SECTORING FOR TRACKING

MIRIAM DIAMOND

SEPT 11 2017

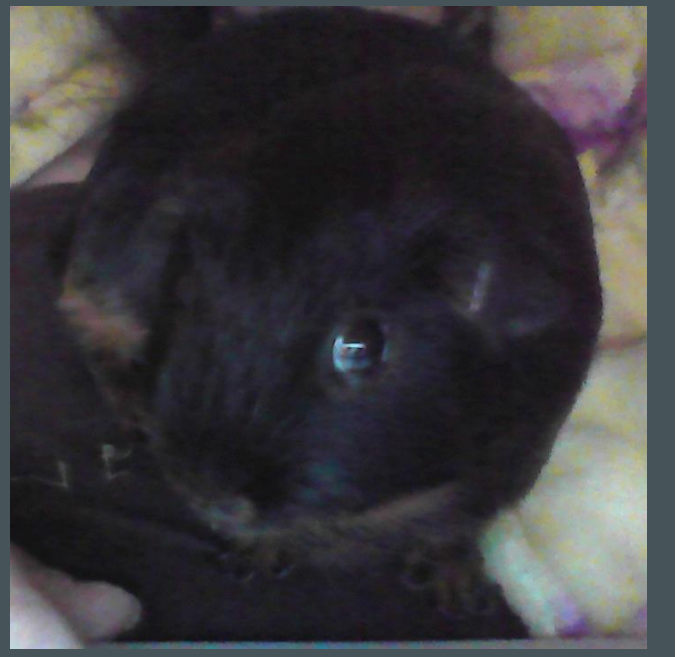
github issue 208

(or, "yet another rabbit-hole")

Good news: recovered some tracks

Bad news: don't really know why

(my new helper for getting out of rabbit-holes)



WHAT IS SECTORING?

- Dividing detector into sectors, and only considering one sector at a time when building tracks
  - Division in phi and/or z
- Should reduce runtime by reducing combinatorics
- Should not affect final tracking results, if sector size is appropriate
  - Decent track shouldn't hop all over the detector
- Machinery exists in hps-java, but currently not activated
  - Sector.java, SeedSectoring.java, SectorManager.java, HitManager.java, FastCheck.java
  - Additional z sectoring ("binning") in FastCheck.java
  - Does anyone remember the history?

WHERE IS SECTORING APPLIED?

- SeedTracker.java:

```
protected void process(EventHeader event) {
    //  Get the hit collection from the event
    List<HelicalTrackHit> hitcol = event.get(HelicalTrackHit.class, _inputCol);

    //  Sort the hits for this event
    _hitmanager.OrganizeHits(hitcol);
    //  Loop over strategies and perform track finding
    for (SeedStrategy strategy : _strategylist) {
        //  Perform track finding under this strategy
        _finder.FindTracks(strategy, _bfield);
        ...
```

```
public void setApplySectorBinning(boolean applySectorBinning) {
    _finder.setApplySectorBinning(applySectorBinning);
    _finder.getConfirmer().setApplySectorBinning(applySectorBinning);
}
```

```
public void OrganizeHits(List<HelicalTrackHit> hitCol) {

    //  Initialize the sector manager
    _smanager.Initialize();

    //  Loop over the hits and let the SectorManager keep track of them
    for (HelicalTrackHit hit : hitCol) {
        //  Tell the sector manager about this hit
        _smanager.AddHit(hit);
    }
}
```

WHERE IS SECTORING APPLIED?

- SeedTrackFinder.java:

```
public boolean FindTracks(SeedStrategy strategy, double bfield, FastCheck checker) {
    if(_applySectorBinning) checker.setDoSectorBinCheck(_hitmanager.getSectorManager());

    //  Find the valid sector combinations
    SeedSectoring ss = new SeedSectoring(_hitmanager, strategy, bfield,_applySectorBinning);
    List<List<Sector>> sslist = ss.SeedSectors();

    //  Loop over the valid sector combinations
    for (List<Sector> slist : sslist) {

        //  Loop over the first seed layer
        for (HelicalTrackHit hit1 : slist.get(0).Hits()) {

            //  Loop over the second seed layer and check that we have a hit pair consistent
            for (HelicalTrackHit hit2 : slist.get(1).Hits()) {
                  … TwoPointCircleCheck etc on candidate
```

WHERE IS SECTORING APPLIED?

