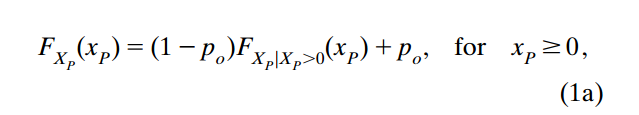
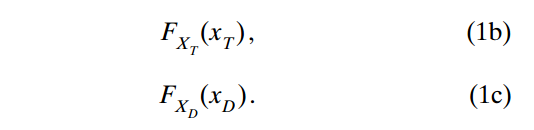
Newman et al. 2015

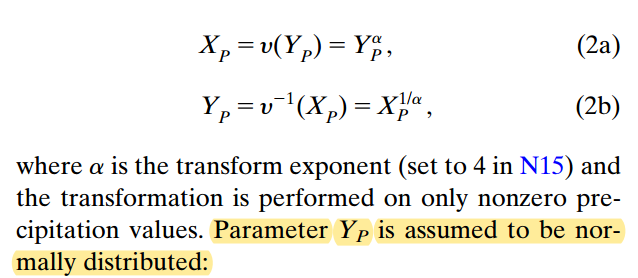
# Theory

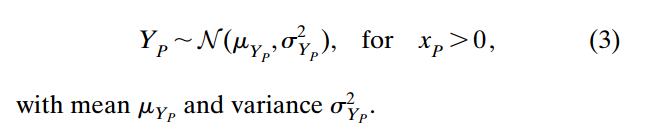
Assume the CDF of precipitation and temperature as: T (mean temperature) and D (daily temperature range) are assumed to be normally distributed.



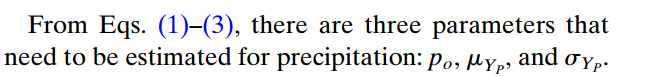


Transformation of precipitation. It is unnecessary to strengthen that only for nonzero precipitation. And it is not normally distributed. In Newman 2019, the power-law transformation is changed to box-cox transformation.



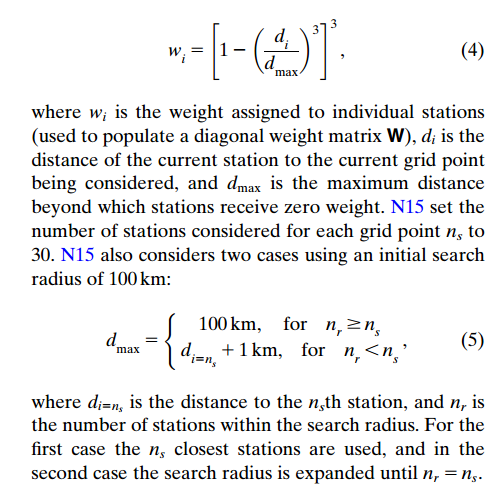


# Parameter estimation

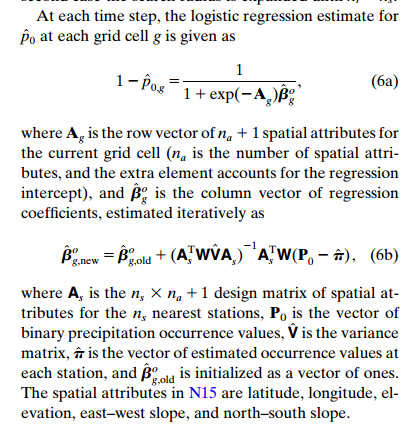


## p0,g is estimated using locally weighted logistic regression. The subscript g represents the target grid cell.

For a target grid cell, the weight of station i is in (4). ns is the number of total stations used for each grid cell (set to 30). For one grid cell, sort all stations from close to far distance. Find the nsth station (30): if the distance between this station and target grid cell is < 100 km, dmax is set to 100 km; if the distance > 100 km, dmax = distance + 1 km. The 100 km is used in Newman 2015, but in Newman 2019, it is 10 km and the station number is reduced from 30 to 25.

