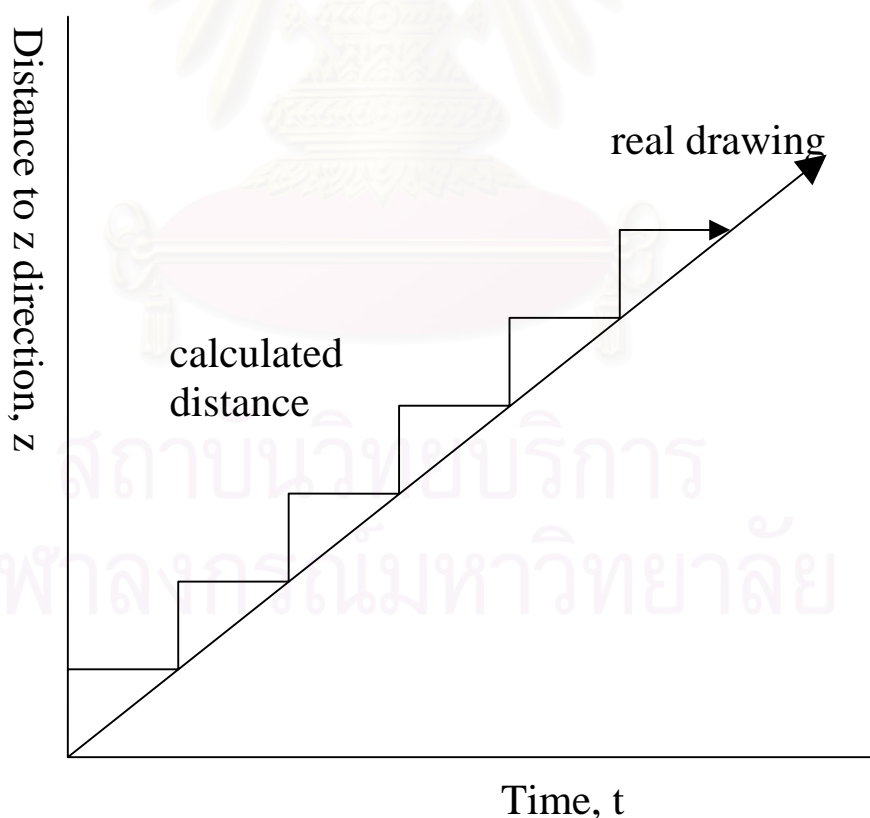


รูปที่ 3.7 แสดงเงื่อนไขขอบเขต

3.4 โปรแกรมแสดงอุณหภูมิที่ตำแหน่งต่างๆแบบสามมิติในสภาวะชั่วคราว ที่พัฒนาจากแบบจำลองที่คิดการถ่ายเทความร้อนในสามมิติ เพื่อใช้ในการหล่อแบบต่อเนื่องสำหรับโลหะนอกกลุ่มเหล็กและเหล็กกล้าในช่วงเริ่มต้นของกระบวนการ

3.4.1 สมการเบื้องต้น

ในการหล่อโลหะนอกกลุ่มเหล็กเช่น อะลูมิเนียม ทองแดง การถ่ายเทความร้อนในแนวตั้งไม่สามารถตัดทิ้งได้เนื่องจากวัสดุเหล่านี้มีการนำความร้อนที่ดี โปรแกรมนี้จึงใช้การคิดการถ่ายเทความร้อนในสภาวะชั่วคราว โดยหลักการคิดจะในการคำนวณในสภาวะไม่คงตัวนั้นจะใช้สมการการถ่ายเทความร้อนที่โดยจะไม่ใช้พจน์ของความเร็วในการตั้ง โดยการคำนวณจะถือว่า ณ เวลาที่คำนวณ รูปร่างของชิ้นงานจะคงที่อยู่ชั่วขณะหนึ่ง ซึ่งถือว่าอยู่ในสภาวะไม่คงตัว โดยการคำนวณจะแบ่งออกเป็นช่วงๆดังแสดงในรูปที่ 3.8



รูปที่ 3.8 แสดงแนวความคิดรูปที่โดยใช้ระเบียบวิธีความแตกต่างและทิศทางการถ่ายเทความร้อนในระนาบสามมิติ

โดยสมการในการคำนวณคือ

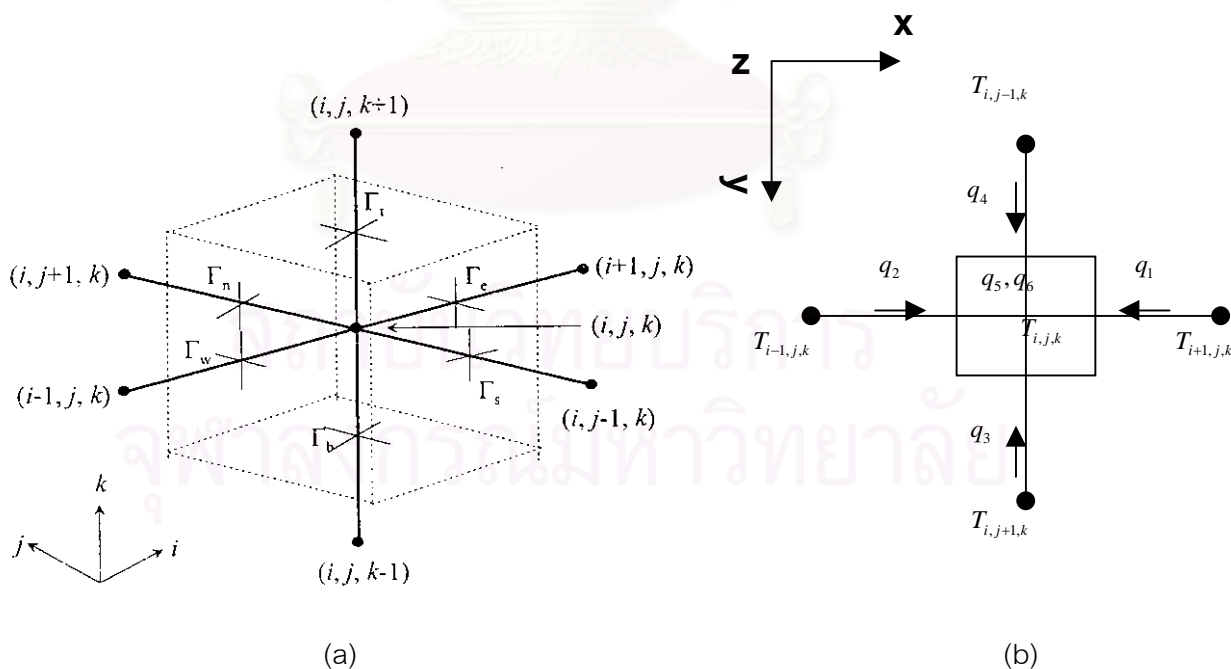
$$\rho C_p \left(\frac{\partial T}{\partial t} \right) = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \quad (3.19)$$

3.4.2 การดุลความร้อน

ทำการแบ่งย่อยชิ้นงานออกเป็นส่วนเล็กๆ และกำหนดจุดโดยใช้วิธีการผลต่างสี่เหลี่ยม ดังแสดงในรูปที่ 3.9 (a) เมื่อพิจารณาชิ้นงานที่ถูกแบ่งย่อยแต่ละพบว่ามี การถ่ายเทความร้อนสาม ทางในแนวแกน x y และ z ดังแสดงดังรูปที่ 3.9 (b)

ความร้อนที่ไหลเข้าในระบบทั้งหมดคือ

$$\begin{aligned} q_1 &= kA \frac{(T_{i+1,j,k} - T_{i,j,k})}{\Delta x} & q_2 &= kA \frac{(T_{i-1,j,k} - T_{i,j,k})}{\Delta x} \\ q_3 &= kA \frac{(T_{i,j,k+1} - T_{i,j,k})}{\Delta z} & q_4 &= kA \frac{(T_{i,j,k-1} - T_{i,j,k})}{\Delta z} \\ q_5 &= kA \frac{(T_{i,j,k+1} - T_{i,j,k})}{\Delta z} & q_6 &= kA \frac{(T_{i,j,k-1} - T_{i,j,k})}{\Delta z} \end{aligned}$$



รูปที่ 3.9 แสดงการแบ่งรูปโดยใช้ระเบียบวิธีความแตกต่างและทิศทางการถ่ายเทความร้อนในระนาบสามมิติ

สมการการถ่ายเทความร้อนคือ

$$\rho C_p V \frac{\Delta T}{\Delta t} = q_1 + q_2 + q_3 + q_4 + q_5 + q_6$$

และ

$$\alpha = \frac{k}{\rho C_p}$$

ดังนั้นจึงสามารถจัดอยู่ในรูปของ

$$T_{i,j,k}^{t+\Delta t} = \left(1 - \frac{\alpha \Delta t}{(\Delta x)^2} - \frac{\alpha \Delta t}{(\Delta x)^2} - \frac{\alpha \Delta t}{(\Delta y)^2} - \frac{\alpha \Delta t}{(\Delta y)^2} - \frac{\alpha \Delta t}{(\Delta z)^2} - \frac{\alpha \Delta t}{(\Delta z)^2} \right) T_{i,j,k}^t \quad (3.20)$$

$$+ \frac{\alpha \Delta t}{(\Delta x)^2} T_{i+1,j,k}^t + \frac{\alpha \Delta t}{(\Delta x)^2} T_{i-1,j,k}^t + \frac{\alpha \Delta t}{(\Delta y)^2} T_{i,j+1,k}^t + \frac{\alpha \Delta t}{(\Delta y)^2} T_{i,j-1,k}^t + \frac{\alpha \Delta t}{(\Delta z)^2} T_{i,j,k+1}^t + \frac{\alpha \Delta t}{(\Delta z)^2} T_{i,j,k-1}^t$$

ซึ่งสมการนี้เป็นผลเฉลยของสมการสามารถนำไปหาค่าการกระจายตัวของอุณหภูมิที่เวลาต่างๆได้ เมื่อทราบอุณหภูมิในการเหนี่ยวนำโลหะ

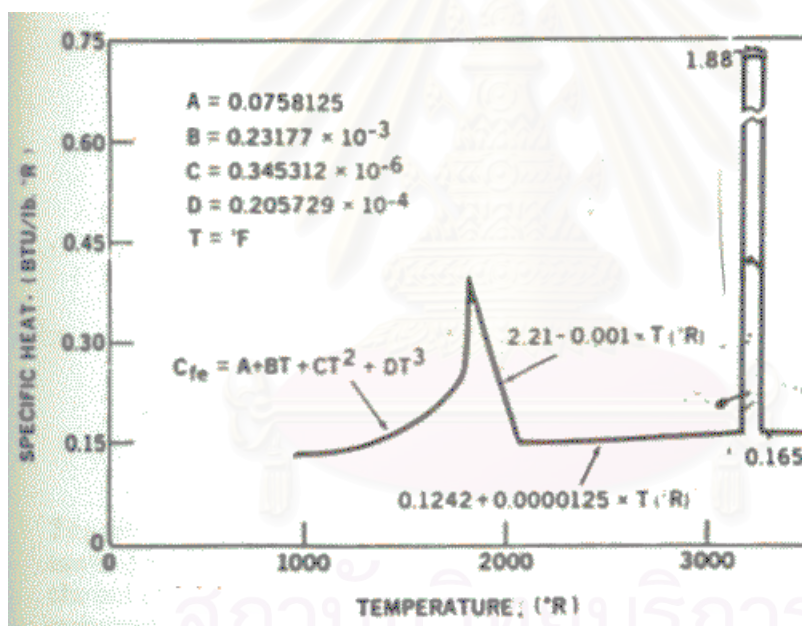
3.5 การคิดความร้อนแฝงจากการแข็งตัว

เนื่องจากการแข็งตัวของน้ำโลหะจะมีการคายความร้อนออกมาจำนวนหนึ่ง เรียกว่า ความร้อนแฝงจากการแข็งตัว (latent heat of freezing) ซึ่งในการดูแลความร้อนของเรา ไม่ได้คิด พจน์การคายความร้อนเข้าไปในสมการด้วย ดังนั้นหากไม่มีการนำความร้อนส่วนนี้ไปคำนวณด้วย แล้ว ผลการคำนวณที่ได้จะมีความคลาดเคลื่อนไปอย่างมาก ในการประดิษฐ์โปรแกรมนี้จึงได้นำ ระเบียบวิธีเทียบเท่าความร้อนจำเพาะ (equivalent specific heat method) ซึ่งจะนำความร้อนที่ คายออกมาไปเพิ่มให้กับค่าความร้อนจำเพาะในช่วงอุณหภูมิที่เกิดการแข็งตัว ซึ่งแสดงดังในรูปที่ 3.10 นอกจากนี้แล้วเพื่อป้องกันความผิดพลาดอันเกิดจากปริมาณการแข็งตัวในเอลิเมนต์ที่ สมบูรณ์ต่างกันในแต่ละรอบการคำนวณ จึงได้นำระเบียบวิธีการกู้คืนอุณหภูมิ (temperature recovery method) เพื่อป้องกันไม่ให้เกิดความผิดพลาดในการคิดความร้อนแฝงที่มากเกินไป หรือน้อยเกินไป ซึ่งหลักการของวิธีนี้จะพิจารณาดูว่า เอลิเมนต์นั้นก่อนคำนวณเวลาถัดไป อยู่ใน ช่วงการแข็งตัวหรือไม่ และหลักจากคำนวณอุณหภูมิที่เวลาถัดไปแล้วอยู่ในช่วงการแข็งตัวหรือไม่ ซึ่ง ข้อมูลทั้งสองนี้ทำให้ทราบว่าปริมาณน้ำโลหะที่แข็งตัวจริงๆมีปริมาณเท่าใด จากนั้นจึงทำการปรับ

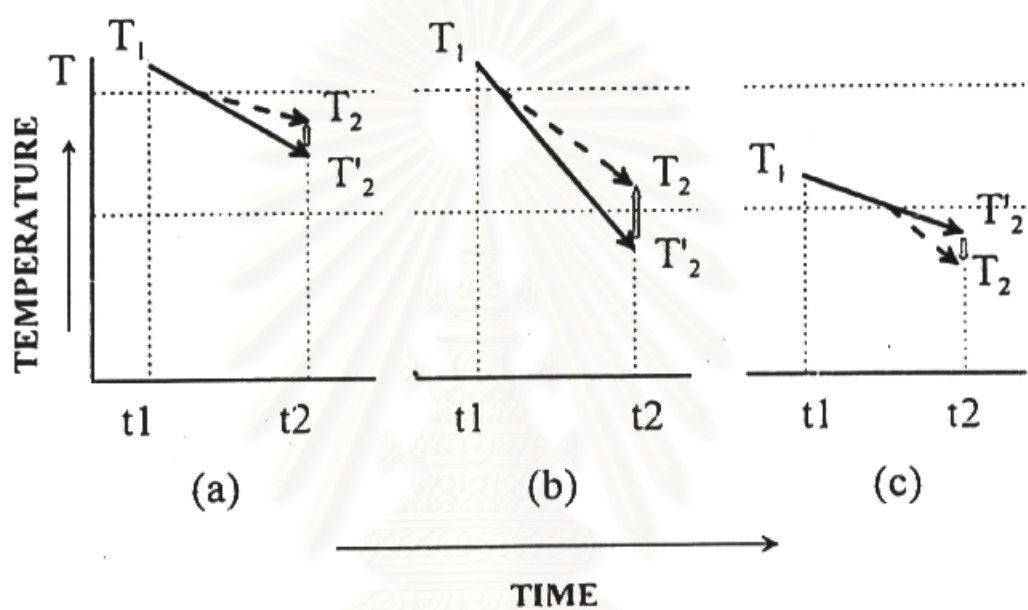
แก้คุณสมบัติที่เวลาถัดไปที่คำนวณได้ก่อนหน้าใหม่อีกครั้ง เงื่อนไขที่พิจารณาลักษณะการแข็งตัว แสดงดังในรูปที่ 3.11

3.6 แผนภาพแสดงลำดับขั้นตอนการคำนวณโดยใช้โปรแกรมทางคอมพิวเตอร์

แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติและในสองมิติ ซึ่งลำดับขั้นตอนการคำนวณแสดงดังรูปที่ 3.12 และสำหรับแบบจำลองที่คิดการถ่ายเทความร้อนในสามมิติแสดงดังรูปที่ 3.13 เห็นได้ว่าทั้งสองแบบมีการกำหนดสภาวะเริ่มต้น ที่เวลา เท่ากับศูนย์ คุณสมบัติของส่วนแบ่งต่างๆจะมีค่าเท่ากับคุณสมบัติของโลหะหลอมเหลว จากนั้นใช้สมการความสัมพันธ์ต่างๆที่ได้กล่าวมาข้างต้นทำการคำนวณหาค่าคุณสมบัติที่ตำแหน่งต่างๆ

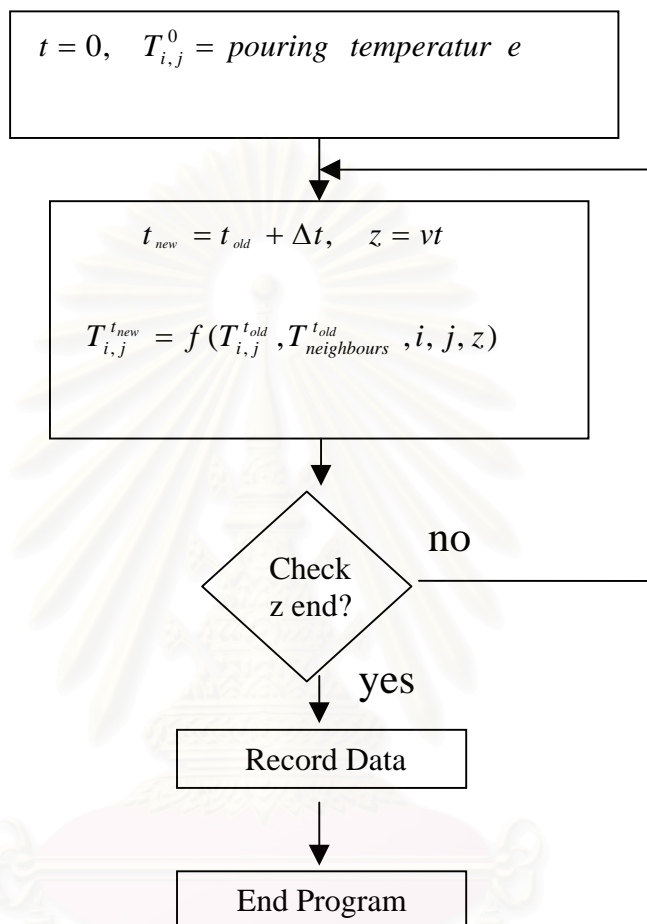


รูปที่ 3.10 แสดงการคิดค่าความร้อนจำเพาะที่อุณหภูมิต่างๆ โดยใช้ระเบียบวิธีเทียบเท่าความร้อนจำเพาะเพื่อชดเชยความร้อนแฝงจากการแข็งตัว⁵



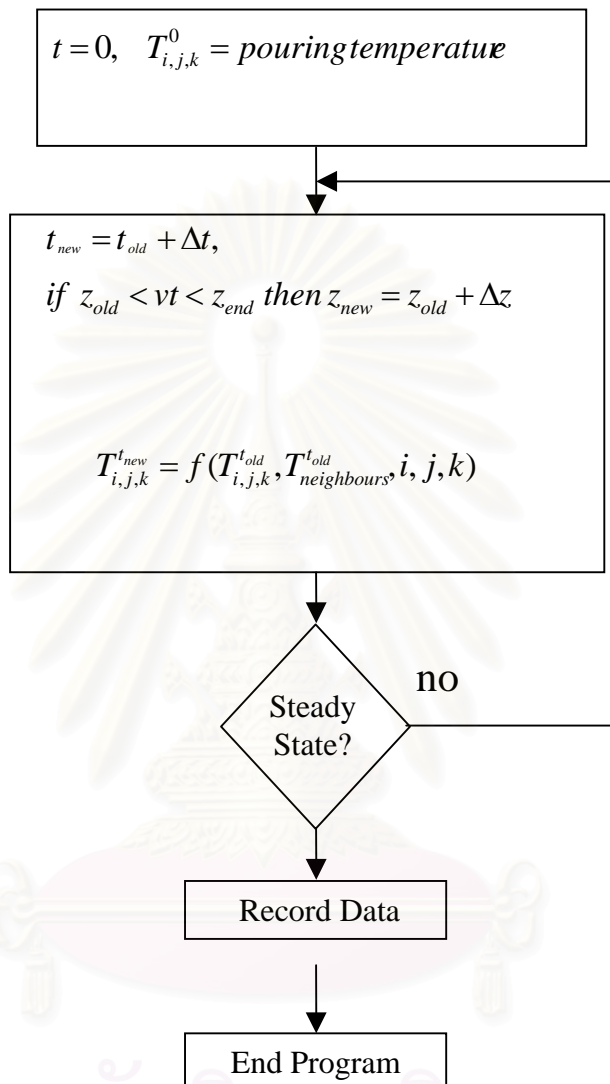
รูปที่ 3.11 แสดงการคิดการชดเชยความผิดพลาดที่เกิดจากลักษณะการแข็งตัวแบบ
 ต่างๆในแต่ละเอลิเมนต์ของระเบียบวิธีการกู้คืนอุณหภูมิ

สถาบันนวัตกรรมการ
 จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.12 แสดงลำดับขั้นตอนในการหาการกระจายตัวของอุณหภูมิในการถ่ายเทความร้อนในระนาบหนึ่งมิติและสองมิติ

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.13 แสดงลำดับขั้นตอนในการหาการกระจายตัวของอุณหภูมิในการถ่ายเทความร้อนในระนาบสามมิติ

บทที่ 4

ผลการวิเคราะห์ข้อมูล

4.1 แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติเพื่อใช้ในการหล่อแบบต่อเนื่องแบบแท่งแบนสำหรับเหล็กกล้า

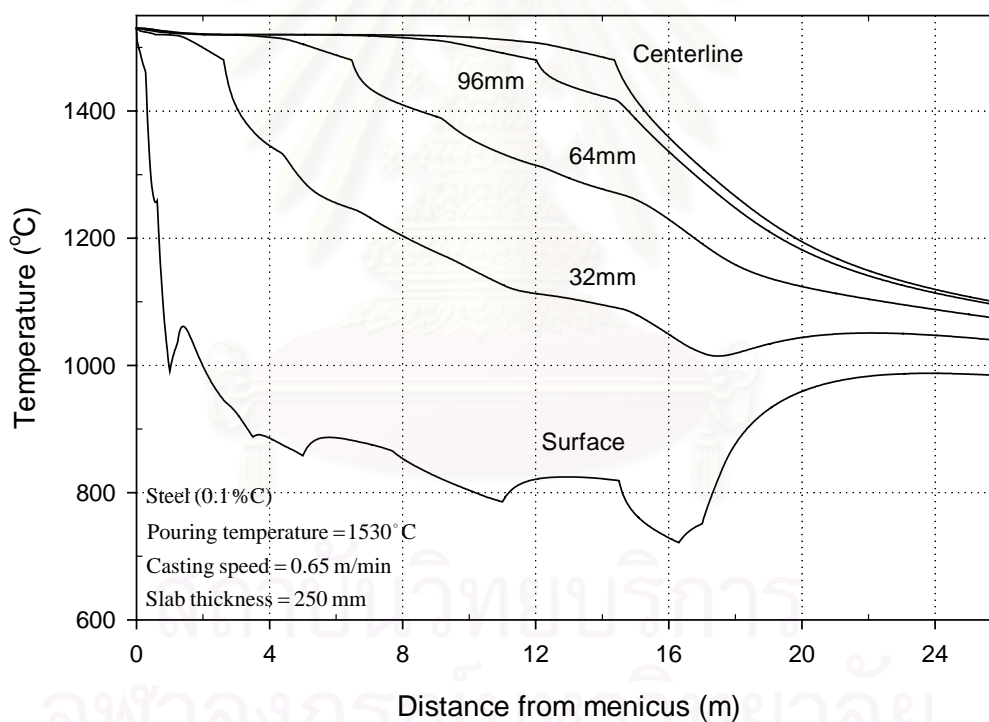
4.1.1 การเปรียบเทียบผลการคำนวณที่ได้กับงานวิจัยที่อ้างอิง