- ConfirmerExtender.java:

```
Map<SeedLayer, List<HelicalTrackHit>> hitmap = new HashMap<SeedLayer, List<HelicalTrackHit>>();

//  Loop over the layers to be checked
for (SeedLayer lyr : inputseed.getUncheckedLayers()) {

    //  Create a list of hits to check on this layer
    List<HelicalTrackHit> hitlist = new ArrayList<HelicalTrackHit>();

    //  Loop over the sectors on this layer to collect hits to check
    for (Sector sector : _hmanager.getSectors(lyr)) {

        //  If there are no hits, skip this sector
        if (sector.Hits().isEmpty()) continue;

        //  See if this sector is consistent with this seed
        if (!checker.CheckSector(inputseed, sector)) continue;

        //  Add the hits for this sector
        hitlist.addAll(sector.Hits());
     }

    //  Save the list of hits in the hitmap
    if (!hitlist.isEmpty()) hitmap.put(lyr, hitlist);
}
```

CURRENT CODE

- Sector sizes:
  - dphi = 2π / nphi
  - dz (always makes + and – different sectors)
- Sector binning currently enabled in TrackerReconDriver, propagates to FastCheck

```
// enable the use of sectoring using sector binning in SeedTracker
private boolean _applySectorBinning = true;
```

- What sets the sector sizes?
  - [disabled] "autosectoring": SectorManager optimizes sector sizes
  - [overridden] default: SectorManager sets nphi=4, dz=100.0
  - [active] arbitrary code in TrackerReconDriver.initialize(): sets nphi=1, dz=100000

```
SeedTracker stFinal = new SeedTracker(sFinallist, this._useHPSMaterialManager, this.includeMS);
…
// stFinal.setSectorParams(false); //this doesn't actually seem to do anything
stFinal.setSectorParams(1, 10000);
add(stFinal);
```

PHI SECTORING

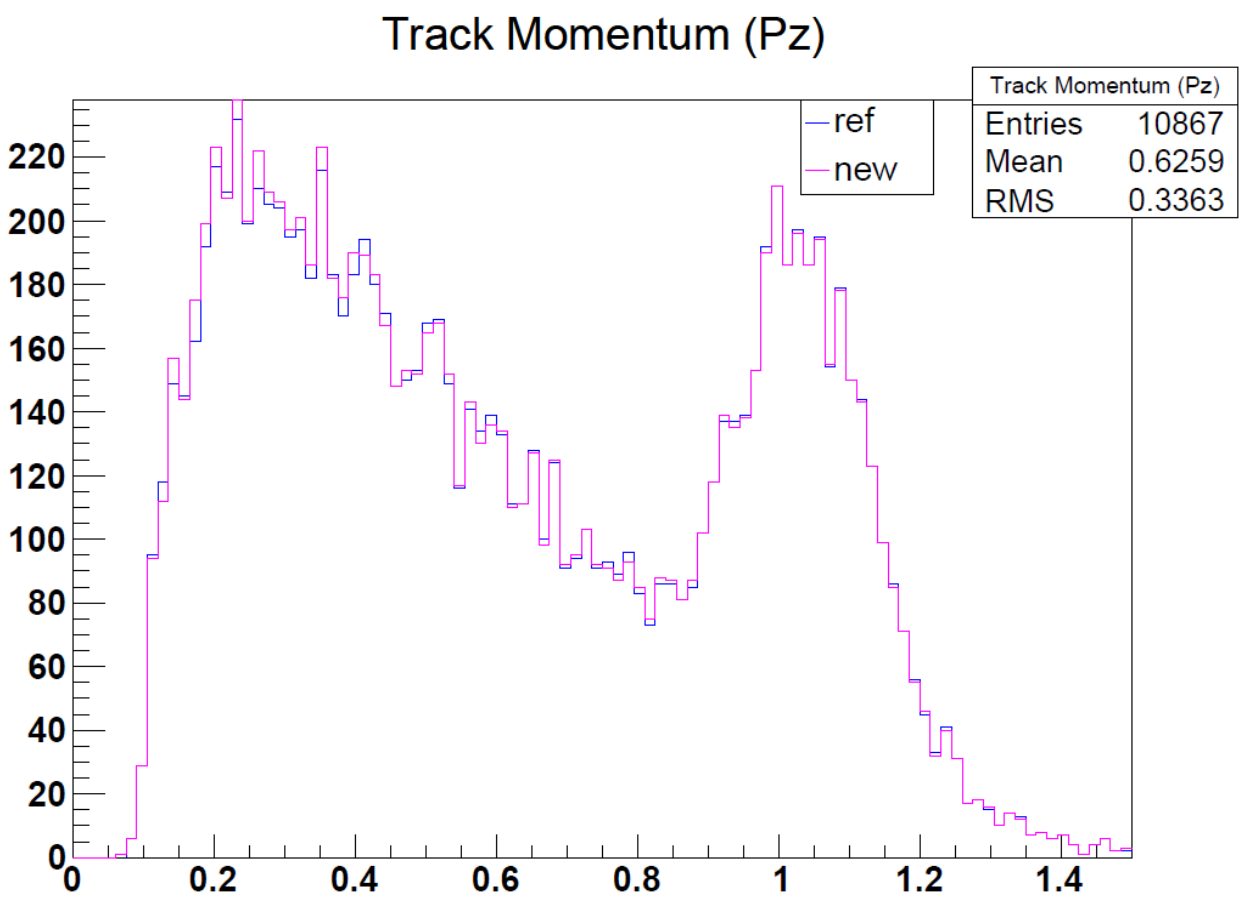
- Tried setting nphi to something >1, just to test what would happen
- Expected too high an nphi might cause loss of some tracks
- But found, setting nphi yielded more tracks (??!)
  - Run 5772 data, 10K events, strategy s123_c4_e56, SimpleAmbiguityResolver

