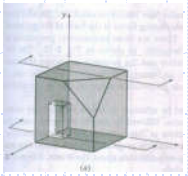
Computer Aided Geometric Design

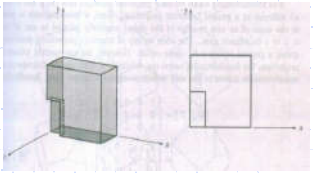
Indicative Term Projects

Instructor: Dr. Prashant K. Jain
Associate Professor ME Discipline
PDPM Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Jabalpur, INDIA
Resources: web.iiitdmj.ac.in/~pkjain/
Email: pkjain@iiitdmj.ac.in, pkjain2006@gmail.com
http://www.iiitdmj.ac.in/Faculty/pkjain.html
http://in.linkedin.com/in/pkjain2006
https://www.facebook.com/pkjain2006

PDPM
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING JABALPUR
(An Institute of National Importance (INI) established by MHRD, Govt. of INDIA)

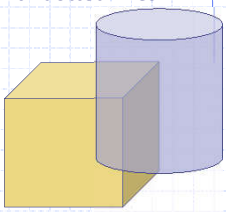
Find and draw the sectional view of the given solid

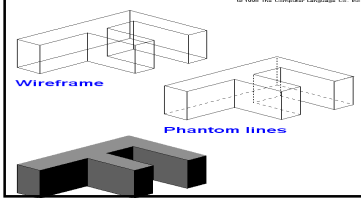




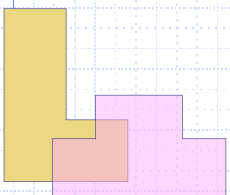
Implement the Hidden Line Removal Operation

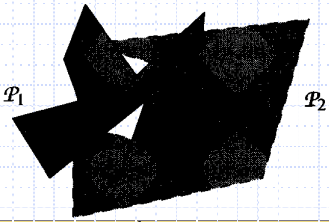
- Find hidden lines and replace with dotted lines





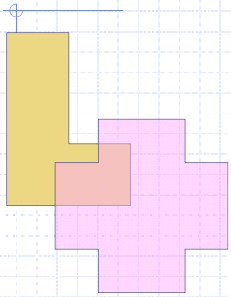
Perform Boolean operations of given 2D polygons: Union

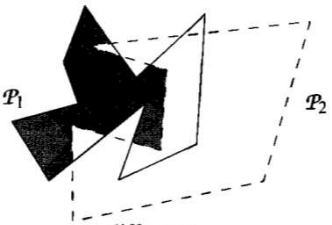




union

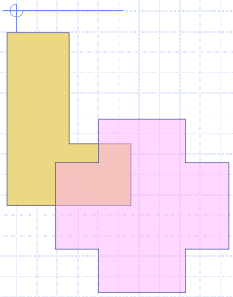
Perform Boolean operations of given 2D polygons: Subtraction

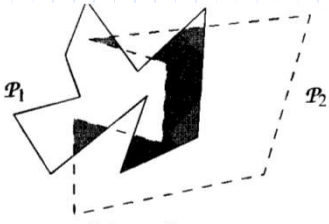




difference

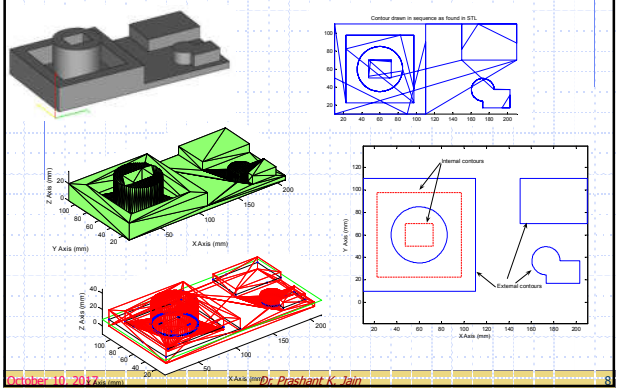
Perform Boolean operations of given 2D polygons: Intersection