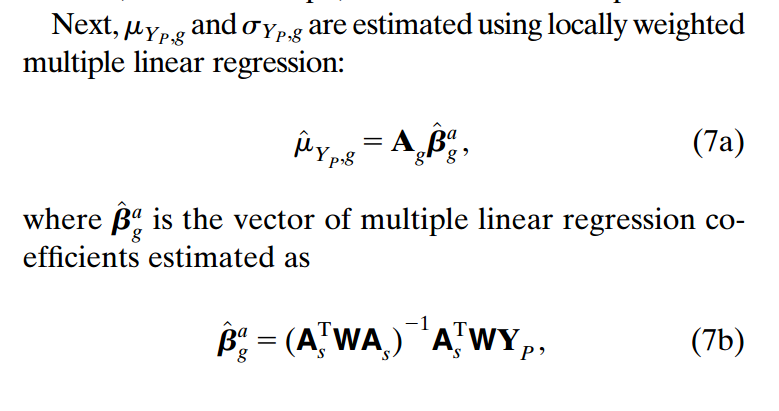


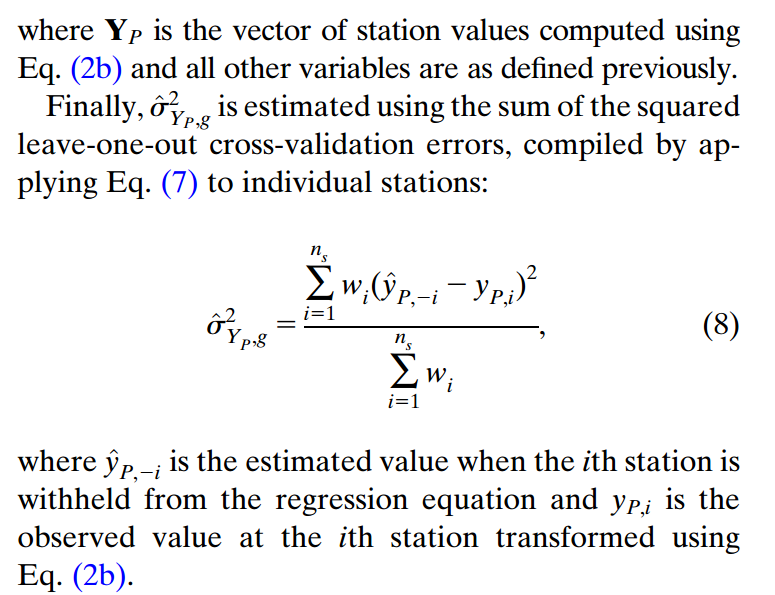
The, the estimate of p0 at the target grid cell is derived using logistics regression based on the ns (i.e., 30) stations. The below figure just shows how to do logistic regression.



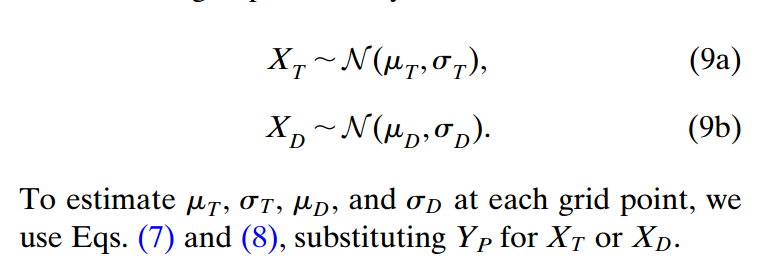
## and is estimated using locally weighted multiple linear regression.

g is the target grid cell. Yp is the transformed precipitation. To estimate , leave-one-out cross-validation is used. Each time, one station is excluded; then precipitation is estimated using other stations at this location; then the estimate errors is calculated; then repeat these steps for all stations and get errors for each station; then the weighted sum of errors is used as .





## Temperature and the DTR are assumed to be Gaussian without having to perform any transformation. Their mean and variance are estimated using locally weighted multiple linear regression.

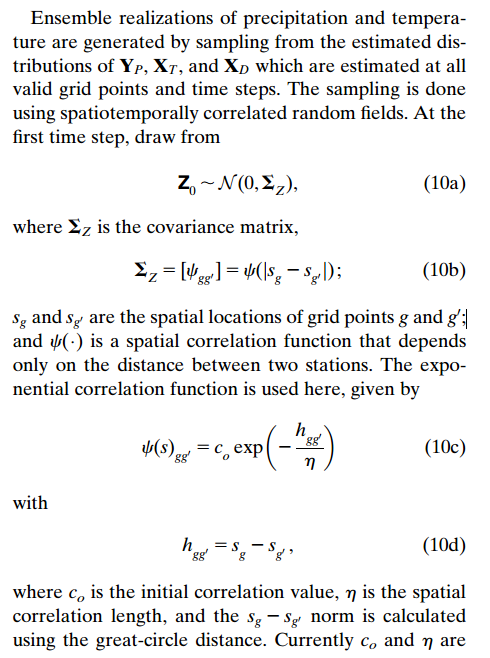


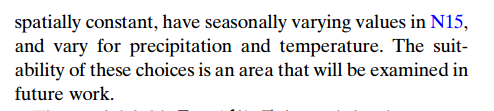
**In this step, the distribution of precipitation and temperature (CDF) for each grid cell is determined based on the parameters (3 parameters for precipitation, and 2 parameters for temperature). There is no actual interpolated precipitation or temperature in this step.**

# Ensemble estimation

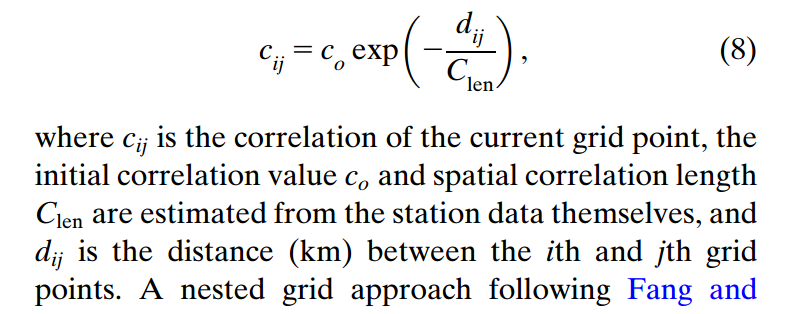
## Spatial correlation random distribution

For any pair of grids in the study region (g and g’), their correlation is determined by distance according Eq 10c. For all grid pairs in study area, the spatial correlation is represented using which is a covariance matrix. The parameters in Eq 10c is determined using station pair data, which is clearer in Newman 2015. Thus, is deterministic. To generate spatiotemporally correlated random fields, the fields are sampled from a normal distribution with the mean value of 0 and standard deviation of . Z0 is the spatial correlation random fields at the first time step.





