Online appendix to
"`fereg`: Fast high-dimensional **f**ixed-**e**ffects **reg**ressions in Stata"

Johannes F. Schmieder
Department of Economics, Boston University

November 2026

B Absorption methods: exact numerical rules

This appendix accompanies section 5 of the article. Appendix B.1 lists the methods that `solver(auto)` selects, appendix B.2 states the accuracy check and the margins of the iterative methods, and the remaining sections give the exact rules of each method, including the methods that can be requested through `solver()` (appendix B.8). Equations and sections numbered without a letter prefix are those of the article. Throughout, τ denotes the value of `tolerance()`, $\tau_\circ = \max(\tau/10, 10^{-14})$, and $\tau_* = \max(\tau/100, 10^{-14})$; either is replaced by τ if it would exceed τ.

B.1 Methods selected by `solver(auto)`

Table B.1 summarizes the methods described in sections 5.2 and 5.4 to 5.7 of the article, with the target at which each method stops its own iterations and what happens if its result fails the check.

B.2 Accuracy check and iteration margins

Let v be a variable, $r = v - D\hat{\alpha}$ a candidate residual reconstructed from candidate effects $\hat{\alpha}$, and $s = \max(\|W^{1/2}v\|, \|W^{1/2}r\|)$, where $\|\cdot\|$ is the Euclidean length. For group g of absorbed term ℓ, let $D_{\ell g}$ hold the columns of D that belong to the group, restricted to the observations in the group: an intercept column and one column for each slope variable. Let $W_{\ell g}$ and $r_{\ell g}$ be the corresponding weights and residuals, and let $Q_{\ell g}$ be a matrix with orthonormal columns that span the columns of $W_{\ell g}^{1/2} D_{\ell g}$. The check statistic is

$$c(r) = \max_{\ell, g} \frac{\|Q_{\ell g}' W_{\ell g}^{1/2} r_{\ell g}\|}{s},$$

and `fereg` accepts the residuals if $c(r) \le \tau$ for the outcome and for every covariate. If $s = 0$, only a zero projection is accepted. For a group of an intercept-only effect, $\|Q_{\ell g}' W_{\ell g}^{1/2} r_{\ell g}\| = \sqrt{n_g}\,|\bar{r}_g|$, which gives (6). The statistic is computed from the original data and weights, not from quantities produced inside a solver.

Table B.1: Methods selected by `solver(auto)`

Design	Method	Iteration target	If the check fails
`fereg`			
One absorbed term	Subtract group means, or fit each group's own regression on its slopes (section 5.2)	Exact in one pass	Repeat the pass
Two categorical effects	Eliminate the effect with more groups; conjugate gradients on the reduced system, preconditioned by group sizes and, after 16 steps, by combinatorial multigrid; direct solution if cheaper (section 5.4)	$\tau/2$; one correction at $\tau/100$	Alternating projections
Three or more categorical effects	LSMR, preconditioned by group sizes and, if many iterations remain, by the pairwise preconditioner; with three unweighted effects, a short pilot of alternating projections (section 5.5)	$\tau/10$; one correction at $\tau/100$	Alternating projections
Two terms with slopes	Eliminate one term; conjugate gradients on the other (section 5.6)	$\tau/4$	Error
Three or more terms with slopes	Alternating projections (section 5.3)	τ	Error
`feppml`, within each iteration			
One absorbed term	As for `fereg`	Exact in one pass	As for `fereg`
Two or more absorbed terms	Eliminate the term with the most coefficients; conjugate gradients for the remaining effects and the coefficient update together (section 5.7)	$\tau_k/4$	Absorb each variable separately; error if that fails
Two categorical effects	As above, with a multigrid preconditioner that is built when needed and reused across iterations	$\tau_k/4$	As above

Notes: τ is the tolerance set by `tolerance()`. The iteration target is the value at which a method stops its own iterations. Linear absorptions must then pass (6) at τ; the smaller targets leave a margin for reconstruction error. For `feppml`, τ_k tightens from 10^{-6} to $\min(10^{-12}, \tau/10)$. Joint steps require reconstructed fixed-effect moments at $\tau_k/2$ and independently computed covariate scores at τ_k; the final weighted fit checks separate absorptions at the strict tolerance (section 5.7). Joint steps require at least one covariate and sufficient workspace. Optional preconditioners are built only if their workspace fits in memory. An error stops the command without posting results.

