TRANSP Pellet Ablation Model

J. S. Sachdev

TRANSP User Group Meeting 2017

May 4-5, 2017

Computational Plasma Physics Group

Princeton Plasma Physics Laboratory

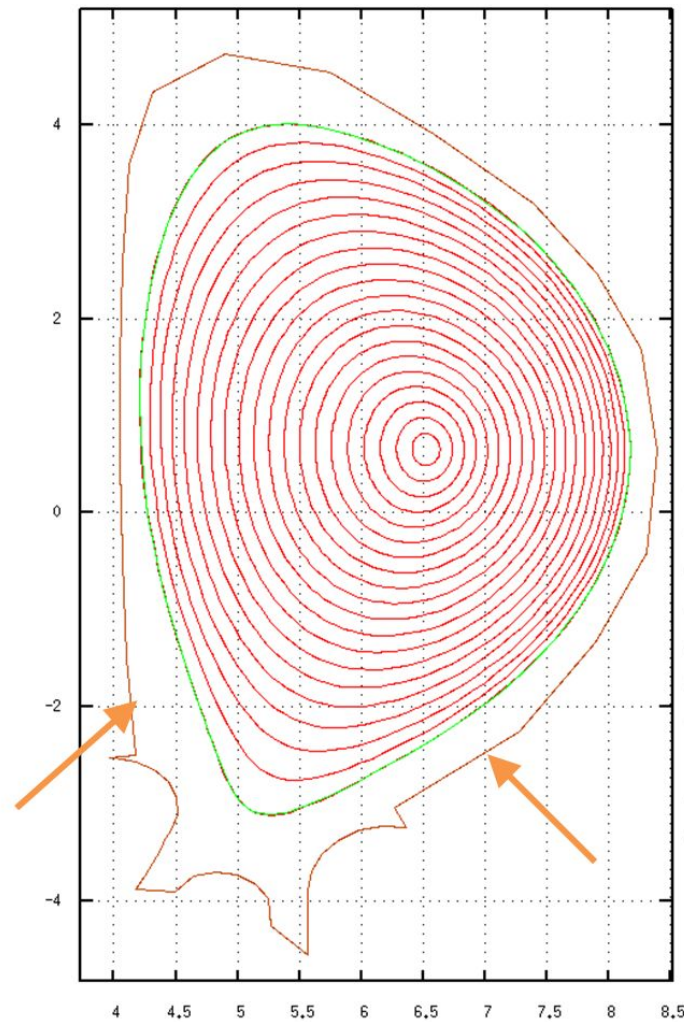
Motivation and Objective

- The ability to model pellet injection is important for a number of tokamaks
- NSTX-U, DIII-D, and EAST use Lithium Granule Injection (LGI) for wall conditioning and Edge-Localized Mode (ELM) mitigation
- MAST-U, ASDEX-U, JET, and ITER will be using pellet injection for both fueling and ELM mitigation
- Original pellet ablation model removed from TRANSP many years ago
- Some code infrastructure supporting pellet events left in place, including:
  - Input parameters
  - Code that schedules and performs pellet events
  - Actual model used not retained in SVN or CVS repositories
- The objective of this project is to reinstate a pellet ablation capability and extend to more modern models

Pellet Injection

- In TRANSP, pellet injection is treated as an ‘event’
    - Pellet injection schedule is checked at the beginning of a time-step
    - The pellet is ‘popped’ if time for the next injection has been reached
    - Similar to how saw-tooth events are processed

- Event steps:
    - Pellet event is triggered
    - Pellet trajectory through plasma is computed
    - Electron density and temperature profiles along the trajectory are saved
    - Marching done in physical space; compute ablation along path
    - Ablation along the path is computed until the pellet radius is zero or leaves the plasma
    - Change in electron density and temperature is mapped back into flux coordinates



Pellet Ablation Models

- Started with pellet ablation model from the Tokamak Simulation Code (TSC)
- Switched to a FORTRAN module developed by Houlberg/Baylor/Parks
  - Includes several ablation models for hydrogenic and impurity pellets
  - Module will be updated as new models are devised
  - Objective: Use module as is, only bug fixes
- Parks' baseline model for hydrogenic pellets:

$$\frac{dN}{dt} = 1.12 \times 10^{16}\, n_e^{0.333}\, T_e^{1.64}\, r_p^{1.333}\, M^{-0.333}$$

