LINUX PARALLEL GYROKINETIC SOLVER

J. Candy, R.E. Waltz, M.N. Rosenbluth, and F.L. Hinton

General Atomics
San Diego, CA 92186-5608, U.S.A.

American Physical Society
Division of Plasma Physics

15-19 November 1999
Seattle, WA

Introduction and Motivation

- The electromagnetic gyrokinetic equations encompass the full physics of turbulence and transport due to low-frequency modes ($\omega \ll \omega_{\rm ci}$), and for plasmas where the ion gyroradius is small compared with the magnetic field scale length.

- Previously, linear and nonlinear equations have been formulated using a high-n ballooning mode representation (BMR) in which the relative ion gyroradius, $\rho_* \equiv \rho_i/a$, is vanishingly small.

- Previously, nonlinear simulations have been limited to at least one of: adiabatic electrons, electrostatic perturbations, gyrofluid approximation, ballooning limit, $s-\alpha$ model geometry.

An Aggressive Approach

- We want to solve **electromagnetic gyrokinetic-Maxwell equations with full profile effects**.
- Why? Because it is well-known from BM-ITG simulations that shear in the $\mathbf{E} \times \mathbf{B}$ velocity can completely quench the turbulence when the shear rate,

$$\gamma_E = \frac{r}{q}\frac{\partial}{\partial r}\left(\frac{q v_{E\times B}}{r}\right) \ ,$$

is comparable to $\gamma_{\rm l,max}$ from balloning modes.

- In reality, the shear rate in the finite-ρ_* diamagnetic velocity is comparable to the shear rate above, and is expected to have a similar effect.
- To compute finite-ρ_* effects we use a real radial grid-space representation – avoiding the Fourier representation of the BMR (flux tube).

Gyrokinetic Equation

- For each species, denoted by index s, we must solve the gyrokinetic equation

$$\frac{\partial g_\mathrm{s}}{\partial t} + \left(v_{||}\hat{b} + \vec{v}_\mathrm{D} \right) \cdot \nabla g_\mathrm{s} - \mathcal{C}(g_\mathrm{s}) = -z_\mathrm{s}\, e\, \frac{\partial \langle U \rangle_\mathrm{s}}{\partial t} \frac{\partial F_\mathrm{s}}{\partial E} - \frac{c}{B} \left(\hat{b} \times \nabla \langle U \rangle_\mathrm{s} \right) \cdot \nabla \left(F_\mathrm{s} + g_\mathrm{s} \right) \ ,$$

where U is a linear combination of fields

$$U \equiv \varphi - v_{||} A_{||} \ .$$

- The nonadiabatic distribution of gyrocenters, g, is defined in terms of the full perturbed distribution, f, according to

$$f(\vec{x}, \vec{v}, t) = e\varphi(\vec{x}, t) \frac{\partial F_0}{\partial E} + g(\vec{R}, E, \mu, \sigma, t) \ .$$

Gyroaveraging Considerations

- It is important to understand that the coordinate $\vec{R}$ labels the position of a gyrocenter, such that the particle position is $\vec{x} = \vec{R} + \vec{\rho}(\zeta)$, where ζ is the gyroangle and $\vec{\rho}$ the gyrovector. Note also that

$$\vec{\rho} = \frac{\hat{b} \times \vec{v}_\perp}{\Omega} \qquad \text{and} \qquad \frac{d\vec{\rho}}{dt} = \vec{v}_\perp \ .$$

- In the present work, we do not consider any but lowest-order terms in the gyroaveraging procedure:

$$\langle f \rangle \, (\vec{R}, \mu) \equiv \frac{1}{2\pi} \oint d\zeta \, f(\vec{R} + \vec{\rho}) \ .$$

- More rigorous averaging methods exist (Brizard, Hahm).

The Maxwell Equations

- The field evolution is determined by reduced Maxwell equations:

  (i) Poisson:

$$-\nabla^2\varphi = 4\pi \sum_{\mathrm{s}} e z_{\mathrm{s}} \int d^3v \left(z_{\mathrm{s}} e \varphi(\vec{x}) \frac{\partial F_{\mathrm{s}}}{\partial E} + \langle g \rangle_{\mathrm{s}} \right) \; ;$$

  (ii) Ampère:

$$-\nabla^2 A_{||} = 4\pi \sum_{\mathrm{s}} e z_{\mathrm{s}} \int d^3v \, v_{||} \langle g \rangle_{\mathrm{s}} \; .$$

- These equations are time-independent, and must be solved using a suitable time-implicit method.

Time-advance Algorithm

- We emphasize that the memory requirements of full nonlinear problem **alone** (~ 40 Gb) necessitate massively-parallel processing (MPP) solution.

- The time-advance algorithm has been designed with various obstacles in mind:

  (i) latency and poor bandwidth of MPI-communication calls;

  (ii) large matrix sizes (storage) connected with solution of multi-dimensional PDE's.

- To advance the coupled gyrokinetic-Maxwell equations in time, we split the various partial differential operators into radial, poloidal, collisional (pitch-angle), and nonlinear parts, and thus split the time advance into stages.

- The result, after a sequence of separate steps, gives the fully-advanced fields and distributions accurate to first-order in time.

Numerical Grid Indices

- A minimial nonlinear simulation requires grid dimensions on the order of

Species: $i_s \leq 2$

Energy: $i_{\mathcal{E}} \leq 5$

Pitch Angle: $i_k \leq 25$

Toroidal mode: $i_n \leq 10$

Radius: $i \leq 100$

Poloidal Angle: $j \leq 32$

Field: $i_{\mathrm{f}} \leq 2$

Distributed Objects

- Two fundamental distributed data structures are the nonadiabatic distribution, g:

$$g\left(\{p_{i_n,i_\mathcal{E},i_k}\},i_s,j_s,i\right) \longleftrightarrow g\left(\{p_{i_\mathcal{E},i_k,j_s}\},i_s,i_n,i\right)$$

and the field-solve matrix, F:

$$F\left(n,\{\mu\},\{\mu'\}\right)$$

- Curly braces indicate a distributed variable.
- $\mu \longrightarrow (i,j_s,i_s)$.

Data Redistribution (1D Parallelization)

- Different integrator stages require different data parallelization.
- For example, the poloidal and radial steps are taken with g distributed in $p_{i_n,i_\mathcal{E},i_k}$:

$$g\left(\{p_{i_n,i_\mathcal{E},i_k}\},i_s,j_s,i\right)$$

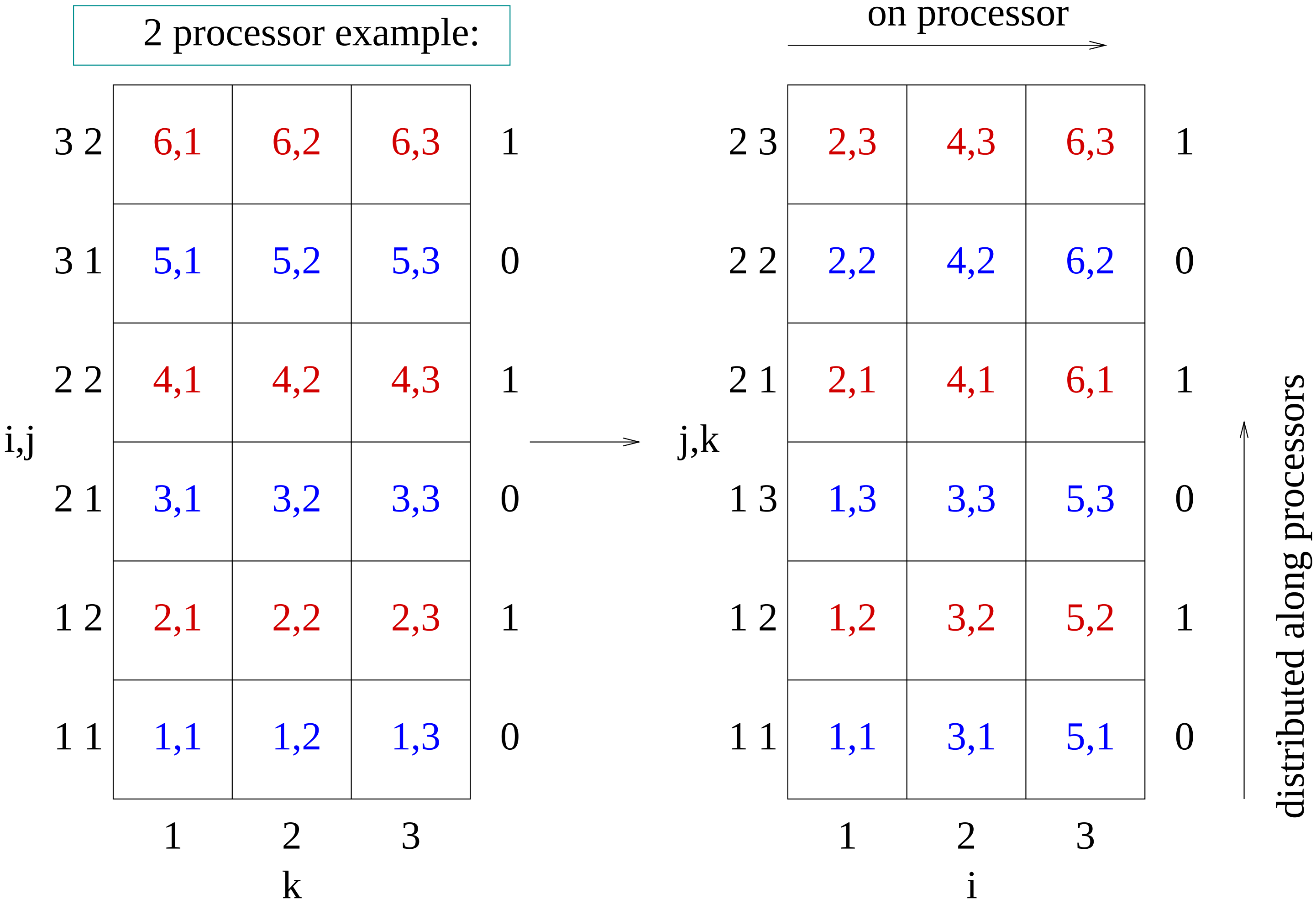
- However, a nonlinear step is taken with data distribution along $p_{i_\mathcal{E},i_k,j_s}$:

$$g\left(\{p_{i_\mathcal{E},i_k,j_s}\},i_s,i_n,i\right)$$

- Generic MPI communication subroutines have been developed for this data transformation.

Data Redistribution (1D Parallelization)

$$g\left(\{i,j\},k\right) \longleftrightarrow g\left(\{j,k\},i\right)$$



2 processor example:

i,j				
3 2	6,1	6,2	6,3	1
3 1	5,1	5,2	5,3	0
2 2	4,1	4,2	4,3	1
2 1	3,1	3,2	3,3	0
1 2	2,1	2,2	2,3	1
1 1	1,1	1,2	1,3	0
	1	2	3	
		k		

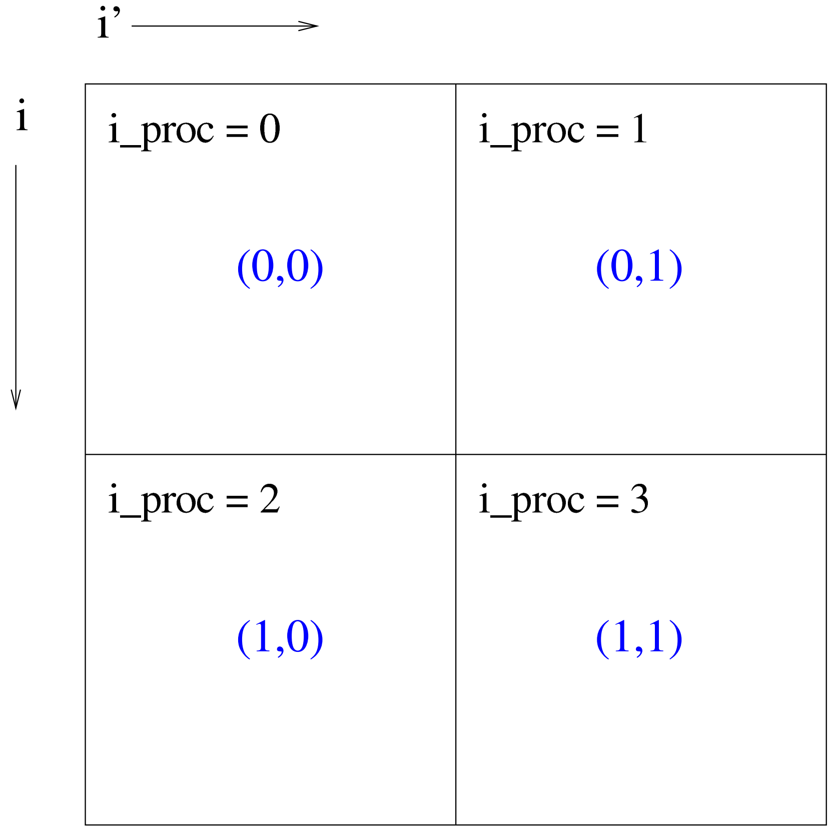
on processor

j,k				
2 3	2,3	4,3	6,3	1
2 2	2,2	4,2	6,2	0
2 1	2,1	4,1	6,1	1
1 3	1,3	3,3	5,3	0
1 2	1,2	3,2	5,2	1
1 1	1,1	3,1	5,1	0
	1	2	3	
		i		

distributed along processors

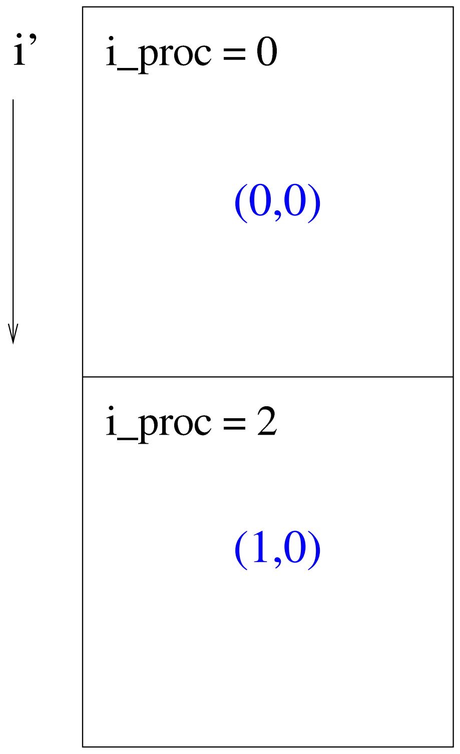
Field Advance (2D Parallelization)

$$\varphi_i = F_{ii'} S_{i'}$$



F(i,i')

Field-advance matrix is distributed across 2D processor grid



S(i')

Vectors (source/field) are stored only in processor column 0

Full-step Solution

- The splitting method we use is applicable in general to a kinetic-Maxwell system of equations:

$$\partial_t g = \mathrm{L}(g) + \partial_t U + \mathrm{N}(g, U)$$

$$U = \mathrm{M}(g)$$

- An equivalent single equation for g is:

$$\partial_t g = \mathrm{L}^*(g) + \mathrm{N}^*(g, \mathrm{M}g) \ ,$$

where $\mathrm{L}^* \equiv (1 - \mathrm{M})^{-1}\mathrm{L}$.

- Correct to $\mathcal{O}(\Delta t)$, the value of $g(t + \Delta t)$ is

$$g(t + \Delta t) = g(t) + \Delta t \, \mathrm{L}^* g(t) + \Delta t \, \mathrm{N}^*[g(t), \mathrm{M}g(t)]$$

Nonlinear Step

- The nonlinear terms are accounted for through a numerical solution of the continuous equation

$$\partial_t g = \mathrm{N}(g, \mathrm{M}g) \ .$$

- Independent of the solution technique (which we are currently developing), we require only that said solution satisfy:

$$g' = g(t) + \Delta t \, \mathrm{N}[g(t), \mathrm{M}g(t)] + \mathcal{O}(\Delta t^2) \ ,$$

given the initial value $g(t)$.

DIII-D
NATIONAL FUSION FACILITY
SAN DIEGO

Radial and Collision Step

- The radial and collision steps are nothing more that diffrential equations of the simple type

$$\partial_t g = \mathrm{L}_0(g) \ ,$$

where L_0 acts in only one dimension.

- Although in both cases we solve the linear equation by fully-implicit methods, for now it is important only that the solution satisfy

$$g'' = g' + \Delta t \, \mathrm{L}_0 g' + \mathcal{O}(\Delta t^2)$$

given the initial value g'.

Poloidal Step

- This step is perhaps more subtle; we must solve the coupled equations

$$\partial_t g = \mathrm{L}_\theta(g) + \partial_t U$$

$$U = \mathrm{M}(g)$$

- The solution must satisfy

$$g(t+\Delta t) = g'' + U(t+\Delta t) - U(t) + \Delta t\,\mathrm{L}_\theta g'' + \mathcal{O}(\Delta t^2)$$

$$U(t+\Delta t) = \mathrm{M}[g(t+\Delta t)]$$

- It is easy to verify that the totality of split-step solutions add together to give a valid first-order time-advance algorithm.

