Simulating the Transport of Very Flexible Sheets: Beam-Type Solutions

Ted Diehl
Senior Development Engineer
Eastman Kodak Company
Kodak Park, Bldg. 23
Rochester, NY 14652-4392

ABAQUS Users' Conference Proceedings
Aachen, Germany
June 23–25, 1993

Abstract

Analyzing the transport of flexible sheet media such as paper or film has traditionally been extremely difficult due to the complexities of large displacements, large rotations, intermittent contact, and friction. This paper describes methods of obtaining accurate solutions to these problems with the nonlinear finite element methods available in ABAQUS. A user-supplied MPC (Multi-Point Constraint) subroutine is developed to simulate the changing boundary conditions at the point where the sheet leaves the drive rollers. The user MPC subroutine is found to be more efficient and robust than the use of either rigid surfaces or multiple steps with ever-changing boundary conditions. Both quasi-static and dynamic examples are presented and compared to finite difference solutions of the Elastica.

1.0 Introduction

A common problem in the design of equipment that transports thin, very flexible sheets such as paper or film is that the medium undergoes large displacements, large rotations, and continuously changing boundary conditions as it is transported. In many design problems, a sheet is often pushed by a set of nip drive rollers through channels with complicated contours (see Figure 1). Because the sheet is very thin, it deforms primarily due to bending and experiences only small strains. Traditionally, problems of this type have been analyzed with the equations of the Elastica. The basic theory of the Elastica governs an inextensible, large displacement, large rotation, infinitesimally strained beam.[1] The Elastica equations can be very nonlinear and difficult to solve. For problems involving nontrivial boundary conditions such as intermittent contact, solution of the Elastica requires numerical methods, typically finite difference techniques. Very accurate solutions to these problems can also be obtained with the nonlinear finite element methods available in ABAQUS. The generic nature of a commercial finite element code is extremely advantageous when developing solutions to complicated sheet-transport problems.

1. See Timoshenko 1961 and Benson 1981 for further details on the theory of the Elastica.

2.0 Problem Description

The basic problem addressed is the process of driving a very flexible sheet with a set of nip-drive rollers through a contoured channel, depicted schematically in Figure 1. For all cases in this paper, only beam-type solutions are presented.[1] All solutions are restricted to the X-Y plane with gravity acting in the negative Y direction as shown in Figure 1. Typical analysis goals are to determine the load requirements of the drive rollers, deflected shapes of the sheet, initial sheet curl effects, and potential jamming locations of the medium as it is transported by the drive nips into a channel. The analysis must be capable of simulating the ever-changing *clamped* boundary conditions that the nip exerts on the sheet as it is transported as well as the changing contact conditions that the sheet experiences inside the channel. Several methods are available in ABAQUS to analyze this type of problem; some are very efficient and robust, while others are not as effective.

2.1 Methods of Simulating the Boundary Conditions of the Drive Nip

The three principal methods that can model the transport of media by drive nips are (1) the *BOUNDARY method, (2) the user MPC method, and (3) the rigid-surface method. These methods are depicted in Figure 2.

2.1.1 *BOUNDARY Method

This method uses multiple steps with discrete, ever-changing boundary conditions. It is the most direct method of simulating the boundary conditions produced by drive nips. As it moves through the nip, the sheet is clamped by the rollers and subjected to an enforced displacement in the X direction. The boundary conditions for the sheet under the nip are

$$u_x = \text{specified} \qquad u_y = 0 \qquad \theta_z = 0 \qquad \text{(EQ 1)}$$

Because it is also subjected to gravity loads, the sheet to the left of the nip (as depicted in Figures 1 and 2) must also be supported. Hence, the boundary conditions of equation 1 also apply to all the medium left of the drive nips. As the sheet transports to the right, the boundary conditions must be removed from those nodes that have passed the nip. This is accomplished with multiple steps that discretely remove the boundary conditions from the nodes that will have passed the nip by the end of each given step.

2.1.2 User MPC Method

This method simulates the nip boundary conditions by using *rigid beam* MPCs that connect the farthest left node of the sheet to all other nodes in the sheet. With respect to the MPC constraint equations, the farthest left node would be the independent DOF (Degree Of Freedom) and all other nodes would be dependent DOFs. This is depicted in Figure 2. The standard BEAM MPC in the ABAQUS library (a MPC representing a rigid beam between two nodes) cannot be used for this purpose because there is no method of *turning off* the MPC relationship for those dependent nodes that have passed the drive nip. To accomplish this, a user subroutine is written for a rigid-beam MPC that has internal logic to apply and remove the MPC relationships for the nodes that are considered inside (left of) or outside (right of) the drive nip, respectively.[2] The maximum error in nip clamp location with this method is one element length. For typical models, this error is very small.

The user MPC subroutine consists of two main components: constraint equations between independent and dependent degrees of freedom, and on/off logic to determine when the MPC should be applied or removed.

1. Shell effects in the Z direction are assumed negligible.

2. See Diehl 1993 for detailed derivations of the MPC relationships and internal on/off MPC logic, and for a complete FORTRAN listing of the user-supplied MPC subroutine.

The constraint equations are the same as those developed in the ABAQUS/Standard Users's Manual (ABAQUS, 1992). The MPC subroutine is coded for MPC MODE = NODE which means that the constraint equations for all three dependent DOF's (u_x, u_y, θ_z) are imposed at the same time for a given call to the subroutine. To apply general on/off logic capabilities, the user supplies additional *dummy* nodes on the *MPC card that are used to define the drive nip location. The user subroutine listed in the ABAQUS/Standard Users's Manual requires slight modification because of the additional dummy nodes. Additional derivatives of the constraint functions with respect to the dummy nodes must be computed as well as matrix DOF identifiers for the dummy nodes. These additional constraint function derivatives are trivial since they are all zero.

Figure 3 shows the two types of on/off logic that are coded into the user subroutine. For TYPE = 21, the MPC subroutine only checks to see if the X coordinate of a given dependent node is past a single dummy node (depicted as node 99 in Figure 3). For most cases, this check is sufficient. However, if the loading is such that any node would attempt to deflect such that its X location is less than that of the dummy node ($t = j$ in Figure 3), the solution will die. This is because the MPC will be reapplied to the dependent node which in turn will cause this node to be constrained to its original length from the independent node (node 100 in the figure). The second type of on/off logic, TYPE = 22, uses a more complex checking algorithm to determine if a given dependent node should be constrained or not. This second MPC type checks whether or not a given node is inside a four-sided region defined by four dummy nodes. This quadrilateral is divided into two triangles and the algorithm uses the shape functions of a triangle to determine a given node's status. If the node is inside the quadrilateral, the MPC is applied; otherwise it is removed. This additional checking adds little to the computational costs of the solution. Both MPC types, TYPE = 21 and TYPE = 22, are capable of simulating problems that involve *pushing* sheets out of a nip or *pulling* sheets into a nip.

