# Package: conicfit (via r-universe)

September 2, 2024

**Type** Package

**Title** Algorithms for Fitting Circles, Ellipses and Conics Based on the Work by Prof. Nikolai Chernov

**Version** 1.0.4

**Encoding** UTF-8

**Description** Geometric circle fitting with Levenberg-Marquardt (a, b, R), Levenberg-Marquardt reduced (a, b), Landau, Spath and Chernov-Lesort. Algebraic circle fitting with Taubin, Kasa, Pratt and Fitzgibbon-Pilu-Fisher. Geometric ellipse fitting with ellipse LMG (geometric parameters) and conic LMA (algebraic parameters). Algebraic ellipse fitting with Fitzgibbon-Pilu-Fisher and Taubin.

**License** GPL (>= 3)

**Depends** R (>= 2.7.0), pracma, geigen

**Author** Jose Gama [aut, cre], Nikolai Chernov [aut, cph]

**Maintainer** Jose Gama <rxprtgama@gmail.com>

**Repository** CRAN

**Repository/R-Forge/Project** conicfit

**Repository/R-Forge/Revision** 10

**Repository/R-Forge/DateTimeStamp** 2015-10-05 15:19:30

**Date/Publication** 2015-10-05 22:57:57

**NeedsCompilation** no

# Contents

---

`AtoG` *Conversion of algebraic parameters to geometric parameters*

---

## Description

`AtoG` converts algebraic parameters (A, B, C, D, E, F) to geometric parameters (Center(1:2), Axes(1:2), Angle).

## Usage

```
AtoG(ParA)
```

## Arguments

| | |
|---|---|
| `ParA` | vector or array with geometric parameters (A, B, C, D, E, F) |

## Format

code is: -1 - degenerate cases 0 - imaginary ellipse 4 - imaginary parell lines 1 - ellipse 2 - hyperbola 3 - parabola

## Value

`list(ParG, exitCode`
: list with algebraic parameters (Center(1:2), Axes(1:2), Angle) and exit code

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## Examples

```
AtoG(c(0.0551,-0.0908,0.1588,0.0489,-0.9669,0.1620))
```

---

# calculateCircle *Generate points from a circle*

---

## Description

`calculateCircle` generates points from a circle with many options, equally spaced, randomly spaced, with noise added to the radius or limited to a segment of angle alpha.

## Usage

```
calculateCircle(x, y, r, steps=50,sector=c(0,360),randomDist=FALSE,
randomFun=runif, noiseFun = NA, ...)
```

## Arguments

| | |
|---|---|
| `x` | center point x |
| `y` | center point y |
| `r` | radius |
| `steps` | number of points |
| `sector` | limited circular sector |
| `randomDist` | logical, TRUE = randomly spaced |
| `randomFun` | random function for the position of the points in the circle |
| `noiseFun` | random function for the noise |
| `...` | optional parameters to pass to randomFun |

### Value

| | |
|---|---|
| `points` | array n x 2 of point coordinates. |

### Author(s)

Jose Gama

### Examples

```
## Not run:
# 100 points from a circle at c(0,0) with radius=200
a<-calculateCircle(0,0,200,100)
plot(a[,1],a[,2],xlim=c(-250,250),ylim=c(-250,250))
par(new=T)
# 12 points from a circle at c(0,0) with radius=190, points between 0 and 90
#degrees
a<-calculateCircle(0,0,190,12,c(0,90))
plot(a[,1],a[,2],xlim=c(-250,250),ylim=c(-250,250),col='red')
par(new=T)
# 12 points from a circle at c(0,0) with radius=180, points between 0 and 180
#degrees, uniform random distribution
a<-calculateCircle(0,0,180,12,c(0,180),TRUE)
plot(a[,1],a[,2],xlim=c(-250,250),ylim=c(-250,250),col='green')
par(new=T)
# 12 points from a circle at c(0,0) with radius=170, points between 0 and 180
#degrees, normal random distribution
a<-calculateCircle(0,0,170,12,c(0,180),TRUE,rnorm)
plot(a[,1],a[,2],xlim=c(-250,250),ylim=c(-250,250),col='blue')
par(new=T)
# 12 points from a circle at c(0,0) with radius=200, points between 0 and 180
#degrees, positioned by uniform random distribution, noise=normal random
#distribution with sd=10
a<-calculateCircle(0,0,200,12,c(180,360),TRUE,noiseFun=function(x)
(x+rnorm(1,mean=0,sd=10)))
plot(a[,1],a[,2],xlim=c(-250,250),ylim=c(-250,250),col='orange')

## End(Not run)
```

