Variational Bayes: DEBUG2 - Testing with 15 Groups (vs 6)

2026-01-13 10:20:19

Contents

```r
# ============================================================================
# MODEL TYPE SELECTION
# M1: Linear regression with   and  ² (residual variance)
#     - No random effects, no variance component

# M3: Hierarchical model with  , u (random effects),  ,
#     - Demonstrates variance component under-dispersion
#     - Two scenarios: 30 sparse groups vs 6 rich groups
model_type <- "M3"  # "M1" for linear regression, "M3" for hierarchical

# SHARED PARAMETERS (both scenarios)
n <- 300  # Total observations
p <- 4    # Number of fixed effects (beta_0, beta_1, beta_2, beta_3)

# True Parameter Values
tau_e_true <- 5
tau_u_true <- if (model_type == "M1") NULL else 0.5  # Lower = more variance between groups
beta_true  <- c(5, 0.5, 0.5, 0.5)

# Prior Hyperparameters
alpha_e <- 0.01
gamma_e <- 0.01
alpha_u <- 1.0   # Prior for tau_u centred at 0.5
gamma_u <- 2.0   # Mean = alpha_u / gamma_u = 1.0 / 2.0 = 0.5

# Gibbs Sampler Settings
run_gibbs    <- TRUE   # Set FALSE for quick testing, TRUE for full run
gibbs_iter   <- 5000
gibbs_burnin <- 1000

# Display Settings
RENDER_FUNCTIONS <- FALSE  # TRUE to show function code, FALSE to hide

# SCENARIO 1: 50 random effect levels (sparse groups) - M3 only
q_30levels  <- if (model_type == "M1") 0 else 50
nq_30levels <- if (model_type == "M1") 0 else 6
```

```r
# SCENARIO 2: 15 random effect levels (moderate groups) - M3 only
q_6levels  <- if (model_type == "M1") 0 else 15
nq_6levels <- if (model_type == "M1") 0 else 20

# ============================================================================
```

1 Scenario 1: Conditional on Model Type

```
cat(glue("\n=== Running {model_type} Analysis ===\n"))
```

```
## === Running M3 Analysis ===
```

```
q_s1  <- q_30levels
nq_s1 <- nq_30levels

X_s1 <- cbind(1, matrix(rnorm(n*(p-1)), nrow=n, ncol=p-1))

if (model_type == "M3") {
  u_true_s1 <- rnorm(q_s1, 0, 1/sqrt(tau_u_true))
  Z_s1 <- table(1:n, rep(1:q_s1, each=nq_s1))
  K_s1 <- diag(q_s1)
  linear_predictor_s1 <- X_s1 %*% beta_true + Z_s1 %*% u_true_s1
} else {
  u_true_s1 <- NULL
  Z_s1 <- matrix(0, nrow=n, ncol=1)
  K_s1 <- matrix(1)
  linear_predictor_s1 <- X_s1 %*% beta_true
}

residuals_true_s1 <- rnorm(n, 0, 1/sqrt(tau_e_true))
y_s1 <- as.vector(linear_predictor_s1 + residuals_true_s1)
```

```
scenario_name <- if (model_type == "M3") "SCENARIO 1: 30 levels (10 obs each)" else "SCENARIO 1: Linear
cat(glue("\n=== {scenario_name} ===\n"))
```

```
## === SCENARIO 1: 30 levels (10 obs each) ===
```

```
results_s1 <- run_vb_algorithm(
  X          = X_s1,
  Z          = Z_s1,
  y          = y_s1,
  K          = K_s1,
  p          = p,
  q          = q_s1,
  n          = n,
  alpha_e    = alpha_e,
  gamma_e    = gamma_e,
  alpha_u    = alpha_u,
  gamma_u    = gamma_u,
  model_type = model_type
)
```

```
##
## === Iteration 1 ===
##   E_tau_e=1.000000, E_tau_u=0.500000
##   mu_betau[1:3]: [5.2560, 0.4878, 0.5245]
##   Sigma_betau[1,1]=0.043501, Sigma_betau[p+1,p+1]=0.194512
##   mu_u: [0.3644, -0.7775, 0.2906, 0.0586, -1.6325, 0.3059]
##   ||mu_u||^2 = 77.329520
##   quad_form = mu_u' * K_inv * mu_u = 77.329520
##   trace_u = tr(K_inv * Sigma_uu) = 9.620668
##   a_u_new = 26.0000, b_u_new = 45.4751
##   E[tau_u] = a_u/b_u = 0.571742
##   Change in E[tau_u]: 0.071742
##
## === Iteration 2 ===
##   E_tau_e=2.906788, E_tau_u=0.571742
##   mu_betau[1:3]: [5.2561, 0.4898, 0.5277]
##   Sigma_betau[1,1]=0.036186, Sigma_betau[p+1,p+1]=0.090817
##   mu_u: [0.3828, -0.8145, 0.3046, 0.0619, -1.7114, 0.3217]
##   ||mu_u||^2 = 85.120362
##   quad_form = mu_u' * K_inv * mu_u = 85.120362
##   trace_u = tr(K_inv * Sigma_uu) = 4.500779
##   a_u_new = 26.0000, b_u_new = 46.8106
##   E[tau_u] = a_u/b_u = 0.555430
##   Change in E[tau_u]: -0.016311
##
## === Iteration 3 ===
##   E_tau_e=4.346208, E_tau_u=0.555430
##   mu_betau[1:3]: [5.2562, 0.4902, 0.5285]
##   Sigma_betau[1,1]=0.036814, Sigma_betau[p+1,p+1]=0.073784
##   mu_u: [0.3872, -0.8234, 0.3079, 0.0627, -1.7304, 0.3256]
##   ||mu_u||^2 = 87.054032
##   quad_form = mu_u' * K_inv * mu_u = 87.054032
##   trace_u = tr(K_inv * Sigma_uu) = 3.661774
##   a_u_new = 26.0000, b_u_new = 47.3579
##   E[tau_u] = a_u/b_u = 0.549011
##   Change in E[tau_u]: -0.006419
## Converged at iteration 9
## Max relative change: 6.10e-06
```

```
exact_s1 <- exact_linear_posterior(
  X           = X_s1,
  y           = y_s1,
  prior_mu    = rep(0, p),
  prior_sigma = 10,
  prior_a     = alpha_e,
  prior_b     = gamma_e
)

cat("\nExact posterior parameters:\n")
```

```
##
## Exact posterior parameters:
```

```
cat("E[beta]:", exact_s1$mu_n, "\n")
```

```
## E[beta]: 5.253169 0.4526869 0.4673524 0.4262505
```

```
cat("E[tau_e]:", exact_s1$a_n / exact_s1$b_n, "\n")
```

```
## E[tau_e]: 0.5052656
```

```
cat("\nRunning Laplace approximation...\n")
```

```
##
## Running Laplace approximation...
```

```r
laplace_s1 <- run_laplace(
  X          = X_s1,
  Z          = Z_s1,
  y          = y_s1,
  p          = p,
  q          = q_s1,
  n          = n,
  alpha_e    = alpha_e,
  gamma_e    = gamma_e,
  alpha_u    = alpha_u,
  gamma_u    = gamma_u,
  model_type = model_type
)

cat("Laplace MAP estimates:\n")
```

```
## Laplace MAP estimates:
```

