



دانشگاه صنعتی شاهرود

دانشکده معدن و ژئوفیزیک

پایان نامه کارشناسی ارشد

موضوع :

# تهیه نرم افزار تحت ویندوز جهت شبیه سازی زمین آماری

استاد راهنما : دکتر رضا خالو کاکائی

امیر یوسفی ۸۱۱۸۱۰۴

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## چکیده

زمین بعنوان یکی از منابع تامین نیازهای بشر همواره مورد توجه بوده و شناخت منابع معدنی از دیر باز حائز اهمیت بوده است. همچنین با شکل گیری زندگی صنعتی این مقوله اهمیت بیشتری یافته است.

امروزه با توجه به کاهش حجم ذخائر غنی معدنی و اهمیت توجه به ذخائر عمقی و کم عیارتر عملیات اکتشافی از حساسیت بیشتری برخوردار شده اند بطوری که مطالعه و ترسیم هر چه نزدیکتر به واقعیت توزیع مکانی مقادیر در یک محدوده اکتشافی که باستناد نمونه برداری انجام می گیرد، قسمت عمده عملیات اکتشافی در بخش تحلیل داده ها و نتیجه گیری را به خود اختصاص داده است. در حال حاضر روشهای زمین آماری بهترین و تایید شده ترین روشها بدین منظور به حساب می آیند. این روشها عمدتاً شامل مجموعه عملیات تخمین و شبیه سازی می باشند که از آن جمله می توان به متد های مختلف تخمین بروش کریجینگ و شبیه سازیهای مرحله ای (اعم از گوسی و شاخص)، شبیه سازی انیلینگ و ... اشاره نمود. پایه و اساس این عملیات تابع وابستگی مکانی داده ها به یکدیگر یا وریوگرام می باشد. با وجودی که شناخت اصول و تئوری این روشها به سالیان اخیر باز نمی گردد، لیکن محاسباتی که هر کدام در بر می گیرند چنان حجیم و بعضاً پیچیده است که قبل از توسعه و کاربرد همه جانبه کامپیوتر امکان استفاده و توسعه روشهای زمین آماری ممکن نبود. به همین منظور نرم افزارهای مختلفی در این زمینه طراحی و ارائه شده است که استفاده از هر کدام از آنها با مشکلات و مسائل متعددی روبروست. هدف از انجام این پایان نامه تهیه و ارائه نرم افزاری است که در عین سادگی کاربرد، دارای محیط مناسب و طراحی استاندارد بوده و بتواند مراحل اجرای یک پروژه کامل زمین آماری را پوشش دهد. بدین منظور نرم افزار موسوم به Gridat با اتکا به اصول تئوریک و پایه زمین آمار و با استفاده از زبان برنامه نویسی C# و در محیط برنامه نویسی و طراحی Visual Studio .Net جهت اجرا در

سیستم عاملهای ۳۲ بیتی (محیط ویندوز) تهیه شد. نرم افزار مذکور دارای مجموعه الگوریتم هایی جهت نمایش آماری داده ها با هدف بررسی توزیع و آماده سازی آنها، واریوگرافی شامل محاسبه مقادیر واریوگرام و مدل سازی آن، مجموعه عملیات لازم جهت ایجاد شبکه توزیع مقادیر به روش تخمین و شبیه سازی زمین آماری و یک سری ابزار جهت بررسی مدل های ایجاد شده بروش تایید متقاطع می باشد.

اجرای عملیات ارائه شده در Gridat به آسانی ممکن می باشد و کاربر قادر است فایل های داده با فرمت های مختلف را بعنوان داده ورودی منظور و یا آن را مستقیما ایجاد نماید. همچنین نرم افزار توانایی ارائه نتایج عملیات مختلف زمین آماری را بصورت تصاویر و یا گراف داراست.

در پایان جهت اطمینان از صحت عملکرد نرم افزار، خروجی های آن با خروجی های مجموعه GSLIB90 مقایسه شده است که حاصل این مقایسه تایید درستی عملکرد الگوریتم های بکار گرفته شده در Gridat می باشد.

## فهرست مطالب

### فصل اول

#### مقدمه

۱

۲

۱-۱-مقدمه

۳

۲-۱- نرم افزارهای زمین آماری

۴

۳-۱-هدف از این پایان نامه

۵

۴-۱- سازماندهی پایان نامه

### فصل دوم

#### مقدمه ای بر زمین آمار و شبیه سازی زمین آماری

۶

۷

۱-۲- تعریف زمین آمار

۸

۲-۲- لزوم استفاده از زمین آمار

۹

۳-۲- تئوری زمین آمار

۱۰

۴-۲- وریوگرام

۱۰

۲-۴-۱- تعریف

۱۱

۲-۴-۲- رابطه وریوگرام و کوواریانس

۱۲

۳-۴-۲- اجزای وریوگرام

۱۴

۴-۴-۲- محاسبه وریوگرام

۱۶

۵-۴-۲- مدل های وریوگرام

۱۸

۴-۲-۶- عوامل تأثیر گذار در مدل وریوگرام

۲۰

۲-۵- کریجینگ

۲۰

۲-۵-۱- تعریف

۲۱

۲-۵-۲- کریجینگ ساده

۲۳

۳-۵-۲- کریجینگ معمولی

۲۴

۴-۵-۲- کریجینگ عام

۲۵

۵-۵-۲- کوکریجینگ

۲۵

۶-۵-۲- کریجینگ شاخص

۲۶	۷-۵-۲-آزمون نتایج کریجینگ
۲۸	۶-۲- شبیه سازی زمین آماری
۲۸	۱-۶-۲- تعریف
۲۹	۲-۶-۲- ارتباط تخمین و شبیه سازی
۳۱	۳-۶-۲- تئوری شبیه سازی
۳۲	۴-۶-۲- شبیه سازی گوسی مرحله ای
۳۳	۵-۶-۲- شبیه سازی شاخص مرحله ای
۳۵	۶-۶-۲- انیلینگ شبیه سازی شده
۳۸	<b>فصل سوم</b>
۳۹	<b>نرم افزار Gridat ، پیکره کلی برنامه و الگوریتم های اصلی</b>
۳۹	۱-۳- معرفی نرم افزار
۴۱	۲-۳- پیکره کلی برنامه
۴۵	۳-۳- الگوریتم های عملیاتی
۴۶	۱-۳-۳- الگوریتم نمایش نقاط نمونه برداری
۴۸	۲-۳-۳- الگوریتم بررسی آماری توزیع داده ها
۵۳	۳-۳-۳- الگوریتم واریوگرافی و مدل سازی وریوگرام
۶۳	۴-۳-۳- الگوریتم تخمین بروش کریجینگ
۶۷	۵-۳-۳- الگوریتم شبیه سازی مرحله ای
۷۱	۶-۳-۳- الگوریتم انیلینگ شبیه سازی شده
۷۴	۷-۳-۳- الگوریتم ترکیب شبکه ها
۷۷	<b>فصل چهارم</b>
۷۸	<b>نرم افزار Gridat ، نصب ، راه اندازی و روش اجرا</b>
۷۸	۱-۴- معرفی نرم افزار
۷۸	۲-۴- نصب نرم افزار
۸۰	۳-۴- قطعات اصلی و روش کلی اجرای Gridat
۸۴	۴-۴- منوها در Gridat
۸۵	۱-۴-۴- File
۸۵	۱-۱-۴-۴- گزینه New
۸۵	۲-۱-۴-۴- گزینه Open
۸۵	۳-۱-۴-۴- گزینه CloseAll
۸۵	۴-۱-۴-۴- گزینه Save
۸۶	۵-۱-۴-۴- گزینه Save As

۸۶	Export	گزینه ۶-۱-۴-۴
۸۶	Import	گزینه ۷-۱-۴-۴
۸۷	Page Setup	گزینه ۸-۱-۴-۴
۸۷	Print Preview	گزینه ۹-۱-۴-۴
۸۷	Print	گزینه ۱۰-۱-۴-۴
۸۷	Exit	گزینه ۱۱-۱-۴-۴
۸۸	Edit	منوی ۲-۴-۴
۸۸	Undo	گزینه ۱-۲-۴-۴
۸۸	Redo	گزینه ۲-۲-۴-۴
۸۸	Copy	گزینه ۳-۲-۴-۴
۸۸	Paste	گزینه ۴-۲-۴-۴
۸۸	Cut	گزینه ۵-۲-۴-۴
۸۸	Delete	گزینه ۶-۲-۴-۴
۸۹	Find	گزینه ۷-۲-۴-۴
۸۹	Find Next	گزینه ۸-۲-۴-۴
۸۹	Replace	گزینه ۹-۲-۴-۴
۸۹	Select All	گزینه ۱۰-۲-۴-۴
۹۰	Format	منوی ۳-۴-۴
۹۰	Command and Option	گزینه ۱-۳-۴-۴
۹۴	Formula	گزینه ۲-۳-۴-۴
۹۵	Chart	گزینه ۳-۳-۴-۴
۱۰۲	View	منوی ۴-۴-۴
۱۰۲	Action	منوی ۵-۴-۴
۱۰۳	Post	گزینه ۱-۵-۴-۴
۱۰۴	Distribution	گزینه ۲-۵-۴-۴
۱۰۵	Variogram	گزینه ۳-۵-۴-۴
۱۰۷	Kriging	گزینه ۴-۵-۴-۴
۱۱۰	Sequential Simulation	گزینه ۵-۵-۴-۴
۱۱۲	Simulated annealing	گزینه ۶-۵-۴-۴
۱۱۳	Multiple Grid	گزینه ۷-۵-۴-۴
۱۱۴	Window	منوی ۶-۴-۴
۱۱۴	Help	منوی ۷-۴-۴

۱۱۴	۵-۴- جعبه ابزار Gridat
۱۱۵	۶-۴- جستجو گر پروژه
۱۱۶	۷-۴- میله نشانگر وضعیت
۱۱۶	۸-۴- زمینه
۱۱۷	<b>فصل پنجم</b>
۱۱۸	<b>بررسی عملکرد نرم افزار Gridat</b>
۱۱۸	۱-۵- روش بررسی عملکرد Gridat
۱۲۰	۲-۵- خروجی های WinGSlib
۱۲۳	۳-۵- اجرای یک پروژه زمین آماری
۱۲۳	۱-۳-۵- ایجاد پروژه جدید در Gridat
۱۲۳	۲-۳-۵- بررسی فایل های داده و آماده سازی داده ها در Gridat
۱۲۹	۳-۳-۵- واریوگرافی و مدل سازی وریوگرام در Gridat
۱۳۱	۴-۳-۵- ایجاد شبکه مقادیر در Gridat
۱۳۱	۴-۳-۵- الف- روش تخمین
۱۳۵	۴-۳-۵- ب- روش شبیه سازی مرحله ای گوسی
۱۳۶	۴-۵- مقایسه دو نرم افزار
۱۳۸	۵-۵- دیگر ابزار Gridat
۱۴۵	<b>فصل ششم</b>
۱۴۶	<b>نتیجه گیری و پیشنهادات</b>
۱۴۸	<b>فهرست منابع و ماخذ</b>
۱۵۰	<b>پیوست</b>
۱۵۱	پیوست الف: کدهای برنامه
۱۵۱	پیوست الف-۱) کد برنامه الگوریتم نمایش نقاط نمونه برداری
۱۵۳	پیوست الف-۲) کد برنامه الگوریتم بررسی آماری توزیع داده ها
۱۵۸	پیوست الف-۳) کد برنامه الگوریتم واریوگرافی و مدل سازی وریوگرام
۱۶۷	پیوست الف-۴) کد برنامه الگوریتم تخمین بروش کریجینگ
۱۷۴	پیوست الف-۵) کد برنامه الگوریتم شبیه سازی مرحله ای
۱۸۱	پیوست الف-۶) کد برنامه الگوریتم انیلینگ شبیه سازی شده
۱۸۷	پیوست الف-۷) کد برنامه الگوریتم ترکیب شبکه ها
۱۹۱	پیوست ب: راهنمای واژگان لاتین



# فصل اول

## مقدمه

## فصل اول

### مقدمه

#### ۱-۱- مقدمه

مسلماً عملیات اکتشاف مواد معدنی طی سالیان اخیر ابعاد گسترده تری یافته است. کاهش ذخائر غنی معدنی از یک سو و افزایش تقاضا برای مواد اولیه از سوی دیگر نظر علوم اکتشاف را به سمت ذخائر با شاخص های اقتصادی ضعیفتر نظیر منابع با عیار پایین تر یا منابع عمقی سوق می دهد که این امر مستلزم دید علمی تر و بررسی های دقیق تر بر پایه اصول علمی - کاربردی در مراحل عملیات اکتشافی اعم از برداشتهای صحرایی و تحلیل داده ها و نتیجه گیری می باشد.

زمین آمار از جمله علوم کاربردی در زمینه تحلیل اکتشافی می باشد.

زمین آمار با ارائه شاخص وابستگی مکانی داده ها به نام وریوگرام، مقادیر نمونه برداری شده را بر اساس موقعیت مکانی شان در منطقه و فاصله نمونه ها از یکدیگر مورد تحلیل قرار می دهد که این ویژگی وجه تمایز زمین آمار با روشهایی است که هر نمونه را بطور مستقل از دیگر نمونه ها بررسی می کنند؛ نظیر روش های مبنی بر پایه آمار کلاسیک، می باشد.

وریوگرام در تخمین تصاویر نقاط نمونه برداری نشده منطقه و بررسی توزیع منطقه ای مقادیر بکار گرفته می شود که روش کریجینگ راهکاری جهت بکارگیری وریوگرام بدین منظور می باشد. علاوه بر این روشهای مختلف شبیه سازی خصوصاً شبیه سازی های مرحله ای بر پایه شاخص وابستگی مکانی داده ها یا همان وریوگرام انجام می گیرد.

با وجودی که بررسی کاربردی روشهای زمین آماری جهت تخمین و شبیه سازی به سالیان اخیر باز نمی گردد لیکن حجم بالای محاسبات علاوه بر وقت گیر بودن احتمال خطا و اشتباه را در اینگونه عملیات بالا خواهد برد و تا زمانی که استفاده از کامپیوتر و نرم افزار راه گشای حل اینگونه مسائل نشد، استفاده از این روشها مرسوم نمی بود. با اتکا به علم برنامه نویسی می توان محاسبات واریوگرافی، تخمین و شبیه سازی را به کامپیوتر واگذار نمود.

به طور کلی هر پروژه زمین آماری شامل مجموعه عملیات واریوگرافی، تخمین و شبیه سازی می باشد. بنابراین هر نرم افزاری که جهت اجرای پروژه های زمین آماری طراحی شود باید دارای حداقل گزینه های یاد شده باشد این موارد علاوه بر ابزاری جهت بررسی آماری توزیع مقادیر خواهد بود.

بررسی آماری توزیع مقادیر نمونه برداری شده قدم اول در هر پروژه می باشد و پس از آن مطالعه وابستگی مکانی داده ها نسبت به یکدیگر و یا احتمالاً نسبت موقعیت مکانی شان یا واریوگرافی، انجام می گیرد تخمین بروش کریجنیک و شبیه سازی، نتایج نهایی مورد انتظار می باشند که ارائه دهنده شبکه ای مقدار دهی شده است و بیانگر توزیع مکانی مقادیر در محیط خواهد بود و مقایسه شبکه های متفاوت منجر به بررسی های عدم قطعیت سنجی می گردد.

## ۱-۲- نرم افزارهای زمین آماری

جهت اجرای پروژه های زمین آماری نرم افزارهای مختلفی ارائه شده که استفاده از هر کدام با مسائل و مشکلات خاص خود همراه است، بعضی نرم افزارها با اهداف خاص و جهت استفاده در پروژه ای مشخص طراحی شده اند و سپس با اهداف جامع تر مورد توجه قرار گرفته اند. نظیر برنامه ای با نام Variowin که در سال ۱۹۹۵ بعنوان قسمتی از یک تز دکتری ارائه شد و صرفاً با اهداف واریوگرافی و مدل سازی واریوگرام بکار می رود [۲]. بعضی دیگر نیز به اندازه کافی قدیمی شده اند و عملاً کارآیی ندارند. مثل مجموعه ای با نام Geostatistical tool box (ویرایش ۱/۳ در سال ۱۹۹۰ و GeoEAS) [۲].

از دیگر نرم افزارهای موجود در زمین آمار می توان به مجموعه (gstat(1997، GSWin(version7) و Isatis اشاره نمود که دستیابی به آنها ملزم به صرف هزینه بوده و تهیه آنها

از عهده کاربران معمولی خارج است [۲]. بعنوان مثال نرم افزار GSWin که ویرایش ۷ آن اخیراً منتشر شده است شامل عملیات آنالیز داده ها، واریوگرافی، تخمین و شبیه سازی در محیط دو بعدی و سه بعدی و با استفاده از داده های چند متغیره می باشد ولی دریافت حداقل امکانات این مجموعه از طریق شبکه ملزم به صرف هزینه بالغ بر ۴۰۰۰ دلار خواهد بود.

از مجموعه نرم افزارهایی که در محاسبات زمین آماری معمول می باشند می توان به Surfer, Variowin, GS+, GSLIB (1992, 1998), SAS, S+, GSTAT اشاره نمود [۸].

مرسوم ترین نرم افزار زمین آماری در حال حاضر مجموعه ای موسوم به WinGSlib است که به عنوان یک نرم افزار با مقاصد زمین آماری نظیر واریوگرافی، کریجینگ و شبیه سازی شناخته می شود و در حقیقت رابط گرافیکی کاربر ویندوز با برنامه تحت سیستم عامل DOS با نام GSLIB می باشد. اجرای مجموعه برنامه های GSLIB که یک سری کدهای برنامه به زبان فرترن ۷۷ می باشد، در محیط DOS با ایجاد فایل های ورودی، فایل های تنظیم مقادیر (پارامتر) و فایل های خروجی برنامه امکان پذیر است. ایجاد فایل های پارامتری با فرمت ASCII و ذخیره سازی آنها در حافظه مهمترین و در عین حال مشکل ترین بخش کار با این مجموعه بشمار می رود و کوچکترین اشکال در نوشتن این فایل ها منجر به عدم اجرای صحیح الگوریتم خواهد شد [۱]. هر چند WinGSlib تا حد قابل قبولی مشکل را حل نمود ولی استفاده کنندگان از این مجموعه همچنان با مشکلاتی نظیر مدیریت فایل های پروژه و مشاهده خروجی مواجه اند.

### ۱-۳- هدف از این پایان نامه

علیرغم وجود نرم افزارهای متفاوت در زمینه زمین آمار، لیکن تعداد نرم افزارهایی که صرفاً با این اهداف طراحی شده باشد و کلیه مراحل یک پروژه زمین آماری را شامل شود انگشت شمار بوده و بعضاً دسترسی به آنها آسان نمی باشد. لذا طراحی و ارائه نرم افزاری که علاوه بر در دسترس بودن و متناسب بودن با انتظارات کاربران اهداف کلی پروژه های زمین آماری را بر پایه فرضیات اولیه زمین آمار ارائه دهد، لازم بنظر

می‌رسد. یک چنین نرم افزاری باید دارای محیط اجرایی شامل گزینه‌ها و تنظیمات مورد نیاز بنحوی باشد که کاربر را به سمت اجرای صحیح پروژه راهبری نماید.

هدف از انجام این پایان نامه طراحی و ارائه نرم افزاری برای انجام محاسبات زمین آماری و شبیه سازی است که در عین سادگی کاربرد، دارای محیط مناسب و طراحی استاندارد باشد.

#### ۱-۴- سازماندهی پایان نامه

برای دسترسی به هدف پایان نامه، ابتدا الگوریتم های مورد نظر طراحی و بعضاً در محیط برنامه نویسی ساده تر (نظیر ویژوال بیسیک) کد نویسی و آزمایش گردید. سپس نرم افزار موسوم به Gridat با استفاده از زبان برنامه نویسی C# تهیه شد و با داده های مختلف عملکرد آن مورد بررسی قرار گرفت. این مجموعه قابل اجرا در محیط سیستم عاملهای ۳۲ بیتی نظیر ویندوز ۹۸ و XP می باشد.

طی پایان نامه به جزئیات طراحی نرم افزار اشاره نشده و فقط الگوریتم های اصلی، عملکرد و نحوه اجرای آنها مورد توجه بوده است. سازماندهی فصول دیگر پایان نامه بدین ترتیب می باشد:

در فصل دوم به اصول تئوریک زمین آمار نظیر وریوگرام، تخمین و شبیه سازی که بیشتر در طراحی نرم افزار مورد توجه بوده اشاره شده و فصل سوم به معرفی و بررسی اجزای اصلی نرم افزار و ارتباط و عملکرد هر کدام از الگوریتم های اصلی اختصاص داده شده است. فصل چهارم پایان نامه به معرفی نرم افزار می پردازد. بررسی منوها و سایر اجزا و همچنین تشریح روش کار با نرم افزار نیز طی این فصل انجام می گیرد. فصل پنجم پایان نامه به مطالعه موردی و مقایسه نرم افزار با مجموعه WinGSlib اختصاص داده شده است. فصل ششم نیز شامل نتیجه گیری و پیشنهادات است. ضمیمه پایان نامه حاوی کدهای اصلی نرم افزار و واژه نامه انگلیسی با راهنمای صفحه می باشد.

## فصل دوم

مقدمه ای بر زمین آمار و شبیه سازی زمین آماری

## فصل دوم

### مقدمه ای بر زمین آمار و شبیه سازی زمین آماری

#### ۲-۱- تعریف زمین آمار

از یک نظر زمین آمار را می توان راهکاری جهت درونیایی و ارتباط دادن داده ها در یک محیط غیر متجانس دانست. لیکن این تصور تا حدودی ابتدایی و ساده انگارانه است. زیرا متد های درونیایی و برونایی مختلفی قبل از اینکه زمین آمار مطرح شود پذیرفته شده بودند که از جمله می توان روشهای عکس فاصله و روشهای آنالیز روند سطحی را نام برد [۹].

قبل از هر موردی باید خاطر نشان ساخت زمین آمار با دید خاص خود به داده ها می نگرد که ویژگی عمده آن اهمیت مکانی داده ها و همچنین ارتباط مقادیر داده ها با مختصات آنها است. در معنی کلی زمین آمار شامل کلیه تحلیلها و پیشگوئی های خاص دارای وابستگی های زمانی و مکانی می گردد. از جمله می توان به توزیع عیار فلزات در محیط، تخلخل سازندهای در برگیرنده سیال، درصد آلودگی محیط، تغییرات قیمت نفت، پیشگوئی های هوا شناسی و ... اشاره نمود.

هر چند خواستگاه علم زمین آمار علوم معدنی بوده است و پسوند geo درعنوان geostatistics یادآور علوم مرتبط با زمین شناسی می باشد، لیکن گستردگی این علم در شاخه های متفاوت علوم کاربردی انکار ناپذیر بوده و امروز زمین آمار نامی است که به مجموعه روشهای متعدد که به منظور تحلیل مقادیر وابسته با مکان یا زمان بکار می روند اطلاق می گردد [۹].

در زمین آمار نیز اکثراً با این نوع داده‌ها سروکار داریم و نیازمند ایجاد ارتباط و همبستگی یابی<sup>۱</sup> در مجموعه آنها هستیم. این نوع عملیات یعنی همبستگی یابی داده‌ها با وابستگی مکانی با نام آنالیز ساختاری<sup>۲</sup> و در نوع خاص آن با نام واریوگرافی<sup>۳</sup> شناخته می‌شود. بعد از آنالیز ساختاری داده‌ها، نقاط فاقد مقدار به روشهای متعدد از جمله تخمین<sup>۴</sup> و شبیه‌سازی<sup>۵</sup> مقدار دهی می‌گردد. به طور خلاصه می‌توان گفت مراحل اصلی یک مطالعه زمین آماری اکتشافی عبارتند از:

الف) تحلیل اکتشافی - زمین‌شناسی داده‌ها

ب) آنالیز ساختاری (واریوگرافی و مدل‌سازی وریوگرام)

ج) مدل‌سازی توزیع منطقه‌ای مقادیر (تخمین و شبیه‌سازی)

## ۲-۲- لزوم استفاده از زمین آمار

اما لزوم استفاده از روشهای زمین آماری در چیست؟ در حالی که روشهای دیگری به منظور همبستگی یابی مقادیر شبکه ارائه شده که با حجم کمتر محاسبات و تئوری ساده‌تر نسبت به روشهای زمین آماری تصاویر مجهول را محاسبه می‌کنند [۹]. از جمله این روشها می‌توان به روش مثلث بندی<sup>۶</sup> و عکس فاصله<sup>۷</sup> اشاره نمود. ایراد عمده این روشها در عبارت محاسبه مقادیر می‌باشد. اصولاً در علوم مرتبط با زمین با فاکتوری به نام ضریب اطمینان مواجه ایم. لذا مقادیر شبکه باید حدس زده شوند<sup>۸</sup> و مسلماً هر حدسی توأم با خطا خواهد بود و آن خطا همان واریانس منطقه‌ای مقادیر می‌باشد که توسط روشهای زمین آماری نظیر تخمین یا مطالعات عدم قطعیت سنجی<sup>۹</sup> ارائه می‌شود [۴] و [۱۰].

- 
- 1 - Correlation
  - 2 - Structural analysis
  - 3 - Variography
  - 4 - Estimation
  - 5 - Simulation
  - 6 - Triangulation
  - 7 - Inverse distance
  - 8 - Prediction
  - 9 - Uncertainty



در روشهایی مانند مثلث بندی یا عکس فاصله که برای تخمین مقادیر مجهول بکار می روند نتایج با قطعیت همراه است (بدین علت این روشها را با نام روشهای همبستگی یابی قطعی<sup>۱۰</sup> نام نهاده اند) که مسلماً خلاف واقع است. هر چند در روشهای همبستگی یابی قطعی نیز می توان مقادیری با عنوان خطا محاسبه نمود لیکن این امر مستلزم محاسبات پیچیده و خسته کننده است در حالی که روشی مانند روش تخمین کریجینگ که بر پایه زمین آمار و محاسبات واریوگرافی بنیان نهاده شده با ارائه فاکتوری با نام واریانس تخمین در هر نقطه شبکه، امکان بررسی کیفیت محاسبات را در اختیار کاربر قرار می دهد [۹].

از دیگر مزایای روشهای همبستگی یابی زمین آماری منظور نمودن انیزوتروپی<sup>۱۱</sup> و حذف اثر توزیع غیر یکنواخت نقاط نمونه برداری با بکارگیری الگوریتم به مراتب مختصر تر و ساده تر از روشهای همبستگی یابی قطعی می باشد.

البته محاسبات زمین آماری در مجموع وقت بیشتری می طلبد و نیاز به صرف زمان دارد. بعنوان مثال قسمت عمده عملیات را محاسبات واریوگرافی و مدل سازی وریوگرام به خود اختصاص می دهد، هر چند این زمان بری و صرف انرژی در ازای ارزش و کیفیت نتایج حاصل ناچیز و کوچک می نماید.

## ۲-۳- تئوری زمین آمار

آغاز بکارگیری زمین آمار به صورت نظام یافته و مستقل بواسطه تلاشهای جرج ماترون<sup>۱۲</sup> در منطقه معدنی اکول دس ماین<sup>۱۳</sup> بوده است [۵]. ماترون این روش را با نام تئوری متغیرهای ناحیه ای<sup>۱۴</sup> می شناخت [۹]. متغیر ناحیه ای به هر متغیری اطلاق می گردد که در محیط یا زمان دارای توزیع باشد. تئوری مذکور معتقد است اندازه گیری این متغیرها در محیط یا زمان می تواند منطبق بر یک تابع توزیع تصادفی باشد (مانند تابع توزیع لاگ نرمال) که این تابع خود ناشی از یک فعالیت اتفاقی، یک مجموعه شامل توزیع های تصادفی دیگر و یا یک فعالیت با تغییرات در

---

10 - Deterministic interpolation techniques

11 - Anisotropy

12 - George Matheron

13 - Ecole des mines

14 - Theory of regionalized variables

حوزه زمان یا مکان باشد و این تئوری بنیان و پایه علوم زمین آمار در معنای کاربردی را تشکیل داد [۹].

هر چند بعضی دانشمندان علوم زمین با فرضیه فوق مخالف می باشند. از دیدگاه آنها توزیع داده‌های مرتبط با زمین تصادفی نمی باشد. صرفاً به علت پیچیدگی فرآیند توزیع مقادیر و کمبود عمومی داده ها در مطالعات زمین شناسی تابع توزیع مشخصی نمی توان به آنها منطبق نمود. با وجودی که این نظریه می تواند صحیح باشد لیکن مدل‌های توزیع تصادفی که در مطالعات زمین آمار کاربردی بکار گرفته شده اند در عمل نتایج رضایت بخشی به همراه داشته اند و این یعنی که با اتکا به فرض توزیع تصادفی مقادیر می توان فقدان اطلاعات جهت بررسی تابع توزیع واقعی مقادیر را جبران نمود. بنابراین اندازه گیری های محدود زمین شناسی می تواند به یک تابع توزیع تصادفی انطباق داده شود که اساس محاسبات واریوگرافی، تخمین، شبیه سازی و ... و همچنین محاسبه میزان خطا بر اساس این فرض انجام گیرد [۹].

توجه شود که منطبق نمودن مقادیر به مدل توزیع تصادفی صرفاً یک روش است و جدای از اینکه به توزیع تصادفی مقادیر معتقدیم و یا اینکه داده های زمین شناسی را دارای قطعیت بدانیم مجاز به انتخاب این مدل بوده و در نهایت خطای عملیات را می توانیم محاسبه نماییم. رضایت بخش بودن نتایج حاصل از فرآیند فوق، خود اثبات این مدعاست.

## ۲-۴- وریوگرام

### ۲-۴-۱- تعریف

وریوگرام و همزاد آن کوواریانس پایه های اصلی محاسبات زمین آماری می باشند. وریوگرام به معنی همبستگی یابی و مدل سازی مقادیر دو متغیر وابسته به زمان یا مکان می باشد. اکثر افراد می دانند که دو مقدار نزدیک به هم در محیطی نظیر محیط برداشت نمونه هایی اکتشافی شباهت بیشتر با یکدیگر دارند تا دو نقطه ای که فاصله شان از هم بیشتر است. به بیان دیگر شباهت دو نقطه نزدیکتر به یکدیگر در مقایسه با دو نقطه دورتر، بیشتر خواهد بود و این موردی است که آمار کلاسیک بدان توجه نمی کند.

از نظر آمار کلاسیک دو توزیع که دارای میانگین و واریانس یکسان می باشند کاملاً مشابهند هر چند از نظر ابعاد مکانی و زمانی با هم اختلافاتی داشته باشند [۹].

زمین آمار به کاربر خود اجازه بررسی رابطه مقادیر با توجه به بعد فاصله ای آنها را می دهد که این رابطه را با نام وریوگرام می شناسد. مقادیر وریوگرام به منظور حدس زدن و مقدار دهی به نقاط نمونه برداری نشده حرف اول را می زند. یعنی مدل سازی وریوگرام پیش نیاز عملیات تخمین و شبیه سازی زمین آماری می باشد.

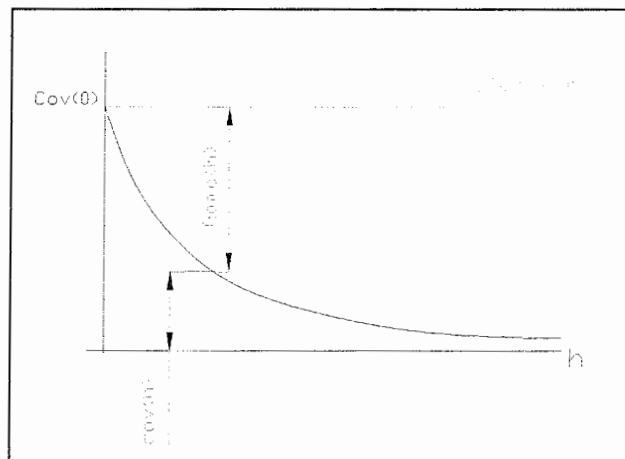
وریوگرام توسط دو پارامتر اساسی شناخته می شود. شعاع تأثیر<sup>۱۵</sup> که در فواصل کوچکتر از آن مقادیر با یکدیگر وابستگی دارند و آستانه<sup>۱۶</sup> که مقدار عددی وریوگرام بر حسب فاصله می باشد. جهت مدل سازی وریوگرام نیز مدل های استاندارد مورد توجه قرار می گیرد.

#### ۲-۴-۲- رابطه وریوگرام و کوواریانس

رابطه وریوگرام و کوواریانس در رابطه زیر خلاصه می گردد [۵]:

$$\gamma(h) = cov(0) - cov(h) \quad (1-2)$$

این رابطه را می توان به شکل (۱-۲) نشان داد.



شکل (۱-۲) رابطه وریوگرام و کوواریانس

با توجه به رابطه (۱-۲) مقادیر وریوگرام بر حسب کوواریانس و بالعکس قابل محاسبه خواهد بود. همانطور که از شکل (۱-۲) پیداست می توان گفت سیر مقادیر کوواریانس و وریوگرام در جهت عکس یکدیگر می باشد. یعنی بازای افزایش مقادیر وریوگرام، کاهش مقدار کوواریانس و بالعکس را انتظار داریم.

به طور کلی به ازای افزایش فاصله نمونه ها و کاهش تأثیر گذاری آنها بر یکدیگر، افزایش مقادیر وریوگرام و کاهش مقدار کوواریانس مشاهده می گردد (شکل ۱-۲).

در حالتی که تغییرات کوواریانس فاقد مقداری است ( $cov(h)=0$ ) تنها یک مقدار ثابت برای وریوگرام موجود خواهد بود. این وضعیت در حالتی پیش می آید که متغیرها هیچ گونه تأثیری بر یکدیگر نداشته و دارای ظرفیت پراکندگی بی نهایت باشند. در این شرایط محیط پایا<sup>۱۷</sup> است (شکل ۱-۲) [۹].

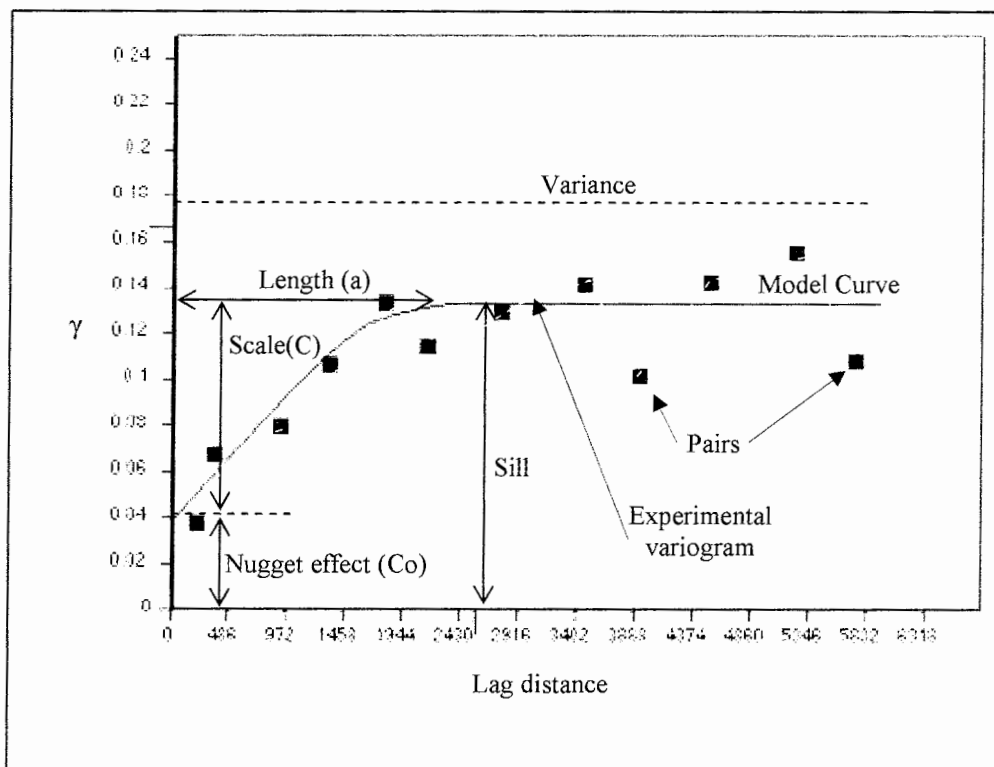
توجه به این نکته ضروری است که همواره تغییرات کوواریانس و بدنبال آن وریوگرام در حوزه عملکرد زمین آمار مورد توجه بوده و با اطمینان می توان گفت با شرایط پایا سروکار نداریم.

## ۲-۴-۳- اجزای وریوگرام

وریوگرام یک تابع سه بعدی می باشد (در سه جهت) که دارای دو متغیر مستقل امتداد<sup>۱۸</sup> و گام<sup>۱۹</sup> و یک متغیر وابسته  $\gamma(h, a)$  می باشد و هنگامی که وریوگرام برای محاسبات کریجینگ بکار گرفته شود مقادیر سقف دامنه تأثیر و مولفه تصادفی واریانس<sup>۲۰</sup> اهمیت پیدا می کنند. این مقادیر و مقادیر انیزوتروپی در هر نوع محاسبه زمین آماری بعنوان پارامترهای اساسی جهت ایجاد شبکه محاسبه وریوگرام، مدل سازی، تخمین و شبیه سازی های وابسته به وریوگرام مورد توجه قرار دارند. پس از محاسبه مقادیر وریوگرام یک مدل به نتایج محاسبات انطباق داده خواهد شد که عمده فاکتورهای قابل استخراج از مدل وریوگرام به شرح شکل (۲-۲) می باشد.

---

17 - Stationary  
18 - Direction  
19 - Lag  
20 - Nugget effect



شکل ۲-۲) مدل کلی وریوگرام و پارامترهای آن.

مولفه تصادفی واریانس ( $C_0$ ) به کیفیت نمونه‌برداری، میزان خطای موجود در برداشت و انتقال نمونه و مقادیر آن و همچنین تغییرات در مقیاس کوچکتر بستگی دارد [۱۵].

مولفه ساختاری واریانس  $C_1$  ( $C$ ): برابر با مقدار عمودی تغییرات در مدل وریوگرام می‌باشد. هر مدل برازنده شده به مقادیر وریوگرام می‌تواند ترکیبی از چند مدل مجزای متفاوت باشد که هر کدام مقدار  $C$  منحصر به خود را دارد [۱۵].

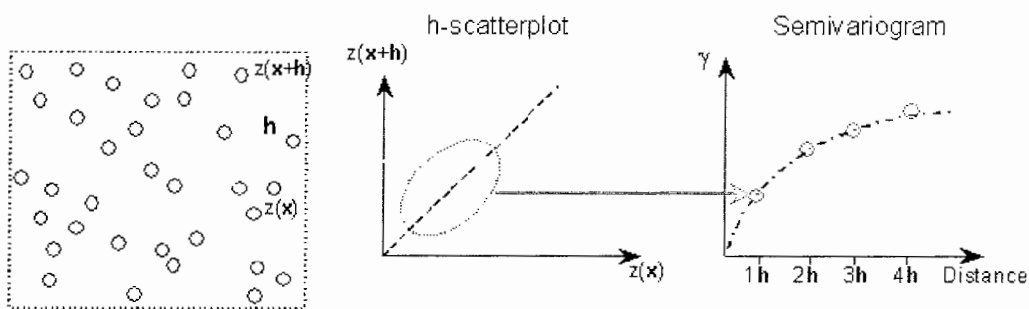
سقف: به مجموع مقادیر افقی برای هر کدام از مدل‌ها اطلاق می‌گردد [۱۵].

شعاع تأثیر ( $a$ ): حد تأثیر نمونه‌ها در بعد مسافت به یکدیگر. بعضی مدل‌ها فاقد این مقدار می‌باشند زیرا هیچ‌گاه مقادیر وریوگرام ثابت نخواهند شد [۱۵].

#### ۲-۴-۴- محاسبه وریوگرام

محاسبات مربوط به مقادیر و اجزای وریوگرام به بیان مختصر و ساده چنین است [۹]:

ابتدا یک مقدار بر حسب مقادیر واحد مختصات (مثلاً ۱۰۰ متر) بعنوان واحد گام تعریف می شود سپس توان دوم مقدار اختلاف هر دو نمونه ای که دارای فاصله ای برابر با گام می باشند محاسبه می گردد. میانگین این اختلافات مقدار وریوگرام به ازای واحد گام (۱۰۰ متر) را بدست می دهد. این عملیات برای مقادیر دو برابر گام تا  $n$  برابر گام (۲۰۰، ۳۰۰، ۴۰۰ و ... متر) محاسبه می گردد و مقادیر وریوگرام نسبت به فاصله نمونه ها از یکدیگر ترسیم خواهد شد. (شکل ۳-۲)



(شکل ۳-۲) رویه محاسبه مقادیر وریوگرام تجربی [۹]

مجموعه عملیات فوق به بیان ریاضی چنین است [۱۰]:

$$2\gamma(h) = \frac{1}{N(h)} \sum_{N(h)} [Z(u) - Z(u+h)]^2 \quad (2-2)$$

که

$2\gamma(h)$  مقدار وریوگرام برای دو نمونه فاصله  $h$  از یکدیگر،

$N(h)$  تعداد جفت نمونه ها با فاصله  $h$  از یکدیگر،

$Z(u)$  مقدار متغیر نمونه با موقعیت مکانی  $u$  می باشد.

که در محاسبات زمین آماری به نصف مقادیر وریوگرام ( $\gamma(h)$ ) به عنوان مقدار وریوگرام استناد می گردد.

به طور معمول با افزایش ضریب گام یا افزایش فاصله نمونه ها، مقدار وریوگرام افزایش می یابد که از شباهت و وابستگی بیشتر مقادیر نزدیکتر به هم ناشی می گردد. با گذر مقادیر گام از شعاع تأثیر مقدار وریوگرام برابر با سقف وریوگرام خواهد شد و نمونه ها با فاصله بیشتر از این هیچ

گونه وابستگی به یکدیگر نخواهند داشت. لذا محاسبه مقدار وریوگرام دیگر لزومی نخواهد داشت. روش فوق منجر به ترسیم گراف وریوگرام تجربی<sup>۲۲</sup> یا به بیان ساده تر وریوگرام خواهد شد مجموعه عملیات فوق را واریوگرافی می نامند.

روش یاد شده، روش کلاسیک و کلی جهت محاسبات واریوگرافی است. در عمل و در شرایطی که نمونه برداری در شبکه منظم انجام نگرفته ممکن است یافتن چند جفت نمونه که دارای فاصله برابر با گام باشد ممکن نبوده و اختصاص یک مقدار ثابت به عنوان گام عملی نباشد. همچنین در روش فوق به جهت انتخاب نمونه توجه نشده است، بعبارت دیگر بدون توجه به اینکه ممکن است تأثیر پذیری نمونه ها از یکدیگر در جهات مختلف تفاوت داشته باشد (انیزوتروپی محیط) جفت نمونه انتخاب و در محاسبات واریوگرافی بکار گرفته می شوند.

راهکاری که بدین منظور پیشنهاد شده انتخاب نمونه ها براساس موقعیت مکانی آنها نسبت به یکدیگر با منظور نمودن حدود مجاز نوسان از فاصله و امتداد ثابت می باشد (شکل ۲-۴) بعبارت دیگر جفت نمونه های انتخاب شده بازای هر کدام از ضرایب گام دارای فاصله ای برابر با گام، کمتر یا بیشتر از گام خواهند بود و در عین حال امتداد آنها نسبت به یکدیگر دارای مقادیر کمتر، برابر یا بیشتر از مقدار مشخص شده خواهد بود، به شرط اینکه این تغییرات از حدود مجاز نوسان فاصله<sup>۲۳</sup> و حدود مجاز نوسان امتداد<sup>۲۴</sup> تجاوز ننماید.

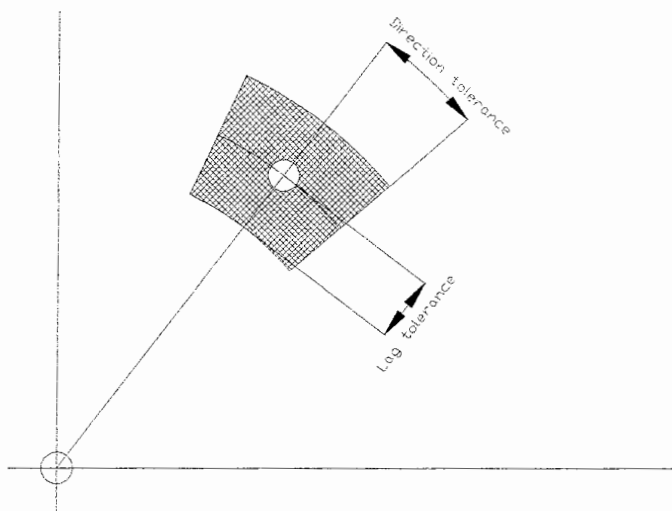
حدود مجاز نوسان فاصله از  $\pm 0/5$  برابر گام بیشتر نمی باشد و تغییرات مجاز امتداد از صفر درجه تا حالت تمام صفحه (۱۸۰ درجه) قابل قبول است. مورد اخیر در برگیرنده کلیه نقاط موجود در حدود مجاز فاصله ای خواهد بود.

---

22 - Experimental variogram

23 - Lag tolerance

24 - Direction tolerance



شکل (۴-۲) نمایش حدود مجاز نوسان فاصله ای و امتدادی

#### ۲-۴-۵- مدل های وریوگرام

استخراج اجزای وریوگرام (نظیر شعاع تاثیر، مولفه تصادفی و ...) از نتایج واریوگرافی متسلازم انطباق مدل به مقادیر وریوگرام می باشد. شکل (۲-۲) که پیشتر آورده شد نشان دهنده مدل مرسوم کروی می باشد. اکثراً در عملیات اکتشافی تمایل داریم وریوگرام را با این مدل منطبق سازیم. عملیات واریوگرافی به نحوی که گراف حاصل به انطباق کامل یک مدل استاندارد منجر شود مستلزم تجربه و تلاش بیشتر می باشد و کاربران زمین آمار نکات خاصی را دریافتن مدل‌های وریوگرام مد نظر قرار می دهند.

از جمله روشهایی که در هر چه نزدیکتر ساختن نتایج واریوگرافی به مدل استاندارد مد

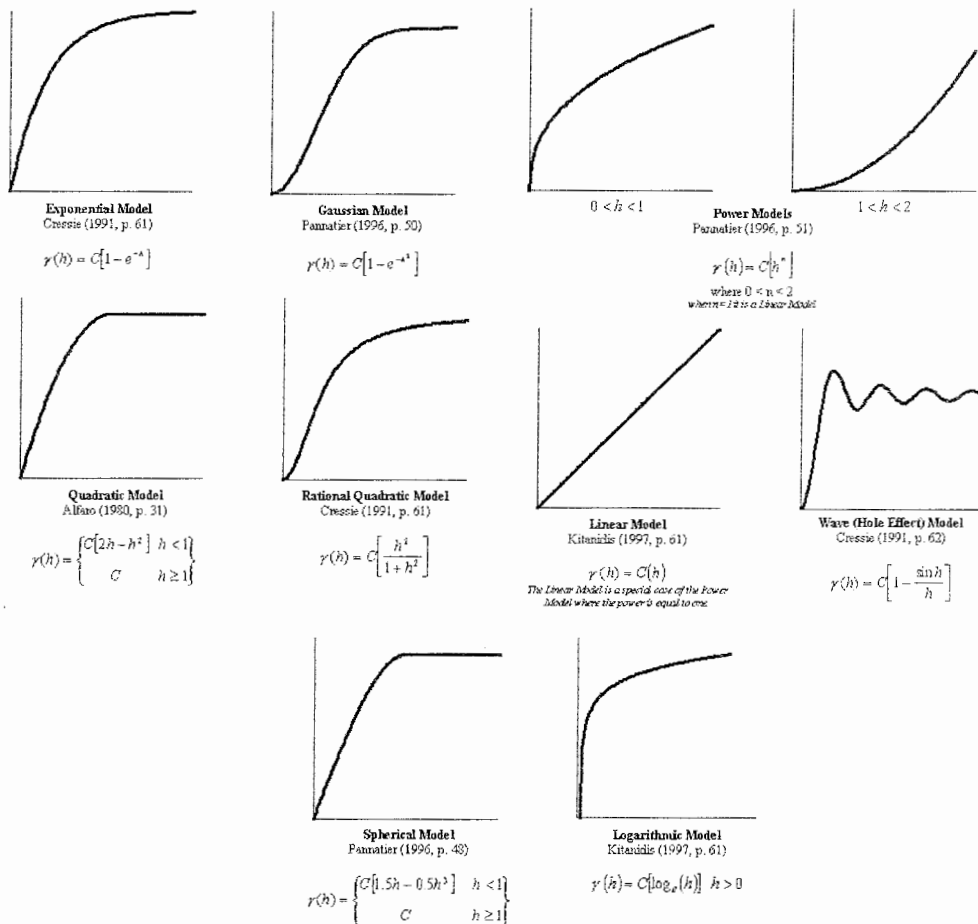
نظر قرار می گیرد می توان به موارد زیر اشاره نمود [۹].

- ✓ بررسی وجود تعداد کافی جفت نمونه (۳۰ تا ۵۰ جفت).
- ✓ انتخاب حدود مجاز نوسان برای گام و امتداد به نحوی که به تعداد کافی جفت نمونه عاید گردد.
- ✓ حذف داده های پرت.



✓ افزایش حدود تغییرات امتداد حتی در حالت تمام صفحه (۱۸۰ درجه). زیرا در نظر گرفتن زوایای کوچک به منظور دخالت دادن ساختار شبکه اکتشافی و انیزوتروپی شبکه در حالتی که واریوگرافی بدرستی انجام نمی گیرد، امری بی معناست.

✓ نرمال نمودن مقادیر در حالتی که توزیع احتمال دارای چولگی مثبت است.



شکل ۲-۵) بعضی مدل‌های استاندارد وریوگرام.

بعضی مدل‌های مرسوم وریوگرام که در شکل (۲-۵) نشان داده شده بقرار زیر می باشد:

- مدل خطی<sup>۲۵</sup>
- مدل توانی<sup>۲۶</sup>

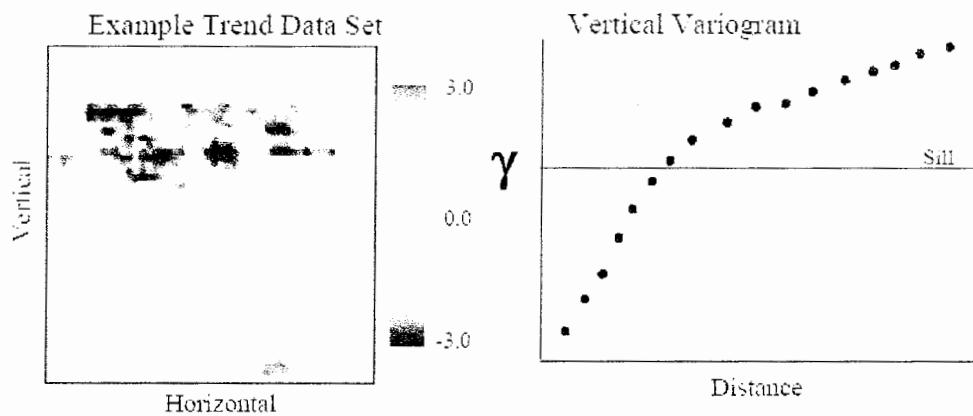
25 - Linear model  
26 - Power model

- مدل لوگاریتمی<sup>۲۷</sup>
- مدل گوسی<sup>۲۸</sup>
- مدل کروی<sup>۲۹</sup>
- مدل نمایی<sup>۳۰</sup>
- مدل چند جمله ای<sup>۳۱</sup> و
- مدل سینوسی<sup>۳۲</sup>.

#### ۴-۲-۶- عوامل تأثیر گذار در مدل وریوگرام

از عوامل تأثیر گذار در مدل وریوگرام می توان به موارد زیر اشاره نمود [۱۰]:

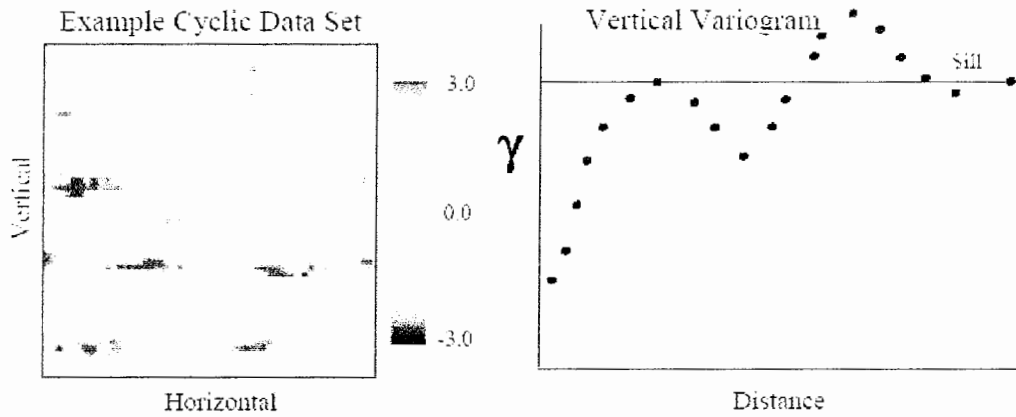
روند<sup>۳۳</sup>: که باعث وجود شیب عمومی در مدل وریوگرام میگردد بدین معنی که به جای رسیدن وریوگرام به سقف ثابت، نمودار به شیب ثابت که ناشی از روند عمومی منطقه است خواهد رسید (شکل ۲-۶).



شکل ۲-۶) نمایش عمومی روند و تأثیر آن بر وریوگرام [۱۰].

