

In[1046]:=

Uživatelské nastavení parametrů ;

```
vyska = 1; (* m; vyska vnitřní části zásobníku teple vody *)
objem = 220; (* l; objem zásobníku *)
teplotaOhrivaneVody = 55; (* °C; teplota, na kterou chci ohrivat vodu *)
teplotaStudeneVody = 15; (* °C;
teplota neohrate vody, která vteka do zásobníku *)
vykonBojleru = 2400; (* W *)
periodaSpinani = 14000; (* s *)
c = 4181; (*  $\frac{J}{kg \cdot K}$  *)
rho = 990; (*  $\frac{kg}{m^3}$  *)
alpha = 1.5 * 10^-7; (*  $\frac{W}{m \cdot K}$  *)
OffsetDown = 50; (* udava,
jaka muze byt minimalni teplota vody v zásobníku *)
```

In[1053]:=

Parametry a deklarace nutné pro chod programu (nezadáva uživatel);

```
$HistoryLength = 2;
SetDirectory[NotebookDirectory[]];
odbery = Import["odbery.xlsx"][[1]];
teploty = Import["teploty3.xlsx"][[1]];
aktualniTeplota = 55;
ohratCele = 0;
den = 1;
fceTeplot = {};
dodatTepla = 10^-3 * objem * c * rho * (teplotaOhrivaneVody - aktualniTeplota);
Q = 0; (* J *)
Qtotal = 0;
prumer = 2 * Sqrt[ $\frac{\text{objem} * 10^{-3}}{\text{Pi} * \text{vyska}}$ ]; (* metr *)
```

In[1065]:=

```
(* zobrazí teplotu vody v case
vstup:
vystup: *)
ZobrazfceTeplaVoda[] := Plot[fceTeplaVoda[x * 3600], {x, 0, 24},
GridLines -> Automatic, AxesLabel -> {"cas [hod]", "teplota vody [°C]"},
PlotLabel -> "Teplota vody v case", PlotRange -> {0, 60}];
```

In[1066]:=

```
(* zobrazí v case kolik tepla je nutne
dodat pro ohrati veskere vody na pozadovanou teplotu
vstup:
vystup: *)
ZobrazfceDodatTepla[] := Plot[fceDodatTepla[x * 3600] * 10^-6, {x, 0, 24},
GridLines -> Automatic, AxesLabel -> {"cas [hod]", "chybejici teplo [MJ]"},
PlotLabel -> "Teplo pro ohrev vody v case", PlotRange -> All];
```

```
In[1067]:= (* zobrazí odbery (objemovy prutok) v case
vstup:
vystup: *)
ZobrazfceOdbery[] := Plot[fceOdbery[x * 3600], {x, 0, 24},
GridLines → Automatic, AxesLabel → {"cas [hod]", "objemovy prutok [l/s]"},
PlotLabel → "Odber teple vody v case", PlotRange → All];
```

```
In[1068]:= Funkce řešící první větu termodynamickou;
```

```
(* resi 1. veta termodynamickou
vstup: doba trvani daneho casoveho useku,
dodany tepelny tok, odebirany hmotnosti tok
vystup: reseni 1. vety termodynamicke *)
VyresVetuTermodynamiky[tend_, Qsteckou_, msteckou_] := Module[{heq, poc, sol},
heq = 10^(-3) * c * ρ * objem * D[T[t], t] ==
Qsteckou + c * msteckou * (teplotaStudeneVody - T[t]);
poc = T[0] == aktualniTeplota;
sol = Quiet[NDSolve[{heq, poc}, T[t],
{t, 0, tend}, MaxStepSize → { $\frac{\text{tend}}{1000}$ }, AccuracyGoal → 3]];
aktualniTeplota = T[t] /. sol[[1]] /. t → tend;
{sol}];
```

```
In[1070]:= (* resi 1. veta termodynamickou
vstup: doba trvani daneho casoveho useku,
dodany tepelny tok, odebirany hmotnosti tok, aktualni teplota
vystup: koncova teplota *)
VetaTermodynamiky[tend_, Qsteckou_, msteckou_, teplota_] :=
Module[{heq, poc, sol, koncovaTeplota},
heq = 10^(-3) * c * ρ * objem * D[T[t], t] ==
Qsteckou + c * msteckou * (teplotaStudeneVody - T[t]);
poc = T[0] == teplota;
sol = Quiet[NDSolve[{heq, poc}, T[t],
{t, 0, tend}, MaxStepSize → { $\frac{\text{tend}}{100}$ }, AccuracyGoal → 3]];
koncovaTeplota = T[t] /. sol[[1]] /. t → tend;
koncovaTeplota];
```

```
In[1071]:= (* pocita teplotu vody a teplo,
ktere je nutne dodat vode aby se ohrala na pozadovanou uroven,
pridava hodnoty teploty vody do Listu teplotaVodyVCase
vstup: zacatek zkoumaneho obdobi, konec zkoumaneho obdobi,
celkove dodane teplot, celkova odebrana hmotnost vody
vystup:
*)
PrestupTepla[tstart_, tend_, dodaneTeplo_, odber_] := Module[{posun, sol},
  If[tend > tstart, {
    VyresVetuTermodynamiky[tend - tstart,  $\frac{\text{dodaneTeplo}}{\text{tend} - \text{tstart}}$ ,  $\frac{\text{odber}}{\text{tend} - \text{tstart}}$ ][[1, 1]];
    dodatTepla = c * ρ * objem * (teplotaOhrivaneVody - aktualniTeplota);
    teplotaVodyVCase = Append[teplotaVodyVCase, {tend, aktualniTeplota}];
    dodatTeplaVCase = Append[dodatTeplaVCase, {tend, dodatTepla}];
  }];
```