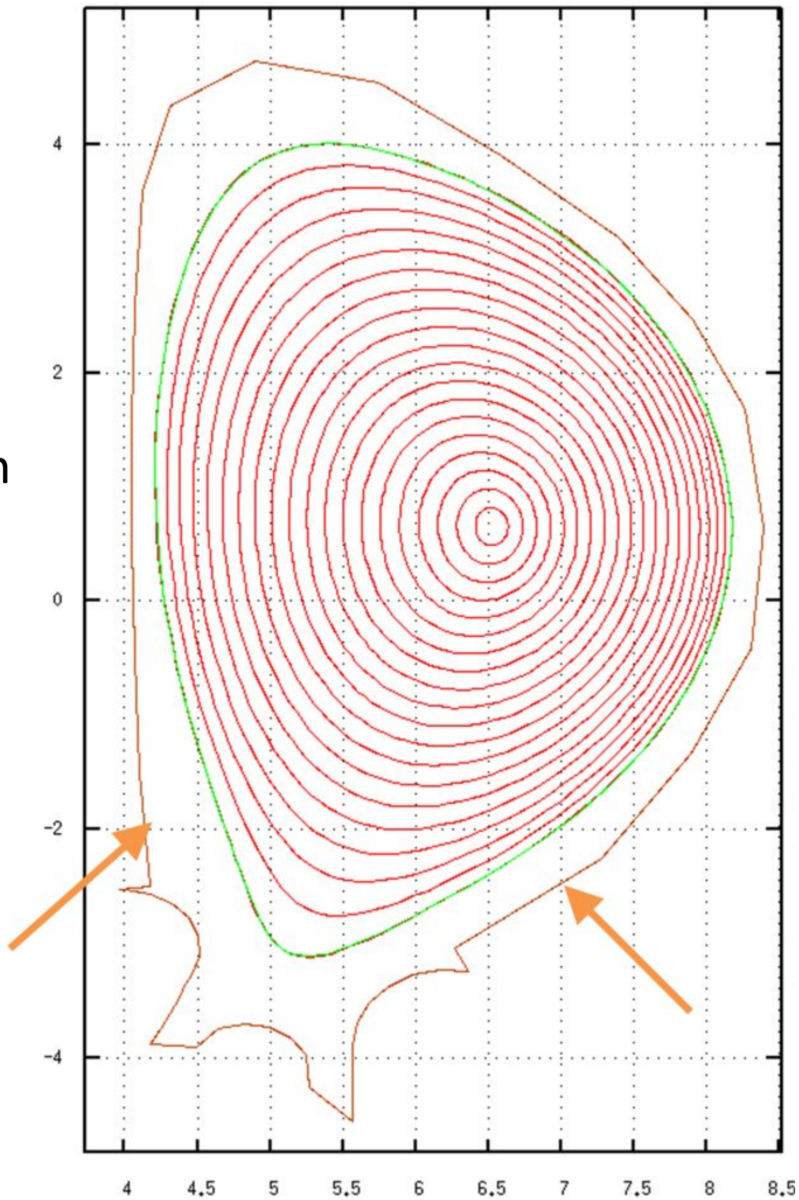
- Other ablation models for hydrogenic pellets:
  - Kuteev's model for hydrogenic pellets for high plasma parameters
  - Macaulay's model for hydrogenic pellets (fit from 2D simulations)
  - Parks' Q-model, determines ablation rate assuming constant heat-flux
  - Parks' IPADBASE model, computes ablation rate from generated fits
  - Neutral Gas Shielding (NGS) model of Houlberg***
- Impurity pellet models:
  - Kuteev's model for Li, Be, B, C, Ne, Ar, Ti, Kr, Mo, Xe, and W
  - Parks' model for Ne, Ar, Kr, and Xe

