```
(*Parameters of ditributions to be modelled*)
(*Parameter of generalised distribution in the numerator*)
rn = {3};
λn = {2};
δn = {2};

(*Generalised gammas in the denominator*)
rd = {5};
λd = {10};
δd = {1/4};
(*Precision parameter for near-exact distributions*)
precision = 20;

(*Defining modules which are use in the first near-exact,
second near-exact and simulated distribution*)
(************************
 Start: Emperical distribution*****************************)
(*Simulation random variates of Z=
 logY where Y is a product of independent generalises gamma random variables*)
(*Notations: (r,λ,δ) are parameters of generalised gamma
  random variabl. sz is a sample size, e.g For 10 000 sz=3  *)
ran[r_, λ_, δ_, sz_] := RandomVariate[GammaDistribution[r, λ, δ, 0], 10^sz];
random[shape_, rate_, power_, sz_] := Module[{n, array, simul},
n = Length[shape];
array = Table[ran[shape[[j]], rate[[j]], power[[j]], sz], {j, 1, n}];
Table[Product[array[[j, t]], {j, 1, n}], {t, 1, 10^sz}]];
(************************
 End: Emperical distribution*****************************)

(*********************************
 Start: Preparing parameters********************)
ParameterPrep[rn_, rd_, λn_, λd_, δn_, δd_, precision_] :=
 Module[{exppar, l, GIGshape, uni, Uniexppar,
   ml, a, unipar, lneg, lpos, z1ExactCha, z1AppChar},
Clear[r, λ, δ];
r = rn; λ = λn; δ = δn;
Do[r = Append[r, rd[[k]]], {k, 1, Length[rd]}];
Do[λ = Append[λ, λd[[k]]^-1], {k, 1, Length[λd]}];
Do[δ = Append[δ, -δd[[k]]], {k, 1, Length[δd]}];
n = Length[r];
(****************************************************************************)
```

```
(*Determining SDGIG parameters*)
(*Calculating exponetial parameters
   and counting number of equal exponetial parameters*)
Clear[m];
exppar =
   Sort[Flatten[Table[Table[δ[[j]] * (r[[j]] + k), {j, 1, n}], {k, 0, precision - 1}]]];
  l = Length[exppar];
m = Table[1, l]; GIGshape = {1}; a = 1;
Do[If[exppar[[t]] == exppar[[t - 1]], a = m[[t - 1]] + 1, a = 1];
   m[[t]] = a;, {t, 2, l}];
  ml = Length[m];
  uni = {1};
  Do[If[m[[k + 1]] <= m[[k]], uni = Append[uni, k], uni = uni], {k, 1, ml - 1}];
  uni = Append[uni, ml];
  uni = uni[[2 ;; Length[uni]]];
Uniexppar = exppar[[uni]];
m = m[[uni]];

(*Separating negative parameters from positive exponetial parameters*)
unipar = Length[Uniexppar]; lneg = 0;
Do[If[Uniexppar[[k]] < 0, lneg = lneg + 1, lneg = lneg + 0], {k, 1, unipar}];
lpos = unipar - lneg;
Bneg = Uniexppar[[1 ;; lneg]];
mneg = m[[1 ;; lneg]];