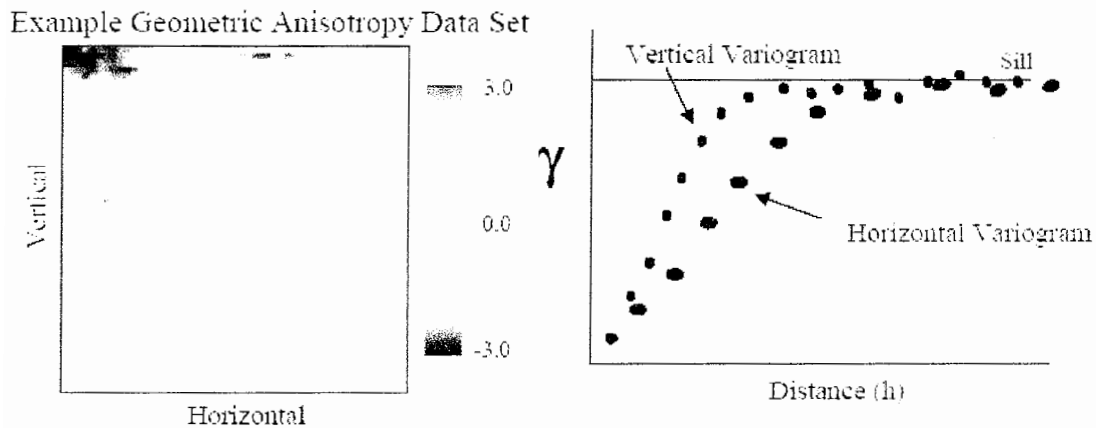
- 27 - Logarithmic model
- 28 - Gaussian model
- 29 - Spherical model
- 30 - Exponential model
- 31 - Quadratic model
- 32 - Wave(Hole effect) model
- 33 - Trend

تناوب<sup>۳۴</sup>: که عبارت از تناوب در تغییرات مقادیر حاصل از نمونه برداری می باشد. این عامل می تواند ناشی از ساختار خاص زمین شناسی منطقه یا محدودیت نمونه برداری باشد که به دنباله وریوگرام شکل سینوسی می دهد (شکل ۷-۲).



شکل ۷-۲) نمایش عمومی تناوب و تاثیر آن بر وریوگرام [۱۰].

انیزوتروپی هندسی<sup>۳۵</sup>: که باعث تغییر شعاع تأثیر در جهات متفاوت می گردد. (شکل ۸-۲).



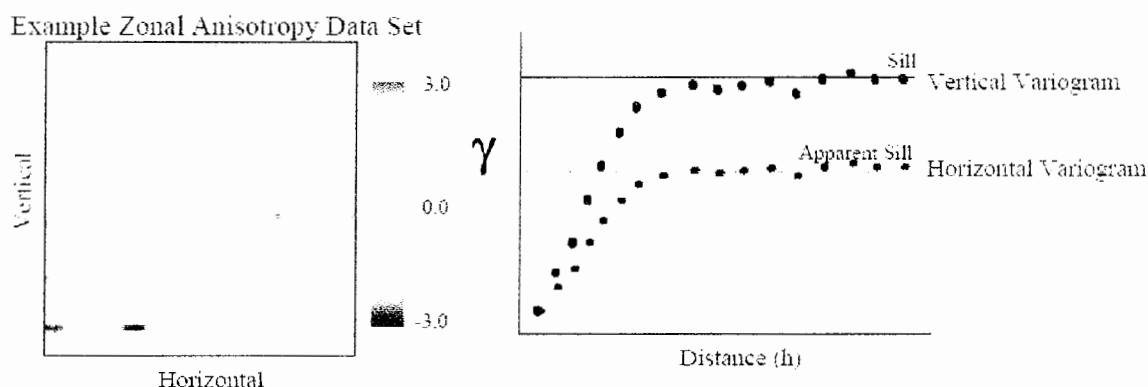
شکل ۸-۲) نمایش عمومی انیزوتروپی هندسی و تاثیر آن بر وریوگرام [۱۰].

34 - Cyclicity

35 - Geometric anisotropy

انیزوتروپی منطقه ای<sup>۳۶</sup>: که باعث تغییر در سقف وریوگرام در جهات متفاوت می گردد

(شکل ۹-۲).



شکل ۹-۲) نمایش عمومی انیزوتروپی منطقه ای و تاثیر آن بر وریوگرام [۱۰].

## ۲-۵- کریجینگ<sup>۳۷</sup>

### ۲-۵-۱- تعریف

کریجینگ یک روش تخمین زمین آماری می باشد. این روش بمنظور تخمین مقدار یک نقطه مجهول از ترکیب خطی مقادیر اولیه (نقاط دارای مقادیر اندازه گیری شده) حول آن استفاده می نماید. در این روش به هر کدام از نقاط دارای مقدار ضریب وزنی اختصاص داده می شود و این ضرایب طوری محاسبه می شود که واریانس تخمین حداقل مقدار ممکن باشد [۹].

نامگذاری روش به نام کریجینگ برگرفته از نام د. ج. کریج<sup>۳۸</sup> می باشد. او یک مهندس معدن در کشور آفریقای جنوبی بود که این روش را جهت تخمین دقیقتر مشخصات ذخائر بسط و توسعه داد. از آن پس روش کریجینگ به عنوان روشی مفید و مطرح در مطالعات زمین آماری مورد توجه قرار گرفت و روشها و شیوه های متفاوتی از آن ارائه شد [۶].

از جمله می توان به موارد زیر اشاره نمود: کریجینگ معمولی<sup>۳۹</sup>.

✓ کریجینگ ساده<sup>۴۰</sup>.

36 - Zonal anisotropy  
37 - Kriging  
38 - D. G. Krig  
39 - Ordinary Kriging

✓ کریجینگ عام<sup>۴۱</sup>.

✓ کریجینگ IRF-K

✓ کریجینگ شاخص<sup>۴۲</sup>

## ۲-۵-۲- کریجینگ ساده

قدم اول در تخمین بروش کریجینگ معمولی انجام واریوگرافی با استفاده از مقادیر نقاط نمونه برداری (نقاط با مقدار اولیه) و انطباق مدل وریوگرام می باشد.

با استفاده از مدل وریوگرام می توان ضرایب مورد استفاده در تخمین به روش کریجینگ ساده را حساب نمود. ضرایب مذکور در تساوی به شکل زیر به کار گرفته خواهند شد [۱۰].

$$Z(u_x) = \sum_{i=1}^n \lambda_i Z(u_i) \quad (3-2)$$

که  $n$  تعداد نقاط نمونه برداری شد (دارای مقدار اولیه) و  $Z(u_i)$  مقدار هر کدام از این نقاط و  $\lambda_i$  ها ضرایب وزنی می باشند.

بعنوان مثال تخمین مقدار مجهول  $p_0$  که در محیط شامل نقاط مقدار دار  $p_1$ ,  $p_2$  و  $p_3$  واقع شده برابر است با :

$$p_0 = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3$$

رابطه مذکور مشابه تساوی مورد استفاده در روشهای دیگر نظیر عکس فاصله می باشد و تفاوت اصلی در محاسبه ضرایب وزنی است. ضرایب بکار گرفته شده در تخمین بروش کریجینگ بر پایه مدل وریوگرام محاسبه می گردد.

در مثال فوق رویه محاسبه ضرایب  $\lambda_i$  که به نام ضرایب کریجینگ شناخته می شود از دستگاه معادلات بدین شکل حاصل خواهند شد [۱۵]:

$$\gamma(1,0) = \lambda_1 \gamma(1,1) + \lambda_2 \gamma(1,2) + \lambda_3 \gamma(1,3)$$

$$\gamma(2,0) = \lambda_1 \gamma(2,1) + \lambda_2 \gamma(2,2) + \lambda_3 \gamma(2,3)$$

$$\gamma(3,0) = \lambda_1 \gamma(3,1) + \lambda_2 \gamma(3,2) + \lambda_3 \gamma(3,3)$$

40 - Simple Kriging

41 - Universal Kriging

42 - Indicator Kriging

$$\sum_j \lambda_j \gamma(u_i, u_j) = 2\gamma(u, u_i) \quad i=1, 2, \dots, n \quad (7-2)$$

عبارت فوق یکدستگاه  $n$  معادله و  $n$  مجهول می باشد که معادل دستگاه معادلات رابطه (۴-۲) است و مقادیر مجهول ( $\lambda_i$  ها) ضرایب مورد استفاده در روش تخمینی با نام کریجینگ ساده خواهند بود. با محاسبه ضرایب کریجینگ مقدار پارامتر مورد مطالعه در نقطه مجهول برابر خواهد بود با:

$$Z^*(u) = m + \sum_{i=1}^n (\lambda_i Z(u_i) - m) \quad (8-2)$$

که  $m$  میانگین عیار بوده و مستقل از مختصات است. (شرایط پایایی محیط)،  $\lambda_i$  اوزان کریجینگ ساده و  $Z(u_i)$  مقدار عبارت در نقطه معادل موقعیت مکانی  $u_i$  می باشد. معادله (۸-۲) اصطلاحاً تخمینگر کریجینگ نامیده می شود.

با استفاده معادله (۶-۲) و با محاسبه ضرایب کریجینگ می توان خطای تخمین در هر نقطه از شبکه را محاسبه نمود و این از ویژگیهای شاخص روش کریجینگ می باشد. واریانس تخمین در هر نقطه نیز تابعی است از فاصله نقطه مورد نظر از نقاط دارای مقدار اولیه اطراف آن که در نهایت تساوی زیر را می توان نتیجه گیری نمود [۱۰].

$$\sigma_{est}^2 = \lambda_1 \cdot \gamma(1,0) + \lambda_2 \cdot \gamma(2,1) + \dots = \sum \lambda_i \cdot \gamma(i,0) \quad (9-2)$$

### ۳-۵-۲- کریجینگ معمولی

در شرایطی که مقدار میانگین عیار مجهول باشد به روش کریجینگ معمولی متوسل می شویم به شرط آنکه میانگین عیار مستقل از مختصات باشد. وابستگی عیار یا واریانس به مکان موجب روند در مدل وریوگرام می گردد [۱۰].

اگر مشابه معادلات کریجینگ ساده برای این حالت هم انجام گیرد و در پایان واریانس تخمین محاسبه شود بمنظور مینیموم کردن مقدار  $\sigma_{est}^2$  و بهینه سازی مقادیر ضرایب کریجینگ ضریب لاکرانژ ( $\mu$ ) در محاسبات ظاهر خواهد شد و ماتریس معادلات به شکل زیر در خواهد آمد [۱۰].

$$\begin{bmatrix} \gamma(1,1) & \gamma(1,2) & \gamma(1,3) & 1 \\ \gamma(2,1) & \gamma(2,2) & \gamma(2,3) & 1 \\ \gamma(3,1) & \gamma(3,2) & \gamma(3,3) & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ -\mu \end{bmatrix} = \begin{bmatrix} \gamma(0,1) \\ \gamma(0,2) \\ \gamma(0,3) \\ 1 \end{bmatrix}$$

در معادلات فوق به جای  $\gamma(i,j)$  از  $c(i,j)$  (کوواریانس) هم میتوان استفاده نمود.

## ۲-۵-۴- کریجینگ عام

این روش جهت اعمال کریجینگ در محیط های غیر پایا (روند دار) بکار می رود. به عبارت دیگر روش کلی کریجینگ که بدون توجه به پایایی و یا عدم پایایی محیط می توان بکار برد روش کریجینگ عام می باشد.

توضیح این نکته ضروری بنظر می رسد که منظور از پایایی محیط عدم وابستگی مقادیر میانگین و واریانس داده ها به مکان می باشد. این مورد که قبلاً نیز بدان اشاره شد در شرایط ظرفیت پراکندگی بی نهایت متغیرها بوجود می آید که مقادیر کوواریانس ثابت بوده و در زمین آمار کاربردی مطلقاً با این حالت سروکار نخواهیم داشت [۹].

در زمین آمار حالت خاصی از پایایی محیط براساس فرضیه ذاتی منظور می گردد. طبق این فرض محیط پایا محیطی است که مقادیر نمونه ها از مختصاتشان تأثیر نمی پذیرد و صرفاً تأثیر پذیری نمونه ها از یکدیگر و براساس فاصله شان از هم می باشد. در مقابل در محیط غیر پایا نمونه ها علاوه بر تأثیر پذیری از یکدیگر از مختصات نیز تأثیر پذیرفته و لذا مقادیر میانگین و واریانس منطقه ای تابعی از مختصات می باشد [۹].

در روشهای کریجینگ معمولی و ساده که پیشتر به آنها اشاره شد محیط پایا فرض شد و از وجود روند احتمالی صرف نظر شده بود. لیکن در صورت اهمیت منظور نمودن روند که به واسطه عدم پایایی یا وابستگی مکانی مقادیر حاصل می گردد در تخمین از روش کریجینگ عام استفاده خواهد شد.

در روش کریجینگ عام مدل ساخته شده برای وریوگرام به سقف ثابت میل نخواهد کرد

(نظیر مدل توانی) [۱۴].

در این حالت ابتدا معادله بنیادی توزیع عیار در منطقه تعیین می گردد و در نهایت در ماتریس محاسبه ضرایب کریجینگ به تعداد متغیرهای این معادله ضرایب لاکرانژ ظاهر خواهد شد که با محاسبه این مقادیر علاوه بر ضرایب کریجینگ مقدار و واریانس تخمین در هر نقطه محاسبه خواهد شد [۱۴].

## ۲-۵-۵- کوکریجینگ

در مطالعات اکتشافی بعضاً ممکن است با دو سری داده متفاوت از یک نوع سروکار داشته باشیم مثلاً در مطالعات هیدروژئولوژی ممکن است نفوذپذیری لایه ها با استفاده از نمونه برداری مستقیم (داده های اولیه) در بعضی نقاط و با استفاده از روشهای ژئوفیزیکی (داده های ثانویه) بدست آمده باشد. در این صورت می توان روش کوکریجینگ را جهت دستیابی به نتایج ارزنده تر اعمال نمود. بدین منظور لازم است علاوه بر مدل سازی وریوگرام برای داده های اولیه و ثانویه مدل سازی وریوگرام توام<sup>۴۳</sup> بین داده های اولیه و ثانویه نیز انجام گیرد [۹].

در مواردی که با کمبود نقاط نمونه برداری مواجه ایم با استفاده از دو یا چند پارامتر اندازه گیری شده طی نمونه برداری و آنالیز، و اعمال روش کوکریجینگ تا حدودی مسئله کمبود داده جبران می شود. از نظر تئوریک این روش به روشهای معمول کریجینگ ارجحیت دارد. لیکن مشکلاتی نظیر پیچیدگی محاسبات و مشکلات مدل سازی وریوگرام چند متغیره محدودیتهای اصلی استفاده از این روش می باشند [۹].

## ۲-۵-۶- کریجینگ شاخص

روش کریجینگ شاخص نوعی دیگر از روش تخمین شبکه می باشد. در این روش مقادیر اولیه داده ها تبدیل به مقادیر صفر و یک می شود. بدین ترتیب که نقاط دارای مقدار اولیه بزرگتر از یک حد آستانه ای خاص برابر با یک و مابقی برابر با صفر قرار داده می شود. سپس کریجینگ معمولی بر این اساس انجام می گیرد و خروجی نیز شبکه ای خواهد بود که مقادیر آن صفر تا یک است و بیانگر احتمال وقوع حد آستانه ای مشخص شده در هر نقطه می باشد.



موارد اصلی استفاده از این روش عبارتند از [۶]:

محاسبه سطح اعتماد.

حذف اثر مقادیری خارج از رده.

مدل سازی جوامع آماری مختلف.

طبقه بندی داده ها.

تابع توزیع منطقه ای احتمال بازای حدود آستانه ای مشخص.

## ۲-۵-۷- آزمون نتایج کریجینگ

مدل وریوگرام تأثیرگذارترین عامل در صحت و کیفیت نتایج تخمین بروش کریجینگ می باشد. افزایش شعاع تأثیر مدل باعث هموارتر شدن شبکه و همبستگی یابی بهتر مقادیر می گردد. همچنین مدل وریوگرام در شکل نهایی توزیع منطقه ای مقادیر تأثیر گذار است. بعنوان مثال مدل گوسی نسبت به مدلهای دیگر اشکال هموارتری بدست می دهد.

البته هموار سازی نتایج نهایی لزوماً کیفیت بهتر نتایج را معنی نمی دهد، کما اینکه با توسل به روشهای شبیه سازی سعی در حذف اثر هموارسازی را داریم.

افزایش مقادیر سقف مدل وریوگرام نیز به افزایش واریانس تخمین منجر می گردد.

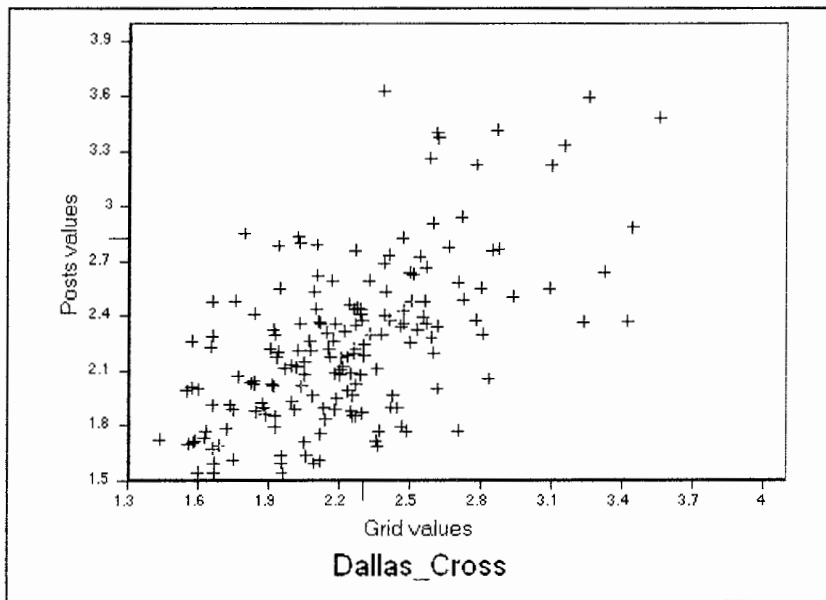
با توجه به موارد یاد شده می توان گفت در صورتی که کیفیت نتایج تخمین رضایت بخش نباشد و مقادیر از صحت لازم برخوردار نباشند، نیاز به مدل سازی مجدد وریوگرام محتمل ترین حالت خواهد بود.

روش مرسوم جهت آزمون نتایج تخمین، روشی با نام تأیید متقاطع<sup>۴۴</sup> می باشد [۹]. در این روش هر یک از نقاط معلوم بعنوان مجهول فرض شده و مقدار آن با استفاده از نقاط دیگر تخمین زده می شود. این کار برای کلیه نقاط معلوم انجام می شود در نهایت برای هر نقطه نمونه برداری شده دو سری داده شامل دادهای اولیه یا همان مقادیر نمونه برداری شده و مقادیر تخمین زده شده همین نقاط وجود خواهد داشت. با رسم این دو سری داده در یک دستگاه مختصات دو بعدی می توان به وضعیت درستی مدل وریوگرام و از آنجا صحت نتایج تخمین پی برد. واضح است

انتظار داریم حاصل یک خط راست با شیب ۱ باشد. ضریب رگرسیون این خط (شکل ۲-۱۰) مقداری عددی است که بیانگر کیفیت عملیات تخمین می باشد. در صورت رضایت بخش نبودن نتایج عملیات فوق، نظر کاربر به موارد چندی جلب می شود از جمله اینکه :

- ✓ آیا شعاع جستجوی نقاط به اندازه کافی وسعت داشته است ؟
- ✓ آیا مقادیر خارج از حد تخمین<sup>۴۵</sup> در نتایج پدیدار گشته است ؟
- ✓ آیا در محیط روند وجود دارد ؟
- ✓ و از همه مهمتر آیا مدل سازی وریوگرام بدرستی انجام شده و مقادیر صحیح پارامترهای واریوگرافی استخراج شده است ؟

توجه به این نکته ضروری است که بدست آوردن خط با ضریب رگرسیون ۱ در روش تأیید متقاطع، لزوماً تأییدکننده کیفیت واریوگرافی و صحت مدل وریوگرام نمی باشد.



شکل ۲-۱۰) آزمون نتایج کریجینگ بروش تأیید متقاطع

۴۴- مقادیر خارج از حد تخمین ناشی از افزایش بیش از اندازه و غیر واقعی ضرایب کریجینگ برای بعضی نقاط به وجود می آید. این وضعیت ممکن است بعلت فرض نامحدود بودن قلمرو تخمین شرایط خاص زمین شناسی نظیر محدود شدن سازندهای زمین شناسی توسط عوامل زمین ساختی و تکتونیکی و ... بوجود آید [۱۴].

## ۲-۶- شبیه سازی<sup>۴۶</sup> زمین آماری

### ۲-۶-۱- تعریف

معمولاً اطلاعات موجود در هر سیستم اندازه گیری و محاسبه مقادیر به سه دسته اصلی قابل تفکیک می باشند. بعضی اطلاعات جوابگوی یک سری نیازهای اولیه هستند. این داده‌ها به صورت مستقیم و طی برداشتهای صحرایی عاید می گردند (مانند مقادیر پارامترهای هیدرولیکی یک مخزن) [۳].

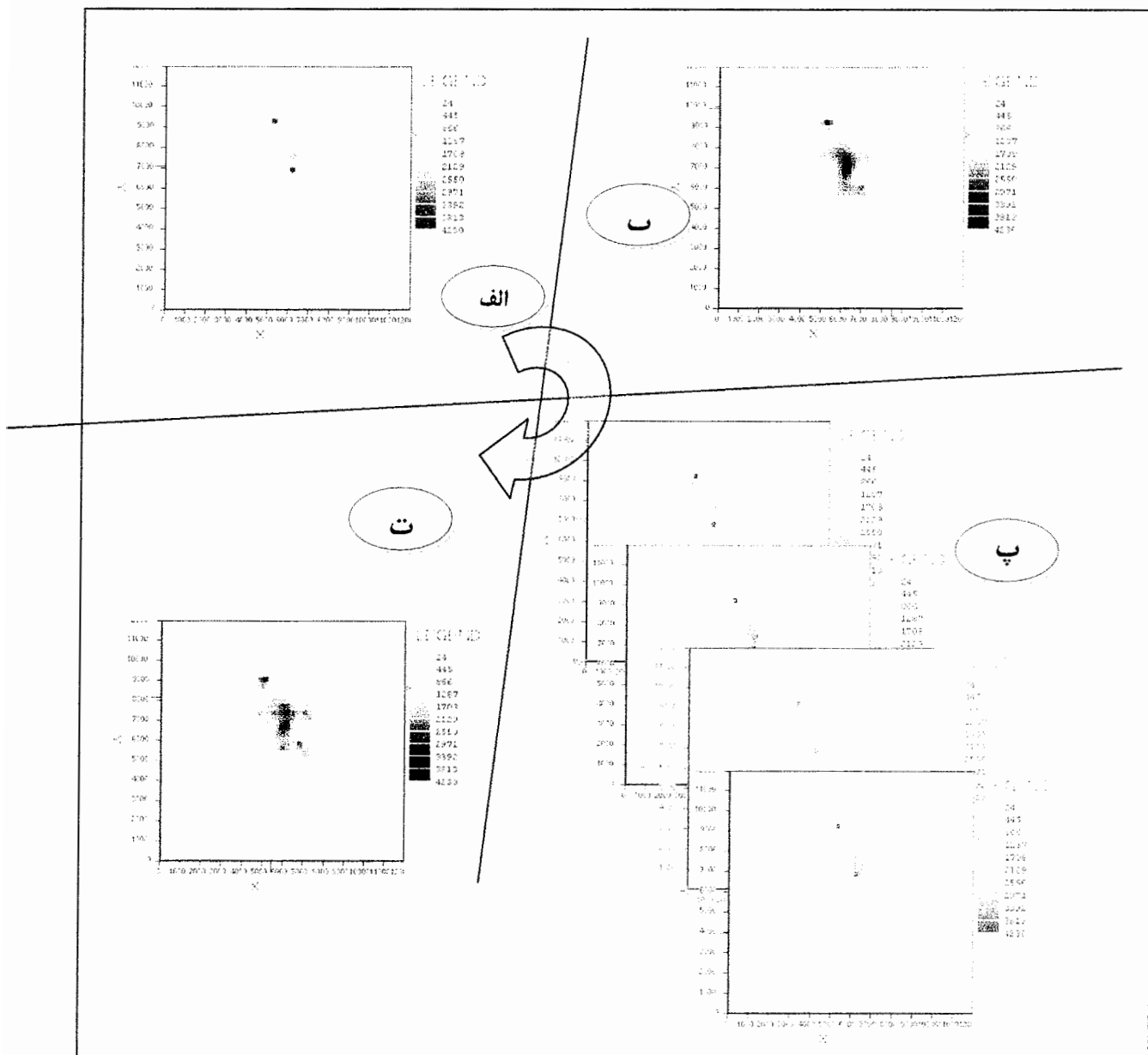
یک سری اطلاعات نیز به صورت بسیط تر و جامع تر و به شکل توزیع منطقه ای گسترده مورد توجه قرار می گیرند (مثلاً مدل جریان سیال در مخزن). این نوع اطلاعات اکثراً بصورت نقشه توزیع مقادیر منطقه ای نمایش داده می شود. روش کریجینگ مجموعه راهکاری جهت تهیه این قبیل نقشه ها می باشد. همزمان با تهیه این نقشه فاکتورهای دیگری مورد توجه قرار می گیرد که آنها را در دسته سوم اطلاعات قرار می دهیم. از جمله میزان عدم قطعیت<sup>۴۷</sup> محاسبات که مایلیم به صورت مقادیر عددی بیان گردد و به صورت توزیع منطقه ای نمایش داده شود. این نوع نقشه در محاسبات مربوط به تناژ و عیار ذخائر معدنی که مسئله کلاس و کاتاگوری ذخیره مورد توجه است اهمیت ویژه ای می یابد [۳].

شکل (۲-۱۱) نشانگر این سه دسته اطلاعات می باشد.

روشهای شبیه سازی زمین آماری مجموعه روشهایی هستند که ابتدا به منظور بررسی کمی عدم قطعیت شبکه توزیع مقادیر بکار گرفته شدند و به مرور جای خود را در طیف جدیدی از روشهای مدل سازی تصادفی<sup>۴۸</sup> و حل مسائل زمین آماری باز نمودند و هم اکنون نه تنها با اهداف عدم قطعیت سنجی بلکه جهت اهداف دیگری نظیر آنالیز ریسک با آنالیز تصمیم گیری و مدل سازی شبکه توزیع مقادیر مورد استفاده قرار می گیرند [۷].

---

46 - Simulation  
47 - Uncertainty  
48 - Stochastic



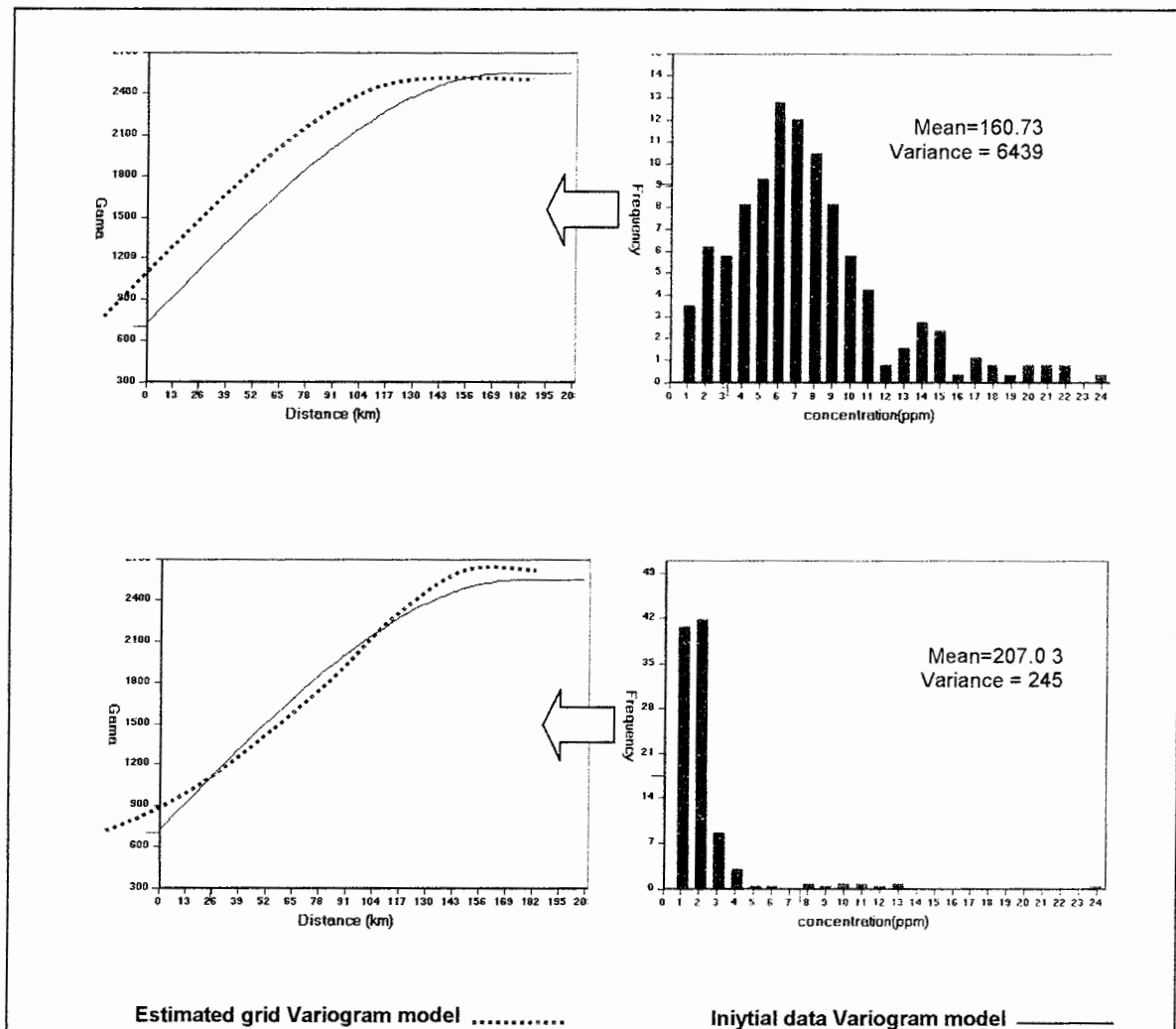
شکل (۲-۱۱) دسته بندی اطلاعات در یک پروژه زمین آماری. الف- داده های اولیه (اطلاعات نوع اول)، ب- نمایش توزیع مقادیر منطقه ای (اطلاعات نوع دوم)، پ- ترکیب چندین شبکه از نوع (ب) و ت- نمایش ترکیب شبکه ها با اهداف خاص (اطلاعات نوع سوم).

## ۲-۶-۲- ارتباط تخمین و شبیه سازی

همان طور که قبلاً اشاره شد هدف از تخمین حدس زدن عددی بعنوان مقدار یک نقطه نمونه برداری نشده با استفاده از مقادیر نمونه های اطراف آن به صحیح ترین نحو ممکن می باشد که این عملیات عمدتاً با روش کریجینگ انجام می گیرد و پایه و اساس آن حداقل نمودن خطا

است. هم اکنون آنچه مورد توجه است وریوگرام داده های تخمین در مقایسه با مدل وریوگرام داده های اصلی می باشد.

همان طور که در شکل (۲-۱۲) مشخص است تضمینی وجود ندارد که همیشه این دو وریوگرام بر هم منطبق باشند. علت آن هم مشخص است در مدل سازی توزیع بروش کریجینگ هموار سازی صورت می گیرد و این امر تغییر تابع توزیع تصادفی مقادیر را به همراه دارد. روشهای شبیه سازی درصد ایجاد مدلی می باشد که علاوه بر ایجاد نقشه ای مشابه آنچه در روش کریجینگ بدست آمد، تابع توزیع احتمال مقادیر را حفظ نماید.



شکل (۲-۱۲) مقایسه میزان انطباق وریوگرام واقعی و وریوگرام مقادیر تخمین در دو توزیع با چولگی متفاوت

در صورت اعمال هموارسازی به نتایج شبیه سازی همان نقشه تخمین بدست خواهد آمد. از نظر تئوریک می توان گفت میانگین مقادیر حاصل از نقشه های شبیه سازی شده به سمت مقادیر حاصل از تخمین میل می نماید.

## ۲-۶-۳- تئوری شبیه سازی [۷]

روش تخمین کریجینگ را به خاطر آورید. اگر قصد تخمین مقدار  $Z(x)$  را با استفاده از مقادیر معلوم مجموعه  $Z(x_1), Z(x_2), \dots, Z(x_n)$  که یک مجموعه با توزیع نرمال است را به روش کریجینگ داشته باشیم، در نهایت مجموعه مقادیری نتیجه می شود که دارای توزیع تک متغیره<sup>۴۹</sup> نرمال با مقادیر میانگین و واریانس به شرح زیر می باشد:

$$Z_{SK} = m + [Z-m].\Gamma^l.\Gamma_x$$

$$\sigma_{est}^2 = \Gamma(0) + \Gamma_x^T.\Gamma^l.\Gamma$$

محاسبه تک تک نقاط صفحه براساس توزیعی با مشخصات فوق ایده اولیه شبیه سازی به

روش مرحله ای<sup>۵۰</sup> می باشد.

مراحل اصلی روش بدین شرح است :

- ✓ انتخاب یک نقطه شبکه به طور تصادفی
  - ✓ تخمین مقدار و واریانس نقطه به روش کریجینگ ( $\sigma_{est}^2$  و  $Z_{SK}$ )
  - ✓ نمونه برداری تصادفی از توزیع  $N(\sigma_{est}^2, Z_{SK})$  و اختصاص مقدار نمونه برداری شده به نقطه
  - ✓ افزودن این نقطه به نقاط اصلی دارای مقدار
  - ✓ بازگشت به حالت اول تا ایجاد مدل کامل توزیع منطقه ای مقادیر
- معنی عملیات فوق این است که مجموعه نقاط  $Z(x_1), Z(x_2), \dots, Z(x_n)$  که یک مجموعه  $n$  عضوی نرمال می باشد می تواند به یک مجموعه  $n$  تایی از توابع توزیع تک متغیره تعبیر گردد. لذا مدل توزیع ایجاد شده بدین روش لزوماً دارای توزیع تک متغیره نظیر آنچه در کریجینگ لازم است، نخواهد بود.

در صورتی که مجموعه  $Z(x_i)$  دارای توزیع نرمال نباشد، قبل از عملیات فوق با یک انتقال توزیع به حالت نرمال، عملیات شبیه سازی انجام می گیرد. در این حالت به عملیات، شبیه سازی گوسی مرحله ای<sup>۵۱</sup> اطلاق می گردد که از روشهای شبیه سازی مرسوم و معمول در زمین آماری می باشد. این روش جزء روشهای شرطی<sup>۵۲</sup> است.

از دیگر روشهای شبیه سازی که در زمین آمار کاربرد دارند می توان به شبیه سازی شاخص مرحله ای<sup>۵۳</sup>، شبیه سازی فضای احتمال<sup>۵۴</sup>، و انیلینگ شبیه سازی شده<sup>۵۵</sup> اشاره نمود.

## ۲-۶-۴- شبیه سازی گوسی مرحله ای

- ۱) مراحل شبیه سازی گوسی مرحله ای به بیان جزئی تر بدین شرح است [۱۵]:
- ۲) تبدیل داده های اولیه به توزیع نرمال (در صورت نیاز).
- ۳) ایجاد یک شبکه بر حسب مختصات داده های اولیه.
- ۴) رسم و مدل سازی وریوگرام بر اساس داده های تبدیل یافته.
- ۵) انطباق داده های اندازه گیری شده با شبکه مذکور.
- ۶) انتخاب یکی از نقاط شبکه بصورت تصادفی.
- ۷) تخمین مقدار این نقطه  $(Z_i)$  با استفاده از کریجینگ معمولی و با استفاده از داده های اندازه گیری شده و کلیه داده های شبیه سازی شده قبلی و محاسبه مقدار واریانس کریجینگ برای این نقطه.
- ۸) ایجاد یک مقدار تصادفی با نمونه گیری از توزیع نرمال  $N(Z_i, \sigma_{est}^2)$  و انتصاب این مقدار به نقطه مورد نظر.
- ۹) تکرار عملیات از مرحله ۵ تا زمانی که کل شبکه پوشش داده شود.
- ۱۰) تبدیل معکوس مقادیر محاسبه شده به توزیع اولیه در صورت نیاز.

51 - Sequential Gaussian simulation

52 - Conditional

53 - Sequential indicator simulation

54 - Probability field simulation

55 - Simulated annealing

۱۱) تکرار عملیات از مرحله ۴ تا زمانی که به تعداد لازم مقادیر شبیه سازی از منطقه بدست آید.

در مقدار دهی به شبکه (مرحله ۴) توجه به این نکته ضروری می باشد که نقاط نمونه برداری لزوماً با نقاط شبکه منطبق نمی باشند. لذا می توان نزدیک ترین نقطه شبکه به هر کدام از نقاط نمونه برداری را دارای مقداری معادل مقدار اندازه گیری شده دانست یا اینکه مقادیر اندازه گیری شده مستقیماً به نقاط شبکه منتسب نشده و کلیه نقاط شبکه تخمین زده شود. انطباق نقاط نمونه برداری با نقاط شبکه باعث افزایش سرعت عملیات و در نتیجه اجرای برنامه خواهد شد. علاوه بر این مقادیر نمونه برداری شده نیز در شبکه نهایی قرار گرفته و این امر کاهش خطای سیستماتیک را به همراه دارد. هر چند بواسطه این عملیات ناگزیر به جابجایی مختصات نقاط نمونه برداری خواهیم بود و در مواردی که نقطه نمونه برداری به فاصله مساوی از نقاط شبکه قرار دارد تولید داده های مضاعف و یا حذف داده ها را خواهیم داشت. بنابراین انطباق یا عدم انطباق دادن داده ها با نقاط شبکه بسته به وضعیت شبکه، تعداد نقاط نمونه برداری و گستردگی منطقه (چگالی برداشت) ممکن است توصیه شود.

در اولین بار تکرار حلقه (مراحل ۵ تا ۸) تخمین تنها براساس نقاط نمونه برداری و مقادیر موجود در فایل داده ها انجام می گیرد ولی با هر بار تکرار حلقه که یک مقدار جدید ایجاد می شود تخمین با توجه به مقادیر موجود در فایل داده ها و مقادیر شبیه سازی شده قبلی انجام خواهد گرفت.

#### ۲-۶-۵- شبیه سازی شاخص مرحله ای [۷]

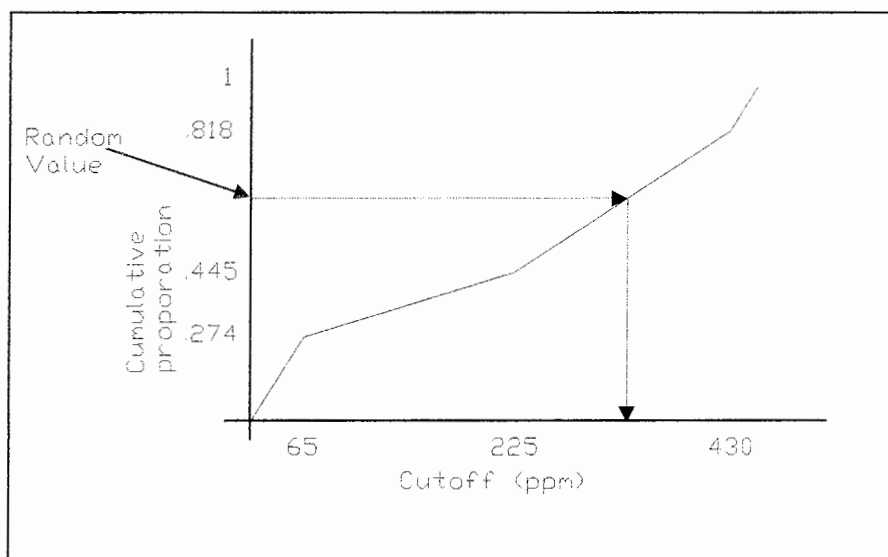
روش کار در شبیه سازی شاخص مرحله ای بدین ترتیب است که ابتدا مقادیر براساس حدود آستانه ای مشخص مثلاً صدکهای بیستم، چهلم، شصتم و هشتادم متمایز می گردند. سپس نمونه های با مقدار بیش از حد آستانه ای دارای مقدار ۱ و در غیر این صورت دارای مقدار صفر خواهند شد. در نهایت بازای هر کدام از مقادیر آستانه ای توزیع منطقه ای از مقادیر صفر و یک در نقاط نمونه برداری خواهیم داشت. در ادامه برای هر یک از مجموعه مقادیر مدل و ریوگرام شاخص ایجاد می گردد. لذا تعداد مدلهای و ریوگرام با تعداد مقادیر آستانه ای برابر خواهد بود.



مراحل اصلی عملیات شبیه سازی شاخص از هم اکنون آغاز می گردد. ابتدا یک نقطه تصادفی در منطقه انتخاب شده و به ازای مقادیر متناسب با هر کدام از حدود آستانه ای مقداری به روش کریجینگ برای آن تخمین زده خواهد شد.

مسئلاً مقادیر حاصل مابین صفر و یک خواهد بود و حاصل نهایی تابع توزیع تجمعی احتمال در نقطه مجهول می باشد. (شکل ۲-۱۳) سپس مقداری بصورت تصادفی از این توزیع انتخاب و به نقطه نسبت داده می شود. این عملیات تا پوشش کامل شبکه تکرار خواهد شد، با توجه به این نکته که در شبیه سازی هر نقطه، مقدار شبیه سازی شده نقاط قبلی نیز بعنوان داده های اصلی تلقی می گردد.

نتیجه نهایی عملیات یک شبکه حاوی مقادیر شبیه سازی شده می باشد و به منظور بررسی های عدم قطعیت سنجی چندین شبکه از این نوع ایجاد خواهد شد.



شکل ۲-۱۳) تابع توزیع احتمال حاصل از تخمین در حدود آستانه ای و نمونه برداری تصادفی از توزیع عملیات شبیه سازی شاخص مرحله ای

روش انیلینگ شبیه سازی شده جزو روشهای مرسوم به رویه های بهینه سازی ترکیبی (C.O.S.)<sup>56</sup> می باشد. ساختار کلی این گونه روشهای با هدف مینیموم سازی تابع خاصی شکل می گیرد. این روشها شامل یک سری معیارهای مشخص می باشند که خط و مشی عملیات مینیموم سازی تابع هدف را مشخص می نماید. بعلت تنوع در این معیارها است که رویه های بهینه سازی ترکیبی شامل شیوه ها و متدهای متعدد می گردد [۱۱].

توجه به این نکته ضروری است که روش بکار گرفته شده تعیین کننده سرعت عملیات شبیه سازی نیز می باشد و تابع هدف مفصل تر، عملیات کندتر و حجم محاسبات بیشتری خواهد داشت.

به طور کلی قدم اول الگوریتم انیلینگ ایجاد توزیع تصادفی مقادیر با استفاده از هیستوگرام فراوانی مشخص می باشد و مرحله بعد ایجاد آشفستگی در محیط خواهد بود. این عمل می تواند با جابجایی دو به دوی نقاط انجام گیرد و هدف این باشد که پس از هر بار ایجاد آشفستگی در محیط توزیع منطقه ای مقادیر به توزیع واقعی شبیه تر گردد. در صورتی که این امر حادث شود، اصطلاحاً سیستم به سطح انرژی پایین تر رسیده است و آشفستگی مورد پذیرش قرار می گیرد [۱۱].

بنابراین جهت انجام شبیه سازی بروش انیلینگ ابتدا یک توزیع تصادفی از مقادیر در شبکه ایجاد می کنیم. این توزیع با نمونه برداری تصادفی از تابع توزیع مقادیر اولیه یا هیستوگرام فراوانی مقادیر ایجاد می شود. سپس معیار مقایسه را مشخص می نمائیم. این معیار عموماً وریوگرام است. ولی معیارهای دیگری هم علاوه بر وریوگرام می توان منظور نمود.

همان طور که قبلاً نیز اشاره شد، ایده کلی عملیات انطباق معیار مقایسه توزیع تصادفی مفروض با مقادیر همان مقدار در توزیع دادهای اولیه می باشد. یعنی سعی داریم مدل وریوگرام تابع توزیع تصادفی مفروض را به مدل وریوگرام دادهای اصلی نقاط نمونه برداری شده منطبق نمائیم. بدین منظور با ایجاد آشفستگی در محیط و محاسبه مقادیر وریوگرام به بررسی این نکته می پردازیم که آیا مدل وریوگرام محیط به مدل دادهای اولیه نزدیکتر شده است؟ اگر جواب مثبت

باشد آشفتگی تأیید می گردد، در غیر این صورت توزیع مقادیر محیط به حالت قبلی باز گردانده می شود و عملیات از سر گرفته خواهد شد. یکی از راههای آشفته نمودن محیط جابجایی دو نقطه به صورت تصادفی می باشد.

در حالتی که معیار فقط وریوگرام می باشد تابع هدف برابر خواهد بود با [۱۰]:

$$\sum_{i=1}^{n_h} [\gamma^*(h_i) - \gamma(h_i)]^2 \quad (10-2)$$

که  $\gamma(h_i)$  مقادیر وریوگرام داده‌های اصلی به ازای فواصل جفت نمونه‌ها برابر  $h$  و  $\gamma^*(h_i)$  همین مقدار در مدل توزیع تصادفی مقادیر مفروض می باشد. ابتدا مقدار تابع برای توزیع اولیه بدست می آید و سپس با جابجا نمودن دو نقطه در محیط مقدار آن محاسبه می شود. در صورتی که مقدار تابع هدف کاهش یابد شبکه جدید بعنوان توزیع مقادیر منطقه پذیرفته می شود. بمنظور جلوگیری از قرار گرفتن الگوریتم در حالت‌های بهینه محلی بعضی تعویض‌ها که مقدار تابع هدف را افزایش می دهند نیز مورد قبول واقع می شوند که احتمال پذیرش این جابجایی‌ها از رابطه زیر بدست می آید:

$$p(\text{accept}) = e^{-\frac{O_{old} - O_{new}}{t}} \quad (11-2)$$

در این رابطه

$p(\text{accept})$  مقدار احتمال پذیرش جابجایی

$O_{old}$  مقدار تابع هدف قبل از جابجایی،

$O_{new}$  مقدار تابع هدف پس از جابجایی و

$t$  که بعنوان درجه حرارت نامیده می شود مقداری است که ارزش رسیدن الگوریتم به راه حل را تعیین می کند. مقادیر بالای این پارامتر سرعت دستیابی الگوریتم به جواب را کاهش می دهد و مقادیر پایین آن نیز احتمال قرار گرفتن الگوریتم در حالت‌های بهینه محلی را افزایش می دهد. لذا انتخاب یک مقدار بهینه برای آن بسته به کاربرد روش شبیه سازی انیلینگ در پروژه از اهمیت برخوردار است. عموماً در الگوریتم انیلینگ یک مقدار اولیه به این پارامتر اختصاص داده می شود که طی اجرای الگوریتم مقدار آن کاهش و به حداقل خود می رسد. منظور نمودن مقادیر حداقل و حداکثر بسته به شرایط اجرای پروژه تفاوت می کند.

- عملیات تا زمانی که مقدار تابع هدف از حد ثابتی کوچکتر نشود و اصطلاحاً محیط منجمد شود ادامه یافته و در نهایت شبکه توزیع مقادیر ایجاد می گردد [۱].
- به طور کلی الگوریتم انیلینگ را می توان در موارد زیر طراحی نمود:
- ۱) تعریف شبکه منظم.
  - ۲) محاسبه و رسم واریوگرام برای داده های اولیه.
  - ۳) مقداردهی اولیه تصادفی بر اساس تابع توزیع مقادیر اولیه به کلیه نقاط شبکه. برای نقاط منطبق به نقاط نمونه برداری همان مقادیر اولیه منظور می گردد.
  - ۴) محاسبه و رسم وریوگرام برای مقادیر حاضر شبکه. مسلماً در حالت تصادفی مطلق (اولین اجرای حلقه) وریوگرام صرفاً شامل اثر قطعه ای خواهد بود.
  - ۵) بررسی میزان انطباق وریوگرام داده های اصلی با وریوگرام نقاط شبکه بر اساس تغییرات تابع هدف و در صورتی که انطباق قابل قبولی نداشته باشد:
  - ۶) جابجایی دو نقطه شبکه و محاسبه مجدد وریوگرام.
  - ۷) بررسی مجدد انطباق وریوگرام داده های اولیه با وریوگرام حاصل. اگر مدل شبیه سازی شده به مدل اصلی نزدیکتر شد جابجایی پذیرفته می شود و در غیر این صورت جابجایی رد خواهد شد.
  - ۸) بازگشت به مرحله ۶ تا زمانی که انطباق قابل قبولی مابین وریوگرام داده های اولیه و وریوگرام نقاط شبیه سازی شده بدست آید.

## فصل سوم

نرم افزار Gridat، پیکره کلی برنامه و

الگوریتم های اصلی

## فصل سوم

### نرم افزار Gridat، پیکره کلی برنامه و الگوریتم های اصلی

#### ۳-۱- معرفی نرم افزار

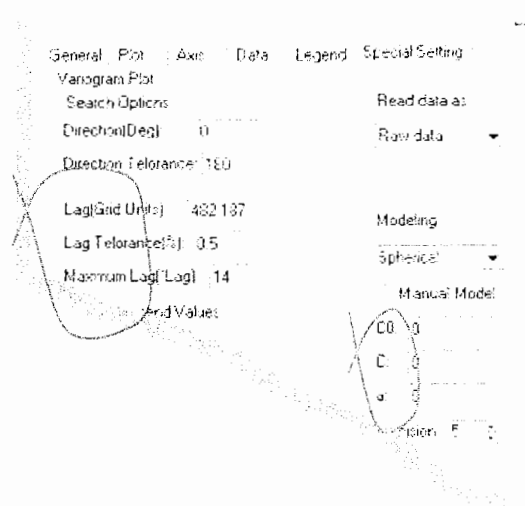
طراحی و برنامه نویسی نرم افزاری با نام Gridat موضوع اصلی و محور پایان نامه حاضر می باشد. این نرم افزار با اهداف زمین آماری طراحی شده و حداقل مراحل اصلی یک پروژه زمین آماری را در بر می گیرد.

در حال حاضر معمول ترین و مرسوم ترین نرم افزار زمین آماری WinGslib 1.3 است که در حقیقت رابط گرافیکی مجموعه برنامه های GSLIB90 به محیط ویندوز می باشد و در سال ۱۹۹۷ تهیه شده است [۱۰]. یکی از علل محبوبیت آن؛ علاوه بر امکان حذف محدودیت زمانی و در نتیجه رایگان بودن، جامعیت این نرم افزار است. بطوری که کلیه مراحل عملیات زمین آماری اعم از بررسی و آنالیز داده ها، واریوگرافی، تخمین و شبیه سازی در مجموعه های تک متغیره و چند متغیره و در حالت دو بعدی و به طور محدودتر سه بعدی ارائه می دهد. لیکن این نرم افزار هم به اندازه کافی قدیمی شده و استفاده از آن مشکلاتی نظیر مدیریت فایلها، مشاهده نتایج و محدودیت در مشاهده همزمان فایلهای داده و نتایج عملیات، تنظیمات پیش فرض فایلهای پارامتری و ... را در بردارد علاوه بر اینکه ترکیب و ساختار کلی آن با شکل عمومی و کلی نرم افزارهای مرسوم امروزی سازگاری ندارد و مجموعه این عوامل کاربران را به استفاده از آن ترغیب نخواهد کرد.

هدف از طراحی و ساخت نرم افزار Gridat ارائه محیطی بود که در قالب منوها و گزینه های مرسوم نرم افزارهای ویندوز، الگوریتم ها و عملیات معمول زمین آمار را ارائه دهد،

همچنین مشاهده و دسترسی به داده ها همزمان با اجرای الگوریتم و مشاهده گرافیکی نتایج در محیط نرم افزار ممکن باشد. در طراحی این مجموعه سعی شده است کاربر به سمت اجرای صحیح پروژه هدایت شود. مثلاً مجموعه کادرهای محاوره ای دریافت پارامترها شامل کنترلهای کد گذاری شده با علائم مرسوم در زمین آمار کلاسیک می باشد و لذا استفاده از نرم افزار برای همه بخصوص کاربرانی که تجربه کمتری در زمینه کار با نرم افزارهای زمین آماری دارند توصیه می شود.

شکل (۱-۳) نشانگر کادر محاوره ای دریافت پارامترهای واریوگرافی و مدل سازی وریوگرام است. به نامگذاری کنترلها در این شکل توجه کنید. عبارات  $C_0, C, a, \text{Lag}$  مجموعه علائم و عبارات آشنا در زمین آمار می باشند.



