Evaluation of the Lindstrom-Bates FOCE Algorithm with Simulated Pharmacokinetic Datasets

J. Chittenden, R. H. Leary, B. Matzuka, M. Dunlavey

Pharsight Corporation, Cary, NC, USA

Pharsight
A Certara™ Company

OBJECTIVE

Accuracy, run time, and robustness are primary concerns when selecting a method for estimation of a population pharmacokinetic model. Historically, the most popular methods have been the First Order Conditional Estimation (FOCE) methods provided by NONMEM®, which generally provide acceptable accuracy in parameter estimates and likelihood evaluation. An alternative FOCE algorithm was proposed by Lindstrom and Bates [1] that allows a significant simplification to the optimization of the likelihood and should result in faster runtimes. To our knowledge, an extensive investigation of the accuracy, speed, and robustness of the Lindstrom-Bates FOCE (FOCE-LB) algorithm with respect to pharmacokinetic data has not been conducted. This work compares the FOCE-LB algorithm implemented in Phoenix® NLME 1.1 with NONMEM VII FOCE (FOCE-ELS) results.

METHODS

For this evaluation we use a large set of test data and models that were previously generated by Laveille et al [2] for an evaluation of the SAEM algorithm implemented in MONOLIX. The models were transcribed to Pharsight Modeling Language (PML) for use with Phoenix NLME and both Phoenix NLME and NONMEM model sets were automated in the same environment with equivalent settings and initial estimates. We compare run time of the main algorithm (excludes covariance estimate), ELS log-likelihood to ascertain quality of convergence, convergence code or message, and parameter estimates. We compare cases where FOCE-ELS and FOCE-LB both converge and highlight cases where convergence is different.

RESULTS

1. The FOCE-LB algorithm converges in 94% of the cases, compared to 73% for FOCE-ELS.
2. The FOCE-LB algorithm was faster in 94% of the cases where both algorithms converged. On average FOCE-LB was 4 times faster than FOCE-ELS and it was 13 times faster over all the mutually converged problems.
3. The FOCE-LB algorithm converged in 97% of the cases in which FOCE-ELS also converged. Of these, 25% were significantly worse and 23% were significantly better than FOCE-ELS results.
4. The FOCE-ELS algorithm converged in 77% of the cases in which FOCE-LB also converged. Of these, 25% converged to a different and significantly worse result and 21% converged to a different and significantly better result than FOCE-LB results.

DISCUSSION

1. When comparing estimates, we consider results equivalent when: if NONMEM returned standard errors from the covariance step the FOCE-LB estimate falls within the confidence interval, or if no standard errors are available, the estimate is within 5%.
2. When comparing objective function values we take a difference greater than 2 to be significant.
3. When comparing runtimes, we use ratios. The base 2 logarithm is convenient for comparison because it spans several integer values (for these results) and can be interpreted as factors of 2 faster or slower.
4. Convergence of FOCE-ELS was gauged as either successful or unsuccessful. The only failures seen were termination due to precision, which corresponds to the FOCE-LB return code (-1). Messages of success or probable success from FOCE-ELS correspond, we think, to FOCE-LB return codes of (1,2) and (3) respectively. The FOCE-LB codes 1, 2, and 3 correspond to convergence by tolerance on gradient, parameter values, and objective function value, respectively.
5. We found that convergence and runtimes are sensitive to initial estimates of residual error, with FOCE-ELS performing better with a lower initial estimate while FOCE-LB fared better with a larger estimate.
6. The models were all reformulated to use a log-additive error model. While this is certainly beneficial for the FOCE-LB case, it actually helped FOCE-ELS to converge in cases where the Monolix evaluation [2] used FO.
7. Figure 1 depicts the log2 runtime ratios for each of the model runs. Red symbols indicate convergence, while black indicate non-convergence. Circles are used for FOCE-LB and crosshairs are used for FOCE-ELS. The black line is the cumulative runtime distribution, with values on the left axis. The right axis gives the NLME (FOCE-LB) runtime in seconds. The bottom axis shows the log2(runtime ratio). Values < 0 indicate FOCE-LB was faster.
8. Figure 2 shows the objective function value for each model. Symbols are the same as for Figure 1. Values are truncated at ±100.

