1 逐步积分算法

Opensees 逐步积分算法共有 12 大类，包括 AlphaOS(α OS);BackwardEuler(Euler); CentralDifference(CDM); CollocationHS(CHS);GeneralizedAlpha(G-α),HHT;Houbolt;KR-AlphaExplicit(KR-α);Newmark; ParkLMS3(PLMS3); TRBDF;Wilson-Theta(W-θ) 法, 如图1所示。

其中：AlphaOS 包括 4 种算法，如图2所示，CDM 包括 3 种算法，如图3所示，HHT 包括 14 种算法，如图4所示，KR-AlphaExplicit 算法包括 2 种算法，如图5所示，，NEM 算法类包括 6 种算法，如图6所示。算法的具体介绍可见官网对Integrator的解释及源代码。

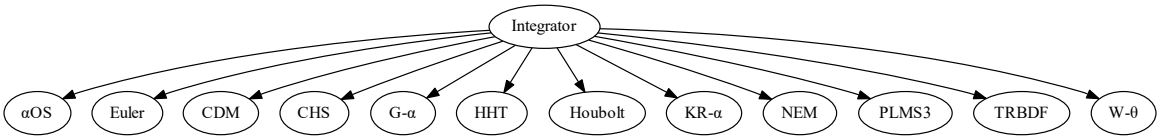

图 1: Opensees 逐步积分算法

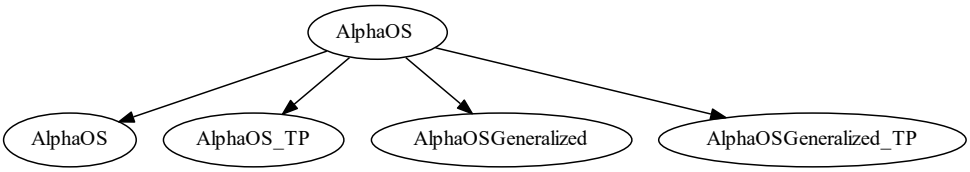

