دانشکده مهندسی مکانیک

رشته مهندسی مکانیک

گرایش تبدیل انرژی

پروژه سوم دینامیک سیالات محاسباتی

حل عددی معادله مشتقات جزئی انتقال حرارت با استفاده از الگوریتم‌های مختلف گسسته‌سازی ترم جابجایی

دکتر حسن خالقی

نگارنده

سجاد خدادادی

۹۶۶۵۶۱۲۰۰۲

تیر ۱۳۹۷


بسم الله الرحمن الرحيم

فهرست مطالب

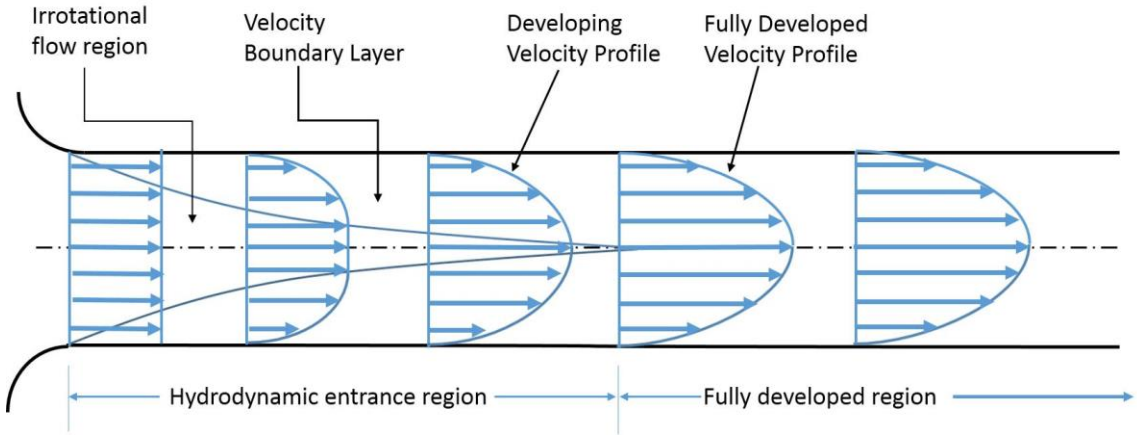
چکیده

در این پروژه به حل عددی معادله انتقال حرارت دوبعدی ناپایا با اســـتفاده از الگوریتم‌های مختلف گسسته‌سازی پرداخته شده است. برای دستگاه معادلات تشکیل شده در روش ADI از TDMA استفاده و در نهایت نتایج با هم مقایسه شده اند. کلیه کدها به زبان c++ نوشته شده‌اند.

مقدمه

جریان داخلی جریانی است که در آن سیال توسط یک سطح محصور می شود و اثرات لزجت رشد کرده و در تمام جریان مشاهده می‌گردد (مانند جریان در لوله). لذا لایه مرزی نمی تواند بدون محدودیت گسترش یابد.

هنگام بررسی جریان خارجی، فقط این سوال مطرح است که جریان لایه ای است یا متلاطم. ولی برای جریان داخلی باید وجود ناحیه ورودی یا ناحیه کاملاً فراگیر نیز بررسی شود.



شکل۱: ساختار کلی مساله

جریان لایه‌ای را بین دو صفحه تخت در نظر بگیرید، که در آن سیال با سرعت یکنواخت وارد لوله می‌شود. می‌دانیم که وقتی سیال با سطح تماس می‌گیرد، اثر ویسکوز قابل توجه می‌شود و لایه مرزی با افزایش x رشد می‌کند. در نتیجه ناحیه جریان ناویسکوز کوچک می‌شود و با فراگیری لایه مرزی در خط مرکزی از بین می‌رود.

پس از آن، اثر ویسکوز تمام مقطع عرضی را فرامی گیرد و نمایه سرعت با افزایش x تغییر نمی‌کند. در این حالت می‌گویند جریان کاملاً فراگیر است و فاصله از ورودی را تا جایی که این حالت روی می‌دهد طول ورودی هیدرودینامیکی($x_{fd,h}$) می‌گویند. نمایه سرعت کاملاً فراگیر برای جریان لایه‌ای بین دو صفحه تخت به صورت سهمی است. در جرین متلاطم نمایه صاف‌تر است و این ناشی از آمیختگی متلاطم در جهت شعاعی است. هنگام بررسی جریانهای داخلی اطلاع از وسعت ناحیه ورودی اهمیت دارد این وسعت به لایه‌ای یا متلاطم بودن جریان بستگی دارد .

معرفی مسئله و معادلات حاکم

عدد رینولدز تنها پارامتری است که بر طول ورودی تاثیر می‌گذارد:

$$Re = \frac{\rho U_m d}{\mu} \quad (۱)$$

که در آن U_m سرعت متوسط سیال در مقطع عرضی و d فاصله دو صفحه از هم است. در جریان کاملاً فراگیر عدد رینولدز بحرانی برای شروع تلاطم از مرتبه دو هزار می‌باشد یا به عبارتی:

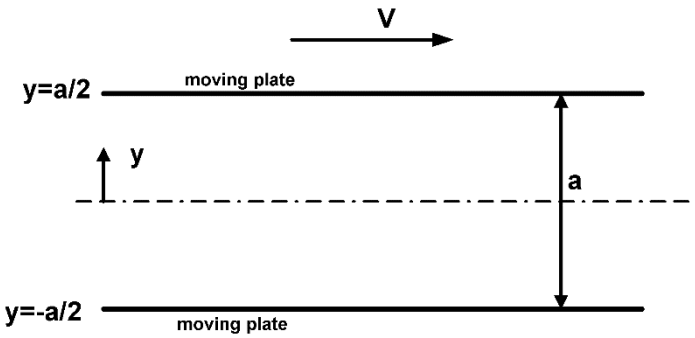
$Re_C \approx 2000$

البته برای برقراری شرایط کاملاً متلاطم عدد رینولدز باید خیلی بزرگتر باشد(Re ≈10000). گذار از جریان لایه ای به جریان متلاطم ممکن است در لایه مرزی ناحیه ورودی که در حال گسترش است روی دهد.

برای جریان لایه ای(Re ≈2000) طول ورودی هیدرودینامیکی را از عبارت زیر می توان به دست آورد:

$$\frac{x}{D} \approx 0.05 Re_D \quad (۲)$$

در این پروژه فرض بر این است که دو صفحه تخت متحرک باشند و سرعت آن‌ها و سرعت سیال ورود به آنها یکسان باشد. به ازای این شرط سرعت یک‌نواختی بر کل دامنه محاسباتی اعمال می‌شود که این سرعت برابر سرعت ورود سیال به این دو صفحه تخت می‌باشد. معادله انرژی برای این مساله حل شده و توزیع دما در این هندسه بدست خواهد آمد.



