

## Case 1 (Three Level Ruge-Stuben Solver):

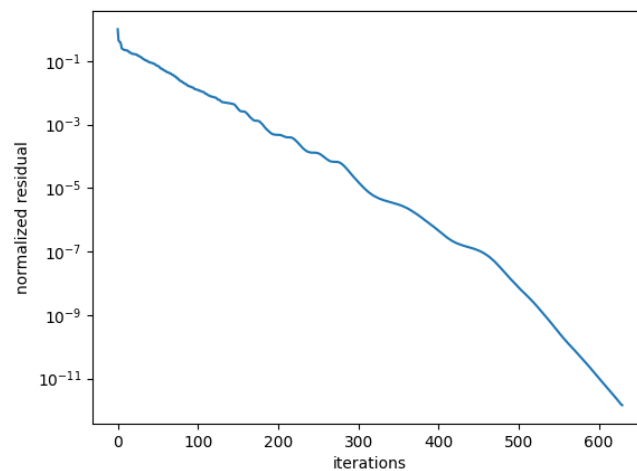
### Parameters:

```
# AMG Parameters
maximum_levels = 3          # Max levels in hierarchy
maximum_coarse = 3          # Max points allowed on coarse grid
tolerance = 1e-9            # Residual convergence tolerance
coarse_solver = 'splu'
krylov = 'gmres'
keep = False
strength=('classical', {'theta': 0.01})
splitting = ('RS', {'second_pass': True})
interp = 'classical'
restriction = ('air', {'theta': 0.01, 'degree': 2})
prerelax = ('cf_jacobi', {'omega': 1.0, 'iterations': 1, 'withrho': True, 'f_iterations': 2, 'c_iterations': 1})
postrelax = ('fc_jacobi', {'omega': 1.0, 'iterations': 1, 'withrho': True, 'f_iterations': 2, 'c_iterations': 1})
filter_entries = None
maxiter = 1000
```

### Solver:

```
# Create a multilevel solver using classical Ruge-Stuben AMG Solver
ml = pyamg.ruge_stuben_solver(A, strength = strength, CF = splitting, interpolation = interp,
                              presmoothing = prerelax, postsmoothing = postrelax,
                              max_levels = maximum_levels, max_coarse = maximum_coarse,
                              keep = keep, coarse_solver=coarse_solver)
```

### Result:



```
MultilevelSolver
Number of Levels: 3
Operator Complexity: 1.527
Grid Complexity: 1.339
Coarse Solver: 'splu'
level unknowns nonzeros
0 169808 9663776 [65.51%]
1 49125 4301453 [29.16%]
2 8501 786543 [5.33%]
```

n	iter	rho	OpCx	GdCx	Work
169808	632	0.96	1.5	1.3	81

## Case 2 (Multilevel Ruge-Stuben Solver):

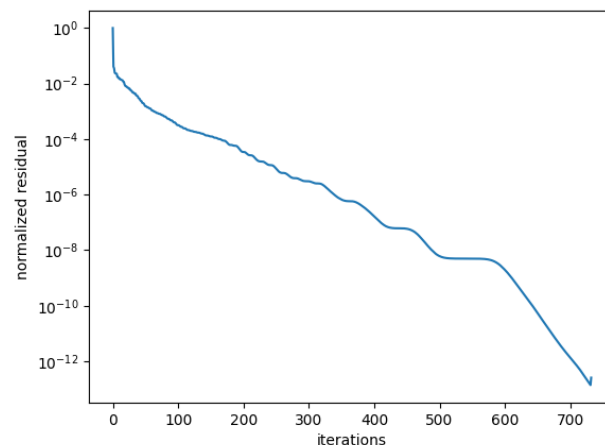
### Parameters:

```
# AMG Parameters
maximum_levels = 30      # Max levels in hierarchy
maximum_coarse = 30      # Max points allowed on coarse grid
tolerance = 1e-9         # Residual convergence tolerance
coarse_solver = 'splu'
krylov = 'gmres'
keep = False
strength=('classical', {'theta': 0.01})
splitting = ('RS', {'second_pass': True})
interp = 'classical'
restriction = ('air', {'theta': 0.01, 'degree': 2})
prerelax = ('cf_jacobi', {'omega': 1.0, 'iterations': 1, 'withrho': True, 'f_iterations': 2, 'c_iterations': 1})
postrelax = ('fc_jacobi', {'omega': 1.0, 'iterations': 1, 'withrho': True, 'f_iterations': 2, 'c_iterations': 1})
filter_entries = None
maxiter = 1000
```

### Solver:

```
# Create a multilevel solver using classical Ruge-Stuben AMG Solver
ml = pyamg.ruge_stuben_solver(A, strength = strength, CF = splitting, interpolation = interp,
                              presmoothing = prerelax, postsmoothing = postrelax,
                              max_levels = maximum_levels, max_coarse = maximum_coarse,
                              keep = keep, coarse_solver=coarse_solver)
```

### Result:



```
MultilevelSolver
Number of Levels:      7
Operator Complexity:   1.539
Grid Complexity:       1.350
Coarse Solver:         'splu'
  level  unknowns  nonzeros
    0      169808   9663776 [65.00%]
    1      49125   4301453 [28.93%]
    2       8501   786543 [5.29%]
    3       1416   98012 [0.66%]
    4        300   15362 [0.10%]
    5         71    2451 [0.02%]
    6         21     417 [0.00%]
```

