

**زیربرنامه Coarsening**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Coarsening** | | | |
| **Dimension** | **Type** | **Description** | **Parameters** |
|  | Integer | Maximum **Dim**ension of Arrays | Dim |
|  | Real(8) | Coarsening Coefficient | C |
|  | Real(8) | The criteria for putting Nodes together in coarse mesh | Beta |
|  | Real(8) | Angle criteria for apexes | Teta0 |
| (1:4,1:Dim) | Real(8) | Conservative Values at (N+1)th Time Step | WNP1 |
| (1:Np,1:Np) | Real(8) | **D**istance of Two Nodes | d |
| (1:Dim) | Real(8) | **N**earest **N**eighbor of Node **P** | NNp |
| (1:Dim) | Real(8) | The Initial Coarsening Function of Node P | F0 |
| (1:Dim) | Real(8) | Coarsen Value of Initial Coarsening Function | F0\_C |
| (1:Dim) | Real(8) | The Final Coarsening Function of Node P | F1 |
| (1:Dim) | Real(8) | An array to store the number of nodes should be in coarse mesh (**N**umber **O**f **C**oarse **P**oint) | NOCP |
| (1:Dim) | Real(8) | An array to store the number of nodes should be in MIS algorithm | Point1 |
| (1:Dim) | Real(8) | An array to store the number of nodes should be in MIS algorithm | Point2 |
| (1:Dim) | Real(8) | An array to store the number of nodes can be in coarse mesh but not necessarily | Cp1 |
| (1:Dim) | Real(8) | An array to store the number of nodes can be in coarse mesh but not necessarily | Cp2 |
| (1:Dim) | Real(8) | An array to store the number of nodes of Cp1 and Cp2 arrays that should be in coarse mesh | Coarsepoint |
| (1:Dim) | Real(8) | The face not should be in coarse mesh (**D**eleted **F**ace) | DelFace |
| (1:Dim) | Real(8) | The Nodes not should be in coarse mesh (**N**ot **I**n **C**oarse **M**esh) | NICM |
| (1:Dim) | Real(8) | The nodes of boundary should be in coarse mesh (**B**oundary **S**egment) | BS |

* 1. **وظایف**

این زیربرنامه نقاط قابل حذف جهت درشت سازی یک شبکه را تولید می کند. از این رو اطلاعات شبکه مورد نظر از طریق فایل Mesh.txt از کاربر گرفته می شود.

* 1. **توضیحات و تئوری ها**

برای درشت سازی یک شبکه روش های متعددی وجود دارد که یکی از این روش ها، روش Function based coarsening می باشد. هدف این روش تولید شبکه ای درشت می باشد که گره های آن مطابق با یک تابع فاصله (Spacing function) در کنار هم قرار گرفته باشند. بنابراین گام اول در این روش محاسبه تابع فاصله برای تمامی گره های شبکه می باشد. برای محاسبه این مقدار ابتدا باید برای هر گره شبکه فاصله نزدیک ترین گره به آن () محاسبه شود. بعد از محاسبه این مقدار برای تمام گره ها، تابع فاصله اولیه به صورت زیر بدست می آید :

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

*در مرحله بعد باید این مقدار برای تمامی گره ها درشت شده، و در نهایت تابع فاصله نهایی را بدست آوریم که به صورت رابطه زیر انجام می شود:*

|  |  |
| --- | --- |
| **(2)** |  |
| **(3)** |  |

که در روابط فوق تابع فاصله نهایی، C ضریب درشت سازی و فاصله بین دو گره و می باشد. مقدار C با توجه به تعداد مراحل درشت سازی به صورت می باشد که () i نشان دهنده مرحله درشت سازی است.

بعد از محاسبه تابع فاصله جدید، باید یک زیر مجموعه از گره های شبکه که مطابق با این تابع فاصله هستند، انتخاب شود. برای این کار از conflict graph استفاده می شود. در این گراف دو گره از لحاظ فیزیکی در مقایسه با مقیاس طول تعریف شده برای آن دو، خیلی نزدیک بهم می باشند. این مقیاس طولی به صورت زیر می باشد:

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

گره هایی که معیار فوق برای آن ها رعایت نشده باشد، اصطلاحأ گفته می شود که این مجموعه گره ها، می باشد یعنی دایره هایی به شعاع به ترتیب با مرکز های ملزمند که با هم تداخل نداشته باشند.