شکل۲: دو صفحه متحرک با سرعتی برابر سرعت جریان ورودی

در دیواره‌ها شرط مرزی دما ثابت وضع شده و فرض بر این است که خروجی کاملا توسعه یافته و یا به عبارتی دیگر گرادیان صفر است. در ورودی نیز شرط مرزی دما ثابت است.

معادله کلی انتقال دو بعدی به فرم زیر است:

$$\frac{\partial \phi}{\partial t}+\frac{\partial(\rho u \phi)}{\partial x}+\frac{\partial(\rho v \phi)}{\partial y}=\frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma \frac{\partial \phi}{\partial y}\right)+S_{\phi} \quad (۳)$$

که اگر در معادله بالا به جای ϕ دما جایگزین شود معادله انرژی حاصل می‌شود. هدف از این پروژه بررسی روش‌های گسسته‌سازی ترم جابجایی است. در ابتدا این ترم با روش مرکزی سپس با روش بالادست و در نهایت با روش ترکیبی که ترکیب روش‌های مرکزی و بالادست است، استفاده می‌شود.

با جایگزینی دما در رابطه بالا خواهیم داشت:

$$\rho C_p\left(\frac{\partial T}{\partial t}+\frac{\partial(\rho u T)}{\partial x}+\frac{\partial(\rho v T)}{\partial y}\right)=k\left[\left(\frac{\partial^2 T}{\partial x^2}\right)+\left(\frac{\partial^2 T}{\partial y^2}\right)\right]+S_{\phi} \quad (۴)$$

که در آن S_ϕ عبارتست از:

$$S_{\phi}=\mu\left[2\left\{\left(\frac{\partial^2 u}{\partial x^2}\right)+\left(\frac{\partial^2 v}{\partial y^2}\right)\right\}+\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2\right] \quad (۵)$$

۱-۱- روش مرکزی- مرکزی

در این گسسته‌سازی، ترم جابجایی به صورت مرکزی تقریب زده می‌شود. همانطور که برمی‌آید، دقت بالاتر در این تقریب وجود دارد و عدم نفوذ عددی از ویژگی مثبت دیگر این روش است. از معایب این روش عدم همخوانی با فیزیک است که پایین دست جریان بر ترم جابجایی تاثیر می‌گذارد درحالی که در واقعیت این امر ممکن نیست.

$$\rho C_p\left(\frac{\partial T}{\partial t}+\frac{\partial(\rho u T)}{\partial x}+\frac{\partial(\rho v T)}{\partial y}\right)=k\left[\left(\frac{\partial^2 T}{\partial x^2}\right)+\left(\frac{\partial^2 T}{\partial y^2}\right)\right]+S_{\phi} \quad (۶)$$

$$\rho C_p\left(\frac{T_{i,j}-{T_{i,j}}^n}{\Delta t}+u\frac{T_{i+1,j}-T_{i-1,j}}{2\Delta x}+v\frac{T_{i,j+1}-T_{i,j-1}}{2\Delta y}\right) = k\left[\left(\frac{T_{i+1,j}-2T_{i,j}+T_{i-1,j}}{\Delta x^2}\right)+\left(\frac{T_{i,j+1}-2T_{i,j}+T_{i,j-1}}{\Delta y^2}\right)\right]+S_{\phi} \quad (۷)$$

با تعریف $r=(\frac{\Delta x}{\Delta y})^2$ و عدد پکلت به صورت زیر خواهیم داشت:

$$Pe_x = \frac{\rho u \Delta x}{k} \quad (۸)$$

$$Pe_y = \frac{\rho v \Delta y}{k} \quad (۹)$$

با جایگذاری روابط بالا در معادله (۷):

$$\begin{aligned}\left(-1-\frac{C_p}{2}Pe_x\right)T_{i-1,j}+\left(\frac{\rho Cp}{k\Delta t}\Delta x^2+2+2r\right)T_{i,j}+\left(-1+\frac{C_p}{2}Pe_x\right)T_{i+1,j}\\ +\left(-r-\frac{C_p}{2}Pe_y\right)T_{i,j-1}+\left(-r+\frac{C_p}{2}Pe_y\right)T_{i,j+1}\\ =\frac{\rho Cp}{k\Delta t}\Delta x^2 {T_{i,j}}^n+\frac{S_\phi}{k}(\Delta x^2)\end{aligned} \quad (۱۰)$$

که شرایط نوسانی نشدن حل در آن عبارتند از:

$$|Pe_x| \le \frac{2}{C_p} \quad (۱۱)$$

$$|Pe_y| \le \frac{2}{C_p} \quad (۱۲)$$

۱-۲- روش بالادست- مرکزی

در این گسسته‌سازی، ترم جابجایی به صورت بالادست تقریب زده می‌شود. همانطور که برمی‌آید، دقت پایین‌تر در این تقریب وجود دارد و وجود نفوذ عددی از ویژگی منفی این روش است. از مزایای این روش همخوانی با فیزیک است که پایین دست جریان بر ترم جابجایی تاثیر نمی‌گذارد.

$$\rho C_p\left(\frac{\partial T}{\partial t}+\frac{\partial(\rho u T)}{\partial x}+\frac{\partial(\rho v T)}{\partial y}\right)=k\left[\left(\frac{\partial^2 T}{\partial x^2}\right)+\left(\frac{\partial^2 T}{\partial y^2}\right)\right]+S_\phi \quad (۱۳)$$

$$\begin{aligned}\rho C_p\left(\frac{T_{i,j}-{T_{i,j}}^n}{\Delta t}+u\frac{T_{i,j}-T_{i-1,j}}{\Delta x}+v\frac{T_{i,j}-T_{i,j-1}}{\Delta y}\right)\\ =k\left[\left(\frac{T_{i+1,j}-2T_{i,j}+T_{i-1,j}}{\Delta x^2}\right)+\left(\frac{T_{i,j+1}-2T_{i,j}+T_{i,j-1}}{\Delta y^2}\right)\right]+S_\phi\end{aligned} \quad (۱۴)$$

با تعریف $r=(\frac{\Delta x}{\Delta y})^2$ و عدد پکلت به صورت زیر خواهیم داشت:

$$Pe_x = \frac{\rho u \Delta x}{k} \quad (۱۵)$$

$$Pe_y = \frac{\rho v \Delta y}{k} \quad (۱۶)$$

با جایگذاری روابط بالا در معادله (۷):

$$\left(-1 - \frac{C_p}{2} Pe_x\right) T_{i-1,j} + \left(\frac{\rho Cp}{k\Delta t}\Delta x^2 + 2 + 2r + Cp(Pe_x + rPe_y)\right) T_{i,j} + (-1)T_{i+1,j} + \left(-r - \frac{C_p}{2} Pe_y\right) T_{i,j-1} + (-r)T_{i,j+1} = \frac{\rho Cp}{k\Delta t}\Delta x^2 {T_{i,j}}^n + \frac{S_\phi}{k}(\Delta x^2) \quad (۱۷)$$

ای روش هیچ گونه شرطی ندارد و بدون محدودیت برقرار است.

