Three Dimensional Approximation of the Peach in MAPLE

Stanislav Bartoň
Mendel University of Agriculture and Forestry in Brno
Faculty of Agriculture, Department of Principal of Technics
e-mail: barton@mendelu.cz

Abstract

Linearized Gauss-Newton iteration method is used to determine main axes of three-dimensional ellipsoid approximating peach. Three independent photos displaying peach as ground, side and front view are used as data sources. Programm MAPLE 11 was used as a computer environment. A practical exapmle is presented in order to demonstrate the usage of all required commands. The quality of approximation is evaluated as a final part of the paper.

1 Introduction

Knowledge of the object's shape is usually one of critical parameters for evaluating its mechanical properties. Concerning peaches, the dimensions are significant e.g. for storage processing, deep freezing namely. Since only undamaged and healty fruits can be processed, it is important to determine maximum levels of mechanical loading, thus to determine the modulus of elasticity [1]. Modulus of elasticity can be determined by numerous nondestructive methods. One of them is based on measuring of acoustic wave propagation [2].

The most precise method of the object dimensions measuring is 3D scanning [3]. But this method is very expensive, demands a lot of time and manual work and finally produces huge bulks of data. Digging the main shape parameters represents rather complicated problem. This article shows the procedure of approximation of peach shape by triaxial ellipsoid and use of three digital photos. These photos must be taken in mutually perpendicular orientations, see Figure 1.

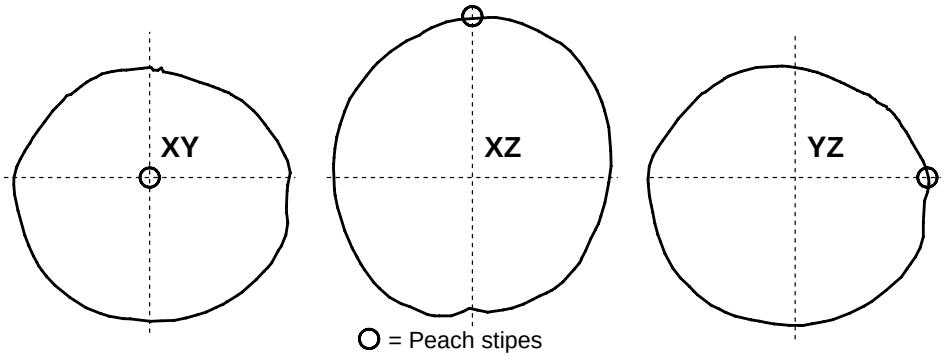

Figure 1: Basic peach photos.

There are plenty of softwares intended for extraction of pixels coordinates from the objects perimeter. Approach described in [4] was used in this paper. Coordinates of the perimeter's pixels were saved in the three independent files. Used algoritm reduces number of pixels and orders them in such way, that resulting polygon approximates perimeter with accuracy better than ± 1 pixel. This method reduces number of pixels from thousands into hundreds of polygon vertexes without loss of accuracy.

2 Coordinate system

Let us assume that peach will be approximated by an triaxial ellipsoid with the axis A, B, and C in the spherical coordinate system. If the peach is properly oriented, see Figure 1, photos can be interpreted as equatorial cross section - XY, and cut along 0° meridian cross section - XZ and 90° meridian - YZ. Cross section curves may be approximated by mutually dependent ellipses. Their semiaxis will be A, B for XY, A, C for XZ and B, C for YZ. It must be assumed that these ellipses will have different centers and may be rotated by a small angle around different centers. Following description will be used: $[Ax, Ay]$ = temporary unknown center of the ellipse approximating XY cut, Ra - rotation angle of the approximating ellipse around center. Variables $[Bx, Bz]$, Rb, $[Cy, Cz]$ and Rc have the same meaning for cross sections XZ and YZ. Parametric shape will be used to describe all three ellipses. The first ellipse will be decribed by the function $[A\cos(fa), B\sin(fa)]$, the second one by $[A\cos(fb), C\sin(fb)]$ and the third one by $[B\cos(fc), C\sin(fc)]$, where fa, fb, fc are mutually independent parameters, $-\pi \leq fa, fb, fc \leq \pi$. $[X, Y]$, $[Y, Z]$ and $[Y, Z]$ are coordinates of the polygon vertexes, saved in three different files. Using these variables we can determine difference between coordinates of the general vertex point the nearest corresponding point on the ellipse. We shall use the Least Squares Method, *(LSQ)*, to compute all unknown variables which will minimize $Qxy+Qxz+Qyz$, where:

