Supplemental Information: `pestpp-ies` optional arguments and work flow

argument name	argument type	description	default
++ies_parameter_ensemble(<par_en>)	string	CSV file (or binary file) containing the parameter ensemble. Must contain a header that includes (at least) all adjustable parameter and an index of unique identifiers	None, resulting in a parameter ensemble of size <num_reals >being drawn using the multivariate normal distribution defined by the initial parameter values and the prior parameter covariance matrix
++ies_observation_ensemble(<obs_en>)	string	CSV file (or binary file) containing the observation ensemble. Must contain a header that includes (at least) all non-zero weight observations and an index of unique identifiers	None, resulting in an observation ensemble of size <num_reals >being drawn using the multivariate normal distribution defined by the observed values and covariance matrix implied by the observation weights
++ies_num_reals(<num_reals >)	integer	number of realizations to draw to form the parameter and observation ensembles	50
++ies_restart_obs_en(<restart_obs_en >)	string	CSV file (or binary file) containing the simulated outputs from evaluating the parameter ensemble (running the model once for each parameter realization). Must contain a header that includes (at least) all non-zero weight observations and an index of unique identifiers (must be commensurate with <obs_en >if specified).	None, resulting in the parameter ensemble being evaluated during the initialization phase

Continued on next page

argument name	argument type	description	default
++ies_bad_phi(<bad_phi >)	float	defines maximum acceptable objective function value. During each parameter ensemble evaluation, realizations yielding an objective function larger than <bad_phi >will be dropped	1e30
++ies_lambda_mults(<lambda_mults >)	comma-separated list of floats	multiplied by the current value of lambda to yield multiple values of lambda to test during an upgrade calculation.	1.0
++ies_initial_lambda(<init_lambda >)	float	initial value of lambda	$10^{floor(log_{10}\frac{\mu\phi}{2.0*nobs})}$
++ies_subset_size(<subset_size >)	integer	number of realizations to evaluate during testing of multiple lambdas	1e6
++ies_reg_factor(<reg_factor >)	float in (0.0,1.0)	regularization factor as a fraction of the measurement objective function; can be used to enforce more regularization (if desired).	0.0
++ies_use_approx(<use_approx >)	"true" or "false"	flag to use the approximate upgrade calculation formula. Testing has shown the approx solution to be effective in most settings	"false"

Continued on next page

argument name	argument type	description	default
++ies_use_prior_scaling(<use_prior_scaling >)	"true" or "false"	flag to use $\mathbf{\Sigma}_\theta$ in the calculation of $\mathbf{\Delta}_{\text{par}}$. Testing has shown prior scaling can be important for finding optimal solutions for large-scale problems but can hamper the progress of smaller problems	"false"
++parcov_filename(<filename >)	string	ASCII or binary file containing the prior parameter covariance matrix ($\mathbf{\Sigma}_\theta$)	if not specified and if <par_en >is specified and use_empirical_prior is "true", a diagonal $\mathbf{\Sigma}_\theta$ is constructed from the empirical variances implied in the parameter ensemble. Otherwise, a diagonal $\mathbf{\Sigma}_\theta$ is constructed using the parameter bounds, assuming these define the 95% confidence range
++ies_use_empirical_prior(<use_empirical_prior >)	"true" or "false"	flag to construct $\mathbf{\Sigma}_\theta$ as a diagonal matrix with parameter variances calculated from the ensemble in par_en. If true, <par_en >must have been passed	"false"
++ies_verbose_level (<verbose_level >)	0,1,2,3	integer flag for the level of output produce by `pestpp-ies`. If 3, all intermediate matrices are saved to ASCII files, which can require considerable storage	1