شکل (۱-۳) نمونه کادر محاوره ای دریافت پارامترها

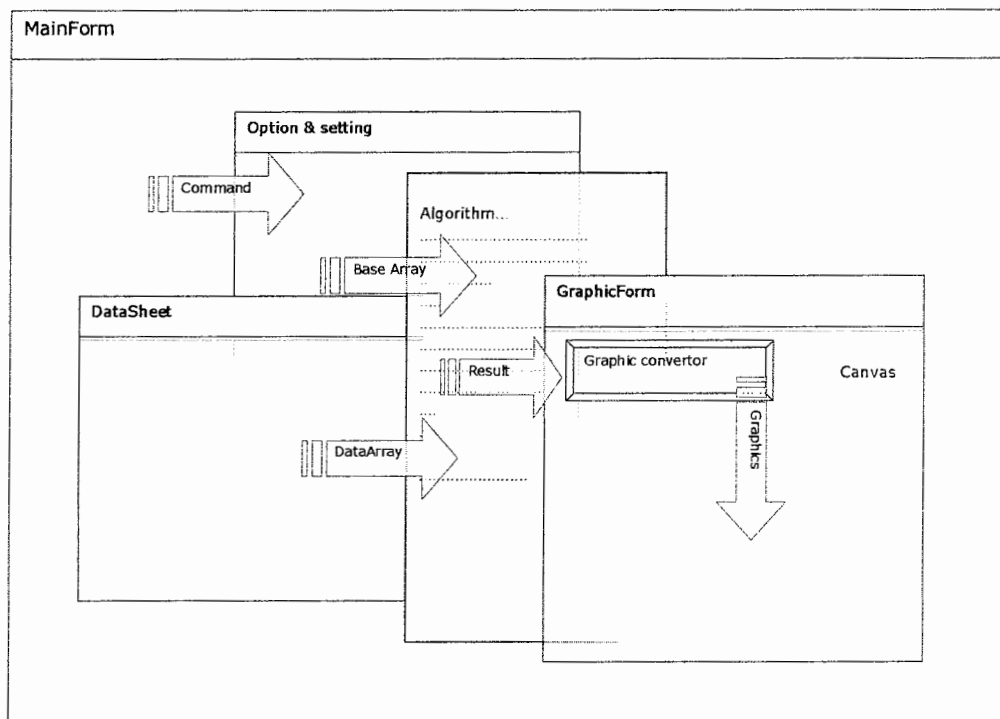
همان طور که پیشتر اشاره شد این نرم افزار شامل حداقل امکانات جهت اجرای یک پروژه زمین آماری می باشد. این امکانات شامل انجام عملیات زمین آماری نظیر بررسی فایل های داده، واریوگرافی، تخمین بروش کریجینگ (گوسی و شاخص) و شبیه سازی (گوسی و شاخص) در محیط دو بعدی و مجموعه های تک متغیره می باشد.

الگوریتم ها بر پایه اصول و فرمولهای زمین آمار کلاسیک طراحی شده و حتی المقدور از روشهای عددی در ساده سازی و مختصر نمودن محاسبات خودداری شده است. مسلماً کارایی این نرم افزار در اجرای یک پروژه بعهدہ کاربران خواهد بود و رفع نواقص آن علاوه بر تکمیل و افزودن گزینه های دیگر (نظیر مدل سازی سه بعدی و زمین آمار چند متغیره) منوط بر گذر موفق ترم افزار از این مرحله می باشد.

### ۳-۲- پیکره کلی برنامه

- نرم افزار Gridat شامل چهار قسمت اصلی به شرح زیر می باشد.
- ✓ فرم در برگیرنده داده ها: در هر پروژه تنها یک فرم داده داریم که می تواند حاوی چندین سری داده مجزا باشد و مدیریت داده ها از طریق این فرم انجام می گیرد.
  - ✓ فرم نمایش گرافیکی: نتایج گرافیکی حاصل از اجرای برنامه در این صفحات نمایش داده می شود این فرم به تعداد مورد نیاز طی اجرای برنامه تولید می گردد.
  - ✓ کادر محاوره ای تنظیمات: تعیین پارامترها و گزینه های اولیه جهت اجرای هر الگوریتم و همچنین ایجاد تغییر در آنها پس از اجرا و مشاهده گرافیکی نتایج از طریق این کادر صورت می گیرد.
  - ✓ فرم اصلی: که در برگیرنده کلیه اجزاء فوق بوده و وظیفه ارتباط و مدیریت آنها را بعهده دارد. همچنین اجرای الگوریتمها و مدیریت فایلها از طریق این فرم انجام می گیرد.
- به طور خلاصه می توان عملکرد مجموعه فوق را چنین بیان نمود :
- در آغاز اجرای نرم افزار تنها فرم مادر موجود است. در قدم اول فایل داده ها ایجاد شده و مقادیر آن ثبت می گردد. در حقیقت ایجاد فایل داده ها آغاز یک پروژه زمین آماری در نرم افزار Gridat می باشد. دستورات لازم جهت اجرای هر کدام از الگوریتم ها از طریق انتخاب گزینه های موجود در منوهای فرم اصلی صادر می گردد. کادر محاوره ای تنظیمات قبل از اجرای الگوریتم ظاهر شده و پس از تعیین پارامترهای لازم در این کادر، الگوریتم مربوطه فراخوانی می گردد.
- الگوریتم، اطلاعات را از فایل داده ها دریافت و براساس تنظیمات کادر محاوره ای اجرا می شود. در پایان نتایج به فرم نمایش گرافیکی ارسال خواهد شد. شکل (۳-۲) نشانگر چهارچوب اصلی و ارتباط اجزای نرم افزار است.





شکل (۲-۳) چهارچوب اصلی و ارتباط اجزای نرم افزار Gridat

جهت اجرای مراحل یاد شده نرم افزار دارای یک الگوریتم کنترل کننده علاوه بر الگوریتم های عملیاتی می باشد. بعبارت دیگر نرم افزار از یک الگوریتم اصلی تشکیل شده است که وظیفه آن مدیریت روند اجرای الگوریتم های عملیاتی است. مراحل اصلی این الگوریتم در رابطه با الگوریتم های عملیاتی بدین ترتیب است:

الف) دریافت دستور از فرم اصلی (بعنوان مثال دستور "VAR" که به منظور واریوگرافی صادر می گردد).

ب) نمایش کادر محاوره ای تنظیمات (در مورد دستور "VAR" نام صفحه داده ها، تنظیمات خاص مثل گام، شعاع تأثیر، امتداد و ... و تنظیمات عمومی نظیر رنگ نمودار، حدود نمایش و ...).

ج) ایجاد یک آرایه پارامتر جدید (آرایه پایه<sup>۱</sup>) جهت دستور جاری با استفاده از مقادیر ارسالی از کادر محاوره ای تنظیمات و همچنین ایجاد یک آرایه جهت انتقال داده ها از فایل داده ها.

د) ارسال آرایه پایه و آرایه داده ها به الگوریتم عملیاتی.

ه) دریافت نتایج اجرای الگوریتم در قالب یک یا چند آرایه مجزا توسط آرایه پایه.

و) ارسال نتایج به فرم نمایش گرافیکی در غالب آرایه پایه.

ز) عبور نتایج از یک مبدل گرافیکی.

ح) نمایش گرافیکی نتایج

ط) اعلام پایان اجرا

در مرحله فوق فرض بر این است که پروژه ایجاد شده و فایل داده ها موجود می باشد. ذخیره سازی پروژه نیز بصورت دو فایل مجزا می باشد. یک فایل داده با فرمت xml و دیگری مجموعه ای از آرایه های پایه ایجاد شده در پروژه که به صورت کد نویسی<sup>۲</sup> شده و با پسوند gda. ذخیره خواهد گشت.

لازم به یاد آوری است مدیریت فایلها (ذخیره<sup>۳</sup>، فراخوانی<sup>۴</sup>، ارسال<sup>۵</sup> و دریافت<sup>۶</sup> اطلاعات به اشکال مختلف، و ...)، مدیریت گرافیکی (نمایش اجزاء، نمایش همزمان فرم ها و چاپ<sup>۷</sup> نتایج) و مدیریت اجرایی پروژه (ایجاد پروژه، بررسی و تصمیم گیری در مورد مراحل پروژه و ...) از وظایف دیگر الگوریتم کنترل کننده می باشد.

- 
- 1 - Base Array
  - 2 - Serialized
  - 3 - Save
  - 4 - Open
  - 5 - Export
  - 6 - Import
  - 7 - Print



شکل ۳-۳) فلوچارت الگوریتم کنترل کننده، نرم افزار Gridat

در صورت بروز اشکال در هر کدام از مراحل الگوریتم کنترل کننده، عملیات به مرحله آخر یعنی اعلام پایان اجرا هدایت خواهد شد. مثلاً اگر فایل داده‌ها فاقد مقدار باشد اجرای الگوریتم از مرحله (ج) به مرحله (ط) موکول می‌گردد. شکل (۳-۳) فلوجارت الگوریتم کنترل کننده را نشان می‌دهد.

### ۳-۳- الگوریتم‌های عملیاتی

نرم افزار شامل هفت الگوریتم عملیاتی علاوه بر الگوریتم کنترل کننده می‌باشد که عبارتند از:

- ✓ الگوریتم نمایش نقاط نمونه برداری
- ✓ الگوریتم بررسی آماری توزیع داده‌ها
- ✓ الگوریتم واریوگرافی و مدل سازی واریوگرام
- ✓ الگوریتم تخمین بروش کریجینگ
- ✓ الگوریتم شبیه سازی مرحله ای
- ✓ الگوریتم شبیه سازی انیلینگ
- ✓ الگوریتم ترکیب شبکه‌ها

هر یک از الگوریتم‌ها وظیفه خاصی دارند و با هدف مشخصی طراحی شده‌اند ولی پیکره اصلی همه آنها مشابه می‌باشد. بعنوان مثال به منظور ترسیم نقشه نقاط نمونه برداری موجود در فایل داده‌های جاری، دستور "POST" توسط فرم اصلی ارسال می‌گردد. کادر محاوره‌ای تنظیمات با توجه به این دستور ظاهر شده و پارامترهای گرافیکی نظیر شکل، رنگ، اندازه و... در این کادر مشخص می‌گردد. سپس این اطلاعات توسط یک آرایه با نام آرایه پایه و داده‌ها توسط آرایه ای دیگر به الگوریتم نمایش نقاط نمونه برداری ارسال می‌گردد. الگوریتم عملیات لازم را انجام داده و مقادیر ایجاد شده مربوط به مختصات هر نقطه را از طریق آرایه پایه به همراه خصوصیات گرافیکی آن به مبدل گرافیکی ارسال می‌نماید. در نهایت فرم نمایش گرافیکی جدیدی تولید و این مقدار به نمایش در خواهد آمد. فلوجارت الگوریتم کنترل کننده را در شکل (۳-۳) مشاهده می‌کنید. مقدار آرایه پایه در حافظه برنامه باقی می‌ماند تا در صورت لزوم و

به منظور ایجاد تغییرات در گراف ایجاد شده دوباره بارگذاری گردد و در نهایت پروژه در قالب دو فایل مجزای داده ها و آرایه پایه ذخیره خواهد شد.

در ادامه به تشریح عملکرد هر کدام از الگوریتم ها می پردازیم.

### ۳-۳-۱- الگوریتم نمایش نقاط نمونه برداری

نمایش محلی نقاط نمونه برداری در صفحه اولین موردی است که در هر عملیاتی که با نمونه برداری سروکار دارد مورد توجه قرار می گیرد. با استفاده از این نقشه می توان توزیع نقاط نمونه برداری را از نظر وسعت، چگالی برداشت و توزیع چگالی برداشت بررسی نمود و در مورد مراحل بعدی عملیات تصمیم گیری نمود.

همچنین در صورتی که مقادیر نقاط برداشت شده بوسیله رنگ یا درج مقادیر در کنارشان مشخص شده باشد می توانیم تا حدودی از توزیع تقریبی مقادیر در منطقه آگاهی پیدا کنیم. الگوریتم این عملیات محاسبه و مراحل خاصی را نمی طلبد و صرفاً به تبدیل مقادیر مختصات داده ها به مختصات در خور فرم نمایش گرافیکی محدود می شود.

تنظیمات گرافیکی اختصاصی این الگوریتم، علاوه بر تنظیمات عمومی برنامه شامل شکل، رنگ و اندازه نمایش گر نقاط، مشخصات شناسه ها (مقادیر درج شده در کنار هر نقطه برداشت) و آیتم تعیین کننده تبعیت رنگ نقاط از مقدارشان می باشد.

مراحل اصلی این الگوریتم در موارد زیر خلاصه شده است. شکل (۳-۴) نمایش فلوجارت الگوریتم می باشد.

الف) بررسی آرایه داده های ارسالی و اطمینان از وجود مقادیر معتبر در فایل داده ها.

ب) بررسی حدود تغییرات داده ها.

ج) ایجاد متغیرها و آرایه های محلی<sup>۸</sup> جدید.

د) شناسایی آرایه پایه ارسالی.

ه) تبدیل مقادیر مختصات نقاط به مقادیر در خور فرم گرافیکی پیش فرض

(بیکسل ۶۰۰×۸۰۰).

- و) مقدار دهی به آرایه ها و متغیرهای ایجاد شده حین عملیات.
- ز) انتقال مقادیر متغیرها به آرایه پایه.
- ح) نابودی متغیرهای و آرایه های محلی.
- ط) اعلام پایان عملیات جاری.
- ی) بازگشت به آرایه کنترل کننده جهت ارسال به مبدل گرافیکی.
- در صورت بروز خطا در هر کدام از مراحل عملیات فوق، پایان عملیات اجرا می گردد.
- (مرحله ح)) پیوست (الف - ۱) حاوی کد برنامه<sup>۹</sup> الگوریتم فوق می باشد.

Controller  
Algorithm

Data  
examination

New local  
variables

base array  
recognition

Sort data by  
values

Calculate data  
coordinates scaled  
to default graphical  
resolution.

Charge base array  
using local variables

Dispose  
local variables.

Ready message

Controller  
Algorithm

شکل ۳-۴) فلوچارت الگوریتم نمایش نقاط نمونه برداری

### ۳-۳-۲- الگوریتم بررسی آماری توزیع داده ها

بررسی آماری توزیع داده ها قبل از انجام هر عملیات دیگر از ملزومات پروژه های زمین آماری است که اهمیت این مطلب از اهمیت توزیع نرمال در این مطالعات ناشی می شود. بطوری که پیروی از تابع توزیع نرمال از ملزومات داده های مورد استفاده در عملیاتی نظیر کریجینگ و شبیه سازی گوسی می باشد.

روش های مختلفی جهت بررسی نرمالیت توزیع پیشنهاد شده که می توان آنها را در دو مجموعه اصلی روش های عددی و روش های گرافیکی خلاصه نمود [۱۲].

روش های عددی که بر پایه محاسبه گشتاور مرتبه سوم و چهارم انحراف از معیار بنا شده اند. (چولگی و کورتوزیس). توزیع نرمال مقادیر صفر را برای چولگی و ۳ را برای کورتوزیس بدست می دهد [۱۲].

روش های گرافیکی بر پایه مشاهده رفتار توزیع متغیر یا متغیرها در نمودارها و چارتهای متفاوت بنا شده است. از جمله می توان به نمودارهای درختی<sup>۱۰</sup> برای داده های پیوسته و زمانی که تعداد نمونه برداری کم است، نمودار جعبه ای<sup>۱۱</sup> و نقطه ای<sup>۱۲</sup> برای داده های کلاسه شده با نمونه برداری کم، هیستوگرام برای مقادیر بالاتر نمونه برداری و داده های کلاسه بندی شده و نمودارهای احتمال - احتمال و مقدار - مقدار برای مقایسه توزیع متغیرها با یک توزیع نرمال استاندارد بکار می روند [۱۲].

از دیگر گرافهای مرسوم در بررسی نرمالیت توزیع داده ها، نمودار توزیع احتمال می باشد. این گراف شامل یک محور با درجه بندی توزیع نرمال استاندارد بوده و محور دیگر آن شامل مقادیر متغیر اندازه گیری شده می باشد و بنای کار ترسیم فراوانی تجمعی داده ها بدون توجه به کلاس بندی است [۱۲].

امکاناتی که در نرم افزار Gridat جهت بررسی توزیع مقادیر منظور شده شامل موارد زیر

است:

- 10 - Stem & left Plot
- 11 - Box Plot
- 12 - Dot Plot

بررسی به روشهای عددی: که از طرق مختلفی دسترسی به آن ممکن می باشد. از جمله بطور مستقیم در کادر محاوره‌ای تنظیمات و مشاهده گزارش فرم های نمایش گرافیکی که به موقع توضیحاتی در مورد آن ارائه خواهد شد.

نمودار جعبه ای: که شامل نمایش جعبه ای و نمایش نقطه ای توزیع مقادیر شبکه نمونه برداری است و مشاهده آن همواره از طریق کادر محاوره ای تنظیمات ممکن می باشد. این امر به تصمیم گیری در مورد عملیات جاری و داده های پرت<sup>۱۳</sup> در زمان اجرای هر کدام از مراحل پروژه کمک می کند.

هیستوگرام های توزیع فراوانی نسبی و توزیع فراوانی نسبی- تجمعی: گرافهای توزیع نرمال و توزیع نرمال تجمعی با توجه به پارامترهای آماری هیستوگرام ترسیم می گردد و میزان انطباق هیستوگرام با گراف مذکور به نوعی بیانگر میزان تبعیت توزیع مقادیر از حالت نرمال است. نمودار احتمال: که امکان بررسی نرمالیت توزیع مقادیر به صورت خام و کلاسه بندی شده را فراهم می آورد.

نمودار احتمال لگاریتمی: که به منظور بررسی لاگ نرمال بودن توزیع بکار می رود. الگوریتم بررسی آماری توزیع داده ها وظیفه ارائه ابزار یاد شده را بعهدہ دارد. طی این الگوریتم داده ها پس از مرتب شدن به ترتیب افزایش مقادیر متغیر کلاسه بندی شده و بسته به نوع چارت مورد نظر مقادیر فراوانی و فراوانی تجمعی هر دسته شمارش می شود. در صورت نیاز این مقادیر به تعداد کل داده ها تقسیم می شود تا مقادیر فراوانی نسبی و نسبی تجمعی هر دسته بدست آید. سپس مقادیر با توجه به ابعاد پیش فرض صفحه نمایش به مقادیر مختصات صفحه تبدیل می شوند و در صورت نسبی بودن چارت، مقادیر پارامترهای آماری استخراج شده و مدل توزیع نرمال براساس آنها تعیین می گردد.

این الگوریتم تنظیمات گرافیکی اختصاصی مفصل تری دارا می باشد. نوع هیستوگرام، رنگ، ابعاد، شناسه ها و حتی نمایش مقیاس نرمال (نمودار احتمال) و مقیاس لگاریتمی (نمودار احتمال لگاریتمی) جزء تنظیمات گرافیکی این الگوریتم می باشند. مقادیر ارسالی در دو حالت



صرفاً نسبی - تجمعی می باشد و عملیات تبدیل مقادیر به مقادیر نرمال یا لگاریتمی وظیفه مبدل گرافیکی است.

مراحل اصلی این الگوریتم به ترتیب زیر است :

الف) بررسی آرایه داده های ارسالی و اطمینان از وجود مقادیر معتبر در فایل داده ها

ب) ایجاد متغیرها و آرایه های جدید به صورت محلی

ج) شناسایی آرایه پایه ارسالی

د) ایجاد ترتیب صعودی در داده ها

ه) تقسیم داده ها به تعداد مشخص شده کلاس و شمارش جمعیت هر کلاس

و) شناسایی نوع چارت مورد نظر

و-۱) اگر هیستوگرام فراوانی مورد نظر باشد: مرحله (ز)

و-۲) اگر هیستوگرام فراوانی تجمعی مورد نظر باشد: محاسبه مقادیر تجمعی هر کلاس و

مرحله (ز)

و-۳) اگر هیستوگرام فراوانی نسبی مورد نظر باشد: محاسبه مقادیر فراوانی نسبی و مرحله

(ز)

و-۴) اگر هیستوگرام فراوانی نسبی - تجمعی مورد نظر باشد: محاسبه مقادیر فراوانی

نسبی - تجمعی و مرحله (ز)

و-۵) اگر نمودار احتمال مورد نظر باشد: محاسبه مقادیر فراوانی نسبی - تجمعی و مرحله

(ز)

و-۶) اگر نمودار احتمال لگاریتمی مورد نظر باشد: محاسبه مقادیر فراوانی نسبی - تجمعی

و مرحله (ز)

ز) بررسی حدود مقادیر تولید داده شده در هر کلاس

ح) تبدیل مقدار هر کلاس به مقادیر در خور ابعاد صفحه نمایش پیش فرض و مقدار دهی

به متغیرهای محلی

ط) محاسبه پارامترهای آماری داده ها و مقادیر مدل توزیع نرمال در صورت نیاز و مقدار

دهی به متغیرهای محلی

ی) انتقال مقادیر متغیرها به آرایه پایه

ک) نابودی متغیرها و آرایه های محلی

ل) اعلام پایان عملیات جاری

م) بازگشت به الگوریتم کنترل کننده جهت ارسال به مبدل گرافیکی

فلوچارت الگوریتم فوق مطابق شکل (۳-۵) و ارائه کد برنامه آن در پیوست (الف-۲)

می باشد.



شکل ۳-۵) فلوچارت الگوریتم بررسی آماری توزیع داده ها

### ۳-۳-۳- الگوریتم واریوگرافی و مدل سازی وریوگرام

الگوریتم بکار گرفته شده جهت واریوگرافی از الگوریتم استاندارد محاسبات وریوگرام

پیروی می کند.

✓ جهت محاسبه مدل وریوگرام حداقل مقادیر بدین شرح نیازمندیم :

✓ مقادیر داده ها و مختصات آنها (نرمالیز شدن)

✓ امتداد های محاسبه وریوگرام و تعداد آنها و دامنه تغییرات آنها<sup>۱۴</sup>

✓ تعداد گام ها و فاصله گام ها و دامنه تغییرات در آنها

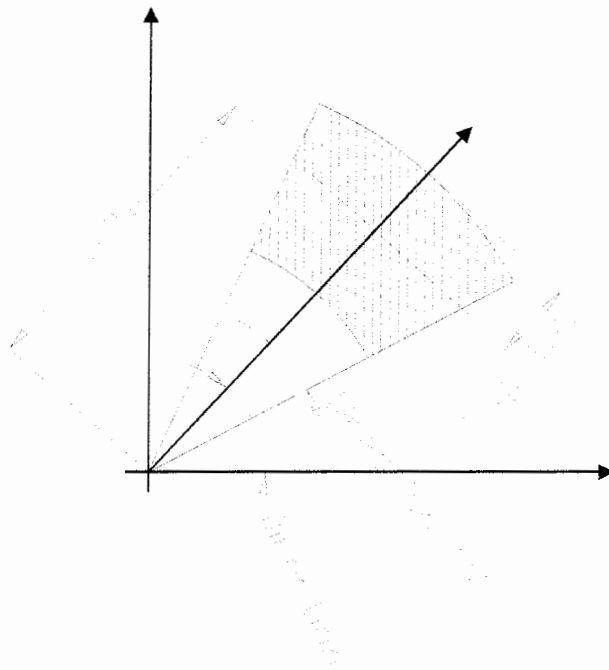
✓ تعداد و نوع وریوگرام

الگوریتم کلی محاسبه مدل بدین ترتیب است که با موجود بودن مقادیر فوق شبکه ای

جهت جستجوی نقاط نمونه برداری به شرح زیر منظور می کند.

ابتدا بر روی اولین نقطه نمونه برداری قرار گرفته (با توجه به مختصات) و محدوده ای

قطاعی به شکل زیر در نظر می گیرد:



شکل ۳-۶) روش جستجوی نقاط شبکه غیر منظم در محاسبه وریوگرام

همانطور که در تصویر مشاهده می شود مقادیر پارامتر های یاد شده منظور گشته اند.

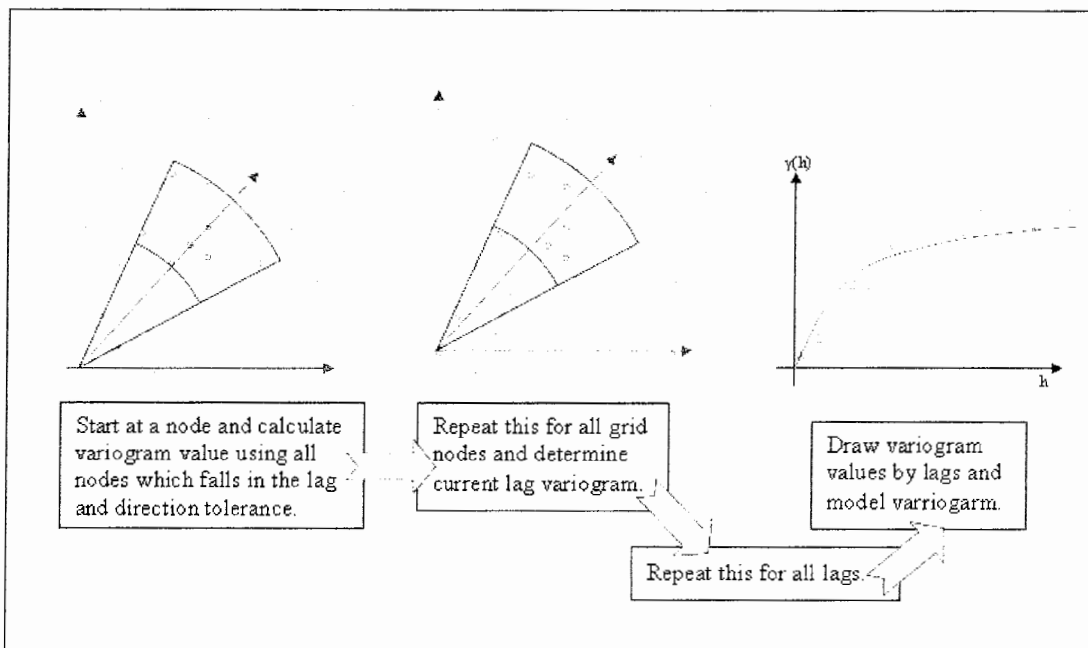
هم اکنون با استفاده از فرمول

$$2\gamma(h) = \frac{1}{N(h)} \sum [Z(u) - Z(u+h)]^2 \quad (1-3)$$

مقادیر وریوگرام با مبدا حاضر نسبت به هر کدام از نقاط موجود در قطاع محاسبه و با جابجایی مبدا بر روی کلیه نقاط شبکه تعدادی وریوگرام به ازای فاصله برابر با گام بدست می آید.

این عملیات برای گامهای بعدی نیز اعمال می گردد و بدین ترتیب مقادیر وریوگرام برای

گام های متفاوت محاسبه و نمودار آماده انطباق مدل می گردد (شکل ۷-۳)



شکل ۷-۳) مراحل مدل سازی وریوگرام.

در الگوریتم واریوگرافی و مدل سازی وریوگرام به منظور مدل سازی از روشهای عددی (مدل های خطی، توانی و لگاریتمی) و روش کمترین مربعات وزنی (دیگر مدل های وریوگرام) استفاده شده است.

روش های عددی شامل یک سری فرمولها و روابط می باشد که مقادیر ضرایب معادلات مدل را با توجه به مقادیر داده های دو بعدی محاسبه می نماید. در حقیقت این روشها نوعی معادلات رگرسیون می باشند.

در الگوریتم مدل سازی وریوگرام به منظور ایجاد مدل خطی با فرمول  $C_0 + C.h$  روابط زیر بکار گرفته شده است [۴].

$$C_0 = \frac{\sum x \sum xy - \sum y \sum x^2}{(\sum x)^2 - n \sum x^2} \quad (۲-۳)$$

$$C = \frac{\sum x \sum y - n \sum xy}{(\sum x)^2 - n \sum x^2} \quad (۳-۳)$$

و جهت مدل سازی توانی با فرمول  $C_0 + C.h^a$ ، مقادیر  $C$  و  $a$  به روش عددی (روابط ۳-۴ و ۳-۵) و جهت محاسبه  $C_0$  از روش کمترین مربعات وزنی استفاده شده است. روابط محاسبه  $C$  و  $a$  بدین ترتیب است [۴].

$$a = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} \quad (۴-۳)$$

$$C = e^{\frac{\sum y - a \sum x}{n}} \quad (۵-۳)$$

و مدل سازی لگاریتمی به شکل  $C_0 + 3C.In(h)$  با استفاده از روابط زیر انجام می گیرد [۴].

$$C = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} \quad (۶-۳)$$

$$C_0 = \frac{\sum y - C \sum x}{n} \quad (۷-۳)$$

در روابط فوق که مقادیر گام به محور  $x$  و مقادیر وریوگرام به محور  $y$  منطبق شده اند و لذا  $n$  تعداد گام می باشد،  $\sum x^2$ ،  $\sum x$  و  $(\sum x)^2$  به ترتیب مجموع مقادیر، مجموع توان دوم مقادیر و توان دوم مجموع مقادیر گام می باشند. به همین ترتیب  $\sum y^2$ ،  $\sum y$  و  $(\sum y)^2$  مجموع مقادیر، مجموع توان دوم مقادیر و توان دوم مجموع مقادیر وریوگرام هستند و  $\sum xy$  نیز مجموع حاصل ضرب مقادیر گام با مقدار وریوگرام متناظر با آن است.

روش کمترین مربعات وزنی یکی از روشهای عددی انطباق مدل به وریوگرام می باشد

[۱۴]. این روش در سال ۱۹۸۵ توسط کرزی<sup>۱۵</sup> ارائه شده است. وبستر و مک برانتی<sup>۱۶</sup> روشهای

متفاوتی از جمله روش کمترین مربعات، کمترین مربعات تعمیم یافته، حداکثر درست نمایی و

15 - Cresse

16 - Webster & Mc Brantey

کمترین مربعات وزنی را مقایسه کرده و نتیجه گرفتند که این روش نسبت به روشهای دیگر هم دارای محاسبات کمتری است و هم اینکه نتایج بهتری را بدست می دهد [۱۴].

بعد از انتخاب یک مدل استاندارد برای وریوگرام می توان از روش کمترین مربعات وزنی پارامترهای مدل مورد نظر را تخمین زد. برای این منظور باید تابع

$$f(h, \lambda) = \sum_{i=1}^k \frac{N_h(j)}{\gamma^2[h(j), \lambda]} \{ \gamma^*[h(i)] - \gamma[h(i), \lambda] \}^2 \quad (۸-۳)$$

مینیمم شود.

در این رابطه  $h(j)$  فواصل گام با ضریب  $j$  ام،  $Nh(j)$  تعداد جفت نمونه ها با فاصله  $h(j)$  از یکدیگر  $\gamma^*[h(j)]$  مقدار وریوگرام تجربی در  $h(j)$  و  $\gamma[h(j), \lambda]$  مقدار واریوگرام تئوریک به ازای پارامترهای تخمینی ( $\lambda$ ) است. نسبت  $N_h(j) / \gamma^2[h(j), \lambda]$  وزن نامیده می شود و موجب می گردد به نقاطی از وریوگرام که فاصله آنها نسبت به دامنه تأثیر کوچک است و یا تعداد زوج نمونه بیشتری در محاسبه آنها موثر است وزن بیشتری داده شود.

در رابطه فوق برای هر مدل می توان معادله آن را به جای  $\gamma[h(i), \lambda]$  قرار داد و با تغییر ضرایب آن به مقدار مینیمم تابع  $F(h, \lambda)$  دست یافت. به ازای آن ضرایبی که مقدار تابع مینیمم شده است، مدل وریوگرام تعریف می گردد [۱۴].

در الگوریتم واریوگرافی و مدل سازی وریوگرام از روش جستجوی ترتیبی به منظور یافتن ضرایب معادله مدل استفاده شده است. این روش علاوه بر سرعت بالای اجرا در اکثر مواقع نتایج رضایت بخشی ارائه می دهد. روش کار بدین ترتیب است که ابتدا حدودی برای هر یک از ضرایب تعریف می شود به نحوی که اطمینان داشته باشیم مقادیر ضرایب از این حوزه خارج نخواهند بود. سپس به تعداد ضرایب مجهول حلقه های تو در تو تشکیل داده و با تناوب یک دهم حدود به تابع مقدار می دهیم و آن مقدار از ضرایبی که باعث ایجاد مقدار حداقل مقدار شده اند به عنوان جواب اولیه منظور می شود. حال حدود ضرایب را دوباره تعریف می کنیم. این بار حوزه تغییرات برابر با مقدار هر کدام از ضرایب با نوسان برابر با تناوب حلقه ها یعنی یک دهم حدود اولیه خواهد بود. مجدداً حلقه های تو در تو، این بار نیز تناوب یک دهم حدود (که هم اکنون محدودتر شده) تشکیل می شود و عملیات مجدداً تکرار خواهد شد.

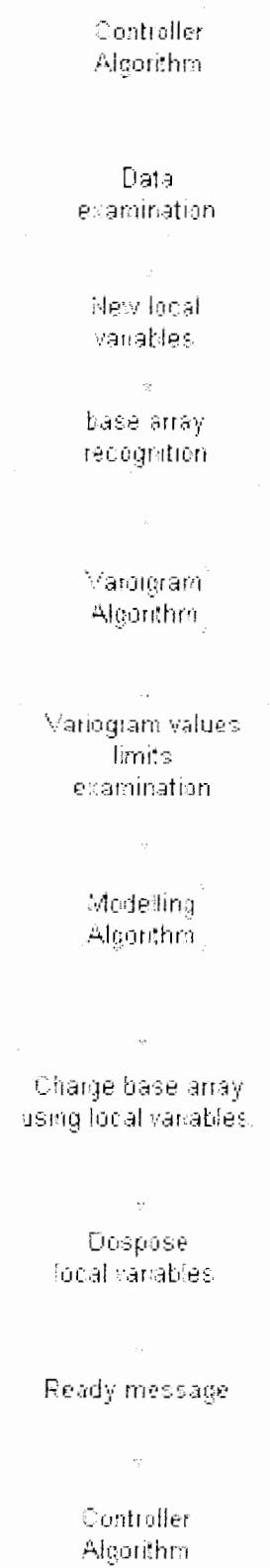
مجموعه عملیات فوق تا زمانی که به دقت مطلوب دست بیابیم ادامه خواهد یافت. مراحل کلی الگوریتم واریوگرافی و مدل سازی وریوگرام در موارد زیر و شکل (۳-۸) خلاصه شده است.

- الف) بررسی داده های ارسالی و اطمینان از وجود مقادیر معتبر در فایل داده ها.
  - ب) تعریف متغیرها و آرایه های محلی.
  - ج) شناسایی آرایه پایه.
  - د) محاسبه مقدار وریوگرام با فراخوانی الگوریتم مربوطه به ذخیره سازی مقادیر حاصل از واریوگرافی در یک آرایه محلی.
  - ه) بررسی حدود تغییرات مقادیر وریوگرام.
  - و) مدل سازی وریوگرام با فراخوانی الگوریتم مربوطه.
  - ز) تبدیل مقادیر وریوگرام (واریوگرافی و مدل) به مقادیر در خور صفحه نمایش پیش فرض و مقدار دهی به متغیرهای محلی.
  - ح) ارسال مقادیر وریوگرام و مدل به آرایه پایه.
  - ط) نابودی متغیرها و آرایه های محلی.
  - ی) اعلام پایان عملیات.
  - ک) بازگشت به الگوریتم کنترل کننده جهت ارسال به مبدل گرافیکی.
- عملکرد الگوریتم محاسبات واریوگرافی به ترتیب موارد زیر و شکل (۳-۹) می باشد. این الگوریتم مقادیر داده ها را دریافت و آرایه ای حاوی مقادیر در برگیرنده مقدار وریوگرام و فاصله باز می گرداند.
- الف) ایجاد متغیرها و آرایه های محلی و قرار دادن مقادیر اولیه (نظیر گام، امتداد، نوسان مجاز و ... ) در آنها.
  - ب) ایجاد یک متغیر به منظور ذخیره سازی مقادیر حین محاسبه واریوگرام.
  - ج) محاسبه توان دوم اختلاف مقدار یک نقطه با نقاط دیگر صفحه، به شرط اینکه نقطه دوم در حدود تأثیر نقطه اول با توجه به امتداد، گام و نوسان مجاز هر کدام، قرار داشته باشد و افزودن مقدار به متغیر (ب).

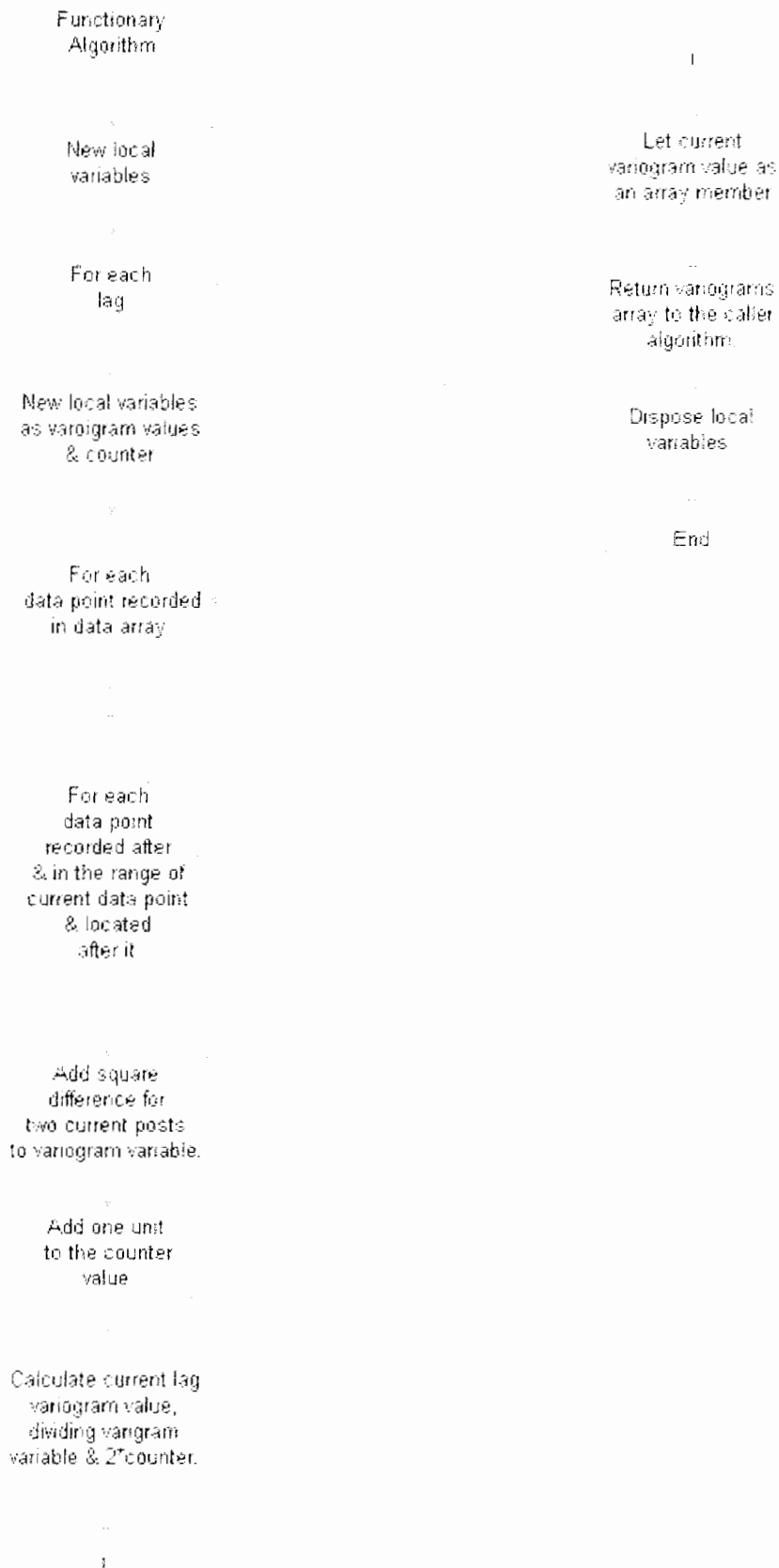


- (د) شمارش تعداد انجام عملیات مرحله (ج) (تعداد جفت نمونه ها).
- (ه) بازگشت به مرحله (ج) تا به بررسی آخرین عضو آرایه داده ها.
- (و) بازگشت به مرحله (ج) تا بررسی آخرین عضو آرایه داده ها.
- (ز) محاسبه مقدار وریوگرام به ازای گام جاری با تقسیم نمودن مقدار بدست آمده در مرحله (ج) به تعداد شمارش شده جفت نمونه ها (مرحله د).
- (ک) قرار دادن مقادیر گام جاری و وریوگرام در یک آرایه محلی.
- (ل) بازگشت به مرحله (ب) تا پوشش مقادیر شعاع تأثیر توسط مقادیر گام.
- (م) ارسال مقادیر محاسبه شده به صورت آرایه به الگوریتم فراخواننده.
- (ن) نابود سازی مقادیر محلی و پایان عملیات.
- دیگر الگوریتم مورد توجه در این عملیات الگوریتم انطباق مدل استاندارد وریوگرام است. این الگوریتم مقادیر وریوگرام را به صورت یک آرایه دریافت و مقادیر ضرایب مدل  $(a, C_0, C)$  را باز می گرداند. مراحل این عملیات به ترتیب شکل (۳-۱۰) و شرح زیر است:
- (الف) تعریف متغیرها و آرایه های محلی.
- (ب) اگر مدل خطی یا لگاریتمی مورد نظر باشد:
- ب-۱) محاسبه مقادیر  $C$  و  $C_0$  با استفاده از فرمولهای  $۲-۳$ ،  $۳-۳$ ،  $۴-۳$  و  $۶-۳$  و  $۷-۳$ .
- (ج) اگر مدل توانی مورد نظر باشد:
- ج-۱) محاسبه مقادیر  $C$  و  $a$  با استفاده از فرمول های  $۳-۳$  و  $۴-۳$ .
- ج-۲) تعیین حدود اولیه برای  $C_0$ .
- ج-۳) جستجوی ترتیبی برای بهترین مقادیر  $C_0$  بروش کمترین مربعات وزنی (پیوست ۱-ج).
- (د) اگر مدل های دیگر مورد نظر باشد.
- د-۱) تعیین حدود اولیه برای  $C, C_0$  و  $a$
- د-۲) جستجوی ترتیبی برای بهترین مقادیر  $C, C_0$  و  $a$  بروش کمترین مربعات وزنی.
- (ه) باز گرداندن مقادیر  $C, C_0$  و  $a$  به الگوریتم فراخواننده.
- (و) نابودی متغیرهای محلی و پایان عملیات.

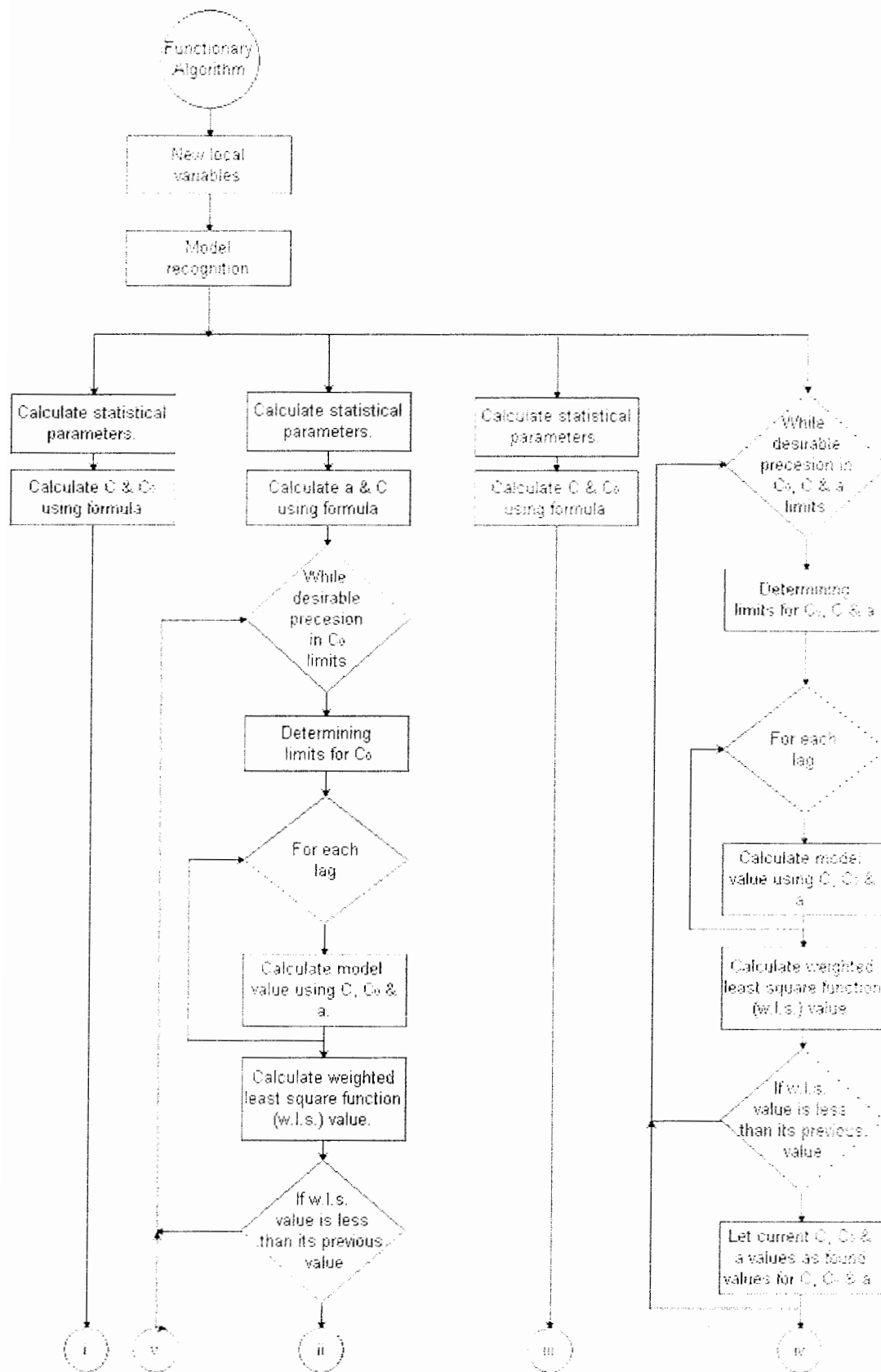
کد برنامه الگوریتم واریوگرافی و مدل سازی وریوگرام به پیوست (الف- ۳) می باشد.



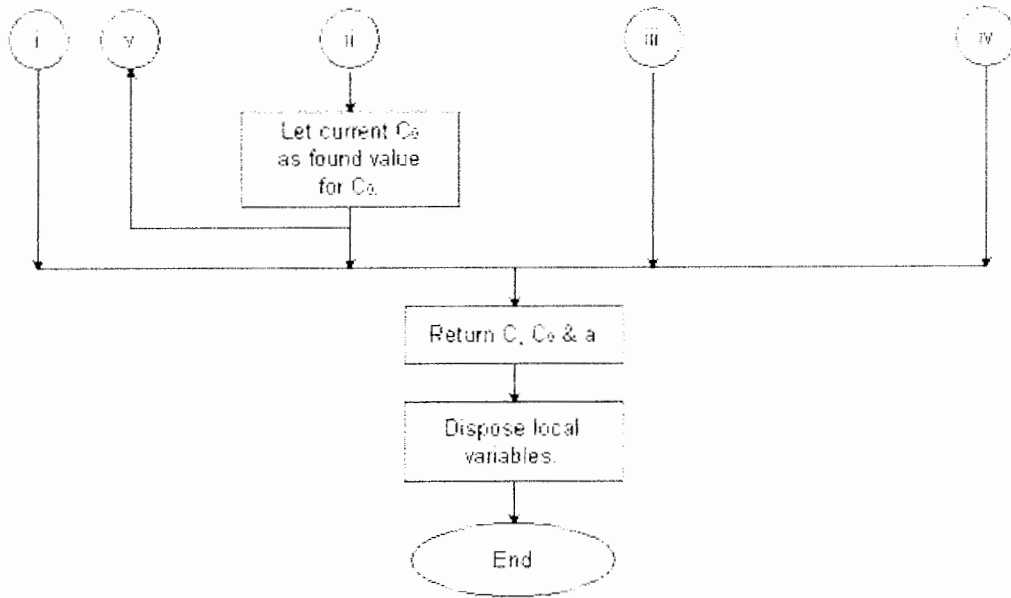
شکل ۳-۸) فلوچارت الگوریتم واریوگرافی و مدل سازی وریوگرام



شکل ۳-۹) فلوجارت الگوریتم واریوگرافی



شکل ۳-۱۰) فلوجارت الگوریتم مدل سازی و ریوگرام



ادامه شکل ۳-۱۰) فلوجارت الگوریتم مدل سازی وریوگرام

### ۳-۳-۴- الگوریتم تخمین بروش کریجینگ

این الگوریتم به منظور ایجاد شبکه توزیع منطقه ای مقادیر تخمین زده شده به روش کریجینگ و همچنین شبکه توزیع واریانس تخمین طراحی شده است.

به بیانی ساده می توان گفت این الگوریتم و الگوریتم های دیگری که از این پس مورد بررسی قرار می گیرند (شبه سازی و ترکیب شبکه ها) مشابه الگوریتم اول یعنی نمایش نقاط نمونه برداری عمل می کنند، با این تفاوت که کلیه نقاط شبکه دارای مقدارند. این مقدار می تواند مستقیماً توسط نمونه برداری بدست آمده باشد و یا این که بروش تخمین، شبه سازی و یا ترکیب مقادیر دو شبکه بدست آمده باشد.

روش کلی ترسیم این نوع شبکه ها بدین ترتیب است که ابتدا ابعاد شبکه تعیین شده و سپس مقادیر شبکه به روش تخمین یا شبه سازی محاسبه می گردد. این مقادیر به همراه مختصات نقاط شبکه در آرایه ای قرار داده می شود و سرانجام با استفاده از الگوریتمی مشابه الگوریتم ترسیم نقاط نمونه برداری، نقاط شبکه نمایش داده می شوند.

شبکه ساخته شده به ترتیب فوق به همراه تنظیمات گرافیکی دیگر نظیر وضعیت نمایش نقاط نمونه برداری و ... به آرایه پایه و این آرایه مانند قبل به مبدل گرافیکی ارسال خواهد گشت<sup>۱۷</sup>.

توجه به این نکته ضروری است که در الگوریتم کریجینگ آرایه داده ها ممکن است به صورت تغییر یافته به حالت توزیع نرمال یا حالت شاخص ارسال گردد که بسته به تنظیمات آرایه پایه، ممکن است در پایان بازگشت مقادیر به حالت اول منظور گردد.

عملیات تخمین بروش کریجینگ همانطور که قبلاً هم اشاره شد، به طور خلاصه شامل ایجاد و حل معادله  $\Gamma A = \Gamma_0$  برای هر کدام از نقاط شبکه، در این الگوریتم مقادیر ماتریسهای  $\Gamma$  و  $\Gamma_0$  به ازای هر نقطه شبکه ایجاد و مقدار ماتریس  $A$  با حل معادله فوق محاسبه می شود و در نهایت مقادیر نقاط شبکه تخمین زده شده و واریانس تخمین محاسبه می گردد.

---

۱۷- ذخیره سازی شبکه ساخته شده در آرایه پایه به واسطه یک متغیر از نوع Enhanced windows metafile(emf) انجام می گیرد.

در حال معادله فوق از روشی مشابه روش گوس - جردن<sup>۱۸</sup> استفاده شده است [۱۳]. روش گوس - جردن یکی از روشهای مرسوم در حل دستگاههای معادلات است که با تعویض سطرها، ضرب یک عدد ثابت در هر سطر و جمع مقادیر سطرها به یک ماتریس قطری دست خواهیم یافت و در نتیجه مقادیر ایجاد شده در ماتریس  $\Gamma_0$  مقادیر جواب خواهند بود [۱۳].

الگوریتم بکار گرفته شده در نرم افزار حاضر، با انجام عملیات خطی در ماتریس  $\Gamma$  به یک ماتریس مثلثی با ضرایب صفر در پایین قطر دست می یابد و سپس با حل معادلات یک مجهولی از پایین ترین خط ماتریس محاسبه مقادیر  $\lambda$  (ماتریس  $A$ ) انجام می گیرد.

این روش در عمل سرعت بیشتری از روش گوس - جردن نشان داده است.

فلوچارت الگوریتم تخمین بروش کریجینگ در شکل (۳-۱۱) آورده شده و مراحل اصلی

الگوریتم تخمین به شرح زیر می باشد:

الف) بررسی آرایه داده های ورودی و اطمینان از وجود مقادیر معتبر در فایل داده ها.

ب) تعریف متغیرها و آرایه های محلی و شناسایی آرایه پایه.

ج) ایجاد ماتریس ضرایب برای کل داده ها ( $\Gamma$ ).

د) تبدیل مختصات مکانی داده ها به مختصات در خور صفحه نمایش.

ه) انتخاب اولین نقطه شبکه بعنوان نقطه جاری و جستجوی نقاط داده اولیه که با توجه به

انیزوتروپی، تعداد قطاع و تعداد نقاط داده مجاز در هر قطاع که در محدوده تأثیر نقطه جاری شبکه

قرار می گیرند.

ز) ایجاد ماتریس ضرایب جدید ( $\Gamma$ ) برای داده های انتخاب شده در مرحله (و).

ح) ایجاد ماتریس طرف دوم معادله ( $\Gamma_0$ ) بازای داده های انتخاب شده در مرحله (و).

ط) ارسال ماتریس های  $\Gamma$  و  $\Gamma_0$  به الگوریتم حل معادله و دریافت ماتریس ضرایب ( $A$ ).

ی) محاسبه مقدار نقطه جاری شبکه و واریانس تخمین.

ک) انتخاب نقطه بعدی شبکه و بازگشت به مرحله (و) تا پوشش کامل شبکه.

ل) تبدیل معکوس مقادیر شبکه در صورت لزوم.

م) بررسی حدود تغییرات مقادیر تخمین.

