

اللهم لا تحزننا ولا تجعلنا
من الحزينين



دانشگاه صنعتی شاهرود

دانشکده مهندسی مکانیک و مکاترونیک

پایان نامه کارشناسی ارشد مهندسی مکانیک-تبدیل انرژی

حل عددی جریان نانوسیال داخل کانال موج دار به کمک روش SADI

روی پردازنده گرافیکی

نگارنده: سید محمدحسن تقوی

استاد راهنما

دکتر پوریا اکبرزاده

استاد مشاور

دکتر حسین محمودی داریان

شهریور ۱۳۹۸

شماره: ۳۰۲۹۸/۱۲۱
تاریخ: ۹۸/۶/۲۴

باسمه تعالی



مدیریت تحصیلات تکمیلی

فرم شماره (۳) صورتجلسه نهایی دفاع از پایان نامه دوره کارشناسی ارشد

با نام و یاد خداوند متعال، ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد آقای سید محمد حسن تقوی با شماره دانشجویی ۹۵۰۳۵۵۴ رشته مهندسی مکانیک گرایش تبدیل انرژی تحت عنوان حل عددی جریان نانوسیال داخل کانال موج دار به کمک روش SADI روی پردازنده گرافیکی که در تاریخ ۱۳۹۸/۰۶/۱۷ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

<input type="checkbox"/> مردود <input checked="" type="checkbox"/> قبول (با درجه: <u>تالیف</u>)			
نوع تحقیق: <input checked="" type="checkbox"/> نظری <input type="checkbox"/> عملی			
امضاء	مرتبه علمی	نام و نام خانوادگی	عضو هیأت داوران
	دانشیار	دکتر پوریا اکبرزاده	۱- استاد راهنمای اول
_____	_____	_____	۲- استاد راهنمای دوم
<u>عضو حضور در جلسه دفاع</u>	استادیار	دکتر حسین محمودی داریان	۳- استاد مشاور
	استادیار	دکتر احمد نظری	۴- نماینده تحصیلات تکمیلی
	دانشیار	دکتر علی جباری مقدم	۵- استاد ممتحن اول
	استاد	دکتر محمدحسن کیهانی	۶- استاد ممتحن دوم

نام و نام خانوادگی رئیس دانشکده: دکتر محمدحسن شاه مردان

تاریخ و امضاء و مهر دانشکده:

تصوه: در صورتی که کسی مردود شود حداکثر یکبار دیگر (در مدت مجاز تحصیل) می تواند از پایان نامه خود دفاع نماید (دفاع مجدد نباید زودتر از ۴ ماه برگزار شود).



تقدیم اثر

تقدیم به مهربان فرشتگانی که:

بخطات ناب باور بودن، لذت و غرور دانستن، جسارت خواستن، عظمت رسیدن و تمام تجربه‌های یکتا و زیبای زندگی ام، مدیون

حضور سبز آن هاست

تقدیم به خانواده‌ی عزیزم.

تشکر و قدردانی

از خدای متعال سپاس گزارم که توفیق کسب علم و دانش را به من عطا فرمود تا بتوانم این مرحله از علم آموزی را با موفقیت به پایان برسانم. از خانواده‌ی عزیزم به خاطر حمایت و محبت‌های بی‌دینگی که نسبت به من داشته و دارند، کمال تشکر و سپاس را دارم. از استاد ارجمندم، جناب آقای دکتر پوریا اکبرزاده به خاطر راهنمایی‌های ارزشمند و زحمات ایشان در کلیه‌ی مراحل انجام پایان‌نامه تقدیر و تشکر می‌نمایم. همچنین از آقای دکتر حسین محمودی داریان به پاس زحمات و راهنمایی‌های ارزشمند ایشان در انجام این مطالعه تشکر می‌نمایم. در انتها از تمامی اساتید محترم دانشکده‌ی مکانیک و دانشگاه صنعتی شاهرود که توفیق ساگردی‌شان را داشتم، سپاس گزارم. از خدای متعال سلامت و توفیق برای ایشان مسألت دارم.

سید محمد حسن تقوی

شهریور ۱۳۹۸

تعمیرنامه

این جانب سید محمدحسن تقوی دانشجوی دوره‌ی کارشناسی ارشد رشته‌ی مهندسی مکانیک گرایش تبدیل انرژی دانشکده‌ی مهندسی مکانیک و مکاترونیک دانشگاه صنعتی شاهرود نویسنده‌ی پایان‌نامه‌ی حل عددی جریان نانو سیال داخل کانال موج‌دار به کمک روش SADI روی پردازنده‌ی گرافیکی تحت راهنمایی دکتر پوریا اکبرزاده متعهد می‌شوم.

- تحقیقات در این پایان‌نامه توسط این جانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورداستفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه‌ی حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می‌باشد و مقالات مستخرج بانام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده‌اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه‌ی مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آن‌ها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه‌ی مراحل انجام این پایان‌نامه، در مواردی که به حوزه‌ی اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه‌ی حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

چکیده

افزایش نرخ انتقال حرارت و نوع سیال انتقال‌دهنده‌ی حرارت موضوع بسیاری از تحقیقات صنعتی و مهندسی در دهه‌های اخیر بوده است. تغییر خواص ترموفیزیکی سیال (مانند اضافه کردن نانوذره به سیال پایه) و هندسه‌ی جریان (مانند موج‌دار کردن دیواره‌های کانال)، دو روش رایج برای بهبود نرخ انتقال حرارت است. از این‌رو، در پژوهش حاضر به بررسی جریان و موضوع انتقال حرارت نانوسیال در داخل کانال با دیواره‌ی موج‌دار پرداخته شده است. معادلات حاکم برای حل جریان نانوسیال شامل معادله بقای جرم، بقای ممنتوم و بقای انرژی بوده که با استفاده از روش عددی SADI (Spline ADI) گسسته می‌شود. گسسته‌سازی معادله‌های حاکم بر مسأله، منجر به تشکیل یک دستگاه معادله‌ی سه‌قطری می‌شود. حل سریع این دستگاه معادله یک موضوع چالش‌برانگیز در دینامیک سیالات محاسباتی است که در این پژوهش برای اولین بار جهت حل این دستگاه معادله از پردازنده‌ی گرافیکی استفاده می‌شود. ساختار معماری پردازنده‌های گرافیکی به‌گونه‌ای است که می‌تواند در زمانی کوتاه یک دستورالعمل محاسباتی را روی تعداد زیادی داده به‌صورت موازی و هم‌زمان اجرا کند. نتایج حاصل، افزایش انتقال حرارت در اثر افزایش دامنه‌ی موج کانال، عدد رینولدز، عدد پرانتل و کسر حجمی نانوسیال را در بردارد. همچنین، حل دستگاه معادلات با اعمال الگوریتم توماس روی پردازنده‌ی گرافیکی در شبکه با اندازه‌ی 1600×1600 نسبت به پردازنده‌ی مرکزی منجر به ضریب تسریع $7/69$ شد.

کلمات کلیدی: کانال با دیواره‌های موج‌دار، روش اسپلاین با جهت متغیر، نانوسیال، پردازنده‌ی گرافیکی، کودا.

لیست مقالات مستخرج از پایان نامه

-۱

-۲

-۳

فهرست مطالب

س	فهرست جداول
ص	فهرست اشکال
ع	فهرست علائم
۱	فصل ۱: مقدمه
۲	۱-۱ مقدمه
۴	۲-۱ نانوسیال
۹	۳-۱ دیواره‌ی موج‌دار
۱۲	۴-۱ پردازش موازی
۲۳	۵-۱ اهداف پایان‌نامه
۲۳	۶-۱ نوآوری
۲۴	۷-۱ مروری بر فصل‌های پایان‌نامه
۲۵	فصل ۲: مقدمه‌ای بر پردازش موازی و پردازنده‌ی گرافیکی
۲۶	۱-۲ مقدمه
۲۶	۲-۲ محاسبات موازی
۲۹	۳-۲ معماری کامپیوتر موازی
۳۱	۱-۳-۲ سیستم حافظه مشترک
۳۲	۲-۳-۲ سیستم حافظه‌ی توزیع یافته
۳۲	۴-۲ شبکه‌های ارتباط داخلی
۳۳	۱-۴-۲ مدل برنامه‌نویسی موازی
۳۴	۲-۴-۲ مدل حافظه مشترک

۳۶مدل انتقال پیام.....۳-۴-۲
۳۷الگوریتم‌های موازی.....۵-۲
۳۸طراحی الگوریتم موازی.....۱-۵-۲
۳۹معیار ارزیابی سیستم موازی.....۶-۲
۴۰یک نمونه الگوریتم موازی.....۷-۲
۴۳معماری پردازنده‌ی گرافیکی.....۸-۲
۴۵اجزای سخت‌افزاری و نرم‌افزاری پردازنده‌ی گرافیکی.....۹-۲
۴۵چندپردازنده.....۱-۹-۲
۴۷رسمان.....۲-۹-۲
۴۷انشعاب.....۳-۹-۲
۴۸انواع حافظه در پردازنده‌ی گرافیکی.....۴-۹-۲
۴۹حافظه‌ی نهان یا کش.....۱-۴-۹-۲
۴۹حافظه‌ی مشترک.....۲-۴-۹-۲
۵۰ثبات‌ها و حافظه‌ی محلی.....۳-۴-۹-۲
۵۰حافظه‌ی سراسری.....۴-۴-۹-۲
۵۳ملاحظات نرم‌افزاری و برنامه‌نویسی روی پردازنده‌ی گرافیکی.....۵-۹-۲
۵۷	فصل ۳: معرفی مسأله، روابط حاکم، الگوریتم حل مسأله، گسسته‌سازی عددی.....
۵۸تعریف هندسه‌ی مسأله.....۱-۳
۵۸فرضیات مسأله و معادلات حاکم.....۲-۳
۵۹معادله‌ی پیوستگی.....۱-۲-۳
۵۹معادلات حرکت.....۲-۲-۳
۶۰معادلات تابع جریان و تاوایی.....۳-۲-۳
۶۱معادله‌ی انرژی.....۴-۲-۳
۶۲خواص نانوسیال.....۵-۲-۳
۶۳تبدیل مختصات معادلات دیفرانسیل حاکم.....۳-۳

۶۴ روش جبری شبکه‌سازی
۶۶ روش شبکه‌سازی با معادلات دیفرانسیل
۶۸ روش بیضوی
۶۹ شرایط مرزی
۷۱ الگوریتم حل مسأله
۷۱ روش ضمنی اسپلاین با جهت متغیر
۷۲ روابط اصلی اسپلاین
۷۳ حل معادله‌های دیفرانسیل پاره‌ای با تقریب اسپلاین
۷۷ حل معادله‌های حاکم با روش (SADI)
۸۲ مزایای روش (SADI)
۸۳ فصل ۴: نتایج
۸۴ مقدمه
۸۵ صحت‌سنجی و اعتبارسنجی
۸۶ شبکه‌ی محاسباتی
۸۸ نتایج عددی
۸۸ بررسی تأثیر دامنه‌ی موج دیواره‌های کانال
۹۴ بررسی تأثیر عدد رینولدز
۹۹ بررسی تأثیر عدد پرانتل
۱۰۲ بررسی تأثیر کسر حجمی نانوسیال
۱۰۶ بررسی تأثیر قطر نانوذره
۱۰۸ مقایسه‌ی عملکرد پردازنده‌ی مرکزی و گرافیکی
۱۲۱ فصل ۵: نتیجه‌گیری و پیشنهادها
۱۲۲ بحث و نتیجه‌گیری
۱۲۳ پیشنهادها

فهرست جداول

- جدول ۱-۲: قابلیت محاسباتی و تراشه و کارت‌های متناظر آن‌ها ۴۴
- جدول ۲-۲: ریز معماری‌های مختلف و قابلیت‌های محاسباتی متناظر با آن‌ها ۴۴
- جدول ۳-۲: پیشرفت و تغییرات ثبات‌ها در طی زمان و در تراشه‌های گرافیکی از نسل‌های مختلف ۵۰
- جدول ۴-۲: محدودیت پردازنده‌ها با قابلیت محاسباتی مختلف از نظر تعداد نخ و بلوک ۵۶
- جدول ۱-۳: خواص ترموفیزیکی سیال و نانو ذره ۶۳
- جدول ۲-۳: مقادیر معلوم $F_{i,j}$, $G_{i,j}$ و $S_{i,j}$ در زمان n ۷۹
- جدول ۳-۳: مقادیر معلوم $F_{i,j}$, $G_{i,j}$ و $S_{i,j}$ در زمان $n + \frac{1}{2}$ ۸۱
- جدول ۱-۴: تأثیر دامنه‌ی موج روی عدد ناسلت متوسط در شرایط $\varphi = 0\%$, $Pr = 6.93$, $Re = 300$ ۹۰
- جدول ۲-۴: تأثیر عدد رینولدز روی عدد ناسلت متوسط در شرایط $\varphi = 0\%$, $Pr = 6.93$, $a = 0.2$ ۹۵
- جدول ۳-۴: تأثیر عدد پرانتل روی عدد ناسلت متوسط در شرایط $\varphi = 0\%$, $a = 0.2$, $Re = 300$ ۹۹
- جدول ۴-۴: تأثیر کسر حجمی نانوسیال آب-مس اکسید روی عدد ناسلت متوسط در شرایط $Pr = 6.93$, $dp = 10\text{ nm}$, $a = 0.2$, $Re = 300$ ۱۰۳
- جدول ۵-۴: تأثیر قطر نانوذره‌ی مس اکسید روی عدد ناسلت متوسط در شرایط $\varphi = 5\%$, $Pr = 6.93$, $a = 0.1$ و $Re = 300$ ۱۰۷

جدول ۴-۶: توابع اجراشده روی پردازنده‌ی گرافیکی ۱۱۶

فهرست اشکال

- شکل ۱-۱: خنک‌کاری یک پردازنده با کلاک بالاتر از ۴ گیگاهرتز توسط نیتروژن مایع با دمای (۱۹۵-) درجه‌ی سانتی‌گراد [۳۳] ۱۳
- شکل ۱-۲: تغییرات سرعت پردازنده‌های مرکزی از ابتدا تاکنون [۳۳] ۱۵
- شکل ۱-۲: شکلی شماتیک از محاسبات سریال ۲۷
- شکل ۲-۲: شکلی شماتیک از محاسبات موازی ۲۷
- شکل ۳-۲: انواع معماری مدل MIMD ۳۱
- شکل ۴-۲: انواع شبکه‌های ارتباطی ایستا ۳۳
- شکل ۵-۲: مدل PRAM برای محاسبات موازی ۳۶
- شکل ۶-۲: مثالی از الگوریتم Sum_EREW با ۸ ورودی ۴۲
- شکل ۷-۲: مقایسه‌ی ساختار معماری پردازنده‌ی مرکزی و گرافیکی ۴۵
- شکل ۸-۲: شکل شماتیک یک چندپردازنده: الف) معماری فرمی ب) معماری تسلا [۳۳] ۴۶
- شکل ۹-۲: انشعاب ایجادشده در هسته‌های پردازشی و نخ‌ها توسط دستور شرطی if [۳۳] ۴۸
- شکل ۱۰-۲: سلسله‌مراتب حافظه در کارت گرافیک [۳۳] ۴۸
- شکل ۱۱-۲: دسترسی غیرهم‌مکان نخ‌های محاسباتی به حافظه‌ی سراسری ۵۲
- شکل ۱۲-۲: دسترسی مناسب و هم‌مکان نخ‌های محاسباتی به حافظه‌ی سراسری ۵۲
- شکل ۱۳-۲: رابطه‌ی بین انواع حافظه و نخ‌های محاسباتی و ارتباط کلی بین میزبان و پردازنده‌ی مرکزی ۵۳

- شکل ۲-۱۴: ساختار اجرای برنامه با کمک پردازنده‌ی گرافیکی و تقسیم‌بندی نخ‌ها در هنگام فراخوانی کرنل‌ها ۵۵
- شکل ۳-۱: شماتیک مدل فیزیکی ۵۸
- شکل ۳-۲: فضای فیزیکی مسأله ۶۴
- شکل ۳-۳: فضای محاسباتی ۶۵
- شکل ۳-۴: شرایط مرزی مسأله ۷۰
- شکل ۴-۱: ضریب اصطکاک پوسته‌ای در شرایط $Re = 300$, $Pr = 6.93$ و $a = 0.2$ ۸۵
- شکل ۴-۲: عدد ناسلت محلی در شرایط $Re = 300$, $Pr = 6.93$ و $a = 0.2$ ۸۶
- شکل ۴-۳: مطالعه‌ی استقلال حل از شبکه برای ضریب اصطکاک پوسته‌ای در شرایط $Pr = 6.93$, $Re = 300$ و $a = 0.2$ ۸۷
- شکل ۴-۴: مطالعه‌ی استقلال حل از شبکه برای عدد ناسلت محلی در شرایط $Pr = 6.93$, $Re = 300$ و $a = 0.2$ ۸۷
- شکل ۴-۵: ضریب اصطکاک پوسته‌ای محلی در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$ ۹۰
- شکل ۴-۶: عدد ناسلت محلی در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$ ۹۱
- شکل ۴-۷: عدد ناسلت متوسط در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$ ۹۱
- شکل ۴-۸: نمودار خط جریان در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$ ۹۲
- شکل ۴-۹: نمودار خطوط هم‌دما در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$ ۹۳
- شکل ۴-۱۰: ضریب اصطکاک پوسته‌ای در شرایط $a = 0.2$, $Pr = 6.93$, $\varphi = 0\%$ ۹۵
- شکل ۴-۱۱: عدد ناسلت محلی در شرایط $a = 0.2$, $Pr = 6.93$, $\varphi = 0\%$ ۹۶
- شکل ۴-۱۲: عدد ناسلت متوسط در شرایط $a = 0.2$, $Pr = 6.93$, $\varphi = 0\%$ ۹۶
- شکل ۴-۱۳: نمودار خط جریان در شرایط $a = 0.2$, $Pr = 6.93$, $\varphi = 0\%$ ۹۷

- شکل ۴-۱۴: نمودار خط هم‌دما در شرایط $a = 0.2$, $Pr = 6.93$, $\varphi = 0\%$ ۹۸
- شکل ۴-۱۵: عدد ناسلت محلی در شرایط $Re = 300$, $a = 0.2$, $\varphi = 0\%$ ۱۰۰
- شکل ۴-۱۶: عدد ناسلت متوسط در شرایط $Re = 300$, $a = 0.2$, $\varphi = 0\%$ ۱۰۰
- شکل ۴-۱۷: عدد ناسلت متوسط برحسب رینولدز برای دامنه‌ی موج مختلف در شرایط $\varphi = 0\%$,
 $Pr = 0.71$ ۱۰۱
- شکل ۴-۱۸: نمودار خطوط هم‌دما در شرایط $Re = 300$, $a = 0.2$, $\varphi = 0\%$ ۱۰۲
- شکل ۴-۱۹: ضریب اصطکاک پوسته‌ای در شرایط $a = 0.2$, $dp = 10\text{ nm}$, $Pr = 6.93$,
 $Re = 300$ ۱۰۴
- شکل ۴-۲۰: عدد ناسلت محلی در شرایط $Re = 300$, $Pr = 6.93$, $dp = 10\text{ nm}$, $a = 0.2$,
 $Re = 300$ ۱۰۴
- شکل ۴-۲۱: عدد ناسلت متوسط در شرایط $Re = 300$, $Pr = 6.93$, $dp = 10\text{ nm}$, $a = 0.2$,
 $Re = 300$ ۱۰۵
- شکل ۴-۲۲: تغییرات عدد ناسلت متوسط برحسب عدد رینولدز در شرایط $a = 0.1$,
 $Pr = 6.93$, $dp = 10\text{ nm}$ ۱۰۵
- شکل ۴-۲۳: عدد ناسلت محلی برای نانوذره‌ی Al_2O_3 در شرایط $a = 0.2$, $dp = 10\text{ nm}$,
 $Re = 300$, $Pr = 6.93$ ۱۰۶
- شکل ۴-۲۴: تغییرات عدد ناسلت محلی برای نانوذره‌ی مس‌اکسید در شرایط $\varphi = 5\%$,
 $Re = 300$ و $a = 0.1$, $Pr = 6.93$ ۱۰۷
- شکل ۴-۲۵: تغییرات عدد ناسلت محلی برای نانوذره‌ی آلومینیوم‌اکسید در شرایط
 $Re = 300$ و $a = 0.1$, $Pr = 6.93$, $\varphi = 5\%$ ۱۰۸
- شکل ۴-۲۶: نمایش مراحل روش ضمنی اسپلین جهت متغیر ۱۱۰

شکل ۴-۲۷: آرایش نخ‌های محاسباتی در مرحله‌ی اول اعمال روش ضمنی اسپلاین جهت متغیر

۱۱۱.....

شکل ۴-۲۸: آرایش نخ‌های محاسباتی در مرحله‌ی دوم اعمال روش ضمنی اسپلاین جهت متغیر

۱۱۱.....

شکل ۴-۲۹: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $\varphi = 0\%$, $Pr =$

۶.93, $a = 0.1$ و $Re = 300$ ۱۱۳.....

شکل ۴-۳۰: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $\varphi = 1\%$, $Pr =$

۶.93, $a = 0.1$ و $Re = 300$ ۱۱۳.....

شکل ۴-۳۱: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $\varphi = 3\%$, $Pr =$

۶.93, $a = 0.1$ و $Re = 300$ ۱۱۴.....

شکل ۴-۳۲: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $\varphi = 0\%$, $Pr =$

۶.93, $a = 0.2$ و $Re = 300$ ۱۱۴.....

شکل ۴-۳۳: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $\varphi = 1\%$, $Pr =$

۶.93, $a = 0.2$ و $Re = 300$ ۱۱۵.....

شکل ۴-۳۴: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $\varphi = 3\%$, $Pr =$

۶.93, $a = 0.2$ و $Re = 300$ ۱۱۵.....

فهرست علائم

\bar{a} (m)	دامنه‌ی موج
a	دامنه‌ی موج بی‌بعد
$C_f \left(\frac{J}{kgK} \right)$	ضریب اصطکاک
$C_p \left(\frac{J}{kgK} \right)$	ضریب گرمایی ویژه
d_p (m)	قطر نانوذره
$h_{i,j}$	طول بازه
$k_{i,j}$	طول بازه
$k_f \left(\frac{W}{mK} \right)$	ضریب انتشار حرارتی سیال
$k_{nf} \left(\frac{W}{mK} \right)$	ضریب انتشار حرارتی نانوسیال
\bar{L} (m)	نصف ارتفاع کانال
$l_{i,j}$	مشتق اول در جهت y
$L_{i,j}$	مشتق دوم در جهت y
$m_{i,j}$	مشتق اول در جهت x
$M_{i,j}$	مشتق دوم در جهت x
Nu	عدد ناسلت
Nu_m	عدد ناسلت متوسط
Pr	عدد پرانتل

Re	عدد رینولدز
\bar{S}	تابع هندسی سطح
$t(s)$	زمان
$\bar{T}(K)$	دما
$\bar{T}_{in}(K)$	دمای ورودی
$\bar{T}_w(K)$	دمای دیوار
$U_m \left(\frac{m}{s}\right)$	سرعت متوسط
$\bar{u}, \bar{v} \left(\frac{m}{s}\right)$	مؤلفه‌های سرعت
u, v	مؤلفه‌های سرعت بی‌بعد
$\bar{x}_s(m)$	مختصات شروع موج
$\bar{x}_e(m)$	مختصات پایان موج
$\bar{x}, \bar{y} (m)$	مختصات کارتیزین
x, y	مختصات کارتیزین بی‌بعد
ξ, η	مختصات انتقال یافته
$\mu_f(Pa. s)$	لزجت دینامیکی سیال
$\mu_{nf}(Pa. s)$	لزجت دینامیکی نانوسیال
$\rho_f \left(\frac{kg}{m^3}\right)$	چگالی سیال
$\rho_{nf} \left(\frac{kg}{m^3}\right)$	چگالی نانوسیال
θ	دمای بی‌بعد
τ	زمان بی‌بعد
φ	درصد حجمی نانوسیال

$\bar{\psi}$	تابع جریان
ψ	تابع جریان بی بعد
$\bar{\omega}$	تاوایی
ω	تاوایی بی بعد

فصل ۱: مقدمه

۱-۱ مقدمه

افزایش نرخ انتقال حرارت و نوع سیال انتقال‌دهنده‌ی حرارت موضوع بسیاری از تحقیقات صنعتی و مهندسی در دهه‌های اخیر بوده است. روش‌های بهبود میزان انتقال حرارت در سیستم‌های حرارتی در دو دسته‌ی کلی فعال^۱ و غیرفعال^۲ طبقه‌بندی می‌شوند. تغییر خواص ترموفیزیکی سیال و هندسه‌ی جریان، دو روش رایج برای بهبود نرخ انتقال حرارت است. راندمان یک انتقال حرارت بین سیال عامل، جسم و محیط پیرامونی بستگی به ویژگی‌های فیزیکی سیال از قبیل هدایت حرارتی، لزجت، چگالی و ظرفیت گرمایی دارد. در روش‌های فعال از توان خارجی استفاده می‌شود. در چنین مواردی می‌توان از میدان الکتریکی، صوتی یا ارتعاش سطوح و سطوح مرتعش برای برهم زدن لایه‌ی مرزی^۳ و افزایش تلاطم استفاده کرد. در حالی که در روش‌های غیرفعال از هندسه‌های خاص و یا مواد افزودنی برای افزایش انتقال حرارت استفاده می‌شود. روش‌های غیرفعال به دلیل هزینه‌ی کمتر معمولاً بیشتر مورد توجه قرار دارند. استفاده از کانال با دیواره‌ی موج‌دار^۴ یکی از روش‌های غیرفعال بهبود انتقال حرارت است. این کانال‌ها با افزایش اختلاط سیال به وسیله‌ی سطوح موج‌دار، انتقال حرارت را افزایش می‌دهند. پایین بودن مشخصه‌ی هدایت حرارتی، اغلب مهم‌ترین محدودیت سیالات در موضوع انتقال حرارت است که برای رفع این محدودیت از نانوسیال^۵ به دلایل ویژگی‌های مناسب ترموفیزیکی آن‌ها، به‌عنوان سیال انتقال حرارت استفاده می‌شود. معادلات حاکم برای حل جریان نانوسیال در داخل کانال موج‌دار شامل معادله‌ی بقای جرم، بقای ممنتوم و بقای انرژی است که برای حل عددی آن‌ها در بسیاری از مسائل مهندسی، گسسته‌سازی معادلات حاکم منجر به تشکیل یک دستگاه معادله‌ی سه‌قطری یا دستگاه معادله‌ی سه‌قطری-بلوکی می‌شود. روش عددی اسپلاین با جهت متغیر^۶ (SADI)

¹ Active Technique

² Passive Technique

³ Boundary Layer

⁴ Wavy Channel

⁵ Nano Fluid

⁶ Spline Alternating Direction Implicit

برای گسسته‌سازی معادله‌های مذکور استفاده می‌شود. مزیت این روش نسبت به دیگر روش‌های عددی این است که شرایط مرزی مورد نیاز توسط روش اسپلاین و با استفاده از روابط اسپلاین مکعبی پایه از شکل پیچیده خارج شده و همچنین دستگاه معادلات سه‌قطری-بلوکی متداول را به سه دستگاه معادلات سه‌قطری کاهش می‌دهد، حل سریع این دستگاه معادلات یک موضوع چالش‌برانگیز در دینامیک سیالات محاسباتی^۱ محسوب می‌گردد. در سال‌های اخیر با توجه به نیاز جوامع علمی به بررسی مسائل مختلف با پیچیدگی بیشتر، نیاز به کامپیوترهایی با سرعت و حافظه‌ی بیشتر احساس می‌شد، بنابراین محققان به استفاده از پردازنده‌های مرکزی^۲ با سرعت بیشتر روی آوردند. در کنار این رویکرد، امروزه بهره‌گیری از پردازنده‌های گرافیکی^۳ نیز، به دلیل عملکرد بسیار بالا و پهنای باند حافظه^۴ی زیاد، جایگاه ویژه‌ای را در شبیه‌سازی پدیده‌ها و محاسبات علمی به خود اختصاص داده‌اند. ساختار معماری پردازنده‌های گرافیکی به‌گونه‌ای است که می‌تواند در زمانی کوتاه یک دستورالعمل محاسباتی را روی تعداد زیادی داده به‌صورت موازی و هم‌زمان اجرا نماید. به دلیل موازی بودن ذاتی پردازنده‌های گرافیکی، در سال ۲۰۰۶ سازندگان این تراشه‌ها این ایده را ارائه دادند که اگر پردازنده‌های گرافیکی برای انجام محاسبات پیچیده‌ی وضوح تصویر توانا است دلیل و مانعی وجود ندارد که در محاسبات عددی و علمی دیگر نیز کاربرد نداشته باشند. این تفکر باعث ایجاد مفهوم پردازنده‌های گرافیکی همه‌منظوره شد. امروزه رویکردهای مختلفی برای استفاده از پردازنده‌های گرافیکی همه‌منظوره تعریف شده است. همان‌طور که برنامه‌نویسی موازی برای سیستم‌های دارای حافظه‌ی توزیع‌یافته (مانند کلاسترهای تیغه‌ای و مجموعه‌ی آن‌ها) نیاز به میان‌افزار رابط انتقال پیام^۵ دارد، ارتباط کاربر و پردازنده‌ی گرافیکی نیز به ابزارهایی مانند میان‌افزار کودا^۶ و غیره نیاز دارد. در

¹ Computational Fluid Dynamics (CFD)

² Central Processor Unit (CPU)

³ Graphic Processor Unit (GPU)

⁴ Memory Band Width

⁵ Message Passing Interface (MPI)

⁶ CUDA (Compute Unified Device Architecture)

واقع این ابزارها چهارچوبی هستند که راه استفاده از پردازنده‌های گرافیکی را برای محاسباتی که ارتباطی با نمایش گرافیکی ندارند هموار می‌سازند. در حال حاضر میان‌افزار کودا پرکاربردترین ابزار برنامه‌نویسی بر پایه‌ی پردازنده‌ی گرافیکی است. کودا یک محیط نرم‌افزاری برای محاسبات موازی و مقیاس‌پذیر است که بر پایه‌ی زبان C/C++ طراحی شده است. میان‌افزار کودا مختص پردازنده‌های ساخته‌شده توسط شرکت انویدیا^۱ است.

۱-۲ نانوسیال

نانوسیال‌ها که از توزیع نانوذرات^۲ (ذرات با ابعاد نانو) در سیالات معمولی حاصل می‌گردند، نسل جدیدی از سیالات با خواص حرارتی بسیار زیاد نسبت به سیال پایه به وجود می‌آورند که در صنایع پزشکی و دارویی و سرطانی، درمان سرطان، انرژی، نساجی، خودرو و همچنین به‌منظور توسعه‌ی دستگاه‌های کنترل حرارت در بسیاری کاربردها از جمله وسایل نقلیه‌ی سنگین مورد استفاده قرار می‌گیرند. اندازه‌ی ذرات مورد استفاده در نانوسیالات از ۱ نانومتر تا ۱۰۰ نانومتر است. در تعریفی علمی نانوسیال‌ها متشکل از سوسپانسیون‌هایی هستند که با افزودن ذرات جامد یا فیبرهای با اندازه‌ی کمتر از ۱۰۰ نانومتر به یک سیال پایه به دست می‌آیند. روش‌های متنوعی برای تولید نانوذرات وجود دارد. این روش‌ها به‌طور کلی به سه گروه تقسیم می‌شوند که عبارت است از: چگالش بخاری، سنتز شیمیایی و فرآیندهای حالت جامد.

نانوذرات رایج در تحقیقات علمی پژوهشی شامل مس^۳، آلومینیوم اکسید^۴، تیتانیوم اکسید^۵، طلا^۶، نقره^۷ و اکسیدمس^۱ می‌باشند. در دهه‌های اخیر افزودن ذرات جامد به سیال‌های انتقال‌دهنده‌ی

^۱ Nvidia

^۲ Nanoparticles

^۳ Cu

^۴ Al₂O₃

^۵ TiO₂

^۶ Au

^۷ Ag

حرارت به‌عنوان روش مفیدی برای افزایش نرخ انتقال حرارت شناخته‌شده است [۱ و ۲]. اندازه‌ی ذرات جامد به علت مشکلات ناشی از رسوب و سایش سوسپانسیون‌های جامد-مایع در حد میکرومتر و میلی‌متر در نظر گرفته می‌شود.

در سال ۱۹۹۳ چوی [۳]، پیشنهاد داد که برای بهبود خواص حرارتی سیالات عامل رایج مانند آب، اتیلن گلیکول و روغن صنعتی مواد فلزی با ابعاد نانومتر افزوده شود. وی مخلوطی از سیال پایه و ذرات با ابعاد نانومتر را نانوسیال نامید که در منابع علمی نیز بیشتر به همین نام خوانده‌شده است. اندازه‌ی متوسط ذرات در نانوسیالات می‌تواند از ۱۰ نانومتر تا ۱۰۰ نانومتر یا حتی کوچک‌تر در نظر گرفته شود. چوی عنوان نمود که نانوسیالات می‌توانند به‌عنوان نسل بعدی سیالات عامل باشند، زیرا قابلیت بهتری برای انتقال گرما نسبت به سیالات پایه (سیالات بدون نانوذرات) از خود نشان می‌دهند. خواص نانوسیالات از مخلوط‌های حاوی ذرات با ابعاد میکرونی و میلی‌متری بهتر است زیرا با کوچک‌تر شدن ذرات نه‌تنها نسبت سطح به حجم ذرات بیشتر شده (که خود می‌تواند عاملی در بهبود خواص باشد)، بلکه پایداری مخلوط‌های نانوسیال نیز از مخلوط‌های میکرونی و میلی‌متری بیشتر می‌شود. همچنین حرکت براونی^۲ نانوذرات به سبب جرم کمترشان، بیشتر بوده و باعث افزایش بیشتر ضریب هدایت حرارتی می‌شود. محققان زیادی به مدل‌سازی و شبیه‌سازی عددی میدان جریان و انتقال حرارت نانوسیالات در مجاری و محفظه‌ها پرداخته‌اند. در شبیه‌سازی عددی مسائل نانوسیالات دو دیدگاه وجود دارد. در دیدگاه اول که رایج‌تر است، نانوسیال، تک‌فازی در نظر گرفته می‌شود و اثر نانوذرات در خواص معادل لحاظ می‌شود و در دیدگاه دوم از طرح دوفازی بهره گرفته می‌شود که در تحقیقات کمتر استفاده شده است. طرح تک‌فازی ساده‌تر بوده و از نظر محاسباتی مقرون‌به‌صرفه است ولی دقت کمتری نسبت به طرح‌های دوفازی دارد. رفتار نانوسیال تحت اثر عوامل مختلفی مانند

¹ Cuo

² Brownian Motion

حرکت براونی، بلوری شدن در سطح مشترک سیال-جامد، زنجیره‌ای شدن نانوذرات و اصطکاک بین سیال و ذرات جامد است. گفتنی است که توصیف ریاضی و شبیه‌سازی تمامی این موارد از مشکلات مدل‌سازی عددی نانوسیالات است [۳ و ۴].

تاکنون مطالعات گسترده‌ای در مورد انتقال حرارت جابجایی نانوسیال به‌صورت اجباری و طبیعی توسط محققان زیادی صورت گرفته است. در سال ۱۹۹۵ چوی [۳] به‌صورت آزمایشگاهی ثابت کرد که سوسپانسیون نانوذرات با مقیاس طولی معمولی ۱ تا ۵۰ نانومتر با ضریب هدایت حرارتی بالا، ضریب هدایت مؤثر و ضریب انتقال حرارت جابجایی سیال پایه را افزایش می‌دهد. چوی و همکارانش [۵] در سال ۲۰۰۱ نشان دادند که مقدار کم اضافی (کم‌تر از ۱ درصد حجمی) نانوذرات در انتقال حرارت جابجایی مایعات، هدایت گرمایی سیال را در حدود ۲ برابر افزایش می‌دهد. خانافر و همکاران [۶] در سال ۲۰۰۳ نانوسیال آب-اکسید مس را در یک محفظه‌ی بسته‌ی مربعی مورد بررسی قرار دادند و گزارش نمودند که انتقال حرارت با افزایش درصد حجمی نانوذرات در هر عدد گراشف افزایش می‌یابد. روی و همکاران [۷] برای اولین بار در سال ۲۰۰۴ تحقیقات عددی را به‌منظور افزایش انتقال حرارت به‌وسیله‌ی نانوسیال آلومینیوم اکسید و آب در جریان آرام محوری برای یک سیستم خنک‌کننده انجام دادند. نتایج نشان داد که با افزودن نانوسیال ضریب هدایت گرمایی سیال پایه افزایش می‌یابد در نتیجه یکی از مؤثرترین روش‌ها برای افزایش انتقال حرارت در لوله‌ها استفاده از نانوسیال است. مایگا و همکاران [۷ و ۸] در سال ۲۰۰۴ خواص حرارتی و هیدرودینامیکی نانوسیالات عبوری از لوله‌ای که به‌صورت یکنواخت حرارت داده می‌شود را در جریان آرام با مدل تک‌فازی به‌صورت عددی بررسی نموده‌اند. نتایج نشان داده است که افزودن نانوذرات، انتقال حرارت را نسبت به سیال پایه به طرز قابل‌توجهی افزایش می‌دهد. معایب استفاده از نانوسیالات در انتقال حرارت نیز توسط مایگا و همکاران [۹] در سال ۲۰۰۵ مورد بررسی قرار گرفت. نتایج نشان داد که افزودن نانوذرات سبب می‌شود تا تنش برشی دیواره زیادتر شود که این افزایش با نسبت حجمی نانوذرات

نسبت مستقیم دارد. در سال ۲۰۰۶ بانجیورنو [۱۰] بیان کرد که افزایش بیش تر در هدایت حرارتی به علت حضور دو اثر مهم پخش براونی^۱ و پخش ترموفورتیک^۲ نانوذرات است. ابوندا و همکاران [۱۱] در سال ۲۰۰۸ نشان دادند که افزایش انتقال حرارت جابجایی طبیعی به طور عمده وابسته به مقدار عدد رایلی^۳ است. مثلاً برای عدد رایلی 10^4 افزایش نانوذرات اثری بر انتقال حرارت ندارد درحالی که برای اعداد رایلی بالاتر مؤثر است.

بررسی نانوسیال با خواص متغیر (لزجت و ضریب هدایت حرارتی متغیر با دما) در یک محفظه‌ی بسته توسط ابوندا و همکاران [۱۱] در سال ۲۰۰۸ با استفاده از تقریب بوزینسک^۴ انجام شد. آن‌ها دریافتند که در اعداد رایلی بالا عدد ناسلت^۵ متوسط به مدل لزجت متغیر بیشتر از مدل ضریب هدایت حرارتی متغیر وابسته است. آن‌ها گزارش دادند که برای نانوسیال آب - اکسید آلومینیوم در اعداد رایلی بالا با افزایش درصد حجمی نانوذرات بالای ۵ درصد، عدد ناسلت کاهش و در اعداد رایلی پایین افزایش می‌یابد. برای نانوسیال آب - اکسید مس، عدد ناسلت در اعداد رایلی بالا به طور پیوسته کاهش می‌یابد و در اعداد رایلی پایین به تغییرات کسر حجمی نانوذرات حساس نیست. ابوندا [۱۲] در سال ۲۰۱۰ نشان داد که افزایش انتقال حرارت در جابجایی طبیعی اساساً به اندازه‌ی عدد رایلی وابسته است. در حقیقت، برای یک عدد رایلی معین، مثل 10^4 ، انتقال حرارت به نانوذرات وابسته نیست. درحالی که مقادیر بالاتر عدد رایلی، افزایش در انتقال حرارت را نشان می‌دهد. به طور مفهومی، انتقال حرارت جابجایی طبیعی به وسیله‌ی خواص نانوسیال نظیر لزجت و ضریب انتقال حرارت هدایت تحت تأثیر قرار می‌گیرد. فریدون‌مهر و همکارانش [۱۳] در سال ۲۰۱۴ پدیده‌ی جابه‌جایی طبیعی ناپایای مغناطیسی روی یک سطح کشیده‌ی عمودی را مورد مطالعه قرار دادند و از روش عددی رانج کوتائ^۶

¹ Brownian Diffusion

² Thermophoretic Diffusion

³ Rayleigh Number

⁴ Boussinesq Approximation

⁵ Nusselt Number

⁶ Runge-Kutta

مرتبه‌ی چهارم که بر پایه‌ی روش شوتینگ^۱ است، برای حل معادلات حاکم استفاده کردند. نتایج به‌دست‌آمده نشان داد که با کاهش کسر حجمی نانوسیال، عوامل ناپایا و میدان مغناطیسی یا افزایش اثرات شناوری^۲، ضریب اصطکاک پوسته‌ای افزایش می‌یابد. همچنین با افزایش کسر حجمی نانوسیال، عوامل ناپایا و میدان مغناطیسی یا کاهش اثرات شناوری، عدد ناسلت محلی افزایش می‌یابد. نتایج نشان داد که انتخاب نانوذرات اکسید آلومینیوم و مس باعث ایجاد کمترین و بیشترین ضریب اصطکاک پوسته‌ای و انتخاب نانوذرات مس و اکسید تیتانیوم باعث ایجاد بیشترین و کمترین عدد ناسلت محلی می‌شوند. اودین و همکارانش [۱۴] در سال ۲۰۱۵ به بررسی جریان در یک میکروکانال^۳ ذوزنقه‌ای شکل به همراه نانوذره و در حالت خاص گرمای یکنواخت از جداره‌ی پایین جسم پرداختند و دریافتند که انتقال حرارت و انتقال جرم در این حالت خاص دارای بیشترین مقدار عددی است. رحمان و همکارانش [۱۵] در سال ۲۰۱۶ انتقال حرارت جابه‌جایی طبیعی ناپایا در یک نانولوله‌ی کربنی با سیال پایه‌ی آب و یک حفره‌ی داخلی به همراه یک گرم‌کن درونی را با روش عددی المان محدود تحلیل کردند. نتایج نشان داد که قدرت جریان با افزایش کسر حجمی نانوسیال و برای تمامی مقادیر عدد رایلی کاهش می‌یابد و همچنین گرادیان دما با افزایش زمان بدون بعد افزایش و بیشینه‌ی مقدار گرادیان دما با افزایش عدد رایلی کاهش می‌یابد. افسه و همکارانش [۱۶] در سال ۲۰۱۷ به‌صورت آزمایشگاهی به اندازه‌گیری و بهینه‌سازی هدایت حرارتی نانوسیال هیبرید با سیال پایه‌ی اتیلن گلیکول و نانوذره‌ی سیلیسیم اکسید^۴ با قطر متوسط ۲۰ تا ۳۰ نانومتر، کسر حجمی ۰/۰۵ تا ۱/۹۵ درصد در نانولوله‌ی کربنی چنددیواری با قطر داخلی و خارجی ۲-۶ و ۵-۲۰ نانومتر در دمای بین ۳۰ تا ۵۰ درجه‌ی سانتی‌گراد پرداختند. نتایج نشان داد نرخ هدایت حرارتی به‌صورت غیرخطی با افزایش دما و غلظت افزایش می‌یابد. بیشترین افزایش نرخ هدایت حرارتی در غلظت ۱/۹۴ درصد و در

¹ Shooting Method

² Bouncy Effect

³ Micro Channel

⁴ SiO₂

دمای ۵۰ درجه‌ی سانتی‌گراد ۲۲/۲ درصد بود. همچنین نتایج نشان می‌دهد که استفاده از نانوسیال‌های هیبریدی از لحاظ هزینه و درصد نرخ هدایت حرارتی مقرون‌به‌صرفه است. میکولا و همکارانش [۱۷] در سال ۲۰۱۸ تأثیر خواص ذرات مانند اندازه و هدایت حرارتی را بر انتقال حرارت جابه‌جایی نانوسیالات به‌صورت آزمایشگاهی موردبررسی قرار دادند. برای این منظور آن‌ها نانوسیال‌های سیال پایه‌ی آب با ذرات مختلفی چون پلی‌استرین^۱، آلومینیوم اکسید^۲، سیلیسیم اکسید^۳ و میسل^۴ را با درصد حجمی ۰/۱ تا ۱/۸ درصد و اندازه‌ی ذرات ۸ تا ۵۸ نانومتر را در یک مبدل حرارتی لوله‌حلقوی مطالعه کردند. نتایج نشان داد افزودن هرگونه نانوذرات تأثیر مطلوبی بر تغییر خواص سیال ندارد به‌گونه‌ای که با افزایش چشمگیر ویسکوزیته تنها افزایش متوسط در هدایت حرارتی به‌دست می‌آید. با این حال، افزایش ویسکوزیته و در نتیجه کاهش انتقال حرارت جابه‌جایی برای نانوسیالات با نانوذرات کوچک کمتر است که نشان می‌دهد ذرات با اندازه‌ی کوچک برای کاربردهای انتقال حرارت جابه‌جایی ارجح هستند.

۳-۱ دیواره‌ی موج‌دار

امروزه موضوع افزایش انتقال حرارت در وسایل گرمایی توجه زیادی را به خود جلب کرده است. افزایش انتقال حرارت می‌تواند به مدیریت و ذخیره‌ی انرژی کمک کند. استفاده از کانال‌های پیشرفته به‌عنوان یک تکنیک توسعه‌یافته‌ی انتقال حرارت می‌باشد که در بسیاری از زمینه‌ها از جمله انتقال حرارت، واکنش شیمیایی، سیستم‌های انرژی، تراشه‌های الکترونیکی و هوافضا مورد استفاده قرار می‌گیرد. در کاربردهای عملی این کانال‌ها به دلیل داشتن اندازه‌ی کوچک، وزن سبک، قیمت ارزان و غیره عملکرد بالایی دارند. یک نوع از مهم‌ترین کانال‌های پیشرفته، کانال با دیواره‌ی موج‌دار^۵ می‌باشد.

^۱ Polystyrene

^۲ Al₂O₃

^۳ SiO₂

^۴ Micelles

^۵ Peristaltic

دیواره‌ی موج‌دار پدیده‌ای است که در علوم فیزیولوژی مختلف و فرآیندهای صنعتی پدیدار شده است و نقش بسیار مهمی در علم فیزیولوژی و انتقال سیالات فیزیولوژی و انتقال حرارت دارد. به همین دلیل در نظر گرفتن ویژگی‌های دیوار مثل تنش و نوسانات در موقعیت‌های کاربردی ضروری هستند [۴ و ۱۸]. کارهای پیشنهادی لاتام، شاپیرو در سال ۱۹۶۴ و لئو را می‌توان از جمله کارهای اولیه در این زمینه معرفی نمود. پس‌از آن، مطالعات روبه‌جلو برای جریان با دیواره‌ی موج‌دار تحت پیکربندی‌های مختلف جریان انجام شد [۱۹ و ۲۰]. بورن و پارک [۲۱] در سال ۱۹۶۹ جریان در کانال با دیواره‌های موج‌دار را با استفاده از سری فوریه^۱ و فرض جریان استوکس^۲ برای تابع جریان بررسی کردند.

