Comparing fit_gls.m and fit_fgls.m

Objectives

1. Compare the runtime of `fit_gls` and `fit_fgls`.
2. Evaluate the estimation accuracy of the coefficients for different:
   - Number of observations (T).
   - AR model orders (p).
   - Variance settings (σ^2).

fit_fgls

The `fit_fgls` function leverages MATLAB's built-in `fgls` method to perform GLS estimation with AR(p) innovations.

Simulation Settings

Data Generation

The response variable y is generated using the following model:

$$y_t = \mathbf{X}_t\boldsymbol{\beta} + \epsilon_t$$

where:

- $\mathbf{X}$: Design matrix ($T \times 3$) including an intercept and two predictors.
- $\boldsymbol{\beta} = [2, 3, -1]^T$: True coefficients.
- ϵ_t: Follows an AR(p) process:

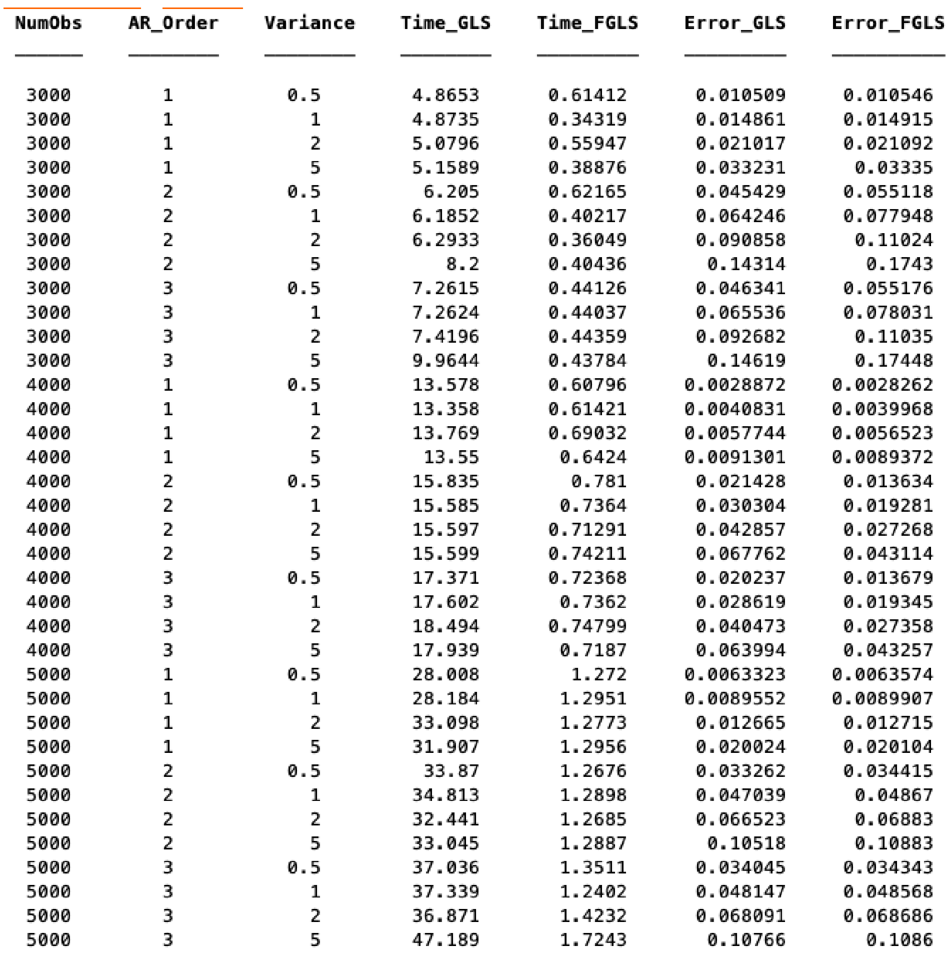
$$\epsilon_t = \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + \cdots + \phi_p\epsilon_{t-p} + \nu_t$$

with $\nu_t \sim N(0, \sigma^2)$ and $\phi_i = 0.5$.

Simulation Parameters

- **Number of Observations (T)**: 300, 500, 1000.
- **AR Orders (p)**: 1, 2, 3.
- **Variance (σ^2)**: 1, 2, 5.

