Computing Tools Project

Bone Fingers Stress Strain IGA-NURBS-BSplines Applications.

F. Ramírez[a,b], V. M. Calo[d],
Andrés González-Mancera[a,c],
G. A. Espinosa[a,b]

Uniandes Ph.D. Student:
Gabriel Andrés Espinosa Barrios
g-espino@uniandes.edu.co

2013-10 Semester.

[a]Biomedical Engineering Group (GIB), Universidad de los Andes, Bogotá D.C., Colombia
[b]Department of Civil and Environmental Engineering, Universidad de los Andes, Bogotá D.C., Colombia
[c]Department of Mechanical Engineering, Universidad de los Andes, Bogotá D.C., Colombia
[d]Faculty of Science and Engineering, Curtin University, Bentley, Perth, Western Australia

Contents

Dealing with patient-specific geometry modeling and meshes from imaging data is one of the biggest issues in biomechanical simulation-based medical planning for Bio-inspired concepts. Typical site of musculoskeletal disorders injuries are related with the daily use of the hand. For complex geometry and PDE´s (Partial Differential Equations) governed problems, the NURBS (Non-Uniform Rational B-Spline) based on IGA(isogeometric analysis) is proposed as accurate computational solving technique. This study uses both presents approach for construction of solid NURBS meshes for patient-specific hand metacarpus bones geometric models for PDE-IGA and a computerized axial tomography imaging reconstruction, for those 3D(Three-Dimensional) scanning process to obtain acurate IGA geometry of the model for stress-strain analysis of patient pain. Hypermatrix model for the molecular relationship of osteocyte bone cells of the fingers are segmented with pixel analysis for bone cells trying to understand contours fractures as the metacarpal area. The nonlinear model with the static study solved by step-increasing the load and updating the geometry. It was observed that the join of the osteocyte bone and tendons cells are stiff to analyze, and segment; but is the relevant interface region related with the disease of general biological bones tissues.

A strong bibliographic review of NURBS (Non-uniform rational B-splines)was make to what it was and what it means. Background to study the basis functions of Splines, Bezier Curves, and the B-Splines. On the other hand, it was working to acquire the first Medical Images segmentation that works with the NURBS modeling of the bones of the hand. It was obtaining a first segmentation algorithm and the extraction of the cloud control points of the bones of the hand. The first limitation of this was that the segmentation is not perfect. There is a loss of information of the bone and there are additional data of other bones that it needs to be removed by hand.

To having more clear what are the NURBS and where it comes. The developed to implement own algorithms to generate not only Surface NURBS, that is like traditional, but also Solid Volumetric NURBS. Furthermore, revised the theory about geometrical meaning of the weights is important. It also was implemented what is known as Interpolation NURBS to force to pass the curves on the control points. Theories were reviewed about how to modify NURBS with least squares, but it wasn't possible to implement.

The main goal, target, of this project is to obtain an algorithm and methodology for the surface NURBS of the bone, and Solid Volumetric NURBS file for Stress - Strain simulations of the bones applying python codes..

1 Introduction

1.1 Introduction; Review of: "IGA(Isogeometric Analysis) Toward Integration of CAD and FEA". T. J.R. Hughes et al.[1]

Dealing with patient-specific geometry modeling and meshes from imaging data is one of the biggest issues in biomechanical simulation-based medical planning for Bio-inspired concepts. Typical site of musculoskeletal disorders injuries are related with the daily use of the hand. For complex geometry and PDE´s (Partial Differential Equations) governed problems, the NURBS (Non-Uniform Rational B-Spline) based on IGA(isogeometric analysis) is proposed as accurate computational solving technique. This study uses both presents approach for construction of solid NURBS meshes for patient-specific hand metacarpus bones geometric models for PDE-IGA and a computerized axial tomography imaging reconstruction, for those 3D(Three-Dimensional) scanning process to obtain acurate IGA geometry of the model for stress-strain analysis of patient pain. Hypermatrix model for the molecular relationship of osteocyte bone cells of the fingers are segmented with pixel analysis for bone cells trying to understand contours fractures as the metacarpal area. The nonlinear model with the static study solved by step-increasing the load and updating the geometry. It was observed that the join of the osteocyte bone and tendons cells are stiff to analyze, and segment; but is the relevant interface region related with the disease of general biological bones tissues.

