



**زیربرنامه:**

DoLayering3D

|  |  |  |
| --- | --- | --- |
| **توسعه دهندگان** | مرتضی نامور |  |
| کامیار صفری |  |
| **تهیه کنندگان مستند** | مرتضی نامور، کامیار صفری | |
| **تاییدکنندگان** | مرتضی نامور | |
| **تاریخ تنظیم سند** | 24/10/1396 | |
| **شناسه سند** |  | |
| **زبان برنامه‌نویسی** | **Fortran 90** | |

1. وظایف

این زیربرنامه رئوس ناحیه‌ی غیرهمسانگرد شبکه را لایه بندی میکند. به نحوی که هر نقطه با یک شماره لایه مشخص شود.

1. توضیحات و تئوری

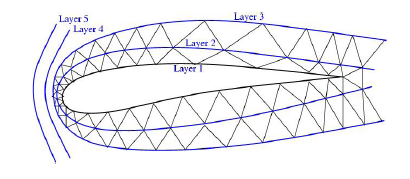
در بخش هایی از عملیات درشت سازی شبکه، نیاز داریم از لایه بندی رئوس شبکه استفاده کنیم. در این رابطه، برای هر گره ی i، یک اندیس L(i) تعریف کنیم. که این اندیس نشان دهنده ی شماره لایه ی گره ی مورد نظر می باشد.

برای لایه بندی رئوس یک شبکه، میتوانیم از اتصال بین نقاط استفاده کنیم. به این صورت که ابتدا از یک لایه برای شروع کار استفاده میکنیم که مطمئن هستیم منطقا متعلق به یک لایه هستند. برای اینکار میتوانیم از گره های داخلی مرزی بر روی یک airfoil استفاده کنیم و تمامی آنها را در شروع کار متعلق به یک لایه در نظر بگیریم. سپس با توجه به لایه ی شروع انتخاب شده، سایر نقاط باقیمانده را نیز لایه بندی کنیم.

به طور کلی برای لایه بندی گره های لایه مرزی میتوانیم از الگوریتم زیر استفاده کنیم:

برای هر گره ی i، یک اندیس L(i) تعریف کنیم. که این اندیس مشخص کننده ی شماره لایه‌ایست که هرکدام از گره های لایه مرزی به آن لایه متعلق هستند.

به عنوان مثال در شکل زیر، یک لایه بندی ساده بر روی گره های یک شبکه مشخص شده است.



1. یک نمونه از لایه‌بندی ناحیه لایه مرزی شبکه

برای لایه بندی رئوس یک شبکه، میتوانیم از اتصال بین نقاط استفاده کنیم. به این صورت که ابتدا از یک لایه برای شروع کار استفاده میکنیم که مطمئن هستیم منطقا متعلق به یک لایه هستند. برای اینکار میتوانیم از گره های داخلی مرزی بر روی یک airfoil استفاده کنیم و تمامی آنها را در شروع کار متعلق به یک لایه در نظر بگیریم. سپس با توجه به لایه ی شروع انتخاب شده، سایر نقاط باقیمانده را نیز لایه بندی کنیم.

1. بخش‌های زیربرنامه

در این قسمت، توضیح تمامی بخش‌های زیربرنامه، مطابق شماره‌گذاری انجام شده در متن برنامه کامپیوتری ارائه شده است.

1. اجرای حلقه اصلی زیربرنامه

در این بخش یک حلقه تا زمانی اجرا میشود که دیگر شماره‌ی هیچ نقطه ای در شبکه تغییر پیدا نکند. اینکار را با استفاده از یک متغیر منطقی انجام میدهیم. در صورتی که تغییری در شماره لایه‌ی مربوط به نقاط شبکه داده شود، این متغیر با مقدار TRUE مقداردهی شده و حلقه‌ی اصلی برای یکبار دیگر نیز اجرا میشود.

1. پیمایش تمام نقاط شبکه

در این بخش، در ابتدا متغیری که نشان دهنده ی بروز تغییر در شماره لایه‌های مربوط به نقاط است را برابر با مقدار اولیه‌ی False قرار میدهیم و همچنین شماره لایه را یک واحد افزایش میدهیم. در ادامه تمامی نقاط شبکه را برای بررسی اینکه آیا میتوانیم به آنها یک شماره لایه تخصیص دهیم یا نه، پیمایش میکنیم. در هربار تکرار حلقه، یک نقطه را انتخاب میکنیم. در صورتی که شماره لایه ای به آن تخصیص نداده باشیم(شماره لایه‌ی مربوط به آن صفر باشد) و اینکه نقطه ی مورد نظر در شبکه موجود باشد، آن نقطه را پردازش میکنیم. در غیر این صورت از آن صرفنظر میکنیم.

1. بررسی نقاط مربوط به لایه‌ی مجاور نقطه‌ی انتخاب شده

در ابتدا یک متغیر منطقی را که نشان دهنده ی وجود نقطه ای با شرایط مورد نظر است را برابر با مقدار اولیه‌ی False قرار میدهیم. سپس از تمامی سلولهای متصل به نقطه ی مورد نظر، نقاط آن سلولها را در یک حلقه ی داخلی تر پیمایش میکنیم.

1. بررسی شرایط وجود نقطه‌ی قابل مقداردهی

با انتخاب هرکدام از نقاط مربوط به هر سلول، در صورتی که شماره لایه‌ی آن نقطه بزرگتر از صفر باشد(به آن شماره تخصیص داده باشیم) و همچنین شماره‌ی آن از شماره لایه‌ی فعلی کوچکتر باشد، متغیر منطقی مورد نظر را برابر با True مقداردهی کرده و از حلقه خارج میشویم.

1. مقداردهی شماره لایه‌ی مربوط به نقطه‌ی مورد نظر

در صورتی که متغیر مورد نظر برابر با True باشد، یعنی اینکه حداقل یک گره با شرایط مورد نظر در لایه های مجاور نقطه ی مورد نظر وجود دارد. بنابراین شماره لایه ی مورد نظر را به نقطه تخصیص داده و متغیر تغییر را برابر با True قرار میدهیم تا یکبار دیگر حلقه ی اصلی برنامه بتواند اجرا شود.