Parallel Implementation of a Subsystem-by-Subsystem Solver

Yun Guan   Jan Verschelde

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science
http://www.math.uic.edu/~guan   http://www.math.uic.edu/~jan
guan@math.uic.edu   jan@math.uic.edu

HPCS 2008 – High Performance Computing Symposium
Université Laval, Québec City, Canada, June 9-11, 2008

Outline

1. Solving Polynomial Systems with PHCpack
   - solving polynomial systems
   - numerical homotopy continuation methods
   - PHCpack: a package for Polynomial Homotopy Continuation
   - PHCwulf.py: interactive client/server computing
2. Running Diagonal Homotopies in Parallel
   - witness sets represent positive dimensional solution sets
   - parallel implementation uses jumpstarting
3. Parallel Implementation of the Solver
   - a subsystem-by-subsystem solver
   - divide and conquer algorithms
   - software & equipment
   - experimental results

Solving Polynomial Systems

numerical algebraic geometry: isolated solutions and components of solutions

Two families of *benchmark* polynomial systems:

1. **katsura**, from a magnetism problem in physics:

$$\left\{ \begin{array}{ll} \displaystyle\sum_{j=-n}^{n} u_{j,n} u_{k-j,n} - u_{k,n} = 0, & k = 1, 2, \ldots, n \\ \displaystyle\sum_{j=-n}^{n} u_{j,n} - 1 = 0, & \text{for } j < 0 : u_j = u_{-j} \end{array} \right.$$

   The solution set consists of 2^n isolated points.

2. **adjacent minors**, from algebraic statistics

$$x_{1,k} x_{2,k+1} - x_{2,k} x_{1,k+1} = 0, \quad k = 1, 2, \ldots, n.$$

   The solution set has dimension $n + 1$ and degree 2^{n-1}.

The Total Degree Homotopy

to intersect two quadrics

Solve by considering a simpler system in a homotopy

$$\underbrace{\left(\left\{ \begin{array}{r} x_1^2 + x_2 - 3 = 0 \\ x_1 + 0.125 x_2^2 - 1.5 = 0 \end{array} \right. \right)}_{\text{target system}} t + \gamma \underbrace{\left(\left\{ \begin{array}{r} x_1^2 - 1 = 0 \\ x_2^2 - 1 = 0 \end{array} \right. \right)}_{\text{start system}} (1 - t) = \mathbf{0}$$

where t goes from 0 to 1, and $\gamma \in \mathbb{C}$ is a random constant.

For almost all choices of $\gamma \in \mathbb{C}$, every isolated solution of multiplicity m is reached by exactly m solution paths.

also called "the gamma trick"

If we take $\gamma = 1$, then at $t \approx 0.92$ singular solutions occur.

The Total Degree Homotopy

to intersect two quadrics

Solve by considering a simpler system in a homotopy

$$\underbrace{\left(\left\{ \begin{array}{r} x_1^2 + x_2 - 3 = 0 \\ x_1 + 0.125 x_2^2 - 1.5 = 0 \end{array} \right. \right)}_{\text{target system}} t + \gamma \underbrace{\left(\left\{ \begin{array}{r} x_1^2 - 1 = 0 \\ x_2^2 - 1 = 0 \end{array} \right. \right)}_{\text{start system}} (1-t) = \mathbf{0}$$

where t goes from 0 to 1, and $\gamma \in \mathbb{C}$ is a random constant.

For almost all choices of $\gamma \in \mathbb{C}$, every isolated solution of multiplicity m is reached by exactly m solution paths.

also called "the gamma trick"

If we take $\gamma = 1$, then at $t \approx 0.92$ singular solutions occur.

The Total Degree Homotopy

to intersect two quadrics

Solve by considering a simpler system in a homotopy

$$\underbrace{\left(\left\{ \begin{array}{r} x_1^2 + x_2 - 3 = 0 \\ x_1 + 0.125 x_2^2 - 1.5 = 0 \end{array} \right. \right)}_{\text{target system}} t + \gamma \underbrace{\left(\left\{ \begin{array}{r} x_1^2 - 1 = 0 \\ x_2^2 - 1 = 0 \end{array} \right. \right)}_{\text{start system}} (1-t) = \mathbf{0}$$

where t goes from 0 to 1, and $\gamma \in \mathbb{C}$ is a random constant.

