Implementation of a Leaf Spring Model in ADAMS/Rail

Chris Verheul
Sayfield International
&
NS Materieel Engineering

Agenda

- Introduction
- Leaf Spring Model Description

- General Implementation
- XY- Plane Force Implementation
- GUI Implementation

- Conclusions

Introduction

- Application area: Cars, Trucks, Rail
- Aspects of use of Leaf Springs:
  - \- Behavior depends on environment (moistness, sand)
  - \+ Integrated suspension approach: Maintenance
    - Vertical & in-plane
    - Spring & damper
- Still heavily used in special area's:
  - High reliability
  - Cost efficient

- Problems in modeling leaf springs:
  - Stiffness, Damping (friction or hysteresis) depend on:

    amplitude & frequency of force & deflection
  - Geometry variation during jounce/rebound
- Methods for modeling (see tire modeling):
  - Physical (FEM):
    - -- Large amount of parts and equ’s --> long simulations
    - + Much physical insight
  - Empirical:
    - - No geometrical non-linearity: i.e. roll center
    - + Easily fitted to purpose and complexity
    - + Method is based on measurements results
  - *Mathematical* :
    - TH Darmstadt: goniometric kinematics

- Modeling method applied:

  Hybrid: UMTRI + Math (geometry)

◆ UMTRI: Univ. of Michigan Transport Research Institute

- SAE 800905 1980, still best documented emirical model
- Proof: 1995 OECD DIVINE Element 4:

  Simulation of Heavy vehicle dynamic wheel loads:

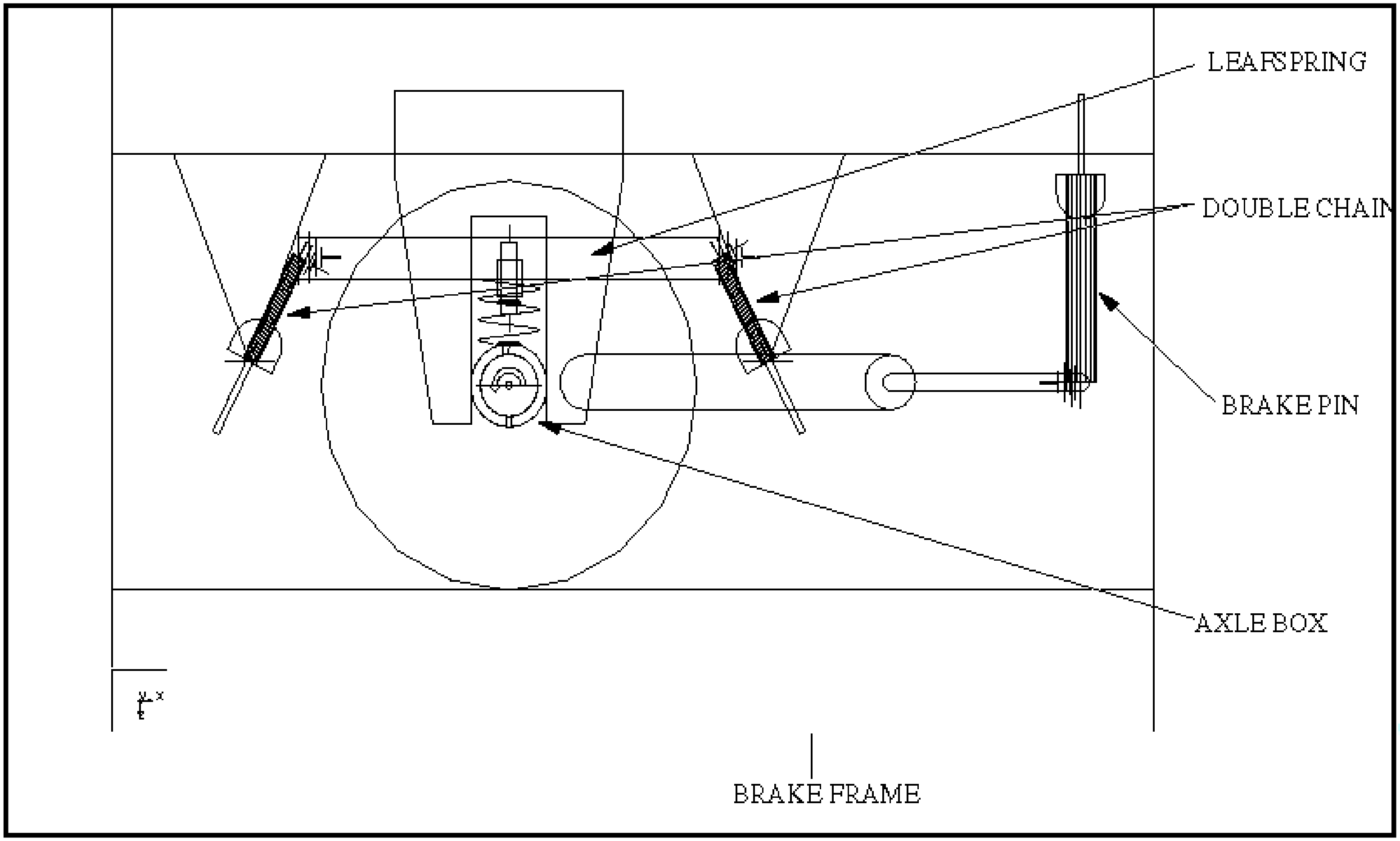
  - UMTRI spring +/- 95 % accurate
  - Better than complex models
  - Winner ran real time models !!

