

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

ما رميت اذ رميت ولكن الله رمي



دانشکده معدن، نفت و ژئوفیزیک

گروه استخراج

تهیه برنامه کامپیوتری تعیین هندسه بلوک‌های توده‌سنگ درزه‌دار در حالت دو

بعدی

دانشجو: رضا یاراحمدی

اساتید راهنما:

پروفسور رضا کاکائی

دکتر حسین میرزائی نصیرآباد

استاد مشاور:

دکتر راحب باقرپور

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

شهریور ۱۳۹۱



دانشگاه صنعتی شاهرود

مدیریت تحصیلات تکمیلی

فرم شماره (۶)

بسمه تعالی

شماره :

تاریخ ۱۳۹۷/۵/۲۵
ویرایش :

فرم صورتجلسه دفاع از پایان نامه تحصیلی دوره کارشناسی ارشد

با تأییدات خداوند متعال و با استعانت از حضرت ولی عصر (عج) ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای - رضا یاراحمدی رشته معدن گرایش استخراج تحت عنوان تهیه برنامه کامپیوتری تعیین حجم و شکل بلوک های توده سنگ درزه دار در حالت دو بعدی که در تاریخ ۱۳۹۱/۶/۲۶ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

قبول (با درجه : کاملاً مناسب امتیاز ۱۸) ☐ دفاع مجدد ☐ مردود ☐

۲- بسیار خوب (۱۸ - ۱۸/۹۹)

۱- عالی (۲۰ - ۱۹)

۴- قابل قبول (۱۵/۹۹ - ۱۴)

۳- خوب (۱۷/۹۹ - ۱۶)

۵- نمره کمتر از ۱۴ غیر قابل قبول

امضاء	مرتبه علمی	نام و نام خانوادگی	عضو هیأت داوران
	استاد استادیار	پروفسور رضا کاکائی دکتر حسین میرزائی نصیرآباد	۱- استاد راهنما
	استادیار	دکتر راجب باقرپور	۲- استاد مشاور
	استادیار	دکتر محمد کاراموزیان	۳- نماینده شورای تحصیلات تکمیلی
	دانشیار	دکتر فرهنگ سرشکی	۴- استاد ممحن
	استاد	پروفسور محمد عطائی	۵- استاد ممحن

رئیس دانشکده :



تقدیم به

پدر و مادر مهربانم

و همسر عزیزم

تشکر و قدردانی

خداوند منان را شاکرم که به من توفیق بندگیش را داد. از جناب آقای دکتر رضا کاکائی، جناب آقای دکتر حسین میرزائی و جناب آقای دکتر راحب باقر پور که مرا در تهیه این تحقیق یاری نمودند، کمال تشکر را دارم و از خداوند موفقیت روز افزون را برای ایشان خواهانم. از تمام اساتید دانشکده معدن دانشگاه صنعتی شاهرود بخصوص آقایان دکتر عطائی، دکتر سرشکی، دکتر کاراموزیان، دکتر ترابی، دکتر جلالی، دکتر رمضانزاده و دکتر زارع که معلم بنده بوده‌اند، قدردانی می‌کنم انشاء الله در سایه حق سعادت‌مند باشند.

تعهد نامه

اینجانب **رضا یاراحمدی** دانشجوی دوره کارشناسی ارشد رشته مهندسی معدن دانشکده معدن، نفت و ژئوفیزیک دانشگاه صنعتی شاهرود نویسنده پایان نامه تهیه برنامه کامپیوتری تعیین هندسه بلوک‌های توده‌سنگ درزه-دار در حالت دو بعدی تحت راهنمایی پروفسور رضا کاکائی و دکتر حسین میرزائی نصیر آباد متعهد می‌شوم.

تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .

در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .

مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .

کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .

حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.

در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .

در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ ۱۳۹۱/۷/۲۳

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب ، برنامه‌های رایانه‌ای ، نرم‌افزارها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می‌باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده

در حوزه‌های مختلف مهندسی سنگ نظیر طراحی سیستم نگهداری فضاها و زیرزمینی احداث شده در توده‌سنگ درزه‌دار، طراحی الگوی آتشباری توده‌های سنگی، بهینه سازی خردایش سنگ، تعیین بلوک‌دهی سنگ‌های ساختمانی و ... اطلاع دقیق از هندسه توده سنگ ضروری می‌باشد. انواع ناپیوستگی‌ها محیط سنگی را به صورت مجموعه‌ای از بلوک‌های سنگی در می‌آورند. هندسه بلوک‌ها یعنی شکل و حجم بلوک‌ها در کاربردهای مختلف در رفتار توده‌سنگ مؤثر هستند. بنابراین اطلاع از توزیع حجمی و شکل بلوک‌ها ضروری می‌باشد. یکی از اهداف انجام این پایان‌نامه بررسی روش‌های موجود تعیین هندسه بلوک‌های توده‌سنگ و انتخاب الگوریتم مناسب جهت تهیه برنامه کامپیوتری است و هدف اصلی تهیه برنامه کامپیوتری تعیین هندسه بلوک‌های توده‌سنگ می‌باشد. در این پایان‌نامه با استفاده از الگوریتم ماتریسی دو بعدی، برنامه کامپیوتری تعیین هندسه توده‌سنگ در محیط نرم افزار MATLAB تهیه شده است. نتایج بدست آمده از برنامه کامپیوتری با توجه به اعتبار سنجی انجام شده صحیح می‌باشد. چند مثال کاربردی در زمینه سنگ ساختمانی، آتشباری، رقیق شدگی ماده معدنی در روش‌های استخراج تخریبی و پایداری بلوک‌های توده‌سنگ در معادن روباز و زیرزمینی با استفاده از این برنامه بررسی شده است.

کلمات کلیدی: توده‌سنگ درزه‌دار، بلوک، برنامه کامپیوتری، روش‌های تعیین حجم و هندسه توده‌سنگ

فهرست عناوین

- ۱- کلیات..... ۱
 - ۱-۱ مقدمه..... ۲
 - ۲-۱ ضرورت انجام مطالعه..... ۲
 - ۳-۱ پیشینه مطالعه..... ۵
 - ۴-۱ هدف از انجام مطالعه..... ۱۰
 - ۵-۱ ساختار پایان نامه..... ۱۱
- ۲- توده سنگ و تعیین هندسه بلوک‌ها توسط پارامترهای هندسی..... ۱۳
 - ۱-۲ مقدمه..... ۱۴
 - ۲-۲ توده سنگ درزه دار و اجزای آن..... ۱۴
 - ۳-۲ نوع و شکل مختلف بلوک‌های توده سنگ..... ۲۰
 - ۴-۲ توده سنگ و توزیع حجم بلوک‌های آن..... ۲۳
 - ۱-۴-۲ روش‌های شبیه سازی هندسی درزه‌های توده سنگ..... ۲۴
 - ۲-۴-۲ روش قطعی و محدود..... ۲۵
 - ۳-۴-۲ روش زمین آماری..... ۲۵
 - ۴-۴-۲ روش آماری..... ۲۶
 - ۵-۲ محاسبه حجم بلوک‌های توده سنگ با پارامترهای هندسی..... ۲۷
 - ۱-۵-۲ محاسبه سر زمین یا با استفاده از مغزه‌های حفاری..... ۲۸
 - ۲-۵-۲ استفاده از فاکتور حجمی بلوک (Ib)..... ۲۸
 - ۳-۵-۲ استفاده از فاصله‌داری درزه‌ها..... ۲۹
 - ۴-۵-۲ استفاده از فاکتور تعداد درزه حجمی (Jv)..... ۲۹
 - ۱-۴-۵-۲ محاسبه Jv بر اساس فاصله‌داری درزه‌ها..... ۲۹
 - ۲-۴-۵-۲ محاسبه Jv بر اساس فرکانس دو بعدی درزه..... ۳۰
 - ۳-۴-۵-۲ محاسبه Jv در امتداد خط برداشت یا مغزه حفاری..... ۳۰
 - ۴-۴-۵-۲ محاسبه Jv با استفاده از شاخص کیفیت سنگ (RQD)..... ۳۱
 - ۶-۲ نتیجه‌گیری..... ۳۳
- ۳- تعیین هندسه بلوک‌های توده سنگ..... ۳۴
 - ۱-۳ مقدمه..... ۳۵
 - ۲-۳ تعیین هندسه بلوک‌ها توسط الگوریتم‌های مختلف..... ۳۵
 - ۱-۲-۳ الگوریتم تئوری گرافی..... ۳۶
 - ۱-۱-۲-۳ تبدیل خطوط و محل برخورد آنها به گراف..... ۳۷
 - ۲-۱-۲-۳ یافتن چند ضلعی‌ها با استفاده از گراف‌ها..... ۳۹
 - ۳-۱-۲-۳ ساختن چند ضلعی‌ها..... ۴۰
 - ۲-۲-۳ الگوریتم مبتنی بر تکنیک‌های عددی..... ۴۱

۴۵.....	محاسبه تقاطع ناپیوستگی‌ها.....	۱-۲-۳
۴۹.....	مرتب‌سازی یال‌ها.....	۲-۲-۳
۵۱.....	عملگر مرزی در سیستم دو بعدی.....	۳-۲-۳
۵۳.....	قانون زاویه دست چپی.....	۴-۲-۳
۵۳.....	تشخیص بلوک‌ها.....	۵-۲-۳
۵۵.....	اتصال و آنالیزهای مکانیکی در بلوک‌ها.....	۶-۲-۳
۶۰.....	الگوریتم شی مبتنی بر گراف‌های جهت‌دار.....	۳-۲-۳
۶۱.....	محاسبه نقاط برخورد.....	۱-۳-۲-۳
۶۴.....	اتصال بین نقاط برخورد.....	۲-۳-۲-۳
۶۵.....	برش درختی.....	۳-۳-۲-۳
۶۵.....	شناسایی بلوک‌ها.....	۴-۳-۲-۳
۶۷.....	شناسایی بلوک‌های واقعی.....	۵-۳-۲-۳
۶۷.....	روش زاویه‌ای.....	۱-۵-۳-۲-۳
۶۸.....	روش نقطه‌ای.....	۲-۵-۳-۲-۳
۶۹.....	روش مساحت.....	۳-۵-۳-۲-۳
۶۹.....	یک مثال برای درک بهتر الگوریتم.....	۶-۳-۲-۳
۷۳.....	الگوریتم تئوری بلوک مبتنی بر روش‌های برداری.....	۴-۲-۳
۷۴.....	حجم، لبه‌ها و گوشه‌های بلوک Π وجهی.....	۱-۴-۲-۳
۷۸.....	الگوریتم مبتنی بر ماتریس‌ها.....	۵-۲-۳
۷۹.....	نتیجه‌گیری.....	۳-۳
۸۰.....	۴- برنامه کامپیوتری تعیین هندسه بلوک‌های توده‌سنگ.....	
۸۱.....	مقدمه.....	۱-۴
۸۱.....	بررسی الگوریتم‌های موجود و انتخاب الگوریتم مناسب.....	۲-۴
۸۳.....	تشریح کامل الگوریتم ماتریسی دو بعدی.....	۳-۴
۸۵.....	ورود هندسه ناپیوستگی‌ها، دسته درزه‌ها و ابعاد محدوده.....	۱-۳-۴
۸۷.....	محاسبه نقاط برخورد ناپیوستگی‌ها.....	۲-۳-۴
۸۹.....	یافتن یال‌ها و مرتب‌سازی آن‌ها.....	۳-۳-۴
۹۳.....	تشکیل ماتریس‌های اساسی.....	۴-۳-۴
۹۷.....	شناسایی بلوک‌ها.....	۵-۳-۴
۹۸.....	مثال ۱ برای تشریح مراحل الگوریتم ماتریسی.....	۶-۳-۴
۱۰۵.....	تهیه برنامه کامپیوتری.....	۴-۴
۱۰۶.....	نحوه تهیه برنامه کامپیوتری.....	۱-۴-۴
۱۰۸.....	ورودی برنامه کامپیوتری.....	۲-۴-۴
۱۰۹.....	خروجی برنامه کامپیوتری.....	۳-۴-۴
۱۱۲.....	مثال ۲: سنجش میزان دقت برنامه.....	۴-۴-۴
۱۱۳.....	مثال ۳ و ۴: اعتبار سنجی برنامه تهیه شده.....	۵-۴-۴
۱۱۳.....	مثال ۳.....	۱-۵-۴-۴

۱۱۶.....	۴-۵-۲ مثال ۴.....
۱۲۰.....	۴-۵ مثال‌های کاربردی در زمینه مهندسی معدن.....
۱۲۰.....	۴-۵-۱ مثال ۵: بلوک دهی در معدن سنگ ساختمانی.....
۱۲۳.....	۴-۵-۲ مثال ۶: بررسی انرژی خردایش در توده‌سنگ.....
۱۲۵.....	۴-۵-۳ مثال ۷: طراحی پارامترهای آتشفشانی.....
۱۲۶.....	۴-۵-۴ مثال ۸: پایداری و طراحی نگهداری بلوک‌ها در شیروانی‌ها.....
۱۲۷.....	۴-۵-۵ مثال ۹: پایداری و طراحی نگهداری بلوک‌ها در معادن زیرزمینی.....
۱۲۸.....	۴-۵-۶ مثال ۱۰: پیش‌بینی رقیق‌شدگی ماده معدنی و باطله در روش‌های استخراج تخریبی.....
۱۳۱.....	۴-۶ نتیجه‌گیری.....
۱۳۲.....	۵- جمع‌بندی، نتیجه‌گیری و پیشنهادات.....
۱۳۳.....	۵-۱ جمع‌بندی.....
۱۳۴.....	۵-۲ نتیجه‌گیری.....
۱۳۴.....	۵-۳ پیشنهادات.....
۱۳۵.....	۶- منابع.....
۱۳۸.....	۷- پیوست.....

فهرست اشکال

- شکل ۱-۲: اجزای تشکیل دهنده توده سنگ ۱۵
- شکل ۲-۲: انواع ناپیوستگی‌ها در توده سنگ ۱۷
- شکل ۳-۲: نحوه برداشت ناپیوستگی‌ها به روش خط برداشت ۱۹
- شکل ۴-۲: برداشت ناپیوستگی‌ها در یک محیط مربعی و دایره‌ای ۲۰
- شکل ۵-۲: تقسیم بندی بلوک‌ها ۲۲
- شکل ۶-۲: مهمترین بلوک‌های موجود در توده سنگ ۲۲
- شکل ۷-۲: مقادیر فاکتور شکل بلوک برای توده سنگ با سه دسته درزه ۲۳
- شکل ۸-۲: نمونه‌ای از توده سنگ با توزیع حجمی بلوک‌های آن ۲۷
- شکل ۹-۲: بلوک‌های نمایان در سطح زمین که به راحتی می‌توان حجم آن‌ها را محاسبه کرد ۲۸
- شکل ۱۰-۲: رابطه شاخص کیفیت سنگ و درزه‌ها ۳۱
- شکل ۱۱-۲: رابطه بین شاخص کیفیت توده سنگ و تعداد درزه حجمی با حجم بلوک‌ها ۳۲
- شکل ۱۲-۲: ارتباط بین روش‌های مختلف اندازه‌گیری حجم بلوک‌ها با استفاده از پارامترهای هندسی ۳۳
- شکل ۱-۳: نمونه‌ای از برخورد چند خط در صفحه ۳۸
- شکل ۲-۳: گراف تشکیل شده از محل برخورد خطوط و نقاط پایانی آن‌ها ۳۸
- شکل ۳-۳: کوچکترین چرخه‌ها یا گراف‌های بسته برای شکل (۳-۱) ۳۹
- شکل ۴-۳: یافتن چند ضلعی‌ها با استفاده از کوچکترین چرخه‌ها ۴۰
- شکل ۵-۳: سطوح آویزان و منفرد در شبیه سازی سه بعدی ۴۲
- شکل ۶-۳: خطوط آویزان و منفرد در شبیه سازی دو بعدی ۴۲
- شکل ۷-۳: مرتب‌سازی شبکه ناپیوستگی‌ها (الف) شبکه ناپیوستگی اصلی (ب) مرتب‌سازی شبکه ناپیوستگی برای یافتن بلوک (ج) مرتب‌سازی شبکه ناپیوستگی بعد از حذف زیر شبکه‌های غیر مرتبط با شبکه اصلی ۴۴
- شکل ۸-۳: (الف) ناپیوستگی‌های مستقیم با دو رأس ($P1$ و $P2$)، (ب) یک بلوک با رئوس و یال‌های آن ۴۵
- شکل ۹-۳: بلوک‌های دو بعدی (الف) بلوک‌های گود (حفره‌ها) (ب) بلوک‌های برجسته (ج) بلوک‌های ترکیبی ۴۵
- شکل ۱۰-۳: (الف) دو ناپیوستگی که تقاطعی ندارند. (ب) دو ناپیوستگی که یک نقطه تقاطع دارند ۴۶
- شکل ۱۱-۳: رئوس مشخص شده در طول یک ناپیوستگی ۴۸
- شکل ۱۲-۳: مثالی از یک شبکه ناپیوستگی (الف) شبکه ناپیوستگی با رئوس آن (ب) ماتریس ناپیوستگی‌ها و نقاط برخورد روی آن‌ها ۴۹
- شکل ۱۳-۳: (الف) گراف جهتدار از یک شبکه ناپیوستگی (ب) ماتریس رئوس ناپیوستگی‌ها (ج) ماتریس جهت ۵۰
- شکل ۱۴-۳: (الف) زنجیره اصلی رئوس (ب) رئوس تجزیه و علامت‌دار شده ۵۱

شکل ۳-۱۵:	نمونه‌ای از اجرای عملیات عملگر مرزی بر روی یک بلوک با شش یال.....	۵۲
شکل ۳-۱۶:	قانون زاویه دست چپی.....	۵۳
شکل ۳-۱۷:	(الف) بلوکی با مثلث‌هایی با مساحت مثبت (ب) بلوکی با مثلث‌هایی با مساحت مثبت و منفی (هاشور خورده).....	۵۴
شکل ۳-۱۸:	پروسه یافتن بلوک‌ها در یک شبکه ناپیوستگی با استفاده از قانون زاویه دست چپی.....	۵۶
شکل ۳-۱۹:	گراف و ماتریس اتصال بین رئوس در مثال شکل (۳-۱۲).....	۵۷
شکل ۳-۲۰:	گراف و ماتریس اتصال بین بلوک‌ها.....	۵۸
شکل ۳-۲۱:	نام گذاری مجدد رئوس و یال‌ها.....	۵۹
شکل ۳-۲۲:	(الف) مثالی از یک شبکه ناپیوستگی واقعی (ب) یافتن بلوک‌ها با استفاده از کد UDEC.....	۵۹
شکل ۳-۲۳:	تبدیل یک منحنی به تعدادی پاره‌خط مستقیم.....	۶۱
شکل ۳-۲۴:	برخورد دو پاره‌خط در صفحه.....	۶۲
شکل ۳-۲۵:	دو پاره‌خط با یک نقطه برخورد مجازی.....	۶۳
شکل ۳-۲۶:	روش زاویه‌ای (الف) گراف با فضای داخلی (ب) گراف با فضای خارجی.....	۶۸
شکل ۳-۲۷:	روش نقطه‌ای (الف) تعداد برخورد فرد (فضای داخلی) (ب) تعداد برخورد زوج (فضای خارجی).....	۶۸
شکل ۳-۲۸:	نمای گرافیکی ناپیوستگی‌ها.....	۶۹
شکل ۳-۲۹:	نقاط برخورد ناپیوستگی‌های شبکه بلوکی.....	۷۰
شکل ۳-۳۰:	برش درختی ناپیوستگی‌ها.....	۷۱
شکل ۳-۳۱:	گراف تشکیل شده از نتیجه برش درختی.....	۷۲
شکل ۳-۳۲:	گوشه‌های واقعی یک چند ضلعی.....	۷۶
شکل ۳-۳۳:	(الف) یک پنج وجهی (ب) تقسیم سطح $P2$ به دو مثلث (ج) تقسیم پنج وجهی به سه، چهار وجهی.....	۷۸
شکل ۴-۱:	روند شناسایی بلوک‌ها در روش ماتریسی.....	۸۴
شکل ۴-۲:	بلوک اصلی و دسته درزه‌های متقاطع با آن.....	۸۵
شکل ۴-۳:	(الف) یک شبکه ناپیوستگی (ب) تعیین نقاط برخورد (ج) حذف یال‌های بدون تقاطع (د) مرتب سازی نهایی.....	۹۱
شکل ۴-۴:	روند نمای مراحل مرتب سازی یال‌ها.....	۹۲
شکل ۴-۵:	قانون زاویه دست راستی دو بعدی.....	۹۴
شکل ۴-۶:	روند نمای مراحل تشکیل ماتریس اتصال یال‌ها.....	۹۶
شکل ۴-۷:	روند نمای مراحل شناسایی بلوک‌ها.....	۹۹
شکل ۴-۸:	نمای گرافیکی مثال مذکور.....	۱۰۰
شکل ۴-۹:	رئوس، یال‌ها و بلوک‌ها.....	۱۰۴
شکل ۴-۱۰:	خروجی برنامه برای مثال ۱.....	۱۱۱
شکل ۴-۱۱:	خروجی تصویری برنامه برای مثال ۱.....	۱۱۱

شکل ۴-۱۲: مثال ۲ بلوکی با ابعاد $(10^{-14} * 10^{-14})$	۱۱۲
شکل ۴-۱۳: نمای گرافیکی ناپیوستگی‌های مثال ۳.....	۱۱۴
شکل ۴-۱۴: نحوه ورود اطلاعات و خروجی برنامه برای مثال ۳.....	۱۱۵
شکل ۴-۱۵: شکل مثال ۴ برای اعتبار سنجی برنامه.....	۱۱۷
شکل ۴-۱۶: نتایج ارائه شده برای مثال ۴.....	۱۱۸
شکل ۴-۱۷: خروجی برنامه تهیه شده برای مثال ۴.....	۱۱۹
شکل ۴-۱۸: نمای جبهه کار معدن سنگ.....	۱۲۱
شکل ۴-۱۹: نمای جبهه کار معدن با اعمال برش.....	۱۲۲
شکل ۴-۲۰: نمودار درصد بلوک‌های باطله، قابل حمل به سنگبری و سالم.....	۱۲۲
شکل ۴-۲۱: جبهه کار به همراه درزه‌های آن در مثال ۶.....	۱۲۴
شکل ۴-۲۲: نمودار توزیع تجمعی اندازه بلوک‌های توده‌سنگ.....	۱۲۴
شکل ۴-۲۳: نمای پله یک معدن رو باز و درزه‌های آن.....	۱۲۵
شکل ۴-۲۴: هندسه یک شیروانی و بلوک کلیدی در این شکل.....	۱۲۶
شکل ۴-۲۵: دهانه تونل یک معدن در عمق ۱۰ متری.....	۱۲۷
شکل ۴-۲۶: نمای افقی منطقه و بلوک‌های آن.....	۱۳۰
شکل ۴-۲۷: نمودار توزیع تجمعی اندازه بلوک‌های مثال ۱۰.....	۱۳۰

فهرست جداول

جدول ۱-۱: روش‌های ارائه شده برای تعیین هندسه بلوک‌های توده‌سنگ.....	۹
جدول ۱-۳: بلوک‌های شناسایی شده در شبکه ناپیوستگی.....	۷۳
جدول ۱-۴: مختصات نقاط ابتدا و انتها درزه‌ها در هر دسته درزه.....	۸۶
جدول ۲-۴: مختصات ناپیوستگی‌های.....	۹۸
جدول ۳-۴: لیست دستورات و توابع به کار رفته در برنامه تهیه شده.....	۱۰۶
جدول ۴-۴: مختصات ناپیوستگی‌های مثال ۳.....	۱۱۳
جدول ۵-۴: محاسبه مساحت بلوک‌های مثال ۳ به صورت دستی.....	۱۱۴
جدول ۶-۴: مقایسه نتایج حل دستی و حل توسط برنامه مثال ۳.....	۱۱۶
جدول ۷-۴: ناپیوستگی‌های برداشت شده بالای تونل.....	۱۲۸
جدول ۸-۴: مشخصات هندسی بلوک‌های تشکیل شده در این مثال.....	۱۲۸
جدول ۹-۴: دسته درزه‌های مثال ۱۰.....	۱۲۹

فصل اول

کلیات

۱-۱ مقدمه

در اثر عوامل مختلف نظیر فعالیت‌های تکتونیکی شکستگی‌هایی در توده‌سنگ ایجاد می‌شود. این شکستگی‌ها یا ناپیوستگی‌ها باعث ایجاد بلوک‌هایی با ابعاد و هندسه متفاوت در توده‌سنگ می‌شوند. اطلاع از هندسه توده‌سنگ و بلوک‌های آن در زمینه‌های مختلف مهندسی سنگ نظیر طراحی سیستم نگهداری فضاها، زیرزمینی احداث شده در توده‌سنگ درزه‌دار، طراحی الگوی آتشیاری توده‌های سنگی، بهینه‌سازی خردایش سنگ، تعیین بلوک‌دهی سنگ‌های ساختمانی، بررسی رقیق‌شدگی ماده معدنی در روش‌های استخراج تخریبی و ... ضروری می‌باشد. دو رویکرد مختلف برای مدل‌سازی هندسی ناپیوستگی‌های موجود در توده‌سنگ وجود دارد. در یک رویکرد ناپیوستگی‌ها موازی و با طول نامحدود فرض می‌شوند و در رویکرد دیگر ناپیوستگی‌ها با طول محدود و با شیب و امتداد شیب واقعی‌تر در نظر گرفته می‌شوند. هندسه بلوک‌های برجای توده‌سنگ با توجه به این دو رویکرد در دو و سه بعد تعیین می‌شود. در این فصل ابتدا ضرورت انجام این مطالعه بیان می‌شود در ادامه مطالعات و پیشرفت‌های انجام شده توسط محققین دیگر توضیح داده می‌شود. در پایان هدف از انجام مطالعه و ساختار گزارش تهیه شده در این فصل توضیح داده می‌شود.

۲-۱ ضرورت انجام مطالعه

روش‌های گوناگونی برای تعیین شکل و حجم بلوک‌ها در توده‌سنگ درزه‌دار وجود دارد که در این روش‌ها مشخصات هندسی بلوک‌های بر جای توده‌سنگ محاسبه می‌شود. برخی از روش‌ها ناپیوستگی‌های توده‌سنگ را موازی و با طول نامحدود فرض می‌کنند که در این روش‌ها با استفاده از پارامترهای هندسی توده‌سنگ درزه‌دار نظیر تعداد دسته‌درزه‌ها، فاصله‌داری درزه‌ها و ... هندسه بلوک‌های توده‌سنگ تعیین می‌شود. در برخی دیگر از روش‌ها ناپیوستگی‌های موجود در توده‌سنگ به صورت واقعی‌تر و با طول محدود فرض می‌شوند. در این روش‌ها مشخصات هندسی توده‌سنگ و

بلوک‌های آن با استفاده از الگوریتم‌های مختلف تعیین می‌شود. ناپیوستگی‌ها در دو بعد به صورت خط و در سه بعد به صورت صفحه فرض می‌شوند. این الگوریتم‌ها بر مبنای عملیات ریاضی و هندسی بر روی شبکه‌ای از خطوط یا صفحات متقاطع در فضا هستند. مطالعه و آشنایی با روش‌های و الگوریتم‌های مختلف به منظور تهیه برنامه کامپیوتری تعیین خصوصیات هندسی توده‌سنگ و بلوک‌های آن ضروری می‌باشد. همچنین به دلیل استفاده روز افزون از کامپیوتر در علوم مختلف به علت سرعت و دقت در محاسبات، تهیه برنامه کامپیوتری برای حوزه‌های مختلف مهندسی معدن از جمله موضوع مورد مطالعه امری ضروری است. از کاربردهای اطلاع از هندسه توده‌سنگ می‌توان به موارد زیر اشاره کرد:

- در معادن سنگ ساختمانی با اطلاع از حجم برجای بلوک‌های توده‌سنگ می‌توان بلوک‌دهی معدن و همچنین جهت پیشروی مناسب را مورد بررسی قرار داد. در این معادن با داشتن مشخصات درزه‌های موجود در معدن و با توجه به هندسه توده‌سنگ می‌توان میزان بلوک‌های سالم و باطله را مشخص نمود و به این ترتیب اقتصادی بودن یک معدن سنگ را با توجه به درصد انواع بلوک‌ها و پارامترهای اقتصادی می‌توان مشخص کرد. از جهتی نیز با توجه به موقعیت درزه‌ها و ابعاد بلوک‌های موجود در توده‌سنگ می‌توان جهت پیشروی مناسب را طوری انتخاب کرد که بیشترین بلوک‌های سالم هنگام استخراج تولید شود.

- در معادن زیرزمینی و روباز نیز برای بررسی نگهداری و استخراج ماده معدنی دانستن حجم و شکل بلوک‌های توده‌سنگ بسیار اهمیت دارد. در دیواره معادن روباز و دیواره تونل‌ها و کارگاه‌های معادن زیرزمینی در اثر وجود شکستگی‌ها بلوک‌هایی با ابعاد مختلف ایجاد می‌شوند. برای تحلیل پایداری و طراحی نگهداری این بلوک‌ها نیاز به هندسه برجای این بلوک‌ها است.

- برای تهیه الگوی آتشباری اطلاع از شکل و توزیع حجم بلوک‌ها در انتخاب پارامترهای آتشباری مؤثر است. برای بررسی انرژی منتقل شده به توده‌سنگ به هنگام آتشباری نیاز به توزیع حجم بلوک‌های برجای توده سنگ می‌باشد. برای طراحی الگوی آتشباری در معادن نیز با داشتن مشخصات هندسی توده‌سنگ می‌توان پارامترهایی نظیر بار سنگ و فاصله‌داری چال‌ها و ... را به طور مناسبی انتخاب نمود.

- در معدنی که برای استخراج ماده معدنی از روش‌های استخراج تخریبی استفاده می‌شود؛ هندسه بلوک‌های برجای توده‌سنگ در مکانیسم رقیق شدگی ماده معدنی تأثیر دارد. با داشتن توزیع حجم بلوک‌های برجای توده‌سنگ می‌توان میزان رقیق شدگی ماده معدنی را کنترل نمود.

- در بسیاری از پروژه‌های عمرانی نیز کاربردهای مشابه را می‌توان مشاهده کرد.

با توجه به کاربردهای متعدد تعیین مشخصات هندسی توده‌سنگ و بلوک‌های آن و در دسترس نبودن کدهای مرجع هیچ یک از این برنامه‌های ارائه شده در این زمینه و هزینه بالای تهیه آن‌ها، طراحی برنامه‌های جدید مورد نیاز خواهد بود. همچنین برنامه‌ای که قرار است تهیه شود می‌تواند با توجه به کاربردهای ذکر شده برای مطالعات بعدی مورد استفاده قرار گیرد. به طور کلی ضرورت انجام این پایان‌نامه را می‌توان به صورت زیر خلاصه کرد:

- ۱- کاربردهای متعدد تعیین هندسه بلوک‌های توده‌سنگ
- ۲- در دسترس نبودن کدهای مرجع برنامه‌های کامپیوتری تهیه شده
- ۳- هزینه بالای خرید نرم افزارهای تهیه شده
- ۴- امکان توسعه برنامه‌ی کامپیوتری که قرار است تهیه شود.

۳-۱ پیشینه مطالعه

همانطور که اشاره شد دو رویکرد کلی برای تعیین هندسه توده سنگ و حجم و شکل بلوک‌های آن در دو و سه بعد وجود دارد. در گذشته از روش‌های رویکرد اول که ناپیوستگی‌ها را نامحدود و موازی فرض می‌کند، برای تعیین خصوصیات هندسی توده سنگ استفاده می‌شد. شارما^۱ و همکاران در کتاب "خصوصیات برجای سنگ‌ها" روش‌های موجود با این رویکرد که اکثراً نتیجه تحقیقات و مطالعات پالمستروم^۲ است را گردآوری کرده‌اند [۱]. در این روش‌ها مشخصات یک بلوک به عنوان نماینده همه بلوک‌های توده سنگ انتخاب می‌شود و خصوصیات آن به همه بلوک‌های توده سنگ اختصاص داده می‌شود.

واربرتون^۳ با توجه به پارامترهای هندسی توده سنگ الگوریتم و نرم افزاری برای تعیین خصوصیات هندسی بلوک‌های برجای توده سنگ ارائه داده است. در این الگوریتم درزه‌ها به صورت موازی و سراسری فرض می‌شوند و با استفاده از روابط ارائه شده در مورد محاسبه هندسه بلوک‌های توده سنگ به وسیله پارامترهای هندسی حجم بلوک‌ها محاسبه می‌شود [۲].

لو و لاتهام^۴ [۳]، استارزک و تسانگ^۵ [۴] و اسمیت^۶ [۵] نیز از کسانی بوده‌اند که با توجه به پارامترهای هندسی به بررسی هندسه بلوک‌های توده سنگ پرداخته‌اند. به دلیل خطای بالا و فرضیات غیر واقع این روش‌ها کمتر مورد استفاده قرار می‌گیرند. در این پایان‌نامه نیز در فصل دوم به طور کلی به این روش‌ها و روابط مربوط به آن‌ها پرداخته می‌شود.

¹ Sharma

² Palmstrom

³ Warburton

⁴ Lu & Latham

⁵ Starzec & Tsang

⁶ Smith

در روش‌های رویکرد دوم ناپیوستگی‌ها را با طول محدود و به صورت واقعی فرض می‌کنند. در این روش‌ها هدف تعیین بلوک‌های ایجاد شده در اثر برخورد خطوط و صفحات معین می‌باشد. برای تعیین بلوک‌ها در دو و سه بعد الگوریتم‌های متفاوتی ارائه شده است که بر اساس ادعای فریرا و همکاران^۱ [۶] بهینه‌ترین الگوریتم از لحاظ زمان لازم و حجم مورد نیاز برای پردازش اطلاعات، الگوریتم مبتنی بر تئوری گراف‌ها است. در این الگوریتم ابتدا نقاط برخورد خطوط در دو بعد محاسبه و پس از آن گرافی با استفاده از این نقاط و خطوط بین آن‌ها تشکیل داده می‌شود. با استفاده از این گراف چند ضلعی‌هایی که در شبکه خطوط به وجود آمده‌اند تعیین می‌شوند. این چند ضلعی‌ها در دو بعد همان بلوک‌ها هستند. از مطالعات انجام شده توسط محققان رشته مهندسی معدن با توجه به کاربرد موضوع می‌توان به روش‌ها و الگوریتم‌هایی که در ادامه بیان می‌شود اشاره نمود.

هلیوت^۲ [۷] الگوریتمی را بر مبنای سیستم‌های عددی در سه بعد برای مدل‌سازی ناپیوستگی‌ها و تعیین مشخصات هندسی بلوک‌های توده‌سنگ ارائه داده است. این الگوریتم ناپیوستگی‌ها را با طول محدود فرض می‌کند و دسته درزه‌هایی با طول نامحدود در این روش در نظر گرفته نمی‌شوند.

گودمن و شی^۳ [۸] برای شناسایی بلوک‌های توده‌سنگ الگوریتمی بر مبنای روش‌های برداری ارائه داده‌اند. در این الگوریتم با استفاده از نقاط برخورد بلوک‌های ایجاد شده در دو و سه بعد شناسایی می‌شوند.

¹ Ferreira et al

² Heliot

³ Goodman & Shi

شی^۱ [۹] در قسمتی از رساله دکتری خود به بررسی شناسایی بلوک‌های تشکیل شده در اثر برخورد ناپیوستگی‌های منحنی شکل و مستقیم پرداخته است. مراحل الگوریتم ارائه شده توسط ایشان بلوک‌های دو بعدی را به وسیله گراف‌های جهت‌دار تعیین می‌کند.

لین و همکاران^۲ [۱۰] نیز برای حل این مسأله الگوریتمی بر مبنای تکنیک‌های مکان‌شناسی ارائه داده‌اند. این الگوریتم سه بعدی بوده و قادر است بلوک‌های ایجاد شده در اثر برخورد صفحات در فضا را شناسایی کند.

مارز و جرمین^۳ [۱۱] برای تعیین هندسه بلوک‌های توده‌سنگ از مدل‌سازی استفاده کرده‌اند، آن‌ها برای تعیین اندازه بلوک‌ها در اطراف فضاها زیر زمینی نرم افزار SIMBLOCK را طراحی نموده‌اند.

جینگ و استفانسون^۴ [۱۲ و ۱۳ و ۱۴] از کسانی بوده‌اند که بر روی این مسأله تحقیقات فراوانی داشته‌اند و الگوریتم ارائه شده توسط آن‌ها در طی سال‌های مختلف توسعه داده شده است. این الگوریتم به بررسی ناپیوستگی‌ها و بلوک‌های توده‌سنگ در دو و سه بعد می‌پردازد و بر مبنای تکنیک‌های عددی می‌باشد. الگوریتم دو بعدی آن با در نظر گرفتن دو جهت برای هر خط ارائه شده است. در این الگوریتم با تبدیل شبکه خطوط به گراف جهت‌دار و طی مراحل مختلف بلوک‌ها شناسایی می‌شوند. همچنین آخرین الگوریتم ارائه شده توسط آن‌ها به بررسی مکانیکی و جریان سیالات در ناپیوستگی‌ها نیز می‌پردازد.

¹ Shi

² Line et al

³ Maerz & Germain

⁴ Jing & Stephansson

الگوریتم ارائه شده توسط لو و همکاران^۱ [۲] بر اساس روش‌های برداری بلوک‌ها را در سه بعد مشخص می‌کند. به این صورت که ابتدا صفحات ناپیوستگی را به صورت محدود فرض کرده و سپس نقاط حاصل از تقاطع خطوط فصل مشترک صفحات را به صورت بردار به یکدیگر متصل می‌کند. این بردارها اساس مراحل بعدی برای تشکیل بلوک خواهند بود.

روش ارائه شده توسط اولکر و توران بوی^۲ [۱۵] برای تعیین بلوک‌ها در سه بعد بر مبنای الگوریتم ژنتیک است. این الگوریتم بر اساس ساختار درختی برنامه ریزی شده است. پس از برداشت اطلاعات صحرایی ناپیوستگی‌ها و تبدیل آن به اطلاعات نرم افزاری ابتدا نقاط برخورد ناپیوستگی‌ها در سه بعد تعیین می‌شوند و پس از آن با استفاده از پاره‌خط‌های تشکیل دهنده بلوک‌ها و بر اساس یک الگوریتم درختی هندسه بلوک‌ها تعیین می‌شود.

جعفری و همکاران الگوریتمی ابتکاری بر مبنای عملیات بر روی ماتریس‌های اساسی برای شناسایی بلوک‌ها در سه بعد را ارائه داده‌اند. خیشوند و همکاران نیز این الگوریتم را برای مسائل دو بعدی توسعه داده‌اند [۱۶].

ژانگ و همکاران^۳ [۱۷ و ۱۸] نیز از کسانی بوده‌اند که در سال‌های اخیر در مورد شناسایی و تحلیل بلوک‌های توده‌سنگ روش‌هایی را ارائه داده‌اند.

در جدول (۱-۱) روش‌های ارائه شده توسط محققین مختلف برای تعیین هندسه بلوک‌های توده‌سنگ آورده شده است.

¹ Lu et al

² Ulker & Turanboy

³ Zhang et al

جدول ۱-۱: روش‌های ارائه شده برای تعیین هندسه بلوک‌های توده‌سنگ

ردیف	پدید آورنده	سال انتشار	مبنای روش	بعد روش
۱	پالمستروم	۱۹۸۲	استفاده از پارامترهای هندسی	دو و سه بعدی
۲	واربرتون	۱۹۸۳	استفاده از پارامترهای هندسی	دو و سه بعدی
۳	لین و همکاران	۱۹۸۷	تکنیک‌های مکان شناسی	سه بعدی
۴	هلیوت	۱۹۸۸	سیستم‌های عددی	سه بعدی
۵	شی	۱۹۸۸	استفاده از گراف‌های جهت دار	دو بعدی
۶	جینگ و استفانسون	۱۹۹۴	تکنیک‌های عددی	دو بعدی
۷	مارز و جرمین	۱۹۹۶	استفاده از پارامترهای هندسی	سه بعدی
۸	لو و لاتهام	۱۹۹۹	استفاده از پارامترهای هندسی	دو و سه بعدی
۹	فریرا و همکاران	۲۰۰۰	تئوری گراف‌ها	دو و سه بعدی
۱۰	جینگ	۲۰۰۰	تکنیک‌های عددی	سه بعدی
۱۱	استارزک و تسانگ	۲۰۰۲	استفاده از پارامترهای هندسی	دو و سه بعدی
۱۲	اسمیت	۲۰۰۴	استفاده از پارامترهای هندسی	دو و سه بعدی
۱۳	جینگ و استفانسون	۲۰۰۷	تکنیک‌های عددی	دو و سه بعدی
۱۴	لو و همکاران	۲۰۰۸	روش‌های برداری	سه بعدی
۱۵	اولکر و توران بوی	۲۰۰۹	الگوریتم ژنتیک	سه بعدی
۱۶	ژانگ و همکاران	۲۰۱۰	مدل‌های المان محدود و روش‌های برداری	سه بعدی
۱۷	جعفری و همکاران	۲۰۱۰	ماتریس‌های اساسی	سه بعدی
۱۸	خیشوند و همکاران	۲۰۱۱	ماتریس‌های اساسی	دو بعدی
۱۹	ژانگ و همکاران	۲۰۱۲	تکنیک‌های شبکه‌های مشبک	سه بعدی

در این جدول مبنای روش ارائه شده برای محاسبه هندسه توده‌سنگ و بلوک‌های آن، سال و بعد فرض شده برای ناپیوستگی‌های توده‌سنگ وجود دارد. روش‌های ذکر شده دارای مزایا و معایبی هستند که مربوط به فرضیات روش می‌باشد. این فرضیات هر چه به واقعیت نزدیک‌تر باشد، روش ارائه شده از اعتبار بیشتری برخوردار است. در روش‌هایی که برای تعیین هندسه توده‌سنگ از پارامترهای هندسی استفاده می‌شود، فرضیات نامحدود بودن و موازی بودن درزه‌ها از معایب اصلی روش هستند. در روش‌هایی که ناپیوستگی‌ها را واقعی‌تر فرض می‌کنند نیز پیچیدگی مربوط به الگوریتم ارائه شده برای یافتن بلوک‌ها در شبکه ناپیوستگی مهمترین پارامتر برای ارزیابی روش است. در فصل سوم به بررسی کامل تعدادی از الگوریتم‌های دو بعدی ذکر شده برای تعیین هندسه بلوک‌های توده‌سنگ پرداخته خواهد شد.

در این زمینه نرم افزارهای مختلفی نظیر SIMBLOCK، UDEC، برنامه ارائه شده توسط خیشوند و همکاران و ... طراحی شده است. این برنامه‌ها برای شناسایی بلوک‌ها در دو و سه بعدی طراحی شده‌اند و تفاوت این برنامه‌ها در الگوریتم و روش مورد استفاده، زبان برنامه نویسی و سرعت پردازش اطلاعات می‌باشد.

۴-۱ هدف از انجام مطالعه

هدف‌های اصلی این پایان‌نامه عبارتند از:

۱- مطالعه و بررسی الگوریتم‌های موجود برای شناسایی بلوک‌های توده‌سنگ در دو بعد

۲- انتخاب الگوریتم مناسب جهت تهیه برنامه کامپیوتری

۳- تهیه برنامه کامپیوتری تعیین شکل و مساحت بلوک‌های توده‌سنگ در دو بعد

در این پایان‌نامه ابتدا به بررسی و مطالعه روش‌های تعیین حجم و شکل بلوک‌های موجود در توده‌سنگ پرداخته می‌شود. پس از آن انتخاب الگوریتم مناسب و تهیه برنامه کامپیوتری هدف‌های بعدی خواهند بود. برای مطالعه روش‌های مختلف تعیین حجم و شکل توده‌سنگ از منابع مختلف و در دسترس استفاده می‌شود و پس از بررسی روش‌ها، الگوریتم مناسب جهت تهیه برنامه کامپیوتری انتخاب می‌شود. برای تهیه برنامه کامپیوتری از محیط نرم افزار MATLAB و دستورات لازم برای پیاده سازی الگوریتم استفاده می‌شود.

۵-۱ ساختار پایان‌نامه

برنامه کامپیوتری که قرار است در این پایان‌نامه تهیه شود هر دو رویکرد مختلف در روش‌های تعیین هندسه بلوک‌ها را پوشش می‌دهد. به عبارت دیگر هم دسته درزه‌های نامحدود و موازی و هم درزه‌های محدود و منفرد می‌توانند ورودی برنامه کامپیوتری باشند. به همین دلیل این دو رویکرد در فصل‌های دوم و سوم مورد بررسی قرار می‌گیرند. ابتدا در فصل دوم کلیاتی در مورد توده‌سنگ درزه‌دار و اجزای آن بیان می‌شود؛ در ادامه نحوه تعیین مشخصات ناپیوستگی‌ها و نحوه محاسبه هندسه بلوک‌های توده‌سنگ با استفاده از پارامترهای هندسی در این فصل بررسی می‌شود.