---

## calculateEllipse *Generate points from a ellipse*

---

### Description

`calculateEllipse` generates points from a ellipse with many options, equally spaced, randomly spaced, with noise added to the radius or limited to a segment of angle alpha.

### Usage

```
calculateEllipse(x, y, a, b, angle = 0, steps = 50, sector = c(0, 360),
randomDist = FALSE, randomFun = runif, noiseFun = NA, ...)
```

### Arguments

| | |
|---|---|
| `x` | center point x |
| `y` | center point y |
| `a` | axis a |
| `b` | axis b |
| `angle` | tilt angle |
| `steps` | number of points |
| `sector` | limited circular sector |
| `randomDist` | logical, TRUE = randomly spaced |
| `randomFun` | random function for the position of the points in the ellipse |
| `noiseFun` | random function for the noise |
| `...` | optional parameters to pass to randomFun |

### Value

| | |
|---|---|
| `points` | array n x 2 of point coordinates. |

### Author(s)

Jose Gama

### Examples

```
## Not run:
# 50 points from an ellipse at c(0,0) with axis (200, 100), angle 45 degrees
a<-calculateEllipse(0,0,200,100,45,50)
plot(a[,1],a[,2],xlim=c(-250,250),ylim=c(-250,250))
par(new=T)
# 10 points from an ellipse at c(0,0) with axis (200, 100), angle 45 degrees,
#points between 0 and 180 # degrees, normal random distribution
b<-calculateEllipse(0,0,200,100,45,10,c(0,90))
plot(b[,1],b[,2],xlim=c(-250,250),ylim=c(-250,250),col='red')
par(new=T)
# 50 points from an ellipse at c(0,0) with axis (200, 100), angle 45 degrees
a<-calculateEllipse(0,0,200,100,45,50, randomDist=TRUE,noiseFun=function(x)
(x+rnorm(1,mean=0,sd=10)))
plot(a[,1],a[,2],xlim=c(-250,250),ylim=c(-250,250),col='cyan')

## End(Not run)
```

CircleFitByKasa *Algebraic circle fit (Kasa method)*

## Description

CircleFitByKasa applies the simple algebraic circle fit (Kasa method)

## Usage

```
CircleFitByKasa(XY)
```

## Arguments

| | |
|---|---|
| XY | array of sample data |

## Value

| | |
|---|---|
| vector(a, b, R) | vector with the values for the circle: center (a,b) and radius R |

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## Examples

```
xy<-calculateCircle(0,0,200,50,randomDist=TRUE,noiseFun=function(x) (x+rnorm(1,mean=0,sd=50)))
plot(xy[,1],xy[,2],xlim=c(-250,250),ylim=c(-250,250));par(new=TRUE)
c3 <- CircleFitByKasa(xy)
xyc3<-calculateCircle(c3[1],c3[2],c3[3])
plot(xyc3[,1],xyc3[,2],xlim=c(-250,250),ylim=c(-250,250),col='green',type='l');par(new=TRUE)
```

---

`CircleFitByLandau` *Geometric circle fit (minimizing orthogonal distances) by Landau algorithm*

---

## Description

`CircleFitByLandau` applies the Geometric circle fit (minimizing orthogonal distances) by Landau algorithm