$$\begin{aligned} Qxy = \quad & \Sigma_{i=1}^{Nxy} \quad (A\cos(fa_i) - (X_i - Ax)\cos(Ra) - (Y_i - Ay)\sin(Ra))^2 \\ & + (B\sin(fa_i) + (X_i - Ax)\sin(Ra) - (Y_i - Ay)\cos(Ra))^2 \ , \end{aligned} \tag{1}$$

$[X_i, Y_i]$ are coordinates of the polygon vertexes of the XY cut and fa_i are parameters of the points laying on the ellipse approximating polygon. Qxz and Qyz are defined very similarly.

3 Preparation of the Iteration

Equation (1) is non-linear for all unknowns. It means that iteration method based on linearization must be used. One of the best approaches derived by Newton and improved by Gauss is well known Gauss-Newton method. Purpose of the presented article is not to explain this method, but to show how to use it on complicated and large problem. This method is based on the matrices, that is why MAPLE library LinearAlgebra should be used. Library plots simplify results visualisation. Approach presented in this article arises as generalization of [5] and [6].

```
> restart; with(LinearAlgebra): with(plots):
```

In order to limit the scope, only commands and variables describing coordinate difference between vertex and corresponding point on the ellipse for XY cross section will be described. Other MAPLE commands are very similar.

```
> xa:=A*cos(fa)-((X-Ax)*cos(Ra)+(Y-Ay)*sin(Ra)):
> ya:=B*sin(fa)+((X-Ax)*sin(Ra)-(Y-Ay)*cos(Ra)):
> xb:=A*cos(fb)-((X-Bx)*cos(Rb)+(Z-Bz)*sin(Rb)):
> zb:=C*sin(fb)+((X-Bx)*sin(Rb)-(Z-Bz)*cos(Rb)):
> yc:=B*cos(fc)-((Y-Cy)*cos(Rc)+(Z-Cz)*sin(Rc)):
> zc:=C*sin(fc)+((Y-Cy)*sin(Rc)-(Z-Cz)*cos(Rc)):
```

The difference between origin of the series expansion and actual variable value will be substituted as a new variable.This variable has D at the begining of its name. Value of this variable will represent the correction for the corresponding variable during iteration process.

```
> Dsu:=map(u->u-cat(u||0)=cat(D||u),Var);
> Mxa:=subs(Dsu,Mxa); Mya:=subs(Dsu,Mya): Mxb:=subs(Dsu,Mxb):
> Mzb:=subs(Dsu,Mzb): Myc:=subs(Dsu,Myc): Mzc:=subs(Dsu,Mzc):
```

$$\begin{aligned} Mxa \quad := \quad & A0\cos(fa0) - (Y - Ay0)\sin(Ra0) - (X - Ax0)\cos(Ra0) + DA\cos(fa0) \\ & -A0\sin(fa0)Dfa + DAx\cos(Ra0) + DAy\sin(Ra0) \\ & +((X - Ax0)\sin(Ra0) - (Y - Ay0)\cos(Ra0))DRa \end{aligned}$$