Eq 10c is the same with Eq 8 in Neman 2015. Clen and C0 are acquired using station data over the whole study area for each season.



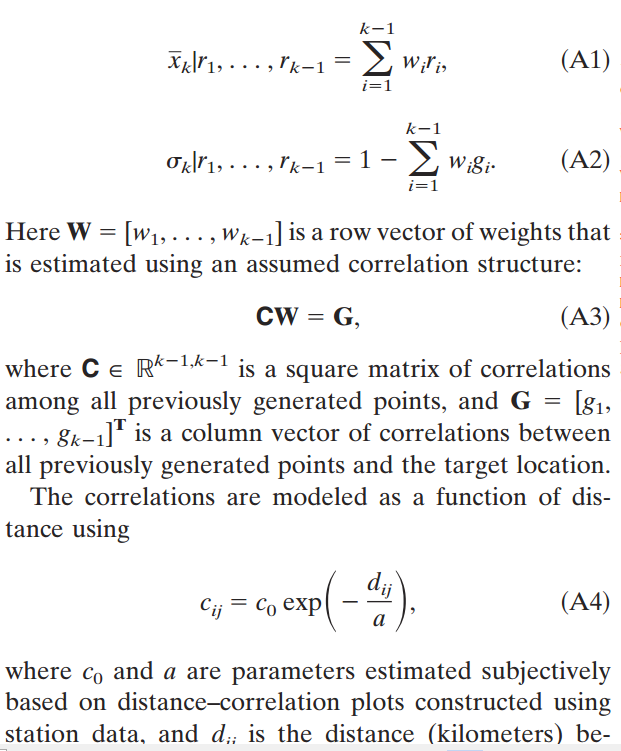
However, in codes (spcorr\_grd.f90), the function is corr (iprev, jprev) = exp (-(dist/clen)) ! the correlation model introduced in Martyn's paper, which is different with that in the paper.

### Martyn 2006: generate mean and std for each point

Loop structure: loop-1: nest (2^9, 2^8, 2^7, …, 2^0), loop-2: 1st dimension of the gridded domain, loop-3: 2nd dimension

*A problem: For the first several nests, the spatial distance is very large 2^9, 2^8, 2^7, …, 2^0. Thus, the number of previously generated points is low, indicating that the generated weight and variance for the target grid cell may not be reliable.*

The conditional mean (xk) and standard deviation (σk) for each point (k) is:



*call ludcmp (corr, indx, tmp)*

*call lubksb (corr, indx, twgt)*

The two lines in spcorr\_grd.f90 is used to solve the equation of CW=G. C is corr, square matrix of correlations among all previously generated points (number=k-1). G is a column vector of correlations between all previously generated points and the target location. W is weight.

The search of k-1previously generated stations is within a space window, maxp = (nloc\*2+1) \*\* 2, nloc=3. maxp is the max number of previously generated stations. nloc=3 is grid cells.

### Martyn 2006: Generate random numbers

After getting mean and variance for each grid cell, search from the first grid. Generate a random number (~ Z(0,1)) for the first grid). For the following grids, first, find previously generated grids which is done in 3.1.1, and extract their random numbers; generate the current random number (~ Z(0,1)), which is used to multiply with standard deviation generated in 3.1.1; weighted averaging the random numbers of previously generated points, in this way, the current point is correlated to previous points, the weight is in 3.1.1; the final random number for the grid generate.

*Code is field\_rand.f90 is:*

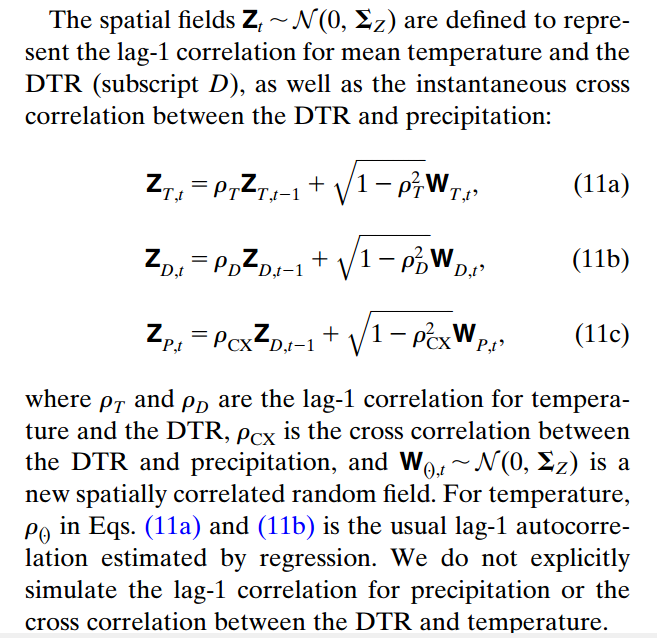
call gasdev (aran) ! *Returns a normally distributed deviate with zero mean and unit variance, using RAN1(IDUM) as the source of uniform deviates. (http://people.sca.uqam.ca/~gauthier/var3Dbrowser/html\_code/var3d/gasdev.ftn.html)*

xbar = dot\_product (vprv(1:nprv), spcorr(ilon, ilat)%wght(1:nprv)) ! vprv is cran of previously points, and spcorr(ilon, ilat)%wght is their weights

cran (ilon, ilat) = xbar + spcorr(ilon, ilat)%sdev \* aran ! cran is the random number for this grid cell

