ZÁPADOČESKÁ UNIVERZITA V PLZNI

www.KMA.zcu.cz SINCE 1954

Solving Poisson equation over planar NURBS domains with isogeometric analysis

Bohumír Bastl[1]
Email: bastl@kma.zcu.cz

(joint research with P. Anděl)

[1] Department of Mathematics, Faculty of Applied Sciences
University of West Bohemia in Plzeň, Czech Republic

Spliny a IsoGeometrická Analýza 2011

Praha, February 9, 2011

Outline

Isogeometric analysis

- method for the analysis of problems governed by partial differential equations, proposed by T.J.R. Hughes et al. in 2005
- many features common with the finite element method
- inspired by CAGD – primary goal is to be geometrically exact independently of the discretization
- typically in engineering practice, design is done in CAD systems and meshes, needed for the finite element analysis, are generated from CAD data
- in some cases, mesh can be generated automatically, but in most cases, it is done semi-automatically – it usually needs human interaction
- it is estimated that about 80% of overall analysis time is spent in mesh generation in automotive, aerospace and ship industries
- each design change requires generation of new meshes which takes a lot of time
- further, inaccuracies in geometric representation can lead to problems with precision of obtained solutions, e.g. thin shell analysis is extremely sensitive to geometric imperfections or problems in fluid mechanics

Isogeometric analysis

- method for the analysis of problems governed by partial differential equations, proposed by T.J.R. Hughes et al. in 2005
- many features common with the finite element method
- inspired by CAGD – primary goal is to be geometrically exact independently of the discretization
- typically in engineering practice, design is done in CAD systems and meshes, needed for the finite element analysis, are generated from CAD data
- in some cases, mesh can be generated automatically, but in most cases, it is done semi-automatically – it usually needs human interaction
- it is estimated that about 80% of overall analysis time is spent in mesh generation in automotive, aerospace and ship industries
- each design change requires generation of new meshes which takes a lot of time
- further, inaccuracies in geometric representation can lead to problems with precision of obtained solutions, e.g. thin shell analysis is extremely sensitive to geometric imperfections or problems in fluid mechanics

Isogeometric analysis

- method for the analysis of problems governed by partial differential equations, proposed by T.J.R. Hughes et al. in 2005
- many features common with the finite element method
- inspired by CAGD – primary goal is to be geometrically exact independently of the discretization
- typically in engineering practice, design is done in CAD systems and meshes, needed for the finite element analysis, are generated from CAD data
- in some cases, mesh can be generated automatically, but in most cases, it is done semi-automatically – it usually needs human interaction
- it is estimated that about 80% of overall analysis time is spent in mesh generation in automotive, aerospace and ship industries
- each design change requires generation of new meshes which takes a lot of time
- further, inaccuracies in geometric representation can lead to problems with precision of obtained solutions, e.g. thin shell analysis is extremely sensitive to geometric imperfections or problems in fluid mechanics

Isogeometric analysis

- method for the analysis of problems governed by partial differential equations, proposed by T.J.R. Hughes et al. in 2005
- many features common with the finite element method
- inspired by CAGD – primary goal is to be geometrically exact independently of the discretization
- typically in engineering practice, design is done in CAD systems and meshes, needed for the finite element analysis, are generated from CAD data
- in some cases, mesh can be generated automatically, but in most cases, it is done semi-automatically – it usually needs human interaction
- it is estimated that about 80% of overall analysis time is spent in mesh generation in automotive, aerospace and ship industries
- each design change requires generation of new meshes which takes a lot of time
- further, inaccuracies in geometric representation can lead to problems with precision of obtained solutions, e.g. thin shell analysis is extremely sensitive to geometric imperfections or problems in fluid mechanics

Isogeometric analysis

- method for the analysis of problems governed by partial differential equations, proposed by T.J.R. Hughes et al. in 2005
- many features common with the finite element method
- inspired by CAGD – primary goal is to be geometrically exact independently of the discretization
- typically in engineering practice, design is done in CAD systems and meshes, needed for the finite element analysis, are generated from CAD data
- in some cases, mesh can be generated automatically, but in most cases, it is done semi-automatically – it usually needs human interaction
- it is estimated that about 80% of overall analysis time is spent in mesh generation in automotive, aerospace and ship industries
- each design change requires generation of new meshes which takes a lot of time
- further, inaccuracies in geometric representation can lead to problems with precision of obtained solutions, e.g. thin shell analysis is extremely sensitive to geometric imperfections or problems in fluid mechanics

Isogeometric analysis

- method for the analysis of problems governed by partial differential equations, proposed by T.J.R. Hughes et al. in 2005
- many features common with the finite element method
- inspired by CAGD – primary goal is to be geometrically exact independently of the discretization
- typically in engineering practice, design is done in CAD systems and meshes, needed for the finite element analysis, are generated from CAD data
- in some cases, mesh can be generated automatically, but in most cases, it is done semi-automatically – it usually needs human interaction
- it is estimated that about 80% of overall analysis time is spent in mesh generation in automotive, aerospace and ship industries
- each design change requires generation of new meshes which takes a lot of time
- further, inaccuracies in geometric representation can lead to problems with precision of obtained solutions, e.g. thin shell analysis is extremely sensitive to geometric imperfections or problems in fluid mechanics

Isogeometric analysis

- the creation of meshes is costly, time-consuming and involve inaccuracies into computations
- the idea behind isogeometric analysis is to use directly CAD data for the analysis
- since the most common standard in current CAD systems are NURBS objects, it is reasonable to use NURBS representation of object stored in CAD system also for analysis
- then all objects are represented exactly (as in CAD system) and we do not need to create any other mesh – the mesh of the so-called “NURBS elements” is acquired directly from CAD representation
- further refinement of the mesh (knot insertion – h-refinement) or increasing the order of basis functions (order elevation – p-refinement) are very simple, also the suitable combination of these methods called k-refinement seems to be very efficient and robust
- the word “isogeometric” reflects the fact that isoparametric philosophy is used – the solution space for dependent variables is represented by the same basis functions which represent the geometry

Isogeometric analysis

- the creation of meshes is costly, time-consuming and involve inaccuracies into computations
- the idea behind isogeometric analysis is to use directly CAD data for the analysis
- since the most common standard in current CAD systems are NURBS objects, it is reasonable to use NURBS representation of object stored in CAD system also for analysis
- then all objects are represented exactly (as in CAD system) and we do not need to create any other mesh – the mesh of the so-called “NURBS elements” is acquired directly from CAD representation
- further refinement of the mesh (knot insertion – h-refinement) or increasing the order of basis functions (order elevation – p-refinement) are very simple, also the suitable combination of these methods called k-refinement seems to be very efficient and robust
- the word “isogeometric” reflects the fact that isoparametric philosophy is used – the solution space for dependent variables is represented by the same basis functions which represent the geometry

Isogeometric analysis

- the creation of meshes is costly, time-consuming and involve inaccuracies into computations
- the idea behind isogeometric analysis is to use directly CAD data for the analysis
- since the most common standard in current CAD systems are NURBS objects, it is reasonable to use NURBS representation of object stored in CAD system also for analysis
- then all objects are represented exactly (as in CAD system) and we do not need to create any other mesh – the mesh of the so-called "NURBS elements" is acquired directly from CAD representation
- further refinement of the mesh (knot insertion – h-refinement) or increasing the order of basis functions (order elevation – p-refinement) are very simple, also the suitable combination of these methods called k-refinement seems to be very efficient and robust
- the word "isogeometric" reflects the fact that isoparametric philosophy is used – the solution space for dependent variables is represented by the same basis functions which represent the geometry

Isogeometric analysis

- the creation of meshes is costly, time-consuming and involve inaccuracies into computations
- the idea behind isogeometric analysis is to use directly CAD data for the analysis
- since the most common standard in current CAD systems are NURBS objects, it is reasonable to use NURBS representation of object stored in CAD system also for analysis
- then all objects are represented exactly (as in CAD system) and we do not need to create any other mesh – the mesh of the so-called “NURBS elements” is acquired directly from CAD representation
- further refinement of the mesh (knot insertion – h-refinement) or increasing the order of basis functions (order elevation – p-refinement) are very simple, also the suitable combination of these methods called k-refinement seems to be very efficient and robust
- the word “isogeometric” reflects the fact that isoparametric philosophy is used – the solution space for dependent variables is represented by the same basis functions which represent the geometry