Connection with Ballooning Modes

- For an equally-spaced grid in x, fields have a Fourier representation:

$$\Psi_m(\theta) = \sum_{p=-J}^{J-1} \widetilde{\Psi}_p(\theta)\, e^{i\pi\sigma_* pm/J} \ ,$$

with $J \equiv N_r/2$.

- Evidently, the inverse transform yields

$$\widetilde{\Psi}_p(\theta) = \frac{1}{2J} \sum_{m=-J}^{J-1} \Psi_m(\theta)\, e^{-i\pi\sigma_* pm/J} \ .$$

A Sample Reconstruction

- In general, from a real-space solution we can reconstruct the ballooning-space wavefunction $\Psi_*(\theta)$
- For $J_* = J = 2$ and $\ell_* = 1$ (most unstable mode), the reconstruction takes the form

$$\Psi_*(\theta + 2\pi) = \widetilde{\Psi}_1(\theta)\, f_b \ ,$$

$$\Psi_*(\theta) = \widetilde{\Psi}_0(\theta) \ ,$$

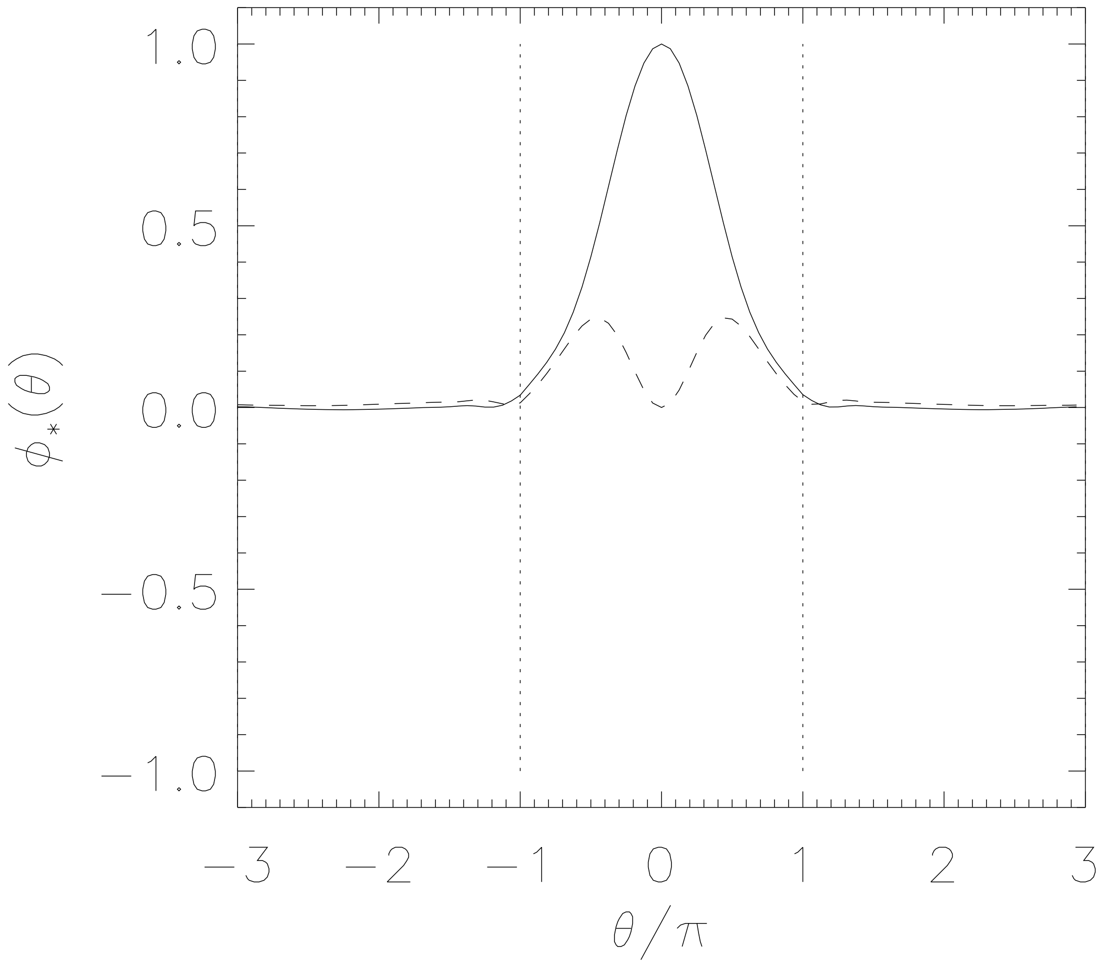
$$\Psi_*(\theta - 2\pi) = \widetilde{\Psi}_{-1}(\theta)\, f_b^{-1} \ ,$$

$$\Psi_*(\theta - 4\pi) = \widetilde{\Psi}_{-2}(\theta)\, f_b^{-2} \ .$$

for $\theta \in [-\pi, \pi)$.

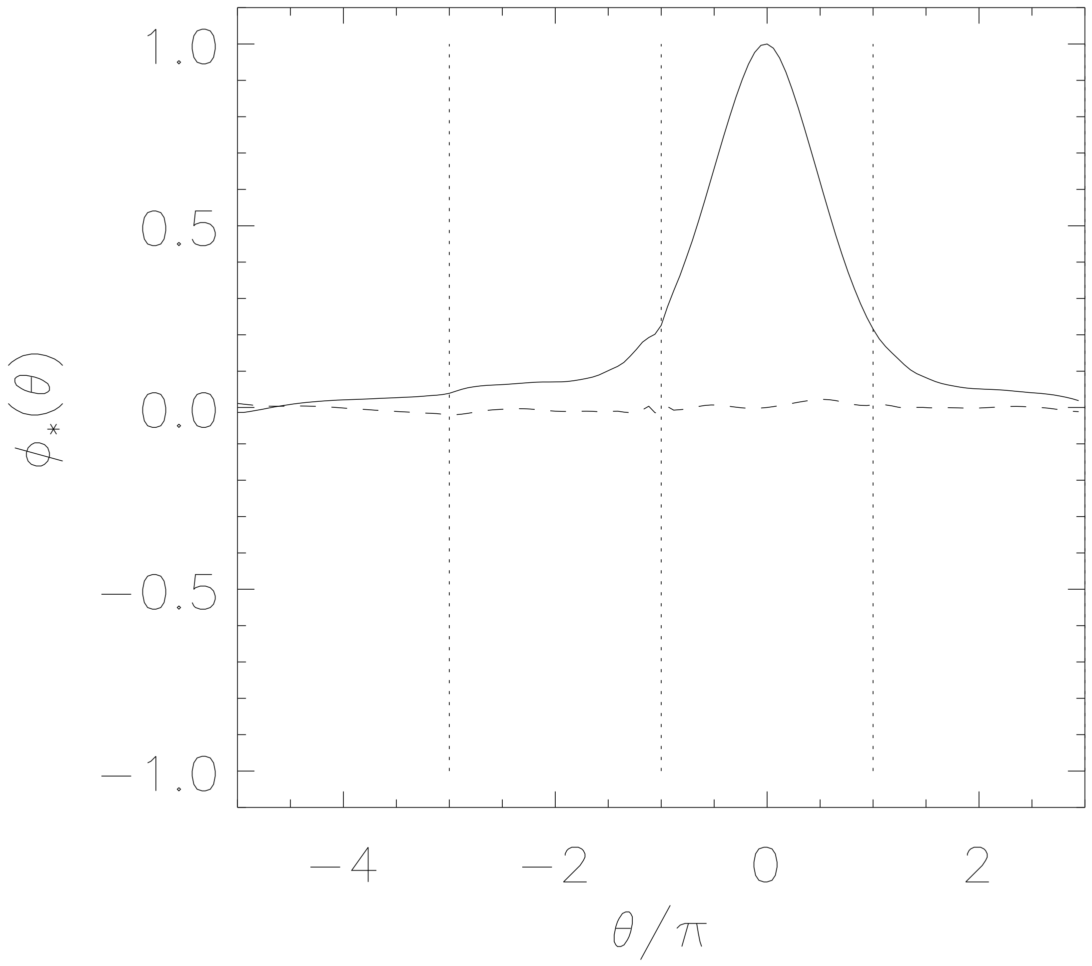
Linear Electrostatic Benchmark

- Plot of $\phi_*(\theta)$; eigenfrequency within 3% of GKS code (Kotschenreuther).



