Models of Black Holes and Black Box Models

Andrew Benson

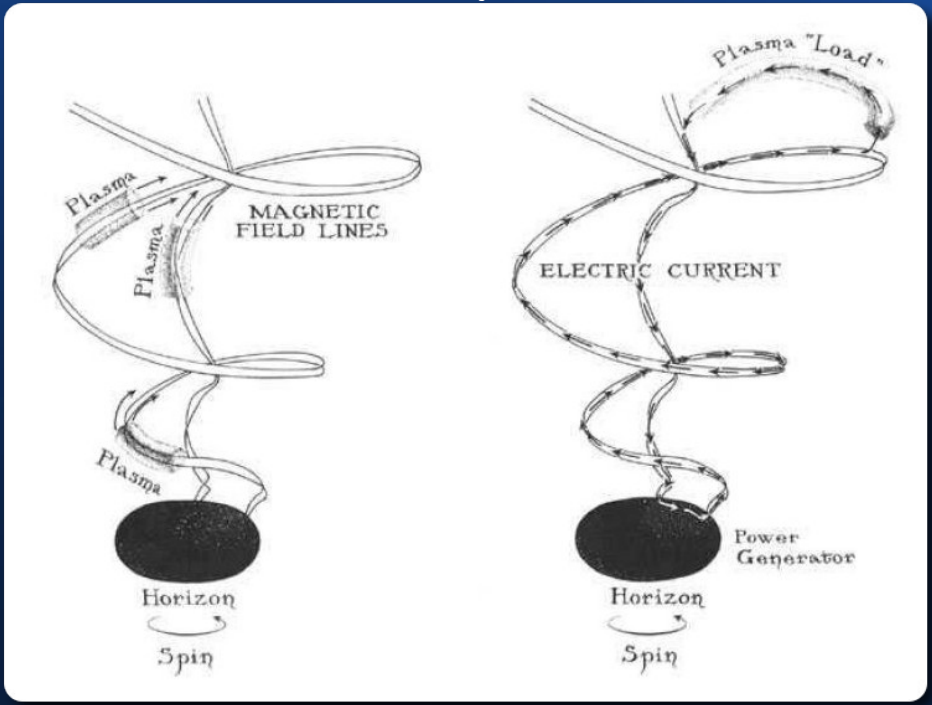
California Institute of Technology

Overview

- Models of black hole spins and jet power
  - Crucial for AGN feedback models
  - Useful fitting formulae
- New model of galaxy formation
  - Emphasis on extensibility and flexibility

