Iterative Methods in TAXI (GMRES and BICGSTAB)

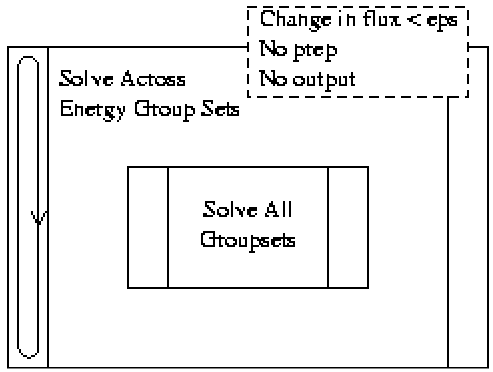
Tim Smith
Computer Science
May 11, 2005

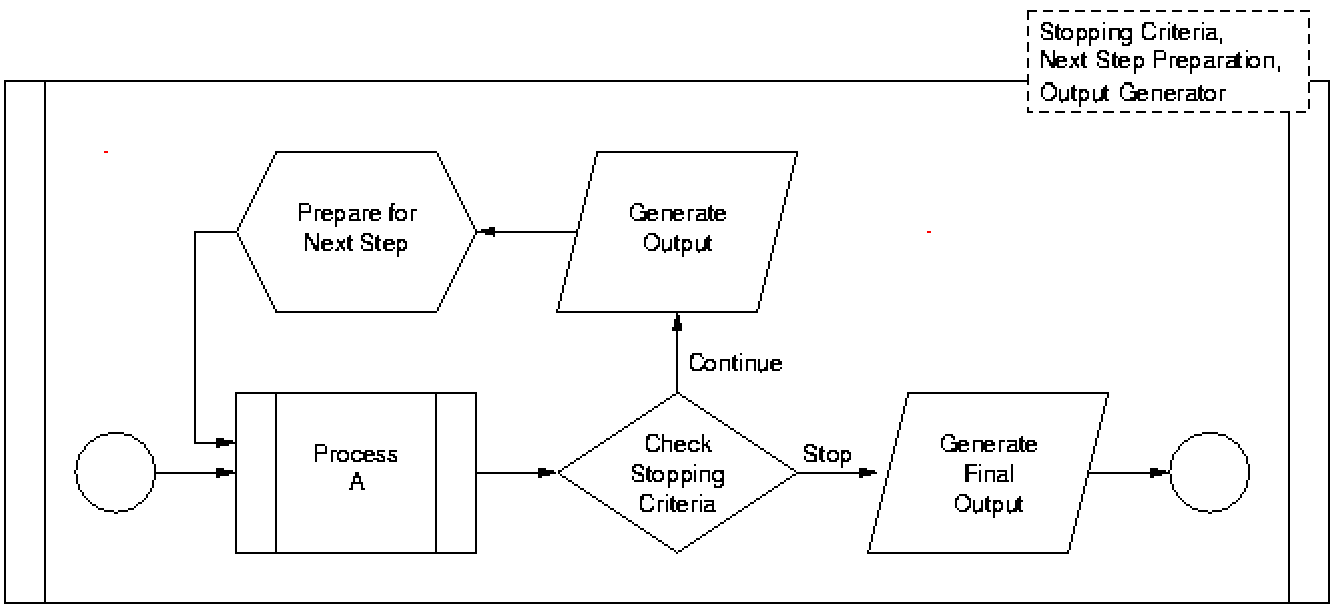
Parasol
Smarter computing.
Texas A&M University

Background and Motivation

- Source Iteration (SI) can be slow to converge
- TSA methods used to accelerate SI
  - β-TSA, ε-TSA, Stretched and Filter TSA implemented
  - SI with TSA can be slow to converge
- Using Krylov methods may improve convergence
  - Easily wrapped around existing transport methods
  - TSA can be used as a preconditioner for the solver
  - GMRES implemented by Jae Chang

