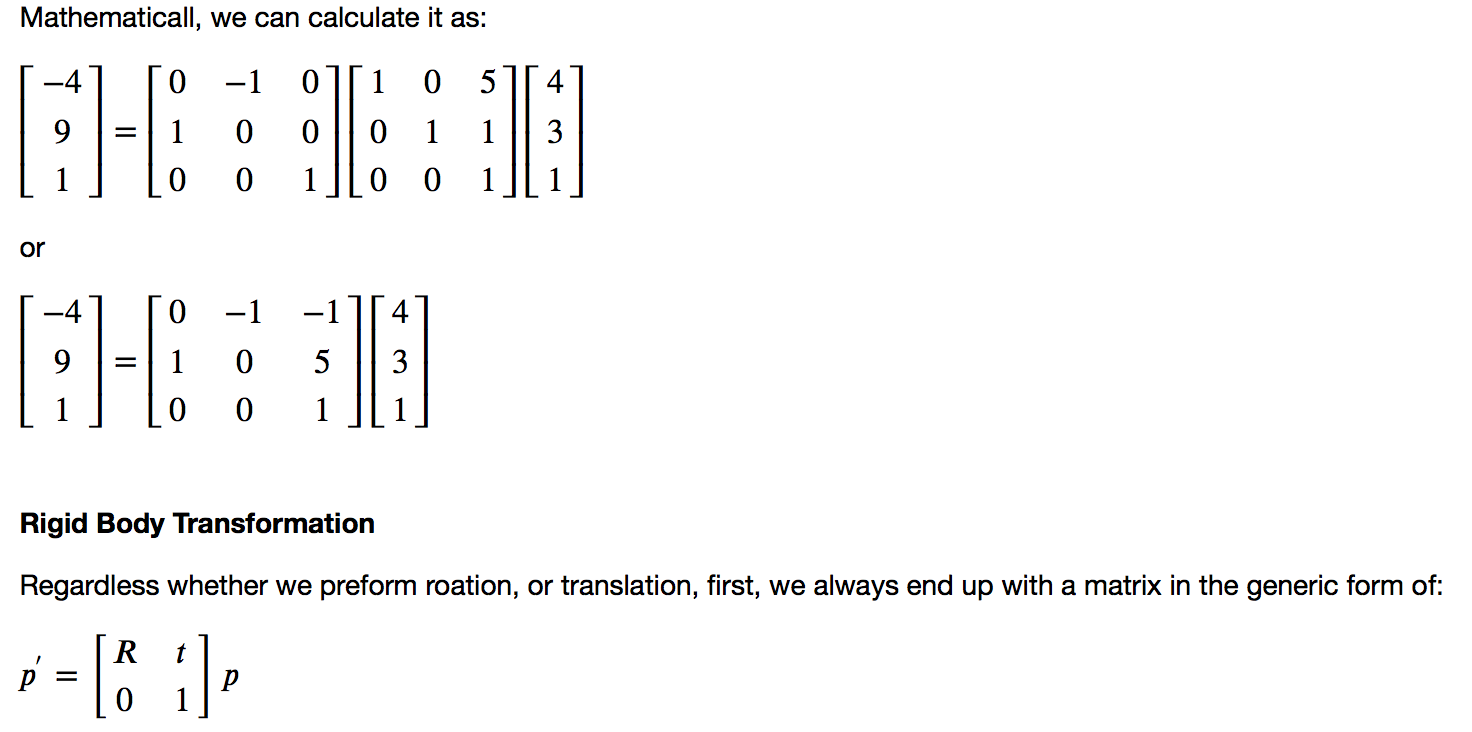
AIP\_19\_Day transformation



AIP 20 matual initialization

**Registration**

The specific registration task at hand estimates a 3D rigid transformation between images of different modalities. There are multiple components from each group (optimizers, similarity metrics, interpolators) that are appropriate for the task. Note that each component selection requires setting some parameter values. We have made the following choices:

* Similarity metric, mutual information (Mattes MI):
  + Number of histogram bins, 50.
  + Sampling strategy, random.
  + Sampling percentage, 1%.
* Interpolator, sitkLinear.
* Optimizer, gradient descent:
  + Learning rate, step size along traversal direction in parameter space, 1.0 .
  + Number of iterations, maximal number of iterations, 100.
  + Convergence minimum value, value used for convergence checking in conjunction with the energy profile of the similarity metric that is estimated in the given window size, 1e-6.
  + Convergence window size, number of values of the similarity metric which are used to estimate the energy profile of the similarity metric, 10.

Perform registration using the settings given above, and take advantage of the built in multi-resolution framework, use a three tier pyramid.

In this example we plot the similarity metric's value during registration. Note that the change of scales in the multi-resolution framework is readily visible.

registration\_method = sitk.ImageRegistrationMethod()

*# Similarity metric settings.*

registration\_method.SetMetricAsMattesMutualInformation(numberOfHistogramBins=50)

registration\_method.SetMetricSamplingStrategy(registration\_method.RANDOM)

registration\_method.SetMetricSamplingPercentage(0.01)

registration\_method.SetInterpolator(sitk.sitkLinear)

*# Optimizer settings.*

registration\_method.SetOptimizerAsGradientDescent(learningRate=1.0, numberOfIterations=100, convergenceMinimumValue=1e-6, convergenceWindowSize=10)

registration\_method.SetOptimizerScalesFromPhysicalShift()

*# Setup for the multi-resolution framework.*

registration\_method.SetShrinkFactorsPerLevel(shrinkFactors = [4,2,1])

registration\_method.SetSmoothingSigmasPerLevel(smoothingSigmas=[2,1,0])

registration\_method.SmoothingSigmasAreSpecifiedInPhysicalUnitsOn()

*# Don't optimize in-place, we would possibly like to run this cell multiple times.*

registration\_method.SetInitialTransform(initial\_transform, inPlace=**False**)

*# Connect all of the observers so that we can perform plotting during registration.*

registration\_method.AddCommand(sitk.sitkStartEvent, start\_plot)

registration\_method.AddCommand(sitk.sitkEndEvent, end\_plot)

registration\_method.AddCommand(sitk.sitkMultiResolutionIterationEvent, update\_multires\_iterations)

registration\_method.AddCommand(sitk.sitkIterationEvent, **lambda**: plot\_values(registration\_method))

final\_transform = registration\_method.Execute(sitk.Cast(fixed\_image, sitk.sitkFloat32),

sitk.Cast(moving\_image, sitk.sitkFloat32))