اساکو و فوری [۲۲] در سال ۱۹۸۷ نرخ انتقال حرارت را در لوله‌های صاف و کانال‌ها با دیواره‌ی موج‌دار در شرایط یکسان را مورد بررسی قرار دادند. نتایج این مطالعه نشان داد که نرخ انتقال حرارت در کانال‌ها با دیواره‌ی موج‌دار ۴۰ درصد نسبت به لوله‌های صاف بیشتر است. راش و همکارانش [۲۳] در سال ۱۹۹۹ عملکرد انتقال حرارت در کانال‌ها با دیواره‌های به شکل موج‌های سینوسی را به صورت آزمایشگاهی مورد مطالعه قرار دادند. نتایج این مطالعه نشان داد که عدد رینولدز به صورت مستقیم روی عدد ناسلت محلی تأثیر می‌گذارد. وانگ و چن [۲۴] در سال ۲۰۰۲ نرخ انتقال حرارت در جریان آرام و در کانال با دیواره‌های موج‌دار و دما ثابت را مورد مطالعه قرار دادند. نتایج حاصل از این مطالعه نشان داد که با افزایش عدد رینولدز و دامنه‌ی دیواره‌ی موج‌دار، عدد ناسلت و ضریب اصطکاک پوسته‌ای محلی افزایش می‌یابد. نیلپیونگ و وانگویسر [۲۵] در سال ۲۰۰۶ الگوی جریان و افت فشار جریان تک فازی مایع و جریان دوفازی آب‌وهوا را در کانال‌ها با دیواره‌های به شکل موج‌های سینوسی را به صورت آزمایشگاهی به دست آوردند. حیدری و کرمانی [۲۶] در سال ۲۰۱۰ نرخ انتقال حرارت در جریان نانوسیال مس و آب در کانال با دیواره‌های موج‌دار را به صورت عددی مورد بررسی قرار دادند. نتایج آن‌ها

¹ Fourier Serie

² Stokes Flow

نشان داده استفاده از نانوسیال در کانال با دیواره‌های موج‌دار نرخ انتقال حرارت را ۵۰ درصد افزایش می‌دهد. احمد و همکارانش [۲۷] در سال ۲۰۱۳ روش دوبعدی عددی و ویژگی‌های ترموهیدرولیکی^۱ نانوسیال مس و آب در کانال با دیواره‌های به شکل موج‌های مثلثی، سینوسی و دوزنقه‌ای را مورد بررسی قرار دادند. نتایج به‌دست‌آمده نشان داد که با افزایش عدد رینولدز و دامنه‌ی موج دیواره عدد ناسلت و ضریب اصطکاک پوسته‌ای محلی افزایش می‌یابد. همچنین با افزایش نانوسیال عدد ناسلت افزایش می‌یابد. خوش وقت علی‌آبادی و همکارانش [۲۸] در سال ۲۰۱۴ تأثیر شکل کانال روی افزایش نرخ انتقال حرارت در مبدل‌های حرارتی را به‌صورت آزمایشگاهی مورد مطالعه قرار دادند. نتایج آن‌ها نشان داد که نانوسیال سیلیس^۲ بیشترین ضریب گرمایی-هیدرولیکی را دارد. وانکی و همکارانش [۲۹] در سال ۲۰۱۴ تأثیر شکل نانوذرات در جریان جابجایی اجباری نانوسیال و اتیلن گلیکول روی عملکرد کانال با دیواره‌های موج‌دار را بررسی کردند. آن‌ها مشاهده کردند که نانوذره به شکل پلاکت (جسم مسطح و کوچک) بهترین انتقال حرارت را دارد. در سال ۲۰۱۵ اکبر و همکاران [۳۰] نتایج خود را در مورد تحلیل عددی نانولوله‌های کربنی^۳ برای دیواره‌ی موج‌دار در کانال منحنی با انتقال حرارت را به‌صورت زیر منتشر کردند: (۱) گرادیان فشار در مرکز کانال در مقایسه با دیواره‌ها بیشتر است. گرادیان فشار به‌طور مستقیم با عدد گراشف و نرخ جریان و به‌طور معکوس با کمیت انحنا در ارتباط است. (۲) افزایش فشار با عدد گراشف به‌طور مستقیم در ارتباط است و تغییراتش در نانوسیال آب - مس نسبت به آب خالص سریع‌تر است.

نظری و طقرایی [۳۱] در سال ۲۰۱۷ انتقال حرارت نانوسیال اکسید-مس را در کانال متخلخل با دیواره‌های سینوسی شبیه‌سازی کردند. نتایج آن‌ها نشان داد که افزودن نانوذرات به مایع پایه، عدد ناسلت و ضریب انتقال حرارت را افزایش می‌دهد. همچنین آن‌ها دریافتند که در هر طول موج کانال

¹ Thermal-Hydraulic

² SiO₂

³ Carbon Nanotubes

سینوسی، عدد ناسلت محلی در بخش همگرا بیشتر از بخش واگرا است. آژل و همکاران [۳۲] در سال ۲۰۱۸ ویژگی‌های حرارتی و هیدرولیکی جریان نانوسیال آشفته را در دو کانال با دیواره‌ی موج‌دار-ذوزنقه‌ای شکل متفاوت (پروفیل‌های متقارن و زیگزاگ) به صورت عددی بررسی کردند. نتایج آن‌ها نشان داد که پروفیل موج‌دار نسبت به پروفیل صاف و زیگزاگ تأثیر بیشتری بر عملکرد حرارتی دارد. همچنین، آن‌ها نتیجه گرفتند که نانوسیال سیلیسیم-اکسید آب بهترین بهبود در انتقال حرارت را در میان انواع نانوسیال‌های آزمایش‌شده (به‌عنوان مثال اکسید زینک^۱، آلومینیوم اکسید و مس اکسید) دارد.

۱-۴ پردازش موازی

معادلات ناویر استوکس^۲ برای شبیه‌سازی جریان سیالات در رشته‌های مختلف علمی و مهندسی استفاده می‌شوند. این معادلات در تحقیقات تغییرات اقلیمی و پیش‌بینی وضعیت آب‌وهوا، در تحقیقات پزشکی مانند بررسی جریان خون و بررسی چگونگی حرکت داروها در شریان‌ها و در مهندسی در طراحی اتومبیل، هواپیما و غیره کاربرد فراوانی دارند. حل دقیق و شبیه‌سازی مسائل مهندسی مرتبط با جریان سیالات اغلب دارای محاسبات سنگین و وقت‌گیر کامپیوتری هستند که هم به زمان و هم به حافظه‌ی بسیار زیادی احتیاج دارد. این شبیه‌سازی‌ها به راحتی می‌توانند ساعت‌ها و یا روزها وقت بگیرند؛ بنابراین افزایش سرعت در حل مسائل می‌تواند بسیار راه‌گشا باشد و به محققان و طراحان اجازه خواهد داد تا مسائل را با دقت و سرعت بیشتری بررسی کنند.

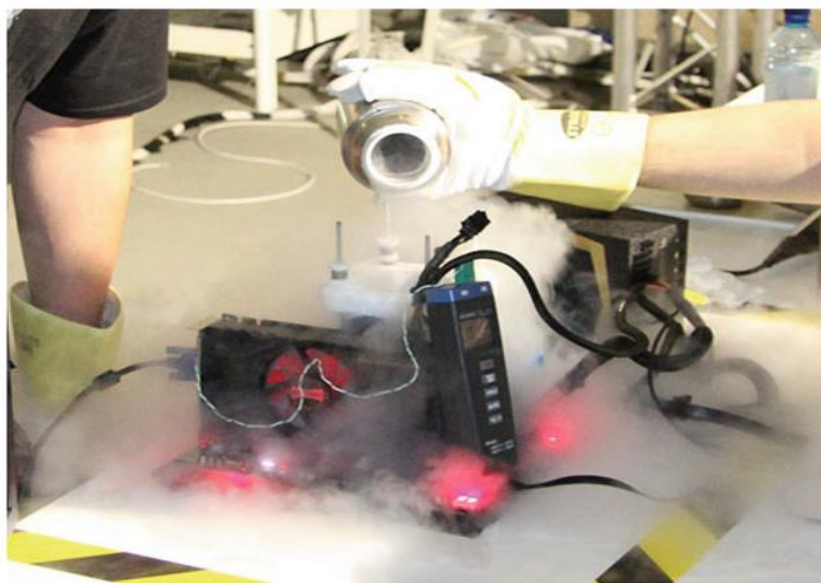
برای دستیابی به سرعت پردازش بیشتر، تولیدکنندگان پردازنده‌ی مرکزی^۳ تا سال ۲۰۰۳ برای افزایش قدرت پردازشی پردازنده، عمدتاً برافزایش تعداد ترانزیستورهای موجود در پردازنده‌ی مرکزی و

1 ZnO

2 Navier Stokes Equations

3 Central Processor Unit(CPU)

افزایش سرعت پردازش هسته (کلاک^۱: سرعت پردازش با واحد هرتز) تکیه داشتند؛ اما در این سال مشاهده کردند که با افزایش سرعت هسته‌ی پردازشی پردازنده، حرارت بسیار بالایی تولید می‌شود که دفع این حرارت به تمهیدات ویژه‌ای نیاز دارد که در واقع برای عموم قابل‌استفاده نخواهند بود و در شرایط خاص پیاده‌سازی و استفاده از چنین سرعتی از لحاظ عملی مشکل خواهد بود. در واقع امروزه این امکان وجود دارد که به کلاک بالاتر از ۴ گیگاهرتز برای یک پردازنده رسید اما خنک نگاه‌داشتن پردازنده کار آسانی نخواهد بود (شکل ۱-۱).



شکل ۱-۱: خنک‌کاری یک پردازنده با کلاک بالاتر از ۴ گیگاهرتز توسط نیتروژن مایع با دمای (۱۹۵-)
درجه‌ی سانتی‌گراد [۳۳]

راه‌حل دیگر در دستیابی به سرعت پردازش بالا استفاده از روش پردازش موازی^۲ است. به بیان ساده در این روش چندپردازنده‌ی معمولی با همکاری یکدیگر به اجرای یک برنامه می‌پردازند که طی این همکاری، برنامه با سرعت بالاتری اجرا می‌شود. برای اینکه بتوان از امکانات و تجهیزات یک سیستم موازی استفاده نمود، نیاز به نرم‌افزارها و سیستم‌عامل‌هایی است که از موازی‌سازی‌ها پشتیبانی کنند،

¹ Clock

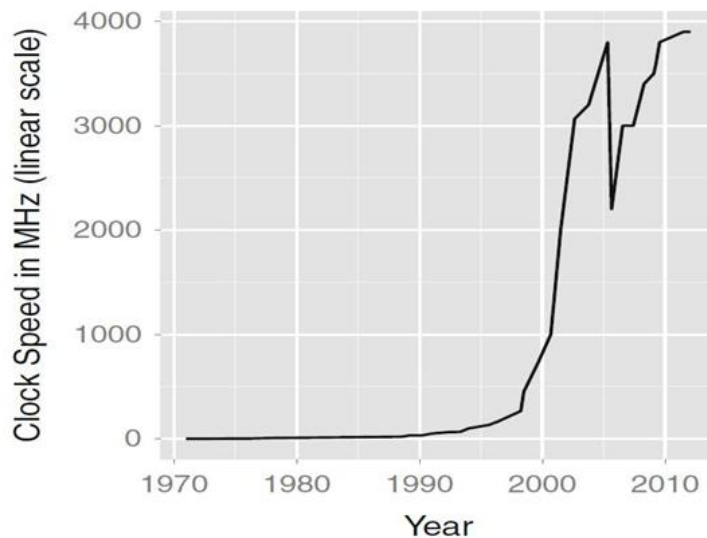
² Parallel Processing

همچنین برای حل مسائلی که قابلیت موازی‌سازی را دارند، لازم است که الگوریتم‌هایی موازی نیز ارائه شوند. اولین گام در برنامه‌نویسی موازی این است که طراح برنامه، بتواند به صورت موازی فکر کند، مسأله را به درستی شناخته و روی قسمت‌هایی از مسأله که قابلیت اجرای هم‌زمان را دارند، تمرکز کند. در غیر این صورت استفاده از پردازش موازی ممکن نخواهد بود یا نتیجه معکوس خواهد داد و به جای اینکه افزایش سرعت در زمان حل مسأله داشته باشیم، کاهش سرعت خواهیم داشت.

رایانه‌های خوشه‌ای^۱ اولین نوع از سامانه‌ها با قابلیت پردازش موازی بودند. رایانه‌ی خوشه‌ای به دسته‌ای از رایانه‌ها (حداقل دو رایانه) اطلاق می‌شود که برای انجام یک هدف مشترک باهم همکاری می‌کنند. اولین سیستم خوشه‌ای در سال ۱۹۹۴ در سازمان تحقیقان فضایی آمریکا پیاده‌سازی شد. این سیستم از ۱۶ رایانه‌ی خانگی که توسط یک شبکه به هم متصل شده بودند به وجود آمد. نکته جالب اینکه هزینه‌ی این پروژه تنها ۴ درصد هزینه‌ی یک پردازنده با توان پردازشی معادل این ۱۶ رایانه بود؛ اما رایانه‌های خوشه‌ای با اینکه امروزه در بسیاری از مراکز علمی و تحقیقاتی به کار برده می‌شوند اما در دسترس همگان نمی‌تواند باشد. بعد از مدتی در سال ۲۰۰۳ ایده‌ی پردازنده‌ی مرکزی با دو یا چند هسته‌ی پردازشی به وجود آمد که هر هسته می‌تواند مستقل از هسته‌ی دیگر با حافظه در ارتباط باشد و دستوراتی مجزا از همتای خود اجرا کند که این نوع پردازش که سیستم‌عامل از آن بهره می‌برد نیز نوعی پردازش موازی است. بدین ترتیب فصل جدیدی روی پردازنده‌های مرکزی گشوده شد. به مرور زمان تعداد هسته‌های پردازشی افزایش یافتند و به ۴، ۸ و ۱۲ هسته رسیدند. بنابراین با تعداد هسته‌ی پردازشی بالاتر نیاز به کلاک کمتری در مقایسه با تعداد هسته کمتر بود. همچنین با تولید چنین پردازنده‌هایی نرم‌افزارهای علمی نیز طوری برنامه‌ریزی شدند که در صورت لزوم بتوانند از قدرت تمام هسته‌ها بهره ببرند؛ اما پس از مدتی مجدداً با پیشرفت نرم‌افزارها و

^۱ Cluster Computers

همچنین پیچیده‌تر شدن محاسبات علمی علاوه بر تعداد هسته‌ی زیاد کلاک بالاتر نیز مورداحتیاج بود. در شکل ۲-۱ تاریخچه‌ی تغییرات کلاک پردازنده‌های مرکزی باگذشت زمان مشخص شده است.



شکل ۲-۱: تغییرات سرعت پردازنده‌های مرکزی از ابتدا تاکنون [۳۳]

با توجه به شکل ۲-۱ سرعت در این سالیان متمادی از ۴ گیگاهرتز بالاتر نرفته است، اگرچه رسیدن به کلاک بالاتر از ۴ گیگاهرتز امکان‌پذیر است اما در این کلاک همان‌طور که گفته شد حرارت ایجادشده باعث می‌شود که سازندگان از یک حدود سرعت پردازش بالاتر نروند. به مشکل تولید حرارت زیاد، دیوار حرارتی می‌گویند. همچنین بعد از سال ۲۰۰۳ یک افت در سرعت کلاک وجود دارد که ناشی از تولید پردازنده‌های چندهسته‌ای است [۳۳].

از طرفی دیگر کارت‌های گرافیکی خود بر مبنای فلسفه‌ی پردازش موازی طراحی شده‌اند و قابلیت بالقوه‌ی پردازش موازی را دارا هستند. پردازنده‌های گرافیکی از تعداد زیادی هسته‌ی پردازشی با سرعت پردازش نه‌چندان بالا و همچنین پهنای باند حافظه‌ی وسیع برای انتقال هم‌زمان حجم بالایی از اطلاعات، تشکیل شده‌اند؛ اما چرا کارت گرافیکی به چنین چیزی احتیاج دارد؟ درواقع جواب این

سؤال در صفحه‌ی نمایشگر نهفته است! هر نمایشگر از تعداد بسیار زیادی نقاط نورانی به نام پیکسل^۱ ساخته شده است. هر پیکسل خود از چند واحد نورانی دیگر به نام زیرپیکسل تشکیل شده است که هرکدام از این زیرپیکسل‌های کوچک قابلیت نمایش یکرنگ را دارند. وظیفه‌ی کارت گرافیک محاسبه‌ی رنگ و شدت رنگ تک‌تک این ریزپیکسل‌ها است. پردازش مربوط به هر پیکسل بسیار ساده است اما تعداد آن بسیار زیاد است. اگر پردازنده‌ی مرکزی بخواهد این عملیات را انجام دهد باید از اولین پیکسل شروع کند و به ترتیب تمام پیکسل‌های صفحه‌نمایش را پوشش دهد که این روش پردازش جوابگوی چنین نیازی نیست و نتیجه‌ی خوبی در بر ندارد. برای مثال زمانی که درایور کارت گرافیک نصب نباشد صفحه نمایشگر برای نشان دادن تصاویر ثابت هیچ‌گونه لرزشی نخواهد داشت اما حتی اگر یک پنجره‌ی ساده جابه‌جا شود پردازنده‌ی مرکزی از پس این کار ساده برنخواهد آمد! پس اینجا بحث پردازنده‌ای مطرح می‌شود که بتواند تعداد زیادی از داده‌های بسیار ساده را پردازش کند که همان کارت‌های گرافیکی و به‌طور خاص پردازنده‌های گرافیکی هستند.

یک پردازنده‌ی گرافیکی از تعداد زیادی پردازنده ساخته شده که می‌تواند چهار عمل اصلی را انجام دهند؛ اما به دلیل مشکلات نرم‌افزاری برای سال‌ها پتانسیل نهفته‌ی این نوع پردازنده‌ها استفاده نشده بود. در گذشته تعداد کمی از محققین به این پتانسیل روی آوردند اما به دلیل مشکلات بسیار زیاد در برنامه‌نویسی، راه به جایی نبردند. مهم‌ترین مشکل پیش روی این افراد برقراری ارتباط با کارت گرافیکی بود که تنها از طریق نرم‌افزارهای خاصی مثل دایرکت‌ایکس^۲ و اپن‌جی‌ال^۳ ممکن بود که این واسطه‌های نرم‌افزاری فقط و فقط مخصوص کار با پیکسل‌ها ساخته شده بودند. پس برای جمع دو عدد باید ابتدا آن‌ها را به دورنگ تبدیل، سپس نتیجه‌ی تلاقی این دورنگ را به دست آورده حاصل را دوباره به عدد تبدیل کرد؛ که این امر دشواری فراوانی برای برنامه‌نویس ایجاد می‌کند. به همین دلیل

¹ Pixel

² DirectX

³ OpenGL

تا سال‌ها استفاده از قدرت پردازنده‌های گرافیکی در محاسبات امکان‌پذیر نبود. اولین تلاش‌ها برای به کار بردن پردازنده‌های گرافیکی در محاسبات علمی و عددی در سال ۲۰۰۱ رخ داد. در این سال لارسن و مک‌آلیستر [۳۴] برای اولین بار نتایج تحقیقی را منتشر کردند که توسط کار با مفاهیم گرافیکی نظیر رنگ‌ها توانستند دو ماتریس را در هم ضرب کنند؛ اما این کار بسیار مشکل است و در محاسبات پیچیده می‌تواند اشکالات و موانع زیادی ایجاد کند. به‌عنوان یک نمونه‌ی دیگر در سال ۲۰۰۵ گالوپو و همکاران [۳۵] جز اولین کسانی بودند که با استفاده از پردازنده‌ی گرافیکی و با روش فاکتورگیری ال‌یو^۱ دستگاه معادلات را سریع‌تر از کد مشابه روی پردازنده‌ی مرکزی حل کردند. اما همچنان کار و ارتباط با کارت و پردازنده‌ی گرافیکی مشکل و سخت بود.

در سال ۲۰۰۶ شرکت انویدیا یک پل ارتباطی جدید بین کاربر و کارت گرافیک ایجاد کرد که موجب تسهیل کار برنامه‌نویسی و محاسبات روی کارت گرافیک شد. این میان‌افزار کودا نامیده شد. با استفاده از میان‌افزار کودا می‌توان در کنار قدرت پردازنده‌ی مرکزی از قدرت پردازنده‌ی گرافیکی در یک برنامه‌ی محاسباتی بهره برد. کودا بستری فراهم می‌کند که بتوان قسمتی از برنامه را که قابلیت پردازش به‌صورت موازی را دارد را به پردازنده‌ی گرافیکی منتقل و پردازش کرد، سپس با انتقال مجدد نتایج ادامه‌ی محاسبات را روی پردازنده‌ی مرکزی انجام داد. به این نوع پردازش، پردازش ناهمگن^۲ می‌گویند. کودا صرفاً برای پردازنده‌های شرکت انویدیا طراحی شده است. از طرفی دیگر میان‌افزار اپن‌سی‌ال^۳ نیز نقش یک رابط را برای کاربر و پردازنده‌های ساخت شرکت ای‌تی‌آی^۴ بازی می‌کند. این تحولات باعث شد عصر جدیدی از پردازش موازی به نام پردازش همه‌منظوره با پردازنده‌های گرافیکی^۵ آغاز شود. پس‌از آن در زمینه‌های مختلف علمی استفاده از پردازنده‌های گرافیکی رواج پیدا کرد، از جمله علوم مربوط به مکانیک سیالات و به‌طور خاص دینامیک سیالات محاسباتی که با توجه

¹ LU Factorization

² Heterogeneous Process

³ Open CL

⁴ ATI

⁵ General Purpose Graphical Processor Unit (GPGPU)

به پردازش‌های حجیم و زمان‌بری که در این علم وجود دارد، توجه محققان در این رشته را نیز به پردازش با پردازنده‌های گرافیکی جلب کرده است.

در دینامیک سیالات محاسباتی با توجه به اینکه بعد از گسسته‌سازی معادلات حاکم یک دستگاه معادلات حاصل می‌شود بخش بزرگی از تحقیقات روی پیاده‌سازی روش‌های حل دستگاه معادلات و اجرای آن‌ها روی پردازنده‌ی گرافیکی انجام گرفته است. با در نظر گرفتن این موضوع که بسیاری از الگوریتم‌های موجود برای حل و شبیه‌سازی یک پدیده‌ی فیزیکی و مکانیکی ترتیبی هستند، یافتن راه‌حلی دیگر یا یک الگوریتم جایگزین یک چالش جدی محسوب می‌شود. البته مفهوم پردازش موازی و استفاده از آن با اتصال مجموعه‌ای از پردازنده‌های مرکزی تا قبل از پیدایش میان‌افزارهایی مانند کودا رواج داشته است، بنابراین الگوریتم‌های موازی مختلفی نیز تا آن زمان ارائه شده بودند اما پیاده‌سازی این الگوریتم‌ها روی پردازنده‌ی گرافیکی با توجه به معماری آن (توضیحات بیشتر در فصل ۲) و انتخاب یا ارائه‌ی یک الگوریتم متناسب با پردازنده‌ی گرافیکی و بهینه‌سازی و پیاده‌سازی، مباحثی هستند که محققان علوم محاسباتی را به خود مشغول کرده‌اند.

همان‌طور که گفته شد، تقریباً تمامی تلاش‌ها برای حل و بررسی عددی مسائل مهندسی منجر به یک سری دستگاه معادلات جبری خطی یا غیرخطی می‌شوند. این دستگاه‌ها می‌توانند متراکم^۱ یا نامتراکم^۲ (تنک) باشند؛ یعنی در حالت اول ماتریس ضرایب دارای درایه‌هایی اکثراً مقداردار است و در حالت دوم اکثر درایه‌های ماتریس صفر هستند مانند ماتریس سه‌قطری^۳. در تحقیقات مختلف روی هر دو نوع این دستگاه‌ها بررسی‌هایی انجام شده است. با توجه به اینکه در ماتریس‌های نامتراکم اکثر درایه‌ها صفر هستند برای انجام یک عمل جبری مانند ضرب عاقلانه نیست که تمام درایه‌ها در یکدیگر ضرب شوند و این هزینه‌ی محاسباتی سنگینی ایجاد می‌کند. پس بهتر است از الگوریتم‌های

¹ Dense

² Sparse

³ TriDiagonal Matrix

متناسب با این‌گونه ماتریس‌ها استفاده شود. در اجرا و پیاده‌سازی عملیات جبری و محاسباتی مختلف روی پردازنده‌های گرافیکی، دستگاه‌های متراکم افزایش سرعت قابل‌توجهی از خود نشان داده‌اند اما در دستگاه‌های نامتراکم افزایش سرعت حل به‌اندازه‌ی دستگاه‌های متراکم نبود.

از آنجایی که دستگاه‌های نامتراکم یکی از کلیدی‌ترین قسمت‌های علوم محاسباتی‌اند و در بسیاری از کاربردهای مهندسی با چنین دستگاه‌هایی مواجه می‌شویم و با توجه به این واقعیت که افزایش سرعت چشم‌گیری هم در پردازش موازی از خود نشان نداده‌اند می‌توان بررسی این نوع از ماتریس‌ها روی پردازنده‌ی گرافیکی را یک مبحث پرچالش در پردازش موازی نامید. پرکاربردترین نوع دستگاه‌های نامتراکم، دستگاه‌های سه‌قطری هستند که موضوع این پایان‌نامه هم به بررسی و پیاده‌سازی دستگاه‌های سه‌قطری به شکل موازی اختصاص دارد.

در مورد تحقیقاتی که در مورد دستگاه‌های نامتراکم ارائه‌شده، برای مثال می‌توان به عملیات مختلف جبری روی ماتریس- بردار نامتراکم^۱ اشاره کرد. در سال ۲۰۰۸ بل و همکاران [۳۶] عملیات جبری روی شکل‌های مختلف ماتریس نامتراکم را توسط میان‌افزار کودا بررسی کردند و نتایج را با پیاده‌سازی روی پردازنده‌ی مرکزی مقایسه کردند. آن‌ها خاطر نشان کردند که استفاده از حافظه‌ی بافت^۲ در پردازنده‌ی گرافیکی می‌تواند تا ۲۵ درصد راندمان را افزایش دهد. باسکاران و همکاران [۳۷] با پیاده‌سازی الگوریتم‌های ماتریس- بردار نامتراکم و به‌طور خاص روی ساختار ذخیره‌سازی ردیف فشرده^۳، افزایش سرعت و نتایج را با توابع کتابخانه‌ای انویدیا مقایسه کرده‌اند.

حل‌گرهای تکراری پیش‌شرط‌سازی شده در حل با پردازنده‌های گرافیکی در مرجع [۳۸] مطالعه شده‌اند. تمام مراحل پیش‌شرط‌سازی و حل‌گرها، روی پردازنده‌ی گرافیکی توسط رویپنگ و همکاران انجام شده است. آن‌ها گزارش کرده‌اند که عملیات جبری ماتریس- بردار نامتراکم روی پردازنده‌ی

^۱ Sparse Matrix-Vector

^۲ Texture Memory

^۳ Compressed Row Storage(CRS)

گرافیکی پرسرعت‌تر از مشابه آن روی پردازنده‌ی مرکزی هستند. همچنین مطالعاتی روی شکل‌های ذخیره‌سازی و اثرات آن‌ها روی پردازنده‌ی گرافیکی انجام شده است. همچنین آن‌ها روش‌های گرادین مزدوج پیش‌شرط‌سازی شده به روش چولسکی ناقص را هم روی پردازنده‌ی گرافیکی پیاده‌سازی کردند و به افزایش سرعت متوسطی (۵ برابر پردازش موازی با پردازنده‌های مرکزی) رسیدند. در سال ۲۰۰۸ مولمیکر و همکاران [۳۹] روش چند شبکه‌ای^۱ را برای حل معادله‌ی پواسن فشار روی پردازنده‌ی گرافیکی پیاده‌سازی کردند. افزایش سرعت حل به حدود ۵۰ برابر سرعت پردازنده‌ی مرکزی نیز رسید. البته در تحقیق اشاره شده است که روش به‌کاربرده شده برای هندسه‌های پیچیده کاربرد ندارد.

چند تحقیقی که تاکنون معرفی شد مربوط به حالت کلی ماتریس‌های نامتراکم بود؛ اما یکی از پرکاربردترین انواع دستگاه‌های نامتراکم، دستگاه معادلات سه‌قطری هستند. یکی از روش‌های حل دستگاه‌های سه‌قطری، حل توسط الگوریتم توماس^۲ است که بر مبنای روش حذفی گوس^۳ است. این الگوریتم یک روش بسیار قدرتمند، سریع و کم‌هزینه از نظر محاسباتی است و استفاده از آن رواج بسیاری دارد؛ اما این الگوریتم یک الگوریتم ترتیبی است و قابلیت اجرای مستقیم روی پردازنده‌ی گرافیکی را ندارد، اما برای حل این موضوع، الگوریتم‌های جایگزینی برای حل معادلات سه‌قطری با رویکرد پردازش موازی در پردازنده‌های گرافیکی نظیر کاهش متناوب^۴ و کاهش متناوب موازی^۵ معرفی شده‌اند. این الگوریتم‌ها و نظایر آن‌ها که به همین منظور ابداع شده‌اند (مانند الگوریتم‌های موجود در مرجع [۴۰]) ابزارهای مناسبی برای بسیاری از محققین هستند که به دنبال تقویت عملکرد کاربرد پردازنده‌های گرافیکی در حل مسائل مختلف می‌باشند.

^۱ MultiGrid

^۲ Thomas Algorithm

^۳ Gauss Elimination

^۴ Cyclic Reduction(CR)

^۵ Parallel Cyclic Reduction(PCR)

ژانگ و همکاران [۴۰] در سال ۲۰۰۹ ابتدا روی قابلیت اجرای الگوریتم‌های کاهش متناوب و کاهش متناوب موازی روی پردازنده‌های گرافیکی بحث کرده‌اند، سپس یک روش ترکیبی از الگوریتم‌های معرفی شده را پیشنهاد و عملکرد هر الگوریتم را روی حل ماتریس‌های سه‌قطری آزمایش کردند. لازم به توضیح است که الگوریتم کاهش متناوب مشکل تداخل حافظه^۱ در حافظه‌ی مشترک^۲ کارت گرافیکی را دارد و همچنین برای به کار بردن الگوریتم‌های کاهش متناوب و کاهش متناوب موازی اندازه‌ی دستگاه سه‌قطری حتماً باید محدود به ۲ⁿ باشد. در سال ۲۰۱۱ گودک و همکاران [۴۱] در مورد محدودیت ابعاد ماتریس در حافظه‌ی مشترک کارت گرافیکی بحث کرده‌اند. آن‌ها نشان دادند که انتقال پیاپی اطلاعات بین حافظه‌ی مشترک و حافظه‌ی سراسری^۳ کارت گرافیکی بر عملکرد الگوریتم تأثیر می‌گذارد. در سال ۲۰۱۱ دیویدسون و همکاران [۴۲] یک الگوریتم چندمرحله‌ای برای حل دستگاه معادلات سه‌قطری با پردازنده‌ی گرافیکی پیشنهاد کردند که نتایج کار آن‌ها نشان می‌داد سرعت الگوریتم پیشنهادی تا ۵ برابر افزایش سرعت در مقابل الگوریتم‌های حل غیرموازی دستگاه معادلات سه‌قطری به دست آورده است. اگلوف [۴۳] الگوریتم کاهش متناوب موازی را برای حل دستگاه معادلات سه‌قطری با اندازه‌ی بسیار بزرگ در معادلات دیفرانسیل با مشتقات جزئی به کار برد و به دلیل استفاده از حافظه‌ی سراسری، ۶۰ درصد کاهش سرعت را گزارش داد. کیم و همکاران [۴۴] به دلیل محدودیت در حافظه‌ی مشترک، پیشنهاد دادند که برای دستگاه‌های بزرگ ابتدا آن‌ها را توسط الگوریتم کاهش متناوب موازی به دستگاه‌های کوچک‌تر تبدیل، سپس این دستگاه معادلات مجزا توسط الگوریتم توماس حل شوند. به کار بستن این ایده و اجرا روی پردازنده‌ی گرافیکی در مقایسه با حل گر ام‌کال^۴ که یک کتابخانه از توابع مفید در محاسبات علمی است، باعث افزایش ۸ تا ۴۹ برابری سرعت حل شد. اصفهانیان و همکاران [۴۵] در سال ۲۰۱۳ با به کار بستن الگوریتم کاهش

¹ Bank Conflict

² Shared Memory

³ Global Memory

⁴ Math Kernel Library (MKL)

متناوب بهینه‌شده به حل و شبیه‌سازی جریان روی استوانه و ایرفویل در دو و سه بعد پرداختند و به افزایش سرعت حل ۱/۹ تا ۱۵/۲ برابر در دو بعد و ۶/۴ تا ۲۰/۳ برابر در سه بعد دست یافتند.

از دیگر تحقیقات در مورد استفاده‌ی پردازنده‌های گرافیکی در دینامیک سیالات محاسباتی می‌توان به موارد زیر اشاره کرد. بولنت توتکن و همکاران [۴۶] بر روی شبیه‌سازی جریان توسط معادلات مرتبه بالا و حل آن بر روی پردازنده‌های گرافیکی بحث کرده‌اند. آن‌ها نشان داده‌اند که عملکرد مؤثر یک الگوریتم پردازش موازی برای شبیه‌سازی عددی معادلات مرتبه‌ی بالا از نظر سرعت حل، بستگی زیادی به سرعت حل گر دستگاه سه‌قطری دارد. در سال ۲۰۱۲ اصفهانیان و همکاران [۴۷] با پیاده‌سازی روش‌های مرتبه بالا کارایی پردازنده‌ی گرافیکی را برای حل معادلات هذلولوی، بررسی کردند. در سال ۲۰۱۴ محمودی و همکاران [۴۸] نیز به بررسی و حل معادلات اوپلر پرداختند، این ارزیابی توسط دو نوع پردازنده‌ی گرافیکی مختلف انجام شد و به حداکثر افزایش سرعت ۱۰۵ برابری نسبت به پردازنده‌ی مرکزی دست یافتند. در سال ۲۰۱۷ محمودی [۴۹] به پیاده‌سازی روش‌های مرتبه‌ی بالا با دو پردازنده‌ی گرافیکی ناهمگن پرداخت که سرعت‌هایی در حدود ۲/۵ برابر در مقایسه با پیاده‌سازی با یک پردازنده‌ی گرافیکی حاصل شد. نتایج نشان داد که برای پردازنده‌های گرافیکی ناهمگن، کد بهینه‌سازی شده برای یک پردازنده‌ی گرافیکی ممکن است که برای پردازنده‌ی دیگر مطلوب نباشد.

روش شبکه‌ی بولتزمان^۱ روشی است که نظر بسیاری از محققین را به‌خاطر ذات موازی‌اش به خود جلب کرده است و برای اجرا روی پردازنده‌ی گرافیکی بسیار مناسب است. تحقیقات بسیاری روی اجرای روش شبکه‌ی بولتزمان به‌صورت موازی انجام‌گرفته است. یک تحقیق در سال ۲۰۰۴ قبل از آنکه کودا پا به عرصه بگذارد در مورد استفاده از پردازنده‌های گرافیکی به‌عنوان شتاب‌دهنده به‌سرعت حل روش شبکه‌ی بولتزمان انجام‌شده [۵۰] و به‌عنوان نمونه در تحقیقی دیگر در سال ۲۰۱۲ رینالدی

¹ Lattice Boltzman Method

و همکاران [۵۱] با اجرای روش شبکه‌ی بولتزمن روی پردازنده‌ی گرافیکی به افزایش سرعت ۱۰۰ برابر نسبت به پردازنده‌ی مرکزی در حل مسأله‌ی جریان درون حفره دست‌یافته‌اند. در سال ۲۰۱۶، رن و چان [۵۲] با پیاده‌سازی روش شبکه‌ی بولتزمن روی پردازنده‌ی گرافیکی، به ضریب تسریع ۲۰ برابری نسبت به اجرای غیر موازی در حل مسأله‌ی انتقال حرارت جابه‌جایی طبیعی با آرایه‌ای از موانع جامد در محفظه دست‌یافته‌اند؛ بنابراین روش شبکه‌ی بولتزمن هم یکی از مباحثی است که پتانسیل بالایی برای اجرای پردازش موازی دارد.

۱-۵ اهداف پایان‌نامه

در این پایان‌نامه حل عددی جریان نانوسیال عبوری از یک کانال موج‌دار روی پردازنده‌ی گرافیکی به کمک روش عددی اسپلین با جهت متغیر (SADI) مورد بررسی قرار گرفته است. در این مطالعه ابتدا معادلات حاکم معرفی می‌شوند و سپس با روش اسپلین با جهت متغیر (SADI) گسسته می‌شوند و سپس برنامه‌ی عددی آن روی پردازنده‌ی مرکزی و پردازنده‌ی گرافیکی نوشته و باهم مقایسه می‌شود.

۱-۶ نوآوری

امروزه دستیابی به پاسخ دقیق یک مسأله در زمانی کوتاه یکی از نیازهای صنعتی و علمی دنیا می‌باشد. روش‌های عددی مرتبه‌ی بالا به دلیل دقت بیشتر آن‌ها در زمان استفاده از تعداد شبکه‌ی کمتر یکی از رویکردهای مورد توجه محققان دانش دینامیک سیالات محاسباتی (CFD) می‌باشد. از طرفی پیشرفت‌های سخت‌افزاری در حوزه‌ی روش‌های عددی نظیر ظهور کارت گرافیکی و پردازنده‌ی گرافیکی (GPU) نیز کمک شایانی در تسریع حل‌های عددی و کاهش زمان و هزینه‌ی محاسباتی کرده‌است. در این پژوهش برای اولین بار از پردازنده‌ی گرافیکی برای حل مسائل فیزیک به کمک

روش عددی مرتبه‌ی چهار SADI استفاده شده و دقت و سرعت رسیدن به جواب آن با پردازنده‌ی مرکزی (CPU) مقایسه شده است.

۱-۷ مروری بر فصل‌های پایان‌نامه

این پایان‌نامه از یک فصل به‌عنوان مقدمه (فصل حاضر) و چهار فصل اصلی و یک بخش جهت معرفی مراجع مورد استفاده تشکیل شده است. در فصل دوم مفاهیم استفاده شده در این مطالعه به تفصیل توضیح داده شده است. در فصل سوم به معرفی مسأله پرداخته شده و الگوریتم حل مسأله ارائه شده است. در فصل چهارم روش‌های حل معادلات سه‌قطری، اعتبار و صحت‌سنجی و سپس نتایج حاصل از این پژوهش ارائه می‌گردد و در فصل پنجم به معرفی نتایج این پایان‌نامه و ارائه‌ی پیشنهادها و توصیه‌هایی برای ارتقای سطح کیفی تحقیق حاضر و انجام مطالعه‌ی جامع‌تر در راستای موضوع این پایان‌نامه، می‌پردازد.

فصل ۲: مقدمه‌ای بر پردازش موزی و
پردازنده‌ی گرافیکی

۲-۱ مقدمه

این فصل مروری اجمالی بر مفهوم پردازش موازی، اهمیت و کاربرد آن، معماری سامانه‌های رایانه‌ای، شبکه‌های ارتباط داخلی و همچنین مروری بر مفاهیم و ملزومات برنامه‌نویسی و توضیحاتی در مورد معماری و روش برنامه‌نویسی پردازنده‌های گرافیکی خواهد بود.

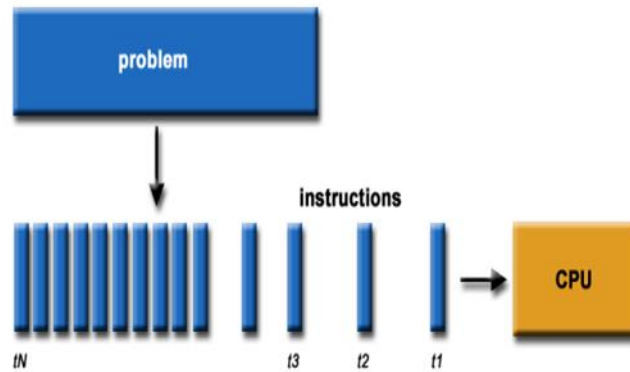
پردازنده‌های گرافیکی مانند پردازنده‌های مرکزی برای انجام محاسبات ایجاد شدند اما برخلاف پردازنده‌ی مرکزی که برای انجام محاسبات عمومی سیستم در نظر گرفته‌شده، پردازنده‌ی گرافیکی به‌طور اختصاصی برای پردازش و ایجاد تصاویر خلق شده است و این امر شامل پردازش حجم عظیمی از اطلاعات در زمان کوتاهی می‌شود که این دستگاه دقیقاً متناسب با همین نیاز طراحی شده است. این نوع معماری و طراحی می‌تواند به محاسبات مهندسی سرعت دهد و همین موضوع باعث شده تا پردازنده‌ی گرافیکی در حیطه‌ی بررسی این پایان‌نامه قرار گیرد.

برای به دست آوردن یک افزایش سرعت مناسب در حل مسائل عددی روی پردازنده‌ی گرافیکی باید با معماری این پردازنده‌ها آشنا بود تا بتوان بهترین و مناسب‌ترین روش برنامه‌نویسی را پیاده‌سازی کرد. آشنایی با انواع حافظه، برنامه‌نویسی در کودا، نخ‌ها، بلوک‌ها و غیره از اهداف این فصل هستند.

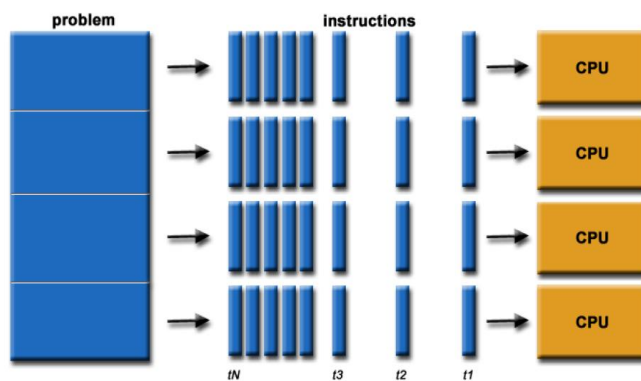
۲-۲ محاسبات موازی

محاسبه‌ی موازی اجرای هم‌زمان قسمت‌های مختلف یک برنامه در چندپردازنده به‌منظور حصول سریع‌تر نتایج است. در پردازش ترتیبی، دستورات به ترتیب در پردازنده اجرا می‌شوند و سرعت اجرا متناسب با سرعت پردازنده است (شکل ۲-۱). در پردازش موازی دستورات در چندپردازنده اجرا می‌شوند ولی سرعت اجرا الزاماً برابر با تعداد پردازنده‌ها در سرعت یک پردازنده نیست (شکل ۲-۲) [۵۳]. محاسبه‌ی موازی در بخش‌های مختلف کامپیوتر اعم از سخت‌افزار و نرم‌افزار شکل می‌گیرد،

بنابراین برای بررسی کلیات محاسبه‌ی موازی باید به جنبه‌های مختلف سخت‌افزاری و نرم‌افزاری آن پرداخت.



شکل ۱-۲: شماتیک از محاسبات سریال



شکل ۲-۲: شماتیک از محاسبات موازی

از پردازش موازی در جهت افزایش قدرت کامپیوترها استفاده می‌شود؛ اما اصلی‌ترین استفاده‌ی آن در حل مسائل علمی و مهندسی است. از جمله‌ی این حوزه‌ها می‌توان به موارد زیر اشاره کرد:

- فیزیک کاربردی، هسته‌ای، ذرات بنیادی، ماده‌ی چگال، گداخت هسته‌ای، فوتونیک و نانو
- اتمسفر، زمین و محیط‌زیست
- فناوری زیستی و ژنتیک

- مهندسی مکانیک؛ از اندام‌های مصنوعی تا سازه‌های فضایی
- مهندسی الکترونیک؛ طراحی مدار، میکروالکترونیک
- علوم کامپیوتر و ریاضی

اما امروزه نه تنها حل مسائل علمی احتیاج به پردازش موازی دارد بلکه برخی نرم‌افزارهای تجاری نیز به کامپیوترهای سریع نیاز دارند. بسیاری از این برنامه‌ها احتیاج به پردازش حجم زیادی از داده‌ها به شکل بسیار پیچیده دارند. از جمله‌ی این برنامه‌ها می‌توان به موارد زیر اشاره کرد:

- پایگاه عظیم داده‌ها و عملیات داده‌کاوی^۱
- موتورهای جستجو و تشخیص پزشکی
- طراحی و شبیه‌سازی دارو
- مدل‌سازی اقتصادی
- گرافیک پیشرفته و فناوری واقعیت مجازی^۲
- فناوری چندرسانه‌ای و شبکه‌ی ویدیویی

از دلایل اصلی استفاده از محاسبات موازی عبارت‌اند از:

- صرفه‌جویی در زمان و هزینه: استفاده از منابع بیشتر در یک کار، زمان انجام کار را کاهش می‌دهد. علاوه بر این استفاده از چندین منبع ارزان به جای استفاده از نوع گران‌قیمت آن‌ها سبب کاهش هزینه‌ها می‌شود.
- حل مسائل بزرگتر: بسیاری از مسائل بزرگ و پیچیده که حل آن‌ها توسط یک کامپیوتر، به‌خصوص در کامپیوتر با حافظه‌ی محدود، غیرعملی و ناممکن است، با استفاده از کامپیوترهای موازی شده قابل‌حل هستند. برای مثال در موتورهای جستجوگر وب و پایگاه‌های داده لازم است تا میلیون‌ها تراکنش در ثانیه پردازش شوند.

¹ Data Mining

² Virtual Reality Technology

- فراهم نمودن هم‌زمانی: در منابع منفرد، در یک‌زمان تنها یک عمل را می‌توان انجام داد، درحالی‌که در منابع محاسباتی چندگانه، چندین کار به‌طور هم‌زمان قابل انجام هستند.

۳-۲ معماری کامپیوتر موازی

پرفردارترین طبقه‌بندی معماری سامانه‌های کامپیوتری، در سال ۱۹۶۶ توسط فلین^۱ تعریف و ارائه شد. طرح طبقه‌بندی فلین^۲ بر اساس جریان اطلاعات صورت گرفته است. اطلاعاتی که پردازنده با آن سروکار دارد را می‌توان به دودسته‌ی داده و دستورالعمل تقسیم کرد.

طبقه‌بندی فلین جریان دستورالعمل یا داده می‌تواند به یکی از صورت‌های تکی یا چندتایی باشد. بر اساس معماری، سامانه‌های کامپیوتری را می‌توان به چهار دسته تقسیم نمود:

۱. جریان یک دستورالعمل، یک داده (SISD): این معماری برای یک کامپیوتر ترتیبی یا سریال است. در این روش تنها یک جریان دستوری توسط پردازنده در طول هر پالس ساعت مورد عمل واقع می‌شود و همین‌طور یک جریان داده در طول پالس ساعت مورد استفاده قرار می‌گیرد. در این روش دستورها به‌طور قطعی انجام می‌شوند و وابسته به عمل پردازنده‌ی دیگر نیستند. کامپیوترهای ترتیبی متعارف ون نیومن در این دسته قرار می‌گیرند. این روش مورد استفاده در بیشتر کامپیوترها، از سامانه‌های قدیمی تا کامپیوترهای شخصی امروزی است.

۲. جریان یک دستورالعمل چند داده (SIMD): این معماری، یکی از انواع کامپیوترهای موازی است. آرایه‌ی پردازنده‌ها^۵ در این دسته قرار می‌گیرند. ماشین‌های SIMD یک واحد کنترل^۶ دارند که

^۱ Flynn

^۲ Flynn's Taxonomy

^۳ Single Instruction-Single Data

^۴ Single Instruction-Multi Data

^۵ Array Processors

^۶ Control Unit

مثل ماشین فن نیومن^۱ عمل می‌کند (دستور اجرا می‌کند) ولی بیش از یک عنصر پردازشگر دارند. واحد کنترل، سیگنال‌های کنترل را برای تمام پردازنده‌ها تولید می‌کند و این عناصر پردازنده، یک عمل مشابه را در طول هر پالس ساعت، روی داده‌های متفاوت انجام می‌دهند. این روش برای حل مسائل خاص که از داده‌هایی با الگوی ثابت پیروی می‌کنند مناسب است. از جمله‌ی این مسائل می‌توان به پردازش تصویر اشاره کرد.

