

****

**عنوان:**

بهینه‌سازی کیفیت شبکه محاسباتی با تبدیل شبکه مثلثی به چهار ضلعی

|  |  |  |
| --- | --- | --- |
| **نویسندگان** | مرتضی نامور |  |
| کورش مرادیان |  |
| **تاریخ تنظیم سند** | 19/9/1396 | |
| **شناسه سند** |  | |

**فهرست مطالب**

[فصل 1- راهنمای کاربری 1](#_Toc512959600)

[1-1- فایل ورودی 1](#_Toc512959601)

[1-2- اجرای برنامه 1](#_Toc512959602)

[1-3- فایل‏های خروجی 1](#_Toc512959603)

[1-4- ساختار داده‏ای شبکه 2](#_Toc512959604)

[فصل 2- اعتبارسنجی و نتایج 5](#_Toc512959605)

[فصل 3- تئوری و الگوریتم 14](#_Toc512959606)

[3-1- تعریف و دسته‏بندی لبه‏های Front 17](#_Toc512959607)

[3-2- انتخاب Front مناسب بر اساس اولویت برای پردازش 18](#_Toc512959608)

[3-3- ایجاد ضلع‏های کناری 18](#_Toc512959609)

[3-4- ایجاد ضلع بالایی 21](#_Toc512959610)

[3-5- تولید چهارضلعی 23](#_Toc512959611)

[3-6- بهبود کیفیت چهارضلعی جدید 23](#_Toc512959612)

[3-7- بروزرسانی و دسته‏بندی مجدد لبه‏های Front 30](#_Toc512959613)

[3-8- عملیات ویژه Seam 30](#_Toc512959614)

[3-9- عملیات ویژه Transition Seam 31](#_Toc512959615)

[3-10- عملیات ویژه Transition Split 31](#_Toc512959616)

[3-11- تشخیص متقاطع بودن دولبه 32](#_Toc512959617)

[3-12- نوآوری‏ها 33](#_Toc512959618)

[3-13- پاکسازی توپولوژیکی 35](#_Toc512959619)

[3-14- هموارسازی کلی 36](#_Toc512959620)

[فصل 4- پیاده‌سازی و زیربرنامه‌های مورد استفاده 38](#_Toc512959621)

[4-1- برنامه اصلی QMorph 38](#_Toc512959622)

[فصل 5- مراجع 40](#_Toc512959623)

**چکیده:**

در این پروژه هدف ایجاد شبکه‏های محاسباتی بهینه با استفاده از الگوریتم Q-Morph است. این الگوریتم که از سه بخش اصلی تشکیل شده است یکی از معروف‏ترین الگوریتم‏های موجود در زمینه تبدیل شبکه‏های مثلثی به چهارضلعی می‏باشد. بخش‏های این الگوریتم عبارتند از: 1- تبدیل شبکه مثلثی به شبکه چهارضلعی، 2- بهبود ساختار شبکه ایجاد شده با استفاده از مجموعه‏ای از الگوریتم‏ها که با تغییر اتصالات توپولوژی شبکه را بهبود می‏بخشند. و 3- بهبود کیفیت المان‏های شبکه با اعمال یک الگوریتم ترکیبی که با تغییر مکان سیستماتیک نودهای شبکه، کیفیت شبکه را افزایش می‏دهد. در بخش اول مجموعه‏ای از عملیات شامل شناسایی و اولویت‏بندی ضلع‏های پایه، ساخت ضلع‏های کناری، ایجاد ضلع بالایی، تشکیل چهارضلعی و بهبود کیفیت المان ایجاد شده به صورت محلی[[1]](#footnote-1) انجام می‏شوند. در این بخش در برخی شرایط خاص عملیات‏های ویژه‏ای تعریف شده‏اند که در این موارد، عملیات ویژه به جای روند ذکر شده صورت می‏پذیرد. این بخش با تبدیل شدن کامل شبکه مثلثی به شبکه چهارضلعی پایان می‏پذیرد. در بخش دوم شبکه چهارضلعی حاصله از بخش قبلی پردازش می‏شود و با یافتن و اصلاح کردن مجموعه‏ای از الگوهای ساختاری شبکه، نودهای نامعمول شبکه حداقل‏سازی می‏شوند. در بخش سوم از الگوریتمی ترکیبی از یک روش هیورستیک به همراه یک روش گرادیان کاهشی تکرارشونده در جهت بهبود کیفیت المان‏های شبکه بهره گرفته می‏شود. در خلال پیاده‏سازی الگوریتم Q-Morph با اضافه کردن عملیات‏هایی خاص به برخی از قسمت‏های بخش اول، الگوریتم توسعه داده شده است.

**کلمات کلیدی:** Q-Morph، تبدیل شبکه مثلثی به شبکه چهارضلعی، توپولوژی، تغییر اتصالات، روش گرادیان کاهشی تکرارشونده.

# راهنمای کاربری

## فایل ورودی

در این برنامه یک قالب مشخص، که سعی می شود در تمام برنامه های مربوط به حوزه تولید شبکه دو بعدی از آن استفاده گردد، بکار گرفته شده است. برای اطلاعات بیشتر در مورد این قالب و ساختار فایل ورودی می توانید به مستندات زیربرنامه Read\_2DMeshC مراجعه نمایید. تمام اطلاعات لازم باید در قالب یک فایل بنام MeshIn.cgid تهیه شده و در پوشه حاوی برنامه قرار داده شود.

## اجرای برنامه

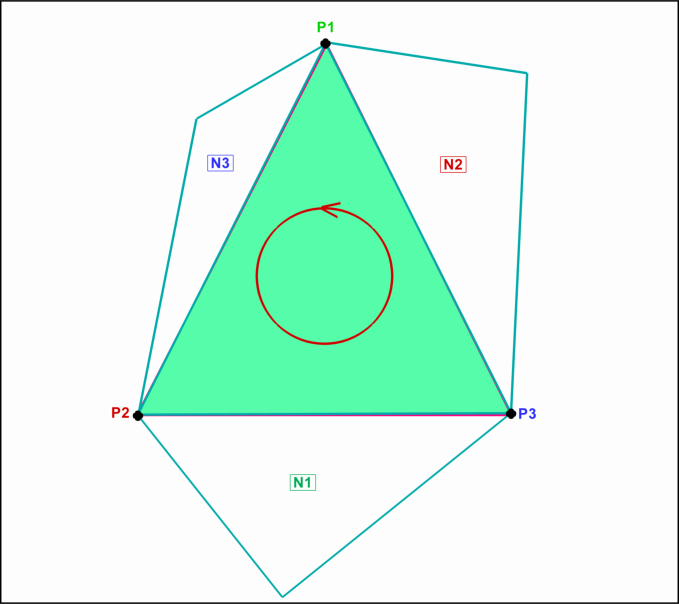
این برنامه تحت نرم افزار Visual Studio 2013 و Intel Parallel Studio XE 2015 توسعه یافته است بنابراین جهت اجرای آن بهتر است این نرم افزارها را به ترتیب ذکر شده نصب نمایید. در اینجا یک پارامتری بنام Dim برای تعیین مقدار ماکزیمم بعد آرایه های استفاده شده، در نظر گرفته شده است. یک مقدار پیش فرض برای این پارامتر تعیین شده است و در صورتیکه با خطای Array bound exceed مواجه شدید باید مقدار پارامتر Dim را افزایش دهید.

## فایل‏های خروجی

یک قالب مشخص برای چاپ اطلاعات در نظر گرفته شده است تا با ساختار فایل های ورودی سازگاری داشته باشد. به این ترتیب ارتباط بین برنامه های نوشته شده براحتی امکانپذیر می باشد. برای اطلاعات بیشتر در مورد این قالب و ساختار فایل خروجی می توانید به مستندات زیربرنامه Write\_2DMeshC مراجعه نمایید. بنابراین اطلاعات خروجی با توجه به ساختارهای در نظر گرفته شده در یک فایل بنام 2DMeshC.Txt چاپ می شود. همچنین جهت مشاهده شبکه و مرزها می توان فایل 2DMeshC.Plt را مشاهده نمود.

## ساختار داده‏ای شبکه

باید در نظر داشت که برای یک شبکه دوبعدی نحوه ذخیره نقاط تشکیل دهنده هر المان (فایل های ورودی و خروجی) بگونه ایست که بنابر قانون دست راست جهت بردار عمود بر المان بطرف خارج از صفحه و برای یک شبکه سطحی این جهت بسمت داخل جسم می باشد. همچنین برای المان‏های مثلثی نحوه ذخیره همسایه ها متناظر با ترتیب نقاط تشکیل دهنده المان می باشد. برای مثال همسایه شماره 1 مثلث زیر روبری اولین نقطه تشکیل دهنده آن است و به همین ترتیب سایر همسایه ها ذخیره شده اند. این نحوه ذخیره اطلاعات برای فایل ورودی و خروجی بکار برده می شود. برای روشن شدن این موضوع به ‏شکل (1) توجه کنید.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **نقطه اول** | **نقطه دوم** | **نقطه سوم** | **نقطه چهارم** | **همسایه اول** | **همسایه دوم** | **همسایه سوم** | **همسایه چهار** |
| P1 | P2 | P3 | φ | N1 | N2 | N3 | φ |

1. نحوه ذخیره نقاط تشکیل دهنده و همسایه های یک المان مثلثی

این ساختار را می‏توان برای شبکه‏های چهارضلعی و حتی شبکه‏های ترکیبی استفاده کرد. ساختار کلی به این صورت تعریف می‏شود که اطلاعات مربوط به نقاط تشکیل دهنده المان‏های شبکه در یک ماتریس دو بعدی با ابعاد Dim\*4 ذخیره می‏شود. به هر المان مثلثی یا چهارضلعی یک سطر اختصاص داده می‏شود که به ترتیب نقاط اول تا چهارم را مشخص می‏کنند. برای المان‏های مثلثی که از سه نقطه تشکیل شده‏اند نقطه چهارم برابر با مقدار مشخص φ قرار داده می‏شود که بیانگر نداشتن نقطه چهارم می‏باشد. در پیاده‏سازی به زبان فرترن از آنجاییکه اندیس آرایه‏ها از یک شروع می‏شود می‏توان از عدد صفر به عنوان مقدار مشخص φ استفاده کرد. برای المان‏های چهارضلعی نیز از تعریف نقاط تشکیل دهنده المان به صورت پادساعتگرد[[2]](#footnote-2) است. در چهارضلعی شکل زیر نیز قانون دست راست صدق می‏کند.