Black Holes in Galaxy Formation

Blandford-Znajek Process



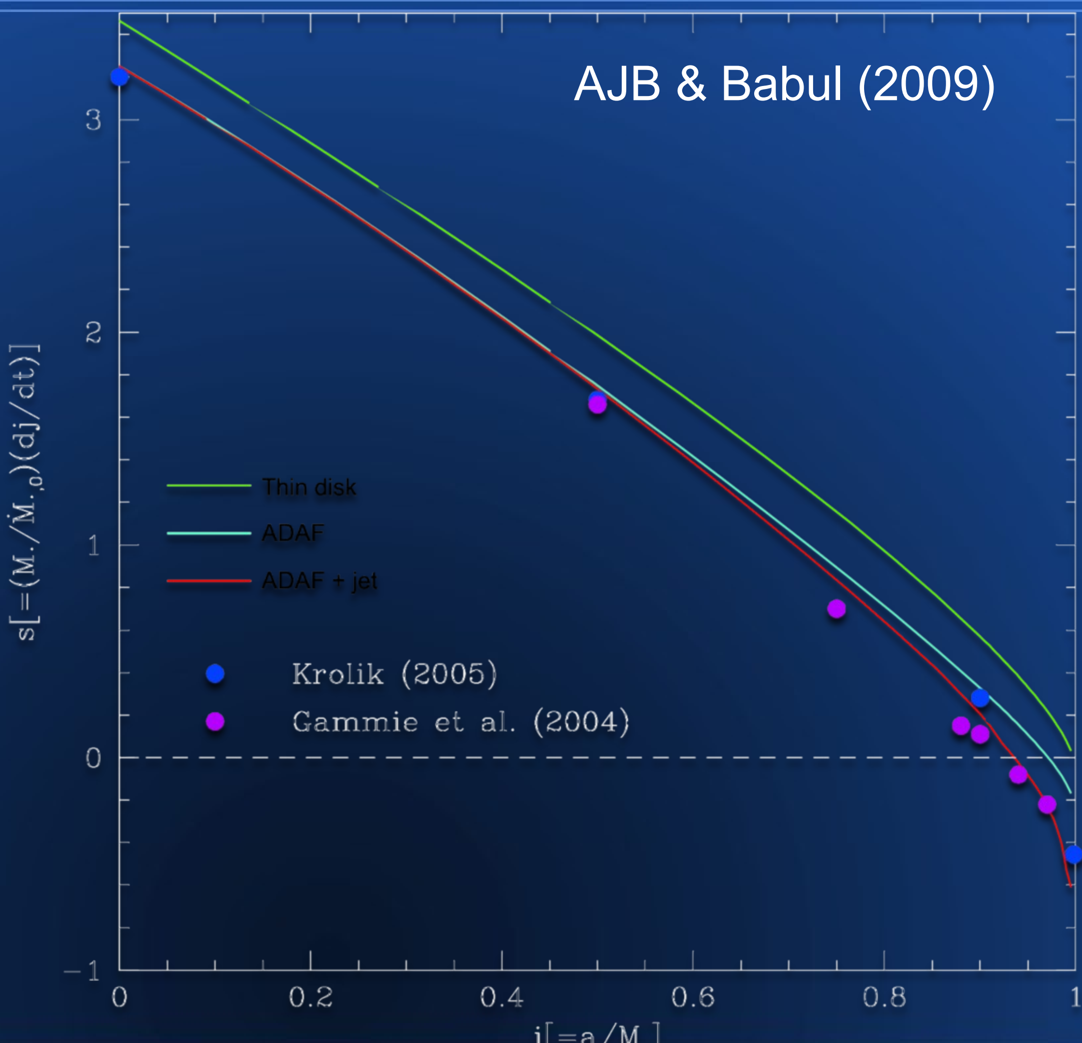
Thorne (1994)

- AGN feedback
  - Key process in galaxy formation
- Modeling
  - Accretion system
  - Black hole
  - Jets/Outflows
  - Feedback

Modeling Black Hole Spin/Jets

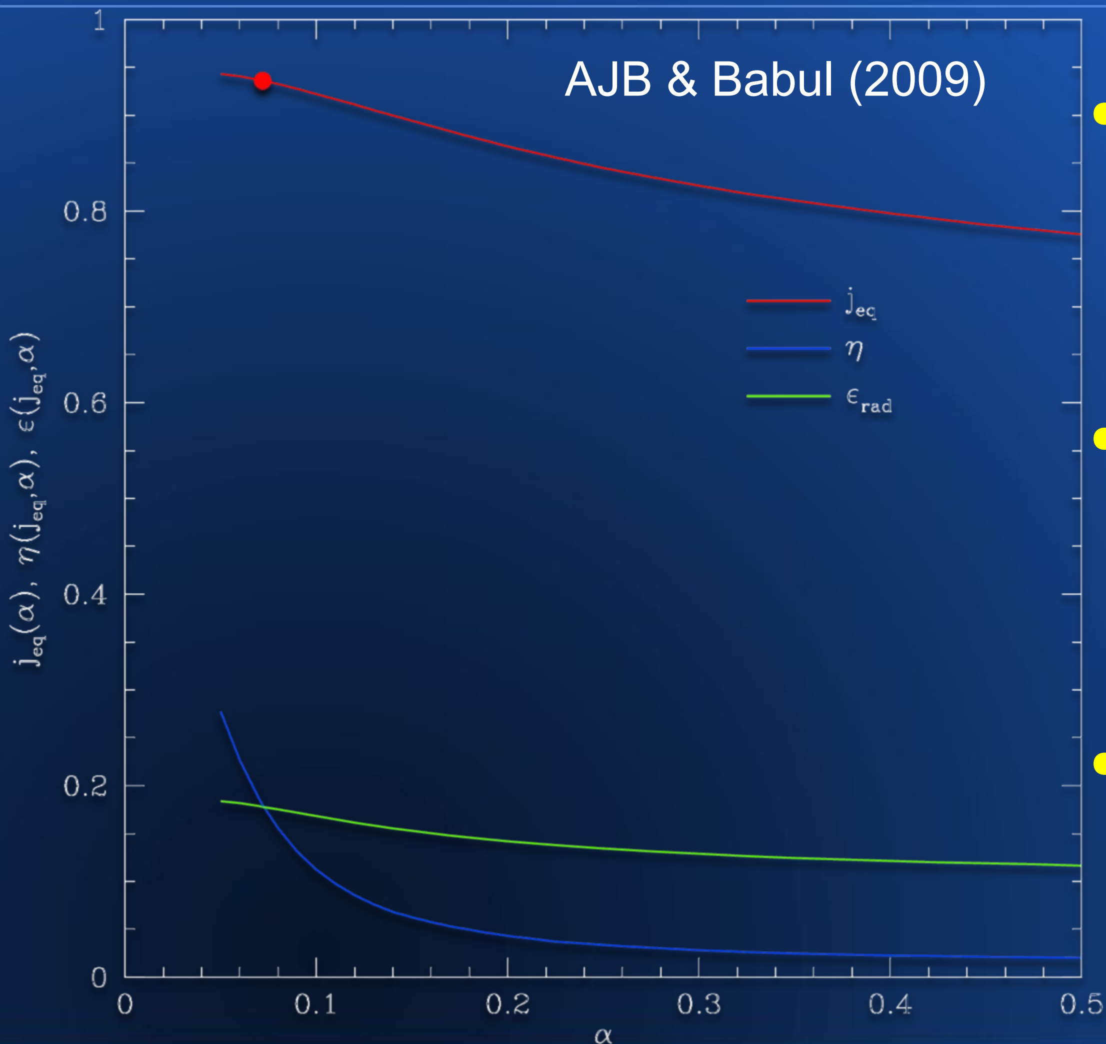
- AJB & Babul (2009)
  - Model of advection dominated accretion flow (ADAF)
    - Fitting formulae to Narayan & Yi numerical solutions
  - Estimate flow properties in Kerr metric
  - Blandford-Znajek etc. estimate of jet power
    - From black hole spin and disk rotation
  - Calculate torques due to power extracted from hole