Care must be taken when coding the MPC subroutine with logic that turns the constraint equations on and off throughout the analysis. For each MPC defined, the MPC subroutine is called three times during each analysis iteration. The first call is to compute the transformation matrix required for elimination of dependent DOFs from the system matrix. After the system matrix is solved, the MPC subroutine is called two more times: once again to compute the transformation matrix[1] and once to recover the dependent DOF values. It is important that the MPC state (on/off) does not change during an iteration to avoid a change in the transformation matrix definition. The checking logic of the subroutine must use the ABAQUS supplied variable UINIT (displacements at the beginning of the iteration) to determine on/off logic. The ABAQUS supplied variable U (current values of displacements) must be used for all other constraint-type calculations. This method does present the possibility that the MPC constraint logic may lag the solution by one iteration. In practice, as long as two or more iterations are performed in a given increment, this will not occur. ABAQUS can be forced to almost always take at least two increments by entering the following cards in the first *STEP section of the input deck:

```
*CONTROLS, PARAMETER=FIELD
5.0E-30, , , ,0.005
*CONTROLS, PARAMETER=TIME
, ,1
```

The first two lines make the initial convergence criterion for any given increment unreasonably tight and almost impossible to pass. The command sets the alternative convergence criterion to the default initial convergence (0.5%). The last two lines instruct ABAQUS to use the alternative convergence criterion after one unsuccessful iteration.

1. This is recomputed because the MPC subroutines do not store it the first time.

2.1.3 Rigid-Surface Method

A third potential method of simulating the drive nip is the use of frictionless rigid surfaces. Figure 2 depicts two potential forms of this method. The essence of the method is that the driving force (or more likely displacement) is applied at the far left edge of the sheet. If the sheet is constrained within a very narrow channel left of the drive nip, it will be constrained from buckling and will behave as if it were actually being driven at the nip.[1] There are three basic techniques of simulating this effect with ABAQUS: use two narrowly spaced rigid surfaces, use one rigid surface that loops back over itself with a narrow gap, or use one rigid surface with a distributed pressure load acting normal to the sheet pushing it against the rigid surface. The last technique requires a user-defined DLOAD subroutine that turns the pressure off when a given element passes the drive nip location. This is not very practical. The first two rigid surface techniques are very simple to code in an ABAQUS input deck but are not computationally efficient or robust when simulating drive nips in actual problems.

Rigid surfaces are required to simulate the effects of the channel that the sheet is driven into. Using a single rigid surface that loops back over itself as opposed to two unique rigid surfaces when modelling a closed channel usually provides faster solutions because the total number of unknown variables is less. Rigid elements use Lagrange multipliers to evaluate contact and add additional DOFs to the system of equations. By using one rigid surface instead of two, the number of Lagrange multipliers introduced by the rigid surface is reduced by a factor of two.

2.2 Other Common Calculations

Typically two other components for the simulation of sheet transport are important: modeling the effects of initial sheet curl, and computing the relative speed ratio between the leading tip of the sheet and the drive nip.

2.2.1 Initially curled media

The schematic in Figure 1 depicts a flat sheet being driven into a channel. Often the sheet is initially nonflat and has initial curl: free from loads, the sheet will have some initial radius of curvature. With beam-type solutions, a sheet that is initially curled about an axis parallel to the lead edge of the sheet (curled about the Z axis in Figure 1) can be analyzed easily. If the curl is about the X axis (Figure 1), then beam-type solutions are more complicated and shell-type solutions are usually preferred. We will only consider media curled about the Z axis. One way to model this is to make the original geometry in the shape of the curled sheet. This approach causes many problems and is very inefficient. The geometry of the sheet is no longer defined by a straight line and getting it flat in order to drive it through the nip is not always a trivial task. A more efficient and robust technique is to model the sheet as initially flat with the appropriate initial stress state in the sheet. For a curled sheet, only bending stresses are present and the stress state is easily calculated from simple beam equations. The initial stress state is defined in ABAQUS with the *INITIAL CONDITIONS, TYPE=STRESS option. *Although the User Manual requests stress information, for beam elements defined by *BEAM GENERAL SECTION, the values input for this option are initial section forces and moments and not initial stresses (this is not specified in the manual).* For a beam with an initial radius of curvature R, the equation for the initial bending moment is

$$M = \frac{EI}{R} \qquad \text{(EQ 2)}$$

1. This is valid since the sheet is orders of magnitude stiffer in axial compression than in bending.

where E is the elastic modulus and I is the area cross-sectional moment of inertia for the beam. Equation 2 is valid for large rotations and small strains. Using the *INITIAL CONDITIONS method keeps the input deck short and avoids the initial step of flattening out of the sheet prior to driving it by the nip rollers.

2.2.2 Quasi-static relative speed calculations

The relative speed between the leading tip of the sheet and the drive nip is important for timing considerations. Dynamic simulations directly give this information but are more costly than quasi-static simulations. Velocities, ignoring inertial effects, can be computed from a quasi-static solution by taking numerical derivatives of the displacements of the nodes of interest with respect to the quasi-static time scale. Alone, these velocities are not physical since the time scale is purely artificial. The relative velocities between two different nodes is physical and very useful. To decrease the amount of chatter in the numerical derivatives, a general three-point Lagrange polynomial finite difference derivative formula for unequally spaced nodes is used. The general form for the derivative, *f'(x),* is (Burden, 1985)

$$f'(x_j) = f(x_0)\left[\frac{2x_j - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)}\right] + f(x_1)\left[\frac{2x_j - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)}\right] + f(x_2)\left[\frac{2x_j - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)}\right] \quad \text{(EQ 3)}$$

This formula can compute a three-point forward difference derivative (j = 0), three-point central difference derivative (j = 1), or three-point backward difference derivative (j = 2). For the quasi-static ABAQUS solution, the step increment measures the pseudotime (denoted by x in equation 3) and the nodal displacements are the dependent variables (denoted by the function *f(x)* in equation 3) for which pseudovelocities are calculated. In general, the three-point central difference derivative is used mostly except for the first and last data points for which the forward and backward difference versions of equation 3 are used, respectively. The relative speed of a given node with respect to the drive nip is

$$|v_{rel}| = \sqrt{\frac{v_x^2 + v_y^2}{\tilde{v}_x^2 + \tilde{v}_y^2}} \quad \text{(EQ 4)}$$

where v_x and v_y are the pseudovelocities in the x and y direction of any given node and $\tilde{v}_x$ and $\tilde{v}_y$ are the pseudovelocities of the drive point on the sheet.

3.0 Example Problems

Five example problems, three quasi-static and two dynamic, are computed to demonstrate the capabilities of these methods in simulating sheet transport. For the quasi-static solutions: elastic modulus equals $2.38\text{x}10^5$ psi, shear modulus equals $1.19\text{x}10^5$ psi, and mass density equals $6.92\text{x}10^{-5}$ lbf sec^2 in^{-4}. For the dynamic solutions, elastic modulus equals $1.87\text{x}10^5$ psi, shear modulus equals $9.36\text{x}10^4$ psi, and mass density equals $6.40\text{x}10^{-5}$ lbf sec^2 in^{-4}. The sheet in all solutions has a width (Z direction) of 11.0 inches and a thickness of 0.004 inch. For simplicity, all solutions presented ignore frictional effects.[1] Electrostatic and aerodynamic effects are also not included in the analyses. Since only elastic deformations are considered, the *BEAM GENERAL SECTION option is used instead of the *BEAM SECTION option for defining the beam element properties. This can save in some cases up to 35% CPU time since numerical integration through the

1. The new algorithms in V5.2 are robust and handle friction much better than V4.9 for these types of problems.

thickness of the beam is avoided at each iteration. All problems in this paper were computed with ABAQUS/Standard V5.2 on a Sun SPARCstation 10-30.