۳. جریان چند دستورالعمل، یک داده (MISD)^۲: در این طرح موازی، یک جریان داده به واحد پردازش داده، ارسال می‌شود. هر واحد پردازش به‌طور مستقل با جریان‌های دستور مستقل روی داده‌ها عمل می‌کند. تاکنون تعداد کمی از نمونه‌های واقعی این کلاس از کامپیوترهای موازی وجود دارند. به‌عنوان نمونه، می‌توان به کامپیوتر آزمایشی کارنگی ملون^۳ اشاره کرد. از جمله کاربردهایی که برای این روش می‌توان مثال زد، اعمال چند فیلتر فرکانسی روی یک جریان سیگنال و دیگری، اعمال چند الگوریتم رمزگذاری در باز کردن یک پیغام کد شده است.

۴. جریان چند دستورالعمل، چند داده (MIMD)^۴: به این دسته چندپردازنده نیز می‌گویند. در این معماری هر پردازنده امکان اجرای چند جریان دستوری جداگانه را دارد و این دستورات روی چند جریان داده‌ی مختلف اعمال می‌شود. سوپرکامپیوترهای امروزی، کامپیوترهای موازی خوشه‌ای^۵، کامپیوترهای چندپردازنده‌ی متقارن^۶ و کامپیوترهای چند هسته‌ای^۷ امروزی از این معماری استفاده می‌کنند. البته بیشتر کامپیوترهای با معماری MIMD از زیر مؤلفه‌های اجرایی SIMD بهره می‌برند.

^۱ Von Neuman

^۲ Multi Instruction-Single Data

^۳ Carnegie-Mellon

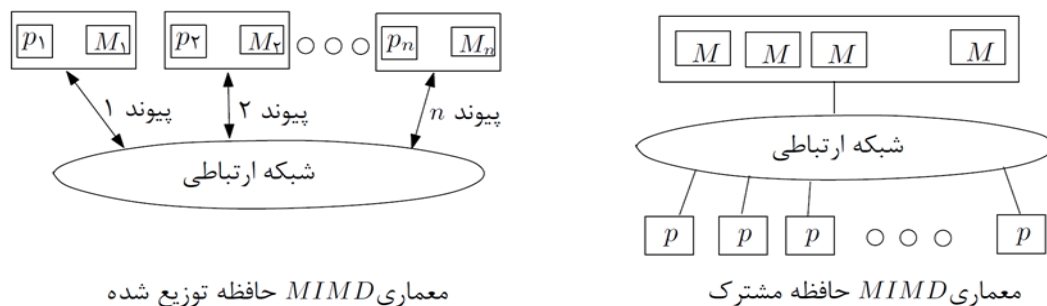
^۴ Multi Instruction-Multi Data

^۵ Cluster Parallel Computers

^۶ Symmetric Multiprocessor(SMP)

^۷ Multi Core Computers

یک ماشین MIMD (دسته‌ی چهارم)، دارای پردازنده‌هایی با واحد کنترل اختصاصی بوده و قادر است دستورالعمل‌های مختلف را بر روی داده‌های مختلف به صورت هم‌زمان اجرا نماید. این روش معمول‌ترین طرح کامپیوتر موازی است و کامپیوترهای مدرن به سمت این معماری حرکت می‌کنند. این نوع معماری‌ها از چندین پردازنده و چندین نوع حافظه که از طریق شبکه‌های ارتباطی به هم متصل شده‌اند، ساخته می‌شوند و به دو گروه اصلی تقسیم می‌شوند: حافظه‌ی مشترک و حافظه‌ی توزیع‌شده (انتقال پیام). شکل ۲-۳ ساختار کلی این دو گروه را نشان می‌دهد. در این شکل، p نشان‌دهنده‌ی پردازنده و M نشان‌دهنده‌ی حافظه است [۳۳].



شکل ۲-۳: انواع معماری مدل MIMD

۲-۳-۱ سیستم حافظه مشترک

در سامانه‌های حافظه مشترک^۱ تمام پردازنده‌ها یک حافظه‌ی سراسری مشترک دارند. ارتباط میان عملیات در حال اجرا روی پردازنده‌های مختلف با نوشتن و خواندن از حافظه‌ی سراسری برقرار می‌شود. همچنین هماهنگ‌سازی و هم‌زمانی تمام پردازنده‌های میانی از طریق این حافظه‌ی مشترک انجام می‌شود. چنانچه تمامی پردازنده‌ها زمان دسترسی یکسانی به هر مکان حافظه داشته باشند، سیستم حافظه مشترک را سیستم چندپردازنده‌ی متقارن می‌نامند. در طراحی سامانه‌های حافظه

^۱ Shared Memory

مشترک برخی مسائل اساسی نظیر کنترل دسترسی و وابستگی داده‌ها، هم‌زمانی، حفاظت و امنیت وجود دارد که باید مورد توجه قرار گیرد [۵۳].

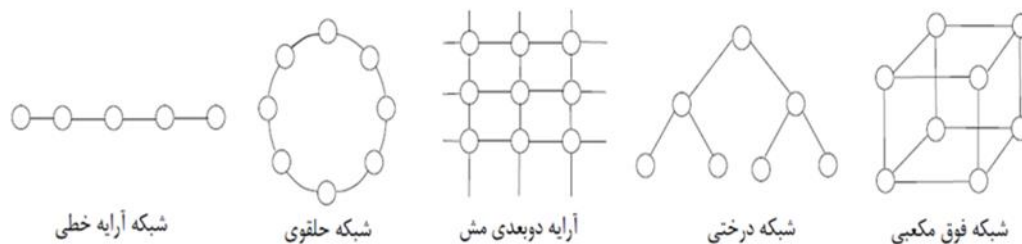
۲-۳-۲ سیستم حافظه‌ی توزیع یافته

سامانه‌های مبتنی بر حافظه‌ی توزیع شده (که سیستم انتقال پیام هم نامیده می‌شود)، گروهی از چندپردازنده‌ها هستند که در آن هر پردازنده به حافظه‌ی محلی خود دسترسی دارد. برخلاف سیستم‌های حافظه مشترک، ارتباطات در این سیستم از طریق دستورات و دریافت پیام صورت می‌گیرد که باید توسط برنامه‌نویس در نرم‌افزار کاربردی نوشته شود. یک گره در چنین سیستمی شامل یک پردازنده و حافظه‌ی محلی آن است. هر گره معمولاً قابلیت ذخیره‌سازی پیام‌ها در میان‌گیر (مکان‌های موقتی از حافظه که پیام‌ها در آنجا منتظر می‌مانند تا ارسال یا دریافت شوند) را دارد و عملیات ارسال/دریافت را هم‌زمان با پردازش انجام دهند. پردازش پیام و انجام محاسبات به صورت هم‌زمان توسط سیستم‌عامل موجود صورت می‌گیرد. سیستم‌های با حافظه‌ی توزیع شده از قابلیت توسعه‌ی بسیار بالایی برخوردار هستند و واحدهای پردازشگر آن‌ها را با روش‌های متنوع، از ساختارهای ارتباطی با معماری خاص تا شبکه‌های پراکنده‌ی جغرافیایی می‌توان به یکدیگر مرتبط نمود. قابلیت توسعه بدان معناست که تعداد پردازنده‌ها را بدون کاهش قابل توجه کارایی، بتوان افزایش داد.

۲-۴ شبکه‌های ارتباط داخلی

شبکه‌های ارتباط سامانه‌های چندپردازنده را می‌توان بر اساس معیارهای مختلفی طبقه‌بندی نمود که یکی از این معیارها توپولوژی شبکه است. توپولوژی مشخص می‌کند که چگونه پردازنده‌ها و حافظه‌ها به پردازنده‌ها و حافظه‌های دیگر موجود در سیستم متصل می‌شوند. به صورت کلی توپولوژی شبکه را می‌توان به دو گروه ایستا و پویا تقسیم کرد. در شبکه‌های ایستا تمامی ارتباطات در زمان طراحی بدون

توجه به این که مورد نیاز هستند یا نه ساخته می‌شوند. در این شبکه‌ها پیام‌ها باید از پیوندهای معینی عبور کنند. شبکه‌های ارتباطی پویا در صورت نیاز اتصالاتی بین دو یا چند گره می‌سازند تا پیام‌ها از این پیوندها عبور کنند. توپولوژی خطی، حلقوی، مش^۱، درختی و فوق مکعبی^۲ که طرح آن‌ها در شکل ۲-۴ مشخص است، نمونه‌هایی از شبکه‌های ارتباطی ایستا هستند و شبکه‌های تک‌باس^۳، چند باس^۴، کراس بار^۵ و چندلایه‌ای^۶ نمونه‌هایی از شبکه‌های ارتباطی پویا هستند [۵۴].



شکل ۲-۴: انواع شبکه‌های ارتباطی ایستا

۲-۴-۱ مدل برنامه‌نویسی موازی

مدل‌های تجریدی به علت طبیعت آرمان‌گرای آن ممکن است در دنیای واقعی مناسب به نظر نرسد. اما ماشین‌های تجریدی در مطالعه‌ی الگوریتم‌های موازی توزیع‌شده و ارزیابی کارایی پیش‌بینی‌شده، مستقل از ماشین‌های موازی بسیار مفیدند.

بدیهی است که اگر اجرای یک الگوریتم روی یک سیستم تجریدی رضایت‌بخش نباشد، پیاده‌سازی آن روی سیستم واقعی بی‌معنی است. با وجود این که مدل‌های تجریدی برخی ملاحظات عملی در سامانه‌های واقعی موازی توزیع‌شده را در نظر نمی‌گیرند، دشواری یافتن محدودیت‌های اجرایی و برآوردهای پیچیدگی را کاهش می‌دهند. بنابراین الگوریتم‌های موازی همواره طبق یک مدل انتخابی

^۱ Mesh

^۲ Hyper Cube

^۳ Single Bus

^۴ Multiple Bus

^۵ Crossbar

^۶ Multistage

طراحی شده و سپس تلاش می‌شود تا مدل به یک برنامه قابل اجرا تبدیل گردد. برای اجرایی کردن یک مدل در دنیای واقعی مجموعه‌ای از زبان‌ها، کامپایلرها، کتابخانه‌ها، سامانه‌های ارتباطی وردی و خروجی موازی به کار گرفته می‌شود. تاکنون مدل‌های موازی مختلفی نظیر حافظه مشترک، داده-موازی، رشته‌ها، انتقال پیام و غیره ارائه شده است که در ادامه دو نوع رایج آن شرح داده می‌شود [۵۵].

۲-۴-۲ مدل حافظه مشترک

در مدل حافظه مشترک، یک برنامه‌ی موازی به قسمت‌های مختلفی تقسیم شده‌اند که به هر قسمت وظیفه^۱ گفته می‌شود. اجرای هر وظیفه به یک پردازنده محول می‌شود و تمام پردازنده‌ها بر روی داده‌های ذخیره شده در حافظه‌ی مشترک عمل می‌کنند. از مکانیزم‌های هم‌زمانی مختلفی نظیر قفل‌ها (روش هم‌زمانی هستند که سیستم‌عامل به کار می‌گیرد تا مانع استفاده‌ی هم‌زمان چندین وظیفه از یک منبع گردد) برای کنترل دستی هم‌زمان پردازنده‌ها به خانه‌های حافظه استفاده می‌شود. در تحلیل الگوریتم‌های موازی نوشته شده در این مدل، معیارهایی نظیر زمان اجرا، تعداد پردازنده‌هایی که الگوریتم برای حل مسأله به کار می‌برد و هزینه‌ی الگوریتم موازی که حاصل ضرب زمان اجرا در تعداد پردازنده‌ها است مورد توجه قرار می‌گیرد.

یکی از مدل‌هایی که در سیستم حافظه مشترک به‌طور گسترده مورد استفاده قرار می‌گیرد، مدل ماشین دسترسی تصادفی موازی (PRAM) است. یک PRAM متشکل از یک واحد کنترل و یک حافظه‌ی سراسری است که برای p پردازنده مشترک است. علاوه بر حافظه‌ی سراسری که ارتباط پردازنده‌ها از طریق آن برقرار می‌شود، هر پردازنده به‌منظور کاهش مراجعات به حافظه‌ی مشترک و تسریع پردازش، حافظه‌ی اختصاصی خود را دارد. **Error! Reference source not found.**

¹ Task

دیگرامی را نشان می‌دهد که اجزای این مدل را شرح می‌دهد. در این مدل فرض می‌شود که به تعداد دلخواه پردازنده در اختیار هست و هیچ دو پردازنده‌ای مستقیماً به یکدیگر متصل نیستند و ارتباطات تنها از طریق خواندن و نوشتن در حافظه‌ی مشترک انجام می‌شود. حالت‌های مختلفی برای عملیات خواندن و نوشتن در حافظه‌ی مشترک وجود دارد که بر این اساس، می‌توان PRAM را به کلاس‌های زیر تقسیم نمود:

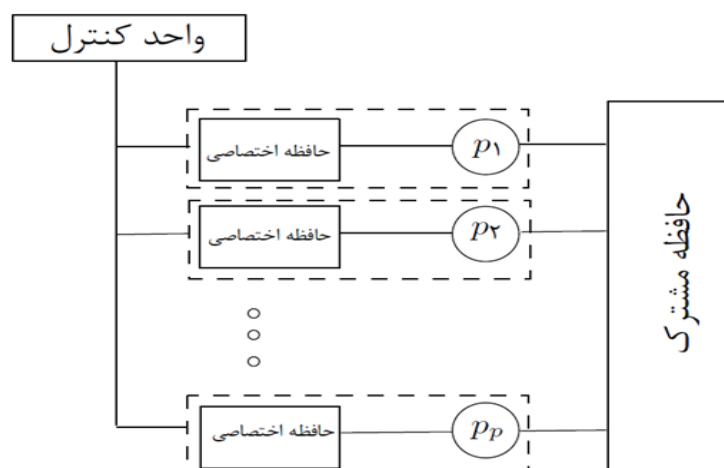
۱. خواندن انحصاری، نوشتن انحصاری (EREW)^۱: دسترسی‌های خواندن و نوشتن در یک مکان حافظه، انحصاری است.
۲. خواندن انحصاری، نوشتن هم‌زمان (ERCW)^۲: چندپردازنده اجازه نوشتن هم‌زمان را دارند ولی خواندن از حافظه انحصاری است.
۳. خواندن هم‌زمان، نوشتن انحصاری (CREW)^۳: اجازه دسترسی‌های هم‌زمان خواندن داده می‌شود ولی دسترسی نوشتن انحصاری است.
۴. خواندن هم‌زمان، نوشتن انحصاری (CRCW)^۴: دسترسی خواندن و نوشتن هم‌زمان مجاز است.

¹ Exclusive-Read Exclusive-Write

² Exclusive-Read Concurrent-Write

³ Concurrent-Read Exclusive-Write

⁴ Concurrent-Read Concurrent-Write



شکل ۲-۵: مدل PRAM برای محاسبات موازی

۲-۴-۳ مدل انتقال پیام

مدل انتقال پیام شامل مجموعه‌ای از پردازنده‌ها است که هر یک تنها حافظه‌ی محلی مختص به خود را دارند و پردازنده‌ها با ارسال و دریافت پیام با یکدیگر ارتباط برقرار می‌کنند. انتقال داده بین پردازنده‌ها نیازمند عملیات متقابل بین پردازنده‌ها است به این معنی که هر عمل ارسال متناظر با یک عمل دریافت داده است. این مدل به دلیل مزیت گوناگونی که دارد به‌طور گسترده در زمینه‌ی محاسبات موازی مورد استفاده قرار می‌گیرد. برخی از این مزیت‌ها عبارت‌اند از:

- سازگاری با سخت‌افزار: این مدل برای استفاده در ابرکامپیوترها و کلاسترها که شامل پردازنده‌های مجزایی هستند که توسط شبکه‌های ارتباطی به یکدیگر متصل‌اند، مناسب است.
- ساختار تابعی: مدل انتقال پیام یک مجموعه‌ی کاملی از توابع را ارائه می‌دهد که امکان کنترل‌هایی را فراهم می‌کند که در مدل‌های دیگر نظیر مدل داده موازی وجود ندارد.
- کارایی: استفاده‌ی مؤثر از پردازنده‌های مدرن امروزی، نیاز به مدیریت قوی بر سلسله‌مراتب حافظه، به‌ویژه حافظه‌های کش دارد. این مدل با اعطای ابزار صریح کنترل به برنامه‌نویس، امکان مدیریت داده‌ها را فراهم می‌آورد.

مشکل اصلی این روش این است که مسئولیت آنچه اتفاق می‌افتد بر عهده‌ی برنامه‌نویس است؛ یعنی برنامه‌نویس باید صریحاً با فراخوانی توابع موجود، داده‌ها را بین پردازنده‌ها توزیع کرده و هم‌زمانی و ارتباط بین پردازنده‌ها را مدیریت کند.

مدل‌های مختلفی از انتقال پیام نظیر CGM، BSP و MPI¹ و غیره ارائه شده است که هر یک کتابخانه‌ی مربوط به خود را دارند.

۲-۵ الگوریتم‌های موازی

بسیاری از الگوریتم‌ها به‌منظور استفاده از سخت‌افزار موازی باید دوباره طراحی شوند. برنامه‌هایی که در یک سیستم تک پردازنده درست کار می‌کند ممکن است در یک محیط موازی هرگز اجرا نشود و خروجی غیرقابل قبولی در برداشته باشد. این بدان علت است که چند کپی از یک برنامه ممکن است باهم تداخل کنند. (برای نمونه تداخل در اختصاص دهی یک مقدار به یک پارامتر در یک محل از حافظه)؛ بنابراین نیاز اصلی یک سیستم موازی، برنامه‌نویسی دقیق مختص به خود است.

طراحی و توسعه برنامه‌های موازی اغلب یک فرآیند دستی محسوب می‌شوند. برنامه‌نویس مسئول تعیین و اجرای واقعی موازی‌سازی است. توسعه‌ی دستی کدهای موازی اغلب فرآیندی زمان‌بر، پیچیده، تکراری و مستعد ایجاد خطا است. در سال‌های اخیر بسیاری از سامانه‌های نرم‌افزاری جهت برنامه‌نویسی کامپیوترهای موازی طراحی شده‌اند که به برنامه‌نویس در تبدیل برنامه‌ی سریال به موازی کمک می‌کند. این دستگاه‌ها هم در سطح سیستم‌عامل و هم در سطح زبان‌های برنامه‌نویسی وجود دارند. آن‌ها باید سازوکاری جهت تقسیم یک مسأله به چندوظیفه و تخصیص این وظایف به پردازنده داشته باشند. چنین سازوکارهایی می‌تواند شامل موازی‌سازی ضمنی و یا موازی‌سازی صریح باشد.

¹ Message Passing Interface

در روش موازی‌سازی ضمنی، سیستم (می‌تواند کامپایلر یا برنامه‌ی دیگری باشد) به‌طور خودکار مسأله را به چندوظیفه تقسیم کرده و هر یک را به پردازنده‌ای اختصاص می‌دهد اما در روش موازی‌سازی صریح، برنامه‌نویس شخصاً مسأله را به چندوظیفه تقسیم کرده و هر یک را به پردازنده‌ای ارجاع می‌دهد. روش موازی‌سازی ضمنی محدود به زیرمجموعه‌ای از کدها (اغلب حلقه‌های تکرار) است و انعطاف‌پذیری کمتری نسبت به موازی‌سازی صریح دارد. همچنین ممکن است نتایج اشتباهی تولید کند و کارایی را کاهش دهد. از این‌رو اغلب برنامه‌نویسی‌های موازی به‌صورت صریح انجام می‌شود.

۲-۵-۱ طراحی الگوریتم موازی

اولین گام در طراحی الگوریتم‌های موازی این است که برنامه‌نویس یاد بگیرد چگونه موازی فکر کند. قسمت‌هایی از مسأله که قابلیت موازی‌سازی دارند را تشخیص داده و پس از انتخاب مدل، بدون توجه به ملزومات اجرایی، ذهن خود را صرفاً بر روی ارائه‌ی بهترین الگوریتم موازی متمرکز کند.

در حل یک مسأله به روش موازی به نکات مختلفی باید توجه نمود. قبل از هر چیز باید مشخص شود که آیا مسأله قابلیت موازی‌سازی دارد؟ به‌عنوان مثال مسأله‌ی ساخت دنباله‌ی فیبوناچی به علت ماهیت وابستگی داده‌ها، یک مسأله‌ی غیر موازی است. پس از این مرحله باید نقاط اصلی مسأله که بیشترین محاسبات در آن قسمت انجام می‌شود، شناسایی شده و حتی‌المقدور بر روی موازی‌سازی این نقاط اصلی^۱ برنامه متمرکز شود. همچنین تنگناهای^۲ مسأله باید مشخص شود یعنی نقاطی که به علت وابستگی داده‌ها و یا نیاز به عملیات ورودی-خروجی، عمل موازی‌سازی متوقف شود.

پس از این نوبت به تجزیه‌ی مسأله به بخش‌های مختلف است که هر بخش می‌تواند تحت عنوان یک وظیفه به یک پردازنده محول شود. دو روش اساسی برای تقسیم وظایف محاسباتی بین پردازنده‌ها

^۱ Hotspot

^۲ Bottlenecks

وجود دارد: تجزیه‌ی دامنه‌ای و تجزیه‌ی تابعی. در تجزیه‌ی دامنه‌ای داده‌های مسئله تقسیم‌شده و هر پردازنده دستورات یکسانی را بر روی داده‌های مربوط به خود اجرا می‌کند. در تجزیه‌ی تابعی دستورات محاسباتی که باید اجرا شوند بین پردازنده‌ها تقسیم می‌شوند. پس از تجزیه‌ی مسئله به وظایف مختلف، چنانچه نیاز به ارتباط بین وظایف باشد، باید از رویه‌های هم‌زمانی و عملیات ارتباطی بین پردازنده‌ها استفاده گردد.

۲-۶ معیار ارزیابی سیستم موازی

یکی از معیارهای ارزیابی سیستم موازی ضریب تسریع^۱ یا $S(p)$ است که به صورت زیر تعریف می‌شود: نسبت زمان موردنیاز برای حل یک مسئله توسط یک پردازنده که با $T_{sequential}$ نمایش داده می‌شود، به زمان موردنیاز برای حل همان مسئله توسط یک سیستم موازی که از p پردازنده‌ی موازی تشکیل شده است. زمان سیستم موازی با $T_{parallel}$ نمایش داده می‌شود.

$$s(p) = \frac{T_{sequential}}{T_{parallel}} \quad (1-2)$$

چنانچه $T_{parallel} = O\left(\frac{T_{sequential}}{p}\right)$ باشد، ضریب تسریع برابر p است و سیستم موازی بهینه است. در عمل، دستیابی به افزایش خطی سرعت (افزایش متناسب با تعداد پردازنده‌ها) بسیار مشکل است. این مشکل از طبیعت ترتیبی بسیاری از الگوریتم‌ها است به طوری که قسمت‌هایی از یک الگوریتم قابل موازی‌سازی و قسمت‌هایی غیرقابل موازی‌سازی هستند. بر طبق قانون آمدال^۲، میزان تسریع یک الگوریتم موازی نسبت به یک الگوریتم ترتیبی به تعداد پردازنده‌ی مورد استفاده بستگی ندارد بلکه به قسمتی از الگوریتم که قابل موازی‌سازی نیست وابسته است. اگر F برابر کسری از کل الگوریتم باشد که قابل موازی‌سازی نیست، و حتماً باید به صورت ترتیبی اجرا شود، ضریب تسریع، طبق قانون آمدال به صورت زیر تعریف می‌شود:

¹ Speed Up

² Amdahl's law

$$s(p) = \frac{1}{F + \frac{F-1}{p}} \quad (2-2)$$

فرض کنید ده درصد یک الگوریتم قابلیت موازی‌سازی ندارد یعنی $F = 10\%$ اما بقیه‌ی الگوریتم به‌طور موازی توسط ۲۰ پردازنده اجرا می‌شود. در این حالت سرعت اجرای برنامه (نسبت به زمانی که روی یک پردازنده اجرا می‌شود) ۲۰ برابر نمی‌شود بلکه مطابق قانون آمدال ضریب تسریع تقریباً ۷ برابر $s(20) = \frac{1}{0.1 + \frac{0.1-1}{20}} \cong 7.0$ می‌شود.

معیار دیگری که می‌تواند در ارزیابی سیستم‌های موازی به کار گرفته شود، کارایی^۱ یا $E(p)$ است که برابر با نسبت هزینه‌ی اجرای یک الگوریتم در سیستم ترتیبی به هزینه‌ی اجرای همان الگوریتم در یک سیستم موازی است. هزینه‌ی اجرای الگوریتم برابر حاصل ضرب زمان اجرا در تعداد پردازنده‌های مورد استفاده است.

$$E(p) = \frac{1 \times T_{sequential}}{p \times T_{parallel}} = \frac{s(p)}{p} \quad (3-2)$$

۲-۷ یک نمونه الگوریتم موازی

در این بخش برای مسأله‌ی ساده‌ی محاسبه‌ی جمع n عدد، الگوریتمی موازی در مدل حافظه مشترک ارائه می‌شود. این الگوریتم‌ها لزوماً بهینه نیستند و صرفاً برای نمایش تعامل پردازنده‌ها در مدل بیان شده‌اند.

الگوریتم محاسبه حاصل جمع آرایه‌ای از اعداد در مدل EREW PRAM

فرض کنید آرایه‌ی $A[1 \dots n]$ در حافظه‌ی مشترک ذخیره شده است. هدف محاسبه‌ی حاصل جمع این اعداد به کمک $n/2$ پردازنده است. از آنجایی که نیازی به انجام چند عملیات هم‌زمان خواندن و نوشتن روی یک مکان حافظه نیست، حالت EREW انتخاب می‌شود. جمع کل در نهایت در مکان

^۱ Efficiency

$A[n]$ ذخیره می‌شود. برای سادگی n توان صحیحی از ۲ در نظر گرفته می‌شود. الگوریتم در $\log n$ تکرار به شرح زیر عمل می‌کند. در تکرار اول تمامی پردازنده‌ها فعال هستند. در تکرار دوم تنها نیمی از پردازنده‌ها فعال خواهند بود و بقیه‌ی تکرارها هم به همین ترتیب خواهد بود. جزئیات در الگوریتم ۱ تشریح شده است.

الگوریتم ۱: $Sum - EREW$

ورودی: آرایه $A[1 \dots n]$ که در حافظه مشترک ذخیره شده است.

خروجی: جمع عناصر آرایه A

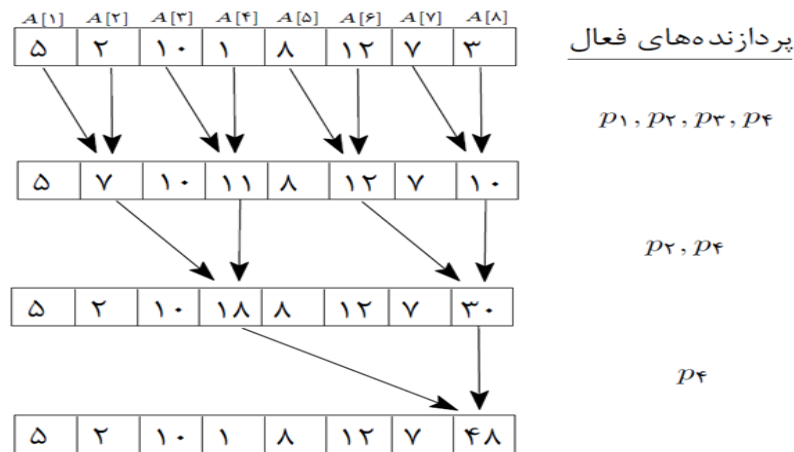
برای مقادیر $i = 1$ تا $i = \log n$ محاسبات زیر را تکرار کن:

۱. برای همه پردازنده‌ها p_j که $1 \leq j \leq \frac{n}{2}$ است به صورت موازی بررسی کن:

اگر شرط $2j \bmod 2^i = 0$ برقرار است، آنگاه جایگذاری زیر را انجام بده:

$$A[2j - 2^{i-1}] + A[2j] \rightarrow A[2j]$$

شکل ۶-۲ نحوه‌ی اجرای الگوریتم را روی یک آرایه با اندازه‌ی ۸ نشان می‌دهد. برای جمع این ۸ عنصر سه تکرار به شرح زیر موردنیاز است. در اولین تکرار، پردازنده‌های p_1, p_2, p_3, p_4 به ترتیب مقادیر ذخیره شده در مکان‌های ۱، ۳، ۵، و ۷ را با اعداد ذخیره شده در مکان‌های ۲، ۴، ۶، و ۸ جمع می‌کنند. در تکرار دوم پردازنده‌های p_2, p_4 به ترتیب مقادیر ذخیره شده در مکان‌های ۲ و ۶ را با اعداد ذخیره شده در مکان‌های ۴ و ۸ جمع می‌کنند. در نهایت، در تکرار سوم پردازنده p_4 مقدار ذخیره شده در مکان ۴ را با مقدار مکان ۸ جمع می‌کند و مکان ۸ مقدار جمع تمام آرایه‌ها خواهد بود.



شکل ۲-۶: مثالی از الگوریتم Sum_EREW با ۸ ورودی

تحلیل پیچیدگی الگوریتم: دستور شماره‌ی الگوریتم به تعداد $\log n$ بار اجرا می‌شود و هر

تکرار زمان پیچیدگی ثابتی دارد. از این‌رو زمان اجرای این الگوریتم $\log n$ است. معیارهای پیچیدگی

الگوریتم به‌صورت زیر خلاصه می‌شود:

زمان اجرا: $T(n) = O(\log n)$ تعداد پردازنده‌ها: $P = \frac{n}{2}$

هزینه اجرا: $C(n) = O(n \log n)$ تسریع: $S(n) = n / \log n$

کارایی: $E(n) = 1 / (\log n)$

که الگوریتم ترتیبی مناسب می‌تواند حاصل جمع لیستی از n عنصر را در $O(n)$ محاسبه کند،

بنابراین الگوریتم موازی ارائه‌شده زمان اجرا را کاهش می‌دهد ولی دارای هزینه‌ی بهینه نیست و تنها

برای زمانی که تعداد پردازنده‌های موجود از $O(n)$ باشد مناسب است. به چنین سیستم‌های موازی که

تعداد پردازنده‌ها تقریباً از مرتبه‌ی ورودی است و هر پردازنده به‌طور میانگین، تقریباً $O(1)$ داده

پردازش می‌کند، به‌اصطلاح پردازش دانه‌ریز^۱ گفته می‌شود.

^۱ Fine Grained

۸-۲ معماری پردازنده‌ی گرافیکی

در انتهای سال ۲۰۰۶ شرکت انویدیا معماری جدیدی از پردازنده‌های گرافیکی ارائه کرد (تراشه‌ی G80). هم‌زمان با این معماری جدید میان‌افزار کودا هم معرفی شد و یک چهارچوب^۱ طراحی شد تا بتوان مستقیماً روی پردازنده‌های گرافیکی برنامه‌نویسی کرد. میان‌افزار کودا بر پایه‌ی زبان سی^۲ طراحی و نوشته‌شده است و شامل یک کامپایلر و کتابخانه‌هایی برای اجرا و همکاری با پردازنده‌ی مرکزی است.

شرکت انویدیا هم‌زمان با ارتقاء نسل و معماری پردازنده‌های خود، میان‌افزار کودا را هم بسط و گسترش می‌دهد. همان‌طور که اولین نسل معماری پردازنده‌های گرافیکی دارای محدودیت‌هایی مانند:

- عدم انجام محاسبات با دقت مضاعف^۳
- نداشتن توانایی در ارتباط با دیگر قسمت‌های سیستم و در نتیجه کاربرد نداشتن توابع ورودی-خروجی مانند *printf*.

بود، کودا نیز در نسخه‌های اولیه‌ی خود بسیار ابتدایی و با توابع کتابخانه‌ای بسیار محدود بود؛ اما با پیشرفت و توسعه‌ی سخت‌افزاری، نسل‌های مختلف معماری پردازنده، با عنوان قابلیت محاسباتی^۴ شناخته می‌شوند. تا این لحظه تا نسخه‌ی ۱۰/۱ میان‌افزار کودا منتشر شده است که دارای توانایی‌های بسیاری است. در جدول ۱-۲ قابلیت محاسباتی تراشه‌های پردازنده‌ی گرافیکی و کارت‌های متناظر آن‌ها آورده شده‌اند. کارت‌های معرفی‌شده در جدول ۱-۲ پرچم‌داران هر نسل از قابلیت محاسباتی هستند [۳۳].

^۱ Framework

^۲ C Language

^۳ Double percision

^۴ Compute Capablity

جدول ۱-۲: قابلیت محاسباتی و تراشه و کارت‌های متناظر آنها

5.2	5.0	3.5	3.0	2.1	2.0	1.3	1.2	1.0	
GM 200	GM107, GM108	GK110	GK104	GF106	GF100 GF110	GT200	GT215	G80	تراشه
GTX 950	GeForce GTX 750	Tesla K20	GTX 680	GTS 450	GTX 480	GTX 285	GT 240	8800 GTX	کارت

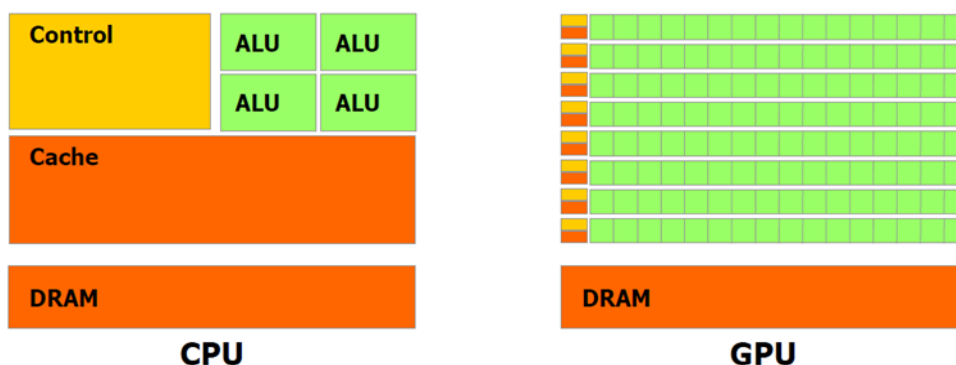
پردازنده‌های گرافیکی از نظر ریز معماری^۱ هم دسته‌بندی جداگانه‌ای دارند که در جدول ۲-۲ مشاهده می‌شوند [۳۳].

جدول ۲-۲: ریز معماری‌های مختلف و قابلیت‌های محاسباتی متناظر با آنها

Maxwell	Kepler	Fermi	Tesla	نام ریز معماری
5.0 - 5.3	3 - 3.7	2, 2.1	1 - 1.2	قابلیت محاسباتی

اگر مقایسه‌ای بین پردازنده‌ی مرکزی و گرافیکی انجام شود، اولین تفاوتی که جلب توجه می‌کند تفاوت در تعداد هسته‌های پردازشی آنها است. در شکل ۲-۷ به صورت شماتیک ساختار پردازنده‌ی مرکزی و گرافیکی نمایش داده شده‌اند. در شکل ۲-۷ خانه‌های سبزرنگ نشان‌دهنده‌ی واحد پردازش عددی و منطقی هستند که در پردازنده‌ی گرافیکی به تعداد بسیار بالا دیده می‌شوند (ویژگی بارز پردازنده‌ی گرافیکی)؛ اما از نظر حافظه‌ی کش و واحد کنترل از پردازنده‌ی مرکزی ضعیف‌تر است.

¹ Micro Architecture



شکل ۲-۷: مقایسه‌ی ساختار معماری پردازنده‌ی مرکزی و گرافیکی

لازم به ذکر است که ALU^۱ در پردازنده‌ی مرکزی در واقع واحدهایی برای اجرا و نظارت بر عملیات منطقی و محاسبات رقمی پردازنده هستند.

۲-۹ اجزای سخت‌افزاری و نرم‌افزاری پردازنده‌ی گرافیکی

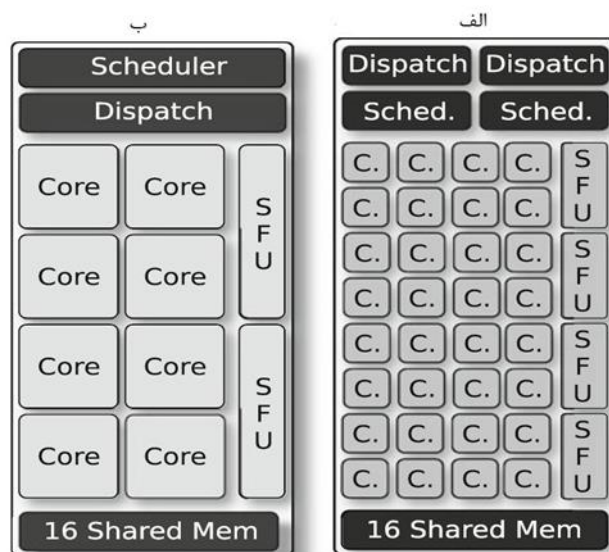
برای بررسی بیشتر، مهم‌ترین اجزای سخت‌افزاری و نرم‌افزاری پردازنده‌ی گرافیکی، در ادامه از نظر گذرانده می‌شوند.

۲-۹-۱ چندپردازنده

در نگاهی دقیق‌تر و موشکافانه‌تر به یک چندپردازنده^۲ گرافیکی، شکلی مانند شکل ۲-۸ خواهد بود. در این شکل در قسمت الف اجزای یک چندپردازنده از نسل تسلا و فرمی در کنار هم نشان داده شده‌اند.

^۱ Arithmetic Logic Unit

^۲ Multi-Processor



شکل ۲-۸: شکل شماتیک یک چندپردازنده: الف) معماری فرمی ب) معماری تسلا [۳۳]

در شکل ۲-۸ دو چندپردازنده از دو نسل مختلف دیده می‌شوند که چندپردازنده‌ی الف از نسل فرمی با ۳۲ هسته و چندپردازنده‌ی ب از نسل تسلا با ۸ پردازنده جریانی^۱ یا هسته‌ی کودا^۲ هستند. چندپردازنده‌های جریانی از پایه‌ای‌ترین المان‌های سخت‌افزاری پردازش موازی با پردازنده‌ی گرافیکی هستند. این هسته‌های پردازشی شباهت‌های زیادی با هسته‌های پردازنده مرکزی دارند. این هسته‌ها می‌توانند کلاک مخصوص به خود را داشته باشند و با کلاک اسمی کل پردازنده‌ی گرافیکی تفاوت داشته باشد. در نتیجه یک چندپردازنده، مجموعه‌ای از هسته‌های پردازشی به همراه حافظه‌ی اشتراکی و یک یا چند واحد دستور^۳ هستند. در ریز معماری‌های مختلف تعداد هسته‌های پردازشی مختلف هستند (شکل ۲-۸). هر چندپردازنده یک سری ثبات دارد که به هسته‌های پردازشی اختصاص داده شدند که نخ‌های پردازشی^۴ (مجموعه‌ای از دستورات-یک واحد نرم‌افزاری) بتوانند روی هر هسته اجرا شوند. هر چندپردازنده در نسل تسلا به‌طور هم‌زمان می‌تواند ۷۶۸ نخ را به شکل

^۱ Streaming Processor

^۲ CUDA Core

^۳ Instruction Unit

^۴ Thread

همزمان اجرا کند و در کل می‌تواند ۱۲,۲۸۸ نخ را همزمان اجرا کند. این عدد برای معماری فرمی ۱۵۳۶ و برای معماری کیپلر ۲۰۴۸ عدد است. همان‌طور که گفته شد چندپردازنده‌ها چند المان سخت‌افزاری دیگر دارند که در مورد آن‌ها صحبت خواهد شد.

۲-۹-۲ ریسمان

هر چندپردازنده نخ‌های مربوط به خود را به شکل یک دستورالعمل-چند داده اجرا می‌کند؛ بنابراین در هر چندپردازنده گروهی از نخ‌ها همیشه و همزمان باهم اجرا می‌شوند که به آن‌ها ریسمان^۱ می‌گویند. استراتژی‌های مختلفی برای اجرای ریسمان روی یک چندپردازنده وجود دارد، برای مثال در تراشه‌ی مدل G80 که ۸ هسته‌ی پردازشی و یک واحد دستور دارد، تنها یک ریسمان قابلیت اجرا دارد؛ اما از نسل تراشه‌ی GF100 به بعد دو ریسمان همزمان می‌توانند اجرا شوند.

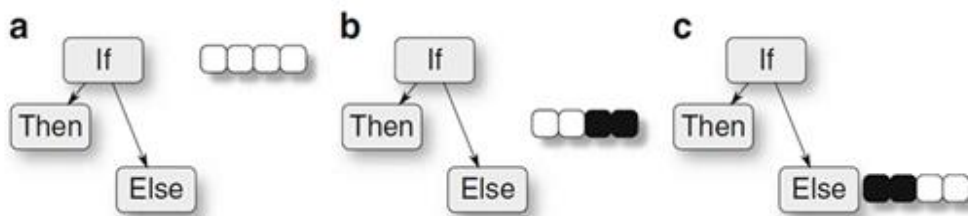
۳-۹-۲ انشعاب

همان‌طور که ذکر شد در یک چندپردازنده، تعدادی هسته زیر نظر یک واحد دستور قرار دارند. اجرای دستورات غیر یکسان در یک‌زمان روی این هسته‌ها امکان‌پذیر نیست. برای اجرای چنین برنامه‌ای که در محاسبات بسیار پیش می‌آید، هسته‌ها مجبور هستند تا منشعب شوند (برنامه‌ی ۱)، گروهی از هسته‌ها قسمت if را انجام دهند درحالی‌که بقیه‌ی هسته‌ها معطل هستند و در قسمت بعدی هسته‌های گروه if معطل و هسته‌های گروه else مشغول محاسبات می‌شوند (شکل ۲-۹).

```
If ( threadIdx/2==0) { //then part }  
Else { // else part }
```

¹ Warp

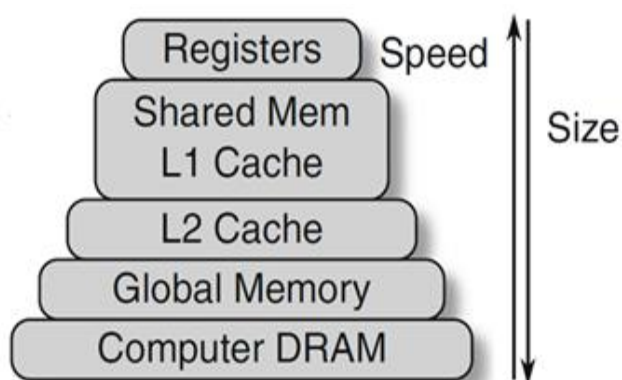
آشکار است که این انشعاب^۱ باعث معطل ماندن گروهی از نخ‌ها و در نتیجه کاهش راندمان شود. فقط برای هسته‌های پردازشی که داخل یک چندپردازنده هستند انشعاب تعریف می‌شود نه برای هسته‌های دو چندپردازنده‌ی جدا از هم [۳۳].



شکل ۲-۹: انشعاب ایجاد شده در هسته‌های پردازشی و نخ‌ها توسط دستور شرطی if [۳۳]

۲-۹-۴ انواع حافظه در پردازنده‌ی گرافیکی

با توجه به تعداد بسیار زیاد هسته‌های پردازشی و نوع معماری خاص آن‌ها، به تناسب به سلسله مراتبی از حافظه برای پاسخگویی به این نوع معماری نیاز است. در شکل ۲-۱۰ سلسله مراتب حافظه در پردازنده‌های گرافیکی نشان داده شده است. در این شکل از بالا به پایین سرعت پاسخگویی کم و مقدار حافظه زیاد می‌شود. در مورد سلسله مراتب این حافظه‌ها بیشتر توضیح داده خواهد شد.



شکل ۲-۱۰: سلسله مراتب حافظه در کارت گرافیک [۳۳]

^۱ Divergence

۲-۹-۴-۱ حافظه‌ی نهان یا گش

این حافظه یک حافظه‌ی بسیار سریع و با مقدار بسیار کم است که روی تراشه‌ی پردازنده نزدیک به هسته‌های پردازشی تعبیه شده که دستور بعدی که قرار است پردازنده پردازش کند را به صورت موقت در خود جای می‌دهد. کاربرد حافظه‌ی نهان طی توسعه‌ی پردازش موازی با پردازنده‌ی گرافیکی بسیار تغییر کرده است. برای مثال در اولین نسل این پردازنده‌ها این حافظه‌ها تنها خواندنی بوده‌اند، اما در نسل فرمی این قابلیت به خواندنی و نوشتنی ارتقا پیدا کرد. در نسل فرمی به بعد حافظه‌ی نهان به دو بخش لایه‌ی ۱ و لایه‌ی ۲ تبدیل شد. در مورد مقدار حافظه هم برای مثال در نسل معماری فرمی برای هر چند پردازنده حافظه‌ی نهان لایه‌ی ۱ تنها ۴۸ کیلوبایت فضا دارد که بین هسته‌ها این مقدار تقسیم می‌شود. همچنین لایه‌ی ۲ نیز حداکثر مقدار ۷۶۸ کیلوبایت دارد.

۲-۹-۴-۲ حافظه‌ی مشترک

هر چند پردازنده به یک حافظه به نام حافظه‌ی مشترک دسترسی دارد که در واقع بخشی از لایه‌ی اول حافظه‌ی نهان است. تمام نخ‌های واقع در یک بلوک یک چند پردازنده می‌توانند مقادیر را از این حافظه بخوانند و بنویسند. اما نخ‌های یک چند پردازنده‌ی دیگر به حافظه‌ی دیگر چند پردازنده‌ها دسترسی ندارند. این حافظه در قسمت‌هایی به نام بانک دسته‌بندی شده‌اند و سرعت ورود و خروج اطلاعات در آن بسیار زیاد است. یکی از نکاتی که در استفاده از این حافظه باید دقت کرد، عدم تداخل و عدم استفاده‌ی هم‌زمان دو هسته از یک بانک حافظه است که بانام تداخل بانک^۱ مطرح می‌شود و باعث افت راندمان و سرعت پردازش می‌شود.

^۱ Bank Conflict

۲-۹-۴-۳ ثبات‌ها و حافظه‌ی محلی

هر نخ یک حافظه‌ی محلی و مخصوص به خود دارد. حافظه‌ی محلی روی حافظه‌ی سراسری است پس حافظه‌ی محلی همان خواص حافظه‌ی سراسری را دارد. ثبات‌ها^۱ روی تراشه‌ی پردازنده قرار دارند و دارای بیشترین سرعت در پاسخگویی هستند. متغیرها وقتی در برنامه‌نویسی با کودا تعریف می‌شوند روی حافظه‌ی سراسری یا محلی قرار می‌گیرند؛ اما کودا به شکل خودکار بعضی از این متغیرها را با توجه به مقدار آن‌ها به ثبات‌ها انتقال می‌دهد و کاربر کنترلی روی این فرآیند ندارد. بسیار واضح است که تعداد حافظه‌ی ثبات در پردازنده‌ی گرافیکی اهمیت بیشتری نسبت به پردازنده‌های معمولی دارد. نبود یا کمبود این حافظه سبب مراجعات بیشتر به حافظه سراسری می‌شود و این خود باعث کاهش راندمان و سرعت پردازش است. در جدول ۲-۳ تعداد ثبات‌هایی که یک نخ می‌تواند استفاده کند و تعداد ثباتی که یک چندپردازنده می‌تواند استفاده کند در پردازنده‌های نسل‌های مختلف نشان داده شده است.

