



**زیربرنامه:**

ReadMsh

|  |  |  |
| --- | --- | --- |
| **توسعه دهندگان** | مرتضی نامور |  |
| میثم وکیلی | D:\Darsi\Karshenasi Arshad\Others\usc logo.gif |
| **تهیه کنندگان مستند** | میثم وکیلی | |
| **تاییدکنندگان** | مرتضی نامور | |
| **تاریخ تنظیم سند** | 15/08/1394 | |
| **شناسه سند** | **MC5F077F1** | |
| **زبان برنامه‌نویسی** | **Fortran 90/95** | |

1. وظایف

در این زیربرنامه اطلاعات مربوط به یک شبکه دوبعدی یا سه بعدی، که شامل کدهای مختلف برای تعریف شرایط هندسی و مرزی مساله می­باشد از یک فایل ورودی به نام MeshIn.msh خوانده می­شوند.

1. توضیحات و تئوری­ها

در این زیربرنامه امکان بازخوانی فایل خروجی نرم افزارهای تولید شبکه وجود دارد. این فایل در نرم افزارهای تجاری با فرمت \*.msh شناخته شده که شامل کدهای مختلف برای تعریف شرایط هندسی و مرزی مساله می­باشد. در این قسمت، کدهای مورد نیاز هر مساله مورد بررسی قرار گرفته و جزئیات آنها به همراه مثال توضیح داده خواهند شد. کلیه­ی اعداد استفاده شده در این فایل­ها در مبنای 16 می­باشند. دلیل اینکه اعداد در مبنای 16 هستند اینست که ورود و خروج فایل­ها برای خواندن و تحلیل توسط نرم افزارهای تحلیلگر از جمله فلوئنت سریعتر انجام گیرد.

هر دستور در فایل ورودی یک کد داشته که به آن شاخص می­گویند. این شاخص مشخص­کنندة این مطلب است که آن بخش از دستور بیان کنندة کدام ویژگی مساله می­باشد. ویژگی­های مساله مشخصاتی از داده­های اولیه از هندسه شکل هستند که باید در فایل ورودی تعریف شوند. این داده­های اولیه شامل موارد زیر هستند:

1. بُعد مساله (دو بعدی یا سه بعدی)
2. تعداد نقاط و مختصات آنها
3. تعداد صفحه­ها (یا تعداد اضلاع در دو بعد) و اطلاعات مربوط به هر صفحه شامل نقاط تشکیل­دهندة هر صفحه بعلاوة شماره دو سلول اطراف صفحه مورد نظر.
4. تعداد سلول­های تشکیل دهنده شبکه محاسباتی
5. نوع هر کدام از سلول­های تشکیل دهنده شبکه محاسباتی

|  |
| --- |
| D:\Notes\kasri fluent\reports\Presentation1.jpg |
| 1. شبکه مربعی ساده |

این موارد شامل دستورات اصلی هر مساله هستند که باید در برنامه نوشته شوند. برای هریک از این موارد، شاخص خاصی در نظر گرفته شده که در ادامه با ذکر یک مثال به توضیح شاخص­های لازم پرداخته می­شود.

همانطور که در ‏شکل (1) مشاهده می­شود این شکل مربعی دارای 16 نقطه، 24 صفحه و 9 سلول می­باشد. ضلع بالا و پایین مربع اشاره شده، دیوار در نظر گرفته شده و ضلع چپ ورودی سیال و ضلع سمت راست خروجی می­باشد. نوع سلول­ها نیز چنانچه در شکل دیده می­شود مربعی هستند.

در ادامه بخش­های مختلف فایل حاوی شبکه مطابق با مثال فوق توضیح داده می­شود.

* 1. توضیحات

قبل از اینکه اطلاعات مربوط به شکل در برنامه وارد شود، نیاز است تا توضیحاتی در برنامه داده شود. قسمت توضیح می­تواند هرجایی غیر از داخل دستور دیگر نوشته شود. چنانچه نیاز باشد قبل یا بعد از اینکه دستوری شروع می شود توضیحی داده شود، می­توان از این مورد استفاده کرد. شاخص این دستور عدد 0 بوده و به صورت زیر در فایل ورودی ظاهر می­شود:

(0 "Comment text")

|  |
| --- |
|  |

پیشنهاد می­شود قبل از شروع هر قسمت که ممکن است طولانی باشد، توضیحی در مورد آن نوشته شده تا مشخص شود آن دستور مربوط به چه قسمتی از مساله است. لازم بذکر است که نوشتن این دستور اختیاری است. حداقل یک فاصله باید بین عدد 0 و کوتیشن باشد. دیگر فاصله ها می­تواند کمتر یا بیشتر باشد. در مثال فوق یک توضیح در مورد کل برنامه داده شده است:

(0 "This is a Grid file which is created by FORTRAN: ")

این توضیح بیان می­کند که فایل حاوی شبکه توسط برنامه فرترن نوشته شده است.

* 1. بعد شبکه

در این مرحله، ابعاد مساله بیان می­شود که شاخص آن 2 می باشد. این دستور اختیاری است. یعنی می­تواند در فایل ورودی وجود داشته باشد یا نداشته باشد. این دستور به صورت زیر نوشته می­شود:

(2 ND)

که ND می تواند 2 یا 3 باشد. این دستور در مثال فوق بصورت زیر نوشته می شود که بیان کنندة دو بعدی بودن مساله است:

(0 "Dimensions:")

(2 2)

* 1. اطلاعات نقاط شبکه و مختصات آنها

بعد از مشخص شدن ابعاد مساله، نوبت به معرفی نقاط و مختصات آنها می­رسد. شاخص مربوط به این دستور عدد 10 بوده و در دو خط نوشته می­شود. نوشتن این دستور در کلیه­ی مسائل ضروری بوده و دارای ساختار زیر است:

(10 (zone-id first-index last-index type ND))

(10 (zone-id first-index last-index type ND))

خط اول یک معرفی از اطلاعات کلی مربوط به نقاط ارائه داده و بعد از کد 10 به جای zone-id عدد 0 که نشان­دهندة معرفی دستور است در داخل پرانتز نوشته می شود. این اطلاعات شامل تعداد کل نقاط و ابعاد مساله است. در خط دوم همان اطلاعات تکرار شده با این تفاوت که به جای zone-id عدد 1 نوشته شده و در انتها یک پرانتز باز می­شود. در هر دو خط در این دستور، first-index شماره­ی اولین نقطه­ و last-index شماره­ی آخرین نقطه­ی شکل مساله است. یادآوری می­شود این اعداد در مبنای 16 نوشته می‌شوند.

عبارت Type بیان­کنندة این مطلب است که آیا آن ناحیه از شبکه ناحیة فعال است (جامد یا سیال) یا غیر فعال (فقط سلول­های والد نتیجه شده از انطباق نقاطی معلق). ناحیه­های فعال با مقدار Type=1 و ناحیه­های غیر فعال با Type=32 مشخص می­شوند.

در انتهای خط دوم، مختصات مربوط به تمام نقاط تشکیل­دهندة شبکه آورده می­شوند. بلافاصله پس از دو خط بالا مختصات نقاط بصورت زیر نوشته ­شده و در آخر دو پرانتز باز شده باید بسته شوند.

x1 y1 z1

x2 y2 z2

.

.

.

))

قابل ذکر است دستوراتی که در فایل ورودی هستند، اصولا دوبار نوشته می­شوند. یک بار برای معرفی کلی آن دستور در مورد هندسه­ی مساله است. مثلا در مورد دستور گره­ ها، یکبار مشخص می­کند که شکل مساله دارای چند گره می­باشد. در این قسمت zone-id برابر صفر خواهد بود. در دفعه­ی دوم به جزئیات آن دستور در مورد هندسه پرداخته می­شود و zone-id بیشتر از صفر است. در کل، zone-id مشخص کننده­ی ناحیه­ای از مساله است که به طور دلخواه عدد گذاری می­شود. ناحیه­ی نقاط با عدد 1، ناحیه­ی مش­ها با عدد 2 و در ادامه مرزها به ترتیب عدد گذاری می­شوند. در انتها نیز ناحیه­ی سلول­ها عدد گذاری می­شود. این ترتیب اجباری نبوده و جای اعداد می­تواند عوض شود.