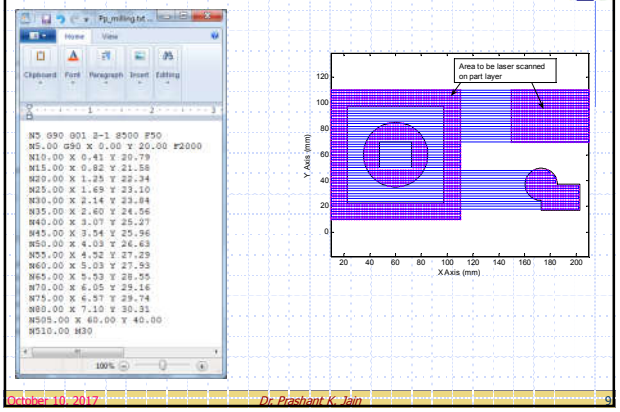


intersection

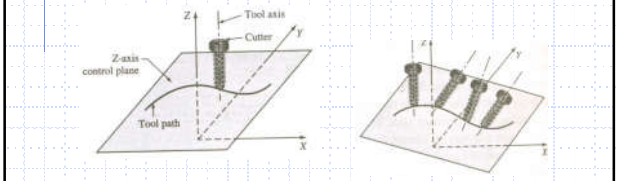
Generate part program from a solid model for milling



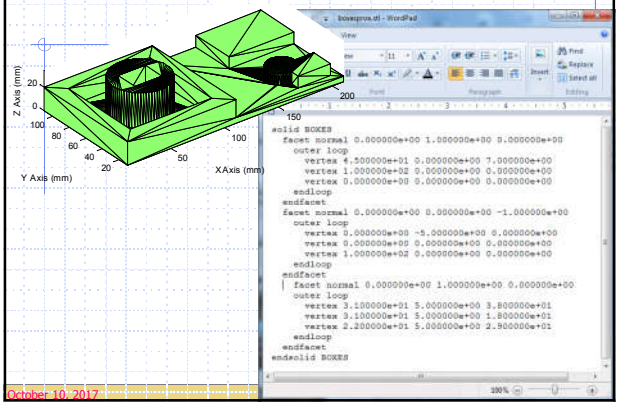
Generate part program from a solid model for milling



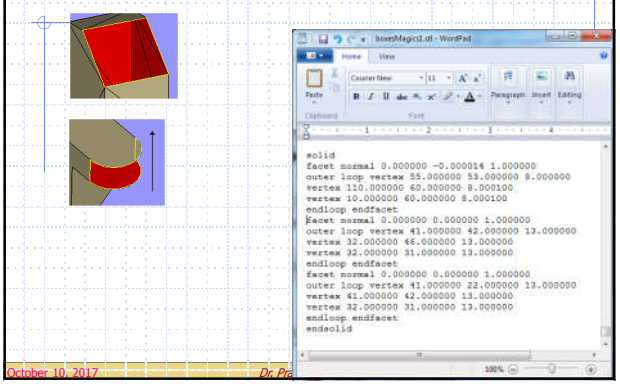
Tool path generation for free form surfaces



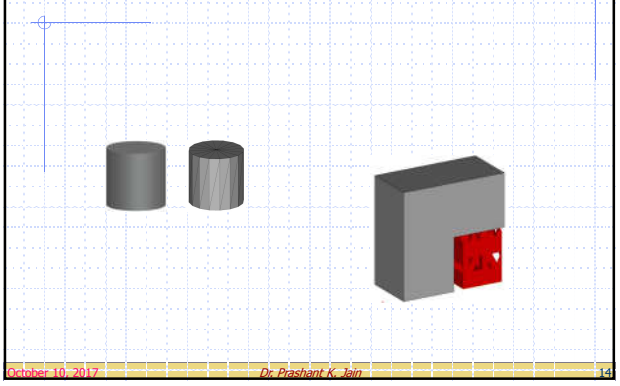
Generate STL file through point cloud data



Repairing of STL file (find and fill missing facets)

