



زیربرنامه **RBF\_Moving\_Mesh\_Greedy\_Method\_3D**

|  |  |  |
| --- | --- | --- |
| **توسعه دهندگان:** | **مرتضی نامور** |  |
| **محمد امین ذوالجناحی** |  |
| **تهیه کننده مستند:** | **مرتضی نامور، محمد امین ذوالجناحی** | |
| **تاریخ تنظیم سند:** | **07/07/1395** | |
| **تایید کنندگان:** |  | |
| **شماره سند:** | **MC5F031F1** | |
| **زبان برنامه نویسی:** | **Fortran 90** | |

|  |  |  |  |
| --- | --- | --- | --- |
| **RBF\_Moving\_Mesh\_greedy\_method\_3D(Istp,Dim,NBP,NP,IBP,X,Y,Z,DelX,DelY,DelZ,NSBP,ACTV)** | | | |
| **Dimension** | **Type** | **Description** | **Intent** |
|  |  |  | **Intput** |
|  | Integer | Maximum **Dim**ension of Arrays | Dim |
|  | Integer | Iteration countor | Istp |
| (1:Dim) | Integer | **I**ndex of **B**oundary **P**oints | IBP |
|  | Integer | **N**umber of **B**oundary **P**oints | NBP |
|  | Real(8) | **N**umber of **P**oints | NP |
| (1:Dim) | Real(8) | Coordinate of Points | X,Y,Z |
|  |  |  | **InOut** |
| (1:Dim) | Real(8) | Displacement in x and y ,z Direction | DelX,DelY,DelZ |
|  |  |  | **Outlet** |
|  | Integer | **N**umber of **S**elected **B**oundary **P**oints | NSBP |
| (1:NBP) | Integer | Active list | ACTV |

* 1. وظایف

در این زیربرنامه جابجایی تمام نقاط داخلی به علاوه تعدادی از نقاط سطحی بوسیله نقاط سطحی منتخب و ضرایب محاسبه شده ی آنها با استفاده از الگوریتم First Greedy1 یا First Greedy 2 یا hybrid 1 یا hybrid 2بدست می آید.