Isogeometric analysis

- the creation of meshes is costly, time-consuming and involve inaccuracies into computations
- the idea behind isogeometric analysis is to use directly CAD data for the analysis
- since the most common standard in current CAD systems are NURBS objects, it is reasonable to use NURBS representation of object stored in CAD system also for analysis
- then all objects are represented exactly (as in CAD system) and we do not need to create any other mesh – the mesh of the so-called “NURBS elements” is acquired directly from CAD representation
- further refinement of the mesh (knot insertion – h-refinement) or increasing the order of basis functions (order elevation – p-refinement) are very simple, also the suitable combination of these methods called k-refinement seems to be very efficient and robust
- the word “isogeometric” reflects the fact that isoparametric philosophy is used – the solution space for dependent variables is represented by the same basis functions which represent the geometry

Isogeometric analysis

- the creation of meshes is costly, time-consuming and involve inaccuracies into computations
- the idea behind isogeometric analysis is to use directly CAD data for the analysis
- since the most common standard in current CAD systems are NURBS objects, it is reasonable to use NURBS representation of object stored in CAD system also for analysis
- then all objects are represented exactly (as in CAD system) and we do not need to create any other mesh – the mesh of the so-called “NURBS elements” is acquired directly from CAD representation
- further refinement of the mesh (knot insertion – h-refinement) or increasing the order of basis functions (order elevation – p-refinement) are very simple, also the suitable combination of these methods called k-refinement seems to be very efficient and robust
- the word “isogeometric” reflects the fact that isoparametric philosophy is used – the solution space for dependent variables is represented by the same basis functions which represent the geometry

NURBS surfaces

- NURBS surface of degree p, q is determined by a control net $\mathbf{P}$ (of control points $P_{i,j}$, $i = 0, \ldots, n$, $j = 0, \ldots, m$), weights $w_{i,j}$ of these control points and two knot vectors $U = (u_0, \ldots, u_{n+p+1})$, $V = (v_0, \ldots, v_{m+q+1})$ and is given by a parametrization

$$S(u,v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} P_{i,j} N_{i,p}(u) M_{j,q}(v)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} N_{i,p}(u) M_{j,q}(v)} = \sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} P_{i,j} R_{i,j}(u,v)$$

- B-spline basis functions $N_{i,p}(u)$ and $M_{j,q}(v)$ are determined by knot vectors U and V and degrees p and q, respectively, by a formula

$$N_{i,0}(t) = \begin{cases} 1 & t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p}(t)$$

- knot vector is a non-decreasing sequence of real numbers which determines the distribution of a parameter on the corresponding curve/surface and can be uniform or non-uniform and periodic or non-periodic

NURBS surfaces

- NURBS surface of degree p, q is determined by a control net $\mathbf{P}$ (of control points $P_{i,j}$, $i = 0, \ldots, n$, $j = 0, \ldots, m$), weights $w_{i,j}$ of these control points and two knot vectors $U = (u_0, \ldots, u_{n+p+1})$, $V = (v_0, \ldots, v_{m+q+1})$ and is given by a parametrization

$$S(u,v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} P_{i,j} N_{i,p}(u) M_{j,q}(v)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} N_{i,p}(u) M_{j,q}(v)} = \sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} P_{i,j} R_{i,j}(u,v)$$

- B-spline basis functions $N_{i,p}(u)$ and $M_{j,q}(v)$ are determined by knot vectors U and V and degrees p and q, respectively, by a formula

$$\begin{aligned} N_{i,0}(t) &= \begin{cases} 1 & t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(t) &= \frac{t - t_i}{t_{i+p} - t_i} N_{i,p}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p}(t) \end{aligned}$$

- knot vector is a non-decreasing sequence of real numbers which determines the distribution of a parameter on the corresponding curve/surface and can be uniform or non-uniform and periodic or non-periodic

NURBS surfaces

- NURBS surface of degree p, q is determined by a control net $\mathbf{P}$ (of control points $P_{i,j}$, $i = 0, \ldots, n$, $j = 0, \ldots, m$), weights $w_{i,j}$ of these control points and two knot vectors $U = (u_0, \ldots, u_{n+p+1})$, $V = (v_0, \ldots, v_{m+q+1})$ and is given by a parametrization

$$S(u,v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} P_{i,j} N_{i,p}(u) M_{j,q}(v)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} N_{i,p}(u) M_{j,q}(v)} = \sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} P_{i,j} R_{i,j}(u,v)$$

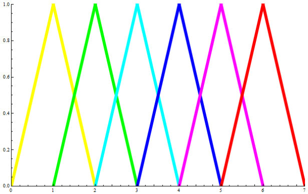
- B-spline basis functions $N_{i,p}(u)$ and $M_{j,q}(v)$ are determined by knot vectors U and V and degrees p and q, respectively, by a formula

$$\begin{aligned} N_{i,0}(t) &= \begin{cases} 1 & t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(t) &= \frac{t - t_i}{t_{i+p} - t_i} N_{i,p}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p}(t) \end{aligned}$$

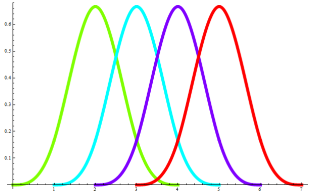
- knot vector is a non-decreasing sequence of real numbers which determines the distribution of a parameter on the corresponding curve/surface and can be uniform or non-uniform and periodic or non-periodic

NURBS surfaces – B-spline basis functions

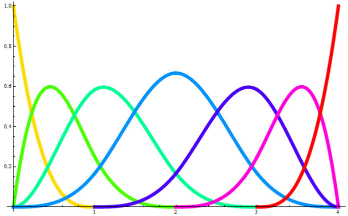
$T = (0, 1, 2, \ldots, 7),\ p = 1$



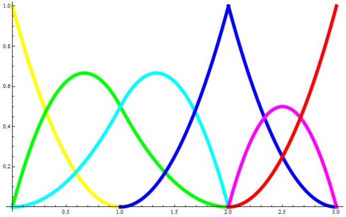
$T = (0, 1, 2, \ldots, 7),\ p = 3$



$T = (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$

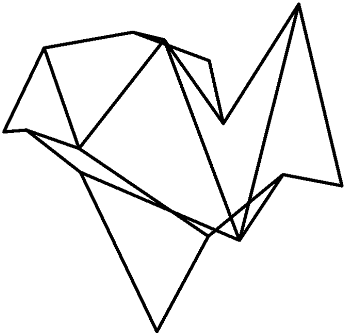


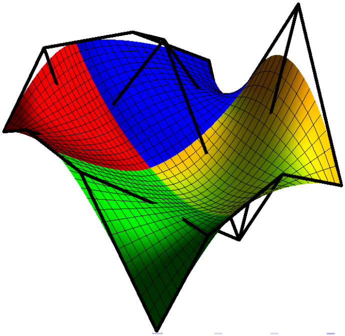
$T = (0, 0, 0, 1, 2, 2, 3, 3, 3)$



NURBS surfaces – properties

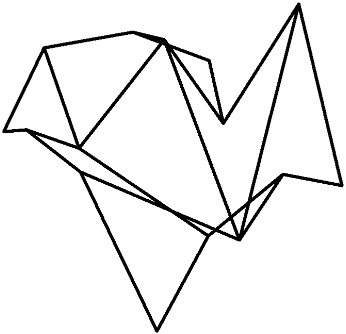
- B-spline basis functions of degree p are C^{p-1}-continuous in general
- knot repeated k times in the knot vector decreases the continuity of B-spline basis functions by $k-1$
- support of B-spline basis functions is local – it is nonzero only on the interval $[t_i, t_{i+p+1}]$ in the parameter space
- each B-spline basis function is non-negative, i.e., $N_{i,p}(t) \geq 0, \forall t$

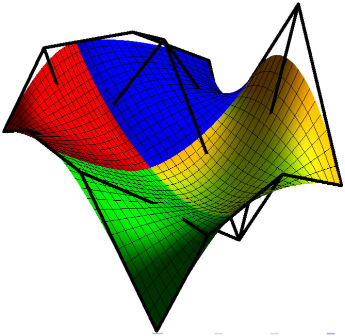




NURBS surfaces – properties

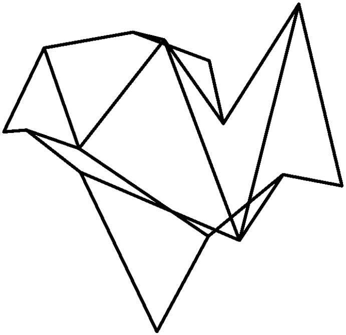
- B-spline basis functions of degree p are C^{p-1}-continuous in general
- knot repeated k times in the knot vector decreases the continuity of B-spline basis functions by $k-1$
- support of B-spline basis functions is local – it is nonzero only on the interval $[t_i, t_{i+p+1}]$ in the parameter space
- each B-spline basis function is non-negative, i.e., $N_{i,p}(t) \geq 0, \forall t$