3.1 Quasi-Static Examples

Three quasi-static examples are computed: feeding a straight sheet into a gravity field, feeding a straight sheet onto a circular drum, and feeding straight and curled media into both open and closed channels. The results of these examples are shown in Figures 4 through 7. In all cases, inertial dynamic effects are ignored.

3.1.1 Feeding a thin sheet into a gravity field

The simulation of feeding a thin sheet into a gravity field (Figure 4) is used as a basis for comparing the efficiency of the many different beam elements available in ABAQUS as well as the different methods of simulating the boundary conditions produced by the drive nips. A finite difference solution based on the equations of the Elastica (Stack, 1992) provides the benchmark. Tables 1 through 3 summarize computational requirements for the different beam formulations tested. For solutions listed in Tables 1 and 2, the initial increment size for each step is 100% of the total step while solutions listed in Table 3 have an initial increment size of 5% of the total step. Table 4 compares efficiency and output storage requirements for solutions with B21H elements and various methods of simulating the nip (see Section 2.1 for discussion of these methods).

For all elements and nip simulation methods, the displacement results are very similar and agree well with the Elastica solution of Stack (Figure 4). The results in Tables 1 through 3 indicate that the B21H element is the most reliable and efficient element for problems of this type. This is as expected since the hybrid elements are specially formulated for problems were the axial stiffness is very large compared to the bending stiffness. Table 4 indicates that both rigid surface techniques are extremely expensive compared with the other drive nip techniques. The *BOUNDARY method and the user MPC method were very efficient with respect to CPU time. However, the *BOUNDARY method creates large amounts of file output and generation of the input deck requires more effort than the user MPC method. Also, in more complex simulations, the *BOUNDARY method is very inefficient because the method essentially is equivalent to a fixed time-step algorithm.

3.1.2 Feeding a thin sheet onto a frictionless circular drum

The results of this simulation are found in Figure 5 and are used to verify the method of computing the quasi-static relative tip speed of the sheet (equations 3 and 4). A finite difference based Elastica solution (Benson, 1983) provides the benchmark. This case has no gravity loading. The nip is simulated using the user MPC method with two steps: one step for feeding up to initial contact with the drum and the second step for feeding onto the roller. Two steps are used to obtain an accurate prediction of relative tip speed at first contact with the drum. As depicted in the plots of relative tip speed and rigid surface reaction force, the ABAQUS solution compares well with the benchmark. Computation time for the solution was 82 CPU seconds.

3.1.3 Simulation of feeding straight and curled media in both open and closed channels

Figures 6 and 7 demonstrate the capabilities of the user MPC method on a real-world design problem with gravity loading included. Four separate solutions are presented: feeding straight media into an open channel, feeding curled media into an open channel, feeding straight media into an closed channel, and feeding curled media into an closed channel. Computation times for each solution ranged from 240 to 295 CPU seconds. As presented in both Figures 6 and 7, the simulations are capable of predicting the media roll-over (jam) in an open channel due to initial curl and the successful feeding for the more robust closed-channel design. Feeder force and relative tip velocities are displayed in Figure 7. As seen in Figure 6 for the case with initial curl and an open channel, the tip of the sheet begins to climb the channel but then slide back down. Hence, there should be a point at which the pseudovelocity and the relative velocity of the tip are zero. Figure 7 shows this value to be slightly greater than zero. This error is caused by the robust automatic incrementation algorithms of ABAQUS taking too large of a pseudotime increment. A more accurate solution can be obtained by limiting the largest increment size allowed for the solution.

3.2 Dynamic Examples

Two dynamic cases, Figures 8 and 9, simulate a sheet being dynamically driven into a gravity field. The nip feeder is again simulated with the user MPC method. The sheet is 8.5 inches in length and is divided into 100 B21H elements. These solutions include inertial dynamic effects. Finite difference solutions based on the equations of the Elastica (Stolte, 1992) provide the benchmarks. Computation times for the two solutions in Figure 8 range from 110 to 150 CPU seconds each. The one solution in Figure 9 requires 1154 CPU seconds. All simulations compare well with the benchmark. For the extension/retraction simulation of Figure 9, the solution at the beginning of the retraction phase has very large moment HALFTOL residuals (100 times typical moments) implying that significant time integration errors are most likely present. The solution of Stolte does not check for such a variable and may also be in question in this region. When the HALFTOL parameter is tightened, the ABAQUS solution failed to converge during retraction. It should be noted that the simulation depicted in Figure 9 is very extreme. It is simulating a 11.0 inch sheet of paper being driven out of the nip and then pulled back into the nip in a total of 0.86 second.

4.0 Conclusions

Analyzing the transport of flexible sheet media can efficiently be performed in ABAQUS with the use of a user-defined MPC subroutine with on/off logic. Both quasi-static and dynamic analyses are possible. The user MPC method is found to be more efficient and robust than multiple steps with ever-changing boundary conditions or closely-spaced rigid surfaces. For these thin-media problems, the B21H beam element is found to be the most efficient formulation of those available in ABAQUS. For analyzing initial sheet curl, the most efficient and robust method is to impose initial conditions of beam section moments on a flat model of the sheet as opposed to modelling the sheet with curled geometry.

Acknowledgments

Thanks to Jean Claude Ramirez and Ian Stevenson from HKS for their detailed explanation of the MPC routines in ABAQUS. Thanks to Ken Stack, Jim Stolte and Richard Benson for supplying the benchmarks.

5.0 References

ABAQUS/Standard Users's Manual, V 5.2, 1992, section 5.2.11

Benson, R., C., "The Deformation of a Thin, Incomplete, Elastic Ring in a Frictional Channel," *Journal of Applied Mechanics*, Vol. 48, No. 4, Dec. 1981, pp. 895-891.

Benson, R. C., "The Feeding of a Thin, Inextensible Beam onto a Rigid Surface," MFSP Internal Report, Department of Mechanical Engineering, University of Rochester, NY, 1983.

Burden, R. L., Faires, J. D., *Numerical Analysis*, Third Edition, Prindle, Weber & Schmidt, Boston, 1985.

Diehl, T., "Formulation of a User MPC to Simulate Beam-Type Transport Problems," MFSP Internal Report, Department of Mechanical Engineering, University of Rochester, NY, 1993.

Stack, K. D., Private communication, 1992.

Stolte, J., “An Extending Dynamic Elastica: Analysis of Deformation Due To Inertial Loads and Impact With a Surface,” Ph.D. Dissertation, Department of Mechanical Engineering, University of Rochester, NY, 1992.

Timoshenko, S. P., Gere, J. M., *The Theory of Elastic Stability*, McGraw-Hill, New York, 1961.