Complications arise with the parameters. For each vertex/point the best possible initial approximation is needed. If we have N points we have to find N corresponding initial parameters. That is why correction for parameters from the Taylor series must be separated. Now it is possible to create a lists containing terms multiplying individual differences.

```
> JFxa:=select(has,Mxa,Dfa)/Dfa: JFya:=select(has,Mya,Dfa)/Dfa:
> JFxb:=select(has,Mxb,Dfb)/Dfb: JFzb:=select(has,Mzb,Dfb)/Dfb:
> JFyc:=select(has,Myc,Dfc)/Dfc: JFzc:=select(has,Mzc,Dfc)/Dfc:
> JXa:=map(u->select(has,Mxa,u)/u,map(u->rhs(u),Dsu[4..-1]));
> JYa:=map(u->select(has,Mya,u)/u,map(u->rhs(u),Dsu[4..-1]));
> JXb:=map(u->select(has,Mxb,u)/u,map(u->rhs(u),Dsu[4..-1])):
> JZb:=map(u->select(has,Mzb,u)/u,map(u->rhs(u),Dsu[4..-1])):
> JYc:=map(u->select(has,Myc,u)/u,map(u->rhs(u),Dsu[4..-1])):
> JZc:=map(u->select(has,Mzc,u)/u,map(u->rhs(u),Dsu[4..-1])):
```

$$JXa := [\cos(fa0), 0, 0\cos(Ra0), 0, 0, 0, 0, (X - Ax0)\sin(Ra0) - (Y - Ay0)\cos(Ra0), 0, 0]$$

Finally, separation of free terms is necessary.

```
> RXa:=remove(has,Mxa,map(u->rhs(u),Dsu));
> RYa:=remove(has,Mya,map(u->rhs(u),Dsu)):
> RXb:=remove(has,Mxb,map(u->rhs(u),Dsu)):
> RZb:=remove(has,Mzb,map(u->rhs(u),Dsu)):
> RYc:=remove(has,Myc,map(u->rhs(u),Dsu)):
> RZc:=remove(has,Mzc,map(u->rhs(u),Dsu)):
```

$$RXa := A0\cos(fa0) - (Y - Ay0)\sin(Ra0) - (X - Ax0)\cos(Ra0)$$

4 Initial values iteration

At this point, the files containing vertexes coordinates can be read. The whole polygon will be moved, so its mass center will be in the center of the coordinate system. Na, Nb, Nc represents number of vertexes in the idividual cross sections and Ta, Tb, Tc are corresponding centers of their masses.

```
> read "007a_3.sav": Na:=nops(XYL):
> Ta:=sum(XYL[i],i=1..Na)/Na: XY:=map(u->u-Ta,XYL):
> read "007b_3.sav": Nb:=nops(XYL):
> Tb:=sum(XYL[i],i=1..Nb)/Nb: XZ:=map(u->u-Tb,XYL):
> read "007c_3.sav": Nc:=nops(XYL):
> Tc:=sum(XYL[i],i=1..Nc)/Nc:  YZ:=map(u->u-Tc,XYL):
> 'Na'=Na, 'Nb'=Nb, 'Nc'=Nc, 'Ta'=Ta, 'Tb'=Tb, 'Tc'=Tc;
```

$$Na = 92,\ Nb = 80\ Nc = 96,$$
$$Ta = [407.744, 331.126],\ Tb = [446.333, 324.705],\ Tc = [400.202, 363.912]$$

Iinital values for axes may be determined as average of maximal diameters of individual cross sections and arguments of the vertexes in the polar coordinate system may be used as initial values of parameters. If photos were taken carefully the initial values for displacements and rotations may be zero

```
> A0:=0.5*(max(map(u->u[1],XY)[])+max(map(u->u[1],XZ)[])):
> Fa:=map(u->op(2,polar(u[1]+I*u[2])),XY):
> C0:=0.5*(max(map(u->u[2],XZ)[])+max(map(u->u[2],YZ)[])):
> Fb:=map(u->op(2,polar(u[1]+I*u[2])),XZ):
> Fc:=map(u->op(2,polar(u[1]+I*u[2])),YZ):
> Ax0:=0: Ay0:=0: Bx0:=0: Bz0:=0: Cy0:=0: Cz0:=0: Ra0:=0: Rb0:=0: Rc0:=0:
```