นำโปรแกรมคอมพิวเตอร์ที่ประดิษฐ์ขึ้นมาจำลองการหล่อแบบต่อเนื่องแบบแท่งแบนที่เป็นเหล็กกล้าคาร์บอน 0.1% C ขนาดความหนา 250 มม. โดยมีอุณหภูมิในการเท 1530 °C ที่ความเร็วในการหล่อเท่ากับ 0.65 เมตร/ นาที โดยแบบจำลองคิดการถ่ายเทความร้อนในหนึ่งมิติ และเนื่องจากชิ้นงานมีความสมมาตรจึงคิดรูปร่างของชิ้นงานเพียงครึ่งเดียว ซึ่งผลการคำนวณที่ได้ดังแสดงในรูปที่ 4.1 แสดงการเย็นตัวที่ตำแหน่งต่างๆในชิ้นงาน เปรียบเทียบกับ รูปที่ 4.2 แสดงผลการทดลองจากงานวิจัยที่นำมาอ้างอิงที่มีการทดลองยืนยัน พบว่ามีความใกล้เคียงกันมาก โดยอุณหภูมิที่ผิวที่คำนวณได้อยู่ในช่วงที่วัดได้จริงจากการผลิตจริงในโรงงาน ดังนั้นสามารถสรุปได้ว่าโปรแกรมที่ประดิษฐ์ขึ้นมาสามารถนำมาใช้จำลองการกระจายตัวของอุณหภูมิในการจำลองการหล่อแบบต่อเนื่องแบบแท่งแบนที่คิดการถ่ายเทความร้อนในหนึ่งมิติได้ และเนื่องจากวิธีการคิดการถ่ายเทความร้อนในสองมิติมีหลักการในการประดิษฐ์เหมือนกัน คือใช้สมการพื้นฐานในการคำนวณเหมือนกันกับการคิดในหนึ่งมิติ และมีระเบียบวิธีคิดในการคำนวณเหมือนกันจึงสามารถเทียบความน่าเชื่อถือได้แบบเดียวกัน ส่วนในกรณีวิธีการคิดการถ่ายเทความร้อนในระนาบสามมิติ ถึงแม้จะมีระเบียบวิธีคิดบางส่วนที่ต่างจากการถ่ายเทความร้อนในระนาบหนึ่งมิติ แต่สามารถเทียบความน่าเชื่อถือจากการคิดการถ่ายเทความร้อนในระนาบหนึ่งมิติได้ เนื่องจากมีระเบียบวิธีการคำนวณทางความร้อนต่างๆเหมือนกัน ส่วนรูปที่ 4.3 แสดงรูปร่างของบ่อน้ำโลหะที่คำนวณได้จากแบบจำลอง

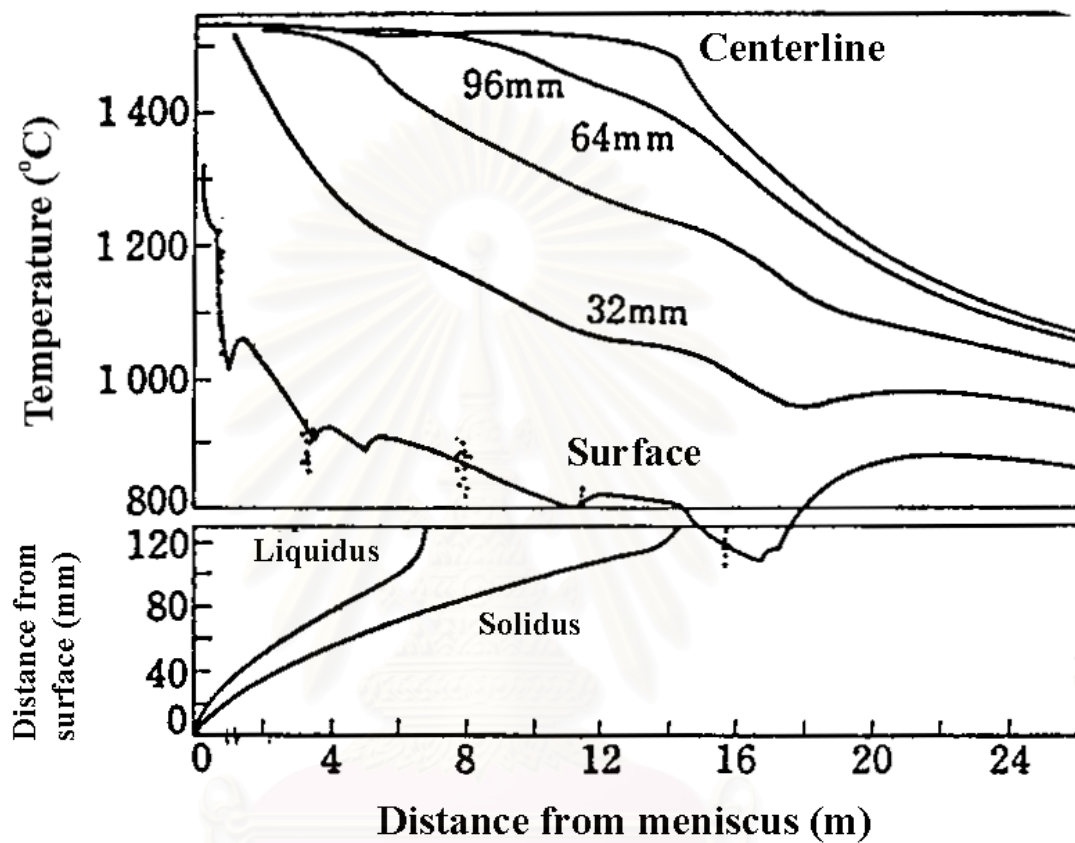
4.1.2 ผลของความเร็วในการหล่อต่อการเย็นตัวของชิ้นงาน

ผลของความเร็วในการหล่อต่อการเย็นตัวที่ตำแหน่งต่างๆ แสดงในรูปที่ 4.4 พิจารณาความเร็วที่ใช้ตั้งเท่ากับ 0.65 เมตรต่อนาที และ แบบหล่อมีความยาว 0.625 เมตร อุณหภูมิบริเวณผิวจะลดลงอย่างรวดเร็วเมื่อสัมผัสกับแบบหล่อ โดยอุณหภูมิที่ผิวขณะที่ออกจากแบบหล่อประมาณ 1250 °C และจะลดลงอย่างช้าๆ ภายหลังจากฉีดน้ำ(ระยะ 0.62-17.0 เมตรจากเมนส์คัส) เมื่อผ่านบริเวณที่มีการฉีดน้ำครั้งสุดท้ายจะมีอุณหภูมิต่ำสุดประมาณ 720 °C ที่ระยะห่างจากเมนส์คัส ประมาณ 16.2 เมตร เมื่อปล่อยให้เย็นตัวในอากาศอุณหภูมิจะเพิ่มขึ้นและจนเข้าสู่

สภาวะคงที่ ที่ระยะห่างจากเมนัสคัส ประมาณ 22 เมตรและมีอุณหภูมิประมาณ 985°C อุณหภูมิที่แกนกลางจะลดลงอย่างช้าๆจนถึง 1480°C ซึ่งจะแข็งตัวโดยสมบูรณ์ ที่ตำแหน่ง 14.5 เมตรจากเมนัสคัส จากนั้นจะอุณหภูมิจะค่อยๆลดลงจนถึงอุณหภูมิประมาณ 1100°C ที่ตำแหน่ง 25 เมตรจากเมนัสคัส ซึ่งที่จุดนี้บริเวณผิวและแกนกลางจะมีอุณหภูมิต่างกันประมาณ 120°C เมื่อทำการเปลี่ยนความเร็วในการหล่อพบว่า เมื่อลดความเร็วในการหล่อเป็น 0.55 เมตรต่อวินาที อุณหภูมิที่ออกจากแบบหล่อเท่าเดิม เมื่อผ่านบริเวณคีดน้ำอุณหภูมิจะลดลงมากกว่า เนื่องจากมีเวลาสัมผัสกับน้ำได้มากกว่า โดยมีอุณหภูมิต่ำสุดในช่วงนี้ประมาณ 700°C และเมื่อเย็นตัวในอากาศ พบว่า

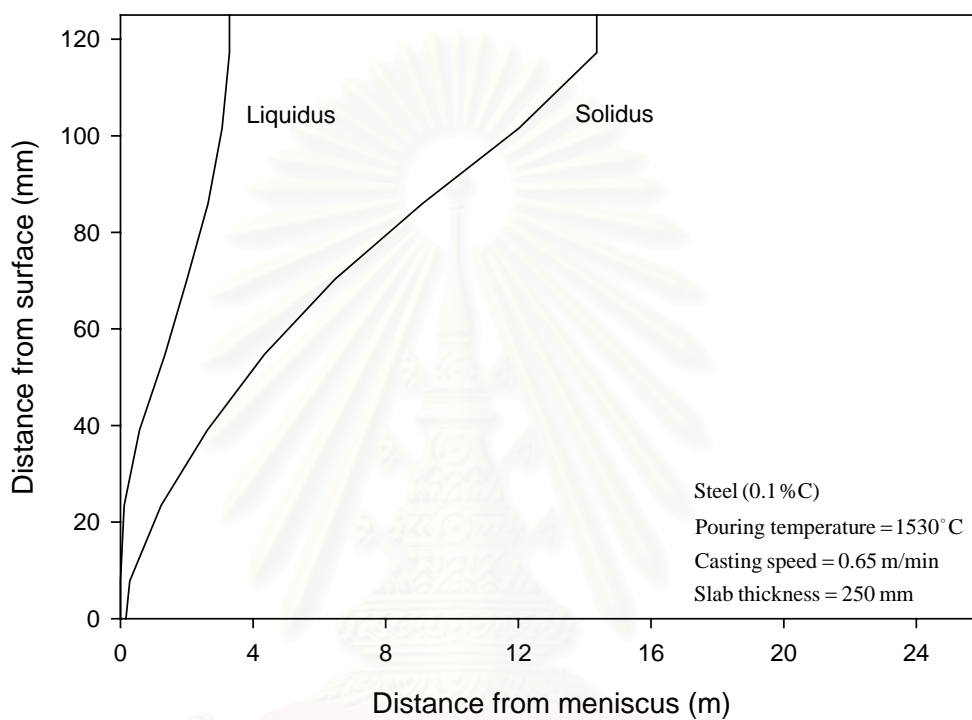


รูปที่ 4.1 แสดงผลการคำนวณการเย็นตัวที่ตำแหน่งต่างๆ โดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ



รูปที่ 4.2 แสดงผลการคำนวณการเย็นตัวที่ตำแหน่งต่าง (แสดงด้วยเส้นกราฟ) โดยมีผลเปรียบเทียบกับข้อมูลที่ได้จากการวัดจริงในโรงงาน (แสดงเป็นจุดในภาพ)

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.3 แสดงรูปร่างของบ่อน้ำโลหะที่คำนวณจากแบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ

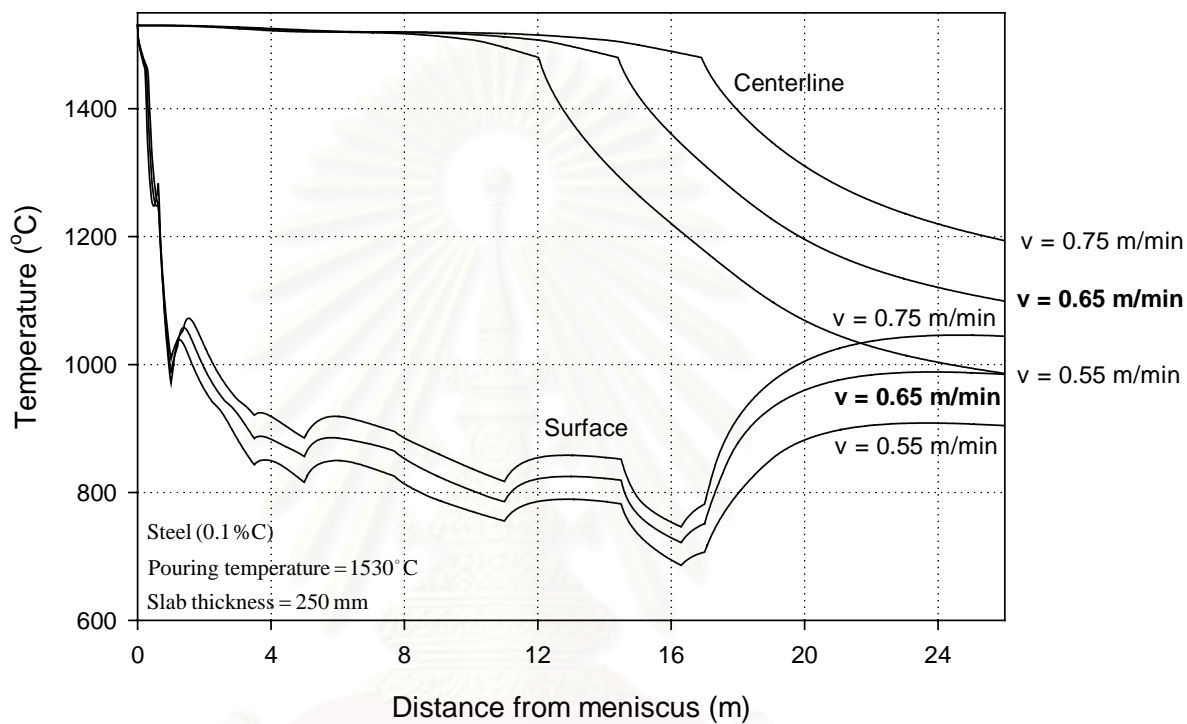
สถาบันวิทยบริการ
 จุฬาลงกรณ์มหาวิทยาลัย

จะมีอุณหภูมิคงที่ที่ประมาณ $900\text{ }^{\circ}\text{C}$ ในขณะที่บริเวณแกนกลางระยะที่ซึ่งแข็งตัวโดยสมบูรณ์จะเกิดที่ระยะที่สั้นลงมา โดยเกิดที่ตำแหน่งห่างจากเมนิสคัส ประมาณ 12 เมตร และจะมีอุณหภูมิประมาณ $885\text{ }^{\circ}\text{C}$ ที่ตำแหน่ง 25 เมตร จากเมนิสคัส บริเวณผิวและแกนกลางจะมีอุณหภูมิต่างกันประมาณ $80\text{ }^{\circ}\text{C}$ เมื่อเพิ่มความเร็วในการหล่อเป็น 0.75 เมตร พบว่าอุณหภูมิที่เหล็กออกจากแบบหล่อไม่แตกต่างกันมากนัก อุณหภูมิในช่วงที่ฉีดน้ำจะสูงขึ้นเนื่องจากการเวลาที่สัมผัสกับน้ำลดลง โดยมีอุณหภูมิสูงสุดที่ประมาณ $700\text{ }^{\circ}\text{C}$ เมื่อปล่อยให้เย็นในอากาศจะมีอุณหภูมิเริ่มคงที่ที่ประมาณ $1045\text{ }^{\circ}\text{C}$ และอุณหภูมิบริเวณแกนกลางจะแข็งตัวโดยสมบูรณ์ ที่ตำแหน่ง 17 เมตร และอุณหภูมิลดลงเรื่อยๆ จนถึงประมาณ $1200\text{ }^{\circ}\text{C}$ ที่ตำแหน่ง 25 เมตร จากเมนิสคัส บริเวณผิวและแกนกลางจะมีอุณหภูมิต่างกันประมาณ $160\text{ }^{\circ}\text{C}$

ดังนั้นจึงสามารถสรุปได้ว่า การลดความเร็วหรือเพิ่มความเร็วในการหล่อในช่วงที่ทำการจำลองไม่มีผลต่ออุณหภูมิที่ออกจากแบบหล่อมามากนัก แต่จะมีผลมากต่อบริเวณที่ฉีดน้ำและระยะห่างจากเมนิสคัสที่มีการแข็งตัวทั้งหมด เมื่อลดความเร็วในการหล่อสามารถลดอุณหภูมิที่ผิวได้มากกว่า และ จะมีการแข็งตัวทั้งหมดที่ระยะห่างจากเมนิคัส ที่ระยะสั้นกว่าเนื่องจากการถ่ายเทความร้อนออกไปได้มาก ทำให้ความแตกต่างอุณหภูมิระหว่างผิวและแกนกลางภายหลังการเย็นตัวในอากาศลดลง ในขณะที่การเพิ่มความเร็วในการหล่อนั้นจะทำให้อุณหภูมิที่ผิวสูงขึ้นและระยะที่แข็งตัวทั้งหมดจากเมนิสคัส จะเพิ่มขึ้นและ ความแตกต่างของอุณหภูมิที่ผิวและแกนกลางในช่วงที่เย็นตัวในอากาศจะเพิ่มสูงขึ้นด้วย

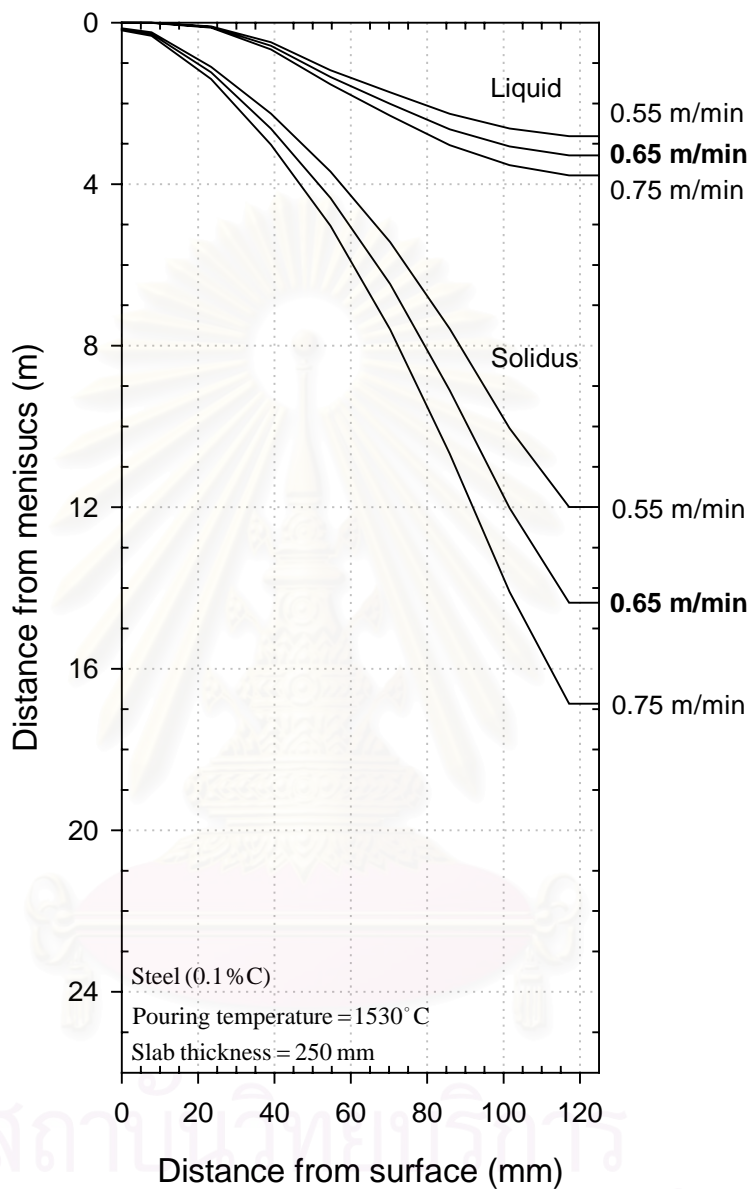
4.1.3 ผลของความเร็วในการหล่อต่อรูปร่างของบ่อน้ำโลหะ

รูปที่ 4.5 แสดงผลของความเร็วในการหล่อต่อรูปร่างของบ่อน้ำโลหะ พบว่าที่ความเร็วในการหล่อ 0.65 เมตรต่อวินาที ความหนาของเปลือกที่แข็งตัวจะค่อยๆเพิ่มขึ้นเมื่อระยะห่างจากเมนิสคัส เพิ่มขึ้นและมีความลึกของบ่อน้ำโลหะประมาณ 14.5 เมตร เมื่อลดความเร็วในการหล่อลงเป็น 0.55 เมตรต่อวินาที ความลึกของบ่อน้ำโลหะลดลงเหลือประมาณ 12 เมตร และเมื่อเพิ่มความเร็วในการหล่อเป็น 0.75 เมตรต่อวินาทีความลึกของบ่อน้ำโลหะประมาณ 17 เมตร ดังนั้นจึงสรุปได้ว่า ความเร็วในการหล่อมมีผลต่อความลึกของบ่อน้ำโลหะ ถ้าความเร็วเพิ่มขึ้นความลึกของบ่อน้ำโลหะมากขึ้นเนื่องจากการระบายความร้อนออกไปได้ยากขึ้น แต่ในกรณีที่ลดความเร็วใน



รูปที่ 4.4 ผลของความเร็วในการหล่อต่อการเย็นตัวที่บริเวณผิวและตรงกลางชิ้นงานที่คำนวณจากแบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.5 ผลของความเร็วในการหล่อต่อช่วงการแข็งตัวของน้ำโลหะในการถ่ายเทความร้อนในหนึ่งมิติ

การหล่อ ความร้อนสามารถส่งผ่านออกไปได้มากขึ้นเนื่องจากสัมผัสกับน้ำที่ฉีดได้นานขึ้น ซึ่งความลึกของบ่อโลหะมีผลอย่างมากต่อระยะเวลาตัด ซึ่งเป็นผลโดยตรงต่อการออกแบบกระบวนการและกำลังการผลิต

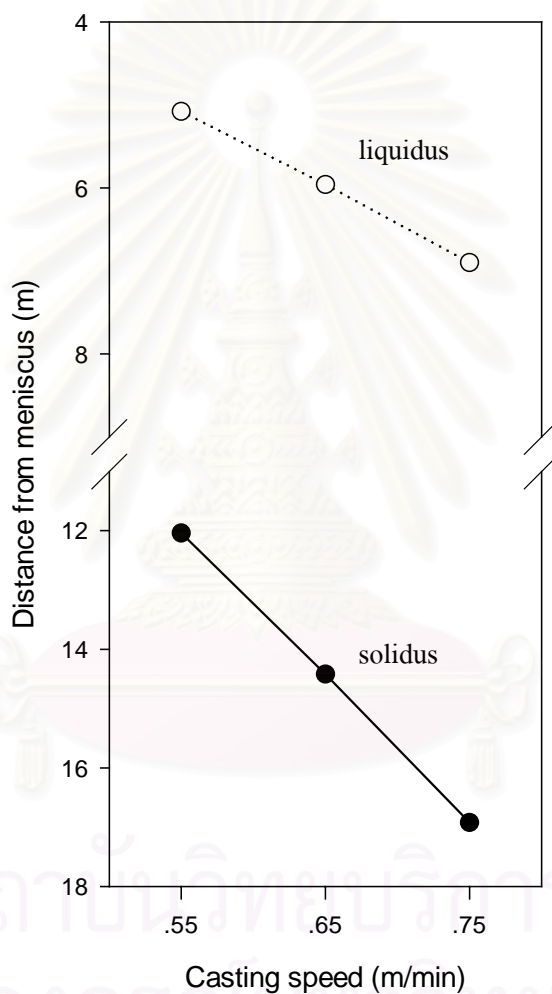
4.1.4 ผลของความเร็วในการหล่อต่อช่วงการแข็งตัวของตำแหน่งแกนกลางของชิ้นงาน

รูปที่ 4.5 ยังแสดงให้เห็นถึงผลของการเปลี่ยนแปลงความเร็วในการหล่อต่อการแข็งตัวของน้ำโลหะที่ตำแหน่งต่างๆของชิ้นงาน การแข็งตัวจะเกิดขึ้นที่บริเวณผิวโดยความเร็วของการดึงจะมีผลน้อยมากในบริเวณนี้ผลของความเร็วจะเห็นได้ชัดขึ้นที่ระยะห่างจากแบบหล่อมากขึ้นโดยเฉพาะบริเวณกลางของชิ้นงาน จากกราฟพบว่า ที่ความเร็วในการหล่อ 0.55 เมตรต่อนาที น้ำโลหะจะเริ่มมีการแข็งตัวและสิ้นสุดการแข็งตัวเร็วที่สุดในขณะที่ความเร็วในการหล่อ 0.75 เมตรต่อนาที มีการแข็งตัวช้าที่สุด รูปที่ 4.6 แสดงผลของความเร็วในการหล่อต่อเส้นการหลอมเหลวและเส้นการแข็งตัวของน้ำโลหะที่ตำแหน่งแกนกลาง จากรูปจะเห็นว่า ที่ความเร็วในการหล่อเท่ากับ 0.65 เมตรต่อนาที พบว่าน้ำโลหะเริ่มแข็งตัวที่ระยะ 6 เมตรจากเมนัสคัส และแข็งตัวโดยสมบูรณ์ที่ระยะ 14.5 เมตร เมื่อลดความเร็วในดึงเป็น 0.55 เมตรต่อนาที น้ำโลหะเริ่มแข็งตัวที่ระยะประมาณ 5 เมตร และแข็งตัวโดยสมบูรณ์ที่ระยะ 12 เมตร ในขณะที่เพิ่มความเร็วในการหล่อเป็น 0.75 เมตรต่อนาที น้ำโลหะจะแข็งตัวที่ระยะประมาณ 7 เมตร และ แข็งตัวโดยสมบูรณ์ที่ระยะ 17 เมตร ดังนั้นสรุปได้ว่า ความเร็วในการหล่อมีผลต่อความเร็วในการแข็งตัวของน้ำโลหะที่แกนกลางมากกว่าบริเวณผิวของชิ้นงาน ถ้าความเร็วในการหล่อช้าลง การแข็งตัวก็จะเกิดที่ระยะห่างจากเมนัสคัสที่สั้นลง และมีช่วงของการแข็งตัวสั้นลง ในขณะที่เพิ่มความเร็วในการหล่อ จะเพิ่มระยะในการแข็งตัวของน้ำโลหะ