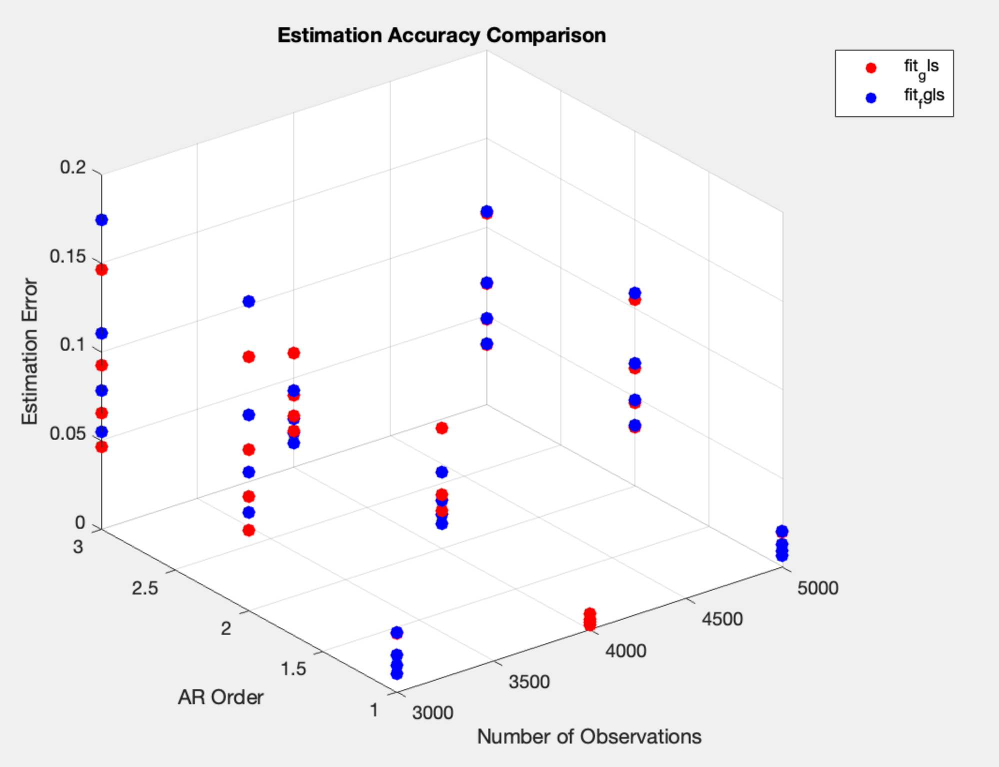
Results



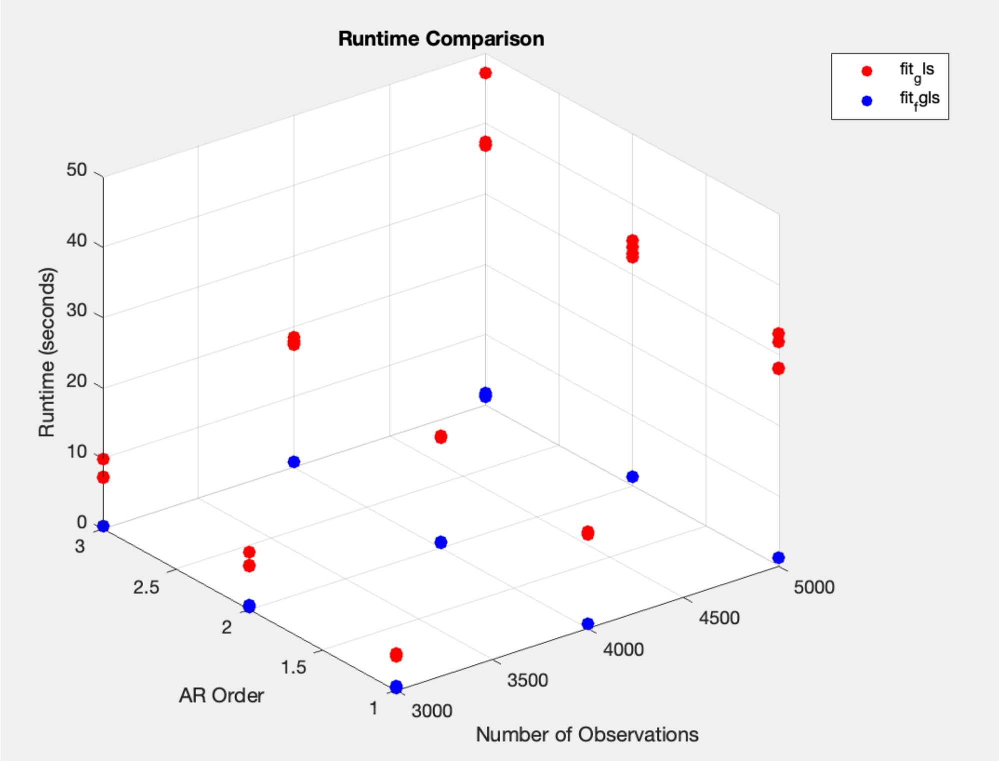
```
NumObs    AR_Order    Variance    Time_GLS    Time_FGLS    Error_GLS    Error_FGLS
______    ________    ________    ________    _________    _________    __________

 3000        1           0.5       4.8653      0.61412      0.010509      0.010546
 3000        1             1       4.8735      0.34319      0.014861      0.014915
 3000        1             2       5.0796      0.55947      0.021017      0.021092
 3000        1             5       5.1589      0.38876      0.033231       0.03335
 3000        2           0.5        6.205      0.62165      0.045429      0.055118
 3000        2             1       6.1852      0.40217      0.064246      0.077948
 3000        2             2       6.2933      0.36049      0.090858       0.11024
 3000        2             5          8.2      0.40436       0.14314        0.1743
 3000        3           0.5       7.2615      0.44126      0.046341      0.055176
 3000        3             1       7.2624      0.44037      0.065536      0.078031
 3000        3             2       7.4196      0.44359      0.092682       0.11035
 3000        3             5       9.9644      0.43784       0.14619       0.17448
 4000        1           0.5       13.578      0.60796     0.0028872     0.0028262
 4000        1             1       13.358      0.61421     0.0040831     0.0039968
 4000        1             2       13.769      0.69032     0.0057744     0.0056523
 4000        1             5        13.55       0.6424     0.0091301     0.0089372
 4000        2           0.5       15.835        0.781      0.021428      0.013634
 4000        2             1       15.585       0.7364      0.030304      0.019281
 4000        2             2       15.597      0.71291      0.042857      0.027268
 4000        2             5       15.599      0.74211      0.067762      0.043114
 4000        3           0.5       17.371      0.72368      0.020237      0.013679