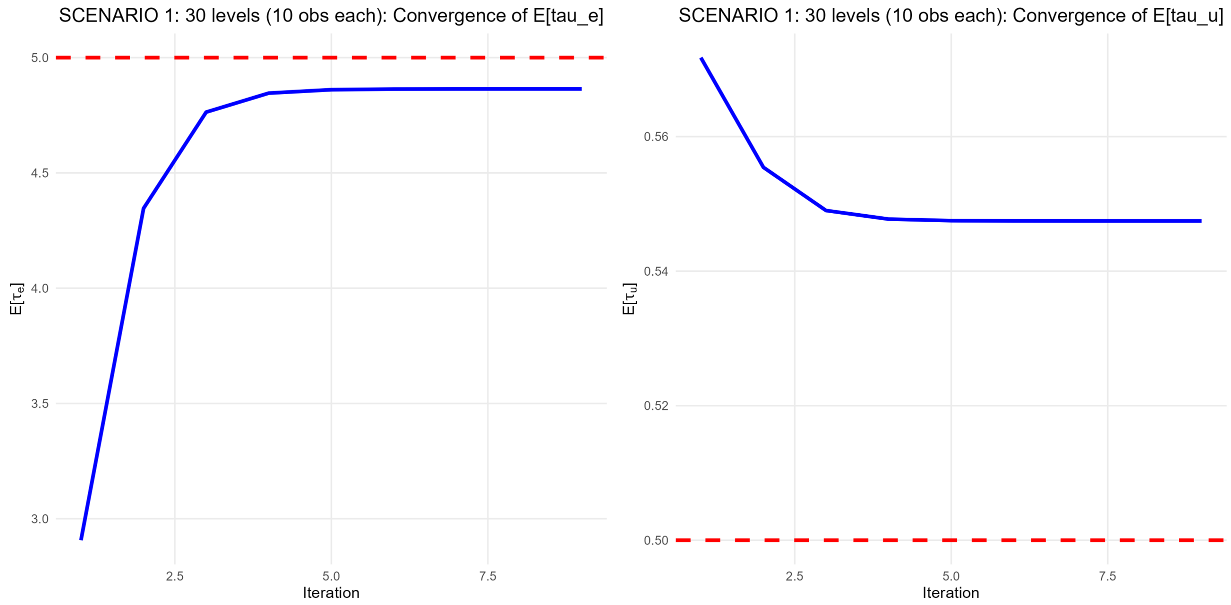
```r
if (model_type == "M1") {
  cat("beta:", laplace_s1$theta_map[1:p], "\n")
  cat("tau_e:", exp(laplace_s1$theta_map[p + 1]), "\n")
} else {
  cat("beta:", laplace_s1$theta_map[1:p], "\n")
  cat("tau_e:", exp(laplace_s1$theta_map[p + q_s1 + 1]), "\n")
  cat("tau_u:", exp(laplace_s1$theta_map[p + q_s1 + 2]), "\n")
}
```