ن) انتقال مقادیر متغیرها و آرایه ها به آرایه پایه

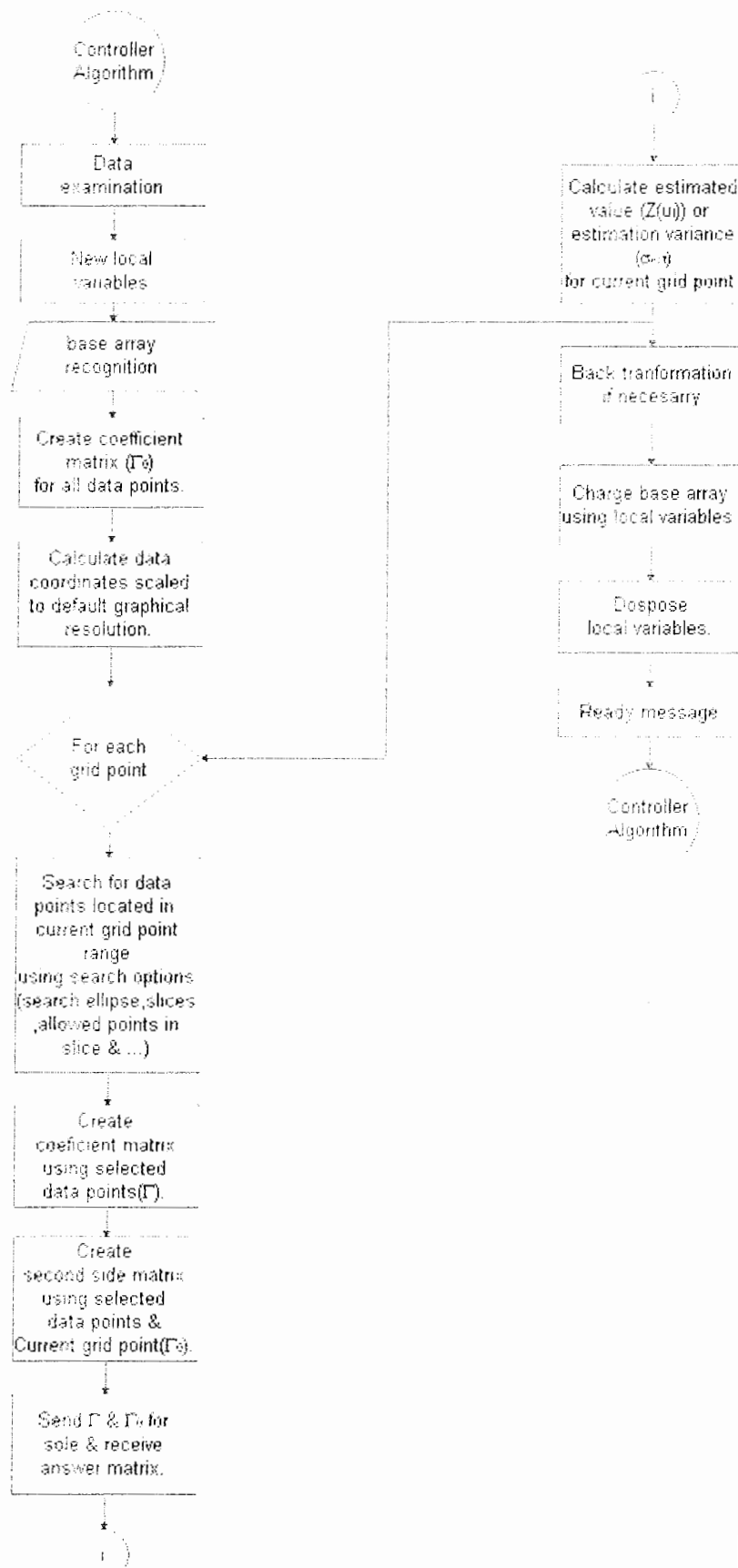
س) نابودی مقادیر محلی.

ع) اعلام پایان عملیات.

ف) بازگشت به الگوریتم کنترل کننده جهت مراحل بعدی.

کد برنامه بدنه اصلی برنامه به پیوست (الف - ۴) می باشد.





شکل ۳-۱۱) فلوجارت الگوریتم تخمین بروش کریجینگ

### ۳-۳-۵- الگوریتم شبیه سازی مرحله ای

این الگوریتم با هدف شبیه سازی گوسی و شاخص طراحی شده است. مقادیر داده به صورت نرمال یا شاخص به الگوریتم وارد شده و شبکه مقادیر شبیه سازی شده ایجاد می گردد. رویه نمایش و استفاده از آرایه پایه مشابه الگوریتم کریجینگ می باشد. کلیات الگوریتم نیز به الگوریتم تخمین شباهت دارد، با درج تفاوت هایی که لازمه عملیات شبیه سازی مرحله ای است. ابتدا مختصات نقاط شبکه بر حسب ابعاد شبکه در یک آرایه تک بعدی قرار داده می شود. سپس اعضای این آرایه به طور تصادفی جابجا شده و با ایجاد حلقه ای که یک به یک اعضای آن را احضار می نماید، مجموعه نقاط شبکه به صورت تصادفی انتخاب می گردد.

برای هر نقطه که به ترتیب فوق و در ردیف تصادفی انتخاب شده است تخمین بروش کریجینگ (معمولی، نرمال یا شاخص) انجام خواهد گرفت. بدین منظور می توان با احضار الگوریتم کریجینگ مقدار نقطه را تخمین زد، لیکن با توجه به زمان بر بودن این عملیات، الگوریتم کریجینگ با ایجاد یک سری تغییرات و به طور خلاصه در الگوریتم شبیه سازی تکرار شده است. از آنجایی که در الگوریتم شبیه سازی ابتدا داده های اولیه در شبکه قرار داده می شوند، لذا روش جستجوی الگوریتم شبیه سازی صرفاً براساس نقاط شبکه است و داده های اولیه؛ بر خلاف روش جستجو در الگوریتم کریجینگ، به طور مستقیم در آن نقشی ندارد. در پایان نیز توزیع نرمال با میانگین مقدار تخمین و واریانس برابر با خطای تخمین ایجاد می شود و مقدار حاصل از نمونه برداری تصادفی از این توزیع به نقطه جاری شبکه اختصاص داده خواهد شد و از این پس این نقطه نیز جزو نقاط داده محسوب خواهد گشت. زیرا عملیات جستجو در نقاط شبکه انجام می گیرد. این الگوریتم سرعت عملیات را نسبت به روش احضار الگوریتم کریجینگ برای هر نقطه به طرز چشمگیری افزایش می دهد. هر چند محدودیتهایی را به نرم افزار اعمال می نماید که اصلی ترین آن لزوم انطباق نقاط داده اولیه با شبکه می باشد. به منظور نمونه برداری از توزیع نرمال با آیتم های  $[Z(u_i), \sigma_{Est}]$  نیاز به تابع توزیع نرمال معکوس داریم. با انتخاب یک مقدار تصادفی به عنوان احتمال (بین ۰ و ۱) و قرار دادن آن در تابع توزیع نرمال معکوس مقدار نقطه شبکه حدس زده می شود.

رابطه مورد استفاده جهت محاسبه مقدار توزیع نرمال معکوس به ازای احتمال  $p$  و با

مقادیر میانگین برابر با  $Z(u_i)$  و واریانس  $\sigma_{Est}$  بدین شکل است (اسمیت و تیلور<sup>۱۹</sup>، ۱۹۷۰) [۱۵]

If  $p \leq 0.5$

$$\nu = \sqrt{-2 \ln(p)} \quad (9-3)$$

$$x = m - \sigma \left( \nu - \frac{2.515515 + 0.802853\nu + 0.010328\nu^2}{1 + 1.432788\nu + 0.189269\nu^2 + 0.001308\nu^3} \right)$$

و

If  $p > 0.5$

$$\nu = \sqrt{-2 \ln(1-p)} \quad (10-3)$$

$$x = m + \sigma \left( \nu - \frac{2.515515 + 0.802853\nu + 0.010328\nu^2}{1 + 1.432788\nu + 0.189269\nu^2 + 0.001308\nu^3} \right)$$

مراحل اصلی الگوریتم بشرح زیر می باشد :

(الف) بررسی آرایه داده های ورودی و اطمینان از وجود مقادیر معتبر در فایل داده ها.

(ب) تعریف متغیرها و آرایه های محلی و شناسایی آرایه پایه.

(ج) انطباق نقاط داده اصلی با شبکه تعریف شده.

(د) تبدیل مختصات مکانی داده ها به مختصات در خور صفحه نمایش پیش.

(ه) قراردادن مقادیر مختصات نقاط شبکه (بر حسب ابعاد شبکه) در یک آرایه تک بعدی.

(و) جا به جایی اعضای آرایه تک بعدی مذکور جهت دستیابی به ترتیب تصادفی نقاط

شبکه.

(ز) انتخاب نقاط شبکه به ترتیب درج شده در آرایه تک بعدی مذکور.

(ح) جستجو برای نقاط دارای مقدار در شبکه با توجه به انیزوتروپی، قطاع و تعداد نقاط

مجاز در هر قطاع.

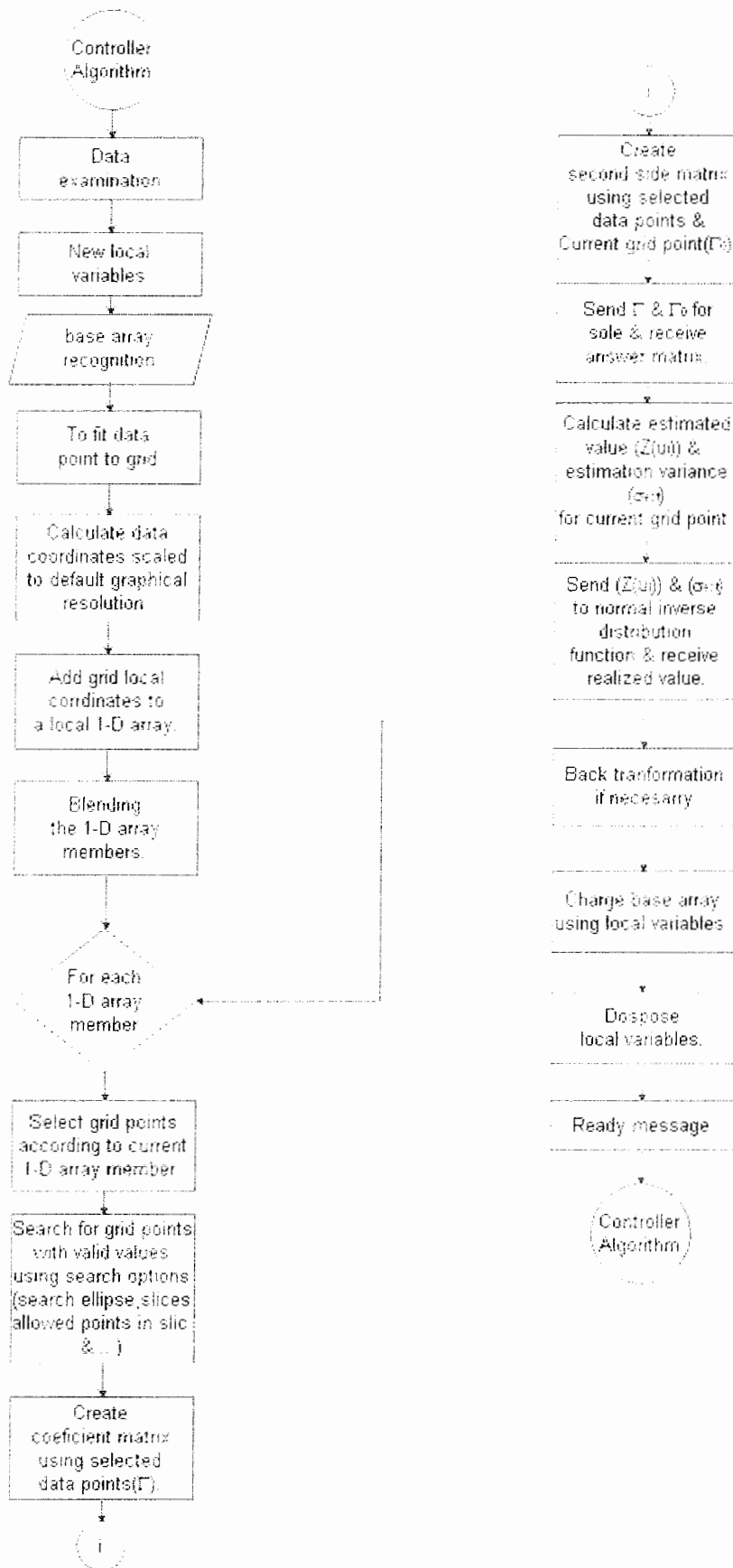
(ط) ایجاد ماتریس ضرایب ( $I'$ ) برای نقاط انتخاب شده در مرحله (ح).

(ی) ایجاد ماتریس طرف دوم معادله ( $I_0$ ) به ازای نقاط انتخاب شده در مرحله (ح).

(ک) ارسال ماتریس های مذکور ( $I'$  و  $I_0$ ) به الگوریتم حل معادله و دریافت ماتریس

ضرایب ( $A$ ).

- ل) محاسبه مقدار تخمین و واریانس تخمین برای نقطه جاری.
- م) ارسال مقادیر واریانس و مقدار به تابع توزیع معکوس نرمال، نمونه برداری تصادفی از آن و دریافت مقدار برای نقطه جاری.
- ن) بازگشت به مرحله (ز) تا آخرین عضو آرایه تک بعدی.
- ص) تبدیل معکوس مقادیر شبکه در صورت نیاز.
- ع) انتقال مقادیر متغیرها و آرایه ها به آرایه پایه.
- ف) نابودی متغیرها و مقادیر محلی.
- ذ) اعلام پایان برنامه.
- ق) بازگشت به الگوریتم کنترل کننده جهت مراحل بعدی عملیات.
- فلوچارت برنامه در شکل ۳-۱۲ و کد برنامه آن در پیوست (الف-۵) آورده شده است.



شکل ۳-۱۲) فلوچارت الگوریتم شبیه سازی مرحله ای

### ۳-۳-۶- الگوریتم شبیه سازی انیلینگ

الگوریتم کلی بکار گرفته شده در شبیه سازی فرایند انیلینگ مطابق الگوریتم استاندارد این روش می باشد یعنی پس از ایجاد تابع توزیع تجمعی احتمال مقادیر اولیه و نمونه برداری تصادفی از آن، مقادیر شبکه را در نقاطی که با نقاط نمونه برداری اولیه مطابقت ندارد مقدار دهی می نماییم و با تعریف یک تابع هدف، جابجایی نقاط مذکور و محاسبه مقدار تابع هدف سعی در هر چه کوچکتر نمودن مقدار آن می نماییم. بدین ترتیب که در صورت کوچکتر شدن تابع هدف پس از هر بار جابجایی نقاط، شبکه جدید تایید می گردد و در غیر این صورت احتمال پذیرش جابجایی محاسبه شده (رابطه ۲-۱۱) و بر آن اساس جابجایی پذیرفته شده و یا اینکه رد می شود و عملیات جابجایی نقاط بازگردانده خواهد شد. این عملیات بازای مقادیر مختلف درجه حرارت (پارامتر  $t$ ) انجام می شود. در این الگوریتم دما از میزان حداکثر (۸۰۰ درجه) با ضریب  $0/8$  تا دمای حداقل (۱۱۰ درجه) کاهش می یابد.

تابع هدف کلاسیک در زمین آمار مطابق تساوی زیر می باشد [۹]:

$$\sum_{i=1}^{n_h} [\gamma^*(h_i) - \gamma(h_i)]^2 \quad (11-3)$$

که  $\gamma^*(h_i)$  واریوگرام شبکه شبیه سازی شده و  $\gamma(h)$  مقادیر واریوگرام بازای مقادیر اولیه می باشد.

الگوریتم انیلینگ با استفاده از تابع هدف مذکور صرف زمان طولانی که بعضاً از عهده پردازنده سیستم خارج می باشد نتایج مطلوب تولید خواهد کرد. علت نیز پیچیدگی محاسبات واریوگرافی و همچنین تغییر ناچیزی است که به ازای جابجایی نقاط شبکه در مقادیر آن ایجاد می شود. به همین دلیل روش انیلینگ معمولاً بعنوان یک فرآیند ثانویه (بهینه سازی) بر روی تصاویر یا حالت‌های تحقق یافته سایر روش های شبیه سازی مورد توجه است.

مراحل اصلی الگوریتم بشرح زیر می باشد :

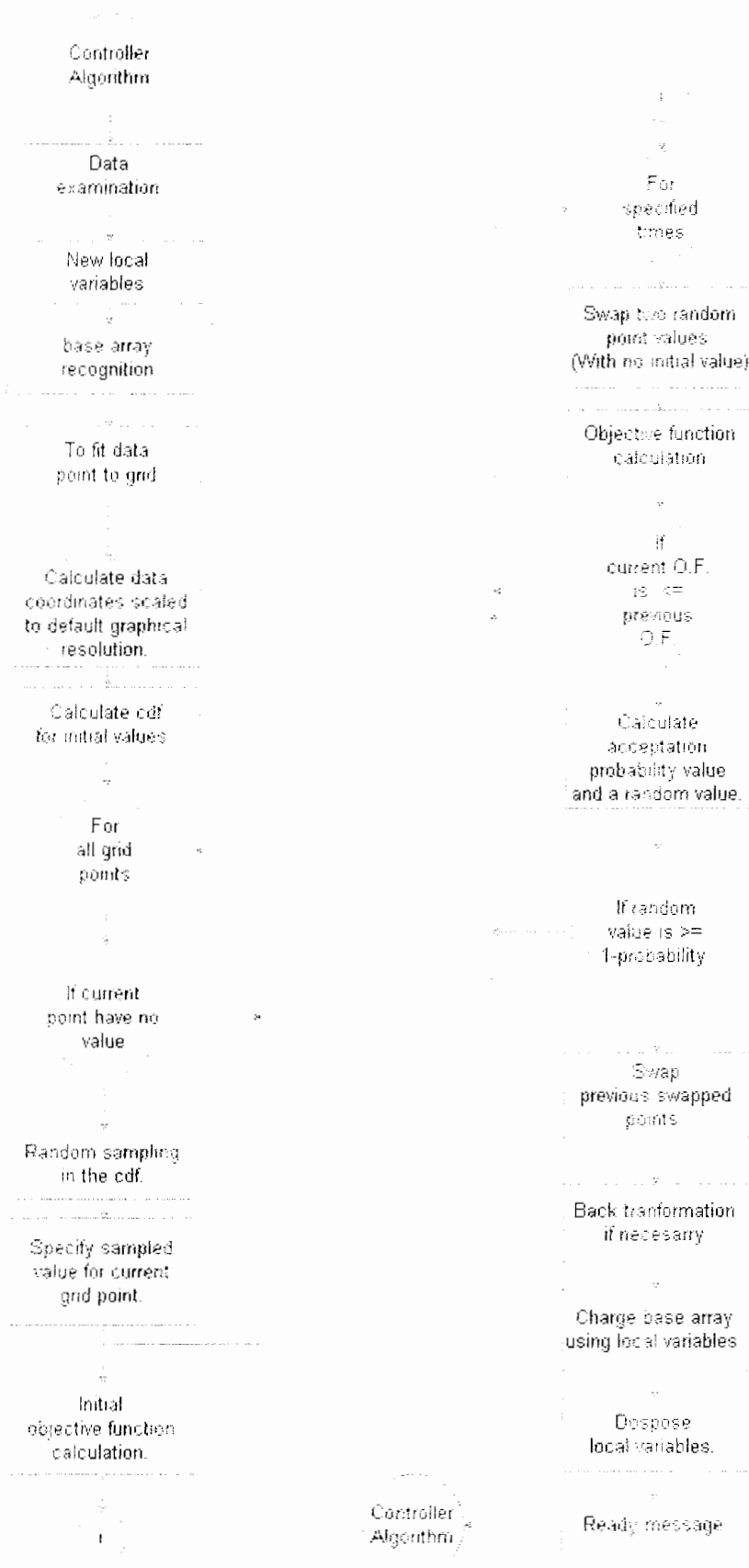
الف) بررسی آرایه داده های ورودی و اطمینان از وجود مقادیر معتبر در فایل داده ها.

ب) تعریف متغیرها و آرایه های محلی و شناسایی آرایه پایه.

ج) انطباق نقاط داده اصلی با شبکه تعریف شده.

د) تبدیل مختصات مکانی داده ها به مختصات در خور صفحه نمایش پیش فرض.

- ه) محاسبه مقادیر توزیع احتمال تجمعی برای داده های اولیه.
- و) نمونه برداری تصادفی از توزیع احتمال فوق و اختصاص مقدار حاصل به نقاط فاقد مقدار در شبکه.
- ز) بازگشت به مرحله (و) تا پوشش کامل شبکه.
- ح) تعریف مقدار دمای اولیه.
- ط) محاسبه مقدار تابع هدف برای شبکه حاصل.
- ی) جابجایی دو نقطه از شبکه به طور تصادفی و به شرط این که منطبق بر داده های اصلی نباشند.
- ک) محاسبه مجدد تابع هدف.
- م) اگر تابع هدف کوچکتر شده بازگشت به مرحله (ی) تا دستیابی به دفعات مشخص شده تکرار عملیات.
- م) اگر تابع هدف کوچکتر نشده باشد، احتمال پذیرش جابجایی محاسبه و یک عدد تصادفی بین ۰ و ۱ تولید می گردد. در صورتی که عدد تصادفی مذکور در حوضه احتمال پذیرش جابجایی قرار نگرفته باشد جابجایی دو نقطه مورد قبول قرار نمی گیرد. سپس بازگشت به مرحله (ی) تا دستیابی به دفعات مشخص شده تکرار عملیات.
- ن) کاهش مقدار منظور شده بعنوان دما و بازگشت به مرحله (ی) تا زمان دستیابی به دمای حداقل.
- ص) تبدیل معکوس مقادیر شبکه در صورت نیاز.
- ع) انتقال مقادیر متغیرها و آرایه ها به آرایه پایه.
- ف) نابودی متغیرها و مقادیر محلی.
- ض) اعلام پایان برنامه.
- ق) بازگشت به الگوریتم کنترل کننده جهت مراحل بعدی عملیات.
- فلوچارت الگوریتم در شکل (۳-۱۳) و کد برنامه آن در پیوست (الف-۶) آورده شده است.



شکل ۳-۱۳) فلوجارت الگوریتم شبیه سازی انیلینگ



### ۳-۳-۷- الگوریتم ترکیب شبکه ها

الگوریتم ترکیب شبکه ها، الگوریتم ساده ای است که با دریافت چندین گراف حاوی شبکه از محیط نرم افزار و ترکیب مقادیر آنها طی عملیات مشخص یک شبکه نهایی ایجاد می نماید. این شبکه می تواند حاوی جمع، میانگین یا واریانس مقادیر چندین شبکه باشد. رویه ایجاد و نمایش شبکه، نوع متغیرها در آرایه پایه این الگوریتم مشابه الگوریتم های شبکه ساز قبلی است.

در این الگوریتم ابتدا دو ماتریس که ابعاد متناظر با ابعاد شبکه مرکب دارند، ایجاد می شود که یک ماتریس حاوی جمع مقادیر شبکه ها و دیگری حاوی جمع توان دوم مقادیر شبکه ها خواهد بود.

الگوریتم با احضار آرایه پایه چارتهای مورد نظر و دریافت مقادیر شبکه آنها، نقاط نظیر شبکه مرکب را در هر چارت شناسایی کرده و بسته به نوع عملیات هدف (جمع، میانگین یا واریانس) مقادیر شبکه نهایی را محاسبه می نماید. اگر هدف نمایش جمع مقادیر چندین شبکه باشد، مقادیر موجود در ماتریس اول مستقیماً در شبکه مرکب قرار می گیرد. اگر هدف نمایش میانگین مقادیر شبکه ها باشد، مقادیر موجود در ماتریس اول به تعداد چارتهای تقسیم شده و در شبکه نهایی قرار می گیرد.

اگر هدف نمایش توزیع واریانس مقادیر چندین شبکه باشد، با استفاده از فرمول

$$\sigma^2 = (\overline{x^2}) - (\overline{x})^2 \quad (۳-۱۳)$$

مقادیر در ماتریس ترکیب و در شبکه نهایی قرار خواهد گرفت.

مراحل الگوریتم ترکیب شبکه ها بدین قرار می باشد.

الف) بررسی آرایه داده های ورودی و اطمینان از وجود مقادیر معتبر در فایل داده ها.

ب) تعریف متغیرها و آرایه های محلی.

ج) تبدیل مختصات مکانی داده ها به مختصات در خور صفحه نمایش پیش فرض.

د) شناسایی آرایه پایه و استخراج گرافهای مورد نظر جهت ترکیب و اطمینان از وجود

گرافهای مورد نظر.

ه) دریافت آرایه پایه مربوط به شبکه جاری.

و) انتخاب اولین نقطه شبکه مرکب.

ز) انتخاب مقدار معادل نقطه جاری شبکه مرکب از گراف احضار شده جاری.

ح) افزودن مقدار استخراج شده به عضو معادل نقطه شبکه مرکب در یک ماتریس و توان

دوم آن در ماتریس دیگر.

ط) انتخاب نقطه دیگری از شبکه و بازگشت به مرحله (و) تا پوشش کامل شبکه.

ی) بازگشت به مرحله (ه) تا آخرین گراف مورد نظر.

ک) محاسبه مقدار میانگین یا واریانس برای نقاط شبکه مورد نظر.

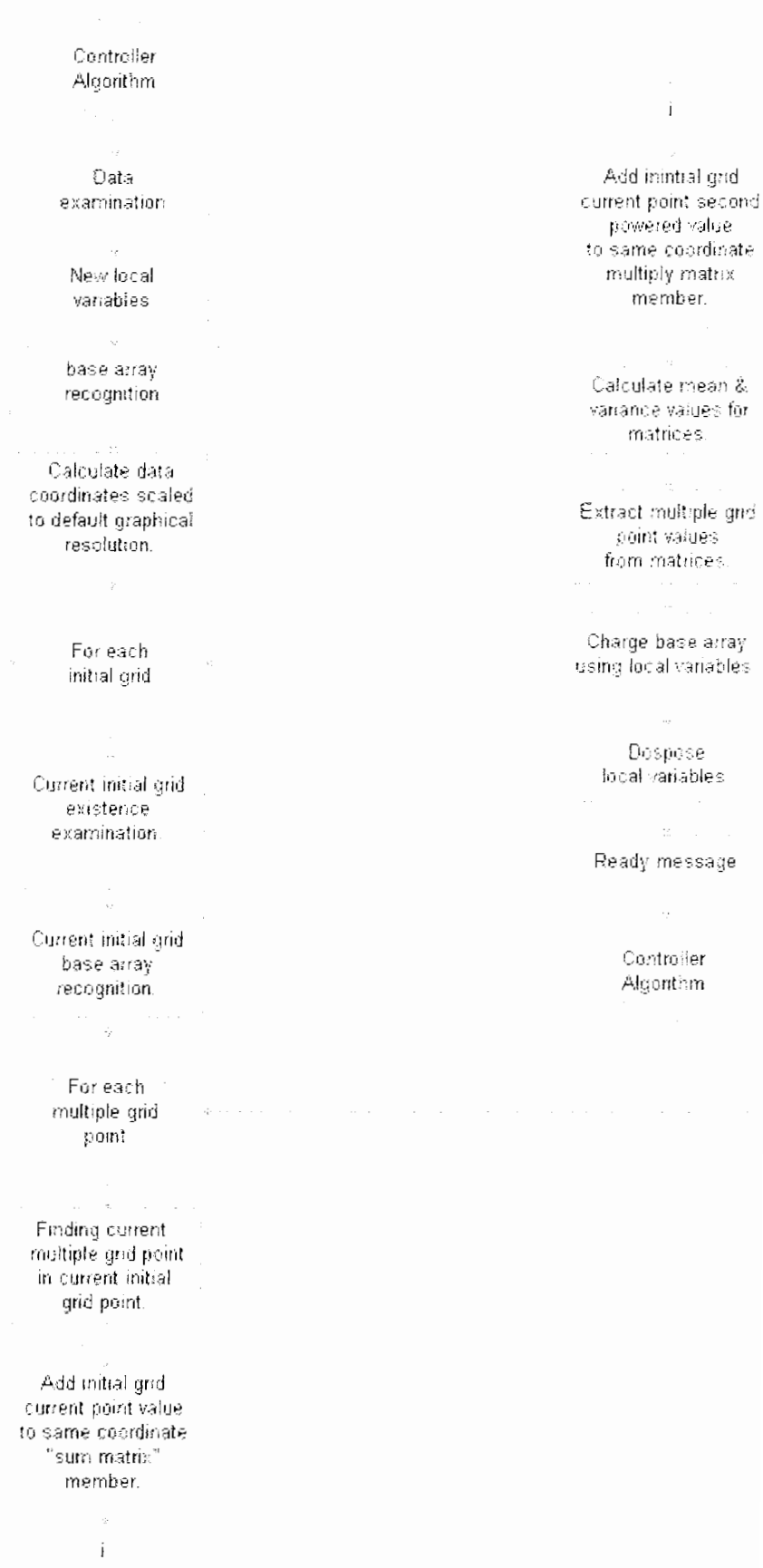
م) انتقال مقادیر متغیرها و آرایه‌ها به آرایه پایه.

ن) نابودی متغیرها و مقادیر محلی و رها سازی آرایه پایه گرافهای اولیه.

س) اعلام پایان برنامه.

ع) بازگشت به الگوریتم کنترل کننده جهت مراحل بعدی عملیات.

فلوچارت برنامه در شکل ۳-۱۴ و کد برنامه آن در پیوست (الف-۷) آورده شده است.



شکل ۳-۱۴) فلوچارت الگوریتم ترکیب شبکه ها

## فصل چهارم

نرم افزار Gridat، نصب، راه اندازی و روش اجرا

## فصل چهارم

### نرم افزار Gridat، نصب، راه اندازی و روش اجرا

#### ۴-۱- معرفی نرم افزار

تفکر ارائه نرم افزار موسوم به Gridat با هدف اجرای پروژه های زمین آماری بوده و در طراحی آن سعی شده علاوه بر حفظ سادگی محیط اجرا و خلاصه بودن مجموعه ابزار یک پروژه کامل زمین آماری پوشش داده شود.

فراخوانی، ایجاد و ویرایش داده ها پس از اجرای عملیات، نمایش نتایج و همچنین امکان ایجاد تغییر در چارتهای و گرافهای ایجاد شده از ویژگی های نرم افزار می باشد.

قابل توجه است که در این فصل به کلیه نتایج گرافیکی اعم از نمودار و نقشه نام گراف و به فرم حاوی داده ها، فرم نمایشگر داده ها اطلاق شده است. فرم نمایشگر داده اطلاعات را از فایل داده دریافت می دارد که فایل داده ها خود شامل چندین سری داده مجزا و مستقل از هم می باشد و در قالب صفحات گسترده<sup>۱</sup> در فایل داده ذخیره می شوند.

#### ۴-۲- نصب نرم افزار

نصب نرم افزار در محیط ویندوز و مشابه دیگر نرم افزارهای معمول انجام می گیرد. با اجرای فایل نصب کننده<sup>۲</sup> عملیات نصب آغاز می گردد. پس از گذر از کادرهای محاوره ای معمول، فایلها و اجزای لازم جهت اجرای نرم افزار در حافظه ذخیره شده و اطلاعات لازم در رجیستری

---

1 - Spreadsheet

2 - Setup

سیستم ثبت می گردد. قبل از آغاز عملیات نصب، فایل نصب کننده بررسی هایی جهت حداقل ملزومات سیستم انجام می دهد. با توجه به اینکه نرم افزار در محیط Visual studio Net طراحی شده است حداقل ملزومات سیستم جهت اجرای آن عبارتند از :

- ✓ Internet Explorer 5.5 or higher
- ✓ Microsoft.Net Framework 1.1 or higher

در صورت عدم وجود هر کدام از موارد فوق در سیستم، نصب کننده قادر به بارگذاری نرم افزار نخواهد بود و عملیات نصب ناتمام می ماند. موارد فوق از شبکه اینترنت و به طور رایگان قابل دریافت می باشند.<sup>۳</sup>

همچنین جهت نمایش فایل داده ها به مجموعه ابزار میکروسافت آفیس، ویرایش ۱۰ (OWC10)<sup>۴</sup> نیاز خواهد بود. با وجودی که اگر این مورد در سیستم موجود نباشد، فرآیند نصب به طور کامل انجام خواهد شد، لیکن اجرای صحیح نرم افزار ممکن نخواهد بود. در صورتی که سیستم به مجموعه میکروسافت آفیس XP مجهز باشد این مجموعه نیز خواه و ناخواه موجود است و در غیر اینصورت می توان از طریق شبکه اینترنت مجموعه OWC10 را به طور رایگان دریافت نمود.<sup>۵</sup>

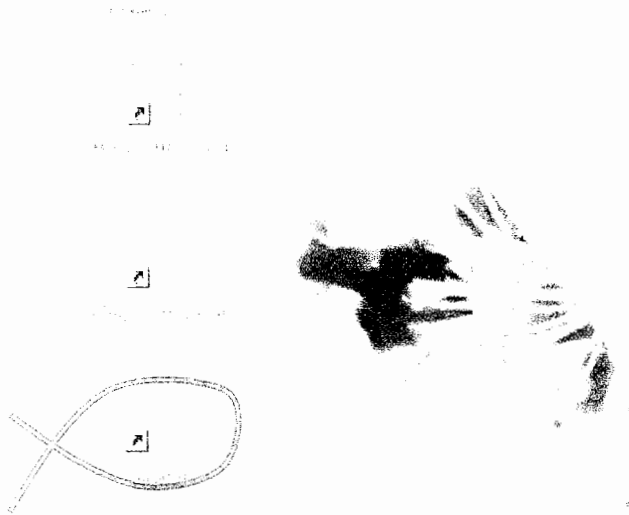
با رعایت موارد فوق نرم افزار بدرستی نصب شده و اجرای آن از طریق منوی Start و یا با استفاده از شمایل ظاهر شده در روی صفحه نمایش (شکل ۴-۱) ممکن می باشد.

---

3 - <http://www.microsoft.com/windows/ie/downloads>

4 - Office Web Component 10

5 - <http://www.microsoft.com/downloads>



شکل ۴-۱) نرم افزار Gridat نصب شده و آماده اجرا می باشد.

#### ۴-۳- قطعات اصلی و روش کلی اجرای Gridat

اجرای نرم افزار با ظهور پنجره معرفی آغاز می گردد. در ابتدا که هیچ پروژه ای فعال نیست گزینه های محدودی از منوها را می توان انتخاب نمود. از جمله گزینه New که به منظور ایجاد پروژه جدید بکار می رود و یا گزینه Open که پروژه از قبل ایجاد شده ای را فراخوانی می کند. با فعال شدن یک پروژه می توان به گزینه های دیگر منوها دسترسی پیدا کرد. ایجاد پروژه با نمایش یک فایل داده که فاقد هر نوع مقداری است آغاز می گردد. می توان با وارد نمودن مقادیر اولیه در این فایل، داده ها را به پروژه افزود. ترتیب ثبت داده به صورت ستونی می باشد، یعنی یک ستون به مختصات داده ها در جهت شرقی-غربی، یک ستون به مختصات شمالی جنوبی و یک ستون به مقادیر داده های متناظر با مختصات اختصاص داده خواهد شد. اگر فایل داده ها را قبلاً توسط نرم افزار دیگری ایجاد شده باشد؛ به شرط اینکه فایل مزبور دارای پسوندهای مناسب نظیر .dat، .txt یا .xml باشد، می توان در منوی فایل به جای گزینه New گزینه Import را انتخاب نمود. این گزینه امکان انتقال داده ها از یک فایل داده موجود را

فراهم می آورد. بایستی توجه داشت این گزینه تنها زمانی فعال است که پروژه ای فعال نباشد. برای خروج از یک پروژه می توان گزینه Close را از منوی File انتخاب نمود.

فشردن کلید Save یا انتخاب این گزینه از منوی File منجر به ذخیره سازی پروژه می گردد. هر پروژه ایجاد شده در Gridat شامل دو فایل مجزاست، یک فایل داده و یک فایل شامل مقادیر ثبت شده در آرایه پایه. اجرای گزینه های عملیاتی نرم افزار و ذخیره نمودن پروژه منجر به تبدیل اطلاعات آرایه پایه به کد دودویی<sup>۶</sup> و درج آن در قالب این فایل می گردد. اجازه دهید قبل از پرداختن به جزئیات اجرا، قسمت های اصلی نرم افزار و ارتباط آنها با یکدیگر را بررسی کنیم.

همان طور که قبلاً نیز اشاره شد، نرم افزار شامل چهار قطعه اصلی می باشد.

- ✓ فرم نمایشگر داده ها که شامل نمایش فایل داده ها است.
  - ✓ فرم نمایشگر گرافیکی که به تعداد لازم ایجاد می گردد.
  - ✓ کادر محاوره ای تنظیمات که گزینه های لازم جهت عملیات اجرایی پروژه و تنظیمات گرافیکی را ارائه می دهد.
  - ✓ کادر اصلی که محیط اولیه نرم افزار است و دیگر قطعات وابسته به آن می باشد.
- واضح است که عملکرد این قطعات در رابطه با یکدیگر منجر به ایجاد خروجی نرم افزار و نتیجه گیری نهایی خواهد شد.

فرم نمایشگر داده ها حاوی کنترل نمایشگر داده و مستقیماً در ارتباط با فایل داده های پروژه است و می تواند شامل چندین سری داده مجزا باشد که هر کدام به طور مستقل در محیط نرم افزار قابل نمایش خواهند بود. در هر کدام از این صفحات امکان ایجاد فایل داده با<sup>۱۶</sup> ردیف داده ممکن می باشد. الگوریتم کنترل کننده نیز قادر است اطلاعات را به طور مجزا از فایل داده دریافت نماید و پس از ارسال آن به الگوریتم های عملیاتی نتایج را به صورت تصاویر در فرم نمایش گرافیکی ارائه دهد.

کنترل نمایشگر داده از ابزار میکروسافت آفیس به امانت گرفته شده و دارای تنظیمات مختص خود می باشد که در مورد کادر محاوره ای تنظیمات آن صحبت خواهد شد. دستیابی به





بعضی ابزارهای این کنترل با راست کلیک نمودن بر روی آن و انتخاب گزینه های منوی شناور ظاهر شده ممکن می باشد. به عنوان مثال افزودن و حذف خطوط و یا ستونها در فایل داده ها از طریق انتخاب زیر گزینه های دو گزینه Delete و Insert امکان پذیر است.

فرم نمایش نتایج که در فصل قبل از آن با نام مبدل گرافیکی نام برده شد، قدرت تفکیک انواع گراف اعم از نمودار و نقشه را داشته و در عین سادگی ظاهر، امکانات تنظیم گرافیکی متنوعی در اختیار کاربران قرار می دهد. در مورد این فرم و تنظیمات آن نیز به واسطه منوی Action توضیحاتی داده خواهد شد. این فرم حاوی الگوریتم قدرتمندی جهت تبدیل مقادیر عددی به گرافیکی می باشد که خود در نقش یک نرم افزار مستقل ایفای نقش می نماید.


فرم نمایشگر علاوه بر موارد یاد شده شامل یک جعبه ابزار و یک منوی شناور می باشد.


جعبه ابزار یاد شده شامل کلیدهایی بدین شرح است:



کلید  نقش انتخاب گزینه Export از منوی File را ایفا می کند.


کلید  نقش انتخاب گزینه Copy از منوی Edit را ایفا می کند.

کلید  نقش انتخاب گزینه Paste از منوی Edit را ایفا می کند.

کلید  کادر محاوره ای تنظیمات را نمایش می دهد.

کلید  سری داده ای که گراف با آن مربوط است را فعال می نماید.

کلیدهای   متناظر با انتخاب گزینه های Undo و Redo می باشند.

کلید  فرم اطلاعات مربوط به کنترل گرافیکی را نمایش می دهد.

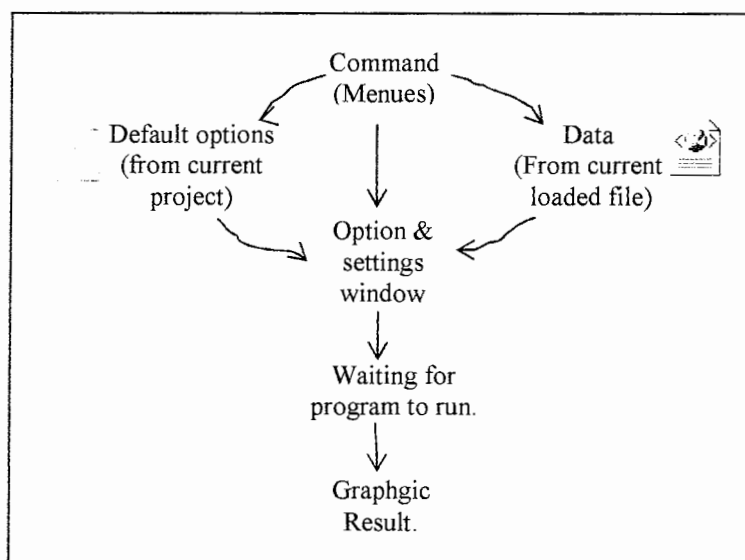
منوی شناور نمایشگر گرافیکی با راست کلیک نمودن بر روی آن نمایان می شود که شامل گزینه های سودمندی می باشد. مثلا جعبه ابزار یاد شده را می توان از طریق این منو نمایش داد یا محو نمود.

در فصل بعد و در حین اجرای عملی یک پروژه بیشتر با قابلیت های فرم نمایش گرافیکی آشنا خواهد شد.

اجزای یاد شده و دیگر عوامل نرم افزار همگی وابسته به پنجره اصلی می باشند. این پنجره به عبارتی رابط گرافیکی الگوریتم کنترل کننده است و با در برداشتن منوها و ابزار مختلف

امکان صدور فرامین را برای کاربران فراهم آورده و وظیفه حفظ ارتباط بین اجزاء، سازماندهی و مدیریت پروژه را به عهده دارد و از این بابت آسودگی خاطر کاربران را فراهم می آورد و کاربران نیازی به انجام عملیات اضافی جهت حفظ ارتباط بین عوامل پروژه را نخواهند داشت. شکل (۲-۳) در فصل سوم، گویای این ارتباط می باشد.

ارتباط اجزای نرم افزار از دید کاربر بدین ترتیب است که پس از افزودن فایل داده به پروژه، با انتخاب یکی از گزینه های مربوط به الگوریتم های عملیاتی (که در منوی Action قرار دارند) داده ها از فایل داده خوانده شده و کادر محاوره ای تنظیمات با تنظیمات پیش فرض ظاهر می گردد. الگوریتم های عملیاتی پارامترهای اولیه را از این کادر دریافت نموده و اجرا می شود. در نهایت نتایج در پنجره ای جداگانه نمایش داده می شود (شکل ۲-۴). مجموعه این عملیات و مراحل وابسته به کادر اصلی نرم افزار می باشد.



شکل (۲-۴) ارتباط قسمتهای اصلی Gridat از دید کاربر

اجزای کادر اصلی نرم افزار را بدین ترتیب است (شکل ۳-۴):

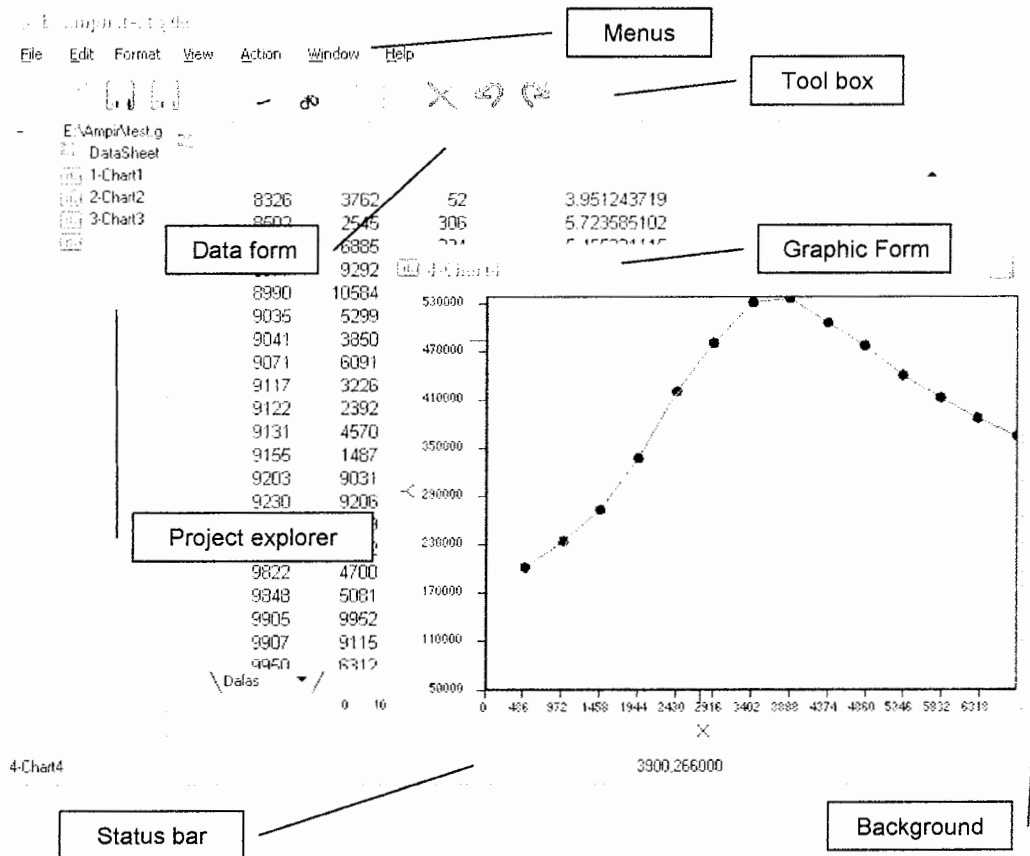
✓ منوها.

✓ جعبه ابزار.

✓ جستجو گر پروژه<sup>۹</sup>.

✓ میله نشانگر وضعیت<sup>۱۰</sup>.

✓ زمینه<sup>۱۱</sup>.



شکل ۴-۳) پنجره اصلی نرم افزار Gridat و قسمت‌های اصلی آن

#### ۴-۴- منوها در Gridat

مجموعه منوها وظیفه ارسال دستورها از طریق الگوریتم کنترل کننده را بعهده دارند. امروزه منوها جزء عناصر اصلی نرم افزارهای تحت ویندوز می باشند، مجموعه منوهای موجود در نرم افزار Gridat شامل گزینه های استاندارد ویندوز و تعدادی گزینه های اختصاصی نرم افزار است. عملکرد هر کدام از منوها به شرح ذیل است.

- 9 - Project explorer
- 10 - Status bar
- 11 - Back ground

#### ۴-۴-۱- منوی File

این منو جهت اجرای عملیاتی بارگذاری و ذخیره سازی پروژه و همچنین مجموعه عملیات چاپ بکار می رود. گزینه های این منو عبارتند از:

#### ۴-۴-۱-۱- گزینه New:

به منظور ایجاد یک پروژه جدید انتخاب می گردد. انتخاب این گزینه منجر به بسته شدن پروژه حاضر (در صورت فعال بودن) و ایجاد یک فایل داده خام خواهد شد.

#### ۴-۴-۱-۲- گزینه Open:

این گزینه به منظور اجرای یک پروژه از قبل ساخته شده بکار می رود. با انتخاب این منو کادر محاوره ای استاندارد ویندوز جهت بارگذاری پروژه ظاهر می گردد. در این کادر تنها فایلها با پسوند gda قابل شناسایی می باشند. فایل داده هر پروژه با پسوند xml. که در کنار فایل gda و با نامگذاری متناسب ذخیره شده است به طور همزمان با پروژه از حافظه خوانده و بارگذاری می گردد. لذا نیازی به اجرای مجزای فایل داده نمی باشد. توجه به این نکته ضروری است که حذف یا تغییر نام فایل داده ها یا پروژه از طریق محیط ویندوز می تواند منجر به نمایش پیغام یافت نشدن فایل داده در هنگام بارگذاری مجدد پروژه ظاهر گردد.

#### ۴-۴-۱-۳- گزینه CloseAll:

این گزینه پروژه جاری را می بندد. در صورتی که پروژه قبلاً ذخیره نشده باشد یا تغییراتی در آن ایجاد شده باشد، قبل از بستن آن در مورد ذخیره سازی سؤال خواهد شد.

#### ۴-۴-۱-۴- گزینه Save:

این گزینه بمنظور ذخیره سازی تغییرات ایجاد شده در پروژه جاری بکار می رود. اگر پروژه جدید باشد کادر محاوره ای استاندارد ذخیره سازی ویندوز ظاهر و پروژه با پسوند gda ذخیره خواهد شد. در کنار پروژه فایل داده با پسوند xml. به طور خودکار ایجاد می گردد. نام

گذاری فایل داده بر اساس نام پروژه با افزودن عبارت dat به انتهای آن انجام می‌گیرد مثلاً اگر نام پروژه name.gda باشد نام فایل داده namedat.xml خواهد بود. فایل داده مذکور با نرم افزارهای دیگر نظیر میکروسافت اکسل نیز قابل مشاهده و ویرایش است.

#### ۴-۴-۱-۵- گزینه Save As :

به منظور ذخیره سازی پروژه جاری با نام دیگر از این گزینه می‌توان استفاده نمود. پس از انجام این عملیات پروژه با نام جدید فعال خواهد شد.

#### ۴-۴-۱-۶- گزینه Export :

این گزینه به منظور ارسال مقادیر داده ها یا گرافها با فرمتهای مختلف بکار می‌رود. از طریق این منو می‌توان داده های جاری را به صورت فایل داده با پسوند هایی نظیر فایل داده استاندارد (dat)، صفحه گسترده اکسل (xls) و فرمت شبکه (html) و گرافها را با فرمتهای استاندارد گرافیکی (bmp, jpeg, gif, ...)، صفحه گسترده اکسل، فایل داده استاندارد و xml ارسال نمود. هدف از ذخیره سازی گرافها به صورت فایلهای داده دریافت مقادیر خام محاسبه شده چارتهای و شبکه ها است. به عنوان مثال ذخیره نمودن یک شبکه تخمین با فرمت صفحه گسترده اکسل منجر به ایجاد یک فایل شامل سه ستون داده ها می‌شود که در دو ستون اول مختصات هر کدام از نقاط شبکه و در ستون سوم مقادیر تخمینی آنها قرار خواهد داشت.

#### ۴-۴-۱-۷- گزینه Import :

این گزینه را تنها زمانی می‌توان انتخاب نمود که پروژه ای فعال نباشد (پروژه ای فراخوانی یا ایجاد نشده باشد). در این حالت تعداد محدودی از گزینه ها در دسترس هستند که گزینه Import یکی از آنها است. هدف از انتخاب این گزینه بارگذاری یک فایل داده از قبل ایجاد شده توسط نرم افزارهای دیگر یا فایل های داده پروژه های دیگر، بعنوان فایل داده یک پروژه جدید می‌باشد. با انتخاب این گزینه کادر محاوره ای استاندارد Open ظاهر شده و فایلهای داده با

فرمت xml و فایل داده استاندارد (dat) انتخاب می گردد و پس از ایجاد یک پروژه جدید فایل داده مذکور بعنوان داده های پروژه قرار می گیرد.

#### ۴-۴-۱-۸- گزینه **Page Setup** :

با انتخاب این گزینه کادر محاوره ای استاندارد تنظیمات صفحه جهت ارسال به چاپگر ظاهر می گردد. این کادر شامل اندازه کاغذ، جهت امتدادی کاغذ و اندازه حاشیه های آن می باشد.

#### ۴-۴-۱-۹- گزینه **Print Preview** :

با انتخاب این گزینه تصویر ارسالی به چاپگر در کادر استاندارد Print Preview ویندوز ظاهر می گردد. انتخاب این گزینه قبل از ارسال هر کدام از اجزای پروژه به چاپگر توصیه می شود.

#### ۴-۴-۱-۱۰- گزینه **Print** :

با انتخاب این گزینه تصویر جاری پروژه به چاپگر ارسال می گردد که می تواند نمایش هر کدام از گرافها یا سری های داده باشد. دستور چاپ از طریق کادر محاوره ای Print Preview نیز قابل ارسال است.

#### ۴-۴-۱-۱۱- گزینه **Exit** :

که به منظور بستن پروژه جاری و خروج کامل از نرم افزار انتخاب می گردد. در صورتی که تغییرات پروژه ذخیره نشده باشد، در مورد ذخیره سازی آن سؤال خواهد شد. همچنین این منو شامل چهار گزینه اضافی می باشد که مسیر و عناوین آخرین چهار پروژه ی اجرا شده می باشند.

#### **۴-۴-۲- Edit منوی**

منوی Edit شامل گزینه های استاندارد این منو در نرم افزارهای معمول ویندوز می باشد. مجموعه این گزینه ها مشتملند بر:

#### **۴-۴-۲-۱- Undo : گزینه**

به منظور بازگرداندن آخرین تغییرات ایجاد شده در پروژه بکار می رود.

#### **۴-۴-۲-۲- Redo : گزینه**

به منظور ابطال عمل گزینه Undo انتخاب می گردد.

#### **۴-۴-۲-۳- Copy : گزینه**

این گزینه مقدار داده های انتخاب شده در فایل داده یا تصویر موجود در گراف جاری را دریافت و در حافظه کلیپ بورد<sup>۱۲</sup> قرار می دهد.

#### **۴-۴-۲-۴- Paste : گزینه**

به منظور فراخوانی مقادیر کپی شده بکار می رود.

