

**زیربرنامه Interpolation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| توسعه دهنده: | نام و مقطع تحصیلی  محمد هاشمی اباتری، ارشد | رشته | مهندسی مکانیک | C:\Users\Mohammad\Desktop\University_of_Tehran_logo.svg.png |
| گرایش | تبدیل انرژی |
| حوزه تخصصی پروژه | آیرودینامیک |
| نرم افزار/ زبان برنامه نویسی استفاده شده | Fortran 90 |
| استاد راهنما دکتر/ همکار .................. | | | C:\Users\Mohammad\Desktop\University_of_Tehran_logo.svg.png |
| تهیه کننده مستند: | محمد هاشمی اباتری | | | |
| تاریخ تنظیم سند: | 9 **/** 10 **/**1395 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Interpolation(Dim,WNP1,RES,Error,NOL,FTC)** | | | |
| **Dimension** | **Type** | **Description** | **Intent** |
|  |  |  | **Input** |
|  | Integer | Maximum Dimension of Arrays | Dim |
|  | Integer | **N**umber **o**f **L**evel | NOL |
|  | Integer | **F**ine **t**o **C**oarse | FTC |
|  |  |  | **Input/Output** |
| (1:4,1:Dim) | Real(8) | Conservative Values at (N+1)th Time Step | WNP1 |
| (1:4,1:Dim) | Real(8) | **Res**idual | RES |
| (1:4,1:Dim) | Real(8) | Error | Error |

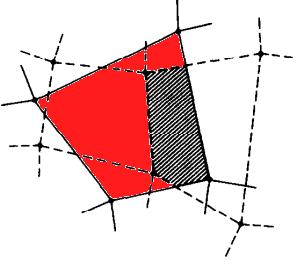
* 1. **وظایف**

در این زیربرنامه اطلاعات دو شبکه از طریق دو فایل Mesh1.txt و Mesh2.txt از کاربر گرفته شده، تا درون یابی داده ها بین آن دو انجام گیرد.

* 1. **توضیحات و تئوری ها**

در زیر مراحل صورت گرفته در زیربرنامه Interpolation توضیح داده شده است. کلیت زیربرنامه به این صورت می باشد که می بایست برای درون یابی، تقاطع بین سلول های شبکه ریز و درشت پیدا شده و مساحت آن مورد محاسبه قرار گیرد. برای این منظور باید مکان هر ضلع و میزان حضور آن را در سلول های شبکه 2 (یعنی شبکه مجهول) بیابیم. (منظور از میزان حضور یک ضلع اینست که ممکن است یک ضلع به طور کامل درون یک سلول قرار گیرد و یا اینکه بخشی از آن در سلول باشد)

از آنجایی که تقاطع بین دو سلول یک چند ضلعی را تشکیل می دهد و این چند ضلعی حاوی اضلاع شبکه 1 و هم شبکه 2 می باشد، می بایست مکان یابی اضلاع، برای اضلاع شبکه 2 بروی شبکه 1 نیز صورت پذیرد. شکل زیر به درک مفاهیم فوق کمک بیشتری خواهد نمود.

****

شکل (1) نمایی از دو شبکه قرار گرفته روی هم جهت درون یابی

همانطور که شکل فوق نشان می دهد به دنبال محاسبه مساحت ناحیه هاشور خورده هستیم. هر کدام از اضلاع ناحیه هاشور خورده سهمی در این مساحت دارند که به صورت رابطه زیر محاسبه می شود:

|  |  |
| --- | --- |
| (1) |  |

در این رابطه نقاط ابتدا و انتهای هر ضلع که درون ناحیه هاشور خورده قرار دارند به ترتیب با و نشان داده شده است و و مختصات این نقاط می باشد. در صورتی که یک ضلع به طور کامل درون ناحیه نباشد باید مختصات محل برخورد را بدست آوریم. رابطه فوق فقط جهت محاسبه ناحیه هاشور خورده مورد استفاده قرار می گیرد که بعد از محاسبه رابطه فوق برای تمامی اضلاع، مساحت این ناحیه به صورت زیر بدست می آید:

|  |  |
| --- | --- |
| (2) |  |

ولی برای انتقال داده ها مسئله به گونه دیگه ایست. اضلاعی که به صورت خط چین در شکل 1 نشان داده شده است مربوط به شبکه 1 و اضلاعی که به صورت خطوط ممتد هستند مربوط به سلول های شبکه 2 می باشند. برای یافتن مقادیر جریان در سلول های شبکه 2 مثل شکل 1 که با رنگ قرمز نشان داده شده است (ناحیه هاشور خورده نیز محسوب می شود) باید تأثیر اضلاع خط چین (مربوط به شبکه 1) و اضلاع ممتد (مربوط به خود شبکه 2) که درون ناحیه قرمز قرار دارند محاسبه شود. تاثیر اضلاع خط چین در درون یابی به صورت زیر محاسبه می شود:

|  |  |
| --- | --- |
| (3) |  |

که در آن J شماره سلول ناحیه قرمز رنگ و ME و NE سلول های چپ و راست ضلع مورد بررسی از شبکه 1 می باشد. در صورتی که ضلع مورد بررسی از اضلاع مرزی باشد مقدار خواهد بود. بعد از محاسبه رابطه فوق برای تمامی اضلاع موجود از شبکه 1 در ناحیه قرمز رنگ، میزان مشارکت اضلاع این شبکه در این ناحیه به صورت زیر بدست می آید:

|  |  |
| --- | --- |
| (4) |  |

در رابطه فوق مساحت ناحیه قرمز رنگ می باشد. برای محاسبه میزان مشارکت اضلاع ممتد در ناحیه قرمز رنگ از روابط (5) و (6) استفاده می شود. از آنجایی که این ضلع برای دو سلول مشترک است، باید برای هر دو محاسبه شود یعنی هم سلول سمت راست و هم سلول سمت چپ این ضلع، که با یک علامت منفی از هم متمایز می شوند.

|  |  |
| --- | --- |
| (5) |  |
| (6) |  |

در این رابطه I سلولی از شبکه 1 می باشد که ضلع شبکه 2 درون آن قرار دارد. همچنین ME و NE سلول هایی از شبکه 2 می باشد که قرار است مقدار میدان روی آنها درون یابی شود. در نهایت بعد از محاسبه رابطه فوق برای تمامی اضلاع ممتد در ناحیه قرمز رنگ، تاثیر این اضلاع برای درون یابی این سلول به صورت زیر بدست می آید: (در این قسمت برای مثال و انجام ادامه محاسبات فرض شده است که ناحیه قرمز رنگ در سمت چپ تمامی اضلاع ممتد باشد ولی در حالت کلی ممکن است این ناحیه برای برخی اضلاع به عنوان سلول ME و برای برخی سلول NE باشد)

|  |  |
| --- | --- |
| (7) |  |

بعد از انجام محاسبات فوق مقدار پارامتر مورد نظر از جریان، به صورت زیر برای ناحیه قرمز رنگ بدست خواهد آمد:

|  |  |
| --- | --- |
| (8) |  |

* 1. **بخش های زیربرنامه**

1. فراخوانی دو شبکه جهت درون یابی

در این قسمت دو شبکه متوالی که قرار است مقادیر میدان از اولی روی دومی درون یابی شود به صورت جداگانه فراخوانده می شود. به دلیل اینکه برخی اطلاعات هر دو شبکه در طی درون یابی مورد نیاز است همین ابتدا هر دو شبکه به صورت جدا فراخوانی شده است. ازین به بعد شبکه حاوی مقادیر میدان، شبکه 1 و شبکه ای که این مقادیر روی آن درون یابی می شود، شبکه 2 نام گذاری خواهد شد.

1. تعریف دو شمارنده جهت ذخیره شماره های نقاط تلاقی

چون در درون یابی دو شبکه اضلاع آنها با هم تلاقی دارند، و از آنجا که مختصات این تلاقی باید در ماتریس x , y ذخیره شود در این قسمت دو شمارنده برای هر دو شبکه تعریف می شود که مقدار اولیه آنها تعداد گره های هر شبکه می باشد. این کار باعث می شود که مختصات گره های اصلی شبکه دچار تغییر نگردد.

1. **ذخیره مقادیر در یک ماتریس جهت انتقال به شبکه 2**

در این قسمت مقادیری که قرار است از شبکه 1 به شبکه 2 منتقل شوند در آرایه های FUNC ذخیره می شوند.

1. صفر کردن مقادیر ماتریس ها

ماتریس هایی که برای انتقال مقادیر WNP1 , RES مورد استفاده قرار می گیرند، مقدار صفر در آن ها ذخیره می شود این کار زمانی که چند مرحله درشت سازی داشته باشیم، باید انجام گیرد.