در مثال بالا، این دستور بصورت زیر نوشته می شود:

(10 (0 1 10 1 2))

(10 (1 1 10 1 2))

0.00000000e+00 0.00000000e+00

1.00000000e+00 0.00000000e+00

2.00000000e+00 0.00000000e+00

3.00000000e+00 0.00000000e+00

3.00000000e+00 1.00000000e+00

3.00000000e+00 2.00000000e+00

3.00000000e+00 3.00000000e+00

2.00000000e+00 3.00000000e+00

1.00000000e+00 3.00000000e+00

0.00000000e+00 3.00000000e+00

0.00000000e+00 2.00000000e+00

0.00000000e+00 1.00000000e+00

1.00000000e+00 1.00000000e+00

2.00000000e+00 1.00000000e+00

1.00000000e+00 2.00000000e+00

2.00000000e+00 2.00000000e+00

))

خط اول بیان می­کند که این مساله دارای 16 نقطه بوده که در مبنای 16 برابر با 10 است (در مبنای 16 اعداد 1 تا 9 خود همان اعداد می­شوند). در خط دوم، مختصات نقاط جزو ناحیة 1 در نظر گرفته شده و اطلاعات خط بالای آن تکرار شده اند. مختصات نقاط در همان مبنای دهدهی بوده و حداقل یک فاصله بین ستون x و y وجود دارد. بین شاخص 10 و پرانتز نیازی به فاصله نبوده و اختیاری است. همچنین بین اعداد داخل پرانتز بعد از شاخص نیز حداقل یک فاصله مورد نیاز است. اگر این فاصله بیشتر شود بلامانع می باشد.

معمولا first-index و last-index همان اعدادی هستند که در خط قبل از آن که به ترتیب شماره اولین و آخرین گره تشکیل­دهنده­ی هندسه­ی مساله بودند. همانطور که گفته شد Type نیز همیشه برابر با 1 است. گزینه­ی NDنیز مشخص­کننده­ی ابعاد مختصات نقاطی است که در صورتی که دوبعدی باشند، فقط ستون x و ستون y در فایل ورودی ارائه خواهند شد. چنانچه ابعاد مختصات سه بعدی باشند، مختصات z نیز اضافه شده و مختصات دارای سه ستون خواهد شد.

* 1. صفحه­های تشکیل دهنده شبکه

در این مرحله نوبت به صفحه­های موجود در هندسة مساله می­رسد. شاخص مربوط به این دستور عدد 13 بوده و مانند دستور قبل در دو خط نوشته می­شود. خط اول تعداد کل صفحه­ها و خط دوم تعداد صفحه­های مربوط به هر بخش به همراه اطلاعات صفحات آن ناحیه را بیان می­کند. این دستور نیز مانند نقاط ضروری بوده و قالب آن به صورت زیر می­باشد.

(13 (zone-id first-index last-index bc-type face-type))

در رابطه­ی فوق داریم:

zone-id: شماره­ی ناحیه­ی مورد نظر.

first-index: شماره اولین صفحه در آن ناحیه.

last-index: شماره آخرین صفحه در آن ناحیه.

bc-type: شماره­ی شرط مرزی مربوط به ناحیه­ی مورد نظر با توجه به ‏جدول (1)‏ .

face-type: شماره­ی نوع صفحه در آن قسمت با توجه به جدول (2).

لازم بذکر است که صفحات (اضلاع) تشکیل­دهندة شبکه می­توانند به چندین ناحیه تقسیم شوند. برای مثال مرز دیواره، مرز دوردست، غیر مرزی و غیره. در این بخش صفحات تشکیل­دهندة هر ناحیه در یک بخش جداگانه آورده می­شود.

انواع شرایط مرزی متداول موجود با توجه به ‏جدول (1)‏ مشخص شده که در قسمت bc-type استفاده می­شوند. برای نوع صفحه که گزینه­ی آخر این دستور است، از ‏جدول (2) استفاده می­شود:

|  |  |  |
| --- | --- | --- |
| 1. کدهای مربوط به گزینه­ی bc-type | | |
| نوع شرط مرزی | معادل انگلیسی | کد مربوطه |
| داخلی | Interior | 2 |
| دیوار | Wall | 3 |
| فشار ورودی، دریچه ورودی، مجرای مکش | Pressure-Inlet,Inlet-Vent, Intake-Fan | 4 |
| فشار خروجی، پروانه خروجی، دریچه خروجی | Pressure-Outlet,Exhaust-Fan,Outlet-Vent | 5 |
| تقارن | Symmetry | 7 |
| سایه­ی تکرار شونده | Periodic-Shadow | 8 |
| فشار دور دست | Pressure-Far-Field | 9 |
| سرعت ورودی | Velocity-Inlet | 10 |
| تکرار شونده | Periodic | 12 |
| پروانه، پرش متخلخل، رادیاتور | Fan, Porous-Jump, Radiator | 14 |
| جریان جرم ورودی | Mass-Flow-Inlet | 20 |
| واصل | Interface | 24 |
| والد (گره­ی معلق) | Parent (Hanging Node) | 31 |
| جریان خروجی | Outflow | 36 |
| محور | Axis | 37 |
|  |  |  |
| 1. کدهای مربوط به گزینه­ی face-type | | |
| نوع صفحه | توصیف | گره / صفحه |
| 0 | مخلوط\* | ----- |
| 2 | خطی | 2 |
| 3 | مثلثی | 3 |
| 4 | مربعی | 4 |
| 5 | چند گوشه | NN |

\* ترکیبی از انواع صفحه­ها

در ‏جدول (2) NN متناسب با نوع چند وجهی تغییر می­کند.

همانند دستورات قبل اگر zone-id صفر باشد، معرفی کلی از تعداد صفحه­های مساله دارد.

برای مثال فوق معرفی دستور 13 بصورت زیر بیان می شود:

در کل، 24 صفحه در این مساله وجود دارند که در مبنای 16 برابر با 18 است. در نتیجه معرفی این دستور بصورت زیر نمایش داده می­شود:

|  |
| --- |
| (13 (0 1 18 0 0)) |

هنگامیکه zone-id صفر نیست، هر صفحه با جزئیات اتصال و رابطه آن با صفحه­های مجاور را توضیح می­دهد. هر خطی از این دستور، یک صفحه را توصیف کرده و دارای ساختار زیر است:

|  |  |
| --- | --- |
|  | n0 n1 n2 c0 c1 |

که در این رابطه:

|  |  |
| --- | --- |
| نقاط یا رئوس هر صفحه | n\*= |
| سلول­های مجاور | c\*= |

این رابطه قالبی است برای یک شبکه سه بعدی با صفحه­های مثلثی شکل. تعداد نقاط به نوع صفحه بستگی دارد. نوشته شدن ترتیب شاخص نقاط مهم است و با قانون دست راست محاسبه می­شود. اگر انگشتان دست راست در جهت ترتیب شماره نقاط پیچانده شود، شصت در جهت c1 خواهد بود.

برای شبکه­های دو بعدی n2 حذف می­شود. c1 با ضرب خارجی بردارهای r و k بوجود می­آید که r برداری در امتداد n0 به n1 بوده و ابتدای بردار k در نقطه n0 و در جهت بیرون صفحه به سمت بیننده می­باشد. اگر انگشتان دست راست در جهت بردار r بوده و در جهت زاویه بین بردارهای r و k پیچانده شود، شصت در جهت r**×**k به سمت c1 خواهد بود.

