Numerical Computation of Geodesics on Combined Piecewise Smooth Surfaces

Ron Caplan, Michael Davis, Jenny Dehner, Nick Hooper, Heather Mattie

Advisor: Ali Nadim   Boeing Liaison: Tom Grandine

Claremont Graduate University

Boeing Clinic 08-09

Claremont Graduate University

- 2188 graduate students

- A member of the Claremont Colleges
  - **Pomona College**
  - **Harvey Mudd College**
  - **Claremont McKenna College**
  - **Etc.**
- School of Mathematical Sciences
  - **Mathematics Clinic**

Claremont

GRADUATE UNIVERSITY

The Proposed Problem

http://en.wikipedia.org/wiki/File:Boeing_787.jpg

- Boeing 787
  - **composite tape**
  - **must be laid flat**
  - **no wrinkles, creases**

- Laying each strip along a geodesic minimizes creasing and wrinkling

- Need an algorithm that can aid in the process of finding geodesics along the combined surfaces that make up a fuselage

Overview

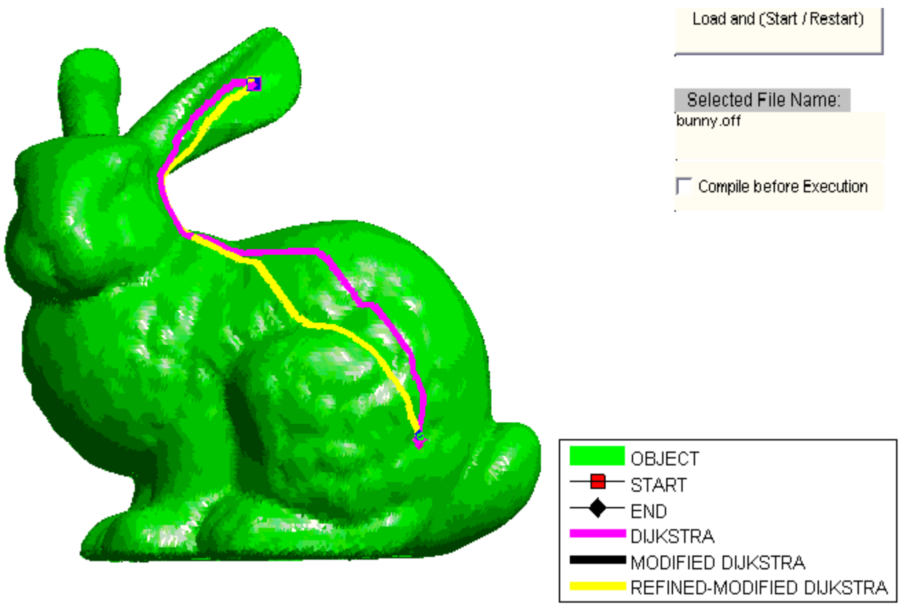
- Foundational work
- The main algorithm
- A few test runs
- Challenges and improvements
- Conclusions

Getting Acquainted with Geodesics

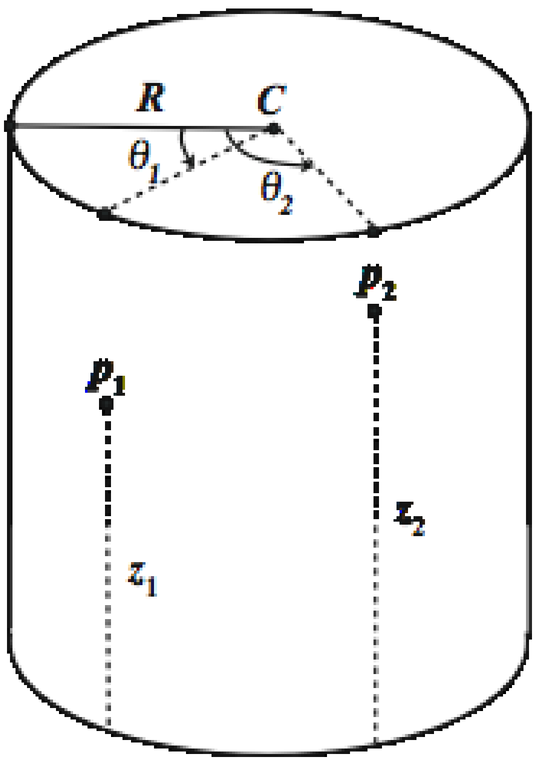
- How we got started:
  - Reviewed previous work from CGU Boeing Clinic Team 07-08
  - Performed literature review
  - Reviewed topics in differential geometry
  - “Evolution” algorithm seemed best path to follow

