Performing the estimation

shorten? Quite repetitive at the beg. of every new chapter

Having created all ~~the~~ relevant settings for the physical environment (see Environment Setup) dynamical model (see Propagation Setup), the parameters that are to be estimatd (see Parameter settings), the settins for the observation models (see Observation Model Setup) and the actual observations (simulated or real; see Observation Simulation), the estimation can be performed. Both a full estimation and a covariance analysis are performed by using the `Estimator` object, which is created as follows:

```
estimator = numerical_simulation.Estimator(
    bodies,
    parameters_to_estimate,
    observation_settings_list,
    propagator_settings)
```

where the propagator settings may be single-, multi- or hybrid arc. Creating an `Estimator` object automatically propagates the dynamics and variational equations for the specifief propagator and parameter settings.

specific

Covariance analysis

The settings for a covariance analysis described here can be used to compute the covariance using the `compute_covariance()` function.

```
covariance_analysis_output = estimator.compute_covariance(
    covariance_analysis_settings)
```

where the `covariance_analysis_settings` is an object of type `CovarianceAnalysisOutput` from which the design matrix, covariance, etc. can be retrieved.

Normalization

should be in chapter with cov. var. theory

The partial derivative matrix $\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{p}}$ is computed automatically for all observations and parameters, from which the inverse covariance $\mathbf{P}^{-1}$ is then computed, as described here. However, due to the potentially huge difference in order of magnitude of the estimated parameters (for instance, the Sun's gravitational parameter, at approximately 1.3267:math:*cdot 10^{20}* m^3/s^2, and the bias of a VLBI observaion, at 10^{-9} radians), the inversion of the matrix $\mathbf{P}^{-1}$ can be extremely ill-posed. We partly correct for this problem by normalizing the parameters.

math mode

The normalization is achieved by computing a vector $\mathbf{N}$ (of the same size as the parameter vector $\mathbf{p}$, such that for each column of the matrix $\mathbf{H}$, we have:

$$\max_i \left| \frac{H_{ij}}{N_j} \right| = 1$$

That is, the entries of $\mathbf{N}$ are chosen such that they normalize the corresponding column of $\mathbf{H}$ to be in the range $[-1, 1]$. We denote the normalized quantities with a tilde, so that:

$$\tilde{H}_{ij} = \frac{H_{ij}}{Nj}$$
$$\tilde{P}_{ij} = P_{ij} N_i N_j$$

When inverting the normal equations, normalized quantities are always used. Both the normalized and regular quantities can be retrieved from the `CovarianceAnalysisOutput` class.

Full estimation

Similarly, the settings for a full estimation described here can be used to perform the full estimation using the `perform_estimation()` function.

```
estimation_output = estimator.perform_estimation(
    estimation_settings)
```

where the `estimation_output` is an object of type `EstimationOutput`, which (in addition to all information in `CovarianceAnalysisOutput`) contains information on the iteration process (depending on the specific output settings provided in `estimation_settings`.