TABLE 1. Gravity simulation, * BOUNDARY method, 50 steps

Elem Type	# of Elem.	# of Variables	# of Inc.	# of Cutbacks	Total Iterations	CPU Secs.
b21h	50	253	50	0	135	117
b21	50	153	50	0	220	117
b23h	50	403	50	0	131	157
b23	50	253	50	0	218	142

TABLE 2. Gravity simulation, * BOUNDARY method, 10 steps

Elem Type	# of Elem.	# of Variables	# of Inc.	# of Cutbacks	Total Iterations	CPU Secs.
b21h	50	253	10	0	43	16
b21	50	153	46	14	364	62
b23h	50	403	10	0	53	32
b23	50	253	25	8	213	59

TABLE 3. Gravity simulation, user MPC method, 1 step, auto incrementation

Elem Type	# of Elem.	# of Variables	# of Inc.	# of Cutbacks	Total Iterations	CPU Secs.
b21h	50	265	11	1	61	35
b21	50	165	158	31	835	294
b23h	50	415	11	0	55	62
b23	50	265	31	3	226	93

TABLE 4. Gravity simulation with 50 B21H beam elements: different nip methods

Nip Method	# of Var.	# of Steps	# of Inc.	# of Cuts	Total Iter.	CPU Secs.	Output Mbytes
One rigid surface	356	2	168	21	1158	795	1
Two rigid surfaces	459	2	172	23	1199	1186	3
*BOUNDARY method	253	10	10	0	43	16	3
*BOUNDARY method	253	50	50	0	135	117	68
User MPC method, auto inc.	265	1	11	1	61	35	1
User MPC method, manual inc.	265	1	50	0	139	85	1

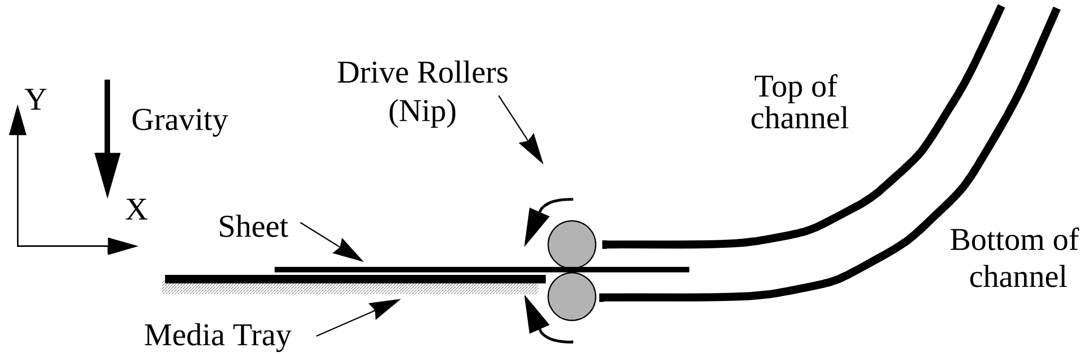


FIGURE 1. Schematic of sheet being transported into a channel.

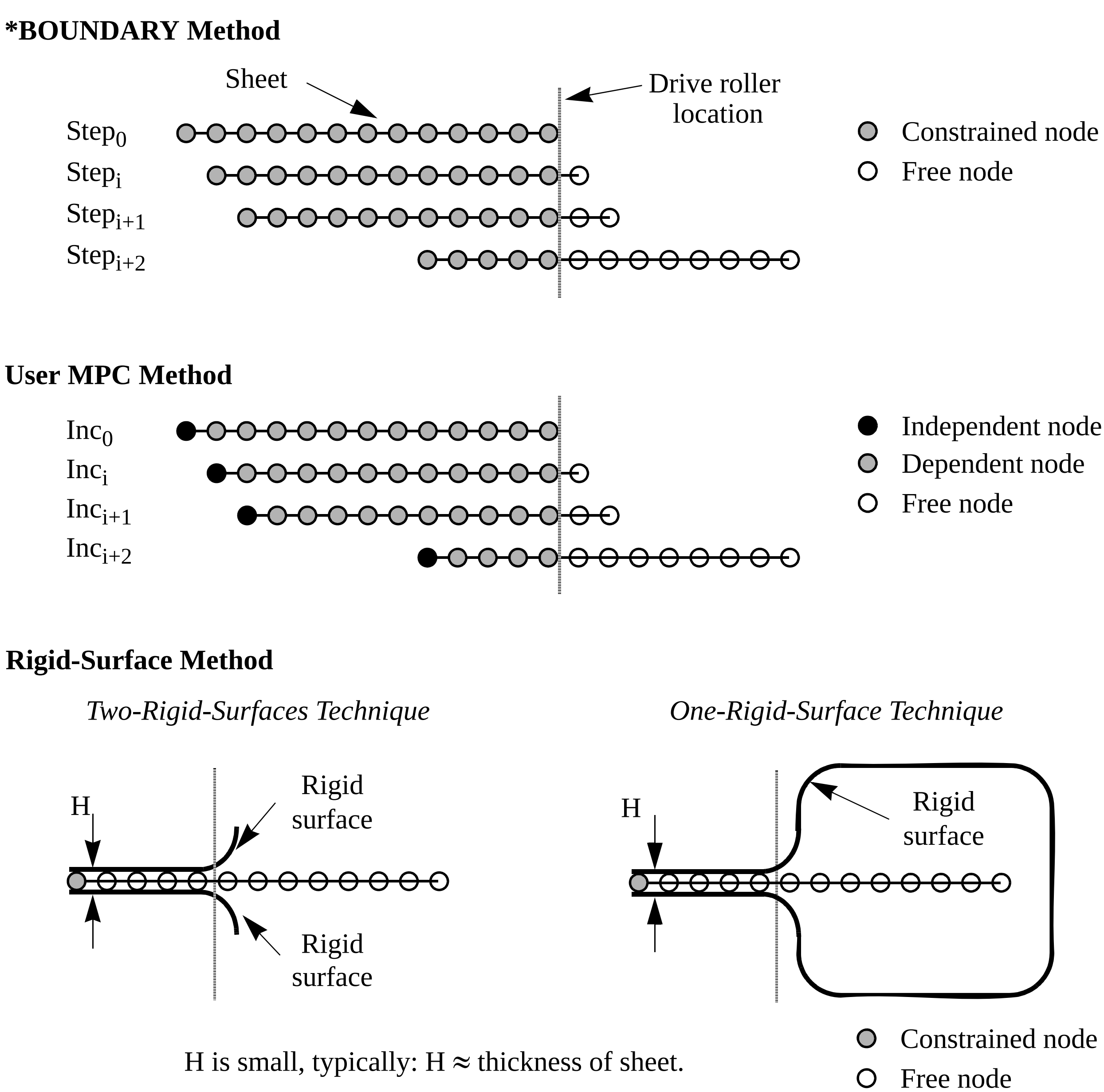
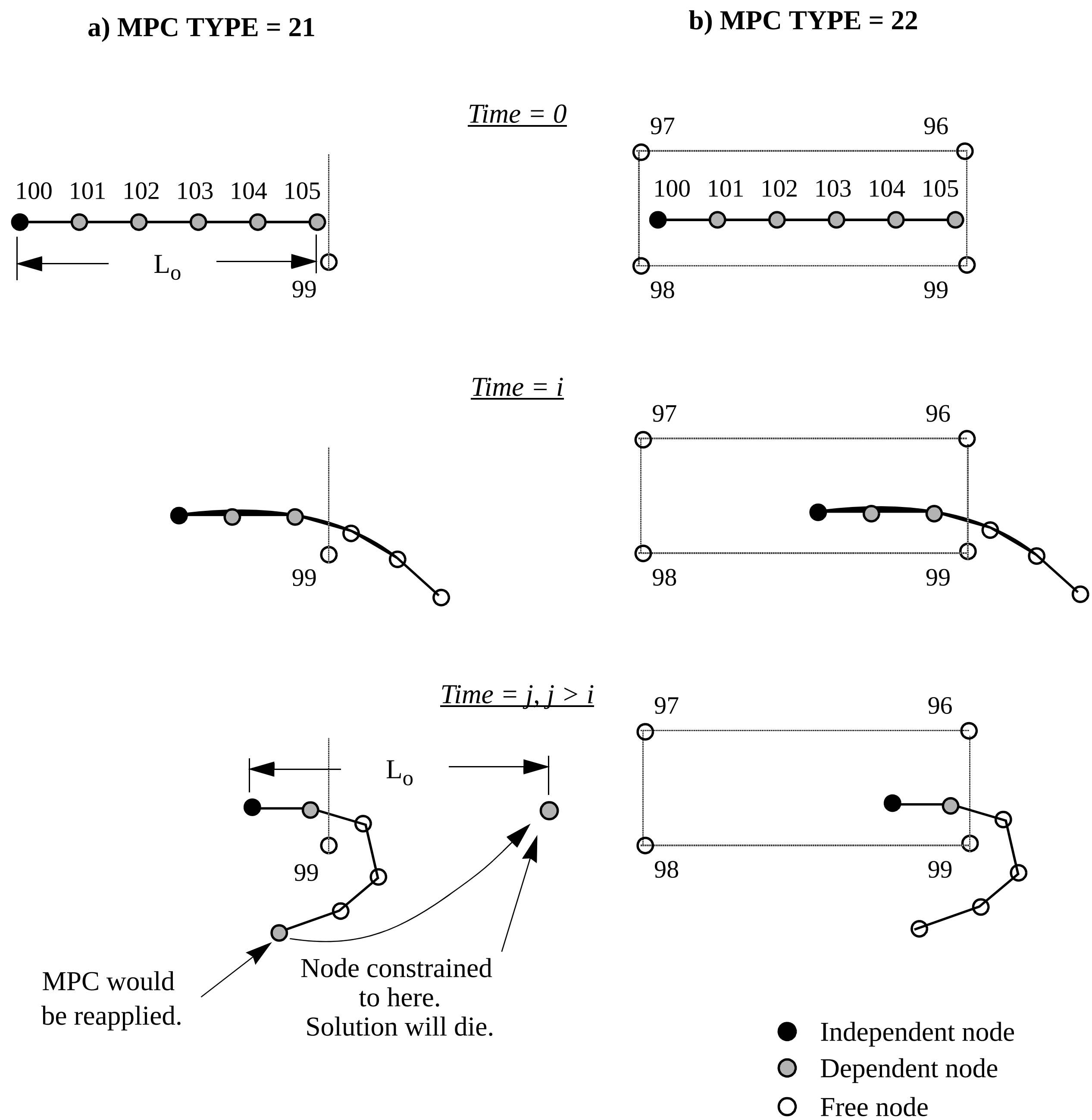