#### **۴-۴-۲-۵- Cut : گزینه**

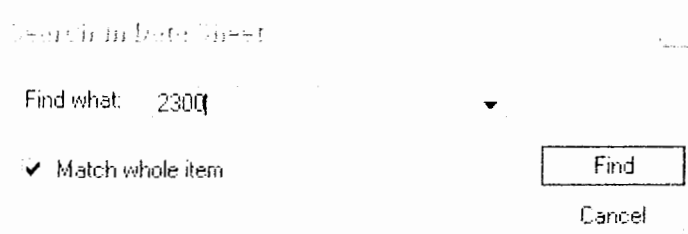
این گزینه عمل کپی و حذف مقادیر انتخاب شده را همزمان انجام می دهد. در حقیقت این گزینه به منظور جابجایی مقادیر بکار می آید. در مورد گرافها عمل حذف مقادیر از طریق این گزینه ممکن نمی باشد.

#### **۴-۴-۲-۶- Delete : گزینه**

به منظور حذف مقادیر انتخاب شده در فایل داده یا در گراف جاری انتخاب می شود.

#### ۴-۲-۷-۴-۴ گزینه Find :

به منظور جستجو در فایل داده ها این گزینه انتخاب می شود. جستجو از طریق کادر محاوره ای ظاهر شده (در شکل ۴-۴) انجام می گیرد.



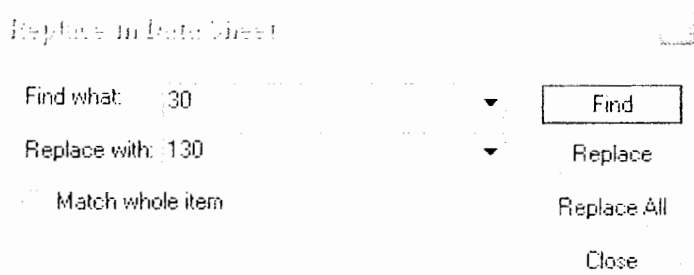
شکل ۴-۴) کادر محاوره ای Find

#### ۴-۲-۸-۴-۴ گزینه Find Next :

ادامه عملیات جستجو با شرایط تعیین شده قبلی با انتخاب این گزینه ممکن می باشد.

#### ۴-۲-۹-۴-۴ گزینه Replace :

جا به جایی عبارات یا اعداد در فایل داده ها با عبارات یا اعداد دیگر از طریق انتخاب این گزینه (کادر محاوره ای شکل ۴-۵) ممکن می باشد.



شکل ۴-۵) کادر محاوره ای Replace

#### ۴-۲-۱۰-۴-۴ گزینه Select All :

کل محدوده حاوی داده در فایل داده های جاری را انتخاب می نماید.



#### ۴-۴-۳- منوی Format

این منو شامل دو گزینه می باشد: گزینه Data sheet که خود شامل زیر گزینه های

Command and Options و Formula است و گزینه Chart.

همان طور که از عنوان این منو بر می آید هدف از اجرای گزینه های آن ایجاد تغییرات

در تنظیمات اجرائی پروژه است.

#### ۴-۴-۳-۱- گزینه Command and Options :

این گزینه برای تنظیمات گرافیکی کنترل نمایشگر فایل داده، افزودن سری جدید داده به

فایل داده ها و تغییر ویژگی های ظاهری آن بکار می رود.

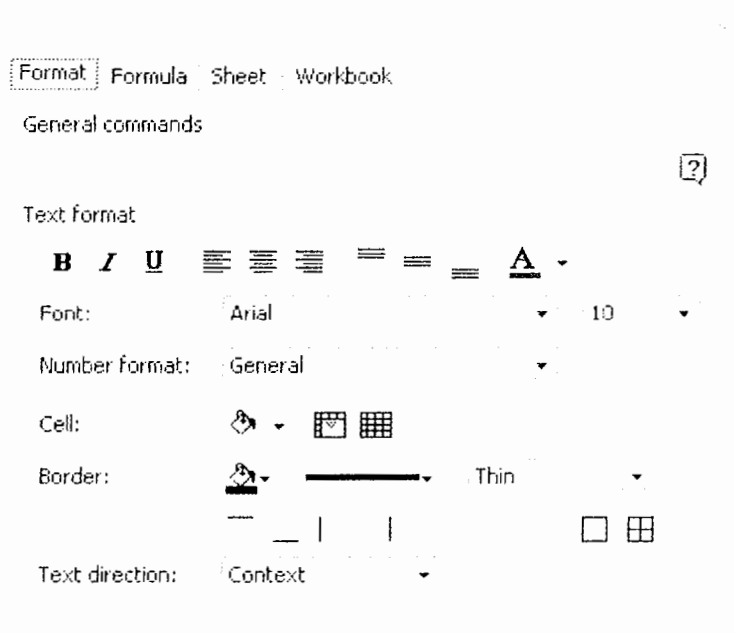
با انتخاب این گزینه کادر محاوره ای تنظیمات فایل داده ها ظاهر می گردد. این کادر

شامل چهار زبانه اصلی با عناوین Format , Formula , Sheet و Workbook می باشد.










زبانه اول (شکل ۴-۶) دارای امکاناتی است که برای تغییرات ظاهری در فایل داده ها بکار




گرفته می شوند. با انتخاب هر کدام از گزینه های موجود در این زبانه تغییرات به سلولهای انتخاب






شده فایل داده اعمال می گردد.




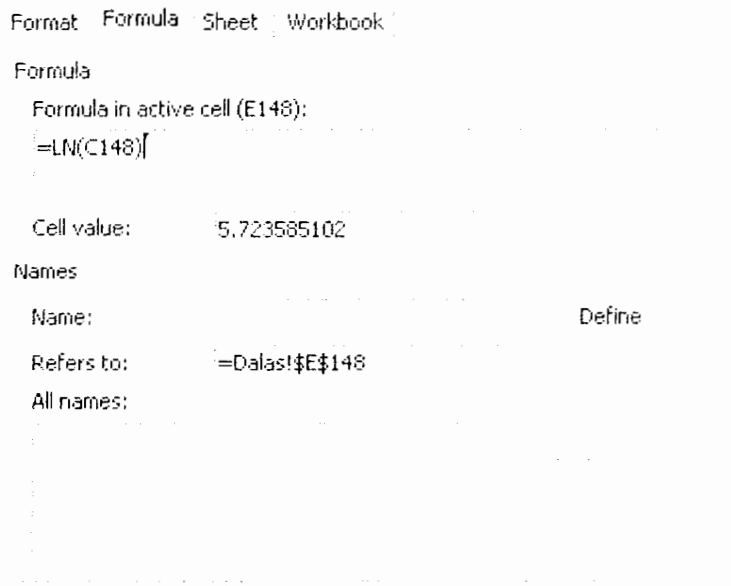
شکل ۴-۶) زبانه Format از کادر محاوره ای تنظیمات مجموعه داده ها

انتخاب کلید  در این زبانه منجر به بطلان آخرین تغییرات می گردد. مجموعه کلیدهای         به ترتیب به منظور تغییر ضخامت، حالت قلم بکار گرفته شده و همچنین موقعیت عبارت در سلول ها و رنگ نگارش مقادیر بکار گرفته می شود.

تغییر نوع و اندازه قلم بکار گرفته شده از دیگر امکانات تعبیه شده در این زبانه می باشد. فرمت مقادیر سلولها با انتخاب نوع آن در کنترل آبخاری Number format ممکن می باشد. رنگ داخل سلولها با انتخاب کلید  قابل تنظیم است. کلیدهای   به منظور ترکیب<sup>۱۳</sup> چند سلول و تفکیک<sup>۱۴</sup> سلولهای ترکیب شده بکار می رود. در انتخاب این کلیدها باید به وضعیت داده های موجود در سلولها توجه نمود. ترکیب سلولها ممکن است منجر به حذف مقادیر بعضی از آنها و تفکیک یک سلول منجر به تغییر مکان مقدار آن می گردد.

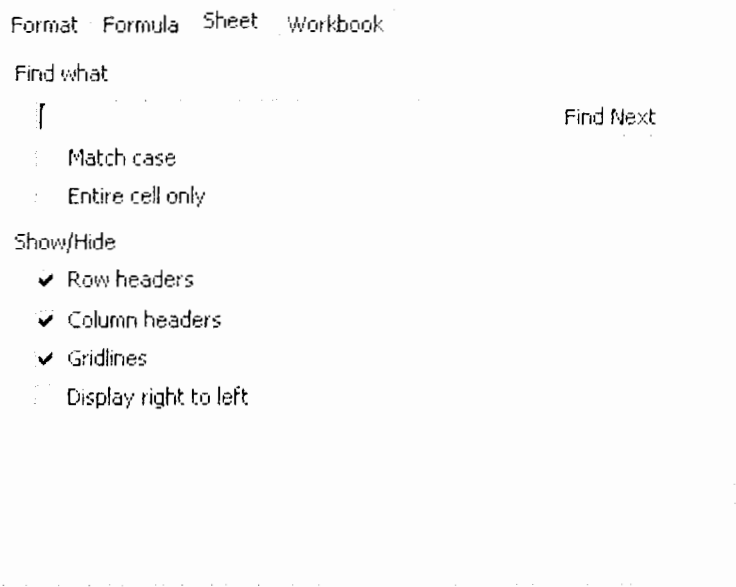
رنگ حاشیه سلولها بوسیله کلید  تعیین می گردد و منظور نمودن حاشیه برای آنها با انتخاب کلیدهای     امکان پذیر است. وضعیت نگارش مقادیر در سلولها (حالت پیش فرض، چپ به راست یا راست به چپ) در کنترل آبخاری با عنوان Text direction تعیین می گردد.

از کلید  به منظور راهنمایی در مورد امکانات کنترل نمایشگر داده ها استفاده کنید.



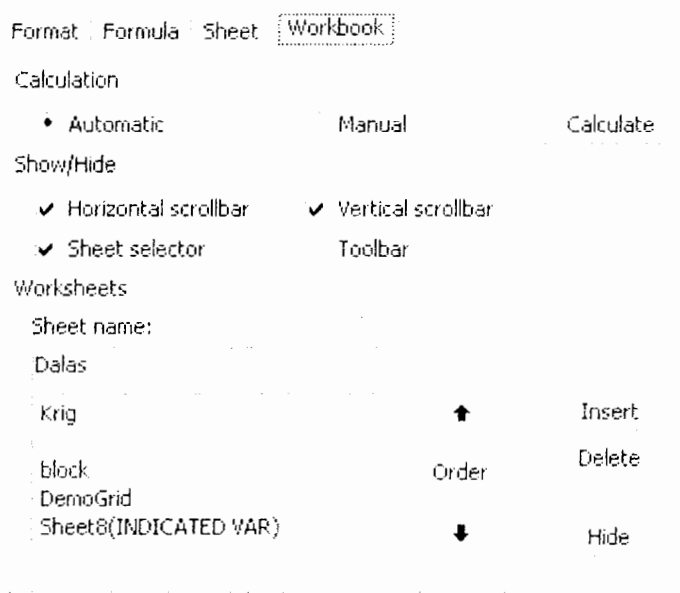
شکل ۷-۴) زبانه Formula از کادر محاوره ای تنظیمات مجموعه داده ها

زبانه Formula (شکل ۷-۴) به منظور بررسی فرمول موجود در سلول جاری بکار می رود ارائه مقدار محاسباتی در غالب فرمول از طریق تایپ مستقیم در فایل داده ها یا بطور جامع تر و کاملتر از طریق انتخاب گزینه مربوطه در منوی Format ممکن می باشد.



شکل ۸-۴) زبانه Sheet از کادر محاوره ای تنظیمات مجموعه داده ها


زبانہ Sheet (شکل ۴-۸) شامل دو قسمت اصلی است. یک قسمت عملیات جستجو در سلولها را انجام می دهد که عملکرد آن مشابه گزینه Find در منوی Edit می باشد و قسمت دیگر مربوط به نمایش فایل داده ها است. عناوین خطوط، عناوین ستونها و خطوط نمایش گر سلولها را می توان از طریق این زبانہ حذف یا پدیدار ساخت. امتداد قرارگیری سلولها (چپ به راست و راست به چپ) نیز در این زبانہ تعیین می گردد.



شکل ۴-۹) زبانہ Workbook از کادر محاوره ای تنظیمات داده ها


زبانہ Workbook (شکل ۴-۹) تنظیمات مربوط به فایل داده موجود را در بر دارد. تعیین این که آیا محاسبه عبارات فرمولی به طور خودکار انجام شود یا در اختیار کاربر باشد، از مواردی است که به عهده این زبانہ است. افزودن سری داده جدید یا حذف سری داده و جا به جایی مکانی آنها با استفاده از کلیدهای  $\uparrow$  ،  $\downarrow$  ، Insert و Delete ممکن می باشد. از طریق این زبانہ می توان مجموعه جعبه ابزاری را در بالای کادر داده ها نمایش داد. این جعبه ابزار شامل مجموعه امکاناتی بدین شرح است:


کلید  $\leftarrow$  : به منظور ابطال آخرین تغییرات بکار می رود.


کلیدهای  : به منظور مجموعه عملیات کپی و فراخوانی مجدد مقادیر کپی شده کاربرد دارند.


کلید  $\Sigma$  : که حاصل جمع مقادیر ستونی یا سطری سلولها را تولید می نماید.

کلیدهای  $\uparrow$  و  $\downarrow$  : که به منظور مرتب نمودن مقادیر ستونها به ترتیب صعودی و نزولی استفاده می شوند.

کلید  : وظیفه اعمال فیلتر به ستونهای داده ها را بعهده دارد. با استفاده از این کلید می توان مقادیر یا عبارات خاصی را از شکل داده ها حذف نمود.

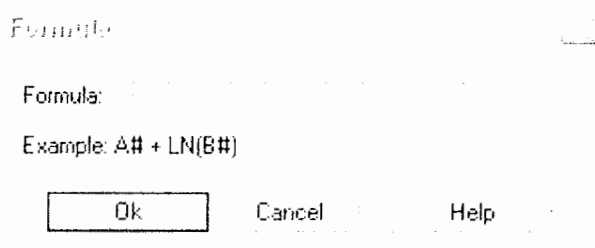
کلید  : با هدف ارسال داده ها به محیط میکروسافت اکسل تعبیه شده است.

کلید  : کادر محاوره ای تنظیمات چارت را نمایش می دهد.

کلید  : فایل راهنمای کنترل نمایشگر داده ها را اجرا می کند.

#### ۴-۳-۲- گزینه Formula :

گزینه Data Sheet در منوی Format شامل مورد دیگری با عنوان Formula می باشد. همانطور که قبلاً نیز اشاره شد این گزینه جهت فرمول دهی به سلولهای شبکه بکار می آید. با روش درج عبارات محاسباتی در سلولها آشنا هستیم. مثلاً اگر مقدار "A1+B1" در سلولی تایپ شود مقدار موجود در آن برابر با حاصل جمع مقادیر سلولهای A1 و B1 خواهد بود. حال اگر قصد داشته باشیم به ازای کل یا تعدادی از خطوط جاری داده مقدار حاصل جمع ستونهای اول و دوم را در ستون سوم قرار دهیم با انتخاب سلولهای مورد نظر در ستون سوم و انتخاب گزینه Formula (کادر محاوره ای شکل ۴-۱۰) و تایپ فرمول مذکور امکان پذیر است.



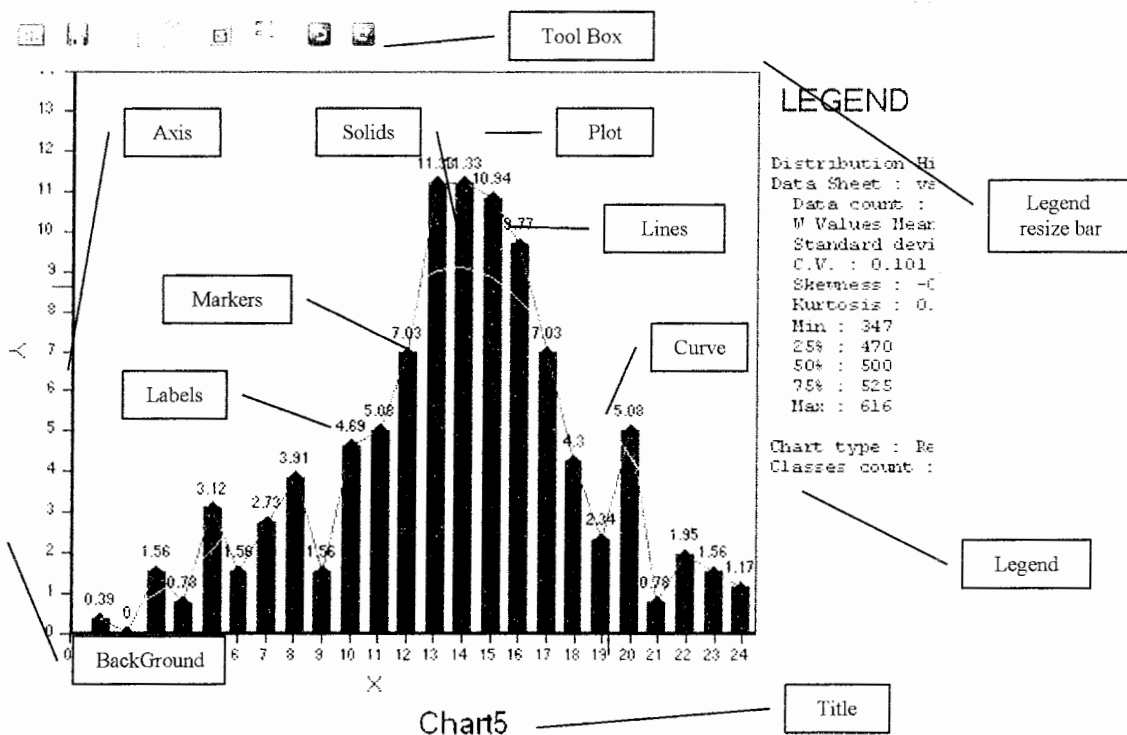
شکل ۴-۱۰) کادر محاوره ای دریافت فرمول

در کنترل متنی با عنوان Formula مقدار فرمول بمانند روش استاندارد ثبت می شود، با این تفاوت که به جای اعداد شناسه سلولها از علامت شارپ "#" استفاده می گردد. یعنی در مثال یاد شده عبارت "A#+B#" در متن مذکور تایپ می گردد. با فشردن کلید Ok مقداری فرمول برای کل سلولهای شبکه منظور خواهد شد. مثلاً در خط پنجاهم ( در صورتی که انتخاب شده باشد) مقدار "A50+B50" ثبت می گردد.

#### ۴-۳-۳- گزینه Chart :

گزینه دیگر منوی Format گزینه Chart است. در حالتی که فرم نمایشگر داده ها فعال است این گزینه عملکردی ندارد. ولی در حالت فعال بودن گرافها، با انتخاب این گزینه کادر محاوره ای مربوط به تنظیمات گرافیکی و محاسباتی گراف جاری ظاهر می گردد. این کادر قبل از ایجاد یک گراف جدید و یا با کلیک دو گانه بر روی گرافها نیز ظاهر خواهد شد. کادر مذکور شامل شش زبانه اصلی می باشد که هر کدام به نوبه خود دارای تنظیمات و امکانات متنوعی است.

هر گراف شامل چندین قسمت اصلی است، زمینه، پلات<sup>۱۵</sup>، راهنما<sup>۱۶</sup>، جعبه ابزار و میله تنظیم اندازه راهنما (شکل ۴-۱۱)



شکل ۴-۱۱) اجزای فرم نمایشگر گرافیکی

General Plot Axis Data Legend Special Setting

Title

Chart Title: Chart5

Background:  ▾

Font: [Font: Name=Microsoft Sa ...

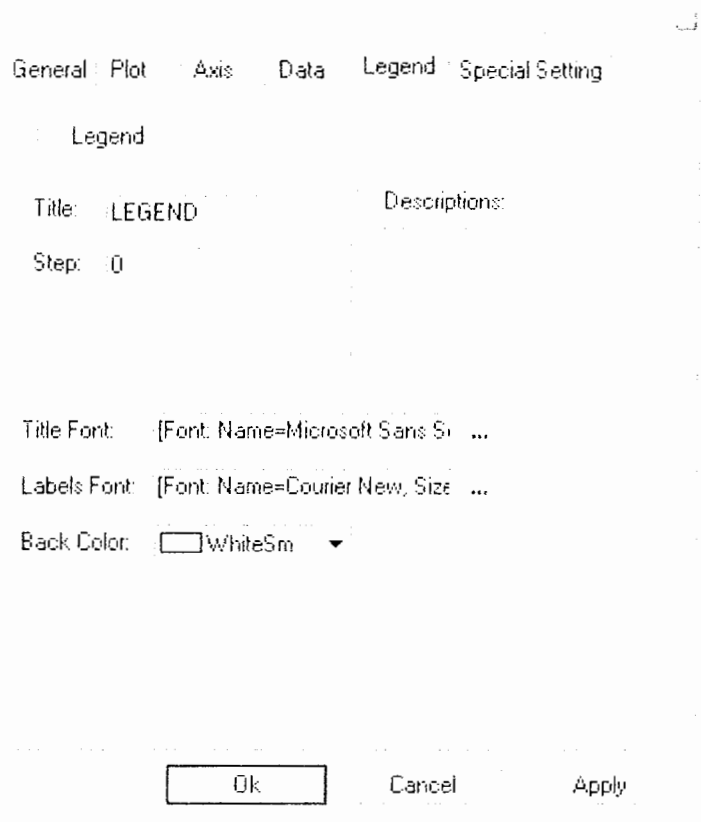
Ok

Cancel

Apply

شکل ۴-۱۲) زبانه General در کادر محاوره ای تنظیمات گراف

زبانه General (شکل ۴-۱۲) به منظور تنظیمات گرافیکی اولیه زمینه طراحی شده است. عنوان گراف و نمایش آن، قلم مورد استفاده در عنوان و رنگ زمینه جزء تنظیمات این زبانه می باشند.

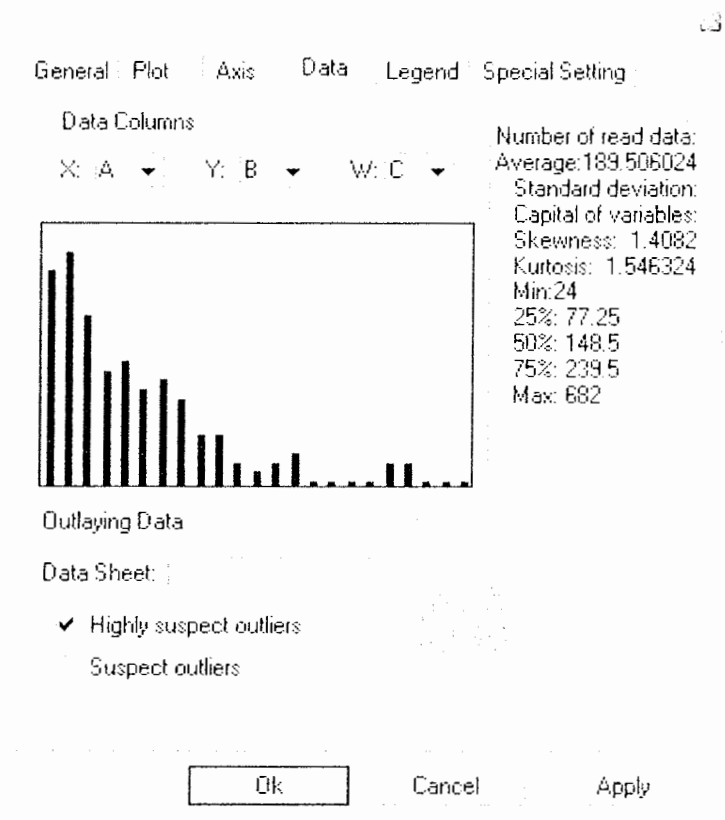


شکل ۴-۱۳) زبانه Legend در کادر محاوره ای تنظیمات گراف

زبانه Legend (شکل ۴-۱۳) تنظیمات راهنمای گرافها را عهده دار است. نمایش یا عدم نمایش راهنما از طریق انتخاب گزینه Legend در این زبانه تعیین می گردد. عنوان راهنما، تناوب مقادیر راهنما و عبارت متنی که بعنوان توضیح در راهنما ثبت می شود در این زبانه تعیین می گردد. نوع قلم عنوان راهنما و قلم توضیحات و مقادیر ثبت شده در آن با انتخاب کلید ... از مقابل عناوین هر یک از آیتم های یاد شده مشخص می گردد. رنگ زمینه راهنما از موارد دیگری است که در این زبانه می توان تعیین نمود.



در صورتی که پلات جاری حاوی شبکه مقدار دهی شده باشد، کادر راهنما دارای خاصیت ویژه ای جهت کلاسه بندی نتایج خواهد بود که در فصل بعد به این مورد خواهیم پرداخت.



شکل ۴-۱۴) زبانه Data در کادر محاوره ای تنظیمات گراف

زبانه Data (شکل ۴-۱۴) طراحی و امکانات ارائه شده در این زبانه به نحوی است که کارائی و ویژگی های فوق العاده ای را برای آن به همراه آورده است.

ایجاد تغییرات در این زبانه تنها قبل از شکل گیری گراف و زمانی که برای اولین بار گرافی ایجاد می شود ممکن می باشد. تنها در این وضعیت است که می توانیم در مورد داده های ورودی تصمیم گیری نمائیم.

نرم افزار Gridat داده ها را تنها دو بعدی و به صورت تک متغیره بررسی می نماید. لذا دو ستون به منظور مقادیر مختصات داده ها و یک ستون به منظور مقادیر متغیر در قالب کنترل‌های

آبشاری<sup>۱۷</sup> با عناوین "A", "B" و "W" از طریق این زبانه به الگوریتم کنترل کننده معرفی می گردند. مقادیر موجود در هر کدام از کنترل‌های آبشاری یاد شده مقادیر عناوین ستونهای فایل داده ها می باشد و انتخاب هر کدام از ستونهای مذکور، توزیع فراوانی مقادیر و پارامترهای آماری مربوط به آن را در دو کادر مجزا نمایش می دهد. پارامترهای آماری محاسبه شده عبارتند از تعداد داده ها و متوسط مقدار آنها، انحراف معیار، ضریب تغییرات، چولگی، کورتوزیس، مقادیر حداکثر، حداقل و چهارک های اول، دوم و سوم. این مقادیر به همراه کادر نمایش توزیع فراوانی کمک مؤثری جهت بررسی نرمالیده توزیع می نماید. در صورتی که چندین ستون داده با مقادیر خام، لگاریتمی و ... در فایل داده ها موجود باشد می توانیم با اتکا به این دو کادر، در مورد انتخاب مناسبترین توزیع تصمیم گیری نمائیم.

در کادر دیگری با نام Outlying Data که در این زبانه تعبیه شده است، عنوان سری داده های مربوط (که صفحه جاری فایل داده می باشد) و نمایش جعبه ای و نقطه ای توزیع برای ستون متغیرها (مقدار کنترل آبشاری با عنوان : W) ارائه شده است که به تصمیم گیری در مورد نرمالیده توزیع و مقادیر خارج از رده کمک می کند. از طریق گزینه های این کادر می توان داده های خارج از رده مشکوک<sup>۱۸</sup> و خیلی مشکوک<sup>۱۹</sup> را شناسایی و حذف نمود.

زبانه Axis (۴-۱۵) به منظور تنظیمات محورهای گراف طراحی شده است. محورها جزء اجزای زمینه گراف می باشند. لیکن بدلیل مفصل بودن تنظیمات آن زبانه ای جدا از زبانه General در نظر گرفته شده است. در صورتی که گرافی موجود نباشد و برای اولین بار قصد ایجاد آن را داشته باشیم گزینه های این زبانه غیر فعال است و پس از ایجاد گراف، نرم افزار با شناسایی داده ها مقادیری را بعنوان پیش فرض برای هر کدام از گزینه ها منظور می دارد که پس از ایجاد گراف می توان نسبت به تغییر آنها اقدام نمود. برای هر کدام از محورها، عنوان، مقادیر حداکثر و حداقل، تناوب، نمایش یا عدم نمایش مقادیر محورها و تناوب نمایش آنها در نظر گرفته شده است. همچنین قلم مورد استفاده برای عنوان و مقادیر محورها از طریق این زبانه تعیین می گردد.

---

17 - Combo box

18 - Suspect outliers

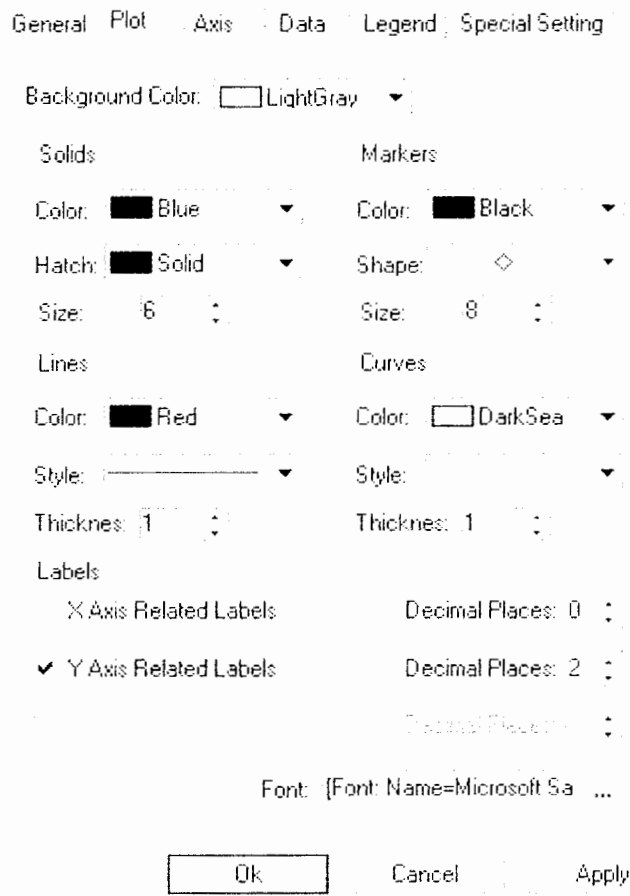
19 - Highly suspect outliers



شکل ۴-۱۵) زبانه Axis در کادر محاوره ای تنظیمات گراف

زبانه Plot با هدف تنظیمات اختصاصی پلات طراحی شده است. به شکل (۴-۱۱) مراجعه نمایید. این شکل یک هیستوگرام فراوانی نسبی را نمایش می دهد که حداکثر امکانات ممکن در آن منظور شده است. مبدل گرافیکی که قبلاً به وظیفه آن اشاره شد حاوی متغیرهای در بر گیرنده خواص زمینه، احجام<sup>۲۰</sup>، نشانگرها<sup>۲۱</sup>، خطوط<sup>۲۲</sup>، منحنی ها<sup>۲۳</sup> و شناسه ها<sup>۲۴</sup> می باشد. همچنین در مواردی که با گرافهای حاوی شبکه سروکار داریم مقدار گرافیکی مربوط به شبکه نیز به موارد فوق افزوده می شود. شکل (۴-۱۱) حاوی کلیه این متغیرها بغیر از متغیر گرافیکی شبکه می باشد. زبانه Plot (شکل ۴-۱۶) به منظور اعمال تغییر در مقادیر هر کدام از متغیرهای یاد شده بکار می رود.

- 20 - Solids
- 21 - Markers
- 22 - Lines
- 23 - Curves
- 24 - Labels



شکل ۴-۱۶) زبانه Plot در کادر محاوره ای تنظیمات گراف

کنترل آبخاری با عنوان **Back ground color**، رنگ زمینه پلات را تعیین می کند و از طریق کنترل آبخاری مشابهی در کادر **Solid** رنگ احجام و اندازه آنها (که تنها در مورد چارتهای توزیع کاربرد دارند) تنظیم می گردد. نشانگرها عواملی هستند که در هر نوع پلاتی وجود دارند و تنظیمات آنها نظیر رنگ، شکل و اندازه در کادر **Markers** ارائه شده است. به منظور حذف نمایش نشانگرها می توان مقدار اندازه آن را برابر با ۱ تنظیم نمود.

کادرهای **Lines** و **Curves** مشابه یکدیگر بوده و به منظور تنظیم رنگ، ضخامت و شکل خطوط و منحنی ها بکار می رود.

تنظیمات شناسه ها در کادر **Labels** ممکن می باشد. هر شناسه بسته به نوع گراف می تواند حاوی مقادیر مختصات نقاط داده و مقدار آنها باشد. میزان دقت شناسه (تعداد ارقام بعد

از ممیز) و قلم آن نیز در این کادر تعیین می شود. رنگ شناسه ها از رنگ نشانگرها پیروی می کنند.

زبانه Special Setting بسته به نوع گراف مورد نظر می تواند شامل گزینه ها و کنترل‌های مختلفی باشد و هدف از طراحی آن عمدتاً اعمال تغییر در روند محاسبات الگوریتم ها می باشد. عبارت دیگر پارامترهای مورد نیاز جهت اجرای الگوریتم ها در این زبانه تعیین می گردد. توضیح دقیقتر عملکرد این زبانه را به توضیحات منوی Action ماکول می کنیم.

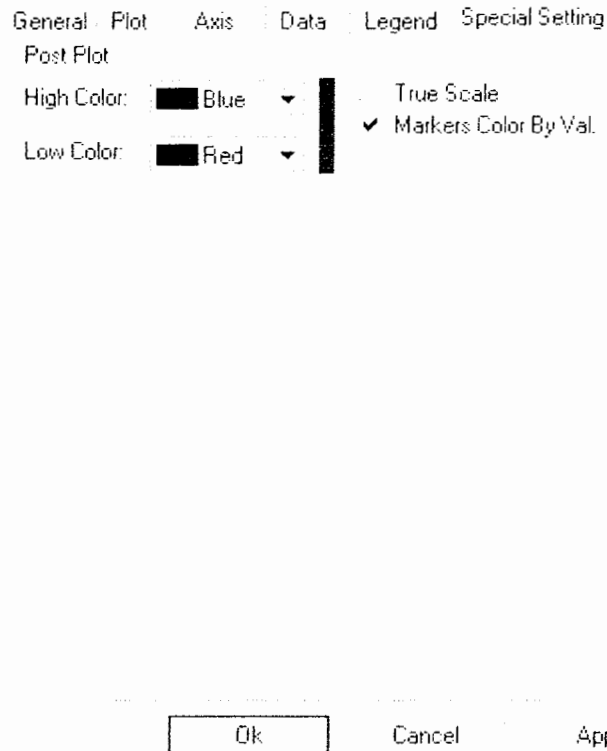
#### ۴-۴-۴-۴ منوی View

عملکرد این منو مانند عملکرد آن در نرم افزارهای استاندارد ویندوز می باشد. یعنی به منظور نمایش یا عدم نمایش اجزای کادر اصلی نرم افزار بکار می رود. نمایش یا حذف جستجوگر پروژه، میله نشانگر وضعیت، جعبه ابزار و همچنین نمایش جعبه ابزار فرم نمایشگر داده ها و گرافها (در صورت وجود) از طریق این منو ممکن می باشد.

#### ۴-۴-۵-۴ منوی Action

این منو که منوی اختصاصی نرم افزار می باشد به منظور انجام عملیات اصلی زمین آماری تعبیه شده و شامل هفت گزینه با عناوین Simulated Annealing, Sequential Simulation, Kriging, Variogram, Distribution, Post Multiple Grid است که هر کدام دستور اجرای الگوریتم های مربوط به عملیات زمین آماری را از طریق الگوریتم کنترل کننده صادر می نمایند. با انتخاب هر کدام از این گزینه ها کادر محاوره ای تنظیمات گراف ظاهر می شود که زبانه پیش فرض، آخرین زبانه یعنی Special Setting می باشد. همان طور که قبلاً اشاره شد شکل و نمایش این زبانه بسته به گزینه انتخاب شده تفاوت می کند.

عملکرد هر کدام از گزینه های این منو بدین شرح می باشد :



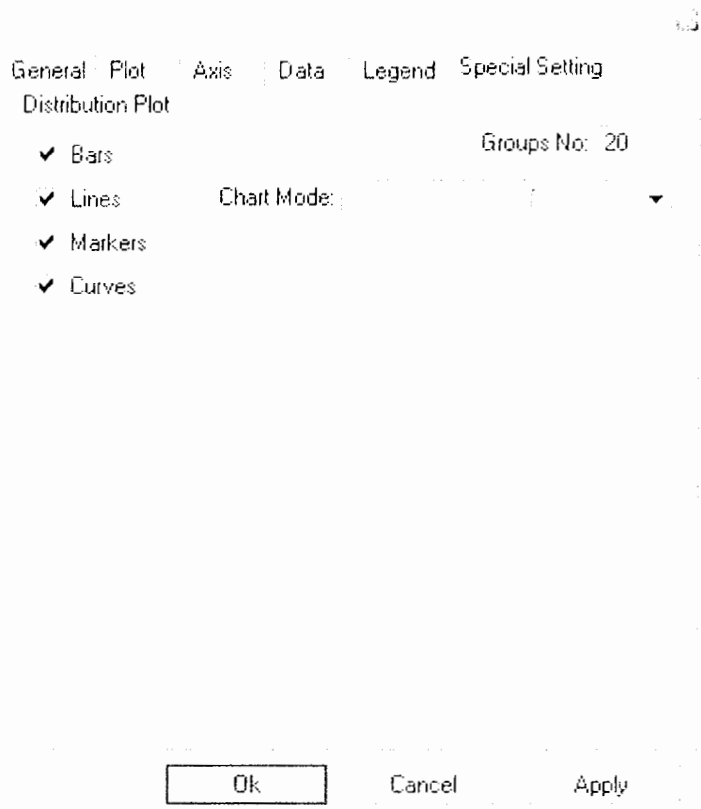
شکل ۴-۱۷) زبانه Special Setting با هدف ایجاد نقاط نمونه برداری در کادر محاوره ای تنظیمات گراف

#### ۴-۵-۱- گزینه Post :

این گزینه فراخواننده الگوریتم نمایش نقاط نمونه برداری است (فصل سوم). یعنی به منظور ایجاد نقشه نقاط نمونه برداری انتخاب می شود. تنظیمات ویژه این گزینه در شکل (۴-۱۷) مشاهده می شود از آنجایی که عملیات مرتبط با این گزینه به پارامترهای خاصی نیاز ندارد. کنترل‌های چندانی در زبانه مرتبط با آن مشاهده نمی گردد و تنها تنظیمات موجود در این رابطه شامل گزینه تعیین پیروی رنگ نشانگرها از مقدار آنها و رنگهای نشان دهنده مقادیر حداقل و حداکثر می باشد و همچنین گزینه دیگری به منظور یکسان سازی مقیاس دو محور با عنوان True Scale تعبیه شده است.

#### ۴-۴-۵-۲- گزینه Distribution :

هدف از انتخاب این گزینه ترسیم انواع هیستوگرام و نمودار احتمال است. این گزینه در رابطه با الگوریتم بررسی آماری توزیع داده ها می باشد. در صورت انتخاب این گزینه کادر محاوره ای تنظیمات گراف با طراحی زبانه Special Setting مطابق شکل (۴-۱۸) ظاهر می شود.



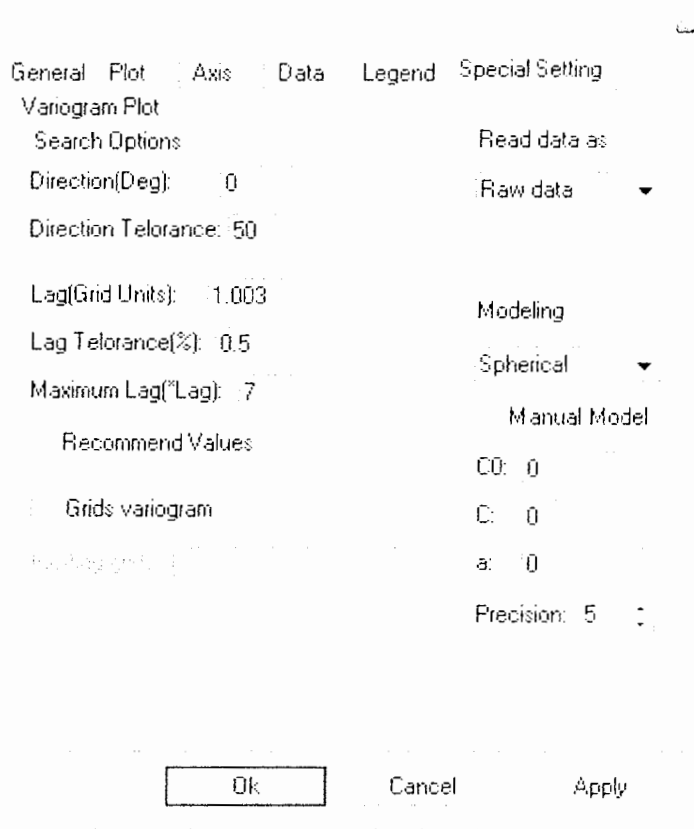
شکل (۴-۱۸) زبانه Special Setting با هدف ایجاد چارت توزیع مقادیر در کادر محاوره ای تنظیمات گراف

در این زبانه چهار گزینه به منظور تعیین نمایش میل های هیستوگرام، خطوط متصل کننده آنها به یکدیگر و نشانگرهای هر کدام در نظر گرفته شده است. نوع چارت را می توان از کنترل آبخاری Chart Mode انتخاب نمود. این کنترل طبق الگوریتم بررسی آماری توزیع داده ها شامل شش عضو است که به ترتیب عبارتند از هیستوگرام فراوانی، هیستوگرام فراوان تجمعی، هیستوگرام فراوانی نسبی، هیستوگرام فراوانی نسبی تجمعی، نمودار احتمال و نمودار احتمال لگاریتمی. در صورتی که انواع نسبی یا نسبی تجمعی انتخاب شود می توانیم با انتخاب گزینه

Curves، مدل توزیع نرمال مطابق با میانگین و واریانس توزیع را در چارت مشاهده کنیم. انتخاب گزینه مذکور در حالات دیگر هیچ تأثیری در نمایش چارت خروجی نخواهد داشت. علاوه بر موارد فوق تعداد کلاسها (C) که در ابتدا به طور پیش فرض بر اساس تعداد داده ها (N) و با استفاده از فرمول  $C = 10/\ln(N)$  محاسبه شده است را می توان در این کادر تغییر داد.

#### ۴-۴-۵-۳- گزینه Variogram :

این گزینه به منظور واریو گرافی و مدل سازی وریوگرام به کار می رود و ارتباط آن با الگوریتم واریوگرافی و مدل سازی وایوگرام می باشد. طراحی زبانه Setting Special در صورت انتخاب این گزینه بمانند شکل (۴-۱۹) است.



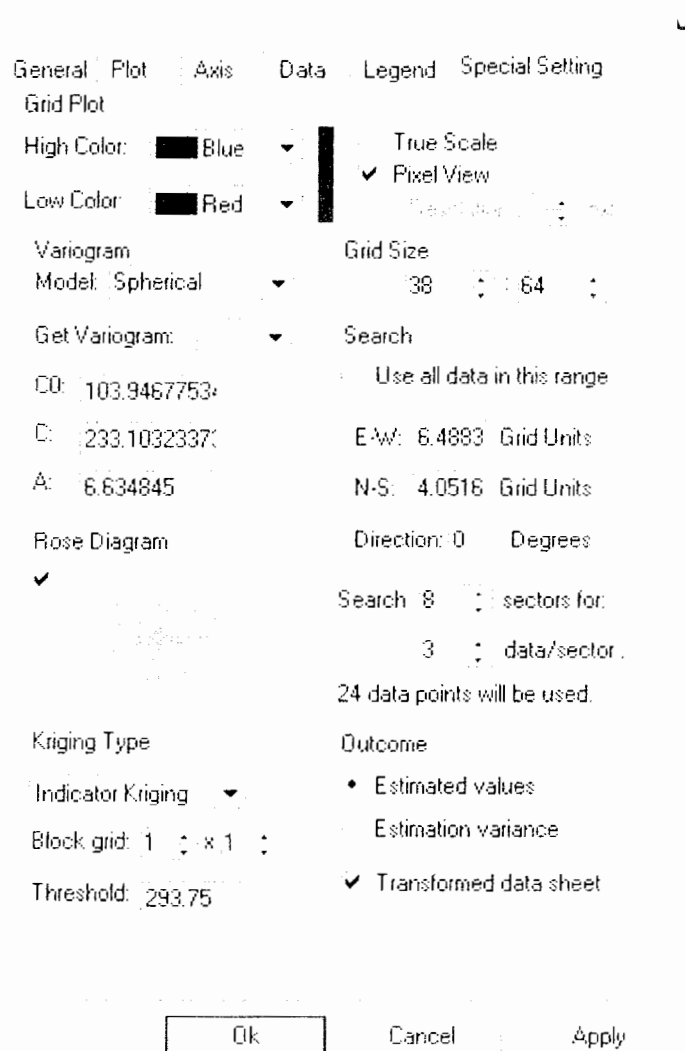
شکل ۴-۱۹) زبانه Special Setting با هدف واریوگرافی و مدل سازی وریوگرام در کادر محاوره ای تنظیمات



همان طور که مشاهده می شود این زبانه شامل سه کادر اصلی تنظیمات جستجو، نحوه خواندن داده ها و تنظیمات مدل می باشد. در کادر تنظیمات جستجو، امتداد و حدود تغییرات مجاز آن، فواصل گام و حدود مجاز تغییرات آن و حداکثر تعداد گام مشخص می گردد. اگر این کادر اولین بار و برای ایجاد یک گراف جدید ظاهر شده باشد، مقادیر پارامترهای جستجو توسط نرم افزار محاسبه و به عنوان مقادیر پیش فرض منظور می گردد. در صورتی که پس از ایجاد تغییرات قصد ثبت مجدد مقادیر پیش فرض را داشته باشیم با فشردن کلید (Recommand Values) این مقادیر به محل های مربوط باز می گردند. کادر کوچکی که در حاشیه راست بالای زبانه با عنوان Read data as مشاهده می شود به منظور فراهم آوردن امکان واریوگرافی با استفاده از مقادیر لگاریتمی یا شاخص می باشد. با انتخاب هر کدام از اعضای کنترل آبشاری موجود در این کادر می توانیم بدون ایجاد تغییر در فایل داده ها عملیات واریوگرافی لگاریتمی و شاخص علاوه بر واریوگرافی معمولی انجام دهیم. این دو نوع وریوگرام در تخمین و شبیه سازی گوسی و شاخص کاربرد خواهند داشت. وظیفه کادر Modeling نیز مشخص است. هرچند تا زمانی که کاربر گزینه Manual Model را انتخاب نکند نیازی به مقدار دهی خاصی در این کادر نمی باشد و تنها مدل وریوگرام مشخص می شود. زیرا پس از فشردن کلید Ok مقادیر پارامترهای مدل محاسبه شده و مدل در چارت به نمایش در می آید. از این پس اگر کادر محاوره ای تنظیمات چارت احضار شود مقادیر پارامترهای مدل در محل های مربوطه ثبت شده اند. توصیه می شود در صورتی که مدل محاسبه شده توسط نرم افزار رضایت بخش نبود (که ندرتاً اتفاق می افتد) با انتخاب گزینه Manual Model به پارامترها مقدار دهی شود و مدل مناسب به نتایج واریوگرافی منطبق گردد. در این کادر نوع مدل استاندارد و پارامترهای آن یعنی  $C_0$  و  $a$  و دقت محاسبات تعیین می گردد. در صورتی که نشانگر ماوس را بر روی کنترل آبشاری مدل حرکت دهید تصویر شماتیکی از مدل انتخاب شده نمایش داده می شود که به انتخاب مدل مناسب کمک می کند.

#### ۴-۴-۵-۴- گزینه Kriging :

این گزینه در رابطه با الگوریتم تخمین بروش کریجینگ می باشد و عملیات شبکه سازی به روش کریجینگ، لاگ کریجینگ و کریجینگ شاخص با انتخاب این گزینه انجام می گیرد. کادر محاوره ای تنظیمات شامل کنترل ها و گزینه های نشان داده شده در شکل (۴-۲۰) می باشد.



شکل ۴-۲۰) زبانه Special Setting با هدف تخمین بروش کریجینگ در کادر محاوره ای تنظیمات گراف

این کادر محاوره ای قبل از اینکه نمایش داده شود به محاسبه مقادیر آنیزوتروپی محیط می پردازد لذا در این زبانه کادری با عنوان Rose Diagram مشاهده می شود که بر اساس آن می توان در مورد مواردی چند از جمله بیضوی جستجو تصمیم گیری نمود. ابعاد شبکه نیز بر

اساس این نمودار محاسبه و منظور شده اند، لیکن ایجاد تغییر در آنها تأثیر خاصی در نتایج محاسبات نخواهد داشت.

زبان حاضر برای کلیه گزینه هایی که در نهایت منجر به ساخت شبکه می شود، دارای کنترل‌های مشترکی جهت تنظیمات مربوط به رنگهای متناسب با مقادیر حداکثر و حداقل، گزینه تعیین مقدار یکسان در محورها، کنترل‌های تعیین کننده ابعاد شبکه و گزینه وضعیت نمایش شبکه می باشد. مورد اخیر تعیین می کند که آیا مقادیر محاسبه شده شبکه به طور مستقیم رنگ آمیزی و در محل نقاط شبکه نمایش داده شود یا اینکه این مقدار به مرکز شبکه اختصاص داده شده و رنگ آمیزی با هموار سازی مقادیر در نقاط دیگر انجام گیرد. انتخاب گزینه مذکور تصویر زیباتری از محاسبات ارائه می دهد، لیکن توصیه نمی شود. در صورت انتخاب این گزینه می توان شدت هموار سازی را بر حسب واحد نمایش گرافیکی صفحه<sup>۲۵</sup> مشخص نمود.

کادریهای دیگر موجود در این زبان عبارتند از کادر وریوگرام، کادر جستجو، کادر نوع کریجینگ و کادر شبکه خروجی.

کادر وریوگرام پارامترهای مدل وریوگرام مورد استفاده در عملیات تخمین را دریافت می دارد این مقادیر به طور پیش فرض بر اساس مدل کروی محاسبه شده اند. لیکن اگر وریوگرام خاصی در پروژه موجود است که قصد اختصاص آن را به عملیات داریم، با انتخاب عنوان چارت مذکور از کنترل آبشاری Get Variogram، مقادیر پارامترهای وریوگرام مذکور در مکانهای مربوطه ثبت می گردد.

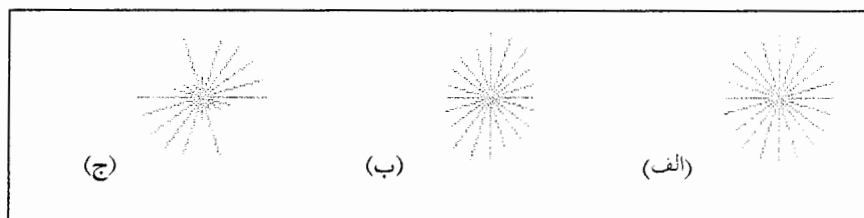
در کادر جستجو (با عنوان Search) محدوده بیضوی جستجو با توجه به رز دیاگرام (شکل ۴-۲۱-الف) تعیین می گردد. توجه فرمایید که بیضوی منطبق شده به مقادیر دیاگرام ممکن است از دقت کافی برخوردار نباشد که توصیه های چندی به شرح زیر در اصلاح بیضوی جستجو کارآمد خواهد بود :

✓ اگر مقادیر رز دیاگرام دارای پیوستگی مناسب بوده و آرایش خطوط رسم شده در آن بنحوی باشد که انطباق بیضی به آنها با خطای قابل قبول ممکن باشد ولی بیضوی ترسیم شده توسط نرم افزار با این مقادیر انطباق مناسبی نداشته باشد، با تغییر در مقادیر

Direction و N-S Grid Units , E-W Grid Units بیضوی مناسبی به نمودار منطبق

نمایید (شکل ۴-۲۱-ب).

✓ اگر مقادیر رز دیاگرام دارای پیوستگی لازم نباشد توصیه می شود محیط جستجو را دایره ای با شعاع برابر با بزرگترین مقدار محاسبه شده منظور نمایید (شکل ۴-۲۱-ج).



شکل ۴-۲۱) وضعیت های ممکن نمایش رز دیاگرام

اگر در کادر جستجو گزینه Use all data in this range انتخاب شود، به ازای هر نقطه شبکه کلیه نقاط معلوم در بیضوی جستجو در محاسبات منظور می گردند (در صورتی که هیچ داده ای در شعاع جستجوی نقطه جاری موجود نباشد مقداری به آن اختصاص نخواهد یافت)، در غیر این صورت جستجو بروش قطاعی انجام خواهد شد. در این روش بیضوی جستجو به تعداد مشخصی قطاع (پیش فرض ۸ قطعه) تقسیم میشود و از هر قطاع به تعداد مشخص شده (پیش فرض سه عدد) داده انتخاب می گردد، سپس تخمین با استفاده از این نقاط انجام می گیرد. این روش در حذف عدم یکنواختی توزیع نقاط نمونه برداری مفید می باشد لیکن استفاده از آن در مواردی که با کمبود داده مواجه ایم توصیه نمی گردد.

نوع کریجینگ مورد نظر از کنترل آبخاری Kriging Type انتخاب می گردد. در صورتی که عملیات لاگ کریجینگ مورد نظر باشد گزینه ای به منظور تصمیم گیری در مورد تبدیل داده ها به حالت اول<sup>۲۶</sup> ظاهر می شود. گزینه دیگر با عنوان Standardize الگوریتم را وادار به تبدیل توزیع به حالت نرمال استاندارد (در صورتی که توزیع لاگ نرمال باشد) می نماید.