بعد تشکیل conflict graph از مجموعه گره های شبکه، باید یک MIS از این گراف تشکیل دهیم. الگوریتم MIS سبب حذف گره هایی می شود که با توجه به معیار طولی تعریف شده، در کنار هم قرار داشتند. این الگوریتم به این گونه عمل می کند که تمامی همسایگان یک گره از گراف مورد نظر، در شبکه درشت وجود نخواهند داشت. بنابراین سایر گره هایی باقی مانده، به صورت در کنار هم قرار خواهند داشت، که گره های شبکه درشت را تشکیل می دهند. حال این مجموعه گره ها با هر روش شبکه بندی قابل مثلث بندی مجدد می باشد.

نکته ای در رابطه با درشت سازی باندری:

در مرحله حذف کردن گره، گره هایی که روی باندری قرار دارند می توانند حذف شوند اما به طوری که خود باندری تغییری زیادی نکند که با باندری اولیه تفاوت داشته باشد. ولی در طول مراحل درشت سازی ممکن به مرحله ای برسیم که باندری با باندری اولیه تفاوت زیادی داشته باشد به همین دلیل می توان از محدودیت هایی استفاده کرد تا پروفیل باندری با باندری اولیه تفاوت چندانی نداشته باشد که در صورت استفاده از این محدودیت ها روش function based coarsening خود را به راحتی با آن وفق می دهد. نقاطی که apex نامیده می شوند نباید از باندری حذف شوند. apex ها نقاطی می باشند که در قسمت های تیز باندری قرار دارند مانند راس های مربع. به طور کلی راس هایی apex نامیده می شوند که زاویه ی دو ضلع برخورد کننده در این راس ها بیشتر از 20 درجه باشد. در این صورت این راس ها نباید از باندری حذف شوند.(این زاویه قابل تغییر است)

مراحل فوق به صورت دقیق تر در زیر توضیح داده شده است.

* 1. **بخش های زیربرنامه**

در این قسمت تمام بخش های زیربرنامه مطابق با شماره گذاری موجود در برنامه کامپیوتری ارائه شده است.

1. باز کردن فایل حاوی شبکه و ذخیره نتایح

در این زیربرنامه دو فایل به نام های Mesh.txt و Eliminating nodes.txt به ترتیب برای فراخوانی اطلاعات شبکه و ذخیره گره های قابل حذف، ایجاد می شود.

1. مقداردهی به برخی پارامترها

در این قسمت C میزان درشت سازی شبکه را مشخص می کند همچنین Beta معیاری جهت قرار گیری گره های شبکه درشت با یک فاصله مشخص از هم می باشد که در قسمت تئوری و توضیحات، شرح داده شده است. teta0 زاویه معیار (بر حسب درجه) جهت انتخاب گره های شبکه به عنوان نقاط تیز را مشخص می کند.

1. خواندن فایل شبکه

جهت درشت سازی، اطلاعات شبکه مورد نظر، ذخیر شده در فایل Mesh.txt، از طریق زیربرنامه Read\_2DMesh فراخوانی می شود.

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. **محاسبه فاصله نزدیک ترین همسایه، تابع فاصله اولیه و تابع فاصله درشت شده**

طبق الگوریتم این روش، ابتدا باید فاصله نزدیک ترین گره به هر یک از گره های شبکه () محاسبه شده و سپس با استفاده از آن تابع فاصله اولیه () برای هر کدام از گره ها محاسبه گردد. در نهایت تابع فاصله جدید () با استفاده از درشت سازی تابع فاصله اولیه بدست می آید.

1. **مقداردهی اولیه به و**

جهت محاسبه فاصله نزدیک ترین همسایه و تابع فاصله، ابتدا این دو پارامتر برای تمامی گره ها مقداردهی می شوند.

1. **محاسبه**

ابتدا فاصله هر گره با سایر گره های شبکه را بدست آورده و سپس کمترین مقدار (نزدیک ترین گره به گره مورد بررسی) را برای قرار می دهیم.

1. در این قسمت با استفاده از رابطه زیر فاصله بین دو گره محاسبه می شود :

|  |  |
| --- | --- |
| **(5)** |  |

1. کمترین مقدار برای هر گره در این قسمت محاسبه شده که مقدار گره مورد نظر می باشد.
2. **محاسبه**

بعد از محاسبه فاصله نزدیک ترین همسایه برای هر گره ، تابع فاصله اولیه را که نصف این مقدار می باشد، برای همه گره ها محاسبه می کنیم.

1. **محاسبه**

با استفاده از رابطه زیر مقدار تابع فاصله جدید را برای هر گره بدست می آوریم :

|  |  |
| --- | --- |
| **(6)** |  |

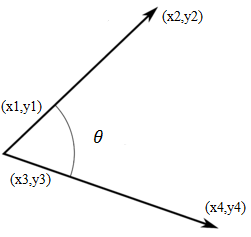
1. کمترین مقدار نتیجه فوق برای هر گره تابع فاصله آن را نتیجه می دهد که در این قسمت محاسبه شده است.

