Finite element method example, elastostatics

We consider the domain $D\subset\mathbb{R}^3$ as a homogeneous elastic isotropic body with density ρ and Lame coefficients λ , μ (in terms of Young modulus E and shear modulus μ we have $\lambda=\frac{\mu E-2\mu^2}{3\mu-E}$)

The Lame coefficients, as known , must be positive.

As external forces, on the body acts the gravitational volumic force, as a vector field on *D,* $b:D\to\mathbb{R}^3$, $b=b(x_1,x_2,x_3)$ and on the boundary surface $S_t\subset\partial D$ acts the superficial force field $\hat{t}:S_t\to\mathbb{R}^3$

On the boundary surface domain S_u with $S_u\cup S_t=\partial D$ we suppose that the displacements field under deformation at equilibrium is given as $u\Big|_{S_u}=\hat{u}:S_u\to\mathbb{R}^3$

The Cauchy stress tensor $T=T(x)$, $x=(x_1,x_2,x_3)$ satisfies as we know (see [1]) :

$T_{ij}=\lambda\delta_{ij}u_{k,k}+\mu(u_{i,j}+u_{j,i})$ or $T_{ij}=C_{ijkl}E_{kl}$ with the Einstein summation convention, where

$$E_{kl}=\frac{1}{2}(u_{k,l}+u_{l,k}) \text{ and } C_{ijkl}=\lambda\delta_{ij}\delta_{kl}+2\mu\delta_{ik}\delta_{jl}$$

We suppose also that the deformations are infinitesimal : $u\in O(\varepsilon)$

In this case, the displacements field (if it is of class C^1 on the closure of the domain) satisfies at equilibrium :

$$T_{ij,j}+b_i=0 \quad (\text{ or } \nabla\cdot T+b=0 \quad) \qquad (1)$$

$t=T_{ij}n_j=\hat{t}$ on S_t where n is the outside normal on $S=\partial D$ $\qquad (2)$

$$u_i=\hat{u}_i \text{ on } S_u \qquad (3) \text{ for } i=1,2,3$$

Consider $u'\in\{v\in C^1(\bar{D},\mathbb{R}^3)|\quad v\big|_{S_u}=0\}=F_0$ and corresponding tensor fields T' and E'

Because the symmetry of (C_{ijkl}) we have $T'\cdot E=T\cdot E'$

For $\bar{u}=u+\epsilon u'$, $\epsilon\in\mathbb{R}$ we define the potential energy

$$\Pi(\bar{u})=\frac{1}{2}\int_D \bar{T}\cdot\bar{E}\,dV-\int_D b\cdot\bar{u}\,dV-\int_{S_t}\hat{t}\cdot\bar{u}\,dA$$

with *dV* , *dA* the volume element in *D* , respective the surface element on *S.*

According to Gauss-Ostrogradski formula we have :

$$\int_D T\cdot E'\,dV=\int_D T'\cdot E\,dV=\int_D T_{ij}u'_{i,j}\,dV=\int_{\partial D}T_{ij}n_j u'_i\,dA-\int_D T_{ij,j}u'_i\,dV=$$

$$=\int_{S_t}\hat{t}\cdot u'\,dA+\int_D b\cdot u'\,dV$$

and therefore it is easy to see that we have :

$$\Pi(\bar{u})-\Pi(u)=\epsilon^2\int_D T'\cdot E'\,dV$$

Because *C* is obviously positive definite as a bilinear form on symmetric tensors it follows

$\Pi(\bar{u})\geq\Pi(u)$ for any $\bar{u}\in\{v\in C^1(D,\mathbb{R}^3)|\quad v\big|_{S_u}=\hat{u}\quad\}=F$

which is the well known minimum potential energy theorem : the equilibrium displacements field minimizes the potential energy on the set of boundary restricted displacements.