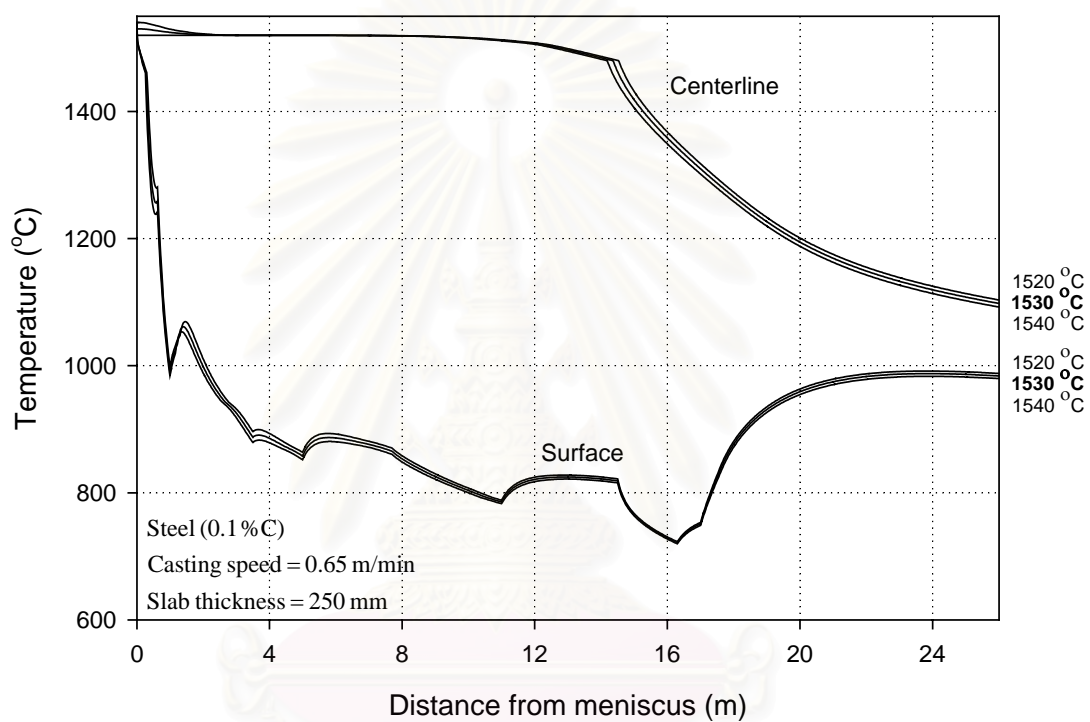
4.1.5 ผลของอุณหภูมิเทต่อการแข็งตัวของน้ำโลหะ

รูปที่ 4.7 แสดงผลของอุณหภูมิเทต่อการเย็นตัวของตำแหน่งต่างๆ และรูปที่ 4.8 แสดงผลของอุณหภูมิในการเทต่อเส้นการแข็งตัวของน้ำโลหะ ที่อุณหภูมิในการเท 1520 °C พบว่าน้ำโลหะเริ่มมีการแข็งตัวทันที ซึ่งยากต่อการควบคุมในทางปฏิบัติ เมื่อเพิ่มอุณหภูมิในการเทเป็น 1530 °C น้ำโลหะมีการแข็งตัวช้าลง เมื่อเพิ่มอุณหภูมิของน้ำโลหะสูงขึ้นน้ำโลหะเริ่มแข็งตัวที่ระยะห่างจากเมนัสคัสมากขึ้นโดยระยะสิ้นสุดการแข็งตัวก็เพิ่มขึ้นตามดังแสดงดังในรูป ผลของอุณหภูมิเทต่อการแข็งตัวของน้ำโลหะที่ตำแหน่งแกนกลางชิ้นงานแสดงดังรูปที่ 4.9 พบว่าที่อุณหภูมิในการเท 1530 °C น้ำโลหะเริ่มมีการแข็งตัวที่ระยะประมาณ 6 เมตร จากเมนัสคัส และแข็งตัวโดยสมบูรณ์ที่ระยะห่าง 14.5 เมตร เมื่อลดอุณหภูมิการเทลงเป็น 1520 °C ของเหลวเริ่มแข็งตัวทันทีที่สัมผัส

กับแบบหล่อ และแข็งตัวโดยสมบูรณ์ที่ระยะ ประมาณ 14.2 เมตร เมื่อเพิ่มอุณหภูมิสูงขึ้นเป็น 1540 °C ของเหลวเริ่มแข็งตัวที่ระยะประมาณ 7 เมตรและแข็งตัวหมดที่ระยะ 14.75 เมตร และเมื่อเพิ่มอุณหภูมิเป็น 1550 °C

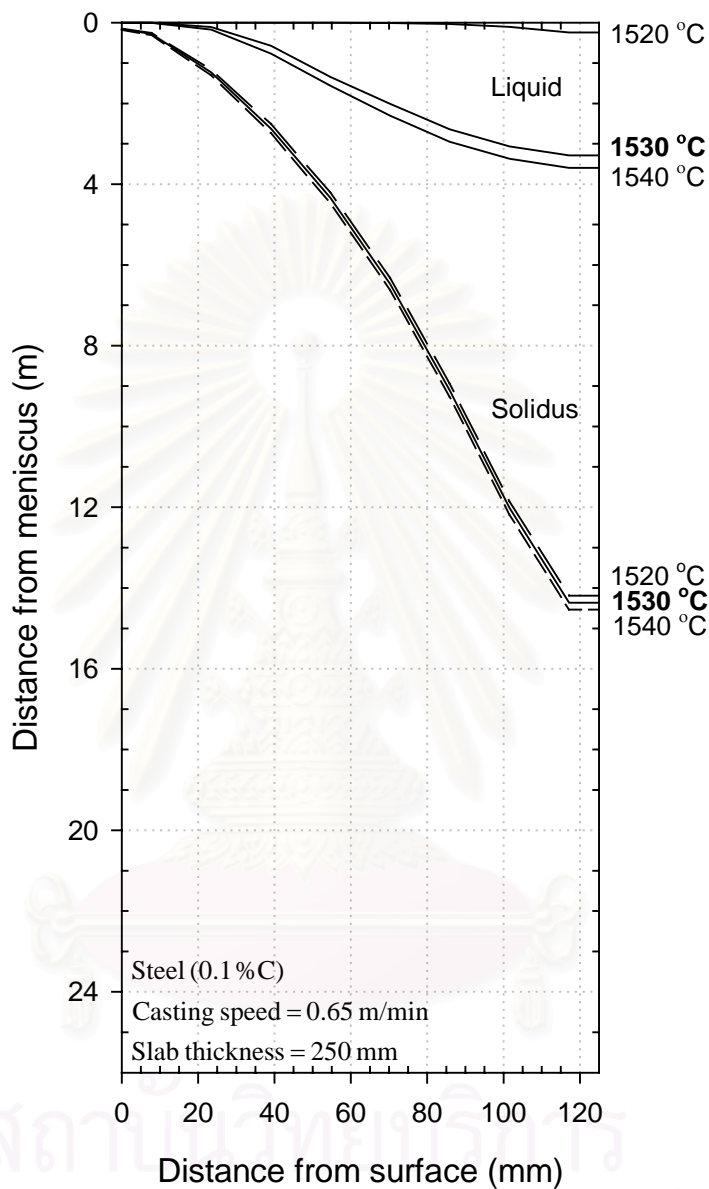


รูปที่ 4.6 ผลของความเร็วจนการหล่อต่อช่วงการแข็งตัวของน้ำโลหะที่ตำแหน่งแกนกลางที่คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ

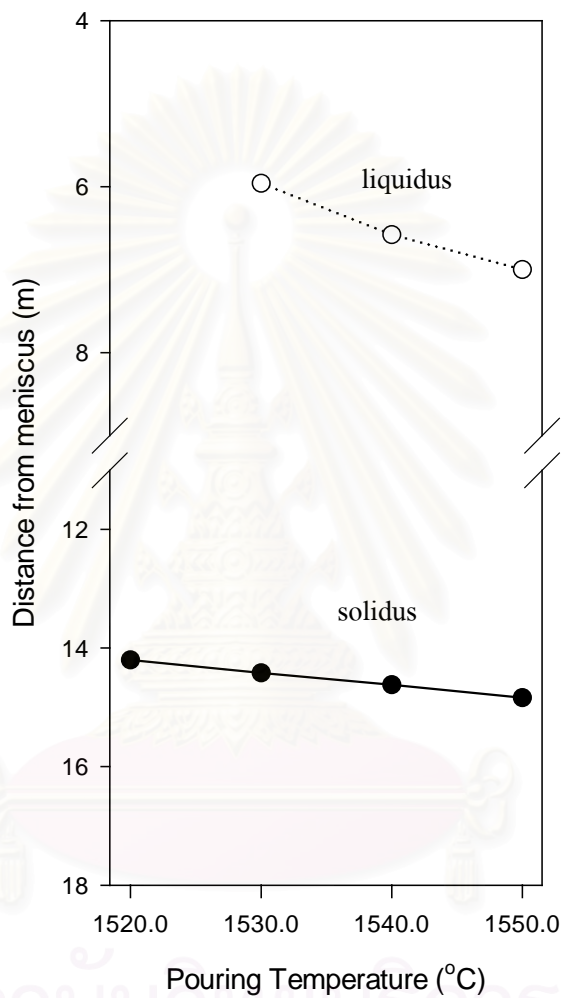


รูปที่ 4.7 ผลของอุณหภูมิต่อการเย็นตัวของผิวและตรงกลางชิ้นงานที่คำนวณจากแบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.8 ผลของอุณหภูมิในการติดต่อช่วงการแข็งตัวของน้ำโลหะจากแบบจำลองที่คิด
การถ่ายเทความร้อนในหนึ่งมิติ



รูปที่ 4.9 ผลของอุณหภูมิในการหล่อต่อการแข็งตัวของน้ำโลหะ ที่ตำแหน่งแกนกลาง ที่คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ

ของแข็งเริ่มแข็งตัวที่ระยะประมาณ 7.5 เมตร และแข็งตัวหมดที่ระยะประมาณ 15 เมตร ดังนั้นจึงสรุปได้ว่าผลของอุณหภูมิต่อการแข็งตัวของน้ำโลหะ จะมีผลต่อจุดเริ่มต้นและจุดสิ้นสุดของการแข็งตัวของของเหลว เมื่ออุณหภูมิในการเทสูงขึ้นระยะในการเริ่มแข็งตัวของน้ำโลหะเพิ่มขึ้นและระยะที่น้ำโลหะสิ้นสุดการแข็งตัวเพิ่มขึ้นโดยอุณหภูมิเทจะมีผลต่อระยะที่เริ่มมีการแข็งตัวของน้ำโลหะมากกว่าระยะที่สิ้นสุดการแข็งตัวของน้ำโลหะ

4.1.6 ผลของความเร็วในการหล่อต่อความหนาของเปลือกที่แข็งตัว

ความสัมพันธ์ระหว่างความเร็วในการหล่อต่อความหนาของเปลือกที่แข็งตัวภายในแบบหล่อแสดงดังรูปที่ 4.10 พบว่าที่อุณหภูมิในการเท 1530°C ความหนาของเปลือกที่แข็งตัวที่ความเร็วในการหล่อเท่ากับ 0.55 เมตรต่อนาที มีความหนามากที่สุด และที่ความเร็วในการหล่อเท่ากับ 0.75 เมตรจะมีความบางน้อยที่สุด ที่ตำแหน่งปลายแบบหล่อตรงบริเวณทางออกจากแบบหล่อ ที่ความเร็ว 0.55 เมตรต่อนาทีมีความหนาของเปลือกที่แข็งตัวประมาณ 19.5 มิลลิเมตร ที่ความเร็วในการหล่อ 0.65 เมตรต่อนาทีมีความหนาประมาณ 18.5 มิลลิเมตร และที่ความเร็วในการหล่อ 0.75 เมตรต่อวินาที มีความหนาประมาณ 17.5 มิลลิเมตร เมื่อพิจารณาการแข็งตัวทั้งหมดซึ่งแสดงดังรูปที่ 4.11 พบว่า ความแตกต่างของความหนาเพิ่มมากขึ้นเมื่อผ่านระยะของการฉีดน้ำ(ระยะที่ออกจากแบบหล่อจนถึงระยะ 17 เมตรจากเมนัสคัส) ที่ความเร็วในการหล่อ 0.55 และ 0.65 เมตรต่อนาทีโลหะแข็งตัวทั้งหมดในช่วงการฉีดน้ำ ในขณะที่ความเร็วในการหล่อ 0.75 เมตรจะแข็งตัวหมดในช่วงรอยต่อระหว่างช่วงการฉีดน้ำและช่วงการเย็นตัวในอากาศ ความหนาของเปลือกที่แข็งตัวที่ตำแหน่งออกจากแบบหล่อมีความสำคัญมากต่อการหล่อเนื่อง เพราะส่งผลต่อความแข็งแรงของชิ้นงานหล่อในขณะที่ออกจากแบบ โดยปกติในการหล่อเหล็กแท่งแบน ควรมีความหนาของเปลือกที่แข็งตัวที่ตำแหน่งทางออกควรมากกว่า 8 มิลลิเมตร³

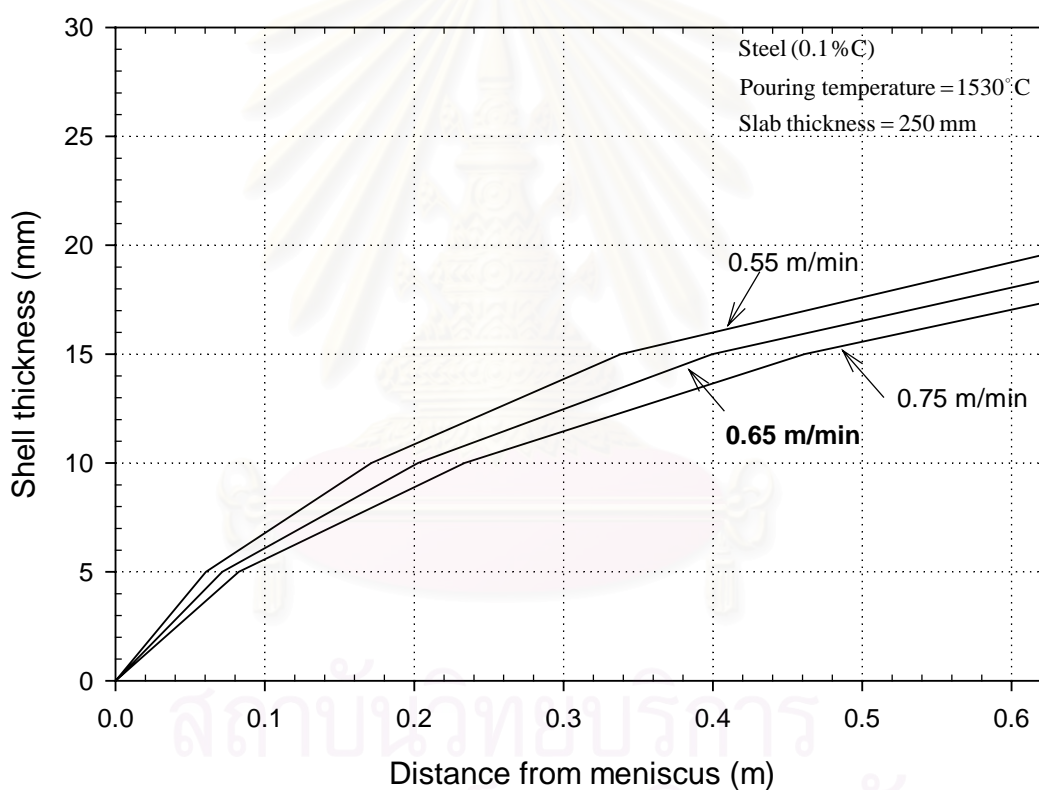
ดังนั้นจากผลการคำนวณจึงสามารถสรุปได้ว่าความหนาของเปลือกที่แข็งตัวในแบบหล่อเกี่ยวข้องกับความเร็วในการหล่อ กล่าวคือเมื่อความเร็วในการหล่อเพิ่มขึ้นจะทำให้ความหนาของเปลือกที่แข็งตัวที่ตำแหน่งทางออกของแบบหล่อลดลง

4.1.7 ผลของอุณหภูมิเทต่อความหนาของเปลือกที่แข็งตัว

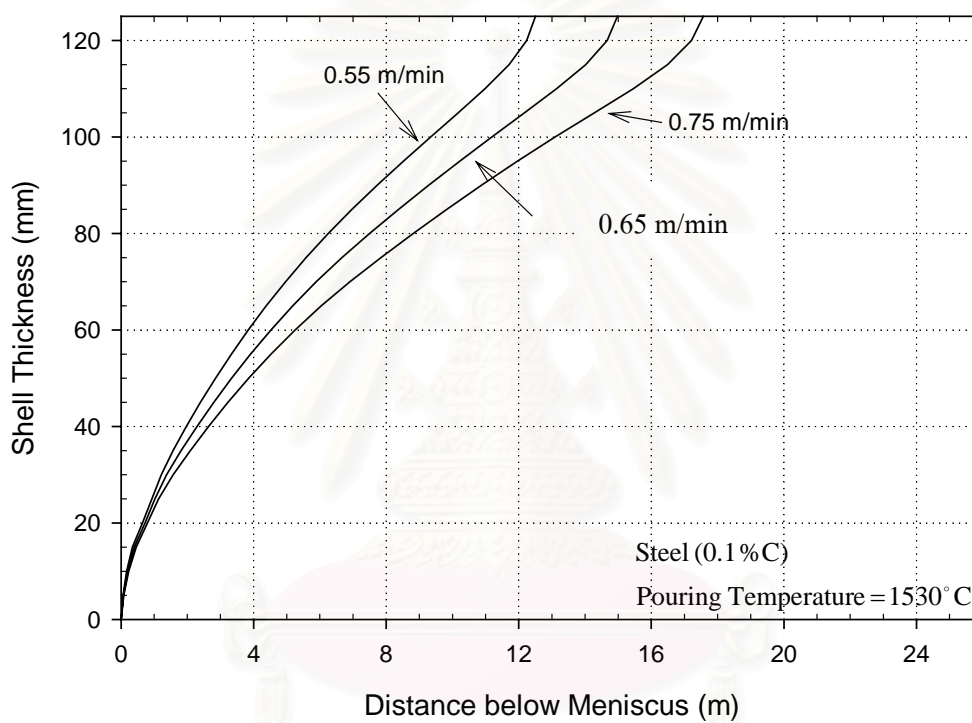
ผลของอุณหภูมิเทต่อความหนาของเปลือกที่แข็งตัว แสดงดังรูปที่ 4.12 และ 4.13 เมื่ออุณหภูมิในการเทสูงขึ้นน้ำโลหะมีการแข็งตัวที่ระยะห่างจากเมนัสคัสมากขึ้นเป็นผลให้ความหนาของเปลือกที่แข็งตัวที่ตำแหน่งทางออกลดลงดังแสดงในรูปที่ 4.11 โดยมีความหนาดำแหน่งทางออกอยู่ในช่วง 16.75-18.5 มิลลิเมตร เมื่อเพิ่มอุณหภูมิในการเทจาก 1520°C เป็น 1540°C เมื่อ

พิจารณาการแข็งตัวของผิวหลังการเย็นตัวด้วยการฉีดน้ำดังแสดงในรูปที่ 4.13 พบว่า อุณหภูมิที่จะมีผลต่อความหนาของเปลือกที่แข็งตัวที่บริเวณฉีดน้ำมากกว่าภายในแบบหล่อโดยเฉพาะจุดที่มีการแข็งตัวหมด ที่อุณหภูมิในการเทต่ำกว่าจะมีความหนาของเปลือกที่แข็งตัวที่บริเวณการฉีดน้ำมากกว่าที่อุณหภูมิในการเทสูงกว่า

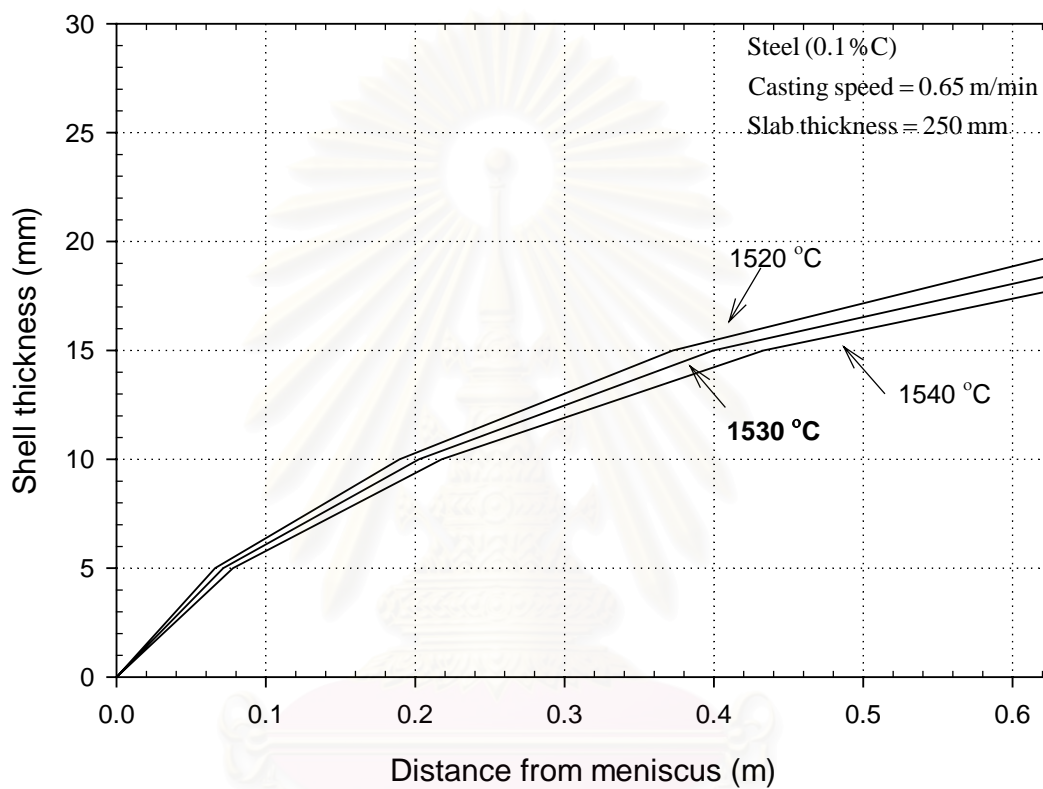
เมื่อเปรียบเทียบกับผลของความเร็วในการหล่อพบว่า ผลของอุณหภูมิส่งผลต่อความหนาของเปลือกที่แข็งตัวน้อยกว่าผลของความเร็วในการหล่อ



รูปที่ 4.10 ผลของความเร็วในการหล่อต่อความหนาของเปลือกที่แข็งตัวที่ ตำแหน่งต่างๆในแบบหล่อที่คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ

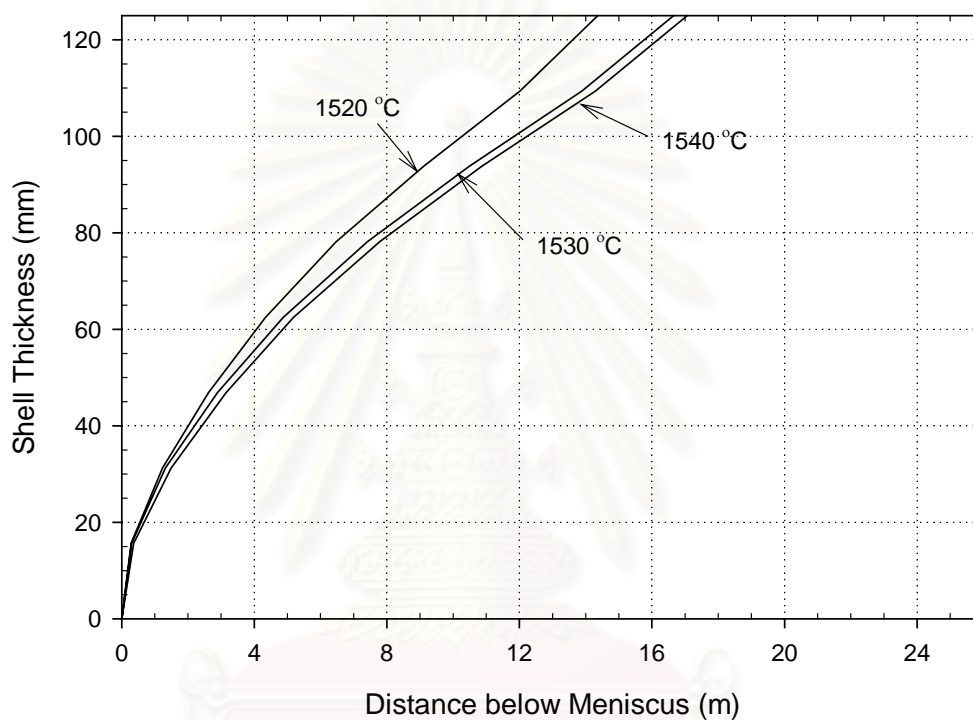


รูปที่ 4.11 ผลของความเร็วในการหล่อในการทดสอบความหนาของเปลือกที่แข็งตัวที่ตำแหน่งต่างๆในแบบหล่อคำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ



รูปที่ 4.12 ผลของอุณหภูมิเตาในการตั้งในการเทต่อความหนาของเปลือกที่แข็งตัวที่ระยะต่างๆ คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.13 ผลของอุณหภูมิในการเกิดความหนาของเปลือกที่แข็งตัวที่ระยะต่างๆ

คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติ

จุฬาลงกรณ์มหาวิทยาลัย

4.2 แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติเพื่อใช้ในการหล่อแบบต่อเนื่องแบบ แท่งสำหรับเหล็กกล้า

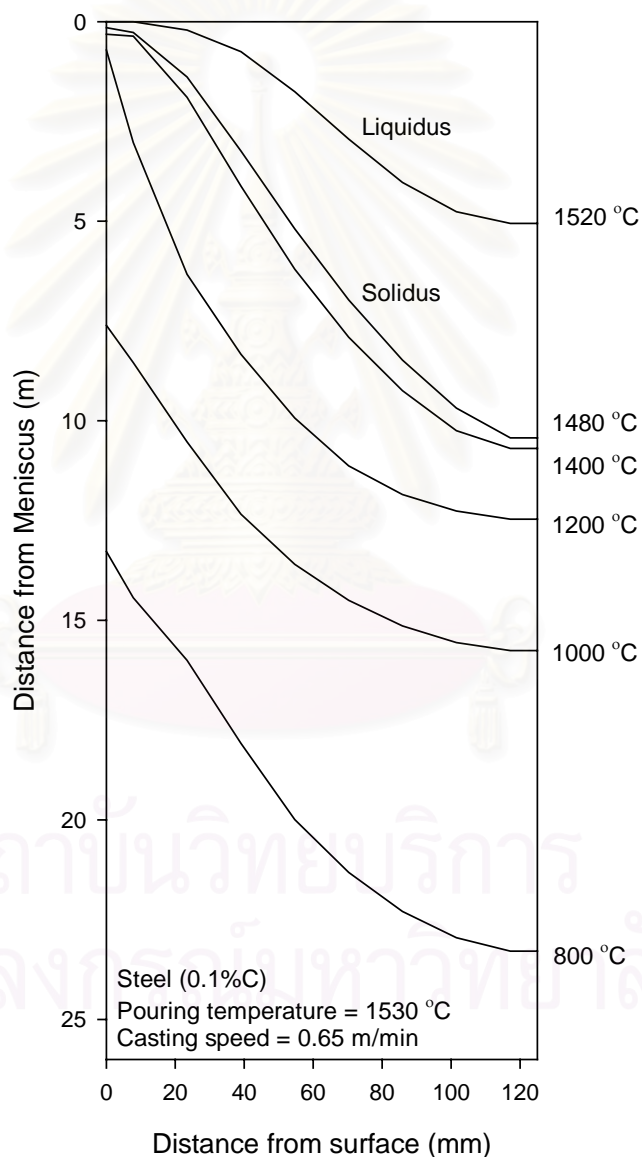
4.2.1 การคำนวณการกระจายตัวของอุณหภูมิโดยใช้แบบจำลองที่คิดการถ่ายเท ความร้อนในสองมิติ

การกระจายตัวของอุณหภูมิในแบบหล่อที่ระยะต่างๆจากเมนีสคัส โดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ ตามแนวยาวที่ตำแหน่งกึ่งกลาง แสดงดังรูปที่ 4.14 ชิ้นงานหล่อแบบแท่งเป็นเหล็กกล้า 0.1 %C รูปหน้าตัดเป็นสี่เหลี่ยมจัตุรัสขนาด 250 มม. X 250 มม. (เนื่องจากรูปร่างสมมาตรในสองแกน จึงนำมาคำนวณในแบบจำลองด้วยขนาดเพียง 1/4 ส่วน) ถูกดึงด้วยความเร็ว 0.65 เมตรต่อนาที โดยมีอุณหภูมิเท 1530 °C น้ำโลหะเริ่มแข็งตัวทันทีผิวสัมผัสกับแบบหล่อ ในขณะที่แกนกลางเริ่มแข็งตัวที่ระยะประมาณ 5 เมตรดังรูป และสิ้นสุดการแข็งตัวที่ระยะประมาณ 11 เมตร การกระจายตัวของอุณหภูมิตามภาคตัดขวางที่ตำแหน่งต่างๆแสดงดังรูปที่ 4.15 แสดงการกระจายตัวของอุณหภูมิตามภาคตัดขวางที่ตำแหน่งต่างๆ โดยรูป (a) แสดงการกระจายตัวของอุณหภูมิที่กึ่งกลางของแบบหล่อ รูป (b) แสดงที่ทางออกแบบหล่อ ซึ่งที่บริเวณนี้ชิ้นงานมีความหนาของเปลือกที่แข็งตัวประมาณ 18 มิลลิเมตรและมีอุณหภูมิผิวสูงสุดประมาณ 1390 °C ความหนาของเปลือกจะเพิ่มขึ้นเรื่อยๆเมื่อระยะห่างจากเมนีสคัสเพิ่มขึ้นเนื่องจากการขีดน้ำ โดยที่ระยะ 3 เมตรความหนาของเปลือกที่แข็งตัวมีค่าประมาณ 40 มิลลิเมตรโดยมีอุณหภูมิผิวสูงสุดประมาณ 1130 °C ที่ระยะ 6 เมตรจากเมนีสคัส น้ำโลหะที่เหลือจะอยู่ในช่วงเฟสของแข็งกับของเหลวโดยมีความหนาของเปลือกที่แข็งตัวประมาณ 70 มิลลิเมตรและมีอุณหภูมิผิวสูงสุดประมาณ 1063 °C ที่ระยะ 9 เมตรจากเมนีสคัส น้ำโลหะมีการแข็งตัวเกือบหมดโดยมีความหนาของเปลือกประมาณ 100 มิลลิเมตรและอุณหภูมิผิวสูงสุดประมาณ 930 °C และสุดท้ายที่ระยะห่างจากเมนีสคัส 12 เมตร ชิ้นงานมีการแข็งตัวหมดแล้วและมีอุณหภูมิที่ผิวน้อยกว่า 830 °C

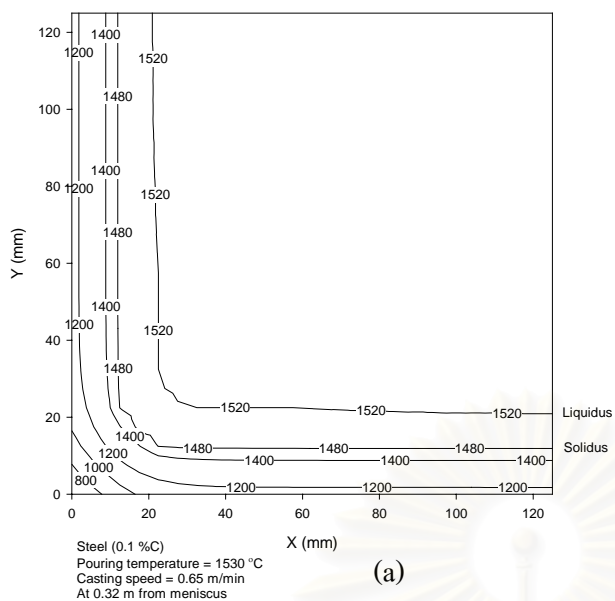
เมื่อเทียบกับผลการคำนวณที่ใช้แบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติที่ความเร็วในการหล่อที่เท่ากัน มีตัวแปรและพารามิเตอร์อื่นๆที่เหมือนกันพบว่า ที่ตำแหน่งแกนกลางการคำนวณโดยใช้การคิดการถ่ายเทความร้อนในสองมิติ น้ำโลหะมีการแข็งตัวเร็วมากขึ้นอันเนื่องมาจากมีการคิดการถ่ายเทความร้อนในสองทิศทาง ทำให้ความร้อนสามารถถ่ายเทได้เพิ่มขึ้นในอีกทิศทาง ช่วงของการแข็งตัวจึงสั้นกว่าในกรณีการคิดในแบบระนาบหนึ่งมิติ ดังนั้นการนำแบบจำลองที่คิดการถ่ายเทความร้อนในหนึ่งมิติมาคำนวณการถ่ายเทความร้อนในการหล่อแบบแท่งจึงให้ผลที่คลาดเคลื่อนไปมาก

4.2.2 ผลของความเร็วในการหล่อต่อความหนาของเปลือกโลหะ

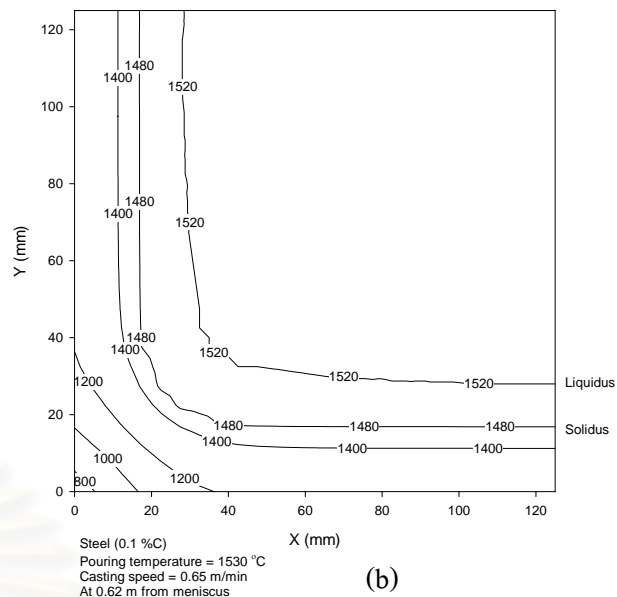
รูปที่ 4.15 , 4.16 และ 4.17 แสดงผลการคำนวณการกระจายตัวของอุณหภูมิในแนวภาคตัดขวางในตำแหน่งต่างๆที่ความเร็วในการหล่อ 0.65, 0.55 และ 0.75 เมตรต่อวินาทีตามลำดับ ในขณะที่น้ำโลหะอยู่ในแบบหล่อ ที่ระยะกึ่งกลางของแบบหล่อ ที่ความเร็วในการหล่อ 0.55 เมตรต่อวินาทีความหนาของเปลือกที่แข็งตัวประมาณ 20 มิลลิเมตร เมื่อความเร็วในการหล่อเพิ่มเป็น 0.65 เมตรต่อวินาที ความหนาของเปลือกที่แข็งตัวประมาณ 18 มิลลิเมตร และที่ความเร็วในการ



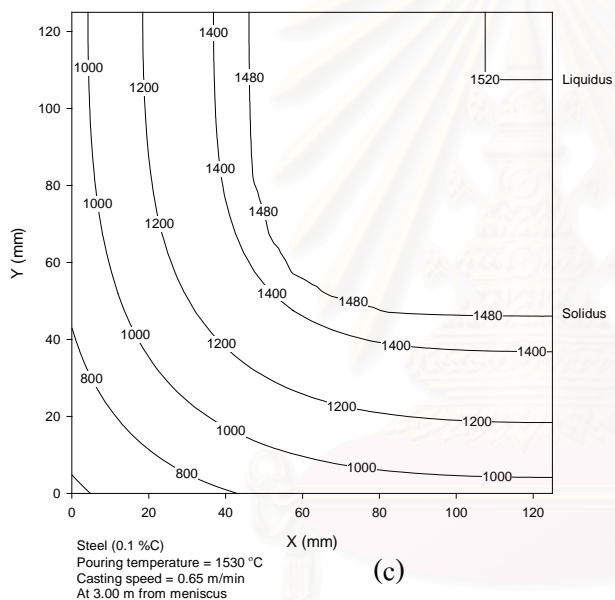
รูปที่ 4.14 แสดงการกระจายตัวของอุณหภูมิที่ระยะต่างๆจากเมนิสคัส คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ



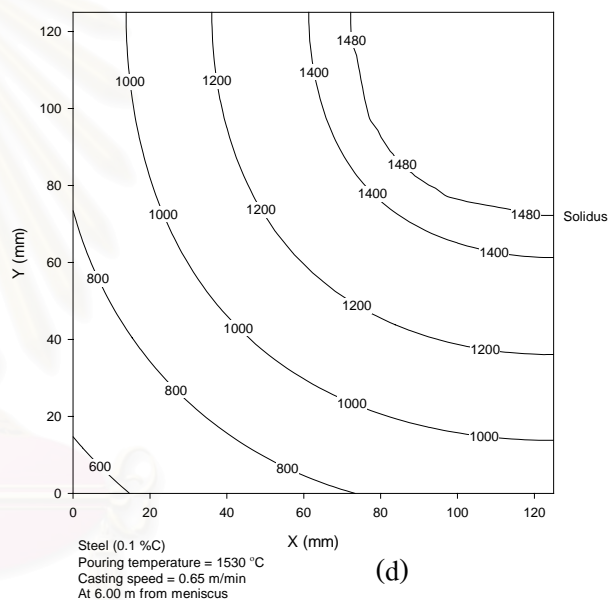
(a)



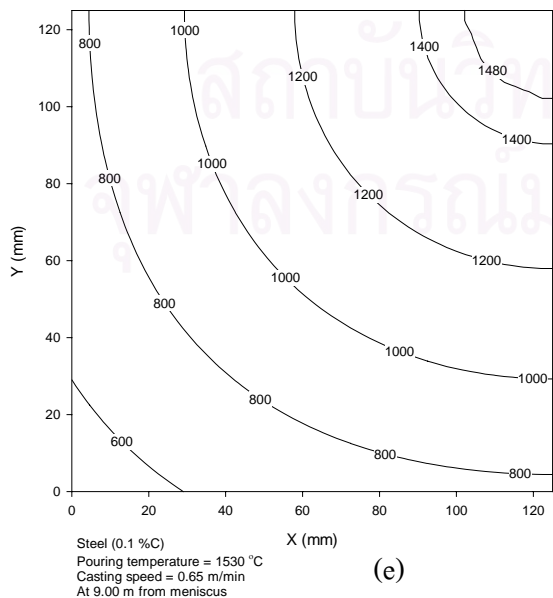
(b)



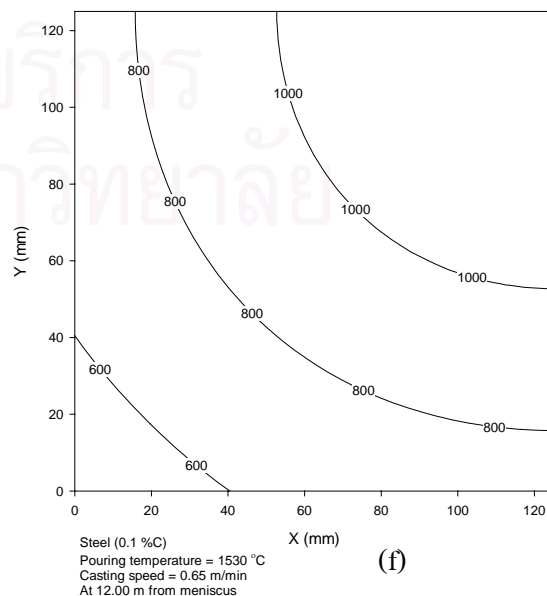
(c)



(d)

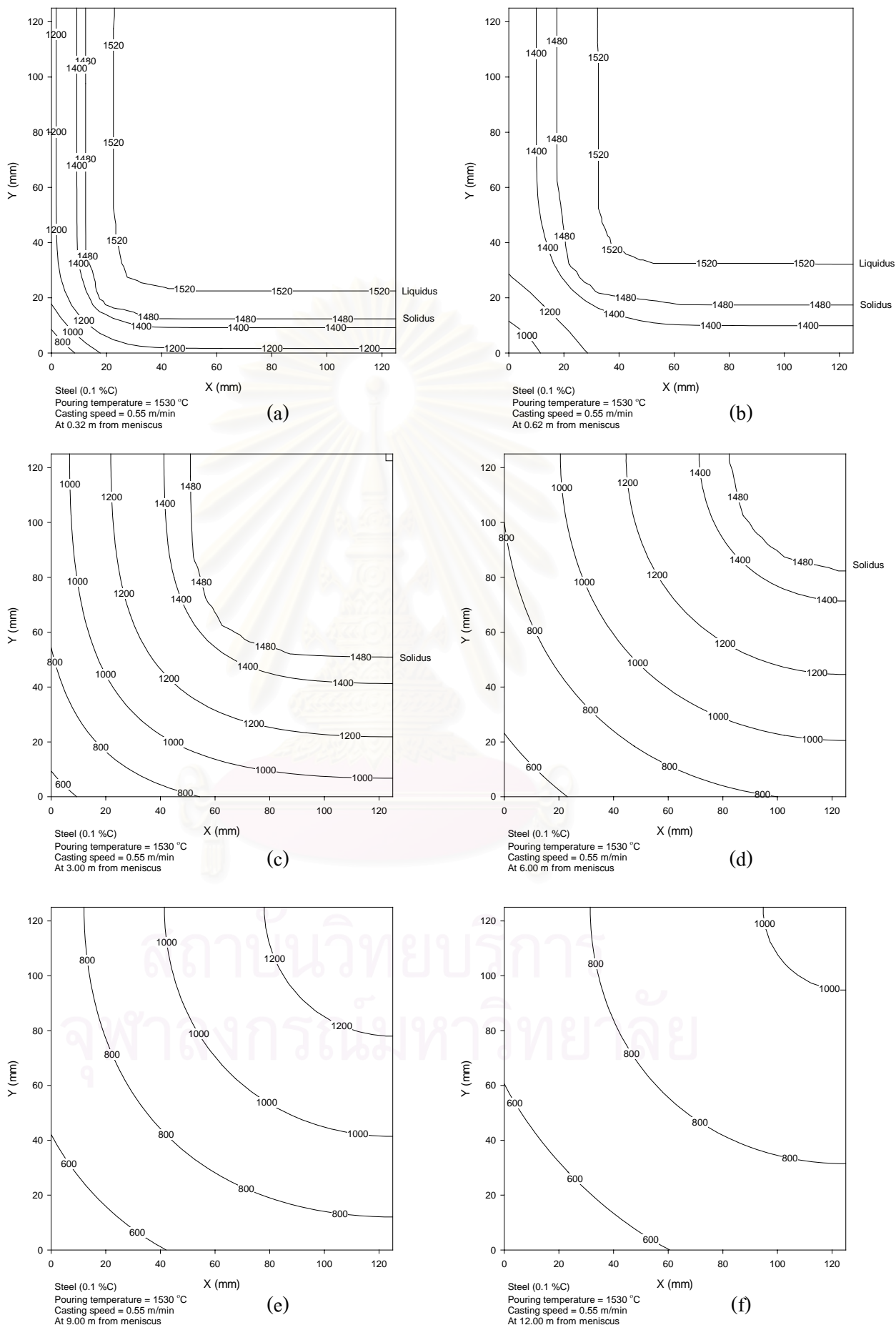


(e)

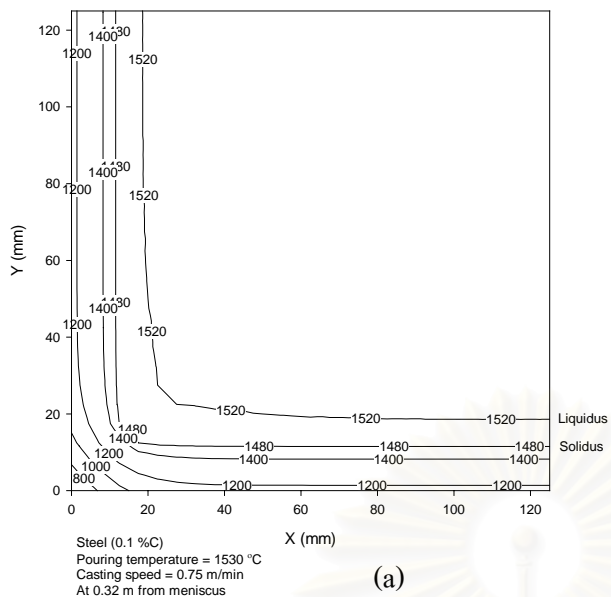


(f)

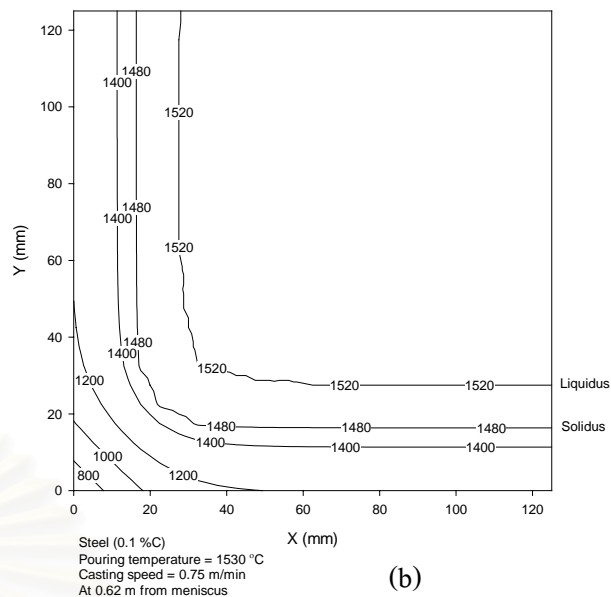
รูปที่ 4.15 แสดงการกระจายตัวของอุณหภูมิในภาคตัดขวาง ที่ความเร็วในการหล่อ 0.65 m/min



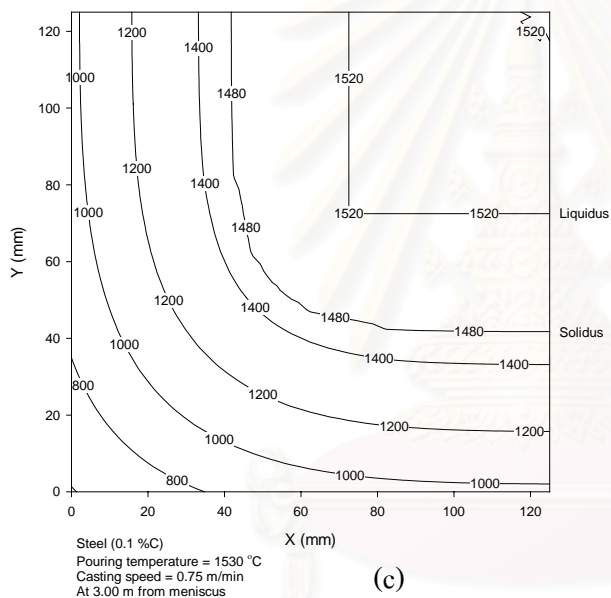
รูปที่ 4.16 แสดงการกระจายตัวของอุณหภูมิในภาคตัดขวาง ที่ความเร็วในการหล่อ 0.55 m/min



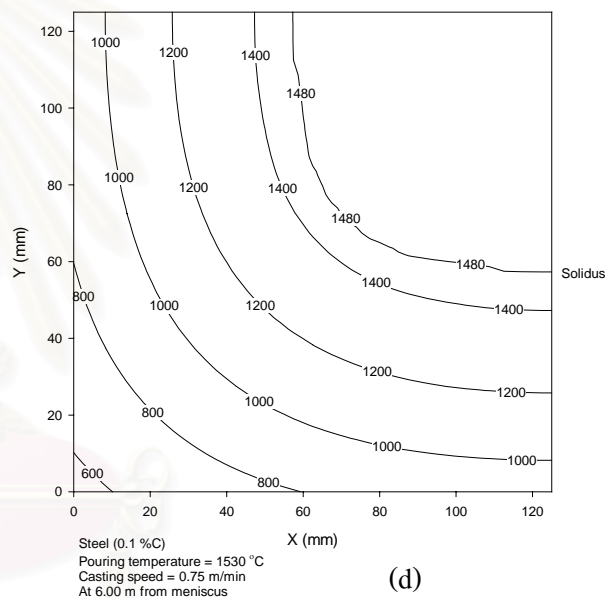
(a)



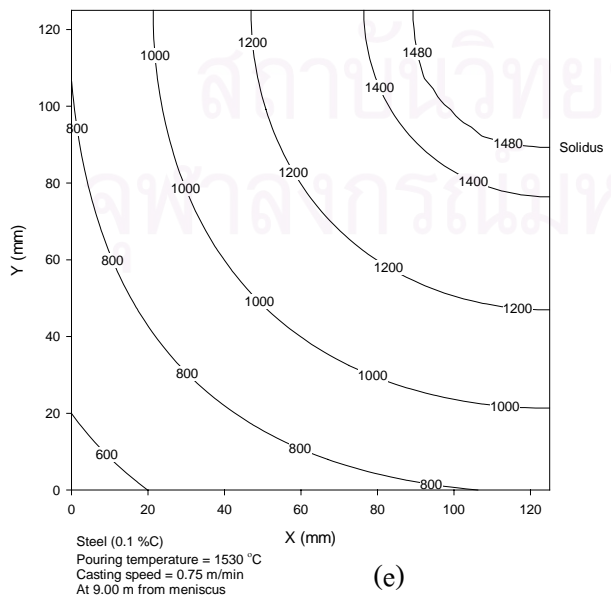
(b)