Schedule and Tasks

		9	10	11	12	1	2	3	4	5	6	7	8
1	Understanding current infrastructure for pellet injection	▶			⬢								
2	Add baseline pellet ablation models			▶			⬢						
3	Extend inputs to include injectors						▶		⬢				
4	Verification with ITER fueling scenario						▶			⬢			
5	Validation with JET fueling scenario								▶		⬢		
6	Upgrade OMFIT to include pellet inputs								▶		⬢		
7	Implement advanced ablation model									▶		⬢	
8	Application to DIII-D ELM mitigation with lithium pellets										▶		⬢
9	Documentation and tutorial											▶	⬢

Pellet Input Parameters for TRANSP

- Figure shows approximate pellet injector locations for an ITER scenario
- High-Field Side injector fires at 16 pellets/s
- Low-Field Side injector fires at 40 pellets/s
- 56 pellets injected after 1 s
- Previously, the user would have had to type in 56 pellet events into the TRANSP input file
- Added capability for the user to specify pellet injectors instead of individual pellet events
- Note: maximum number of pellets is 1000



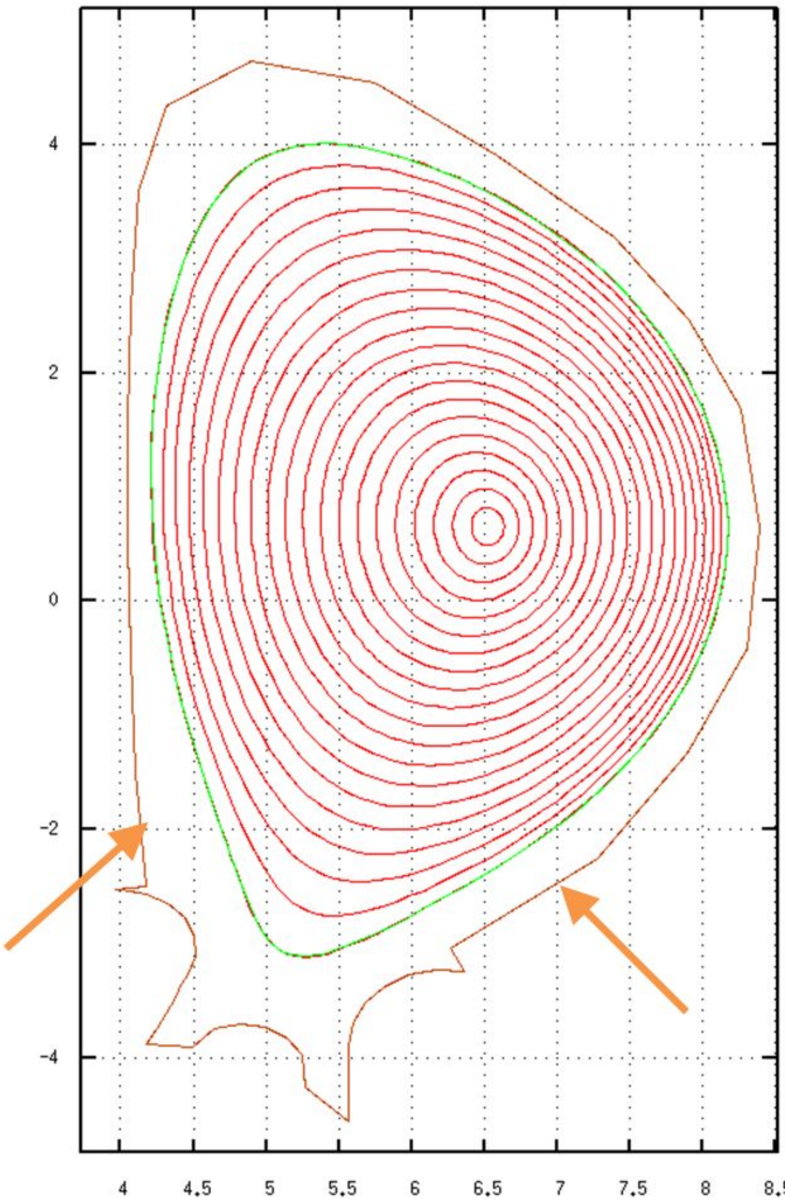
Pellet Input Parameters for TRANSP, cont'd