Continued on next page

argument name	argument type	description	default
++ies_add_bases(<add_bases >)	"true" or "false"	flag to augment $\mathbf{\Theta}$ and $\mathbf{D}_{obs}$ with the initial (prior minimum error variance) parameter values and noiseless observation values, respectively. The added realization is "base"; this argument is ignored if a "base" realization exists	"true"
++ies_enforce_bounds(<enforce_bounds >)	"true" or "false"	flag to flag to enforce parameter bounds listed in `pest` control file during realization draws and upgrade calculations	"true"
++ies_save_binary(<save_binary >)	"true" or "false"	If "true", save output files to `pest`-style binary files; if "false", save output files to CSV files	"false"
++par_sigma_range(<par_sigma_range >)	non-negative float	the number of standard deviations spanned by the parameter bounds, used in construction of a diagonal $\mathbf{\Sigma}_\theta$	4.0
++ies_accept_phi_fac(<accept_phi_fac >)	float >= 1.0	During lambda testing, if a parameter ensemble yields a μ_ϕ <$\mu_{\phi,last}$ * accept_phi_fac, it is accepted. Otherwise, it is not accepted. This factor is also used when evaluating subset ensemble results.	1.05
++ies_lambda_inc_fac(<lambda_inc_fac >)	float >1.0	factor to increase current λ by if ϕ is not acceptable	10.0

Continued on next page

argument name	argument type	description	default
++ies_lambda_dec_fac(<lambda_dec_fac >)	float <1.0	factor to decrease best λ by if ϕ is acceptable	0.75
++ies_save_lambda_en(<save_lambda_en >)	bool	flag to save upgrade ensembles each iteration	false
++lambda_scale_fac(<lambda_scale_fac >)	list of comma-separated floats >0.0	line search factors to apply to each λ upgrade matrix	1.0

Algorithm 1: `pestpp-ies` initialization workflow

Data: Given: pest control file
read noptmax from pest control file
read parameter and observation information from pest control file
read optional inputs: *parameter_ensemble, observation_ensemble, restart_obseravtion_ensemble, parcov_filename, num_reals, use_prior_scaling, use_approx, init_lambda, lambda_mults, subset_size, add_bases, use_empirical_prior* from '++' arguments
build $\mathbf{\Sigma}_\epsilon$ from observation weights
if *parcov_filename* **then**
  load $\mathbf{\Sigma}_\theta$ from *parcov_filename*
else
  build $\mathbf{\Sigma}_\theta$ from parameter bounds
end
if *parameter_ensemble* **then**
  load $\mathbf{\Theta}$ from file *parameter_ensemble*
  if *use_empirical_prior* **then**
    construct diagonal $\mathbf{\Sigma}_\theta$ from $\mathbf{\Theta}$
else
  draw $\mathbf{\Theta}$ of size *num_reals* using initial parameter values and $\mathbf{\Sigma}_\theta$
end
if *observation_ensemble* **then**
  load $\mathbf{D}_{\text{obs}}$ from file *obseravation_ensemble*
else
  draw $\mathbf{D}_{\text{obs}}$ of size *num_reals* using observation values and $\mathbf{\Sigma}_\epsilon$
end
if *add_bases* **then**
  add initial parameter values to $\mathbf{\Theta}$
  add observation values to $\mathbf{D}_{obs}$
if *noptmax = 0* **then**
  calculate mean parameter vector $\overline{\mathbf{\Theta}}$
  add $\overline{\mathbf{\Theta}}$ to the run queue
  make model runs
  process run queue $\rightarrow \overline{\mathbf{D}}_{sim}$
  exit
$\mathbf{\Theta}^0 = \mathbf{\Theta}$
if *restart_observation_ensemble* **then**
  load $\mathbf{D}_{\text{sim}}$ from file *restart_observation_ensemble*
else
  add $\mathbf{\Theta}$ to run queue
  make model runs
  procee run queue $\rightarrow \mathbf{D}_{sim}$
end
λ = *init_lambda*
calculate objective function statistics: $\mathbf{D}_{sim}, \mathbf{D}_{obs} \rightarrow \mu_{\phi,last}, \sigma_{\phi,last}$
if *use_approx is False* **then**
  if *use_prior_scaling* **then**
    $\mathbf{\Delta}_{\text{par}} = \mathbf{\Sigma}_\theta^{-\frac{1}{2}}(\mathbf{\Theta}^0 - \overline{\mathbf{\Theta}}^0)$
  else
    $\mathbf{\Delta}_{\text{par}} = \mathbf{\Theta}^0 - \overline{\mathbf{\Theta}}^0$
  end
  $\mathbf{U}, \mathbf{s}, \mathbf{V^t} \leftarrow \mathbf{\Delta}^0_{par}$
  $\mathbf{A}_m = \mathbf{U}\mathbf{s}^{-1}$