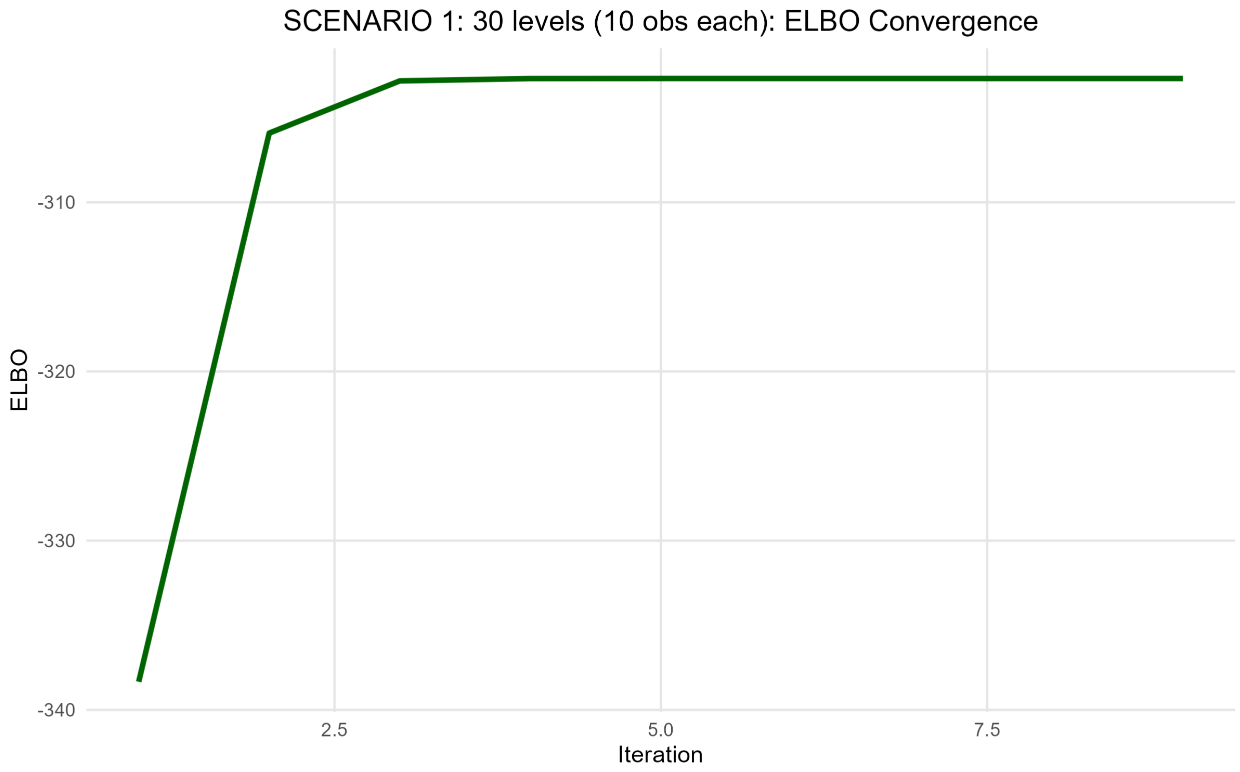
```
## beta: 5.254224 0.4904948 0.5289162 0.4710682
## tau_e: 5.853474
## tau_u: 0.5431257
```

```r
# Save results for VI-COMPARE
if (model_type == "M3") {
  saveRDS(list(
    vb = results_s1,
    laplace = laplace_s1,
    data = list(X = X_s1, Z = Z_s1, y = y_s1),
    truth = list(tau_u = tau_u_true, tau_e = tau_e_true),
    config = list(q = q_s1, nq = nq_30levels)
  ), file.path("..", "results", "debug2_s1.rds"))
}
```

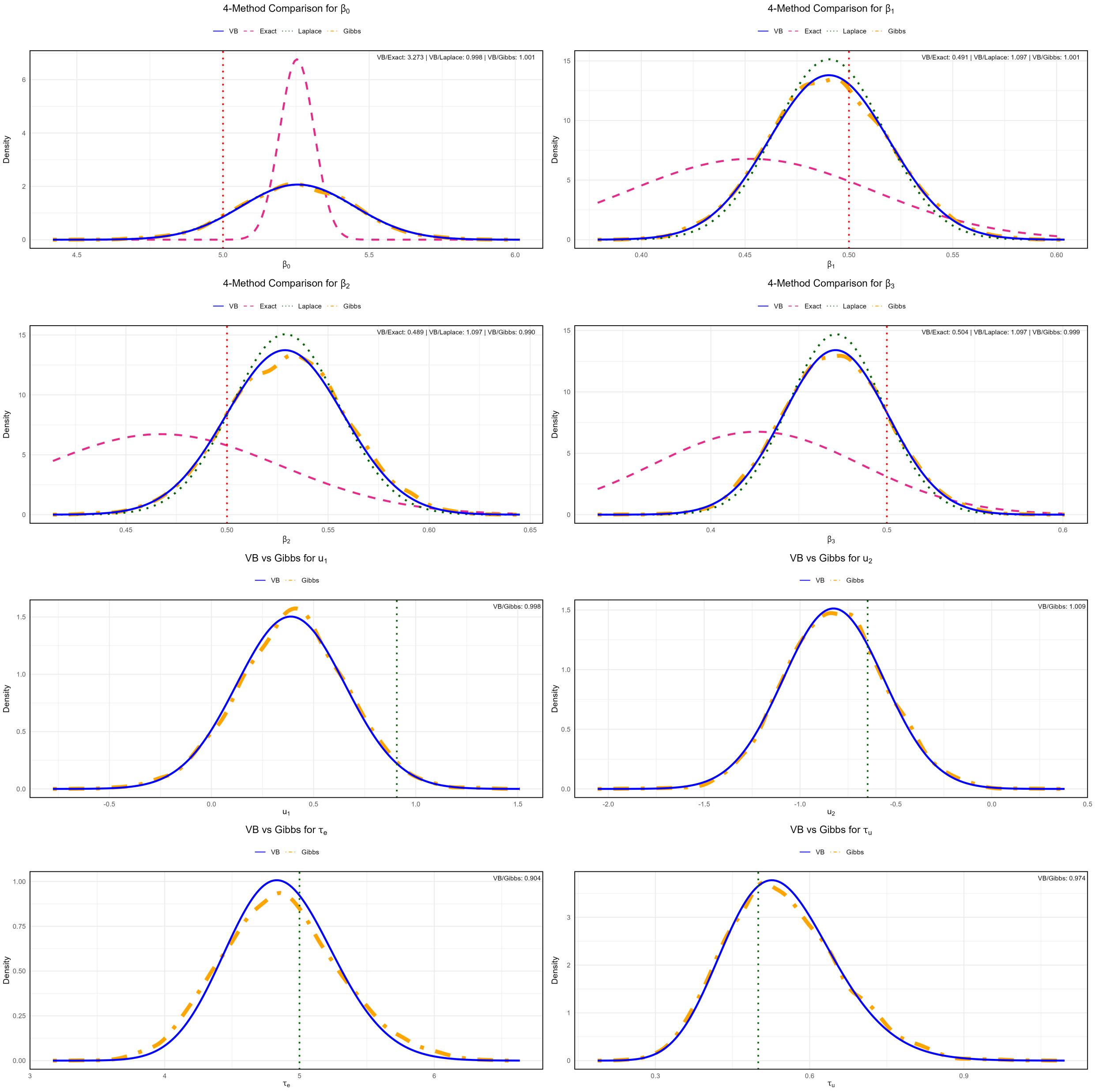
```r
if (run_gibbs) {
  cat("\nRunning Gibbs sampler...\n")
  gibbs_s1 <- run_gibbs_sampler(
    X          = X_s1,
    Z          = Z_s1,
    y          = y_s1,
    p          = p,
    q          = q_s1,
    n          = n,
    alpha_e    = alpha_e,
    gamma_e    = gamma_e,
    alpha_u    = alpha_u,
    gamma_u    = gamma_u,
    model_type = model_type,
    n_iter     = gibbs_iter,
    n_burnin   = gibbs_burnin
  )

  cat("Gibbs posterior means:\n")
  cat("beta:", colMeans(gibbs_s1[, 1:p]), "\n")
  cat("tau_e:", mean(gibbs_s1[, "tau_e"]), "\n")
  if (model_type == "M3") {
    cat("tau_u:", mean(gibbs_s1[, "tau_u"]), "\n")
  }

  gibbs_tau_e_s1 <- gibbs_s1[, "tau_e"]
  gibbs_tau_u_s1 <- if (model_type == "M3") gibbs_s1[, "tau_u"] else NULL
} else {
  gibbs_s1 <- NULL
  gibbs_tau_e_s1 <- NULL
  gibbs_tau_u_s1 <- NULL
}
```

```
##
## Running Gibbs sampler...
## Gibbs posterior means:
## beta: 5.252275 0.4902334 0.5297066 0.4710027
## tau_e: 4.86065
## tau_u: 0.5483044
```





```
## M3: 8-panel ggplot saved
```