جدول ۲-۳: پیشرفت و تغییرات ثبات‌ها در طی زمان و در تراشه‌های گرافیکی از نسل‌های مختلف

GK110	GK104	GF1xx	GT200	G8x-G9x	
255	63	63	128	128	تعداد ثبات برای هر نخ
64000	64000	32000	16000	8000	تعداد ثبات در هر چندپردازنده

۲-۹-۴-۴ حافظه‌ی سراسری

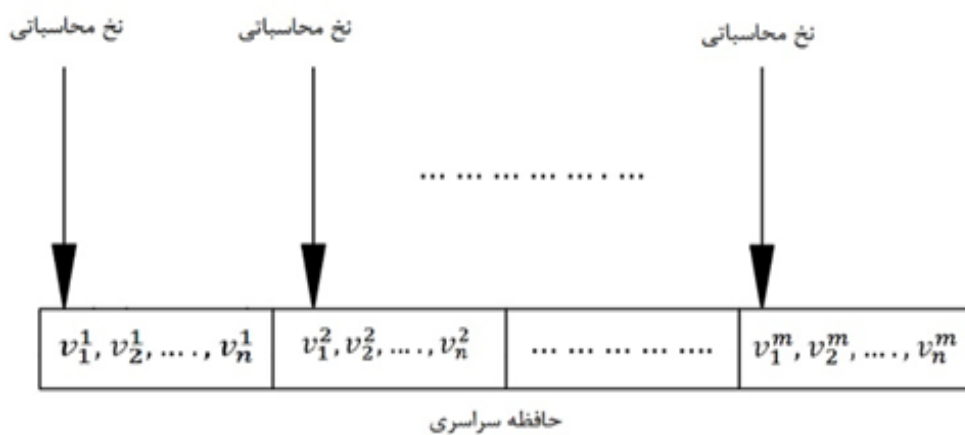
بزرگ‌ترین مقدار حافظه از نظر ظرفیت، حافظه‌ی سراسری است. همچنین این حافظه پهنای باند وسیعی دارد و مقادیر موضعی و سراسری روی این حافظه ذخیره‌سازی می‌شوند. تمام هسته‌ها، چندپردازنده‌ها و طبیعتاً تمام نخ‌های محاسباتی به این حافظه دسترسی دارند و اجازه‌ی خواندن و نوشتن روی این حافظه را نیز دارا هستند. یکی از نکاتی که در دسترسی نخ‌ها به حافظه‌ی سراسری

¹ Registers

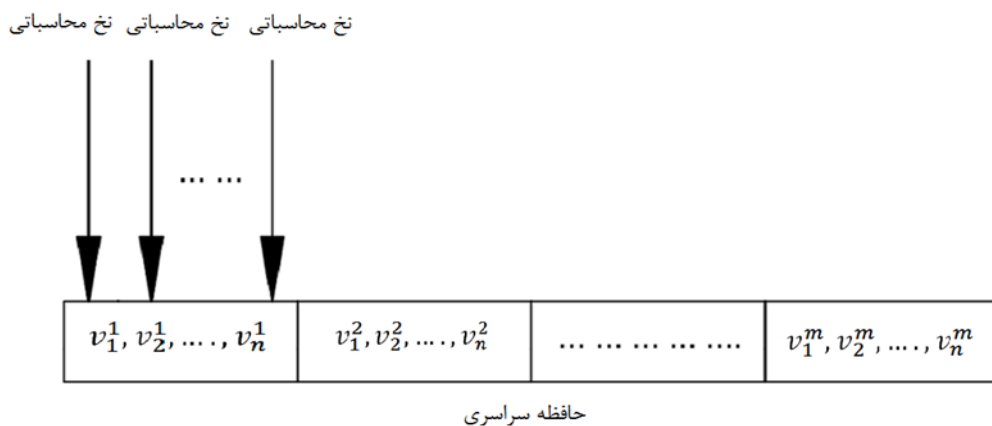
باید مدنظر داشت، این مسأله است که نخ‌های موازی در هر لحظه به یک منطقه از مکان حافظه دسترسی داشته باشند. به علت بالا بودن تعداد نخ‌ها و امکان دسترسی همه نخ‌ها به حافظه‌ی سراسری، اگر هر نخ بخواهد به قسمت‌های مختلف حافظه از نظر آدرس و محل قرارگیری، دسترسی داشته باشد، با توجه به محدود بودن پهنای باند، سرعت پردازش کاهش پیدا می‌کند. به این نوع دسترسی به حافظه‌ی سراسری دسترسی غیر هم‌مکان^۱ می‌گویند (شکل ۲-۱۱). اما روش مناسب برای استفاده از حافظه‌ی سراسری دسترسی هم‌مکان^۲ است یعنی در هر لحظه تمام نخ‌ها به محدوده مشخصی از حافظه که در کنار هم قرار دارند رجوع کنند و مقدار موردنظر را فراخوانی یا ذخیره کنند (شکل ۲-۱۲). در شکل ۲-۱۱ و شکل ۲-۱۲ به‌عنوان نمونه یک آرایه‌ی دوبعدی (برای مثال یک ماتریس که به‌صورت خطی ذخیره‌شده است) در حافظه‌ی سراسری نمایش داده‌شده و نوع دسترسی نخ‌ها نیز مشخص هستند. در حالت دسترسی غیر هم‌مکان هر نخ به مقادیر ابتدای هر سطر ماتریس دسترسی دارند. اما در دسترسی هم‌مکان نخ‌ها به مقادیر سطر اول رجوع می‌کنند و این مقادیر در کنار یکدیگر قرار دارند، به همین دلیل با فراخوانی قسمتی از حافظه نیاز کل نخ‌ها برطرف می‌شود. دسترسی هم‌مکان بهترین حالت برای بالا نگه‌داشتن سرعت پردازش است که در ادامه تأثیر این مسأله روی سرعت بررسی‌شده است. از دیگر ویژگی‌های حافظه‌ی سراسری امکان ارتباط این حافظه با پردازنده‌ی مرکزی و حافظه‌ی آن است. حافظه‌های مشترک، ثبات و نهان امکان ارتباط مستقیم با میزبان را ندارند و تنها حافظه‌ای که چنین ویژگی دارد همین حافظه‌ی سراسری است. پس در محاسبات اگر نیاز بود تا داده‌ای برای پردازش موازی از میزبان به دستگاه برای انجام پردازش موازی منتقل شود، این اطلاعات روی حافظه‌ی سراسری ذخیره خواهند شد و در صورت نیاز از آنجا به دیگر حافظه‌ها منتقل شوند.

^۱ UnCoalesced Memory Access

^۲ Coalesced Memory Access

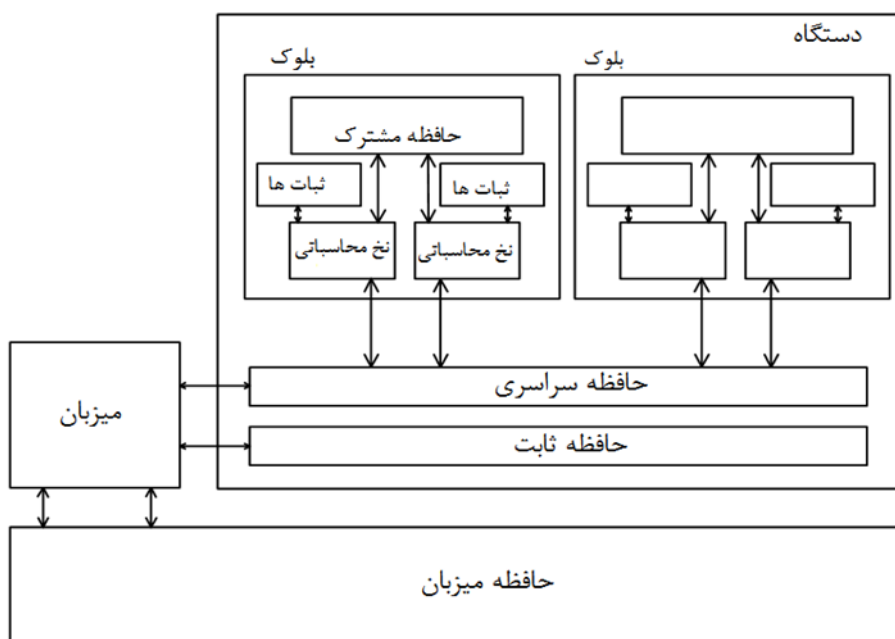


شکل ۱۱-۲: دسترسی غیر هم‌مکان نخ‌های محاسباتی به حافظه‌ی سراسری



شکل ۱۲-۲: دسترسی مناسب و هم‌مکان نخ‌های محاسباتی به حافظه‌ی سراسری

در انتها در شکل ۲-۱۳ یک شمای کلی از نوع برقراری ارتباط بین نخ‌های محاسباتی و انواع حافظه نمایش داده شده است. همچنین رابطه‌ی بین میزبان و پردازنده‌ی گرافیکی یا همان دستگاه نیز نشان داده شده‌اند.



شکل ۲-۱۳: رابطه‌ی بین انواع حافظه و نخ‌های محاسباتی و ارتباط کلی بین میزبان و پردازنده‌ی مرکزی

۲-۹-۵ ملاحظات نرم‌افزاری و برنامه‌نویسی روی پردازنده‌ی

گرافیکی

تاکنون در مورد هسته‌های پردازشی و انواع حافظه در پردازنده‌های گرافیکی توضیحاتی داده شده است؛ اما در پردازش موازی موضوعات دیگری نیز تأثیرگذار هستند و آن انتخاب و سازمان‌دهی کمیت‌های نرم‌افزاری هستند.

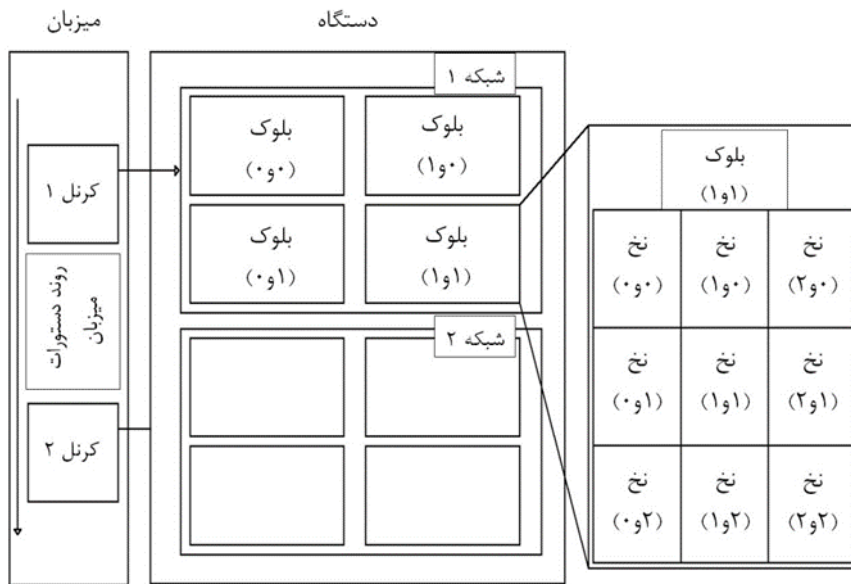
گفته شد که کوچک‌ترین واحد نرم‌افزاری، یک سری وظایف و دستوراتی هستند که به یک هسته سپرده می‌شوند تا آن‌ها را انجام دهد و به آن‌ها نخ یا رشته می‌گویند. این نخ‌ها در دسته‌هایی تقسیم‌بندی می‌شوند به نام بلوک^۱ که می‌توانند یک، دو یا سه‌بعدی باشند. بلوک‌ها نیز در دسته‌های بزرگ‌تری دسته‌بندی می‌شوند بانام شبکه که بزرگ‌ترین واحد تقسیم‌بندی نخ‌ها به حساب می‌آیند (شکل ۲-۱۴) روند کلی برنامه‌نویسی با پردازنده‌های گرافیکی به این صورت است که برنامه ابتدا روی

¹ Block

میزبان شروع به کار می‌کند، سپس در هر قسمتی از برنامه که الگوریتم قابلیت اجرای هم‌زمان یا موازی را داشت، پردازنده‌ی مرکزی با فراخوانی پردازنده‌ی گرافیکی و انتقال اطلاعات به آن این قسمت از پردازش را به عهده‌ی هسته‌های پرتعداد و قدرتمند پردازنده گرافیکی می‌گذارد.

فراخوانی پردازنده گرافیکی توسط یک کرنل^۱ صورت می‌گیرد. کرنل‌ها همانند توابع در زبان سی هستند که با فراخوانی آن‌ها پردازنده گرافیکی دستورات داده‌شده در کرنل را انجام و در صورت نیاز اطلاعات را به میزبان برمی‌گرداند، سپس برنامه روند عادی خود را طی می‌کند و پردازنده‌ی گرافیکی از چرخه‌ی پردازش خارج می‌شود تا دوباره در صورت لزوم توسط یک کرنل دیگر فراخوانی شود. فراخوانی با کرنل‌ها علاوه بر اینکه نیاز به متغیرهای اعلان‌شده‌ای در ورودی خود دارد اطلاعات مربوط به ابعاد بلوک‌ها و شبکه و تعداد بعد آن‌ها باید برای کرنل و پردازنده‌ی گرافیکی باید مشخص باشد. با توجه به اطلاعات ورودی در مورد ابعاد و تعداد نخ‌ها و بلوک‌ها در کرنل، پردازنده‌ی گرافیکی با دسته‌بندی هسته‌ها شروع به پردازش می‌کند. در شکل ۲-۱۴ مشخص است که هنگام فراخوانی کرنل نخ‌ها در بلوک‌ها و شبکه طبقه‌بندی شده‌اند؛ اما در مسائل مختلف انتخاب و با توجه به نوع معماری پردازنده‌ی گرافیکی استفاده‌شده، تعداد نخ‌ها و بلوک‌ها و ابعاد آن‌ها با محدودیت‌هایی روبرو است. برای مثال در معماری فرمی تعداد نخ‌ها در یک بلوک به صورت $1024 \times 1024 \times 64$ نمایش داده می‌شود یعنی در دو بعد حداکثر تعداد نخ‌ها ۱۰۲۴ عدد است و در بعد سوم حداکثر ۶۴ نخ در یک بلوک قابل استفاده خواهد بود و در مورد تعداد بلوک‌ها در یک شبکه، در هر بعد شبکه تعداد 65535 عدد بلوک قابل دسترسی و استفاده هستند.

¹ Kernel



شکل ۲-۱۴: ساختار اجرای برنامه با کمک پردازنده‌ی گرافیکی و تقسیم‌بندی نخ‌ها در هنگام فراخوانی کرنل‌ها

نمایش و مقداردهی به تعداد نخ‌ها و بلوک‌ها در فراخوانی یک کرنل به صورت زیر است:

$kernel \lll blocks\ per\ grid , threads\ per\ blok \ggg (input\ variable1 , \dots)$

تفاوت فراخوانی توابع در کودا و زبان سی فقط در علامت « \lll », « \ggg » است. عدد اول تعداد بلوک در یک شبکه و عدد دوم تعداد نخ در یک بلوک را نمایش می‌دهد. همان‌طور که ذکر شد انتخاب تعداد نخ در بلوک و تعداد بلوک‌ها می‌تواند نقش مؤثری در بازدهی پردازش ایجاد کند. وقتی به یک چندپردازنده یک یا چند بلوک برای پردازش سپرده می‌شود، چندپردازنده آن‌ها را به ریسمان تبدیل می‌کند با توجه به اینکه تعداد نخ در ریسمان ۳۲ است بهتر است تعداد نخ در هر بلوک ضربی از ۳۲ باشد برای مثال ۱۲۸ یا ۲۵۶. اما تعداد بلوک‌ها در هر شبکه می‌تواند مقادیر مختلفی داشته باشد. در نهایت خلاصه‌ای از محدودیت نخ‌ها و بلوک‌ها در پردازنده‌ها باقابلیت‌های پردازشی متفاوت در جدول ۲-۴ نمایش داده شده‌اند.

جدول ۲-۴: محدودیت پردازنده‌ها با قابلیت محاسباتی مختلف از نظر تعداد نخ و بلوک

قابلیت محاسباتی				
3-3.5	2.x	1.2-1.3	1-1.1	
2	2	2	2	حداکثر ابعاد بلوک‌ها
$2^{31} - 1$	2^{16}	2^{16}	2^{16}	حداکثر اندازه در راستای x
2^{16}	2^{16}	2^{16}	2^{16}	حداکثر اندازه در راستای z,y
3	3	3	3	حداکثر ابعاد نخ‌ها
64	64	64	64	حداکثر ابعاد نخ‌ها در راستای z
32	32	32	32	ابعاد ریسمان
1024	1024	512	512	حداکثر تعداد نخ‌ها در بلوک
2048	1536	1024	768	حداکثر تعداد نخ‌ها در چندپردازنده
16	16	8	8	حداکثر تعداد بلوک‌ها در چندپردازنده

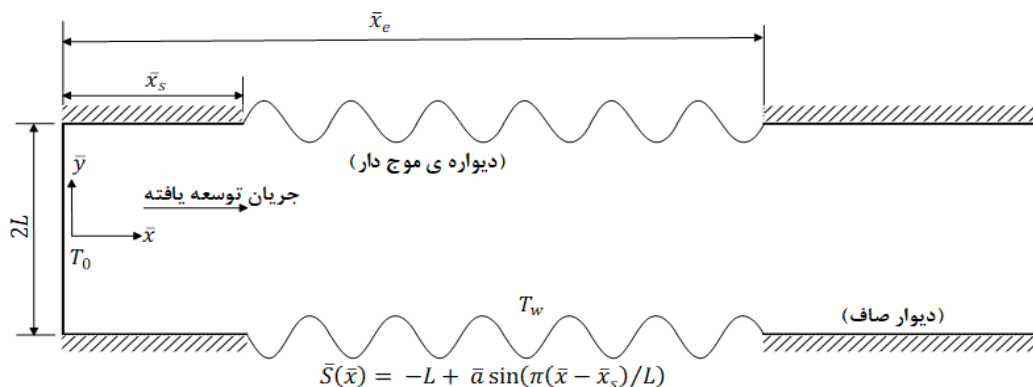
فصل ۳ : معرفی مسأله، روابط حاکم، الگوریتم حل
مسأله، گسسته‌سازی عددی

۱-۳ تعریف هندسه‌ی مسأله

مطابق (شکل ۱-۳) هندسه‌ی مسأله از کانال دوبعدی با دیواره‌های موج‌دار که با نانوسیال تراکم‌ناپذیر پر شده است تشکیل می‌شود. هندسه‌ی کانال شامل دو بخش صاف و یک بخش موج‌دار سینوسی با دامنه‌ی \bar{a} و ارتفاع $2\bar{L}$ است. طول قسمت صاف قبل از قسمت موج‌دار \bar{x}_S و قسمت موج‌دار شامل شش موج سینوسی با طول $\bar{x}_e - \bar{x}_S$ فرض شده است. معادله‌ی قسمت موج‌دار دیواره‌ی پایینی کانال با رابطه‌ی (۱-۳) بیان می‌گردد:

$$\bar{s}(\bar{x}) = -\bar{L} - \bar{a} \sin \left[\frac{\pi(\bar{x} - \bar{x}_S)}{\bar{L}} \right] \quad \bar{x}_S \leq \bar{x} \leq \bar{x}_e \quad (1-3)$$

در این مسأله به دیواره‌های بخش موج‌دار کانال به صورت هم‌دما حرارت داده می‌شود و دیواره‌های بخش صاف کانال عایق می‌باشند.



شکل ۱-۳: شماتیک مدل فیزیکی

۲-۳ فرضیات مسأله و معادلات حاکم

در مطالعه‌ی حاضر، جریان دوبعدی، نیوتونی با خواص ثابت، آرام، پایا و تراکم‌ناپذیر در نظر گرفته شده است. معادله‌های پایداری حاکم بر حرکت سیال در شکل دیفرانسیلی از قوانین زیر استخراج می‌شوند:

۱. بقای جرم (پیوستگی)

۲. بقای ممنتوم خطی (قانون دوم نیوتن)

۳. بقای انرژی (قانون اول ترمودینامیک)

دستگاه معادلات به دست آمده را معادلات ناویر استوکس می نامند. در ادامه به بررسی معادلات ناویر استوکس برای یک کانال موج دار دوبعدی که با نانوسیال تراکم ناپذیر پر شده است، پرداخته می شود.

۳-۲-۱ معادله پیوستگی

طبق تعریف، معادله پیوستگی در مختصات دکارتی و به شکل غیر بقایی به صورت زیر نوشته می شود:

$$\frac{\partial \bar{u}}{\partial \bar{x}} + \frac{\partial \bar{v}}{\partial \bar{y}} = 0 \quad (2-3)$$

۳-۲-۲ معادلات حرکت

طبق تعریف، معادلات حرکت در مختصات دکارتی و به شکل غیر بقایی به صورت زیر نوشته می شود:

$$\bar{u} \frac{\partial \bar{u}}{\partial \bar{x}} + \bar{v} \frac{\partial \bar{u}}{\partial \bar{y}} + \frac{1}{\rho_{nf}} \frac{\partial \bar{p}}{\partial \bar{x}} = \nu_{nf} \left(\frac{\partial^2 \bar{u}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{u}}{\partial \bar{y}^2} \right) \quad (3-3)$$

$$\bar{u} \frac{\partial \bar{v}}{\partial \bar{x}} + \bar{v} \frac{\partial \bar{v}}{\partial \bar{y}} + \frac{1}{\rho_{nf}} \frac{\partial \bar{p}}{\partial \bar{y}} = \nu_{nf} \left(\frac{\partial^2 \bar{v}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{v}}{\partial \bar{y}^2} \right) \quad (4-3)$$

در روابط بالا زیرنویس nf بیانگر خواص نانوسیال، (\bar{u}, \bar{v}) مؤلفه های سرعت، (\bar{x}, \bar{y}) مختصات کارتزین، \bar{p} فشار، ρ چگالی و ν ویسکوزیته سینماتیکی است. سروکار داشتن با معادلات (۳-۳) و (۴-۳) به دلیل وابسته بودن به یکدیگر دشواری های مخصوص به خود را دارد. ولی در بعضی مسائل نظیر مدل سازی جریان آشفته، مطالعه ای جابه جایی آزاد یا مختلط، جریان های تراکم ناپذیر دوبعدی یا تقارن محوری می توان از تعاریف و ابزارهای قدرتمندی مثل تابع جریان و تاوایی برای ساده سازی معادلات ناویر استوکس استفاده کرد. در بخش بعدی معادلات تابع جریان و تاوایی ارائه شده است.

۳-۲-۳ معادلات تابع جریان و تاوایی

در حالت کلی تابع جریان $(\bar{\psi})$ مطابق معادله‌ی (۵-۳) و تاوایی $(\bar{\omega})$ مطابق رابطه‌ی (۶-۳) تعریف می‌شوند.

$$\bar{u} = \frac{\partial \bar{\psi}}{\partial \bar{y}} \quad \text{و} \quad \bar{v} = -\frac{\partial \bar{\psi}}{\partial \bar{x}} \quad (۵-۳)$$

$$\bar{\omega} = \frac{\partial \bar{v}}{\partial \bar{x}} - \frac{\partial \bar{u}}{\partial \bar{y}} \quad (۶-۳)$$

اگر از معادله‌ی (۳-۳) نسبت به \bar{y} و از معادله‌ی (۴-۳) نسبت به \bar{x} مشتق گرفته شود (معادلات (۷-۳) و (۸-۳)) و این دو معادله از هم کم شوند، بعد از یک سری ساده‌سازی، معادله‌ی (۹-۳) بانام معادله‌ی پخشی-انتقالی تاوایی به دست می‌آید.

$$\frac{\partial}{\partial \bar{y}} \left(\frac{\partial \bar{u}}{\partial t} + \bar{u} \frac{\partial \bar{u}}{\partial \bar{x}} + \bar{v} \frac{\partial \bar{u}}{\partial \bar{y}} + \frac{1}{\rho_{nf}} \frac{\partial \bar{p}}{\partial \bar{x}} \right) = \nu_{nf} \left(\frac{\partial^2 \bar{u}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{u}}{\partial \bar{y}^2} \right) \quad (۷-۳)$$

$$\frac{\partial}{\partial \bar{x}} \left(\frac{\partial \bar{v}}{\partial t} + \bar{u} \frac{\partial \bar{v}}{\partial \bar{x}} + \bar{v} \frac{\partial \bar{v}}{\partial \bar{y}} + \frac{1}{\rho_{nf}} \frac{\partial \bar{p}}{\partial \bar{y}} \right) = \nu_{nf} \left(\frac{\partial^2 \bar{v}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{v}}{\partial \bar{y}^2} \right) \quad (۸-۳)$$

$$\bar{u} \frac{\partial \bar{\omega}}{\partial \bar{x}} + \bar{v} \frac{\partial \bar{\omega}}{\partial \bar{y}} = \nu_{nf} \left(\frac{\partial^2 \bar{\omega}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{\omega}}{\partial \bar{y}^2} \right) \quad (۹-۳)$$

همچنین با جایگذاری رابطه‌ی (۷-۳) در معادله‌ی (۸-۳)، رابطه‌ی بین تابع جریان و تاوایی به صورت رابطه‌ی (۱۰-۳) به دست می‌آید.

$$\frac{\partial^2 \bar{\psi}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{\psi}}{\partial \bar{y}^2} = -\bar{\omega} \quad (۱۰-۳)$$

معادلات (۹-۳) و (۱۰-۳)، معادلات اساسی روش تابع جریان-تاوایی هستند. این روش حل برای جریان سیالات تراکم‌ناپذیر کاربرد دارد و ویژگی آن این است که دو معادله برای تحلیل جریان در اختیار می‌گذارد و تعداد معادلات را از سه به دو عدد کاهش می‌دهد؛ زیرا معادله‌ی پیوستگی خودبه‌خود توسط تعریف تابع جریان ارضا می‌شود. نکته‌ی مهم دیگر از بین رفتن عبارت فشار در معادلات است که سهولت بیشتری در انجام محاسبات ایجاد می‌کند.

۳-۲-۴ معادله‌ی انرژی

شکل کلی معادله‌ی انرژی در مختصات دکارتی و با صرف نظر کردن از عبارات تلفات ویسکوز به شکل غیر بقایی به صورت رابطه‌ی (۱۱-۳) نوشته می‌شود.

$$(\rho c_p)_{nf} \left(\bar{u} \frac{\partial \bar{T}}{\partial \bar{x}} + \bar{v} \frac{\partial \bar{T}}{\partial \bar{y}} \right) = k_{nf} \left(\frac{\partial^2 \bar{T}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{T}}{\partial \bar{y}^2} \right) \quad (11-3)$$

در رابطه‌ی (۱۱-۳) کمیت‌های ρ ، c_p و k به ترتیب بیانگر چگالی، ظرفیت گرمایی ویژه، دما و ضریب هدایت گرمایی می‌باشند. برای ساده‌تر شدن معادلات جریان، تاوایی و انرژی تعریف کمیت‌های بدون بعد و همچنین فرضیات ساده کننده ضروری است. برای بی‌بعد سازی از کمیت‌های بدون بعدی که در ادامه آورده می‌شوند استفاده می‌شود.

$$x = \frac{\bar{x}}{\bar{L}}, y = \frac{\bar{y}}{\bar{L}}, u = \frac{\bar{u}}{U_m}, v = \frac{\bar{v}}{U_m}, a = \frac{\bar{a}}{\bar{L}}, x_s = \frac{\bar{x}_s}{\bar{L}}, x_e = \frac{\bar{x}_e}{\bar{L}}, s(x) = \frac{\bar{s}(x)}{\bar{L}}$$

$$\psi = \frac{\bar{\psi}}{\bar{L}U_m}, \quad \omega = \frac{\bar{\omega}\bar{L}}{U_m}, \quad \tau = \frac{tU_m}{\bar{L}}$$

$$\theta = \frac{\bar{T} - \bar{T}_{in}}{\bar{T}_w - \bar{T}_{in}}, \quad \text{Re} = \frac{\rho_f U_m \bar{L}}{\mu_f}, \quad \text{Pr} = \frac{\mu_f c_{pf}}{k_f} \quad (12-3)$$

در روابط فوق کمیت U_m سرعت متوسط، Re عدد رینولدز، Pr عدد پرانتل، τ زمان بدون بعد، a دامنه‌ی موج دیواره‌های کانال و c_{pf} ، μ_f ، k_f ، ρ_f و ψ و ω به ترتیب بیانگر ظرفیت گرمایی، لزجت دینامیکی، ضریب هدایت گرمایی، چگالی سیال پایه، تابع جریان و تاوایی بدون بعد است. معادلات (۹-۳)، (۱۰-۳) و (۱۱-۳) پس از بی‌بعد سازی به صورت روابط (۱۳-۳) الی (۱۵-۳) نوشته می‌شوند:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \quad (13-3)$$

$$u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \frac{\mu_{nf}}{\rho_{nf}} \frac{1}{\text{Re}} \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) \quad (14-3)$$

$$u \frac{\partial \theta}{\partial x} + v \frac{\partial \theta}{\partial y} = \frac{\alpha_{nf}}{\alpha_f} \frac{1}{\text{Re}} \frac{1}{\text{Pr}} \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} \right) \quad (15-3)$$

در این روابط کمیت μ_{nf} بیانگر لزجت دینامیکی نانوسیال، کمیت ρ_{nf} بیانگر چگالی نانوسیال و کمیت‌های α_f و α_{nf} به ترتیب بیانگر ضریب نفوذ گرمایی سیال و نانوسیال است.

۳-۲-۵ خواص نانوسیال

تقریب‌های زیادی برای مدل‌سازی جریان دوفازی جامد-مایع وجود دارد. در این مسأله از مدلی استفاده می‌شود که در آن تأثیر قطر و نوع نانوذره مشخص باشد. معادله‌های مدل استفاده‌شده به صورت زیر نوشته می‌شود [۵۶]:

$$\rho_{nf} = (1 - \varphi)\rho_f + \varphi\rho_{np} \quad (۱۶-۳)$$

$$\frac{\text{Re}_{nf}}{\text{Re}_f} = \frac{v_f}{v_{nf}} = (1 - \varphi)^{2.5} \left(1 - \varphi + \varphi \frac{\rho_p}{\rho_f} \right) \quad (۱۷-۳)$$

$$\frac{k_{nf}}{k_f} = \frac{k_p + 2k_f - 2\varphi(k_f - k_p)}{k_p + 2k_f + 2\varphi(k_f - k_p)} + \frac{k_{Br}}{k_f} \quad (۱۸-۳)$$

$$\frac{\text{Pr}_{nf}}{\text{Pr}_f} = \left[\frac{1 - \varphi + \varphi \frac{(\rho c)_p}{(\rho c)_f}}{(1 - \varphi)^{2.5} \left(1 - \varphi + \varphi \frac{\rho_p}{\rho_f} \right)} \right] k_{nf} \quad (۱۹-۳)$$

در روابط بالا φ کسر حجمی نانوسیال، زیرنویس p و f به ترتیب مربوط به خواص نانوذره و سیال پایه است. k_{Br} ضریب حرکت براونین ذرات است و از معادله‌ی (۳-۲۰) محاسبه می‌شود [۵۷].

$$k_{Br} = 5 \times 10^4 \beta \varphi (\rho c)_f \sqrt{\frac{1.3807 \times 10^{-23} \bar{T}}{\rho_p d_p}} \quad (۲۰-۳)$$

در رابطه‌ی (۳-۲۰)، d_p قطر نانوذره، \bar{T} دمای متوسط سیال و β کمیتی متفاوت با نوع سیال است. کمیت β و خواص ترموفیزیکی برای نانوذره‌ی مس و آلومینیوم در جدول ۳-۱ نشان داده شده است.

معادله‌های (۳-۱۴) و (۳-۱۵) با در نظر گرفتن خواص نانوسیال (معادله‌های (۳-۱۶) الی (۳-۱۹))

به شکل زیر بازنویسی می‌شوند:

$$u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \frac{1}{\text{Re}} \frac{\rho_f}{\rho_{nf}} \frac{1}{(1 - \varphi)^{2.5}} \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) \quad (۲۱-۳)$$

$$u \frac{\partial \theta}{\partial x} + v \frac{\partial \theta}{\partial y} = \frac{1}{\text{RePr}} \frac{k_{nf}}{k_f} \frac{(\rho c_p)_f}{(\rho c_p)_{nf}} \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} \right) \quad (۲۲-۳)$$

جدول ۳-۱: خواص ترموفیزیکی سیال و نانو ذره

خواص ترموفیزیکی	سیال پایه (آب)	مس	آلومینیوم اکسید
ظرفیت گرمایی	4179	535.6	756
چگالی	997.1	6530	3970
ضریب هدایت گرمایی	0.631	69	46
ویسکوزیته سینماتیکی	1.005×10^{-6}	-	-
β (برای کسر حجمی نانوسیال بیشتر از ۱٪)	-	$0.0011(100\phi)^{-0.7272}$	$0.0017(100\phi)^{-0.0841}$

۳-۳ تبدیل مختصات معادلات دیفرانسیل حاکم

هندسه‌ی فیزیکی مسأله‌ی حاضر غیر مستطیلی و با مرزهای نامنظم است، بنابراین اعمال قلمرو محاسبات مستطیلی شکل بر چنین قلمروهای فیزیکی به‌نوعی میانینانی برای اعمال شرایط مرزی به‌حساب می‌آید. در شبکه‌های غیریکنواخت، در نزدیکی مرزها، پیچیدگی‌های بیشتری در معادلات به وجود می‌آید، زیرا از تقریب‌های با فواصل نامساوی استفاده شده است. برای غلبه بر این مشکل، فضای فیزیکی به فضای محاسباتی انتقال داده می‌شود. این انتقال با معرفی دستگاه مختصات کلی انجام می‌شود و شبکه‌ی غیر مستطیلی موجود در فضای فیزیکی را به شبکه‌ی یکنواخت مستطیلی در فضای محاسباتی تصویر می‌کند [۵۸]. برای ایجاد شبکه‌سازی روش‌های مختلفی وجود دارد که عبارت‌اند از: ۱- شبکه‌سازی به روش جبری ۲- شبکه‌سازی با استفاده از معادلات دیفرانسیل ۳- شبکه‌سازی با استفاده از اعداد مختلط و نگاشت هم‌مدیس^۱. در این پایان‌نامه روش‌های شبکه‌سازی به کمک روابط جبری و معادلات دیفرانسیل ارائه می‌شود.

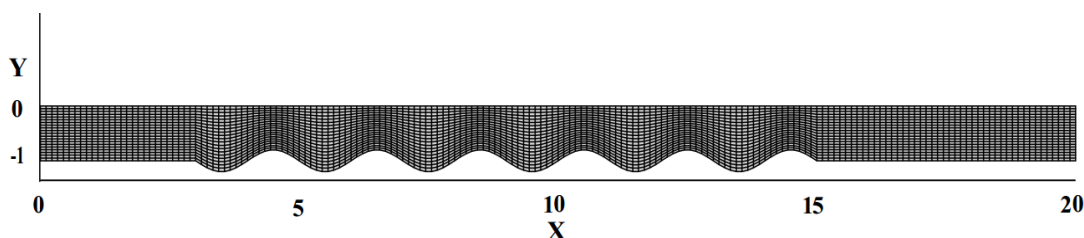
^۱ Conformal Mapping

۳-۱-۳ روش جبری شبکه‌سازی

ساده‌ترین شیوهی تولید شبکه، روش جبری است. بزرگ‌ترین حسن این روش سرعت بالای شبکه‌سازی است. در این روش از یک معادله‌ی جبری برای ایجاد ارتباط بین نقاط شبکه در قلمرو محاسباتی و نقاط نظیر در قلمرو فیزیکی استفاده می‌شود. این هدف با به‌کارگیری یک روش میانجی برای ایجاد نقاط داخلی شبکه از نقاط مرزی معلوم حاصل می‌شود. در این روش با داشتن معادلات مرزی هندسه‌ی موردنظر، می‌توان رابطه‌ای تحلیلی بین مختصات فضای محاسباتی و فیزیکی به دست آورد. بدیهی است که معادلات جبری مختلفی را می‌توان برای این منظور به کاربرد. با معرفی روابط جبری زیر، قلمرو غیر مستطیلی فیزیکی به یک قلمرو مستطیلی تبدیل می‌شود [۵۸].

$$\xi = x, \quad \eta = \frac{y}{s(x)} \quad (3-23)$$

قلمرو فیزیکی مسأله مطابق شکل ۳-۲ در نظر گرفته می‌شود. در فضای فیزیکی و محاسباتی تعداد نقاط شبکه با IM (تعداد ماکزیمم نقاط در راستای x و ξ) و JM (تعداد ماکزیمم نقاط در راستای y و η) مشخص می‌شود.



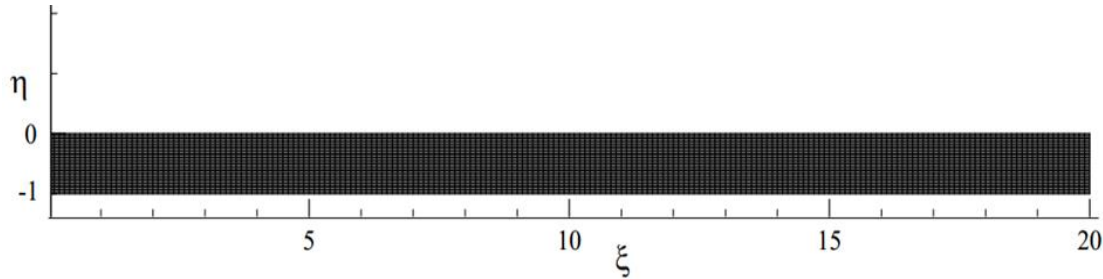
شکل ۳-۲: فضای فیزیکی مسأله

در فضای محاسباتی فواصل بین نقاط مساوی و به‌صورت روابط (۳-۲۴) و (۳-۲۵) تعریف می‌شود.

$$\Delta\xi = \frac{L}{IM - 1} \quad (3-24)$$

$$\Delta\eta = \frac{L}{JM - 1} \quad (3-25)$$

فواصل هندسی با طول مساوی که در معادلات (۳-۲۴) و (۳-۲۵) تعریف شده، فضای محاسباتی یکنواختی را می‌سازد. در شکل ۳-۳ فضای محاسباتی نشان داده شده است.



شکل ۳-۳: فضای محاسباتی

بنابراین از سیستم مختصات ثابت شده^۱ استفاده شده و معادلات حاکم که بر مسأله به مختصات ξ و η انتقال می‌یابند:

$$\nabla^2 \psi = -\omega \quad (۲۶-۳)$$

$$u \left(\frac{\partial \omega}{\partial \xi} + \eta_x \frac{\partial \omega}{\partial \eta} \right) + v \eta_y \frac{\partial \omega}{\partial \eta} = \frac{1}{\text{RePr}} \frac{\rho_f}{\rho_{nf}} \frac{1}{(1-\varphi)^{2.5}} \nabla^2 \omega \quad (۲۷-۳)$$

$$u \left(\frac{\partial \theta}{\partial \xi} + \eta_x \frac{\partial \theta}{\partial \eta} \right) + v \eta_y \frac{\partial \theta}{\partial \eta} = \frac{1}{\text{RePr}} \frac{k_{nf}}{k_f} \frac{(\rho c_p)_f}{(\rho c_p)_{nf}} \nabla^2 \theta \quad (۲۸-۳)$$

در روابط بالا کمیت‌های سرعت، تاوایی و مشتق‌های انتقال یافته از رابطه‌های زیر به دست می‌آیند:

$$u = \eta_y \frac{\partial \psi}{\partial \eta}, \quad v = - \left(\frac{\partial \psi}{\partial \xi} + \eta_x \frac{\partial \psi}{\partial \eta} \right) \quad (۲۹-۳)$$

$$\omega = \frac{\partial v}{\partial \eta} + \eta_x \frac{\partial v}{\partial \eta} - \eta_y \frac{\partial u}{\partial \eta} \quad (۳۰-۳)$$

$$\nabla^2 = \frac{\partial^2}{\partial \xi^2} + (\eta_x^2 + \eta_y^2) \frac{\partial^2}{\partial \eta^2} + 2 \eta_x \frac{\partial^2}{\partial \xi \partial \eta} + (\eta_{xx} + \eta_{yy}) \frac{\partial}{\partial \eta} \quad (۳۱-۳)$$

قبل از حل معادله‌های تبدیل شده نخست باید متریک‌ها^۲ را محاسبه کرد. به ضرایب مشتقات نسبت

به ξ و η متریک گفته می‌شود. در بیشتر مدل‌های جبری متریک‌ها را می‌توان به صورت تحلیلی

^۱ Body-fitted

^۲ Metrics

محاسبه کرد. بدیهی است که این یکی از فواید روش جبری است، زیرا محاسبه‌ی عددی متریک‌ها خود نیاز به زمان محاسباتی اضافی دارد و ممکن است در دستگاه معادلات حرکت مورد حل خطا ایجاد کند. متریک‌های موجود در معادلات بالا از رابطه‌های (۳۲-۳) و (۳۳-۳) محاسبه می‌گردد:

$$\eta_y = \frac{1}{s(x)}, \quad \eta_x = \frac{-ys'(x)}{s^2(x)} \quad (۳۲-۳)$$

$$\eta_{yy} = 0, \quad \eta_{xx} = \frac{y(2s'^2(x) - s(x)s''(x))}{s^3(x)} \quad (۳۳-۳)$$

۳-۳-۲ روش شبکه‌سازی با معادلات دیفرانسیل

این روش شبکه‌سازی، مبتنی بر معادله‌ی دیفرانسیل پاره‌ای است. در این روش یک دستگاه معادله‌ی دیفرانسیل پاره‌ای حل می‌شود تا نقاط شبکه در فضای فیزیکی به دست آید. درحالی‌که فضای محاسباتی یک شبکه‌ی مستطیلی با فواصل یکنواخت است. این روش به سه روش معادلات دیفرانسیل بیضوی، سهموی و هذلولوی دسته‌بندی می‌شود. روش بیضوی تولید شبکه، روشی است که بیشتر توسعه‌یافته و روی آن کار شده است. این روش بیشتر برای مسائل دوبعدی به کار می‌رود. در ادامه معادلات روش بیضوی ارائه خواهد شد. هرگاه مرزهای فیزیکی مشخص باشند، تولید بیضوی شبکه بسیار خوب عمل می‌کند. روابط زیر که بین فضای فیزیکی و فضای محاسباتی تعریف شده است را در نظر بگیرید [۵۸].

$$\xi = \xi(x, y) \quad (۳۴-۳)$$

$$\eta = \eta(x, y) \quad (۳۵-۳)$$

قانون زنجیره‌ای برای مشتق‌های اول پاره‌ای به صورت زیر است:

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial}{\partial \eta} \quad (۳۶-۳)$$

$$\frac{\partial}{\partial y} = \frac{\partial \xi}{\partial y} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial}{\partial \eta} \quad (۳۷-۳)$$

تابع f را در نظر بگیرید. برای به دست آوردن مشتق مرتبه‌ی دوم نسبت به x عملیات ریاضی زیر انجام می‌شود.

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= \frac{\partial}{\partial x} \left[\frac{\partial f}{\partial x} \right] = \frac{\partial}{\partial x} [\xi_x f_\xi + \eta_x f_\eta] = \left[\xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} \right] [\xi_x f_\xi + \eta_x f_\eta] \\ &= \xi_x^2 f_{\xi\xi} + \xi_x f_\xi \frac{\partial}{\partial \xi} [\xi_x] + \xi_x \eta_x f_{\xi\eta} + \xi_x f_\eta \frac{\partial}{\partial \xi} [\eta_x] \\ &\quad + \xi_x \eta_x f_{\xi\eta} + \eta_x f_\xi \frac{\partial}{\partial \eta} [\xi_x] + \eta_x^2 f_{\eta\eta} + \eta_x f_\eta \frac{\partial}{\partial \eta} [\eta_x] \end{aligned} \quad (38-3)$$

در این بخش کمیتی معرفی می‌شود به نام ژاکوبین تبدیل که عبارت است از: نسبت سطح (نسبت حجم در حالت سه‌بعدی) در فضای فیزیکی به سطح در فضای محاسباتی. ژاکوبین تبدیل از فرمول زیر محاسبه می‌گردد:

$$J = \frac{1}{x_\xi y_\eta - y_\xi x_\eta} \quad (39-3)$$

همچنین می‌توان متریک‌ها را به صورت زیر نوشت:

$$\xi_x = J y_\eta \quad (40-3)$$

$$\xi_y = -J x_\eta \quad (41-3)$$

$$\eta_x = -J y_\xi \quad (42-3)$$

$$\eta_y = J x_\xi \quad (43-3)$$

با توجه به رابطه‌ی (39-3) می‌توان معادله‌ی مشتق مرتبه‌ی دوم را برحسب ژاکوبین تبدیل نوشت.

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= J^2 (y_\eta^2 f_{\eta\eta} - 2y_\xi y_\eta f_{\xi\eta} + y_\xi^2 f_{\xi\xi}) \\ &\quad + J^3 \{ (y_\eta^2 y_{\xi\xi} - 2y_\eta y_\xi y_{\xi\eta} + y_\xi^2 y_{\eta\eta}) (x_\eta f_\xi - x_\xi f_\eta) \\ &\quad + (y_\eta^2 x_{\xi\xi} - 2y_\eta y_\xi x_{\xi\eta} + y_\xi^2 x_{\eta\eta}) (y_\xi f_\eta - y_\eta f_\xi) \} \end{aligned} \quad (44-3)$$

همچنین معادله‌ی مشتق مرتبه‌ی دوم در جهت y به صورت زیر نوشته می‌شود:

$$\begin{aligned} \frac{\partial^2 f}{\partial y^2} &= J^2 (x_\eta^2 f_{\xi\xi} - 2x_\xi x_\eta f_{\xi\eta} + x_\xi^2 f_{\eta\eta}) \\ &\quad + J^3 \{ (x_\eta^2 y_{\xi\xi} - 2x_\eta x_\xi y_{\xi\eta} + x_\xi^2 y_{\eta\eta}) (x_\eta f_\xi - x_\xi f_\eta) \\ &\quad + (x_\eta^2 x_{\xi\xi} - 2x_\eta x_\xi x_{\xi\eta} + x_\xi^2 x_{\eta\eta}) (y_\xi f_\eta - y_\eta f_\xi) \} \end{aligned} \quad (45-3)$$

حال، معادله‌ی لاپلاس را در نظر بگیرید.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (46-3)$$

معادلات (44-3) و (45-3) در معادله (46-3) جایگزین می‌شود. بعد از ساده‌سازی معادله (47-3) به دست می‌آید.

$$\begin{aligned} \nabla^2 f = J^2 & \left((x_\eta^2 + y_\eta^2) f_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta) f_{\xi\eta} + (x_\xi^2 + y_\xi^2) f_{\eta\eta} \right) \\ & + J^3 \{ [(x_\xi^2 + y_\xi^2) y_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta) y_{\xi\eta} \\ & + (x_\eta^2 + y_\eta^2) y_{\eta\eta}] (x_\eta f_\xi - x_\xi f_\eta) \\ & + [(x_\eta^2 + y_\eta^2) x_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta) x_{\xi\eta} \\ & + (x_\xi^2 + y_\xi^2) x_{\eta\eta}] (y_\xi f_\eta - y_\eta f_\xi) \} \end{aligned} \quad (47-3)$$

برای ساده‌سازی معادله (47-3) عبارتهای زیر تعریف می‌شود:

$$a = x_\eta^2 + y_\eta^2 \quad (48-3)$$

$$b = x_\xi x_\eta + y_\xi y_\eta \quad (49-3)$$

$$c = x_\xi^2 + y_\xi^2 \quad (50-3)$$

$$g = ax_{\xi\xi} - 2bx_{\xi\eta} + cx_{\eta\eta} \quad (51-3)$$

$$h = ay_{\xi\xi} - 2by_{\xi\eta} + cy_{\eta\eta} \quad (52-3)$$

$$d = J(gy_\xi - hx_\xi) \quad (53-3)$$

$$e = J(hx_\eta - gy_\eta) \quad (54-3)$$

با جایگذاری عبارتهای بالا در معادله (47-3) معادله (55-3) به دست می‌آید:

$$\nabla^2 f = J^2 (af_{\xi\xi} - 2bf_{\xi\eta} + cf_{\eta\eta} + df_\eta + ef_\xi) \quad (55-3)$$

۳-۲-۱ روش بیضوی

هرگاه مرزهای فیزیکی مشخص باشند، تولید بیضوی شبکه بسیار خوب عمل می‌کند. دستگاه معادلات دیفرانسیل بیضوی به صورت معادله لاپلاس یا معادله پواسون معرفی می‌شود؛ که از حل آن‌ها مختصات نقاط شبکه در فضای فیزیکی به دست می‌آید. دستگاه معادلات بیضوی زیر را در نظر بگیرید.

$$\nabla^2 \xi = 0 \quad (56-3)$$

$$\nabla^2 \eta = 0 \quad (57-3)$$

با توجه به معادلات (55-3)، (56-3) و (57-3) روابط زیر محاسبه می‌گردد:

$$ax_{\xi\xi} - 2bx_{\xi\eta} + cx_{\eta\eta} = 0 \quad (58-3)$$

$$ay_{\xi\xi} - 2by_{\xi\eta} + cy_{\eta\eta} = 0 \quad (59-3)$$

با توجه به معادلات (58-3) و (59-3) متریک‌ها محاسبه می‌شوند. با استفاده از معادلات (13-3) و

(21-3) می‌توان یک جریان دوبعدی را شبیه‌سازی و حل کرد. پس از حل معادلات (13-3) و (21-3)

و به دست آمدن مقادیر تاوایی و تابع جریان روی یک دامنه‌ی محاسباتی، می‌توان به راحتی توسط

روابط (5-3) مؤلفه‌های عمودی و افقی سرعت را به دست آورد.