در فصل سوم روش‌های تعیین هندسه بلوک‌ها با استفاده از الگوریتم‌های مختلف دو بعدی توضیح داده می‌شود. در این فصل از بین الگوریتم‌های موجود برای تعیین هندسه بلوک‌های توده‌سنگ در حالت دو بعدی الگوریتم مبتنی بر تئوری گراف‌ها، الگوریتم تکنیک‌ها عددی، الگوریتم مبتنی بر گراف‌های جهت‌دار و الگوریتم مبتنی بر روش‌های برداری مورد بررسی قرار می‌گیرند. مراحل، مزایا و معایب این روش‌ها در این فصل بررسی می‌گردد. در فصل چهارم ابتدا به بررسی الگوریتم‌های شرح داده شده در فصل سوم و انتخاب الگوریتم مناسب جهت تهیه برنامه کامپیوتری پرداخته می‌شود. در ادامه الگوریتم مبتنی بر ماتریس‌های اساسی به منظور پیاده‌سازی به طور کامل توضیح داده خواهد

شد. در این فصل همچنین نحوه تهیه برنامه کامپیوتری تعیین هندسه بلوک‌های توده‌سنگ در حالت دو بعدی و داده‌های ورودی و خروجی برنامه بررسی می‌گردد و پایان فصل چهارم چند مثال کاربردی در زمینه مهندسی معدن و حل آن توسط برنامه تهیه شده خواهد بود. در پایان جمع‌بندی، نتیجه گیری و پیشنهادات مربوط به این پایان نامه در فصل پنجم آورده خواهد شد.

فصل دوم

توده سنگ و تعیین هندسه بلوک ها توسط

پارامترهای هندسی

۱-۲ مقدمه

هندسه سیستم ناپیوستگی‌ها در توده‌های سنگی یکی از مهمترین جنبه‌های تعیین خصوصیات مدل در علم مهندسی سنگ می‌باشد. در یک توده سنگ دارای ناپیوستگی، سه نوع المان ساختاری وجود دارد که عبارتند از: ناپیوستگی، زمینه تشکیل دهنده سنگ و سیالات درون آن (مثل آب، نفت و گاز و ...). تنش‌های وارده بر توده سنگ و دمای آن از عوامل محیطی به شمار می‌روند. یکی از مهم‌ترین مسائل در مدل‌سازی عددی در علم مهندسی سنگ، نشان دادن سیستم ناپیوستگی‌های یک توده سنگ به صورت یک مدل هندسی است. این امر مستلزم توصیف کمی محل، جهت یابی، اندازه، شکل، میزان باز شدگی ناپیوستگی‌ها و همچنین نحوه اتصال این ناپیوستگی‌ها با هم می‌باشد.

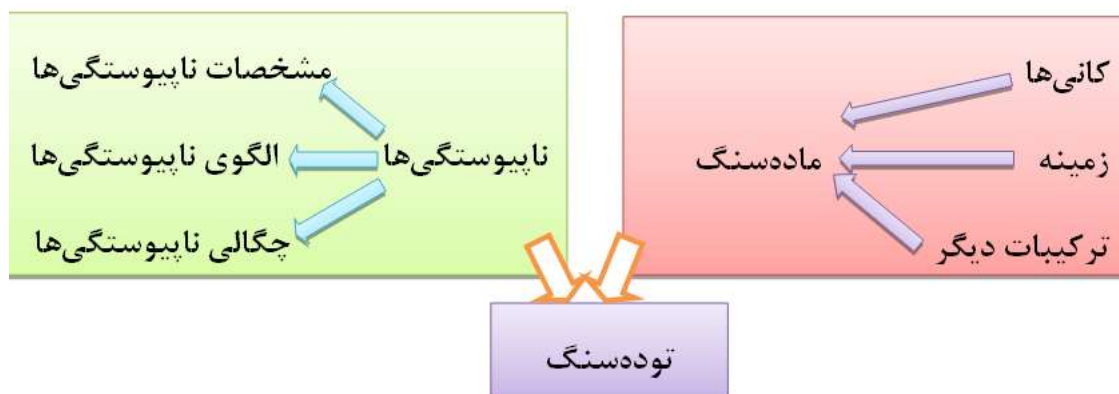
در برخی از روش‌هایی که در فصل اول اشاره شد برای محاسبه حجم بلوک‌های توده سنگ از پارامترهای هندسی مانند فاصله‌داری درزه‌ها، تعداد درزه حجمی، فاکتور حجمی بلوک و ... استفاده می‌شود. در این فصل ابتدا به بررسی اجزای تشکیل دهنده توده سنگ، انواع ناپیوستگی‌ها و بلوک‌های موجود در آن پرداخته می‌شود. در این فصل روش‌های محاسبه حجم بلوک‌های توده سنگ با استفاده از پارامترهای هندسی توضیح داده می‌شود.

۲-۲ توده سنگ درزه‌دار و اجزای آن

توده سنگ‌های موجود در طبیعت با توجه به نحوه تشکیل دارای انواع مختلف آذرین، دگرگونی و رسوبی هستند. توده سنگ‌ها بر اثر عوامل مختلف نظیر عوامل تکتونیکی تغییر شکل می‌یابند و ناپیوستگی‌های مختلف با خصوصیات متفاوت در آن‌ها ایجاد می‌شوند. با توجه به میزان باز شدگی، طول و شیب ناپیوستگی‌ها آن‌ها را به انواع مختلف که در ادامه ذکر خواهد شد، تقسیم بندی می‌کنند. با توجه به انواع ناپیوستگی‌ها و فاصله‌داری آن‌ها بلوک‌هایی با ابعاد و شکل‌های مختلف در توده سنگ

ایجاد می‌شود. نوع ماده سنگ تشکیل دهنده سنگ‌های توده سنگ، الگو و چگالی ناپیوستگی‌ها تأثیر زیادی در خصوصیات مکانیکی و هندسی توده سنگ دارد.

شکل (۱-۲) اجزای تشکیل دهنده توده سنگ که شامل ماده سنگ و ناپیوستگی‌ها می‌باشد را نشان می‌دهد.



شکل ۱-۲: اجزای تشکیل دهنده توده سنگ [۱]

۱-۲-۲ ماده سنگ

ماده سنگ از کانی‌های مختلف، زمینه سنگ و ترکیبات دیگر تشکیل شده است. نوع ماده سنگ ویژگی‌های هندسی و مکانیکی توده سنگ را تعیین می‌کند. به عنوان مثال مقاومت توده سنگ تحت تأثیر مستقیم مقاومت ماده سنگ است.

۲-۲-۲ ناپیوستگی‌های توده سنگ

ناپیوستگی‌ها زیادی در توده سنگ وجود دارد که تعاریف مختلفی از نظر متخصصین برای آن‌ها ارائه شده است. این ناپیوستگی‌ها تشکیل دهنده بلوک‌های توده سنگ بوده و رفتار مکانیکی توده سنگ را

کنترل می‌کنند. مطابق شکل (۲-۲) در سال ۱۹۷۵ انجمن مهندسين مکانیک سنگ^۱ تعاریف زیر را بر اساس فاصله صفحات و طول باز شدگی برای بازشدگی‌های مختلف ارائه داده‌اند که به انواع زیر تقسیم می‌شود [۱]:

- **درزه^۲:** صفحه‌ای به وجود آمده در توده سنگ که هیچ گونه جابجایی نداشته باشد را درزه گویند که طول آن بین ۵ سانتی‌متر تا ۱۰۰ متر است.
- **ترک^۳:** یک جدا شدگی یا ناپیوستگی کوچک ناقص که طول آن کمتر از یک سانتی‌متر باشد را ترک گویند.
- **شکاف^۴:** ناپیوستگی در سنگ ناشی از چین و گسل‌های شدید که همراه با باز شدگی زیاد باشد را شکاف در سنگ گویند.
- **تورق^۵:** صفحه یا سطح افقی بین لایه‌های سنگ را تورق گویند.
- **گسیختگی^۶:** ناپیوستگی‌های حاصل از استخراج یا فعالیت‌های انسانی را گسیختگی گویند.
- **برش^۷:** رگه‌ای از سنگ‌های خرد شده که معمولاً ضخامتی بیشتر از درزه‌ها دارند را برش گویند.
- **رگه^۸:** ناپیوستگی‌هایی با ضخامت زیاد که با سنگ یا خاک پر شده باشد را رگه گویند.

^۱ ISRM (International Society of Rock Mechanic)

^۲ Joint

^۳ Crack

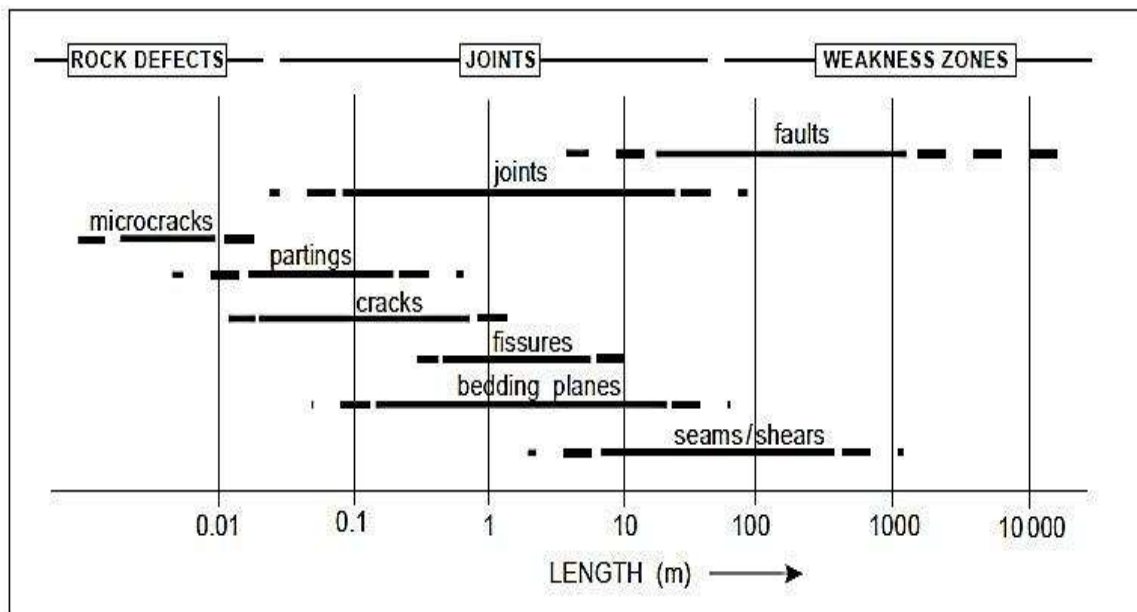
^۴ Fissure

^۵ Parting

^۶ Rupture

^۷ Shear

^۸ Seam



شکل ۲-۲: انواع ناپیوستگی‌ها در توده سنگ [۱]

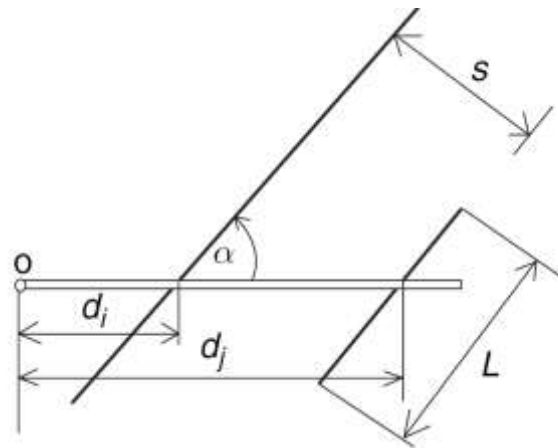
ناپیوستگی‌ها رفتار مکانیکی توده سنگ را کنترل می‌کنند؛ بنابراین شناخت ناپیوستگی‌ها برای کنترل رفتار مکانیکی توده سنگ امری ضروری می‌باشد. جهت‌یابی، چگالی، فاصله‌داری، شکل، اندازه، رخنمون، میزان بازشدگی و موقعیت از خصوصیات اصلی ناپیوستگی‌ها هستند. با دانستن همه این خصوصیات یک ناپیوستگی را می‌توان به طور کامل شناخت و در برنامه‌های کامپیوتری مدل‌سازی کرد.

خصوصیات ناپیوستگی‌ها به ترتیب زیر تعریف می‌شوند [۱۴]:

- **جهت‌گیری:** جهت‌گیری شامل زاویه و جهت شیب ناپیوستگی می‌باشد. این پارامتر مهم‌ترین خصوصیت یک ناپیوستگی است و در حد مطلوبی در توده سنگ برای انواع ناپیوستگی‌ها از طریق برداشت‌های صحرایی و روش‌های دیگر قابل اندازه‌گیری می‌باشد.

- **چگالی:** تعداد ناپیوستگی در واحد حجم یک توده سنگ را چگالی گویند. این پارامتر رابطه غیر مستقیم با فاصله‌داری ناپیوستگی‌ها در توده سنگ دارد. هر چقدر میزان فاصله‌داری بیشتر باشد، مقدار چگالی ناپیوستگی‌ها کمتر است.
- **فاصله‌داری:** فاصله عمودی بین دو ناپیوستگی متوالی را فاصله‌داری گویند. با توجه به این که فاصله ناپیوستگی‌ها در توده سنگ معمولاً ثابت نیست، اغلب برای معرفی این پارامتر از دو مقدار متوسط و بیشترین میزان فاصله‌داری استفاده می‌شود.
- **شکل:** در دو بعد ناپیوستگی‌ها را به صورت پاره‌خط‌های صاف و مستقیم یا منحنی شکل در نظر می‌گیرند. در سه بعد نیز ناپیوستگی‌ها را به صورت صفحات مستطیل شکل یا به صورت دیسک فرض می‌کنند. این پارامتر از خصوصیات ناپیوستگی‌ها اغلب ناشناخته است.
- **اندازه:** میزان پراکندگی یک ناپیوستگی را اندازه ناپیوستگی گویند. اندازه ناپیوستگی‌ها با توجه به عدم شناخت نسبت به سطح زیر زمین غیر قابل محاسبه است و از رخنمون ناپیوستگی میزان اندازه آن محاسبه می‌شود.
- **رخنمون:** به رد یا اثر ناپیوستگی بر روی توده سنگ رخنمون ناپیوستگی گویند. این پارامتر از طریق برداشت‌های صحرایی قابل محاسبه است.
- **میزان باز شدگی:** فاصله بین دو صفحه ناپیوستگی را میزان باز شدگی گویند. اغلب به دلیل اینکه ناپیوستگی‌ها را به صورت پاره خط و صفحه فرض می‌کنند، میزان باز شدگی معمولاً صفر در نظر گرفته می‌شود.
- **موقعیت:** مختصات مرکز هندسی یک ناپیوستگی را موقعیت آن گویند. برای معرفی هندسه ناپیوستگی‌ها اغلب از این پارامتر استفاده می‌شود.

خصوصیات معرفی شده، پارامترهایی هستند که در برداشت‌های صحرایی یا از طریق مغزه‌های حفاری برای معرفی ناپیوستگی‌ها اندازه‌گیری می‌شوند. با توجه به اینکه این پارامترها با خطای بالایی اندازه‌گیری می‌شوند بهتر است از توزیع‌های احتمالی برای تعیین این پارامترها استفاده شود. یکی از روش‌های معمول در برداشت ناپیوستگی‌ها استفاده از خط برداشت^۱ است. در این روش خصوصیات ناپیوستگی‌هایی که با خط برداشت برخورد دارند اندازه‌گیری می‌شود. در شکل (۲-۳) نحوه برداشت ناپیوستگی‌ها به روش خط برداشت نشان داده شده است.



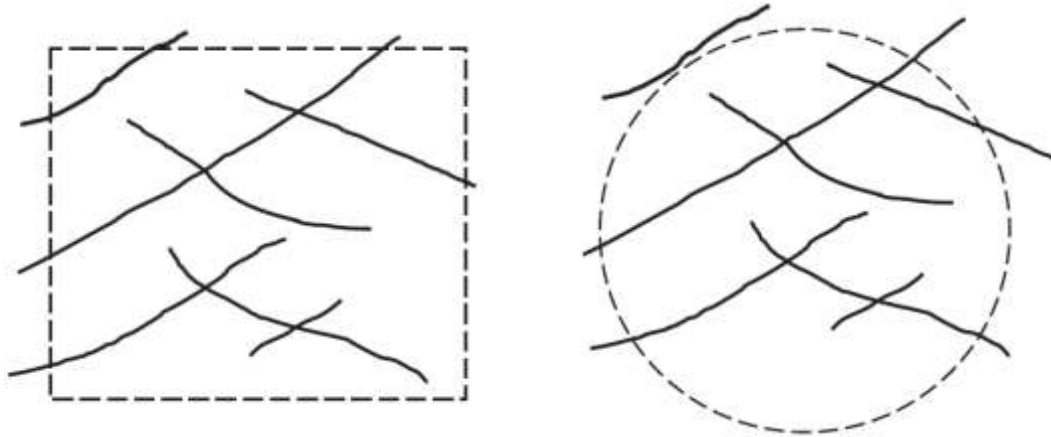
شکل ۲-۳: نحوه برداشت ناپیوستگی‌ها به روش خط برداشت [۱۴]

در این شکل d_i و d_j فاصله برخورد ناپیوستگی تا ابتدای خط برداشت و α زاویه امتداد ناپیوستگی با خط برداشت هستند. L و S نیز به ترتیب طول ناپیوستگی و فاصله دو ناپیوستگی می‌باشند. به دلیل اینکه تعداد زیادی از ناپیوستگی‌ها با خط برداشت برخورد نمی‌کنند، دقت برداشت ناپیوستگی‌ها در این روش کم است.

روش دیگر در برداشت ناپیوستگی‌ها استفاده از برداشت در یک محیط مشخص از توده سنگ است. در این روش یک محیط به شکل مربع یا دایره با ابعاد مشخص از توده سنگ انتخاب و خصوصیات تمامی

^۱ Scanline

ناپیوستگی‌های موجود در این محیط اندازه‌گیری می‌شود. این روش به پنجره برداشت^۱ معروف است. در شکل (۴-۲) پنجره برداشت مربعی و دایره‌ای مشاهده می‌شود. برای محاسبه فاصله‌داری، شیب و جهت شیب ناپیوستگی‌ها از مغزه‌های حفاری یا عکس‌های تهیه شده از آن‌ها نیز می‌توان استفاده کرد.



شکل ۴-۲: برداشت ناپیوستگی‌ها در یک محیط مربعی و دایره‌ای [۱۴]

۳-۲ نوع و شکل مختلف بلوک‌های توده سنگ

بلوک‌های ایجاد شده توسط دسته درزه‌ها و سایر ناپیوستگی‌ها با توجه به فاصله بین درزه‌ها، شیب درزه‌ها و تعداد درزه‌های نامرتب دارای شکل‌های و اندازه‌های متفاوتی هستند. دیرمن^۲ با توجه به ویژگی‌های مختلف سنگ‌ها یک سیستم تقسیم بندی مطابق با شکل (۵-۲) را برای انواع بلوک‌های توده سنگ ارائه داده است. در این طبقه‌بندی بلوک‌ها به انواع بلوک‌های چندوجهی^۳، هم‌بعد^۴، منشوری^۵، لوحی^۱ یا تخته‌ای، لوزی شکل^۲ و ستونی^۳ تقسیم می‌شوند [۱]. پالمستروم نیز مطابق

^۱ Windows Mapping

^۲ Dearman

^۳ Polyhedral Blocks

^۴ Equidimensional Blocks

^۵ Prismatic Blocks

شکل (۶-۲) بلوک ایجاد شده توسط درزه‌ها را با توجه به درزه‌های تشکیل دهنده آن به انواع بلوک‌های هم‌بعد^۴، تخت^۵، طویل^۶، و تخت و طویل تقسیم بندی کرده است. شکل بلوک‌ها بستگی زیادی به فاصله‌داری درزه‌ها در توده سنگ دارد. برای بررسی شکل بلوک‌ها در توده سنگ از فاکتور شکل بلوک (β) استفاده می‌شود. این فاکتور از طریق رابطه (۱-۲) بدست می‌آید [۱]:

$$\beta = \frac{(\alpha_2 + \alpha_2 * \alpha_3 + \alpha_3)^3}{(\alpha_2 * \alpha_3)^2} \quad (۱-۲)$$

که در آن α_2 از تقسیم فاصله متوسط درزه‌ها (S_2) بر کمترین فاصله (S_1) و α_3 از تقسیم بیشترین فاصله بین درزه‌ها (S_3) بر کمترین فاصله بدست می‌آیند.

برای توده سنگ با سه دسته درزه مرتب مقادیر فاکتور شکل بلوک با استفاده از شکل (۵-۲) بدست می‌آید. برای بلوک‌هایی که بیش از ۶ وجه داشته باشند محاسبه فاکتور شکل بلوک دشوار است. برای این بلوک‌ها رابطه (۲-۲) پیشنهاد شده است [۱]:

$$\beta = 20 + \frac{7a_3}{a_1} \quad (۲-۲)$$

که در آن a_1 کمترین و a_3 بیشترین بعد بلوک است. برای بلوک‌های تخت نیز مقدار فاکتور شکل بلوک ۳۶ در نظر گرفته می‌شود.

¹ Tabular Blocks

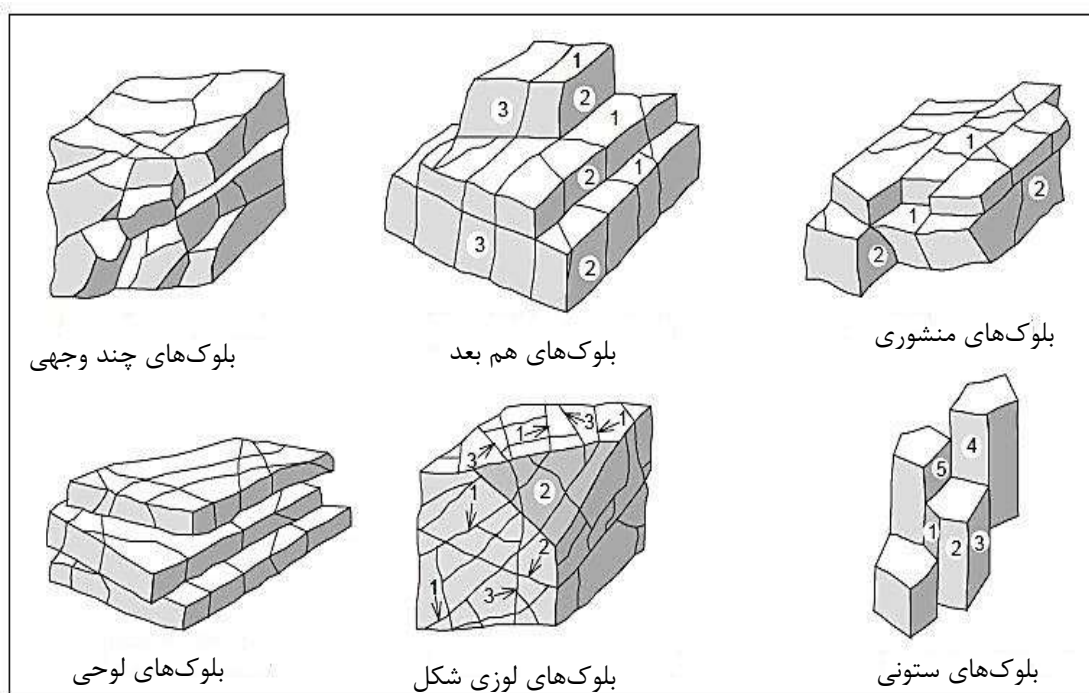
² Rhombohedral Blocks

³ Columnar Blocks

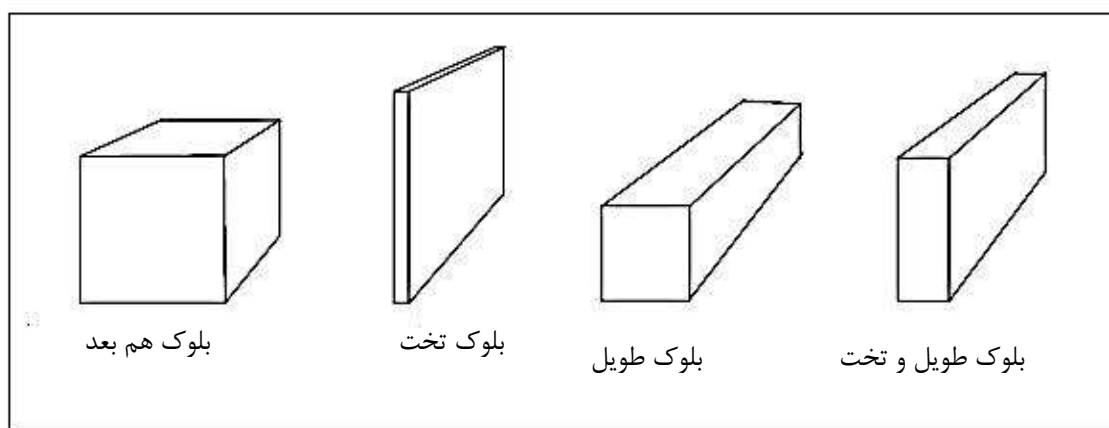
⁴ Compact Block

⁵ Flat Block

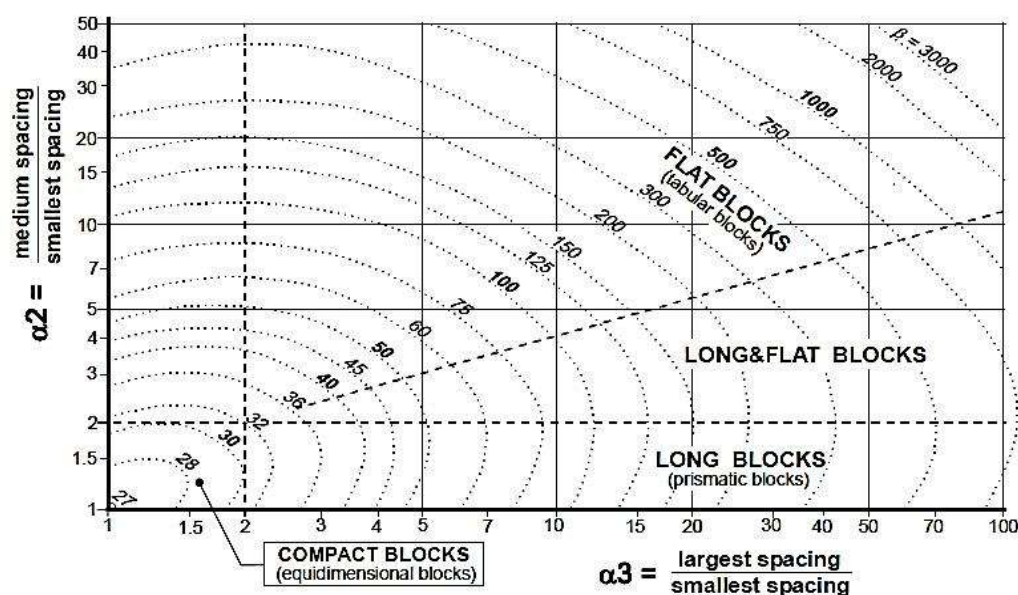
⁶ Long Block



شکل ۲-۵: تقسیم بندی بلوک‌ها [۱]



شکل ۲-۶: مهمترین بلوک‌های موجود در توده سنگ [۱]



شکل ۲-۷: مقادیر فاکتور شکل بلوک برای توده‌سنگ با سه دسته درزه [۱]

۴-۲ توده‌سنگ و توزیع حجم بلوک‌های آن

با توجه به این که انواع بلوک‌ها در توده‌سنگ تحت تأثیر چگالی درزه‌ها به وجود می‌آیند، دانستن توزیع حجم بلوک‌های توده‌سنگ می‌تواند درک مناسبی از بلوک‌های تشکیل دهنده آن باشد. با داشتن توزیع حجم بلوک‌ها می‌توان پیش بینی بهتری برای نگهداری بلوک‌های توده‌سنگ یا بلوک‌دهی آن در سنگ‌های ساختمانی و ... داشت. برای محاسبه هندسه بلوک‌های توده‌سنگ ابتدا باید ناپیوستگی‌ها یا درزه‌های موجود در آن مشخص شوند.

از لحاظ تئوری اگر در یک مسئله تعداد ناپیوستگی‌های موجود در یک توده‌سنگ کم باشد، تعیین خصوصیات سیستم ناپیوستگی به سادگی و با برداشت‌های صحرائی همه ناپیوستگی‌ها صورت می‌پذیرد. برای نمایش صریح تمام ناپیوستگی‌ها (شامل محل، جهت یابی، شکل و اندازه و میزان باز شدگی آن‌ها) و بلوک‌های تشکیل شده توسط آن‌ها می‌توان به طور مستقیم خصوصیات همه ناپیوستگی‌ها را تعیین و از آن‌ها برای تشکیل یک مدل هندسی معین استفاده کرد. اما این فرایند،

فرایندی غیر عملی است و به ندرت مورد استفاده قرار می‌گیرد. چون در مسائل کاربردی مطالعه تعداد بسیار زیاد ناپیوستگی‌ها با اندازه‌های مختلف مورد نیاز می‌باشد. یکسان نبودن خصوصیات ناپیوستگی‌ها باعث بوجود آمدن عدم قطعیت در پارامترهای شکل، اندازه و توزیع فضایی مربوط به ناپیوستگی‌ها می‌شود. روش معمول برای این منظور، در نظر گرفتن صرفاً ناپیوستگی‌های بزرگ مقیاس، به تعداد محدود به عنوان ناپیوستگی‌های منفرد و مجزا می‌باشد و دیگر ناپیوستگی‌های توده سنگ نیز باید از طریق روش‌های مختلف شبیه سازی کرد.

۲-۴-۱ روش‌های شبیه سازی هندسی درزه‌های توده سنگ

نبود اطلاعات کافی و غیر ممکن بودن برداشت همه ناپیوستگی‌ها باعث می‌شود برای دسترسی به اطلاعات توده سنگ، درزه‌ها را شبیه سازی کرد. برای این کار مراحل زیر توسط اسایب^۱ [۱۹] پیشنهاد شده است:

- ۱- برداشت ناپیوستگی‌ها
- ۲- پیاده سازی بر روی استریونت
- ۳- تشخیص درزه‌ها یا ناپیوستگی‌های غالب
- ۴- شبیه سازی و تولید ناپیوستگی‌های جدید
- ۵- مدل سازی توده سنگ و تعیین هندسه بلوک‌های توده سنگ

ابتدا ناپیوستگی‌های توده سنگ توسط روش‌های مختلف برداشت می‌شود. امروزه در پروژه‌های بزرگ برای بالا بردن دقت برداشت از روش‌های تصویر برداری و استفاده از مغزه‌های حفاری برای برداشت درزه‌ها استفاده می‌شود.

¹ Essaieb

مرحله بعد آوردن ناپیوستگی‌های برداشت شده بر روی صفحه استریونت برای تشخیص دسته درزه‌های غالب و پارامترهای آن‌ها می‌باشد. با توجه به صفحه استریونت ناپیوستگی‌های غالب تعیین می‌شوند. تعداد ناپیوستگی‌ها، شیب و جهت شیب، امتداد و فاصله‌داری آن‌ها از این طریق محاسبه می‌شود.

به دلیل اینکه همه ناپیوستگی‌های توده سنگ غیر قابل برداشت هستند، با استفاده از ناپیوستگی‌های برداشت شده و توابع توزیع احتمالی ناپیوستگی‌های جدید شبیه سازی می‌شوند. در نهایت با استفاده از ناپیوستگی‌های اختصاص داده شده به توده سنگ هندسه بلوک‌های آن تعیین می‌شود. روش‌های مختلفی برای شبیه سازی هندسی ناپیوستگی‌ها وجود دارد که عبارتند از:

- روش قطعی و محدود

- روش زمین آمار

- روش آماری

در ادامه روش‌های مذکور به اختصار توضیح داده می‌شوند [۲۰]:

۲-۴-۱ روش قطعی و محدود

در این روش فقط تعداد محدودی از ناپیوستگی‌های توده سنگ که اغلب ناپیوستگی‌های اصلی می‌باشند مورد مطالعه قرار می‌گیرند. گاهی هر دسته درزه نیز تنها با شیب و جهت شیب مشخص می‌شود که باعث ایجاد بلوک‌هایی با ابعاد ماکزیمم می‌شود.

۲-۴-۲ روش زمین آماری

پارامترهای هندسی ناپیوستگی‌ها در سنگ می‌تواند توسط روش‌های زمین آماری تخمین زده شوند. این روش این امکان را می‌دهد تا وابستگی مکانی پارامترها بررسی شود. به عنوان مثال پارامتر

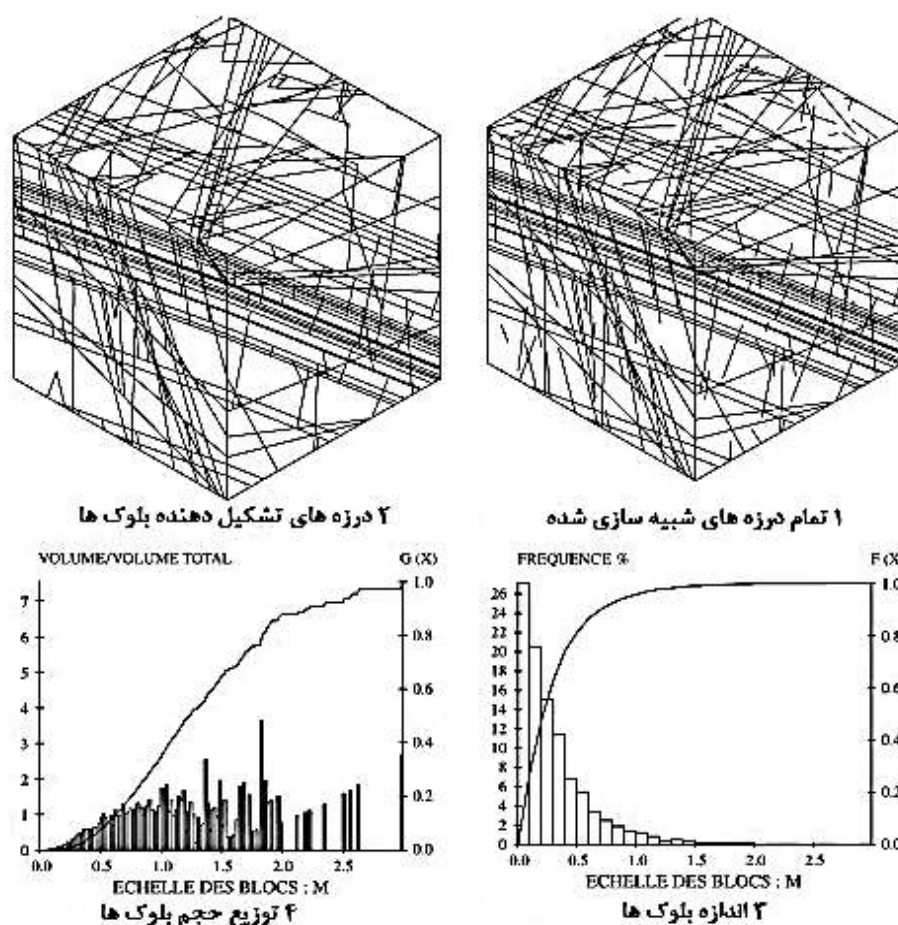
جهت‌داری یک درزه به طور عمومی نزدیکی زیادی به جهت‌داری درزه‌ای دیگر در نزدیکی درزه اول دارد. این ویژگی‌ها وابستگی مکانی نامیده می‌شوند. پارامترهای هندسی که امکان بررسی وابستگی آن‌ها وجود دارد، جهت‌داری، ابعاد و بازشدگی درزه‌ها هستند. در این روش ابتدا موقعیت مکانی درزه‌ها باید تعیین شود و سپس با استفاده از اطلاعات موجود در اطراف هر مرکز درزه مشخصات و پارامترهای زمین آماری تخمین زده و درزه‌های توده‌سنگ شبیه سازی می‌شوند.

۲-۴-۱-۳ روش آماری

در روش‌های آماری باید الگوی مناسب برای درزه‌های توده‌سنگ با توجه به درزه‌های برداشت شده معرفی شود. در این روش‌ها تغییرات پارامترها و مختصات آن مورد توجه قرار می‌گیرند و اثر آن‌ها بر مدل منظور می‌شود. مدل‌سازی آماری معمولاً در سه سیستم اصلی که در زیر آمده است، صورت می‌گیرد:

- سیستم دیسک‌های تصادفی
- سیستم ناپیوستگی‌های ترتیبی نامحدود
- شبیه سازی مجموعه بلوکی

شکل (۲-۸) نمونه‌ای از توده‌سنگ و توزیع حجمی آن را نشان می‌دهد. در این شکل مراحل تعیین توزیع حجم بلوک‌های توده‌سنگ به صورت شماتیک نشان داده شده است. ابتدا تمام ناپیوستگی‌های توده‌سنگ شبیه سازی می‌شوند و پس از آن ناپیوستگی‌هایی که در تشکیل بلوک‌ها تأثیری ندارند حذف می‌شوند. برای تعیین توزیع حجم بلوک‌ها نیاز به تعیین حجم تک تک بلوک‌های توده‌سنگ می‌باشد که در این شکل این مراحل نشان داده شده‌اند.



شکل ۲-۸: نمونه‌ای از توده سنگ با توزیع حجمی بلوک‌های آن [۱۹]

۲-۵ محاسبه حجم بلوک‌های توده سنگ با پارامترهای هندسی

درزه‌ها و ناپیوستگی‌های موجود در توده سنگ باعث ایجاد بلوک‌هایی با ابعاد متفاوت در توده سنگ می‌شوند. یکی از روش‌های محاسبه حجم این بلوک‌ها استفاده از پارامترهای هندسی است که خود به ۴ روش محاسبه حجم بلوک‌ها سر زمین یا با استفاده از مغزه‌های حفاری، استفاده از فاکتور حجمی بلوک، استفاده از فاصله‌داری درزه‌ها و استفاده از فاکتور تعداد درزه حجمی بلوک تقسیم بندی می‌شوند. این روش‌ها در ادامه توضیح داده می‌شوند [۱]:

۲-۵-۱ محاسبه سر زمین یا با استفاده از مغزه‌های حفاری

در این روش ابعاد بلوک‌هایی که در سطح زمین نمایان هستند را اندازه گیری کرده و حجم آن‌ها را محاسبه می‌کنند. این روش مخصوص مناطق خاص که بلوک‌ها در سطح زمین نمایان هستند و دسترسی به آن‌ها آسان است، می‌باشد. در این روش همانند توده‌سنگ نشان داده شده در شکل (۲-۹) ابعاد بلوک‌ها را با استفاده از متر اندازه گیری می‌کنند. با استفاده از مغزه‌های حفاری نیز می‌توان طول ناپیوستگی‌ها را اندازه گرفت و حجم بلوک‌ها را به طور تقریبی محاسبه کرد.



شکل ۲-۹: بلوک‌های نمایان در سطح زمین که به راحتی می‌توان حجم آن‌ها را محاسبه کرد [۵]

۲-۵-۲ استفاده از فاکتور حجمی بلوک (I_b)

فاکتور حجمی بلوک توسط ISRM در سال ۱۹۷۸ معرفی شد و معرف حجم بلوک‌های توده‌سنگ است. این فاکتور توسط انتخاب تعدادی بلوک شاخص و میانگین‌گیری ابعاد آن‌ها بدست می‌آید. در حالت خاص برای یک توده‌سنگ با سه دسته درزه فاکتور حجمی بلوک (I_b) به صورت زیر محاسبه می‌شود:

$$I_b = \frac{S_1 + S_2 + S_3}{3} \quad (2-3)$$

که در آن S_1 ، S_2 و S_3 فاصله بین درزه‌ها هستند.

۲-۵-۳ استفاده از فاصله‌داری درزه‌ها

در این روش با استفاده از فاصله‌داری درزه‌ها حجم یک بلوک به عنوان نماینده بلوک‌های توده‌سنگ محاسبه می‌شود. در بیشتر مواقع تعداد دسته درزه‌های غالب در یک منطقه سه دسته است و بیشتر فرمول‌های ارائه شده بر این مبنا می‌باشد. رابطه (۲-۴) نحوه محاسبه حجم بلوک (V_b) با استفاده از فاصله‌داری درزه‌ها را نشان می‌دهد.

$$V_b = S_1 \times S_2 \times S_3 \times (\sin \gamma_1 \times \sin \gamma_2 \times \sin \gamma_3) \quad (2-4)$$

که در آن S_1 ، S_2 و S_3 فاصله بین دسته درزه‌ها و γ_1 ، γ_2 و γ_3 زاویه بین دسته درزه‌ها هستند.

در رابطه‌های زیر به ترتیب محاسبه حجم بلوک برای یک و دو دسته درزه مشخص شده است.

$$V_b = S_1 \times 5S_1 \times 10S_1 \quad (2-5)$$

$$V_b = S_1 \times S_2 \times 5S_1 \quad (2-6)$$

۲-۵-۴ استفاده از فاکتور تعداد درزه حجمی (J_v)

در این روش ابتدا روش محاسبه فاکتور تعداد درزه حجمی بررسی می‌شود و با استفاده از این فاکتور حجم بلوک‌ها به صورت تخمینی محاسبه می‌شود. فاکتور تعداد درزه حجمی از روش‌های زیر محاسبه می‌شود:

۲-۵-۴-۱ محاسبه J_v بر اساس فاصله‌داری درزه‌ها

$$J_v = \frac{1}{S_1} + \frac{1}{S_2} + \frac{1}{S_3} + \dots \quad (2-7)$$

که در آن S_1 ، S_2 و S_3 فاصله بین درزه‌ها و J_v فاکتور تعداد درزه حجمی است. برای مناطقی که درزه‌های نامرتب یا تصادفی نیز حضور داشته باشند، رابطه (۲-۸) توصیه می‌شود.

$$J_v = \frac{1}{s_1} + \frac{1}{s_2} + \frac{1}{s_3} + \dots + \frac{Nr}{5} \quad (8-2)$$

که در آن Nr تعداد درزه‌های تصادفی است.

۲-۴-۵-۲ محاسبه J_v بر اساس فرکانس دو بعدی درزه

این روش بر اساس اندازه‌گیری‌های دو بعدی بر روی سطح زمین است. فرکانس دوبعدی درزه (N_a)، تعداد درزه‌های اندازه‌گیری شده در سطح را نشان می‌دهد. به دلیل این که طول درزه‌ها ممکن است بیشتر یا کمتر از منطقه مورد مطالعه باشد، تصحیح زیر باید اعمال شود:

$$N_a = \left(\frac{1}{\sqrt{A}}\right) \sum (n_{a_i} \times L_i) + N_{a_j} \quad (9-2)$$

که در آن A سطح منطقه مورد مطالعه، N_{a_j} و n_{a_i} تعداد درزه‌هایی که طول آن‌ها به ترتیب بیشتر و کمتر از طول منطقه مورد مطالعه باشد و L_i طول درزه‌ها است. و در نهایت فاکتور تعداد درزه حجمی از طریق رابطه زیر بدست می‌آید:

$$J_v = N_a \times K_a \quad (10-2)$$

که در آن J_v فاکتور تعداد درزه حجمی، N_a فرکانس دوبعدی درزه‌ها و K_a ضریب همبستگی است که بین ۱ تا ۲/۵ تغییر می‌کند و میانگین آن ۱/۵ است و بیشترین مقدار زمانی اتخاذ می‌شود که طول منطقه مورد مطالعه برابر با طول درزه اصلی باشد.

۲-۴-۵-۳ محاسبه J_v در امتداد خط برداشت یا مغزه حفاری

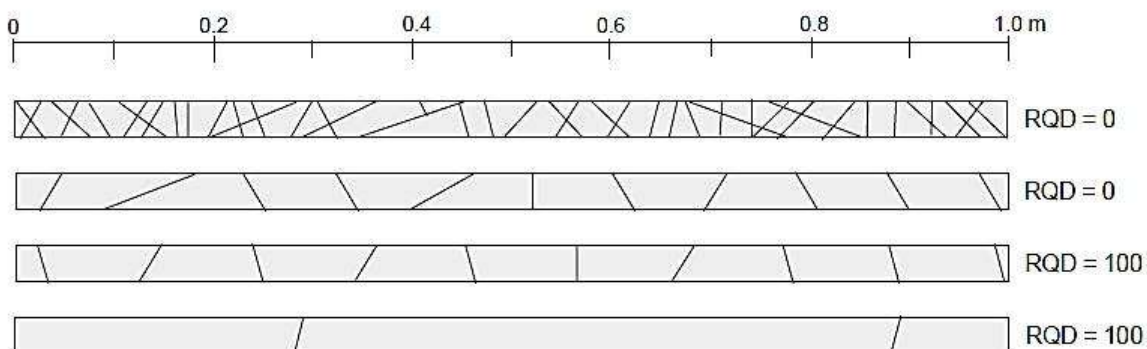
در این روش تعداد درزه‌هایی که در امتداد یک خط مشخص یا تعداد درزه‌های موجود در مغزه حفاری را شمارش می‌کنند.

$$J_v = N_l \times K_l \quad (11-2)$$

که در آن N_1 تعداد درزه‌ها و K_1 ضریب همبستگی است که از مقدار $1/25$ تا 6 تغییر می‌کند و متوسط آن 2 است. البته مقدار فاکتور تعداد درزه حجمی که به این شیوه به دست می‌آید دقیق نیست و خطای زیادی دارد.

۴-۴-۵-۲ محاسبه J_v با استفاده از شاخص کیفیت سنگ (RQD)

درصد طول قطعه‌های بیشتر از 10 سانتی متر در طول مشخصی از یک مغزه حفاری را شاخص کیفیت سنگ گویند. شاخص کیفیت سنگ نشان دهنده میزان خرد شدگی سنگ می‌باشد که دارای مقدار صفر برای سنگ‌های کاملاً خرد شده و مقدار 100 برای سنگ‌های یک پارچه است. رابطه شاخص کیفیت سنگ و درزه‌ها به صورت گرافیکی در شکل (۲-۱۰) نشان داده شده است.