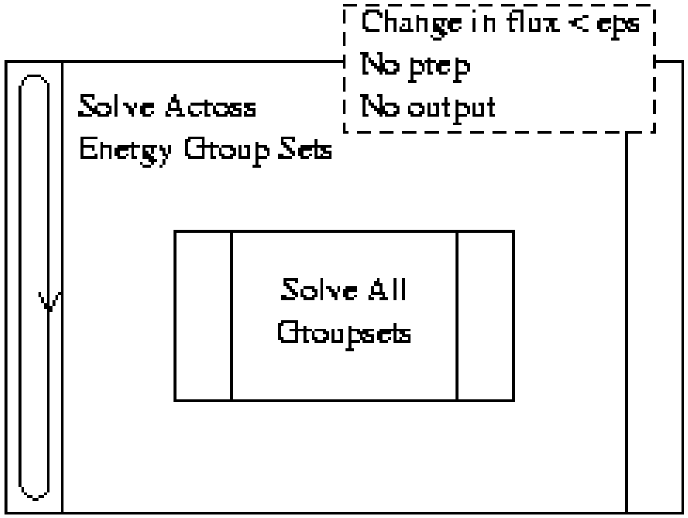
Loop Construct in Infrastructure





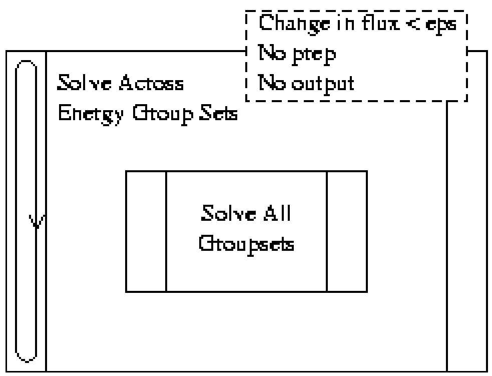
Where Iterative Methods Fit

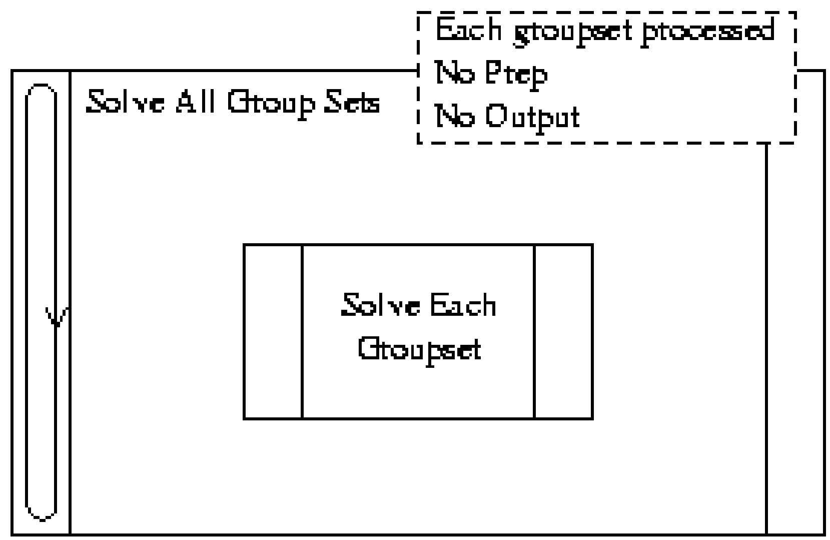
- We've chosen
  - Problem Type
    - Transport of some sort
  - Time discretization
    - May have been done
    - Not considered
  - Energy Discretization
    - Multigroup
  - Spatial discretization
    - Not considered
- Solve across groupsets
  - Iterate until solution converges in all energy groups



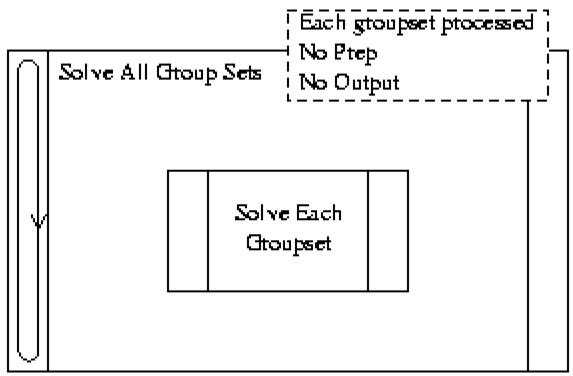
Solve All Energy Group Sets

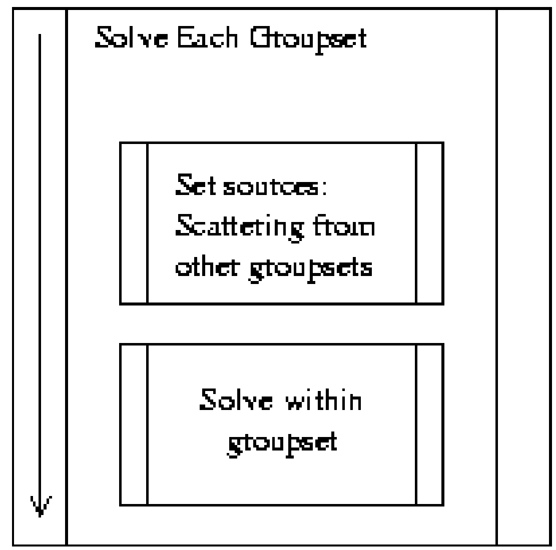
- Iteration is a loop over all groupsets