1. در روش مالتی گرید می توان از چند شبکه متوالی استفاده کرد در این قسمت با استفاده از متغیر NOL برنامه متوجه خواهد شد که انتقال بین کدام یک از دو شبکه متوالی صورت خواهد گرفت. برای مثال اگر NOL=1 باشد بدان معناست که انتقال داده ها بین شبکه 1 و2 انجام می پذیرد. ذخیره عدد 2 در این متغیر به معنی بررسی شبکه های 2 و 3 می باشد. همچنین متغیر NMF در این قسمت نشانگر این است که کدام شبکه قرار است فراخوانی شود برای مثال NMF=1 نشان دهنده فراخوانی شبکه ذخیره شده در فایل Mesh1.txt می باشد. متغیر استفاده شده دیگر در این قسمت FTC می باشد که اگر مقدار آن یک باشد به معنی انتقال داده ها از شبکه ریز به درشت و در صورت صفر بودن به معنی انتقال داده ها از شبکه درشت به ریز است.
2. فراخوانی شبکه پذیرنده مقادیر میدان

در اینجا با توجه به مقدار NMF در قسمت قبل شبکه ای که قرار است درون یابی روی آن انجام شود و مقادیر را از شبکه قبلی دریافت کند فراخوانی خواهد شد.

1. شماره گذاری مجدد اضلاع برای اعمال شرایط مرزی

با فراخوانی زیربرنامه MeshBC اضلاع غیر مرزی به ابتدای آرایه مربوط به ذخیره اطلاعات اضلاع تشکیل دهنده شبکه منتقل شده و همچنین سایر نواحی شبکه متناسب با شرایط مرزی مربوطه شماره گذاری مجدد می گردد.

1. **ذخیره شماره گره ها**

با فراخوانی زیربرنامه Edge\_To\_Cell شماره گره های مربوط به هر مثلث به صورت پادساعتگرد برای شبکه 2 در یک آرایه ذخیره می شود.

1. **ذخیره شماره گره ها در آرایه ای دیگر**

چون در ادامه زیربرنامه Edge\_To\_Cell فراخوانی شده در قسمت 7، برای شبکه 1 هم فراخوانی خواهد شد، در اینجا اطلاعات مربوط به گره های مثلث شبکه 2 در ماتریس دیگر ذخیره می شود.

1. **مکان یابی اضلاع شبکه 1**

در این قسمت اضلاع شبکه 1 در یک حلقه روی مثلث های شبکه 2 جستجو می شوند تا مکان آن ها در این شبکه مشخص شود.

1. **تعریف سه شمارنده با کاربردهای خاص**
2. counter5 : ممکن است ضلعی از شبکه ریز باشد که گره ابتدایی آن در هیچ مثلث از شبکه 2 قرار نگیرد این شمارنده با داشتن مقدار صفر گویای این قضیه می باشد و در صورتی که مقدار آن یک شود نشان دهنده این است که مکان آن در یکی از مثلث های شبکه 2 می باشد.
3. counter6 : اگر گره ای ابتدایی ضلعی از شبکه ریز در هیچ مثلثی از شبکه درشت نباشد در این صورت برنامه جستجو را بر اساس گره انتهایی آن ضلع انجام می دهد در این صورت این شمارنده مقدار عدد 1 را به جهت تغییر در انجام محاسبات در خود ذخیره می کند در غیر این صورت مقدار صفر برای آن ثابت می ماند.
4. counter13 : در صورتی که ضلعی از شبکه 1 دقیقا روی ضلعی از شبکه 2 قرار گیرد مقدار این شمارنده 1 و در غیر این صورت صفر باقی می ماند.
5. **ذخیره سلول ها و نقاط یک ضلع در پارامترهای محلی**

سلول های چپ و راست و نقاط ابتدایی و انتهایی ضلع شبکه 1 در پارامترهای محلی ذخیره می شوند.

1. **حلقه مربوط به سلول های شبکه 2**

حلقه ای جهت شمارش مثلث های شبکه 2 برای هر ضلع شبکه 1 جهت اینکه هر ضلع در کدام مثلث قرار دارد.

1. **ذخیره گره های مثلث تحت بررسی در ماتریس محلی**

برای راحتی سه گره مثلث شبکه 2 جهت مکان یابی اضلاع شبکه 1 در ماتریس P\_1 ذخیره شده است.

1. **بررسی مکان گره ابتدایی ضلع شبکه 1**

در این قسمت مقادیر F1 , F2 , F3 برای گره ابتدایی ضلع شبکه 1 جهت اینکه در کدام مثلث از شبکه 2 قرار دارد، محاسبه می شود.

1. **صفر کردن مقادیر کوچک بدلیل عدم محاسبه دقیق توسط نرم افزار**

ممکن است در محاسبات قسمت 14 مقدار یکی از پارامترها صفر شود ولی بخاطر اینکه نرم افزار قادر به نمایش صفر مطلق نمی باشد آن را با عدد کوچکی تخمین می زند. در اینجا بیشترین عددی که نرم افزار جهت نشان دادن عدد صفر استفاده می کند فرض شده است.