شکل ۲-۱۰: رابطه شاخص کیفیت سنگ و درزه‌ها [۱]

مقدار فاکتور تعداد درزه حجمی با استفاده از شاخص کیفیت سنگ از رابطه (۲-۱۲) بدست می‌آید.

$$J_v = \frac{115 - RQD}{3.3} \quad (2-12)$$

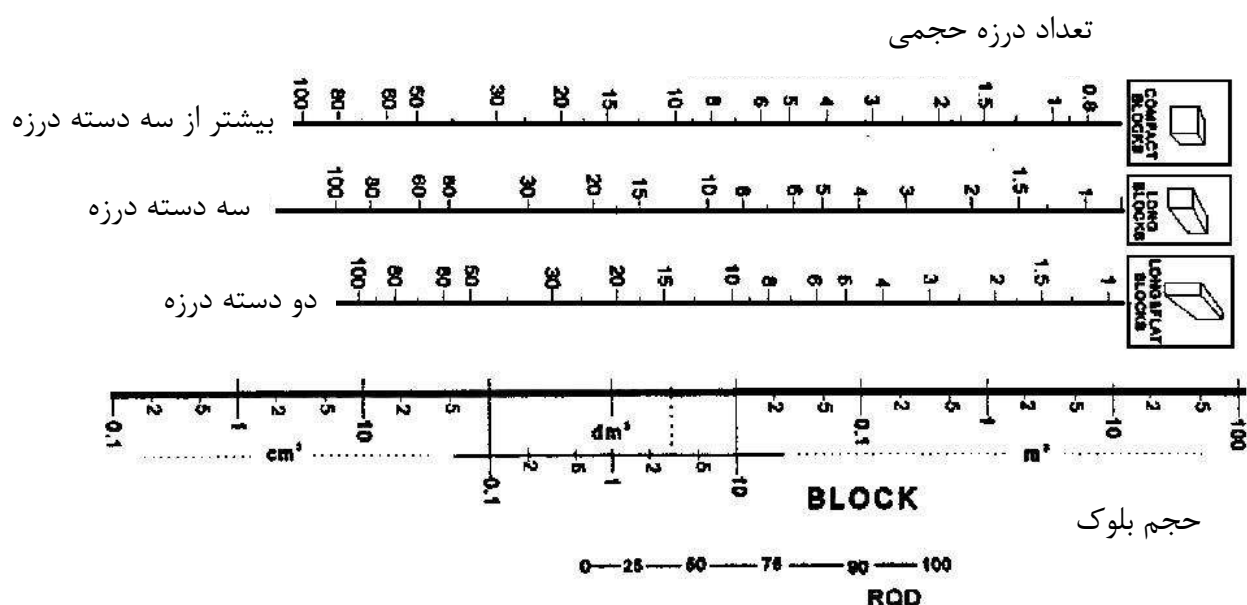
که در آن RQD شاخص کیفیت سنگ و J_v فاکتور تعداد درزه حجمی است و برای J_v بیشتر از 35 ، RQD برابر صفر و برای J_v کمتر از $4/5$ ، RQD برابر 100 است.

پس از محاسبه فاکتور تعداد درزه حجمی می‌توان حجم بلوک‌ها را از طریق رابطه (۲-۱۳) محاسبه کرد.

$$V_b = \beta * J_v * \frac{1}{\sin \gamma_1 * \sin \gamma_2 * \sin \gamma_3} \quad (2-13)$$

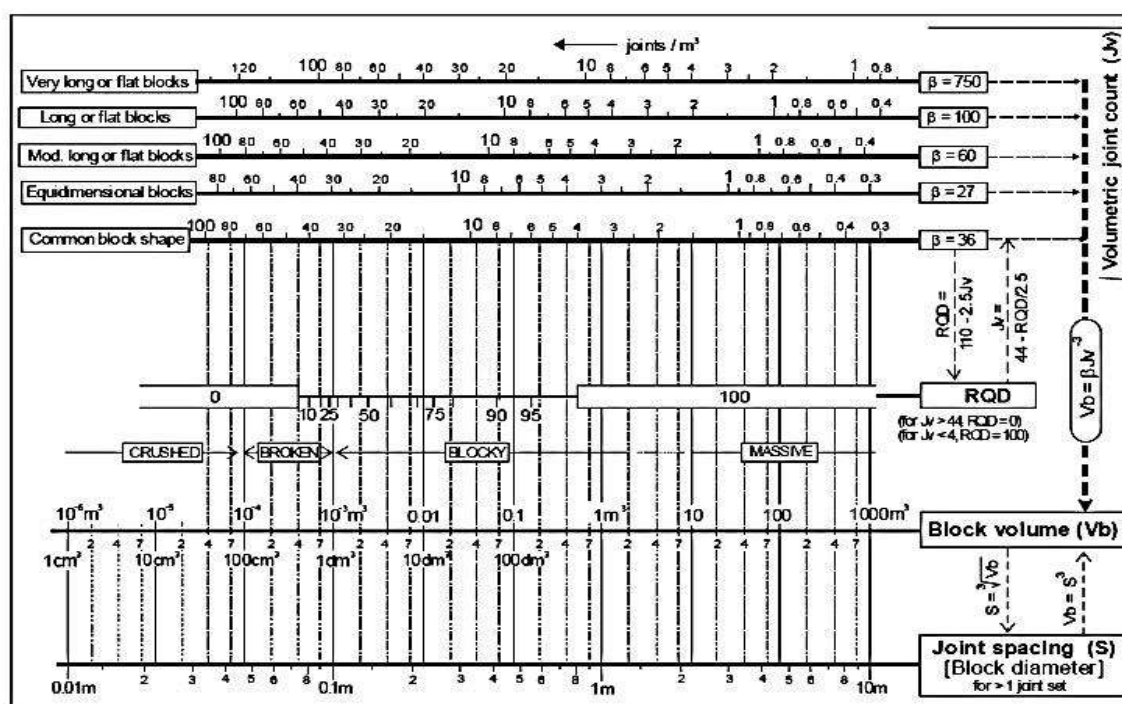
که در آن β فاکتور تغییر شکل، J_v فاکتور تعداد درزه حجمی و γ زاویه بین درزه‌ها است.

هر چند شاخص کیفیت سنگ معیار مناسبی برای اندازه بلوک‌های موجود در توده سنگ است، اما رابطه مناسبی که این دو پارامتر را با هم ارتباط دهد وجود ندارد. شکل (۲-۱۱) نشان دهنده رابطه بین شاخص کیفیت توده سنگ و تعداد درزه حجمی با حجم بلوک‌ها به صورت نمودار است. در این شکل با داشتن تعداد دسته درزه حجمی یا شاخص کیفیت توده سنگ میانگین حجم بلوک‌های موجود در توده سنگ را می‌توان بدست آورد.



شکل ۲-۱۱: رابطه بین شاخص کیفیت توده سنگ و تعداد درزه حجمی با حجم بلوک‌ها [۲۱]

شکل (۲-۱۲) ارتباط بین روش‌های مختلف اندازه‌گیری حجم بلوک‌ها با استفاده از پارامترهای هندسی را به صورت نمودار نشان می‌دهد.



شکل ۲-۱۲: ارتباط بین روش‌های مختلف اندازه‌گیری حجم بلوک‌ها با استفاده از پارامترهای هندسی [۲۱]

۲-۶ نتیجه‌گیری

توده‌سنگ متشکل از ماده‌سنگ و ناپیوستگی‌ها است که در این فصل به بررسی آن‌ها پرداخته شد. اطلاع دقیق از هندسه توده‌سنگ درزه‌دار با توجه به کاربردهای متعدد که اشاره شد امری ضروری می‌باشد. از سال‌های گذشته مطالعات زیادی برای تعیین حجم بلوک‌های توده‌سنگ انجام شده است که در برخی از روش‌ها از پارامترهای هندسی برای محاسبه حجم بلوک‌های توده‌سنگ استفاده می‌شود. در این فصل چهار روش از این روش‌ها بررسی شد و با توجه به این که در این روش‌ها درزه‌ها سراسری و موازی فرض می‌شوند. این روش‌ها دارای خطای بالا بوده و امروزه کمتر مورد استفاده قرار می‌گیرند.

فصل سوم

تعیین هندسه بلوک‌های

توده‌سنگ

۱-۳ مقدمه

تعیین حجم بلوک‌های موجود در توده‌سنگ توسط روش‌های مختلفی صورت می‌گیرد؛ در همه این روش‌ها بین مقادیر تخمین زده شده و مقادیر واقعی اختلاف وجود دارد. این اختلافات یا خطاها ناشی از طبیعت زمین و عدم امکان برداشت همه ناپیوستگی‌های توده‌سنگ می‌باشد. در سال‌های اخیر با توجه به پیشرفت تکنولوژی برای کاهش این خطاها روش‌های دقیق‌تری برای برداشت ناپیوستگی‌ها و تعیین حجم بلوک‌ها ابداع شده است. برای تعیین هندسه بلوک‌های توده‌سنگ همانطور که در فصل قبل توضیح داده شد ابتدا ناپیوستگی‌ها شبیه سازی می‌شوند و هندسه بلوک‌های آن از طریق الگوریتم‌های مختلف و روش‌های هندسی محاسبه می‌شود. در بعضی از الگوریتم‌ها توده‌سنگ در دو بعد و در برخی دیگر در سه بعد مدل‌سازی می‌شود.

در این فصل تعیین هندسه بلوک‌های حاصل از برخورد چند خط با استفاده از الگوریتم‌های مختلف توضیح داده می‌شود.

۲-۳ تعیین هندسه بلوک‌ها توسط الگوریتم‌های مختلف

پس از شبیه سازی درزه‌ها در توده‌سنگ باید حجم بلوک‌های تشکیل شده توسط این درزه‌ها محاسبه شود. در همه الگوریتم‌ها ناپیوستگی‌ها به صورت خطوط در دو بعد و صفحات در سه بعد فرض می‌شوند و با استفاده از عملیات هندسی بلوک‌های تشکیل شده شناسایی می‌شوند. منظور از شناسایی بلوک‌ها تعیین مختصات رئوس و اضلاع بلوک است. در سال‌های اخیر الگوریتم‌هایی با جواب قطعی مانند الگوریتم مبتنی بر تکنیک‌های عددی و الگوریتم مبتنی بر تئوری گرافی و الگوریتم‌هایی ابتکاری مانند الگوریتم مبتنی بر ماتریس‌ها برای محاسبه مشخصات هندسی بلوک‌های حاصل از تلاقی خطوط یا صفحات ارائه شده است. در ادامه مراحل شناسایی بلوک‌ها برای تعدادی از

الگوریتم‌های دو بعدی تعیین هندسه بلوک‌های توده‌سنگ توضیح داده می‌شود. این الگوریتم‌ها عبارتند از:

- الگوریتم تئوری گرافی
- الگوریتم تکنیک‌های عددی
- الگوریتم شی مبتنی بر گراف‌های جهت‌دار
- الگوریتم تئوری بلوکی مبتنی بر روش‌های برداری
- الگوریتم مبتنی بر ماتریس‌ها

مراحل مختلف الگوریتم، معایب و مزایای هر یک از الگوریتم‌ها در ادامه بررسی می‌شود.

۱-۲-۳ الگوریتم تئوری گرافی

برای محاسبه مساحت بلوک‌ها در دو بعد باید چند ضلعی‌های حاصل از تلاقی چند خط تعیین شوند. در الگوریتم ارائه شده توسط فریرا و همکاران تشخیص یک چند ضلعی تشکیل شده از تلاقی چند خط با استفاده از چهار گام اصلی صورت می‌گیرد، که این چهار گام عبارتند از [۶]:

- تعیین محل تلاقی خط‌ها با استفاده از الگوریتم بنتلی-اوتام^۱ [۲۲].
- رسم گراف‌های بسته که رئوس آن محل تلاقی خط‌ها و یال‌های آن قسمتی از خط بین دو محل تلاقی است.
- یافتن کوچک‌ترین چرخه مبنا (MCB^۲) از گراف‌های مرحله قبل با استفاده از الگوریتم ارائه شده توسط هورتون^۱ [۲۳].

¹ Bentley-Ottmann

² Minimum Cycle Basis

- ساختن چند ضلعی‌ها با استفاده از گراف‌های مرحله قبل.

۱-۱-۲-۳ تبدیل خطوط و محل برخورد آن‌ها به گراف

در برخورد چند خط با یکدیگر امکان دارد تعداد زیادی تقاطع بین آن‌ها وجود داشته باشد که تعداد محدودی از آن‌ها چند ضلعی‌ها را تشکیل می‌دهند؛ بنابراین لازم است این نقاط شناسایی شوند. در مرحله اول بدست آوردن N محل تلاقی از M خط یکی از مسأله‌های بنیادی در هندسه است. برای حل مسأله فوق بالابان^۲ [۲۴]، چازل و ادلبرونر^۳ [۲۵]، کلارکسون^۴ و شور^۵ [۲۶] الگوریتم‌هایی با دشواری‌های زمانی و حجمی متفاوتی ارائه داده‌اند. در سال ۱۹۷۹ بنتلی و اوتام ساده‌ترین الگوریتم را برای حل این مسئله ارائه نمودند که پس از سال‌ها هنوز این الگوریتم مورد استفاده قرار می‌گیرد. در مرحله بعد باید با استفاده از نقاط تقاطع خطوط کوچکتر بدون تقاطع تولید کرد. برای این کار از الگوریتم باروشکا و همکاران^۶ [۲۷] که شکل توسعه یافته الگوریتم بنتلی اوتام است، استفاده می‌شود. این الگوریتم گراف تشکیل شده در اثر برخورد چند خط را محاسبه می‌کند. رئوس گراف محل تلاقی و نقاط انتهایی خطوط و یال‌ها پاره خط‌ها هستند. شکل (۱-۳) نمونه‌ای از برخورد چند خط در یک صفحه را نشان می‌دهد و شکل (۲-۳) گراف تشکیل شده توسط این خطوط را نشان می‌دهد.

¹ Horton

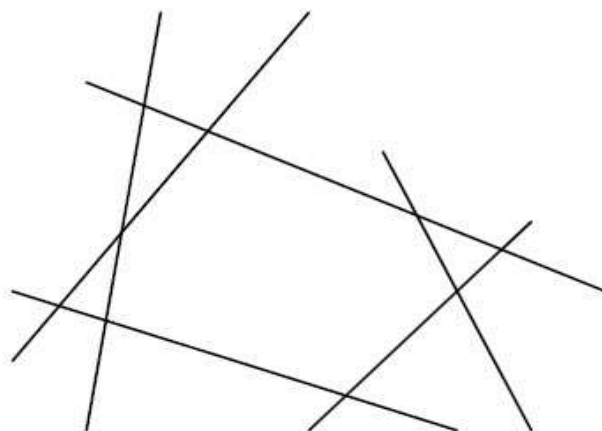
⁵ BaLaban

³ Chazelle & Edelsbrunner

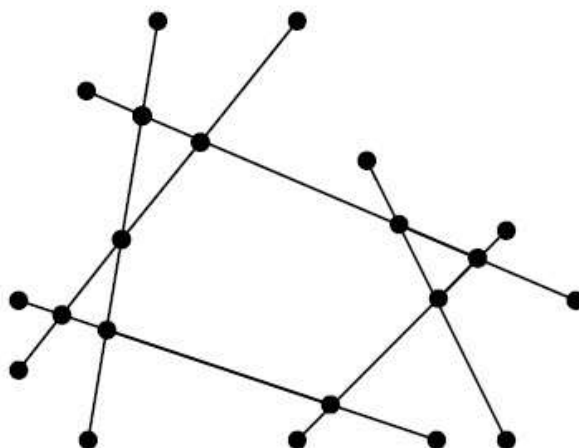
⁷ Clarkson

⁵ Shor

⁶ Bartuschka et al



شکل ۳-۱: نمونه‌ای از برخورد چند خط در صفحه [۶]

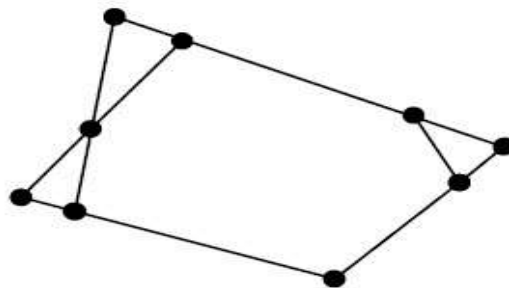


شکل ۳-۲: گراف تشکیل شده از محل برخورد خطوط و نقاط پایانی آن‌ها [۶]

در این شکل نقاط برخورد رئوس گراف و خطوط بین آن‌ها یال‌های گراف را تشکیل می‌دهند.

۲-۱-۲-۳ یافتن چند ضلعی‌ها با استفاده از گراف‌ها

پس از تعیین گراف‌ها مرحله بعد یافتن چند ضلعی‌ها با استفاده از این گراف‌ها است. اولین الگوریتم توسط سیسلو^۱ برای این منظور ارائه شد [۶]. پس از آن داگروسوز و کریشنامورتی^۲ [۲۸] الگوریتم ساده‌تری را برای یافتن گراف‌های بسته از بین چند گراف ارائه نمودند. با توجه به این که هدف بدست آوردن کوچکترین گراف‌های بسته است، نیاز نیست که تمام گراف‌های بسته شناسایی شوند. بنابراین باید طبق الگوریتمی خاص کوچکترین چرخه‌های بسته در گراف مبنا مشخص شوند. هورتون اولین الگوریتم را برای این منظور ارائه داده است. الگوریتم بنتلی-اوتام الگوریتمی ساده‌تر برای یافتن کوچکترین چرخه مبنا می‌باشد. برای محاسبه کوچکترین چرخه‌ها باید کمترین فاصله بین رؤس مشخص شود که می‌توان از الگوریتم فلویید وارشال^۳ یا دیکسترا^۴ استفاده کرد [۲۹]. برای یافتن کوچکترین چرخه‌ها نیز از الگوریتم هورتون استفاده می‌شود. شکل (۳-۳) کوچکترین چرخه‌ها یا گراف‌های بسته را برای شکل (۱-۳) نشان می‌دهد.



شکل ۳-۳: کوچکترین چرخه‌ها یا گراف‌های بسته برای شکل (۱-۳) [۶]

^۱ Syslo

^۲ Dogrusoz & Krishnamoorthy

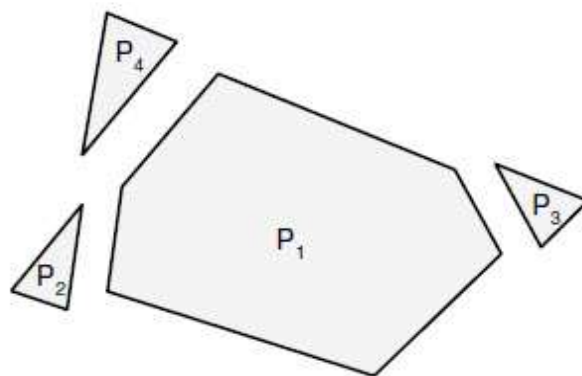
^۳ Floyd-Warshall

^۴ Dijkstra

۳-۱-۲-۳ ساختن چند ضلعی‌ها

مرحله نهایی در یافتن چند ضلعی‌ها ساختن آن‌ها از کوچکترین چرخه‌های تعیین شده در گراف اصلی است. اگر فرض شود Θ مجموعه‌ای از چند ضلعی‌ها و P مجموعه‌ای از چرخه‌های رئوس (V) باشد، مراحل ساختن چندضلعی‌ها به صورت زیر می‌باشد:

- مرحله اول: صفر در مجموعه Θ قرار می‌گیرد.
 - مرحله دوم: برای هر چرخه در مجموعه کوچکترین چرخه‌ها، چرخه جدید در پارامتر P قرار می‌گیرد.
 - مرحله سوم: برای هر v در رئوس این رأس به P اضافه می‌شود.
 - مرحله چهارم: چندضلعی P به مجموعه Θ اضافه می‌شود.
 - مرحله پنجم: و در نهایت چند ضلعی‌ها به عنوان خروجی تعیین می‌شوند.
- شکل (۳-۴) چندضلعی‌های حاصل از چرخه‌های بسته در گراف اصلی را نشان می‌دهد.



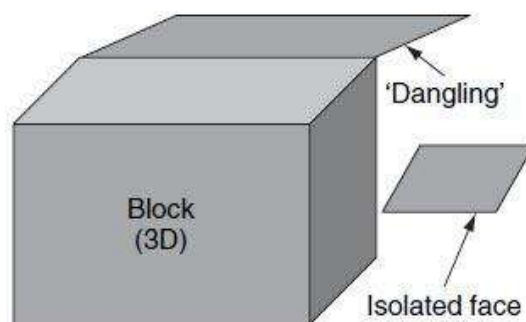
شکل ۳-۴: یافتن چند ضلعی‌ها با استفاده از کوچکترین چرخه‌ها [۶]

الگوریتم مورد استفاده در این روش طبق ادعای ارائه کننده بهینه‌ترین الگوریتم از لحاظ زمان و حجم مورد نیاز برای پردازش اطلاعات است. اما این روش از یافتن بلوک‌های دو بعدی به دلیل پیچیدگی نیاز به زمان زیاد برای پیاده سازی الگوریتم دارد.

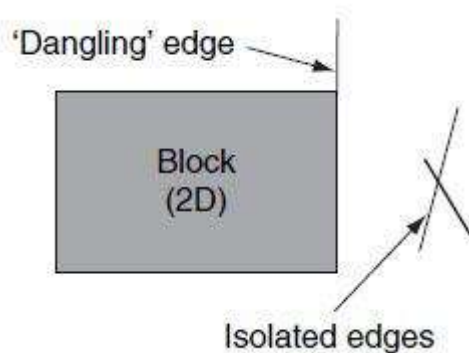
۲-۲-۳ الگوریتم مبتنی بر تکنیک‌های عددی

یکی از روش‌های تحلیل توده‌سنگ و تعیین بلوک‌های آن استفاده از روش‌های عددی است. تحلیل سنگ‌های شکسته به دلیل سیستم هندسی سه بعدی ناپیوستگی‌ها، ثابت نبودن زمینه سنگ و خصوصیات ناپیوستگی‌ها، حتی اگر سیستم ناپیوستگی به صورت دو بعدی در نظر گرفته شود، مشکل است. بیشتر تحقیقات صورت گرفته در این موضوع سیستم ناپیوستگی‌ها را به صورت دو بعدی در نظر گرفته‌اند. این روش‌ها از قانون‌های نظری خاص در تحقیقات و پیشرفت خود استفاده کرده‌اند. اما تحلیل آن‌ها در مورد مدل‌های سه بعدی برای شرایط خاص می‌باشد که به واقعیت نزدیک نیست. در ادامه روش تکنیک‌های عددی که توسط جینگ و استفانسون ارائه شده است توضیح داده می‌شود [۱۴]:

برای شناسایی بلوک‌ها یکی از مراحل حذف یا مرتب سازی ناپیوستگی‌ها است. برای مرتب سازی ناپیوستگی‌ها ابتدا باید سطوح منفرد در شبیه سازی سه بعدی و خطوط منفرد در شبیه سازی دو بعدی حذف شوند. سطوح و خطوط منفرد به صورت شماتیک در شکل‌های (۳-۵) و (۳-۶) نشان داده شده‌اند. فرمول Euler-Poincare یک فرمول ریاضی است که برای ارتباط سطوح و خطوط در یک شبکه متصل به کار می‌رود. این فرمول در رابطه (۳-۱) آورده شده است. به دلیل این که سطوح و خطوط منفرد در شبکه با دیگر اجزا متصل نیستند و نقشی در ساختن بلوک‌ها ندارند، فرمول Euler-Poincare با وجود آن‌ها قابل استفاده نیست.



شکل ۳-۵: سطوح آویزان و منفرد در شبیه سازی سه بعدی [۱۴]



شکل ۳-۶: خطوط آویزان و منفرد در شبیه سازی دو بعدی [۱۴]

یک شبکه مرتب شده ناپیوستگی ویژگی‌های زیر را دارد:

- در سطوح هر خط دو و فقط دو رأس دارد که از تقاطع با سایر ناپیوستگی‌ها به وجود آمده است.
- در بلوک‌ها هر صفحه فقط و فقط بین دو چند ضلعی مشترک است.
- اگر تعداد شبکه‌های فرعی مستقل به عنوان N_{sn} و تعداد رئوس (تقاطع) به عنوان N_v و تعداد خطوط (یال‌ها) به عنوان N_e و تعداد چند ضلعی‌های به عنوان N_f در نظر گرفته شود رابطه (۳-۱) باید برای این شبکه برقرار باشد.

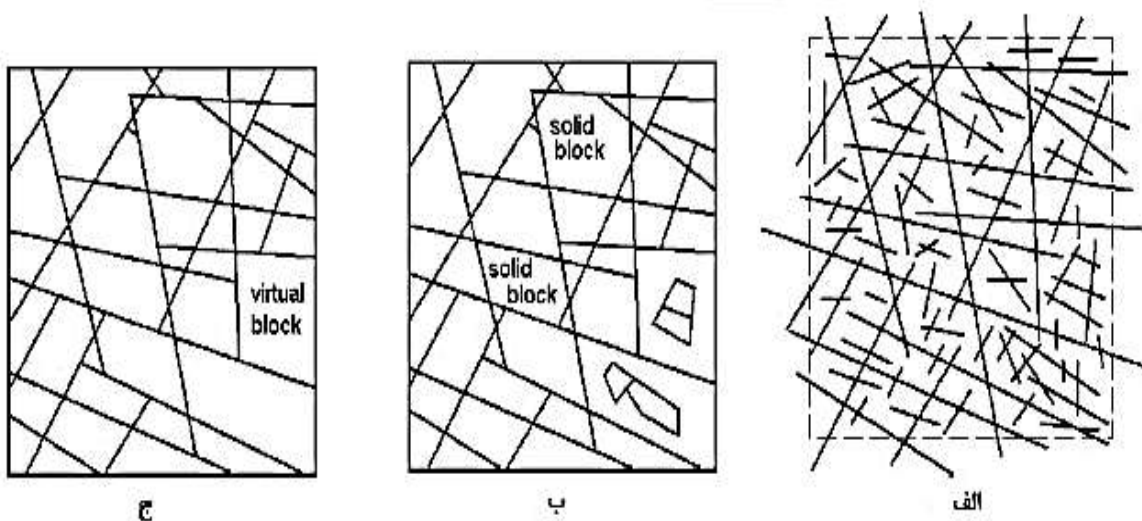
$$N_v + N_f - N_e = 1 + N_{sn} \quad (۳-۱)$$

رابطه (۱-۳) فرمول Euler-poincare برای سیستم‌های چند ضلعی در مکان شناسی است و یک شبکه متصل کامل باید در رابطه مذکور دارای جواب باشد. به بیان دیگر شبکه‌ای که در رابطه بالا دارای جواب نباشد شامل اجزای آویزان و منفرد است. این رابطه به عنوان مقیاسی برای روند و مراحل مرتب‌سازی شبکه ناپیوستگی به کار می‌رود. یافتن بلوک‌ها و مسیرهای جریان سیالات و مشخص کردن سیستم‌های فرعی منفصل، می‌تواند توسط یک عملگر مرزی انجام شود. شکل (۷-۳) یک شبکه ناپیوستگی دو بعدی و مرتب سازی آن را نشان می‌دهد. شکل (۷-۳ الف) سیستم ناپیوستگی‌های واقعی با ۱۲۶ تقاطع را نشان می‌دهد. نتایج حاصل از مرتب سازی شامل ۳۴ بلوک (سطح)، ۱۴۶ خط (بخش‌هایی از ناپیوستگی) و ۹۲ رأس (تقاطع) که نتیجه در ۳ شبکه (۱ شبکه اصلی ۲ شبکه فرعی اضافی مستقل) که در شکل (۷-۳ ب) نشان داده شده است، می‌باشد. با توجه به رابطه (۱-۳) مقادیر زیر برای شبکه نشان داده شده در شکل (۷-۳) محاسبه می‌شوند:

$$N_v + N_f - N_e = 1 + N_{sn} = 92 + 57 - 146 = 1 + 2 = 3$$

با از بین بردن دو شبکه فرعی کوچک مستقل، فقط یک شبکه اصلی شامل ۱۳۰ ناپیوستگی و ۵۳ سطح و ۷۸ رأس باقی می‌ماند (شکل ۷-۳ ج). بنابراین مقادیر جدید برای رابطه (۱-۳) به صورت زیر است:

$$N_{sn} = N_v + N_f - N_e = 1 + 78 + 53 - 130 = 1 + 0 = 1$$



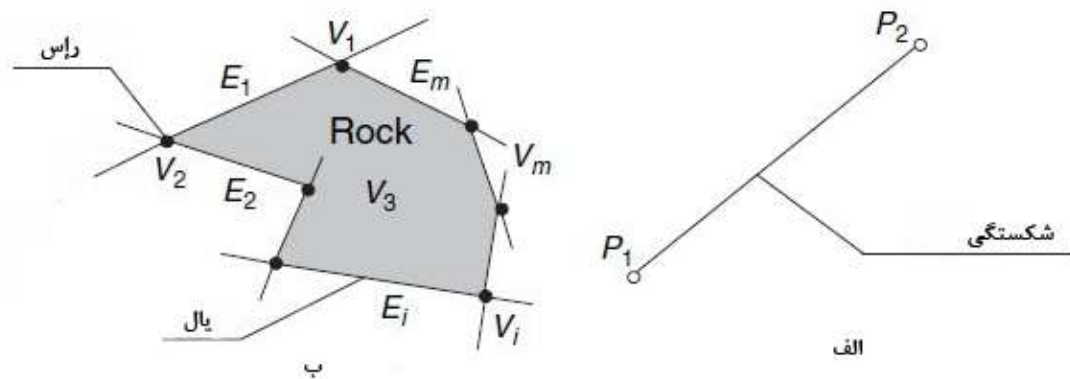
شکل ۳-۷: مرتب‌سازی شبکه ناپیوستگی‌ها (الف) شبکه ناپیوستگی اصلی (ب) مرتب‌سازی شبکه ناپیوستگی برای یافتن بلوک (ج) مرتب‌سازی شبکه ناپیوستگی بعد از حذف زیر شبکه‌های غیر مرتبط با شبکه اصلی [۱۴]

پس از مرتب‌سازی ناپیوستگی‌ها یعنی حذف ناپیوستگی‌های منفرد باید مساحت بلوک‌های سیستم دو بعدی محاسبه شود. تحلیل ساختار یک سیستم ناپیوستگی - بلوک در دو بعد معمولاً آسان است در این روش از یافتن بلوک‌ها در شبکه ناپیوستگی‌ها از عملگرهای مرزی دو بعدی برای یافتن بلوک‌ها یا چند ضلعی‌ها استفاده می‌شود. فرضیات زیر برای آسان‌تر کردن مسأله اتخاذ شده‌اند:

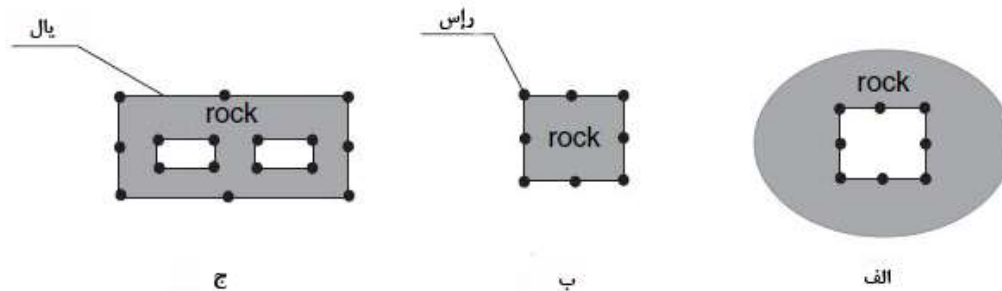
- تمامی ناپیوستگی‌ها بدون انحنا هستند و یال‌ها دارای دو رأس انتهایی با طول محدود هستند. (شکل ۳-۸ الف)

- بلوک‌ها به وسیله چند ضلعی‌هایی که به وسیله پاره خط‌ها تشکیل شده‌اند، شکل یافته‌اند. (شکل ۳-۸ ب) و ناپیوستگی‌های منحنی شکل را می‌توان به زیر ناپیوستگی‌های مستقیم تقسیم کرد.

- در این روش برای هر ناپیوستگی یا خط جهتی در نظر گرفته می‌شود با توجه به جهت ناپیوستگی‌های تشکیل دهنده یک بلوک، این بلوک می‌تواند گود، برجسته و یا ترکیبی باشد (شکل ۳-۹). مرزهای موهومی نیز به عنوان یک ناپیوستگی مصنوعی عمل می‌کنند.



شکل ۳-۸: (الف) ناپیوستگی‌های مستقیم با دو رأس (P_1 و P_2)، (ب) یک بلوک با رئوس و یال‌های آن [۱۴]



شکل ۳-۹: بلوک‌های دو بعدی (الف) بلوک‌های گود (حفره‌ها) (ب) بلوک‌های برجسته (ج) بلوک‌های ترکیبی [۱۴]

۳-۲-۱ محاسبه تقاطع ناپیوستگی‌ها

برای تعیین بلوک‌ها ابتدا باید محل برخورد ناپیوستگی‌ها یا خطوط مشخص شود. فرض می‌شود d_i و d_j دو ناپیوستگی به شعاع r_i و r_j باشند (نصف طول ناپیوستگی) و دارای مختصات نقطه مرکزی $C_i(X_i^c, Y_i^c)$ و $C_j(X_j^c, Y_j^c)$ در یک سیستم مختصات $XY-O$ باشند. بردارهای نرمال این دو ناپیوستگی به ترتیب زیر است:

$$n_i(n_x^i, n_y^i) = (\cos\theta_i, \sin\theta_i), \quad n_j(n_x^j, n_y^j) = (\cos\theta_j, \sin\theta_j) \quad (۲-۳)$$

که در آن n_i و n_j بردار نرمال ناپیوستگی‌ها و θ_i و θ_j زاویه ناپیوستگی‌ها با افق هستند. فاصله بین مرکز C_i و مرکز C_j به وسیله رابطه (۳-۳) بدست می‌آید:

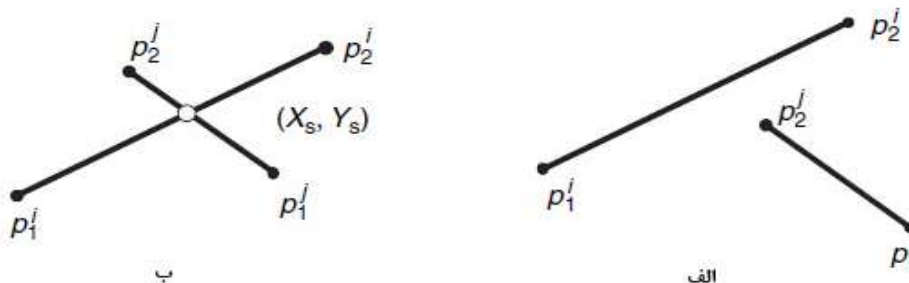
$$d_{ij}^c = \sqrt{(x_i^c - x_j^c)^2 + (y_i^c - y_j^c)^2} \quad (۳-۳)$$

که در آن x_i^c و x_j^c مختصات طول مرکز ناپیوستگی‌ها و y_i^c و y_j^c مختصات عرض مرکز ناپیوستگی‌ها هستند و d_{ij}^c فاصله بین دو مرکز ناپیوستگی است. اگر یکی از دو شرط زیر برقرار شود دو ناپیوستگی هیچ تقاطعی با هم ندارند.

$$d_{ij}^c > r_i + r_j \quad (۴-۳)$$

$$(n_i) \cdot (n_j) = n_x^i n_x^j + n_y^i n_y^j \quad (۵-۳)$$

اگر شرایط فوق برقرار باشد دو ناپیوستگی یا از هم خیلی دور و یا با همدیگر موازی هستند. در غیر این صورت دو ناپیوستگی ممکن است با توجه به موقعیت نسبی‌شان یک تقاطع داشته باشند. اگر نقاط $P_1^i(x_1^i, y_1^i)$ و $P_2^i(x_2^i, y_2^i)$ شروع و پایان ناپیوستگی i ام و نقاط $P_1^j(x_1^j, y_1^j)$ و $P_2^j(x_2^j, y_2^j)$ شروع و پایان ناپیوستگی j ام باشند (شکل ۱۰-۳ الف) نقطه $v(x_s, y_s)$ نقطه تقاطع دو پاره خط خواهد بود (شکل ۱۰-۳ ب).



شکل ۱۰-۳: الف) دو ناپیوستگی که تقاطعی ندارند. ب) دو ناپیوستگی که یک نقطه تقاطع دارند [۱۴]

رابطه پارامتری تقاطع پاره خط i ام از رابطه زیر بدست می‌آید.

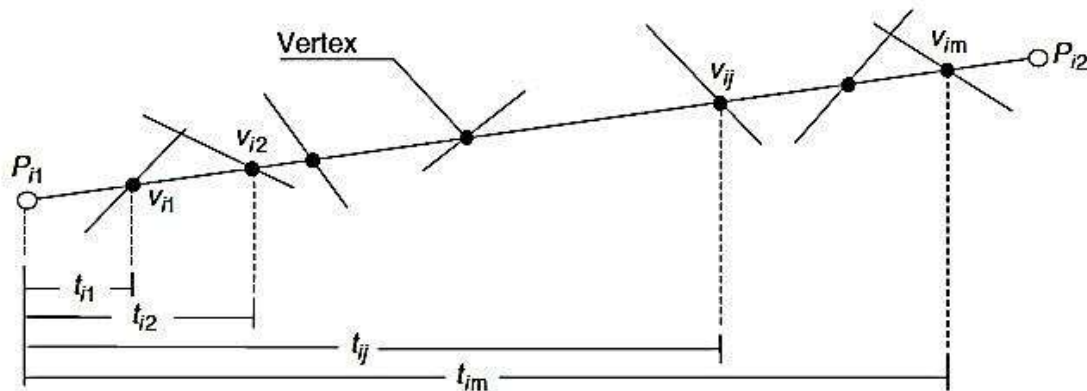
$$\begin{cases} x = x_1^i + (x_2^i - x_1^i)t_i \\ y = y_1^i + (y_2^i - y_1^i)t_i \end{cases} \quad (۶-۳)$$

که در آن $0 \leq t_i \leq 1$ است. با توجه به رابطه (۶-۳) برای محاسبه نقطه برخورد از روابط زیر استفاده می‌شود:

$$\begin{cases} x_s = \frac{1}{\Delta} [(x_1^j y_2^j - x_2^j y_1^j)(x_2^i - x_1^i) - (x_1^i y_2^i - x_2^i y_1^i)(x_2^j - x_1^j)] \\ y_s = \frac{1}{\Delta} [(x_1^i y_2^i - x_2^i y_1^i)(y_2^j - y_1^j) - (x_1^j y_2^j - x_2^j y_1^j)(y_2^i - y_1^i)] \end{cases} \quad (۷-۳)$$

$$\Delta = \begin{vmatrix} y_1^i - y_2^i & x_2^i - x_1^i \\ y_1^j - y_2^j & x_2^j - x_1^j \end{vmatrix}, \quad t_i = \frac{x_s - x_1^i}{x_2^i - x_1^i}, \quad t_j = \frac{x_s - x_1^j}{x_2^j - x_1^j}$$

که در آن x و y مختصات نقاط ابتدا و انتها هستند. اگر شرایط $0 \leq t_i \leq 1$ ، $0 \leq t_j \leq 1$ و $\Delta \neq 0$ همزمان برقرار باشند، دو پاره خط دارای یک تقاطع با مختصات (x_s, y_s) هستند که یک رأس جدید از بلوک‌های شبکه را مشخص می‌کند. در غیر این صورت هیچ تقاطعی وجود نخواهد داشت. پاره‌خط بین دو رأس مجاور که اضلاع بلوک‌ها هستند را یال می‌نامند. گام بعدی یافتن تمامی سطوح مشخص شده توسط حلقه یا چرخه یال‌ها اعم از ناپیوستگی‌ها و مرزها است. جفت رأس‌های مجاور $(v_{i,j}, v_{i,j+1})$ یال‌های ناپیوستگی i ام را تشکیل می‌دهند. با استفاده از پارمتر t و رابطه (۷-۳) می‌توان این یال‌ها را مشخص کرد (شکل ۳-۱۱).



شکل ۳-۱۱: رئوس مشخص شده در طول یک ناپیوستگی [۱۴]

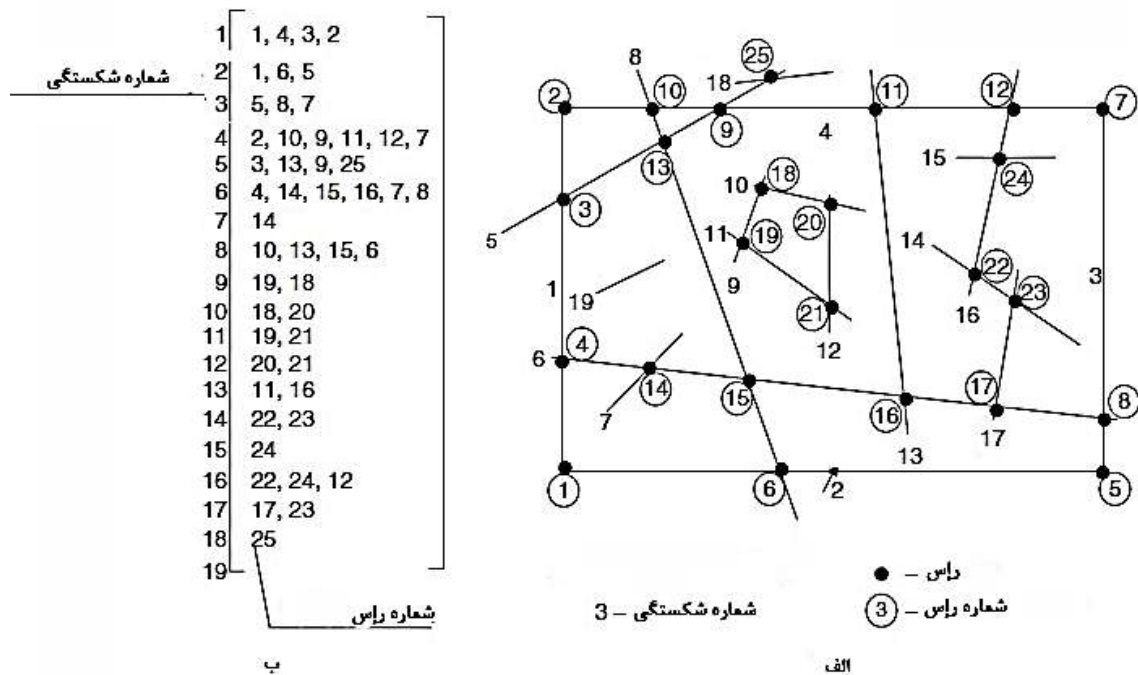
سه دسته اطلاعات باید در انتهای عملیات تعیین تقاطع‌ها مشخص شود که عبارتند از: مختصات رئوس $(V(X))$ ، یال‌های تعریف شده با جفت رئوس یا $(E(V))$ و اتصال ناپیوستگی‌ها و یال‌ها $(F(V))$. این مشخصات از شبکه ناپیوستگی به صورت زیر تعریف می‌شوند:

$$V(X) = \{(x_i, y_i), i = 1, 2, \dots, N_v\}$$

$$E(V) = \{(v_i^s, v_i^e), i = 1, 2, \dots, N_e\}$$

$$F(V) = \{(v_{i,1}, v_{i,2}, \dots, v_{i,M}), i = 1, 2, \dots, N_d\}$$

که N_v تعداد کل رئوس، N_e تعداد کل یال‌ها و N_d تعداد کل ناپیوستگی‌ها هستند. مطابق شکل (۳-۱۲) یک مثال با شبکه‌ای دارای ۱۹ ناپیوستگی و ۲۵ رأس در نظر گرفته می‌شود. همانطور که در شکل دیده می‌شود ابتدا نقاط برخورد هر ناپیوستگی محاسبه می‌شوند و پس از آن یال‌های تشکیل شده توسط این نقاط تشکیل می‌شوند. در ادامه نحوه تشخیص بلوک‌ها بر روی این مثال توضیح داده خواهد شد.



شکل ۳-۱۲: مثالی از یک شبکه ناپیوستگی (الف) شبکه ناپیوستگی با رئوس آن (ب) ماتریس ناپیوستگی‌ها و نقاط برخورد روی آن‌ها [۱۴]

۲-۲-۲-۳ مرتب‌سازی یال‌ها

پس از تشکیل یال‌ها با استفاده از نقاط برخورد بین ناپیوستگی‌ها باید یال‌های تشکیل شده مرتب شوند. منظور از مرتب‌سازی یال‌ها حذف خطوط و پاره‌خط‌هایی است که در تشکیل بلوک‌ها یا چند ضلعی‌ها مؤثر نیستند، برای مرتب‌سازی یال‌ها باید ابتدا ناپیوستگی‌هایی که دارای تقاطع نیستند و یا یک تقاطع بیشتر ندارند را حذف کرد. یک شبکه مرتب شده از رئوس و یال‌ها با اتصالات کامل نشان دهنده یک گراف است. در این روش برای هر یال گراف دو جهت در نظر گرفته می‌شود که هر یال دقیقاً توسط دو رأس (رئوس آغازی و پایانی) معین شده است و دقیقاً در دو چند ضلعی (بلوک) مجاور مشترک است که دارای دو جهت مخالف هستند. به یک گراف که یال‌های آن دارای جهت باشند گراف جهت‌دار گفته می‌شود. شکل (۳-۱۳) گراف جهت‌دار متناظر با شبکه ناپیوستگی شکل (۳-۱۲) را نشان می‌دهد. در این شکل علاوه بر ماتریس رئوس $(F(V))$ ، ماتریس جهت (O) نیز وجود دارد.