4-Method Comparison for β0
VB Exact Laplace Gibbs
VB/Exact: 3.273 | VB/Laplace: 0.998 | VB/Gibbs: 1.001
Density
0 2 4 6
4.5 5.0 5.5 6.0
β0
4-Method Comparison for β1
VB Exact Laplace Gibbs
VB/Exact: 0.491 | VB/Laplace: 1.097 | VB/Gibbs: 1.001
Density
0 5 10 15
0.40 0.45 0.50 0.55 0.60
β1
4-Method Comparison for β2
VB Exact Laplace Gibbs
VB/Exact: 0.489 | VB/Laplace: 1.097 | VB/Gibbs: 0.990
Density
0 5 10 15
0.45 0.50 0.55 0.60 0.65
β2
4-Method Comparison for β3
VB Exact Laplace Gibbs
VB/Exact: 0.504 | VB/Laplace: 1.097 | VB/Gibbs: 0.999
Density
0 5 10 15
0.4 0.5 0.6
β3
VB vs Gibbs for u1
VB Gibbs
VB/Gibbs: 0.998
Density
0.0 0.5 1.0 1.5
-0.5 0.0 0.5 1.0 1.5
u1
VB vs Gibbs for u2
VB Gibbs
VB/Gibbs: 1.009
Density
0.0 0.5 1.0 1.5
-2.0 -1.5 -1.0 -0.5 0.0 0.5
u2
VB vs Gibbs for τe
VB Gibbs
VB/Gibbs: 0.904
Density
0.00 0.25 0.50 0.75 1.00
3 4 5 6
τe
VB vs Gibbs for τu
VB Gibbs
VB/Gibbs: 0.974
Density
0 1 2 3
0.3 0.6 0.9
τu