AMG solver results					
n	iter	rho	OpCx	GdCx	Work
169808	733	0.96	1.5	1.4	90

### Case 3 (Multilevel AIR Solver):

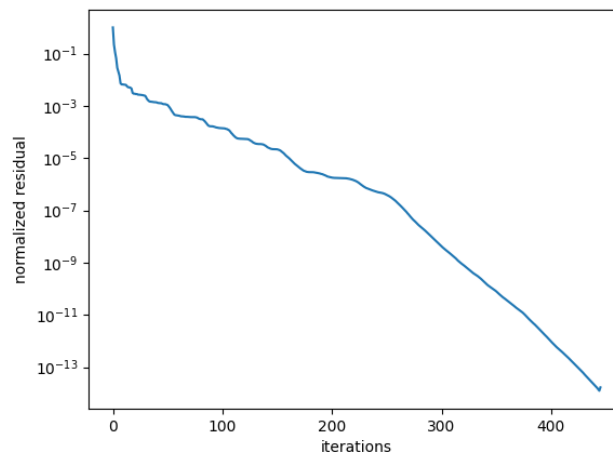
#### Parameters:

```
maximum_levels = 30      # Max levels in hierarchy
maximum_coarse = 30      # Max points allowed on coarse grid
tolerance = 1e-9         # Residual convergence tolerance
coarse_solver = 'splu'
krylov = 'gmres'
keep = False
strength=('classical', {'theta': 0.01})
splitting = ('RS', {'second_pass': True})
interp = 'classical'
restriction = ('air', {'theta': 0.05, 'degree': 2})
prerelax = ('cf_jacobi', {'omega': 1.0, 'iterations': 1, 'withrho': True, 'f_iterations': 2, 'c_iterations': 1})
postrelax = ('fc_jacobi', {'omega': 1.0, 'iterations': 1, 'withrho': True, 'f_iterations': 2, 'c_iterations': 1})
#prepost = ('gauss_seidel_nr', {'sweep': 'symmetric', 'iterations': 1})
filter_entries = None
maxiter = 1000
```

#### Solver:

```
ml = air_solver(A, strength=strength, CF=splitting, interpolation=interp, restrict=restriction,
               presmoothing=prerelax, postsmoothing=postrelax, filter_operator = filter_entries,
               max_levels=maximum_levels, keep=keep, max_coarse=maximum_coarse, coarse_solver=coarse_solver)
```

#### Result:

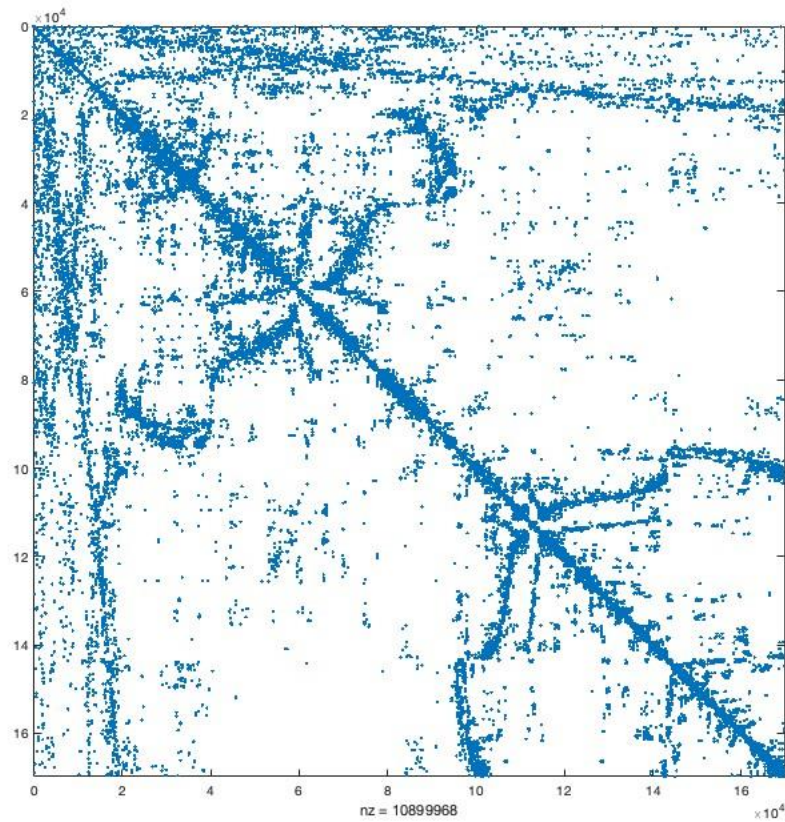


```
MultilevelSolver
Number of Levels:      10
Operator Complexity:   1.884
Grid Complexity:       1.394
Coarse Solver:         'splu'
  level  unknowns  nonzeros
    0      169808   9663776 [53.07%]
    1      49125   5821117 [31.97%]
    2      11696   1956177 [10.74%]
    3       3703    551989 [3.03%]
    4       1414    157028 [0.86%]
    5        573     43066 [0.24%]
    6        233     11001 [0.06%]
    7         101      3094 [0.02%]
    8          45       860 [0.00%]
    9          21       252 [0.00%]
```

AMG solver results					
n	iter	rho	OpCx	GdCx	Work
169808	446	0.93	1.9	1.4	61

## Sparsity Pattern:

---



**Maximum number of non-zeros in each row = 76**  
**Minimum number of non-zeros in each row = 44**