FIGURE 2. Three methods of modelling the changing boundary conditions produced by the drive nips.



Nodes 96, 97, 98, 99 are dummy nodes to define where MPC is on/off

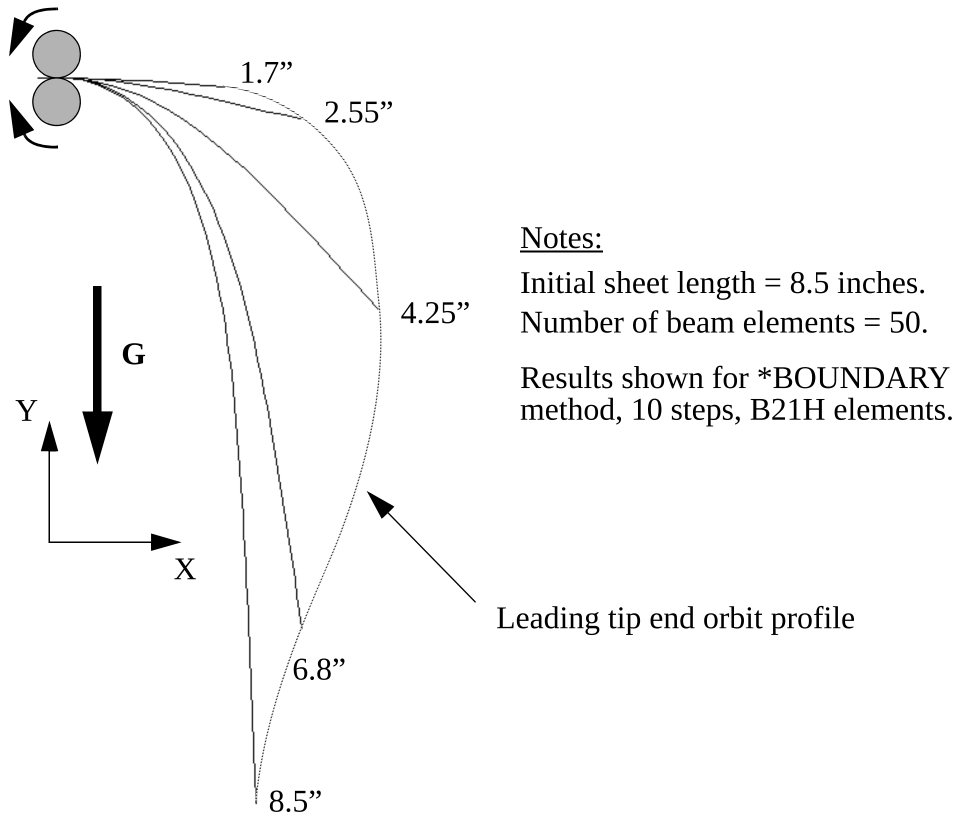
Input Deck Cards

```
*NSET, NSET=DEPEND, GENERATE
101, 105, 1
*MPC, USER, MODE=NODE
21, DEPEND, 100, 99
```

```
*NSET, NSET=DEPEND, GENERATE
101, 105, 1
*MPC, USER, MODE=NODE
22, DEPEND, 100, 96, 97, 98, 99
```

FIGURE 3. Schematic of two versions of user MPC subroutine: (a) simple check of X coordinates of nodes and (b) check of whether nodes are inside or outside of the quadrilateral defining the drive mechanism space.