5 Iteration

The most important variable in the iteration process is the Jacobi Matrix J. This matrix is composed from 24 submatrices, 12 of them are zero matrices, see construction of the Maple variable J. Its construction shows the Figure 2 and exactly corresponds with the structure of

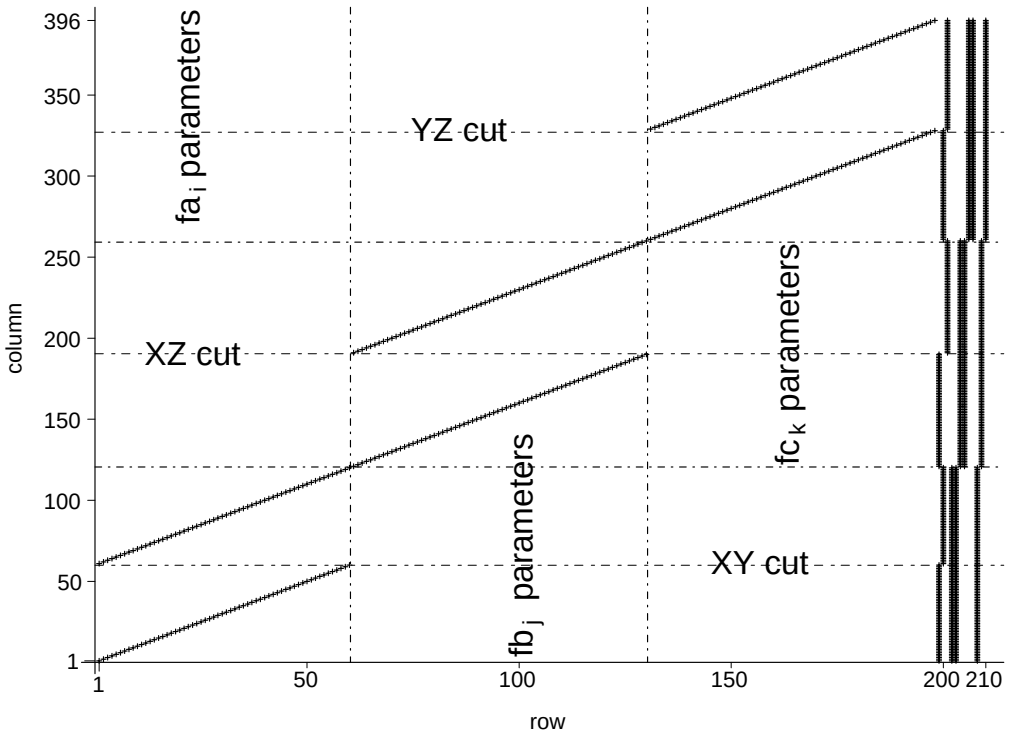

Figure 2: Structure of the Jacobi matrix.

the command line for the variable J. Non zero elements are shown as crosses. It can be seen, that Jacobi matrix is very sparse, partially filled up columns corespond to corrections of the variables [A, B, C, Ax, Ay, Bx, Bz, Cy, Cz, Ra, Rb, Rc].

The vector of right sides contains variable R and vector DV is vector of corrections of initial values of searched variables. Variable *eps* controls whole iteration, which runs until *eps* is less than 10^{-5}. Quadratic norm of the vector of corrections DV is assigned into *eps*. To reduce the big steps between corrections, corrections will be reduced to one half. Because at first few passes through loop corrections the parameters may be higher than 2π, it is necessary to recompute