ماتریس جهت برای نشان دادن استفاده از دو جهت یال‌های گراف است. به ازای r رأس در ماتریس رئوس، $r-1$ عضو در ماتریس جهت وجود دارد که مقادیر درایه‌های این ماتریس به صورت زیر محاسبه می‌شود:

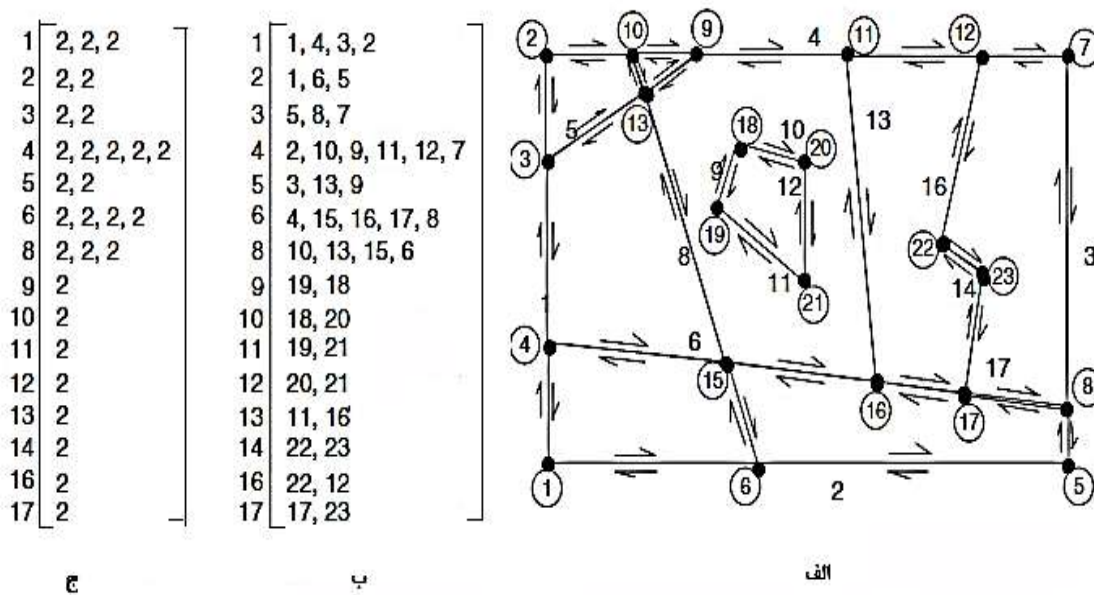
(از هر دو جهت یال $\overline{v_{ij}, v_{i,j+1}}$ استفاده نشده باشد.)
 $O_{ij} = 2$

(در یال $\overline{v_{ij}, v_{i,j+1}}$ از جهت رأس v_{ij} به v_{ij+1} استفاده نشده باشد.)
 $O_{ij} = 1$

(در یال $\overline{v_{ij}, v_{i,j+1}}$ از جهت رأس v_{ij+1} به v_{ij} استفاده نشده باشد.)
 $O_{ij} = -1$

(از هر دو جهت یال $\overline{v_{ij}, v_{i,j+1}}$ استفاده شده باشد.)
 $O_{ij} = 0$

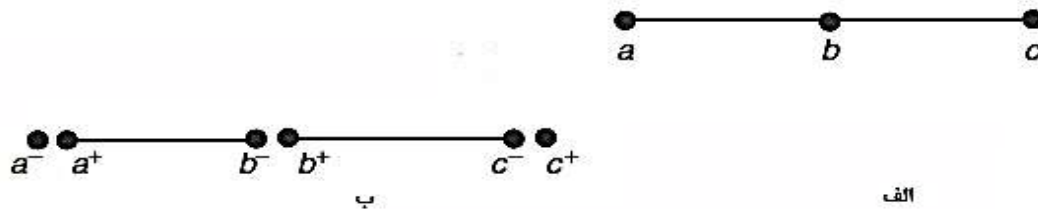
شناسایی بلوک‌ها زمانی به پایان می‌رسد که تمامی درایه‌های ماتریس جهت برابر صفر شوند.



شکل ۳-۱۳: (الف) گراف جهت‌دار از یک شبکه ناپیوستگی (ب) ماتریس رئوس ناپیوستگی‌ها (ج) ماتریس جهت [۱۴]

۳-۲-۲-۳ عملگر مرزی در سیستم دو بعدی

برای شناسایی بلوک‌ها یک عملگر مرزی به نام ∂ در شبکه ناپیوستگی‌ها تعریف می‌شود این عملگر دو بعدی یک سری یال جهت‌دار در شبکه تولید می‌کند. این عملگرها نشان دهنده حلقه‌های مرزی شبکه هستند. برای حلقه‌های بسته عملگر مرزی به صورت $\partial(\partial(k)) = 0$ است. یافتن بلوک بر اساس عملگر مرزی دسته یال‌ها صورت می‌گیرد. برای تعیین کردن عملگر مرزی، از مفهوم تجزیه شدن و رئوس علامت‌دار استفاده می‌شود. اگر v یک رأس از یالی باشد v^- شروع یال و v^+ نقطه انتهایی یال در یک حلقه با جهت مثبت (جهت عقربه‌های ساعت) هستند. شکل (۳-۱۴) نحوه اعمال عملگر مرزی بر دو یال را به صورت شماتیک نشان می‌دهد.



شکل ۳-۱۴: (الف) زنجیره اصلی رئوس (ب) رئوس تجزیه و علامت‌دار شده [۱۴]

(v^-, v^+) به یک رأس v بر می‌گردند اما به دو یال مجاور متصل به رأس v تعلق دارند. برای یک جفت تجزیه کننده رئوس علامت‌دار (v^-, v^+) رابطه (۳-۸) برقرار است.

$$v^+ + v^- = 0 \quad (۳-۸)$$

این رابطه این امکان را می‌دهد که شروع و پایان یک رأس از یال را محاسبه کرد. نتیجه عملگر مرزی بر روی یک یال تنها به صورت رابطه زیر می‌باشد:

$$\partial(v_i, v_j) = (v_i^+) + (v_j^-) \quad (9-3)$$

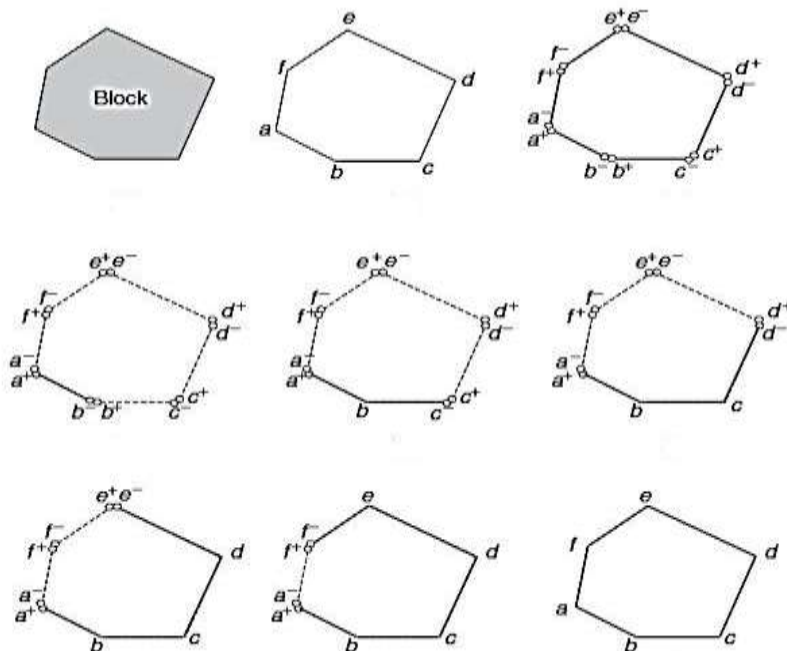
برای یک سری یال متصل به هم، نتیجه عملگر مرزی به صورت زیر است:

$$\partial(\Sigma(v_i, v_j)) = \Sigma \partial(v_i, v_j) = \Sigma[(v_i^+) + (v_j^-)] \quad (10-3)$$

برای یک حلقه بسته از یال‌ها که بلوک را معین می‌کنند، نتیجه عملگر مرزی بر روی یال‌ها برابر صفر می‌شود.

$$\partial(\Sigma(v_i, v_{i+1})) = 0 \quad (11-3)$$

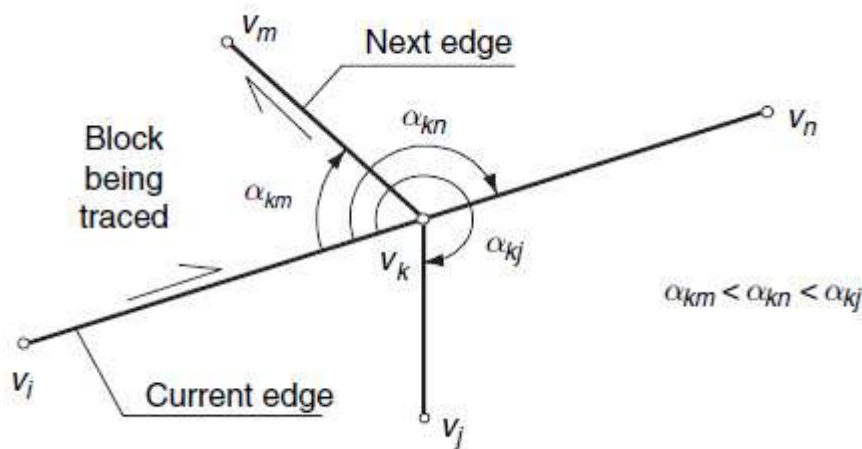
رابطه (۱۱-۳) یک عملگر مرزی دو بعدی است که برای مشخص کردن تکمیل فرایند یافتن بلوک استفاده می‌شود. شکل (۱۵-۳) روند اعمال عملگر مرزی را بر روی یک سری یال که تشکیل دهنده یک بلوک هستند، نشان می‌دهد.



شکل ۱۵-۳: نمونه‌ای از اجرای عملیات عملگر مرزی بر روی یک بلوک با شش یال [۱۴]

۴-۲-۲-۳ قانون زاویه دست چپی

یافتن بلوک با هر رأس و هر یال متصل به آن آغاز می‌شود. تمامی یال‌های متصل به این رأس با استفاده از ماتریس ناپیوستگی-یال $F(E)$ مشخص می‌شوند. برای یافتن بلوک‌ها باید چرخه‌ای بسته از یال‌ها را انتخاب کرد. برای این منظور از یک یال مشخص عملیات تعیین چرخه‌ای از یال‌ها شروع می‌شود. تعدادی از یال‌های دیگر به این یال متصل هستند که برای تعیین یال بعدی یک چرخه از قانون زاویه دست چپی استفاده می‌شود. در این قانون یال بعدی انتخاب شده برای یک چرخه باید کمترین زاویه از سمت چپ با یال مبنا را داشته باشد. در شکل (۳-۱۶) این نحوه اعمال این قانون نشان داده شده است. برای یافتن بلوک‌ها یک آرایه کمکی نیز برای ثبت تعداد مرتبه‌ایی که هر یال مورد استفاده قرار می‌گیرد نیاز است.

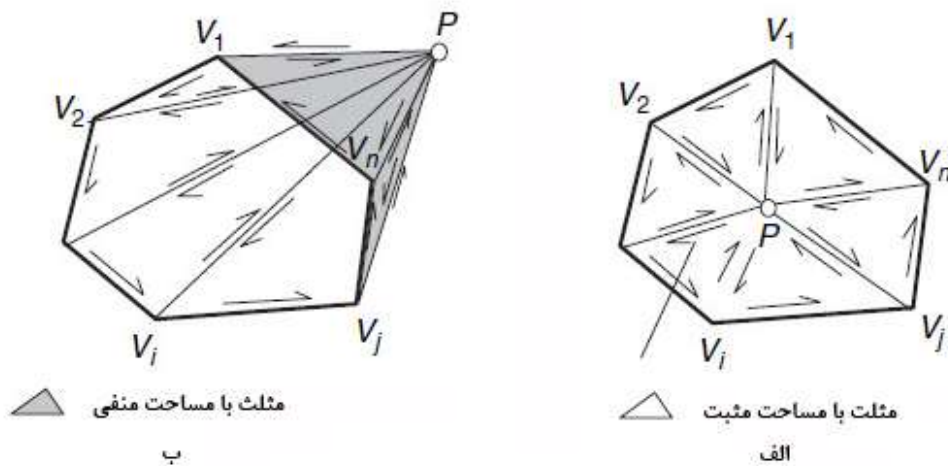


شکل ۳-۱۶: قانون زاویه دست چپی [۱۴]

۵-۲-۲-۳ تشخیص بلوک‌ها

الگوریتم یافتن بلوک در این روش بر اساس در نظر گرفتن دو جهت برای هر ناپیوستگی می‌باشد. بنابراین تمامی انواع بلوک‌های مقعر (گود)، محدب (برجسته) و ترکیبی در این روش تعیین

می‌شوند. همانطور که در شکل (۳-۱۷) مشاهده می‌گردد برای محاسبه مساحت یک چند وجهی از یک نقطه کمکی مانند P استفاده می‌شود. با استفاده از نقطه P و سایر رئوس چند ضلعی، مثلث‌هایی تشکیل داده می‌شود. در نتیجه محاسبه مساحت همه مثلث‌های تشکیل شده، مساحت چند ضلعی محاسبه می‌شود. فرض می‌شود اگر اضلاع چند ضلعی در جهت عقربه‌های ساعت مرتب شده باشند این چند ضلعی دارای مساحت منفی و اگر اضلاع آن در خلاف جهت عقربه‌های ساعت مرتب شده باشند دارای مساحت مثبت است. به بلوک‌های دارای مساحت منفی بلوک‌های مقعر و به بلوک‌های دارای مساحت مثبت بلوک‌های محدب گفته می‌شود. اگر نقطه P درون بلوک محدب با n رأس که در خلاف جهت عقربه‌های ساعت مرتب شده‌اند، باشد، تمامی مثلث‌های حاصل از نقطه P و جفت رأس‌های چند ضلعی آن دارای مساحت مثبت هستند یا به عبارت دیگر این بلوک یک بلوک محدب است و اگر این نقطه خارج از بلوک باشد تعدادی از مثلث‌ها دارای مساحت منفی می‌شوند ولی برابری مساحت همه مثلث‌ها مثبت می‌شود.



شکل ۳-۱۷: (الف) بلوکی با مثلث‌هایی با مساحت مثبت (ب) بلوکی با مثلث‌هایی با مساحت مثبت و منفی (هاشور خورده) [۱۴]

مساحت مثلث‌های تشکیل شده از رئوس چند ضلعی و نقطه P از رابطه زیر بدست می‌آید:

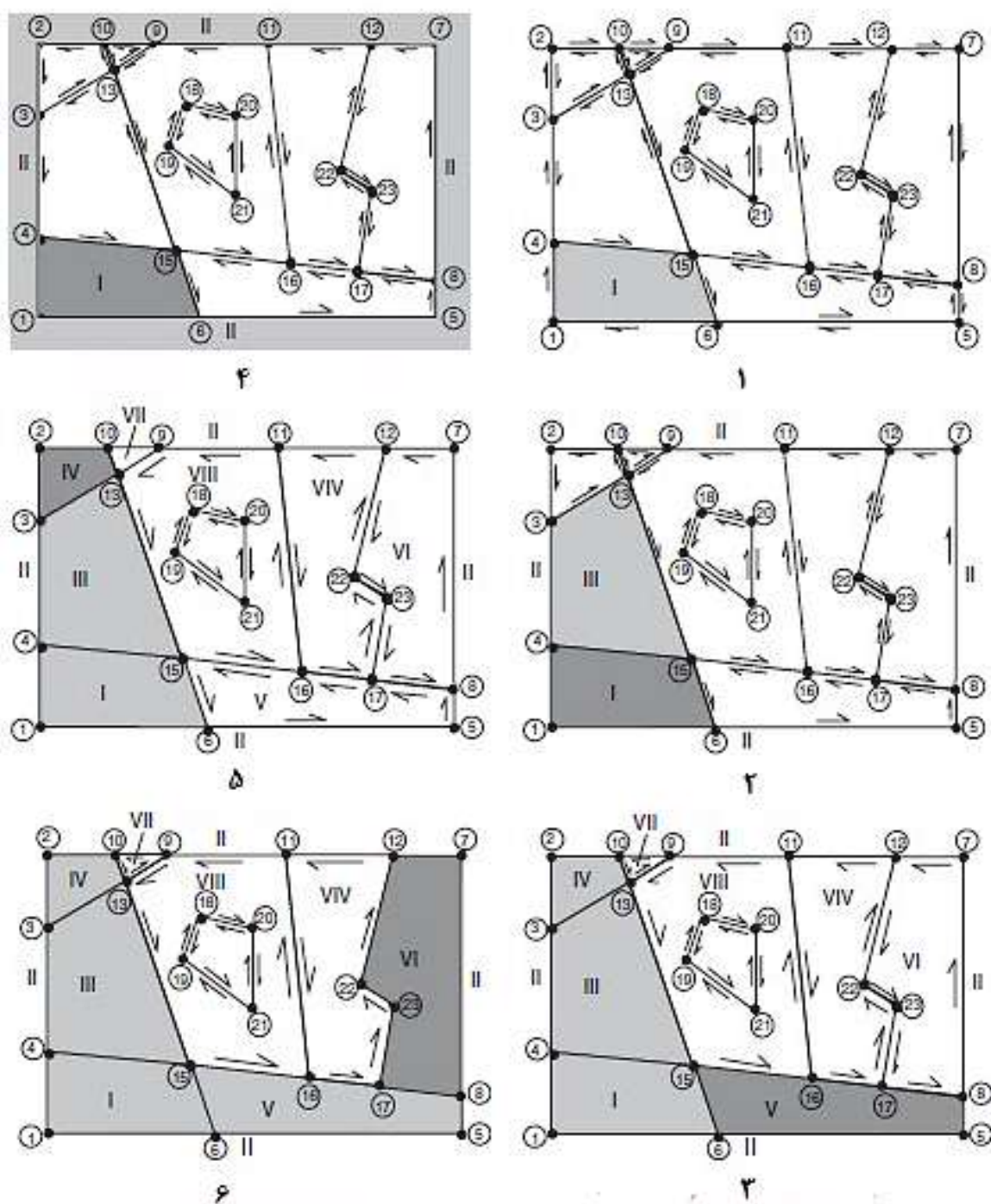
$$A^i = \frac{1}{2} \begin{vmatrix} 1 & x_p & y_p \\ 1 & x_i & y_i \\ 1 & x_j & y_j \end{vmatrix} \quad (۱۲-۳)$$

که در این رابطه x_p و y_p مختصات نقطه P و x_i و y_i مختصات نقاط رئوس بلوک‌ها هستند.

نقطه $p(x_p, y_p)$ را نقطه مرجع گویند. برای تشخیص بلوک‌های ترکیبی، اگر نقطه مرجع انتخاب شده برای دو بلوک مثبت و منفی درون یک بلوک قرار گیرد، آن بلوک ترکیبی است. برای مثال نشان داده شده در شکل (۱۲-۳) پروسه یافتن بلوک‌ها در شکل (۱۸-۳) نشان داده شده است. در این روش برای هر یال دو جهت در نظر گرفته می‌شود و از هر جهت یال به عنوان یالی جداگانه برای یافتن بلوک‌ها استفاده می‌شود. به عنوان مثال همانطور که در شکل دیده می‌شود ابتدا از یال متصل به رئوس ۱ و ۶ و جهت راست یا در جهت عقربه ساعت در این یال شروع کرده و طبق قانون زاویه دست چپی یال متصل به رئوس ۶ و ۱۵ و همان جهت به عنوان یال بعدی بلوک انتخاب می‌شود و به همین ترتیب بلوک ۱ و سایر بلوک‌ها مشخص می‌شوند. پروسه یافتن بلوک‌ها زمانی به پایان می‌رسد که از تمام یال‌ها و دو جهت آن‌ها استفاده شده باشد.

۳-۲-۶ اتصال و آنالیزهای مکانیکی در بلوک‌ها

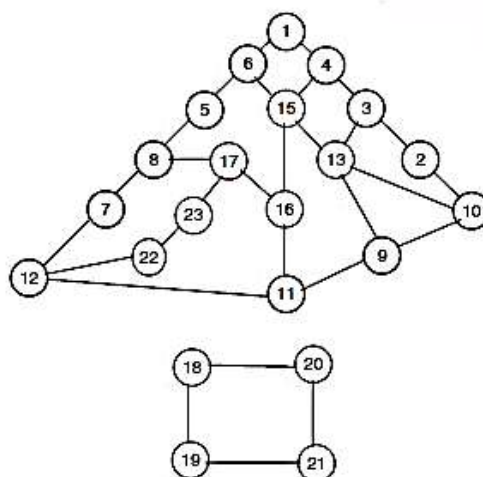
برای تکمیل کردن فرایند یافتن بلوک‌ها دو عمل دیگر نیز باید صورت گیرد. آنالیز مسیرهای جریان بین بلوک‌ها و آنالیز تنش، کرنش، و تغییر شکل‌پذیری بلوک‌ها یا به عبارت دیگر آنالیز هیدرومکانیکی بلوک‌ها باید انجام شود. اتصال بین بلوک‌ها برای آنالیز جریان بین بلوک‌ها و آنالیز مکانیکی بلوک‌ها کاربرد دارد. برای نشان دادن اتصال بین رئوس از ماتریس $c_v = [c_{ij}^v]$ و برای نشان دادن اتصال بین بلوک‌ها از ماتریس $c_b = [c_{ij}^b]$ که به صورت زیر تعیین می‌شوند، استفاده می‌شود این ماتریس‌ها در شکل‌های (۱۹-۳) و (۲۰-۳) نشان داده شده‌اند.



شکل ۳-۱۸: پروسه یافتن بلوک‌ها در یک شبکه ناپیوستگی با استفاده از قانون زاویه دست چپی [۱۴]

$$c_{ij}^v = \begin{cases} 0 \\ 1 \end{cases} \quad (\text{اگر رأس } i \text{ و } j \text{ اتصال داشته باشند } 1 \text{ در غیر این صورت } 0) , c_{ij}^v = c_{ji}^v$$

$$c_{ij}^b = \begin{cases} 0 \\ 1 \end{cases}, c_{ij}^b = c_{ji}^b \text{ (اگر بلوک } i \text{ و } j \text{ اتصال داشته باشند ادر غیر این صورت ۰)}$$

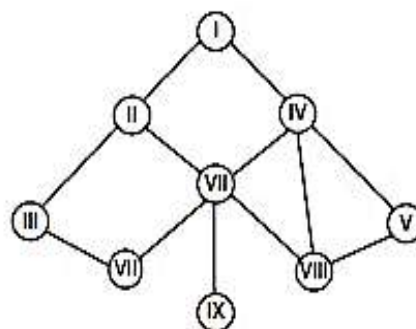
[illegible]

عراق اتصال رؤوس

১৮

I	1, 1, 0, 1, 0, 0, 0, 0, 0
II	1, 1, 0, 0, 0, 1, 0, 0
III	1, 0, 0, 1, 0, 0, 0, 0
IV	1, 1, 0, 1, 1, 0
V	1, 0, 0, 1, 0
VI	1, 1, 0, 0
VII	1, 1, 1
VIII	1, 0
IX	1

ماتریس اتصال

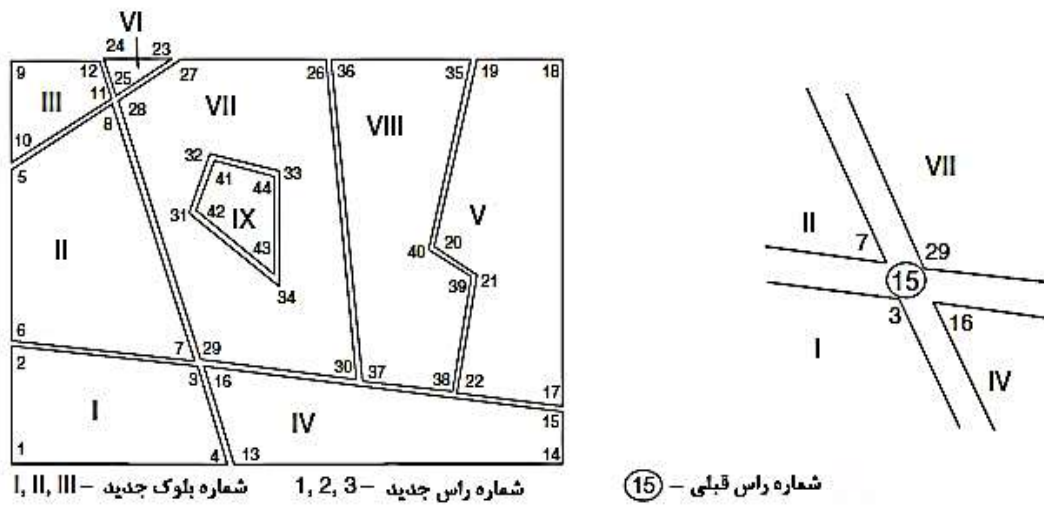


گراف اتصال

شکل ۳-۲۰: گراف و ماتریس اتصال بین بلوک‌ها [۱۴]

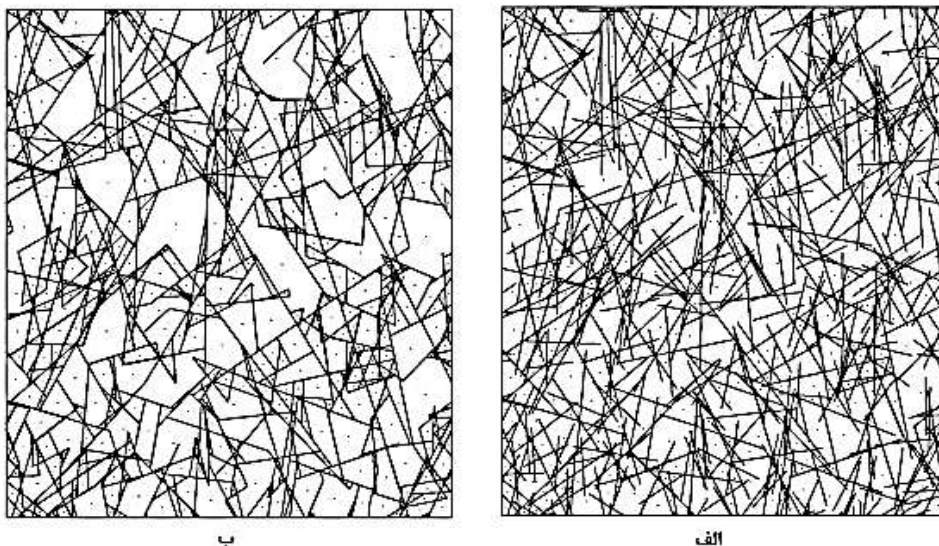
برای آنالیزهای مکانیکی بلوک‌ها نمی‌توان به طور مستقیم از فرایند یافتن بلوک‌ها که ذکر شد استفاده کرد، چون در فرایند قبل با توجه به نام گذاری رئوس و یال‌های بلوک‌ها، از لحاظ مکانیکی بلوک‌های مجاور در یکدیگر قفل شده‌اند و نیاز است تا دوباره یال‌ها و رئوس نام گذاری شوند. نام گذاری مجدد رئوس و یال‌ها روش‌های مختلفی دارد که یکی از این روش‌ها افزودن یک رأس در تمامی نقاطی که رئوس قرار دارند است. شکل (۳-۲۱) نمونه‌ای از نام گذاری مجدد رئوس برای شبکه مثال مورد بحث را نشان می‌دهد.

الگوریتم یافتن بلوک‌ها به روش تکنیک‌های عددی تنها الگوریتم مناسب برای مسائل دو بعدی نیست؛ الگوریتم‌های زیادی بر اساس قانون کوچکترین زاویه چرخش به چپ برای یافتن بلوک‌ها وجود دارد، به هر حال استفاده از الگوریتم روش مرزی یک جواب قطعی بدون نیاز به بررسی خطا را ارائه می‌دهد. شکل (۳-۲۲) مثالی از یک شبکه ناپیوستگی و یافتن بلوک‌های آن با استفاده از کد UDEC را نشان می‌دهد.



شکل ۳-۲۱: نام گذاری مجدد رئوس و یال‌ها [۱۴]

الگوریتم استفاده شده در این روش یک الگوریتم ریاضی با جواب قطعی می‌باشد و آنالیز هیدرولیکی و مکانیکی بلوک‌ها را نیز مورد بررسی قرار می‌دهد. اما استفاده از عملگر مرزی و جهت‌دار کردن ناپیوستگی‌ها باعث طولانی شدن این الگوریتم شده است. از طرفی این الگوریتم یک الگوریتم پیچیده نیز می‌باشد و پیاده‌سازی آن دشوار است.



شکل ۳-۲۲: الف) مثالی از یک شبکه ناپیوستگی واقعی ب) یافتن بلوک‌ها با استفاده از کد UDEC [۱۴]

۳-۲-۳ الگوریتم شی مبتنی بر گراف‌های جهت‌دار

شی در قسمتی از رساله دکتری خود الگوریتمی بر مبنای استفاده از گراف‌های جهت‌دار برای شناسایی بلوک‌ها ارائه داده است. این الگوریتم در ادامه توضیح داده می‌شود [۹]:

در اثر برخورد ناپیوستگی‌های مختلف به یکدیگر در توده‌سنگ بلوک‌هایی با تعداد رئوس و ابعاد مختلف ایجاد می‌شوند. این ناپیوستگی‌ها با طول‌ها و شیب‌های متفاوت در هر کجای توده‌سنگ ممکن است حضور داشته باشند. برخی از الگوریتم‌های موجود ناپیوستگی‌ها را به صورت صاف و مستقیم فرض می‌کنند. این فرض چندان از واقعیت دور نیست؛ در حالی که می‌توان ناپیوستگی‌های خمیده را به ناپیوستگی‌های کوچکتر و مستقیم تبدیل کرد. الگوریتم شناسایی بلوک‌ها باید به گونه‌ای باشد که در هر ترکیبی از ناپیوستگی‌ها تمام بلوک‌های موجود در توده‌سنگ را شناسایی کند. الگوریتمی که توسط شی ارائه شده قادر است ناپیوستگی‌های منحنی شکل را نیز در شناسایی بلوک‌ها در نظر بگیرد. در این الگوریتم بلوک‌هایی با اضلاع منحنی شکل نیز ایجاد می‌شوند.

مراحل این الگوریتم به صورت زیر می‌باشد:

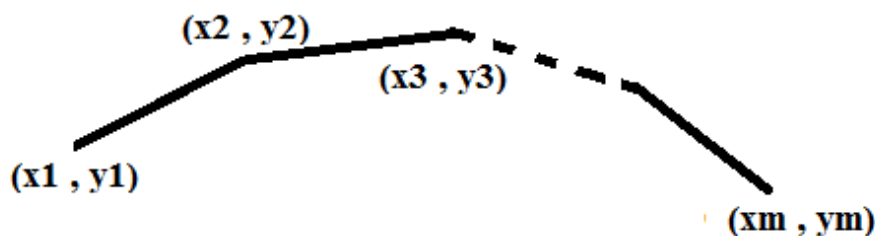
- ۱- محاسبه نقاط برخورد ناپیوستگی‌ها
- ۲- اتصال بین نقاط برخورد
- ۳- برش درختی
- ۴- شناسایی همه بلوک‌ها
- ۵- شناسایی بلوک‌های واقعی

۳-۲-۱ محاسبه نقاط برخورد

مرحله اول در شناسایی بلوک‌ها در این الگوریتم تعیین نقاط برخورد بین ناپیوستگی‌ها است. ناپیوستگی‌ها در دو بعد به صورت صاف و مستقیم یا منحنی شکل هستند. ناپیوستگی‌های منحنی شکل را می‌توان به صورت خطوط خمیده در نظر گرفت. خطوط خمیده نیز به صورت تعدادی پاره‌خط مستقیم فرض می‌شوند. بنابراین یک ناپیوستگی منحنی شکل به صورت تعدادی نقطه ابتدا و انتهای پاره‌خط به صورت زیر می‌باشد:

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_m, y_m)$$

همانطور که در شکل (۳-۲۳) مشاهده می‌شود نقاط مرزی پاره‌خط‌های تشکیل دهنده ناپیوستگی خمیده هستند.



شکل ۳-۲۳: تبدیل یک منحنی به تعدادی پاره‌خط مستقیم [۹]

تابع طول $l(x, y)$ در راستای ناپیوستگی خمیده از نقطه (x_1, y_1) تا نقطه (x, y) تعریف می‌شود. برای هر نقطه گوشه، فاصله نقطه (x_1, y_1) تا نقطه (x_i, y_i) در راستای ناپیوستگی منحنی شکل به صورت رابطه (۳-۱۲) می‌باشد:

$$l_i = \sum_{j=1}^{i-1} \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2} \quad (۳-۱۲)$$

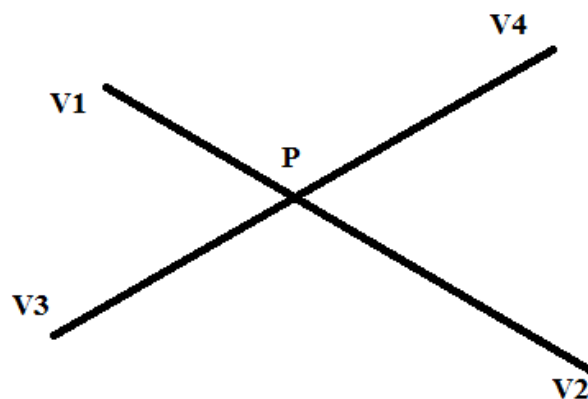
و با توجه به رابطه بالا تابع طول برای یک ناپیوستگی خمیده به صورت رابطه زیر تعریف می‌شود:

$$l(x, y) = l_i + \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (۱۳-۳)$$

با دادن یک مقدار از تابع $l(x, y)$ هر نقطه از ناپیوستگی منحنی شکل قابل محاسبه خواهد بود. با توجه به این که ناپیوستگی‌های منحنی شکل به صورت تعدادی پاره‌خط مستقیم فرض می‌شوند، برای محاسبه نقاط برخورد دو ناپیوستگی کافی است همانند شکل (۳-۲۴) نقاط برخورد بین دو پاره‌خط V_1V_2 و V_3V_4 را تعیین کرد. معادله پاره‌خط V_1V_2 به صورت رابطه زیر می‌باشد:

$$x = x_1 + (x_2 - x_1)t, \quad 0 \leq t \leq 1 \quad (۱۴-۳)$$

$$y = y_1 + (y_2 - y_1)t, \quad 0 \leq t \leq 1$$



شکل ۳-۲۴: برخورد دو پاره‌خط در صفحه [۹]

معادله پاره‌خط V_3V_4 نیز به صورت رابطه زیر نوشته می‌شود:

$$x = x_3 + (x_4 - x_3)T, \quad 0 \leq T \leq 1 \quad (۱۵-۳)$$

$$y = y_3 + (y_4 - y_3)T, \quad 0 \leq T \leq 1$$

که در این دو رابطه اخیر x و y مختصات نقاط ابتدا و انتهای دو پاره‌خط هستند. برای محاسبه نقطه برخورد این دو پاره‌خط از رابطه زیر استفاده می‌شود. در این رابطه برای محاسبه نقطه برخورد دو پاره خط V_1V_2 و V_3V_4 طول و عرض این دو پاره‌خط همانند رابطه (۳-۱۶) مساوی با یکدیگر قرار داده می‌شوند.

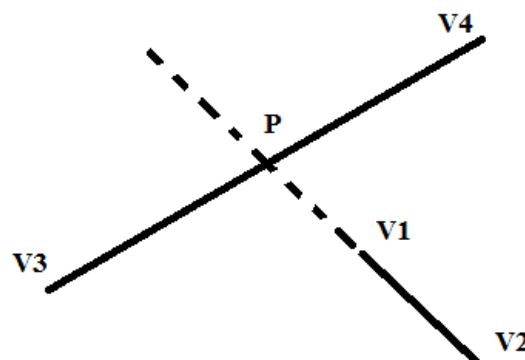
$$x_1 + (x_2 - x_1)t = x_3 + (x_4 - x_3)T \quad (۳-۱۶)$$

$$y_1 + (y_2 - y_1)t = y_3 + (y_4 - y_3)T$$

این رابطه یک رابطه خطی با دو مجهول t و T است که می‌توان آن را به صورت رابطه (۳-۱۷) بازنویسی کرد.

$$\begin{pmatrix} x_2 - x_1 & x_3 - x_4 \\ y_2 - y_1 & y_3 - y_4 \end{pmatrix} \begin{pmatrix} t \\ T \end{pmatrix} = \begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \end{pmatrix} \quad (۳-۱۷)$$

اگر جواب معادله فوق وجود داشته باشد و دو شرط $0 \leq t \leq 1$ و $0 \leq T \leq 1$ نیز همزمان برقرار باشند دو پاره‌خط دارای یک نقطه تقاطع هستند. که از طریق رابطه (۳-۱۷) قابل محاسبه است. اگر این رابطه دارای جواب باشد ولی شرایط فوق برقرار نباشند دو پاره‌خط همانند شکل (۳-۲۵) دارای یک نقطه تقاطع مجازی هستند.



شکل ۳-۲۵: دو پاره‌خط با یک نقطه برخورد مجازی [۹]

اگر دترمینان رابطه (۳-۱۷) برابر صفر شود دو پاره خط V_1V_2 و V_3V_4 با هم موازی یا بر یکدیگر منطبق می‌باشند. این دترمینان به صورت رابطه زیر است:

$$\begin{vmatrix} x_2 - x_1 & x_3 - x_4 \\ y_2 - y_1 & y_3 - y_4 \end{vmatrix} = 0 \quad (3-18)$$

با حل معادله رابطه (۳-۱۸) نقطه تقاطع دو ناپیوستگی محاسبه می‌شود.

۳-۲-۲-۳ اتصال بین نقاط برخورد

در شبکه ناپیوستگی‌ها نقاط برخورد به عنوان رئوس بلوک و پاره خط بین آن‌ها به عنوان اضلاع بلوک تعیین می‌شوند. نقاط برخورد هر ناپیوستگی با دیگر ناپیوستگی‌ها محاسبه شده و در سطر یک ماتریس ذخیره می‌شود.

ماتریس Q برای ذخیره نقاط برخورد هر شکستگی به صورت زیر تعریف می‌شود:

$$Q = [Q_{ij}]$$

که در هر سطر از ماتریس Q نقاط برخورد بر روی هر ناپیوستگی ثبت می‌شود. نقاط برخورد بر روی هر ناپیوستگی بر اساس مختصات طول نقطه مرتب می‌شوند. اگر مختصات نقاط برخورد برابر یا بسیار نزدیک به هم باشند این نقاط به عنوان یک نقطه در نظر گرفته می‌شوند. در این حالت دو یا چند ناپیوستگی در یک نقطه با ناپیوستگی دیگری برخورد می‌کنند.

برای شناسایی بلوک‌های تشکیل شده نیاز به زاویه ناپیوستگی‌ها در نقاط تقاطع با هم است. برای محاسبه این زاویه‌ها از پاره‌خط‌های تعریف شده بر هر ناپیوستگی منحنی شکل و تابع طول استفاده می‌شود.

۳-۳-۲-۳ برش درختی

در شبکه ناپیوستگی‌ها، اضلاع بلوک‌های تشکیل شده قسمتی از ناپیوستگی‌ها و نقاط برخورد رؤس بلوک‌ها را تشکیل می‌دهند. در یک شبکه ناپیوستگی قسمتی از ناپیوستگی‌هایی که در تشکیل بلوک‌ها مؤثر نیستند باید حذف شوند؛ در واقع ناپیوستگی‌هایی که اضلاع بلوک‌ها را تشکیل می‌دهند باقی می‌مانند. برای انجام عملیات برش درختی از ماتریس رؤس (Q) استفاده می‌شود. در این ماتریس هر سطری که کمتر از ۲ مقدار غیر صفر در درایه‌های آن وجود داشته باشد باید از ماتریس حذف شود. هنگامی که درایه‌ای از یک سطر ماتریس Q حذف می‌شود باید این مقدار از کل ماتریس نیز حذف شود. بنابراین برخی از سطرهای دیگر این ماتریس نیز ممکن است تعداد درایه‌های غیر صفر آن‌ها کمتر از ۲ شود. ماتریس Q در هر مرحله از حذف درایه‌های غیر صفر اصلاح می‌شود و این عمل تا زمانی ادامه می‌یابد که هیچ درایه‌ای از این ماتریس قابل حذف نباشد. به عمل حذف درایه‌ها و سطرهای ماتریس Q که در واقع همان ناپیوستگی‌های بدون اتصال هستند، برش درختی گفته می‌شود.

۴-۳-۲-۳ شناسایی بلوک‌ها

برای یافتن تمام بلوک‌های ایجاد شده در شبکه بلوکی سه دسته اطلاعات لازم و کافی است. این اطلاعات عبارتند از:

- ۱- نقاط برخورد و ترتیب آن‌ها بر روی هر ناپیوستگی
- ۲- تمام خطوطی که از نقاط برخورد عبور می‌کنند.
- ۳- جهت زاویه پاره‌خط‌هایی که از نقاط برخورد عبور می‌کنند.

برای شناسایی بلوک‌ها، شبکه بلوکی تصحیح شده در مرحله قبل به گرافی جهت‌دار تبدیل می‌شود. رئوس این گراف نقاط برخورد و یال‌های آن پاره‌خط‌های واصل بین دو رأس است. مرز یک بلوک چرخه‌ای از یال‌های موجود در گراف می‌باشد. برای شناسایی بلوک‌ها از ماتریس رئوس (Q) استفاده می‌شود. در این ماتریس هر دو رأس متوالی q_i و q_{i+1} که درایه‌های موجود در سطر i ماتریس Q هستند، تشکیل یک یال از گراف را می‌دهند. شناسایی بلوک‌ها در این روش بر مبنای استفاده از یال‌های جهت‌دار در گراف می‌باشد. برای تعریف جهت یال‌ها در گراف از ماتریس M استفاده می‌شود. این ماتریس در بر دارنده جهت یال‌های گراف است و با استفاده از ماتریس Q به صورت زیر تعریف می‌شود:

$$M = [m_{ij}]$$

این ماتریس بر اساس یال‌های به وجود آمده در هر سطر تشکیل می‌شود و درایه‌های آن به صورت زیر تعریف می‌شوند:

$m_{ij} = 00$ از هر دو یال q_i, q_{i+1} و q_i, q_{i+1} استفاده شده است.

$m_{ij} = 01$ فقط از یال q_i, q_{i+1} استفاده شده است.

$m_{ij} = 10$ فقط از یال q_{i+1}, q_i استفاده شده است.

$m_{ij} = 11$ از هیچکدام از یال‌های q_i, q_{i+1} و q_{i+1}, q_i استفاده نشده است.

شناسایی بلوک‌ها زمانی به پایان می‌رسد که از همه یال‌ها و هر دو جهت آن‌ها در گراف استفاده شده باشد به عبارت دیگر تمام درایه‌های ماتریس M برابر با صفر شوند.

۵-۳-۲-۳ شناسایی بلوک‌های واقعی

همانطور که بیان شد در این روش برای شناسایی بلوک‌ها از گراف‌های جهت‌دار استفاده می‌شود. هر گراف دارای دو جهت از ترتیب یال‌ها می‌باشد. دو جهت یال‌ها در هر گراف فضای موجود را به دو فضای داخلی و خارجی گراف تبدیل می‌کند. مرحله نهایی در شناسایی بلوک‌ها در این روش تشخیص فضای خارجی و داخلی یک گراف بسته می‌باشد. فضای داخلی هر گراف بلوکی از شبکه را تشکیل می‌دهد و فضای خارجی آن باید حذف شود. روش‌های زیر برای تشخیص فضای داخلی و خارجی یک گراف جهت‌دار در این روش پیشنهاد شده است:

۱-۵-۳-۲-۳ روش زاویه‌ایی

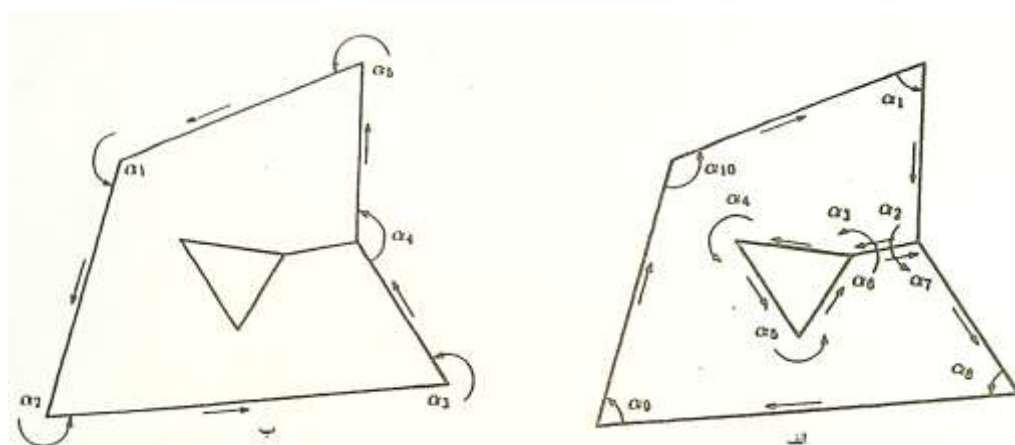
هر چرخه بسته از گراف اصلی با تعداد s یال یک بلوک با s ضلع را تشکیل می‌دهد. همانطور که در شکل (۳-۲۶ الف) مشاهده می‌شود، اگر زاویه بین ضلع‌های مجاور در گراف α_i باشد. اگر مجموع این زوایا در رابطه زیر صدق کند، یک گراف با فضای داخلی آن شناسایی می‌شود:

$$\sum_{i=1}^s \alpha_i = 180(s - 2) \quad (۱۹-۳)$$

که در آن s تعداد یال‌ها و α_i زاویه بین یال‌های مجاور می‌باشد. در این حالت چرخه یال‌های گراف فضای داخلی گراف را نشان می‌دهند و در واقع یک بلوک واقعی شناسایی شده است. اگر مجموع این زوایا در رابطه (۳-۲۰) صدق کند، یک گراف با فضای خارجی آن شناسایی می‌شود:

$$\sum_{i=1}^s \alpha_i = 180(s + 2) \quad (۲۰-۳)$$

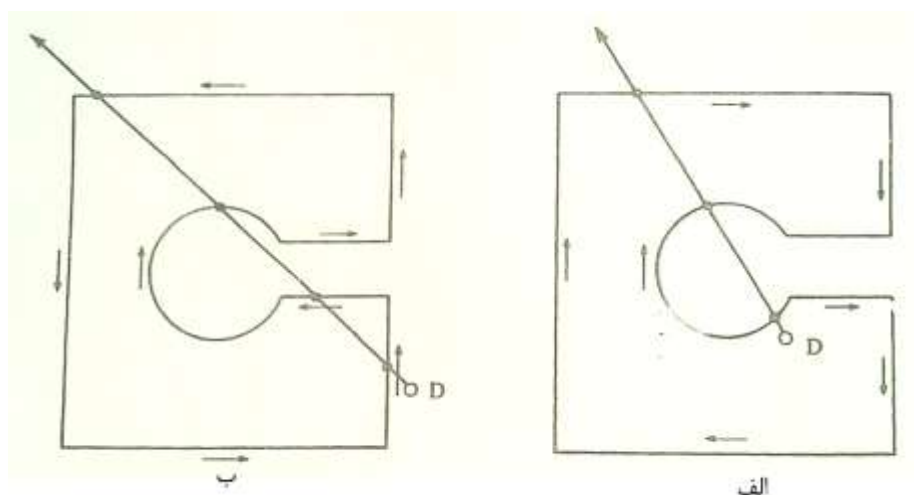
در این حالت چرخه یال‌های گراف فضای خارجی گراف را نشان می‌دهند (شکل ۳-۲۶ ب).