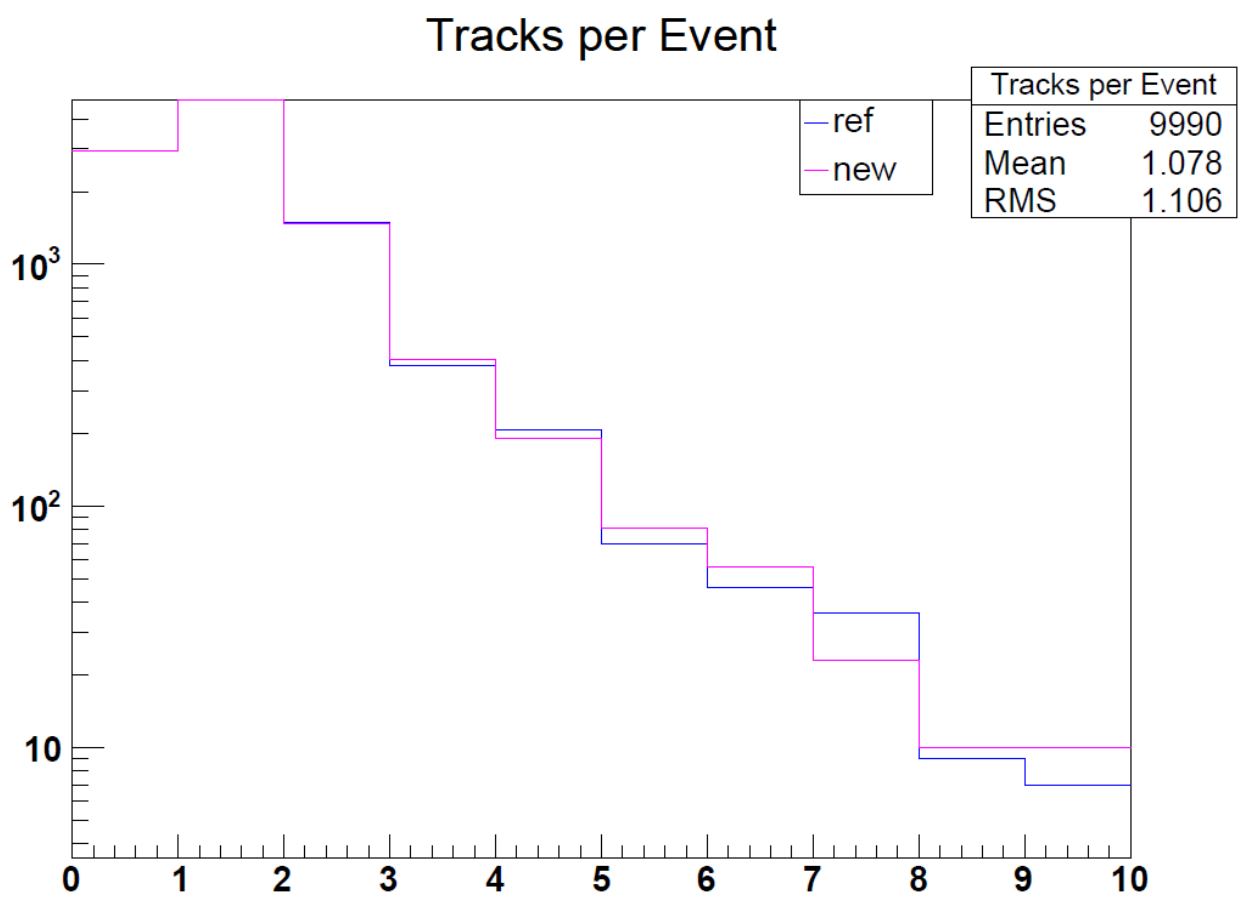
nphi	# tracks
1 (master)	10892
4 (overridden default)	10958
8	10958
16	10958
32	10957