图 2: AlphaOS 算法类

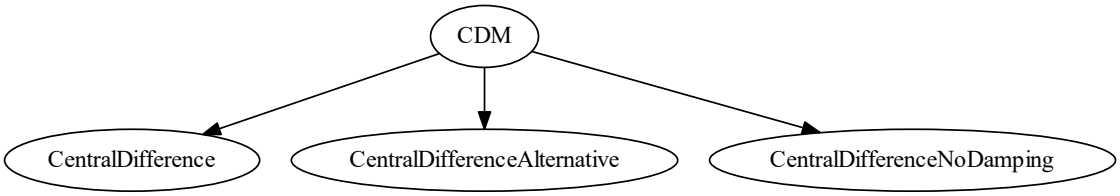

图 3: CDM 算法类

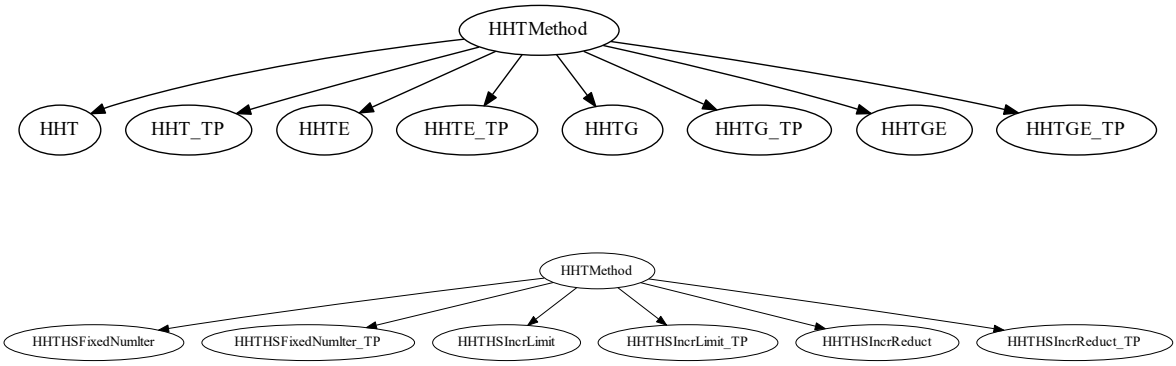

图 4: HHT 算法类

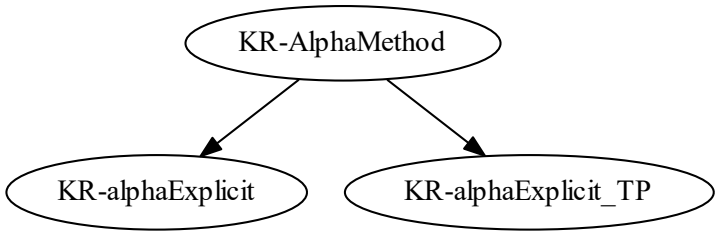

图 5: KR-AlphaExplicit 算法类

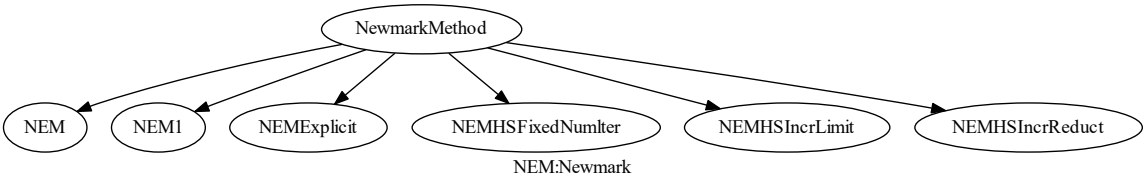

图 6: NEM 算法类

2 Element 单元

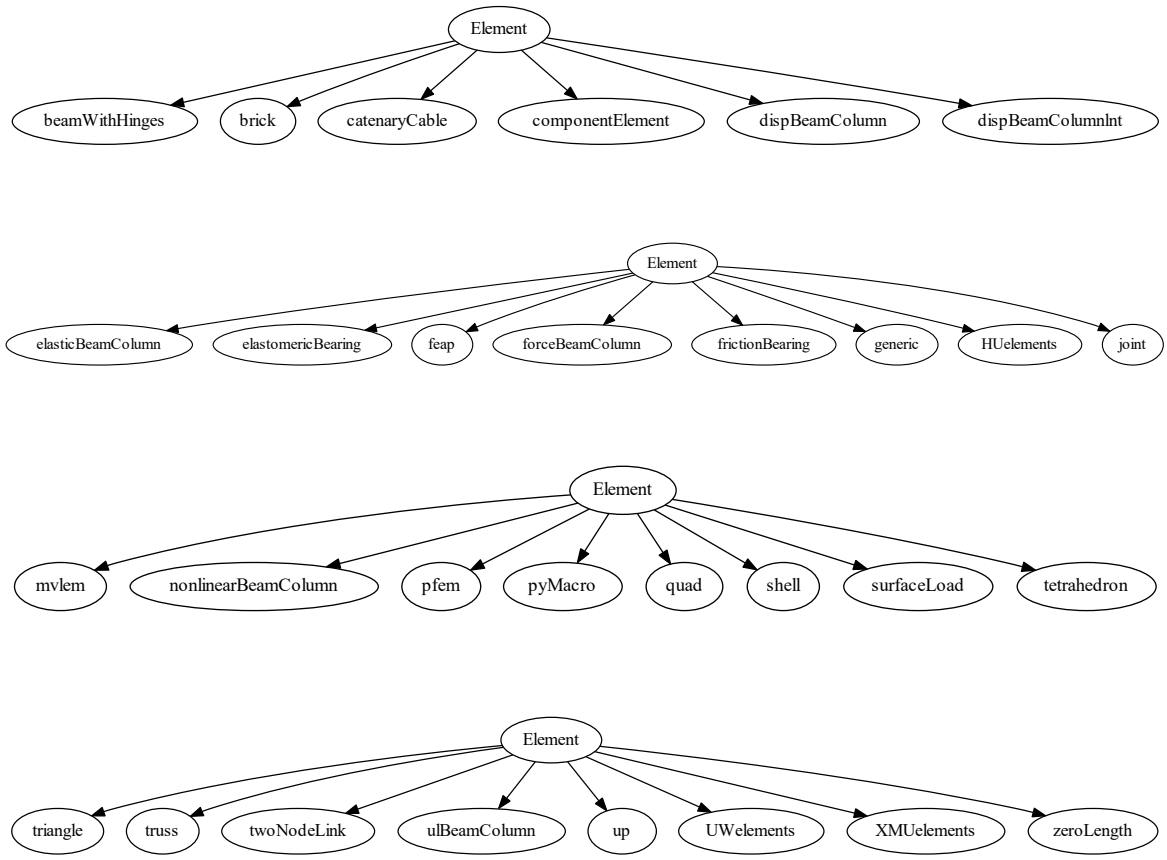

图 7: Element 分类

2.1 Truss Element

2.1.1 Truss Element

该命令用于构建一个桁架元素对象。有两种方法来构建一个桁架元素对象：一种方法是指定一个区域和一个 UniaxialMaterial 标识符：

```
element truss $eleTag $iNode $jNode $A $matTag
```