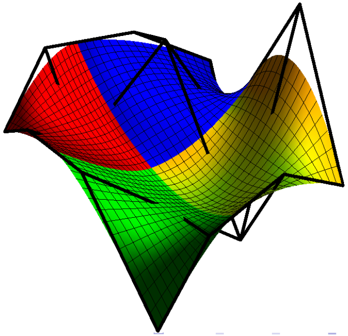




NURBS surfaces – properties

- B-spline basis functions of degree p are C^{p-1}-continuous in general
- knot repeated k times in the knot vector decreases the continuity of B-spline basis functions by $k-1$
- support of B-spline basis functions is local – it is nonzero only on the interval $[t_i, t_{i+p+1}]$ in the parameter space
- each B-spline basis function is non-negative, i.e., $N_{i,p}(t) \geq 0, \forall t$





Planar NURBS domains

- one of the key geometric problems in isogeometric analysis is a description of NURBS domains/volumes from their boundary data (contained in CAD model)
- recently, this problem was studied e.g. in Xu et al. (2010) for planar domains, where a sufficient condition for injectivity of planar B-spline parameterization is proposed and also an influence of different parameterizations on computation result is partially studied
- further, Manh et al. (201x) proposes two linear methods for extension of a B-spline parameterization from the boundary of a domain onto its interior during shape optimization of vibrating membranes:
  - the first method is inspired by ideas coming from linear elasticity and is based on a spring model of the mesh (works well for convex domains)
  - the other method is based on a "quasi-conformal deformation" – parameterization of an initial reference shape is found by solving optimization problems, then inner control points are generated by quasi-conformal deforming the reference shape into the resulting configuration
- in general, the resulting domain parameterization should satisfy $\det(J) > 0$ everywhere in the parameter domain

Planar NURBS domains

- one of the key geometric problems in isogeometric analysis is a description of NURBS domains/volumes from their boundary data (contained in CAD model)
- recently, this problem was studied e.g. in Xu et al. (2010) for planar domains, where a sufficient condition for injectivity of planar B-spline parameterization is proposed and also an influence of different parameterizations on computation result is partially studied
- further, Manh et al. (201x) proposes two linear methods for extension of a B-spline parameterization from the boundary of a domain onto its interior during shape optimization of vibrating membranes:
  - the first method is inspired by ideas coming from linear elasticity and is based on a spring model of the mesh (works well for convex domains)
  - the other method is based on a "quasi-conformal deformation" – parameterization of an initial reference shape is found by solving optimization problems, then inner control points are generated by quasi-conformal deforming the reference shape into the resulting configuration
- in general, the resulting domain parameterization should satisfy $\det(J) > 0$ everywhere in the parameter domain

Planar NURBS domains

- one of the key geometric problems in isogeometric analysis is a description of NURBS domains/volumes from their boundary data (contained in CAD model)
- recently, this problem was studied e.g. in Xu et al. (2010) for planar domains, where a sufficient condition for injectivity of planar B-spline parameterization is proposed and also an influence of different parameterizations on computation result is partially studied
- further, Manh et al. (201x) proposes two linear methods for extension of a B-spline parameterization from the boundary of a domain onto its interior during shape optimization of vibrating membranes:
  - the first method is inspired by ideas coming from linear elasticity and is based on a spring model of the mesh (works well for convex domains)
  - the other method is based on a "quasi-conformal deformation" – parameterization of an initial reference shape is found by solving optimization problems, then inner control points are generated by quasi-conformal deforming the reference shape into the resulting configuration
- in general, the resulting domain parameterization should satisfy $\det(J) > 0$ everywhere in the parameter domain

Planar NURBS domains

- one of the key geometric problems in isogeometric analysis is a description of NURBS domains/volumes from their boundary data (contained in CAD model)
- recently, this problem was studied e.g. in Xu et al. (2010) for planar domains, where a sufficient condition for injectivity of planar B-spline parameterization is proposed and also an influence of different parameterizations on computation result is partially studied
- further, Manh et al. (201x) proposes two linear methods for extension of a B-spline parameterization from the boundary of a domain onto its interior during shape optimization of vibrating membranes:
  - the first method is inspired by ideas coming from linear elasticity and is based on a spring model of the mesh (works well for convex domains)
  - the other method is based on a “quasi-conformal deformation” – parameterization of an initial reference shape is found by solving optimization problems, then inner control points are generated by quasi-conformal deforming the reference shape into the resulting configuration
- in general, the resulting domain parameterization should satisfy $\det(J) > 0$ everywhere in the parameter domain

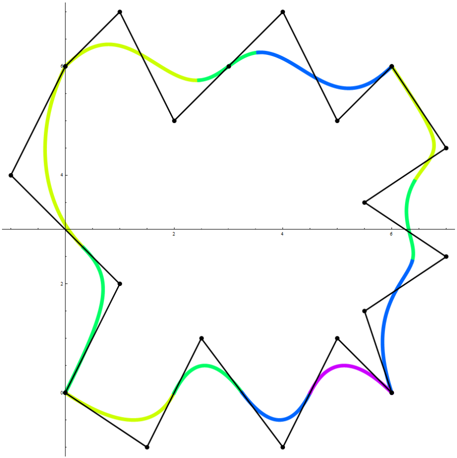
Planar NURBS domains

- one of the key geometric problems in isogeometric analysis is a description of NURBS domains/volumes from their boundary data (contained in CAD model)
- recently, this problem was studied e.g. in Xu et al. (2010) for planar domains, where a sufficient condition for injectivity of planar B-spline parameterization is proposed and also an influence of different parameterizations on computation result is partially studied
- further, Manh et al. (201x) proposes two linear methods for extension of a B-spline parameterization from the boundary of a domain onto its interior during shape optimization of vibrating membranes:
  - the first method is inspired by ideas coming from linear elasticity and is based on a spring model of the mesh (works well for convex domains)
  - the other method is based on a "quasi-conformal deformation" – parameterization of an initial reference shape is found by solving optimization problems, then inner control points are generated by quasi-conformal deforming the reference shape into the resulting configuration
- in general, the resulting domain parameterization should satisfy $\det(J) > 0$ everywhere in the parameter domain

Domain bounded by 4 NURBS curves

- let a simply connected domain be given by 4 boundary NURBS curves – we want to find a control net and corresponding knot vectors for planar NURBS surface describing this domain:

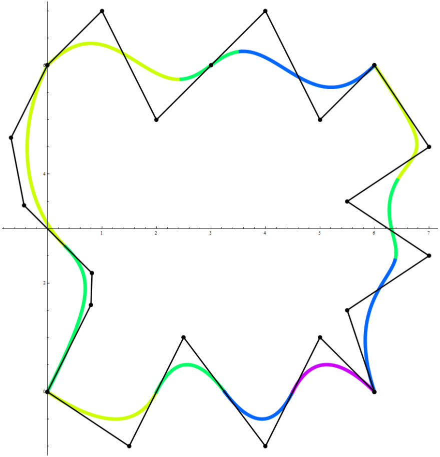
1. for each pair of “opposite” curves we need these curves to have the same degree (degree elevation)
2. ... and the same knots (knot insertion)
3. then we can compute interior control points – we have several possibilities
4. finally, we obtain NURBS surface describing the domain which preserves given boundary curves



Domain bounded by 4 NURBS curves

- let a simply connected domain be given by 4 boundary NURBS curves – we want to find a control net and corresponding knot vectors for planar NURBS surface describing this domain:

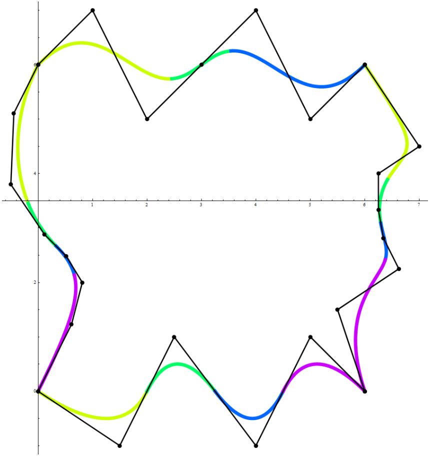
1. for each pair of “opposite” curves we need these curves to have the same degree (degree elevation)
2. ... and the same knots (knot insertion)
3. then we can compute interior control points – we have several possibilities
4. finally, we obtain NURBS surface describing the domain which preserves given boundary curves



Domain bounded by 4 NURBS curves

- let a simply connected domain be given by 4 boundary NURBS curves – we want to find a control net and corresponding knot vectors for planar NURBS surface describing this domain:

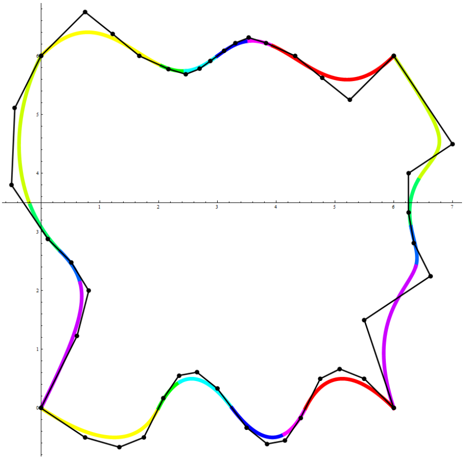
1. for each pair of “opposite” curves we need these curves to have the same degree (degree elevation)
2. ... and the same knots (knot insertion)
3. then we can compute interior control points – we have several possibilities
4. finally, we obtain NURBS surface describing the domain which preserves given boundary curves



Domain bounded by 4 NURBS curves

- let a simply connected domain be given by 4 boundary NURBS curves – we want to find a control net and corresponding knot vectors for planar NURBS surface describing this domain:

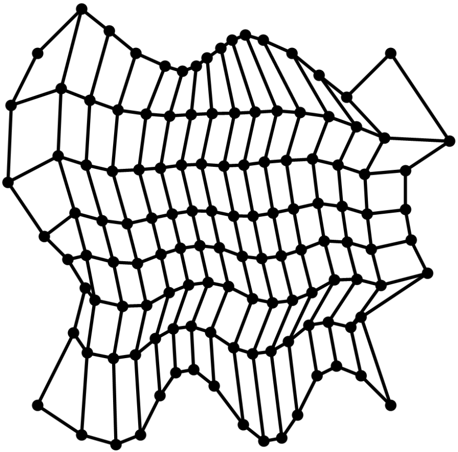
1. for each pair of "opposite" curves we need these curves to have the same degree (degree elevation)
2. ... and the same knots (knot insertion)
3. then we can compute interior control points – we have several possibilities
4. finally, we obtain NURBS surface describing the domain which preserves given boundary curves



Domain bounded by 4 NURBS curves

- let a simply connected domain be given by 4 boundary NURBS curves – we want to find a control net and corresponding knot vectors for planar NURBS surface describing this domain:

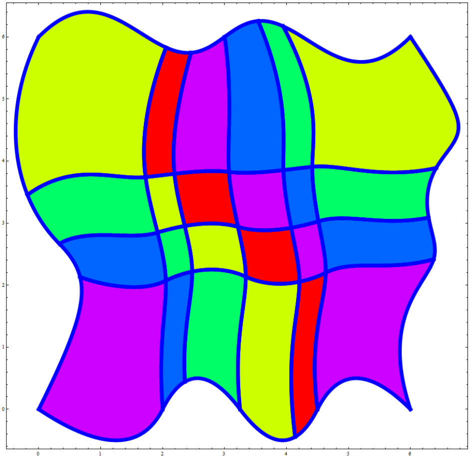
1. for each pair of “opposite” curves we need these curves to have the same degree (degree elevation)
2. ... and the same knots (knot insertion)
3. then we can compute interior control points – we have several possibilities
4. finally, we obtain NURBS surface describing the domain which preserves given boundary curves



Domain bounded by 4 NURBS curves

- let a simply connected domain be given by 4 boundary NURBS curves – we want to find a control net and corresponding knot vectors for planar NURBS surface describing this domain:

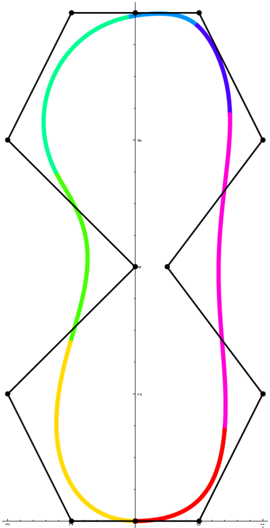
1. for each pair of “opposite” curves we need these curves to have the same degree (degree elevation)
2. ... and the same knots (knot insertion)
3. then we can compute interior control points – we have several possibilities
4. finally, we obtain NURBS surface describing the domain which preserves given boundary curves



Domain bounded by a closed NURBS curve

- let a domain be given by a closed boundary NURBS curve – similarly, we need to find control net and knot vectors for NURBS surface representing given domain

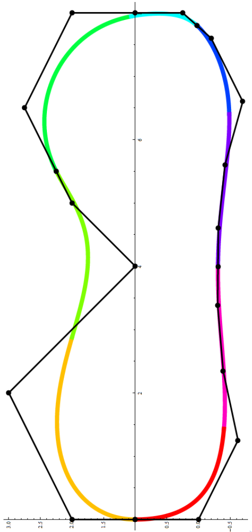
1. using knot insertion, we subdivide curve into 4 parts with equal number of segments
2. then we continue as in the case of 4 boundary NURBS curves – we match degrees and knots of opposite curves
3. we generate interior control points
4. finally, we obtain NURBS surface describing the domain with prescribed boundary curve



Domain bounded by a closed NURBS curve

- let a domain be given by a closed boundary NURBS curve – similarly, we need to find control net and knot vectors for NURBS surface representing given domain

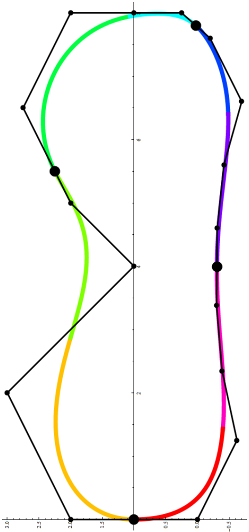
1. using knot insertion, we subdivide curve into 4 parts with equal number of segments
2. then we continue as in the case of 4 boundary NURBS curves – we match degrees and knots of opposite curves
3. we generate interior control points
4. finally, we obtain NURBS surface describing the domain with prescribed boundary curve



Domain bounded by a closed NURBS curve

- let a domain be given by a closed boundary NURBS curve – similarly, we need to find control net and knot vectors for NURBS surface representing given domain

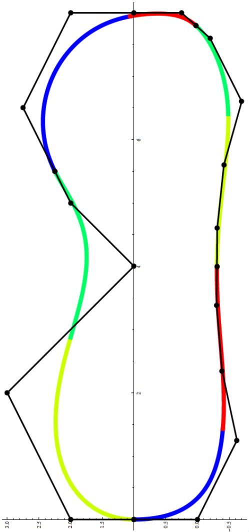
1. using knot insertion, we subdivide curve into 4 parts with equal number of segments
2. then we continue as in the case of 4 boundary NURBS curves – we match degrees and knots of opposite curves
3. we generate interior control points
4. finally, we obtain NURBS surface describing the domain with prescribed boundary curve



Domain bounded by a closed NURBS curve

- let a domain be given by a closed boundary NURBS curve – similarly, we need to find control net and knot vectors for NURBS surface representing given domain

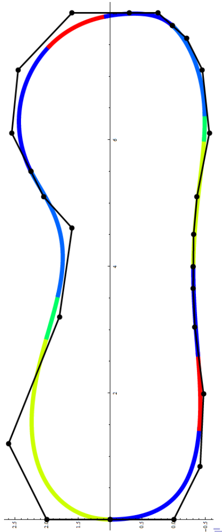
1. using knot insertion, we subdivide curve into 4 parts with equal number of segments
2. then we continue as in the case of 4 boundary NURBS curves – we match degrees and knots of opposite curves
3. we generate interior control points
4. finally, we obtain NURBS surface describing the domain with prescribed boundary curve