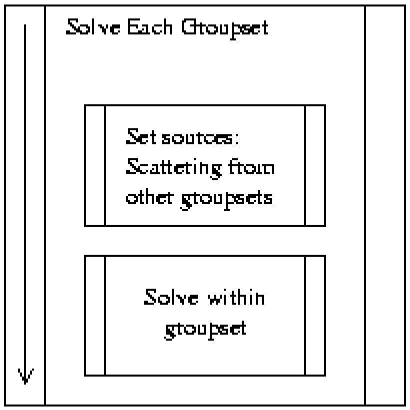


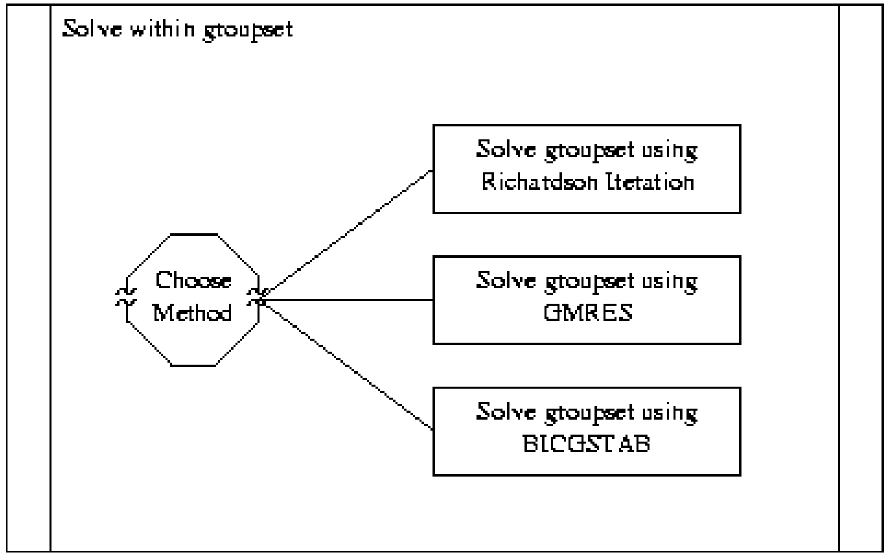
Solve Each Energy Group Set





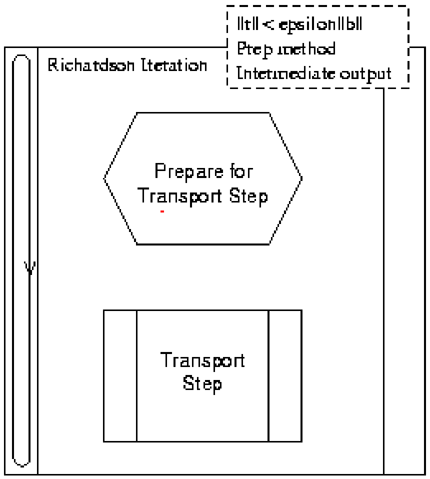
Solve within Energy Group Set

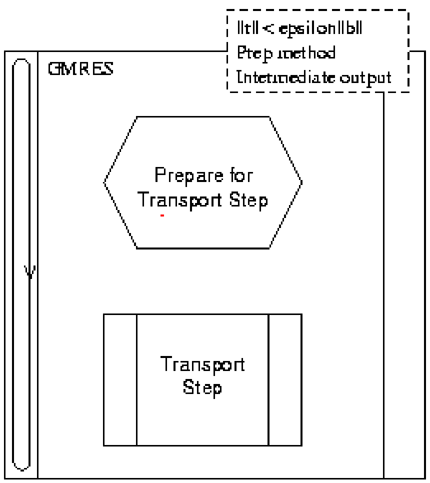




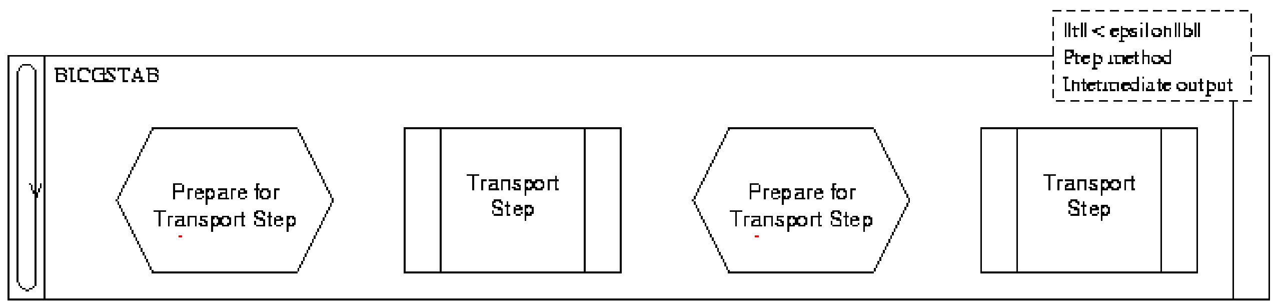
- Still not committed to a spatial discretization method
