

****

**عنوان:**

تبدیل شبکه‌هایی با فرمت SU2 به شبکه‌هایی با ساختار سلول محور و ضلع محور دوبعدی

|  |  |  |
| --- | --- | --- |
| **نویسندگان** | مرتضی نامور |  |
| آرمین مسلمی پاک | Optimized_image_813b5ec2.png |
| **تاریخ تنظیم سند** | 1/9/1396 | |
| **شناسه سند** |  | |

**فهرست مطالب**

[فصل 1- راهنمای کاربری 1](#_Toc512597488)

[1-1- فایل ورودی 1](#_Toc512597489)

[1-2- اجرای برنامه 2](#_Toc512597490)

[1-3- فایل های خروجی 2](#_Toc512597491)

[1-4- توانایی ها و محدودیت ها 3](#_Toc512597492)

[فصل 2- اعتبارسنجی نتایج 4](#_Toc512597493)

[فصل 3- تئوری و الگوریتم 10](#_Toc512597494)

[3-1- شبکه‌های SU2 10](#_Toc512597495)

[3-2- الگوریتم تبدیل شبکه‌های SU2 به ساختار مرسوم سلول محور و ضلع محور 11](#_Toc512597496)

[فصل 4- پیاده‌سازی و زیربرنامه‌های مورد استفاده 12](#_Toc512597497)

**چکیده:**

هدف از این پروژه توسعه­ی کد محاسباتی به منظور تبدیل شبکه­هایی با فرمت \*.SU2 به شبکه­هایی با ساختار سلول محور و ضلع محور می­باشد. شبکه­هایی با پسوند \*.SU2 برای استفاده در نرم­افزار متن باز با همین نام SU2 که توسط دانشگاه استنفورد پایه­گذاری شده است، به کارگیری می­شوند. برای همین منظور توسعه­ی کد محاسباتی برای تبدیل خروجی این نرم­افزار و استفاده از آن­ها به عنوان ورودی برای کدهای توسعه یافته­ی دیگر، امری غیرقابل اجتناب می­باشد.

**کلمات کلیدی:** شبکه­ی محاسباتی، ضلع محور، سلول محور، SU2.

# راهنمای کاربری

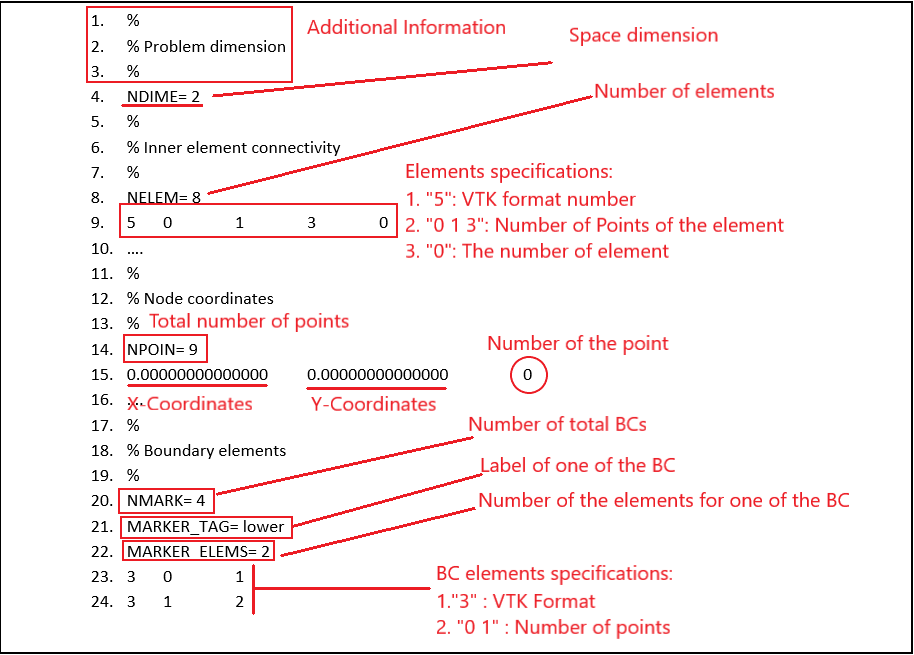
## فایل ورودی

در این برنامه از یک فایل ورودی با پسوند \*.SU2 استفاده شده است. این فایل محصول خروجی نرم­افزاری متن باز با همین نام SU2 می­باشد. برای اطلاعات بیشتر در مورد این قالب و ساختار فایل ورودی می‏توانید به مستندات زیربرنامه Read\_2DMeshSU2 مراجعه نمایید. تمام اطلاعات لازم باید در قالب یک فایل به نام Mesh.su2 تهیه شده و در پوشه حاوی برنامه قرار داده شود.

اطلاعات زیر باید در فایل ورودی وجود داشته باشد که در غیر این صورت برنامه قابل اجرا شدن نیست:

1. ابعاد فضای محاسباتی (دو بعدی یا سه بعدی)
2. تعداد المان­ها، نحوه­ی اتصال المان­ها به یکدیگر و نوع هر یک از المان­ها (مثلثی، مربعی و غیره)
3. تعداد نقاط (گره­ها) و مختصات آن­ها
4. تعداد المان­های مرزی

نام هر مرز، شماره­ی آن، تعداد المان­های آن، نوع مرز و مختصات اجزای تشکیل­دهنده­ی آن (مرز در حالت دو بعدی ضلع است و در حالت سه بعدی مثلثی، مربعی و غیره می­باشد.) می‌بایست در فایل ورودی وجود داشته باشند. نمونه­ای از این فایل در ‏شکل (1) نشان داده شده است.



1. نمونه­ای از محتویات یک فایل \*.SU2

## اجرای برنامه

این برنامه تحت نرم افزار Comaq Visual Fortran توسعه یافته است. بنابراین جهت اجرای آن بهتر است این نرم افزار را نصب نمایید. در اینجا یک پارامتر 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.

## فایل های خروجی

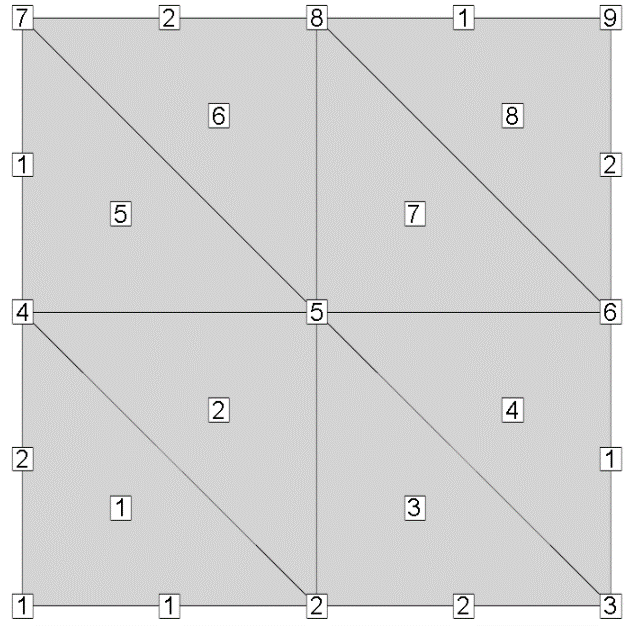
یک قالب مشخص برای چاپ اطلاعات در نظر گرفته شده است تا با ساختار فایل های ورودی سازگاری داشته باشد. به این ترتیب ارتباط بین برنامه های نوشته شده به راحتی امکان‏پذیر می­باشد. برای اطلاعات بیشتر در مورد این قالب و ساختار فایل خروجی می توانید به مستندات زیربرنامه‌های WritePlaneMesh\_cgid\_plt، WriteBoundCrv\_cgid\_plt و WriteMesh\_gid مراجعه نمایید. بنابراین اطلاعات خروجی با توجه به ساختارهای در نظر گرفته شده جهت مشاهده شبکه و مرزها به ترتیب در فایل­هایی با عناوین PlaneMesh.plt بصورت سلول محور[[1]](#footnote-1)، BoundCrv.plt شامل اطلاعات مرزهای شبکه و MeshOut.gid بصورت ضلع محور[[2]](#footnote-2) در خروجی برنامه چاپ می‌شوند.

## توانایی ها و محدودیت ها

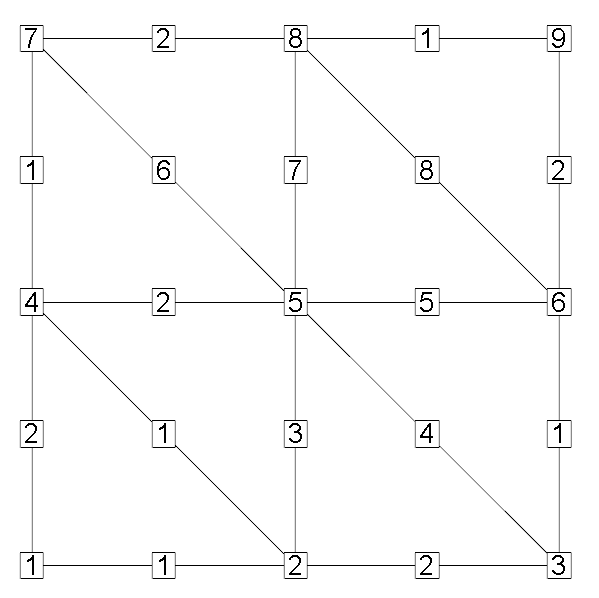
در این برنامه، شبکه­های دو بعدی با فرمت \*.SU2 به شبکه­هایی با ساختار ضلع محور و سلول محور تبدیل می­شوند. از جمله توانایی این برنامه، امکان استفاده از شبکه­های تولید شده با فرمت \*.SU2 توسط نرم­افزار متن باز برای، برنامه­های توسعه یافته­ی دیگر می­باشد. برای شبکه­های سه بعدی می­بایست از یک برنامه­ی دیگری استفاده نمود، که در حال آماده­سازی می­باشد.

# اعتبارسنجی نتایج

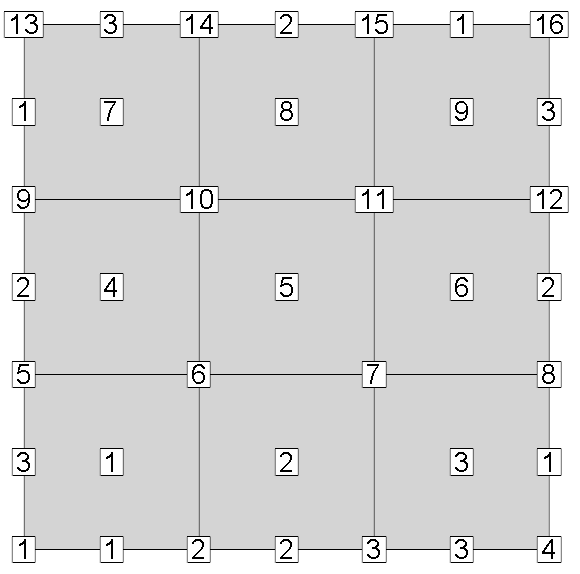
در این بخش چند نمونه از تبدیل­های انجام شده با استفاده از برنامه­ی ساخته شده ارائه می­گردد. اولین و دومین شبکه مربوط به شبکه­های ساده با ساختار مثلثی و مستطیلی می­باشند. برنامه­ی نوشته شده قابلیت تبدیل شبکه­ها با اشکال مثلثی و مستطیلی یا مرکب[[3]](#footnote-3) را دارد.



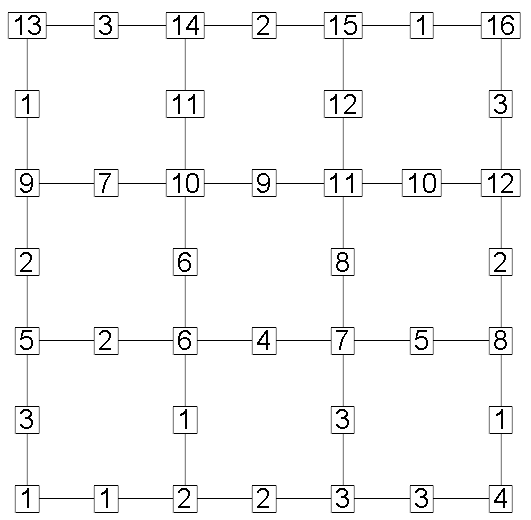
1. شبکه­ی ساده­ی مثلثی از شبکه­ی اولیه­ی با فرمت \*.SU2 با ساختار سلول­ محور



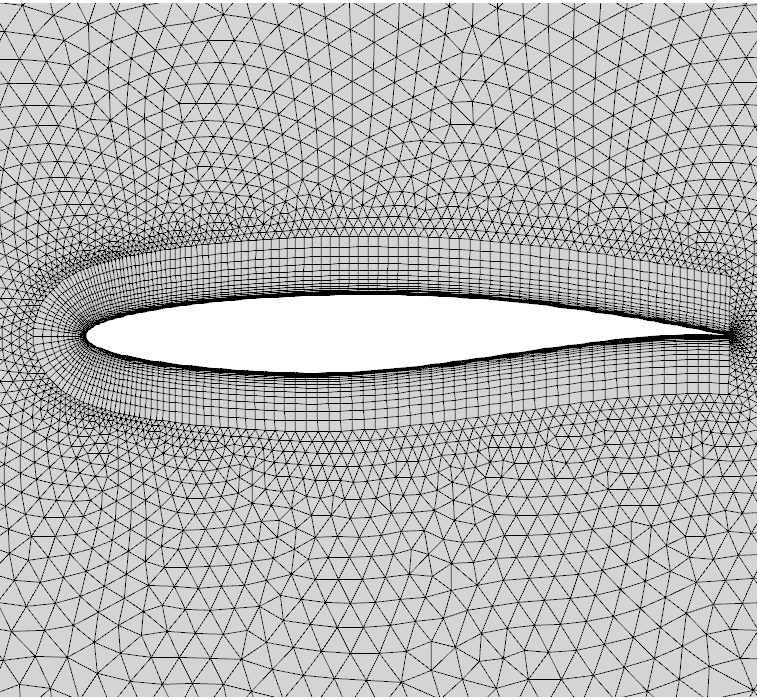
1. شبکه­ی ساده­ی مثلثی از شبکه­ی اولیه­ی با فرمت \*.SU2 با ساختار ضلع محور



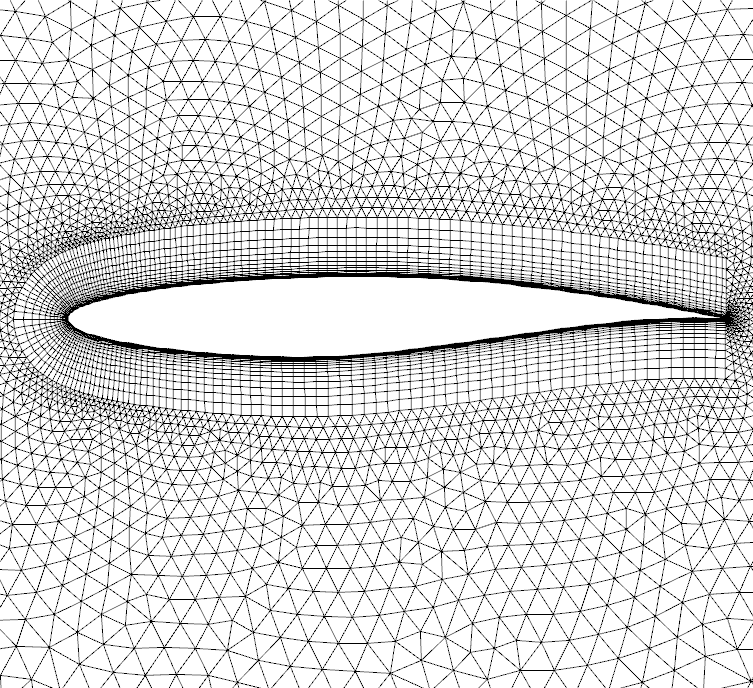
1. شبکه­ی ساده­ی مستطیلی از شبکه­ی اولیه­ی با فرمت \*.SU2 با ساختار سلول­ محور



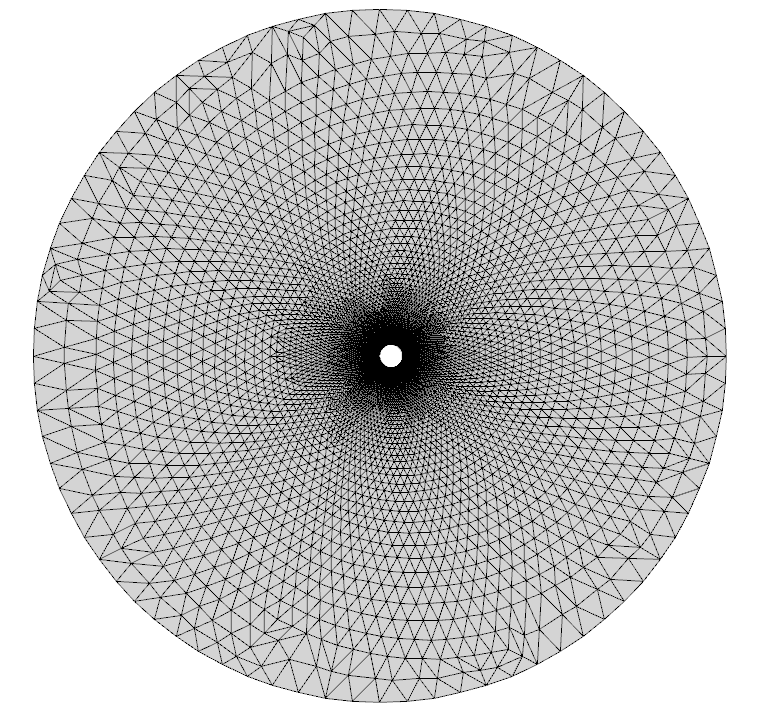
1. شبکه­ی ساده­ی مستطیلی از شبکه­ی اولیه­ی با فرمت \*.SU2 با ساختار ضلع محور



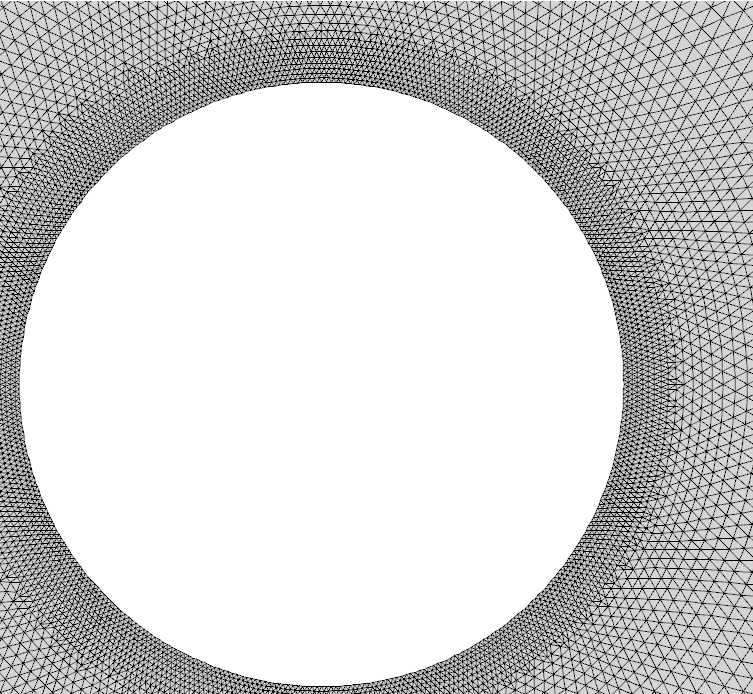
1. شبکه­ی مرکب از ایرفویل RAE2822 از شبکه­ی اولیه­ی با فرمت \*.SU2   
   با ساختار سلول­ محور



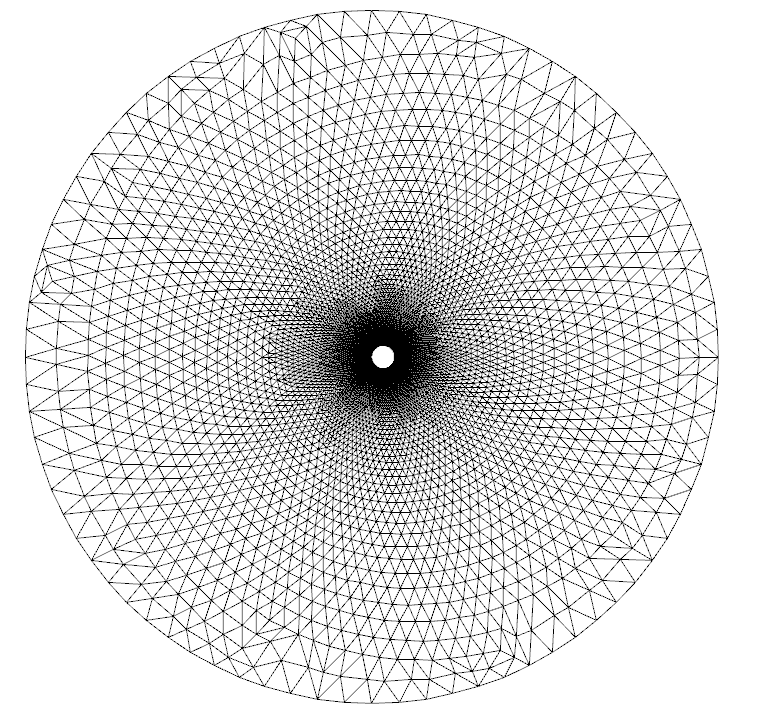
1. شبکه­ی مرکب از ایرفویل RAE2822 از شبکه­ی اولیه­ی با فرمت \*.SU2   
   با ساختار ضلع محور



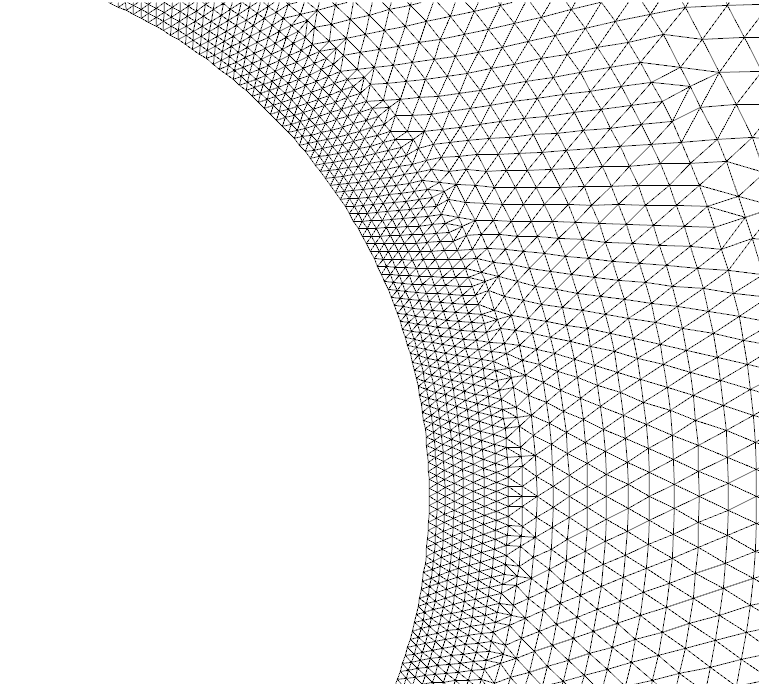
1. شبکه­ی مثلثی اطراف یک سیلندر از شبکه­ی اولیه­ی با فرمت \*.SU2 با ساختار سلول­ محور



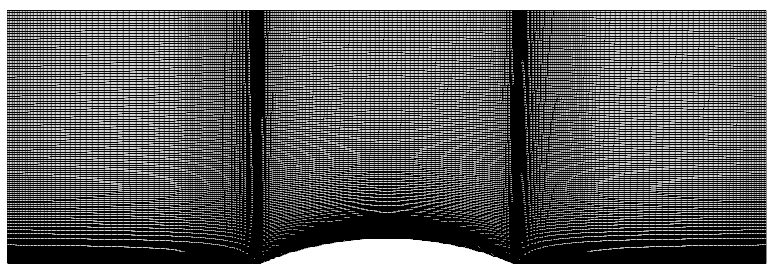
1. نمایی نزدیک­تر از شکل 8



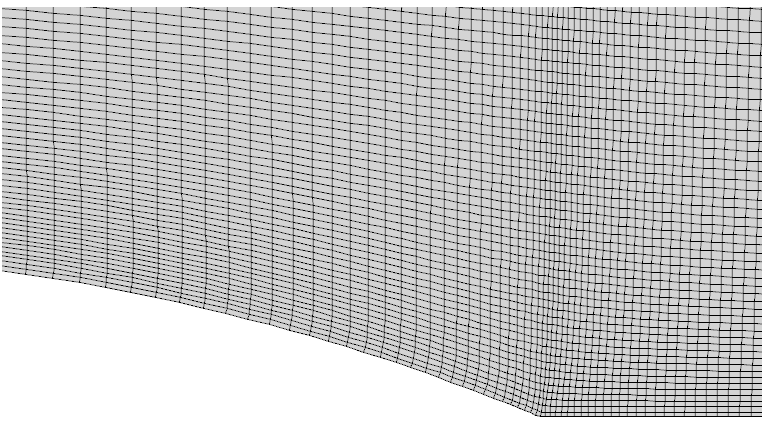
1. شبکه­ی شکل 8 با ساختار ضلع محور



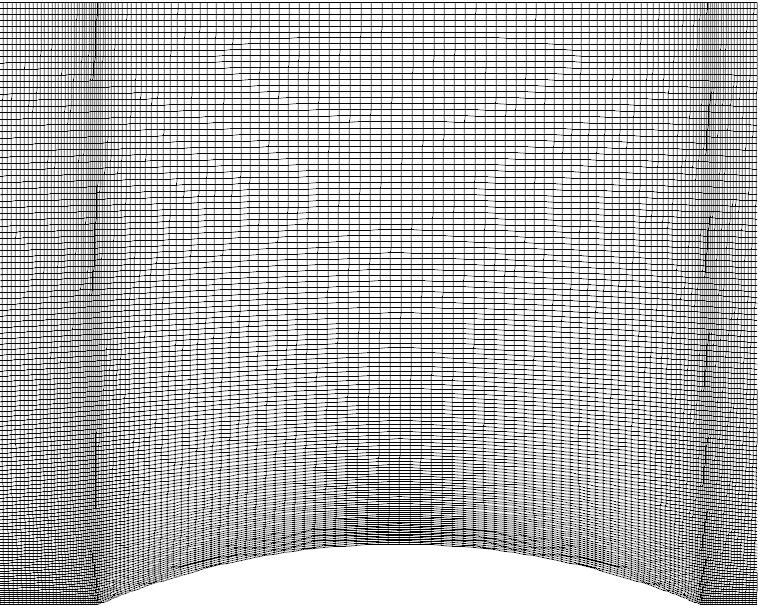
1. نمایی نزدیک­تر از شکل 10



1. شبکه­ی مستطیلی اطراف یک کانال از شبکه­ی اولیه­ی با فرمت \*.SU2   
   با ساختار سلول­ محور



1. نمایی نزدیک­تر از شکل 12



1. شبکه­ی شکل 12 با ساختار ضلع محور

# تئوری و الگوریتم

## شبکه‌­های SU2

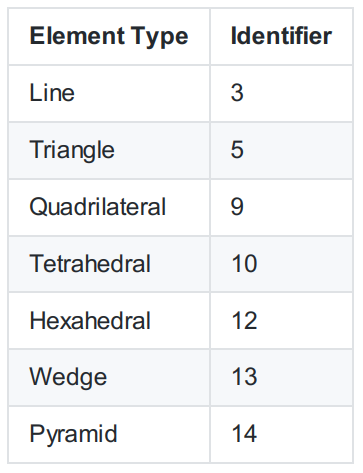
در دنیای دینامیک سیالات محاسباتی[[4]](#footnote-4) نرم­افزارهای تجاری زیادی مورد استفاده قرار می­گیرند. در کنار این نرم­افزارها که ساختار کاربر پسندانه­تری دارند، امروزه نرم­افزارهای متن باز[[5]](#footnote-5) سهم بیشتری را در میان کاربران و مهندسان خاص خود اختصاص داده است. یکی از دلایل مهم این پیشرفت، هزینه­های سنگین نرم­افزارهای تجاری و غیرقابل تغییر بودن آنها مگر به وسیله­ی سازنده­ی آن­ها می­باشد.

یکی از این نرم­افزارهای متن باز، نرم­افزاری با نام SU2 می­باشد که توسط دانشگاه استنفورد[[6]](#footnote-6) توسعه یافته است. با توجه به ساختار خاص شبکه­های ورودی و خروجی این نرم­افزار، داشتن برنامه­ای به منظور تبدیل این ساختار ویژه به ساختارهای مرسوم و بکارگیری این شبکه­ها در سایر کدهای نوشته شده، امری اجتناب­ناپذیر می­باشد. برای این منظور کدی برای تبدیل ساختارهای شبکه­ی SU2 به ساختارهای مرسوم سلول محور و ضلع محور توسعه یافته است.

## الگوریتم تبدیل شبکه‌­های SU2 به ساختار مرسوم سلول محور و ضلع محور

1. خواندن اطلاعات فایل ورودی با پسوند \*.SU2 و مقداردهی متغیرهای شبکه­ با ساختار سلول محور
2. به دست آوردن همسایه­های المان­ها
3. گرفتن خروجی­های اول و دوم برای شبکه با ساختار سلول محور
4. تبدیل ساختار شبکه­ی سلول محور به ضلع محور
5. مرتب­سازی اضلاع بر اساس نواحی شبکه
6. گرفتن خروجی­های سوم و چهارم برای شبکه با ساختار ضلع محور

همانطور که در ‏شکل (1) نشان داده شده است، شبکه­هایی با ساختار \*.SU2 دارای فرمتی به صورت آن شکل می­باشد. برای تعیین شکل هندسی المان­ها از فرمت VTK استفاده شده است. در این فرمت به هر شکل هندسی مرسوم در شبکه­های محاسباتی یک کد اختصاص داده می­شود. کدها و شکل هندسی آن­ها در ‏شکل (15) نشان داده شده است.



1. فرمت VTK

# پیاده‌سازی و زیربرنامه‌های مورد استفاده

در ادامه زیربرنامه­های بکار رفته برای پیاده­سازی توضیح داده می‏شود. برای مطالعه مستندات برخی از زیربرنامه­ها باید به مستندات آن زیربرنامه مراجعه شود.

1. گرفتن اطلاعات از کاربر

ابتدا با فراخوانی زیربرنامه Read\_2DMeshSU2 فایل ورودی با پسوند \*.SU2 خوانده شده و تمام اطلاعات موجود در این فایل از کاربر گرفته می­شود. سپس با استفاده از ساختار تعریف شده برای شبکه­های SU2، تمامی متغیرهای مورد نیاز برای تشکیل یک شبکه با ساختار سلول محور مقداردهی می­گردند.

1. به دست آوردن المان­های همسایه

با توجه به آنکه در شبکه­هایی با ساختار سلول محور نیاز به اطلاعات همسایه­های هر المان می­باشد و این اطلاعات در شبکه­ها با فرمت \*.SU2 وجود ندارد، لذا به دست آوردن شماره­ی المان­های همسایه هر یک از المان­ها امری غیرقابل اجتناب می­باشد. برای این منظور با فراخوانی زیربرنامه­ی FindNeib2D همسایه­ها به دست آورده خواهند شد.

1. گرفتن خروجی شبکه­ای با ساختار سلول محور و اطلاعات مرزهای شبکه

با استفاده از زیربرنامه‌ها­ی WritePlaneMesh\_cgid\_plt و WriteBoundCrv\_cgid\_plt و به کارگیری متغیرهای مقداردهی شده در قسمت اول و همسایه­های استخراج شده در قسمت دوم، به ترتیب شبکه­ای با ساختار سلول­محور در فایلی با عنوان PlaneMesh.plt و اطلاعات مرزهای شبکه در فایلی دیگر با عنوان BoundCrv.plt تولید می­شوند.

1. تبدیل ساختار سلول محور به ضلع محور

در این قسمت با فراخوانی زیربرنامه­ی Cell\_to\_Edge\_Hybrid2D، ساختار سلول محور به ساختار ضلع محور تبدیل می­یابد. در این زیربرنامه با استفاده از شماره­ی نقاط تشکیل­دهنده­ی المان­ها (متغیر Corn)، شماره­ی همسایه­های تشکیل­دهنده­ی هر المان (متغیر Neib) و شکل هندسی هر المان (منظور مثلثی یا مستطیلی بودن المان است که با متغیر CellType تعیین می­شود)، تعداد اضلاع تشکیل­دهنده­ی شبکه (NF) و اطلاعات اضلاع (IDS) شامل شماره­ی نقاط تشکیل­دهنده و المان­های اصلی (ME) و همسایه (NE) استخراج می­شود.

1. مرتب سازی اضلاع شبکه بر اساس نواحی موجود در شبکه

در این قسمت از برنامه با فراخوانی زیربرنامه­ی OrganiseIDS اضلاع موجود در شبکه بر اساس نواحی موجود در شبکه مرتب­ سازی می­شوند. این مرتب­سازی با استفاده از تعداد اضلاع مرزی (NBoundCrv)، تعداد اضلاع تشکیل­­دهنده­ی هر مرز (NFacCrv) و نقاط تشکیل­دهنده­ی اضلاع مرزی (BFacPt) انجام می­شود.

1. آماده سازی ساختار ضلع محور برای گرفتن خروجی­ها

قبل از فراخوانی زیربرنامه­ی WriteMesh\_gid به منظور گرفتن خروجی­ مطلوب ضلع محور، باید برخی از متغیرها آماده شده و برای ورودی این زیربرنامه استفاده گردند. ابتدا با توجه به قابلیت دو بعدی بودن، می­بایست متغیر Z صفر مقداردهی شود. سپس مشخصات مرزها در ماتریس­های جداگانه­ای ذخیره می­گردد. سپس تعداد ناحیه­های شبکه (NR) و اسم آن­ها (BCTitle) با استفاده از متغیرهای شبکه­ی سلول محور برای استخراج شبکه­ای با ساختار ضلع محور مقداردهی می­شوند.

1. چاپ شبکه بصورت ضلع محور در خروجی

با فراخوانی زیربرنامه WriteMesh\_gid به منظور تولید شبکه با ساختار ضلع محور فایلی درخروجی یا عنوان MeshOut.gid چاپ می‌شود.

1. Cell based [↑](#footnote-ref-1)
2. Edge based [↑](#footnote-ref-2)
3. Hybrid [↑](#footnote-ref-3)
4. CFD [↑](#footnote-ref-4)
5. Open Source [↑](#footnote-ref-5)
6. Stanford [↑](#footnote-ref-6)