The iteration targets in table B.1 are often smaller than τ. Evaluating the check requires reconstructing r from the estimated effects and a full pass over the data, which is too expensive to repeat at every step. An iterative method instead tracks its progress with an internal measure that it updates almost for free. For conjugate gradients on the reduced system (7), the measure is the check statistic for the groups of the remaining effect, computed from a running estimate of the residual that the method updates at each step rather than recomputes. For LSMR, it is an estimate of how far the current solution is from satisfying the normal equations as a whole. These measures follow the check closely but not exactly: a running estimate drifts away from the true residual as rounding errors accumulate, a measure for the system as a whole does not bound the worst group, and reconstructing r adds rounding errors of its own. Aiming below τ, at $\tau/2$ for conjugate gradients and $\tau_\circ$ for LSMR, uses additional iterations to reduce the risk that the reconstructed result fails the check. The cost depends on the conditioning of the problem, and the margins do not guarantee acceptance. If the check fails nonetheless, one correction can aim at τ_*, subject to the remaining iteration budget. The margins affect only when a method stops, not the standard it must meet.

B.3 Two categorical effects

In the notation of (7), the worker effects follow from the firm effects as $a_1 = N_1^{-1}(D_1'Wv - Ca_2)$, and the residual is $\tilde{v} = v - D_1a_1 - D_2a_2$. Let c_j' be row j of C and $n_j = \sum_k C_{jk}$ the size of worker j. Then

$$S = \sum_j L_j, \qquad L_j = \operatorname{diag}(c_j) - \frac{c_j c_j'}{n_j}.$$

Each L_j is the Laplacian of a complete graph on the firms at which worker j is observed, with weight $C_{jk}C_{jl}/n_j$ on the link between firms k and l; its rows sum to zero because $\sum_k C_{jk} = n_j$. Conjugate gradients uses the preconditioner $M = N_2$ and stops when its estimate of the check statistic for the groups of the remaining effect falls below $\tau/2$. If the check then fails, one further run starts from the reconstructed residual and targets τ_*, provided that $\tau_* < \tau/2$; the final residuals receive no such run. Conjugate gradients may use three quarters of the limit set by `iterate()`, and the remaining quarter is reserved for alternating projections. If the check still fails, or if conjugate gradients breaks down numerically, `fereg` continues with alternating projections from the current residuals. If they do not pass the check within the limit, the command stops with an error.

If some variable has not converged after 16 steps, `fereg` builds a multigrid preconditioner (Koutis et al., 2011) for

$$\tilde{S} = \sum_{j:\, d_j \le 5} L_j + \sum_{j:\, d_j > 5} T_j,$$

where d_j is the number of firms at which worker j is observed and T_j is the Laplacian of a spanning tree on those firms, drawn at random with link weights chosen so that $\mathbb{E}(T_j) = L_j$ (Gao et al., 2026). Eliminating a worker observed at d_j firms links every pair of those firms, which gives $d_j(d_j - 1)/2$ links; the tree connects the same firms with $d_j - 1$ links and equals the exact links on average (Kyng and Sachdeva, 2016; Gao et al., 2026). Repeated observations of a worker at the same firm

are combined before the links are formed. Conjugate gradients then continues from the current solution with one multigrid cycle for $\tilde{S}$ in place of M^{-1}, while the system it solves remains S. The trees are drawn with a fixed seed, so the same command on the same data makes the same draws. Waiting 16 steps spares designs that converge quickly the cost of building the preconditioner, which is then reused for later calculations, such as the final residuals. `fereg` considers this step once per regression, while it absorbs the outcome and the covariates together, or the outcome alone under `memory(low)`.

Once multigrid has been applied, products with S are computed from the cells C_{jk} rather than from the observations.

Before conjugate gradients starts, `fereg` computes an upper bound on the number of arithmetic operations needed to form and factor the reduced matrices of all connected sets of groups and to reconstruct the residuals, for elimination of either effect. It uses the cheaper of the two direct solutions if the bound is at most four times the operation count of one conjugate-gradient step. A direct solution, once chosen, excludes the multigrid step, and its result must pass the same check.

B.4 Three or more categorical effects

LSMR is applied to $W^{1/2}D$ with right-hand side $W^{1/2}v$. With at least 256 fixed-effect coefficients, the preconditioner is