In[1072]= **Jednotlivé varianty ohřevu a odběru vody ;**
Varianta 1 – přímý ohřev kdykoliv je to možné;

```
(* Funkce pro variantu ohrevu cislo 1,
kdy se voda v zasobniku ohriva pomoci topneho telesa
kdykoliv je to mozne. Dodana energie je teda primo umerna
spotrebe. Vstupem do funkce je casovy usek, kdy probiha odber,
pripadne casovy usek mezi dvema odbery a odebrany objem teple
vody v litrech. Jelikoz se cely casovy usek odberu vyhodnocuje
pouze jednou na zacatku odberu, je zde v pripade odberu delsiho
nez casova konstanta periodaSpinani zarazen while cyklus,
ktery umoznuje vyhodnocovat a pripadne i realizovat odber vzdy po
casovem useku periodaSpinani. Pokud by v prubehu casoveho useku
na vstupu zacal vysoky tarif, zavola se funkce VysokyTarifBojler[]
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
NizkyTarifBojler[tstart_, tend_, odber_] :=
Module[{ret, trvani, pristi, cas, Q, tstarthelp, tendhelp},
  If[tend - tstart > periodaSpinani, {
    tstarthelp = tstart;
    tendhelp = tstarthelp + periodaSpinani;
    While[tstarthelp < tend,
      {Q = c * ρ *  $\frac{\text{Pi} * \text{prumer}^2}{4}$  * vyska * (teplotaOhrivaneVody - aktualniTeplota);
      (* kolik tepla je potreba aktualne dodat *)
      {ret, trvani, pristi} = NizkyTarifOhrevuVody[tstarthelp];
```


kdy se voda v zasobniku ohřiva pomocí tepelného čerpadla
 kdykoliv je to možné. Dodaná energie je závislá na COP čerpadla,
 které je závislé na aktuální venkovní teplotě. Vstupem
 do funkce je časový úsek, kdy probíhá odběr,
 případně časový úsek mezi dvěma odběry a odebraný objem teple
 vody v litrech. Jelikož se celý časový úsek odběru vyhodnocuje
 pouze jednou na začátku odběru, je zde v případě odběru delšího
 než časová konstanta periodaSpinani zarazen while cyklus,
 který umožňuje vyhodnocovat a případně i realizovat odběr vždy po
 časovém úseku periodaSpinani. Pokud by v průběhu časového úseku na
 vstupu začal vysoký tarif, zavolá se funkce VysokyTarifCerpadlo[]
 vstup: časový úsek jednoho odběru nebo pauzy mezi odběry,
 objem odebrané vody v litrech
 výstup: *)

```
NizkyTarifCerpadlo[tstart_, tend_, odber_] := Module[
  {ret, trvani, pristi, cas, Q, QCerpadlo, tendCerpadlo, tstarthelp, tendhelp},
  If[tend - tstart > periodaSpinani, {
    tstarthelp = tstart;
    tendhelp = tstarthelp + periodaSpinani;
    While[tstarthelp < tend,
      {{ret, trvani, cas} = VypocetCasu[tstarthelp, tendhelp];
      If[tstarthelp + trvani < tendhelp && tend ≠ tendhelp,
        {Ohrev[cas, tstarthelp, tstarthelp + trvani,  $\frac{\text{trvani}}{\text{tend} - \text{tstart}} * \text{odber}$ ,
          NIntegrate[TCPtep[fceTeplot[[den]][t], teplotaOhrivaneVody],
            {t, tstarthelp, tstarthelp + trvani}, AccuracyGoal → 0], 2],
          VysokyTarifCerpadlo[tstarthelp + trvani, tend,
            odber -  $\frac{\text{tstarthelp} + \text{trvani} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$ ], tstarthelp = tend},
        {Ohrev[cas, tstarthelp, tendhelp,  $\frac{\text{tendhelp} - \text{tstarthelp}}{\text{tend} - \text{tstart}} * \text{odber}$ ,
          NIntegrate[TCPtep[fceTeplot[[den]][t], teplotaOhrivaneVody],
            {t, tstarthelp, tstarthelp + cas}, AccuracyGoal → 0], 2];
        If[tstarthelp + periodaSpinani ≥ tend && ret == 1,
          tstarthelp = tendhelp, tstarthelp = tstarthelp + periodaSpinani];
        If[tendhelp + periodaSpinani ≥ tend && ret == 1, tendhelp = tend,
          tendhelp = tendhelp + periodaSpinani]}}] ]],
  {{ret, trvani, cas} = VypocetCasu[tstart, tend];
  Ohrev[cas, tstart, tend, odber,
    NIntegrate[TCPtep[fceTeplot[[den]][t], teplotaOhrivaneVody],
      {t, tstart, tstart + cas}, AccuracyGoal → 0], 2];
  ];
```

```
In[1078]:= (* Urci, kdy zacne platit nizky tarif a v danou dobu zavola
prislusnou funkci. Pokud nizky tarif nezacina primo v cas tstart,
tak se zavola funkce PrestupTepla pokryvajici casovy
interval tstart az zacatek startu nizkeho tarifu
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
VysokyTarifCerpadlo[tstart_, tend_, odber_] := Module[{ret, trvani, pristi},
  {ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
  If[pristi < tend,
    {If[tstart ≠ pristi, PrestupTepla[tstart, pristi, 0,  $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$ ]},
    NizkyTarifCerpadlo[pristi, tend, odber -  $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$ ]},
    {PrestupTepla[tstart, tend, 0, odber]}];
  ];
```