Spin Up of Black Holes



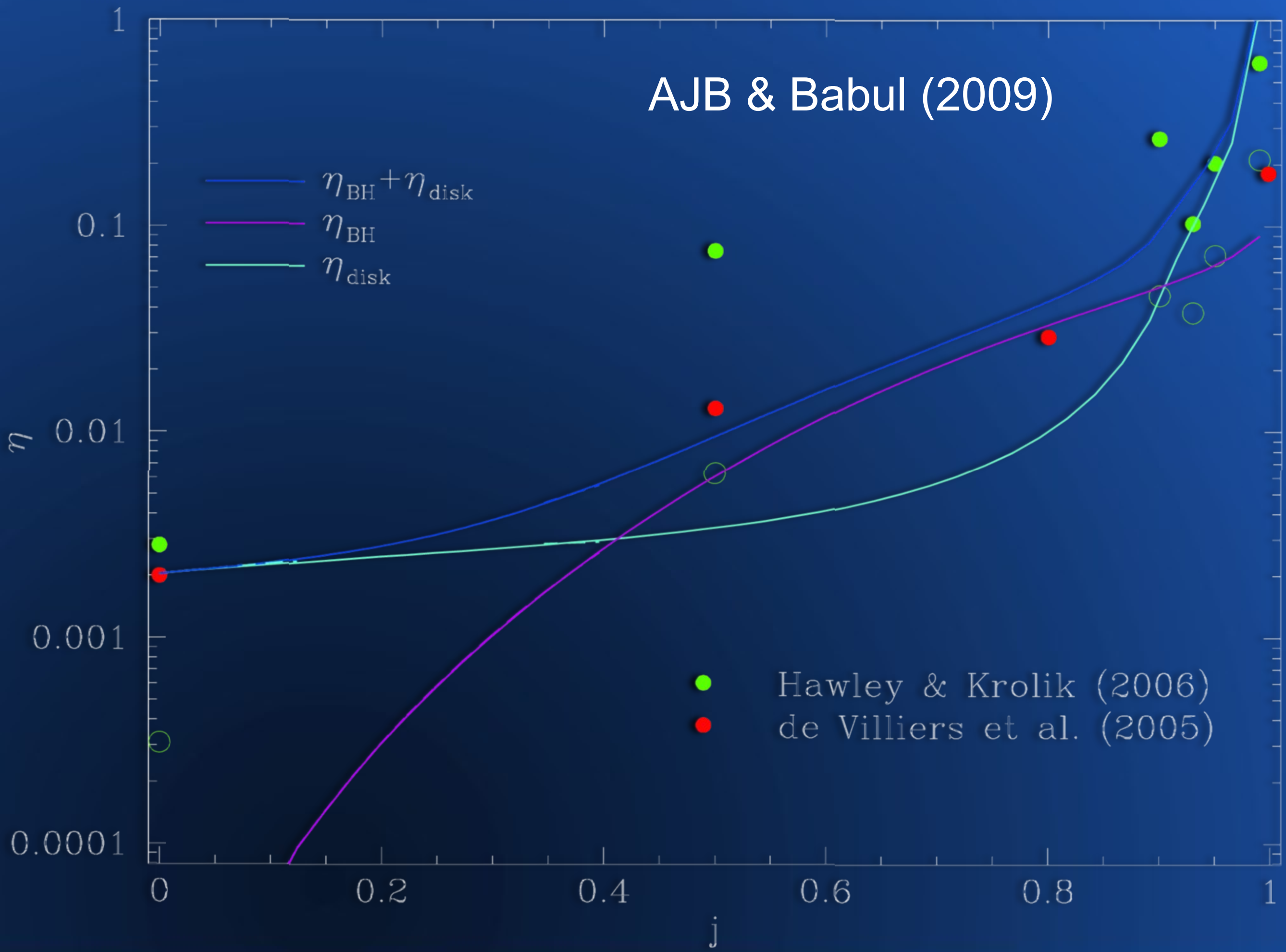
- Spin up by angular momentum of accreted matter
- Spin down by torques that drive jets
- Possibility of equilibrium spin

