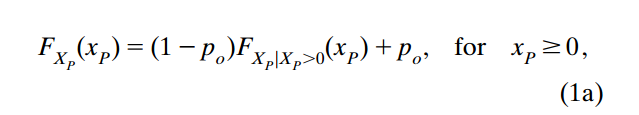
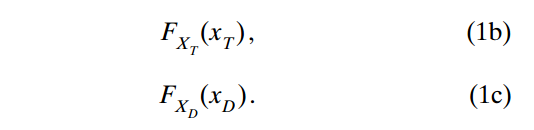
Newman et al. 2015

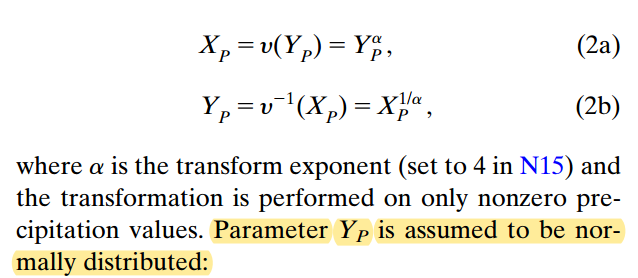
# Theory

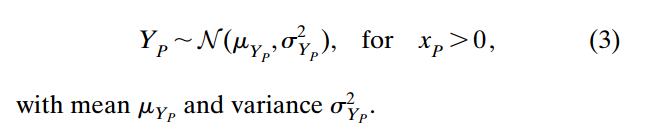
Assume the CDF of precipitation and temperature as: T (mean temperature) and D (daily temperature range) are assumed to be normally distributed.



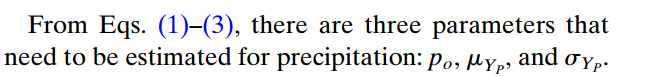


Transformation of precipitation. It is unnecessary to strengthen that only for nonzero precipitation. And it is not normally distributed. In Newman 2019, the power-law transformation is changed to box-cox transformation.



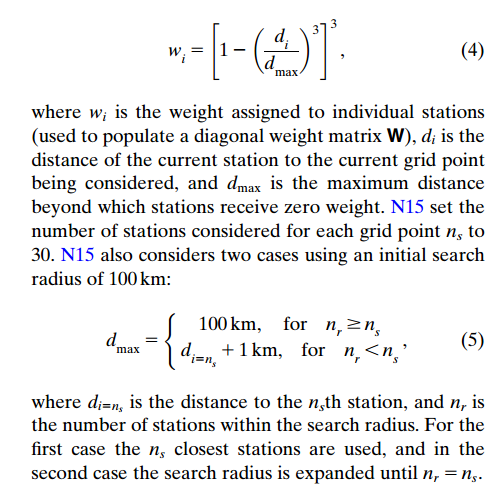


# Parameter estimation

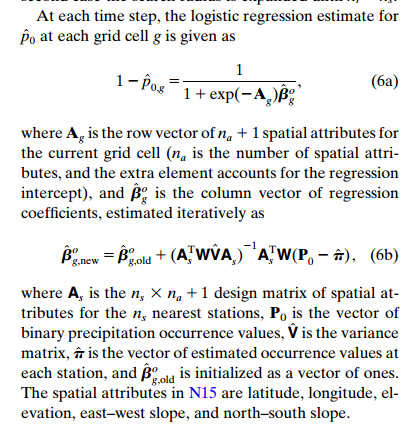


## p0,g is estimated using locally weighted logistic regression. The subscript g represents the target grid cell.

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.

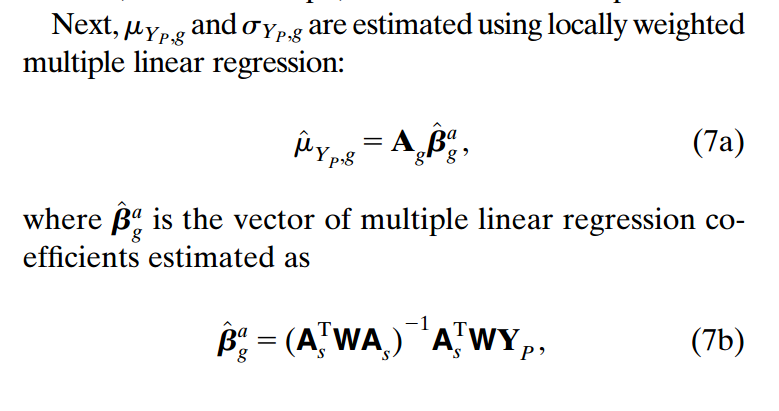


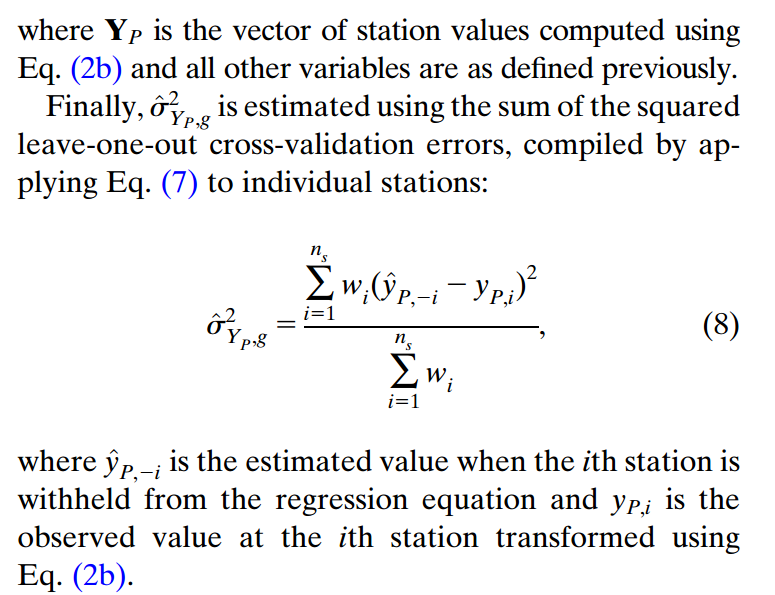
The, the estimate of p0 at the target grid cell is derived using logistics regression based on the ns (i.e., 30) stations. The below figure just shows how to do logistic regression.