2 Scenario 2: Conditional on Model Type (M3 only)

```
if (model_type == "M3") {
  q_s2      <- q_6levels
  nq_s2     <- nq_6levels
  u_true_s2 <- rnorm(q_s2, 0, 1/sqrt(tau_u_true))

  X_s2 <- cbind(1, matrix(rnorm(n*(p-1)), nrow=n, ncol=p-1))
  Z_s2 <- table(1:n, rep(1:q_s2, each=nq_s2))
  K_s2 <- diag(q_s2)

  linear_predictor_s2 <- X_s2 %*% beta_true + Z_s2 %*% u_true_s2
  residuals_true_s2   <- rnorm(n, 0, 1/sqrt(tau_e_true))
  y_s2 <- as.vector(linear_predictor_s2 + residuals_true_s2)

  cat("\n=== DEBUG: Scenario 2 Data Generation ===\n")
  cat(sprintf("True tau_u = %.4f, True tau_e = %.4f\n", tau_u_true, tau_e_true))
  cat(sprintf("True random effects u: mean=%.4f, sd=%.4f\n", mean(u_true_s2), sd(u_true_s2)))
  cat(sprintf("Range of u: [%.4f, %.4f]\n", min(u_true_s2), max(u_true_s2)))
  cat(sprintf("Empirical var(u) = %.4f, theoretical = %.4f\n", var(u_true_s2), 1/tau_u_true))
} else {
  cat("\nScenario 2 skipped (M3 only)\n")
}
```

```
##
## === DEBUG: Scenario 2 Data Generation ===
## True tau_u = 0.5000, True tau_e = 5.0000
## True random effects u: mean=0.5714, sd=1.4670
## Range of u: [-1.6816, 3.0288]
## Empirical var(u) = 2.1520, theoretical = 2.0000
```

```
cat("\n=== SCENARIO 2: 15 levels (20 obs each) ===\n")
```

```
##
## === SCENARIO 2: 15 levels (20 obs each) ===
```

```
results_s2 <- run_vb_algorithm(
  X          = X_s2,
  Z          = Z_s2,
  y          = y_s2,
  K          = K_s2,
  p          = p,
  q          = q_s2,
  n          = n,
  alpha_e    = alpha_e,
  gamma_e    = gamma_e,
  alpha_u    = alpha_u,
  gamma_u    = gamma_u,
  model_type = model_type
)
```

```
##
## === Iteration 1 ===
##   E_tau_e=1.000000, E_tau_u=0.500000
##   mu_betau[1:3]: [5.5476, 0.5208, 0.5128]
##   Sigma_betau[1,1]=0.136857, Sigma_betau[p+1,p+1]=0.178963
##   mu_u: [-0.9498, 0.7178, 1.7779, -1.5672, 2.4897, -0.2437]
##   ||mu_u||^2 = 29.765605
##   quad_form = mu_u' * K_inv * mu_u = 29.765605
##   trace_u = tr(K_inv * Sigma_uu) = 2.690997
##   a_u_new = 8.5000, b_u_new = 18.2283
##   E[tau_u] = a_u/b_u = 0.466308
##   Change in E[tau_u]: -0.033692
##
## === Iteration 2 ===
##   E_tau_e=4.170146, E_tau_u=0.466308
##   mu_betau[1:3]: [5.5464, 0.5168, 0.5167]
##   Sigma_betau[1,1]=0.143812, Sigma_betau[p+1,p+1]=0.154121
##   mu_u: [-0.9686, 0.7335, 1.8129, -1.5985, 2.5409, -0.2493]
##   ||mu_u||^2 = 30.968755
##   quad_form = mu_u' * K_inv * mu_u = 30.968755
##   trace_u = tr(K_inv * Sigma_uu) = 2.313446
##   a_u_new = 8.5000, b_u_new = 18.6411
##   E[tau_u] = a_u/b_u = 0.455982
##   Change in E[tau_u]: -0.010326
##
## === Iteration 3 ===
##   E_tau_e=5.153037, E_tau_u=0.455982
##   mu_betau[1:3]: [5.5464, 0.5166, 0.5170]
##   Sigma_betau[1,1]=0.146889, Sigma_betau[p+1,p+1]=0.155241
##   mu_u: [-0.9697, 0.7344, 1.8151, -1.6004, 2.5440, -0.2497]
##   ||mu_u||^2 = 31.043389
##   quad_form = mu_u' * K_inv * mu_u = 31.043389
##   trace_u = tr(K_inv * Sigma_uu) = 2.329947
##   a_u_new = 8.5000, b_u_new = 18.6867
##   E[tau_u] = a_u/b_u = 0.454870
##   Change in E[tau_u]: -0.001112
## Converged at iteration 7
## Max relative change: 1.27e-06
```

```r
exact_s2 <- exact_linear_posterior(
  X           = X_s2,
  y           = y_s2,
  prior_mu    = rep(0, p),
  prior_sigma = 10,
  prior_a     = alpha_e,
  prior_b     = gamma_e
)

cat("\nExact posterior parameters:\n")
```

```
##
## Exact posterior parameters:
```

```r
cat("E[beta]:", exact_s2$mu_n, "\n")
```

```
## E[beta]: 5.602548 0.7110188 0.3240186 0.4700695
```

```r
cat("E[tau_e]:", exact_s2$a_n / exact_s2$b_n, "\n")
```

```
## E[tau_e]: 0.4550584
```

```r
cat("\nRunning Laplace approximation...\n")
```

```
##
## Running Laplace approximation...
```

```r
laplace_s2 <- run_laplace(
  X          = X_s2,
  Z          = Z_s2,
  y          = y_s2,
  p          = p,
  q          = q_s2,
  n          = n,
  alpha_e    = alpha_e,
  gamma_e    = gamma_e,
  alpha_u    = alpha_u,
  gamma_u    = gamma_u,
  model_type = model_type
)

cat("Laplace MAP estimates:\n")
```

```
## Laplace MAP estimates:
```

```r
cat("beta:", laplace_s2$theta_map[1:p], "\n")
```

```
## beta: 5.537657 0.5164583 0.5170766 0.4772193
```

```r
cat("tau_e:", exp(laplace_s2$theta_map[p + q_s2 + 1]), "\n")
```

```
## tau_e: 5.527687
```

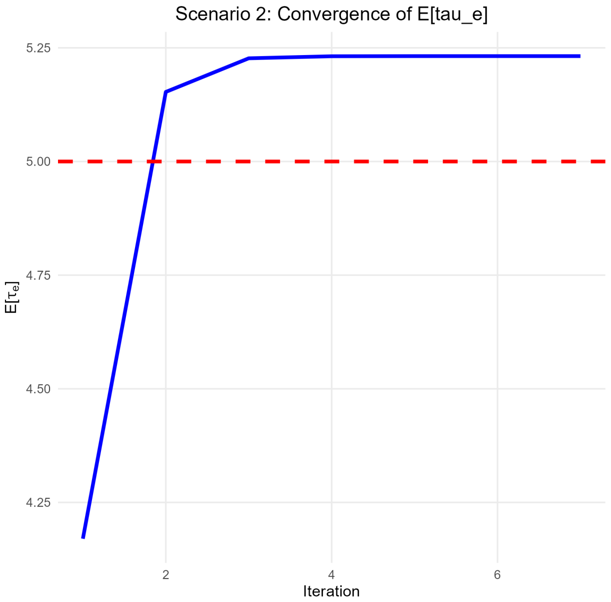
```r
cat("tau_u:", exp(laplace_s2$theta_map[p + q_s2 + 2]), "\n")
```