Algorithm 2: `pestpp-ies` iteration workflow

$iter = 0$
while $iter < noptmax$ **do**
  if *use_prior_scaling* **then**
    $\mathbf{\Delta}_{\text{par}} = \mathbf{\Sigma}_{\theta}^{-\frac{1}{2}}(\mathbf{\Theta} - \overline{\mathbf{\Theta}})$
  else
    $\mathbf{\Delta}_{\text{par}} = \mathbf{\Theta} - \overline{\mathbf{\Theta}}$
  end
  $\mathbf{\Delta}_{\text{sim}} = \mathbf{\Sigma}_{\epsilon}^{-\frac{1}{2}}(\mathbf{D}_{\text{sim}} - \overline{\mathbf{D}}_{\text{sim}})$
  $\boldsymbol{U}, \boldsymbol{s}, \boldsymbol{V}^T \leftarrow \mathbf{\Delta}_{\text{sim}}$
  for λ_m *in lambda_mults* **do**
    $\lambda_i = \lambda * \lambda_m$
    $\mathbf{\Theta}_{\lambda_i} =$ `upgrade_calculation`(λ_i)
    keep $\mathbf{\Theta}_{\lambda_i}$
  end
  if *subset_size* $<$ *num_reals* **then**
    add first *subset_size* realizations from each $\mathbf{\Theta}_{\lambda_i}$ to the run queue
    (e.g., add $\mathbf{\Theta}_{\lambda_i}$[: *subset_size*] realizations to the run queue)
  else
    add $\mathbf{\Theta}_{\lambda_i}$ to the run queue
  end
  make model runs
  for *each* $\mathbf{\Theta}_{\lambda_i}$ **do**
    process run queue $\rightarrow \mathbf{D}_{sim,\lambda_i}$
    calculate objective function statistics $\mathbf{D}_{sim,\lambda_i}, \mathbf{D}_{obs} \rightarrow \mu_{\phi,\lambda_i}, \sigma_{\phi,\lambda_i}$
    if $\mu_{\phi,\lambda_i} < \mu_{\phi,best}$ **then**
      $\lambda_{best} = \lambda_i$
      $\mu_{\phi,\lambda_{best}} = \mu_{\phi,\lambda_i}$
  end
  if *subset_size* $<$ *num_reals* **then**
    if $\mu_{\phi,\lambda_{best}} > \mu_{\phi,best} * accept\_phi\_fac$ **then**
      increase lambda: $\lambda = \lambda * lambda\_inc\_fac$
      continue `// skip running the rest of the ensemble`
    add last (*num_reals* - *subset_size*) realizations of $\mathbf{\Theta}_{\lambda_{best}}$ to run queue
    make model runs
    proces run queue $\rightarrow \mathbf{D}_{sim,\lambda_{best}}[num\_reals - subset\_size :]$
  calcuate objective function statistics $\mathbf{D}_{sim,\lambda_{best}}, \mathbf{D}_{obs} \rightarrow \mu_{\phi,\lambda_i}, \sigma_{\phi,\lambda_i}$
  if $\mu_{\phi,best} < (\mu_{\phi,last} * accept\_phi\_fac)$ **then**
    update parameter ensemble: $\mathbf{\Theta} = \mathbf{\Theta}_{\lambda_{best}}$
    update observation ensemble: $\mathbf{D}_{sim} = \mathbf{D}_{sim,\lambda_{best}}$
    if $\sigma_{best} < \sigma_{last}$ **then**
      update lambda: $\lambda = lambda\_dec\_fac * \lambda_{i,best}$
    $\mu_{\phi,last}, \sigma_{\phi,last} = \mu_{\phi,best}, \sigma_{\phi,best}$
  else
    increase lambda: $\lambda = \lambda * lambda\_inc\_fac$
  end
  $iter$ += 1
end