◆ Combined with geometry of **rigid** leafspring

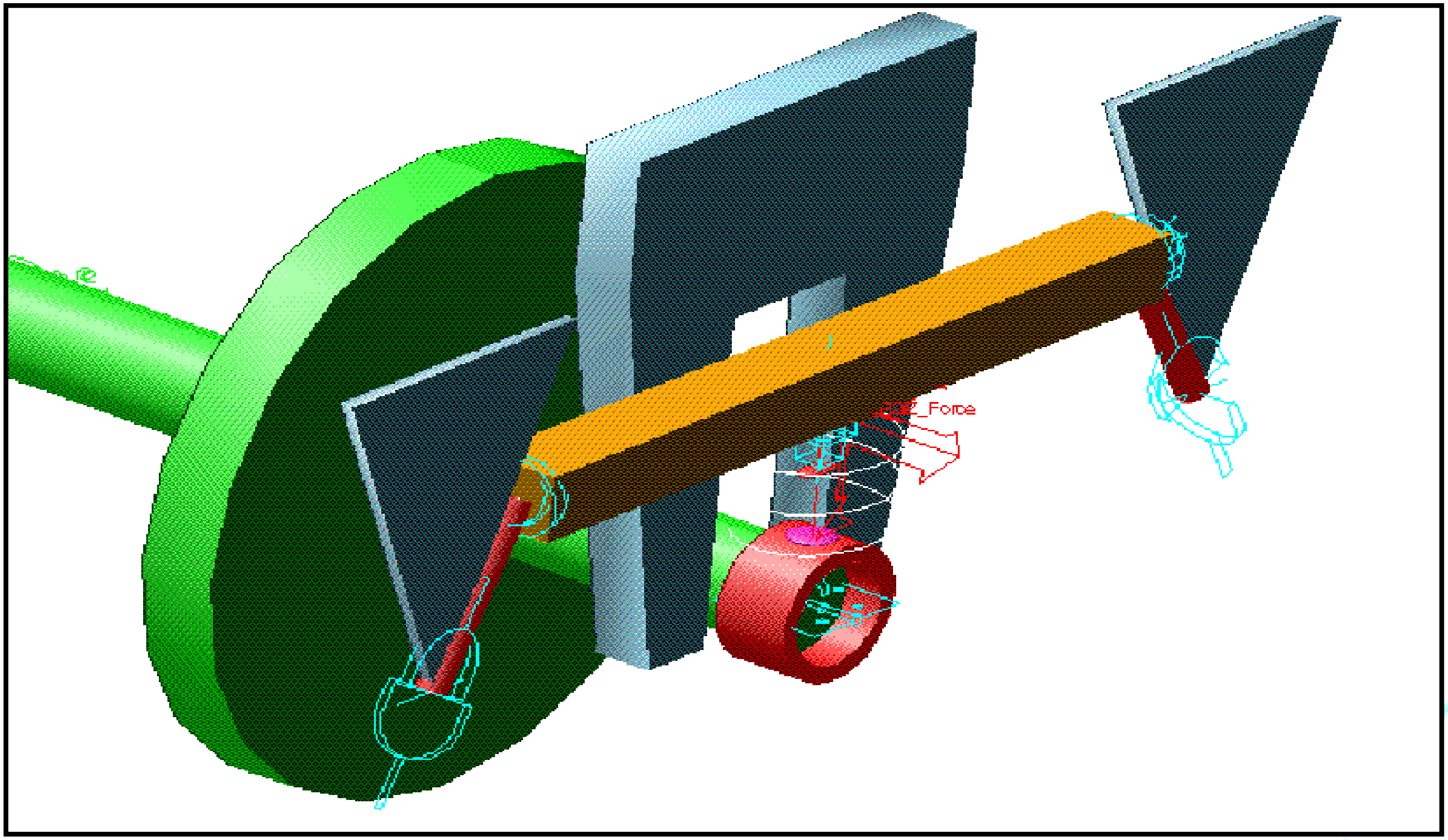
Leaf Spring Model Description

- Rigid Parts: 2 Chains, 1 Spring
- Joint connection topology:
  - *Z* *Spring* Tr *Axlebox* Rv *Axle*
  - *XY* *Chassis* Hk *Chain* Sp *Spring* Sp *Chain* Hk *Chassis*
  - Total 2-axled car: 19 Parts & 18 DOF
- Force connection topology:
  - Z Spring bending *Spring* Gfo *Axlebox*
  - XY Chains & Box guides *Chassis* Gfo *Axlebox*

Leaf Spring Topology Layout

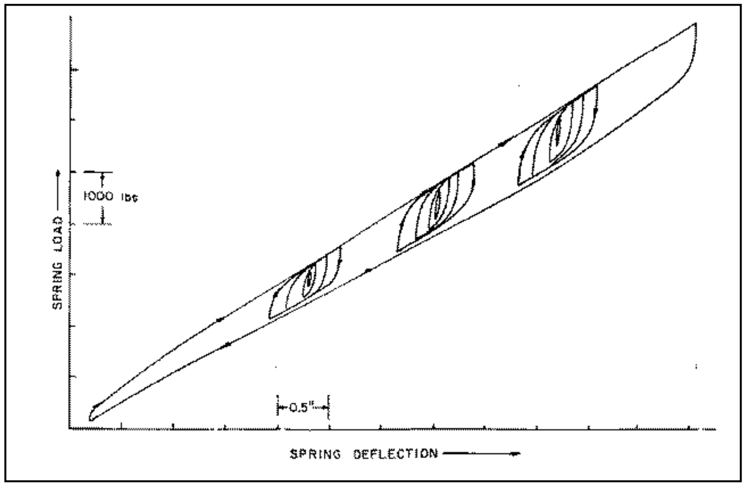


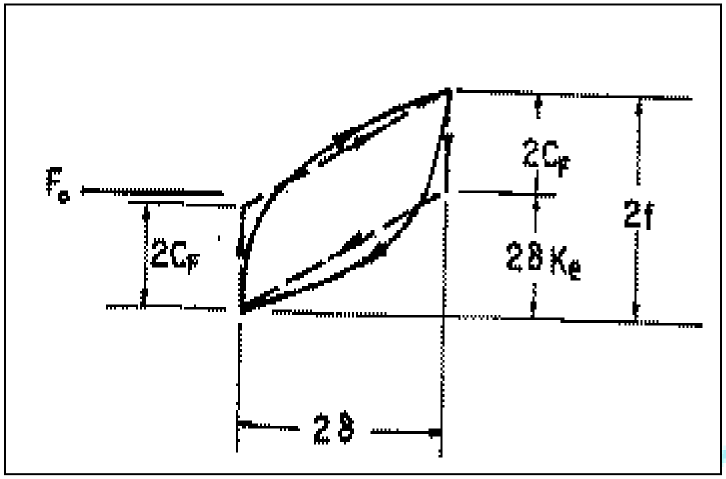
Leaf Spring Topology Layout



- Goal: Describe energy dissipated in one cycle using:
  - K_e Effective spring rate
  - C_f Average coulomb damping

