

بررسی انواع روش های چند شبکه ای و بکارگیری روش چندشبکه ای هندسی روی یک حلگر سیالاتی دو بعدی

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| توسعه دهنده: | نام و مقطع تحصیلی  محمد هاشمی اباتری، ارشد | رشته | مهندسی مکانیک | C:\Users\Mohammad\Desktop\University_of_Tehran_logo.svg.png |
| گرایش | تبدیل انرژی |
| حوزه تخصصی پروژه | آیرودینامیک |
| نرم افزار/ زبان برنامه نویسی استفاده شده | Fortran 90 |
| استاد راهنما دکتر/ همکار .................. | | | C:\Users\Mohammad\Desktop\University_of_Tehran_logo.svg.png |
| تهیه کننده مستند: | محمد هاشمی اباتری | | | |
| تاریخ تنظیم سند: | 18 **/** 10 **/**1395 | | | |

فهرست مطالب

[فصل 1: راهنمای کاربری 8](#_Toc469663163)

[1-1. تنطیمات لازم 8](#_Toc469663164)

[1-2. فایل های ورودی 10](#_Toc469663165)

[1-3. فایل های خروجی 10](#_Toc469663166)

[1-4. زیربرنامه های مورد استفاده در برنامه Multigrid 10](#_Toc469663167)

[فصل 2: اعتبار سنجی و نتایج 12](#_Toc469663168)

[2-1. هندسه های مورد بررسی 14](#_Toc469663169)

[2-2. آزمایش های عددی 22](#_Toc469663170)

[2-3. بحث و جمع بندی 61](#_Toc469663171)

[فصل 3: راهنمای آموزشی 62](#_Toc469663172)

[3-1. مقدمه 62](#_Toc469663173)

[3-2. رویه های موجود در روش چند شبکه ای بر اساس نوع معادله 66](#_Toc469663174)

[3-2-1. دستگاه معادلات خطی 66](#_Toc469663175)

[3-2-2. دستگاه معادلات غیرخطی 72](#_Toc469663176)

[3-2-3. به کار بردن روش های مالتی گرید در حلگرهای چند مرحله ای (Multilevel Scheme) 74](#_Toc469663177)

[3-3. دسته بندی روش های چندشبکه ای 76](#_Toc469663178)

[3-3-1. چند شبکه ای هندسی 76](#_Toc469663179)

[3-3-2. چندشبکه ای جبری 76](#_Toc469663180)

[3-4. روش های درشت سازی و درون یابی در روش مالتی گرید 80](#_Toc469663181)

[3-4-1. روش های درشت سازی در مالتی گرید هندسی 80](#_Toc469663182)

[3-4-2. روش های درون یابی در مالتی گرید هندسی 93](#_Toc469663183)

[3-4-3. روش های درشت سازی در مالتی گرید جبری 103](#_Toc469663184)

[3-4-4. روش های درون یابی در مالتی گرید جبری 107](#_Toc469663185)

[فصل 4: پیاده سازی و زیربرنامه های مورد استفاده 109](#_Toc469663186)

[4-1. برنامه اصلی Multigrid 109](#_Toc469663187)

فهرست اشکال

[شکل (1) وارد شدن به منوی Project setting در برنامه فرترن 9](#_Toc469663489)

[شکل (2) افزایش مقدار Stack در Project setting 9](#_Toc469663490)

[**شکل (3) شبکه بی سازمان اطراف ایرفویل** NACA0012**. (الف) و (ب) شبکه ریز و درشت (نمای دور). (ج) و (د) شبکه ریز و درشت (نمای نزدیک)** 14](#_Toc469663491)

[**شکل (4)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای دور)** 15](#_Toc469663492)

[**شکل (5) دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای نزدیک)** 15](#_Toc469663493)

[**شکل (6)** **شبکه بی سازمان اطراف ایرفویل** NACA0012**. (الف) و (ب) شبکه ریز و درشت (نمای دور). (ج) و (د) شبکه ریز و درشت (نمای نزدیک)** 16](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663494)

[**شکل (7)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای دور)** 17](#_Toc469663495)

[**شکل (8)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای نزدیک)** 17](#_Toc469663496)

[**شکل (9)** **شبکه بی سازمان اطراف ایرفویل** NACA0012**. (الف) و (ب) شبکه ریز و درشت (نمای دور). (ج) و (د) شبکه ریز و درشت (نمای نزدیک)** 18](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663497)

[**شکل (10)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای دور)** 19](#_Toc469663498)

[**شکل (11)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای نزدیک)** 19](#_Toc469663499)

[**شکل (12)** **شبکه بی سازمان اطراف ایرفویل** NLR7301**. (الف) و (ب) شبکه ریز و درشت (نمای دور). (ج) و (د) شبکه ریز و درشت (نمای نزدیک)** 20](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663500)

[**شکل (13)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NLR7301 **(نمای دور)** 21](#_Toc469663501)

[**شکل (14)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NLR7301 **(نمای نزدیک)** 21](#_Toc469663502)

[**شکل (15) نمودار همگرایی در** NACA0012 **با 1520 سلول برای**  **و** 22](#_Toc469663503)

[شکل (16) نمودار ضریب فشار حاصله در NACA0012 برای و از سه حلگر RK4، V-Cycle و FSG 23](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663504)

[**شکل (17) کانتور فشار حاصله در** NACA0012 **برای و از سه حلگر الف)** RK4 **ب)** V-Cycle **ج)** FSG 24](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663505)

[**شکل (18)** **نمودار همگرایی در** NACA0012 **با 1520 سلول برای**  **و** 25](#_Toc469663506)

[شکل (19) نمودار ضریب فشار حاصله درNACA0012برای **و** از سه حلگر RK4، V-Cycle و FSG 26](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663507)

[شکل (20) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 27](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663508)

[**شکل (21)** **نمودار همگرایی در** NACA0012 **با 1520 سلول برای**  **و** 28](#_Toc469663509)

[شکل (22) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی 29](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663510)

[شکل (23) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 30](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663511)

[**شکل (24)** **نمودار همگرایی در** NACA0012 **با 2904 سلول برای**  **و** 31](#_Toc469663512)

[شکل (25) نمودار ضریب فشار حاصله در NACA0012 برای و از سه حلگر RK4، V-cycle، FSG و داده های تجربی 32](#_Toc469663513)

[شکل (26) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 33](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663514)

[**شکل (27)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و** 34](#_Toc469663515)

[شکل (28) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی 35](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663516)

[شکل (29) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 36](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663517)

[**شکل (30)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و** 37](#_Toc469663518)

[شکل (31) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle و FSG 38](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663519)

[شکل (32) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 39](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663520)

[**شکل (33)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و** 40](#_Toc469663521)

[شکل (34) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی 41](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663522)

[شکل (35) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 42](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663523)

[**شکل (36)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و** 43](#_Toc469663524)

[شکل (37) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی 44](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663525)

[شکل (38) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 45](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663526)

[**شکل (39)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و** 46](#_Toc469663527)

[شکل (40) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی 47](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663528)

[شکل (41) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 48](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663529)

[**شکل (42)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و** 49](#_Toc469663530)

[شکل (43) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی 50](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663531)

[شکل (44) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 51](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663532)

[**شکل (45)** **نمودار همگرایی در** NLR7301 **با 7144 سلول برای**  **و** 52](#_Toc469663533)

[شکل (46) نمودار ضریب فشار حاصله درNLR7301برایواز سه حلگر RK4، V-Cycle و FSG 53](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663534)

[شکل (47) کانتور فشار حاصله درNLR7301برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 54](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663535)

[**شکل (48)** **نمودار همگرایی در** NLR7301 **با 7144 سلول برای**  **و** 55](#_Toc469663536)

[شکل (49) نمودار ضریب فشار حاصله درNLR7301برایواز سه حلگر RK4، V-Cycle و FSG 56](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663537)

[شکل (50) کانتور فشار حاصله درNLR7301برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 57](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663538)

[**شکل (51)** **نمودار همگرایی در** NLR7301 **با 7144 سلول برای**  **و** 58](#_Toc469663539)

[شکل (52) نمودار ضریب فشار حاصله درNLR7301برایواز سه حلگر RK4، V-Cycle و FSG 59](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663540)

[شکل (53) نمودار ضریب فشار حاصله درNLR7301برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG 60](file:///C:\Users\Mohammad\Desktop\kasri\Multigrid%20Report.docx#_Toc469663541)

[**شکل (54) تغییر دامنه برای موج هایی با فرکانس پایین و بالا** 63](#_Toc469663542)

[**شکل (55) سیکل V به صورت شماتیک** 69](#_Toc469663543)

[**شکل (56) سیکل W به صورت شماتیک** 70](#_Toc469663544)

[**شکل (57) سیکل FMG به صورت شماتیک** 71](#_Toc469663545)

[**شکل (58) سیکل F به صورت شماتیک** 71](#_Toc469663546)

[**شکل (59) مقایسه مراحل دو روش هندسی و جبری** 78](#_Toc469663547)

[**شکل (60) استراتژی تولید شبکه های ریز از شبکه درشت** 80](#_Toc469663548)

[شکل ( 61) نمایی از ناحیه Protective zone جهت جلوگیری از تولید سلول هایی با Aspect ratio نامناسب 83](#_Toc469663549)

[شکل ( 62) نحوه Collapse کردن یک ضلع 87](#_Toc469663550)

[شکل (63) مجموعه S اولیه 88](#_Toc469663551)

[شکل ( 64) شرایط بوجود آمده پس از درشت سازی که منجر به پایین آمدن کیفیت شبکه می شود 90](#_Toc469663552)

[شکل (65) نمایی از نحوه درشت سازی در روش Agglomeration 92](#_Toc469663553)

[شکل (66) سلسله ای از شبکه های درشت تولید شده به روش Agglomeration 92](#_Toc469663554)

[**شکل (67) نمایی از قرار گرفتن دقیق دو گره از لحاظ فیزیکی روی همدیگر** 93](#_Toc469663555)

[شکل ( 68) نحوه عملکرد در درون یابی به روش های 5 نقطه ای و 9 نقطه ای 94](#_Toc469663556)

[**شکل (69) نحوه عملکرد درون یابی به روش خطی** 95](#_Toc469663557)

[**شکل (70) احاطه شدن یک گره از شبکه درشت در مثلثی از شبکه ریز** 95](#_Toc469663558)

[**شکل (71) مکان یابی تمامی گره های شبکه درشت در سلول های شبکه ریز** 96](#_Toc469663559)

[**شکل (72) درون یابی به روش مساحت مثلث ها برای شبکه های مثلثی منظم و سازمان یافته** 98](#_Toc469663560)

[**شکل (73) تقاطع بین مثلث های شبکه ریز و درشت** 99](#_Toc469663561)

[**شکل (74) شرایط قرار گیری یک گره درون یک مثلث** 101](#_Toc469663562)

[**شکل (75) الگوریتم انتخاب گره های شبکه درشت در روش** RS 105](#_Toc469663563)

[**شکل (76) الگوریتم انتخاب گره های شبکه درشت در روش** PMIS 107](#_Toc469663564)

فهرست جداول

[**جدول (1) زیر برنامه های مورد استفاده جهت پیاده سازی** 11](#_Toc469663743)

[**جدول (2) لیست آزمایشات انجام شده در پروژه** 13](#_Toc469663744)

[**جدول (3) نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 1520 سلول برای**  **و** 22](#_Toc469663745)

[**جدول (4) نتایج عددی مربوط به همگرایی در** NACA0012 **با 1520 سلول برای**  **و** 25](#_Toc469663746)

[**جدول (5)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 1520 سلول برای**  **و** 28](#_Toc469663747)

[**جدول (6)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 2904 سلول برای**  **و** 31](#_Toc469663748)

[**جدول (7)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و** 34](#_Toc469663749)

[**جدول (8)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و** 37](#_Toc469663750)

[**جدول (9)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و** 40](#_Toc469663751)

[**جدول (10)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و** 43](#_Toc469663752)

[**جدول (11)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و** 46](#_Toc469663753)

[**جدول (12)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و** 49](#_Toc469663754)

[**جدول (13)** **نتایج عددی مربوط به همگرایی** **در** NLR7301 **با 7144 سلول برای**  **و** 52](#_Toc469663755)

[**جدول (14)** **نتایج عددی مربوط به همگرایی** **در** NLR7301 **با 7144 سلول برای**  **و** 55](#_Toc469663756)

[**جدول (15)** **نتایج عددی مربوط به همگرایی** **در** NLR7301 **با 7144 سلول برای**  **و** 58](#_Toc469663757)

چکیده

در این پروژه به بررسی روش های مالتی گرید هندسی و جبری پرداخته شده است. الگوریتم های موجود و قابل استفاده برای هر کدام از این روش ها آورده شده است. سپس انواع سیکل های روش چند شبکه ای معرفی شده و نحوه عملکرد هر کدام به صورت جداگانه شرح داده شده است. هر کدام از این سیکل ها شامل مولفه هایی همچون درشت سازی و درون یابی بوده که برای هر کدام چند روش ذکر شده و مفصل مورد بررسی قرار گرفته است. در نهایت از دو روش مالتی گرید V\_Cycle و Full Single Grid و درشت سازی بر اساس تابع فاصله و درون یابی بر اساس مساحت های وزن دار برای انواع شبکه های مثلثی استفاده شده و نتایج آن ها با روش رانج کوتای مرتبه 4 مقایسه شده است. همچنین در برخی موارد ضریب فشار حاصل از روش های عددی جهت بررسی درستی حل با نتایج تجربی مقایسه شده است.

**کلمات کلیدی**: مالتی گرید هندسی و جبری، درشت سازی، درون یابی، V\_Cycle، Full Single Grid، رانج کوتای مرتبه 4

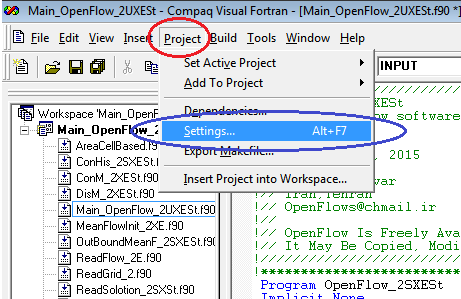
1. راهنمای کاربری

در این فصل فایل های ورودی و خروجی در برنامه و همچنین زیر برنامه های مورد استفاده در برنامه اصلی معرفی خواهند شد.

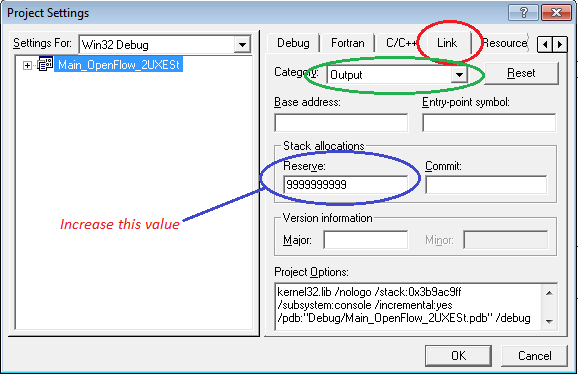
* 1. تنطیمات لازم

این برنامه با استفاده از زبان برنامه نویسی فرترن تدوین و از کامپایلر Compaq Visual Fortran 90 استفاده شده است. جهت آماده سازی برنامه جهت اجرا ابتدا باید تمام زیربرنامه های مورد نیاز که در جدول (1) آورده شده است، تهیه و به برنامه اصلی Multigrid اضافه شود.

در ابتدای برنامه یک پارامتر بنام Dim تعریف شده که مقدار حداکثر بعد آرایه های استفاده شده را تعیین می کند. این مقدار باید در حدی انتخاب شود که Stack برنامه اجازه می دهد. البته در صورتیکه بدلیل بزرگ بودن شبکه با خطایStack overflow مواجه شویم و لازم باشد مقدار Stack را افزایش دهیم می توان طبق شکل های (1) و (2) این کار را در نرم افزار Compaq Visual Fortran 90 انجام داد:



شکل (1) وارد شدن به منوی Project setting در برنامه فرترن



شکل (2) افزایش مقدار Stack در Project setting

* 1. فایل های ورودی

سه فایل ورودی تحت عنوان های Mesh2.txt , Mesh1.txt , Setting.txt که به ترتیب حاوی مشخصات جریان آزاد به منزله تنظیمات اولیه و شبکه های محاسباتی است، به عنوان ورودی های لازم جهت اجرای برنامه در پوشه حاوی برنامه اصلی قرار دارد. لازم به ذکر است که فایل Mesh1.txt حاوی اطلاعات شبکه اولیه یعنی شبکه ریز و فایل Mesh2.txt حاوی اطلاعات شبکه درشت می باشد.

* 1. فایل های خروجی

پس از اجرای برنامه نتایج حاصله در چهار فایل جداگانه به عنوان خروجی برنامه ذخیره می شود. در فایل های Contours.Plt و CP.Plt به ترتیب مقادیر بی بعد جریان در هر کدام از نقاط شبکه و مقدار ضریب فشار در نقاط روی مرز دیوار ذخیره خواهد شد. فایل ResMass.Plt روند همگرایی را بر اساس مقدار باقیمانده های معادله جرم چاپ خواهد کرد. همچنین نتایج مربوط به حل یعنی مقادیر بقایی در فایل SolutionData.txt چاپ خواهد شد.

* 1. زیربرنامه های مورد استفاده در برنامه Multigrid

در جدول زیر نام زیربرنامه های مورد استفاده برای پیاده سازی و شماره سند مربوط به هر کدام آورده شده است. همچنین فایل فرترن هر زیربرنامه با نام زیربرنامه در دسترس است.

**جدول (1) زیر برنامه های مورد استفاده جهت پیاده سازی**

|  |  |  |
| --- | --- | --- |
| **ردیف** | **نام زیربرنامه** | **شماره سند** |
| 1 | Read\_2DMesh | MC2F003F1 |
| 2 | Read\_SettingV1 | MC2F001F1 |
| 3 | MeshBC | MC2F022F1 |
| 4 | GeoCal2D | MC2F026F1 |
| 5 | InitMeanFlow\_Inviscid | MC2F010F1 |
| 6 | BC\_Wall | MC2F011F1 |
| 7 | BC\_Riemann | MC2F015F1 |
| 8 | BC\_InFlow | MC2F017F1 |
| 9 | BC\_SubOutFlow | MC2F016F1 |
| 10 | BC\_Symmetry | MC2F018F1 |
| 11 | TimSTP\_Inviscid | MC2F012F1 |
| 12 | ConMeanFlow\_AUSM | MC2F008F1 |
| 13 | Interpolation | MC2F100F1 |
| 14 | Coarsening | MC2F101F1 |
| 15 | Read\_2DMesh1 |  |
| 16 | Read\_2DMesh2 |  |
| 17 | Edge\_To\_Cell | MC5F088F1 |
| 18 | ResMass | MC2F004F1 |
| 19 | Write\_ResultsV1 | MC2F002F1 |

1. اعتبار سنجی و نتایج

در این پروژه دو روش مالتی گرید V-Cycle و Full Single Grid مورد بررسی قرار گرفته است. در برنامه نوشته برای این دو روش از رانج کوتای مرتبه 4 استفاده شده که نتایج بدست آمده را با حالتی که حلگر فقط از رانج کوتای مرتبه 4 استفاده می کند، مورد بررسی و مقایسه قرار داده ایم. آزمایشات روی ایرفویل هایی با 3 شبکه متفاوت انجام گرفته است. ابتدا از شبکه درشت تری استفاده شده و سپس شبکه های ریز تری را مورد بررسی قرار داده ایم. جهت بررسی درستی کدهای نوشته شده برخی از نتایج بدست آمده همچون ضریب فشار و کانتور فشار با نتایج بدست آمده از روش رانج کوتای مرتبه 4 مقایسه شده است. در جدول زیر مشخصات آزمایش های انجام شده آورده شده است.

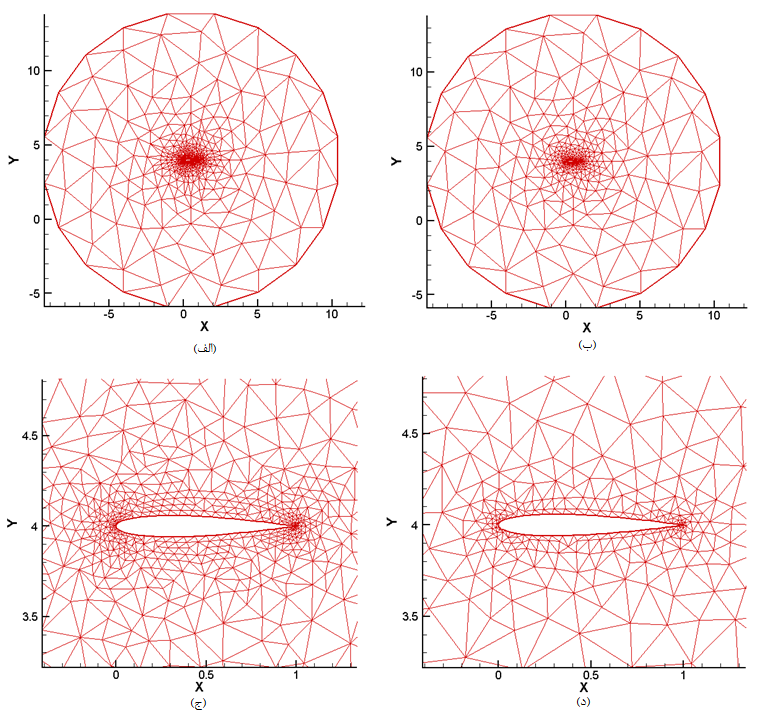
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **زاویه حمله** | **عدد ماخ** | **تعداد سلول های شبکه درشت** | **تعداد سلول های شبکه ریز** | **عنوان هندسه** | **شماره آزمایش** |
| 0 | 0.65 | 892 | 1520 | NACA0012 | 1-1 |
| 4 | 0.75 | 892 | 1520 | NACA0012 | 1-2 |
| 1 | 0.85 | 892 | 1520 | NACA0012 | 1-3 |
| 3 | 0.5 | 1133 | 2904 | NACA0012 | 2-1 |
| 0 | 0.5 | 1520 | 3218 | NACA0012 | 3-1 |
| 4 | 0.75 | 1520 | 3218 | NACA0012 | 3-2 |
| 0 | 0.8 | 1520 | 3218 | NACA0012 | 3-3 |
| 1 | 0.85 | 1520 | 3218 | NACA0012 | 3-4 |
| 0 | 0.95 | 1520 | 3218 | NACA0012 | 3-5 |
| 0 | 1.2 | 1520 | 3218 | NACA0012 | 3-6 |
| 0 | 0.3 | 3429 | 7144 | NLR7301 | 4-1 |
| 0 | 0.5 | 3429 | 7144 | NLR7301 | 4-2 |
| 0 | 0.65 | 3429 | 7144 | NLR7301 | 4-3 |

**جدول (2) لیست آزمایشات انجام شده در پروژه**

* 1. هندسه های مورد بررسی

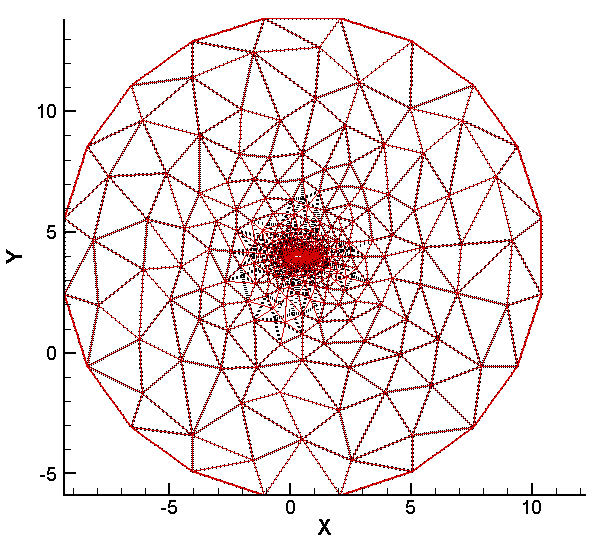
هندسه مورد استفاده در آزمایش های شماره 1 :

ایرفویل مورد بررسی در این قسمت NACA0012 دارای 1520 سلول بوده که به صورت شماتیک به همراه شبکه درشت حاصل ازآن بعد از یک مرحله درشت سازی در زیر نشان داده شده است.

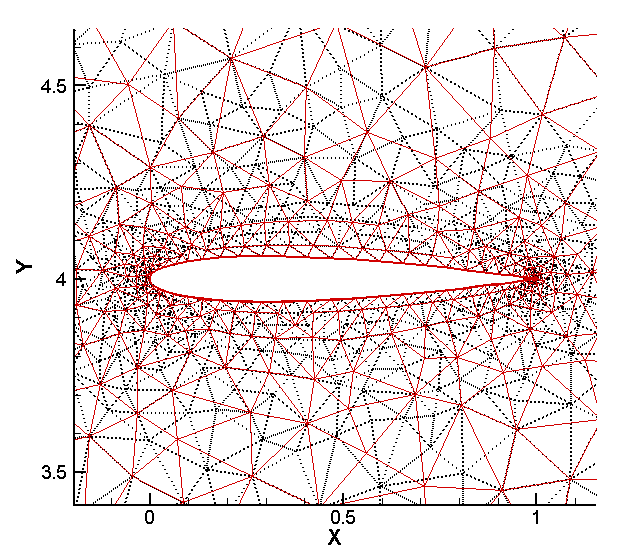
****

**شکل (3) شبکه بی سازمان اطراف ایرفویل** NACA0012**. (الف) و (ب) شبکه ریز و درشت (نمای دور). (ج) و (د) شبکه ریز و درشت (نمای نزدیک)**

همانطور که گفته شد، روش مالتی گرید از سلسله ای از شبکه های درشت استفاده می کند که همگی از شبکه ریز اولیه بدست آمده اند. برای این منظور در این جا شبکه اولیه بعد از یک مرحله درشت سازی دارای 892 سلول می شود. برای مشاهده میزان درشت سازی، شبکه ها را روی هم قرار دادیم که در زیر نمایش داده شده است. سلول های با رنگ مشکی نشان دهنده شبکه ریز و سلول ها با رنگ قرمز نشان دهنده شبکه درشت تر می باشد.