For almost all choices of $\gamma \in \mathbb{C}$, every isolated solution of multiplicity m is reached by exactly m solution paths.

also called "the gamma trick"

If we take $\gamma = 1$, then at $t \approx 0.92$ singular solutions occur.

The Total Degree Homotopy

to intersect two quadrics

Solve by considering a simpler system in a homotopy

$$\underbrace{\left(\left\{ \begin{array}{r} x_1^2 + x_2 - 3 = 0 \\ x_1 + 0.125x_2^2 - 1.5 = 0 \end{array} \right. \right)}_{\textbf{target system}} t + \gamma \underbrace{\left(\left\{ \begin{array}{r} x_1^2 - 1 = 0 \\ x_2^2 - 1 = 0 \end{array} \right. \right)}_{\textbf{start system}} (1-t) = \mathbf{0}$$

where t goes from 0 to 1, and $\gamma \in \mathbb{C}$ is a random constant.

For almost all choices of $\gamma \in \mathbb{C}$, every isolated solution of multiplicity m is reached by exactly m solution paths.

also called "the gamma trick"

If we take $\gamma = 1$, then at $t \approx 0.92$ singular solutions occur.

The Total Degree Homotopy

to intersect two quadrics

Solve by considering a simpler system in a homotopy

$$\underbrace{\left(\left\{ \begin{array}{r} x_1^2 + x_2 - 3 = 0 \\ x_1 + 0.125 x_2^2 - 1.5 = 0 \end{array} \right. \right)}_{\text{target system}} t + \gamma \underbrace{\left(\left\{ \begin{array}{r} x_1^2 - 1 = 0 \\ x_2^2 - 1 = 0 \end{array} \right. \right)}_{\text{start system}} (1-t) = \mathbf{0}$$

where t goes from 0 to 1, and $\gamma \in \mathbb{C}$ is a random constant.

For almost all choices of $\gamma \in \mathbb{C}$, every isolated solution of multiplicity m is reached by exactly m solution paths.

also called "the gamma trick"

If we take $\gamma = 1$, then at $t \approx 0.92$ singular solutions occur.

The Total Degree Homotopy

to intersect two quadrics

Solve by considering a simpler system in a homotopy

$$\underbrace{\left(\left\{\begin{array}{r} x_1^2 + x_2 - 3 = 0 \\ x_1 + 0.125x_2^2 - 1.5 = 0 \end{array}\right.\right)}_{\textbf{target system}} t + \gamma \underbrace{\left(\left\{\begin{array}{r} x_1^2 - 1 = 0 \\ x_2^2 - 1 = 0 \end{array}\right.\right)}_{\textbf{start system}} (1-t) = \mathbf{0}$$

where t goes from 0 to 1, and $\gamma \in \mathbb{C}$ is a random constant.

For almost all choices of $\gamma \in \mathbb{C}$, every isolated solution of multiplicity *m* is reached by exactly *m* solution paths.

also called "the gamma trick"

If we take $\gamma = 1$, then at $t \approx 0.92$ singular solutions occur.

The Total Degree Homotopy

to intersect two quadrics

Solve by considering a simpler system in a homotopy

$$\underbrace{\left(\left\{ \begin{array}{r} x_1^2 + x_2 - 3 = 0 \\ x_1 + 0.125 x_2^2 - 1.5 = 0 \end{array} \right. \right)}_{\textbf{target system}} t + \gamma \underbrace{\left(\left\{ \begin{array}{r} x_1^2 - 1 = 0 \\ x_2^2 - 1 = 0 \end{array} \right. \right)}_{\textbf{start system}} (1 - t) = \mathbf{0}$$

where t goes from 0 to 1, and $\gamma \in \mathbb{C}$ is a random constant.

For almost all choices of $\gamma \in \mathbb{C}$, every isolated solution of multiplicity *m* is reached by exactly *m* solution paths.

also called "the gamma trick"

If we take $\gamma = 1$, then at $t \approx 0.92$ singular solutions occur.