Variable	Type	Description
ipelinp	`integer`	Pellet input model: 0 = event-based, 1 = injector-based
`npel`	`integer`	Number of pellet events OR number of pellet injectors
`tpel`	`real(npel)`	Time pellet popped OR time pellet injector starts firing [s]
tpelend	`real(npel)`	Time pellet injector stops firing [s]
freqpel	`integer(npel)`	Frequency pellets are injected [Hz]
`aplrsta`	`real(npel)`	Pellet starting r-position [cm]
`aplysta`	`real(npel)`	Pellett starting y-position (+ve above midplane) [cm]
`aplphia`	`real(npel)`	Toroidal aiming angle [degrees]
`aplthea`	`real(npel)`	Poloidal aiming angle (+ve up) [degrees]
`apel`	`real(npel)`	Atomic weight of pellet material
`pelrad`	`real(npel)`	Initial pellet radius [cm]
`pelvel`	`real(npel)`	Magnitude of pellet velocity [cm/s]
kpellet	`integer(npel)`	Pellet ablation model

Pellet Input Parameters for TRANSP, cont'd

- Example inputs for HFS and LFS injectors
- User specifies 2 injectors instead of 56 events

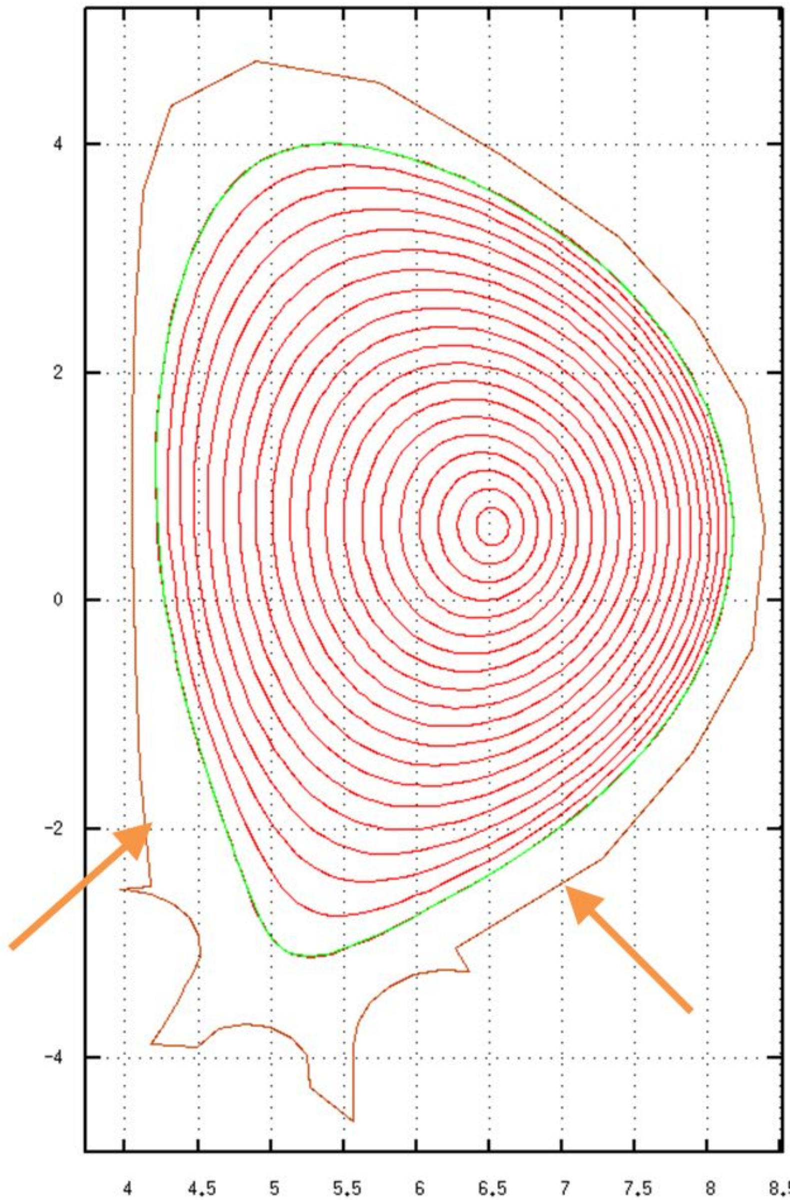
```
ipelinp    =   1,
npel       =   2,
tpel       =   2.0, 4.0,
tpelend    =   3.0, 5.0,
freqpel    =   16, 40,
aplrsta    =   415.0, 700.0,
aplysta    =   -200.0, -245.0,
aplphia    =   132.77, 67.8,
aplthea    =   0, 0,
apel       =   2.0, 2.0,
pelrad     =   0.5, 0.3,
pelvel     =   3.0e5, 3.0e5,
kpellet    =   4, 4,
```