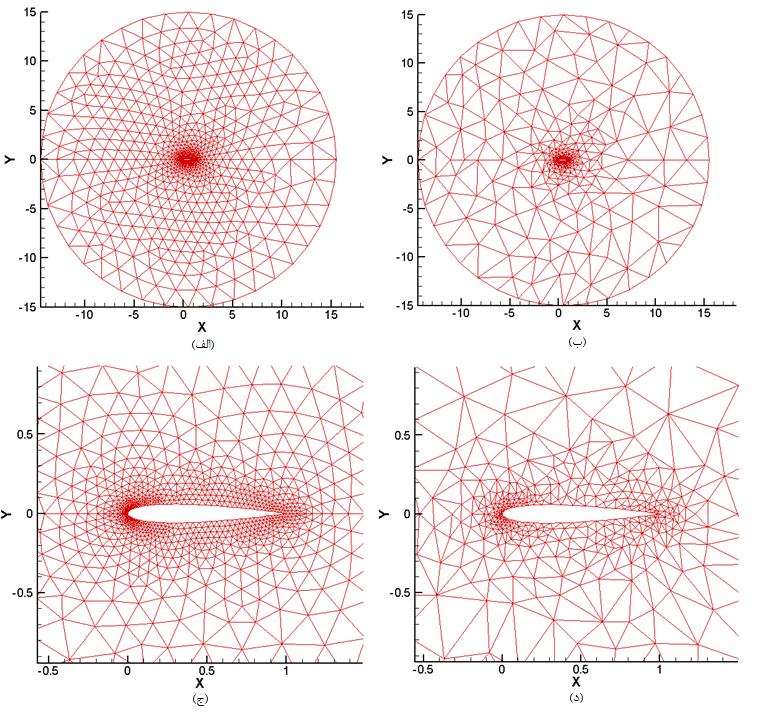
**شکل (4)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای دور)**



**شکل (5) دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای نزدیک)**

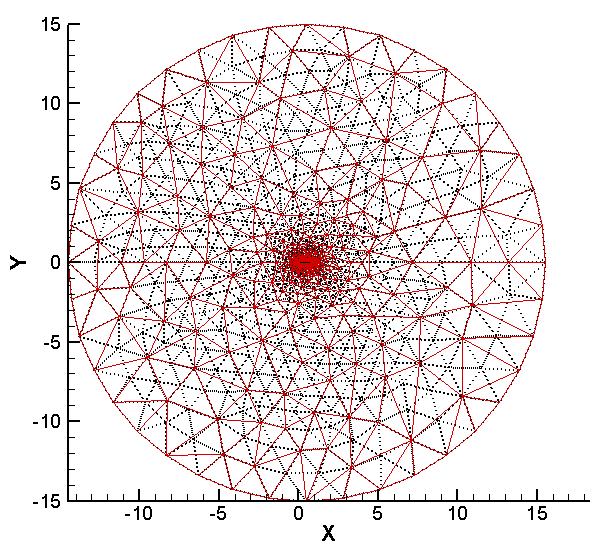
هندسه مورد استفاده در آزمایش های شماره 2 :

ایرفویل مورد بررسی NACA0012 با 2904 سلول می باشد. در زیر این هندسه با دو شبکه ریز و درشت استفاده شده در الگوریتم چند شبکه ای نشان داده شده است.

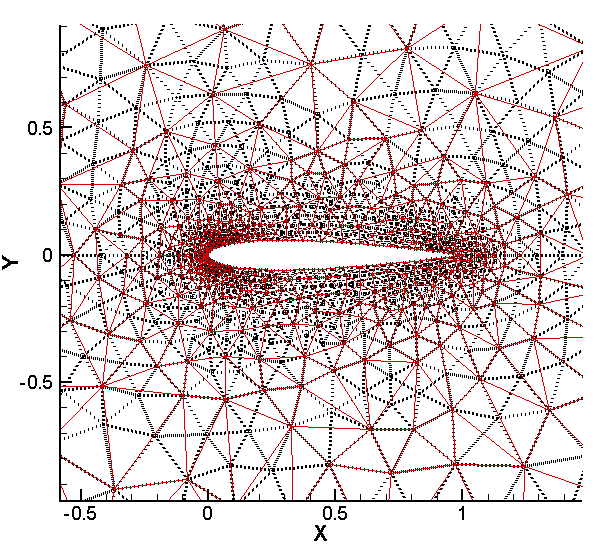


**شکل (6)** **شبکه بی سازمان اطراف ایرفویل** NACA0012**. (الف) و (ب) شبکه ریز و درشت (نمای دور). (ج) و (د) شبکه ریز و درشت (نمای نزدیک)**

تعداد سلول های شبکه هندسه فوق بعد از یکبار درشت سازی به 1133سلول می رسد. برای مشاهده میزان درشت سازی، شبکه ها را روی هم قرار دادیم که در زیر نمایش داده شده است. سلول های با رنگ مشکی نشان دهنده شبکه ریز و سلول ها با رنگ قرمز نشان دهنده شبکه درشت تر می باشد.



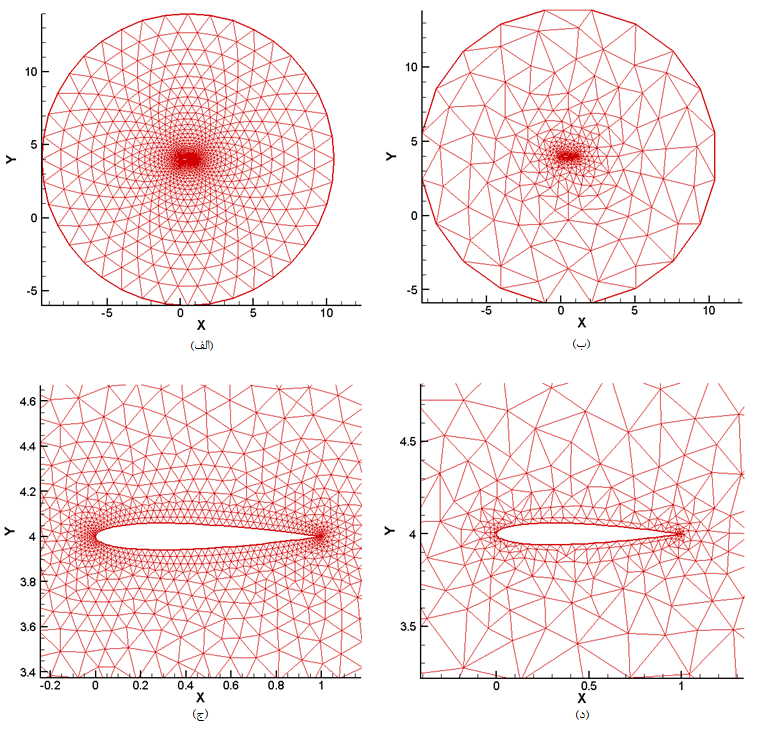
**شکل (7)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای دور)**



**شکل (8)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای نزدیک)**

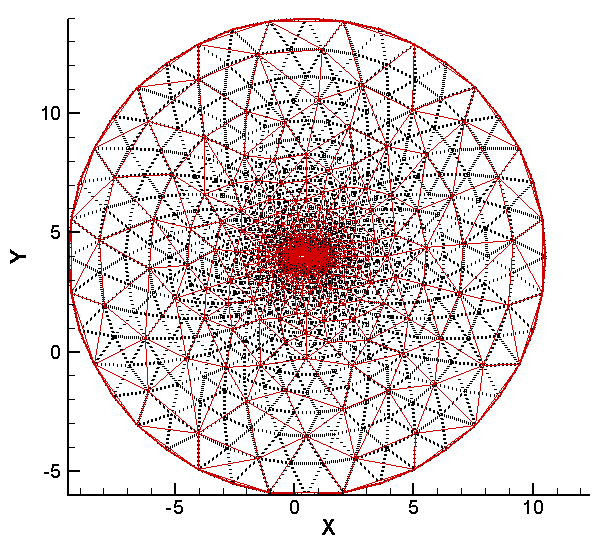
هندسه مورد استفاده در آزمایش های شماره 3 :

ایرفویل مورد بررسی NACA0012 با 3218 سلول می باشد. در زیر دو شبکه ریز و درشت این هندسه مورد استفاده در روش چند شبکه ای نشان داده شده است.

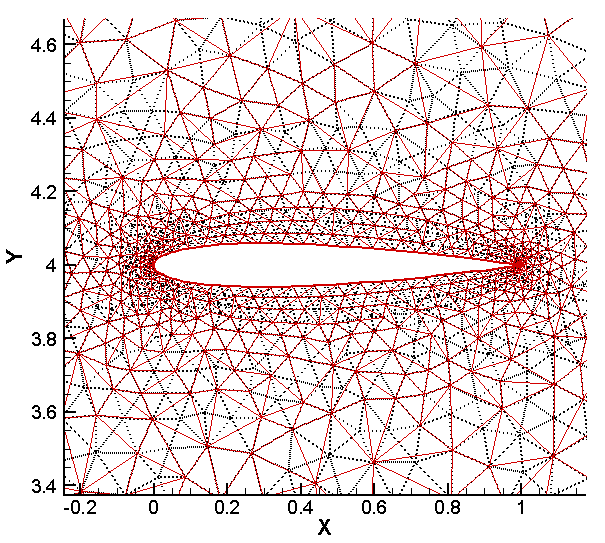


**شکل (9)** **شبکه بی سازمان اطراف ایرفویل** NACA0012**. (الف) و (ب) شبکه ریز و درشت (نمای دور). (ج) و (د) شبکه ریز و درشت (نمای نزدیک)**

تعداد سلول های این شبکه بعد از یکبار درشت سازی به 1520 سلول کاهش می یابد. با قرار دادن دو شبکه ریز و درشت روی یکدیگر، میزان درشت سازی به صورت شماتیک قابل درک خواهد بود که در زیر آورده شده است. در این قسمت نیز سلول های با رنگ مشکی نشان دهنده شبکه ریز و سلول ها با رنگ قرمز نشان دهنده شبکه درشت تر می باشد.



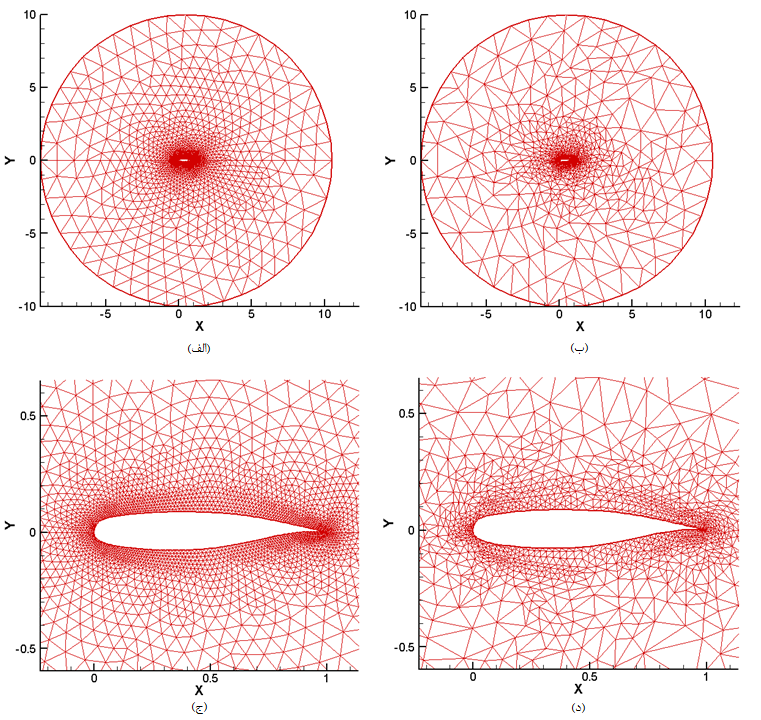
**شکل (10)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای دور)**



**شکل (11)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NACA0012 **(نمای نزدیک)**

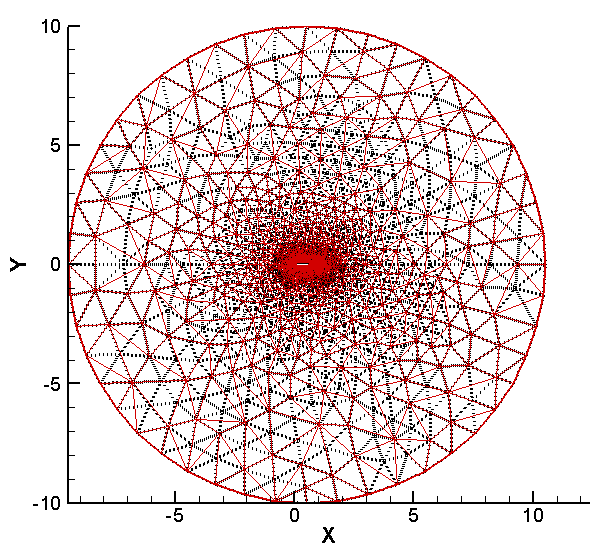
هندسه مورد استفاده در آزمایش های شماره 4 :

در این آزمایش از NLR7301 با 7144 سلول استفاده شده است. این هندسه به همراه شبکه درشت مورد استفاده در الگوریتم چند شبکه ای در زیر نشان داده شده است:



**شکل (12)** **شبکه بی سازمان اطراف ایرفویل** NLR7301**. (الف) و (ب) شبکه ریز و درشت (نمای دور). (ج) و (د) شبکه ریز و درشت (نمای نزدیک)**

این شبکه بعد از یکبار درشت سازی دارای 3429 سلول شده است. در این قسمت همانند دو آزمایش قبل جهت درک میزان درشت سازی دو شبکه را روی هم قرار داده ایم:



**شکل (13)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NLR7301 **(نمای دور)**



**شکل (14)** **دو شبکه ریز و درشت قرار گرفته روی هم اطراف ایرفویل** NLR7301 **(نمای نزدیک)**

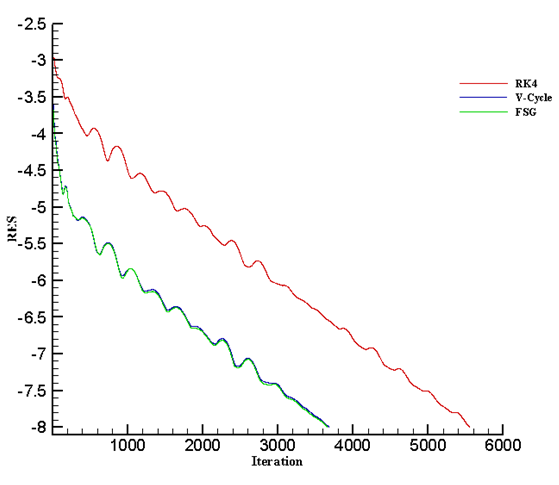
* 1. آزمایش های عددی

در این قسمت نتایج عددی مربوط به هندسه های فوق آورده می شود.

آزمایش 1-1

این آزمایش برای هندسه 1 برای ماخ 0.65 و زاویه حمله صفر برای سه حلگر RK4، V-Cycle وFull Single Grid (FSG) مورد بررسی قرار گرفته است که نتایج آن به صورت زیر می باشد.

* + 1. روند همگرایی بر اساس تعداد تکرارها



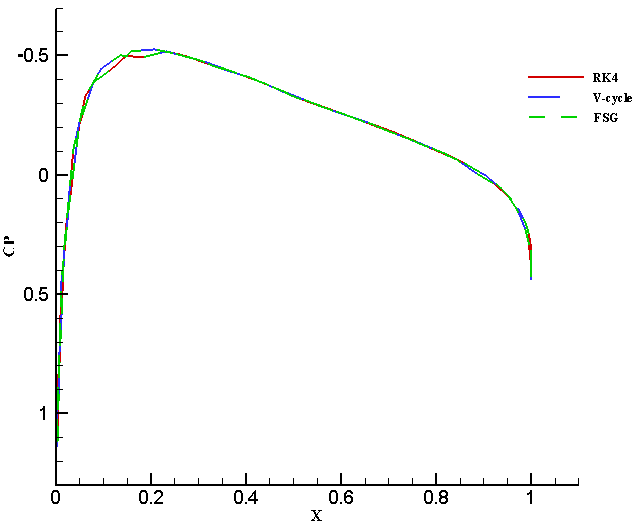
**جدول (3) نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 1520 سلول برای**  **و**

**شکل (15) نمودار همگرایی در** NACA0012 **با 1520 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 5563 | 43.14 | RK4 |
| 3688 | 37.18 | V\_Cycle |
| 3662 | 36.92 | Full Single Grid |

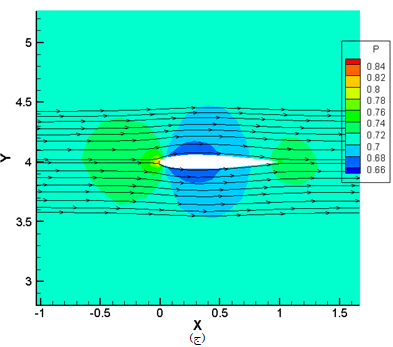
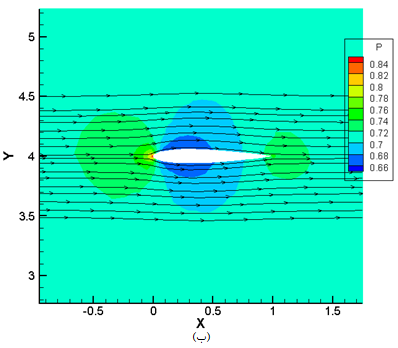
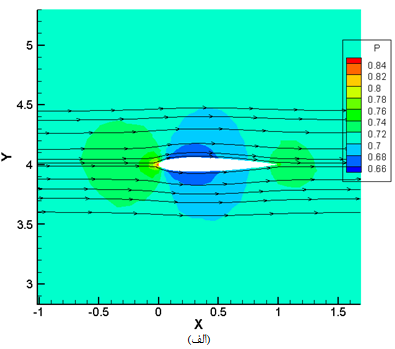
* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش

شکل (16) نمودار ضریب فشار حاصله در NACA0012 برای و از سه حلگر RK4، V-Cycle و FSG



* + 1. مقایسه کانتور فشار P برای هر سه روش

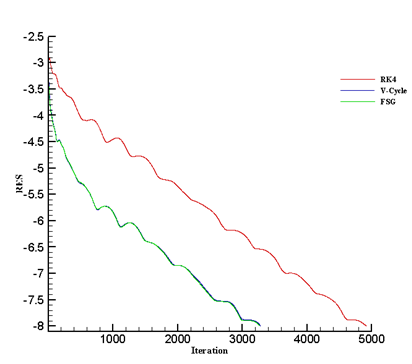
**شکل (17) کانتور فشار حاصله در** NACA0012 **برای و از سه حلگر الف)** RK4 **ب)** V-Cycle **ج)** FSG



آزمایش 1-2

در این آزمایش، هندسه قبل را برای ماخ 0.75 و زاویه حمله 4 درجه مورد بررسی قرار می دهیم.

* + 1. روند همگرایی بر اساس تعداد تکرارها

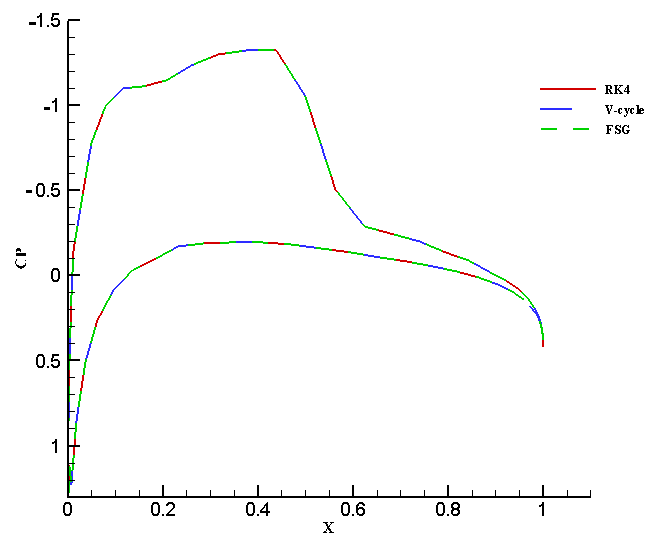


**شکل (18)** **نمودار همگرایی در** NACA0012 **با 1520 سلول برای**  **و**

**جدول (4) نتایج عددی مربوط به همگرایی در** NACA0012 **با 1520 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **حلگر مورد استفاده** | **زمان همگرایی (s)** | **تعداد تکرارهای همگرایی** |
| RK4 | 39.04 | 4917 |
| V-Cycle | 32.86 | 3284 |
| Full Single Grid | 32.4 | 3270 |

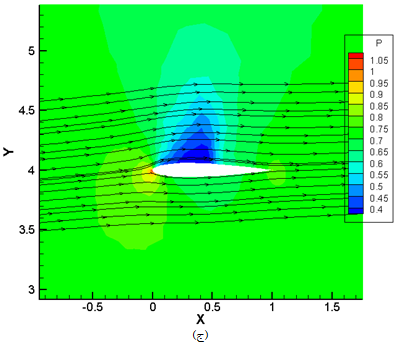
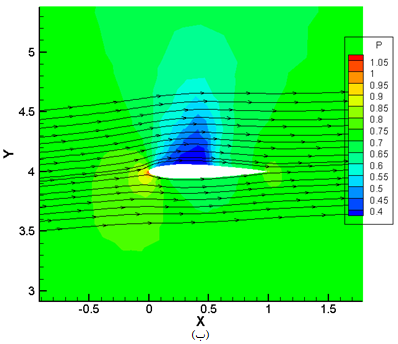
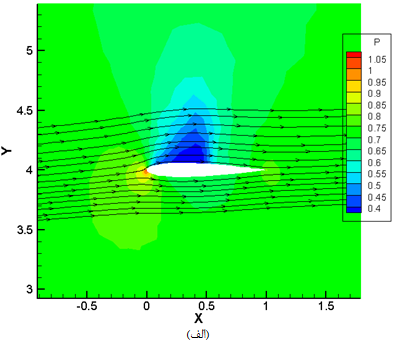
* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش



شکل (19) نمودار ضریب فشار حاصله درNACA0012برای **و** از سه حلگر RK4، V-Cycle و FSG

* + 1. مقایسه کانتور فشار P برای هر سه روش

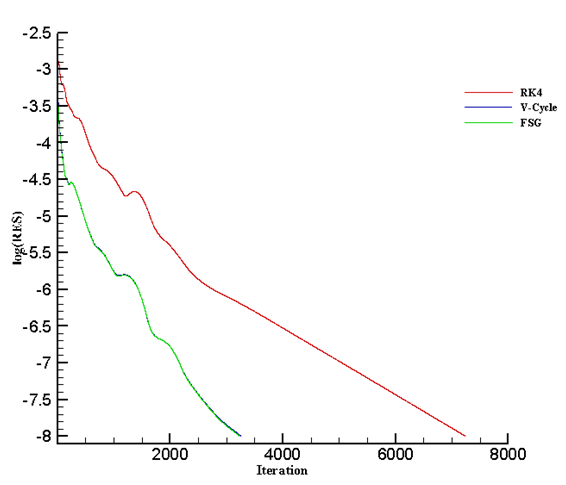
شکل (20) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG



آزمایش 1-3

آزمایش هندسه قبل برای ماخ 0.85 و زاویه حمله 1 درجه برای هر سه حلگر تکرار شده است.

* + 1. روند همگرایی بر اساس تعداد تکرارها

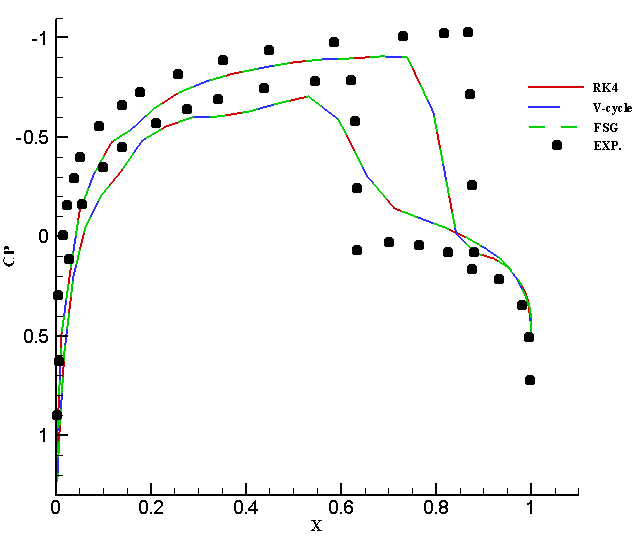


**شکل (21)** **نمودار همگرایی در** NACA0012 **با 1520 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 7241 | 58.61 | RK4 |
| 3260 | 35.5 | V\_Cycle |
| 3238 | 33.3 | Full Single Grid |

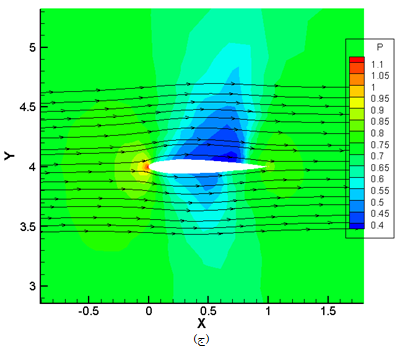
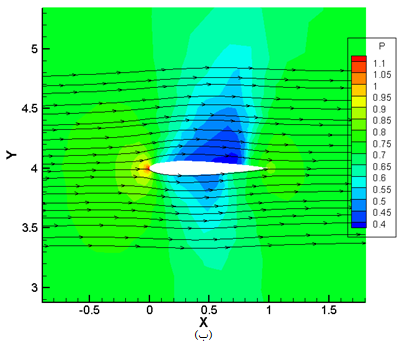
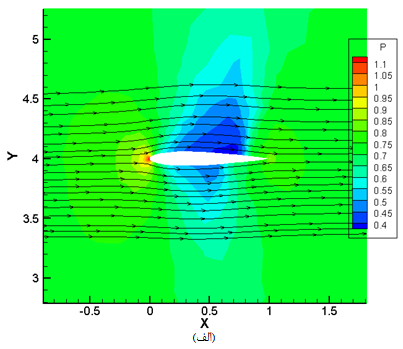
**جدول (5)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 1520 سلول برای**  **و**

* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش



شکل (22) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی

* + 1. مقایسه کانتور فشار P برای هر سه روش

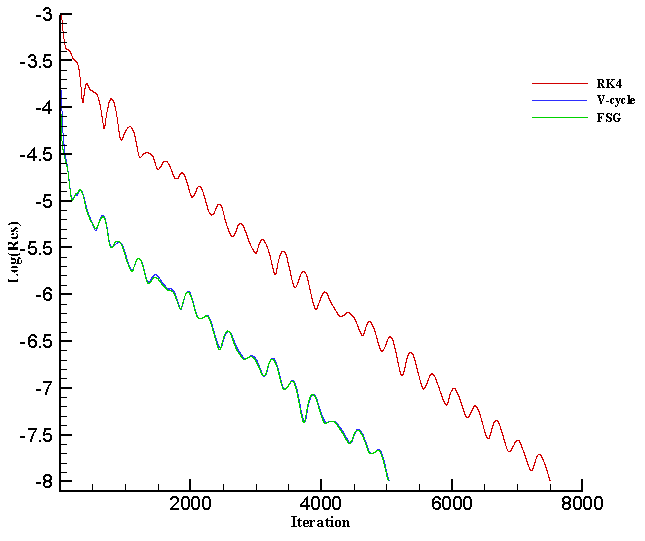


شکل (23) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG

آزمایش 2-1

در این آزمایش هندسه شماره 2 برای ماخ 0.5 و زاویه حمله 3 برای سه حلگر RK4، V-Cycle وFull Single Grid (FSG) مورد بررسی قرار گرفته است که نتایج آن به صورت زیر می باشد.

