DESIGN STRUCTURE AND ITERATIVE RELEASE ANALYSIS OF SCIENTIFIC SOFTWARE

Ahmed Tahsin Zulkarnine

Supervisor: Dr. Shahadat Hossain

Thesis Defence
MSc. In Computer Science

May 23rd , 2012

Department of Math and Computer Science
University of Lethbridge, Alberta, Canada

Outline

- Objectives
- Methodology
- Experimental Results
- Iterative Release Comparison
- Release Cost Estimation
- Findings and future work

Scientific Research Software

- General-purpose Commercial Software
  - Employ formal methods from software engineering discipline
  - Well known problem domains
  - Trained engineers familiar with tested 'best-practices'
- Scientific Research Software
  - 'Proof-of-concept' code vs 'Large Scale simulation'
  - Designed by highly trained scientists
  - Focuses on narrow and highly specialized domain.

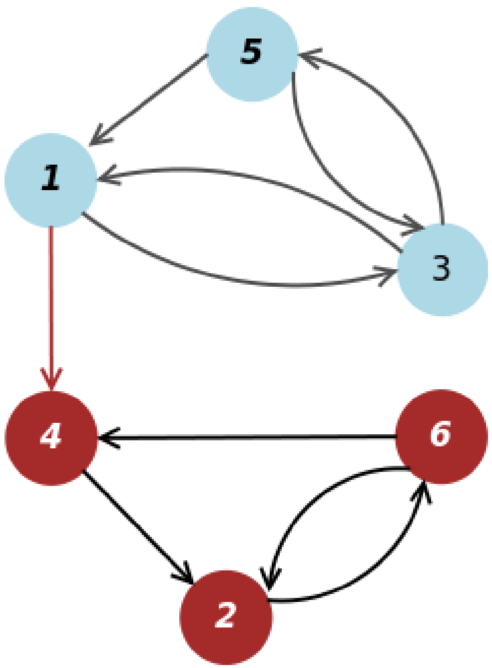
Objectives

- Study and analyze design structure of scientific software systems with suitable design structural metrics and DSM to investigate:
  - Modularity
  - Maintainability
  - Extensibility, etc
- In our research we have chosen the open source scientific computing software that focuses on below application domains:
  - AD (Automatic Differentiation)
  - LP (Linear Programming)
  - MIP (Mixed Integer Programming)

Modelling Dependencies with DSM

- DSM : Square matrix with identical row and column.
- DSM has been used to capture and analyze the dependencies of the software.
- Call graph extractor used to extract static source code dependencies.
- User defined functions are basic design elements.

Call graphs to DSM



		1	2	3	4	5	6
Vertex 1	1	X		X	X		
Vertex 2	2		X				X
Vertex 3	3	X		X		X	
Vertex 4	4		X		X		
Vertex 5	5	X		X		X	
Vertex 6	6		X		X		x

- Dependencies between two user defined functions are denoted by an off-diagonal mark in DSM.

Structural Metrics

- **Characteristic path length :**

$$l = \frac{\sum_{i \neq j} d_{ij}}{N(N-1)}$$

where d_{ij} : length of the shortest path connecting the nodes i and j

  - This provides us information regarding the efficiency of software

- **Clustering co-efficient:**

$$C = \frac{1}{N}\sum_{i=1}^{N} C_i$$

where $$C_i = \frac{2 * n_i}{k_i(k_i - 1)}$$

denotes C_i : the clustering co-efficient of node i
k_i : number of nodes i is adjacent to
n_i : actual number of edges between node i's adjacent nodes.

  - This provides us information regarding the modularity of the software

Structure Metrics (contd..)

- **Average nodal degree:** $$k = \frac{1}{N}\sum_{i=1}^{N} k_i$$

  where k_i : number of nodes adjacent to node i

  - This provides us information regarding the degree of dependencies of system elements.

- **Propagation cost:** $$\frac{1}{N}\sum_{i=1}^{N} p_i$$

  where p_i: number of nodes reachable from node i

  - This provides us information regarding the sensitivity of the system elements

- **Centrality measure:** An index that measures the centrality of a node by the number of shortest path in the graph containing that node.