 4000        3             1       17.602       0.7362      0.028619      0.019345
 4000        3             2       18.494      0.74799      0.040473      0.027358
 4000        3             5       17.939       0.7187      0.063994      0.043257
 5000        1           0.5       28.008        1.272     0.0063323     0.0063574
 5000        1             1       28.184       1.2951     0.0089552     0.0089907
 5000        1             2       33.098       1.2773      0.012665      0.012715
 5000        1             5       31.907       1.2956      0.020024      0.020104
 5000        2           0.5        33.87       1.2676      0.033262      0.034415
 5000        2             1       34.813       1.2898      0.047039       0.04867
 5000        2             2       32.441       1.2685      0.066523       0.06883
 5000        2             5       33.045       1.2887       0.10518       0.10883
 5000        3           0.5       37.036       1.3511      0.034045      0.034343
 5000        3             1       37.339       1.2402      0.048147      0.048568
 5000        3             2       36.871       1.4232      0.068091      0.068686
 5000        3             5       47.189       1.7243       0.10766        0.1086
```

Runtime Results Table



Runtime Comparison Plot



Discussion

- **Runtime**:

  - `fit_fgls` is consistently faster than `fit_gls`, particularly for larger datasets and higher AR orders.
  - The performance gap increases as the dataset size grows.

- **Estimation Accuracy**:
  - Both methods perform similarly in terms of estimation accuracy.
  - `fit_fgls` shows slightly lower errors, particularly for larger datasets and higher AR orders.

Conclusion

This simulation study demonstrates that: 1. `fit_fgls` is more efficient in terms of runtime, particularly for larger datasets. 2. Both methods provide accurate parameter estimates, but `fit_fgls` is slightly more reliable for complex settings.