Domain bounded by a closed NURBS curve

- let a domain be given by a closed boundary NURBS curve – similarly, we need to find control net and knot vectors for NURBS surface representing given domain

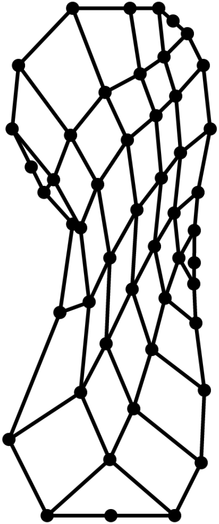
1. using knot insertion, we subdivide curve into 4 parts with equal number of segments
2. then we continue as in the case of 4 boundary NURBS curves – we match degrees and knots of opposite curves
3. we generate interior control points
4. finally, we obtain NURBS surface describing the domain with prescribed boundary curve



Domain bounded by a closed NURBS curve

- let a domain be given by a closed boundary NURBS curve – similarly, we need to find control net and knot vectors for NURBS surface representing given domain

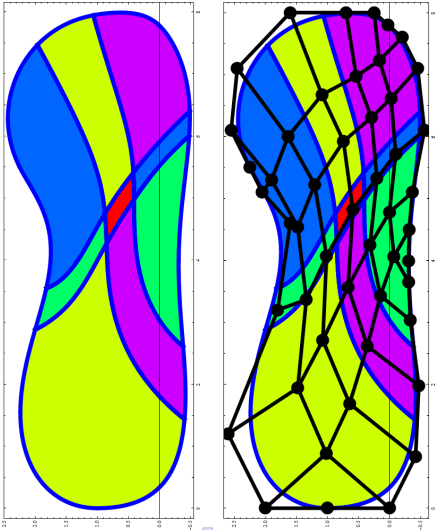
1. using knot insertion, we subdivide curve into 4 parts with equal number of segments
2. then we continue as in the case of 4 boundary NURBS curves – we match degrees and knots of opposite curves
3. we generate interior control points
4. finally, we obtain NURBS surface describing the domain with prescribed boundary curve



Domain bounded by a closed NURBS curve

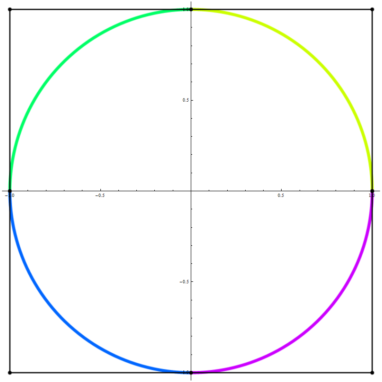
- let a domain be given by a closed boundary NURBS curve – similarly, we need to find control net and knot vectors for NURBS surface representing given domain

1. using knot insertion, we subdivide curve into 4 parts with equal number of segments
2. then we continue as in the case of 4 boundary NURBS curves – we match degrees and knots of opposite curves
3. we generate interior control points
4. finally, we obtain NURBS surface describing the domain with prescribed boundary curve



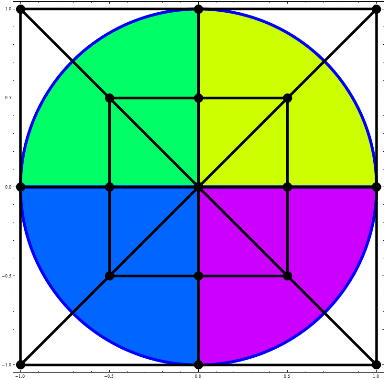
NURBS domain – example

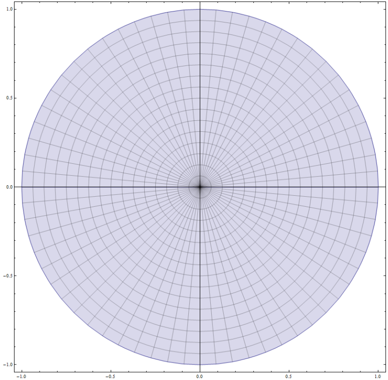
- let us find a parameterization of the unit disc, if the boundary unit circle is given in NURBS form



NURBS domain – example

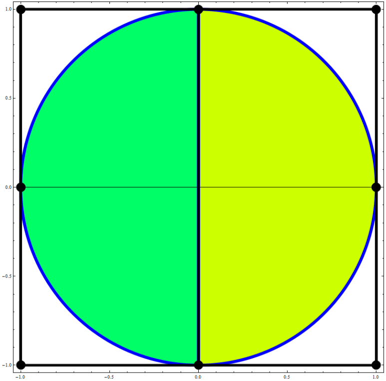
1. we can add circle center to the control net and connect all boundary control points with the center

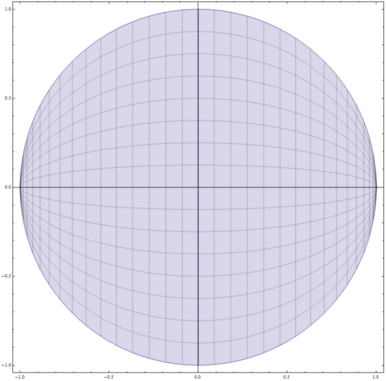




NURBS domain – example

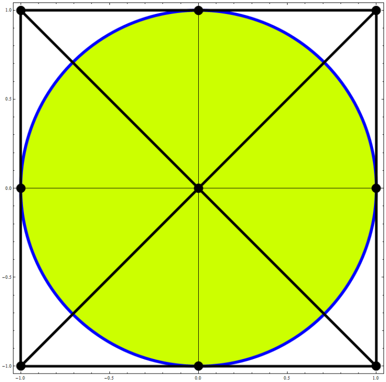
2. we can divide circle into two parts and use similar algorithm as for 4 boundary NURBS curves

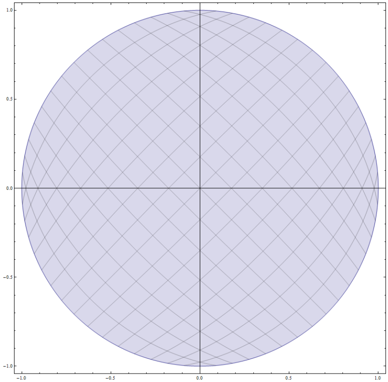




NURBS domain – example

3. or, we can divide circle into four parts and use algorithm for 4 boundary NURBS curves





NURBS volumes

- the boundary surfaces of the object are given as NURBS surfaces and we need to generate a volume parameterization which preserves the given boundary surfaces
- this is one of the main issues concerning the isogeometric analysis – it is very difficult open problem for general CAD objects, but it is possible to obtain results for special classes of free-form objects
- Aigner et al. propose a variational framework for generating NURBS parameterizations of swept volumes, which are obtained by sweeping a closed curve through space
- Martin et al. (2009) presents a method for finding NURBS volume parameterzation based on discrete volumetric harmonic functions which uses closed triangle mesh representing the exterior geometric shape of the object and interior triangle meshes that can represent material attributes or other interior features

NURBS volumes

- the boundary surfaces of the object are given as NURBS surfaces and we need to generate a volume parameterization which preserves the given boundary surfaces
- this is one of the main issues concerning the isogeometric analysis – it is very difficult open problem for general CAD objects, but it is possible to obtain results for special classes of free-form objects
- Aigner et al. propose a variational framework for generating NURBS parameterizations of swept volumes, which are obtained by sweeping a closed curve through space
- Martin et al. (2009) presents a method for finding NURBS volume parameterzation based on discrete volumetric harmonic functions which uses closed triangle mesh representing the exterior geometric shape of the object and interior triangle meshes that can represent material attributes or other interior features

NURBS volumes

- the boundary surfaces of the object are given as NURBS surfaces and we need to generate a volume parameterization which preserves the given boundary surfaces
- this is one of the main issues concerning the isogeometric analysis – it is very difficult open problem for general CAD objects, but it is possible to obtain results for special classes of free-form objects
- Aigner et al. propose a variational framework for generating NURBS parameterizations of swept volumes, which are obtained by sweeping a closed curve through space
- Martin et al. (2009) presents a method for finding NURBS volume parameterzation based on discrete volumetric harmonic functions which uses closed triangle mesh representing the exterior geometric shape of the object and interior triangle meshes that can represent material attributes or other interior features

