

Numerical Integration Options for the Force-Based Beam-Column Element in OpenSees

Michael H. Scott*

January 26, 2011

A wide range of numerical integration options are available in OpenSees to represent distributed plasticity or non-prismatic section details in force-based beam-column elements, i.e., across the entire element domain $[0, L]$. For the Tcl interpreter of OpenSees, there is a specific input format for each integration option that follows a common input format for the element tag, nodes, and transformation tags. The general form of the `forceBeamColumn` command is:

```
set integration <specific integration arguments>
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

- `tag` – unique integer that will identify the element
- `ndI/ndJ` – integer tag for node I/J of the element
- `transfTag` – integer tag for geometric transformation type of the element
- `integration` – string indicating the type of numerical integration for the element and the specific integration arguments, as described in the remainder of this document. Note: the OpenSees Tcl interpreter performs a recursive parse of this string so that integration parameters can be stored in a Tcl string variable.

Integration Methods for Distributed Plasticity

Distributed plasticity methods permit yielding at any integration point along the element length.

Gauss-Lobatto Integration

Gauss-Lobatto integration is the most common approach for evaluating the response of force-based elements [3] because it places an integration point at each end of the element, where

*Associate Professor, School of Civil and Construction Engineering, Oregon State University, Corvallis, OR, 97331, michael.scott@oregonstate.edu

bending moments are largest in the absence of interior element loads.

```
set integration "Lobatto $secTag $N"  
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

Places N Gauss-Lobatto integration points along the element. The location and weight of each integration point are tabulated in references on numerical analysis [1]. The force-deformation response at each integration point is defined by the section with tag `secTag`. The order of accuracy for Gauss-Lobatto integration is $2N-3$.

Gauss-Legendre Integration

Gauss-Legendre integration is more accurate than Gauss-Lobatto; however, it is not common in force-based elements because there are no integration points at the element ends.

```
set integration "Legendre $secTag $N"  
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

Places N Gauss-Legendre integration points along the element. The location and weight of each integration point are tabulated in references on numerical analysis [1]. The force-deformation response at each integration point is defined by the section with tag `secTag`. The order of accuracy for Gauss-Legendre integration is $2N-1$.

Gauss-Radau Integration

Gauss-Radau integration is not common in force-based elements because it places an integration point at only one end of the element; however, it forms the basis for optimal plastic hinge integration methods.

```
set integration "Radau $secTag $N"  
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

Places N Gauss-Radau integration points along the element with a point constrained to be at `ndI`. The location and weight of each integration point are tabulated in references on numerical analysis [1]. The force-deformation response at each integration point is defined by the section with tag `secTag`. The order of accuracy for Gauss-Radau integration is $2N-2$.

Newton-Cotes Integration

Newton-Cotes places integration points uniformly along the element, including a point at each end of the element.

```
set integration "NewtonCotes $secTag $N"
```

```
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

Places N Newton-Cotes integration points along the element. The weights for the uniformly spaced integration points are tabulated in references on numerical analysis [1]. The force-deformation response at each integration point is defined by the section with tag `secTag`. The order of accuracy for Gauss-Radau integration is $N-1$.

Fixed Location Integration

This option allows user-specified locations of the integration points. The associated integration weights are computed by the method of undetermined coefficients (Vandermonde system).

$$\sum_{i=1}^N x_i^{j-1} w_i = \int_0^1 x^{j-1} dx = \frac{1}{j} \quad (j = 1, \dots, N) \quad (1)$$

Note that Newton-Cotes integration is recovered when the integration point locations are equally spaced.

```
set locations "0.0 0.2 0.5 0.8 1.0"
set secTags "1 2 2 2 1"
set integration "FixedLocation $N $secTags $locations"
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

Places N integration points along the element, whose locations are defined in a Tcl list `locations` on the natural domain $[0, 1]$. The force-deformation response at each integration point is defined by the sections with tags stored in the Tcl list `secTags`. Both the `locations` and `secTags` lists should be of length N . The order of accuracy for Fixed Location integration is $N-1$.

