Toolkit 1

Normal Linear Algebra

1, gaussElimin *need rearranging the matric before implementing*: gaussElimin(a, b)

2, vandermode (matrix generation)]: vandermode(, B)

3, Doolittle decomposition (LU\_decomposite): LUmain(, b)

4, Doolittle decomposition for tridiagonal coefficient matrix: Ludecomp\_for\_trig(a, b)

5, Choleski’s decompoition *only used with symmetric matrices*: choleski\_func(a, b)

6, symmetric coefficient matrix A=LDLT

Inversion of Matrices

1, inversion\_LU & inversion\_gauss: matInv(a)

Swap Linear Algebra

1, swap\_gauss: gaussPivot(a, b)

2, swap\_LU: Lumain(a, b)

Polynomial Extrapolation

1, newton\_ploy: np\_main(xData, yData, x)

2, neville\_poly: neville(xData, yData, x)

3, rational: rational(xData, yData, x)

4, least\_square\_fit: polyFit(xData, yData, m)

**stdDev(c, xData, yData)**

5, poly\_plot: plotPoly(xData, yData, coeff, xlab='x', ylab='y')

Root Searching Tools

1, rootsearch: main(f, x1, x2)

2, rootsearch\_modified: main(f, x1, x2)

3, newton\_raphson: main(f, df, x1, x2), recommend re\_raph(f, df, x)

Ne\_raph\_diff(f, x)

Differentiation:

derivative({xData}, {yData})

Integration:

trapezoid: trap\_main(f, a, b)

euler\_integrate: euler\_old(f, a, b, steps) or euler\_new(f, a, b, steps)

runge\_gutta\_init: integrate(f, xStart, [yStart], xStop, steplength)

Eigenvalue & eigenvector of matrix

Jacobi: jacobi(matrix)

Inverse power: inver\_power(matrix, approximation)

Optimisation

Gold search: gold\_search(f, xStart, step\_size)

Course

Curve fitting & Interpolating

Plotting curves for a dataset with N+1 number of y-xs such that

1, np.polyfit(**xARRAY**, **yARRAY**, deg\_of\_fit)---**poly\_coeffARRAY**: Highest power coeff on the back; best deg=max power or degree of polynomial=(N+1)-1

Degree N polynomial has N+1 coefficients

2, np.plot1d(**poly\_coeffARRAY**)---*function*(x), return def style func.

3, sci.interpolate.lagrange**(xARRAY**, **yARRAY**), return def style func

4, newton\_new.func\_newton(**xARRAY,** **yARRAY, x\_range\_ARRAY**), return def style func