# MATHEMATICA ROUTINES FOR DIMENSION AND EULER CHARACTERSTIC OF GRAPHS

OLIVER KNILL

Abstract. The following Mathematica routines are also available on the Wolfram demonstration project, where the source code can be grabbed without having to retype it. They allow a reader to experiment with the concept and look at more examples to the theorems.

Depending on the context, computers can deal with graphs **geometrically** using concrete points in $R^n$, **algebraically** using incidence matrices or **graph theoretically**, where a graph is dealt with as two sets $V, E$, the vertex set and $E \subset V \times V$. As computer algebra systems have evolved, it is natural to use more and more to the abstract method. The following code is pretty condensed. I feel that compact code is in general also faster to grasp. If you should not agree, it should be easy to rearrange the individual procedures. It can help to look under the hood.

The first routines compute the dimension of an arbitrary graph. Mathematica already knows how to compute the unit ball. We have to take away the vertex itself to get the unit sphere. Dimension is then defined inductively as 1 plus the average of the dimensions of the unit spheres. At the end we include an example on how to call the routine. It produces a list of the local dimensions of a random graph with 15 vertices and 40 edges.

```
UnitSphere[s_,a_]:=Module[{},b=NeighborhoodGraph[s,a];
  If[Length[VertexList[b]]<2,Graph[{}],VertexDelete[b,a]]];
Dimension[s_]:=Module[{v,VL=VertexList,u,ord,size,e},
  v=VL[s];ord=Length[v];e=EdgeList[s]; size= Length[e];
  If[ord==0,u={},u=Table[UnitSphere[s,v[[k]]],{k,ord}]];
  If[size==0,0,Sum[If[Length[VL[u[[k]]]]==0,0,1]+
  Dimension[u[[k]]],{k,ord}]/ord]];
Dimension[s_,x_]:=Module[{v,VL=VertexList,u,ord,size,e},
  v=VL[s];ord=Length[v];e=EdgeList[s]; size= Length[e];
  If[ord==0,u={},u=UnitSphere[s,x]];
  If[size==0,0,If[Length[VL[u]]==0,0,1]+Dimension[u]]];
DimensionList[s_]:=Module[{},vl=VertexList[s];
  Table[Dimension[s,vl[[k]]],{k,Length[vl]}]];
DimensionList[RandomGraph[{15,40}]]
```

Here is an example, which shows how to get the dimensions of all 13 Archimedean solids. This appeared in one of the figures of [2].

*Date*: December 23, 2011.

1991 *Mathematics Subject Classification.* Primary: 05C80,05C82,05C10,90B15,57M15 .

*Key words and phrases.* Random Graph Theory, Complex Networks, Graph Dimension, Graph Curvature, Euler characteristic.

```
PD=PolyhedronData; a=PD["Archimedean"]; Table[
s=UndirectedGraph[Graph[PD[a[[k]],"SkeletonRules"]]];
{a[[k]],"with dimension",Dimension[s]},{k,Length[a]}]
```

Now we compute the sum of all dimensions over all subgraphs of a complete graph with 5 vertices by brute force. It is $1451 = d_5(1/2)2^{\binom{5}{2}}$. This computations done for $p = 1/2$ first triggered the paper [3] because we found it intriguing that the sum was always an integer. It suggested that there is more structure behind dimension.

```
F[n_,p_]:=Module[{v=Range[n],f,e,m,g,u},
 g[k_,l_]:=p^l (1-p)^(Binomial[n,2]-l);
 e=Flatten[Table[v[[i]]->v[[j]],{i,n},{j,i+1,n}]];
 f=Subsets[e]; m=Length[f]; d=Map[Length,f];
 u=Table[Dimension[UndirectedGraph[Graph[v,f[[k]]]]],{k,m}];
 Sum[u[[k]] g[k,d[[k]]],{k,Length[u]}]];F[5,0.5]*2^10
```

