**GSSA MATLAB code**

**Version Notes**

Current version is 1.3, dated 11 Feb 2012

* Added MATLAB code from Judd, Maliar & Maliar.
  + "Num\_Stab\_Approx.m" - implements the numerically stable LS and LAD approximation methods.
  + "Ord\_Polynomial\_N.m" - constructs the sets of basis functions for ordinary polynomials of the degrees from one to five, for the N-country model.
  + "Monomials\_1.m" - constructs integration nodes and weights for an N-dimensional monomial (non-product) integration rule with 2N nodes.
  + "Monomials\_2.m" - constructs integration nodes and weights for an N-dimensional monomial (non-product) integration rule with 2N^2+1 nodes.
  + "GH\_Quadrature.m" - constructs integration nodes and weights for the Gauss-Hermite rules with the number of nodes in each dimension ranging from one to ten.
* These allow greater richness in the fitting functions and also implementation of Euler error calculations.

Version 1.2, dated 4 Feb 2012

* Changed variable names and notation to allow for linear to be special case of quadratic.
* Changed functions GSSA\_fittype1.m & GSSA\_fittype2.m to GSSA\_fittype.m, which now works for all functional forms for converting parameters.

Issues:

* Regressions in the levels (linear & quadratic) tend to not converge. Perhaps a smaller convexifier parameter should be used. Or maybe a different regression method.

Things to do include:

* Allow for more general forms of quadrature in calculating expectations, including integration over many shocks, not just one.
* Optimally choose the weights and nodes for this quadrature.

Version 1.1, dated 4 Feb 2012

* Added options for quadratic (fittype=2) & log-quadratic (fittype=3) fitting.
* Added function GSSA\_fittype2.m for converting quadratic parameters.
* Added a Monte Carlo average of the Euler errors to the outputs.

Version 1.0, dated 3 Feb 2012

GSSA is called as a function, GSSA.m

Inputs are: 1) a vector of steady state value guesses & 2) a string with the model name.

Outputs are: 1) a vector of approximation function parameters & 2) a vector of steady state values.

A function file called “modelname”\_dyn.m is required. It must take column vector inputs for X(t+2), X(t+1), X(t), Y(t+1), Y(t), Z(t+1) & Z(t) and produce a column vector output for the stacked sets of equations defining the model. The first *ny* equations define the jump variables, Y, and take the form:

The last first *nx* equations define the endogenous state variables, X, and take the form:

Stand-alone functions GSSA\_fittype***x***.m are also required to convert the vector of parameters into their matrix forms. The ***x*** indexes the type of regression that is run. Currently there is only ***x***=1, for linear or log-linear parameters.