اگر نوع ناحیه صفحه مخلوط باشد (face-type=0)، هر خط جزئیات این دستور با شماره تعداد نقاط تشکیل دهنده صفحه آغاز خواهد شد و دارای ساختار زیر است:

برای شبکه­های دو بعدی، n2 حذف می­شود. c1 با ضرب خارجی بردارهای در امتدادn0 به n1 بوده و ابتدای بردار در نقطه­ی n0 و در جهت بیرون صفحه به سمت بیننده می­باشد. اگر انگشتان دست راست در جهت بردار بوده و در جهت زاویه­ی بین بردارهای و پیچانده شوند، شصت در جهت به سمت c1 خواهد بود.

x n0 n1 … nf c0 c1

که در این رابطه x و nf بصورت زیر تعریف می­شوند:

= x تعداد نقاطی تشکیل­دهنده­ی صفحه.

= nf گره پایانی صفحه.

مقادیر سلول­ها، صفحات و نقاط مثبت هستند. اگر صفحه­ای تنها یک سلول در مجاورت خود داشته باشد، آنگاه c0 یا c1 صفر خواهد بود.

برای مثال ذکر شده، از روش دوم استفاده شده است. یعنی نوع ناحیة صفحه مخلوط در نظر گرفته شده و face-type = 0 است. در نتیجه فرمت نوشتاری اطلاعات مربوط به هر صفحه بصورت زیر نوشته می­شود:

x n0 n1 c0 c1

که x تعداد نقاط تشکیل دهندة هر صفحه است. چون در دو بعد هستیم کلیة صفحات از دو نقطه تشکیل شده اند. n0 و n1 دو نقطة رئوس صفحه بوده و c0 شماره سلول سمت چپ بردار n0n1 و c1 شماره سلول سمت راست آن می باشد.

همانطور که اشاره شد، صفحات به 4 گروه اصلی شامل، فضای داخلی[[1]](#footnote-1)، دیواره­ها (که دو ضلع بالا و پایین در ‏شکل (1) هستند)، سرعت ورودی (که ضلع سمت چپ ‏شکل (1) می باشد) و جریان خروجی (که ضلع سمت راست شکل (1) می باشد) می­شوند. برای وارد کردن کدهای مربوط به این شرایط مرزی می­توان از ‏جدول (1) استفاده کرد. همچنین نوع صفحات که در ‏جدول (2) آمده­اند، مخلوط انتخاب شده و در نتیجه تعداد نقاط تشکیل دهنده­ی هر صفحه (که در اینجا همه­ی صفحات از دو نقطه تشکیل شده­اند)، بطور جداگانه برای آنها در هر خط نوشته شده است. بعنوان مثال، برای صفحه­های مربوط به فضای داخلی هندسه­ی مساله، دستورات بصورت زیر نوشته می­شوند. این صفحات بعنوان ناحیة 2 نامگذاری شده­اند.

|  |
| --- |
| (13 (2 1 c 2 0)(  2 c d 2 1  2 d e 5 4  2 e 5 8 7  2 6 f 8 9  2 f 10 5 6  2 10 b 2 3  2 2 d 1 4  2 d 10 2 5  2 10 9 3 6  2 3 e 4 7  2 e f 5 8  2 f 8 6 9  )) |

خط اول این دستور بیان می­کند که صفحات 1 تا c در فضای داخلی قرار دارند. عدد c در مبنای 16 معادل عدد 12 بوده که در ‏شکل (1) بصورت f12 نمایش داده شده است. در خط دوم اطلاعات مربوط به صفحه­ی اول داده شده است. عدد اول نشان دهنده­ی تعداد نقاط تشکیل­دهنده­ی صفحه که در این مساله دو نقطه بوده، عدد دوم شماره­ نقطه­ی اول و عدد سوم شماره نقطه­ی دوم تشکیل­دهنده­ی صفحه­ی اول می­باشند. عدد چهارم شماره سلول سمت چپ این صفحه و عدد پنجم شماره سلول سمت راست آن هستند. به همین ترتیب خطوط بعدی، اطلاعات مربوط به صفحه­ی دوم تا دوازدهم ناحیه­ی داخلی شکل را بیان می­کنند.

دستورات مربوط به گروه صفحاتِ دیواره­ها، سرعت ورودی و جریان خروجی که به ترتیب در ناحیه­های 3، 4 و 5 نامگذاری شده­اند، بصورت زیر نوشته می­شوند.

|  |
| --- |
| (13 (3 d 12 3 0))  2 1 2 1 0  2 2 3 4 0  2 3 4 7 0  2 7 8 9 0  2 8 9 6 0  2 9 a 3 0  ))  (13 (4 13 15 a 0))  2 a b 3 0  2 b c 2 0  2 c 1 1 0  ))  (13 (5 16 18 24 0)(  2 4 5 7 0  2 5 6 8 0  2 6 7 9 0  )) |

* 1. نوع و تعداد سلول­ها

در مرحلة آخر باید نوع و تعداد سلول­ها مشخص شوند. شاخص مربوط به این دستور عدد 12 بوده و مانند دیگر دستورات در دو خط نوشته می­شود. همچنین این دستور جزو دستورات ضروری و فرمت آن بصورت زیر می­باشد:

(12 (zone-id first-index last-index type element-type))

مجددا، اگر zone-id صفر باشد، فقط مشخص­کننده­ی تعداد سلول­های موجود در مساله است. اگر last-index برابر با صفر باشد، هیچ سلولی در مساله وجود ندارد. در بخش معرفی این دستور، type مقدار صفر داشته و element-type حذف می­شود. برای مثال ذکر شده داریم:

|  |
| --- |
| (12 (0 1 9 0)) |

دستور فوق بیان می­کند که مساله دارای 9 سلول است که این مورد باید در مبنای 16 نوشته شود. بخش معرفی هر دستور نوشته می شود و از ملزومات فایل ورودی می­باشد. گزینه­ی element-type مشخص­کننده­ی نوع سلول­ها بوده که طبق ‏جدول (3) تعیین می­شود:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. کدهای مربوط به گزینه­ی element-type | | | | |
| نوع المان | توصیف | معادل انگلیسی | گره / سلول | صفحه / سلول |
| 0 | مخلوط | Mixed | ----- | ----- |
| 1 | مثلثی | Triangular | 3 | 3 |
| 2 | چهار ضلعی | Tetrahedral | 4 | 4 |
| 3 | مربعی | Quadrilateral | 4 | 4 |
| 4 | شش سطحی | Hexahedral | 8 | 6 |
| 5 | هرمی | Pyramid | 5 | 5 |
| 6 | گوه­ای | Wedge | 6 | 5 |
| 7 | چند وجهی | Polyhedral | NN | NF |

در این جدول NN و NF بسته به نوع چند وجهی متفاوت می­باشند.

بخش دوم این دستور، اگر از نوع سلول مخلوط استفاده نشود، مانند دستورات قبل دارای جزئیات نبوده ولی باید نوشته شود. در خط بعد نوع سلول­ها که مربعی بوده و طبق ‏جدول (3) کد 3 به آن تعلق دارد، مشخص شده است. ناحیه­ی سلول­ها با عدد 6 نامگذاری شده­اند.

(12 (6 1 9 1 3))

همانگونه که در دستورات پیشین اشاره شد، فقط بین اعداد داخل پرانتز باید حداقل یک فاصله وجود داشته باشد و وجود فاصله بین عدد شاخص و پرانتزهای قبل و بعد آن و یا پرانتزهای انتهایی اختیاری است و اگر فاصله­ای در این مکان­ها باشد، مشکلی در برنامه ایجاد نمی­کند. اگر نوع سلول­ها از نوع مخلوط باشد (element-type=0) آنگاه عبارت Type برابر با صفر خواهد شد. در این مواقع، بعد از خط دوم که نوع سلول­ها مشخص می شود، نوع تمام سلول­ها نیز بیان می شود. به عنوان مثال:

|  |
| --- |
| (12 (9 1 3d 0 0) )  1 1 1 3 3 1 1 3 1  .  .  .  )) |

عبارت فوق بیان می­کند که شبکة مساله دارای 3d سلول (در مبنای 16 معادل 61 سلول) می­باشد و در ناحیة نهم قرار دارد. در این شبکه سه سلول اول مثلثی، 2 سلول بعد مربعی، مجددا دو سلول بعد مثلثی و به همین ترتیب نوع سلول­ها به ترتیب مشخص می­شود. بعبارتی بعنوان ورودی مساله باید بدانیم که سلول شمارة چندم چه نوع سلولی است. تعداد ستون­های این درخت دارای تعداد دلخواه بوده و اگر چند سلول آخر، ستون­ها را پر نکردند، مشکلی ایجاد نمی­شود. بین هر دو سلول که نوع آنها مشخص شده است، باید حداقل یک فاصله وجود داشته باشد.