- Choice of method used here can be based on characteristics of problem input

Solver Methods

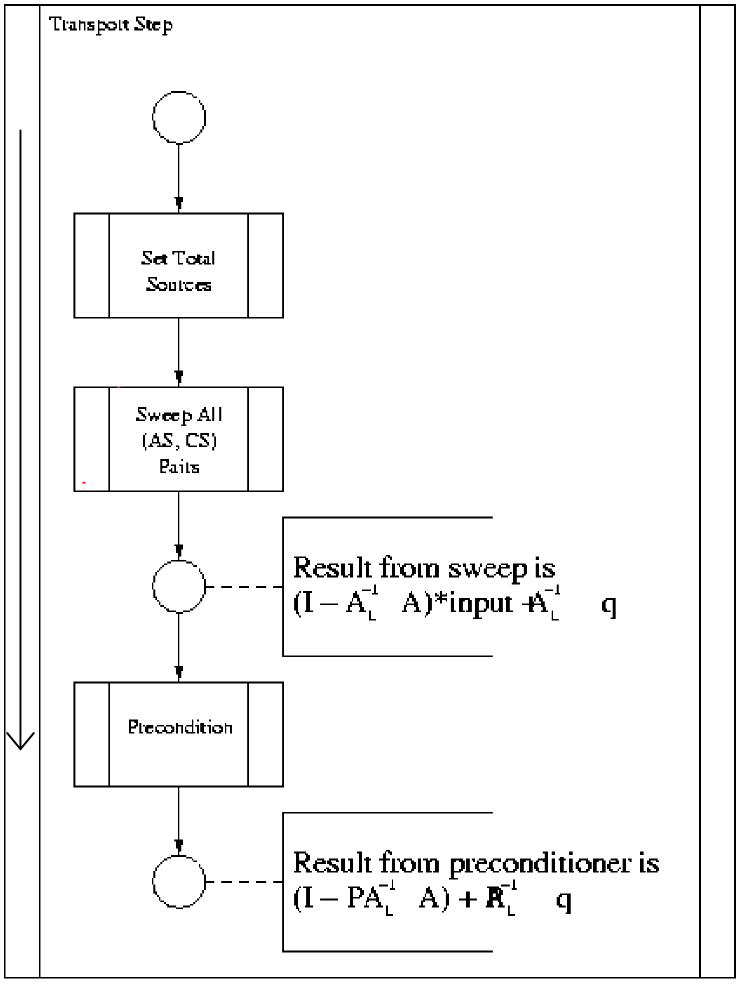


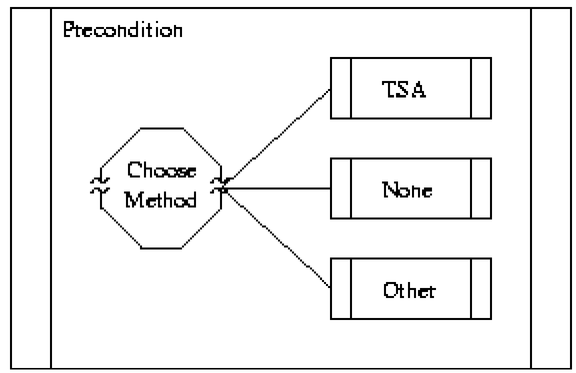


- Iteration in each method is iteration to a fixed point
- BICGSTAB has two transport steps per iteration