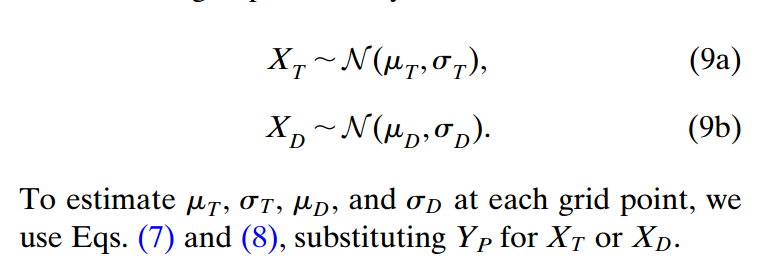
## and is estimated using locally weighted multiple linear regression.

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.

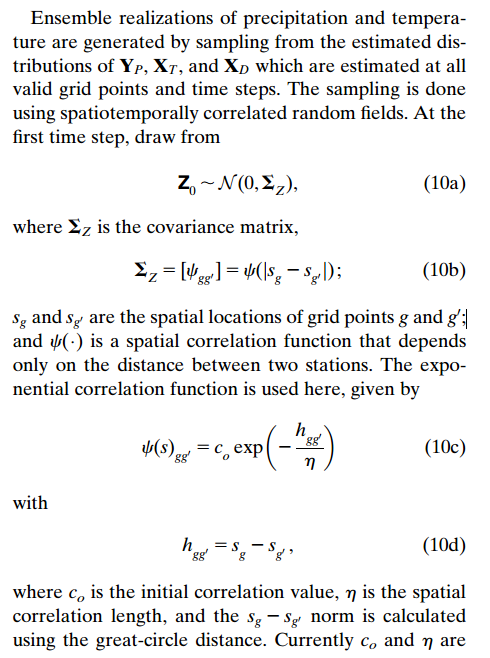


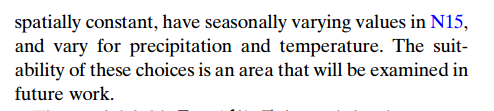
**In this step, the distribution of precipitation and temperature (CDF) for each grid cell is determined based on the parameters (3 parameters for precipitation, and 2 parameters for temperature). There is no actual interpolated precipitation or temperature in this step.**

# Ensemble estimation

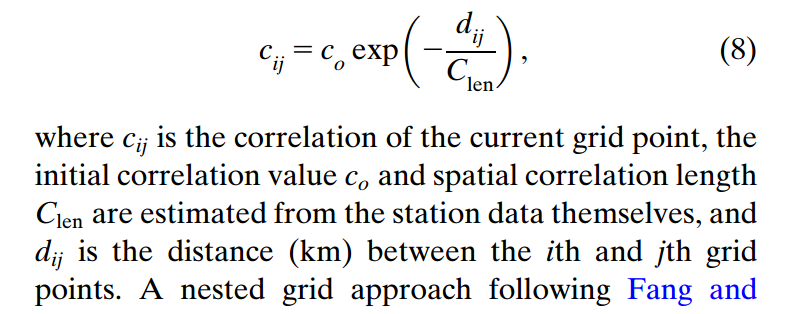
## Spatial correlation random distribution

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.



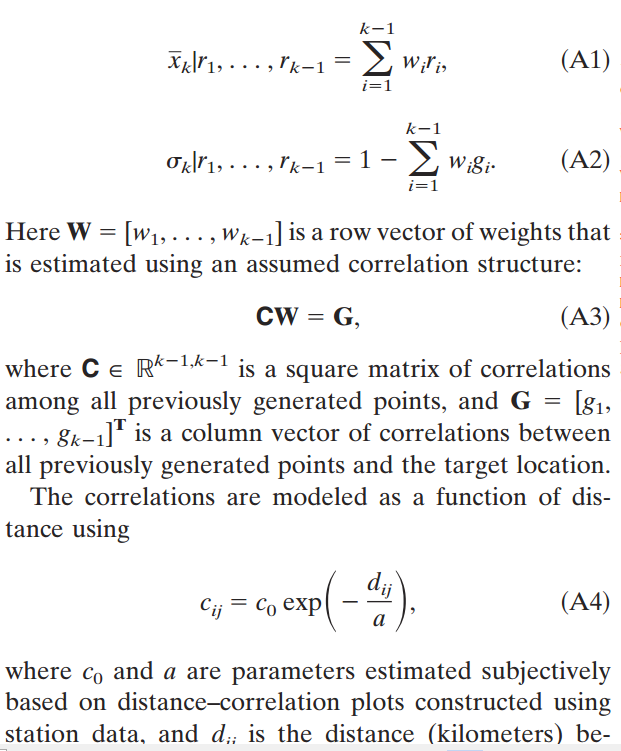
However, in codes (spcorr\_grd.f90), the function is corr (iprev, jprev) = exp (-(dist/clen)) ! the correlation model introduced in Martyn's paper, which is different with that in the paper.

### 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:



*ri* in the figure is random number.

*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:*

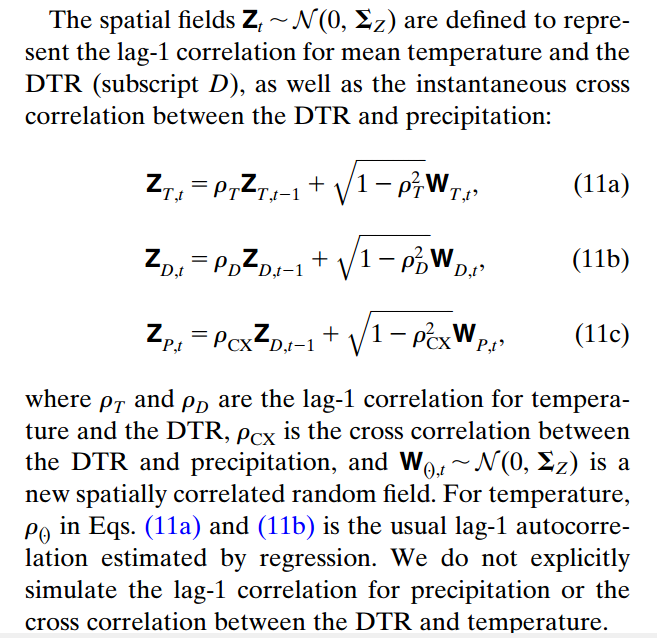
call gasdev (aran) ! *Returns a normally distributed deviate with zero mean and unit variance, using RAN1(IDUM) as the source of uniform deviates. (http://people.sca.uqam.ca/~gauthier/var3Dbrowser/html\_code/var3d/gasdev.ftn.html)*