## Sample from random field distribution to generate ensembles

For time step t, the spatiotemporally correlated random field Zt is obtained from the number from the previous step and the newly sampled fields from the normal distribution N(0, ). and are the lag-1 auto-correlation of temperature and DTR. is actually very weak. WT,t, WD,t, and WP,t are sampled from N(0, ), just like Z0. So, it is better to change the name of Z and W to be clear.



## Generate probabilistic estimates

Fz is the CDF of precipitation, temperature and DTR, which is obtained in the second step. ug is the cumulative probability of Z at gird point g. ug is uP, uT, and uD in Eq. 13, xu,g is also xu,P, xu,T, and xu,D, which should be clearly stated in the paper. So essentially, Eq. 12 is the same with Eq. 13.

**A problem is that, where do we get up? I find this in generate\_ensemble.f90. How do random numbers relate to the cumulative probability?**

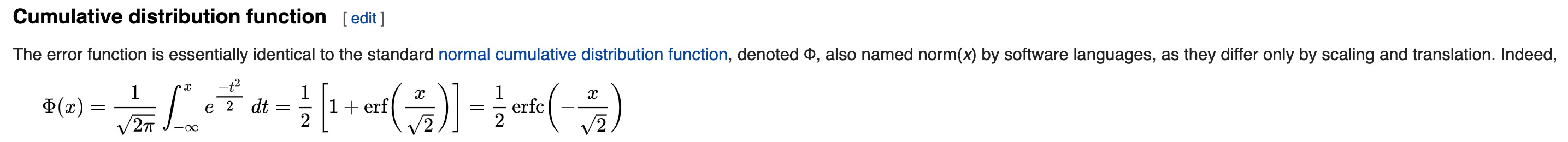
**pcp\_random is the random field, isp1 and isp2 correspond to grid location.**

! find cumulative probability

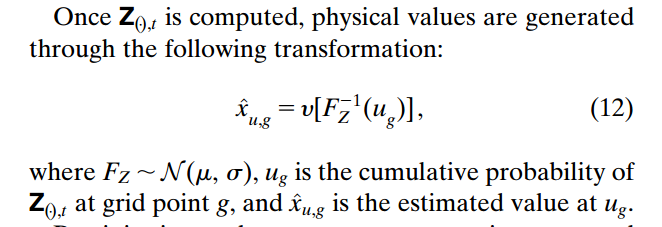
acorr = real (pcp\_random(isp1, isp2), kind(sp)) / sqrt (2.\_sp)

aprob = erfcc (acorr) ! *Computation of the complementary error function erfc(x) = 1-erf(x) with a fractional error everywhere less than 1.2 x 10^(-7) (formula by Press et al., 'Numerical Recipes in Fortran 77'). erf(x) computes the error function of x, defined as: http://fortranwiki.org/fortran/show/erf. −1≤erf(𝑥)≤1. 函数erf(x)在数学中为误差函数（也称之为高斯误差函数，error function or Gauss error function），是一个非基本函数（即不是初等函数），其在概率论、统计学以及偏微分方程和半导体物理中都有广泛的应用（图片：*[*https://zhidao.baidu.com/question/159734542.html*](https://zhidao.baidu.com/question/159734542.html)*）。https://en.wikipedia.org/wiki/Error\_function*

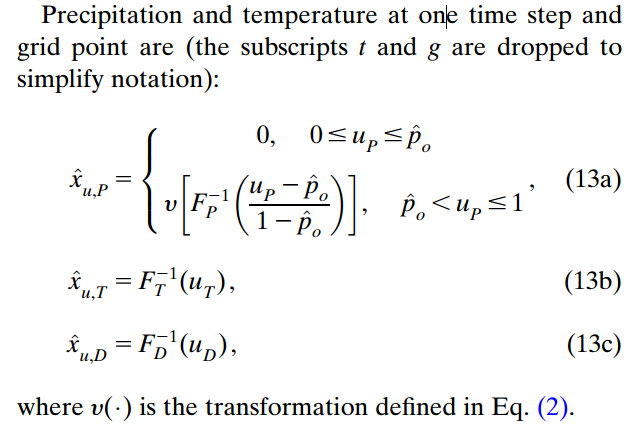
cprob = (2.d0-real(aprob, kind(dp))) / 2.d0 ! aprob is between 0 and 2. This will make sure cprob is between 0 and 1.

**

According to the above equation, cprob is actually the cumulative probability ( corresponding to x (the random number). So, the random number is actually used to represent the cumulative probability that a certain value of precipitation or temperature occurs.