## Usage

```
CircleFitByLandau(XY,ParIni = NA, epsilon = 1e-06, IterMAX = 50)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParIni` | initial guess (a, b, R) |
| `epsilon` | tolerance (small threshold) |
| `IterMAX` | maximal number of iterations, with a bad initial guess it may take >100 iterations |

## Value

| | |
|---|---|
| `vector(a, b, R)` | vector with the values for the circle: center (a,b) and radius R |

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

Geometric circle fit (minimizing orthogonal distances) by Landau algorithm M. Landau, "Estimation of a circular arc center and its radius", Computer Vision, Graphics and Image Processing, Vol. 38, pages 317-326, (1987)

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

Geometric circle fit (minimizing orthogonal distances) by Landau algorithm M. Landau, "Estimation of a circular arc center and its radius", Computer Vision, Graphics and Image Processing, Vol. 38, pages 317-326, (1987)

### Examples

```
xy<-calculateCircle(0,0,200,50,randomDist=TRUE,noiseFun=function(x) (x+rnorm(1,mean=0,sd=50)))
plot(xy[,1],xy[,2],xlim=c(-250,250),ylim=c(-250,250));par(new=TRUE)
c6 <- CircleFitByLandau(xy)
xyc6<-calculateCircle(c6[1],c6[2],c6[3])
plot(xyc6[,1],xyc6[,2],xlim=c(-250,250),ylim=c(-250,250),col='purple',type='l');par(new=TRUE)
```

---

## CircleFitByPratt *Algebraic circle fit by Pratt*

---

### Description

`CircleFitByPratt` applies the Algebraic circle fit by Pratt

### Usage

```
CircleFitByPratt(XY)
```

### Arguments

| | |
|---|---|
| `XY` | array of sample data |

### Value

| | |
|---|---|
| `vector(a, b, R)` | vector with the values for the circle: center (a,b) and radius R |

### Author(s)

Jose Gama

### Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

### References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## Examples

```
xy<-calculateCircle(0,0,200,50,randomDist=TRUE,noiseFun=function(x) (x+rnorm(1,mean=0,sd=50)))
plot(xy[,1],xy[,2],xlim=c(-250,250),ylim=c(-250,250));par(new=TRUE)
c2 <- CircleFitByPratt(xy)
xyc2<-calculateCircle(c2[1],c2[2],c2[3])
plot(xyc2[,1],xyc2[,2],xlim=c(-250,250),ylim=c(-250,250),col='blue',type='l');par(new=TRUE)
```

---

# `CircleFitBySpath` *Geometric circle fit by Spath*

---

## Description

`CircleFitBySpath` applies the Geometric circle fit by Spath

## Usage

```
CircleFitBySpath(XY, ParIni = NA, epsilon = 1e-06, IterMAX = 50)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParIni` | initial guess (a, b, R) |
| `epsilon` | tolerance (small threshold) |
| `IterMAX` | maximal number of iterations, with a bad initial guess it may take >100 iterations |

## Value

| | |
|---|---|
| `vector(a, b, R)` | vector with the values for the circle: center (a,b) and radius R |

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## Examples

```
xy<-calculateCircle(0,0,200,50,randomDist=TRUE,noiseFun=function(x) (x+rnorm(1,mean=0,sd=50)))
plot(xy[,1],xy[,2],xlim=c(-250,250),ylim=c(-250,250));par(new=TRUE)
c5 <- CircleFitBySpath(xy)
xyc5<-calculateCircle(c5[1],c5[2],c5[3])
plot(xyc5[,1],xyc5[,2],xlim=c(-250,250),ylim=c(-250,250),col='magenta',type='l');par(new=TRUE)
```

---

# CircleFitByTaubin *Algebraic circle fit (Taubin method)*

---

## Description

`CircleFitByTaubin` applies the simple algebraic circle fit (Taubin method)

## Usage

```
CircleFitByTaubin(XY)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |

## Value

| | |
|---|---|
| `vector(a, b, R)` | vector with the values for the circle: center (a,b) and radius R |

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## Examples