چنانچه دستورات مربوط به مثال فوق پشت سر هم در فایل متنی نوشته شده و با پسوند \*.msh ذخیره شوند، می­توان آنرا در نرم افزار فلوئنت برای حل آماده­سازی کرد.

1. بخش­های زیربرنامه

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

1. خواندن فایل ورودی

در این خط فایل ورودی زیربرنامه با فرمت msh باز می­شود.

1. مقداردهی اولیه به برخی متغیرها

بدون توضیح.

1. **خواندن خط به خط فایل** 'MeshIn.msh'

این حلقه، حلقه اصلی و کلی زیربرنامه را تشکیل می دهد. با استفاده از این حلقه، فایل ورودی خط به خط خوانده می شود و برای تحلیل، بخش مربوطه اجرا می­شود. اگر خطی مربوط به شاخص دستور باشد نوع شاخص تشخیص داده می شود، اگر خط در برگیرنده مقادیر مختصات نقاط یا اطلاعات اضلاع شبکه باشد این مقادیر در متغیرهای مربوطه قرار می گیرند. متغیر RowNum که شمارنده حلقه است دارای مقدار اولیه 100 می باشد. این مقدار متناسب با تعداد شاخصهای دستور موجود در فایل ورودی است. همچنین در ابتدای حلقه، برخی از متغیرهایی که در هر بار اجرای حلقه باید به مقدار اولیه خود برگردند، مقدار دهی اولیه می شوند.

1. خواندن یک سطر از فایل ورودی

در اینجا یک سطر به طول 200 کاراکتر از فایل ورودی خوانده شده و در متغیر رشته ای TempStr برای تحلیل های بعدی قرار می دهد. گزینه IOSTAT در دستور خواندن فایل، برای مشخص کردن وضعیت انتقال داده با استفاده از یک متغیر (که در اینجا با نام IO مشخص شده است) به کار می رود. اگر مقدار IO مثبت باشد یک خطا رخ داده است. اگر مقدار IO منفی باشد به معنای پایان رکورد یا پایان فایل است. اگر صفر باشد یعنی هیچکدام از شرایط قبلی رخ نداده است. برای جلوگیری از وقوع خطا و به خصوص رسیدن به پایان رکورد یا فایل، بهتر است وضعیت اشاره گر فایل (متغیرIO ) در یک متغیر ذخیره شود. در خط بعد نیز با دستور Adjustl فضاهای خالی ابتدای رشته TempStr حذف می شوند تا بررسی رشته راحت تر و درست تر صورت گیرد.

1. تشخیص سطر شاخص دستور

این بخش به طور کلی برای تشخیص و شروع خواندن و تحلیل شاخص دستورات به کار می رود. در دستور If بررسی می شود که آیا کاراکتر اول سطری که در بخش 4 خوانده شد و در متغیر TempStr قرار گرفته، برابر با پرانتز باز است یا خیر. اگر برابر باشد به معنای شروع یک سطر حاوی شاخص دستور است. در غیر این صورت، دستور Else اجرا شده، ابتدا یک واحد از شمارنده حلقه اصلی زیربرنامه (RowNum) کاسته شده و سپس با دستور Cycle اجرا به ابتدای حلقه اصلی (بخش 3) باز می گردد. عبارت String(i:j) به طور کلی یعنی از کاراکتر سمت چپ داخل پرانتز (i) تا کاراکتر سمت راست داخل پرانتز (j)؛ بنابراین در عبارت TempStr(1:1) چون هر دو مقدار 1 است، یعنی کاراکتر اول متغیر TempStr.

1. خواندن شاخص دستورات

بعد از اینکه در بخش قبل تشخیص داده شد که سطر خوانده شده مربوط به شاخص یک دستور است، در این حلقه تا زمانیکه در رشته TempStr به کاراکتر Space نرسیده ایم، یک کاراکتر یک کاراکتر رشته را می خوانیم و توسط دستور خط اول این حلقه کاراکترها را به متغیر IndexChars اضافه می کنیم. عبارت IndexChars(:J)//TempStr(I:I) به این معناست که کاراکتر جاری رشته TempStr را به ادامه رشته IndexChars، از کاراکتر J به بعد اضافه کن. متغیر J دارای مقدار اولیه 1 است و با هربار اجرای حلقه یکی به مقدار آن اضافه می­شود.

1. قرار دادن متغیر رشته ای شاخص دستور در متغیر عددی

با توجه به اینکه قرار است در ادامه شاخص دستور خوانده شده توسط دستور Select مورد بررسی قرار گیرد، و چون در دستورات Case از عملگر مقایسه و تساوی استفاده می شود، بهتر است برای انجام این عملیات شاخص دستور خوانده شده در یک متغیر عددی که در اینجا IndexCode نام دارد استفاده شود. عملکرد دستور Read که در این بخش استفاده شده، به این صورت است که مقدار متغیر رشته ای IndexChars را در متغیر عددی IndexCode قرار می دهد.

1. انتخاب شاخص دستور

این دستور Select متغیر IndexCode که مقدار شاخص دستور در آن قرار دارد را مورد بررسی قرار داده و متناسب با مقدار آن، دستورات Case موردنیاز را اجرا می کند.

1. بررسی شاخص دستور 0 (شاخص مربوط به کامنت­ها)

بطور کلی دستورات داخل Case(0) برای شاخص دستور 0 که مربوط به کامنتهای دستورات است به کار می روند. اما از بین کامنتهایی که قبل از شاخص دستورات مختلف می آیند، تنها کامنتهای مربوط به شاخص دستور 13 برای ما اهمیت دارند. زیرا زمانی که از یک نوع منحنی مرزی چندین مورد با نام های مختلف وجود داشته باشد، این کامنتها مشخص کننده تفاوت هر مورد با موردهای دیگر است. ضمن اینکه در نهایت برای هر نوع منحنی مرزی، کامنت مربوط به آن نیز در فایل خروجی txt برنامه اصلی نوشته می شود.

1. بررسی کامنت­های مربوط به شاخص دستور 13

یا توجه به اینکه در فایل msh، کامنتهای قبل از شاخص دستور 13، برای صفحات داخلی با عبارت "Interior faces of zone" و برای سایر صفحات با عبارت "Faces of zone" شروع می شوند، بنابراین در دستور If چک می شود که آیا رشته TempStr که در بخش 4 خوانده شد، حاوی این عبارات می باشد یا خیر. دستور Index(TempStr,"Faces of zone")>0 اگر عبارت Faces of zone در رشته TempStr وجود داشته باشد، محل شروع آن را بر می گرداند، که مسلم است این مقدار در صورت وجود بزرگتر از صفر است.

1. خواندن عبارت کامنت مربوط به شاخص 13

در حلقه ابتدایی این بخش، شمارنده I از 3 شروع می شود. به این دلیل که دو کاراکتر اول سطر مربوط به شاخص دستور توضیحات برابر با عبارت (0 می باشد که قبلا خوانده و تشخیص داده شده است. حلقه تا طول رشته TempStr که فاصله های خالی از انتهای آن حذف شده اند ادامه پیدا می کند. در خط بعد در دستور If، بررسی می شود که اگر کاراکتر جاری رشته TempStr مخالف پرانتز بسته است (که نشان دهنده رسیدن به انتهای دستور می باشد) و مخالف دابل کوتیشن نیز می باشد (زیرا دابل کوتیشن به معنای شروع یا پایان عبارت کامنت است و نباید ذخیره شود)، کاراکتر جاری در متغیر CommentStr ذخیره می شود. روش کار خط سوم، مشابه توضیحاتی است که در بخش 6 داده شد. در انتهای حلقه مقدار J را یک واحد افزایش می دهیم تا کاراکتر بعدی خوانده شود.