CONCLUSION

The FOCE-LB algorithm as implemented Phoenix NLME is a fast, accurate, and reliable method for estimating population pharmacokinetic models in the 150 Monolix test cases investigated here..

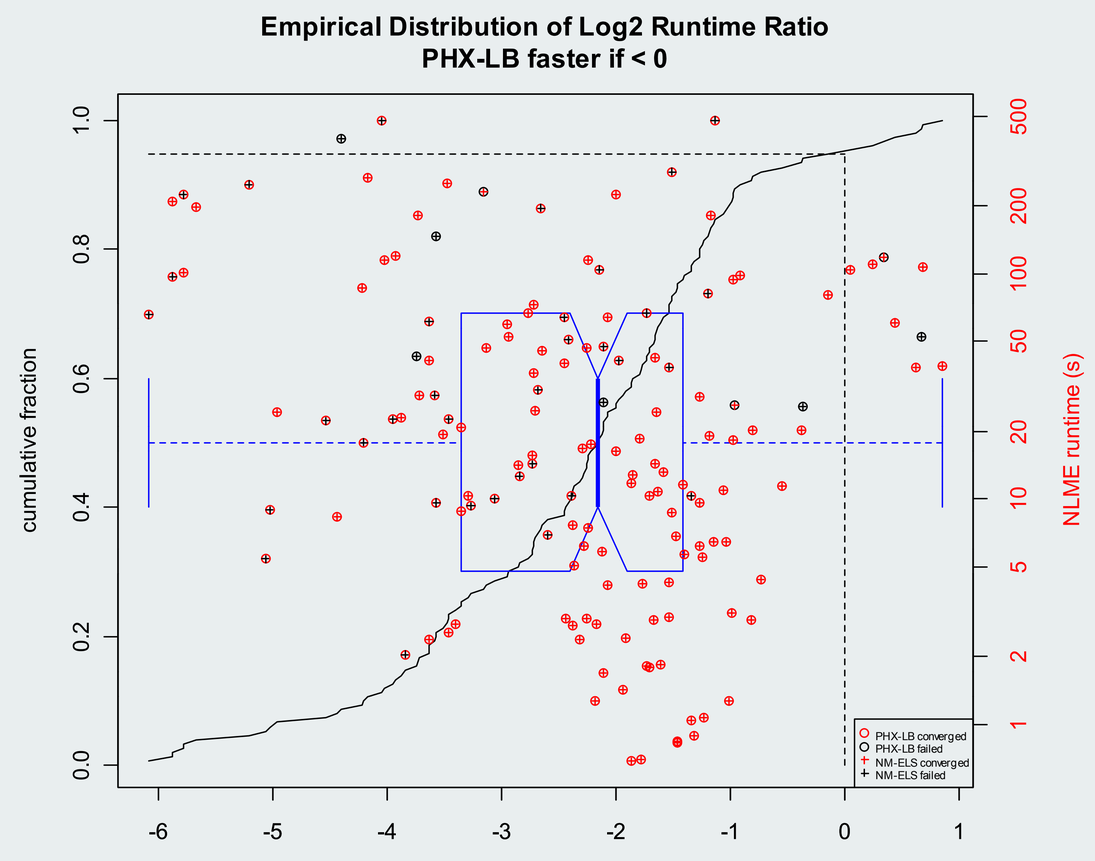

Figure 1

ELS Objective Function: Nonmem VII FOCEI - Phoenix NLME FOCEI-LB

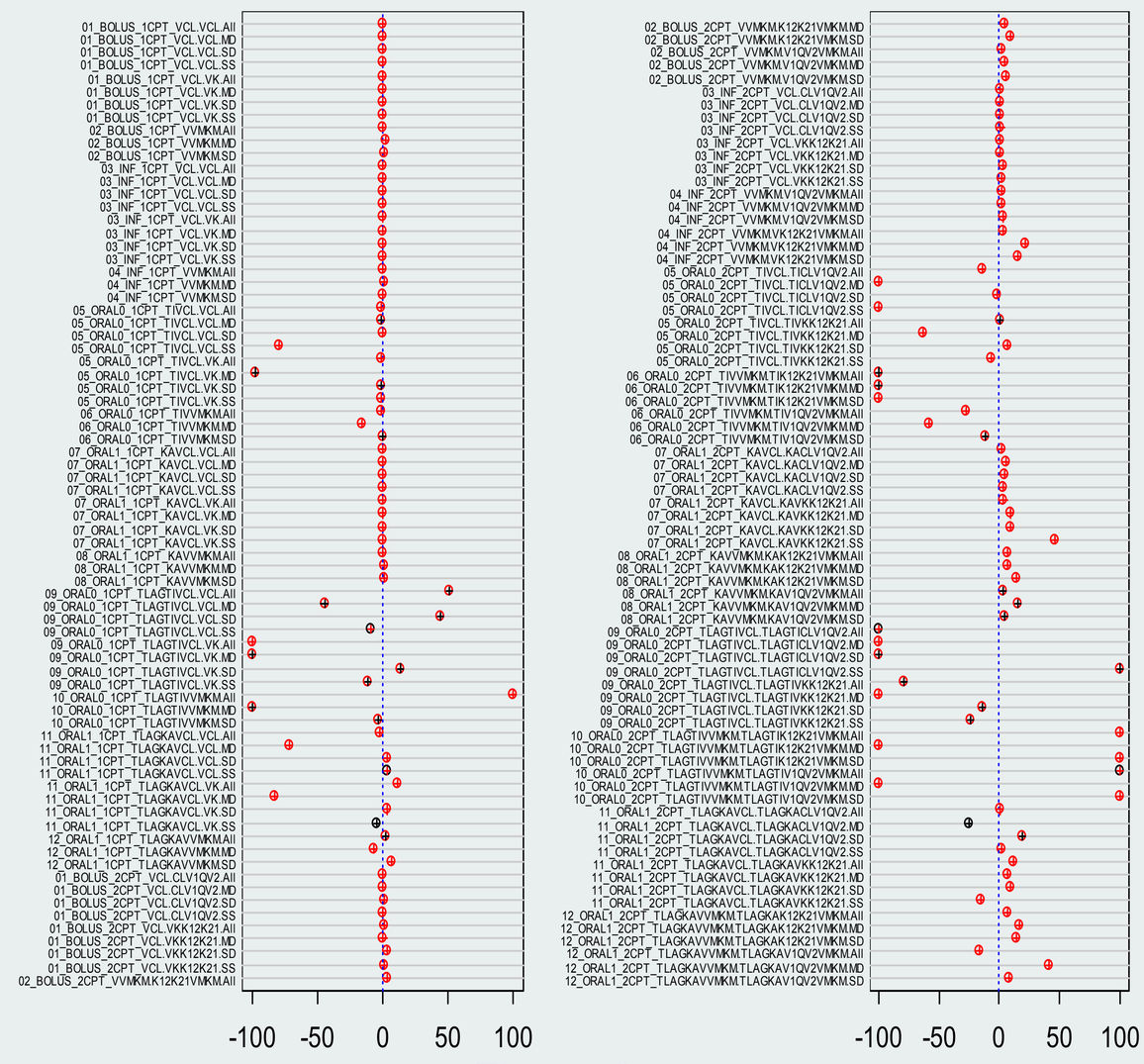

Figure 2

	FOCE-LB converged	FOCE-LB failed
FOCE-ELS converged	107	3
FOCE-ELS failed	34	6

Table 1 - Convergence

REFERENCES

1. [1] Nonlinear Mixed Effects Models for Repeated Measures Data, M. Lindstrom, D. Bates, Biometrics , Vol. 46, No. 3 (Sep., 1990), pp. 673-687
2. [2] C. Laveille, M. Lavielle, K. Chatel, P. Jacqmin; Evaluation of the PK and PK-PD libraries of MONOLIX: A comparison with NONMEM; PAGE 2008