



**زیربرنامه:**

dlt(Point)

|  |  |  |
| --- | --- | --- |
| **توسعه دهندگان** | مرتضی نامور |  |
| **تهیه کنندگان مستند** | مرتضی نامور | |
| **تاییدکنندگان** | مرتضی نامور | |
| **تاریخ تنظیم سند** | 22/02/1394 | |
| **شناسه سند** | **MC2F003F1** | |
| **زبان برنامه‌نویسی** | **Fortran 90** | |

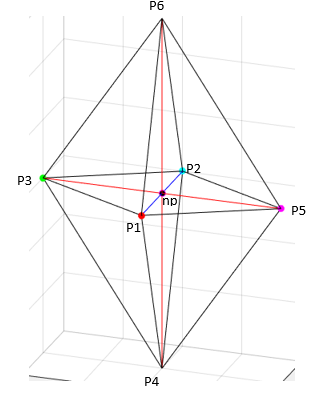
1. وظایف

این متد به منظور حذف یک گره از شبکه هندسی تعریف شده است. ورودی متد آدرس یک گره (POINT) می باشد.

1. توضیحات و تئوری­ها

بمنظور حذف یک گره از شبکه هندسی در عمل بایستی مختصات آن گره را بر یکی از گره های مجاورش که توسط یک یال به آن متصل است، منطبق کنیم. با این حال امکان انجام این عملیات همیشه وجود ندارد. لذا باید بررسی شود که آیا با حذف آن گره از شبکه هندسی هیچ المانی با حجم منفی بوجود می آید یا خیر. در صورتیکه چنین المانی بوجود نیاید حذف گره ممکن خواهد بود .

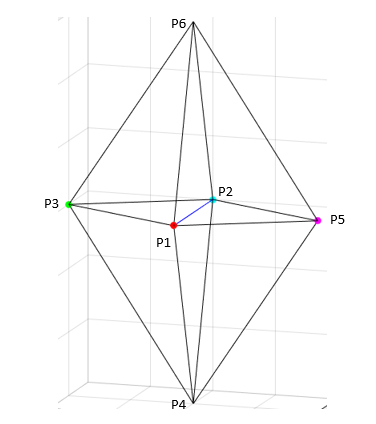
به شکل زیر توجه کنید :



1. مجموعه ای از المان ها و گره منتخب برای حذف شدن

در شکل 1 قصد داریم گره np را حذف کنیم. ابتدا به ضلع بین np و گره 1 توجه شود. چهار المان در شکل 1 وجود دارد که در این ضلع مشترک هستند. در صورتیکه گره np را بر گره 1 منطبق می کنیم این چهار المان باید حجمشان صفر شده و از بین بروند. همچنین در هر المان دو وجه که دارای ضلع مذکور هستند نیز حذف می شود. همچنین یک وجه که دارای گره np هست ، بر وجه مقابل خود که دارای گره 1 است منطبق خواهد شد. این قائده یک قائده کلی است و در تمام حالاتی که امکان حذف گره بدین صورت وجود داشته باشد ، اتفاق خواهد افتاد.

نتیجه حذف گره np در شکل زیر مشاهده می شود :



1. ترکیب المان ها پس از حذف گره np
2. بخش­های زیربرنامه
3. تعریف پارامتر­های اولیه

در این بخش ابتدا بررسی می شود که گره ورودی به متد یک گره مرزی نباشد. در صورتیکه گره مرزی نباشد آن گاه پارامترهای اولیه متد تعریف می شود.

1. محاسبه حجم المان های متصل به گره ورودی متد

در این بخش ابتدا بررسی می شود که همه المان های متصل به گره ورودی از نوع تتراهدرال باشند. در صورتیکه حتی یکی از المان ها تتراهدرال نباشد برنامه از متد خارج می شود. در صورتکه شرط برقرار باشد آن گاه با نوشتن یک حلقه حجم تک تک المان های متصل به گره را محاسبه کرده و مجموع آن ها را در آرایه A ذخیره می کنیم.

1. پیدا کردن گره مناسب جهت انطباق گره ورودی بر روی آن

در این بخش در بین المان های متصل به گره ورودی جستجو می کنیم. گره های آن المان را می یابیم و اولین گرهی را که مخالف گره ورودی به متد باشد را انتخاب کرده و در متغیر Ptest قرار می دهیم.