all parameters into range $< -\pi, \pi >$. Subsequently, the corrections to all other variables are added.

```
> eps:=1: while eps>1e-5 do;
>  J:=Matrix([[DiagonalMatrix(map(u->subs(fa0=u,JFxa),Fa)),ZeroMatrix(Na,Nb),
>   ZeroMatrix(Na,Nc), Matrix(zip((u,v)->subs(fa0=u,X=v[1],Y=v[2],JXa),Fa,XY))],
>   [DiagonalMatrix(map(u->subs(fa0=u,JFya),Fa)),ZeroMatrix(Na,Nb),
>   ZeroMatrix(Na,Nc),Matrix(zip((u,v)->subs(fa0=u,X=v[1],Y=v[2],JYa),Fa,XY))],
>   [ZeroMatrix(Nb,Na),DiagonalMatrix(map(u->subs(fb0=u,JFxb),Fb)),
>   ZeroMatrix(Nb,Nc), Matrix(zip((u,v)-subs(fb0=u,X=v[1],Z=v[2],JXb),Fb,XZ))],
>   [ZeroMatrix(Nb,Na),DiagonalMatrix(map(u->subs(fb0=u,JFzb),Fb)),
>   ZeroMatrix(Nb,Nc),Matrix(zip((u,v)->subs(fb0=u,X=v[1],Z=v[2],JZb),Fb,XZ))],
>   [ZeroMatrix(Nc,Na),ZeroMatrix(Nc,Nb),DiagonalMatrix(map(u->subs(fc0=u,JFyc),
>   Fc)),Matrix(zip((u,v)->subs(fc0=u,Y=v[1],Z=v[2],JYc),Fc,YZ))],
>   [ZeroMatrix(Nc,Na),ZeroMatrix(Nc,Nb),DiagonalMatrix(map(u->subs(fc0=u,JFzc),
>   Fc)),Matrix(zip((u,v)->subs(fc0=u,Y=v[1],Z=v[2],JZc),Fc,YZ))]]):
>  R:=-Vector([zip((u,v)->subs(fa0=u,X=v[1],Y=v[2],RXa),Fa,XY)[],
>   zip((u,v)->subs(fa0=u,X=v[1],Y=v[2],RYa),Fa,XY)[],zip((u,v)->subs(fb0=u,
>   X=v[1],Z=v[2],RXb),Fb,XZ)[],zip((u,v)->subs(fb0=u,X=v[1],Z=v[2],RZb),
>   Fb,XZ)[],zip((u,v)->subs(fc0=u,Y=v[1],Z=v[2],RYc),Fc,YZ)[],
>   zip((u,v)->subs(fc0=u,Y=v[1],Z=v[2],RZc),Fc,YZ)[]]):
>  DV:=LeastSquares(J,R):eps:=Norm(DV,2):DV:=0.5*DV;
>  Fa:=Fa+convert(DV[1..Na],list):Fb:=Fb+convert(DV[1+Na..Na+Nb],list):
>  Fc:=Fc+convert(DV[1+Na+Nb..Na+Nb+Nc],list):A0:=A0+DV[Na+Nb+Nc+1];
>  B0:=B0+DV[Na+Nb+Nc+2]; C0:=C0+DV[Na+Nb+Nc+3];Ax0:=Ax0+DV[Na+Nb+Nc+4];
>  Ay0:=Ay0+DV[Na+Nb+Nc+5]; Bx0:=Bx0+DV[Na+Nb+Nc+6];Bz0:=Bz0+DV[Na+Nb+Nc+7];
>  Cy0:=Cy0+DV[Na+Nb+Nc+8]; Cz0:=Cz0+DV[Na+Nb+Nc+9];Ra0:=Ra0+DV[Na+Nb+Nc+10];
>  Rb0:=Rb0+DV[Na+Nb+Nc+11]; Rc0:=Rc0+DV[Na+Nb+Nc+12];
>  Fa:=map(u->op(2,polar(cos(u)+I*sin(u))),Fa):Fb:=map(u->op(2,polar(cos(u)+
>  I*sin(u))),Fb):Fc:=map(u->op(2,polar(cos(u)+I*sin(u))),Fc):
>  Ra0:=op(2,polar(cos(Ra0)+I*sin(Ra0))):Rb0:=op(2,polar(cos(Rb0)+I*sin(Rb0))):
>  Rc0:=op(2,polar(cos(Rc0)+I*sin(Rc0))):
> end do:
```