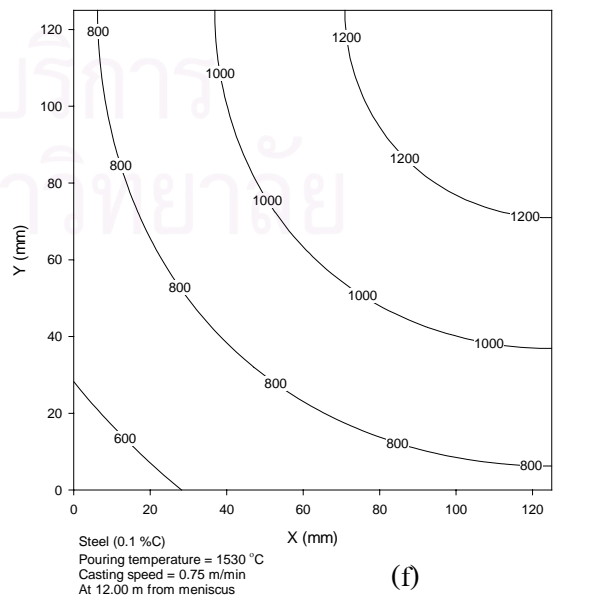
(c)



(d)



(e)



(f)

รูปที่ 4.17 แสดงการกระจายตัวของอุณหภูมิในภาคตัดขวาง ที่ความเร็วในการหล่อ 0.75 m/min

หล่อมีค่าเท่ากับ 0.75 เมตรต่อวินาทีความหนาของเปลือกที่แข็งตัวลดลงเหลือประมาณ 10 มิลลิเมตร เมื่อขึ้นงานผ่านการฉีดด้วยน้ำ (ระยะ 0.62-17.0 เมตรจากเมนิสคัส) ที่ระยะ 6 เมตรจากเมนิสคัสความหนาของเปลือกที่ความเร็ว 0.55 เมตรต่อวินาทีคือ 90 มิลลิเมตร ที่ความเร็วในการหล่อ 0.65 เมตรต่อวินาทีหนาประมาณ 70 มิลลิเมตร และที่ความเร็วในการหล่อ 0.75 เมตรต่อวินาทีหนาประมาณ ประมาณ 50 มิลลิเมตร จากผลการทดลองจึงสามารถสรุปได้ว่า ความเร็วในการหล่อก็มีผลต่อความหนาของเปลือกที่แข็งตัว กล่าวคือที่ตำแหน่งเดียวกันเมื่อเพิ่มความเร็วในงานหล่อจะลดความหนาของเปลือกที่แข็งตัว โดยจะมีผลต่อความหนาของเปลือกตรงบริเวณที่มีการฉีดน้ำมากกว่าบริเวณที่น้ำโลหะแข็งตัวในแบบหล่อ

4.2.3 ผลของความเร็วในการหล่อต่อการแข็งตัวของน้ำโลหะ

รูปที่ 4.18 แสดงผลการคำนวณการกระจายตัวของอุณหภูมิที่แกนกลางโดยใช้ความเร็วในการหล่อ 0.55, 0.65 และ 0.75 เมตรต่อวินาทีตามลำดับ ที่ความเร็วในการหล่อ 0.55 เมตรต่อวินาที น้ำโลหะเริ่มมีการแข็งตัวทันทีที่บริเวณผิว ที่แกนกลางเริ่มมีการแข็งตัวที่ระยะประมาณ 4.5 เมตรจากเมนิสคัส และแข็งตัวเสร็จสิ้นที่ระยะประมาณ 8.5 เมตร ที่ความเร็วในการหล่อ 0.65 เมตรต่อวินาที ที่แกนกลางน้ำโลหะเริ่มแข็งตัวที่ระยะประมาณ 5 เมตร และแข็งตัวเสร็จสิ้นที่ระยะประมาณ 10 เมตร และในการดึงด้วยความเร็ว 0.75 เมตรต่อวินาทีน้ำโลหะเริ่มแข็งตัวที่ระยะประมาณ 6 เมตร และสิ้นสุดการแข็งตัวที่ระยะ ประมาณ 13 เมตร ดังนั้นจากผลการคำนวณพบว่า เมื่อความเร็วในการหล่อเพิ่มขึ้นมีผลทำให้น้ำโลหะเริ่มมีการแข็งตัวและสิ้นสุดการแข็งตัวระยะห่างจากเมนิสคัสที่เพิ่มขึ้น

รูปที่ 4.19 แสดงผลของความเร็วในการหล่อต่อการเย็นตัวของชิ้นงานที่ตำแหน่งผิวชิ้นงาน ตรงบริเวณกึ่งกลางระหว่างผิวชิ้นงานกับแกนกลาง และบริเวณแกนกลางของชิ้นงาน พบว่าที่บริเวณผิวภายในแบบหล่อมีการเย็นตัวอย่างรวดเร็วโดยมีอุณหภูมิออกจากแบบหล่อประมาณ 1200 °C ที่บริเวณนี้ ความเร็วในการหล่อก็มีผลไม่มากนักต่อการกระจายตัวของอุณหภูมิในแบบหล่อ แต่จะมีผลมากเมื่อออกจากแบบหล่อ ในตำแหน่งที่มีการฉีดด้วยน้ำพบว่าที่ตำแหน่งปลายของการฉีดจะมีอุณหภูมิประมาณ 550-650 °C และในช่วงที่เย็นตัวในอากาศจะมีอุณหภูมิประมาณ 650-780 °C จากกราฟพบว่าเมื่อเพิ่มความเร็วในการหล่อจะทำให้อุณหภูมิที่บริเวณผิวสูงขึ้นเช่นเดียวกับที่บริเวณกึ่งกลางระหว่างผิวกับแกนกลาง และที่ตำแหน่งแกนกลาง โดยที่ตำแหน่งแกนกลางความเร็วในการหล่อจะมีผลมากที่สุดต่อการสิ้นสุดในการแข็งตัว

4.2.4 ผลของอุณหภูมิต่อการแข็งตัวของน้ำโลหะ

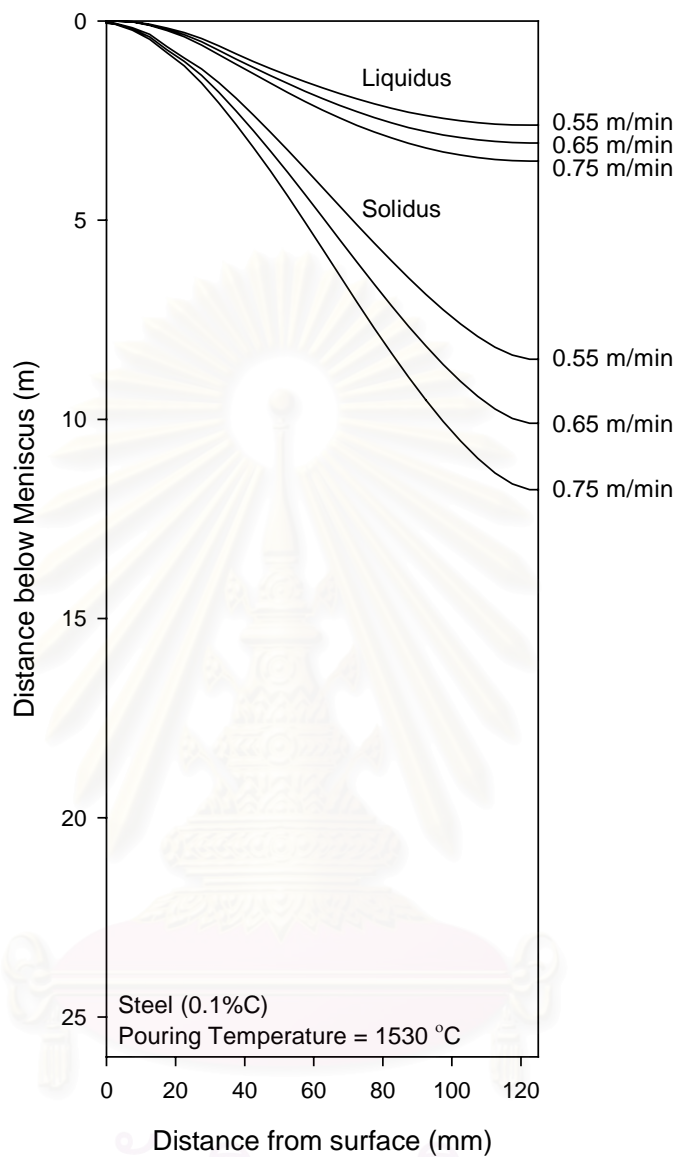
รูปที่ 4.20 แสดงผลของอุณหภูมิในการทดสอบการแข็งตัวของน้ำโลหะ จากผลการคำนวณพบว่า การเพิ่มอุณหภูมิในการทดสอบผลเล็กน้อยต่อจุดเริ่มต้นการแข็งตัวและจุดสิ้นสุดการแข็งตัวของน้ำโลหะ ยกเว้นอุณหภูมิในการทดสอบที่ 1520 °C โดยเพิ่มอุณหภูมิในการทดสอบทำให้จุดที่เริ่มการแข็งตัวห่างจากเมนิสคัสมากขึ้นเล็กน้อยโดยที่ตำแหน่งแกนกลางจะมีจุดเริ่มต้นการแข็งตัวห่างจากเมนิสคัสประมาณ 4 เมตร ในขณะที่จุดสิ้นสุดการแข็งตัวจะคงที่ที่ระยะประมาณ 11 เมตรถึงแม้ว่าอุณหภูมิในการทดสอบเพิ่มขึ้นก็ตาม ดังนั้นอุณหภูมิในการทดสอบจึงไม่มีผลมากนักต่อช่วงการแข็งตัวของน้ำโลหะที่ความเร็วในการหล่อคงที่

4.2.5 ตัวอย่างผลการคำนวณการเย็นตัวสำหรับการหล่อเหล็กแท่งเล็ก

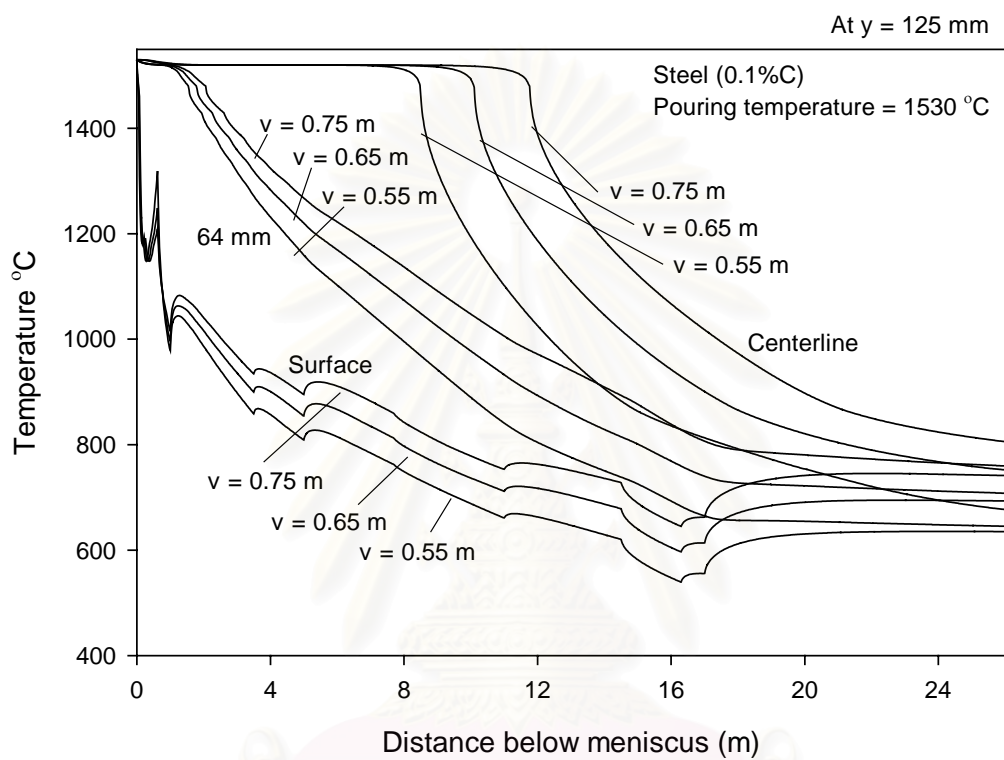
รูปที่ 4.21 แสดงผลการคำนวณการเย็นตัวของชิ้นงานในการหล่อแท่งเล็กที่ระยะต่างๆ ห่างจากผิวโดยมีอุณหภูมิเทเท่ากับ 1530 °C และ มีความเร็วในการหล่อเท่ากับ 1.50 เมตรต่อนาที จากรูปพบว่าที่ผิวมีการเย็นตัวลดลงอย่างรวดเร็วโดยมีอุณหภูมิออกจากแบบหล่อประมาณ 1250 °C และอุณหภูมิต่ำสุดประมาณ 1160 °C ภายหลังจากการฉีดน้ำ และเย็นตัวในอากาศที่อุณหภูมิประมาณ 1200 °C จากการแข็งตัวของน้ำโลหะที่ตำแหน่งแกนกลางพบว่าปอน้ำโลหะลึกประมาณ 8 เมตรและที่แกนกลางภายหลังจากที่มีการแข็งตัวหมดแล้วอุณหภูมิจะลดลงอย่างรวดเร็ว ทั้งนี้การคำนวณการกระจายตัวของอุณหภูมิ การศึกษาผลของพารามิเตอร์เช่น ความเร็วในการหล่อ อุณหภูมิเทต่อความหนาของเปลือกที่แข็งตัวที่ตำแหน่งต่างๆภายในแบบหล่อ สามารถใช้แบบจำลองการคำนวณที่ประดิษฐ์นี้เป็นแนวทางเบื้องต้นในการพิจารณาสำหรับงานหล่อในทางปฏิบัติได้

4.3 ตัวอย่างผลการคำนวณการเย็นตัวสำหรับการหล่ออะลูมิเนียม

รูปที่ 4.22 แสดงผลการคำนวณการเย็นตัวของชิ้นงานในการหล่อแท่งอะลูมิเนียม โดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสามมิติ ซึ่งผลการคำนวณใช้โปรแกรม Micro AVS เพื่อช่วยในการแสดงผล ซึ่งความแม่นยำของการคำนวณขึ้นอยู่กับ เงื่อนไขขอบเขต และคุณสมบัติทางความร้อนที่ให้กับโปรแกรม



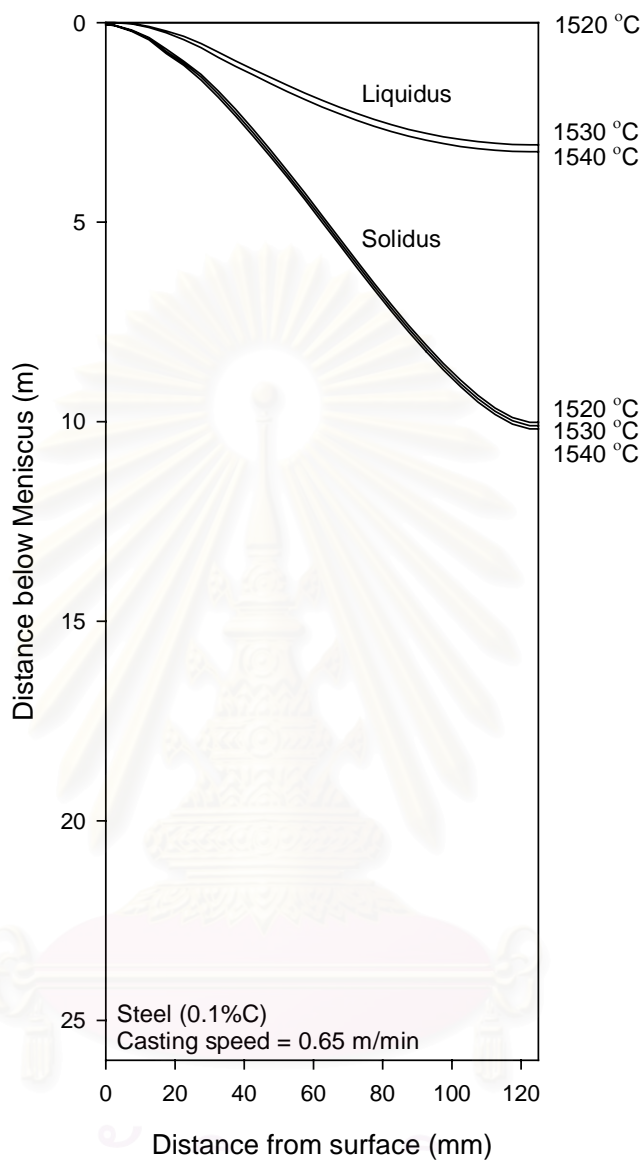
รูปที่ 4.18 ผลของความเร็วในการหล่อต่อช่วงการแข็งตัวของน้ำโลหะที่ระยะต่างๆจาก
 เมนิสคัส คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ



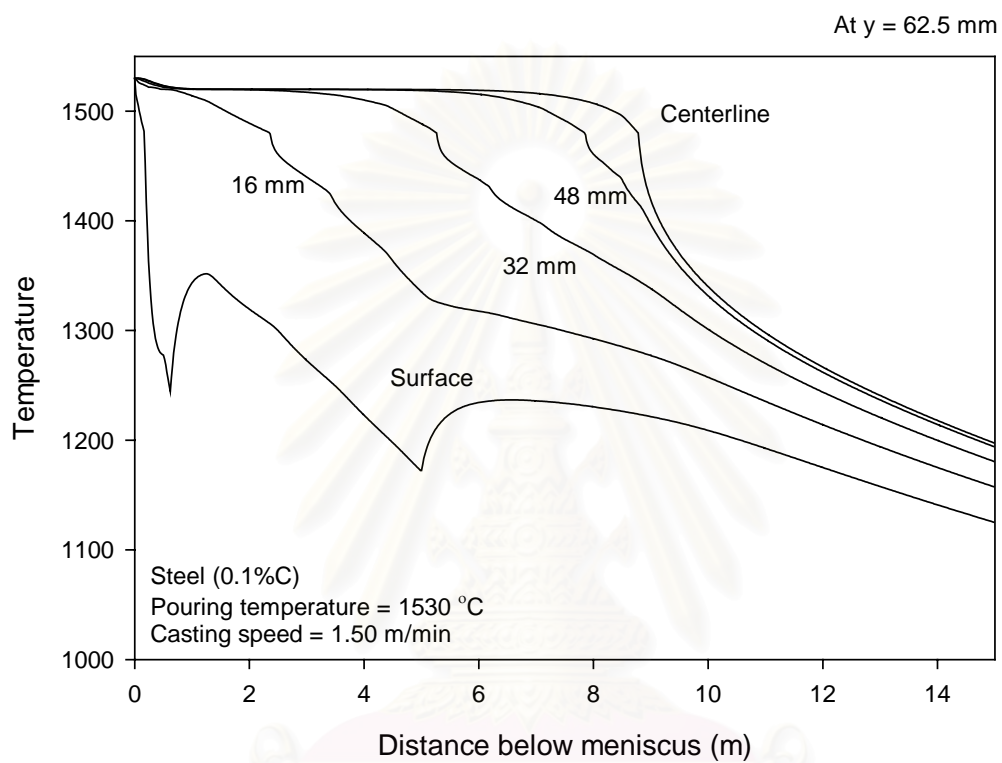
รูปที่ 4.19 ผลของความเร็วในการหล่อต่อช่วงการเย็นตัวของชิ้นงานที่ระยะต่างๆจากเม

นีสคัส คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

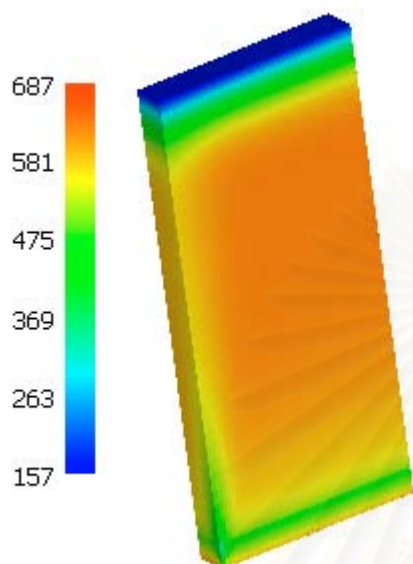


รูปที่ 4.20 ผลของอุณหภูมิเทต่อช่วงการแข็งตัวของน้ำโลหะที่ระยะต่างๆจากเมนิสคัส
คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสองมิติ



รูปที่ 4.21 แสดงการกระจายตัวของอุณหภูมิของชิ้นงานหล่อแท่งเล็กที่ตำแหน่งต่างๆจากเมนิสคัส

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.22 แสดงการกระจายตัวของอุณหภูมิของชิ้นงานหล่อแท่งอะลูมิเนียมที่คำนวณโดยใช้แบบจำลองที่คิดการถ่ายเทความร้อนในสามมิติ โดยใช้โปรแกรม Micro AVS แสดงผล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

สรุปผลการวิจัย

สรุปผลการวิจัย

- 1) โปรแกรมคอมพิวเตอร์สามโปรแกรมได้มีการพัฒนาเพื่อศึกษาการกระจายตัวของ อุณหภูมิในระนาบหนึ่งมิติและสองมิติสำหรับเหล็กกล้า และสามมิติสำหรับโลหะนอกกลุ่มเหล็ก สำหรับการหล่อแบบต่อเนื่องได้
- 2) เมื่อเปรียบเทียบระหว่างโปรแกรมการคำนวณกับงานวิจัยที่ใช้เปรียบเทียบ ให้ผล ความแม่นยำในระดับดี
- 3) จากการคำนวณโดยการใช้โปรแกรมพบว่าพารามิเตอร์ที่สำคัญต่อการหล่อแบบต่อ เนื่องคือ ความเร็วในการตั้ง อุณหภูมิเท และการเย็นตัว
- 4) โปรแกรมทั้งสามสามารถแสดงผลของพารามิเตอร์ที่สำคัญต่อ การกระจายตัวของ อุณหภูมิที่ตำแหน่งต่างๆได้ ทำให้สามารถเข้าใจและนำไปประยุกต์ใช้งานจริงได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Thomas, B.G. Brimacombe Lecture. Modeling of the Continuous Casting of Steel Past Present and Future 59th Electric Furnace Conf., Phoenix, AZ, ISS, Warrendale, PA, (2001)
2. Incropera, F. P.; and DeWitt, D. P. Introduction to Heat Transfer. 2nd ed. Singapore: John Wiley & Sons, 1990.
3. Brimacombe, J. K. Design of Continuous Casting Machines Based on a Heat-Flow Analysis: State-of-the-Art Review. Can.Met.Quart vol.15 (1976): 163-175.
4. Brimacombe, J. K.; Samarasekera, I.V.; and Lait, J.E. Continuous Casting Heat Flow Solidification and Crack Formation. Vol 2. United State of America: BookCrafters, 1984.
5. Mikikar, E. A. Mathematical Heat Transfer Model for Solidification of Continuously Cast Steel Slabs. Trans. TMS-AIME vol. 239 (1967): pp. 1747-1753.
6. Savage, J.; and Pritchard, W. H. The Problem of Rupture of the Billet in The Continuous Casting of Steel. J. Iron Steel Inst. Vol. 178 (1954): 269-277
7. Tong, C.; and Yokomizo, K. Analysis of Thermal Stress and Deformation in Continuous Casting Proceeding of 2nd Pacific Rim International Conference on Modeling of Casting and Solidification Processes (1995):pp. 51-59.
8. Hong, C. P.; Umeda, T.; and Kimura, Y. Numerical Models for Casting Solidification: Part2. Application of the Boundary Element Method to Solidification Problems. Metallurgical Transaction vol. 15B (March 1984): 101-107



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ไฟล์ 1d.c

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

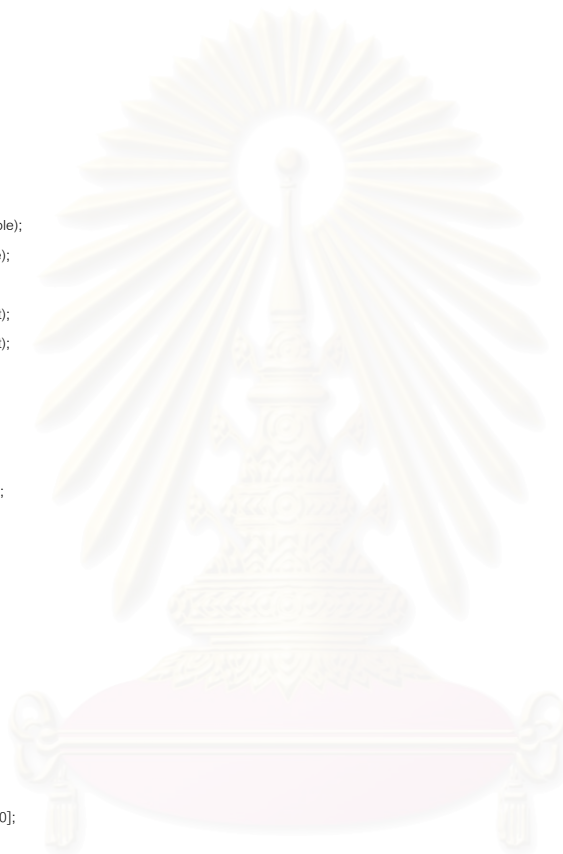
double fk(double);
double fb(double);
double frho(double);
double fcp(double);
double check_temp(double,double);
int out1(int,double,double,double);
int out2(double,double,double);
int out3(double,double,double,int);
int out4(double,double,double,int);
double shell_thickness(void);

double liquid_temp,solid_temp;
double temp[100],tempnew[100];
int post[20];
int n;
int dn;
double dx,lx;
double z,dz,zstop,v;
double t,dt,tstop;
double rho,qout;
double zigma,epsilon;
double tair,tw,pour_temp;
double zmold,zspray[10];
double tendmold,tspray[10],hs[10];
int num_spray;
int num_show;
double zshow1,zshow2,dzshow;
double tshow1,tshow2,dtshow;
int solid_show,tshow1_stat,tshow2_stat;
double alpha1,alpha2;

main()
{
    int j,i;

    lx = 0.125;
    n = 25;
    dx = lx/(n);
    dt = 0.1;
    rho = 7850;
    pour_temp = 1520;
    liquid_temp = 1520;
    solid_temp = 1480;
    tw = 25;

```



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```

tair = 100;
zigma = 5.669e-8;
epsilon = 0.8;
// v = 0.010833;
v = .75/60.;
dz = v*dt;
zshow1 = 0.625;
zshow2 = 17;
zmold = 0.625;
num_spray = 8;
zspray[0] = 1;
zspray[1] = 3.5;
zspray[2] = 5;
zspray[3] = 7.7;
zspray[4] = 11;
zspray[5] = 14.5;
zspray[6] = 16.3;
zspray[7] = 17;
zstop = 26;
dzshow = 0.02;
tshow1 = zshow1/v;
tshow2 = zshow2/v;
tshow1_stat = 0;
tshow2_stat = 0;
solid_show = 0;

tendmold = zmold/v;
tspray[0] = zspray[0]/v;hs[0] = 824.567;
tspray[1] = zspray[1]/v;hs[1] = 482.645;
tspray[2] = zspray[2]/v;hs[2] = 410.539;
tspray[3] = zspray[3]/v;hs[3] = 311.684;
tspray[4] = zspray[4]/v;hs[4] = 330.292;
tspray[5] = zspray[5]/v;hs[5] = 240.741;
tspray[6] = zspray[6]/v;hs[6] = 381.464;
tspray[7] = zspray[7]/v;hs[7] = 226.785;
tstop = zstop/v;