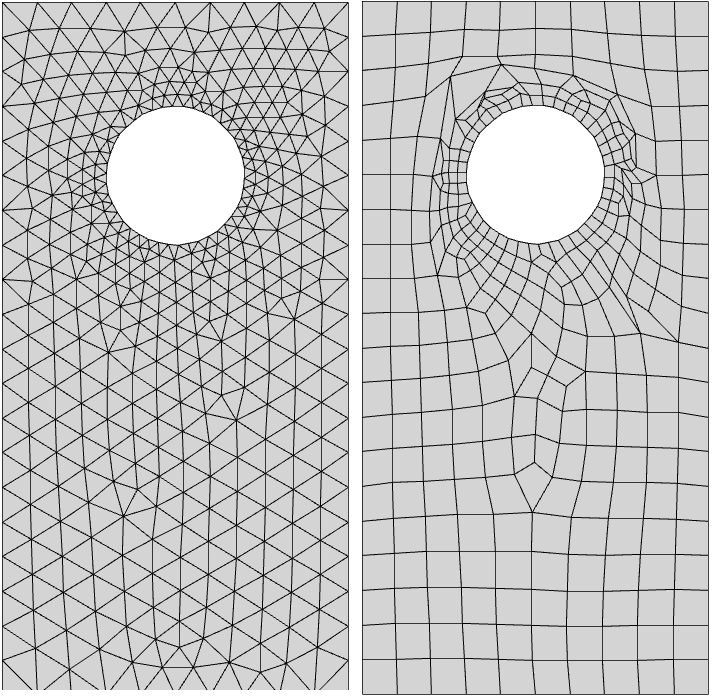
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **نقطه اول** | **نقطه دوم** | **نقطه سوم** | **نقطه چهارم** | **همسایه اول** | **همسایه دوم** | **همسایه سوم** | **همسایه چهار** | | V1 | V2 | V3 | V4 | N1 | N2 | N3 | N4 | |

1. نحوه ذخیره‏سازی نقاط تشکیل دهنده و همسایه‏های یک المان چهارضلعی

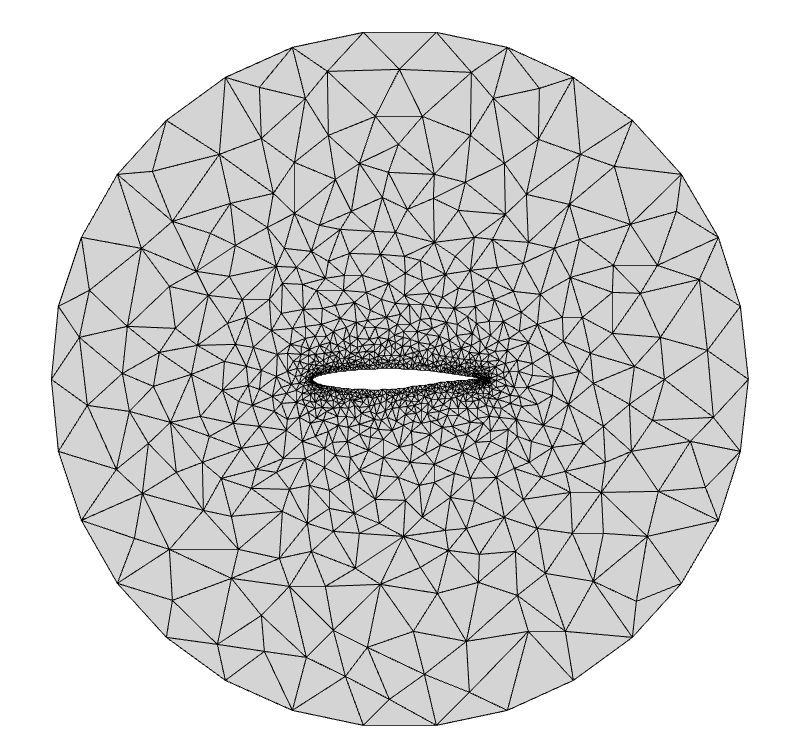
برای ذخیره کردن اطلاعات همسایه‏های هر المان نیز می‏توان از ماتریس مشابهی با همان ابعاد استفاده کرد که به ترتیب همسایه‏های اول تا چهارم هر المان را در سطر متناظر آن نشان می‏دهد. مانند آنچه در بالا در ‏شکل (1) توصیف شده است برای المان‏های مثلثی همسایه مشخص شده در هر درایه المان روبروی نقطه تشکیل دهنده آن المان در همان درایه می‏باشد. همسایه چهارم مثلث نیز با مقدار مشخصی مقداردهی می‏شود. در تعریف همسایه‏های المان‏های چهارضلعی‏ نمی‏توان به شیوه استفاده شده در مورد المان‏های مثلثی عمل کرد. در این المان‏ها المان مجاور هر ضلع از چهارضلعی بسته به مقایسه اندیس‏های نقاط تشکیل دهنده آن ضلع در درایه مشخصی ذخیره می‏شود. قاعده کلی به این صورت است که همواره اختلاف شماره اندیس‏های نقاط تشکیل دهنده ضلع مربوطه اندازه‏گیری می‏شود. در صورتی که این اختلاف برابر یک باشد المان مجاور آن ضلع در درایه با اندیس کوچکتر در ماتریس همسایه‏ها ذخیره می‏شود در غیر اینصورت اگر اختلاف محاسبه شده برابر یک نباشد المان مجاور ضلع مربوطه در درایه با اندیس بزرگتر ذخیره می‏شود. این نحوه ذخیره‏سازی در ‏شکل (2) برای المان چهارضلعی بالا نشان داده شده است.

# اعتبارسنجی و نتایج

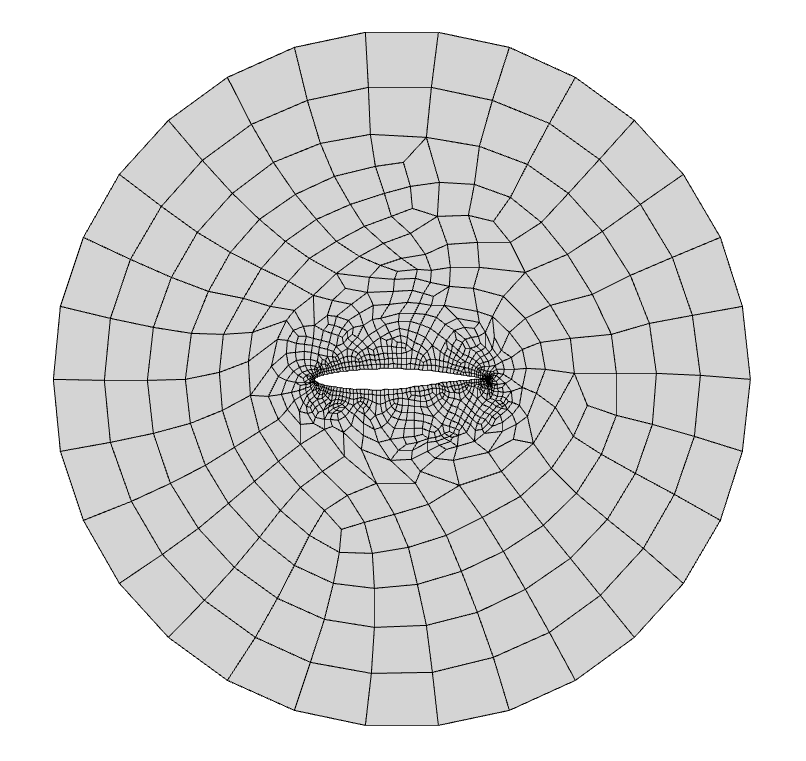
در این بخش چند نمونه از شبکه­های تولید شده توسط برنامه آورده شده است. اولین شبکه مربوط به میدان قرار گرفته شده در بین یک دایره و مستطیل اطراف آن می باشد. در تبدیل این شبکه‏ها درجه نودها مقداری بین 3 تا 5 خواهد بود و الگوریتم Q-Morph سعی در کمینه کردن نودهای با درجه 3 و 5 دارد. در تبدیل شبکه‏ها کیفیت المان‏ها بر طبق مقادیر α و β تعریف شده در [1] تضمین می‏شوند. شبکه اول و تبدیل آن در ادامه در ‏شکل (3) نشان داده شده‏اند. شبکه دوم یک ایرفویل را نشان می­دهد. همانگونه که مشخص است شبکه مثلثی در ‏شکل (4) به یک شبکه چهارضلعی با کیفیت نشان داده شده در ‏شکل (5) تبدیل شده است. در ادامه تبدیل‏های شبکه‏های دیگری نیز نشان داده شده اند.



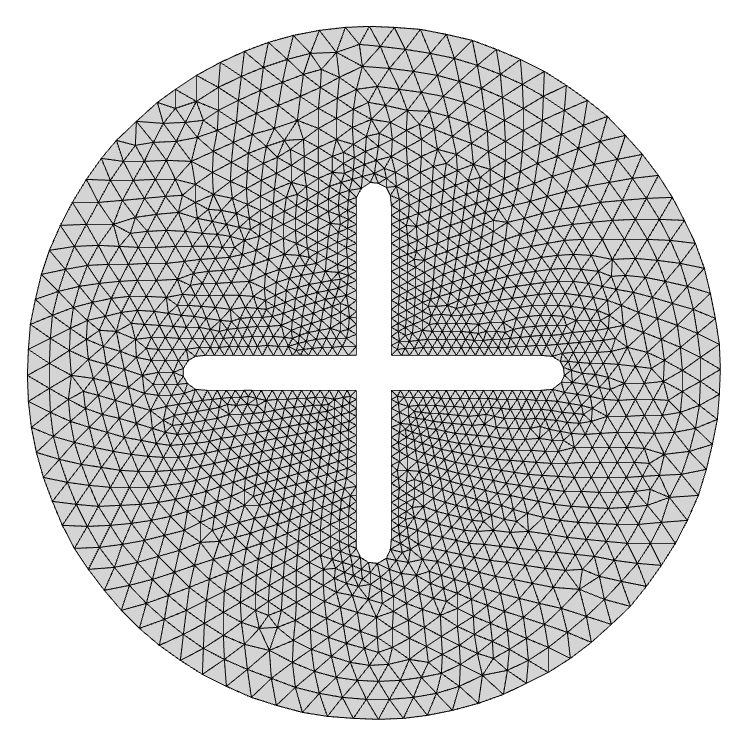
1. تبدیل شبکه در میدان بین یک دایره و مستطیل