$$M^{-1} = \sum_i R_i' \, \Omega_i \, \hat{A}_i^{+} \, \Omega_i R_i,$$

where i runs over the connected sets of the graphs formed by each pair of absorbed effects, R_i selects the groups in connected set i, $A_i = R_i D'WDR_i'$ is the corresponding block of $D'WD$, and Ω_i is diagonal with entry $1/\sqrt{m_g}$ for group g, where m_g is the number of connected sets that contain g (Toselli and Widlund, 2005; Fischer and Schröder, 2026). After the sign of one effect's coefficients is reversed, A_i is a graph Laplacian. $\hat{A}_i^{+}$ approximates its pseudo-inverse: it eliminates the effect with more groups using sampled trees as above and factors the remaining Laplacian by approximate Cholesky factorization (Gao et al., 2026), or exactly if the remaining system has at most 24 groups. With fewer than 256 coefficients, $M = \mathrm{diag}(D'WD)$ throughout. Otherwise LSMR starts with $M = \mathrm{diag}(D'WD)$. From the third iteration on, each variable projects the iterations it still needs from its last two contraction factors of the normal-equation residual; once a projected total exceeds 40 iterations, the additive preconditioner is built once and every unfinished variable continues from its current iterate. LSMR runs until its convergence measure falls below $\tau_\circ$, using at most half of the limit set by `iterate()`, and the result is checked at τ. If the check fails, LSMR continues once from the rejected coefficients and targets τ_* with its remaining iterations. If the check still fails, alternating projections restart from the original variable with the remaining iterations and are checked at τ. If the workspace for LSMR does not fit in memory, `fereg` uses alternating projections directly.

With exactly three effects and unit weights, `fereg` first bounds the contraction of alternating projections from the overlap of each pair of effects. If the bound predicts convergence to τ_* within

four sweeps, `fereg` runs at most four sweeps, checks at τ after the predicted sweep, the sweep before it, and the fourth sweep, and skips LSMR if a check passes; the sweeps count against the LSMR limit.

B.5 Absorbed slopes and alternating projections

With exactly two absorbed terms and at least one slope, conjugate gradients targets $\tau/4$ for the coefficients of the term that is not eliminated, accepts the reconstructed result internally at $\tau/2$, and is then checked at τ. Alternating projections use symmetric sweeps and compute the check every four sweeps.

B.6 Final residuals

When residuals, saved fixed effects, or predictions that include the absorbed effects are requested, `fereg` computes and checks the final residuals with $\min(\tau, 10^{-14})$ in place of τ.

B.7 Poisson regression

Let $f = \min(10^{-12}, \tau/10)$. In iteration k of iteratively reweighted least squares, `feppml` absorbs with tolerance

$$\tau_k = \max\{f,\ \min(t_k,\ p_k,\ \tau_{k-1})\}, \qquad \tau_0 = p_1 = 10^{-6},$$

where $t_k = \exp\{(1-\lambda_k)\log 10^{-6} + \lambda_k \log f\}$ with $\lambda_k = \min\{(k-1)/7, 1\}$ tightens the tolerance over eight iterations. For $k > 1$, the progress target p_k equals f once the previous accepted iteration satisfies both outer convergence criteria; otherwise it equals $0.1\max(\Delta_{k-1}, \sigma_{k-1})$, bounded to $[f, 10^{-6}]$, where Δ is the relative change in the deviance and σ is the largest normalized score. Once the outer criteria hold, `feppml` solves the weighted regression with tolerance f and resumes iterating if the estimates move. Before this strict solve, `feppml` waits while the fit is still contracting: with m_k the larger of the relative changes in the linear predictor and in the coefficients between the last two accepted iterations, it enters the strict solve once $m_k^2/m_{k-1} \le \min(\tau, 10^{-8})$, or at once if $m_k/m_{k-1} > 1/2$, if fewer than three iterations remain, or after eight such deferrals.

In ordinary iterations with at least two absorbed terms and at least one covariate, `feppml` eliminates the term with the most coefficients exactly and solves for the remaining fixed-effect coefficients and the coefficient update δ jointly by conjugate gradients. The preconditioner is block diagonal, with the current weighted group sizes $\tilde{N}$ for the fixed effects and the Cholesky factor of $\widetilde{X}'H\widetilde{X}$ for δ, where $\widetilde{X}$ holds the covariates after the eliminated effect is removed and $H = \mathrm{diag}(h_i)$. The recurrence stops at $\tau_k/4$. Reconstructing the residual from the original inputs must then satisfy the fixed-effect moment check at $\tau_k/2$, and normalized scores for the covariates, computed separately from the original inputs, must be at most τ_k. This joint-step check does not require separate absorptions of every covariate. The final weighted fit does compute those absorptions and applies the accuracy check of appendix B.2 at the strict tolerance f. With exactly two categorical effects, a solve that reaches step 64 builds a multigrid preconditioner P for the current reduced

matrix, as for `fereg`, or rebuilds one built under earlier weights, and restarts from its current iterate. Later iterations apply $\tilde{N}^{1/2}P\tilde{N}^{1/2}$ to the fixed-effect block, with $\tilde{N}$ evaluated at the current weights. Before the final strict solve, `feppml` attempts to rebuild a preconditioner built under earlier weights with the current weights. If none exists, `feppml` solves for the first variable on the largest connected set without one and builds one if that solve needs more than 32 steps. The final solve uses the preconditioner for connected sets with at least 64 groups. A preconditioner is built or used only if its workspace fits in memory; if a rebuild does not fit, the earlier preconditioner can be kept. `feppml` attempts at most three constructions per regression, including those refused for lack of memory.