* + 1. روند همگرایی بر اساس تعداد تکرارها

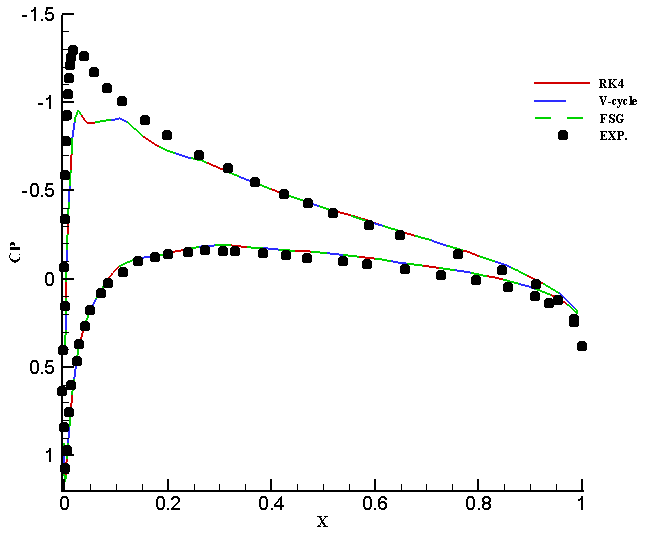


**شکل (24)** **نمودار همگرایی در** NACA0012 **با 2904 سلول برای**  **و**

**جدول (6)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 2904 سلول برای**  **و**

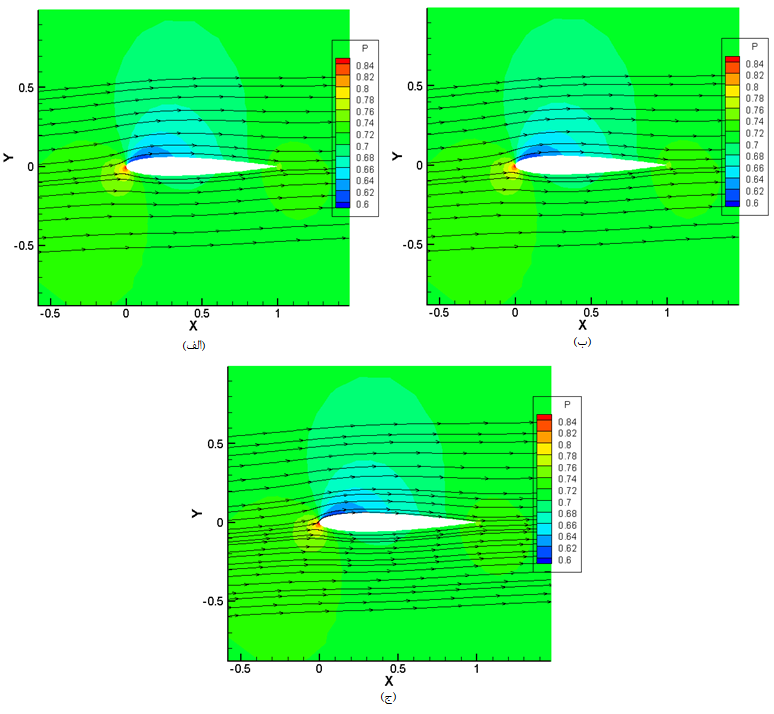
|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 7508 | 108.16 | RK4 |
| 5039 | 83.2 | V\_Cycle |
| 5029 | 81.3 | Full Single Grid |

* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش



شکل (25) نمودار ضریب فشار حاصله در NACA0012 برای و از سه حلگر RK4، V-cycle، FSG و داده های تجربی

* + 1. مقایسه کانتور فشار P برای هر سه روش

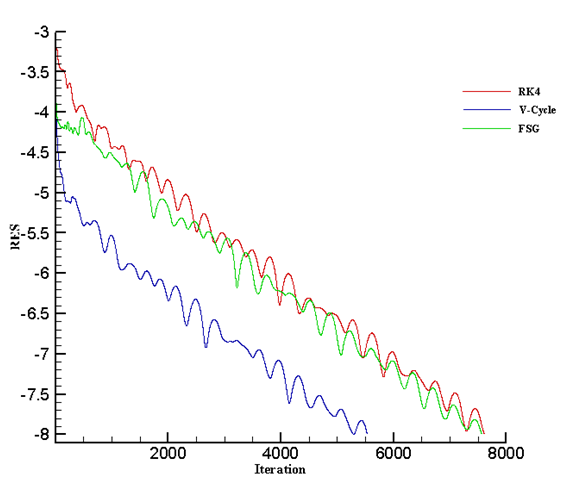


شکل (26) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG

آزمایش 3-1

در این آزمایش هندسه شماره 3 برای ماخ 0.5 و زاویه حمله صفر برای سه حلگر RK4، V-Cycle وFull Single Grid (FSG) مورد بررسی قرار گرفته است که نتایج آن به صورت زیر می باشد.

* + 1. روند همگرایی بر اساس تعداد تکرارها

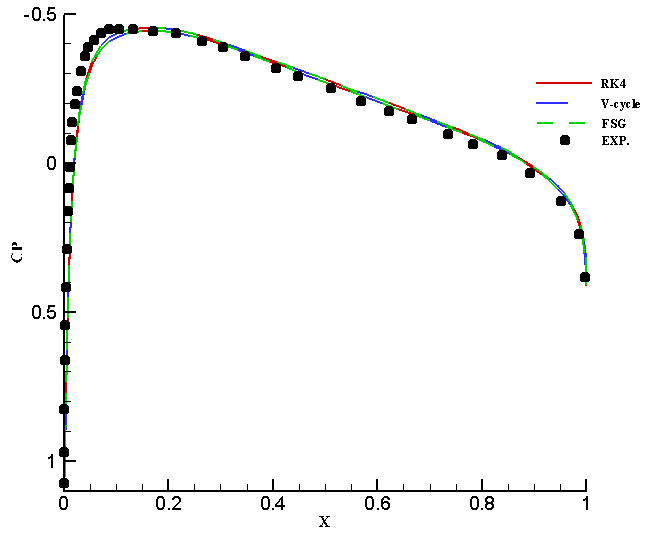


**شکل (27)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و**

**جدول (7)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 7611 | 123.37 | RK4 |
| 5530 | 105.09 | V\_Cycle |
| 7565 | 136.15 | Full Single Grid |

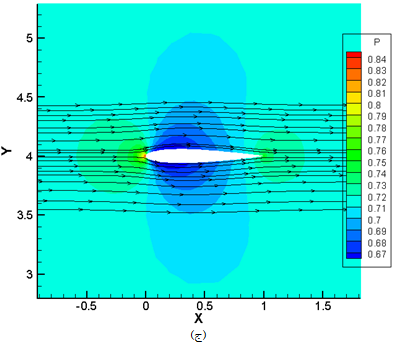
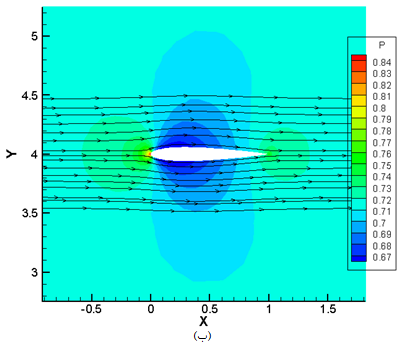
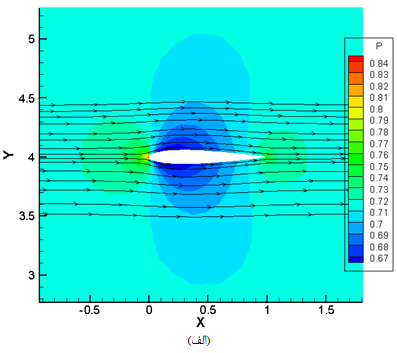
* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش



شکل (28) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی

* + 1. مقایسه کانتور فشار P برای هر سه روش

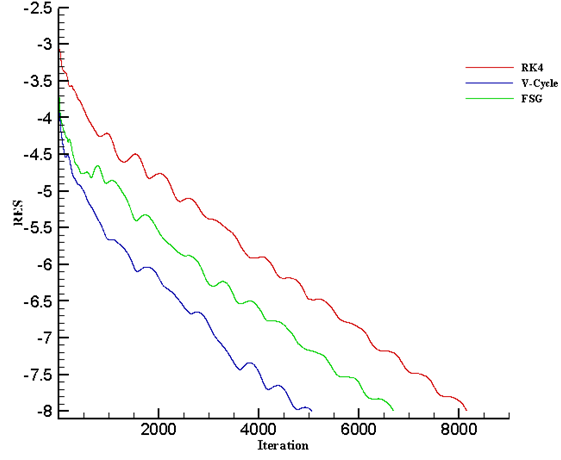
شکل (29) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG



آزمایش 3-2

در این آزمایش هندسه قبل برای ماخ 0.75 و زاویه حمله 4 درجه مورد بررسی قرار گرفته است.

* + 1. روند همگرایی بر اساس تعداد تکرارها

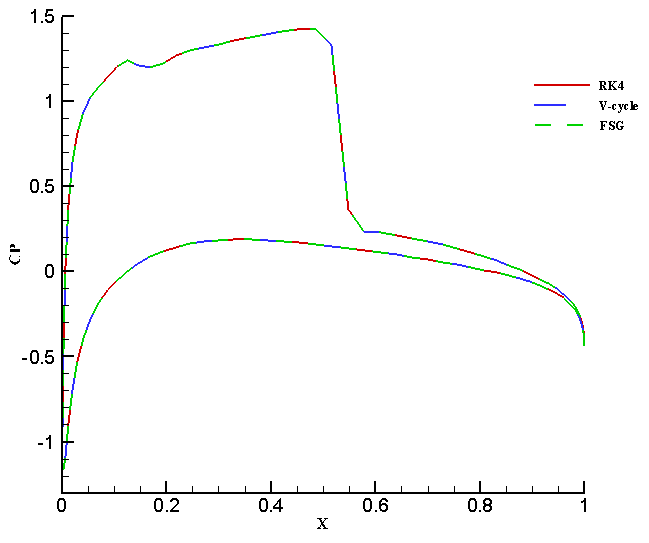


**شکل (30)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و**

**جدول (8)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 8155 | 129.81 | RK4 |
| 5063 | 96.5 | V\_Cycle |
| 6690 | 120.79 | Full Single Grid |

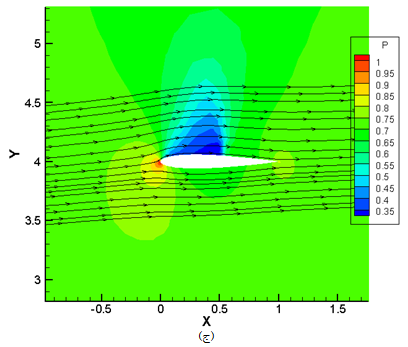
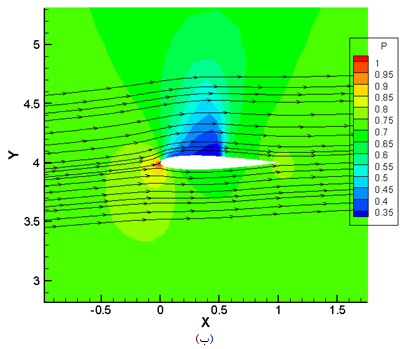
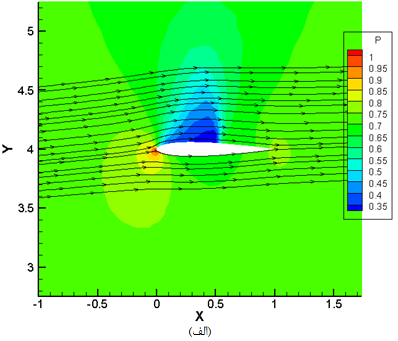
* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش



شکل (31) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle و FSG

* + 1. مقایسه کانتور فشار (P) برای هر سه روش

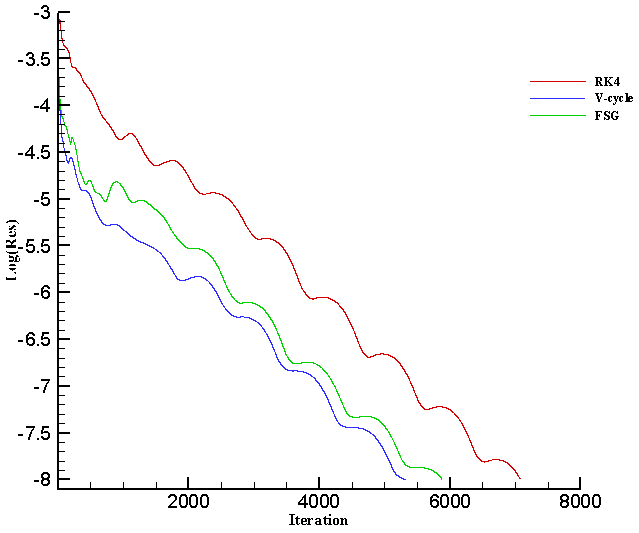
شکل (32) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG



آزمایش 3-3

در این آزمایش هندسه شماره 3 برای ماخ 0.8 و زاویه حمله صفر درجه برای سه حلگر RK4، V-Cycle و Full Single Grid (FSG) مورد بررسی قرار گرفته است.

* + 1. روند همگرایی بر اساس تعداد تکرارها

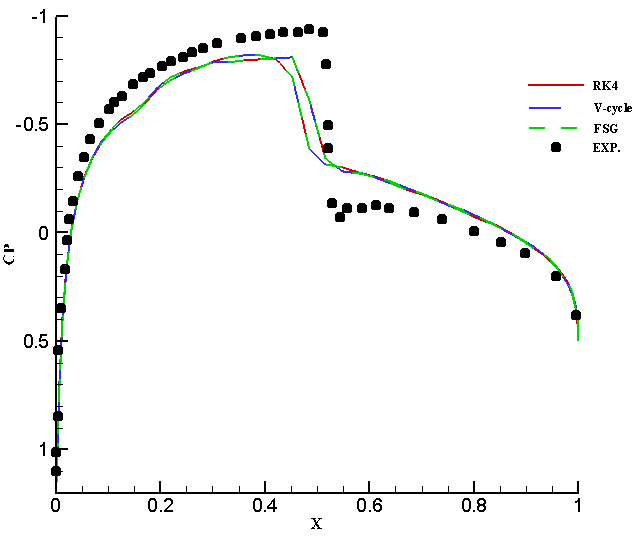


**شکل (33)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 7078 | 112.1 | RK4 |
| 5337 | 102.8 | V\_Cycle |
| 5884 | 108.47 | Full Single Grid |

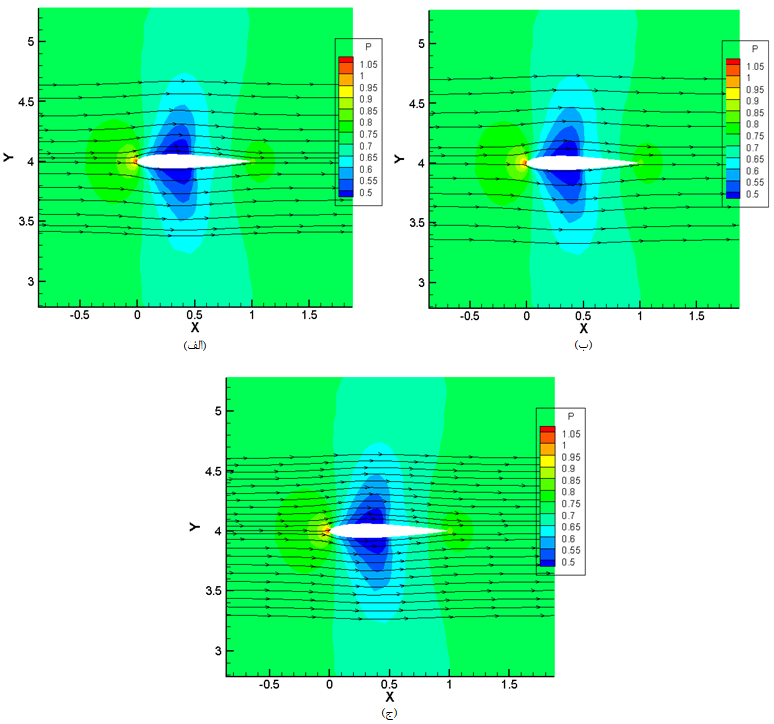
**جدول (9)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و**

* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش



شکل (34) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی

* + 1. مقایسه کانتور فشار (P) برای هر سه روش

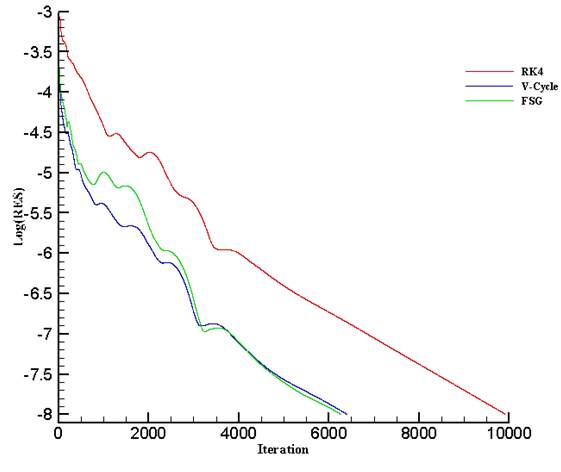


شکل (35) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG

آزمایش 3-4

در این آزمایش هندسه شماره 3 برای ماخ 0.85 و زاویه حمله 1 درجه برای سه حلگر RK4، V-Cycle و Full Single Grid (FSG) مورد بررسی قرار گرفته است.

* + 1. روند همگرایی بر اساس تعداد تکرارها

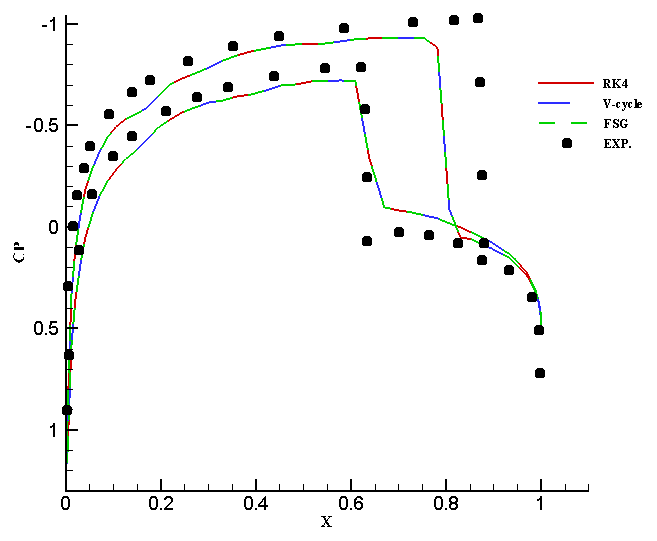


**شکل (36)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 9925 | 159.9 | RK4 |
| 6408 | 123.65 | V\_Cycle |
| 6277 | 114.9 | Full Single Grid |

**جدول (10)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و**

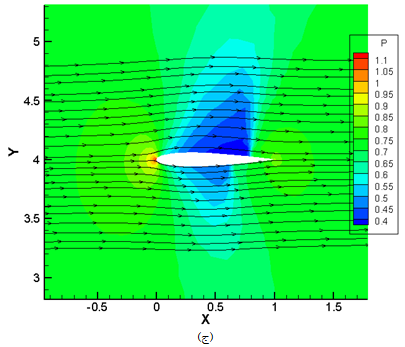
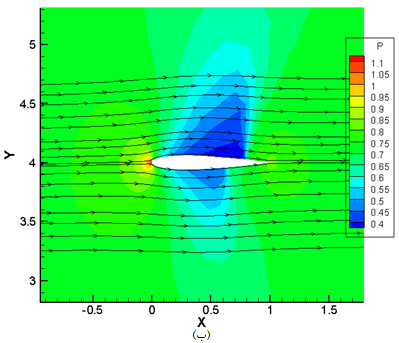
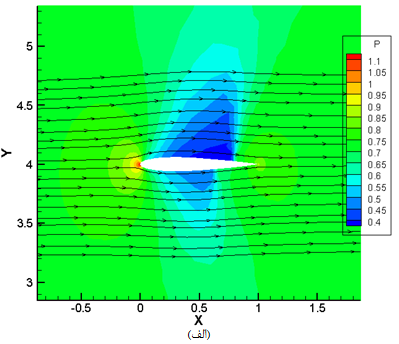
* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش



شکل (37) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی

* + 1. مقایسه کانتور فشار (P) برای هر سه روش

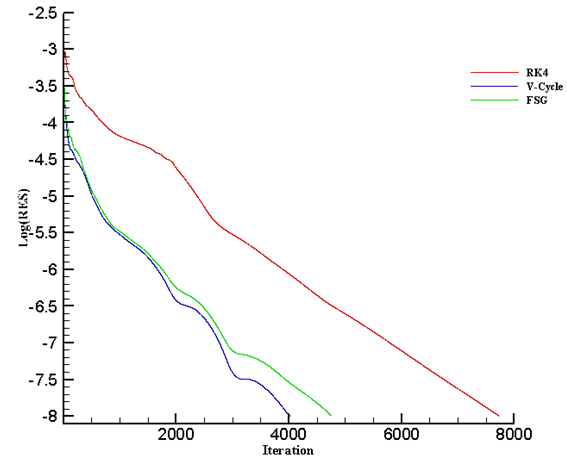
شکل (38) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG



آزمایش 3-5

آزمایش قبل با ماخ 0.95 و زاویه حمله صفر درجه تکرار شده است.

* + 1. روند همگرایی بر اساس تعداد تکرارها



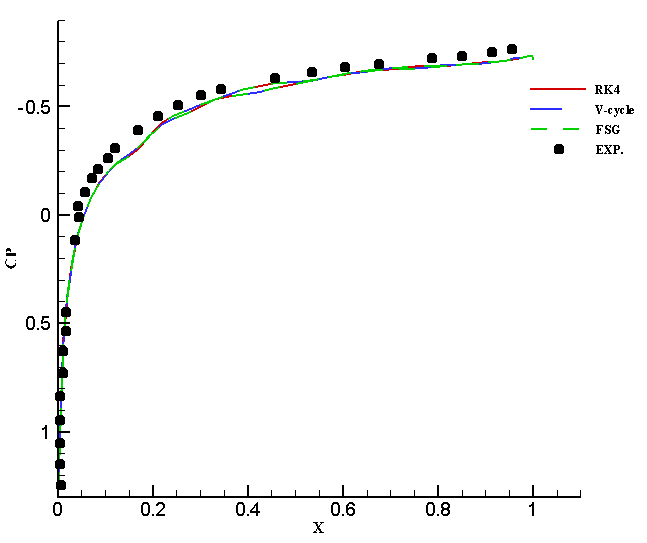
**شکل (39)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و**

**جدول (11)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 7740 | 125.09 | RK4 |
| 4027 | 86.92 | V\_Cycle |
| 4769 | 91.48 | Full Single Grid |

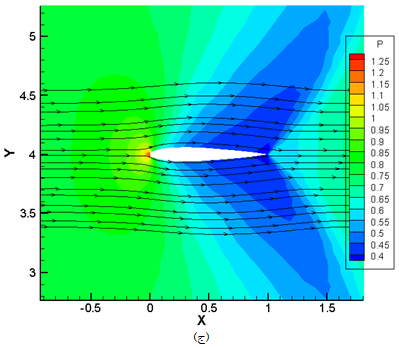
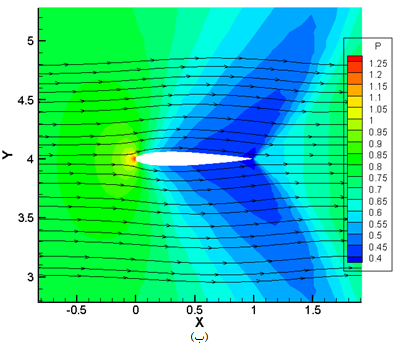
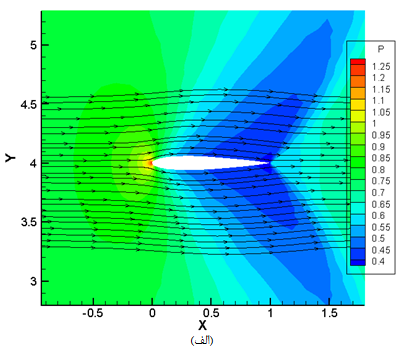
* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش

شکل (40) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی



* + 1. مقایسه کانتور فشار (P) برای هر سه روش

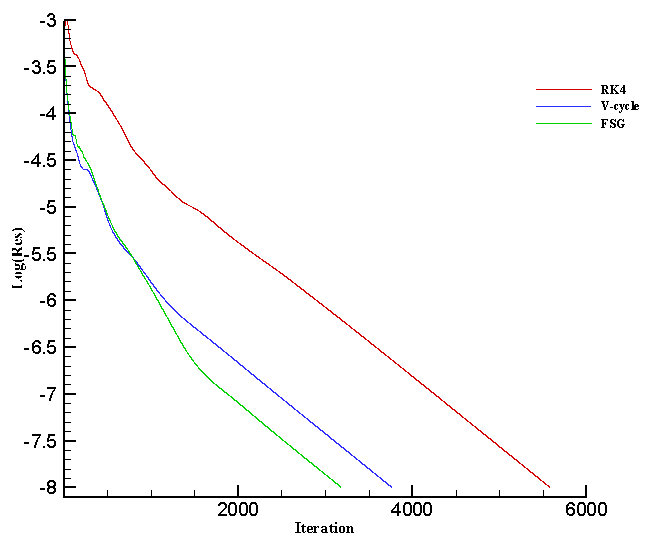
شکل (41) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG



آزمایش 3-6

آزمایش قبل با ماخ 1.2 و زاویه حمله صفر درجه تکرار شده است.

* + 1. روند همگرایی بر اساس تعداد تکرارها



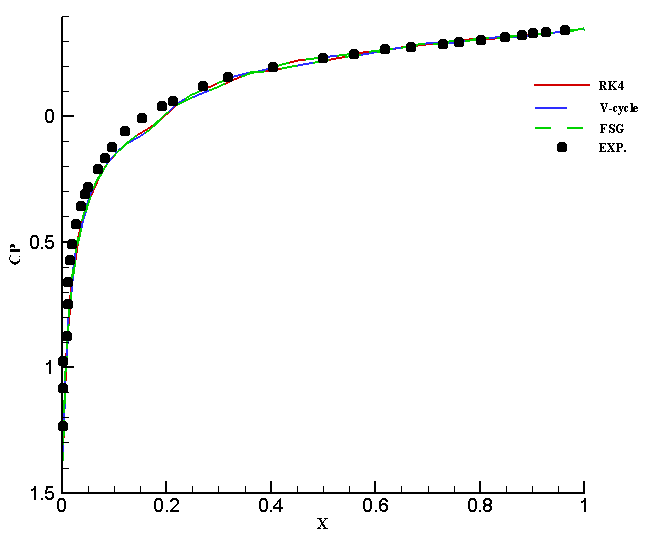
**شکل (42)** **نمودار همگرایی در** NACA0012 **با 3218 سلول برای**  **و**

**جدول (12)** **نتایج عددی مربوط به همگرایی** **در** NACA0012 **با 3218 سلول برای**  **و**

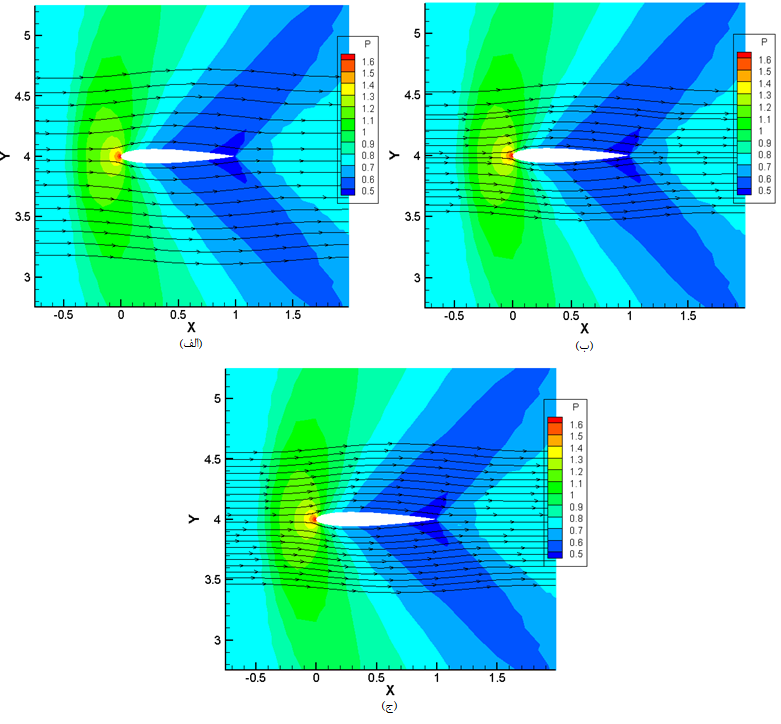
|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 5582 | 88.54 | RK4 |
| 3768 | 78.9 | V\_Cycle |
| 3189 | 66.44 | Full Single Grid |

* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش

شکل (43) نمودار ضریب فشار حاصله درNACA0012برایواز سه حلگر RK4، V-Cycle، FSG و داده های تجربی



* + 1. مقایسه کانتور فشار (P) برای هر سه روش

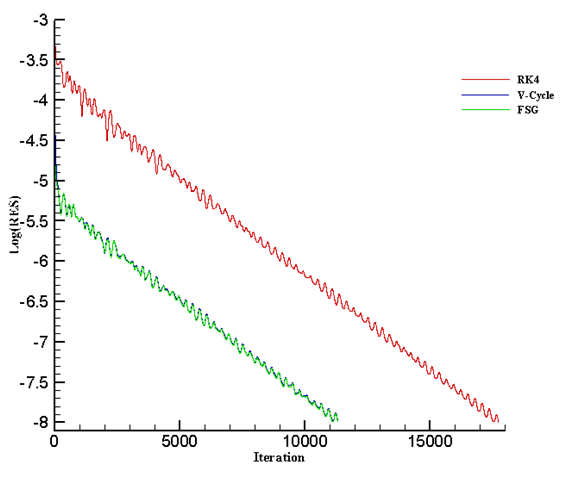


شکل (44) کانتور فشار حاصله درNACA0012برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG

آزمایش 4-1

در این آزمایش هندسه شماره 4 برای ماخ 0.3 و زاویه حمله صفر درجه برای سه حلگر RK4، V-Cycle و Full Single Grid (FSG) مورد بررسی قرار گرفته است.