In[1079]:=

Varianta 3 – ohřev pomocí TČ když klesne teplota TV pod danou úroveň;

```
(* Funkce pro variantu ohrevu cislo 3,
kdy se voda v zasobniku ohriva pomoci tepelneho cerpadla v pripade,
kdy teplota vody klesne pod urcenu hodnotu OffsetDown. Dodana energie
je zavisla na COP cerpadla, ktere je zavisle na aktualni venkovni
teplote. Vstupem do funkce je casovy usek, kdy probiha odber,
pripadne casovy usek mezi dvema odbery a objem odebrane teple
vody v litrech. Na zacatku funkce se spocita casOdkladu,
ktery rika za jak dlouho bude s aktualnim odberem teplota vody na
hranicni hodnotu. Pokud tento okamzik nastane za dobu aktualniho odberu,
spusti se ohrev. Pokud by aktualni odber skoncil drive,
nez bude voda ohrata na pozadovanou teplotu,
ohrev bude pokracovat i v dalsim odberu/pauze mezi odbery. Pokud
by v prubehu casoveho useku na vstupu zacial vysoky tarif,
zavola se funkce VysokyTarifCerpadlo[]
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
NizkyTarifCerpadloOffset[tstart_, tend_, odber_] :=
Module[{casOdkladu, ret, trvani, pristi, cas, ohrev,
  bilance, Q, QCerpadlo, tendCerpadlo, tstart1, odber1},
  tstart1 = tstart;
  odber1 = odber;
  If[odber1 > 0, casOdkladu =
    VypocetCasuOdkladu[tstart, tend, odber, OffsetDown], casOdkladu = 0];
  If[casOdkladu < tend - tstart1 && casOdkladu > 0 && tstart ≠ tend,
```

```

{PrestupTepla[tstart1, tstart1 + cas0dkladu, 0,  $\frac{\text{cas0dkladu}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ],
  tstart1 = tstart1 + cas0dkladu, odber1 -=  $\frac{\text{cas0dkladu}}{\text{tend} - \text{tstart1}} * \text{odber1}$ },
{If[cas0dkladu > tend - tstart1 && tstart1 ≠ tend, {PrestupTepla[tstart1,
  tend, 0, odber1], tstart1 = tend, odber1 = 0}], cas0dkladu = 0}];
{ret, trvani, cas} = VypocetCasu[tstart1, tend];
If[tend > tstart1 &&
  (VetaTermodynamiky[tend - tstart1, 0,  $\frac{10^{(-3)} * \text{odber1} * \rho}{\text{tend} - \text{tstart1}}$ , aktualniTeplota] <
    OffsetDown || aktualniTeplota < OffsetDown ||
    ohratCele == 1) && tstart1 + cas < tend && cas > 60,
{If[tend > tstart1 && VetaTermodynamiky[tend - tstart1, 0,
   $\frac{10^{(-3)} * \text{odber1} * \rho}{\text{tend} - \text{tstart1}}$ , aktualniTeplota] < OffsetDown, ohratCele = 1];
ohrev = NIntegrate[TCPtep[fceTeplot[[den]][t], teplotaOhrivaneVody],
  {t, tstart1, tstart1 + cas}, AccuracyGoal → 0];
dodanaEnergie = Append[dodanaEnergie, {tstart1 + cas,
  NIntegrate[TCPel[fceTeplot[[den]][t], teplotaOhrivaneVody],
  {t, tstart1, tstart1 + cas}, AccuracyGoal → 0]}}];
PrestupTepla[tstart1, tstart1 + cas, ohrev,  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ],
  odber1 -=  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ , tstart1 = tstart1 + cas;
}, cas = 0];
{ret, trvani, cas} = VypocetCasu[tstart1, tend];
If[aktualniTeplota > teplotaOhrivaneVody - 1, ohratCele = 0];
If[tend > tstart1 + cas + 1 && cas > 0 && odber1 > 0 && tstart1 ≠ tend,
{NizkyTarifCerpadloOffset[tstart1 + cas, tend,
  odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ], {ret, trvani, pristi} =
  NizkyTarifTepelnehoCerpadla[tstart1 + cas], If[ret == 1 && tstart1 ≠ tend,
  {PrestupTepla[tstart1, tend, 0,  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ]}],
  If[tstart1 ≠ tend && tstart1 + cas < tend, VysokyTarifCerpadloOffset[
    tstart1 + cas, tend, odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ]]]]];
];

```

```
In[1081]:= (* Urci, kdy zacne platit nizky tarif a v danou dobu zavola
prislusnou funkci. Pokud nizky tarif nezacina primo v cas tstart,
tak se zavola funkce PrestupTepla pokryvajici casovy
interval tstart az zacatek startu nizkeho tarifu
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
VysokyTarifCerpadloOffset[tstart_, tend_, odber_] :=
Module[{ret, trvani, pristi},
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
If[pristi < tend && tstart ≠ tend,
{If[tstart ≠ pristi, PrestupTepla[tstart, pristi, 0,  $\frac{pristi - tstart}{tend - tstart} * odber$ ]},
NizkyTarifCerpadloOffset[pristi, tend, odber -  $\frac{pristi - tstart}{tend - tstart} * odber$ ]},
{PrestupTepla[tstart, tend, 0, odber]}];
];
```