With one absorbed term, `feppml` uses the exact calculation of section 5.2. If the model has no covariates, or if the joint solve does not fit in memory, `feppml` absorbs the working response and the covariates separately, with conjugate gradients on the coefficients as in section 5.6 or with alternating projections. If the joint solve fails its check, `feppml` repeats the iteration once by absorbing every variable separately, using the methods of section 5.4 with two categorical effects, LSMR with three when its workspace fits, and the methods of section 5.6 or alternating projections otherwise. If that also fails, the command stops with an error.

B.8 Methods requested through `solver()`

For `fereg`, an explicitly requested method stops with an error if its result fails the check, rather than falling back on alternating projections. The one exception is `solver(cmg)`: if the multigrid preconditioner cannot be built or applied, it continues with plain conjugate gradients and reports that it did so. Because explicit LSMR has no fallback, its iteration target is $\tau/100$, floored at 10^{-15} but never above τ. `solver(cmg)` can absorb up to 64 additional fixed-effect columns alongside the two largest categorical effects, but not absorbed slopes.

For `feppml`, explicit `solver(lsmr)` requires at least two categorical effects, and `solver(cmg)` requires two or three; with three, the additional effect may have at most 64 non-base indicators. Both require the fast memory plan. Their final fit, which absorbs every covariate separately, uses its own absorption, which must pass the accuracy check. If combinatorial multigrid fails to converge, `feppml` reports this and falls back to the default methods; if the final fit fails its check, it first retries once with LSMR at a tighter tolerance. Lack of memory and unsupported designs remain errors. The stored results `e(feppml_solver_*)` record which of these steps ran.

B.9 Two convergence stress tests

These small, identified examples illustrate stopping-rule failures under deliberate numerical stress.[1]

[1] Comparisons use `fereg` 0.5.0, `reghdfe` 6.14.1, `ppmlhdfe` 2.3.4 and `fixest` 0.14.2 on one Mac with one worker. Scripts and the full grids, including warnings and refusals, are in `replication/certification_rhs_scale/`.

OLS and covariate units. For each $g = 1, \ldots, 100$, connect to $h = g$ and $h = g + 1$, wrapping 101 to 1. Each pair contains all four combinations of $u, v \in \{-1, 1\}$, giving 800 observations. Set $y = 2u + v$ and $x = s(A_g + B_h + u)$, where

$$A_g = 100[\cos(4\pi g/100) + \sin(10\pi g/100)],$$
$$B_h = 100[\sin(2\pi h/100) + \cos(6\pi h/100) + \sin(14\pi h/100)].$$

Absorbing g and h leaves su and $2u + v$, so the exact slope in common units is $s\hat{\beta} = 2$ and RSS is 800; an explicit-dummy SVD confirms this on the stored inputs. Changing s from 1 to 10^{-4} moves default `fixest`'s slope from 1.999999975 to 1.992224802 without an omission or convergence warning. At the smaller scale, its demeaned covariate gives 3.59×10^{-6} in (6), about 359 times the default threshold. Both `fereg` and `reghdfe` recover 2; `fixest` also does so with `fixef.tol=1e-11`.

PPML and small positive outcomes. Each of 100 groups contains $x = (0, 0, 1, 1)$ and $y = (0.5, 1.5, 0.5\delta, 1.5\delta)$, with $\delta = 10^{-10}$. The model $\mu_i = \exp(\alpha_g + \beta x_i)$ has finite optimum $\alpha_g = 0$ and $\beta = \log \delta = -23.02585093$; all outcomes are positive. Default `ppmlhdfe` and `fixest` report convergence at -21.27668584 and -21.20153855, while `feppml` recovers the optimum. After optimizing the group effects conditional on each returned slope, independently recomputed normalized scores are 1.77×10^{-5} and 1.87×10^{-5}, respectively, exceeding 10^{-8}. `ppmlhdfe` warns about low standardized outcomes. Its `tolerance(1e-12)` gives -23.02479761; `fixest` with `glm.tol=1e-12`, `fixef.tol=1e-11` and `glm.iter=100` gives -23.02584963. Neither example measures the frequency of such errors in empirical applications.