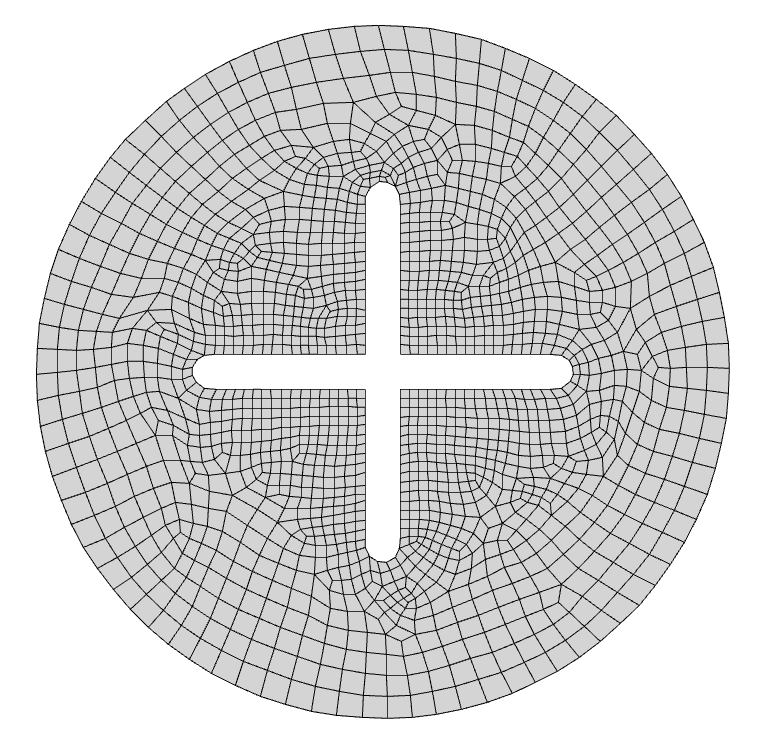
1. شبکه مثلثی در اطراف ایرفویل



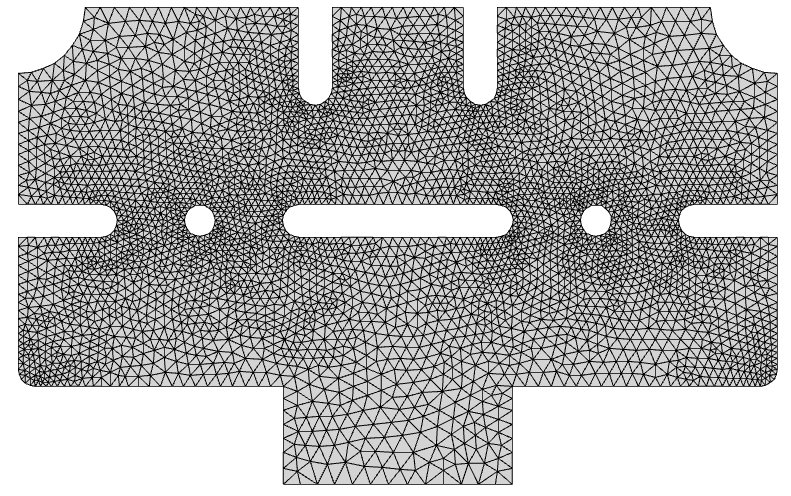
1. تبدیل شبکه در اطراف ایرفویل



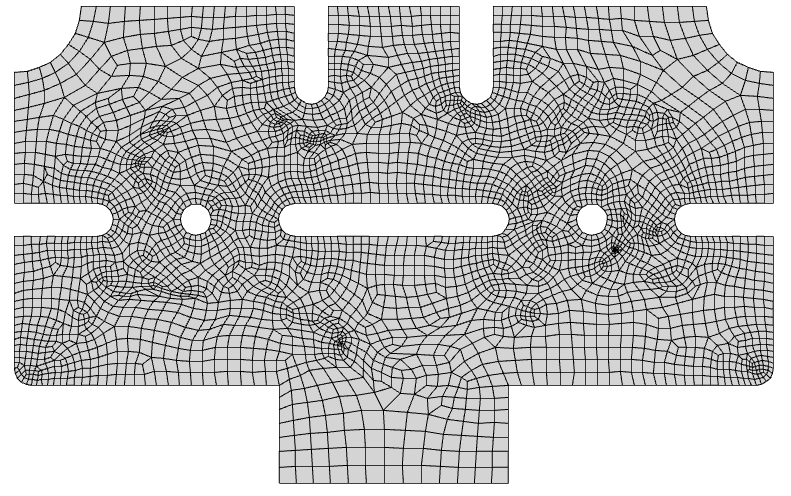
1. شبکه مثلثی با حفره به شکل علامت جمع



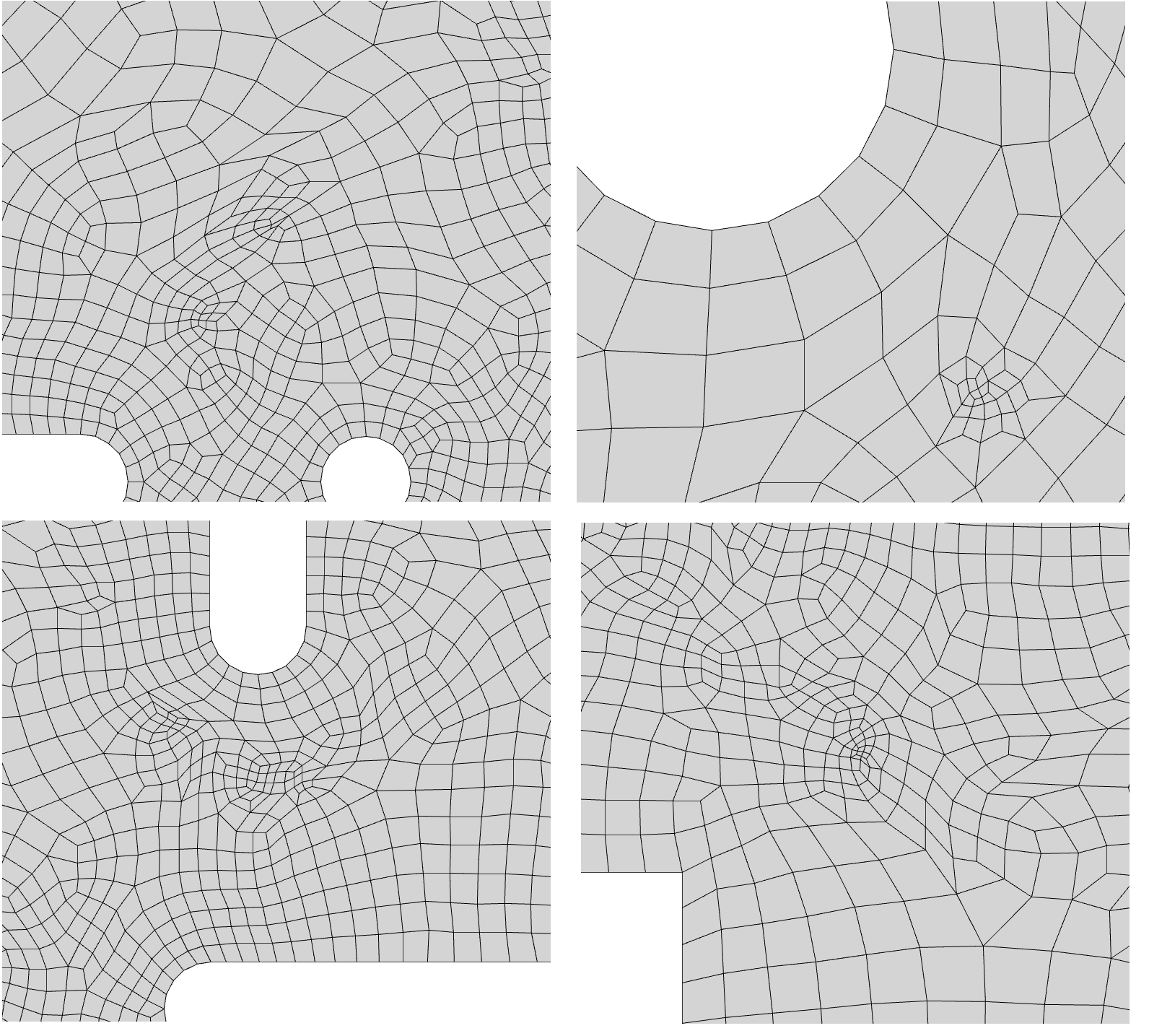
1. تبدیل شبکه مثلثی در شکل قبلی



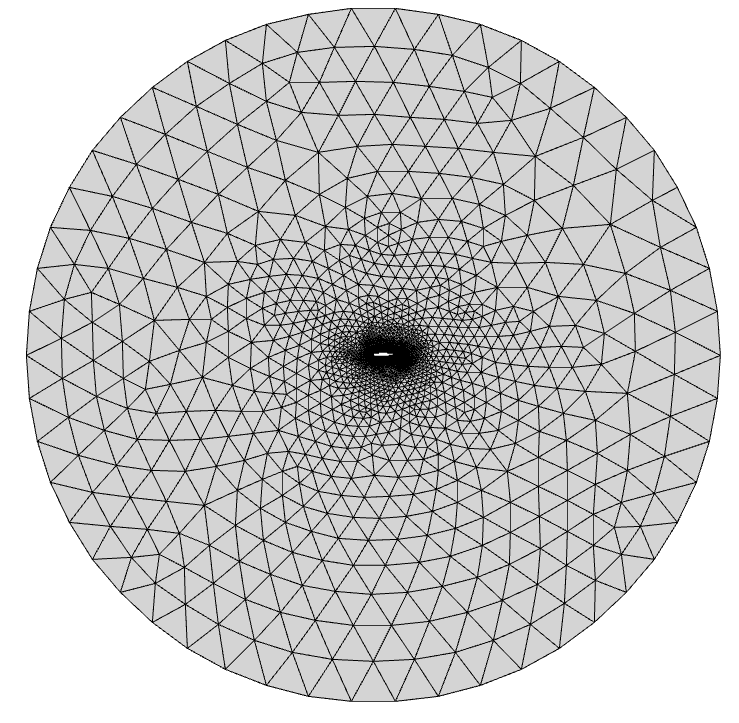
1. شبکه مثلثی با چند حفره



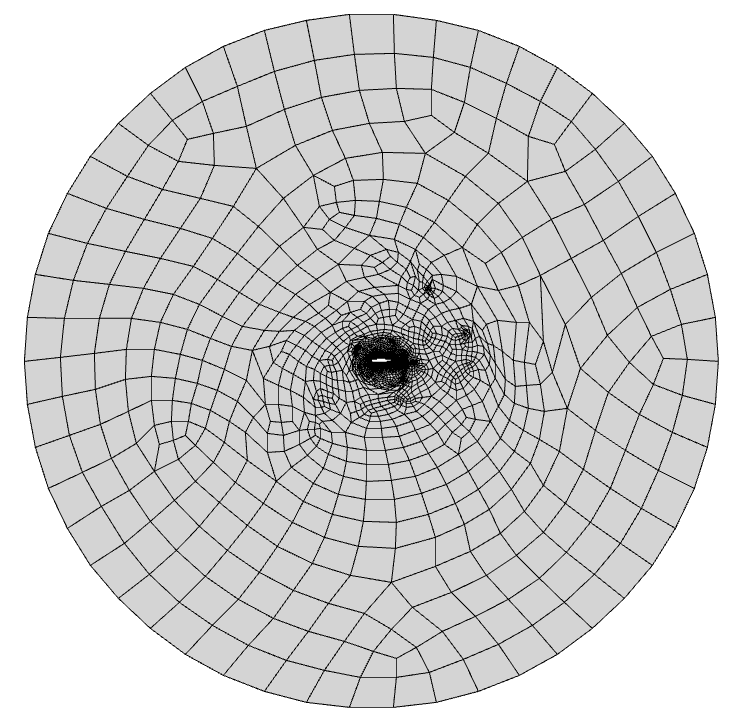
1. تبدیل شبکه مثلثی شامل چند حفره و تورفتگی