شکل ۳-۲۶: روش زاویه‌ای (الف) گراف با فضای داخلی (ب) گراف با فضای خارجی [۹]

۳-۲-۳-۵-۲ روش نقطه‌ای

در این روش نزدیک‌ترین نقطه در سمت راست گراف جهت‌دار انتخاب می‌شود. از این نقطه یک خط در هر جهتی رسم می‌شود که این خط گراف را در m نقطه قطع می‌کند. اگر m فرد باشد، چرخه تشکیل دهنده گراف فضای داخلی را نشان می‌دهد و اگر m زوج باشد چرخه تشکیل دهنده گراف فضای خارجی را نشان می‌دهد (شکل ۳-۲۷).



شکل ۳-۲۷: روش نقطه‌ای (الف) تعداد برخورد فرد (فضای داخلی) (ب) تعداد برخورد زوج (فضای خارجی) [۹]

۳-۲-۳-۳ روش مساحت

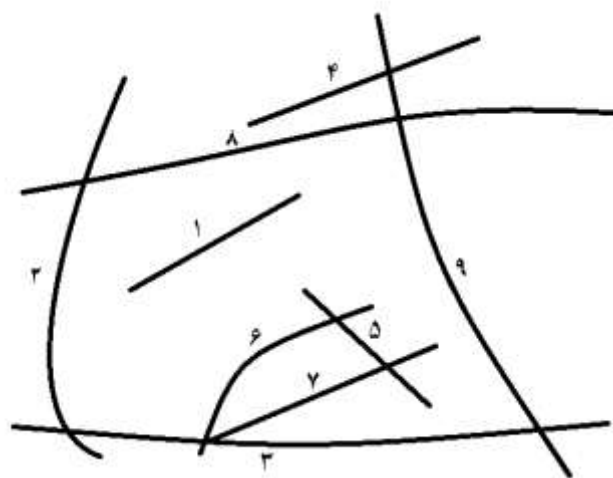
رئوس هر گراف به عنوان رئوس یک بلوک در نظر گرفته می‌شود. مساحت حاصل از این رئوس برای هر بلوک از طریق رابطه زیر محاسبه می‌شود:

$$A = \frac{1}{2} \sum_{i=1}^{s-1} \begin{vmatrix} 1 & x_0 & y_0 \\ 1 & x_i & y_i \\ 1 & x_{i+1} & y_{i+1} \end{vmatrix} \quad (۲۱-۳)$$

که در آن x و y مختصات نقاط رئوس بلوک هستند و s تعداد اضلاع بلوک می‌باشد. اگر مساحت بدست آمده از طریق رابطه فوق منفی باشد، چرخه یال‌های گراف فضای داخل گراف را نشان می‌دهند؛ و اگر این مساحت مثبت باشد چرخه یال‌های گراف فضای خارجی گراف را نشان می‌دهند. در واقع چرخه‌ایی از یال‌ها که دارای مساحت منفی هستند، بلوک‌ها را تشکیل می‌دهند.

۳-۲-۳-۶ یک مثال برای درک بهتر الگوریتم

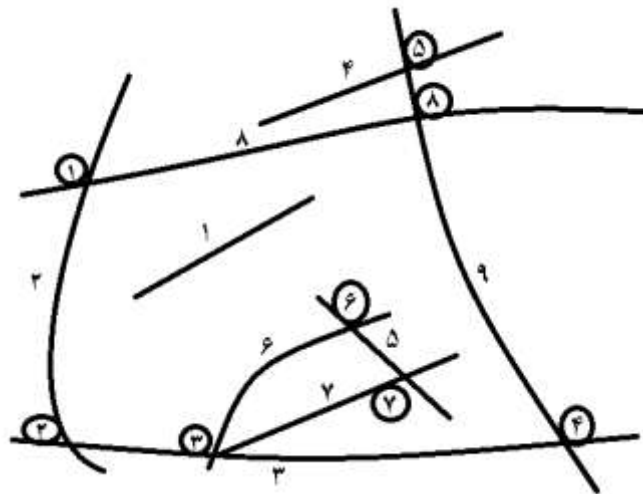
تعداد ۹ ناپیوستگی همان‌گونه که در شکل (۳-۲۸) نشان داده شده است در صفحه وجود دارند. هدف بدست آوردن مساحت و رئوس بلوک‌های موجود در این شبکه ناپیوستگی با استفاده از الگوریتم شی است.



شکل ۳-۲۸: نمای گرافیکی ناپیوستگی‌ها [۹]

ابتدا نقاط برخورد بین ناپیوستگی‌ها محاسبه شده و ماتریس اتصال رئوس (Q) برای این مثال با توجه به شکل (۳-۲۹) به صورت زیر محاسبه می‌شود:

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 0 \\ 2 & 3 & 4 \\ 5 & 0 & 0 \\ 6 & 7 & 0 \\ 3 & 6 & 0 \\ 3 & 7 & 0 \\ 1 & 8 & 0 \\ 5 & 8 & 4 \end{bmatrix}$$



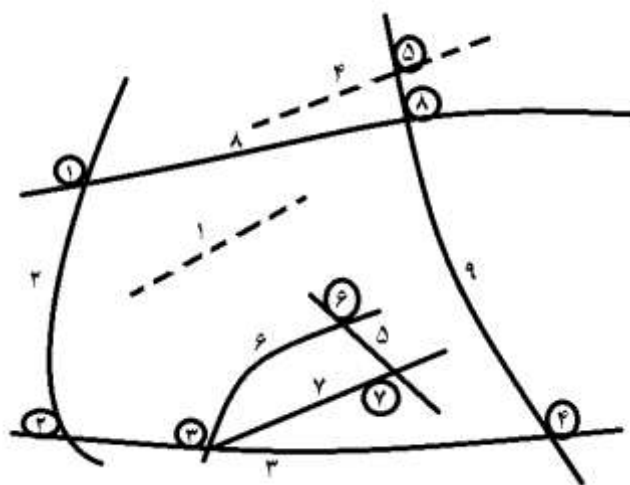
شکل ۳-۲۹: نقاط برخورد ناپیوستگی‌های شبکه بلوکی [۹]

در این ماتریس هر سطر مربوط به یک ناپیوستگی و درایه‌های روی آن نقاط برخورد آن ناپیوستگی با دیگر ناپیوستگی‌ها است. تمام ناپیوستگی‌هایی که در نقطه خاصی برخورد دارند را می‌توان مشخص کرد. به عنوان مثال رأس ۵ در سطرهای ۴ و ۹ وجود دارد و این بدان معنا است که این رأس در ناپیوستگی‌های ۴ و ۹ مشترک است. در این ماتریس هر سطری که کمتر از ۲ مقدار غیر صفر در درایه‌های آن وجود داشته باشد باید از ماتریس حذف شود. به عنوان مثال در سطر اول ماتریس Q برای مثال مورد بحث هیچ مقدار غیر صفری وجود ندارد، بنابراین این سطر باید از ماتریس حذف

شود. هنگامی که درایه‌ای از یک سطر ماتریس Q حذف می‌شود باید این مقدار از کل ماتریس نیز حذف شود. بنابراین برخی از سطرها را دیگر نیز ممکن است تعداد درایه‌های غیر صفر آن‌ها کمتر از ۲ شود. ماتریس اصلاح شده نهایی (Qr) برای این مثال به صورت زیر می‌باشد:

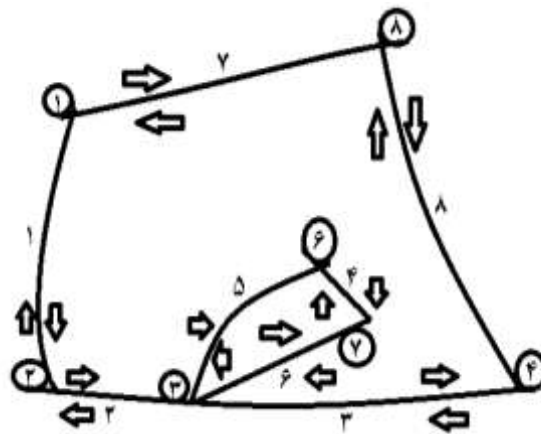
$$Qr = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 3 & 4 \\ 6 & 7 & 0 \\ 3 & 6 & 0 \\ 3 & 7 & 0 \\ 1 & 8 & 0 \\ 8 & 4 & 0 \end{bmatrix}$$

شکل (۳-۳۰) نتیجه ماتریس اصلاح شده اتصال رئوس (Qr) یا همان برش درختی را برای مثال مذکور نشان می‌دهد. در این شکل ناپیوستگی‌هایی که با خط چین مشخص شده‌اند از شبکه بلوکی حذف می‌شوند.



شکل ۳-۳۰: برش درختی ناپیوستگی‌ها [۹]

پس از این مرحله همانطور که در شکل (۳-۳۱) نشان داده شده است، برای شناسایی بلوک‌ها شبکه بلوکی به یک گراف جهت‌دار تبدیل می‌شود. رئوس گراف نقاط برخورد و یال‌های آن خطوط واصل بین این رئوس است.



شکل ۳-۳۱: گراف تشکیل شده از نتیجه برش درختی [۹]

در این گراف همه یال‌ها دارای دو جهت رفت و برگشتی هستند و برای شناسایی کامل بلوک‌ها باید از هر دو جهت این یال‌ها استفاده شود. ماتریس جهت (M) با توجه به نحوه تشکیل آن که ذکر شد برای این مثال به صورت زیر می‌باشد:

$$M = \begin{bmatrix} 11 & - \\ 11 & 11 \\ 11 & - \\ 11 & - \\ 11 & - \\ 11 & - \\ 11 & - \end{bmatrix}$$

هر دو رأس متوالی در ماتریس Qr تشکیل یک یال را می‌دهند. به عنوان مثال در سطر اول این ماتریس برای مثال مذکور دو رأس ۱ و ۲ وجود دارند؛ این دو رأس تشکیل یک یال را می‌دهند و در ماتریس جهت به ازای این یال درایه ۱۱ قرار داده می‌شود. عملیات شناسایی بلوک‌ها زمانی به پایان می‌رسد که تمام درایه‌های ماتریس M صفر شوند. پس از این مرحله بلوک‌های واقعی با توجه به روش‌های بیان شده مشخص می‌شوند. نتیجه این الگوریتم برای مثال مذکور در جدول (۳-۱) مشاهده می‌شود.

جدول ۳-۱: بلوک‌های شناسایی شده در شبکه ناپیوستگی

شماره بلوک	شماره یال‌های بلوک	شماره رئوس بلوک
۱	۱ و ۲ و ۵ و ۴ و ۶ و ۳ و ۸ و ۷	۱ و ۲ و ۳ و ۶ و ۷ و ۴ و ۸
۲	۴ و ۵ و ۶	۳ و ۶ و ۷

الگوریتم ارائه شده توسط شی ناپیوستگی‌های منحنی شکل را مورد بررسی قرار می‌دهد و قادر است در هر ترکیبی از ناپیوستگی‌هایی با طول محدود، بلوک‌های ایجاد شده را شناسایی نماید. اما معایب این روش عبارتند از:

- این الگوریتم به دلیل جهت‌دار کردن یال‌های گراف، یک الگوریتم طولانی است.
- در مورد نحوه محاسبه زاویه بین یال‌ها در گراف اصلی توضیحی داده نشده است.

۳-۲-۴ الگوریتم تئوری بلوک مبتنی بر روش‌های برداری

یکی از روش‌های تشخیص بلوک‌ها در دو و سه بعد در توده‌سنگ تئوری بلوکی است که در سنگ‌های سخت مورد استفاده قرار می‌گیرد. در این روش به بررسی جابجایی بلوک‌های بحرانی نیز پرداخته می‌شود. این روش توسط گودمن و شی ارائه شده است که در ادامه توضیح داده می‌شود [۸]:

فرضیات تئوری بلوکی عبارتند از:

- فرض می‌شود که سطح تمام درزه‌ها مسطح می‌باشد. این فرض برای بسیاری از درزه‌ها و گسل‌ها همراه با واقعیت است. اما برای سطوح لایه بندی نظیر آنچه در یال چین‌ها اتفاق می‌افتد، کاملاً اشتباه است. فرض مسطح بودن درزه‌ها به تشریح شکل بلوک‌ها با استفاده از معادلات برداری خطی کمک می‌کند و از طرف دیگر تعمیم تئوری برای صفحات منحنی شکل نیز آسان است.

- فرض می‌شود که سطوح درزه‌ها در تمام حجم مورد بررسی به طور کامل گسترش یافته است. این بدان معنا است که هیچ یک از ناپیوستگی‌ها به ناحیه داخل بلوک کلیدی محدود نمی‌شود. بدین ترتیب تمام بلوک‌ها به طور کامل توسط سطوح درزه موجود مشخص شده و ایجاد هیچ ترک جدیدی در جابجایی بلوک‌ها مورد بررسی قرار نمی‌گیرد.
- فرض می‌شود که بلوک‌های حاصل از تقاطع درزه‌ها صلب هستند. در این صورت تغییر شکل در بلوک به وجود نیامده و شکل اولیه آن تغییر نخواهد کرد.

۳-۲-۴-۱ حجم، لبه‌ها و گوشه‌های بلوک n وجهی

تقاطع صفحه درزه‌ها با یکدیگر بلوک‌هایی با اشکال مختلف به وجود می‌آورد که اغلب بیش از چهار وجه دارند. برای محاسبه حجم این بلوک‌ها آن‌ها را به بلوک‌های چهار وجهی تقسیم کرده و حجم آن‌ها را با استفاده از رابطه (۳-۲۵) محاسبه می‌کنند.

$$V = \frac{1}{6} \begin{vmatrix} 1 & X_1 & Y_1 & Z_1 \\ 1 & X_2 & Y_2 & Z_2 \\ 1 & X_3 & Y_3 & Z_3 \\ 1 & X_4 & Y_4 & Z_4 \end{vmatrix} \quad (۳-۲۵)$$

در این رابطه مختصات طول، عرض و ارتفاع نقاط چهار وجهی مورد استفاده قرار می‌گیرد. بلوک سه بعدی با n وجه از n صفحه تشکیل شده است. هر صفحه کل فضا را به نیم فضای بالایی (تمام فضای بالاتر از این صفحه) و نیم فضای پایینی تقسیم می‌کند. نیم فضای بالایی را با U_i و نیم فضای پایینی را با L_i نشان می‌دهند. تقاطع نیم فضاهای صفحات تعیین کننده ابعاد و شکل بلوک‌ها است. معادله یک صفحه در فضا به صورت زیر نوشته می‌شود.

$$\begin{cases} AX + BY + CZ = D \\ A = \sin(\alpha) \sin(\beta), B = \sin(\alpha) \cos(\beta), C = \cos(\alpha) \end{cases} \quad (۳-۲۶)$$

که در این رابطه α و β شیب و زاویه شیب هستند.

طی مراحل زیر یک بلوک با رئوس آن مشخص می‌شود:

• مرحله اول: برای هر صفحه i ثابت‌های A_i, B_i, C_i و D_i را باید تعیین کرد. این ضرایب با استفاده از رابطه (۳-۲۶) و معلوم بودن شیب و جهت شیب صفحه i محاسبه می‌شوند. زاویه شیب همیشه بین 0° و 90° بوده و بنابراین C_i همیشه دارای مقدار مثبت است. بدین ترتیب از دو جهت ممکن برای بردار نرمال، جهتی که به سمت بالا باشد، انتخاب می‌شود. ضریب D_i نیز با داشتن یک نقطه از صفحه بدست می‌آید.

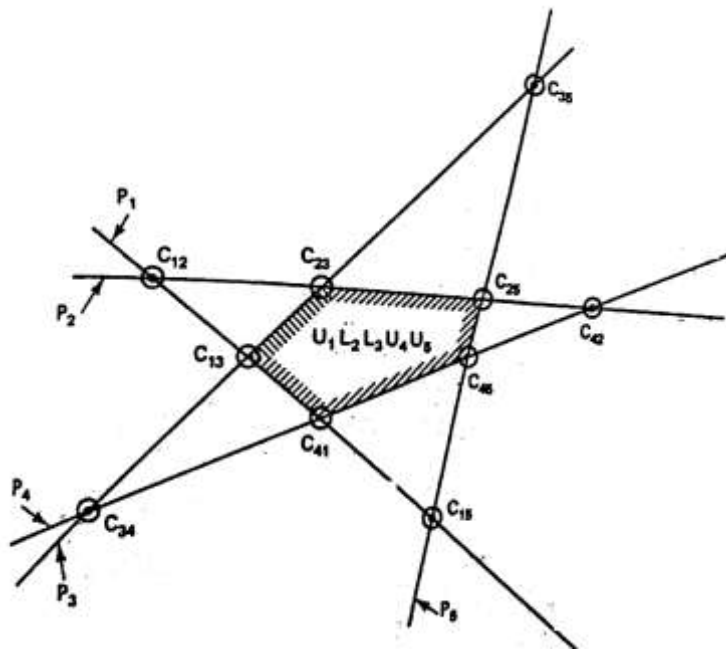
• مرحله دوم: مختصات تمام گوشه‌های بلوک را باید تعیین کرد. مختصات گوشه C_{ijk} که نقطه تقاطع سه صفحه i, j و k است از حل دستگاه معادله این سه صفحه محاسبه می‌شود. تعداد گوشه‌های محاسبه شده با تعداد ترکیبات n شیء از ۳ شیء برابر است $(C(n, 3))$. به عنوان مثال برای یک متوازی السطوح با ۶ صفحه و ۲۰ گوشه محاسبه می‌شود که ۸ تای آن‌ها رئوس اصلی متوازی السطوح هستند.

• مرحله سوم: برای هر وجه (صفحه) m و تمامی گوشه‌های C_{ijk} اگر رابطه زیر برقرار باشد در این صورت این گوشه یک گوشه از بلوک است.

$$\begin{cases} A_m X_{ijk} + B_m Y_{ijk} + C_m Z_{ijk} \geq D_m & \text{اگر بلوک در نیم فضای بالایی صفحه } m \text{ باشد} \\ A_m X_{ijk} + B_m Y_{ijk} + C_m Z_{ijk} \leq D_m & \text{اگر بلوک در نیم فضای پایینی صفحه } m \text{ باشد} \end{cases} \quad (3-27)$$

• مرحله سوم را باید برای تمامی وجوه تکرار و گوشه‌های بلوک‌ها را مشخص کرد. با یک مثال که در شکل (۳-۳۲) نشان داده شده است این مرحله و مرحله قبل توضیح داده می‌شود. این شکل یک ۵ ضلعی را نشان می‌دهد که از خطوط P_1 تا P_5 ساخته شده و تعداد تقاطع‌های این خطوط برابر با $C_5^2 = 10$ است و با C_{ij} مشخص شده‌اند. بلوک خاصی نظیر U_1, L_2, L_3 و U_4 و U_5 در نظر گرفته می‌شود. با در نظر گرفتن U_1 و بررسی رابطه (۳-۲۷) گوشه C_{34}

حذف می‌شود و به همین ترتیب گوشه‌های بلوک مورد نظر C_{13} ، C_{23} ، C_{25} ، C_{45} و C_{41} هستند.



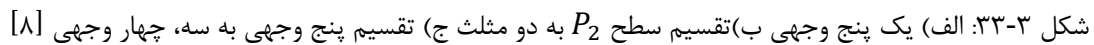
شکل ۳-۳۲: گوشه‌های واقعی یک چند ضلعی [۸]

- مرحله چهارم: در این مرحله وجوه بلوک باید مشخص شود. یک وجه واقعی از سه یا تعداد بیشتری رأس تشکیل شده است که دارای یک اندیس مشترک می‌باشند. به عنوان مثال برای وجه m ناحیه مثلثی بین گوشه‌های C_{1m2} ، C_{34m} و C_{m25} است.
- مرحله پنجم: در این مرحله لبه‌های بلوک باید مشخص شود. لبه واقعی، خط بین یک جفت رأس واقعی C_{ijk} خواهد بود که دارای دو اندیس مشترک باشند. برای مثال خط اتصال دهنده گوشه‌های C_{ij3} و C_{4ij} تشکیل دهنده یک لبه موازی با فصل مشترک i و j است.
- مرحله ششم: در این مرحله چند وجهی مشخص شده به هرم‌های چند وجهی و سپس به هرم‌های چهار وجهی تقسیم می‌شود. گوشه واقعی C_{ijk} به عنوان رأس هرم در نظر گرفته می‌شود. انتخاب این رأس به صورت اختیاری است و فقط یک رأس باید انتخاب شود. رأس

C_{ijk} نقطه تقاطع سطوح i , j و k است. بجز این سه صفحه بقیه سطوح بلوک یعنی $n-3$ وجه دیگر بلوک را به مثلث‌هایی به این شرح تقسیم می‌کند. به طور کلی هر وجه m یک چند ضلعی متشکل از t رأس است. گوشه‌های سطح m و اتصال آن به نقاط انتهایی هر لبه این سطح، سطح m را به مثلث‌هایی تقسیم می‌کند.

• مرحله هفتم: گوشه‌های مثلث مربوط به سطوح $n-3$ را به رأس هرم یعنی گوشه C_{ijk} باید اتصال داد تا مجموعه‌ای از چهار وجهی‌ها بدست آید که مجموع حجم آن‌ها برابر با حجم بلوک است.

شکل (۳-۳۳) مثالی از روش تشریح شده در مراحل ششم و هفتم را نشان می‌دهد. برای تبدیل پنج وجهی این شکل به بلوک‌های چهار وجهی ابتدا گوشه C_{135} به دلخواه انتخاب می‌شود. بدین ترتیب سطوح P_1 , P_3 و P_5 نیاز به مثلث بندی نخواهند داشت و از سطوح باقی مانده یعنی P_2 و P_4 ، سطح P_2 به دو مثلث $I(C_{235}, C_{234}, C_{124})$ و $II(C_{235}, C_{125}, C_{124})$ تقسیم می‌شود. که در قسمت (ب) شکل (۳-۳۳) نشان داده شده است. سطح P_4 خود یک مثلث است و نیازی به مثلث بندی ندارد. پس از مثلث بندی این مثلث‌ها به رأس C_{135} متصل می‌شوند و بلوک پنج وجهی به سه بلوک چهار وجهی تقسیم می‌شود (قسمت ج) شکل (۳-۳۳).



۵-۲-۳ الگوریتم مبتنی بر ماتریس‌ها

۱- ورود هندسه ناپیوستگی‌ها، دسته درزه‌ها و ابعاد محدوده

 $\vee \wedge$

۳- یافتن یال‌ها و مرتب سازی آن‌ها

۴- تشکیل ماتریس‌های اساسی

۵- شناسایی بلوک‌ها

این الگوریتم به طور کامل در فصل چهارم توضیح داده خواهد شد.

۳-۳ نتیجه‌گیری

برای مدل‌سازی کامپیوتری توده‌سنگ درزه‌دار ابتدا درزه‌ها را شبیه سازی کرده یعنی با توجه به درزه‌ها برداشت شده از توده‌سنگ، یک الگوی مناسب توسط روش‌های آماری و زمین آماری برای توده‌سنگ تعریف می‌شود. پس از آن برای تعیین هندسه توده‌سنگ و بلوک‌های موجود در آن از الگوریتم‌های مختلف همانند الگوریتم مبتنی بر تئوری گرافی، الگوریتم مبتنی بر تکنیک‌های عددی، الگوریتم ماتریسی و الگوریتم مبتنی بر روش‌های برداری و ... استفاده می‌شود. تفاوت این الگوریتم‌ها در روش مورد استفاده است. همانطور که اشاره شد الگوریتم‌های مورد بحث در این فصل دارای مزایا و معایبی هستند که برای پیاده‌سازی و تهیه برنامه کامپیوتری حائز اهمیت است. در فصل بعد روش مبتنی بر ماتریس‌ها با جزئیات بیشتر بررسی می‌شود.

فصل چهارم

برنامه کامپیوتری تعیین هندسه

بلوک‌های توده‌سنگ

۴-۱ مقدمه

خصوصیات هندسی توده‌سنگ و بلوک‌های آن همانند شکل و حجم بلوک‌ها در کاربردهای مختلف از جمله بلوک‌دهی سنگ‌های ساختمانی، انرژی خردایش و طراحی آتشفشانی، پایداری بلوک‌های سنگی و رقیق شدگی ماده معدنی در روش‌های استخراج تخریبی و ... استفاده می‌شود. برای تعیین مشخصات هندسی نیاز به مدل‌سازی کامپیوتری توده‌سنگ و بلوک‌های آن است. برای این کار از یک الگوریتم مناسب باید استفاده کرد. تعدادی از الگوریتم‌های دو بعدی تعیین شکل و مساحت بلوک‌های توده‌سنگ در فصل قبل بررسی شد. این الگوریتم‌ها دارای پیچیدگی برای پیاده‌سازی و طراحی برنامه کامپیوتری هستند. در ادامه به طور دقیق‌تر به دلیل عدم استفاده از این الگوریتم‌ها پرداخته می‌شود. الگوریتم مورد استفاده در این پایان‌نامه، الگوریتم ماتریسی دو بعدی است که برنامه کامپیوتری بر اساس آن طراحی می‌شود. این الگوریتم بر مبنای استفاده از ماتریس‌های اساسی است. در این فصل ابتدا به بررسی الگوریتم‌های موجود و دلیل انتخاب الگوریتم ماتریسی برای تهیه برنامه کامپیوتری پرداخته می‌شود. در ادامه الگوریتم ماتریسی دو بعدی به صورت کامل توضیح داده می‌شود و پس از آن مثالی برای درک الگوریتم ماتریسی طبق مراحل این الگوریتم حل خواهد شد. نحوه تهیه برنامه کامپیوتری و انواع داده‌ها ورودی و خروجی این برنامه در ادامه توضیح داده می‌شود. در پایان نیز چند مثال کاربردی در زمینه معدن به همراه حل توسط برنامه تهیه شده در این فصل بررسی می‌شوند.

۴-۲ بررسی الگوریتم‌های موجود و انتخاب الگوریتم مناسب

الگوریتم‌های مختلفی برای تعیین هندسه بلوک‌های توده‌سنگ وجود دارد. از بین الگوریتم‌های دو بعدی، الگوریتم مبتنی بر تئوری گراف، الگوریتم مبتنی بر تکنیک‌های عددی، الگوریتم مبتنی بر گراف‌های جهت‌دار، الگوریتم مبتنی بر روش‌های عددی و الگوریتم مبتنی بر ماتریس‌های اساسی در

فصل قبل بررسی شدند. تفاوت این الگوریتم‌ها در دشواری‌های زمانی و حجمی آن‌ها است. منظور از دشواری زمانی و حجمی مدت زمان لازم و میزان حافظه مورد نیاز برای پردازش اطلاعات در الگوریتم است. بهینه‌ترین الگوریتم از نظر دشواری‌های زمانی و حجمی در حل این مسأله الگوریتم مبتنی بر تئوری گراف است. این الگوریتم دارای پیچیدگی و دشواری‌های ریاضی بوده و نیاز به زمان زیاد برای اجرا و پیاده‌سازی آن می‌باشد. همچنین با توجه به این که روش مذکور از تلفیق الگوریتم‌های متعددی تشکیل شده، درک این روش پیچیده و تهیه برنامه کامپیوتری آن مشکل می‌باشد.

الگوریتم مبتنی بر تکنیک‌های عددی با توجه به در نظر گرفتن دو جهت برای هر ناپیوستگی یا درزه یک الگوریتم با تعداد مراحل زیاد است. همچنین نحوه اعمال عملگر مرزی در الگوریتم به صورت واضح توضیح داده نشده است.

الگوریتم ارائه شده توسط شی نیز به صورت واضح توضیح داده نشده است؛ به عنوان مثال نحوه محاسبه زاویه بین ناپیوستگی‌ها در این الگوریتم بیان نشده است. همچنین با توجه به جهت‌دار کردن گراف اصلی تشکیل شده توسط نقاط برخورد، یک الگوریتم طولانی نیز می‌باشد.

الگوریتم مبتنی بر روش‌های برداری بر اساس شمارش کامل خطوط برای هر بلوک بر مبنا حضور بلوک در نیم فضای بالایی یا پایینی است. در نتیجه یک روش طولانی می‌باشد و دارای پیچیدگی برای پیاده‌سازی است.

برای تهیه برنامه کامپیوتری در این پایان نامه از الگوریتم مبتنی بر ماتریس‌ها در دو بعد بنا بر دلایل زیر استفاده شده است:

- زمان محاسبات کم

با توجه به این که مراحل این الگوریتم کمتر از سایر الگوریتم‌ها می‌باشد و روند ساده‌تری برای مراحل مختلف ارائه شده است، زمان محاسبات در این الگوریتم از سایر الگوریتم‌ها کمتر می‌باشد.

- فهم ساده و پیاده سازی راحت

مراحل مختلف این الگوریتم به صورت کاملاً واضح توضیح داده شده است و در نتیجه فهم و پیاده سازی آن آسان‌تر از سایر الگوریتم‌ها می‌باشد.

- این الگوریتم بر مبنای عملیات بر روی ماتریس‌ها است و در محیط نرم افزار MATLAB این منظور آسان‌تر پیاده سازی می‌شود.

۳-۴ تشریح کامل الگوریتم ماتریسی دو بعدی

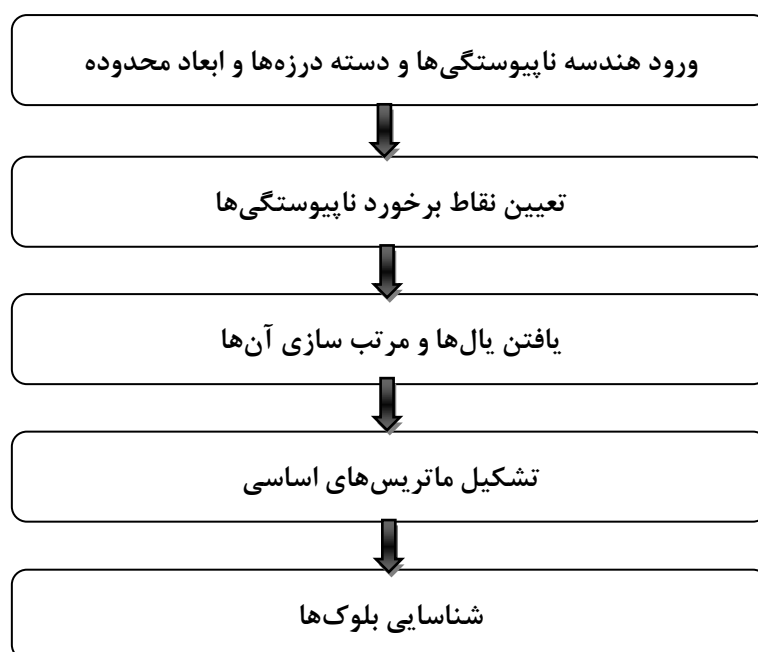
الگوریتم مبتنی بر ماتریسی دو بعدی یک روش ابتکاری می‌باشد. این الگوریتم در سال ۲۰۱۱ توسط خیشوند و همکاران ارائه و از جدیدترین الگوریتم‌های شناسایی بلوک‌ها یا چند ضلعی‌ها در دو بعد است. این الگوریتم بر مبنای عملیات بر روی ماتریس‌های مربعی است که در ادامه توضیح داده می‌شود [۱۶]:

فرضیات این روش عبارتند از:

- درزه‌ها مسطح و بدون انحنا هستند.

• دسته درزه‌ها در مدل با طول نامحدود در نظر گرفته می‌شوند. هر چند این روش قادر است در قالب مدل‌های شبکه شکستگی‌های مجزا یا DFN^1 ، ناپیوستگی‌ها با طول محدود را نیز در نظر بگیرد.

در این الگوریتم ابتدا هندسه ناپیوستگی‌ها و دسته درزه‌ها توسط داده‌های ورودی ساخته و پس از آن تقاطع ناپیوستگی‌های محاسبه می‌شود. پس از محاسبه تقاطع ناپیوستگی‌ها باید ناپیوستگی‌هایی که در تشکیل بلوک‌ها مؤثر نیستند را حذف کرد. پس از آن ماتریس‌های اساسی این روش که بر اساس اتصال رئوس و یال‌ها در مدل هستند، برای محاسبات بعدی باید تشکیل شوند. در نهایت طی مراحل که در ادامه توضیح داده خواهد شد بلوک‌های موجود در مدل شناسایی می‌شوند. روند کلی این الگوریتم در شکل (۴-۱) نشان داده شده است.

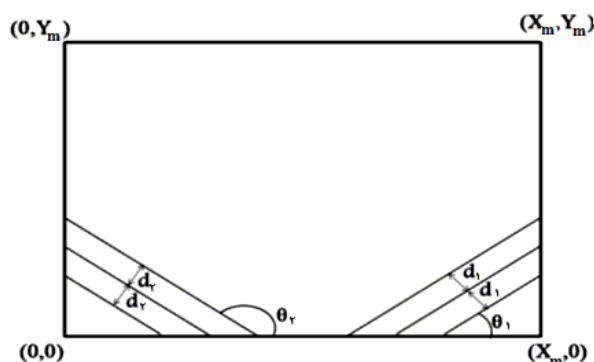


شکل ۴-۱: روند شناسایی بلوک‌ها در روش ماتریسی [۱۶]

¹ Discrete Fracture Network

۴-۳-۱ ورود هندسه ناپیوستگی‌ها، دسته درزه‌ها و ابعاد محدوده

هندسه ناپیوستگی‌های ورودی در مدل شامل معرفی درزه و دسته درزه‌ها است. مرحله‌ی اول در تشکیل سیستم ناپیوستگی‌ها وارد کردن مختصات یک محدوده یا بلوک اصلی که قرار است ناپیوستگی‌ها درون آن بررسی شوند و مختصات دسته درزه‌ها و ناپیوستگی‌های منفرد می‌باشد. البته این محدوده یا بلوک اصلی اختیاری است و می‌توان فقط مختصات درزه‌ها را در ماتریسی ذخیره کرد و با استفاده از آن‌ها بلوک‌های تشکیل شده را محاسبه کرد. در حالت دو بعدی دسته درزه‌ها توسط فاصله‌داری درزه‌ها (d) و زاویه‌ی درزه‌ها با محور افقی (θ) تعریف می‌شوند به عنوان مثال دو دسته درزه به ترتیب با فاصله‌داری d_1 و d_2 زاویه شیب θ_1 و θ_2 در محدوده‌ای با ابعاد $X_m \times Y_m$ در شکل (۴-۲) نشان داده شده است.



شکل ۴-۲: بلوک اصلی و دسته درزه‌های متقاطع با آن [۱۶]

جدول (۴-۱) نحوه محاسبه نقاط شروع و پایان هر درزه که با بلوک اصلی برخورد دارد را نشان می‌دهد. در این جدول X_m و Y_m ابعاد بلوک اصلی هستند و m تعداد تکرار درزه‌ها در هر دسته درزه است. ستون اول این جدول مربوط به اندازه زاویه با افق است، ستون دوم موقعیت مختصات درزه‌ها را نسبت به ابعاد اصلی بلوک نشان می‌دهد و ستون‌های بعدی نحوه محاسبه مختصات نقاط ابتدا و انتهای هر یک از درزه‌ها را نشان می‌دهد. ناپیوستگی‌های منفرد مانند گسل‌ها را می‌توان به وسیله نقاط شروع و پایان به صورت دستی در فایل ورودی برنامه اضافه کرد.

جدول ۴-۱: مختصات نقاط ابتدا و انتها درزه‌ها در هر دسته درزه [۱۶]

y_e	x_e	y_s	x_s	موقعیت	زاویه با افق Θ
md	X_m	Md	0	$md \leq Y_m$	$\Theta = 0$
$\frac{md}{\cos \theta}$	X_m	0	$X_m - \frac{md}{\sin \theta}$	$\frac{md}{\sin \theta} \leq X_m$	$0 < \theta < \frac{\pi}{2}$
				$\frac{md}{\cos \theta} \leq Y_m$	
$\frac{md}{\cos \theta}$	X_m	$\left(\frac{md}{\sin \theta} - X_m\right) \tan \theta$	0	$\frac{md}{\sin \theta} > X_m$	
				$\frac{md}{\cos \theta} \leq Y_m$	
Y_m	$X_m - \frac{\frac{md}{\cos \theta} - Y_m}{\tan \theta}$	0	$X_m - \frac{md}{\sin \theta}$	$\frac{md}{\sin \theta} \leq X_m$	
				$\frac{md}{\cos \theta} > Y_m$	
Y_m	$X_m - \frac{\frac{md}{\cos \theta} - Y_m}{\tan \theta}$	$\left(\frac{md}{\sin \theta} - X_m\right) \tan \theta$	0	$\frac{md}{\sin \theta} > X_m$	
				$\frac{md}{\cos \theta} > Y_m$	
Y_m	Md	0	Md	$md \leq X_m$	$\Theta = \frac{\pi}{2}$
$\frac{md}{\cos(\pi - \theta)}$	0	0	$\frac{md}{\sin(\pi - \theta)}$	$\frac{md}{\sin(\pi - \theta)} \leq X_m$	$\frac{\pi}{2} < \theta < \pi$
				$\frac{md}{\cos(\pi - \theta)} \leq Y_m$	
$\frac{md}{\cos(\pi - \theta)}$	0	$\left(\frac{md}{\sin(\pi - \theta)} - X_m\right) \tan \theta$	X_m	$\frac{md}{\sin(\pi - \theta)} > X_m$	
				$\frac{md}{\cos(\pi - \theta)} \leq Y_m$	
Y_m	$\frac{\frac{md}{\cos(\pi - \theta)} - Y_m}{\tan(\pi - \theta)}$	0	$\frac{md}{\sin(\pi - \theta)}$	$\frac{md}{\sin(\pi - \theta)} \leq X_m$	
				$\frac{md}{\cos(\pi - \theta)} > Y_m$	
Y_m	$\frac{\frac{md}{\cos(\pi - \theta)} - Y_m}{\tan(\pi - \theta)}$	$\left(\frac{md}{\sin(\pi - \theta)} - X_m\right) \tan \theta$	X_m	$\frac{md}{\sin(\pi - \theta)} > X_m$	
				$\frac{md}{\cos(\pi - \theta)} > Y_m$	
Md	X_m	Md	0	$md \leq Y_m$	$\Theta = \pi$

۲-۳-۴ محاسبه نقاط برخورد ناپیوستگی‌ها

مرحله دوم در پروسه یافتن بلوک‌ها محاسبه نقاط برخورد بین تمامی خطوط ناپیوستگی‌های توده سنگ است. تمام ناپیوستگی‌ها به صورت صاف و مستقیم فرض می‌شوند و در محدوده مدل طول محدودی دارند. برای محاسبه نقاط برخورد بین ناپیوستگی‌ها فرض می‌شود d_i و d_j دو ناپیوستگی به شعاع r_i و r_j باشند (نصف طول ناپیوستگی) و دارای مختصات نقطه مرکزی $C_i(x_i^c, y_i^c)$ و $C_j(x_j^c, y_j^c)$ در یک سیستم مختصات XY-O باشند. بردارهای نرمال این دو ناپیوستگی به ترتیب زیر است:

$$n_i(n_x^i, n_y^i) = (\cos\theta_i, \sin\theta_i), n_j(n_x^j, n_y^j) = (\cos\theta_j, \sin\theta_j) \quad (1-4)$$

که در آن n_i و n_j بردار نرمال ناپیوستگی‌ها و θ_i و θ_j زاویه ناپیوستگی‌ها با افق هستند. فاصله بین مرکز C_i و مرکز C_j به وسیله رابطه (۲-۴) بدست می‌آید:

$$d_{ij}^c = \sqrt{(x_i^c - x_j^c)^2 + (y_i^c - y_j^c)^2} \quad (2-4)$$

که در آن x_i^c و x_j^c مختصات طول مرکز ناپیوستگی‌ها و y_i^c و y_j^c مختصات عرض مرکز ناپیوستگی‌ها هستند و d_{ij}^c فاصله بین دو مرکز ناپیوستگی است. اگر یکی از دو شرط زیر برقرار شود دو ناپیوستگی هیچ تقاطعی با هم ندارند.

$$d_{ij}^c > r_i + r_j \quad (3-4)$$

$$(n_i) \cdot (n_j) = n_x^i n_x^j + n_y^i n_y^j \quad (4-4)$$

اگر شرایط فوق برقرار باشد دو ناپیوستگی یا از هم خیلی دور و یا با همدیگر موازی هستند. در غیر این صورت دو ناپیوستگی ممکن است با توجه به موقعیت نسبی‌شان یک تقاطع داشته باشند.

اگر نقاط $P_1^i(x_1^i, y_1^i)$ و $P_2^i(x_2^i, y_2^i)$ شروع و پایان ناپیوستگی i ام باشند معادله این ناپیوستگی از طریق رابطه (۵-۴) محاسبه می‌شود.

$$\begin{cases} x = x_1^i + (x_2^i - x_1^i)t_i \\ y = y_1^i + (y_2^i - y_1^i)t_i \end{cases} \quad (۵-۴)$$

که در آن $0 \leq t_i \leq 1$ است. مختصات نقطه برخورد $v(x_s, y_s)$ دو ناپیوستگی از طریق رابطه (۶-۴) محاسبه می‌شود.

$$\begin{cases} x_s = \frac{1}{\Delta} [(x_1^j y_2^j - x_2^j y_1^j)(x_2^i - x_1^i) - (x_1^i y_2^i - x_2^i y_1^i)(x_2^j - x_1^j)] \\ y_s = \frac{1}{\Delta} [(x_1^i y_2^i - x_2^i y_1^i)(y_2^j - y_1^j) - (x_1^j y_2^j - x_2^j y_1^j)(y_2^i - y_1^i)] \end{cases} \quad (۶-۴)$$

$$\Delta = \begin{vmatrix} y_1^i - y_2^i & x_2^i - x_1^i \\ y_1^j - y_2^j & x_2^j - x_1^j \end{vmatrix}, \quad t_i = \frac{x_s - x_1^i}{x_2^i - x_1^i}, \quad t_j = \frac{x_s - x_1^j}{x_2^j - x_1^j}$$

که در آن x و y مختصات نقاط ابتدا و انتها هستند. اگر شرایط $0 \leq t_i \leq 1$ ، $0 \leq t_j \leq 1$ و $\Delta \neq 0$ همزمان برقرار باشند، دو ناپیوستگی دارای یک تقاطع با مختصات (x_s, y_s) هستند که یک رأس جدید را مشخص می‌کند. در غیر این صورت هیچ تقاطعی وجود نخواهد داشت. نقاط برخورد بین ناپیوستگی‌ها دو به دو محاسبه می‌شوند، ماتریس C برای ذخیره تمام نقاط برخورد به صورت زیر تعریف می‌شود:

$$C = [c_{ij}]_{N_d \times N_d}$$

اگر ناپیوستگی i و j با هم برخورد داشته باشند، درایه این ماتریس برابر یک و در غیر این صورت برابر صفر در نظر گرفته می‌شود که N_d تعداد همه نقاط برخورد در مدل است. ماتریس C یک ماتریس متقارن است و مجموع درایه‌های روی سطر i آن نشان دهنده تعداد نقاط برخورد بر روی ناپیوستگی i است. از این ماتریس برای مشخص کردن تعداد نقاط برخورد ناپیوستگی‌ها در مدل استفاده می‌شود.

نقاط برخورد محاسبه شده در ماتریس نقاط برخورد به نام Pin ذخیره می‌شود. این ماتریس به صورت زیر است:

$$Pin=[Pin_{ij}]_{Ni \times 2}$$

که در آن Ni تعداد کل نقاط برخورد است و درایه‌های این ماتریس مختصات طول و عرض نقاط برخورد هستند.