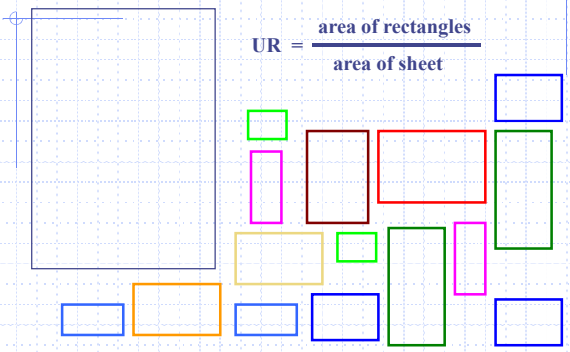


Volume of support required in RP



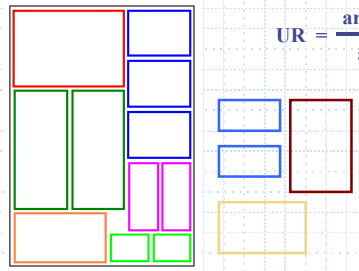
Stock Cutting Problem in 2D

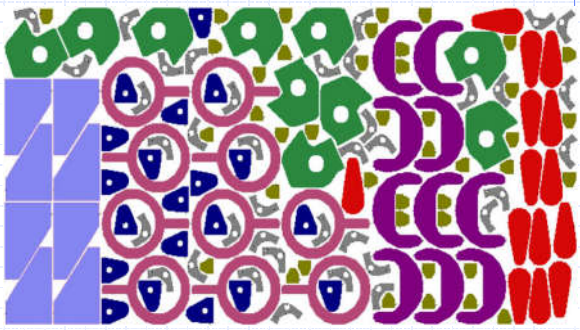
$$UR = \frac{\text{area of rectangles}}{\text{area of sheet}}$$



Stock Cutting Problem in 2D

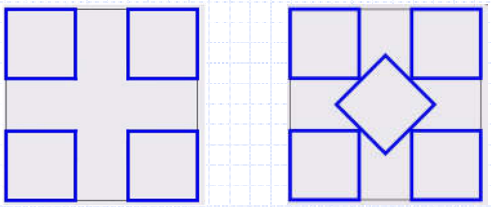
$$UR = \frac{\text{area of rectangles}}{\text{area of sheet}}$$





2D Nesting for same objects in specified area

- In 2D packing the goal is to fit as many items as possible into a specified area, without overlapping.

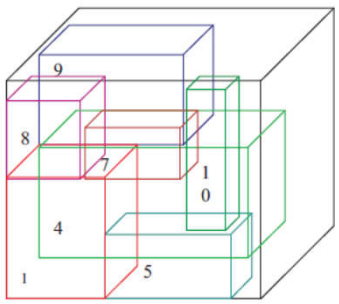


2D Nesting for same objects with one dimension varying

- In 2D packing the goal is to fit as many items as possible into a specified sheet with length can be varied to maximize utilization, without overlapping.

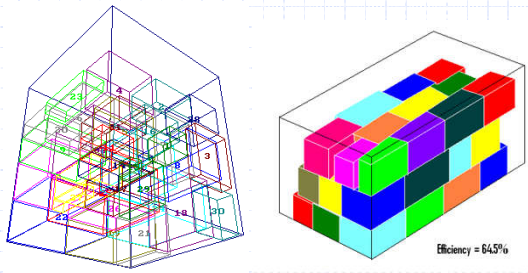
Find out optimum volume for packing different boxes

Find optimum rectangular size of the box to pack all the boxes



Compact Packing Problem in 3D

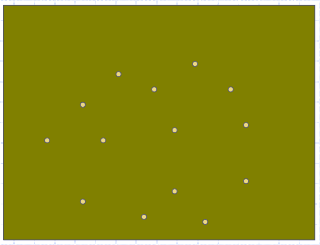
In 3D packing the goal is to fit as many items as possible into a specified volume to maximize utilization, without interference.



Optimum tool path for drilling multiple holes

- Traveling Salesman Problem

A number of holes have to be drilled in a plate. Following are the locations where the holes have to be drilled. Do you suggest any sequence in which the holes have to be drilled so that total time required for drilling is minimized.

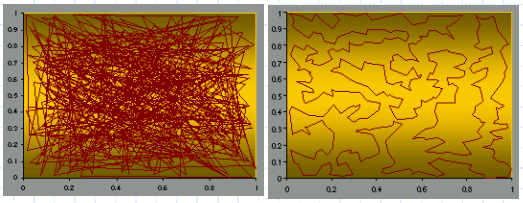


No of Solutions = 87178291200

Optimum tool path for Soldering

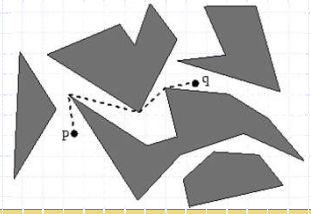
- Traveling Salesman Problem

Soldering is to be carried out at large number of locations in PCB. Following are the locations where soldering is to be done. Do you suggest any sequence in which soldering is to be carried out so that total time required for soldering is minimized.