```
xy<-calculateCircle(0,0,200,50,randomDist=TRUE,noiseFun=function(x) (x+rnorm(1,mean=0,sd=50)))
plot(xy[,1],xy[,2],xlim=c(-250,250),ylim=c(-250,250));par(new=TRUE)
c1 <- CircleFitByTaubin(xy)
xyc1<-calculateCircle(c1[1],c1[2],c1[3])
plot(xyc1[,1],xyc1[,2],xlim=c(-250,250),ylim=c(-250,250),col='red',type='l');par(new=TRUE)
```

---

EllipseDirectFit *Algebraic ellipse fit method by Fitzgibbon-Pilu-Fisher*

---

## Description

`EllipseDirectFit` applies the algebraic ellipse fit method by Fitzgibbon-Pilu-Fisher

## Usage

```
EllipseDirectFit(XY)
```

## Arguments

| | |
|---|---|
| XY | array of sample data |

## Value

`vector(A, B, C, D, E, F)`
vector of algebraic parameters of the fitting ellipse: ax^2 + bxy + cy^2 +dx + ey + f = 0

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

A. W. Fitzgibbon, M. Pilu, R. B. Fisher, 1999 Direct Least Squares Fitting of Ellipses IEEE Trans. PAMI, Vol. 21, pages 476-480

A. W. Fitzgibbon, M. Pilu, R. B. Fisher, "Direct Least Squares Fitting of Ellipses", IEEE Trans. PAMI, Vol. 21, pages 476-480 (1999) Halir R, Flusser J (1998) Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualization, Numerically stable direct least squares fitting of ellipses (WSCG, Plzen, Czech Republic), pp 125–132.

### References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

A. W. Fitzgibbon, M. Pilu, R. B. Fisher, 1999 Direct Least Squares Fitting of Ellipses IEEE Trans. PAMI, Vol. 21, pages 476-480

A. W. Fitzgibbon, M. Pilu, R. B. Fisher, "Direct Least Squares Fitting of Ellipses", IEEE Trans. PAMI, Vol. 21, pages 476-480 (1999) Halir R, Flusser J (1998) Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualization, Numerically stable direct least squares fitting of ellipses (WSCG, Plzen, Czech Republic), pp 125–132.

### Examples

```
xy<-calculateEllipse(0,0,200,100,45,50, randomDist=TRUE,noiseFun=function(x)
(x+rnorm(1,mean=0,sd=50)))
plot(xy[,1],xy[,2],xlim=c(-250,250),ylim=c(-250,250),col='magenta');par(new=TRUE)

ellipDirect <- EllipseDirectFit(xy)
ellipDirectG <- AtoG(ellipDirect)$ParG
xyDirect<-calculateEllipse(ellipDirectG[1], ellipDirectG[2], ellipDirectG[3],
ellipDirectG[4], 180/pi*ellipDirectG[5])
plot(xyDirect[,1],xyDirect[,2],xlim=c(-250,250),ylim=c(-250,250),type='l',
col='cyan');par(new=TRUE)
```

---

## EllipseFitByTaubin *Algebraic ellipse fit by Taubin*

---

### Description

`EllipseFitByTaubin` applies the Algebraic ellipse fit by Taubin

### Usage

```
EllipseFitByTaubin(XY)
```

### Arguments

| | |
|---|---|
| XY | array of sample data |

### Value

`vector(A, B, C, D, E, F)`
vector with the values for the ellipse

### Author(s)

Jose Gama

### Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

### References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

### Examples