- Obtain data a-s-a-p from Measurements



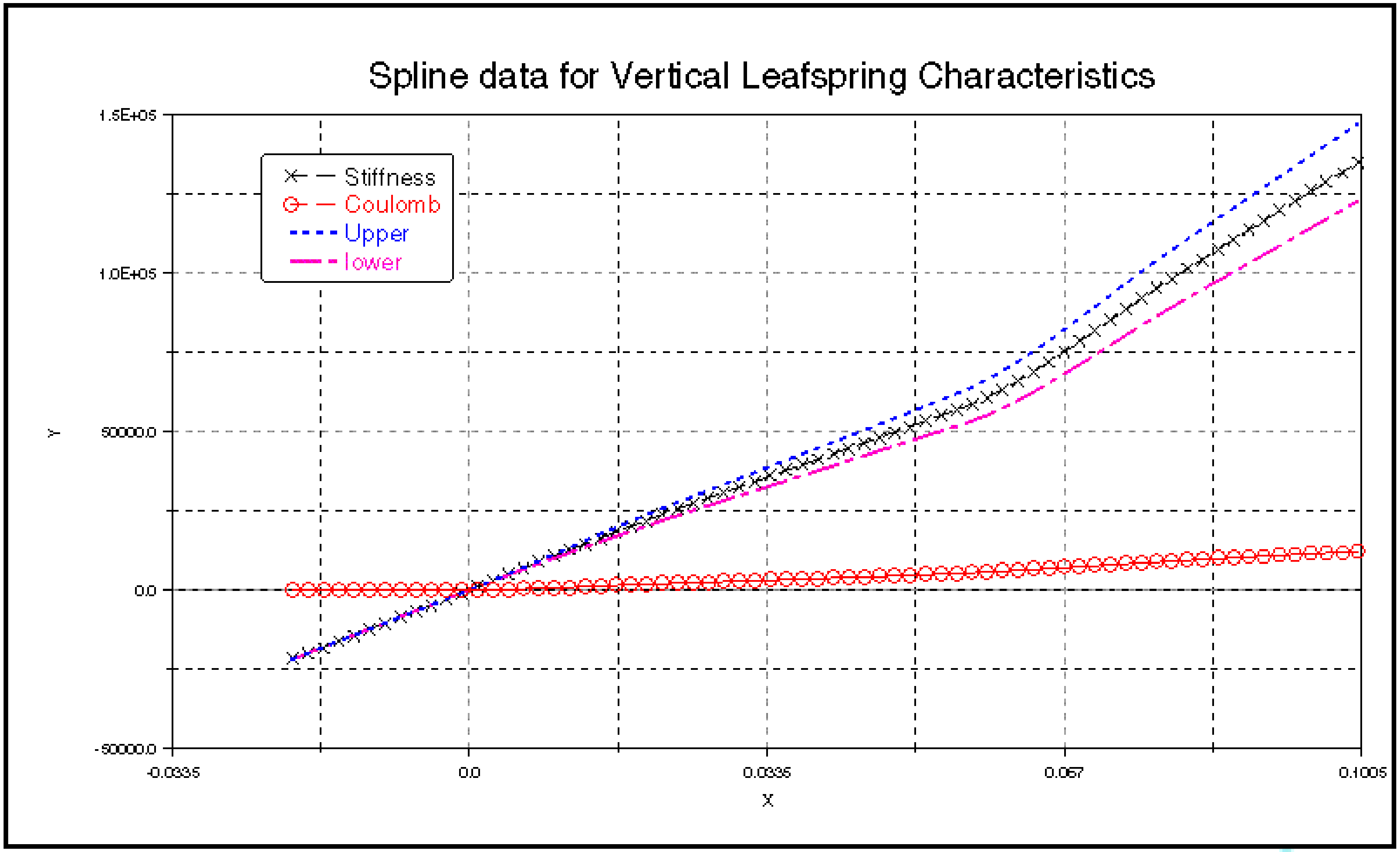


- Method must describe:
  - Decrease of K_e at increasing amplitude δ
  - Increase of Cf at increasing amplitude δ
- Base Formula: $F = F_{env} + F_{transient}$

$$= F_{env} + (F_{i-1} - F_{env})\, e^{-|\delta - \delta_{i-1}|/\beta}$$

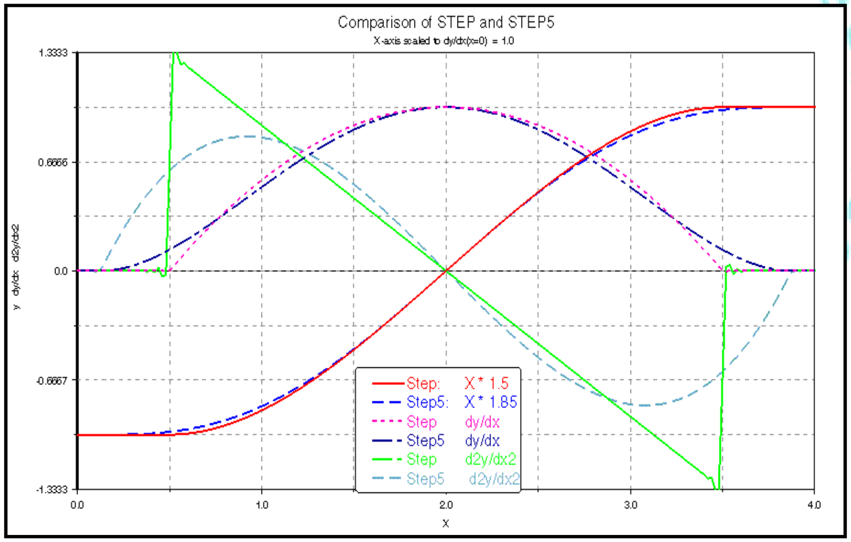
F, F_{i-1}	current and previous force
F_{env}	force envelope: $F_K + F_c$ for $\delta>0$ and $F_k - F_c$ for $\delta<0$
δ, δ_{i-1}	current and previous deflection
β	envelope approach factor: β_u for $\delta>0$ and β_l for $\delta<0$