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

1. **گره های مرزی که نباید حذف شوند**

برخی گره هایی روی مرز هستند که با حذف آن ها هندسه مسئله دچار تغییر قابل توجهی می شود بنابراین نباید این گره ها از روی مرز حذف شوند. برای این منظور اضلاع مرزی که باهم زاویه ای بیشتر از می سازند گره مشترک بین آن ها نباید حذف شود. در این قسمت این نوع گره ها مورد بررسی قرار گرفته است. برنامه جهت بررسی اضلاعی که روی باندری هستند با توجه به این که کدام گره ها آن ها با هم مشترک می باشد وارد یکی از قسمت های 1 یا 2 شده و زاویه دو ضلع را در این گره محاسبه می کند که روابط آن به صورت زیر می باشد.

هر ضلع را به صورت یک بردار در نظر گرفته و با استفاده از نقاط ابتدایی و انتهایی هر ضلع، زاویه بین آن ها را به صورت زیر بدست می آوریم:



در نهایت زاویه بین دو بردار فوق به صورت زیر بیان می شود:

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

در صورتی که مقدار فوق از زاویه معیار تعیین شده برای نقاط تیز بیشتر باشد، گره مورد نظر باید در شبکه درشت حضور داشته باشد تا از تغییر شکل باندری جلوگیری شود.

1. **مشخص کردن گره های مرزی برای قرار گرفتن در الگوریتم** MIS

اگر دو گره همسایه با توجه معیار گفته شده در الگوریتم Function based coarsening در کنار هم قرار داشته باشند، (یعنی از یک حدی بهم نزدیکتر باشند) از حضور آن دو گره به طور همزمان در شبکه درشت خودداری می شود. در صورتی که یک گره این معیار فاصله را با تمام گره های همسایه خود حفظ کند، (یعنی فاصله آن گره با تمام گره های اطرافش از یک حد بهم نزدیکتر نباشد) از آن گره به عنوان گره شبکه درشت استفاده شده و نباید حذف شود. این معیار فاصله به صورت زیر می باشد :

|  |  |
| --- | --- |
| **(9)** |  |

در این قسمت بدلیل اینکه شکل هندسه تغییر نکند ابتدا گره های مرزی مورد بررسی قرار گرفته و حضور آن ها در شبکه درشت بررسی می شود و سپس به دنبال گره های غیر مرزی خواهیم رفت. دو شمارنده MIS و Coarse برای مشخص شدن تعداد گره های قرار گرفته در الگوریتم MIS و همچنین تعداد گره هایی محتمل جهت قرار گرفتن در شبکه درشت مقدار دهی اولیه می شوند.

1. در صورتی که یکی از گره های ضلع مرزی جز گره های نقاط تیز باشد (یعنی قرار گرفتن حتمی در شبکه درشت) به سراغ ضلع بعدی می رویم و معیار فاصله را برای این ضلع بررسی می کنیم.
2. در صورتی که نامساوی (9) برقرار نباشد برنامه وارد این قسمت شده و گره های ضلع مورد بررسی را جهت قرار گرفتن در الگوریتم MIS در یک آرایه ذخیره می کند.
3. در صورتی که مورد 2 برقرار نباشد برنامه گره های ضلع را به عنوان گره های محتمل شبکه درشت در یک آرایه ذخیره می کند. از این رو محتمل گفته می شود چون ممکن گره ذخیره شده در این قسمت برای ضلع دیگر در الگوریتم MIS قرار گرفته و از شبکه حذف شود.
4. **تأیید گره ها جهت قرار گرفتن در شبکه درشت**

گره های قرار گرفته در قسمت 3 در مرحله قبل در این قسمت مورد بررسی قرار می گیرد. از آنجایی که هر گره، مشترک چند ضلع می باشد و با علم به این موضوع که یک گره ممکن است در یک ضلع معیار فاصله در آن رعایت شده باشد و در ضلع دیگر به این صورت نباشد، در این قسمت به بررسی این موضوع که گره مورد نظر معیار فاصله را برای تمامی گره های همسایه خود رعایت نموده، پرداخته شده است. این امر به این صورت انجام می پذیرد که بعد از انجام مرحله 11 برای تمامی اضلاع، گره هایی که به عنوان گره های محتمل شبکه درشت در آرایه Cp1 و Cp2 ذخیره شده اند با گره هایی که جهت قرار گیری در الگوریتم MIS در آرایه Point1 و Point2 ذخیره شده اند، مقایسه می شوند در صورتی که گره مورد بررسی در آرایه های مربوط به الگوریتم MIS حضور نداشته باشد به عنوان گره شبکه درشت انتخاب می شود.