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?**

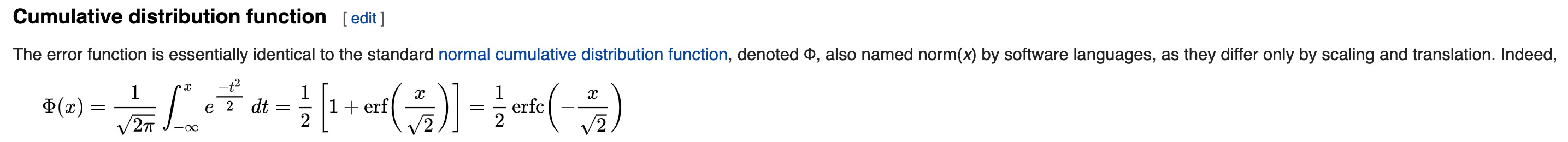
**pcp\_random is the random field, isp1 and isp2 correspond to grid location.**

! find cumulative probability

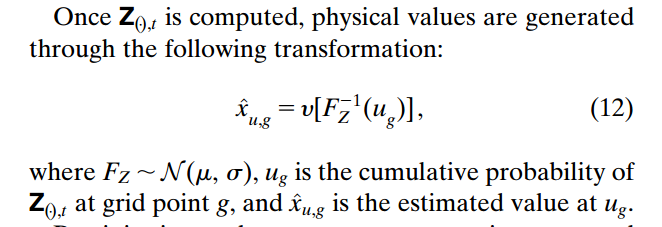
acorr = real (pcp\_random(isp1, isp2), kind(sp)) / sqrt (2.\_sp)

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://en.wikipedia.org/wiki/Error\_function*

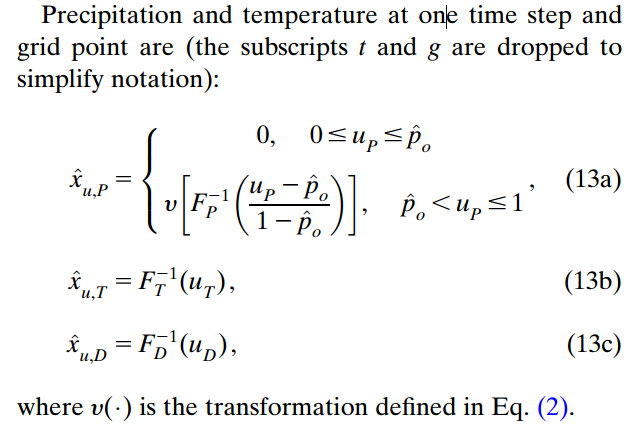
cprob = (2.d0-real(aprob, kind(dp))) / 2.d0 ! aprob is between 0 and 2. This will make sure cprob is between 0 and 1.

**

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.

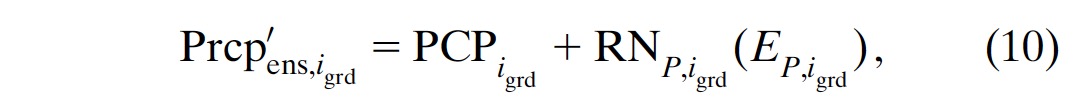


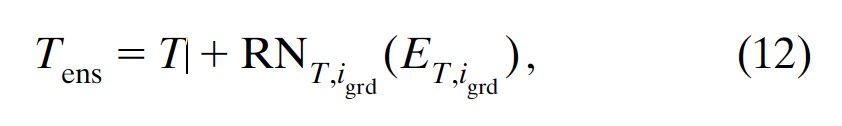
Fp is the CDF of non-zero precipitation, which is also not clearly stated in the paper. Why don’t we use one uniform CDF for precipitation, which can be divided into zero part and non-zero part, which should be clearer than the current version in the paper? So P, T, and D are Xp, Xt and Xd is the first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