Spline data for force envelopes

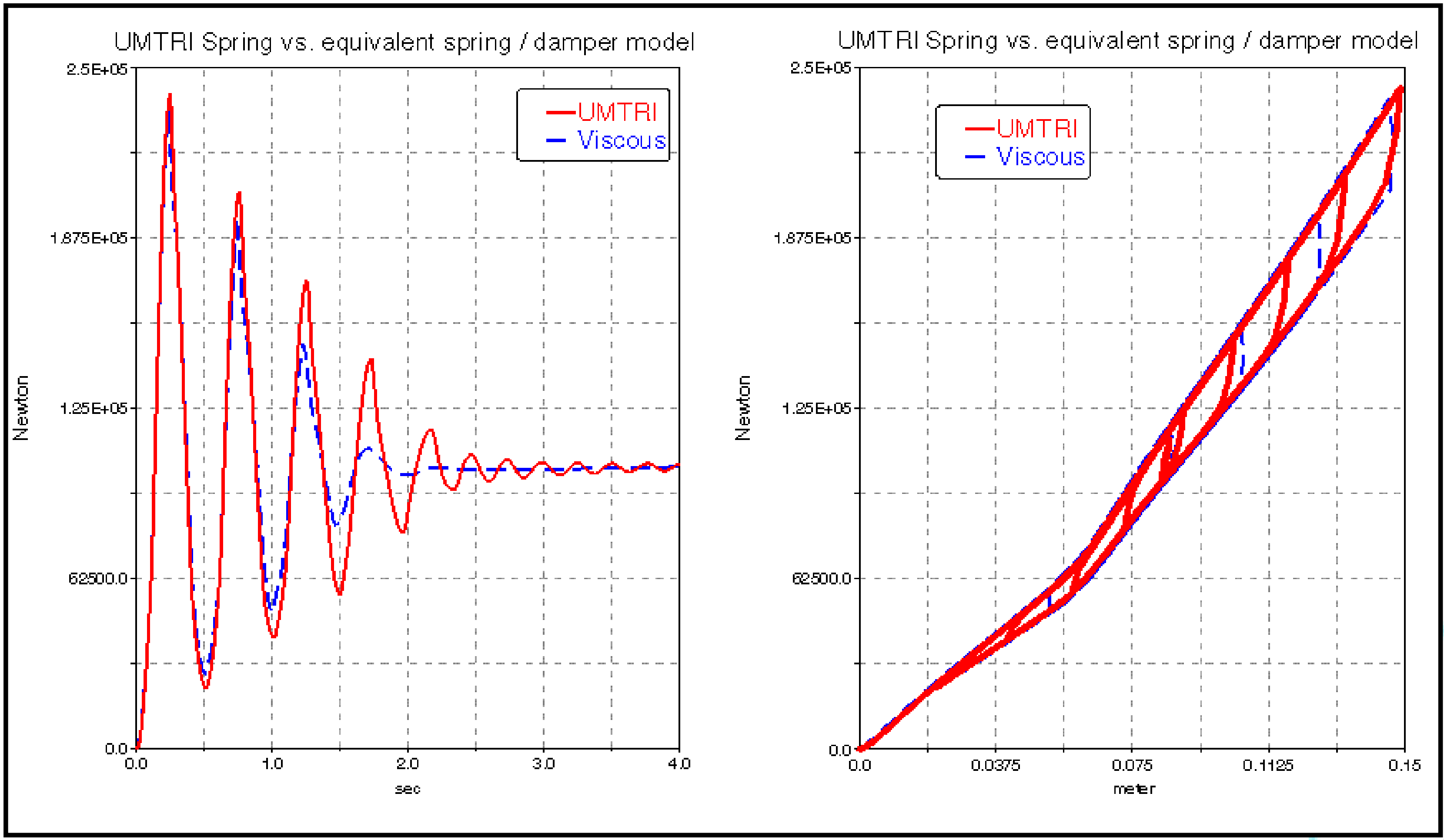


General Implementation

- Problem: What is X $_{i-1}$ in ADAMS ?
  - Variable step, Stiff integration etc.
  - --> Approximate F_{i-1} and δ_{l-1} using diff. Eq's
- Definition of F_k , F_c :
  - Splines as function of δ
  - Step5 transfer in-out
- Definition of β_u , β_l :
  - 1st order polynomials
  - Step5 transfer in-out

