Parallel Implementation of the Polyhedral Homotopy Method

Jan Verschelde and Yan Zhuang

email: jan@math.uic.edu   yzhuan1@math.uic.edu

0. Motivation

Apply polyhedral homotopies to polynomial system arising in mechanisms design.
Special structure: not so sparse but still mixed volume < degree bounds
Five basic joints:
Revolute: R Prismatic: P Cylindric: C Universal: T Spherical: S

Basic chains	Surface	Total deg	LPD bound	Mixvol
PRS	elliptic cylinder	2,097,152	247,968	125,888
RRS	circular torus	2,097,152	868,352	474,112
RRS	general torus	4,194,304	448,702	226,512

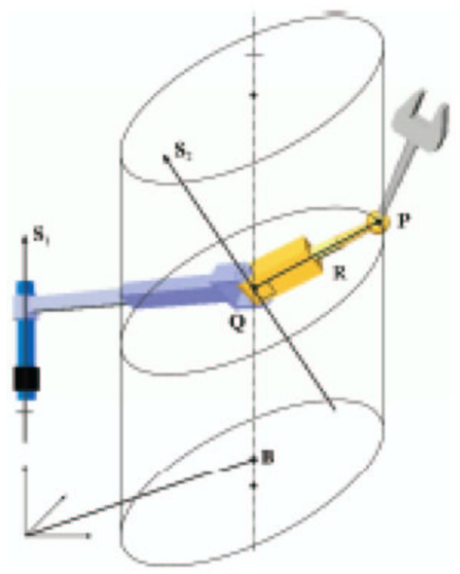

Figure 4.4: The elliptic cylinder reachable by a PRS serial chain.

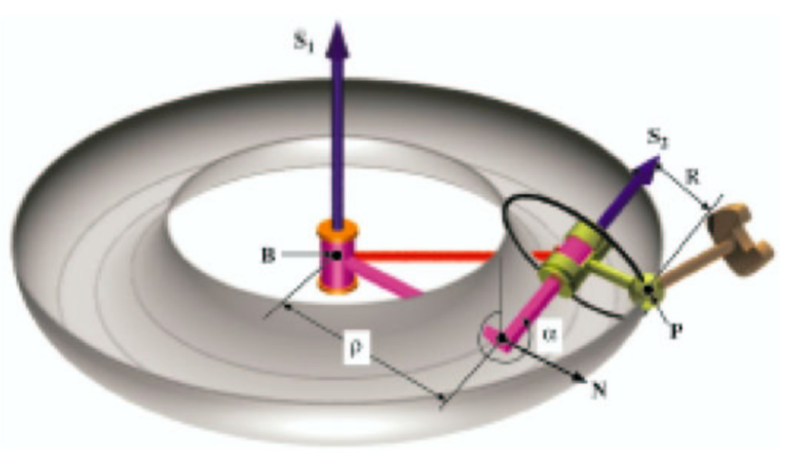

Figure 4.8: The general torus reachable by the wrist center of an RRS serial chain.

1. Three Stages to Solve a Polynomial system $f(\mathbf{x}) = 0$

1. Compute mixed volume of the Newton polytopes spanned by the supports of f
2. Solve a random coefficient start system $g(\mathbf{x}) = 0$ which has the same monomials as f with random coefficients and has exactly mixed volume isolated solutions.
3. Use $(1-t)g(\mathbf{x}) + tf(\mathbf{x}) = \mathbf{0}$ to solve $f(\mathbf{x}) = \mathbf{0}$.

Stages 2 and 3 are computationally most intensive ($1 \ll 2 < 3$).

References

[1] H.-J. Su, J.M. McCarthy, and L.T. Watson. **Generalized linear product homotopy algorithms and the computation of reachable surfaces.** *ASME Journal of Information and Computer Sciences in Engineering, 2004.*

[2] H.-J. Su, C.W. Wampler, and J.M. McCarthy. **Geometric design of cylindric PRS serial chains** *ASME Journal of Mechanical Design, 2004.*

2. Static Work Load Balancing

Since polyhedral homotopies solve a generic system, we expect every path to take the same amount of work.

Algorithm: Sketch of a parallel version of polyhedral homotopies.
Input: $\triangle_\omega$, $G(\mathbf{x}) = \mathbf{0}$. *mixed-cell configuration and generic system*
Output: $G^{-1}(\mathbf{0})$. *all solutions to $G(\mathbf{x}) = \mathbf{0}$*

Manager		Workers	
read input file			
broadcast data	→	receive data	*data = system and lifting*
distribute cells	→	receive cells	*static workload distribution*
		track paths	*compute solutions*
collect solutions	←	send solutions	
write to file			

We introduce the static distribution of the cells with an example.

manager	worker 1	worker 2	worker 3
Vol(cell 1) $= 5$	cell 1 : 5		
Vol(cell 2) $= 4$	cell 2 : 4		
Vol(cell 3) $= 4$	cell 3 : 4		
Vol(cell 4) $= 6$	cell 4 : 1	cell 4 : 5	
Vol(cell 5) $= 7$		cell 5 : 7	
Vol(cell 6) $= 3$		cell 6 : 2	cell 6 : 1
Vol(cell 7) $= 4$			cell 7 : 4
Vol(cell 8) $= 8$			cell 8 : 8
total #paths : 41	track paths : 14	track paths : 14	track paths : 13

Wall time for start systems to solve the cyclic n-roots problems, using a cluster configuration with 13 workers, with static load distribution.