Suppose now that one $u\in F$ satisfies $\Pi(u)=\min_{v\in F}\Pi(v)$

It follows $\frac{\partial}{\partial \epsilon}\Pi(u+\epsilon u')\big|_{\epsilon=0}=0$ for any $u'\in F_0$

$$\int_D T\cdot E'\,dV=\int_D b\cdot u'\,dV+\int_{S_t}\hat{t}\cdot u'\,dA$$

$$-\int_D(\nabla\cdot T+b)\cdot u'\,dV+\int_{S_t}(t-\hat{t})\cdot u'\,dA=0 \qquad (4)$$

We can take in (4) *u'* arbitrary with compact support in *D* and so we have :

$\nabla\cdot T+b=0$ on D if $u\in C^2(D,\mathbb{R}^3)$

Taking now , again in (4), *u'* arbitrary in F_0 it follows $t=\hat{t}$ on S_t

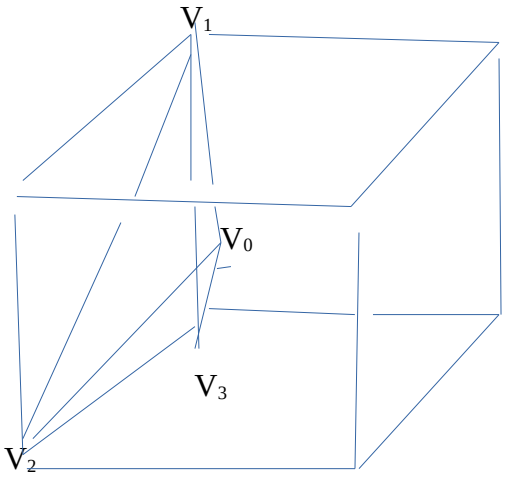
Hence if Π has a minimum on F at u , then u satisfies
the equilibrium conditions (1), (2), (3)

This provides the basis for a finite element variational approach of the elastostatic problem.

We consider that *D* is a ball of radius $R=$re , $D=\{x\in\mathbb{R}^3 | \|x\|<R\}$
which is fixed on its lower half : $S_u=\{x\in\partial D | x_3\le 0\}$, $\hat{u}=0$
endures the gravitational force field : $b=(0,0,-\rho g)$, $\rho g=$vo
a normal pressure pp on the region $S_t=\{x\in S | h1\le x_3\le h2\}$ which produces also by rotating the sphere around the vertical axis a tangential horizontal friction force (friction coefficient 1) on the same region S_t and a concentrated force (fc1, fc2, fc3) with application (pa1, pa2, pa3) on *S*.

The input variables lm and my are related to the Lame coefficients as $\lambda/2$ and $\mu/2$

The sphere is represented in the Python program algorithm by a sphere with centre at (me, me, me) and radius me (me input variable) as a reunion of finite elements (generated by the procedure nelem()), each element a 3-simplex with one first corner point at points with coordinates odd numbers between 0 and 2 x me and the other 3 corner points next laticial integer coordinates points from the same face of the cube surrounding the first point