* + 1. روند همگرایی بر اساس تعداد تکرارها

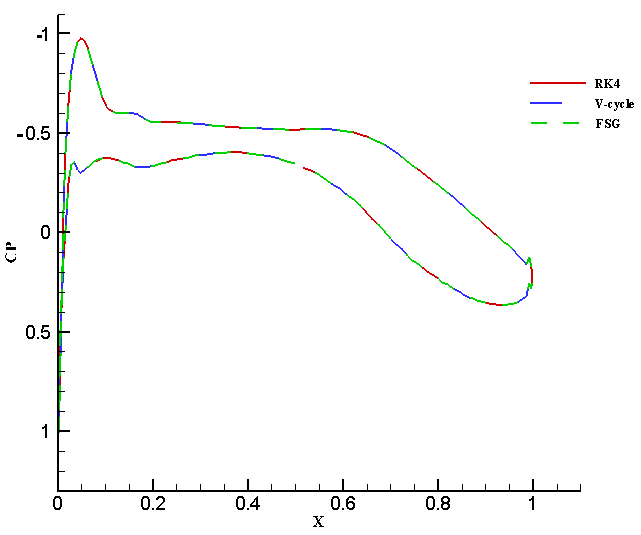


**شکل (45)** **نمودار همگرایی در** NLR7301 **با 7144 سلول برای**  **و**

**جدول (13)** **نتایج عددی مربوط به همگرایی** **در** NLR7301 **با 7144 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 17732 | 590.9 | RK4 |
| 11322 | 464.5 | V\_Cycle |
| 11315 | 438.9 | Full Single Grid |

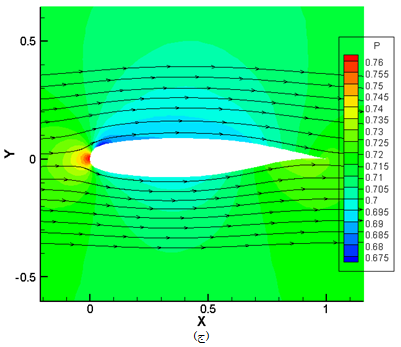
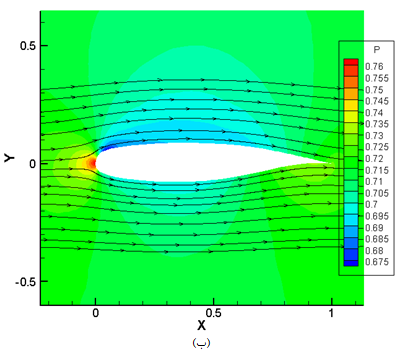
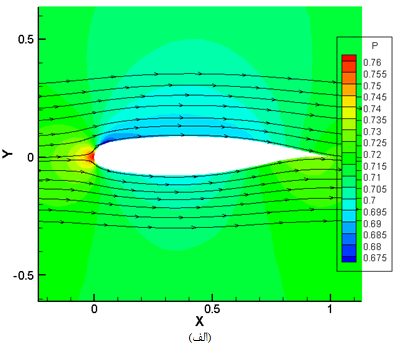
* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش



شکل (46) نمودار ضریب فشار حاصله درNLR7301برایواز سه حلگر RK4، V-Cycle و FSG

* + 1. مقایسه کانتور فشار (P) برای هر سه روش

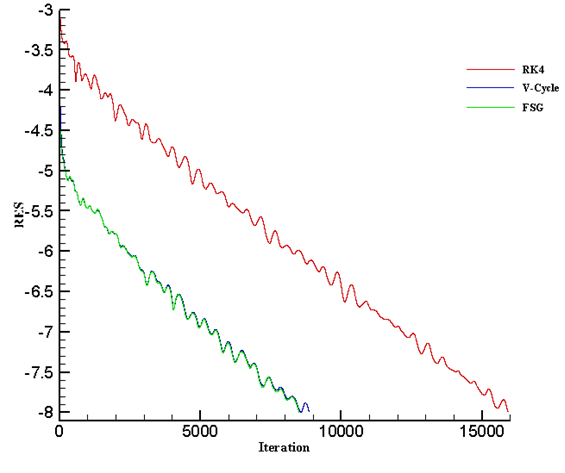
شکل (47) کانتور فشار حاصله درNLR7301برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG



آزمایش 4-2

آزمایش قبل برای ماخ 0.5 و زاویه حمله صفر درجه برای سه حلگر تکرار شده است.

* + 1. روند همگرایی بر اساس تعداد تکرارها



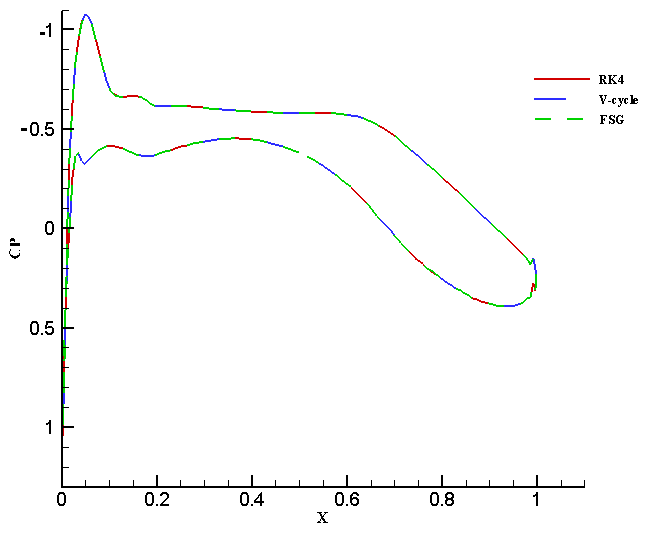
**شکل (48)** **نمودار همگرایی در** NLR7301 **با 7144 سلول برای**  **و**

**جدول (14)** **نتایج عددی مربوط به همگرایی** **در** NLR7301 **با 7144 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 15921 | 536.59 | RK4 |
| 8859 | 376.39 | V\_Cycle |
| 8515 | 346.78 | Full Single Grid |

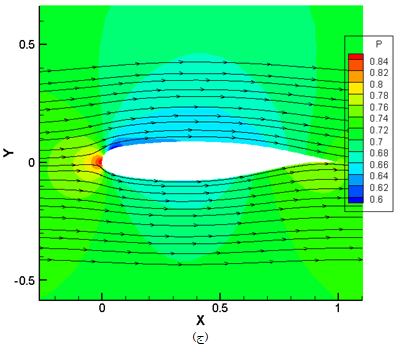
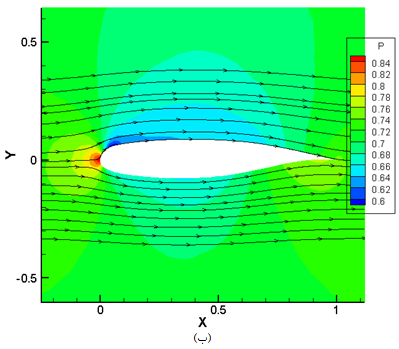
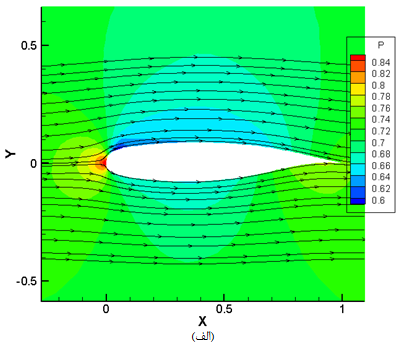
* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش

شکل (49) نمودار ضریب فشار حاصله درNLR7301برایواز سه حلگر RK4، V-Cycle و FSG



* + 1. مقایسه کانتور فشار (P) برای هر سه روش

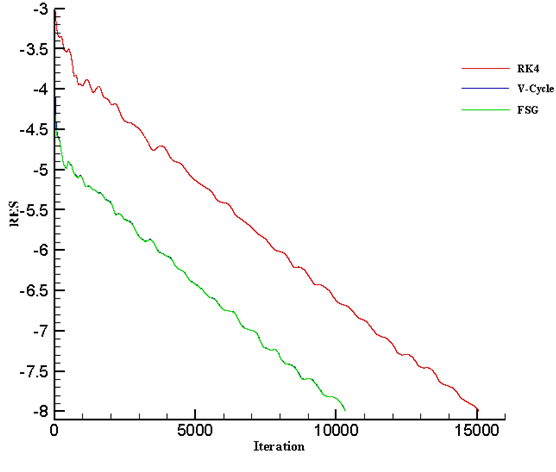
شکل (50) کانتور فشار حاصله درNLR7301برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG



آزمایش 4-3

آزمایش قبل را برای ماخ 0.65 و زاویه حمله صفر درجه تکرار کرده ایم.

* + 1. روند همگرایی بر اساس تعداد تکرارها



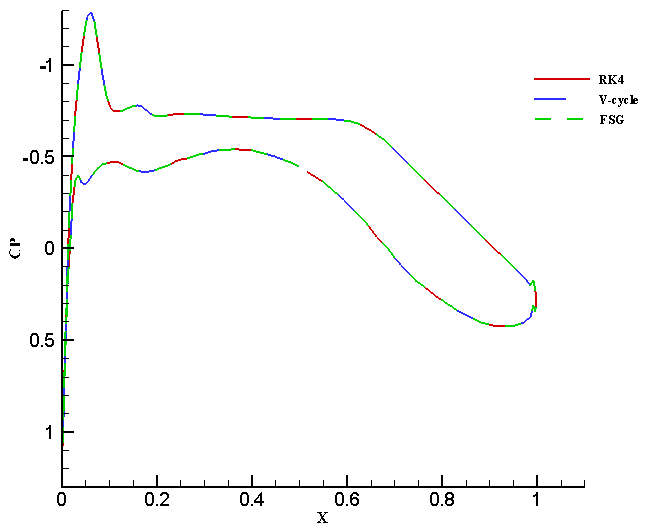
**شکل (51)** **نمودار همگرایی در** NLR7301 **با 7144 سلول برای**  **و**

**جدول (15)** **نتایج عددی مربوط به همگرایی** **در** NLR7301 **با 7144 سلول برای**  **و**

|  |  |  |
| --- | --- | --- |
| **تعداد تکرارهای همگرایی** | **زمان همگرایی (s)** | **حلگر مورد استفاده** |
| 15068 | 500.6 | RK4 |
| 10341 | 425.57 | V\_Cycle |
| 10334 | 409.8 | Full Single Grid |

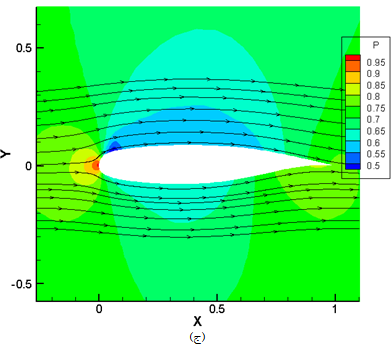
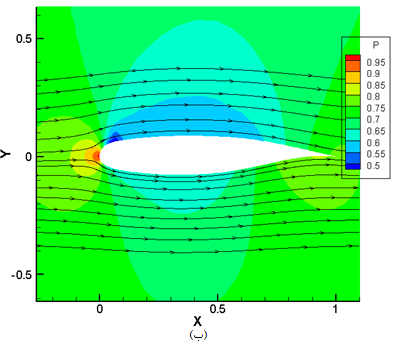
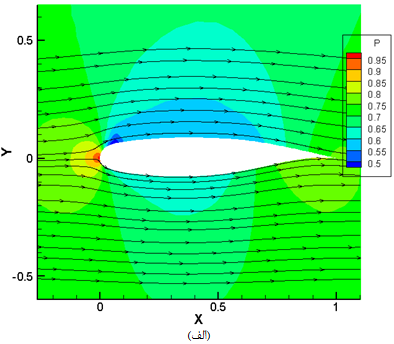
* + 1. مقایسه نمودار ضریب فشار (Cp) برای هر سه روش

شکل (52) نمودار ضریب فشار حاصله درNLR7301برایواز سه حلگر RK4، V-Cycle و FSG



* + 1. مقایسه کانتور فشار (P) برای هر سه روش

شکل (53) نمودار ضریب فشار حاصله درNLR7301برایواز سه حلگر الف) RK4 ب) V-Cycle ج) FSG



* 1. بحث و جمع بندی

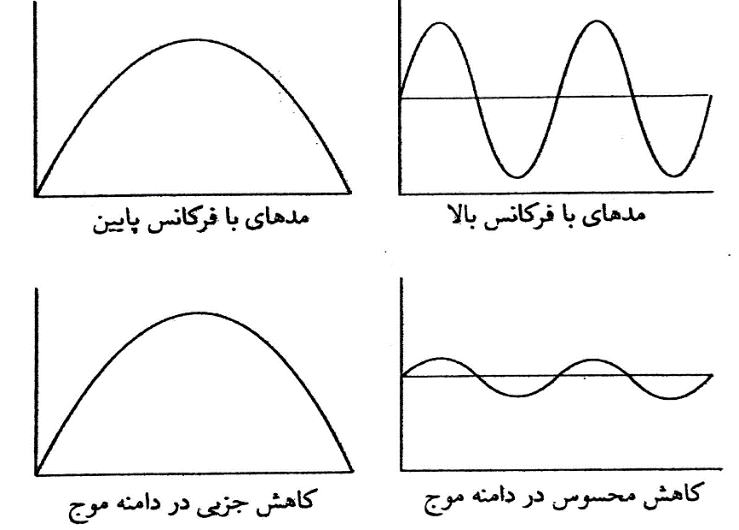
همانطور که از نتایج بدست آمده، مشاهده شد، دو روش مالتی گرید استفاده شده در این پروژه از همگرایی بهتری نسبت به روش رانج کوتا برخوردار است. دلیل این امر را می توان در تاثیر روش مالتی گرید در حذف طول موج هایی با فرکانس پایین در شبکه درشت یافت. از مقایسه نمودارهای ضریب فشار برای دیواره هندسه با نتایج تجربی و همچنین کانتورهای فشار با نتایج بدست آمده از رانج کوتای مرتبه 4، درستی روش های بکاربرده شده مورد بررسی قرار گرفت، که از دقت بالایی برخوردار بودند.

در مقایسه دو روش V-Cycle و FSG، همانطور که مشاهده شد، در برخی آزمایشات هر دو روش از همگرایی نزدیک به هم برخوردار بودند و در برخی نتایج متفاوت بود. آنچه که در این جا حائز اهمیت است آن است که روش V-Cycle در تمامی موارد نسبت به روش رانج کوتا از همگرایی بهتری برخوردار بود گرچه این همگرایی در برخی مواقع با روش FSG تقریبا برابری می کرد، ولی آزمایشاتی وجود داشت که روش FSG از نرخ همگرایی خوبی برخوردار نبود و با روش رانج کوتا تقریبا برابری کرده و حتی در یک آزمایش تعداد تکرارهای همگرایی بیشتری نسبت به رانج کوتا داشت. ازآنجایی‌که در مسائل غیرخطی فرکانس خطاها با طول شبکه تغییر می‌کند حرکت کردن در شبکه‌های متوالی برای به دست آوردن حدس اولیه برای شبکه ریز چندان سودمند نخواهد بود. در این‌گونه مسائل استفاده از انتقال خطا مشکل را حل می‌کند که اساس کار روش تقریب کامل می‌باشد نتیجه گیری که از این بحث می توان داشت این است روش FSG، گزینه مطمئنی برای استفاده به عنوان یکی از روش های مالتی گرید نمی باشد و روش V-Cycle تک شبکه ای گزینه خوبی برای جایگزینی آن است. در نهایت اینکه می توان نتایج حاصله از این روش را با تعداد مراحل درشت سازی بیشتر ارتقا داد که از سرعت همگرایی بالاتری نسبت به حالتی دارد که تنها از یک مرحله درشت سازی استفاده می کند.

1. راهنمای آموزشی
   1. مقدمه

برای حل عددی معادلات دیفرانسیل به وسیله کامپیوتر، روش های زیادی مورد استفاده قرار می گیرد که از جمله مهمترین های این روش ها، روش های تکراری می باشند. روش های تکراری مانند ژاکوبی[[1]](#footnote-1)، گاوس-سایدل[[2]](#footnote-2)، [[3]](#footnote-3)SOR و غیره بسیار پر کاربرد می باشند. از جمله مزایای این روش ها این است که می توان با چند تکرار به جواب های تقریبی مناسبی دست یافت هرچند حدس اولیه ای که برای شروع حل انتخاب می شود چندان مناسب نباشد. از جمله مسائلی که حل های تکراری با انها مواجه هستند، روند همگرایی و زمان محاسبات می باشد. این مشکل برای مسائلی که شبکه بندی به صورت فشرده و زیاد روی میدان صورت می گیرد بیشتر به چشم می خورد. اگرچه روش های ضمنی مانند [[4]](#footnote-4)ADI که در تسریع روند همگرایی موثر می باشند امروزه بیشتر مورد توجه برنامه نویسان قرار گرفته اند، هنوز این روش ها چندان پاسخگو نیستند. روش چند شبکه ای[[5]](#footnote-5) در طی سالیان گذشته به عنوان یک ابزار قوی به منظور تسریع همگرایی به کار رفته و کابردهای آن توسعه یافته است.

در شبکه ریز، خطاهایی که در تحلیل سری فوریه، مودهای فرکانس بالا محسوب می شوند و دارای طول موج های هم مرتبه با اندازه شبکه هستند به سرعت از بین می روند اما خطاهای فرکانس پایین که در شبکه ریز قابل رویت نیستند به سادگی و به سرعت از بین نمی روند (شکل 51) و باعث تخریب و کند شدن روند همگرایی می شوند.



**شکل (54) تغییر دامنه برای موج هایی با فرکانس پایین و بالا**

این خطاها که در شبکه ریز[[6]](#footnote-6) دارای طول موج بزرگ می باشند در شبکه بزرگتر دارای طول موج کوچک (نسبت به اندازه شبکه) می باشند و در شبکه درشت[[7]](#footnote-7) قابل مشاهده هستند و به سرعت از بین می روند. در روش چند شبکه ای با استفاده از انتقال حل بین شبکه های متوالی ریز و درشت این امکان وجود دارد که خطاهای موجود به سرعت کاهش یابند و با سرعت بیشتری به جواب های حقیقی همگرا شوند. اولین روش دو شبکه ای توسط فدرنکو[[8]](#footnote-8) [1] ارائه شد. همچنین فدرنکو [2] بود که در سال 1964 اولین الگوریتم چندشبکه ای را فرمول بندی کرد و رفتار واقعی آن را اثبات نمود. در سال 1966 باخوالوف[[9]](#footnote-9) [3] وضعیت های پیچیده تری را برای معادله بیضوی مرتبه دوم با ضرایب متغیر بررسی نمود. در سال 1972 برنت[[10]](#footnote-10) [4] کارایی کشف شده روش چند شبکه ای را به نمایش گذارد، تاکید او به ترکیب تکنیک چندشبکه ای با تکنیک های اضافی بود که به یک روش تطبیقی چندشبکه ای منجر گردید. شاید بتوان گفت موثرترین کار در زمینه کاربرد روش چندشبکه ای در مسائل بیضوی در سال 1977 توسط برنت [5] صورت گرفت که در ضمن آن استفاده از تحلیل مود موضعی را پیشنهاد کرد . در حالی که قضایای در دسترس در تئوری روش های چندشبکه ای عموما از فرض بیضوی بودن استفده می کنند. به نظر می رسد که روش چندشبکه ای قادر است با استفاده از گام های زمانی بزرگتر در شبکه های درشت تر، که در نتیجه آن اغتشاشات سریعتر میدان را ترک می کند پیمایش حل یک سیستم هذلولی را به سمت حالت دائمی تسریع کند. در سال 1979 جیمسون[[11]](#footnote-11) [6]همگرایی حل معادله پتانسیل کامل بقایی گذر صوت که ماهیتی مختلط از بیضوی و هذلولی دارد را به صورت موفقیت آمیزی به روش چندشبکه ای تسریع کرد که در آن نتایج همگرا شده در طی چند سیکل چندشبکه ای بدست آمدند.

چندین سال پیش برای شبکه های بی سازمان نشان داده شد که روش های چندشبکه ای یک مکانیزم حل کارآمد برای جریان های پایا چه در دو بعد و چه در سه بعد می باشد. وقتی که با شبکه های بی سازمان سرکار داریم، مشکل اصلی در ارتباط با الگوریتم چندشبکه ای در تولید شبکه های درشت می باشد. در حالی که برای شبکه های سازمان یافته، شبکه های درشت با حذف nامین نقطه در هر راستا، و سپس شبکه بندی گره های باقی مانده تشکیل می شود.

از جمله کارهای انجام شده در ارتباط با استفاده روش های چندشبکه روی شبکه های بی سازمان می توان به کارهای جیمسون و ماوریپلیس[[12]](#footnote-12) در [7] اشاره نمود. در این تحقیق از روش های چندشبکه ای برای حل معادلات اویلر استفاده شده است. گسسته سازی معادلات در مطالعه انجام شده به صورت حجم محدود و ضلع محور و شبکه مورد استفاده مثلثی بی سازمان می باشد. در این تحقیق هر دو نوع ذخیره داده ها یعنی راس محور[[13]](#footnote-13) و سلول محور[[14]](#footnote-14) مورد بررسی قرار داده شده است. برای انتقال اطلاعات به شبکه ریز و درشت در روش سلول محور از [[15]](#footnote-15)area weighting transfer استفاده کرده اند. در روش راس محور برای انتقال اطلاعات از شبکه ریز به شبکه درشت و بالعکس از درون یابی خطی استفاده شده است. در این تحقیق زمان همگرایی روش های تک شبکه و چندشبکه ای سلول محور و راس محور مقایسه شده است و نتایج به دست آمده نشان می دهد که روش چندشبکه ای نسبت به روش تک شبکه زمان همگرایی را بسیار تسریع می بخشد. همچنین مقایسه انجام شده بین روش های سلول محور و راس محور نشان می دهد که روش چندشبکه ای بر پایه سلول محور نسبت به راس محور زمان همگرایی کمتری دارد. از دیگر کارهای مشترک جیمسون و ماوریپلیس در [8] می باشد که از روش چندشبکه ای برای حل معادلات اویلر در دو بعد روی یک شبکه مثلثی منظم استفاده کرده اند. در تحقیق ابتدا شبکه ریز تولید شده و سپس شبکه های درشت تر برای بکار بردن در الگوریتم چندشبکه ای با استفاده از این شبکه ریز ایجاد می شوند. به دلیل منظم بودن مثلث ها، هر چهار مثلث شبکه ریز، یک مثلث شبکه درشت را تشکیل می دهند. در این مقاله نیز از همین شیوه جهت درشت سازی استفاده شده است. انتقال داده ها به این صورت می باشد که مقدار داده در مثلث درشت برابر با مجموع داده های چهار مثلث تشکیل دهنده این مثلث درشت، در نسبت مساحت های آنها به مثلث درشت می باشد. برای انتقال مانده هم جمع مانده 4 مثلث ریز تشکیل دهنده مثلث درشت، مانده آن را تعیین می کند. برای انتقال داده از شبکه درشت به شبکه ریز هم از درون یابی خطی استفاده شده است. این شیوه در دقسمت درون یابی اطلاعات به طور کامل توضیح داده شده است.از دیگر کارهای ماوریپلیس در روش چندشبکه ای استفاده از این روش برای تسریع همگرایی حل معادلات ناویر-استوکس در روی مش مثلثی بی سازمان می باشد[9]. در این زمینه می توان به تحقیق دیگر صورت گرفته توسط سمیر موزفریجا[[16]](#footnote-16) اشاره نمود[10]. در آن حل معادلات با استفاده از حجم محدود و روش های چندشبکه ای و نحوه ایجاد دنباله ای از شبکه ها و همچنین انتقال اطلاعات در این دنباله ها را مورد مطالعه قرار داده شده است. در این تحقیق حل معادلات با استفاده از روش چندشبکه ای با شبکه های بی سازمان مورد بحث و بررسی قرار گرفته است. نویسنده برای تولید دنباله ای از شبکه جهت استفاده در روش چندشبکه ای، استراتژی تقسیم بندی مجدد را در پیش گرفته و با استفاده از شبکه درشت اولیه، شبکه های ریزتر را با تقسیم مجدد اضلاع المان ها ایجاد می کند.

اریک دیک[[17]](#footnote-17) و رایمسلا[[18]](#footnote-18) در [13] روش چندشبکه ای راس محور را برای حل معادلات اویلر در حالت پایا بکار برده اند. در این مطالعه از دنباله ای از شبکه های بی سازمان جهت استفاده در الگوریتم روش چندشبکه ای استفاده شده است. دنباله ی شبکه های درشت از یک شبکه ریز و با استفاده از telescoping nodes تولید می شود یعنی تمامی گره های شبکه درشت در شبکه ریز قرار دارد. طبق آن چیزی که نویسنده در این مقاله می گوید، مرحله درشت سازی[[19]](#footnote-19) دو فاز دارد. ابتدا باید مجموعه نقاطی که باید در شبکه درشت ظاهر شوند از شبکه ریز انتخاب شود و سپس شبکه درشت تولید شود. پیچیدگی های عملگرهای انتقال داده ها از شبکه درشت به شبکه ریز و برعکس با توجه به اینکه از روش چندشبکه ای راس محور استفاده شده است و همچنین درشت سازی با استفاده از telescoping nodes انجام گرفته، کاهش می یابد. به این صورت انتقال داده های آن دسته از گره های شبکه درشت و ریز که از نظر فیزیکی دقیقا روی هم قرار دارند به وسیله تزریق مستقیم انجام می گیرد.

در تحقیق ما معادلات به صورت حجم محدود گسسته شده و شبکه های مورد استفاده بر حل معادلات جریان به صورت بی سازمان می باشند. به طور کلی الگوریتم های حل معادلات جریان به روش حجم محدود بر روی شبکه های بی سازمان به دو صورت سلول محور و راس محور می باشد. در دیدگاه سلول محور تمامی اطلاعات حل، مانند خواص جریان و خطاها و مانده ها در مرکز سلول ذخیره می شود و در مقابل برای دیدگاه راس محور این اطلاعات در راس های سلول ها (در گره های شبکه) ذخیره می شوند که در این تحقیق از دیدگاه از سلول محور استفاده می شود.