Deformed Shapes of Media predicted by ABAQUS



End Orbit Profile

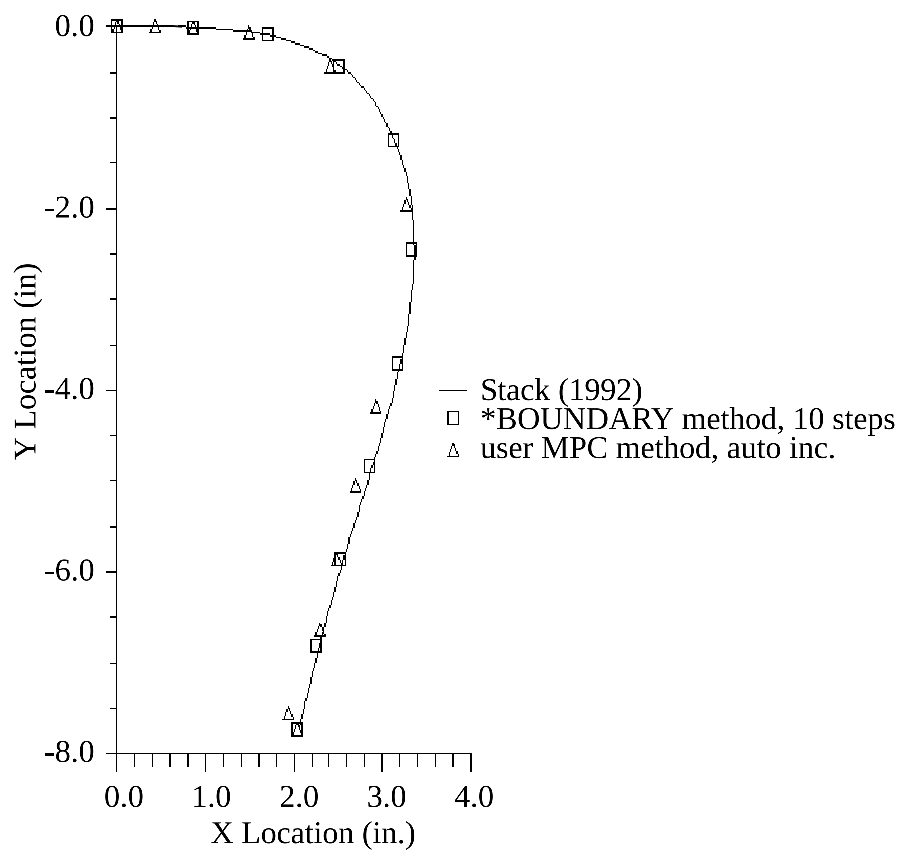


FIGURE 4. Quasi-static feeding of a thin sheet into a gravity field.

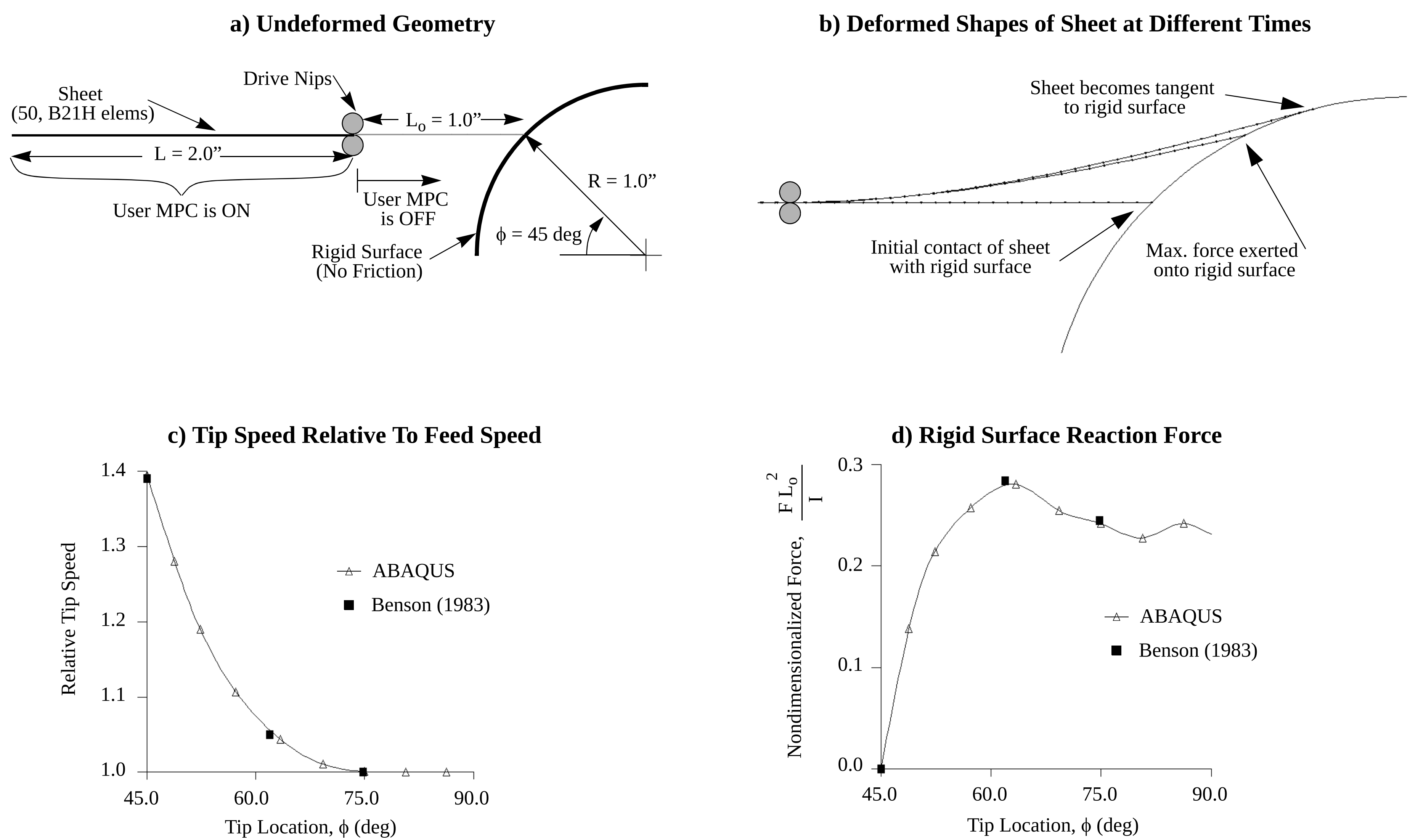


FIGURE 5. Quasi-static feeding of a sheet onto a circular frictionless drum. Gravity loading is not included.