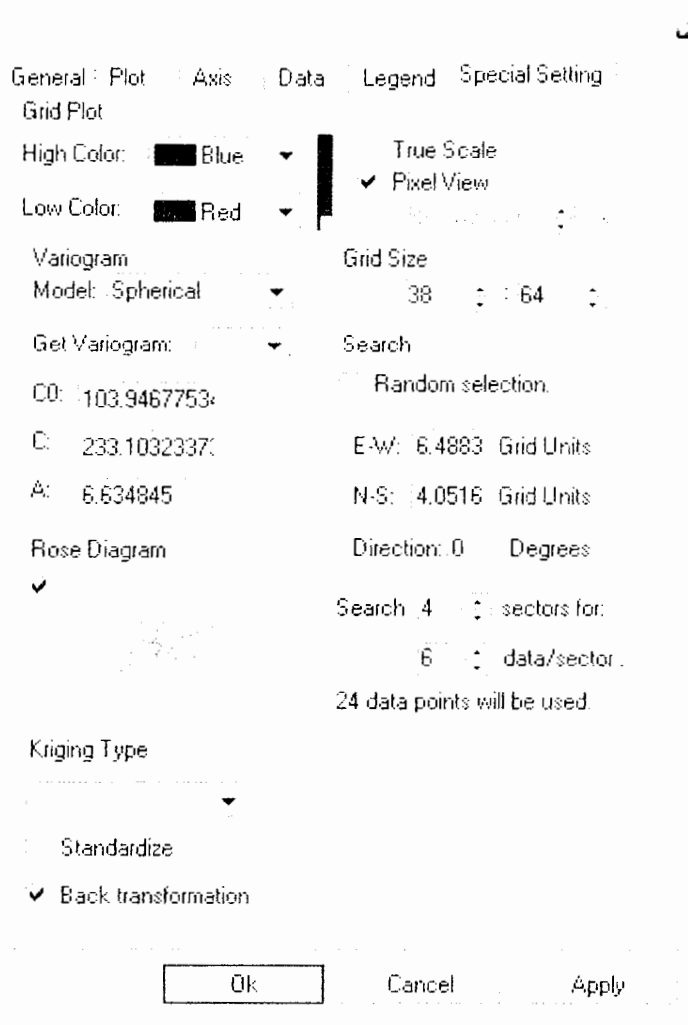
در صورتی که حالت کریجینگ شاخص مورد نظر باشد کنترل متنی که مقدار پیش فرض آن معادل چهارک سوم توزیع است بعنوان حد آستانه ای پدیدار خواهد گشت. در کادر خروجی

عملیات می توان با انتخاب گزینه Transformed data sheet سری جدیدی از داده ها ایجاد نمود که حاوی مقادیر لگاریتمی شده یا شاخص مورد استفاده در تخمین باشد.

ابعاد منظور شده بعنوان Block grid بیانگر تعداد نقاط در نظر گرفته شده در هر بلوک یا میزان تفکیک بلوک ها در عملیات کریجینگ می باشند. این مقادیر بطور پیش فرض برابر با ۱ منظور می گردند. یعنی عملیات پیش فرض نرم افزار، کریجینگ نقطه ای می باشد

#### ۴-۴-۵- گزینه Sequential Simulation :

همان طور که از عنوان این گزینه بر می آید، هدف آن اجرای الگوریتم شبیه سازی مرحله ای است. ظاهر زبانه Special Setting مرتبط با این گزینه مطابق شکل (۴-۲۲) می باشد.

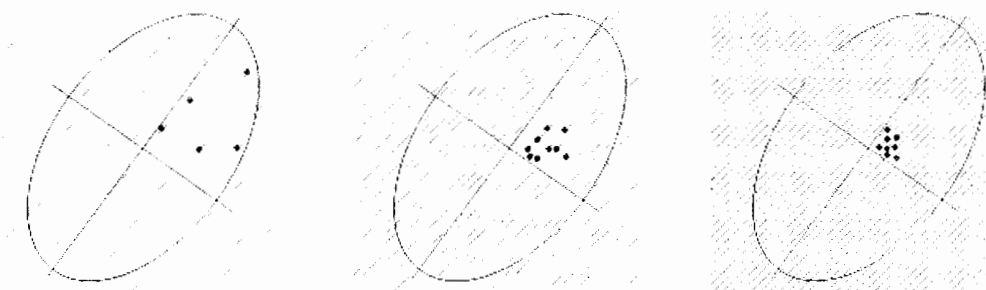


شکل ۴-۲۲) زبانه Special Setting با هدف شبیه سازی مرحله ای در کادر محاوره ای تنظیمات گراف

همان طور که ملاحظه می کنید طراحی آن مشابه طراحی کادر محاوره ای کریجینگ بوده و عملکرد کنترلرها و گزینه ها نیز کاملاً مشابه است و علت نیز شباهت عملیات تخمین بروش کریجینگ و شبیه سازی مرحله ایست. وضعیت نمایش شبکه، تنظیمات و ریوگرام، پارامترهای جستجو و نوع کریجینگ مورد استفاده در عملیات، از تنظیمات موجود در این زبانه می باشند.

عملکرد گزینه Random selection که در بالای کادر جستجو مشاهده می شود نیاز به توضیح بیشتری دارد. همانطور که گفته شد در انتخاب نقاط داده معلوم به روش جستجوی قطعی ابتدا بیضوی جستجو به تعداد مشخصی قطاع تقسیم می شود که از هر قطاع به تعداد مشخصی داده انتخاب خواهد شد. در توضیحات الگوریتم شبیه سازی مرحله ای به این مطلب اشاره کردیم که مقادیر داده ها مطابق با مقادیر حائز مقدار در شبکه می باشند. اولین نقطه شبکه (که به طور تصادفی انتخاب شده است) را در نظر بگیرد. مسلماً تعدادی نقاط مقدار دار که منطبق بر نقاط نمونه برداری می باشند در بیضوی جستجو قرار گرفته اند. اگر بیضوی به چهار قطاع تقسیم شده باشد (شکل ۴-۲۳-الف) از هر قطاع به تعداد مورد نیاز (مثلاً شش عدد) داده انتخاب می شود.

حال وضعیت الگوریتم را در میانه عملیات تصور کنید (شکل ۴-۲۳-ب). مسلماً تعداد نقاط حائز مقدار در بیضوی جستجو افزایش یافته است. با توجه به اینکه الگوریتم جستجو را از نزدیکترین مقادیر به نقطه مجهول آغاز می کند و نیز با توجه به افزایش نقاط مقدار دار، تعداد شش داده در هر قطاع سریع تر یافت می شود و این بمعنی کاهش شعاع جستجو میباشد.



• نقاط دارای مقدار کم در شبیه سازی بتار من روبرو

نقاط فاقد مقدار

نقاط دارای مقدار

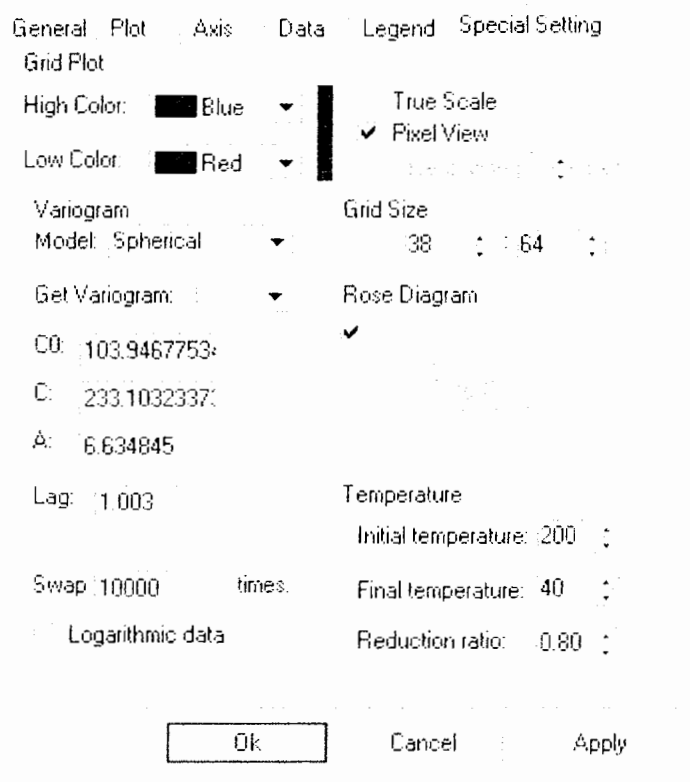
شکل ۴-۲۳) کاهش شعاع جستجو با پیشرفت عملیات شبیه سازی مرحله ای

این عملیات را برای آخرین نقطه شبکه تصور نمایید. در این حالت شبیه سازی تنها با اتکا به نقاط مجاور نقطه مجهول انجام می گیرد، زیرا کلیه آنها مقدار دهی شده اند (شکل ۴-۲۳-ج) بنابراین بر اساس الگوریتم استاندارد شبیه سازی مرحله ای، شعاع تخمین با پیشرفت عملیات کاهش می یابد منظور نمودن گزینه Random selection الگوریتم جستجو را مکلف می نماید که به جای آغاز عملیات جستجو از نزدیکترین نقاط به نقطه مجهول، عملیات را به طور تصادفی و در کل سطح قطاع انجام دهد. لذا شعاع تأثیر همواره برابر با مقدار اولیه منظور می گردد.

#### ۴-۴-۵-۶- گزینه Simulated annealing :

احضار الگوریتم شبیه سازی انیلینگ از طریق این گزینه انجام می شود تنظیمات نمایش شبکه و تنظیمات مدل و ریوگرام ورودی، مشابه موارد پیشین می باشد. و ریوگرام وارد شده، تعیین کننده تابع هدف الگوریتم خواهد بود. تعداد دفعات جابجایی در شبکه در کنترل متنی Swap (شکل ۴-۲۴) تعیین می گردد.

ن

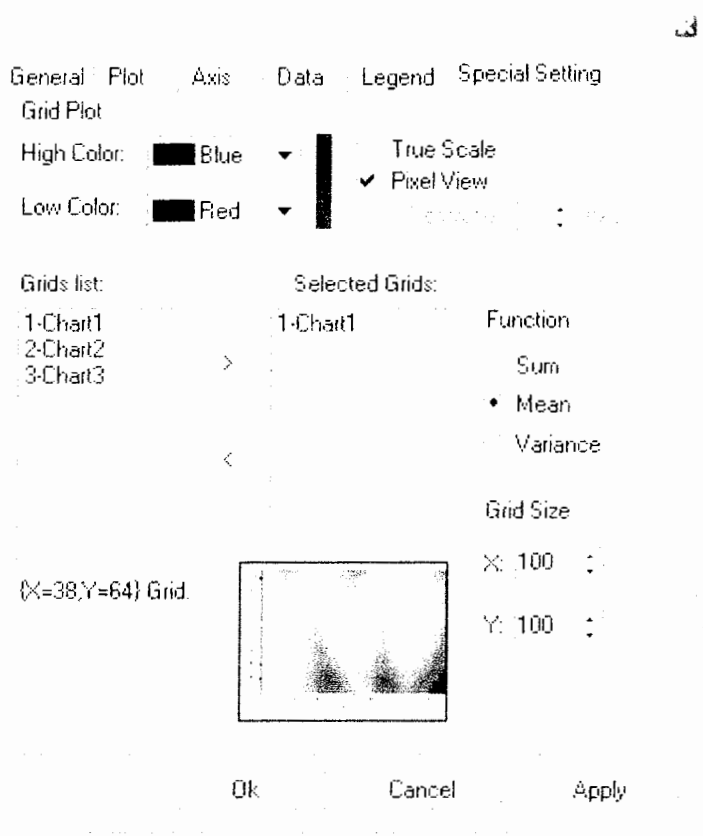


شکل ۴-۲۴) زبانه Special Setting با هدف اجرای شبیه سازی انیلینگ در کادر محاوره ای تنظیمات گراف

گزینه Logarithmic data مقادیر داده ها را به صورت لگاریتمی به الگوریتم ارسال کرده که در پایان مقادیر به حالت اول بازگردانده می شوند. انتخاب این گزینه در مواقعی که خروجی عملیات شبکه مطلوبی نمی باشد (مواردی که توزیع چولگی بالایی دارد) توصیه می شود. میزان تاثیر اجزای تابع هدف (رابطه ۳-۱۲) در کادر Objective function تعیین می گردد.

#### گزینه ۴-۵-۷- Multiple Grid :

این گزینه به منظور ترکیب چندین شبکه بکار می رود. طراحی زبانه Special Setting در رابطه با گزینه مذکور در شکل (۴-۲۵) مشاهده می شود. این زبانه هم اکنون شامل لیستی با عنوان Grids List است که گرافهایی با امکان ترکیب شدن (گرافهای کریجینگ، شبیه سازی و گرافهای مرکب) در آن ثبت شده اند. با انتخاب عنوان هر کدام از آنها، ابعاد شبکه در زیر لیست ثبت شده و شمای کلی آن نمایش داده می شود.



شکل (۴-۲۵) زبانه Special Setting با هدف ترکیب شبکه ها در کادر محاوره ای تنظیمات گراف



با انتخاب کلیدهای < > می توان شبکه های مورد نظر را جهت ترکیب با یکدیگر انتخاب نمود و به لیست Selected Grids اضافه نمود. در کادر مقابل این لیست نوع عملیات ترکیب مشخص می شود که مورد پیش فرض، عملیات محاسبه میانگین مقادیر شبکه ها می باشد. همچنین محاسبه جمع مقادیر و واریانس مقادیر شبکه ها نیز ممکن خواهد بود. شبکه خروجی به طور پیش فرض با ابعاد صد در صد ایجاد خواهد شد. برای نقاطی که معادل مختصاتی آنها در یکی از شبکه های اولیه فاقد مقدار باشد، هیچ مقداری محاسبه نمی شود.

#### ۴-۴-۶- منوی Window

این منو از منوهای استاندارد می باشد و کاربرد آن هم مرتب نمودن فرم های وابسته به فرم اصلی است. گزینه های این منو فرم های موجود در پروژه اعم از داده و گراف را با نظام های معمول ویندوز مرتب می نماید.

#### ۴-۴-۷- منوی Help

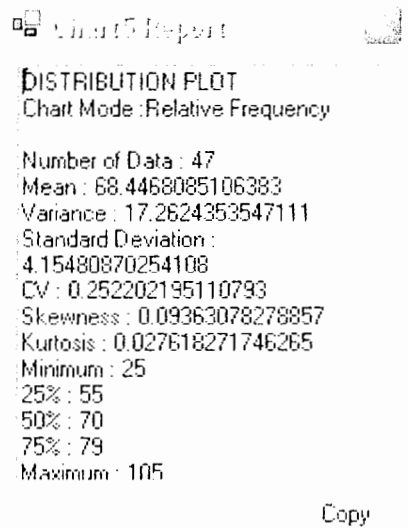
این منو شامل سه گزینه است. گزینه Help فایل راهنمای نرم افزار را فراخوانی میکند. گزینه Office Help فایل راهنمای کنترل نمایشگر داده ها را احضار می نماید و کاربرد گزینه About نیازمند توضیح نمی باشد.

#### ۴-۵- جعبه ابزار Gridat

جعبه ابزار شامل کلیدهای میانبر گزینه های منوها می باشد. موقعیت جعبه ابزار نرم افزار Gridat را در شکل (۴-۳) مشاهده می کنید. حذف یا نمایش آن در صفحه با استفاده از منوی View و انتخاب گزینه Tool Box ممکن می باشد. کلیدهای موجود در این جعبه ابزار به ترتیب میانبرهای گزینه های New, Open, Save, Save as, Print Previw, Print و از منوهای فایل و میانبر گزینه های Redo, Undo, Delete و Find از منوی Edit و میانبر گزینه Help از منوی Help می باشد.

#### ۴-۶- جستجو گر پروژه

این کادر با هدف دسترسی سریع و سازماندهی شده به اعضای پروژه طراحی شده است. نمایش این کادر از طریق منوی View و انتخاب گزینه Project Explorer ممکن می باشد. با کلیک دو گانه در حاشیه تیره این کادر نیز می توان آن را مخفی نمود. در صورتی که بر روی هر یک از اعضای این لیست راست کلیک نمایید، منوی شناور حاوی سه گزینه Remove, Report و Setting ظاهر می گردد که در صورت انتخاب گزینه Setting کادر محاوره ای تنظیمات آیتم انتخاب شده (فایل داده ها یا گراف) ظاهر می گردد. انتخاب گزینه Remove برای گراف، موجب حذف آن خواهد شد و انتخاب گزینه Report باعث نمایش کادر محاوره ای شامل توضیحاتی در مورد گراف جاری و عملیات انجام شده جهت ایجاد آن می گردد (شکل ۴-۲۶). نمایش مجدد گرافی که با انتخاب کلید  بسته شده است با کلیک نمودن بر روی عنوان آن در این کادر ممکن می باشد. دو گزینه Report و Remove برای فایل داده ها تعریف نشده است.



شکل ۴-۲۶) کادر محاوره ای Report

#### ۴-۷- میله نشانگر وضعیت

این میله در پایین اکثر پنجره های ویندوز مشاهده می شود که وظیفه آن آگاه نمودن کاربر از روند اجرای پروژه می باشد. قسمت اول این میله محل نمایش پنجره جاری نرم افزار (اسامی گراف ها یا فرم نمایشگر داده ها) و اعلام عملیات جاری در حین اجرای الگوریتم ها می باشد. قسمت دوم میله نشانگر وضعیت در زمان اجرای الگوریتم های طولانی (نظیر شبیه سازی ها) فعال می شود و درصد انجام عملیات را به صورت گرافیکی و عددی نمایش خواهد داد. در پایان نیز مدت زمان صرف شده جهت اجرای الگوریتم در قسمت سوم میله مذکور ثبت خواهد شد و تا زمانی که نشانگر ماوس متوجه گراف خاصی نشده باقی خواهد ماند. حرکت ماوس به روی گرافها باعث نمایش مکان جاری نشانگر با توجه به مختصات گراف جاری در این قسمت می گردد.

#### ۴-۸- زمینه

این قسمت فاقد کارآیی خاصی بوده و صرفاً محل نگهداری فرم های وابسته به فرم اصلی است.

## فصل پنجم

### بررسی عملکرد نرم افزار Gridat

## فصل پنجم

### بررسی عملکرد نرم افزار Gridat

#### ۵-۱- روش بررسی عملکرد Gridat

صرف زمان و هزینه جهت اجرای هر پروژه مادامی ارزشمند خواهد بود که در پایان نتایج مطلوب و مورد نظر، که بعنوان اهداف پروژه تعریف شده اند، عاید گردد. به همین ترتیب نرم افزاری که خروجی های مورد نظر طراح را تولید نکند و قادر به فراهم آوردن نیازهای کاربران نباشد مجموعه موفق محسوب نخواهد شد، هر چند که اصول استاندارد طراحی محیط نرم افزار دقیقاً رعایت شده باشد.

امروزه طراحی نرم افزار توجه به دو جنبه مجزا را می طلبد. یک طرف که در همه نرم افزارهای تحت سیستم عامل های مرسوم امروزی مشترک می باشد عبارت است از رابط گرافیکی کاربر<sup>۱</sup> که شامل مجموعه منوها، جعبه ابزار، فرمها و ... می باشد و وظیفه آن ایجاد ارتباط بین کاربر و مجموعه است. ایجاد، ذخیره سازی و فراخوانی پروژه و چاپ نتایج و از این قبیل عملیات در طراحی این قسمت مورد توجه قرار می گیرد. خوشبختانه امروزه استانداردهای مشخصی جهت رابط گرافیکی کاربر تعریف شده و سازندگان نرم افزار حتی المقدور خود را ملزم به رعایت آن اصول می دانند. این رویه باعث می گردد کاربر در نگاه اول به نرم افزار مسیر کلی برنامه را بیابد و ساختار کلی اجرای پروژه را در ذهن خود تداعی کند که این امر از سرگردانی کاربران در محیط نرم افزار ممانعت بعمل می آورد.

---

1 - Graphical User Interface (GUI)

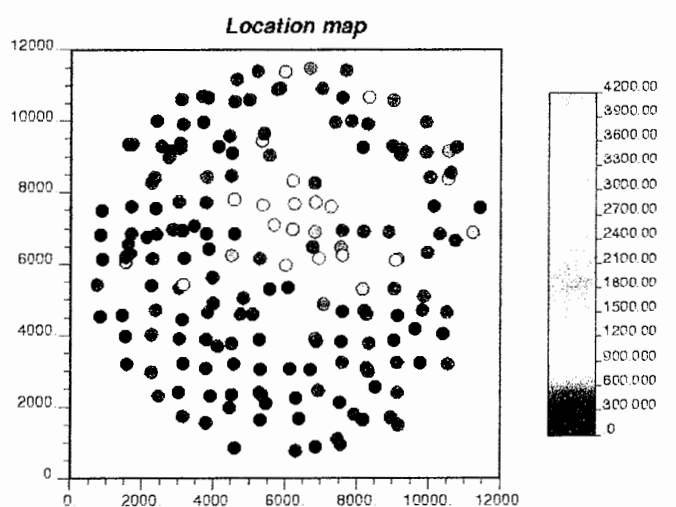
- واریوگرافی و مدل سازی وریوگرام

- ایجاد شبکه توزیع منطقه ای مقادیر بروش شبیه سازی یا کریجینگ

طی بخشهای حاضر ابتدا نتایج حاصل از اجرای یک پروژه زمین آماری در WinGSLib ارائه شده و سپس در قالب بررسی روش اجرای پروژه توسط Gridat، سعی شده روند تنظیم و اجرای نرم افزار مورد توجه قرار گیرد و نتایج با خروجی های WinGSLib مقایسه گردد. در نگارش این فصل فرض بر این گذارده شده که خوانندگان محترم با روش استفاده و اجرای WinGSLib آشنایی کامل دارند.

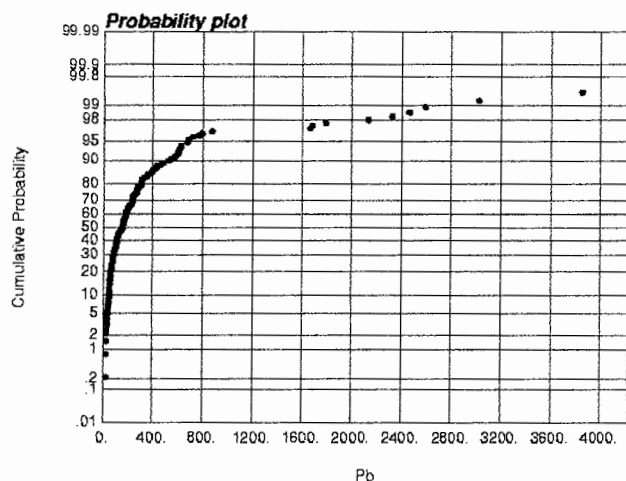
### ۵-۲- خروجی های WinGSLib

فایل داده ای با نام Dallas.dat شامل مقادیر و مختصات داده های اولیه می باشد. این فایل در فرمت Geo-EAS تهیه شده و شامل سه ستون اطلاعات است که به ترتیب بیانگر طول و عرض محلی نقاط برداشت و مقادیر حاصل از برداشت می باشد. پس از ایجاد پروژه و اعمال تنظیمات پیش فرض می توان تصویر پراکندگی و موقعیت هر کدام از نقاط را ایجاد نمود که در این خصوص خروجی WinGSLib مطابق شکل (۵-۱) می باشد.



شکل ۵-۱) نمایش پراکندگی نقاط برداشت داده های اولیه (خروجی WinGSLib).

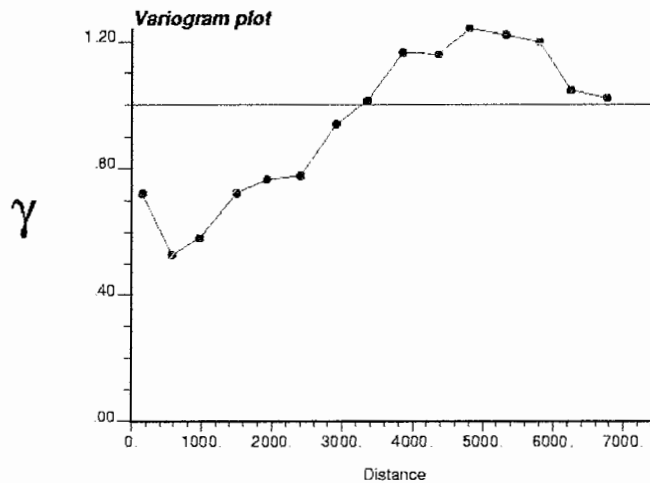
جهت بررسی نرمالیت توزیع داده ها نمودار احتمال ایجاد می گردد. نمودار احتمال ایجاد شده توسط WinGSlib مطابق شکل (۲-۵) خواهد بود.



شکل (۲-۵) بررسی توزیع مقادیر فایل Dallas.dat با استفاده از نمودار احتمال (خروجی WinGSlib).

بمنظور نرمال نمودن توزیع مقادیر، WinGSlib ابزاری ارائه داده است که خروجی آن یک فایل داده است. این فایل حاوی یک ستون شامل مقادیر نرمال شده داده ها علاوه بر مقادیر مندرج در فایل داده های اولیه می باشد. همچنین فایل دیگری جهت بازگرداندن توزیع مقادیر به حالت اولیه ایجاد می گردد.

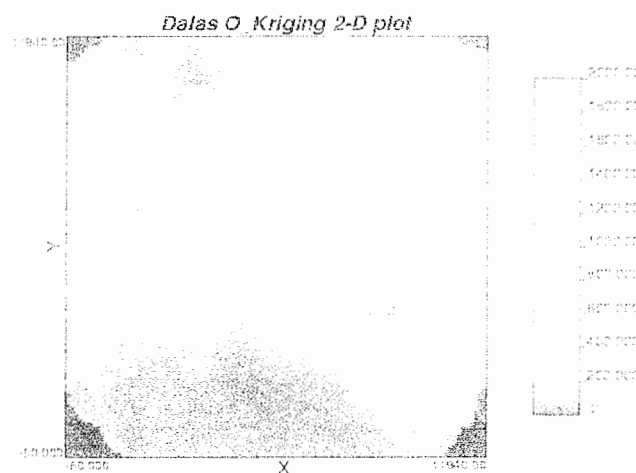
هم اکنون داده های نرمال شده جهت واریوگرافی بکار گرفته خواهد شد. نتایج واریوگرافی و سپس مدل سازی وریوگرام در WinGSlib مطابق شکل (۳-۵) می باشد. بر اساس این شکل مقدار Nugget=0.39، مقدار C=0.8 مدل کروی با شعاع تاثیر ۵۰۰۰ بعنوان مقادیر پارامترهای مدل وریوگرام در قسمتهای بعدی پروژه اعمال خواهد شد.



شکل ۳-۵) واریوگرافی و اعمال مدل واریوگرم به داده های نرمال شده. خروجی WinGSlib.

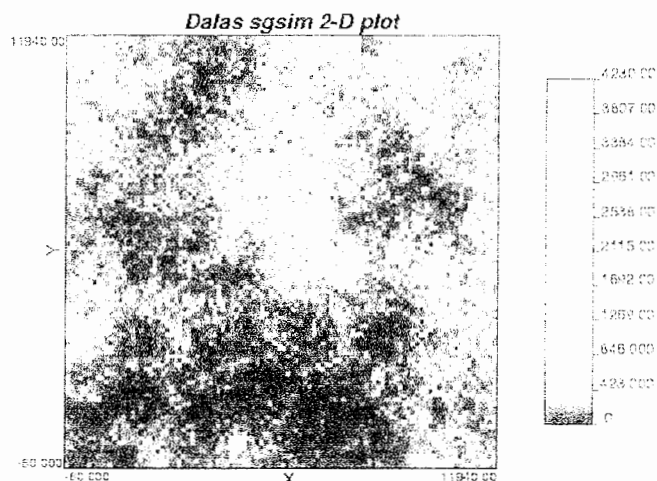
خروجی WinGSlib با هدف ایجاد شبکه توزیع مقادیر به روش کریجینگ مطابق شکل

(۴-۵) و به روش شبیه سازی مرحله ای گوسی مطابق شکل (۵-۵) خواهد بود.



شکل ۴-۵) تخمین شبکه بروش کریجینگ معمولی، فایل داده Dallas.dat (خروجی WinGSlib).





شکل ۵-۵) شبیه سازی بروش گوسی مرحله ای (خروجی WinGSlb).

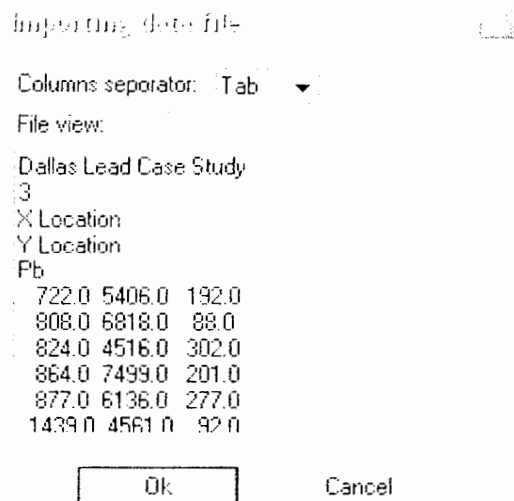
### ۵-۳- اجرای یک پروژه زمین آماری

#### ۵-۳-۱- ایجاد پروژه جدید در Gridat


ایجاد پروژه جدید در Gridat مستلزم مراحل و عملیات خاصی نمی باشد و صرفاً با ایجاد یک فایل داده جدید (گزینه New از منوی File) یا فراخوانی فایل داده در نرم افزار (گزینه Import منوی File) پروژه ایجاد می گردد. تنها توصیه ای که در این مورد مفید بنظر می رسد ایجاد یک شاخه جدید در مسیر مشخص جهت ایجاد پروژه می باشد، بدین منظور که فایل داده و تنظیمات پروژه (که تنها اجزای ذخیره شونده توسط Gridat می باشند) در شاخه جداگانه ای قرار داشته باشند.

#### ۵-۳-۲- بررسی فایل‌های داده و آماده سازی داده ها در Gridat

پس از اجرای نرم افزار با انتخاب گزینه Import از منوی File کادر محاوره ای استاندارد فراخوانی نمایش داده می شود. فایل داده Dallas.dat را در این کادر بیابید و با فشردن کلید Open کادر محاوره ای شکل (۵-۶) ظاهر می گردد. در این کادر می توانید نمایشی از محتویات فایل مذکور را مشاهده نمایید. از آنجایی فرمت Geo-EAS مورد نیاز Gridat نمی باشد توجه به ستونهای فایل داده معطوف می گردد. جدا کننده ستونها در این فایل کاراکتر فاصله (Space) می باشد. لذا در کنترل آبشاری با عنوان Columns separator عنوان Space را انتخاب کنید.

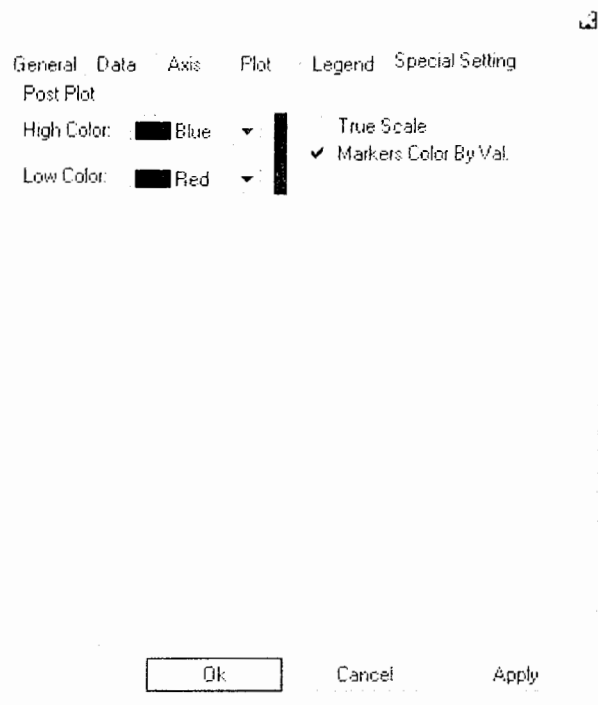


شکل ۵-۶) کادر محاوره ای دریافت داده با فرمت .dat در Gridat

با فشردن کلید Ok اطلاعات موجود در فایل به کنترل نمایشگر داده ها منتقل می گردد. بهتر است در این فایل چند سطر اول و ستون اول (که خالی می باشند) را حذف نمایید. این کار از طریق کلیک راست نمودن بر روی شبکه و انتخاب زیر گزینه های Delete در منوی شناور ظاهر شده ممکن می باشد. با انتخاب گزینه Command and Options... در این منو کادر محاوره ای تنظیمات فایل داده ظاهر می شود. در زبانه Workbook نام شبکه جاری را از Sheet1 به Dallas تغییر دهید. گزینه Save را بمنظور ذخیره سازی پروژه انتخاب نمایید. در کادر محاوره ای استاندارد ویندوز ابتدا با فشردن کلید  شاخه جدیدی با نام Dallasproj ایجاد نمایید و پروژه را در این شاخه با نام dallasproj ذخیره کنید.

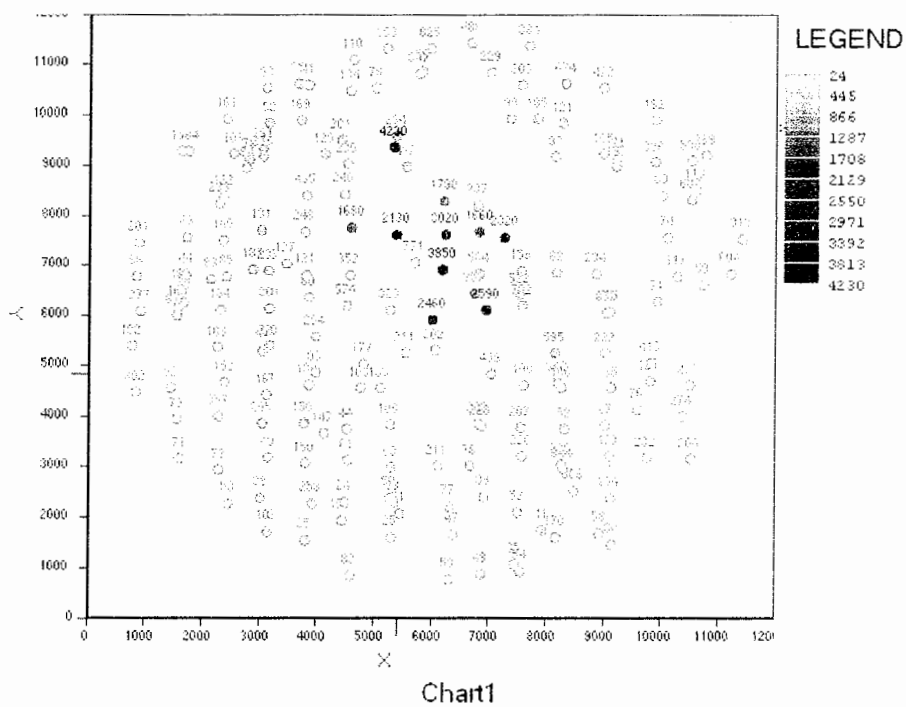
اگر هم اکنون از طریق محیط سیستم عامل به شاخه مذکور مراجعه کنید با دو فایل مواجه می شوید. یکی با نام dallasproj.gda که فایل تنظیمات پروژه است و دیگری dallasprojdat.xml که فایل داده ها است و توسط نرم افزارهای مختلف از جمله ماکروسافت اکسل قابل مشاهده می باشد.

جهت مشاهده نقشه توزیع مکانی نقاط نمونه برداری گزینه Post از منوی Action را انتخاب نمایید. کادر محاوره ای شکل (۵-۷) ظاهر می گردد.



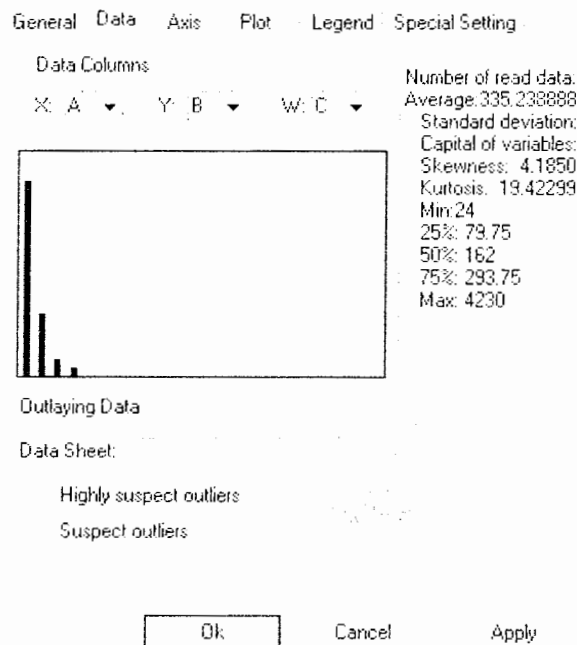
شکل ۵-۷) کادر محاوره ای ایجاد نقشه نقاط نمونه برداری در Gridat

در صورت تمایل می توانید رنگ نمایش نقاط نمونه برداری را در این کادر تغییر دهید و با انتخاب زبانه Plot تغییرات گرافیکی دیگر ممکن می باشد. توصیه می شود با انتخاب گزینه W Value Related Labels از زبانه Plot مقادیر اندازه گیری شده نقاط را در نقشه مشخص نمایید. در زبانه Legend گزینه مربوط به نمایش راهنما را انتخاب نموده و تنظیمات را با فشردن کلید Ok ثبت نمایید. نقشه نمایش نقاط نمونه برداری بلافاصله ظاهر می گردد (شکل ۵-۸). این نقشه را با خروجی WinGSLib (شکل ۵-۱) مقایسه نمایید.



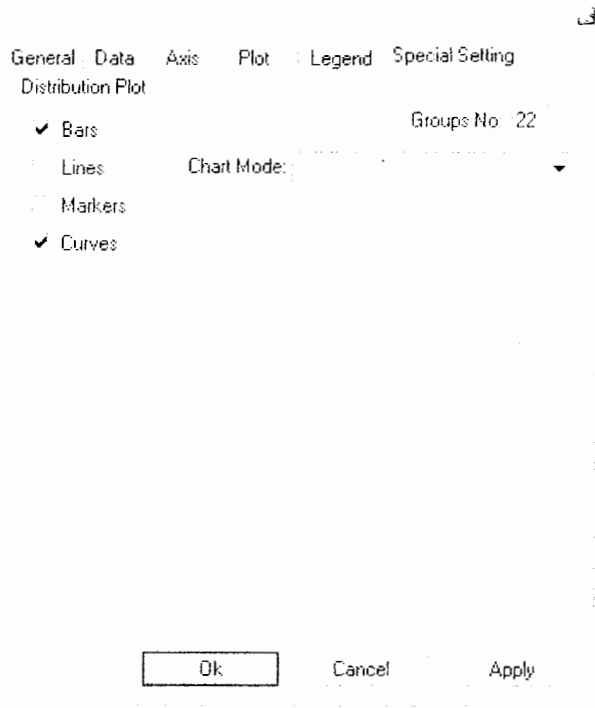
شکل ۵-۸) نقشه نمایش نقاط نمونه برداری ایجاد شده در Gridat

به منظور بررسی تابع توزیع مقادیر از منوی Action گزینه Distribution و زبانه Data از کادر محاوره ای ظاهر شده را انتخاب کنید. همانطور که ملاحظه می شود (شکل ۵-۹) مقادیر ستونهای A و B بعنوان مقادیر پیش فرض مختصات داده ها و مقدار C بعنوان ستون معادل مقادیر متغیر منظور شده اند. نمایش میله ای هیستوگرام توزیع حکایت از چولگی مثبت و توزیع لاگ نرمال مقادیر دارد ( در Gridat کلیه مجموعه ها دارای توزیع نرمال یا لاگ نرمال فرض می شوند). از کادر مربوط به پارامترهای آماری می توانید مقادیری نظیر میانگین، واریانس و چولگی را استخراج نمایید. اگر در این کادر راست کلیک نمایید منویی ظاهر می شود که انتخاب تنها گزینه آن با عنوان Send to legend منجر به انتقال محتویات این کادر به کادر Discription در زبانه Legend می گردد.

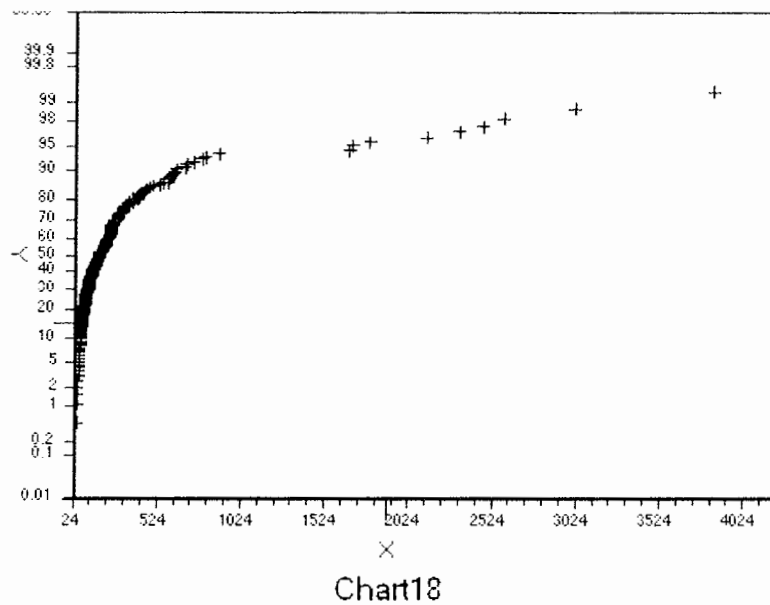


شکل ۵-۹) زبانه Data، کادر محاوره ای تنظیمات مربوط به گزینه Distribution در Gridat

هم اکنون اگر در زبانه Special Setting و از کنترل آبخاری Chart Mode (شکل ۵-۱۰) گزینه Probability Plot را انتخاب کنید نمودار احتمال مقادیر توسط نرم افزار ایجاد و نمایش داده می شود. برای نمایش بهتر چارت در زبانه Axis مقدار Step را برابر با ۱۰۰ و مقدار Interval را ۵ قرار داده و شکل نمایشگرها را در زبانه Plot به شکل "+" تغییر دهید. خروجی (شکل ۵-۱۱) را با خروجی WinGSlip (شکل ۵-۲) مقایسه کنید. آنچه از این قسمت پروژه نتیجه گرفتیم نیاز به تغییر در مقادیر اولیه داده ها به حالت توزیع نرمال می باشد که در مراحل بعدی اجرا لحاظ خواهد شد.



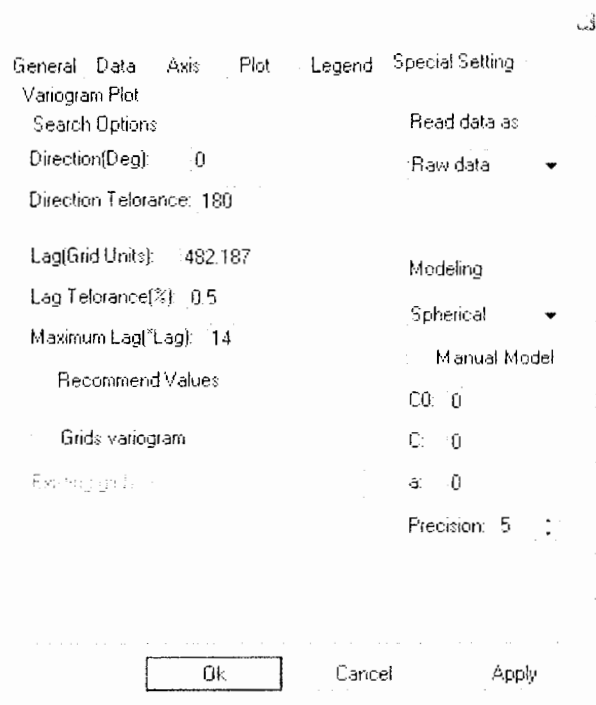
شکل ۵-۱۰) زبانه Special Setting، کادر محاوره ای تنظیمات مربوط به گزینه Distribution در Gridat



شکل ۵-۱۱) نمودار احتمال ایجاد شده توسط Gridat

### ۵-۳-۳- واریوگرافی و مدل سازی وریوگرام در Gridat

به منظور واریوگرافی و مدل سازی وریوگرام گزینه Variogram از منوی Action را انتخاب نمایید. کادر محاوره ای شکل (۵-۱۲) ظاهر می گردد. همانطور که ملاحظه می کنید مقادیر امتداد، گام و حدود تغییرات مجاز هر یک بطور پیش فرض ثبت شده است. حدود مجاز تغییرات امتداد همواره برابر با ۱۸۰ درجه منظور می شود و مقدار حدود تغییرات گام با توجه به توزیع نقاط برداشت محاسبه شده است. پیشنهاد می شود اگر مقدار حدود تغییرات گام برابر با ۰/۵ است (۵۰ درصد طول گام) مقدار آن را به ۰/۳ و مقدار حدود تغییرات امتداد را برابر با مقادیر کوچکتر (مثلا ۷۰ درجه) تغییر دهید و در صورتی که خروجی نهایی دلخواه عاید نشد می توانید با فشردن کلید Recommend Values مقادیر پیش فرض را مجددا محاسبه نمایید. از آنجایی که توزیع مقادیر لاگ نرمال تشخیص داده شده است، در کنترل آبشاری Read data as عنوان Normalized data را انتخاب کنید.



شکل (۵-۱۲) زبانه Special Setting، کادر محاوره ای تنظیمات مربوط به گزینه Variogram در Gridat

با رها نمودن گزینه Manual Model در کادر Modeling در حالت انتخاب نشده، مدل سازی وریوگرام را به نرم افزار واگذار نمایید. با فشردن کلید Ok، نمایشگر روند اجرای عملیات در پایین فرم اصلی (میله نشانگر وضعیت) ظاهر شده و در نهایت حاصل عملیات نمایش داده می شود (شکل ۵-۱۳). مدل برازنده شده به چارت از نوع کروی می باشد که در پذیرش آن تردید نکنید. هر چند می توانید با راست کلیک بر روی چارت و انتخاب گزینه Setting و سپس در کادر محاوره ای تنظیمات با انتخاب زبانه Special Setting نوع مدل را به حالات دیگر نظیر گوسی، سینوسی و ... تغییر دهید و حتی پارامترهای آن را پس از انتخاب گزینه Manual Model شخصا وارد نمایید.

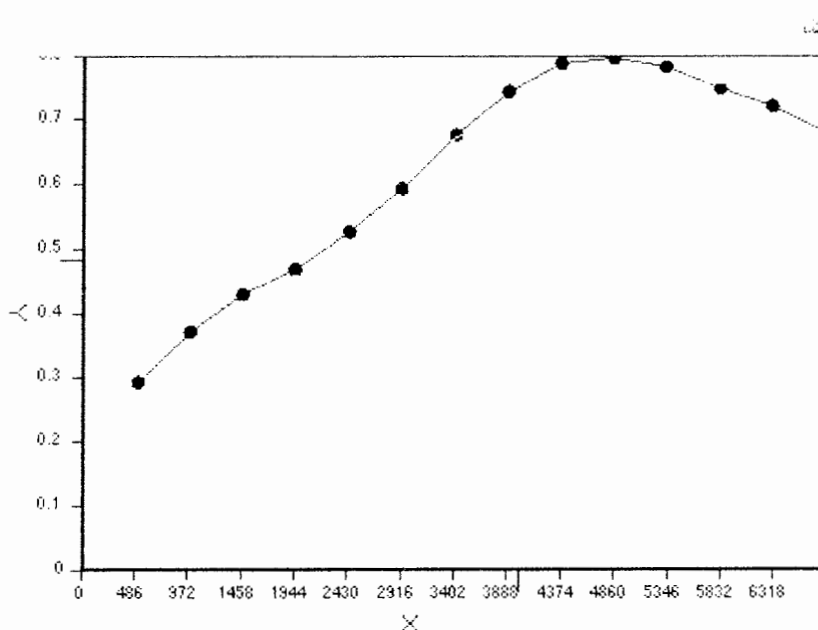


Chart3

شکل ۵-۱۳) نمایش وریوگرام در Gridat

قبل از فشردن کلید Ok و ایجاد خروجی وریوگرام به زبانه Legend مراجعه نمایید. در این زبانه کادری با عنوان Distribution وجود دارد که مطالب ثبت شده در آن در کادر راهنمای نقشه نمایش داده خواهد شد. قبل از ایجاد گراف این کادر خالی می باشد و می توانید مطالب مورد نظرتان را در آن تایپ نمایید. ولی توصیه می شود آن را خالی رها نموده و پس از ایجاد گراف



با فراخوانی مجدد کادر محاوره ای تنظیمات در این مورد تصمیم گیری نمایید. با توجه به اینکه پس از ایجاد هر گراف مقادیر پارامترهای آماری و یک سری اطلاعات اضافی در این کادر ثبت می گردد، می توانید با حذف و افزودن مطالبی در این کادر توضیحات مناسب را به راهنمای نقشه بیافزایید. اگر مایل نیستید این اطلاعات در راهنما نمایش داده شود، گزینه Distribution را از حالت انتخاب شده خارج نمایید.

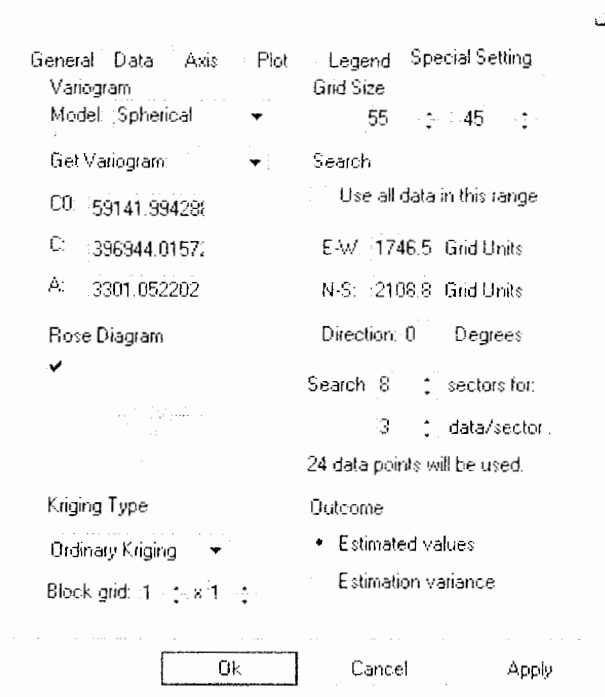
در این زبانه گزینه دیگری با عنوان Grids variogram مشاهده می کنید که بررسی آن را به بخشهای بعد موكول می نمایم. خروجی برنامه را با شکل (۳-۵) که خروجی WinGSLib می باشد و با مشقت فراوان ایجاد شده مقایسه کنید.

### ۵-۳-۴- ایجاد شبکه مقادیر در Gridat

به مانند قسمت قبل شبکه توزیع منطقه ای مقادیر را بروشهای کریجینگ و شبیه سازی گوسی مرحله ای ایجاد می نمایم.

### ۵-۳-۴- روش تخمین

از منوی Action گزینه Kriging را انتخاب کنید تا کادر محاوره ای شکل (۵-۱۴) ظاهر گردد. تنظیمات پیش فرض بنحوی است که اگر توزیع مقادیر اولیه نرمال باشد می توانید هم اکنون کلید Ok را فشرده و شبکه تخمین را مشاهده فرمایید. لیکن توزیع لاگ نرمال در داده های اولیه شما را مجاب به انتخاب عنوان Lognormal Ordinary Kriging از کادر Kriging type می نماید که این امر منجر به ظاهر شدن گزینه های Back transformation به منظور باز گرداندن نتایج نهایی از حالت نرمال شده به توزیع اولیه و Standardize به منظور ایجاد توزیع نرمال استاندارد در مقادیر می شود. گزینه Back transformation را در این کادر انتخاب کنید. اگر مقادیر پیش فرض Block grid را تغییر ندهید، عملیات کریجینگ نقطه ای انجام خواهد شد.

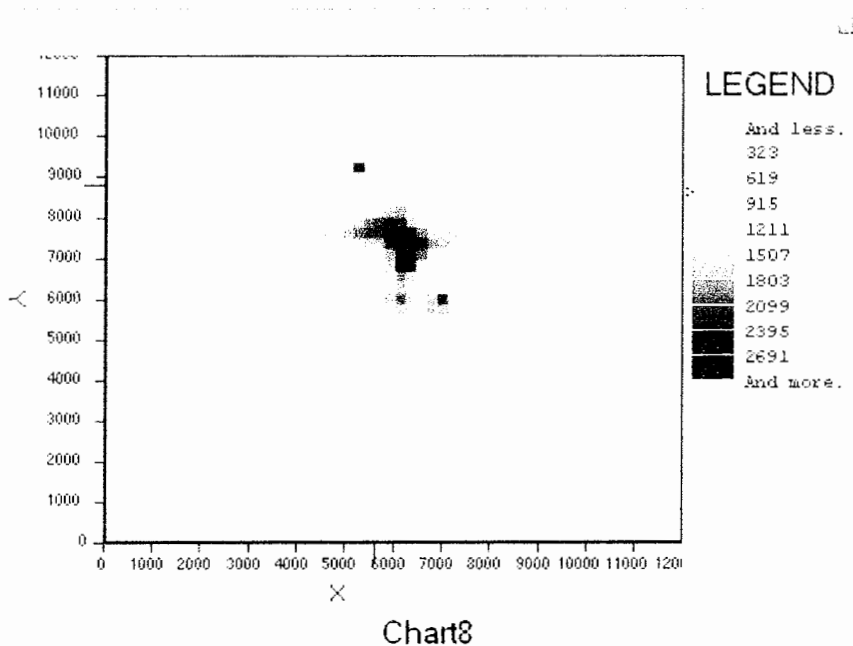


شکل ۵-۱۴) زبانه Special Setting، کادر محاوره ای تنظیمات مربوط به گزینه Kriging در Gridat

تنظیمات شعاع جستجو بر اساس بیضوی انیزوتروپی (کادر رز دیاگرام) انجام شده است و از آنجایی که بیضوی به خطوط دیاگرام انطباق قابل قبولی دارد نیازی به تغییر در تنظیمات جستجو نمی باشد. با این وجود می تواند با تغییر مقادیر کنترل‌های متنی E-W ، N-S و Direction تغییرات بیضوی را مشاهده کنید. ابعاد شبکه که بر اساس انیزوتروپی محیط محاسبه شده اند را می توانید آن را تا ۶۰۰×۸۰۰ نقطه تغییر دهید، ولی شبکه های بزرگتر از ۵۶۲۵ گره برای این نرم افزار توصیه نمی گردد.