۳-۴ شرایط مرزی

برای حل معادله‌های تابع جریان و تاوایی باید شرایط مرزی را تعیین کرد. به دلیل تقارن هندسه‌ی

فیزیکی دامنه‌ی محاسباتی حل نصف هندسه در نظر گرفته می‌شود. در مجموع، شرایط مرزی به چهار

دسته تقسیم می‌شود: جریان ورودی، جریان خروجی، شرایط مرزی دیواره و خط تقارن شکل ۳-۴ در

ورودی جریان با توجه به ابعاد کانال و طول توسعه‌یافتگی مؤلفه‌ی افقی سرعت به صورت پروفیل

مرتبه‌ی دو در نظر گرفته می‌شود و با توجه به رابطه‌ی (5-3) مقدار تابع جریان در ورودی محاسبه

می‌گردد. مقدار تاوایی در جریان ورودی از رابطه‌ی (13-3) محاسبه می‌شود. دما در ورودی جریان

ثابت در نظر گرفته می‌شود.

$$\psi = \frac{3}{2} \left(\eta - \frac{\eta^3}{3} \right), \quad \omega = 3\eta, \quad \theta = 0 \quad (60-3)$$

در مرز خروجی، مقدار تابع جریان، تاوایی و دما از برون‌یابی جواب داخل میدان به دست می‌آید.

$$\frac{\partial \psi}{\partial \xi} = 0, \quad \frac{\partial \omega}{\partial \xi} = 0, \quad \frac{\partial \theta}{\partial \xi} = 0 \quad (61-3)$$

دیواره‌ی کانال یک خط جریان است و به همین دلیل مقدار تابع جریان روی آن ثابت است. در

دیواره‌ی کانال شرایط مرزی برای تاوایی وجود ندارد و باید شرایط مرزی را برای آن ساخت. با توجه

به شرط عدم لغزش و معادله‌ی (۳-۱۵) شرایط مرزی برای تاوایی روی دیواره‌ی کانال از رابطه‌ی زیر محاسبه می‌گردد. در قسمت صاف دیواره‌ی کانال شرط مرزی آدیاباتیک و در قسمت موج‌دار کانال شرط مرزی دما ثابت در نظر گرفته می‌شود.

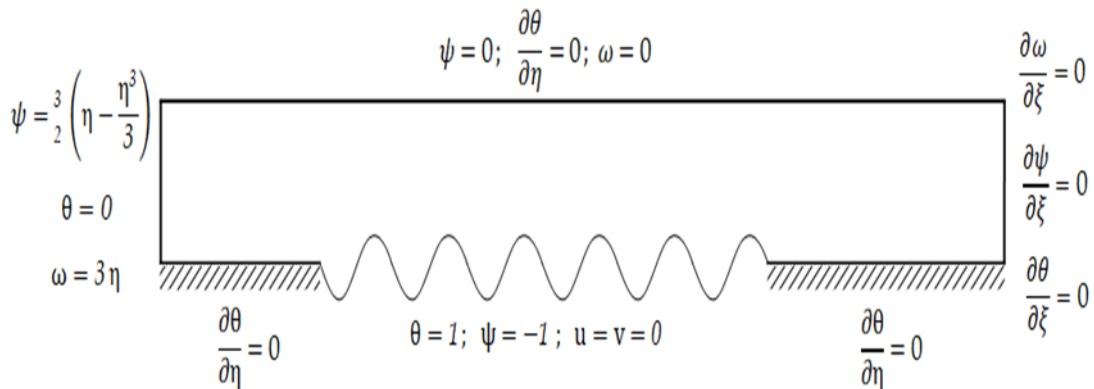
$$\psi = 1, \quad \omega = \frac{\partial^2 \psi}{\partial y^2}$$

$$\theta = 1 \quad \xi_s \leq \xi \leq \xi_e$$

$$\frac{\partial \theta}{\partial \eta} = 0 \quad \xi < \xi_s \text{ یا } \xi \geq \xi_e \quad (۳-۶۲)$$

اگر مرز تقارن منطبق بر خط تقارن در نظر گرفته شود مرز تقارن را می‌توان به‌عنوان خط جریان فرض کرد. بنابراین، مقدار تابع جریان در امتداد این مرز ثابت در نظر گرفته می‌شود. مؤلفه‌های سرعت در امتداد عمود بر مرز تقارن صفرند، پس با توجه به ثابت بودن تابع جریان در مرز تقارن و معادله‌ی (۳-۱۴) تاوایی صفر می‌شود. در مرز تقارن شرط مرزی آدیاباتیک در نظر گرفته می‌شود.

$$\psi = \omega = 0, \quad \frac{\partial \theta}{\partial \xi} = 0 \quad (۳-۶۳)$$



شکل ۳-۴: شرایط مرزی مسأله

در رابطه‌ی (۳-۶۳) کمیت ξ_s مربوط به طول قسمت صاف قبل از شروع موج و کمیت ξ_e مربوط به طول قسمت صاف قبل از موج به‌اضافه‌ی قسمت موج‌دار است. عدد ناسلت محلی و ناسلت متوسط

روی دیواره‌ی پایین از روابط زیر تعریف می‌گردد.

$$Nu_x = \frac{h_x D_h}{k_{nf}} = - \frac{k_{nf}}{k_f} \frac{\partial \theta}{\partial y} \Big|_{y=0} \quad (۶۴-۳)$$

$$Nu_m = \frac{1}{x - x_s} \int_{x_s}^x (1 + s'^2(x))^{0.5} Nu_x dx \quad (۶۵-۳)$$

و همچنین ضریب اصطکاک پوسته‌ای روی دیواره‌ی پایین طبق رابطه‌ی زیر محاسبه می‌گردد.

$$C_{fx} = \frac{\mu_{nf}}{\mu_f} \frac{\rho_f}{\rho_{nf}} \frac{1}{Re} \frac{\partial u}{\partial y} \Big|_{y=0} \quad (۶۶-۳)$$

۳-۵ الگوریتم حل مسأله

برای حل مسأله باید رابطه‌های (۲۶-۳)، (۲۷-۳) و (۲۸-۳) با توجه به شرایط مرزی ارائه‌شده، به صورت هم‌زمان حل شوند. الگوریتم عددی حل بر اساس روش ضمنی اسپلاین با جهت متغیر (SADI) است.

۳-۵-۱ روش ضمنی اسپلاین با جهت متغیر

تعریف: تابع S تعریف‌شده بر بازه‌ی $[a, b]$ یک اسپلاین درجه‌ی k با گره‌های $a = x_0 < x_1 < \dots < x_n = b$ نامیده می‌شود هرگاه:

الف) بر بازه‌ی $[x_i, x_{i+1}]$ که $i = 0, 1, \dots, n - 1$ یک چند جمله‌ای حداکثر از درجه‌ی k باشد.

ب) $S, S', S'', \dots, S^{(k-1)}$ بر بازه‌ی $[a, b]$ پیوسته باشند.

اسپلاین مکعبی: اگر درجه‌ی اسپلاین سه باشد آنگاه اسپلاین را مکعبی می‌گویند. اسپلاین‌های

مکعبی یکی از متداول‌ترین نوع اسپلاین‌ها است [۵۹]. تابع تقریب اسپلاین کاربرد وسیعی در پروژه‌های کاربردی، علوم گرافیکی و گراف دارد.

۳-۵-۲ روابط اصلی اسپلاین

بازه‌ی $[a, b]$ شامل نقاط گره‌ای x_i را در نظر بگیرید. فاصله‌ی بین هر دو نقطه از این بازه h_i نامیده می‌شود [۶۰].

$$h_i = x_i - x_{i-1} > 0 \quad (۶۷-۳)$$

تابع u در نقاط گره‌ای بازه‌ی $[a, b]$ تعریف می‌شود به طوری که:

$$u(x_i) = u_i \quad (۶۸-۳)$$

تابع اسپلاین مکعبی $S_p(x)$ برای دو نقطه‌ی x_i و x_{i-1} با توجه به شرایط زیر تعریف می‌شود:

$$S_p(x_i) = u_i$$

$$S_p(x_{i-1}) = u_{i-1}$$

$$S_p''(x_i) = u_i'' = M_i$$

$$S_p''(x_{i-1}) = u_{i-1}'' = M_{i-1} \quad (۶۹-۳)$$

با توجه به این که $S_p(x)$ یک چند جمله‌ای مکعبی است و قطعه به قطعه لذا $S_p''(x)$ در بازه‌ی $x \in [x_{i-1}, x_i]$ یک تابع خطی است که با توجه به رابطه‌ی (۶۹-۳) به صورت زیر تعریف می‌شود:

$$S_p''(x) = M_{i-1} \left(\frac{x_i - x}{h_i} \right) + M_i \left(\frac{x - x_{i-1}}{h_i} \right) \quad (۷۰-۳)$$

برای محاسبه‌ی تابع اسپلاین از رابطه‌ی (۷۰-۳) دو بار انتگرال گرفته و شرایط معادله‌ی (۶۹-۳) در آن جایگذاری می‌شود:

$$S_p(x) = M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{(x - x_{i-1})^3}{6h_i} + \left(u_{i-1} - \frac{M_{i-1}h_i^2}{6} \right) \frac{x_i - x}{h_i} + \left(u_i - \frac{M_i h_i^2}{6} \right) \frac{x - x_{i-1}}{h_i} \quad (۷۱-۳)$$

و مشتق اول تابع اسپلاین از رابطه‌ی (۷۱-۳) محاسبه می‌شود:

$$S_p'(x) = -M_{i-1} \frac{(x_i - x)^2}{2h_i} + M_i \frac{(x - x_{i-1})^2}{2h_i} + \left(u_{i-1} - \frac{M_{i-1}h_i^2}{6} \right) \frac{-1}{h_i} + \left(u_i - \frac{M_i h_i^2}{6} \right) \frac{1}{h_i} \quad (۷۲-۳)$$

اگر مقدار $S'_p(x_i^-) = m_i^-$ در بازه $[x_{i-1}, x_i]$ و مقدار $S'_p(x_i^+) = m_i^+$ در بازه $[x_i, x_{i+1}]$ در نظر گرفته شود آنگاه طبق خاصیت اسپلین، برای مشتق در هر نقطه‌ی گره‌ای رابطه‌های زیر محاسبه می‌شوند [۶۱]:

$$m_i^- = m_i^+ = m_i \quad (۷۳-۳)$$

$$M_i^- = M_i^+ = M_i \quad (۷۴-۳)$$

$$\frac{1}{h_i} m_{i-1} + 2 \left(\frac{1}{h_i} + \frac{1}{h_{i+1}} \right) m_i + \frac{1}{h_{i+1}} m_{i+1} = \frac{3(u_{i+1} - u_i)}{h_{i+1}^2} + \frac{3(u_i - u_{i-1})}{h_i^2} \quad (۷۵-۳)$$

$$h_i M_{i-1} + 2(h_i + h_{i+1}) M_i + h_{i+1} M_{i+1} = \frac{6(u_{i+1} - u_i)}{h_{i+1}} - \frac{6(u_i - u_{i-1})}{h_i} \quad (۷۶-۳)$$

$$m_{i+1} - m_i = \frac{h_{i+1}}{2} (M_i + M_{i+1}) \quad (۷۷-۳)$$

$$m_i = \frac{h_i}{3} M_i + \frac{h_i}{6} M_{i-1} + \frac{u_i - u_{i-1}}{h_i} \quad (۷۸-۳)$$

$$M_i = \frac{2m_{i-1}}{h_i} + \frac{4m_i}{h_i} - \frac{6(u_i - u_{i-1})}{h_i^2} \quad (۷۹-۳)$$

۳-۵-۳ حل معادله‌های دیفرانسیل پاره‌ای با تقریب اسپلین

در حالت کلی معادله‌ی دیفرانسیل پاره‌ای مرتبه‌ی دوم زیر را در نظر بگیرید:

$$u_t = f(u, u_x, u_{xx}) \quad (۸۰-۳)$$

اگر m_i مشتق اول و M_i مشتق دوم تابع u باشد. در این صورت معادله‌ی (۸۰-۳) را می‌توان به صورت رابطه‌ی (۸۱-۳) بازنویسی کرد:

$$(u_t)_i = f(u, m_i, M_i) \quad (۸۱-۳)$$

مشتق زمانی معادله‌ی (۸۱-۳) طبق روش معمول تفاضل محدود نوشته می‌شود:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = (1 - \theta) f^n + \theta f^{n+1} \quad (۸۲-۳)$$

در معادله‌ی (۸۲-۳) اگر مقدار $\theta = 0$ باشد روش حل صریح و اگر $\theta = 1$ باشد کاملاً ضمنی است. همچنین اگر $\theta = \frac{1}{2}$ باشد روش حل کرانک نیکلسون است. برای حل معادله‌ی (۸۲-۳) باید یک دستگاه ماتریس 3×3 حل شود. برای کاربرد آسان اسپلین مکعبی، دستگاه ماتریس 3×3 به

مجموعه‌ای از معادله‌های اسکالر تشکیل شده از u_i , m_i , M_i کاهش می‌یابد. در نتیجه تابع u را می‌توان به صورت رابطه‌ی (۸۳-۳) نوشت [۶۲ و ۶۳]:

$$u_i^{n+1} = u_i^n + \Delta t(1 - \theta)f^n + \Delta t\theta f^{n+1} = F_i + G_i m_i^{n+1} + S_i M_i^{n+1} \quad (۸۳-۳)$$

در رابطه‌ی (۸۳-۳) مقادیر F_i , G_i و S_i در زمان معلوم هستند. با ترکیب رابطه‌های (۷۵-۳) و (۸۳-۳) رابطه‌ی بین مشتق‌های اول و تابع u بین دونقطه‌ی گره‌ای محاسبه می‌گردد [۶۴]:

$$m_i + a'_i m_{i+1} = b'_i u_i + c'_i u_i + d'_i \quad (۸۴-۳)$$

$$a'_i = \frac{S_i}{S_{i+1}} \left(\frac{h_{i+1} G_{i+1} + 2S_{i+1}}{\Delta'_i} \right) \quad (۸۵-۳)$$

$$b'_i = \frac{h_{i+1}}{\Delta'_i} \quad (۸۶-۳)$$

$$c'_i = \frac{h_{i+1}}{\Delta'_i} \frac{S_i}{S_{i+1}} \quad (۸۷-۳)$$

$$d'_i = -\frac{h_{i+1} S_i}{\Delta'_i} \left(\frac{F_i}{S_i} + \frac{F_{i+1}}{S_{i+1}} \right) \quad (۸۸-۳)$$

$$\Delta'_i = h_{i+1} G_{i+1} - 2S_{i+1} \quad (۸۹-۳)$$

همچنین می‌توان با در نظر گرفتن معادله‌های (۷۷-۳)، (۷۸-۳) و (۸۴-۳) رابطه‌ی بین مشتق اول و تابع u را محاسبه کرد:

$$m_i = a_i u_{i-1} + b_i u_i + c_i \quad (۹۰-۳)$$

$$a_i = \frac{h_i}{\Delta_1} - \frac{1}{\Delta_2 h_i} (3S_{i-1} + h_i^2) \quad (۹۱-۳)$$

$$b_i = \frac{h_i}{\Delta_1} \frac{S_{i-1}}{S_i} + \frac{1}{\Delta_2} \left(\frac{3S_{i-1}}{h_i} - \frac{S_{i-1}}{2S_i} h_i \right) \quad (۹۲-۳)$$

$$c_i = -h_i F_{i-1} \left(\frac{1}{\Delta_1} - \frac{1}{\Delta_2} \right) - h_i F_i \frac{S_{i-1}}{S_i} \left(\frac{1}{\Delta_1} - \frac{1}{2\Delta_2} \right) \quad (۹۳-۳)$$

$$\Delta_1 = \frac{S_{i-1}}{S_i} (h_i G_i + 2S_i) - \frac{S_{i-1}}{2S_i} h_i G_i \left(\frac{2S_{i-1} - h_i G_{i-1}}{3S_{i-1} - h_i G_{i-1}} \right) \quad (۹۴-۳)$$

$$\Delta_2 = \frac{S_{i-1}}{S_i} (h_i G_i + 2S_i) \left(\frac{3S_{i-1} - h_i G_{i-1}}{2S_{i-1} - h_i G_{i-1}} \right) - \frac{S_{i-1}}{2S_i} h_i G_i \quad (۹۵-۳)$$

در نتیجه با توجه به معادله‌های (۷۶-۳)، (۸۳-۳) و (۹۰-۳) معادله‌ی جبری u_i برای نقاط داخلی

محاسبه می‌گردد [۶۴]:

$$A_i u_{i-1} + B_i u_i + C_i u_{i+1} = D_i \quad (96-3)$$

$$A_i = a_i \left(1 - \frac{G_i h_{i-1}}{3S_i} \right) \quad (97-3)$$

$$B_i = b_i \left(1 - \frac{G_i h_{i+1}}{3S_i} \right) + \frac{h_{i+1}}{6} \left(\frac{2}{S_i} - a_{i+1} \frac{G_{i+1}}{S_{i+1}} \right) + \frac{1}{h_{i+1}} \quad (98-3)$$

$$C_i = \frac{h_{i+1}}{6S_{i+1}} (1 - G_{i+1} b_{i+1}) - \frac{1}{h_{i+1}} \quad (99-3)$$

$$D_i = \frac{h_{i+1}}{6} \left(\frac{2F_i}{S_i} + \frac{F_{i+1}}{S_{i+1}} \right) + c_i \left(\frac{G_i h_{i+1}}{3S_i} - 1 \right) + \frac{G_{i+1}}{6S_i} h_{i+1} c_{i+1} \quad (100-3)$$

در مسائلی که شرایط مرزی آن برحسب مشتق اول است می‌توان از روابط مشتق اول برای نقاط

گرای استفاده کرد. این روابط از معادله‌های (۷۵-۳)، (۷۸-۳) و (۸۳-۳) به دست می‌آید [۶۴]:

$$A'_i m_{i-1} + B'_i m_i + C'_i m_{i+1} = D'_i \quad (101-3)$$

$$A'_i = \frac{1}{3h_i} - \frac{4S_{i-1} + 2S_i - G_{i-1}h_i}{h_i^3 \Delta_i} \quad (102-3)$$

$$B'_i = \frac{2}{3} \left(\frac{1}{h_i} + \frac{1}{h_{i+1}} \right) + \frac{G_i h_{i+1} - 4S_i - 2S_{i+1}}{h_{i+1}^3 \Delta_{i+1}} - \frac{G_i h_i + 2S_{i+1} + 4S_i}{h_i^3 \Delta_i} \quad (103-3)$$

$$C'_i = \frac{1}{3h_{i+1}} - \frac{G_{i+1} h_{i+1} + 2S_i + 4S_{i+1}}{h_{i+1}^3 \Delta_{i+1}} \quad (104-3)$$

$$D'_i = \frac{F_{i+1} - F_i}{h_{i+1}^2 \Delta_{i+1}} + \frac{F_i - F_{i-1}}{h_i^2 \Delta_i} \quad (105-3)$$

$$\Delta_i = 1 + \frac{6(S_{i-1} + S_i)}{h_i^2} \quad (106-3)$$

همچنین می‌توان روابط را برای مشتق دوم نقاط گرای به صورت زیر نوشت [۶۵]:

$$A''_i M_{i-1} + B''_i M_i + C''_i M_{i+1} = D''_i \quad (107-3)$$

$$A''_i = \frac{1}{\Delta''_i} \left(-S_{i+1} + \frac{h_{i+1}}{6} (2G_{i-1} + G_i - \Delta''_i) \right) \quad (108-3)$$

$$B''_i = \frac{1}{\Delta''_i} \left(S_{i+1} + \frac{h_{i+1}}{6} (G_{i-1} + 2G_i + 2\Delta''_i) \right) - \frac{1}{\Delta''_i} \left(-S_i + \frac{h_{i+1}}{6} (2G_i + G_{i+1} - 2\Delta''_{i+1}) \right) \quad (109-3)$$

$$G_i'' = -\frac{1}{\Delta_{i+1}''} \left(S_{i+1} + \frac{h_{i+1}}{6} (G_i + 2G_{i+1} - \Delta_{i+1}'') \right) \quad (110-3)$$

$$D_i'' = \frac{F_{i+1} - F_i}{\Delta_{i+1}''} + \frac{F_i - F_{i-1}}{\Delta_i''} \quad (111-3)$$

$$\Delta_i'' = h_i + G_{i-1} - G_i \quad (112-3)$$

روابط اسپلاین ارائه شده برای حل معادله‌های یک‌بعدی استفاده می‌شود. می‌توان این روابط را برای

حالت دوبعدی هم ارائه کرد. معادله‌ی دیفرانسیل پاره‌ای دوبعدی زیر را در نظر بگیرید:

$$u_t = f(u, u_x, u_{xx}, u_y, u_{yy}) \quad (113-3)$$

اگر $m_{i,j}$ مشتق اول تابع u نسبت به x و $M_{i,j}$ مشتق دوم این تابع نسبت به x و $l_{i,j}$ مشتق اول

آن نسبت به y و $L_{i,j}$ مشتق دوم این تابع نسبت به y باشد. در این صورت معادله‌ی (113-3) را

می‌توان به شکل زیر بازنویسی کرد [۶۵]:

$$(u_t)_{i,j} = f(u_{i,j}, m_{i,j}, M_{i,j}, l_{i,j}, L_{i,j}) \quad (114-3)$$

مشتق زمانی معادله‌ی (114-3) برای حالت دوبعدی در دو گام زمانی نوشته می‌شود. در گام اول

مشتق اول و دوم نسبت به x در زمان جدید $(n + \frac{1}{2})$ نوشته می‌شود (معادله‌ی (115-3)). در گام

دوم مشتق اول و دوم نسبت به y در زمان جدید $(n + 1)$ نوشته می‌شود (معادله‌ی (116-3)).

$$u_{i,j}^{n+\frac{1}{2}} = u_{i,j}^n + \frac{\Delta t}{2} f \left(u_{i,j}^n, m_{i,j}^{n+\frac{1}{2}}, M_{i,j}^{n+\frac{1}{2}}, l_{i,j}^n, L_{i,j}^n \right) \quad (115-3)$$

$$u_{i,j}^{n+1} = u_{i,j}^{n+\frac{1}{2}} + \frac{\Delta t}{2} f \left(u_{i,j}^{n+\frac{1}{2}}, m_{i,j}^{n+\frac{1}{2}}, M_{i,j}^{n+\frac{1}{2}}, l_{i,j}^{n+1}, L_{i,j}^{n+1} \right) \quad (116-3)$$

در حالت دوبعدی فاصله‌ی بین دونقطه‌ی گره‌ای در جهت x ($h_{i,j}$) و فاصله‌ی بین دونقطه‌ی گره‌ای

در جهت y ($k_{i,j}$) به صورت زیر تعریف می‌شود:

$$h_{i,j} = x_{i,j} - x_{i-1,j} > 0 \quad (117-3)$$

$$k_{i,j} = y_{i,j} - y_{i,j-1} > 0 \quad (118-3)$$

رابطه‌ی مشتق اول نسبت به x در نقاط گره‌ای و همچنین مشتق دوم نسبت به x به صورت زیر تعریف

می‌شود:

$$\frac{1}{h_{i,j}} m_{i-1,j} + 2 \left(\frac{1}{h_{i,j}} + \frac{1}{h_{i+1,j}} \right) m_{i,j} + \frac{1}{h_{i+1,j}} m_{i+1,j} = \frac{3(u_{i+1,j} - u_{i,j})}{h_{i+1,j}^2} + \frac{3(u_{i,j} - u_{i-1,j})}{h_{i,j}^2} \quad (119-3)$$

$$h_{i,j} M_{i-1,j} + 2(h_{i,j} + h_{i+1,j}) M_{i,j} + h_{i+1,j} M_{i+1,j} = \frac{6(u_{i+1,j} - u_{i,j})}{h_{i+1,j}} + \frac{6(u_{i,j} - u_{i-1,j})}{h_{i,j}} \quad (120-3)$$

همچنین رابطه‌ی مشتق اول نسبت به y در نقاط گره‌ای و همچنین مشتق دوم نسبت به x

به صورت زیر تعریف می‌شود:

$$\frac{1}{k_{i,j}} l_{i,j-1} + 2 \left(\frac{1}{k_{i,j}} + \frac{1}{k_{i,j+1}} \right) l_{i,j} + \frac{1}{k_{i,j+1}} l_{i,j+1} = \frac{3(u_{i,j+1} - u_{i,j})}{k_{i,j+1}^2} + \frac{3(u_{i,j} - u_{i,j-1})}{k_{i,j}^2} \quad (121-3)$$

$$k_{i,j} L_{i-1,j} + 2(k_{i,j} + k_{i,j+1}) L_{i,j} + k_{i,j+1} L_{i,j+1} = \frac{6(u_{i,j+1} - u_{i,j})}{k_{i,j+1}} + \frac{6(u_{i,j} - u_{i,j-1})}{k_{i,j}} \quad (122-3)$$

به همین ترتیب می‌توان سایر روابط در حالت یک‌بعدی را به حالت دوبعدی تعمیم داد.

۳-۵-۴ حل معادله‌های حاکم با روش (SADI)

برای حل معادله‌های حاکم با روش اسپلین به جهت متغیر (SADI)، معادلات (۳-۲۶) الی (۳-۲۸)

برحسب مشتق‌های اول و دوم کمیت‌های ψ ، ω و θ بازنویسی می‌شود. برای حل مسائل جریان پایا،

سه روش وجود دارد: الف) معادلات غیر پایا حل شود تا به حالت پایا برسد. در این شیوه باید به کل

زمان محاسبات توجه داشت زیرا ممکن است بیش از حد زیاد شود. ب) معادلات پایا را می‌توان

به صورت دستگاه معادلات، شامل دو معادله بیضوی و با استفاده از روش‌های تکراری حل کرد. ج)

معادلات تابع جریان و تاوایی را می‌توان از روش شبه‌گذرا حل کرد. در این روش از زمان مجازی^۱ استفاده می‌گردد. گسسته‌سازی معادله‌ها در جهت i به صورت زیر است.

$$\frac{\psi_{i,j}^{n+\frac{1}{2}} - \psi_{i,j}^n}{\frac{\Delta\tau}{2}} + \nabla^2 \psi_{i,j} = -\omega_{i,j}^n \quad (۱۲۳-۳)$$

$$\frac{\omega_{i,j}^{n+\frac{1}{2}} - \omega_{i,j}^n}{\frac{\Delta\tau}{2}} + u \left(m_{\omega}^{n+\frac{1}{2}} + \eta_x l_{\omega}^n \right) + v \eta_y l_{\omega}^n = \frac{1}{\text{Re}} \frac{\rho_f}{\rho_{nf}} \frac{1}{(1-\varphi)^{2.5}} \nabla^2 \omega_{i,j} \quad (۱۲۴-۳)$$

$$\frac{\theta_{i,j}^{n+\frac{1}{2}} - \theta_{i,j}^n}{\frac{\Delta\tau}{2}} + u \left(m_{\theta}^{n+\frac{1}{2}} + \eta_x l_{\theta}^n \right) + v \eta_y l_{\theta}^n = \frac{1}{\text{RePr}} \frac{k_{nf}}{k_f} \frac{(\rho c_p)_f}{(\rho c_p)_{nf}} \nabla^2 \theta_{i,j} \quad (۱۲۵-۳)$$

که در روابط (۱۲۳-۳) الی (۱۲۵-۳) مؤلفه‌های سرعت و مشتقات به صورت زیر تعریف می‌شوند:

$$u = \eta_y l_{\psi}^n \quad v = - \left(m_{\psi}^{n+\frac{1}{2}} + \eta_x l_{\psi}^n \right) \quad (۱۲۶-۳)$$

$$\nabla^2 = M_{\phi}^{n+\frac{1}{2}} + (\eta_x^2 + \eta_y^2) L_{\phi}^n + 2\eta_x \frac{(m_{\phi})_{i,j+1}^n - (m_{\phi})_{i,j-1}^n}{2\Delta\eta} + (\eta_{xx} + \eta_{yy}) l_{\phi}^n \quad (۱۲۷-۳)$$

$$m_{\phi} = \frac{\partial\phi}{\partial\xi} \quad M_{\phi} = \frac{\partial^2\phi}{\partial\xi^2} \quad l_{\phi} = \frac{\partial\phi}{\partial\eta} \quad L_{\phi} = \frac{\partial^2\phi}{\partial\eta^2} \quad (۱۲۸-۳)$$

در روابط بالا کمیت ϕ می‌تواند هر یک از مقادیر ψ ، ω و θ باشد. با توجه به روش اسپلین

معادله‌های (۱۲۳-۳) الی (۱۲۵-۳) به صورت معادله‌ی (۱۲۹-۳) نوشته می‌شود:

$$\phi_{i,j}^{n+\frac{1}{2}} = F_{i,j} + G_{i,j} m_{i,j}^{n+\frac{1}{2}} + S_{i,j} M_{i,j}^{n+\frac{1}{2}} \quad (۱۲۹-۳)$$

که در رابطه‌ی بالا n زمان معلوم است. مقادیر $F_{i,j}$ ، $G_{i,j}$ و $S_{i,j}$ مقادیر معلوم هستند. این مقادیر معلوم

در جدول ۲-۳ ارائه شده‌اند. با توجه به روابط (۹۶-۳) الی (۱۰۰-۳) می‌توان معادله‌ی (۱۲۹-۳) را

به صورت ماتریس سه‌قطری زیر نوشت:

$$A_{i,j} \phi_{i-1,j}^{n+\frac{1}{2}} + B_{i,j} \phi_{i,j}^{n+\frac{1}{2}} + C_{i,j} \phi_{i+1,j}^{n+\frac{1}{2}} = D_{i,j}^n \quad (۱۳۰-۳)$$

^۱ False Transient

معادله‌ی (۳-۱۳۰) یک ماتریس سه‌قطری است که این ماتریس سه‌قطری را می‌توان با روش الگوریتم توماس حل کرد. با حل معادله‌ی (۳-۱۳۰) مقادیر ψ ، ω و θ در جهت i و در زمان $n + \frac{1}{2}$ به دست می‌آید.

جدول ۲-۳: مقادیر معلوم $F_{i,j}$ ، $G_{i,j}$ و $S_{i,j}$ در زمان n

ψ	$F_{i,j}$	$\psi_{i,j}^n - \frac{\Delta\tau}{2} \left[\omega_{i,j}^n + (\eta_x^2 + \eta_y^2) L_\psi^n + 2\eta_x \frac{(m_\psi)_{i,j+1}^n - (m_\psi)_{i,j-1}^n}{2\Delta\eta} + (\eta_{xx} + \eta_{yy}) l_\psi^n \right]$
	$G_{i,j}$	0
	$S_{i,j}$	$-\frac{\Delta\tau}{2}$
ω	$F_{i,j}$	$\omega_{i,j}^n + \frac{\Delta\tau}{2} \left[-\eta_y l_\psi^n \eta_x l_\omega^n + (m_\psi^n + \eta_x l_\psi^n) \eta_y l_\omega^n + \frac{1}{\text{Re} \rho_{nf}} \frac{\rho_f}{(1-\varphi)^{2.5}} \left[(\eta_x^2 + \eta_y^2) L_\omega^n + 2\eta_x \frac{(m_\omega)_{i,j+1}^n - (m_\omega)_{i,j-1}^n}{2\Delta\eta} + (\eta_{xx} + \eta_{yy}) l_\omega^n \right] \right]$
	$G_{i,j}$	$-\frac{\Delta\tau}{2} \eta_y l_\psi^n$
	$S_{i,j}$	$\frac{\Delta\tau}{2} \frac{1}{\text{Re} \rho_{nf}} \frac{\rho_f}{(1-\varphi)^{2.5}}$
θ	$F_{i,j}$	$\theta_{i,j}^n + \frac{\Delta\tau}{2} \left[-\eta_y l_\psi^n \eta_x l_\theta^n + (m_x^n + \eta_x l_\psi^n) \eta_y l_\theta^n + \frac{1}{\text{RePr}} \frac{k_{nf}}{k_f} \frac{(\rho c_p)_f}{(\rho c_p)_{nf}} \left[(\eta_x^2 + \eta_y^2) L_\theta^n + 2\eta_x \frac{(m_\theta)_{i,j+1}^n - (m_\theta)_{i,j-1}^n}{2\Delta\eta} + (\eta_{xx} + \eta_{yy}) l_\theta^n \right] \right]$
	$G_{i,j}$	$-\frac{\Delta\tau}{2} \eta_y l_\psi^n$
	$S_{i,j}$	$\frac{\Delta\tau}{2} \frac{1}{\text{RePr}} \frac{k_{nf}}{k_f} \frac{(\rho c_p)_f}{(\rho c_p)_{nf}}$

پس از محاسبه‌ی مقادیر ψ ، ω و θ در زمان $n + \frac{1}{2}$ با توجه به روابط (۳-۷۵) الی (۳-۷۹) مقادیر مشتق‌های اول و دوم در زمان $n + \frac{1}{2}$ به دست می‌آید. گسسته‌سازی معادلات در جهت z همانند جهت i است. معادلات گسسته شده در جهت z به صورت زیر است:

$$\frac{\psi_{i,j}^{n+1} - \psi_{i,j}^{n+\frac{1}{2}}}{\frac{\Delta\tau}{2}} + \nabla^2 \psi_{i,j} = -\omega_{i,j}^n \quad (۱۳۱-۳)$$

$$\frac{\omega_{i,j}^{n+1} - \omega_{i,j}^{n+\frac{1}{2}}}{\frac{\Delta\tau}{2}} + u \left(m_{\omega}^{n+\frac{1}{2}} + \eta_x l_{\omega}^{n+1} \right) + v \eta_y l_{\omega}^{n+1} = \frac{1}{\text{Re}} \frac{\rho_f}{\rho_{nf}} \frac{1}{(1-\varphi)^{2.5}} \nabla^2 \omega_{i,j} \quad (۱۳۲-۳)$$

$$\frac{\theta_{i,j}^{n+1} - \theta_{i,j}^{n+\frac{1}{2}}}{\frac{\Delta\tau}{2}} + u \left(m_{\theta}^{n+\frac{1}{2}} + \eta_x l_{\theta}^{n+1} \right) + v \eta_y l_{\theta}^{n+1} = \frac{1}{\text{RePr}} \frac{k_{nf}}{k_f} \frac{(\rho c_p)_f}{(\rho c_p)_{nf}} \nabla^2 \theta_{i,j} \quad (۱۳۳-۳)$$

که در روابط (۱۳۱-۳) الی (۱۳۳-۳) مؤلفه‌های سرعت و مشتقات به صورت زیر تعریف می‌شوند:

$$u = \eta_y l_{\psi}^{n+1} \quad v = - \left(m_{\psi}^{n+\frac{1}{2}} + \eta_x l_{\psi}^{n+1} \right) \quad (۱۳۴-۳)$$

$$\nabla^2 = M_{\phi}^{n+\frac{1}{2}} + (\eta_x^2 + \eta_y^2) L_{\phi}^{n+1} + 2\eta_x \frac{(m_{\phi})_{i+1,j}^{n+\frac{1}{2}} - (m_{\phi})_{i-1,j}^{n+\frac{1}{2}}}{2 \Delta\xi} + (\eta_{xx} + \eta_{yy}) l_{\phi}^{n+1} \quad (۱۳۵-۳)$$

$$m_{\phi} = \frac{\partial \phi}{\partial \xi} \quad M_{\phi} = \frac{\partial^2 \phi}{\partial \xi^2} \quad l_{\phi} = \frac{\partial \phi}{\partial \eta} \quad L_{\phi} = \frac{\partial^2 \phi}{\partial \eta^2} \quad (۱۳۶-۳)$$

با توجه به روش اسپلاین معادله‌های (۱۳۱-۳) الی (۱۳۳-۳) به صورت معادله‌ی (۱۳۷-۳) نوشته می‌شود:

$$\phi_{i,j}^{n+1} = F_{i,j} + G_{i,j} l_{i,j}^{n+1} + S_{i,j} L_{i,j}^{n+1} \quad (۱۳۷-۳)$$

که در رابطه‌ی بالا $n + \frac{1}{2}$ زمان معلوم است. مقادیر $F_{i,j}$ ، $G_{i,j}$ و $S_{i,j}$ مقادیر معلوم (در زمان $n + \frac{1}{2}$) هستند.

این مقادیر معلوم در جدول ۳-۳ ارائه شده‌اند. با توجه به روابط (۹۶-۳) الی (۱۰۰-۳) می‌توان

معادله‌ی (۱۳۷-۳) را به صورت ماتریس سه قطری زیر نوشت:

$$A_{i,j} \phi_{i,j-1}^{n+1} + B_{i,j} \phi_{i,j}^{n+1} + C_{i,j} \phi_{i,j+1}^{n+1} = D_{i,j}^{n+\frac{1}{2}} \quad (۱۳۸-۳)$$

معادله‌ی (۱۳۸-۳) یک ماتریس سه قطری است که با روش الگوریتم توماس حل می‌گردد. با حل

معادله‌ی (۱۳۸-۳) مقادیر ψ ، ω و θ در جهت z و در زمان $n + 1$ به دست می‌آید.

جدول ۳-۳: مقادیر معلوم $F_{i,j}$, $G_{i,j}$ و $S_{i,j}$ در زمان $n + \frac{1}{2}$

ψ	$F_{i,j}$	$\psi_{i,j}^{n+\frac{1}{2}} - \frac{\Delta\tau}{2} \left[\omega_{i,j}^n + M_{\psi}^{n+\frac{1}{2}} + 2\eta_x \frac{(l_{\psi})_{i+1,j}^n - (l_{\psi})_{i-1,j}^n}{2\Delta\xi} \right]$
	$G_{i,j}$	0
	$S_{i,j}$	$-\frac{\Delta\tau}{2}$
ω	$F_{i,j}$	$\omega_{i,j}^{n+\frac{1}{2}} + \frac{\Delta\tau}{2} \left[-\eta_y l_{\psi}^n \eta_x m_{\omega}^{n+\frac{1}{2}} + (m_x^n + \eta_x l_{\psi}^n) \eta_y l_{\omega}^n + \frac{1}{\text{Re} \rho_{nf}} \frac{\rho_f}{(1-\varphi)^{2.5}} \left[(\eta_x^2 + \eta_y^2) M_{\omega}^{n+\frac{1}{2}} + 2\eta_x \frac{(l_{\omega})_{i+1,j}^n - (l_{\omega})_{i-1,j}^n}{2\Delta\xi} + (\eta_{xx} + \eta_{yy}) l_{\omega}^n \right] \right]$
	$G_{i,j}$	$-\frac{\Delta\tau}{2} \eta_y l_{\psi}^n$
	$S_{i,j}$	$\frac{\Delta\tau}{2} \frac{1}{\text{Re} \rho_{nf}} \frac{\rho_f}{(1-\varphi)^{2.5}}$
θ	$F_{i,j}$	$\theta_{i,j}^{n+\frac{1}{2}} + \frac{\Delta\tau}{2} \left[-\eta_y l_{\psi}^n \eta_x m_{\theta}^{n+\frac{1}{2}} + (m_x^n + \eta_x l_{\psi}^n) \eta_y l_{\theta}^n + \frac{1}{\text{RePr}} \frac{k_{nf}}{k_f} \frac{(\rho c_p)_f}{(\rho c_p)_{nf}} \left[(\eta_x^2 + \eta_y^2) M_{\theta}^{n+\frac{1}{2}} + 2\eta_x \frac{(l_{\theta})_{i+1,j}^{n+\frac{1}{2}} - (l_{\theta})_{i-1,j}^{n+\frac{1}{2}}}{2\Delta\eta} + (\eta_{xx} + \eta_{yy}) l_{\theta}^n \right] \right]$
	$G_{i,j}$	$-\frac{\Delta\tau}{2} \eta_y l_{\psi}^n$
	$S_{i,j}$	$\frac{\Delta\tau}{2} \frac{1}{\text{RePr}} \frac{k_{nf}}{k_f} \frac{(\rho c_p)_f}{(\rho c_p)_{nf}}$

پس از محاسبه‌ی مقادیر ψ , ω و θ در زمان $n + 1$ با توجه به روابط (۳-۷۵) الی (۳-۷۹) مقادیر مشتق‌های اول و دوم در زمان $n + 1$ به دست می‌آید. این فرآیند تا رسیدن به حل پایا ادامه می‌یابد. معیار همگرایی به صورت زیر تعریف می‌شود:

$$\max(|\phi_{i,j}^{n+1} - \phi_{i,j}^n|) < \epsilon \quad (۳-۱۳۹)$$

که مقدار ϵ برای تابع جریان و ورتیسیتی $10^{-۴}$ و برای تابع دما $10^{-۵}$ در نظر گرفته می‌شود.

۳-۶ مزایای روش (SADI)

مزایای اصلی استفاده از روش اسپلین با جهت متغیر نسبت به دیگر روش‌های تفاضل محدود به شرح زیر است:

- ✓ دستگاه ماتریس معادلات حاکم همیشه سه قطری بوده در نتیجه استفاده از الگوریتم توماس در این روش همیشه وجود دارد.
- ✓ دستگاه ماتریس معادلات حاکم به یک مجموعه معادله‌های اسکالر کاهش می‌یابد که معادلات شامل خود تابع هستند. مشتقات اول و دوم برای نقاط گره‌ای در فرمول‌های سه‌قطری نگه‌داری می‌شود.
- ✓ این روش برای هر دو مش‌بندی یکنواخت و غیریکنواخت به کار می‌رود. با این حال برای مش‌بندی یکنواخت تقریب اسپلین برای مشتق اول از دقت چهار و برای مش‌بندی غیریکنواخت از دقت سه است. تقریب اسپلین برای مشتق دوم برای هر دو مش‌بندی از مرتبه‌ی دو است.
- ✓ از آنجایی که مشتقات اول و دوم در این روش مستقیم و از فرمول‌های سه‌قطری به دست می‌آید نیازی به گسسته‌سازی تفاضل محدود و مشکلات ناشی از آن نیست.
- ✓ روش (SADI) تمام مزایای روش‌های تفاضل محدود مرسوم را دارد. این روش به شبکه‌ی محاسباتی خیلی ریز نیازی ندارد. به همین دلیل این روش زمان حل و هزینه‌ی کامپیوتری کمی را دارد.

فصل ۲: نتایج

۴-۱ مقدمه

هنگام گسسته‌سازی مسائل مهندسی به صورت ضمنی با روش‌هایی مانند تفاضل محدود^۱ یا حجم محدود^۲ دستگاه‌های خطی پدیدار می‌شوند. برای حل دستگاهی مانند $Ax = b$ می‌توان با معکوس کردن ماتریس ضرایب (A) و ضرب آن در طرفین معادله بردار مجهولات را به دست آورد. اما چنین عملیاتی از نظر هزینه‌ی محاسباتی بسیار سنگین و وقت‌گیر است. مخصوصاً در مسائل غیرخطی مانند حل معادلات ناویر استوکس که برای حل باید با تکرار به همگرایی رسید، بارها و بارها وارون کردن یک ماتریس بسیار نامطلوب است. در این بین روش‌های دیگری که از لحاظ پیچیدگی دارای روندی ساده‌تر و هزینه‌ی محاسباتی مطلوب هستند مانند گرادیان مزدوج، روش حذفی گوس، روش فاکتورگیری LU و غیره که هر کدام نقاط ضعف و قوت مخصوص به خود را دارند موجود هستند.

در بین دستگاه معادلات، دستگاه معادلات سه‌قطری یکی از پرکاربردترین انواع هستند که در روش جهت متناوب ضمنی^۳ [۶۶]، حل گر طیفی پواسن^۴ [۶۷]، تقریب $Spline$ درجه‌ی سه^۵ و روش‌های پیش‌شرط‌سازی^۶ برای حل‌های تکراری و غیره ظاهر می‌شوند. یک روش بسیار مؤثر و سریع در حل چنین دستگاه‌هایی الگوریتم توماس^۷ است که بر پایه‌ی روش حذفی گوس می‌باشد.