Equilibrium Spin



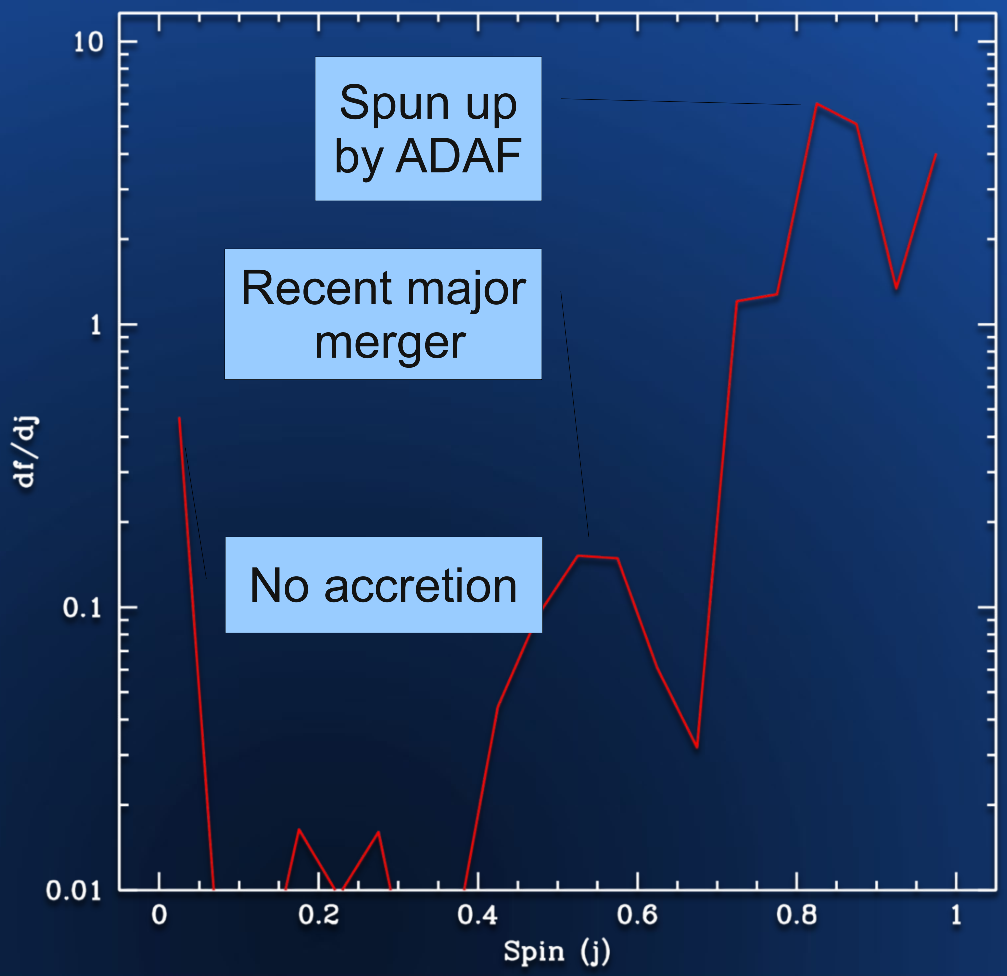
- Equilibria at *j*=0.80–0.95 depending on flow structure
- Radiative efficiencies ~10-20% (for thin disks)
- Jet efficiencies of up to 30%....