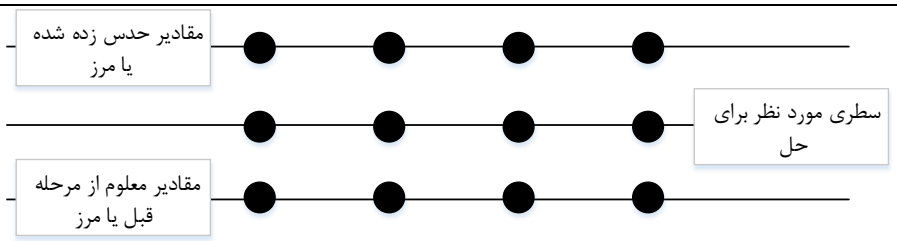
۱-۳- روش ترکیبی- مرکزی

این روش ترکیبی از دو روش بالا می‌باشد به این معنا که هر جا روابط ۱۱ و ۱۲ برقرا باشند از ترم جابجایی با استفاده از روش مرکزی و در غیر این صورت با استفاده از روش بالادست تقریب زده می‌شود. پس برای این روش بر روی عدد پکلت باید شرط گذاشته شود که در متناسب با مقدار آن نوع گسسته‌سازی مشخص شود.

با تشکیل ماتریس به یک ماتریس ۵ قطری می‌رسیم که برای حل آن از روش‌های مختلفی استفاده شده است که در ادامه به بررسی الگوریتم‌های آنان خواهیم پرداخت.

۱-۴- روش سطر به سطر

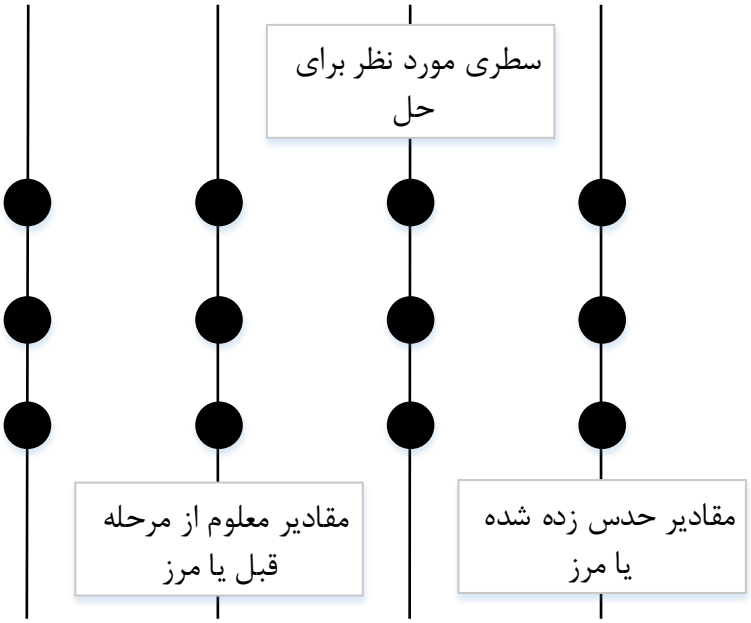
در این روش، با توجه به نوع مسأله، گسسته‌سازی را حول یک سطر یا ستون فرض کرده و عملاً مقادیر سطر پایین را یا از مرز از مرحله قبل و مقادیر سطر بالایی را حدس زده و این روش را تکرار نموده تا پاسخ یک دستگاه معادلات همگرا شود. مزیت این روش، حل دستگاه معادلات سه قطری می‌باشد.