1.2 Math Start Matlab Images Analysis Segmentation Approach

Test to merge Matlab codes to Python applications.
Initial trial uploading DICOM files and generate soldis = **"CTSolidsGeneration.m"**

```
%% % 201304 GAEB Gabriel Andres Espinosa Barrios ID:1102798647 Sincelejo ,
% Sucre, Colombia. University of los Andes Code:200311728
% Computing Tools Project - Bone Fingers Stress Strain IGA-NURBS-BSplines Applications
% Computing Tools(Herramientas Computacionales) - Uniandes.

% Student:
% GAEB(Gabriel Andrs Espinosa Barrios)
%  PhD Graduate Student. Cod:200311728 Id:110279864X.
% University of the Andes(Universidad de los Andes). Bogota(Bogot) D.C., Colombia.

% Gabe Espinosa Start codes in Matlab
% email: gabrielespinosa@gmail.com ; g-espino@uniandes.edu.co
% Ph.D. Student.
% University of los Andes (Universidad de los Andes). Bogota D.C., Colombia.
```

```
%% %-------MANUAL EDITING--------%
%---Manual modifications of control points----%
cmod=Points(:,:,:);
%------------------------------------------
cmod(4,:)=1; % Including weigths
%% NURBS Generation
% polinomials order
p=2;
q=1;
s=1;
% Knot vectors
n=size(cmod,2);
m=size(cmod,3);
uknot=knotvector1D(n,p);
vknot=knotvector1D(m,q);

% spmod=rsmak({uknot,vknot},cmod);
% figure()
% fnplt(spmod)

%% initializing solid Control Points
 csolid=cmod(:,:,:);
 csolid(:,:,:,2)=csolid(:,:,:,1);
 for i=1:size(csolid,3)
      xm=mean(cmod(1,:,i));
      ym=mean(cmod(2,:,i));
       csolid(1,:,i,2)=xm;
       csolid(2,:,i,2)=ym;
 end
wknot=knotvector1D(size(csolid,4),s);
coefs = csolid;
knots = {uknot vknot wknot};
bone = nrbmak(coefs,knots);
geo=bone;
figure()
nrbplot(bone, [40 40 40], 'light', 'off');
hold on
%2D-Surface for testing ctrlpnt
% coefs = cmod;
% knots = {uknot vknot};
% bone = nrbmak(coefs,knots);
% figure()
% nrbctrlplot(bone);

%% Saving adquired geometry
 save ('geof2Mp32', 'geo')
 save ('bpcf2Mp32.mat', 'coefs')
```

1.3 Math Python Images Analysis Segmentation Approach

Test to merge Matlab codes to Python applications.
Initial trial uploading DICOM files = **"loadingprocessdicom.py"**

```
#!/usr/bin/env python ## #!/usr/local/bin/ python
#loading_process_dicom.py
"""
2018 Computing Tools Project  - Bone Fingers Stress Strain IGA-NURBS-BSplines Applications
Computing Tools(Herramientas Computacionales) - Uniandes.

Student:
GAEB(Gabriel Andrés Espinosa Barrios)
Returning/re-entry Engineering PhD Graduate Student. Cod:200311728 Id:110279864X.
University of the Andes(Universidad de los Andes). Bogota(Bogotá) D.C., Colombia.

Gabe Espinosa codes for Python
email: gabrielespinosa@gmail.com ; g-espino@uniandes.edu.co
Ph.D. Student. Returning/re-entry.
University of los Andes (Universidad de los Andes). Bogota D.C., Colombia.
"""

import numpy as np
import pylab as plt
import dicom as dcm
import skimage.filter as filt
import glob as glob
from pyvtk2 import *

def DICOMreadImages():
    """
    Reads in all the DICOM images in a directory and assembles them
    into a single numpy volume. Assumes files are sequentially
    numbered. files types *dcm. Files "number".dcm; example 0388.dcm.
    """
   datafiles = glob.glob("./DICOM/*.dcm")
    vol = None
    for fname in datafiles:
        ds = dcm.read_file(fname)
        num = int(fname.split('.')[1].split("-")[-1])
        I = ds.pixel_array
        if vol == None:
            vol = np.zeros((I.shape[0],I.shape[1],len(datafiles)))
        vol[:,:,num-1] = I
    return vol

DICOMreadImages():
```