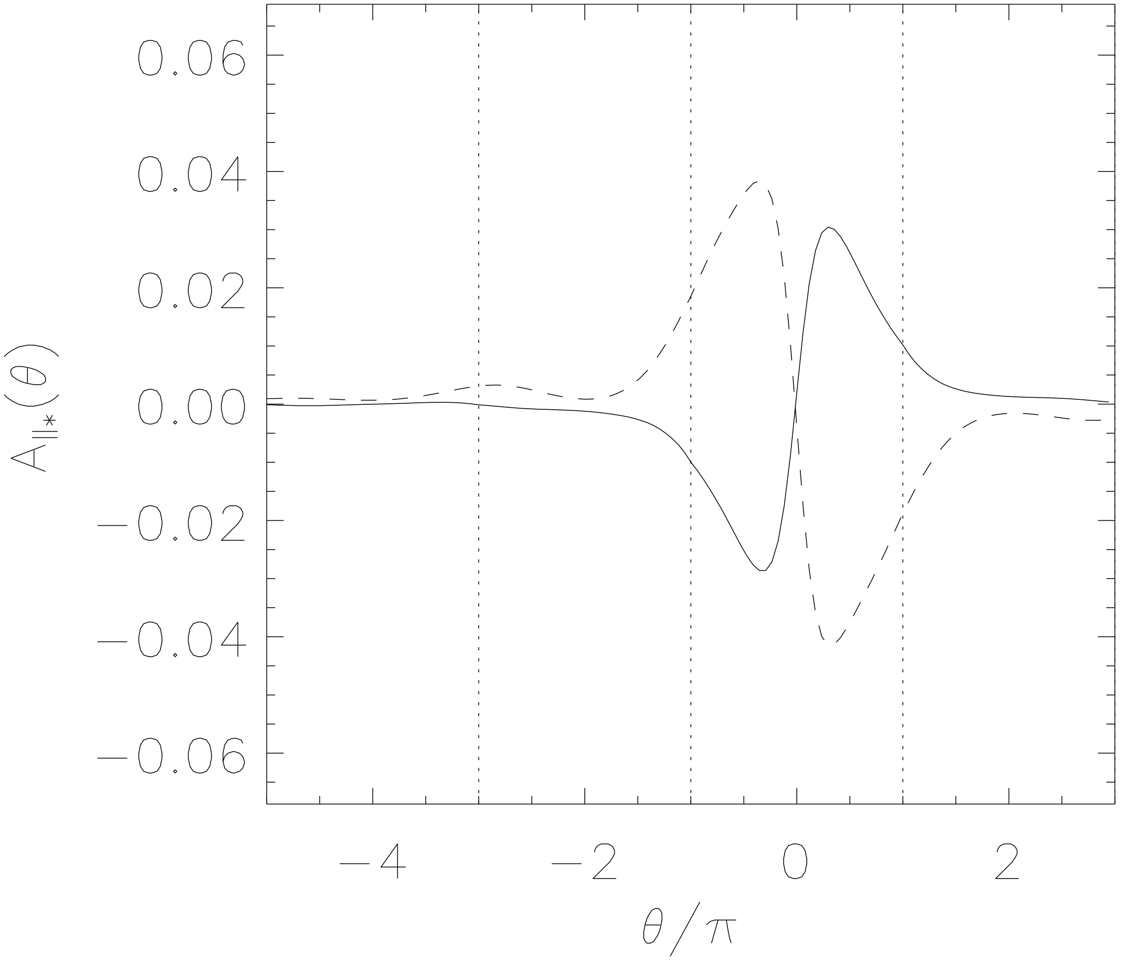
Linear Electromagnetic Benchmark

- Plot of $\phi_*(\theta)$; eigenfrequency within 5% of GKS code (Kotschenreuther).



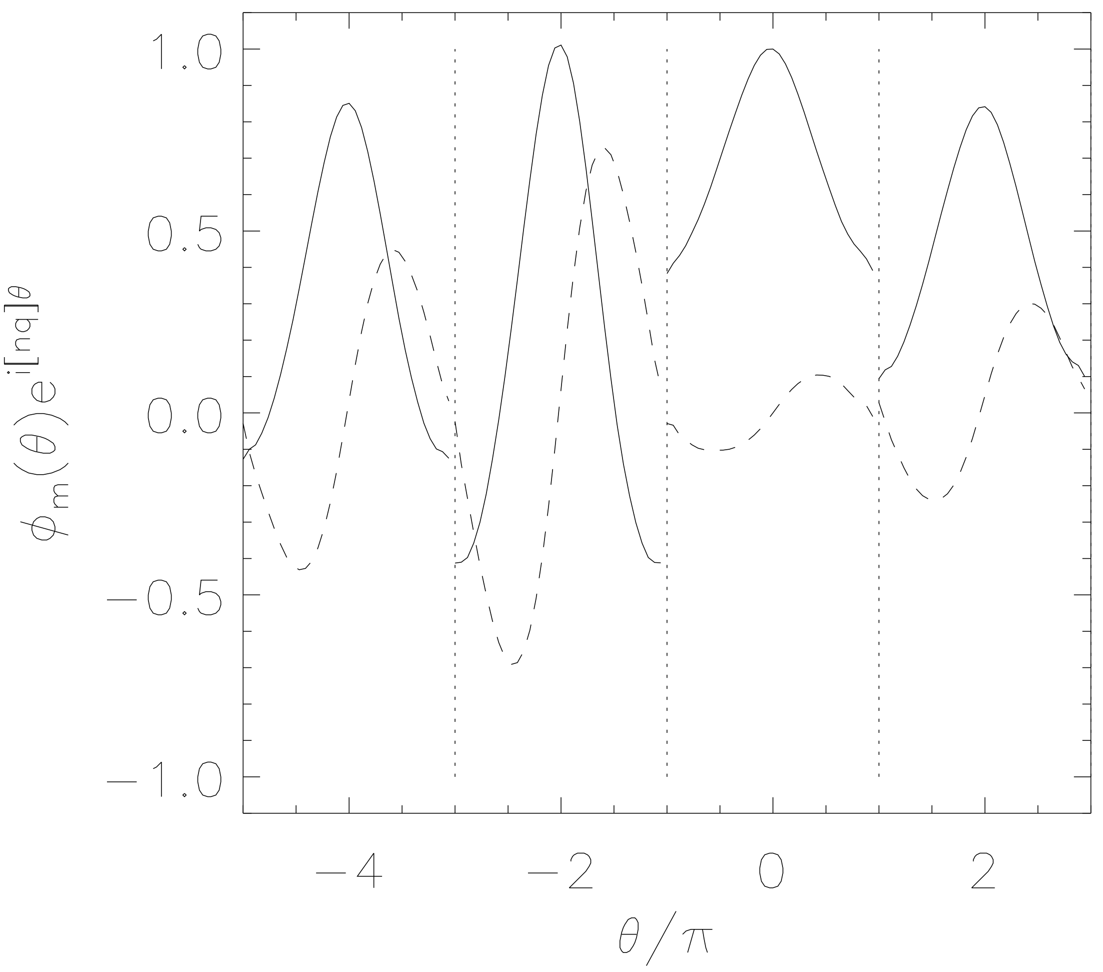
Linear Electromagnetic Benchmark

- Plot of $A_{\|*}(\theta)$; eigenfrequency within 5% of GKS code (Kotschenreuther).



