Block-by-block description of MATLAB code

Block-by-block description

1. Setting convergence and stagnation thresholds

Convergence threshold (target residual norm): This value defines when the optimization is considered converged.

```
target_residual_norm = 1e-3;
```

Stagnation tolerance: If residual norm does not change significantly (difference less than this value), the algorithm is considered stuck.

```
stagnation_tol = 1e17;
```

2. Initialization of initial guess

Initial parameter vector: Pack parameters and fields into a vector.

```
x0 = packFieldsAndInterface(fields, params.M
    , params.N, length(params.c_init), params
    .c_init);
```

3. Definition of anonymous residual function

Anonymous function computing residual vector:

```
fun = @(x) computeResidualVector(x, params,
    dx, dy, X, Y, x_interface);
```

4. Optimization options

Options for `lsqnonlin`: - detailed display, - max iterations 100, - function tolerance 1e-6.

```
options = optimoptions('lsqnonlin', ...
'Display', 'iter-detailed', ...
'MaxIterations', 100, ...
'FunctionTolerance', 1e-6);
```

5. Initialization for restart loop

Variables to control restarts and residual norms:

```
max_restarts = 10;
restart_count = 0;
resnorm = Inf;
previous_resnorm = Inf;
```

6. Restart optimization loop

Run optimization while residual norm is above target and restarts are below maximum:

```
while resnorm > target_residual_norm &&
    restart_count < max_restarts
fprintf('Run #%d\n', restart_count + 1);

[x_opt, resnorm, residual, exitflag, output]
     = lsqnonlin(fun, x0, [], [], options);

fprintf('resnorm after iteration: %.6e\n',
    resnorm);

delta_resnorm = abs(previous_resnorm -
    resnorm);
if delta_resnorm < stagnation_tol
fprintf('resnorm change too small:    = %.2e
     < %.2e     restart\n', delta_resnorm,
    stagnation_tol);
end

previous_resnorm = resnorm;
x0 = x_opt;
restart_count = restart_count + 1;

save('optimization_intermediate.mat', 'x0',
    'resnorm', 'restart_count');
end
```

7. Final output

Check if convergence was successful:

```
if resnorm <= target_residual_norm
fprintf('Convergence achieved: resnorm = %.6
    e\n', resnorm);
else
```

```
4                   fprintf('Accuracy not reached after %d
                        restarts. Final resnorm = %.6e\n',
                        restart_count, resnorm);
5                   end
```

8. Unpacking solution

Convert optimized vector back to fields and parameters:

```
1                   [fields_opt, c_opt] =
                        unpackFieldsAndInterface(x_opt, params.M,
                         params.N, length(params.c_init));
```