  - This provide us information regarding the global information of software.

Experiments

- Experimental Environment
  - **Machine:** HP P6510 F
  - **Processor:** AMD Athlon X4 630 Quad Core processors
  - **Operating System:** Ubuntu 10.04
- We studied and analyzed the following 4 software tools:
  - **ADOL-C**: An AD software
  - **BCP:** A MIP software
  - **CppAD**: An AD
  - **DyLP :** A LP software

Partitioning the DSM

- Partitioning : Reordering of the DSM rows and columns so that new DSM contains minimum number of feedback marks.

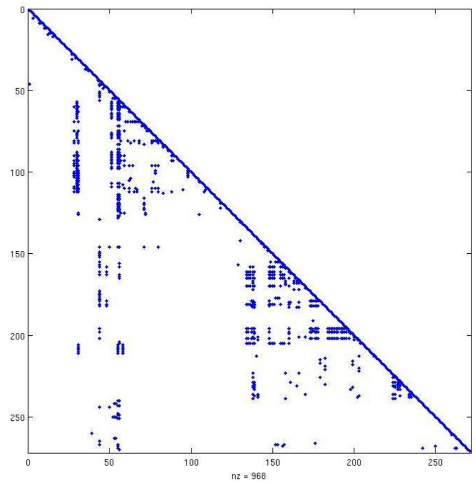
- Partitioned DSM allow us to identify
  - Sequential tasks
  - Parallel tasks
  - Iterative tasks.

- We used Tarzan's algorithm using sparse data structure.

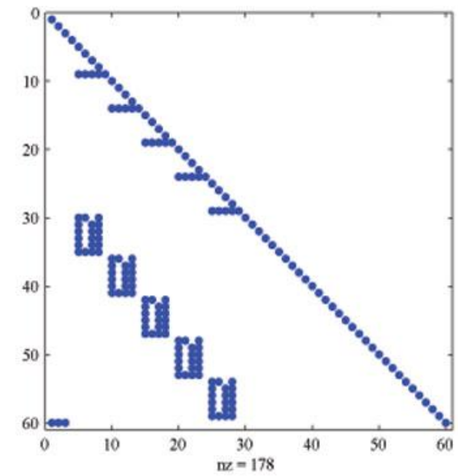
Matrix Name	# of vertices, N	# of components	Boost Timing (s)	Our Timing (s)
NotreDame	325729	231666	1.6812	0.318
amazon0601	403394	1588	11.08	2.418
StanfordBerkeley	683446	109238	22.568	3.80