Affects mostly
- Low p_z
- High d_0
- High χ^2
- Multi-track events

Some *different* tracks as well as *new* tracks

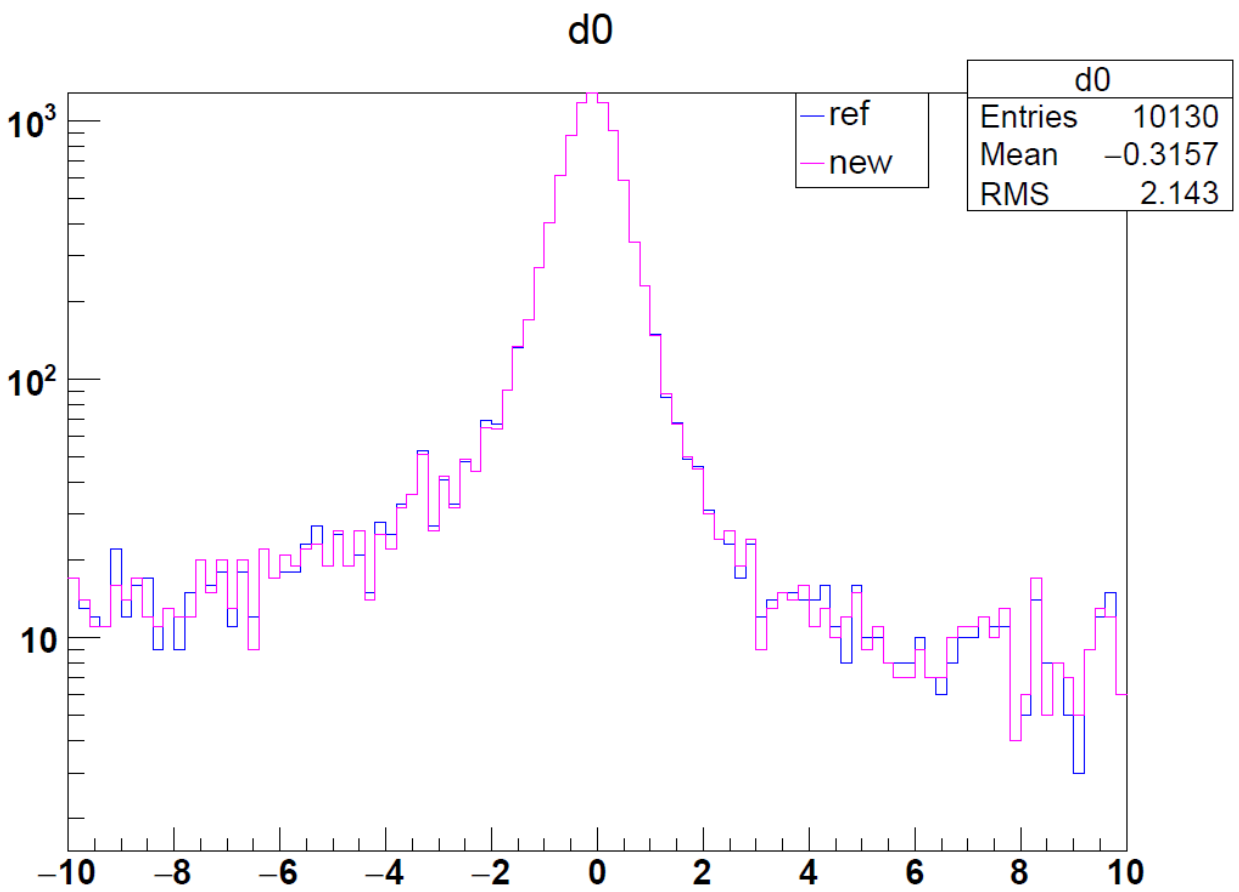
PHI SECTORING

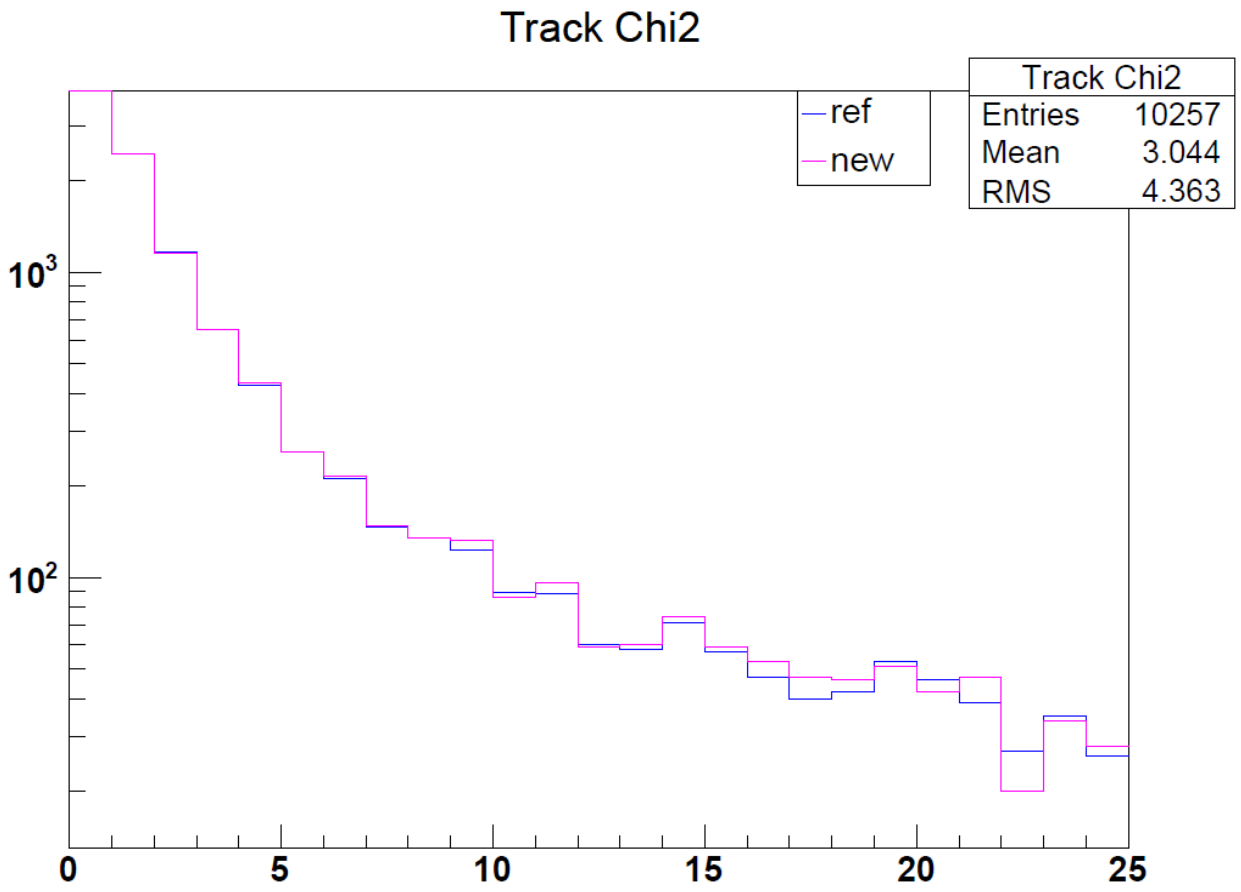




ref: nphi=1 new: nphi=4

PHI SECTORING





ref: nphi=1 new: nphi=4

RELEVANT CODE

lcsim-tracking ▸ src/main/java ▸ org.lcsim.recon.tracking.seedtracker ▸ SectorManager ▸ setSectorParams(List<SeedStrategy>, double, double) : void

```
public void setSectorParams(List<SeedStrategy> slist, double bfield, double rtrk)
    //  Find the average pTMin and MaxZ0
    double dzsum = 0.;
    double ptsum = 0.;
    for (SeedStrategy strategy : slist) {
        ptsum += strategy.getMinPT();
        dzsum += strategy.getMaxZ0();
    }
    double ptave = ptsum / nstrat;
    double dzave = dzsum / nstrat;

    //  If there is a bfield defined, set the size of a phi
    //  segmentation slice to half the change in angle for a
    //  the average minimum momentum particle
    if (bfield > 0.) {
        double RMin = ptave / (Constants.fieldConversion * bfield);
        double dphi = Math.atan(rtrk / (2. * RMin));
        nphi = (int) Math.floor(2. * Math.PI / dphi);
    }

    //  Set the z sectoring to match the average MaxZ0
    dz = dzave;
```

autosectoring

lcsim-tracking ▸ src/main/java ▸ org.lcsim.recon.tracking.seedtracker ▸ SectorManager ▸ AddHit(HelicalTrackHit) : void

```
public void AddHit(HelicalTrackHit hit) {

    //  Get the sector identifier for this hit
    String identifier = FindSectorIdentifier(hit);

    //  Retrieve the sector - create a new sector if one doesn't already exist
    Sector sector;
    if (!_sectormap.containsKey(identifier)) {
        sector = CreateSector(hit);