1. **بررسی شرط قرار گرفتن گره در یک مثلث**

در صورتی که هر سه پارامتر F1 , F2 , F3 همزمان بزرگتر مساوی صفر و یا کوچکتر مساوی صفر باشد گره درون مثلث یا روی مثلث قرار دارد در غیر این صورت به دنبال مثلث بعدی می رویم. در حالتی که برای هیچ یک از مثلث ها شرط فوق برقرار نشود نرم افزار به قسمت 30 رفته و جستجو را بر اساس گره انتهایی ضلع مورد بررسی انجام می دهد. ولی در صورت برقراری شرط برنامه وارد شرط if شده و مقدار counter5، 1 می شود. در این صورت با 4 حالت زیر مواجه هستیم :

1. ضلع شبکه 1 دقیقا روی یکی اضلاع مثلث جاری از شبکه 2 قرار داشته باشد.
2. گره انتهایی داخل مثلث قرار داشته باشد.
3. گره انتهایی درون این مثلث نباشد بلکه درون مثلث دیگری قرار داشته باشد که در این صورت ممکن است ضلع شبکه 1 با یکی از اضلاع مثلث شبکه 2 تقاطع داشته باشد.
4. ممکن است حالت 3 برقرار باشد ولی ضلع شبکه 1 با هیچ یک از اضلاع مثلث شبکه 2 تقاطع نداشته باشد که در این صورت به سراغ مثلث بعدی میرویم.
5. **بررسی مکان گره انتهایی ضلع شبکه 1**

در این قسمت مقادیر F\_p1 , F\_p2 , F\_p3 برای گره انتهایی ضلع شبکه 1 محاسبه می شود.

1. همانند قسمت 15 می باشد.
2. **بررسی قرار گرفتن ضلع شبکه 1 روی اضلاع مثلث جاری از شبکه 2**

مقادیر محاسبه شده در قسمت 17 یکی از 4 حالت ذکر شده در قسمت 16 را به ما نشان می دهد. در اینجا در صورتی که شرط آورده شده برقرار باشد یعنی ضلع شبکه 1 دقیقا روی یکی از اضلاع مثلث شبکه 2 قرار دارد و با افزایش مقدار شمارنده به منزله پی بردن به مکان گره انتهایی بدون هیچ محاسبه ای از برنامه خارج می شویم.

1. **بررسی قرار گرفتن کامل ضلع شبکه 1 درون مثلث جاری از شبکه 2**

در صورتی که شرط قسمت 19 برقرار نباشد برنامه شرط این قسمت را چک می کند. برقراری این شرط به معنی این است که حالت 2 در قسمت 16 اتفاق افتاده و ضلع شبکه 1 به طور کامل درون مثلث شبکه 2 واقع شده است. در اینجا همانند قبل بدلیل پی بردن به مکان گره انتهایی شمارنده عدد 1 را در خود ذخیره می کند. محاسبات مربوطه با توجه به اینکه ضلع شبکه 1 جز اضلاع مرزی می باشد یا خیر انجام می پذیرد و پس از انجام محاسبات از حلقه خارج شده و به دنبال ضلع بعدی می رویم.

1. **بررسی محل برخورد ضلع شبکه 1 با اضلاع مثلث جاری از شبکه 2**

در صورتی که شرط های 19 و 20 برقرار نباشد احتمال این می رود که ضلع با یکی از اضلاع مثلث شبکه 2 که گره ابتدایی درون یا روی آن واقع شده است، برخورد داشته باشد. در این صورت باید به دنبال محل برخورد باشیم. برای این منظور باید شیب ضلع شبکه 1 و سه ضلع مثلث شبکه 2 را بدست آورده و با استفاده از معادله خط آنها، محل تلاقی را در صورتی که درون مثلث قرار گرفته، بدست آورد. باید به این نکته توجه کرد که این روش تلاقی حاصل از امتداد ضلع ها را هم می دهد ولی محل برخورد مورد نظر برای ما می بایست از برخورد خود اضلاع (نه امتدادشان) بوده و درون مثلث باشد.