```

```
// init for output parameter
```

```

dtshow = dzshow/v;
num_show = 7;
post[0] = 0; // for sub surface
post[1] = (96.)/1000./dx; // 96mm from surface
post[2] = (64.)/1000./dx; // 64mm from surface
post[3] = (32.)/1000./dx; // 32mm from surface
post[4] = n-1; // for sub centerline
post[5] = n; // for centerline
post[6] = n+1; // for surface
dn = 1;

```

```

t=0;
for(j=0;j<n+2;j++)
    temp[j] = pour_temp;
out1(num_show,t,dtshow,tstop);
out2(t,dt,tstop);
out3(t,dt,tstop,dn);

```



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```

out4(t,dtshow,tstop,dn);
for(t=dt;t<tstop;t=t+dt)
{
    for(j=1;j<n-1;j++)
    {
        alpha1 = (fk(temp[j])+fk(temp[j-1]))/2./rho(temp[j])/fcp(temp[j]);
        alpha2 = (fk(temp[j])+fk(temp[j+1]))/2./rho(temp[j])/fcp(temp[j]);
        tempnew[j] = (1-alpha1*dt/dx/dx-alpha2*dt/dx/dx)*temp[j]
            + alpha1*dt/dx/dx*temp[j-1] + alpha2*dt/dx/dx*temp[j+1];
        tempnew[j] = check_temp(temp[j],tempnew[j]);
    }

    j = n-1;
    alpha1 = (fk(temp[n-1])+fk(temp[n-2]))/2./rho(temp[n-1])/fcp(temp[n-1]);
    tempnew[n-1] = (1-alpha1*dt/dx/dx)*temp[n-1] + alpha1*dt/dx/dx*temp[n-2];
    tempnew[j] = check_temp(temp[j],tempnew[j]);

    j = 0;
    if(t>=0&&t<=tendmold)
        qout = 3.154*(850000.-100000.*sqrt((t-dt)))/time unit use in equation is second// Savage and Pritchard
        qout = 3.154*(850000.*pow(2.71828,(-.0386*(t-dt)))+4000.*(t-dt)-36000.)/Kraimer and Tarmann
    if(t>tendmold&&t<=tspray[0])
        qout = hs[0]*(temp[0]-tw);
    for(l=1;l<num_spray;l++)
    {
        if(t>=tspray[l-1]&&t<=tspray[l])
            qout = hs[l]*(temp[0]-tw);
    }
    if(t>=tspray[num_spray-1])
        qout = zigma*epsilon*(pow(temp[0],4)-pow(tair,4));
    alpha1 = (fk(temp[0])+fk(temp[1]))/2./rho(temp[0])/fcp(temp[0]);
    tempnew[0] = (1-alpha1*dt/dx/dx)*temp[0] + alpha1*dt/dx/dx*temp[1]
        - qout*dt/rho(temp[0])/fcp(temp[0])/dx;
    tempnew[j] = check_temp(temp[j],tempnew[j]);

    j = n; // for centerline
    tempnew[j] = tempnew[n-1];
    j = n+1; // for surface
    tempnew[j] = tempnew[0]+(tempnew[0]-tempnew[1])/2.;
    for(j=0;j<n+2;j++)
        temp[j] = tempnew[j];

// Show for some results on screen
    if(t>=tshow1&&tshow1_stat==0)
    {
        printf("Surface Temperature at below mold = %f\n",tempnew[0]);
        printf("Shell Thickness at below mold = %f m\n",shell_thickness());
        tshow1_stat=1;
    }
    else if(t>=tshow2&&tshow2_stat==0)
    {
        printf("Surface Temperature at below spray = %f\n",tempnew[0]);
        printf("Shell Thickness at below spray = %f m\n",shell_thickness());
        tshow2_stat=1;
    }
    if(tempnew[n-1]<solid_temp&&solid_show==0)
    {
        printf("Solidification time = %f sec = %f m\n",t,v*t);
    }
}

```

```

        solid_show=1;
    }

// End Show

// Log Data to file
    out1(num_show,t,dtshow,tstop); // Log for Tsurface and Tcenterline
    out2(t,dt,tstop); // Log for Shell Thickness
    out3(t,dt,tstop,dn); // Log for Contour Graph
    out4(t,dtshow,tstop,dn); // Log for All Point Temperature
}
out1(num_show,t,dtshow,tstop);
out2(t,dt,tstop);
out3(t,dt,tstop,dn);
out4(t,dtshow,tstop,dn);
}

double check_temp(double temp1,double temp2)
{
    double cpe;
    double qlat,tliq,tsolid;

    qlat = 2.67e5;
    tliq = liquid_temp;
    tsolid = solid_temp;

    if (temp1>tliq&&temp2<tliq)
    {
        cpe = fcp(temp1) + qlat/(tliq-tsolid);
        temp2 = tliq + (frho(temp1)*fcp(temp1)/frho(temp2)/cpe)*(temp2-tliq);
        return(temp2);
    }
    else if (temp1>tsolid&&temp1<tliq&&temp2<tsolid)
    {
        cpe = fcp(temp1);
        temp2 = tsolid + frho(temp1)*cpe/frho(temp2)/fcp(temp2)*(temp2-tsolid);
        return(temp2);
    }
    else
    {
        return(temp2);
    }
}

double fk(double temp)
{
    double tsolid,tliq;

    tsolid = solid_temp*9./5. + 32 + 459.67;
    tliq = liquid_temp*9./5. + 32 + 459.67;

```

```

temp = temp*9./5. + 32.;
temp = temp+459.67;
if(temp>=0&&temp<=2050)
    return(1.7307*(43.8-0.0137*temp));
else if(temp>2050&&temp<=tliq)
    return(1.7307*(8.0+0.0037*temp));
else if(temp>tliq)
    return(1.7307*(150));
else
{
    printf("Error! Out of temperature range in '\tk'\n");
    exit(0);
}
}

double frho(double temp)
{
    return(rho);
}

double fcp(double temp)
{
    double tsolid,tliq;

    tsolid = solid_temp*9./5. + 32 + 459.67;
    tliq = liquid_temp*9./5. + 32 + 459.67;
    temp = temp*9./5. + 32.;
    temp = temp+459.67;
    if(temp>=900&&temp<=1850)
    {
        temp = temp-459.67;
        return(4186.8*(0.0758125+0.23177e-3*temp+pow(0.345312e-6*temp,2)+pow(0.205729e-4*temp,3)));
    }
    else if(temp>1850&&temp<=2050)
        return(4186.8*(2.21-0.001*temp));
// else if(temp>2050&&temp<=3200)
// else if(temp>2050&&temp<=tsolid)
//         return(4186.8*(0.1242+0.0000125*temp));
// else if(temp>3200&&temp<=3300)
// else if(temp>tsolid&&temp<=tliq)
//         return(4186.8*(1.88));
// else if(temp>3300)
// else if(temp>tliq)
//         return(4186.8*(0.165));
else
{
    return(4186.8*(0.15));
}
//         printf("Error! Out of temperature range in '\tfcpl'\n");
//         exit(0);
}
}

```

```
int out1(int num_show,double t,double dtshow,double tend)// Log for Tsurface and Tcenterline
```

```
{
    static double tshow=0;
    static int init_stat=0;
    static int close_stat=0;
    static FILE *output1;
    int i;

    if(init_stat==0)
    {
        output1 = fopen("output1.csv","w+");
        fprintf(output1,"This is output1\n");
        init_stat=1;
    }
    if(t>=tshow&& t<=tend)
    {
        fprintf(output1,"%f",v*t);
        for(i=0;i<num_show;i++)
        {
            fprintf(output1,"%f",temp[post[i]]);
        }
        fprintf(output1,"\n");
        tshow = tshow + dtshow;
    }
    else if(t>tend&&close_stat==0)
    {
        fclose(output1);
        close_stat=1;
    }
    return(0);
}
```

```
int out2(double t,double dtshow,double tend)// Log for Shell Thickness
```

```
{
    static double tshow=0;
    static int init_stat=0;
    static int close_stat=0;
    static double thickness,oldthickness;
    static FILE *output2;

    if(init_stat==0)
    {
        oldthickness = -1.;
        output2 = fopen("output2.csv","w+");
        fprintf(output2,"This is output2\n");
        init_stat=1;
    }
    if(t>=tshow&& t<=tend)
    {
        thickness = shell_thickness();
        if(thickness>oldthickness||thickness<oldthickness)
        {
            fprintf(output2,"%f",v*t);
            fprintf(output2,"%f\n",shell_thickness());
            oldthickness = thickness;
        }
        tshow = tshow + dtshow;
    }
    else if(t>tend&&close_stat==0)
    {
        fclose(output2);
        close_stat=1;
    }
}
```



```

    }
    return(0);
}

double shell_thickness(void)
{
    int j;

    for(j=0;j<n;j++)
        if(temp[j]>solid_temp)
            return(j*dx);

    return(x);
}

int out3(double t,double dtshow,double tend,int dn)// Log for Contour Graph
{
    static double tshow=0;
    static int init_stat=0;
    static int close_stat=0;
    static FILE *output3;
    static double tempshow[10];
    static int stat[100][10];
    static int num_temp;
    static double x[100][10];

    int j,i;

    tempshow[0] = 1520;
    tempshow[1] = 1480;
    tempshow[2] = 2500;
    tempshow[3] = 1900;
    tempshow[4] = 1700;
    tempshow[5] = 1400;
    tempshow[6] = 900;
    num_temp = 2;

    if(init_stat==0)
    {
        output3 = fopen("output3.csv","w+");
        fprintf(output3,"This is output3\n");
        for(i=0;i<num_temp;i++)
        {
            fprintf(output3,"%f",tempshow[i]);
        }
        fprintf(output3,"\n");
        for(j=0;j<n+2;j++)
            for(i=0;i<num_temp;i++)
                stat[j][i] = 0;
        init_stat = 1;
    }
    if(t>=tshow&& t<=tend)
    {
        for(j=0;j<n+2;j++)
            for(i=0;i<num_temp;i++)
                if(temp[j]<=tempshow[i]&&stat[j][i]==0)
                {
                    x[j][i] = v*t;
                    stat[j][i] = 1;
                }
                else if(temp[j]>=tempshow[i]&&stat[j][i]==1)
                {
                    printf("Error in reheat contour\n");
                }
    }
}

```

```

    }
}
else if(t>tend&&close_stat==0)
{
    j = n+1;
    fprintf(output3,"%f,0.);
    for(l=0;l<num_temp;l++)
        fprintf(output3,"%f",x[l][l]);
    fprintf(output3,"\n");
    for(j=0;j<n;j=j+dn)// **j = j+dn
    {
        fprintf(output3,"%f," dx+dx/2.);
        for(l=0;l<num_temp;l++)
            fprintf(output3,"%f",x[l][l]);
        fprintf(output3,"\n");
    }
    j = n;
    fprintf(output3,"%f",x);
    for(l=0;l<num_temp;l++)
        fprintf(output3,"%f",x[l][l]);
    fprintf(output3,"\n");
    fclose(output3);
    close_stat = 1;
}
return(0);
}

```

```

int out4(double t,double dtshow,double tend,int dn)// Log for All Point Temperature

```

```

{
    static double tshow=0;
    static int init_stat=0;
    static int close_stat=0;
    static FILE *output4;
    int j;

    if(init_stat==0)
    {
        output4 = fopen("output4.csv","w+");
        fprintf(output4,"This is output4\n");
        init_stat=1;
    }
    if(t>=tshow&&t<=tend)
    {
        fprintf(output4,"%f,v,t);
        j = n+1;
        fprintf(output4,"%f",temp[j]);
        for(j=0;j<n;j=j+dn)// **j = j+dn
        {
            fprintf(output4,"%f",temp[j]);
        }
        j = n;
        fprintf(output4,"%f",temp[j]);
        fprintf(output4,"\n");
        tshow = tshow + dtshow;
    }
}

```

```
else if(!tend&&close_stat==0)
{
    fclose(output4);
    close_stat=1;
}
return(0);
}
```



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ไฟล์ 2d.c

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

typedef struct
{
    int x,y;
} position;

double fk(double);
double fb(double);
double frho(double);
double fcp(double);
double check_temp(double,double);
int out1(int,double,double,double);
int out2(double,double,double);
int out3(double,double,double,int,int,int);
int out4(double,double,double,int);
double shell_thickness(int,int);
int output_inp(int,int,int,double,double,double);

double liquid_temp,solid_temp;
double temp[100][200],tempnew[100][200],inptemp[100][200][2000];
position pos[20];
int nx,ny,nzinp=0;
int dn;
double dx,lx,dy,ly;
double z,dz,zstop,v;
double t,dt,tstop;
double rho,qout;
double zigma,epsilon;
double tair,tw,pour_temp;
double zmold,zspray[10];
double tendmold,tspray[10],hs[10];
int num_spray;
int num_show;
double zshow1,zshow2,dzshow;
double tshow1,tshow2,dtshow;
int solid_show,tshow1_stat,tshow2_stat;
double alpha1,alpha2,alpha3,alpha4;

main()
{
    int i,j,l;

    lx = 0.0625;
    ly = 0.0625;
    nx = 12;
    ny = 12;
    dx = lx/(nx);
    dy = ly/(ny);
    dt = 0.1;

```

```

rho = 7850;
pour_temp = 1530;
liquid_temp = 1500;
solid_temp = 1470;
tw = 25;
lair = 100;
zigma = 5.669e-8;
epsilon = 0.8;
v = 1.5/60.;
dz = v*dt;
zshow1 = 0.625;
zshow2 = 17;
zmold = 0.625;
num_spray = 1;
zspray[0] = 5;
zspray[1] = 3.5;
zspray[2] = 5;
zspray[3] = 7.7;
zspray[4] = 11;
zspray[5] = 14.5;
zspray[6] = 16.3;
zspray[7] = 17;
zstop = 15;
dzshow = 0.02;
tshow1 = zshow1/v;
tshow2 = zshow2/v;
tshow1_stat = 0;
tshow2_stat = 0;
solid_show = 0;

tendmold = zmold/v;
tspray[0] = zspray[0]/v;hs[0] = 300;
tspray[1] = zspray[1]/v;hs[1] = 482.645;
tspray[2] = zspray[2]/v;hs[2] = 410.539;
tspray[3] = zspray[3]/v;hs[3] = 311.684;
tspray[4] = zspray[4]/v;hs[4] = 330.292;
tspray[5] = zspray[5]/v;hs[5] = 240.741;
tspray[6] = zspray[6]/v;hs[6] = 381.464;
tspray[7] = zspray[7]/v;hs[7] = 226.785;
tstop = zstop/v;
// init for output parameter

dtshow = dzshow/v;
num_show = 7;
post[0].x = 0; // for sub surface
post[1].x = (16.)/1000./dx; // 96mm from surface
post[2].x = (32.)/1000./dx; // 64mm from surface
post[3].x = (48.)/1000./dx; // 32mm from surface
post[4].x = nx-1; // for sub centerline
post[5].x = nx; // for centerline
post[6].x = nx+1; // for surface
post[0].y = ny-1;
post[1].y = ny-1;
post[2].y = ny-1;
post[3].y = ny-1;
post[4].y = ny-1;
post[5].y = ny-1;
post[6].y = ny-1;
dn = 1;

t=0;
for(j=0;j<ny+2;j++)

```

```

for(i=0;i<nx+2;i++)
    temp[i][j] = pour_temp;
out1(num_show,t,dtshow,tstop);
out2(t,dt,tstop);
out3(t,dt,tstop,dn,nx-1,2);
out4(t,dtshow,tstop,dn);
for(t=dt;t<tstop;t=t+dt)
{
    for(j=1;j<ny-1;j++)
    {
        for(i=1;i<nx-1;i++)
        {
            alpha1 = (fk(temp[i][j])+fk(temp[i-1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
            alpha2 = (fk(temp[i][j])+fk(temp[i+1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
            alpha3 = (fk(temp[i][j])+fk(temp[i][j-1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
            alpha4 = (fk(temp[i][j])+fk(temp[i][j+1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
            tempnew[i][j] = (1-alpha1*dt/dx-alpha2*dt/dx
                -alpha3*dt/dy-alpha4*dt/dy)*temp[i][j]
                + alpha1*dt/dx*dx*temp[i-1][j] + alpha2*dt/dx*dx*temp[i+1][j]
                + alpha3*dt/dy/dy*temp[i][j-1] + alpha4*dt/dy/dy*temp[i][j+1];
            tempnew[i][j] = check_temp(temp[i][j],tempnew[i][j]);
        }
        i = nx-1;
        alpha1 = (fk(temp[nx-1][j])+fk(temp[nx-2][j]))/2./rho(temp[nx-1][j])/fcp(temp[nx-1][j]);
        alpha3 = (fk(temp[i][j])+fk(temp[i][j-1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
        alpha4 = (fk(temp[i][j])+fk(temp[i][j+1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
        tempnew[i][j] = (1-alpha1*dt/dx
            -alpha3*dt/dy-alpha4*dt/dy)*temp[i][j]
            + alpha1*dt/dx*dx*temp[i-1][j]
            + alpha3*dt/dy/dy*temp[i][j-1] + alpha4*dt/dy/dy*temp[i][j+1];
        i = 0;
        if(t>=0&&t<=tendmold)
            qout = 3.154*(850000.-100000.*sqrt((t-dt)))/time unit use in equation is second// Savage and Pritchard
            qout = 3.154*(850000.*pow(2.71828,(-.0386*(t-dt)))+4000.*(t-dt)-36000.)/Kraimer and Tarmann
        if(t>tendmold&&t<=tspray[0])
            qout = hs[0]*(temp[i][j]-tw);
        for(l=1;l<num_spray;l++)
        {
            if(t>tspray[l-1]&&t<=tspray[l])
                qout = hs[l]*(temp[i][j]-tw);
        }
        if(t>tspray[num_spray-1])
            qout = zigma*epsilon*(pow(temp[0][j],4)-pow(tair,4));
        alpha2 = (fk(temp[0][j])+fk(temp[1][j]))/2./rho(temp[0][j])/fcp(temp[0][j]);
        alpha3 = (fk(temp[i][j])+fk(temp[i][j-1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
        alpha4 = (fk(temp[i][j])+fk(temp[i][j+1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
        tempnew[0][j] = (1-alpha2*dt/dx-alpha3*dt/dy-alpha4*dt/dy)*temp[0][j]
            + alpha2*dt/dx*dx*temp[1][j] + alpha3*dt/dy/dy*temp[0][j-1] + alpha4*dt/dy/dy*temp[0][j+1]
            - qout*dt/rho(temp[0][j])/fcp(temp[0][j])/dx;
        tempnew[i][j] = check_temp(temp[i][j],tempnew[i][j]);
        i = nx; // for centerline
        tempnew[nx][j] = tempnew[nx-1][j];
        i = nx+1; // for surface
        tempnew[i][j] = tempnew[0][j]+(tempnew[0][j]-tempnew[1][j])/2.;
    }
}
j = ny-1;
{
    i = 0;
    if(t>=0&&t<=tendmold)
        qout = 3.154*(850000.-100000.*sqrt((t-dt)))/time unit use in equation is second// Savage and Pritchard
        qout = 3.154*(850000.*pow(2.71828,(-.0386*(t-dt)))+4000.*(t-dt)-36000.)/Kraimer and Tarmann
    if(t>tendmold&&t<=tspray[0])

```

```

        qout = hs[0]*(temp[0][j]-tw);
for(i=1;i<num_spray;i++)
{
    if(t>tspray[i-1]&&t<=tspray[i])
        qout = hs[i]*(temp[0][j]-tw);
}
if(t>tspray[num_spray-1])
    qout = zigma*epsilon*(pow(temp[0][j],4)-pow(tair,4));
alpha2 = (fk(temp[i][j])+fk(temp[i+1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
alpha3 = (fk(temp[i][j])+fk(temp[i][j-1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
tempnew[0][j] = (1-alpha2*dt/dx/dx-alpha3*dt/dy/dy)*temp[0][j]
    + alpha2*dt/dx/dx*temp[1][j] - qout*dt/rho(temp[0][j])/fcp(temp[0][j])/dx
    + alpha3*dt/dy/dy*temp[0][j-1];
tempnew[i][j] = check_temp(temp[i][j],tempnew[i][j]);
for(i=1;i<nx-1;i++)
{
    alpha1 = (fk(temp[i][j])+fk(temp[i-1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
    alpha2 = (fk(temp[i][j])+fk(temp[i+1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
    alpha3 = (fk(temp[i][j])+fk(temp[i][j-1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
    tempnew[i][j] = (1-alpha1*dt/dx/dx-alpha2*dt/dx/dx-alpha3*dt/dy/dy)*temp[i][j]
        + alpha1*dt/dx/dx*temp[i-1][j] + alpha2*dt/dx/dx*temp[i+1][j]
        + alpha3*dt/dy/dy*temp[i][j-1];
    tempnew[i][j] = check_temp(temp[i][j],tempnew[i][j]);
}

i = nx-1;
alpha1 = (fk(temp[i][j])+fk(temp[i-1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
alpha3 = (fk(temp[i][j])+fk(temp[i][j-1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
tempnew[i][j] = (1-alpha1*dt/dx/dx-alpha3*dt/dy/dy)*temp[i][j]
    + alpha1*dt/dx/dx*temp[i-1][j] + alpha3*dt/dy/dy*temp[i][j-1];
tempnew[i][j] = check_temp(temp[i][j],tempnew[i][j]);
i = nx;
tempnew[i][j] = tempnew[i-1][j];
i = nx+1;
tempnew[i][j] = tempnew[0][j]+(tempnew[0][j]-tempnew[1][j])/2.;
}

j = 0;
{
    i = 0;
    if(t>=0&&t<=tendmold)
        qout = 3.154*(850000.-100000.*sqrt((t-dt)))/time unit use in equation is second// //Savage and Pritchard
        qout = 3.154*(850000.*pow(2.71828,-.0386*(t-dt))+4000.*(t-dt)-36000.)/Krainer and Tarmann
    if(t>tendmold&&t<=tspray[0])
        qout = hs[0]*(temp[i][j]-tw);
    for(i=1;i<num_spray;i++)
    {
        if(t>tspray[i-1]&&t<=tspray[i])
            qout = hs[i]*(temp[i][j]-tw);
    }
    if(t>tspray[num_spray-1])
        qout = zigma*epsilon*(pow(temp[i][j],4)-pow(tair,4));
    alpha2 = (fk(temp[i][j])+fk(temp[i+1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
    alpha4 = (fk(temp[i][j])+fk(temp[i][j+1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
    tempnew[0][j] = (1-alpha2*dt/dx/dx-alpha4*dt/dy/dy)*temp[0][j]
        + alpha2*dt/dx/dx*temp[1][j] - qout*dt/rho(temp[0][j])/fcp(temp[0][j])/dx
        + alpha4*dt/dy/dy*temp[0][j+1] - qout*dt/rho(temp[0][j])/fcp(temp[0][j])/dy;
    tempnew[i][j] = check_temp(temp[i][j],tempnew[i][j]);
    for(i=1;i<nx-1;i++)
    {
        if(t>=0&&t<=tendmold)