        _sectorlist.add(sector);
        _sectormap.put(identifier, sector);

        //  See if we need to create a new list of Sensors for this detector layer
        String lyrid = sector.LayerID();
        if (!_slistmap.containsKey(lyrid)) {
            List<Sector> slist = new ArrayList<Sector>();
            _slistmap.put(lyrid, slist);
        }

        //  Update the list of sensors for this layer
        _slistmap.get(lyrid).add(sector);

    } else {
        sector = _sectormap.get(identifier);
    }

    //  Add the hit to the sector
    sector.addHit(hit);
}
```

lcsim-tracking ▸ src/main/java ▸ org.lcsim.recon.tracking.seedtracker ▸ SeedSectoring ▸ SeedSectoring(HitManager, SeedStrategy, double, boolean)

```
public SeedSectoring(HitManager hmanager, SeedStrategy strategy,
        double bfield, boolean doSectorBinCheck) {

    _seedsectors = new ArrayList<List<Sector>>();

    FastCheck checker = new FastCheck(strategy, bfield, null);
    if(doSectorBinCheck) checker.setDoSectorBinCheck(hmanager.getSectorManager());

    //  Get the SeedLayers for this strategy
    List<SeedLayer> layers = strategy.getLayers(SeedLayer.SeedType.Seed);
    if (layers.size() != 3)
        throw new RuntimeException("Illegal Strategy "+strategy.getName()+": Number of Seed Layers is not 3");

    List<Sector> slist0 = hmanager.getSectors(layers.get(0));
    List<Sector> slist1 = hmanager.getSectors(layers.get(1));
    List<Sector> slist2 = hmanager.getSectors(layers.get(2));

    for (Sector s0 : slist0) {
        for (Sector s1 : slist1) {
            if (!checker.CheckSectorPair(s0, s1)) continue;
            for (Sector s2 : slist2) {
                if (!checker.CheckSectorPair(s0, s2)) continue;