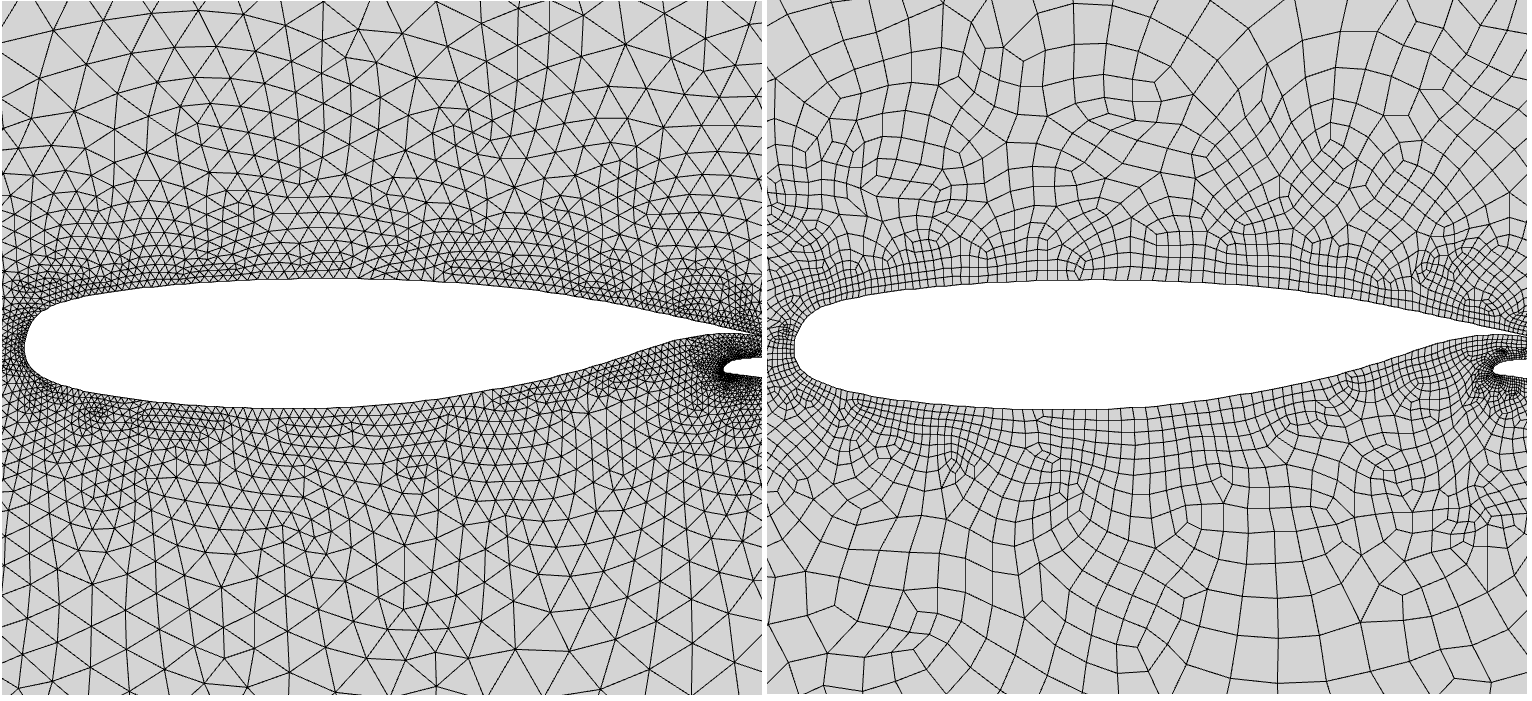
1. بزرگنمایی بخش‏های متراکم شبکه تبدیل شده شکل قبلی



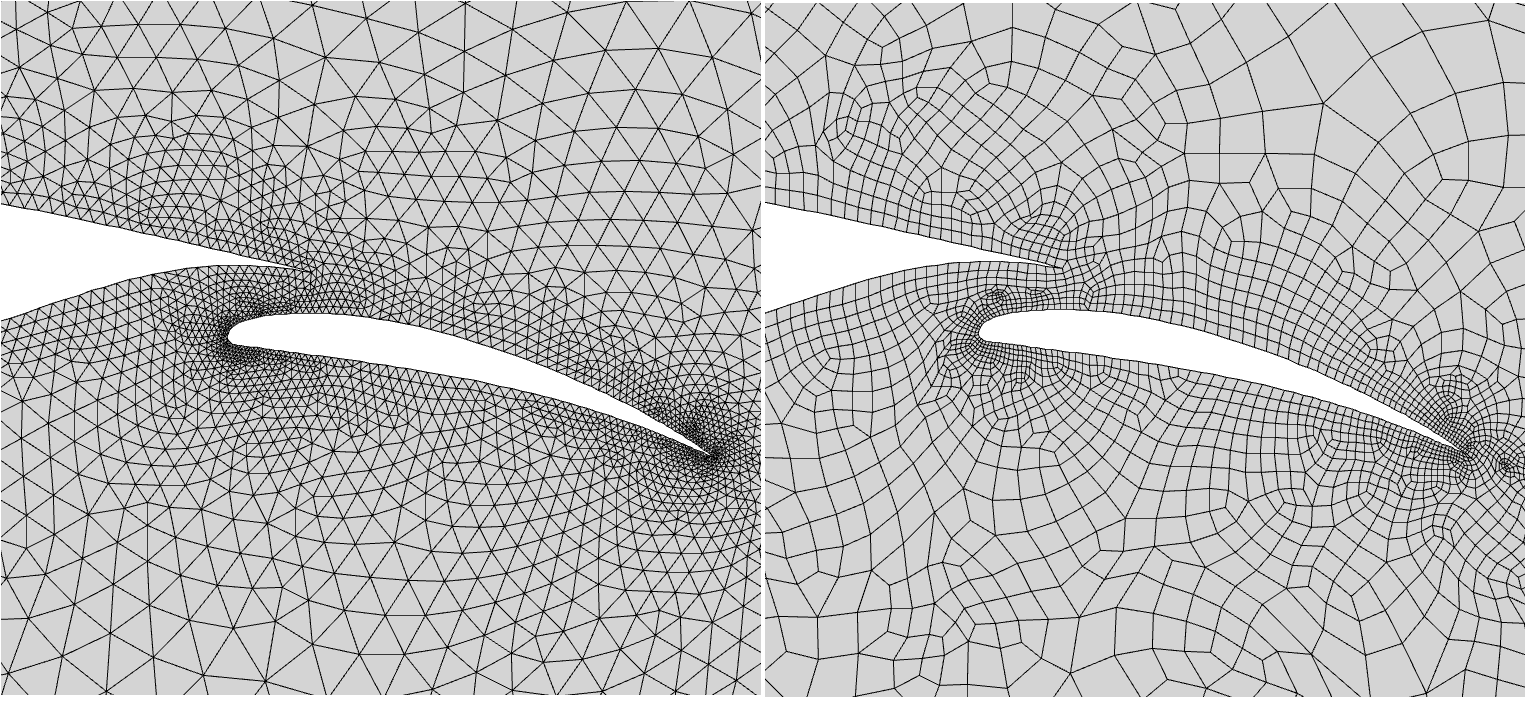
1. نمای کلی شبکه مثلثی شامل دو ایرفویل



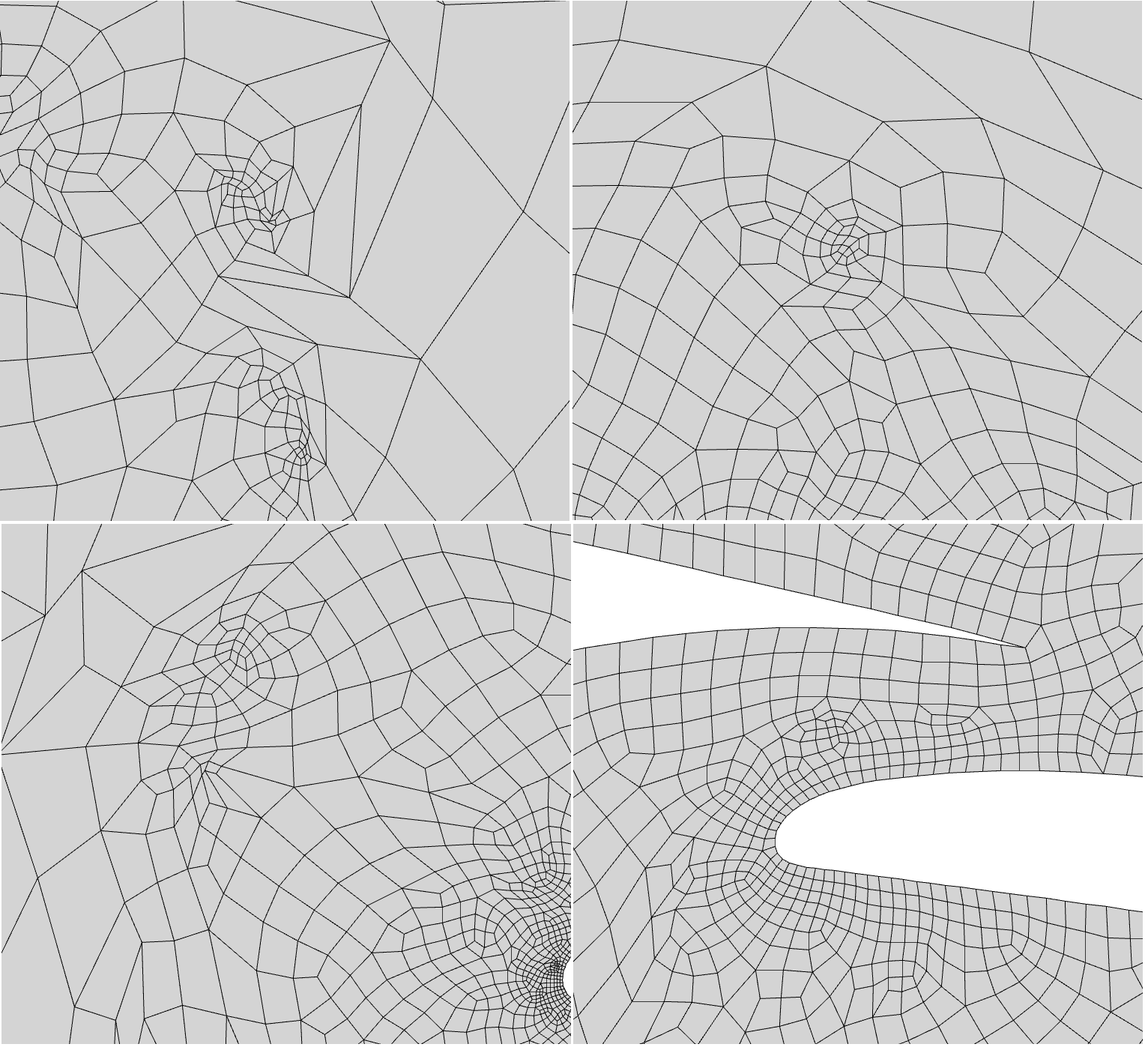
1. نمای کلی شبکه تبدیل شده شامل دو ایرفویل



1. بزرگنمایی اطراف ایرفویل بزرگتر و تبدیل شده آن



1. بزرگنمایی اطراف ایرفویل کوچکتر و تبدیل شده آن



1. بزرگنمایی بخش‏های متراکم شبکه شامل دو ایرفویل

# تئوری و الگوریتم

در این بخش تئوری و الگوریتم‏های به کار رفته در الگوریتم Q-Morph آورده شده است و در مراحل بعدی نحوه پیاده سازی آنها بطور مفصل آورده می شود.