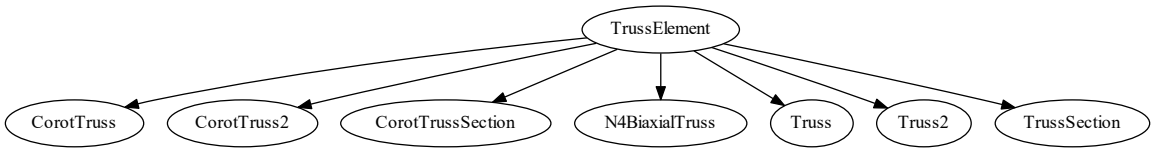

图 8: Truss 分类

另一个是指定一个 Section 标识符：

```
element truss $eleTag $iNode $jNode $secTag
  $eleTag – 独特的元素对象标签；
  $iNode $jNode – 端节点；
  $A – 元素的横截面积；
  $matTag – 与先前定义的UniaxialMaterial关联的标签；
  $secTag – 与先前定义的部分相关联的标签；
```

当用单轴材料构成时，桁架元件考虑应变率效应，因此适合用作阻尼元件。当创建 ElementRecorder 对象时，对 truss 元素的有效查询是'axialForce'，'stiff'，变形，'material matArg1 matArg2 ...'，'section sectArg1 sectArg2 ...' 接口之后会有更多查询所涉及的方法。

2.1.2 Corotational Truss Element

该命令用于构建一个 Corotational Truss（CorotTruss）元素对象。旋转公式采用一组随着元件旋转的旋转轴，从而考虑了局部和全局参照系之间的精确的几何变换。有两种方法来构造一个 Corotational Truss 元素对象：一种方法是指定一个区域和一个 UniaxialMaterial 标识符：

```
element corotTruss $eleTag $iNode $jNode $A $matTag
```

另一个是指定一个 Section 标识符：

```
element corotTruss $eleTag $iNode $jNode $secTag
$eleTag – 独特的元素对象标签；
$iNode $jNode – 端节点；
$A – 元素的横截面积；
$matTag – 与先前定义的UniaxialMaterial对象关联的标签；
$secTag – 与之前定义的Section对象关联的标记；
```

注意：当使用单轴材料对象构建时，桁架单元考虑应变率效应，因此适合用作阻尼单元。当创建一个 ElementRecorder 对象时，对一个旋转桁架元素的有效查询是'axialForce'，'stiff'，'material $matNummatArg1matArg2...''section$ secNum sectArg1 sectArg2 ...'

2.2 Elastic Beam Column Element

该命令用于构造一个 elasticBeamColumn 元素对象。构造弹性梁柱单元的论据取决于问题的维数 ndm：对于一个二维问题：

```
element elasticBeamColumn $eleTag $iNode $jNode $A $E $Iz $transfTag
```

对于一个三维问题：

```
element elasticBeamColumn $eleTag $iNode $jNode $A $E $G $J $Iy $Iz $transfTag
```

创建 ElementRecorder 时对弹性梁柱元素的有效查询是'stiffness' 和'force '。

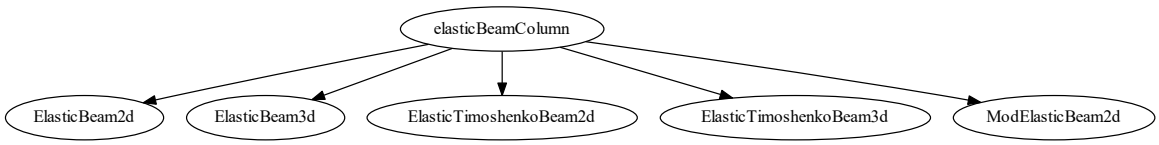

图 9: elasticBeamColumn 分类

2.3 NonLinear Beam-Column Elements

有两种基本类型的非线性梁柱单元：

1. 基于力的单元: 分布可塑性（NonlinearBeamColumn）, 内部具有弹性的可塑性（beamwithHinges）
2. 基于位移的单元: 线性曲率分布的分布可塑性（dispBeamColumn）