1.4 Open Knot Vectors:

An indexed Vector "Knot Vectors" is open when it's first and last value are showed on $\mathbf{P}+1$ times. Where the $\mathbf{P}$ is the polynomial order. For instance, the polynomial cubic $\mathbf{P}=3$ is:

$$\Xi = [0,0,0,1,2,3,4,4,4]$$

The open knot vector are the standard for CAD(Computer Analysis Design).

1.5 Open Uniform B-splines[2]:

The B-Splines are the Blending-Mixture between the Open Uniform B-Splines and the Non-Uniform B-Splines. There are sometimes agree as some special Uniform B-Splines; nevertheless, there are classified as Non-uniform B-Splines.

Examples:

$$[0,0,1,2,3,3] \text{ for } \mathbf{P}(=1)+1=2 \text{ and } \mathbf{n}=3$$

$$[0,0,0,0,1,2,2,2,2] \text{ for } \mathbf{P}=3 \text{ and } \mathbf{n}=4$$

These Knot vectors can be normalized for the intervals $0-1; 0 \leq \mathbf{u} \leq 1$ in this way:

$$[0,0,0,33,0,67,1,1] \text{ for } \mathbf{P}=1 \text{ and } \mathbf{n}=3$$

$$[0,0,0,0,0,5,1,1,1,1] \text{ for } \mathbf{P}=3 \text{ and } \mathbf{n}=4$$

In general for any value of "$\mathbf{P}$" and "$\mathbf{n}$"($\mathbf{n}$ is the number of control points -1) could be generate a "open knot vector" with integer values through the following calculations:

$$\mathrm{u}_j = \begin{cases} 0 & \text{for} \quad 0 \leq j < \mathbf{P}+1 \\ j-\mathbf{P}+1+1 & \text{for} \quad \mathbf{P}+1 \leq j \leq \mathbf{n} \\ \mathbf{n}-\mathbf{P}+1+2 & \text{for} \quad j > \mathbf{n} \end{cases}$$

Size of Knot Vectors is $= \mathbf{n}+\mathbf{P}+1+1$.

(1)

For j that show from 0 until $\mathbf{n}+\mathbf{P}+1$. The first $\mathbf{P}+1$ Knots are assigned to zero 0; Otherwise, the last $\mathbf{P}+1$ knots have a value of $\mathbf{n}-p+3$.

1.6 Open Knot Vectors(Continuations):

"*The Open B-Splines have very similar characteristics to the* 'Bézier Splines'."[2] Even though, when ($\mathbf{P}+1=\mathbf{n}+1 \Rightarrow \mathbf{n}=\mathbf{P}$)there are polynomial order of control points(CP) +1 (CP=$\mathbf{P}+1$); the open B-Splines are reduced to the "Bézier Splines" and all of the values of the index vector "Knot Values" are 0 or 1. For instance, with a open B-Spline of four-4 control points($\mathbf{n}=3$) and cubic ($\mathbf{P}=3$) the index vector is:

$$\Xi = [0,0,0,0,1,1,1,1]$$

The polynomial curve of a Open B-Spline go through for the first and the last Control Point - CP. Figure 1.