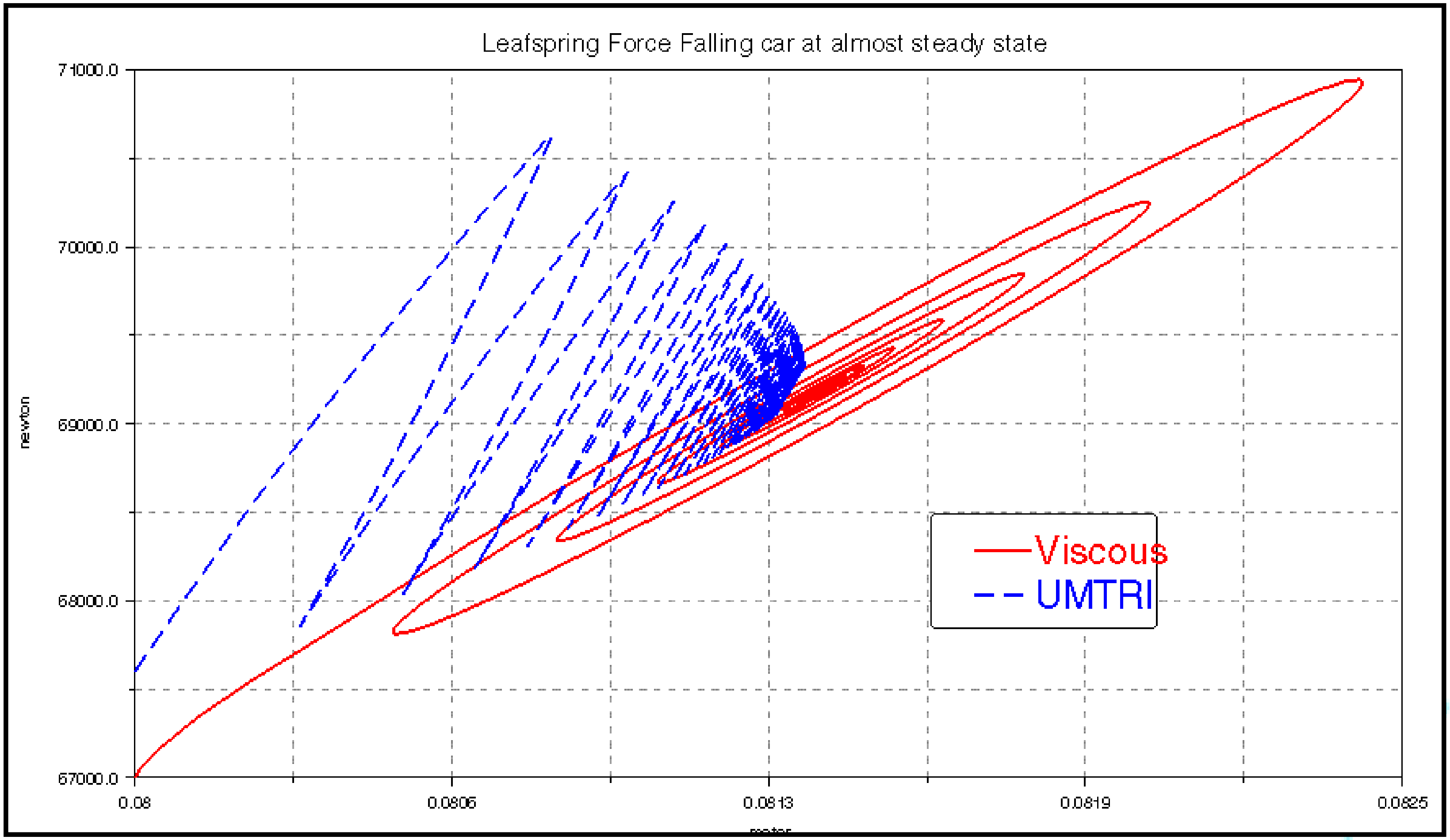


UMTRI Model vs. Viscous



- Implementation aspect:
  - Coulomb force must be zero in initialization phase
- Full model ↔ Viscous_model:
  - $F = F_{env} + F_{tr} = F_{Stiffness}\ F_{damping}$
  - Switch to viscous model with same parameters
  - Purpose:
    - Model speedup & concept testing
    - *Predictable* linear analysis

Bonus: *sticky* behavior at small amp.



XY-Plane Force Implementation

- Model is able to describe in-Plane data for:
  - Stiffness at increasing $\delta_{x,y}$:

→ Gravity effect of chains: pendulum geometry

→ $K_{x,y}$ higher (double) at contact chain part to chassis:

⇨ spline data η_1 ($\delta_{\mathbf{x,y}}$) $F_z = F_{zlo}$ and η_2 ($\delta_{\mathbf{x,y}}$) $F_z = F_{zhi}$

→ Full stop at contact axle box to chassis:

⇨ spline data η_1 ($\delta_{\mathbf{x,y}}$) $F_z = F_{zlo}$ and η_2 ($\delta_{\mathbf{x,y}}$) $F_z = F_{zhi}$

- Damping at increasing $\delta_{x,y}$:

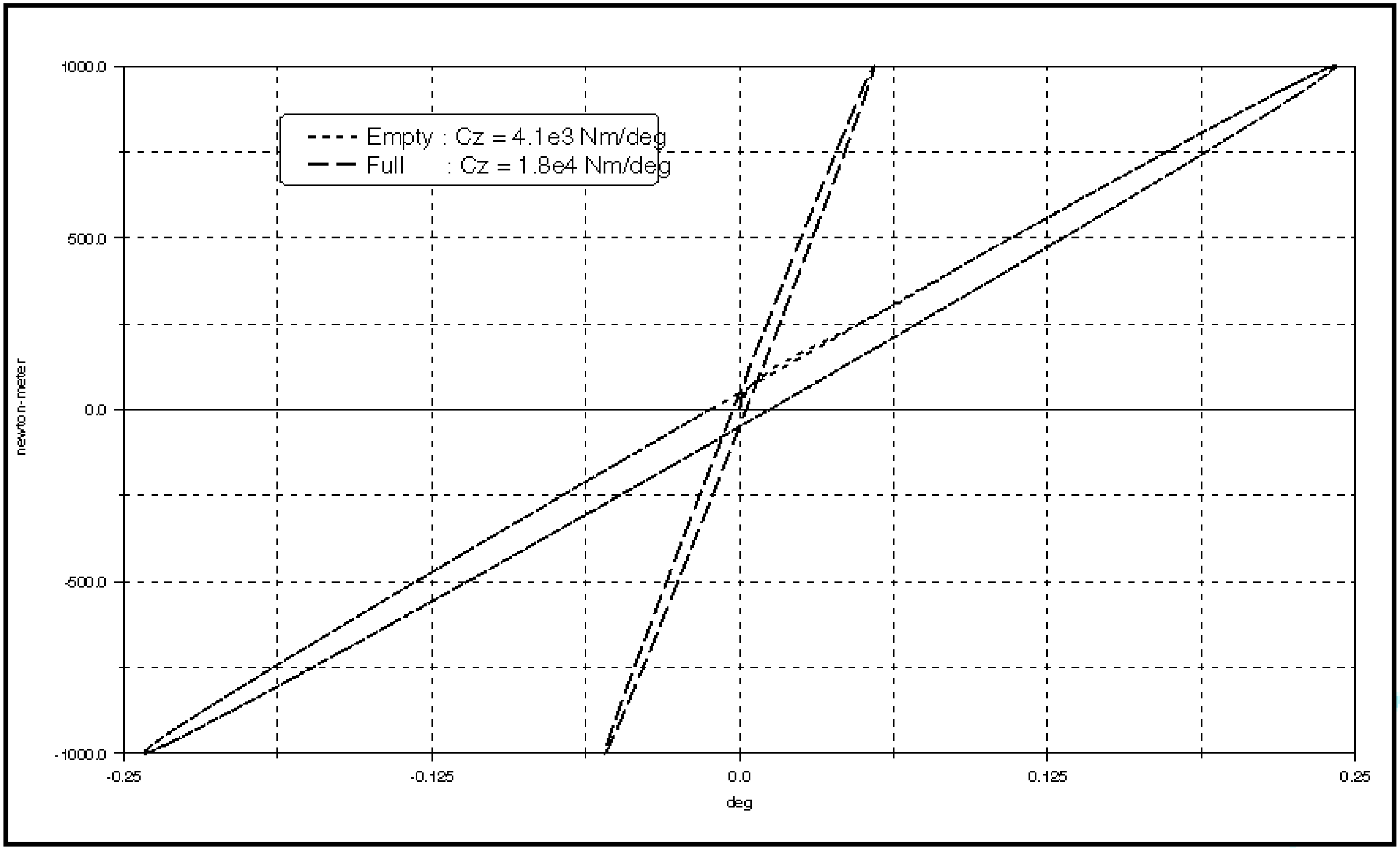
→ Friction between chain parts:

⇨ spline data φ_1 ($\delta_{\mathbf{x,y}}$) $F_z = F_{zlo}$ and φ_2 ($\delta_{\mathbf{x,y}}$) $F_z = F_{zhi}$

→ Friction at contact box to chassis:

⇨ spline data φ_1 ($\delta_{\mathbf{x,y}}$) $F_z = F_{zlo}$ and φ_2 ($\delta_{\mathbf{x,y}}$) $F_z = F_{zhi}$

Using the (linear) model to fit data



- Parameter Fz_Model: user switch for:
  - Low_Only: Use data for low F_z load splines
  - High_Only: Use data for high Fz load splines
  - Full_Load: Interpolate between splines for F_{zlo} and F_{zhi}
- Loss of T_z must be compensated for
  - 1 Force at center replaces 2 forces at spring ends
  - Extra Torque due to shift of lateral forces: $Tz = \Lambda\ (\gamma_z)$

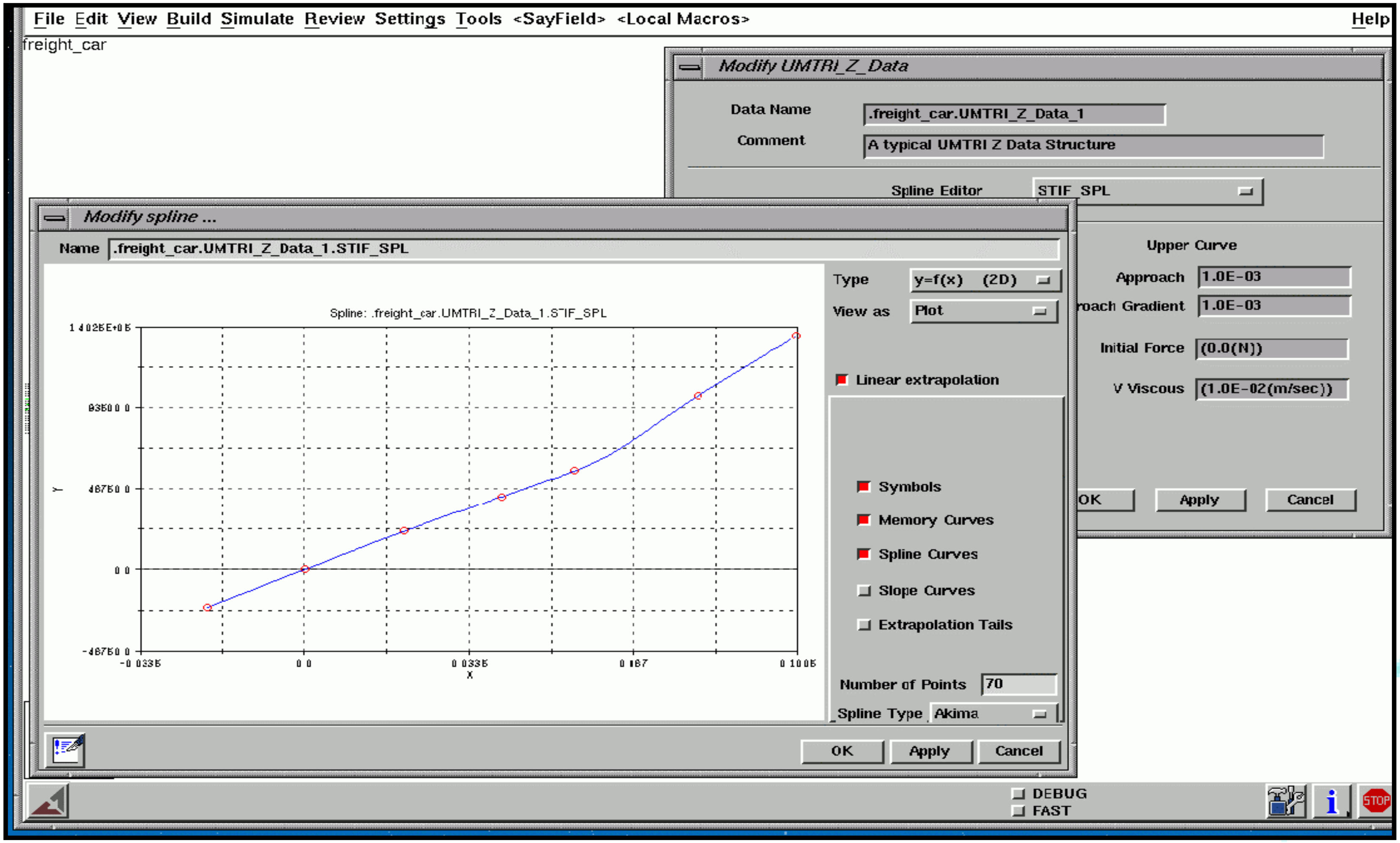
where $\gamma_z = (\delta_{\mathbf{y,front}} - \delta_{\mathbf{y,rear}}) / \mathbf{L}_{spr}$

$\Lambda\ (\gamma_z) = \eta_{1,2}\,(\mathbf{L}_{spr}\,\gamma_z) + \Pi_{1,2}\,(\mathbf{L}_{spr}\,\gamma_z)$