1. قرار دادن عبارت کامنت خوانده شده در متغیر مربوطه

بعد از اینکه عبارت کامنت مربوط به شاخص دستور 13 در بخش قبل خوانده شد، این عبارت باید در یک متغیر که در اینجا آرایه BCTitle است ذخیره شود. از متغیر BCTitleCount نیز به عنوان شمارنده این آرایه استفاده شده است که هربار یک واحد به آن اضافه می شود.

1. بررسی شاخص دستور 2 (شاخص مربوط به بعد شبکه)

بطور کلی دستورات داخل Case(2) برای شاخص دستور 2 که مربوط به بعد مسئله است به کار می روند.

1. خواندن بعد شبکه

روند کار این بخش در بخش های قبل توضیح داده شده. حلقه تا زمانیکه کاراکتر جاری رشته TempStr به پرانتز بسته نرسیده است ادامه می یابد. همانطور که قبلا نیز گفته شد، رسیدن به پرانتز بسته به معنای رسیدن به انتهای سطر یا به عبارتی رسیدن به انتهای دستور جاری است. در دستور If بررسی می شود که آیا کاراکتر جاری رشته TempStr مخالف Space است یا خیر. اگر مخالف Space باشد، آن مقدار که در واقع یا عدد 2 یا 3 است که نشان دهنده بعد شبکه می باشد خوانده شده و در متغیر رشته ای TempChars قرار می گیرد. اما چون قرار است بعد شبکه به صورت یک متغیر Integer بعدا مورد استفاده قرار گیرد، با استفاده از دستور Read(TempChars,'(I1)') MeshDim، مقدار بعد شبکه در متغیر عددی MeshDim قرار خواهد گرفت. در انتهای حلقه نیز یک واحد به شمارنده حلقه اضافه می شود.

1. بررسی شاخص دستور 10 (شاخص مربوط به نقاط شبکه)

بطور کلی دستورات داخل Case(10) برای شاخص دستور 10 که مربوط به نقاط شبکه است به کار می روند. ابتدا سطر مربوط به شاخص این دستور مورد بررسی قرار گرفته و اندیس ابتدایی و انتهایی نقاط بدست می­آیند. سپس به تعداد تفاضل اندیسهای انتهایی و ابتدایی و با توجه به دو بعدی یا سه بعدی بودن شبکه، مختصات نقاط شبکه خوانده شده و در آرایه های مربوطه قرار می گیرند.

1. خواندن سطر شاخص دستور 10

همانگونه که قبلا توضیح داده شد، هر شاخص دستور در 2 خط بیان می شود. خط اول یک معرفی کلی از دستور است که zone-id آن برابر صفر است و در بخش 4 خوانده شد. در اینجا خط دوم شاخص دستور 10 خوانده می شود که تفاوت آن با خط اول تنها در این است که zone-id آن بزرگتر از صفر است. این مقدار برای ادامه بررسی ها مجددا در متغیر رشته ای TempStr قرار داده می شود.

1. حلقه خواندن و بررسی شاخص دستور 10

عملکرد حلقه این بخش همانند حلقه به کار رفته در بخش 14 است.

1. خواندن و بررسی اجزای شاخص دستور 10

همانگونه که قبلا توضیح داده شد، بعد از خط دوم مربوط به شاخص دستور 10، در همان خط یا خط بعد یک پرانتز باز شده و در خط بعد از پرانتز باز مختصات نقاط آورده می شود. بنابراین در شرط If بررسی می شود که آیا کاراکتر جاری رشته TempStr مخالف پرانتز بازمی باشد (که به معنای رسیدن به شروع سطرهای حاوی مختصات نقاط است)، و هم چنین مخالف Space نیز هست. اگر مخالف هر دو مورد باشد، یک کاراکتر یک کاراکتر از رشته TempStr خوانده شده و در متغیر رشته ای TempChars قرار می گیرد، مطابق توضیحات بخش 6. در خط بعد هم یک واحد به متغیر L اضافه می شود تا شمارنده رشته TempChars به کاراکتر بعدی اشاره کند.

در صورتیکه شروط موجود در دستور If برقرار نباشد، شروط دستور Else If بررسی می شود. اگر کاراکتر جاری رشته TempStr برابر Space باشد و طول رشته TempChars که داخل دستور If ساخته شده بزرگتر از صفر باشد دستورات داخل Else If اجرا می شوند. Len\_Trim(TempChars) طول رشته TempChars که فاصله های خالی از انتهای آن حذف شده اند را بر می گرداند. سپس خط Read(TempChars,'(Z20)') NodeIndex(K) اجرا می شود. این خط وظیفه قرار دادن اجزای شاخص دستور 10 در آرایه NodeIndex را برعهده دارد. یعنی به ترتیب zone-id، first-index، last-index، type و ND در آرایه ذخیره می شوند. هر قسمت با فرمت دسیمال در آرایه ذخیره می شود. سپس به شمارنده آرایه NodeIndex یعنی K، یک واحد اضافه شده و متغیر L نیز مقدار اولیه خود را می گیرد.

در انتهای حلقه مقدار متغیر I یک واحد اضافه شده که باعث می شود در چرخه بعدی حلقه While اشاره گر رشته TempStr یک کاراکتر به جلو برود و کاراکتر بعدی را بخواند.

1. بررسی شروع خواندن مختصات نقاط شبکه

زمانی که شرط ادامه حلقه بخش 17 نقض شود، یعنی کاراکتر جاری رشته TempStr برابر با پرانتز بسته باشد، شرط دستور If بررسی می شود. همانطور که در بخش 18 توضیح داده شده، برای شروع مختصات نقاط شبکه، یا پرانتز باز در انتهای خط دوم شاخص دستور 10 آورده می شود، یا در خط بعدی. در اینجا، در شرط If بررسی می شود که آیا کاراکتر بعد از کاراکتر جاری در رشته TempStr پرانتز باز است یا خیر. اگر پرانتز باز نباشد، پس پرانتز باز در خط بعدی قرار دارد. بنابراین با دستور Read خط بعدی که تنها یک کاراکتر دارد و آن هم پرانتز باز است خوانده می شود.

1. خواندن مختصات نقاط شبکه

در این بخش عملیات خواندن مختصات شبکه صورت می گیرد. ابتدا در یک شرط If بررسی می شود که اگر مقدار MeshDim، یعنی بعد شبکه، که در بخش 14 خوانده شده برابر 2 باشد، سطر به سطر مختصات نقاط خوانده شده و در آرایه های X و Y با اندیس Count که همان شمارنده حلقه است قرار می گیرد. برای این کار یک حلقه Do از NodeIndex(2) تا NodeIndex(3) اجرا خواهد شد. این مقادیر به ترتیب برابر با اندیس شروع و پایان نقاط شبکه هستند که در بخش 18 خوانده شدند. اگر شبکه سه بعدی باشد، دستورات داخل Else If اجرا می شوند و مختصات بعد سوم نیز در آرایه Z قرار خواهد گرفت.

1. محاسبه تعداد نقاط شبکه

در این بخش با تفاضل درایه های سوم و دوم آرایه NodeIndex بعلاوه یک، تعداد نقاط شبکه بدست آمده و در متغیر NP یا همان Number of Existing Points قرار می گیرد. دلیل افزودن عدد یک این است که مثلا اگر دو نقطه در شبکه داشته باشیم، تفاضل درایه های دوم و سوم 2-1=1 می شود. بنابراین باید یک واحد به این تفاضل اضافه کنیم تا تعداد نقاط به درستی بدست آید.

1. بررسی شاخص دستور 12 (شاخص مربوط به سلول­های شبکه)

بطور کلی دستورات داخل Case(12) برای شاخص دستور 12 که مربوط به اطلاعات سلول های شبکه است به کار می روند. همانگونه که قبلا توضیح داده شد، در صورتیکه نوع سلولهای شبکه مخلوط نباشند، دستور 12 فقط در دو خط بیان می شود و دارای جزئیات دیگری نیست. اما اگر نوع سلولها مخلوط باشند، در سطرهای بعدی نوع هر سلول نیز باید مشخص شود. بنابراین در این شاخص دستور تنها به تعداد سلولهای شبکه، و در صورتیکه نوع سلولها مخلوط باشند به نوع هر سلول نیاز داریم.