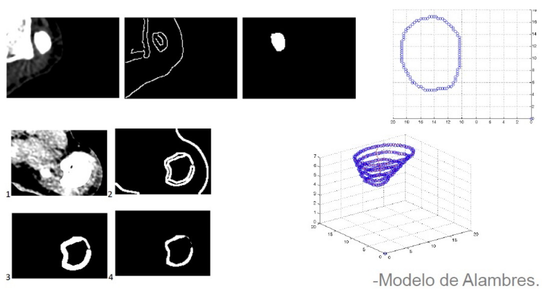

Figure 1: Results of the polynomial curve of the B-Spline Segmentation.

Even as in the Bézier curves; specify multiples control points in just one coordinate, push any B-Spline to that location. Therefore, it could be generate close curves with "open B-Splines" specifying the first and the last control points in the same position. Figure 2.

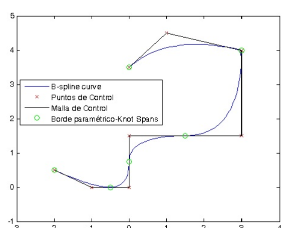

Figure 2: Polynomial curve of the Open B-Spline.

[1] As a consequence of the use of "Open Knot Vectors" in multiple dimensions; the border of one "B-Spline object" with "$\mathbf{d}$" parametric dimensions is that the "B-Spline object" will had "$\mathbf{d}$" -1 dimensions.
i.e. each border of one B-Spline surface is one B-Spline curve. Figure 3.

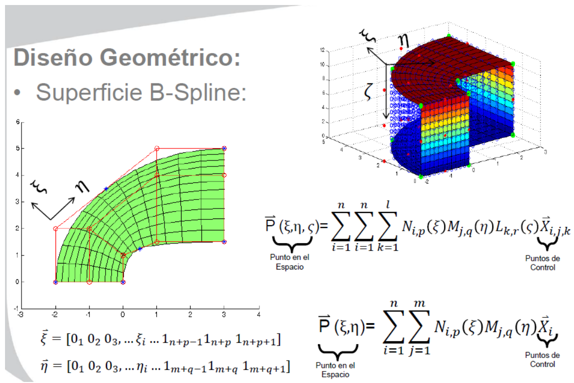

Figure 3: 2D-3D Polynomial B-Spline Geometry.

2 Basic Functions(A.K.A:"Blending Functions". Open Knot Vectors Continuations)[1]:

$$\mathbf{N}_{i,0}(\xi)\text{=¿? } \Leftrightarrow \mathrm{B}_{k,1}(\mathbf{u})$$

$i = \mathbf{n}$ control points $\Leftrightarrow$ control point i
$0 = \mathbf{P}$. Polynomial Order $\Leftrightarrow$ B-Spline or Basic Polynomial order degree zero 0
$k = \mathbf{n} + 1$
$1 =$ "$\mathbf{d}$" Parametric dimensions $\Leftrightarrow$ Parametric dimensions order one 1

2.1 Recursive Functions of Cox-de Boor(Basic Functions Continuations):

The Cox-de Boor recursion formula are functions to join the B-Splines curves. Usually, we can obtain a expression for the calculus to the positions along of a B-Spline curve, with the couple of functions of the "Blending-function"(Basic Functions) formulation thereby:

$$\mathbf{P}(\mathbf{u}) = \sum_{k=0,1}^{\mathbf{n}} \mathbf{P}_k \mathbf{B}_{k,\mathbf{P}}(\mathbf{u})$$
$$\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}$$
$$1 \leq \mathbf{P} \leq \mathbf{N} \text{ (Control points)}$$

(2)

Where:

1. $\mathbf{P}_k$= There are $\mathbf{n}+1$ control points.
2. $\mathbf{B}_{k,\mathbf{d}}$= There are the polynomials of degree "$\mathbf{P}$". Where "$\mathbf{P}$" could be selected to any integer between 1 until the last control points ($\mathbf{N} = \mathbf{n} + 1$).