Software Systems

many recent programming efforts

Starring in alphabetical order:

- **Bertini**, first released in Fall 2006, by D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler.
- **HOM4PS-2.0** by T.L. Lee, T.Y. Li, and C.H. Tsai (2007), extends **HOM4PS** by T. Gao and T.Y. Li.
- **PHoMpara** by T. Gunji, S. Kim, K. Fujisawa, and M. Kojima (2006) is a parallel version of **PHoM** by T. Gunji, S. Kim, M. Kojima, A. Takeda, K. Fujisawa and T. Mizutani (2004).
- **POLSYS_GLP** is Algorithm 857 of ACM TOMS (2006) by H.-J. Su, J.M. McCarthy, M. Sosonkina, and L.T. Watson extends **HOMPACK90** by L.T. Watson, M. Sosonkina, R.C. Melville, A.P. Morgan, and H.F. Walker (1997) and **HOMPACK** by L.T. Watson, S.C. Billups, and A.P. Morgan (1987).

Parallel PHCpack

parallel implementation of polynomial homotopy continuation methods

PHC = Polynomial Homotopy Continuation

- Version 1.0 archived as Algorithm 795 by ACM TOMS (1999)
- Pleasingly parallel implementations

  + Yusong Wang of Pieri homotopies (HPSEC'04)
  + Anton Leykin of monodromy factorization (HPSEC'05)
  + Yan Zhuang of polyhedral homotopies (HPSEC'06)
- Interactive Parallel Computing:

  + Yun Guan: PHClab, experiments with MPITB in Octave
  + Kathy Piret: bindings with Python, use of sockets

Release v2.3.42 extends **phcpy** and a preliminary **PHCwulf.py**.

PHCwulf.py: interactive client/server computing

jointly with Kathy Piret, released in v2.3.42 of PHCpack

- PHCpack is developed in Ada with the gnu-ada compiler
  PHClib is an interface for the C programmer,
  for parallel path trackers using MPI.
  phcpy is a Python module, implemented on top of PHClib.

  ```
  >>> import phcpy
  >>> S = phcpy.solve(P)
  ```

  Both P and S are lists of strings, strings of polynomials and solutions respectively.

- **PHCwulf.py** provides routines to send and receive lists of strings for use with Python's sockets module.
  A multithreaded server accepts connections from clients.
  Handler threads of the server send jobs to the clients.
  The clients compute and send the results back to the server.

Witness Sets

to represent solution sets

How do we represent positive dimensional solution sets?

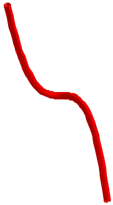
$\rightarrow$ data structure for ***numerically stable*** algorithms

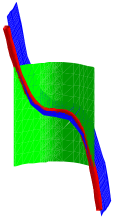
- A k-dimensional solution set of degree d is represented by
  1. k general hyperplanes; and
  2. d isolated solutions on those k hyperplanes.
- Witness sets are computed either
  1. *top down*: via a cascade of homotopies; or
  2. *bottom up*: diagonal homotopies intersect witness sets.
- Once solution sets of different dimensions are separated as different witness sets, with monodromy and traces we compute ***a numerical irreducible decomposition***.

Representing a Space Curve

Consider the twisted cubic:

$$\begin{cases} y - x^2 = 0 \\ z - x^3 = 0 \end{cases}$$





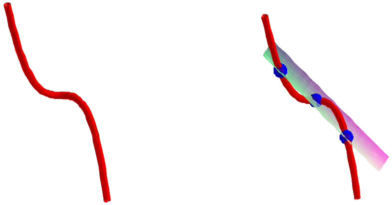
Important attributes are dimension and degree:

- dimension: cut with one random plane,
- degree: #points on the curve and in the plane.

Witness Set for a Space Curve

Consider the twisted cubic:

$$\begin{cases} y - x^2 = 0 \\ z - x^3 = 0 \end{cases} \qquad \begin{cases} y - x^2 = 0 \\ z - x^3 = 0 \\ c_0 + c_1 x + c_2 y + c_3 z = 0 \end{cases}$$



Intersect with a random plane $c_0 + c_1 x + c_2 y + c_3 z = 0$
$\rightarrow$ find three generic points on the curve.