توجه شود در اینجا یک حلقه کلی بر روی تمامی المان های متصل به گره ورودی وجود دارد. این امر به این علت است که اگر هیچ گرهی در اولین المان وجود نداشت تا بتوان گره ورودی را بر آن منطبق کرد و تمامی شروط حذف گره برقرار باشد آن گاه متد به اول این حلقه برگشته و المان دیگری در همسایگی گره انتخاب کند و عملیات حذف را با یکی از گره های آن انجام دهد.

1. تشکیل ضلع مورد نظر و یافتن المان های متصل به آن

در این بخش ابتدا مختصات گره ورودی را در متغیری به نام Pold ذخیره می کنیم تا در صورتیکه عملیات حذف انجام نشد، گره ورودی را به مختصات اولیه اش بر گردانیم. سپس یک ضلع با گره ورودی و گره Ptest (یکی از گره های مجاور گره ورودی) می سازیم. در ادامه به کمک متد find المان های متصل به این ضلع را می یابیم و آن ها را درون آرایه dc کپی می کنیم. توجه شود در صورتیکه عملیات حذف گره انجام شود این المان ها نیز از شبکه حذف خواهند شد. در انتها مختصات گره Ptest را در گره ورودی کپی می کنیم. یعنی گره ورودی را بر گره Ptest منطبق می کنیم.

1. محاسبه حجم المان های متصل به گره ورودی در حالت جدید

با توجه به تغییر مختصات گره ورودی اکنون حجم المان های متصل به گره را مجددا محاسبه می کنیم و مجموع آن را در پارامتر B ذخیره می کنیم.

1. بررسی شرط امکان حذف گره

در این بخش ابتدا بررسی می شود که اگر مقدار نرمال تفاضل مجموع حجم المان ها در حالت اولیه و ثانویه از یک عدد بسیار کوچک ، کمتر باشد (عملا صفر باشد) آن گاه بررسی می شود که حجم تک تک المان های ذخیره شده در آرایه dc برابر صفر شده باشد. در صورتیکه حجم هر کدام از آن ها برابر صفر نباشد آن گاه عملیات حذف امکان پذیر نیست و باید گره ورودی را به مختصات اولیه برگردانیم و به ابتدای متد برگردیم. در صورتیکه حجم همه برابر صفر باشد آن گاه به بخش بعد خواهیم رفت.

1. محاسبه تعداد المان های با حجم صفر

در این بخش با محاسبه حجم تک تک المان های متصل به گره ورودی (که اکنون تغییر مختصات داده است) تعداد المان های با حجم صفر را می یابیم. این تعداد باید دقیقاً برابر با تعداد المان های متصل به ضلع ساخته شده در بخش 4 باشد. در صورتیکه برابر نباشد مجددا عملیات حذف امکان پذیر نیست و باید به ابتدای متد برگردیم.

1. پاک کردن آدرس المان های با حجم صفر از لیست المان های متصل به گره های آن ها

بدون توضیح

1. پیدا کردن وجوه مربوط به المان هایی که اکنون دارای حجم صفر هستند.

در این بخش ابتدا بر روی المان های درون آرایه dc حرکت کرده و وجوه آن ها را در آرایه F وارد می کنیم. توجه شود که هر المان دارای 4 وجه است پس درایه های آرایه F چهارتا چهارتا مربوط به یک المان می باشد.

همانطور که در بخش تئوری توضیح داده شد می دانیم که در المان هایی که حذف می شوند، دو وجه به طور کامل از بین می رود (یعنی مساحتش صفر می شود) و دو وجه دیگر مساحتی غیر صفر دارند . اما این دو وجه غیر صفر نیز بر یکدیگر منطبقند. با ذکر این نکات بسراغ مرتب کردن وجوه در آرایه F می رویم. ابتدا دو وجهی را که دارای مساحت غیر صفر هستند را پیدا کرده و در درایه های با اندیس F[4\*i] و F[4\*i+1] قرار می دهیم. توجه شود دو درایه F[4\*i+2] و F[4\*i+3] نیز مربوط به وجوه المان i ام هستند. اما پس از مرتب کردن این دو وجه اخیر دارای مساحت صفر می باشند.