Transport Step





BICGSTAB [Van Der Vorst '92]

$x_0 = 0$; $r_0 = b$; rhat = r_0
$\rho_0 = \alpha = \omega_0 = 1$; $v_0 = p_0 = 0$; $i = 0$;
While (!converged && i < maxiterations)
  ρ_i = (rhat, r_{i-1}); $\beta = (\rho_i / \rho_{i-1})(\alpha / \omega_{i-1})$;
  $p_i = r_{i-1} + \beta(p_{i-1} - \omega_{i-1} v_{i-1})$; // find search vector
  $v_i = Ap_i$;
  $\alpha = \rho_i$/(rhat/v_i);
  $s = r_{i-1} - \alpha v_i$;
  $t = As$;
  $\omega_i = (t,s)/(t,t)$;
  $x_i = x_{i-1} + \alpha p_i + \omega_i s$; // update solution
  if x_i is accurate enough then converged = true;
  $r_i = s - \omega_i t$;
End

Memory Usage

	Extra pArrays	Solution vector storage	Total
GMRES	4	Restart value	(4 + restart) * total # elems
BICGSTAB	8	None	8 * total # elems

- BICGSTAB uses less than GMRES with restart values > 4
- This memory is in addition to SI data structures
  - Grid, Quadrature objects, etc.

Operations

	Setup	Per Iteration				
		Sweeps	Norms (dotprod, 2-norm)	Vec-scalar (αv)	Vec-vec Ops (daxpy)	Total (not counting sweeps)
BICGSTAB	1 Sweep	2	5	7	11	$O(23n/p + 5\log_2 p)$
GMRES Every Iteration	1 Sweep 2 norms	1	i+3	i+2	2i+2	$O((4i+7)n/p + (i+3)\log_2 p)$
GMRES Every mth iteration		1	i+3		1	

- i in GMRES counts is the current size of the iteration matrix
  - varies from 1 to restart value
  - BICGSTAB uses less memory than GMRES with restart value >= 4
- Each sweep and norm operation uses a global barrier

Breakdown Conditions

- Restarted GMRES break down
  - Due to unavailability of previous search vectors
  - Progress towards solution with new search vectors is slowed
  -
- BICGSTAB breaks down when rhat is poorly chosen
  - ρ_i = (rhat, r_{i-1}) = 0 (or very small)
  - Progress towards solution is not made.
  - May choose new rhat or use GMRES.

We test the methods on a homogenous problem.

- 100 x 100 cells
- 100 x 100 mfp
- $\sigma_t = 1.0$
- Scattering ratio = 0.99
- Incident source on the left face
- No volumetric source
- Convergence tolerance = 1.0E-07
- Diamond difference spatial discretization