```

```

Pritchard
//
qout = 3.154*(850000.-100000.*sqrt((t-dt)))/time unit use in equation is second// //Savage and

qout = 3.154*(850000.*pow(2.71828,(-.0386*(t-dt)))+4000.*(t-dt)-36000.)/Kraimer and Tarmann

if(t>tendmold&&t<=tspray[0])
    qout = hs[0]*(temp[i][j]-tw);
for(i=1;i<num_spray;i++)
{
    if(t>tspray[i-1]&&t<=tspray[i])
        qout = hs[i]*(temp[i][j]-tw);
}
if(t>tspray[num_spray-1])
    qout = zigma*epsilon*(pow(temp[i][j],4)-pow(tair,4));
alpha1 = (fk(temp[i][j])+fk(temp[i-1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
alpha2 = (fk(temp[i][j])+fk(temp[i+1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
alpha4 = (fk(temp[i][j])+fk(temp[i][j+1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
tempnew[i][j] = (1-alpha1*dt/dx/dx-alpha2*dt/dx/dx-alpha4*dt/dy/dy)*temp[i][j]
    + alpha1*dt/dx/dx*temp[i-1][j] + alpha2*dt/dx/dx*temp[i+1][j]
    + alpha4*dt/dy/dy*temp[i][j+1] - qout*dt/rho(temp[i][j])/fcp(temp[i][j])/dy;
tempnew[i][j] = check_temp(temp[i][j],tempnew[i][j]);
}
i = nx-1;
if(t>=0&&t<=tendmold)
    qout = 3.154*(850000.-100000.*sqrt((t-dt)))/time unit use in equation is second// //Savage and Pritchard
//
qout = 3.154*(850000.*pow(2.71828,(-.0386*(t-dt)))+4000.*(t-dt)-36000.)/Kraimer and Tarmann

if(t>tendmold&&t<=tspray[0])
    qout = hs[0]*(temp[i][j]-tw);
for(i=1;i<num_spray;i++)
{
    if(t>tspray[i-1]&&t<=tspray[i])
        qout = hs[i]*(temp[i][j]-tw);
}
if(t>tspray[num_spray-1])
    qout = zigma*epsilon*(pow(temp[i][j],4)-pow(tair,4));
alpha1 = (fk(temp[i][j])+fk(temp[i-1][j]))/2./rho(temp[i][j])/fcp(temp[i][j]);
alpha4 = (fk(temp[i][j])+fk(temp[i][j+1]))/2./rho(temp[i][j])/fcp(temp[i][j]);
tempnew[i][j] = (1-alpha1*dt/dx/dx-alpha4*dt/dy/dy)*temp[i][j]
    + alpha1*dt/dx/dx*temp[i-1][j] + alpha4*dt/dy/dy*temp[i][j+1]
    - qout*dt/rho(temp[i][j])/fcp(temp[i][j])/dy;
tempnew[i][j] = check_temp(temp[i][j],tempnew[i][j]);
i = nx;
tempnew[i][j] = tempnew[i-1][j];
i = nx+1;
tempnew[i][j] = tempnew[0][j]+(tempnew[0][j]-tempnew[1][j])/2.;
}
j = ny; // for centerline
{
    for(i=1;i<nx-1;i++)
        tempnew[i][j] = tempnew[i][j-1];
    i = 0;
    tempnew[i][j] = tempnew[i][j-1];
    i = nx-1;
    tempnew[i][j] = tempnew[i][j-1];
    i = nx;
    tempnew[i][j] = tempnew[i][j-1];
    i = nx+1;
    tempnew[i][j] = tempnew[i][j-1];
}
j = ny+1; // for surface
{
    for(i=1;i<nx-1;i++)
        tempnew[i][j] = tempnew[i][j-1]+(tempnew[i][j-1]-tempnew[i][j-2])/2.;
    i = 0;

```



```

        tempnew[i][j] = tempnew[i][0]+(tempnew[i][0]-tempnew[i][1])/2.;
    i = nx-1;
        tempnew[i][j] = tempnew[i][0]+(tempnew[i][0]-tempnew[i][1])/2.;
    i = nx;
        tempnew[i][j] = tempnew[i][0]+(tempnew[i][0]-tempnew[i][1])/2.;
    i = nx+1;
        tempnew[i][j] = tempnew[i][0]+(tempnew[i][0]-tempnew[i][1])/2.;
}
for(j=0;j<ny+2;j++)
    for(i=0;i<nx+2;i++)
        temp[i][j] = tempnew[i][j];

// Show for some results on screen
if(t>=tshow1&&tshow1_stat==0)
{
    printf("Surface Temperature at below mold = %f %f\n",tempnew[0][0],tempnew[0][ny-1]);
    printf("Shell Thickness at below mold = %f m\n",shell_thickness(ny-1,1));
    tshow1_stat=1;
}
else if(t>=tshow2&&tshow2_stat==0)
{
    printf("Surface Temperature at below spray = %f %f\n",tempnew[0][0],tempnew[0][ny-1]);
    printf("Shell Thickness at below spray = %f m\n",shell_thickness(ny-1,1));
    tshow2_stat=1;
}
if(tempnew[nx-1][ny-1]<solid_temp&&solid_show==0)
{
    printf("Solidification time = %f sec = %f m\n",t,v*t);
    solid_show=1;
}

// End Show

// Log Data to file
out1(num_show,t,dtshow,tstop); // Log for Tsurface and Tcenterline
out2(t,dt,tstop); // Log for Shell Thickness
out3(t,dt,tstop,dn,nx-1,2); // Log for Contour Graph
out4(t,dtshow,tstop,dn); // Log for All Point Temperature
}
out1(num_show,t,dtshow,tstop);
out2(t,dt,tstop);
out3(t,dt,tstop,dn,nx-1,2);
out4(t,dtshow,tstop,dn);
output_inp(nx,ny,nzinp,dx,dy,v*dtshow);
}

double check_temp(double temp1,double temp2)
{
    double cpe;
    double qlat,tliq,tsolid;

    qlat = 2.67e5;
    tliq = liquid_temp;
    tsolid = solid_temp;

    if (temp1>tliq&&temp2<tliq)
    {

```

```

        cpe = fcp(temp1) + qlat/(tliq-tsolid);
        temp2 = tliq + (frho(temp1)*fcp(temp1)/frho(temp2)/cpe)*(temp2-tliq);
        return(temp2);
    }
    else if (temp1>tsolid&&temp1<tliq&&temp2<tsolid)
    {
        cpe = fcp(temp1);
        temp2 = tsolid + frho(temp1)*cpe/frho(temp2)/fcp(temp2)*(temp2-tsolid);
        return(temp2);
    }
    else
    {
        return(temp2);
    }
}

double fk(double temp)
{
    double tsolid,tliq;

    tsolid = solid_temp*9./5. + 32 + 459.67;
    tliq = liquid_temp*9./5. + 32 + 459.67;
    temp = temp*9./5. + 32.;
    temp = temp+459.67;
    if(temp>=0&&temp<=2050)
        return(1.7307*(43.8-0.0137*temp));
    else if(temp>2050&&temp<=tliq)
        return(1.7307*(8.0+0.0037*temp));
    else if(temp>tliq)
        return(1.7307*(150));
    else
    {
        printf("Error! Out of temperature range in `fk`\n");
        exit(0);
    }
}

double frho(double temp)
{
    return(rho);
}

double fcp(double temp)
{
    double tsolid,tliq;

    tsolid = solid_temp*9./5. + 32 + 459.67;
    tliq = liquid_temp*9./5. + 32 + 459.67;
    temp = temp*9./5. + 32.;
    temp = temp+459.67;
    if(temp>=900&&temp<=1850)
    {
        temp = temp-459.67;
        return(4186.8*(0.0758125+0.23177e-3*temp+pow(0.345312e-6*temp,2)+pow(0.205729e-4*temp,3)));
    }
    else if(temp>1850&&temp<=2050)
        return(4186.8*(2.21-0.001*temp));
    // else if(temp>2050&&temp<=3200)
    else if(temp>2050&&temp<=tsolid)
        return(4186.8*(0.1242+0.0000125*temp));
    // else if(temp>3200&&temp<=3300)
    else if(temp>tsolid&&temp<=tliq)

```

```

        return(4186.8*(1.88));
//     else if(temp>3300)
//     else if(temp>tliq)
        return(4186.8*(0.165));
    else
    {
        return(4186.8*(0.15));
//     printf("Error! Out of temperature range in \'tcp\");
//     exit(0);
    }
}

```

```
int out1(int num_show,double t,double dtshow,double tend)// Log for Tsurface and Tcenterline
```

```

{
    static double tshow=0;
    static int init_stat=0;
    static int close_stat=0;
    static FILE *output1;
    int i;

    if(init_stat==0)
    {
        output1 = fopen("output1.csv","w+");
        fprintf(output1,"This is output1\n");
        init_stat=1;
    }
    if(t>=tshow&&t<=tend)
    {
        fprintf(output1,"%f",v^t);
        for(i=0;i<num_show;i++)
        {
            fprintf(output1,"%f",temp[post[i].x][post[i].y]);
        }
        fprintf(output1,"\n");
        tshow = tshow + dtshow;
    }
    else if(t>tend&&close_stat==0)
    {
        fclose(output1);
        close_stat=1;
    }
    return(0);
}

```

```
int out2(double t,double dtshow,double tend)// Log for Shell Thickness
```

```

{
    static double tshow=0;
    static int init_stat=0;
    static int close_stat=0;
    static double thickness,oldthickness;
    static FILE *output2;

    if(init_stat==0)
    {
        oldthickness = -1.;
        output2 = fopen("output2.csv","w+");
        fprintf(output2,"This is output2\n");
        init_stat=1;
    }
    if(t>=tshow&&t<=tend)
    {
        thickness = shell_thickness(ny-1,1);
    }
}

```

```

        if(thickness>oldthickness||thickness<oldthickness)
        {
            fprintf(output2,"%f",v*t);
            fprintf(output2,"%f\n",shell_thickness(ny-1,1));
            oldthickness = thickness;
        }
        tshow = tshow + dtshow;
    }
    else if(t>tend&&close_stat==0)
    {
        fclose(output2);
        close_stat=1;
    }
    return(0);
}

```

```
double shell_thickness(int point,int dir)
{

```

```

    int i;

    if(dir==1)
    {
        for(i=0;i<nx;i++)
            if(temp[i][point]>solid_temp)
                return(i*dx);
        return(i*x);
    }
    else if(dir==2)
    {
        for(i=0;i<ny;i++)
            if(temp[point][i]>solid_temp)
                return(i*dy);
        return(i*y);
    }
    return(-1);
}

```

```
int out3(double t,double dtshow,double tend,int dn,int point,int dir)// Log for Contour Graph
{

```

```

    static double tshow=0;
    static int init_stat=0;
    static int close_stat=0;
    static FILE *output3;
    static double tempshow[10];
    static int stat[100][10];
    static int num_temp;
    static double x[100][10];

```

```

    static int l,m,nm;
    static double dm,lm;

```

```

    tempshow[0] = 1520;
    tempshow[1] = 1480;
    tempshow[2] = 1400;
    tempshow[3] = 1200;
    tempshow[4] = 1000;
    tempshow[5] = 800;
    tempshow[6] = 600;
    tempshow[7] = 400;
    tempshow[8] = 200;
    tempshow[9] = 100;
    num_temp = 10;

```

```

if(init_stat==0)
{
    output3 = fopen("output3.csv","w+");
    fprintf(output3,"This is output3\n");
    for(l=0;l<num_temp;l++)
    {
        fprintf(output3,"%f",tempshow[l]);
    }
    fprintf(output3,"\n");
    if(dir==1)
    {
        nm = nx;
        lm = lx;
        dm = dx;
    }
    else if(dir==2)
    {
        nm = ny;
        lm = ly;
        dm = dy;
    }
    for(m=0;m<nm+2;m++)
        for(l=0;l<num_temp;l++)
            stat[m][l] = 0;
    init_stat = 1;
}
if(t>=tshow&& t<=tend)
{
    if(dir==1)
    {
        for(m=0;m<nm+2;m++)
        {
            for(l=0;l<num_temp;l++)
            {
                if(temp[m][point]<=tempshow[l]&&stat[m][l]==0)
                {
                    x[m][l] = v*t;
                    stat[m][l] = 1;
                }
                else if(temp[m][point]>=tempshow[l]&&stat[m][l]==1)
                {
                    printf("Error in reheat contour\n");
                }
            }
        }
    }
    else if(dir==2)
    {
        for(m=0;m<nm+2;m++)
        for(l=0;l<num_temp;l++)
            if(temp[point][m]<=tempshow[l]&&stat[m][l]==0)
            {
                x[m][l] = v*t;
                stat[m][l] = 1;
            }
            else if(temp[point][m]>=tempshow[l]&&stat[m][l]==1)
            {
                printf("Error in reheat contour\n");
            }
        }
    }
}
}

```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```

else if(t>tend&&close_stat==0)
{
    m = nm+1;
    fprintf(output3,"%f",0.);
    for(i=0;i<num_temp;i++)
        fprintf(output3,"%f",x[m][i]);
    fprintf(output3,"\n");
    for(m=0;m<nm;m=m+dn) // **j = j+dn
    {
        fprintf(output3,"%f",m*dm+dm/2.);
        for(i=0;i<num_temp;i++)
            fprintf(output3,"%f",x[m][i]);
        fprintf(output3,"\n");
    }
    m = nm;
    fprintf(output3,"%f",lm);
    for(i=0;i<num_temp;i++)
        fprintf(output3,"%f",x[m][i]);
    fprintf(output3,"\n");
    fclose(output3);
    close_stat = 1;
}
return(0);
}

int out4(double t,double dtshow,double tend,int dn) // Log for All Point Temperature
{
    static double tshow=0;
    static int init_stat=0;
    static int close_stat=0;
    static FILE *output4;
    int i,j;

    if(init_stat==0)
    {
        output4 = fopen("output4.csv","w+");
        fprintf(output4,"This is output4\n");
        init_stat=1;
    }
    if(t>=tshow&&t<=tend)
    {
        fprintf(output4,"%f",v[t]);
        j = ny+1;
        {
            i = nx+1;
            fprintf(output4,"%f",temp[i][j]);
            for(i=0;i<nx;i=i+dn)
                fprintf(output4,"%f",temp[i][j]);
            i = nx;
            fprintf(output4,"%f",temp[i][j]);
            fprintf(output4,"\n");
        }
        for(j=0;j<ny;j=j+dn) // **j = j+dn
        {
            i = nx+1;
            fprintf(output4,"%f",temp[i][j]);
            for(i=0;i<nx;i=i+dn)
            {
                fprintf(output4,"%f",temp[i][j]);
                inptemp[i][j][nzinp] = temp[i][j];
            }
            i = nx;

```

```

        fprintf(output4,"%f",temp[i][j]);
        fprintf(output4,"\n");
    }
    j = ny;
    {
        i = nx+1;
        fprintf(output4,"%f",temp[i][j]);
        for(i=0;i<nx;i+=dn)
            fprintf(output4,"%f",temp[i][j]);
        i = nx;
        fprintf(output4,"%f",temp[i][j]);
        fprintf(output4,"\n");
    }
    tshow = tshow + dtshow;
    nzinp++;
}
else if(t>tend&&close_stat==0)
{
    fclose(output4);
    close_stat=1;
}
return(0);
}

int output_inp(int nx,int ny,int nz,double dx,double dy,double dz)
{
    FILE* output;
    int i,j,k;
    int ii;

    output = fopen("output.inp","w+");
    // Prepare Output File
    ii=1;
    for(k=0;k<nz;k++)
    {
        for(i=0;i<nx;i++)
            for(j=0;j<ny;j++)
                ii++;
        fprintf(output,"%d %d %d %d %d\n",(nx+1)*(ny+1)*(nz+1),ii-1,0,1,0);
        for(k=0;k<nz+1;k++)
        {
            for(i=0;i<nx+1;i++)
                for(j=0;j<ny+1;j++)
                {
                    fprintf(output,"%d %f %f %f\n",k*(nx+1)*(ny+1)+i*(ny+1)+j+1,i*dx,j*dy,k*dz);
                }
        }
        ii=1;
        for(k=0;k<nz;k++)
        {
            for(i=0;i<nx;i++)
                for(j=0;j<ny;j++)
                {
                    fprintf(output,"%d %d hex %d %d %d %d %d %d %d\n",ii,0,
                        k*(nx+1)*(ny+1)+i*(ny+1)+j+1,k*(nx+1)*(ny+1)+i*(ny+1)+j+1,k*(nx+1)*(ny+1)+i*(ny+1)+j+1,
                        (ny+1)+j+1,k*(nx+1)*(ny+1)+i*(ny+1)+j+1),
                        (k+1)*(nx+1)*(ny+1)+i*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+i*(ny+1)+j+1,(k+1)*
                        (nx+1)*(ny+1)+i*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+i*(ny+1)+j+1);
                    ii++;
                }
            }
        }
}

```

```
        }  
    }  
    fprintf(output,"%d %d\n",1,1);  
    fprintf(output,"Temperature,Degree C\n");  
  
    ii=1;  
    for(k=0;k<nz;k++)  
    {  
        for(i=0;i<nx;i++)  
        {  
            for(j=0;j<ny;j++)  
            {  
                fprintf(output,"%d %f\n",ii,1500./inptemp[i][j][k]);  
                ii++;  
            }  
        }  
    }  
    fclose(output);  
    return(0);  
}
```



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ไฟล์ 3d.c

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

double cal(int,int,int);
double check_temp(double,double);
int output_inp(int,int,int,double,double,double);
double fk(double);
double frho(double);
double fcp(double);
int output_all(long int,long int,long int,double,double,double);
int output_some(long int,long int,long int,double,double,double,int,int);
int output_select(long int,long int,long int,int,double);
int output_inp2(long int,long int,long int,double,double,double);

double temp[50][50][5000];
double tempnew[50][50][5000];
int end_cell;
double lx,ly,lz;
int nx,ny,nz;
double dx,dy,dz;
int ndz;
double t1;
double dt,v;
double tw,tair,pour_temp;
int ii=0;
double zigma,epsilon;
double lzstart;
int nzstart;
double rho,liquid_temp,solid_temp;
double zshow1,zshow2,zmold,zspray[20],hs[20];
int num_spray;
int nzspray[20],nzstop;
double dzshow,dtshow,tshow1,tshow2;
int tshow1_stat,tshow2_stat,solid_show,dn,nzmold,nzend;
double tendmold,zstop;
double endt = 200.;

int main()
{
//   int init_output(FILE *output);

    int i,j,k;
    FILE *output,*output1;
    int iteration = 0;
    double t,tstart,tstop;
    double t1;
    int kstart,kstop;
    int stat;
    double diff;

    output1 = fopen("out.txt","w+");
    lx = (double).90;nx = 45;dx = lx/hx;
    ly = (double).15;ny = 15;dy = ly/hy;

```

```

lz = (double)2.;dz = dy;nzend = lz/dz;
//
lzstart = .62;nzstart = lzstart/dz;
ndz = 2;
dt = 0.01;
rho = 2680;
pour_temp = 657.+30.;
liquid_temp = 657.;
solid_temp = 646.;
tw = 25;
tair = 50;
zigma = 5.669e-8;
epsilon = 0.8;
v = 1./60.;
//
dt = dz/v;
zshow1 = 0.05;
zshow2 = 1.;
zmold = 0.05;
nzmold = zmold/dz;
num_spray = 3;
zspray[0] = .3;
zspray[1] = 1.;
zspray[2] = 2.;
zspray[3] = 7.7;
zspray[4] = 11;
zspray[5] = 14.5;
zspray[6] = 16.3;
zspray[7] = 17;
//
zstop = 26;
dzshow = 0.02;
tshow1 = zshow1/v;
tshow2 = zshow2/v;
tshow1_stat = 0;
tshow2_stat = 0;
solid_show = 0;
//
tendmold = zmold/v;
nzspray[0] = zspray[0]/dz;hs[0] = 0.005*41868.;
nzspray[1] = zspray[1]/dz;hs[1] = 0.2*0.005*41868.;
nzspray[2] = zspray[2]/dz;hs[2] = 0.3*0.005*41868.;
nzspray[3] = zspray[3]/dz;hs[3] = 311.684;
nzspray[4] = zspray[4]/dz;hs[4] = 330.292;
nzspray[5] = zspray[5]/dz;hs[5] = 240.741;
nzspray[6] = zspray[6]/dz;hs[6] = 381.464;
nzspray[7] = zspray[7]/dz;hs[7] = 226.785;
//
nzsstop = zstop/dz;
// init for output parameter
//
dtshow = dzshow/v;
/*
num_show = 7;
post[0].x = 0; // for sub surface
post[1].x = (16.)/1000./dx; // 96mm from surface
post[2].x = (32.)/1000./dx; // 64mm from surface
post[3].x = (48.)/1000./dx; // 32mm from surface
post[4].x = nx-1; // for sub centerline
post[5].x = nx; // for centerline
post[6].x = nx+1; // for surface
post[0].y = ny-1;
post[1].y = ny-1;
post[2].y = ny-1;
post[3].y = ny-1;
post[4].y = ny-1;
post[5].y = ny-1;
post[6].y = ny-1;*/
dn = 1;

```

```

ndz = 2;
// t1 = 25*dt; //t1 = 5 sec
t1 = 5.;

// Define each zone at t=0 liquid fill in mould
// t = 0;

nz = nzmold;
for(k=0;k<nz;k++)
    for(j=0;j<ny;j++)
        for(i=0;i<nx;i++)
            temp[i][j][k] = pour_temp;

//calculate for 1st holding time
// 0 < t < t1
tstart = dt;
tstop = t1;
for(t=tstart;t<tstop;t+=dt)
{
// kstart = 0;
// kstop = nz1;
for(k=0;k<nz;k++)
    for(j=0;j<ny;j++)
        for(i=0;i<nx;i++)
        {
            tempnew[i][j][k] = cal(i,j,k); // Casting Zone
            tempnew[i][j][k] = check_temp(temp[i][j][k],tempnew[i][j][k]);
        }

// Swap cell and cellnew
stat = 1;
for(k=ndz;k<nz;k++)
    for(j=0;j<ny;j++)
        for(i=0;i<nx;i++)
        {
            diff = (tempnew[i][j][k] - temp[i][j][k])/temp[i][j][k];
            if(fabs(diff)>0.00001)
                stat = 0;
        }
    for(k=0;k<nz;k++)
        for(j=0;j<ny;j++)
            for(i=0;i<nx;i++)
                temp[i][j][k] = tempnew[i][j][k];

    if(stat==1)
        break;
}

while(stat == 0&& t<endt)
{
// switch for next length
// kstart = 0;
nz = nz+ndz;
if(nz>nzend)

```

```

        nz = nzend;
    for(k=nz-ndz-1;k>=0;k--)
    {
        for(j=0;j<ny;j++)
            for(i=0;i<nx;i++)
            {
                temp[i][j][k+ndz] = temp[i][j][k];
            }
    }
    for(k=0+ndz-1;k>=0;k--)
    {
        for(j=0;j<ny;j++)
            for(i=0;i<nx;i++)
            {
                temp[i][j][k] = pour_temp;
            }
    }
    //calculate for next n*dt
    tstart = t;//tstart should = 5.2
    tstop = tstart+ndz*dz/v;//tstop should = 5.2+2*.2*5*10 = 7.2
    for(t=tstart;t<tstop;t+=dt)
    {
        for(k=0;k<nz;k++) // mould cooling
        {
            for(j=0;j<ny;j++)
            {
                for(i=0;i<nx;i++)
                {
                    tempnew[i][j][k] = cal(i,j,k); // Casting Zone
                    tempnew[i][j][k] = check_temp(temp[i][j][k],tempnew[i][j][k]);
                }
            }
        }
        stat = 1;
        for(k=0+ndz;k<nz;k++) // mould cooling
        {
            for(j=0;j<ny;j++)
            {
                for(i=0;i<nx;i++)
                {
                    diff = (tempnew[i][j][k] - tempnew[i][j][k-ndz])/tempnew[i][j][k-ndz];
                    if(fabs(diff)>0.000001)
                        stat = 0;
                }
            }
        }
        for(k=0;k<nz;k++)
            for(j=0;j<ny;j++)
                for(i=0;i<nx;i++)
                    temp[i][j][k] = tempnew[i][j][k];
        if(stat==1)
            break;
    }
    printf("tstart = %f  tstop = %f  t = %f  nz = %d\n",tstart,tstop,t,nz);
    printf("0,0,0 = %f,nx-1,0,0 = %f,0,ny-1,0 = %f,0,0,nz-1 = %f,nx-1,ny-1,nz-1 = %f\n",temp[0][0][0],temp[nx-1][0][0],temp[0][ny-1][0],temp[1][1][60],temp[nx-1][ny-1][nz-1]);
    //    fprintf(output1,"tstart = %f  tstop = %f  t = %f  nz = %d\n",tstart,tstop,t,nz);
    //    fprintf(output1,"%f,0,0 = %f,nx-1,0,0 = %f,0,ny-1,0 = %f,0,0,nz-1 = %f,nx-1,ny-1,nz-1 = %f\n",temp[0][0][0],temp[nx-1][0][0],temp[0][ny-1][0],temp[0][0][nz-1],temp[nx-1][ny-1][nz-1]);
    }
    // Check Criteria
    // while(stat = 0)