1. **محاسبه شیب ضلع شبکه 1**

ممکن است ضلع شبکه ریز عمودی بوده و دارای شیب بی نهایت باشد این قسمت این موضوع را مورد بررسی قرار می دهد در صورتی مختصه افقی دو گره ابتدایی و انتهایی مساوی باشد ضلع دارای شیب بی نهایت بوده و محل برخورد در همین مختصه اتفاق می افتد بنابراین مختصه x محل برخورد بدین صورت پیدا می شود. که این موضوع با ذخیره عدد یک در counter10 همراه است که در ادامه از آن استفاده خواهد شد. در صورتی که شرط برقرار نباشد شیب این ضلع با استفاده از نقاط ابتدا و انتها محاسبه خواهد شد.

1. **محاسبه شیب اضلاع شبکه 2**

حال باید شیب مربوط به اضلاع مثلث درشت را محاسبه کنیم. برای اینکار از یک حلقه استفاده کرده که در صورت یافتن محل برخورد نیاز به چک کردن بقیه اضلاع نمی باشد.

1. در این قسمت از پارامترهای محلی جهت ذخیره نقاط ابتدایی و انتهایی هر ضلع مثلث درشت استفاده می شود.
2. **بررسی شیب بی نهایت اضلاع**

در این قسمت نیز همانند 22 ابتدا چک می شود که شیب ضلع بی نهایت نباشد. در صورت برقراری شرط، ذخیره عدد 1 در counter11 نشان دهنده آن است. در این حالت مختصه x برخورد دو ضلع مشخص خواهد شد. در صورتی که شیب بی نهایت نباشد، به دنبال محاسبه آن می رویم.

1. **محاسبه مختصات محل برخورد**

بعد از محاسبات فوق حالت هایی وجود دارد که به بررسی آن ها می پردازیم.

1. در صورتی که یکی از دو ضلع عمودی باشد یا counter10 مساوی یک خواهد بود یا counter11. از آنجایی که در صورت وقوع یکی از این حالت ها مختصه x برخورد مشخص می شود با استفاده معادله مختصه y بدست آمده برای ضلع غیرعمود، مکان عمودی برخورد را نیز می یابیم.
2. در صورتی که هر دوضلع عمودی و یا اینکه دارای شیب یکسانی باشند، هیچ گاه باهم برخورد نخواهند داشت و به دنبال ضلع دیگر مثلث جاری از شبکه 2 می رویم.
3. در حالت سوم ممکن است هیچ کدام از دو ضلع مورد بررسی عمودی نبوده و همچنین شیب یکسانی هم نداشته باشند. در این صورت با استفاده از روابط داده شده مختصات محل برخورد دو ضلع بدست خواهد آمد.
4. **محاسبات مورد نیاز جهت تأیید مختصات بدست آمده به عنوان محل برخورد دو ضلع**

در این قسمت برای اینکه مختصات بدست آمده به عنوان محل برخورد، مورد قبول واقع شوند شرایطی را بررسی می کنیم.

1. محاسبات این قسمت همانطور که در قبل هم داشتیم برای فهمیدن اینکه آیا محل برخورد در درون مثلث درشت قرار دارد یا خیر، انجام می گیرد.
2. این قسمت همانند قسمت 15 می باشد.
3. محاسبات مربوط در این جا به منظور فهمیدن این موضوع است که محل برخورد حاصل امتداد اضلاع نمی باشد.
4. همانند قسمت 15 می باشد.
5. **شرط لازم جهت پذیرفتن مختصات بدست آمده به عنوان محل برخورد**

در صورتی که شرط برقرار باشد مختصات برخورد به عنوان محل برخورد دو ضلع مورد قبول واقع می شود.

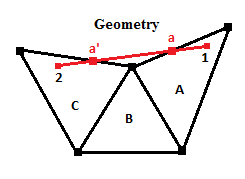
1. ممکن است نقطه ابتدایی تا محل برخورد روی یکی از اضلاع مثلث واقع شود که در این صورت بدون انجام محاسبات با قرار دادن مختصات محل برخورد به عنوان نقطه ابتدایی به دنبال محل قرار گرفتن بقیه ضلع در سایر مثلث ها می رویم. در اینجا ذکر این نکته قابل اهمیت است که محل برخورد به عنوان یک گره فرضی در نظر می گیریم و شماره گره مربوط به آن برابر با تعداد کل گره های شبکه 1 به اضافه یک می باشد تا مختصات بقیه گره ها دچار تغییر نشود.
2. در صورتی که حالت فوق برقرار نباشد، برنامه وارد این قسمت شده و محاسبات را با توجه به اینکه جستجو براساس گره ابتدایی (counter6=0) بوده یا گره انتهایی (counter6=1) ضلع، انجام می دهد، در این جا هم همانند قبل محاسبات به دو بخش اضلاع مرزی و غیر مرزی تقسیم می شود. در آخر این قسمت همانند حالت فوق مختصات محل برخورد به عنوان یک گره ابتدایی فرضی در نظر گرفته می شود تا مکان یابی برای بقیه ضلع مورد نظر ادامه یابد(از محل برخورد تا انتهای ضلع).
3. **عدم حضور گره ابتدایی در مثلث جاری**

در این قسمت در صورتی که نقطه ابتدایی ضلع مورد نظر در مثلث شبکه 2 نباشد برنامه بجای وارد شدن در قسمت 16 وارد این قسمت شده و هیچ یک از مراحل قبل اجرا نخواهد شد. در نتیجه برنامه به دنبال مثلث دیگر از شبکه 2 می رود تا اینکه مکان نقطه ابتدایی از ضلع مورد نظر یافت شود.

1. **جستجو بر اساس گره انتهایی**

اگر در قسمت 29، برنامه جایگاهی برای گره ابتدایی نیابد (صفر بودن counter5 به همین معنی می باشد) برنامه وارد این قسمت شده تا جستجو را بر اساس گره انتهایی انجام دهد. برای اینکه متوجه شویم برنامه جستجو را بر اساس گره انتهایی انجام می دهد عدد یک را وقتی که برنامه وارد این قسمت می شود در counter6 ذخیره می کنیم تا محاسبات مربوط به آن جهت درون یابی با یک علامت منفی انجام شود. ذکر این نکته حائز اهمیت است که در صورتی مکانی برای گره انتهایی هم پیدا نشود دیگر برنامه وارد این قسمت نخواهد شد.

حالتی دیگری نیز وجود دارد که ممکن است برنامه وارد این قسمت شود. ممکن است مکان گره ابتدایی پیدا شود () ولی حالت زیر پیش بیاید.



شکل (2) حالت خاص ایجاد شده در قرار گرفتن دو شبکه روی هم

به این صورت که مکان گره 1 که گره ابتدایی است پیدا شود و برنامه محل برخورد را در a بیابد بعد از آن طبق کد، نقطه a به عنوان گره ابتدایی فرضی انتخاب می شود تا بقیه ضلع مکان یابی شود. از آنجایی که برنامه گره a را در مثلث A می یابد و انتظار می رود که بقیه ضلع با یکی از سه ضلع همین مثلث برخورد داشته باشد طبق شکل فوق این اتفاق نخواهد افتاد و در نتیجه برنامه ضلع بعدی را مورد بررسی قرار می دهد و در نتیجه تاثیر قسمت 2a’ در نظر گرفته نمی شود. برای جلوگیری از این مشکل شمارنده counter13 تعریف شده و در صورتی که مقدار آن صفر بماند برنامه وارد این قسمت یعنی قسمت 30 شده و جستجو را بر اساس گره انتهایی انجام می دهد.

1. در اینجا هم در صورتی که برنامه به دلیل مشکل فوق وارد قسمت 30 شود مقدار 1 در counter13 تا از ورود مجدد به این قسمت جلوگیری شود.
2. این قسمت همانند قسمت 4 می باشد با این تفاوت که در قسمت 4 تاثیر اضلاع شبکه 1 را جهت درون یابی مورد بررسی قرار دادیم و دنبال مکان آن ها در شبکه 2 بودیم. حال در این قسمت به دنبال تاثیر اضلاع شبکه 2 در درون یابی می باشیم و باید مکان آنها را در مثلث های شبکه 1 بیابیم. بدین منظور با توجه به مرحله ای که در کد اصلی قرار داریم با انتخاب مناسب NMF به دنبال ذخیره اطلاعات گره های تشکیل دهنده هر مثلث برای شبکه 1 می باشیم.
3. **فراخوانی شبکه حاوی مقادیر معلوم میدان**

در این قسمت با توجه به مقدار NMF شبکه 1 فراخوانی می شود.