NURBS volumes

- the boundary surfaces of the object are given as NURBS surfaces and we need to generate a volume parameterization which preserves the given boundary surfaces
- this is one of the main issues concerning the isogeometric analysis – it is very difficult open problem for general CAD objects, but it is possible to obtain results for special classes of free-form objects
- Aigner et al. propose a variational framework for generating NURBS parameterizations of swept volumes, which are obtained by sweeping a closed curve through space
- Martin et al. (2009) presents a method for finding NURBS volume parameterzation based on discrete volumetric harmonic functions which uses closed triangle mesh representing the exterior geometric shape of the object and interior triangle meshes that can represent material attributes or other interior features

Poisson equation with Dirichlet boundary condition

- we want to solve the Dirichlet boundary value problem given by

$$\begin{cases} u_{xx} + u_{yy} = f & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where Ω is a connected open region in xy-plane whose boundary $\delta\Omega$ is "nice" (e.g. a smooth curve or a polygon)

- using Green's theorem, if u solves (1), then for any v ($v = 0$ on $\delta\Omega$) it holds

$$\int_\Omega f v ds = \int_\Omega v \Delta u ds = -\int_\Omega \nabla u \cdot \nabla v ds = -\phi(u, v)$$

- the basic idea is similar to FEM – we need to replace an infinite dimensional problem (u, $v \in \mathbf{H}_0^1$) by a finite dimensional one ($u \in \mathbf{V}_g$, $v \in \mathbf{V}$)
- $\mathbf{V}$ is usually chosen as a set of piecewise linear functions over Ω and one of the advantages of this choice of basis functions is that integrals

$$\int_\Omega v_j v_k ds \quad \text{a} \quad \int_\Omega \nabla v_j \cdot \nabla v_k ds$$

vanish if vertices x_j and x_k do not share a common edge in the triangulation

Poisson equation with Dirichlet boundary condition

- further, we can write f and u as a linear combination of the new basis functions

$$u(x) = \sum_{k=1}^{n} u_k v_k(x), \; f(x) = \sum_{k=1}^{n} f_k v_k(x)$$

- then we obtain

$$-\phi(u,v) = \int fv \longrightarrow -\sum_{k=1}^{n} u_k \phi(v_k, v_j) = \sum_{k=1}^{n} f_k \int v_k v_j, \; j = 1, \dots, n \qquad (2)$$

- if $\mathbf{u} = (u_1, \dots, u_n)^T$, $\mathbf{f} = (f_1, \dots, f_n)^T$ and $L = (L_{ij}) = (\phi(v_i, v_j))$, $M = (M_{ij}) = (\int v_i v_j)$, then (2) can be rewritten into the form

$$-L\mathbf{u} = M\mathbf{f}$$

- since basis functions v_k have small support, L and M are sparse matrices
- moreover, L is symmetric and positive definite, so efficient solvers for system of linear equations can be used (e.g. conjugate gradient method)

Poisson equation with Dirichlet boundary condition

- further, we can write f and u as a linear combination of the new basis functions

$$u(x) = \sum_{k=1}^{n} u_k v_k(x), \ f(x) = \sum_{k=1}^{n} f_k v_k(x)$$

- then we obtain

$$-\phi(u,v) = \int fv \longrightarrow -\sum_{k=1}^{n} u_k \phi(v_k, v_j) = \sum_{k=1}^{n} f_k \int v_k v_j, \ j = 1, \dots, n \quad (2)$$

- if $\mathbf{u} = (u_1, \dots, u_n)^T$, $\mathbf{f} = (f_1, \dots, f_n)^T$ and $L = (L_{ij}) = (\phi(v_i, v_j))$, $M = (M_{ij}) = (\int v_i v_j)$, then (2) can be rewritten into the form

$$-L\mathbf{u} = M\mathbf{f}$$

- since basis functions v_k have small support, L and M are sparse matrices
- moreover, L is symmetric and positive definite, so efficient solvers for system of linear equations can be used (e.g. conjugate gradient method)

Poisson equation with Dirichlet boundary condition

- further, we can write f and u as a linear combination of the new basis functions

$$u(x) = \sum_{k=1}^{n} u_k v_k(x), \; f(x) = \sum_{k=1}^{n} f_k v_k(x)$$

- then we obtain

$$-\phi(u,v) = \int fv \longrightarrow -\sum_{k=1}^{n} u_k \phi(v_k, v_j) = \sum_{k=1}^{n} f_k \int v_k v_j, \; j = 1, \ldots, n \qquad (2)$$

- if $\mathbf{u} = (u_1, \ldots, u_n)^T$, $\mathbf{f} = (f_1, \ldots, f_n)^T$ and $L = (L_{ij}) = (\phi(v_i, v_j))$, $M = (M_{ij}) = (\int v_i v_j)$, then (2) can be rewritten into the form

$$-L\mathbf{u} = M\mathbf{f}$$

- since basis functions v_k have small support, L and M are sparse matrices
- moreover, L is symmetric and positive definite, so efficient solvers for system of linear equations can be used (e.g. conjugate gradient method)

Poisson equation with Dirichlet boundary condition

- further, we can write f and u as a linear combination of the new basis functions

$$u(x) = \sum_{k=1}^{n} u_k v_k(x), \; f(x) = \sum_{k=1}^{n} f_k v_k(x)$$

- then we obtain

$$-\phi(u,v) = \int fv \longrightarrow -\sum_{k=1}^{n} u_k \phi(v_k, v_j) = \sum_{k=1}^{n} f_k \int v_k v_j, \; j = 1, \dots, n \qquad (2)$$

- if $\mathbf{u} = (u_1, \dots, u_n)^T$, $\mathbf{f} = (f_1, \dots, f_n)^T$ and $L = (L_{ij}) = (\phi(v_i, v_j))$, $M = (M_{ij}) = (\int v_i v_j)$, then (2) can be rewritten into the form

$$-L\mathbf{u} = M\mathbf{f}$$

- since basis functions v_k have small support, L and M are sparse matrices
- moreover, L is symmetric and positive definite, so efficient solvers for system of linear equations can be used (e.g. conjugate gradient method)

Poisson equation with Dirichlet boundary condition

- further, we can write f and u as a linear combination of the new basis functions

$$u(x) = \sum_{k=1}^{n} u_k v_k(x), \; f(x) = \sum_{k=1}^{n} f_k v_k(x)$$

- then we obtain

$$-\phi(u,v) = \int fv \longrightarrow -\sum_{k=1}^{n} u_k \phi(v_k, v_j) = \sum_{k=1}^{n} f_k \int v_k v_j, \; j = 1, \dots, n \qquad (2)$$

- if $\mathbf{u} = (u_1, \dots, u_n)^T$, $\mathbf{f} = (f_1, \dots, f_n)^T$ and $L = (L_{ij}) = (\phi(v_i, v_j))$, $M = (M_{ij}) = (\int v_i v_j)$, then (2) can be rewritten into the form

$$-L\mathbf{u} = M\mathbf{f}$$

- since basis functions v_k have small support, L and M are sparse matrices
- moreover, L is symmetric and positive definite, so efficient solvers for system of linear equations can be used (e.g. conjugate gradient method)

Poisson equation with Dirichlet boundary condition

- in isogeometric analysis, we use rational B-splines as basis function of $\mathbf{V}$, i.e., we express the right hand side and the approximate solution as linear combinations of the basis functions representing the geometry

$$f(s,t) = \sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} f_{i,j} R_{i,j}(s,t), \ u(s,t) = \sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} u_{i,j} R_{i,j}(s,t)$$

- then

$$\begin{aligned} -\phi(u,v) &= \int_\Omega f v ds \\ &\downarrow \\ -\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} u_{i,j} \phi(R_{i,j}, R_{k,l}) &= \sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} f_{i,j} \varphi(R_{i,j}, R_{k,l}), \\ & \quad k = 0,\dots,n, l = 0,\dots,m \end{aligned}$$

where

$$\begin{aligned} \phi(R_{i,j}, R_{k,l}) &= \int_A (\nabla R_{i,j} \cdot J^{-1}) \cdot (\nabla R_{k,l} \cdot J^{-1}) |\det(J)| dA \\ \varphi(R_{i,j}, R_{k,l}) &= \int_A R_{i,j} R_{k,l} |\det(J)| dA \end{aligned}$$