1. بررسی متغیر CellFlag و مقداردهی به متغیرهای مورد نیاز

مقدار اولیه متغیر CellFlag برابر صفر تعریف شده است. بنابراین زمانی که برای اولین بار شاخص دستور 12 خوانده می شود، دستورات داخل این If اجرا نمی شود. هدف از به کار بردن شرط If و متغیر CellFlag این است که زمانی که برای اولین بار وارد شاخص دستور 12 می شویم، اطلاعات مربوط به تعداد سلول ها خوانده شده و در صورتیکه نوع سلول ها مخلوط باشند (پارامتر پنجم یعنی element-type برابر صفر باشد) نوع هر سلول نیز خوانده شده و در یک آرایه بزرگ قرار می گیرد. بنابراین تمام اطلاعاتی که ما در مورد سلول های شبکه نیاز داریم در همین بار اول خوانده می شوند. اما در برخی از فایل های ورودی msh ممکن است چندین بار شاخص دستور 12 تکرار شده باشد که هر کدام مربوط به بخشی از سلول های شبکه باشند. ما دیگر نیاز به خواندن و بررسی این شاخص ها نداریم. بنابراین زمانی که برای بار دوم شاخص 12 خوانده می شود، ابتدا توسط این شرط If چک می شود که آیا متغیر CellFlag برابر یک است یا خیر. یک بودن این متغیر به معنای این است که شاخص دستور 12 قبلا خوانده شده و اطلاعات مورد نیاز استخراج شده است. پس دیگر نیاز به اجرای بقیه دستورات Case(12) نیست و بعد از کم کردن یک واحد از متغیر RowNum، دستور Cycle اجرا شده و به ابتدای حلقه اصلی While بر می گردیم.

1. خواندن خط دوم شاخص دستور 12

همانطور که اشاره شد، هر شاخص دستور دارای دو خط است. خط اول مربوط به معرفی دستور است که دارای zone-id صفر است. در خط بعدی نوع و تعداد سلول ها مشخص شده و zone-id بزرگتر از صفر است. بنابراین با توجه به اینکه تمام اطلاعات خط اول در خط دوم موجود است، ما از خط دوم برای بررسی دستور و استخراج اطلاعات مورد نیاز استفاده می کنیم.

1. خواندن و بررسی اجزای شاخص دستور 12

شاخص دستور 12 نیز مانند سایر دستورات در حالت نرمال داری 5 قسمت است. بنابراین برای خواندن این 5 قسمت، تا زمانیکه شمارنده حلقه While یعنی متغیر K (که مقدار اولیه آن 1 است) کوچکتر یا مساوی 5 است، حلقه اجرا می شود، یعنی 5 بار. در عبارت شرطی دستور If بررسی می شود که اگر کاراکتر جاری رشته TempStr مخالف پرانتز باز و بسته و جای خالی است، این کاراکتر به ادامه رشته TempChars که در ابتدا تهی است اضافه شود. روش کار این دستورات به طور دقیق در بخش 6 توضیح داده شد. در صورتیکه کاراکتر جاری رشته TempStr برابر با Space شد، یعنی یکی از بخش های داخل شاخص دستور 12 خوانده شده است. پس Len\_Trim(TempChars) نیز بزرگتر از صفر بوده و دستورات داخل Else If اجرا خواهند شد. سپس توسط دستور Read(TempChars,'(Z20)') CellIndex(K)، مقدار رشته TempChars که برابر با یکی از بخش های دستور 12 است با فرمت دسیمال در درایه K-ام آرایه CellIndex قرار می گیرد. این آرایه دارای 5 عنصر است که در هر عنصر مقدار متناظر با شاخص دستور 12 قرار می گیرد. سپس در خطهای بعد مقدار شمارنده حلقه While یعنی K یک واحد اضافه شده و مقدار متغیر L به حالت اول باز می گردد. در انتهای حلقه نیز شمارنده رشته TempStr که با متغیر I نشان داده شده یک واحد به جلو می رود.

1. محاسبه تعداد سلول های شبکه

در این بخش با تفاضل درایه های سوم و دوم آرایه CellIndex بعلاوه یک، تعداد سلول های شبکه بدست آمده و در متغیر NC یا همان Number of Existing Cells قرار می گیرد.

1. مقدار دهی به متغیر CellFlag

در این بخش مقدار متغیر CellFlag برابر با یک می شود.کاربرد این متغیر در بخش 23 توضیح داده شده است.

1. بررسی مخلوط بودن یا نبودن سلول های شبکه

به طور کلی در این بخش بررسی می شود که آیا سلول های شبکه مخلوط هستند یا خیر. همانگونه که قبلا توضیح داده شد، برای اینکه مشخص شود نوع سلولهای شبکه مخلوط هستند باید مقدار گزینه element-type، یعنی پنجمین بخش شاخص دستور 12، برابر با صفر باشد. پس باید مقدار CellIndex(5) برابر با صفر باشد. اگر اینطور باشد، همانطور که در بخش 23 توضیح داده شد، نوع هر یک از سلولها در خطوط بعد از شاخص دستور 12 آورده می شود که باید خوانده شوند. در این حالت نوع هر سلول در یک درایه از آرایه ای بزرگ به نام CellType قرار می گیرد. حتما برای آشنایی با نحوه ثبت نوع سلول ها در فایل ورودی msh به توضیحات ارائه شده در ابتدای این سند رجوع کنید.

1. خواندن سطر اول نوع سلول های شبکه (در حالتی که نوع سلولها مخلوط باشند)

قبل از اینکه وارد حلقه خواندن و بررسی سطرهای مربوط به نوع سلول ها در حالت مخلوط شویم، ابتدا باید سطر اول سلول ها را بخوانیم تا بتوانیم در شرط حلقه بخش 30، مقدار هر سطر را بررسی کنیم. بنابراین ابتدا یک سطر را می خوانیم و توسط دستور Adjustl(TempStr) در صورت وجود فاصله های خالی در ابتدای رشته TempStr، آنها را حذف می کنیم.

1. حلقه خواندن نوع سلول های شبکه (در حالتی که نوع سلولها مخلوط باشند)

این حلقه وظیفه خواندن و نوشتن نوع سلول ها در آرایه CellType را بر عهده دارد. در این حلقه نوع سلول ها سطر به سطر خوانده شده و تا زمانی که به انتهای سلول ها نرسیده ایم، یعنی تا زمانی که رشته TempStr مخالف ')' یا '))' است، عملیات خواندن سطرها ادامه می یابد. دلیل این امر آن است که انتهای سلول ها ممکن است با دو سطر جداگانه که در هر سطر یک پرانتز بسته وجود دارد مشخص شود، یا اینکه هر دو پرانتز بسته در یک سطر قرار داشته باشند.

1. خواندن و بررسی نوع سلول های شبکه (در حالتی که نوع سلولها مخلوط باشند)

برای اینکه بتوانیم نوع سلول های شبکه را در حالت مخلوط تعیین کنیم، باید تک تک سطرها را بخوانیم. برای این کار باید طول هر سطر را محاسبه کنیم. زیرا ممکن است طول سطرها متفاوت باشند. بنابراین ابتدا طول هر سطر خوانده شده که در رشته TempStr قرار دارد را با دستور Len\_Trim(TempStr) محاسبه کرده و در متغیر K قرار می دهیم. همانطور که قبلا توضیح داده شد، هر سطر دارای تعداد متفاوتی از اعداد مختلف است که نشان دهنده نوع سلول ها می باشد. سپس در یک حلقه به طول K، هر یک از سلولها خوانده شده و در یکی از درایه های آرایه CellType قرار می گیرد. برای اینکه هر بار که سلولها در آرایه CellType قرار می گیرند، در ادامه عناصر قبلی نوشته شوند، از یک شمارنده به نام CellCount استفاده می شود که تنها مخصوص همین آرایه است. نحوه عملکرد دستورات داخل حلقه در بخش های قبل توضیح داده شده است.