Partitioned DSM

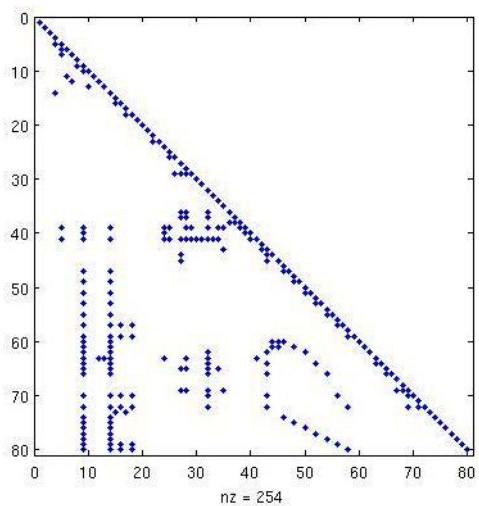
ADOL-C



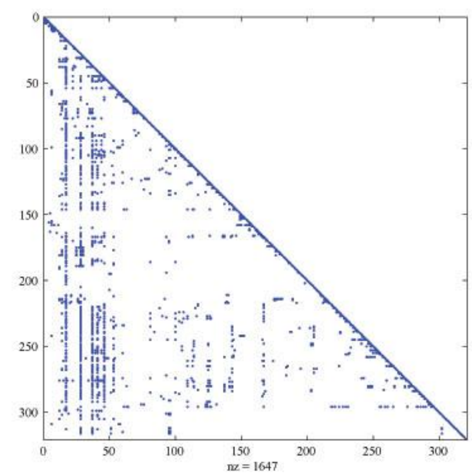
Bcp



CppAD



DyLP

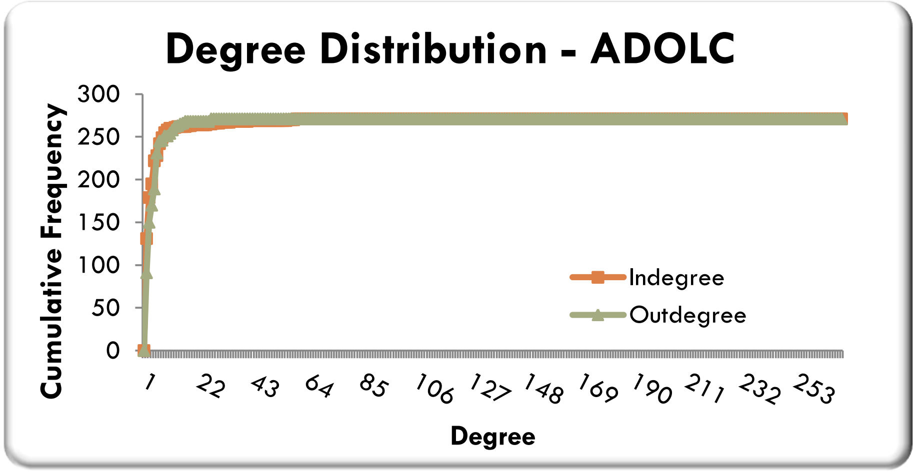


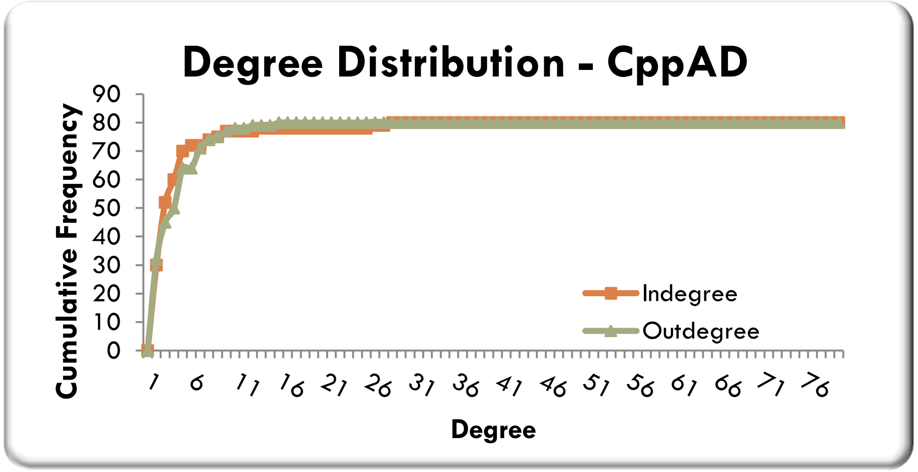
Structural Properties & Metrics

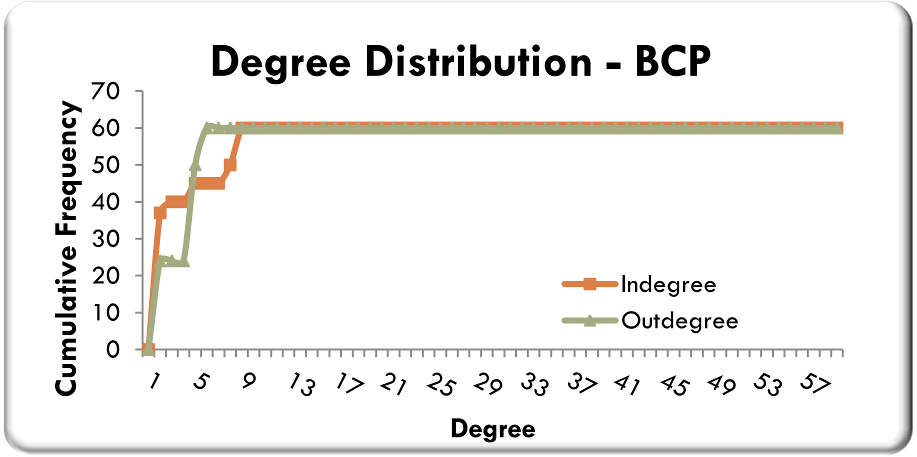
Software	Nodes		Directed Edges		Sparsity
	Files	User functions	Files	User functions	
ADOL-C	60	271	66	703	0.95 %
Bcp	45	60	29	118	3.33 %
CppAD	66	80	67	175	2.74 %
DyLP	51	315	333	1321	1.34 %