در زیر مولفه هایی که برای الگوریتم چند شبکه ای مورد نیاز می باشند برای دو دستگاه معادلات خطی و غیر خطی توضیح داده خواهند شد.

* 1. رویه های موجود در روش چند شبکه ای بر اساس نوع معادله
     1. دستگاه معادلات خطی

دو رویه مورد استفاده از روش های چند شبکه ای در مسائل خطی :

1. روش تک شبکه ای کامل[[20]](#footnote-20)
2. تصحیح شبکه درشت[[21]](#footnote-21)

**روش تک شبکه ای کامل**

ایده اصلی این روش این است که به جای حل روی یک شبکه ریز که مسلما نیاز به کار و زمان زیادی دارد، معادلات را روی یک شبکه درشت حل نموده و جواب های بدست آمده را با میانیابی[[22]](#footnote-22) به شبکه ریز منتقل می کنیم. جواب های به دست آمده در شبکه ریز به جواب های دقیق نزدیک تر هستند و بنابراین در شبکه ریز سریع تر به جواب دقیق می رسیم.

در واقع با این روش ما تنها کاری که می کنیم این است که حدس های اولیه مناسبی برای حل به دست می آوریم که این به نوبه خود روند همگرایی را تسریع می کند. اگرچه این روش کارامد می باشد ولی دارای معایبی نیز خواهد بود. از آنجایی که در مسائل غیرخطی فرکانس خطاها با طول شبکه تغییر می کنند، حرکت کردن در شبکه ها چندان هم سودمند نخواهد بود. در این گونه مسائل استفاده از مانده[[23]](#footnote-23) این مشکل را حل می کند که اساس روش چندشبکه ای می باشد.

الگوریتم حل روش تک شبکه ای کامل:

فرض کنید که می خواهیم معادله دیفرانسیل زیر را حل کنیم :

|  |  |
| --- | --- |
| **(1)** |  |

در این رابطه جواب دقیق معادله می باشد. جواب تقریبی این معادله با نمایش می دهیم. همچنین فرض کنید دنباله ای از شبکه ها به صورت داشته باشیم که در آن درشت ترین شبکه و ریزترین شبکه می باشد. و به ترتیب جواب دقیق و جواب تقریبی در این شبکه ها می باشد.

الگوریتم :

1. حل روی شبکه تا به خطای مناسبی همگرا شود.
2. انتقال حل به دست آمده به شبکه ریز بعدی با استفاده از میانیابی.
3. حل روی شبکه تا به خطای مناسبی همگرا شود.
4. انتقال حل به دست آمده به شبکه ریز بعدی با استفاده از میانیابی.
5. ادامه روند فوق
6. حل روی شبکه تا به خطای مورد نظر همگرا شود که در این صورت به جواب رسیده ایم.

**تصحیح شبکه درشت**

در این رویه بر عکس روند قبلی به جای حل خود معادله در شبکه های متوالی با استفاده از معادله مانده مقدار خطا را در شبکه های متوالی به دست می آوریم و به جواب اصلی اضافه می کنیم.

فرض کنید که دستگاه معادلات خطی ما به صورت رابطه 1 باشد. همچنین فرض کنید که حل دقیق مورد نظر در ریزترین شبکه و حل تقریبی در شبکه kام باشد. بنابراین حل تقریبی در شبکه ریز را با نمایش می دهیم. جواب تقریبی در هر مرحله با استفاده از روش های تکراری مانند گاوس-سایدل و یا SOR با 2 یا 3 تکرار بدست می آید. همانطور که مشخص است، میزان خطای حل تقریبی و حل دقیق مطابق با رابطه زیر می باشد :

|  |  |
| --- | --- |
| **(2)** |  |

با ضرب این معادله در عمگر داریم :

|  |  |
| --- | --- |
| **(3)** |  |

و با قرار دادن به جای خواهیم داشت :

|  |  |
| --- | --- |
| **(4)** |  |

سمت راست معادله فوق را مانده می نامیم. با حل کردن دستگاه معادلات فوق می توان مقدار خطای را به دست آورد و با جاگذاری در معادله خواهیم داشت :

|  |  |
| --- | --- |
| **(5)** |  |

بدین گونه می توان با حساب کردن خطا و افزودن آن به حل تقریبی به یک جواب بهبود یافته برای حل دست یافت.

چرخه های روش چند شبکه ای در تصحیح شبکه درشت :

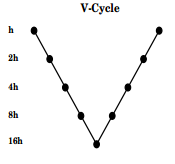
در محاسبات CFD روند انتقال شبکه ها پیچیده و مشکل تر می باشد و چرخه های مختلفی برای درشت کردن و ریز کردن مش با برنامه های خاصی برای محدود کردن و گسترش دادن استفاده می شود. از جمله چرخه های روش چندشبکه ای عبارت است از : چرخه V[[24]](#footnote-24)، چرخه W[[25]](#footnote-25)، چرخه F[[26]](#footnote-26)، چرخه FMG[[27]](#footnote-27)

**چرخه** V **:**

ساده ترین چرخه در این روش است که شامل دو پایه است. محاسبات از ریزترین شبکه آغاز می شود. بعد از تعداد کمی تکرار در مرحله ریزترین شبکه، باقی مانده حاصل شده به یک مرحله شبکه درشت تر انتقال داده می شود و این روند ادامه می یابد تا به درشت ترین شبکه برسیم. بعد از آخرین تکرارها در درشت ترین شبکه، مرحله به مرحله به عقب برگشته و دوباره مقادیر گسترش یافته تا به ریزترین شبکه برسیم. این چرخه در شکل 52 نشان داده شده است.

الگوریتم سیکل V:

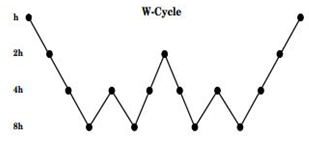
1. ابتدا در ریزترین شبکه تکرار روی خود معادله با استفاده از روش گاوس-سایدل یا SOR و حدس اولیه صورت می گیرد تا بدست آید. در عمل را برابر با 2 یا 3 ویا .... قرار می دهند.(smoothing)
2. مقدار مانده را بدست می آوریم: .
3. با استفاده از عملگرهای محدود سازی[[28]](#footnote-28) مانده را به شبکه درشت تر منتقل می کنیم .
4. با استفاده از رابطه مقدار خطا را با حدس اولیه e=0 با استفاده از روش های تکراری بدست می آوریم.(در روش چند شبکه ای جبری ماتریس به صورت بدست می آید ( در این روش عملگر های محدود سازی و گسترش سازی[[29]](#footnote-29) معمولا ترانهاده هم می باشند).
5. مانده را در این شبکه بدست می آوریم : .
6. حال این مانده را به شبکه درشت بعدی منتقل می کنیم :
7. مقدار خطا را با رابطه و با تکرار با روش های تکراری محاسبه می کنیم..
8. این روند را ادامه می دهیم تا به درشت ترین شبکه برسیم.
9. در درشت ترین شبکه را با استفاده از رابطه بدست می آوریم.
10. با استفاده از عملگرهای گسترش سازی مقدار را به شبکه انتفال داده و خطای این شبکه را اصلاح می کنیم :
11. رابطه را با حدس اولیه با روش های تکراری و تکرار حل می کنیم تا جدید به دست آید.
12. مراحل 7 و 8 را تا رسیدن به ریزترین شبکه ادامه می دهیم.
13. پس از تصحیح خطای شبکه ، یعنی ، آن را به ریزترین شبکه انتقال می دهیم و با مقادیر میدان جمع می زنیم تا مقادیر بهبود یافته میدان در ریزترین شبکه بدست آید :
14. در انتها با استفاده از روش های تکراری و با استفاده از حدس اولیه معادله را حل می کنیم تا مقدارهای دقیق عددی میدان بدست آید.



**شکل (55) سیکل V به صورت شماتیک**

**چرخه** W **:**

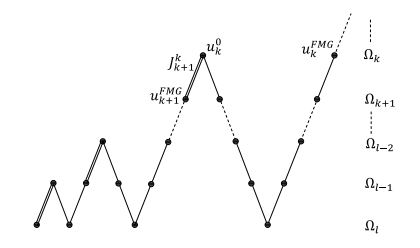
در این چرخه روند کار همانند چرخه V می باشد با این تفاوت که حرکت در شبکه های درشت تر در این چرخه بیشتر صورت می گیرد که این امر باعث می شود تا خطای محاسبه شده در چرخه W به خطای واقعی نزدیک تر باشد و در نتیجه روند همگرایی از چرخه V سریعتر می شود. حرکت در چرخه W به صورت زیر می باشد :



**شکل (56) سیکل W به صورت شماتیک**

**چرخه** FMG **:**

همانطور که در[14] آمده است، الگوریتم FMG ترکیب یک روش تکراری با چرخه های روش چندشبکه ای می باشد، که تقریب شبکه های درشت تر به عنوان حدس اولیه در شبکه ریز چرخه های روش چندشبکه ای به عنوان شروع این چرخه ها می باشد. به طور واضح تر ، این الگورتیم از درشت ترین شبکه یعنی شروع می کند و معادلات حاکم را روی این شبکه حل نموده تا جواب دقیق بدست آید. حالا این جواب با روش های مناسب درون یابی (گسترش سازی) به شبکه ریز بعدی یعنی منتقل می شود. حال این مقادیر بدست آمده به عنوان حدس اولیه یکی از چرخه های روش چندشبکه ای استفاده می شود تا تقریب روی شبکه بدست آید. دوباره این حل به شبکه ریز بعدی منتقل می شود و بار دیگر از همین چرخه روش چندشبکه ای استفاده می شود و روند به همین صورت ادامه داده می شود. در ریزترین شبکه بار دیگر همین چرخه بکار برده می شود تا بدست آید. در شکل زیر الگوریتم چرخه FMG برای چرخه v نمایش داده شده است.

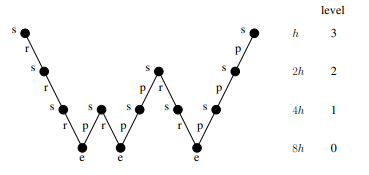


**شکل (57) سیکل FMG به صورت شماتیک**

در این روش محاسبات از درشت ترین شبکه آغاز می شود و پاسخ های حاصل به مش های ریزتر منتقل می شود تا به ریزترین شبکه برسیم. جواب بدست آمده به عنوان نقطه شروع یکی از چرخه های فوق می باشد.

**چرخه** F **:**

همانطور که در شکل 55 نشان داده شده است چرخه F از ریزترین شبکه شروع می شود تا به درشت ترین شبکه برسد پس از آن، در مسیر حرکت از درشت ترین شبکه به ریزترین شبکه بعد از رسیدن به هر مرحله از شبکه ها بار دیگر عمل محدود سازی را انجام می دهد تا به درشت ترین شبکه برسد این عمل تا زمانی که محدود سازی در هر مرحله از شبکه ها به درشت ترین شبکه صورت گیرد ادامه می یابد تا اینکه در نهایت به ریزترین شبکه برسیم.



**شکل (58) سیکل F به صورت شماتیک**

* + 1. دستگاه معادلات غیرخطی

روش Full Aproximation Scheme

در مسائل غیر خطی حل به روش چندشبکه ای تا اندازه ای با مسائل خطی تفاوت دارد. اگرچه مفاهیم کلی به همان صورت باقی می مانند. تفاوت اصلی حل مسائل غیرخطی و خطی در نحوه ی محاسبه مانده و خطا می باشد. در این نوع مسائل دیگر نمی توان از روش coarse grid correction استفاده کرد و برای حل باید از روش Full Aproximation Scheme یا FAS استفاده کرد که به صورت زیر می باشد.

فرض کنید معادله مورد نظر به صورت زیر باشد :

|  |  |
| --- | --- |
| **(6)** |  |

که این بار A یک عملگر غیر خطی است و حل دقیق می باشد. با در نظر گرفتن حل تقریبی و مقدار خطای در این شبکه، می توان حل دقیق را به صورت بیان نمود. با توجه به معادله بالا داریم :

|  |  |
| --- | --- |
| **(7)** |  |

با کم کردن مقدار از دو طرف معادله فوق می توان نوشت :

|  |  |
| --- | --- |
| **(8)** |  |

در مسائل خطی می توان سمت چپ معادله فوق را ساده نمود و به معادله خطا که قبلا به دست آورده بودیم رسید. ولی در مسائل غیرخطی، به علت غیرخطی بودن عملگر A نمی توان سمت چپ را ساده نمود.

همانطور که از معادله فوق مشخص است می توان با تغییر متغیر به معادله زیر دست یافت :

|  |  |
| --- | --- |
| **(9)** |  |

با حل معادله فوق می توان مقدار را در شبکه درشت به دست آورد و برای به دست آوردن مقدار خطا باید مقدار را از آن کم کرد.

|  |  |
| --- | --- |
| **(10)** |  |

حال با انتقال خطا به شبکه ریز می توان حل در آن شبکه را تصحیح کرد. همانطور که در [15] آورده شده است الگوریتم V\_cycle برای مسائل غیر خطی به صورت زیر می باشد.

**FAS V-cycle Algorithm:**

1. ابتدا باید معادله را به فرم بنویسیم. مقدار دقیق حل این معادله را در هر شبکه با و مقدار تقریبی را با نشان می دهیم. دنباله شبکه ها به صورت زیر می باشند :

که ریزترین شبکه است.

1. یک حدس اولیه برای حل معادله فوق در ریزترین شبکه در نظر می گیریم و سپس با استفاده از این حدس اولیه و یکی از روش های تکراری مانند گاوس-سایدل یا ژاکوبی بار تکرار روی معادله انجام می دهیم که مقدار می تواند 1یا 2 یا 3 باشد.(smoothing). با این کار برای ریزترین شبکه مقدار تقریبی را بدست می آوریم.
2. با استفاده از این مقدار مانده را محاسبه می کنیم :
3. مقدار مانده به دست آمده را با استفاده از عملگرهای محدود سازی مخصوص انتقال مانده به شبکه درشت منتقل می کنیم: .
4. مقدار را با استفاده از عملگرهای محدود سازی مخصوص میدان به شبکه درشت منتقل می کنیم : .
5. همانطور که در بالا گفته شد مقدار در نظر گرفتیم و به این نتیجه رسیدیم که می باشد. با علم به این روابط مقدار را محاسبه می کنیم.
6. با مقدار بدست آمده معادله را با حدس اولیه با استفاده از روش های تکراری گاوس-سایدل یا ژاکوبی بار حل می کنیم. که با این کار خطاهای ناشی از انتقال هموار شده و در نتیجه جدیدی را بدست می آوریم.
7. بار دیگر باید با استفاده از این مانده را حساب کنیم: .
8. این مراحل را اینقدر ادامه می دهیم تا به درشت ترین شبکه برسیم.
9. در درشت ترین شبکه مقدار را با استفاده از و که با محدود سازی و به درشت ترین شبکه بدست آمده اند را با استفاده از رابطه محاسبه می کنیم.
10. با استفاده از مقدار معادله با روش های تکراری حل می کنیم. با حل این معادله مقدار جدید را بدست می آوریم.
11. حال باید مقدار خطا در درشت ترین شبکه را با استفاده از رابطه زیر بدست آوریم :
12. با محاسبه خطا، آن را به شبکه ریزتر یعنی منتقل می کنیم. سپس این خطای منتقل شده را به مقادیر میدان در این شبکه اضافه می کنیم تا میدان جدید حاصل شود. .
13. معادله با حدس اولیه اصلاح شده با مرتبه تکرار حل می کنیم. تا مقدار دقیق عددی بدست آید سپس بار دیگر خطا را بدست می آوریم : .
14. این مراحل را ادامه می دهیم تا به ریزترین شبکه برسیم.
15. در ریزترین شبکه مقدار را با خطای شبکه تصحیح می کنیم : .
16. با استفاده از این مقدار اصلاح شده به عنوان حدس اولیه معادله را با روش های تکراری و بار تکرار حل می کنیم تا جواب دقیق در ریزترین شبکه حاصل شود.

**FAS W-cylce Algorithm:**

این روش همانند روش توضیح داده شده در دستگاه معادلات خطی می باشد با این تفاوت که باید از روند گفته شده در دستگاه معادلات غیر خطی استفاده کنیم.

**FAS-FMG Algorithm:**

این روش همانند روشی می باشد که برای مسائل خطی توضیح داده شد با این تفاوت که باید بجای الگوریتم v-cycle خطی از غیر خطی استفاده شود.[16]

* + 1. به کار بردن روش های مالتی گرید در حلگرهای چند مرحله ای (Multilevel Scheme)

در برخی مسائل به جای روش های تکراری ژاکوبی و گاوس سایدل و... از روش های چند مرحله ای جهت حل استفاده می شود که از دقت بالاتری نسبت به روش های ذکر شده برخوردارند. از جمله این روش های چند مرحله ای به روش رانج کوتا با مرتبه های مختلف می توان اشاره نمود. فرم کلی روش های چند مرحله ای به صورت زیر می باشد :

|  |  |
| --- | --- |
| **(11)** |  |

در رابطه بالا *Dt* و *Ai* بترتیب مقدار گام زمانی و مساحت سلول محاسباتی می باشد و بالانويس *n* نشان دهندة مرحلة زماني و بالانويس *m* نشان دهندة مرحلة رانگ-كوتا مي‌باشد. مقدار استاندارد ضرايب  تا *am* از رابطة زير قابل‌محاسبه مي‌باشد:

|  |  |
| --- | --- |
| **(12)** |  |

برای بکار بردن روش های مالتی گرید در روش های چند مرحله ای به صورت زیر عمل می کنیم [17]. برای مثال مالتی گرید V-Cycle توضیح داده خواهد شد. در الگوریتم زیر اندیس h به منزله ریزترین شبکه و به همین ترتیب شبکه های درشت تر به صورت 2h و 4h و ... نشان داده شده است.

1. در شبکه ریز با یک حدس اولیه معادلات فوق را تا 2 یا 3 تکرار حل می کنیم. ()
2. مقادیر مانده را در این شبکه محاسبه می کنیم.
3. مانده ها و مقادیر میدان را به شبکه درشت منتقل می کنیم. ) و (
4. با استفاده از مقادیر میدان انتقالی، مانده ها را در این شبکه بدست می آوریم.
5. با استفاده از مانده های انتقالی از شبکه ریز و مانده های بدست آمده با مقادیر میدان انتقالی ، تابع تاثیر[[30]](#footnote-30) را به صورت زیر تعریف می کنیم :

|  |  |
| --- | --- |
| **(13)** |  |

1. اگر شبکه حاضر درشت ترین شبکه باید به مرحله 7 می رویم در غیر این صورت مراحل 1 تا 5 را آنقدر تکرار می کنیم تا به درشت ترین شبکه برسیم.
2. در درشت ترین شبکه معادلات به صورت زیر بازنویسی شده وجواب دقیق آن ها را در این شبکه بدست می آوریم :

|  |  |
| --- | --- |
| **(14)** |  |

1. بعد حل دقیق معادلات فوق همانند الگوریتم های قبل خطا را در این شبکه بدست آورده و به شبکه ما قبل جهت تصحیح مقادیر میدان منتقل می کنیم. این عمل را تا رسیدن به ریزترین شبکه ادامه می دهیم.
2. در نهایت در ریزترین شبکه معادلات را با حدس اولیه تصحیح شده تا رسیدن به جواب با دقت از قبل تعیین شده حل می کنیم.
   1. دسته بندی روش های چندشبکه ای
      1. چند شبکه ای هندسی[[31]](#footnote-31)

در روش چند شبکه ای هندسی، تراکم گره ها، سلول ها، حجم کنترل ها و المان ها در سطح هندسی رخ می دهد. مجموعه ای از داده ها و شبکه بندی جدید در هر مرحله شبکه بندی درشت تر به کار می روند. در روش چند شبکه ای هندسی یک دنباله مش تولید می شود و در هر مرحله معادلات گسسته سازی می شوند. مزیت این روش نسبت به روش جبری این است که در مسائل غیرخطی، عوامل غیرخطی در هر مرحله گسسته سازی به شبکه درشت تر هم منتقل شده و به خوبی دیده می شوند.

* + 1. چندشبکه ای جبری[[32]](#footnote-32)

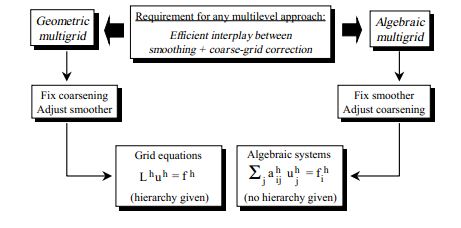
در این روش ایجاد شبکه های درشت تر تنها با استفاده از ماتریس ضرایب انجام می شود. در روش چندشبکه ای جبری ضرایب خاصی از ماتریس ضرایب با هم ترکیب می شوند تا ماتریس ضرایب شبکه درشت تر ایجاد شود و نیازی به گسسته سازی روی شبکه درشت تر نیست. روش های ترکیب این ضرایب با توجه به فیزیک مسأله و الگوریتم حل می تواند متفاوت باشد. مزیت این روش این است که نیاز به ساخت و ذخیره سازی هیچ شبکه درشتی نیست و همچنین نیازی به محاسبه شارها و ترم منابع در شبکه های درشت تر نیست. این روش برای شبکه های بی سازمان بسیار مناسب است. عیب این روش این است که در مسائل غیرخطی، خاصیت غیرخطی بودن در شبکه های درشت تر احساس نمی شود.

*در روزهای اولیه پیدایش روش چندشبکه ای، نظریه تصحیح شبکه درشت بر پایه استراتژی های درشت سازی ساده (معمولا با دو برابر کردن اندازه شبکه در هر راستا) و عملگرهای انتقال اطلاعات هندسی ساده و همچنین عملگرهای شبکه درشت مشابه عملگرهای ریزترین شبکه داده شده بود. بعدها فهمیدند که مولفه های ساده شبکه درشت برای نمونه های متفاوتی از مسئله های پیچیده تر همانند معادله های دیفیوژن مناسب نمی باشد. از این رو عملگر گالرکین به عنوان جایگزین عملگرهای شبکه درشت معرفی شد. از نگاه عملی از برتری های این عملگر این است که می تواند کاملا به صورت جبری ساخته شود و این باعث می شود که کار با آن بسیار راحت باشد. معرفی عملگر درون یابی وابسته (درون یابی که به طور مستقیم به گسسته سازی استنسیل ها وابسته است) خیلی اهمیت داشت. همراه با عملگر گالرکین این نوع درون یابی اجازه رفتار با دسته بزرگی از معادلات شامل مسائلی همراه با ضرایب به شدت ناپیوسته را می داد. معضل اصلی در ارتباط با چنین مسائلی بعد از بکار بردن پروسه* smoothing *می بود، که خیلی وقت ها خطا به طور هندسی هموار نمی شد. خطای هموار شده همان ناپیوستگی هایی حل مسئله را از خود نشان می داد.*

*همه نظریه های چندشبکه ای هندسی بر روی دنباله ای از شبکه های از پیش تعریف شده عمل می کنند. این به این معنی است که پروسه درشت سازی تعیین شده و تا جایی که ممکن است ساده سازی می شود. اگرچه تعیین دنباله ها نیازهای ویژه ای را روی هموارکننده ها بوجود می آورد تا کارکرد کارامدی بین تصحیح شبکه درشت و هموارکننده ها تضمین کند. در کل مولفه هایی از خطا که نمی توانند در شبکه درشت تصحیح شوند می بایست به طور موثری توسط هموارکننده کاهش یابند(همینطور برعکس).*

*به عنوان مثال فرض کنید شبکه های درشت تر به صورت درشت سازی بدست آمده باشند،* pointwise relaxation *برای مسائل ایزوتروپیک بسیار کارآمد می باشند. برای مسائل غیرایزوتروپیک* pointwise relaxation *خواص هموارکنندگی خوبی فقط در راستای خاصی را می دهد. نتیجتا هموارکننده های پیچیده تری مانند* line-relaxation *و* ILU *نیاز می باشد تا سرعت همگرایی روش چندشبکه ای را حفظ کند. نظریه های چندشبکه ای که برای گروه بزرگی از مسائل اثر متقابلی بین هموارکننده و تصحیح شبکه درشت اعمال می کند تا به صورت کارآمد عمل کنند، اغلب نیرومند خوانده می شوند.*

*در حالی که بکارگیری موثر و نیرومند هموارکننده ها در دو بعد سخت نبود ولی در سه بعد و برای شبکه های پیچیده کاری پرزحمت می بود. راجع به اثر متقابل بین هموارکننده و تصحیح شبکه درشت، روش* AMG *می تواند به عنوان روشی با حفظ هموارکننده های ساده و همگرایی نیرومند توجه شود. پیشرفت* AMG *در اوایل دهه 80 زمانی که پروسه تصحیح شبکه درشت برپایه گالرکین و به ویژه عملگر وابسته درون یابی در روش چندشبکه ای هندسی معرفی شدند، شروع شد. یکی از انگیزه های استفاده از* AMG *این بود که عملگر وابسته درون یابی و عملگر گالرکین می توانست بدون هیچ رجوعی به شبکه و با استفاده از ماتریس های اساسی مستقیما بدست آورده شود. این تفاوت های مفهومی اساسی را بین روش چندشبکه ای هندسی و جبری ایجاد نمود. نظریه چندشبکه ای هندسی دنباله ای از شبکه های تعیین شده را بکار گرفته و سپس کارکرد موثری را بین هموارکننده و تصحیح شبکه درشت با انتخاب مناسب پروسه هموار کننده تضمین می کند. در مقابل آن، روش چندشبکه ای جبری هموارکننده ساده ای را مانند روش گاوس-سایدل تعیین کرده سپس یک کارکرد موثری بین تصحیح شبکه درشت با انتخاب مناسب درون یابی و شبکه های درشت تر اجرا می کند. به طور هندسی روش جبری تلاش می کند تا درشت سازی را فقط در جهتی انجام دهد که خطاها واقعا هموار می شوند. در این روش پروسه درشت سازی به صورت کاملا خودکار انجام می شود. این ویژگی یزرگترین دلیل برای انعطاف پذیری روش جیری در تطبیق دادن خود با نیازهای اصلی مسئله ای که حل می شود می باشد و دلیل اصلی برای نیرومندی آن در حل دسته بزرگی از معادلات با وجود هموار کننده* point-wise *ساده می باشد.انعطاف پذیری و سادگی استفاده از این روش یک هزینه دارد و آن هم اینست که یک مرحله ستاپ یعنی تشکیل شبکه های درشت و همه عملگرها باید تولید شوند تا مرحله حل مسئله بتواند شروع شود. این هزینه یک دلیلی می باشد برای اینکه روش* AMG *نسبت به* GMG *از کارایی کمتری برخوردار است( البته این برای مسائلی می باشد که روش هندسی را بتوان به صورت موثر به کار برد). دلیل دیگر این است که نمی توان انتظار بهینه شدن مولفه های روش جبری را داشت، آن ها همیشه بر پایه سازش بین کار عددی و بازده کلی ساخته می شوند. یا این وجود اگر این روش برای مسائل بیضی گون استاندارد بکار برده شود مقدار محاسبات آن با مقدار محاسبات انجام گرفته برای روش هندسی نیرومند قابل مقایسه می باشد. اگرچه به روش* AMG *نباید به عنوان یک رقیب با روش هندسی توجه شود. نقطه قوت روش* AMG *نیرومندی آن بوده و قابل کاربرد برای هندسه های پیچیده و حتی برای مسائلی می باشد که حل آن توسط روش هندسی دور از دست می باشد.* [21]



**شکل (59) مقایسه مراحل دو روش هندسی و جبری**

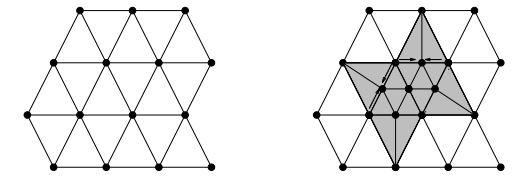
تولید دنباله ای از شبکه ها جهت استفاده در الگوریتم چندشبکه ای هندسی [9]

1. **تولید شبکه های ریز با استفاده از شبکه درشت اولیه**

یکی از ساده ترین استراتژی ها برای روش چندشبکه ای با شبکه بی سازمان تولید دنباله ای از شبکه های ریز با استفاده از شبکه درشت اولیه با تقسیم سلول های شبکه می باشد. این استراتژی در مقایسه با روش های دیگر چندشبکه ای این تفاوت را دارد که سایر روش ها با ریزترین شبکه شروع می شود و شبکه های درشت تر از این شبکه ریز ساخته می شود.