گسسته‌سازی به روش ضمنی اسپلاین با جهت متغیر (SADI) به دستگاه معادلات سه‌قطری می‌انجامد. به منظور حل دستگاه معادلات سه‌قطری حاصل برنامه‌ای عددی با زبان برنامه‌نویسی C++ و CUDAC++ نوشته شده است. به منظور اعتبارسنجی مطالعه‌ی انجام شده، نتایج عددی موجود با نتایج عددی به دست آمده توسط وانگ و چن [۲۴] مقایسه می‌شود. پس از بررسی صحت نتایج حاصل از الگوریتم، تأثیر هندسه و نانوسیال روی انتقال حرارت مورد بررسی قرار می‌گیرد. در نهایت به منظور

¹ Finite Difference

² Finite Volume

³ Alternating Direction Implicite Method (ADI)

⁴ Spectral Poisson

⁵ Cubic Spline Approxmiation

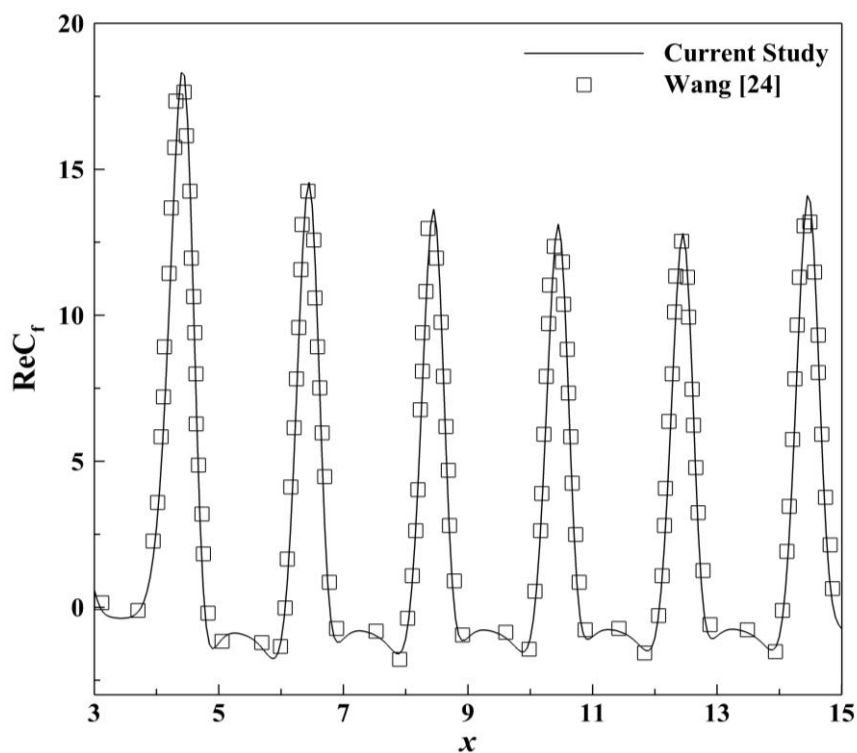
⁶ Preconditioner

⁷ Thomas Algorithm

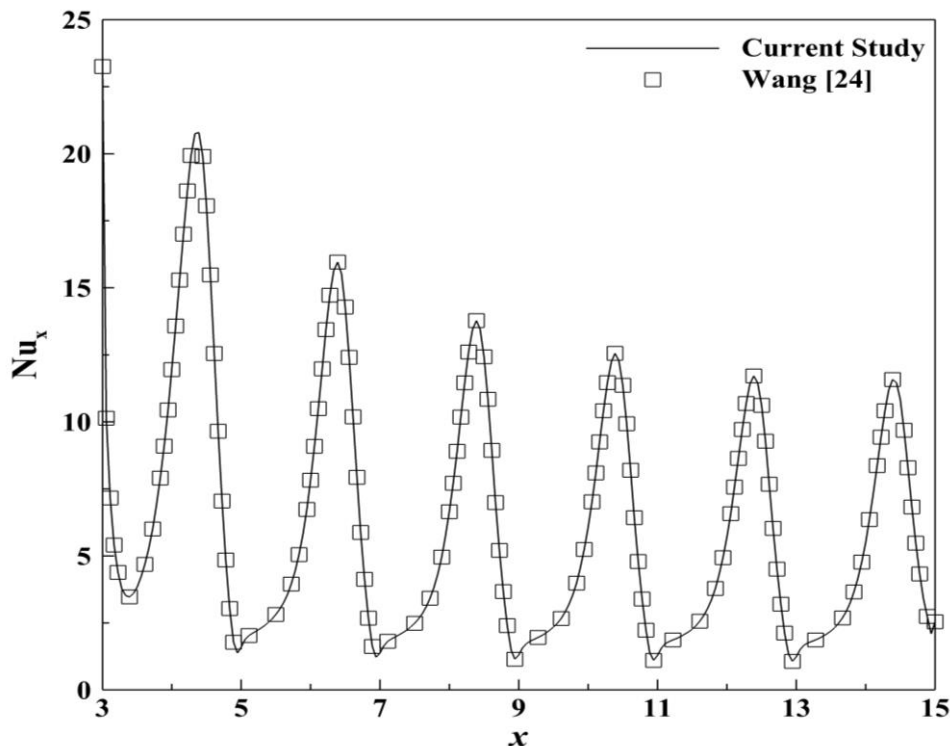
به دست آوردن ضریب تسریع به مقایسه‌ی زمان حل عددی روی پردازنده‌ی مرکزی و پردازنده‌ی گرافیکی می‌پردازیم.

۲-۴ صحت‌سنجی و اعتبار‌سنجی

برای مقایسه‌ی نتایج به‌دست‌آمده از روش اسپلین با جهت متغیر (SADI) در این مطالعه، از نتایج عددی به‌دست‌آمده از همین روش عددی توسط وانگ و چن [۲۴] استفاده‌شده است. در شکل ۱-۴ ضریب اصطکاک پوسته‌ای و در شکل ۲-۴ عدد ناسلت محلی در شرایط $Re = 300$, $Pr = 6.93$ و $a = 0.2$ مورد مقایسه قرار گرفته است. با توجه به نتایج به‌دست‌آمده، الگوریتم عددی حاضر قابلیت و دقت لازم برای حل جریان در کانال با دیواره‌های موج‌دار را داراست.



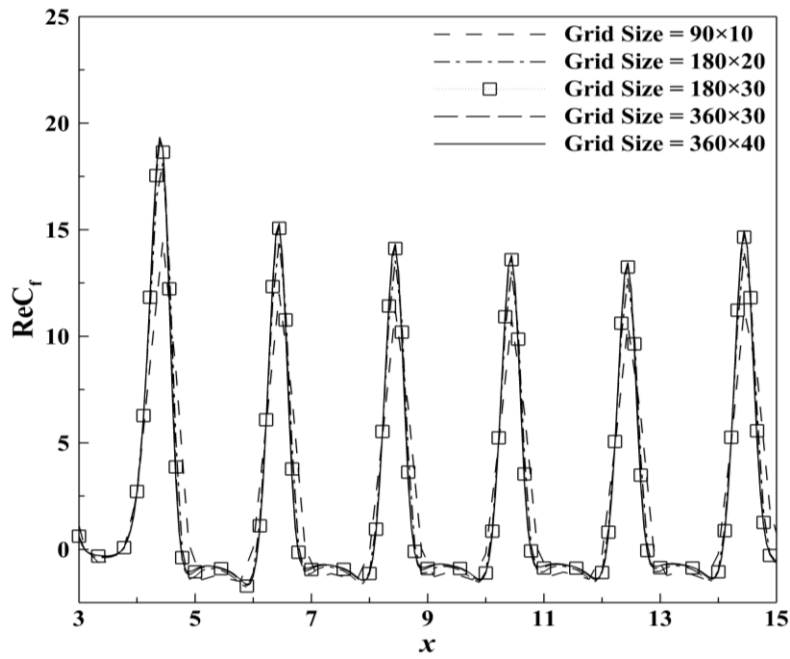
شکل ۱-۴: ضریب اصطکاک پوسته‌ای در شرایط $Re = 300$, $Pr = 6.93$ و $a = 0.2$



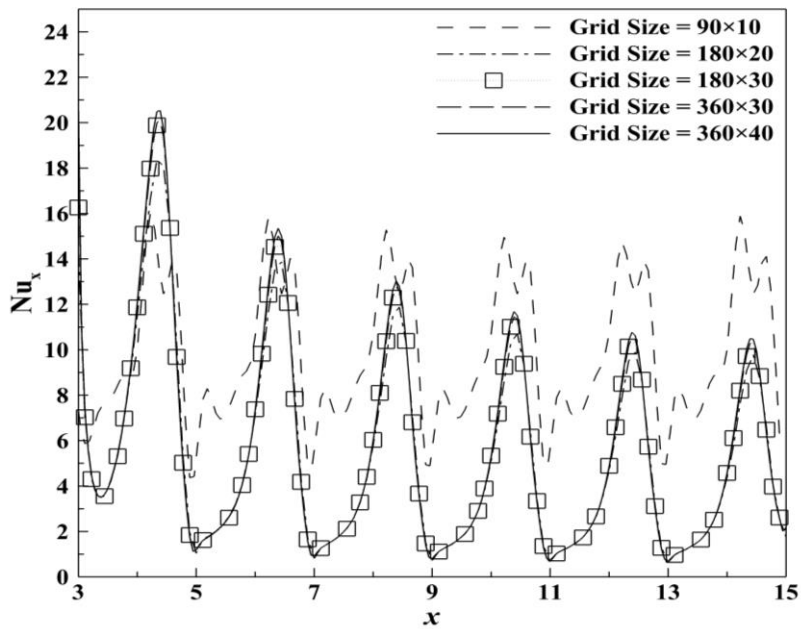
شکل ۴-۲: عدد ناسلت محلی در شرایط $Pr = 6.93$, $Re = 300$ و $a = 0.2$

۴-۳ شبکه‌ی محاسباتی

دامنه‌ی محاسباتی پس از بررسی استقلال حل از شبکه‌ی محاسباتی یک مستطیل با اندازه‌ی شبکه‌ی 180×30 در نظر گرفته می‌شود. بدین منظور پنج شبکه با ابعاد 90×10 ، 180×20 ، 180×30 ، 360×30 و 360×40 در نظر گرفته شده است. نتایج مربوط به ضریب اصطکاک پوسته‌ای و عدد ناسلت محلی در شرایط $Pr = 6.93$ ، $Re = 300$ و $a = 0.2$ در شکل ۴-۳ و شکل ۴-۴ نمایش داده شده است. همان‌طور که مشاهده می‌شود نتایج شبکه با ابعاد 180×30 ، 360×30 و 360×40 هم‌پوشانی و انطباق خوبی دارند، با توجه به این‌که مقادیر عدد ناسلت محلی و ضریب اصطکاک پوسته‌ای در این سه شبکه از یک‌مرتبه هستند برای بررسی اثر دامنه‌ی موج (a)، عدد رینولدز (Re)، عدد پرانتل (Pr)، درصد نانوسیال (ϕ) و قطر نانوسیال (d_p) بر روی جریان نانوسیال از شبکه‌ی محاسباتی با اندازه‌ی 180×30 استفاده شده است.



شکل ۳-۴: مطالعه‌ی استقلال حل از شبکه برای ضریب اصطکاک پوسته‌ای در شرایط $Pr = 6.93$ ، $a = 0.2$ و $Re = 300$



شکل ۴-۴: مطالعه‌ی استقلال حل از شبکه برای عدد ناسلت محلی در شرایط $Pr = 6.93$ ، $Re = 300$ و $a = 0.2$

۴-۴ نتایج عددی

همان‌طور که در بخش‌های قبل بیان گردید، جریان یک نانوسیال در کانال با دیواره‌ی موج‌دار به‌صورت عددی با روش (SADI) مورد بررسی قرار گرفته است. در این بخش به بررسی اثر دامنه‌ی موج (a) ، عدد رینولدز (Re) ، عدد پرانتل (Pr) ، درصد نانوسیال (φ) و قطر نانوسیال (d_p) بر روی جریان پرداخته شده است.

۴-۴-۱ بررسی تأثیر دامنه‌ی موج دیواره‌های کانال

در این بخش به بررسی تأثیر دامنه‌ی موج بر ضریب اصطکاک پوسته‌ای، عدد ناسلت محلی، عدد ناسلت متوسط، خطوط جریان و خطوط هم‌دما در شرایط $Re = 300$ ، $Pr = 6.93$ ، $\varphi = 0\%$ پرداخته شده است. همان‌طور که در شکل ۴-۵ مشاهده می‌شود به دلیل وجود شرط ورودی کاملاً توسعه‌یافته ضریب اصطکاک پوسته‌ای، در دامنه‌ی موج $a = 0$ (کانال با دیواره‌های صاف) در تمام نقاط ثابت می‌ماند. برای دامنه‌ی موج $a = 0.1$ منحنی‌های هارمونیک ضریب اصطکاک محلی دارای فرکانسی همانند سطح موج‌دار است و مقادیر بیشینه و کمینه تقریباً در بیشینه و کمینه‌ی سطح موج‌دار اتفاق می‌افتد. با افزایش دامنه‌ی موج، جریان برگشتی در قسمت واگرا شده اتفاق می‌افتد و اندازه‌ی این ناحیه افزایش می‌یابد. جریان برگشتی باعث منفی شدن ضریب اصطکاک در ناحیه‌ی واگرا شده‌ی صفحه‌ی موج‌دار می‌شود. باید به این نکته توجه شود که افزایش اندازه‌ی ناحیه‌ی جریان برگشتی سبب می‌شود که منحنی هارمونیک ضریب اصطکاک نسبت به سطح موج‌دار کانال قابل توجه باشد. همان‌طور که در شکل ۴-۶ مشاهده می‌شود برای دامنه‌ی موج $a = 0$ لایه‌ی مرزی در ورودی کانال نازک‌تر از هر نقطه‌ی دیگری در امتداد کانال است به همین دلیل ناسلت محلی در ورودی کانال بیشترین مقدار را نسبت به هر نقطه‌ای در امتداد کانال دارد. عدد ناسلت محلی در بخش همگرا شده‌ی کانال بیشتر از قسمت واگرای شده‌ی کانال است. این پدیده به این علت است که ناحیه‌ی

همگرا شده دارای سرعت میانگین و گرادیان سرعت بالایی است که باعث افزایش نرخ انتقال حرارت می‌شود. برعکس، جریان برگشتی گرادیان سرعت پایینی در نزدیکی سطح دیواره در قسمت واگرا شده‌ی کانال دارد، که نسبت انتقال حرارت را کاهش می‌دهد. هر چه دامنه‌ی موج افزایش می‌یابد اختلاط سیال بیشتر شده و تبادل گرما بین سیال و دیواره بهتر می‌شود در نتیجه عدد ناسلت محلی افزایش می‌یابد.

پس در دامنه‌ی موج $a = 0.1$ عدد ناسلت محلی در بخش همگرا شده افزایش و در بخش واگرا شده کاهش می‌یابد. برای $a = 0.1$ مقدار کمینه‌ی عدد ناسلت در فاصله‌ی کمی از نقطه‌ی بیشینه‌ی هر موج واقع است و این فاصله با افزایش دامنه‌ی موج به $a = 0.2$ افزایش می‌یابد. این پدیده بدین علت است که وقتی که دامنه‌ی $a = 0.2$ افزایش می‌یابد، قدرت جریان برگشتی نیز افزایش می‌یابد و نقطه‌ی جدایی جریان به نقطه‌ی کمینه‌ی کانال موج‌دار نزدیک‌تر می‌شود. نکته‌ی قابل‌توجه دیگر این است که به دلیل لایه‌ی مرزی حرارتی تشکیل‌شده، بیشترین مقدار عدد ناسلت در موج اول است و در امتداد کانال به دلیل کاهش گرادیان دما در دیوار روند ناسلت محلی کاهش می‌یابد، اگرچه توزیع عدد ناسلت دارای الگوی ثابتی در طول کانال است.

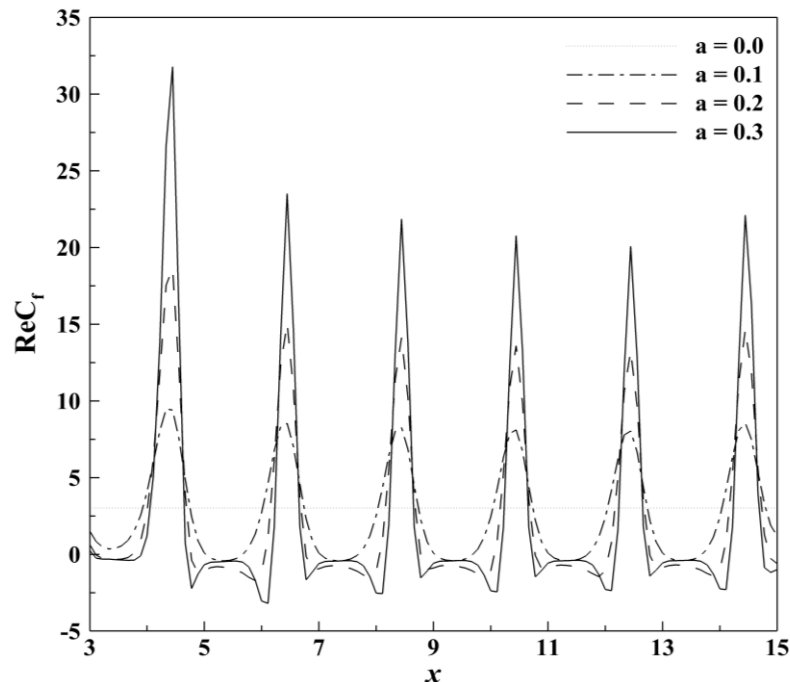
مطابق شکل ۴-۷ بیشترین عدد ناسلت متوسط متعلق به اولین موج است و در پایین‌دست جریان از موج سوم به بعد عدد ناسلت یکنواخت است این نتیجه به دلیل این است که جریان به‌طور کامل توسعه‌یافته است. عدد ناسلت متوسط در دامنه‌ی موج کم، تغییرات زیادی ندارد. به این دلیل که عدد ناسلت در بخش واگرا شده‌ی هر موج سهم کمی دارد ولی در دامنه‌های موج بیشتر عدد ناسلت متوسط تغییرات بیشتری دارد.

در شکل ۴-۸ خطوط جریان برای دامنه‌های موج مختلف در شرایط $\varphi = 0\%$ ، $Pr = 6.93$ ، $Re = 300$ مورد بررسی قرار گرفته است. مطابق این شکل به دلیل وجود جریان برگشتی در بازه‌ی $4.5 \leq x \leq 6.5$ نقطه‌ی جدایش جریان در $x = 5.1$ و نقطه‌ی اتصال در $x = 5.67$ اتفاق می‌افتد. با

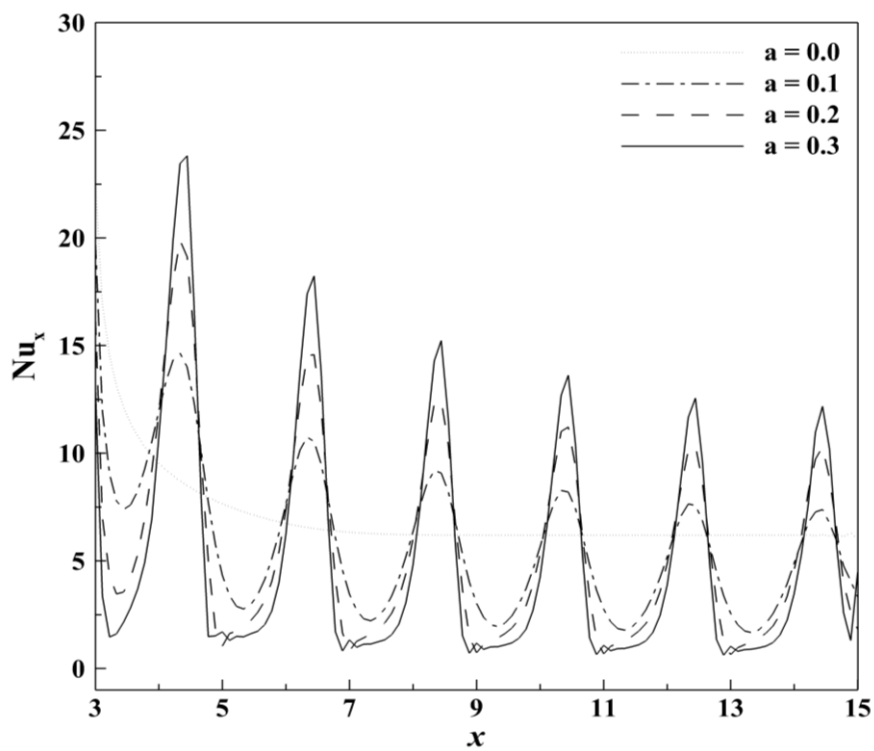
افزایش دامنه‌ی موج فاصله‌ی بین نقاط جدایش و اتصال افزایش می‌یابد. در شکل ۹-۴ خطوط هم‌دما برای دامنه‌های مختلف در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$ مورد بررسی قرار گرفته است. همان‌طور که در شکل ۹-۴ مشاهده می‌شود با افزایش دامنه، گرادیان دمای نزدیک دیوار افزایش می‌یابد و باعث اختلاط سیال سرد در مرکز با سیال گرم نزدیک دیواره می‌شود. در جدول ۱-۴ تغییرات عدد ناسلت متوسط برای دامنه‌های موج مختلف در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$ مورد بررسی قرار گرفته است. مطابق این جدول افزایش دامنه‌ی موج کانال سبب افزایش عدد ناسلت متوسط می‌گردد.

جدول ۱-۴: تأثیر دامنه‌ی موج روی عدد ناسلت متوسط در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$

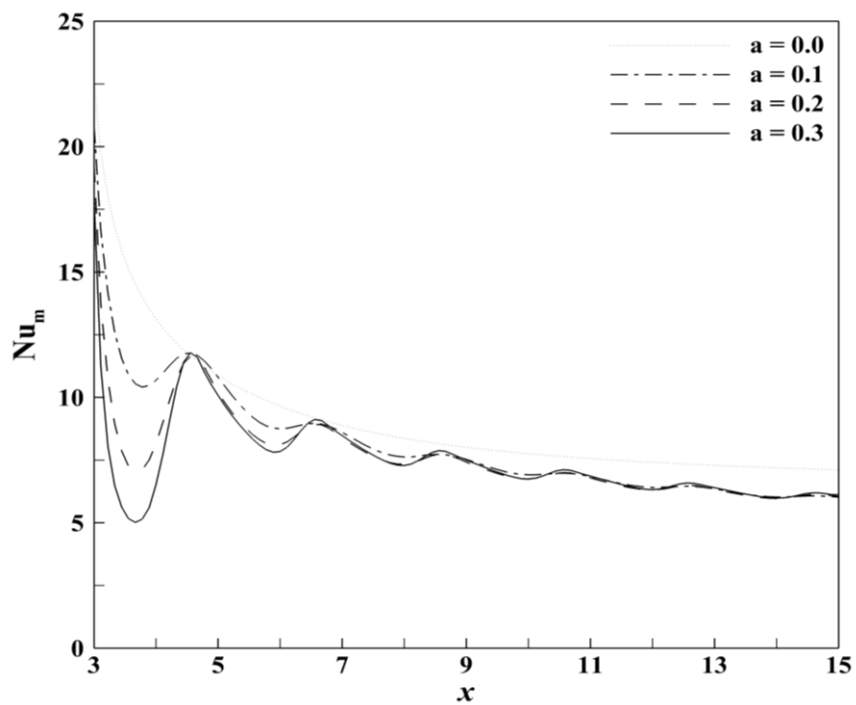
a	۰/۱	۰/۲	۰/۳
Nu_m	۶/۰۷۴۲۱	۶/۱۰۴۰۳	۶/۱۷۷۳۸



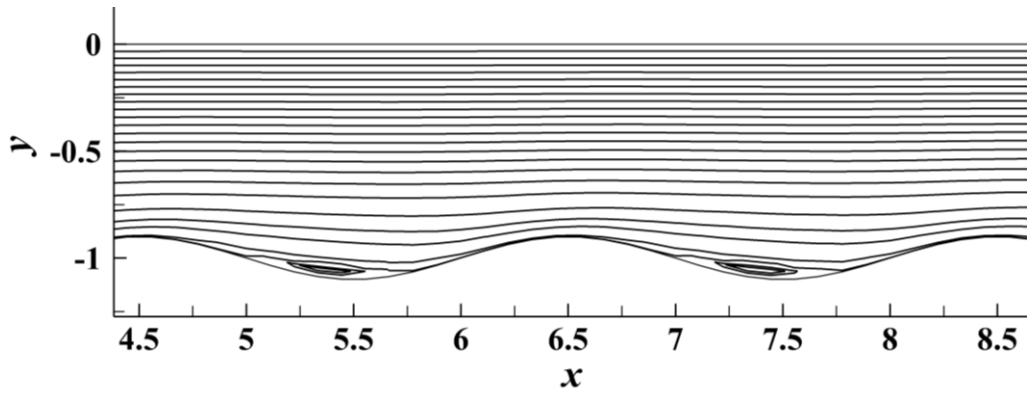
شکل ۹-۵: ضریب اصطکاک پوسته‌ای محلی در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$



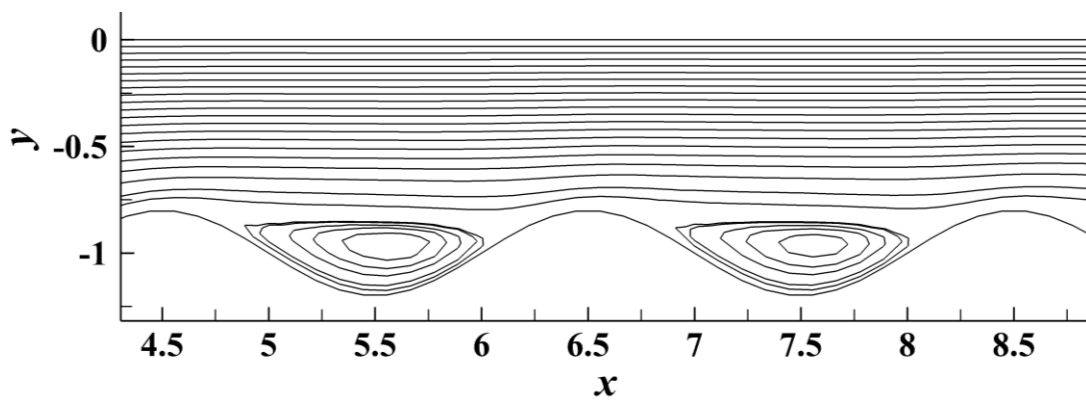
شکل ۴-۶: عدد ناسلت محلی در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$



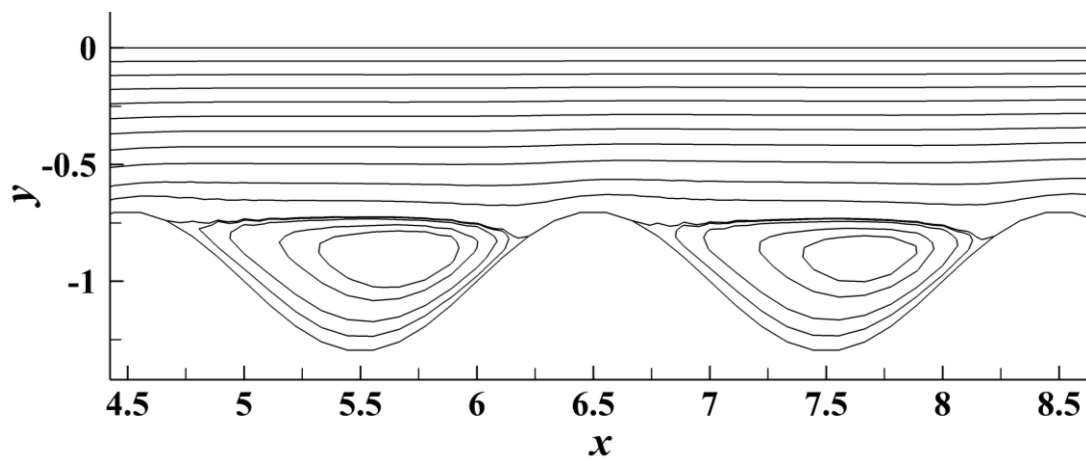
شکل ۴-۷: عدد ناسلت متوسط در شرایط $Re = 300$, $Pr = 6.93$, $\varphi = 0\%$



الف: $a = 0.1$

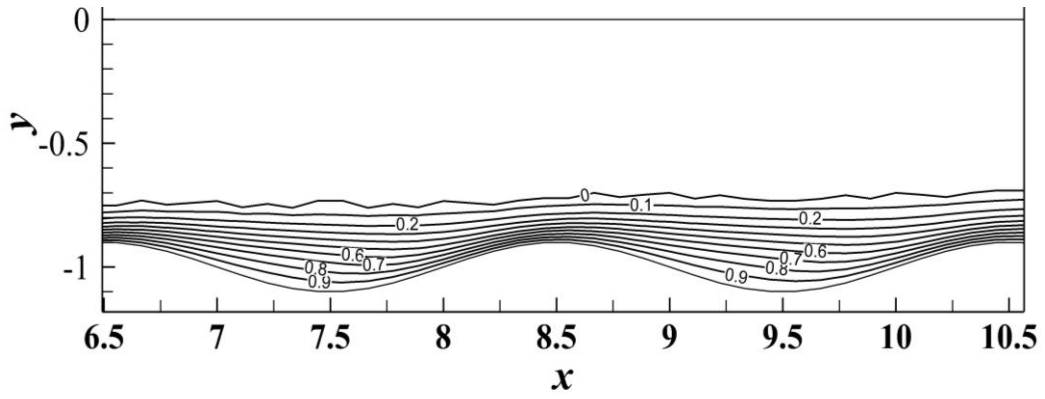


ب: $a = 0.2$

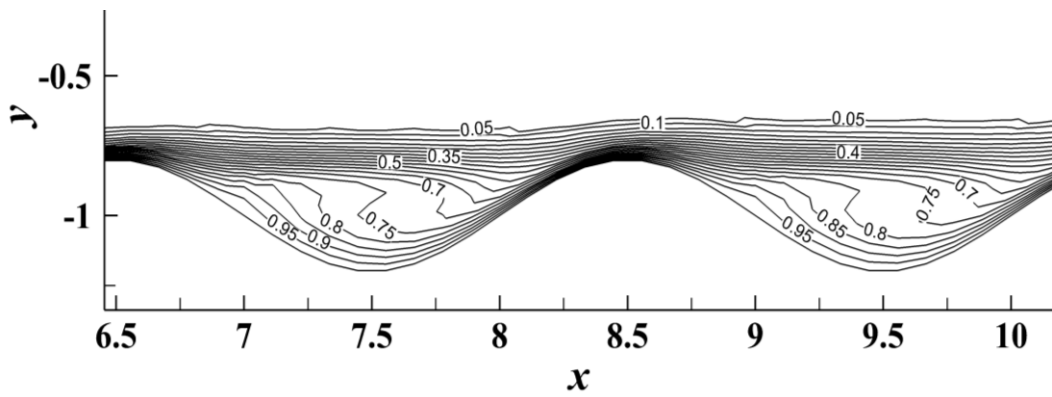


ج: $a = 0.3$

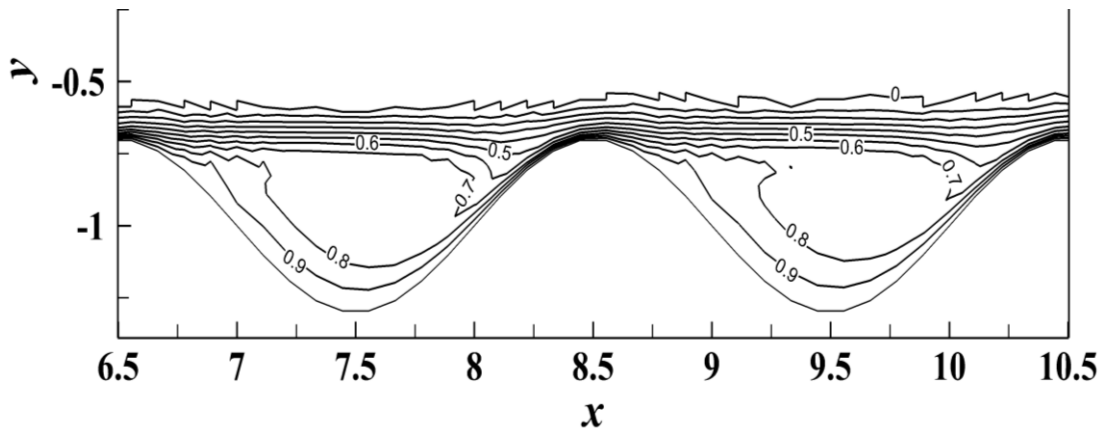
شکل ۴-۸: نمودار خط جریان در شرایط $Re = 300$ ، $Pr = 6.93$ ، $\varphi = 0\%$



الف: $a = 0.1$



ب: $a = 0.2$



ج: $a = 0.3$

شکل ۹-۴: نمودار خطوط هم‌دما در شرایط $\varphi = 0\%$, $Pr = 6.93$, $Re = 300$

۴-۴-۲ بررسی تأثیر عدد رینولدز

در این بخش اثر عدد رینولدز بر ضریب اصطکاک پوسته‌ای، عدد ناسلت محلی، عدد ناسلت متوسط، خطوط جریان و خطوط هم‌دما در شرایط $\phi = 0\%$ ، $Pr = 6.93$ ، $a = 0.2$ مورد بررسی قرار گرفته است.

با افزایش عدد رینولدز گرادیان سرعت در نزدیکی دیواره افزایش می‌یابد در نتیجه ضریب اصطکاک پوسته‌ای افزایش می‌یابد (شکل ۴-۱۰). همان‌طور که ملاحظه می‌شود در اعداد رینولدز پایین جدایش جریان روی نمی‌دهد. با کاهش عدد رینولدز و رسیدن عدد رینولدز به ۱۰۰ گردابه تشکیل می‌شود. لازم به ذکر است زمانی که عدد رینولدز از ۵۰۰ به ۱۰۰ کاهش می‌یابد مقدار کمینه‌ی ضریب اصطکاک پوسته‌ای محلی هر موج افزایش می‌یابد تا به مقدار صفر می‌رسد و دلیل این امر آن است که در عدد رینولدز ۱۰۰ جمله‌ی غیرخطی جابه‌جایی نسبت به جمله‌ی نفوذ کوچک است. مطابق شکل ۴-۱۱ با افزایش عدد رینولدز عدد ناسلت افزایش می‌یابد. با افزایش عدد رینولدز مقادیر عدد ناسلت در قسمت هم‌گرا شده افزایش می‌یابد ولی در قسمت واگرا شده تقریباً ثابت می‌ماند. در شکل ۴-۱۲ عدد ناسلت متوسط برای مقادیر مختلف رینولدز رسم شده است. بالاترین متوسط نرخ انتقال حرارت در موج اول اتفاق می‌افتد. از آنجایی که عدد ناسلت متوسط از توزیع عدد ناسلت محلی در طول کانال به دست می‌آید انتظار می‌رود دامنه‌ی آن در طول کانال کاهش یابد. نقاط بیشینه و کمینه‌ی نمودار مربوط به بخش‌های هم‌گرا شده و واگرای شده‌ی دیواره‌ی کانال است.

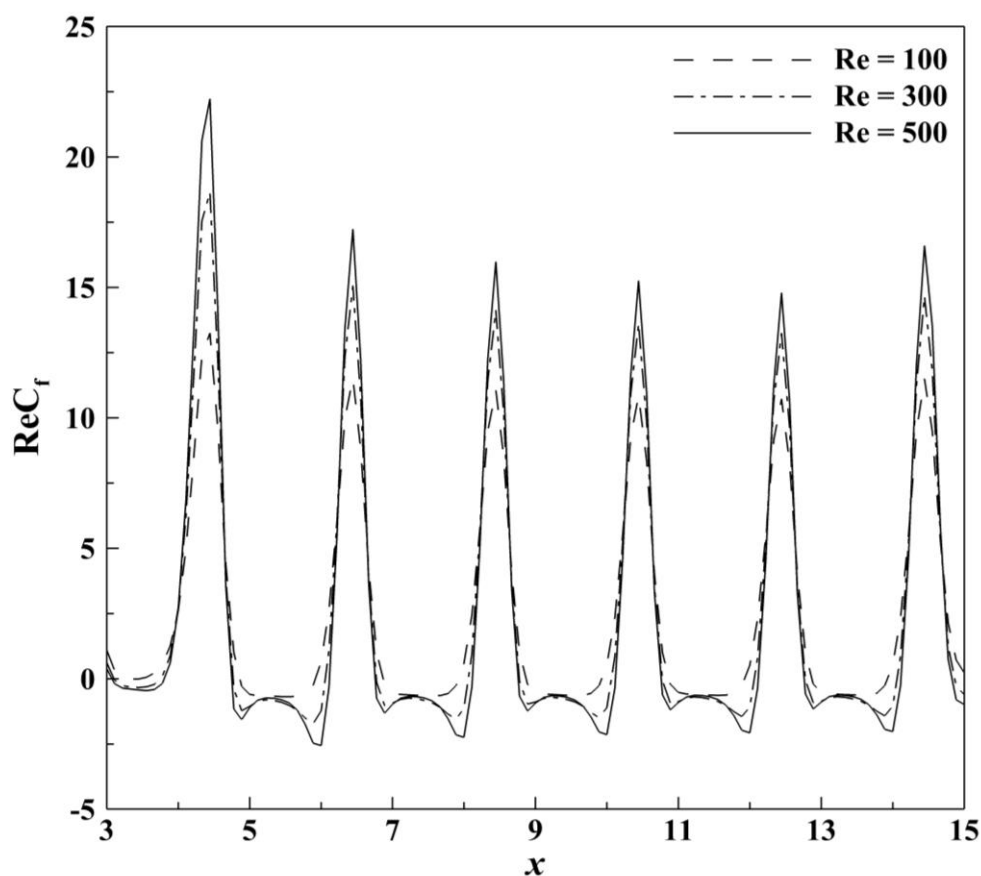
مطابق شکل ۴-۱۳ خطوط جریان برای اعداد رینولدز مختلف مورد بررسی قرار گرفته است. محل جدایش جریان و محل تشکیل مجدد لایه‌ی مرزی در اعداد رینولدز مختلف نشان داده شده است. ملاحظه می‌شود که با افزایش عدد رینولدز فاصله‌ی نقاط جدایش و تشکیل مجدد لایه‌ی مرزی بیشتر می‌شود. در این شکل مشاهده می‌شود که با افزایش عدد رینولدز اندازه‌ی گردابه بزرگ‌تر می‌شود. علاوه بر این، محل نقاط جدایش و اتصال به سمت بخش‌های هم‌گرا شده حرکت می‌کند.

در شکل ۴-۱۴ خطوط هم‌دما برای اعداد رینولدز مختلف مورد بررسی قرار گرفته است. همان‌طور که در شکل مشاهده می‌شود با افزایش عدد رینولدز گرادیان دمای نزدیک دیوار افزایش می‌یابد و باعث اختلاط سیال سرد در مرکز با سیال گرم نزدیک دیواره می‌شود.

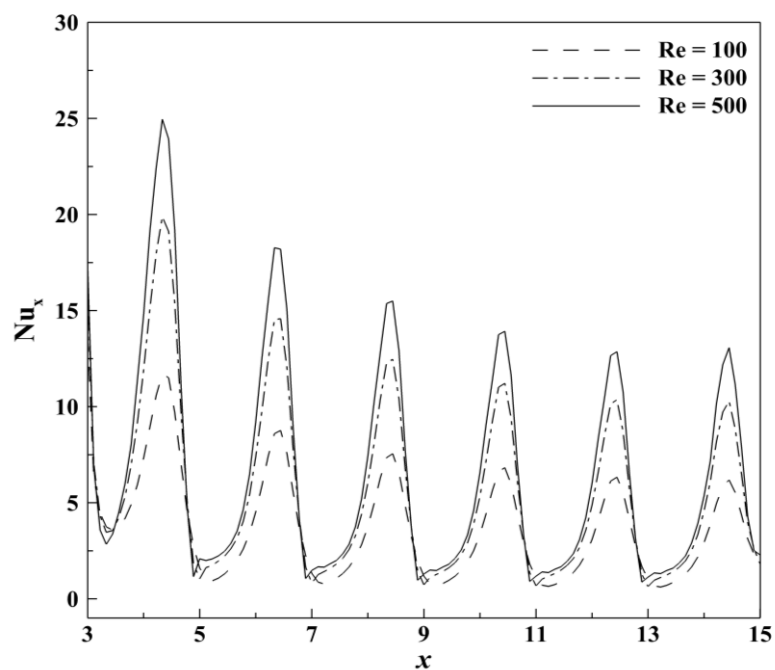
جدول ۴-۲ تغییرات عدد ناسلت متوسط را همراه با تغییرات عدد رینولدز نشان می‌دهد، همان‌طور که از این جدول مشخص است، افزایش عدد رینولدز سبب افزایش عدد ناسلت متوسط می‌شود.

جدول ۴-۲: تأثیر عدد رینولدز روی عدد ناسلت متوسط در شرایط $\varphi = 0\%$ ، $Pr = 6.93$ ، $a = 0.2$

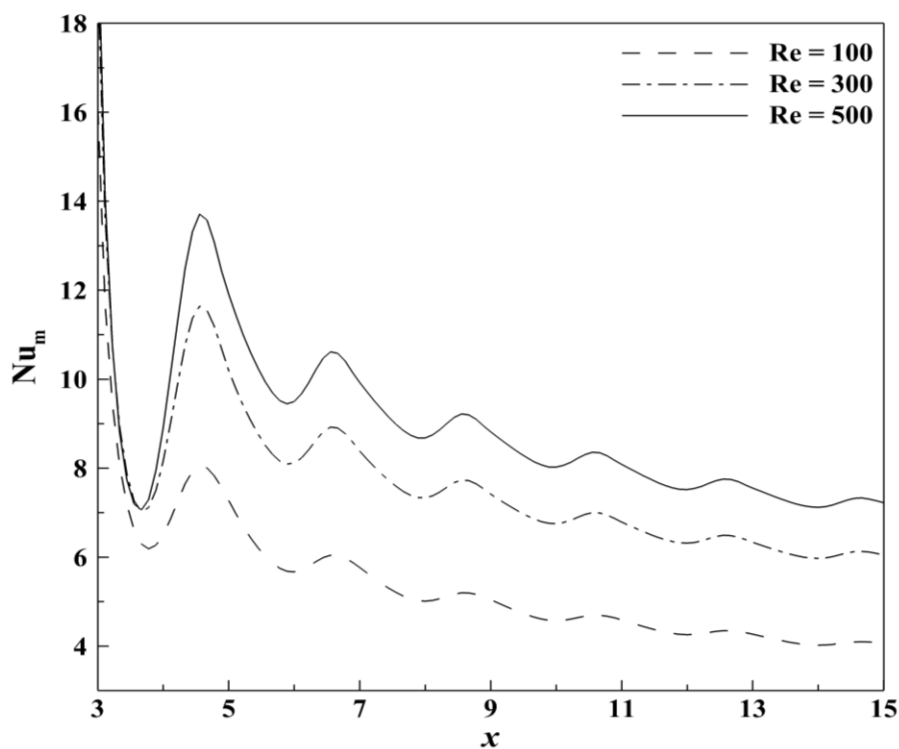
Re	۱۰۰	۳۰۰	۵۰۰
Nu_m	۴/۰۹۹۵۷	۶/۱۰۴۰۳	۷/۲۹۰۷۹



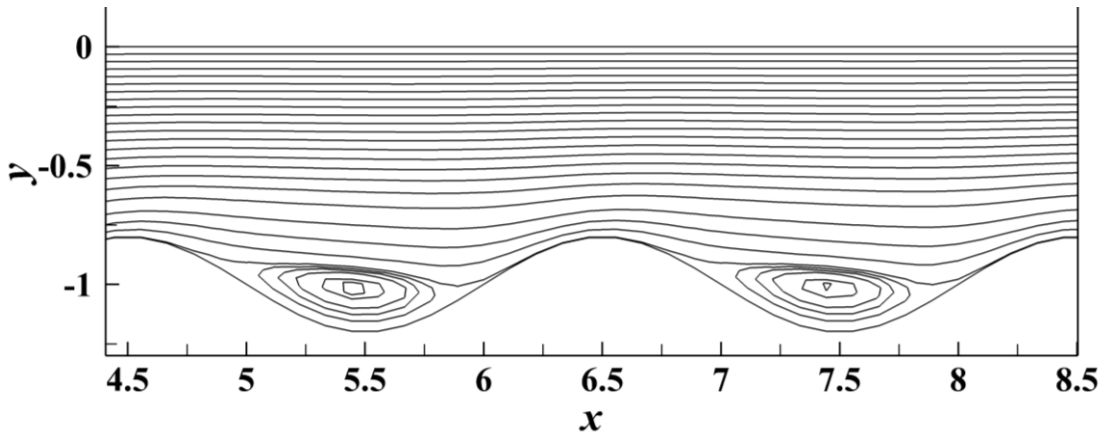
شکل ۴-۱۰: ضریب اصطکاک پوسته‌ای در شرایط $\varphi = 0\%$ ، $Pr = 6.93$ ، $a = 0.2$



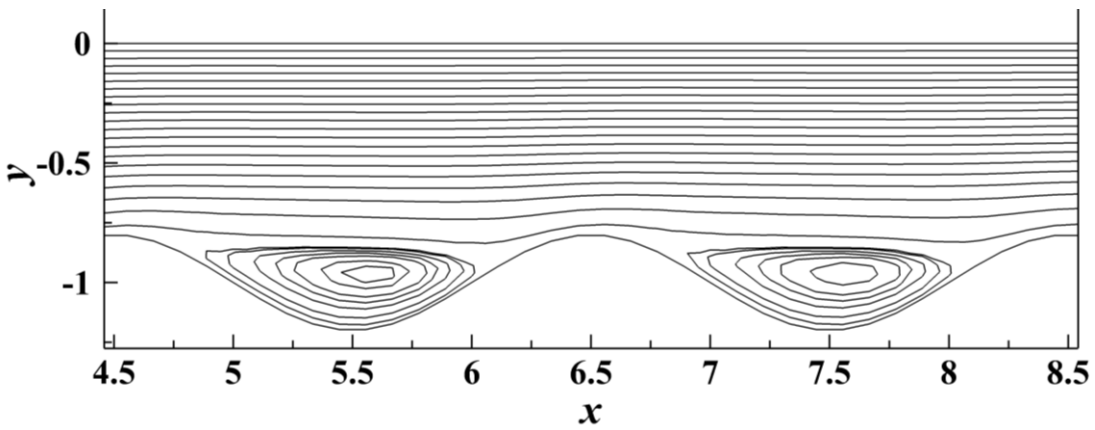
شکل ۴-۱۱: عدد ناسلت محلی در شرایط $a = 0.2$, $Pr = 6.93$, $\varphi = 0\%$



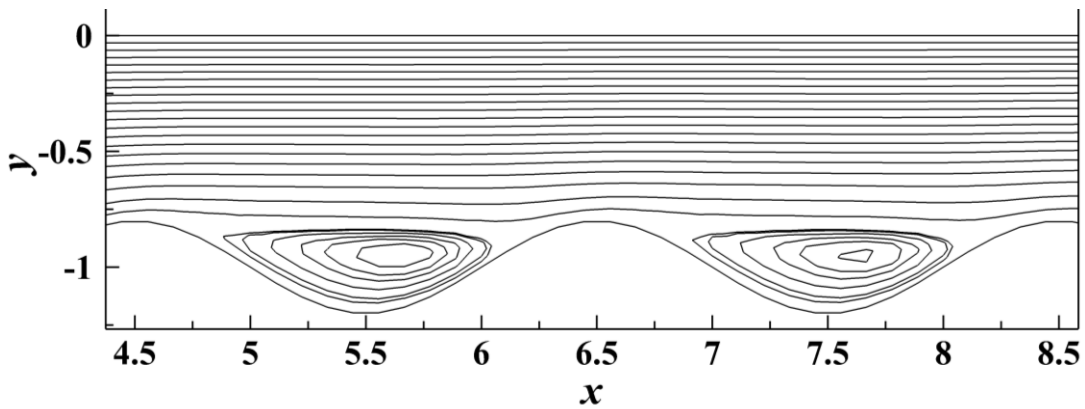
شکل ۴-۱۲: عدد ناسلت متوسط در شرایط $a = 0.2$, $Pr = 6.93$, $\varphi = 0\%$



الف: $Re = 100$

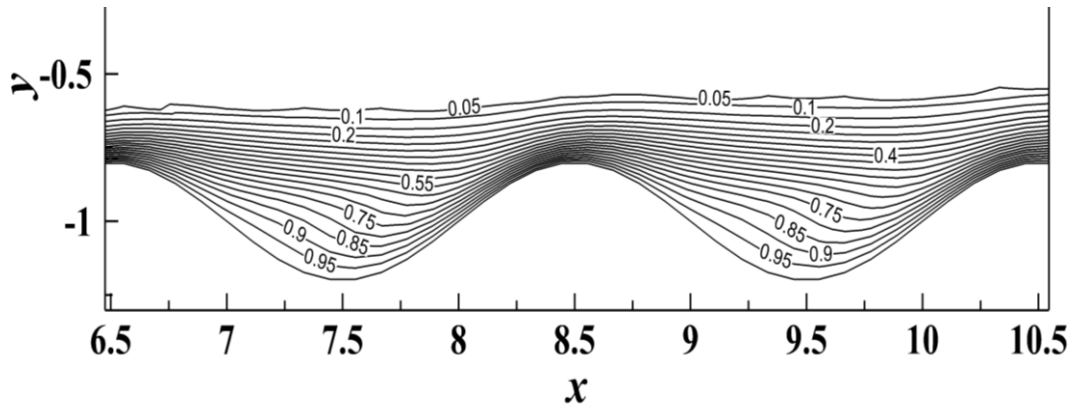


ب: $Re = 300$

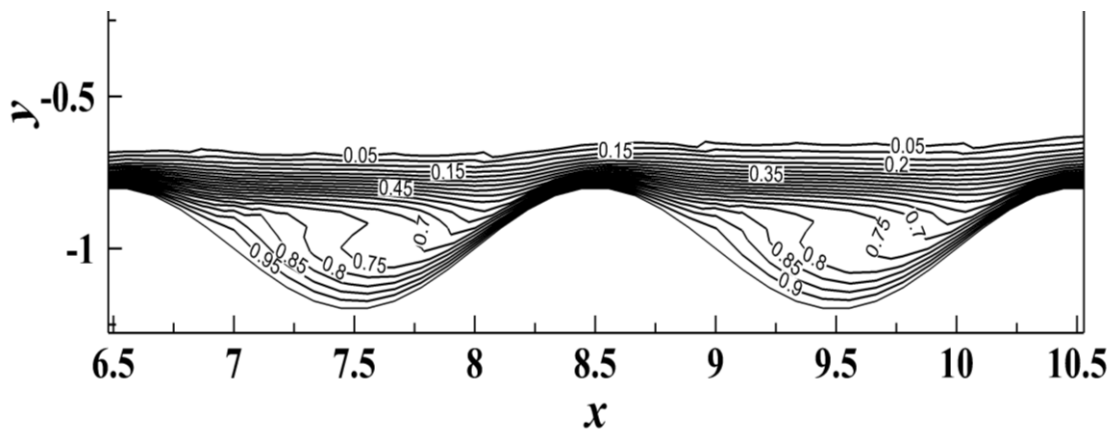


ج: $Re = 500$

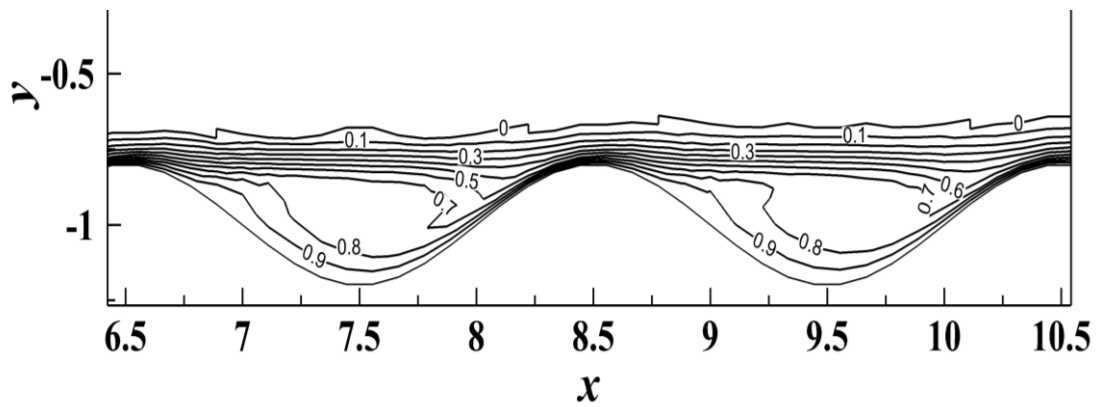
شکل ۴-۱۳: نمودار خط جریان در شرایط $a = 0.2$, $Pr = 6.93$, $\varphi = 0\%$



الف: $Re = 100$



ب: $Re = 300$



ج: $Re = 500$

شکل ۴-۱۴: نمودار خط هم‌دما در شرایط $a = 0.2$, $Pr = 6.93$, $\varphi = 0\%$

۴-۳-۴ بررسی تأثیر عدد پراتل

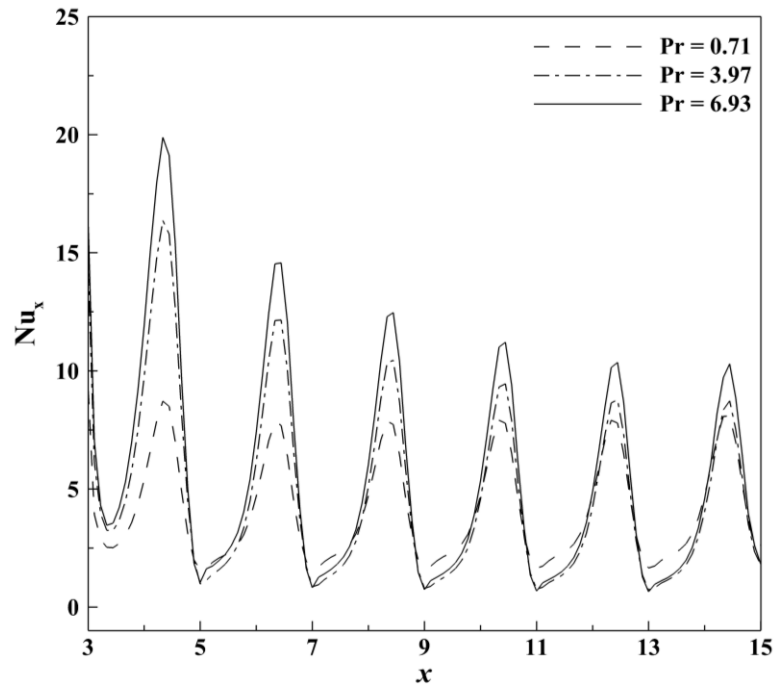
در این بخش اثر عدد پراتل روی جریان در شرایط $\varphi = 0\%$ ، $a = 0.2$ ، $Re = 300$ مورد بررسی قرار گرفته شده است. عدد پراتل بیانگر نسبت نفوذ اندازه حرکت (لزجت سینماتیکی) به نفوذ گرمایی است. رابطه‌ی جابه‌جایی درون کانال به شکل $Nu \approx Re^{0.5} Pr^{0.3}$ است. طبق این رابطه با افزایش عدد پراتل، عدد ناسلت محلی افزایش می‌یابد (شکل ۴-۱۵). با توجه به اینکه افزایش عدد پراتل تأثیری در افزایش و یا کاهش گرادیان فشار ندارد، اندازه‌ی گردابه‌ها در بخش موج‌دار با افزایش عدد پراتل تغییری نمی‌کند. از آنجایی‌که عدد ناسلت متوسط بیانگر توزیع عدد ناسلت محلی در امتداد کانال است، با افزایش عدد ناسلت محلی عدد ناسلت متوسط نیز افزایش می‌یابد (شکل ۴-۱۶).