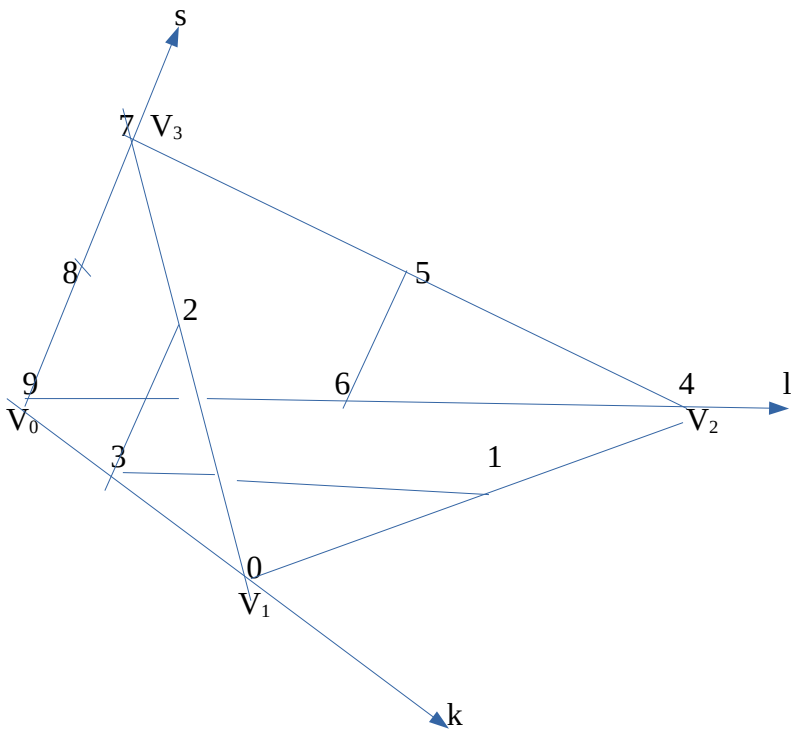


$V_0(2i+1,2j+1,2k+1)$, $V_1(2i+1\pm1,2j+2,2k+2)$, $V_2(2i+1\pm1,2j,2k)$
$V_3(2i+1\pm1,2j+2,2k)$ and $V_3(2i+1\pm1,2j,2k+2)$ with $i,j,k=\overline{0,me-1}$

The list te of simplexes which are in the interior of the ball will be selected such that
$te[m]$ is an array of shape (4,3) of vertexes te[m] = [v[0], v[1], v[2], v[3], v[4]].
For each simplex te[m] = v , class elem computes the transformation elements for mapping the simplex on the unit simplex [(0,0,0), (1,0,0), (0,1,0), (0,0,1)] such that self.mat represents the linear transformation matrix coefficients, self.det represents the volume element determinant.
The nodes in the unit simplex are indexed by (k,l,s) with k,l,s in {0,1,2} such that k + l + s <= 2 with ind(m, k, l, s) the procedure what assigns to each node from the element m in element list te, at coordinates (k/2, l/2, s/2) in the unit simplex a calculated index n. Also, the procedure lst() returns the list c of tuples such that c[n] = (m, k, l,s).

When numbering the nodes in the m-th simplex as it is shown in the figure below the expression for ind(m, k, l, s) is given by

$$10m+\sum_{i=0}^{2-k-1}\sum_{j=1}^{i+1} j+\sum_{i=0}^{2-k-l-1}(i+1)+2-k-l-s$$



The displacements field on the simplex te[m] = v will be approximated by degree 2 polynomials in the variables (ξ,η,ζ) = (xi, et, ze) , which are the corresponding coordinates in the unit simplex of the current point x from v : [xi, et, ze] = elem(v).itr(x) with self.itr(x) being the class elem procedure which computes the point corresponding to x in the unit simplex. We will have :

$u_i(x)=\sum \chi_m \alpha^j_{mp} f_{mp}$, where χ_m is the characteristic function of te[m] , α^j_{mp} are variational parameters of the problem and f_{mp} are the respective polynomials such that if for the simplex te[m] we have the 10 nodes P^m_p , $p=\overline{0,9}$ then $f_{mi}(P^m_j)=\delta_{ij}$

Therefore $u^i(P^m_p)=\alpha^j_{mp}$ and

$f_{mp}(x)=\sum_{q=0}^{9} A^m_{pq}\xi^k\eta^l\zeta^s$ where in the sum we have the correspondent indexing (0, k, l, s) = c[q]

The coefficients A^m_{pq} are computed as elem(v).cf[p, q]

The class elem procedure self.fc(n, x) computes the value of $f_{mn}(x)$ when te[m] = v

The procedure ing(k, l, s) computes the required values of $\int_T \xi^k\eta^l\zeta^s d\xi d\eta d\zeta$ where T is the unit simplex

$$T=\{(\xi,\eta,\zeta)\in[0,1]^3 | \xi+\eta+\zeta\le 1\}$$

The procedure lpc(n) computes the node P^m_p which corresponds to index n : (m, k, l, s) = c[n] and p = ind(0, k, l, s)

The procedure ta(n) computes the list of indexes m for which the corresponding element te[m] contains the node n.