In[1082]:=

Varianta 4 – ohřev pomocí TČ optimalizováno dle teploty;

```
(* Funkce pro variantu ohrevu cislo 4,
kdy se voda v zasobniku ohriva pomoci tepelneho cerpadla v pripade,
kdy teplota vody klesne pod urcenou hodnotu OffsetDown nebo je
idealni casovy usek pro ohrev vody optimalizovany dle venkovni
teploty. Dodana energie je zavisla na COP cerpadla, ktere je zavisle
na aktualni venkovni teplotě. Vstupem do funkce je casovy usek,
kdy probiha odber, pripadne casovy usek mezi dvema odbery a objem
odebrane teple vody v litrech. Na zacatku funkce se spocita casOdkladu,
ktery rika za jak dlouho bude s aktualnim odberem teplota vody na
hranicni hodnote nebo za jak dlouho bude idealni casovy usek pro
ohrev vody. Pokud tento okamzik nastane za dobu aktualniho odberu,
spusti se ohrev. Pokud by aktualni odber skoncil drive,
nez bude voda ohrata na pozadovanou uroven,
ohrev bude pokracovat i v dalsim odberu/pauze mezi
odbery. Pokud se ohrev spusti z duvody prislis nizke teploty vody,
ohrev bude trvat jen kratkou chvili (15 min), aby voda nedosla,
ale zaroven se nebude ohrivat cely zasobnik mimo idealni cas
ohrevu. Pokud by v prubehu casoveho useku na vstupu zacal vysoky tarif,
zavola se funkce VysokyTarifCerpadlo[]
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery,
objem odebrane vody v litrech
vystup: *)
NizkyTarifCerpadloOptimalizace[tstart_, tend_, odber_] :=
Module[{casOdkladu, casOdkladu1, casOdkladu2,
ret, trvani, pristi, cas, ohrev, bilance, Q, QCerpadlo,
```

```

tendCerpadlo, tstart1, odber1, cas0hrevu, index, help},
tstart1 = tstart;
odber1 = odber;
index = 1;
help = 1;
While[casy0hrevu[[index, 2]] < tstart1 && help == 1,
  If[index + 1 ≤ Length[casy0hrevu], index += 1, help = 0];
];
cas0hrevu = casy0hrevu[[index]];
cas0dkladu1 = cas0hrevu[[1]] - tstart1;
If[odber1 > 0, cas0dkladu2 =
  VypocetCasuOdkladu[tstart, tend, odber, OffsetDown], cas0dkladu2 = 0];
If[(cas0dkladu1 ≤ cas0dkladu2 || cas0dkladu2 == 0) && cas0dkladu1 > 0,
  {cas0dkladu = cas0dkladu1, ohratCele = 1}, cas0dkladu = cas0dkladu2];
If[cas0dkladu < tend - tstart1 && cas0dkladu > 0 && tstart ≠ tend,
  {PrestupTepla[tstart1, tstart1 + cas0dkladu, 0,  $\frac{\text{cas0dkladu}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ],
  tstart1 = tstart1 + cas0dkladu, odber1 = odber1 -  $\frac{\text{cas0dkladu}}{\text{tend} - \text{tstart1}} * \text{odber1}$ },
  {If[cas0dkladu > tend - tstart1 && tstart1 ≠ tend,
  {{ret, trvani, cas} = VypocetCasu[tstart1, tend],
  If[tstart1 > cas0hrevu[[1]] && tstart1 < cas0hrevu[[1]] + cas,
  {PrestupTepla[tstart1, tend, NIntegrate[TCPTep[fceTeplot[[den]][t],
  teplotaOhrivaneVody], {t, tstart1, tstart1 + cas}, AccuracyGoal → 0],
  odber1], dodanaEnergie = Append[dodanaEnergie, {tstart1 + cas,
  NIntegrate[TCPEl[fceTeplot[[den]][t], teplotaOhrivaneVody],
  {t, tstart1, tstart1 + cas}, AccuracyGoal → 0]}]},
  {PrestupTepla[tstart1, tend, 0, odber1]}], tstart1 =
  tend, 0}], cas0dkladu = 0}];
{ret, trvani, cas} = VypocetCasu[tstart1, tend];
If[cas0hrevu[[1]] > tstart1 && cas > 900, {cas = 900}];
If[tend > tstart1 &&
  (VetaTermodynamiky[tend - tstart1, 0,  $\frac{10^{(-3)} * \text{odber1} * \rho}{\text{tend} - \text{tstart1}}$ , aktualniTeplota] <
  OffsetDown || aktualniTeplota < OffsetDown ||
  ohratCele == 1) && tstart1 + cas < tend && cas > 60, {
ohrev = NIntegrate[TCPTep[fceTeplot[[den]][t], teplotaOhrivaneVody],
  {t, tstart1, tstart1 + cas}, AccuracyGoal → 0];
PrestupTepla[tstart1, tstart1 + cas, ohrev,  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ],
dodanaEnergie =
  Append[dodanaEnergie, {tstart1 + cas, NIntegrate[TCPEl[fceTeplot[[den]][t],
  teplotaOhrivaneVody], {t, tstart1, tstart1 + cas}, AccuracyGoal → 0]}],
odber1 = odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ , tstart1 = tstart1 + cas;

```

```

    }, cas = 0];
    {ret, trvani, cas} = VypocetCasu[tstart1, tend];
    If[casOhravu[[1]] > tstart1 && casOhravu[[1]] < tend && cas > 900, {cas = 900}];
    If[aktualniTeplota > teplotaOhrivaneVody - 1 ||
      tstart1 < casOhravu[[1]] || tstart1 > casOhravu[[2]], ohratCele = 0];
    If[tend > tstart1 + cas + 1 && cas > 0 && odber1 > 0 && tstart1 ≠ tend,
      {NizkyTarifCerpadloOptimalizace[tstart1 + cas, tend,
        odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ], {ret, trvani, pristi} =
        NizkyTarifTepelnehoCerpadla[tstart1 + cas], If[ret == 1 && tstart1 ≠ tend,
        PrestupTepla[tstart1, tend, 0,  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ],
        If[tstart1 ≠ tend && tstart1 + cas < tend, VysokyTarifCerpadloOptimalizace[
          tstart1 + cas, tend, odber1 -  $\frac{\text{cas}}{\text{tend} - \text{tstart1}} * \text{odber1}$ ]]]]];
  ];

```

In[1084]:= (* Urci, kdy zacne platit nizky tarif a v danou dobu zavola prislusnou funkci. Pokud nizky tarif nezacina primo v cas tstart, tak se zavola funkce PrestupTepla pokryvajici casovy interval tstart az zacatek startu nizkeho tarifu
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery, objem odebrane vody v litrech
vystup: *)
VysokyTarifCerpadloOptimalizace[tstart_, tend_, odber_] :=
Module[{ret, trvani, pristi},
 {ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
 If[pristi < tend,
 {If[tstart ≠ pristi, PrestupTepla[tstart, pristi, 0, $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$]],
 NizkyTarifCerpadloOptimalizace[pristi, tend,
 odber - $\frac{\text{pristi} - \text{tstart}}{\text{tend} - \text{tstart}} * \text{odber}$], {PrestupTepla[tstart, tend, 0, odber]}}];
];

```

In[1085]:= (* Vypocita dodany vykon a hmotnost odberu. Nasledne
            tyto parametry preda funkci PrestupTepla. Dale pocita
            spotrebovanou elektrickou energii pri ohrevu vody.
            vstup: cas ohrevu, start casoveho useku, konec casoveho useku,
            odber behem casoveho useku, dodany vykon pri ohrevu teple vody za cas,
            zpusob ohrevu (1 = elektrika, 2 = TC)
            vystup: *)
Ohrev[cas_, tstart_, tend_, odber_, dodanyVykon_, zpusobOhrevu_] :=
Module[{balance, cas1},
  cas1 = cas;
  If[cas1 > 60, {
    PrestupTepla[tstart, tstart + cas1, dodanyVykon,  $\frac{cas1}{tend - tstart} * odber$ ];
    If[zpusobOhrevu == 1,
      dodanaEnergie = Append[dodanaEnergie, {tstart + cas1, dodanyVykon}],
      dodanaEnergie = Append[dodanaEnergie, {tstart + cas1,
        NIntegrate[TCPel[fceTeplo[[den]][t], teplotaOhrivaneVody],
          {t, tstart, tstart + cas1}, AccuracyGoal -> 0]}]}, cas1 = 0];
    If[tend != tstart + cas1 + 1, PrestupTepla[tstart + cas1, tend,
      0,  $\frac{tend - tstart - cas1}{tend - tstart} * odber$ ]];
  ];