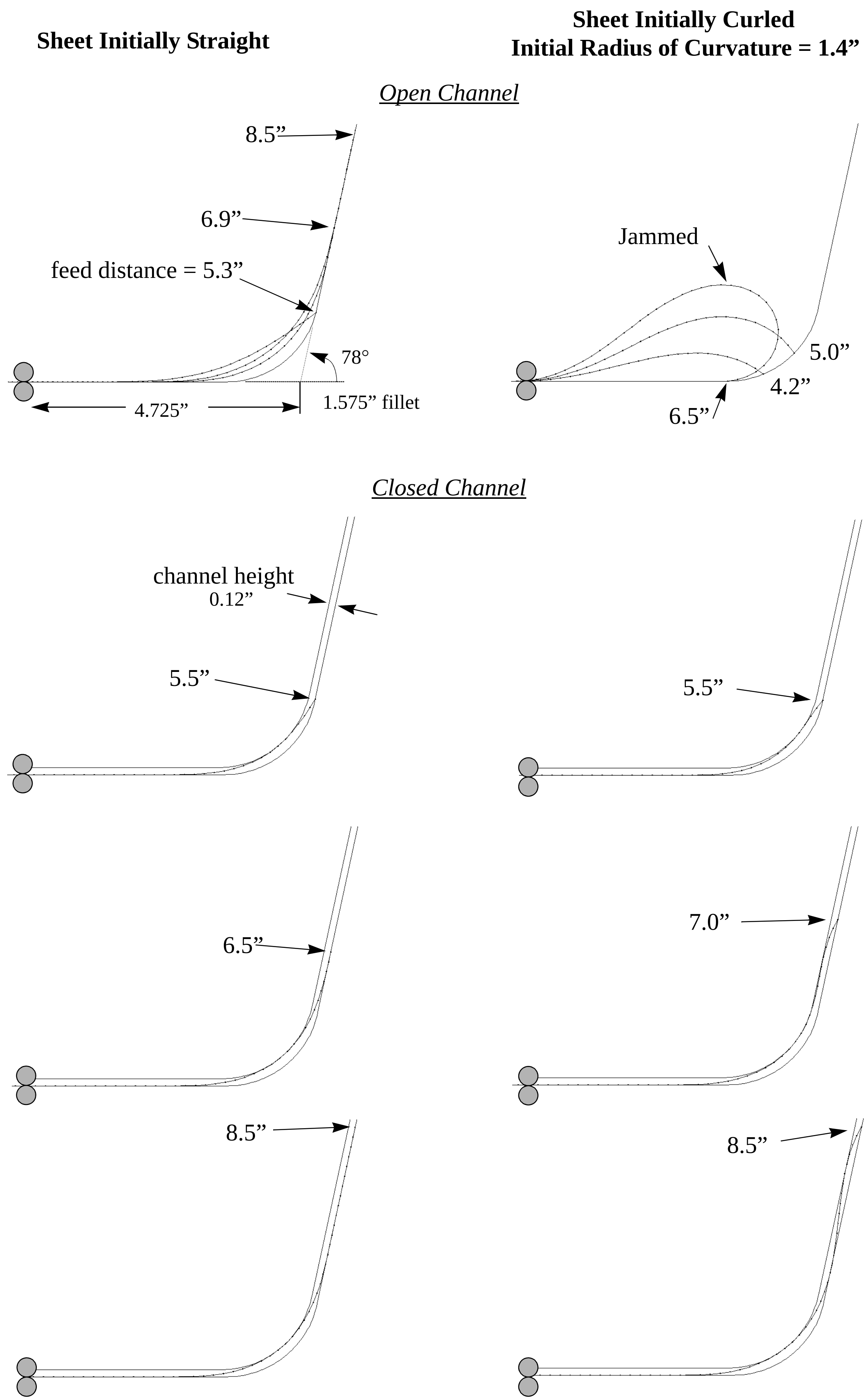


FIGURE 6. Feeding straight and curled media into both open and closed channels: Deflected shapes for different feed distances. Gravity loading is included.

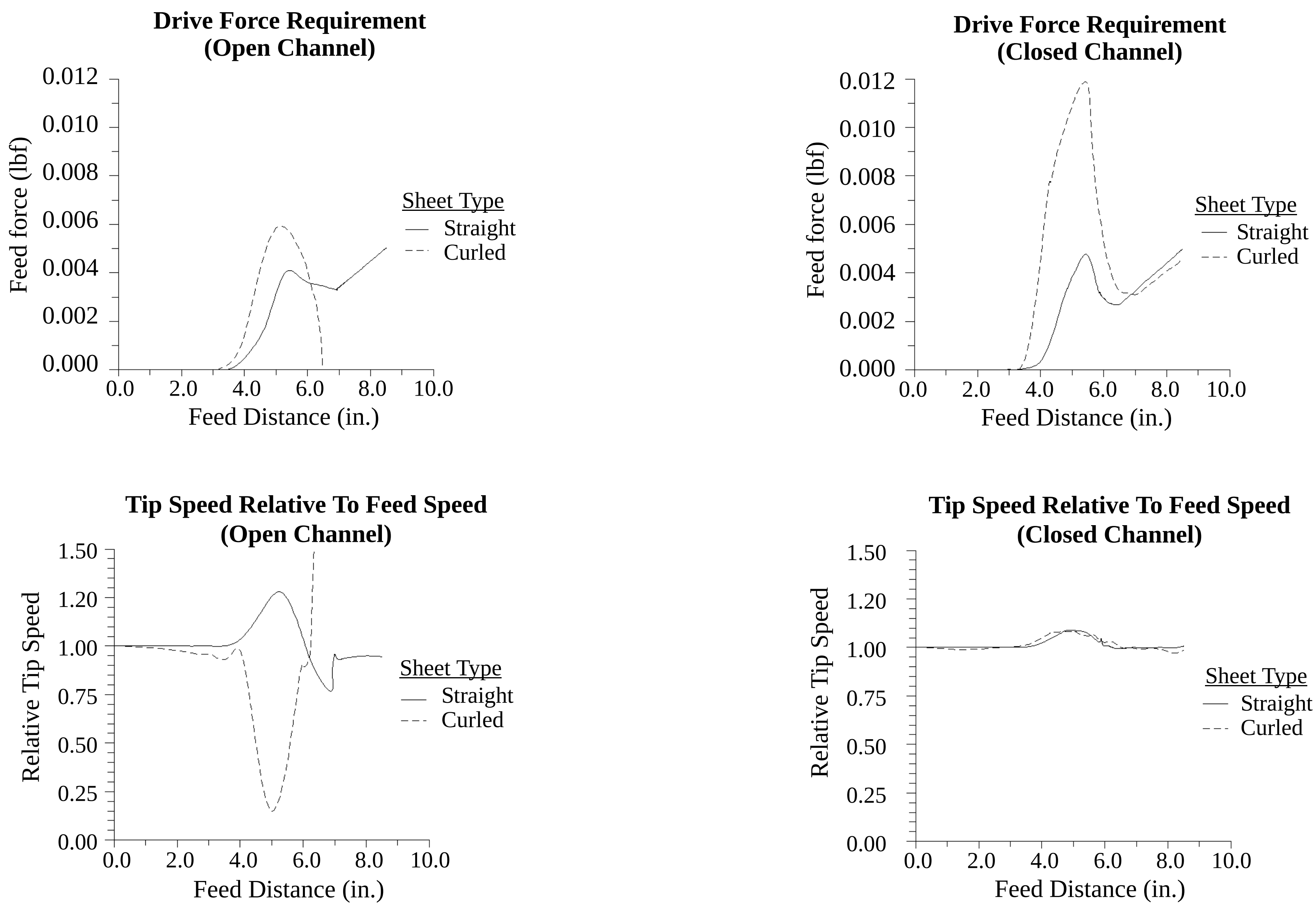


FIGURE 7. Feeding straight and curled media into both open and closed channels: drive requirements and relative tip speeds. Gravity loading is included.