6 Results interpretation

At first, the accuracy of the approximation of individual cross sections can be displayed, see Figure 3.

```
> Xa:=A0*cos(f); Ya:=B0*sin(f):
> Xra:=cos(Ra0)*Xa-sin(Ra0)*Ya+Ax0: Yra:=sin(Ra0)*Xa+cos(Ra0)*Ya+Ay0:
> Xb:=A0*cos(f); Zb:=C0*sin(f):
> Xrb:=cos(Rb0)*Xb-sin(Rb0)*Zb+Bx0:Zrb:=sin(Rb0)*Xb+cos(Rb0)*Zb+Bz0:
> Yc:=B0*cos(f); Zc:=C0*sin(f): Yrc:=cos(Rc0)*Yc-sin(Rc0)*Zc+Cy0:
> Zrc:=sin(Rc0)*Yc+cos(Rc0)*Zc+Cz0;
> P1:=plot(XY,style=point,symbol=circle,symbolsize=20,color=black):
> P2:=plot([Xra,Yra,f=0..2*Pi],color=black,thickness=2,thickness=2,linestyle=1):
> P3:=plot(XZ,style=point,symbol=cross,symbolsize=20,color=black):
> P4:=plot([Xrb,Zrb,f=0..2*Pi],color=black,thickness=2,thickness=2,linestyle=3):
> P5:=plot(YZ,style=point,symbol=box,symbolsize=20,color=black):
> P6:=plot([Yrc,Zrc,f=0..2*Pi],color=black,thickness=2,linestyle=4):
> display({P1,P2,P3,P4,P5,P6});
```

At second, correlation between vertex points and corresponding points on the individual ellipses and the average of the relative displacements can be computed.

```
> R1:=map(u->sqrt(u[1]^2+u[2]^2),XY):
> R2:=map(u->evalf(subs(f=u,sqrt(Xra^2+Yra^2))),Fa):
> DR:=zip((u,v)->abs(u-v),R1,R2): AR:=zip((u,v)->0.5*(u+v),R1,R2):
> RRa:=zip((u,v)->u/v,DR,AR): MRa:=sum(RRa[i],i=1..Na)/Na:
> CRa:=stats[describe,linearcorrelation](R1,R2):
> R1:=map(u->sqrt(u[1]^2+u[2]^2),XZ):
```