What does a diagonal homotopy do?

input/output specification

Input: two irreducible components A and B
given by two witness sets:

Witness Set for A	Witness Set for B
$\begin{cases} f_A(x) = 0 \\ L_A(x) = 0 \end{cases}$	$\begin{cases} f_B(x) = 0 \\ L_B(x) = 0 \end{cases}$
$\sharp L_A = dim$(A)=a	$\sharp L_B = dim$(B)=b
$\{\alpha_1, \alpha_2, \ldots, \alpha_{deg(A)}\}$	$\{\beta_1, \beta_2, \ldots, \beta_{deg(B)}\}$

Output: witness sets for all pure dimensional components of A∩B

A.J. Sommese, J. Verschelde, and C.W. Wampler:
Homotopies for intersecting solution components of polynomial systems. *SIAM J. Numerical Anal.* 42(4):1552-1571, 2004.

What does diagonal homotopy do?

a special case

1. Solution pairs start a cascade of homotopies.
2. Hyperplanes are removed one by one in the cascade.

Special case: A and B are complete intersections, stored as
$\sharp\{\mathbf{x} \in \mathbb{C}^n | f_A(\mathbf{x}) = 0, L_A(\mathbf{x}) = 0\} = \deg(A)$,
$\sharp\{\mathbf{y} \in \mathbb{C}^n | f_B(\mathbf{y}) = 0, L_B(\mathbf{y}) = 0\} = \deg(B)$, and $\dim(A \cap B) = 0$,
then the diagonal homotopy is

$$h(\mathbf{x}, \mathbf{y}, t) = \left\{ \begin{array}{l} f_A(\mathbf{x}) = 0, f_B(\mathbf{y}) = 0 \\ (1-t)\begin{pmatrix} L_A(\mathbf{x}) \\ L_B(\mathbf{y}) \end{pmatrix} + t(\mathbf{x} - \mathbf{y}) = 0, \end{array} \right.$$

starting at the $\deg(A) \times \deg(B)$ solutions in $A \times B \in \mathbb{C}^{n+n}$.

At $t = 1$, we find solutions at the diagonal $\mathbf{x} = \mathbf{y}$, in $A \cap B$.

Parallel Diagonal Homotopy

some implementation issues

- Runs in various stages: every stage removes one hyperplane in the cascade of homotopies.
- Currently we use the extrinsic version of the diagonal homotopy.
- For memory efficiency, **jumpstarting** homotopy:
  1. The manager computes a start solution or reads it from file **“just in time”** whenever a worker needs a path tracking job.
  2. As soon as a worker finishes tracking a path, the solution is **written to file**.

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

An Illustration

intersecting quadric with quartic surface using 5 nodes

Assume two witness sets are completed, of degrees 2 and 4.
Using 5 nodes:

manager		**workers**
path 1 to node 1	*(1,1)*	node 1 receives path 1
path 2 to node 2	*(1,2)*	node 2 receives path 2
path 3 to node 3	*(1,3)*	node 3 receives path 3
path 4 to node 4	*(1,4)*	node 4 receives path 4
resetting file for witness set 2		
path 5 to node 1	*(2,1)*	node 1 receives path 5
path 6 to node 2	*(2,2)*	node 2 receives path 6
path 7 to node 3	*(2,3)*	node 3 receives path 7
path 8 to node 4	*(2,4)*	node 4 receives path 8

A Subsystem-by-Subsystem Solver

Extension of Previous Work

- A.J. Sommese, J. Verschelde, and C.W. Wampler:
  Solving Polynomial Systems Equation by Equation.
  In IMA Volume 146 on *Algorithms in Algebraic Geometry*,
  pages 133-152, Springer, 2008.
- The equation-by-equation solver is a limiting case
  of the subsystem-by-subsystem approach.
- Here we apply the diagonal homotopy
  in a more flexible way.