۳-۳-۴ یافتن یال‌ها و مرتب سازی آن‌ها

پس از محاسبه نقاط برخورد بین همه ناپیوستگی‌ها باید یال‌ها یا پاره خط‌های ایجاد شده بر روی آن‌ها تعیین شوند. به دلیل این که نقاط ابتدا و انتهای ناپیوستگی‌ها در صورت برخورد نداشتن با دیگر ناپیوستگی‌ها در تشکیل بلوک‌ها مؤثر نیستند، از ابتدا در تشکیل یال‌ها مورد استفاده قرار نمی‌گیرند. اگر مجموع نقاط برخورد یک ناپیوستگی با دیگر ناپیوستگی‌ها برابر m_i باشد، بیشترین تعداد یال‌های ایجاد شده بر روی این ناپیوستگی برابر با $m_i + 1$ است. برای یافتن یال‌های ایجاد شده بر روی هر ناپیوستگی ابتدا باید نقاط برخورد روی آن را بر اساس ترتیب قرار گیری، مرتب شوند. برای این کار از رابطه (۷-۴) استفاده می‌شود.

$$t_{ij} = \frac{x_{sj} - x_s^i}{x_e^i - x_s^i} = \frac{y_{sj} - y_s^i}{y_e^i - y_s^i}, \quad 0 \leq t_{ij} \leq 1 \quad (7-4)$$

که در آن x_{sj} و y_{sj} مختصات نقاط برخورد، x_s^i و y_s^i مختصات نقطه ابتدا ناپیوستگی و x_e^i و y_e^i مختصات نقطه انتهای ناپیوستگی است.

پارامتر t_{ij} بر اساس فاصله نقاط برخورد از ابتدای ناپیوستگی است و مرتب کردن نقاط بر اساس این پارامتر موجب مرتب سازی نقاط به ترتیب قرار گیری بر روی هر ناپیوستگی می‌شود. یال‌های تشکیل

شده در مدل از طریق این نقاط مرتب شده مشخص می‌شوند. به این صورت که در هر ناپیوستگی هر جفت نقطه برخورد پشت سر هم تشکیل یک یال را می‌دهند. برای تشخیص یال‌ها در هر ناپیوستگی ماتریس P برای ثبت نقاط برخورد مرتب شده بر روی هر ناپیوستگی تعریف می‌شود. این ماتریس به صورت زیر است:

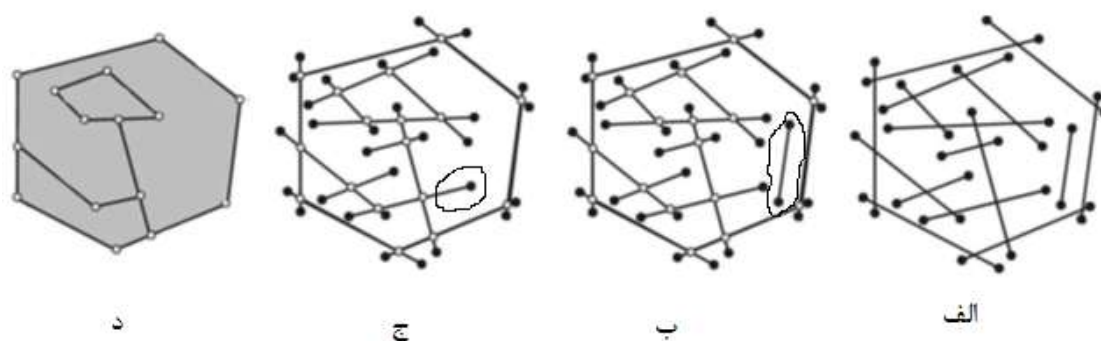
$$P=[p_{ij}]_{N_e \times N_p}$$

که در آن N_p ، ۲ برابر بیشترین تعداد نقاط برخورد بر روی ناپیوستگی‌ها و N_e تعداد کل ناپیوستگی‌ها است و درایه‌های این ماتریس مختصات نقاط برخورد بر روی هر شکستگی به طور جداگانه را نشان می‌دهد. ماتریس یال‌ها (Ed) با استفاده از ماتریس P تشکیل می‌شود.

$$Ed=[Ed_{ij}]_{N_e \times 4}$$

جفت درایه‌های روی هر سطر ماتریس P تشکیل دهنده ماتریس یال‌ها هستند. یعنی نقاط برخورد روی هر شکستگی به ترتیب دو به دو تشکیل یک یال را می‌دهند.

در مرحله بعد یال‌هایی که در تشکیل بلوک‌ها تأثیری ندارند باید حذف شوند. همانطور که در شکل (۳-۴) مشاهده می‌شود یال‌هایی که در کمتر از دو رأس با دیگر یال‌ها اشتراک دارند حذف می‌شوند. همانطور که در این شکل مشاهده می‌شود ابتدا نقاط برخورد خطوط محاسبه شده و در مرحله اول مرتب سازی، خطوط بدون برخورد از مدل حذف می‌شوند. در مرحله دوم خطوطی که دارای یک نقطه برخورد هستند حذف می‌شوند و در نهایت خطوطی که بلوک‌ها را تشکیل می‌دهند یا اضلاع بلوک‌ها باقی می‌مانند.



شکل ۳-۴: الف) یک شبکه ناپیوستگی ب) تعیین نقاط برخورد ج) حذف یال‌های بدون تقاطع د) مرتب سازی نهایی [۱۶]

برای حذف یال‌هایی که در تشکیل بلوک‌ها تأثیری ندارند ماتریس C_p به صورت زیر تعریف و محاسبه می‌شود:

$$C_p = [C_{p_{ij}}]_{N_1 \times N_1}$$

این ماتریس یک ماتریس مربعی بوده که N_1 تعداد تمام یال‌های مدل است. این ماتریس با استفاده از ماتریس یال‌ها (E_d) تشکیل می‌شود. اگر یال i با یال j نقطه مشترک داشته باشند درایه $C_{p_{ij}}$ این ماتریس برابر یک و در غیر این صورت این درایه برابر صفر خواهد بود. برای $i=j$ نیز درایه این ماتریس برابر صفر است.

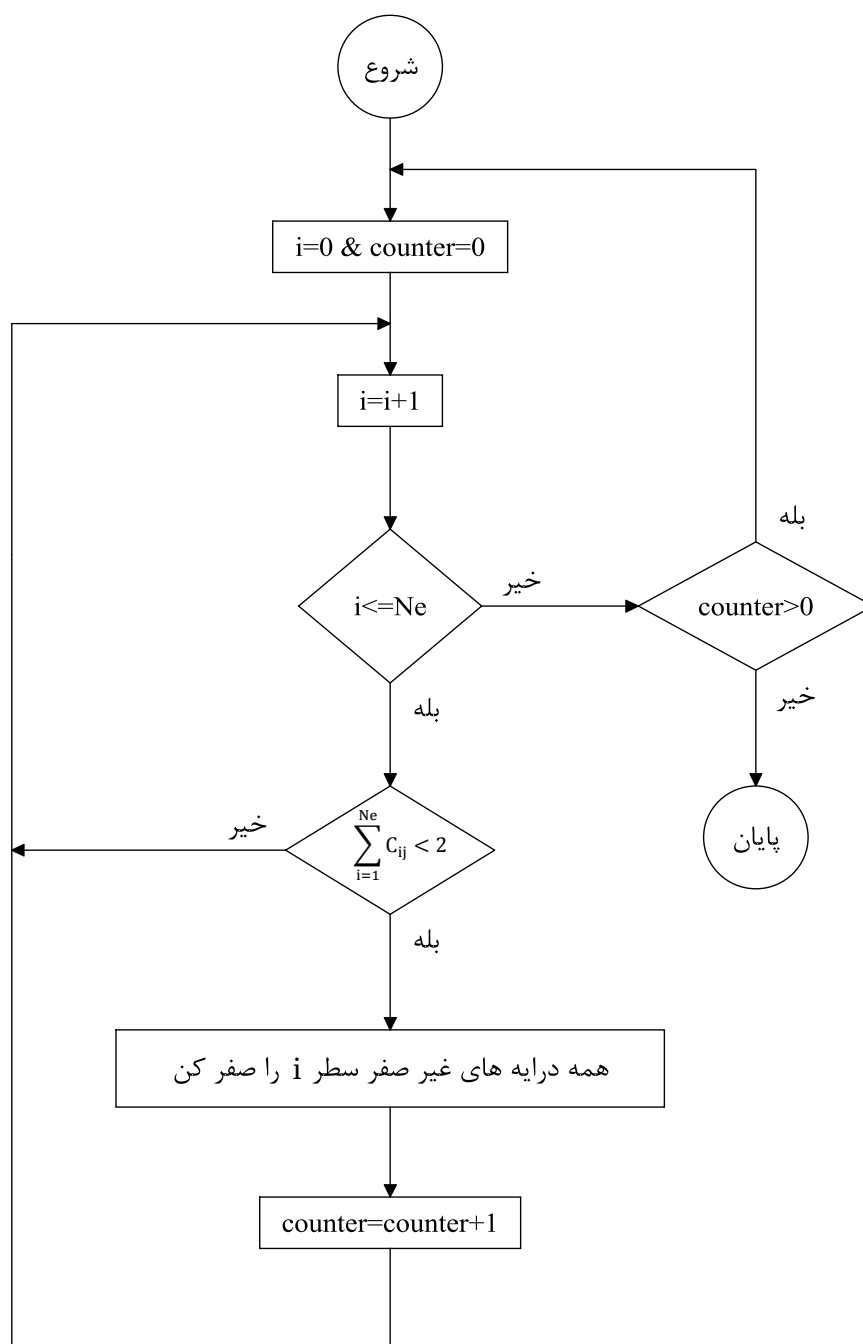
مراحل مرتب سازی یال‌ها به صورت زیر است :

- شروع
- ورودی: ماتریس C_p
- مرحله اول: اگر مجموع درایه‌های سطر i کمتر از ۲ بود یال را حذف کن.
- مرحله دوم: ماتریس C_p را اصلاح کن و به مرحله اول برگرد.
- مرحله سوم: مرتب سازی یال‌ها به پایان رسیده است.

• خروجی: یال‌های مرتب شده است.

• پایان

روندنمای مراحل مرتب سازی یال‌ها در شکل (۴-۴) نشان داده شده است.



شکل ۴-۴: روند نمای مراحل مرتب سازی یال‌ها [۱۶]

۴-۳-۴ تشکیل ماتریس‌های اساسی

ماتریس اتصال رئوس (Cv) اولین ماتریس اساسی در الگوریتم شناسایی بلوک‌ها است که نقاط برخورد یا رئوس را به یکدیگر متصل می‌کند. این ماتریس بر اساس نتیجه یال‌های مرتب شده تشکیل می‌شود. قبل از معرفی ماتریس اتصال رئوس لازم است ماتریس رئوس به صورت زیر تعریف شود:

$$Vx=[Vx_{ij}]_{N_v \times 2}$$

که در آن N_v تعداد کل رئوس است و درایه‌های این ماتریس مختصات نقاط برخورد هستند. این نقاط در تشکیل ماتریس اتصال رئوس مورد استفاده قرار می‌گیرند.

البته باید توجه داشت که در ماتریس برخورد از نقاط تکراری فقط یکی باید مورد استفاده قرار گیرد. ماتریس اتصال رئوس که نشان دهنده اتصال بین رئوس مدل است، به صورت زیر تعریف می‌شود:

$$Cv=[Cv_{ij}]_{N_v \times N_v}$$

در این ماتریس N_v تعداد کل رئوس و اگر رأس i و j در ماتریس رئوس Vx تشکیل یک یال را بدهند Cv_{ij} برابر با شماره یال تشکیل شده در ماتریس یال‌ها (Ed) است و در غیر این صورت مقدار درایه مربوطه برابر صفر می‌شود.

- قانون زاویه دست راستی دو بعدی

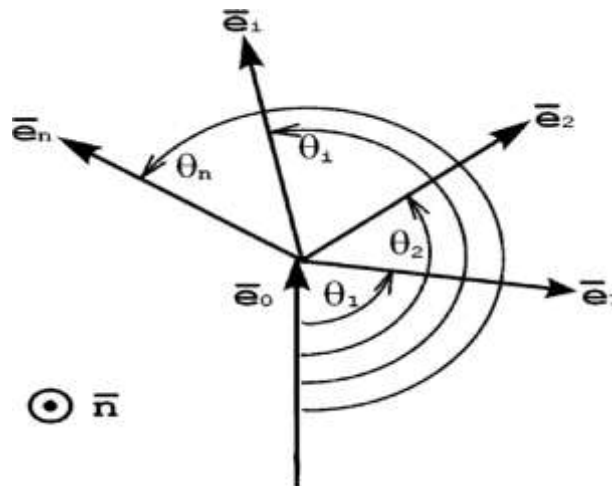
برای تشکیل ماتریس اساسی اتصال یال‌ها (Ce) نیاز به قانون زاویه دست راستی دو بعدی است. این قانون یک کلید برای یافتن چرخه‌ها در شبکه‌های ناپیوستگی است. طبق شکل (۴-۵) یال ممکن بعدی برای یک چرخه بعد از یال e_0 یال‌های $e_i (i = 1, 2, \dots, n)$ هستند، رابطه زیر این قانون را تشریح می‌کند.

$$\Delta = n \cdot (e_i \times e_0) = \begin{vmatrix} 0 & 0 & 1 \\ e_x^i & e_y^i & 0 \\ e_x^0 & e_y^0 & 0 \end{vmatrix} = e_x^i e_y^0 - e_y^i e_x^0$$

$$\text{If } \Delta \geq 0 \Rightarrow \theta_i = \pi - \cos^{-1}(e_i \cdot e_0) \quad (۸-۴)$$

$$\text{If } \Delta < 0 \Rightarrow \theta_i = \pi + \cos^{-1}(e_i \cdot e_0)$$

که e_0 و e_i بردار یکه یال مبنا و یال i ام هستند و n بردار نرمال صفحه‌ای که بردارها در آن هستند، است.



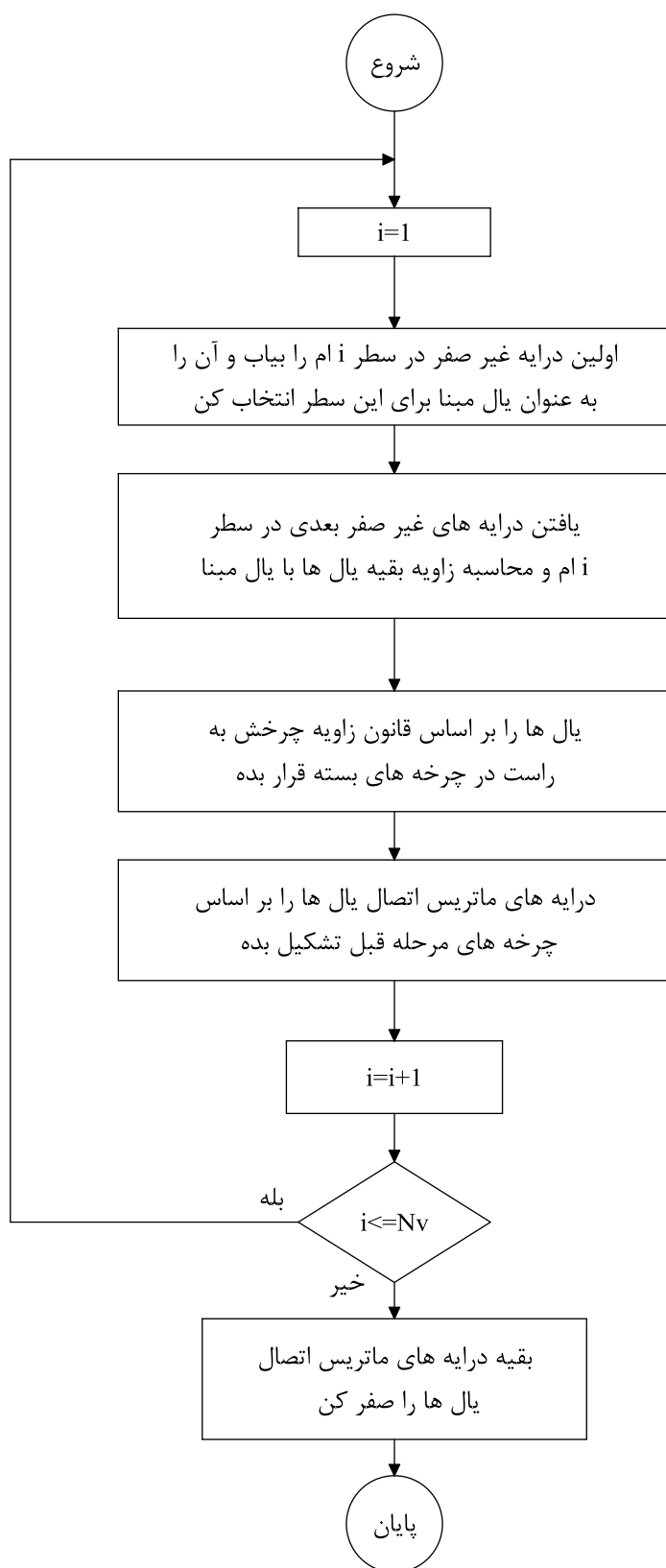
شکل ۴-۵: قانون زاویه دست راستی دو بعدی [۱۶]

ماتریس اتصال یال‌ها (Ce)، ماتریس اساسی دیگر در شناسایی بلوک‌ها است که نشان دهنده اتصال بین یال‌ها در مدل است. این ماتریس با استفاده از ماتریس اتصال رئوس به صورت زیر تشکیل می‌شود:

$$Ce = [Ce_{ij}]_{N_e \times N_e}$$

این ماتریس نیز یک ماتریس مربعی با ابعاد $N_e \times N_e$ است. که N_e تعداد همه یال‌های موجود در مدل است. و اگر یال i با یال j در رأس n اشتراک داشته باشند درایه Ce_{ij} این ماتریس برابر n می‌شود. مراحل تشکیل ماتریس اتصال یال‌ها به صورت زیر است:

- شروع
 - ورودی: ماتریس اتصال رئوس C_v
 - مرحله اول: $i=1$
 - مرحله دوم: یافتن اولین درایه غیر صفر در سطر i ام و تعریف آن به عنوان یال مبنا برای این سطر (درایه‌های غیر صفر هر سطر نشان دهنده شماره یال‌ها است).
 - مرحله سوم: یافتن درایه‌های غیر صفر بعدی در سطر i ام.
 - مرحله چهارم: محاسبه زاویه یال‌های مرحله سوم با یال مبنا بر اساس قانون زاویه دست راستی دو بعدی.
 - مرحله پنجم: مرتب کردن یال‌ها بر اساس زوایای بدست آمده در مرحله قبل از کوچکترین زاویه (به صورت صعودی)، یال مبنا به ابتدا و انتهای یال‌های فوق اضافه می‌شود.
 - مرحله ششم: شماره یال‌های مرحله قبل برای تشکیل ماتریس اتصال یال‌ها استفاده می‌شود. شماره سطر هر جفت یال در مرحله قبل درایه‌ای برای ماتریس اتصال یال‌ها است.
 - مرحله هفتم: $i=i+1$
 - مرحله هشتم: اگر $i \leq N_v$ به مرحله دوم برگرد در غیر این صورت برو به مرحله بعد.
 - مرحله نهم: تمام درایه‌های ماتریس اتصال رئوس که استفاده نشده‌اند را صفر کن.
 - خروجی: ماتریس اتصال یال‌ها است.
 - پایان
- در شکل (۴-۶) روندنمای مراحل تشکیل ماتریس اتصال یال‌ها نشان داده شده است.



شکل ۴-۶: روند نمای مراحل تشکیل ماتریس اتصال یال ها [۱۶]

۵-۳-۴ شناسایی بلوک‌ها

برای شناسایی بلوک‌ها و بدست آوردن مساحت آن‌ها از ماتریس‌های اتصال یال‌ها، ماتریس رئوس و ماتریس یال‌های مرتب شده استفاده می‌شود. مراحل شناسایی بلوک‌ها به صورت زیر است:

- شروع
- ورودی: ماتریس اتصال یال‌ها و $N_f = 0$ (تعداد بلوک‌ها)
- مرحله اول: $i=1, j=1$
- مرحله دوم: اگر $C_{ij}^e \neq 0$ برو به مرحله پنجم.
- مرحله سوم: $j=j+1$. اگر $j \leq N_e$ برو به مرحله دوم.
- مرحله چهارم: $j=1$ و $i=i+1$. اگر $i \leq N_e$ برو به مرحله دوم در غیر این صورت برو به مرحله دوازدهم.
- مرحله پنجم: C_{ij}^e را در پارامتر hold ذخیره کن و i اولین یال بلوک است.
- مرحله ششم: $C_{ij}^e = 0$ و i را ذخیره کن.
- مرحله هفتم: z را در i قرار بده و برو به مرحله بعد.
- مرحله هشتم: درایه‌های غیر صفر سطر i را پیدا کن، اگر بیش از یک درایه غیر صفر در این سطر وجود دارد، درایه‌ای را که با مقدار ذخیره شده برابر نیست انتخاب کن و ستون این درایه را در k ذخیره کن.
- مرحله نهم: $hold = C_{ik}^e$ و برو به مرحله بعد.
- مرحله دهم: اگر i برابر با یال مبنا است، $N_f = N_f + 1$ و به مرحله اول برگرد.
- مرحله یازدهم: k را در z قرار بده و به مرحله ششم برو.
- مرحله دوازدهم: شناسایی بلوک‌ها به پایان رسیده است.

• خروجی: رئوس بلوک‌های مدل است.

• پایان

در شکل (۷-۴) روندنمای مراحل تشکیل شناسایی بلوک‌ها نشان داده شده است.

در پایان مرحله شناسایی بلوک‌ها مختصات رئوس هر بلوک به طور جداگانه تعیین می‌شود. مساحت

بلوکی که از رئوس مشخصی تشکیل شده است از رابطه (۸-۴) قابل محاسبه است.

$$Area = \frac{1}{2} \begin{vmatrix} x_1 & x_2 & \dots & x_n & x_1 \\ y_1 & y_2 & \dots & y_n & y_1 \end{vmatrix} \quad (8-4)$$

که در آن x و y مختصات طول و عرض نقاط رئوس بلوک هستند. در این الگوریتم یک بلوک اصلی که

دربرگیرنده بقیه بلوک‌ها است نیز شناسایی می‌شود. علامت مساحت این بلوک همیشه مثبت و علامت

مساحت بقیه بلوک‌ها منفی است و می‌توان آن را به راحتی شناسایی و در صورت لزوم حذف کرد.

۶-۳-۴ مثال ۱ برای تشریح مراحل الگوریتم ماتریسی

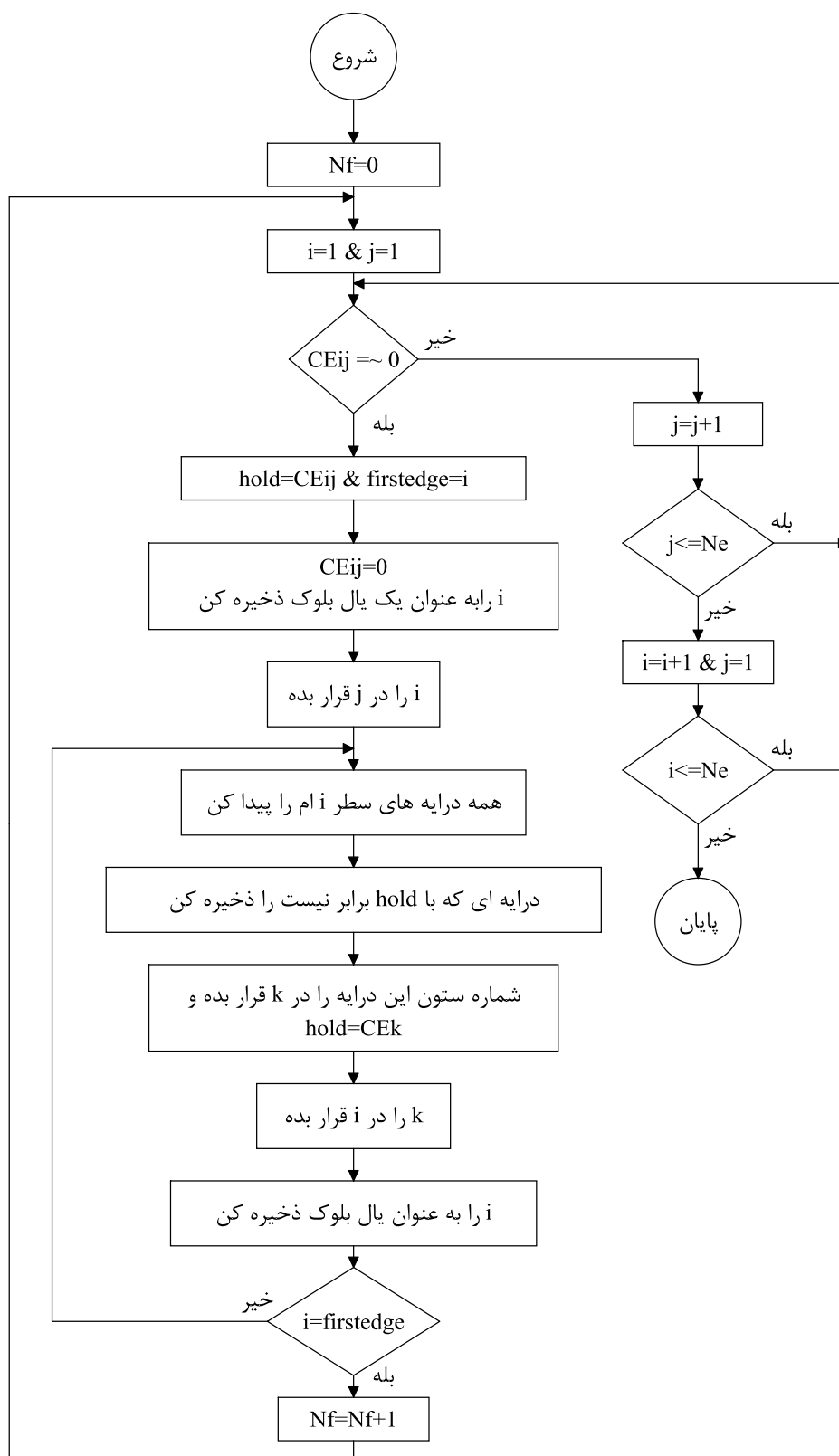
برای درک بهتر الگوریتم مبتنی بر ماتریس‌ها مراحل مختلف عملیات شناسایی بلوک بر روی یک

مثال توضیح داده می‌شود. مختصات نقاط ابتدا و انتهای تعدادی ناپیوستگی در جدول (۲-۴) آورده

شده است.

جدول ۲-۴: مختصات ناپیوستگی‌های

شماره ناپیوستگی	طول نقطه ابتدا (x_s)	عرض نقطه ابتدا (y_s)	طول نقطه انتها (x_e)	عرض نقطه انتها (y_e)
۱	۰	۰	۲	۶
۲	۲	۶	۵	۰
۳	۰	۰	۶	۱
۴	۰	۰	۳	۳
۵	۲	۶	۳	۳
۶	۳	۱	۵	۴



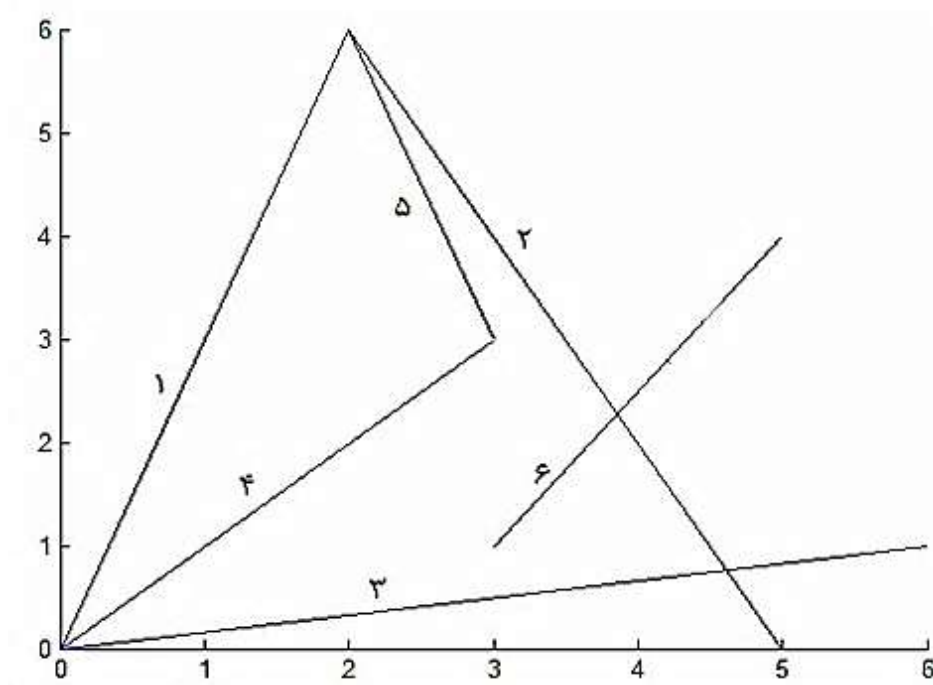
شکل ۴-۷: روندنمای مراحل شناسایی بلوک‌ها [۱۶]

مرحله اول ورود هندسه شکستگی‌ها در مدل است، برای این منظور مختصات ناپیوستگی‌ها در ماتریس Exp1 ذخیره می‌شود تا فرایند شناسایی بلوک‌ها بر روی آن انجام شود. این ماتریس برای مثال مذکور به صورت زیر است:

$$Exp1 = \begin{bmatrix} 0 & 0 & 2 & 6 \\ 2 & 6 & 5 & 0 \\ 0 & 0 & 6 & 1 \\ 0 & 0 & 3 & 3 \\ 2 & 6 & 3 & 3 \\ 3 & 1 & 5 & 4 \end{bmatrix}$$

در این ماتریس ستون اول و ستون دوم مختصات طول و عرض نقطه ابتدا و ستون دوم و سوم مختصات طول و عرض نقطه انتها است. و تعداد سطرها نیز برابر با تعداد ناپیوستگی‌ها است.

در شکل (۴-۸) نمای گرافیکی این مثال نشان داده شده است. اعداد روی هر خط شماره ناپیوستگی است.



شکل ۴-۸: نمای گرافیکی مثال مذکور

مرحله بعد محاسبه نقاط برخورد است. با استفاده از روابطی که ذکر شد مختصات نقاط برخورد این ناپیوستگی‌ها را می‌توان محاسبه نمود. ماتریس نقاط برخورد برای این مثال به صورت زیر است:

$$Pin = \begin{bmatrix} 2 & 6 \\ 0 & 0 \\ 0 & 0 \\ 2 & 6 \\ 4.61 & 0.77 \\ 2 & 6 \\ 3.86 & 2.29 \\ 0 & 0 \\ 3 & 3 \end{bmatrix}$$

در این ماتریس ستون اول مختصات طول نقطه و ستون دوم مختصات عرض نقاط برخورد است. به عنوان مثال نقطه (۶ و ۲) نقطه برخورد دو ناپیوستگی ۱ و ۵ است.

پس از محاسبه ماتریس نقاط برخورد باید ماتریس برخورد ناپیوستگی‌ها (C) مشخص شود. این ماتریس به صورت زیر می‌باشد:

$$C = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

در این ماتریس شماره سطرها و ستون‌ها نشان دهنده شماره ناپیوستگی و درایه‌های این ماتریس مشخص کننده وجود نقطه برخورد دو ناپیوستگی است. به عنوان مثال ناپیوستگی‌های ۱ و ۴ در یک نقطه برخورد دارند ولی ناپیوستگی‌های ۴ و ۲ دارای نقطه برخورد نیستند.

برای یافتن یال‌ها نقاط برخورد هر شکستگی باید با توجه به پارامتر t_{ij} مرتب شوند. این نقاط مرتب شده در ماتریس P ذخیره می‌شوند. ماتریس P برای مثال مذکور به صورت زیر است:

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 6 & 2 & 6 \\ 2 & 6 & 2 & 6 & 3.86 & 2.29 & 4.61 & 0.77 \\ 0 & 0 & 0 & 0 & 4.61 & 0.77 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 3 & 0 & 0 \\ 2 & 6 & 2 & 6 & 3 & 3 & 0 & 0 \\ 3.86 & 2.29 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

در این ماتریس مختصات نقاط برخورد برای هر ناپیوستگی بر روی یک سطر به ترتیب وجود دارد. به عنوان مثال ناپیوستگی ۴ در نقاط (۰ و ۰) و (۳ و ۳) با دیگر ناپیوستگی‌ها برخورد دارد. همانطور که در شکل (۸-۴) دیده می‌شود ناپیوستگی ۶ دارای ۱ نقطه برخورد است و درایه‌های بعد از این نقطه صفر در نظر گرفته می‌شوند. بنابراین باید برای تشکیل یال‌ها باید دقت داشت که از این نقطه فقط استفاده می‌شود. در واقع به دلیل این که یک نقطه برخورد بیشتر وجود ندارد یالی در این سطر تشکیل نمی‌شود.

برای شناسایی یال‌ها از ماتریس P استفاده می‌شود در این ماتریس از نقاط برخورد تکراری بر روی هر یال صرف نظر می‌شود و هر دو نقطه پشت سر هم در سطرها ی ماتریس P یک یال را تشکیل می‌دهند. به عنوان مثال در سطر اول ماتریس P نقاط (۰ و ۰) و (۶ و ۲) تشکیل یک یال را می‌دهند. ماتریس یال‌ها (Ed) به صورت زیر می‌باشد:

$$Ed = \begin{bmatrix} 0 & 0 & 2 & 6 \\ 2 & 6 & 3.86 & 2.29 \\ 3.86 & 2.29 & 4.61 & 0.77 \\ 0 & 0 & 4.61 & 0.77 \\ 0 & 0 & 3 & 3 \\ 2 & 6 & 3 & 3 \end{bmatrix}$$

در این ماتریس هر سطر نشان دهنده یک یال و ستون‌های آن مختصات نقاط ابتدا و انتهای یال است. پس از یافتن نقاط برخورد بین همه ناپیوستگی‌ها و یال‌های تشکیل شده توسط آن‌ها، هر یالی که می‌تواند یال یک بلوک یا چند ضلعی باشد، باید مشخص شود. برای این کار باید یال‌هایی که در

تشکیل بلوک‌ها مؤثر نیستند را حذف کرد. ماتریس C_p برای مرتب سازی یال‌ها برای این مثال به صورت زیر می‌باشد:

$$C_p = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

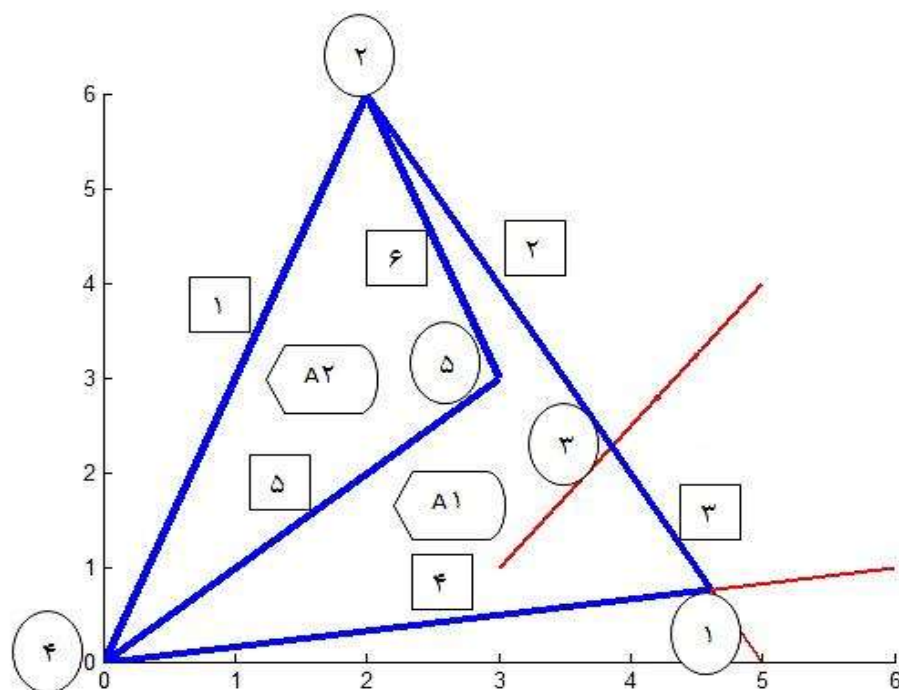
در این ماتریس هر سطری که مجموع ستون‌های آن کمتر از ۲ باشد باید حذف شود. در این مثال چون مجموع ستون‌های هیچ سطری کمتر از ۲ نیست پس یالی نیز حذف نمی‌شود. برای تشکیل ماتریس‌های اساسی نیاز به ماتریس رئوس است، این ماتریس برای مثال مذکور به صورت زیر می‌باشد:

$$V_x = \begin{bmatrix} 4.61 & 0.77 \\ 2 & 6 \\ 3.86 & 2.29 \\ 0 & 0 \\ 3 & 3 \end{bmatrix}$$

شکل (۴-۹) نشان دهنده یال‌ها و رئوس برای این مثال است.

پس از این مرحله ماتریس‌های اساسی اتصال رئوس (C_v) و اتصال یال‌ها (C_e) باید تعیین شوند. طبق روش توضیح داده شده ماتریس C_v برای مثال مذکور به صورت زیر می‌باشد:

$$C_v = \begin{bmatrix} 0 & 0 & 3 & 4 & 0 \\ 0 & 0 & 2 & 1 & 6 \\ 3 & 2 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 5 \\ 0 & 6 & 0 & 5 & 0 \end{bmatrix}$$



شکل ۴-۹: رئوس (○)، یال‌ها (□) و بلوک‌ها (⬡)

به عنوان مثال رأس سوم و اول یال سوم را تشکیل می‌دهند. با توجه به این که حداقل بلوک ایجاد شده توسط یال‌ها مثلث است (از نظر تعداد اضلاع) بنابراین درایه‌های غیر صفر روی هر سطر و ستون ماتریس اتصال رئوس باید ۲ باشد (یعنی هر رأس باید حداقل در ۲ یال مشترک باشد).

ماتریس اتصال یال‌ها (Ce) نیز با توجه به مراحل تشکیل آن که ذکر شد به صورت زیر می‌باشد:

$$Ce = \begin{bmatrix} 0 & 0 & 0 & 4 & 0 & 2 \\ 2 & 0 & 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 4 & 0 \\ 4 & 0 & 0 & 0 & 0 & 5 \\ 0 & 2 & 0 & 0 & 5 & 0 \end{bmatrix}$$

و در نهایت بلوک‌های تشکیل شده برای این مثال با توجه به مراحل شناسایی بلوک به صورت زیر می‌باشد:

$$\begin{cases} N_f = 1 : 4 - 1 - 3 - 2 \\ N_f = 2 : 2 - 5 - 4 \\ N_f = 3 : 3 - 1 - 4 - 5 - 2 \end{cases}$$

یعنی رئوس ۲، ۵ و ۴ بلوک ۲ را تشکیل می‌دهند. در این الگوریتم همیشه بلوکی که همه بلوک‌ها را در بر می‌گیرد شناسایی می‌شود (بلوک سوم در مثال ۱)، در حالی که این بلوک باید کنار گذاشته شود. شناسایی بلوکی که در بر گیرنده بقیه بلوک‌ها است توسط علامت مساحت آن است. به این صورت که همیشه این بلوک دارای مساحت مثبت و بقیه بلوک‌ها دارای مساحت منفی هستند. برای مثال مذکور مساحت بلوک‌های بدست آمده به صورت زیر می‌باشد:

$$\begin{cases} N_f = 1 : A_1 = 13/08 \\ N_f = 2 : A_2 = -6 \\ N_f = 3 : A_3 = -7/08 \end{cases}$$

همانطور که گفته شد بلوک اصلی دارای مساحت مثبت و برابر با مجموع مساحت دو بلوک دیگر است. برای محاسبات بعدی مساحت بلوک اصلی کنار گذاشته می‌شود و مساحت بلوک‌های دیگر مثبت در نظر گرفته می‌شود.

۴-۴ تهیه برنامه کامپیوتری

برنامه‌های کامپیوتری زیادی همانند SIMBLOCK، UDEC، برنامه تهیه شده توسط خیشوند و همکاران و ... برای تعیین شکل و حجم توده‌سنگ درزه‌دار در دو و سه بعد ارائه شده است که توسط الگوریتم‌های متفاوت، زبان‌های برنامه نویسی مختلف و نرم افزارهای مختلف تهیه شده‌اند. اما کدهای مرجع هیچ یک از این برنامه‌ها در اختیار عموم قرار نمی‌گیرد و هزینه تهیه این نرم افزارها نیز بالا است، بنابراین نیاز به تهیه نرم افزار همیشه وجود دارد. برنامه کامپیوتری که قرار است در این

پایان‌نامه تهیه شود در نرم افزار MATLAB کد نویسی می‌شود. در ادامه نحوه تهیه برنامه، داده‌های ورودی و خروجی برنامه کامپیوتری و اعتبار سنجی برنامه بررسی می‌شوند.

۴-۴-۱ نحوه تهیه برنامه کامپیوتری

برای تهیه برنامه کامپیوتری از زبان‌های برنامه نویسی و نرم افزارهایی مانند MATLAB استفاده می‌شود. تفاوت این زبان‌ها و نرم افزارها در توابع تعریف شده برای آن‌ها و دستورات متفاوت آن‌ها است. از زبان‌های برنامه نویسی می‌توان به Basic، C، Pascal، FORTRAN و ... اشاره کرد. برنامه تهیه شده در این پایان‌نامه در نرم افزار MATLAB تهیه شده است. برای تهیه برنامه از دستورات و توابع مختلف که در جدول (۴-۳) آورده شده است، استفاده می‌شود. در این جدول همچنین شرح این دستورات و توابع قابل مشاهده است.

جدول ۴-۳: لیست دستورات و توابع به کار رفته در برنامه تهیه شده

دستور یا تابع	شرح
For	با آزمایش کردن یک شرط یکسری دستورات را به صورت تکراری اجرا می‌کند.
If	با بررسی شرطی دستورات خواسته شده را اجرا می‌کند.
Line	تابعی است که برای ترسیم خطوط به کار می‌رود.
While	تا زمانی که شرطی برقرار باشد یکسری دستورات را به صورت تکراری انجام می‌دهد.
Sort	دستوری برای مرتب کردن اعداد در ماتریس‌ها است.
Ceil	برای گرد کردن اعداد به بالا از این دستور استفاده می‌شود.
Floor	برای گرد کردن اعداد به پایین از این دستور استفاده می‌شود.

ادامه جدول	
Abs	بیان کننده تابع قدر مطلق است.
Size	دستوری است که ابعاد یک ماتریس را نشان می‌دهد.
Sum	بیان کننده عملیات جمع است.
Nor	تابعی برای محاسبه بردار نرمال یک بردار است.
Dot	بیان کننده عملیات ضرب داخلی است.
Input	تابعی برای ورود داده‌ها است.
Disp	تابعی برای نشان دادن داده‌ها و یا عبارات متنی در خروجی است.

کد تهیه شده در پیوست گزارش قرار داده شده است و قسمتی از آن که مربوط به تعیین مساحت بلوک‌های توده‌سنگ است در ادامه مشاهده می‌شود:

```
[m,n]=size(CE);
f=1;
g=1;
for i=1:m
    for j=1:n
        if CE(i,j)~=0
            holde(f,g)=CE(i,j);
            firsted=i;
            CE(i,j)=0;
            l=j;
        while l~=firsted
            for k=1:n
```

```

if l~=firsted && CE(l,k)~=holde(f,g) && CE(l,k)~=0
    g=g+1;
    holde(f,g)=CE(l,k);
    CE(l,k)=0;
    l=k;
end
end
end
g=1;
f=f+1;
end
end
end
end

```

این قسمت از کد برای تعیین رئوس بلوک‌های توده‌سنگ با استفاده از ماتریس اتصال یال‌ها است. این قسمت شامل سه حلقه for و دو دستور شرطی if و یک حلقه while است.

۲-۴-۴ ورودی برنامه کامپیوتری

هر برنامه کامپیوتری که تهیه می‌شود دارای یک یا چند ورودی با انواع مختلف است. ورودی برنامه‌ها می‌تواند تصاویر گرافیکی، فایل‌های صوتی، فایل‌های ویدئویی، فایل‌های متنی و یا مجموعی از این نوع داده‌ها باشد. داده‌های ورودی به برنامه‌های کامپیوتری تحت دستورات قابل فهم برای کامپیوتر پردازش می‌شود. ورودی برنامه کامپیوتری تهیه شده به صورت زیر می‌باشد:

۱- ماتریسی شامل مختصات ناپیوستگی‌های منفرد

این ماتریس دارای چهار ستون و به تعداد ناپیوستگی‌ها سطر می‌باشد. ستون اول این ماتریس مختصات طول نقطه ابتدای ناپیوستگی و ستون دوم مختصات عرض نقطه ابتدای

ناپیوستگی است. ستون‌های سوم و چهارم نیز به ترتیب مختصات طول و عرض انتهایی ناپیوستگی است.

۲- وارد کردن دستی محدوده مورد مطالعه

یک محدوده اختیاری برای بررسی ناپیوستگی‌ها درون آن می‌توان تعریف کرد. برای این کار طول و عرض محدوده را از طریق صفحه کلید و هنگام درخواست برنامه می‌توان وارد کرد. طول و عرض محدوده در ماتریسی با ابعاد 1×2 به صورت $[X \ Y]$ است.

۳- وارد کردن دستی شیب و فاصله‌داری دسته درزه‌ها

دسته درزه‌ها نیز با وارد کردن مشخصات آن‌ها از طریق صفحه کلید معرفی می‌شوند. در برنامه تهیه شده تعداد دسته درزه‌ها محدودیتی ندارد و هر تعداد دسته درزه را می‌توان به مدل اضافه کرد. برای هر دسته درزه شیب و فاصله داری درزه‌ها باید به ترتیب رعایت شود به عنوان مثال برای وارد کردن سه دسته درزه با فاصله‌داری به ترتیب ۵، ۲ و $1/5$ و دارای شیب به ترتیب ۲۵، ۹۰ و ۱۵۰ درجه باید ماتریس‌های $[5 \ 2 \ 1/5]$ برای فاصله‌داری و $[25 \ 90 \ 150]$ برای شیب دسته درزه‌ها را از طریق صفحه کلید و هنگام درخواست برنامه به مدل اضافه کرد.