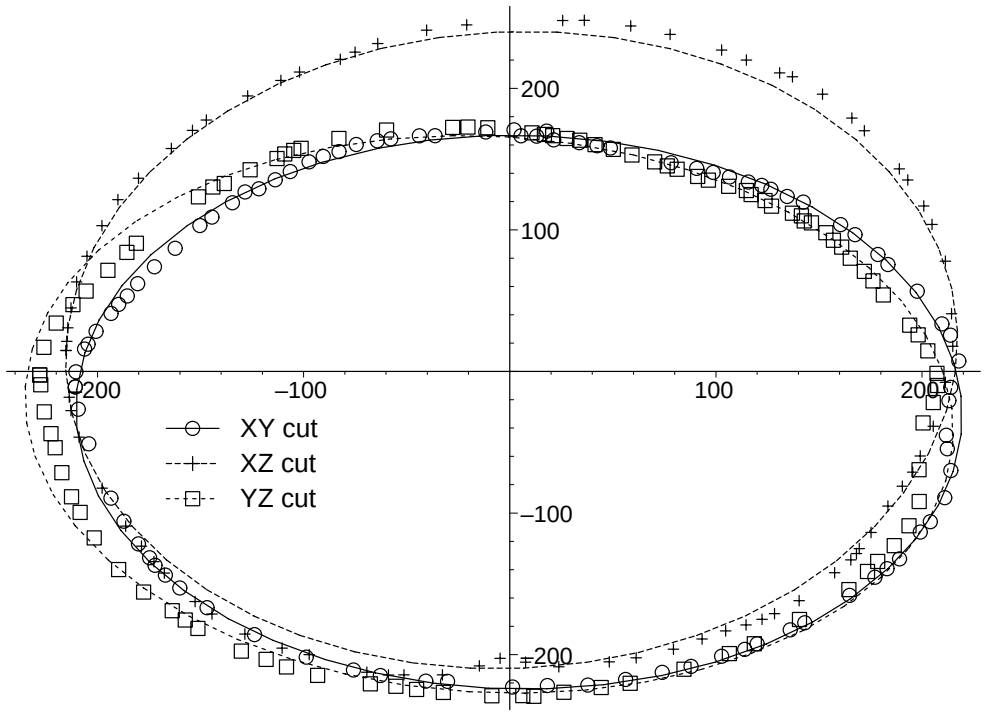

Figure 3: Accuracy of the individual cross sections.

```
> R2:=map(u->evalf(subs(f=u,sqrt(Xrb^2+Zrb^2))),Fb):
> DR:=zip((u,v)->abs(u-v),R1,R2): AR:=zip((u,v)->0.5*(u+v),R1,R2):
> RRb:=zip((u,v)->u/v,DR,AR): MRb:=sum(RRb[i],i=1..Nb)/Nb:
> CRb:=stats[describe,linearcorrelation](R1,R2):
> R1:=map(u->sqrt(u[1]^2+u[2]^2),YZ):
> R2:=map(u->evalf(subs(f=u,sqrt(Yrc^2+Zrc^2))),Fc):
> DR:=zip((u,v)->abs(u-v),R1,R2): AR:=zip((u,v)->0.5*(u+v),R1,R2):
> RRc:=zip((u,v)->u/v,DR,AR): MRc:=sum(RRc[i],i=1..Nc)/Nc:
> CRc:=stats[describe,linearcorrelation](R1,R2):
> Correlations:=[CRa,CRb,CRc];
> Mean_relative_displacement:=[MRa,MRb,MRc];
```

$$Correlations := [0.992, 0.948, 0.084]$$
$$Mean\_relative\_displacements := [0.012, 0.021, 0.021]$$

As can be seen, mean displacements are close to 2%, the highest individual dispacement is lower than 7%, and the crorrelation coefficient between vertexes and corresponding points on the individual ellipses is higher than 90%. These results enable to declare approximation as successfull. Finally, 3D plot showing triaxial ellipsoid with projections of the cross sections can be displayed, see Figue 3.

```
> PCH:=plot3d([A0*cos(f)*cos(l),B0*cos(f)*sin(l),C0*sin(f)],
>  l=-Pi..Pi,f=-Pi/2..Pi/2,style=wireframe,color=grey):
> Sa:=spacecurve(map(u->[cos(Ra0)*(u[1]-Ax0)+sin(Ra0)*(u[2]-Ay0),
>   -sin(Ra0)*(u[1]-Ax0)+cos(Ra0)*(u[2]-Ay0),0],XY),
>   color=black,thickness=3,linestyle=1):
> Sb:=spacecurve(map(u->[cos(Rb0)*(u[1]-Bx0)+sin(Rb0)*(u[2]-Bz0),0,
```

```
>   -sin(Rb0)*(u[1]-Bx0)+cos(Rb0)*(u[2]-Bz0)],XZ),
>   color=black,thickness=3,linestyle=3):
> Sc:=spacecurve(map(u->[0,cos(Rc0)*(u[1]-Cy0)+sin(Rc0)*(u[2]-Cz0),
>   -sin(Rc0)*(u[1]-Cy0)+cos(Rc0)*(u[2]-Cz0)],YZ),
>   color=black,thickness=3,linestyle=4):
> display({PCH,Sa,Sb,Sc},scaling=constrained,axes=boxed);
```