Algorithm 3: `pestpp-ies` upgrade calculation workflow

Given: λ_i
$\mathbf{I}_s = (\lambda_i + \mathbf{s}^2)^{-1}$
$\mathbf{X}_1 = \mathbf{U}^T \mathbf{\Sigma}_\epsilon^{-\frac{1}{2}} (\mathbf{D}_{sim} - \mathbf{D}_{obs})$
$\mathbf{X}_2 = \mathbf{I}_s \mathbf{X}_1$
$\mathbf{X}_3 = \mathbf{V}\mathbf{s}\mathbf{X}_2$
if *use_prior_scaling* **then**
    $\mathbf{upgrade}_1 = \mathbf{\Sigma}_\theta^{-\frac{1}{2}} \mathbf{\Delta}_{par} \mathbf{X}_3$
else
    $\mathbf{upgrade}_1 = \mathbf{\Delta}_{par} \mathbf{X}_3$
end
if *iter > 1 and use_approx is False* **then**
    $\mathbf{X}_4 = \mathbf{A}_m^T \mathbf{\Sigma}_\theta^{-\frac{1}{2}} (\mathbf{\Theta} - \mathbf{\Theta}^0)$
    $\mathbf{X}_5 = \mathbf{A}_m \mathbf{X}_4$
    $\mathbf{X}_6 = \mathbf{\Delta}_{par}^T \mathbf{X}_5$
    $\mathbf{X}_7 = \mathbf{V}\mathbf{I}_s\mathbf{V}^T \mathbf{X}_6$
    if *use_prior_scaling* **then**
        $\mathbf{upgrade}_2 = \mathbf{\Sigma}_\theta^{-\frac{1}{2}} \mathbf{\Delta}_{par} \mathbf{X}_7$
    else
        $\mathbf{upgrade}_2 = \mathbf{\Delta}_{par} \mathbf{X}_7$
    end
$\mathbf{\Theta}_{\lambda_i} = \mathbf{\Theta} + \mathbf{upgrade}_1 + \mathbf{upgrade}_2$
return $\mathbf{\Theta}_{\lambda_i}$

Quantity	Definition
$\mathbf{\Sigma}_\theta$	prior parameter covariance matrix
$\mathbf{\Sigma}_\epsilon$	observation noise covariance matrix
$\mathbf{\Theta}$	current parameter ensemble
$\mathbf{\Theta}^0$	initial parameter ensemble
$\mathbf{D}_{obs}$	observation ensemble (ensemble of observation values with realizations of measurement noise)
$\mathbf{D}_{sim}$	simulated observation ensemble (e.g., ensemble of simulated equivalents to observations resulting from evaluating $\mathbf{\Theta}$)
λ	current value of Marquardt lambda
λ_m	lambda multiplier value
λ_i	value of lambda to test, product of $\lambda * \lambda_m$
μ_ϕ	mean value of objective function values, function of both $\mathbf{D}_{sim}$ and $\mathbf{D}_{obs}$
σ_ϕ	standard deviation of objective function values, function of both $\mathbf{D}_{sim}$ and $\mathbf{D}_{obs}$
$\overline{\mathbf{\Theta}}$	mean parameter value vector
$\mathbf{\Delta}_{par}$	ensemble (matrix) of differences of parameter values from mean parameter values
$\overline{\mathbf{D}}_{sim}$	mean simulated observation value vector
$\mathbf{\Delta}_{sim}$	ensemble (matrix) of differences of simulated observation values from mean simulated observation values