در اینجا لازم است توجه تان را به مقادیر ثبت شده بعنوان پارامترهای مدل وریوگرام جلب نمایم. این مقادیر نیز همزمان با ایجاد رز دیاگرام محاسبه و در محلهای مربوطه ثبت می گردند که تقریباً برابر با مقادیر محاسبه شده برای مدل وریوگرامی می باشند که پس از تغییر داده های ورودی به حالت نرمال و با پذیرش کلیه محاسبات و پیشنهادات، توسط نرم افزار محاسبه شده است. ولی اگر گراف حاوی وریوگرام خاصی را در پروژه ایجاد نموده اید و قصد اختصاص پارامترهای مدل آن را به محاسبات تخمین دارید می توانید عنوان گراف وریوگرام مذکور را از کنترل آبشاری Get Varioqram انتخاب کنید تا مقادیر مدل مربوط به آن در محلهای مربوطه

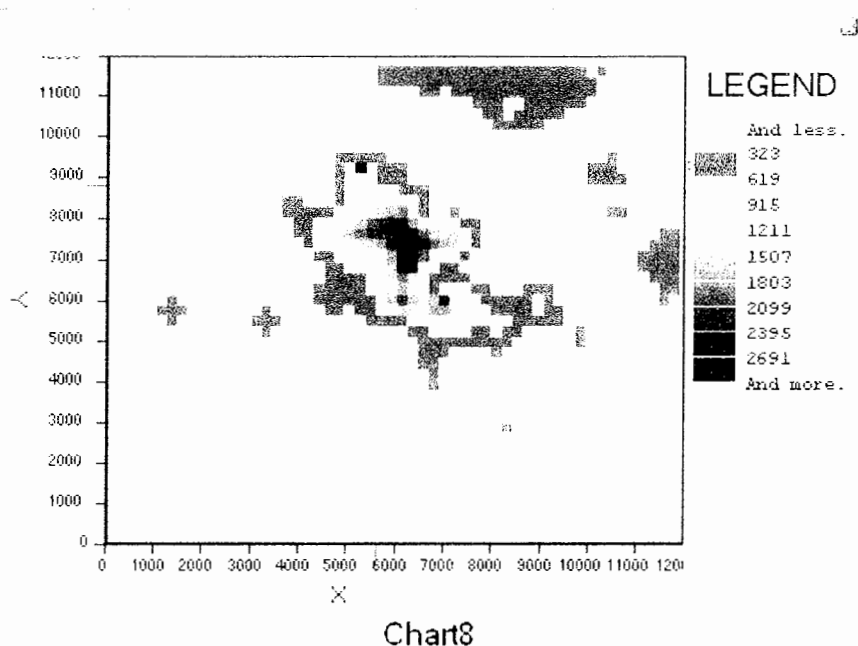
ثبت گردد. هم اکنون می توانید کلید Ok را کلیک کنید و منتظر مشاهده نتایج باشید ولی قبل از آن بهتر است از زبانه Legend گزینه مربوط به نمایش راهنمای نقشه را نیز انتخاب نمایید. نمایشگر روند اجرای برنامه در پایین فرم اصلی ظاهر می گردد و پس از چند ثانیه خروجی برنامه که فرم حاوی شبکه توزیع منطقه ای مقادیر تخمین است ظاهر می گردد. (شکل ۵-۱۵) این خروجی را با خروجی WinGSlub (شکل ۵-۴) مقایسه کنید. شکل کلی توزیع مقادیر در دو نقشه یکسان است، هر چند مقادیر ایجاد شده توسط Gridat هموار شدگی کمتری دارند و مقادیر تخمین از رنج تغییرات بیشتری برخوردارند که یکی از علل آن می تواند تفاوت شکل محدوده جستجو در دو گراف باشد.



شکل ۵-۱۵) نمایش شبکه تخمین بروش کریجینگ در Gridat

اگر به فایل داده مراجعه کنید متوجه خواهید شد یک سری داده جدید با نام Sheet4(NORMALIZED/Dallas) به مجموعه افزوده شده است. این شبکه حاوی مقادیر نرمال شده اولیه می باشد که طبق سفارش شما توسط نرم افزار ایجاد شده است.

قبل از گذر از این قسمت توجه تان را به ویژگی دیگر کادر راهنما جلب می‌کنم. هم‌اکنون این کادر شامل نشانگرهای رنگی می‌باشد که البته تک‌رنگ نیستند، بلکه شامل طیفی از رنگها می‌باشند که مقدار معادل حد بالایی و پایینی هر طیف در مقابل‌شان نوشته شده است و اگر روی هر کدام کلیک کنید کادر انتخاب‌رنگ استاندارد ویندوز ظاهر می‌گردد. انتخاب یک رنگ مشخص منجر به تغییر رنگ کلیه مقادیر معادل طیف مربوطه به رنگ انتخاب‌شده می‌گردد (شکل ۵-۱۶).



شکل ۵-۱۶) مشخص نمودن کلاس خاص مقادیر در نمایش شبکه تخمین، Gridat

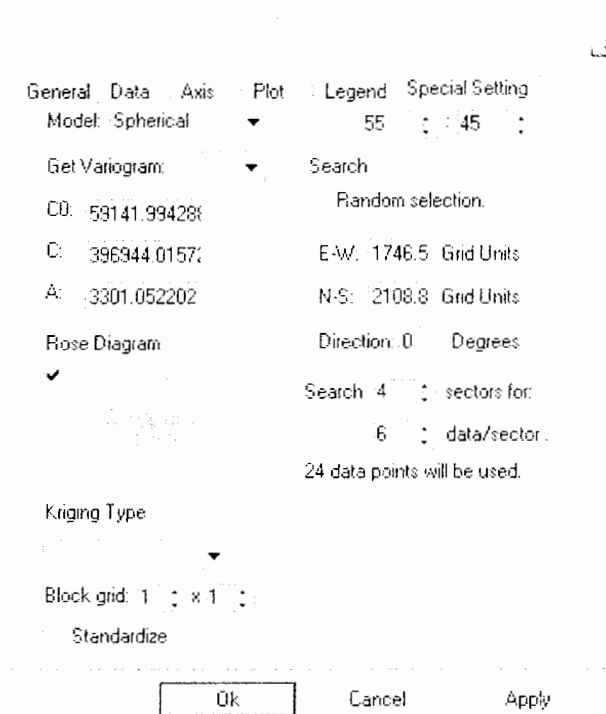
با راست کلیک نمودن بر روی کادر راهنما و انتخاب گزینه Reset نقشه را به حالت اول

باز گردانید.<sup>۲</sup>

<sup>۲</sup>- توصیه می‌کنم فعلاً گزینه‌های دیگر این منو را فراموش کنید

### ۵-۳-۴-ب- روش شبیه سازی مرحله ای گوسی

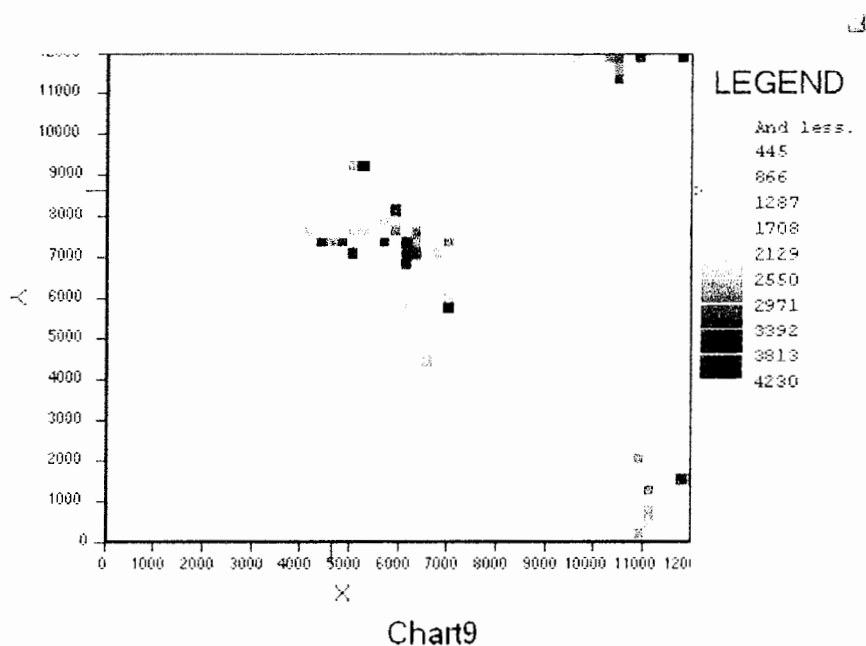
عملیات شبیه سازی مرحله ای نیز بطور مشابه انجام می گیرد. گزینه Sequential Simulation را از منوی Action انتخاب کنید تا کادر محاوره ای شکل (۵-۱۷) پدیدار گردد. تنظیمات و مقاردهی به پارامترها کاملا مشابه تنظیمات کریجینگ انجام می گیرد.



شکل (۵-۱۷) زبانه Special Setting، کادر محاوره ای تنظیمات مربوط به گزینه Sequential Simulation در Gridat

در این جا نیز لازم است در کادر Kriging type عنوان Lognormal Ordinary Kriging را انتخاب نمایید. با فشردن کلید Ok منتظر نمایش نتایج بمانید. خروجی این برنامه مشابه شکل (۵-۱۸) می تواند باشد. انتظار نداشته باشید این خروجی با خروجی WinGSlib (شکل ۵-۵) یکی باشد و حتی انتظار نداشته باشید خروجی که شما هم اکنون از Gridat گرفته اید دقیقا مشابه شکل (۵-۱۸) باشد. در نظر داشته باشید که اگر در Gridat، WinGSlib یا هر نرم افزار دیگری چندین شبکه شبیه سازی شده با تنظیمات کاملا مشابه ایجاد کنید، خروجی هر بار تفاوت خواهد

کرد و این از ویژگی های شبیه سازی است. لیکن در همه حالات شکل کلی توزیع منطقه ای مقادیر مشابه می باشد، نظیر آنچه در مقایسه خروجی WinGSlib و Gridat می بینید.



شکل ۵-۱۸) نمایش شبکه شبیه سازی بروش گوسی مرحله ای در Gridat

#### ۵-۴- مقایسه دو نرم افزار

نرم افزار WinGSlib مستلزم یک سری تمهیدات بمنظور ایجاد پروژه می باشد که این امر بواسطه لزوم تعیین مسیرهای فایل های ورودی، پارامتری و خروجی مقرر شده است. Gridat نیاز به عملیات خاصی جهت ایجاد پروژه ندارد و فراخوانی یا ایجاد فایل داده به معنی ایجاد پروژه ای جدید است.

همانطور که مشاهده نمودید آماده سازی داده ها در Gridat در حد نتیجه گیری بصری از خروجی های برنامه و آگاهی از وضعیت توزیع داده ها در طول اجرای پروژه است. بدین ترتیب که بررسی وضعیت توزیع مقادیر در حین و پس از ایجاد چارتهای توزیع فراوانی و نمودار احتمال و تغییرات فایل داده همزمان با اجرای هر قسمت از پروژه ممکن می باشد. لیکن فرض لاگ نرمال بودن توزیع در صورت عدم تبعیت آن از حالت نرمال موردی است که همواره و در کلیه مراحل اجرای پروژه بوسیله Gridat منظور می گردد. در حالی که WinGSlib این محدودیت را نداشته و

بدون توجه به وضعیت توزیع اولیه، سعی در بکارگیری مقادیر با توزیع نرمال استاندارد دارد. ایجاد و تاثیر دادن ضرایب حذف تجمع مورد دیگری است که در WinGSlib منظور شده است. تعدیل چگالی نقاط برداشت در Gridat صرفا با اتکا به روش جستجوی قطعی انجام می گیرد.

یکی از ویژگی های بارز Gridat خلاصه بودن مجموعه عملیات است. به عنوان مثال مشاهده هیستوگرام توزیع فراوانی داده ها در WinGSlib مستلزم اجرای چندین مرحله و مشاهده نتایج در محیط نرم افزار دیگر می باشد که این امر افزایش تعداد فایل های مربوط به پروژه را نیز بهمراه دارد. در حالی که مشاهده این نوع هیستوگرام بهمراه پارامترهای آماری و نمودار جعبه ای توزیع مقادیر اجزای فرعی و معمول کادرهای محاوره ای Gridat می باشند.

توجه در خروجی دو نرم افزار شباهت نتایج عملیات واریوگرافی و مدل سازی وریوگرام را نشان می دهد. لازم است به مختصر بودن و سادگی عملیات در Gridat نسبت به WinGSlib اشاره شود. درج مجموعه مقادیر پارامترهای اولیه در کادرهای محاوره ای مربوط به واریوگرافی و یافتن بهترین مدل وریوگرام در نرم افزار WinGSlib عملیات وقت گیری می باشد که طی چندین مرحله انجام می گیرد.

می توان گفت ابزار واریوگرافی و مدل سازی وریوگرام در Gridat از برجسته ترین و شاخص ترین گزینه ها می باشد و طراحی آن نیز وقت، حوصله و دقت بیشتری را به خود اختصاص داده است. علت آن هم واضح است، صحت هر نوع عملیات تخمین و شبیه سازی زمین آماری در درجه اول بستگی به دقت و صحت مدل سازی وریوگرام دارد.

گزینه وریوگرام علاوه بر ابزار یاد شده ارائه دهنده عملیات دیگری به منظور بررسی درستی نتایج تخمین و شبیه سازی بروش تایید متقاطع می باشد که به این ویژگی نیز اشاره خواهد شد.

الگوریتمهای تخمین و شبیه سازی ارائه شده در Gridat نسبت به آنچه WinGSlib ارائه می دهد مختصر و ساده می باشد. لیکن در طراحی این قسمت نرم افزار نیز سهولت اجرای عملیات مورد توجه بوده است. همانطور که مشاهده کردید پس از انتخاب گزینه های مربوطه مقادیر انیزوتروپی، پارامترهای مدل وریوگرام، بیضوی جستجو و شعاع تاثیر در کادر محاوره ای تنظیمات درج می گردد و کاربر می تواند بدون نگرانی به این محاسبات اعتماد کند در حالی که

تنظیم همین پارامترها در WinGSlib خود مقوله ای مفصل است که درج آنها صرفاً به تایپ اعداد در کادرهای محاوره ای محدود نمی شود و مهارت‌های ویژه کاربر را می طلبد.

ارائه همزمان تخمین و شبیه سازی بروشهای معمولی، لگاریتمی و شاخص ویژگی دیگر Gridat است که به اجرای هر چه سریعتر و ساده تر آن کمک نموده است.

در مجموع می توان گفت Gridat نسبت به مجموعه WinGSlib از سادگی اجرای چشمگیری برخوردار است. علت آن نیز یکی رعایت اصول استاندارد طراحی نرم افزار در Gridat و دیگری بکار بردن امکانات جدید برنامه نویسی و ساخت نرم افزار در طراحی است که پی آمد آن امکان مشاهده همزمان داده ها، تنظیمات و نتایج در محیط یک نرم افزار و امکان ذخیره سازی آن در قالب دو فایل داده و تنظیمات می باشد. هر چند نبود امکانات بررسی داده های چند متغیره و همچنین امکانات محاسبات و نمایش سه بعدی نیز در خلاصه نمودن نرم افزار نقش داشته است.

در نهایت می توان چنین اظهار نظر نمود که نرم افزار Gridat از نظر ویژگی های گرافیکی، دسترسی به داده ها و نتایج، تکامل محیط اجرا و مدیریت پروژه نسبت به WinGSlib ارجح می باشد و در مقابل نرم افزار مذکور از نسبت به Gridat امکانات بیشتری در زمینه بکار گیری داده های چند متغیره و یا مدل سازی بلوکی (داده ها و خروجی های سه بعدی) برخوردار است.

## ۵-۵- دیگر ابزار Gridat

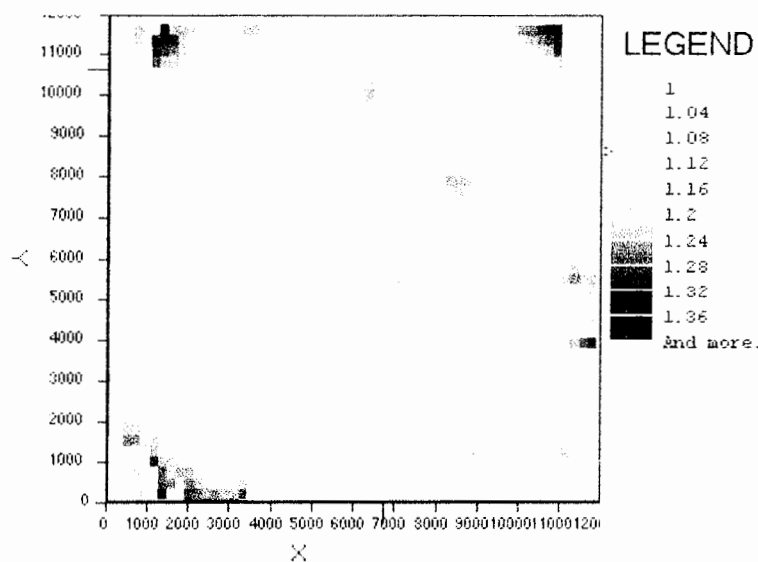
علاوه بر امکانات فوق که جهت محک زدن Gridat به آنها اشاره شد، مجموعه ای از ابزار و گزینه ها در نرم افزار منظور شده است که لازم است مورد توجه قرار گیرد. از جمله می توان به ایجاد شبکه های توزیع مقادیر واریانس اشاره نمود.

بدین منظور عنوان شبکه ای را که بروش کریجینگ ایجاد شده در کادر Project Explorer بیابید و با راست کلیک نمودن بر روی آن و انتخاب گزینه Setting کادر محاوره ای مربوط به آن را نمایان سازید. در زبانه Special Setting و در کادر Outcome گزینه Estimation variance را انتخاب کنید. لازم است که گزینه Transformed data sheet را از



حالت انتخاب خارج نمایید، زیرا این سری داده را قبلا ایجاد کرده ایم و نیازی به ساخت مجدد آن نمی باشد.

با فشردن کلید Ok شبکه توزیع منطقه ای مقادیر واریانس ایجاد می شود. خروجی مطابق شکل (۵-۱۹) است. تصویر در نگاه اول کمی عجیب به نظر می رسد. جهت پی بردن به راز آن در حالی که گراف فعال است گزینه Chart از منوی Format را انتخاب کنید تا کادر محاوره ای تنظیمات گراف ظاهر گردد. در زبانه Plot اندازه نمایشگر نقاط را برابر با ۶ قرار دهید و در صورتی که راهنمای نقشه نمایش داده نشده در زبانه Legend گزینه نمایش راهنما را انتخاب کنید. همانطور که انتظار می رود واریانس تخمین حول نقاط برداشت نمونه دارای مقادیر کمتری است. مشابه این عملیات برای شبکه های شبیه سازی شده نیز ممکن می باشد.

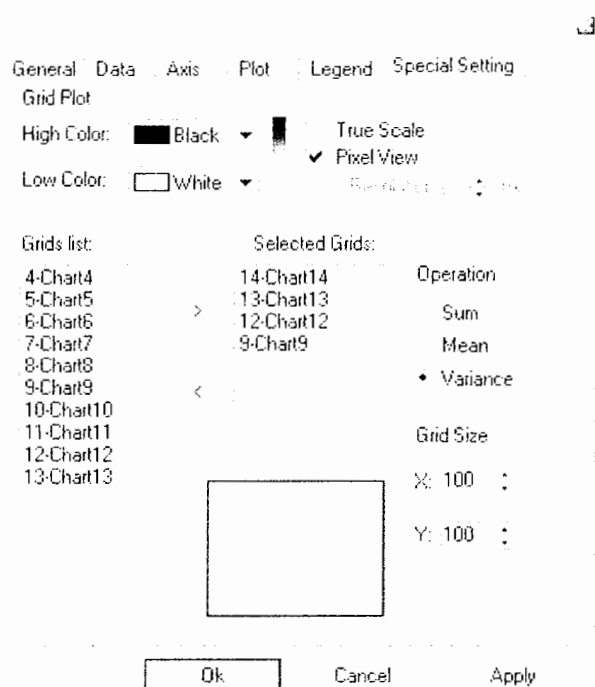


شکل (۵-۱۹) نمایش شبکه واریانس تخمین بروش کریجینگ در Gridat

روش کار این چنین است که ابتدا به تعداد ۴ تا ۵ شبکه شبیه سازی شده با تنظیمات کاملا یکسان ایجاد نمایید. این کار چند دقیقه ای وقت شما را می گیرد. حین ایجاد گرافها به نامگذاری آنها توجه فرمایید. نرم افزار بطور پیش فرض نامگذاری ترتیبی را برای گراف ها منظور

می دارد. بهتر است اسامی این چند گراف را از طریق زبانه General در کادر محاوره ای تنظیمات چنان تغییر دهید که از مابقی گرافها مجزا باشند.

پس از ایجاد شبکه ها از منوی Action گزینه Mutiple Grid را انتخاب کنید. کادر محاوره ای شکل (۲۰-۵) ظاهر می گردد. عناوین گرافهایی را که بروش شبیه سازی ایجاد نموده اید در لیست شبکه ها انتخاب کرده و به لیست Selected Grids بیافزائید. در کادر Operation گزینه Variance را انتخاب کنید و پس از انتخاب گزینه نمایش راهنما، کلید Ok را بفشارید تا خروجی نظیر شکل (۲۱-۵) ظاهر گردد. این خروجی را با اولین گرافی که ساختیم، یعنی گراف نمایش نقاط نمونه برداری مقایسه کنید. می توانید مانند قبل این نقاط را بر روی شبکه هم نمایش دهید. برابر با انتظار شما واریانس محاسبات در گوشه های محدوده دایره ای شکلی که برداشت صحرائی در آن انجام نشده و در ناحیه میانی که مقادیر نمونه ها دچار تغییرات بیشتری نسبت به دیگر نقاط هستند افزایش نشان می دهد.



شکل (۲۰-۵) زبانه Special Setting، کادر محاوره ای تنظیمات مربوط به ایجاد شبکه مرکب در Gridat

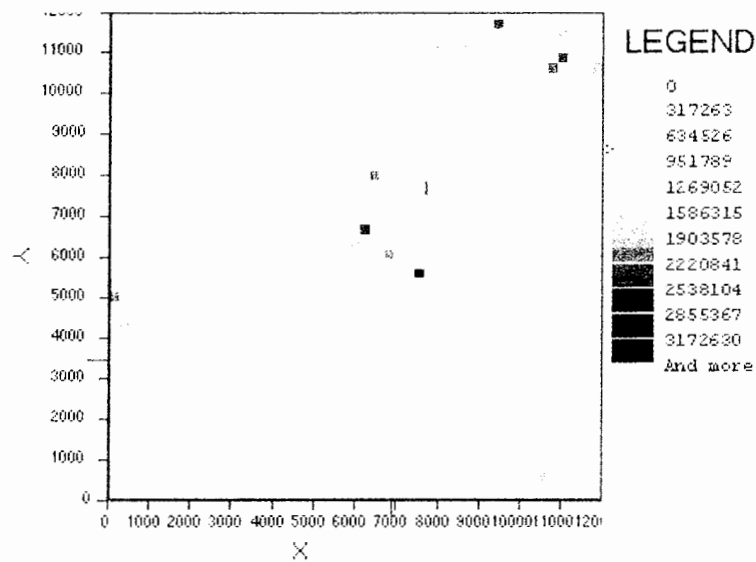


Chart17

شکل ۵-۲۱) خروجی حاصل از محاسبه واریانس مقادیر چندین شبکه در Gridat

روش یاد شده روش مناسبی جهت بررسی درستی نتایج شبکه های شبیه سازی شده دیگر نظیر شبکه های انیلینگ می باشد. جهت انجام این نوع شبیه سازی از منوی Action گزینه مربوطه را انتخاب کنید تا کادر محاوره ای تنظیمات شبیه سازی انیلینگ ظاهر گردد. توجه کنید که روش انیلینگ به داده ها با مقادیر بالای چولگی مثبت نظیر داده های پروژه حاضر حساسیت دارد و ممکن است نتوانید بسادگی نتایج مطلوب را از آن بدست آورید. لیکن انتخاب گزینه Logarithmic data و افزایش دفعات جابجایی نقاط تا ۳۰۰۰ دفعه می تواند پس از صرف مدت زمان نسبتاً طولانی تر، شبکه ای مشابه شکل (۵-۲۲) ایجاد نماید.

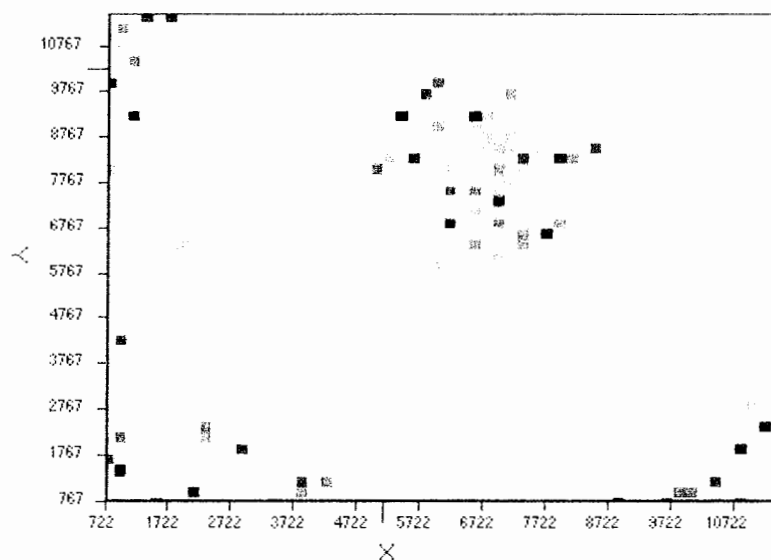


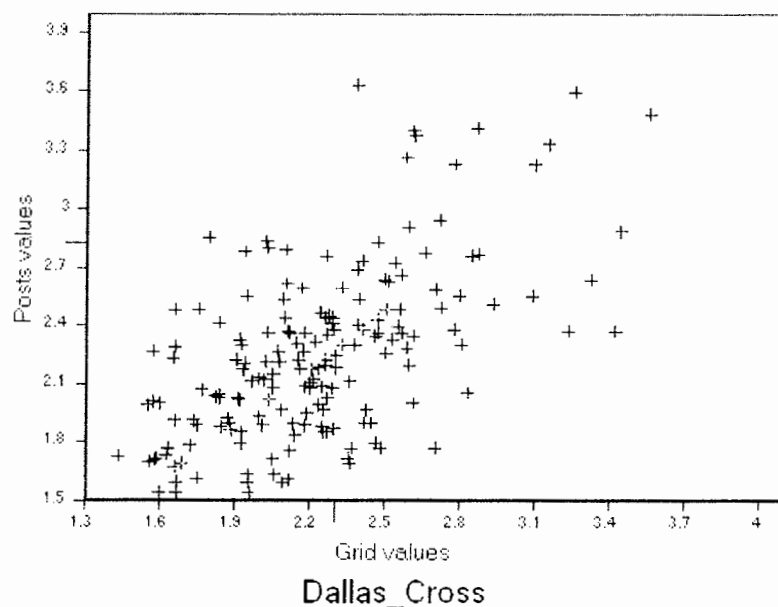
Chart17

شکل ۵-۲۲) خروجی شبیه سازی انیلینگ در Gridat

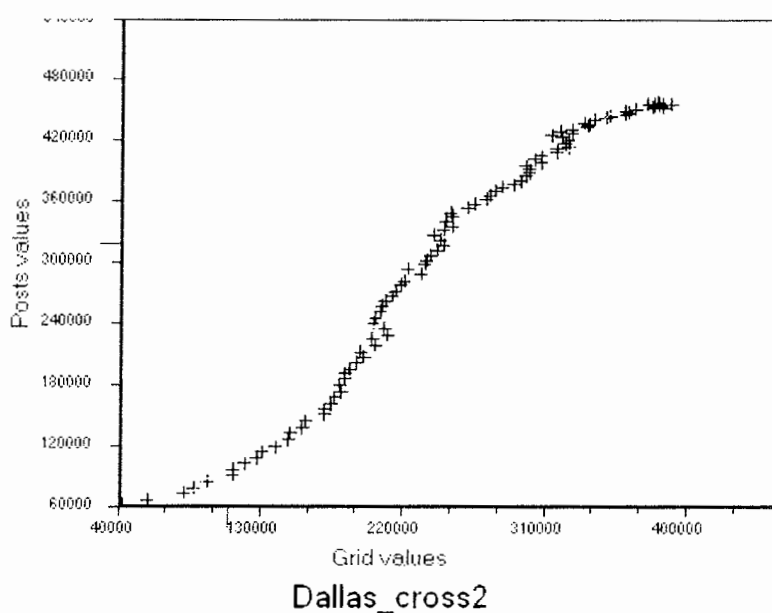
توجه کنید که مقادیر در گوشه های نقشه که فاقد نقاط نمونه برداری بوده اند همچنان از تابع توزیع منطقه ای تصادفی؛ مطابق با مقادیر تابع توزیع هیستوگرام اولیه برخوردارند. به این دلیل است که روش ایجاد چندین شبکه جهت بررسی خروجی این الگوریتم پیشنهاد می شود، با وجودی که ابزار دیگری جهت آزمون نتایج شبکه سازی در نرم افزار وجود دارد.

اساس کار ابزار یاد شده مقایسه مقادیر تخمین زده شده با مقادیر اولیه در نقاطی که دارای مقدار اولیه می باشند و یا ایجاد وریوگرام مقادیر شبکه شبیه سازی شده و مقایسه آن با وریوگرام مقادیر اولیه بروش تایید متقاطع می باشد. بعبارت دیگر روش تایید متقاطع در حالتی که به شبکه تخمین اعمال شود مقادیر اولیه و تخمین را مقایسه می کند و در حالتی که قصد اعمال آن به شبکه شبیه سازی شده را داریم مقادیر مدل وریوگرام اولیه و شبکه مورد توجه خواهند بود. به منظور اجرای این روش از منوی Action گزینه وریوگرام را انتخاب کنید و پس از انتخاب گزینه Grids variogram عنوان شبکه مذکور (که توصیه می شود از نوع تخمین یا شبیه سازی شده مرحله ای گوسی باشد) را در کنترل آبخاری با نام Existing grids بیابید. توجه کنید

که در این لیست فقط شبکه هایی لیست شده اند که وابسته به فایل داده های جاری می باشند.  
وظیفه بعدی کلیک نمودن کلید Ok و منتظر ماندن است.



شکل ۵-۲۳) تایید مقاطع برای شبکه های تخمین در Gridat



شکل ۵-۲۴) تایید مقاطع برای شبکه های شبیه سازی در Gridat

چارت خروجی بسته به نوع شبکه انتخاب شده حاوی دو محور در بردارنده مقادیر اولیه و مقادیر شبکه (در حالتی که شبکه اولیه از نوع تخمین باشد، شکل ۵-۲۳) و یا مقادیر وریوگرام اولیه و وریوگرام شبکه (در حالتی که شبکه اولیه از نوع شبیه سازی شده باشد، شکل ۵-۲۴) خواهد بود. به همین ترتیب خط منطبق شده به نقاط بیان گر نزدیکی مقادیر تخمین و اولیه و یا میزان شباهت وریوگرام هاست که معادله و ضریب رگرسیون آن را می توانید با راست کلیک نمودن بر روی عنوان آن در پنجره Project Explorer و انتخاب گزینه Report مشاهده نمایید.

## فصل ششم

### نتیجه گیری و پیشنهادات

## فصل ششم

### نتیجه گیری و پیشنهادات

آنچه می توان بعنوان جمع بندی کلی از پایان نامه حاضر بیان نمود چنین خلاصه می گردد:

مطالعه و توجه به اصول و شیوه های اجرای پروژه های زمین آماری به وضوح بیانگر این مطلب است که استفاده از نرم افزار در حل مسائل مرتبط با پروژه هایی اینچنین سودمند، راهگشا و حتی ناگزیر می باشد و لذا دسترسی به نرم افزارهای مناسب زمین آماری در ردیف ملزومات و ابزار اولیه طراحان و مدیران درگیر با این نوع پروژه ها می باشد.

از جمله روشهای تهیه هر نرم افزار؛ بغیر از خریداری آن، طراحی نرم افزار توسط برنامه نویسان آشنا با اصول تئوریک مربوطه است. بدین ترتیب می توان بسته به نوع نیاز و وضعیت پروژه، طراحی خاصی برای نرم افزار منظور نمود و آن را در سطح نیازهای پروژه توسعه داد.

نرم افزار Gridat که در محیط Visual Studio .Net و با هدف ارائه یک مجموعه شامل اجرای روشهای اصلی زمین آمار طراحی شده، حایز ویژگی های خاصی می باشد که در نرم افزار های مرسوم فعلی کمتر به چشم می خورد. رعایت اصول استاندارد طراحی محیط نرم افزار و به تبع آن سهولت تنظیم پارامترهای ورودی، محاسبه مقادیر پیش فرض پارامترهای اولیه، امکان مشاهده و ویرایش فایل های داده در نرم افزار و به صورت مجزا از قسمت های دیگر و امکان مشاهده همزمان خروجی به صورت گرافیکی در محیط نرم افزار از جمله ویژگی های آن است. همچنین امکان فراخوانی فایل های داده با فرمت های معمول و مرسوم و نیز امکان ارسال خروجی ها در قالب فایل های تصویر یا داده از خصوصیات دیگر نرم افزار می باشند.



طراحی و ساخت نرم افزار Gridat به صراحت بیان می دارد که استفاده کنندگان از هر نرم افزار نباید لزوما در لیست خریداران آن قرار داشته باشند و همواره راه حل دیگری با نام طراحی نرم افزار بر اساس نیازهای کاربران وجود دارد. امروزه نادیده گرفتن این مطلب عباراتی نظیر غفلت و عقب ماندگی را به همراه خواهد داشت.

نرم افزار Gridat دارای حداقل ابزار لازم جهت اجرای یک پروژه زمین آماری در حالت دو بعدی می باشد. بطور کلی امکانات مجموعه شامل موارد زیر است:

- ✓ بررسی توزیع آماری مقادیر اولیه
  - ✓ امکانات لازم جهت واریوگرافی و مدل سازی وریوگرام
  - ✓ ایجاد شبکه های تخمین بر روش کریجینگ، لاگ کریجینگ، کریجینگ شاخص
  - ✓ ایجاد شبکه های شبیه سازی مرحله ای گوسی و شاخص و نیز شبیه سازی انیلینگ
- داده های ورودی همواره دو بعدی است. هر چند فایل داده امکان ثبت چندین ستون داده را فراهم می آورد ولی نرم افزار صرفا قادر به انجام عملیات روی یک ستون داده بعنوان مقادیر متغیر می باشد که با توجه به انجام مطالعه موردی و مقایسه نرم افزار با مجموعه WinGSlib توانایی Gridat در ارائه نتایج مطلوب به اثبات می رسد.
- جهت گسترش کاربرد نرم افزار موارد چندی بدین شرح می تواند مورد توجه باشد:
- ✓ گسترش نرم افزار با هدف انجام عملیات زمین آماری سه بعدی
  - ✓ توانایی ساخت و نمایش مدل های سه بعدی
  - ✓ افزودن امکاناتی جهت اجرای دیگر روشهای کریجینگ به نرم افزار
  - ✓ ایجاد امکان اجرای شبیه سازی به روشهای دیگر نظیر LU Sim.
  - ✓ امکان انجام عملیات بر روی داده های چند متغیره با هدف اجرای عملیاتی نظیر کوکریجینگ.

1. Deutsch, C.V. & Journel, A.G., **GSLIB, Geostatistical software library and users guide**, Oxford university press, 1992.
2. Dubois, G., **Geostatistical software suggestions**, [www.ai-geostats.org](http://www.ai-geostats.org), Last updated: 16 Aug. 2004,
3. Gotway, C. A. and Rutherford, B. M., **The components of geostatistical simulation**, Department of energy documents (DOE), Office of scientific and technical information, [www.osti.gov](http://www.osti.gov), 2001.
4. James J., **Scatter plots and regression lines**, Lecture notes, Chapter11, [www.richland.edu](http://www.richland.edu), Last updated January 19.
5. Kosakowski, G., **Introduction to Applied Geostatistics 1**, Waste management laboratory, Paul Scherrer Institute, Switzerland, [people.web.psi.ch/kosakowski](http://people.web.psi.ch/kosakowski) 2001.
6. Kosakowski, G., **Applied Geostatistics 2, Part 1: Kriging**, Waste management, laboratory, Paul Scherrer Institute, Switzerland, [people.web.psi.ch/kosakowski](http://people.web.psi.ch/kosakowski), 2002.
7. Kosakowski, G., **Applied Geostatistics 2, Part 3: Simulations**, Waste, management laboratory, Paul Scherrer Institute, Switzerland, [people.web.psi.ch/kosakowski](http://people.web.psi.ch/kosakowski), 2002.
8. Mueller, T., **A summary of some geostatistical software conventions**, Dep. of agronomy, Univ. of Kentucky, [www.ai-geostats.org](http://www.ai-geostats.org), Last changed: 2 Feb 2005.
9. Shibli, S.A., **The entire geostatistics FAQ**, [www.ai-geostats.org](http://www.ai-geostats.org), Copyright 1996-1997.
10. Statios PDF resources ([www.statios.com/Resources/](http://www.statios.com/Resources/)), last update 2004, Copyright 2000.
11. Tauzin, E., **Integration of well test data into stochastic modeling**, A report submitted to the department of petroleum engineering of Stanford University, [citeseer.ist.psu.edu](http://citeseer.ist.psu.edu), April 1995.

12. Testing Normality, UITS Center for statistical and mathematical computing,  
<http://pytheas.ucs.indiana.edu>, 2001.

۱۳. آر. اچ. پنینگتون، آنالیز عددی و روشهای کامپیوتری، ترجمه پرویز جبه دار مارالانی و منصور نیکخواه بهرامی، انتشارات دانشگاه تهران، تیرماه ۷۳.

۱۴. دکتر علی اصغر حسنی پاک، زمین آمار (ژئواستاتیسستیک)، انتشارات دانشگاه تهران، ۱۳۷۷.

۱۵. دکتر رضا کاکائی، جزوه درسی زمین آمار، دانشگاه صنعتی شاهرود، ۱۳۸۲.

پیوست

## پیوست الف: کدهای منبع

پیوست الف-۱) کد منبع الگوریتم نمایش نقاط نمونه برداری

```
private void build()
{
    if(sav.Mins[0] == sav.Maxs[0] || sav.Mins[1] == sav.Maxs[1])
    {
        sav.Maxs[0]++;
        sav.Maxs[1]++;
        MessageBox.Show("Invalid limits has been
entered.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    //else
    //{
    if((double)sav.Reg[41] != 0 || (double)sav.Reg[42] != 0)
    {
        sav.Mins[0] = (double)sav.Reg[41];
        sav.Maxs[0] = (double)sav.Reg[42];
        sav.Mins[1] = (double)sav.Reg[51];
        sav.Maxs[1] = (double)sav.Reg[52];
    }
    else
        roundExterms();

    sav.MarkerList = new Point[sav.postList.Count];
    sav.ColorList = new Color[sav.postList.Count];
    sav.LabelList = new string[sav.postList.Count];

    int m, n;
    EachPoint pl;
    for (int i=0; i<sav.postList.Count; i++)
    {
        pl = (EachPoint)sav.postList[i];
        m = (int)((pl.X - (sav.Mins[0])) * 800 / ((sav.Maxs[0] -
sav.Mins[0])));
        n = (int)((pl.Y - (sav.Mins[1])) * 600 / ((sav.Maxs[1] -
sav.Mins[1])));

        sav.MarkerList[i].X = m;
        sav.MarkerList[i].Y = n;

        string st = "";
    }
}
```

```

        if((bool)sav.Reg[31]) st =
(Math.Round(pl.X, (int)sav.Reg[32])).ToString();
        if((bool)sav.Reg[31] && (bool)sav.Reg[33]) st += ",";
        if((bool)sav.Reg[33]) st +=
(Math.Round(pl.Y, (int)sav.Reg[34])).ToString();
        if(((bool)sav.Reg[31] && (bool)sav.Reg[35]) ||
((bool)sav.Reg[33] && (bool)sav.Reg[35])) st += ",";
        if((bool)sav.Reg[35]) st +=
(Math.Round(pl.Z, (int)sav.Reg[36])).ToString();

        sav.LabelList[i] = st;
    }

    sav.Reg[4] = sav.postList;

    if((double)sav.Reg[41] == 0 && (double)sav.Reg[42]== 0)
    {
        roundExterms();
        sav.Reg[41]=sav.Mins[0];
        sav.Reg[42]=sav.Maxs[0];
        sav.Reg[51]=sav.Mins[1];
        sav.Reg[52]=sav.Maxs[1];
        sav.Reg[43]=steps(sav.Maxs[0],sav.Mins[0]);
        sav.Reg[53]=steps(sav.Maxs[1],sav.Mins[1]);
    }
    //}
}

```



```

        }
        else
        {
            freq[i] = pl.Z;
            int ci = i;
            for(int j=i-1;j>=0;j--)
            {
if(freq[ci]<freq[j])
{
    double sw = freq[ci];
    freq[ci] = freq[j];
    freq[j] = sw;
    ci = j;
}
else
break;
            }
        }

    }
    if (hystType == 1 || hystType == 3)
    {
        double f=0;
        for(int j=0;j<freq.Length;j++)
        {
            f+=freq[j];
            freq[j] = f;
        }
        maxFreq = sav.postList.Count;
    }
    if (hystType == 2 || hystType == 3)
    {
        for(int j=0;j<freq.Length;j++)
            freq[j] = freq[j]*100/sav.postList.Count;
        maxFreq = maxFreq*100/sav.postList.Count;
    }

    //sort(ref sortList);
    StatisticParam sta = new StatisticParam(sortList);
    int m=0,n=0;

    if((double)sav.Reg[41] != 0 || (double)sav.Reg[42] != 0)

    {
        sav.Mins[0]=(double) sav.Reg[41];
        sav.Maxs[0]=(double) sav.Reg[42];
        sav.Mins[1]=(double) sav.Reg[51];
        sav.Maxs[1]=(double) sav.Reg[52];
    }
    else
    {
        sav.Mins[1] = 0;
        sav.Maxs[1] = Math.Round(maxFreq *1.2);
        if (hystType == 0 || hystType == 1 || hystType == 2 ||
hystType == 3)
        {
            sav.Mins[0] = 0;

```



```

        sav.Maxs[0] = cCount+0.5;
    }
    else if (hystType == 4 || hystType == 5)
    {
        sav.Mins[0] = freq[0];
        sav.Maxs[0]= freq[freq.Length-1];
        this.roundExterms();
        //MessageBox.Show(sav.Mins[0].ToString() + "_"
+sav.Maxs[0].ToString());
    }
}

sav.MarkerList = new Point[cCount];
sav.LabelList = new string[cCount];
sav.ColorList = new Color[cCount];

ArrayList outPut = new ArrayList();
for (int j=1;j<=cCount;j++)
{
    EachPoint p3 = new EachPoint();
    if (hystType != 4 && hystType != 5)
    {
        m = (int)(j*800/(sav.Maxs[0]));
        p3.X = j;

    }
    else if (hystType != 5)
    {
        m = (int)((freq[j-1]-
sav.Mins[0])*800/(sav.Maxs[0]-sav.Mins[0]));
        p3.X = j;
    }
    else
    {
        m = (int)(Math.Log10(freq[j-1]-
sav.Mins[0])*800/Math.Log10(sav.Maxs[0]-sav.Mins[0]));
        p3.X = j;
    }

    if (hystType != 4 && hystType != 5)
    {
        n = (int)(freq[j-1]*600/(sav.Maxs[1]));
        p3.Y = freq[j-1];
    }
    else
    {
        n =
(int)(((this.normalInvers((double)j/freq.Length-0.0001,0,1)-
this.normalInvers(.0001,0,1))/(2*this.normalInvers(.9999,0,1)))*600)
;
        p3.Y = j;
    }
    outPut.Add(p3);

    sav.MarkerList[j-1].X = m;
    sav.MarkerList[j-1].Y = n;
}

```

```

        sav.ColorList[j-1] = (Color)sav.Reg[22];
        if (labelX)
        {
            if (hystType != 4 && hystType != 5)
            {
                double lim = (sav.Maxs[2]-sav.Mins[2])/cCount;
                st = "";
                st = "(";
                st = st + (Math.Round((j-1)*lim+sav.Mins[2],xDec)).ToString();
                st = st + "-";
                st = st + (Math.Round((j)*lim+sav.Mins[2],xDec)).ToString();
                st = st + ")";
            }
            else
            {
                st = "";
                st = st + Math.Round((freq[j-1]),yDec).ToString();
            }
            sav.LabelList[j-1] = st;
        }
        if (labelY)
        {
            st = "";
            if (hystType != 4 && hystType != 5)
            {
                st = st + Math.Round((freq[j-1]),yDec).ToString();
            }
            else
            {
                st = st + Math.Round(j*100/(double)freq.Length,yDec).ToString();
            }
            if (labelX) sav.LabelList[j-1] += "," + st ; else
            sav.LabelList[j-1] = st;
        }
    }
    sav.Reg[4] = outPut;

    if (hystType == 4 || hystType == 5)
    {
        sav.Reg[71] = false;
        sav.Reg[73] = true;
    }
    if ((bool)sav.Reg[74])
    {
        if (hystType != 2 && hystType != 3) sav.CurveList= new
Point[0];
        if (hystType == 3)
        {
            for(int j=freq.Length-1;j>0;j--)
            freq[j]-=freq[j-1];
        }

        if (hystType == 2 || hystType == 3)
        {
            Point[] ps = new Point[freq.Length];
            double fAve=0,stdD=0;
            stdD = stdDev(freq);
            for(int j=0;j<freq.Length;j++)

```

```

fAve+=freq[j]*(j+1)/100;
//MessageBox.Show(fAve.ToString());

sav.CurveList= new Point[freq.Length];

for(int j=0;j<freq.Length;j++)
{
sav.CurveList[j].X = (int)((j+1)*800/(sav.Maxs[0]));
if (hystType == 2) sav.CurveList[j].Y =
(int)(this.normal(j+1, fAve, stdD, false)*60000/(sav.Maxs[1]));
else
{
if (j==0) sav.CurveList[j].Y
=(int)(this.normal(j+1, fAve, stdD, true)*60000/(sav.Maxs[1]));
else sav.CurveList[j].Y
=(int)(this.normal(j+1, fAve, stdD, true)*60000/(sav.Maxs[1])) +
sav.CurveList[j-1].Y ;
}
}
}
if((double)sav.Reg[41] == 0 && (double)sav.Reg[42]== 0)

{
roundExterms();
sav.Reg[41]=sav.Mins[0];
sav.Reg[42]=sav.Maxs[0];
if (hystType == 4 || hystType == 5)
sav.Reg[43]=(sav.Maxs[0]-sav.Mins[0])/(cCount);
else sav.Reg[43]=(double)1;
sav.Reg[51]=sav.Mins[1];
sav.Reg[52]=sav.Maxs[1];
sav.Reg[53]=steps(sav.Maxs[1], sav.Mins[1]);
}
}

```

```

private void build()
{
    int xDec = (int)sav.Reg[32];
    int yDec = (int)sav.Reg[34];
    bool labelX = (bool)sav.Reg[31];
    bool labelY = (bool)sav.Reg[33];
    bool manual = (bool)sav.Reg[76];

    drawVar();

    string st="";
    EachPoint p2 = new EachPoint();
    for(int i=0;i<varList.Count;i++)
    {
        varPoint vP;
        vP = (varPoint)varList[i];
        p2.X = (double)((i+1)*lag);
        p2.Y = vP.Val;
        if (i==0)
        {
            sav.Maxs[0] = p2.X;
            sav.Mins[0] = p2.X;
            sav.Maxs[1] = p2.Y;
            sav.Mins[1] = p2.Y;
        }
        else
        {
            if (p2.X>sav.Maxs[0]) sav.Maxs[0] = p2.X;
            if (p2.X<sav.Mins[0]) sav.Mins[0] = p2.X;
            if (p2.Y>sav.Maxs[1]) sav.Maxs[1] = p2.Y;
            if (p2.Y<sav.Mins[1]) sav.Mins[1] = p2.Y;
        }
    }

    if (sav.Mins[0]>0) sav.Mins[0]=0;
    if (!manual)
    {
        fit();
    }

    if(!(bool)sav.Reg[86])
    {
        #region variogram
        if (sav.Mins[1] > choosedC0) sav.Mins[1] = choosedC0;
        sav.Mins[1] = sav.Mins[1] - (sav.Mins[1]%10);

        if((double)sav.Reg[41] != 0 || (double)sav.Reg[42] != 0)
        {
            sav.Mins[0]=(double)sav.Reg[41];
            sav.Maxs[0]=(double)sav.Reg[42];
            sav.Mins[1]=(double)sav.Reg[51];
            sav.Maxs[1]=(double)sav.Reg[52];
        }
    }
}

```

```

sav.MarkerList = new Point[varList.Count];
sav.ColorList = new Color[varList.Count];
sav.LabelList = new string[varList.Count];

ArrayList outPut = new ArrayList();
for(int i=0;i<varList.Count;i++)
{
    varPoint vP;
    st = "";
    vP = (varPoint)varList[i];
    p2.X = (float)((i+1)*lag);
    p2.Y = vP.Val;

    EachPoint p3 = new EachPoint();
    p3.X = (float)((i+1)*lag);
    p3.Y = vP.Val;
    outPut.Add(p3);

    sav.MarkerList[i].X = (int)((p2.X-
(sav.Mins[0]))*800/((sav.Maxs[0]-sav.Mins[0])));
    sav.MarkerList[i].Y = (int)((p2.Y-
(sav.Mins[1]))*600/((sav.Maxs[1]-sav.Mins[1])));
    sav.ColorList[i] = (Color)sav.Reg[22];

    if (labelX) st = Math.Round(p2.X,xDec).ToString();

    if (labelY && labelX) st += ",";
    if (labelY) st +=
Math.Round(p2.Y,yDec).ToString();
    sav.LabelList[i] = st;
}
sav.Reg[4] = outPut;

double gama;
int ks=0,m=0,n=0;
sav.CurveList = new Point[100];
for (double i=0;i<sav.Maxs[0]+1;i+=sav.Maxs[0]/100)
{
    gama = Gama(choosedC0,choosedC,i,choosedA);
    m = (int)((i-(sav.Mins[0]))*800/((sav.Maxs[0]-
sav.Mins[0])));
    n = (int)((gama-(sav.Mins[1]))*600/((sav.Maxs[1]-
sav.Mins[1])));
    if (ks<100)
    {
sav.CurveList[ks].X = m;
sav.CurveList[ks].Y = n;
    }
    ks++;
}
#endregion
}
else
{

```

```

#region cross valid
double[] arrV = drawCross();
for(int i=1;i<arrV.Length ;i++)
{
    if(i==1)
    {
sav.Mins[0]=arrV[i];
sav.Maxs[0]=arrV[i];
sav.Mins[1]=Gama(choosedC0,choosedC,i*choosedA/100,choosedA);
sav.Maxs[1]=Gama(choosedC0,choosedC,i*choosedA/100,choosedA);
    }
    else
    {
if(sav.Mins[0]>arrV[i]) sav.Mins[0] = arrV[i];
if(sav.Maxs[0]<arrV[i]) sav.Maxs[0] = arrV[i];
if(sav.Mins[1]>Gama(choosedC0,choosedC,i*choosedA/100,choosedA))
sav.Mins[1] = Gama(choosedC0,choosedC,i*choosedA/100,choosedA);
if(sav.Maxs[1]<Gama(choosedC0,choosedC,i*choosedA/100,choosedA))
sav.Maxs[1] = Gama(choosedC0,choosedC,i*choosedA/100,choosedA);

    }
}
sav.Maxs[0]+=(sav.Maxs[0]-sav.Mins[0])/5;
sav.Maxs[1]+=(sav.Maxs[1]-sav.Mins[1])/5;

if((double)sav.Reg[41] != 0 || (double)sav.Reg[42] != 0)
{
    sav.Mins[0]=(double)sav.Reg[41];
    sav.Maxs[0]=(double)sav.Reg[42];
    sav.Mins[1]=(double)sav.Reg[51];
    sav.Maxs[1]=(double)sav.Reg[52];
}
sav.MarkerList = new Point[100];
sav.ColorList = new Color[100];
sav.LabelList = new string[100];

ArrayList outPut = new ArrayList();
double zx=0;
double zy=0;
double zx2=0;
double zxy=0;
for(int i=1;i<arrV.Length ;i++)
{
    p2.Y =
Gama(choosedC0,choosedC,i*choosedA/100,choosedA);
    p2.X = arrV[i];

    zx+=p2.X;
    zy+=p2.Y;
    zx2+=Math.Pow(p2.X, 2);
    zxy+=p2.X*p2.Y;

    EachPoint p3 = new EachPoint();