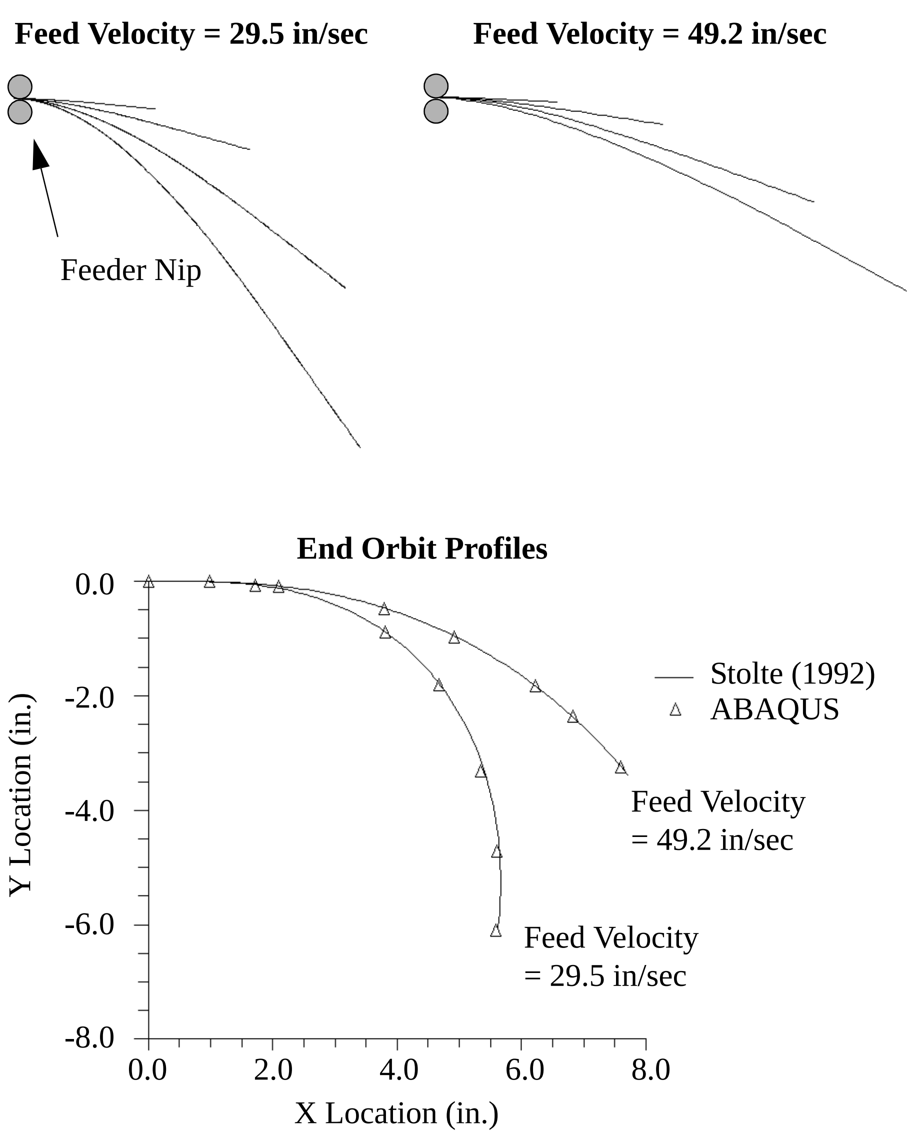


FIGURE 8. Dynamic extension of a sheet into a gravity field at different feed velocities.

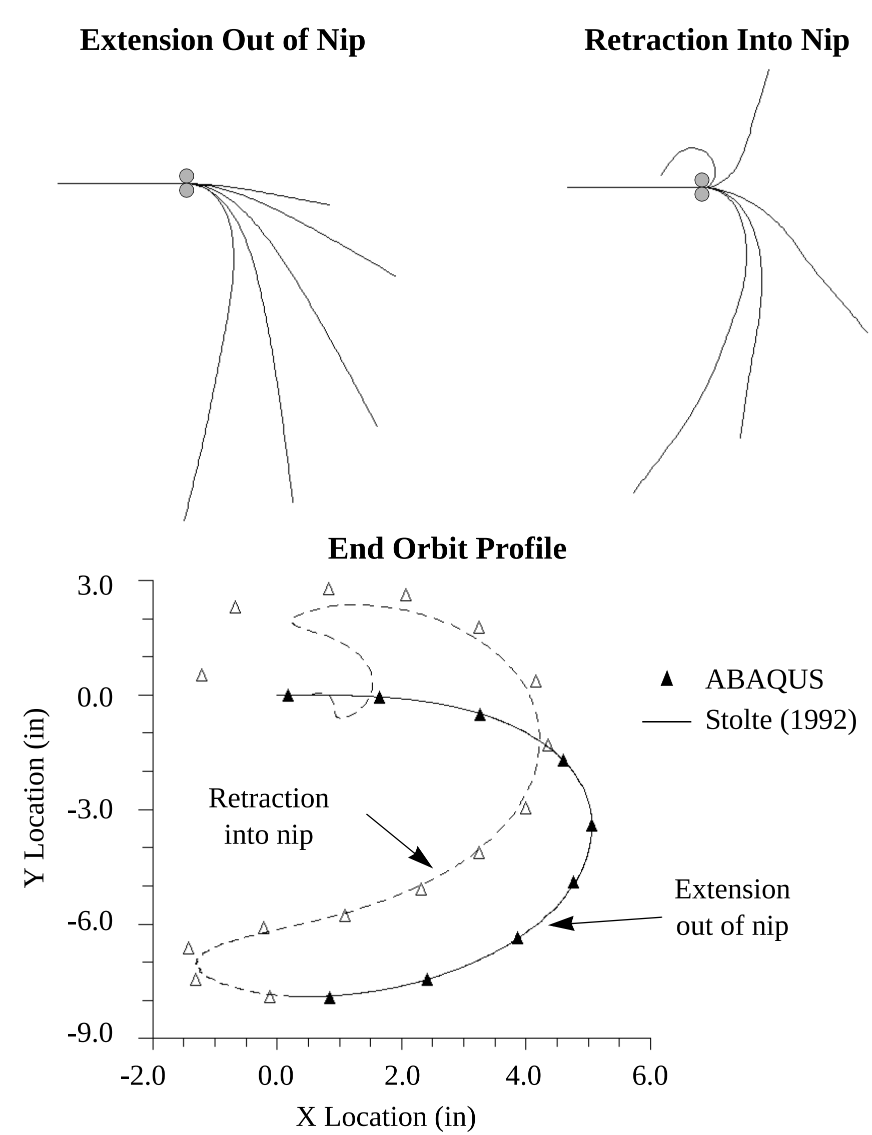


FIGURE 9. Dynamic extension and retraction of a sheet in a gravity field at a constant feed acceleration of -91.13 in/sec^2. Initial Velocity = 39.37 in/sec.