```
xy<-calculateEllipse(0,0,200,100,45,50, randomDist=TRUE,noiseFun=function(x)
(x+rnorm(1,mean=0,sd=50)))
plot(xy[,1],xy[,2],xlim=c(-250,250),ylim=c(-250,250),col='magenta');par(new=TRUE)

ellipTaubin <- EllipseFitByTaubin(xy)
ellipTaubinG <- AtoG(ellipTaubin)$ParG
xyTaubin<-calculateEllipse(ellipTaubinG[1], ellipTaubinG[2], ellipTaubinG[3],
ellipTaubinG[4], 180/pi*ellipTaubinG[5])
plot(xyTaubin[,1],xyTaubin[,2],xlim=c(-250,250),ylim=c(-250,250),type='l',
col='red');par(new=TRUE)
```

---

## `ellipticity` *Formulas for the ellipse*

---

### Description

`ellipticity` ellipticity = flattening factor `ellipseEccentricity` eccentricity of the ellipse `ellipseFocus` focus of the ellipse `ellipseRa` radius at apoapsis (the farthest distance) `ellipseRp` radius at periapsis (the closest distance) `ellipse.l` semi-latus rectum l

### Usage

```
ellipticity(minorAxis,majorAxis)
```

### Arguments

| | |
|---|---|
| `minorAxis` | minor ellipse axis |
| `majorAxis` | major ellipse axis |

### Value

scalar result

## Author(s)

Jose Gama

## Source

Wikipedia Ellipse http://en.wikipedia.org/wiki/Ellipse#Mathematical_definitions_and_properties

## References

Wikipedia Ellipse http://en.wikipedia.org/wiki/Ellipse#Mathematical_definitions_and_properties

---

estimateInitialGuessCircle
*Estimate Initial Guess Circle values*

---

## Description

`estimateInitialGuessCircle` estimates initial guess values for the center and radius of the circle

## Usage

```
estimateInitialGuessCircle(XY)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |

## Value

| | |
|---|---|
| `vector(a, b, R)` | vector with the estimates for the circle: center (a,b) and radius R |

## Author(s)

Jose Gama

## Examples

```
xy<-calculateCircle(0,0,200,50,randomDist=TRUE,noiseFun=function(x) (x+rnorm(1,mean=0,sd=50)))
estimateInitialGuessCircle(xy)
```

fit.conicLMA *Fitting a conic to a given set of points (Implicit method)*

## Description

fit.conicLMA fits a conic to a given set of points (Implicit method) using algebraic parameters. Conic: Ax^2 + Bxy + Cy^2 +Dx + Ey + F = 0

## Usage

```
fit.conicLMA(XY, ParAini, LambdaIni, epsilonP = 1e-10, epsilonF = 1e-13,
IterMAX = 2e+06)
```

## Arguments

| | |
|---|---|
| XY | array of sample data |
| ParAini | initial parameter vector c(A,B,C,D,E,F) |
| LambdaIni | initial value of the control parameter Lambda |
| epsilonP | tolerance (small threshold) |
| epsilonF | tolerance (small threshold) |
| IterMAX | maximum number of (main) iterations, usually 10-20 will suffice |

## Value

list(ParA, RSS, iters

list with algebraic parameters (Center(1:2), Axes(1:2), Angle), Residual Sum of Squares and number of iterations

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

N. Chernov and H. Ma, 2011 Least squares fitting of quadratic curves and surfaces In: Computer Vision, Editor S. R. Yoshida, Nova Science Publishers; pp. 285-302.

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

N. Chernov and H. Ma, 2011 Least squares fitting of quadratic curves and surfaces In: Computer Vision, Editor S. R. Yoshida, Nova Science Publishers; pp. 285-302.

## Examples

```
XY <- matrix(c(1,7,2,6,5,8,7,7,9,5,3,7,6,2,8,4),8,2,byrow=TRUE)
ParAini <- matrix(c(0.2500,0, 1.0000, 0, 0, -1.0000),ncol=1)
LambdaIni=0.1
fit.conicLMA(XY,ParAini,LambdaIni)
```

---

# fit.ellipseLMG *Fitting an ellipse using Implicit method*

---

## Description

`fit.ellipseLMG` Fits an ellipse to a given set of points (Implicit method) using geometric parameters. Conic:

## Usage

```
fit.ellipseLMG(XY,ParGini,LambdaIni = 1, epsilon = 1e-06, IterMAX = 200,
L = 200)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParGini` | initial parameter vector c(Center(1:2), Axes(1:2), Angle) |
| `LambdaIni` | initial value of the control parameter Lambda |
| `epsilon` | tolerance (small threshold) |
| `IterMAX` | maximum number of (main) iterations, usually 10-20 will suffice |
| `L` | boundary for major/minor axis |

## Value

`list(ParG, RSS, iters, TF)`
list with geometric parameters (A,B,C,D,E,F), Residual Sum of Squares, number of iterations and TF==TRUE if the method diverges

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

N. Chernov and H. Ma, 2011 Least squares fitting of quadratic curves and surfaces In: Computer Vision, Editor S. R. Yoshida, Nova Science Publishers; pp. 285-302.

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

N. Chernov and H. Ma, 2011 Least squares fitting of quadratic curves and surfaces In: Computer Vision, Editor S. R. Yoshida, Nova Science Publishers; pp. 285-302.

## Examples

```
XY <- matrix(c(1,7,2,6,5,8,7,7,9,5,3,7,6,2,8,4),8,2,byrow=TRUE)
ParGini <- matrix(c(0,0,2,1,0),ncol=1)
LambdaIni=0.1
fit.ellipseLMG(XY,ParGini,LambdaIni)
```

---

# fitbookstein — *Linear ellipse fit using bookstein constraint*

---

## Description

`fitbookstein` Linear ellipse fit using bookstein constraint

`conic2parametric` Diagonalise A - find Q, D such at A = Q' * D * Q

`fitggk` Linear least squares with the Euclidean-invariant constraint Trace(A) = 1

## Usage

```
fitbookstein(x)
```

## Arguments

| | |
|---|---|
| `x` | array of sample data |

## Value

`list(z, a, b, alpha)`
: list with fitted ellipse parameters

## Author(s)

Jose Gama

## Source

Richard Brown, May 28, 2007 http://www.mathworks.com/matlabcentral/fileexchange/15125-fitellipse-m/content/demo/html/ellipsedemo.html

## References

Richard Brown, May 28, 2007 http://www.mathworks.com/matlabcentral/fileexchange/15125-fitellipse-m/content/demo/html/ellipsedemo.html

W. Gander, G. H. Golub, R. Strebel, 1994 Least-Squares Fitting of Circles and Ellipses BIT Numerical Mathematics, Springer

---

GtoA *Conversion of geometric parameters to algebraic parameters*

---

## Description

GtoA converts geometric parameters (A, B, C, D, E, F) to algebraic parameters (Center(1:2), Axes(1:2), Angle).

## Usage

```
GtoA(ParG)
```

## Arguments

| | |
|---|---|
| ParG | list with geometric parameters (A, B, C, D, E, F) |

## Value

| | |
|---|---|
| ParA | vector or array with algebraic parameters (Center(1:2), Axes(1:2), Angle) |

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

### References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

### Examples

```
GtoA(c(0,0,20,60,45))
```

---

## JmatrixLMA — *Compute the Jacobian matrix using algebraic parameters*

---

### Description

`JmatrixLMA` Computes the Jacobian matrix(Implicit method) using algebraic parameters

### Usage

```
JmatrixLMA(XY,ParA,XYproj)
```

### Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParA` | initial parameter vector c(Center(1:2), Axes(1:2), Angle) |
| `XYproj` | corresponding n projection points on the conic |

### Value

| | |
|---|---|
| `list(Res, J)` | list with the Residual Sum of Squares and the Jacobian matrix |

### Author(s)

Jose Gama

### Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

### References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

### Examples

```
XY <- matrix(c(1,7,2,6,5,8,7,7,9,5,3,7,6,2,8,4),8,2,byrow=TRUE)
ParA <- matrix(c(0.250000000000000,0,1,0,0,-1),ncol=1)
XYproj=matrix(c(0.394467220216675,0.980356518335872,0.833315950425981,
0.909063326557293,1.40466123643977,0.711850899213363,
1.70601340510202,0.521899957274429,1.89925244997324,0.313384799914835,
1.06482258038841,0.846485805004280,1.95308457257492,
0.215325713960169,1.91319150256275,0.291418202297698),8,2,byrow=TRUE)
JmatrixLMA(XY,ParA,XYproj)
```

---

`JmatrixLMG` *Compute the Jacobian matrix using geometric parameters*

---

### Description

`JmatrixLMG` Computes the Jacobian matrix (Implicit method) using geometric parameters

### Usage

```
JmatrixLMG(XY,A,XYproj)
```

### Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `A` | initial parameter vector c(Xc,Yc,a,b,alpha) |
| `XYproj` | corresponding n projection points on the conic |

### Value

| | |
|---|---|
| `list(Res, J)` | list with the Residual Sum of Squares and the Jacobian matrix |

### Author(s)

Jose Gama

### Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

### References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## Examples

```
XY <- matrix(c(1,7,2,6,5,8,7,7,9,5,3,7,6,2,8,4),8,2,byrow=TRUE)
A <- matrix(c(0,0,2,1,0),ncol=1)
XYproj=matrix(c(0.394467220216675,0.980356518335872,0.833315950425981,
0.909063326557293,1.40466123643977,0.711850899213363,
1.70601340510202,0.521899957274429,1.89925244997324,0.313384799914835,
1.06482258038841,0.846485805004280,1.95308457257492,
0.215325713960169,1.91319150256275,0.291418202297698),8,2,byrow=TRUE)
JmatrixLMG(XY,A,XYproj)
```

---

`LMcircleFit` *Geometric circle fit (minimizing orthogonal distances) based on the Levenberg-Marquardt method*

---

## Description

`LMcircleFit` applies a Geometric circle fit (minimizing orthogonal distances) based on the standard Levenberg-Marquardt scheme

## Usage

```
LMcircleFit(XY, ParIni, LambdaIni = 1, epsilon = 1e-06, IterMAX = 50)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParIni` | initial guess (a, b, R) |
| `LambdaIni` | initial value for the correction factor lambda |
| `epsilon` | tolerance (small threshold) |
| `IterMAX` | maximum number of (main) iterations, usually 10-20 will suffice |

## Value

| | |
|---|---|
| `vector(a, b, R)` | vector with the estimates for the circle: center (a,b) and radius R |

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares `http://people.cas.uab.edu/~mosya/cl/`

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## Examples

```
xy<-calculateCircle(0,0,200,50,randomDist=TRUE,noiseFun=function(x) (x+rnorm(1,mean=0,sd=50)))
plot(xy[,1],xy[,2],xlim=c(-250,250),ylim=c(-250,250));par(new=TRUE)
c4 <- LMcircleFit(xy)
xyc4<-calculateCircle(c4[1],c4[2],c4[3])
plot(xyc4[,1],xyc4[,2],xlim=c(-250,250),ylim=c(-250,250),col='cyan',type='l')
```

---

# LMreducedCircleFit — *Geometric circle fit (minimizing orthogonal distances) based on the Levenberg-Marquardt method*

## Description

`LMreducedCircleFit` applies a Geometric circle fit (minimizing orthogonal distances) based on the standard Levenberg-Marquardt scheme in the "reduced" (a,b) parameter space

## Usage

```
LMreducedCircleFit(XY, ParIni, LambdaIni = 1, epsilon = 1e-06,
IterMAX = 50)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParIni` | initial guess (a, b) |
| `LambdaIni` | initial value for the correction factor lambda |
| `epsilon` | tolerance (small threshold) |
| `IterMAX` | maximum number of (main) iterations, usually 10-20 will suffice |

## Value

| | |
|---|---|
| `vector(a, b, R)` | vector with the estimates for the circle: center (a,b) and radius R |

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

Nikolai Chernov, 2010 Circular and linear regression: Fitting circles and lines by least squares Chapman & Hall/CRC, Monographs on Statistics and Applied Probability, Volume 117

## Examples