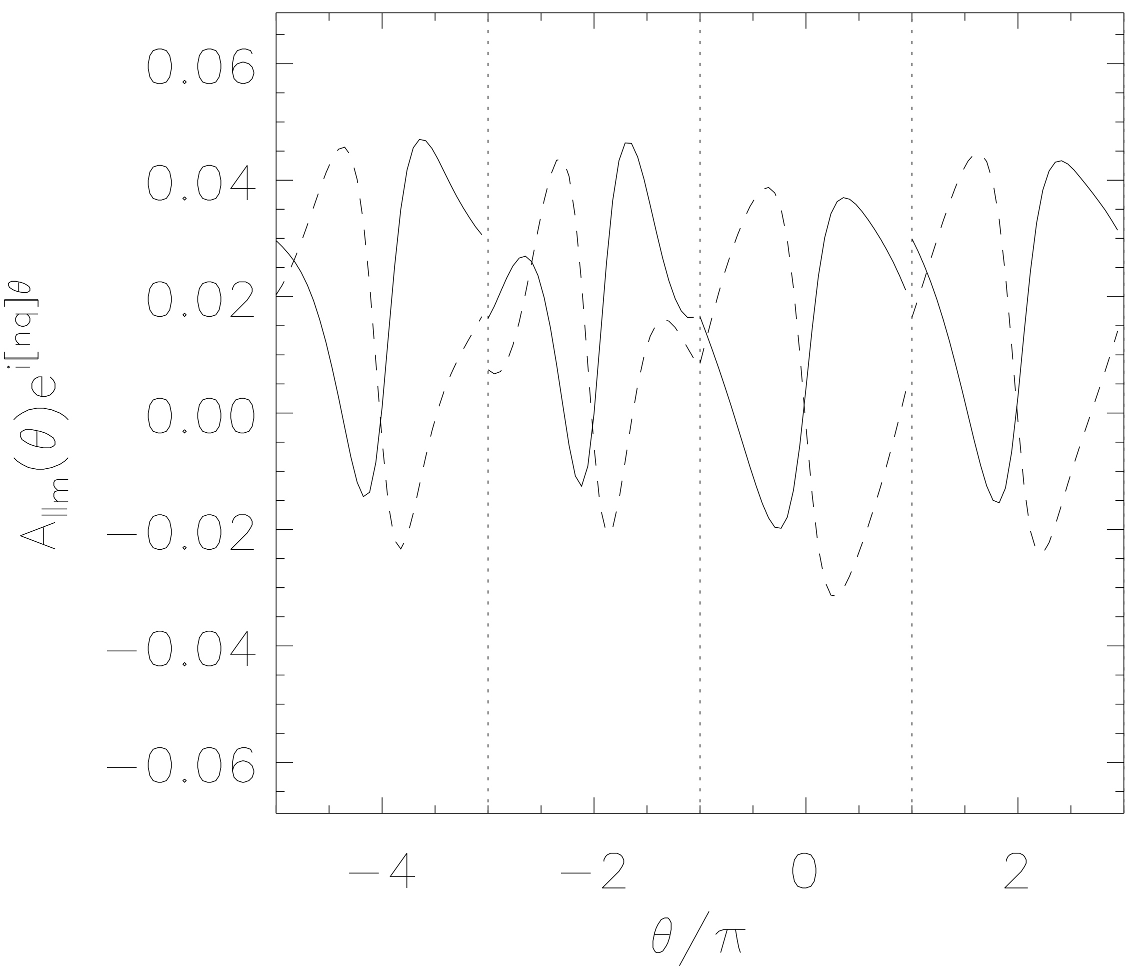
Linear Electromagnetic Benchmark

- Plots of $\phi(\theta)$; eigenfrequency within 5% of GKS code (Kotschenreuther).



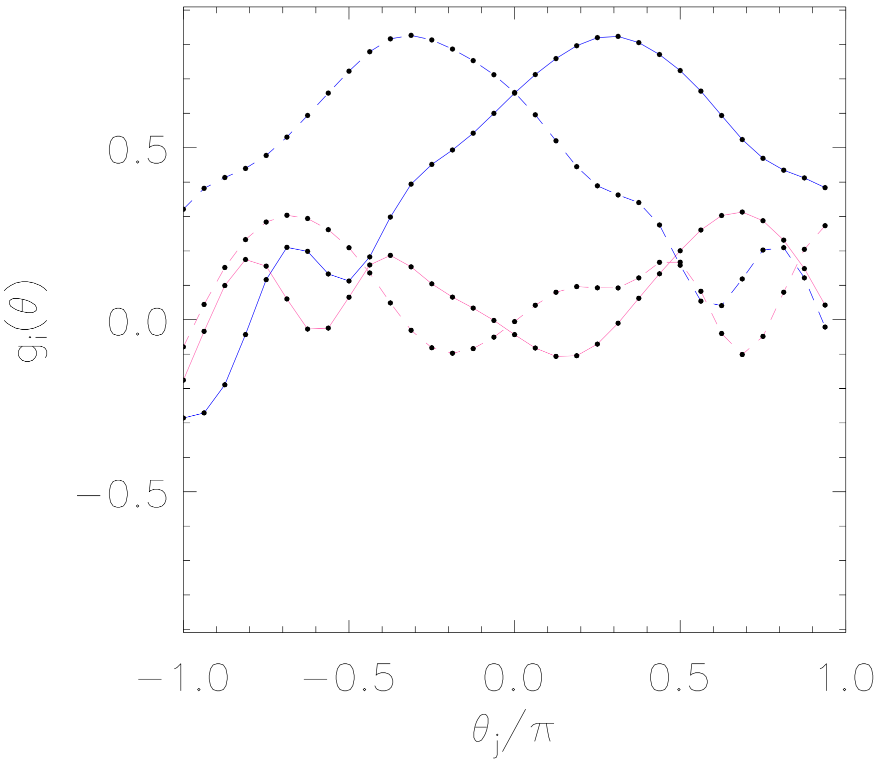
Linear Electromagnetic Benchmark

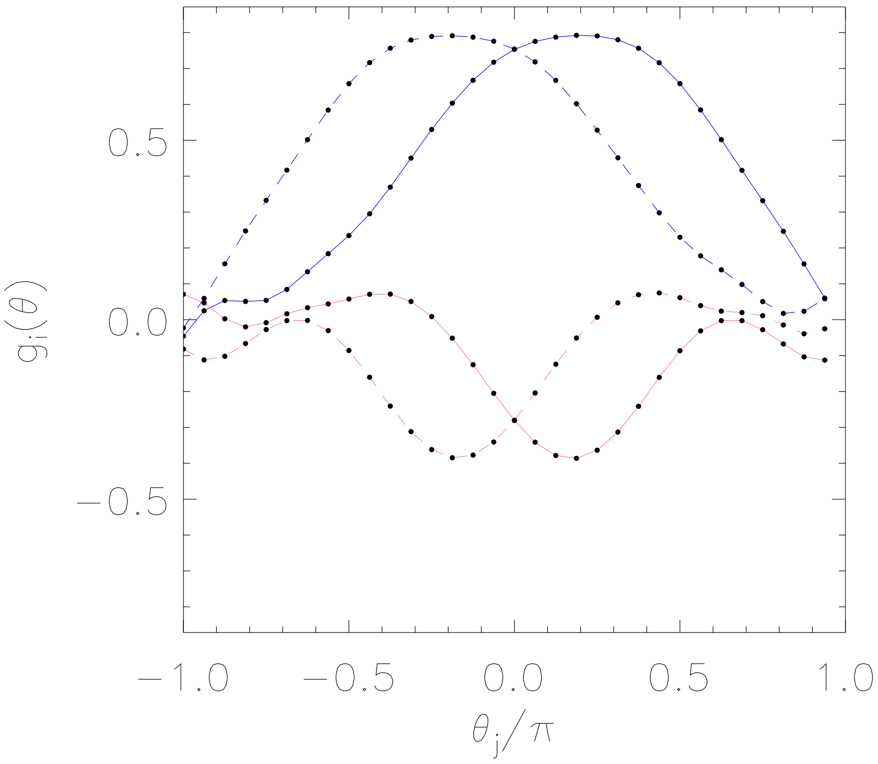
- Plots of $A_{\|}(\theta)$; eigenfrequency within 5% of GKS code (Kotschenreuther).