Previous Team’s Work

- Dijkstra’s Algorithm
  - **Limited to points and edges on the mesh**
- Fast Marching Method and Level Set Method
  - **Difficult to program and run**
- Modified Dijkstra’s Algorithm
  - **Does not take the curvature of the surface into account**
  - **Can only be used on meshed surfaces**



Toy Problem: A Cylinder



- Our first simple introduction to shortest paths and geodesics
  - **Compute shortest paths analytically using distance formulas**
  - **Gives us a reference for methods developed later**

Our New Method

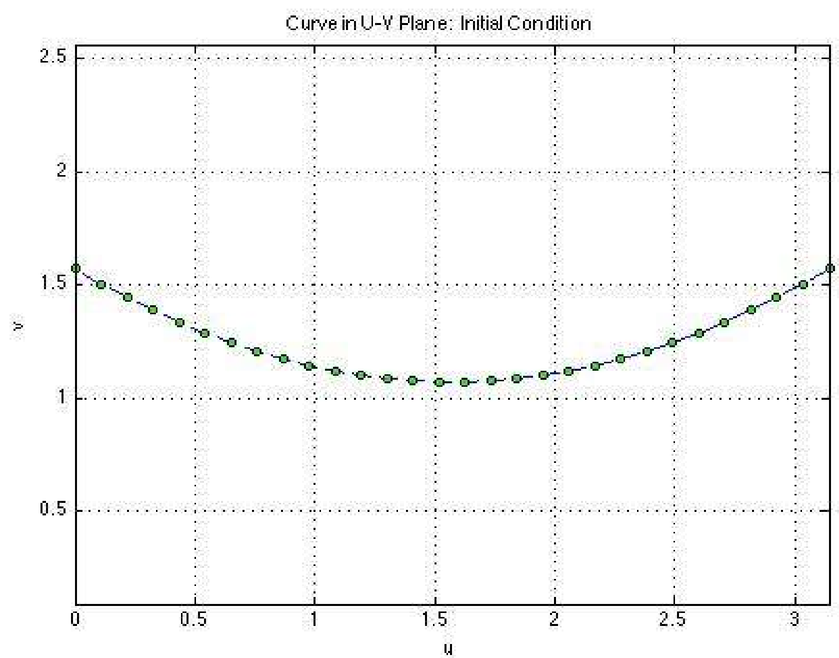
- An algorithm for evolution of an initial curve toward the geodesic
- Achieved by computing numerical properties of the curve and moving marker points along the curve in proportion to the geodesic curvature vector

The Evolution Algorithm

- Given some 3-D surface parameterized by [redacted] represented by the equation:

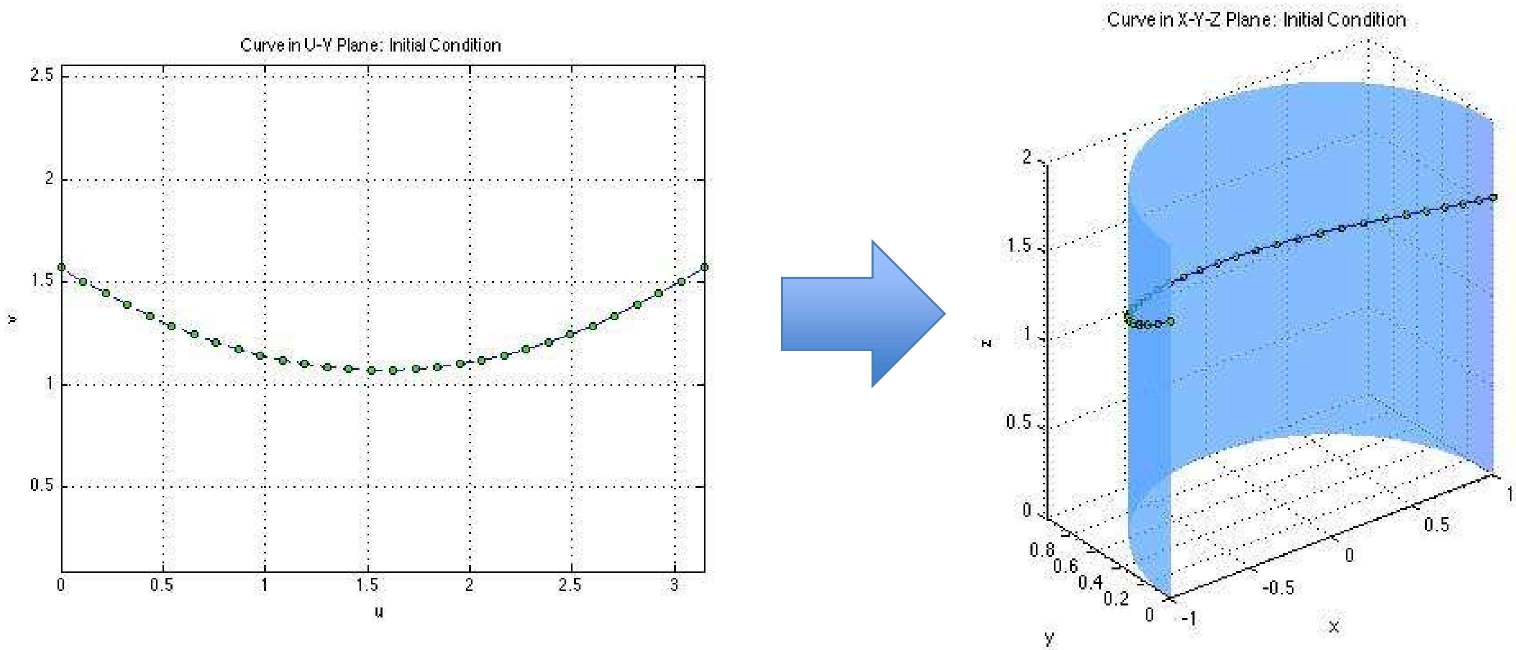
[redacted]

the initial point [redacted] and the end point [redacted], connect the two points in 2-dimensional space with [redacted] , some smooth initial curve. (Many times, a straight line between the points is the easiest.)



The Evolution Algorithm

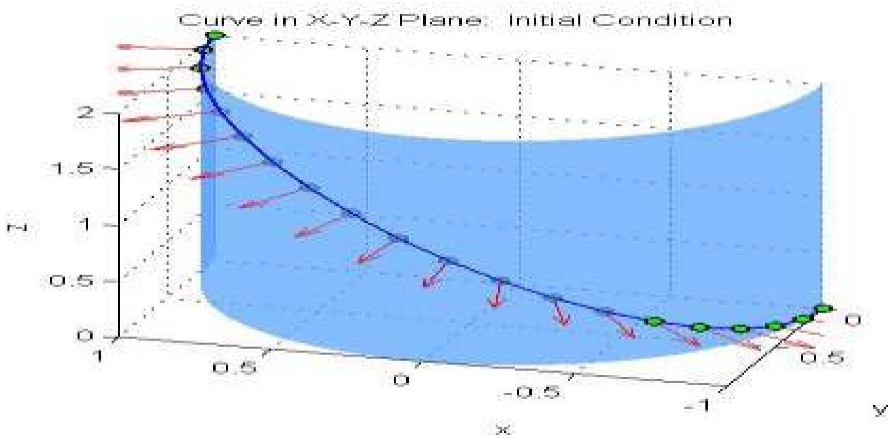
- Discretize the smooth curve γ, parameterized by arclength, into N points such that $\gamma(s_i)$ refers to the i^{th} point in the list of points and map these points onto the surface.



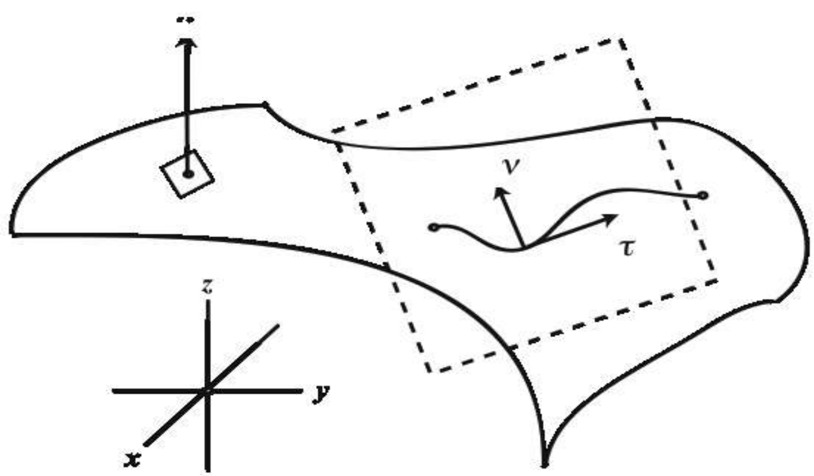
- Calculate arclength at each point in the u, v plane and on the 3-D surface.

The Evolution Algorithm

- In 3-dimensional space, calculate the normal $\hat{\mathbf{n}}$ to the surface at every point on the curve.



$$\hat{\mathbf{n}} = \frac{\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v}}{\left|\left|\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v}\right|\right|}$$



ɔ in 3-dimensional space, calculate the /ature vector:

$$\mathbf{k} = \frac{\partial \tau}{\partial s} = \frac{\partial^2 x}{\partial s^2}\hat{\mathbf{i}} + \frac{\partial^2 y}{\partial s^2}\hat{\mathbf{j}} + \frac{\partial^2 z}{\partial s^2}\hat{\mathbf{k}}$$

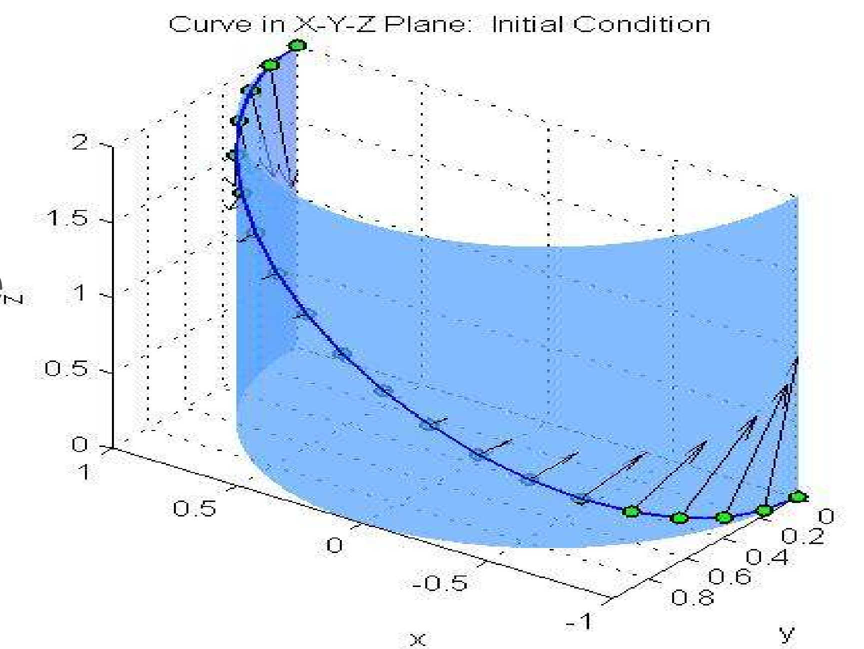
The Evolution Algorithm

- From the previous steps, calculate the *geodesic curvature* [redacted] at every point.

[redacted]

- Geodesic occurs when: [redacted]
- *Stopping Criterion*:
  - **Check sum of squares of 2-norm. If it is larger than some tolerance, take a evolution step.**

$$\left(\sum_{i=1}^{n} |\mathbf{k_g}(\mathbf{s_i})|^2\right)^{1/2} = \epsilon_{\text{tol}} \approx 0$$



The Evolution Algorithm

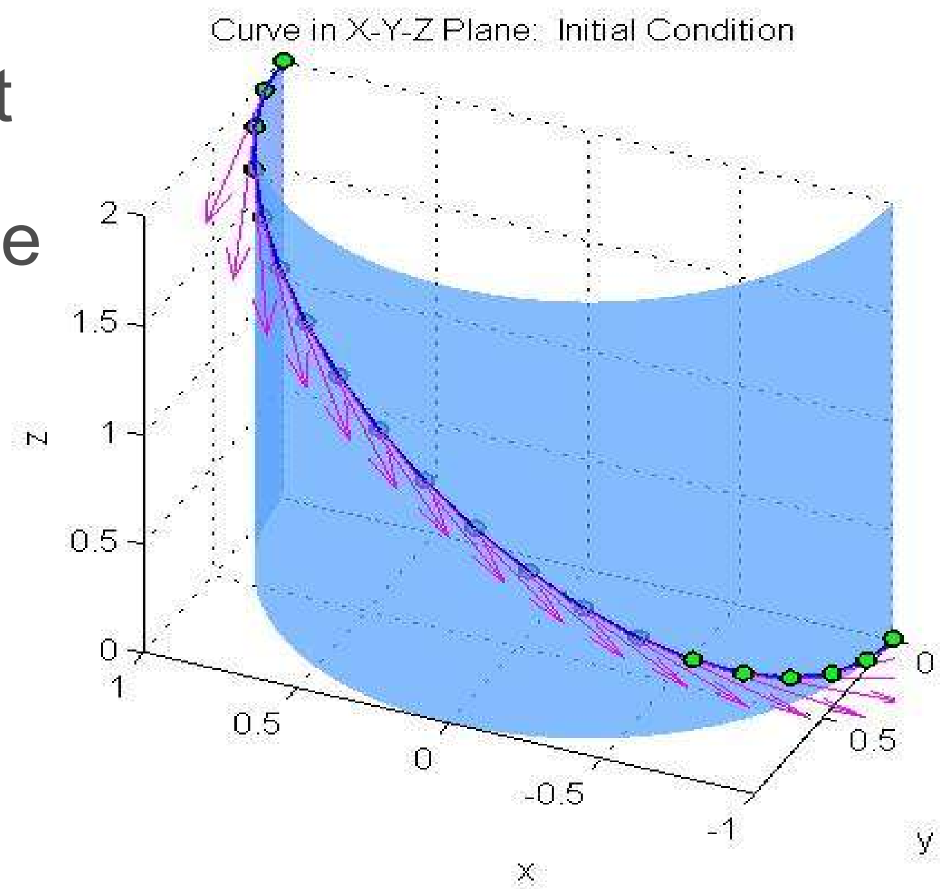
- Compute the tangent vector to the curve at each point

$$\tau = \frac{\partial x}{\partial s}\hat{\mathbf{i}} + \frac{\partial y}{\partial s}\hat{\mathbf{j}} + \frac{\partial z}{\partial s}\hat{\mathbf{k}}$$

- Calculate a signed scalar quantity at each point that represents a velocity with which each point will move

$$M_s = (\hat{\mathbf{n}} \times \hat{\tau}) \cdot \mathbf{k_g}$$

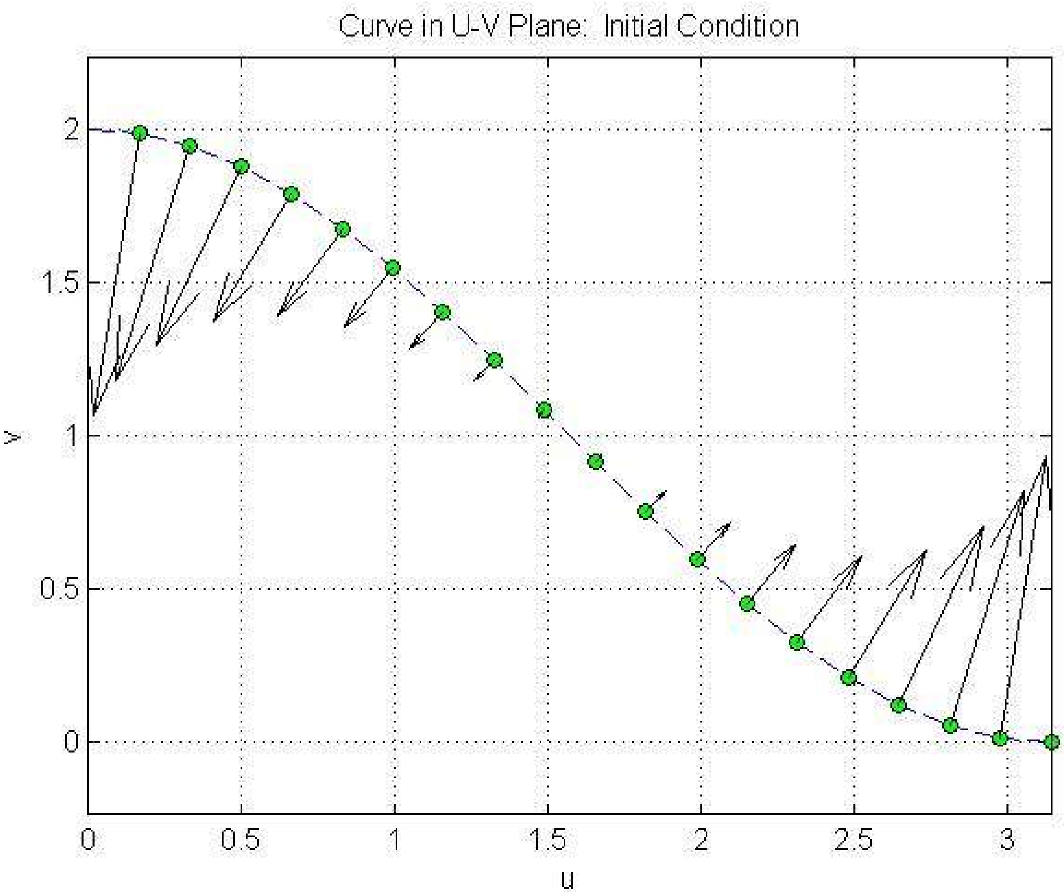
- At every point along the curve in the u, v plane calculate a local normal to the curve $\hat{\mu}$



The Evolution Algorithm

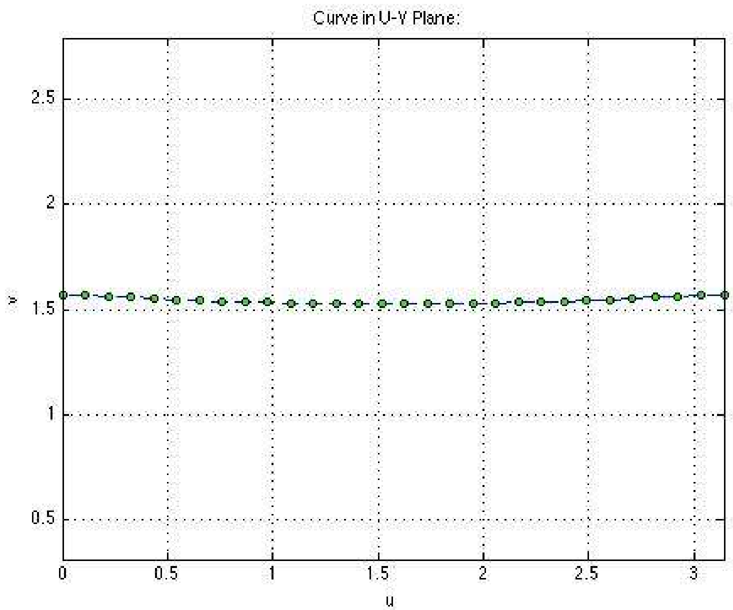
- Move each point in the [redacted] plane in the direction of the normal vector to the curve and proportional to the calculated velocity

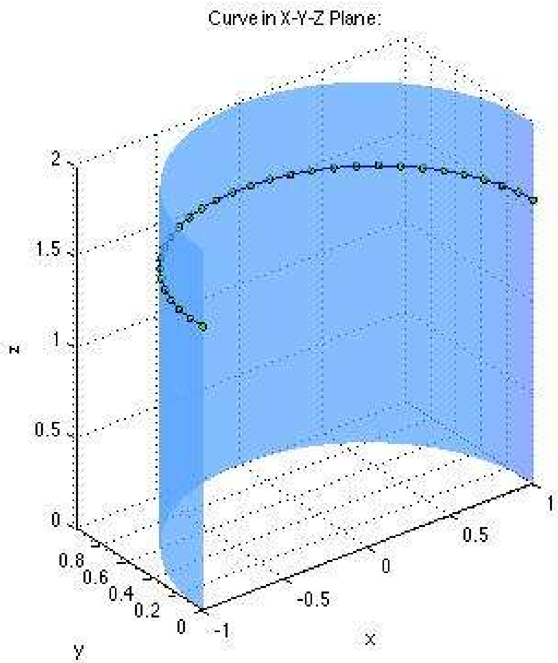
$$\frac{d\mathbf{r}}{dt} = \hat{\mu}\mathbf{M_s}$$



The Evolution Algorithm

- Repeat previous steps until the stopping criterion ($\left(\sum_{i=1}^{n} |\mathbf{k_g(s_i)}|^{\mathbf{2}}\right)^{1/2} = \epsilon_{tol} \approx 0$) is satisfied:





Implementation

- Derivatives Subroutine (CGUBC_deriv)
  - **Central Difference Method**
    - Calculates first derivatives quickly

$$\mathbf{f'(x)} \approx \frac{\mathbf{\Delta x_L}}{\mathbf{\Delta x_L + \Delta x_R}}\left(\frac{\mathbf{\Delta f_R}}{\mathbf{\Delta x_R}}\right) + \frac{\mathbf{\Delta x_R}}{\mathbf{\Delta x_L + \Delta x_R}}\left(\frac{\mathbf{\Delta f_L}}{\mathbf{\Delta x_L}}\right)$$

    - Computes second derivatives with low accuracy
  - **Cubic Spline Method**
    - Numerically calculates more accurate second derivatives
    - Increases complexity of subroutine

Implementation

- Time Integration:
- 4th Order Runga-Kutta:

$$\mathbf{r^{n+1}} = \mathbf{r^{n}} + \frac{\boldsymbol{\Delta}\mathbf{t}}{\mathbf{6}}(\mathbf{k_1} + \mathbf{2k_2} + \mathbf{2k_3} + \mathbf{k_4}),$$

$$k_1 = F(\mathbf{r^n}),$$

$$k_2 = F(\mathbf{r^n} + \frac{\boldsymbol{\Delta}\mathbf{t}}{\mathbf{2}}\mathbf{k_1}),$$

$$k_3 = F(\mathbf{r^n} + \frac{\boldsymbol{\Delta}\mathbf{t}}{\mathbf{2}}\mathbf{k_2}),$$

$$k_4 = F(\mathbf{r^n} + \boldsymbol{\Delta}\mathbf{t}\mathbf{k_3}),$$

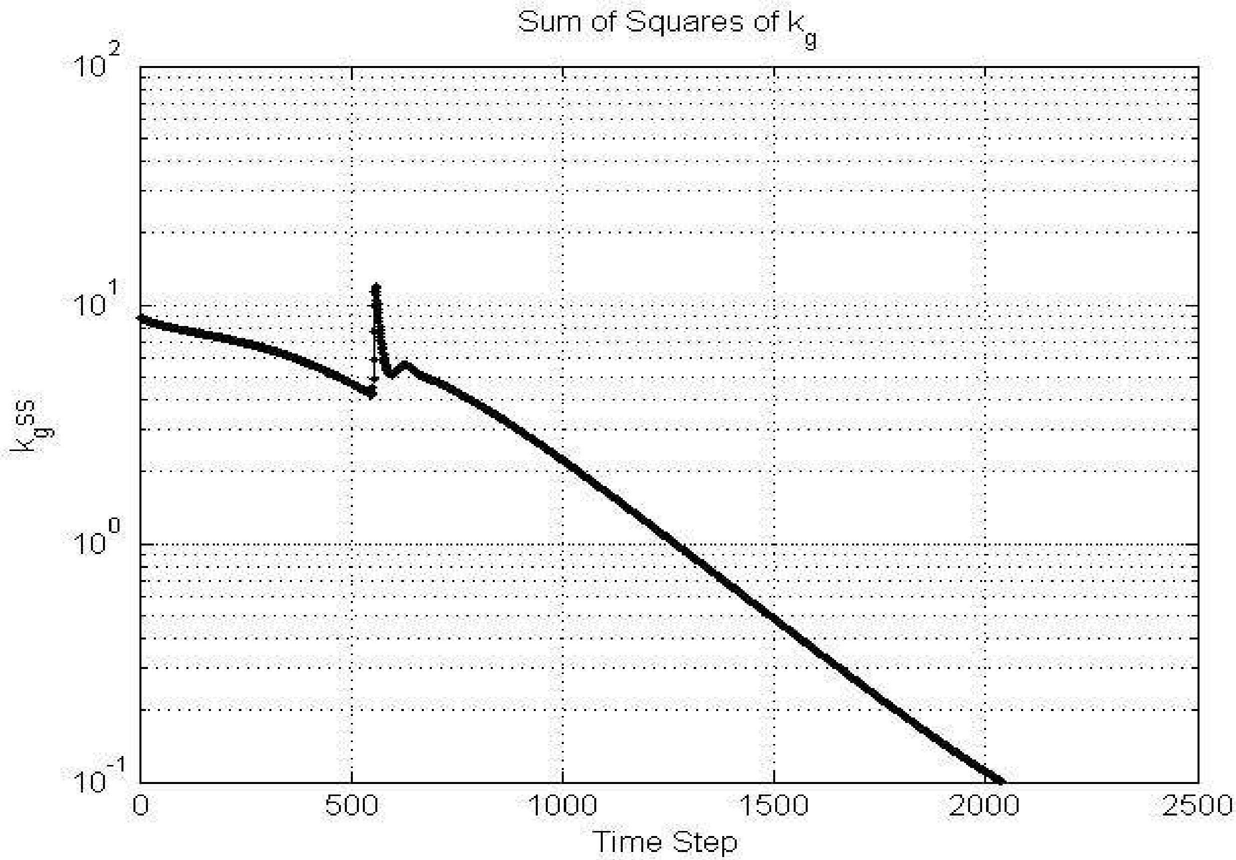
- Time step needs to be chosen carefully

Example #1

- Surface: Cylinder
- Initial condition near geodesic

Example #2

- Surface: Cylinder
- Complicated initial condition
- We see clumping of points and instability “blip”



Example #3a

- Surface: Cylinder
- We see boundary condition enforcement
- Clumping, but no blip.

Example #3b

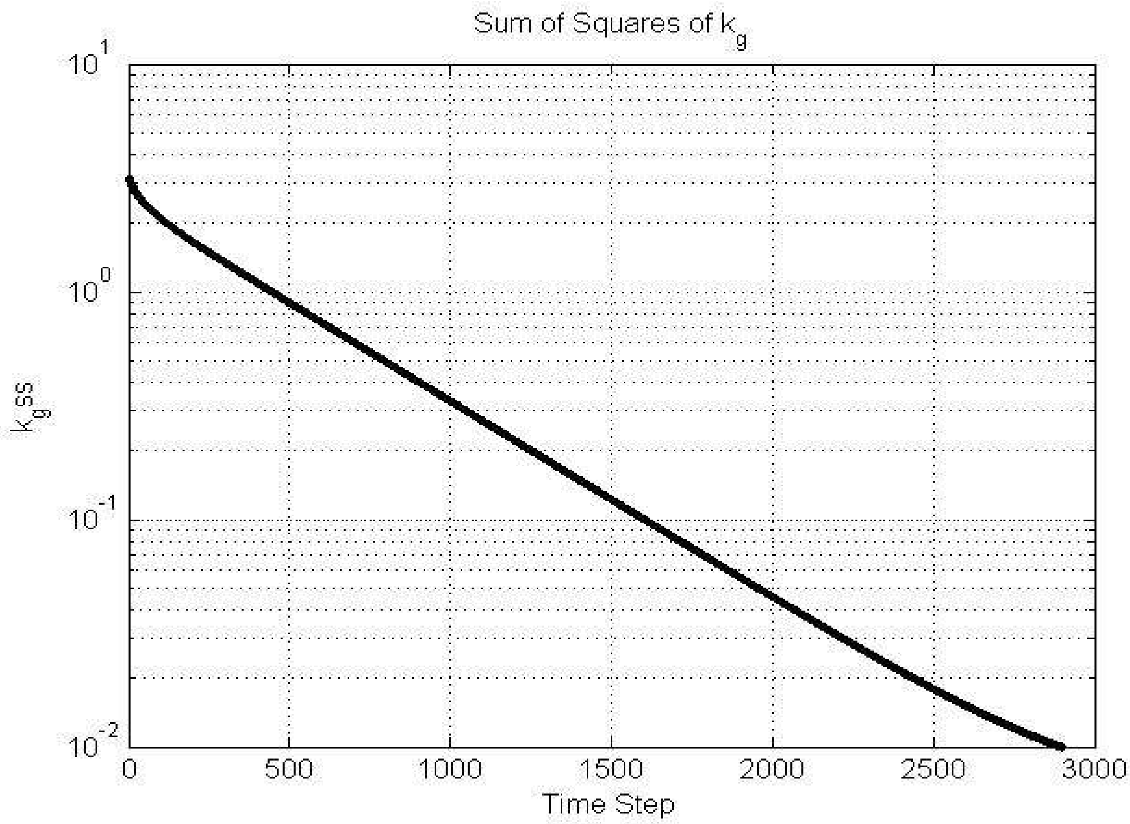
- Surface: Cylinder
- Using cubic spline derivatives
  - **Less clumping**

Example #3c

- Surface: Cylinder
- Using u-redistribution
  - **No clumping**

Example #4

- Surface: Sphere
- More sensitive
- Still works well for “nice” initial curves away from pole



Challenges & Solutions

- Challenges
  - **Convergence rate slow for some problems**
  - **Stability requirements on time step**
  - **“Singular” points cause breakdown**

- Possible solutions
  - **Implicit time scheme**
  - **Manual/automatic singular point computations**

Conclusions / Results

- Current evolution process can be used to relax an initial curve toward a geodesic
- Some issues remain to be addressed
- In principle, the algorithm should be able to handle constraints (with properly modified stopping criterion) and multiple surfaces

Future Work

- Constraints on surfaces
  - **Include regions that are inaccessible**
- Multiple surfaces
  - **Crossing edges**
- Other test cases
  - **Objects with corner**

Acknowledgements

Tom Grandine, Boeing Liaison

Ali Nadim, CGU faculty mentor

This material is based upon work supported by the National Science Foundation under Grant No. 0538663.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] S.V. Fomin I.M. Gelfand. *Calculus of Variations*. Dover Publications, Inc., 2000.

[2] C.H. Papadimitrious J.S.B. Mitchell, D.M. Mount. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.

[3] M. Schmies K. Polithier. Straightest geodesics on polyhedral surfaces. pages 30–38, 2006.

[4] E.J. Candes L. Ying. Fast geodesics computation with the phase flow method. *Journal of Computational Physics*, 220(1):6–18, 2006.

[5] E.J. Candes L. Ying. The phase flow method. *Journal of Computational Physics*, 220(1):184–215, 2006.

[6] J.A. Sethian R. Kimmel. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences of the USA*, 95(15):8431–8435, 1998.

[7] D. Breen S. Mauch. A fast marching method of computing closest points.

[8] V. Toponogov. *Differential geometry of curves and surfaces: a concise guide.* Springer, 2006.

[9] D. Kirsanov S.J. Gortler H. Hoppe V. Surazhsky, T. Sur. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph*, 24(3):553–560, 2005.

[10] S.-M. Hu Y.-J. Liu, Q.-Y. Zhou. Handling degenerate cases in exact computation on triangle meshes. *The Visual Computer*, 23(9):661–668, September 2007.

[11] D. Zorin. Curvature and geodesics, discrete laplacian and related smoothing methods. pages 1–6, 2002.