The procedure lsf() computes psf, the set of indexes of the nodes which correspond to the boundary of the domain.

To determine the coefficients of variational parameters which appear in the external superficial forces in the potential energy expression we consider a triangulation of S_t . The procedure

coeftri(vf) computes the respective coefficient associated with triangle vf.
To determine the variational parameters we must seek for minimization of the potential energy in terms of (α^i_{mp}) so that the restraints given by the boundary conditions for u on S_u and by the coincidence of nodes at the adjacency of elements are satisfied.
For this purpose we add Lagrange multipliers for each independent coincidence relation $\lambda^{m'p'}_{mpi}$ for the relation $\alpha^i_{mp}=\alpha^i_{m'p'}$ and $\mu_{m''p''i}$ for $\alpha^i_{m''p''}=0$ as additional variational parameters and to the potential energy expression we add the Lagrange multipliers terms $\lambda^{m'p'}_{mpi}(\alpha^i_{mp}-\alpha^i_{m'p'})$ and respective $\mu_{m''p''i}\alpha^i_{m''p''}$
Differentiating the obtained extended expression of potential energy we obtain the respective coefficients of the matrix of the system to solve in variational parameters. To set these coefficients and expand the system matrix, procedure lag() is defined.
The system of linear equations to solve in variational parameters will be $\bar{K}\bar{x}=\bar{b}$ where $\bar{K}$ is the basic matrix *K* expanded with the Lagrange multipliers coefficients computed by lag().
With be, the length of list te, the matrix *K* is a shape (30be, 30be) matrix which contains the $K_{mph\ mqk}$ coefficients of α^k_{mq} obtained by differentiating the approximated deformation energy expression $\frac{1}{2}\int_D T\cdot E\,dV$ with respect to variational parameter α^h_{mp} (we have the integral on *D* as a sum of integrals on each simplex element) the other coefficients of *K* being obviously zero.
The expression of $K_{mph\ mqk}$ comes from the deformation energy expression for the m-th element :

$$W_m=\frac{1}{2}\int_{T_m} C_{ijkl}u_{h,j}u_{k,l}dV=\frac{1}{2}\int_{T_m} C_{ijkl}\alpha^j_{mp}\alpha^k_{mq}f_{mp,j}f_{mq,l}dV$$

(with the Einstein summation convention)
Differentiating with respect to α^h_{mp} and considering the symmetry of *C* we obtain :

$$K_{mph\ mqk}=\int_{T_m}(\lambda f_{mp,h}f_{mq,k}+2\mu\,\delta_{hk}f_{mp,j}f_{mq,j})dV$$

For $(\xi^i)=(\xi,\eta,\zeta)$ we can express $f_{mr,s}=\frac{\partial f_{mr}}{\partial \xi^i}\frac{\partial \xi^i}{\partial x_s}$ where the coefficients $\frac{\partial \xi^i}{\partial x_s}$ are determined by the class elem attribute el.mat[i, s] with el = elem(te[m]) the corresponding m-th element linear mapping to the unit simplex.
Now, procedure cemat(m, p, q, h, k) computes integrals of products $f_{mp,h}f_{mq,k}$ integrating on the unit simplex using the volume element transformation el.det and procedure kat() completes the computing of the matrix $\bar{K}$ stored as ka.
The procedure col() computes the dependent variables column $\bar{b}$ of the system storing it as br by using the external forces inputs and the coeftri() calculations.
Also, the procedure col() returns xs, ys, zs the arrays of coordinates of vertices of elements which are in the superficial external forces application area $(h1\le x_3\le h2)$
That area will be plotted in the finite elements figure as triangulated surface in light color.
The “Import pdf” file contains the Python code “atest” with the above mentioned procedures and the read data command lines and the “Finite element pdf” file contains the main program Python code for computing the variational parameters array and plotting the chosen displaced oblique surface after computing the corresponding displacements.
Note that if we take the domain $B(me)=\{x\in\mathbb{R}^3 \| x-\overline{me}\|\le me\}$ with $\overline{me}=(me,me,me)$ we measure the distances in the unit re/me and so we must write the potential energy expression in the form