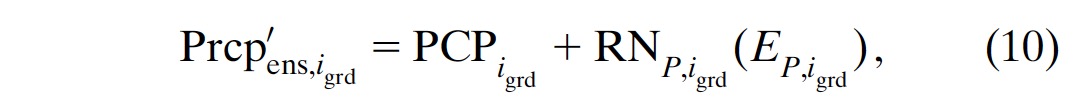


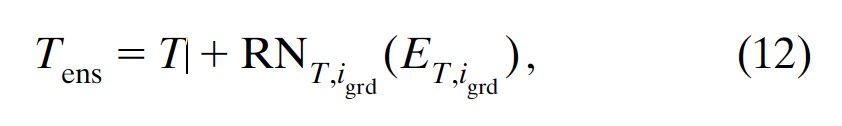
Fp is the CDF of non-zero precipitation, which is also not clearly stated in the paper. Why don’t we use one uniform CDF for precipitation, which can be divided into zero part and non-zero part, which should be clearer than the current version in the paper? So P, T, and D are Xp, Xt and Xd is the first step. This is inconsistent.



**Eq 13 in Newman 2019 is not the same with Eq 10 and 12 in Newman 2015 due to the framework change (Simon?). This is particularly true for T and D, because in Newman 2015, ensemble estimates of T and D are just from MeanValue + RandomNumber \* ErrorEstimation. MeanValue and ErrorEstimation are both in the second step.**

The corresponding codes in generate\_ensemble.f90 are as below. These are Newman 2015.





! tmean

ra = real (tmean(isp1, isp2, istep), kind(dp)) + real (tmean\_random(isp1, isp2), &

& kind(dp)) \* real (tmean\_error(isp1, isp2, istep)/3.0, kind(dp))

tmean\_out (isp1, isp2, istep) = real (ra, kind(sp))

! trange

ra = real (trange(isp1, isp2, istep), kind(dp)) + real (trange\_random(isp1, isp2), &

& kind(dp)) \* real (trange\_error(isp1, isp2, istep)/3.0, kind(dp))

trange\_out (isp1, isp2, istep) = real (ra, kind(sp))