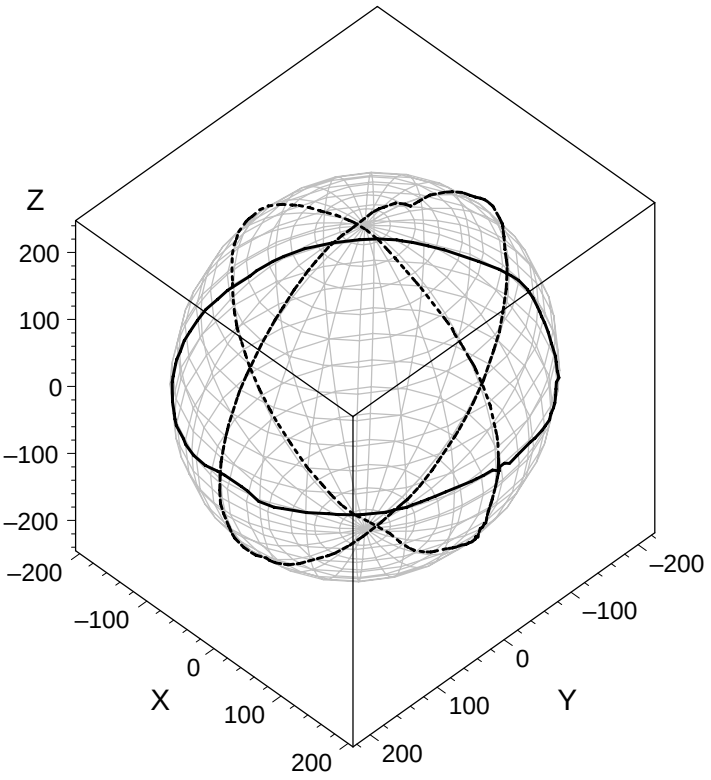

Figure 4: 3D Peach approximation.

7 Conclusions

Regarding above stated high correlation coefficients and relative displacements, and with reference to results shown on Figures 3 and 4, it can be concluded that presented approach provides excellent results. It means that it can be used as a tool for the space approximation of simple three-dimensional objects, such as peach, apricot, apple, cucumber etc.

Acknowledgement. This research has been supported by the Grant Agency of the Czech Academy of Sciences under Contract No. IAA201990701, *Behaviour of selected agricultural products under impact loading.*

References

[1] J. Buchar, Š. Nedomová, L. Severa: *High strain rate behaviour of peaches.* Proceedings of the 2008 SEM XI International Congress and Exposition on Experimental and Applied Mechanics. USA: Society for Experimental Mechanics, Inc., 2008, pp. 163–170. ISBN 0-912053-99-2.

[2] P. Dvořáková, Š. Nedomová, L. Severa, J. Trnka, J. Buchar: *The impulse response method for measuring the overall firmness of fruit.* Book of Contributions of 46th International Scientific Conference "EXPERIMENTAL STRESS ANALYSIS 2008". Ostrava, Czech Republic: VŠB - Technical University of Ostrava, 2008, pp. 43–47. ISBN 978-80-248-1774-3.

[3] L. Severa: *Shape and strength of Red Haven peaches at the different stages of their maturity.* Acta univ. agric. et silvic. Mendel. Brun. 2008, LVI, No. 4 , pp. 169–177. ISSN 1211-8516

[4] S. Bartoň: In: J. Diblík, (Ed.), *Shape determination of an agricultural product* Proceedings of 6. Math. workshop. Brno: FAST VUT Brno, 2007, pp. 1–12. ISBN 80-214-2741-8.

[5] W. Gander, S. Bartoň: *Least Squares Fit with Piecewise Functions.* In: W. Gander, J. Hřebíček (Eds.), Solving problems in Scientific Computing using Maple and Matlab. 4. edition. 1. Heidelberg: Springer, 2004. pp. 433–450. ISBN 3-540-21127-6.

[6] W. Gander, S. Bartoň: *Metod najmensich kvadratov dlja kusocno-nepreryvnych funkcij.* In: W. Gander, J. Hřebíček (Eds.), Resenie zadac v naucmych vycislenijach s primeneniem Maple i Matlab. Minsk: Vassamedia, 2005. pp. 417–442. ISBN 985-6642-06-X.