1. همانند قسمت 6 می باشد.
2. همانند قسمت 7 می باشد.
3. گره های هر مثلث شبکه 1 در یک ماتریس جداگانه ذخیره می شود.
4. پاضلاع شبکه 2 روی مثلث های شبکه 1 جستجو می شوند تا مکان آنها مشخص شود.
5. شمارنده های این قسمت همانند قسمت 10 می باشد. شمارنده جدید تعریف شده در این قسمت counter3 می باشد. در صورتی که یکی از اضلاع شبکه 2 روی اضلاع شبکه 1 قرار گیرد برای انجام محاسبات مربوطه برای سلول های چپ و راست آن نیازمند مقادیری از شبکه 1 که این سلول ها در آن قرار دارند می باشیم. یعنی اگر به همچین اضلاعی برخورد کنیم باید بفهمیم سلول ME و NE این ضلع در کدام سلول شبکه 1 قرار دارند و از مقادیر میدان مربوط به آن برای انجام محاسبات و در نتیجه انتقال داده ها استفاده نماییم.
6. همانند قسمت 11، اینبار برای ضلع شبکه 2 انجام می شود.
7. همانند قسمت 12، اینبار برای سلول های شبکه 1 انجام می گیرد.
8. همانند قسمت 13 می باشد.
9. همانند قسمت 14 ولی برای گره ابتدایی ضلع شبکه 2 محاسبات انجام می شود.
10. همانند قسمت 15 می باشد.
11. **بررسی شرط قرار گرفتن گره در یک مثلث**

همانند قسمت 16 ابتدا به دنبال مکان گره ابتدایی ضلع شبکه 2 در یکی از مثلث های شبکه 1 هستیم. در صورتی که این اتفاق نیفتد برنامه جستجو را بر اساس گره انتهایی همین ضلع شروع خواهد کرد. در صورت یافتن مکان گره ابتدایی شمارنده 15 عدد یک را به خود اختصاص می دهد.

1. **بررسی مکان گره انتهایی ضلع شبکه 2**

همانند قبل در این قسمت به دنبال موقعیت گره انتهایی ضلع مورد بررسی هستیم. که حالت های ذکر شده در قسمت 16 نیز در اینجا برقرار است.

1. همانند قسمت 15 می باشد.
2. **بررسی قرار گرفتن ضلع شبکه2 روی اضلاع مثلث جاری از شبکه 1**

در این قسمت به بررسی این موضوع که دو ضلع از شبکه 2 و 1 روی هم قرار دارند، پرداخته شده است. در قسمت اول کد با بررسی این موضوع، برنامه بدون محاسبه از حلقه خارج شده و جستجو برای ضلع بعدی انجام می گرفت. ولی در اینجا محاسبات ابتدا صورت گرفته و سپس از حلقه خارج می شویم. در کل برای این قسمت محاسبات را برای اضلاعی انجام می دهیم که مقادیر میدان در شبکه مربوط به آنها مجهول است. مسئله ای که در اینجا حائز اهمیت است اینست که اینگونه از اضلاع برای دو سلول از شبکه 1 مشترک هستند و در نتیجه در محاسبات می بایست از یکی از مقادیر استفاده کرد. راه حلی که در این قسمت بکار برده شده است اینست که برای سلول سمت چپ این ضلع از شبکه 2، از مقادیر میدان سلولی از شبکه 1 استفاده شود که با آن مساحت مشترک دارد و برای سلول سمت راست نیز به همین منوال. به این نکته باید توجه کرد که اگر ضلع مورد بررسی جز اضلاع مرزی نباشد ضلع مشترک دو سلول و اگر جز اضلاع مرزی باشد فقط برای یک سلول می باشد. شمارنده 3 نیز جهت یافتن سلول هایی که این ضلع در آن مشترک است، تعریف شده است.

1. در این قسمت محاسبات لازم جهت پی بردن اینکه سلول شبکه 1 در سمت راست یا چپ ضلع مورد بررسی قرار دارد، انجام می شود. چنانچه سلول در سمت چپ واقع شده باشد مقدار F\_h>0 بوده و محاسبات را برای سلول سمت چپ شبکه 2 انجام می دهیم در غیر این صورت محاسبات را برای سلول سمت راست انجام می دهیم.
2. همانند قسمت 15 می باشد.
3. سومین گره سلولی که ضلع مشترک با این ضلع از شبکه 2 دارد، مورد بررسی قرار می گیرد تا به قرار گرفتن این گره در سمت چپ و یا راست این ضلع پی برده شود. بنابراین در این قسمت شرط هایی قرار داده شده تا گره سوم سلول شبکه 1 مشخص شود که این امر مستلزم آن است که با ضلع مورد بررسی گره مشترک نداشته و در امتداد این ضلع هم قرار نگرفته باشد.
4. در صورتی که ضلع مورد بررسی جز اضلاع مرزی نباشد در این قسمت به ادامه جستجو پرداخته تا سلول بعدی که با این ضلع از شبکه 2، ضلع مشترک دارد، پیدا شود و محاسبات با استفاده از مقدار آن برای سلول سمت راست یا چپ (با توجه به مقدار F\_h) ضلع مورد بررسی انجام گیرد.
5. **بررسی قرار گرفتن کامل ضلع شبکه 2 درون مثلث جاری از شبکه 1**

در اینجا همانند قبل، به بررسی وجود گره انتهایی ضلع، در سلولی که گره ابتدایی در آن حضور دارد، پرداخته می شود. سپس در صورت وجود آن، محاسبات با توجه مرزی و غیرمرزی بودن ضلع انجام گرفته و در نهایت از حلقه خارج شده و ضلع بعدی را مورد بررسی قرار می دهد.