$$\frac{1}{2}\int_{B(\mathrm{me})} T\cdot E\,dV\left(\frac{\mathrm{re}}{\mathrm{me}}\right)^3-\int_{B(\mathrm{me})} b\cdot u\,dV\left(\frac{\mathrm{re}}{\mathrm{me}}\right)^3-\int_{S_t(\mathrm{me})}\hat{t}\cdot u\,dA\left(\frac{\mathrm{re}}{\mathrm{me}}\right)^2-\mathrm{fc}\cdot u(\mathrm{pa})\left(\frac{\mathrm{re}}{\mathrm{me}}\right)$$

and we must take into consideration the factors in re/me when computing the coefficients.

For the examples we took

$\lambda = 2000\,N/cm^2$, $\mu = 2000\,N/cm^2$, $\rho g = 0.023\,N/cm^3$ (silicone rubber)

what gives the proportional parameters my = 1 , lm = 1 , vo = 0.000023

Also we took me = 3 , re = 10 cm , h1 = 11.5 cm , h2 = 17 cm , pa = (5.75, 5.75, 18) cm

For the first example : pp = 0.06 (corresponding to 60 N/cm^2) , fc = (0.02, 0.02, -0.02) with 0.02 corresponding to 20 N

For the second example pp = 0.1 (corresponding to 100 N/cm^2) , fc = (0, 0, 0).

For each example we have the respective first and second “Displacements in oblique plane ” figures.

For determining the variational parameters array extended with the Lagrange multipliers we used the conjugate residual gradient method to solve the system of equations represented by ka x = br , where x , br are taken as column vectors , br computed by procedure col() and ka is the matrix computed by procedure kat() . (Note that ka is symmetric.)

Conjugate residual gradient method

Consider first a symmetric positive definite non-singular matrix *A* of shape (n, n) and a n-dimensional column vector *b*.

It is clear that solving for column n-dimensional vector *x* the system *Ax = b* is the same as minimizing for $x \in \mathbb{R}^n$ the convex functional $f(x) = \frac{1}{2} x^T A x - x^T b$

where exponential ^{T} denotes the transposed matrix.

Taking the recurring sequences :

(p_k) , (x_k) , (α_k) , $(r_k) k = 0, 1, 2, 3, ...$

$$r_0 = b - A x_0 \qquad (1)$$

$$p_0 = r_0 \qquad (2)$$

$$p_k = r_k - \sum_{i<k} \frac{p_i^T A r_k}{p_i^T A p_i} p_i \qquad (3)$$

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k} \qquad (4)$$

$$x_{k+1} = x_k + \alpha_k p_k \qquad (5)$$

$$r_{k+1} = r_k - \alpha_k A p_k \qquad (6)$$

From (1), (5) and (6) follows by induction $r_k = b - A x_k$

From (3) follows $p_k^T A p_i = 0$ for $i < k$

From (6) we have :

$$r_{k+1} = r_l - \sum_{l \le i \le k} \alpha_i A p_i \quad \text{for } l \le k$$

From (3) and (7) follows that for $j > l$ we have $r_l^T A p_j = 0$

Therefore from (8), (3), (7) and (4) follows :

$$r_l^T r_{k+1} = r_l^T r_l - \alpha_l r_l^T A p_l = r_l^T r_l - \alpha_l p_l^T A p_l = 0$$

and so we have proven that $r_i^T r_j = 0$ for $i \neq j$

From (6) we have now, as long as $p_i \neq 0$ and $\alpha_i \neq 0$

$p_i^T A r_{k+1} = r_{k+1}^T A p_i = -\frac{1}{\alpha_i}(r_{k+1}^T r_{i+1} - r_{k+1}^T r_i)$ and so

$$p_i^T A r_{k+1} = -\frac{r_{k+1}^T r_{i+1}}{\alpha_i} \quad \text{for} \quad i \leq k$$

