**1:** ---org/apache/commons/math/optimization/direct/MultiDirectional.java

**2:** +++org/apache/commons/math/optimization/direct/MultiDirectional.java

**3:** protected void iterateSimplex(final Comparator<ValuePair> comparator)

**4:** throws OptimizationException

**5:** {

**6:** while (true) {

**7:** if (++iterations > maxIterations) {

**8:** throw new OptimizationException(new

**9:** MaxIterationsExceededException(maxIterations));

**10:** }

**11:** ...

**12:** final RealPointValuePair contracted = evaluateNewSimplex(original,

**13:** gamma,comparator);

**14:** if (comparator.compare(contracted, best) < 0) {

**15:** return;

**16:** }

**17:**

**19:**+ final int iter = getIterations();

**20:**+ boolean converged = true;

**21:**+ for (int i = 0; i < simplex.length; ++i) {

**22:**+ converged &= checker.converged(iter, original[i], simplex[i]);

**23:**+ }

**24:**+ if (converged) {

**25:**+ return;

**26:**+ }

**1:** ---org/apache/commons/math/optimization/direct/MultiDirectional.java

**2:** +++org/apache/commons/math/optimization/direct/MultiDirectional.java

**3:** protected void iterateSimplex(final Comparator<ValuePair> comparator)

**4:** throws OptimizationException

**5:** {

**6:** while (true) {

**7:** if (++iterations > maxIterations) {

**8:** throw new OptimizationException(new

**9:** MaxIterationsExceededException(maxIterations));

**10:** }

**11:** ...

**12:** final RealPointValuePair contracted = evaluateNewSimplex(original,

**13:** gamma,comparator);

**14:**+ if (true)

**15:**+ return;

**16:** if (comparator.compare(contracted, best) < 0) {

**17:** return;

**18:** }