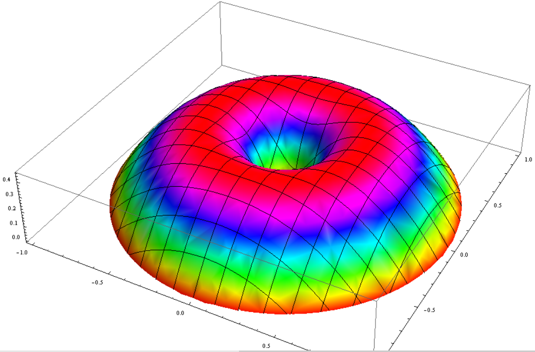
Homogeneous Dirichlet boundary value problem

- we want to solve the homogeneous Dirichlet boundary value problem

$$\begin{aligned} u_{xx} + u_{yy} &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

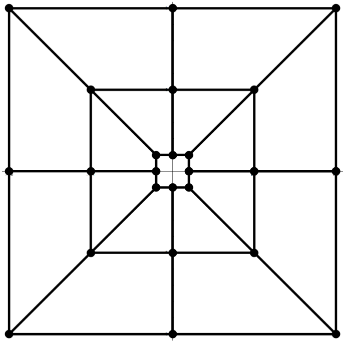
  where $\Omega = \{(x,y) \in \mathbb{R}^2 : \frac{1}{100} < x^2 + y^2 < 1\}$ with the help of isogeometric analysis

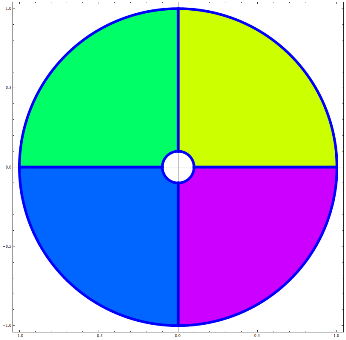
- for $f = -4$, the exact solution is $u(x,y) = \frac{99 \log\left(x^2+y^2\right)}{200 \log(10)} - x^2 - y^2 + 1$



Homogeneous Dirichlet boundary value problem

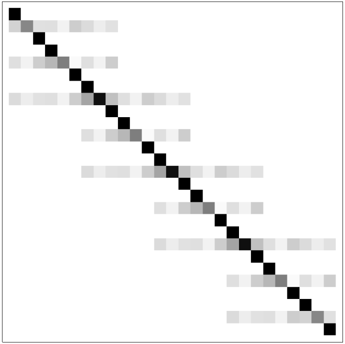
- 4 elements (knot vectors are $V = (0, 0, 0, 1/4, 1/4, 1/2, 1/2, 3/4, 3/4, 1, 1, 1)$, $U = (0, 0, 0, 1, 1, 1)$, control net shown below)

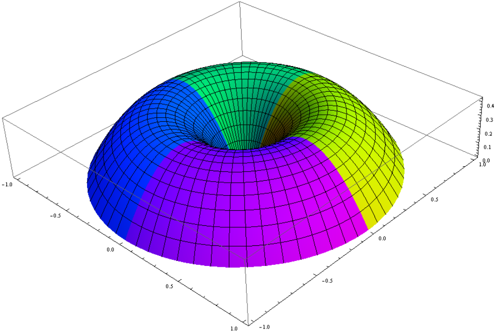




Homogeneous Dirichlet boundary value problem

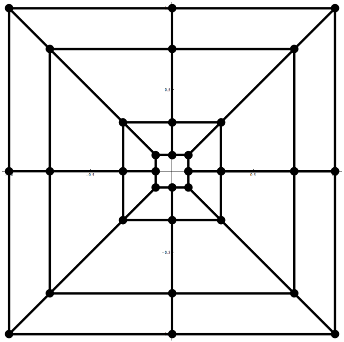
- stiffness matrix is sparse and of size 27×27 and the obtained approximate solution is shown

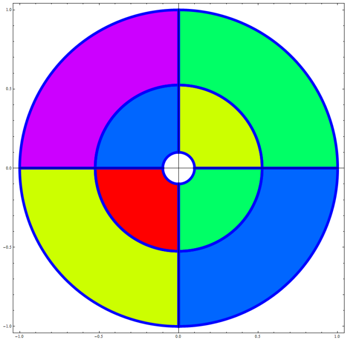




Homogeneous Dirichlet boundary value problem

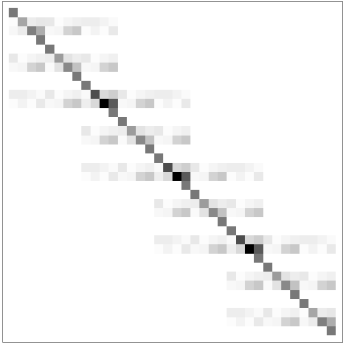
- 8 elements (knot vectors are $V = (0, 0, 0, 1/4, 1/4, 1/2, 1/2, 3/4, 3/4, 1, 1, 1)$, $U = (0, 0, 0, 1/2, 1, 1, 1)$, control net shown below)

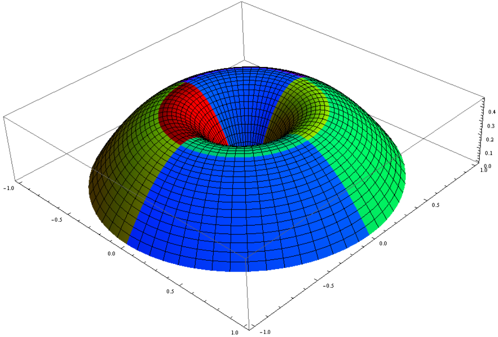




Homogeneous Dirichlet boundary value problem

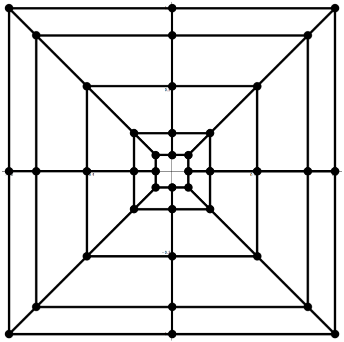
- stiffness matrix is sparse and of size 36×36 and the obtained approximate solution is shown

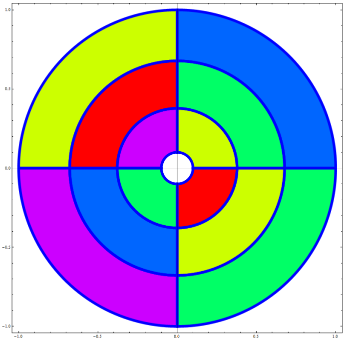




Homogeneous Dirichlet boundary value problem

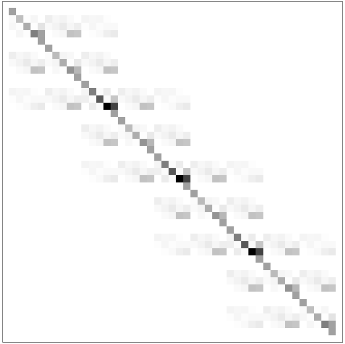
- 12 elements (knot vectors are $V = (0, 0, 0, 1/4, 1/4, 1/2, 1/2, 3/4, 3/4, 1, 1, 1)$, $U = (0, 0, 0, 1/3, 2/3, 1, 1, 1)$, control net shown below)

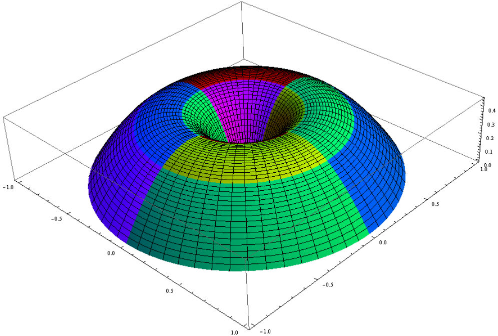




Homogeneous Dirichlet boundary value problem

- stiffness matrix is sparse and of size 45×45 and the obtained approximate solution is shown

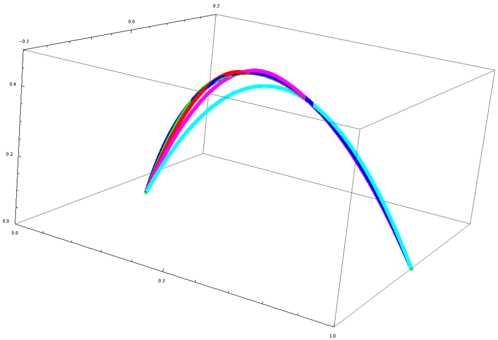




Homogeneous Dirichlet boundary value problem

- approximate solution tends quickly to exact solution

Parts	Error	Ratio
4	0.05801	
8	0.02248	2.58
12	0.00989	2.27
16	0.00526	1.88



NURBS surface approximation

- let Ω be a planar NURBS domain given by control points $\mathbf{P} = (P_{i,j})$ and knot vectors U, V in xy-plane and $f(x, y)$ be an arbitrary function defined over Ω
- we want to find a NURBS approximation of $f(x, y)$ over Ω, i.e., we need to assign an appropriate z-coordinate to each $P_{i,j}$ such that the resulting NURBS surface approximates given function $f(x, y)$