Bpos = Uniexppar[[lneg + 1 ;; unipar]];
mpos = m[[lneg + 1 ;; unipar]];
Clear[θ1];
```

$$\text{z1ExactCha[t\_]} = \prod_{j=1}^{n} \frac{\text{Gamma}\left[\text{r[[j]]} + \text{precision} - \frac{\text{I*t}}{\delta\text{[[j]]}}\right]}{\text{Gamma[r[[j]] + precision]}};$$

```
(*Parameter for gamma estimating Z1*)
θ1 = SetPrecision[(I^(-1)) * D[z1ExactCha[t], t] /. t → 0, 250];
```

$$\text{Shift1} = \left(\theta1 + \sum_{j=1}^{n} \text{Log}[\lambda\text{[[j]]]}\right);$$

```
Clear[ρ, ℓ, θ2];
```

$$\text{z1AppChar[t\_]} = \left(1 - \frac{\text{t * I}}{\ell}\right)^{-\rho} * e^{\text{I*t*}\theta 2};$$

```
mom =
   Table[SetPrecision[I^(-h) * D[z1ExactCha[t], {t, h}] /. t → 0, 250], {h, 1, 3}];
mom1 = Table[SetPrecision[I^(-h) * D[z1AppChar[t], {t, h}] /. t → 0, 250], {h, 1, 3}];
{ρ, ℓ, θ2} = {ρ, ℓ, θ2} /. Flatten[Solve[{mom[[1]] == mom1[[1]],
       mom[[2]] == mom1[[2]], mom[[3]] == mom1[[3]]}, {ρ, ℓ, θ2}]];
```

```
Shift2 = θ2 + Sum[Log[λ[[j]]], {j, 1, n}];]

(*********************************
 End: Preparing parameters********************)

(*module to calculate K-factors*)
FactorK[λ_, r_] := Module[{kfactor, g},
g = Length[λ];
  kfactor = Product[λ[[t]]^r[[t]], {t, 1, g}]]
(****************************************************************************)

(*module to calculate c[[i,k]] factors*)
cik[t_, r_, λ_] := Module[{cc, R, g, aa, ri},
g = Length[λ]; ri = r[[t]]; cc = Range[ri];
cc[[ri]] = (1/(ri - 1) !) * (Product[(λ[[j]] - λ[[t]])^-r[[j]], {j, 1, t-1}]) * (Product[(λ[[j]] - λ[[t]])^-r[[j]], {j, t+1, g}]);
R[n_, j_] := Sum[r[[s]] * (λ[[j]] - λ[[s]])^(-n-1), {s, 1, j-1}] + Sum[r[[s]] * (λ[[j]] - λ[[s]])^(-n-1), {s, j+1, g}];
Do[cc[[ri - k]] =
    1/k (Sum[(((ri - k + j - 1) !/(ri - k - 1) !) * R[j - 1, t] * cc[[ri - (k - j)]]), {j, 1, k}]), {k, 1, ri - 1}];
  cc]
(****************************************************************************)


(*cdf and pdf of GNIG cdf for independent erland and gamma*)
GNIGcdf[z_, elλ_, Γλ_, elr_, Γr_] := Module[{CDFelang, secterm, cdfFz, Test},
CDFelang = (Γλ * z)^Γr/Gamma[Γr + 1] * Hypergeometric1F1[Γr, Γr + 1, -Γλ * z];
secterm = Sum[elλ^t * z^(t+Γr)/Gamma[t + Γr + 1] Hypergeometric1F1[Γr, t + Γr + 1, -(Γλ - elλ) * z], {t, 0, elr-1}];
secterm = Γλ^Γr * ⅇ^(-elλ*z) * secterm;
cdfFz = CDFelang - secterm]

GNIGpdf[z_, elλ_, Γλ_, elr_, Γr_] := Module[{secterm},
c = Table[cik[q, elr, elλ], {q, 1, 1}];
  secterm =
   Sum[c[[1, k]] Gamma[k]/Gamma[k + Γr] * z^(k+Γr-1) Hypergeometric1F1[Γr, k + Γr, -(Γλ - elλ) * z], {k, 1, elr[[1]]}];