It could be achieve B-Spline local control, defining the join functions "Blending functions"(Basic Functions) about the subinterval of the total degree-rank of $\mathbf{u}$:

$$\mathbf{N}_{i,0}(\xi) \Leftrightarrow \mathbf{B}_{k,1}(\mathbf{u}) = \begin{cases} 1 & \text{if} \quad \mathbf{u}_k \leq \mathbf{u} \leq \mathbf{u}_{k+1} \ (\mathbf{P}=0);(\mathbf{d}=1=\mathbf{P}+1) \\ 0 & \text{if} \quad \text{otherwise} \end{cases} \tag{3}$$

$$\mathbf{B}_{k,\mathbf{P}+1}(\mathbf{u}) = \frac{\mathbf{u}+\mathbf{u}_k}{\mathbf{u}_{k+\mathbf{P}}-\mathbf{u}_k} \ \mathbf{B}_{k,\mathbf{P}}(\mathbf{u}) + \frac{\mathbf{u}_{k+\mathbf{P}+1}-\mathbf{u}}{\mathbf{u}_{k+\mathbf{P}+1}-\mathbf{u}_{k+1}} \ \mathbf{B}_{k+1,\mathbf{P}}(\mathbf{u})$$
$$\mathbf{P} = 1, 2, 3, 4, ...$$
$$\mathbf{d} = 2, 3, 4, 5, ...$$

(4)

Given that $\mathbf{u}_{k+\mathbf{P}} = \mathbf{u}_{\mathbf{P}}$; or $\mathbf{u}_{k+\mathbf{P}+1} = \mathbf{u}_{k+1} \Rightarrow$ any term evaluated as $\frac{0}{0} = 0$ will be equal to zero 0.
It is important is to show that even though the definitions before, we always need to take into a account that each Join Function or Basic Function constitute a part of unit i.e.:

$$\sum_{i=1}^{\mathbf{n}} \mathbf{N}_{i,\mathbf{P}}(\xi) = 1 \ \forall \ \xi \text{ or } \sum_{k=0}^{\mathbf{n}} \mathbf{B}_{k,\mathbf{d}}(\mathbf{u}) = 1 \ \forall \ \mathbf{u}$$

(5)

2.2 Relation between continuity and multiplicity(Basic Functions Continuations):

In general the basic functions of order $\mathbf{P}$ have $\mathbf{P} - \mathbf{m}_i$ continuities derivatives on the "knot" index $\mathbf{u}_i$.

Where $\mathbf{m}_i$: is multiplicity of the value $\mathbf{u}_i$ in the "Knot vector".

We need to remember that the multiplicity in a polynomial is the number of times that the value of $\mathbf{u}_i$ is the base-root of the polynomial. For instance:

$$\mathbf{P}(x) = x^3 + 2x^2 - 7x + 4 = (x+4)(x-1)^2$$

Where: -4 = multiplicity 1; 1 = multiplicity 2

(6)

In B-Splines-IGA the multiplicity of a Index Knot value $\mathbf{m}_i$ is given by the number of times that this value is repeated in the "Knot Vector".

Example:

$\Xi = [0, 1, 2, 3, 4, 4, 5]$ in this case $0 = multiplicity$ 1; and $4, 4 = multiplicity$ 2

3 Medical Image Analysis Segmentation:

Medical Image Analysis Segmentation is make for each finger bone. In each finger of the one hand the Distal phalange, Proximal phalange, Metacarpal, and Trapezium, are segmented with the analysis of the pixels, Voxels of each DICOM(Digital Imaging and Communications in Medicine) Medical Image. See Figure 4.

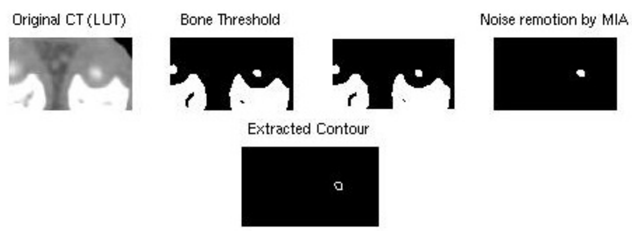

Figure 4: Pixel-Voxel Boundering Segmentation.