```
xy<-calculateCircle(0,0,200,50,randomDist=TRUE,noiseFun=function(x) (x+rnorm(1,mean=0,sd=50)))
plot(xy[,1],xy[,2],xlim=c(-250,250),ylim=c(-250,250));par(new=TRUE)
c7 <- LMreducedCircleFit(xy)
xyc7<-calculateCircle(c7[1],c7[2],c7[3])
plot(xyc7[,1],xyc7[,2],xlim=c(-250,250),ylim=c(-250,250),col='pink',type='l')
```

---

# `Residuals.ellipse` *Projecting a given set of points onto an ellipse*

---

## Description

`Residuals.ellipse` projects a given set of points onto an ellipse and computing the distances from the points to the ellipse

## Usage

```
Residuals.ellipse(XY,ParG)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParG` | vector 5x1 of the ellipse parameters (Center(1:2), Axes(1:2), Angle) |

## Value

| | |
|---|---|
| `list(Res, J)` | list with the Residual Sum of Squares and the Jacobian matrix |

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

## Examples

```
XY <- matrix(c(1,7,2,6,5,8,7,7,9,5,3,7,6,2,8,4),8,2,byrow=TRUE)
ParG <- matrix(c(0,0,2,1,0),ncol=1)
Residuals.ellipse(XY,ParG)
```

---

# Residuals.hyperbola *Projecting a given set of points onto an hyperbola*

---

## Description

`Residuals.hyperbola` projects a given set of points onto an hyperbola and computing the distances from the points to the hyperbola

## Usage

```
Residuals.hyperbola(XY,ParG)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParG` | vector 5x1 of the hyperbola parameters (Center(1:2), Axes(1:2), Angle) |

## Value

`list(RSS, XYproj)`
list with the Residual Sum of Squares and the array of coordinates of projections

## Author(s)

Jose Gama

### Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

### References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

### Examples

```
XY <- matrix(c(1,7,2,6,5,8,7,7,9,5,3,7,6,2,8,4),8,2,byrow=TRUE)
ParG <- matrix(c(0,0,2,1,0),ncol=1)
Residuals.hyperbola(XY,ParG)
```

---

## `Residuals.parabola` *Projecting a given set of points onto an parabola*

---

### Description

`Residuals.parabola` projects a given set of points onto an parabola and computing the distances from the points to the parabola

### Usage

```
Residuals.parabola(XY,ParG)
```

### Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParG` | vector 4x1 of the parabola parameters (Vertex(1:2), p, Angle) |

### Value

`list(RSS, XYproj)`
list with the Residual Sum of Squares and the array of coordinates of projections

### Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

## Examples

```
XY <- matrix(c(1,7,2,6,5,8,7,7,9,5,3,7,6,2,8,4),8,2,byrow=TRUE)
ParG <- matrix(c(0,0,2,1,0),ncol=1)
Residuals.parabola(XY,ParG)
```

---

`ResidualsG` *Projecting a given set of points onto an ellipse*

---

## Description

`ResidualsG` projects a given set of points onto an ellipse and computing the distances from the points to the ellipse

## Usage

```
ResidualsG(XY,ParG)
```

## Arguments

| | |
|---|---|
| `XY` | array of sample data |
| `ParG` | vector 5x1 of the ellipse parameters (Center(1:2), Axes(1:2), Angle) |

## Value

`list(RSS, XYproj)`
list with the Residual Sum of Squares and the array of coordinates of projections

## Author(s)

Jose Gama

## Source

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

## References

Nikolai Chernov, 2014 Fitting ellipses, circles, and lines by least squares http://people.cas.uab.edu/~mosya/cl/

N. Chernov, Q. Huang, and H. Ma, 2014 Fitting quadratic curves to data points British Journal of Mathematics & Computer Science, 4, 33-60.

## Examples

```
XY <- matrix(c(1,7,2,6,5,8,7,7,9,5,3,7,6,2,8,4),8,2,byrow=TRUE)
ParG <- matrix(c(0,0,2,1,0),ncol=1)
ResidualsG(XY,ParG)
```

# Index