Divide and Conquer

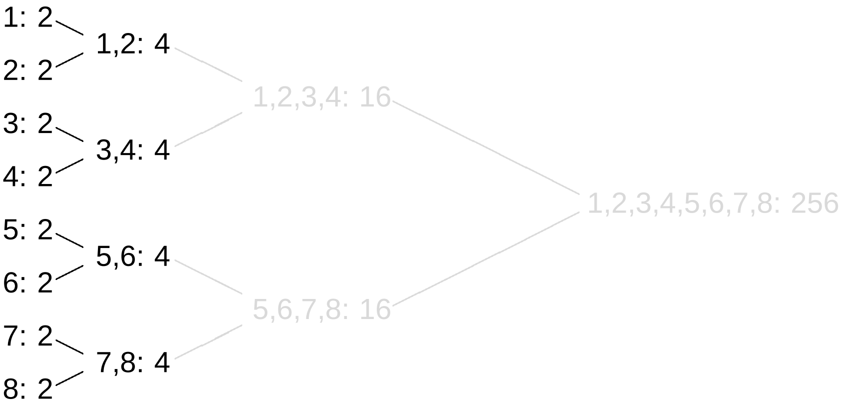
Schematic overview of solving a system of eight quadrics.

1: 2
2: 2
1,2: 4
3: 2
4: 2
3,4: 4
1,2,3,4: 16
5: 2
6: 2
5,6: 4
7: 2
8: 2
7,8: 4
5,6,7,8: 16
1,2,3,4,5,6,7,8: 256

Assume homotopy is optimal: no diverging paths.

Divide and Conquer

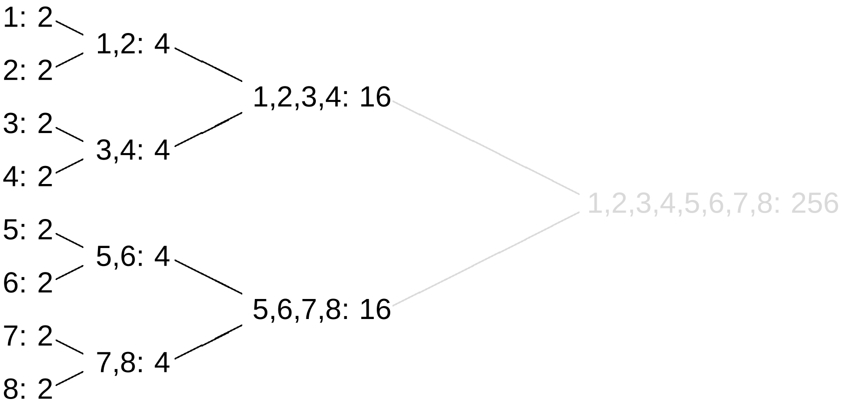
Schematic overview of solving a system of eight quadrics.



Assume homotopy is optimal: no diverging paths.

Divide and Conquer

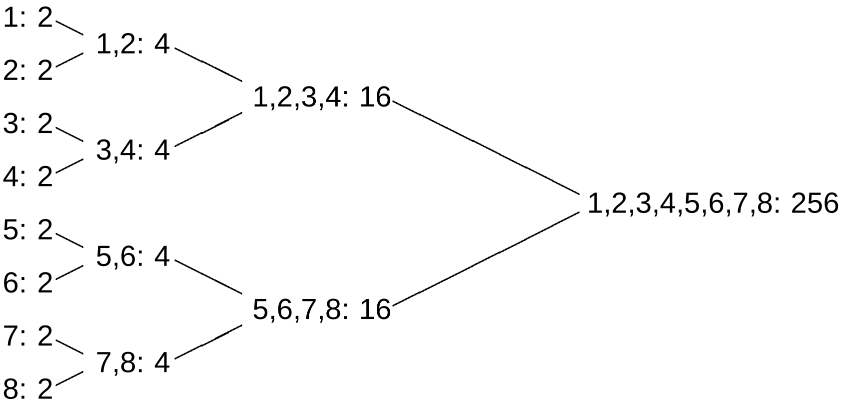
Schematic overview of solving a system of eight quadrics.



Assume homotopy is optimal: no diverging paths.

Divide and Conquer

Schematic overview of solving a system of eight quadrics.



Assume homotopy is optimal: no diverging paths.