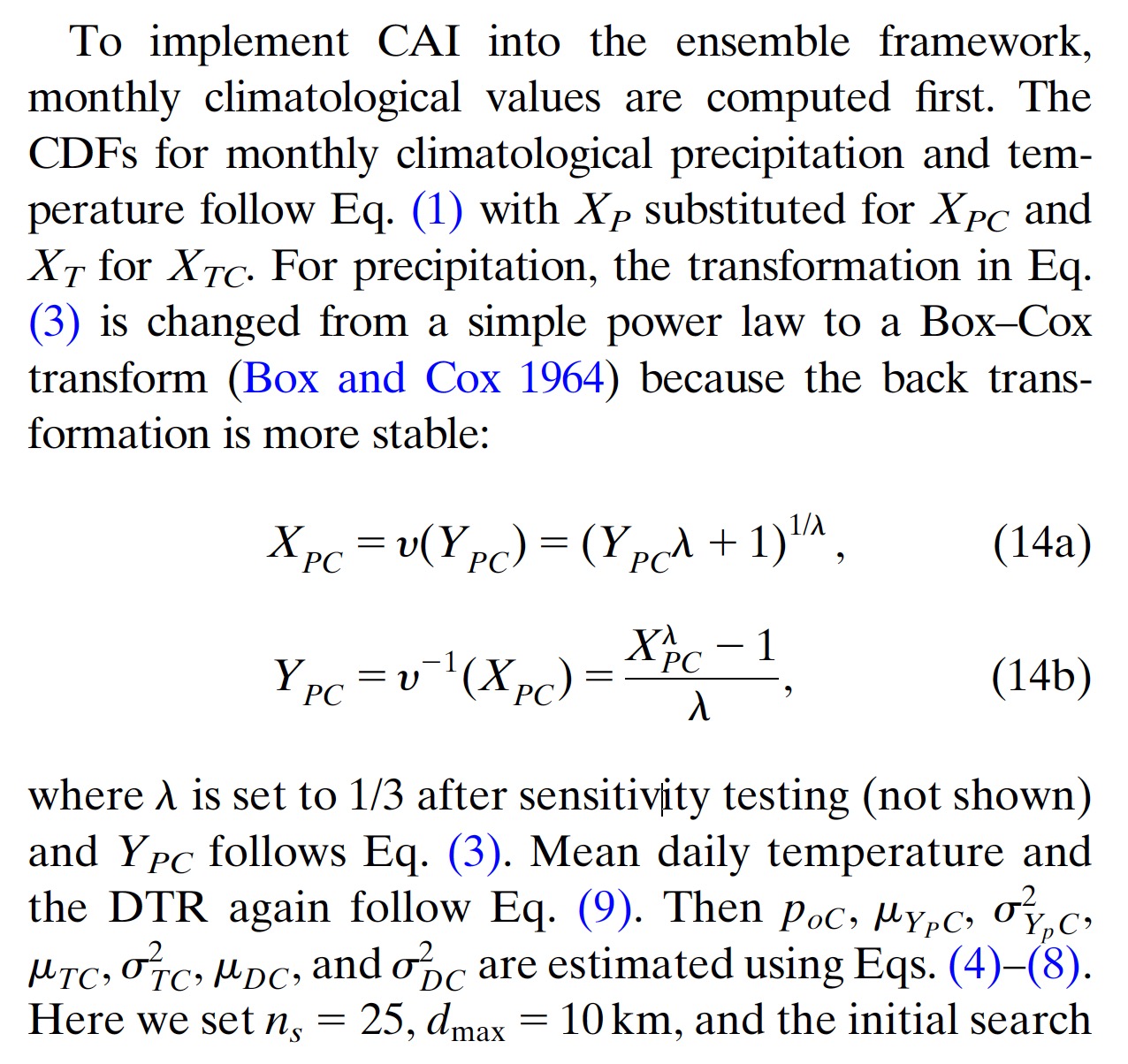
4. loop grid cells

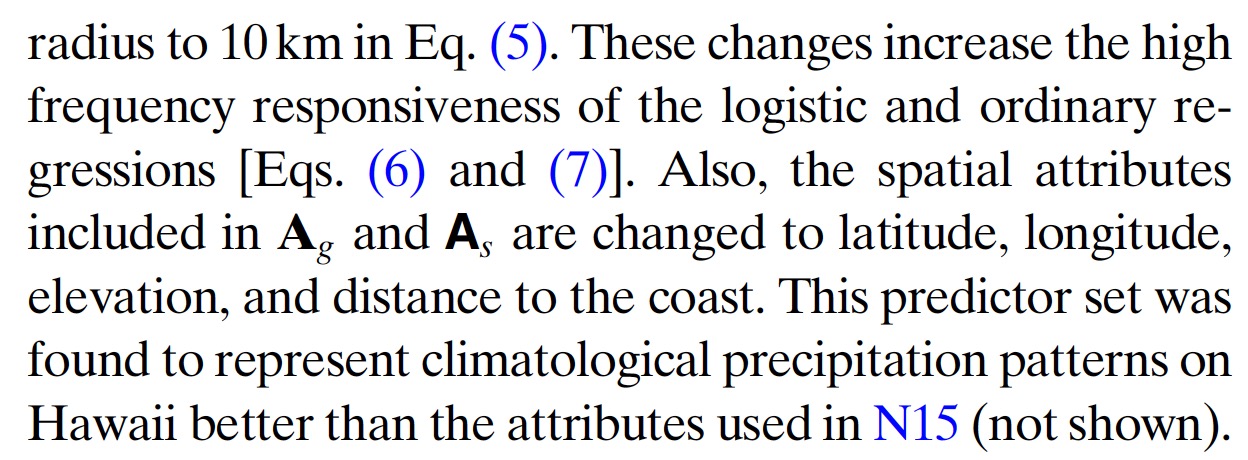
After loop-4 is finished, random numbers is updated for the next loop-3. After loop-3 is finished, the updated random numbers are propagated to the next loop-2, a new ensemble member. That means if we change ensemble numbers or time step numbers, the random fields will also change.

## Ensemble climatologically aided interpolation

### Data preparation

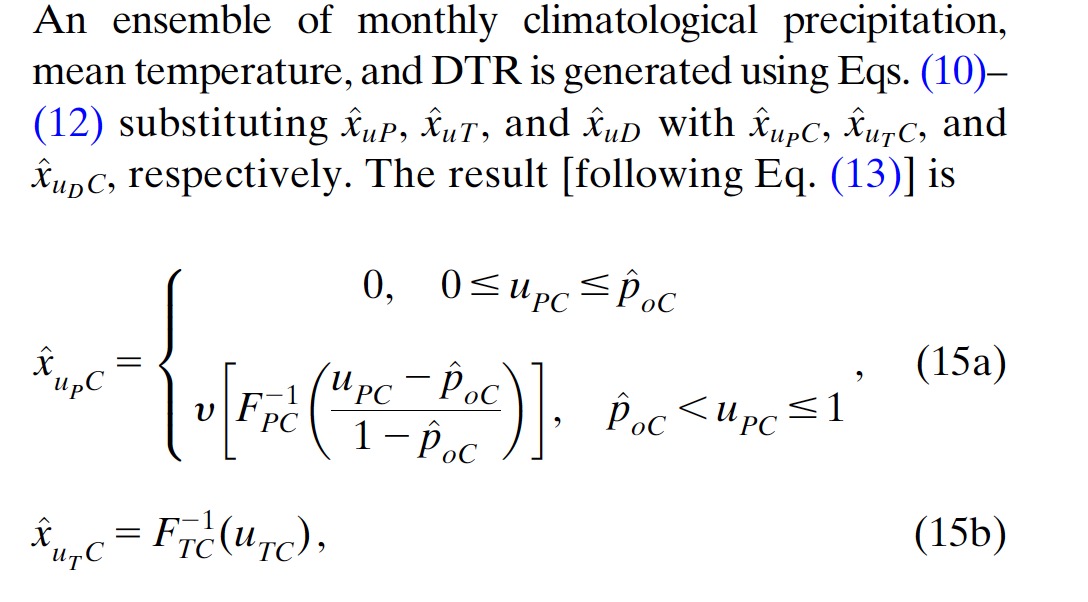
Replace the daily symbols with monthly symbols. Box-cox transformation is used for precipitation.