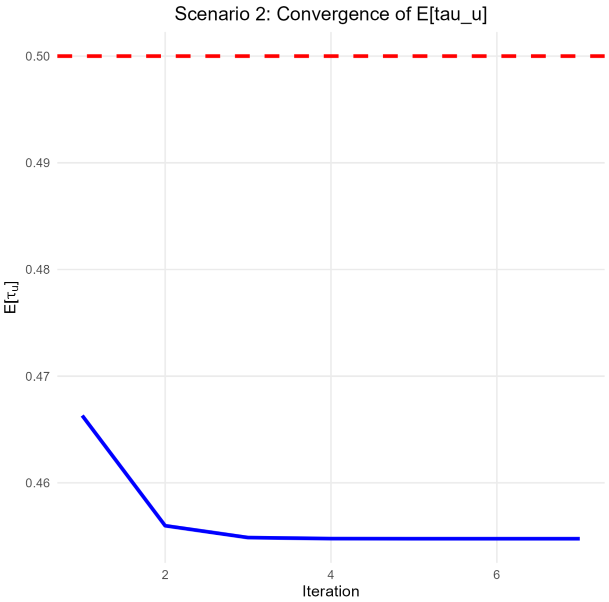
```
## tau_u: 0.4275906
```

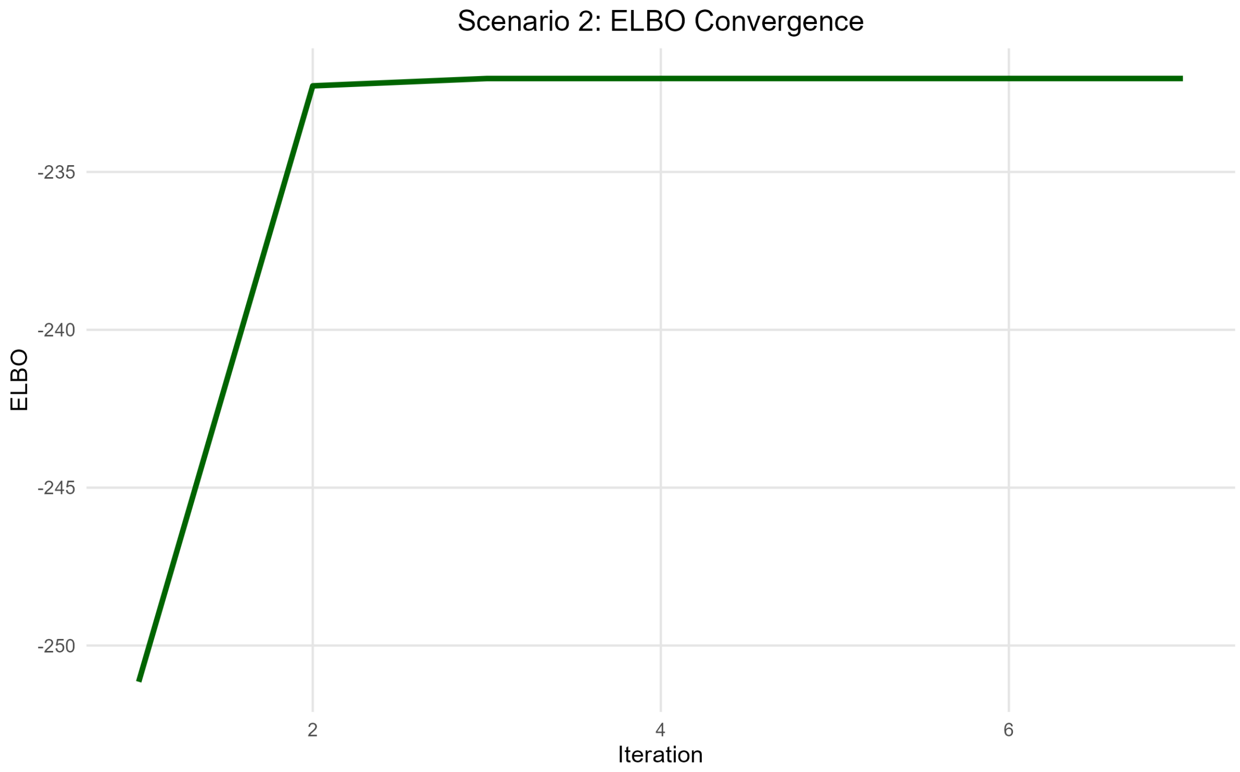
```r
# Save results for VI-COMPARE
saveRDS(list(
  vb = results_s2,
  laplace = laplace_s2,
  data = list(X = X_s2, Z = Z_s2, y = y_s2),
  truth = list(tau_u = tau_u_true, tau_e = tau_e_true),
  config = list(q = q_s2, nq = nq_6levels)
), file.path("..", "results", "debug2_s2.rds"))
```

```r
if (run_gibbs) {
  cat("\nRunning Gibbs sampler...\n")
  gibbs_s2 <- run_gibbs_sampler(
    X          = X_s2,
    Z          = Z_s2,
    y          = y_s2,
    p          = p,
    q          = q_s2,
    n          = n,
    alpha_e    = alpha_e,
    gamma_e    = gamma_e,
    alpha_u    = alpha_u,
    gamma_u    = gamma_u,
    model_type = model_type,
    n_iter     = gibbs_iter,
    n_burnin   = gibbs_burnin
  )

  cat("Gibbs posterior means:\n")
  cat("beta:", colMeans(gibbs_s2[, 1:p]), "\n")
  cat("tau_e:", mean(gibbs_s2[, "tau_e"]), "\n")
  cat("tau_u:", mean(gibbs_s2[, "tau_u"]), "\n")

  gibbs_tau_e_s2 <- gibbs_s2[, "tau_e"]
  gibbs_tau_u_s2 <- gibbs_s2[, "tau_u"]
} else {
  gibbs_s2 <- NULL
  gibbs_tau_e_s2 <- NULL
  gibbs_tau_u_s2 <- NULL
}
```

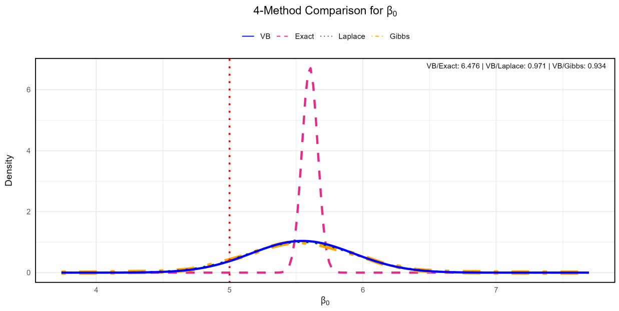
```
##
## Running Gibbs sampler...
## Gibbs posterior means:
## beta: 5.535208 0.5167041 0.5171746 0.4771759
## tau_e: 5.225994
## tau_u: 0.4543885
```