Problem	#Paths	CPU Time
cyclic 6-roots	156	0.19m
cyclic 7-roots	924	0.30m
cyclic 8-roots	2,560	0.78m
cyclic 9-roots	11,016	3.64m
cyclic 10-roots	35,940	21.33m
cyclic 11-roots	184,756	2h 39m
cyclic 12-roots	500,352	24h 36m

More References

[3] H.-J. Su, J.M. McCarthy, M. Sosonkina, and L.T. Watson. **Algorithm 8xx: POLSYS_GLP:A parallel general linear product homotopy code for solving polynomial systems of equations.** *To appear in ACM Trans. Math. Softw.*

[4]T.Gao, T.Y.Li and M.Wu **Algorithm 846: MixedVol: A software package for mixed volume computation.** *ACM Trans. Math. Softw.,2005.*

[5]Takayuki Gunji, Sunyoung Kim, Katsuki Fujisawa and Masakazu Kojima **PHoMpara-Parallel Implementation of Polyhedral Homotopy Continuation Method for Polynomial Systems.** *Computing Volume 77, Issue 4 2006.*

[6]J. Verschelde **Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation.** *ACM Trans. Math. Softw.,1999.*

[7]J. Verschelde and Y. Zhuang **Parallel implementation of the polyhedral homotopy method.** *Proceedings of HPSEC 2006.*

3. Dynamic Work Load Balancing

For polynomials which are not so sparse as mechanisms design problems, dynamic load balancing is needed.

Algorithm: Dynamic distribution of cells executed by the manager.

Input: $\triangle_\omega$, V, p. *mixed-cell configuration, volume, #processors*
Output: $G^{-1}(\mathbf{0})$. *all solutions to $G(\mathbf{x}) = \mathbf{0}$*
if $\#\triangle_\omega \leq p$ **then** *distribute path by path*
  distribute V paths;
else
  distribute the first $\#\triangle_\omega - 1$ cells; *distribute cell by cell*
  distribute the last cell; *distribute path by path*
end if.

Algorithm: Dynamic distribution of cells executed by all the workers.

Input: $G(\mathbf{x}) = \mathbf{0}$, V. **Output:** a subset of $G^{-1}(\mathbf{0})$.
do
  receive C; *receive cell*
  create polyhedral homotopy $\widehat{G}_C(\mathbf{y}, s) = \mathbf{0}$; *perform coordinate transformation*
  solve the start system $\widehat{G}_C(\mathbf{y}, 0) = \mathbf{0}$; *one linear system to solve*
  track paths; *track $\mathrm{Vol}(C)$ paths or just one path*
  send solutions to manager; *message to manager reporting results*
end do.

The speedup on the cyclic 7-roots problem for an increasing number of workers

#workers	Static versus Dynamic on our cluster				Dynamic on argo	
	Static	Speedup	Dynamic	Speedup	Dynamic	Speedup
1	50.7021	–	53.0707	–	29.2389	–
2	24.5172	2.1	25.3852	2.1	15.5455	1.9
3	18.3850	2.8	17.6367	3.0	10.8063	2.7
4	14.6994	3.4	12.4157	4.2	7.9660	3.7
5	11.6913	4.3	10.3054	5.1	6.2054	4.7
6	10.3779	4.9	9.3411	5.7	5.0996	5.7
7	9.6877	5.2	8.4180	6.3	4.2603	6.9
8	7.8157	6.5	7.4337	7.1	3.8528	7.6
9	7.5133	6.8	6.8029	7.8	3.6010	8.1
10	6.9154	7.3	5.7883	9.2	3.2075	9.1
11	6.5668	7.7	5.3014	10.0	2.8427	10.3
12	6.4407	7.9	4.8232	11.0	2.5873	11.3
13	5.1462	9.8	4.6894	11.3	2.3224	12.6

Wall time for mechanism design problems on our cluster and argo.

Surface	Bounds on #Solutions			dynamic load distribution	
	Total deg	LPD bound	Mixvol	our cluster	time on argo
elliptic cylinder	2,097,152	247,968	125,888	11h 33m	6h 12m
circular torus	2,097,152	868,352	474,112	7h 17m	4h 3m
general torus	4,194,304	448,702	226,512	14h 15m	6h 36m

4. Conclusion

A static work load distribution provides already a decent speedup of the polyhedral homotopies on a cluster computer for "small" or very sparse system. However, for polynomial systems which are not so sparse as the mechanisms design problems, dynamic load balancing is needed.