شکل۲: شماتیک حل به روش خط به خط سطری

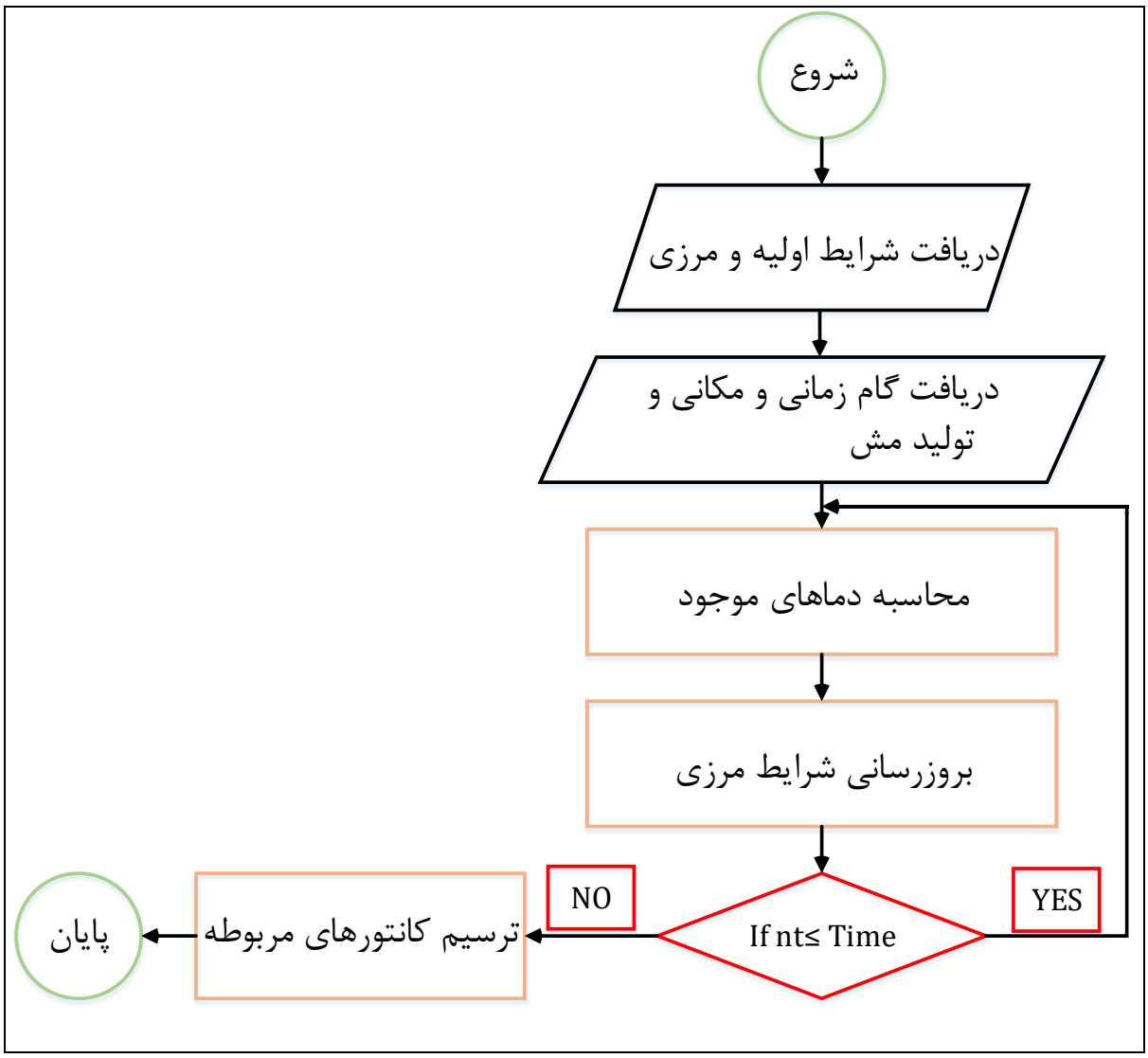
۱-۵- روش ستون به ستون

در این روش، با توجه به نوع مسأله، گسسته‌سازی را حول یک ستون فرض کرده و عملاً مقادیر ستون پایین دست را یا از مرز از مرحله قبل و مقادیر ستون بالادست را حدس زده و این روش را تکرار نموده تا پاسخ یک دستگاه معادلات همگرا شود. مزیت این روش، حل دستگاه معادلات سه قطری می‌باشد.



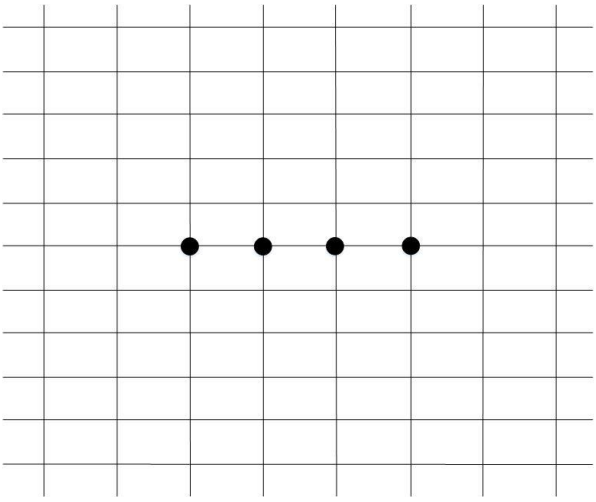
شکل۲: شماتیک حل به روش خط به خط سطری

فلوچارت روش سطر به سطر

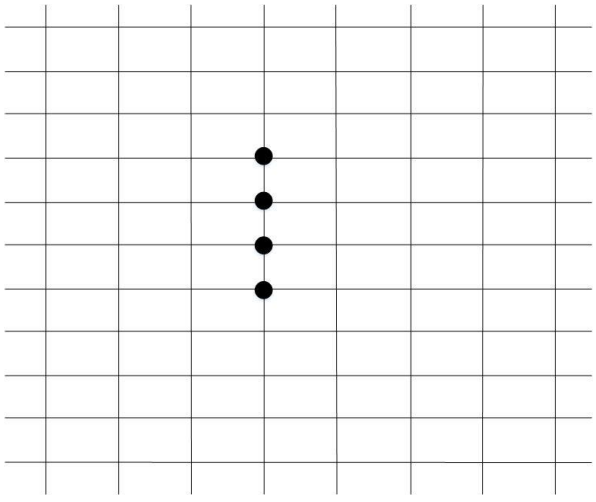


۱-۶- روش ADI

این روش یک روش ضمنی و ترکیبی از دو روش بالا است. تفاوت آن با ضمنی کامل در این است که برای جلوگیری از ۵ قطری شدن ماتریس ضرایب و مشکلات ناشی از آن یک جهت را صریح در نظر می‌گیرد. این روش دومرحله‌ای است. در مرحله اول در یک جهت(مثلا x) معادلات نوشته شده و یک دستگاه معادله تشکیل می‌شود و مقادیر در جهت دیگر معلوم فرض می‌شود(مثلا y). در مرحله بعد این رویه برعکس شده و جهت‌ها تعویض می‌شوند.