1. خواندن سطرهای دوم به بعد از نوع سلول های شبکه (در حالتی که نوع سلولها مخلوط باشند)

دستورات این بخش دقیقا مانند بخش 29 هستند با این تفاوت که برای سطرهای دوم تا آخر به کار می روند.

1. نوشتن نوع سلول های شبکه در آرایه CellType (در حالتی که نوع سلولها مخلوط نباشند)

ما در هر صورت، چه سلولهای شبکه مخلوط باشند و چه نباشند، باید نوع آنها را در آرایه CellType ذخیره کنیم. زیرا این آرایه یکی از پارامترهای خروجی زیربرنامه است و بعدا در زیربرنامه چاپ کننده اطلاعات شبکه مورد استفاده قرار خواهد گرفت. بنابراین در این بخش، در صورتیکه که شرط دستور If بخش 28 ارضا نشود و نوع سلولها مخلوط نباشند، اجرا خواهد شد. در این بخش، نوع سلولهای شبکه که در درایه پنجم آرایه CellIndex قرار دارد و برای همه سلولهای شبکه یکسان است، در تمام درایه های متغیر CellType قرار می گیرد.

1. بررسی شاخص دستور 13 (شاخص مربوط به صفحات شبکه)

بطور کلی دستورات داخل Case(13) برای شاخص دستور 13 که مربوط به اطلاعات صفحه های شبکه است به کار می روند. در این بخش ابتدا جزئیات شاخص دستور 13 خوانده و بررسی شده، سپس شماره نقاط تشکیل دهنده هر صفحه به همراه سلولهای مجاور آن خوانده شده و در یک آرایه بزرگ قرار می گیرند.

1. حلقه خواندن و بررسی شاخص دستور 13

اندیس شروع این حلقه برابر با متغیر I است. مقدار I برابر با آخرین مقداری است که در بخش خواندن شاخص دستورات (بخش 6) در آن قرار گرفته است. بنابراین با توجه به اینکه شاخص دستورات نهایتا دو رقمی هستند و یک پرانتز باز نیز در ابتدای سطر مربوط به شاخص دستورات قرار دارد، مقدار I حداکثر برابر 4 است (رجوع شود به بخش 6). مقدار اندیس پایانی حلقه نیز برابر با طول رشته TempStr است که فاصله های خالی از انتهای آن حذف شده اند.

1. خواندن و بررسی اجزای شاخص دستور 13

جزئیات این بخش دقیقا مانند بخش 25 است. با این تفاوت که آرایه ای که در این بخش برای ذخیره اجزای شاخص دستور 13 استفاده می شود FaceIndex نام دارد.

1. محاسبه تعداد کل صفحات (اضلاع) شبکه

در این بخش ابتدا بررسی می شود که آیا اولین جزء شاخص دستور 13، یعنی zone-id، برابر صفر است یا خیر (همانطور که قبلا اشاره شد صفر بودن zone-id به این معنا است که این خط، خط معرفی دستور است). در این صورت با تفاضل درایه های سوم و دوم آرایه FaceIndex بعلاوه یک، تعداد اضلاع یا همان صفحات شبکه بدست آمده و در متغیر NF یا همان Number of Existing Faces قرار می گیرد. دلیل اینکه از خط معرفی دستور برای بدست آوردن تعداد کل اضلاع شبکه استفاده می شود این است که تنها در این شاخص دستور 13 است که تفاضل اندیس های ابتدایی و انتهایی تعداد کل اضلاع شبکه را می دهد. وگرنه در سایر شاخص های دستور 13 که zone-id آنها بیشتر از صفر است، تفاضل اندیس های ابتدایی و انتهایی، تعداد صفحات مربوط به آن ناحیه از شبکه را باز می گرداند.

1. محاسبه تعداد اضلاع متعلق به هر نوع منحنی

در صورتیکه دستور If بخش قبل اجرا نشود، یعنی برای شاخص دستورهایی با zone-id بزرگتر از صفر، دستورات داخل Else این بخش اجرا خواهند شد. ابتدا به مقدار متغیر RegionNum که دارای مقدار اولیه صفر است یک واحد اضافه می شود. این متغیر برای شمارش تعداد ناحیه های شبکه به کار می رود. در خط دوم تعداد اضلاع متعلق به منحنی ای که در حال بررسی است محاسبه شده و در درایه متناظر متغیر NFR یا **N**umber of **F**aces Belong to each **R**egions قرار می گیرد. در خط سوم نیز نوع شرط مرزی که جزء چهارم شاخص دستور 13 است، و در عنصر چهارم آرایه FaceIndex قرار دارد، در درایه متناظر متغیر BCType قرار می گیرد.

1. محاسبه تعداد منحنی ها

در این بخش تعداد منحنی های موجود در شبکه که در متغیر RegionNum قرار دارند، در پارامتر خروجی زیربرنامه یعنی NR قرار می گیرند. منحنی های شبکه شامل منحنی های مرزی و همچنین منحنی های غیر مرزی می باشند.

1. بررسی شروع خواندن اطلاعات صفحات شبکه

بعد از اینکه شاخص دستور 13 خوانده و بررسی شد، در این بخش بررسی می شود که آیا پرانتز بازی که نشانه شروع اطلاعات صفحلات شبکه است در سطر جاری قرار دارد یا سطر بعد. روش کار و توضیحات این بخش تقریبا مشابه بخش 19 است. چیزی که به طور اضافه تر در این دستور If بررسی می شود این است که آیا zone-id دستور جاری مخالف صفر است یا خیر. بنابراین اگر کاراکتر جاری رشته TempStr مخالف *پرانتز باز* و zone-id دستور جاری مخالف صفر باشد (FaceIndex(1)/=0) یک کاراکتر از سطر بعد که همان کاراکتر *پرانتز باز* است خوانده شده و در متغیر کاراکتری Char قرار می گیرد.

1. شروع خواندن اطلاعات صفحات شبکه

به طور کلی در این بخش اطلاعات صفحات شبکه با توجه به نوع صفحات خوانده می­شود. طبق شروط دستور If، اگر کاراکتر بعد از کاراکتر جاری در رشته TempStr یا متغیر Char برابر با پرانتز باز باشند، در خط بعد یک حلقه Do از FaceIndex (2) تا FaceIndex (3) اجرا خواهد شد. این مقادیر به ترتیب برابر با اندیس شروع و پایان صفحات شبکه در ناحیه جاری هستند که در بخش 35 خوانده شدند.

1. خواندن یک سطر از اطلاعات صفحات شبکه

با این دستور، یک سطر از اطلاعات صفحات شبکه در ناحیه فعلی خوانده شده (شامل نقاط تشکیل دهنده صفحه و شماره سلول های مجاور) و در متغیر TempStr قرار می گیرد. همچنین شمارنده حلقه بخش 42 (متغیر M) را با عدد یک مقدار دهی اولیه می کند.

1. بررسی سطر خوانده شده از اطلاعات صفحه

در این حلقه، اطلاعات صفحه ای که در بخش قبل خوانده شد مورد بررسی قرار می گیرد. این حلقه تا زمانیکه به انتهای رشته TempStr، یعنی سطر خوانده شده در بخش قبل نرسیده ایم شمارنده رشته را یکی به جلو می برد.