```

```

In[1086]:= (* Vypocita, kolik je potreba aktualne dodat tepla pro
ohrati vody v zasobniku na pozadovanou uroven a jak dlouho by to
trvalo. Pokud je tato hodnota mensi nez doba trvani aktualniho
odberu/pauzy mezi odbery a dobou platnosti nizkeho tarifu,
ohreje se cely zasobnik. V opacnem pripade ohrev trva jen do konce
platnosti nizkeho tarifu nebo konce odbery/pauzy mezi odbery.
vstup: casovy usek jednoho odberu nebo pauzy mezi odbery
vystup: logicka hodnota o platnosti nizkeho tarifu (0= ne, 1= ano),
jak dlouho bude jeste nizky tarif trvat, vysledny cas ohrevu *)
VypocetCasu[tstart_, tend_] :=
Module[{tendCerpadlo, QCerpadlo, Q, cas, ret, trvani, pristi},
  Q = c * rho *  $\frac{\text{Pi} * \text{prumer}^2}{4}$  * vyska * (teplotaOhrivaneVody - aktualniTeplota);
  (* kolik tepla je potreba aktualne dodat *)
  QCerpadlo[tendC_?NumericQ] := NIntegrate[TCPTep[fceTeplo[[den]][t],
    teplotaOhrivaneVody], {t, tstart, tendC}, AccuracyGoal -> 0];
  tendCerpadlo = FindRoot[QCerpadlo[tendC] == Q,
    {tendC,  $\frac{Q}{\text{TCPTep}[f\text{ceTeplo}[[\text{den}]] [t\text{start}], \text{teplotaOhrivaneVody}]}$ },
    AccuracyGoal -> 0][[1, 2]];
  {ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
  cas = Min[trvani, tend - tstart - 1, tendCerpadlo - tstart];
  {ret, trvani, cas}];

In[1087]:= (* vypocita, za jak dlouho klesne teplota vody v zasobniku na danou uroven
vstup: zacatek a konec vysetrovaneho casoveho obdobi,
objem odebrane vody za dane obodbi, minimalni mozna teplota
vystup: cas odkladu*)
VypocetCasuOdkladu[tstart_, tend_, objem_, teplota_] :=
Module[{interval, cas},
  interval = 100;
  cas = 1;
  While[interval >= 1,
    While[VetaTermodynamiky[cas + interval,
      0,  $\frac{10^{(-3)} * \text{objem} * \rho}{\text{tend} - \text{tstart}}$ , aktualniTeplota] > teplota,
      cas += interval;
    ];
    interval /= 10;
  ];
  cas]

```


In[1088]= Funkce pro odběr a ohřev vody;

```
(* na zaklade varianty ohrevu zavola prislusnou
funkci pro ohrev (NizkyTarifxxx nebo VysokyTarifxxx)
pro samodny odber a pote i pro pauzu mezi dvema odbery
vstup: cas a konec jednoho odberu, objem odebrane teple vody,
cas a konec jedne pauzy mezi odbery, varianta ohrev (1 - 4)
vystup: *)
Ohrivej[tstartOdber_, tendOdber_, objemOdber_, tstartOhrev_,
tendOhrev_, varianta_] := Module[{Q, ret, trvani, pristi},
If[varianta == 1, {{ret, trvani, pristi} = NizkyTarifOhrevuVody[tstartOdber];
If[ret != 0, NizkyTarifBojler[tstartOdber, tendOdber, objemOdber],
VysokyTarifBojler[tstartOdber, tendOdber, objemOdber]],
{ret, trvani, pristi} = NizkyTarifOhrevuVody[tstartOhrev];
If[ret != 0, NizkyTarifBojler[tstartOhrev, tendOhrev, 0],
VysokyTarifBojler[tstartOhrev, tendOhrev, 0]];]];

If[varianta == 2,
{{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOdber];
If[ret != 0, NizkyTarifCerpadlo[tstartOdber, tendOdber, objemOdber],
VysokyTarifCerpadlo[tstartOdber, tendOdber, objemOdber]],
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOhrev];
If[ret != 0, NizkyTarifCerpadlo[tstartOhrev, tendOhrev, 0],
VysokyTarifCerpadlo[tstartOhrev, tendOhrev, 0]];]];

If[varianta == 3,
{{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOdber];
If[ret != 0, NizkyTarifCerpadloOffset[tstartOdber, tendOdber, objemOdber],
VysokyTarifCerpadloOffset[tstartOdber, tendOdber, objemOdber]],
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOhrev];
If[ret != 0, NizkyTarifCerpadloOffset[tstartOhrev, tendOhrev, 0],
VysokyTarifCerpadloOffset[tstartOhrev, tendOhrev, 0]];]];

If[varianta == 4,
{{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOdber];
If[ret != 0, NizkyTarifCerpadloOptimalizace[
tstartOdber, tendOdber, objemOdber],
VysokyTarifCerpadloOptimalizace[tstartOdber, tendOdber, objemOdber]],
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpada[tstartOhrev];
If[ret != 0, NizkyTarifCerpadloOptimalizace[tstartOhrev, tendOhrev, 0],
VysokyTarifCerpadloOptimalizace[tstartOhrev, tendOhrev, 0]];]];
];
```

```

In[1090]:= (* nacte odbery a vola funkce pro ohrev vody,
tvori funkce o stavu teple vody v zasobniku v case
vstup: varianta ohrevu
vystup: *)
NactiOdbery[varianta_] := Module[{fun, ret, trvani, pristi, Q, cas, ohrev},
  If[varianta == 4, {VypocetCasuOptimalizace[]}];
  Ohrivej[0, 1, 0, 1, odbery[[2, 1]], varianta];
  Ohrivej[odbery[[#, 1]], odbery[[#, 2]], odbery[[#, 3]], odbery[[#, 2]],
    odbery[[# + 1, 1]], varianta] & /@ Range[2, Length[odbery] - 1];
  Ohrivej[odbery[[Length[odbery], 1]], odbery[[Length[odbery], 2]], odbery[[
    Length[odbery], 3]], odbery[[Length[odbery], 2]], 24 * 3600, varianta];

  {odberyVCase = Append[odberyVCase, {odbery[[#, 1]] - 1, 0}];
  odberyVCase =
    Append[odberyVCase, {odbery[[#, 1]],  $\frac{\text{odbery}[[\#, 3]]}{\text{odbery}[[\#, 2]] - \text{odbery}[[\#, 1]]}$ }]};
  odberyVCase = Append[odberyVCase, {odbery[[#, 2]],
     $\frac{\text{odbery}[[\#, 3]]}{\text{odbery}[[\#, 2]] - \text{odbery}[[\#, 1]]}$ }]};
  odberyVCase = Append[odberyVCase, {odbery[[#, 2]] + 1, 0}];} & /@
  Range[2, Length[odbery]];
  odberyVCase = Append[odberyVCase, {24 * 3600, 0}];

  fceOdbery = Interpolation[odberyVCase, InterpolationOrder → 1];
  fceTeplaVoda = Interpolation[teplotaVodyVCase, InterpolationOrder → 1];
  fceDodatTepla = Interpolation[dodatTeplaVCase, InterpolationOrder → 1];
];