1. **بررسی محل برخورد ضلع شبکه 2 با اضلاع مثلث جاری از شبکه 1**

این قسمت مربوط به پیدا کردن تلاقی بین ضلع شبکه 2 و مثلث شبکه 1 می باشد. که توضیحات مربوطه در قسمت اول کد به طور کامل آورده شده است با این تفاوت که در اینجا تلاقی بین ضلع شبکه 2 و مثلث شبکه 1 انجام می گیرد.(مراحل 21 تا 28)

1. همانند قسمت 29 می باشد.
2. این قسمت همانند بخش 30 می باشد.
3. **ذخیره مقادیر میدان در ماتریس های جداگانه**

در این قسمت مقادیر میدان روی شبکه ای که مشخص است یعنی شبکه 1، در ماتریس های جداگانه ای ذخیره شده تا در مراحل بعدی از آن استفاده شود.

1. در این قسمت با توجه به اینکه در کدام مرحله از روش قرار داریم شبکه مورد نظر جهت درون یابی انتخاب شده تا مقادیر میدان را روی آن بدست آوریم.
2. شبکه مورد نظر فراخوانی می شود.
3. همانند قسمت 6 می باشد.
4. **محاسبه مساحت سلول ها و مختصات مرکز آنها**

با فراخوانی زیربرنامه GeoCal2Dمساحت، بردارهای عمود با بعد، طول اضلاع و مختصات مرکز هر کدام از سلول های شبکه محاسبه شده و در آرایه های مربوطه ذخیره می­شود.

1. **محاسبه مقادیر میدان در شبکه 2**

مقادیر میدان در شبکه مجهول بدست می آید.

1. **استفاده از روش تزریق مستقیم در درون یابی**

در این قسمت سلول های دو شبکه مورد بررسی قرار می گیرند در صورتی که دو سلول کاملا روی هم قرار گرفته باشند مقدار میدان در سلول معلوم عینا در سلول مجهول قرار می گیرد. در این قسمت برای پی بردن به اینکه دو سلول کاملا روی هم قرار دارند مختصات آنها مورد بررسی قرار گرفته است. در صورتی که یکی از شرط ها برقرار باشد به منزله قرار گرفتن دو سلول بر روی هم می باشد که این موضوع با مقدار دهی غیر صفر به شمارنده 9 مشخص می شود.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interpolation** | | | |
| **Dimension** | **Type** | **Description** | **Parameters** |
|  | Real(8) | The slop of Mesh1 faces | M1 |
|  | Real(8) | The slop of Mesh2 faces | M2 |
|  | Real(8) | X-coordinate collision of two face | collisionX |
|  | Real(8) | Y-coordinate collision of two face | collisionY |
| (1:3) | Real(8) | The slop of Mesh1 cell faces | M\_1 |
| (1:3) | Real(8) | The slop of Mesh2 cell faces | M\_2 |
| (1:Dim) | Real(8) | X-coordinate of nodes in Mesh1 | X1 |
| (1:Dim) | Real(8) | Y-coordinate of nodes in Mesh1 | Y1 |
| (1:Dim) | Real(8) | X-coordinate of nodes in Mesh2 | X2 |
| (1:Dim) | Real(8) | Y-coordinate of nodes in Mesh2 | Y2 |
| (1:4,1:Dim) | Real(8) | Number of nodes in a cell | Corn |
| (1:3,1:Dim) | Real(8) | Number of nodes in a cell of Mesh1 | Corn1 |
| (1:3,1:Dim) | Real(8) | Number of nodes in a cell of Mesh2 | Corn2 |
| (1:4,1:Dim) | Real(8) | The array will be interpolated | FUNC1 |
| (1:4,1:Dim) | Real(8) | The array will be interpolated | FUNC2 |
| (1:4,1:Dim) | Real(8) | The array will be interpolated | FUNC3 |
| (1:4,1:Dim) | Real(8) | The value of FUNC1 before interpolation | PFUNC1 |
| (1:4,1:Dim) | Real(8) | The value of FUNC2 before interpolation | PFUNC2 |
| (1:4,1:Dim) | Real(8) | The value of FUNC3 before interpolation | PFUNC3 |