Data Structures

The triangular state table

1	.....	.....	.....
2	.....	.....	
3	.....		
4	.....		
5			
6			
7			
8			

of completed jobs

Queue of jobs

...	job 4	job 3	job 2	job 1

Queue of idle workers

...	worker 3	worker 2	worker 1

One job

2	1	2	1	2	3	4	2

Data Structures

The triangular state table

1	.....	.....	.....
2	.....	.....	
3	.....		
4	.....		
5			
6			
7			
8			

of completed jobs

Queue of jobs

...	job 4	job 3	job 2	job 1

Queue of idle workers

...	worker 3	worker 2	worker 1

One job

2	1	2	1	2	3	4	2

Data Structures

The triangular state table

1	.....	.....	.....
2	.....	.....	
3	.....		
4	.....		
5			
6			
7			
8			

of completed jobs

Queue of jobs

...	job 4	job 3	job 2	job 1

Queue of idle workers

...	worker 3	worker 2	worker 1

One job								
	2	1	2	1	2	3	4	2

Data Structures

The triangular state table

1	.....	.....	.....
2	.....	.....	
3	.....		
4	.....		
5			
6			
7			
8			

of completed jobs

Queue of jobs

...	job 4	job 3	job 2	job 1

Queue of idle workers

...	worker 3	worker 2	worker 1

One job

2	1	2	1	2	3	4	2

Initial Job Distribution

manager		**worker**	
broadcast file name	$\rightarrow$	receive file name	*file with equations*
send data	$\rightarrow$	receive data solve equation write to file	*data =* *equation indices* *terminated by 0*
receive data	$\leftarrow$	send data	*synchronization*

Initial Job Distribution

manager		worker	
broadcast file name	$\rightarrow$	receive file name	*file with equations*
send data	$\rightarrow$	receive data solve equation write to file	*data =* *equation indices* *terminated by 0*
receive data	$\leftarrow$	send data	*synchronization*

Initial Job Distribution

manager		**worker**	
broadcast file name	$\rightarrow$	receive file name	*file with equations*
send data	$\rightarrow$	receive data solve equation write to file	*data =* *equation indices* *terminated by 0*
receive data	$\leftarrow$	send data	*synchronization*

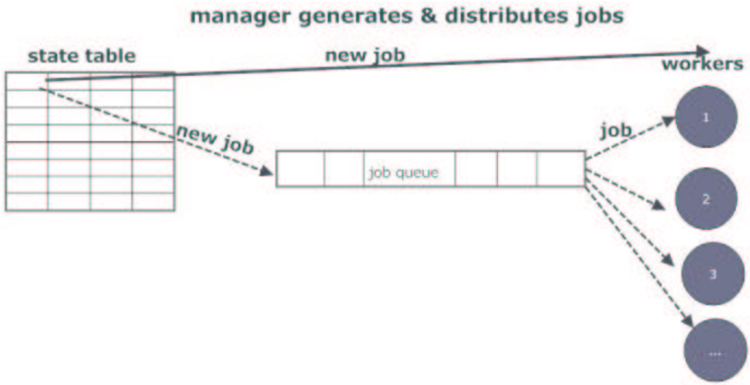
Initial Job Distribution

manager		**worker**	
broadcast file name	$\rightarrow$	receive file name	*file with equations*
send data	$\rightarrow$	receive data solve equation write to file	*data =* *equation indices* *terminated by 0*
receive data	$\leftarrow$	send data	*synchronization*

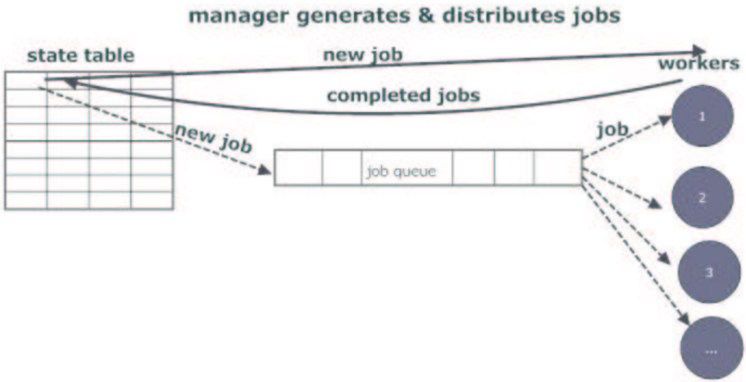
Job Scheduling: the main loop

- Runs in $\lceil \log_2(n) \rceil$ stages, $n = \#$equations.
- Homotopies in stage k involve 2^k equations.
- The manager maintains the state table, the job queue, and the queue of idle workers.