                if (!checker.CheckSectorPair(s1, s2)) continue;
                List<Sector> slist = new ArrayList<Sector>();
                slist.add(s0);
                slist.add(s1);
                slist.add(s2);
                _seedsectors.add(slist);
            }
        }
    }
}
```

```
public boolean CheckSectorPair(Sector s1, Sector s2) {

    if (_skipchecks) return true;

    //  Calculate the maximum change in azimuth
    double dphi1 = dphimax(s1.rmin(), s2.rmax());
    double dphi2 = dphimax(s1.rmax(), s2.rmin());

    //  Calculate the angular difference between the midpoints
    double mid1 = (s1.phimax() + s1.phimin()) / 2.0;
    double mid2 = (s2.phimax() + s2.phimin()) / 2.0;
    double dmid = phidif(mid1, mid2);

    //  Calculate the half widths of the 2 sectors
    double wid1 = s1.phimax() - mid1;
    double wid2 = s2.phimax() - mid2;

    //  Check that the sectors are compatible in the bend coord
    boolean phiOK;

    phiOK = dmid < dphi1 + wid1 + wid2;
    if (!phiOK) phiOK = dmid < dphi2 + wid1 + wid2;
    if (!phiOK) return false;

    // Get the minimum and maximum path lengths
    double s1min = smin(s1.rmin());
    double s2min = smin(s2.rmin());
    double s1max = smax(s1.rmax());
    double s2max = smax(s2.rmax());

    //  Get the minimum and maximum z's
    double z1min = s1.zmin();
    double z2min = s2.zmin();
    double z1max = s1.zmax();
    double z2max = s2.zmax();

    //  Check that the sectors are compatible in the non-bend coordinate
    boolean zOK = checkz0(s1min, s1max, z1min, z1max, s2min, s2max, z2min, z2max);

    if (!zOK) return false;

    boolean zSectorOK = true;

    if(_doSectorBinCheck) {
        zSectorOK = zSectorCheck(s1,s2);
    }

    return zSectorOK;
```