مطابق شکل ۴-۱۷ که تغییرات عدد ناسلت متوسط با عدد رینولدز برای دامنه‌ی موج مختلف در شرایط $Pr = 0.71$ را نشان می‌دهد، عدد ناسلت متوسط با افزایش دامنه‌ی موج و عدد رینولدز افزایش می‌یابد. اختلاف عدد ناسلت متوسط در کانال موج‌دار با دامنه‌های کم ($a = 0.1$) با کانال صاف کم می‌باشد، اما در دامنه‌های موج بالاتر مخصوصاً در اعداد رینولدز بالا قابل توجه است. با افزایش عدد پراتل مقادیر عدد ناسلت متوسط افزایش می‌یابد.

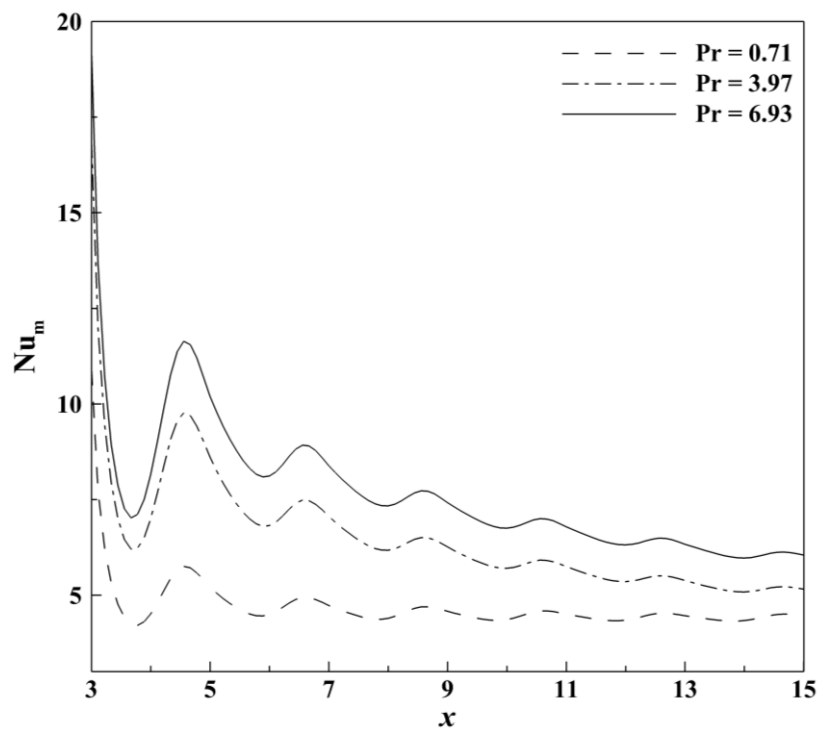
در شکل ۴-۱۸ خطوط هم‌دما برای اعداد پراتل مختلف در شرایط $\varphi = 0\%$ ، $a = 0.2$ ، $Re = 300$ نمایش داده شده است. مطابق این شکل هر چه عدد پراتل کمتر باشد نفوذ حرارت به بالادست جریان بیشتر است، به عبارتی هر چه عدد پراتل کمتر باشد عدد ناسلت کاهش می‌یابد. مطابق جدول ۴-۳ که تأثیر عدد پراتل روی عدد ناسلت متوسط را نشان می‌دهد، افزایش عدد پراتل سبب افزایش مقدار عدد ناسلت متوسط می‌گردد.

جدول ۴-۳: تأثیر عدد پراتل روی عدد ناسلت متوسط در شرایط $Re = 300$ ، $a = 0.2$ ، $\varphi = 0\%$

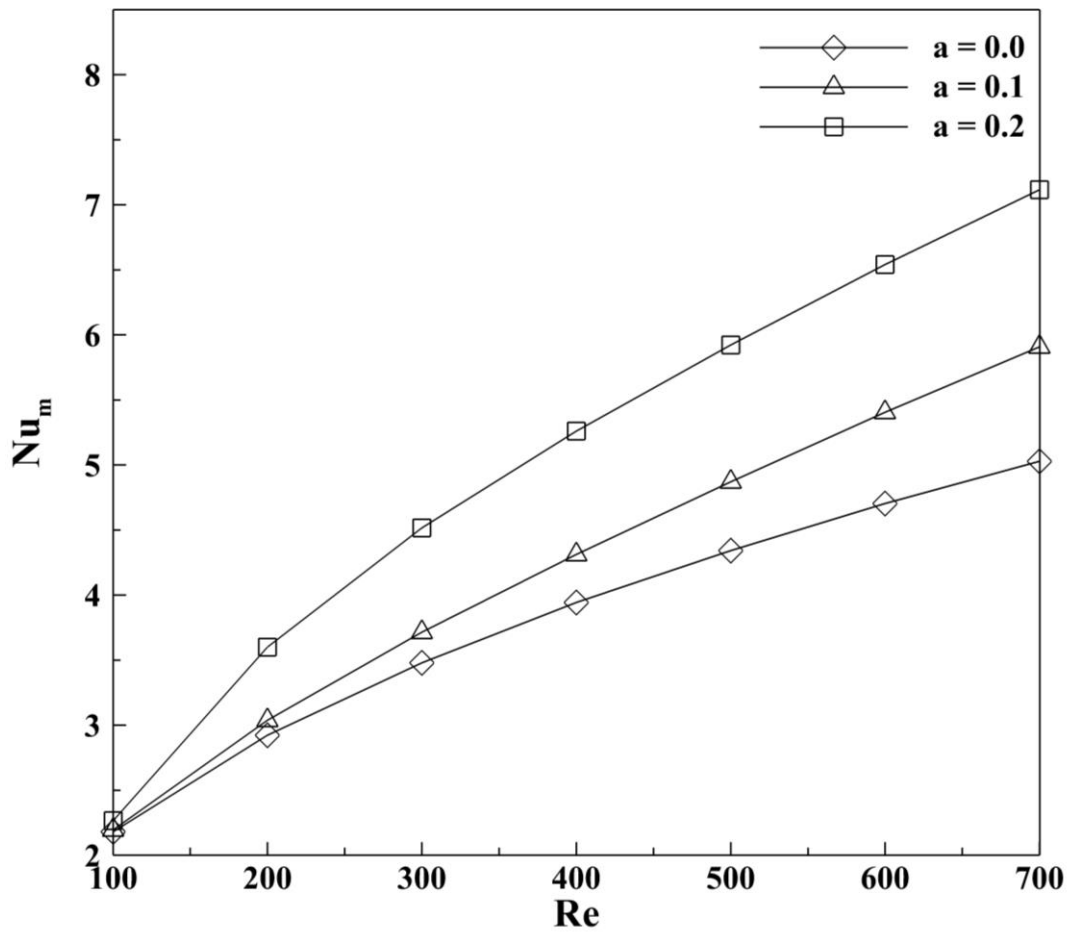
۶/۹۳	۳/۹۷	۰/۷۱	Pr
۶/۱۰۴۰۳	۵/۲۰۵۶۶	۴/۵۱۶۲۱	Nu_m



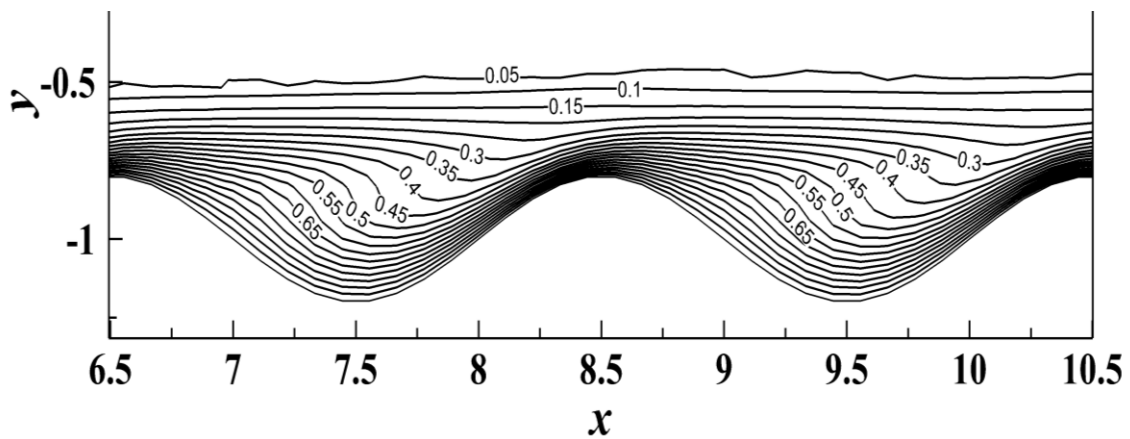
شکل ۴-۱۵: عدد ناسلت محلی در شرایط $Re = 300$, $a = 0.2$, $\varphi = 0\%$



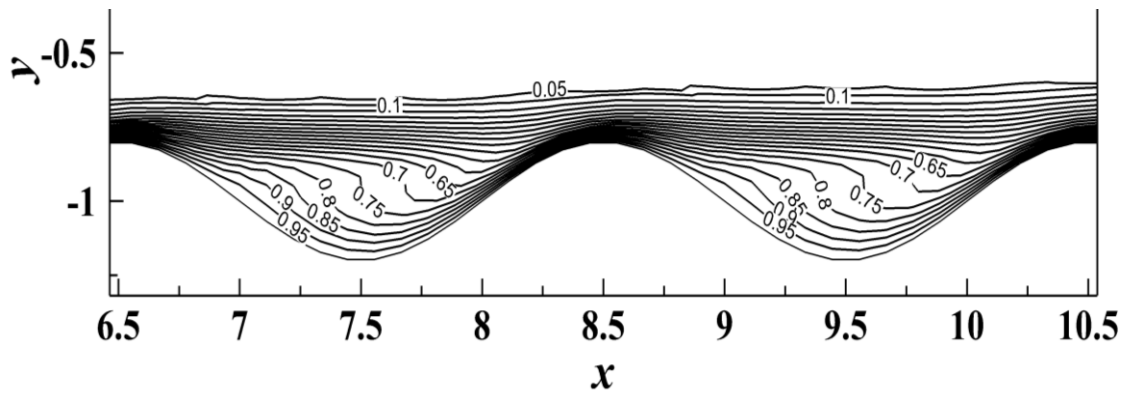
شکل ۴-۱۶: عدد ناسلت متوسط در شرایط $Re = 300$, $a = 0.2$, $\varphi = 0\%$



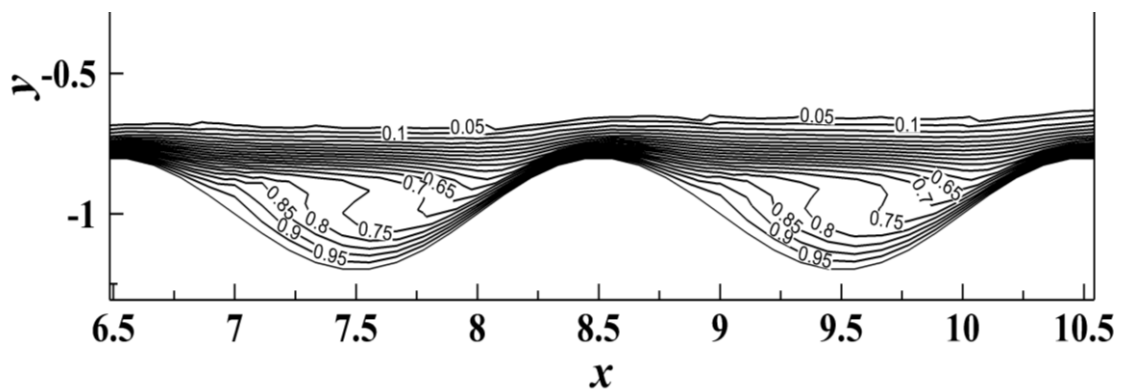
شکل ۴-۱۷: عدد ناسلت متوسط بر حسب رینولدز برای دامنه‌ی موج مختلف در شرایط $\rho = 0\%$ ، $Pr = 0.71$



الف: $Pr = 0.71$



ب: $Pr = 3.97$



ج: $Pr = 6.93$

شکل ۴-۱۸: نمودار خطوط هم‌دما در شرایط $Re = 300$, $a = 0.2$, $\varphi = 0\%$

۴-۴-۴ بررسی تأثیر کسر حجمی نانوسیال

در این بخش به بررسی تأثیر کسر حجمی نانوسیال آب-مس اکسید بر جریان در شرایط $a = 0.2$ ، $Re = 300$ ، $Pr = 6.93$ ، $d_p = 10 \text{ nm}$ پرداخته شده است. با افزایش کسر حجمی نانوسیال، ضریب اصطکاک پوسته‌ای محلی به اندازه‌ی ناچیزی کاهش می‌یابد (شکل ۴-۱۹). کاهش ضریب اصطکاک پوسته‌ای محلی به این دلیل است که اگرچه ویسکوزیته‌ی سیال (μ_{nf}) بیشتر از مایع خالص است اما از آنجایی که گرادین سرعت در دیواره به طور قابل توجهی کاهش می‌یابد پس اثر افزایش (μ_{nf}) خنثی می‌شود. افزودن نانوذرات به سیال پایه، باعث افزایش میزان انتقال حرارت می‌شود، هنگامی که مقدار کسر حجمی نانوذره افزایش می‌یابد مقادیر بیشینه‌ی عدد ناسلت نیز افزایش می‌یابد، چراکه ضریب

هدایت حرارتی مؤثر نانوسیال افزایش می‌یابد (شکل ۴-۲۰). به‌طور مشابه این تحلیل برای عدد ناسلت متوسط صادق است (شکل ۴-۲۱).

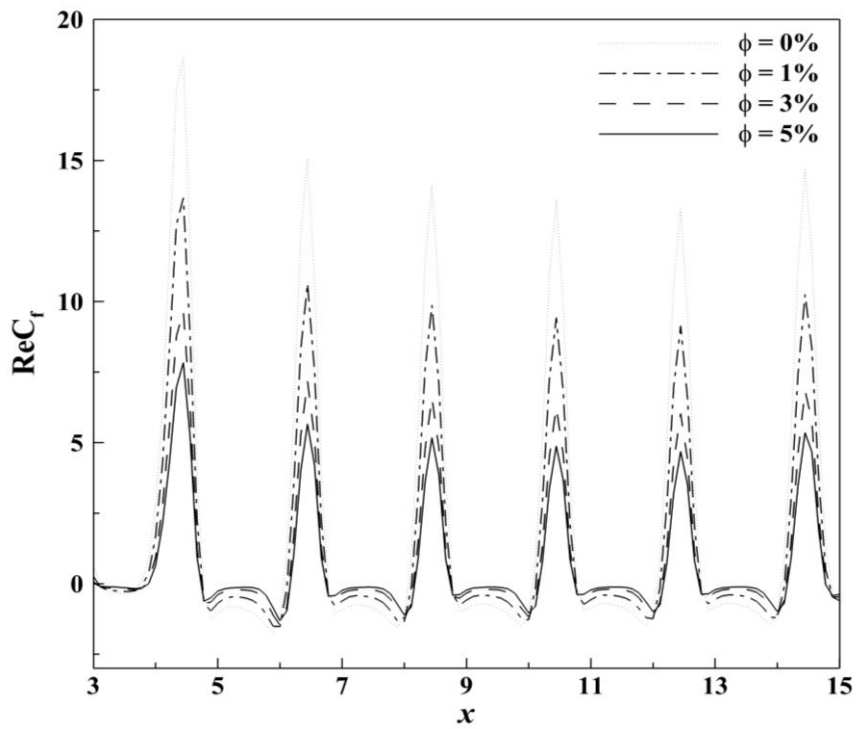
شکل ۴-۲۲ تغییرات عدد ناسلت متوسط را برای اعداد رینولدز در کسرهای حجمی مختلف نانوسیال در شرایط $a = 0.1$, $d_p = 10 \text{ nm}$, $Pr = 6.93$ نشان می‌دهد. همان‌طور که قبلاً بیان گردید افزایش عدد رینولدز باعث افزایش انتقال حرارت و افزایش عدد ناسلت روی دیواره می‌شود. عدد ناسلت متوسط با افزودن کسر حجمی نانوسیال افزایش می‌یابد.

در شکل ۴-۲۳ عدد ناسلت محلی برای کسر حجمی‌های مختلف نانوسیال آب-آلومینیوم اکسید در شرایط $d_p = 10 \text{ nm}$, $a = 0.2$, $Pr = 6.93$, $Re = 300$ نشان داده شده است. مطابق این شکل مقادیر بیشینه در نانوسیال آب-مس اکسید بیشتر از آب-آلومینیوم اکسید است.

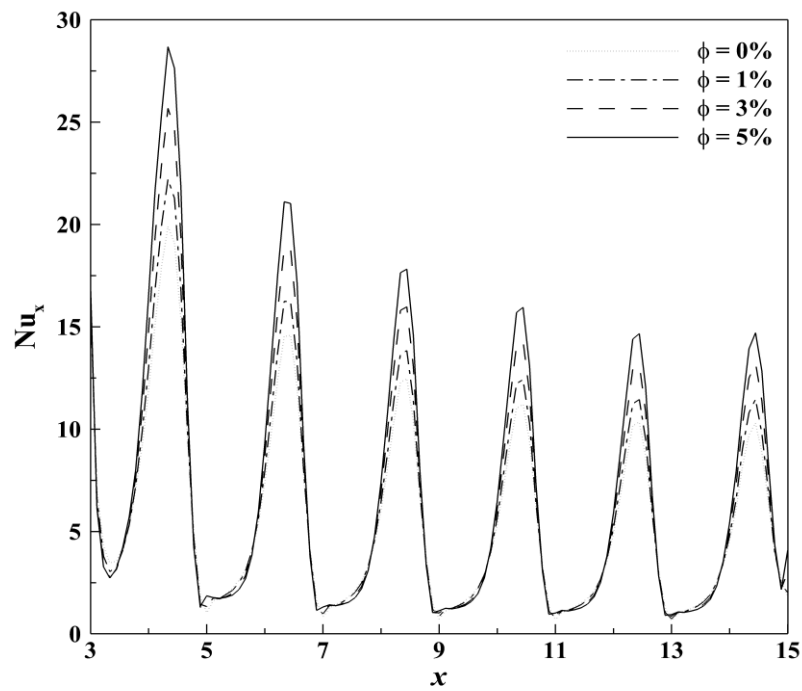
جدول ۴-۴ تأثیر کسر حجمی نانوسیال آب-مس اکسید را روی عدد ناسلت متوسط در شرایط $Re = 300$, $Pr = 6.93$, $a = 0.2$ نشان می‌دهد. طبق این جدول افزایش کسر حجمی نانوذره سبب افزایش مقدار عدد ناسلت متوسط می‌گردد.

جدول ۴-۴: تأثیر کسر حجمی نانوسیال آب-مس اکسید روی عدد ناسلت متوسط در شرایط $a = 0.2$, $Re = 300$, $Pr = 6.93$, $d_p = 10 \text{ nm}$

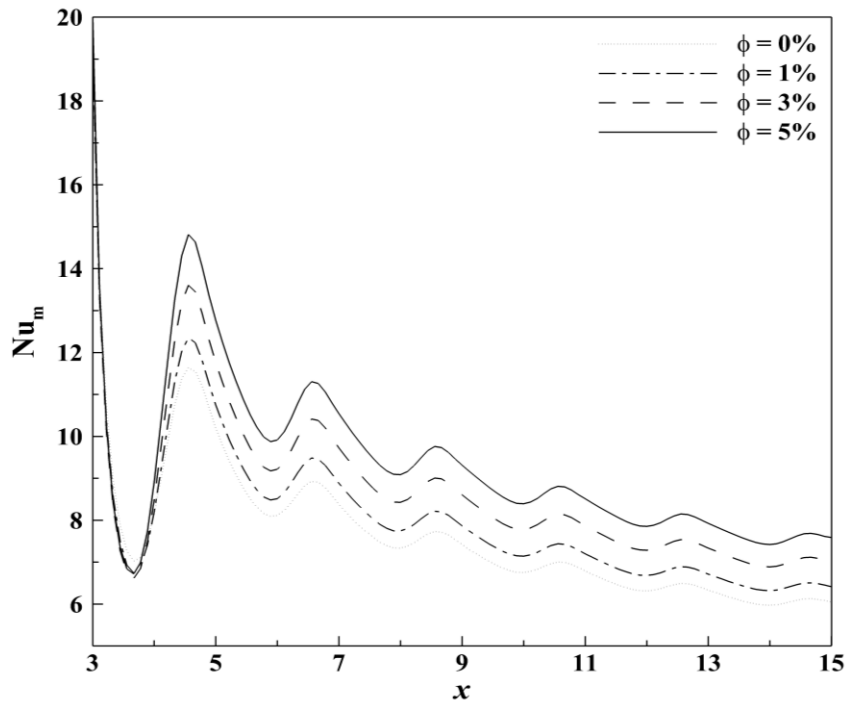
φ	٪۱	٪۳	٪۵
Nu_m	۶/۷۰۴۹۱	۶/۹۵۰۴۷	۷/۱۸۸۳۰



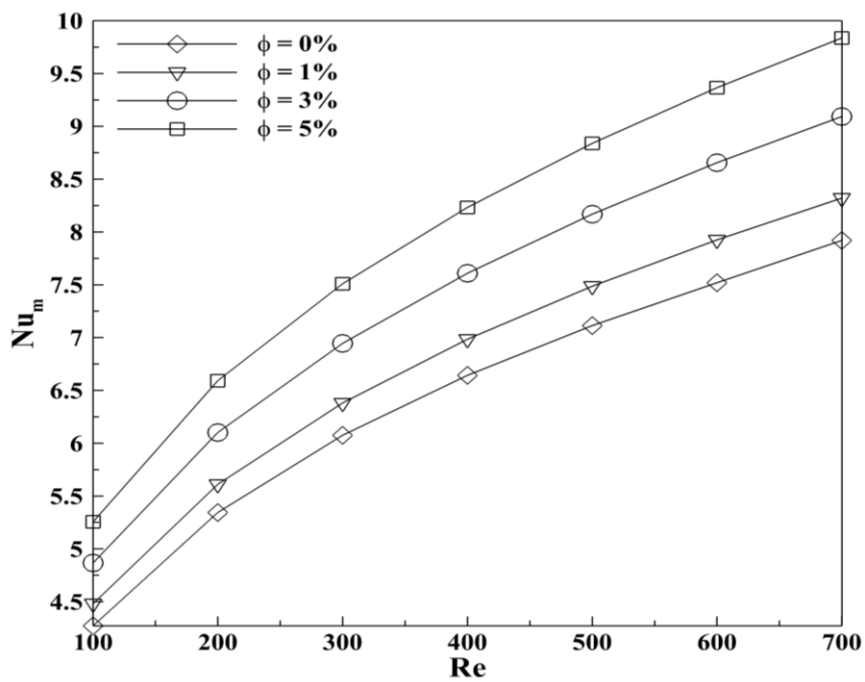
شکل ۴-۱۹: ضریب اصطکاک پوسته‌ای در شرایط $Re = 300$, $Pr = 6.93$, $d_p = 10 \text{ nm}$, $a = 0.2$



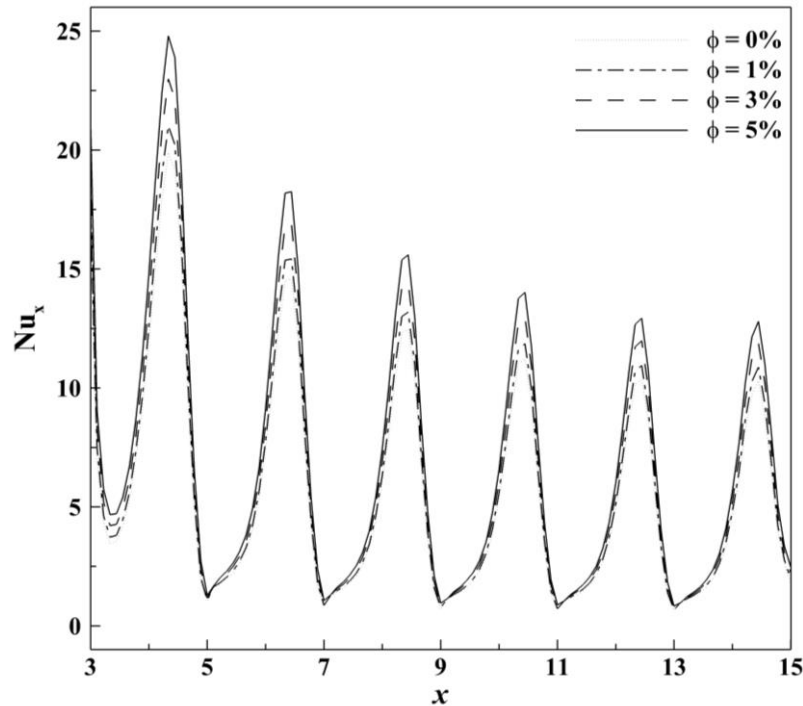
شکل ۴-۲۰: عدد ناسلت محلی در شرایط $Re = 300$, $Pr = 6.93$, $d_p = 10 \text{ nm}$, $a = 0.2$



شکل ۴-۲۱: عدد ناسلت متوسط در شرایط $Re = 300$, $Pr = 6.93$, $d_p = 10 \text{ nm}$, $a = 0.2$



شکل ۴-۲۲: تغییرات عدد ناسلت متوسط برحسب عدد رینولدز در شرایط $d_p = 10 \text{ nm}$, $a = 0.1$, $Pr = 6.93$



شکل ۴-۲۳: عدد ناسلت محلی برای نانوذره ی Al_2O_3 در شرایط $a = 0.2$, $d_p = 10\text{ nm}$, $Pr = 6.93$, $Re = 300$

۵-۴-۴ بررسی تأثیر قطر نانوذره

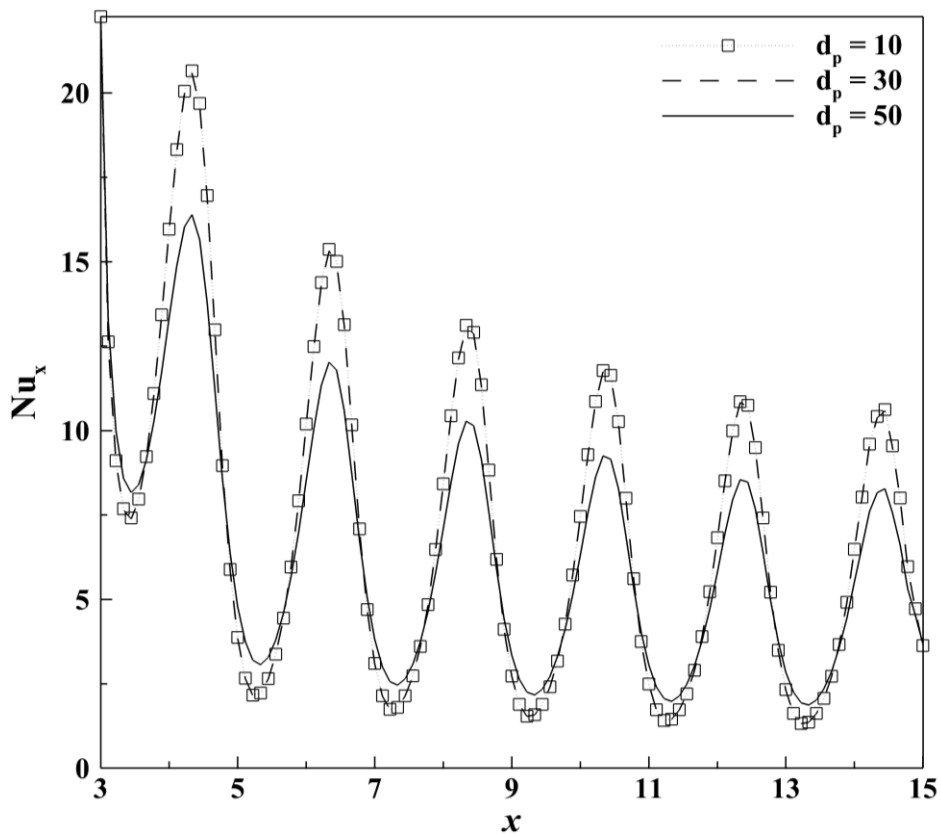
در این بخش به بررسی اثر قطر نانوذره بر روی جریان در شرایط $a = 0.1$, $Pr = 6.93$, $\phi = 5\%$ و $Re = 300$ پرداخته شده است. برای این منظور دو نانوسیال مس و آلومینیوم اکسید در نظر گرفته شده است. مطابق شکل ۴-۲۴، مشاهده می شود که با کوچک تر شدن ذرات نه تنها نسبت سطح به حجم ذرات بیشتر شده، بلکه پایداری مخلوط های نانوسیال نیز بیشتر می شود، همچنین حرکت بیرونی نانوذرات به سبب جرم کمترشان بیشتر بوده و باعث افزایش ضریب رسانایی می شود؛ اما این تأثیر بسیار ناچیز است. در شکل ۴-۲۵ تأثیر قطر نانوذره ی آلومینیوم اکسید روی عدد ناسلت محلی بررسی شده است. همان طور که در شکل مشاهده می شود افزایش قطر نانوذره باعث کاهش عدد ناسلت محلی می شود، اما تأثیر قطر نانوذره در افزایش عدد ناسلت بسیار کم است.

در جدول ۴-۵ تأثیر قطر نانوذره ی مس روی عدد ناسلت متوسط نشان داده شده است. طبق جدول

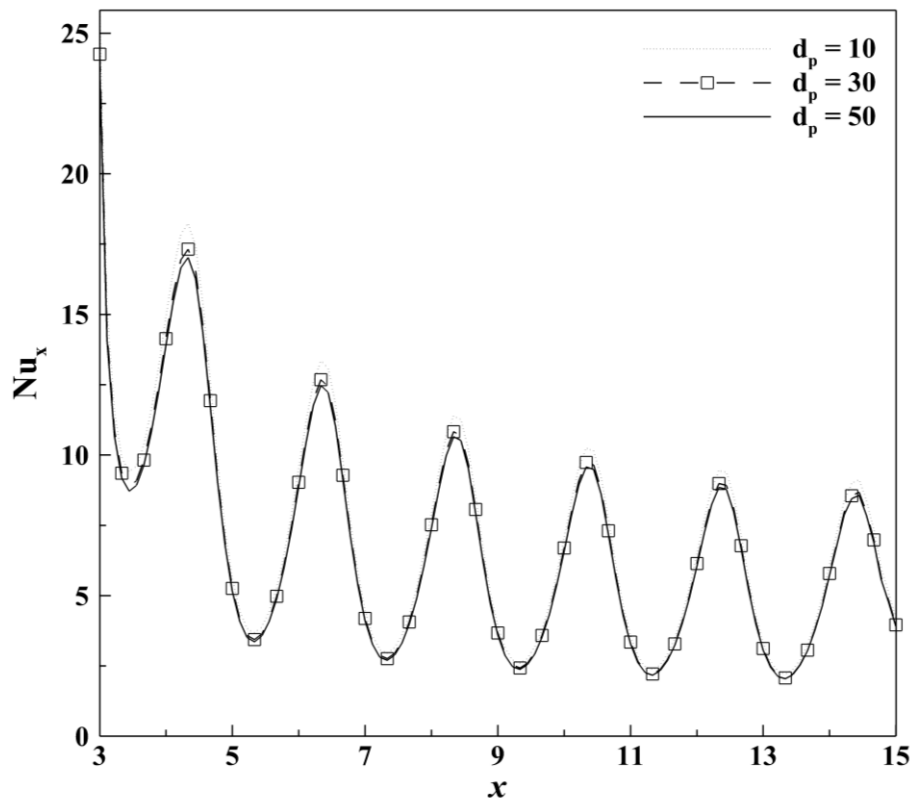
با افزایش قطر نانوسیال عدد ناسلت متوسط کاهش می یابد.

جدول ۴-۵: تأثیر قطر نانوذره‌ی مس اکسید روی عدد ناسلت متوسط در شرایط $\text{Pr} = 6.93, \phi = 5\%$ ،
 $\text{Re} = 300$ و $a = 0.1$

d_p	۱۰	۳۰	۵۰
Nu_m	۷/۶۵۷۵۷	۷/۶۲۹۷۱	۷/۱۸۸۳۰



شکل ۴-۲۴: تغییرات عدد ناسلت محلی برای نانوذره‌ی مس اکسید در شرایط $\text{Pr} = 6.93, \phi = 5\%$ ، $a = 0.1$ و $\text{Re} = 300$



شکل ۴-۲۵: تغییرات عدد ناسلت محلی برای نانوذره‌ی آلومینیوم اکسید در شرایط $Pr = 6.93, \phi = 5\%$ ، $Re = 300$ و $a = 0.1$

۴-۵ مقایسه‌ی عملکرد پردازنده‌ی مرکزی و گرافیکی

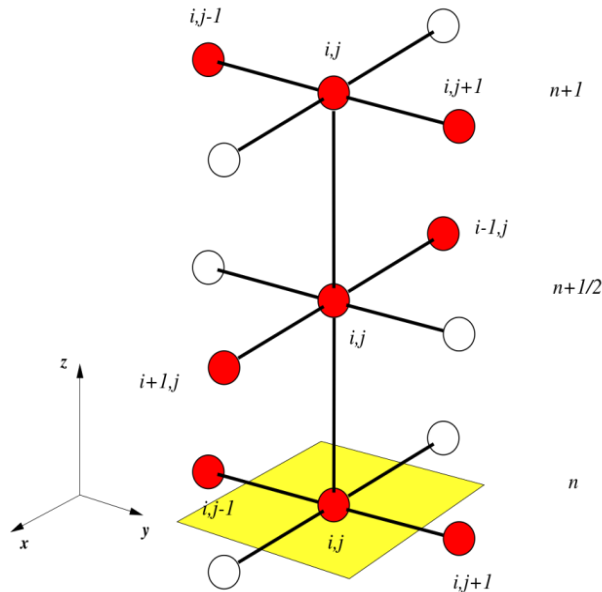
پردازنده‌ی گرافیکی مورد استفاده در این تحقیق، کارت گرافیک GeForce GTX 970 با معماری ماکسول و با قابلیت محاسباتی ۵/۲ است. این پردازنده دارای ۱۶۶۴ هسته‌ی پردازشی است و دارای ۴ گیگابایت حافظه‌ی سراسری است. پردازنده‌ی مرکزی مورد استفاده نیز Intel Core i7 - 5930K است. این نکته قابل ذکر است که برای مقایسه از یک کارت گرافیک و پردازنده‌ی مرکزی هم‌نسل و هم‌رده استفاده شده تا مقایسه صحیح و عادلانه باشد. همچنین از نسخه‌ی ۱۰/۱ میان‌افزار کودا برای برنامه‌نویسی روی پردازنده‌ی گرافیکی استفاده شده است. در برنامه‌نویسی با میان‌افزار کودا اندازه‌ی بلوک و ابعاد آن قبل از اجرای برنامه باید مشخص باشد، همچنین تعداد بلوک‌های درون یک شبکه در هر راستا از ۶۵۵۳۵، و تعداد نخ‌های محاسباتی درون یک بلوک نیز در هر راستا از ۱۰۲۴ نمی‌تواند

بیشتر باشد. در الگوریتم توماس موازی نمی‌توان از حافظه‌ی مشترک در مسائل با ابعاد بسیار بزرگ استفاده کرد به این دلیل که حافظه‌ی مشترک دارای این خصیصه است که فقط نخ‌های محاسباتی همان بلوک به آن دسترسی دارند و اگر اندازه‌ی هر دستگاه معادله از ۱۰۲۴ بیشتر شود به‌ناچار تعداد بلوک‌ها، حداقل دو بلوک خواهند بود و نخ‌های محاسباتی یک بلوک به مقادیر بلوک دیگر دسترسی ندارند. این موضوع که نیمی از متغیرها به مقدار عددی نیم دیگر دسترسی نداشته باشند در الگوریتم توماس پذیرفته نیست. اما برای مسائلی که اندازه‌ی هر دستگاه مستقل کمتر از ۱۰۲۴ باشد استفاده از حافظه‌ی مشترک برای ذخیره‌سازی متغیرها امکان‌پذیر است.

در مرجع [۴۰] به این مطلب اشاره شده است که، اگر استفاده از پردازنده‌ی گرافیکی فقط برای شتاب‌دهی به بخشی از محاسبات استفاده شده، باید زمان انتقال اطلاعات از حافظه‌ی میزبان به حافظه‌ی دستگاه را محاسبه نمود و بعد در مورد میزان مؤثر بودن استفاده از پردازش موازی قضاوت کرد. اما اگر تمام مراحل حل از ابتدا روی پردازنده‌ی گرافیکی انجام شده باشند، در انتها برای مشاهده‌ی نتایج باید اطلاعات را از دستگاه به میزبان انتقال داد، در این صورت نیازی به در نظر گرفتن زمان انتقال نیست. در این تحقیق نیز تمام مراحل حل روی دستگاه صورت گرفته و بنابراین در پایان مراحل حل، زمان انتقال متغیرهایی نظیر سرعت، تاوایی و غیره از دستگاه به میزبان در نظر گرفته نشده است.

همان‌طور که گفته شد برای گسسته‌سازی حالت ضمنی و برای حل دستگاه معادلات روش ضمنی اسپلاین با جهت متغیر (SADI) در نظر گرفته می‌شود. روش ضمنی جهت متغیر با توجه به تعداد بعد، مسأله و گام زمانی را به همان تعداد تقسیم می‌کند، یعنی اگر مسأله دوبعدی باشد معادلات به دو بخش تقسیم می‌شوند و کل دامنه‌ی محاسباتی در یک راستا (برای مثال در راستای x) به شکل سه‌قطری یک‌بار حل می‌شوند و در واقع حل، نصف یک گام زمانی پیشروی داشته و در مرحله‌ی دوم با حل معادلات در کل دامنه‌ی محاسباتی به شکل سه‌قطری در راستای y نصف دیگر یک گام زمانی

طی می‌شود (شکل ۴-۲۶) و با انجام این دو مرحله در واقع حل، یک گام زمانی به جلو رفته است. خاصیت و مزیت این روش در سه‌قطری بودن معادلات است که توسط الگوریتم توماس که یک الگوریتم سریع و بهینه است می‌توانند حل شوند و در نتیجه یک سرعت خوب در حل ایجاد می‌شود.



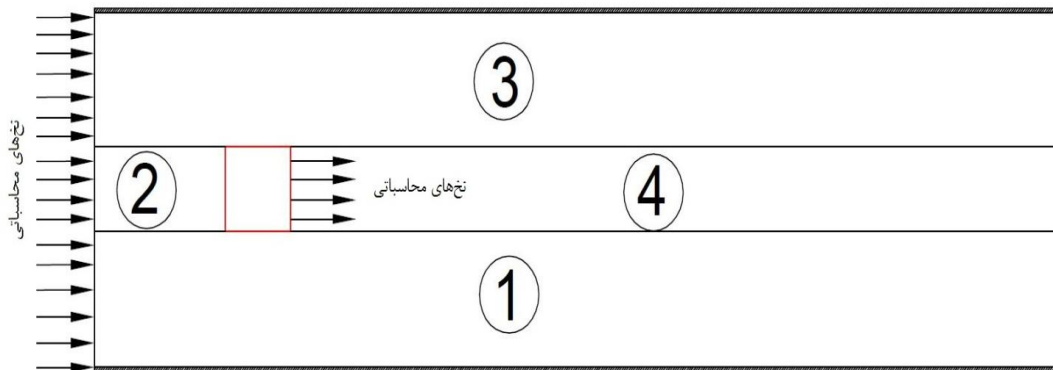
شکل ۴-۲۶: نمایش مراحل روش ضمنی اسپلاین جهت متغیر

معادلات (۴-۱) و (۴-۲) که در فصل سوم به دست آمده‌اند را در نظر بگیرید استراتژی حل به این شکل است که برای اجرای مرحله‌ی اول و حل معادله‌ای مانند (۴-۱) ناحیه‌ی محاسباتی به چهار بخش تقسیم می‌شود و هر سطر یک دستگاه معادله‌ی سه‌قطری است که با الگوریتم توماس حل می‌شوند. در پردازنده‌ی مرکزی هر سطر شکل ۴-۲۷ به ترتیب حل و پردازش می‌شوند. برای پیاده‌سازی و حل روی پردازنده‌ی گرافیکی آرایش نخ‌های محاسباتی به شکل ۴-۲۷ درمی‌آیند. هر نخ محاسباتی یک سطر محاسباتی را برای پردازش در دست می‌گیرد و هر نخ یک دستگاه معادله‌ی سه‌قطری را با الگوریتم توماس حل می‌کند. در مرحله‌ی دوم حل باید در راستای y و طبق معادله‌ی (۴-۲) دستگاه معادلات حل شوند تا یک گام زمانی کامل شود. در این مرحله نخ‌ها در پردازنده‌ی گرافیکی آرایشی

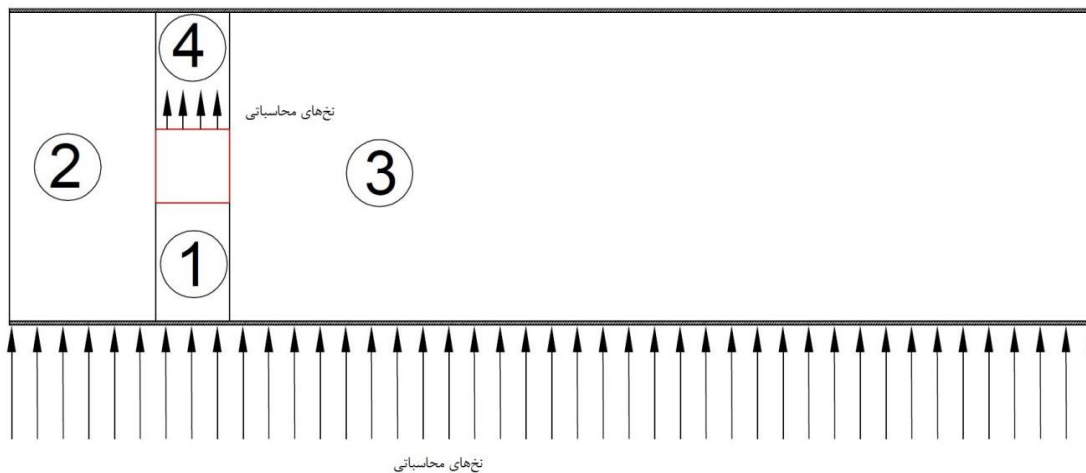
همچون شکل ۲۸-۴ پیدا می‌کنند. در مرحله‌ی دوم دسترسی به حافظه‌ی سراسری و متغیرها از نوع هم‌مکان خواهد بود که یک حالت ایده‌آل برای پردازش موازی با پردازنده‌ی گرافیکی است.

$$A_{i,j}\phi_{i-1,j}^{n+\frac{1}{2}} + B_{i,j}\phi_{i,j}^{n+\frac{1}{2}} + C_{i,j}\phi_{i+1,j}^{n+\frac{1}{2}} = D_{i,j}^n \quad (۱-۴)$$

$$A_{i,j}\phi_{i,j-1}^{n+1} + B_{i,j}\phi_{i,j}^{n+1} + C_{i,j}\phi_{i,j+1}^{n+1} = D_{i,j}^{n+\frac{1}{2}} \quad (۲-۴)$$



شکل ۲۷-۴: آرایش نخهای محاسباتی در مرحله‌ی اول اعمال روش ضمنی اسپلین جهت متغیر



شکل ۲۸-۴: آرایش نخهای محاسباتی در مرحله‌ی دوم اعمال روش ضمنی اسپلین جهت متغیر

ساختار برنامه‌ی نوشته‌شده برای پردازنده‌ی گرافیکی به ترتیب زیر است:

(۱) اختصاص مقدار اولیه به پارامترهای فیزیکی مانند سرعت و تاوایی و دما در سرتاسر فضای محاسباتی.