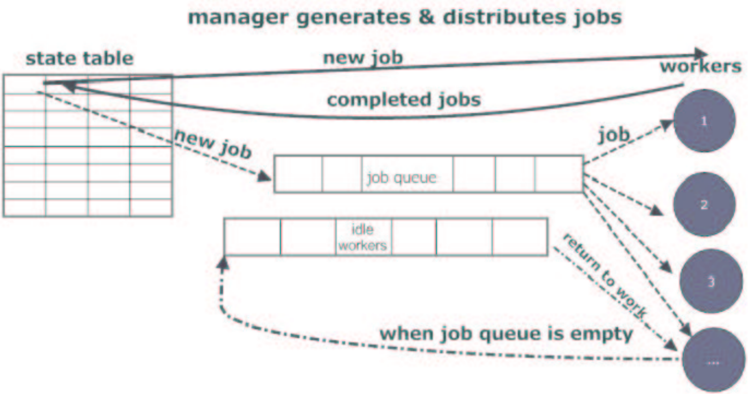
Job Scheduling: main loop, a picture



Job Scheduling: main loop, continued



Job Scheduling: main loop, finally



Software & Equipment

Diagonal homotopies are available in PHCpack.
`http://www.math.uic.edu/~jan/download.html`

1. Parallel code uses and improves sequential versions.
2. `PHClib` forms interface with PHCpack as library.
3. Main parallel programs use MPI for communication.

Computers used:

- Software development on personal cluster:
  1. One workstation with two dual 2.4Ghz processors.
  2. Two Rocketcalc clusters: one with four and an other with eight 2.4Ghz processors.
- NCSA Tungsten cluster is a supercomputer: 1280 3.2GHz processors, running Linux.
- MacPro two quad core processors.

Complexity of Job Scheduling

Job scheduling uses dynamic load balancing.

Some additional concerns:

- There must be sufficient points in both witness sets in order to intersect a pair of witness sets.
- New jobs can be formed only when pairs of witness points are completed.
- The solutions for the second witness set are arriving much slower than those for the first witness set.

Synchronization currently prevents optimal speedup.

Solving `katsura8`

on a 2.4Ghz Rocketcalc personal cluster

	wall	#paths/node	
p	time	max	min
2	459s	1,408	1,408
3	277s	787	621
4	175s	514	391
5	140s	375	289
6	104s	307	240
7	98s	251	207
8	86s	218	173
9	85s	193	147
10	81s	167	132
11	72s	152	124
12	68s	147	110

$p = 2$: 1 worker

double #workers (1,2,4,8):

$p = 2 \rightarrow 3 \rightarrow 5 \rightarrow 9$

time: 459s

$\rightarrow$ 277s

$\rightarrow$ 140s

$\rightarrow$ 85s

Solving `adjmin28`

on a 2.4Ghz Rocketcalc personal cluster

	wall	#paths/node	
p	time	max	min
2	2,179s	1,408	1,408
3	1,255s	771	637
4	887s	497	423
5	619s	375	310
6	477s	299	247
7	417s	244	218
8	392s	223	180
9	363s	194	160
10	359s	181	143
11	331s	151	128
12	320s	135	119

$p = 2$: 1 worker

double #workers (1,2,4,8):

$p = 2 \rightarrow 3 \rightarrow 5 \rightarrow 9$

time: 2,179s

$\rightarrow$ 1,255s

$\rightarrow$ 619s

$\rightarrow$ 363s

Summary

- parallel diagonal homotopy allows jumpstarting for efficient memory management
- dynamic load balancing leads to acceptable speedup
- synchronization along stages gives overhead