The next routine computes the expected dimension for graphs with $n$ nodes and percolation probability $p$ using the formulas we know. Fixing $p$ allows to compute the expectations of dimension, where the number of vertices is large. The number of computation steps is $O(n^2)$.

```
G[n_,p_]:=Module[{r,b=Binomial,g,F,l},l[d_]:=Length[d]-1;
g[k_,m_]:=N[b[m,k] p^k(1-p)^(m-k)]; r={-1,0};
F[d_]:=Module[{m=l[d]},1+Sum[d[[k+1]]g[k,m],{k,0,m}]];
Do[r=Append[r,N[F[r]]],{n}];r[[n+1]]]; G[5,0.5]*2^10
```

Here is the formula for the average Euler characteristic in $G(n, 1/2)$

```
h[n_,k_]:=Binomial[n,k] p^Binomial[k,2];
H[n_]:=Sum[(-1)^(k+1) h[n,k],{k,1,n}];
Table[H[n],{n,1,8}] /. p->1/2
```

Here is the code to compute the Euler characteristic of an arbitrary graph

```
Cliques[K_,k_]:=Module[
  {n,m=Length[EdgeList[K]],s,u,V=VertexList[K],U},
  s=Subsets[V,{k,k}]; n=Length[V]; Bi=Binomial;
  If[k==1,u=n,If[k==2,u=m, u=Sum[
    If[Length[EdgeList[Subgraph[K,s[[j]]]]]==Bi[k,2],1,0],
    {j,Length[s]}]];u];
EulerCharacteristic[K_]:=Sum[(-1)^(k-1)
  Cliques[K,k],{k,1,Length[VertexList[K]]}];
EulerCharacteristic[KnightTourGraph[5, 3]]
```

And here is the code to compute the curvature at a point as well as the list of curvatures. Adding them up gives the Euler characteristic:

```
Curvature[s_,v_]:=Module[{s1,k,r,u},s1=UnitSphere[s,v];
  vl=VertexList[s1];n=Length[vl];r=Table[(-1)^k/(k+1),{k,n}];
  u=Table[Cliques[s1,k],{k,n}]; 1+u.r];
Curvatures[s_]:=Module[{},vl=VertexList[s];
```

```
  Table[Curvature[s,vl[[k]]],{k,Length[vl]}]]
Curvatures[KnightTourGraph[5, 3]]
```

Here are the signature functions, which are probabilistic invariants of a host graph. These polynomials are then computed for a cyclic graph $C_5$ to illustrate its use:

```
W[n_,m_,p_]:=p^m (1-p)^(n-m); EC=EulerCharacteristic;
F[s_]:=Module[{e,g,v,n}, e=EdgeRules[s]; g=Subsets[e];
  v=VertexList[s]; n=Length[e]; Sum[W[n,Length[g[[k]]],p]*
  Dimension[UndirectedGraph[Graph[v,g[[k]]]]],{k,Length[g]}]];
G[s_]:=Module[{e,g,v,n}, e=EdgeRules[s]; g=Subsets[e];
  v=VertexList[s]; n=Length[e]; Sum[W[n,Length[g[[k]]],p]*
  EC[UndirectedGraph[Graph[v,g[[k]]]]],{k,Length[g]}]];
F[CycleGraph[5]]
G[CycleGraph[5]]
```

## References


[1] O. Knill. A discrete Gauss-Bonnet type theorem. *Elemente der Mathematik (to appear)*, 2010. http://arxiv.org/abs/1009.2292.

[2] O. Knill. A graph theoretical Gauss-Bonnet-Chern theorem. http://arxiv.org/abs/1111.5395, 2011.

[3] O. Knill. *On the Dimension and Euler characteristic of random graphs.* http://arxiv.org/abs/1111.5395, 2011.



*E-mail address*: knill@math.harvard.edu

Department of Mathematics, Harvard University, Cambridge, MA, 02138