Example: Ion distribution

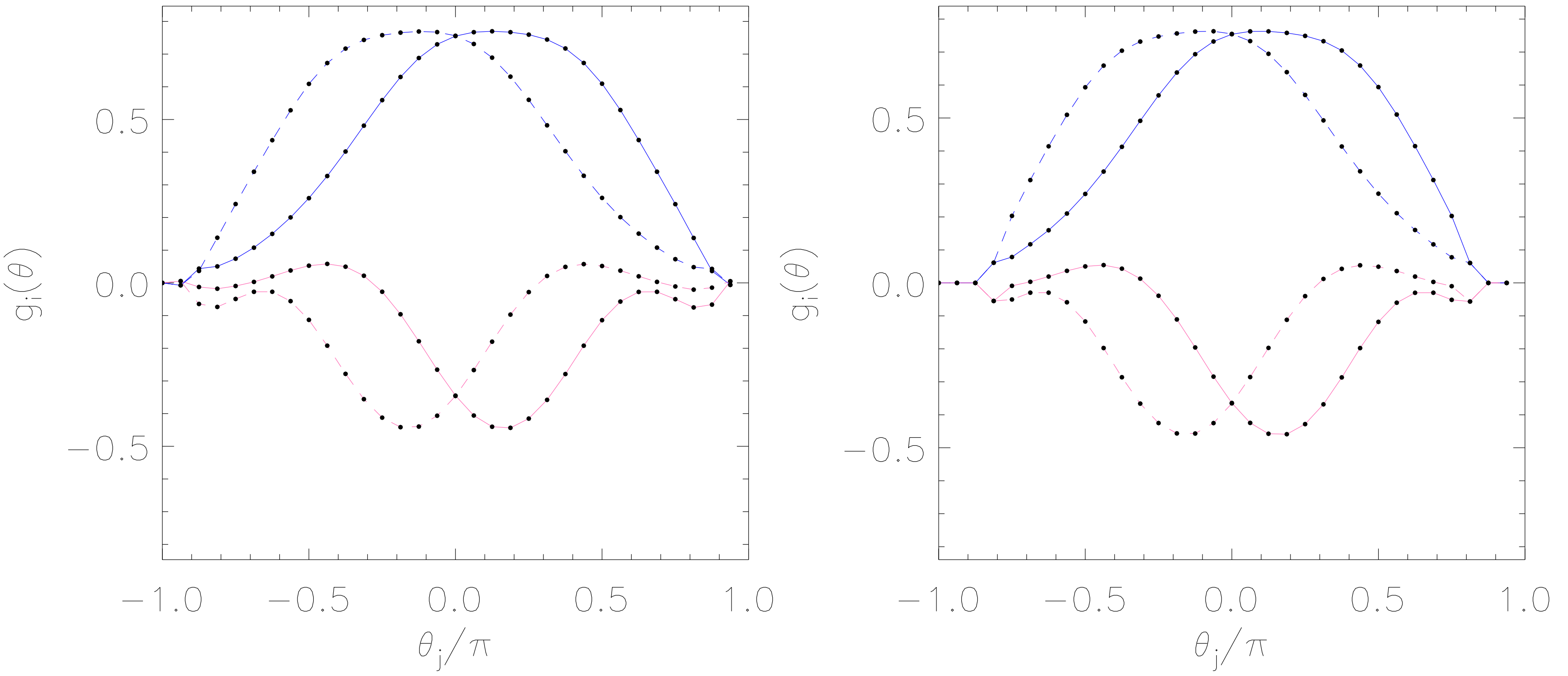
- $i_k = 0, 2$





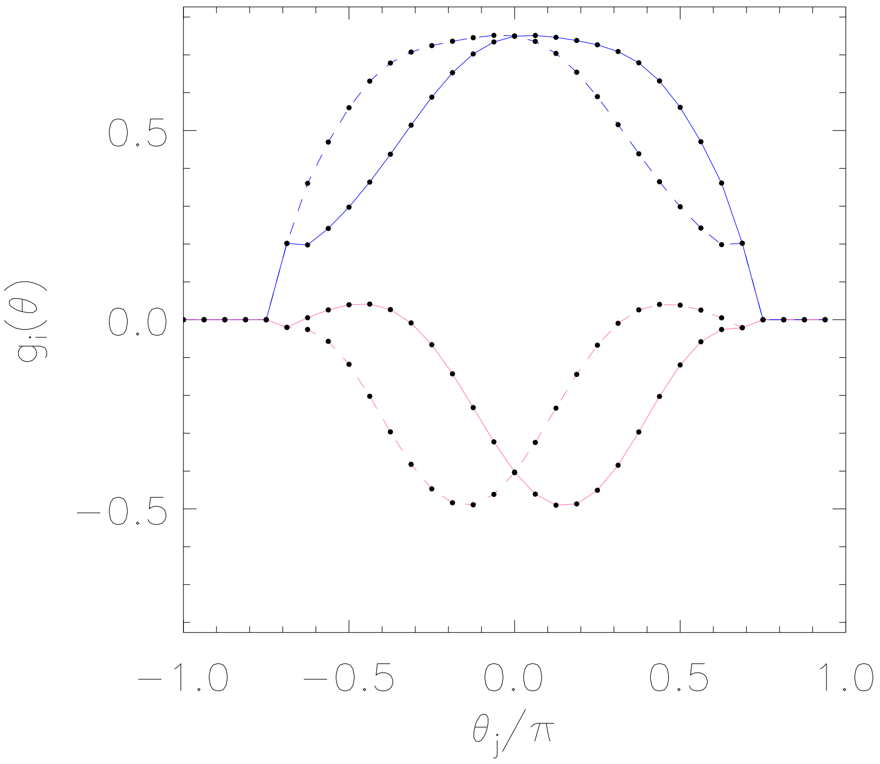
Example: Ion distribution

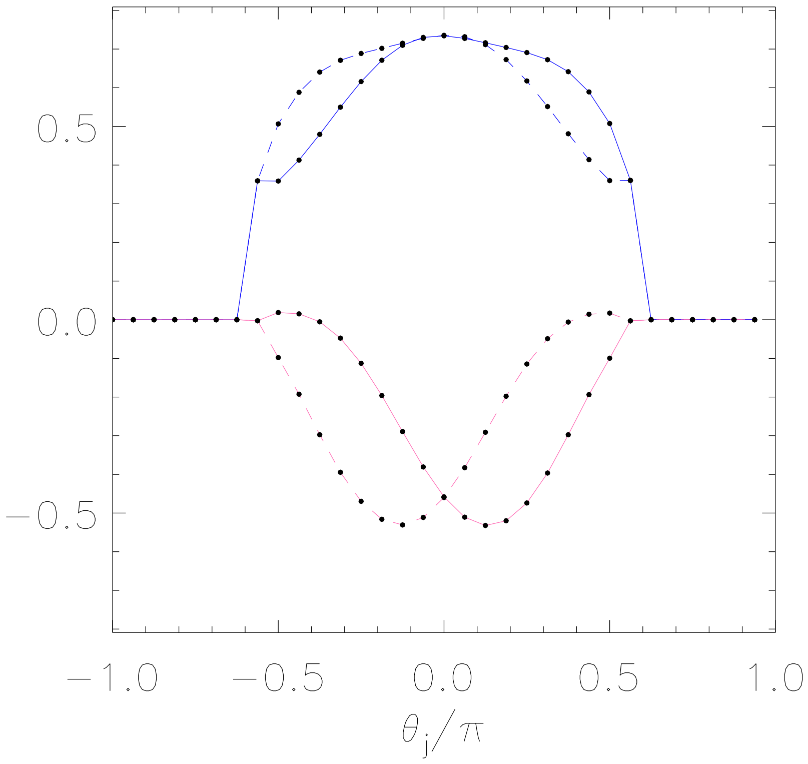
- $i_k = 4, 6$



Example: Ion distribution

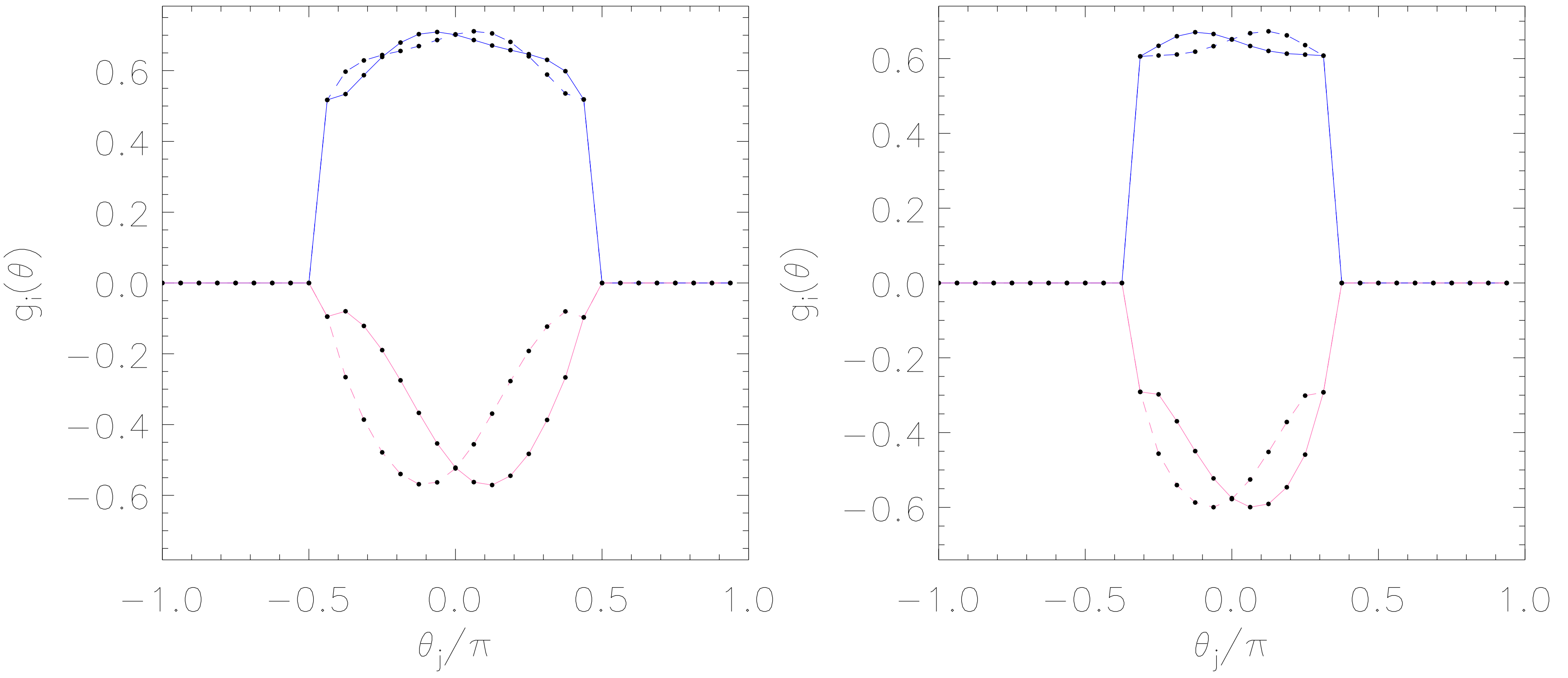
- $i_k = 8, 10$





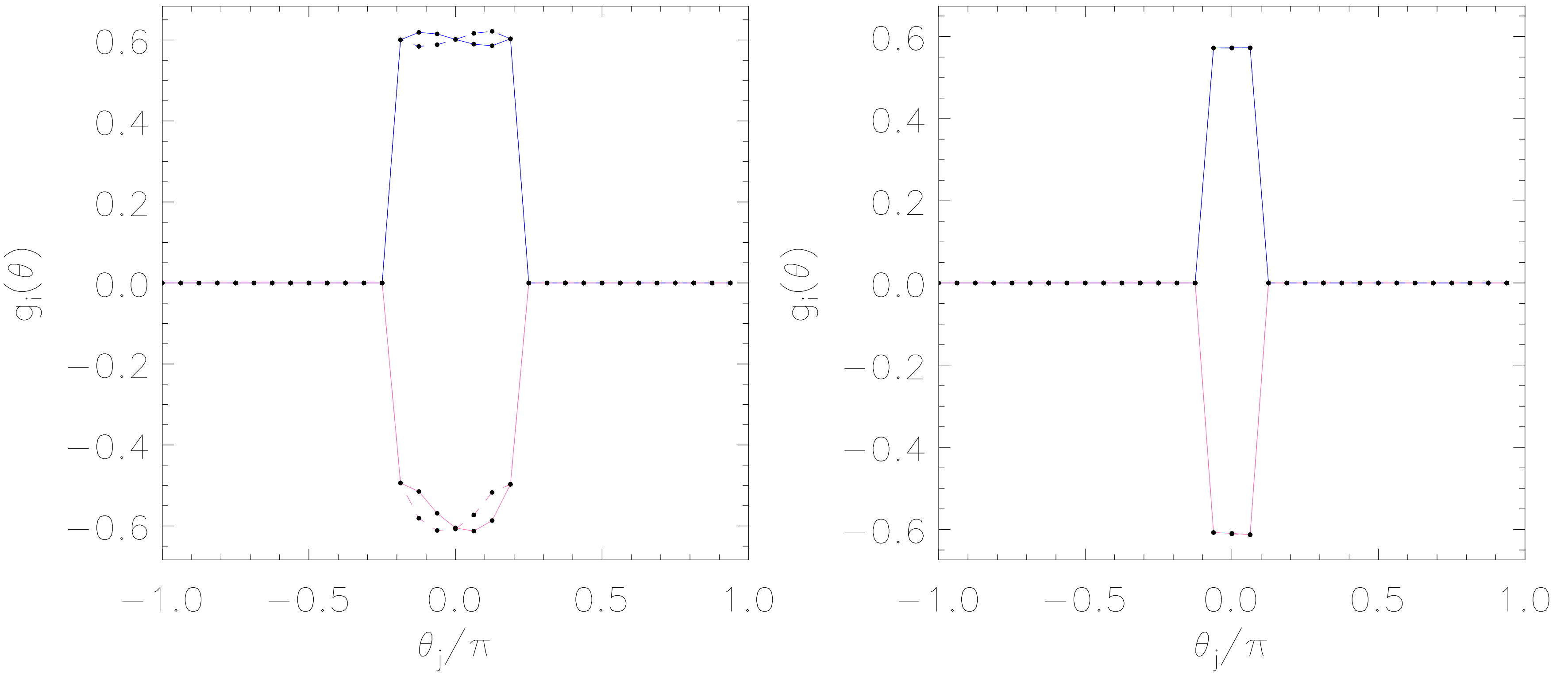
Example: Ion distribution

- $i_k = 12, 14$

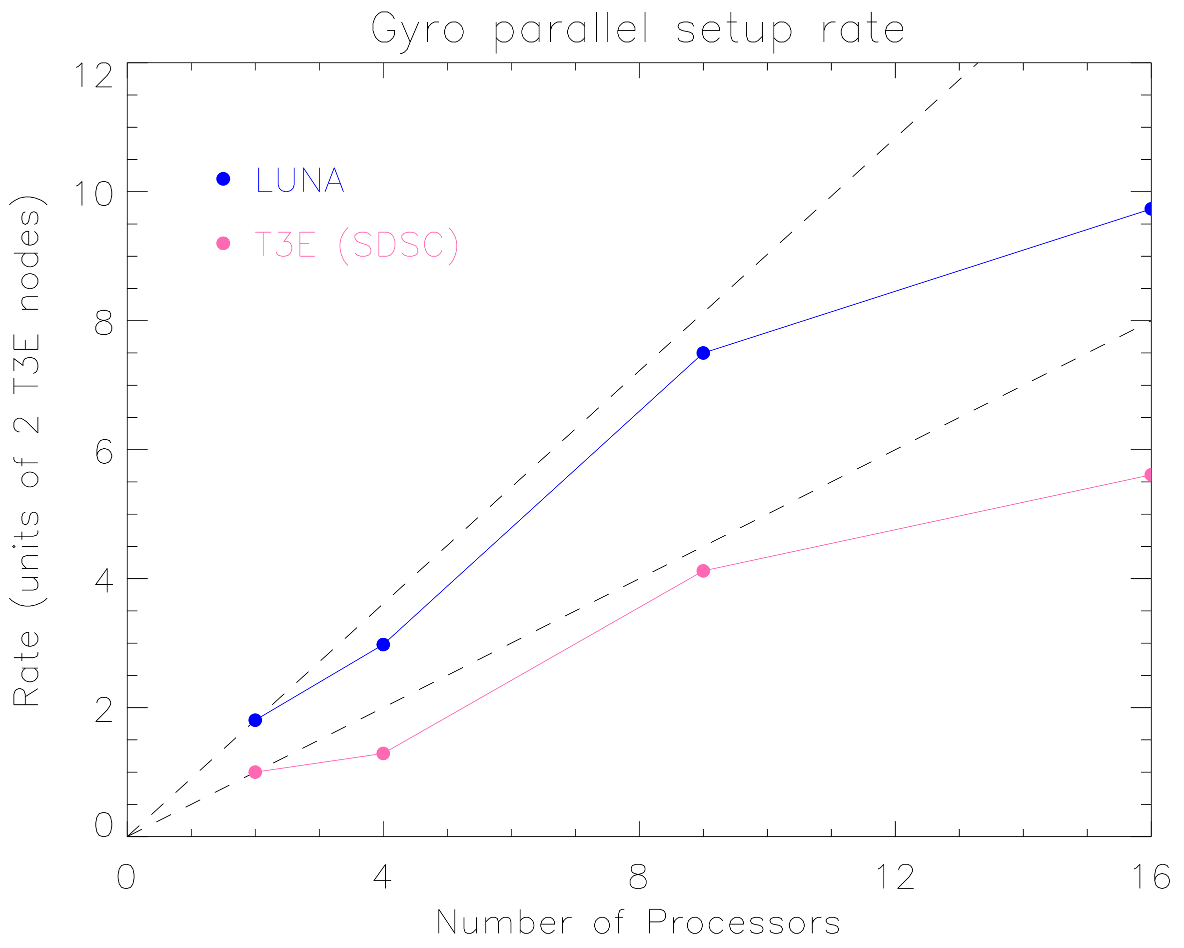