این استراتژی چند عیب دارد. یکی از بدهی ترین عیب های آن، انعطاف پذیری کم آن می باشد. در واقع این روش نیازمند یک اتصال محکمی بین تولید شبکه و روش چندشبکه ای می باشد. از دیگر عیب های آن مربوط به توانایی درشت ترین شبکه در تولید خواص همگرایی کارآمد برای الگوریتم چندشبکه ای و کیفیت شبکه ریز می باشد، به این معنی که ممکن است شبکه درشت قادر به تولید شبکه ریز با کیفیت نباشد و در نتیجه بر روی دقت حل تاثیر بگذارد. البته همانطور که در [1] به آن اشاره شده، این تکنیک در صورتی که با بهبود سازی قابل تطبیق(adaptive refinement) ترکیب شود، روش قدرتمندی خواهد بود.

از مزایای این روش همانطور که در شکل نشان داده شده است تودرتو[[33]](#footnote-33) بودن دنباله شبکه های تولیدی می باشد که سبب می شود انتقال داده ها بین شبکه ها با عملگرهای ساده تری صورت پذیرد. برای مثال در استراتژی راس محور، راس هایی که دقیقا روی هم قرار می گیرند انتقال اطلاعات با روش بسیار ساده تزریق مستقیم (بعدا توضیح داده می شود) صورت می گیرد و از آنجا که شبکه های ریز با نصف شدن اضلاع شبکه های درشت تولید می شود، راس هایی که در وسط اضلاع شبکه درشت قرار دارند انتقال اطلاعات به وسیله درون یابی خطی یعنی میانگین دو راس ضلع ضبکه درشت انجام می گیرد. این استراتژی در شکل زیر نشان داده شده است :



**شکل (60) استراتژی تولید شبکه های ریز از شبکه درشت**

1. **تولید شبکه های درشت مستقل با استفاده شبکه ریز اولیه**

یک روش جایگزین برای روش چندشبکه ای بی سازمان تولید دنباله ای از شبکه های ریز و درشت کاملا مستقل می باشد. شبکه ها ممکن است با هر روش تولید شبکه ، ساخته شده و در حالت کلی تودرتو نباشند. تنها چیزی که در این روش لازم است، تطبیق شبکه به همان مرزهای دامنه اولیه می باشد. این استراتژی نسبت به استراتژی قبلی بسیار انعطاف پذیرتر می باشد، چون شبکه های ریز و درشت مقید نمی باشند و می تواند به طور مستقل برای دقت حل و سرعت همگرایی بهینه شوند.

در این تحقیق ما شیوه دوم یعنی درشت سازی شبکه ریز را در الگوریتم چند شبکه ای مورد استفاده قرار می دهیم. انواع روش های درشت سازی در زیر آورده شده است.

* 1. روش های درشت سازی و درون یابی در روش مالتی گرید
     1. روش های درشت سازی در مالتی گرید هندسی

مراحل مختلف در روش چندشبکه ای هندسی با درشت سازی شبکه ریز اولیه و تولید دنباله ای از شبکه های درشت انجام می شود. شبکه مورد نظر را به عنوان یک گراف در نظر می گیریم. روند کلی انتخاب یک زیر مجموعه از گره های شبکه ریز و تولید شبکه بعدی توسط این گره ها می باشد. کارایی روش چندشبکه ای هندسی شدیدا به انتخاب این گره ها بستگی دارد. این گره ها می بایست تقریبی از توپولوژی شبکه ریزتر باشند. گره ها به منظور درشت سازی با هدف کاهش سایز شبکه و با حفظ یک توزیع خوب انتخاب می شوند.[18,23] یک درشت سازی ممکن است المان-تودرتو یا گره-تودرتو یا اصلا تودرتو نباشد. در درشت سازی المان-تودرتو هر المان از شبکه می تواند به عنوان یک المان از شبکه بهبودیافته خود یعنی ظاهر شود. اگر هر گره از شبکه گره ای از شبکه باشد، به این درشت سازی، درشت سازی گره-تودرتو گویند. در غیر این موارد درشت سازی تودرتو نمی باشد. در حالت کلی یک شبکه مثلثی درشت سازی المان-تودرتو ندارد مگر اینکه آن به دقت همانند شبکه اولیه تولید شود. در زیر چند شیوه درشت سازی که در مقالات مختلف آورده شده است، توضیح داده می شود.

#### روش اول MISCoarsening

همانطور که در [18,23] آورده شده است، MISمخفف عبارت Maximal Independent Set می باشد به این معنی که اگر ما شبکه اولیه را به عنوان یک گراف در نظر بگیریم که گره های گراف و یال های آن (اضلاع شبکه) باشند در صورتی که بتوانیم را به عنوان زیر مجموعه به گونه ای تشکیل دهیم که شرط های زیر را ارضا کند:

1. شرط استقلال. یعنی اینکه اگر دو گره وجود داشته باشد، این دو گره در به هم متصل نباشد.
2. با اضافه کردن یک گره از به استقلال آن از بین نرود.

در این صورت یک شبکه درشت از شبکه ریز اولیه داده شده را تشکیل داده ایم. این روش درشت سازی به این گونه عمل می کند که تمامی همسایگان یک گره از شبکه ریز، در شبکه درشت وجود نخواهند داشت. حال این مجموعه از گره ها از شبکه درشت با هر روش شبکه بندی قابل مثلث بندی مجدد می باشد. در ضمن این نکته را باید ذکر کرد که این روش در عین سادگی، کیفیت هندسی را تضمین نمی کند. همانطور که در [19] ذکر شده است این روش برای بکار گیری بسیار ساده می باشد و برای یک یا دو مرحله درشت سازی نتایج بسیار عالی می دهد. در حالی که بعد از چند بار درشت سازی (بیش از 2 بار) کیفیت شبکه با این روش درشت سازی حتی با استفاده از روش های بهینه سازی بین مراحل درشت سازی به شدت کاهش می یابد و در نتیجه مثلث هایی با طول ضلع های خیلی متفاوت در شبکه ظاهر می شود.

الگوریتم MIS :

فرض کنید گراف G(V,E) را داریم :

#### روش دوم Function based coarsening

در مقاله [18,23] یک شیوه جدید برای درشت سازی شبکه ارائه شده است. این شیوه درشت سازی بر پایه تابع فاصله[[34]](#footnote-34) در هر مرحله از درشت سازی می باشد. فرض کنید شبکه اولیه باشد. این تابع فاصله که با نشان داده می شود به نوعی بیان کننده سایز و فاصله گره های می باشد. توسط مقدارش در گره های شبکه تعریف می شود. ایده این روش در محاسبه تابع فاصله، برای هر مرحله درشت سازی و استفاده از آن برای تولید شبکه از مجموعه گره اولیه (یا قبلی) می باشد. پس از محاسبه تابع فاصله، کار بعدی تولید یک مجموعه گره قرار گرفته مطابق با این تابع و سپس مثلث بندی مجدد آن می باشد. به این تکنیک درشت سازی، function-based coarsening می گویند که شامل چهار مرحله می باشد. قبل از توضیح این چهار مرحله ابتدا چند تعریف آورده می شود :

تعریف (1-lipschitz): یک تابع f، روی دامنه 1-lipschitz می باشد هر گاه برای هر متعلق به داشته باشیم :

برای قرار دادن گره ها مطابق با تابع فاصله، حول گره ها دایره هایی با شعاع متناسب با تابع فاصله رسم کرده و محدود می کنیم که دایره های با هم برخورد نداشته باشند.

تعریف (*): فرض کنید* یک عدد حقیقی و یک تابع 1-lipschitz باشد. یک مجموعه نقاط P، *می باشد اگر برای هر دونقطه داشته باشیم:*

|  |  |
| --- | --- |
| **(15)** |  |

یعنی دایره هایی به شعاع به ترتیب با مرکز های ملزمند که با هم تداخل نداشته باشند.

*تعریف (*Nearest Neighbour*): تابعی است که برای مجموعه نقاط ، برای هر گره فاصله اش تا گره که نزدیکترین گره به آن است را مشخص می کند.*

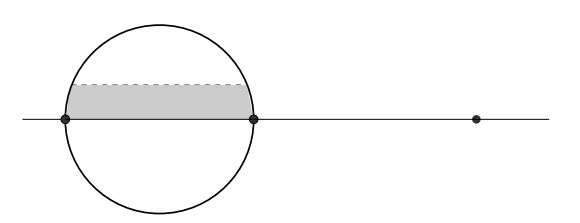
تعریف (conflict graph): یک conflict graph CG(P) از یک مجموعه نقاط P نسبت به یک تابع f β-spacing *و* *یک باندری* B *گرافی* CG(P)=(P,E) *است با همان تعداد نقاط و اضلاع زیر* *:*

در conflict graph یک ضلع دو رأسی را بهم وصل می کند که از لحاظ فیزیکی در مقایسه با مقیاس طول تعریف شده برای آن دو، خیلی نزدیک بهم باشند.

تعریف (protective zone): این قضیه بسیار مهم است تا یک مکانیزمی صورت گیرد تا از نزدیک شدن خیلی زیاد گره های داخل شبکه به اضلاع باندری جلوگیری شود در غیر این صورت مثلث هایی پهن با aspect ratio نامناسب می تواند تشکیل شود. یک راه حل برای این مسئله جلوگیری از وجود گره های درون شبکه، داخل دایره ای می باشد که مرکز آن رو ضلع باندری و شعاع آن نصف ضلع باندری است(شکل 58). اما این روش ممکن است مثلث هایی دراز و نازک نزدیک باندری تولید کند. برای جلوگیری از این اتفاق، یک پارامتر R تعریف می شود که همه گره هایی که داخل دایره قرار می گیرند از نسبت زیر باید پیروی کنند :

|  |  |
| --- | --- |
| **(16)** |  |

در صورتی که گره به باندری خیلی نزدیک باشد آن گره مناسب نیست. برای روش function-based coarsening، و برای روش MIS در نظر گرفته می شود.



شکل (61) نمایی از ناحیه Protective zone جهت جلوگیری از تولید سلول هایی با Aspect ratio نامناسب

4 مرحله مربوط به روش Function based coarsening در زیر آورده شده است:

1. بدست آوردن تابع فاصله شبکه اولیه.

ابتدا باید برای تک تک گره های شبکه تابع فاصله را محاسبه کنیم. این مقدار با نشان داده می شود که گره های شبکه ریز مورد نظر می باشد. همانطور که گفته شد را شعاع دایره هایی در نظر می گیریم که نصف مقدار می باشد یعنی :

|  |  |
| --- | --- |
| **(17)** |  |

1. افزایش مقدار این تابع برای گره های شبکه برای بدست آوردن تابع فاصله جدید.

بعد از محاسبه تابع فاصله اولیه برای هر گره می بایست این مقدار برای هر گره درشت شود (coarsening spacing function). این مقدار جدید را با نشان می دهیم که به صورت زیر به دست می آید:

|  |  |
| --- | --- |
| **(18)** |  |

که ثابت درشت سازی می باشد و بزرگتر از 1 است. در واقع در نظر می گیریم و برای هر مرحله را تغییر می دهیم مثلا برای مرحله اول درشت سازی و برای مرحله دوم و همینطور تا آخر.

1. حذف کردن تعدادی از گره های شبکه به طوری که گره های باقی مانده مطابق با تابع فاصله جدید قرار گرفته باشند.

بعد از محاسبه تابع فاصله جدید، سپس باید یک زیر مجموعه از گره های شبکه ریز را که مطابق با این تابع فاصله جدید هستند، انتخاب کنیم. برای این کار ازconflict graph استفاده می کنیم. پس از اینکه از مجموعه گره های شبکه ریز یک conflict graph تشکیل دادیم باید یک MIS از گره های آن را بدست آوریم.

برای تشکیل گره های شبکه درشت به صورت زیر عمل می کنیم :

ابتدا نقاط مربوط به باندری segment (مانند چهار نقطه تشکیل دهنده مربع) Q را تشکیل می دهد، سپس یک MIS از conflict graph به دست آمده برای باندری به Q اضافه می شود و یک باندری جدید را با این نقاط تشکیل می دهیم. در پایان Q با اضافه شدن گره های داخلی که زیاد به اضلاع باندری نزدیک نباشند و همچنین یک MIS از conflict graph مربوط به این گره ها تشکیل می شود.

نکته :در مرحله حذف کردن گره، گره هایی که روی باندری قرار دارند می توانند حذف شوند اما به طوری که خود باندری تغییری زیادی نکند که با باندری اولیه تفاوت داشته باشد. ولی در طول مراحل درشت سازی ممکن به مرحله ای برسیم که باندری با باندری اولیه تفاوت زیادی داشته باشد به همین دلیل می توان از محدودیت هایی استفاده کرد تا پروفیل باندری با باندری اولیه تفاوت چندانی نداشته باشد که در صورت استفاده از این محدودیت ها روش function based coarsening خود را به راحتی با آن وفق می دهد. ولی گره هایی که برای خود باندری می باشد مثلا اگه باندری مربع باشد گره هایی که مربوط به راس های مربع می شود در طول درشت سازی نباید حذف شوند.

1. تولید شبکه جدید با مثلث بندی مجدد گره های بدست آمده در مرحله قبلی.

بعد از بدست آوردن گره هایی که مطابق با تابع فاصله تعریف شده، حال می توان برای تولید شبکه درشت، عمل شبکه بندی را انجام داد.

تولید دنباله ای از شبکه ها:

روش توضیح داده شده در فوق فقط برای یک مرحله درشت سازی می باشد. برای تولید دنباله ای از شبکه ها دو استراتژی وجود دارد.

1. اولین شبکه درشت تولید شود و سپس شبکه درشت بعدی از شبکه درشت قبلی حاصل شود یعنی در هر مرحله از درشت سازی شبکه از شبکه قبلی یعنی استفاده شود.
2. استراتژی دوم استفاده از یک ثابت درشت سازی برای تولید شبکه درشت مستقیما از شبکه ریز می باشد.

نویسنده در این مقاله روش دوم را ترجیح می دهد چون با توجه نتایج بدست آمده ذکر شده است که این روش همیشه یک دنباله ای از شبکه ها با ساختار خوب را نتیجه می دهد در حالی که روش اول تنها برای شبکه های شبه-یکنواخت این اتفاق می افتد.

الگوریتم این روش :

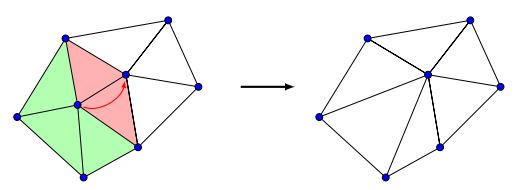
#### روش سوم edge collapse یا edge contraction

همانطور که در مقاله [20] آورده شده است یکی دیگر از راه های درشت سازی شبکه edge collapse می باشد. این روش شامل چند مرحله برای دستیابی به یک شبکه درشت می باشد که در زیر الگوریتم های این مراحل آورده شده و هر یک توضیح داده خواهند شد :

1. Collapse کردن یک ضلع

فرض کنید شبکه ریز M را داشته باشیم در صورتی که ضلع e برایcollapse شدن انتخاب شد و و دو راس این ضلع باشند، را روی منتقل می کنیم، در این صورت تمامی المان هایی که ضلع e به عنوان یکی از اضلاع آن ها می باشد از شبکه M حذف می شوند.(شکل 59)

1. الگوریتم edge collapse :



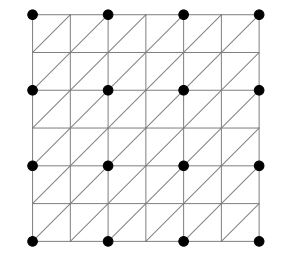
شکل ( 62) نحوه Collapse کردن یک ضلع

1. یافتن یک مجموعه مستقل

برای از این روش ابتدا می بایست یک مجموعه مستقل از رئوس تشکیل شده و تمامی راس های این مجموعه مجاز به collapse شدن نمی باشند و باید در شبکه درشت حضور داشته باشند. متاسفانه یافتن MIS از یک شبکه مشکل می باشد ولی ما برای یک توزیع خوب از رئوس به جای یافتن MIS مجموعه ی مستقلی را می یابیم که تنها شرط اولیه گفته شده در تعریف MIS را ارضا کند(یعنی هیچ دو گره شبکه درشت در شبکه ریز بهم متصل نباشند).

1. الگوریتم یافتن مجموعه مستقل :

*در مقاله آورده شده است که درشت کردن باندری امکان پذیر است و لی مطلوب نیست زیرا که ممکن است به شدت در شکل باندری تاثیر گذار باشد. از این رو به جای اینکه مجموعه مستقل*S *تهی تعریف شود آن را با یک مجموعه اولیه غیر تهی که شامل تمامی رئوس باندری می باشد تعریف می کنیم ازین رو* S *دیگر یک مجموعه مستقل طبق تعریف نمی باشد چرا که اکنون شامل رئوسی می باشد که در شبکه ریز بهم متصل می باشند.* (همانند نشان داده شده در شکل 60)

**

شکل (63) مجموعه S اولیه

1. انتخاب اضلاع برای درشت سازی

با collapse کردن ضلع e با دو راس ، المان هایی که این ضلع در آن ها مشترک است حذف خواهند شد و آن دسته از المانهایی که راس در آن ها مشترک است نازک و کشیده می شوند و در نتیجه شبکه درشت تولید شده از کیفیت ضعیفی برخوردار خواهد بود. به منظور غلبه بر این مشکل بهترین و سریعترین راه انتخاب کوتاهترین ضلع المان می باشد. علاوه برای این قید، خود مجموعه مستقل قید دیگری بر انتخاب ضلع محسوب می شود که نمونه بارز آن راس های اضلاع باندری می باشد که هر دو راس یک ضلع باندری در مجموعه مستقل S قرار دارد.

1. الگوریتم انتخاب ضلع :

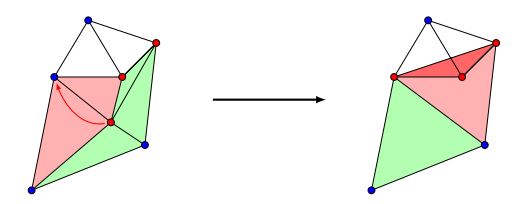
الگوریتم فوق ضلع مورد نظر برایcollapse را انتخاب می کند به این صورت که برای هر سه ضلع المان K از شبکه M این شرط را بررسی می کند که اگر هر دو راس اضلاع این المان در مجموعه مستقل باشند مجاز به درشت کردن K نمی باشیم ولی در صورتی که اضلاعی از این المان پیدا شوند که هر دو راس در مجموعه مستقل نباشند آنگاه آن ضلعی که از همه کوتاهتر است را به عنوان ضلع e جهت collapse کردن، انتخاب می کنیم.

1. انتخاب راس (گره) جهت collapse کردن

بعد انتخاب ضلع مورد نظر جهت collapse، این سول در ذهن بوجود می آید که کدام یک از دو راس این ضلع باید روی دیگری قرار گیرد. اگر یکی از این دو راس در مجموعه مستقل قرار داشته باشد پاسخ به این سول نسبتا آسان است، چون راسی که در مجموعه مستقل نباشد مجاز به حذف شدن (یا انتقال) می باشد. اما اگر هر دو راس در مجموعه مستقل نباشد، هر کدام از آنها قابل حذف می باشند. این مسئله هم به راحتی قابل حل است به این صورت که راس با شماره پایین تر روی راس با شمار بالاتر قرار می گیرد. همانطور هم که قبلا هم گفته شد در صورتی که هر دو راس در S باشند ما مجاز به collapse کردن این ضلع نمی باشیم. الگوریتم زیر انتخاب راس جهت collapse کردن را نشان می دهد :

1. الگوریتم انتخاب راس :
2. کیفیت شبکه :

چون collapse کردن ضلع باعث کشیده شدن المان های باقی مانده در طول ضلع collapse شده، می شود این المان ها نازک و دراز شده و در نتیجه کیفیت شبکه پایین می آید. این مسئله مکررا برای هندسه های پیچیده اتفاق می افتد. برای جلوگیری از این مسئله یک بررسی کیفیت شبکه به الگوریتم edge collapse اضافه می شود. چون edge collapse یک عملکرد محلی می باشد کافیست تنها المان هایی که مجاور گره حذف شده هستند، کیفیت آن ها مورد بررسی قرار گیرد. به عنوان یک معیار برای کیفیت یک سلول، نسبت حجم به قطر در سه بعد (تتراهدرا) و نسبت مساحت به قطر در دو بعد (مثلث) در نظر گرفته می شود. که قطر دو برابر circumradius تعریف می شود. (circumradius شعاع دایره ای که از همه رئوس یک چند ضلعی (در اینجا مثلث) عبور می کند). حتی بدتر از یک کیفیت شبکه بد یک شبکه نادرست می باشد. یک مثال دو بعدی برای همچین شرایطی در شکل 61 نشان داده شده است. خوشبختانه این مسئله را می توان با مقایسه جهت سلول ها قبل و بعد از collapse کردن فهمید زیرا آن در همچین مواردی برعکس می شود.



شکل ( 64) شرایط بوجود آمده پس از درشت سازی که منجر به پایین آمدن کیفیت شبکه می شود

1. الگوریتم کیفیت شبکه :

الگوریتم کامل درشت سازی به روش edge collapse :

Algorithm b

Algorithm c

Algorithm d

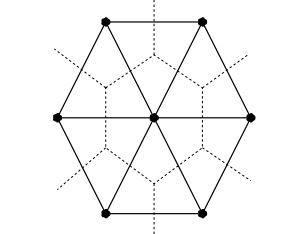
Algorithm a

Algorithm e

الگوریتم فوق نشان می دهد در صورتی که یک سلول در مرحله اول قابل collapse شدن نباشد ممکن است در مراحل بعدی بخاطر تغییر شبکه حاصل ازcollapse شدن سایر المان ها، این امکان فراهم شود. به همین خاطر در انتهای الگوریتم آورده شده است تا زمانی که تعداد المان های شبکه تغییر نکرده حلقه ادامه پیدا کند.

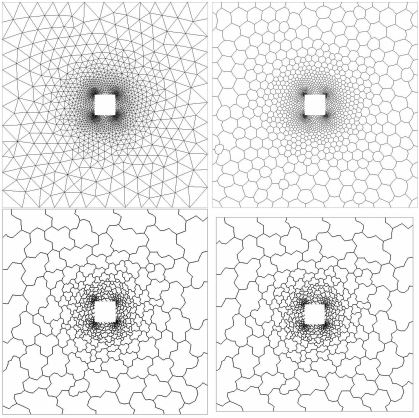
#### روش چهارم agglomeration

در حالی که ممکن است استراتژی های درشت سازی خودکار بعضی از دشواری های کاربردی در ایجاد مراحل شبکه درشت را در الگوریتم چندشبکه ای با شبکه های بی سازمان را برطرف سازد، ولی آن ها نمی توانند مسئله نیرومندی ساختارهای شبکه درشت را پوشش دهند. برای مثال ممکن است دیده شود یک روند درشت سازی خودکار یک یا چندین گره ضروری و مهم را که هندسه را توصیف می کند حذف کند، که نتیجه آن تغییر در هندسه بین مراحل مختلف تولید شبکه و عدم موفقیت در الگوریتم چندشبکه ای می باشد. در واقع مثلث بندی یک مجموع از گره های درشت حول یک هندسه پیچیده کار دشواری می باشد. این دشواری ها به طور کارآمدی توسط روشagglomeration قابل برطرف شدن می باشد[9]. روش های agglomeration بر پایه حجم کنترل می باشد و بنابراین می تواند برای رویه های سلول محور و راس محور بکار برده شود. برای رویه سلول محور خود مثلث ها به عنوان حجم کنترل در نظر گرفته می شوند در مقابل برای رویه راس محور حجم های کنترل همانند شکل زیر سلول هایی تعریف می شود که از اتصال میانه مثلث ها شکل می گیرند.



شکل (65) نمایی از نحوه درشت سازی در روش Agglomeration

ایده روش agglomeration آمیختن حجم های کنترل همسایه در شبکه ریز برای تولید یک مجموعه کوچکتر از چندضلعی های بزرگتر می باشد. این پروسه برای ایجاد دنباله ای از شبکه درشت در شکل زیر نشان داده شده است. پیچیدگی تعریف سلول یا حجم کنترل با پیش رفتن مراحل درشت سازی افزایش می یابد.



شکل (66) سلسله ای از شبکه های درشت تولید شده به روش Agglomeration

نکته ای در رابطه با درشت سازی باندری:

همانطور که در [19] آورده شده است نقاطی که apex نامیده می شوند نباید از باندری حذف شوند. apex ها نقاطی می باشند که در قسمت های تیز باندری قرار دارند مانند راس های مربع که قبلا گفته شد. با توجه به تعریف صورت گرفته در این مقاله به طور کلی راس هایی apex نامیده می شوند که زاویه ی دو ضلع برخورد کننده در این راس ها بیشتر از 20 درجه باشد. در این صورت این راس ها نباید از باندری حذف شوند.

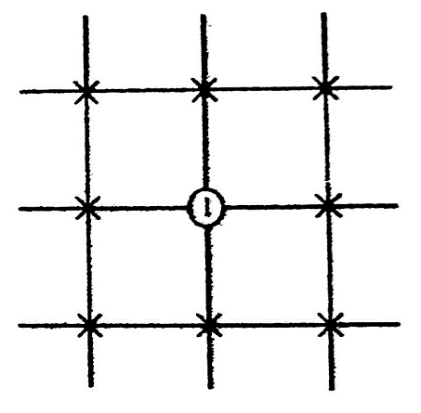
* + 1. روش های درون یابی در مالتی گرید هندسی

برای انتقال اطلاعات روش های مختلفی وجود دارد که در این جا ما آن را براساس نوع ذخیره داده ها در شبکه یعنی سلول محور و راس محور دسته بندی می کنیم [24]

1. راس محور :

* تزریق مستقیم[[35]](#footnote-35) برای انتقال داده ها از شبکه ریز به شبکه درشت :

برای انتقال داده ها از شبکه ریز به شبکه درشت از عملگر نمادین استفاده می کنیم. راحت ترین روش برای این انتقال، عمل تزریق مستقیم می باشد. بدین صورت که اطلاعات مقادیری از شبکه ریز را که از لحاظ فیزیکی دقیقا روی شبکه درشت قرار می گیرند روی نقاط این شبکه منتقل می کنیم. این روش برای معادلاتی که خاصیت بیضوی قوی دارند مانند معادله لاپلاس بسیار مناسب می باشد.