Considering (3) for $k+1$, because we have already proven that (r_i) is an orthogonal sequence, we will have :

$$p_{k+1} = r_{k+1} + \beta_k p_k \quad \text{with} \quad \beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \qquad (9)$$

Taking (9), which involves only the most recent values of the recurring sequences ,instead of (3) in the recurrence relations list (1) – (6) we can define the algorithm to determine the values of x_k until $p_k = 0$ or $\alpha_k = 0$

From (9) we have also :

$$p_{k+1} = r_{k+1} + r_{k+1}^T r_{k+1} \sum_{i=0}^{k} \frac{1}{r_i^T r_i} r_i \quad (10) \quad \text{and so}$$

$$p_{k+1}^T r_{k+1} = r_{k+1}^T r_{k+1}$$

Hence as long as $r_k \neq 0$ we have also $p_k \neq 0$ and $\alpha_k \neq 0$

and the recurrence relations (1)-(6), (9) are valid until r_{k+1} becomes 0.

If $r_k = 0$ we have $A x_k = b$ and so the algorithm has led to the solution of the system

Relation (5) leads to $x_{l+1} = x_0 + \alpha_0 p_0 + \ldots + \alpha_l p_l$ for *l* increasing from 0 to *k* until $r_{k+1} = 0$

It is easy to see that

$$\frac{\partial f}{\partial \alpha_i}(x_{l+1}) = p_i^T (A x_{l+1} - b) = p_i^T r_{l+1} = 0$$

(because of (10) and $i < l + 1$)

Therefore , if r_l has not become 0 for $l = \overline{0, n-1}$

$\{p_0, p_1, \ldots p_{n-1}\}$ is an A-orthogonal set of non-zero vectors and form a basis in $\mathbb{R}^n$.

Then from (11) follows $\frac{\partial f}{\partial \alpha_i}(x_n) = 0$ for $i = \overline{0, n-1}$ and because f is convex and differentiable x_n minimizes the functional *f* and is the solution of the system $A x = b$ and $r_n = 0$

If *A* is not symmetric and positive definite (but is non-singular) we consider the conjugate gradient

algorithm defined by the recurrence (1) – (6), (9) for the system $A^T A x = A^T b$ $A^T A$

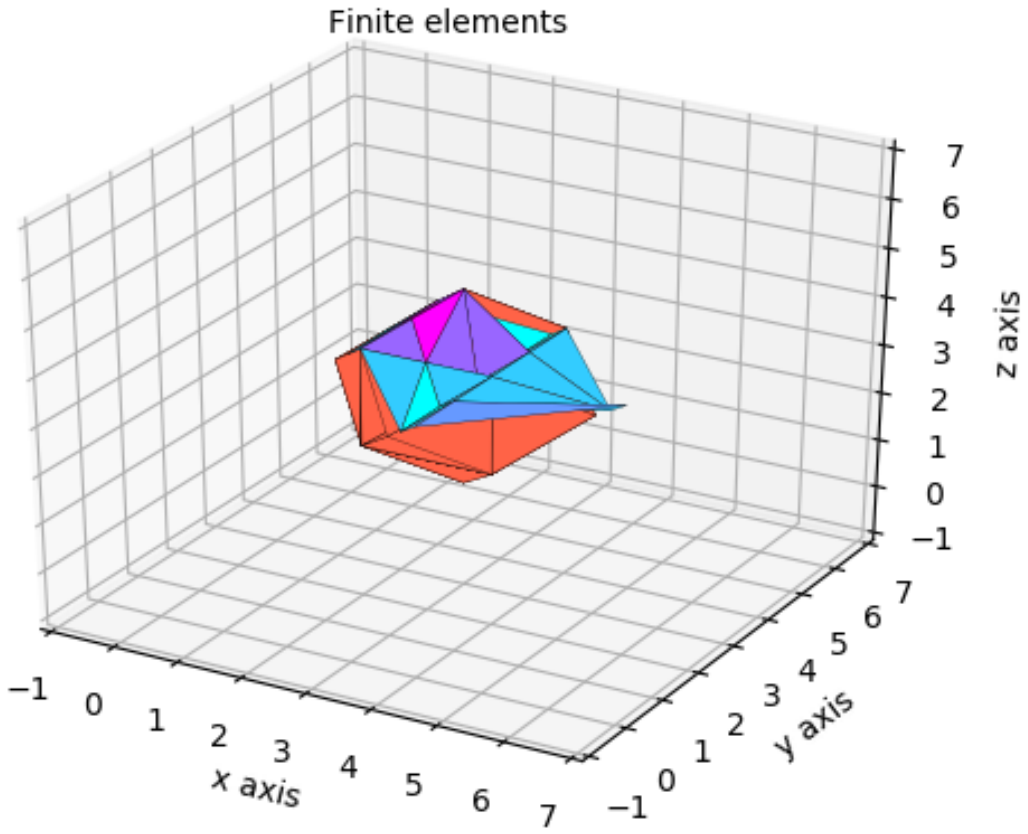
being symmetric and positive definite and so we have for determining *x* the so called conjugate residual gradient method.