در انتها بین دو وجه با اندیس F[4\*i] و F[4\*i+1] جستجو می کنیم و آن وجهی که گره ورودی در آن قرار نداشت را در درایه F[4\*i] قرار می دهیم.

اکنون ترتیب وجوه برای هر المان در آرایه F به این صورت است که اولین وجه ، وجهی است که بدون تغییر مانده ، دومین وجه چون دارای گره ورودی است لذا بر وجه اول منطبق می شود. وجه سوم و چهارم نیز باید از شکه حذف شوند.

1. اصلاح وجوه المان های همسایه وجوه با اندیس 4\*i+1

در این بخش بر روی وجوهی که باید منطبق بر وجه اول هر المان شوند حرکت کرده و به کمک متد find المان متصل به آن را می یابیم. در بین وجوه این المان ، آن وجهی را می یابیم که مطابق با وجوه با اندیس 4\*i+1 است و بجای آن وجه اول آن المان یعنی وجه با اندیس 4\*i را قرار می دهیم. برای بهتر متوجه شدن مطلب به شکل 1 دقت کنید. در این شکل وجهی که با گره های np ، 3 و 6 ساخته شده است پس از انطباق np بر گره 1 ، بر وجهی که با گره های 1 ، 3 و 6 ساخته شده است منطبق می شود. حال المانی که با گره های np ، 2 ، 3 و 6 ساخته شده است را ملاحظه کنید. یکی از وجوه این المان همان وجه با گره های np ، 3 و 6 است. در این بخش ما این وجه را در المان مذکور پیدا می کنیم و بجای آن وجه با گره های 1 ، 3 و 6 را قرار می دهیم. این کار برای تمامی المان های باقیمانده اتفاق می افتد. توجه شود اکنون حجم المانی که با گره های np ، 1 ، 3 و 6 ساخته شده است برابر با صفر می باشد.

1. اضافه کردن المان های متصل به ضلع ورودی ، به لیست المان های متصل به گره دیگر ضلع حذف شده

در بخش 8 آدرس المان هایی که حجمشان صفر شده بود را از لیست المان های متصل به گره ورودی به متد حذف کردیم. اکنون باید مابقی المان های متصل به آن گره را که حجمشان برابر صفر نشده است را به لیست المان های متصل به گره Ptest (گرهی که گره ورودی بر آن منطبق شده است) اضافه کنیم.

1. اصلاح گره مربوط به وجوه مرتبط با المان های متصل به گره ورودی

با توجه به اینکه در انتهای متد می خواهیم آدرس گره ورودی را از شبکه حذف کنیم لذا قبل از آن باید در بین وجوه المان های متصل به این گره جستجو کنیم و هروجهی که گره ورودی یکی از گره های آن بوده است را پیدا کرده و بجای گره ورودی در آن گره Ptest را قرار دهیم. به طور مثال در شکل 1 وجهی که با گره های np ، 2 و 6 ساخته شده است را باید پیدا کنیم و بجای گره np در آن گره 1 را قرار دهیم.

1. حذف المان ها و وجوهی که باید از شبکه حذف شوند

در این بخش المان های موجود در آرایه dC را به کمک متد push\_back به وکتور V انتقال می دهیم (تا بعدا به واسطه متد clear از شبکه حذف شود) .

همچنین وجوهی که در آرایه F با شماره اندیس های 4\*i+1 ، 4\*i+2 و 4\*i+3 قرار دارند را نیز به وکتور V می فرستیم تا بعدا از شبکه حذف شود.

1. جلوگیری از حذف شدن اشیاء تکراری

با توجه به اینکه وجوه قرار گرفته در آرایه F کمابیش تکراری هستند لذا باید اشیاء تکراری را از درون وکتور V پاک کنیم تا چند مرتبه برای حذف شدن فرستاده نشوند.

1. حذف گره ورودی

در انتها آدرس گره ورودی ه متد را به کمک متد Push\_Back به درون وکتور V می فرستیم تا بعدا به کمک متد clear از شبکه حذف شود.