GUI Implementation

- UDE’s: seamless expansion of functionality
- User defined Data types:
  - Z_Data Vertical spring data definition
  - Z_Force Vertical spring equations
    - In Z_Data, IJR_Mar, Type
    - Out $F_z = \Phi (Eps_z , Vel_z)$
  - XY_Data In-plane spring data definition
  - XY_Force In-plane spring equations
    - In Z_Force, XY_Data, IJR_Mar, Type, Fz_Model
    - Out $F_{x,y} = \Phi (F_z , Eps_{x,y} , Vel_{x,y})$

ADAMS Native GUI: Defining Z-Data



Modify UMTRI_Z_Data

Data Name: .freight_car.UMTRI_Z_Data_1

Comment: A typical UMTRI Z Data Structure

Spline Editor: STIF_SPL

Upper Curve

Approach: 1.0E-03

roach Gradient: 1.0E-03

Initial Force: (0.0(N))

V Viscous: (1.0E-02(m/sec))

Modify spline ...

Name: .freight_car.UMTRI_Z_Data_1.STIF_SPL

Type: y=f(x) (2D)

View as: Plot

☒ Linear extrapolation

☒ Symbols

☒ Memory Curves

☒ Spline Curves

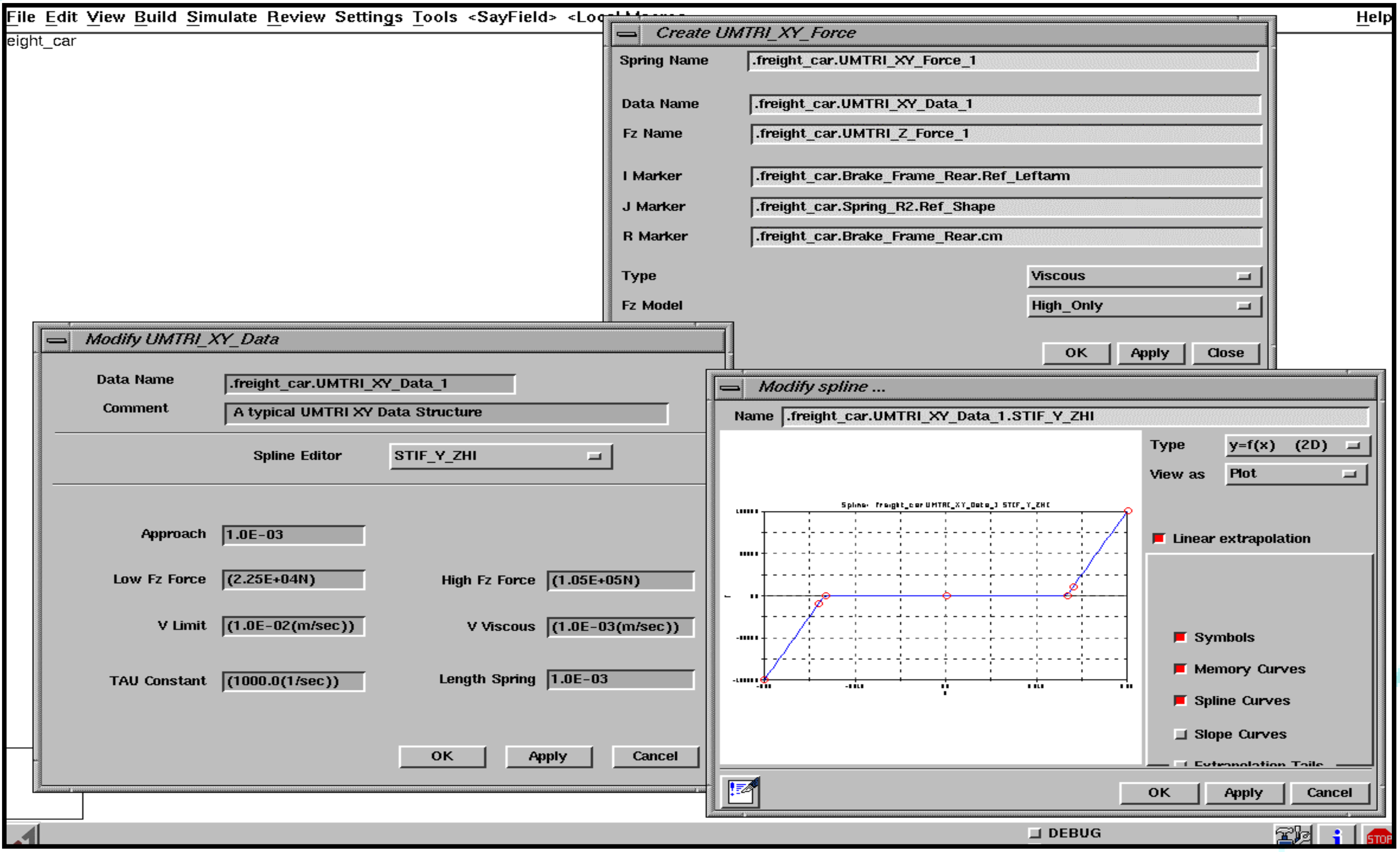
☐ Slope Curves

☐ Extrapolation Tails

Number of Points: 70

Spline Type: Akima

ADAMS Native GUI: Defining XY-Force



Create UMTRI_XY_Force

Spring Name: .freight_car.UMTRI_XY_Force_1

Data Name: .freight_car.UMTRI_XY_Data_1

Fz Name: .freight_car.UMTRI_Z_Force_1

I Marker: .freight_car.Brake_Frame_Rear.Ref_Leftarm

J Marker: .freight_car.Spring_R2.Ref_Shape

R Marker: .freight_car.Brake_Frame_Rear.cm

Type: Viscous

Fz Model: High_Only

Modify UMTRI_XY_Data

Data Name: .freight_car.UMTRI_XY_Data_1

Comment: A typical UMTRI XY Data Structure

Spline Editor: STIF_Y_ZHI

Approach: 1.0E-03

Low Fz Force: (2.25E+04N)

High Fz Force: (1.05E+05N)

V Limit: (1.0E-02(m/sec))

V Viscous: (1.0E-03(m/sec))

TAU Constant: (1000.0(1/sec))

Length Spring: 1.0E-03

Modify spline ...

Name: .freight_car.UMTRI_XY_Data_1.STIF_Y_ZHI

Type: y=f(x) (2D)

View as: Plot

☒ Linear extrapolation

☒ Symbols

☒ Memory Curves

☒ Spline Curves

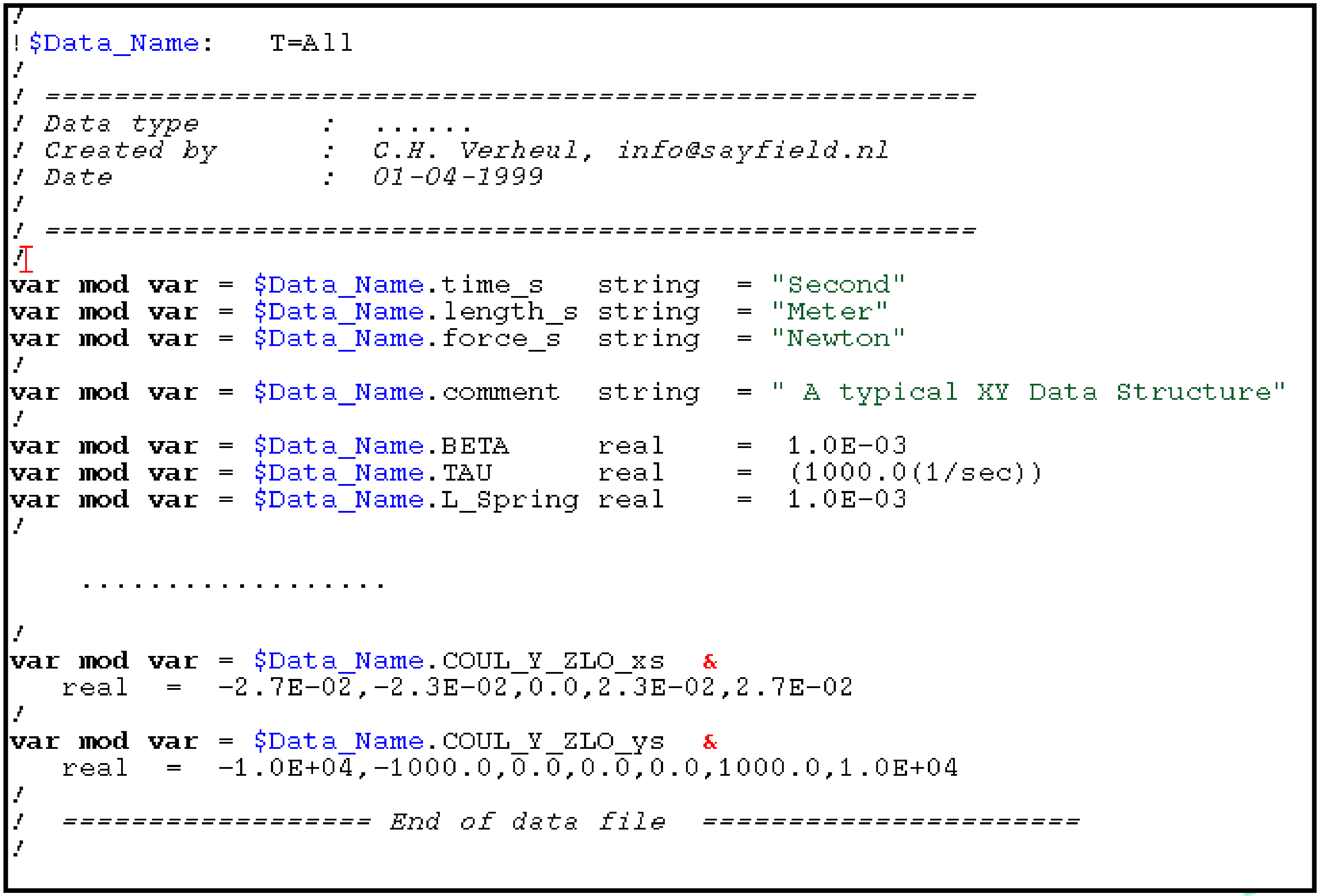
☐ Slope Curves

☐ Extrapolation Tails

- Component data storage Method:
  - N components - 1 data structure
  - Special *Component Data* UDE's
  - Data Libraries: Global - Local
  - Automated Unit Conversion
  - Data export/import uses standard ADAMS functionality
  - Spline Data stored as Design Variables

- UDE method will be used in development of A/Rail

Format of Assembly Data Files



```
!
! $Data_Name:    T=All
!
! =====================================================
! Data type         :  ......
! Created by        :  C.H. Verheul, info@sayfield.nl
! Date              :  01-04-1999
!
! =====================================================
!
var mod var = $Data_Name.time_s    string  = "Second"
var mod var = $Data_Name.length_s  string  = "Meter"
var mod var = $Data_Name.force_s   string  = "Newton"
!
var mod var = $Data_Name.comment   string  = " A typical XY Data Structure"
!
var mod var = $Data_Name.BETA      real    =  1.0E-03
var mod var = $Data_Name.TAU       real    =  (1000.0(1/sec))
var mod var = $Data_Name.L_Spring  real    =  1.0E-03
!

    ..................

!
var mod var = $Data_Name.COUL_Y_ZLO_xs   &
   real  =  -2.7E-02,-2.3E-02,0.0,2.3E-02,2.7E-02
!
var mod var = $Data_Name.COUL_Y_ZLO_ys   &
   real  =  -1.0E+04,-1000.0,0.0,0.0,0.0,1000.0,1.0E+04
!
!  ================== End of data file  ======================
!
```

Conclusions

- *Best Practice* model for full-vehicles (10-100 springs)
- Advised component parameter determination:
  - First Stiffness and damping in Z-direction
  - Verify in-plane stiffness due to geometry at 2 loads
  - Measure extra in-plane stiffness and damping due to stops
- Component parameters unique: Library
- Described implementation optimal for:
  - Calculation speed
  - Flexibility in use
  - Model readability
  - Automated use (A/Rail, .., A/Anything)
- UDE's extremely powerful for new functionality