1. **ذخیره گره های مرزی جهت حضور در شبکه درشت**

در این قسمت گره های مرزی شبکه درشت ذخیره می شوند. در ابتدای این قسمت از یک شرط جهت جلوگیری از ذخیره تکراری گره ها استفاده شده است.

1. **استفاده از الگوریتم** MIS **برای سایر گره های مرزی**

در این قسمت از الگوریتم MIS برای سایر گره های مرزی که در شبکه درشت قرار نگرفته اند، استفاده می شود. به این صورت که اگر بنا به قرار گرفتن یک گره در شبکه درشت باشد، طبق این الگوریتم همسایه های گره نباید در گره درشت حضور داشته باشند. از شمارنده m جهت ذخیره اضلاع و گره های حذفی در آرایه های مربوطه استفاده می شود.

اضلاعی که جهت قرار گرفتن در الگوریتم MIS ذخیره شده اند در این قسمت توسط یک حلقه مورد بررسی قرار می گیرند.

1. در این قسمت اگر ضلع مورد بررسی جز اضلاع حذف شده باشد باید به سراغ ضلع بعدی برویم. اضلاع حذف شده در قسمت 15 مشخص می شوند.
2. اگر گره مورد بررسی جز گره هایی باشد که نباید در شبکه درشت قرار بگیرد به سراغ گره بعدی همان ضلع می رویم. در صورتی که برای گره دوم هم این شرط برقرار باشد به سراغ ضلع بعدی می رویم. گره هایی که نباید در شبکه درشت قرار بگیرند در قسمت 15 مشخص می شوند.
3. در صورتی که شرط 2 برای هر دو گره برقرار باشد توسط counter مشخص می شود. ذخیره عدد 2 در این قسمت به این معناست که برای ضلع مورد بررسی، برنامه دوبار وارد این شرط شده است. در این صورت برنامه از حلقه داخلی خارج می شود.
4. در صورتی برقراری یکی از حالات 1 و یا 3 برنامه وارد این قسمت شده و ضلع بعدی مورد بررسی قرار می گیرد.
5. **انتخاب گره شبکه درشت و حذف همسایه های گره**

در صورتی که برنامه برای ضلع مورد نظر وارد هیچ یک از شرط های فوق نشود گره اول این ضلع باید در شبکه درشت قرار بگیرد. در صورتی که فقط یکبار وارد شرط 2 شویم و فقط گره مورد بررسی با گره دوم این ضلع عوض شود، گره دوم باید در شبکه درشت قرار گیرد. در صورت برقراری حالت اول برنامه وارد قسمت 1 و در صورت برقرای حالت دوم برنامه وارد قسمت 2 می شود.

1. در صورتی که گره اول ضلع در شبکه درشت قرار گیرد برنامه وارد این قسمت می شود. در این صورت اضلاعی که شامل این گره می باشد به طور کامل حذف شده و در الگوریتم MIS مورد بررسی قرار نمی گیرند. همچنین گره هایی که با این گره همسایه هستند نیز نمی توانند در شبکه درشت قرار گیرند ازین رو جهت بررسی در الگوریتم MIS حذف خواهند شد.
2. در صورتی که گره دوم ضلع در شبکه درشت قرار گیرد برنامه وارد این قسمت می شود. سایر موارد بیان شده در قسمت 1 برای این قسمت نیز صادق است.

ضلع حذف شده در آرایه DelFace و گره حذف شده در آرایه NICM ذخیره می شود.

1. **مشخص کردن گره های غیرمرزی برای قرار گرفتن در الگوریتم** MIS

این قسمت همانند قسمت 11 می باشد. فقط ذکر این نکته حائز اهمیت است که اگر ضلع مورد بررسی شامل گره مرزی باشد، برنامه به سراغ ضلع بعدی می رود.

1. **تأیید گره ها جهت قرار گرفتن در شبکه درشت**

این قسمت همانند قسمت 12 می باشد با این تفاوت که اضلاع غیرمرزی مورد بررسی قرار گرفته اند.

1. **ذخیره گره های غیرمرزی جهت حضور در شبکه درشت**

همانند قسمت 13 می باشد.

1. **استفاده از الگوریتم** MIS **برای سایر گره های غیرمرزی**

همانند قسمت 14 می باشد.

1. **انتخاب گره شبکه درشت و حذف همسایه های گره**

همانند قسمت 15 می باشد.

1. ذخیره گره های قابل حذف

در این قسمت گره هایی که باید از شبکه ریز جهت تولید شبکه درشت حذف شوند، ذخیره می شود. counter2 در این قسمت جهت پی بردن به برقرای شرط If استفاده شده است.