C Computational performance: additional results

This appendix accompanies section 7 of the article. It describes the benchmark protocol (appendix C.1), repeats the configurations of table 5 on a newer node (appendix C.2), reports how the accuracy of the coefficients depends on `tolerance()` (appendix C.3), summarizes the variation between repeated runs (appendix C.4), and lists every simulated configuration (appendix C.5).

C.1 Benchmark protocol

The measurements ran on dedicated compute nodes of Boston University's Shared Computing Cluster (Linux), using the published package of `fereg`. Five of the six nodes in the main simulated comparisons had 28-core Intel Xeon E5-2680 v4 processors (Broadwell); one had 28-core Xeon Gold 6132 processors (Skylake). Each comparison pairs commands on the same node. The Veneto panels use separate Gold 6132 nodes; the additional hardware comparison (appendix C.2) uses one 32-core Intel Xeon 6737P node (Granite Rapids). The versions of the comparison commands were held fixed throughout the measurements: `fixest` 0.14.2 ran in R 4.5.2, `PyFixest` in Python 3.12, and `FixedEffectModels.jl` 2.1.0 in Julia 1.11.7.

For `reghdfe` and `ppmlhdfe`, the default options are alternating projections accelerated by conjugate gradients, with symmetric Kaczmarz sweeps and a tolerance of 10^{-8}; `areg` used its `dfabsorb` option, which accounts for collinear absorbed variables. The strict tolerances were 10^{-11} for the fixed effects in `fixest` and `PyFixest`, with 10^{-10} for the PPML iterations, and 10^{-10} in `FixedEffectModels.jl`. Each package was limited to the assigned cores through its own setting: `set processors` in Stata, `setFixest_nthreads()` in R, `RAYON_NUM_THREADS` for `PyFixest`, and the number of Julia threads, with the linear-algebra libraries of every language on one thread. Because `FixedEffectModels.jl` compiles its code on first use, each of its runs first fit the same specification to a small dataset, as for the Python and R packages.

The order of the commands rotated across repetitions, and every command was restricted to the same processor cores. Each model and design ran on one node, and the same reference fit took between 0.79 and 1.47 seconds across the six nodes, so absolute times across designs, and across the tables and figures of the article, also reflect differences between nodes; ratios compare commands on the same node. All runs are retained.

C.2 A newer node

Table C.1 repeats the configurations of table 5 on one whole 32-core Intel Xeon 6737P node (Granite Rapids), with the same data, commands, and protocol and four cores per command. The main worker–firm comparisons ran on Broadwell nodes.

On Granite Rapids, `fereg` took 0.49 seconds in the baseline linear worker–firm model and `feppml` 2.22 seconds in PPML. `reghdfe` and `ppmlhdfe` took 15.4 and 21.1 times as long, `areg` 10.6 times, `fixest` 2.6 and 2.1 times, and `PyFixest` 2.2 and 2.1 times. The corresponding ratios for the Stata commands in the main comparisons were 18.0 and 16.1. Each ratio compares commands within one host; three to five repetitions do not measure variation across hosts.

C.3 Coefficient accuracy and `tolerance()`

The accuracy check (6) bounds the group means of the residuals, not the error in the coefficients (section 5.1). Table C.2 measures the coefficients directly. It fits three OLS datasets at tolerances from 10^{-12} to 10^{-6} and compares each fit with a fit of the same data at 10^{-14}: the sparse three-effect and clustered-mobility designs of the simulations and the Veneto largest connected set. At the default tolerance of 10^{-8}, the largest coefficient change was about 1.3×10^{-10} of its standard error, and at 10^{-7} about 8.4×10^{-9}. At 10^{-6}, the largest change reached 2×10^{-6} standard errors, on the Veneto sample, and in the sparse design with three effects the check of the covariate scores (section 6.4) failed, so the command stopped with an error instead of posting estimates. The error message names the loose tolerance and suggests a smaller one. The reported standard errors were unchanged in every completed fit. Tighter tolerances add few iterations down to 10^{-12} (in the sparse design with three effects, 33 instead of 23 at the default), but the reference fit at 10^{-14} needed 526.