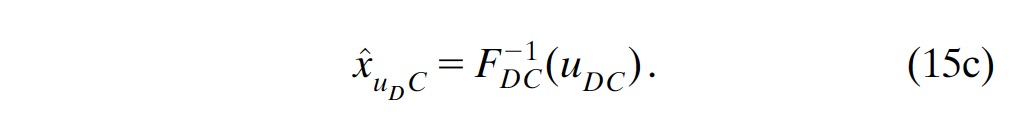




### Generate monthly climatological ensemble (12 months)

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



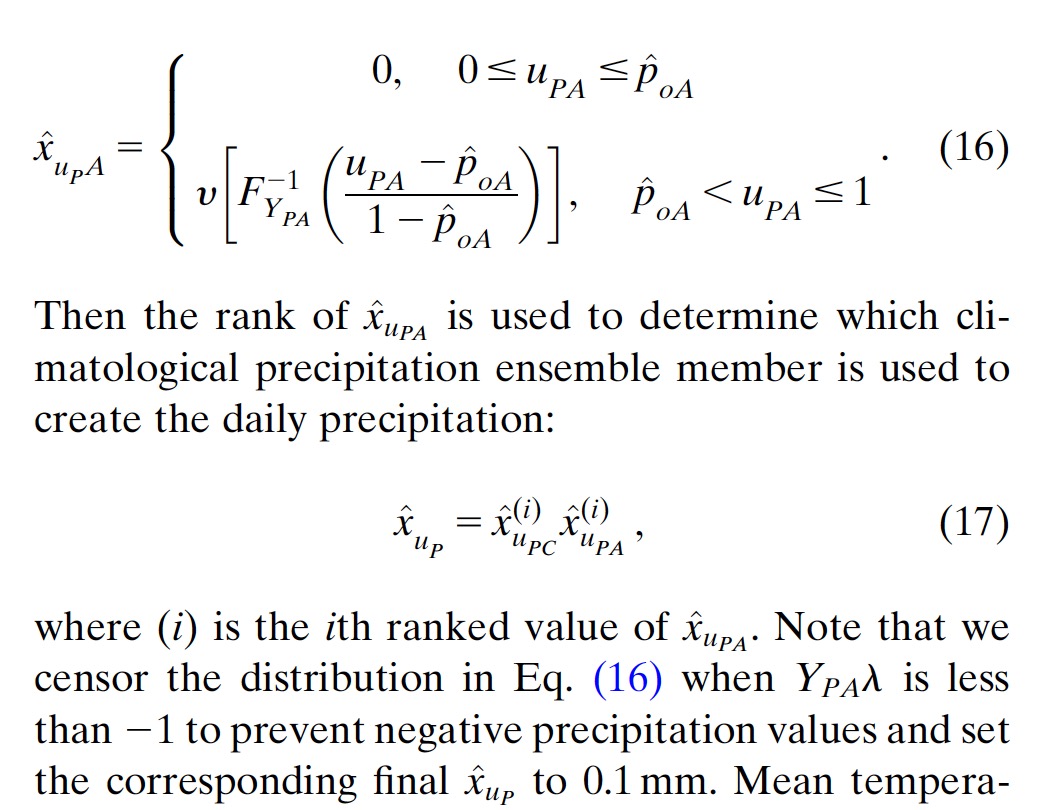


### Generate daily ratio and anomaly ensemble

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.

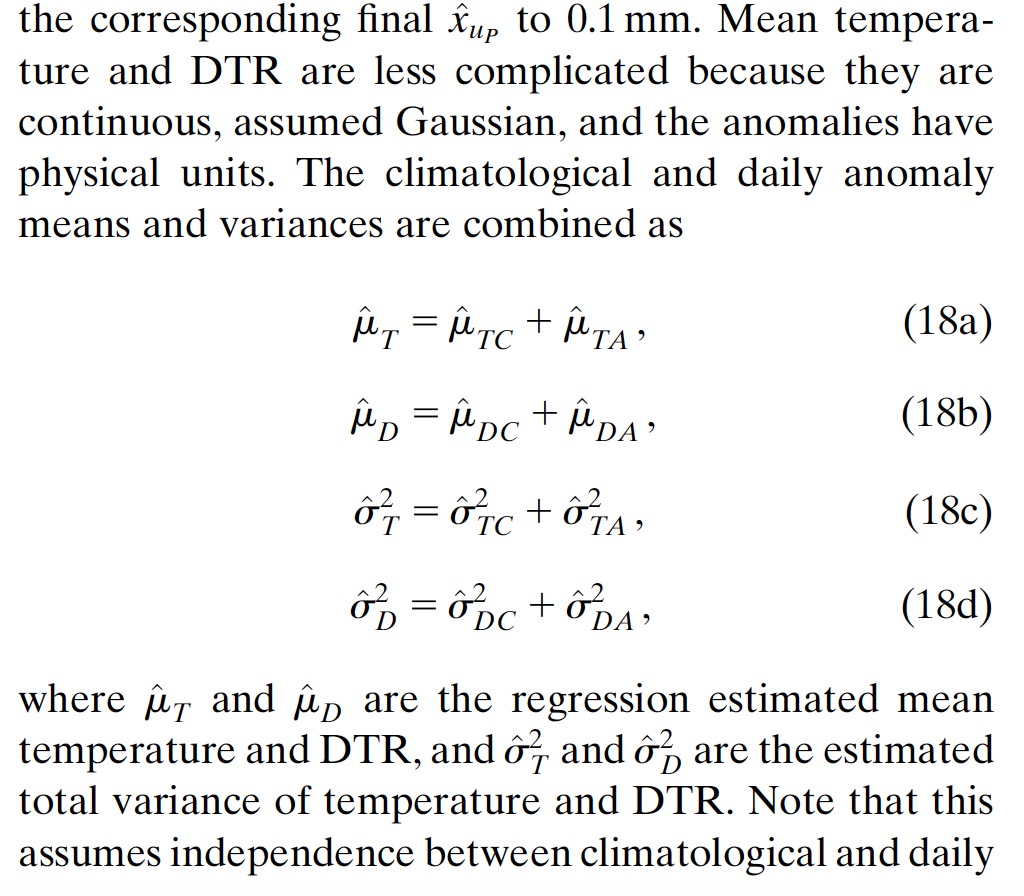
Then, daily anomalies are interpolated using regression, same with daily precipitation/temperature.

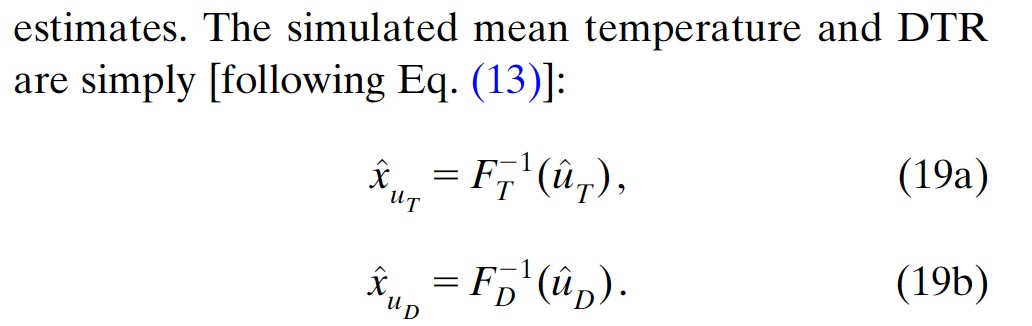
Then, daily anomaly ensembles are generated. For precipitation, daily anomaly ensemble is generated using Eq 16. For one ensemble, find its rank among all ensemble members (e.g., 100), and then find the same ranked ensemble of monthly climatology. Then, multiply the daily precipitation anomaly ratio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