  secterm = elλ^elr * Γλ^Γr * ⅇ^(-elλ*z) * secterm]
```

```
(****************************************************************************)

(*cdf for difference between independent erland and gamma*)
DGamErlangcdf[z_, elλ_, Γλ_, elr_, Γr_] :=
 Module[{Term11, Term12, Term21, Term22, cdfFz},
Term11 = Sum[elλ^t/t! Sum[Binomial[t, k] * (-z)^k Gamma[t - k + Γr, (Γλ + elλ) * z]/(Γλ + elλ)^(t-k+Γr), {k, 0, t}], {t, 0, elr-1}];
Term12 = 1 - Gamma[Γr, Γλ * z]/Gamma[Γr] + Γλ^Γr/Gamma[Γr] * E^(elλ*z) * Term11;
Term21 = Sum[elλ^t/t! Sum[Binomial[t, k] * (-z)^k Gamma[t - k + Γr]/(Γλ + elλ)^(t-k+Γr), {k, 0, t}], {t, 0, elr-1}];
Term22 = Γλ^Γr/Gamma[Γr] * E^(elλ*z) * Term21;
cdfFz = If[z ≥ 0, Term12, Term22]]
(****************************************************************************)

(*First near-exact:cdf of independent SDGIG*)
SDGIGcdf[z_, r1_, r2_, λ1_, λ2_] :=
 Module[{p1, p2, k1, k2, c, d, term11, term12, term21},
p1 = Length[λ1];
p2 = Length[λ2];
k1 = FactorK[λ1, r1];
  k2 = FactorK[λ2, r2];
c = Table[cik[q, r1, λ1], {q, 1, p1}];
d = Table[cik[q, r2, λ2], {q, 1, p2}];
term11 = Sum[((k1 * k2) * d[[l, h]])/λ2[[l]]^(h-t)
     Sum[c[[j, k]] * (h - 1)!/t! * (k + t - 1)!/(λ2[[l]] + λ1[[j]])^(k+t), {j, 1, p1}, {k, 1, r1[[j]]}], {l, 1, p2}, {h, 1, r2[[l]]}, {t, 0, h-1}];
term12 = Sum[((k1 * k2) * c[[j, k]])/λ1[[j]]^(k-t) Sum[d[[l, h]] * (k - 1)!/t! *
       (h + t - 1)!/(λ2[[l]] + λ1[[j]])^(h+t), {l, 1, p2}, {h, 1, r2[[l]]}] * CDF[ErlangDistribution[k - t, λ1[[j]]], z], {j, 1, p1}, {k, 1, r1[[j]]}, {t, 0, k-1}];
  term21 = Sum[((k1 * k2) * d[[l, h]])/λ2[[l]]^(h-t) Sum[c[[j, k]] * (h - 1)!/t! *
       (k + t - 1)!/(λ2[[l]] + λ1[[j]])^(k+t), {j, 1, p1}, {k, 1, r1[[j]]}] * CDF[ErlangDistribution[h - t, λ2[[l]]], -(z)], {l, 1, p2}, {h, 1, r2[[l]]}, {t, 0, h-1}];
If[z >= 0, term11 + term12, -term21 + term11]]
(****************************************************************************)
```

```
(*Second near-
 exact:cdf of independent SDGIG plus gamma of positive shape parameter*)
SDGIGcdfpos[z_, r1_, r2_, λ1_, λ2_, Γλ_, Γr_] :=
 Module[{p1, p2, k1, k2, c, d, sum1pos, sum2pos, cdf2, sumpos2, sum2neg},
p1 = Length[λ1];
p2 = Length[λ2];
k1 = FactorK[λ1, r1];
  k2 = FactorK[λ2, r2];
c = Table[cik[q, r1, λ1], {q, 1, p1}];
d = Table[cik[q, r2, λ2], {q, 1, p2}];
sum1pos = Sum[Sum[Sum[((k1 * k2) * c[[j, k]] / λ1[[j]]^(k-t) Sum[Sum[d[[l, h]] * (k - 1)! / t! *