Table C.1: Time and memory against Stata commands on a newer node

Inference and output	Median seconds (times slower than `fereg`)			Peak memory (GiB)		
	`fereg`	`reghdfe`/`ppmlhdfe`	`areg`	`fereg`	`reghdfe`/`ppmlhdfe`	`areg`
Linear regression						
IID	0.49	7.55 (15.4×)	5.21 (10.6×)	0.62	0.76	1.07
Robust	0.54	7.61 (14.1×)	5.46 (10.1×)	0.63	0.76	1.07
Clustered by worker	0.54	7.66 (14.1×)	5.59 (10.3×)[a]	0.64	0.76	1.08
IID, saved effects	0.60	8.36 (13.9×)[b]	—	0.71	0.76	—
Poisson regression (PPML)						
Robust	2.22	46.86 (21.1×)	—	1.30	1.27	—
Clustered by worker	2.25	45.98 (20.5×)	—	1.32	1.27	—
Robust, saved effects	2.38	47.48 (19.9×)[b]	—	1.30	1.27	—

Notes: Simulated worker–firm panel with one million observations, 16 controls, 20% mobility, and worker and firm effects; four cores for every command, all on one 32-core Intel Xeon 6737P node (Granite Rapids), with the same inputs and settings as table 5. Medians of five fresh-process runs; parentheses give the comparator's time divided by `fereg`'s. Every command starts from the data in Stata's memory, and every time includes reading them. `areg` cannot save individual fixed effects (—). Peak memory is the largest resident set size of the estimating process during the command, including the data held in memory. [a] Does not discount fixed effects nested within clusters, so standard errors differ. [b] Saved fixed effects or predictions missed the 10^{-7} accuracy check.

Table C.2: Coefficient accuracy and `tolerance()`

Data	`tolerance()`						
	10^{-14}	10^{-12}	10^{-10}	10^{-9}	10^{-8}	10^{-7}	10^{-6}
Panel A. Largest coefficient change, in standard errors							
Three effects, sparse	ref.	1×10^{-11}	8×10^{-12}	1×10^{-11}	5×10^{-11}	1×10^{-11}	stopped[a]
Clustered mobility	ref.	5×10^{-12}	2×10^{-11}	1×10^{-11}	3×10^{-11}	2×10^{-10}	2×10^{-8}
Veneto, largest connected	ref.	2×10^{-10}	5×10^{-10}	3×10^{-10}	1×10^{-10}	8×10^{-9}	2×10^{-6}
Panel B. Iterations							
Three effects, sparse	526	33	28	26	23	21	stopped[a]
Clustered mobility	24	21	18	16	14	13	11
Veneto, largest connected	51	43	36	32	28	23	19

Notes: Linear regressions with four cores, untimed. Each fit is compared with the fit of the same data at a tolerance of 10^{-14} (ref.): panel A gives the largest absolute change of a coefficient divided by its standard error in that reference fit. The simulated designs have one million observations and 16 controls; the Veneto sample is that of table 7. The default is 10^{-8}. Iterations count the iterations of the absorption method. [a] The accuracy check (6) failed, and the command stopped with an error instead of posting estimates.

C.4 Run-to-run variation

The article reports the median of three or five fresh-process runs of each configuration. Table C.3 summarizes how far the repeated runs of each command spread. `fereg`'s repeated runs differed by a median of 3% of their median time. The range exceeded 15% in 3 configurations and reached 72%. These observations are retained in the medians; no slow run was replaced. Ratios in the article that differ by less than these ranges should not be read as differences between the commands.

Table C.3: Run-to-run variation of the measured times

		Range of repeated times (% of median)		
Command	Configurations	Median	90th percentile	Largest
Simulated designs				
`fereg`	53	2.7	8.5	71.8
`reghdfe`	21	0.5	2.1	6.9
`ppmlhdfe`	17	1.2	2.5	6.2
`areg`	17	0.8	3.9	12.4
`fixest`	50	1.2	11.6	25.8
`fixest` (strict)	36	1.4	8.6	17.1
`PyFixest`	50	1.9	6.9	17.3
`PyFixest` (strict)	39	1.4	5.1	25.8
`FixedEffectModels.jl`	25	1.7	35.4	73.0
`FixedEffectModels.jl` (strict)	29	2.2	23.4	86.2
Veneto panels				
`fereg`	10	2.0	2.8	2.8
`reghdfe`	3	1.9	1.9	1.9
`ppmlhdfe`	3	0.6	2.2	2.2
`fixest`	10	2.1	8.0	8.0
`fixest` (strict)	8	1.2	3.1	3.1
`PyFixest`	10	1.7	8.8	8.8
`PyFixest` (strict)	10	2.0	11.2	11.2
`FixedEffectModels.jl`	5	2.4	4.8	4.8
`FixedEffectModels.jl` (strict)	5	4.0	35.2	35.2

Notes: For each configuration in which a command completed every planned run (five, or three above one million observations and on the Veneto panels), the difference between its slowest and fastest run divided by its median.