الگوریتم Q-Morph یکی از معروفترین الگوریتم‏های ارائه شده در تولید شبکه‏های چهارضلعی به شمار می‏رود. این الگوریتم در یکی از بسته‏های نرم‏افزاری ANSYS به عنوان یکی از ابزارها گنجانده شده است. این الگوریتم با انجام سه سطح از عملیات شبکه چهارضلعی با کیفیتی ایجاد می‏کند. در سطح اول عملیات تبدیل شبکه مثلثی به شبکه چهارضلعی انجام می‏شود. در سطح دوم مجموعه‏ای از عملیات بر روی شبکه چهارضلعی ایجاد شده برای بهبود توپولوژی آن انجام می‏شود. در سطح آخر عملیات‏هایی برای بهبود کیفیت المان‏های شبکه انجام می‏شود. به طور کلی دسته‏بندی‏های مختلفی برای روش‏های تولید شبکه‏های چهارضلعی ارائه شده‏اند. در یک دسته‏بندی، این روش‏ها به روش‏های مستقیم و روش‏های غیرمستقیم تقسیم‏بندی می‏شوند. در روش‏های مستقیم، با در دست داشتن اطلاعات مرزی ناحیه مورد نظر، با قرار دادن المان ها در داخل ناحیه مورد نظر، شبکه چهارضلعی ایجاد می‏شود. اما در روش غیر مستقیم یک شبکه مثلثی به شبکه چهارضلعی تبدیل می‏شود [2]. الگوریتم Q-Morph به عنوان یک روش غیر مستقیم دسته‏بندی می‏شود.

روش‏های غیرمستقیم معمولاً نودهای نامعمول[[3]](#footnote-3) زیادی در شبکه تولید شده ایجاد می‏کنند. در شبکه‏های چهارضلعی به نودهایی که به بیشتر یا کمتر از چهار المان متصل باشند نودهای نامعمول گفته می‏شود. هرچند که وجود چنین نودهایی در مواردی که اندازه المان های مجاور تغییر می‏کند ضروری هستند اما بایستی تعداد آنها در شبکه حداقل باشد. حداقل‏سازی این نودها در شبکه موجب بهبود کیفیت شبکه و به تبع آن بهبود کیفیت شبکه محاسباتی می‏شود [3]. سطح دوم عملیات‏های Q-Morph به حل این مشکل می‏پردازد. Q-Morph با بهره‏گیری از کارهای پیشین انجام شده در زمینه رفع این مشکل در [4]–[6] اقدام به حداقل‏سازی نودهای نامعمول می‏کند. در این مراجع الگوهایی از ساختار شبکه شناسایی و اصلاح می‏شوند. به این مجموعه عملیات در اصطلاح پاکسازی توپولوژیکی[[4]](#footnote-4) گفته می‏شود. پاکسازی توپولوژیکی به تنهایی نمی‏تواند کیفیت شبکه را بهبود ببخشد و در واقع تغییر مکان نودهای شبکه در جهت بهبود کیفیت شبکه امری ضروری است. Q-Morph برای بهبود کیفیت شبکه و تضمین کیفیت شبکه از الگوریتم ترکیبی پیشنهاد شده در [1] در سطح نهایی عملیات و پس از پاکسازی توپولوژیکی استفاده می‏کند.

در این تحقیق عملیات‏هایی به سطح اول عملیات تبدیل شبکه اضافه شده‏اند که نوآوری‏های این تحقیق را شامل می‏شوند. به طور کلی در سطح اول عملیات در Q-Morph که به تبدیل شبکه مثلثی به شبکه چهارضلعی می‏پردازد مجموعه عملیات زیر به ترتیب انجام می‏شوند:

1. تعریف و دسته‏بندی لبه‏های خاصی تحت عنوان Front در شبکه که به عنوان ضلع پایه برای چهارضلعی استفاده می‏شوند.
2. انتخاب Front مناسب بر اساس اولویت برای پردازش
3. ایجاد ضلع‏های کناری برای چهارضلعی جدید
4. ایجاد ضلع بالایی برای چهارضلعی جدید
5. تولید چهارضلعی با حذف مثلث‏های محصور بین اضلاع چهارضلعی
6. بهبود کیفیت چهارضلعی با تغییر مکان رئوس تشکیل دهنده آن
7. بروزرسانی لبه‏های Front و دسته‏بندی مجدد آنها

البته ممکن است پس از مرحله دو مراحل سه تا پنج انجام نشوند و بسته به شرایط Front انتخاب شده، یکی از عملیات‏های خاص تعریف شده در مقاله که تحت شرایط خاصی قابل اجرا می‏شوند به جای عملیات‏های سه تا پنج انجام شوند. این عملیات‏ها به طور خلاصه عبارتند از:

* Seam
* Transition Seam
* Transition Split

هر کدام از این عملیات‏ها و عملیات‏های پیشین به تفصیل در ادامه شرح داده خواهند شد. برای درک بهتر نحوه عملکرد الگوریتم Q-Morph به فلوچارت نشان داده شده در شکل زیر توجه کنید. در این فلوچارت عملیات‏هایی که با رنگ خاکستری مشخص شده‏اند به عنوان نوآوری این تحقیق و توسعه این روش به آن اضافه شده‏اند.

|  |
| --- |
| C:\Users\Kourosh\Desktop\QMorph.png |

1. فلوچارت نحوه عملکرد الگوریتم Q-Morph برای تولید شبکه چهارضلعی

در این تحقیق هدف تولید شبکه‏های دو بعدی بوده است اما الگوریتم Q-Morph قابلیت تعمیم به شبکه‏های سه بعدی را نیز دارد.

## تعریف و دسته‏بندی لبه‏های Front

در الگوریتم Q-Morph لبه‏ها یا اتصالاتی بنام Front به عنوان ضلع پایه برای ایجاد چهارضلعی در نظر گرفته می‏شوند. این اضلاع در واقع لبه‏هایی در شبکه هستند که فقط به یک مثلث تعلق دارند. بنابراین تعریف اگر لبه‏ای در مش که بین یک مثلث و یک چهارضلعی قرار گرفته است نیز به عنوان Front در نظر گرفته می‏شود. به اولین دسته از لبه‏های Front که طبیعتاً لبه‏های مرزی مش خواهند بود Front های سطح صفر گفته می‏شود. به همین ترتیب پس از تولید چهارضلعی بر روی Front های سطح صفر به لبه‏هایی که در مرز مشترک المان‏های مثلثی با اولین ردیف از المان‏های چهارضلعی قرار می‏گیرند Front های سطح اول گفته می‏شود. به همین ترتیب الی آخر سطح Front ها تعریف می‏شود. سطح لبه‏ها در انتخاب لبه برای پردازش تاثیرگذار است. هر چه سطح یک لبه Front کمتر باشد اولویت پردازش بالاتری دارد. همین ویژگی الگوریتم Q-Morph آنرا به الگوریتمی پیشرونده به جلو[[5]](#footnote-5) تبدیل کرده است. انتخاب Front برای پردازش علاوه بر سطح به وضعیت آن نیز بستگی دارد. ویژگی لبه‏های Front موجب می‏شود که آنها همواره در مش به صورت حلقه‏هایی از لبه‏ها به هم متصل باشند. زاویه بین هر لبه با لبه‏های مجاور در دو سمت آن وضعیت آن لبه را مشخص می‏کنند. طبق قرارداد به وضعیت هر لبه در هر سمت یک بیت اختصاص داده می‏شود. در صورتی که زاویه بین Front جاری با Front مجاورش در آن سمت کمتر از 135 درجه باشد بیت مربوطه یک خواهد شد و در غیر اینصورت بیت مربوطه صفر خواهد شد. در شکل زیر اگر در هر حالت لبه ضخیم‏تر را Front مورد بررسی در نظر بگیرید وضعیت‏های مختلف نشان داده شده‏اند.

|  |
| --- |
|  |

1. وضعیت‏های مختلفی برای یک Front

به طور کلی چهار حالت وجود خواهد داشت که در ‏جدول (1) مقادیر متناظری برای آنها در نظر گرفته شده‏اند که از این مقادیر در پیاده‏سازی استفاده شده‏اند.

1. وضعیت‏های مختلف به همراه مقادیر متناظر آنها

|  |  |
| --- | --- |
| **وضعیت Front** | **مقدار عددی متناظر** |
| 0-0 | 0 |
| 0-1 | 1 |
| 1-0 | 2 |
| 1-1 | 3 |

## انتخاب Front مناسب بر اساس اولویت برای پردازش

انتخاب یک Front برای پردازش در ابتدا از میان لبه‏هایی خواهد بود که در پایین‏ترین سطح قرار دارند و لیستی که حاوی لبه‏های پایین‏ترین سطح است لبه‏ای که در بالاترین وضعیت قرار دارد بیشترین اولویت را دارد. به طور کلی اولویت انتخاب لبه از میان لبه‏هایی که همسطح هستند در جدول زیر نشان داده شده است.

1. اولویت پردازش لبه‏ها در هر وضعیت

|  |  |
| --- | --- |
| **وضعیت** | **اولویت** |
| 1-1 | 3 (بیشترین) |
| 0-1 | 2 |
| 1-0 | 1 |
| 0-0 | 0 (کمترین) |

همان طور که مشاهده می‏کنید لبه‏هایی که بیشترین اولویت را دارند لبه‏هایی هستند که با کمترین عملیات امکان تبدیل شدن به چهارضلعی را دارند. به عنوان مثال لبه‏هایی که در وضعیت 1-1 یا 3 قرار دارند فقط به یک ضلع بالایی برای تبدیل شدن به یک چهارضلعی نیاز دارند.

## ایجاد ضلع‏های کناری

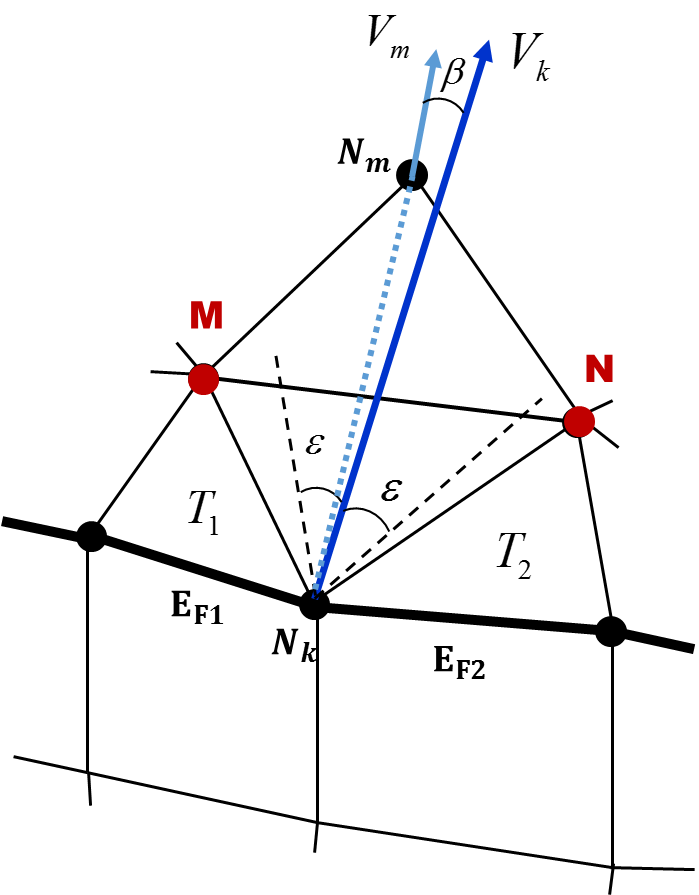
برای ایجاد ضلع‏های کناری سه روش در الگوریتم معرفی شده‏اند. هر کدام از این روش‏ها در شرایط خاصی بکار گرفته می‏شوند. با در نظر گرفتن  به عنوان نیمساز زاویه بین دو لبه Front، روش‏ها به شرح زیر هستند:

1. **یک لبه موجود** به عنوان لبه کناری در نظر گرفته شود. برای اینکه یک لبه موجود را بتوان به عنوان لبه کناری در نظر گرفت بایستی همانطور که در ‏شکل (18) نشان داده شده است لبه مورد نظر کمترین فاصله زاویه‏ای را با نیمساز زاویه بین Front جاری و Front مجاور در نقطه مشترک آنها داشته باشد. بنابر قرارداد، مقدار  به عنوان بازه قابل قبول در نظر گرفته شده است.

|  |
| --- |
|  |

1. انتخاب ضلع موجود به عنوان ضلع کناری
2. **swap کردن اتصال** موجود بین نقاط N و M از مثلث‏های T1 و T2 که مجاور Front های مورد نظر هستند. در این روش اتصال  به عنوان ضلع کناری انتخاب می‏شود. این روش در ‏0 نشان داده شده است. چنین انتخابی به طول این اتصال و همچنین اندازه زاویه‏  که بردار  با بردار  ایجاد می‏کند وابسته است. شرایط لازم برای انجام این روش را به طور خلاصه می‏توان در معادله ‏(1) توصیف کرد:

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



1. استفاده از Swap برای انتخاب لبه کناری
2. **split کردن اتصال** موجود بین نقاط N و M از مثلث‏های T1 و T2 که مجاور Front های مورد نظر هستند. این روش در صورتی مورد استفاده قرار می‏گیرد که شرایط مورد نیاز برای انجام روش‏های قبلی مهیا نباشد. به طور کلی شرایط انتخاب روش‏های مختلف در معادله ‏(2) نشان داده شده است:

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

در این روش نقطه جدید  به شبکه اضافه می‏شود و همانطور که در ‏شکل (20) نشان داده شده است دو المان مثلثی به چهار المان تقسیم می‏شوند. اتصال  به عنوان ضلع کناری در نظر گرفته می‏شود.

|  |
| --- |
|  |

1. روش Split برای انتخاب ضلع کناری

## ایجاد ضلع بالایی

عملیات ساختن ضلع بالایی به مسئله متصل کردن دو نقطه در شبکه تعبیر می‏شود که در اینجا دو نقطه مورد نظر نقاط انتهایی ضلع‏های کناری هستند و به یک اتصال برای شکل دادن به چهارضلعی جدید نیاز دارند. ایجاد چنین اتصالی با معاوضه سیستماتیک لبه‏هایی که در مسیر اتصال مورد نظر قرار می‏گیرند انجام می‏شود. به طور مفصل، این مسئله به دو مسئله قابل تقسیم است: 1- یافتن لبه‏هایی در شبکه که در مسیر اتصال مورد نظر قرار دارند. و 2- معاوضه سیستماتیک آنها تا حصول اتصال نهایی مورد نظر.

همان طور که در [2] اشاره شده است به این مسئله در [7]–[9] پرداخته شده است. برای درک بهتر نحوه حل این مسئله به ‏شکل (21) دقت کنید. در این شکل اتصال بین نقاط  و  با معاوضه کردن سیستماتیک برخی از لبه‏ها ایجاد می‏شود.

|  |
| --- |
| C:\Users\Kourosh\Desktop\EdgeRecovery.PNG |

1. مراحل ایجاد اتصال از a تا d

نحوه ایجاد اتصال در قالب دو الگوریتم بیان شده است. الگوریتم اول که در ‏جدول (3) آورده شده است فرایند ایجاد اتصال را انجام می‏دهد.

1. فرایند ایجاد لبه یا اتصال

|  |
| --- |
| Algorithm 1 |
| C:\Users\Kourosh\Desktop\alg1.PNG |

الگوریتم 1 از الگوریتم 2 که در ‏جدول (4) آورده شده است برای یافتن لبه‏هایی که در مسیر اتصال مورد نظر قرار می‏گیرند استفاده می‏کند.

1. الگوریتم یافتن لبه‏هایی مورد نیاز برای معاوضه

|  |
| --- |
| Algorithm 2 |
| C:\Users\Kourosh\Desktop\alg2.PNG |

توجه داشته باشید که عملیات ایجاد اتصال بین دو نقطه مشخص در اکثر عملیات‏های اصلی الگوریتم Q-Morph مورد نیاز است.

## تولید چهارضلعی

ایجاد چهارضلعی به معنی شناسایی و حذف المان‏های مثلثی می‏باشد که در بین چهار ضلع تعریف شده برای چهارضلعی قرار گرفته‏اند. برای انجام این کار می‏توان از المان مثلثی مجاور ضلع پایه یا همان Front شروع کرد و به ترتیب هر مثلث را حذف کرد و به صورت بازگشتی به المان‏های مجاور آن پیشروی کرد. به رفتن از یک المان به المان مجاور بایستی تا زمانیکه به یکی از ضلع‏های چهارضلعی می‏رسیم ادامه داد. به این صورت تمامی المان‏های مثلثی محصور بین اضلاع چهارضلعی جدید حذف می‏شوند و نمونه پیش‏ساخته[[6]](#footnote-6) چهارضلعی جدید ایجاد می‏شود.

## بهبود کیفیت چهارضلعی جدید

در الگوریتم Q-Morph در دو بخش بهبود کیفیت المان‏ها به وسیله تغییر مکان نودها صورت می‏گیرد. در بخش اول پس از ایجاد هر چهارضلعی بلافاصله کیفیت آن و حتی المان‏های مثلثی مجاور آن در صورت وجود بهبود می‏یابد که این کار با استفاده از الگوریتم‏های پیشنهادی در [10] و [11] انجام می‎‏شود. بخش دوم که در آن بهبود کیفیت المان‏ها با تغییر مکان نودها انجام می‏شود مربوط به زمانی است که تمامی شبکه مثلثی به المان‏های چهارضلعی تبدیل شده‏اند و همچنین پاکسازی توپولوژیکی شبکه انجام شده باشد که در ادامه به این مقوله‏ها خواهیم پرداخت. در اینجا هدف بررسی نحوه بهبود کیفیت چهارضلعی جدید و المان‏های مثلثی مجاور آن است.

بهبود کیفیت المان‏ها با تغییر مکان نودها اصطلاحاً هموارسازی[[7]](#footnote-7) نامیده می‏شود. برای هموارسازی چهارضلعی‏های جدید، از روش پیشنهادی بلکر و همکاران [10] استفاده می‏شود. برای جابجایی نود  فرض کنید  برداری از مبدا تا نود  باشد. با فرض اینکه نود  به  المان چهارضلعی متصل است. با در نظر گرفتن  ،  و  به عنوان بردارهایی از مبدا به نودهای  ،  و  از  امین المان به طوری که نودها به صورت ساعتگرد یا پادساعتگرد قرار گرفته باشند می‏توان بردار جدید  را که به مکان جدید نود  اشاره دارد با استفاده از معادله زیر محاسبه کرد:

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

با فرض اینکه بردار  میزان تغییر مکان نود  باشد می توان به سادگی آنرا از معادله زیر محاسبه کرد:

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

این هموارسازی که هموارسازی ایزوپارامتریک[[8]](#footnote-8) نامیده می‏شود نودها را به گونه‏ای جابجا می‏کند که المان به سمت متوازی‏الاضلاع شدن میل می‏کند و ضرورتاً اندازه و مربع بودن المان را حفظ نمی‏کند. این مسئله زمانی که نود مورد نظر دقیقاً به دو المان چهارضلعی متصل است مشکل‏ساز خواهد شد. برای چنین نودهایی اصلاحاتی به هموارسازی ایزوپارامتریک اعمال می‏شوند.

مطابق ‏شکل (22) فرض کنید نود  نود مقابل نود  باشد که بر روی ضلع مشترک بین دو چهارضلعی قرار دارد. با فرض اینکه  طول ایده‏ال لبه  و  طول بردار  از  به مکان جدید نود  باشد بردار میزان تغییر موقعیت  با معادله ‏(5) محاسبه می‏شود.

|  |
| --- |
|  |

1. تعیین مکان نودهای متصل به دو المان

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

این تنظیم را می‏توان تنظیم طولی بردار  در نظر گرفت که نیاز به تنظیم زاویه‏ای دارد. این تنظیم می‏تواند در حفظ اندازه المان کمک کند. برای تعیین تنظیم زاویه‏ای مطابق شکل (25) نودهای  و  به ترتیب نودهای قبلی و بعدی نود  هستند. اگر  ،  و  بردارهایی از نود  به ترتیب به نودهای  ،  و  باشند بردار جدید  به عنوان نیمساز زاویه بین  و  تعریف می‏شود و همچنین بردار  به عنوان نیمساز زاویه بین  و  تعریف می‏شود. نقطه Q به عنوان محل برخورد امتداد دادن بردار  با اتصال حاصل از وصل کردن نقاط  و  در نظر گرفته می‏شود. اگر  طول فاصله نقطه Q از  باشد و همچنین  طول ایده‏ال لبه  باشد طول بردار  بر اساس طول نسبی طول‏های  و  مطابق معادله ‏(6) محاسبه می‏شود.

|  |
| --- |
|  |

1. تنظیم زاویه‏ای نود مورد نظر

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

تغییر موقعیت برای هموارسازی زاویه‏ای  از معادله زیر محاسبه می‏شود:

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

در نهایت برای نودهایی که دقیقاً به دو المان مثلثی متصل هستند بردار تغییر مکان  از معادله زیر بدست خواهد آمد:

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

برای تعیین مقدار ایده‏ال  در حالتی که  که نسبت اندازه بزرگترین به کوچکترین لبه Front است کمتر از 2.5 باشد از روش بلکر [10] بدون اصلاح کردن استفاده می‏شود. در صورتی که بین 2.5 تا 20 باشد مطابق ‏شکل (24) از معادله ‏(9) استفاده می‏شود.

|  |
| --- |
| C:\Users\Kourosh\Desktop\a1.PNG |

1. تعیین اندازه 

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

برای حالتی که مقدار  بیشتر از 20 باشد از میانگین طول لبه‏های متصل به نود  استفاده می‏شود. برای نودهای مجاور نودهای چهارضلعی که متعلق به المان‏های مثلثی هستند از روش هموارسازی لاپلاسین [11] استفاده می‏شود که در این روش هر نود به مختصات میانگین مختصات نقاط مجاورش طبق معادله زیر منتقل می‏شود:

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

که در آن N تعداد رئوس مجاور نود i، و  مکان راس jام و  مکان جدید راس iام است.

ممکن است حین انجام عملیات هموارسازی برخی از المان‏ها وارونه شوند. برای ادامه یافتن عملیات الگوریتم Q-Morph لازم است که هیچ المانی وارونه نباشد. برای اطمینان حاصل کردن از اینکه هیچ المانی وارونه نمی‏شود بایستی بعد از جابجا کردن هر نود المان‏های متصل به آن بررسی شوند و در صورتی که المان وارونه وجود داشت نود بر روی برداری که از مکان اولیه به مکان جدید کشیده شده است به صورت تدریجی حرکت داده شود تا زمانی که دیگر هیچ المانی وارونه نباشد.

برای پیاده‏سازی این راه‏حل پیشنهادی در صورتی که المان وارونه وجود داشته باشد ابتدا نود مربوطه را به وسط بردار مذکور انتقال می‏دهیم. دو حالت پیش می‏آید:

1. هنوز المان وارونه وجود دارد: در این حالت با نزدیکتر کردن نقطه به مکان اولیه سعی در رفع المان وارونه می‏شود.
2. هیچ المانی وارونه نیست: در این حالت با حرکت دادن نقطه به سمت مکان جدید تا حد ممکن هموارسازی انجام می‏شود.