* 1. توضیحات و تئوری­ها

با توجه به روش توابع پایه ای شعاعی که بر مبنای اجرای درونیابی برای تمام نقاط با استفاده از تمام نقاط سطحی در مسائل سه بعدی می توان به این نتیجه رسیدکه اگرچه موقعیت جدید تولید شده ی نقاط شبکه از دقت بسیار بالایی برخوردار است، اما هزینه محاسباتی روش کامل بسیار زیاد است مخصوصا در شبکه های سه بعدی که از تعداد سلول های بسیار زیادی تشکیل شده اند. زمان محاسبه و حافظه ی مورد نیاز برای این نوع جابجایی ها حائز اهمیت بوده و برای روش پایه با مقیاس های هزینه و ارزیابی می­شوند و این در حالیست که شبکه ی کلی با مقیاس هزینه جابجا می شود. بنابر این هدف این پروژه این است که علی رغم افزایش کیفیت حرکت شبکه ی موجود ، نیازمندی برنامه به حافظه و زمان حل کمتر شود و این مقادیر به میزان قابل مدیریت تقلیل یابند. با در نظر گرفتن اینکه چگونه به این هدف دست یابیم، مشاهده می شود که فاکتور اصلی در هزینه­ی روش مقدار است که به طور جامعی بر هزینه های حل و به روز رسانی درونیابی تاثیر می گذارد. دو فلسفه ی مجزا برای روشن کردن روش موجود است: یکی به دنبال حل دقیق برای یک مساله ی تقریبی در حالی که دیگری به دنبال یک حل تقریبی برای یک مساله ی دقیق است. روش Greedy [1]–[3] در دسته دوم جای می گیرد. این درحالیست که بیشتر تکنیک های RBF مثل ناحیه ی واحد[4]–[6] ، چند مرحله ای[7] روش های فرافکنی تناوبی[4] در زمره­ی دومین دسته جای می گیرند، اگر چه همپوشانی برای این روش ها زمانی که با هم ادغام شوند وجود دارد[2]. برای حرکت شبکه روش ناحیه ی واحد، که اساس کارش تشکیل درونیابی های محلی و بعد سرهم کردن این نواحی بوسیله ی یک تابع وزنی است، کمتر مورد توجه قرار می گیرد چون ناپیوستگی ها یا افت های نرم کردن شبکه می تواند در نواحی مختلف موجود باشد و درست در قسمت هایی که درونیابی ها همپوشانی می کنند اتفاق می افتند. روش های چند مرحله ای و فرافکنی ناوبی نیز جزو موثر ترین روش ها در سرعت رده­ی تحیلی قرار می گیرند، اما ممکن است فقط بر هزینه های بروز رسانی در حد بسیار کم تاثیر بگذارند. یک روش کارآمد به این صورت است که از تمامی نقاط سطحی برای تعریف حرکت نقاط حجمی استفاده نشود بلکه مجموعه ای از نقاط منتخب سطحی که با استفاده از آنها بتوان حرکت شبکه را تبعیین کرد مورد نیاز است که در عین حال دقت کافی داشته باشد. جاکوبسون و همکارانش[8] از این انتخاب برای نازک کردن ابر نقاط سطحی بدون در نظر گرفتن دقت حل استفاده کردند. اگرچه این احتمال هم وجود دارد که روشی باشد که هم هزینه و همزمان دقت را به بهینه ترین حالت برساند. اینگونه روش اخیرا برای روش درونیابی RBF با راهکاری به نام Greedy algorithms نشان داده شده است که بتوان از آن به نحو احسن بهره وری لازم را به عمل آورد[9]–[11]. اگر تنها مجموعه ی تقلیل یافته از نقاط روی سطوح ( نقاط کنترلی) حرکت را تعریف کنند و هیچ چیزی در مجموعه ی کار دخیل نباشد، آنگاه هر نقطه ی روی سطح که جزو نقاط کنترلی نیستند، بوسیله ی عملیات درونیابی جابجا می شود. این به این معنی است که روشی باید باشد که بوسیله آن سطوح شبکه حجمی تابع جابجایی نقاط کنترلی است وتنها زمانی امکان پذیر است که خطاها در این نقاط منتخب کمتر از یک حد مطلوب باشد که در نهایت باید با به کار گیری یک انتخاب مناسب برای نقاط کنترلی همراه است. متناوبا نقاط نادرست سطحی می توانند به طور دقیق تصحیح شوند، اگر کیفیت شبکه در اولین لایه از سلول ها دور از سطوح از بین نرود. همچنین روند تصحیح می تواند در چند لایه ی سلولی خارج از سطح با استفاده از یک تابع تجزیه انجام شودکه باعث افزایش کیفیت شبکه در این ناحیه میگردد. اینکه دقیقا روش Greedy چطور می تواند داخل یک کد CFD جای بگیرد، بستگی به این دارد که آیا داخل حلقه ی زمان قرار بگیرد یا خارج از آن. اگر داخل جای بگیرد آنگاه در هر گام زمانی یا در هر n گام زمانی می تواند اجرا شود. تا از تغییر شکل واقعی سطوح استفاده کند و خطاها را به حداقل برساند. اما با قرار گیری این روش خارج از حلقه تکرار می تواند مناسب باشد اگر نقاط انتخابی در طول روند تغییر شبکه ثابت در نظر گرفته شوند. اگر نقاط به این طریق انتخاب شوند آنگاه این امکان نیز هست که یک مجموعه را برای افزایش سرعت حل ذخیره کرد.

* + 1. اجرا در برنامه

برای وضوح بیشتر نحوه اجرای روش Greedy در یک کد سعی می‌گردد تا آنچه که تاکنون به عنوان مفاهیم ریاضی روش بیان شده است، به صورت جزئی تر بیان گردد تا متن وضوح بیشتری داشته باشد.

در مسائل دینامیک سیالات محاسباتی، دامنه محاسباتی به دو بخش نواحی داخلی و مرزی تقسیم می‌گردند. روش‌های اصلاح شبکه برپایه‌ی تغییرات مختصات نقاط مرزی قرار گرفته اند، به عبارت دیگر در همه‌ی این روش‌ها باید در ابتدا جابجایی نقاط مرزی به عنوان یک معلوم تعیین گردد و در گام بعد جابجایی نقاط داخلی محاسبه می‌شوند. در گام اول یکی از چهار روش نام برده First Greedy1 و First Greedy2 و hybrid Greedy1 و hybrid Greedy2 برا محاسبه نقاط کنترلی و ضرایب مربوطه بکار گرفته می شود و البته حائز اهمیت است که ذکر شود این روش ها تنها در اولین گام زمانی و اولین گام جابجایی مرز ها نقاط کنترلی را انتخاب نموده و لیستی از این نقاط را به همراه ضرایب درونیابی ارائه می دهند که در گام های بعدی از آن نقاط استفاده شود.

بردار جابجایی نقاط مرزی  شامل مقادیر معلوم جابجایی برای نقاط گسسته می‌باشد. تابع میانیابی ، قابلیت محاسبه‌ی جابجایی نقاط (اعم از داخلی و مرزی) را دارا می‌باشد. بنابراین همان طور که بیان شد: . در اینجا  همان نقاط انتخابی یکی از الگوریتم هاست.