Low Order Integration

This option is a generalization of the Fixed Location and User Defined integration approaches and is useful for moving load analysis [2]. The locations of the integration points are user-defined, while a selected number of weights are specified and the remaining weights are computed by the method of undetermined coefficients.

$$\sum_{i=1}^{N_f} x_{fi}^{j-1} w_{fi} = \frac{1}{j} - \sum_{i=1}^{N_c} x_{ci}^{j-1} w_{ci} \quad (2)$$

Note that Fixed Location integration is recovered when N_c is zero.

```
set locations "0.0 1.0 0.2 0.5 0.8"
set weights "0.2 0.2"
set secTags "1 1 2 2 2"
```

```

set integration "LowOrder $N $secTags $locations $weights"
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration

```

Places N integration points along the element, which are defined in the Tcl list `locations` on the natural domain $[0, 1]$. The force-deformation response at each integration point is defined by the section tags stored in the Tcl list `secTags`. Both the `locations` and `secTags` lists should be of length N . The weights at user-selected integration points are specified (on $[0, 1]$) in the `weights` list, which can be of length N_c equals 0 up to N . These specified weights are assigned to the first N_c entries in the `locations` and `secTags` lists, respectively. The order of accuracy for Low Order integration is $N - N_c - 1$.

Note: N_c is determined from the length of the `weights` list. Accordingly, `FixedLocation` integration is recovered when `weights` is an empty list and `UserDefined` integration is recovered when the `weights` and `locations` lists are of equal length.

Mid-Distance Integration

This option allows user-specified locations of the integration points. The associated integration weights are determined from the midpoints between adjacent integration point locations. $w_i = (x_{i+1} - x_{i-1})/2$ for $i = 2 \dots N-1$, $w_1 = (x_1 + x_2)/2$, and $w_N = 1 - (x_{N-1} + x_N)/2$.

```

set locations "0.1 0.2 0.5 0.8 0.9"
set secTags "1 2 2 2 1"
set integration "MidDistance $N $secTags $locations"
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration

```

Places N integration points along the element, whose locations are defined in a Tcl list `locations` on the natural domain $[0, 1]$. The force-deformation response at each integration point is defined by the sections with tags stored in the Tcl list `secTags`. Both the `locations` and `secTags` lists should be of length N . This integration rule can only integrate constant functions exactly since the sum of the integration weights is one.

For the `locations` shown above, the associated integration weights will be 0.15 0.2 0.3 0.2 0.15.

User Defined Integration

This option allows user-specified locations and weights of the integration points.

```

set locations "0.1 0.3 0.5 0.7 0.9"
set weights "0.2 0.15 0.3 0.15 0.2"
set secTags "1 2 2 2 1"
set integration "UserDefined $N $secTags $locations $weights"
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration

```

Places N integration points along the element, which are defined in the Tcl list `locations` on the natural domain $[0, 1]$. The weight of each integration point is defined in the Tcl list `weights`, also on the $[0, 1]$ domain. The force-deformation response at each integration point is defined by the sections with tags stored in the Tcl list `secTags`. The `locations`, `weights`, and `secTags` lists should be of length N . In general, there is no accuracy for this approach to numerical integration.

Plastic Hinge Integration Methods

Plastic hinge integration methods confine material yielding to regions of the element of specified length while the remainder of the element is linear elastic. A summary of plastic hinge integration methods is found in [4].

Midpoint Hinge Integration

Midpoint integration over each hinge region is the most accurate one-point integration rule; however, it does not place integration points at the element ends and there is a small integration error for linear curvature distributions along the element.

```
set integration "HingeMidpoint $secTagI $lpI $secTagJ $lpJ $secTagE"
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

The plastic hinge length at end I (J) is equal to lpI (lpJ) and the associated force-deformation response is defined by the section with tag `secTagI` (`secTagJ`). The force-deformation response of the element interior is defined by the section with tag `secTagE`. Typically, the interior section is linear-elastic, but this is not necessary.

Endpoint Hinge Integration

Endpoint integration over each hinge region moves the integration points to the element ends; however, there is a large integration error for linear curvature distributions along the element.