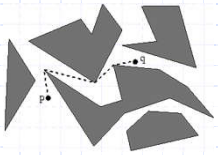


Robot Path Optimization Problem

In automated factories mobile robots are used to move machined components from one machine to another. Let us say robot has to travel from one machine to another in a shortest possible time avoiding obstacles which come in its path. If obstacles can be approximated as polygonal areas, what should be the path which robot takes to reach from one machine to another.

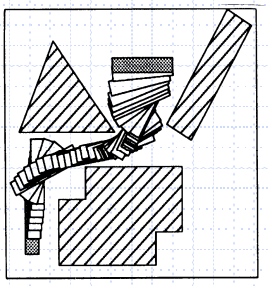


Robot Path Optimization Problem

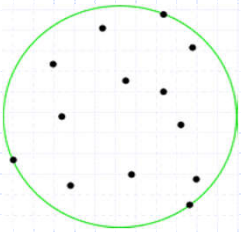


Simplification	Real Life Situation
Moving object is a point	Moving object is a geometric entity
Obstacles are polygons	Obstacles are curved objects
Path Planning	Motion Planning
Stationary obstacles	Moving obstacles
2D	3D

Robot Path Optimization Problem



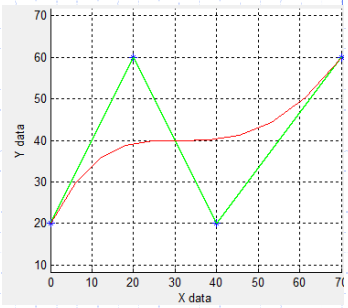
Layout Planning Problem



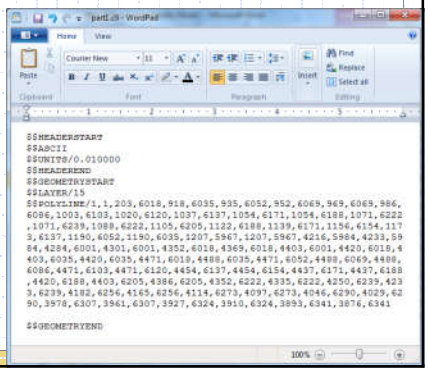
Center Point Algorithm

Find circle of minimum radius to encircle all the points

Remove collinear points from linear approximation of a freeform curve and write CLI file

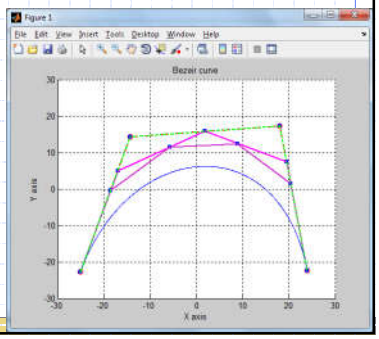


Remove collinear points from linear approximation of a freeform curve and write CLI file



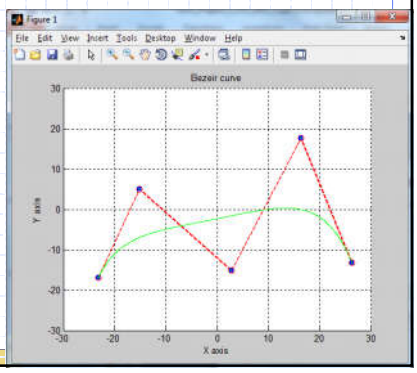
Degree decreasing/raising of Bezier curve

- Decrease the degree of curve by eliminating control polygon vertex and redistribute the vertex of new polygon so that Bezier curve remain same