Black Hole Jet Efficiencies



- Disk and black hole contribute
- High efficiencies for high spins

Black Hole Spin Distribution



- Evolve spins in cosmological context
  - Galacticus semi-analytic model
  - Bondi-Hoyle accretion
  - Merging
  - Jets

Black Holes Summary

- Jet power of black holes is a crucial ingredient for AGN feedback models
  - AJB & Babul (2009) provides simple fitting formula for:
    - ADAF structure
    - Jet power
    - Spin-up/down rate
  - Predicts equilibrium spin j~0.8 - 0.95
  - Jet efficiency at equilibrium η~15%

A New Semi-Analytic Model

- Why?
  - Adding in new features (e.g. self-consistent reionization, noninstantaneous recycling, new star formation rules) should be easy
  - Permit user to focus on physics
- How?
  - Create a code which is modular by design, isolating assumptions so that they don't have consequences throughout the code.

GALACTICUS

- Freely available for anyone to use
- Modular design
  - Each function can have multiple implementations, selected by input parameter.
  - “Node” can have arbitrary number of components (e.g. DM halo, disk, spheroid), all with multiple implementations
- Combination of smooth (ODE) evolution and instantaneous events (e.g. mergers)

Modularity

- New implementation of function easily added:
  - Write a module containing the function
  - Add directives indicating that this function is for disk star formation timescale calculations
  - Recompile – build system automatically finds this new module and works out how to compile it into the code

Modularity

- Modules are self-contained and independent
- Self-initializing and recursive
- For example – deterministic halo spins:
  - Request spin of halo
  - Module reads in parameters of model, initializes
  - Needs spins of progenitor halos, so calls itself for those nodes...
  - ...which call the same routine for their progenitors....

Node Components

- Component could be, e.g. disk (exponential)
- Stores various types of data:
  - Properties – evolved within ODE system
  - Data – internal data, not evolved
  - Histories – records of past/future history (e.g. star formation history)
- Allowance for multiple components of each type (coming soon.....)

Node Components

- Defining a component:
  - Set of ODEs giving rates of change of properties (can access properties of other components/nodes as needed)
  - Responses to events (merging, becoming satellite etc.)
  - Specify properties to be output

Node Evolution

- Code repeatedly walks tree – finds nodes that it can evolve:
  - Cannot evolve if still have children
  - Can't evolve beyond their satellites
  - Limit on time step
  - Arbitrary other factors can be included
- Evolve those nodes forward in time
- Stops when no more nodes to evolve

Node Evolution

- All component properties fed into ODE solver
- Evaluate derivatives – evolve forward in time
- No need for fixed timesteps or analytic solutions
    - Makes implementing, for example, Kennicutt-Schmidt law trivial (just add new star formation timescale function)
- Evolution can be interrupted as needed (e.g. when galaxy merges)