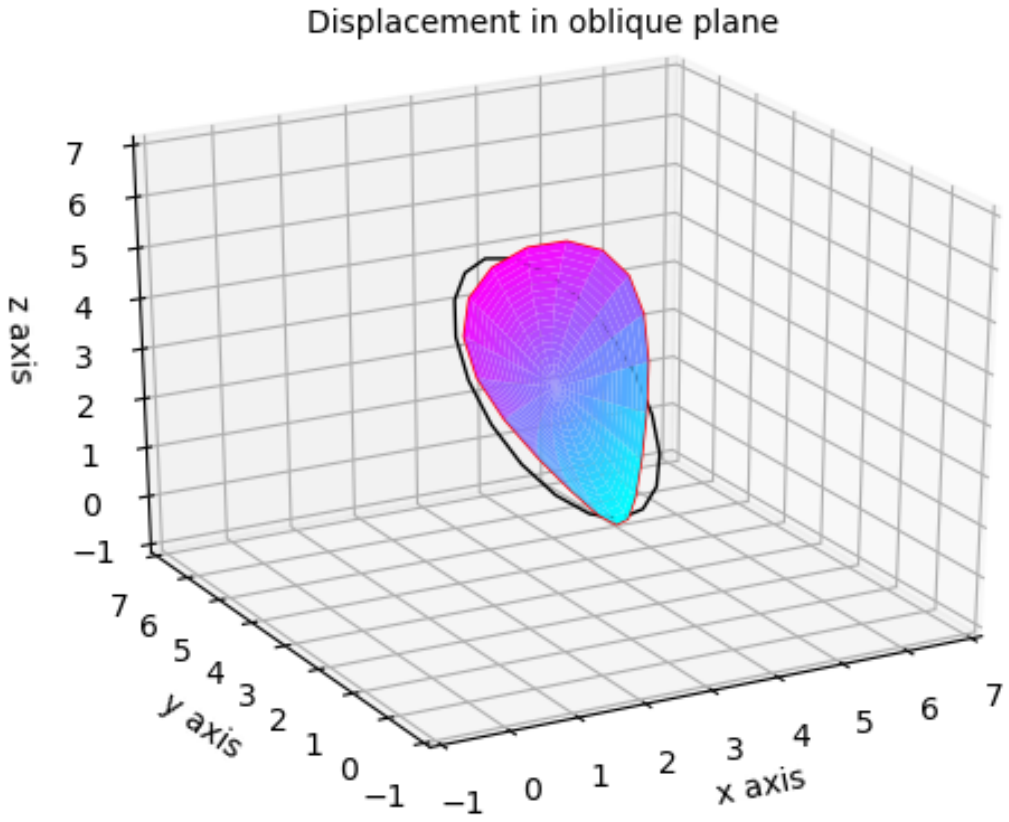
The conjugate gradient method (respective the conjugate residual gradient method), viewed as a direct method, in the absence of round-off error produces the exact solution after a finite number of iterations, which is not larger than the size of the matrix. However the method is unstable with respect to even small perturbations, but as an iterative method improves approximations to the exact solution, and may reach the required tolerance after a relatively small (compared to the problem size) number of iterations. The improvements speed is determined by a condition number of the system matrix.