## 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200km），不同站点之间的权重几乎没有差别，但是没关系，因为这个权重根本就没用上**。这个参数后面被替换掉了（见下面的“预处理阶段”）。此外，配置文件中**有一个参数maxdistance设置为100 km，但是在程序里面似乎就没用上？唯一用上的就是**“预处理阶段”的权重计算值？文章里的描述“computed the weights with MAXD set to 100km if all 30 stations were within 100 km.”没有体现。
* 开始循环
* 第一层循环:

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.
* 第二层循环:

对于每一个网格g（ngrid）

**预处理阶段**

* + 找到sta\_limit（30个）临近站点距离网格的最大距离（max\_distance）
  + 对降水，提取sta\_limit临近站点的临时权重（对角矩阵w\_pcp\_red）、位置/ID、降水（y\_red）、基本信息（x）。用max\_distance作为maxd计算得到了临时权重（tmp\_weight）。计算每个站点有无降水yp\_red（binary vector）。然后找到降水（y\_red）的最大值，存储在y\_max (g, t)中。
  + 对tmean和trange重复前面一步。
  + 检查station availability for precip and temp。如果没有临近站点，则赋予降水0值，温度前一个时刻（t-1）的值（如果t=1，温度赋值为-999）。

**Regression阶段：降水pop**

* + 要求临近站点至少一个降水>0，否则pop=0
  + **逻辑回归**，利用临近站点的降水得到目标网格的pop。此处有一个对网格坡度的要求，**如果坡度<3.6，说明网格很平，在逻辑回归的时候站点属性只使用[1, lat, lon, altitude]**。
  + **Least square线性回归**，得到目标网格的降水量pcp。Solve linear equation for x (Ax = b => x = bA^-1) using LU decomposition and back substitution。同时使用one-leave-out cross-validation的方法得到降水量估计误差pcperr，即对每一个临近站点，利用其余临近站点线性回归得到其降水，然后将估计降水与站点降水作差并求平方，然后对所有站点的误差估计值利用站点的权重进行加权平均，然后再开方，得到pcperr (g, t)，其实就是类似的**加权均方根误差**。

**Regression阶段：温度tmean trange**

* 必须有>=1个临近站点，否则采用前一个时刻的温度估计
* 温度估计时，只使用[1, lat, lon, altitude]，不考虑坡度slope属性
* **Least square线性回归**，跟降水量估计类似，得到温度和温度误差的估计

## Time\_mode is daily\_climo

**在执行完5.2之后，执行ensemble步骤（6），得到了daily-scale的ensemble，然后执行processing\_testexample里面的两个MATLAB程序**，其功能为：

1. read all ensemble members, and sort prcp from high to low, and calculate the std of tmean and trange save in netcdf
2. calculate the ratio (prcp) or difference (temperature) between station data and ensemble mean, and save those file in

通过这两个程序，基于站点数据和天尺度ensemble数据，得到了站点的climatology和anomaly数据，并存为了nc文件。

5.3与5.2基本一致，采用了slope判别。

## Time\_mode is daily\_anom

5.4的程序结构与5.2基本一致，但是没有slope判别，因为这部分不采用slope作为输入。此外，降水normalize的时候指数采用的是3，而不是4。

**所以执行顺序是daily、climo、anom**

# eCAI GMET program work flow: ensemble

**generate\_ensembles.f90**

## Preparation

* stop\_ens必须大于start\_ens
* 读取网格信息： call read\_nc\_grid
* 读取5中生成的regression的pop、pcp及其误差、tmean/trange及其误差、温度的auto\_corr、trange和pcp的相关tp\_corr、每个网格所有临近站点得到的最大降水ymax，如果ymax大于5，则赋值为5（5^4 =625 mm/day），**但是这个数值是可能会发生的**。
* 生成grid结构，存储的信息包括：grid%idx%spl1\_start =1， grid%idx%spl2\_start =1， grid%idx%spl1\_count=行数，grid%idx%spl2\_count=列数，grid%lat(spl1\_count, spl2\_count)，grid%lon(spl1\_count, spl2\_count)，grid%elv(spl1\_count, spl2\_count，经度lat、纬度lon、高程elv用第二步中读取网格数据进行赋值。

print \*, 'Generating weights for spatially correlated random field (SCRF)...'

call spcorr\_grd (nspl1, nspl2, grid)

sp\_pcp = spcorr

call field\_rand (nspl1, nspl2, pcp\_random)

! setup sp\_corr structure for temperature with larger correlation length

clen = 800.0 !rough estimate based on observations

call spcorr\_grd (nspl1, nspl2, grid)

sp\_temp = spcorr

call field\_rand (nspl1, nspl2, tmean\_random)

call field\_rand (nspl1, nspl2, trange\_random)

print \*, 'Done generating weights...'

## 生成spatially correlated random field (SCRF)

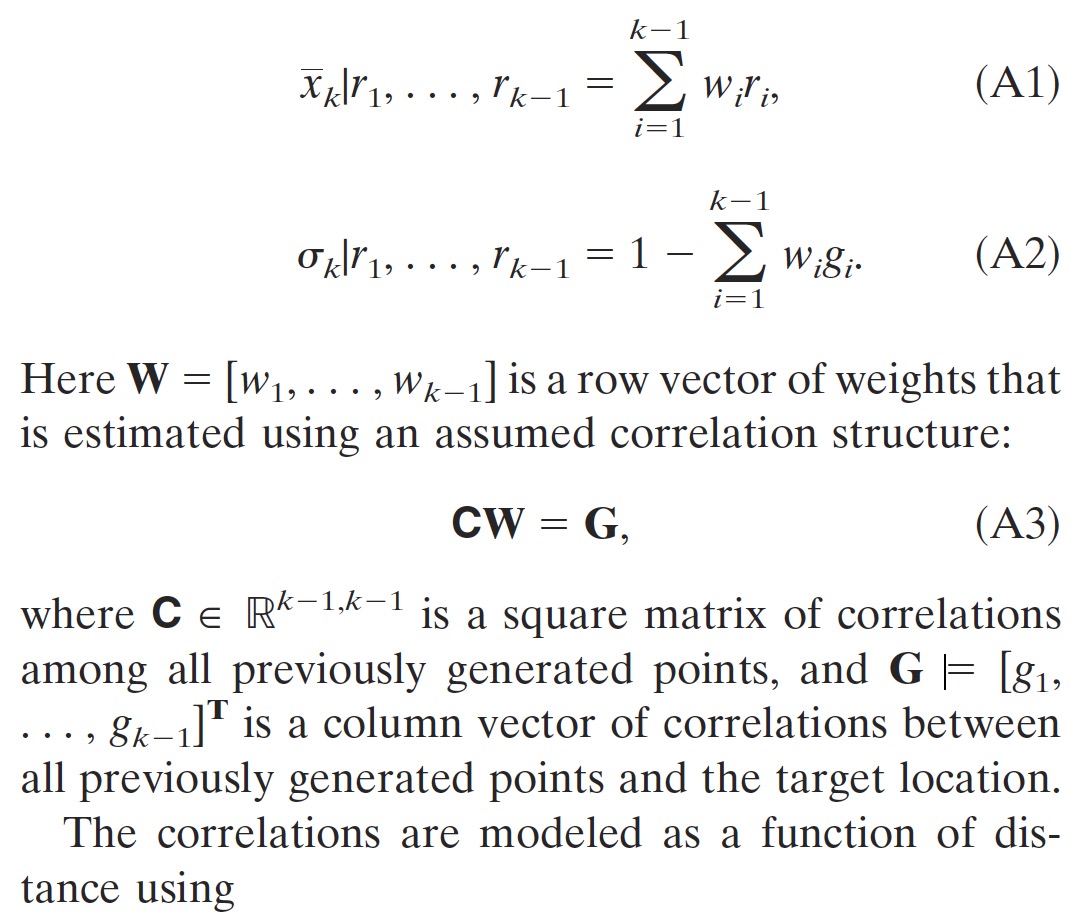
### 提取每个网格的若干个previously generated grids的位置、权重和标准差