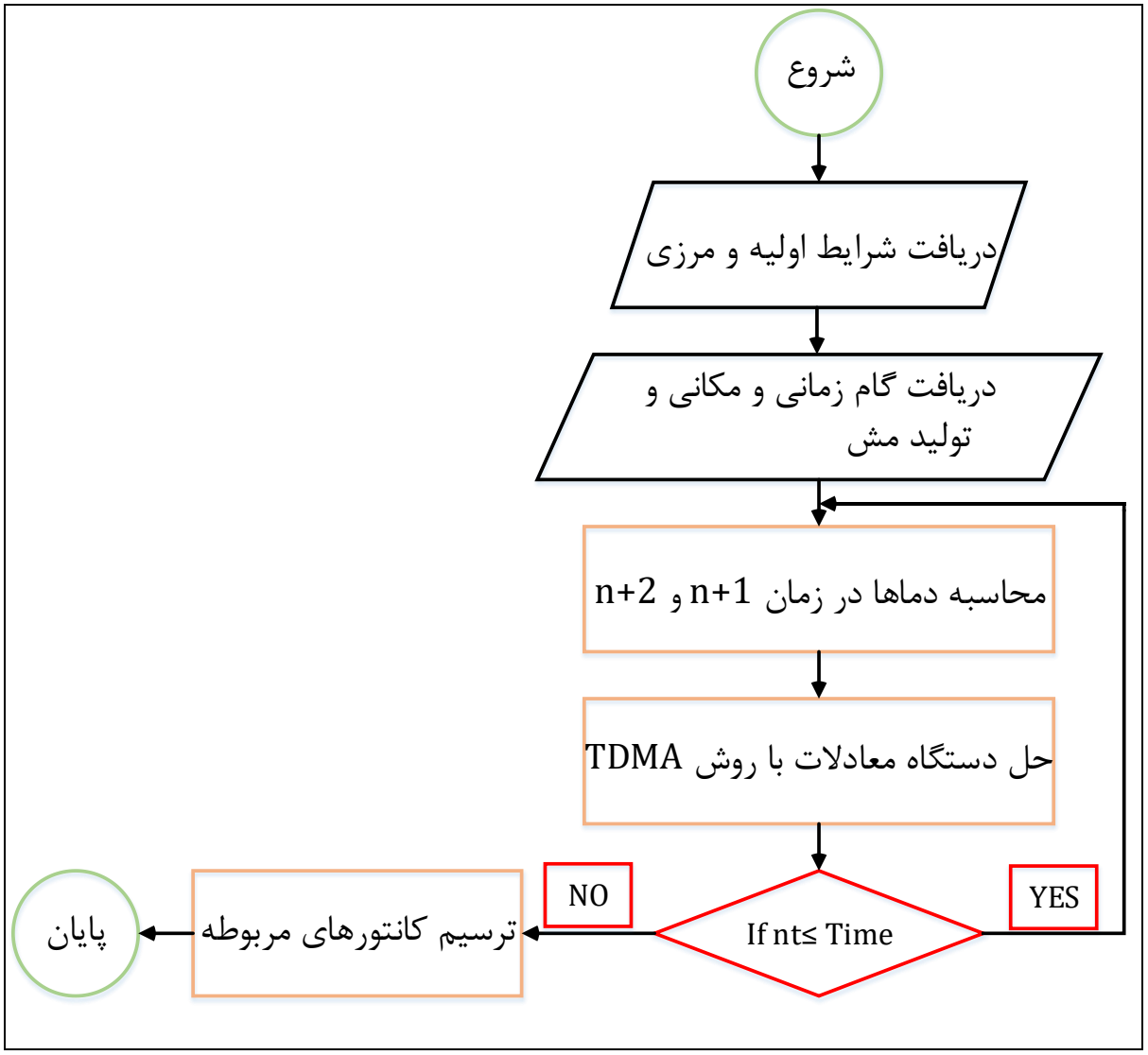


شکل۲: مرحله اول روش ADI



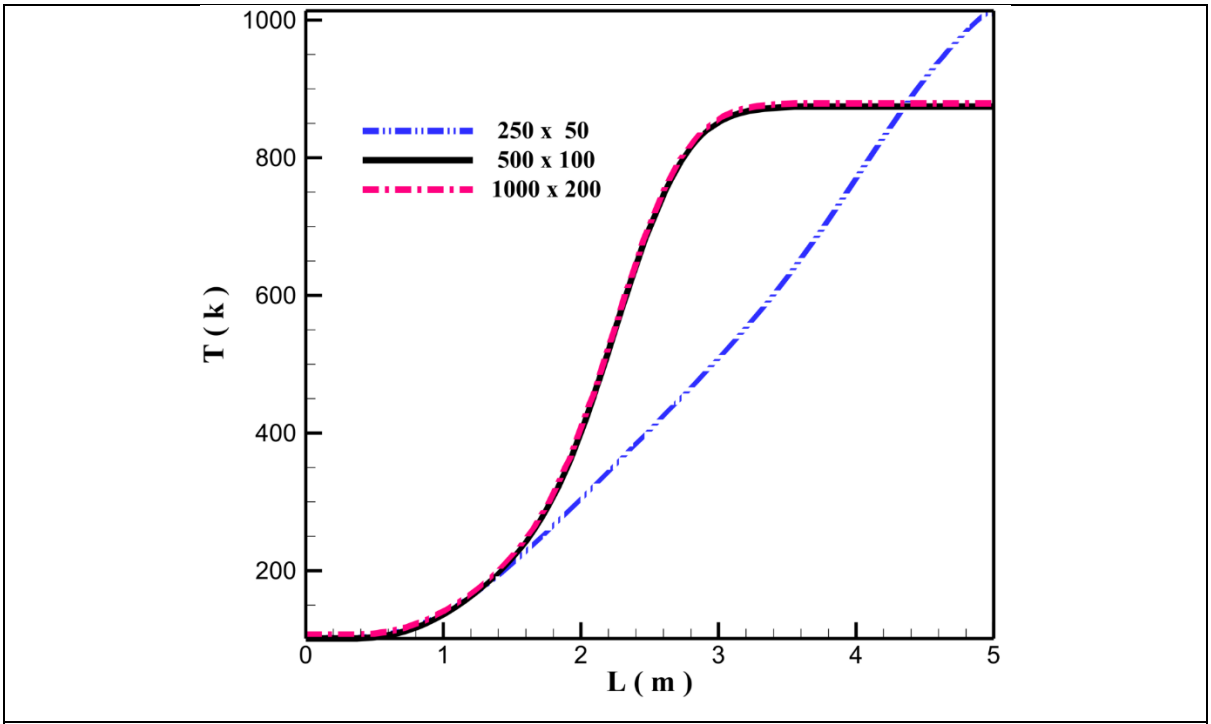
شکل۳: مرحله دوم روش ADI

فلوچارت روش ADI



استقلال حل از شبکه

برای این قسمت باید حالت بهینه‌ای بین هزینه محاسبات و دقت در حل را پیدا نمود. اندازه سلول‌های محاسباتی را انقدر ریز می‌کنیم تا تغییرات محسوسی در جواب‌ها حاصل نشود.



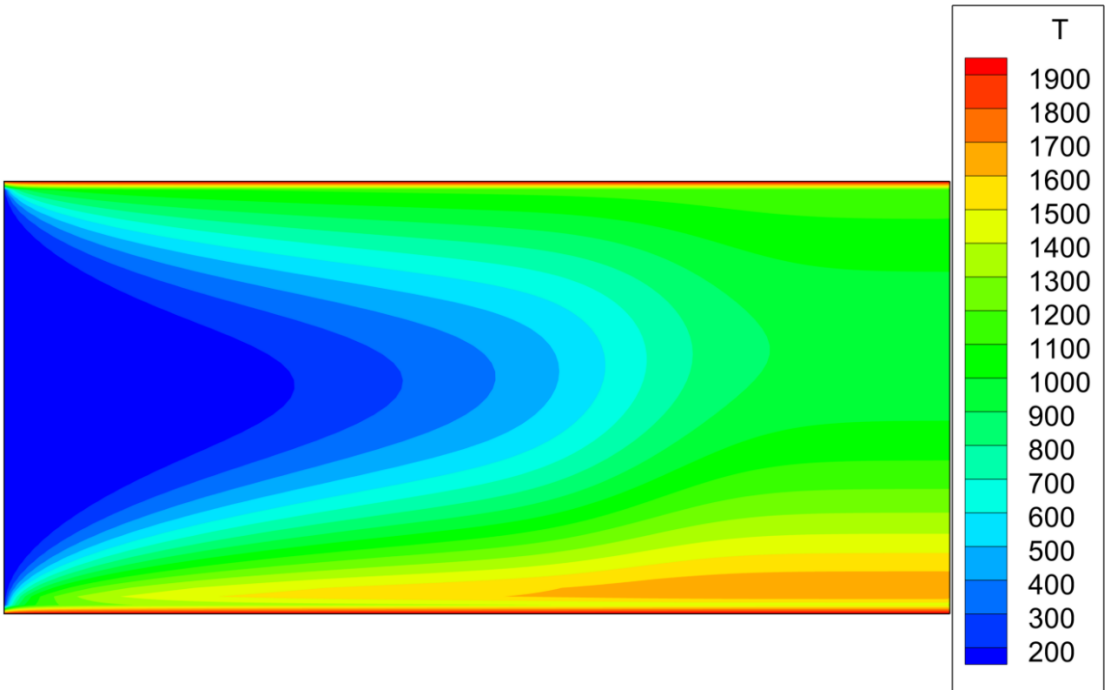
شکل۲: بررسی استقلال از شبکه

با استفاده از شکل بالا که دمای خط مرکزی دو صفحه تخت را نشان می‌دهد می‌توان دریافت که در راستای طول ۵۰۰ و در راستای عرض ۱۰۰ سلول ما را به جواب درست می‌رساند و از طرفی هزینه محاسبان بهینه است.