Verification with ITER Scenarios

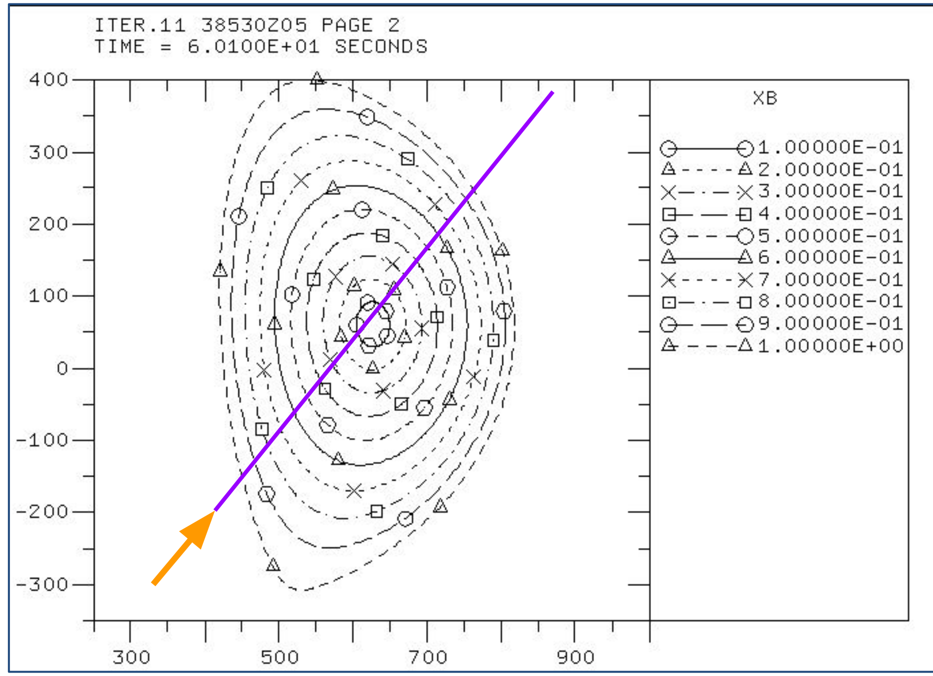
- Have been using the ITER scenario to verify:
  - Pellet code framework is working as expected
    - Pellet event scheduling
    - Pellet trajectory tracing
    - ...
  - Pellet ablation models are being called correctly and give qualitatively realistic results
  - Code restarts correctly
  - Still need to add output profiles to NETCDF file