Node Evolution

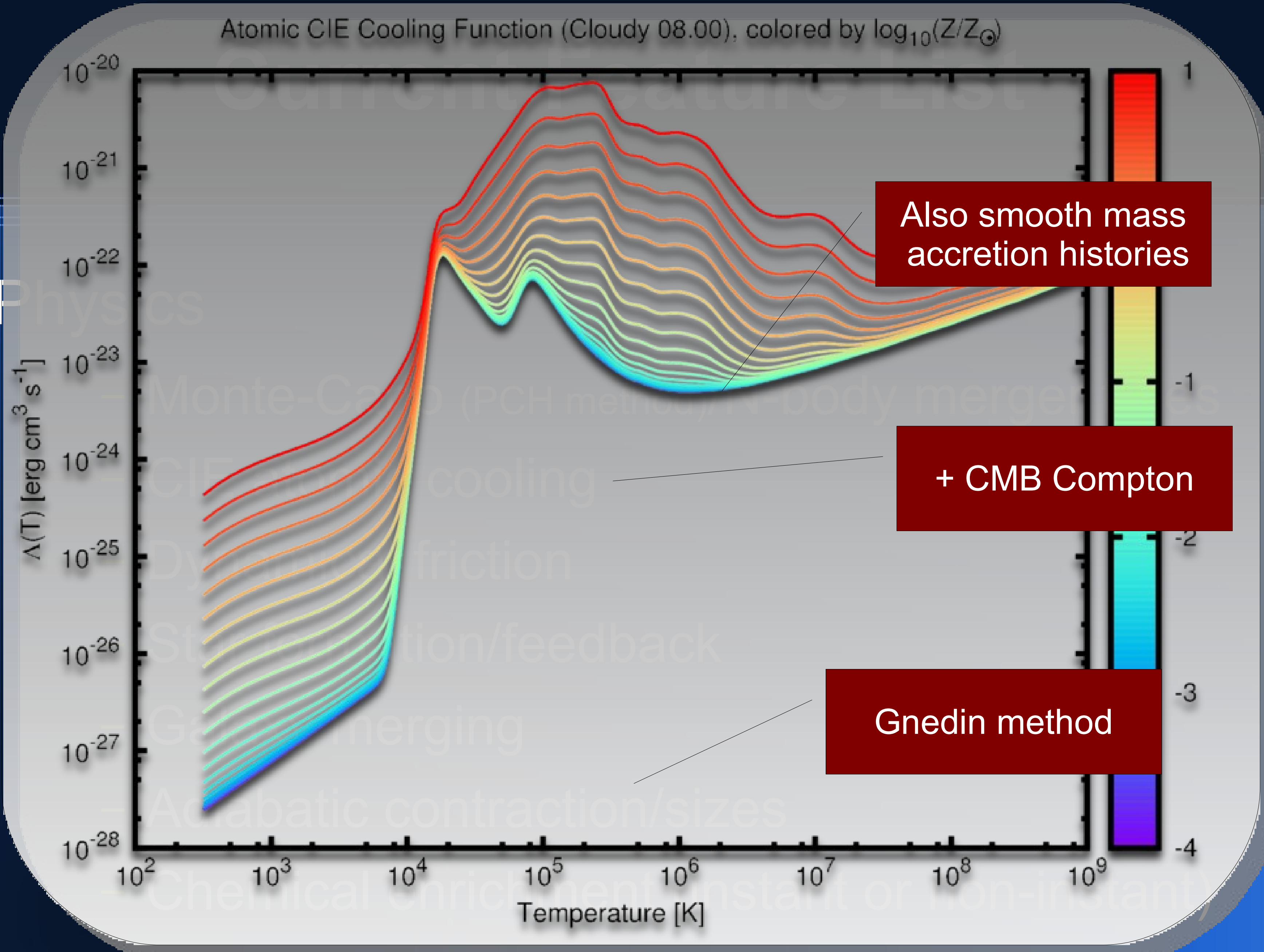
- Component creation:
  - Nodes begin with only basic component (mass, time)
  - If accretion from IGM occurs, stop and create a hot halo component
  - If cooling occurs, stop and create a disk component
  - Components can be destroyed as needed also

Advantages

- Modularity makes it highly flexible:
  - Add new star formation rule in 5 minutes
  - Change in cooling model confined to few modules which compute cooling time and rate
- Unified ODE solver makes new features simple:
  - Time stepping handled automatically
  - No need for analytic solutions
  - Implemented noninstantaneous recycling in one afternoon rather than two months....

Current Feature List

- Components
  - Dark matter profile [isothermal/NFW]
  - Hot halo
  - Disk [exponential]
  - Spheroid [Hernquist]
  - Black holes (grow via Bondi-Hoyle accretion)
  - plus components that track things such as spin, merging time etc.