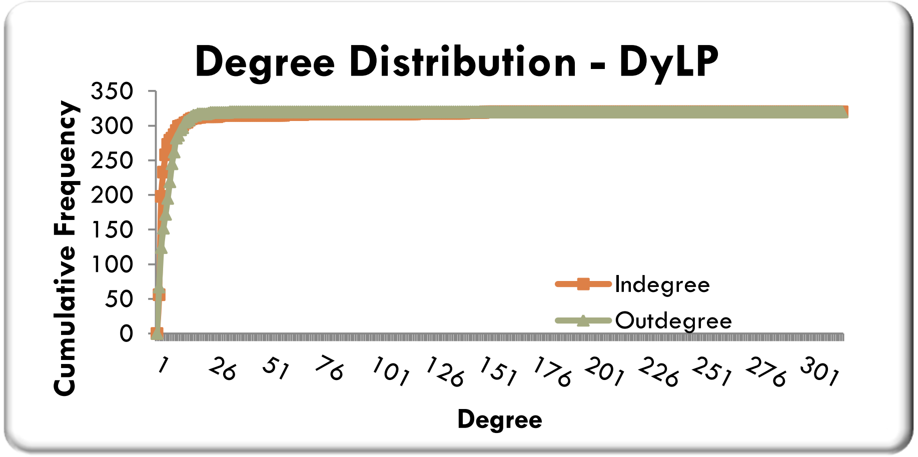
Software	Characteristic Path length, l	Clustering co-efficient, C	Nodal Degree	Propagation Cost (%)
ADOL- C	2.05005	0.080382	5.18819	3.41635
Bcp	0.264972	0	3.93333	4.94444
CppAD	2.37373	0.0342364	4.375	6.64062
DyLP	2.67967	0.245807	8.3873	5.17208

Degree Distribution









Power Law & Scale Free Networks

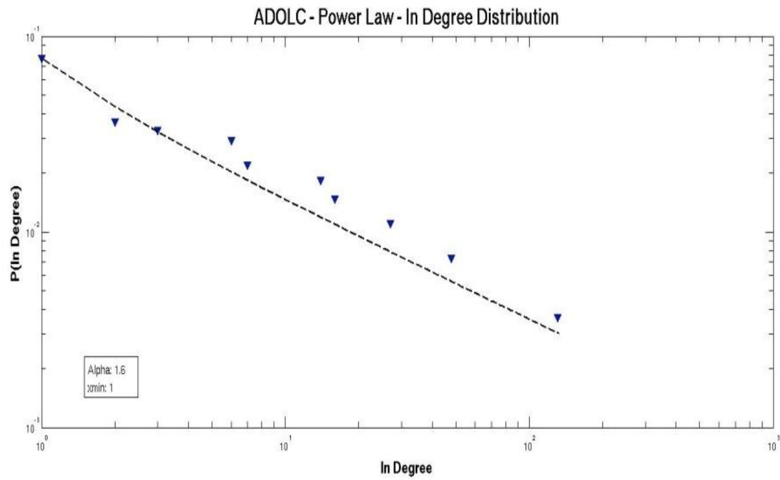
- Power Law: $$p(x) \propto x^{-\alpha}$$