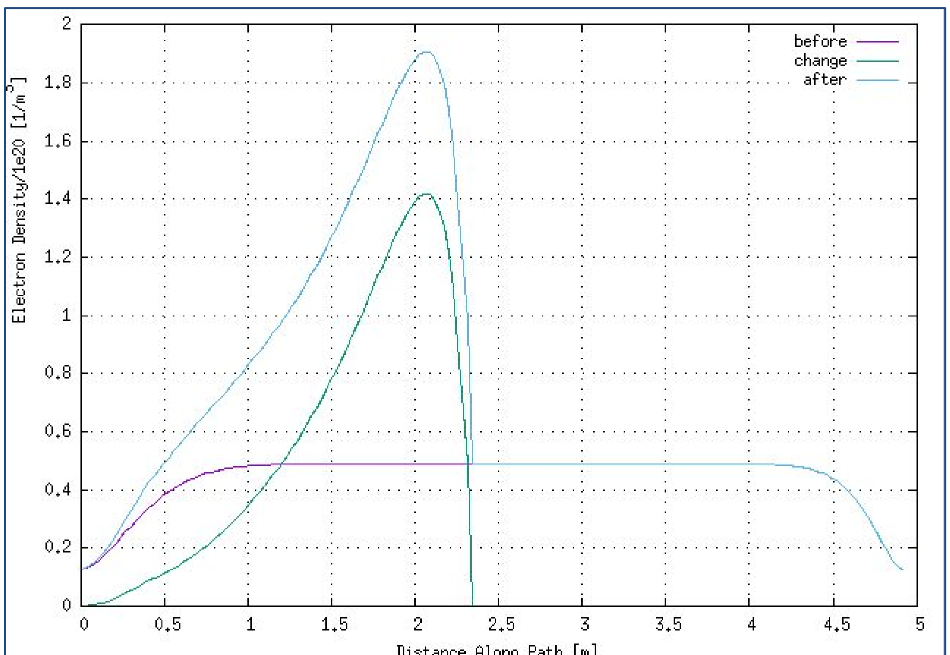
- Issues to be aware of:
  - Multiple pellet injections between UFILE data points (likely not disastrous)
  - Multiple pellet injections during a time-step (likely trouble ∴ not-allowed)

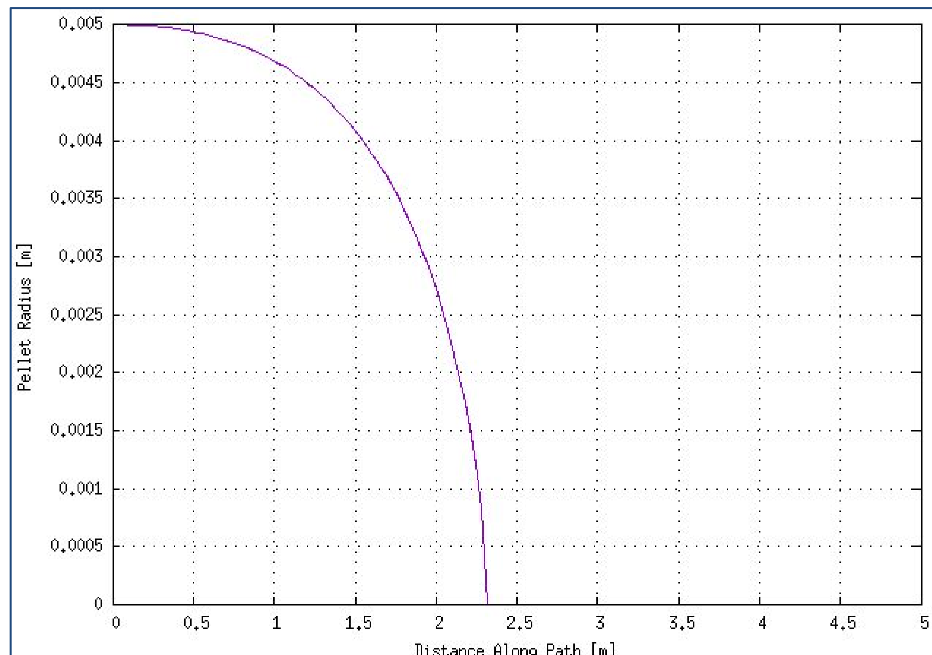


Preliminary Results

- ITER shot number 38530
- Plasma shown at 60.1 s (L-mode)
- Profiles along path for HFS injector
    - Electron density before, change and during from HFS injector
    - Pellet radius
- Qualitatively the code is working…







Next Steps

- Continue verification with ITER scenarios
  - F. Poli has provided additional cases which will isolate the effect of the pellets
- Validation with JET fueling scenario
  - Working with F. Poli to compare TRANSP results with a JET experiment and pellet ablation code
- Upgrade TRANSP module in OMFIT
  - Include pellet inputs
  - Add capability to test pellet aiming by plotting trajectory against plasma state at different times
- Implement advanced pellet ablation models
  - NGS model of Houlberg
  - New model from Parks
- Implement diagnostics that would be beneficial for comparison with experiments
- Application to DIII-D ELM mitigation with lithium pellets
  - Will be working with A. Bortolon to compare with other codes he has run
- Documentation and tutorial (for an OMFIT based set-up)