lcsim-tracking ▸ src/main/java ▸ org.lcsim.recon.tracking.seedtracker ▸ FastCheck ▸ CheckSector(SeedCandidate, Sector) : boolean

```
public boolean CheckSector(SeedCandidate seed, Sector sector) {

    if (_skipchecks) return true;

    //  Get limits on r, phi, and z for hits in this sector
    double rmin = sector.rmin();
    double rmax = sector.rmax();
    double phimin = sector.phimin();
    double phimax = sector.phimax();
    double zmin = sector.zmin();
    double zmax = sector.zmax();

    //  Calculate the midpoint and half the span in phi for this layer
    double midphisec = (phimin + phimax) / 2.;
    double dphisec = 0.5 * (phimax - phimin);

    //  Check each hit for compatibility with this sector
    for (HelicalTrackHit hit : seed.getHits()) {

        //  Adjust the hit position for stereo hits
        CorrectHitPosition(hit, seed);

        //  Calculate the max track angle change between the hit and se
        double dphitrk1 = dphimax(hit.r(), rmin);
        double dphitrk2 = dphimax(hit.r(), rmax);
        double dphitrk = Math.max(dphitrk1, dphitrk2);

        //  Calculate the phi dev between the hit and midpoint of the s
        double dphi = phidif(hit.phi(), midphisec);

        //  The maximum dphi is the sum of the track bend and half the
        double dphimx = dphitrk + dphisec;
        if (dphi > dphimx) return false;

        double smin1 = smin(rmin);
        double smax1 = smax(rmax);
        double r = hit.r();
        double smin2 = smin(r);
        double smax2 = smax(r);

        //  Get the z limits for the hit
        double zlen = 0.;
        if (hit instanceof HelicalTrack2DHit) {
            zlen = ((HelicalTrack2DHit) hit).zlen();
        }
        double zmin2 = hit.z() - 0.5 * zlen;
        double zmax2 = zmin2 + zlen;

        //  Check the z0 limits
        boolean zOK = checkz0(smin1, smax1, zmin, zmax, smin2, smax2, zmin2, zmax2);

        if(!zOK) return false;

        boolean zSectorOK = true;

        if(_doSectorBinCheck) {
            zSectorOK = zSectorCheck(hit,sector);
        }

        if(!zSectorOK) return false;

    }
    return true;
}
```

lcsim-tracking ▸ src/main/java ▸ org.lcsim.recon.tracking.seedtracker ▸ FastCheck

```
protected boolean zSectorCheck(Sector s1, Sector s2) {
    return s1.zSector()==s2.zSector();
}

protected boolean zSectorCheck(HelicalTrackHit hit, Sector sector) {
    int zSector = sector.zSector();
    int zBin = this._sectorManager.ZBin(hit);
    return zBin==zSector;
}
```