```

In[1091]:= **Funkce udávající platnost jednotlivých distribučních tarifů elektřiny;**

(* rika, zda je aktualne platny nizky tarif
distribucni sazby elektriny pro tepelna cerpadla
vstup: cas startu tepelneho cerpadla
vystup: logicka hodnota zda je aktualne platny nizky tarif (1- ano, 0- ne),
jak dlouho jeste bude nizky tarif trvat,
pokud neni platny, tak za jak dlouho bude platny *)
NizkyTarifTepelnehoCerpada[tstart_] := Module[{ret, pristi, trvani},

```
ret = 0;
pristi = 0;
trvani = 0;
If[0 ≤ tstart < 6 * 3600, {ret = 1, trvani = 6 * 3600 - tstart}];
If[6 * 3600 ≤ tstart < 7 * 3600, pristi = 7 * 3600];
If[7 * 3600 ≤ tstart < 9 * 3600, {ret = 1, trvani = 9 * 3600 - tstart}];
If[9 * 3600 ≤ tstart < 10 * 3600, pristi = 10 * 3600];
If[10 * 3600 ≤ tstart < 13 * 3600, {ret = 1, trvani = 13 * 3600 - tstart}];
If[13 * 3600 ≤ tstart < 14 * 3600, pristi = 14 * 3600];
If[14 * 3600 ≤ tstart < 16 * 3600, {ret = 1, trvani = 16 * 3600 - tstart}];
If[16 * 3600 ≤ tstart < 17 * 3600, pristi = 17 * 3600];
If[17 * 3600 ≤ tstart < 24 * 3600, {ret = 1, trvani = 24 * 3600 - tstart}];
{ret, trvani, pristi}];
```

In[1093]:= (* rika, zda je aktualne platny nizky
tarif distribucni sazby elektriny pro bojleru
vstup: cas startu bojleru
vystup: logicka hodnota zda je aktualne platny nizky tarif (1- ano, 0- ne),
jak dlouho jeste bude nizky tarif trvat,
pokud neni platny, tak za jak dlouho bude platny *)

```
NizkyTarifOhrevuVody[tstart_] := Module[{ret, pristi, trvani},
ret = 0;
pristi = 0;
trvani = 0;
If[0 ≤ tstart < 6 * 3600, {ret = 1, trvani = 6 * 3600 - tstart}];
If[6 * 3600 ≤ tstart < 19 * 3600, pristi = 19 * 3600];
If[19 * 3600 ≤ tstart < 21 * 3600, {ret = 1, trvani = 21 * 3600 - tstart}];
If[21 * 3600 ≤ tstart < 24 * 3600, pristi = 24 * 3600];
{ret, trvani, pristi}];
```

```

In[1094]:= Načtení výkonu tepelného čerpadla a denních teplot;
(* nacte vykonove parametry tepelneho cerpadla,
vytvori funkce pro prikoni, vykon a COP
vstup:
vystup: *)
NactiTepelneCerpadlo[] :=
Module[{T, n, Ptep45, Ptep55, Ptep60, Pel45, Pel55, Pel60, radekEl45, radekEl55,
radekEl60, radekTep45, radekTep55, radekTep60, dataEl, dataTep},
(* TC vzduch-voda Vitocal 222-A *)
(* https://www.viessmann.cz/cs/obytno-budovy/tepelna-
cerpadla/tepelna-cerpadla-vzduchvoda/vitocal-222a.html *)
T = {-20, -15, -7, 2, 7, 10, 20, 30};
n = Length[T];
Ptep45 = {3.88, 4.38, 5.41, 5.43, 5.06, 6.65, 7.85, 8.93};
Ptep55 = {0, 4.47, 5.56, 2.88, 4.4, 4.53, 5.58, 6.95};
Ptep60 = {0, 0, 5.12, 3.01, 4.14, 4.44, 5.48, 6.81};
Pel45 = {2.21, 2.29, 2.4, 1.06, 1.28, 1.26, 1.25, 1.22};
Pel55 = {10^-6, 2.67, 2.84, 1.33, 1.51, 1.57, 1.57, 1.55};
Pel60 = {10^-6, 10^-6, 2.89, 1.49, 1.75, 1.76, 1.77, 1.76};

(*vytvoreni listu dat s hodnotami*)
radekEl45[i_] := {T[[i]], 45., Pel45[[i]]};
radekEl55[i_] := {T[[i]], 55., Pel55[[i]]};
radekEl60[i_] := {T[[i]], 60., Pel60[[i]]};
radekTep45[i_] := {T[[i]], 45., Ptep45[[i]]};
radekTep55[i_] := {T[[i]], 55., Ptep55[[i]]};
radekTep60[i_] := {T[[i]], 60., Ptep60[[i]]};
dataEl =
Flatten[Table[{radekEl45[i], radekEl55[i], radekEl60[i]}, {i, n}], 1];
dataTep = Flatten[Table[{radekTep45[i], radekTep55[i],
radekTep60[i]}, {i, n}], 1];

(*vyjadreni prubehu zavislosti energetickych hodnot na Tvzduch, Tvoda*)
(*Elektricky prikon*)
TCPel[Tvzduch_, Tvoda_] =
10^3 * Interpolation[dataEl, InterpolationOrder -> {1, 1}][Tvzduch, Tvoda];
(*Topny vykon*)
TCPtep[Tvzduch_, Tvoda_] =
10^3 * Interpolation[dataTep, InterpolationOrder -> {1, 1}][Tvzduch, Tvoda];
(*Topny faktor*)
TCCOP[Tvzduch_, Tvoda_] =  $\frac{TCPtep[Tvzduch, Tvoda]}{TCPel[Tvzduch, Tvoda]}$ ;
];