C.5 All simulated configurations

Table C.4 lists every simulated configuration of the article's comparisons with `fereg`'s median time and each comparator's time relative to it, including the runs with strict tolerances that the article summarizes in its text. The counts and ranges stated in section 7 can be checked against it.

Table C.4: All simulated configurations

	`fereg`	Stata		`fixest`		`PyFixest`		`FixedEffectModels.jl`	
Configuration	seconds	`reghdfe`	`areg`	default	strict	default	strict	default	strict
Linear regression									
Worker–firm	0.82	18.0	11.0	2.48	3.54	2.72	3.65	2.00[a]	3.04[a]
Worker–firm, saved	1.07	15.5[b]		2.49[b]	3.21	8.55[b]	9.33[b]	2.74[b]	3.59[b]
Worker–firm, robust	0.93	16.1	10.3	2.59	3.56	2.44	3.27	1.86[a]	2.77[a]
Worker–firm, robust, saved	1.18	14.2[b]		2.59[b]	3.27	7.83[b]	8.44[b]	2.59[b]	3.35[b]
Worker–firm, clustered	0.92	16.5	10.5[c]	2.37	3.33[d]	2.83	3.65	1.84[a]	2.78[a]
Worker–firm, clustered, saved	1.22	14.1[b]		2.30[b]	2.96[d]	7.88[b]	8.50[b]	2.49[b]	3.22[b]
Worker–firm, 1 core	1.82	15.5	8.12	2.23	3.32	2.76	3.94	1.29[a]	2.39[a]
Worker–firm, 2 cores	1.14	16.8	9.40	2.35	3.41	2.90	4.02	1.65[a]	2.72[a]
Worker–firm, 8 cores	0.58			2.83	3.92	2.85	3.67	2.46[a]	3.58[a]
Worker–firm, 16 cores	0.52			3.47	4.51	2.87	3.73	2.71[a]	3.85[a]
Worker–firm, 16 cores, clustered, saved	0.84			2.99[b]	3.59[d]	11.1[b]	11.3[b]	3.19[b]	3.96[b]
Two effects, dense, 16 cores	0.39			3.74	4.15	1.91	1.90	1.80[a]	1.80[a]
Three effects, sparse, 16 cores	2.34			6.94[a]	24.7[a]	0.92[a]	1.11[a]	5.89[d]	10.2[a]
Worker–firm, 4 controls	0.49	12.9	12.2	1.75	3.13	1.55	2.04	1.75[a]	2.27[a]
Worker–firm, 32 controls	1.29	23.5	11.5	2.54	3.57	3.23	4.37	2.09[a]	3.35[a]
Clustered mobility	1.02	44.3	78.0	4.86[a]	9.75	2.52[a]	2.93[a]	3.86[d]	8.15[a]
One effect	0.56	4.77	1.99	1.81	1.78	1.42	1.43	0.71	0.71
Two effects, dense	0.65	9.90	7.28	2.04	2.28	1.28	1.28	1.12[a]	1.12[a]
Two effects, sparse	0.91	62.5	>600[e]	9.51[d]	38.6	2.69[a]	3.21[a]	3.84[d]	9.65[a]
Three effects, dense	1.03	8.92	5.25	1.43	1.73	0.98	1.00	0.84[a]	0.88[a]
Three effects, sparse	4.32	29.6	>600[e]	5.76[a]	23.2[a]	0.80[a]	1.02[a]	2.84[d]	7.09[a]
Worker–firm, 100,000 obs.	0.14	9.21	5.28	1.90	2.41	1.40	1.76	2.69[a]	3.28[a]
Worker–firm, 300,000 obs.	0.33	12.6	7.03	1.97	2.72	1.93	2.49	2.44[a]	3.29[a]
Worker–firm, 3,000,000 obs.	2.42	22.4	15.0	2.67	4.14	3.28	4.49	2.12[a]	3.78[a]
Worker–firm, 10,000,000 obs.	8.30	27.9	14.0	2.94	4.97	4.78	6.69	2.32[a]	4.66[a]
Three effects, sparse, 100,000 obs.	0.49	16.6	11.3	2.67[a]	8.85[a]	0.62[a]	0.75[a]	2.17[d]	4.26[a]
Three effects, sparse, 300,000 obs.	1.31	21.5	108	3.77[a]	14.0	0.78[a]	0.98[a]	2.49[d]	5.34[a]
Three effects, sparse, 3,000,000 obs.	13.68	31.5	>600[e]	6.39[a]	24.4[a]	0.83[a]	1.06[a]	2.13[d]	5.99[a]
Three effects, sparse, 10,000,000 obs.	51.57	46.6	>600[e]	6.55[a]	>600[e]	1.07[a]	1.38[a]	2.02[d]	6.39[a]
Poisson regression (PPML)									
Worker–firm	5.72	16.1		1.67	2.51	1.99	2.48		
Worker–firm, saved	6.15	15.3[b]		1.86[b]	2.51	3.08[b]	3.50		
Worker–firm, clustered	5.79	15.7		1.68	2.43	2.09	2.51		
Worker–firm, clustered, saved	6.31	14.8[b]		1.61[b]	2.36	3.13[b]	3.44		
Worker–firm, 1 core	9.42	22.2		2.08		2.76			
Worker–firm, 2 cores	7.39	17.2		1.86		2.24			
Worker–firm, 8 cores	4.62			1.56		2.04			
Worker–firm, 16 cores	4.20			1.57		2.01			
Worker–firm, 16 cores, clustered, saved	4.98			1.50[b]		3.36[b]			
Two effects, dense, 16 cores	2.46			1.50		1.13			
Three effects, sparse, 16 cores	9.55			9.47[a]		1.09[a]			
Worker–firm, 4 controls	3.95	6.75		0.95	1.44	1.47	1.59		
Worker–firm, 32 controls	7.92	25.1		2.74	3.80	2.57	3.15		