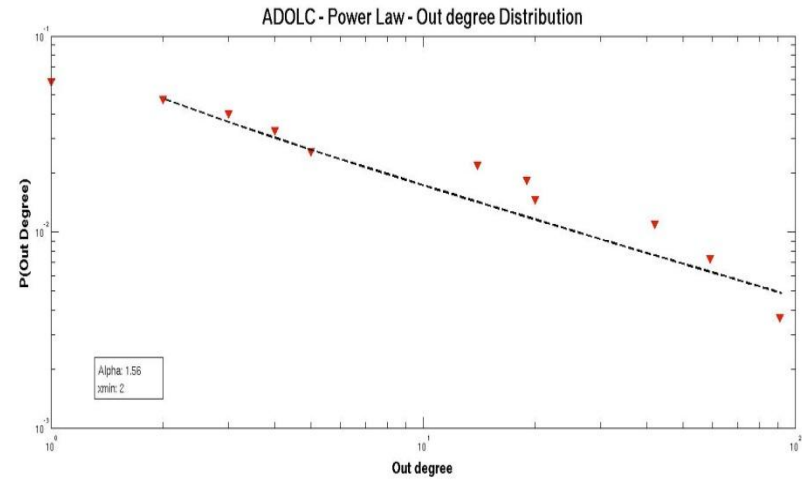
  Where α is the scaling factor

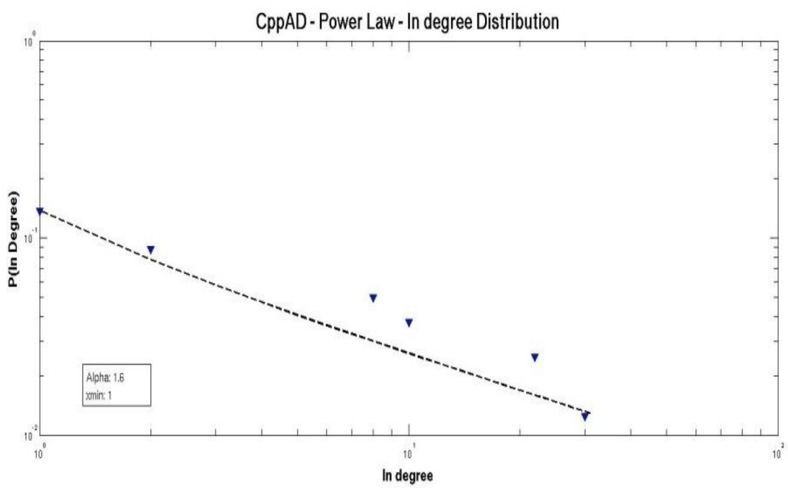
  Power law applies for values greater than $x_{\min}$

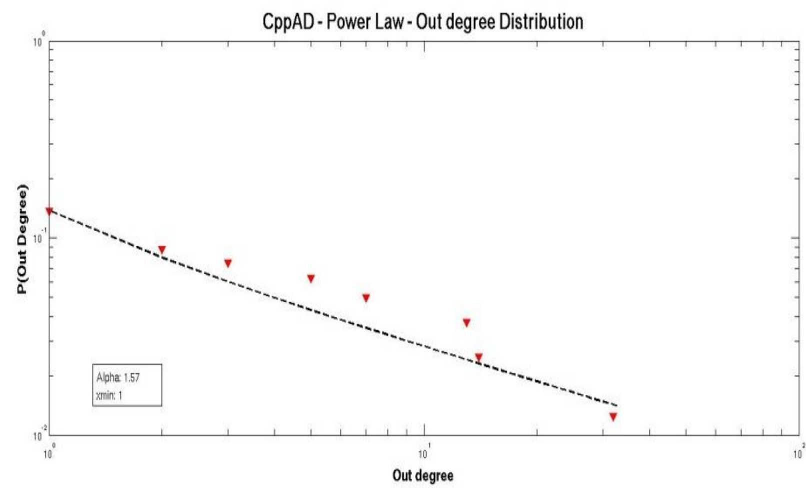
- Scale free networks: Networks with power law degree distribution.
- Scale free networks characteristics:
  - Contains Hubs
  - Network Robustness to failure