```

In[1096]:=

```
(* nacte hodinove teploty v dany den roku 2020 a vytvori z nich casove funkce
vstup:
vystup: *)
NactiVenkovniTeploty[] := Module[{T, index},
  T = 3600 * Range[0, 24];
  teplotyPole =
    teploty[#[#, Range[2, Length[teploty[[1]]]]] & /@Range[1, Length[teploty]];
  index = 1;
  While[index ≤ Length[teplotyPole],
    fceTeplot = Append[fceTeplot, Interpolation[
      {T[[#]], teplotyPole[[index, #]]} & /@Range[1, Length[T]]];
    index += 1;
  ];
];
```

In[1097]:=

```
(* simuluje funkci zasobniku teple vody za jeden den
vstup: varianta ohrevu
vystup: *)
SimulujDen[varianta_] := Module[{celkovaDodanaEnergie},
  Print["Prave bezi den ", den];
  teplotaVodyVCase = {{0, aktualniTeplota}};
  dodatTeplaVCase = {{0, dodatTepla}};
  odberyVCase = {{0, 0}};
  dodanaEnergie = {{0, 0}};
  casyOhrevu = {};
  NactiOdbery[varianta];
  celkovaDodanaEnergie =
    
$$\frac{1}{3.6 * 10^6} * \text{Total}[dodanaEnergie[[#, 2]] \& /@ \text{Range}[\text{Length}[dodanaEnergie]]];$$

  Print[ZobrazfceTeplaVoda[]];
  Print[ZobrazfceOdbery[]];
  Print["Celková dodaná energie za ",
    den, ". den je ", celkovaDodanaEnergie, " kWh."];
  Print[""];
  den = den + 1;
  Qtotal += celkovaDodanaEnergie;
];
```

In[1098]:=

Funkce pro výpočet ideálního času ohřevu ;

```

In[1099]:= (* vypocita idealni casy ohrevu dle algoritmu popsaneho v praci
vstup:
vystup: *)
VypocetCasuOptimalizace[] :=
Module[{min, hodnoticiFunkceBody, hodnoticiFunkce, ret, trvani, pristi, tstart,
tstartfinal, tendfinal, VTtstart, VTtend, lokalniMaximaNeserazene,
lokalniMaxima, odberyCelkem, Q, ohrevStart, ohrevEnd,
ohrev, hodnoceniList, hodnoceni, QCerpadlo},
hodnoticiFunkceBody = (10 * TCCOP[fceTeplot[[den]][[#]], teplotaOhrivaneVody]) ^
2 & /@ Range[0, 24 * 3600 - 1];
min = Min[hodnoticiFunkceBody];
hodnoticiFunkceBody -= 0.9 * min;
tstart = 0;
odberyCelkem = 0;
While[tstart < 24 * 3600,
{ret, trvani, pristi} = NizkyTarifTepelnehoCerpadla[tstart];
If[ret == 1, {VTtstart = tstart + trvani, tstart = VTtstart},
{VTtend = pristi, tstart = VTtend}];
If[VTtstart < VTtend, hodnoticiFunkceBody[[#]] = 0; & /@
Range[VTtstart, VTtend]];
];
lokalniMaximaNeserazene = FindPeaks[hodnoticiFunkceBody];
lokalniMaxima = Sort[lokalniMaximaNeserazene, #1[[2]] > #2[[2]] &];
If[odbery[[#, 2]] <= lokalniMaxima[[1, 1]], odberyCelkem += odbery[[#, 3]] & /@
Range[Length[odbery]]];
If[odbery[[#, 1]] < lokalniMaxima[[1, 1]] &&
odbery[[#, 2]] > lokalniMaxima[[1, 1]],
odberyCelkem +=  $\frac{\text{lokalniMaxima}[[1, 1]] - \text{odbery}[[\#, 1]]}{\text{odbery}[[\#, 2]] - \text{odbery}[[\#, 1]]} * \text{odbery}[[\#, 3]]$  & /@
Range[Length[odbery]]];
Q = 10^(-3) *  $\rho$  * objem * c * (teplotaOhrivaneVody - VetaTermodynamiky[
lokalniMaxima[[1, 1]], 0,  $\frac{10^(-3) * \text{odberyCelkem} * \rho}{\text{lokalniMaxima}[[1, 1]]}$ , aktualniTeplota]);
hodnoticiFunkce = Interpolation[hodnoticiFunkceBody];
casyOhrevu = NajdiIntervaly[hodnoticiFunkce, Q, lokalniMaxima];
casyOhrevu = SortBy[casyOhrevu, Last];
Print["casyOhrevu ", casyOhrevu];
];

