Budyko-Sellers Finite Differences Gradient

```
$ cd budyko_sellers
$ gfortran -c budyko_sellers.f
$ gfortran -c fwd_driver.f
$ gfortran -o exec_forward budyko_sellers.o fwd_driver.o
$ cd forward_experiment; ./exec_forward
```

OR

```
$ cd budyko_sellers
$ make -f MakefileForward clean;
$ make -f MakefileForward exec_forward;
$ cd forward_experiment; ./exec_forward
```

Budyko-Sellers TLM Gradient

```
$ gcc -c ../ADFirstAidKit/adStack.c
$ gcc -c ../ADFirstAidKit/adBinomial.c
$ tapenade -tangent -tgtvarname %_tl -tgtfuncname %_tl -head "BUDYKO_SELLERS(J)/(XXS)" budyko_sellers.f
$ gfortran -c budyko_sellers_tl.f
$ gfortran -c tl_driver.f
$ gfortran -o tap_exec_tl adStack.o adBinomial.o budyko_sellers_tl.o tl_driver.o
$ cd tapenade_experiment; ./tap_exec_tl
```

```
$ staf -toplevel BUDYKO_SELLERS -input XXS -output J -forward -arglist budyko_sellers.f -l taf_forward.log -f77 -pure
$ gfortran -c budyko_sellers_tl.f
$ gfortran -c tl_driver.f
$ gfortran -o taf_exec_tl budyko_sellers_tl.o tl_driver.o
$ cd tapenade_experiment; ./taf_exec_tl
```

Budyko-Sellers TLM Gradient

```
$ cd budyko_sellers
$ make -f MakefileTapenade clean;
$ make -f MakefileTapenade tap_exec_tl;
$ cd tapenade_experiment; ./tapenade_exec_tl
```

```
$ cd budyko_sellers
$ make -f MakefileTAF clean;
$ make -f MakefileTAF taf_exec_tl;
$ cd taf_experiment; ./taf_exec_tl
```

Budyko-Sellers Adjoint Gradient

```
$ gcc -c ../ADFirstAidKit/adStack.c
$ gcc -c ../ADFirstAidKit/adBinomial.c
$ tapenade -reverse -adjvarname %_ad -adjfuncname %_ad -head "BUDYKO_SELLERS(J)/(XXS)" budyko_sellers.f
$ gfortran -c budyko_sellers_ad.f
$ gfortran -c ad_driver.f
$ gfortran -o tap_exec_ad adStack.o adBinomial.o budyko_sellers_ad.o ad_driver.o
$ cd tapenade_experiment; ./tap_exec_ad
```

```
$ staf -toplevel BUDYKO_SELLERS -input XXS -output J -reverse -arglist budyko_sellers.f -l taf_forward.log -f77 -warning -info
$ cat taf_reverse.log
$ gfortran -c budyko_sellers_ad.f
$ gfortran -c ad_driver.f
$ gfortran -o taf_exec_ad budyko_sellers_ad.o ad_driver.o
$ cd tapenade_experiment; ./taf_exec_ad
```

Budyko-Sellers Adjoint Gradient

```
$ cd budyko_sellers
$ make -f MakefileTapenade clean;
$ make -f MakefileTapenade tap_exec_ad;
$ cd tapenade_experiment; ./tapenade_exec_ad
```

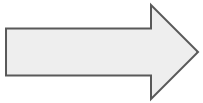
```
$ cd budyko_sellers
$ make -f MakefileTAF clean;
$ make -f MakefileTAF taf_exec_ad;
$ cd taf_experiment; ./taf_exec_ad
```

TAF vs Tapenade

- For this super simple simulation, everything fits in memory, so we should not be recomputing anything.
- Tapenade is store-all by default. The tape is initialized and stacked and popped as a default setting.
- TAF is recompute-all by default. Therefore, we need to tell it to store everything in memory instead of recomputing everything. You will see the following comments sprinkled in the code

Start of code:
CADJ INIT tapex = 'TAF_tape'

Inside time loop:
CADJ STORE t = tapex



These are comments in F-77 but they are also directives for TAF.