Power Law Analysis - Degree Distribution

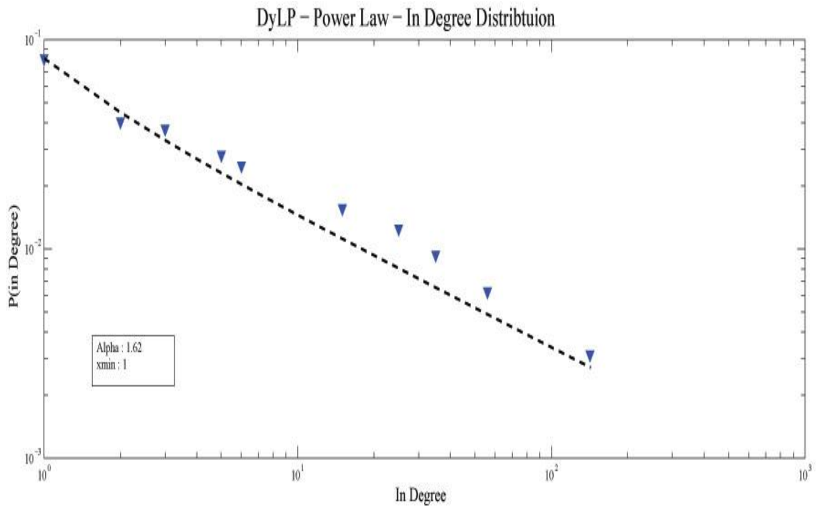


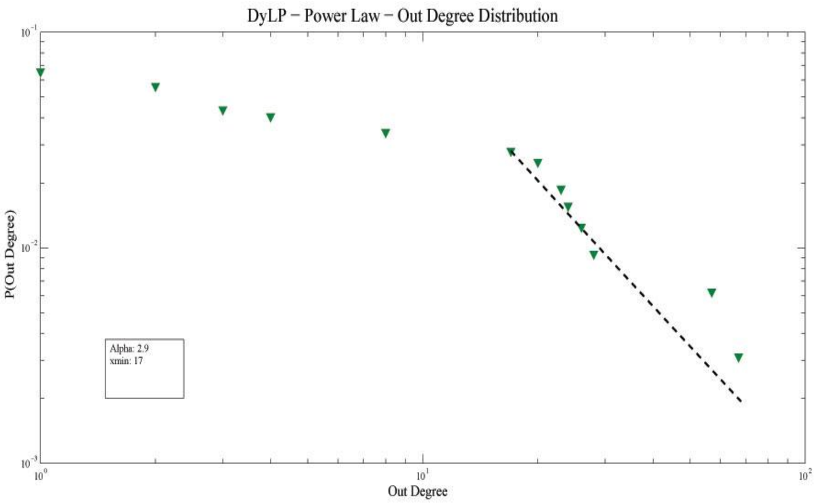






Power Law Analysis - Degree Distribution (contd)





	x_{min}	*a*	*p*
ADOLC – In degree	1	1.6	0.267
ADOLC – Out degree	2	1.56	0.286
CppAD – In degree	1	1.6	0.388
CppAD – Out degree	1	1.7	0.493
DyLP – In degree	1	1.62	0.10
DyLP – Out degree	17	2.9	0.12

Iterative Release Analysis

- New customer requirements necessitate iterative releases.
- Feature enhancement, improving computational efficiency, etc drives iterative release in scientific software.
- Iterative release analysis allows us to investigate the changes in structural properties and metrics of scientific software releases.

Iterative Release Analysis Results

University of Lethbridge

Software	Compared releases	Changes in Release	Maximum New Elements added	Change in Central function
ADOLC	10	1 major, 4 minor	116	Yes
Bcp	**7**	**No Change**	**0**	**No**
CppAD	10	1 major, 1 minor	33	No
DyLP	10	1 major, 2 minor	65	No

- ADOLC :

ADOLC Versions	Characteristic Path length, l	Clustering co-efficient, C	Nodal Degree	Number of Components	Propagation Cost (%)
V 1.9	3.36142	0.107767	6.55873	315	3.53238
V 1.10.2	3.25725	0.106083	6.48125	320	3.43262
V 2.1.2	2.04977	0.0803177	5.19557	271	3.42316
V 2.1.4	2.0611	0.0777237	5.13971	272	3.38452
V 2.1.12	2.20408	0.0803834	5.26236	263	3.58831
V 2.2.1	2.16312	0.0799071	5.21509	265	3.50303

Iterative Release Analysis Results(contd..)