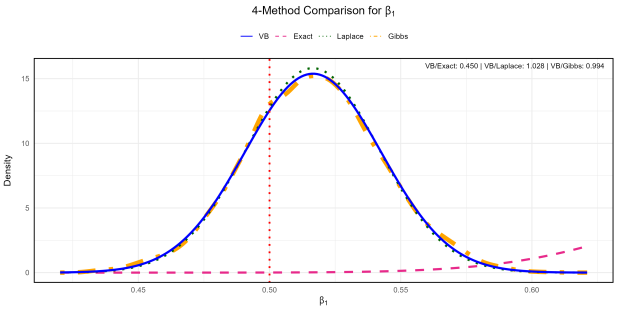




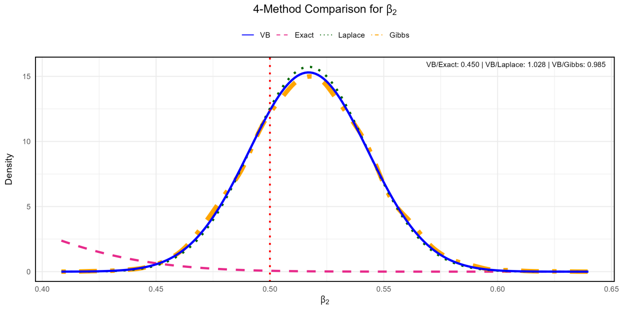
```
## 6 levels: 8-panel ggplot saved
```



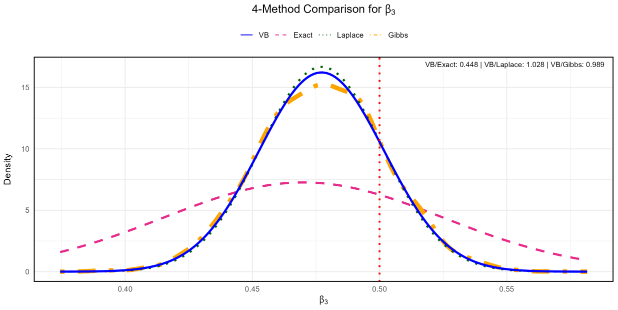
4-Method Comparison for β_0



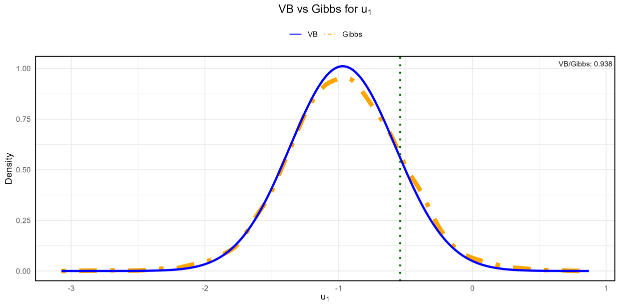
4-Method Comparison for β_1



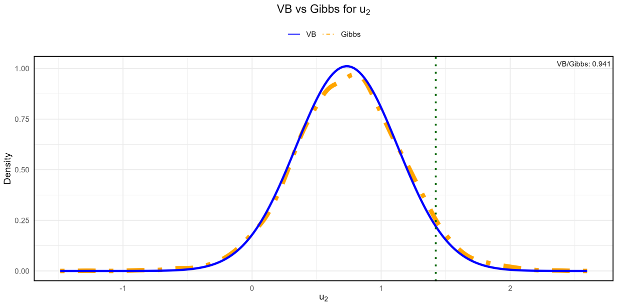
4-Method Comparison for β_2



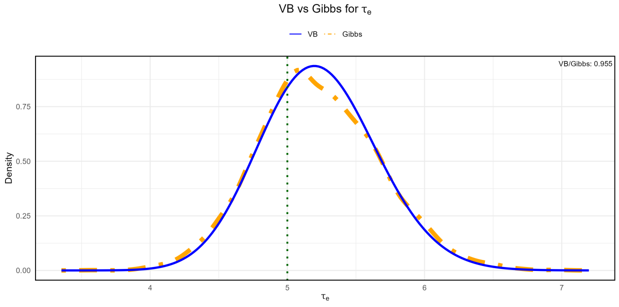
4-Method Comparison for β_3



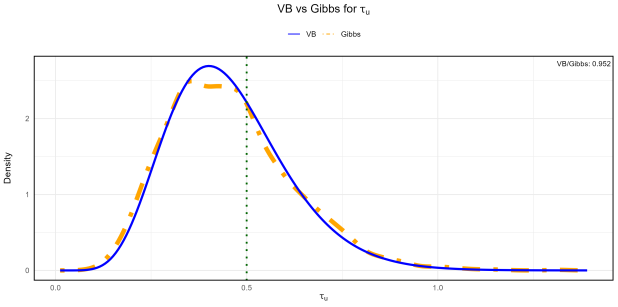
VB vs Gibbs for u_1



VB vs Gibbs for u_2



VB vs Gibbs for τ_e



VB vs Gibbs for τ_u

3 Comparison Between Scenarios (M3 only)

```
comparison_df <- data.frame(
  Parameter   = c("E[tau_e]", "E[tau_u]", "sigma^2_e", "sigma^2_u"),
  True_Value  = c(tau_e_true, tau_u_true, 1/tau_e_true, 1/tau_u_true),
  Scenario_30 = c(results_s1$E_tau_e, results_s1$E_tau_u,
                  1/results_s1$E_tau_e, 1/results_s1$E_tau_u),
  Scenario_6  = c(results_s2$E_tau_e, results_s2$E_tau_u,
                  1/results_s2$E_tau_e, 1/results_s2$E_tau_u)
)

print(comparison_df)
```

```
##   Parameter True_Value Scenario_30 Scenario_6
## 1  E[tau_e]        5.0   4.8640457  5.2318118
## 2  E[tau_u]        0.5   0.5474557  0.4547621
## 3 sigma^2_e        0.2   0.2055902  0.1911384
## 4 sigma^2_u        2.0   1.8266317  2.1989520
```

```
cat("\nUnder-dispersion in tau_u estimates:\n")
```

```
##
## Under-dispersion in tau_u estimates:
```

```
cat("30 levels: VB tau_u =", round(results_s1$E_tau_u, 4),
    "vs True =", tau_u_true,
    "(ratio:", round(results_s1$E_tau_u / tau_u_true, 4), ")\n")
```