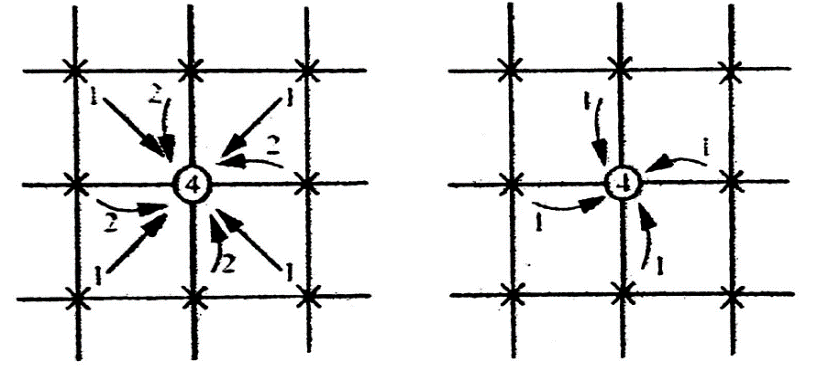
**شکل (67) نمایی از قرار گرفتن دقیق دو گره از لحاظ فیزیکی روی همدیگر**

* عملگرهای 5 و 9 نقطه ای برای انتقال داده ها از شبکه ریز به شبکه درشت

ازاین روش برای شبکه های مربعی و برای معادلاتی که خاصیت بیضوی ضعیفی دارند یا اینکه اصولا خاصیت غیرخطی دارند، استفاده می کنیم که به صورت زیر تعریف می شوند:

*عملگر 5 نقطه ای*

*عملگر 9 نقطه ای*

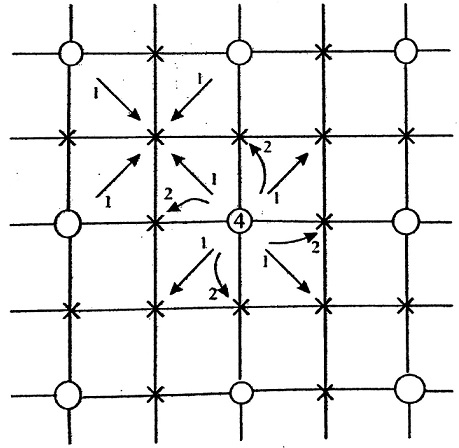
**

شکل ( 68) نحوه عملکرد در درون یابی به روش های 5 نقطه ای و 9 نقطه ای

* میانیابی خطی[[36]](#footnote-36)

برای انتقال داده ها از شبکه درشت به شبکه ریز از عملگر نمادین استفاده می کنیم. معمول ترین روش برای این عمل، استفاده از میانیابی خطی در دو جهت می باشد که نتیجه کلی آن به صورت زیر می باشد:

|  |  |
| --- | --- |
| **(19)** |  |



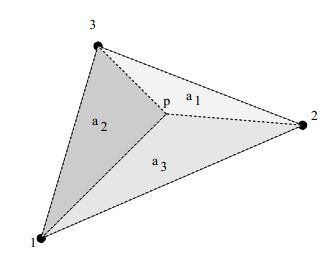
**شکل (69) نحوه عملکرد درون یابی به روش خطی**

* Piecewise linear interpolation

در این روش برای انتقال داده ها از شبکه ریز به شبکه درشت ابتدا می بایست مشخص کنیم که هر گره شبکه درشت در کدام سلول از شبکه ریز قرار دارد و مقدار متغیر در گره درشت از طریق میانگین وزنی سه گره مثلث ریز که این گره را احاطه کرده اند، توسط رابطه زیر به دست می آید[9] :

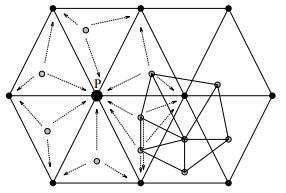
|  |  |
| --- | --- |
| **(20)** |  |

مقدار متغیر در گره احاطه شده یعنی گره شبکه درشت و همچنین متغیرهای مقادیر مربوط به گره ها مثلث ریز می باشد. مساحت مربوط به مثلث ریز که گره درشت در آن قرار دارد، و مساحت مربوط به مثلث های نشان داده شده در شکل زیر می باشد :

**

**شکل (70) احاطه شدن یک گره از شبکه درشت در مثلثی از شبکه ریز**

برای انتقال مانده از شبکه ریز به شبکه درشت، ابتدا باید تمام مثلث هایی که شامل این گره درشت می باشند را پیدا کنیم. مقدار مانده در گره شبکه درشت مساوی است با جمع مانده در گره های شبکه ریز که درون این مثلث ها از شبکه درشت قرار دارند.



**شکل (71) مکان یابی تمامی گره های شبکه درشت در سلول های شبکه ریز**

برای انتقال داده ها از شبکه درشت به شبکه ریز همانند فرمول فوق عمل می کنیم با این تفاوت که نقش گره های ریز و درشت عوض می شود یعنی ما به دنبال محاسبه مقدار متغیر در گره ریز می باشیم از این رو با استفاده از گره های مثلث شبکه درشت که این گره ریز را احاطه کرده است به صورت همان شیوه ای که در بالا توضیح داده شده است عمل می کنیم.

1. سلول محور :

از آنجایی که شبکه مورد نظر ما در این تحقیق مثلثی می باشد آن را به 2 قسمت شبکه های مثلثی منظم و سازمان یافته و شبکه های مثلثی بی سازمان دسته بندی می کنیم :

* شبکه های مثلثی منظم و سازمان یافته :

روش اول (تزریق مستقیم) :

همانند شبکه های راس محور در صورتی که اطلاعات مقادیر مراکز سلول ها از شبکه ریز که از لحاظ فیزیکی دقیقا روی شبکه درشت قرار می گیرند روی نقاط این شبکه منتقل می کنیم.

روش دوم :

در این مورد چون مثلث ها منظم هستند عمل درشت سازی با دو برابر کردن اضلاع مثلث ها انجام می شود به این صورت که هر چهار مثلث شبکه ریز تشکیل یک مثلث شبکه درشت را می دهند[8].

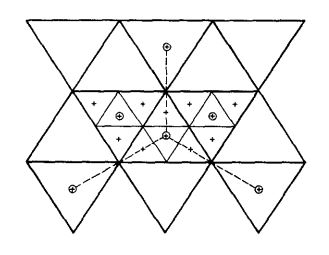
در این نوع مثلث بندی برای انتقال داده ها از شبکه ریز به درشت از رابطه زیر استفاده می شود :

|  |  |
| --- | --- |
| **(21)** |  |

که در این رابطه و به ترتیب مقادیر مربوط به داده و مساحت مثلث درشت و و مقادیر و مساحت مربوط به مثلث ها در شبکه ریز می باشد. از آنجایی که شبکه بندی منظم می باشد مساحت هر مثلث درشت 4 برابر مساحت هر مثلث از شبکه ریز می باشد. همچنین برای انتقال باقی مانده ها (Residual) از شبکه ریز به شبکه درشت از رابطه استفاده می کنیم:

|  |  |
| --- | --- |
| **(22)** |  |

که در رابطه فوق و به ترتیب مقادیر باقی مانده در شبکه درشت و شبکه ریز می باشند. رابطه فوق نشان می دهد که مقدار باقی مانده از جمع باقی مانده های چهار مثلث در شبکه ریز بدست می آید. حال برای انتقال داده ها از شبکه درشت به ریز از آنجایی که مثلث ها منظم هستند، مرکز هر مثلث از شبکه ریز در مسیر خطی قرار دارد که مراکز دو مثلث از شبکه درشت را به همدیگر وصل می کند و بنابراین با درون یابی خطی می توان مقدار متغیر مثلث کوچک را بدست آورد.



**شکل (72) درون یابی به روش مساحت مثلث ها برای شبکه های مثلثی منظم و سازمان یافته**

* در صورتی که شبکه مثلثی بی سازمان داشته باشیم :

برای مش بی سازمان به دلیل اینکه سلول های شبکه ریز و درشت از لحاظ فیزیکی دقیقا روی هم قرار نمی گیرند، روش تزریق مستقیم برای انتقال داده ها از شبکه ریز به شبکه درشت قابل کاربرد نمی باشد. در صورتی که با چنین شبکه هایی سروکار داشته باشیم 2 روش زیر پیشنهاد می شود :

روش اول :

در این روش برای انتقال اطلاعات از شبکه ریز به شبکه درشت از linearly weighted restriction oprator استفاده می کنیم [10] که رابطه آن به صورت زیر می باشد :

|  |  |
| --- | --- |
| **(23)** |  |

در رابطه فوق ابتدا باید سلول هایی از شبکه ریز که از نظر فیزیکی داخل سلول شبکه درشت قرار می گیرد را بیابیم و سپس مقدار متغیر مثلث از شبکه درشت برابر با میانگین مقادیر سلول های ریزی می باشد که داخل این مثلث درشت قرار دارد. در این رابطه و و همچنین اندیس های به ترتیب مربوط به سلول شبکه درشت و شبکه ریز می باشد. پارامتر تعداد مثلث های شبکه ریز که درون مثلث شبکه درشت قرار دارند را نشان می دهد.

در این روش برای انتقال مانده از شبکه ریز به شبکه درشت به جای رابطه فوق از رابطه زیر استفاده می­شود:

|  |  |
| --- | --- |
| **(24)** |  |

برای حالت گسترش سازی یعنی انتقال اطلاعات از شبکه درشت به شبکه ریز، از آنجایی که داده ذخیره شده در مرکز سلول نشان گر مقدار متغیر مورد نظر در کل سلول می باشد به همین خاطر آن دسته از سلول های شبکه ریز که از نظر فیزیکی داخل این سلول شبکه درشت قرار می گیرند، همین مقدار داده سلول درشت برای آن ها در نظر گرفته می شوند، یعنی :

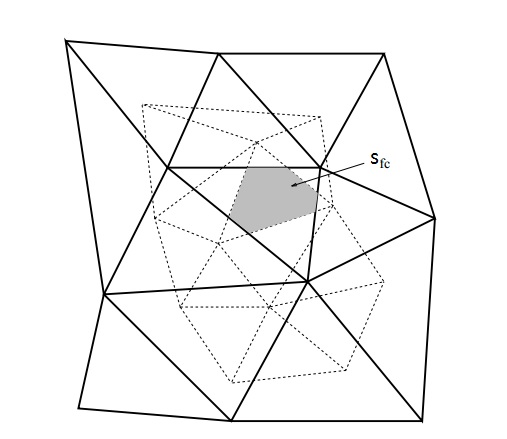
|  |  |
| --- | --- |
| **(25)** |  |

*روش* Area weighting transfer*:*

*در این روش برای انتقال داده ها به شبکه ریز و درشت از تقاطع سلول های این دو شبکه استفاده می کنیم و سپس از رابطه هایی که در ادامه آورده می شود این انتقال را انجام می دهیم.*[11,12]

*برای انتقال داده ها از شبکه ریز به شبکه درشت رابطه زیر را داریم:*

|  |  |
| --- | --- |
| **(26)** |  |

**

**شکل (73) تقاطع بین مثلث های شبکه ریز و درشت**

*همچنین برای انتقال مانده ها از شبکه ریز به شبکه درشت از رابطه ای مشابه با رابطه فوق استفاده می کنیم :*

|  |  |
| --- | --- |
| **(27)** |  |

*همچنین برای حالت برعکس یعنی انتقال داده ها از شبکه درشت به شبکه ریز از رابطه زیر استفاده می­کنیم:*

|  |  |
| --- | --- |
| **(28)** |  |

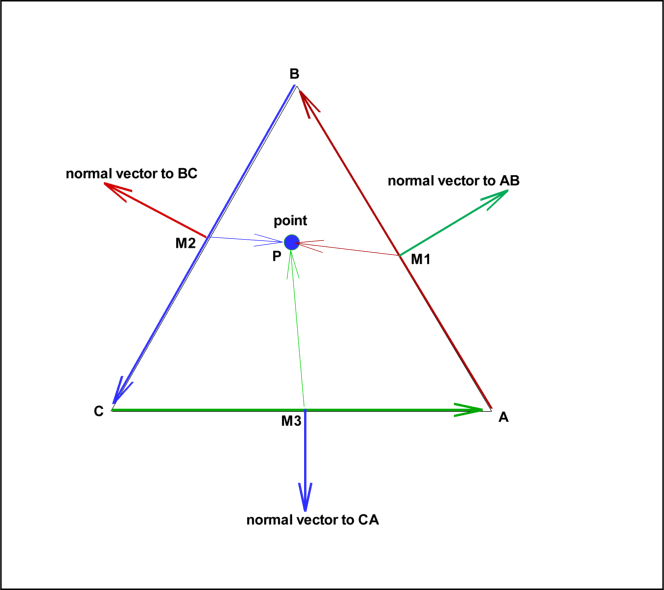
*در روابط فوق اندیس های و متغیر های و به ترتیب مربوط به شبکه درشت و شبکه ریز می باشد. همچنین مساحت مشترک بین دو مثلث متقاطع از شبکه درشت و ریز و و مساحت های مربوط به یک مثلث خاص در شبکه درشت و شبکه ریز است. و مقدار مانده ها در شبکه درشت و ریز را نشان می دهد.*

*در این روش مسئله ما به پیدا کردن مساحتی از دو سلول شبکه ریز و درشت که با هم تقاطع دارند محدود می شود.(یعنی ) برای این منظور باید بفهمیم که هر گره از شبکه ریز در کدام سلول شبکه درشت قرار دارد. یک روش برای این جستجو این است که عملیات جستجو را از گره اول شبکه ریز برای تمامی سلول های شبکه درشت انجام دهیم و زمانی که مکان این گره پیدا شد، همین روند را برای تمامی گره های شبکه ریز انجام دهیم. همانطور که می دانیم یکی از مزایای روش های چندشبکه ای کوتاه کردن زمان همگرایی می باشد، در حالی که این جستجو خود به تنهایی زمان بر می باشد و حتی ممکن است زمانی که برای این جستجو صرف می شود از زمان حل معادلات بیشتر شود که مطلوب ما نمی باشد. برای این منظور در زیر الگوریتمی برای تسریع در زمان مکان یابی گره های شبکه ریز در شبکه درشت ارائه می شود که علاوه بر در این روش، در روشPiecewise linear interpolation قابل استفاده است. در این الگوریتم به این صورت عمل می کنیم:*

*ابتدا یک گره شبکه ریز که مکان آن برای ما مشخص است مثلا یک گره از trailing edge در یک ایرفویل را انتخاب می کنیم و عملیات جستجو را از مثلثی در شبکه درشت که حدس می زنیم این گره داخل این مثلث یا حداقل اطراف آن قرار دارد، شروع می کنیم. طبق رابطه ای که در ادامه آورده می شود محاسبه می کنیم که این گره داخل این مثلث قرار دارد یا خیر. اگر داخل مثلث قرار نداشت به سراغ همسایه های این مثلث می رویم از آنجایی که هر مثلث حداکثر سه همسایه دارد عملیات جستجو را برای هر سه مثلث انجام می دهیم و اگر باز هم درون این همسایه ها هم قرار نداشت عملیات جستجو را برای همسایه های تک تک همسایه های مثلث اولیه انجام می دهیم و این عمل را تا جایی که مکان گره مشخص شود ادامه می دهیم. زمانی که مثلثی از شبکه درشت که این گره شبکه ریز داخل آن قرار دارد پیدا شد، همین کار را برای گره دیگر انجام می دهیم اما این بار این جستجو را از همسایه های این گره و مثلثی که درون آن قرار دارد شروع می کنیم. این کار به این دلیل انجام می شود که چون مکان مثلث مشخص است و از آنجا که گره مورد نظر همسایه گره قبلی می باشد احتمال قرار گرفتن درون این مثلث و یا همسایه های آن بسیار زیاد است. به طور میانگین 10 جستجو برای پیدا کردن مثلثی که گره داخل آن قرار دارد مورد نیاز است که این تعداد به سایز شبکه بستگی ندارد بلکه به چگالی بین شبکه های ریز و درشت بستگی دارد. به این ترتیب به وسیله این الگوریتم زمان جستجو بسیار کاهش می یابد. از آنجایی که ما تقاطع بین مثلث های شبکه ریز و درشت را بدست آورده ایم برای انتقال داده ها از شبکه درشت به شبکه ریز دیگر نیاز به جستجو گره های شبکه درشت در شبکه ریز نمی باشد و با استفاده از مساحت مشترک بین دو مثلث شبکه ریز و درشت این انتقال را همانند فوق انجام می دهیم.*

برای پیدا کردن اینکه یک نقطه درون یک مثلث قرار دارد از ضرب داخلی هر یک از بردارهای تشکیل دهنده اضلاع یک مثلث با بردار متصل کننده وسط اضلاع به نقطه جدید استفاده می­شود. شکل زیر این الگوریتم را نشان می­دهد. بنابراین با توجه به این شکل در صورتیکه سه شرط زیر بطور همزمان برقرار باشد، نقطه مورد نظر درون مثلث قرار می­گیرد:





**شکل (74) شرایط قرار گیری یک گره درون یک مثلث**

بهبود سازی شبکه :

یکی از برتری های مهم در شبکه های بی سازمان امکان بهبود بخشیدن به شبکه به صورت محلی در طول حل می باشد.[13] برای مثال در جریان ترنسونیک روی یک ایرفویل نواحی مانند shockwaves و stagnation regions و trailing edges باید مورد بررسی قرار بگیرند. در این نواحی ممکن است بخاطر خطای زیاد جوابی که از حل معادلات بدست می آوریم دقیق نباشند و باید شبکه این نواحی بهبود یابد که این بهبود محلی در هنگام حل همانطور که گفته شد از مزایای شبکه های بی سازمان می باشد.

*دو معیار برای اینکه بفهیمم شبکه یک ناحیه نیاز به بهبود دارد، موجود می باشد :*

1. *معیار اختلاف فشار روی گره های یگ ضلع*

*این معیار توسط رابطه زیر تعریف می شود :*

|  |  |
| --- | --- |
| **(29)** |  |

در صورتی که این شرط برقرار باشد ضلع ij با قرار دادن یک گره در مرکز آن بهبود داده می شود. در این رابطه و به ترتیب فشار در گره و و طول ضلع می باشد. و ماکزیمم و می نیمم فشار روی همه گره های شبکه و طول معیار مسئله است که در ایرفویل وتر آن می باشد. در این رابطه هم ثابت حساسیت برای معیار اختلاف فشار می باشد. این قانون باعث بهبود نواحی مربوط به shockwaves و stagnation regions می شود.

1. معیار تفاوت آنتروپی روی گره های یک ضلع

معیار دوم مشابه معیار اول ولی براساس تفاوت آنتروپی روی یک ضلع می باشد که به صورت زیر تعریف می شود :

|  |  |
| --- | --- |
| **(30)** |  |

در رابطه فوق s آنتروپی به صورت با ثابت آدیاباتیک می باشد. برای مثال ضریب حساسیت برای معیار فشار و معیار انتروپی در ایرفویل NACA-0012 به ترتیب و می باشد.

* + 1. روش های درشت سازی در مالتی گرید جبری

دو راه متفاوت برای انتخاب گره های شبکه درشت وجود دارد. نظریه اول تلاش می کند تا همه گره های شبکه ریز را به دو قسمت گره درشت (C-points) یعنی گره هایی که باید در شبکه درشت وجود داشته باشند و گره ریز (F-points) که از گره های C درون یابی می شوند، تقسیم کند. نظریه دوم مربوط به درشت سازی توسط aggregation می باشد. [21]

#### درشت سازی کلاسیک

از آنجایی که همه ضرایب ماتریس اسپارس به طور یکسان دارای اهمیت از جهت حضور در شبکه درشت نیستند، باید درایه هایی که اهمیت بیشتری دارند انتخاب شوند. برای توضیح درشت سازی AMG ابتدا تعاریفی آورده می شود :

تعریف : یک گره I به شدت وابسته به گره j (strong dependence) می باشد یا برعکس گره j به شدت روی گره i (strong influence) تاثیر می گذارد اگر رابطه زیر برقرار باشد :

|  |  |
| --- | --- |
| **(31)** |  |

گره هایی که به شدت روی گره های دیگه تاثیر می گذارند به عنوان گره های شبکه درشتC و آن گره هایی که به شدت وابسته هستند به عنوان گره F انتخاب می شوند.

تعریف : برای گره شبکه ریز i همسایه های آن را در صورتی که و باشد در مجموعه ای به نام قرار می دهیم.

تعریف : گره هایی از مجموعه که به شدت روی i تاثیر می گذارند را با مجموعه نمایش می دهیم. در این مجموعه گره هایی می باشد که i به همه آن ها وابسته است.

تعریف : گره هایی از مجموعه که به شدت به i وابسته اند را با مجموعه نمایش می دهیم.

تعریف strength matrix S :

#### روش RS

در این روش تلاش بر این است که دو شرط زیر اجرا شود :

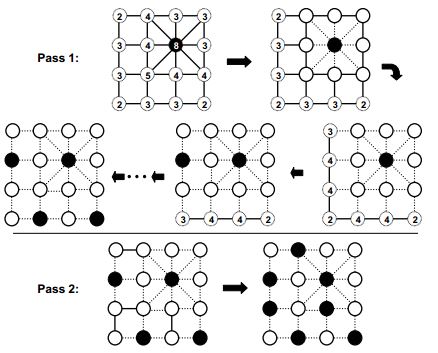
1. *برای هر گره j که به شدت روی یک گره i متعلق به F تاثیر می گذارد،j یا یک گره شبکه درشت و متعلق به C می باشد یا اینکه آن به شدت وابسته به یک گره k متعلق به C می باشد که همچنین خود K روی گره i تاثیر می گذارد.*
2. *گره های مجموعه C باید یک زیر مجموعه مستقل از همه گره ها را تشکیل دهند یعنی دو گره در مجموعه C نباید با یک دیگر در شبکه ریز متصل باشند و اگر یک گره شبکه درشت به مجموعه C اضافه شود استقلال مجموعه از بین نرود.*

*شرط 1 طراحی می شود تا کیفیت درون یابی را تضمین کند در حالی که شرط 2 سایز گره های شبکه درشت را محدود می کند. در کل هر دو شرط همزمان با هم نمی توانند برقرار باشند، بنابراین شرط اول اجرا می شود در حلی که شرط دوم به عنوان یک خط راهنما استفاده می شود.*

الگوریتم انتخاب گره شبکه درشت :

برای اینکه مشخص شود که کدام گره شبکه ریز به عنوان گره شبکه درشت انتخاب شود ابتدا ماتریس اسپارس داده شده را به عنوان یک گراف در نظر می گیریم که هر گره متناسب با یک ضریب ماتریس است و برای تمام این گره ها مقدار را مشخص می کنیم. معیاری است که که تعداد گره های همسایه گره i که به شدت به این گره وابسته است را نشان می دهد. بنابراین یک گره با بیشترین مقدار ابتدا انتخاب شده و برای حضور در شبکه درشت و مجموعه C انتخاب می شود که معمولا چندتا گره وجود دارد.

بنابراین تمام گره هایی که به شدت به گره i وابسته اند در مجموعه F قرار می گیرند. حال مقدار گره هایی که با این گره های جدید F همسایه بوده اند افزایش می یابد تا شانس حضور آن ها در شبکه درشت افزایش یابد. این روند مطابق شکل 71 تا زمانی که همه گره ها در دو مجموعه C و F قرار گیرند ادامه می یابد.

**

**شکل (75) الگوریتم انتخاب گره های شبکه درشت در روش** RS

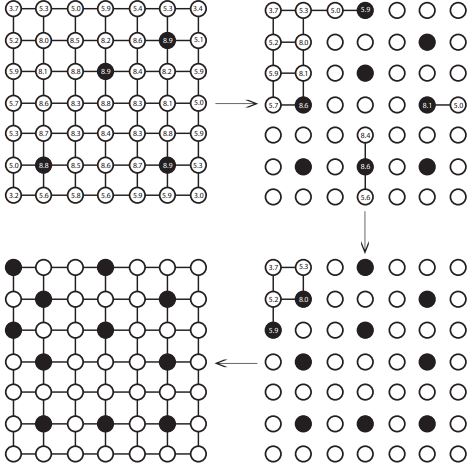
همانطور که در شکل فوق مشاهده می شود مرحله اول درشت سازی شرط دوم را ارضا می کند ولی شرط اول را نه و از آنجایی که شرط اول برای ما مهمتر است مرحله دوم نیز اضافه می شود تا شرط 1 ارضا شود.

گره هایی از مجموعه F که با هم اتصال دارند (F-F conections) ولی توسط یک گره مشترک از مجموعه C تاثیر نمی پذیرند. این گره ها با خط bold شده در شکل فوق در مرحله دوم مشخص شده اند. برای اینکه شرط 1 برقرار شود باید از این گره ها که باعث نقض شرط 1 شده اند به مجموعه C اضافه شود تا شرط ارضا شود این کار باید به صورتی انجام شود که کمترین گره ممکن به مجموعه C اضافه شود. [22]

#### PMIS Coarsening

این روش شبیه روش RS می باشد با این تفاوت که شرط اول نیاز به ارضا شدن نمی باشد. یعنی اینکه وجود ارتباط F-F بدون یک گره درشت مشترک (C-point) مجاز می باشد. در این روش همانند روش قبل ابتدا برای تمام گره های شبکه ریز اولیه مقدار را تعیین کرده و علاوه بر آن برای اینکه گره ها دارای های یکسان نباشند یک مقدار رندوم بین صفر تا 1 به اضافه می شود. در این روش بجای انتخاب گره با بیشترین مقدار در بین همه گره ها در هر مرحله، گره ایی با بیشترین در بین همسایه های خودش انتخاب می شود. این روش به این صورت می باشد که ابتدا یک مجموعه مستقل از گره ها انتخاب شده و در صورتی که دارای بیشترین نسبت به همسایه های خود باشد به عنوان گره شبکه درشت انتخاب می شود و تمام همسایه های آن در مجموعه F-points قرار خواهند گرفت. ازبین گره های باقی انده بار دیگر یک مجموعه مستقل تشکیل داده و این روند تا جایی که همه گره ها به دو مجموعه C و F تعلق گیرند (شکل 72) ادامه می یابد. الگوریتم این روش در زیر آورده شده است.

الگوریتم PMIS :



**شکل (76) الگوریتم انتخاب گره های شبکه درشت در روش** PMIS

* + 1. روش های درون یابی در مالتی گرید جبری

برای توضیح درون یابی روش جبری باید بعضی از اصطلاحات که لازمه تمام روش های درون یابی در چند شبکه ای جبری می باشند، معرفی گردند.[21]

تعریف : برای گره شبکه ریز i همسایه های آن را در صورتی که و باشد در مجموعه ای به نام قرار می دهیم.

این را به صورت زیر تقسیم بندی می کنیم :

* همسایه های گره i متعلق به شبکه درشت (C-points) که به شدت روی i تاثیر می گذارند را با نشان داده و مجموعه coarse interpolatory نامیده می شود.
* همسایه های گره i متعلق به شبکه ریز (F-points) که به شدت روی i تاثیر می گذارند را با نشان می دهیم.
* گره هایی که به شدت روی i تاثیر نمی گذارند را با نشان می دهیم. این مجموعه را weakly connected neighbours می نامند و ممکن است هم در C-points و هم در F-points باشند.

**درون یابی کلاسیک :**

درون یابی کلاسیک یکی از مولفه های استاندارد در بسیاری از الگوریتم هایAMG می باشد و یک روش موثر برای مسائل متنوع است.