- CppAD :

CppAD Version	Characteristic Path length, l	Clustering co-efficient, C	Nodal Degree	Number of Components	Propagation Cost (%)
V 110101.0	2.35228	0.0363174	4.31579	76	6.95983
V 110308	2.37373	0.0342364	4.375	80	6.64062
V 111103	2.44108	0.0265913	3.80583	103	9.6239

- DyLP

DyLP Version	Characteristic Path length, l	Clustering co-efficient, C	Nodal Degree	Number of Components	Propagation Cost (%)
V 1.3.0	2.67341	0.261488	8.18729	299	5.51112
V 1.4.0	2.6719	0.258967	8.22074	299	5.51784
V 1.5.0	2.67967	0.245807	8.3873	315	5.17208
V 1.7.0	2.63341	0.24186	8.29375	320	5.04883

Iterative Release Cost

- Total implementation cost of release *n*,

$$Tc_n = Ic_n + Rc_n$$

- Ic_n is the summation of the cost to implement all the new architectural element .
- We assumed implementation cost of each architectural element is 1.
- Release rework cost, Rc_n is calculated using:

$$Rc_n = \sum_{j=1}^{m} I[j] \times P_{n-1}$$

where

m : No. of new elements added

$I[j]$: No. of old version dependency these new element j have.

P_{n-1} : propagation cost of previous release n-1.

Release Cost Estimation

- ADOL-C :

Old Version	New Version	No. of New Elements	p_{n-1}	Ic_n	Rc_n	Tc_n
V1.9	V1.10.0	6	0.0353238	6	0.52986	**6.52986**
V1.10.0	V2.1.0	116	0.0343262	116	13.696	**129.696**
V2.1.0	V2.1.4	1	0.0342316	1	0.03423	**1.03423**
V2.1.4	V2.1.12	1	0.0338452	1	0.0338452	**1.0338452**
V2.1.12	V2.2.1	7	0.0358831	7	0.28707	**7.28707**

- CppAD:

Old Version	New Version	No. of New Elements	p_{n-1}	Ic_n	Rc_n	Tc_n
V 110101.0	V 110308	4	0.0695983	4	0.34799	**4.34799**
V 110308	V 111103	33	0.0664062	33	3.5859	**36.5859**

- DyLP:

Old Version	New Version	No. of New Elements	p_{n-1}	Ic_n	Rc_n	Tc_n
V 1.3.0	V 1.4.0	65	0.0551112	65	12.896	**77.896**
V 1.4.0	V 1.5.0	21	0.0551784	21	4.2487	**25.2487**
V 1.5.0	V 1.7.0	6	0.0517208	6	1.08614	**7.08614**

Findings

Properties	General Purpose Commercial Software	Scientific Research Software
Characteristic path Length	2.8 - 3.2	2.2 - 2.7
Clustering co-efficient	0.2 - 0.45	0 - 0.2
Average Nodal Degree	7 - 20	3 – 8
Propagation Cost	5 - 17	3 - 7
Feedback Marks	Yes	No

- Iterative release analysis indicates
  - Clustering co-efficient decreased across the releases.
  - The most central function remained the same in all the releases.
  - The clustering co-efficient plays a vital role in the determination of release rework cost.

Software	Old Version	New Version	No. of New Elements	Clustering co-efficient	Rc_n
ADOL-C	V1.10.0	V2.1.0	116	0.106083	13.696
DyLP	V 1.3.0	V 1.4.0	65	0.258967	12.896

Software	Old Version	New Version	No. of New Elements	Clustering co-efficient	Rc_n
CppAD	V 110308	V 111103	33	0.0363174	3.5859
DyLP	V 1.4.0	V 1.5.0	21	0.258967	4.2487

Future Work

- There can a number of extension to this work
  - How to estimate the integration effort
  - Domain specific structural metrics

Thank You