نتایج

دمای ورودی به لوله ۱۰۰، دمای دیواره ها ۲۰۰۰ درجه و دمای اولیه ۷۰۰ درجه می‌باشد. ضمنا شرط مرزی دمایی در خروجی گرادیان صفر است که نشان دهنده توسعه یافتگی حرارتی در انتهای لوله می‌باشد. سرعت در جهت x ، ۵ متر بر ثانیه و در راستای y صفر می‌باشد.

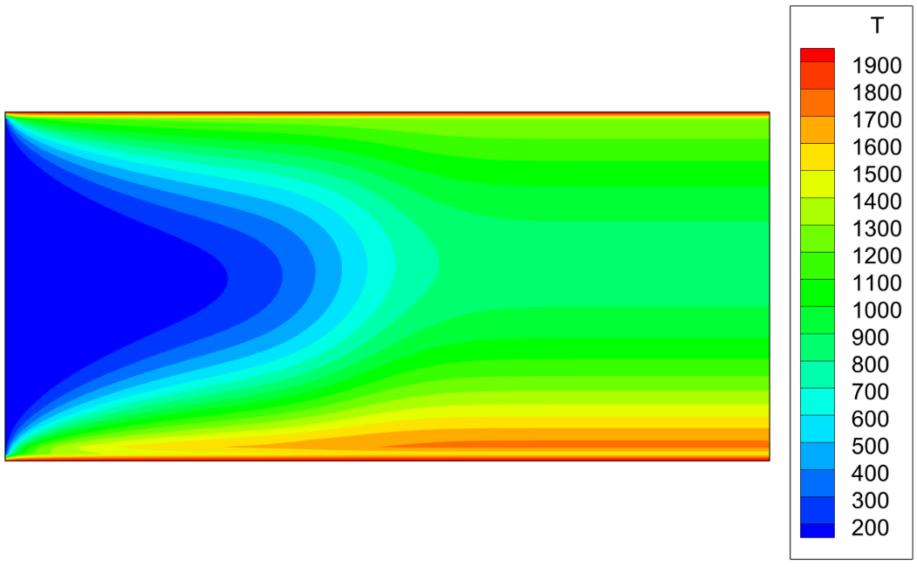
در ادامه این حالت، به بررسی نتایج برای گسسته‌سازی‌های مختلف ، به ازای سه روش حل دستگاه معادلات توضیح داده شده‌می‌پردازیم و از آن پس، با مبنا قرار دادن بهترین روش حل دستگاه معادلات، باقی محاسبات را انجام می‌دهیم. در قسمت‌های چالش‌های حل، به بررسی دلایل و علل عدم دریافت پاسخ فیزیکی مناسب از حل دستگاه به روش خط‌به‌خط پرداخته شده‌است.



شکل۲: گسسته سازی ترم جابجایی به صورت مرکزی و حل دستگاه معادله به روش ADI

شکل۲: گسسته سازی ترم جابجایی به صورت مرکزی و حل دستگاه معادله به روش خط به خط

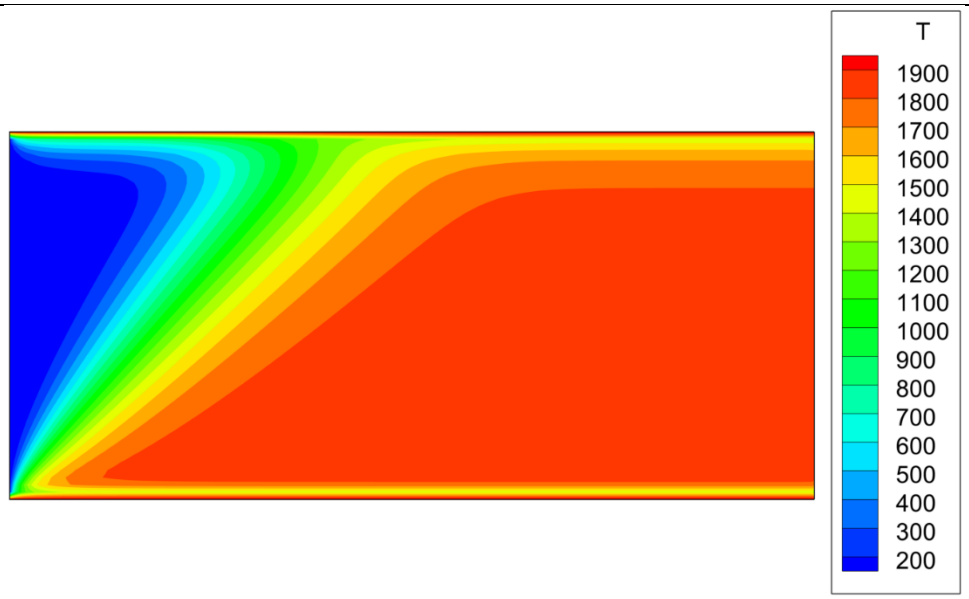
همانطور که قابل مشاهده است، پاسخی نسبتاً فیزیکی دریافت نمی‌شود، اثر جابجایی را به وضوح نمی‌توان دید. ولی اثر نفوذ قابل مشاهده می‌باشد.



شکل۲: گسسته سازی ترم جابجایی به صورت بالادست و حل دستگاه معادله به روش ADI

جواب این حالت، جوابی فیزیکی‌تر و قابل قبول می‌باشد و جابجایی را نیز درک می‌کند. سرعت حل و نیاز به تعداد تکرار کم برای هر مرحله زمانی برای همگرایی، از نکات مثبت این روش می‌باشد.