برای درون یابی به این روش گره هایی که متعلق به شبکه درشت می باشند مقدار خطای بدست آمده در آنها به طور مستقیم به شبکه ریز منتقل می شوند. ولی برای بدست آوردن مقدار خطا در شبکه ریزی که متعلق به F-points می باشند باید از مقدار خطای گره هایی که متعلق به C-points می باشند استفاده نمود. رابطه روش درون یابی کلاسیک () برای گره i در شبکه ریز به صورت زیر می باشد :

|  |  |
| --- | --- |
| **(32)** |  |

*در رابطه فوق تنها چیزی که باقی می ماند محاسبه سیگما می باشد که این جمع روی همسایه هایی از گره ریز که در شبکه درشت حضور دارند انجام می شود. در این رابطه وزن درون یابی از رابطه زیر بدست می آید :*

|  |  |
| --- | --- |
| **(33)** |  |

عملگر درون یابی (گسترش سازی و محدود سازی) با دو رابطه فوق برای دورن یابی کلاسیک بدست می آید.

1. پیاده سازی و زیربرنامه های مورد استفاده

با توجه به اینکه در نحوه پیاده سازی توجه به آموزشی بودن برنامه در اولویت بوده است، توجه ویژه ای به بهینه بودن کدنویسی نشده است. برای مثال از دستورات مربوط به اختصاص حافظه پویا استفاده نشده است.

هر کدام از قسمت های برنامه که بصورت یک زیربرنامه جداگانه نوشته شده است دارای مستندات کامل مربوط به خود می باشد که خواننده در صورت تمایل می تواند به مستندات مربوط به آن که در جدول (1) آورده شده است، مراجعه نماید.

* 1. برنامه اصلی Multigrid

در برنامه اصلی پس از تعریف پارامترها و آرایه­های لازم، موارد زیر بترتیب اجرا خواهد شد. لازم بذکر است که شماره گذاری زیر بر اساس شماره گذاری موجود در فایل فرترن برنامه می باشد. همچنین پارامترها و آرایه هایی که در کل برنامه استفاده می شود طبق پیوست 1 می باشد.

1. خواندن فایل شبکه

با توجه به اینکه در چه مرحله ای قرار داریم شبکه مورد نظر با توجه به مقدار NMF (مقدار صفر یعنی ریزترین شبکه و مقدار یک یعنی شبکه درشت اول) با استفاده از زیربرنامه Read\_2DMeshخوانده شده و اطلاعات لازم ذخیره می­شود. لازم است توجه شود که شرایط مرزی در این فایل (فایل مربوط به شبکه) باید اعمال گردد که در مستندات این زیربرنامه بطور مفصل به آن پرداخته شده است.

1. خواندن پارامترهای لازم برای حل معادلات

با توجه به الگوریتم حل اشاره شده، با فراخوانی زیربرنامه Read\_SettingV1اطلاعات لازم برای حل معادلات از کاربر گرفته و ذخیره می­شود.

1. شماره گذاری مجدد اضلاع برای اعمال شرایط مرزی

با فراخوانی زیربرنامه MeshBC اضلاع غیر مرزی به ابتدای آرایه مربوط به ذخیره اطلاعات اضلاع تشکیل دهنده شبکه منتقل شده و همچنین سایر نواحی شبکه متناسب با شرایط مرزی مربوطه شماره گذاری مجدد می گردد.

1. محاسبه مساحت سلول ها و مختصات مرکز آنها

با فراخوانی زیربرنامه GeoCal2Dمساحت، بردارهای عمود با بعد، طول اضلاع و مختصات مرکز هر کدام از سلول های شبکه محاسبه شده و در آرایه های مربوطه ذخیره می­شود.

1. مقداردهی اولیه

در اینجا با فراخوانی زیربرنامه InitMeanFlow\_Inviscid برخی از آرایه ها و پارامترها که در ادامه مورد استفاه قرار خواهند گرفت، از جمله مقادیر بقایی مقداردهی می شود. باید بخاطر داشت در صورتیکه مقدار پارمتر init، که یکی از ورودی های این زیربرنامه می باشد، برابر 1 باشد، مقادیر بقایی با استفاده از فایل SolutionData.txt که در تکرارهای قبل ذخیره شده است، مقداردهی می شود.

1. محاسبه فشار

مقدار فشار برای هر کدام از سلول ها محاسبه می گردد.



1. تعیین شرایط مرزی

با فراخوانی زیربرنامه های مربوط به اعمال شرایط مرزی، مقادیر بقایی و همچنین فشار در میانه اضلاع مرزی تعیین و در آرایه مربوطه ذخیره می گردد. این کار قبل از شروع حلقه تکرار مربوط به گام زمانی انجام می گیرد تا مقادیر مربوط به شرایط مرزی مقداردهی اولیه شوند و همچنین از این مقادیر برای محاسبه گام زمانی در بخش های بعدی استفاده گردد. همانگونه که قبلا گفته شد تعیین شرایط مرزی در فایل شبکه توسط کاربر انجام می شود. شرایط مرزی تهیه برای این برنامه در جدول (1) آورده شده است که برای آشنایی بیشتر می توانید به مستندات زیربرنامه های مربوطه مراجعه نمایید.

1. پیشروی در زمان در یک حلقه تکرار

دریک حلقه معادلات در ریزترین شبکه تا دو تکرار حل می شوند تا حدس اولیه بهبود داده شده و به عنوان مقدار اولیه جدید برای شبکه درشت مورد استفاده قرار گیرد.

1. مقداردهی به برخی آرایه های بکار رفته در روش رانگ-کوتا

همانگونه که گفته شد در روش رانگ-کوتا از مقادیر گام زمانی قبل استفاده می شود که لازم است قبل از شروع حلقه تکرار مربوط به روش رانگ-کوتا این مقادیر مقداردهی شوند. همچنین این مقادیر برای محاسبه باقیمانده معادله جرم یا هر معادله دیگری می تواند بکار رود.

1. تعیین گام زمانی

با فراخوانی زیربرنامه TimSTP\_Inviscidگام زمانی هر کدام از سلول های شبکه محاسبه می گردد. در اینجا از گام زمانی متغیر استفاده شده است به این معنی که هر سلول با گام زمانی مربوط به خود در زمان پیشروی خواهد کرد تا سرعت همگرایی جهت بدست آوردن حل پایدار سریعتر باشد. کاربر می تواند پس از اجرای این زیربرنامه کوچکترین گام زمانی را بعنوان گام زمانی تمام سلول ها انتخاب نماید تا به این ترتیب بتواند حل ناپایا را بدست آورد.

1. حل معادلات در حلقه مربوط به روش رانگ-کوتا

در یک حلقه به تعداد مراحل رانگ-کوتا معادلات حل خواهند شد.

1. محاسبه ضرائب روش رانگ-کوتا

با توجه به رابطه ‏0 ضریب هر کدام از مراحل رانگ-کوتا محاسبه می شود و در یک پارامتر محلی ذخیره می گردد.

1. محاسبه بخش جابجایی

همانگونه که قبلا اشاره شد، بخش جابجایی بصورت بالادست و با استفاده از روش AUSM گسسته سازی شده است که در اینجا با فراخوانی زیربرنامه ConMeanFlow\_AUSM این بخش محاسبه می گردد.

1. محاسبه مقادیر بقایی و فشارتمام سلول های شبکه

در یک حلقه تکرار بر روی تمام سلول های شبکه مقادیر بقایی تمام سلول های شبکه با توجه به روش گسسته سازی صریح رانگ کوتا محاسبه می گردد. سپس مقدار فشار محاسبه می گردد.

1. تعیین شرایط مرزی

با فراخوانی زیربرنامه های مربوط به اعمال شرط مرزی مقادیر بقایی و همچنین فشار در میانه اضلاع مرزی تعیین و در آرایه مربوطه ذخیره می گردد تا در مراحل بعدی از آنها استفاده گردد.

1. ذخیره نتایج

در یک آرایه جدید مقادیر میدان ذخیره شده تا در انتهای چرخه V دوباره مورد استفاده قرار گیرد.

1. محاسبه بخش جابه جایی

بخش جابه جایی با مقادیر جدید میدان جهت انتقال باقی مانده (Residual) به شبکه درشت محاسبه می شود.

1. ذخیره باقی مانده

با استفاده از یک حلقه باقی مانده های شبکه ریز جهت انتقال به شبکه درشت در یک آرایه ذخیره می شوند.

1. درون یابی از شبکه ریز به شبکه درشت

با توجه به اینکه در چه مرحلی قرار داریم با توجه به مقدار NOL انتقال داده ها از شبکه ریز به شبکه درشت بعدی انجام می شود. NOL=1 بیانگر این است که داده ها از اولین شبکه ریز به اولین شبکه درشت انجام می شود.

1. خواندن پارامترهای لازم برای حل معادلات

با توجه به اینکه در کدام شبکه درشت قرار داریم بعد از مقداردهی NMF اطلاعات مربوط به شبکه مورد نظر با استفاده از زیربرنامه Read\_2DMesh فراخوانی می شود.

1. همانند قسمت 3 می باشد.
2. همانند قسمت 4 می باشد.
3. همانند قسمت 6 می باشد.
4. همانند قسمت 7 می باشد.
5. همانند قسمت 17 می باشد.
6. ذخیره کردن مقادیر باقی مانده در یک آرایه

باقی مانده های شبکه درشت جدید جهت محاسبه تابع تاثیر در یک آرایه جدید ذخیره می شود.

1. محاسبه تابع تاثیر (Forcing Function)

تابع تاثیر با استفاده از باقی مانده های منتقل شده از شبکه ریز و باقی مانده های بدست آمده در شبکه درشت محاسبه می شود.

1. ذخیره مقادیر میدان منتقل شده از شبکه ریز

با توجه به اینکه شبکه درشت حاضر، درشت ترین شبکه می باشد، مقادیر میدان منتقل شده از شبکه ریز قبلی در آرایه ای ذخیره شده تا بعد از بدست آوردن مقادیر دقیق میدان در این شبکه، مقادیر خطا را برای هر سلول محاسبه کنیم.

1. مقداردهی اولیه به برخی پارامتر ها

در اینجا از مقدار باقیمانده معادله جرم برای اطمینان از همگرایی استفاده شده است. بنابراین پروسه حل تا زمانی ادامه می یابد که مقدار باقیمانده معادله جرم از یک مقدار تعیین شده توسط کاربر بزرگتر باشد. یک شمارنده وجود دارد که تعداد گام های زمانی را شمارش می کند که در اینجا لازم است این مقدار برابر صفر قرار داده شود.

1. پیشروی در زمان در یک حلقه تکرار

با توجه به اینکه شبکه حاضر درشت ترین شبکه می باشد، در یک حلقه تکرار تا ارضا شدن شرط همگرایی، حل معادلات انجام می شود تا مقادیر دقیق میدان بدست آید. در اینجا از مقدار باقیمانده معادله جرم برای اطمینان از همگرایی استفاده شده است. بنابراین پروسه حل تا زمانی ادامه می یابد که مقدار باقیمانده معادله جرم از یک مقدار تعیین شده توسط کاربر بزرگتر باشد. توجه شود که در اینجا حل حالت پایدار مورد نظر می باشد.

1. بروز رسانی تعداد گام های زمانی

با شروع اجرای حلقه تکرار یک واحد به پارامتر نشاندهنده تعداد گام های زمانی اضافه می گردد.

1. مقداردهی به برخی آرایه های بکار رفته در روش رانگ-کوتا

همانگونه که گفته شد در روش رانگ-کوتا از مقادیر گام زمانی قبل استفاده می شود که لازم است قبل از شروع حلقه تکرار مربوط به روش رانگ-کوتا این مقادیر مقداردهی شوند. همچنین این مقادیر برای محاسبه باقیمانده معادله جرم یا هر معادله دیگری می تواند بکار رود.

1. تعیین گام زمانی

با فراخوانی زیربرنامه TimSTP\_Inviscidگام زمانی هر کدام از سلول های شبکه محاسبه می گردد. در اینجا از گام زمانی متغیر استفاده شده است به این معنی که هر سلول با گام زمانی مربوط به خود در زمان پیشروی خواهد کرد تا سرعت همگرایی جهت بدست آوردن حل پایدار سریعتر باشد. کاربر می تواند پس از اجرای این زیربرنامه کوچکترین گام زمانی را بعنوان گام زمانی تمام سلول ها انتخاب نماید تا به این ترتیب بتواند حل ناپایا را بدست آورد.

1. حل معادلات در حلقه مربوط به روش رانگ-کوتا

در یک حلقه به تعداد مراحل رانگ-کوتا معادلات حل خواهند شد.

1. محاسبه ضرائب روش رانگ-کوتا

با توجه به رابطه ‏0 ضریب هر کدام از مراحل رانگ-کوتا محاسبه می شود و در یک پارامتر محلی ذخیره می گردد.

1. محاسبه بخش جابجایی

همانگونه که قبلا اشاره شد، بخش جابجایی بصورت بالادست و با استفاده از روش AUSM گسسته سازی شده است که در اینجا با فراخوانی زیربرنامه ConMeanFlow\_AUSM این بخش محاسبه می گردد.

1. محاسبه مقادیر بقایی و فشارتمام سلول های شبکه

در یک حلقه تکرار بر روی تمام سلول های شبکه مقادیر بقایی تمام سلول های شبکه با توجه به روش گسسته سازی صریح رانگ کوتا محاسبه می گردد. سپس مقدار فشار با استفاده از رابطه ‏0 محاسبه می گردد.

1. تعیین شرایط مرزی

با فراخوانی زیربرنامه های مربوط به اعمال شرط مرزی مقادیر بقایی و همچنین فشار در میانه اضلاع مرزی تعیین و در آرایه مربوطه ذخیره می گردد تا در مراحل بعدی از آنها استفاده گردد.

1. محاسبه باقیمانده های معادله جرم

با فراخوانی زیربرنامه ResMass1 مقدار باقیمانده معادله جرم محاسبه می گردد.

1. محاسبه خطا

با توجه به اینکه در درشت ترین شبکه قرار داریم مقدار دقیق خطا با استفاده از مقادیر میدان دقیق بدست آمده محاسبه می شود.

1. درون یابی از شبکه درشت به شبکه ریز

با توجه به اینکه در چه مرحلی قرار داریم با توجه به مقدار NOL خطای بدست آمده در شبکه درشت به شبکه ریز منتقل می شود. NOL=1 بیانگر این است که انتقال از اولین شبکه درشت به اولین شبکه ریز انجام می شود.

1. خواندن پارامترهای لازم برای حل معادلات

با توجه به اینکه در کدام شبکه ریز قرار داریم بعد از مقداردهی NMF اطلاعات مربوط به شبکه مورد نظر با استفاده از زیربرنامه Read\_2DMesh فراخوانی می شود.

1. همانند قسمت 3 می باشد.
2. همانند قسمت 4 می باشد.
3. تصحیح مقادیر میدان در شبکه ریز

با استفاده از خطای منتقل شده از شبکه درشت مقادیر میدان که قبل ذخیره شده بود بهبود داده می شود.

1. همانند قسمت 6 می باشد.
2. همانند قسمت 7 می باشد.
3. تمامی مراحل این قسمت همانند قسمت های 29 تا 39 می باشد.
4. چاپ نتایج

در تکرار های خاصی نتایج حل جریان در فایل های مربوطه چاپ خواهد شد که این مقدار توسط کاربر تعیین و توسط فایل ورودی به برنامه معرفی می شود. برای آگاهی از فایل های خروجی می توانید به زیربرنامه Write\_ResultsV1 مراجعه فرمایید.

* 1. زیربرنامه Read\_2DMesh

به سند MC2F003F1 مراجعه شود.

* 1. زیربرنامه Read\_SettingV1

به سند MC2F001F1 مراجعه شود.

* 1. زیربرنامه MeshBC

به سند MC2F022F1 مراجعه شود.

* 1. زیربرنامه GeoCal2D

به سند MC2F026F1 مراجعه شود.

* 1. زیربرنامه InitMeanFlow\_Inviscid

به سند MC2F010F1 مراجعه شود.

* 1. زیربرنامه BC\_Wall

به سند MC2F011F1 مراجعه شود.

* 1. زیربرنامهBC\_Riemann

به سند MC2F015F1 مراجعه شود.

* 1. زیربرنامهBC\_InFlow

به سند MC2F017F1 مراجعه شود.

* 1. زیربرنامهBC\_SubOutFlow

به سند MC2F016F1 مراجعه شود.

* 1. زیربرنامهBC\_Symmetry

به سند MC2F018F1 مراجعه شود.

* 1. زیربرنامهTimSTP\_Inviscid

به سند MC2F012F1 مراجعه شود.

* 1. زیربرنامهConMeanFlow\_AUSM

به سند MC2F008F1 مراجعه شود.

* 1. زیربرنامهInterpolation

به سند [MC2F100F1](CODE/DOC/MC2F100F1_Interpolation.docx) مراجعه شود.

* 1. زیربرنامه Read\_2DMesh1

به سند مراجعه شود.

* 1. زیربرنامه Read\_2DMesh2

به سند مراجعه شود.

* 1. زیربرنامه Edge\_To\_Cell

به سند مراجعه شود.

* 1. زیربرنامهResMass

به سند MC2F004F1 مراجعه شود.

* 1. زیربرنامهWrite\_ResultsV1

به سند MC2F002F1 مراجعه شود.

مراجع

[1] R.P.Fedorenko , USSR comput.math phys.1 1962

[2] R.P.Fedorenko , USSR comput.math phys.4 1964

[3] N.S Bakhvalov , USSR comput.Math.phys.6 1966

[4] A.Brandt,J.Math.comput.13 1972

[5] A.Brandt,J.Math.comput.31 1977

[6] A.Jameson,”*Acceleration of transonic potential flow calculations on arbitrary meshes by the*

*multiple grid method*,” AIAA paper 79-1458,1979

[7] Mavriplis, D. and A. Jameson. *Multigrid Solution of the Two-dimensional Euler Equations on Unstructured Triangular Meshes*; AIAA J.28(8), 1415-1425 (1990)

[8] Jameson, A. and Mavriplis, D. J. , “*Finite Volume Solution of the Two Dimensional Euler Equation on a Regular Triangular Mesh*”, AIAA Journal, Vol. 24, NO. 4, 1986, pp. 611-618

[9] D. J. Mavriplis,*Multigrid techniques for unstructured meshes*, ICASE Report No. 95-27, NASA

Langley research center, 1995. Also available as Lecture notes for the 26th CFD Lecture

Series of von Karman Institute; AGARD Publication

[10] Muzaferija, S. *Adaptive* *Finite Volume Method for Flow Prediction Using Unstructured Meshes and Multigrid Approach*, PhD Thesis, Imperial College of Science, Technology, and Medicine, London, 1994

[11] Mavriplis, D. and A. Jameson. *Multigrid Solution of the Two-dimensional Euler Equations on Unstructured Triangular Meshes*; AIAA-87-0353 (1987)

[12] Mavriplis, D.J., Jameson, A.; Martinelli, L.: *Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes*. ICASE Report,NO. 89-11, 1989

[13] Riemslagh, K.; Dick, E.: *A Multigrid Method for Steady Euler Equations on Unstructured Adaptive Grids*. Proc. 6th Copper Mountain Conf. on Multigrid Methods, 1993, pp. 527-542

[14] C Rodrigo, FJ Gaspar, CW Oosterlee, I Yavneh SIAM,” [*Accuracy measures and Fourier analysis for the full multigrid algorithm*](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=c61zP0UAAAAJ&citation_for_view=c61zP0UAAAAJ:9yKSN-GCB0IC)”, Journal on Scientific Computing 32 (5), 3108-3129

[15] M. Dumett, P. Vassilevski, C.S. Woodward, *A multigrid method for nonlinear unstructured finite element elliptic equations*, SIAM J. Sci. Comput. 2003, submitted.

[16] Adams, M. F., “*A low memory, highly concurrent multigrid algorithm*,” arXiv preprint arXiv:1207.6720, 2012.

[17] Jameson A. , *Solution of the Euler equations for two dimensional transonic flow by a multigrid method*, Appl. Math. Comp. 13 (1983), 327-355

[18] Miller, G.L., Talmor, D.,Teng,S.-H,: *Optimal coarsening of unstructured meshes*. J.Alg.31(1), 29-65 (1999)

[19] Carl Ollivier-Gooch, *Coarsening unstructured meshes by edge contraction*, Int. J. Numer. Meth. Engng., 57 (2003), pp. 391{414}.

[20] Reuter, B., *Coarsening of Simplicial Meshes for Large Scale Parallel FEM Computations with DOLFIN HPC*, 2013

[21] K.stuben. Algebraic multigrid (AMG): *an introduction with application*. In U.Trottenberg, C.Oosterlee,and A.Schuller,editors,Multigrid. Academic Press,2001

[22] Butler and Jeffrey. *Improving Coarsening and Interpolation for Algebraic Multigrid*. PhD thesis, Department of Applied Mathematics, University of Waterloo,2006.

[23] Dafna Talmor. *Well-spaced points for numerical methods*. PhD thesis, Carnegie Mellon University, 1997

[24] کتاب دینامیک سیالات محاسباتی دکتر اصفهانیان

**پیوست 1: پارامترها و آرایه­های مورد استفاده در برنامه اصلی**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dimension** | **Type** | **Description** | **Parameter** |
|  | IntegerParameter | Maximum **Dim**ension of Arrays | Dim |
|  | Integer | **N**umber of **R**egions | NR |
|  | Integer | **N**umber of Existing **P**oints | NP |
|  | Integer | **N**umber of Existing **C**ells | NC |
|  | Integer | **N**umber of **F**aces Constructing Mesh | NF |
|  | Integer | Index of 1st Non-Boundary **F**aces | NF1 |
|  | Integer | Index of Last Non-Boundary **F**aces | NF2 |
|  | Integer | Index of 1st **F**aces on **W**all Boundary | NFW1 |
|  | Integer | Index of Last **F**aces on **W**all Boundary | NFW2 |
|  | Integer | Index of 1st **F**aces on **F**ar **F**ield Boundary | NFF1 |
|  | Integer | Index of Last **F**aces on **F**ar **F**ield Boundary | NFF2 |
|  | Integer | Index of 1st **F**aces on **I**nlet Boundary | NFI1 |
|  | Integer | Index of Last **F**aces on **I**nlet Boundary | NFI2 |
|  | Integer | Index of 1st **F**aces on **O**utflow Boundary | NFO1 |
|  | Integer | Index of Last **F**aces on **O**utflow Boundary | NFO2 |
|  | Integer | Index of 1st **F**aces on **S**ymmetry Boundary | NFS1 |
|  | Integer | Index of Last **F**aces on **S**ymmetry Boundary | NFS2 |
|  | Integer | Index of 1st **F**aces on **I**nter**F**sce Boundary | NFIF1 |
|  | Integer | Index of Last **F**aces on **I**nter**F**sce Boundary | NFIF2 |
|  | Integer | **N**umber of **R**unge **K**utta **S**tages | NRKS |
|  | Integer | **N**umber of Cycle to **Write** Results | NWrite |
|  | Integer | **Init**ialize from Available Data or Infinite Flows | Init |
|  | Integer | **N**umber of **Cyc**le | Ncyc |
|  | Integer | **N**umber **M**esh **F**ile | NMF |
|  | Integer | **N**umber **O**f  **C**ycle | NOL |
|  | Integer | **F**ine **t**o **C**oarse | FTC |
|  | Integer | **N**umber **M**esh **F**ile | NMF |
|  | Real(8) | **G**ama Constant (Specific Heat Ratio) | GM |
|  | Real(8)-Degree | Infinite Flow Angle to X Axis | ALF |
|  | Real(8) | Density of Infinite Flow | R0 |
|  | Real(8) | **P**ressure of Infinite Flow | P0 |
|  | Real(8) | Sound Speed of Infinite Flow | C0 |
|  | Real(8) | Infinite Flow Velocity in X Direction | U0 |
|  | Real(8) | Infinite Flow Velocity in Y Direction | V0 |
|  | Real(8) | **M**aximum **Er**ror in E**x**plicit Approach | ERmx |
|  | Real(8) | **M**aximum **Er**ror in E**x**plicit Approach in coarsest mesh | ERmx1 |
|  | Real(8) | Currant Number for E**x**plicit Methods | CFLx |
|  | Real(8) | **M**uch Number of **inf**inite Flow | Minf |
|  | Real(8)-Kelvin | **T**otal **T**emprature | Tt |
|  | Real(8) | **R**esidual of **M**ass Equation | Rm |
| (1:100) | Integer | **N**umber of **F**aces Belong to each **R**egions | NFR |
| (1:100) | Integer | **B**oundary **C**ondition | BC |
| (1:4,1:Dim) | Integer | **I**nformation of Grid **D**ata **S**tructure | IDS |
| (1:4,1:Dim) | Real(8) | Conservative Values at (N+1)th Time Step | WNP1 |
| (1:4,1: Dim) | Real(8) | Conservative Values at (N)th Time Step | Wn |
| (1:4,1:Dim) | Real(8) | **Con**vection Term of Mean flow Equations | Con |
| (1:4,1:Dim) | Real(8) | **S**olution in **M**esh**1** | SolM1 |
| (1:4,1:Dim) | Real(8) | **F**orcing **F**unction **M**esh**2** | FFM2 |
| (1:4,1:Dim) | Real(8) | **Res**idual | RES |
| (1:4,1:Dim) | Real(8) | **Res**idual in **M**esh**2** | ResM2 |
| (1:4,1:Dim) | Real(8) | **Error** in coarsest mesh | Error |
| (1:5,1:Dim) | Real(8) | Conservative Values and Pressure at **B**oundary Faces | WB |
| (1:Dim) | Real(8) | Coordinate of Points | X,Y |
| (1:Dim) | Real(8) | Coordinate of Element’s Center | Xc,Yc |
| (1:Dim) | Real(8) | Normal Vectors of each Face | NX,NY |
| (1:Dim) | Real(8) | **A**rea of each cell | A |
| (1:Dim) | Real(8) | Length of each Face | DA |
| (1:Dim) | Real(8) | Explicit Time Step | DT |
| (1:Dim) | Real(8) | **P**ressure | P |

1. Jacobi [↑](#footnote-ref-1)
2. Gauss-Seidel [↑](#footnote-ref-2)
3. Successive Over Relaxation [↑](#footnote-ref-3)
4. Alternating Direction Implicit [↑](#footnote-ref-4)
5. Multigrid Method [↑](#footnote-ref-5)
6. Fine grid [↑](#footnote-ref-6)
7. Coarse grid [↑](#footnote-ref-7)
8. Fedorenko [↑](#footnote-ref-8)
9. Bakhvalov [↑](#footnote-ref-9)
10. Brandt [↑](#footnote-ref-10)
11. Jameson [↑](#footnote-ref-11)
12. Mavriplis [↑](#footnote-ref-12)
13. Cell-Vertex [↑](#footnote-ref-13)
14. Cell-Center [↑](#footnote-ref-14)
15. درون یابی با استفاده از مساحت های وزن دار [↑](#footnote-ref-15)
16. Samir Muzaferija [↑](#footnote-ref-16)
17. Erik dick [↑](#footnote-ref-17)
18. Riemslagh [↑](#footnote-ref-18)
19. Coarsening [↑](#footnote-ref-19)
20. Full single grid [↑](#footnote-ref-20)
21. Coarse grid correction [↑](#footnote-ref-21)
22. Interpolation [↑](#footnote-ref-22)
23. Residual [↑](#footnote-ref-23)
24. V-cycle [↑](#footnote-ref-24)
25. W-cycle [↑](#footnote-ref-25)
26. F-cycle [↑](#footnote-ref-26)
27. FMG-cycle [↑](#footnote-ref-27)
28. Restriction [↑](#footnote-ref-28)
29. Prolongation [↑](#footnote-ref-29)
30. Forcing Function [↑](#footnote-ref-30)
31. Geometric multigrid [↑](#footnote-ref-31)
32. Algebraic multigrid [↑](#footnote-ref-32)
33. Nested [↑](#footnote-ref-33)
34. Spacing Function [↑](#footnote-ref-34)
35. Injection [↑](#footnote-ref-35)
36. Linear interpolation [↑](#footnote-ref-36)