(۲) تعیین و اعمال شرطهای مرزی دما، تاوایی، تابع جریان و سرعت.

(۳) حل معادله‌ی گسسته شده‌ی تاوایی در راستای x به صورت ضمنی و با رهیافت توماس موازی و انجام یک زیر گام زمانی به اندازه‌ی $n/2$

(۴) حل معادله‌ی گسسته شده‌ی تاوایی در راستای y به صورت ضمنی و با رهیافت توماس موازی انجام یک زیر گام زمانی دیگر به اندازه‌ی $n/2$

(۵) حل معادله‌ی گسسته شده‌ی تابع جریان در راستای x به صورت ضمنی و با رهیافت توماس موازی و انجام یک زیر گام زمانی به اندازه‌ی $n/2$

(۶) حل معادله‌ی گسسته شده‌ی تابع جریان در راستای y به صورت ضمنی و با رهیافت توماس موازی انجام یک زیر گام زمانی دیگر به اندازه‌ی $n/2$

(۷) محاسبه‌ی مقادیر سرعت در تمام فضای محاسباتی

(۸) رفتن به مرحله‌ی دوم و تکرار این حلقه تا رسیدن به حل مورد نظر.

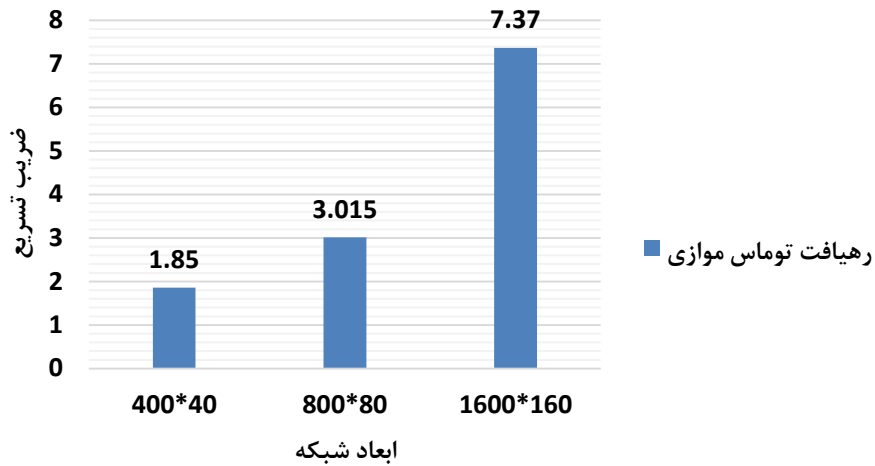
(۹) حل معادله‌ی گسسته شده‌ی دما در راستای x به صورت ضمنی و با رهیافت توماس موازی و انجام یک زیر گام زمانی به اندازه‌ی $n/2$

(۱۰) حل معادله‌ی گسسته شده‌ی دما در راستای y به صورت ضمنی و با رهیافت توماس موازی انجام یک زیر گام زمانی دیگر به اندازه‌ی $n/2$

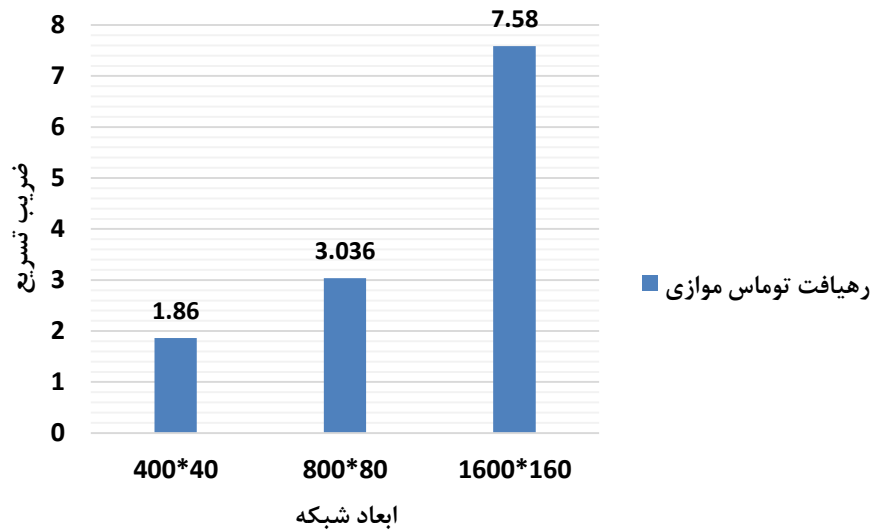
(۱۱) رفتن به مرحله‌ی دوم و تکرار این حلقه تا رسیدن به حل مورد نظر.

با توجه به سبک برنامه‌نویسی در پردازنده‌های گرافیکی هر کدام از بخش‌های نامبرده شده باید توسط یک کرنل به سرانجام برسند. این امر بدیهی است که هر چه نخ‌های محاسباتی بیشتری را بتوان به کار گرفت اصولاً سرعت پردازش بیشتری به دست خواهد آمد. در ادامه زمان پردازش برای شبکه‌های

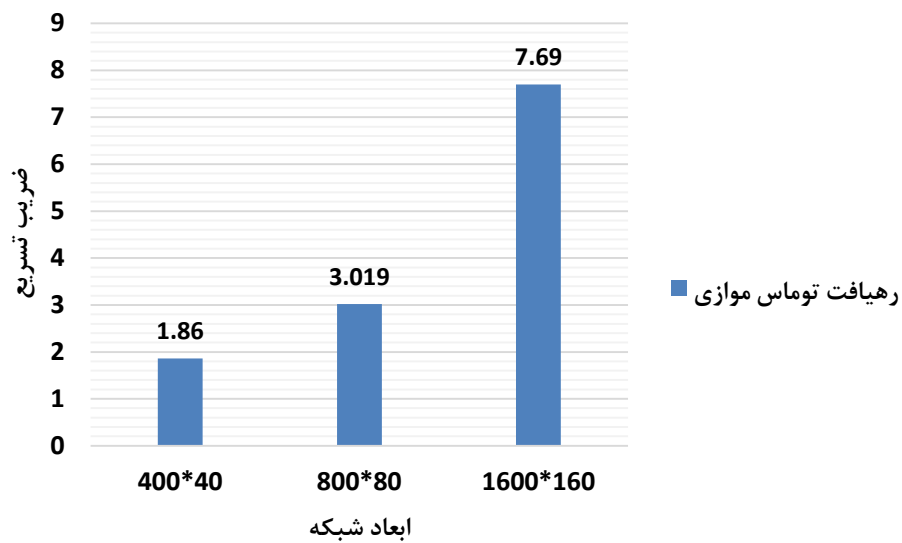
محاسباتی با ابعاد مختلف برای نانوسیال آب-مس اکسید با دامنه‌ی موج و کسر حجمی متفاوت روی پردازنده‌ی گرافیکی اندازه‌گیری شده و با زمان حل روی پردازنده‌ی مرکزی مقایسه شده‌اند (شکل ۲۹-۴ - شکل ۳۴-۴).



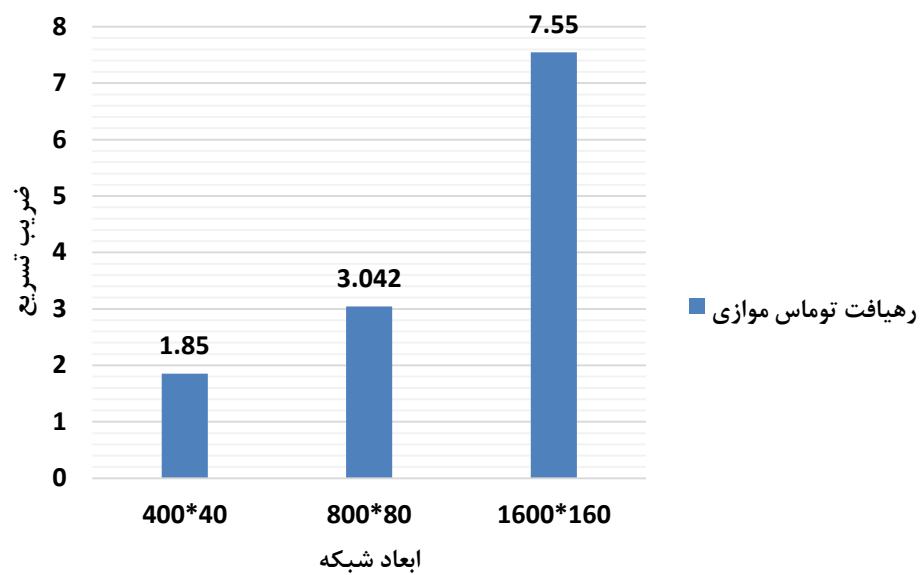
شکل ۲۹-۴: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $a = 0.1$ ، $Pr = 6.93$ ، $\varphi = 0\%$ و $Re = 300$



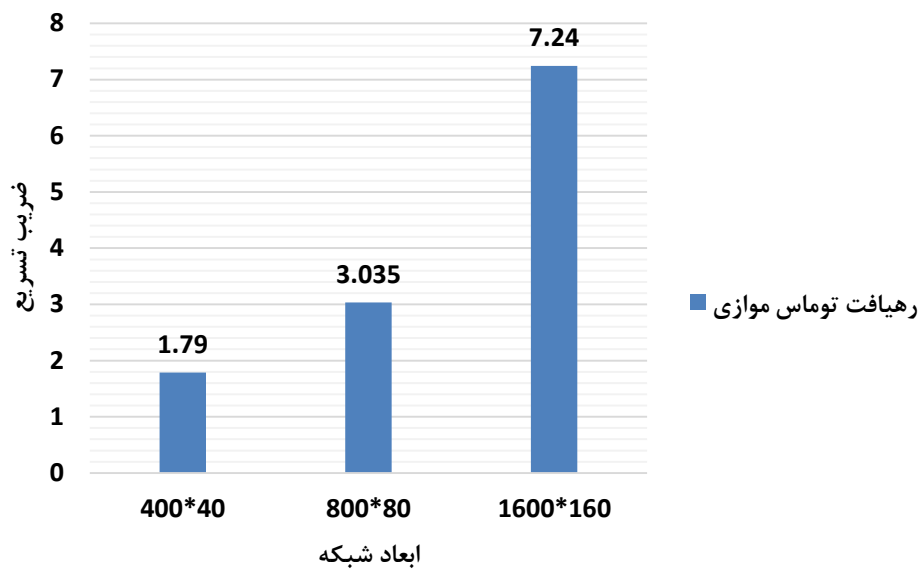
شکل ۳۰-۴: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $a = 0.1$ ، $Pr = 6.93$ ، $\varphi = 1\%$ و $Re = 300$



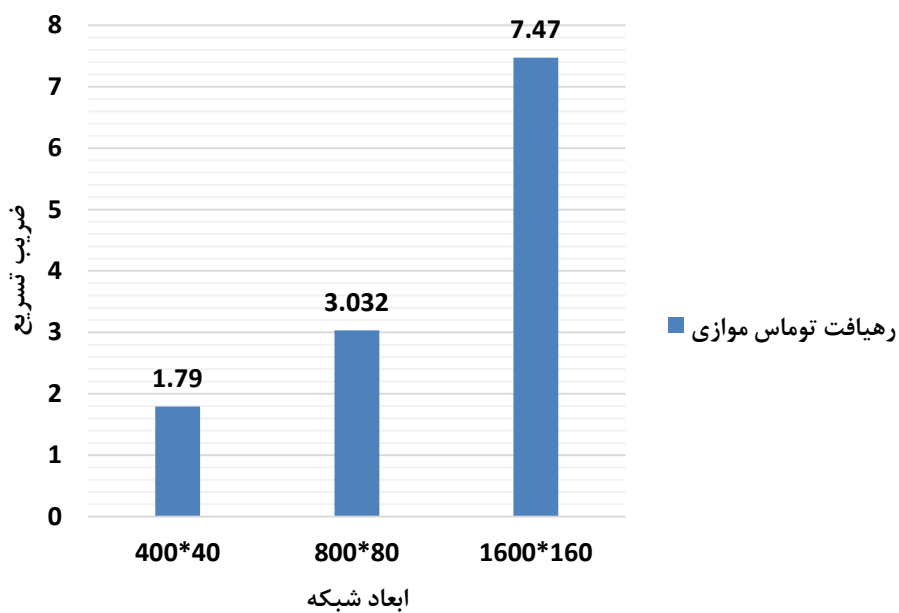
شکل ۴-۳۱: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $a = 0.1$ ، $Pr = 6.93$ ، $\varphi = 3\%$ و $Re = 300$



شکل ۴-۳۲: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $Pr = 6.93$ ، $\varphi = 0\%$ و $a = 0.2$ و $Re = 300$



شکل ۳۳-۴: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $\varphi = 1\%$, $Pr = 6.93$, $a = 0.2$ و $Re = 300$



شکل ۳۴-۴: افزایش سرعت نسبت به زمان حل پردازنده‌ی مرکزی در شرایط $\varphi = 3\%$, $Pr = 6.93$, $a = 0.2$ و $Re = 300$

در نهایت برای داشتن دید بهتر نسبت به آنچه در پردازنده‌ی گرافیکی رخ می‌دهد در جدول ۴-۶ سهم هر کرنل یا تابع نوشته‌شده در یکبار اجرای تمام برنامه روی پردازنده‌ی گرافیکی، ابعاد شبکه و ابعاد بلوک نمایش داده‌شده است. به‌طور مشخص تابعی که در جهت محور طولی دستگاه‌های سه‌قطری را حل می‌کند، بیشترین سهم و بیشترین زمان حل را به خود اختصاص می‌دهد زیرا با تعداد نخ نسبتاً کم باید دستگاه‌هایی با اندازه‌ی بزرگ‌تر نسبت به تابع دیگر که در راستای محور عرضی عمل می‌کند، را حل و پردازش کند. اما مشخصاً این داده‌ها برای یکبار اجرای توابع حاصل‌شده‌اند و در روند حل تعداد تکرار این توابع متفاوت خواهند بود.

جدول ۴-۶: توابع اجراشده روی پردازنده‌ی گرافیکی

Function Name	Grid Dimensions	Block Dimensions	Duration (μ s)	Occupancy
XTDMAmS	{1, 1, 1}	{32, 1, 1}	1077.274	0.25
XTDMAmT	{1, 1, 1}	{32, 1, 1}	1025.622	0.25
TETA_XTDMA	{1, 1, 1}	{32, 1, 1}	1012.725	0.25
XTDMAmW	{1, 1, 1}	{32, 1, 1}	1007.029	0.25
OMEGA_XTDMA	{1, 1, 1}	{32, 1, 1}	1003.156	0.25
PSI_XTDMA	{1, 1, 1}	{32, 1, 1}	1002.484	0.25
XTDMAmW	{1, 1, 1}	{32, 1, 1}	912.653	0.25
XTDMAmS	{1, 1, 1}	{32, 1, 1}	910.956	0.25
XTDMAmW	{1, 1, 1}	{32, 1, 1}	910.093	0.25
XTDMAmS	{1, 1, 1}	{32, 1, 1}	909.805	0.25
XTDMAmT	{1, 1, 1}	{32, 1, 1}	907.276	0.25
XTDMAmT	{1, 1, 1}	{32, 1, 1}	906.444	0.25
meshMapping	{12, 2, 1}	{16, 16, 1}	167.789	0.625
updateMT	{12, 2, 1}	{16, 16, 1}	116.714	0.75
Initialization	{13, 3, 1}	{16, 16, 1}	108.489	0.75
TETA_SADI_Y_FILLER	{12, 2, 1}	{16, 16, 1}	91.784	0.75
OMEGA_SADI_Y_FILLER	{12, 2, 1}	{16, 16, 1}	91.528	0.75
PSI_SADI_Y_FILLER	{12, 2, 1}	{16, 16, 1}	91.528	0.75
PSI_SADI_X_FILLER	{12, 2, 1}	{16, 16, 1}	90.6	0.75
TETA_SADI_X_FILLER	{12, 2, 1}	{16, 16, 1}	86.023	0.75
OMEGA_SADI_X_FILLER	{12, 2, 1}	{16, 16, 1}	82.374	0.75
PSI_SADI_X_FILLER	{12, 2, 1}	{16, 16, 1}	71.142	0.625
YTDMAIT	{6, 1, 1}	{32, 1, 1}	70.598	0.25
OMEGA_YTDMA	{6, 1, 1}	{32, 1, 1}	68.389	0.25
PSI_YTDMA	{6, 1, 1}	{32, 1, 1}	67.59	0.25

calculatemW	{12, 2, 1}	{16, 16, 1}	67.429	0.75
calculatemS	{12, 2, 1}	{16, 16, 1}	66.822	0.75
calculatemT	{12, 2, 1}	{16, 16, 1}	66.822	0.75
TETA_YTDMA	{6, 1, 1}	{32, 1, 1}	66.79	0.25
TETA_TD_X_FILLER	{12, 2, 1}	{16, 16, 1}	60.773	0.625
calculatemS	{12, 2, 1}	{16, 16, 1}	58.533	0.75
calculatelT	{12, 2, 1}	{16, 16, 1}	58.245	0.75
calculatemT	{12, 2, 1}	{16, 16, 1}	57.508	0.75
calculatemS	{12, 2, 1}	{16, 16, 1}	57.285	0.75
PSI_TD_Y_FILLER	{12, 2, 1}	{16, 16, 1}	57.125	0.625
calculatemW	{12, 2, 1}	{16, 16, 1}	56.709	0.75
YTDMAIS	{6, 1, 1}	{32, 1, 1}	56.228	0.25
OMEGA_TD_Y_FILLER	{12, 2, 1}	{16, 16, 1}	55.973	0.625
YTDMAIS	{6, 1, 1}	{32, 1, 1}	55.813	0.25
YTDMAIT	{6, 1, 1}	{32, 1, 1}	55.78	0.25
YTDMAIW	{6, 1, 1}	{32, 1, 1}	55.621	0.25
calculatemW	{12, 2, 1}	{16, 16, 1}	55.397	0.75
YTDMAIW	{6, 1, 1}	{32, 1, 1}	55.301	0.25
TETA_TD_Y_FILLER	{12, 2, 1}	{16, 16, 1}	55.301	0.625
YTDMAIW	{6, 1, 1}	{32, 1, 1}	55.141	0.25
YTDMAIS	{6, 1, 1}	{32, 1, 1}	55.141	0.25
YTDMAIT	{6, 1, 1}	{32, 1, 1}	55.141	0.25
BoundrayConditionmT	{6, 1, 1}	{32, 1, 1}	53.668	0.25
calculatemT	{12, 2, 1}	{16, 16, 1}	53.604	0.75
BoundrayConditionmT	{6, 1, 1}	{32, 1, 1}	53.253	0.25
BoundrayConditionmT	{6, 1, 1}	{32, 1, 1}	52.804	0.25
calculatelS	{12, 2, 1}	{16, 16, 1}	51.333	0.75
calculatelT	{12, 2, 1}	{16, 16, 1}	50.468	0.75
OMEGA_TD_X_FILLER	{12, 2, 1}	{16, 16, 1}	50.02	0.625
calculatelT	{12, 2, 1}	{16, 16, 1}	49.956	0.75
calculatelS	{12, 2, 1}	{16, 16, 1}	49.86	0.75
calculatelW	{12, 2, 1}	{16, 16, 1}	49.829	0.75
calculatelW	{12, 2, 1}	{16, 16, 1}	49.7	0.75
calculatelS	{12, 2, 1}	{16, 16, 1}	49.572	0.75
calculatelW	{12, 2, 1}	{16, 16, 1}	49.156	0.75
TETA_SADI_X	{12, 2, 1}	{16, 16, 1}	42.82	1
TETA_SADI_Y	{12, 2, 1}	{16, 16, 1}	34.019	1
OMEGA_SADI_Y	{12, 2, 1}	{16, 16, 1}	33.571	1
OMEGA_SADI_X	{12, 2, 1}	{16, 16, 1}	31.842	1
PSI_SADI_X	{12, 2, 1}	{16, 16, 1}	27.907	1
updateMW	{12, 2, 1}	{16, 16, 1}	25.346	1
PSI_SADI_Y	{12, 2, 1}	{16, 16, 1}	22.594	1

calculateLT	{12, 2, 1}	{16, 16, 1}	20.898	1
parameterNanoFluid	{1, 1, 1}	{1, 1, 1}	18.913	0.25
updateLT	{12, 2, 1}	{16, 16, 1}	15.265	1
PSIBoundaryCondition	{6, 1, 1}	{32, 1, 1}	15.201	0.25
updateLW	{12, 2, 1}	{16, 16, 1}	15.137	1
velocity	{12, 2, 1}	{16, 16, 1}	15.073	1
updateLS	{12, 2, 1}	{16, 16, 1}	14.497	1
updateBCTETA	{6, 1, 1}	{32, 1, 1}	13.761	0.25
calculateLS	{12, 2, 1}	{16, 16, 1}	13.409	1
updateMS	{12, 2, 1}	{16, 16, 1}	13.153	1
calculateLW	{12, 2, 1}	{16, 16, 1}	12.706	1
updateBCOMEGA	{6, 1, 1}	{32, 1, 1}	11.937	0.25
calculateNusselt	{6, 1, 1}	{32, 1, 1}	10.464	0.25
BCvelocity	{6, 1, 1}	{32, 1, 1}	9.889	0.25
calculateRECF	{6, 1, 1}	{32, 1, 1}	9.217	0.25
calculateError	{12, 2, 1}	{16, 16, 1}	8.577	1
OMGBoundaryCondition	{6, 1, 1}	{32, 1, 1}	7.617	0.25
updateFieldHalfTETA	{12, 2, 1}	{16, 16, 1}	6.976	1
BoundaryConditionlS	{6, 1, 1}	{32, 1, 1}	6.785	0.25
BoundaryConditionmW	{6, 1, 1}	{32, 1, 1}	6.336	0.25
BoundaryConditionlS	{6, 1, 1}	{32, 1, 1}	6.145	0.25
BoundaryConditionlS	{6, 1, 1}	{32, 1, 1}	5.984	0.25
calculateError	{12, 2, 1}	{16, 16, 1}	5.984	1
BoundaryConditionmW	{6, 1, 1}	{32, 1, 1}	5.952	0.25
BoundaryConditionmW	{6, 1, 1}	{32, 1, 1}	5.729	0.25
calculateError	{12, 2, 1}	{16, 16, 1}	5.472	1
BoundaryConditionmS	{6, 1, 1}	{32, 1, 1}	4.705	0.25
BoundaryConditionmS	{6, 1, 1}	{32, 1, 1}	4.513	0.25
TETABoundaryCondition	{6, 1, 1}	{32, 1, 1}	4.352	0.25
calcT	{12, 2, 1}	{16, 16, 1}	4.128	1
BoundaryConditionlW	{6, 1, 1}	{32, 1, 1}	3.776	0.25
BoundaryConditionlT	{6, 1, 1}	{32, 1, 1}	3.712	0.25
BoundaryConditionmS	{6, 1, 1}	{32, 1, 1}	3.585	0.25
updateField	{12, 2, 1}	{16, 16, 1}	3.521	1
BoundaryConditionlT	{6, 1, 1}	{32, 1, 1}	3.489	0.25
BoundaryConditionlT	{6, 1, 1}	{32, 1, 1}	3.488	0.25
updateField	{12, 2, 1}	{16, 16, 1}	3.457	1
BoundaryConditionlW	{6, 1, 1}	{32, 1, 1}	3.392	0.25
BoundaryConditionlW	{6, 1, 1}	{32, 1, 1}	3.36	0.25
outLet	{1, 1, 1}	{32, 1, 1}	3.104	0.25
outLet	{1, 1, 1}	{32, 1, 1}	3.104	0.25
updateField	{12, 2, 1}	{16, 16, 1}	3.104	1

outLet	{1, 1, 1}	{32, 1, 1}	3.072	0.25
readData	{1, 1, 1}	{1, 1, 1}	2.976	0.25
outLet	{1, 1, 1}	{32, 1, 1}	2.88	0.25
outLet	{1, 1, 1}	{32, 1, 1}	2.817	0.25
outLet	{1, 1, 1}	{32, 1, 1}	2.816	0.25
outLetl	{1, 1, 1}	{32, 1, 1}	2.656	0.25
outLetl	{1, 1, 1}	{32, 1, 1}	2.528	0.25
outLetl	{1, 1, 1}	{32, 1, 1}	2.496	0.25
outLetl	{1, 1, 1}	{32, 1, 1}	2.465	0.25
outLetl	{1, 1, 1}	{32, 1, 1}	2.464	0.25
outLetl	{1, 1, 1}	{32, 1, 1}	2.464	0.25
outLetl	{1, 1, 1}	{32, 1, 1}	2.433	0.25
outLetl	{1, 1, 1}	{32, 1, 1}	2.432	0.25
outLetl	{1, 1, 1}	{32, 1, 1}	2.208	0.25
updateHalfBCOMEGA	{6, 1, 1}	{32, 1, 1}	1.888	0.25

فصل ۵: نتیجه گیری و پیشنهادها

۵-۱ بحث و نتیجه‌گیری

در این پژوهش پردازش و شبیه‌سازی جریان نانوسیال در کانال با دیواره‌ی موج‌دار با استفاده از پردازنده‌ی گرافیکی مورد بحث و بررسی قرار گرفته و همچنین اثر کمیت‌های مختلف نظیر دامنه‌ی موج، عدد رینولدز، عدد پراتل، کسر حجمی نانوسیال و قطر نانوذره مورد بررسی قرار گرفته است. همان‌طور که گفته شد استفاده از قدرت پردازنده‌ی گرافیکی بعد از سال ۲۰۰۶ رواج بیشتری پیدا کرد و پژوهشگران در این عرصه، تحقیقات بسیاری انجام داده‌اند، مخصوصاً در زمینه‌ی دینامیک سیالات محاسباتی (CFD) که بر پایه و مبنای محاسبات عددی بنا شده است.

نتایج حاصل از این پژوهش به شرح ذیل است:

- با افزایش دامنه‌ی موج دیواره‌ی کانال، گرادیان سرعت روی دیواره زیاد می‌گردد که عاملی در جهت افزایش انتقال حرارت است. همچنین افزایش مقدار دامنه، ضریب اصطکاک پوسته‌ای را افزایش می‌دهد.
- از آنجایی که عدد رینولدز به صورت نسبت نیروی اینرسی به نیروی لزجت است، افزایش این کمیت موجب افزایش انتقال حرارت و ضریب اصطکاک پوسته‌ای می‌شود.
- افزایش عدد پراتل موجب افزایش عدد ناسلت و انتقال حرارت می‌شود ولی روی ضریب اصطکاک پوسته‌ای تأثیری نمی‌گذارد.
- افزایش کسر حجمی نانوسیال ضریب هدایت گرمایی سیال را افزایش می‌دهد که منجر به افزایش انتقال حرارت می‌شود اگرچه افزایش کسر حجمی نانوسیال باعث کاهش ضریب اصطکاک پوسته‌ای می‌شود.
- افزایش قطر نانوذره به کاهش انتقال حرارت منجر می‌شود.

- با بررسی جریان عبوری از کانال موج‌دار با پردازنده‌ی گرافیکی و مرکزی و با رهیافت توماس موازی (همان الگوریتم توماس کلاسیک که به شکل موازی اجرا می‌شود) و توماس کلاسیک مشخص شد که رهیافت توماس موازی در یک شبکه با اندازه‌ی 160×160 به ضریب تسریع $7/69$ دست پیدا می‌کند. در نهایت رهیافت توماس موازی با بهره‌گیری از الگوریتم توماس که بسیار سریع و قابل‌درک‌تر نسبت به الگوریتم‌های موجود موازی مانند کاهش متناوب و کاهش متناوب موازی، برای حل دستگاه معادلات سه‌قطری است، می‌تواند در گسترش استفاده از پردازش موازی در تحلیل مسائل مختلف مهندسی، بسیار راه‌گشا باشد و محققان را تشویق و ترغیب کند که با توجه آسان‌تر بودن و سریع‌تر بودن استفاده از چنین حل‌گری در قیاس با حل‌گرهای موازی موجود، از پردازنده‌های گرافیکی که پتانسیل و قدرت پردازشی بسیار بالایی دارند بهره‌ی بیشتری ببرند.

۲-۵ پیشنهادها

- به‌منظور ارتقای سطح کیفی پژوهش حاضر و انجام مطالعه‌ی جامع‌تر در راستای موضوع این پژوهش، پیشنهادهای ذیل مطرح می‌گردد:
- حل عددی جریان نانوسیال در کانال با دیواره‌های موج‌دار به همراه چشمه‌ی حرارتی و واکنش شیمیایی
 - حل عددی جریان نانوسیال در کانال با دیواره‌های موج‌دار به همراه چشمه‌ی حرارتی با در نظر گرفتن اثرات تشعشع^۱
 - حل عددی جریان نانوسیال در کانال با دیواره‌های موج‌دار با زاویه‌های موج متفاوت
 - حل عددی جریان نانوسیال در کانال با دیواره‌های موج‌دار به همراه چشمه‌ی حرارتی با در

¹ Radiation Effect

نظر گرفتن ویژگی‌های یک سیال غیر نیوتنی

- بررسی عملکرد چند دستگاه کارت گرافیک با چندپردازنده‌ی مرکزی یا یک پردازنده‌ی مرکزی با چندین هسته‌ی سهیم در پردازش
- بررسی عملکرد رهیافت توماس موازی در مسائلی که معادلات بیضوی و حل‌های تکراری ندارند برای مثال معادله هدایت حرارتی گذرا
- بررسی مسائلی که منجر به ایجاد دستگاه معادلاتی با تعداد قطر بیشتر مانند ۵ یا ۷ قطر می‌شوند با رهیافت توماس موازی
- بررسی عملکرد الگوریتم‌های موجود موازی مانند کاهش متناوب و کاهش متناوب موازی و مقایسه با رهیافت توماس موازی، برای حل دستگاه معادلات سه‌قطری و دستگاه معادلاتی با تعداد قطر بیشتر مانند ۵ یا ۷ قطر

مراجع

- [1] Bergles, A., Recent developments in convective heat transfer augmentation, *Applied Mechanics Reviews*, **26**, 1973, 675-682.
- [2] Ahuja, A. S., Augmentation of heat transport in laminar flow of polystyrene suspensions. I. Experiments and results, *Journal of Applied Physics*, **46**, 1975, 3408-3416.
- [3] Choi, S., Enhancing thermal conductivity of fluids with nanoparticles, *ASME-Publications-Fed*, **231**, 1995, 99-106.
- [4] Mustafa, M., Hina, S., Hayat, T. and Alsaedi, Influence of wall properties on the peristaltic flow of a nanofluid: analytic and numerical solution, *International Journal of Heat and Mass Transfer*, **55**(17), 2012, 4871-4877.
- [5] Choi, S., Zhang, Z., Yu, W., Lockwood, F. and Grulke, E., Anomalous thermal conductivity enhancement in nanotube suspensions, *Applied physics letter*, **79**(14), 2001, 2252-2254.
- [6] Khanfer, K., Vafai, K. and Lightstone, M., Buoyancy-driven heat transfer enhancement in two-dimensional enclosure utilizing nanofluids, *International Journal of Heat and Mass Transfer*, **46**(19), 2003, 3639-3653.
- [7] Maïga, S.E.B., Nguyen, C.T., Galanis, N. and Roy, G., Heat transfer behaviours of nanofluids in a uniformly heated tube, *Superlattices and Microstructures*, **35**(3), 2004, 543-557.
- [8] Maiga, S., Nguyen, C.T., Galanis, N. and Roy, G., Hydrodynamic and thermal behaviours of a nanofluid in a uniformly heated tube, *WIT Transactions on Engineering Sciences*, **46**, 2004, 543-557.
- [9] Maiga, S.E.B., Palm, S.J., Nguyen, C.T., Roy, G. and Galanis, N., Heat transfer enhancement by using nanofluids in forced convection flow, *International Journal of Heat and Fluid Flow*, **26**(4), 2005, 530-546.
- [10] Buongiorno, J., Convective transport in nanofluids, *Journal of Heat Transfer*, **128**(3), 2006, 240-250.

- [11] Abu-Nada, E., Masoud, Z. and Hijazi, A., Natural convection heat transfer enhancement in horizontal concentric using nanofluids, *International Communications in Heat and Mass Transfer*, **35**(5), 2008, 657-665.
- [12] Abu-Nada, E., Masoud, Z., Oztop, H.F. and Campo, A., Effect of nanofluid variable properties on natural convection in enclosures, *International Journal of Thermal Sciences*, **49**(3), 2010, 479-491.
- [13] Freidoonimehr, N., Rashidi, M.M. and Mahmud, S., Unsteady MHD free convective flow past a permeable stretching vertical surface in a nano-fluid, *International Journal of Thermal Sciences*, **87**, 2015, 136-145.
- [14] Uddin, M.B., Rahman, M., Khan, M., Saidur, R. and Ibrahim, T.A., Hydromagnetic double-diffusive mixed convection in trapezoidal enclosure due to uniform and nonuniform heating at the bottom side: Effect of Lewis number, *Alexandria Engineering Journal*, **55**(2), 2016, 1165-1176.
- [15] Rahman, M., et al., Unsteady analysis of natural convection in a carbon nanotube-water filled cavity with an inclined heater, *Numerical Heat Transfer, Part A: Applications*, **69**(7), 2016, 794-809.
- [16] Esfe, M.H., Behbahani, P.M., Arani, A.A.A, and Sarlak, M.R., Thermal conductivity enhancement of SiO₂-MWCNT (85: 15%)-EG hybrid nanofluids, *Journal of Thermal Analysis and Calorimetry*, **128**(1), 2017, 249-258.
- [17] Mikkola, V., Puupponen, S., Granbohm, H., Saari, K., Ala-Nissila, T. and Seppala, A., Influence of particle properties on convective heat transfer of nanofluids, *International Journal of Thermal Sciences*, **124**, 2018, 187-195.
- [18] Ali, N., Sajid, M., Javed, T. and Abbas, Z., Heat transfer analysis of peristaltic flow in a curved channel, *International Journal of Heat and Mass Transfer*, **53**(15-16), 2010, 3319-3325.
- [19] Latham, T., "Motion in a peristaltic pump [MS thesis]," edition: *MIT-Press, Cambridge, Mass, USA*, 1966.
- [20] Shapiro, A.H., Jaffrin, M.Y. and Weinberg, S.L., Peristaltic pumping with long wavelengths at low Reynolds number, *Journal of fluid mechanics*, **37**(4), 1969, 799-825.
- [21] Burns, J., A review of steam soak operations in California, *Journal of Petroleum Technology*, **21**(01), 1969, 25-34.

- [22] Asako, Y. and Faghri, M., Finite-volume solutions for laminar flow and heat transfer in a corrugated duct, *Journal of Heat Transfer*, **109**(3), 1987, 627-634.
- [23] Rush, T., Newell, T. and Jacobi, A., An experimental study of flow and heat transfer in sinusoidal wavy passages, *International journal of heat and mass transfer*, **42**(9), 1999, 1541-1553.
- [24] Wang, C.C. and Chen, C.K., Forced convection in a wavy-wall channel, *International Journal of Heat and Mass Transfer*, **45**(12), 2002, 2587-2595.
- [25] Nilpueng, K. and Wongwises, S., Flow pattern and pressure drop of vertical upward gas-liquid flow in sinusoidal wavy channels, *Experimental thermal and fluid science*, **30**(6), 2006, 523-534.
- [26] Heidary, H. and Kermani, M., Effect of nano-particles on forced convection in sinusoidal-wall channel, *International Communications in Heat and Mass Transfer*, **37**(10), 2010, 1520-1527.
- [27] Ahmed, M., Yusoff, M. and Shuaib, N., Effects of geometrical parameters on the flow and heat transfer characteristics in trapezoidal-corrugated channel using nanofluid, *International Communications in Heat and Mass Transfer*, **42**, 2013, 69-74.
- [28] Khoshvaght-Aliabadi, M., Hormozi, F. and Zamzamian, A., Role of channel shape on performance of plate-fin heat exchangers: experimental assessment, *International Journal of Thermal Sciences*, **79**, 2014, 183-193.
- [29] Vanaki, S.M., Mohammed, H., Abdollahi, A. and Wahid, M., Effect of nanoparticle shapes on the heat transfer enhancement in a wavy channel with different phase shifts, *Journal of Molecular Liquids*, **196**, 2014, 32-42.
- [30] Akbar, N.S. and Butt, A.W., Carbon nanotubes analysis for the peristaltic flow in curved channel with heat transfer, *Applied Mathematics and Computation*, **259**, 2015, 231-241.
- [31] Nazari, S. and Toghraie, D., Numerical simulation of heat transfer and fluid flow of water-CuO Nanofluid in a sinusoidal channel with a porous medium, *Physica E: Low-dimensional Systems and Nanostructures*, **87**, 2017, 134-140.
- [32] Ajeel, R.K., Salim, W.I. and Hasann, K., Thermal and hydraulic characteristics of turbulent nanofluids flow in trapezoidal-corrugated channel symmetry and zigzag shaped, *Case Studies in Thermal Engineering*, **12**, 2018, 620-635.

- [33] Tsutsui, S. and Collet, P., *Massively parallel evolutionary computation on GPGPUs*. Springer, 2013.
- [34] Larsen, E.S and McAllister, D., "Fast matrix multiplies using graphics hardware," in Proceedings of the 2001 ACM/IEEE conference on Supercomputing, 2001, 55-55 ACM.
- [35] Galoppo, N., Govindaraju, N.K., Henson, M. and Manocha, D., "LU-GPU: Efficient algorithms for solving dense linear systems on graphics hardware," in Proceedings of the 2005 ACM/IEEE conference on Supercomputing, 2005, p. 3: IEEE Computer Society.
- [36] Bell, N. and Garland, M., "Efficient sparse matrix-vector multiplication on CUDA," Nvidia Technical Report NVR-2008-004, Nvidia Corporation 2008.
- [37] Baskaran, M.M. and Bordawekar, R., "Optimizing sparse matrix-vector multiplication on GPUs using compile-time and run-time strategies," *IBM Reserach Report, RC24704 (W0812-047)*, 2008.
- [38] Li, R. and Saad, Y., "GPU-accelerated preconditioned iterative linear solvers," *The Journal of Supercomputing*, **63**(2), 2013, 443-466.
- [39] Molemaker, J, Cohen, J. M, Patel, S., and Noh .J, "Low viscosity flow simulations for animation," in Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2008, 9-18: Eurographics Association.
- [40] Zhang, Y., Cohen, J. and Owens, J. D., "Fast tridiagonal solvers on the GPU," *ACM Sigplan Notices*, **45**(5), 2010, 127-136.
- [41] Goddeke, D. and Strzodka, R., "Cyclic reduction tridiagonal solvers on GPUs applied to mixed-precision multigrid," *IEEE Transactions on Parallel and Distributed Systems*, **22**(1), 2011, 22-32.
- [42] Davidson, A., Zhang, Y. and Owens, J. D., "An auto-tuned method for solving large tridiagonal systems on the GPU," in Parallel & Distributed Processing Symposium (IPDPS), *2011 IEEE International*, 2011, 956-965: IEEE.
- [43] Egloff, D., "High performance finite difference PDE solvers on GPUs," QuantAlea GmbH, Zurich, Switzerland 2010, 2010.

- [44] Kim, H.-S., Wu, S., Chang, L.-w. and Wen-mei, W. H., "A scalable tridiagonal solver for GPUs," in *Parallel Processing (ICPP), 2011 International Conference on*, 2011, 444-453: IEEE.
- [45] Esfahanian, V., Baghapour, B., Torabzadeh, M. and Chizari, H., "An efficient GPU implementation of cyclic reduction solver for high-order compressible viscous flow simulations," *Computers & fluids*, **92**, 2014, 160-171.
- [46] Tutkun, B. and Edis, F.O., "A GPU application for high-order compact finite difference scheme," *Computers & Fluids*, **55**, 2012, 29-35.
- [47] Esfahanian, V., Darian, H. M. and Gohari, S. I., "Assessment of WENO schemes for numerical simulation of some hyperbolic equations using GPU," *Computers & Fluids*, **80**, 2013, 260-268.
- [48] Darian, H. M. and Esfahanian, V., "Assessment of WENO schemes for multi-dimensional Euler equations using GPU," *International Journal for Numerical Methods in Fluids*, **76**(12), 2014, 961-981.
- [49] Darian, H.M., Accelerating high-order WENO schemes using two heterogeneous GPUs, *JOURNAL OF COMPUTATIONAL APPLIED MECHANICS*, **48**(2), 2017, 161-170.
- [50] Fan, Z., Qiu, F., Kaufman, A. and Yoakum-Stover, S., "GPU cluster for high performance computing," in *Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, 2004, 47: IEEE Computer Society.
- [51] Rinaldi, P. R., Dari, E., Vénere, M. J. and Clause, A., "A Lattice-Boltzmann solver for 3D fluid simulation on GPU," *Simulation Modelling Practice and Theory*, **25**, 2012, 163-171.
- [52] Ren, Q. and Chan, C.L, Natural convection with an array of solid obstacles in an enclosure by lattice Boltzmann method on a CUDA computation platform, *International Journal of Heat and Mass Transfer*, **93**, 2016, 273-285.
- [53] Miller, R. and Boxer, L., "Algorithms Sequential and Parallel: A Unified Approach (Charles River Media Computer Engineering (Hardcover)), Charles River Media," Inc., Rockland, MA, 2005.
- [54] H. El-Rewini and M. Abd-El-Barr, *Advanced computer architecture and parallel processing*. John Wiley & Sons, 2005.

- [55] Kumar, V., Grama. A., Gupta, A. and Karypis. G., "*Introduction to Parallel Computing*", 2nd ed: Addison Wesley, 2003.
- [56] Koo, J. and Kleinstreuer, C., "A new thermal conductivity model for nanofluids," *Journal of Nanoparticle Research*, **6**(6), 2004, 577-588.
- [57] Anderson, J. D. and Wendt, J., *Computational fluid dynamics*. Springer, 1995.
- [58] Hammerlin. G. and Hoffmann. K., "Numerical Mathematics. 1991," ed: Springer Verlag, New York.
- [59] Napolitano, M., "Efficient ADI and spline ADI methods for the steady-state Navier-Stokes equations," *International journal for numerical methods in fluids*, **4**(12), 1984, 1101-1115.
- [60] Rubin, S. G., "Incompressible flow along a corner," *Journal of Fluid Mechanics*, **26**(1), 1966, 97-110.
- [61] Rubin, S. G. and Graves Jr, R. A., "Viscous flow solutions with a cubic spline approximation," *Computers & Fluids*, **3**(1), 1975, 1-36.
- [62] Rubin, S., "Incompressible Navier-Stokes and parabolized Navier-Stokes solution procedures and computational techniques," in In Von Karman Inst. for Fluid Dyn. Computational Fluid Dyn. 138 p (SEE N83-19024 09-34), 1982.
- [63] Pu, W., "Spline method of fractional steps in numerical model of unsteady natural convection flow at high Rayleigh number," *Numerical Heat Transfer, Part A Applications*, **11**(1), 1987, 95-108.
- [64] Lauriat, G. and Altimir, I., "A new formulation of the SADI method for the prediction of natural convection flows in cavities," *Computers & fluids*, **13**(2), 1985, 141-155.
- [65] Rubin, S. G. and Graves Jr, R. A., "*A cubic spline approximation for problems in fluid mechanics*," 1975.
- [66] Ho, C. T. and Johnsson, S. L., "Optimizing tridiagonal solvers for alternating direction methods on Boolean cube multiprocessors", *SIAM Journal on Scientific and Statistical Computing*, **11**(3), 1990, 563-592.
- [67] Hockney, R. W., "A fast direct solution of Poisson's equation using Fourier analysis", *Journal of the ACM*, **1**(12), 1965, 95-113.

Abstract

The heat transfer enhancement and type of heat transfer fluid has been the subject of much industrial and engineering research in recent decades. Changing the thermophysical properties of the fluid (such as adding nanoparticles to the base fluid) and the flow geometry (such as channels with corrugated walls) are two common ways to improve the heat transfer rate. Therefore, the present study investigates the flow and the subject of nanofluid heat transfer inside the channel with a corrugated wall. The governing equations for solving the nanofluid flow include mass conservation equation, momentum conservation and energy conservation which are discretized by the Spline Alternating Direction Implicit (SADI) method. The discretization of the governing equations lead to Tri-Diagonal system of equation. The rapid resolution of this equation is a challenging issue in Computational Fluid Dynamics (CFD) that uses the GPU to solve it. The GPU architecture is structured so that it can execute a computational instruction on a large amount of data in a short period of time in parallel.

The results demonstrated an increase in heat transfer due to the increase of the amplitude–wavelength, Reynolds number, Prandtl number and nanofluid volume fraction. Also, the maximum speed up of parallel Thomas algorithm against CPU runtime was about 7.69x in a 1600×160 grid size.

Keywords: Wavy wall channel, SADI, Nanofluid, GPU, CUDA Platform.



Shahrood University of Technology

Faculty of Mechanical Engineering

A Thesis submitted in partial Fulfillment of the Requirement for
The Degree of master of Science In
Mechanical Engineering

Numerical solution of nanofluid flow inside a wavy channel using the SADI method on GPUs

By: Seyyed Mohammad Hassan Taghavi

Supervisor:

Dr. Pooria Akbarzadeh

Advisor:

Dr. Hossien Mahmoodi Darian

September 2019