**Spcorr\_grd.f90**

* 检查spcorr是否已经被赋值，如果是，return
* 定义超参数。
  + Number of nests (nnst) is 10: 2^9, 2^8, 2^7, …, 2^0.
  + number of local points to include in the estimation (nloc) is 3（其实就是对目标网格， 搜索previously generated网格的半径）.
  + Define the maximum number of previously generated points: maxp = (nloc\*2+1) \*\* 2
* 开始计算：
* 第一层循环：ires = nnst - 1, 0, - 1
  + incr = 2 \*\* ires ! increment (2\*\*9=512, …, 2\*\*3=8, 2\*\*2=4, 2\*\*1=2, 2\*\*0=1)
* 第二层循环：grid的第一个纬度: isp1 = 1, nspl1, incr (nspl1行数)
* 第三层循环：grid的第二个纬度: isp2 = 1, nspl2, incr (nspl2列数)

二三层循环就是跳跃式地生成随机数，每次跳跃（increment）就是incr，最后一个nest跳跃就是2\*\*0=1，遍历所有网格

* + 对当前网格（isp1, ips2），寻找previously generated points/grid。搜寻的半径是当前网格上下左右半径为(incr\*nloc)，例如2\*\*1 X 3 = 6个网格的半径长度。搜索窗口内，把已经generated的网格的位置提取出来，存为ipos, jpos。
  + 在ipos和jpos的最后，再添加上当前网格（isp1, ips2），总网格数目为k
  + 如果有>=2 个previous generated points，则：
    - 计算相关系数协方差矩阵：corr(k-1, k-1)，两个网格之间的相关系数为exp (-(dist/clen))，dist为网格间距，clen为通过站点得到的空间长度，温度被固定为800 km
    - 计算相关系数vector：gvec（k-1）, i.e. the correlation between current and previously generated points.
    - 计算权重
      * k=2，即只有一个previously generated point。权重wght (1) = gvec (1)，也就是两个点之间的相关系数。标准差sdev = sqrt (1.-gvec(1)\*\*2.)
      * k>2，那么k-1个previous points的权重(wght)通过求解以下的A3得到(Martyn 2006)，其中C也就是corr(k-1,k-1), G也就是gvec(k-1)。标准差sdev的平方由A2得到（图片中标准差符号少了2）



* + - * 把以上信息存在spcorr结构体中

spcorr(isp1, isp2)%ipos(1:npts-1) = ipos (1:npts-1)! i-position

spcorr(isp1, isp2)%jpos(1:npts-1) = jpos (1:npts-1)! j-position

spcorr(isp1, isp2)%wght(1:npts-1) = wght (1:npts-1)! weight assigned to previously generated points

spcorr(isp1, isp2)%sdev = sdev ! standard deviation of estimate

**6.2.1到此结束**

### Used to estimate a correlated field of random numbers for the basin gridpoints

**field\_rand.f90**

* 初始化随机数：cran(nlon, nlat)
* 对每一个网格循环（do igrd = 1, nlon \* nlat），生成grid随机数
  + 对第一个网格，直接从标准正态分布N(0,1)生成一个随机数(aran): call gasdev (aran)
  + igrd>=2.
    - 从6.2.1生成的spcorr找到当前网格的previously generated points（这些点已经有随机数了），提取这些points的随机数存储为vprv(nprv)，nprv就是6.2.1里面的k-1，也就是previous points的数目
    - 对这些previous points的随机数vprv，进行加权平均，权重在6.2.1
    - 从N(0,1)生成一个随机数aran: gasdev (aran)，然后用这个随机数乘以previous points的标准差
    - 前面两步的结果相加，得到igrd的随机数cran (ilon, ilat)：

call gasdev (aran)

xbar = dot\_product (vprv(1:nprv), spcorr(ilon, ilat)%wght(1:nprv))

cran (ilon, ilat) = xbar + spcorr(ilon, ilat)%sdev \* aran

* cfield = cran，输出，结束

## 生成ensemble estimates

* 初始化，注意三个time\_mode之间的差异，尤其是daily\_anom

if(trim(time\_mode) .eq. 'daily' .or. trim(time\_mode) .eq. 'DAILY') then

! set transform power, shouldn't be hard-coded, but it is for now...

transform = 4.0d0

elseif(trim(time\_mode) .eq. 'climo' .or. trim(time\_mode) .eq. 'CLIMO') then

transform = 4.0d0

elseif(trim(time\_mode) .eq. 'daily\_anom' .or. trim(time\_mode) .eq. 'DAILY\_ANOM') then

t**ransform = 3.0d0**

**allocate(climo\_precip(100,nx,ny)) !climo precip grid**

**allocate(climo\_tmean(nx,ny)) !climo tmean grid**

**allocate(climo\_trange(nx,ny)) !climo trange grid**

**allocate(uncert\_tmean(nx,ny)) !uncert tmean grid**

**allocate(uncert\_trange(nx,ny)) !uncert trange grid**

else

print \*, 'Incorrect time mode: ',trim(time\_mode)

print \*, 'Current options are: daily, climo, daily\_anom'

stop

end if

* 第一层循环（ensemble member）：iens = start\_ens, stop\_ens
* 第二层循环（time steps）: istep = 1, ntimes
* 第三层循环（网格）： igrd = 1, nspl1 \* nspl2
  + 对于daily\_anom模式，需要read climatological fields，读取当前月份的pcp\_sorted，t\_mean，t\_range，t\_mean\_stddev，t\_range\_stddev。这些在5.3里介绍了，是daily ensemble生成后，使用MATLAB程序生成的climatological fields
  + 针对当前网格（isp1, isp2）