```
set integration "HingeEndpoint $secTagI $lpI $secTagJ $lpJ $secTagE"
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

The plastic hinge length at end I (J) is equal to lpI (lpJ) and the associated force-deformation response is defined by the section with tag `secTagI` (`secTagJ`). The force-deformation response of the element interior is defined by the section with tag `secTagE`. Typically, the interior section is linear-elastic, but this is not necessary.

Radau Hinge Integration

Two-point Gauss-Radau integration over each hinge region places an integration point at the element ends and at $2/3$ the hinge length inside the element. This approach represents

linear curvature distributions exactly; however, the characteristic length for softening plastic hinges is not equal to the assumed plastic hinge length.

```
set integration "HingeRadauTwo $secTagI $lpI $secTagJ $lpJ $secTagE"
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

The plastic hinge length at end I (J) is equal to lpI (lpJ) and the associated force-deformation response is defined by the section with tag `secTagI` (`secTagJ`). The force-deformation response of the element interior is defined by the section with tag `secTagE`. Typically, the interior section is linear-elastic, but this is not necessary.

Modified Radau Hinge Integration

Modified two-point Gauss-Radau integration over each hinge region places an integration point at the element ends and at $8/3$ the hinge length inside the element. This approach represents linear curvature distributions exactly and the characteristic length for softening plastic hinges is equal to the assumed plastic hinge length.

```
set integration "HingeRadau $secTagI $lpI $secTagJ $lpJ $secTagE"
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

The plastic hinge length at end I (J) is equal to lpI (lpJ) and the associated force-deformation response is defined by the section with tag `secTagI` (`secTagJ`). The force-deformation response of the element interior is defined by the section with tag `secTagE`. Typically, the interior section is linear-elastic, but this is not necessary.

Regularized Hinge Integration

If it is known *a priori* whether to use distributed plasticity integration for strain-hardening response or a plastic hinge method for strain-softening response, the regularized approach developed in [5] is suggested.

```
set integration "RegularizedHinge distType nIP? secTagI? lpI? zetaI? secTagJ?
lpJ? zetaJ? secTagE?"
element forceBeamColumn $tag $ndI $ndJ $transfTag $integration
```

The plastic hinge length at end I (J) is equal to lpI (lpJ) and the associated force-deformation response is defined by the section with tag `secTagI` (`secTagJ`). The force-deformation response of the element interior is defined by the section with tag `secTagE`.

The `distType` argument is the underlying distributed plasticity integration approach, either Lobatto, Legendre, Radau, or NewtonCotes, with `nIP` integration points. The arguments `zetaI` and `zetaJ` indicate the distance inside the element ends (I and J , respectively) that additional integration points are located in order to enforce numerical consistency in

the case of strain-hardening response. Typical values for `zetaI` and `zetaJ` range from 0.1% to 1.0% of the element length. Further information on this final group of arguments can be found in [5].

References

- [1] M. Abramowitz and C. A. Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, NY, 9th edition, 1972.
- [2] A. Kidarsa, M. H. Scott, and C. C. Higgins. Analysis of moving loads using force-based finite elements. *Finite Elements in Analysis and Design*, 44(4):214–224, 2008.
- [3] A. Neuenhofer and F. C. Filippou. Evaluation of nonlinear frame finite-element models. *Journal of Structural Engineering*, 123(7):958–966, July 1997.
- [4] M. H. Scott and G. L. Fenves. Plastic hinge integration methods for force-based beam-column elements. *Journal of Structural Engineering*, 132(2):244–252, February 2006.
- [5] M. H. Scott and O. M. Hamutcuoglu. Numerically consistent regularization of force-based frame elements. *International Journal for Numerical Methods in Engineering*, 76(10):1612–1631, 2008.