```

```

        p3.X =
Gama (choosedC0, choosedC, i*choosedA/100, choosedA);
        p3.Y = arrV[i];
        outPut.Add(p3);

        sav.MarkerList[i-1].X = (int)((p2.X-
(sav.Mins[0]))*800/((sav.Maxs[0]-sav.Mins[0])));
        sav.MarkerList[i-1].Y = (int)((p2.Y-
(sav.Mins[1]))*600/((sav.Maxs[1]-sav.Mins[1])));
        sav.ColorList[i-1] = (Color)sav.Reg[22];

        if (labelX) st = Math.Round(p2.X,xDec).ToString();

        if (labelY && labelX) st += ",";
        if (labelY) st +=
Math.Round(p2.Y,yDec).ToString();
        sav.LabelList[i-1] = st;
    }
    sav.Reg[4] = outPut;
    double gama;
    int ks=0,m=0,n=0;
    sav.CurveList = new Point[5];
    for (double i=0;i<sav.Maxs[0];i+=sav.Maxs[0]/5)
    {
        gama = (zx*zxy-zy*zx2)/(Math.Pow(zx,2)-
100*zx2)+i*(zx*zy-100*zxy)/(Math.Pow(zx,2)-100*zx2);
        m = (int)((i-(sav.Mins[0]))*800/((sav.Maxs[0]-
sav.Mins[0])));
        n = (int)((gama-(sav.Mins[1]))*600/((sav.Maxs[1]-
sav.Mins[1])));
        if (ks<5)
        {
sav.CurveList[ks].X = m;
sav.CurveList[ks].Y = n;
        }
        sav.Reg[88] = this.decimals((zx*zxy-
zy*zx2)/(Math.Pow(zx,2)-100*zx2)).ToString() + " + " +
this.decimals((zx*zy-100*zxy)/(Math.Pow(zx,2)-100*zx2)).ToString() +
"h";
        ks++;
    }
    #endregion
}

if((double)sav.Reg[41] == 0 && (double)sav.Reg[42]== 0)
{
    roundExterms();
    sav.Reg[41]=sav.Mins[0];
    sav.Reg[42]=sav.Maxs[0];
    sav.Reg[51]=sav.Mins[1];
    sav.Reg[52]=sav.Maxs[1];
    sav.Reg[43]=Math.Round((sav.Maxs[0]-
sav.Mins[0])/maxLag);
    sav.Reg[53]=steps(sav.Maxs[1],sav.Mins[1]);
}
}

```

```

private void drawVar()
{
    int i,k;
    EachPoint p1,p2;
    varList.Clear();
    for (k=1;k<=maxLag;k++)
    {
        varPoint vP = new varPoint();
        for (i=0;i<sav.postList.Count-1;i++)
        {
            p1 = (EachPoint)sav.postList[i];
            for (int j = i+1;j<sav.postList.Count;j++)
            {
                p2 = (EachPoint)sav.postList[j];
                if (Math.Abs(angle(p1.X,p1.Y,p2.X,p2.Y)-direc)<=dDirec)
                {
                    if (Math.Abs(dist(p1.X,p1.Y,p2.X,p2.Y)-lag*k)<=lag*dLag*k)
                    {
                        vP.Val+= (double)Math.Pow((p2.Z-p1.Z),2);
                        vP.Count++;
                    }
                }
            }
            vP.Val /= (2*vP.Count);
            varList.Add(vP);
            //MessageBox.Show(k.ToString() + ":" +
            vP.Val.ToString() + ":" + vP.Count.ToString());
        }
    }
}

private void fit()
{
    //MessageBox.Show(model);

    int precision = (int)sav.Reg[81];

    varPoint vP = new varPoint();
    double minC0,maxC0,minC,maxC,minA,maxA;
    double dC0,dC,dA,h;
    double func=0,minFunc=0,gama;

    Form1 parentFrm = null;
    if (frm.Text != "Options and Settings")
    {
        parentFrm = (Form1)frm;
        parentFrm.statusBar.Panels[0].Text = "Modelling
Variogram";
    }
}

```



```

#region model0
if (model == 0)
{
    double[] xS = new double[varList.Count];
    double[] yS = new double[varList.Count];
    double zXY=0;
    for (int i=0;i<varList.Count;i++)
    {
        vP = (varPoint)varList[i];
        xS[i] = (double)lag*(i+1);
        yS[i] = vP.Val;
        zXY += xS[i]*yS[i];
    }
    StatisticParam stp1 = new StatisticParam(xS);
    StatisticParam stp2 = new StatisticParam(yS);

    choosedC0 = (stp1.ZX*zXY-
stp1.ZX2*stp2.ZX)/(Math.Pow(stp1.ZX,2) - stp1.N*stp1.ZX2);
    choosedC = (stp1.ZX*stp2.ZX-
stp1.N*zXY)/(Math.Pow(stp1.ZX,2) - stp1.N*stp1.ZX2);
    choosedA = double.NaN;

    stp1=null;
    stp2=null;
    xS = null;
    yS = null;
}
#endregion
#region model1
else if (model == 1)
{
    double[] xS = new double[varList.Count];
    double[] yS = new double[varList.Count];
    double zXY=0;
    for (int i=0;i<varList.Count;i++)
    {
        vP = (varPoint)varList[i];
        xS[i] = Math.Log((double)lag*(i+1),Math.E);
        yS[i] = Math.Log(vP.Val,Math.E);
        zXY += xS[i]*yS[i];
    }

    StatisticParam stp1 = new StatisticParam(xS);
    StatisticParam stp2 = new StatisticParam(yS);

    choosedA = (zXY*stp1.N -
stp1.ZX*stp2.ZX)/(stp1.ZX2*stp1.N-Math.Pow(stp1.ZX,2));
    choosedC = (stp2.ZX - choosedA*stp1.ZX)/stp1.N;
    choosedC = Math.Exp(choosedC);

    minC0 = -sav.Maxs[1];
    maxC0 = sav.Maxs[1];
    for(int k=1;k<=precision;k++)
    {
        dC0 = maxC0-minC0;

```

```

        for(double c0 = minC0;c0<maxC0;c0+=dC0/10)
        {
func = 0;
for (int i=0;i<varList.Count;i++)
{
    vP = (varPoint)varList[i];
    h = (double)lag*(i+1);
    gama = Gama(c0,choosedC,h,choosedA);
    func += Func(vP.Count,gama,vP.Val);
}
if (c0 == minC0 ) minFunc = func;
if (minFunc >= func)
{
    minFunc = func;
    choosedC0 = c0;
}
        }

        minC0 = choosedC0 - dC0/(20);
        maxC0 = choosedC0 + dC0/(20);
        if (frm.Text != "Options and Settings")
parentFrm.DrawProgress(k*100/precision);
    }

    stp1=null;
    stp2=null;
    xS = null;
    yS = null;
}
#endregion
#region model6
else if (model == 6)
{
    double[] xS = new double[varList.Count];
    double[] yS = new double[varList.Count];
    double zXY=0;
    for (int i=0;i<varList.Count;i++)
    {
        vP = (varPoint)varList[i];
        xS[i] = Math.Log((double)lag*(i+1),Math.E);
        yS[i] = vP.Val;
        zXY += xS[i]*yS[i];
    }

    StatisticParam stp1 = new StatisticParam(xS);
    StatisticParam stp2 = new StatisticParam(yS);

    choosedA = double.NaN;
    choosedC = (stp1.N*zXY-
stp2.ZX*stp1.ZX)/(stp1.N*stp1.ZX2-Math.Pow(stp1.ZX,2));
    choosedC0 = (stp2.ZX - choosedC*stp1.ZX)/stp1.N;
    choosedC /= 3;
}

```

```

    }
    #endregion
    #region other models
    else
    {
        minC0 = 0;
        maxC0 = sav.Maxs[1];
        minC = 0;
        maxC = sav.Maxs[1];
        minA = 0;
        maxA = sav.Maxs[0];
        for(int k=1;k<=precision;k++)
        {
            dC0 = maxC0-minC0;
            dC = maxC-minC;
            dA = maxA-minA;

            for(double c0 = minC0;c0<maxC0;c0+=dC0/10)
            {
                for(double c = minC;c<maxC;c+=dC/10)
                for(double a = minA;a<maxA;a+=dA/10)
                {
                    func = 0;
                    for (int i=0;i<varList.Count;i++)
                    {
                        vP = (varPoint)varList[i];
                        h = (double)lag*(i+1);
                        gama = Gama(c0,c,h,a);
                        func += Func(vP.Count,gama,vP.Val);
                    }
                    if (c0 == minC0 ) minFunc = func;
                    if (minFunc >= func)
                    {
                        minFunc = func;
                        choosedC0 = c0;
                        choosedC = c;
                        choosedA = a;
                    }
                }
            }
            minC0 = choosedC0 - dC0/(20);
            maxC0 = choosedC0 + dC0/(20);
            minC = choosedC - dC/(20);
            maxC = choosedC + dC/(20);
            minA = choosedA - dA/(20);
            maxA = choosedA + dA/(20);
            if (frm.Text != "Options and Settings")
parentFrm.DrawProgress(k*100/precision);
        }
    }
    #endregion

    if (frm.Text != "Options and Settings")
    {
        parentFrm.statusBar.Panels[0].Text = "Ready";
        parentFrm.DrawProgress(0);
    }
}

```

```
        parentFrm = null;
    }

    sav.Reg[79] = choosedC0;
    sav.Reg[78] = choosedC;
    sav.Reg[80] = choosedA;

    vP = null;
    //MessageBox.Show(choosedC0.ToString() + ":" +
    choosedC.ToString() + ":" + choosedA.ToString() );
}
```

```

private void build()
{
    if(sav.Mins[0] == sav.Maxs[0] || sav.Mins[1] == sav.Maxs[1])
    {
        MessageBox.Show("Invalid limits has been
entered.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        if((double)sav.Reg[41] != 0 || (double)sav.Reg[42] != 0)
        {
            sav.Mins[0]=(double)sav.Reg[41];
            sav.Maxs[0]=(double)sav.Reg[42];
            sav.Mins[1]=(double)sav.Reg[51];
            sav.Maxs[1]=(double)sav.Reg[52];
        }
        else
        {
            roundExterms();
        }

        model = (int)sav.Reg[81];
        double c0V = (double)sav.Reg[82];
        double cV = (double)sav.Reg[83];
        double aV = (double)sav.Reg[84];

        EachPoint p1 = new EachPoint();
        EachPoint p2 = new EachPoint();
        double[,] cM = new
Double[sav.postList.Count+1,sav.postList.Count+1];

        sav.MarkerList = new Point[sav.postList.Count];
        sav.ColorList = new Color[sav.postList.Count];
        sav.LabelList = new string[sav.postList.Count];

        int m,n;
        Color markCol = (Color)sav.Reg[22];

        ArrayList bac_Post =
(ArrayList)this.sav.postList.Clone();

        // Creating coefficient matrix
        for (int i=0;i<=sav.postList.Count ; i++)
        {
            for (int j=0;j<=sav.postList.Count ; j++)
            {
                if (i==sav.postList.Count && j==sav.postList.Count) cM[i,j] = 0;
                else if (i==sav.postList.Count || j==sav.postList.Count) cM[i,j] =
                1;
                else
                {

```

```

    p1 = (EachPoint)sav.postList[i];
    p2 = (EachPoint)sav.postList[j];
    cM[i, j] =Gama(c0V,cV,dist(p1.X,p1.Y,p2.X,p2.Y),aV);

}

        }

                if (i!=sav.postList.Count)
                {
m = (int)((p1.X-(sav.Mins[0]))*800/((sav.Maxs[0]-sav.Mins[0]));
n = (int)((p1.Y-(sav.Mins[1]))*600/((sav.Maxs[1]-sav.Mins[1]));

sav.MarkerList[i].X = m;
sav.MarkerList[i].Y = n;
sav.ColorList[i] = markCol;

string st = "";
if((bool)sav.Reg[31]) st =
(Math.Round(p1.X, (int)sav.Reg[32])).ToString();
if((bool)sav.Reg[31] && (bool)sav.Reg[33]) st += ",";
if((bool)sav.Reg[33]) st +=
(Math.Round(p1.Y, (int)sav.Reg[34])).ToString();
if(((bool)sav.Reg[31] && (bool)sav.Reg[35]) || ((bool)sav.Reg[33] &&
(bool)sav.Reg[35])) st += ",";
if((bool)sav.Reg[35]) st +=
(Math.Round(p1.Z, (int)sav.Reg[36])).ToString();

sav.LabelList[i] = st;
        }
}
//MessageBox.Show(str);

// Creating answer matrix and main grid
// Creating kriging coefficient matrix
grdX = (int)sav.Reg[85];
grdY = (int)sav.Reg[86];
int[, ,] blocks = new
int[grdX+1,grdY+1,sav.postList.Count+1];
double[] ansM = new double[sav.postList.Count+1];
double[] p_AnsM = new double[sav.postList.Count+1];
double[,] nCM = new
double[sav.postList.Count+1,sav.postList.Count+1];

int ea = 0,no = 0;
int l1 = 0,l2 = 0;

double rx = (double)sav.Reg[78];
double ry = (double)sav.Reg[77];
double ang = (double)sav.Reg[79]*Math.PI/180;

GridMatrix = new double[grdX+1,grdY+1];

```

```

        for(int i=0;i<=grdX;i++)
            for(int j=0;j<=grdY;j++)
for(int k=0;k<=sav.postList.Count;k++)
    blocks[i,j,k] = -1;

    Form1 f = (Form1)frm;
    if (sav.Acror=="GRD") f.statusBar.Panels[0].Text =
"Block Recognition";
    else f.statusBar.Panels[0].Text = "Initial Model";
    long tr = System.DateTime.Now.Ticks;

    //block recognition
    //string str = "";
    for (double x=sav.Mins[0]; x<=sav.Maxs[0];
x+=(sav.Maxs[0]-sav.Mins[0])/grdX)
    {
        for (double y=sav.Mins[1]; y<=sav.Maxs[1];
y+=(sav.Maxs[1]-sav.Mins[1])/grdY)
        {
for (int k = 0; k<sav.postList.Count; k++)
    {
        p1 = (EachPoint)sav.postList[k];
        if (Math.Pow((x-p1.X)*Math.Cos(ang)-(y-
p1.Y)*Math.Sin(ang),2)/Math.Pow(rx,2) + Math.Pow((x-
p1.X)*Math.Sin(ang)+(y-p1.Y)*Math.Cos(ang),2)/Math.Pow(ry,2)<1)

            {
blocks[ea,no,l1] = k;
//str+=ea.ToString() + "," + no.ToString() + "," + l1.ToString() +
":" + blocks[ea,no,l1].ToString() + "\n";
l1++;
            }
        }
//MessageBox.Show(str);
if (!(bool)sav.Reg[76])
    {
        for(int i=0;i<l1-1;i++)
for(int j=i;j<l1;j++)
    {
        p1 = (EachPoint)sav.postList[blocks[ea,no,i]];
        p2 = (EachPoint)sav.postList[blocks[ea,no,j]];
        if((x-p1.X)*(x-p1.X)+(y-p1.Y)*(y-p1.Y)>(x-p2.X)*(x-p2.X)+(y-
p2.Y)*(y-p2.Y))
        {
            int swap = blocks[ea,no,i];
            blocks[ea,no,i] = blocks[ea,no,j];
            blocks[ea,no,j] = swap;
        }
    }
}

int oct = (int)sav.Reg[88];
int mxP = (int)sav.Reg[89];
int cP = 0;

```

```

        for (int i=0;i<oct;i++)
        {
l1=0;
cP=0;
double tet1 = (double)sav.Reg[79] + i*360/oct;
double tet2 = (double)sav.Reg[79] + (i+1)*360/oct;
if (tet1>=360) tet1-=360;
if (tet2>360) tet2-=360;
while (blocks[ea,no,l1] != -1 && l1<sav.postList.Count)
{
    if(blocks[ea,no,l1] != -2)
    {
        p1 = (EachPoint)sav.postList[blocks[ea,no,l1]];

        if (tet1<tet2)
        {
            double d = angle(x,y,p1.X,p1.Y);
            if (angle(x,y,p1.X,p1.Y)>=tet1 &&
                angle(x,y,p1.X,p1.Y)<tet2)
            {
                cP++;
                if (cP>mxP) blocks[ea,no,l1] = -2;
            }
        }
        else
        {
            if (angle(x,y,p1.X,p1.Y)>tet1 ||
                angle(x,y,p1.X,p1.Y)<=tet2)
            {
                cP++;
                if (cP>mxP) blocks[ea,no,l1] = -2;
            }
        }
    }
    l1++;
}
}
}

l1=0;
int fu = sav.postList.Count;
while (blocks[ea,no,l1] != -1 && l1<fu)
{
    if(blocks[ea,no,l1] == -2)
    {
        for(int i=l1;i<sav.postList.Count-1;i++)
            blocks[ea,no,i] = blocks[ea,no,i+1];
        blocks[ea,no,sav.postList.Count-1]=-2;
        fu--;
    }
    else
    {
        l1++;
    }
}

l1 = 0;

```



```

no++;
    }
    no = 0;
    ea ++;
    f.DrawProgress(ea*100/(grdX+1));
}
ea = 0;
l1 = 0;
string str = "";
f.DrawProgress(ea*100/(grdX+1));
if (sav.Acrr=="GRD") f.statusBar.Panels[0].Text =
"Kriging Coefficients";
else f.statusBar.Panels[0].Text = "Initial Model";
double mx = sav.Maxs[2];
double mn = sav.Mins[2];
for (double x=sav.Mins[0];x<sav.Maxs[0] ;
x+=(sav.Maxs[0]-sav.Mins[0])/grdX)
{
    for (double y=sav.Mins[1];y<sav.Maxs[1] ;
y+=(sav.Maxs[1]-sav.Mins[1])/grdY)
    {
l1 = 0;
ansM = new double[sav.postList.Count+1];
ncM = new double[sav.postList.Count+1,sav.postList.Count+1];

while(blocks[ea,no,l1]>-1)
{
    l2=0;
    while(blocks[ea,no,l2]>-1)
    {
ncM[l1,l2] = cM[blocks[ea,no,l1],blocks[ea,no,l2]];
l2++;
    }
    ncM[l1,l2] = 1;
    pl = (EachPoint) sav.postList[blocks[ea,no,l1]];
    ansM[l1] = Gama(c0V,cV,dist(pl.X,pl.Y,x,y),aV);
    p_AnsM[l1] = Gama(c0V,cV,dist(pl.X,pl.Y,x,y),aV);

    l1++;
}
for(int i=0;i<l1;i++)
    ncM[l1,i] = 1;
ncM[l1,l1] = 0;
ansM[l1] = 1;
p_AnsM[l1] = 1;

det(ncM,ansM,l1+1);

//estimated values or variance
for (int k=0;k<=l1;k++)
{
    if ((bool)sav.Reg[96])
    {
if (blocks[ea,no,k]!=-1 && k!=0) GridMatrix[ea,no] +=
p_AnsM[k]*ansM[k];

```

```

        }
        else
        {
if (k!=11 || k==0)
{
    if (blocks[ea,no,k]==-1) GridMatrix[ea,no] = double.NaN;
    else
    {
        pl = (EachPoint)sav.postList[blocks[ea,no,k]];
        GridMatrix[ea,no] += pl.Z*ansM[k];
    }
}
}
}
if(!double.IsNaN(GridMatrix[ea,no]))
{
    if(!(bool)sav.Reg[96])
    {
if(GridMatrix[ea,no]>mx) GridMatrix[ea,no]=mx;
if(GridMatrix[ea,no]<mn) GridMatrix[ea,no]=mn;
    }

    if (ea==0 && no==0)
    {
sav.Maxs[2] = GridMatrix[ea,no];
sav.Mins[2] = GridMatrix[ea,no];
    }
    else
    {
if (GridMatrix[ea,no]>sav.Maxs[2]) sav.Maxs[2] = GridMatrix[ea,no];

if (GridMatrix[ea,no]<sav.Mins[2]) sav.Mins[2] = GridMatrix[ea,no];

    }
}

no++;
str = "";

        }
        no=0;
        ea++;
        f.DrawProgress(ea*100/(grdX+1));
    }
    ea=0;

    tr = System.DateTime.Now.Ticks - tr;
    tr/=10000;
    if(sav.Acror=="GRD") f.statusBar.Panels[2].Text =
"Estimate in " + ((double)tr/1000).ToString() + " Seconds.";

    f.DrawProgress(ea*100/(grdX+1));
    if(sav.Acror=="GRD") f.statusBar.Panels[0].Text =
"Ready";
    f = null;
    cM = null;

```

```

        ansM = null;
        p_AnsM = null;
        ncM = null;
        blocks = null;

        // backtransformation
        if (this.Logic && (bool)sav.Reg[92])
        {
            for (int x=0;x<grdX ; x++)
for (int y=0;y<grdY ; y++)
            if(!double.IsNaN(GridMatrix[x,y]))
            {
GridMatrix[x,y] = Math.Pow(Base,GridMatrix[x,y]+Ave);
            }
                for(int i=0;i<bac_Post.Count;i++)
this.sav.postList[i] = bac_Post[i];
            }

            sav.Maxs[2] = double.NaN;
            sav.Mins[2] = double.NaN;
            for (int x=0;x<grdX ; x++)
                for (int y=0;y<grdY ; y++)
if(!double.IsNaN(GridMatrix[x,y]))
{
    if (GridMatrix[x,y]>sav.Maxs[2] || double.IsNaN(sav.Maxs[2]))
sav.Maxs[2] = GridMatrix[x,y];
    if (GridMatrix[x,y]<sav.Mins[2] || double.IsNaN(sav.Mins[2]))
sav.Mins[2] = GridMatrix[x,y];
}

            ArrayList outPut = new ArrayList();
            for (int x=0;x<=grdX ; x++)
                for (int y=0;y<=grdY ; y++)
                {
EachPoint p3 = new EachPoint();
p3.X = x*(sav.Maxs[0]-sav.Mins[0])/grdX+(sav.Mins[0])+(sav.Maxs[0]-
sav.Mins[0])/grdX);
p3.Y = y*(sav.Maxs[1]-sav.Mins[1])/grdY+(sav.Mins[1])+(sav.Maxs[1]-
sav.Mins[1])/grdY);
p3.Z = GridMatrix[x,y];
if (p3.Z < sav.Mins[2]) p3.Z = double.NaN;
outPut.Add(p3);
                }
            sav.Reg[4] = outPut;

            if((double)sav.Reg[41] == 0 && (double)sav.Reg[42]== 0)
            {
                sav.Reg[41]=sav.Mins[0];
                sav.Reg[42]=sav.Maxs[0];
                sav.Reg[51]=sav.Mins[1];
                sav.Reg[52]=sav.Maxs[1];
                sav.Reg[43]=steps(sav.Maxs[0],sav.Mins[0]);
                sav.Reg[53]=steps(sav.Maxs[1],sav.Mins[1]);
            }
        }
    }
}

```

```

private void build()
{
    if(sav.Mins[0] == sav.Maxs[0] || sav.Mins[1] == sav.Maxs[1])

    {
        MessageBox.Show("Invalid limits has been
entered.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        if((double)sav.Reg[41] != 0 || (double)sav.Reg[42] != 0)
        {
            sav.Mins[0]=(double)sav.Reg[41];
            sav.Maxs[0]=(double)sav.Reg[42];
            sav.Mins[1]=(double)sav.Reg[51];
            sav.Maxs[1]=(double)sav.Reg[52];
        }
        else
        {
            roundExterms();
        }

        model = (int)sav.Reg[81];
        double c0V = (double)sav.Reg[82];
        double cV = (double)sav.Reg[83];
        double aV = (double)sav.Reg[84];
        grdX = (int)sav.Reg[85];
        grdY = (int)sav.Reg[86];
        Color markCol = (Color)sav.Reg[22];
        double rx = (double)sav.Reg[78];
        double ry = (double)sav.Reg[77];
        double ang = (double)sav.Reg[79]*Math.PI/180;

        EachPoint p1 = new EachPoint();
        EachPoint p2 = new EachPoint();
        ArrayList s_Post = new ArrayList();
        double[,] cM;
        double[] ansM;
        double[] p_AnsM;
        double[,] ncM = new
double[sav.postList.Count+1, sav.postList.Count+1];
        GridMatrix = new double[grdX+1,grdY+1];
        Point[] eas = new Point[(grdX+1)*(grdY+1)];
        int m=0,n=0;
        int ea = 0,no = 0;
        int oct = (int)sav.Reg[88];
        int mxP = (int)sav.Reg[89];

        sav.MarkerList = new Point[sav.postList.Count];
        sav.ColorList = new Color[sav.postList.Count];
        sav.LabelList = new string[sav.postList.Count];
    }
}

```

```

        ArrayList bac_Post =
(ArrayList)this.sav.postList.Clone();

        #region to fit data to grid
        for (int x=0;x<=grdX ; x++)
            for (int y=0;y<=grdY ; y++)
GridMatrix[x,y] = double.NaN;

        int[,] countMatrix = new int[grdX+1,grdY+1];
        for (int i=0;i<sav.postList.Count ; i++)
        {
            pl = (EachPoint)sav.postList[i];
            ea = (int)((pl.X-
sav.Mins[0])*(grdX))/(sav.Maxs[0]-sav.Mins[0])+.5);
            no = (int)((pl.Y-
sav.Mins[1])*(grdY))/(sav.Maxs[1]-sav.Mins[1])+.5);
            if (double.IsNaN(GridMatrix[ea,no]))
            {
GridMatrix[ea,no] = pl.Z;
countMatrix[ea,no] = 1;
            }
            else
            {
GridMatrix[ea,no] += pl.Z;
countMatrix[ea,no] ++;
            }

            if (i!=sav.postList.Count)
            {
m = (int)((pl.X-(sav.Mins[0]))*800/((sav.Maxs[0]-sav.Mins[0])));
n = (int)((pl.Y-(sav.Mins[1]))*600/((sav.Maxs[1]-sav.Mins[1])));

sav.MarkerList[i].X = m;
sav.MarkerList[i].Y = n;
sav.ColorList[i] = markCol;

string st = "";
if((bool)sav.Reg[31]) st =
(Math.Round(pl.X, (int)sav.Reg[32])).ToString();
if((bool)sav.Reg[31] && (bool)sav.Reg[33]) st += ",";
if((bool)sav.Reg[33]) st +=
(Math.Round(pl.Y, (int)sav.Reg[34])).ToString();
if(((bool)sav.Reg[31] && (bool)sav.Reg[35]) || ((bool)sav.Reg[33] &&
(bool)sav.Reg[35])) st += ",";
if((bool)sav.Reg[35]) st +=
(Math.Round(pl.Z, (int)sav.Reg[36])).ToString();

sav.LabelList[i] = st;
            }
        }

        for (int x=0;x<=grdX ; x++)
            for (int y=0;y<=grdY ; y++)

```

```

GridMatrix[x,y] /= (double)countMatrix[x,y];

    ea = 0;no = 0;
    #endregion

    #region blending grid

    Random rand = new Random();
    for (int j=0; j<=grdY; j++)
        for (int i=0;i<=grdX; i++)
eas[j*(grdX+1)+i] = new Point(i,j);
    for (int i=0;i<eas.Length-1 ; i++)
    {
        int rn = rand.Next(i+1,eas.Length);
        int swap = eas[i].X;
        eas[i].X = eas[rn].X;
        eas[rn].X = swap;

        swap = eas[i].Y;
        eas[i].Y = eas[rn].Y;
        eas[rn].Y = swap;
    }
    #endregion

    no = 1;
    double x0 = 0;
    double y0 = 0;
    double r = 0;
    double b = 0;
    double xx = 0;
    double yy = 0;
    double dxx = 0;
    double dyy = 0;
    int count = 0;
    int[] o_Count = new int[oct];
    int o_Ind = 0;

    Form1 f = (Form1)frm;
    long tr = System.DateTime.Now.Ticks;

    f.statusBar.Panels[0].Text = "Wait...";

    for (int rout=0;rout<=1;rout++)
    for (int ii=0;ii<eas.Length ; ii++)
    {
        ea = eas[ii].X;
        no = eas[ii].Y;
        if (double.IsNaN(GridMatrix[ea,no]))
        {
#region point simulation
// Search for in range posts
x0 = sav.Mins[0]+((sav.Maxs[0]-sav.Mins[0])/(double)grdX)*ea;
y0 = sav.Mins[1]+((sav.Maxs[1]-sav.Mins[1])/(double)grdY)*no;
r = 0;

```

```

b = 0;
xx = 0;
yy = 0;
dxx = 0;
dyy = 0;
count = 0;
o_Count = new int[oct];
o_Ind = 0;
s_Post = new ArrayList();

for (double a=rx/(double)grdX;a<=rx;a+=rx/(double)grdX)
{
    b = a*ry/rx;
    count ++;
    for(double
t=0;t<Math.PI*2;t+=(Math.Atan((double)grdX/(double)grdY))/count)

    {
o_Ind = (int)Math.Floor(oct*t/(Math.PI*2));
if(o_Count[o_Ind]>=mxP)
    break;

p1 = new EachPoint();
r =
a*b/Math.Sqrt(Math.Pow(a*Math.Sin(t),2)+Math.Pow(b*Math.Cos(t),2));

dxx = r*Math.Cos(t);
dyy = r*Math.Sin(t);
xx = dxx*Math.Cos(ang)-dyy*Math.Sin(ang);
yy = dyy*Math.Cos(Math.PI+ang)+dxx*Math.Sin(Math.PI+ang);
xx = xx+((sav.Maxs[0]-sav.Mins[0])/(double)grdX)*ea;
yy = yy+((sav.Maxs[1]-sav.Mins[1])/(double)grdY)*no;
xx = xx*grdX/(sav.Maxs[0]-sav.Mins[0]);
yy = yy*grdY/(sav.Maxs[1]-sav.Mins[1]);
m = (int)xx;
n = (int)yy;
if (m>=0 && n>=0 && m<=grdX && n<=grdY)
    if (!double.IsNaN(GridMatrix[m,n]))
    {
        p1.X = m*((sav.Maxs[0]-
sav.Mins[0])/(double)grdX)+sav.Mins[0];
        p1.Y = n*((sav.Maxs[1]-
sav.Mins[1])/(double)grdY)+sav.Mins[1];
        p1.Z = GridMatrix[m,n];
        s_Post.Add(p1);
        GridMatrix[m,n] = double.NaN;
        o_Count[o_Ind]++;
    }
    }
    if (s_Post.Count>=mxP*oct)
    {
break;
    }
}

//restore main values

```

```

for (int i=0;i<s_Post.Count ; i++)
{
    p1 = (EachPoint)s_Post[i];
    m = (int) (((p1.X-sav.Mins[0])*grdX)/(sav.Maxs[0]-sav.Mins[0])+.5);
    n = (int) (((p1.Y-sav.Mins[1])*grdY)/(sav.Maxs[1]-sav.Mins[1])+.5);
    GridMatrix[m,n] = p1.Z;
}

//to create coefficients matrix
ansM = new double[s_Post.Count+1];
p_AnsM = new double[s_Post.Count+1];
cM = new double[s_Post.Count+1,s_Post.Count+1];

for (int i=0;i<=s_Post.Count ; i++)
{
    for (int j=0;j<=s_Post.Count ; j++)
    {
        if (i==s_Post.Count && j==s_Post.Count) cM[i,j] = 0;
        else if (i==s_Post.Count || j==s_Post.Count) cM[i,j] = 1;
        else
        {
            p1 = (EachPoint)s_Post[i];
            p2 = (EachPoint)s_Post[j];
            cM[i,j] =Gama(c0V,cV,dist(p1.X,p1.Y,p2.X,p2.Y),aV);
        }
        if (i==s_Post.Count)
        {
            ansM[i] = 1;
            p_AnsM[i] = 1;
        }
        else
        {
            ansM[i] = Gama(c0V,cV,dist(p1.X,p1.Y,x0,y0),aV);
            p_AnsM[i] = Gama(c0V,cV,dist(p1.X,p1.Y,x0,y0),aV);
        }
    }
}
det(cM,ansM,s_Post.Count+1);

// Calculate value and variance
xx=0;
for (int k=0;k<=s_Post.Count;k++)
{
    xx += p_AnsM[k]*ansM[k];
    if (k!=s_Post.Count)
    {
        p1 = (EachPoint)s_Post[k];
        //str+=Math.Round(ansM[k],3).ToString()+"\n";
        if (double.IsNaN(GridMatrix[ea,no])) GridMatrix[ea,no] = 0;
        GridMatrix[ea,no] += p1.Z*ansM[k];
    }
}

```



```

xx = Math.Sqrt(xx);
//MessageBox.Show(xx.ToString());
Random rnd = new Random();
if (!double.IsNaN(GridMatrix[ea,no]))
{
    //yy =
(double) frmu.NORMINV(rnd.NextDouble(),GridMatrix[ea,no],xx);

    yy = this.normalInvers(rnd.NextDouble(),GridMatrix[ea,no],xx);

    //Change out of limit values
    if (yy>sav.Maxs[2] || yy<sav.Mins[2])
    {
int coX = 0;
while (yy>sav.Maxs[2] || yy<sav.Mins[2] || coX<10)
{
    //yy =
(double) frmu.NORMINV(rnd.NextDouble(),GridMatrix[ea,no],xx);
    yy = this.normalInvers(rnd.NextDouble(),GridMatrix[ea,no],xx);

    coX++;
}
    }
    GridMatrix[ea,no] = yy;
}
# endregion

if (GridMatrix[ea,no]>sav.Maxs[2]) GridMatrix[ea,no]=sav.Maxs[2];
else if (GridMatrix[ea,no]<sav.Mins[2])
GridMatrix[ea,no]=sav.Mins[2];
    }
    f.DrawProgress(ii*100/(eas.Length));
}
tr = System.DateTime.Now.Ticks - tr;
tr/=10000;
f.statusBar.Panels[2].Text = "Simulate in " +
((double)tr/1000).ToString() + " Seconds.";

    f.DrawProgress(0*100/(grdX+1));
f.statusBar.Panels[0].Text = "Ready";
f = null;
cM = null;
ansM = null;
p_AnsM = null;
s_Post = null;
eas = null;

    // backtransformation
    if (this.Logic && (bool)sav.Reg[92])
    {
        for (int x=0;x<=grdX ; x++)
for (int y=0;y<=grdY ; y++)

```

```

GridMatrix[x,y] = Math.Pow(Base,GridMatrix[x,y]+Ave);

        sav.Maxs[2] = Math.Pow(Base,sav.Maxs[2]+Ave);
        sav.Mins[2] = Math.Pow(Base,sav.Mins[2]+Ave);

        for(int i=0;i<bac_Post.Count;i++)
this.sav.postList[i] = bac_Post[i];
        }

        ArrayList outPut = new ArrayList();
        for (int x=0;x<=grdX ; x++)
            for (int y=0;y<=grdY ; y++)
            {
EachPoint p3 = new EachPoint();
p3.X = x*(sav.Maxs[0]-sav.Mins[0])/grdX+(sav.Mins[0])+(sav.Maxs[0]-
sav.Mins[0])/(grdX);
p3.Y = y*(sav.Maxs[1]-sav.Mins[1])/grdY+(sav.Mins[1])+(sav.Maxs[1]-
sav.Mins[1])/(grdY);
p3.Z = GridMatrix[x,y];
if (p3.Z < sav.Mins[2]) p3.Z = double.NaN;
outPut.Add(p3);
            }
        sav.Reg[4] = outPut;

        if((double)sav.Reg[41] == 0 && (double)sav.Reg[42]== 0)
        {
            sav.Reg[41]=sav.Mins[0];
            sav.Reg[42]=sav.Maxs[0];
            sav.Reg[51]=sav.Mins[1];
            sav.Reg[52]=sav.Maxs[1];
            sav.Reg[43]=steps(sav.Maxs[0],sav.Mins[0]);
            sav.Reg[53]=steps(sav.Maxs[1],sav.Mins[1]);
        }
    }
}

```

```
private void build()
{
    if(sav.Mins[0] == sav.Maxs[0] || sav.Mins[1] == sav.Maxs[1])

        {
            MessageBox.Show("Invalid limits has been
entered.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    else
    {
        long tr = System.DateTime.Now.Ticks;

        grdX = (int)sav.Reg[85];
        grdY = (int)sav.Reg[86];
        Color markCol = (Color)sav.Reg[22];

        double[,] grd_Ini = new double[grdX+1,grdY+1];

        double cte1=0,cte2=0;
        int ea=0,no=0;
        int m=0,n=0;
        EachPoint p1 = new EachPoint();

        int model = (int)sav.Reg[81];
        double c0 = (double)sav.Reg[82];

        double c = (double)sav.Reg[83];

        double a = (double)sav.Reg[84];

        double lag = (double)sav.Reg[69];

        //a*=1.5;

        GridMatrix = new double[grdX+1,grdY+1];

        sav.MarkerList = new Point[sav.postList.Count];
        sav.ColorList = new Color[sav.postList.Count];
        sav.LabelList = new string[sav.postList.Count];
        ArrayList bac_Post =
(ArrayList)this.sav.postList.Clone();
        double[] varIniArr = this.drawIniVar(lag,a);
        string s = "";
        for(int i=0;i<varIniArr.Length;i++)
            s+=varIniArr[i].ToString()+"\n";
    }
}
```

```

        MessageBox.Show(s);
#region //1 to fit data to grid

for(int x=0;x<=grdX;x++)
    for(int y=0;y<=grdY;y++)
    {
        GridMatrix[x,y] = double.NaN;
        grd_Ini[x,y] = double.NaN;
    }

int[,] countMatrix = new int[grdX+1,grdY+1];
for (int i=0;i<sav.postList.Count ; i++)
{
    pl = (EachPoint)sav.postList[i];

    ea = (int)((pl.X-
sav.Mins[0])* (grdX))/(sav.Maxs[0]-sav.Mins[0])+.5);

    no = (int)((pl.Y-
sav.Mins[1])* (grdY))/(sav.Maxs[1]-sav.Mins[1])+.5);

    if (double.IsNaN(GridMatrix[ea,no]))

    {
        GridMatrix[ea,no] = pl.Z;
        countMatrix[ea,no] = 1;
    }
    else
    {
        GridMatrix[ea,no] += pl.Z;
        countMatrix[ea,no] ++;
    }

    if (i!=sav.postList.Count)
    {
        m = (int)((pl.X-
(sav.Mins[0]))*800/((sav.Maxs[0]-sav.Mins[0])));
        n = (int)((pl.Y-
(sav.Mins[1]))*600/((sav.Maxs[1]-sav.Mins[1])));

        sav.MarkerList[i].X = m;
        sav.MarkerList[i].Y = n;
        sav.ColorList[i] = markCol;
        string st = "";
        if((bool)sav.Reg[31]) st =
(Math.Round(pl.X, (int)sav.Reg[32])).ToString();
        if((bool)sav.Reg[31] && (bool)sav.Reg[33])
st += ",";
        if((bool)sav.Reg[33]) st +=
(Math.Round(pl.Y, (int)sav.Reg[34])).ToString();
        if((bool)sav.Reg[31] && (bool)sav.Reg[35])
|| ((bool)sav.Reg[33] && (bool)sav.Reg[35])) st += ",";
        if((bool)sav.Reg[35]) st +=
(Math.Round(pl.Z, (int)sav.Reg[36])).ToString();

```

```

        sav.LabelList[i] = st;
    }
}

for (int x=0;x<=grdX ; x++)
    for (int y=0;y<=grdY ; y++)
        GridMatrix[x,y] /= (double)countMatrix[x,y];
ea = 0;no = 0;
#endregion

#region //2 cdf function
ArrayList cdf = new ArrayList(11);
this.cdf(cdf);
#endregion

#region//3 specify random values

Random rnd = new Random();
for(int x=0;x<=grdX;x++)
    for(int y=0;y<=grdY;y++)
    {
        if(double.IsNaN(GridMatrix[x,y]))

            {
                double val =
this.CDFINV(rnd.NextDouble(),cdf);
                if (val>sav.Maxs[2] ||
val<sav.Mins[2])
                    {
                        int coX = 0;
                        while (val>sav.Maxs[2] ||
val<sav.Mins[2] || coX<10)
                            {
                                checked{val =
this.CDFINV(rnd.NextDouble(),cdf);}
                                coX++;
                            }
                        GridMatrix[x,y]=val;
                        if (GridMatrix[ea,no]>sav.Maxs[2])
GridMatrix[ea,no]=sav.Maxs[2];
                        else if
(GridMatrix[ea,no]<sav.Mins[2]) GridMatrix[ea,no]=sav.Mins[2];
                    }
                else
                {
                    {
                        grd_Ini[x,y] = GridMatrix[x,y];
                    }
                }
            }
#endregion

#region//5 ctel
ctel = double.MaxValue;
#endregion
Form1 f = (Form1)frm;

```

```

f.statusBar.Panels[0].Text = "Wait...";
int tim = (int)sav.Reg[68];
double t1 = (double)((decimal)sav.Reg[90]);
double iniT = (double)((decimal)sav.Reg[90]);
double t2 = (double)((decimal)sav.Reg[91]);
double cool = (double)((decimal)sav.Reg[92]);
while (t2+.1<t1)
{
    for (int routs=0;routs<tim;routs++)

    {
        #region//6 swap points
        double swap = 0;
        int real = rnd.Next(0,grdX+1);
        int rno1 = rnd.Next(0,grdY+1);
        while(!double.IsNaN(grd_Ini[real,rno1]))
        {
            real = rnd.Next(0,grdX+1);
            rno1 = rnd.Next(0,grdY+1);
        }
        int rea2 = rnd.Next(0,grdX+1);
        int rno2 = rnd.Next(0,grdY+1);
        while(!double.IsNaN(grd_Ini[rea2,rno2]))
        {
            rea2 = rnd.Next(0,grdX+1);
            rno2 = rnd.Next(0,grdY+1);
        }
        swap = GridMatrix[real,rno1];
        GridMatrix[real,rno1] =
GridMatrix[rea2,rno2];
        GridMatrix[rea2,rno2] = swap;
        #endregion
        #region//7 secondary model
        double[] varArr =
this.drawVar(lag,a,(sav.Maxs[0]-sav.Mins[0])/grdX,(sav.Maxs[1]-
sav.Mins[1])/grdY);
        #endregion
        if(routs==tim-1)
        {
            string st = "";
            for(int i=0;i<varArr.Length;i++)
                st+=varArr[i].ToString()+"\n";
            MessageBox.Show(st);
        }
        #region//8 cte2
        cte2=0;
        for(int i=1;i<varArr.Length;i++)
        {
            cte2+=Math.Pow((varArr[i]-
varIniArr[i])/varIniArr[i],2);
        }
        cte2 /= varArr.Length;
        #endregion

        #region//9 decision
        if(cte2<=ctel)
        {
            ctel = cte2;

```

```

        }
        else
        {
            double pres = 0;
            //refuse precision
            pres = Math.Exp(1000*(cte1-
cte2)/(t1));
            if (cte1==double.MaxValue ||
rnd.NextDouble()>pres)
            {
                swap = GridMatrix[real,rno1];
                GridMatrix[real,rno1] =
GridMatrix[rea2,rno2];
                GridMatrix[rea2,rno2] = swap;
            }
            else
            {
                cte1 = cte2;
            }
        }
        #endregion
    }
    t1*=cool;
    f.DrawProgress(100-(int)((t1-t2)*100/(iniT-t2)));
}
f.DrawProgress(0);
f.statusBar.Panels[0].Text = "Ready";
tr = System.DateTime.Now.Ticks - tr;
tr/=10000;
f.statusBar.Panels[2].Text = "Simulate in " +
((double)tr/1000).ToString() + " Seconds.";
//back transformation
if ((bool)sav.Reg[67])
{
    for (int x=0;x<=grdX ; x++)
        for (int y=0;y<=grdY ; y++)
            GridMatrix[x,y]=Math.Pow(10,GridMatrix[x,y]);
    sav.Maxs[2] = Math.Pow(10,sav.Maxs[2]);
    sav.Mins[2] = Math.Pow(10,sav.Mins[2]);
    for(int i=0;i<bac_Post.Count;i++)
        this.sav.postList[i] = bac_Post[i];
}
ArrayList outPut = new ArrayList();
for (int x=0;x<=grdX ; x++)
    for (int y=0;y<=grdY ; y++)
    {
        EachPoint p3 = new EachPoint();
        p3.X = x*(sav.Maxs[0]-
sav.Mins[0])/grdX+(sav.Mins[0])+(sav.Maxs[0]-sav.Mins[0])/(grdX);
        p3.Y = y*(sav.Maxs[1]-
sav.Mins[1])/grdY+(sav.Mins[1])+(sav.Maxs[1]-sav.Mins[1])/(grdY);
        p3.Z = GridMatrix[x,y];
        if (p3.Z < sav.Mins[2]) p3.Z = double.NaN;
        outPut.Add(p3);
    }
sav.Reg[4] = outPut;
if((double)sav.Reg[41] == 0 && (double)sav.Reg[42]== 0)

```

```
{
    sav.Reg[41]=sav.Mins[0];
    sav.Reg[42]=sav.Maxs[0];
    sav.Reg[51]=sav.Mins[1];
    sav.Reg[52]=sav.Maxs[1];
    sav.Reg[43]=steps(sav.Maxs[0],sav.Mins[0]);
    sav.Reg[53]=steps(sav.Maxs[1],sav.Mins[1]);
}
}
```



```

private void build()
{

    if(sav.Mins[0] == sav.Maxs[0] || sav.Mins[1] == sav.Maxs[1])

    {
        sav.Maxs[0]++;
        sav.Maxs[1]++;
        MessageBox.Show("Invalid limits has been
entered.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        if((double)sav.Reg[41] != 0 || (double)sav.Reg[42] != 0)
        {
            sav.Mins[0]=(double)sav.Reg[41];
            sav.Maxs[0]=(double)sav.Reg[42];
            sav.Mins[1]=(double)sav.Reg[51];
            sav.Maxs[1]=(double)sav.Reg[52];
        }
        else
        {
            roundExterms();
        }

        sav.MarkerList = new Point[sav.postList.Count];
        sav.ColorList = new Color[sav.postList.Count];
        sav.LabelList = new string[sav.postList.Count];

        int m,n;
        EachPoint p1;
        ArrayList point = new ArrayList();
        for (int i=0;i<sav.postList.Count;i++)
        {
            p1 = (EachPoint)sav.postList[i] ;
            m = (int)((p1.X-(sav.Mins[0]))*800/((sav.Maxs[0]-
sav.Mins[0])));
            n = (int)((p1.Y-(sav.Mins[1]))*600/((sav.Maxs[1]-
sav.Mins[1])));

            sav.MarkerList[i].X = m;
            sav.MarkerList[i].Y = n;

            string st = "";
            if((bool)sav.Reg[31]) st =
(Math.Round(p1.X, (int)sav.Reg[32])).ToString();
            if((bool)sav.Reg[31] && (bool)sav.Reg[33]) st +=
", ";
            if((bool)sav.Reg[33]) st +=
(Math.Round(p1.Y, (int)sav.Reg[34])).ToString();
            if((bool)sav.Reg[31] && (bool)sav.Reg[35]) ||
((bool)sav.Reg[33] && (bool)sav.Reg[35])) st += ", ";

```

```

        if((bool)sav.Reg[35]) st +=
(Math.Round(pl.Z, (int)sav.Reg[36])).ToString();

        sav.LabelList[i] = st;
    }

    bool found = false;
    int func = (int)sav.Reg[76];
    int grdX = (int)sav.Reg[85];
    int grdY = (int)sav.Reg[86];
    int i_grdY = 0;
    double spcX = (sav.Maxs[0]-sav.Mins[0])/grdX;
    double spcY = (sav.Maxs[1]-sav.Mins[1])/grdY;
    GridMatrix = new double[grdX+1,grdY+1];
    double[,] grid_2 = new double[grdX+1,grdY+1];
    for (int x=0;x<=grdX ; x++)
        for (int y=0;y<=grdY ; y++)
            {
GridMatrix[x,y] = double.NaN;
grid_2[x,y] = double.NaN;
            }

    pl = new EachPoint();
    ArrayList charts = (ArrayList)this.sav.Reg[75];
    ArrayList points = new ArrayList();
    chartView ch;

    Form1 f = (Form1)frm;
    f.statusBar.Panels[0].Text = "Wait...";

    for (int i=0;i<charts.Count;i++)
    {
        // loading grids
        int j=0;
        for (j = 1;j<frm.MdiChildren.Length;j++)
            {
ch = (chartView)frm.MdiChildren[j];
if (ch.Text == charts[i].ToString())
    break;
            }
        //

        if (j<frm.MdiChildren.Length)
            {
points.Clear();
ch = (chartView)frm.MdiChildren[j];
points = (ArrayList)((ArrayList)ch.Output[3]).Clone();
i_grdY = (int)ch.Output[2]+1;
for(double x = sav.Mins[0]+spcX/2;x<sav.Maxs[0];x+=spcX)
    for(double y = sav.Mins[1]+spcY/2;y<sav.Maxs[1];y+=spcY)

            {
int xi = 0;
int yi = 0;
pl=(EachPoint)points[0];

```



```

else if(func==2) GridMatrix[x,y] =
Math.Round(grid_2[x,y]/charts.Count - Math.Pow(mid,2),10);
    }

        for (int x=0;x<=grdX ; x++)
            for (int y=0;y<=grdY ; y++)
                {
if (x==0 && y==0)
{
    sav.Maxs[2]=GridMatrix[x,y];
    sav.Mins[2]=GridMatrix[x,y];
}
else
{
    if (GridMatrix[x,y]>sav.Maxs[2] || double.IsNaN(sav.Maxs[2]))

sav.Maxs[2]= GridMatrix[x,y];
    if (GridMatrix[x,y]<sav.Mins[2] || double.IsNaN(sav.Mins[2]))

sav.Mins[2]=GridMatrix[x,y];
}
                }

        ArrayList outPut = new ArrayList();
        for (int x=0;x<=grdX ; x++)
            for (int y=0;y<=grdY ; y++)
                {
EachPoint p3 = new EachPoint();
p3.X = x*(sav.Maxs[0]-sav.Mins[0])/grdX+(sav.Mins[0])+(sav.Maxs[0]-
sav.Mins[0])/(grdX);
p3.Y = y*(sav.Maxs[1]-sav.Mins[1])/grdY+(sav.Mins[1])+(sav.Maxs[1]-
sav.Mins[1])/(grdY);
p3.Z = GridMatrix[x,y];
if (p3.Z < sav.Mins[2]) p3.Z = double.NaN;
outPut.Add(p3);
                }
        sav.Reg[4] = outPut;

        if((double)sav.Reg[41] == 0 && (double)sav.Reg[42]== 0)
        {
            roundExterms();
            sav.Reg[41]=sav.Mins[0];
            sav.Reg[42]=sav.Maxs[0];
            sav.Reg[51]=sav.Mins[1];
            sav.Reg[52]=sav.Maxs[1];
            sav.Reg[43]=steps(sav.Maxs[0],sav.Mins[0]);
            sav.Reg[53]=steps(sav.Maxs[1],sav.Mins[1]);
        }
    }
}

```

**A**

Anisotropy, 9

**B**

Back ground, 84

Back transformation, 109

Base Array, 43

Binary, 81

Box Plot, 48

**C**

Clip Board, 88

Combinatorial optimization schemes, 35

Conditional, 32

Correlation, 8

Cresse, 55

Cross validation, 26

Cross variogram, 25

Curves, 100

Cyclicality, 19

**D**

D. G. Krig, 20

Deterministic interpolation techniques, 9

Direction tolerance, 15

Direction, 12

Dot Plot, 48

**E**

Ecloe des mines, 9

Estimation, 8

Experimental variogram, 15

Export , 43

**G**

Gauss Jordan, 64

Gaussian model, 18

Geometric anisotropy, 19

George Matheron, 9

Graphical User Interface (GUI), 118

## **H**

Highly suspect outliers, 99

## **I**

Import, 43

Indicator Kriging, 21

Inverse distance, 8

## **K**

Kriging, 20

## **L**

Labels, 100

Lag tolerance, 15

Legend, 95

Linear model, 17

Lines, 100

Local variables, 46

Logarithmic model, 18

## **M**

Markers, 100

Menus, 83

Merge, 91

## **N**

Nugget effect, 12

## **O**

Open, 43

Ordinary Kriging, 21

Outliers, 49

## **P**

Pixel, 108

Plot, 95

Power model, 17

Prediction, 8

Print, 43

Probability field simulation, 32

Project explorer, 84

## **Q**

Quadratic model, 18

## **R**

Range, 11

Range, 12

## **S**

Save, 43  
Scale, 13  
Schmidt & Taylor, 68  
Sequential, 31  
Sequential gaussian simulation, 32  
Sequential indicator simulation, 32  
Serialized, 43  
Setup, 78  
Sill, 11  
Simple Kriging, 21  
Simulated annealing, 32  
Simulation, 28  
Simulation, 8  
Solids, 100  
Source code, 47  
Spherical model, 18  
Split, 91  
Spreadsheet, 78  
Stationary, 12  
Status bar, 84  
Stem & left Plot, 48  
Stochastic, 28  
Structural analysis, 8  
Suspect outliers, 99

## **T**

Theory of regionalized variables, 9  
Tolerance, 53  
Toolbox, 83  
Trend, 18  
Triangulation, 8

## **U**

Uncertainty, 28  
Uncertainty, 8  
Univariate, 31  
Universal Kriging, 21

## **V**

Variography, 8

## **W**

Wave(Hole effect) model, 18  
Webster & Mc Brantey, 55

## **Z**