```

```

//      {
//          // switch for next length
//          // calculate for next n*dt
//      }
//      // output
//      // end

/**
Initiate output
output = fopen("output.inp","w+");
init_output(output);
//      End Initiate

//      Print the results
ii=0;
for(k=0;k<nz;k++)
{
    for(i=0;i<nx;i++)
    {
        ii++;
        fprintf(output,"%d %f\n",ii,temp[i][j][k]);
    }
}
//      End Print*/

//      fclose(output);
output_all(nx,ny,nz,dx,dy,dz);
output_some(nx,ny,nz,dx,dy,dz,5,5,5);
output_inp2(nx,ny,nz,dx,dy,dz);
output_select(i,j,k,1,1200);
fclose(output1);
return(0);
}

double cal(int i,int j,int k)
{
    int x = 999,i;
    double alpha1,alpha2,alpha3,alpha4,alpha5,alpha6;
    double k1,k2,k3,k4,k5,k6;
    double qout;

    if(k==0)
    {
        if(j==0)
        {
            if(i==0)
                x = 0;
            else if (i<nx-1)
                x = 1;
            else if (i==nx-1)
                x = 2;
        }
        else if(j<ny-1)
        {
            if(i==0)
                x = 3;
            else if (i<nx-1)
                x = 4;
            else if (i==nx-1)
                x = 5;
        }
        else if(j==ny-1)

```

```

    {
        if(j==0)
            x = 6;
        else if (i<nx-1)
            x = 7;
        else if (i==nx-1)
            x = 8;
    }
}
else if(k>0&&k<nz-1)
{
    if(j==0)
    {
        if(j==0)
            x = 9;
        else if (i<nx-1)
            x = 10;
        else if (i==nx-1)
            x = 11;
    }
    else if(j<ny-1)
    {
        if(j==0)
            x = 12;
        else if (i<nx-1)
            x = 13;
        else if (i==nx-1)
            x = 14;
    }
    else if(j==ny-1)
    {
        if(j==0)
            x = 15;
        else if (i<nx-1)
            x = 16;
        else if (i==nx-1)
            x = 17;
    }
}
else if(k==nz-1)
{
    if(j==0)
    {
        if(j==0)
            x = 18;
        else if (i<nx-1)
            x = 19;
        else if (i==nx-1)
            x = 20;
    }
    else if(j<ny-1)
    {
        if(j==0)
            x = 21;
        else if (i<nx-1)
            x = 22;
        else if (i==nx-1)
            x = 23;
    }
    else if(j==ny-1)
    {

```

```

        if(i==0)
            x = 24;
        else if (i<nx-1)
            x = 25;
        else if (i==nx-1)
            x = 26;
    }
}

if(j==0||j==ny-1||i==0||i==nx-1)
{
    if(k>=0&&k<=nzmold)
    {
        if(temp[i][j][k]>=liquid_temp)
            qout = (temp[i][j][k]-tw)/(1./0.05/418680.+0.05/401.+1./17500);
        else if(temp[i][j][k]>solid_temp&&temp[i][j][k]<liquid_temp)
            qout = (temp[i][j][k]-tw)/(1./0.03/418680.+0.05/401.+1./17500);
        else if(temp[i][j][k]<solid_temp)
            qout = (temp[i][j][k]-tw)/(1./0.01/418680.+0.05/401.+1./17500);
        //
        //
        //
        qout = 3.*1;
        qout = 3.154*(850000.-100000.*sqrt((k*dz/v)))/time unit use in equation is second// //Savage and Pritchard
        qout = 3.154*(850000.*pow(2.71828,(-.0386*(k*dz/v)))+4000.*(t-dt)-36000.);//Kraimer and Tarmann
    }
    if(k>nzmold&&k<=nzspray[0])
        qout = hs[0]*(temp[i][j][k]-tw);
    for(l=1;l<num_spray;l++)
    {
        if(k>nzspray[l-1]&&k<=nzspray[l])
            qout = hs[l]*(temp[i][j][k]-tw);
    }
    if(k>nzspray[num_spray-1])
        qout = zigma*epsilon*(pow(temp[i][j][k],4)-pow(tair,4));
}
alpha1 = (fk(temp[i][j][k])+fk(temp[i-1][j][k]))/2./rho(temp[i][j][k])/fcp(temp[i][j][k]);
alpha2 = (fk(temp[i][j][k])+fk(temp[i+1][j][k]))/2./rho(temp[i][j][k])/fcp(temp[i][j][k]);
alpha3 = (fk(temp[i][j][k])+fk(temp[i][j-1][k]))/2./rho(temp[i][j][k])/fcp(temp[i][j][k]);
alpha4 = (fk(temp[i][j][k])+fk(temp[i][j+1][k]))/2./rho(temp[i][j][k])/fcp(temp[i][j][k]);
if(k1=0)
    alpha5 = (fk(temp[i][j][k])+fk(temp[i][j][k-1]))/2./rho(temp[i][j][k])/fcp(temp[i][j][k]);
if(k1=nz)
    alpha6 = (fk(temp[i][j][k])+fk(temp[i][j][k+1]))/2./rho(temp[i][j][k])/fcp(temp[i][j][k]);
k1 = alpha1*dt/dx/dx;
k2 = alpha2*dt/dx/dx;
k3 = alpha3*dt/dy/dy;
k4 = alpha4*dt/dy/dy;
k5 = alpha5*dt/dz/dz;
k6 = alpha6*dt/dz/dz;

switch(x)
{
    case 0 :
        return(
            (1-k2-k4-k6)*temp[i][j][k]
            + k2*temp[i+1][j][k]
            + k4*temp[i][j+1][k] + k6*temp[i][j][k+1]
            - qout*dt/rho(temp[i][j][k])/fcp(temp[i][j][k])/dx
            - qout*dt/rho(temp[i][j][k])/fcp(temp[i][j][k])/dy
        );
        break;
}

```

case 1 :

```

return(
(1-k1-k2-k4-k6)*temp[i][k]
+ k1*temp[i-1][k] + k2*temp[i+1][k]
+ k4*temp[i][k+1] + k6*temp[i][k+1]
- qout*dt/rho(temp[i][k])/fcp(temp[i][k])/dy
);
break;

```

case 2 :

```

return(
(1-k1-k4-k6)*temp[i][k]
+ k1*temp[i-1][k]
+ k4*temp[i][k+1] + k6*temp[i][k+1]
- qout*dt/rho(temp[i][k])/fcp(temp[i][k])/dy
);
break;

```

case 3 :

```

return(
(1-k2-k3-k4-k6)*temp[i][k]
+ k2*temp[i+1][k] + k3*temp[i-1][k]
+ k4*temp[i][k+1] + k6*temp[i][k+1]
- qout*dt/rho(temp[i][k])/fcp(temp[i][k])/dx
);
break;

```

case 4 :

```

return(
(1-k1-k2-k3-k4-k6)*temp[i][k]
+ k1*temp[i-1][k] + k2*temp[i+1][k] + k3*temp[i-1][k]
+ k4*temp[i][k+1] + k6*temp[i][k+1]
);
break;

```

case 5 :

```

return(
(1-k1-k3-k4-k6)*temp[i][k]
+ k1*temp[i-1][k] + k3*temp[i-1][k]
+ k4*temp[i][k+1] + k6*temp[i][k+1]
);
break;

```

case 6 :

```

return(
(1-k2-k3-k6)*temp[i][k]
+ k2*temp[i+1][k] + k3*temp[i-1][k]
+ k6*temp[i][k+1]
- qout*dt/rho(temp[i][k])/fcp(temp[i][k])/dx
);
break;

```

case 7 :

```

return(

```



```

(1-k1-k2-k3-k6)*temp[i][k]
+ k1*temp[i-1][k] + k2*temp[i+1][k] + k3*temp[i][k-1]
+ k6*temp[i][k+1]
);
break;

```

case 8 :

```

return(
(1-k1-k3-k6)*temp[i][k]
+ k1*temp[i-1][k] + k3*temp[i][k-1]
+ k6*temp[i][k+1]
);
break;

```

case 9 :

```

return(
(1-k2-k4-k5-k6)*temp[i][k]
+ k2*temp[i+1][k]
+ k4*temp[i][k+1] + k5*temp[i][k-1] + k6*temp[i][k+1]
- qout*dt/frho(temp[i][k])/fcp(temp[i][k])/dx
- qout*dt/frho(temp[i][k])/fcp(temp[i][k])/dy
);
break;

```

case 10 :

```

return(
(1-k1-k2-k4-k5-k6)*temp[i][k]
+ k1*temp[i-1][k] + k2*temp[i+1][k]
+ k4*temp[i][k+1] + k5*temp[i][k-1] + k6*temp[i][k+1]
- qout*dt/frho(temp[i][k])/fcp(temp[i][k])/dy
);
break;

```

case 11 :

```

return(
(1-k1-k4-k5-k6)*temp[i][k]
+ k1*temp[i-1][k]
+ k4*temp[i][k+1] + k5*temp[i][k-1] + k6*temp[i][k+1]
- qout*dt/frho(temp[i][k])/fcp(temp[i][k])/dy
);
break;

```

case 12 :

```

return(
(1-k2-k3-k4-k5-k6)*temp[i][k]
+ k2*temp[i+1][k] + k3*temp[i][k-1]
+ k4*temp[i][k+1] + k5*temp[i][k-1] + k6*temp[i][k+1]
- qout*dt/frho(temp[i][k])/fcp(temp[i][k])/dx
);
break;

```

case 13 :

```

return(
(1-k1-k2-k3-k4-k5-k6)*temp[i][k]
+ k1*temp[i-1][k] + k2*temp[i+1][k] + k3*temp[i][k-1]
+ k4*temp[i][k+1] + k5*temp[i][k-1] + k6*temp[i][k+1]

```

```
);
break;
```

case 14 :

```
return(
(1-k1-k3-k4-k5-k6)*temp[i][j][k]
+ k1*temp[i-1][j][k] + k3*temp[i][j-1][k]
+ k4*temp[i][j+1][k] + k5*temp[i][j][k-1] + k6*temp[i][j][k+1]
);
break;
```

case 15 :

```
return(
(1-k2-k3-k5-k6)*temp[i][j][k]
+ k2*temp[i+1][j][k] + k3*temp[i][j-1][k]
+ k5*temp[i][j][k-1] + k6*temp[i][j][k+1]
- qout*dt/frho(temp[i][j][k])/fcp(temp[i][j][k])/dx
);
break;
```

case 16 :

```
return(
(1-k1-k2-k3-k5-k6)*temp[i][j][k]
+ k1*temp[i-1][j][k] + k2*temp[i+1][j][k] + k3*temp[i][j-1][k]
+ k5*temp[i][j][k-1] + k6*temp[i][j][k+1]
);
break;
```

case 17 :

```
return(
(1-k1-k3-k5-k6)*temp[i][j][k]
+ k1*temp[i-1][j][k] + k3*temp[i][j-1][k]
+ k5*temp[i][j][k-1] + k6*temp[i][j][k+1]
);
break;
```

case 18 :

```
return(
(1-k2-k4-k5)*temp[i][j][k]
+ k2*temp[i+1][j][k]
+ k4*temp[i][j+1][k] + k5*temp[i][j][k-1]
- qout*dt/frho(temp[i][j][k])/fcp(temp[i][j][k])/dx
- qout*dt/frho(temp[i][j][k])/fcp(temp[i][j][k])/dy
);
break;
```

case 19 :

```
return(
(1-k1-k2-k4-k5)*temp[i][j][k]
+ k1*temp[i-1][j][k] + k2*temp[i+1][j][k]
+ k4*temp[i][j+1][k] + k5*temp[i][j][k-1]
- qout*dt/frho(temp[i][j][k])/fcp(temp[i][j][k])/dy
);
break;
```

case 20 :

```

return(
(1-k1-k4-k5)*temp[i][k]
+ k1*temp[i-1][k]
+ k4*temp[i+1][k] + k5*temp[i][k-1]
- qout*dt/frho(temp[i][k])/fcp(temp[i][k])/dy
);
break;

```

case 21 :

```

return(
(1-k2-k3-k4-k5)*temp[i][k]
+ k2*temp[i+1][k] + k3*temp[i-1][k]
+ k4*temp[i+1][k] + k5*temp[i][k-1]
- qout*dt/frho(temp[i][k])/fcp(temp[i][k])/dx
);
break;

```

case 22 :

```

return(
(1-k1-k2-k3-k4-k5)*temp[i][k]
+ k1*temp[i-1][k] + k2*temp[i+1][k] + k3*temp[i-1][k]
+ k4*temp[i+1][k] + k5*temp[i][k-1]
);
break;

```

case 23 :

```

return(
(1-k1-k3-k4-k5)*temp[i][k]
+ k1*temp[i-1][k] + k3*temp[i-1][k]
+ k4*temp[i+1][k] + k5*temp[i][k-1]
);
break;

```

case 24 :

```

return(
(1-k2-k3-k5)*temp[i][k]
+ k2*temp[i+1][k] + k3*temp[i-1][k]
+ k5*temp[i][k-1]
- qout*dt/frho(temp[i][k])/fcp(temp[i][k])/dx
);
break;

```

case 25 :

```

return(
(1-k1-k2-k3-k5)*temp[i][k]
+ k1*temp[i-1][k] + k2*temp[i+1][k] + k3*temp[i-1][k]
+ k5*temp[i][k-1]
);
break;

```

case 26 :

```

(1-k1-k3-k5)*temp[i][k]
return(
+ k1*temp[i-1][k] + k3*temp[i-1][k]
+ k5*temp[i][k-1]
);

```

```

        break;
    default :
        printf("Error in switch section\n");
        return(-1);
        break;
    }
}

double check_temp(double temp1,double temp2)
{
    double cpe;
    double qlat,tliq,tsolid;

    qlat = 390000.;
    tliq = liquid_temp;
    tsolid = solid_temp;

    if (temp1>tliq&&temp2<tliq)
    {
        cpe = fcp(temp1) + qlat/(tliq-tsolid);
        temp2 = tliq + (rho(temp1)*fcp(temp1)/rho(temp2)/cpe)*(temp2-tliq);
        return(temp2);
    }
    else if (temp1>tsolid&&temp1<tliq&&temp2<tsolid)
    {
        cpe = fcp(temp1);
        temp2 = tsolid + rho(temp1)*cpe/rho(temp2)/fcp(temp2)*(temp2-tsolid);
        return(temp2);
    }
    else
    {
        return(temp2);
    }
}

double fk(double temp)
{
    double tsolid,tliq;
    if(temp>=liquid_temp)
        return(110);
    else if(temp>solid_temp&&temp<liquid_temp)
        return(170);
    else
        return(234.);
}

double frho(double temp)
{
    if(temp>=liquid_temp)
        return(2390);
    else
        return(2705);
}

double fcp(double temp)
{
    double tsolid,tliq;

    return(900);
}

```

```

int output_inp(int nx,int ny,int nz,double dx,double dy,double dz)
{
    FILE* output;
    int i,j,k;
    int ii;

    output = fopen("output.inp","w+");
    // Prepare Output File
    ii=1;
    for(k=0;k<nz;k++)
    {
        for(i=0;i<nx;i++)
            for(j=0;j<ny;j++)
                ii++;
    }
    fprintf(output,"%d %d %d %d %d\n",(nx+1)*(ny+1)*(nz+1),ii-1,0,1,0);
    for(k=0;k<nz+1;k++)
    {
        for(i=0;i<nx+1;i++)
            for(j=0;j<ny+1;j++)
            {
                fprintf(output,"%d %d %f %f\n",k*(nx+1)*(ny+1)+i*(ny+1)+j+1,"dx,j"dy,k*dz);
            }
    }
    ii=1;
    for(k=0;k<nz;k++)
    {
        for(i=0;i<nx;i++)
            for(j=0;j<ny;j++)
            {
                fprintf(output,"%d %d hex %d %d %d %d %d %d %d\n",ii,0,
                    k*(nx+1)*(ny+1)+i*(ny+1)+(j+1)+1,k*(nx+1)*(ny+1)+i*(ny+1)+j+1,k*(nx+1)*(ny+1)+
                    (i+1)*(ny+1)+j+1,k*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1,
                    (k+1)*(nx+1)*(ny+1)+i*(ny+1)+(j+1)+1,(k+1)*(nx+1)*(ny+1)+i*(ny+1)+j+1,(k+1)*
                    (nx+1)*(ny+1)+(i+1)*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1);
                ii++;
            }
    }
    fprintf(output,"%d %d\n",1,1);
    fprintf(output,"Temperature,Degree C\n");

    ii=1;
    for(k=0;k<nz;k++)
    {
        for(i=0;i<nx;i++)
            for(j=0;j<ny;j++)
            {
                fprintf(output,"%d %f\n",ii,1500./inptemp[i][j][k]');
                ii++;
            }
    }
    }
    fclose(output);
    return(0);
}

int output_all(long int nx,long int ny,long int nz,double dx,double dy,double dz)
{
    FILE* output;

```

```

long int i,j,k;

output = fopen("output1.out","w+");
fprintf(output,"%d,%d,%d\n",nx,ny,nz);
fprintf(output,"%f,%f,%f\n",dx,dy,dz);
for(k=0;k<nz;k++)
{
    for(j=0;j<ny;j++)
        for(i=0;i<nx;i++)
            fprintf(output,"%d,%d,%d,%f\n",i,j,k,temp[i][j][k]);
}
fclose(output);
return(0);
}

int output_some(long int nx,long int ny,long int nz,double dx,double dy,double dz,int sx,int sy,int sz)
{
    FILE* output;
    long int i,j,k;
    long int ii,jj,kk;
    long int nii,njj,nkk;

    output = fopen("output2.out","w+");
    fprintf(output,"%d,%d,%d\n",(nx-1)/sx+1,(ny-1)/sy+1,(nz-1)/sz+1);
    fprintf(output,"%f,%f,%f\n",dx*sx,dy*sy,dz*sz);
    for(k=0,kk=0*kk=sz',nkk=0;k<nz;k++)
        if(k==kk)
        {
            for(j=0,jj=0*jj=sy',njj=0;j<ny;j++)
                if(j==jj)
                {
                    for(i=0,ii=0*ii=sx',nii=0;i<nx;i++)
                        if(i==ii)
                        {
                            fprintf(output,"%d,%d,%d,%f\n",nii,njj,nkk,temp[i][j][k]);
                            ii+=sx;
                            nii++;
                        }
                    jj+=sy;
                    njj++;
                }
            kk+=sz;
            nkk++;
        }
    fclose(output);
    return(0);
}

int output_inp2(long int nx,long int ny,long int nz,double dx,double dy,double dz)
{
    FILE* output;
    long int i,j,k;
    long int ii;

    output = fopen("output2.inp","w+");
    // Prepare Output File
    ii=1;
    for(k=0;k<nz;k++)
    {
        for(i=0;i<nx;i++)
            for(j=0;j<ny;j++)
                // if(cell[i][j][k].material==2)

```

```

        ii++;
    }
    fprintf(output,"%d %d %d %d %d\n",(nx+1)*(ny+1)*(nz+1),ii-1,0,1,0);
//
    fprintf(output,"%d %d %d %d %d\n",(nx+1)*(ny+1)*(nz+1),nx*ny*nz,0,1,0);
    for(k=0;k<nz;k++)
    {
        for(i=0;i<nx+1;i++)
            for(j=0;j<ny+1;j++)
            {
                fprintf(output,"%d %d %f %f\n",k*(nx+1)*(ny+1)+i*(ny+1)+j+1,i*dx,j*dy,k*dz);
            }
    }
    ii=1;
    for(k=0;k<nz;k++)
    {
        for(i=0;i<nx;i++)
            for(j=0;j<ny;j++)
            {
                if(cell[i+1][j+1][k+1].material==2)
                {
                    fprintf(output,"%d hex %d %d %d %d %d %d %d\n",ii,0,
                        k*(nx+1)*(ny+1)+i*(ny+1)+(j+1)+1,k*(nx+1)*(ny+1)+i*(ny+1)+j+1,k*(nx+1)*(ny+1)+
                    (i+1)*(ny+1)+j+1,k*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1,
                        (k+1)*(nx+1)*(ny+1)+i*(ny+1)+(j+1)+1,(k+1)*(nx+1)*(ny+1)+i*(ny+1)+j+1,(k+1)*
                    (nx+1)*(ny+1)+(i+1)*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1);
                    ii++;
                }
            }
        fprintf(output,"%d %d hex %d %d %d %d %d %d %d\n",k*nx*ny+i*ny+j+1,0,
            k*(nx+1)*(ny+1)+i*(ny+1)+(j+1)+1,k*(nx+1)*(ny+1)+i*(ny+1)+j+1,k*(nx+1)*(ny+1)+(i+1)*
            (ny+1)+j+1,k*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1,
            (k+1)*(nx+1)*(ny+1)+i*(ny+1)+(j+1)+1,(k+1)*(nx+1)*(ny+1)+i*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+
            (i+1)*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1);
        //
            k*(nx+1)*(ny+1)+i*(ny+1)+j+1,k*(nx+1)*(ny+1)+i*(ny+1)+j+1,k*(nx+1)*(ny+1)+(i+1)*
            (ny+1)+j+1,k*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1,
            (k+1)*(nx+1)*(ny+1)+i*(ny+1)+(j+1)+1,(k+1)*(nx+1)*(ny+1)+i*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+
            (i+1)*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1);
        //
            k*(nx+1)*(ny+1)+i*(ny+1)+j+1,k*(nx+1)*(ny+1)+i*(ny+1)+j+1,k*(nx+1)*(ny+1)+(i+1)*
            (ny+1)+j+1,k*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1,
            (k+1)*(nx+1)*(ny+1)+i*(ny+1)+(j+1)+1,(k+1)*(nx+1)*(ny+1)+i*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+
            (i+1)*(ny+1)+j+1,(k+1)*(nx+1)*(ny+1)+(i+1)*(ny+1)+(j+1)+1);
    }
    }
    fprintf(output,"%d %d\n",1,1);
    fprintf(output,"Temperature, Degree C\n");

    ii=1;
    for(k=0;k<nz;k++)
    {
        for(i=0;i<nx;i++)
            for(j=0;j<ny;j++)
            {
                if(cell[i][j][k].material==2)
                {
                    fprintf(output,"%d %f\n",ii,temp[i][j][k]);
                    fprintf(output,"%d %f\n",k-1)*nx*ny+(i-1)*ny+j,cell[i][j][k].temp);
                    ii++;
                }
            }
        }
    }
    fprintf(output,"n");
//
    printf("%f,%f,%f,"cell[3][3][3].temp,cell[50][50][3].temp,cell[50][50][50].temp);

    fclose(output);
    return(0);

```

```

}
int output_select(long int i,long int j,long int k,int plane,double temp1)
{
//      long int i,j,k;
      FILE *output,*output2,*output3;
      double tempx[10];
      int stat[10];
      int num_temp,i;

      tempx[0] = 1490;
      tempx[1] = 1440;
      tempx[2] = 1200;
      tempx[3] = 900;
      tempx[4] = 700;
      tempx[5] = 400;
      tempx[6] = 100;
      num_temp = 7;

      output = fopen("output3.out","w+");
      output2 = fopen("output4.out","w+");
      output3 = fopen("output5.csv","w+");
      j = 0;
      for(i=0;i<nx;i++)
      {
          for(l=0;l<num_temp;l++)
              stat[l] = 0;
          fprintf(output,"%f",i*dx);
          for(k=0;k<nz;k++)
          {
              for(l=0;l<num_temp;l++)
              {
                  if((temp[i][j][k] <= tempx[l])&&(stat[l]==0))
                  {
                      fprintf(output,"%f",k*dz);
                      stat[l] = 1;
                  }
                  if((temp[i][j][k] > tempx[l])&&(stat[l]==1)) // new modified
                  {
                      printf("Error reheat contour\n");
                      exit(0);
                  }
              }
          }
          fprintf(output,"\n");
      }
      fclose(output);

      k = 300;
      i=0;
      fprintf(output2,"%f",i*dx);
      for(i=1;i<nx;i++)
      {
          fprintf(output2,"%f",i*dx);
      }
      fprintf(output2,"\n");
      for(j=0;j<ny;j++)
      {
          fprintf(output2,"%f",j*dy);
          for(i=0;i<nx;i++)
          {
              fprintf(output2,"%f",temp[i][j][k]);
          }
      }

```



```
        fprintf(output2,"n");
    }
    fclose(output2);

    j = 0;
    i=0;
    fprintf(output3,"%f",i*dx);
    for(i=1;i<nx;i++)
    {
        fprintf(output3,"%f",i*dx);
    }
    fprintf(output3,"n");
    for(k=0;k<nz;k++)
    {
        fprintf(output3,"%f",k*dz);
        for(i=0;i<nx;i++)
        {
            fprintf(output3,"%f",temp[i][k]);
        }
        fprintf(output3,"n");
    }
    fclose(output3);

    return(0);
}
```



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

ชื่อ-นามสกุล	นาย ปิยะ แก้วนพรัตน์
วัน-เดือน-ปีเกิด	23 ตุลาคม 2517
ที่อยู่	55/503 ซอยนวลจันทร์ ถนนสุขาภิบาล 1 แขวงคลองกุ่ม เขตบึงกุ่ม กรุงเทพมหานคร 10230
วุฒิการศึกษา	ประถมศึกษา โรงเรียนแสงทองวิทยา ปี 2528 มัธยมศึกษาตอนต้น โรงเรียนหาดใหญ่วิทยาลัย ปี 2531 มัธยมศึกษาตอนปลาย โรงเรียนเตรียมอุดมศึกษา ปี 2533 วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมโลหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปี 2541
โทรศัพท์ติดต่อ	01-834-9404



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย