double maxCost- the maximum cost per kWh in the highest period of the day, Note calculate this while entering price data

double minCost the minimum cost per kWh in the lowest cost period of the day, Note calculate this while entering price data

double eLoss-=.80 our energy conversion loss

xMax- maximum charging rate either capacity/4 or capacity/8 for a battery with 45%DOD for c/4 it takes 4 hours to charge, 8 for c/8,

surcharge=$3  (is the cost per kWh of usage during the peak hour each day, $3 used in paper so greedy approach is not used as much)

T is the length of the low-price period(in real time, likely to be only 1 hour on our case)

maxCapacity-usable energy storage capacity that is 50% of their average daily demand.

avgDemand-average daily demand of electricity

avgCost-average  daily electric cost

capacity=current amount of battery charged, can’t be negative or greater than maxCapacity)

#define MAXCAPACITY (some number of maxCapacity)

#define MINCAPACITY (some number of minCapacity)

bool maxChargeRate = true;

/\*\*

\* charge for one hour using charge rate

\*/

Function chargeBattery(double chargeRate){

if (capacity!= MAXCAPACITY){

double tempcapacity=capacity;

double totalChargeHour=0;

//Check to make sure capacity is not > maxCapacity

//if it is change capacity to maxCapacity

if (chargeRate!=maxChargeRate){

capacity += chargeRate\*eLoss;

}

// If we are using the max charge rate,

// This should only charge1/4 of the maxCapacity!

else{

capacity += MAXCAPACITY/4;

}

//the difference between the old capacity and the new

totalChargeHour=capacity-tempcapacity;

if(capacity>MAXCAPACITY){

//note in this case we still wasted the energy trying to fully charge the already fully charged battery

capacity= MAXCAPACITY;

}

}

return totalChargeHour;

}

/\*\*

\* Discharge for one hour using charge rate

\*/

Function dischargeBattery(double dischargeRate){

//Check to make sure capacity is not < 0 if it is change capacity to 0

double tempcapacity=capacity;

double totalChargeHour=0;

if (chargeRate!=maxChargeRate){

//gets the requesting energy accounting for eLoss

capacity -= dischargeRate/eLoss;

}

//remember if we are using the max charge rate,

// this should only discharge1/4 of the maxCapacity!

else{

capacity -= MAXCAPACITY/4;

}

totalChargeHour= tempcapacity-capacity;

//note if we try to discharge more energy than there is, we can only get as much energy there is

if(capacity<0){

capacity=0;

totalChargeHour=tempcapacity;

}

return totalChargeHour;

}

Function batteryBehavior(){

// each hour of data/price in our vector vectDataI, vectPricei

for (int i=0;i<72,i++){

if(vectPriceI <=avgCost){

if(vectDatai<=avgDemand){

if((xMax\*eLoss\* maxCost \*T )− (xMax\* minCost \*T )> xMax\*surcharge){

//charge max rate

chargeBattery (XMax));

//add charged energy-to-energy data for that hour

vectDatai+=chargeRate

}

else{

//charge at rate to sustain the target avgDemand

double chargeRate=avgDemand-vectDatai;

double charge=chargeBattery (chargeRate);

//add charged energy-to-energy data for that hour

vectDatai+=chargeRate

}

}

else{

if((xMax\*eLoss\* maxCost \*T )− (xMax\* minCost \*T )> xMax\*surcharge){

//charge max rate

chargeBattery (XMax);

//add charged energy-to-energy data for that hour

vectDatai+=chargeRate

}

else{

//discharge at rate to sustain the target avgDemand

//this is how much energy is needed including energy loss

double dischargeRate=vectDatai-avgDemand;

double discharge=dischargeBattery (dischargeRate);

//note battery may not have had enough energy to get to target demand, may want to print something if that is the case?

vectDatai-=discharge\*eLoss;

}

}

}

else{

if(vectDatai<=avgDemand){

if((xMax \* eLoss \* maxCost \*T )− (xMax \* minCost \*T )> xMax \*surcharge){

//discharge max rate

//again might not have enough energy in the battery

double discharge=dischargeBattery(XMax);

vectDatai-=discharge\*eLoss;

}

}

else{

//do nothing

}

}

else{

if((xMax\*eLoss\* maxCost \*T )− (xMax\* minCost \*T )> xMax\*surcharge){

//discharge max rate

//again might not have enough energy in the battery

double discharge=dischargeBattery(XMax);

vectDatai-=discharge\*eLoss;

}

else{

//discharge at rate to sustain the target avgDemand

//this is how much energy is needed including energy loss

double dischargeRate=vectDatai-avgDemand;

double discharge=dischargeBattery (dischargeRate);

//note battery may not have had enough energy to get to target demand, may want to print something if that is the case?

vectDatai-=discharge\*eLoss;

}

}

}

}