برای حرکت دادن نقاط به سمت نقطه اولیه یا نقطه جدید به ترتیب از مقادیر رشته‏های عددی در ‏شکل (25) و ‏شکل (26) به عنوان ضریبی برای طول بردار مذکور در هر حرکت استفاده می‏شود.

|  |
| --- |
|  |

1. رشته عددی مورد استفاده برای حرکت به سمت نقطه اولیه

|  |
| --- |
|  |

1. رشته عددی مورد استفاده برای حرکت به سمت نقطه جدید

برای درک بهتر نحوه حرکت دادن نقطه به شکل زیر توجه کنید.

|  |
| --- |
|  |

1. نحوه حرکت دادن و تغییر مکان نقطه

مسئله چگونگی تشخیص وارونگی هر المان بسته به مثلثی یا چهارضلعی بودن متفاوت است. در المان‏های مثلثی با در نظر گرفتن دو ضلع آن به عنوان دو بردار هم مبدا و با استفاده از علامت ضرب خارجی آنها می‏توان وارونگی را تشخیص داد ‏شکل (28). از آنجاییکه المان‏های شبکه به صورت پادساعتگرد تعریف شده‏اند در صورت منفی شدن علامت ضرب خارجی المان وارونه است و بلعکس.

|  |
| --- |
|  |

1. محاسبه وارونگی المان‏های مثلثی

برای المان‏های چهارضلعی مسئله پیچیده می‏شود. برای المان‏های چهارضلعی دو ضلع روبرو را در نظر می‏گیریم. در دو مرحله هر بار یکی از اضلاع چهارضلعی را به عنوان یک ضلع از دو مثلث که راس سوم آنها نودهای ضلع دیگر چهارضلعی است در نظر می‏گیریم و وارونگی مثلث‏ها را محاسبه می‏کنیم. در نهایت با در دست داشتن وضعیت وارونگی چهار المان مثلثی در صورتی که بیش از یکی از آنها وارونه باشد به این معنی است که چهارضلعی وارونه است.

|  |
| --- |
|  |

1. تعیین وارونگی المان چهارضلعی

به عنوان مثال در شکل بالا با در نظر گرفتن ضلع‏های AB و CD به عنوان ضلع‏های روبرو بایستی وضعیت وارونگی مثلث‏های ABC، ABD، CDA و CDB را محاسبه کرد و در نهایت وضعیت وارونگی چهارضلعی را مشخص کرد.

## بروزرسانی و دسته‏بندی مجدد لبه‏های Front

ایجاد هر چهارضلعی جدید موجب می‏شود که حداقل یک لبه‏ Front حذف شود و لبه یا لبه‏های جدیدی که اضلاع چهارضلعی جدید هستند به عنوان لبه Front به مجموعه لبه‏ها اضافه شود. اضافه و کم کردن لبه‏ها بروزرسانی لبه‏ها نامیده می‏شود. بروزرسانی لبه‏ها به تنهایی کافی نیست و از آنجاییکه بر روی المان‏های جدید عملیات هموارسازی انجام می‏شود ممکن است وضعیت Front ها تغییر کند. از اینرو دسته‏بندی مجدد لبه‏های Front ضروری است.

## عملیات ویژه Seam

ممکن است در فرایند تبدیل در الگوریتم Q-Morph شرایطی پیش بیاید که در آن زاویه بین دو لبه Front بسیار کم باشد در چنین حالتی به جای اجرای روند لازم برای تولید چهارضلعی عملیات Seam انجام می‏شود و اصطلاحاً درز موجود از بین می‏رود. در دو حالت مشخص شده در معادله زیر معیارهای لازم برای اجرای این عملیات آورده شده است:

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

در اینجا  تعداد المان‏های چهارضلعی متصل به نقطه مشترک دو لبه Front است. در شکل زیر عملیات Seam نشان داده شده است.

|  |
| --- |
| C:\Users\Kourosh\Desktop\seam.PNG |

1. عملیات بستن درز یا Seam

## عملیات ویژه Transition Seam

در صورتی که درز وجود داشته باشد اما نسبت اندازه دو لبه Front نسبت به هم بیش از 2.5 باشد بایستی به جای عملیات Seam از عملیات Transition Seam استفاده کرد که در شکل زیر مراحل آن نشان داده شده است. در این عملیات ابتدا Front بزرگتر نصف می‏شود و در مرکز آن یک نقطه جدید اضافه می‏شود و در ادامه المان چهارضلعی مجاور این لبه برش داده می‏شود و المان جدیدی در مجاورت آن به مش اضافه می‏شود. در نهایت عملیات‏های هموارسازی و دسته‏بندی مجدد اجرا می‏شوند.

|  |
| --- |
| C:\Users\Kourosh\Desktop\TransSeam.PNG |

1. عملیات Transition Seam

## عملیات ویژه Transition Split

در صورتی که شرایط زاویه‏ای بین دو Front برای عملیات‏های Seam و Transition Seam مهیا نباشد اما نسبت اندازه بین آنها بیش از 2.5 باشد از عملیات Transition Split استفاده می‏شود. در این عملیات ابتدا مرکز المان چهارضلعی مجاور لبه Front بزرگتر یافت می‏شود و این نقطه به شبکه اضافه می‏شود. سپس همانطور که در شکل زیر نشان داده شده است المان مربوطه به دو المان تبدیل می‏شود و بخشی از آن هم جزئی از یک المان چهارضلعی جدید می‏شود. در نهایت عملیات هموارسازی بر روی این المان‏ها انجام می‏شود.

|  |
| --- |
| C:\Users\Kourosh\Desktop\TransSplit.PNG |

1. عملیات Transition Split

## تشخیص متقاطع بودن دولبه

مسئله‏ای که در الگوریتم Q-Morph بسیار پرکاربرد و مهم است تعیین وضعیت دو لبه نسبت به هم است که معادل تشخیص وضعیت دو پاره‏خط نسبت به هم است. در این مسئله چون با دو پاره‏خط سروکار داریم نمی‏توان با آنها همانند دو خط رفتار کرد و مسئله دشوارتر است. برای حل این مسئله از تعاریف برداری [12] برای تعریف پاره‏خط استفاده می‏کنیم.

در یک محیط دو بعدی یک خط به صورت معادله تعریف می‏شود که در آن P یک نقطه از خط و D برداری غیرصفر و t عددی حقیقی است. در صورتی که در این معادله محدودیت t ≥ 0 را در نظر بگیریم این معادله یک نیم خط خواهد بود. در صورتی که محدودیت به صورت 0 ≤ t ≤ 1 در نظر گرفته شود معادله پاره‏خط خواهد بود.

دو خط  و  که در آن s و t مقادیر حقیقی هستند یا متقاطع‏اند یا موازی‏اند و یا منطبق هستند. با فرض داشتن بردارهای  و و تعریف عملیات ضرب نقطه‏ای ویژه دو بردار در معادله زیر داریم:

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

اگر  باشد به این معنی است که دو خط در یک نقطه متقاطع هستند و این نقطه با مقادیر مشخص شده در معادله زیر محاسبه می‏شود:

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

در صورتی که دو لبه را به صورت پاره‏خط در نظر بگیریم با بررسی کردن محدودیت‏های مقادیر s و t می‏توان متقاطع بودن یا نبودن آنها را مشخص کرد.

## نوآوری‏ها

با نگاهی به عملیات‏های مورد استفاده در الگوریتم Q-Morph می‏بینیم که در بسیاری از عملیات ایجاد اتصال بین دو نقطه لازم خواهد بود. در مثالی که در شکل زیر نشان داده شده است امکان انتخاب لبه موجود به عنوان لبه کناری وجود ندارد و باید یکی از عملیات‏های Swap یا Split انجام شوند. برای انجام هر کدام از این عملیات‏ها بایستی اتصال بین نقاط M و N ایجاد شود که در اینجا شدنی نیست زیرا چنین اتصالی المان چهارضلعی را برش می‏زند. همان طور که در خطوط 7 و 17 در الگوریتم‏ ارائه شده در ‏جدول (4) نشان داده شده است امکان شکست خوردن این الگوریتم وجود دارد.

|  |
| --- |
|  |

1. وضعیتی که در آن امکان ایجاد ضلع کناری با روش‏های پیشنهاد شده وجود ندارد

برای رفع این مشکل ما از عملیات برش ویژه‏ای استفاده می‏کنیم که با توجه به وضعیت Front عمل می‏کند. در صورتی که Front مورد نظر در وضعیت 0-0 قرار گرفته باشد مطابق حالت 1 در ‏شکل (34) یک لبه عمود در نود مربوطه ایجاد می‏شود. در وضعیت‏های 0-1 و 1-0 مطابق حالت 2 در ‏شکل (34) یک لبه موازی با لبه کناری دیگر ایجاد می‏شود. علت ایجاد لبه موازی با لبه کناری پیشگیری کردن از برخورد پیدا کردن دو لبه کناری است. ممکن است موارد کمی وجود داشته باشند که در آن موارد ایجاد ضلع کناری عمود یا موازی ممکن نباشد. در این حالت‏ها نزدیک‏ترین لبه به عنوان لبه کناری انتخاب می‏شود.

|  |
| --- |
|  |

1. راه حل پیشنهادی برای ایجاد ضلع کناری

ایجاد اتصال بین نقاط در سایر عملیات‏های اصلی از جمله ایجاد ضلع بالایی و عملیات‏های ویژه نیز کاربرد دارد و از اینرو بروز شرایطی مشابه شرایط ‏شکل (33) موجب می‏شود الگوریتم Q-Morph عملاً از کار بیفتد.

در حین اجرای روند تبدیل شدن شبکه مثلثی به شبکه چهارضلعی ممکن است المان‏هایی که ایجاد می‏شوند علیرغم اعمال عملیات هموارسازی از کیفیت بالایی برخوردار نباشند و المان ایجاد شده ضمن اینکه وارونه نیست اما مقعر[[9]](#footnote-9) باشد. چنین المان‏هایی یک زاویه بزرگتر از 180 درجه دارند و مرکز آنها خارج از آنها قرار می‏گیرد. قرار گرفتن مرکز چنین المان‏هایی در خارج از آنها زمانیکه در شرایط عملیات Transition Split قرار می‏گیرند مشکل‏ساز می‏شود. نمی‏توان برش‏های لازم برای این عملیات را بر روی چنین المانی اعمال کرد.

راه حل پیشنهادی برای غلبه کردن بر چنین مشکلاتی اضافه کردن عملیاتی جایگزین در تمامی موارد مذکور است که بتواند بدون شکست خوردن انجام بپذیرد. ما از روشی مشابه روش پیشنهادی در [13] با کمی تغییر استفاده می‏کنیم. در این روش یک چهارضلعی از ادغام کردن دو مثلث یا تقسیم یکی و ادغام آنها ایجاد می‏شود. تفاوت کار ما در اینجا حذف کردن مرحله اصلاح شبکه پیش از ادغام کردن مثلث‏هاست که هدف از آن ایجاد چهارضلعی‏های بهتر است. ما به جای عملیات اصلاح شبکه که با اضافه کردن المان انجام می‏شود از عملیات هموارسازی پس از ادغام استفاده می‏کنیم.