اگر تمام نقاط مرزی غیر از نقاط کنترلی و همچنین نقاط داخلی شبکه را با *m* نامگذاری کنیم، بر اساس معادله ‏(1) وجود دارد:

1. 

که در این رابطه *NSBP* تعداد نقاط انتخاب شده و  ضرایب آن نقاط انتخابی و  همان مکان نقاط انتخابی یکی از الگوریتم هاست.

بر اساس معادله ‏(1)، اگر تعداد نقاط مرزی برابر باشد با  پس به تعداد  معادله وجود دارد:

1. 

باید توجه شود که  ، به اشاره دارد.

در معادله ‏(2) مجهولات عبارتند از بردار ضرایب  که عبارت است از  و  عضو دارد.

حال دستگاه معادلاتی به شکل رابطه ‏(3) حاصل می‌گردد:

1. 

که:

1. 
2. 
3. 
   1. پیاده سازی

بطور خلاصه می توان گفت که در روش Greedy هدف به دست آوردن ضرایب نقاط منتخب در تابع میانیابی  می‌باشد که باید ضرایب مجهول آن یعنی  مشخص شوند. این ضرایب با حل شدن دستگاه معادلات  بدست می آیند که  به ماتریس ضرایب اختصاص دارد.

1. بدست آمدن نقاط کنترلی منتخب به همراه ضرایب

در این قسمت نقاط کنترلی منتخب بوسیله ی یکی از الگوریتم های بهینه سازی روش RBF بدست می­آیند.

1. تعیین ماتریس مقادیر *c*

در این قسمت ماتریس مقادیر با جایگذاری جابجایی دقیق نقاط کنترلی ایجاد می شود. و این قسمت تنها برای گام های زمانی بالاتر از 1 فعال می شوند چرا که نقاط کنترلی در قسمت 1 توسط الگوریتم های Greedy تعیین شده و در گام های بعد از آنها استفاده می شود.

1. محاسبه‌ی ماتریس ضرایب *B*

ماتریس ضرایب نیز با توجه به رابطه ‏(4) و با فراخوانی زیربرنامه مربوطه مشخص می گردد.

1. حل دستگاه معادلات 

برای حل دستگاه‌های معادلات، نیاز به حل‌گرهای خاص ماتریس است. در این زیر برنامه با توجه به شرایط ماتریس از روش LU استفاده شده است. ماتریس ضرایب  برای جابجایی ها در هر سه جهت x و yو z یکسان هستند. زیرا این ماتریس بر اساس توابع میانیابی ایجاد شده اند که تنها تابعی از فاصله‌ی بین نقاط مرزی هستند. در این قسمت ضرایب نقاط منتخب برای گام های زمانی بالاتر از 1 تعیین می شوند.

1. محاسبه‌ی جابجایی نقاط

بعد از محاسبه‌ی ضرایب منتخب و بدست آوردن تابع میانیابی  محاسبه‌ی جابجایی نقاط سطحی غیرمنتخب و نقاط حجمی با استفاده از زیربرنامه مربوط به اینکار صورت می‌پذیرد.

**مراجع**

[1] R. Schaback and H. Wendland, “Numerical techniques based on radial basis functions,” 2000.

[2] Y. Ohtake, A. Belyaev, and H. Seidel, “Multi-scale and adaptive cs-rbfs for shape reconstruction from clouds of points,” *Adv. Multiresolution*, 2005.

[3] R. Schaback and H. Wendland, “Adaptive greedy techniques for approximate solution of large RBF systems,” *Numer. Algorithms*, 2000.

[4] H. Wendland, “Scattered data approximation,” 2004.

[5] R. Ahrem, A. Beckert, and H. Wendland, “A meshless spatial coupling scheme for large-scale fluid-structure-interaction problems,” *Comput. Model.*, 2006.

[6] H. Wendland, “Fast evaluation of radial basis functions: Methods based on partition of unity,” *Approx. Theory X Wavelets, Splines,* 2002.

[7] A. Iske and J. Levesley, “Multilevel scattered data approximation by adaptive domain decomposition,” *Numer. Algorithms*, 2005.

[8] S. Jakobsson and O. Amoignon, “Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization,” *Comput. Fluids*, 2007.

[9] J. Carr, R. Beatson, J. Cherrie, and T. Mitchell, “Reconstruction and representation of 3D objects with radial basis functions,” *Proc. 28th*, 2001.

[10] S. De Marchi, R. Schaback, and H. Wendland, “Near-optimal data-independent point locations for radial basis function interpolation,” *Adv. Comput.*, 2005.

[11] S. De Marchi, “On optimal locations for radial basis function interpolation: computational aspects,” *Rend. Del Semin. Mat.*, 2003.