داده‌های ورودی این برنامه از نوع داده‌های متنی و در واقع اعداد هستند.

۴-۳ خروجی برنامه کامپیوتری

منظور از خروجی نتایج بدست آمده در اثر پردازش داده‌های ورودی است. خروجی برنامه‌های کامپیوتری داده‌های قابل فهم برای انسان هستند. این داده‌ها شامل تصاویر، فایل‌های متنی، فایل‌های

صوتی، فایل‌های ویدئویی، نمودارها و ... می‌باشند. یک برنامه می‌تواند یک یا چند نوع از این داده‌ها را به عنوان خروجی داشته باشد. خروجی برنامه تهیه شده به صورت زیر می‌باشد:

۱- شکل شبکه ناپیوستگی‌های مدل به صورت تصویر

تصویر ناپیوستگی‌ها و بلوک‌های تشکیل شده در مدل به صورت تصویر در انتهای اجرای برنامه نمایش داده می‌شود. در این برنامه اضلاع بلوک‌ها با رنگ و ضخامت خط متفاوت با دیگر خطوط نمایش داده می‌شود. و از این طریق می‌توان شناسایی همه بلوک‌های مدل را کنترل نمود.

۲- مختصات رئوس بلوک‌های تشکیل شده در مدل در قالب ماتریس

این ماتریس شامل مختصات طول و عرض همه رئوس بلوک‌های تشکیل شده در اثر برخورد ناپیوستگی‌ها است.

۳- مساحت بلوک‌های شناسایی شده در مدل در قالب ماتریس

ماتریسی شامل شماره، مساحت و اضلاع تشکیل دهنده هر بلوک در انتهای اجرای برنامه به نمایش در می‌آید. ستون اول این ماتریس شماره بلوک، ستون دوم مساحت آن و ستون‌های بعدی شماره اضلاع تشکیل دهنده آن را نشان می‌دهند.

۴- توزیع مساحت بلوک‌های بر جای توده‌سنگ به صورت نمودار و در قالب ماتریس

نمودار توزیع مساحت بلوک‌های توده‌سنگ نیز با اجرای دستورات مخصوص نشان داده می‌شود.

برای نمونه تعدادی مثال در ادامه حل خواهد شد و داده‌های ورودی و خروجی هر یک نشان داده می‌شود. در ابتدا مثال ۱ که برای درک بهتر الگوریتم ماتریسی حل شد توسط برنامه نیز حل می‌شود.

ورودی و خروجی این مثال در شکل (۴-۱۰) و خروجی تصویری آن در شکل (۴-۱۱) نشان داده شده است.

If you have not each type of inputing data press 0 (zero) when ask you:

Input matrices of single discountinuties coordinate:expl

Input dimension of master block [Xm , Ym], if exist:0

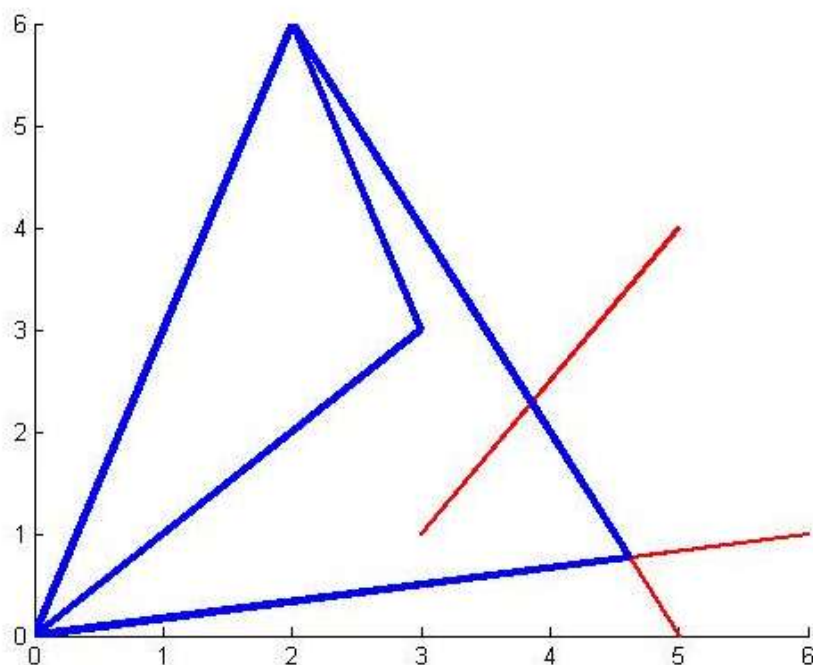
Input angle of joint sets with horizontal line (X) [θ1 θ2 ...]:0

Input Spacing of joint sets (S) [S1 S2 ...]:0

Areas of Blocks in the model:

Block N	Block Area		Block edges			
1.0000	13.0769	4.0000	1.0000	3.0000	2.0000	0
2.0000	-6.0000	2.0000	5.0000	4.0000	0	0
3.0000	-7.0769	3.0000	1.0000	4.0000	5.0000	2.0000

شکل ۴-۱۰: خروجی برنامه برای مثال ۱



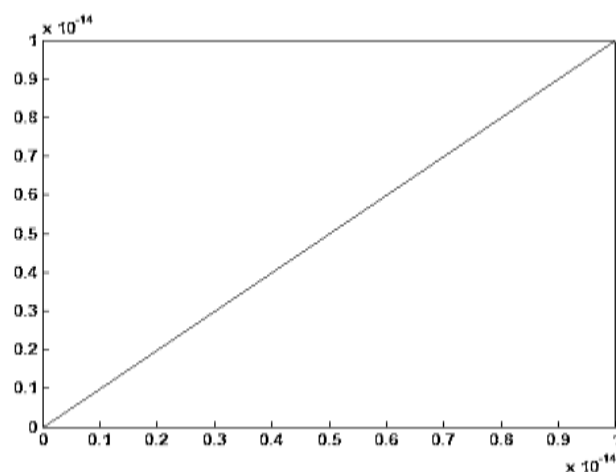
شکل ۴-۱۱: خروجی تصویری برنامه برای مثال ۱

همانطور که در شکل (۴-۱۰) مشاهده می‌شود نتایج برنامه با نتایج حاصل از حل الگوریتم مطابقت دارد.

۴-۴-۴ مثال ۲: سنجش میزان دقت برنامه

منظور از میزان دقت برنامه کوچکترین بلوکی قابل شناسایی در مدل است. برای سنجش میزان دقت برنامه کامپیوتری تهیه شده، مثال ۲ که در شکل (۴-۱۲) دیده می‌شود توسط این برنامه حل می‌شود. ماتریس Exp2 ورودی برنامه به صورت زیر می‌باشد:

$$\text{Exp2} = \begin{bmatrix} 0 & 0 & 10^{-14} & 0 \\ 0 & 0 & 0 & 10^{-14} \\ 10^{-14} & 0 & 10^{-14} & 10^{-14} \\ 0 & 10^{-14} & 10^{-14} & 10^{-14} \\ 0 & 0 & 10^{-14} & 10^{-14} \end{bmatrix}$$



شکل ۴-۱۲: مثال ۲ بلوکی با ابعاد $(10^{-14} * 10^{-14})$

نتایج بدست آمده برای شناسایی بلوک‌ها برای مثال ۲ به صورت زیر است:

$$\begin{cases} N_f = 1 : A_1 = -0.5 * 10^{-28} \\ N_f = 2 : A_2 = 1 * 10^{-28} \\ N_f = 3 : A_3 = -0.5 * 10^{-28} \end{cases}$$

به عبارت دیگر این برنامه قادر به شناسایی بلوک‌هایی تا مساحت 10^{-28} واحد می‌باشد.

۴-۴-۵ مثال ۳ و ۴: اعتبار سنجی برنامه تهیه شده

الگوریتم مبتنی بر ماتریس‌ها یک الگوریتم ابتکاری می‌باشد و برنامه تهیه شده نیز بر اساس این الگوریتم است. از حل تعداد زیادی مثال توسط این برنامه نتایج صحیحی بدست آمده است. به عنوان مثال، مثال ۳ به صورت دستی حل و نتایج آن با نتایج حاصل از حل برنامه مقایسه می‌شود و مثال ۴ نیز با برنامه تهیه شده حل و جواب آن با نتایج ارائه شده در منبع شماره ۱۶ مقایسه می‌شود.

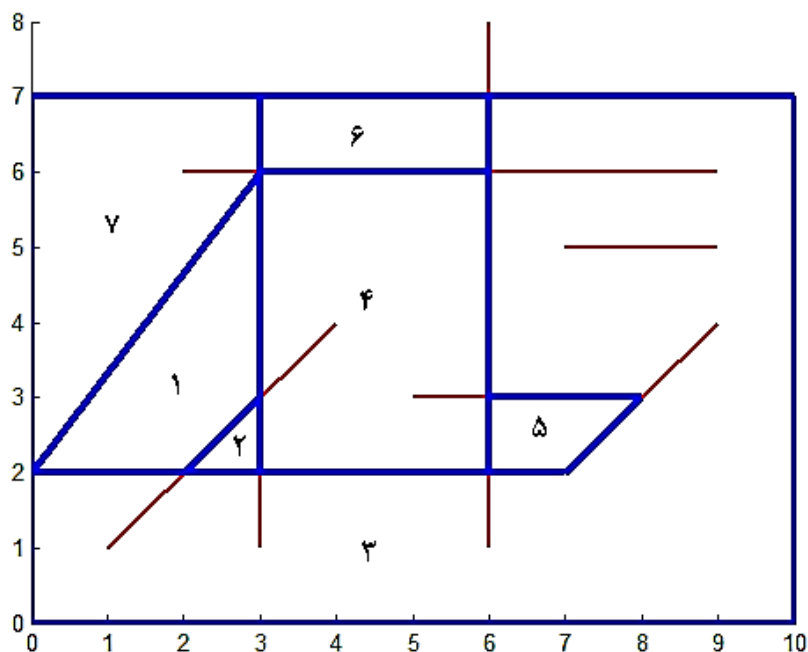
۴-۴-۵-۱ مثال ۳

مثال ۳ شامل تعداد ۱۰ ناپیوستگی است که نقاط ابتدا و انتهای آن‌ها در جدول (۴-۴) آورده شده است. قرار است بلوک‌هایی که در محدوده 10×7 واحد از این سیستم ناپیوستگی تشکیل می‌شوند را تعیین کرد. در واقع ابعاد بلوک اصلی برای این مثال ۱۰ واحد در طول و ۷ واحد در عرض می‌باشد.

جدول ۴-۴: مختصات ناپیوستگی‌های مثال ۳

شماره ناپیوستگی	طول نقطه ابتدا (x_s)	عرض نقطه ابتدا (y_s)	طول نقطه انتها (x_e)	عرض نقطه انتها (y_e)
۱	۱	۱	۴	۴
۲	۰	۲	۷	۲
۳	۷	۲	۹	۴
۴	۸	۳	۵	۳
۵	۹	۵	۷	۵
۶	۹	۶	۲	۶
۷	۳	۶	۰	۲
۸	۳	۷	۳	۱
۹	۶	۸	۶	۷
۱۰	۶	۷	۶	۱

شکل (۴-۱۳) بلوک‌های ایجاد شده توسط این ناپیوستگی‌ها و محدوده یا بلوک اصلی را نشان می‌دهد. با استفاده از روابط ریاضی ساده مساحت هر یک از بلوک‌ها محاسبه می‌شود. جدول (۴-۵) بلوک‌های تشکیل شده و محاسبه مساحت آن‌ها به روش دستی را نشان می‌دهد.



شکل ۴-۱۳: نمای گرافیکی ناپیوستگی‌های مثال ۳

جدول ۴-۵: محاسبه مساحت بلوک‌های مثال ۳ به صورت دستی

شماره بلوک	نوع بلوک (چند ضلعی)	مساحت (واحد)
۱	دوزنقه	$S_1 + S_2 = \frac{3 \times 4}{2} = 6$ $S_2 = \frac{1 \times 1}{2} = 0.5$ $S_1 = 6 - 0.5 = 5.5$
۲	مثلث	$S_2 = \frac{1 \times 1}{2} = 0.5$
۳	هشت ضلعی	$S_3 = 40 - S_3 = 40 - \frac{1+2}{2} \times 1 = 38.5$
۴	مستطیل	$S_4 = 3 \times 4 = 12$
۵	دوزنقه	$S_5 = \frac{1+2}{2} \times 1 = 1.5$
۶	مستطیل	$S_6 = 1 \times 3 = 3$
۷	دوزنقه	$S_7 = \frac{1+5}{2} \times 3 = 9$

برای حل مثال ۳ با برنامه تهیه شده نقاط ابتدا و انتها را در ماتریس Exp3 ذخیره می‌شود. این ماتریس و ابعاد بلوک اصلی (10×7) داده‌های ورودی برنامه هستند. ماتریس Exp3 به صورت زیر است:

$$\text{Exp3} = \begin{bmatrix} 1 & 1 & 4 & 4 \\ 0 & 2 & 7 & 2 \\ 7 & 2 & 9 & 4 \\ 8 & 3 & 5 & 3 \\ 9 & 5 & 7 & 5 \\ 9 & 6 & 2 & 6 \\ 3 & 6 & 0 & 2 \\ 3 & 7 & 3 & 1 \\ 6 & 8 & 6 & 7 \\ 6 & 7 & 6 & 1 \end{bmatrix}$$

شکل (۴-۱۴) خروجی برنامه را برای مثال ۳ نشان می‌دهد.

Input matrices of single discountinuties coordinate:Exp3

Input dimension of master block [Xm , Ym], if exist:[10 7]

Input angle of joint sets with horizontal line (X) [@1 @2 ...]:0

Input Spacing of joint sets (S) [S1 S2 ...]:0

Areas of Blocks in the model:

Block N	Block Area	Block edges									
1	-5.5000	1	10	9	2						
2	-0.5000	2	4	1							
3	-38.5000	1	4	5	3	6	7	8	12	16	14
4	-12.0000	4	2	9	8	7	5				
5	-1.5000	5	7	6	3						
6	-3.0000	9	11	12	8						
7	-9.0000	10	15	11	9						
8	70.0000	14	16	12	11	15	10	13			

شکل ۴-۱۴: نحوه ورود اطلاعات و خروجی برنامه برای مثال ۳

همانطور که در شکل مشاهده می‌شود ابتدا ماتریس Exp3 به عنوان ناپیوستگی‌های منفرد و ابعاد بلوک اصلی ورودی برنامه هستند و مساحت بلوک‌های ۱ تا ۷ که با علامت منفی مشخص هستند به

عنوان بلوک‌های شناسایی شده هستند که مساحت بلوک اصلی حذف و مساحت دیگر بلوک‌ها مثبت در نظر گرفته می‌شود. جدول (۴-۶) مقایسه بین نتایج بدست آمده بوسیله برنامه و حل دستی را نشان می‌دهد. همانطور که در این جدول مشاهده می‌شود نتایج بدست آمده از حل برنامه با نتایج حل دستی کاملاً مطابقت دارد.

جدول ۴-۶: مقایسه نتایج حل دستی و حل توسط برنامه مثال ۳

شماره بلوک	مساحت بلوک‌ها	
	نتایج حل دستی	نتایج حل توسط برنامه
۱	۵/۵	۵/۵
۲	۰/۵	۰/۵
۳	۳۸/۵	۳۸/۵
۴	۱۲	۱۲
۵	۱/۵	۱/۵
۶	۳	۳
۷	۹	۹

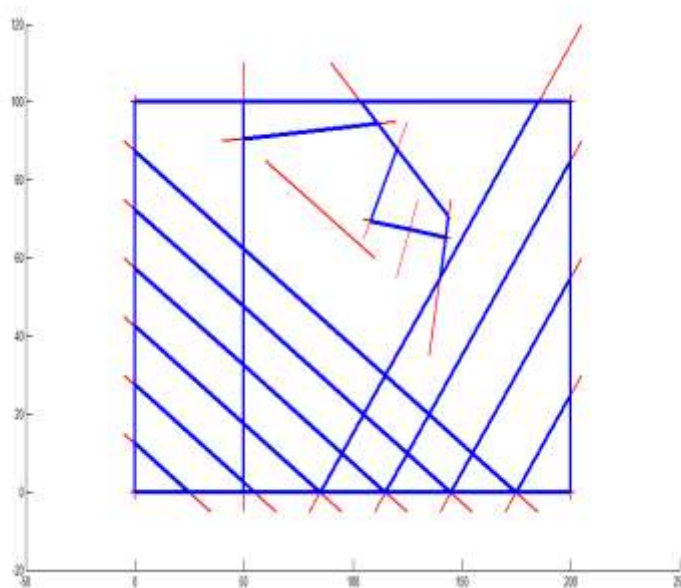
۴-۴-۵-۲ مثال ۴

مثال ۴ نیز برای بررسی صحت برنامه حل می‌شود. نتایج بدست آمده برای شناسایی بلوک‌ها از این برنامه و نتایج ارائه شده در منبع شماره ۱۶ با هم مقایسه خواهد شد. این مثال شامل ۲۲ ناپیوستگی منفرد است. ابعاد محدوده نیز در قالب چهار ناپیوستگی به مدل اضافه شده است.

ماتریس A در برگزیده ناپیوستگی‌های مثال ۴ به صورت زیر می‌باشد:

$$A = \begin{bmatrix} -2 & 0 & 202 & 0 \\ -2 & 100 & 202 & 100 \\ 0 & -2 & 0 & 102 \\ 200 & -2 & 200 & 102 \\ -5 & 15 & 35 & -5 \\ -5 & 30 & 65 & -5 \\ -5 & 45 & 95 & -5 \\ -5 & 60 & 125 & -5 \\ -5 & 75 & 155 & -5 \\ -5 & 90 & 185 & -5 \\ 205 & 30 & 170 & -5 \\ 205 & 60 & 140 & -5 \\ 205 & 90 & 110 & -5 \\ 205 & 120 & 80 & -5 \\ 135 & 35 & 145 & 75 \\ 145 & 70 & 90 & 110 \\ 125 & 95 & 105 & 65 \\ 105 & 70 & 145 & 65 \\ 130 & 75 & 120 & 55 \\ 120 & 95 & 40 & 90 \\ 50 & -5 & 50 & 110 \\ 60 & 85 & 110 & 60 \end{bmatrix}$$

هندسه ناپیوستگی‌های مثال ۴ که توسط برنامه تهیه شده ترسیم شده است، در شکل (۴-۱۵) نشان داده شده است. شکل (۴-۱۶) نتایج ارائه شده برای مثال مذکور در منبع شماره ۱۶ را نشان می‌دهد.



شکل ۴-۱۵: شکل مثال ۴ برای اعتبار سنجی برنامه

Block no.	Area (m ²)	Edge no.
1	20000	1,2,3,4,5,6,7,8,20,21,22,23,12,11,10, 9,19,18,17,16,15,14,13,1
2	-156.25	1,13,24,1,
3	-593.75	2,24,14,26,60,2
4	-6.25	3,60,25,3,
5	-300	4,25,61,27,4,
6	-150	5,47,29,5,
7	-150	6,44,32,6,
8	-150	7,42,36,7,
9	-312.5	8,41,20,8,
10	-1250	9,66,65,40,19,9,
11	-436.533273381294	10,54,59,66,10,
12	-1453.605769230770	11,51,52,53,56,55,54,11,
13	-2137.5	12,23,46,38,50,51,12,
14	-750	15,28,61,26,15,
15	-750	16,31,62,28,16,
16	-750	17,35,63,31,17,
17	-750	18,40,64,35,18,
18	-1050	21,41,36,43,21,
19	-1650	22,43,37,46,22,
20	-600	27,62,30,47,27,
21	-300	29,48,33,44,29,
22	-750	30,63,34,48,30,
23	-300	32,45,37,42,32,
24	-300	33,49,38,45,33,
25	-900	34,64,39,49,34,
26	-3634.477263700670	39,65,59,55,57, 58,52,50,39
27	-419.133693687265	53,58,57,56,53,

شکل ۴-۱۶: نتایج ارائه شده برای مثال ۴ [۱۶]

در شکل (۴-۱۷) خروجی برنامه تهیه شده نشان داده شده است که فقط ترتیب یافتن بلوک‌ها در این برنامه با برنامه ارائه شده در منبع متفاوت است اما نتایج بدست آمده با نتایج ارائه شده در شکل (۴-۱۶) مطابقت دارد. در این شکل مساحت بلوک‌ها نشان داده شده است.

Areas of Blocks in the model:

Block N	Block Area	Block edges																			
	1.0e+004 *																				
01	2.0000	03	05	04	21	23	26	30	02	17	18	19	7	8	9	10	6	16	15	14	13
02	-0.0156	01	11	03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03	-0.0594	03	11	12	20	05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04	-0.0006	05	20	04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05	-0.0300	04	20	22	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
06	-0.0150	21	24	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07	-0.0150	23	27	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
08	-0.0150	26	31	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
09	-0.0313	30	17	02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	-0.1250	10	42	34	16	06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	-0.0437	09	39	42	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	-0.1454	08	35	37	36	38	39	09	0	0	0	0	0	0	0	0	0	0	0	0	0
13	-0.2138	07	19	32	33	35	08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	-0.0750	13	22	20	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	-0.0750	14	25	22	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	-0.0750	15	29	25	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	-0.0750	16	34	29	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	-0.1050	17	30	31	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	-0.1650	18	31	32	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	-0.0600	22	25	24	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	-0.0750	25	29	28	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	-0.0300	24	28	27	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	-0.0900	29	34	33	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	-0.0300	28	33	32	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	-0.0300	27	32	31	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	-0.3634	34	42	39	38	40	41	37	35	33	0	0	0	0	0	0	0	0	0	0	0
27	-0.0419	37	41	40	38	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

شکل ۴-۱۷: خروجی برنامه تهیه شده برای مثال ۴

در این شکل اعداد مربوط به مساحت بلوک‌ها به صورت نماد علمی نشان داده شده‌اند و شماره پاره‌خط‌های تشکیل دهنده هر بلوک نیز در شکل مشخص است. البته شماره پاره‌خط‌های تشکیل دهنده هر بلوک با برنامه تهیه شده توسط خیشوند و همکاران متفاوت است. همه اطلاعات هندسی توده‌سنگ از قبیل مختصات رئوس بلوک‌ها، توزیع تجمی مساحت بلوک‌ها و ... قابل دسترسی خواهند بود.

۴-۵ مثال‌های کاربردی در زمینه مهندسی معدن

در این بخش تعدادی مثال کاربردی در زمینه کاربردهای تعیین هندسه بلوک‌های توده‌سنگ توسط برنامه تهیه شده حل می‌شود. این مثال‌ها در کاربردهای مختلف از جمله بلوک‌دهی سنگ‌های ساختمانی، آنالیز آتشفباری، پایداری بلوک‌های سنگی و پیش‌بینی رقیق شدگی ماده معدنی در روش‌های استخراج تخریبی می‌باشد. در این مثال‌ها مشخصات بلوک‌های توده‌سنگ توسط برنامه قابل دسترس خواهد بود.

۴-۵-۱ مثال ۵: بلوک دهی در معدن سنگ ساختمانی

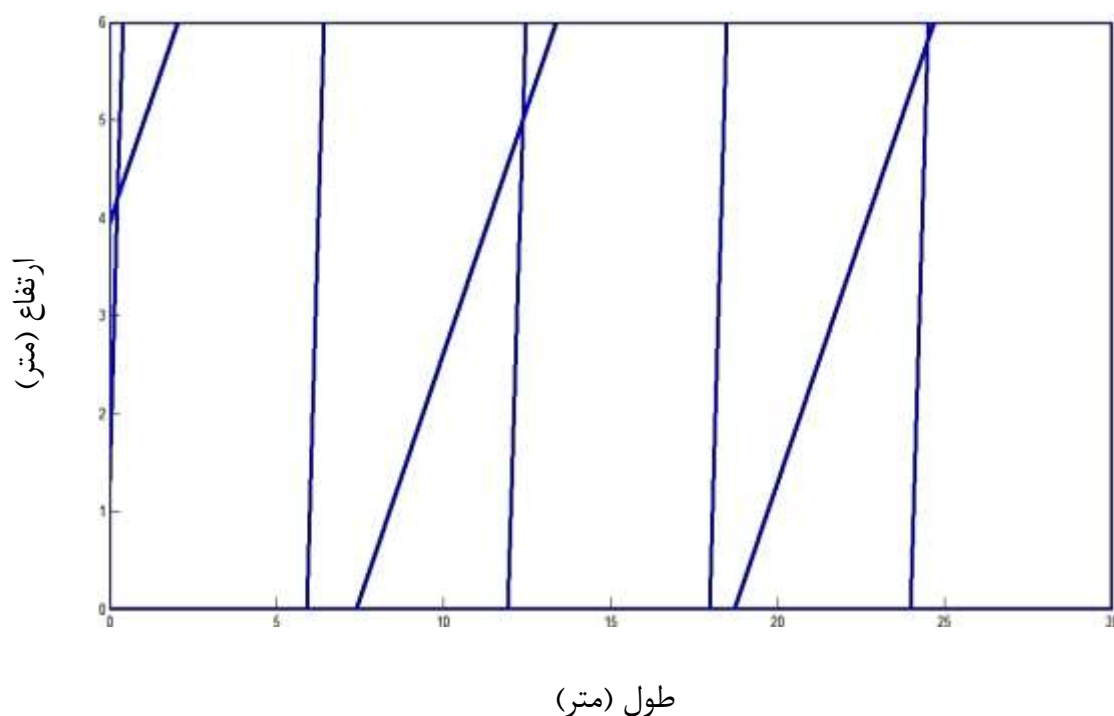
فرض می‌شود جبهه کار یک معدن سنگ ساختمانی با مشخصات زیر وجود دارد، بلوک دهی این معدن در این جبهه کار بررسی می‌شود.

- ابعاد جبهه کار: ارتفاع ۶ متر و طول ۳۰ متر
- مشخصات دسته درزه‌ها: دسته درزه اول: زاویه ۸۵ درجه با افق (شیب) و فاصله داری ۶ متر و دسته درزه دوم: شیب ۴۵ درجه و فاصله داری ۸ متر

ورودی برنامه برای این مثال به صورت دستی و وارد کردن مشخصات دسته درزه‌ها و ابعاد جبهه کار هنگام اجرای برنامه از طریق صفحه کلید است. نمای این جبهه کار در شکل (۴-۱۸) نشان داده شده است.

با توجه به این که بلوک‌های سالم در معادن سنگ ساختمانی ایران حدوداً دارای ابعاد $3 \times 2 \times 1.5$ هستند، با فرضیات زیر درصد بلوک‌های مختلف در این جبهه کار تعیین می‌شود:

الف) ابعاد بلوک مناسب در این معدن سنگ $3 \times 2 \times 1.5$ باشد.



شکل ۴-۱۸: نمای جبهه کار معدن سنگ

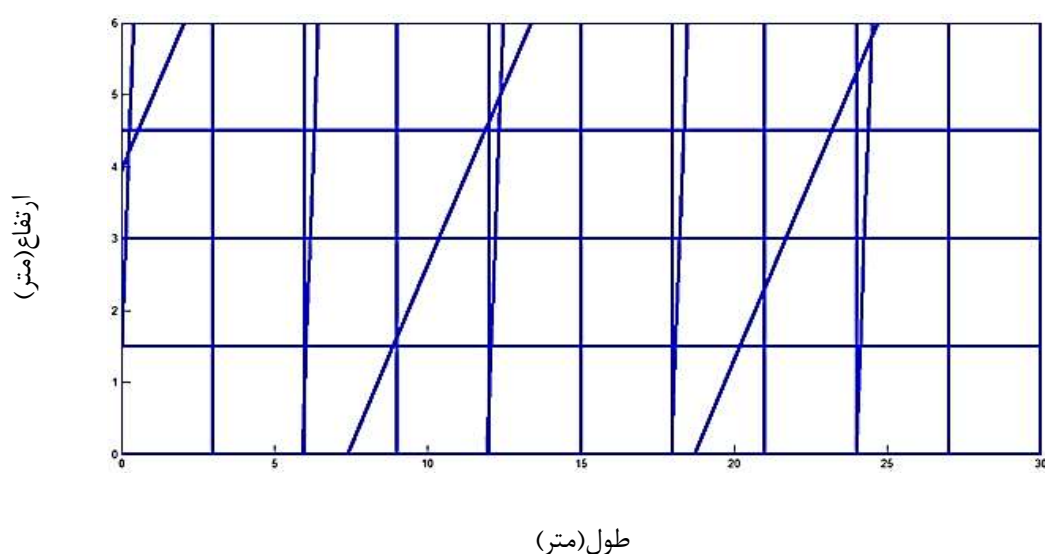
ب) ابعاد بلوک مناسب در دو بعد موجود $3 \times 1/5$ با حداقل مساحت $4/2$ متر مربع باشد. (یعنی عرض برش ۲ متر باشد).

ج) این سینه کار با ابعاد $3 \times 1/5$ متر برش داده شود.

د) بلوک‌های دارای مساحت $4/2$ متر مربع و بزرگتر بلوک سالم (با توجه به ابعاد بلوک مناسب)، بلوک‌هایی که مساحتی بین ۱ و $4/2$ متر مربع دارند، بلوک‌های قابل حمل به سنگبری و بلوک‌های کمتر از ۱ متر مربع باطله باشند.

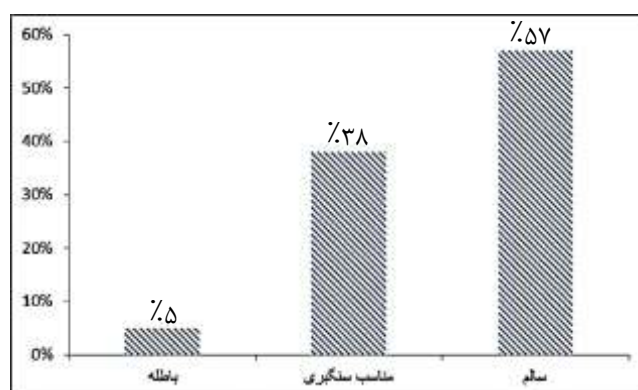
برای این جبهه کار با استفاده از نتایج برنامه کامپیوتری تهیه شده ۵۷٪ از بلوک‌ها بلوک‌های سالم هستند و ۳۸٪ آن‌ها بلوک‌های قابل حمل به سنگبری حمل و ۵٪ آن‌ها نیز باطله می‌باشند. همانطور

که در شکل (۴-۱۹) مشاهده می‌شود با داشتن میزان مختلف بلوک‌ها می‌توان برنامه‌ریزی مفیدتری برای معدن داشت و همچنین با برداشت درزه‌ها از چند جبهه کار می‌توان ذخیره معدن را با توجه به اطلاعات بلوک‌ها پیش بینی نمود. البته باید توجه داشت که به دلیل شکستن بلوک‌ها هنگام استخراج، روش‌های تبدیل بلوک‌های بزرگتر به بلوک‌های کوچکتر و نسبت‌های فوق دقیق نیست اما نتایج با دقت قابل قبولی می‌تواند مورد استفاده قرار گیرد. شکل (۴-۱۹) جبهه کار سنگ همراه با اعمال برش $3 \times 1/5$ توسط برنامه را نشان می‌دهد.



شکل ۴-۱۹: نمای جبهه کار معدن با اعمال برش

نمودار درصد انواع بلوک‌های موجود در این جبهه کار معدن در شکل (۴-۲۰) نشان داده شده است.



شکل ۴-۲۰: نمودار درصد بلوک‌های باطله، قابل حمل به سنگبری و سالم

۴-۵-۲ مثال ۶: بررسی انرژی خردایش در توده‌سنگ

آنالیز خردایش توده سنگ با آتشیاری نیاز به بررسی کامل دو مسأله اساسی، زمین شناسی منطقه و طراحی آتشیاری دارد. زمین شناسی منطقه شامل مشخصات شبکه ناپیوستگی و زمینه سنگ و طراحی آتشیاری نیز شامل انتخاب پارامترهای آتشیاری است [۱۹]. برای مشخص کردن میزان انرژی مصرف شده جهت خردایش هنگام آتشیاری و بهینه کردن پارامترهای آتشیاری نیاز به حجم برجای بلوک‌های توده سنگ است [۳۰]. برای دستیابی به حجم برجای بلوک‌های توده سنگ می‌توان از برنامه کامپیوتری تهیه شده استفاده کرد. فرض می‌شود جبهه کار یک معدن سنگ آهن با مشخصات زیر وجود دارد، توزیع مساحت بلوک‌های توده سنگ در این جبهه کار به عنوان داده‌های لازم برای ارزیابی خردایش و انرژی خردایش محاسبه می‌شود.

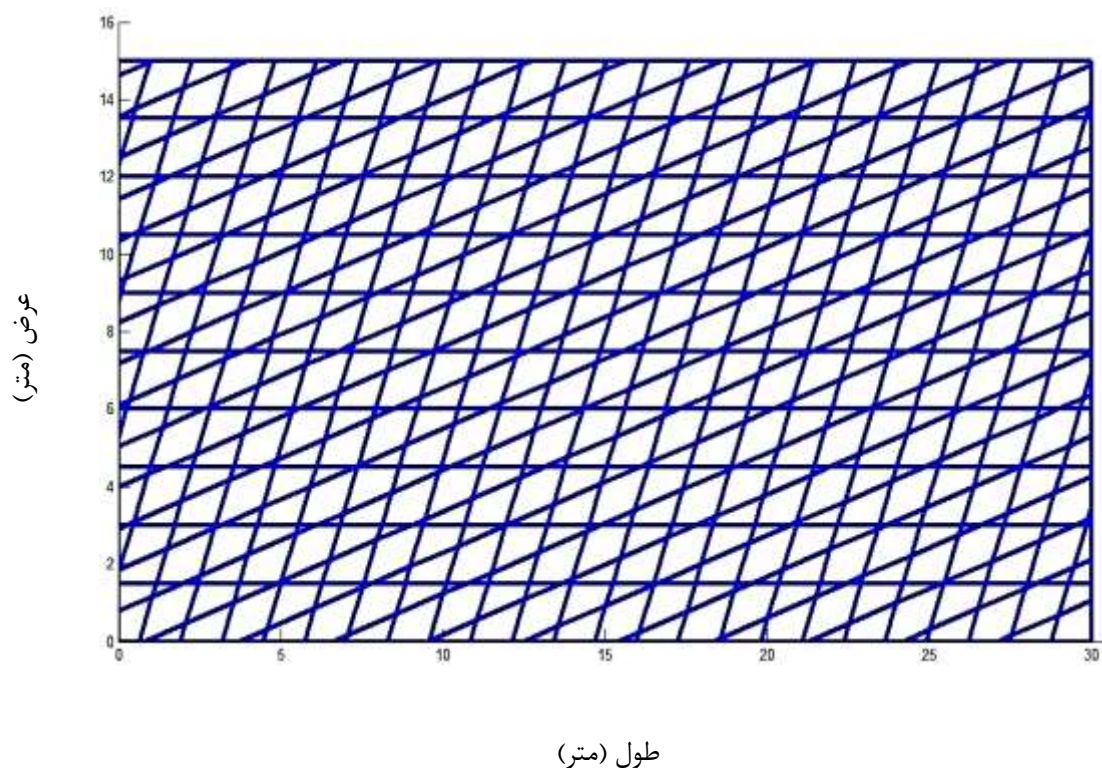
• ابعاد جبهه کار: ارتفاع ۱۵ متر و طول ۳۰ متر

• مشخصات دسته درزه‌ها: دسته درزه اول: شیب صفر درجه و فاصله داری ۱/۵ متر، دسته

درزه دوم: شیب ۲۰ درجه و فاصله داری ۱ متر، دسته درزه سوم: شیب ۷۰ و فاصله داری

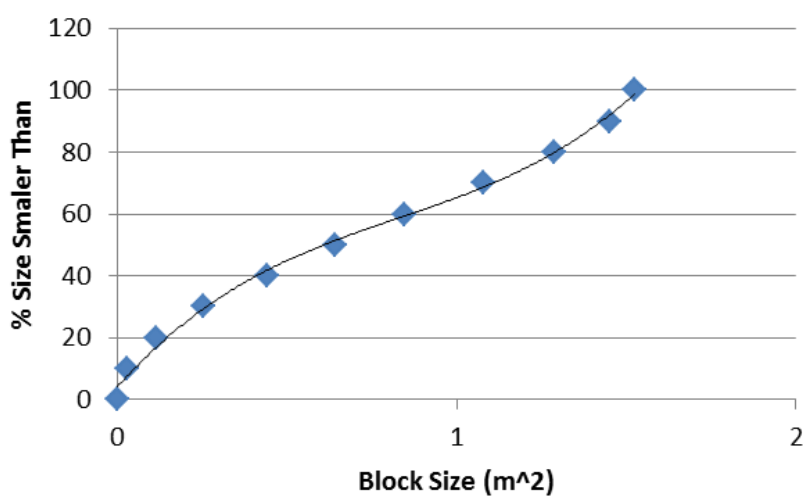
۱/۲ متر

نمای جبهه کار این مثال که خروجی برنامه کامپیوتری است، در شکل (۴-۲۱) نشان داده شده است.



شکل ۴-۲۱: جبهه کار به همراه درزه‌های آن در مثال ۶

توزیع مساحت بلوک‌های این توده سنگ در شکل (۴-۲۲) نشان داده شده است.



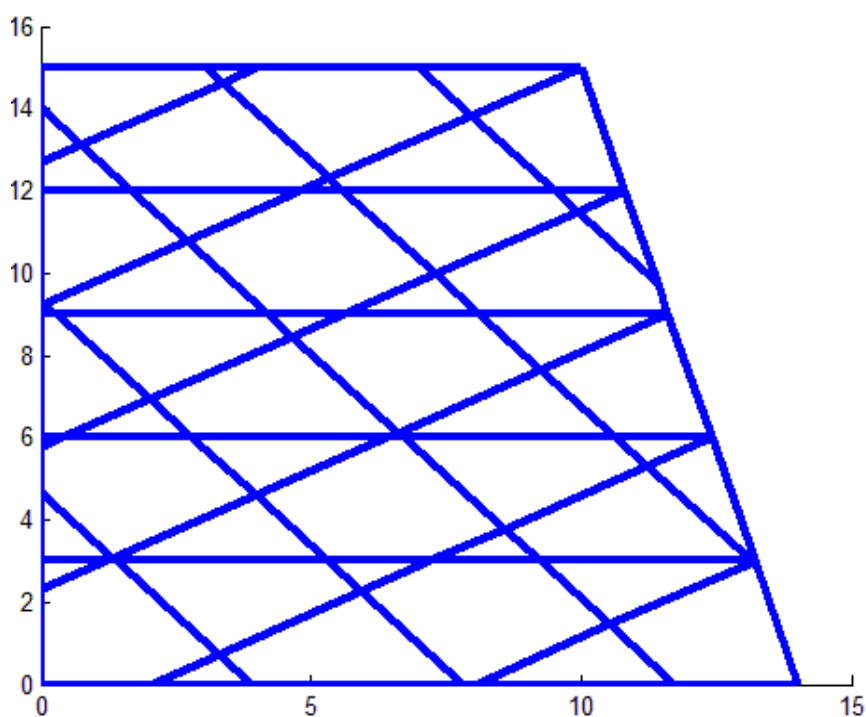
شکل ۴-۲۲: نمودار توزیع تجمعی اندازه بلوک‌های توده‌سنگ

۴-۵-۳ مثال ۷: طراحی پارامترهای آتشیاری

همانطور که در شکل (۴-۲۳) مشاهده می‌شود با توجه به هندسه توده‌سنگ می‌توان پارامترهای آتشیاری نظیر فاصله‌داری، بارسنگ و انتخاب محل تقویت خرج را انتخاب کرد. به عنوان مثال با توجه به مرکز هندسی بلوک‌ها می‌توان محل چال‌ها را طوری انتخاب کرد که از ایجاد بلوک‌های بزرگ جلوگیری شود. یا از چال‌های مورب با زاویه مناسب برای آتشیاری استفاده کرد. در این شکل یک پله معدن روباز با مشخصات زیر وجود دارد:

- ارتفاع پله ۱۵ متر، شیب پله ۷۵ درجه و عرض پله ۱۰ متر
- مشخصات دسته درزه‌ها: دسته درزه اول: شیب صفر درجه و فاصله داری ۳ متر، دسته درزه دوم: شیب ۳۰ درجه و فاصله داری ۳ متر، دسته درزه سوم: شیب ۱۳۰ و فاصله داری

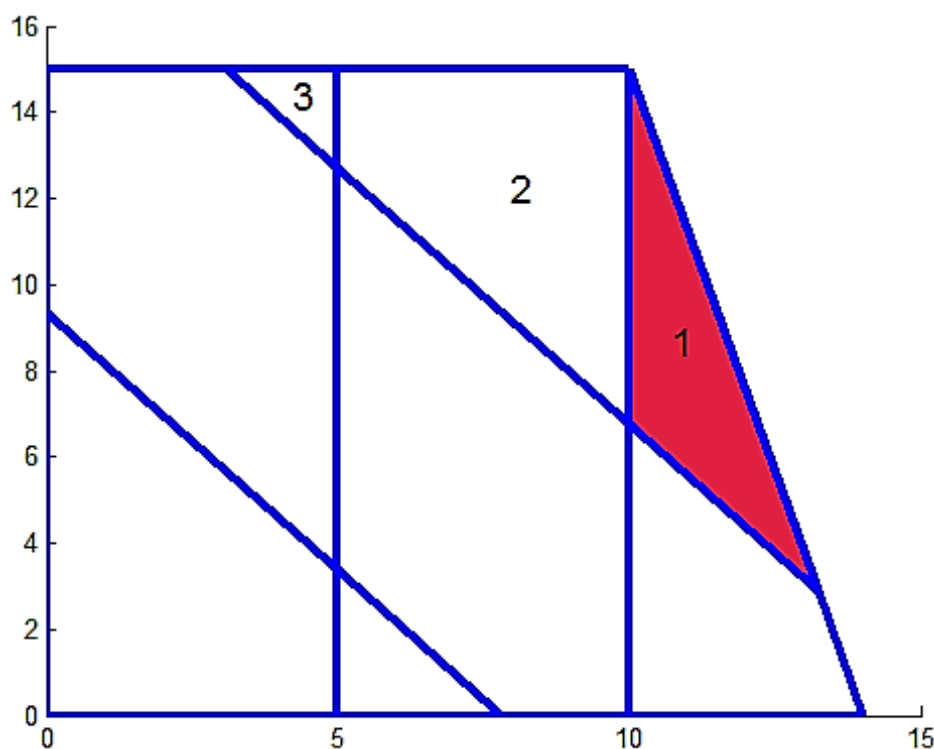
۳ متر



شکل ۴-۲۳: نمای پله یک معدن روباز و درزه‌های آن

۴-۵-۴ مثال ۸: پایداری و طراحی نگهداری بلوک‌ها در شیروانی‌ها

برای نگهداری بلوک‌های ناپایدار در فضاهای زیر زمینی و روباز نیاز به هندسه این بلوک‌ها و پارامترهای زمین شناسی و مقاومتی توده‌سنگ است [۳۱]. بلوک کلیدی به بلوکی گفته می‌شود که پایداری یک یا چند بلوک به پایداری این بلوک بستگی داشته باشد. بلوک ۱ در شکل (۴-۲۴) یک بلوک کلیدی برای پله نشان داده شده در این شکل است. هندسه این بلوک کلیدی برای تحلیل پایداری و طراحی نگهداری آن مورد نیاز می‌باشد.



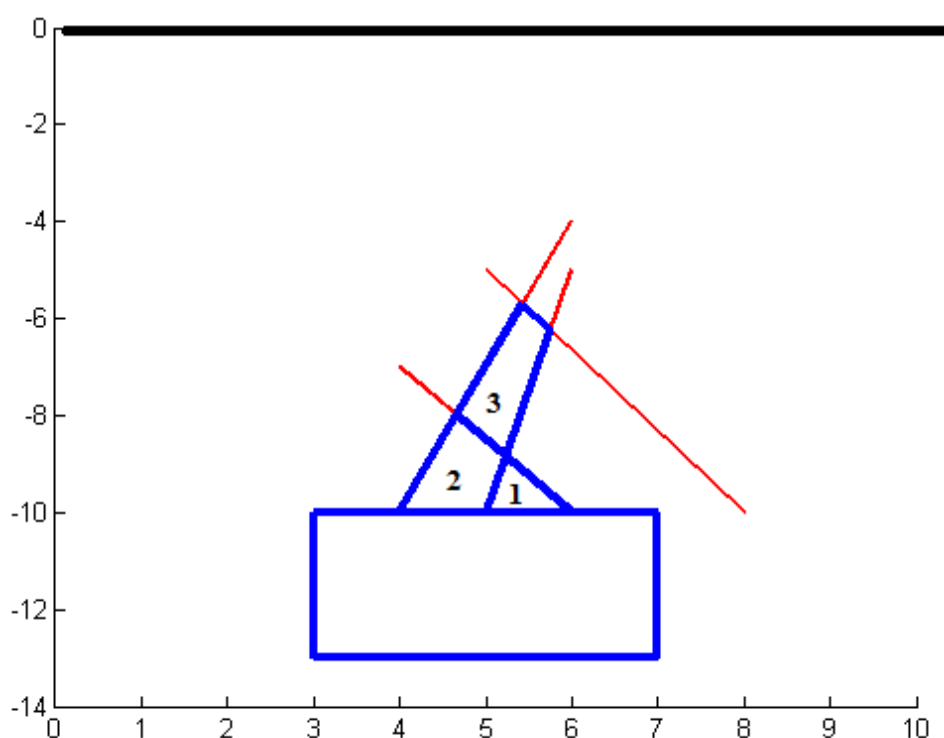
شکل ۴-۲۴: هندسه یک شیروانی و بلوک کلیدی در این شکل

این شیروانی دارای ارتفاع ۱۵ متر و شیب ۷۵ درجه بوده و دو دسته درزه با شیب ۹۰ و ۱۳۰ درجه و فاصله‌داری به ترتیب ۵ و ۶ متر است. با توجه به خروجی برنامه می‌توان تشخیص داد که بلوک ۱ یک بلوک کلیدی برای بلوک‌های ۲ و ۳ است و مختصات رئوس این بلوک برابر با (۱۵، ۱۰)، (۲/۹، ۱۳/۲)

و (۶/۷، ۱۰) بوده و دارای مساحت $۱۳/۴$ متر مربع است که برای طراحی نگهداری بلوک این اطلاعات لازم است.