      (h + t - 1)! / (λ2[[l]] + λ1[[j]])^(h+t), {h, 1, r2[[l]]}], {l, 1, p2}]) * GNIGcdf[z, λ1[[j]], Γλ, k - t, Γr], {t, 0, k-1}], {k, 1, r1[[j]]}], {j, 1, p1}];
sum2pos = Sum[Sum[Sum[((k1 * k2) * d[[l, h]] / λ2[[l]]^(h-t) Sum[Sum[c[[j, k]] * (h - 1)! / t! *
      (k + t - 1)! / (λ2[[l]] + λ1[[j]])^(k+t), {k, 1, r1[[j]]}], {j, 1, p1}]) * DGamErlangcdf[z, λ2[[l]], Γλ, h - t, Γr], {t, 0, h-1}], {h, 1, r2[[l]]}], {l, 1, p2}];
  sum2neg = Sum[Sum[Sum[((k1 * k2) * d[[l, h]] / λ2[[l]]^(h-t) Sum[Sum[c[[j, k]] * (h - 1)! / t! *
      (k + t - 1)! / (λ2[[l]] + λ1[[j]])^(k+t), {k, 1, r1[[j]]}], {j, 1, p1}]) * DGamErlangcdf[z, λ2[[l]], Γλ, h - t, Γr], {t, 0, h-1}], {h, 1, r2[[l]]}], {l, 1, p2}];
sumpos2 = sum1pos + sum2pos; cdf2 = If[z >= 0, sumpos2, sum2neg]]
(***************************************************************************)
```

```
(*Second near-
 exact:cdf of independent SDGIG plus gamma of negative shape parameter*)
SDGIGcdfneg[z_, r1_, r2_, λ1_, λ2_, Γλ_, Γr_] :=
 Module[{nΓλ, p1, p2, k1, k2, c, d, term1, term2, term3, cdf2}, nΓλ = Abs[Γλ];
p1 = Length[λ1];
p2 = Length[λ2];
k1 = FactorK[λ1, r1];
  k2 = FactorK[λ2, r2];
c = Table[cik[q, r1, λ1], {q, 1, p1}];
d = Table[cik[q, r2, λ2], {q, 1, p2}];
```

$$\text{term1} = \sum_{j=1}^{p1}\sum_{k=1}^{r1[[j]]}\sum_{t=0}^{k-1}\left(\frac{(k1*k2)*c[[j,k]]}{\lambda1[[j]]^{k-t}}\sum_{l=1}^{p2}\sum_{h=1}^{r2[[l]]} d[[l,h]]*\frac{(k-1)!}{t!}*\frac{(h+t-1)!}{(\lambda2[[l]]+\lambda1[[j]])^{h+t}}\right)*$$

```
        DGamErlangcdf[-z, λ1[[j]], nΓλ, k - t, Γr];
```

$$\text{term2} = \sum_{j=1}^{p1}\sum_{k=1}^{r1[[j]]}\sum_{t=0}^{k-1}\left(\frac{(k1*k2)*c[[j,k]]}{\lambda1[[j]]^{k-t}}\sum_{l=1}^{p2}\sum_{h=1}^{r2[[l]]} d[[l,h]]*\frac{(k-1)!}{t!}*\frac{(h+t-1)!}{(\lambda2[[l]]+\lambda1[[j]])^{h+t}}\right)*\text{DGamErlangcdf}[-z,\lambda1[[j]],n\Gamma\lambda,k-t,\Gamma r];$$

$$\text{term3} = \sum_{l=1}^{p2}\sum_{h=1}^{r2[[l]]}\sum_{t=0}^{h-1}\left(\frac{(k1*k2)*d[[l,h]]}{\lambda2[[l]]^{h-t}}\sum_{j=1}^{p1}\sum_{k=1}^{r1[[j]]} c[[j,k]]*\frac{(h-1)!}{t!}*\frac{(k+t-1)!}{(\lambda2[[l]]+\lambda1[[j]])^{k+t}}\right)*\text{GNIGcdf}[-z,\lambda2[[l]],n\Gamma\lambda,h-t,\Gamma r];$$

```
cdf2 = If[z >= 0, 1 - term1, 1 - term2 - term3]]
(***************************************************************************)

(*Second near-exact:cdf of independent SDGIG plus gamma*)
SDGIGcdf2[z_, r1_, r2_, λ1_, λ2_, Γλ_, Γr_] := If[nΓλ >= 0,
  SDGIGcdfpos[z, r1, r2, λ1, λ2, Γλ, Γr], SDGIGcdfneg[z, r1, r2, λ1, λ2, Γλ, Γr]]
(***************************************************************************)
```