	fereg	Stata		fixest		PyFixest		FixedEffectModels.jl	
Configuration	seconds	reghdfe	areg	default	strict	default	strict	default	strict
Clustered mobility	5.58	46.9		5.16[a]	18.6[a]	1.72[a]	2.49[a]		
One effect	2.89	8.77		1.36		1.15			
Two effects, dense	3.64	11.8		1.22		0.97			
Two effects, sparse	5.09	86.6		9.66[a]	>600[e]	2.36[a]	3.64[a]		
Three effects, dense	3.99	12.4		1.34		0.96			
Three effects, sparse	14.41	62.3		10.4[a]	>600[e]	1.32[a]	2.21[a]		
Gravity, with zeros	15.14	5.15		0.82[f]	1.24[f]	0.89[f]	1.44[f]		
Worker–firm, 100,000 obs.	0.49	16.7		3.23		3.12			
Worker–firm, 300,000 obs.	2.18	12.9		1.49		1.70			
Worker–firm, 3,000,000 obs.	15.74	17.1		2.36		2.51			
Worker–firm, 10,000,000 obs.	66.06	14.9		2.10		2.69			

Notes: Median seconds of `fereg` and each comparator's median time divided by `fereg`'s in the same configuration and on the same node. Unless noted, one million observations, four cores, 16 controls, and IID (linear) or robust (PPML) standard errors; "clustered" and "saved" denote clustered standard errors and saved fixed effects. For PPML, the `reghdfe` column reports `ppmlhdfe`. Strict columns use tighter convergence tolerances. Blank: not run. Ratios with a mark compare results that differ from `fereg`'s. [a] Counts degrees of freedom differently; its results otherwise agree, and its standard errors differ only by the resulting small-sample factor. [b] Saved fixed effects or predictions missed the 10^{-7} accuracy check. [c] Does not discount fixed effects nested within clusters, so standard errors differ. [d] Coefficients or covariance entries differ by more than the tolerance. [e] Did not finish within its command limit (3,600 seconds for reghdfe/ppmlhdfe, 600 for other commands). [f] Keeps the observations that a control separates and reports a large finite coefficient for it, so its sample differs; its other coefficients agree.

References

Alexander Fischer and Kristof Schröder. Graph preconditioning for high-dimensional fixed effects regression. Working paper, draft of June 2026, GitHub commit bd77e87, 2026. URL https://github.com/py-econometrics/within-paper. Accessed October 28, 2026.

Yuan Gao, Rasmus Kyng, and Daniel A. Spielman. AC(k): Robust solution of Laplacian equations by randomized approximate Cholesky factorization. *SIAM Journal on Scientific Computing*, 48(3):A1643–A1684, 2026. doi: 10.1137/24M1673577.

Ioannis Koutis, Gary L. Miller, and David Tolliver. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. *Computer Vision and Image Understanding*, 115(12): 1638–1646, 2011. doi: 10.1016/j.cviu.2011.05.013.

Rasmus Kyng and Sushant Sachdeva. Approximate Gaussian elimination for Laplacians—fast, sparse, and simple. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 573–582, 2016. doi: 10.1109/FOCS.2016.68.

Andrea Toselli and Olof B. Widlund. *Domain Decomposition Methods—Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, Berlin, 2005. doi: 10.1007/b137868.