```

```

In[1100]:= (* hleda casove intervaly ohrevu takove, aby soucet dodaneho
            tepelneho vykonu v danym intervalech byl roven pozadovanemu teplu Q
            vstup: hodnotici funkce, pozadovane teplo v J,
            seznam lokalnich maxim hodnotici funkce
            vystup: seznam intervalu casu ohrevu *)
NajdiIntervaly[hodnoticiFunkce_, Q_, lokalniMaxima_] :=
Module[{hranice, interval, casy, vystup},
  hranice = lokalniMaxima[[1]];
  interval = 10;
  vystup = {};
  While[interval >= 10^-3,
    While[Qdodane[hranice[[2]] - interval,
      hodnoticiFunkce, lokalniMaxima][[1]] < Q * 1.05,
      vystup = Qdodane[hranice[[2]], hodnoticiFunkce, lokalniMaxima];
      hranice -= interval;
    ];
    interval /= 10;
  ];
  vystup[[2]]];

In[1101]:= (* pocita teplo, ktere by se mohlo dodat v casovych intervalech,
            kde ma hodnotici funkce vyssi hodnotu nez hranice
            vstup:
            hranicni hodnota hodnotici funkce nutne k zapnuti/vypnuti tepelneho cerpadla
            vystup: dodane teplo, seznam intervalu casu ohrevu *)
Qdodane[hranice_, hodnoticiFunkce_, lokalniMaxima_] :=
Module[{Q, casy, maxima},
  casy = {};
  maxima = {};
  Q = 0;
  If[hranice < lokalniMaxima[[#, 2]], maxima =
    Append[maxima, lokalniMaxima[[#]]] & /@ Range[Length[lokalniMaxima]];
  casy = Append[casy, NajdiOhrevStartaOhrevEndVlastni[hodnoticiFunkce,
    maxima[[#]], hranice] & /@ Range[Length[maxima]]];
  casy = casy[[1]];
  Q += NIntegrate[TCPtep[fceTeplo[[den]][t], teplotaOhrivaneVody], {t,
    casy[[#, 1]], casy[[#, 2]]}, AccuracyGoal -> 0] & /@ Range[Length[casy]];
  {Q,
  casy}];

```

```

In[1102]:= (* funkce podobna funkci FindRoot, jen spolehlivejsi a rychlejsi
vstup: hodnotici funkce,
maximum v okoli ktereho si hleda dany casovy interval ohrevu,
minimalni hodnota hodnotici funkce,
ktera tvori okraje hledaneho casoveho intervalu
vystup: start a konec nalezeneho casoveho intervalu ohrevu *)
NajdiOhrevStartaOhrevEndVlastni[hodnoticiFunkce_, maximum_, hranice_] :=
Module[{ohrevStart, ohrevEnd, interval},
interval = 100;
ohrevStart = maximum[[1]];
ohrevEnd = maximum[[1]];
While[interval >= 1,
While[
hodnoticiFunkce[ohrevStart - interval] > hranice && ohrevStart - interval >= 0,
ohrevStart -= interval;
];
interval /= 10;
];
interval = 100;
While[interval >= 1,
While[
hodnoticiFunkce[ohrevEnd + interval] > hranice && ohrevEnd + interval <= 86400,
ohrevEnd += interval;
];
interval /= 10;
];
{ohrevStart, ohrevEnd}];

```

Ovládání celého programu ;

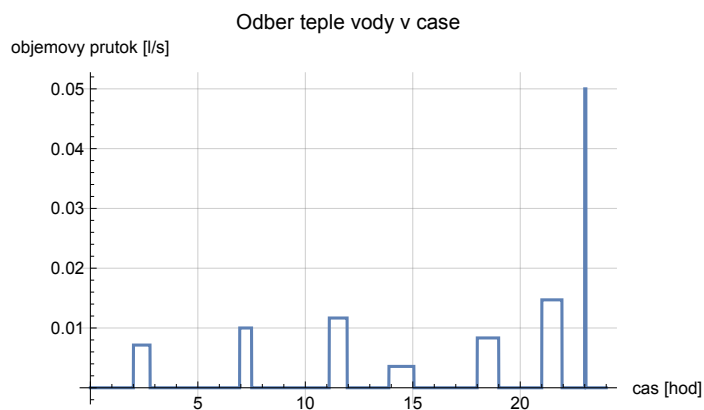
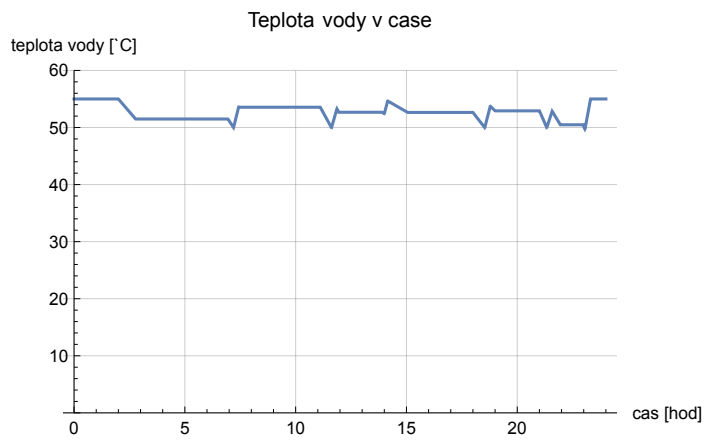
```

NactiTepelneCerpadlo[]
NactiVenkovniTeploty[]
For[i = 0, i < 1, i++, {SimulujDen[4]}];
Print[Qtotal];

```


Prave bezi den 1

casyOhrevu {{50 401, 53 113}}



Celková dodaná energie za 1. den je 3.69331 kWh.

3.69331