## Code structure

For the SRCF codes, the loop structure is

1. generate random numbers

2. loop ensemble members

3. loop time steps

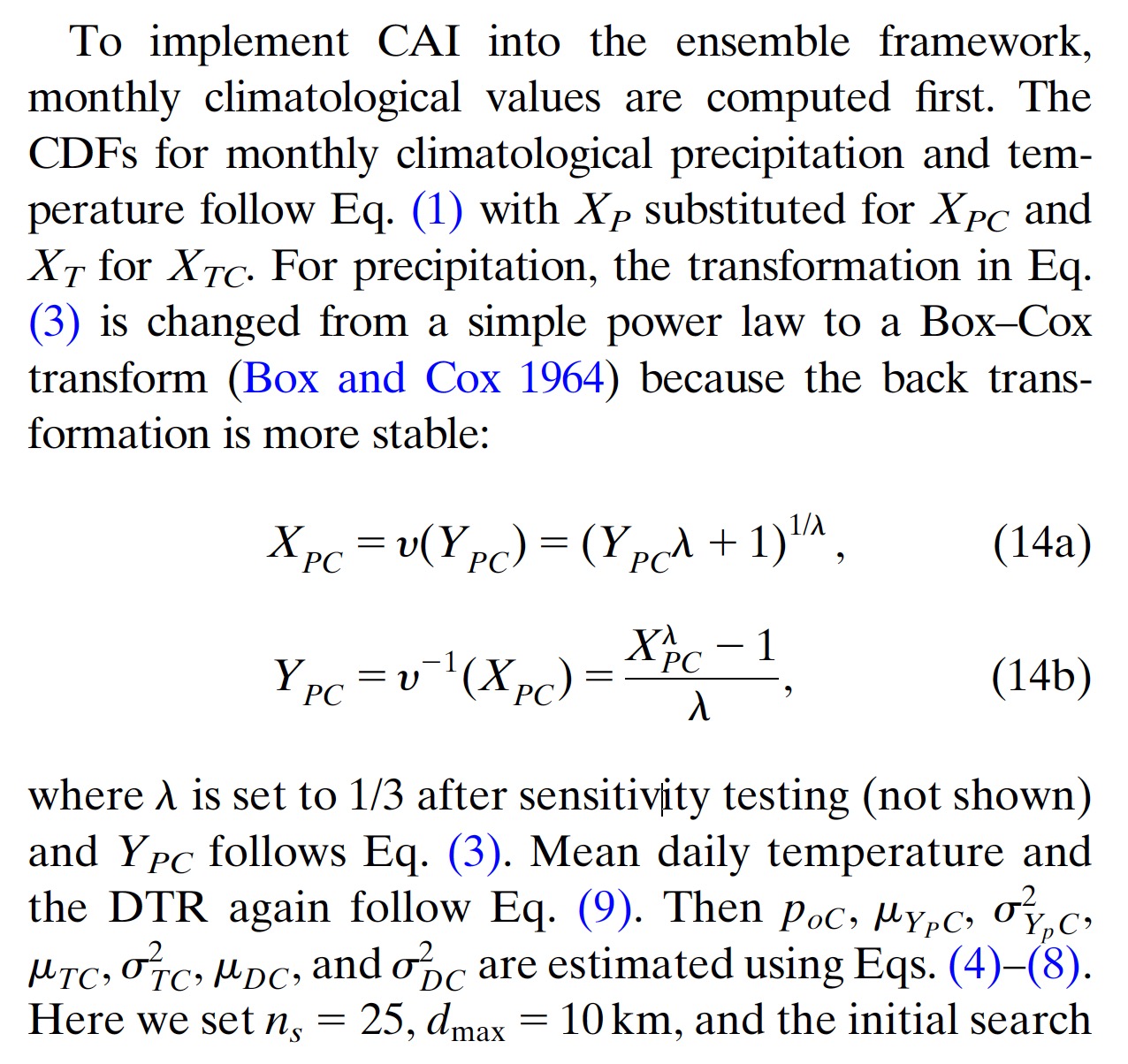
4. loop grid cells

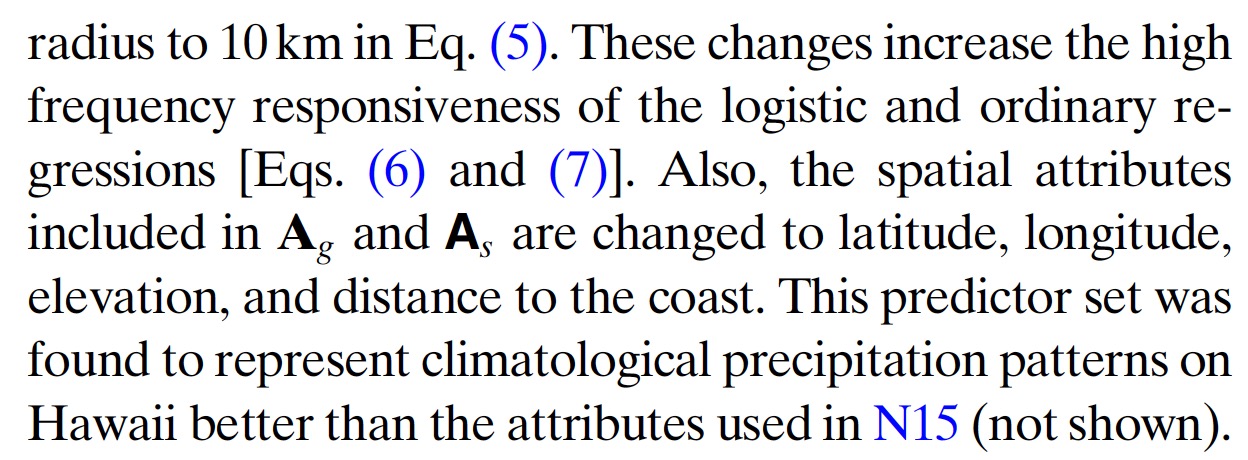
After loop-4 is finished, random numbers is updated for the next loop-3. After loop-3 is finished, the updated random numbers are propagated to the next loop-2, a new ensemble member. That means if we change ensemble numbers or time step numbers, the random fields will also change.

## Ensemble climatologically aided interpolation

### Data preparation

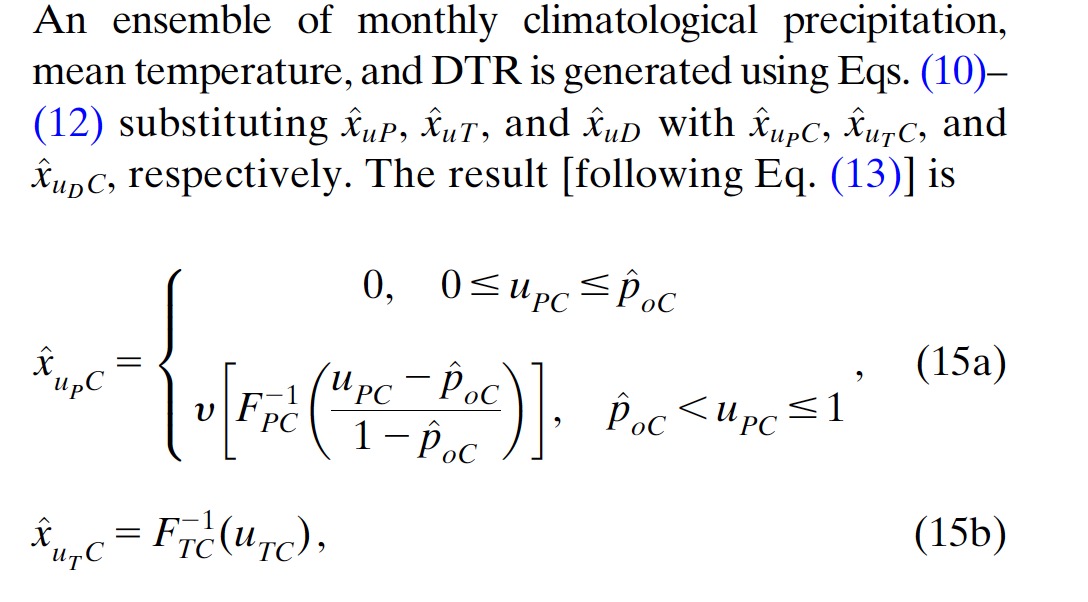
Replace the daily symbols with monthly symbols. Box-cox transformation is used for precipitation.