## پاکسازی توپولوژیکی

یکی از ویژگی‏های منفی الگوریتم‏های غیرمستقیم برای ایجاد شبکه‏های چهارضلعی این است که معمولاً این روش‏ها نودهای نامعمول زیادی به شبکه اضافه می‏کنند. نودهای نامعمول در شبکه‏های چهارضلعی به نودهایی اتلاق می‏شوند که درجه آنها کمتر یا بیشتر از چهار است. هرچند که به واسطه تغییر اندازه المان‏های مجاور در شبکه وجود چنین نودهایی ضروری است اما زیاد بودن آنها بر کیفیت شبکه تاثیر منفی دارد. الگوریتم Q-Morph هم به عنوان یک روش غیر مستقیم از مجموعه الگوریتم‏های ارائه شده در [4]–[6] برای شناسایی و ترمیم الگوهای ساختاری نامناسب ایجاد شده در شبکه استفاده می‏کند. با تمرکز بر تعاریف موجود در [4] برای نوع عملیات‏ها این عملیات‏ها به سه دسته عملیات‏های اساسی، عملیات‏های چند گامی و عملیات‏های مرزی دسته‏بندی می‏شوند.

مسئله‏ای که در منابع به آن پرداخته نشده است ترتیب اجرای آنها و همچنین تعداد تکرارهای مورد نیاز برای حصول اطمینان از بررسی شدن تمامی الگوهاست. برای حل این مشکلات ما از شبه کد پیشنهادی در ‏جدول (5) استفاده می‏کنیم.

با استفاده از این روند پیشنهادی نیازی به تعیین کردن تعداد تکرارها نخواهد بود و همچنین استفاده از عملیات‏ها با اولویت قرار دادن عملیات‏های اساسی بر عملیات‏های چندگامی و در نهایت عملیات‏های مرزی در بررسی کردن بخش‏های شبکه روند مناسبی خواهد بود.

1. شبه کد پیشنهادی برای اجرای الگوریتم‏های پاکسازی توپولوژیکی

|  |
| --- |
| **Algorithm: Topological Cleanup**  **Input:** Quadrilateral Mesh (Me)  **Output:** Quadrilateral Mesh (Me)  1: begin  2: do while no more patterns found  3: do  4: if (basic pattern found) then  5: remove pattern  6: exit loop  7: else if (multiple operation pattern found) then  8: remove pattern  9: exit loop  10: else if (boundary pattern found) then  11: remove pattern  12: exit  13: end if  14: end do  15: end while  16: end |

## هموارسازی کلی

هموارسازی کلی[[10]](#footnote-10) در الگوریتم Q-Morph با استفاده از الگوریتم ترکیبی ارائه شده در [1] انجام می‏شود. این الگوریتم ترکیبی از روش هموارسازی لاپلاسی محدود[[11]](#footnote-11) (CLS) به همراه روش هموارسازی مبتنی بر بهینه‏سازی[[12]](#footnote-12) (OBS) در جهت بهبود کیفیت شبکه استفاده می‏کند.

در روش CLS برای تعیین مکان هر نود ابتدا مرکز چندضلعی حاصل از وصل کردن نقاط مجاور آن با استفاده از روش ارائه شده در [14] محاسبه می‏شود. در این روش با فرض اینکه چند ضلعی از مجموعه نقاط  تشکیل شده باشد مرکز آن نقطه  از روابط معادله زیر بدست می‏آید:

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

که در آن A برابر است با مساحت چندضلعی و از معادله زیر قابل محاسبه است:

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

پس از محاسبه مرکز چندضلعی نود مورد نظر به مرکز چندضلعی انتقال داده می‏شود و با بررسی کردن معیارهایی برای پذیرش محل مورد نظر نسبت به اصلاح یا پذیرش آن اقدام می‏شود. در یک حلقه با 20 تکرار هر بار در صورت پذیرش نشدن مکان مورد نظر فاصله نقطه اولیه تا مرکز چندضلعی نصف می‏شود و نقطه به آنجا منتقل می‏شود. پس از اعمال الگوریتم CLS در دو تکرار بر روی تمامی نودهای شبکه الگوریتم OBS اجرا می‏شود.

الگوریتم OBS یک الگوریتم گرادیان کاهشی تکرارشونده است که نود x را در جهات گرادیانی به منظور افزایش کیفیت کم‏کیفیت‏ترین المان متصل به آن جابجا می‏کند. برای تعیین مسیر گرادیانی تقریبی بایستی مختصات x را در جهات مختلف به اندازه δ تحریک کرد. تحریک به اضافه و کم کردن تصادفی مختصات نقطه x در هر محور انجام می‏شود. بینهایت جهت برای تغییر مکان نود N وجود دارد که به عنوان مثال 8 جهت مختلف در ‏شکل (35) نشان داده شده است.

|  |
| --- |
|  |

1. مسیرهای مورد نظر برای تحریک مختصات نود

# پیاده‌سازی و زیربرنامه‌های مورد استفاده

در ادامه زیربرنامه های بکار رفته برای پیاده سازی توضیح داده می شود. برای مطالعه مستندات برخی از زیربرنامه ها باید به مستندات آن زیربرنامه مراجعه شود.

## برنامه اصلی QMorph

در برنامه اصلی موارد زیر به ترتیب انجام می­شود:

1. عملیات‏های پیش پردازش

مقداردهی اولیه متغیرها در ابتدا انجام می‏شود و در ادامه با فراخوانی تابع Initialize ساختارهای داده‏ای مورد نیازمقداردهی اولیه می‏شوند. در ادامه با فراخوانی زیربرنامه Read\_2DMeshC فایل ورودی خوانده شده و تمام اطلاعات موجود در این فایل از کاربر گرفته می­شود. اطلاعاتی از جمله تعداد نقاط، تعداد المان‏ها، نقاط تشکیل دهنده المان‏ها، اطلاعات مربوط به همسایه‏ها و نقاط مرزی و غیره خوانده می‏شوند. با فراخوانی زیربرنامه Get\_FrontEdges لبه‏های Front شناسایی و اطلاعات آنها ذخیره می‏شوند. زیربرنامه CheckContinuity برای بررسی ساختار لبه‏های Front و پیوستگی حلقه‏وار آنها استفاده می‏شود. در صورت پیوسته بودن لبه‏ها متغیر Continuity مقدارش True خواهد بود.

1. تعیین وضعیت لبه‏ها و تعیین لبه‏های مرزی

با فراخوانی زیربرنامه UpdateStates وضعیت لبه‏ها مشخص می‏شود. در ادامه در یک حلقه تعداد لبه‏های مرزی شمارش می‏شوند.

1. تبدیل شبکه مثلثی به شبکه چهارضلعی

در یک حلقه تکرار بینهایت در هر دور با فراخوانی زیربرنامه ParseLists لیست لبه‏ها بررسی می‏شود و یک لبه انتخاب و پردازش می‏شود. در هر دور با فراخوانی تابع CurrentLevelCompleted پردازش لبه‏های هر سطح برای اطمینان از کامل شدن آن سطح و رفتن به سطح بعد بررسی می‏شود. این کار باعث می‏شود که تمامی لبه‏های سطح جاری پردازش شوند و ایجاد المان‏های چهارضلعی به صورت سطح به سطح انجام شود. شرط خروج از حلقه بینهایت، پردازش شدن تمامی لبه‏هاست و این شرط با فراخوانی تابع AllFrontsProcessed بررسی می‏شود.

1. پاکسازی توپولوژیکی و هموارسازی کلی شبکه

پس از بررسی پردازش شدن تمامی لبه‏ها که به معنی تولید شدن شبکه چهارضلعی است با فراخوانی زیربرنامه TopologicalCleanUp پاکسازی توپولوژیکی انجام می‏شود و با فراخوانی زیربرنامه GlobalSmooth هموارسازی کلی بر روی تمامی المان‏های شبکه انجام می‏شود و کیفیت شبکه بهبود می‏یابد. با فراخوانی زیربرنامه AddElements برای حذف نودهایی که درجه آنها بزرگتر یا مساوی 6 است با اضافه کردن المان‏هایی اقدام می‏شود.

1. چاپ نتایج

با استفاده از زیربرنامه Write\_2DMeshC تمام خروجی­های برنامه در فایل­های خروجی چاپ می­شود.

# مراجع

[1] S. A. Canann, J. R. Tristano, and M. L. Staten, “An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes.,” in *IMR*, 1998, pp. 479–494.

[2] S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal, “Q-Morph: an indirect approach to advancing front quad meshing,” *Int. J. Numer. Methods Eng.*, vol. 44, no. 9, pp. 1317–1340, 1999.

[3] A. A. Rushdi, S. A. Mitchell, C. L. Bajaj, and M. S. Ebeida, “Robust All-quad Meshing of Domains with Connected Regions,” *Procedia Eng.*, vol. 124, pp. 96–108, 2015.

[4] S. A. Canann, S. N. Muthukrishnan, and R. K. Phillips, “Topological improvement procedures for quadrilateral finite element meshes,” *Eng. Comput.*, vol. 14, no. 2, pp. 168–177, 1998.

[5] P. Kinney, “Cleanup: Improving quadrilateral finite element meshes,” in *6th International Meshing Roundtable*, 1997, pp. 437–447.

[6] M. Staten and S. A. Canann, “Post refinement element shape improvement for quadrilateral meshes,” in *220 Trends in Unstructured Mesh Generation, ASME*, 1997.

[7] N. L. Jones, “Solid modelling of earth masses for applications in geotechnical engineering,” 1991.

[8] S. W. Sloan, “A fast algorithm for generating constrained Delaunay triangulations,” *Comput. Struct.*, vol. 47, no. 3, pp. 441–450, 1993.

[9] P. L. George, F. Hecht, and É. Saltel, “Automatic mesh generator with specified boundary,” *Comput. Methods Appl. Mech. Eng.*, vol. 92, no. 3, pp. 269–288, 1991.

[10] T. D. Blacker and M. B. Stephenson, “Paving: A new approach to automated quadrilateral mesh generation,” *Int. J. Numer. Methods Eng.*, vol. 32, no. 4, pp. 811–847, 1991.

[11] D. A. Field, “Laplacian smoothing and Delaunay triangulations,” *Int. J. Numer. Methods Biomed. Eng.*, vol. 4, no. 6, pp. 709–712, 1988.

[12] D. Eberly, *Intersection of linear and circular components in 2D*. 2000.

[13] C. K. Lee and S. H. Lo, “A new scheme for the generation of a graded quadrilateral mesh,” *Comput. Struct.*, vol. 52, no. 5, pp. 847–857, 1994.

[14] P. Bourke, “Calculating the area and centroid of a polygon,” *Swinburne Univ Technol.*, 1988.

1. Local [↑](#footnote-ref-1)
2. Counter ClockWise (CCW) [↑](#footnote-ref-2)
3. Irregular Nodes [↑](#footnote-ref-3)
4. Topological Cleanup [↑](#footnote-ref-4)
5. Advancing Front [↑](#footnote-ref-5)
6. Prototype [↑](#footnote-ref-6)
7. Smooth [↑](#footnote-ref-7)
8. Isoparametric Smooth [↑](#footnote-ref-8)
9. Concave [↑](#footnote-ref-9)
10. Global Smoothing [↑](#footnote-ref-10)
11. Constrained Laplacian Smoothing [↑](#footnote-ref-11)
12. Optimization-based Smoothing [↑](#footnote-ref-12)