3.1 B-Spline curves models for bone segmentation:

A B-Spline curves was generated to understand the mesh and the control points of a DICOM bone Pixel-Voxel values of the images. First was generated a Circle and Oval curves to understand the cloud control points that was obtained for each DICOM image. See Figure 5.
Second, Oval deformed cloud control points was generated to understand this kind of segmentation of the DICOM bone image. See Figure 6.

4 B-Splines and NURBS Bones Modeling:

B-Splines and NURBS codes for bone segmentation process was generated to obtain solid bones of the fingers. B-Splines and NURBS formulate was make to obtain osteocytes cells of the bones. Index or Pointer Finger bones was segmented (see Figure 7) to obtain the solid of the bone to make stress - strain process.

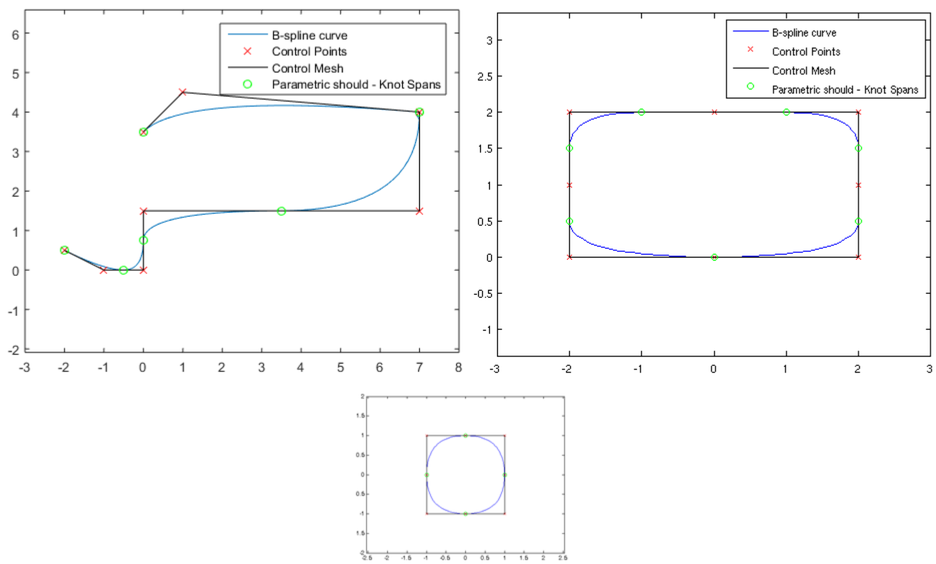

Figure 5: Left general B-Spline Curve. Right Oval B-Spline. Down Circle B-Spline.

images/20161130OvalDeformedBSplineImages.png

Figure 6: Oval Deformed B-Spline.

The codes was generated to segmented each bone of the finger: Distal phalange, Proximal phalange, Metacarpal, and Trapezium. However, the Medical Image Analysis was difficult to obtain the accurate segmentation of the bones. Therefore, Medical Image Analysis segmentation of others bones image its need to be done in order to obtain other accurate information.

Finally, Medical Image Analysis segmentation was make of the Index - Pointer Finger bones of the hand. A solid model of the bones was obtained to make stress - strain simulation to the forces in the bones. Future work is required for Stress - Strain simulation of the bones that was segmented.

References

[1] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, New York, 2009. 2, 4, 8, 9

[2] D. Hearn, M. Baker, Computer Graphics, Prentice-Hall., Englewood Cliffs, NJ, 1994. 2, 7, 8

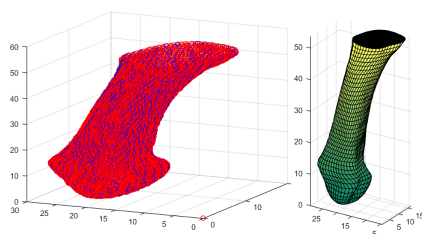

Figure 7: Left cloud control points of the Metacarpal bone. Right Solid results of the Metacarpal bone.