**对于降水**

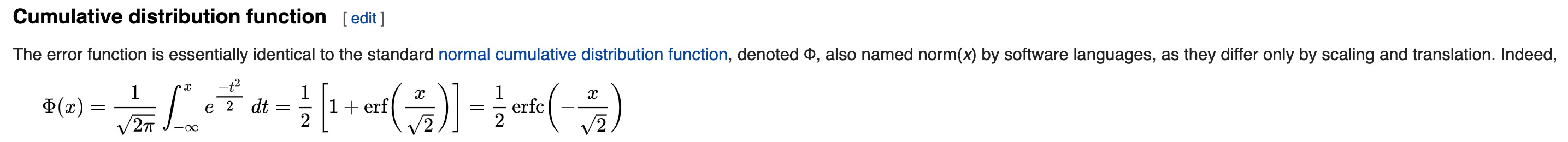
* + - **生成对应当前随机数的累计概率cumulative probability (cprob)**

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

aprob = erfcc (acorr) ! 互补误差函数***erfc(x) = 1-erf(x)****. erf(x)是x的误差函数−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）。****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 below equation, **cprob is actually the cumulative probability ( corresponding to x (the random number, ~N(0,1))**. So, the random number is actually used to represent the cumulative probability that a certain value of precipitation or temperature occurs.

**

* + - 如果**cprob<这个网格的pop（第5.2步），降水赋值为0。**
    - 如果**cprob>=这个网格的pop**
      * 第一步rescale cprob:

cs = (cprob – (1 – pop)) / pop

* + - * **convert cs to a z-score from standard normal。实际上**rn = sqrt (2.\_sp) \* erfinv ((2.\_sp\*real(cs, kind(sp)))-1.0\_sp)是对以上截图求解x，即给定累计概率密度cs，求解在N(0,1)分布下对应的数值rn，rn是正态分布下的数值，之后要转换为实际降水的分布N(mean,std2)，所以也可以将rn直接理解为z-scorre = **(实际数值-mean)/std**

if cs<3e-5, rn =-3.99

if cs>0.99997, rn=3.99

else: rn = sqrt (2.\_sp) \* erfinv ((2.\_sp\*real(cs, kind(sp)))-1.0\_sp)

* 找到cs对应的百分位数，cs\_percentile = ceiling(cs\*100.)
* **生成ensemble降水**

**time\_mode = daily**

* + ra = pcp (isp1,isp2,istep) + rn\*pcp\_error (isp1,isp2,istep)，pcp和pcp\_error都是第5步生成的，rn就是上一步生成的z-score
  + ra = ((ra\*(1.0/transform))+1.0\_dp)\*\*transform。ra就是转化回去的降水，这一步括号里面是干啥的，搞不懂？
  + 限制ra的最大值，使得最大降水在一定范围内，obs\_max = 1.5\*((((obs\_max\_pcp(isp1,isp2,istep)+0.2\*cs)\*(1.0/transform))+1.0)\*\*transform)。这个公式也是很奇怪，纯粹经验？
  + ra输出为最后降水
  + 如果最终将水<0.1，则将其赋值为0.1

**time\_mode = climo**

* + 与之前类似，得到ra
  + ra赋值为最终降水，如果小于0.01则赋值为0.01

**time\_mode = daily\_anom**

* + 与daily类似，生成ra
  + 在daily所有ensemble中（已经在daily\_anom之前完成估计了），降水从大到小排列，找到对应于cs百分位数的那个ensemble，然后将这个ensemble的网格降水乘以ra，得到最终降水
  + 然后又是一个最大降水限制，通过一些看似经验性的公式得到
  + 如果最终将水<0.1，则将其赋值为0.1
    - 生成ensemble温度

**time\_mode = daily**

* ra = trange (isp1,isp2,istep) + trange\_random(isp1,isp2)\*trange\_error (isp1,isp2,istep)。这里不跟降水那么复杂，要估计什么cprob cs。trange\_random直接相当于降水里的z-score，也就是rn。
* ra为最终的trange
* tmean同样的步骤

**time\_mode = climo**

* 跟daily同样的步骤

**time\_mode = daily\_anom**

* 估计combined error： combined\_error = sqrt(tmean\_error\*\*2 + uncert\_tmean\*\*2)。这是两个正太分布相加，其中tmean\_error是第五步中regression得到的（one-leave-out cross validation），uncert\_tmean是daily下所有ensemble估计值的标准差。
* ra = tmean (isp1,isp2,istep) + climo\_tmean(isp1,isp2) + tmean\_random(isp1,isp2) \* combined\_error，其中tmean是基于anomaly得到的，climo\_tmean是对应于整个月的（在climo）中生成，两者相加相当于当前的平均值，tmean\_random相当于z-score，combined\_error是标准差。ra是最终的tmean的估计值。
* trange的流程一样的
  + - check for unrealistic and non-physical trange values。Trange大于40则设置为40（这个是不是也有点小了？），tmean超出正负50则设置为正负50（这个范围有点小了？）。

**至此第二层循环结束，进入下一个time step**

* 仍处于（第二层循环（time steps）: istep = 1, ntimes），对于下一个time step，生成新的SCRF
  + 调用field\_rand，基于spcorr（温度和降水不同，在6.2.1中生成，整个程序不会改变数值），生成新的srcf

以下是针对tmean，trange 是一样的

spcorr = sp\_temp

old\_random = tmean\_random

call field\_rand (nspl1, nspl2, tmean\_random)

tmean\_random = old\_random \* auto\_corr (1) + sqrt (1-auto\_corr(1)\*auto\_corr(1)) \* tmean\_random

以下是针对pcp降水

spcorr = sp\_pcp

call field\_rand (nspl1, nspl2, pcp\_random)

pcp\_random = trange\_random \* tpc\_corr (1) + sqrt (1-tpc\_corr(1)\*tpc\_corr(1)) \* pcp\_random

**整个程序结束！**