اگر مساله در راستای y نیز سرعت داشته باشد خواهیم داشت:



شکل۲: بررسی تاثیر سرعت در راستای y برای گسسته سازی ترم جابجایی به صورت بالادست و حل دستگاه معادله به روش ADI

چالش‌ها

عدم دریافت پاسخ فیزیکی مناسب در صورت حل دستگاه به روش خط به خط ستونی یکی از مشکلات مواجه شده، عدم درک پدیده جابجایی در شرایطی که دستگاه را ستونی حل کنیم می‌باشد. با توجه به نوع شرایط مرزی، درک مناسبی از پدیده پخش را می‌توان دریافت نمود، اما در انتقال یک راهه پدیده ،جابجایی عملکرد مناسبی از خود به جای نگذاشته است. به عنوان راه‌حل پیشنهادی، استفاده از روش ،ADI اعمال همزمان حل سطری و ستونی می‌باشد، تا مشکلات اشاره در این دو مورد تا نداشته و برطرف گردند.

بدلیل وجود شرط مرزی دریکله در انتهای دامنه محاسباتی روش خط به خط سطری با عدم همگرایی مواجه شد.

در روش گسسته‌سازی مرکزی باید به عدد پکلت حواسمان باشد که از حد بحرانی رد نشود.

نتایج

گسسته‌سازی مرکزی کمترین زمان اجرا را نیازمند بود، البته محدودیتی که برای عدد پکلت دارا می‌باشد، به شدت این روش را محدود می‌نماید. با توجه به نوع شبکه‌بندی، تقریبا اکثر نقاط شبکه در محدوده پکلت مناسب برای این گسسته‌سازی قرار می‌گیرند و دقت مناسبی دارا می‌باشد. روش ،بالادست علی رغم محدودیت نداشتن برای عدد ،پکلت اما دقت پایین‌تر،دارد که علت آن نفوذ عددی است.

در مجموع، اگر شبکه مناسبی داشته و از شرط عدد پکلت اطمینان حاصل شود، گسسته‌سازی مرکزی با حلگر ADI بهترین پاسخ را می‌دهد.در غیر این صورت از روش ترکیبی و حلگر ADI استفاده شود.

کد الگوریتم‌های مختلف

کد روش مرکزی-مرکزی و حل با استفاده از ADI

```
#include<iostream>
#include<math.h>
#define Max 100
#define dt 0.01
#define rho 1000
#define Cp 1.068
#define k 0.6
#include <fstream>//New

using namespace std;

      double alpha(double);
int main()
{
       ofstream file;
       file.open("results");
        double
T[Max][Max],Tnew[Max][Max],Tv[Max][Max],A1[Max][Max],A2[Max][Max],X[Max],U[i],u,B1[M
ax],B1[Max],Y[Max],h;
       int t=0,grid=82,Z=0,i=0,criteria,nx,ny,nt;
       double
Xend,Yend,dx,dy,U_inlet,V_inlet,pex,pey,r,dt,BB,error=5,d,a1,a2,a3,a4,a5;
       cout<<"enter time plz:";
       cin>>t;
       dt=0.1;
       X[0]=0.0;
       Y[0]=0.0;
       dx=0.01;
       dy=0.02;
       Xend=5;
       Yend=2;
       h=(Yend-Y[0])/2;
       ny=Xend-X[0];
       ny=Yend-Y[0];
       criteria=2/Cp;
       U_inlet=10;
       V_inlet=2.0;
       pex=rho*U_inlet*dx/k;
       pey=rho*V_inlet*dy/k;
       r=pow((dx/dy),2);
       H=0;
    while(X[i]<=Xend)
              {
                            X[i+1]=X[i]+dx;
                            i=i+1;
              }
              nx=i;
              i=0;
                  while(Y[i]<=Xend)
              {
                            Y[i+1]=Y[i]+dy;
                            i=i+1;
              }
              ny=i;
              i=0;

       for(int i=0;i<=grid;i++)
```

```
	{
		for(int j=0;j<=grid;j++)
		{
			if(i==0)
			{
				T[i][j]=100;
				//200
			}
			if(i!=0&&j==0)
			{
				T[i][j]=1000;
				//10
			}
			if(i==grid&&j!=0)
			{
				T[i][j]=400;
				//25
			}
			if(i!=0&&i!=grid&&j==grid)
			{
				T[i][j]=1000;
				//0
			}
		}

	for(int i=0;i<=grid;i++)
		{
			for(int j=0;j<=grid;j++)
			{
				if(i!=0&&i!=grid&&j!=0&&j!=grid)
				{
					T[i][j]=500;
				}
			}
		}

		a1=(-Cp/2)*pex-1;
		a2=(rho*Cp*dx*dx)/(k*dt)+(2*r)+2;
		a3=(Cp/2)*pex-1;
		a4=(-Cp/2)*pey*r-r;
		a5=(Cp/2)*pey*r-r;
		d=(rho*Cp*dx*dx)/(k*dt);
	for(int i=1;i<ny-1;i++)
		{
			if(i==1)
			{
				A2[i][i]=a2;
				A2[i][i+1]=a5;
			}
			else if(i==ny-1)
			{
				A2[i][i]=a2;
				A2[i][i-1]=a4;
			}
			else
			{
				A1[i][i]=a2;
				A1[i][i-1]=a1;
				A1[i][i-1]=a4;
			}
		BB=i;
		}
```

```
nt=BB;
for(int i=1;i<ny-1;i++)
	{
		if(i==1)
		{
			A2[i][i]=a2;
			A2[i][i+1]=a5;
		}
		else if(i==ny-1)
		{
			A2[i][i]=a2;
			A2[i][i-1]=a4;
		}
		else
		{
			A2[i][i]=a2;
			A2[i][i-1]=a1;
			A2[i][i+1]=a5;
		}
	}

	for(int i=1;i<nx;i++)
	{
		if(i==1)
		{
			A1[i][i]=a2;
			A1[i][i+1]=a3;
		}
		else if(i==ny-1)
		{
			A1[i][i]=a2;
			A1[i][i-1]=a1+a3;
		}
		else
		{
			A1[i][i]=a2;
			A1[i][i-1]=a1;
			A1[i][i+1]=a3;
		}
	}



i=0;
while(H<t)
{
while(error>1)
	{
  for(int j=1;j<=nx;j++)
  {
		for(int V=1;V<=nx,V++)
		{
		B1[j]=d*T[j][V];
		B1[j]=B1-a5*T[j+1][V]-a4*Tnew[j-1][V];
		B1[1]=B1[1]-a1*T[j][1];
		Tv[V][j]=TDMAsolver(A1[V][j],B1[V]);
		Tnew[j][V]=Tv[V][j];
	 T[j][V]=Tnew[V][j];
		}
	 }
```

```
        for(int j=1;j<=nx-1;j++)
        {
                    for(int V=1;V<=nx,V++)
                    {
                  B2[j]=d*T[j][V];
                    B2[j]=B2[i]-a3*T[j][V+1]-a1*T[j][V-1];
                    Tv[V][j]=TDMAsolver(A2[V][j],B2[V]);
                    Tnew[j][i]=Tv[j][V];
                    T[j][i]=Tnew[j][V];
                    }
               }


             error= Tnew[j][i]-T[j][i];
             }
              H=H+dt;
     cout<<"Time:"<<H<<endl;
      }


             for(int i=0;i<=nx;i++)
             {
                    for(int j=0;j<=ny;j++)
                    {
                           file<<i<<" ";
                           file<<j<<" ";
                           file<<Tnew2[i][j]<<endl;
                    }
                           cout<<endl;
             }
    file.close();
       return 0;
}

double TDMAsolver(double X,double Y)
{
       for(int i=0;i<=nx-1;i++)
       {
                        B[i+1]=((-A[i+1][i]/A[i][i])*B[i])+B[i+1];
                 A[i+1]=((-A[i+1][i]/A[i][i])*(A[i][j]))+A[i+1][j];
         }
              for(int j=ny;j<0;j--)
                    {
                                        if(j==ny)
                                               {
                                                      X[i]=B[i]/A[i][i];
                                               }
                                        else
                                               {
                                                      X[i]=B[i]-
X[i+1]*A[i][i+1]/A[i][i];
                                               }

                           }
}
```