1. استخراج اطلاعات صفحه از سطر خوانده شده

نحوه عملکرد کلی این بخش به طور مکرر در بخش های قبلی توضیح داده شد. دستورات این بخش که داخل حلقه بخش قبل قرار دارند، برای بررسی رشته TempStr به کار می روند. شمارنده این حلقه متغیر M می باشد. در دستور If تا زمانیکه کاراکتر جاری رشته TempStr مخالف Space باشد، یک کاراکتر از این رشته خوانده شده و به انتهای رشته TempChars اضافه می شود. چنانچه کاراکتر جاری رشته TempStr برابر Space بوده و طول رشته TempChars نیز بزرگتر از صفر باشد (دستور Else If)، ابتدا مقدار شمارنده SpaceCount (که در واقع شمارنده تعداد اعداد موجود در رشته TempStr است) یک واحد اضافه شده و سپس مقدار رشته TempChars در درایه متناظر در آرایه کمکی TempIDS قرار می گیرد. این آرایه از نظر ابعاد و عملکرد مانند آرایه IDS است که اطلاعات صفحات شبکه را در خود نگه می دارد، شمارنده بعد اول آرایه TempIDS متغیر FaceCount است که در هر بار اجرای حلقه بخش 40 یک واحد به آن اضافه می شود. این متغیر در واقع شمارنده تعداد سطرها یا به عبارت بهتر شمارنده تعداد صفحات موجود در ناحیه جاری است. در مورد شمارنده بعد دوم یعنی SpaceCount نیز که توضیحات لازم داده شد. نکته قابل توجه در این دستورات این است که با توجه به نوع کد نوشته شده در این بخش، نوع صفحات هر چه که باشند، اعم از خطی، مثلثی، مربعی و چند وجهی (هرمی و منشوری)، برای زیربرنامه قابل خواندن و تجزیه و تحلیل است. زیرا آرایه TempIDS به همان ترتیبی که نقاط تشکیل دهنده و سلول های همسایه یک صفحه در هر سطر از فایل ورودی قرار دارند، آنها را خوانده و به همان ترتیب در خود ذخیره می کند تا عملیات قرار دادن این اطلاعات در آرایه اصلی IDS راحت تر صورت گیرد. به عبارتی با این روش، طول آرایه IDS برای چاپ اطلاعات نهایی پویا خواهد بود. عملکرد دستور Read نیز در بخش 7 توضیح داده شد. دستور Read در اینجا، رشته TempChars را در درایه مربوطه در آرایه TempIDS قرار می دهد. در انتهای حلقه نیز یک واحد به شمارنده حلقه (M) افزوده می شود.

1. بررسی مخلوط بودن صفحات شبکه

این بخش برای بررسی مخلوط بودن یا نبودن صفحات شبکه به کار می رود. اگر جزء پنجم شاخص دستور 13 که همان درایه پنجم آرایه FaceIndex است برابر با صفر باشد، بنابراین نوع صفحات مخلوط است. در غیر اینصورت، نوع صفحات بستگی به مقدار موجود در درایه پنجم آرایه FaceIndex دارد. در این بخش از آرایه یک بعدی FaceType برای ذخیره کردن تعداد نقاط موجود در هر صفحه استفاده می شود. در فایل MSH، اگر نوع صفحات مخلوط باشند، تعداد نقاط موجود در هر صفحه در ابتدای سطر مربوط به آن صفحه نوشته می شود. در غیر این صورت تعداد نقاط در ابتدای سطر نوشته نمی شود. اما برنامه در هر دو حالت، باید تعداد نقاط هر صفحه را در درایه متناظر آرایه FaceType ذخیره کند. بنابراین در شرط If بررسی می شود که اگر درایه پنجم آرایه FaceIndex مخالف صفر باشد، بنابراین نوع صفحه مخلوط نیست و تعداد نقاط صفحه نیز در ابتدای سطر وجود ندارد. در نتیجه با تفاضل مقدار SpaceCount (شمارنده تعداد اعداد موجود در هر سطر) از مقدار 2 (چون دو مقدار از بین اعداد موجود در هر سطر مربوط به همسایه های چپ و راست صفحه می باشند)، تعداد نقاط موجود در هر صفحه بدست می آید. این مقدار باید در درایه متناظر آرایه FaceType ذخیره شود. در صورتیکه نوع صفحات مخلوط باشند، دستور Else اجرا می شود. ابتدا متغیر MixFace که نشان دهنده مخلوط بودن یا نبودن صفحات است مقدار یک را به خود می گیرد (MixFace=0 برای صفحات غیر مخلوط و MixFace=1 برای صفحات مخلوط). سپس عنصر اول آرایه TempIDS در سطر جاری (که نشان دهنده تعداد نقاط صفحه جاری است) در درایه متناظر آرایه FaceType ذخیره می شود.

1. قرار دادن اطلاعات صفحات شبکه در آرایه IDS بر اساس ساختار داده­ای ضلع محور

در ساختار داده ای ضلع محور، سلول های مجاور با صفحه فعلی در درایه های اول و دوم آرایه IDS قرار گرفته و به دنبال آنها شماره نقاط تشکیل دهنده صفحه قرار خواهند گرفت. تعداد نقاط بستگی به نوع صفحه دارد. متغیر TempIDS در بخش 43 به ترتیب مقادیری که در فایل ورودی msh وجود دارند مقدار دهی می شود. می دانیم که در فایل msh اطلاعات صفحات اینگونه است که ابتدا شماره نقاط تشکیل دهنده صفحه می آید و بعد شماره سلول های مجاور با صفحه. اما در ساختار داده ای ضلع محور اول شماره سلول های مجاور می آید و بعد شماره نقاط. بنابراین باید مقادیر دو درایه آخر آرایه TempIDS در دو درایه اول آرایه IDS قرار بگیرند. پس در دو خط اول این بخش، ابتدا درایه ماقبل آخر آرایه TempIDS در درایه اول آرایه IDS قرار می گیرد. برای این کار کافیست مقدار متغیر SpaceCount که در واقع شمارنده بعد دوم آرایه TempIDS و به تعداد عناصر موجود در این آرایه اشاره می کند را یک واحد کم کرده و عنصر مربوطه را در درایه اول آرایه IDS قرار دهیم. در درایه دوم آرایه IDS نیز آخرین عنصر آرایه TempIDS قرار می گیرد. برای درایه های سوم به بعد که قرار است شماره نقاط در آرایه IDS قرار بگیرند از یک حلقه به تعداد نقاط موجود در هر صفحه استفاده شده است. همانطور که در بخش قبل توضیح داده شد، تعداد نقاط هر صفحه در درایه متناظر آرایه FaceType ذخیره می شود. در عبارت IDS(FaceCount,i+2)، مقدار 2 به این دلیل است که ذخیره نقاط صفحه باید از عنصر سوم به بعد در آرایه IDS انجام شود. چون مقدار i نیز از 1 شروع می شود، پس حاصل i+2 از 3 شروع شده و افزایش می یابد. در عبارت TempIDS(FaceCount,i+MixFace)، طبق توضیحات بخش قبل، دلیل استفاده از متغیر MixFace این است که اگر نوع صفحات مخلوط باشد، تعداد مقادیر موجود در هر سطر آرایه TempIDS یکی بیشتر از حالت غیر مخلوط است. پس هنگامی که می خواهیم عنصر مورد نظر از آرایه TempIDS را در عنصر مناسب در آرایه IDS قرار دهیم، باید این مسئله را در نظر بگیریم. در واقع طبق مطالب بیان شده در بخش 1-2، عددی که در حالت مخلوط در ابتدای اطلاعات هر صفحه در فایل msh می آید تنها تعداد نقاط را نشان می دهد و ما نمی خواهیم این عدد در آرایه IDS ذخیره شود (چون قبلا آن را در آرایه FaceType ذخیره کرده ایم).

1. افزودن شمارنده تعداد صفحات شبکه

در این بخش شمارنده صفحات شبکه، و در واقع بعد اول آرایه TempIDS یک واحد اضافه می شود تا حلقه ای که وظیفه خواندن سطرهای مربوط به صفحات شبکه در فایل msh را بر عهده دارد برای خواندن سطر بعد آماده شود. در ادامه متغیرهای SpaceCount، M و N نیز به مقدارهای اولیه خود باز می گردند.

1. کاهش شمارنده حلقه اصلی

در این بخش، از شمارنده حلقه اصلی زیربرنامه که در بخش 3 قرار دارد یک واحد کاسته می‌شود.

1. interior [↑](#footnote-ref-1)