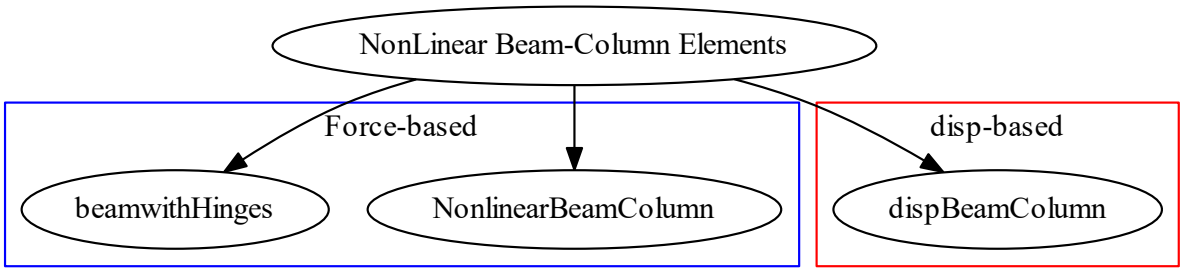

图 10: Nonlinear Beam-Column 分类

```
element nonlinearBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag <-mass $massDens> <-iter $
    maxIters $tol>
```

这个命令用来构造一个非线性的 BeamColumn 元素对象，它是基于非迭代（或迭代）力的公式，并且考虑到沿着元素的可塑性扩展

对于一个二维问题：

```
element beamWithHinges $eleTag $iNode $jNode $secTagI $Lpi $secTagJ $Lpj $E $A $Iz $transfTag <-mass $
    massDens> <-iter $maxIters $tol>
```

对于一个三维问题：

```
element beamWithHinges $eleTag $iNode $jNode $secTagI $Lpi $secTagJ $Lpj $E $A $Iz $Iy $G $J $transfTag <-
    mass $massDens> <-iter $maxIters $tol>
```

这个命令被用来构造一个 beam With Hinges 元素对象，它是基于非迭代的（或迭代的）灵活性公式，并且认为可塑性集中在元素末端的指定铰链长度上。

请注意，beamHithges 元素只在元素端定位塑性铰。

这种类型的元素将元素分为三个部分：两个端部的铰链和中间的线性元素区域。铰链是通过分配给每个以前定义的部分来定义的。每个铰链的长度也由用户指定。

```
element dispBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag <-mass $massDens>
```

此命令用于构造一个 dispBeamColumn 元素对象，它是一个分布式可塑性，基于位移的梁柱元素。

2.4 ZeroLength Elements

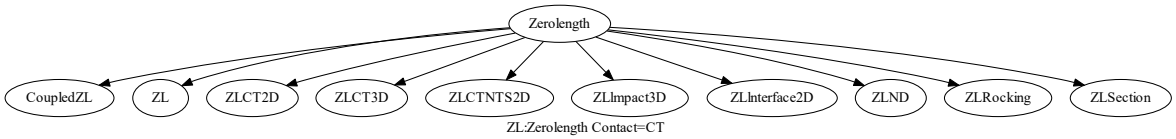

图 11: ZeroLength Elements 分类

2.5 quad Element

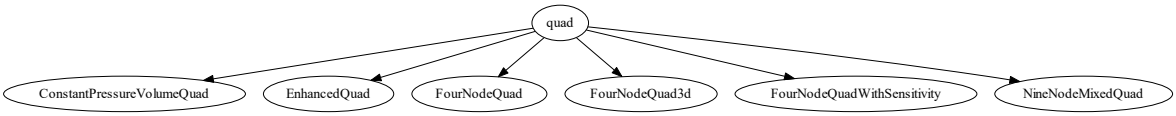

图 12: quad Elements 分类

2.6 Shell Element

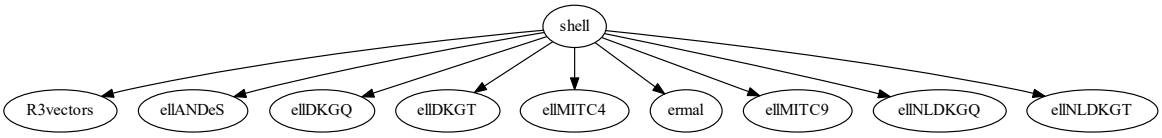

图 13: shell Elements 分类

2.7 brick Element

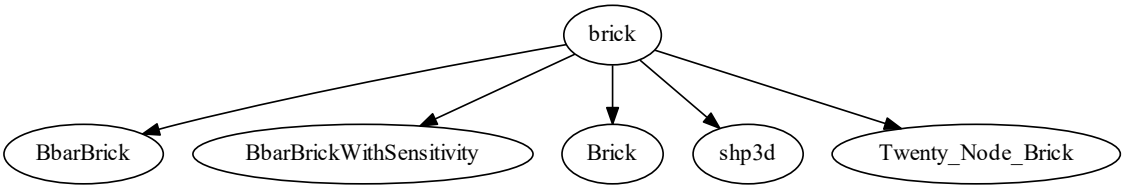

图 14: brick Elements 分类