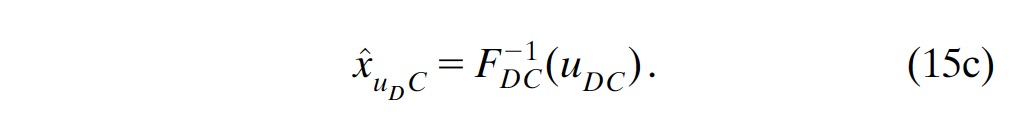




### Generate monthly climatological ensemble (12 months)

Then, monthly ensembles are generated in the same way with daily ensembles. Eq 15 and Eq 13 in Newman 2019 are the same.



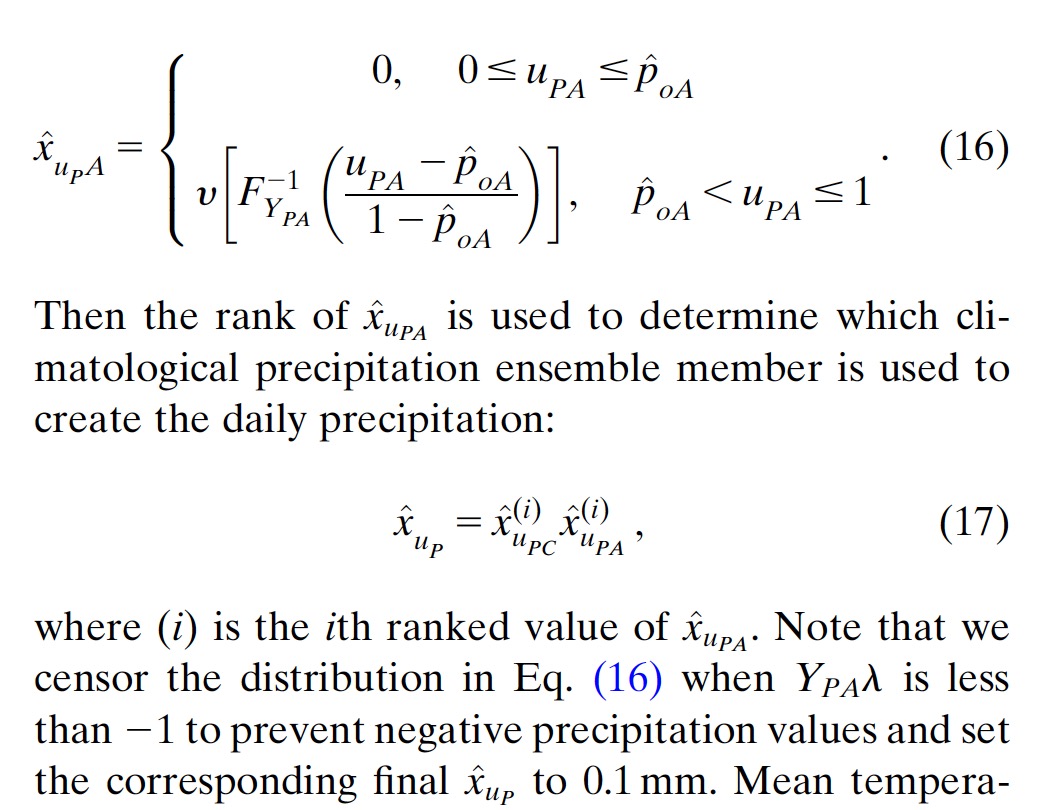


### Generate daily ratio and anomaly ensemble

The daily ratio and daily anomalies for precipitation, temperature, and DTR are computed using the estimated climatological ensemble means **(mean of all ensembles)** for the closest grid point to each station, to force consistency between the daily anomalies and the estimated climatological ensemble.

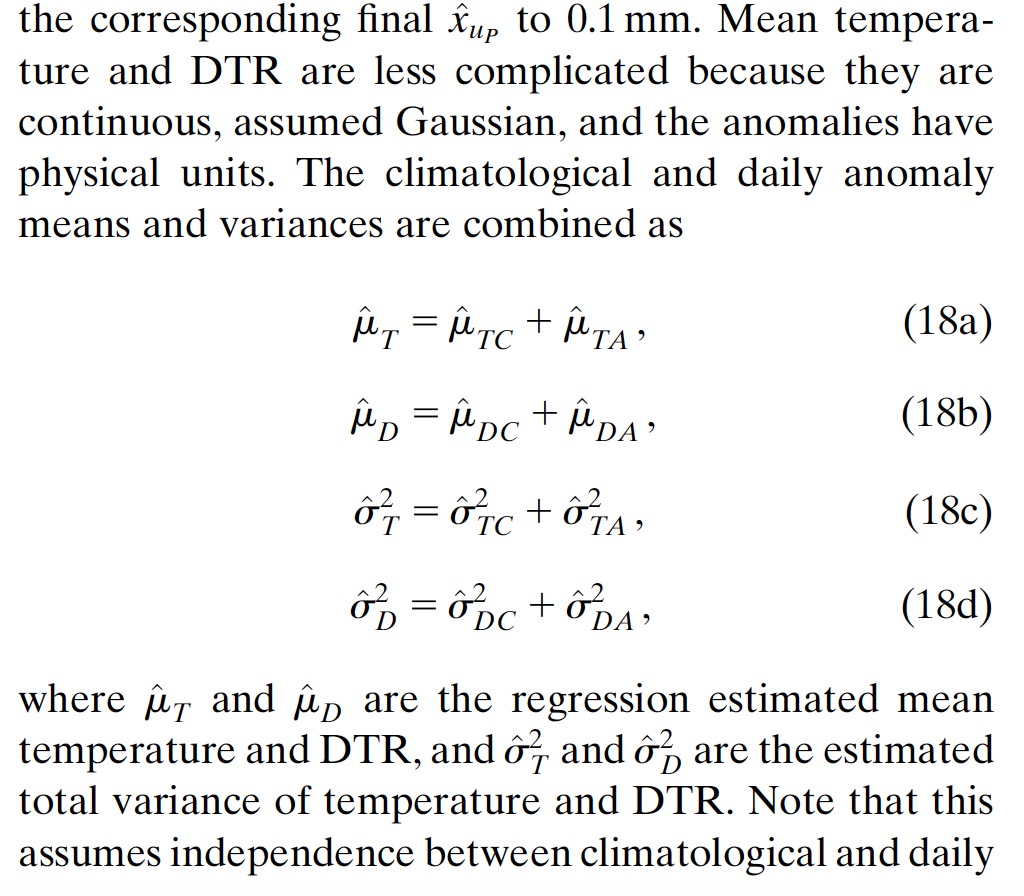
Then, daily anomalies are interpolated using regression, same with daily precipitation/temperature.