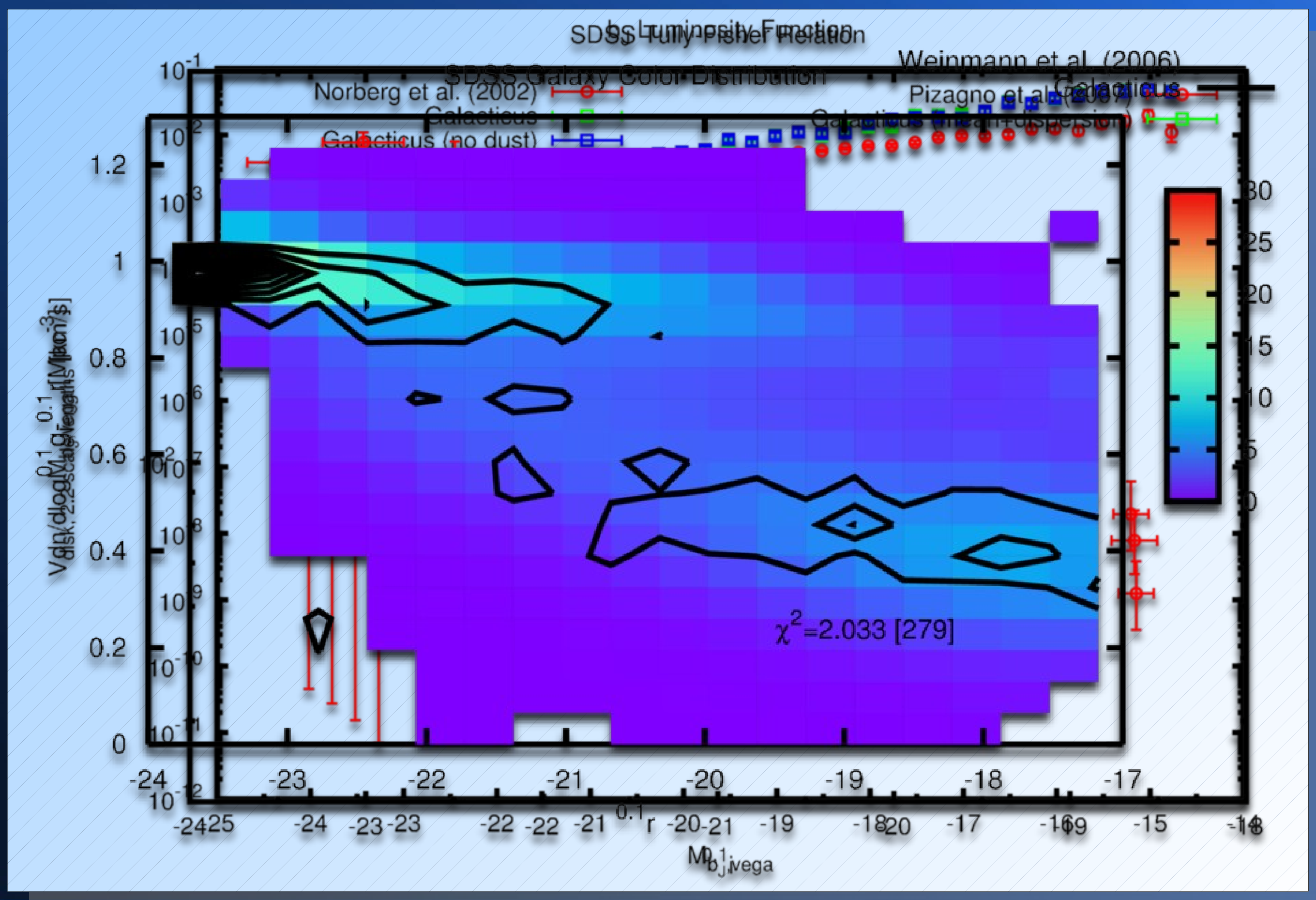

Atomic CIE Cooling Function (Cloudy 08.00), colored by $\log_{10}(Z/Z_\odot)$
$\Lambda(T)$ [erg cm^3 s^{-1}]
Temperature [K]
10^{-20}
10^{-21}
10^{-22}
10^{-23}
10^{-24}
10^{-25}
10^{-26}
10^{-27}
10^{-28}
10^2
10^3
10^4
10^5
10^6
10^7
10^8
10^9
1
-1
-2
-3
-4
Also smooth mass accretion histories
+ CMB Compton
Gnedin method

Current Feature List

- Physics *(cont.)*:
  - Disk instabilities
  - Black hole merging
  - AGN feedback
  - Stellar population synthesis (with arbitrary IMF)

GALACTICUS

http://sites.google.com/site/galacticusmodel/



Conclusions

- AGN feedback/Black hole models
  - AJB & Babul (2009) provides fitting formula for:
    - ADAF structure
    - Black hole jet power and spin
- Galacticus model
  - Free and open source
  - Extensible/flexible
  - Try it!