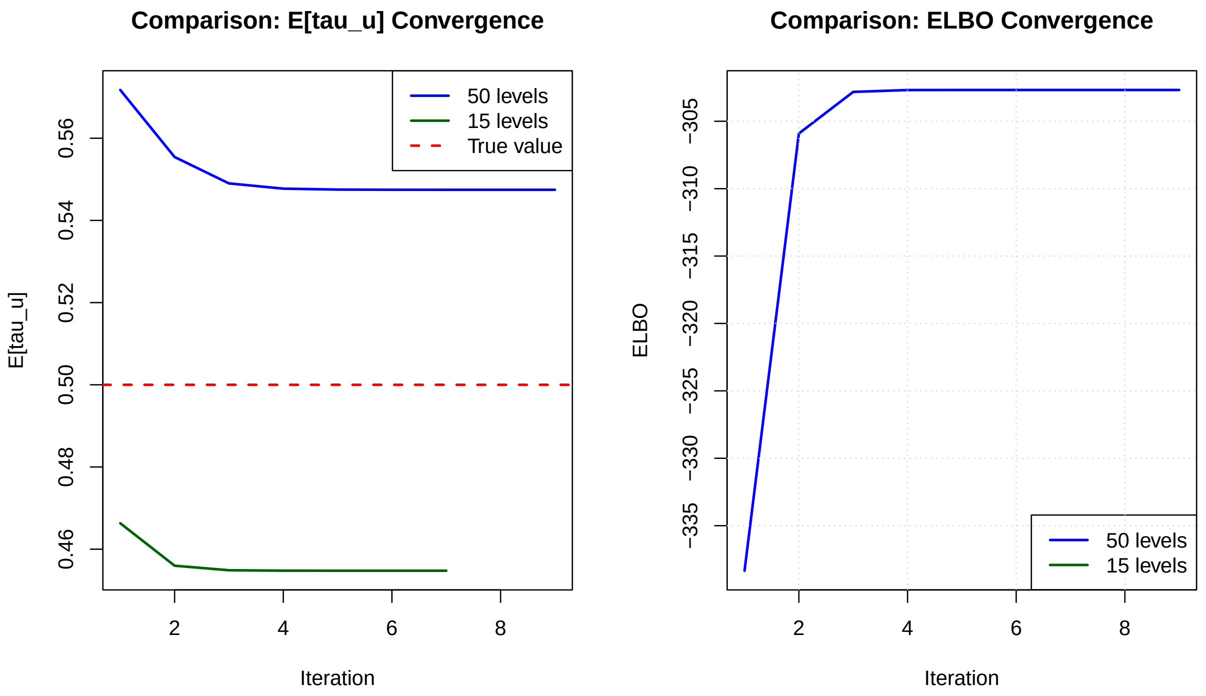
```
## 30 levels: VB tau_u = 0.5475 vs True = 0.5 (ratio: 1.0949 )
```

```
cat("6 levels:  VB tau_u =", round(results_s2$E_tau_u, 4),
    "vs True =", tau_u_true,
    "(ratio:", round(results_s2$E_tau_u / tau_u_true, 4), ")\n")
```

```
## 6 levels:  VB tau_u = 0.4548 vs True = 0.5 (ratio: 0.9095 )
```

```
par(mfrow = c(1, 2))

max_len <- max(length(results_s1$E_tau_u_history),
               length(results_s2$E_tau_u_history))

plot(
  1:length(results_s1$E_tau_u_history),
  results_s1$E_tau_u_history,
  type = 'l',
  lwd  = 2,
  col  = 'blue',
  xlab = 'Iteration',
  ylab = 'E[tau_u]',
  main = 'Comparison: E[tau_u] Convergence',
  xlim = c(1, max_len),
  ylim = range(c(results_s1$E_tau_u_history, results_s2$E_tau_u_history, tau_u_true)))
lines(1:length(results_s2$E_tau_u_history),
      results_s2$E_tau_u_history,
      col = 'darkgreen',
      lwd = 2)
```

```
abline(h = tau_u_true, col = 'red', lty = 2, lwd = 2)
legend('topright',
       legend = c('50 levels', '15 levels', 'True value'),
       col    = c('blue', 'darkgreen', 'red'),
       lty    = c(1, 1, 2),
       lwd    = 2)

plot(
  1:length(results_s1$elbo_history),
  results_s1$elbo_history,
  type = 'l',
  lwd  = 2,
  col  = 'blue',
  xlab = 'Iteration',
  ylab = 'ELBO',
  main = 'Comparison: ELBO Convergence',
  xlim = c(1, max_len))
lines(1:length(results_s2$elbo_history),
      results_s2$elbo_history,
      col = 'darkgreen',
      lwd = 2)
legend('bottomright',
       legend = c('50 levels', '15 levels'),
       col    = c('blue', 'darkgreen'),
       lty    = 1,
       lwd    = 2)
grid()
```


Comparison: E[tau_u] Convergence
E[tau_u]
Iteration
50 levels
15 levels
True value
Comparison: ELBO Convergence
ELBO
Iteration
50 levels
15 levels