کد روش بالادست- مرکزی و حل به روش ADI

```
#include<iostream>
#include<math.h>
#define Max 100
#define dt 0.01
#define rho 0.524
#define Cp 1.068
#define k 0.0515
#include <fstream>//New

using namespace std;

      double alpha(double);
int main()
{
       ofstream file;
       file.open("results");
        double
T[Max][Max],Tnew[Max][Max],Tv[Max][Max],A1[Max][Max],A2[Max][Max],X[Max],U[i],u,B1[M
ax],B1[Max],Y[Max],h;
       int t=0,grid=82,Z=0,i=0,criteria,nx,ny,nt;
       double
Xend,Yend,dx,dy,U_inlet,V_inlet,pex,pey,r,dt,BB,error=5,d,a1,a2,a3,a4,a5;
       cout<<"enter time plz:";
       cin>>t;
       dt=0.1;
       X[0]=0.0;
       Y[0]=0.0;
       dx=0.01;
       dy=0.02;
       Xend=5;
       Yend=2;
       h=(Yend-Y[0])/2;
       ny=Xend-X[0];
       ny=Yend-Y[0];
       criteria=2/Cp;
       U_inlet=10;
       V_inlet=2.0;
       pex=rho*U_inlet*dx/k;
       pey=rho*V_inlet*dy/k;
       r=pow((dx/dy),2);
       H=0;
    while(X[i]<=Xend)
              {
                     X[i+1]=X[i]+dx;
                     i=i+1;
              }
              nx=i;
              i=0;
                  while(Y[i]<=Xend)
              {
                     Y[i+1]=Y[i]+dy;
                     i=i+1;
              }
              ny=i;
              i=0;

       for(int i=0;i<=grid;i++)
       {
              for(int j=0;j<=grid;j++)
              {
```

```
                if(i==0)
                {
                        T[i][j]=100;
                        //200
                }
                if(i!=0&&j==0)
                {
                        T[i][j]=1000;
                        //10
                }
                if(i==grid&&j!=0)
                {
                        T[i][j]=400;
                        //25
                }
                if(i!=0&&i!=grid&&j==grid)
                {
                        T[i][j]=1000;
                        //0
                }
        }

for(int i=0;i<=grid;i++)
        {
                for(int j=0;j<=grid;j++)
                {
                        if(i!=0&&i!=grid&&j!=0&&j!=grid)
                        {
                                T[i][j]=500;
                        }
                }
        }

        a1=(-Cp/2)*pex-1;
        a2=(rho*Cp*dx*dx)/(k*dt)+(2*r)+2+(Pex*Cp)+(r*Pey*Cp);
        a3=-1;
        a4=-Cp*pey*r-r;
        a5=-r;
        d=(rho*Cp*dx*dx)/(k*dt);

for(int i=1;i<ny-1;i++)
        {
                if(i==1)
                {
                        A2[i][i]=a2;
                        A2[i][i+1]=a5;
                }
                else if(i==ny-1)
                {
                        A2[i][i]=a2;
                        A2[i][i-1]=a4;
                }
                else
                {
                        A2[i][i]=a2;
                        A2[i][i-1]=a1;
                        A2[i][i+1]=a5;
                }
        }

        for(int i=1;i<nx;i++)
        {
```

```
                    if(i==1)
                    {
                           A1[i][i]=a2;
                           A1[i][i+1]=a3;
                    }
                    else if(i==ny-1)
                    {
                           A1[i][i]=a2;
                           A1[i][i-1]=a1+a3;
                    }
                    else
                    {
                           A1[i][i]=a2;
                           A1[i][i-1]=a1;
                           A1[i][i+1]=a3;
                    }
             }



      i=0;
      while(H<t)
     {
      while(error>1)
             {
        for(int j=1;j<=nx;j++)
        {
                     for(int V=1;V<=nx,V++)
                     {
                     B1[j]=d*T[j][V];
                     B1[j]=B1-a5*T[j+1][V]-a4*Tnew[j-1][V];
                     B1[1]=B1[1]-a1*T[j][1];
                     Tv[V][j]=TDMAsolver(A1[V][j],B1[V]);
                     Tnew[j][V]=Tv[V][j];
              T[j][V]=Tnew[V][j];
                     }
              }



        for(int j=1;j<=nx-1;j++)
        {
                     for(int V=1;V<=nx,V++)
                     {
                 B2[j]=d*T[j][V];
                     B2[j]=B2[i]-a3*T[j][V+1]-a1*T[j][V-1];
                     Tv[V][j]=TDMAsolver(A2[V][j],B2[V]);
                     Tnew[j][i]=Tv[j][V];
                     T[j][i]=Tnew[j][V];
                     }
              }


             error= Tnew[j][i]-T[j][i];
             }
              H=H+dt;
     cout<<"Time:"<<H<<endl;
       }


             for(int i=0;i<=nx;i++)
```

```
		{
			for(int j=0;j<=ny;j++)
			{
				file<<i<<" ";
				file<<j<<" ";
				file<<Tnew2[i][j]<<endl;
			}
				cout<<endl;
		}
    file.close();
	return 0;
}

double TDMAsolver(double X,double Y)
{
	for(int i=0;i<=nx-1;i++)
	{
			B[i+1]=((-A[i+1][i]/A[i][i])*B[i])+B[i+1];
		A[i+1]=((-A[i+1][i]/A[i][i])*(A[i][j]))+A[i+1][j];
	   }
	     for(int j=ny;j<0;j--)
		{
					if(j==ny)
						{
							X[i]=B[i]/A[i][i];
						}
					else
						{
							X[i]=B[i]-
X[i+1]*A[i][i+1]/A[i][i];
						}

				}
}
```