Algorithm:

1. we assign a pair of parameter values $(u_{i,j}, v_{i,j})$ to each control point $P_{i,j}$ – this is the main issue of the approximation method
2. each pair of parameters $(u_{i,j}, v_{i,j})$ determines point $(x_{i,j}, y_{i,j}) \in \Omega$
3. then, for each pair (i, j) we obtain one equation

$$f(x_{i,j}, y_{i,j}) = \sum_{k,l} P^z_{i,j} R_{k,l}(u_{i,j}, v_{i,j})$$

which give us a system of linear equations for unknown z-coordinates $P^z_{i,j}$ of control points

NURBS surface approximation

- let Ω be a planar NURBS domain given by control points $\mathbf{P} = (P_{i,j})$ and knot vectors U, V in xy-plane and $f(x, y)$ be an arbitrary function defined over Ω
- we want to find a NURBS approximation of $f(x, y)$ over Ω, i.e., we need to assign an appropriate z-coordinate to each $P_{i,j}$ such that the resulting NURBS surface approximates given function $f(x, y)$

Algorithm:

1. we assign a pair of parameter values $(u_{i,j}, v_{i,j})$ to each control point $P_{i,j}$ – this is the main issue of the approximation method
2. each pair of parameters $(u_{i,j}, v_{i,j})$ determines point $(x_{i,j}, y_{i,j}) \in \Omega$
3. then, for each pair (i, j) we obtain one equation

$$f(x_{i,j}, y_{i,j}) = \sum_{k,l} P^z_{i,j} R_{k,l}(u_{i,j}, v_{i,j})$$

which give us a system of linear equations for unknown z-coordinates $P^z_{i,j}$ of control points

NURBS surface approximation

- let Ω be a planar NURBS domain given by control points $\mathbf{P} = (P_{i,j})$ and knot vectors U, V in xy-plane and $f(x, y)$ be an arbitrary function defined over Ω
- we want to find a NURBS approximation of $f(x, y)$ over Ω, i.e., we need to assign an appropriate z-coordinate to each $P_{i,j}$ such that the resulting NURBS surface approximates given function $f(x, y)$

Algorithm:

1. we assign a pair of parameter values $(u_{i,j}, v_{i,j})$ to each control point $P_{i,j}$ – this is the main issue of the approximation method
2. each pair of parameters $(u_{i,j}, v_{i,j})$ determines point $(x_{i,j}, y_{i,j}) \in \Omega$
3. then, for each pair (i, j) we obtain one equation

$$f(x_{i,j}, y_{i,j}) = \sum_{k,l} P^z_{i,j} R_{k,l}(u_{i,j}, v_{i,j})$$

which give us a system of linear equations for unknown z-coordinates $P^z_{i,j}$ of control points

NURBS surface approximation

- let Ω be a planar NURBS domain given by control points $\mathbf{P} = (P_{i,j})$ and knot vectors U, V in xy-plane and $f(x, y)$ be an arbitrary function defined over Ω
- we want to find a NURBS approximation of $f(x, y)$ over Ω, i.e., we need to assign an appropriate z-coordinate to each $P_{i,j}$ such that the resulting NURBS surface approximates given function $f(x, y)$

Algorithm:

1. we assign a pair of parameter values $(u_{i,j}, v_{i,j})$ to each control point $P_{i,j}$ – this is the main issue of the approximation method
2. each pair of parameters $(u_{i,j}, v_{i,j})$ determines point $(x_{i,j}, y_{i,j}) \in \Omega$
3. then, for each pair (i, j) we obtain one equation

$$f(x_{i,j}, y_{i,j}) = \sum_{k,l} P^z_{i,j} R_{k,l}(u_{i,j}, v_{i,j})$$

which give us a system of linear equations for unknown z-coordinates $P^z_{i,j}$ of control points

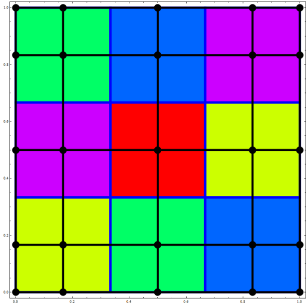
Non-homogeneous Dirichlet boundary value problem

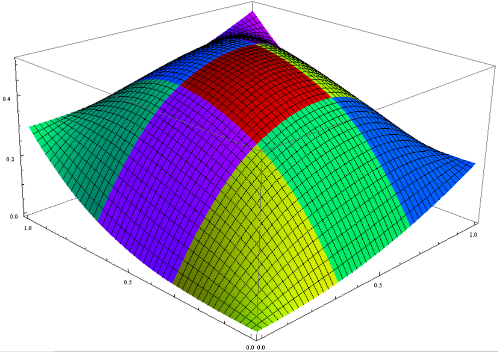
- we solve the non-homogeneous Dirichlet boundary value problem

$$\begin{aligned} u_{xx} + u_{yy} &= -4 && \text{in } \Omega, \\ u &= 0.2x^2 + 0.3y^3 && \text{on } \partial\Omega, \end{aligned}$$

where $\Omega = \{(x,y) \in \mathbb{R}^2 : 0 < x < 1 \wedge 0 < y < 1\}$

- NURBS mesh and the corresponding approximate solution





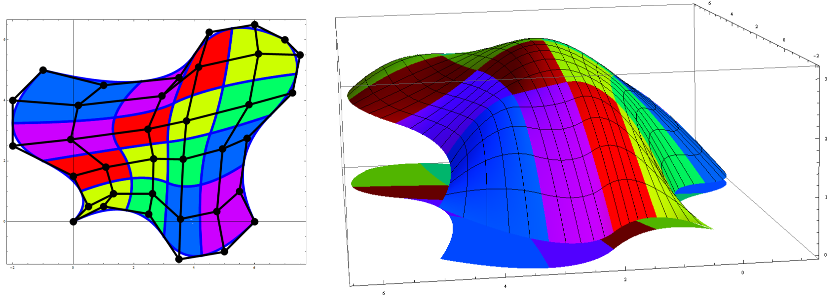
Non-homogeneous Dirichlet boundary value problem

- we solve the non-homogeneous Dirichlet boundary value problem

$$\begin{aligned} u_{xx} + u_{yy} &= -1 \quad \text{in } \Omega, \\ u &= 0.1x + 0.2y \quad \text{on } \partial\Omega, \end{aligned}$$

where Ω is planar NURBS domain given by 4 boundary NURBS curves

- NURBS mesh and the corresponding approximate solution



For Further Reading

Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y.:
Isogeometric Analysis.
John Wiley & Sons, Ltd, 2009.

L. Piegl and W. Tiller
The NURBS Book.
Springer, 1996.

Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y.:
Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement.
Computer Methods in Applied Mechanics and Engineering, 194:4135–4195, 2005.

Y. Bazilevs, L. Beirão de Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli
Isogeometric analysis: approximation, stability and error estimates for refined meshes.
Mathematical Models and Methods in Applied Sciences, 6:1031–1090, 2006.

For Further Reading

M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon and A.-V. Vuong
Swept volume parameterization for isogeometric analysis.
Technical report of EXCITING project, http://www.exciting-project.eu.

T. Martin, E. Cohen, R. M. Kirby:
Volumetric parameterization and trivariate B-spline fitting using harmonic functions.
Computer Aided Geometric Design, Vol. 26, pp. 648-664. Elsevier, 2009.

N. D. Manh, A. Evgrafov, A. R. Gersborg, J. Gravesen:
Isogeometric shape optimization of vibrating membrane.
Computer Methods in Applied Mechanics and Engineering, to appear. DOI:10.1016/j.cma.2010.12.015

G. Xu, B. Mourrain, R. Duvigneau, A. Galligo:
Optimal Analysis-Aware Parameterization of Computational Domain in Isogeometric Analysis.
Advances in Geometric Modeling and Processing: LNCS 6130, pp. 236-254. Springer, 2010.