۴-۵-۵ مثال ۹: پایداری و طراحی نگهداری بلوک‌ها در معادن زیرزمینی

فرض می‌شود همانند شکل (۴-۲۵) دهانه تونل یک معدن زیرزمینی با ابعاد ۳ متر ارتفاع و ۴ متر عرض در عمق ۱۰ متری از سطح وجود داشته باشد. اگر ناپیوستگی‌هایی مطابق با جدول (۴-۷) در بالای دهانه تونل شناسایی شوند، هندسه بلوک‌های ۱، ۲ و ۳ برای تحلیل پایداری و طراحی نگهداری آن‌ها مورد نیاز است.



شکل ۴-۲۵: دهانه تونل یک معدن در عمق ۱۰ متری

جدول ۷-۴: ناپیوستگی‌های برداشت شده بالای تونل

شماره ناپیوستگی	طول نقطه ابتدا (x_s)	عرض نقطه ابتدا (y_s)	طول نقطه انتها (x_e)	عرض نقطه انتها (y_e)
۱	۵	-۱۰	۶	-۵
۲	۴	-۷	۶	-۱۰
۳	۵	-۵	۸	-۱۰
۴	۴	-۱۰	۶	-۴

در جدول (۷-۴) مختصات نقاط ابتدا و انتهای درزه‌های برداشت شده وجود دارد.

برای تحلیل پایداری بلوک‌ها نیاز به مساحت آن‌ها می‌باشد. همچنین مختصات رئوس برای طراحی نگهداری می‌تواند مورد استفاده قرار بگیرد. در جدول (۸-۴) مشخصات هندسی بلوک‌های تشکیل شده بالای تونل وجود دارد.

جدول ۸-۴: مشخصات هندسی بلوک‌های تشکیل شده در این مثال

شماره بلوک	مساحت بلوک (متر مربع)	مختصات رئوس بلوک (متر)
۱	۰/۵۸	(۸/۸۴ و ۵/۲۳)، (۵ و ۱۰)، (۵ و ۱۰)، (۶ و ۱۰)
۲	۱/۴۲	(۸/۸۴ و ۵/۲۳)، (۵ و ۱۰)، (۵ و ۱۰)، (۴ و ۱۰)، (۴ و ۱۰)، (۴ و ۶۶)
۳	۱/۵۲	(۵/۷۱ و ۵/۴۳)، (۶/۲۵ و ۵/۷۵)، (۵ و ۱۰)، (۴ و ۱۰)، (۴ و ۶۶)، (۵/۲۳ و ۸/۸۴)

۴-۵-۶ مثال ۱۰: پیش بینی رقیق شدگی ماده معدنی و باطله در روش‌های استخراج تخریبی

اطلاع از توزیع حجم و شکل بلوک‌های توده‌سنگ کاربردهای فراوانی دارد. به عنوان مثال برای بررسی اختلاط ماده معدنی و باطله در روش‌های تخریبی از توزیع حجمی و شکل بلوک‌های برجای توده سنگ استفاده می‌شود. هندسه بلوک‌ها نیز در مکانیسم رقیق شدگی مؤثر هستند [۳۱].

رقیق شدگی ماده معدنی در معادنی که به روش تخریبی استخراج می‌شوند به وسیله ناپیوستگی‌ها و مشخصه‌های ساختاری توده‌سنگ میزبان کنترل می‌شود. در روشی ارائه شده توسط کالنچوکا^۱ [۳۲] برای کنترل رقیق شدگی ماده معدنی مشخصات برجای توده‌سنگ نظیر توزیع حجم بلوک‌ها مورد نیاز است.

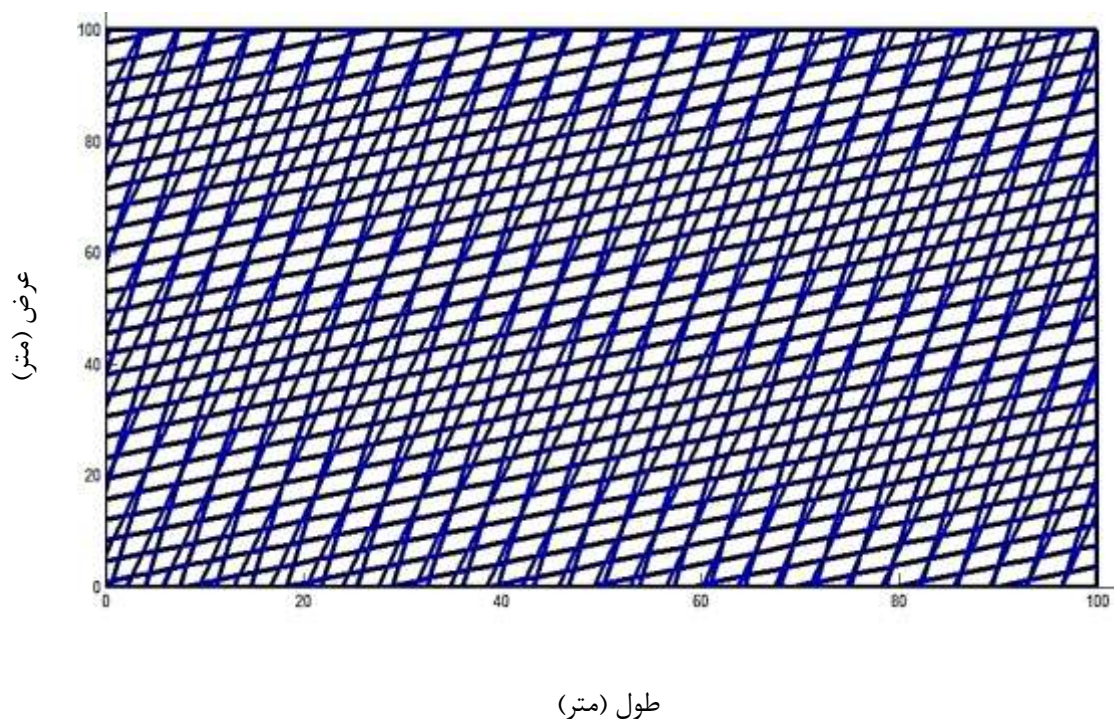
فرض می‌شود در سطح ۱۰۰ متر در ۱۰۰ متر از سطح زمینی که در بر گیرنده ماده معدنی و باطله است، دسته درزه‌هایی مطابق با جدول (۴-۹) وجود دارد، برای بررسی رقیق شدگی ماده معدنی و باطله هنگام تخریب بلوکی نیاز به D_{20} ، D_{40} ، D_{60} و D_{80} حجم بلوک‌های بر جای توده سنگ است.

جدول ۴-۹: دسته درزه‌های مثال ۱۰

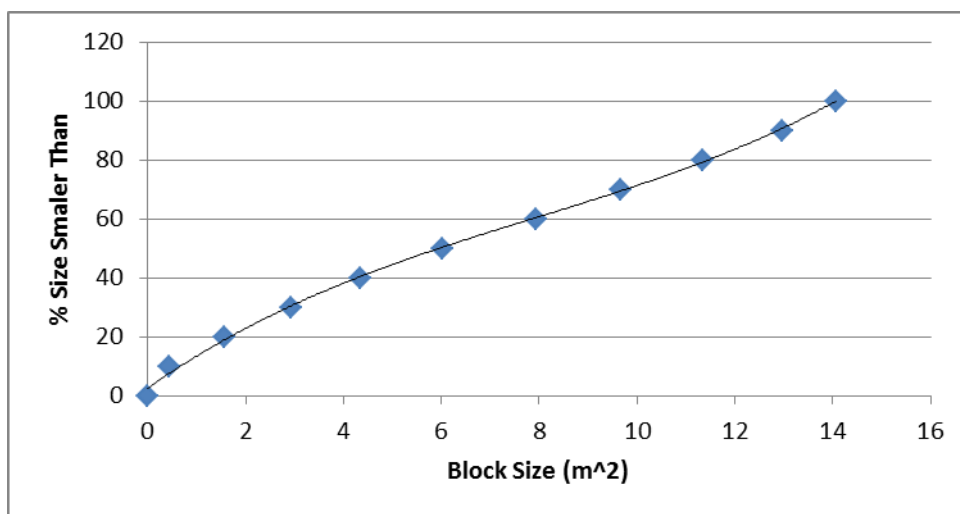
شماره دسته درزه	فاصله‌داری (متر)	امتداد یا زاویه با محور X (درجه)
۱	۳/۵	۲۰
۲	۳/۵	۷۵
۳	۳/۵	۸۰

نمای افقی منطقه مورد مطالعه به همراه دسته درزه‌های آن که خروجی برنامه تهیه شده است در شکل (۴-۲۶) نشان داده شده است. شکل (۴-۲۷) نمودار توزیع مساحت بلوک‌های برجای توده‌سنگ را نشان می‌دهد که برای تحلیل رقیق شدگی ماده معدنی مورد نیاز است.

¹ Kalenchuka



شکل ۴-۲۶: نمای افقی منطقه و بلوک‌های آن



شکل ۴-۲۷: نمودار توزیع تجمعی اندازه بلوک‌های مثال ۱۰

۴-۶ نتیجه‌گیری

در این فصل به تشریح الگوریتم مبتنی بر ماتریس‌ها پرداخته شد، این الگوریتم یک الگوریتم ابتکاری و در مقایسه با سایر الگوریتم‌ها الگوریتم ساده‌تری است. برنامه کامپیوتری تهیه شده در این پایان نامه با استفاده از این الگوریتم و در محیط نرم افزار MATLAB کد نویسی شده است. ورودی این برنامه درزه‌ها و محدوده مورد مطالعه و خروجی برنامه نیز اطلاعات کامل هندسی توده‌سنگ و بلوک‌های آن است. اعتبار سنجی برنامه توسط حل دو مثال به صورت دستی و در مقایسه با نتایج حل برنامه‌های دیگر بررسی شد و مشخص گردید که نتایج خروجی این برنامه صحیح هستند. از این برنامه در کاربردهای مختلف از جمله بلوک‌دهی معادن سنگ ساختمانی، بررسی خردایش، پایداری بلوک‌های سنگی و پیش بینی رقیق شدگی ماده معدنی و باطله در روش‌های استخراج تخریبی، می‌توان استفاده کرد.

فصل پنجم

جمع‌بندی، نتیجه‌گیری و پیشنهادات

۵-۱ جمع‌بندی

پوسته زمین از سنگ و خاک ساخته شده است. فعالیت‌های معدنی و عمرانی در ارتباط با پوسته زمین انجام می‌شوند. با توجه به شرایط زمین‌شناسی منطقه ممکن است این فعالیت‌ها در محیط سنگی یا خاکی انجام شود. در اثر حرکت صفحات زمین، فرسایش و ... در سنگ‌ها ناپیوستگی‌هایی ایجاد می‌شود، به مجموعه ماده‌سنگ و ناپیوستگی‌ها توده‌سنگ گفته می‌شود. انواع ناپیوستگی‌ها باعث تشکیل بلوک‌هایی با ابعاد و اشکال مختلف در توده‌سنگ می‌شوند. این بلوک‌های تشکیل شده رفتار توده‌سنگ را کنترل می‌کنند. در بلوک دهی سنگ‌های ساختمانی، بررسی انرژی خردایش و طراحی الگوی آتشباری، پایداری بلوک‌های سنگی در توده‌سنگ، کنترل رقیق شدگی ماده معدنی و ... اطلاع شناخت توده‌سنگ و بلوک‌های تشکیل شده در آن ضروری است.

در گذشته برای مطالعه توده‌سنگ با استفاده از مشخصات هندسی درزه‌های برداشت شده حجم و شکل یکی از بلوک‌های توده‌سنگ به عنوان نماینده آن مورد تحلیل قرار می‌گرفت. امروزه درزه‌های توده‌سنگ با استفاده از روش‌های آماری و زمین‌آماري شبیه‌سازی شده و یک الگوی مناسب به درزه‌های توده‌سنگ اختصاص می‌یابد. با توجه به این الگو و روش‌های مختلفی که ذکر شد به تحلیل توده‌سنگ و بلوک‌های موجود در آن پرداخته می‌شود. در این روش‌ها با استفاده از الگوریتم‌های مختلف توده‌سنگ و بلوک‌های آن مدل‌سازی می‌شوند. الگوریتم‌ها برای تهیه برنامه‌های کامپیوتری طراحی می‌شوند. برخی از الگوریتم‌های طراحی شده جهت محاسبه شکل و حجم بلوک‌های توده‌سنگ عبارتند از: الگوریتم مبتنی بر تئوری گراف‌ها، الگوریتم مبتنی بر روش‌های برداری، الگوریتم تکنیک‌های عددی، الگوریتم ماتریسی و ... همه این الگوریتم‌ها برای حالت‌های دو و سه بعدی طراحی شده‌اند. نرم افزارهای SIMBLOCK، UDEC، نرم افزار ارائه شده توسط خیشوند و همکاران و در این ارتباط با این کاربرد و کاربردهای دیگر طراحی شده‌اند.

۵-۲ نتیجه‌گیری

با توجه به کاربردهای مختلف تعیین مشخصات توده‌سنگ و بلوک‌های آن نیاز به مطالعه در مورد آن وجود دارد. در این پایان‌نامه بیان شد که دو رویکرد کلی برای تعیین مشخصات هندسی توده‌سنگ وجود دارد. در رویکرد اول ناپیوستگی‌ها نامحدود و موازی با یکدیگر فرض می‌شوند. به دلیل فرضیات غیر واقع و خطای بالا روش‌های این رویکرد کمتر مورد استفاده قرار می‌گیرند. در رویکرد دوم توده‌سنگ و بلوک‌های آن در کامپیوتر مدل‌سازی شده و مشخصات هندسی بلوک‌ها محاسبه می‌شود. برای مدل‌سازی و تهیه برنامه کامپیوتری از الگوریتم‌های مختلف که ذکر استفاده می‌شود. الگوریتم تئوری گرافی دارای پیچیدگی است و زمان زیاد برای فهم و پیاده‌سازی آن لازم است. الگوریتم روش‌های عددی با توجه به در نظر گرفتن دو جهت برای هر ناپیوستگی یا درزه طولانی است. الگوریتم تئوری بلوکی بر اساس شمارش کامل خطوط برای هر بلوک است و لذا بهینه نیست. در نتیجه برای تهیه برنامه کامپیوتری در این پایان‌نامه از الگوریتم ماتریسی به دلیل فهم ساده و پیاده سازی راحت استفاده شده است. برنامه کامپیوتری تهیه شده با توجه به اعتبار سنجی انجام شده دارای نتایج صحیح است و از این برنامه می‌توان در کاربردهای مختلف استفاده کرد.

۵-۳ پیشنهادات

از برنامه تهیه شده در این پایان‌نامه می‌توان در کاربردهای متفاوت از جمله بلوک دهی سنگ‌های ساختمانی، پایداری بلوک‌ها، انرژی خردایش و طراحی آتشفباری و رقیق شدگی ماده معدنی در روش‌های استخراج تخریبی و ... استفاده کرد. برنامه تهیه شده برای شناسایی بلوک‌های دو بعدی است و می‌توان برای شناسایی بلوک‌های سه بعدی نیز با استفاده از این برنامه، برنامه‌های دیگری تهیه کرد.

منابع

- [1] **Sharma V.M.** and Saxena K.R. (2001) "In In-situ characterization of rocks." A.A. Balkema publishers, 49-97.
- [2] **Lu J.** (2002) "Systematic identification of polyhedral rock blocks with arbitrary joints and faults." *Computers and Geotechnics*, 29, 49–72.
- [3] **Lu P.** and Latham J.P. (1999) "Developments in the Assessment of In-situ Block Size Distributions of Rock Masses." *Rock Mechanics and Rock Engineering*, 32 (1), 29-49
- [4] **Starzeca P.** and Tsang C.F. (2002) "Use of fracture-intersection density for predicting the volume of unstable blocks in underground openings." *International Journal of Rock Mechanics & Mining Sciences*, 39, 807–813
- [5] **Smith J.V.** (2004) "Determining the size and shape of blocks from linear sampling for geotechnical rock mass classification and assessment.", *Journal of Structural Geology*, 26, 1317–1339.
- [6] **Ferreira A.** and Fonseca M. Jorge J. (2000) "Polygon Detection from a Set of Lines." Department of Information Systems and Computer Science.
- [7] **Heliot D.** (1988) "Generating a blocky rock mass." *International Journal of Rock Mechanics and Mining Sciences and Geomechanics, Abstracts*, 25, 127–138.
- [۸] **گودمنر و شی ج ه**، ترجمه دهقان بنادکی م م، (۱۳۸۰)، "تئوری بلوک و کاربرد آن در مهندسی سنک"، انتشارات دانشگاه شاهرود
- [9] **Shi G.** (1988) "Discontinuous deformation analysis – a new numerical model for the statics and dynamics of block systems." PhD thesis, University of California, Berkeley, CA, USA.
- [10] **Lin D.**, Fairhurst C. and Starfield M. (1987) "Geometrical identification of three-dimensional rock block systems using topological techniques." *International Journal of Rock Mechanics and Mining Sciences and Geomechanics, Abstracts*, 24, 331–338.
- [11] **Maerz N.H.** and Germain P. (1996) "Block size determination around underground openings using simulations." *Proceedings of the FRAGBLAST 5 Workshop on Measurement of Blast Fragmentation*, Montreal, Quebec, Canada, pp 215-223.
- [12] **Jing L.** and Stephansson O. (1994) "Topological identification of block assemblages for jointed rock masses." *International Journal of Rock Mechanics and Mining Sciences and Geomechanics, Abstracts*, 31, 163–172.
- [13] **Jing, L.** (2000) "Block system construction for three-dimensional discrete element models of fractured rocks." *International Journal of Rock Mechanics and Mining Sciences*, 37, 645–659.
- [14] **Jing L.** and Stephansson, O. (2007) "Fundamentals of discrete element methods for rock engineering: theory and applications" *Developments in Geotechnical Engineering*, Amsterdam: Elsevier Science.

- [15] **Ulker E.** Turanboy A. (2009) "Maximum volume cuboids for arbitrarily shaped in-situ rock blocks as determined by discontinuity analysis—A genetical algorithm approach." Computers & Geosciences, 35, 1470–1480
- [16] **Khishvand M.,** Jafari A. and Rahami H. (2011) "Developing An Algorithm For Reconstruction Blocky System in Discountinuous Media: Two Dimensional Analysis", Geomechanics and Geoengineering, 171-183.
- [17] **Zhang Y.,** Xiao M. and Chen J. (2010) "A new methodology for block identification and its application in a large scale underground cavern complex." Tunnelling and Underground Space Technology, 25, 168–180
- [18] **Zhang Y,** Xiao M. Ding X. and Wua A (2012) "Improvement of methodology for block identification using mesh gridding technique." Tunnelling and Underground Space Technology, 30, 217–229
- [19] **Essaieb H.** and Jean du M. (2005) "A methodology for rock mass characterisation and classification to improve blast results" International Journal of Rock Mechanics & Mining Sciences, 42, 177–194.
- [۲۰] **امینی ا،** قدیر زاده م، یاراحمدی بافقی ع، (۱۳۸۵)، "مدل سازی هندسی-ژئوتکنیکی توده سنگ های درزه دار به روش آماری در محیط Mathematica (مطالعه موردی بلوک ۲ معدن چغارت)"، پنجمین کنفرانس دانشجویی مهندسی معدن، ص ۳۲۳، اصفهان
- [21] **Palmström A.** (2005) "Measurements of and correlations between block size and rock quality designation (RQD)." Tunnelling and Underground Space Technology, 20, 362–377
- [22] **Bentley J.L.** and Ottmann T. (1979) "Algorithms for reporting and counting geometric intersections." IEEE Transactions on Computers, 643–647.
- [23] **Horton J.D.** (1984) "A polynomial-time algorithm to find the shortest cycle basis of a graph." SIAM Journal on Computing, 16(2), 358–366
- [24] **Balaban I.J.** (1995) "An optimal algorithm for finding segment intersections." In Proceedings of the 11th Annual ACM Symposium Comp. Graph, 211–219.
- [25] **Chazelle B.** and Edelsbrunner H. (1992) "An optimal algorithm for intersecting line segments in the plane." Journal of the ACM, 39:1–54.
- [26] **Clarkson K.** and Shor P.W. (1995) "Applications of random sampling in computational geometry." Discrete and Computational Geometry, 4:387–421.
- [27] **Bartuschka U.,** Mehlhorn K. and Naher S. (1997) "A robust and efficient implementation of a sweep line algorithm for the straight line segment intersection", In Proceedings of Workshop on Algorithm Engineering, 124–135.
- [28] **Dogrusoz U.** and Krishnamoorthy M. (1995) "Cycle vector space algorithms for enumerating all cycles of a planar graph." Technical Report 5, Rensselaer Polytechnic Institute, Dept. of Computer Science, Troy, New York.
- [29] **Cormen T.,** Leiserson Ch., Rivest R. and Stein C. (2009) "Introduction to Algorithms." MIT Press, Cambridge, Massachusetts London, England, 3rd Edition.
- [30] **Essaieb H.** and Romdhane R.N. (2008) "Fragmentation Energy in Rock Blasting", Geotech Geol Eng, 26, 133–146.
- [31] **Kalenchuk K.,** Diederichs M. and McKinnon S. (2006) "Characterizing block geometry in jointed rock masses", International Journal of Rock Mechanics & Mining Sciences, 43 , 1212–1225.

[32] **Kalenchuka K.**, McKinnonb S. and Diederichs M. (2008) “*Block geometry and rockmass characterization for prediction of dilution potential into sub-level cave mine voids*”, International Journal of Rock Mechanics & Mining Sciences, 45, 929–940.

پیوست

کد تهیه شده در این پایان نامه

```

%*                In The Name Of Allah                *%
%*                *%
%*                2D Block Detector.m                *%
%+-----+*%
%| Author : Reza Yarahmadi                               |*%
%| Supervisors: Prof. Reza Kakaie & Assist Prof. Hossein Mirzaie    |*%
%| E-mail : r.yarahmadi@shahroodut.ac.ir                     |*%
%| Created: July 2012                                         |*%
%+-----+*%

%* This code is develop for 2D block detection base on mining problem *%

    %* First we should define geometry of model *%

    disp('If you have not each type of inputing data press 0 (zero) when ask you:')
    fprintf('\n')
    Ps=input('Input matrices of single discountinuties coordinate:'); % this matrices
have 4 cloumn that define each joint %
    fprintf('\n')

    MB=input('Input dimension of master block [Xm , Ym], if exist:');
    fprintf('\n')

    Teta=input('Input angle of joint sets with horizontal line (X) [@1 @2 ...]:');

    fprintf('\n')
    S=input('Input Spacing of joint sets (S) [S1 S2 ...]:');
    fprintf('\n')

Pm=zeros(1,4);

[m,n]=size(Ps);

```

```

for i=1:m
    Pm(i,:)=Ps(i,:);
end

if MB~=0
    Mb=zeros(4,4);
    Mb(1,1:2)=0; Mb(1,3)=MB(1,1); Mb(1,4)=0;
    Mb(2,1)=MB(1,1); Mb(2,2)=0; Mb(2,3)=MB(1,1); Mb(2,4)=MB(1,2);
    Mb(3,1)=0; Mb(3,2)=MB(1,2); Mb(3,3)=MB(1,1); Mb(3,4)=MB(1,2);
    Mb(4,1:3)=0; Mb(4,4)=MB(1,2);

    [p,q]=size(Pm);
    for i=1:4
        Pm(i+p,:)=Mb(i,:);
    end
end

if S~=0
if MB~=0
    [m,n]=size(S);
    for i=1:n
        u(i,1)=floor((MB(1,1)/S(i)));
        u(i,2)=floor((MB(1,2)/S(i)));
        if Teta(i)==0
            [p,q]=size(Pm);
            for j=1:u(i,2)
                if (j*S(i))<=MB(1,2)
                    Pm(j+p,1)=0;
                    Pm(j+p,2)=j*S(i);
                    Pm(j+p,3)=MB(1,1);
                    Pm(j+p,4)=j*S(i);
                end
            end
        end
    end
end

```

```

elseif Teta(i)>0 && Teta(i)<90
    alfa(i)=atand(MB(1,2)/MB(1,1));
    c(i)=S(i)/sind(180-(alfa(i)+Teta(i)));
    d(i)=floor(sqrt((MB(1,1)^2+MB(1,2)^2)/c(i)));
    for j=1:d(i)
        [p,q]=size(Pm);
        if ((j*S(i))/sind(Teta(i)))<=MB(1,1) && ((j*S(i))/cosd(Teta(i)))<=MB(1,2)

            Pm(j+p,1)=(MB(1,1)-j*S(i)/sind(Teta(i)));
            Pm(j+p,2)=0;
            Pm(j+p,3)=MB(1,1);
            Pm(j+p,4)=j*S(i)/cosd(Teta(i));

        elseif ((j*S(i))/sind(Teta(i)))>MB(1,1) &&
((j*S(i))/cosd(Teta(i)))<=MB(1,2)

            Pm(j+p,1)=0;
            Pm(j+p,2)=(j*S(i)/sind(Teta(i))-MB(1,1))*tand(Teta(i));
            Pm(j+p,3)=MB(1,1);
            Pm(j+p,4)=j*S(i)/cosd(Teta(i));

        elseif ((j*S(i))/sind(Teta(i)))<=MB(1,1) &&
((j*S(i))/cosd(Teta(i)))>MB(1,2)

            Pm(j+p,1)=(MB(1,1)-j*S(i)/sind(Teta(i)));
            Pm(j+p,2)=0;
            Pm(j+p,3)=MB(1,1)-( j*S(i)/cosd(Teta(i))-MB(1,2) )/tand(Teta(i));
            Pm(j+p,4)=MB(1,2);

        elseif ((j*S(i))/sind(Teta(i)))>MB(1,1) &&
((j*S(i))/cosd(Teta(i)))>MB(1,2)

            Pm(j+p,1)=0;
            Pm(j+p,2)=(j*S(i)/sind(Teta(i))-MB(1,1))*tand(Teta(i));

```

```

        Pm(j+p,3)=MB(1,1)-( j*S(i)/cosd(Teta(i))-MB(1,2) )/tand(Teta(i));
        Pm(j+p,4)=MB(1,2);
    end
end
elseif Teta(i)==90
    [p,q]=size(Pm);
    for j=1:u(i,1)
        if (j*S(i))<=MB(1,1)
            Pm(j+p,1)=j*S(i);
            Pm(j+p,2)=0;
            Pm(j+p,3)=j*S(i);
            Pm(j+p,4)=MB(1,2);
        end
    end
elseif Teta(i)>90 && Teta(i)<180
    alfa(i)=atand(MB(1,2)/MB(1,1));
    c(i)=S(i)/sind(180-(alfa(i)+180-Teta(i)));
    d(i)=floor(sqrt((MB(1,1)^2+MB(1,2)^2)/c(i)));

    for j=1:d(i)
        [p,q]=size(Pm);
        if ((j*S(i))/sind(180-Teta(i)))<=MB(1,1) && ((j*S(i))/cosd(180-
Teta(i)))<=MB(1,2)

            Pm(j+p,1)=(j*S(i))/sind(180-Teta(i));
            Pm(j+p,2)=0;
            Pm(j+p,3)=0;
            Pm(j+p,4)=j*S(i)/cosd(180-Teta(i));

        elseif ((j*S(i))/sind(180-Teta(i)))>MB(1,1) && ((j*S(i))/cosd(180-
Teta(i)))<=MB(1,2)

            Pm(j+p,1)=MB(1,1);

```

```

Pm(j+p,2)=(j*S(i)/sind(180-Teta(i))-MB(1,1))*tand(180-Teta(i));
Pm(j+p,3)=0;
Pm(j+p,4)=j*S(i)/cosd(180-Teta(i));

elseif ((j*S(i))/sind(180-Teta(i)))<=MB(1,1) && ((j*S(i))/cosd(180-
Teta(i)))>MB(1,2)

Pm(j+p,1)=(j*S(i)/sind(180-Teta(i)));
Pm(j+p,2)=0;
Pm(j+p,3)=( j*S(i)/cosd(180-Teta(i))-MB(1,2) )/tand(180-Teta(i));
Pm(j+p,4)=MB(1,2);

elseif ((j*S(i))/sind(180-Teta(i)))>MB(1,1) && ((j*S(i))/cosd(180-
Teta(i)))>MB(1,2)

Pm(j+p,1)=MB(1,1);
Pm(j+p,2)=(j*S(i)/sind(180-Teta(i))-MB(1,1))*tand(180-Teta(i));
Pm(j+p,3)=( j*S(i)/cosd(180-Teta(i))-MB(1,2) )/tand(180-Teta(i));
Pm(j+p,4)=MB(1,2);
end
end
elseif Teta(i)==180
[p,q]=size(Pm);
for j=1:u(i,2)
if (j*S(i))<=MB(1,2)
Pm(j+p,1)=0;
Pm(j+p,2)=j*S(i);
Pm(j+p,3)=MB(1,1);
Pm(j+p,4)=j*S(i);
end
end
end
end
end

```

```

end
end

[m,n]=size(Pm);
k=1;
for i=1:m
    if Pm(i,1)==0 && Pm(i,2)==0 && Pm(i,3)==0 && Pm(i,4)==0
    else
        Pmr(k,:)=Pm(i,:);
        k=k+1;
    end
end
end

% Geometry of model are loaded, let calculate block volume %
[f,g]=size(Pmr);
for i=1:f
    line([Pmr(i,1) , Pmr(i,3)], [Pmr(i,2) , Pmr(i,4)], 'Linewidth',2, 'Color', 'r')
end
[f,g]=size(Pmr);
l=1;          % In this loop we calculate intersection point %
for i=1:f
    ka(i)=0;
    for j=1:f

        r(j)=(1/2)*sqrt((Pmr(j,3) - Pmr(j,1))^2 + (Pmr(j,4) - Pmr(j,2))^2);
        xc(j)=(Pmr(j,1) + Pmr(j,3))/2;
        yc(j)=(Pmr(j,2) + Pmr(j,4))/2;
        if Pmr(j,3)==Pmr(j,1)
            te(j)=90;
        else
            te(j)=atand((Pmr(j,4)-Pmr(j,2))/(Pmr(j,3)-Pmr(j,1)));
        end
        nx(j)=cosd(te(j));

```

```

ny(j)=sind(te(j));
D(i,j)=sqrt((xc(i)-xc(j))^2+(yc(i)-yc(j))^2);
if nx(i)*nx(j)+ ny(i)*ny(j) == 1 || nx(i)*nx(j)+ ny(i)*ny(j) == -1 || D(i,j) >
(r(i)+r(j))
    e(i,j)=0;
else
    del(i,j)=(Pmr(i,3)-Pmr(i,1))*(Pmr(j,4)-Pmr(j,2))-(Pmr(j,3)-Pmr(j,1))*(Pmr(i,4)-
Pmr(i,2));
    xb(i,j)=(1/del(i,j))*((Pmr(j,1)*Pmr(j,4)-Pmr(j,3)*Pmr(j,2))*(Pmr(i,3)-Pmr(i,1))-
(Pmr(i,1)*Pmr(i,4)-Pmr(i,3)*Pmr(i,2))*(Pmr(j,3)- Pmr(j,1)));
    yb(i,j)=(1/del(i,j))*((Pmr(i,1)*Pmr(i,4)-Pmr(i,3)*Pmr(i,2))*(Pmr(j,2)-Pmr(j,4))-
(Pmr(j,1)*Pmr(j,4)-Pmr(j,3)*Pmr(j,2))*(Pmr(i,2)-Pmr(i,4)));
    if (abs(xb-floor(xb))<=(8.882e-16))
        xb=floor(xb);
    elseif (abs(xb-ceil(xb))<=(8.882e-16))
        xb=ceil(xb);
    end
    if (abs(yb-ceil(yb))<=(8.882e-16))
        yb=ceil(yb);
    elseif (abs(yb-floor(yb))<=(8.882e-16))
        yb=floor(yb);
    end

    if Pmr(j,4)~=Pmr(j,2)
        ti(j)=(yb(i,j)-Pmr(j,2))/(Pmr(j,4)-Pmr(j,2));
    else
        ti(j)=(xb(i,j)-Pmr(j,1))/(Pmr(j,3)-Pmr(j,1));
    end
    if Pmr(i,4)~=Pmr(i,2)
        ti(i)=(yb(i,j)-Pmr(i,2))/(Pmr(i,4)-Pmr(i,2));
    else
        ti(i)=(xb(i,j)-Pmr(i,1))/(Pmr(i,3)-Pmr(i,1));
    end
    end
    if abs(ti(i))<=.0001 || abs(ti(i)-1)<=.0001

```

```

        ti(i)=0;
    end
    if abs(ti(j))<=.0001 || abs(ti(j)-1)<=.0001
        ti(j)=0;
    end

    if ti(i)>=0 && ti(i)<=1 && del(i,j)~=0 && ti(j)>=0 && ti(j)<=1

        ka(i)=ka(i)+1;
        if Pmr(i,4)~=Pmr(i,2)
            t(i,ka(i))=(yb(i,j)-Pmr(i,2))/(Pmr(i,4)-Pmr(i,2));
        else
            t(i,ka(i))=(xb(i,j)-Pmr(i,1))/(Pmr(i,3)-Pmr(i,1));

        end
        p(i,2*ka(i)-1)=xb(i,j);
        p(i,2*ka(i))=yb(i,j);
        e(i,j)=1;

        if i<j
            Pin(l,1)=xb(i,j);
            Pin(l,2)=yb(i,j);
            l=l+1;
        end
    end

end

end

end
end

```

```

[m,n]=size(p);
for i=1:m
    for j=1:ka(i)
        for k=j+1:ka(i)
            if t(i,j)>t(i,k)

                c1=p(i,2*j-1);
                p(i,2*j-1)=p(i,2*k-1);
                p(i,2*k-1)=c1;
                c2=p(i,2*j);
                p(i,2*j)=p(i,2*k);
                p(i,2*k)=c2;
                c3=t(i,j);
                t(i,j)=t(i,k);
                t(i,k)=c3;
            end
        end
    end

end

k=1;
[m n]=size(p);
for i=1:m
    if ka(i)>=2
        for j=1:ka(i)-1
            if p(i,2*j-1)==p(i,2*j+1) && p(i,2*j)==p(i,2*j+2)
            else
                ed(k,1:4)=p(i,2*j-1:2*j+2);
                k=k+1;
            end
        end
    end
end
end
end

```

```

w=size(ed,1);
cp=zeros(w,w);
for i=1:w
    for j=1:w
        if i~=j
            if ed(i,1)==ed(j,1) && ed(i,2)== ed(j,2)
                cp(i,j)=1;
            elseif ed(i,3)==ed(j,3) && ed(i,4)== ed(j,4)
                cp(i,j)=1;
            elseif ed(i,1)==ed(j,3)&&ed(i,2)== ed(j,4)
                cp(i,j)=1;
            elseif ed(i,3)==ed(j,1)&&ed(i,4)== ed(j,2)
                cp(i,j)=1;
            end
        end
    end
end
end
cpsum=sum(cp,2);
[m,n]=size(cp);
k=1;
for i=1:m
    if cpsum(i)>=2
        edre(k,:)=ed(i,:);
        k=k+1;
    end
end
end
[m,n]=size(Pin);
k=1;
for i=1:m
    temp(i)=0;
    for j=i+1:m
        if Pin(i,1)==Pin(j,1) && Pin(i,2)==Pin(j,2);
            temp(i)=1;
        end
    end
end

```

```

        end
    end
    if temp(i)~=1
        vx(k,:)=Pin(i,:);
        k=k+1;
    end
end
[m,n]=size(vx);
cv=zeros(m,m);
for i=1:m
    for j=1:m
        for k=1:size(edre,1)
            if i~=j
                if (vx(i,1)==edre(k,1) && vx(i,2)==edre(k,2))||(vx(i,1)==edre(k,3) &&
vx(i,2)==edre(k,4))
                    if (vx(j,1)==edre(k,1) && vx(j,2)==edre(k,2))||(vx(j,1)==edre(k,3) &&
vx(j,2)==edre(k,4))
                        cv(i,j)=k;
                    end
                end
            end
        end
    end
end
end
end
[m,n]=size(cv);
tm=0;
edrg=edre;
for i=1:m
    tem(i)=0;
    for j=1:n
        if cv(i,j)~=0
            tem(i)=tem(i)+1;
        end
    end
end

```

```

end
if tem(i)==1
    for s=1:n
        if cv(i,s)~=0
            tm=cv(i,s);
            edrg((cv(i,s)),1)=0;edrg((cv(i,s)),2)=0;edrg((cv(i,s)),3)=0;edrg((cv(i,s)),4)=0;
        end
    end
end
end
end
[m,n]=size(edrg);
k=1;
for i=1:m
    if edrg(i,1)==0 && edrg(i,2)==0 && edrg(i,3)==0 && edrg(i,4)==0
    else
        edreg(k,:)=edrg(i,:);
        k=k+1;
    end
end
[m,n]=size(edreg);
for i=1:m
    line([edreg(i,1) , edreg(i,3)],[edreg(i,2) , edreg(i,4)],'LineWidth',3)
end
[m,n]=size(cv);
for i=1:m
    k=1;
    el(i)=0;
    for j=1:n
        if cv(i,j)~=0
            cvr(i,k)=cv(i,j);
            k=k+1;
            el(i)=el(i)+1;
        end
    end
end

```

```

    end
end
[m,n]=size(edre);
for i=1:m
    edrev(i,1)=edre(i,3)-edre(i,1);
    edrev(i,2)=edre(i,4)-edre(i,2);
    nor(i,1)=edrev(i,1)/sqrt((edrev(i,1))^2+(edrev(i,2))^2);
    nor(i,2)=edrev(i,2)/sqrt((edrev(i,1))^2+(edrev(i,2))^2);

end
[m,n]=size(cvr);
for i=1:m
    for j=2:el(i)
        if edre(cvr(i,j),1)==edre(cvr(i,1),1) && edre(cvr(i,j),2)==edre(cvr(i,1),2)
            del2(i,j)=-nor(cvr(i,j),1)*nor(cvr(i,1),2) + nor(cvr(i,j),2)*nor(cvr(i,1),1);
            if del2(i,j)>=0
                tet(i,j)=180-acosd(dot(nor(cvr(i,j),:),-nor(cvr(i,1),:)));
            else
                tet(i,j)=180+acosd(dot(nor(cvr(i,j),:),-nor(cvr(i,1),:)));
            end
        elseif edre(cvr(i,j),3)==edre(cvr(i,1),1) && edre(cvr(i,j),4)==edre(cvr(i,1),2)
            del2(i,j)=(nor(cvr(i,j),1))*nor(cvr(i,1),2) - nor(cvr(i,j),2)*nor(cvr(i,1),1);
            if del2(i,j)>=0
                tet(i,j)=180-acosd(dot(-nor(cvr(i,j),:),-nor(cvr(i,1),:)));
            else
                tet(i,j)=180+acosd(dot(-nor(cvr(i,j),:),-nor(cvr(i,1),:)));
            end
        elseif edre(cvr(i,j),1)==edre(cvr(i,1),3) && edre(cvr(i,j),2)==edre(cvr(i,1),4)
            del2(i,j)=(nor(cvr(i,j),1))*nor(cvr(i,1),2) - nor(cvr(i,j),2)*nor(cvr(i,1),1);
            if del2(i,j)>=0
                tet(i,j)=180-acosd(dot(nor(cvr(i,j),:),nor(cvr(i,1),:)));
            else
                tet(i,j)=180+acosd(dot(nor(cvr(i,j),:),nor(cvr(i,1),:)));
            end
        end
    end
end

```

```

        end
elseif edre(cvr(i,j),3)==edre(cvr(i,1),3) && edre(cvr(i,j),4)==edre(cvr(i,1),4)
    del2(i,j)=(-nor(cvr(i,j),1))*nor(cvr(i,1),2) + nor(cvr(i,j),2)*nor(cvr(i,1),1);
    if del2(i,j)>=0
        tet(i,j)=180-acosd(dot(-nor(cvr(i,j),:),nor(cvr(i,1),:)));
    else
        tet(i,j)=180+acosd(dot(-nor(cvr(i,j),:),nor(cvr(i,1),:)));
    end
end
end
end
[m,n]=size(cvr);
for i=1:m
    for j=2:el(i)
        for k=j+1:el(i)
            if tet(i,j)>tet(i,k)
                c1=cvr(i,j);
                cvr(i,j)=cvr(i,k);
                cvr(i,k)=c1;
                c2=tet(i,j);
                tet(i,j)=tet(i,k);
                tet(i,k)=c2;
            end
        end
    end
end
[m,n]=size(cvr);
for i=1:m
    if cvr(i,1)~=0
        if cvr(i,n)~=0
            cvre(i,n+1)=cvr(i,1);
            cvre(i,1:n)=cvr(i,:);
        else

```

```

    for j=1:n
        if cvr(i,j)==0
            cvre(i,1:j-1)=cvr(i,1:j-1);
            cvre(i,j)=cvr(i,1);
            break
        end
    end
end
end
end
end
[m,n]=size(cvre);
for i=1:m
    if cvre(i,1)~=0
        for j=1:el(i)
            CE(cvre(i,j),cvre(i,j+1))=i;
        end
    end
end
end
ce=CE;
[m,n]=size(edre);
[h,v]=size(vx);
for i=1:m
    for j=1:h
        if edre(i,1)==vx(j,1) && edre(i,2)==vx(j,2)
            EV(i,1)=j;
        end
        if edre(i,3)==vx(j,1) && edre(i,4)==vx(j,2)
            EV(i,2)=j;
        end
    end
end
end
[m,n]=size(CE);
f=1;

```

```

g=1;
for i=1:m
    for j=1:n
        if CE(i,j)~=0
            holde(f,g)=CE(i,j);
            firsted=i;
            CE(i,j)=0;
            l=j;
            while l~=firsted
                for k=1:n
                    if l~=firsted && CE(l,k)~=holde(f,g) && CE(l,k)~=0
                        g=g+1;
                        holde(f,g)=CE(l,k);
                        CE(l,k)=0;
                        l=k;
                    end
                end
            end
            end
            g=1;
            f=f+1;
        end
    end
end
[m,n]=size(holde);
for i=1:m
    en(i)=0;
    for j=1:n
        if holde(i,j)~=0
            en(i)=en(i)+1;
            x(i,en(i))=vx(holde(i,j),1);
            y(i,en(i))=vx(holde(i,j),2);
        end
    end
end

```

```

end
[m,n]=size(x);
for i=1:m
    sum1(i)=x(i,en(i))*y(i,1);
    sum2(i)=y(i,en(i))*x(i,1);
    for j=2:n
        A(i)=x(i,j-1)*y(i,j);
        sum1(i)=A(i)+sum1(i);
        B(i)=y(i,j-1)*x(i,j);
        sum2(i)=B(i)+sum2(i);
    end
end
b=size(holde,2)+2;
for i=1:m
    Areat(i,1)=i;
    Areat(i,2)=(1/2)*(sum1(i)-sum2(i));
    Areat(i,3:b)=holde(i,:);
end
for i=1:m
    Area(i)=(1/2)*(sum1(i)-sum2(i));
end
disp('Areas of Blocks in the model:')
disp('  Block N   Block Area                Block edges  ')
disp(Areat)
[m,n]=size(Areat);
k=1;
for i=1:n
    if Area(i)<=0
        Aream(k)=-Area(i);
        k=k+1;
    end
end
end
Areasort=sort(Aream);

```

```
m=size(Areasort,2);
for i=1:11
    if i==1
        tt(i,1)=0;
        tt(i,2)=0;
    else
        tt(i,1)=Areasort(ceil(.1*(i-1)*m));
        tt(i,2)=10*(i-1);
    end
end
```

Abstract

Exact understanding of geometry of rock mass is an essential concern in different domains of rock engineering such as maintenance system of underground spaces built in jointed rock masses, designing blasting pattern of rock masses, optimization of fragmentation, determination of cube block in quarry mines and so on.

Different kinds of discontinuities turn the rock medium into a collection of rock blocks. The geometry of these blocks, that is, their form and size affect the behavior of the rock mass. Thus, it is necessary to know the volumetric distribution and form of the blocks.

The main purpose of this thesis is to study methods of defining geometry of rock mass blocks and choose proper algorithm to develop computer program for this scope and then developing a computer program to determine geometry of rock mass blocks.

In this thesis, computer program to determine the geometry of jointed rock mass is written in MATLAB™. The outputs of program according to validation done were corrected. And finally, by using this program some practical examples in aspect of mining engineering such as quarry, blasting, release and migration of a waste rock block through the ore in a cave mining operation and stability of rock mass blocks in open pit and underground mining was investigated.

Keywords: Jointed rock mass, Block, Computer program, Methods to determine Geometry of rock mass.



Shahrood University of Technology
Faculty of Mining Engineering, Petroleum and Geophysics

Development of 2D computer program to determine Geometry of rock mass blocks

Reza Yarahmadi

Supervisors:

Prof. Reza Kakaie

Dr. Hossein Mirzaei Nasirabad

Advisors:

Raheb Bagherpour

**Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of Master of Science (M.Sc.)
In
Mining engineering**

September 2012