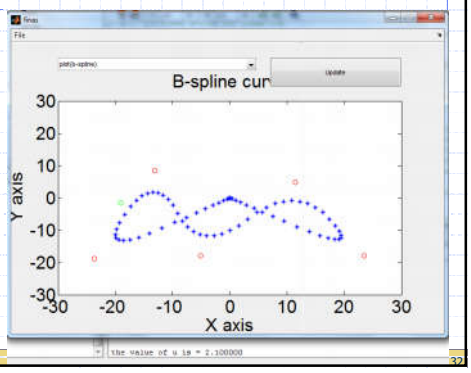
Inverse point solution for Bezier curve

- Find out the parametric value at given point on *n* degree Bezier curve

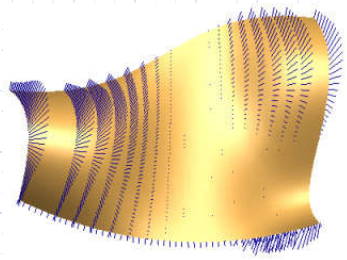


Inverse point solution for B-spline curve

- Find out the parametric value at given point on *n* degree B-Spline curve



Intrinsic properties of Bezier curve, tangent, Normal, curvature plot



Bezier curve/surface fitting

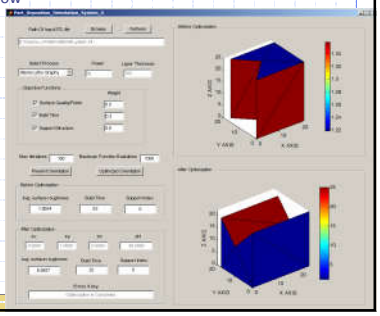
- Fit a n degree Bezier curve in given points
- Fit a nxn degree Bezier surface in given points

B Spline curve/surface fitting

- Fit a n degree B-Spline curve in given points
- Fit a nxn degree B-Spline surface in given points

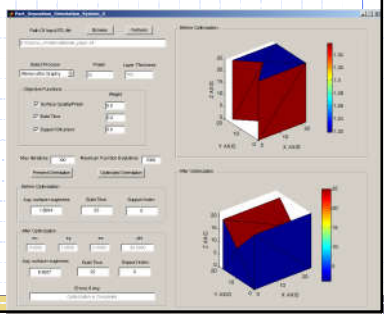
Generate NURBS

- Generate NURBS for given parameters
- Select parameters using GUI window
- Plot the curve in GUI window



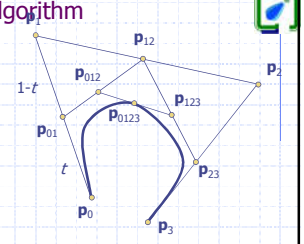
Generate B-Spline curve

- GUI may be developed to select knot vectors degree of curve and no. of points and B-Spline curve plotted in GUI window with toolbar



Implement de Casteljau Algorithm

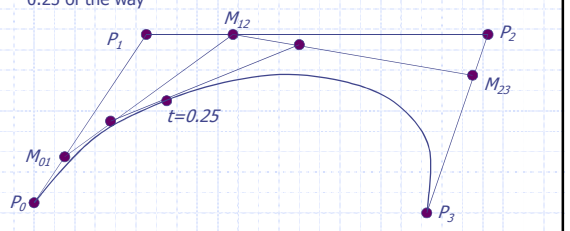
- Cascading lerps
  - $\mathbf{p}_{01} = (1-t)\,\mathbf{p}_0 + t\,\mathbf{p}_1$
  - $\mathbf{p}_{12} = (1-t)\,\mathbf{p}_1 + t\,\mathbf{p}_2$
  - $\mathbf{p}_{23} = (1-t)\,\mathbf{p}_2 + t\,\mathbf{p}_3$
  - $\mathbf{p}_{012} = (1-t)\,\mathbf{p}_{01} + t\,\mathbf{p}_{12}$
  - $\mathbf{p}_{123} = (1-t)\,\mathbf{p}_{12} + t\,\mathbf{p}_{23}$
  - $\mathbf{p}_{0123} = (1-t)\,\mathbf{p}_{012} + t\,\mathbf{p}_{123}$
- Subdivides curve at $\mathbf{p}_{0123}$
  - $\mathbf{p}_0\ \mathbf{p}_{01}\ \mathbf{p}_{012}\ \mathbf{p}_{0123}$
  - $\mathbf{p}_{0123}\ \mathbf{p}_{123}\ \mathbf{p}_{23}\ \mathbf{p}_3$
- Repeated subdivision converges to curve



coordinate free!

de Casteljau's Algorithm

- You can find the point on a Bezier curve for any parameter value *t* with a similar algorithm
- Say you want *t=0.25*, instead of taking midpoints take points 0.25 of the way



THANK YOU