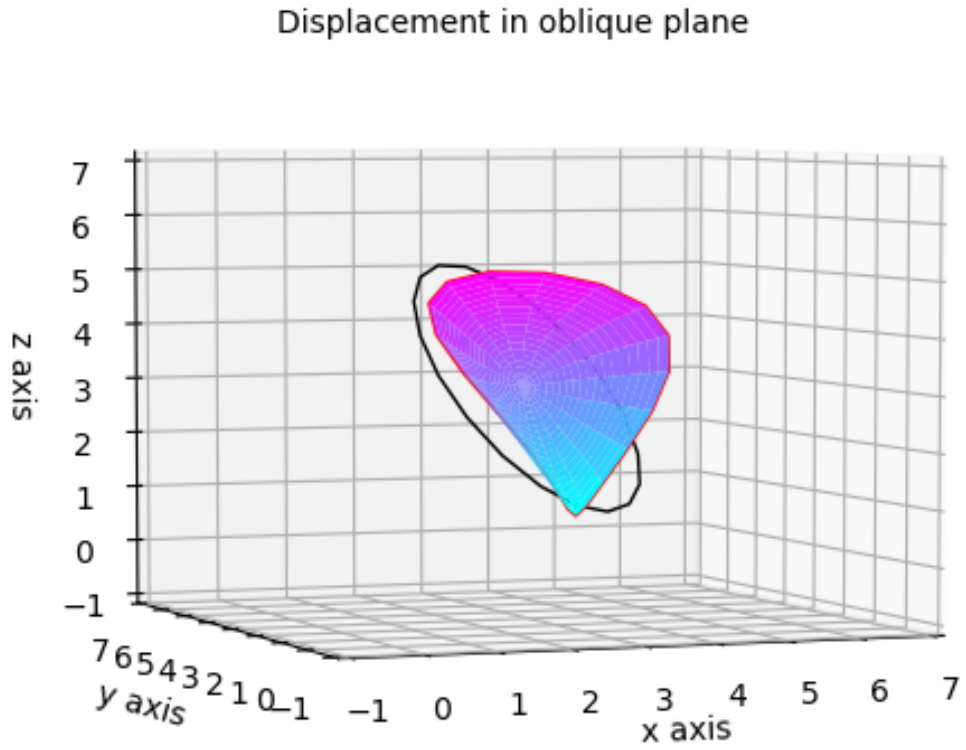
After computing the variational parameters, calling procedure che(x) for determining the finite element which a point x belongs to, the main program computes the displacements arrays in an oblique 30° inclined diametrical plane and plots the displaced surface.
The plots for the first and second example are respective the “Displacements in oblique plane” figures at the end of the file.

REFERENCES

[1] Lazăr Dragoș , Principiile Mecanicii Mediilor Continue, Editura Tehnică, București
[2] Victor Florin Poteraşu, Nicu Mihalache, Dumitru Mangeron, Metode Numerice în Elasticitate şi Plasticitate, Editura Academiei Române, Bucureşti 1993
[3] George Vraciu, Aurel Popa, Metode Numerice cu Aplicații în Tehnica de Calcul, Scrisul Românesc, Craiova, 1982
[4] Conjugate gradient method - Wikipedia


Finite elements
x axis
y axis
z axis
−1
0
1
2
3
4
5
6
7


Displacement in oblique plane
z axis
y axis
x axis
7
6
5
4
3
2
1
0
−1


Displacement in oblique plane
z axis
7
6
5
4
3
2
1
0
−1
y axis
7 6 5 4 3 2 1 0 −1
x axis
−1 0 1 2 3 4 5 6 7