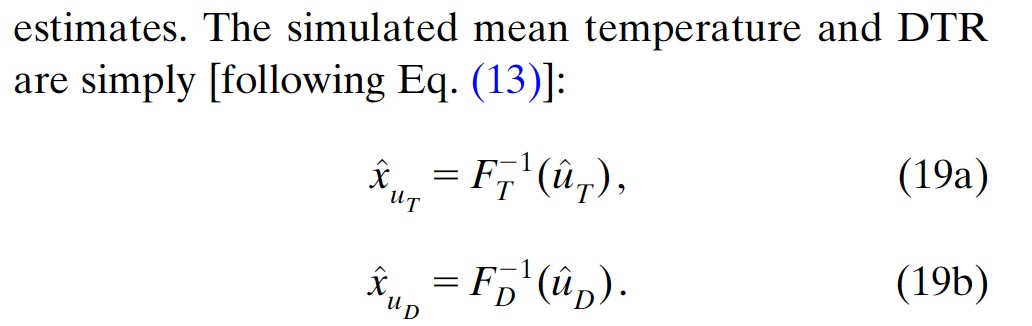
Then, daily anomaly ensembles are generated. For precipitation, daily anomaly ensemble is generated using Eq 16. For one ensemble, find its rank among all ensemble members (e.g., 100), and then find the same ranked ensemble of monthly climatology. Then, multiply the daily precipitation anomaly ratio with monthly climatology precipitation value, to get the daily ensemble precipitation.



For temperature, the process is different with precipitation. In the monthly climatological regression and daily anomaly regression part, the mean value (μ) and standard deviation (σ2) for each grid cell are obtained. Adding the mean value and standard deviation from climatological and anomaly regression obtains regression estimated mean temperature and DTR. Then, the estimated mean value and variance define a CDF, and using the cumulative probability derived from daily anomaly random numbers, the final temperature is estimated.

**A problem is what is the Ut or Ud in Eq 19 represent?**





# Newman 2020 (submitted)

Why using power-law transformation in N15 instead of box-cox transformation in N19?

# eCAI GMET program work flow

## main.f90 in downscale

* 选择mode-2
* 读取站点列表txt文件：call read\_station\_list
* 读取read grid domain file：read grid domain file
* 初始化站点和网格的基本信息数据! x arrays for station variables，! z arrays for grid variables，结构是 [1, lat, lon, altitude, slp\_n, slp\_e]，其中1是用作回归的
* 开始regression。time\_mode三种模式：（1）daily: estimate\_forcing\_regression，（2）daily\_anom： estimate\_climo\_anom\_regression，（3）climo：estimate\_climo\_regression。这三种模式的基本输入输出变量类型一致，但是（1）和（3）使用了[1, lat, lon, altitude, slp\_n, slp\_e]，（2）使用了[1, lat, lon, altitude]
* 输入数据：save\_forcing\_regression

## Time\_mode is daily:

estimate\_forcing\_regression

* 读取站点数据，并计算降水-温度range的相关性（一般负数，小）、温度的autocrrelation，uses an n-day moving average (window) to remove "monthly" cycle from temp and computes autocorrelation on the anomalies
* 针对每一个grid cell，找到距离最近的30个站点，并计算每个站点的基础权重（w\_base）：compute\_station\_weights。权重计算公式如下，搜索站点半径为1000 km，超过这个范围权重设为0，范围以内：weight = (1.0d0-(dist/maxd)\*\*3) \*\* 3，其中maxd = 1000 km，**但是这意味着当距离比较小（如100km），不同站点之间的权重几乎没有差别**。
* 开始循环
* Level 1:

对于每一个时刻t（ntimes）

* + 提取时刻t每一个站点的prcp tmean trange
  + 对于prcp，做power transformation：normalize\_y (4.0d0, y)。 y = y1/4.
* Level 2:

对于每一个网格g（ngrid）

* + 找到sta\_limit（30个）临近站点距离网格的最大距离（