Example: Ion distribution

- $i_k = 16, 18$

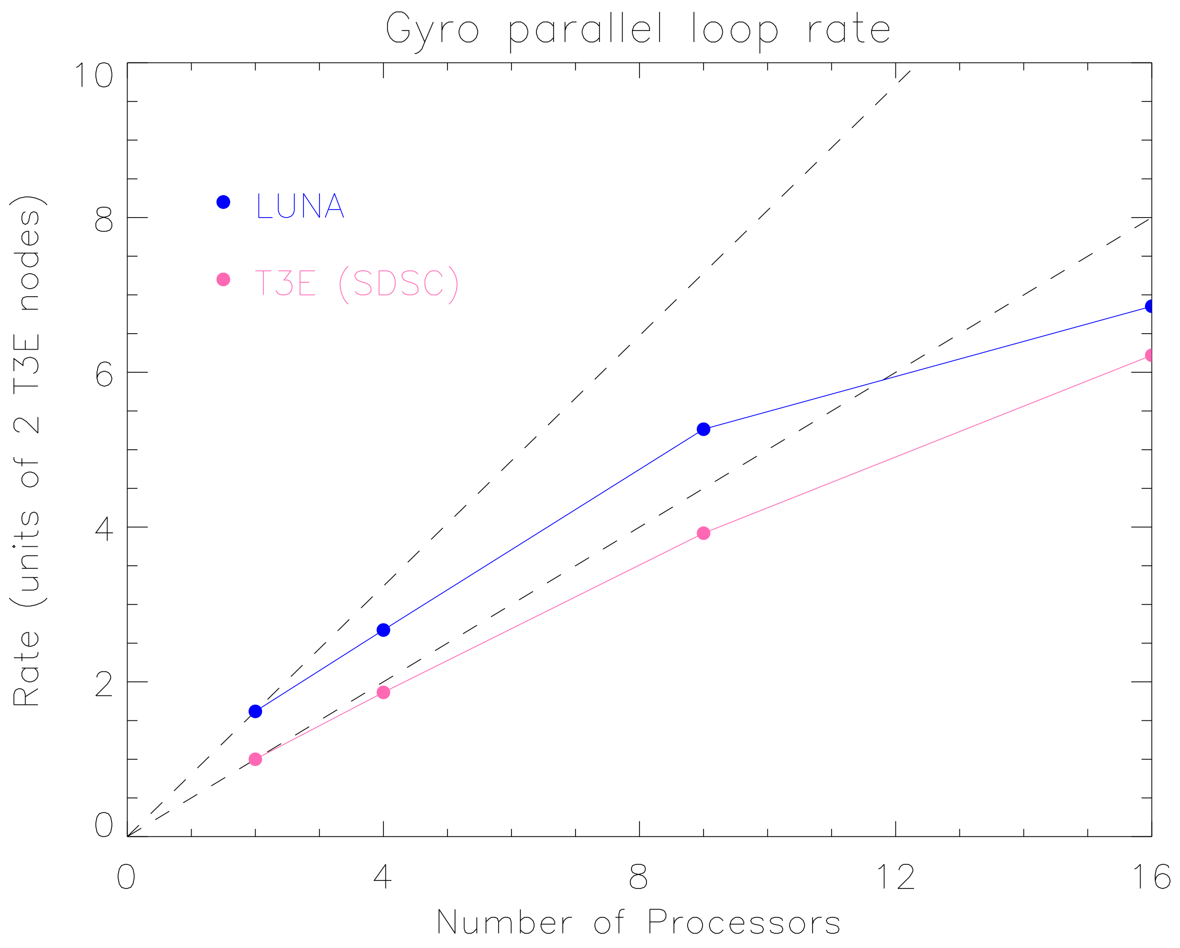


LUNA vs. T3E (SDSC)

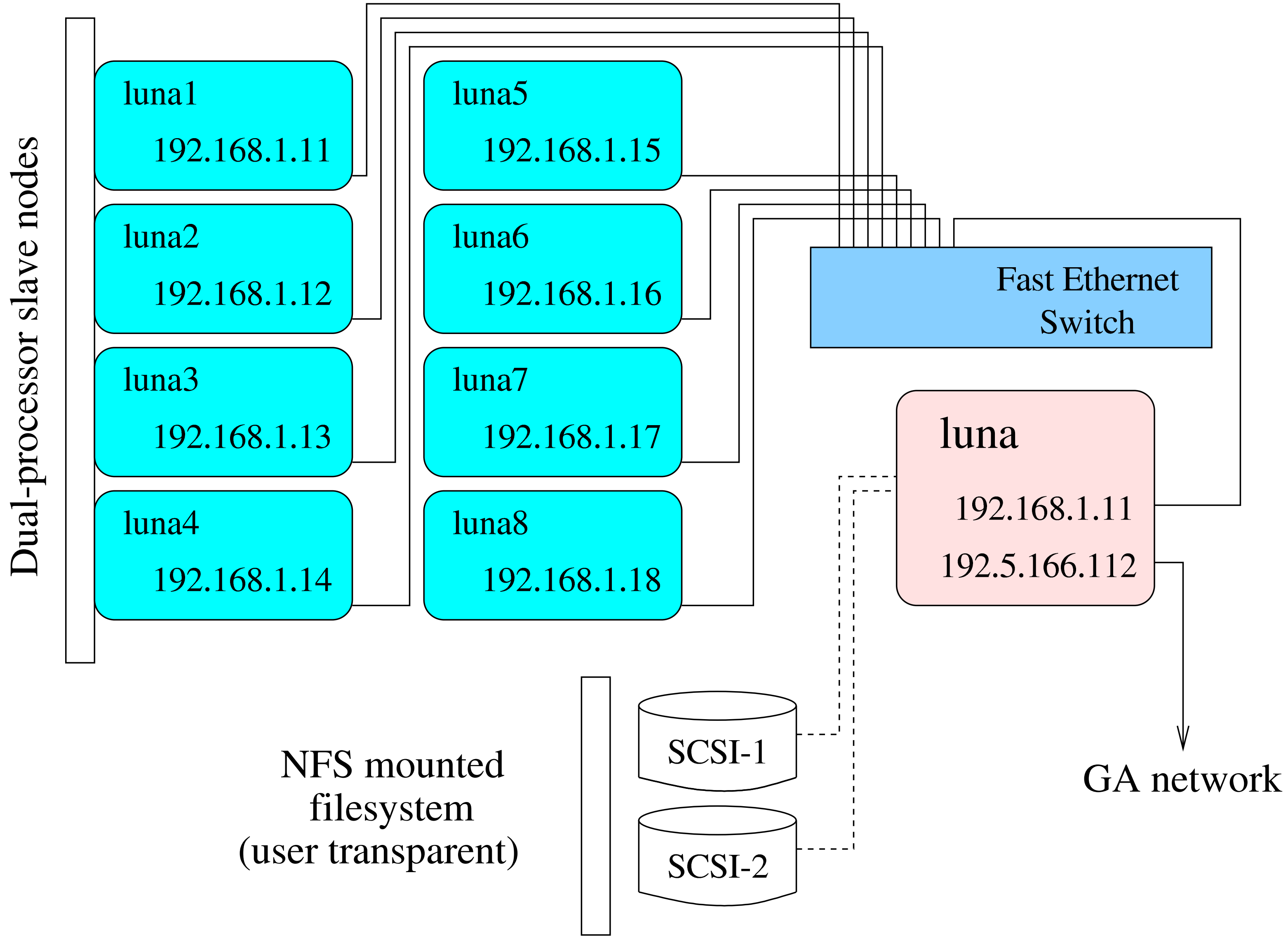


Gyro parallel setup rate

LUNA vs. T3E (SDSC)



LUNA Architecture



Dual-processor slave nodes

luna1 192.168.1.11

luna2 192.168.1.12

luna3 192.168.1.13

luna4 192.168.1.14

luna5 192.168.1.15

luna6 192.168.1.16

luna7 192.168.1.17

luna8 192.168.1.18

Fast Ethernet Switch

luna
192.168.1.11
192.5.166.112

GA network

SCSI-1

SCSI-2

NFS mounted
filesystem
(user transparent)

Software

- Operating System:

  RH Linux 5.2

  2.2.13 (SMP) kernel with Loncaric's TCP/IP patch

- Serial and Parallel Linear algebra packages:

  BLAS, pBLAS, BLACS, LAPACK, scaLAPACK.

- Compilers:

  PGI Workstation (f77, f90, c, c++, profiler, debugger).

- Message Passing Library:

  MPICH

- Utilities:

  xemacs, tcl/tk, ...