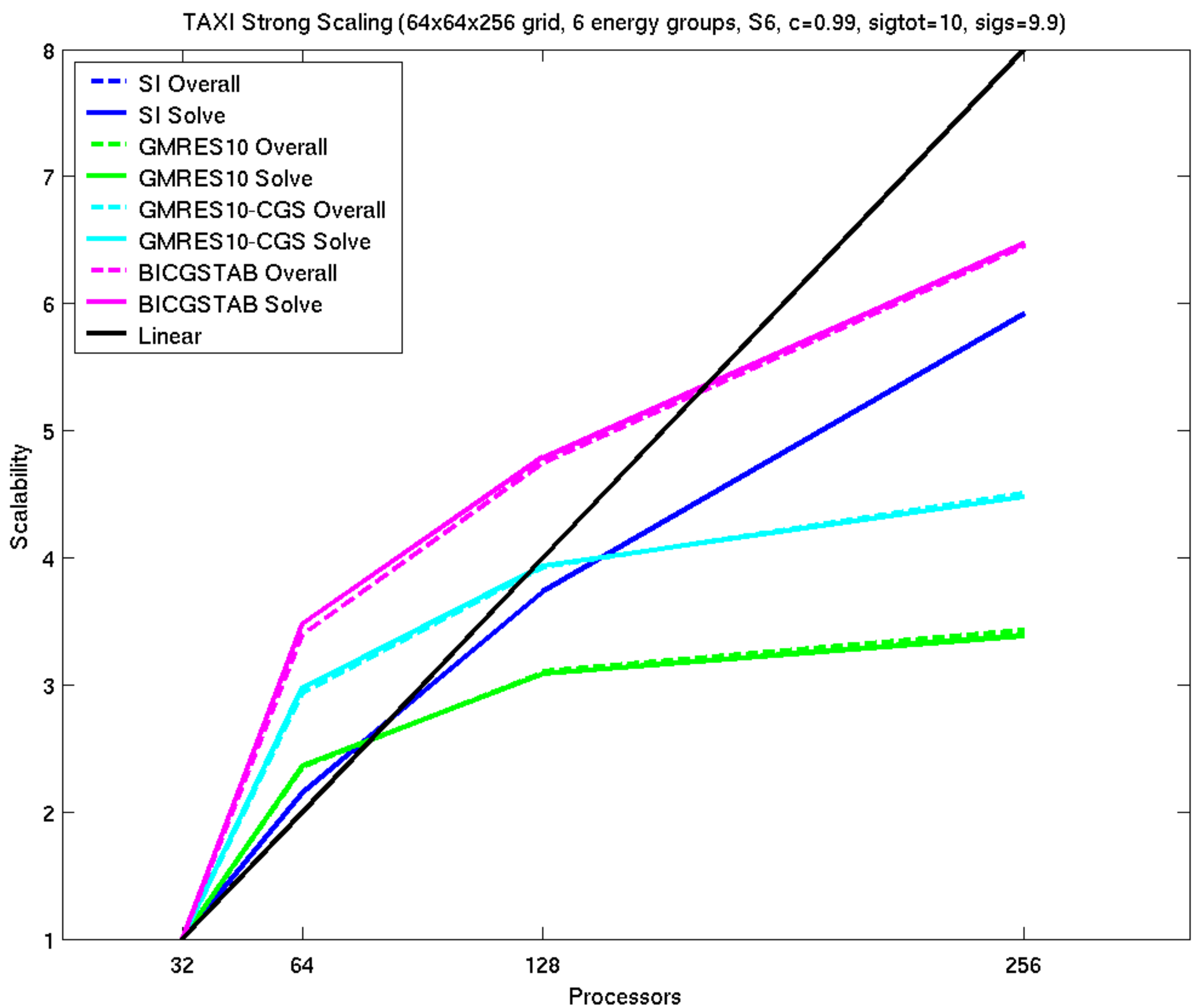
Numerical Results for BICGSTAB

Method	CPU Time (seconds)	Iterations	GMRES CPU Time (seconds)	GMRES Iterations	BICGSTAB CPU Time (seconds)	BICGSTAB Iterations
SI	3841	2270	396	99	322	51
SFTSA (γ=0)	1320	182	482	50	427	24
SFTSA (γ=0.3)	1927	267	389	40	411	23
SFTSA (γ=1.0)	3729	519	261	26	258	14
SFTSA (γ=2.0)	3196	444	194	19	190	10
SFTSA (γ=3.0)	443	59	194	19	208	11
SFTSA (γ=10.0)	1529	211	270	27	258	14

Parallel Scalability

- Weekly Timing Problem
  - Added input section so BICGSTAB or GMRES used
  - No TSA Acceleration
- Input Specs
  - 64x64x256 cells
  - S6, 48 angles (6 angles per angleset)
  - 6 energy groups (6 groups per groupset)
  - σ_t=10, σ_s=9.9
- Experiments run on frost at LLNL
  - 16 Power3 procs @ 300MHz per node
  - 16GB memory per node

Strong Scaling Results



Execution Times

Parasol

	Iterations	Overall Time (seconds)			
		32	64	128	256
SI	1410	37169	17297	9961	6277
GMRES10	74	12930	5497	4172	3777
GMRES10 CGS	74	11408	3883	2907	2529
BICG	44	9362	2757	1974	1450

	Solve Times (seconds)			
	32	64	128	256
SI	36848	17138	9874	6224
GMRES10	12601	5336	4085	3723
GMRES10 CGS	11093	3727	2819	2475
BICG	9038	2596	1887	1397

Conclusions

- Implementing BICGSTAB in our code was easy
  - GMRES used as a guide (e.g. how to do matvec)
- BICGSTAB scales better than GMRES
- Iterative Methods shouldn't use TSA if problems use low order quadratures

Future Work

- Determine when SI, GMRES, or BICGSTAB is the better method
- Identify problem attributes we can use to decide which method to use
- Adaptively choose method to use at run-time